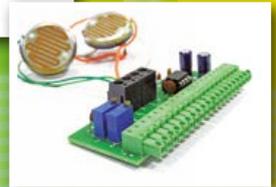


e lektor

Sommer-Doppelausgabe

132 Seiten – mit extra vielen Projekten



Gitarren-Tuner

Erweiterbares Arduino-Shield

IO-Board für das Messen, Steuern und Regeln

Chipkarten über USB auslesen • Knobelspiel mit ATtiny

Strahlungsmesser • 10 Rohde & Schwarz Oszilloskope zu gewinnen

Videogame-Emulator • Serieller Blinker/Dimmer • RC Speed Control

LED-Taschenlampe • Vegane Mausefalle • Ultraschall-Entfernungsmesser

Akku-Regenerierer • Sicherungswächter • Master-Slave-Steckdosenleiste

780X-Ersatz • Punktdisplay-Treiber • Maulwurfscheuche

Signal für Wasserwaage • Steckdosentester • IR-Tester

Schnittstellen-Isolator • Mehr Strom für Lithium-Lader

Kapazitive Erkennung von Flüssigkeiten

Retronik: Computerskop

Cube Clock



Elektor 3D-Drucker

Drucken und Modellieren in der dritten Dimension

Elektor bietet in Kooperation mit dem 3D-Drucksystem-Spezialisten German RepRap diesen 3D-Drucker (mit blauer Speziallackierung) in der Sonderausführung mit exklusivem Lieferumfang an.

Der neue Elektor 3D-Drucker ermöglicht bessere Druckergebnisse und einen wesentlich schnelleren Aufbau als gängige Drucker-Bausätze. Für die im Vergleich wesentlich kürzere Aufbauzeit sorgen Steckanschlüsse bei den Schrittmotoren, bestückte Platinen und ein fester, stabiler Edelstahlrahmen.

Technische Daten:

- Abmessungen (B x H x T): 500 x 460 x 460 mm
- Druckvolumen (X x Y x Z): 230 x 230 x 125 mm
- Geschwindigkeit: 3 mm Material bis 100 mm/Sekunde, 1,75 mm Material bis ca. 180 mm/Sekunde, Leerlauf bis 350 mm/Sekunde
- Betriebsspannung: 230/115 V Wechselspannung
- Material: ABS / PLA / PS / PVA / Laywood (Holzdraht) / Laybrick



Lieferumfang:

- Kunststoffteile (Polyamid-Kunststoff, besonders schlagfest und belastbar)
- Edelstahlrahmen (in blau) mit Nivelierfüßen, Schrauben, Gewindestangen, Wellen, Linearkugelbuchsen, Kugel- und Gleitlager
- T 2.5 Antriebsriemen und gefräste Riemenscheiben
- 5x NEMA17 Schrittmotor mit 0,52 Nm Drehmoment
- Ramps v1.4 Elektronik inkl. Kabel und Netzteil
- Modulares Hot-End 3 mm mit 0,5 mm Düse (optional auch mit 0,4 / 0,3 mm Düse)
- Beheiztes Druckbett 12 V
- 1x PLