



elektor

EMBEDDED LINUX

Software-Entwicklung

+ Elektronik-Grundlagen
LED-Blinker und Oszillatoren

Nunchuk USB-Interface
Game-Controller als Eingabegerät

+ Modellbahn-Steuerung
Mit USB-Anschluss und Skript-Editor

+ USB-Isolator
Nie wieder Ärger mit Masseschleifen!





DAS ORIGINAL SEIT 1994
PCB-POOL
 Beta LAYOUT

FREE Stencil
 bei jeder PCB Prototyp-Bestellung



Easy-going
 17 akzeptierte Layoutformate

www.pcb-pool.com



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



eSTORE
 Beta LAYOUT

Entwickeln, Bestücken und Löten



€ 36,50*

Arduino Mega (ATMega 1280-16AU)
 kompatibel

Reflow-Controller



€ 129,00*

LED Wechselblinker
 SMD-Bausatz



€ 6,00*

Big Beta-Reflow-Kit



€ 129,00*

Tool-Kit Extended



€ 149,00*

* inkl. MwSt und zzgl. Versandkosten

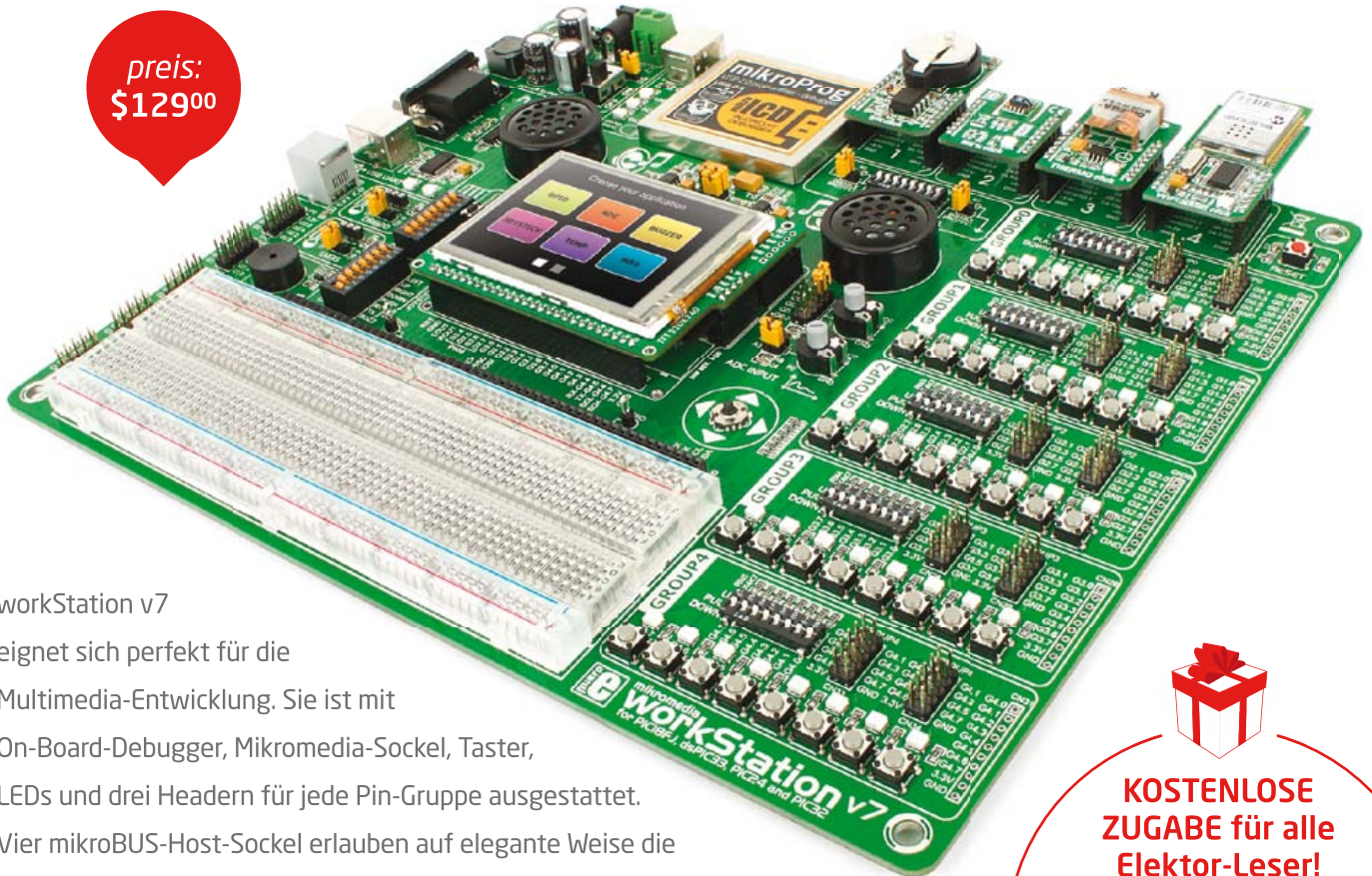
www.beta-eSTORE.com

Beta
 LAYOUT
 create:electronics

mikromedia workStation^{v7}

für PIC18FJ®, dsPIC33®, PIC24® und PIC32®

preis:
\$129⁰⁰



workStation v7

eignet sich perfekt für die

Multimedia-Entwicklung. Sie ist mit

On-Board-Debugger, Mikromedia-Sockel, Taster,

LEDs und drei Headern für jede Pin-Gruppe ausgestattet.

Vier mikroBUS-Host-Sockel erlauben auf elegante Weise die

Erweiterung mit Click-Boards. Das große Steckbrett bietet Raum für eigene

experimentelle Elektronik. Dieses schon fast Wunder-Board eignet sich auch sehr

gut für die Ausbildung. Selbstverständlich wird eine große Palette an Libraries und

Beispielen für unsere mikroC, mikroBasic und mikroPascal Compiler mitgeliefert, die

eigene Entwicklungen vereinfachen.



**KOSTENLOSE
ZUGABE für alle
Elektor-Leser!**

ELEKTOR-09-2012

Am Ende des Bestellformulars kann der
obige Code für eine KOSTENLOSE
Lieferung eingegeben werden!

*FÜR ALLE BESTELLUNGEN

Der Code ist gültig bis 14. September 2012.

PIC18FJ, dsPIC33, PIC24 oder PIC32?

workStation v7 unterstützt alle Mikromedia-Boards mit Mikrocontrollern
von Microchip®. Die gewünschten Boards einfach einstecken.

MikroElektronika
DEVELOPMENT TOOLS | COMPILERS | BOOKS

www.libstock.com

GET IT NOW
www.mikroe.com

* Die große Abbildung zeigt: Mikromedia workStation v7, Mikromedia für PIC32, WiFi-PLUS-Click, THERMO-Click, RTC2-Click und SHT11-Click.

** mikromedia- und Click-Boards sind als optionales Zubehör erhältlich!



Elektor – live!

Manchmal sind auch die Niederländer ganz vorn: Schon vier Mal haben wir in Eindhoven unsere Hausmesse „Elektor-Live!“ veranstaltet. Zwei Mal war ich persönlich dabei, im letzten Jahr durfte ich als „Hilfs-Entwickler“ (neben meinen Kollegen aus dem Labor) ganz stolz den AndroPod und verschiedene ElektorBus-Platinen an einem eigenen kleinen Stand präsentieren. Beide Male war es ein recht langer Messe-Tag, doch wenn man sich abends ins Auto setzt, fühlt man auch, wie viel Spaß das Ganze gemacht hat. Leser „in echt“ treffen, den Geschichten der Elektroniker zuzuhören, direkt Lob, Wünsche und Kritik zu hören und dabei auch ein wenig die Stimmung in der „Szene“ einfangen, das kann man eben nur auf so einer Messe. Und die Leser erfahren umgekehrt aus erster Hand, was sich bei uns tut und was noch alles geplant ist. Elektor-Boards und -Tools zum Anfassen und Ausprobieren haben wir natürlich auch immer dabei. Und damit die Leser etwas für ihr Know-how tun können, gehört zu jeder „ElektorLive!“-Veranstaltung ein attraktives Seminarprogramm zu den unterschiedlichsten Elektronikthemen.

Das Ganze wollen wir unseren deutschen Lesern nicht länger vorenthalten: In diesem Jahr findet „ElektorLive!“ zum ersten Mal in Deutschland statt. Wir haben uns für diese Premiere das verkehrsgünstig gelegene Kongresszentrum Westfalenhallen in Dortmund ausgesucht, doch wenn alles klappt, sollen andere Städte folgen. In Deutschland wird sich der Schwerpunkt des Events ein wenig von einer Ausstellung zu einem kleinen Kongress verlagern. Seminar-Themen wie „Embedded Linux“, „Android“, „FPGAs“, „Trends bei AVR-Controllern“, „Entwickeln mit Eagle“ u.a. dürften die Profis ebenso wie die Hobbyisten ansprechen. Unter www.elektor-live.de findet man alle Infos und einen Link zur Anmeldeseite: Für jeden Besucher sind zwei Seminare im Eintrittspreis inbegriffen, besonders attraktive Konditionen gibt's für unsere Abonnenten sowie alle Studenten & Schüler. Wir sehen uns am 20. Oktober in Dortmund!

Jens Nickel



6 Impressum

Who is who bei Elektor

8 Aktuelles

News, Nachlesen und Updates

16 Nunchuk-USB-Adapter

Die Werte eines dreiachsigen Beschleunigungssensors, eines Joysticks und zweier Drucktaster lassen sich auf einfache Weise auslesen.



24 Modellbahn-Steuerung

Züge steuern mit Skripts, die bis zu 32 Anweisungen enthalten können: Vorwärts, Rückwärts, Station anfahren oder Warten.



32 USB-Isolator

Der Baustein ADuM3160 sorgt für eine galvanische Trennung der Datenleitungen, ein Flyback-Converter LT3575 ist für die Betriebsspannungsleitungen zuständig.



36 Zurück zu den Wurzeln (7)

In diesem Kursteil fangen LEDs an, zu blinken und zu blitzen. Und Signalgeneratoren sorgen für die akustische Untermalung.

42 Rechteck-Signalbox

Unser Low-Cost-Rechteckoszillator überstreicht den weiten Bereich 250 Hz... 60 MHz. Zwischen 250 Hz und 60 kHz kann das Rechtecksignal pulsbreitenmoduliert werden, so dass zahlreiche Objekte getestet werden können.

43 Labcenter

Löten mit Löffel
USB: Strom ohne Ende!
GPIO beim Elektor-Linux-Board
SD-Karten-Korrektur per Script
Elektor-projects.com 4U2
Steckiquitäten 2.0



```
root@gnublin:~# cd
root@gnublin:~# ./
Hello World!
root@gnublin:~#
```

INHALT

43. Jahrgang
September 2012
Nr. 501



16 Nunchuk-USB-Adapter

Neues Leben für alte Game-Controller

Die Spielkonsole Wii ist mit einer zweiten Steuerung nachrüstbar, der Nintendo den Namen „Nunchuk“ gab. Im Nunchuk sind ein dreiachsiger Beschleunigungssensor, ein analoger Joystick und zwei Drucktaster verbaut. Mit unserem Adapter auf Basis eines PIC18F-Controllers kann dieses leistungsfähige Eingabegerät in eigenen Projekten eingesetzt werden.



24 Modellbahn-Steuerung

Mit USB und Skript-Editor

Auf Böden und in Kellern fristet so manche kleine Modellbahn ihr Dasein, die einst die Herzen längst erwachsener Kinder höher schlagen ließ. Eine kleine, zeitgemäße Modellbahn-Steuerung holt das alte Spielzeug vom Abstellgleis zurück. Hochkomplexe Hightech-Züge benötigt es dafür ebenso wenig wie aufwendige Stellwerke.



32 USB-Isolator

Nie wieder Ärger mit Masseschleifen

Masseschleifen zwischen PCs und Geräten, die über USB miteinander kommunizieren, sind oft Ursache hoher Störpegel auf den Signalen. Der USB-Isolator ist der Retter in der Not, wenn die Systeme in störintensiver Umgebung arbeiten müssen. Er trennt die Systeme galvanisch voneinander und schlägt gleichzeitig eine Brücke, sowohl für die Daten- als auch die Betriebsspannungsleitungen.



48 Embedded Linux leicht gemacht (3)

Softwareentwicklung

Erst die richtige Software erweckt einen Mikrocontroller zum Leben. Zusätzlich zur klassischen Firmware muss man bei Embedded GNU/Linux auch die Betriebssystemkomponenten generieren. In diesem Artikel zeigen wir, wie das funktioniert. Und ein erstes kleines C-Programm schreiben wir auch.

47 Bauelemente-Tipp

Mit Hilfe des Chips AS3935 kann ein Mikrocontroller die Entfernung zu einem Gewitter bestimmen.

48 Embedded Linux leicht gemacht (3)

In diesem Artikel generieren wir den Bootloader und den Kernel. Ein erstes kleines Programm schreiben wir auch!

56 Arduino auf Kurs (1b)

Jetzt geht es dem Controller an den Kragen: Der ATmega328 wird auf „low level“ gehackt, um Code zu erzeugen, der Sound-Daten hörbar macht.

62 SDR mit AVR (5)

Diesmal verwenden wir verschiedene Decodierungsmethoden und Filter, um die Signale je eines deutschen, britischen und französischen Zeitzeichensenders in digitale Daten zu verwandeln.

68 Quadrowalker

Spezial-Bauteile sind unserem vierbeinigen Roboter fremd, acht Servo-Antriebe und ein Mikrocontroller sind die wesentlichen Komponenten.

72 2,4 GHz: Vielseitige Trägerfrequenz

In diesem Artikel wird ein kurzer Überblick über die grundlegenden 2,4-GHz-Technologien gegeben. Wir erklären dabei auch, was es mit der berühmten Frequenzspreizung auf sich hat, welche viele Vorteile bietet.

78 Retronik

Der „Pansanitor“ (1928)

82 Hexadoku

Sudoku für Elektroniker

86 Vorschau

Nächsten Monat in Elektor

Unser Team

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

Unser Netzwerk



Internationale Teams

 Großbritannien Wisse Hettinga +31 46 4389428 w.hettinga@elektor.com	 Spanien Eduardo Corral +34 911019395 e.corral@elektor.es	 Indien Sunil D. Malekar +91 9833168815 ts@elektor.in
 USA Hugo Vanhaecke +1 860-875-2199 h.vanhaecke@elektor.com	 Italien Maurizio del Corso +39 2 66504755 m.delcorso@inware.it	 Russland Nataliya Melnikova +7 (965) 395 33 36 Elektor.Russia@gmail.com
 Deutschland Ferdinand te Walvaart +49 241 88 909-0 f.tewalvaart@elektor.de	 Schweden Wisse Hettinga +31 46 4389428 w.hettinga@elektor.com	 Türkei Zeynep Köksal +90 532 277 48 26 zkoks@beti.com.tr
 Frankreich Denis Meyer +31 46 4389435 d.meyer@elektor.fr	 Brasilien João Martins +55 11 4195 0363 joao.martins@editorialbolina.com	 Südafrika Johan Dijk +27 78 2330 694 / +31 6 109 31 926 J.Dijk@elektor.com
 Niederlande Harry Baggen +31 46 4389429 h.baggen@elektor.nl	 Portugal João Martins +351 21413-1600 joao.martins@editorialbolina.com	 China Cees Baay +86 21 6445 2811 CeesBaay@gmail.com

IMPRESSUM

43. Jahrgang, Nr. 501 September 2012
Erscheinungsweise: 11 x jährlich (inkl. Doppelheft Juli/August)

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

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

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

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

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

Die Elektor Community

274920

Mitglieder in

83

Ländern...

Noch kein Mitglied?www.elektor.de/community



Elektor-Newsletter E-weekly jetzt gratis abonnieren!

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

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



Klicken Sie jetzt auf
www.elektor.de/newsletter!



Unsere Partner und Sponsoren

	AudioXpress www.audioamateur.com 23		Linx Technologies www.linxtechnologies.com 9
	Beta Layout www.pcb-pool.com 2		Medlab www.medlab.eu 11
	Eurocircuits www.elektorpcbservice.com 15		MikroElektronika www.mikroe.com 3
	Jackaltac www.jackaltac.com 77		Pico www.picotech.com/PS176 59
	LeitOn www.leiton.de 11		Reichelt www.reichelt.de 88

Sie möchten Partner werden?

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

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

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

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



© 2012 elektor international media b.v.
Druck: Senefelder Misset, Doetinchem (NL)
ISSN 0932-5468

Nicht nachmachen!

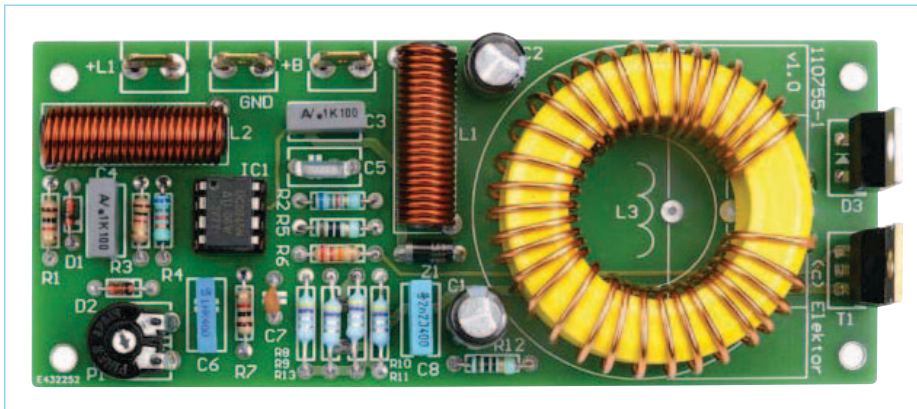
Das unterscheidet Elektor von vielen anderen Magazinen: Hier schreiben Elektronik-Fans für Elektronik-Fans. Ohne unsere internationale Leser-„Community“ könnte unsere Zeitschrift schlicht und einfach nicht funktionieren.

So lernt Hans von Jaime, wie man die Fuse-Bits eines AVR-Controllers setzt, und Charles erfreut sich an den Schaltungskniffen, mit denen Boris sein Power-LED-Netzteil schön stromsparend hinbekommen hat...

Schon immer machte Elektor auch den 1:1-Nachbau der Entwürfe sehr einfach – mit Stücklisten, Bestückungsplänen und einem Platinen-Service. Wenn man für sich (und seine Freunde) eine (oder ein paar) Platine(n) bestückt, bleibt das Ganze auch rechtlich wasserdicht. Mit Problemen behaftet kann es dagegen sein, wer die in Elektor veröffentlichten Schaltungen und Platinen nachmacht, um damit Geld zu verdienen. Es ist ein wenig klein gedruckt, aber

Cockpit angezeigt. Wenn man nun einfach einen Widerstand parallel schaltet, ist das natürlich nicht gerade stromsparend. Hier kommt die im Maiheft gezeigte Schaltung ins Spiel: Statt des Widerstands wird eine Stromquelle parallel geschaltet und der Strom ins Bordnetz zurückführt. Weil die Betriebsspannung der Stromquellenschaltung aber unter der Bordnetz-Spannung liegt, muss man dabei einen Step-up-Kon-

wie ein Sperrwandler oder Flusswandler konzipiert sein kann ...“. In der Patentschrift EP 2 173 140 B1 [3] ist nur ein Prinzipschaltbild, aber keine Schaltung mit realen Bauteilen zu finden. Wer die Idee kommerziell ausnutzen wollte, brauchte bisher also noch einiges an Elektronik-Knowhow. Denn er musste ein geeignetes Schaltregler-IC finden und passend beschalten, die Bauteile dimensionieren, das Ganze testen und schließlich eine passende Platine routen.



nicht unwichtig, was in unserer Zeitschrift auf Seite 7 unten steht: „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.“

Mitnichten ist das nur graue Theorie, wie eine Schaltung beweist, die wir im Mai 2012 veröffentlicht haben. Die „Verlustlose Last“ [1] macht den Ersatz von Kfz-Glühlampen durch LEDs einfacher (Foto). Dabei muss man der Fahrzeug-Elektronik nämlich vortäuschen, dass hier ein Birnchen weiter munter Strom verbraucht, sonst wird in vielen Autos ein Ausfall im

Die Idee ist gut – jedoch nicht neu. Das teilte uns die Firma Carsig GmbH [2] mit, deren Geschäftsführer Thomas Kliem wie viele andere Elektronikprofis zu unseren Lesern gehört. Das Unternehmen aus dem schwäbischen Rottweil, spezialisiert auf Fahrzeug-elektronik (Foto), hatte schon im Herbst 2009 eine Schaltung mit Glühlampensimulation zum Patent angemeldet, „...zur Leistungsrückführung in das Bordnetz bei Last- und Leistungssimulation in Verbindung mit modernen Leuchtmitteln (z.B. LED-Technik) bei Lichtfunktionen wie z.B. Blinker, Bremslicht, Rücklicht...“. Ein weiteres Zitat aus der Patentschrift zeigt, dass es sich tatsächlich um die gleiche Idee handelt: „Das Herzstück der Regelschaltung ... ist der durch den Schalter S3 und die Spule L1 gebildete Schaltregler zum Takten, der beispielsweise

Diese Situation änderte sich freilich im Mai 2012 ... Elektor-Leser Carlo Cianferotti war auf dieselbe Lösung des Problems gekommen, nichtsahnend, dass ein deutsches Unternehmen auf diesem Feld schon einiges an Entwicklung betrieben hatte, wie er uns zwischenzeitlich (ein wenig traurig) schrieb. Und unser Labor und die Redaktion hatten aus seiner Einsendung einen praxisbezogenen, elektor-typischen Nachbau-Artikel gemacht. Sehr zum Leidwesen von Thomas Kliem und seinen Mitarbeitern, denn nun könnte theoretisch jeder (ob Elektroniker oder nicht) die schöne Idee benutzen, um eigene Geräte auf den Markt zu bringen ... Wir können hiervor nur nachdrücklich warnen! Lediglich die private Nutzung der bei uns veröffentlichten Schaltungen ist frei. Für Forderungen und Ansprüche von Drit-

ten, die durch eine unerlaubte kommerzielle Nutzung entstehen, kann der Elektor Verlag keinerlei Verantwortung übernehmen. Es wäre für unsere Autoren und unseren Verlag viel zu aufwendig, wenn wir für jede veröffentlichte Schaltungs-idee noch Recherche betreiben müssten, ob ein Patent- oder Gebrauchsmusterschutz besteht.

Anders kann das Konzept unserer Zeitschrift (und unserer Websites) nicht funk-

tionieren. Und so werden sich hoffentlich noch viele Generationen von Elektronikern an den Ideen, Schaltungen und Tricks von anderen Elektronikern erfreuen können!

[1] www.elektor.de/110755

[2] www.carsig.de

[3] <https://data.epo.org/publication-server/getpdf.jsp?pn=2173140&ki=B1&cc=EP>

Updates und Ergänzungen

AndroPod

Elektor 2/2012, S. 16 (110405) – Elektor 3/2012, S. 20 (120097)

Auf der Website www.elektor.de/110405 ist ein neues Software-Paket downloadbar, es behebt Treiberprobleme mit Windows XP. Auch eine neue Vinculum-Firmware (.rom) ist dort zu finden, die ein Kompatibilitätsproblem mit Samsung Smartphones löst. Eine (englischsprachige) Anleitung zum Flashen der Firmware findet man ebenfalls im Downloadordner.

Ein weiteres Update betrifft die Android-App „ElektorBusBrowserForAndropod“. Ein Bug (Datei „index.htm“ nicht gefunden) wurde behoben, außerdem ist die App jetzt für (fast) alle Tablets (auch ohne Telefoniefunktionen) tauglich. Die neue Version der App ist auch auf Google Play (ehemals Google Android Market) zu finden.

Embedded Linux leicht gemacht (2)

Elektor 6/2012, S. 20 (120146)

Im Schaltplan ist leider ein Fehler: Der Typ von IC7 ist CP2102, nicht CD21021.

Achtung: Wenn man von SD-Karte gebootet hat, muss man das Board erst mit dem Kommando „halt“ herunterfahren, bevor man die SD-Karte entfernt. Ausführlich ist das im dritten Teil der Serie (in diesem Heft) beschrieben.

Ein TAPIR kann's erschnüffeln

Elektor 7-8/2012, S. 112 (120354)

Im Schaltbild in der Sommer-Ausgabe gibt es zwei Widerstände, die mit R2 bezeichnet sind, was natürlich falsch ist. Der Widerstand „R2“ ganz unten neben R4 müsste eigentlich mit „R1“ bezeichnet werden (siehe auch die Stückliste).

Im Schaltbild sind drei Elkos mit 10 µF eingetragen. Im Bausatz werden aber keramische Vielschichtkondensatoren mit 10 µF/10 V geliefert, was den Vorteil hat, dass man die Polung nicht beachten muss.



Im Artikel steht, dass die induktive Antenne aus einer Spule aus Installationsdraht besteht. Hier handelt es sich um einen Übersetzungsfehler. Tatsächlich dient der Installationsdraht nur zum Stützen der Spule (Minidrossel mit 3,3 mH im Bausatz).

Algebra Quiz

1. Expand the following:

$(2x + 3y)^2$ (5 points)

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

+2

Very funny

If only RF could
be so easy.

Linx
TECHNOLOGIES

Wireless made simple®

RF Modules
Remote Controls
Antennas
RF Connectors
Custom Designs

www.linxtechnologies.com



Elektor-Wheelie-Treffen

Von Dr. Thomas Scherer

Wenn man im Internet nach Videos, Blogs und Webseiten von Leuten sucht, die sich einen selbstbalancierenden Roller gebaut haben (oder sich damit beschäftigen), so findet man immerhin einige Dutzend. Darüber hinaus wurden überall auf der Welt weit über 100 Elektor-Wheelie-Bausätze zusammengesetzt [1]. Eine florierende Selbstbauszene ist da also entstanden, die sich unter anderem im Elektor-Forum austauscht. Dort entstand auch die Idee, sich einmal „in echt“ zu treffen, und so fand im Mai die erste

„Elektor Wheelie Convention“ der Welt statt!

Elektor-Autor Günter Gerold (vielen Lesern bekannt als Entwickler des „Geartooth-Sensors“) stellte seine Werkstatt im schwäbischen Ellwangen für das Treffen zur Verfügung. Günter ist Meister für Radio- und Fernseh-Technik und seine Werkstatt nicht nur groß, sondern auch optimal ausgestattet. Und so kam es, dass zehn interessierte Menschen sich gegenseitig ihre zweirädrigen Fahrzeuge zeigten, dass intensiv gefachsimpelt wurde, dass Elektronik abgeglichen und sogar repariert werden konnte. Der Gastgeber war unermüdlich im Einsatz und demonstrierte nebenbei auch, wie man mit einfachen Mitteln perfekt geätzte und mit Lötstopplack versehene Platinen herstellen kann.

Das Allerwichtigste waren (neben dem persönlichen Kennenlernen) aber sicher die zahlreichen Fahrttests und Vergleiche, die bei so vielen unterschiedlichen Zweirädern unternommen werden konnten. Günter konnte sogar zwei originale Segway vom Typ I2 organisieren, um einen direkten Vergleich mit industriell gefertigten Geräten vornehmen zu können. Die bei diesem Treffen entstandene Bildergalerie vermittelt einen Eindruck davon, wieviel Spaß und Vergnügen dieses außergewöhnliche Treffen von Elektor-Lesern gebracht hat.

(120497)

Weblinks

- [1] www.elektor.de/wheelie
- [2] www.gerold-online.de/cms/index.php



Am Abend des ersten Tags: Diverse Elektrozweiräder parken friedlich in Günters Werkstatt.



Hier steht Wolfgang (Mitte) neben seinem Eigenbau auf der Basis von Rollstuhlmotoren. Sein selbstbalancierendes Zweirad fährt zwar nicht so schnell, aber gut.



Keine reine Männersache: Hier fährt Christine auf dem „Rutscherle“ (dem Scooter von Daniel). Marlene, die jüngste Tochter von Günter, steuert einen Segway i2.



Selbstverständlich wurde auch gebastelt und getüftelt. Hier lötet Günter ein paar SMDs unterm Stereo-Mikroskop.



Reifenwechsel: Thomas (links) und Daniel (rechts) zwingen mit vereinten Kräften einen Reifen auf einen BLDC-Hub-Motor.



Vater und Sohn:
Slalomrennen von Johannes (links) gegen Günter (rechts) auf dem Hindernis-Parcour im Hof vor der Werkstatt.

EINKAUFFSSPASS AUCH FÜR MÄNNER.

GROSSES PRODUKTSPEKTRUM ONLINE KALKULIERBAR.



LEITON
RECHNEN SIE MIT BESTEM SERVICE

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

www.leiton.de

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

medlab

medizinische Diagnosegeräte GmbH

Medlab ist ein deutscher Hersteller von Pulsoximetern, Kapnographen und anderen elektromedizinischen Geräten. Wir suchen zum nächstmöglichen Zeitpunkt:

Entwicklungsingenieur(-in), Elektrotechnik

Art der Stelle:
Aufgaben:

Vollzeitstelle
Hardwareentwicklung, Analog- und Digitaltechnik

Ausbildung:
Ausführliche Informationen:
Bewerbungen bitte an:

Studium der Elektrotechnik
www.medlab.eu/stelle1.html
personal@medlab.eu

Medlab GmbH
Helmholtzstrasse 1
76297 Stutensee (Karlsruhe)
Tel.: 07244 741100
personal@medlab.eu
www.medlab.eu



DesignSpark ChipKIT™



Seit Anfang September 2011 beteiligten sich Elektroniker aus der ganzen Welt mit eigenen Entwicklungen auf der Basis des Entwicklungs-Boards chipKIT Max32 von Microchip am Wettbewerb. Während sich diese umweltbewussten Entwickler in den DesignSpark-Foren untereinander austauschten, machten ihre innovativen Projekte Fortschritte und waren schließlich reif für die Beurteilung durch die Jury.

Die Juroren sahen alle Beiträge durch und bewerteten sie nach ihrer technischen Qualität, der Originalität sowie der Qualität des Platinen-Designs der Erweiterungskarten, welche mit den DesignSpark-Platinentools erstellt wurden. Die Arbeit der Jury ist mittlerweile abgeschlossen. An dieser Stelle ein herzliches Dankeschön an alle Teilnehmer dieses spannenden Wettbewerbs und Glückwünsche an die Gewinner!

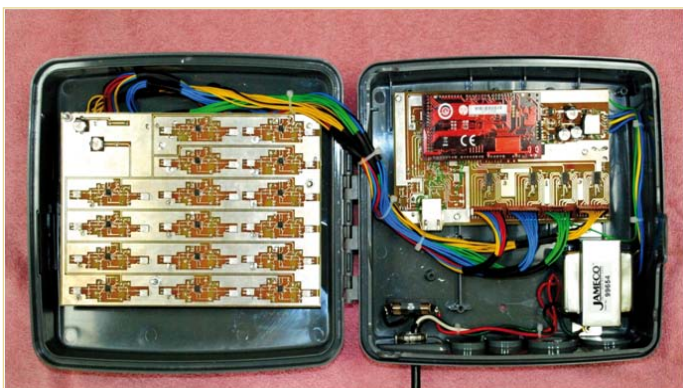
Erster Preis

Dean Boman

USA – d.boman@cox.net

Energy Monitoring System

Das Energy Monitoring System (EMS) stellt Daten des elektrischen Energieverbrauchs für die Bewohner eines Hauses in Echtzeit zur Verfügung. Das EMS besteht aus dem Entwicklungs-Board chipKIT Max32 mit zwei Erweiterungskarten. Auf einem Webserver stehen die Verbrauchsdaten zum Abruf bereit. Die Lösung bietet eine Schnittstelle für ein Heimautomatisierungs-System zur Langzeitüberwachung und Datenspeicherung. Das System nutzt eine in C geschriebene Software, die mit Hilfe der Entwicklungsumgebung MPLAB von Microchip und deren TCP/IP-Stack erstellt wurde.



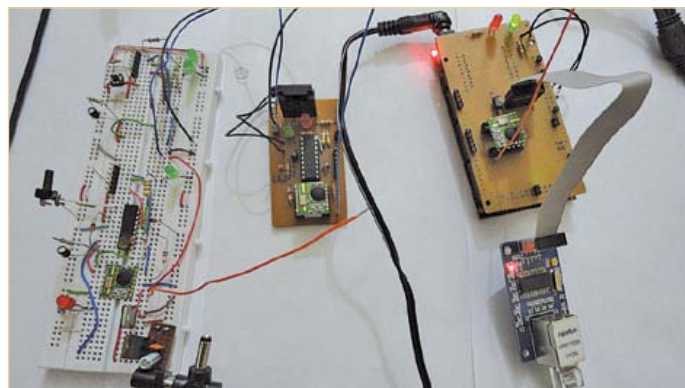
Zweiter Preis

Raul Alvarez Torrico

Bolivien – raul-at@hotmail.com

Home Energy Gateway

Das Home Energy Gateway erlaubt die Erfassung des Energieverbrauchs und die Fernsteuerung von elektrischen Geräten und Anlagen im Haus. Ein auf der Basis von chipKIT Max32 entwickelter Webserver kommuniziert dabei mit zwei Arten intelligenter Geräte im Haus: Einem Smart-Meter zur Erfassung des mittleren aktuellen Verbrauchs und etlichen intelligenten Steckdosen in einem häuslichen Funknetzwerk. Diese Steckdosen können dann per Webinterface ferngeschaltet werden..



Challenge: Die Gewinner

Dritter Preis

Graig Pearen

Canada – graig@pearen.ca

SunSeeker (PV Array Tracker)

Hier geht es darum, Solarzellen-Panels so auszurichten, dass sie immer optimal zur Sonne stehen und so die beste Energieausbeute möglich ist. Auf der Basis von Microchips chipKIT Max32 kann dieses System Solarmodule je nach den Wetterbedingungen und dem Sonnenstand ausrichten und überwachen. Dabei werden diese Bedingungen, die Solarmodul- und die Lufttemperatur gemessen und aus den erhobenen Daten Statistiken erstellt. Darüber hinaus kommuniziert das System mit einem lokalen Server zur Optimierung der Algorithmen. Eine Diagnostik-Software überwacht die Motoren und zeigt Bewegungen und die aktuelle Position an.



Die Handheld PIC18 IDE ist ein autonomes System für das Erstellen, Editieren und die Verwaltung von Source-Dateien für den Controller PIC18 von Microchip. Das binäre Resultat dieses Prozesses kann in einen PIC18 übertragen und auf Source-Ebene debugged werden. Die Hardware ist einfach. Sie besteht aus einem User-Interface (LCD und Tastatur), Speicher für Daten und einem Programmier-Interface. Diese praktische IDE enthält außerdem einen BF-Interpreter für das Schreiben und Ausführen von Scripts. Man kann auch die Programmiersprache BASIC benutzen. Das Ganze ist außerdem auch noch solargespeist und mit einem Lithium-Ionen-Akku als Backup gestützt.

Besondere Erwähnung

John Schuch

United States – hackersbench@gmail.com

Wireless Mesh Network Time Server

Der solargespeiste, drahtlose Mesh Network Time Server holt sich die Angaben für Zeit und Datum von einem GPS-Empfänger und verbreitet sie innerhalb des Mesh-Netzwerks per XBee-Modul. Als Backup dient ein NiCd-Akku, der von einem kleinen Solar-Panel geladen wird. Im normalen Modus findet der Broadcast von Zeit und Datum einmal per Minute statt. Wenn ein anderer Netzwerk-Teilnehmer eine Anfrage schickt, schaltet der Server in den Stream-Mode, indem die Frequenz auf 1/s gesteigert wird. Außerdem wird im Test-Mode die Akkuspannung gemessen und übermittelt. Die recht einfache Software erlaubt die Programmierung in einer Hochsprache (MPIDE). Der Energieverbrauch wird minimiert.

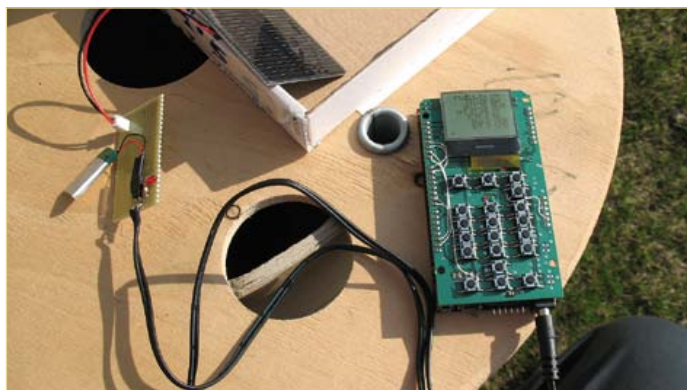


Besondere Erwähnung

Jaromir Sukuba

Slovakia – j.sukuba@gmail.com

Handheld PIC18 IDE



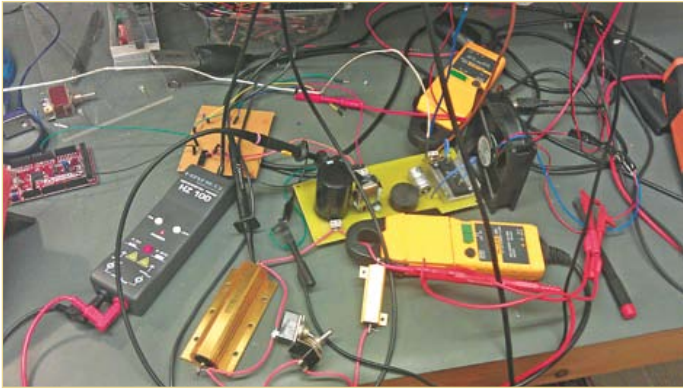
Besondere Erwähnung

Ian Johnson, Sajjad Lalji, David Weight

UK – ian.johnson@wattcircuit.com

MPPT Boost Converter

Ein Maximum Power Point Tracker (MPPT) sorgt für die Gewinnung



des Maximums der möglichen Energie aus z.B. thermoelektrischen und fotovoltaischen Energiequellen oder auch elektromagnetischen Übertragungssystemen. Hierfür wurde eine Impedanz-Anpassungs-Methode auf der Basis einer digital gesteuerten geschlossenen Schleife realisiert. Auf diese Weise ist die Methode auch für Anwendungen tauglich, die einen konstanten Innenwiderstand aufweisen. Ein modifiziertes Verfahren eignet sich für Anwendungen mit variablem Innenwiderstand.

Besondere Erwähnung

Manuel Iglesias Abbatemarco

Venezuela – mmanuel@ieee.org

Eco-Friendly Home Automation Controller

Basis des Systems ist ein Arduino-kompatibles chipKIT-Board. Ein passendes Solarpanel versorgt zwei Lithium-Ionen-Zellen. Auch hier wurde ein MPPT-fähiger Lader integriert, der die nichtlineare Ausgangsleistung der Solarzellen optimal ausnutzt. Vorhanden sind weiter ein Quad-Band-GSM/GPRS-Modem, ein Sockel für ein XBee-Modul, ein SD-Karten-Slot und ein RTCC. Die Software wurde mit



Hilfe von MPIDE erstellt. Die SD-Karte ist zur Speicherung von Sensordaten etc. gedacht.

Besondere Erwähnung

Curtis Brooks

USA – brooksware2000@gmail.com

Internet-Enabled Multizone Thermostat

Ein Internet-fähiger Thermostat erleichtert die Steuerung von Gebäude-Temperaturen. Das System besteht aus drei Teilen: einem XBee-Shield, einem Funk-Temperatur-Modul und einem per I2C angebundenen Interface-Board. Insgesamt sind zwei Temperatur-Module und zwei Interface-Boards möglich. Ein Router sorgt für die Internet-Anbindung. Der erste Teil der Lösung basiert auf



einem chipKIT Max32, einem Max32-Ethernet-Shield und einem XBee-Shield (Platine). Der zweite Teil besteht aus einem Funk-Thermostat für die Steuerung von Heizung und Klimaanlage und erhält Daten von den Raumknoten. Die Raumknoten steuern dann jeden Raum separat.

(120420)

Alle Informationen zum DesignSpark/ChipKITTM Design Challenge findet man unter:

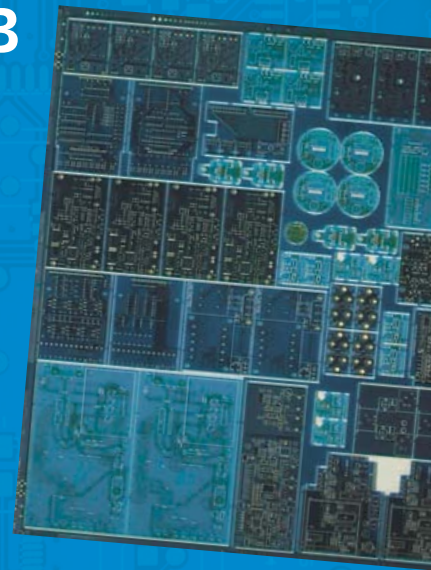
www.circuitcellar.com/contests/chipkit2012

Die europäische Referenz für PCB Prototypen und Kleinserien

Sparen Sie Entwicklungszeit und -kosten mit unseren einfach zu nutzenden Leiterplatten-Pooling-Services.

Wir bieten Ihnen:

- Top Qualität zu niedrigen Pooling Preisen
- Schnelle Lieferung nach Ihren Bedürfnissen
- Umfangreiche Technologie-Unterstützung
- Keine Einmal- oder Werkzeugkosten
- Keine Mindestbestellwerte - ab der 1 Leiterplatte
- Online-Bestellung ohne Vorkasse
- Schablonen Service



PCB proto – spezieller Prototypen-Service für Entwickler, preiswert und schnell

- 1 oder 2 LP in 2, 3, 5 oder 7 Arbeitstagen
- DRC-geprüft, professionelle Ausführung inkl. 2x Lötstopplack und 1x Bestückungsdruck, 150µm Technologie
- 1 x 100 x 80mm in 7AT - 2 Lagen 46.26 € - 4 Lagen 93.94 €
- 2 x 100 x 80mm in 7AT - 2 Lagen 36.28 € je LP - 4 Lagen 73.52 € je LP

Preise inkl. 19% MwSt und ohne Transportkosten

STANDARD pool – die größte Auswahl an Eurocircuits Pooling Optionen

- 1-8 Lagen 150µm Technologie-Leiterplatten
- ab 2 AT

TECH pool – 100µm-Technologie mit allen Pooling-Vorteilen

- 2-8 Lagen 100µm Technologie-Leiterplatten
- ab 4 AT

IMS pool – Aluminiumkern-Leiterplatten für hohe Wärmeableitung (z.B. LED-Anwendung)

- Leiterplatten mit einlagig isoliertem Metallsubstrat
- 1.5mm Alukern mit 75µm thermisch leitfähigem Dielektrikum und 35µm Kupferfolie
- ab 3 AT

On demand – Alle Optionen im Nicht-Pooling für Spezialanwendungen

- 1-16 Lagen bis 90µm-Technologie
- RF- und Hoch-Tg-Materialien
- ab 2 AT

Nunchuk-USB-Adapter

Neues Leben für alte Game-Controller

Von Anthony Le Cren (F)

Die Spielkonsole Wii ist mit einer zweiten Steuerung nachrüstbar, der Nintendo den Namen „Nunchuk“ gab. Im Nunchuk sind ein dreiachsiger Beschleunigungssensor, ein analoger Joystick und zwei Drucktaster verbaut. Mit unserem Adapter auf Basis eines PIC18F-Controllers kann dieses leistungsfähige Eingabegerät in eigenen Projekten eingesetzt werden.



Seiner Video-Spielkonsole Wii gab Nintendo eine über Bluetooth drahtlos angebundene Bedieneinheit mit auf den Weg, die „Wii-mote“ genannt wird. Über ein Kabel kann dort „Nunchuk“ angeschlossen werden, eine zweite Bedieneinheit, die das Spielen mit beiden Händen möglich macht.

Der hier vorgestellte Nunchuk-USB-Adapter dient dazu, die von Nunchuk generierten Daten über I²C zu lesen, so dass die Informationen auch auf anderen Wegen genutzt werden können. An dieser Stelle sollen nur einige Beispiele genannt werden:

- Grafische Darstellung und Interpretation auf einem Bildschirm,
- Weitergabe an einen PC über die Tastatur- oder Maus-Schnittstelle,
- Steuern allgemeiner Schaltfunktionen,
- Datenübertragung über RS232 an einen zweiten Mikrocontroller.

Nunchuk stellt die Daten des Beschleunigungssensors für die Raumachsen X, Y und Z bereit, außerdem liefert dieser Spielecontroller Informationen über die Stellungen des analogen Joysticks und die Schaltzustände der Drucktaster C und Z.

Hardware

In **Bild 1** ist die Schaltung unseres Nunchuk-USB-Adapters dargestellt, zentrales Bauelement ist ein Mikrocontroller PIC18F2550. Das Taktsignal wird mit einem Quarz erzeugt, der auf 20 MHz schwingt. Diese Frequenz ist nicht die Frequenz des internen Taktsignals, denn für den Betrieb der USB-Schnittstelle muss die interne Taktfrequenz ein Vielfaches von 12 MHz betragen. Deshalb ist der Mikrocontroller wie folgt konfiguriert: Zuerst wird die Frequenz 20 MHz durch den Faktor 5 auf 4 MHz herabgeteilt. Das 4-MHz-Signal steuert eine interne PLL, die PLL-Ausgangsfrequenz beträgt 96 MHz. Diese Frequenz wird durch 2 geteilt, so dass ein Taktsignal mit der Frequenz

Eigenschaften

- Controller-Anschluss ohne speziellen Steckverbinder
- I²C-Protokoll
- Lesen der Daten vom Beschleunigungssensor (10 bit), vom Joystick (8 bit) und den Drucktastern (low-aktiv)
- USB-Schnittstelle zum Anschließen an den PC
- PIC-Firmware mit Bootloader und Programm PDFSUSB
- programmierbar
- Programmieren von Applikationen in Flowcode
- D-Sub-Buchsenleisten (weiblich) kompatibel zu E-blocks-Modulen
- Serielle Schnittstelle für zukünftige Erweiterungen

48 MHz entsteht. Wenn der Mikrocontroller zeitkritische Programme ausführen soll, ist die Art der Taktsignalerzeugung nicht beliebig. Eine Übersicht über die möglichen Konfigurationen enthält das Datenblatt des PIC18F2550 von Microchip [4]. Dort wird der interne Taktoszillator im Kapitel *Oscillator Types* beschrieben.

Der PC liefert über das USB-Kabel die Betriebsspannung 5 V. Für den Mikrocontroller wird diese Spannung von einem LP2950ACZ-3.3 (IC2) auf 3,3 V herabgesetzt. Die Pullup-Widerstände des I²C-Busses sind bereits im Mikrocontroller integriert, die Leitungen müssen nicht extern mit Pullup-Widerständen beschaltet werden. Damit die Adapterplatine mit dem PIC18F2550 auch als Entwicklungsboard genutzt werden kann, sind alle Mikrocontroller-Anschlüsse über zwei Steckleisten zugänglich. Während die Sub-D-Steckleiste K4 die Verbindungen zu den analogen Eingängen herstellt, liegen an der Sub-D-Steckleiste K5 die sechs hohen Bits des Ports B. Die Bezeichnungen der Anschlüsse sind so gewählt, dass hier ein LC-Display mit 2 · 16 alphanumerischen Zeichen unkompliziert angeschlossen werden kann. Das LC-Display ist optional, es muss nur vorhanden sein, wenn es der Einsatzzweck erfordert. Die beiden unteren Bits von Port B sind für den I²C-Bus reserviert, die I²C-Bus-Leitungen liegen an Steckverbinder K1. Übrig bleibt noch Steckverbinder K3, er ist für eine zukünftige serielle Schnittstelle (UART) bestimmt.

Periphere Elemente wie Sensoren, Aktoren oder LEDs können beliebig von der Anwendung gesteuert werden. Die LEDs leuchten auf, wenn die Mikrocontroller-Ausgänge auf High liegen. Die Drucktaster S1 und S2 sind bereits mit Funktionen belegt: S1 aktiviert den Bootlader des Mikrocontrollers und S2 führt einen System-Reset herbei.

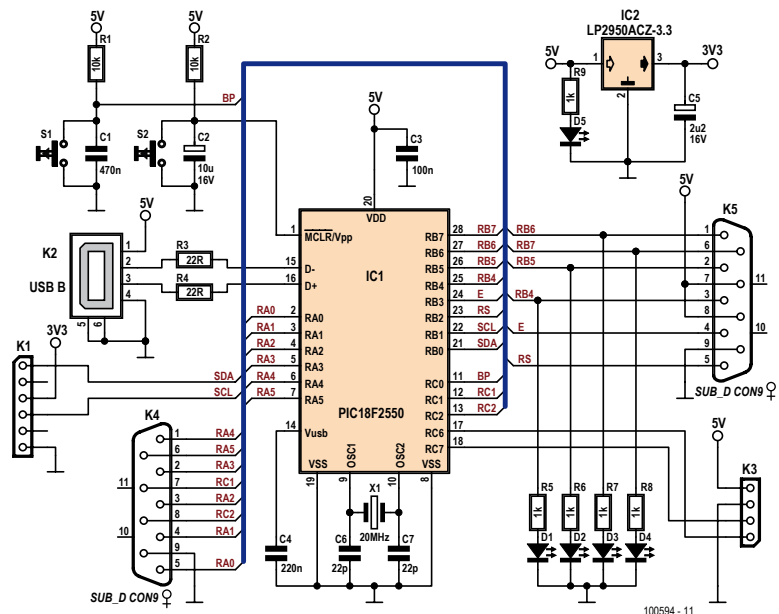


Bild 1. Außer einem Mikrocontroller PIC18F2550 und diversen Steckverbindern sind nur wenige Bauteile vorhanden.

Das I²C-Frame auf den I²C-Leitungen SCL und SDA wird mit einem Oszilloskop oder Bus-Analyser sichtbar (**Bild 2**). Die Synchronisation kann über die Leitung RC2 (Pin 8 von K4) erfolgen. Das Signal SYNCHRO geht auf logisch 1, bevor das Lesen der Nunchuk-Daten startet.

Der Beschleunigungssensor des Nunchuk eröffnet viele Möglichkeiten für eigene Entwicklungen. Auf der Elektor-Projektseite [1] sind Beispiele zusammengefasst, die unverändert oder modifiziert übernommen werden können. Die Datei *nunchuk.hex* steht auf der Elektor-Projektseite zum Download bereit, hier ist ein Bootloader integriert. Nach Laden dieser Datei in den Mikrocontroller bekommt man mit Hilfe des PC-Programms *PDFSUSB* von Microchip [3] Zugriff

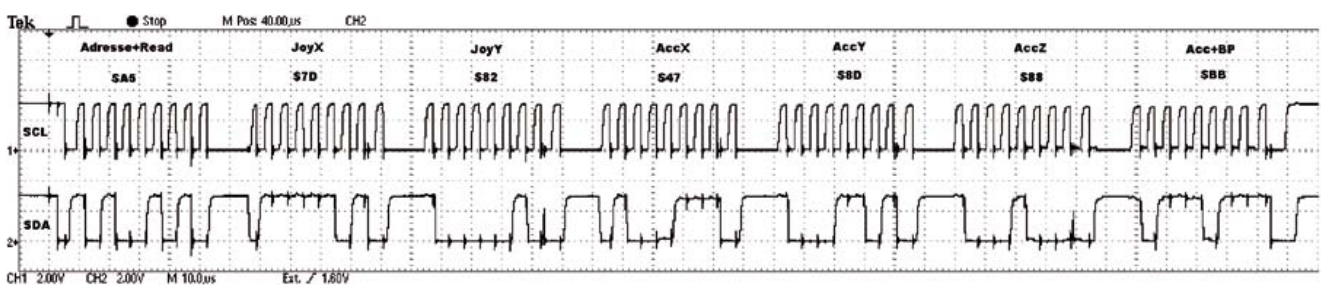


Bild 2. Ein vollständiger Frame besteht aus sieben Bytes.

Tabelle 1.

Datenflussrichtung	Adresse								R/W	hex
	D7	D6	D5	D4	D3	D2	D1	D0		
Schreiben: Datenübertragung vom Adapter zum Nunchuk-Controller	1	0	1	0	0	1	0	0		\$A4
Lesen: Datenübertragung vom Nunchuk-Controller zum Adapter	1	0	1	0	0	1	0	1		\$A5

Tabelle 2. Initialisieren des Nunchuk-Controllers

Starte Bus I ² C	Datenübertragung auf dem I ² C-Bus zum Nunchuk-Controller (Bild 3)
Sende 0xA4	
Sende 0xF0	
Sende 0x55	
Stopp Bus I ² C	
Starte Bus I ² C	
Sende 0xA4	
Sende 0xFB	
Sende 0x00	
Stopp Bus I ² C	

Tabelle 3. Initialisieren des RAM-Pointers im Nunchuk-Controller

Starte Bus I ² C	Datenübertragung auf dem I ² C-Bus zum Nunchuk-Controller (Bild 4)
Sende 0xA4	
Sende 0x00	
Stopp Bus I ² C	

Tabelle 4. Lesen der Daten aus dem Nunchuk-Controller

Starte Bus I ² C	Die Daten werden ohne das erste Byte 0xA5 (Lesekommando) über den I ² C-Bus zur Adapterkarte übertragen (Bild 4).
Sende 0xA5	
Empfange Joystick X in NUN_BUF[0]	
Empfange Joystick Y in NUN_BUF[1]	
Empfange Beschleunigungssensor X (MSB) in NUN_BUF[2]	
Empfange Beschleunigungssensor Y (MSB) in NUN_BUF[3]	
Empfange Beschleunigungssensor Z (MSB) in NUN_BUF[4]	
Empfange Beschleunigungssensor (LSB) und Drucktaster-Status C und Z in NUN_BUF[5]	
Stopp Bus I ² C	

auf das interne 32-KB-Flash-ROM, so dass Anwendungs-Programme einfach in den Controller geladen und modifiziert werden können.

I²C-Protokoll

Auf dem I²C-Bus sind die Funktionen wie folgt verteilt: Der Mikrocontroller PIC18F2550 ist stets Master, der Nunchuk-Controller ist Slave, die Nunchuk-Adresse lautet \$52 (binär 1010010). Die 7-bit-Adresse \$52 wurde vom Hersteller festgelegt. Damit auch andere periphere Geräte, beispielsweise der klassische Wii-Controller, mit dem Wiimote verbunden werden können, muss jedes Wii-Gerät eine individuelle Adresse haben. Im I²C-Protokoll ist festgelegt, dass das erste vom Master gesendete Byte eine Adresse darstellt. Das niederwertigste Bit D0 (R/W) gibt an, ob der Master vom Slave liest oder zum Slave schreibt. Daraus werden zusammen mit der 7-bit-Adresse \$52 das 8-bit-Schreibkommando und das 8-bit-Lesekommando, wie sie in **Tabelle 1** angegeben sind. Bevor die Nunchuk-Daten aus den Registern gelesen werden können, muss der Controller initialisiert werden. Die Initialisierung wird mit den Byte-Sequenzen aus **Tabelle 2** vorgenommen. Eine Webseite von wiibrew.org [2] beschreibt ausführlich die internen Register und ihre Funktionen. Abschließend ist der Pointer des RAMs, in dem die Nunchuk-Daten stehen, mit der in **Tabelle 3** stehenden Byte-Sequenz auf Null zu setzen (siehe **Bild 4**).

Jetzt kann das aus sechs Daten-Bytes bestehende Frame gelesen werden:

- Start
- 1 Byte Adresse mit dem Lesekommando (\$A5), gesendet vom PIC18F2550 (Master),
- 6 Byte Daten, gesendet vom Nunchuk-Controller (Slave),
- Stop.

Die sechs empfangenen Daten-Bytes werden in Tabellenform im RAM des PIC18F2550 abgelegt (**NUN_BUF[0]** bis **NUN_BUF[5]**, siehe **Tabelle 4** und **5**).

Da die Daten des Beschleunigungssensors aus 10 Bit bestehen, wer-

Tabelle 5. I²C-Protokoll für das Lesen des Nunchuk-Controllers

Byte 1 0xA5	Byte 2 NUN_BUF[0]	Byte 3 NUN_BUF[1]	Byte 4 NUN_BUF[2]	Byte 5 NUN_BUF[3]	Byte 6 NUN_BUF[4]	Byte 7 NUN_BUF[5]
Nunchuk-Adresse und Lesebefehl	Joystick X (0 ... 255) 8 bits	Joystick Y (0 ... 255) 8 bits	Beschleunigungs-sensor X (MSB, Bit 9 ... 2)	Beschleunigungs-sensor Y (MSB, Bit 9 ... 2)	Beschleunigungs-sensor Z (MSB, Bit 9 ... 2)	Beschleunigungs-sensor X,Y,Z (LSB) sowie Drucktaster C und Z

Tabelle 6.

D7	D6	D5	D4	D3	D2	D1	D0
Sensor Z	Sensor Z	Sensor Y	Sensor Y	Sensor X	Sensor X	Taster C	Taster Z
Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0	low-aktiv	low-aktiv

den die niedrigen Bits im letzten Byte untergebracht. Diese Bits müssen an den höherwertigen Teil angefügt werden, so dass das Ergebnis ein Wert zwischen dezimal 0 und 1023 ist (siehe **Tabelle 6**).

Das folgende Programmsegment berechnet die Werte der drei Achsen:

```

joy_x = NUN_BUF[0]
joy_y = NUN_BUF[1]
acce1_x = (NUN_BUF[2]*4)+((NUN_BUF[5]/4) AND 0x03)
acce1_y = (NUN_BUF[3]*4)+((NUN_BUF[5]/16) AND 0x03)
acce1_z = (NUN_BUF[4]*4)+((NUN_BUF[5]/64) AND 0x03)
bouton_c = (NUN_BUF[5] / 2 ) AND 0x01
bouton_z = NUN_BUF[5] AND 0x01

```

Aufbau

Da keine SMDs zu den Bauelementen gehören, gestaltet sich der Aufbau von Hand vergleichsweise einfach. Beginnen Sie mit der Montage der flachen Bauteile, gefolgt von der IC-Fassung, den LEDs und den Kondensatoren. Erst zum Schluss sind der Quarz und die Steckverbinder an der Reihe. Um den Nunchuk an die Platine anschließen zu können, müssen zwei Nuten in die Platine gefräst oder gesägt werden. Nur wenn diese Nuten vorhanden sind, lässt sich der originale Nunchuk-Stecker auf die Leiterbahnkontakte aufschieben. Denn anders als meistens gebräuchlich ist ein montierter Steckverbinder auf der Platine nicht vorhanden, der Nunchuk-Stecker hat unmittelbaren Kontakt mit Leiterbahnen auf beiden Platinnenseiten. Beim Aufstecken ist unbedingt auf die korrekte Steckerposition zu achten: Die muldenförmige Steckerseite muss zur Platinenlötseite weisen, die glatte Seite muss auf der Bestückungsseite

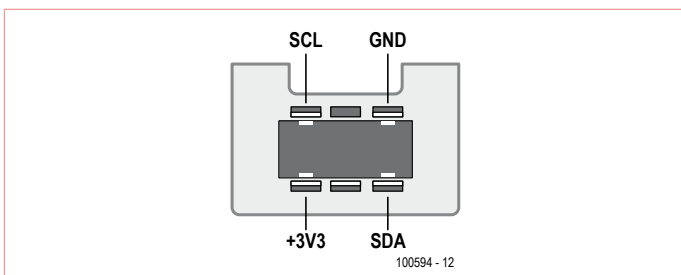


Bild 5. Der Nunchuk-Stecker: Die gemuldeten Seite muss in die gleiche Richtung wie die Platinen-Lötseite zeigen.

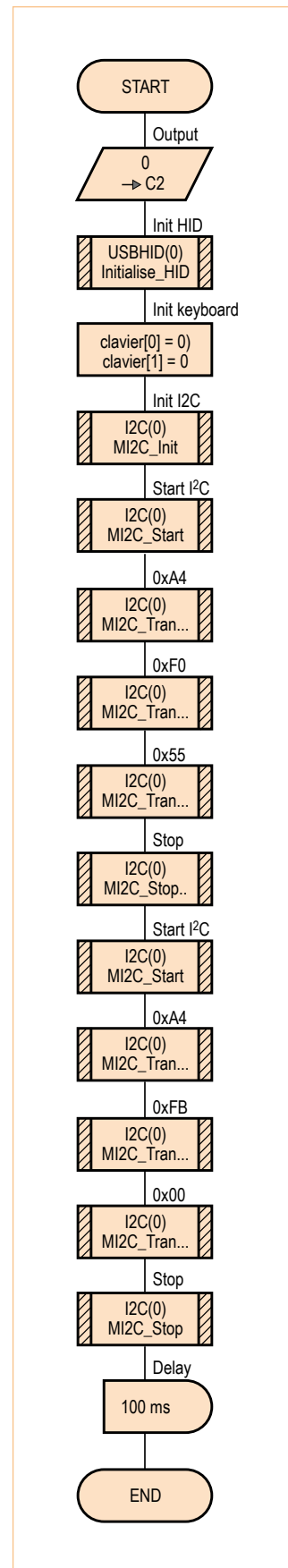


Bild 3. Initialisierung des Nunchuk-Controllers.

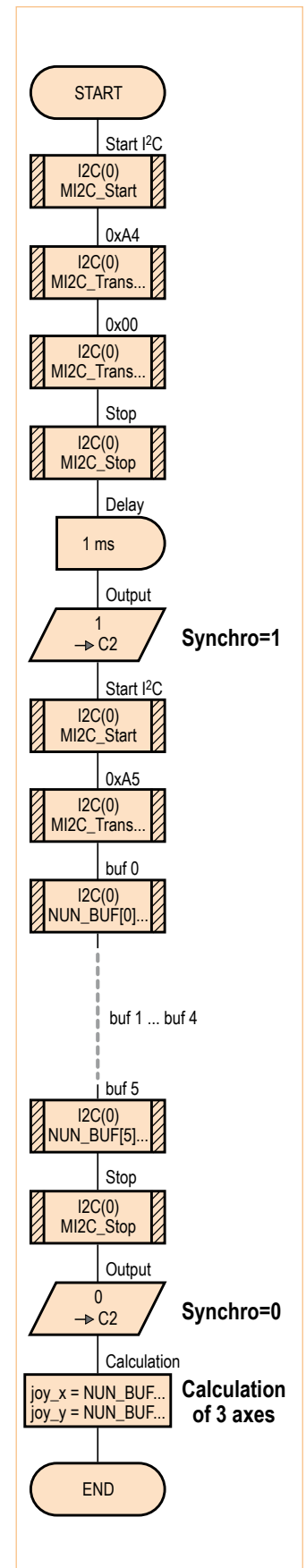


Bild 4. Lesen der Daten aus dem Nunchuk-Controller.



Bild 6. Für den Nunchuk-Stecker müssen in die Platine zwei Nuten gefräst oder gesägt werden.

sichtbar sein (siehe **Bild 5** und **Bild 6**). Ohne die Firmware für den PIC18F2550 (Dateiname: *nunchuk.hex*) ist der Adapter natürlich nicht funktionsfähig. Für das Laden in den Mikrocontroller stehen zwei Optionen zur Wahl: Entweder Sie bestellen den PIC programmiert im Elektor-Shop [1], was die geringste Mühe macht, oder Sie programmieren ein leeres Exemplar mit einem geeigneten Programmier. Die Datei *nunchuk.hex* enthält den Bootlader sowie das Programmbeispiel *hid_clavier_trame.fcf*, das unten kurz beschrieben ist. Auf dem PC muss kein Treiber installiert werden, denn aus Sicht des Betriebssystems ist der Adapter eine PC-Tastatur, die zu den Peripherie-Geräten der HID-Klasse gehört (HID=Human Interface Device). Für solche Peripherie sind die Treiber in den verbreiteten Betriebssystemen wie Windows, Linux, MacOS und anderen bereits integriert.

Tabelle 7.	
Adresse	
	Flash-Speicher
	Flowcode
	Applikation
\$0800	
\$07FF	Bootloader
\$0000	(Nicht überschreiben!)

Software

Die Programme wurden in Flowcode Version 4 geschrieben. Flowcode ist zum Programmieren des PIC18F2550 gut geeignet, das Einbinden der HID-Treiber, der USB-Schnittstelle und des I²C-Busses ist auch für Newcomer problemlos möglich. Wer mit C programmiert, hat noch mehr Möglichkeiten, doch muss der C-Compiler so konfiguriert werden, dass das Programm frühestens bei Adresse \$800 beginnt. Anderenfalls wird, wie aus **Tabelle 7** ersichtlich, der Bootlader überschrieben.

Der Bootloader

Der Mikrocontroller lädt eine Flowcode-Hex-Datei, wenn der Boot-Taster beim Einschalten der Betriebsspannung gedrückt gehalten wird. Wenn die LED D1 blinkt, heißt das, dass noch die Trei-

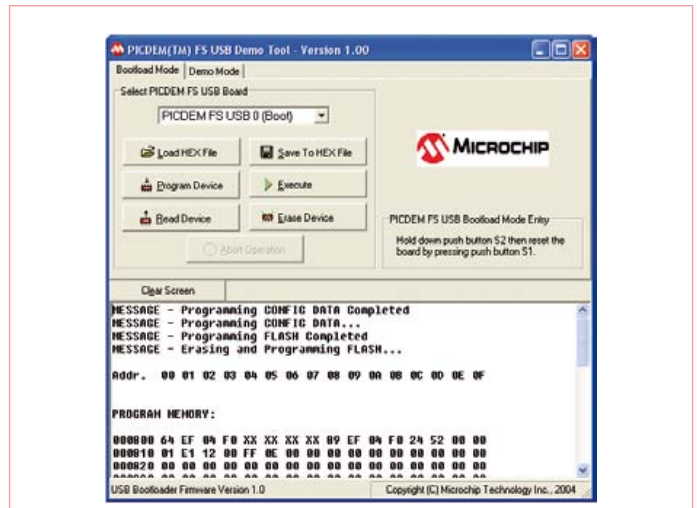


Bild 7. Das Programmier-Tool PDFSUSB.EXE macht den Inhalt des Flash-ROMs im Mikrocontroller PIC18F2550 zugänglich.

ber auf dem PC installiert werden müssen. Die diversen Dateien sind Bestandteile des Pakets *USB Framework* von Microchip [3]. Es ist zweckmäßig, für die Installation den vorgegebenen Ordner zu wählen. Die Treiber befinden sich im Ordner *Microchip Solutions v2010-10-19\USB Tools\MCHPUSB Custom Driver\MCHPUSB Driver\Release*, auch sie können von der Elektor-Website [1] heruntergeladen werden.

Starten Sie nach der Treiberinstallation das Programm *PDFSUSB.exe*, das Sie im Ordner *Microchip Solutions v2010-10-19\USB Tools\Pdfsusb* finden. Wählen Sie den Treiber *PICDEM FSUSB 0 (Boot)* aus. Laden Sie nun eine Hex-Datei, klicken Sie auf *Program Device* und anschließend auf *Execute* (siehe **Bild 7**).

Programmbeispiele

Auf der Elektor-Website [1] finden Sie vier Programme, die Sie beliebig in Ihr Projekt übernehmen können. An dieser Stelle soll eine Kurzbeschreibung genügen.

hid_clavier_trame.fcf (Basisprogramm, gehört zur Firmware)
Mit diesem Programm kann das Adaptersystem getestet werden.

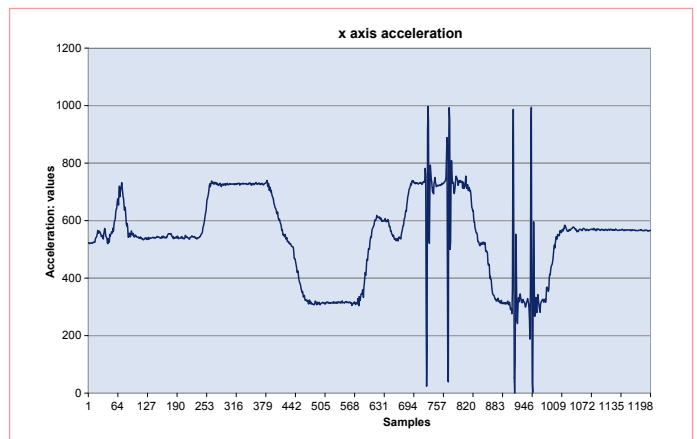


Bild 8. Beim Drücken des Tasters *Boot* werden die X-Werte des Beschleunigungssensors übertragen, sie können wie hier gezeigt grafisch dargestellt werden.

Wenn Sie den Taster Z des Nunchuk drücken, werden die Zahlenwerte der drei Achsen über die USB-Schnittstelle ausgegeben. Da der Adapter eine PC-Tastatur emuliert, erscheinen die Werte in einem Textprogramm wie zum Beispiel *Notepad*. Solange Sie Taster BP1 gedrückt halten, werden nur die Werte der X-Achse übertragen. Die empfangenen Daten sind, wie **Bild 8** zeigt, in einer Grafik darstellbar, dort können sie betrachtet und ausgewertet werden.

hid_joytick.fcf

Der Nunchuk verhält sich, was die X- und Y-Richtung betrifft, wie ein klassischer Joystick. Unter Windows lässt sich der Joystick im Konfigurationsfenster *Gamecontroller* der *Systemsteuerung* kalibrieren. Ohne Kalibrierung ist der Joystick nur bedingt einsatztauglich.

hid_souris_accel.fcf

Mit diesem Programm verwandelt sich der Nunchuk in eine Computermouse: Die X-Achse bewegt den Cursor in horizontaler Richtung, die Y-Achse bewegt ihn vertikal. Die Nunchuk-Taster Z und C sind mit dem linken und rechten Maustaster identisch. Beim Einsatz

Interesse geweckt?
Alle verwendeten Komponenten dieses
Bausatzes inkl. der benötigten Platine
stehen für Sie als fertig gepackter und
Teil für Teil frei editierbarer Warenkorb
zum Bestellen bereit: online unter
<http://r.ch.lt/nunchuk>
Ein Service von reichelt elektronik -
Bestellt • Beliefert • Begeistert



Anzeige

NEU



LabVIEW

Band 1: Einstieg in die Praxis

Das Programmpaket *Laboratory Virtual Instrument Engineering Workbench* (oder kurz LabVIEW) ist ein international anerkannter Standard zur Entwicklung und Gestaltung von Messgeräten und Prozesssteueroberflächen. Seine Universalität konfrontiert den LabVIEW-Einsteiger allerdings mit einer unübersichtlichen Vielfalt von Funktionen, die er ohne fundierte Anleitung kaum überblicken kann.

Hier setzt diese neue mehrteilige Lehrbuchreihe an: Von Grund auf werden in einfach nachvollziehbaren Schritten der Aufbau, die Struktur und die Verwendung von LabVIEW erklärt, in praktischen Beispielen dargestellt und mit Übungen vertieft. Die notwendigen Vorkenntnisse sind äußerst gering, die erreichbaren Ergebnisse dagegen äußerst sehenswert.

Der erste Band erläutert die Grunddatentypen und die zugehörigen numerischen Grundfunktionen ebenso ausführlich wie die elementaren Programmstrukturen.

Diese dreiteilige Lehrbuchreihe richtet sich an Schüler/Auszubildende technischer Bildungseinrichtungen, an Studenten von Fachhochschulen/Universitäten und an den engagierten Praktiker in der Industrie oder im privaten Bereich.

240 Seiten (kart.) · Format 17 x 23,5 cm · ISBN 978-3-89576-253-6
 € 34,80 · CHF 43,20



Weitere Infos & Bestellung unter www.elektor.de/labview1

Stückliste

Widerstände:

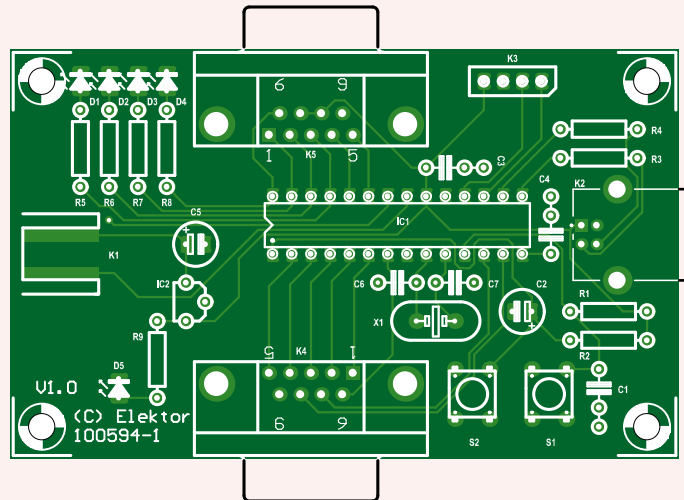
R1,R2 = 10 k
R3,R4 = 22 Ω
R5...R9 = 1 k

Kondensatoren:

C1 = 470 n
C2 = 10 μ /16 V stehend
C3 = 100 n
C4 = 220 n
C5 = 2 μ 2/16 V stehend
C6,C7 = 22 p

Halbleiter:

D1...D5 = LED 3 mm,
Low-current
IC1 = PIC18F2550-I/P, 28-Pin
DIP (programmiert, EPS
100594-1)
IC2 = LP2950ACZ-3.3



Außerdem:

S1,S2 = Drucktaster
K2 = USB-B-Buchse für
Platinenmontage
K3 = Stiftleiste 4-polig,
einreihig
K4,K5 = D-Sub-Buchse
(weiblich) 9-polig, für
Platinenmontage
X1 = Quarz 20 MHz
Platine EPS 100594-1, siehe [1]

als Maus ist die Empfindlichkeit des Nunchuk mehr oder weniger gewöhnungsbedürftig.

Nunchuk_dmx.fcf

Einen Disco-Projektor (Moving Head) steuert der Nunchuk, wenn dieses Programm zum Einsatz kommt. Schnittstelle ist hier der serielle Anschluss K3, er ist mit Leitung TXD (RC6) des Mikrocontrollers verbunden. Was noch fehlt, ist ein RS485-Umsetzer (wie zum Beispiel der 75176). Das Programm arbeitet mit dem DMX-Protokoll, es muss an den verwendeten Projektor angepasst werden.

(100594)gd

Weblinks

- [1] www.elektor.de/100594
- [2] http://wiibrew.org/wiki/Wiimote/Extension_Controllers#Nunchuk
- [3] USB FrameWork Microchip: http://ww1.microchip.com/downloads/en/DeviceDoc/MCHP_App_Lib_v2010_10_19_Installer.zip
Liste der Updates: www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2896
- [4] www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010280

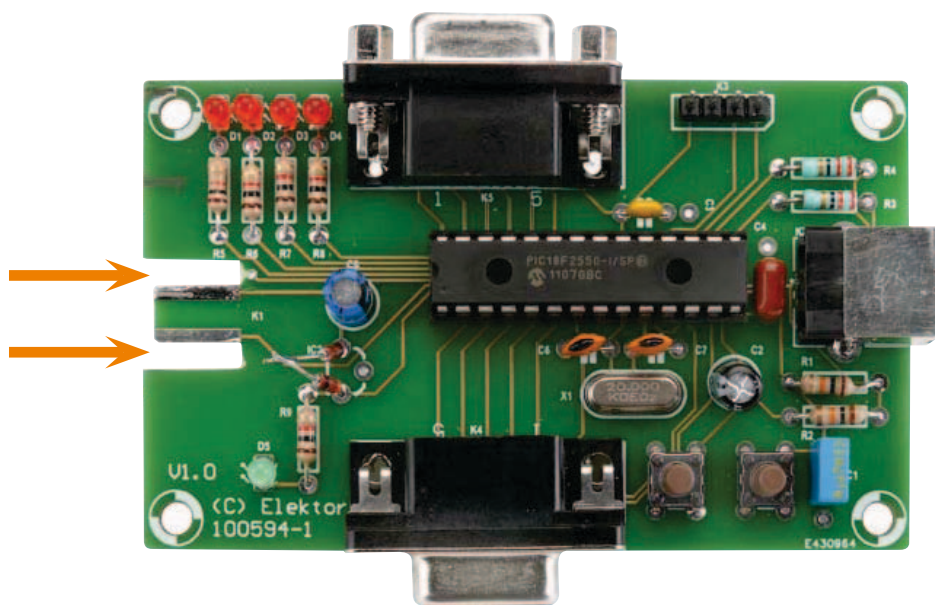


Bild 9. Die Nuten an den Seiten von K1 sind noch nicht vorhanden.

Der serielle Anschluss K3 wird nur von Anwenderprogrammen wie *Nunchuk_dmx.fcf* genutzt.
Beim abgebildeten Prototyp übernehmen provisorisch zwei Dioden die Funktion des 3,3-V-Spannungsreglers IC2.

● **Subscribe** to *audioXpress* magazine!

Do your **electronics speak** to you? Are the words "**audio,**" "**vacuum tubes,**" and "**speaker technology**" music to your ears?

Then you should be **reading audioXpress!**

Recently acquired by The Elektor Group, *audioXpress* has been providing engineers with incredible audio insight, inspiration and design ideas for over a decade. If you're an audio enthusiast who enjoys speaker building and amp design, or if you're interested in learning about tubes, driver testing, and vintage audio, then *audioXpress* is the magazine for you!

What will you find in *audioXpress*?

- In-depth interviews with audio industry luminaries
- Recurring columns by top experts on speaker building, driver testing, and amp construction
- Accessible engineering articles presenting inventive, real-world audio electronics applications and projects
- Thorough and honest reviews about products that will bring your audio experiences to new levels

Choose from print delivery, digital, or a combination of both for maximum accessibility.

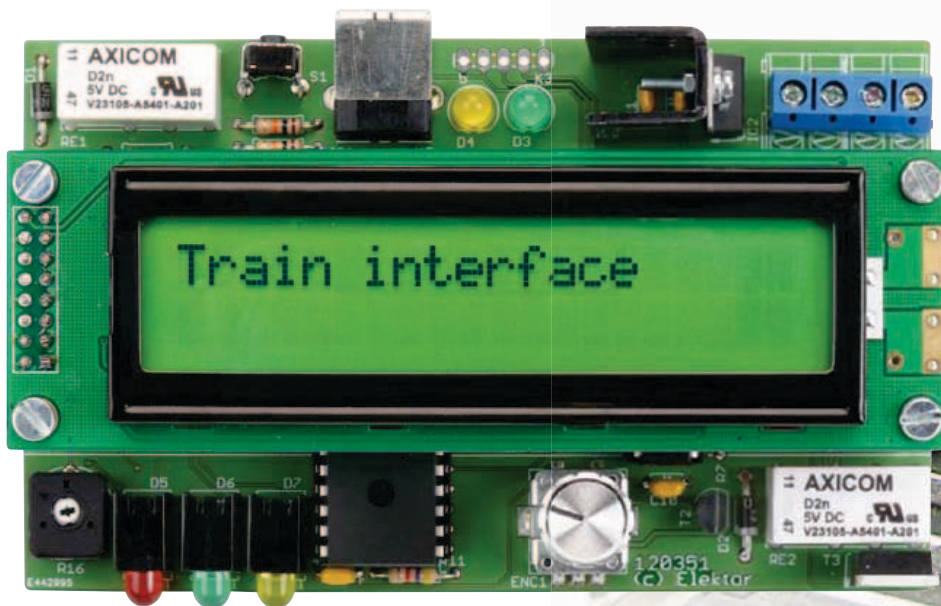
Subscribe to *audioXpress* at
www.audioamateur.com
today!

audioXpress



Modellbahn-Steuerung

Mit USB und Windows-Skript-Editor



Von Willem Tak (NL)

Auf Böden und in Kellern fristet so manche kleine Modellbahn ihr Dasein, die einst die Herzen längst erwachsener Kinder höher schlagen ließ. Heute ist verständlich, dass monoton im Kreis fahrende Bahnen damals schnell ihren Reiz verloren hatten.

Eine kleine, zeitgemäße Modellbahn-Steuerung holt das alte Spielzeug vom Abstellgleis zurück. Hochkomplexe Hightech-Züge benötigt es dafür ebenso wenig wie aufwendige Stellwerke.

Während der Vorweihnachtszeit schmückt meine Familie unser Heim, das Gestalten einer modellhaften Winterlandschaft in einer Zimmerecke hat Tradition. Etwas Bewegung bringt eine schon betagte Modellbahn hinein, doch die unaufhörlichen Runden auf stets gleicher Trasse sind kaum noch des Hinsehens wert. Könnte nicht der Zug seine Fahrtrichtung und seine Geschwindigkeit variieren? Könnte er nicht hier oder dort eine Fahrpause einlegen, so wie im richtigen Zugbetrieb?

Es gibt zahllose Wege, dies zu realisieren, beispielsweise das Modellbahnsystem EDiTS von Elektor. Doch EDiTS wäre für eine solche Aufgabe viel zu groß geraten.

Ich hatte die Idee, den Zug von einem Skript steuern zu lassen, in dem aufeinanderfolgende Anweisungen stehen. Die Anweisungen können beispielsweise den Zug vorwärts oder rückwärts fahren lassen, sie können ihn von Zeit zu Zeit stoppen, oder die Anweisung lautet, eine Station anzufahren und dort zu warten.

Hardware

Die Hardware habe ich für Modellbahnen entworfen, die mit Gleichspannung fahren. Ich habe sie mit dem System 9336 von Fleischmann getestet.

Zentrale Schaltstelle der Steuerung in **Bild 1** ist ein Mikrocontroller PIC18F4550. Dieser Mikrocontroller aus der bekannten PIC-Familie von Microchip kann Daten über USB mit einem PC austauschen. Auf dem PC kann ich die Skripte schreiben und über USB in den Mikrocontroller laden. Vorausset-



zung für USB ist beim PIC18F4550 nur, dass der Quarz X1 auf 24 MHz schwingt.

Für die Bedienung habe ich einen digitalen Drehencoder mit Drucktaster und ein zweizeiliges alphanu-

mung +5 V betrieben, die der PC über die USB-Leitung bereitstellt. Für den PC ist der Mikrocontroller ein sogenanntes HID-Gerät, wobei HID für *Human Interface Device* (Gerät für die Bedienung durch Menschen) steht. Wenn eine externe Betriebsspannung an K1

gern. Das aufbereitete Signal steuert T3, ein N-Kanal-MOSFET vom Typ FQP12N60. Dieser leistungsstarke Typ ist hohen Strömen bis etwa 10 A gewachsen. Der Fahrstrom kleiner Modellbahnen liegt meistens in der Größenordnung von nur 1 A. Für den

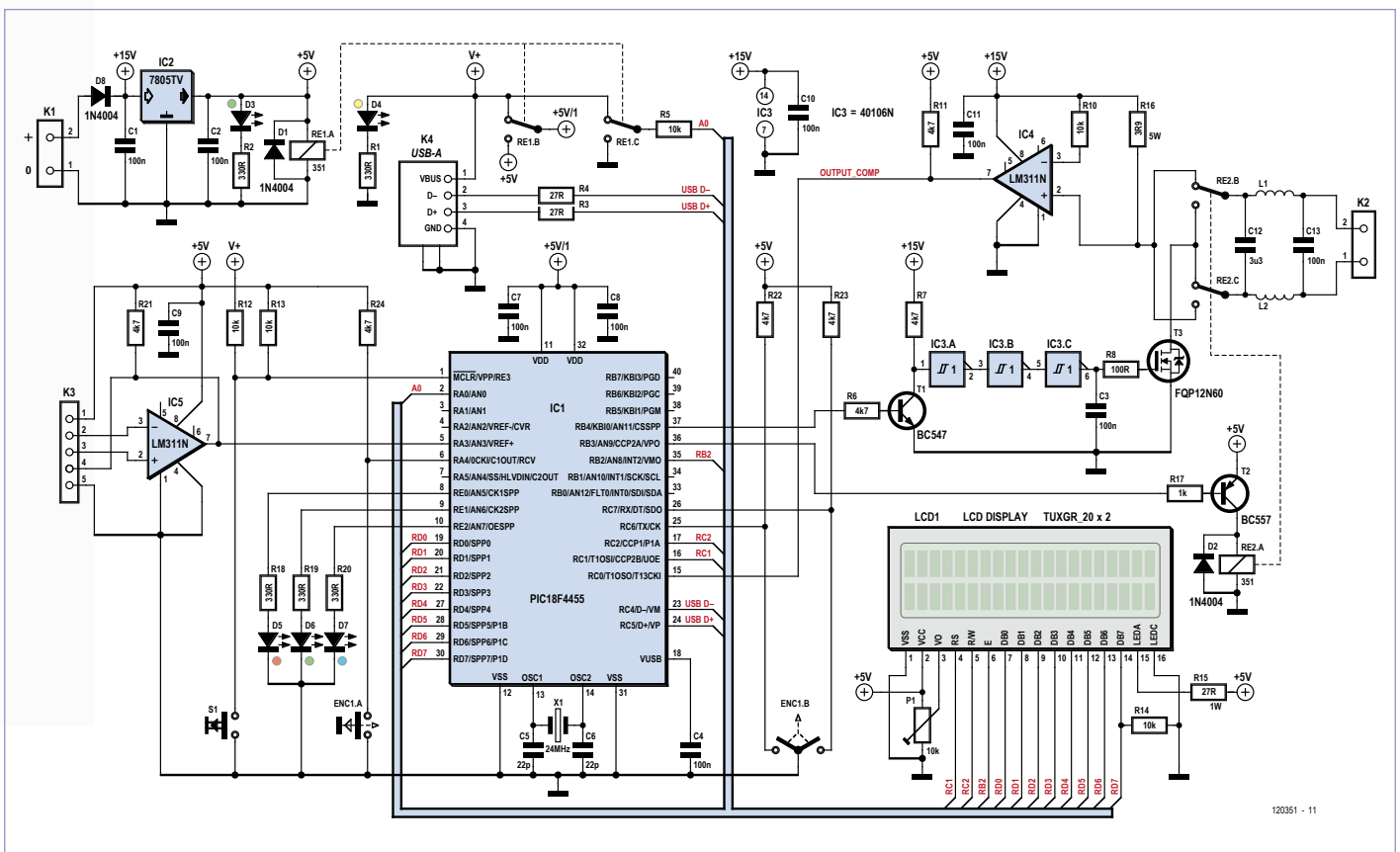


Bild 1. Der Mikrocontroller erzeugt ein pulsbreitenvariables Signal, das über den Power-MOSFET T3 die Modellbahn steuert.

merisches LC-Display gewählt. Das LC-Display wird ohne Zwischenpuffer vom Mikrocontroller gesteuert. Die Datenleitung RD7 wird vom Widerstand R14 nach Masse gezogen, weil die Software das Busy-Flag des LCDs nutzt.

Die Stromversorgung mit Relais Re1 mag ungewöhnlich erscheinen. Doch das liegt hauptsächlich daran, dass die Schaltung zwei Funktionen hat: Auf der einen Seite ist sie eine Zugsteuerung, auf der anderen ein USB-Gerät. Mit dem Relais in Ruhelage wird der Mikrocontroller an der Span-

nung +5 V betrieben, die der PC über die USB-Leitung bereitstellt. Für den PC ist der Mikrocontroller ein sogenanntes HID-Gerät, wobei HID für *Human Interface Device* (Gerät für die Bedienung durch Menschen) steht. Wenn eine externe Betriebsspannung an K1

liegt, zieht das Relais an, so dass der Mikrocontroller seine Betriebsspannung über den Spannungsregler IC2 erhält. Das pulsförmige Signal, das den Zug steuert, gibt der Mikrocontroller auf Leitung RB4 aus. Das Signal ist pulsbreitenvariabel, die Frequenz beträgt ungefähr 100 Hz. Wenn die Impulse schmal sind, fährt der Zug langsam, mit der Impulsbreite steigt auch die Zuggeschwindigkeit. Transistor T1 setzt die Impulshöhe 5 V des Mikrocontrollers auf 15 V herauf, während die Schmitt-Trigger IC3.A...IC3.C die Flankensteilheit stei-

ger. Das aufbereitete Signal steuert T3, ein N-Kanal-MOSFET vom Typ FQP12N60. Dieser leistungsstarke Typ ist hohen Strömen bis etwa 10 A gewachsen. Der Fahrstrom kleiner Modellbahnen liegt meistens in der Größenordnung von nur 1 A. Für den Fahrtrichtungswechsel ist Relais Re2 zuständig, das die Fahrspannung umpolt. Wie die Fahrspannung polrichtig an die Schienen anzuschließen ist, muss ausprobiert werden. Wenn die Fahrtrichtungen Vorwärts und Rückwärts vertauscht sind, lässt sich der Fehler durch Wechseln der Leitungsanschlüsse korrigieren.

Den Störspitzen, die der Lokomotor auf der Fahrspannung verursacht, wirken L1 und L2 sowie C12 und C13 entgegen. Dieses Störfilter ist notwendig, damit der Mikrocontroller und die übrigen Bauelemente

LED-Signale

LEDs intern (D3, D4):

- Gelb (D4) Leuchtet im USB-Modus
- Grün (D3) Leuchtet im Modus Zugsteuerung, bei externer Betriebsspannung

Front-LED (D7), USB-Modus:

- Blau (D7) Blinkt im USB-Modus bei der Initialisierung
- Blau (D7) Leuchtet nach dem Programmstart von *Trein.exe*
- Blau (D7) Kann vom Programm *Trein.exe* geschaltet werden

Front-LEDs (D5, D6, D7), Zugsteuerung:

- Rot (D5) Leuchtet, wenn der Zug fährt
- Rot (D5) und Grün (D6) Leuchtet, wenn der Zug hält oder wartet
- Grün (D6) Leuchtet, wenn der Zug an der Station hält
- Blau (D7) Leuchtet, wenn der Zug hält und die Station nicht gefunden wurde
- Rot, Grün und Blau Leuchtet bei LC-Display-Fehler
- Rot und Blau Leuchtet bei EEPROM-Fehler
- LED-Lauflicht Zugbewegungen weichen vom Skript ab

ungestört arbeiten können. Auch Kondensator C3 zwischen IC3.C und MOSFET T3 trägt zur Störsicherheit bei, was allerdings in geringem Maß zu Lasten der Signal-Flankensteilheit geht.

OpAmp IC4 stellt eine Rückkopplung von der Fahrspannung zum Mikrocontroller her, so dass der Mikrocontroller erkennen kann, ob die Anweisungen ausgeführt werden. Wenn der Zug auf den Schienen fährt,

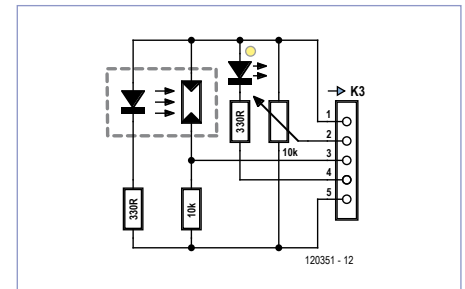


Bild 2. Die Lichtschranke ist mit dem Steckverbinder K3 auf der Mikrocontroller-Platine verbunden.

liegt zwischen den Eingängen des OpAmps ein pulsartiges Signal.

Das Anhalten des Zugs an einer Station erkennt eine Lichtschranke, die Schaltung geht aus **Bild 2** hervor. Die mechanische Konstruktion besteht aus einem hufeisenförmigen Bügel, der am Haltepunkt eine Schiene von unten umschließt. Am linken Hufeisenschwanz ist die LED montiert, am rechten befindet sich der LDR. Sobald der Zug die Lichtschranke erreicht, unterbricht er den Lichtstrahl zwischen LED und LDR. Die Empfindlichkeit der Lichtschranke lässt sich mit dem Trimpoti einstellen.

Die diversen farbigen LEDs signalisieren die Betriebszustände der Modellbahnsteuerung. Wenn die Schaltung über USB mit Strom versorgt wird, leuchtet die gelbe LED D4. Sobald die USB-Verbindung hergestellt ist, blinkt die blaue LED D7. Diese LED leuchtet kontinuierlich, sobald das Programm *Trein.exe* auf dem PC läuft. Das Anschließen einer externen Betriebsspannung für den Zugbetrieb (15 V) hat zur Folge, dass die grüne LED D3 leuchtet. In dieser Konfiguration blinkt die rote LED D5 während der Initialisierung, anschließend leuchtet sie kontinuierlich. Die Übersicht „LED-Signale“ fasst die LEDs und ihre Bedeutungen zusammen.

Software

Das Programm für den Mikrocontroller PIC18F4550 wurde in Assembler geschrieben, die Sprache des PC-Programms war Visual Basic (VB). Das Assembler-Programm besteht aus zwei Modulen: Zugsteuerung und USB-HID für die Skripteingabe.

Bits und Bytes

Jede Anweisung im Skript besteht aus zwei Bytes:

- Hohes Byte, Bit 7: Logisch 1 bei Vorwärtsfahrt
- Hohes Byte, Bit 6: Logisch 1 bei Rückwärtsfahrt
- Hohes Byte, Bit 5: Logisch 1 bei Station anfahren
- Hohes Byte, Bit 4: Logisch 1 bei Wartehalt
- Hohes Byte, Bit 3: Service-Bit (siehe unten)
- Hohes Byte, Bit 2...0: Zuggeschwindigkeit (1...7)
- Niedriges Byte: Dauer der Aktion (1...255 s)

Das Service-Bit (Bit 3) hat unterschiedliche Funktionen, abhängig vom Byte, in dem es steht:

- In Byte 0: Toggle-Bit für EEPROM-Schreibaktionen
- In Byte 2: Logisch 1 markiert Skript 1
- In Byte 4: Logisch 1 markiert Skript 2
- In Byte 6: Logisch 1 markiert Skript 3
- In Byte 8: Status der blauen USB-Front-LED

USB-HID

Für den USB-HID-Teil wurde der Standard des Mikrocontrollers PIC18F4550 übernommen. Die VID lautet 0D59, die PID ist 5275. Nach korrekter Enumeration startet die auf 6 ms eingestellte USB-Schleife. Mit dem PC tauscht der USB-Port die Daten über einen 64 Byte großen Puffer aus. Die Skripte werden aus der VB-Umgebung über den Puffer zum PIC18F4550 transportiert. Das USB-Programm speichert die Skripte im EEPROM des Mikrocontrollers, in umgekehrter Richtung können die gespeicherten Skripte vom USB-Programm gelesen werden. Bis zu drei Skripte lassen sich gleichzeitig verwalten. Die Anweisungen werden in das Formular des VB-Programms eingegeben. Ein Skript kann bis 32 Anweisungen enthalten, eine Länge, die in der Praxis ausreicht. Die maximale Skriptlänge ist an die Kapazität des HID-Puffers (64 Byte) angepasst. Neben den 32 Anweisungen haben noch diverse Steuer-Bits Platz, denn jede Anweisung besteht aus nur sieben Bits. Die diversen Bits, Bytes und ihre Bedeutungen erklärt die nebenstehende Übersicht „Bits und Bytes“.

Der angesprochene Programmteil übernimmt auch das Timing mit dem USB-Port des PCs. Weil die niedrige Schreibgeschwindigkeit des EEPROMs zu Problemen führen kann (nach Angaben von Microchip: 4 ms), wird in jeder USB-Schleife nur ein Byte in das EEPROM geschrieben.

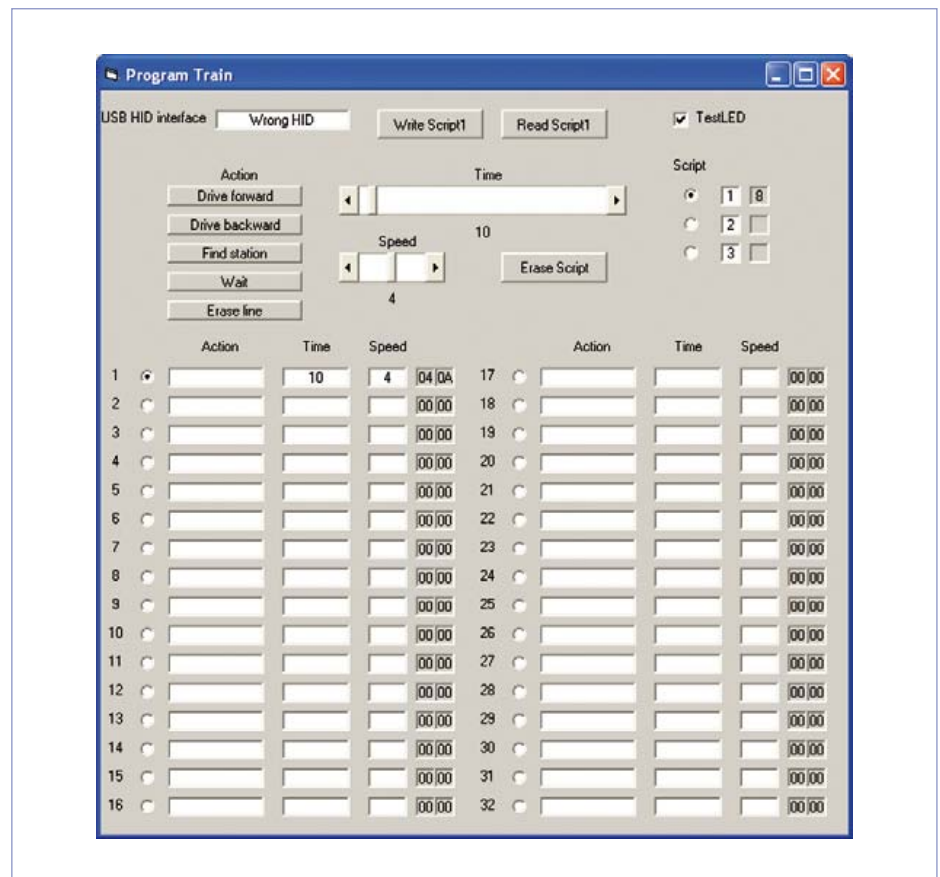
Der Schreibvorgang beginnt mit einem Toggle-Byte aus der VB-Umgebung und dem Setzen diverser Flags im USB-Programmteil. Jedes Byte belegt 64 aneinander anschließende Zyklen. Während dieser Zeit signalisiert der PC, dass eine Wartepause aktiv ist und keine Daten eingegeben werden können.

Die Skript-Daten werden (außer beim Starten) nicht unmittelbar aus dem EEPROM gelesen. Das Programm sorgt dafür, dass die aktuellen Daten über spezielle Puffer des Mikrocontrollers in den USB-Eingangspuffer transportiert werden.

Zugsteuerung

Die reguläre Betriebsart des Mikrocontrollers ist die Zugsteuerung, sie wird durch das Signal auf Leitung A0 aktiviert.

Die Bedienoberfläche besteht aus dem zweizeiligen LC-Display und dem Drehencoder



Fahrbewegungen und Halte werden in ein PC-Programm eingegeben, das unter Windows läuft.

mit Drucktaster. Mit dem Drehencoder kann die Liste der Anweisungen durchscrollt werden, ein Druck auf den Taster wählt die Anweisung aus.

Das Assembler-Programm der Zugsteuerung ist relativ einfach strukturiert. Aus dem Skript wird eine Zeile gelesen, anschließend wird die dort stehende Anweisung ausgeführt. Die meisten Anweisungen beziehen sich auf die Routine, die das pulsbreitenvariable Signal generiert. Dort bestimmt ein Timer, wie lange eine Aktion dauert. Das LC-Display zeigt das Ausführen der Anweisungen an. In der Betriebsart Zugsteuerung erscheint zuerst das Menü auf dem Display. Von dort ist das Ausführen eines von drei Skripten möglich, oder der Zug wird von Hand gesteuert. Die Einstellungen sind über den Reset-Taster zugänglich.

Ein zentraler Teil des Programms ist die Rou-

tine DRIVE. Die Routine wird mit 100 Hz durchlaufen, wobei die Zeiten durch Schleifen, nicht durch Interrupts gesteuert werden. Die Frequenz des generierten pulsbreitenvariablen Signals beträgt ebenfalls 100 Hz. Impuls- und Pausenzeiten werden Tabellen entnommen, sie hängen von der eingestellten Mindestimpulsbreite ab. Ferner erzeugt die Routine ein Sekundensignal, das zum Timing diverser anderer Aktionen dient.

Der Drehencoder kann von zwei Routinen mit unterschiedlichen Empfindlichkeiten (*Fast* und *Secure*) gelesen werden. Der Drucktaster des Drehencoders dient zum Starten oder Abbrechen einer Aktion.

Skripte erstellen

Mit dem Windows-Programm *Trein.exe* werden die Skripte erstellt, die von der

Modellbahnsteuerung ausgeführt werden sollen. Zuerst muss eine USB-Verbindung zu einem PC hergestellt werden, auf dem aktiv das Betriebssystem Windows läuft. Windows muss die Steuerung als HID-Gerät erkennen. Dann wird das Programm *Trein.exe* gestartet. Die blaue LED D7 leuchtet auf, und im Feld *USB HID Interface* des Programms erscheint die Meldung *Trein Interface*. Die Funktion der USB-Verbindung kann mit *Test LED* getestet werden. Beim Aktivieren von *Test LED* muss die blaue LED aufleuchten. Mit dem Programm können nicht nur Skripte erstellt, angepasst und in die Modellbahnsteuerung geladen werden, auch das Lesen des Speichers der Modellbahnsteuerung (EEPROM) ist möglich.

Wie schon erwähnt, können drei Skripte verwaltet werden, jedes Skript kann bis zu 32 Zeilen enthalten. In den Zeilen können folgende Anweisungen stehen: *Vorwärts*, *Rückwärts*, *Station anfahren* und *Warten*. Das Programm wird ausschließlich mit der Maus bedient. Zu jeder der genannten Anweisungen gehört eine variable Zeit, die mit einem Schieber im Bereich 1...255 s wählbar ist. Von der Einstellung hängt die Dauer der Aktion ab.

Mit Ausnahme von *Warten* gehört zu jeder Anweisung auch eine variable Geschwindigkeit, sie kann mit einem zweiten Schieber relativ im Bereich 1...7 eingestellt werden. Das Erstellen eines Skripts ist äußerst einfach: Wählen Sie einen Skriptnamen, markieren Sie eine Zeile, wählen Sie die Anweisung und stellen Sie Zeit und Geschwindigkeit ein. Normalerweise müssen Skripte mit einer Leerzeile enden. Wenn in allen 32 Zeilen Anweisungen stehen, kehrt das Programm nach Ausführen der 32. Zeile zur ersten Zeile zurück. Für die 32. Zeile des ersten Skripts gilt eine Besonderheit: Wenn hier alle Zeilen eine Funktion haben, wird dieses Skript mit der ersten Zeile des zweiten Skripts fortgesetzt. Falls auch dort alle 32 Zeilen mit Anweisungen gefüllt sind, springt das Programm zur ersten Zeile des dritten Skripts. Durch diese Verkettung kann ein Skript ausgeführt werden, das bis 96 Zeilen lang ist.

Über die Schaltflächen *Read Script* oder *Write Script* ist das Lesen oder Schreiben des Speichers im Mikrocontroller möglich. Schreibaufträge werden sofort

ausgeführt, das Programm fordert keine zusätzliche Bestätigung an. Während des Lesens oder Schreibens ist das Programm für andere Aktionen gesperrt, dies wird in einem Fenster angezeigt. In grau getönten Feldern erscheinen die Werte, die in den Speicher geschrieben werden.

Bau

Für die Modellbahnsteuerung hat das Elektor-Labor die Platine in **Bild 4** entworfen, auf der alle Bauteile einschließlich des Power-MOSFETs T3 ihren Platz haben. Der Aufbau ist nicht schwierig, denn alle Bauelemente sind bedrahtet. Für die Wärmeabfuhr an Spannungsregler IC2 ist ein kleinflächiger Kühlkörper nötig. Der MOSFET befindet sich am Platinenrand, damit er zur Kühlung bei Bedarf an die Wand eines Metallgehäuses geschraubt werden kann.

Zuletzt wird der programmierte Mikrocontroller in seine Fassung gesetzt. Der Elektor-Shop hat Mikrocontroller des Typs PIC18F4550 im Angebot, die für die Modellbahnsteuerung programmiert sind. Das LC-Display wird auf 20 mm langen Abstandsrollen montiert, die Befestigungslöcher sind auf der Platine bereits vorhanden. Die Verbindungen zur Platine können über zwei kurze Bandkabelabschnitte oder einen Steckverbinder mit 2 · 8 extra langen Pins hergestellt werden.

Nachdem das LC-Display montiert ist, kann die Platine über USB mit dem PC verbunden werden. Dem Schreiben eines Skripts, das in das EEPROM des Mikrocontrollers geladen wird, steht nichts mehr im Weg.

Betrieb

Das Anschließen der Zugsteuerung an die Modellbahn ist nicht schwierig. Verbinden Sie K1 mit einem Netzteil, das bei 15 V Gleichspannung den Fahrstrom liefern kann (abhängig von der Modellbahn), und K2 verbinden Sie mit den Schienen. Der in der Stückliste für die Drosseln L1 und L2 angegebene Typ verträgt Ströme bis etwa 1 A, was für kleine Modellbahnen mehr als ausreicht.

Nach Einschalten der Betriebsspannung befindet sich das System im Modus „Zugsteuerung“. Die rote LED leuchtet auf, die Zugsteuerung kann jetzt mit dem Drehencoder und dem Drucktaster bedient werden. Auf dem LC-Display erscheint eine kurze Begrüßung, anschließend stehen fünf Aktionen zur Wahl: Skript 1, Skript 2, Skript 3, Handsteuerung und Skript-Test. Mit dem Drucktaster wird die angezeigte Aktion ausgewählt.

Eine Skript-Aktion startet das zugehörige Skript. Das LC-Display zeigt in der ersten Zeile die aktuelle Anweisung zusammen mit ihrer Folgenummer an. Die zweite Zeile gibt Auskunft über die gegenwärtig ausgeführte Aktion, gegebenenfalls einschließlich der Fahrtrichtung. Außerdem wird die verbleibende Zeit bis zum Ende der Aktion in Sekunden angezeigt.

Das Vorwärtsfahren, Rückwärtsfahren und Halten bedarf eigentlich keiner Erklärung, das Anfahren einer Station ist dagegen etwas komplizierter. Diese Aktion besteht aus zwei Teilaktionen: Während der Zug die Lichtschranke ansteuert, zeigt das LC-Display die dafür verfüg-



Stückliste

Widerstände:

R1,R2,R18...R20 = 330 Ω
R3,R4 = 27 Ω
R5,R10,R12...R14 = 10 k
R6,R7,R11,R21...R24 = 4k7
R8 = 100 Ω
R15 = 27 Ω / 1 W
R16 = 3 Ω 9/5 W
R17 = 1 k
P1 = 10 k Trimpoti

Kondensatoren:

C1...C4,C7...C11,C13 = 100 n
C5,C6 = 22 p
C12 = 3 μ 3 MKT, Raster 15 mm

Induktivitäten:

L1,L2 = Drossel 330 μ H, 0,9 A, 0 Ω 32 (z. B. Panasonic ELC10D331E, Farnell 1749073)

Halbleiter:

D1,D2,D8 = 1N4004
D3,D6 = LED 5 mm grün
D4 = LED 5 mm gelb
D5 = LED 5 mm rot
D7 = LED 5 mm blau
T1 = BC547
T2 = BC557
T3 = FQP12N60
IC1 = PIC18F4455-I/P (programmiert: EPS 120351-41)
IC2 = 7805
IC3 = 40106
IC4,IC5 = LM311

Außerdem:

RE1,RE2 = Platinenrelais 5 V (z. B.

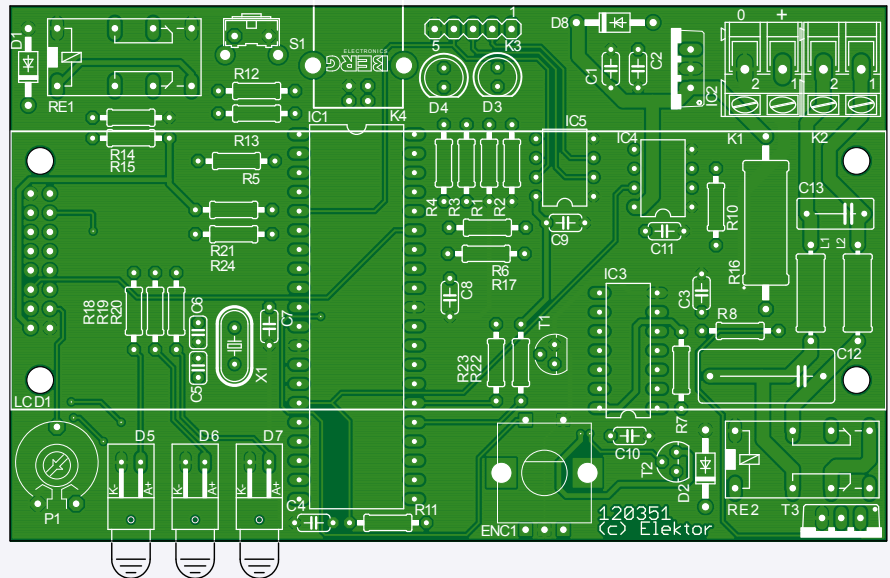


Bild 4. Auf der Platine sind vier Bohrungen für die Montage des LC-Displays mit Abstandsrollen vorhanden.

V23106-A5401-A201)

LCD1 = LCD alphanumerisch 2 · 20 Zeichen, mit Beleuchtung (z. B. Midas MC22005A6W-SPTLY)

ENC1 = Drehencoder mit Drucktaster (z. B. Bourns PEC11-4230F-S0024)

K1,K2 = Schraubklemmverbinder 2-polig, Raster 5 mm

K3 = Stiftleiste 5-polig, einreihig

K4 = USB-B-Buchse, abgewinkelt für Platinenmontage

S1 = Drucktaster (z. B. Omron B3F-3100)

X1 = Quarz 24 MHz

Platine 120351-1

(siehe www.elektor.de/120351)

bare Zeit an. Die tatsächlich benötigte Zeit ist unbekannt, denn sie hängt von der Fahrstrecke und der Zuggeschwindigkeit ab. Ist das Ziel erreicht, beginnt die Wartezeit, die im Skript für den Aufenthalt festgelegt ist. Jede Aktion kann zu beliebiger Zeit durch

Betätigen des Drucktasters abgebrochen werden.

Eine ergänzende Beschreibung, unter anderem mit weiteren Details zur Bedienung, befindet sich auf der Elektor-Projektseite [1] im Internet.

(120351)gd

Weblink

[1] www.elektor.de/120351

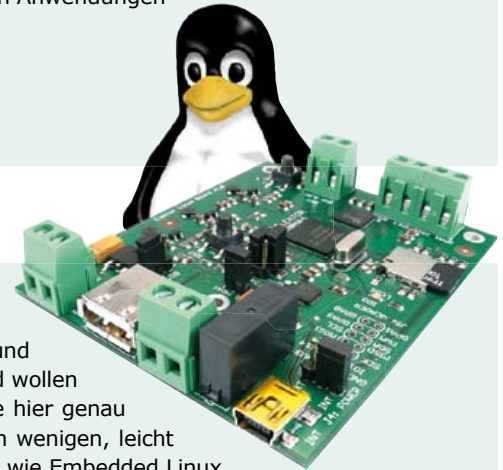


Embedded Linux in Theorie und Praxis (1-tägiges Seminar)

NEU

Dieses Seminar dient als Einstieg in die Linux Welt im Bereich Embedded-Systeme. In diesem Tagesseminar soll dem Mikrocontroller oder auch Computer erfahrenen Benutzer gezeigt werden wie ein Linux Betriebssystem in einem Embedded-System bzw. mit Mikrocontrollern verwendet werden kann. Wie es dort startet und arbeitet, wie einfach Anwendungen in der gewünschten Programmiersprache geschrieben und installiert werden können.

Referent: Dipl.-Inf.(FH) Benedikt Sauter
Teilnahmegebühr: 359,00 € (inkl. MwSt.)



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

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

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

Eagle PCB und Design (2-tägiges Seminar)



In diesem 2-tägigen Kurs werden Sie lernen, wie man mit dem Programm Eagle der Firma Cadsoft GmbH Leiterplatten entflechten kann. Begonnen wird mit dem Zeichnen von Schaltplänen unter Verwendung von Standard-Eagle-Bibliotheken. Sie lernen, wie man Schaltpläne über mehrere Seiten hinweg zeichnet und wie man eigene Bibliotheken und Bauteile erstellt. Anschließend werden Sie lernen, wie man aus dem Schaltplan eine Leiterplatte definiert und diese dann entflechtet (layoutet). Dabei werden auch Spezialkenntnisse zum Layouten von HF-Schaltungen, wie Leitungswellenwiderstand, vermittelt. Nach erfolgreichem Layout werden Produktionsdaten erzeugt, die man benötigt, wenn man die Platine fertigen lassen möchte. Dabei wird auch auf die verschiedenen Produktionsarten wie fräsen und ätzen eingegangen. Zum Abschluss gibt es Tipps und Tricks zum Umgang mit Eagle.

Referent: Prof. Dr.-Ing. Francesco P. Volpe
Teilnahmegebühr: 849,00 € (inkl. MwSt.)

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

NEU



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

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

Schaltungsdesign mit Mikrocontroller (2-tägiges Seminar)

Das 2-tägiges Seminar ist interaktiv gestaltet. Vortrag und praktische Übungen wechseln sich ab. Für die Programmierung von Mikrocontrollern hat sich die Programmiersprache C weitgehend durchgesetzt. Allerdings unterscheidet sich die Programmierung von Mikrocontrollern stark von der Programmierung von Anwendersoftware auf einem PC.

Anhand kleiner praktischer Übungen wie Steuern von LEDs oder Relais, Abfrage von Sensoren (Tastern, Potentiometern etc.), Erfassung von analogen Signalen mit integrierten AD-Wandlern und Kommunikation mit PCs lernt man den Einstieg in die hardwarenahe Programmierung mit der Programmiersprache C dann so nebenher mit.

Referent: Dipl.-Ing. Franz Peter Zantis

Teilnahmegebühr: 799,00 € (inkl. MwSt.)

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

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

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

Referent: Prof. Dr.-Ing. Francesco P. Volpe

Teilnahmegebühr: 1.299,00 € (inkl. MwSt.)

Elektronik steuern mit Android (AndroPod)

NEU

Android bietet als Betriebssystem die notwendigen Freiheiten um mit der Leistung des Smartphones eigene Elektronik ansteuern zu können. Der Einstieg in die das Arbeiten mit Android und eigener Elektronik ist hier denkbar leicht wenn man einige Grundinformationen hat und einem der generelle Ablauf der Entwicklung klar ist. Darum wird in diesem Kurs ein Projekt von Grund auf aufgerollt und in dem Kurs einmal der Ablauf von Beginn bis zum Ende durchlaufen. Somit ist werden durch „Learning by Doing“ alle möglichen Problemstellen im Ablauf einer Entwicklung aufgezeigt und erklärt wie diese behoben werden.

Referent: B. Wörndl-Aichriedler

Teilnahmegebühr: 349,00 € (inkl. MwSt.)



Workshops * Seminare * Kurse * Weiterbildungen
Top-Fachleute aus der Branche referieren
über ein faszinierendes Thema!

SEHR GUT
9 von 10 Seminaren
werden von unseren
Teilnehmern mit
sehr gut bewertet.

Embedded Linux in Theorie und Praxis

11. bis 13.09.2012 Dortmund
21.09.2012 Füssen
13. bis 15.11.2012 Hannover
20. bis 22.11.2012 Zürich

Seminar Schaltungsdesign mit Mikrocontroller

12. + 13.09.2012 Dortmund
20. + 21.09.2012 Füssen
16. + 17.10.2012 München

Eagle PCB und Design

12. + 13.09.2012 Dortmund
18. + 19.09.2012 Füssen
07. + 08.11.2012 Hanau

Elektronik steuern mit Android (AndroPod)

20.09.2012 Füssen

Seminar Sounddesign

27.09.2012 Dortmund

PIC-Mikrocontroller- Programmierung in C

26. bis 28.09.2012 Dortmund
16. bis 18.10.2012 München
11. bis 13.12.2012 Zürich

Seminar Messen, Steuern, Regeln mit VB.NET

25. + 26.09.2012 Dortmund
07. + 08.11.2012 Hanau
12. + 13.12.2012 München



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

USB-Isolator

Nie wieder Ärger mit Masseschleifen

Von Raymond Vermeulen (Elektor-Labor)

Masseschleifen zwischen PCs und Systemen, die über USB miteinander kommunizieren, sind oft Ursache hoher Störpegel auf den Signalen. Der USB-Isolator ist der Retter in der Not, wenn die Systeme in störintensiver Umgebung arbeiten müssen. Er trennt die Systeme galvanisch voneinander und schlägt gleichzeitig eine Brücke, sowohl für die Daten- als auch die Betriebsspannungsleitungen.



Der USB-Isolator entstand, als der Autor an der Entwicklung eines portablen Oszilloskops für Android arbeitete. Ein Problem waren die Störspannungen, die das Nutzsignal überlagerten, sobald ein Netzteil oder ein am Stromnetz betriebener PC angeschlossen wurden. Die Ursache war die von den Masse- und Schutzleitern gebildete Erdungsschleife. Diese Schleife galt es zu unterbrechen, doch an welcher Stelle? War es sinnvoller, eine Trennung im analogen Systemteil oder auf der Seite der USB-Verbindung vorzunehmen? Nach intensiver Analyse war sicher, dass die USB-Verbindung der richtige Ort ist.

Digital Isolator ADuM3160

Der ADuM3160 und der eng verwandte Typ ADuM4160, beide von Analog Devices [1], sind vermutlich die einzigen USB-Isolator-Bausteine, die der Markt gegenwärtig bietet. Auch in den meisten USB-Isolatoren, die von der Geräteindustrie angeboten werden, finden sich diese Chips. Die Bausteine unterstützen sowohl USB Low Speed (1,5 Mbit/s) als auch Full Speed (12 Mbit/s). USB-Isolatoren für High Speed (480 Mbit/s) sind nach Angaben von Analog Devices noch nicht

Wichtige Eigenschaften

- Galvanische Trennung der Daten- und Betriebsspannungsleitungen
- Datenübertragungsgeschwindigkeit 1,5 Mbit/s oder 12 Mbit/s
- Stromversorgung auf der Geräteseite über zweiten USB-Port oder Steckernetzteil (5 V/min. 0,5 A)
- Transformatorisolation = 1500 V_{AC} für 1 min
- ADuM3160-Isolation = 565 V_{Peak} für 50 Jahre
- Maximaler Ausgangsstrom I_{Out} = 500 mA @ 5 V

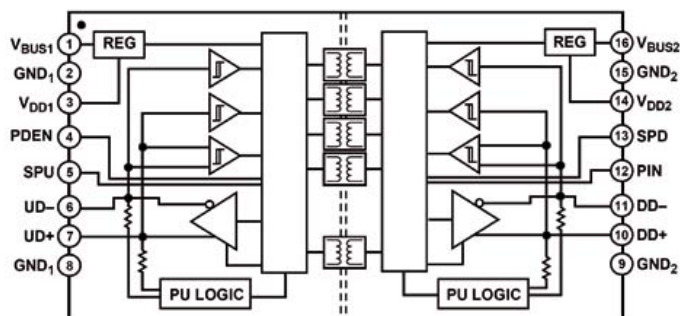


Bild 1. Internes Funktionsschema des „Digital Isolators“ ADuM3160 von AMD. Monolithische Transformatoren übernehmen die galvanische Trennung.

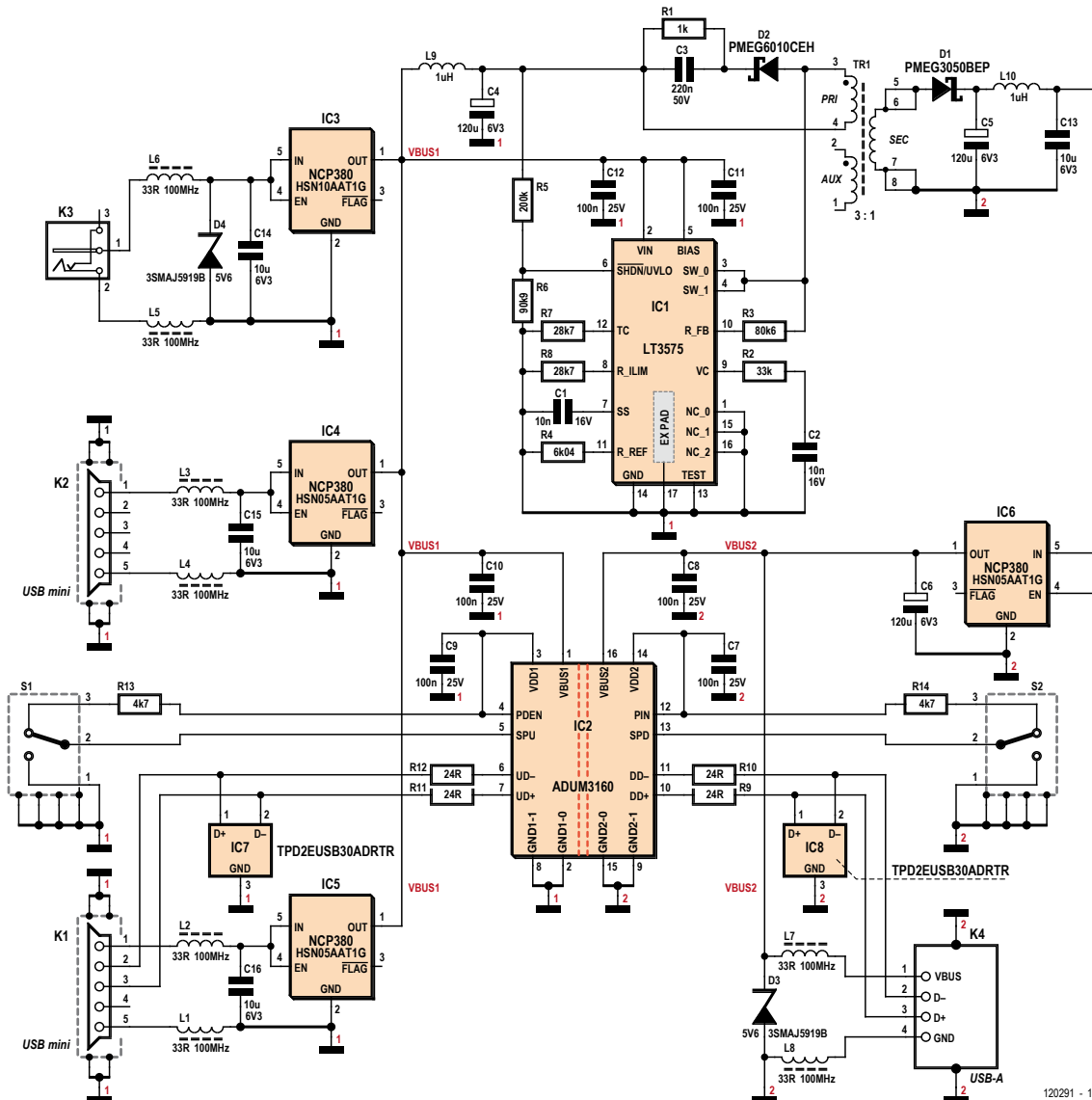


Bild 2. Während der ADuM3160 (IC2) die Datenleitungen isoliert, trennt der Flyback Converter LT3575 (IC1) die Betriebsspannungsleitungen.

realisierbar, weil zuverlässige digitale Isolatoren für hohe Geschwindigkeiten erst in der Entwicklung sind.

In Bild 1 ist das interne Funktionsschema des ADuM3160 wiedergegeben. Durch Kombinieren von High-Speed-CMOS-Technik und Transformatoren mit Luftkoppelungen („iCoupler“) ist es Analog Devices gelungen, die Primärseite und Sekundär-

seite so voneinander zu isolieren, dass auch höchste Sicherheitsanforderungen erfüllt werden. Der Baustein erkennt die Datenflussrichtung, indem er die Eigenschaften der empfangenen Datenblöcke auswertet. Die mittlere Verzögerung der Daten liegt trotz der hervorragenden Isolation bei Zeiten, die mit durchschnittlichen USB-Hubs vergleichbar sind.

Flyback Converter LT3575

USB-Verbindungen der Version 2.0 bestehen aus vier Adern, zwei Adern transportieren die Daten, zwei weitere Adern transportieren die Betriebsspannung 5 V. Der Baustein ADuM3160 trennt jedoch nur die Datenleitungen galvanisch voneinander. Deshalb müssten die Systemseiten eigentlich mit getrennten Stromversorgungen arbeiten.

Stückliste

Widerstände:

(1 %/0,1 W, SMD 0603, sofern nicht anders angegeben)

R1 = 1 k / 1 %, 0,5 W (SMD 1206)
 R2 = 33 k
 R3 = 80k6
 R4 = 6k04
 R5 = 200 k
 R6 = 90k9
 R7, R8 = 28k7
 R9, R10, R11, R12 = 24 Ω
 R13, R14 = 4k7

Kondensatoren:

C1, C2 = 10 n/16 V X7R (SMD 0603)
 C3 = 220 n/50 V X5R (SMD 0603)
 C4, C5, C6 = 120 μ /6,3 V (Nichicon RFS0J121M-CN1G5, Typ C)
 C7, C8, C9, C10, C11, C12 = 100 n/25 V X7R (SMD 0603)
 C13, C14, C15, C16 = 10 μ /6,3 V X5R (SMD 0805)

Induktivitäten:

L1, L2, L3, L4, L5, L6, L7, L8 = Ferritdrossel
 33 Ω @ 100 MHz (SMD 0603, z. B. Murata BLM18PG330SN1D)
 L9, L10 = 1 μ H/1200 mA (SMD 1007, z. B. Taiyo Yuden CB2518T1R0M)

Halbleiter:

D1 = PMEG3050BEP (Schottky-Diode, NXP, SOD-128)
 D2 = PMEG6010CEH (Schottky-Diode, NXP, SOD-123f)
 D3, D4 = Zenerdiode 5,6 V/3 W
 IC1 = LT3575EFE#PBF (Flyback-Controller, LT, TSOP-16)
 IC2 = ADUM3160BRWZ (USB-Data-Isolator, AD, SOICW-16)
 IC3 = NCP380HSN10AAT1G (Current-Limit-Switch 1,0 A, On Semiconductor, TSOP-5)
 IC4, IC5, IC6 = NCP380HSN05AAT1G (Current-Limit-Switch 0,5 A, On Semiconductor, TSOP-5)

IC7, IC8 = TPD2EUSB30ADRTR (USB-ESD-Protection, TI, SOT-3)

Außerdem:

S1, S2 = Schiebeschalter mit Umschaltkontakt (z. B. C&K Components PCM12SMTR)
 K1, K2 = USB-Mini-B-Buchse für Platinenmontage (z. B. On-Shore USB-M26FTR)
 K3 = Stromversorgungsbuchse 1,35 mm (z. B. CUI PJ-014DH-SMT)
 K4 = USB-A-Buchse für Platinenmontage (z. B. FCI 87583-2010BLF)
 TR1 = Transformator 3:1, 25 μ H (z. B. Würth 750310471)
 Gehäuse Hammond 1593KBK
 2 selbstschneidende Kreuzschlitzschrauben 4 - 6,4 mm, DIN7981
 Platine 120291-1 oder:
 Aufgebaute und getestete Platine einschließlich Gehäuse 120291-91

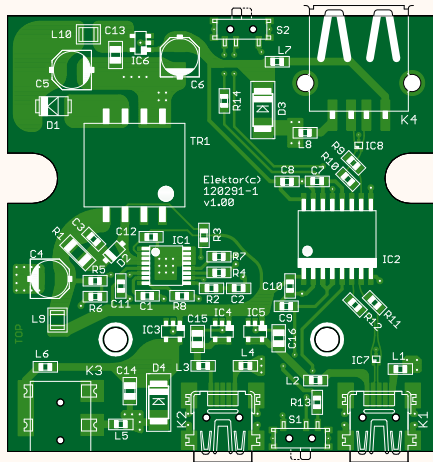


Bild 3. Die Platine ist so bemessen, dass sie in das Gehäuse 1593KBK von Hammond passt.



Eine bessere Lösung ist der Einsatz eines Bausteins, der nicht nur beide Isolator-Seiten mit Strom versorgt, sondern auch den Strom für das angeschlossene USB-Gerät bereitstellt. Nach USB-2.0-Standard darf der Strombedarf bis 500 mA bei 5 V betragen. Aus der Schaltung des USB-Isolators in **Bild 2** geht hervor, dass wir einen Flyback-Converter LT3575 von Linear Technology ausgesucht haben [2]. In der Schaltung fällt auf, dass der LT3575 keine Rückkopplung über einen Optokoppler benötigt. Der

Flyback-Converter arbeitet im so genannten Boundary-Mode, bei dem die Information über die Ausgangsspannung von der sekundären Induktivität zur primären Induktivität rückgeführt wird.

Wenn der Leistungsschalter im Converter abschaltet, steigt die Spannung auf den Wert

$$V_{FLBK} = (V_{OUT} + V_F + I_{SEC} \cdot ESR) \cdot N_{PS}$$

wobei

V_F = Spannung an Diode D1

I_{SEC} = Strom durch die Sekundärwicklung
 ESR = Gesamte Impedanz der Sekundärseite
 N_{PS} = Übersetzungsverhältnis des Transformators

Diese Beziehung liefert zusammen mit einer Temperaturkompensation genügend Informationen für die Regelung der Ausgangsseite. Allerdings muss der Leistungsschalter für eine Mindestzeit abschalten, so dass dem Regler Zeit bleibt, die Informationen auszuwerten. Der Transformator ist hier

ein vergossener, für die SMD-Montage konstruierter Typ aus dem Sortiment des Herstellers Würth.

Betriebssicherheit

Gegen Überspannungen werden die Datenleitungen auf beiden Seiten durch ein Zweikanal-ESD-Schutz-IC von Texas Instruments geschützt, es trägt die Typbezeichnung TPD2E-USB30A (IC7 und IC8). Dieses IC ist nur 1 mm · 1 mm klein, auf der Platine passt es bequem zwischen die Datenleitungen. Nebenbei bemerkt eignet sich dieses IC auch für den Schutz von Datenleitungen nach USB-Standard 3.0.

So genannte High-Side Power Distribution Switches des Typs NCP380 von ON Semiconductor schützen die Betriebsspannungsleitungen vor unverträglichen Strömen. Diese Bausteine arbeiten mit einer aus MOSFETs aufgebauten Fühler- und Reglerschaltung, die den Ausgangsstrom bei einem vorgegebenen Wert begrenzt. In der Schaltung werden zwei Varianten eingesetzt: IC4, IC5 und IC6 begrenzen den Strom bei 500 mA, während IC3 Ströme bis 1 A zulässt.

Die getroffenen Maßnahmen schützen die Eingänge und Ausgänge gegen nahezu alle denkbaren Fehlerzustände. Die Ströme der USB-Port-Betriebsspannungsleitungen werden bei 500 mA gekappt, und der Strom, der einem externen Netzteil entnommen wird, kann 1000 mA nicht übersteigen.

Ein angeschlossenes USB-Gerät darf Ströme bis 500 mA entnehmen. Eigentlich ist dies nicht absolut konform mit dem offiziellen USB-Standard 2.0, nach dem die Grenze zunächst bei 100 mA liegt. Erst wenn das angeschlossene Gerät einen höheren Strombedarf am USB-Bus anmeldet, stehen ihm bis 500 mA zur Verfügung. In der Praxis hat sich gezeigt, dass eine solche Anmeldung meistens nicht notwendig ist.

Die Welligkeit und Störsignalüberlagerung auf den Betriebsspannungsleitungen werden durch die LC-Filter L9/C3 und L10/C13 niedrig gehalten. Durch C4 und C5 fließen vergleichsweise hohe Wechselströme, was bei der Wahl des Kondensator-Typs zu berücksichtigen ist. Elkos mit möglichst niedrigem ESR sind hier sicher keine Fehl-

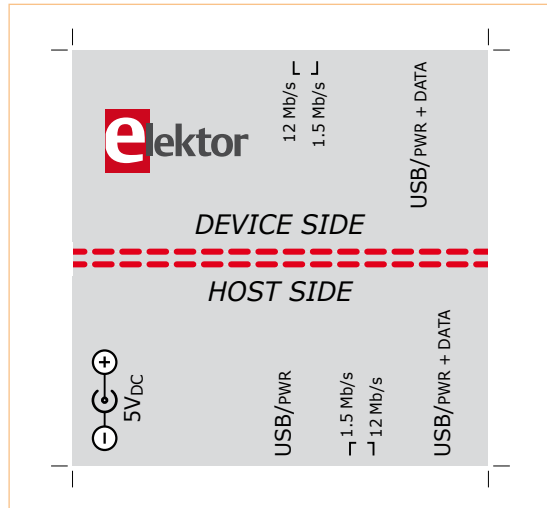


Bild 4. Dieser Aufkleber wird auf ein Klebe-Etikett kopiert und auf dem Gehäuse angebracht.

vestition. Damit die Abschaltspitzen auf der Primärseite der Stromversorgung im ungefährlichen Bereich bleiben, ist dort die Kombination R1, C3, D2 eingefügt.

Aufbau

Schon wegen der hochspeziellen Bauelemente ist für den Aufbau des USB-Isolators eine wohldurchdachte Platine eine unabdingbare Notwendigkeit. Bild 3 zeigt die Platine, die vom Elektor-Labor entworfen wurde. Diese Platine ist auch unbestückt im Elektor-Shop erhältlich. Allerdings sind manche Komponenten wegen ihrer winzigen Abmessungen nur mühsam, wenn nicht sogar schwierig montierbar. Andere Bauelemente gehören zu den Exoten, die nicht jeder Händler im Sortiment hat. Deshalb bietet der Elektor-Shop eine bestückte und getestete Version einschließlich des dazu passenden Gehäuses an. Was auf der Platine noch fehlt, sind einige Aussparungen für die Steckverbinder und die Schiebeschalter. Um die mechanische Stabilität zu erhöhen, werden die Lötflächen der USB-Buchse K4 mit herkömmlichem Lötgerät an die Platine gelötet. Danach kann die Platine mit den beiliegenden Schrauben in das Gehäuse eingebaut werden.

In Bild 4 ist ein Aufkleber abgebildet, der auf ein Klebe-Etikett kopiert - seinen Platz auf der Gehäuse-Oberseite hat. Die Abbil-

dung des Aufklebers steht auch auf der Projektseite im Internet [3] zum Download bereit.

Einsatz

Der Einsatz des USB-Isolators in der Praxis ist vollkommen unkompliziert. Zuerst wird mit den Schaltern S1 und S2 die Übertragungsgeschwindigkeit der USB-Verbindung eingestellt (1,5 Mbit/s oder 12 Mbit/s). Wichtig ist, dass immer auf beiden Seiten die gleiche Geschwindigkeit gewählt werden muss. Normalerweise signalisiert ein USB-Gerät, mit welcher Geschwindigkeit der Datenaustausch stattfinden muss, indem es eine Datenleitung über einen Widerstand auf die Betriebsspannung zieht. Weil dies beim USB-Isolator prinzipiell nicht möglich ist, gibt es für das Einstellen von Hand keine Alternative.

Schließen Sie den „Host“ (meistens der PC) an USB-Buchse K1 an (HOST SIDE: USB/PWR+DATA), verbinden Sie das zu isolierende Gerät mit USB-Buchse K4 (DEVICE SIDE: USB/PWR+DATA). Wenn das Gerät einen höheren Strombedarf als ungefähr 300 mA hat, stellen Sie eine zweite USB-Verbindung über USB-Buchse K2 her (USB/PWR), die nur zur Unterstützung der Stromversorgung dient. Eine Alternative ist ein stabilisiertes 5-V-Steckernetzteil, das an die Strombuchse K3 angeschlossen wird (5 V_{DC}).

Wir empfehlen dringend, den USB-Isolator nicht dort einzusetzen, wo zwischen der Eingangs- und Ausgangsseite Spannungsunterschiede auftreten können, die höher als die Netzspannung sind. Grundsätzlich sollen hohe Spannungsunterschiede nur kurzzeitig anliegen. Ferner ist zu berücksichtigen, dass sich die Isolation langfristig durch Alterungserscheinungen verschlechtern kann.

(120291)gd

Weblinks

- [1] www.analog.com/static/imported-files/data_sheets/ADuM3160.pdf
- [2] <http://cds.linear.com/docs/Datasheet/3575f.pdf>
- [3] www.elektor.de/120291

Zurück zu den Wurzeln (7)

Blinker und Oszillatoren

Im letzten Teil haben wir mit statischen Flipflops und mit Schmitt-Triggern gearbeitet. Jetzt wird es dynamischer, da Kondensatoren für die nötige Rückkopplung eingesetzt werden. Nun fangen LEDs an, zu blinken und zu blitzen. Und Signalgeneratoren sorgen für die akustische Untermalung. Denn Elektronik ist, wenn es blinkt und piepst!

Von Burkhard Kainka (D)

Eine bistabile Schaltung, die selbständig und ohne einen äußeren Anstoß laufend ihren Zustand wechselt, nennt man einen (astabilen) Multivibrator. Einen solchen Multivibrator sieht man in **Bild 1**, und man erkennt sofort, dass hier Kondensatoren für die Rückkopplung zuständig sind. Wenn Elkos verwendet werden, muss man die Polung beachten, da die Spannung am jeweiligen Kollektor im Mittel höher ist als an der gegenüberliegenden Basis. Der Zustand der Schaltung bleibt immer nur so lange stabil, wie die Kondensatoren umgeladen werden. Danach kippt die Schaltung in den jeweils anderen Zustand.

Im praktischen Versuch mit zwei Elkos von 10 μF ergibt sich eine geringe Blinkfrequenz der LEDs mit einer Periodendauer von etwa einer Sekunde. Wir können die Schaltfrequenz des Multivibrators in weiten Grenzen verändern, indem wir die Kondensatoren austauschen. Auch Versuche mit kleineren und mit ungleichen Kondensatoren sind sinnvoll. Mit 100 μF und 100 nF kommt es nur noch zu kurzen Lichtblitzen an einer der beiden LEDs. Mit zwei Kondensatoren von 100 nF ergibt sich ein schnelles Flackern.

Vereinfachter Multivibrator

Der Multivibrator lässt sich vereinfachen, sodass nur noch ein Kondensator nötig ist. Grundsätzlich benötigt die Schaltung zwei Transistoren in Emitterschaltung (welche als Inverter jeweils die Phase des Eingangssignals um 180 Grad drehen). Die Stufen können direkt gekoppelt werden, sodass einer der Kondensatoren entfällt (siehe **Bild 2**). Die notwendige Bedingung für ein sicheres Anschwingen der Schaltung ist ein mittlerer Arbeitspunkt, wenn man die Rückkopplung weglässt. In anderen Fällen wäre der Ausgangstransistor entweder ganz gesperrt

oder voll durchgesteuert. Die Gesamtschaltung hätte dann nicht mehr genügend Verstärkungspotential zum Aufbau von Schwingungen. Hier sorgt eine starke Gegenkopplung am ersten Transistor für einen mittleren Arbeitspunkt (der 10k-Widerstand zwischen Kollektor und Basis). Die Rückkopplung über ein RC-Glied ist aber stärker als die Gegenkopplung und führt im Endeffekt dazu, dass der Ausgangstransistor abwechselnd gesperrt und voll durchgeschaltet wird.

Wir bauen die Schaltung am besten erst mal ohne den Rückkopplungskondensator auf. Die LED sollte schwach leuchten, da der Ausgangstransistor nicht ganz durchgesteuert ist. Mit eingesetztem Kondensator geht die LED abwechselnd ganz an und ganz aus. Mit dem Kondensator von 22 μF blinkt die LED etwa einmal pro Sekunde. Die Schaltung funktioniert auch mit kleineren Kondensatoren bis herunter zu 10 nF. Das Blinken geht in ein schnelles Flackern über. Ein zusätzlich angeschlossener Schallwandler erzeugt ein hörbares Knattern.

LED-Spannungswandler

Eine rote LED braucht 1,5 bis 2 V, eine blaue oder weiße sogar 3 bis 4 V. Meist nimmt man dafür drei Batteriezellen mit zusammen 4,5 V. Die überschüssige Spannung wird in einem Vorwiderstand vernichtet. Besser wäre es, mit 1,5 V auszukommen. Also muss ein Spannungswandler her!

Die entscheidende Zutat ist eine kleine Festinduktivität mit 1,5 Millihenry. Das Bauteil sieht aus wie ein Widerstand. Unter dem Schutzlack befindet sich ein kleiner Ferritkern mit einer Drahtspule. Im Prinzip kann man sich eine passende Spule auch selbst herstellen; etwa 200 Windungen auf einem Ferritstab reichen aus.

Die Schaltung in **Bild 3** zeigt wieder einen einfachen Multivibrator. Der Strom durch die Spule wird schnell ein- und ausgeschaltet. Die Spule arbeitet dabei als magnetischer Energiespeicher. Bei jedem Ausschalten entsteht eine Induktionsspannung, die sich zur Batteriespannung addiert. Die Höhe der Spannung richtet sich nach dem angeschlossenen Verbraucher. Sie passt sich selbst an, sodass z.B. eine weiße LED

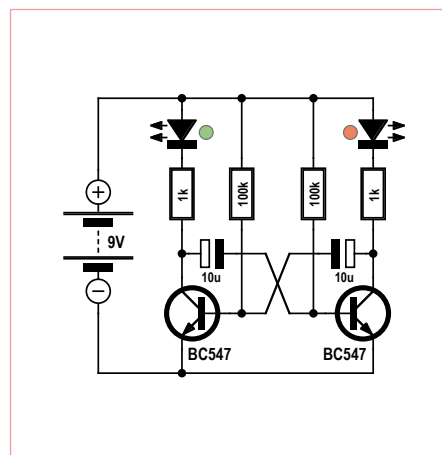


Bild 1. Der Multivibrator.

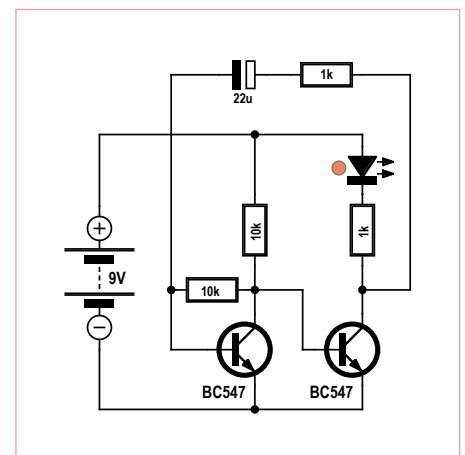


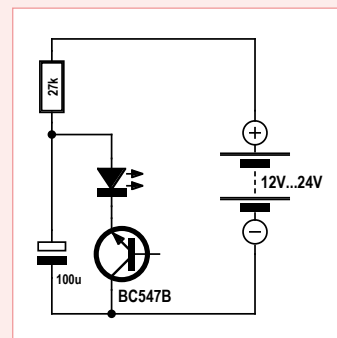
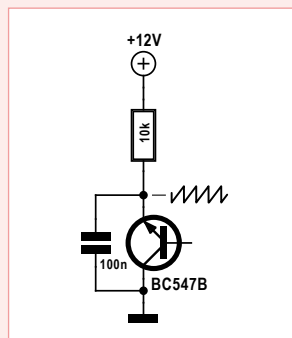
Bild 2. Vereinfachter Blinker.

NPN-Kippschwingungen

So ein Kippgenerator funktioniert auch mit einem einzelnen Transistor, der hier auf sehr ungewöhnliche Weise eingesetzt wird. Der NPN-Transistor sitzt „falsch herum“ in der Schaltung und bekommt eine positive Spannung am Emmitter. Außerdem bleibt der Basisanschluss frei. Man beobachtet, dass die Spannung am Kondensator jeweils bis ca. 9 V ansteigt. Dann wird der Transistor schlagartig leitend und entlädt den Kondensator bis ca. 7 V. Im Datenblatt des Transistors steht nichts davon. Jeder Transistor verhält sich dabei etwas anders. Es lohnt sich, mehrere Transistoren auszuprobieren.

Der Entladestromstoß ist kräftig genug, um eine LED zu betreiben. Hier wird eine Versorgungsspannung von über 12 V gebraucht. Die Schaltung funktioniert sehr gut mit zwei fast völlig leeren 9-V-Batterien. Die LED blinkt noch lange und holt das letzte bisschen Saft aus den Batterien. Die Blitzfrequenz nimmt dabei ab.

Der Transistor hat im inversen Betrieb zwischen Emmitter und Kollektor eine Kennlinie mit negativer Steigung, was sich leicht nachmessen lässt. Die BE-Diode zeigt bei ca. 9 V den bekannten Avalanche-Effekt. Dabei werden Ladungsträger wegen der hohen elektrischen Feldstärke in der sehr dünnen Sperrschicht so schnell, dass sie weitere Ladungsträger aus dem Gitter befreien können. Die Anzahl der Ladungsträger steigt lawinenartig an, und damit der Strom. Dieser



Effekt entspricht genau dem in einer 9-V-Z-Diode. Der Innenwiderstand dieser Diode ist aber positiv.

Doch jetzt kommt zusätzlich der inverse Transistor dazu. Emmitter und Kollektor tauschen zwar die Rollen, aber wegen des prinzipiell symmetrischen Aufbaus funktioniert der Transistor auch so herum. Die Funktion des Transistors ist ja, dass Ladungsträger durch die dünne Basis hindurch in die Sperrschicht gelangen, dieselbe Sperrschicht, in der hier auch der Lawineneffekt auftritt. Also noch mehr Ladungsträger, die noch weitere Ladungsträger aus dem Gitter befreien. Damit entsteht ein sich selbst verstärkender Lawineneffekt.

mehr Spannung erhält als eine rote. Meist verwenden solche Spannungswandler noch einen Gleichrichter und einen Siebelko. Hier kann man darauf verzichten, denn die LED ist ihr eigener Gleichrichter. Durch sie fließt pulsierender Gleichstrom. Er ist im Mittel etwas kleiner als der Batteriestrom, weil die Spannung höher ist. Insgesamt hat die Schaltung einen besseren Wirkungsgrad als die übliche Lösung mit der höheren Batteriespannung und dem Vorwiderstand.

Tongeneratoren

Wenn wir den einfachen Multivibrator mit einem kleineren Kondensator ausstatten,

erhalten wir eine höhere Frequenz, die in den Tonfrequenzbereich reicht. Ein angeschlossener Piezo-Schallwandler erzeugt hörbare Töne (**Bild 4**). Der Piezowandler ist selbst so etwas wie ein Kondensator und beeinflusst die Tonfrequenz. Ein kleines Experiment am Rande: Wir berühren den Wandler mit dem Finger oder mit einem harten Gegenstand. Dabei ändert sich nicht nur die Lautstärke, sondern auch die Tonhöhe. Die schwingende Piezoscheibe erzeugt ihrerseits eine Wechselfspannung, die den Generator beeinflusst. In geringem Maße kann sogar reflektierter Schall einen Einfluss haben.

Die Frequenz lässt sich zusätzlich auch durch den Widerstand im Rückkopplungszweig beeinflussen. Hier könnte man ein Poti einsetzen. Es darf aber auch ein Fotowiderstand (LDR) sein. Die Tonhöhe wird nun durch das dort auftreffende Licht bestimmt. Mit dieser Schaltung (**Bild 5**) können wir nicht nur die Helligkeit, sondern auch die Art des Lichts unterscheiden. Stark flackern des Kunstlicht moduliert die Frequenz des Tonsignals. Das Licht einer Leuchtstoffröhre erzeugt einen schnarrenden Ton. Ebenso verändert das Licht eines PC-Monitors den Klang durch eine Tonmodulation mit der Bildwiederholfrequenz.

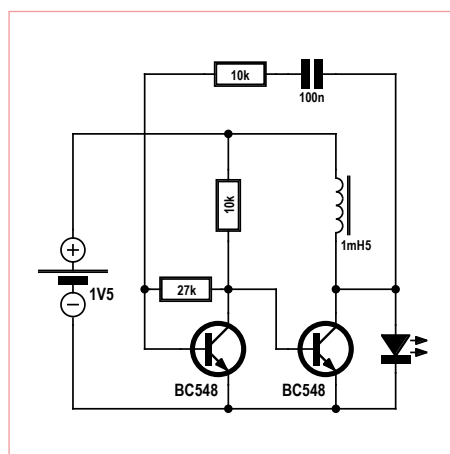


Bild 3. Ein LED-Spannungswandler.

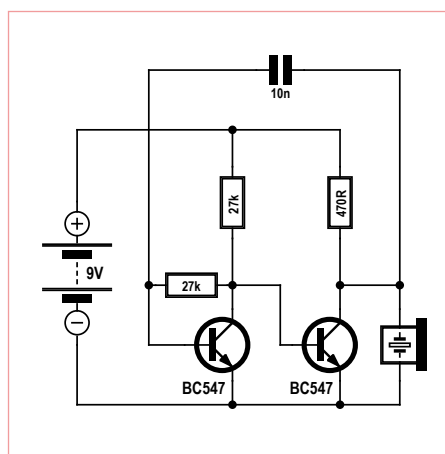


Bild 4. Ansteuerung eines Piezo-Schallwandlers.

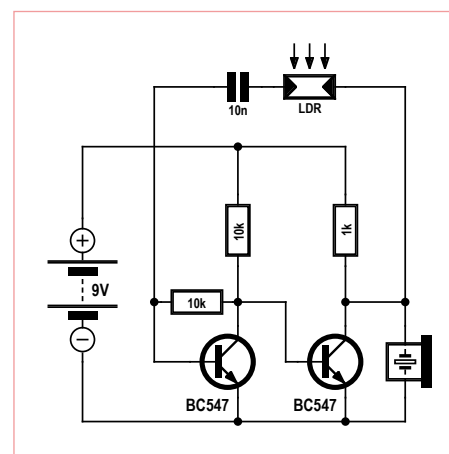


Bild 5. Ein einstellbarer Tongenerator.

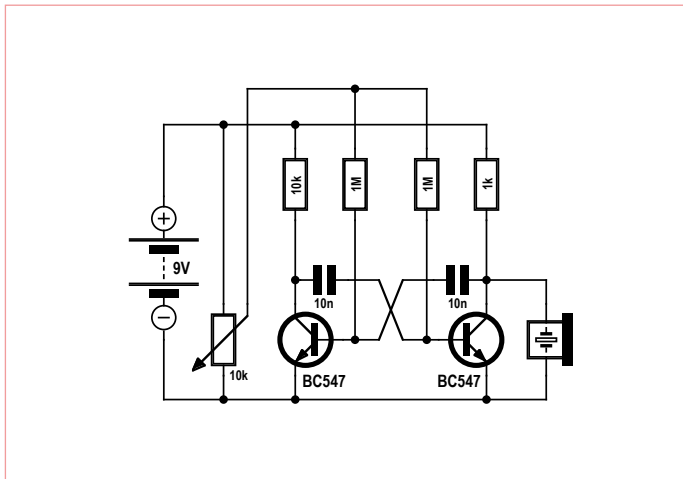


Bild 6. Ein Multivibrator als VCO.

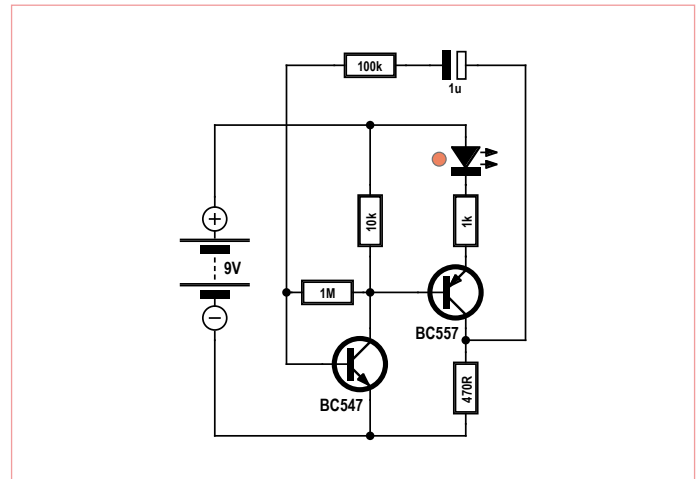


Bild 7. Eine komplementäre Blinkschaltung.

Spannungs-Frequenz-Wandler

Ein spannungsabhängiger Oszillator (Voltage Controlled Oszillator, VCO) kann mit einem Multivibrator nach **Bild 6** aufgebaut werden. Beide Basiswiderstände sind mit demselben Spannungseingang verbunden. Je höher die anliegende Spannung ist, desto größer wird der Ladestrom und damit die Frequenz des Oszillators. Ein einstellbarer Spannungsteiler erlaubt die Vorgabe einer beliebigen Spannung und damit die Einstellung der Frequenz. Die Schaltung kann aber auch als akustisches Voltmeter mit dem Messbereich 1 V bis 9 V verwendet werden.

Man kann so z.B. schnell unterschiedliche Batterien testen.

NPN/PNP-Kippschaltung

Es müssen nicht immer zwei NPN-Transistoren sein. **Bild 7** zeigt einen Blinker mit komplementären Transistoren. Ein stabiler Arbeitspunkt ergibt sich wie oben durch die Kollektor-Basis-Gegenkopplung des NPN-Transistors. Der PNP-Transistor arbeitet als Emitterfolger. In der Kollektorleitung liegt ein Hilfswiderstand, an dem ein phasenrichtiges Signal für die Rückkopplung abgenommen werden kann.

Der Rückkopplungsweig der Schaltung ist sehr hochohmig. Schon mit einem Rückkopplungskondensator von nur 1 μF ergibt sich daher eine Periode von rund einer Sekunde.

Energiesparendes LED-Blitzlicht

In Geschäften sieht man manchmal Werbeschilder mit einer blinkenden LED, die anscheinend ewig arbeiten, und das auch noch mit einer einzelnen Batteriezelle. Die Schaltung nach **Bild 8** besteht aus einem instabilen Multivibrator mit besonderen Eigenschaften. Ein Elko von 100 μF lädt sich

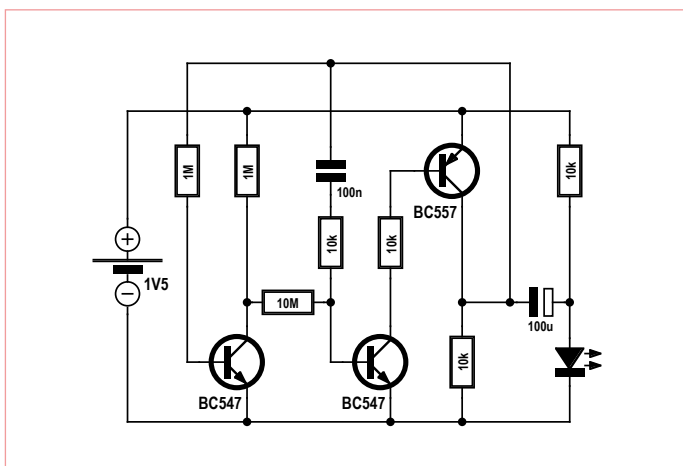


Bild 8. Sparsamer LED-Blitzer.

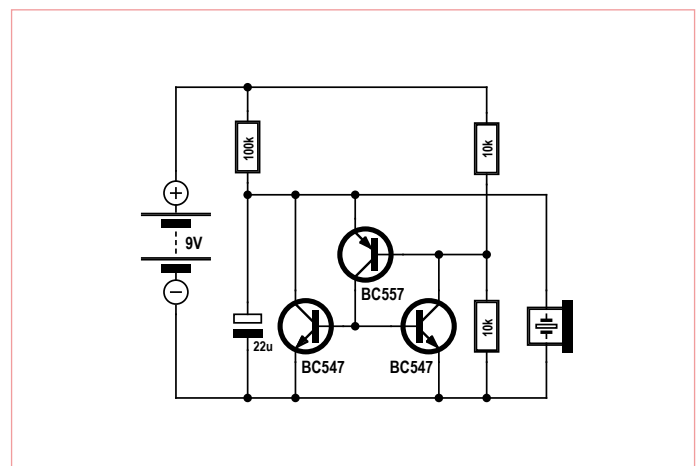
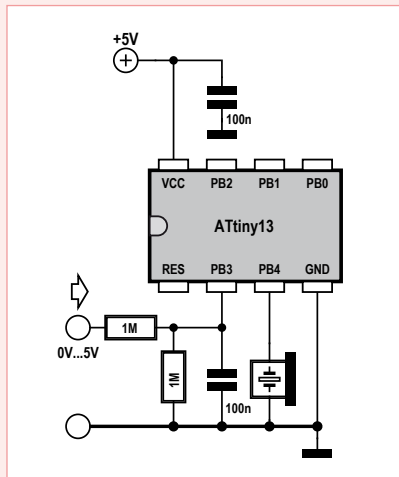


Bild 9. Ein Kippgenerator.

U/F-Wandler mit dem Tiny13

So ein Spannungs-Frequenz-Wandler ist eine feine Sache, denn er kann als akustisches Voltmeter eingesetzt werden. Kammerton A könnte bedeuten, dass die Batteriespannung in Ordnung ist. Auch langsame Änderungen sind für das Ohr gut zu erkennen. Mit dem Mikrocontroller können wir eine Lösung realisieren, bei der die Frequenz linear zur Eingangsspannung ist.

Der Mikrocontroller ATtiny13 wird hier an 5 V betrieben. Damit reicht der Messbereich des A/D-Wandlers ebenfalls bis 5 V. Mit einem hochohmigen Spannungsteiler wird er auf 10 V erweitert. Am Port B4 ist ein Piezo-Schallwandler angeschlossen. Das Programm realisiert einen einfachen DDS-Generator (Direkte digitale Synthese) mit Rechteckausgang. Die Messwerte erhöhen einen Akkumulator A immer wieder, bis das höchstwertige Bit umkippt; daraufhin schalten wir den digitalen Ausgang um. Der Akku hat hier eine Breite von 12 bit. Bei der höchsten Spannung dauert es jeweils vier Messperioden, bis sich der Ausgangszustand ändert. Dabei kommt



man auf knapp 600 Hz. Den Quellcode kann man unter www.elektor.de/120007 downloaden.

```
'U/f Converter  0...5 V 0...600 Hz
$regfile = "attiny13.dat"
$crystal = 1200000
$hwstack = 8
$swstack = 4
$framesize = 4
```

```
Dim U As Word
Dim A As Word
```

```
Config Adc = Single , Prescaler = Auto
Start Adc
Ddrb.4 = 1
```

```
Do
  U = Getadc(3)
  A = A + U
  A = A And &H0FFF
  If A >= &H0800 Then
    Portb.4 = 1
  Else
    Portb.4 = 0
  End If
Loop

End
```

relativ langsam und mit kleinem Strom auf und wird in einem kurzen Impuls über die LED entladen. Dabei kommt es auch zur nötigen Spannungsüberhöhung, denn 1,5 V ist ja für eine LED zu wenig.

Die Schaltung wurde für den Low-Power-Betrieb optimiert. Deshalb arbeitet der eigentliche Multivibrator mit einem NPN- und einem PNP-Transistor. Man kann auf diese Weise die Verschwendung von Steuerstrom vermeiden. Beide Transistoren leiten nur für den kurzen Moment des LED-Blinsens. Damit stabile Bedingungen herrschen und die Schaltung sicher schwingt, gibt es eine zusätzliche Stufe mit einer Gleichspannungs-Gegenkopplung. Auch hier wurde auf besonders hochohmige Widerstände und geringsten Verbrauch geachtet.

Der PNP-Transistor leitet alle paar Sekunden nur für einen kurzen Moment. In der

Zwischenzeit wird der Ausgangselko auf fast 1,5 V aufgeladen. In der aktiven Phase addiert sich die Elkospannung zur Batteriespannung. Im Leerlauf entstehen daher Impulse mit fast 3 V. Eine rote oder grüne LED mit einer Durchlassspannung von 1,8 V bis 2 V blitzt hell auf.

Für die Abschätzung des Verbrauchs kann man vom Ladestrom des Elkos ausgehen. An beiden Ladewiderständen mit je 10 kΩ liegt im Schnitt eine Spannung von insgesamt 1 V. Damit beträgt der durchschnittliche Ladestrom 50 µA. Für die Dauer des LED-Impulses wird noch einmal genau die gleiche Ladung aus der Batterie entnommen. Der mittlere Strom beträgt also rund 100 µA. Geht man von einer Batteriekapazität von 2000 mAh aus, sollte die Batterie etwa 20.000 Stunden halten, das sind über zwei Jahre!

Kippschwingungen

Kippschwingungen mit ihrer typischen Sägezahnform entstehen, wenn ein Kondensator periodisch bis zu einer bestimmten Spannung geladen wird und sich dann schlagartig entlädt. Man schaue sich die Schaltung in **Bild 9** an. Solange der Kondensator noch geladen wird, bleibt der PNP-Transistor gesperrt, und auch die beiden NPN-Transistoren erhalten keinen Basisstrom. Die Kippgrenze wird durch den Spannungsteiler aus zwei 10-kΩ-Widerständen auf rund 4,5 V + 0,6 V festgelegt (ab dann liegt die Spannung an der Basis 0,6 V niedriger als am Emitter). Ab rund 5,1 V fließt also ein Strom, der sich durch Rückkopplung zu einem kräftigen Entladestrom verstärkt. Die Spannung sinkt dabei bis auf 0,6 V. Dann sperren die Transistoren, und der nächste Ladevorgang

Quiz

Ein einfacher spannungsgesteuerter Multivibrator soll als akustisches Voltmeter mit einem 8-Ohm-Lautsprecher verwendet werden. Bei einer Betriebsspannung von 9 V und einer Messspannung von ebenfalls 9 V wurde die Schaltung mit dem Oszilloskop untersucht.

Wenn einer der Transistoren im Multivibrator gerade leitend wird, sinkt die Basisspannung des anderen Transistors bis auf -9 V. Der 10-nF-Kondensator wird dann über 100 k Ω von der positiven Messspannung aus in ca. 0,65 ms bis etwa +0,6 V geladen, sodass der Zustand der Schaltung erneut umkippt.

1) Welche Frequenz ergibt sich bei einer Messspannung von +9 V?

- A) ca. 3,3 kHz
- B) ca. 330 Hz
- C) ca. 770 Hz

2) Wie ändert sich die Frequenz, wenn die Betriebsspannung absinkt, die Messspannung aber gleich bleibt?

- D) Die Frequenz sinkt.
- E) Die Frequenz bleibt konstant.
- F) Die Frequenz steigt.

3) Der Ton soll lauter werden. Was kann verändert werden?

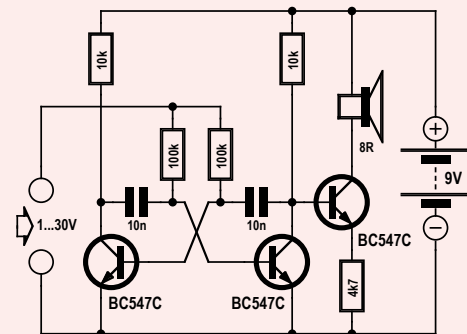
- G) Der Emittterwiderstand 4,7 k Ω des rechten Transistors wird verkleinert.
- H) Der Emittterwiderstand wird vergrößert.
- I) Der Emittterwiderstand wird durch einen Elko mit 100 μ F ersetzt.

Wer uns die richtige Lösung zusendet, kann ein „Minty Geek Electronic 101 Kit“ gewinnen!

Senden Sie dazu den Lösungscode (ergibt sich durch Aneinanderreihung der Buchstaben der drei richtigen Antworten) per E-Mail an: basics@elektor.com.

Als Betreff der E-Mail bitte nur den Lösungscode angeben.

Einsendeschluss ist der 30.09.2012.



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

Der richtige Lösungscode aus dem Maiheft lautet „ADH“.

Hier die Erklärung:

Lösung 1:

Der BF245B stabilisiert etwa 10 mA, deshalb ist A) richtig.

Lösung 2:

Bei großer Eingangsspannung würde der Strom mit einem Vorwiderstand stark ansteigen, was zu hohen Wärmeverlusten führt. Der FET dagegen hält den Strom konstant, sodass die Verluste geringer bleiben. D) ist korrekt.

Lösung 3:

Der Elko stützt die Spannung bei schnellen Änderungen des Laststroms und verbessert die Stabilisierung für hohe Frequenzen. Bei 1 kHz hat der Elko z.B. einen kapazitiven Widerstand von 1,6 Ω und verringert damit den Innenwiderstand der Stabilisierungsspannung. Er trägt für eine Millisekunde noch merklich zur Aufrechterhaltung der Spannung bei, aber nicht viel länger. H) ist korrekt.

beginnt. Die Schaltung nach Bild 9 zeigt eine Version mit drei Transistoren und für eine sehr langsame Ladung. Das Ergebnis ist eine Art Metronom: Tick, tick, tick ... Man kann die Schaltung etwas vereinfachen und den linken Transistor weglassen. In vielen Fällen funktioniert dann alles wie gehabt. Allerdings gibt es manchmal

Schwierigkeiten mit dem Löschen des selbst gebauten NPN-PNP-„Thyristors“ (siehe den letzten Teil der Serie). Die Schaltung kann bei kleinem Ladestrom im leitenden Zustand hängen bleiben. Bei der Drei-Transistor-Variante passiert das nicht, die Schaltung arbeitet zuverlässig in einem weiten Bereich von Ladeströmen.

Noch eine Vereinfachung ist möglich. Da der Piezo-Schallwandler ja selbst schon ein Kondensator ist, kann man den Elko sparen. Aus dem langsamen Taktgenerator wird dann ein schneller Tongenerator!

(120007-1)

ARM-Mikrocontroller

➔ 35 Einsteiger-Projekte in C mit dem mbed-Board

Die Projekte in diesem Buch sind für Einsteiger in C und ARM-Mikrocontroller ausgelegt. Das heißt nicht, dass diese Projekte einfach sind. Sie sind aber einfach zu verstehen. Es wird beispielsweise die USB-Verbindung zur Kommunikation benutzt, eine Methode, die im mbed-Board so einfach integriert ist, dass sie sich auch für ein Einsteiger-Buch eignet. Der mbed NXP LPC1768 nutzt Cloud-Technologie, ein revolutionäres Konzept in der Software-Entwicklung. Es bedeutet, dass man keinerlei Software auf seinem PC installieren muss, um den mbed zu programmieren. Das Einzige, was Sie brauchen, ist ein Webbrowser mit Internetzugang und einen freien USB-Anschluss an Ihrem PC. Sie können von jedem beliebigen Ort der Welt auf Ihr Projekt zugreifen und daran weiterarbeiten. Wenn Sie fertig sind, genügen ein paar einfache Mausklicks, um Ihr Programm auf das mbed-System zu übertragen. Natürlich können Sie die Projekte auch auf Ihren eigenen PC laden und dort speichern. Die Quelltexte zu den Beispielprogrammen stehen gratis unter www.elektor.de/arm-buch zum Download bereit. Das zum Buch gehörige Hardware-Starterkit kann unter www.elektor.de/arm-kit geordert werden.



NEU!

261 Seiten (kart.)
Format 17 x 23,5 cm
ISBN 978-3-89576-262-8
€ 39,80 • CHF 49,40

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

Android

➔ Apps programmieren – Schritt für Schritt

Smartphones und Tablet-Computer mit dem Betriebssystem Android finden immer weitere Verbreitung. Die Anzahl der Anwendungsprogramme – die sogenannten Applikationen oder kurz Apps – mit denen sich die Geräte individuell an die Vorlieben und Wünsche ihrer Benutzer anpassen lassen, steigt täglich an. Die ständig besser werdende technische Ausstattung der Geräte erlaubt inzwischen Anwendungen, die vor wenigen Jahren nur Desktop-Computern oder spezieller Hardware vorbehalten waren. Mittlerweile kann jedes Smartphone seine Position bestimmen, Videos aufnehmen und vieles andere. Man ist bei der Individualisierung seines Smartphones aber nicht auf fix und fertige Applikationen beschränkt. Es ist einfacher als man denkt, Android-Geräte selber zu programmieren und eigene Apps zu schreiben. Dieses Buch bietet eine Einführung in die Programmierung von Apps auf Android-Geräten. Es erklärt leicht nachvollziehbar die Funktionsweise des Android-Systems und Schritt für Schritt die Programmierung von Applikationen. Anhand vieler Beispiele werden die verschiedensten Anwendungsmöglichkeiten gezeigt. Das reicht von einfachen Rechenprogrammen über die Abfrage von Sensoren und GPS-Daten bis zur Programmierung von Applikationen für die Internet-Kommunikation. Neben der Erstellung von Applikationen in der Programmiersprache JAVA wird auch gezeigt, wie Apps mit Hilfe von Javascript oder PHP-Skripten programmiert werden können.



NEU!

256 Seiten (kart.)
Format 17 x 23,5 cm
ISBN 978-3-89576-252-9
€ 34,80 • CHF 43,20

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

Rechteck-Signalbox

Mit Pulsbreitenmodulation und Berührtastern

Von Peter de Bruijn (NL)

Dieser Low-Cost-Rechteckoszillator überstreicht den weiten Bereich 250 Hz...60 MHz. Zwischen 250 Hz und 60 kHz kann das Rechtecksignal pulsbreitenmoduliert werden (PWM), so dass zahlreiche Objekte getestet werden können. Beispiele sind LED-Dimmer, Ultraschall-Empfänger, Step-up- und Step-down-Wandler oder PWM-gesteuerte Motoren. Im Oszillator-Modus (OSC) arbeitet die Rechteck-Signalbox als klassischer Rechteckgenerator. Hier beginnt der Frequenzbereich bei 1 kHz, die obere Grenze liegt bei 60 MHz. Test-

genügt für die wenigen Bauteile auch ein Stück einer Experimentierplatine. Der Autor hat die Schaltung in einem Kleingehäuse mit bedruckter Frontplatte untergebracht. Die Berührtaster bestehen aus Kupferinseln der Größe 1 cm · 1 cm, sie wurden unter der Frontplatte mit Sekundenkleber auf das Gehäuse geklebt. Von der Projektseite [1] kann außer dem Source- und Hexcode für den Mikrocontroller auch eine Kurzanleitung heruntergeladen werden.

(120111)gd

[1] www.elektor.de/120111

Low-Cost-Rechteckgenerator mit eindrucksvollem Bereich

Eigenschaften

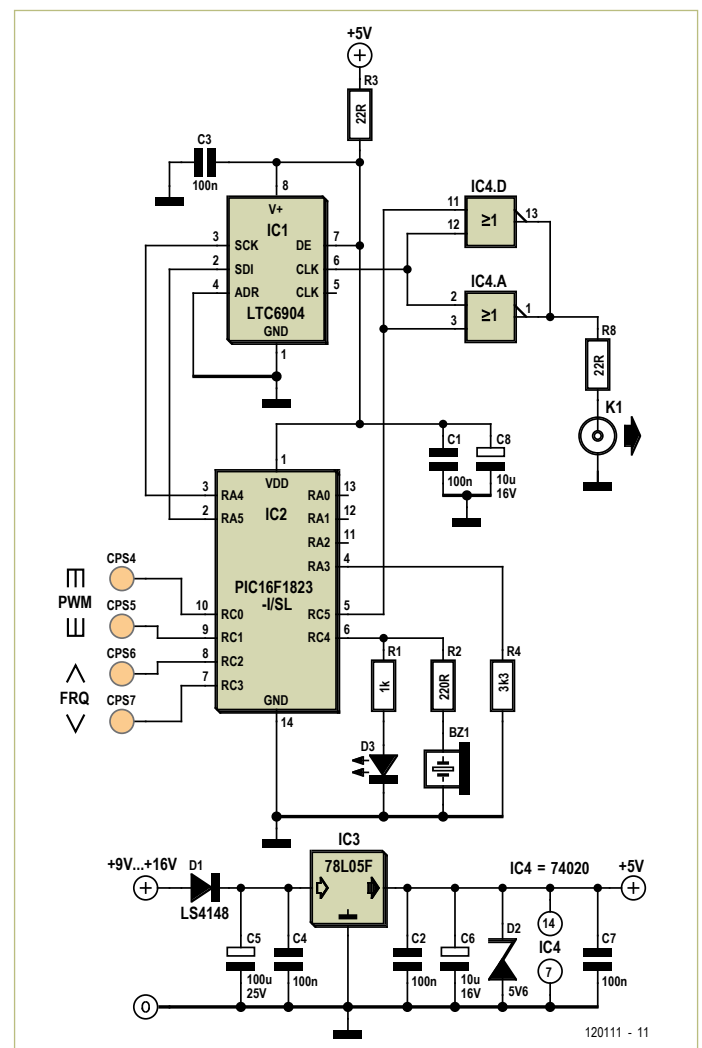
- Startfrequenz: 10 kHz bei 5 V (OSC-Modus)
- Mode 1 PWM: 250 Hz...60 kHz, Drift 1 %, Genauigkeit 1 %
- Mode 2 OSC: 1 kHz...60 MHz, Drift 0,05 %, Genauigkeit 0,75 %

objekte können eine DCF-Antenne (77,5 kHz) oder ein Schwingkreis bis 60 MHz sein. Ein 20-MHz-Taktoszillator für einen Mikroprozessor gehört zu den weiteren denkbaren Funktionen. Die Rechteck-Signalbox wird über vier kapazitive Berührtaster bedient.

Für das Erzeugen der Rechtecksignale sind ein Oszillatorbaustein LTC6904 von Linear Technology und ein Mikrocontroller PIC16F1823 aus der PIC-Familie von Microchip zuständig. Der PIC steuert den LTC6904 über die Leitungen RA4 und RA5. Die kapazitiven Berührtaster sind unmittelbar mit den Eingängen RC0...RC3 verbunden, zusätzliche Hardware ist nicht erforderlich. Beim Berühren gibt der Piezo-Beeper an RC4 ein akustisches Signal ab, während die parallele LED D1 das Berühren visuell anzeigt. Hinter dem Ausgang des LTC6904 folgen zwei parallele EXOR-Gatter, so dass das Ausgangssignal doppelt belastbar ist. Das von Anschluss RC5 des Mikrocontrollers kommende Signal steuert die Pulsbreite.

Die Betriebsspannung wird vom Spannungsregler IC3 stabilisiert, wobei R3, C7, C1 und C8 für die Entkopplung sorgen. Die Zenerdiode D2 schützt den Oszillator-Baustein und den Mikrocontroller, falls die Eingangsspannung des Spannungsreglers versehentlich den zulässigen Wert übersteigt. Ein Steckernetzteil kann die Eingangsspannung liefern (9...16 V).

Wer eine Platine entwerfen will, muss die Leitungen der Berührtaster durch Masseflächen vom Oszillator abschirmen. Natürlich



LABCENTER

GPIO beim Elektor-Linux-Board

Von François-Xavier Maurille (Elektor-Labor)

Bei der Entwicklung eines Musik-Projekts auf Basis des Elektor-Linux-Boards wurde ein DAC per SPI angesprochen. Dieser benötigte Worte in 24 bit Breite – doch das SPI-Hardware Interface des Mikrocontrollers war nicht in der Lage, 24 Bits zu senden und dabei alle SPI-Pins passend zu bedienen. Folglich wurde der CS-Ausgang (chip select) mit einem GPIO-Pin realisiert, um ihn per Software umschalten zu können. Der GPIO-Pin war allerdings nicht ganz so einfach mit dem OS anzusprechen, denn die Reaktionszeit auf einen Befehl war aufgrund des komplexen und intensiven Datenaustauschs zwischen Applikation und Hardware nicht so prickelnd.

Die Diagramme zeigen meine Versuche, die CS-Sequenz „1010“ zu beschleunigen. In meinem ersten Versuch nutzte ich die Funktion `fprintf` C unter Linux. CH2 zeigt das Resultat des Befehls `fprintf(„/sys/class/gpio/gpio11/value“, „%d“, state)`. Die minimale Antwortzeit von 90 µs war zwar besser, aber immer noch zu langsam für meine Applikation.

Als nächster Versuch wurde der Pin per `echo system command` angesprochen. CH1 zeigt das Ausgangssignal nach dem Befehl `System(„echo 1 > /sys/class/gpio/gpio11/value“)`. Keine große Beschleunigung, sondern faktisch ein Rückschritt gegen-

über dem ersten Versuch: Es vergingen sogar 200 ms, bis der CS-Ausgang den Pegel wechselte.

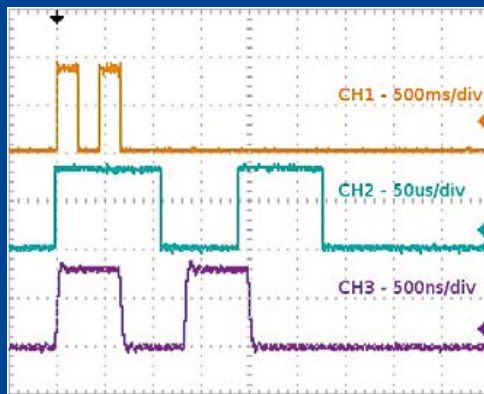
Schließlich sprach ich die GPIO-Register des Mikrocontrollers direkt an. Das ist zwar schwieriger als die Verwendung von OS-Routinen, aber es erwies sich als viel schnelleres Verfahren. Dank der Hilfen in <http://forum.gnublin.org> war es recht einfach, die Funktionen zum Setzen und Rücksetzen von IO-Pins mit Hilfe eines Pointers und Offsets zur IOCONFIG-Register-Adresse (von `mmap()`) zu implementieren, die auf die MOEX-Register zugriffen. Signal CH3 zeigt das Ergebnis des Befehls `*(unsigned int *) (ptr + GPIO_OFFSET + GPIO_MODE0) = 1 << nGPIO;`. Hier wurde eine Reaktionszeit von etwa 700 ns erzielt, was schon mehr meinen Bedürfnissen entsprach.

Einige Beobachtungen:

- Man beachte: Das 4. MOEX-Bit steht für GPIO4, während das 5. MOEX-Bit für GPIO11 (CS) gedacht ist und eben nicht für GPIO5.

- Mit dem Linux-Board hat man eine kleine Zeitmaschine: Wenn das interne Datum bzw. die Board-Zeit nicht synchron mit der des angeschlossenen PCs ist, erhält man beispielsweise folgende Warnung: „*make: Warning: File 'GPIO.c' hat modification Time 52 s in the future*“.

(120457)



SD-Karten-Korrektur per Script

Von François-Xavier Maurille (Elektor-Labor)

Das mittlerweile recht bekannte Elektor-Linux-Board, dessen Artikelserie mit der Mai-Ausgabe 2012 von Elektor startete, bewährt sich als einfacher Einstieg in die Embedded-Linux-Welt. Auch wenn die Hardware prima funktioniert, scheint es doch hie und da bei der Software zu haken. Wer das Elektor-Linux-Board schon benutzt, wird vielleicht über das Problem gestolpert sein, dass manchmal die SD-Karte defekt zu sein scheint. Dabei erscheint die Meldung „`EXT2-fs (mmcblk0p1): error: ext2_lookup: deleted inode referenced: 694962`“.

In dieser Elektor-Ausgabe ist eine Anleitung zu finden, die erklärt, wie man diesen Fehler beseitigt. Wenn man nicht sicher ist, wie man das macht (oder auch bequemlichkeitshalber), kann man einfach die von mir geschriebene Software verwenden, die das Problem beseitigt. Dabei handelt es sich um ein kleines Bash-Script, das auf einem Linux-PC läuft. Dabei werden exakt die gleichen Befehle ausgeführt, die im Artikel beschrieben sind. Mit einer Ausnahme: Sie laufen automatisch ab. Das Script sucht den Namen des defekten Geräts mit den Befehlen `grep` und `sed`, unmountet das Gerät und startet dann den Befehl `e2fsck`.

Dieses Script repariert die SD-Karte auf direkte Weise. Man muss lediglich den angezeigten Instruktionen Folge leisten und gelegentlich die Eingabetaste oder die Y-Taste drücken. Hier eine Kurzanleitung:

Nach dem Download des Bash-Scripts von [1] wechselt man in das Verzeichnis, in dem die Datei abgelegt wurde und entpackt diese (Rechtsklick und *Extract Here* auswählen). Vor dem ersten Start muss man das Script ausführbar machen. Man starte also das Terminal und wechsele in das Verzeichnis mit dem Script. Dann gebe man die folgende Zeile ins Terminal-Fenster ein: `sudo chmod 777 correctSD.sh`. Jetzt sollte das Bash-Script ausführbar sein. Nun also die Eingabe `./correctSD.sh` und abwarten, bis die defekte SD-Karte wieder repariert ist. Eine beigefügte README-Datei enthält ein paar zusätzliche Informationen zur richtigen Vorgehensweise.

(120443)

[1] www.elektor.de/120026

Elektor-projects.com 4U2

[Home](#) [News](#) [Proposals](#) ➔ [In Progress](#) ➔ [Finished](#)

Home

Project Proposals

This page lists all the new project proposals and project ideas that have not been promoted to **In Progress** yet. Promotion of a project depends on its popularity, originality or usefulness.

Create a new project or submit an idea now!

Get help, feedback & votes from other visitors, and maybe your project will be promoted too!

Before you know
it you too are
Elektorized!





Vote for the projects you like! The more votes a project collects the higher the chances that it gets promoted to **In Progress**.

Das Elektor-Labor reicht nun bis ins Internet und so können **auch Sie** an der Entwicklung der Elektor-Projekte teilnehmen, die es Ihnen besonders angetan haben. Wir wollen Sie dazu einladen, Teil der internationalen Elektor-Community zu sein. Stellen Sie hier Ihre eigenen Projekte vor und diskutieren sie Ihre Ideen. Dort können Sie auch sehen, woran im Elektor-Labor gerade gearbeitet wird. Und wenn Ihr Projekt gute Bewertungen erhält, dann besteht eine gute Chance, dass es zu einem Elektor-Projekt wird!

Neue Ideen, Schaltungen und Skizzen etc. kommen in den Bereich *Proposals*. Die interessantesten und vielversprechendsten werden dann vom Elektor-Labor-Team ausgewählt und weiterentwickelt, bis daraus ein richtiges Elektor-Projekt samt Artikel wird. Und Sie können dann stolz auf sich sein.

Werfen Sie einmal einen Blick auf den Beitrag *Switched 7905 Replacement* in der Rubrik *In Progress*. Dieses Projekt wird voraussichtlich in der Oktober-Ausgabe 2012 von Elektor veröffentlicht. Auf der Webseite erhält man einen ersten Eindruck vom Projekt und seinen Entwicklungsschritten.

Die Rubrik *Finished* enthält abgeschlossene Projekte wie *Platino* oder *Improved Radiation Meter*. Abgeschlossen heißt aber nicht, dass sich hier nichts mehr tut...

(120484)

Hinweis: Elektor-projects.com ist für alle kostenlos verfügbar – im Lesemodus. Als Elektor-Plus-Mitglied erhalten Sie automatisch auch jene Zugriffsrechte, die das Einstellen von Beiträgen ermöglichen. Bei vergessenem Passwort kann man sich jederzeit ein neues an die hinterlegte E-Mail-Adresse schicken lassen. Wer (noch) kein Elektor-Plus-Mitglied ist, aber einen interessanten Beitrag zu bieten hat, kann sich melden unter redaktion@elektor.de.

[Home](#) [News](#) [Proposals](#) ➔ [In Progress](#) ➔ [Finished](#)

Home

Projects in Progress

This page lists all projects currently under active development and not yet (completely) finished. Projects on this list may get support from the Elektor team and even make it to print in **Elektor Magazine** and/or become a ready-made product in the shop. That's right, you can **make money from your project!**

[Home](#) [News](#) [Proposals](#) ➔ [In Progress](#) ➔ [Finished](#)

Home » Projects

Switched 7905 replacement



11
votes

[+1](#) [in](#) [Share](#) [Tweet](#) [Vind ik leuk](#)

Project status: In Progress | 0 contributions | 2 members | 0 comments

After having designed a switched 7805 replacement, my editor suggested that I should make a matching negative variant, a switched 7905 replacement. I knew that converting a positive DC voltage to a negative DC was possible, but I had never heard of a switched negative to negative DC converter. So after a bit of research I found a topology called "Negative Buck Converter" in an old National Semiconductor application note. It abuses an asynchronous boost converter to convert a low negative DC voltage to a higher negative DC voltage. Just like an 7905. And I did manage to keep the PCB dimensions in check, as can be seen in the photo.



It is all a bit experimental, so what the final specs will be is still unknown. But I aim for an input is low as -18V which can still output 0,8A at -5V.

Update:
I assembled it and tested it with a light load and it works!!! -12V in, -5V out. I've still got some tests to do. I will keep you posted.

Update1:
I did some tests with different input voltages, -17V till -8V works but higher than that gives incorrect output voltages.

[Home](#) [News](#) [Proposals](#) ➔ [In Progress](#) ➔ [Finished](#)

Home

Finished Projects

The projects on this page are no longer actively developed because they are either finished or have reached a dead-end. Finished projects may still be updated on occasions when a bug is found, a component has turned obsolete or an improvement was added. Some of these projects have been published in **Elektor Magazine**, some have products for sale in the online **Elektor Shop**.

[Home](#) [News](#) [Proposals](#) ➔ [In Progress](#) ➔ [Finished](#)

Home » Projects

Platino - Versatile Board for AVR Microcontrollers



2
votes

[+1](#) [in](#) [Share](#) [Tweet](#) [Vind ik leuk](#)

Project status: Finished | 0 contributions | 1 members | 0 comments

Behind every great circuit there's a great PCB



The name Platino is a playful reference to the French and German word 'Platine' meaning 'circuit board', with a slight wink at 'Arduino'. The goal of this project was to design a PCB that would be useful for many MCU applications that may need an LCD and/or push-buttons and that can be easily programmed using WinAVR, AVR studio, BASCOM, Mikro-C or Arduino. The dimensions of the board are adapted to a standard Bopla enclosure so it is easy to finalise a project properly.

Platino supports most 28-pin and 40-pin DIP 8-bit AVR microcontrollers (ATmega8, 16, 32, 48, 88, 164, 168, 324, 328, 644 & 1284). It has extension connectors compatible with Arduino shields and when equipped with the right AVR (ATmega168 or ATmega328 for instance) it is fully compatible with Arduino programs (sketches) too. It also has extension connectors compatible with arduino boards.

Steckiquitäten 2.0

Von Thijs Beckers (Elektor-Redaktion & Labor)

In der diesjährigen Märzausgabe konnte man sich von einer Auswahl an alten Steckverbindern aus der Sammlung historischer Elektronik meines Kollegen Jan Buiting beeindrucken lassen. Die Bilder hatten eine große Resonanz bei unseren Lesern zur Folge, die sich vom einen oder anderen Stück angesprochen fühlten. Wir erhielten viele Beschreibungen und Bilder der ältesten und seltsamsten Steckvorrichtungen, von denen es sicher viele wert sind, gezeigt zu werden. Hier also eine zufällige Auswahl an Bildern samt Beschreibungen:



Die erste Sammlung von Dr. Klaus Rohwer zeigt Stecker von Hirschmann, von denen einige recht sicher aus den 1950ern stammen. Bei den ersten vier Exemplaren handelt es sich um Netzstecker, das große

Teil ist wohl ein „UPO“ (Unidentified Plug Object). Vermutlich stammt es von einem Telex. Einen meiner Kollegen erinnerte das UPO an etwas, das bei einem alten Telefon-Steckbrett für militärische Zwecke verwendet wurde.

Der zweite Steckersatz besteht hauptsächlich aus Audio-Steckern des WDR (Westdeutscher Rundfunk) in Köln. Elektor-



Leser W. Richter hatte mit diesen seit Anfang der 1950er Jahre zu tun. Es wurden wohl viele Adapter

zur Koppelung unterschiedlicher Stecker gebaut. Herr Richter meint dazu: „Der einzige Adapter, den wir nicht gebaut haben, war der von Lemos auf Gardena.“

Stecker und Kupplung gehören zu einem „Pansanitor“ – einem Reizstromgerät von 1928 (siehe auch die Rubrik RETRONIK in dieser Ausgabe). Laut Herrn Butte sind diese Antiquitäten aus glasiertem Porzellan. Vermutlich sind sie die ältesten der abgebildeten Steckvorrichtungen.



Diese kleine Sammlung stammt von W. Haas. Los geht es mit einem Antennenstecker für ein „LMK-Radio“ links oben. Daneben ist ein Stecker für ein symmetrisches 240-Ω-VHF-Kabel. Dann kommen zwei weiße Stecker für symmetrische 240-Ω-Kabel für das VHF-Band III sowie zwei rote Stecker für symmetrisches Kabel für die UHF-Bänder IV/V. Unten rechts ist ein Stecker für VHF und UHF mit viereckigen Stiften zu sehen. Bei den restlichen Abbildungen handelt es sich um diverse Adapter von „Banane“ auf einzelne Stifte; der Blaue mit Kabel und Befestigungsschraube.

P. van de Meeren-donk aus den Niederlanden schickte uns unter anderem dieses UPO: Es sieht aus wie ein Röhrensockel, ist aber keiner. Soweit man sehen kann, ist es ein Oktalstecker aus Bakelit. Verwendungszweck unbekannt.



Wir haben noch etliche andere Fotos und einige ältere Stecker in natura (reales 3D) geschickt bekommen. Wir erhielten auch Bemerkungen zu den Fotos in der 1.0-Version von „Steckiquitäten“. Unter anderem ist die Nummer 10 wohl ein Steckverbinder für symmetrische 240-Ω-Radio-Antennen mit 3-mm- und 4-mm-Löchern. Die 3 mm sind für VHF und UHF, die 4 mm für VHF (FM) gedacht.

Die Nummer 11 ist der historische Nachfolger von Nummer 10. Dieser gepolte Steckverbinder bietet einen Anschluss für eine AM-Antenne (vertikal) und Masse (horizontal).

Überlegungen gab es auch zu Nummer 6: Dieses Teil ist wohl nicht für Lautsprecher gedacht, denn Lautsprecher mit hohen Impedanzen waren früher oft direkt an die Anode angeschlossen, was zu gefährlich hohen Spannungen an Kabeln und Steckern führte.

(120303)

Bauelemente-Tipp

Von Raymond Vermeulen (Elektor-Labor)

Blitzdetektor AS3935

Nachdem mein Büro durch einen Blitzeinschlag in Mitleidenschaft gezogen wurde, schlug mein Kollege Luc Lemmens den Einsatz des nur 4 x 4 mm² großen Chips AS3935 vor. Eine geeignete Antenne vorausgesetzt kann ein Mikrocontroller mit diesem IC über SPI oder I²C die Entfernung zur Unwettergrenze bestimmen. Wohlgedenkt: Nicht die Entfernung zum Blitz, sondern die Entfernung zum Rand der Gewitterzelle! Die Distanz kann 5 bis 40 km betragen.

(120405)gd

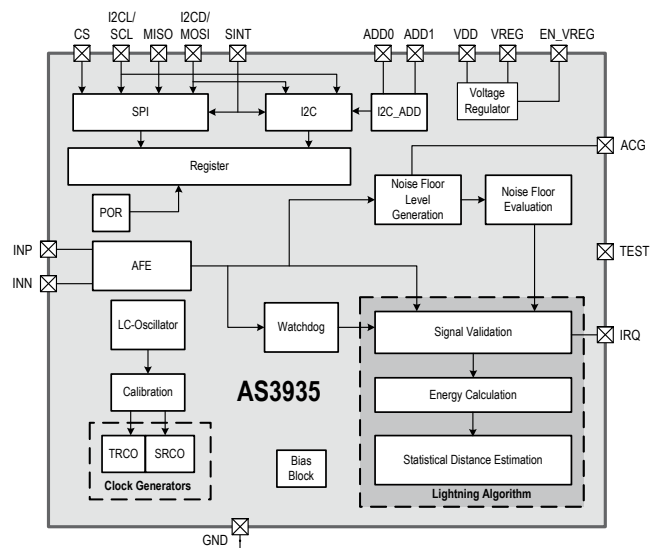
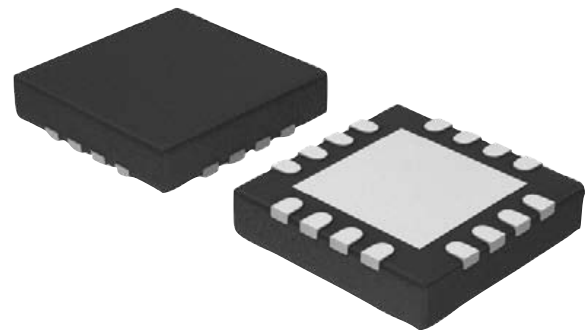
AS3935

Über den Mikrocontroller sind ferner zahlreiche Einstellungen möglich. Erstaunlich ist zweifellos, dass der Algorithmus dieses Chips unterscheiden kann, ob die empfangenen Signale natürlichen Ursprungs oder künstlich vom Menschen verursacht sind. Das bedeutet, dass der Chip auch in Gebäuden einsetzbar ist.

Bemerkenswert ist ferner, dass sich die Rauschschwelle des empfangenen Signals über ein Register anpassen lässt. An den Mikrocontroller wird ein Interruptsignal ausgegeben, sobald das Rauschen das eingestellte Niveau übersteigt. In diesem Fall kann der Mikrocontroller die Rauschschwelle nach oben korrigieren. Der auf dem Chip integrierte Signalverstärker hat einen speziellen Modus für den Einsatz im Freien. Weil künstliche Störquellen den sicheren Betrieb erschweren können, ist über ein anderes Register die Erkennungsschwelle anhebbar. Allerdings geht dies auf Kosten der Messgenauigkeit.

Als Antenne empfiehlt der Hersteller einen Schleifendipol, der einem parallelen LC-Resonanzkreis mit der Güte $Q=15$ und der Resonanzfrequenz 500 kHz entspricht. Da der Bau einer optimalen Antenne schwierig ist, sind auf dem Chip Tools zum Tunen der Antenne eingebaut. In einem Register kann festgelegt werden, dass ein Teilfaktor auf die Resonanzfrequenz angewendet wird, er ist über die IRQ-Leitung steuerbar. In einem anderen Register kann der Mikrocontroller eine Kapazität von 0 bis 120 pF in Stufen von 8 pF zuschalten. Nachdem die Antenne getuned ist, sind auch die internen Oszillatoren kalibrierbar. Bei einem Interrupt geht die Interruptleitung auf hohes Signal. Danach hat der Mikrocontroller Gelegenheit, die Ursache des Interrupts aus dem Chip auszulesen. Auch die minimale Anzahl der Blitzentladungen innerhalb 15 Minuten lässt sich festlegen, falls einzelne, sporadisch auftretende Blitze nicht erfasst werden sollen.

Da Elektor-Leser erfahrungsgemäß ein hohes Interesse an wetterbezogenen Systemen haben, kann es durchaus sein, dass Elektor in absehbarer Zukunft ein Projekt mit dem Blitzdetektor AS3935 entwickelt.



Description	Condition	Value
Supply voltage range	EN_VREG = VDD	2,4...5,5 V
Supply voltage range	EN_VREG = GND	2,4...3,6 V
Power-down mode current	VREG = OFF	1 μ A
Listening mode current	VREG = OFF	8 μ A
Signal verification mode current		350 μ A

Datenblatt AS3935: www1.futureelectronics.com/doc/AUSTRIAMICROSYSTEMS/AS3935.pdf

Application Note des Entwicklungskits:

http://media.digikey.com/pdf/Data%20Sheets/Austriamicrosystems%20PDFs/AS3935_EvalManual_AN.pdf

Embedded Linux leicht gemacht (3)

Softwareentwicklung

Von Benedikt Sauter [1]

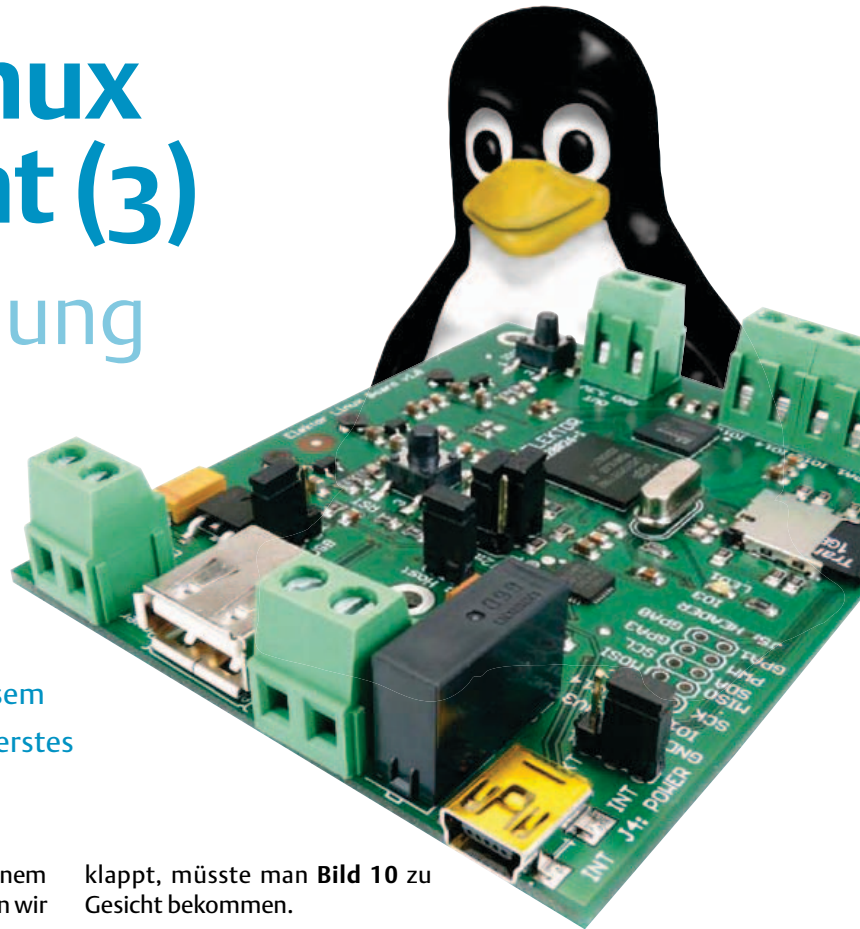
Erst die richtige Software erweckt einen Mikrocontroller zum Leben. Zusätzlich zur klassischen Firmware muss man bei Embedded GNU/Linux auch die Betriebssystemkomponenten generieren. In diesem Artikel zeigen wir, wie das funktioniert. Und ein erstes kleines C-Programm schreiben wir auch!

Ein Embedded-Linux-System entwickelt man am besten mit einem Linux-System, üblicherweise auf einem PC. Als Basis verwenden wir ein „Ubuntu“-Linux in der Version 12.04 [2]. Was benötigt man, um das System zu installieren? Eigentlich nicht viel – freien Platz auf der Festplatte und idealerweise einen Netzwerkanschluss.

Zuerst muss man sich das Abbild der Installations-CD aus dem Internet herunterladen [2]. Für unsere Arbeiten bieten sich die 32-bit- oder 64-bit-Desktop-Versionen an, im Zweifel wählt man die 32-bit-Version.

Nachdem man das CD-Abbild heruntergeladen hat, muss man es nur noch mit einem Brennprogramm auf eine CD brennen. Wichtig ist, dass man es als „Image“ und nicht als Datei auf eine CD brennt. Ist die CD gebrannt, kann man diese in den PC stecken und dort von CD booten (muss man entsprechend im BIOS konfigurieren). Ubuntu meldet sich nun mit dem Startbildschirm in **Bild 1**. Um mit dem System vernünftig arbeiten zu können, sollten wir es auf Festplatte installieren, also wählen wir den zweiten Menüpunkt (**Bild 2**). Dann wird man Schritt für Schritt durch den Installationsvorgang geführt. Zuerst wählt man die Sprache (**Bild 3**).

Third-Party-Software (**Bild 4**) müssen wir nicht installieren. Im folgenden Fenster in **Bild 5** kann man festlegen, ob Linux über die ganze Festplatte verfügen darf (etwa bei einem Computer, auf den noch kein Betriebssystem aufgespielt wurde). Falls auf dem Computer bereits ein anderes Betriebssystem installiert ist, muss man entweder über eine zweite Festplatte verfügen oder man teilt die Platte in mehrere Partitionen ein. In **Bild 6** wird das Laufwerk oder die Partition ausgewählt, dann geht es mit den Fenstern in **Bild 7** und **8** weiter. Was es mit der Passwortvergabe (**Bild 9**) auf sich hat, erklären wir später. Wenn alles



klappt, müsste man **Bild 10** zu Gesicht bekommen.

Eine andere (ziemlich bequeme) Möglichkeit ist, das Betriebssystem in einer virtuellen Maschine laufen zu lassen. Speziell für Elektor-Leser hat der Autor ein fertiges Image eines Linux-Computers für Entwicklungszwecke generiert, das man von der Elektor-Website downloaden kann [3]. Als Virtualisierungsprogramm kommt „VirtualBox“ zum Einsatz, das kostenlos unter [4] erhältlich ist. Nachdem Virtualbox installiert ist, kann man das Image einfach über Hauptmenü > Datei > „Appliance import“ importieren. Ist die Maschine importiert, kann man diese direkt starten.

Toolchain auf CD

Nach dem Start des Betriebssystems geht es im nächsten Schritt darum, die Toolchain zu installieren. Wer sich für das VirtualBox-Image entschieden hat, braucht die Schritte in den nächsten zwei Abschnitten nicht auszuführen, die Toolchain wartet dann



Bild 1. Mit der Pfeiltaste eins nach unten.



Bild 2. „Install Ubuntu“ wählen und Enter eingeben.

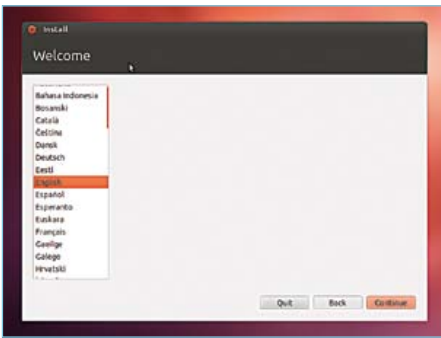


Bild 3. Sprache wählen.

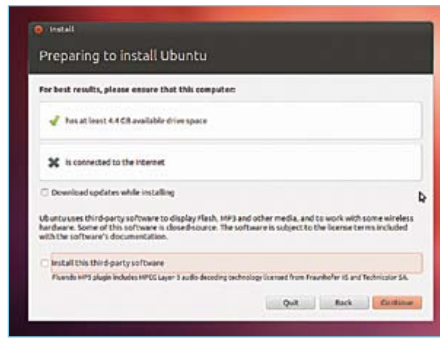


Bild 4. Vorbereitung der Installation.



Bild 5. Installations-Ziel auswählen.

bereits fertig installiert auf ihren Einsatz.

Der schnellste Weg unter Linux ist natürlich die Konsole. Am einfachsten öffnet man sich eine frische Konsole auf dem PC mit der Tastenkombination STRG-ALT-T. In der Konsole wechselt man dann in den Ordner „/tmp“:

```
cd /tmp
```

Dann folgt der Download der ARM-Toolchain-CD direkt von der Konsole aus mit dem Tool wget:

```
wget ftp://ftp.denx.de/pub/eldk/5.0/iso/armv5te-qte-5.0.iso
```

Nachdem das CD-Image heruntergeladen ist, kann man dieses unter Linux direkt öffnen, ohne sich erst eine echte CD brennen zu müssen. Das CD-Abbild muss man aber zuerst „mounten“, was man mit „einhängen eines Datenträgers“ in das Betriebssystem übersetzen könnte.

Vorher wechselt man in den Ordner /media.

```
cd /media
```

Prinzipiell könnte man die CD in jeden beliebigen Punkt im Verzeichnisbaum einhängen, jedoch gibt es unter Linux gewisse Standards; und so kann sich jeder immer schnell in einem neuen System zurechtfinden. Wie ein Standard-Verzeichnisbaum aufgebaut ist, wird später noch beschrieben.

Wir wollen nun einen leeren Ordner „eldk-iso“ anlegen, in den wir das CD-Abbild in das Dateisystem einhängen werden. Als Windows-User darf man sich dies nicht so vorstellen, dass das CD-Image in diesen Ordner tatsächlich kopiert wird, vielmehr wird der Inhalt der CD über dieses Verzeichnis zugänglich gemacht. Denn in Linux wird ja alles durch Verzeich-



Bild 6. Festplatte auswählen.



Bild 7. Zeitzone wählen.

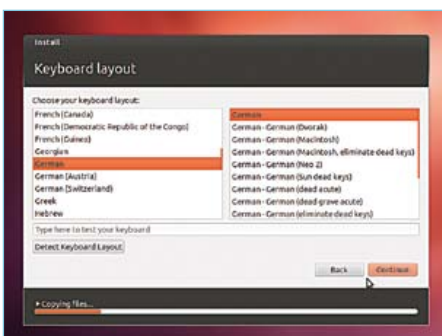


Bild 8. Tastatur-Layout einstellen.

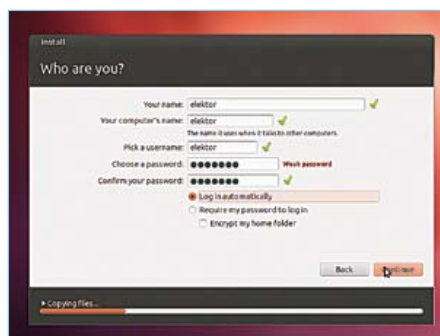


Bild 9. Benutzernamen und Passwort vergeben.



Bild 10. Installation starten.

Benutzerrechte

Unter Linux gibt es immer den sogenannten Benutzer „root“. Dies ist der User mit den meisten Rechten, alle anderen Benutzer haben gewöhnliche Zugangsberechtigungen. Möchte man jedoch auf System-Dateien, Geräte o.ä. zugreifen, braucht man eben diese „Root-Rechte“. Jetzt könnte man der Einfachheit halber immer als Benutzer „root“ arbeiten, doch das sollte man sich nie angewöhnen. Linux ist unter anderem so sicher, weil man dank der Benutzerrechte viele Gefahren eindämmen kann. User sollten grundsätzlich nur die Rechte eingeräumt bekommen, die benötigten Programme

auszuführen, aber nicht an System-Dateien oder andere wichtige Daten herankommen. Als Linux-Programmierer kann man den Anwendern immer gezielt die richtigen Rechte einräumen, so dass es keinen Zwang gibt, komplett als „root“ zu arbeiten. Auf unserem Linux Board werden auch wir schon bald einen Benutzer anlegen, mit dem wir dann arbeiten bzw. unsere Programme ausführen lassen werden.

Benötigt man aber nur kurzfristig „Root-Rechte“, etwa zum Anlegen eines Verzeichnisses in einem System-Ordner, dann kann

man unter den meisten neuen Linux-Distributionen (wie auch bei unserer Ubuntu-Version) einfach das Kommando `sudo` vor den eigentlichen Befehl schreiben. Dies bedeutet, dass der nächste Befehl einmalig als Root ausgeführt wird.

Hat man sich ein eigenes Linux-System installiert, muss man danach das dabei vergebene Passwort eingeben. Wenn man das Elektor-VirtualBox-Image verwendet, dann lautet das Passwort „elektor“ (komplett klein geschrieben).

nisse und Dateien symbolisiert!

Hier ist der Befehl zum Anlegen des Ordners:

```
sudo mkdir eldk-iso
```

Jetzt wird man nach einem Passwort gefragt, weil wir zum Anlegen des Verzeichnisses kurzzeitig mit Root-Rechten arbeiten müssen (siehe Kasten „Benutzerrechte“). Dass wir mit diesen Rechten arbeiten wollen, haben wir der Shell mit dem Kommando `sudo` mitgeteilt.

Hat man sich ein eigenes Linux-System installiert, muss man das dabei vergebene Passwort jetzt eingeben. Wenn man das Elektor-VirtualBox-Image verwendet, dann lautet das Passwort „elektor“ (komplett klein geschrieben).

Jetzt kann das CD-Abbild in das Dateisystem eingehängt werden, so dass wir auf den Inhalt der CD zugreifen können:

```
sudo mount -o loop /tmp/armv5te-qte-5.0.iso /media/eldk-iso
```

Von unserer aktuellen Position im Verzeichnisbaum können wir mit der folgenden Eingabe sozusagen in die CD wechseln:

```
cd eldk-iso
```

Installation der Toolchain

Direkt im Ordner befindet sich ein kleines Skript, mit dem die Toolchain installiert werden kann. Zum Ausführen benötigen wir wieder die Root-Rechte:

```
sudo ./install.sh -s -i qte armv5te
```

Als Ausgabe erscheint so etwas:

```
*** Installing ./targets/armv5te/eldk-eglibc-i686-arm-toolchain-qte-5.0.tar.bz2
```

Ist die Installationsroutine beendet, kann das Verzeichnis verlassen werden:

```
cd ..
```

Dieser Schritt war wichtig, damit wir im nächsten Schritt die CD wieder aushängen können. Niemals darf man sich in dem Ordner befinden, der wieder ausgehängt wird, sonst gibt es eine Fehlermeldung und das Betriebssystem verweigert diesen Befehl.

```
sudo umount /media/eldk-iso
```

Das für die Toolchain-CD erstellte Verzeichnis kann jetzt auch wieder entfernt werden.

```
sudo rmdir eldk-iso/
```

Nach der Installation befindet sich die Toolchain im Ordner „`/opt/eldk-5.0/`“. Damit die Toolchain von jedem Punkt im Verzeichnisbaum über die Konsole aufgerufen werden kann, muss diese in die PATH-Variable mit aufgenommen werden. Diese Linux-Umgebungsvariable enthält die Pfade, in dem das System nach Programmen und Dateien sucht.

Am besten legt man sich hierfür eine kleine Skriptdatei („`set.sh`“) an, die man immer vor der Arbeit in der Konsole aufruft. Dafür öffnet man mit dem Editor die neue Datei ...

```
gedit set.sh
```

... und gibt den folgenden Inhalt ein:

```
#!/bin/bash
```

```
P1=/opt/eldk-5.0/armv5te/sysroots/i686-oesdk-linux/
usr/bin/armv5te-linux-gnueabi/
P2=/opt/eldk-5.0/armv5te/sysroots/i686-oesdk-linux/
bin/armv5te-linux-gnueabi/
```

```
export ARCH=arm
export CROSS_COMPILE=arm-linux-gnueabi-
export PATH=$P1:$P2:$PATH
```

Der letzte Befehl erweitert die PATH-Variable um die oben definierten Pfade.

Anschließend speichert man die Datei am besten direkt im Startverzeichnis des Benutzers ab (auch „Homeverzeichnis“ genannt, dort sollte man sich automatisch befinden). Die Datei muss man zu Beginn der Arbeit mit der Konsole „sourcen“, das heißt von der Shell (dem Linux-Kommandozeilen-Interpreter) einlesen lassen. Dies macht man zum Beispiel mit folgendem Befehl:

```
.. ./set.sh
```

Hierbei genau auf die Schreibweise achten: Punkt Leerstelle Punkt Schrägstrich ...!

Möchte man nicht jedes Mal nach dem Aufruf einer neuen Konsole `.. ./set.sh` eingeben, kann man den Inhalt dieser Datei auch in der Datei `„.bashrc“` unterbringen, diese wird bei jedem Öffnen der Konsole automatisch ausgeführt. Die `„.bashrc“` liegt im Homeverzeichnis, in das man immer mit `cd` gelangt. Dann öffnet man die Datei mit:

```
gedit .bashrc
```

Der Compiler in Aktion

Die Toolchain-Programme zur Erstellung des Linux-Kernels und des Bootladers beginnen mit `„armv5te-“`. Der Compiler GCC trägt den Namen `„armv5te-gcc“`. Mit:

```
armv5te-gcc --version
```

kann man sich die Versionsnummer des Compilers ausgeben lassen (vor `„version“` stehen zwei einzelne Striche!). Hiermit kann man auch prüfen, ob die PATH-Variable nun die passenden Pfade enthält. Um eigene Programme für Linux übersetzen zu können, muss man die Toolchain verwenden, die mit `„arm-linux-gnueabi-“` (nicht `armv5te-!`) startet. Wie wäre es jetzt mit einem einfachen „Hello World“?

Datei anlegen:

```
gedit hello.c
```

Inhalt der Datei eintippen:

```
#include <stdio.h>

int main(void)
{

    printf(„Hello World!\r\n“);
    return 0;

}
```

Auf Speichern und Beenden klicken. Jetzt ist man wieder in der Konsole.

Nun übersetzen wir `„hello.c“` auf dem PC:

```
arm-linux-gnueabi-gcc -o hello hello.c
```

Um zu überprüfen, ob das Ganze erfolgreich war, kann man die Datei `„hello“` nun auf die SD-Karte kopieren. Dafür steckt man die SD-Karte des Elektor-Linux-Boards (aus dem ausgeschalteten Board ziehen) in einen PC-Kartenleser. Nach dem Anstecken des Lesers inklusive der Karte muss man einen kurzen Moment warten. Normalerweise geht jetzt automatisch ein Fenster in Ubuntu auf. Dieses kann man getrost schließen, da wir die Datei per Konsole auf die SD-Karte kopieren wollen.

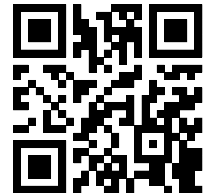
Ein großes Betriebssystem wie Ubuntu mountet den externen Datenträger automatisch ein. Wir müssen jetzt nur herausfinden, wo Ubuntu die SD-Karte eingehängt hat. Am besten wechselt man dafür in den Ordner `/media`

```
cd /media
```

und gibt dort einmal

```
ls
```

(= „list“) ein. Sollten sich dort mehrere Ordner befinden, kann man einfach mal mit



Erste Hilfe bei „zerschossener“ SD-Karte

Wenn das System nach dem Booten von einer SD-Karte nicht korrekt mit halt oder poweroff heruntergefahren wird, besteht die Gefahr, dass die Dateisystemtabelle zerstört wird.

Dann erscheinen typischerweise die „EXT2-fs errors“:

Dateisystem „EXT2-fs (mmcblk0p1): error: ext2_lookup: deleted inode referenced: 694962“:

```
e2fsck 1.42 (29-Nov-2011)
/dev/sdh1 was not cleanly unmounted, check force.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sdh1: 66/24576 files (0.0% non-contiguous), 8294/97988 blocks
```

Glücklicherweise kann man das Dateisystem mit einem Linux-PC über die Konsole wiederherstellen.

Zuerst steckt man die SD-Karte in den Kartenleser und prüft mittels dmesg, welcher Gerätenamen vergeben wurde. Man erhält zum Beispiel folgende Ausgabe:

```
[ 1549.424156] sd 7:0:0:2: [sdh] Assuming drive cache: write through [ 1549.425624] sdh: sdh1 sdh2 [
1549.427527] sd 7:0:0:2: [sdh] Assuming drive cache: write through [ 1549.427533] sd 7:0:0:2: [sdh] Attached SCSI
removable disk [ 1549.730223] EXT2-fs (sdh1): warning: mounting unchecked fs, running e2fsck is recommended
```

Die erste Partition ist die, um die es geht, in diesem Fall lautet der Name „sdh1“. Hier befindet sich ein ext2-formatiertes Dateisystem, welches man reparieren kann.

Als erstes trennen wir die Verbindung in das Dateisystem:

```
umount /dev/sdh1
```

Jetzt kann man das Dateisystem mit dem Tool „e2fsck“ wiederherstellen (mit „fsck.ext2“ funktioniert es auch) ...

```
sudo e2fsck /dev/sdh1
```

... und man bekommt eine Ausgabe, wie sie im Screenshot oben rechts zu sehen ist (gezeigt ist die Ausgabe in englischer Sprache). Ab und zu kommt es vor, dass das Programm fragt, ob bestimmte Aktionen durchgeführt werden sollen. In diesem Fall immer alles mit „y“ beantworten („j“ bei deutscher Sprache).

Jetzt sollte wieder alles fehlerfrei arbeiten!

```
cd <Verzeichnis-Name>
```

in die jeweiligen Ordner schauen (danach jeweils mit cd .. wieder eine Ebene nach oben wechseln). Das Verzeichnis, in dem man Zugriff auf den Inhalt der SD-Karte hat (eingehängt im Ordner /media), erkennt man an einer typischen langen Nummer. In dem Ordner liegt das Dateisystem des Linux-Systems, es enthält zum Beispiel die Dateien mit den Namen „zImage“ oder „swapfile1“. Noch ein Tipp: Wenn man nicht mehr weiß, in welchem Ordner man sich gerade befindet, kann man einfach cd eingeben, dann landet man immer wieder im Startverzeichnis des eigenen Benutzer-Ordners. Oder man gibt pwd ein, dann wird direkt der Pfad ausgegeben.

Hallo Welt!

Um „Hello World“ testen zu können, kann man den SD-Karten-Ordner aufsuchen und die Datei einfach direkt hineinkopieren:

```
cp ~/hello ./
```

Anschließend muss man das Verzeichnis manuell aushängen, damit das Betriebssystem alle noch nicht auf die Speicherkarte geschriebenen Blöcke endgültig überträgt.

```
sudo umount /media/<Verzeichnis-Name>
```

„<Verzeichnis-Name>“ ist jeweils der Name des Ordners, den das Betriebssystem für die SD-Karte vergeben hat.

Jetzt kann man die SD-Karte wieder in das Board stecken und von dieser booten. Wir verbinden uns anschließend von einem PC-Terminalprogramm aus mit dem Board, so wie im letzten Teil beschrieben [5].

Mit Hilfe des Terminalprogramms wechseln wir auf dem Board in das Hauptverzeichnis des Dateisystems:

```
cd /
```

und rufen unser Programm auf:

```
./hello
```

Als Ausgabe sollte nun „Hello World!“ im Terminalprogramm erscheinen (siehe **Bild 11**).

Wenn von der SD-Karte gebootet wird, ist es wichtig, dass man das System nach getaner Arbeit auch korrekt herunterfährt. Hierfür ist der Befehl

```
halt
```

zuständig. Dann muss man warten, bis tatsächlich System halted ausgegeben wurde, sonst besteht die Gefahr, dass die Dateisystemtabelle der SD-Karte zerstört wird. Wenn dies passiert ist, erscheinen typischerweise die „EXT2-fs errors“. Mit einem Linux-PC (so wie wir ihn hier installiert haben) kann man das Dateisystem aber wiederherstellen, siehe Kasten.

Bootloader und Kernel

Jetzt können wir uns an die Übersetzung der Betriebssystemkomponenten Bootloader und Kernel machen.

Die notwendigen Quelltexte (aktuell 290 MB) findet man auf der Elektor-Website [3]. Am einfachsten lädt man die Files mit einem Browser herunter, unter Linux liegen diese anschließend meist im Ordner „Downloads“.

Nach dem Download der Datei sollte man in den Ordner „Downloads“ wechseln und dort die Datei „120026-11.zip“ finden. Die Datei kann einfach in das eigene Benutzer-Verzeichnis verschoben werden:

```
mv ~/Downloads/120026-11.zip ~/
```

Um die Datei zu entpacken, wechselt man mit `cd` in das eigene Benutzer-Verzeichnis und gibt ein:

```
unzip 120026-11.zip
```

Die nun folgende Ausgabe auf der Konsole ist in **Bild 12** zu sehen.

Bootloader erstellen

Der Bootloader wird nach dem Einschalten des Prozessors von der SD-Karte in das interne SRAM des LPC3131 kopiert (wenn die Jumper entsprechend gesetzt sind, siehe [5]). Nachfolgend übersetzen wir diesen Bootloader selbst und übertragen ihn auf die SD-Karte, so dass von dort gebootet werden kann. Bevor es losgeht, müssen noch zwei Pakete in das Ubuntu-Linux installiert werden:

```
sudo apt-get install patch libncurses5-dev
```

Jetzt wechselt man in den Ordner für die Quelltexte ...

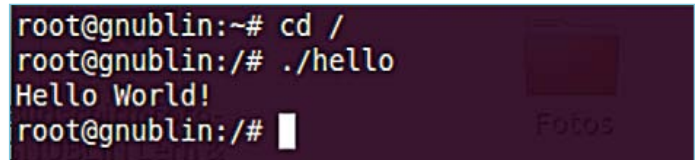


Bild 11. „Hello World“ auf dem Board.

```
cd ElektorLinuxBoardDownload_20120509
```

... und entpackt das tar-Archiv, in dem sich der Bootloader befindet:

```
tar xvzf bootloader.tar.gz
```

Anschließend wechselt man in den neuen Ordner „bootloader“ ...

```
cd bootloader
```

... und entpackt den eigentlichen Quelltext:

```
tar xvzf apex-1.6.8.tar.gz
```

Jetzt kommt ein wesentlicher Schritt für das spätere Weitergeben von Änderungen. Wir erzeugen uns sozusagen eine „Arbeitskopie“ vom Original-Quelltext und werden alle unsere Änderungen dort machen. So könnten wir später auch einfach einen Patch erzeugen und veröffentlichen.

```
mv apex-1.6.8 work_1.6.8
```

```
cd work_1.6.8
```

In unseren Quelltextbaum müssen wir anschließend die Änderungen einspielen, die für das Elektor-Linux-Board nötig sind:

```
patch -p1 < ../apex-1.6.8_lpc313x.patch
patch -p1 < ../gnublin-apex-1.6.8.patch
```

Das Übersetzen des Bootloaders wird durch sogenannte Konfigurationsdateien gesteuert. Wir müssen die Vorlage dieser Datei in den Ordner kopieren, und zwar mit dem neuen Namen „.config“ (auch hier ist der Punkt von dem Wort „config“ sehr wichtig, er kennzeichnet die Datei als „versteckte“ Datei im Betriebssystem).

```
cp ../gnublin-apex-1.6.8.config .config
```

Für diejenigen, die die Toolchain selbst installiert haben:

Falls man die Umgebungsvariablen noch nicht automatisch durch die „bashrc“ gesetzt hat, muss man nun die selbst erstellte Skript-Datei vor dem Übersetzen mit `./set.sh` aufrufen.

Nun startet man den Übersetzungsvorgang:

```
make apex.bin
```

Jetzt würde normalerweise die Stelle kommen, an der man die Firmware (Bootloader) mit einem Programmer in den Flashspeicher des Mikrocontrollers überträgt. In unserem Fall können wir die Firmware ganz einfach mit Standard-Linux-Programmen auf die SD-Karte schreiben.

Die SD-Karte wird dazu wieder in einen Kartenleser am PC gesteckt. Mit dem Befehl

```
dmesg
```

kann man sich einmal die Meldungen des Kernels ansehen. Auf dem PC des Autors sieht dies beispielsweise so aus wie in **Bild 13**. Man sieht dort, dass die SD-Karte als „/dev/sdh“ erkannt wurde und zwei Partitionen „sdh1“ und „sdh2“ enthält. Am besten hängt man die Geräte sicherheitshalber nochmal aus, da wir die Firmware nicht über das Dateisystem, sondern direkt auf den entsprechenden Block der SD-Karte schreiben werden:

```
sudo umount /dev/sdh1
sudo umount /dev/sdh2
```

Mit folgender Zeile (muss man selbst noch anpassen!) kann man die „apex.bin“-Datei auf die SD-Karte schreiben, so dass sie der LPC3131 beim Booten findet:

```
sudo dd if=src/arch-arm/rom/apex.bin of=/dev/sdh2 bs=512
```

Mit der Zeile kopieren wir den Bootloader an den Anfang der zweiten Partition, mit einer Blockgröße von 512 Byte. Unter einem Betriebssystem wie Linux können wir nicht sagen, wann die geänderten Blöcke tatsächlich in den NAND-Speicher der SD-Karte geschrieben werden. Mit dem Befehl

```
sync
```

kann man das Betriebssystem aber zwingen, alle noch nicht geschriebenen Blöcke jetzt definitiv zu schreiben. Anschließend befindet sich der neue Bootloader sicher auf der SD-Karte. Gerne kann man nun einmal ein Test-Boot durchführen.

Kernel übersetzen

Auf ähnliche Weise wie der Bootloader lässt sich auch der Kernel übersetzen. Zuerst kommt der Wechsel in das Hauptverzeichnis des eigenen Benutzers mit `cd`, dann wechselt man in den Ordner für die Quelltexte...

```
cd ElektorLinuxBoardDownload_20120509
```

... und entpackt den Quelltext des Kernels:

```
tar xvzf linux-2.6.33-lpc313x-gnublin-032012.tar.gz
```

Man wechselt in den Ordner des Kernels ...

```
cd linux-2.6.33-lpc3131x
```

... und startet einen Übersetzungslauf, um einen bootbaren Kernel zu erzeugen:

```
make zImage
```

Zusätzlich kann man nachladbare Treiber übersetzen. Dies macht man mit einem eigenen Aufruf:

```
make modules
```

Nach dem Übersetzen muss man den Kernel und die Module auf die SD-Karte kopieren. In unserem Fall liegt der Kernel „zImage“ schon dort, doch sollte man den Kernel zu Übungszwecken einmal austauschen. Die Datei „arch/arm/boot/zImage“ im Kernel-Quelltextbaum muss dazu direkt auf die erste Partition der SD-Karte geschrieben werden. Anschließend darf man das Aushängen der Partition nicht vergessen.

Schaffen Sie den Austausch des Kernels schon selbst?

Ein erster Überblick

Nun sind wir in der glücklichen Lage, den Bootloader und Kernel selbst übersetzen zu können. Das ermöglicht uns nun, auch Änderungen am Kernel vornehmen zu können. Natürlich sollte man sich hierzu zumindest grob auskennen. Einen ersten Blick in die Konfiguration des Linux-Kernels kann man mit

```
make menuconfig
```

wagen, den Befehl muss man im Ordner „linux-2.6.33-lpc3131x“ aufrufen. Das Ergebnis sieht man in **Bild 14**. Möchte man z.B. ein bestimmtes USB-Gerät verwenden, muss man hier gezielt den Treiber aktivieren.

In der blauen Oberfläche kann man mit den Pfeil-Tasten navigieren. Mit der Enter-Taste kann man Menüs öffnen und wieder schließen. Wer hier auf Entdeckungstour geht, findet eventuell auch schon den einen oder anderen Treiber eines Gerätes, das er kennt. Im nächsten Artikel werden wir hier noch einmal näher darauf eingehen.

Boot-Image wiederherstellen

Da wir jetzt beginnen, unter der Motorhaube des Elektor-Linux-Boards zu arbeiten, empfiehlt es sich, vorher eine 1:1-Kopie der SD-Karte anzufertigen.

Nach dem Einlegen der Karte in den Kartenleser und dem Anstecken des Lesers am PC prüft man mit `dmesg`, was für ein Device-Name vom Betriebssystem vergeben wurde.

Sicherheitshalber sollte man nochmals die Partitionen aushängen:

```
umount /dev/sd<Buchstabe>1
umount /dev/sd<Buchstabe>2
```

(statt <Buchstabe> muss hier der vom Betriebssystem vergebene Buchstabe stehen, z.B. /dev/sdb1 und /dev/sdb2 oder /dev/sdh1 und /dev/sdh2).

Jetzt kann man sich die Karte 1:1 in eine Datei „dumpen“:

```
sudo dd if=/dev/sd<Buchstabe> of=Image_SD_Karte_
Sicherheitskopie.img
```

Jetzt dauert es einige Zeit. Schließlich findet man mit folgendem Befehl eine Datei im aktuellen Ordner, die so groß ist wie die Karte:

```
ls -lh
```

Jetzt benötigt man eine gleich große SD-Karte, die man in den Kartenleser steckt. Mittels `dmesg` muss man wieder den Gerätenamen ermitteln.

Mit folgendem Befehl schreibt man das Karten-Image auf die neue Karte:

```
sudo dd if=Image_SD_Karte_Sicherheitskopie.img of=/dev/
sd<Buchstabe>
```

Jetzt dauert es nochmal etwas länger als vorhin. Nach einer Erfolgsmeldung sollte man sicherheitshalber `sync` aufrufen, so dass wirklich alle Blöcke auf die SD-Karte geschrieben werden. Daraufhin kann man die Karte zum Testen in das Elektor-Linux-Board stecken. Ein vorheriges `umount` ist hier übrigens unnötig, da das Dateisystem ja nicht eingehängt war (wir haben direkt die Gerätedatei beschrieben).

So einfach kann man sich Sicherheitskopien erstellen, doch leider versagt die Methode bei unterschiedlichen Kartengrößen. Und manchmal möchte man sich auch eine Karte erstellen, die andere Partitionen und ein anderes Dateisystem als die Originalkarte besitzt. Für diese Fälle gibt es einen grafischen Installer, der im nächsten Artikel kurz vorgestellt wird.

Ausblick

Mit diesem Teil der Serie sind wir auf dem Weg zur eigenen Embedded-GNU/Linux-Anwendung ein gutes Stück vorangekommen. Die Entwicklungsumgebung steht, der Bootloader und Kernel wurden übersetzt und ein kleines Programm haben wir auch selbst geschrieben und laufen lassen.

In der nächsten Runde werden wir uns ein wenig mit der Struktur des Quelltextes von Linux beschäftigen, so dass wir zumindest in der Lage sind, einen eigenen Treiber für ein beliebiges Gerät zu integrieren. Außerdem werden wir zeigen, wie einfach mit Skriptsprachen programmiert werden kann!

(120180)

```
Archive: ElektorLinuxBoardDownload_20120509.zip
  creating: ElektorLinuxBoardDownload_20120509/
  extracting: ElektorLinuxBoardDownload_20120509/bootloader.tar.gz
  inflating: ElektorLinuxBoardDownload_20120509/gnublin-installer-1.3-beta-bin.tar.gz
  inflating: ElektorLinuxBoardDownload_20120509/gnublin-installer-1.3-beta-1386.deb
  inflating: ElektorLinuxBoardDownload_20120509/linux-2.6.33-lpc313x-gnublin-032012.tar.gz
  inflating: ElektorLinuxBoardDownload_20120509/rootfs.tar.gz
```

Bild 12. Ausgabe beim Entpacken des Software-Downloads.

```
[ 1148.953837] sd 5:0:0:2: [sdh] Assuming drive cache: write through
[ 1148.955001] sdh: sdh1 sdh2
[ 1148.958583] sd 5:0:0:3: [sdi] Attached SCSI removable disk
[ 1148.959202] sd 5:0:0:2: [sdh] Assuming drive cache: write through
```

Bild 13. An der Kernel-Meldung sieht man, dass die SD-Karte als „/dev/sdh“ erkannt wurde.

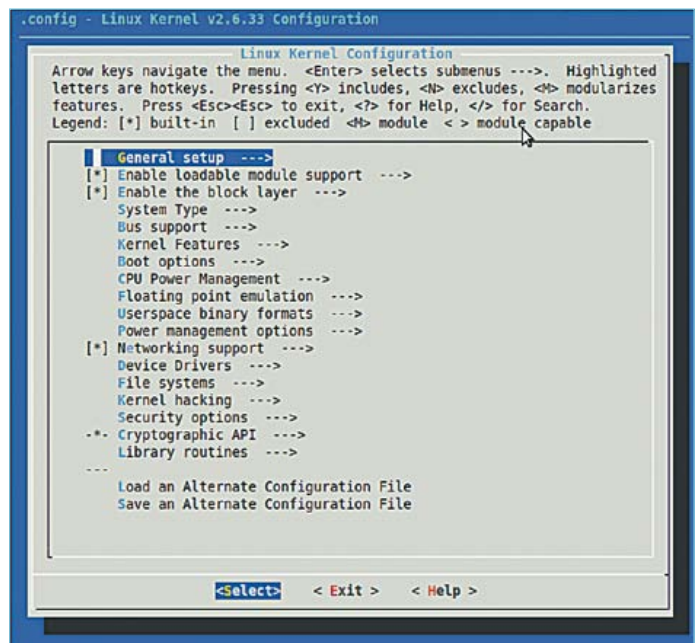


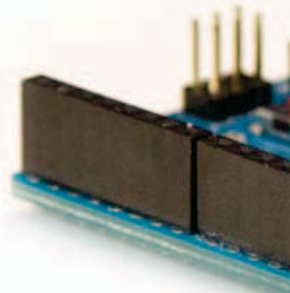
Bild 14. Konfiguration des Linux-Kernels.

Weblinks

- [1] sauter@embedded-projects.net
- [2] www.ubuntu.com
- [3] www.elektor.de/120180
- [4] www.virtualbox.org
- [5] www.elektor.de/120146

Arduino auf Kurs (1b)

Teil 1b: Arduino als Sound-Player



Von David Cuartielles (Spanien)

1-bit-Sound: Wie bitte?!

Bis hierher konnte man sich bei den diversen Code-Experimenten direkt der arduino-eigenen Sprache bedienen. Jetzt aber geht es dem Controller des Arduino Uno an den Kragen: Der ATmega328 wird auf „low level“ gehackt, um Code zu erzeugen, der Daten aus einer WAVE-Datei im Programmspeicher ablegt und diese natürlich auch hörbar machen kann.

Die verwendete Methode ist alles andere als trivial. Zu Anfang ist es gut, Sound einmal spektral anstatt wie üblich zeitbasiert zu betrachten.

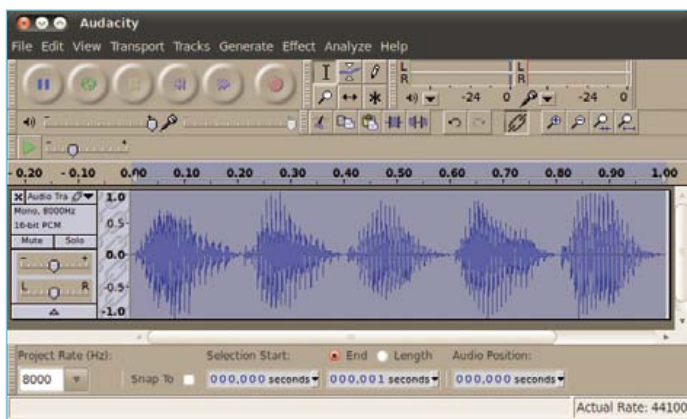


Bild 1. Aufzeichnung menschlicher Sprache „ta-te-ti-to-tu“.

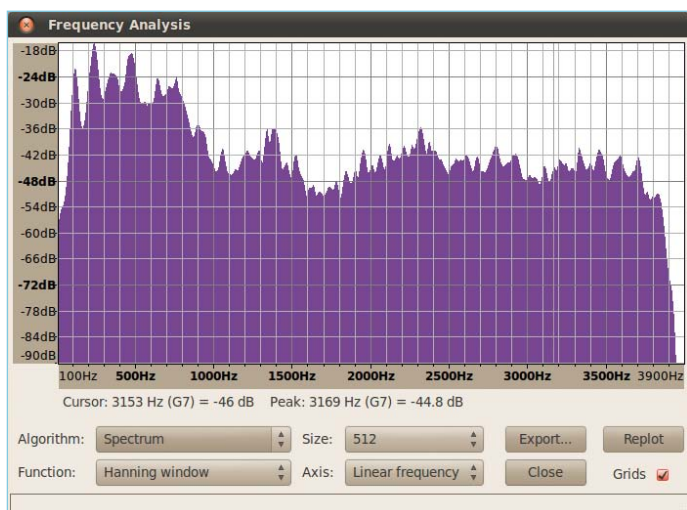


Bild 2. Spektrum des gesprochenen „ta-te-ti-to-tu“.

Sound-Spektrum

Das Spektrum von Audiosignalen ist im Wesentlichen eine Darstellung der Energie als Funktion der Frequenz. Aus einem Spektrum erhält man normalerweise keinen so deutlichen Eindruck des Audiosignals wie in **Bild 1**. Beim Spektrum wird nämlich die Energie in einem Zeitintervall für einzelne diskrete Frequenzen angegeben. Man kann sogar wie in **Bild 2** das Spektrum für einen kompletten Song erstellen.

Alternativ könnte man z.B. die Spektren für aufeinanderfolgende 0,5-s-Abschnitte eines Songs als Folge generieren. Je schmäler die so genannten Time Frames (Zeitabschnitte), desto genauer kann man klangliche Veränderungen über die Zeit erfassen. Die Kurve eines Spektrums kann so gut wie alle möglichen Formen annehmen. Man kann nicht zwingend aus dem Spektrum auf den konkreten Klang schließen, denn zwei unterschiedliche Signale können sehr wohl sehr ähnliche Spektren haben.

Trotzdem ist die Breite der Time Frames entscheidend. Wenn sie nämlich schmal genug sind, werden zwei unterschiedliche Signale mit gleichem Spektrum vom menschlichen Ohr als gleich wahrgenommen, da die Sinneszellen im menschlichen Ohr wie eine Art spektrale Sensoren funktionieren. Dies ist übrigens die Erklärung dafür, warum (verlustbehaftete) Signalkompression bei Audiodaten so gut funktioniert. MP3s z.B. sind für unsere Ohren gut genug, obwohl sich das daraus generierte Signal messbar vom Original unterscheidet.

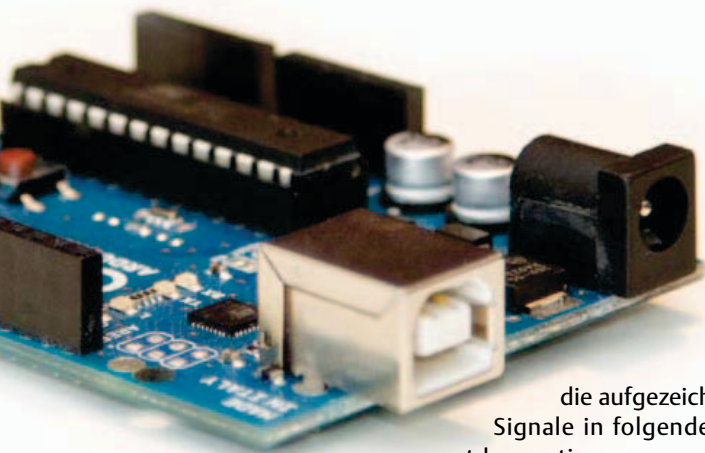
So funktioniert Sound im 1-bit-Verfahren [1]: Das Audiosignal wird durch ein PWM-Signal generiert, dessen Mittelwert in etwa dem gewünschten analogen Signal entspricht. Dieses Procedere ist gut genug, um Audiosignale mittlerer Qualität mit einem Mikrocontroller zu generieren. Nachfolgend wird das Vorgehen beschrieben. Es beginnt mit einer WAVE-Datei, die mit einem PC generiert wird. Als nächstes werden diese Daten gefiltert und in passende Arrays zur Speicherung mit Arduino umgewandelt.

Optimales Digitalisieren und Filtern

Zum Aufnehmen von Audiosignalen gibt es viele Tools. Ich selbst würde immer Audacity [2] den Vorzug geben, denn dabei handelt es sich nicht nur um Open Source Software, Audacity enthält auch Werkzeuge, um für Mikrocontroller passende Daten zu erzeugen. Die **Bilder 1** und **2** sind Screenshots von Audacity.

Bevor es losgeht, sollte das Audiosignal zur Bandbreitenbegrenzung gefiltert werden. Ein Tiefpass mit 4 kHz Grenzfrequenz ist geeignet, da der Arduino-Sound-Player mit einer Abtastfrequenz von 8 kHz arbeitet. Die Signalanteile oberhalb der halben Abtastfrequenz hätten nämlich Artefakte zur Folge.

Auch wenn das reine Abspielen mit akzeptabler Qualität keine wirkliche Herausforderung für einen Mikrocontroller ist, so ist sein Speicher in der Regel doch arg begrenzt. Mit niedrigerer Qualität braucht man weniger Platz und folglich kann man dann auch Signale von einigen Sekunden ohne externe Speicher-Chips abspielen. Ein gut passendes Datenformat ist PCM WAVE mit 8 bit (auch unter „Microsoft unsigned 8 bit“ bekannt). Es reicht aus, um menschliche Sprache gut genug wiederzugeben. Also sollte das Audio-Tool



die aufgezeichneten
Signale in folgendes Form-
at konvertieren: mono, 8 bit,
PCM WAVE.

Audio/Text-Konversion

Jetzt geht es um den Import einer Audio-Datei als Header-Datei, die zu einem Arduino-Sketch hinzugefügt werden kann. Ein ATmega328 verfügt über 32 KB Flash-Speicher, wovon ein Teil zum Speichern der Audiodaten genutzt wird. Mit der Library Progmem aus Atmels Toolchain kann man größere Datenblöcke in den Speicher laden. 8-bit-Audiodaten bestehen aus nichts anderem als einer Folge von Bytes. Man deklariert sie so:

```
const unsigned char sounddata_data[] PROGMEM = {128,
128, 128,
[...],
69, 62, 59, 57, 52, 50, 56, 65, 74, 86, 96, 109, 116, };
```

Ich habe auch ein Tool für Arduinos IDE gebaut, das WAVE-Dateien öffnen und direkt als Teil des Codes importieren kann. Über den angegebenen Link bei den Weblinks kann man es downloaden, zum eigenen Sketchbook-Verzeichnis hinzufügen und dort entpacken. Das Tool erzeugt folgenden neuen Ordner im Sketchbook: tools/SoundData. Nach Neustart der IDE sieht man einen neuen Eintrag im Tools-Menü mit der Bezeichnung SoundData. Ein Klick darauf bringt einen Dialog zum Vorschein, mit dem man eine WAVE-Datei auswählen kann (siehe Bild 3). Der mit Generate Code beschriftete zweite Knopf öffnet die WAVE-Datei, überprüft ihre Codierung und fügt einen neuen Reiter zum Code hinzu, der mit sounddata.h beschriftet ist. Diese neue Datei enthält alles, was zum Abspielen der WAVE-Datei als 1-bit-Sound nötig ist. Die Datei wird so aussehen:

```
// soundData for your Arduino sound project
// automatically generated data
// - original sound file:/development/tmp/matis.wav
// - sampleRate:8000
// - encoding:8 bits
// - channels:1

const int sounddata_length = 7969;

const signed char sounddata_data[] PROGMEM = {127, 127,
127, 127, 127, 127,
[...],
69, 62, 59, 57, 52, 50, 56, 65, 74, 86, 96, 109, 116, };
```

Aber bitte erst weiterlesen, bevor der Knopf Generate Code im Dialog betätigt wird, denn das war noch nicht alles!

Der Sound-Player

Die Wiedergabe aufgenommener Audiodaten ist keine Kleinigkeit, da Vieles an dem, mit dem der Arduino-Core für gewöhnlich arbei-



Bild 3. Dialog zum Arduino-Sketch-Import von WAVE-Dateien in Form von Header-Dateien.

tet, geändert werden muss. Es gibt zwar Libraries für Arduino Uno, die all diese Komplexität im Umgang mit Audiodaten verstecken, doch würde man damit eine Gelegenheit verpassen zu lernen, wie das auf Low-Level-Ebene funktioniert (siehe Bild 4).

Das Verfahren, mit dem Arduino Audiosignale erzeugen kann, basiert auf der so genannten „Fast PWM“, ein Feature von Atmels ATmega-Chips. Auch andere Controller haben so etwas, aber Arduino Uno verwendet eben einen AVR-Controller. Es gibt ein Register, das es erlaubt, PWM mit einer Taktrate zu generieren, die immerhin dem halben Arbeitstakt entspricht. Aus diesem Grund kann man sehr gut Sound mit einem einzigen Pin erzeugen. Die einzige Limitierung für dieses schnelle Verfahren ist, dass es nur eine 8-bit-PWM liefert. Das ist der Grund, warum die Audiodaten in ein 8-bit-Format codiert werden müssen.

Damit das klappt, müssen zwei der drei internen Timer-Register angesprochen werden:

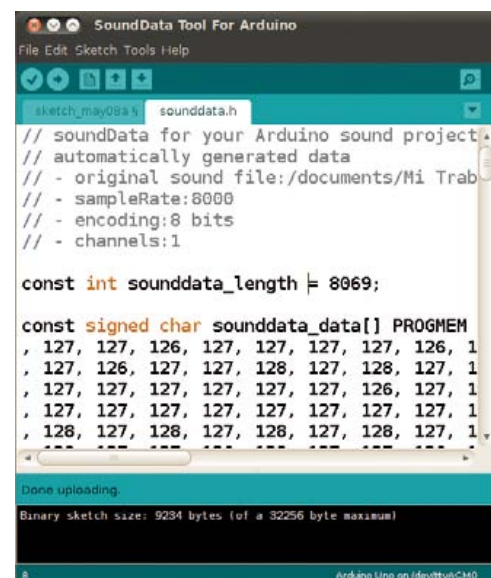


Bild 4. Darstellung der Datei mit „ta-te-ti-to-tu“ in der Arduino-IDE nach Import mit dem SoundData-Tool.

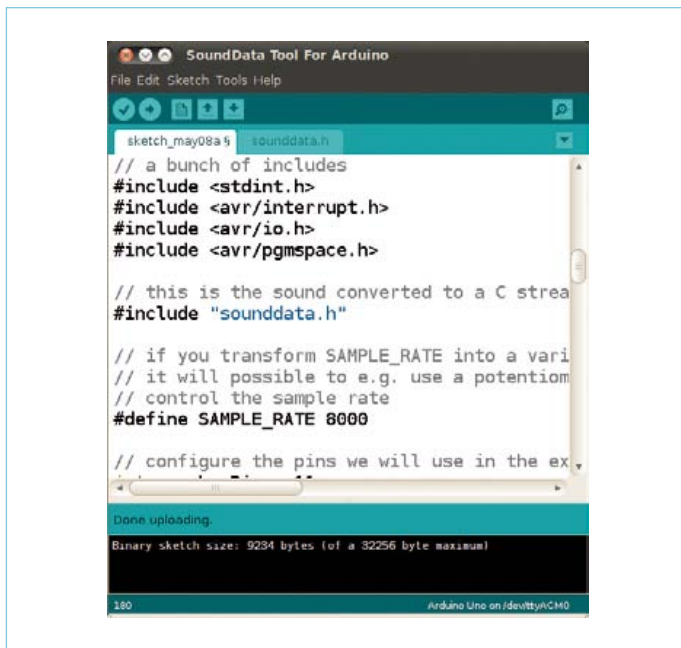


Bild 5. Screenshot des Sound-Players nach der automatisierten Code-Erstellung.

Der erste Takt operiert mit `SAMPLE_RATE` und dient zum Einlesen des nächsten Wertes aus dem Array `sounddata_data`. Dieser Wert bestimmt das Tastverhältnis für das PWM-Signal an Pin 11. Der zweite Timer wird mit 62.500 Hz getaktet (16 MHz / 256), also viel schneller als die Abtastrate des PWM-Signals (`SAMPLE_RATE` = 8 kHz). Wie schon erklärt, erhält man auf diese Weise ein Spektrum, das dem des originalen Signals sehr ähnelt. Je höher die Abtastfrequenz, desto näher kommt das Resultat dem Original. In vielen Anwendungen reichen aber 8 kHz gut aus, um akzeptablen Sound zu erzeugen.

Hinweis: Wenn man diesen Code zum Arduino-Sketch hinzufügt, werden einige Basis-Befehle der Arduino-Sprache überschrieben. PWM wird dann z.B. an einigen Pins nicht mehr funktionieren. Außerdem wird sich die Funktion `delay()` anders verhalten, als man das erwarten würde. Diese Wirkungen betreffen dann aber nur diesen Sketch.

Dieses Verfahren wurde zum ersten Mal auf der „Arduino World 2008“ gezeigt. Michael Smith veröffentlichte ein Beispiel auf Arduino-Basis, das den Mac-Startsound beim Booten abspielt, wenn man ein Reset bei einem Arduino-Diecimila-Board auslöst. Michaels Code basierte wiederum auf der Arbeit von vielen anderen Entwicklern, die in [3] aufgeführt sind.

Das SoundData-Tool erzeugt nicht einfach nur Audiodaten, sondern gleich den Code für das Sound-Playback-Beispiel im Sketch. Der Sketch sollte leer sein beim Aufruf, da alles in der IDE überschrieben wird.

Das Programm zum Abspielen von 1-bit-Sound (**Bild 5**) enthält viele Low-Level-Befehle, um die Timer zu überschreiben. Nachfolgend die Beschreibung Schritt für Schritt. Den Anfang macht `TIMER1`:

```
ISR(TIMER1_COMPA_vect) {
    if (sample >= sounddata_length) {
        if (sample == sounddata_length + lastSample) {
```

```
        // this is the condition of reaching the last
        sample
        stopPlayback();
    } else {
        // Ramp down to zero to reduce the click at
        the end of playback.
        OCR2A = sounddata_length + lastSample
        - sample;
    }
} else {
    // OCR2A is the register in memory that will
    push
    // PWM at high frequency to pin 11
    // pgm_read_byte reads data out arrays stored in
    program memory
    OCR2A = pgm_read_byte(&sounddata_data[sample]);
}

// increase the sample count
++sample;
}
```

ISR ist die Bezeichnung für die Interrupt-Service-Routine des Mikrocontrollers. Ein Interrupt ist ein Ereignis, das den Controller dazu veranlasst, die aktuelle Arbeit erst einmal ruhen zu lassen und sich zunächst um das Ereignis zu kümmern. Prozessoren können interne und externe Interrupts haben. Interne werden z.B. durch Timer verursacht - externe beispielsweise durch Pegeländerungen an bestimmten Pins. Diese ISR-Funktion kümmert sich um `TIMER1`-Events. Bei jedem Feuern von `TIMER1` (Timer-Vergleichswert erreicht) erledigt die Funktion folgendes: Sie

- inkrementiert den Adresszähler der Sound-Daten;
- überprüft, ob das Ende der Sound-Daten erreicht ist;
- wenn noch nicht am Ende, dann wird das nächste Byte des Arrays geladen;
- wenn schon am Ende, dann wird das Audiosignal auf null gefahren.

Sowohl `TIMER1` als auch `TIMER2` werden in der Funktion `startPlayback` initialisiert:

```
void startPlayback()
{
    pinMode(speakerPin, OUTPUT);

    // Set up Timer 2 to do pulse width modulation on
    the speaker
    // pin.

    // Use internal clock (datasheet p.160)
    ASSR &= ~(_BV(EXCLK) | _BV(AS2));
```

```

// Set fast PWM mode (p.157)
TCCR2A |= _BV(WGM21) | _BV(WGM20);
TCCR2B &= ~_BV(WGM22);

// Do non-inverting PWM on pin OC2A (p.155)
// On the Arduino this is pin 11.
TCCR2A = (TCCR2A | _BV(COM2A1)) & ~_BV(COM2A0);
TCCR2A &= ~(_BV(COM2B1) | _BV(COM2B0));

// No prescaler (p.158)
TCCR2B = (TCCR2B & ~(_BV(CS12) | _BV(CS11))) |
_BV(CS10);

// Set initial pulse width to the first sample.
OCR2A = pgm_read_byte(&sounddata_data[0]);

// Set up Timer 1 to send a sample every interrupt.
cli();

// Set CTC mode (Clear Timer on Compare Match)
(p.133)
// Have to set OCR1A *after*, otherwise it gets
reset to 0!
TCCR1B = (TCCR1B & ~_BV(WGM13)) | _BV(WGM12);
TCCR1A = TCCR1A & ~(_BV(WGM11) | _BV(WGM10));

// No prescaler (p.134)
TCCR1B = (TCCR1B & ~(_BV(CS12) | _BV(CS11))) |
_BV(CS10);

// Set the compare register (OCR1A).
// OCR1A is a 16-bit register, so we have to do this
with
// interrupts disabled to be safe.
OCR1A = F_CPU / SAMPLE_RATE;    // 16e6 / 8000 =
2000

// Enable interrupt when TCNT1 == OCR1A (p.136)
TIMSK1 |= _BV(OCIE1A);

lastSample =
pgm_read_byte(&sounddata_data[sounddata_length-1]);
sample = 0;
sei();
}

```

Auch wenn die Low-Level-Befehle im Code kommentiert sind, ist eine kleine Erläuterung vielleicht doch hilfreich:

- Definition des Pins für das PWM-Signal als Ausgang.
- Konfiguration des Boards zwecks Verwendung des internen Takts.
- Initialisierung des Fast-PWM-Modes.
- Konfiguration von TIMER2 als PWM-Generator für die Sound-Ausgabe. Das PWM-Tastverhältnis wird durch Register OCR2A bestimmt.



NEW

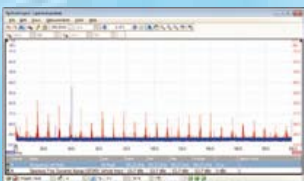
PicoScope®

4-KANAL

PC Oszilloskope



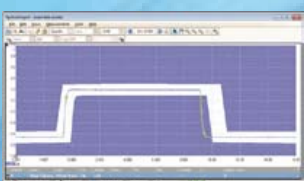
60 - 200 MHz Bandbreite
128 MS Speichertiefe
Echtzeit-Abtastung mit 1 GS/s
Erweiterte digitale Triggerung
Spektrumanalysator mit 200 MHz
Funktionsgenerator und AWG



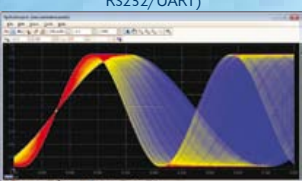
Spektrumanalysator. Vollständiges SDK und erweiterte trigger



Serielle Entschlüsselung (CAN, LIN, FlexRay, I²C, SPI und RS232/UART)



Maskengrenzprüfung, messungen und rechenkanäle



Erweiterte Anzeigemodi, Zuverlässige Signalintegrität

ALLE MODELLE WERDEN MIT TASTKÖPFEN, KOMPLETTER SOFTWARE UND 5 JAHRE GARANTIE GELIEFERT.

www.picotech.com/PS176

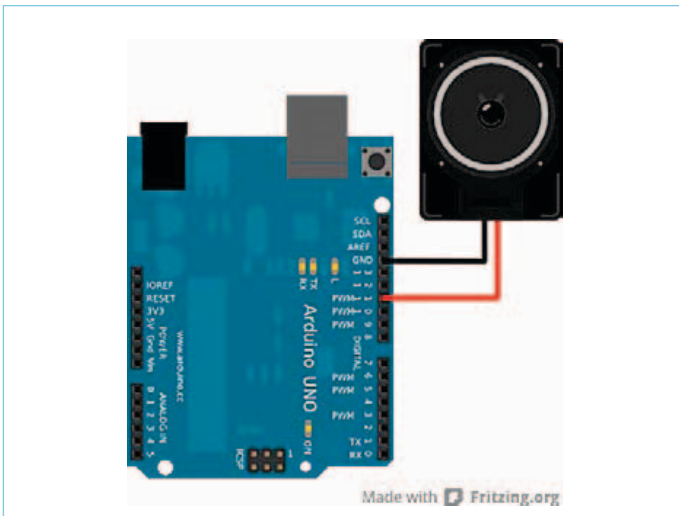


Bild 6. Arduino Uno ist gut dazu geeignet, direkt einen kleinen Lautsprecher zu treiben.

- Laden des ersten Bytes des Audiosignals in OCR2A.
- Stopp der Interrupts für 1 s – cli() – sodass TIMER1 ohne Störungen konfiguriert werden kann.
- Konfiguration des Takts via TIMER1 für das Holen des nächsten Bytes.
- Freigabe der Interrupts – sei().

Ganz analog braucht es noch eine Funktion, welche die Timer am Ende der Sound-Daten stoppt:

```
void stopPlayback()
{
    // Disable playback per-sample interrupt.
    TIMSK1 &= ~_BV(OCIE1A);

    // Disable the per-sample timer completely.
    TCCR1B &= ~_BV(CS10);

    // Disable the PWM timer.
    TCCR2B &= ~_BV(CS10);

    digitalWrite(speakerPin, LOW);
}
```

Auf diese Weise hat man einige Funktionen zur Verfügung, mit denen man in einem eigenen Programm Sound ausgeben kann. In diesem Beispiel wird startPlayback() schon beim Setup aufgerufen. Dadurch wird der Sound nur einmal abgespielt.

```
void setup()
{
    startPlayback();
}

void loop()
{
    // do nothing
}
```

Schlusswort

Dieser Artikel ist eine Einführung in die Möglichkeiten der Ausgabe von Audiosignalen mit einem Arduino-Board. Es wurde gezeigt, wie man mit grundlegenden Arduino-Funktionen Sound generieren kann. Es wurden auch Libraries beschrieben, welche die Erzeugung einfacher Melodien vereinfachen, die dann über preiswerte Piezo-Buzzer in Schall umgesetzt werden. Zum Schluss gab es noch eine Demonstration, wie man per Fast-PWM 1-bit-Sound erzeugen kann. Alle Code-Schnipsel und die Tools mit einigen fertigen Sound-Dateien stecken in einer Zip-Datei [4]. Ich habe auch ein neues Tool für die Arduino-IDE erstellt, das beim Import kurzer WAVE-Dateien in den Programmspeicher von Arduino hilft. Mit diesem Tool kann man Sound-Dateien laden, wiedergeben, ihre Abtastrate ändern, sie rückwärts abspielen oder sie verzerren. Hardwaremäßig wird der Lautsprecher wie in **Bild 6** angeschlossen.

Das war noch nicht alles! Es gibt noch eine Menge zu entdecken - z.B. könnte man Pin 3 zusammen mit Pin 11 für die Ausgabe von Stereo-Sound einsetzen. Auch 8-bit-Synthesizer mit der Möglichkeit des Mischens von vier Sound-Kanälen in einen Kanal wären möglich. Oder wie wäre es, wenn das Arduino-Board als MIDI-Sound-Karte fungieren würde?

Viel Spaß beim Hacken. Bis nächsten Monat...

(120427)

Weblinks

- [1] **Sigma-Delta-DACs mit 1 bit:**
www.digitalsignallabs.com/presentation.pdf
- [2] **Audacity, das Open-Source-Audio-Tool:**
www.audacity.sourceforge.net
- [3] **Referenzen und Beispiele von Michael Smith:**
Arduino-Referenz für die Tone-Library: www.arduino.cc/en/Reference/Tone
Original-Artikel zum Arduino-Playground:
www.arduino.cc/playground/Code/PCMAudio
www.uchobby.com/index.php/2007/11/11/arduino-sound-part-1/
www.atmel.com/dyn/resources/prod_documents/doc2542.pdf
www.evilmadscientist.com/article.php/avrdac
www.gonium.net/md/2006/12/27/i-will-think-before-i-code/
http://bit.ly/P-Inch_ProgrammingPCspeaker
- [4] **Beispiele, Code und Tools zu diesem Artikel:**
www.elektor.de/120427

Von den Machern von Elektor!



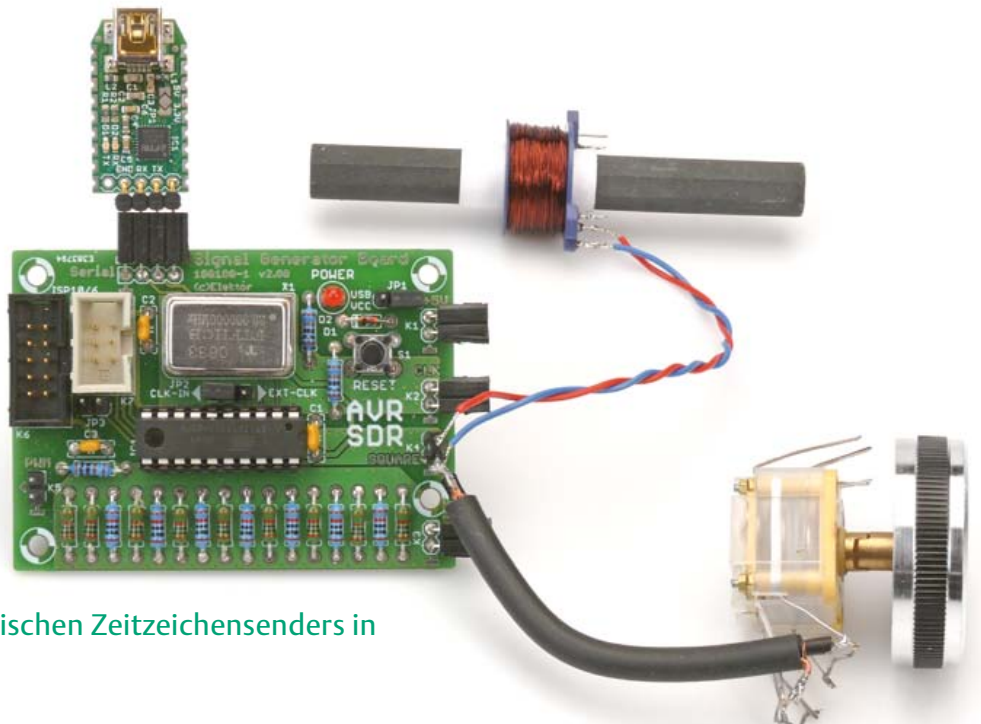
Oder frei Haus unter www.elektor.de/leds4 bestellen!

SDR mit AVR (5)

Decodierung von DCF77, MSF und TDF162: Vom IIR zum Matched Filter

Von Martin Ossmann (D)

Dass sich die beliebten AVR-Controller auch für die digitale Signalverarbeitung eignen, soll diese Serie zeigen. Diesmal verwenden wir verschiedene Decodierungsmethoden und Filter, um die Signale je eines deutschen, britischen und französischen Zeitzeichensenders in digitale Daten zu verwandeln.



In der letzten Folge [4] haben wir uns mit dem digitalen Rundfunkempfang beschäftigt. Mit dem dort beschriebenen digitalen Empfänger wurden Zeitzeichensender wie der DCF77 und Wetterdienste nicht nur empfangen, wir haben auch die Signale decodiert. Jetzt geht es richtig los mit unterschiedlichen Decodierungen. Nach dem Senden und Empfangen von RTTY-Signalen geht es über den deutschen DCF77 zum britischen MSF und dem französischen TDF162. Zum Schluss wird die Anwendung eines „Matched Filters“ zur Bit-Decodierung erläutert.

Drahtlose Datenübertragung auf 125 kHz

Wenn das DDH-Signal (siehe letzter Abschnitt der vierten Folge) nicht stark genug empfangen wird, kann man sich natürlich einen eigenen Testsender aufbauen. Die Hardware ist genau die gleiche wie beim DCF-Testsender in Teil 3 [3]. Man kombiniert einfach eine Ferritantenne und einen Drehkondensator zu einem Serienschwingkreis, den man vom Rechteckausgang des Signalgenerators

speist. Die Anordnung sieht dann aus wie im Bild zu Anfang des Artikels.

Als Software im Sender kommt das Programm „EXP-SQTX-FM-RTTY-V01.c“ zum Einsatz. Die Empfangssoftware „EXP-125kHz-RTTY-RX-V01.c“ muss man nun nur noch auf 125 kHz konfigurieren, und schon kann man Sende/Empfangsversuche starten. Der Autor konnte mit diesem einfachen System eine Reichweite von mehr als 5 m selbst durch Wände hindurch erreichen. Das System würde sich also auch prima als induktive Fernsteuerung eignen. Mit der gleichen Software kann man übrigens auch die klassischen RTTY-Signale in den Amateurfunk-Bändern decodieren. Hierzu muss man lediglich die Filter und das Timing an die verwendeten Übertragungsparameter anpassen. Testweise kann man RTTY-Signale auch mit einer passenden PC-Software wie zum Beispiel MMtTy (Download unter [6]) mit Hilfe der Soundkarte erzeugen.

DCF77-Decodierung

In Bild 3 von Folge 3 war schon das Oszillogramm des demodulierten

Signals beim DCF77-Empfang zu sehen. Nun zum Vorgehen, wie man die in den Austastimpulsen steckende Information zurückgewinnt: Da die Amplitude des Empfangssignals je nach Empfangslage stark unterschiedlich ausfallen kann, muss man zuerst möglichst automatisch einen Schwellwert bestimmen, anhand dessen unterschieden wird, ob die Amplitude gerade abgesenkt ist oder nicht. Als Schwellwert soll der Einfachheit halber der Mittelwert des Signals dienen. Dabei muss allerdings über längere Zeit gemittelt werden. Würde man die Mittelung mit einem CIC-Filter machen, wäre ein sehr großer Zwischenpuffer erforderlich. Aus diesem Grund wird ein sogenanntes IIR-Filter [7] verwendet.

IIR-Filterung

Das Akronym „IIR“ steht hierbei für Infinite Impulse Response. Die lange Impulsantwort (entspricht der Mittelungszeit) wird dabei durch ein rekursiv arbeitendes Filter erreicht. Die Berechnungsvorschrift für ein einfaches IIR-Filter lautet:

$$y_{k+1} = ax_k + (1-a)y_k$$

Dabei ist y_k die Ausgangswertefolge, x_k ist die Eingangswertefolge, und a ist eine Zahl kleiner als 1, aber nahe bei 1. Der neue Wert y_{k+1} ergibt sich als gemittelte Summe aus altem Wert und neuem Eingangswert. Bei dieser Berechnung sind die Werte a und $1-a$ ja normalerweise Zahlen kleiner als 1. Daher ist man leicht dazu geneigt, Fließkommaoperationen zur Berechnung des Filters zu verwenden. Das ist aber viel zu aufwendig. In unserer Realisierung (siehe **Listing 1**) werden die Filterkoeffizienten als Bruch dargestellt, wobei der Nenner mit 65536 eine Zweierpotenz ist.

In diesem Fall ist $a = (65536 - 50) / 65536 = 0,999237...$ und $1-a$ ergibt demnach einen Wert von $0,000763...$ Die Berechnungen werden mit 32 bit breiten Integer-Zahlen durchgeführt. Die Division durch 65536 wird vom Compiler einfach durch die Auswahl der höherwertigen 16 bit ersetzt, was das Rechentempo ziemlich beschleunigt. Auf diese Weise wird das IIR-Filter höchst effizient realisiert. Damit liegt das Signal direkt in binärer Form vor. Durch Auswertung der Impulslängen erhält man schließlich die Uhrzeit.

Für den DCF77-Empfang mit Decodierung benötigt man das einfache Frontend oder den universellen Empfänger, zusammen mit der aktiven Ferritantenne, abgestimmt auf 77,5 kHz. Mit dem Programm „EXP-DCFdecode-RX-V01.c“ im Empfänger-Controller erhält man einen Sekunden-Timer und das Signal wird via PWM-DAC ausgegeben. Die Feldstärke und die Uhrzeit werden dabei auf dem LCD angezeigt und die decodierten Daten über RS232 bzw. USB seriell übermittelt.

MSF-Decodierung

Es gibt in Europa noch weitere Zeit- und Normalfrequenzsender. Einer davon ist der auf 60 kHz sendende MSF [8] aus Großbritannien. Dieser ist am Wohnort des Autors nur relativ schwach zu empfangen, weshalb ein etwas höherer Aufwand nötig ist. Beim MSF wird die Amplitude des Trägersignals jede Sekunde abgesenkt (siehe **Bild 1**). Zu Beginn einer neuen Minute wird der Träger für 0,5 s abgesenkt. Zu Beginn jeder weiteren Sekunde wird

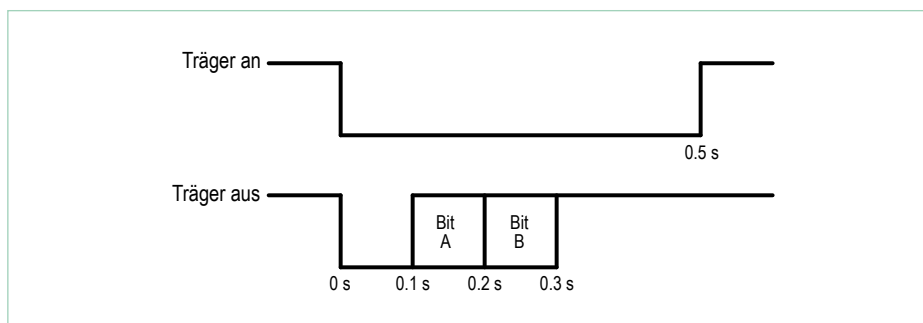


Bild 1. Trägerabsenkung beim MSF-Sender.

Listing 1: IIR-Filter

```
#define ALPHA 50
#define BETA 65536
int32_t Threshold ;

Threshold=(ALPHA*(int32_t)Amplitude+(BETA-ALPHA)*Threshold)/BETA ;
if (Amplitude>Threshold) {
    DCFlevel=0 ;
}
else {
    DCFlevel=1 ;
}
```

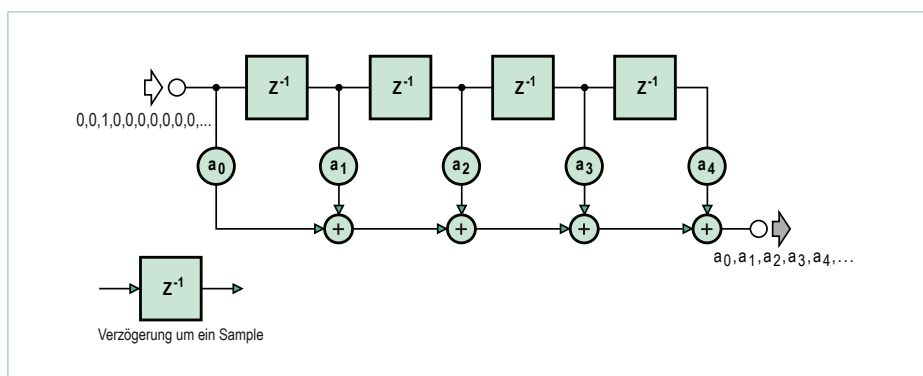


Bild 2. Signalfluss in einem FIR-Filter.

der Träger zunächst für 0,1 s abgesenkt. Die nächsten 0,1 s bilden den Zeitschlitz für das A-Bit. Ist in diesem Zeitfenster der Träger abgesenkt, ist das A-Bit „high“. Die daran anschließenden nächsten 0,1 s bilden den Zeitschlitz für das B-Bit. Ist dort die Amplitude niedrig, wird ebenfalls das B-Bit „high“. A- und B-Bits dienen zur Übertragung von Zeitinformation, zusätzlicher Synchronisation und Fehlererkennung.

Erste Empfangsversuche ergaben ein recht heftig gestörtes Signal, so dass einfache CIC-Filter eine nicht ausreichende Filterwirkung zeigten. Es war also ein besseres Filter mit

einer Grenzfrequenz von etwa 10 Hz und starker Dämpfung notwendig. Eine weitere Anforderung an das Filter ist, dass es die Kurvenform nicht zu stark beeinflussen sollte. Diese Eigenschaften kann man mit einem FIR-Filter [9] genügender Ordnung erreichen.

Zum MSF-Empfang benötigt man neben dem Empfänger die auf 60 kHz abgestimmte aktive Ferritantenne und die Software „EXP-RX-MSF60decode-V01.c“. Damit werden Feldstärke und Uhrzeit auf dem LCD angezeigt und die Daten samt Uhrzeit seriell ausgegeben. Das FIR-Filter fünfter Ordnung ist wie in **Bild 2** aufgebaut.

Listing 2: FIR-Summation

```
#define FIRLength 60

prog_int16_t FIRCoeffs []={
    112,    3,   -17,
    -52, ..... 112 } ;

FIRSum=0 ;
j=FIRPointer ;
for (k=0 ; k<FIRLength ; k++ ) {
    FIRcoef=pgm_read_word( FIRCoeffs1 + k ) ;
    FIRSum += (int32_t)FIRcoef*(int32_t)FIRBuffer[j] ;
    j++ ; if (j==FIRLength) { j=0 ; }
}
return FIRSum/0x10000L ;
```

Die Eingangswerte laufen durch einen Pufferspeicher. Der Ausgangswert ergibt sich durch die mit den Koeffizienten k gewichtete Summe der Werte im Buffer.

In **Listing 2** ist das Ganze als C-Programm realisiert. Auch hier wird wieder die schnelle Ganzzahlarithmetik mit abschließender Division genutzt. Diese Filterroutine filtert das Signal, das mit einer Datenrate von 250 Samples/s registriert wird. Die Rechenzeit ist selbst bei einer Ordnung von 60 mit 300 μ s ausreichend kurz, da hierfür sogar 4 ms an Zeit zur Verfügung stehen.

Bild 3 zeigt die Impulsantwort und den Frequenzgang des Filters. Das Filter fängt bei ca. 8 Hz an stark zu dämpfen; oberhalb von 15 Hz hat es eine Dämpfung von etwa 50 dB.

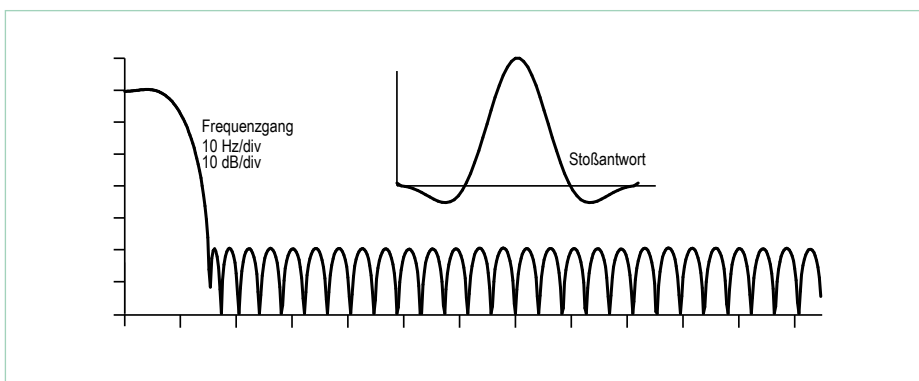


Bild 3. Impulsantwort und Frequenzgang des FIR-Filters.

In **Bild 4** sieht man das demodulierte und gefilterte MSF-Signal (gelbe Spur). Etwa in der Mitte des Bildes sieht man den 0,5 s dauernden Minutenimpuls. Zwei Sekunden vor dem Minutenimpuls ist ein Impuls zu sehen, bei welchem $A = 1$ und $B = 0$ ist. Nach dem Minutenimpuls sieht man einen Impuls mit $A = 0$ und $B = 1$. Danach kommen dann mehrere Impulse mit $A=0$ und $B=0$. Unter der gelben Spur sieht man eine sägezahnförmig verlaufende blaue Spur. Anhand dieses Signals kann man das Timing verfolgen.

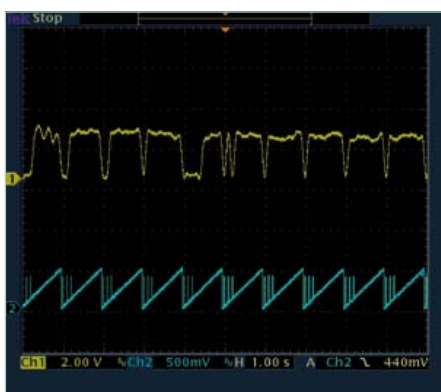


Bild 4. Demoduliertes und gefiltertes MSF-Signal mit blauer Timing-Spur.

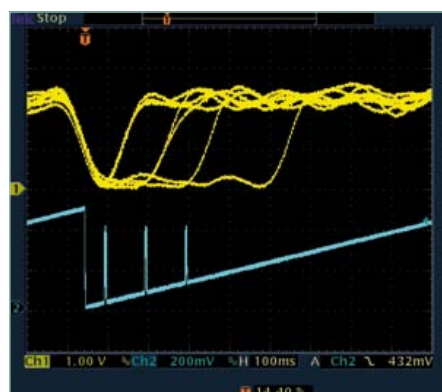


Bild 5. Die drei Impulse zeigen, an welcher Stelle das Signal ausgewertet wird.

Genauer sind die Verhältnisse in **Bild 5** dargestellt. In der Software gibt es eine Variable namens „SecondTimer“, die immer wieder von 0 bis 250 mit einem Takt von 250 Hz zählt. Der Takt entspricht der Abtastrate für das demodulierte Signal. Wenn der Minutenimpuls detektiert wird, wird SecondTimer zurückgesetzt, so dass das Hochzählen immer am Anfang einer Sekunde beginnt. Die Variable SecondTimer wird dann von einem der PWM-DACs ausgegeben. Die drei blauen Impulse zeigen die Stelle, wo die Software das Signal auswertet. Hier wird festgestellt, welche Werte das A- und das B-Bit haben. Sind alle Bits gesammelt, werden diese ausgewertet und die Zeitinformation per LCD ausgegeben. In **Listing 3** ist ein Beispiel ausgegebener Daten zu sehen.

Listing 3: Beispieldaten des Senders MSF

<2000000000000000000010000000110010000010001110001001011111103>	07:09	08.03.10	MONDAY	P=1	P=1	P=1	P=1
<2000000000000000000010000000110010000010001110010000011113103>	07:10	08.03.10	MONDAY	P=1	P=1	P=1	P=1
<2000000000000000000010000000110010000010001110010001011111103>	07:11	08.03.10	MONDAY	P=1	P=1	P=1	P=1

Decodierung von France Inter (TDF)

Auch der Sender France Inter (TDF) sendet Zeitinformationen aus. Dabei verwendet er eine Phasenmodulation. Die Signale für „low“ und „high“ sind in **Bild 6** dargestellt. Die Information ist in diesem Datenstrom gleich wie beim DCF77 codiert. Bei der Decodierung tritt hier aber ein neues Problem auf: Zu Beginn jeder Sekunde wird das Zeit-Code-Bit gesendet, aber danach kommen noch jede Menge an Phasenmodulationssignalen, die für interne Zwecke verwendet werden. Daher muss ein Empfänger zuerst die genaue Lage der Sekundenpulse ermitteln. Vor dem zur Sekunde „0“ gehörenden Sekundenimpuls kommt eine lange Modulationspause, da auch der Impuls „59“ fehlt. Hat der Empfänger diese Pause detektiert, sucht er das nächste Maximum im Phasenverlauf (siehe Bild 6). Das Maximum liegt genau beim gesuchten Sekundenimpuls. Von da an wird immer im Abstand von exakt einer Sekunde geprüft, ob eine „1“ oder eine „0“ empfangen wurde.

Matched Filter mit CIC

Will man direkt die Phase auswerten, darf die Phase natürlich nicht driften. Die Folge dieser Forderung ist, dass man die Phase mit einer PLL stabilisieren muss. Da dies etwas aufwändig ist, und Regelkreise instabil sein können, ist ein anderes Konzept geeigneter. Anstelle der Phase wird einfach die Frequenz ausgewertet. Die Frequenz generiert man aus der Phase durch die Differenzbildung aufeinanderfolgender Werte. Diese Werte filtert man anschließend noch mit einem CIC-Tiefpassfilter, damit die Signalform sauberer wird. Um zu ermitteln, ob „0“ oder „1“ gesendet wurde, wird geprüft, ob das zu „1“ gehörende Signal zum richtigen Zeitpunkt empfangen wurde. Falls dies nicht der Fall sein sollte, entscheidet der Empfänger, dass es sich um „0“ handelt.

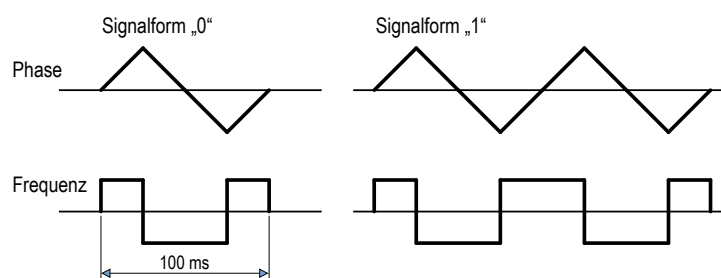


Bild 6. TDF162-Signalformen.

Nun stellt sich die Frage, ob tatsächlich die vorgegebene Signalform empfangen wurde. In der Nachrichtentechnik setzt man dazu ein sogenanntes „Matched Filter“ ein. Es besitzt als Impulsantwort genau die Signalform, nach der gesucht wird (die Impulsantwort ist lediglich zeitlich gespiegelt). Unser „0“-Signal besteht aus einer Kombination von drei Rechteck-Impulsen. Aus diesem Grund kann man ein Matched Filter wieder einfach mit einem CIC-Filter realisieren (die Impulsantwort, also die Antwort auf eine einzelne „1“ am Eingang, ist bei diesem Filter ja ein

einzelner Rechteckpuls, siehe [4]). Dies ist in **Bild 7** schematisch dargestellt. Die Signal-Abtastung erfolgt an dieser Stelle mit einer Rate von 500 Samples/s. Der erste Rechteck-Impuls ist 25 ms lang und entspricht daher zwölf Samples. Der zweite Impuls ist 50 ms lang und kommt daher auf 25 Samples. Der letzte Impuls ist wie der erste wieder 25 ms lang. Aus den Impulsbreiten ergeben sich direkt die nötigen Längen der Verzögerungsstufen. Mit diesem Filter lässt sich die „0“-Signalform optimal detektieren. Da das „1“-Signal in der Form dem „0“-Signal gleicht (zwei „0“-Signale

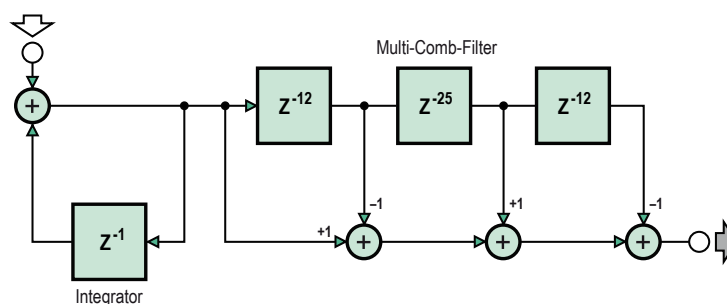


Bild 7. Ein „Matched Filter“, realisiert als CIC-Filter.

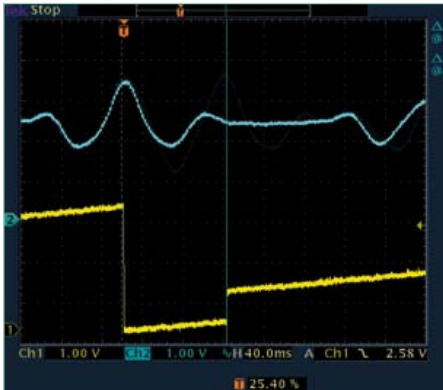


Bild 8. Ausgangssignal des Matched Filters bei Empfang von „0“.

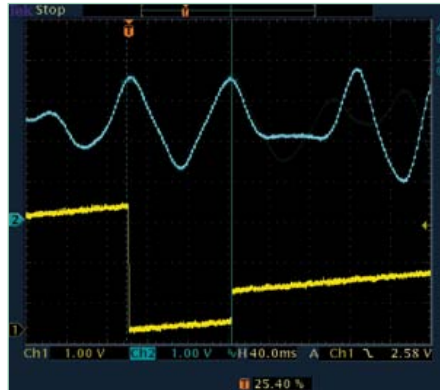


Bild 9. Ausgangssignal des Matched Filters bei Empfang von „1“.

werden einfach hintereinander gesendet), kann man das gleiche Filter verwenden und schaut dann einfach 50 ms später nach. Genau das ist die von unserem Empfänger verwendete Strategie.

Die Oszillogramme (**Bilder 8 und 9**) zeigen die entsprechenden Signalformen. Die gelbe Spur zeigt den Timer, der die Lücke bei der Sekunde „59“ detektiert. Das Eingangssignal wird an den Stellen

abgetastet, an welchen das gelbe Signal springt. Genau dort liegen jeweils die Signalmaxima.

Die Nullen und Einsen werden in einem Feld gespeichert. Diese Daten werden dann nach Empfang des letzten Bits decodiert und im LC-Display angezeigt. Gleichzeitig werden sie mit 19.200 Bd an der seriellen Schnittstelle ausgegeben. Ein Beispiel empfangener Daten ist in **Listing 4** dargestellt.

Im nächsten Heft folgt der letzte Teil der Serie, dann werden wir uns unter anderem mit Bittakt-Synchronisation und dem „Early Late Gate Synchronizer“ beschäftigen.

Abschließend werden wir den Sender BBC198 decodieren und dabei noch einige weitere Decodier-Verfahren kennenlernen.

(120089)

Listing 4: Beispieldaten von TDF162

TDF 162

```
000111000000000000101100110010001001111010110010000000100000: 08:19 17.02 WEDNESDAY 2010 p000
000011000000000000010100001010001001111010110010000000100000: 08:20 17.02 WEDNESDAY 2010 p000
000011000000000000010110001000001001111010110010000000100000: 08:21 17.02 WEDNESDAY 2010 p000
00001100000000000001010001000001001111010110010000000100000: 08:22 17.02 WEDNESDAY 2010 p000
0001110000000000000101110001010001001111010110010000000100000: 08:23 17.02 WEDNESDAY 2010 p000
```

Weblinks

- [1] www.elektor.de/100180
- [2] www.elektor.de/100181
- [3] www.elektor.de/100182
- [4] www.elektor.de/120088
- [5] www.elektor.de/120089

- [6] <http://hamsoft.ca/pages/mmtty.php>
- [7] www.dspguru.com/dsp/faqs/iir/basics
- [8] www.npl.co.uk/science-technology/time-frequency/time/products-and-services/msf-radio-time-signal
- [9] www.dspguru.com/dsp/faqs/fir/basics

Elektor Produkte & Service

- Signalgenerator (Kit mit Platine und allen Bauteilen 100180-71)
- Universeller Empfänger (Kit mit Platine und allen Bauteilen 100181-71)
- Aktive Ferritantenne (Kit mit Platine und allen Bauteilen 100182-71)

- Kombi-Kit aus allen drei Bauteilsätzen plus USB/TTL-Konverter BOB FT232: 100182-72
 - USB/TTL-Konverter BOB FT232, bestückt und getestet 110553-91
 - Gratis Software-Download (Hex-Files und Source-Code)
- Alle Produkte und Downloads sind über die Website zu diesem Artikel erhältlich: www.elektor.de/120089

elektorLive! 2012

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

Am 20. Oktober 2012 findet im Kongresszentrum Westfalenhallen Dortmund die erste **elektorLive!**-Veranstaltung statt.

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

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

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

Geplante Seminare (Änderungen vorbehalten):

- Strahlungsmesser: Radioaktivität messen mit Halbleitern
- Röhrenverstärker: Grundlagen, Schaltungen, Messungen
- Fahrzeugdiagnose mit OBD2
- Elektronik steuern mit Android
- FPGA für Einsteiger
- Embedded Linux
- Neues aus der AVR-Welt: XMEGA und Atmel Studio 6
- Simulieren mit EAGLE
- Gassensoren: Ansteuerung und Praxis

Datum:

Samstag, 20. Oktober 2012

Zeit:

09:00 Uhr – 17:00 Uhr

Ort:

Kongresszentrum
Westfalenhallen Dortmund

Eintritt:

25,00 € für Elektor-Abonnenten

45,00 € für Nicht-Abonnenten

20,00 € für Studenten/Schüler

Der Eintritt berechtigt zur Teilnahme an 2 Seminaren.

Tagesablauf:

09:00 Uhr – 17:00 Uhr: Ausstellung

10:00 Uhr – 11:30 Uhr: 1. Seminar-Runde

13:00 Uhr – 14:30 Uhr: 2. Seminar-Runde

15:00 Uhr – 16:30 Uhr: 3. Seminar-Runde



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



Quadrowalker

Kleiner vierbeiniger Roboter mit acht Servos



Der Quadrowalker ist wie geschaffen für Experimente, bei denen es um Bewegungsabläufe oder das Ausweichen vor Hindernissen geht. Spezial-Bauteile sind der eigenwilligen vierbeinigen Konstruktion fremd, acht Servo-Antriebe und ein Mikrocontroller sind die wesentlichen Komponenten. Gesteuert wird der Quadrowalker mit einer Infrarot-Fernbedienung.

Von Gert Baars (NL)

Roboter können als programmierbare Maschinen betrachtet werden, die spezifische Aufgaben ausführen. Bekannte Beispiele sind die Schweißroboter der Autoindustrie, die am Fließband Autoteile aneinander schweißen und stets gleiche Bewegungen mit gleichem Tempo bei gleichbleibender Qualität ausführen. Auch CNC-Maschinen können den Robotern zugerechnet werden. Andere Roboter wurden so konstruiert, dass sie in ihren Funktionen und im Erscheinungsbild dem Menschen nachempfunden sind. Ein bekanntes Mitglied dieser Kategorie ist der von Honda entwickelte Roboter ASIMO.

Der Quadrowalker gleicht entfernt einem Kleintier auf vier Pfoten, einem Hund oder einer Katze. Vor der Entwicklung des Konzepts wurden verschiedene Arten der Fort-

bewegung analysiert. Spinnen laufen auf sechs Beinen, zu jeder Zeit können drei Beine den Boden berühren, um das Gleichgewicht zu halten. Damit der Quadrowalker nicht zu kompliziert wird, bewegt er sich auf nur vier Beinen. Ein Gleichgewichtsorgan besitzt der Quadrowalker nicht, er kommt ohne Gyroskop oder Beschleunigungssensor aus. Auf vier Beinen ist das sichere Fortbewegen auch ohne solche inneren Organe möglich, allerdings treten dabei Momente des labilen Gleichgewichts auf. Doch bei niedriger Schritthöhe und kurzer Schrittdauer sind diese Momente nicht lang genug, um das Fortbewegen zu behindern. Das Bewegungsmuster des Quadrowalkers ist tatsächlich auch manchen Tierarten eigen. Der Körperschwerpunkt dieser Tiere, beispielsweise der Eidechsen, liegt in nur geringer Höhe über dem Boden. Auch beim Quadrowalker wurde der Schwerpunkt tief gelegt, die gewichtsträchtigen Akkus oder

Batterien befinden sich auf der Unterseite. Studien haben gezeigt, dass auch starre, gerade Beine für die Fortbewegung geeignet sind, wenn sich das einzelne Bein um zwei Achsen drehen kann. Die Bewegungsachsen müssen sich dort befinden, wo das Bein beginnt. Knie- oder Fußgelenke sind dann nicht mehr zwingend notwendig. Es entsteht eine Art Rollgelenk, mit dem das Anheben und Verschieben des Beins gleichzeitig möglich ist. Für das Ausführen einachsiger Bewegungen bietet sich der Einsatz von Servos an, denn Servos können Winkeländerungen und Vorwärtsbewegungen ausführen. Da für jedes Bein zwei Servos erforderlich sind, muss der Quadrowalker wegen seiner vier Beine von insgesamt acht Servos angetrieben werden.

Mechanik

Das Projekt „Quadrowalker“ besteht aus drei Teilen: Mechanik, Elektronik und Soft-

ware. In den Bereich der Mechanik fällt der Anbau der Beine mit den Servos an das Chassis, wobei auch die Servos miteinander gekoppelt werden müssen. Herzustellen sind diverse bügelartige Gestänge, für die Montage der vier Servos auf das Chassis sind zwei rechtwinklige Profilschienen vorgesehen. Jeweils zwei Servos werden mit einer Profilschiene so verschraubt, dass die Bewegungen rechtwinklig zueinander verlaufen. Für einige Servo-Typen sind solche Befestigungsteile im Handel erhältlich. Ferner wird für jedes Bein ein Bügel benötigt, der die mechanische Verbindung zum zweiten Servo herstellt. Das Material besteht aus Alu-Flach- und Winkelprofilen, wie sie beispielsweise in Baumärkten oder Hobbyläden erhältlich sind. Die Profile werden entsprechend gekürzt, anschließend sind diverse Löcher zu bohren. Die Skizze in **Bild 1** gibt Hilfestellung bei der Konstruktion. Es muss nicht betont werden, dass auch hier mehrere Wege zum Ziel führen. Beim Musteraufbau sind die Beine aus 4 mm starken Alu-Rundstäben gebogen. Die Beine können am Servo-Bügel befestigt werden, indem in einen Bolzen ein Loch mit 4 mm Durchmesser gebohrt wird, so dass eine Klemmverbindung entsteht. Der horizontale Teil der um 90° gebogenen Beine darf wegen der möglichst freien Beweglichkeit nicht länger als notwendig sein. Der Schwerpunkt muss vertikal möglichst tief liegen, allerdings nicht so tief, dass die Unterseite den Boden berührt.

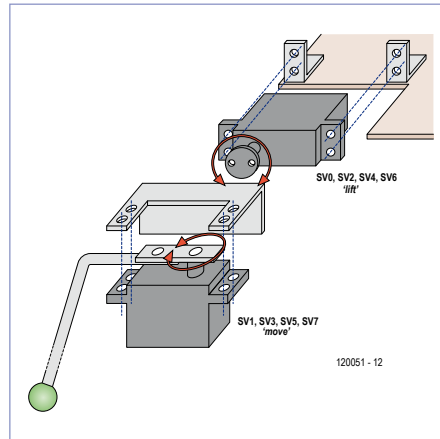


Bild 1. Die Servos werden mit Winkelprofilen befestigt.

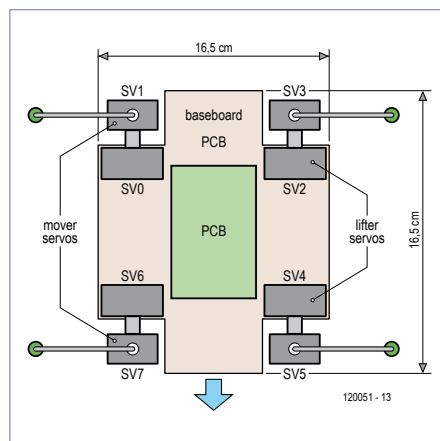
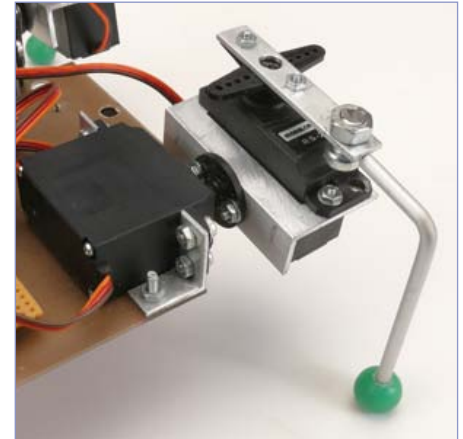
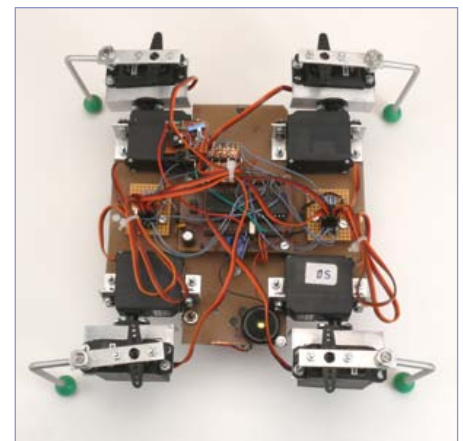


Bild 2. Mechanische Konstruktion mit den Servos.



Die Wahl des Servo-Typs hängt von der Kraft ab, die der Servo entwickeln muss. Die Kraft wird als Torsion in $\text{kg} \cdot \text{m}$ angegeben. Da beim Fortbewegen zwei Beine Bodenberührung haben, muss jeder Servo die Hälfte des Gesamtgewichts tragen können. Das Gewicht des Musteraufbaus einschließlich Stromquellen lag bei 750 g, der horizontale Teil der Beine war 6 cm lang und die Torsion betrug $2,1 \text{ kg} \cdot \text{cm}$. Servos des gängigen und preisgünstigen Typs RS-2 sind für den Antrieb des Quadrowalkers gut geeignet. Die Torsion dieses Typs beträgt bei 5 V mehr als $4 \text{ kg} \cdot \text{cm}$. Beim Quadrowalker hängt die erforderliche Torsion außer vom Gewicht hauptsächlich von der Länge des horizontalen Teils der Beine ab. Da die Stromaufnahme der Servos maßgeblich von der Tor-

sion bestimmt wird, halten die Stromquellen länger durch, wenn die Beine etwas nach innen gebogen sind.

Eine Gesamtansicht der mechanischen Konstruktion ist in **Bild 2** dargestellt. Beim Musteraufbau besteht das Chassis aus einer leeren Platine, die acht Servos befinden sich in den vier Ecken. Die Steuerung mit dem Mikrocontroller ist zentral auf dem Chassis montiert. Der Infrarot-Sensor ragt in die Höhe, was die Reichweite der Fernsteuerung steigert. Die auf der Unterseite befestigten Akkus oder Batterien und ihre Halterung müssen zentriert montiert werden, so dass der Schwerpunkt in der horizontalen Ebene symmetrisch liegt.

Elektronik

Die Steuerung der Servos zusammen mit anderen Funktionen übernimmt ein Mikrocontroller des Typs ATmega32 (siehe **Bild 3**). Eigentlich ist dieser Controller etwas überdimensioniert, doch weil er viel Programmspeicherplatz bietet, lässt sich der Quadrowalker fast nach Belieben erweitern. An acht Ausgängen stehen die pulsbreitenmodulierten Signale für die acht Servos bereit, so dass jeder Servo unabhängig gesteuert werden kann. Die Signale des Infrarot-RC5-Empfängers, die den Quadrowalker fernsteuern, liegen an einem Interrupt-Eingang. Als Fernsteuersender genügt eine TV-Fernbedienung, die nach RC5-Standard arbeitet.

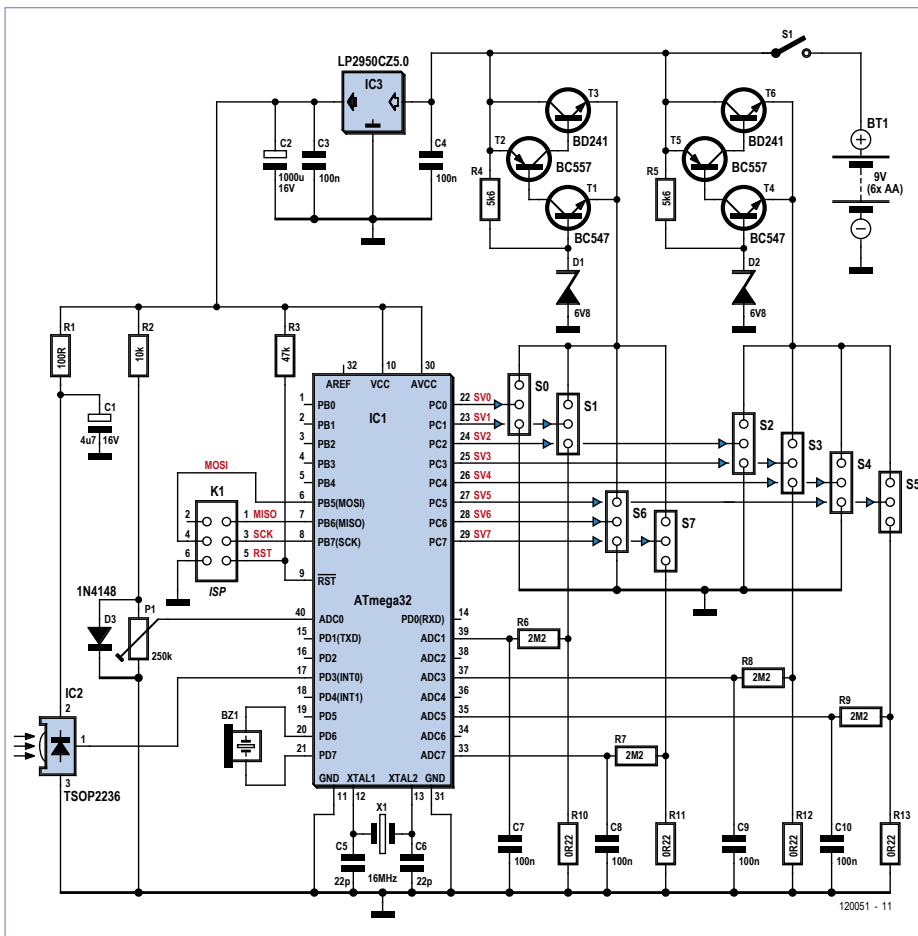


Bild 3. Der Quadrowalker wird von einem ATmega32 gesteuert.

Die Ströme der vier Servos, die für das horizontale Bewegen der Beine zuständig sind, werden über Widerstände in den Masseleitungen gemessen. Da die Servo-Motoren digital gesteuert werden, fallen an den Widerständen keine kontinuierlichen Spannungen ab. Die hinzugefügten Tiefpässe bilden Mittelwerte, die der Mikrocontroller über seine ADC-Eingänge misst. Der Zweck der Stromüberwachung ist das Erkennen von Hindernissen, auf die der Quadrowalker trifft. Das Umgehen der Hindernisse ist die einzige „Intelligenz“, die in diesem kleinen Roboter eingebaut ist. Andererseits genügt sie, damit sich der Quadrowalker „autonom“ nennen darf. Die Schwelle, ab der Hindernisse erkannt werden, lässt sich mit einem Potentiometer einstellen, das mit einem weiteren ADC-Eingang des Mikrocontrollers verbunden ist. Die dort anliegende Spannung nimmt der Mikrocontroller als Referenz.

Die Reibung der Roboterfüße am Boden hat auf die Hindernis-Erkennungsschwelle maßgeblichen Einfluss. Beim Musteraufbau steckten die Beine in harten Plastikugeln,

nicht zuletzt damit der Boden nicht verkratzt. Damit durchgeführte Tests verliefen sowohl auf Holz- und Kunststoffböden als auch auf Teppichen erfolgreich. Von Füßen aus Gummi wird abgeraten, da die Reibung so hoch ist, dass sich der Schwellenwert der Hinderniserkennung nur mühsam einstellen lässt.

Die Servos arbeiten an Spannungen im Bereich 4,8...6 V. Vier frische, in Reihe geschaltete Mignon-Batterien liefern die Spannung 6 V (NiMH-Akkus etwas weniger), so dass ein Spannungsregler nicht erforderlich ist. Leider sinkt die Spannung im Betrieb schnell auf nicht mehr ausreichende Werte. Deshalb gehören zur Stromversorgung sechs Akkus oder Batterien, ihre Spannung wird von Spannungsreglern auf 6 V herabgesetzt. Viele integrierte Regler haben relativ hohe Spannungsgefälle, so dass die Kapazitäten der Akkus oder Batterien nicht optimal genutzt werden können. Eine unkomplizierte Lösung dieses Problems sind zwei einfache, diskret aufgebaute Spannungsregler, das Spannungsgefälle liegt hier unter 1 V. Vorhanden sind zwei

Spannungsregler, einerseits weil damit das Spannungsgefälle am niedrigsten ist, andererseits weil ein Kühlkörper entfällt.

Software

Das Mikrocontroller-Programm für den Quadrowalker wurde in Assembler geschrieben, es setzt sich größtenteils aus Interrupt-Routinen zusammen. An die Servos müssen in Abständen von 20 ms Impulse gesendet werden, die 1...2 ms breit sind. Geschieht dies nicht, beginnen die Beine zu vibrieren. Der Infrarot-Empfänger der Fernsteuerung verlangt ein präzises Timing, auch dafür ist eine Interrupt-Routine nötig. Die Steuerung der acht Servos wurde im Programm über die 20 ms lange Periode verteilt, denn wenn die Servos gleichzeitig gesteuert würden, hätte dies starke impulsförmige Belastungen der Stromquelle zu Folge.

Die Werte der Servo-Positionen stehen in einer internen Sinustabelle, die 256 Werte im Bereich 0...100 enthält. Das entspricht einem Steuerbereich von 1 ms in Schritten von 10 μ s. Wenn die Servos des gleichen Beins über die Sinustabelle mit einer Phasendifferenz angesteuert werden, bewegen sich die Beinengelenke kreis- oder ellipsenförmig. Dadurch sind Schritte möglich, bei denen das Bein gleichzeitig angehoben und vorwärts bewegt oder abgesetzt und rückwärts bewegt werden kann. Die entstehende Laufbewegung kommt einer natürlichen Beinbewegung bereits recht nahe.

Die steuernden Signale der acht Servos müssen so gegeneinander phasenverschoben werden, dass der Quadrowalker vorwärts oder rückwärts läuft, oder sich dreht. Die Richtungsumkehr wird durch die Phasenumkehr der Signale erreicht, dies gilt auch für das Drehen nach rechts oder links. Die Phasenwerte dieser Grundbewegungen sind ebenfalls in einer internen Tabelle abgelegt. Diese Tabelle enthält für jede Bewegung acht Werte, jeder Wert stellt die Phasenverschiebung eines Servo-Signals dar.

Der Quadrowalker führt die Kommandos aus, die er über eine Infrarot-Fernbedienung erhält. Ein bestimmtes Kommando aktiviert die Hinderniserkennung, sie verhilft dem Roboter zu der Fähigkeit, Hindernissen auszuweichen. Der Mikrocontroller vergleicht den Motorstrom mit dem eingestellten Schwellenwert. Beim Überschreiten des

Schwellenwerts starten folgende Aktionen: Aus dem Lautsprecher tönt ein Signal, die aktuell ausgeführte Bewegung stoppt. Der Roboter bewegt sich ein paar Schritte rückwärts und dreht sich um einige Grad, wobei die Drehrichtung von dem Bein abhängt, das zum Zeitpunkt des Auftreffens auf das Hindernis in Bewegung war. Nach einem kurzen Stopp wird der Weg in die korrigierte Richtung fortgesetzt. Auch Wände und Mauern können den Quadrowalker nicht dazu bringen, dass er sich festfährt. Bei eingeschalteter Hinderniserkennung bleibt der Quadrowalker ohne Hilfe von außen endlos in Bewegung, jedenfalls solange seine Akkus oder Batterien durchhalten. Recht spannend dürfte die Frage sein, ob ein Roboter wie der Quadrowalker fähig ist, Fußball zu spielen. Um diese Frage beantworten zu können, wurde in die Software eine weitere Funktion implementiert. Der Roboter führt mit einem Vorderbein eine Kickbewegung aus, wobei das diagonal gegenüberliegende Hinterbein zum Halten der Balance ebenfalls angehoben wird. Die Praxis hat gezeigt, dass der Quadrowalker einen Tischtennisball etwa einen Meter weit schießen kann. Vorausgesetzt, dass das Spielfeld nicht zu groß ist, steht Fußballspielen unter Seinesgleichen nichts im Weg. Die Ausdauer des Quadrowalker mit einer Akkuladung oder einem Satz Batterien hängt stark von seinen Aktivitäten ab. Wenn die Spannung unter eine bestimmte Marke sinkt, wird die Brown-Out-Erkennung des Mikrocontrollers aktiv. Erkennbar ist dies an den auftretenden zuckenden Bewegungen. Bei völlig erschöpften Akkus oder Batterien streckt der Quadrowalker meistens alle vier Beine von sich.

Fernsteuern

Wie schon erwähnt, lässt sich der Quadrowalker mit einer zum RC5-Standard kompatiblen TV-Fernbedienung steuern. Das erste Byte des RC5-Codes wird nicht ausgewertet, so dass eventuell auch andere Fernbedienungs-Typen brauchbar sind. Wenn mehrere Quadrowalker voneinander unabhängig gesteuert werden sollen, kann die Identifizierung im Programm angepasst werden. Die vier Grundkommandos lauten Vorwärts, Rückwärts, Linksdrehen und Rechtsdrehen. Über die „Stumm“-Taste

(Mute) der Fernbedienung sind weitere Funktionen zugänglich, beispielsweise die Hinderniserkennung und das Ausbalancieren. Beim Ausbalancieren erhält das rechte Vorderbein über Servo 6 einen vertikalen Offset, so dass alle vier Beine fest auf dem Boden stehen. Infolge mechanischer Toleranzen kann es nämlich vorkommen, dass ein Bein zu hoch oder zu niedrig steht, was das Fortbewegen erschwert. Der Offset wird nach dem Einstellen im nicht flüchtigen EEPROM des Mikrocontrollers gespeichert, er wird bei jedem Folgestart berücksichtigt. Aus **Tabelle 1** geht hervor, dass die Zifferntasten 1 bis 9 den Grundfunktionen zugeordnet sind. Dem Drücken des

Funktionstasters (Stumm, Mute) muss eine Ziffer folgen. Funktion 1 schaltet die Hinderniserkennung ein, Funktion 2 löst eine Kippbewegung aus, nach der die Schritthöhe und Schrittweite aber auf Null stehen. Diese Funktion ist lediglich für Testzwecke vorgesehen. Mit Funktion 9 kann der Offset des Servo 6 eingestellt und gespeichert werden, was wie folgt geschehen muss: Nach Einschalten des Quadrowalker sind „Funktion“ und anschließend „9“ zu drücken. Während eine Hand unter den Roboter greift, wird Taster „Lautstärke +“ mehrfach gedrückt, so dass Servo 6 hochfährt. Der Roboter muss dann horizontal so ausgerichtet werden, dass die drei anderen Beine den Boden berühren. Dann wird „Lautstärke –“ so lange gedrückt, bis das vierte Bein den Boden berührt. Nach nochmaligem Drücken von „Funktion“ und „9“ werden die Einstellungen gespeichert. Normalerweise muss diese Prozedur nur ein einziges Mal vorgenommen werden.

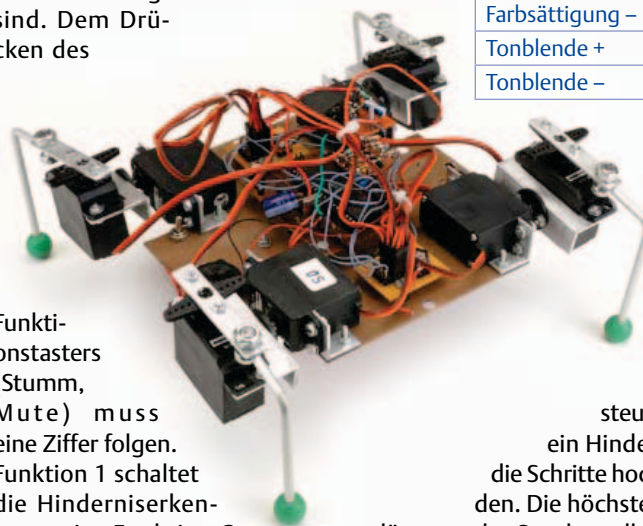


Tabelle 1. Fernbedienungskommandos	
1	Stoß mit Bein links vorn
2	Vorwärts
3	Stoß mit Bein rechts vorn
4	Linksdrehen
5	Stopp
6	Rechtsdrehen
7	Stoß mit Bein links hinten
8	Rückwärts
9	Stoß mit Bein rechts hinten
Stumm (Mute) X	Funktion X (X=0...9)
Lautstärke +	Tempo oder Servo 6 höher
Lautstärke –	Tempo oder Servo 6 niedriger
Helligkeit +	Schrittweite größer
Helligkeit –	Schrittweite kleiner
Farbsättigung +	Schritthöhe größer
Farbsättigung –	Schritthöhe kleiner
Tonblende +	Roboter höher
Tonblende –	Roboter niedriger

Wenn der Quadrowalker unterwegs ist, lässt sich sein Gang wie in der Tabelle angegeben mit den Tastern für Lautstärke, Helligkeit, Farbe und Tonblende steuern. Soll der Quadrowalker ein Hindernis übersteigen, müssen die Schritte hoch und weit eingestellt werden. Die höchste Geschwindigkeit erreicht der Quadrowalker, wenn das Tempo (Lautstärke) und die Schrittweite (Helligkeit) ihre maximalen Werte haben. Auf glatten Böden sind langsame, weite Schritte angebracht, während auf Teppichen auch schnelle und weite Schritte möglich sind. Im Testbetrieb wurde eine Höchstgeschwindigkeit von etwa 1 km/h gemessen. Der Assembler-Quelltext und der Hex-Code für den Mikrocontroller können von der Elektor-Projektseite [1] heruntergeladen werden. Ein Videoclip auf Youtube [2] zeigt den Quadrowalker beeindruckend in voller Aktion.

(120051)gd

Weblinks

- [1] www.elektor.de/120051
- [2] www.youtube.com/watch?v=8ToHa4hQi_0

2,4 GHz: Vielseitige Trägerfrequenz

Ein hochfrequentiertes Frequenzband!

Von Frank Bigrat (F)

Wi-Fi, Bluetooth, Mikrowellenherde... wir alle benutzen tagtäglich und ohne großes Nachdenken Geräte, die hauptsächlich im 2,4-GHz-Band arbeiten. Im vorliegenden Artikel wird ein kurzer Überblick über die grundlegenden 2,4-GHz-Technologien gegeben. Wir erklären dabei auch, was es mit der berühmten Frequenzspreizung auf sich hat, welche viele Vorteile bietet.

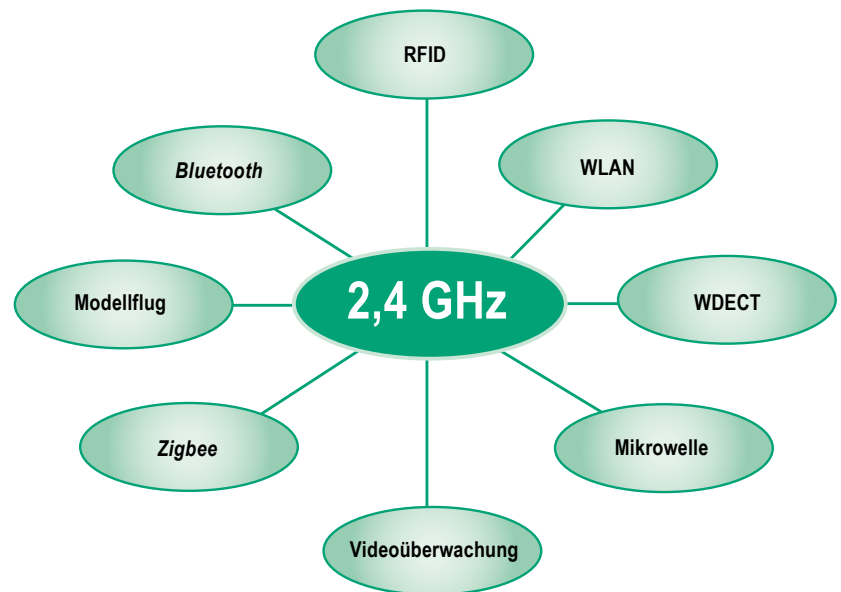


Bild 1. Verschiedene Anwendungen auf 2,4 GHz.

Das ISM-Band

Die 2,4-GHz-Frequenz gehört zu einem Frequenzband, das in Industrie, Wissenschaft und Medizin (**I**ndustrial, **S**cientific and **M**edical Band) genutzt wird und sich wie in **Tabelle 1** zusammengefasst aufteilt. Ein weithin bekanntes Beispiel für ein Gerät, welches im ISM-Band arbeitet, ist der Mikrowellenherd. Weniger bekannt sind medizinische Geräte für die Diathermie, welche menschliches Gewebe durch Emission elektromagnetischer Wellen erwärmen und so z.B. Muskelverspannungen lösen können.

Alleine das 2,4-GHz-Band kann ohne Einschränkungen auf der ganzen Welt eingesetzt werden. Dies erklärt die weite Verbreitung von Kommunikationsgeräten, die in diesem Frequenzband arbeiten. Es finden sich Anwendungen in den verschiedensten Bereichen, die hier kurz angerissen werden sollen (**Bild 1**):

- **Wi-Fi** (*Wireless Fidelity*): dies ist, neben *Bluetooth*, die bekannteste und am weitesten verbreitete 2,4-GHz-Technologie. Heute üblicherweise eingesetzt für drahtlosen Internetzugang tragbarer Rechner und Tablet-PCs erschien Wi-Fi Ende der 90er-Jahre auf Apple-Computern. Lokale Drahtlosnetze werden WLAN genannt, für *Wireless Local Area Network*.
- **RFID** (*Radio Frequency IDentification*): tagtäglich eingesetzt z.B. von Nutzern des ÖPNV, unter anderem als *passe Navigo* in der Region Paris. Diese Transponder können ohne direkten Kontakt

ausgelesen und beschrieben werden. Die notwendige Energie wird meist über die vom Lesegerät ausgesandten Radiowellen geliefert.

- **Bluetooth**: Der Mobiltelefonhersteller *Ericsson* hat diesen Standard drahtloser Kommunikation entwickelt, um Kabel zwischen Elektronikgeräten auf kurzer Distanz zu eliminieren, besonders in Haushaltsanwendungen. Die von den meisten Herstellern verwendeten Module haben eine Sendeleistung von 2,5 mW, was die Reichweite auf 20 m begrenzt. Dieser geringen Leistung der *Bluetooth*-Geräte steht eine Batteriestandzeit von mehreren Monaten gegenüber.
- **Zigbee**: Diese kleinen Module erlauben den Aufbau persönlicher Drahtlosnetze (*Wireless Personal Area Networks*, WPAN) bei geringen Kosten und geringstem Verbrauch, der eine Nutzungsdauer von mehreren Jahren zulässt. Auch die Domotik mit ihren vielen Sensoren und Fernbedienungen verwendet mehr und mehr dieses Protokoll. Man findet ZigBee auch in medizinischen Geräten, Rauch- bzw. Einbruchsmeldern oder bei *freebox*-Fernbedienungen der letzten Generation (eine französische Set-Top-Box).
- **WDECT** (*Wideband Digital Enhanced Cordless Telecommunication*): Technologie für Drahtlostelefone, Weiterentwicklung von DECT (*Digital Enhanced Cordless Telecommunication*), vorgesehen für Heim- oder Büroanwendungen. WDECT verwendet das FHSS-Verfahren (weiter unten erläutert).

Tabelle 1: Das ISM-Band ist reserviert für Anwendungen in Industrie, Wissenschaft und Medizin.

Frequenzband	Mittenfrequenz	Verfügbarkeit
6,765 – 6,795 MHz	6,780 MHz	nach geltenden lokalen Vorschriften
13,553 – 13,567 MHz	13,560 MHz	
26,957 – 27,283 MHz	27,120 MHz	keine besonderen Informationen veröffentlicht
40,66 – 40,70 MHz	40,68 MHz	
433,05 – 434,79 MHz	433,92 MHz	keine besonderen Informationen veröffentlicht
902 – 928 MHz	915 MHz	nur USA, Grönland und einige pazifische Inseln
2.400 – 2,500 GHz	2,450 GHz	
5,725 – 5,875 GHz	5,800 GHz	keine besonderen Informationen veröffentlicht
24 – 24,25 GHz	24,125 GHz	
61 – 61,5 GHz	61,25 GHz	nach geltenden lokalen Vorschriften
122 – 123 GHz	122,5 GHz	nach geltenden lokalen Vorschriften
244 – 246 GHz	245 GHz	nach geltenden lokalen Vorschriften

- **Mikrowellenherd:** Diese für viele unverzichtbar gewordene Einrichtung wurde in den 50er-Jahren erfunden. Die Möglichkeit, Lebensmittel durch Mikrowellen (1 GHz bis 300 GHz) zu erhitzen, wurde durch den Ingenieur Percy Spencer zufällig entdeckt, als er in der Nähe eines sendenden Radars arbeitete. Die Frequenz von 2,45 GHz regt die Wassermoleküle zum Schwingen an und erhitzt so die Speisen.
- **Flugmodellbau:** Auf dem Flugfeld angekommen prüft ein Modellpilot normalerweise zunächst, ob „seine“ Frequenz auch von anderen Piloten genutzt wird. Ein Absturz kann nur

Akkuwächter mit ZigBee“ aus dem Aprilheft 2012 von Elektor genutzt [1].

- **Videoüberwachung:** Elektronikhändler bieten verschiedene Modelle von Überwachungskameras im 2,4-GHz-Band an.

Wie man es erwarten könnte, handelt es sich bei *Wi-Fi*, *Bluetooth*, *Zigbee* und anderen um Handelsnamen oder Marken. Die Kommunikationsprotokolle werden durch Normen mit weniger einprägsamen Bezeichnungen spezifiziert: IEEE 802.11 für *Wi-Fi*, IEEE 802.15 für *Bluetooth* und *Zigbee*. Diese definieren das Datenübertragungspro-

Es ist es schwierig festzustellen, inwieweit die Telekommunikationsbranche in Studien zur Schädlichkeit elektromagnetischer Wellen involviert ist.

vermieden werden, wenn man sich mit mehreren Piloten die verfügbare Flugzeit teilt; dieses Problem kennt der Autor dieser Zeilen nur zu gut. In diesem Zusammenhang ist die 2,4-GHz-Technologie interessant, weil man sich dadurch der Frequenzprobleme entledigen kann. Die bekanntesten Hersteller (*Futaba*, *Graupner*, *Spektrum*, *Multiplex*...) haben die Vorteile digitaler Übertragung erkannt und bieten eine Vielzahl von Sendern und Empfängern für 2,4 GHz an. Auch Telemetrie-Funktionen sind möglich, für Flugparameter wie Höhe oder Batteriezustand. Diese Technik wird im Artikel „LiPo-

tokoll, die Übertragungsrate, die Reichweite in Innenräumen und im Freien, die Sendeleistung und vieles mehr. Für bessere Marktchancen bevorzugt man jedoch anschaulichere und attraktivere Bezeichnungen.

Die Ursprünge des 2,4-GHz-Bandes gehen übrigens auf das Jahr 1947 und die Erfindung des Mikrowellenherdes (s.o.) zurück: Damals entschied die Federal Communications Commission, das 2,4-GHz-Band drahtlosen Kommunikationsgeräten zuzuweisen. Das ist die US-amerikanische Behörde, die sämtliche Geräte zertifiziert und freigibt, die elektromagnetische Wellen aussenden.

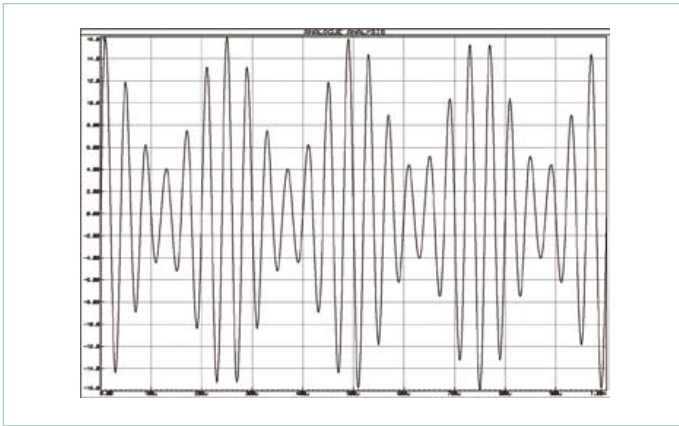


Bild 2. Ein amplitudenmoduliertes Signal (AM).

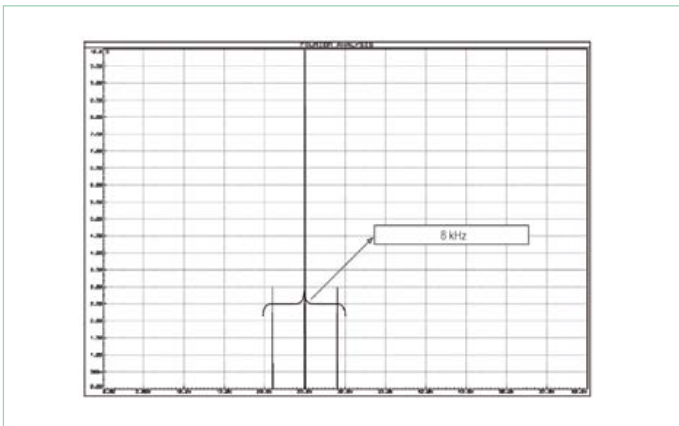


Bild 3. Spektrum des AM-Signals.

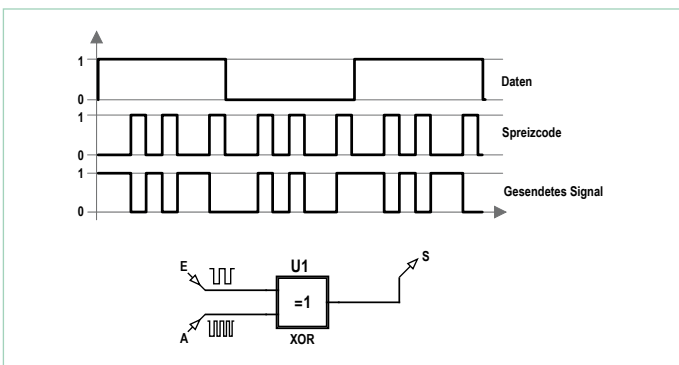


Bild 4. Das Prinzip von DSSS.

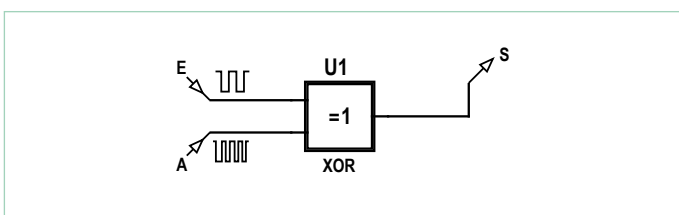


Bild 5. Ein ganz einfacher DSSS-Generator.

Übertragungstechniken: DSSS und FHSS

Betrachtet man die Vorteile der 2,4-GHz-Technologie, so kommt direkt die Frage auf, wie es gelingt, bei mehreren Sendern auf einer Frequenz gegenseitige Störungen auszuschließen.

Die Antwort ist (recht) einfach: man arbeitet mit Frequenzspreizung (*Spread Spectrum*), bei dem - wie der Name schon sagt - ein enger Frequenzbereich frei in einem größeren Frequenzbereich verteilt wird.

Man unterscheidet zwei Typen von Frequenzspreizung:

- *Direct Sequence Spread Spectrum* (DSSS) oder Frequenzspreizung mit direkter Folge
- *Frequency Hopping Spread Spectrum* (FHSS) oder Frequenzsprungverfahren.

Die Technik wurde anfänglich entwickelt, um ein feindliches Abhören von Nachrichten zu verhindern. Die zu übertragende Nutzinformation sollte wie Rauschen erscheinen.

Entwickelt zur Steuerung von Torpedos ist die Technik seit dem Zweiten Weltkrieg bekannt. Sie wurde aber in größerem Maße erst seit den 60er-Jahren eingesetzt, und zwar von den Amerikanern seit der Kuba-Krise im Oktober 1962.

Diese schwere internationale Krise, ausgelöst durch die Stationierung sowjetischer Nuklearraketen auf der Insel Kuba, also wenige Hundert Kilometer vor der Küste Floridas, gipfelte in der Blockade der Insel durch die Vereinigten Staaten und in Machtdemonstrationen, die nicht weit davon entfernt waren, den Kalten Krieg in einen „heißen Krieg“ zu verwandeln.

Obwohl es heute nur mehr anekdotisch ist, ist es doch interessant festzuhalten, dass FHSS von einer bemerkenswerten Person erdacht und patentiert wurde. Es war der Pianist und Komponist (und Autor, Kritiker, selbstproklamierter Endokrinologe...) George Antheil, in enger Zusammenarbeit mit seiner Weggenossin, der Hollywood-Schauspielerin Hedy Lamarr (und geschiedenen Ehefrau eines österreichischen Rüstungsindustriellen).

Bis heute wird diese Technik eingesetzt, um das Risiko von Interferenzen mit anderen Signalen zu vermindern.

Für eine bessere Vorstellung von DSSS und FHSS bietet sich ein kurzer Exkurs zur Modulation an. Um unter guten Bedingungen ein Sinussignal s der Frequenz f (in unserem Beispiel $f = 4$ kHz) zu übertragen, wird dieses in Amplitude oder Frequenz auf ein weiteres Sinussignal P der Frequenz F (hier $F = 25$ kHz), den sogenannten Träger, aufmoduliert.

Analysiert man nun das aus einer klassischen Amplitudenmodulation resultierende Spektrum (**Bild 2**) (welches ein Graph der Amplitude jeder Sinuskomponente des Signals ist), dann erscheint das Signal s verschoben und zentriert um den Träger P (**Bild 3**). Des Weiteren erkennt man unschwer, dass das übertragene Signal ein Frequenzband (genannt Kanal) belegt, und dass die Breite hier gleich 8 kHz ist.

Bei der Übertragung numerischer Signale wird zur Modulation PSK benutzt, welches an anderer Stelle in diesem Artikel besprochen wird. Beim DSSS-Verfahren zur Frequenzspreizung wird jedes Bit

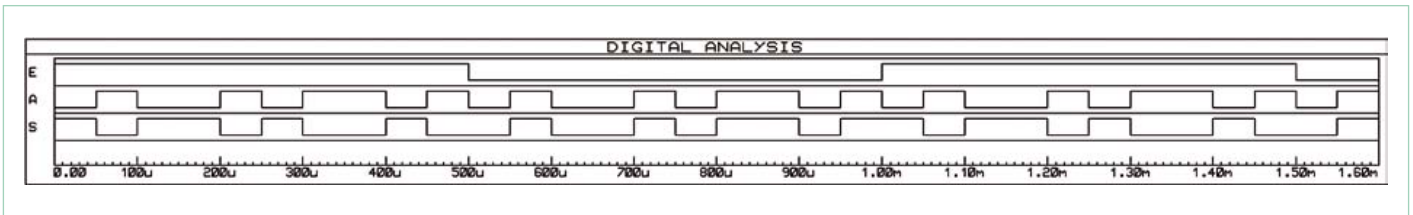


Bild 6. Das Ergebnis der Simulation.

des zu übertragenden numerischen Signals durch eine Folge (direct sequence) von n Bits ersetzt (**Bild 4**). Die Folge wird festgelegt durch Mischen des numerischen Nutzsignals mit einer binären Folge, dem *Spreizcode*. Diese Pseudozufallsfolge ist zwischen Sender und Empfänger verabredet. Bei dieser Technik ist die Bandbreite des übertragenden Signals n -mal so groß.

In unserem Beispiel hat der Spreizcode eine acht Mal höhere Datenrate als das Nutzsignal: das Spektrum wird also auf das Achtfache gespreizt. Das Sendesignal wird per Funk unter Verwendung des Modulationsverfahrens PSK (s.u.) ausgestrahlt.

Wir untersuchen die Auswirkungen von DSSS auf das Signalspektrum mit einem Simulator (z.B. Proteus von *Labcenter Electronics*). Das zu übertragende Signal ist ein Rechtecksignal der Frequenz 1 kHz und der Spreizcode ist die willkürlich gewählte Binärfolge 0100101101.

Bild 5 zeigt den einfachen Aufbau, mit welchem wir DSSS erzeugen und **Bild 6** zeigt das Ergebnis der Simulation.

Das Spektrum des Originalsignals wird in **Bild 7** wiedergegeben. Man erkennt unschwer, dass die Harmonischen, aus denen dieses Signal besteht, sich oberhalb 50 kHz stark abschwächen.

Bild 8 zeigt das Spektrum des Spreizcodes. Man sieht deutlich, dass sich dieses bis auf Frequenzen über 200 kHz erstreckt.

Das Spektrum des Sendesignals nach Spreizung erscheint in **Bild 9**: Auch dieses erstreckt sich bis über 200 kHz. Das Spektrum ist weit gespreizt und erinnert sehr an ein Signal mit Rauschen.

Beim Frequenzsprungverfahren, FHSS, wechselt die Trägerfrequenz periodisch nach einem pseudozufälligen Muster und belegt einen gegebenen Kanal nur zu gegebener Zeit (*dwell time*, 300 ms nach IEEE 802.11). Die Reihenfolge der Frequenzen ist dem Sender und Empfänger bekannt. Einen Überblick gibt **Bild 10**.

FHSS benutzt 79 Kanäle von 1 MHz Breite. Bei dieser Technik ist die Kollisionswahrscheinlichkeit, d.h. dass zwei Sender zu einem gegebenen Zeitpunkt auf demselben Kanal arbeiten, sehr gering.

Modulation des Signals

Bei der Frequenzspreizung wird üblicherweise ein unter Elektronikern wohl bekanntes Verfahren eingesetzt: die Phasenumtastung oder -modulation (PSK, *Phase Shift Keying*).

Bei dieser Art der Modulation ändert man die Phase am Punkt ϕ des Sinussignals (gegeben durch die Gleichung $U(t) = \hat{U} \cdot \sin(2 \cdot \pi \cdot f \cdot t + \phi)$), je nachdem, ob man eine 1 oder eine 0 übertragen will. Legt man willkürlich $\phi = 0$ fest, während eine 1 übertragen wird, und $\phi = \pi$, während eine 0 übertragen wird, dann ergibt sich der Graph in **Bild 11**.

Die MMICs

Doch wie überträgt man eine hohe Frequenz von 2,4 GHz, wenn die Bandbreite üblicher Bauteile etwa bei einigen Hundert MHz liegt? Diese Leistung ist erst seit dem Erscheinen von Hochfrequenz-ICs

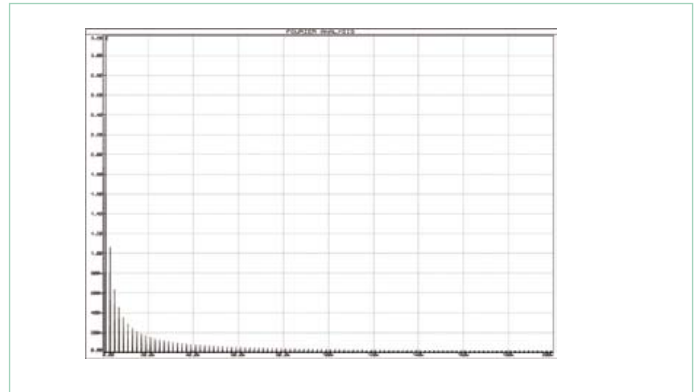


Bild 7. Das Spektrum des zu übertragenden Signals.

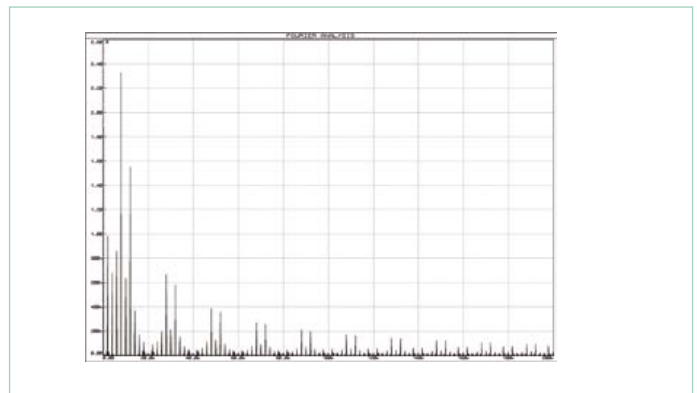


Bild 8. Das Spektrum des Spreizcodes.

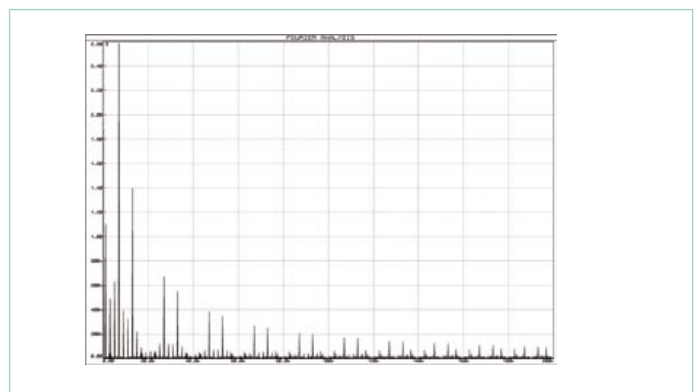


Bild 9. Das Spektrum des Sendesignals nach Spreizung.

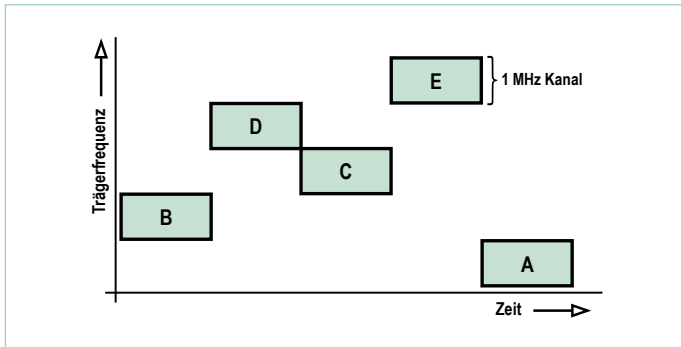


Bild 10. Das Prinzip von FHSS.

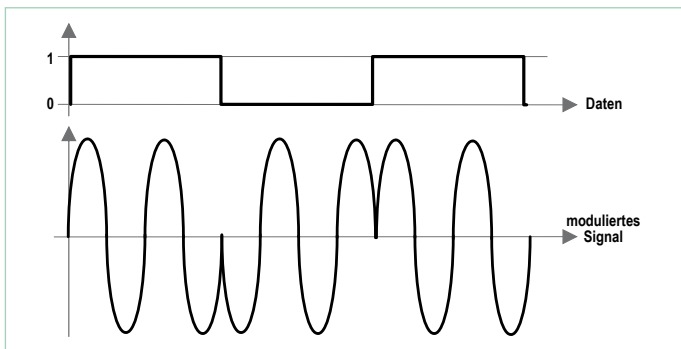


Bild 11. Phasenmodulation (PSK).

möglich, sog. MMIC (ausgesprochen: Mimic, für *Monolithic Micro-Wave Integrated Circuits*).

Diese Schaltkreise auf der Grundlage von Galliumarsenid (GaAs), einem Halbleitermaterial mit wesentlich besseren Eigenschaften als es das übliche Silizium hat, arbeiten bei Frequenzen zwischen 300 MHz und 300 GHz. Sie bieten alle Funktionen, die für eine Funkübertragungskette benötigt werden (Frequenzmischer, Leistungsverstärker, Low-Noise-Verstärker...).

MMICs werden auch auf Basis von Indiumphosphid (InP) hergestellt. Dieser Halbleiter hat noch bessere Eigenschaften als Galliumarsenid und erlaubt es, Komponenten mit noch größeren Bandbreiten- und höherem Preis - herzustellen.

Seit den 40er-Jahren befindet sich die Elektronikindustrie in einem Wettlauf um hohe Frequenzen. Zunächst wurden Röhren eingesetzt (Magnetrons, Wanderfeldröhren...), die Frequenzen bis rund 2 GHz erreichten. Diese Geräte werden auch heutzutage noch verwendet, insbesondere für Anwendungen hoher Leistung wie Mikrowellenherde, Radar oder Satellitensendeanlagen.

Mit dem Erscheinen von integrierten Schaltkreisen begann in den 70er-Jahren eine stürmische Entwicklung, die so unterschiedliche Bereiche betrifft, wie das Militär (Radarleitsysteme, Stör- und Abwehrmaßnahmen...), die Telekommunikation (Satellitenemp-

fang, schnurloses Telefonieren...), die Industrie (Materialforschung), das Gesundheitswesen (Behandlung von Tumoren...), die Weltraumforschung (Weltraumradar wie bei den SPOT-Satelliten...) oder das Automobil (Radar Anti-Kollision, Geschwindigkeitssensoren zur Auslösung von Airbags oder ABS...) und das Verkehrswesen (GPS, Verkehrsradar...).

Einflüsse von Radiowellen auf den Organismus

Trotz des massiven Einsatzes von Mikrowellen in der Telekommunikation sind die langfristigen Auswirkungen von elektromagnetischen Feldern hoher Frequenz auf den menschlichen Körper nicht hinreichend bekannt. Verschiedene jüngere Studien scheinen deren Schädlichkeit zu beweisen (welche sich äußert in Schlafstörungen, starker Müdigkeit, Kopfschmerzen, Schwindelanfällen, Übelkeit, Hörproblemen, Gedächtnisstörungen, Konzentrationsschwierigkeiten...). In einer vielbeachteten Pressemitteilung der Association Santé Environnement de France (www.asef-asso.fr) wird ein Zusammenhang zwischen den oben genannten Auswirkungen bei den Bewohnern mehrerer Wohnanlagen und der Anwesenheit einer Vielzahl von dort installierten Sendeantennen bestätigt. Andere Studien versichern das Gegenteil: das *Scientific Committee on Emerging and Newly Identified Health Risks* (http://ec.europa.eu/health/scientific_committees/emerging/index_en.htm) hat noch im März 2007 bestätigt: „Seit 2001 wurden ... eingehende Forschungen, epidemiologisch, in vivo und in vitro durchgeführt. Hier wurde kein gesundheitlicher Effekt förmlich nachgewiesen.“

Nationale Behörden wie die Agence Française de Sécurité Sanitaire de l'Environnement et du Travail (Afsset) geben Vorsichtsratschläge heraus und erstellen Listen möglicher gesundheitlicher Gefahren. Leider ist es schwierig festzustellen, inwieweit die Telekommunikationsbranche in diese Studien involviert ist, insbesondere mit einer direkten und indirekten Finanzierung, und wie weit ihr Einfluss reicht.

Das einzige Prinzip, welches hier greift, ist das Prinzip der Vorsicht. Doch genügt es, das Telefon, den Rechner, die Headsets und den WLAN-Router auszuschalten, wenn sie nicht benötigt werden?

Die Debatte ist noch lange nicht beendet.

(110331)

[1] www.elektor.de/110759/

Der Autor

Der Autor - geboren 1969 – interessiert sich seit seiner Jugend für Elektronik und Informatik. Nach dem französischen Fachhochschulabschluss lehrt er dieses Fach seit 16 Jahren Schülern der Abschlussklassen an verschiedenen Schulen im Raum Paris.

 Entwicklung industrietauglicher Software und Hardware sowie Elektronik 03303/212166 oder www.jasys.de	Gravuren / Schilder / Frontblenden www.webgrav.de	LCDs und mehr www.lcd-store.de www.LC-Design.de www.crystallfontz.de	Leiterplatten für Pragmatiker!  www.pcb-joker.com
Alles Spule! Wir liefern und fertigen: Drähte, HF-Litzen, Ferrit- und Eisenpulverkerne, Spulenkörper, Isoliermaterial, Klebebänder, Tränklacke, Übertrager, RFID-Spulen, Sensor- und Aktorspulen, Prototypen, Kleinserien, Serien, Ersatzteile und vieles mehr. MM Menting Mikroelektrik www.spulen.com	W2micro www.w2micro.de • diverse SMD-Adapter • RS232-TTL-Pegelwandler • Sonderelektronik (z.B. Audioverteiler) • Leiterplattenbestückung und Kabelkonfektionierung ab 1 Stück	LOETRONIC Embedded MP3 Module www.loetronic.com	www.anttronic.de ab 1 Stck. ANT TRONIC Leiterplatten zu TOP-Preisen!!
HEXWAX LTD www.hexwax.com Treiberunabhängige USB-ICs von einem der Weltmarktführer • USB-UART/SPI/I2C-Konverter • TEAleaf-USB Authentifizierungs-Dongles • expandIO-USB I/O-USB-Expander • USB-FileSys Flash-Drive mit SPI-Interface • USB-DAQ Flash-basierter Datenlogger	Bausätze zu ELEKTOR 1986 bis heute! Teilesätze, Platinen, programmierte Controller sowie Cds zu fast allen Elektor-Projekten vom Spezialist. Alle Elektor-Artikel zum Verlagspreis. Ihr zuverlässiger Partner für aktive und passive elektronische Bauteile und Komponenten: Geist Electronic-Versand GmbH Tel.: 07720/36673 Fax: 07720/36905 Mail: info@geist-electronic.de Shop: www.geist-electronic.de	SCOPES und mehr HAMEG Instruments A Rohde & Schwarz Company MESSTECHNIK zum fairen Preis	Kleinrechner mit FPGA www.bomerenzprojekt.de
Günstige Platinen clever bestellen!  www.pcb-joker.com	NienTech SCNITTSTELLENWANDLER von WLAN LAN USB nach RS485 RS422 RS232 TTY über virtuellen COM-Port ansprechbar www.NienTech.de	USB-ISOLATOR  Galvanische Trennung Störfreie Messungen Sicherheit und Schutz CESYS www.cesys.com Elektronik · Messtechnik · FPGA	Ausgabe: November 2012 Anzeigenschluss: 18. September 2012 Erscheinungstermin: 17. Oktober 2012

www.elektor.de

 PCBs Muuuuch Cheaper... No-frills policy	17.22 EURO* 5 pcbs, 100 mm x 100 mm *per piece, incl. VAT (23%) + shipping costs e. g. Germany 10.89 EURO Jackaltac COM www.jackaltac.com
--	---

Der „Pansanitor“ (1928)

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



Reizstromgeräte versprechen Linderung bei Schmerzen, Muskelverspannungen, Neuralgien und vielen anderen Beschwerden. Die dazu verwendeten nieder- oder hochfrequenten Wechselströme kann man heute mit modernen Halbleitern problemlos erzeugen, doch vor 80 Jahren war der Transistor noch nicht erfunden, und ICs oder gar Mikrocontroller waren schlechterdings unvorstellbar! Dieser Artikel der Retronik-Reihe bietet daher die seltene Gelegenheit, fast vergessenem, historischem Know-how am Beispiel eines originalen „Pansanitor“-Reizstromgerätes aus dem Jahr 1928 nachzuspüren.

Reizstromgeräte: Anwendung und Wirkung

„Den Ausstrahlungen von hochgespannten und hochfrequenten Strömen rühmt man, ähnlich wie den von Influenzmaschinen gelieferten hochgespannten Strömen, eine beruhigende Wirkung auf das Nervensystem im allgemeinen und auf schmerzhaftes Nerven im besonderen nach.“ Mit diesen Worten aus dem „Handbuch für Frauenheilkunde“ [1] wird Hofrat Prof. Dr. Opitz im Buch „Hochfrequenz für Kranke und Gesunde – Ein ärztlicher Ratgeber“ [2] zitiert. Dies war aber nicht der einzige Anwendungsfall für Elektrotherapie. Ob Stoffwechsel, Atmungsorgane oder Haut und Haar betroffen waren: Dank Reizstromgeräten wie dem Pansanitor war man schon so gut wie gesund – so zumindest die Meinung der Autoren.

Auch heute noch werden Reizstromgeräte in der Medizin eingesetzt und damit z.B. Muskelschwächen, Schmerzsyndrome und Durchblutungsstörungen behandelt. Dabei setzt man auch auf deren Wärmewirkung, die tiefer dringt als etwa Infrarotbestrahlung.

Das historische Umfeld

Das vorliegende Gerät stammt aus dem Jahre 1928. Das war das Jahr, in dem auch Ernst Udet mit einem 20-PS-Flugzeug auf der Zugspitze landete, die erste Funksprechverbindung von Deutschland in die USA hergestellt wurde und Alexander Fleming das Penicillin entdeckte [3]. Erst drei Jahre zuvor hatte Julius Edgar Lilienfeld in Kanada das erste Patent zum Prinzip des Transistors angemeldet [4]. Die Grundlagen für die Erfolgsgeschichte der Reizstromgeräte schuf Nicola Tesla bereits in den letzten Jahren des 19. Jahrhunderts. Diesbezügliche Forschungsergebnisse veröffentlichte er 1898 in der Zeitschrift „The Electrical Engineer“ unter dem Titel „High Frequency Oscillators for Electro-Therapeutic and other Purposes“ [5]. Ausgehend von einem normalen Funkeninduktor beschreibt er - mehr aus der Sicht des Elektroingenieurs als aus Sicht eines Arztes - den Aufbau der in seinen Experimenten benutzten Geräte und deren stufenweise Verfeinerung.

Ein Funkeninduktor besteht im Prinzip aus einem Transformator, einem Kondensator, einem Wagner'schen Hammer und einer Funkenstrecke. Aus wirtschaftlichen Gründen schaltete Tesla zwei Kondensatoren parallel, um die Potentialdifferenz an den Anschlüssen der Sekundärseite zu verringern und so Materialeinsparungen bei diesen realisieren zu können. Um den dadurch entstehenden Nachteil der Verringerung der Frequenz der Ströme wieder auszugleichen, fügte er auf der Sekundärseite eine nach ihm benannte Tesla-Spule hinzu [5]. Als Elektroden zur Übertragung der hochfrequenten Ströme benutzte Tesla luftleer gepumpte Glaskolben. Später wurden diese mit Gasen gefüllt, wodurch sie im Betrieb violett leuchteten. Dies trug den Geräten im englischen Sprachraum die Bezeich-

Medizinische Reizstromerzeugung vor 80 Jahren

nung „Violet Wand“ ein. Die Form der Glaskolben wurde stark den zu behandelnden Beschwerden und Körperteilen angepasst – eine Auswahl zeigt **Bild 1**.

Beschreibung des Pansanitors

Das Pansanitor-Gerät und alle Zubehörteile wie zahlreiche Glas-Elektroden, Kabel und der Elektrodenhalter sind in einem handlichen, mit violetterm Plüsch ausgeschlagenen Koffer untergebracht (siehe Bild 1). Die Kontrolleinheit zur Erzeugung der Reizströme befindet sich in einem schwarzen Holzgehäuse, dessen obere Deckplatte mit einer elfenbeinfarbenen Kunststoffplatte versehen ist, auf der die Beschriftungen für die zwei Drehknöpfe und die Anschlussstecker aufgedruckt sind (**Bild 2**). Auch Gerätenamen und Modellbezeichnung sind kaum zu übersehen. Merkwürdigerweise findet sich aber kein Hinweis zum Hersteller. Meine Recherchen im Internet blieben bislang erfolglos.

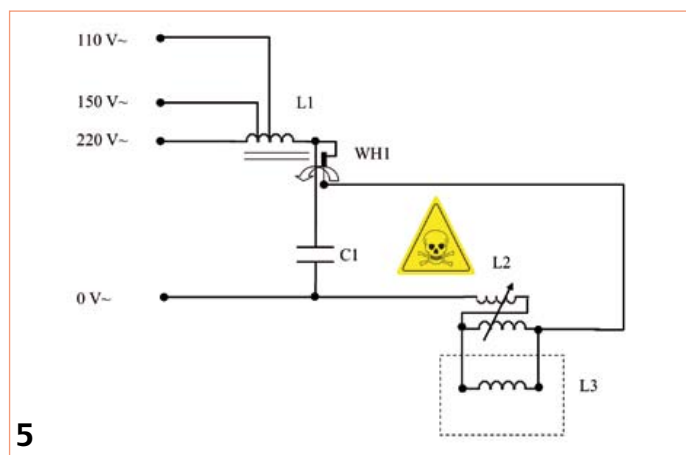
Auf der linken Seite befinden sich für den Anschluss an das Stromnetz vier Steckstifte in liegender T-Anordnung. Dabei ist der mittlere Stift ganz links der gemeinsame Pol für den Betrieb an Wechselstrom mit 110, 150 oder 220 V. Der Hersteller war also international tätig und die Geräte arbeiteten an unterschiedlichen Netzspannungen. Die richtige Auswahl musste allerdings der Kunde treffen. Einen Schutz vor zu großer Spannung gab es nicht, und nicht einmal eine Schmelzsicherung war vorhanden. Unbenutzte Stifte wurden mit schwarzen Kunststoffkappen abgedeckt.

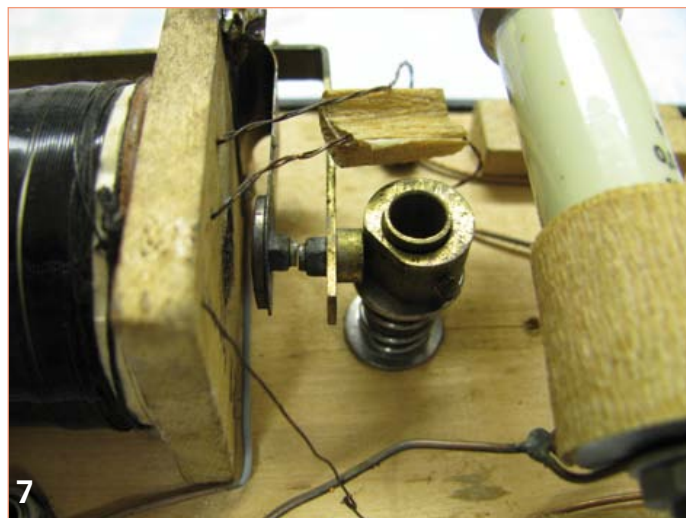
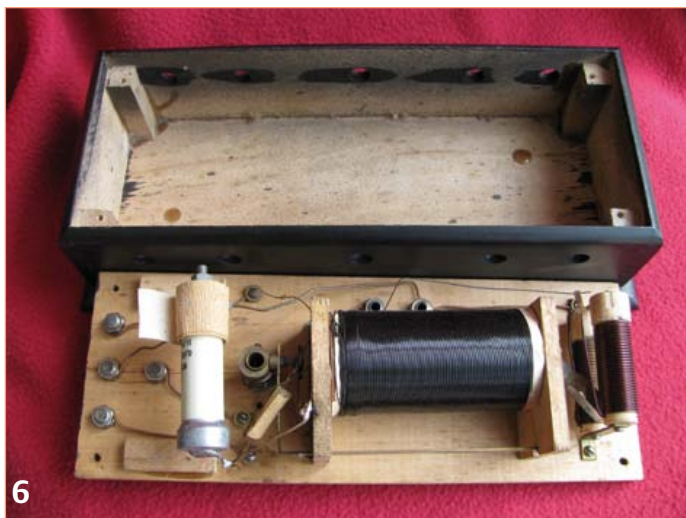
Das Netzkabel wurde von einem Vorbesitzer ausgewechselt, doch die originalen Stecker aus Porzellan (**Bild 3**) sind noch erhalten und in gutem Zustand. Der linke Drehknopf dient als Netzschalter, mit dem rechten Knopf kann man die Stärke des Stroms einstellen.

Die Glaselektroden steckten im als Griff dienenden Elektrodenhalter und waren an die mittleren beiden Buchsen angeschlossen (**Bild 4**). Ich traute mich allerdings nicht, das Gerät einzuschalten und auszuprobieren.

Schaltung des Pansanitors

Bild 5 zeigt den Schaltplan und **Bild 6** das geöffnete Gerät. Die Spule L1 besitzt zwei Anzapfungen für die unterschiedlichen Netzspannungen. Das ganze Gerät steht folglich unter Netzspannung! L1 bildet zusammen mit dem mechanischen Kontakt WH1 einen Wagner'schen Hammer, der gleichzeitig mit Hilfe der Exzenterwelle des Ein-/Ausschalters als Hauptschalter fungiert (**Bild 7**). Wird der Pansanitor eingeschaltet, schließt WH1 und in L1 baut sich ein Magnetfeld auf, wodurch WH1 öffnet. Damit wird der Stromfluss unterbrochen und die entstehende Induktionsenergie in C1 gespeichert. C1 hat eine Kapazität von 0,025 μF und ist bis 2 kV $\sim$ spannungsfest – zumindest war er das einmal. Bei diesem Methusalem von Kondensator sollte man es darauf nicht unbedingt ankommen lassen. Nach Zusammenbruch des Magnetfelds in L1 schließt WH1 wieder, worauf wieder Strom fließt und das Spiel von vorne beginnt. Die schon recht hohe Spannung wird durch den Tesla-Transformator aus der Primärspule L2 und der Sekundärspule L3 weiter hoch transformiert. L2 besteht aus zwei hintereinander geschalteten,





aber übereinander verbauten Spulen (**Bild 8**). Mit einer Spreizklammer zwischen den beiden Teilspulen, die mit dem Intensitätsregler gedreht wird, kann die Hochspannung in Grenzen eingestellt werden. Dazu greift die Spreizklammer einfach Windungen ab, ganz ähnlich wie bei einem Drahtpotentiometer oder einem Regeltrafo. Die Sekundärspule L3 befindet sich im Handstück (Bild 4) und wird über Steckkontakte an das Hauptgerät angeschlossen. L3 besteht aus nur zehn Windungen, die jedoch mit doppeltem Draht ausgeführt sind (**Bild 9**).

Die ganze Schaltung ist handverdrahtet auf der Rückseite der Deckplatte des Gerätes montiert und an Einfachheit und Raffinertheit kaum zu überbieten.

Tesla-Trafo eingebaut!

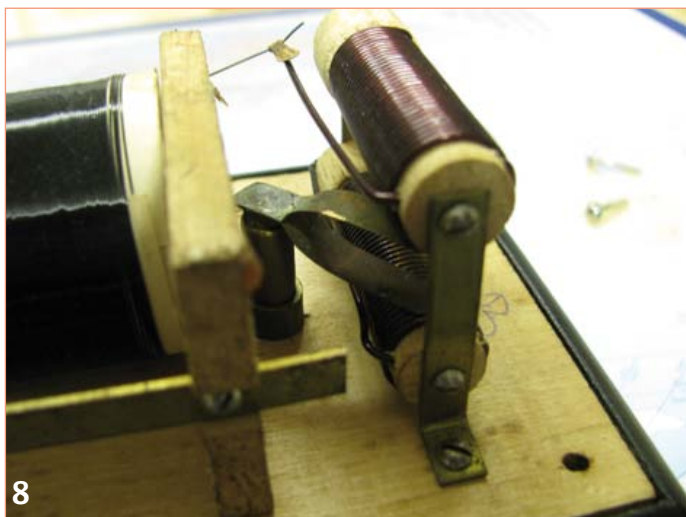
Während der Mikrocontroller in aktuellen Nachfolgern des Pansanitors wohl das wichtigste Bauteil ist, hat diese Rolle beim Pansanitor des Jahres 1928 sicherlich die Tesla-Spule. Mikrocontroller gehören heute zum elektronischen Alltag und viele Elektor-Leser kennen einige davon in- und auswendig. Doch von Tesla-Spulen geht gerade heute etwas Geheimnisvolles aus, dem sich kaum ein Elektronik-Fan entziehen kann. Das liegt sicherlich nicht nur an der charismatischen Person ihres Erfinders Nikola Tesla, sondern auch an ihrer bekanntesten Inkarnation, dem leider 1917 abgerissenen Wardenclyffe Tower [6]. Bilder von gewaltigen Blitzentladungen tun ein übriges [7].

Das eigentlich Faszinierende eines Tesla-Trafos ist aber die Tatsache, dass bei diesem nicht das Windungszahlenverhältnis zwischen Primär- und Sekundär-Spule allein für die Erhöhung der Eingangsspannung verantwortlich ist, sondern Resonanzen der beiden Spulen mitspielen [8].

(120177)

Weblinks & Literatur

- [1] Bergmann: München, 1922; Seite 354
- [2] Thüringer Verlagsanstalt und Druckerei GmbH: Jena, 1930; www.electrotherapymuseum.com/2005/HF/index.htm
- [3] http://de.wikipedia.org/wiki/Alexander_Fleming
- [4] <http://brevets-patents.ic.gc.ca/opic-cipo/cpd/eng/patent/272437/summary.html#Details>
- [5] The Electrical Engineer, Vol. XXVI, No. 550, 1898; www.tfcbooks.com/tesla/1898-11-17.htm
- [6] http://de.wikipedia.org/wiki/Wardenclyffe_Tower
- [7] http://de.wikipedia.org/wiki/File:Lightning_simulator_questacon02.jpg
- [8] <http://de.wikipedia.org/wiki/Tesla-Transformator>



elektor

Elektor Print

Gewohnter Lesespaß auf Papier



Elektor Digital

Neuer Lesespaß auf PC, Notebook
oder Tablet



Elektor PLUS

Ultimativer Lesespaß zu Hause oder unterwegs

**Lesen Sie Elektor im vorteilhaften
PLUS-Abonnement!**

Jetzt abonnieren oder upgraden: www.elektor.de/abo

Hexadoku

Sudoku für Elektroniker

Wenn Sie nach dem Studium der interessanten Projekte in diesem Heft noch etwas Zeit übrig haben: Hier kommt unser berühmtes Hexadoku, das Sie schon ein Weilchen beschäftigen dürfte. Doch das könnte sich durchaus lohnen: Wer uns die richtigen Zahlen in den grauen Kästchen zusendet, kann einen von vier Gutscheinen gewinnen!

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

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

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

Mitmachen und gewinnen!

Unter allen internationalen Einsendern mit der richtigen Lösung verlosen wir

einen **ELEKTOR-Gutschein** im Wert von 100 € und

drei **ELEKTOR-Gutscheine** im Wert von je 50 €.

Einsenden

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

Elektor – Redaktion Süsterfeldstr. 25 52072 Aachen

Fax: 0241 / 88 909-77 E-Mail: hexadoku@elektor.de

Als Betreff bitte nur die Ziffern der Lösung angeben!

Einsendeschluss ist der 30. September 2012!

Die Gewinner des Hexadokus aus dem Juniheft stehen fest!

Die richtige Lösung ist: 7924A.

Der Elektor-Gutschein über 100 € geht an: Thomas Raith aus Ditzingen.

Einen Elektor-Gutschein über je 50 € haben gewonnen: Reino Anttila, Michael Evans und Nuno Tavares.

Herzlichen Glückwunsch!

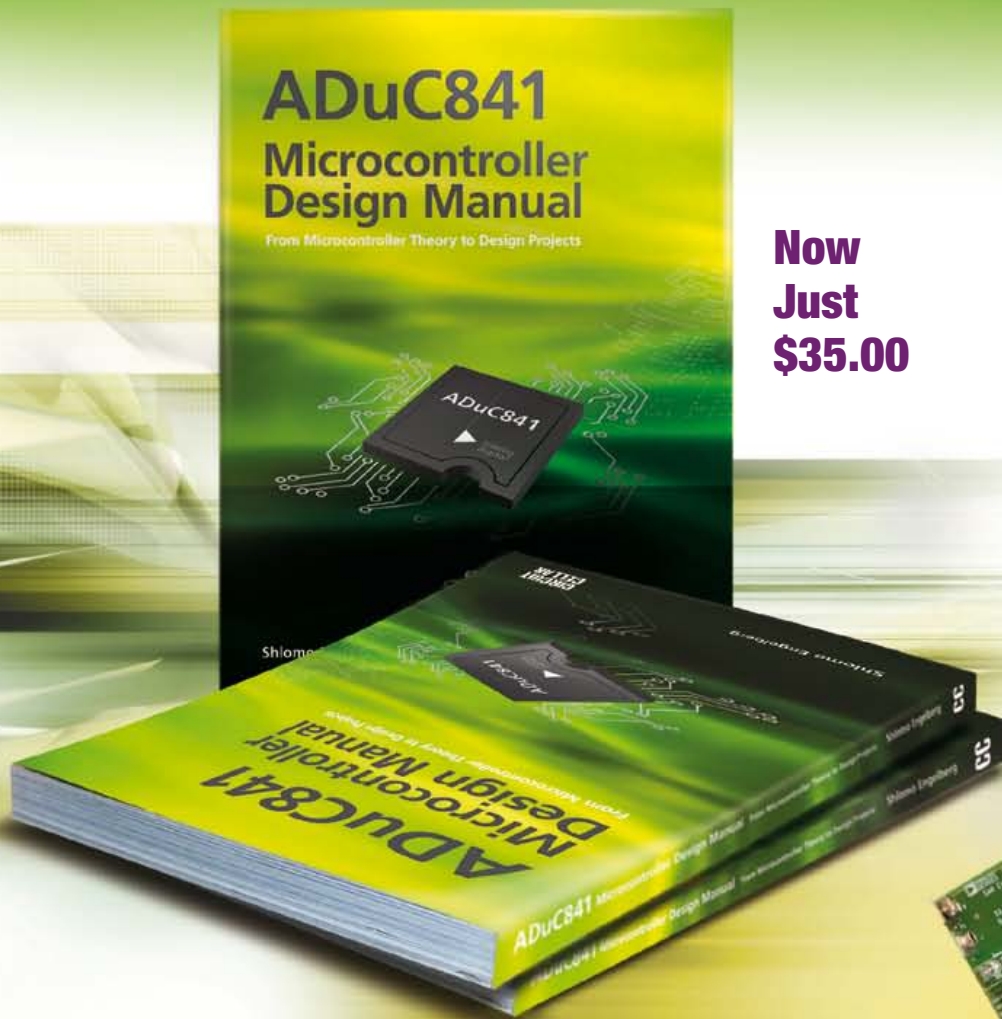
		C		9						E		6			
7			B	C	0				A	F	3				5
A	6				4			0						7	C
E	F			2	8			3	D					A	B
		3		B	2		F	4		E	1		A		
			F	3	4	0			6	B	A	E			
2	5		D		7	A			9	0		8		B	4
	A														9
	8														0
F	2		9		A	C			B	8		6		E	D
			E	5	8	2			A	4	0	F			
		5		F	1		6	9		D	C		2		
5	C			8		6			F	4				2	E
B	0	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

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

CIRCUIT CELLAR

ADuC841 Microcontroller Design Manual: From Microcontroller Theory to Design Projects

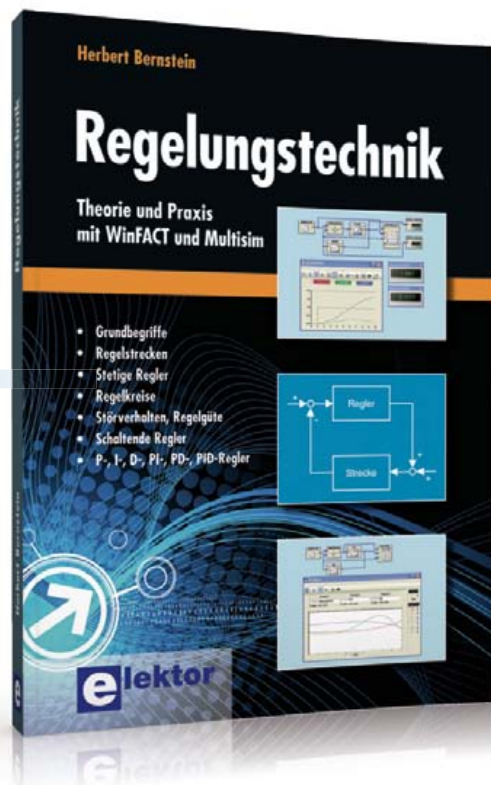
If you've ever wanted to design and program with the ADuC841 microcontroller, or other microcontrollers in the 8051 family, this is the book for you. With introductory and advanced labs, you'll soon master the many ways to use a microcontroller. Perfect for academics!



**Now
Just
\$35.00**

Buy it today!

www.cc-webshop.com



Theorie und Praxis
mit WinFACT und Multisim

1 Regelungstechnik

Die heutige Regelungstechnik hat Verknüpfungspunkte mit fast jedem technischen Gebiet. Ihre Anwendungen reichen von der Elektrotechnik über die Antriebstechnik und den Maschinenbau bis hin zur Verfahrenstechnik. Will man nun die Regelungstechnik anhand der fachlichen Regeln dieser einzelnen Gebiete erklären, so müsste man von einem Regelungstechniker verlangen, jedes Fachgebiet, in dem er Regelungen vornehmen will, fundiert zu beherrschen. Dies ist aber bei dem heutigen Stand der Technik nicht möglich. Bei der Regelung einer Antriebsaufgabe, einer Druck- oder einer Temperaturregelung tauchen Gemeinsamkeiten auf, die man mit einer einheitlichen Vorgehensweise beschreiben kann. Die Grundgesetze der Regelungstechnik gelten in gleicher Weise für alle Regelkreise, ganz unabhängig davon, wie verschieden sie im Einzelnen auch apparativ aufgebaut sein mögen. Dieses Buch richtet sich an den Praktiker, der gründlicher in die Regelungstechnik eindringen möchte, auf ausschweifende theoretische Exkursionen in die Mathematik aber gerne verzichten kann.

365 Seiten (kart.) • ISBN 978-3-89576-240-6

€ 49,00 • CHF 60,80



Kompletter Elektor-Jahrgang 2011
auf DVD

2 Elektor-DVD 2011

Die neue Elektor-Jahrgangs-DVD enthält alle Artikel des Jahrgangs 2011. Sie verfügt über eine sehr übersichtlich gestaltete Benutzeroberfläche. Mit der Elektor-DVD 2011 können Sie Platinenlayouts in perfekter Qualität drucken; diese Layouts mit einem Zeichenprogramm verändern; die Schnellsuchfunktion benutzen, mit der Sie in den einzelnen Artikeln oder im ganzen Jahrgang nach Wörtern, Bauteilen oder Titeln suchen können; Schaltbilder, Platinenlayouts, Illustrationen, Fotos und Texte exportieren.

ISBN 978-90-5381-276-1 • € 27,50 • CHF 34,10

Band 1: Grundlagen

3 Stromversorgung ohne Stress

Die Funktion aller elektronischen Schaltungen und Geräte steht und fällt mit der Stromversorgung. Schon deshalb muss man dieser Baugruppe besondere Aufmerksamkeit widmen. Dieses Buch beinhaltet Grundlagen und Schaltungen der Stromversorgungstechnik für elektronische Geräte aus der Praxis. Dem aktuellen Trend folgend hat der Au-



tor der mobilen Stromversorgungstechnik und der Schaltenteiltechnik besondere Aufmerksamkeit gewidmet.

294 Seiten (kart.) • ISBN 978-3-89576-248-2

€ 38,00 • CHF 47,20

4 Elektor-Linux-Board (Elektor Mai 2012)

Linux läuft heutzutage auf den unterschiedlichsten Geräten – sogar in Kaffeemaschinen. Es gibt daher viele Elektroniker, die an Linux als Basis für eigene Controller-Projekte interessiert sind. Eine Hürde ist jedoch die scheinbar hohe Komplexität, außerdem sind Entwicklungsboards oft recht teuer. Mit diesem kompakten Modul, das bereits für modernste Embedded-Projekte fertig bestückt ausgestattet ist, gelingt der Linux-Einstieg ideal und preiswert zugleich.

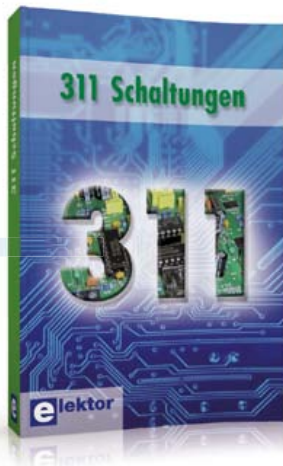
Bestücke und getestete Platine

Art.-Nr. 120026-91 • € 64,95 • CHF 80,60

Kreative Lösungen
aus allen Bereichen der Elektronik

5 311 Schaltungen

Das mittlerweile zwölfte Buch aus Elektors erfolg-



reicher „Dreihunderter“-Schaltungsreihe bietet ein weiteres Mal neue Konzepte sowie einen (fast) unerschöpflichen Fundus zu allen Bereichen der Elektronik: Audio & Video, Spiel & Hobby, Haus & Hof, Prozessor & Controller, Messen & Testen, PC & Peripherie, Stromversorgung & Ladetechnik sowie zu Themen, die sich nicht katalogisieren lassen.

544 Seiten (kart.) • ISBN 978-3-89576-255-0
€ 36,80 • CHF 45,70

Von 0 und 1 zum FPGA

6 Digitale Logik selbst entwickeln

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

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

Bausatz mit allen Bauteilen inklusive Platine(n)

7 TAPIR – E-Smog-Detektor

Der TAPIR (Totally Archaic but Practical Interceptor of Radiation) spürt „strahlendes Missverhalten“ elektronischer Geräte in Ihrer Umgebung auf. Trotz der einfachen Schaltung handelt es sich hierbei um einen ultrasensitiven E-Smog-Detektor, der jede Quelle eines elektrischen oder – mit einer entsprechenden Antenne versehen – magnetischen Feldes aufspürt und dies akustisch signalisiert.

Art.-Nr. 120354-71
€ 14,95 • CHF 18,60

„Die Spannung steigt!“

8 Stromversorgungen in der Praxis

Die Elektronik bestimmt unser tägliches Leben mehr denn je – Tendenz steigend. Dabei benötigen alle elektronischen Geräte und Systeme eine gut funktionierende Stromversorgung mit spezieller Anpassung an die Betriebsbedingungen. Dieses neue Buch beschreibt die entsprechenden Möglichkeiten vom

Transformator bis zum passenden Kühlkörper. Behandelt werden die wichtigsten Merkmale, Einsatzmöglichkeiten und das Betriebsverhalten von vielen unterschiedlichen Stromversorgungsgeräten.

366 Seiten (kart.) • ISBN 978-3-89576-239-0
€ 46,00 • CHF 57,10

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

www.elektor.de

Elektor-Verlag GmbH

Süsterfeldstr. 25

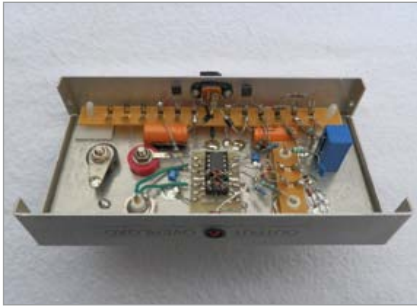
52072 Aachen

Tel. +49 (0)241 88 909-0

Fax +49 (0)241 88 909-77

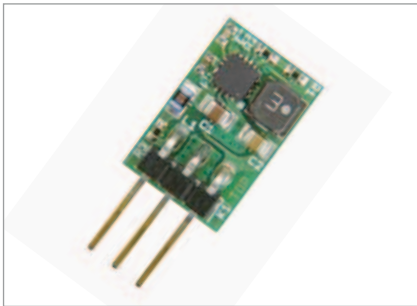
E-Mail: bestellung@elektor.de

elektor



Transkonduktanz-Verstärker

Der OpAmp AD844 wurde von Analog Devices in erster Linie für Anwendungen mit Stromrückkopplung entwickelt. Wegen seines unkonventionellen Konzepts ist er aber auch für den Einsatz als gesteuerte Stromquelle geeignet. Mit ihm lässt sich beispielsweise eine spannungsgesteuerte Stromquelle für messtechnische Aufgaben realisieren. In der nächsten Ausgabe beschreiben wir die Arbeitsweise des AD844 sowie einen mit diesem OpAmp aufgebauten Messvorverstärker.



Der 7805 - neu erfunden!

Zweifellos ist der 7805 der meistverwendete Spannungsregler in der Elektronik. Leider arbeitet er nicht besonders ökonomisch, denn die gesamte überschüssige Energie geht als Verlustwärme verloren. Ein Schaltregler, der den 7805 ersetzt, wäre nicht nur bezogen auf die Energieeffizienz überlegen, auch andere Eigenschaften könnten optimiert werden. Das Elektor-Labor hat einen kompakten Ersatz mit einem Step-Down-Konverter TPS62150 entworfen, der Eingangsspannungen 5,5 V...17 V in stabilisierte 5 V bei maximal 1 A umsetzt. Wenn man einige Widerstandswerte ändert, sind auch andere Ausgangsspannungen machbar.



Erweiterungen für den Strahlungsmesser

Der Strahlungsmesser vom November 2011 fand bei unseren Lesern reges Interesse. Das Gerät zeigt seine Stärken insbesondere beim Messen schwach radioaktiver Stoffe sowie bei Langzeituntersuchungen. Im Mikrocontroller des Systems ist ein Bootloader integriert, so dass die Firmware ohne viel Aufwand modifiziert werden kann. In der nächsten Elektor-Ausgabe beschreiben wir mehrere Firmware-Erweiterungen. Ferner stellen wir eine Hardware-Ergänzung vor, die den Messbereich vergrößert.

Elektor Oktober 2012 erscheint am 19. September 2012.

Elektor gibt es im Buchhandelsbuchhandel, Elektronik-Fachhandel, an ausgewählten Kiosken und garantiert beim Presse-Fachhändler. Sie können Elektor auch direkt bei www.elektor.de bestellen.

Änderungen vorbehalten!

www.elektor.de www.elektor.de www.elektor.de www.elektor.de www.elektor.de www.elektor.de

Elektor im Netz

Alle Magazin-Artikel ab 1996 können einzeln im PDF-Format heruntergeladen werden (gegen Elektor-Credits). Zu jedem Artikel findet man eine kurze Zusammenfassung, technische Daten und eine Stückliste (sofern vorhanden), was bei der Projektauswahl hilft. Dazu kommen eventuelle Updates/Berichtigungen und die Downloads zum Artikel (z.B. Software und Zusatzinfos). Auch die zum Projekt gehörenden Produkte wie Platinen, programmierte Controller und mehr lassen sich hier bestellen.

Im Elektor-Shop findet man alle anderen Angebote von Elektor, wie CD-ROMs, DVDs, Kits, Module, Software und Bücher. Mit der Suchfunktion können Sie die ganze Site nach Stichwörtern durchforsten.

Ebenfalls unter www.elektor.de:

- News aus der Elektronikwelt
- Leserforum
- Download des E-Magazins
- Spezialangebote (zeitlich begrenzt)
- FAQs, Inhaltsverzeichnisse und Kontaktformular



SPECIAL: SAVE 50% SPECIAL: SAVE 50% SPECIAL

Celebrate Circuit Cellar's **25th Anniversary**



\$25 Print or Digital :: **\$50** Combo

Celebrate *Circuit Cellar's* 25th year of bringing readers insightful analysis of embedded electronics technology.

Visit www.circuitcellar.com/el912 to take advantage of these great deals.

BONUS OFFER! BONUS OFFER! BONUS OFFER! BONUS OFF

Sign up today and you'll also receive the **Special 25th Anniversary Edition** with your subscription!





- ✓ über 40 Jahre Erfahrung
- ✓ mehr als 35.000 Produkte auf Lager
- ✓ kein Mindermengenaufschlag
- ✓ schneller 24-Std.-Versand

**Professionelle Qualität
zu Discount-Preisen!**

Weller®

Weller Lötkolben-Set inkl. Sicherheitsablage

- für WX-Versorgungseinheiten
- Präzise Temperaturregelung
- LED-Statusanzeige

WELLER WXP120SET

209,95



120
Watt

Weller WX 2 Lötstation

- Desktop Controller, 240 W
- 2 Kanäle für Lötwerkzeuge
- Innovatives Bedienkonzept über 6 Sensortasten
- digitale Temperaturregelung
- USB-Port (für Speicherstick)
- Temperaturbereich: 50-550 °C

Weller WX 2

449,95



Weller WD 1000 Lötstation

- Versorgungseinheit, 95 W
- Temperaturbereich: 50-450 °C
- Automatische Werkzeuergenerkennung
- ESD-safe, Potentialausgleich
- inkl. WP 80 Kolben und WDH 10 Komfortablage
- Robustes Design für harten Dauereinsatz

WELLER WD 1000

379,00



95
Watt

Entlötpinzette für WD1

- inkl. Ablage
- speziell für SMD-Bauteile

WELLER WMRT

339,95



Kleinschmelzbad für WD1

- Vorbereitung von Komponenten
- verzinnen von Drahtenden und Bauteilen
- Lötbadgröße: Innen Ø 20 mm, Tief 25 mm

WELLER WSB 80

339,95



Weller Zero-Smog Lötdampfabsaugsystem

- sorgt für reine Atemluft
- HEPA E12 = 99,5% Luftreinigung
- für Löt-, Kleb-, Schweiß- und Laseranwendungen
- inkl.: 1x Absaugarm 3 Meter Absaugschlauch Ø 50 mm 1 Tischklemmset mit Absperrventil



WELLER WFE2ESKIT

968,00

Weller®

Bleifreier Lötendraht (250 g)

- mit Silberanteil
- Schmelzpunkt: 221 °C
- Sn96 Ag4 mit F-SW 32

LÖTZINN F2-5250 **32,75** 0,5 mm
 LÖTZINN F2-8250 **30,50** 0,8 mm
 LÖTZINN F2-1250 **29,50** 1,0 mm

- mit Kupferzusatz
- Schmelzpunkt: 230 °C
- Sn99 Cu1 mit F-SW 32

LÖTZINN F3-5250 **15,10** 0,5 mm
 LÖTZINN F3-8250 **12,75** 0,8 mm



FELDER
Löttechnik

Katalog kostenlos!
 Tagesaktuelle Preise:
www.reichelt.de

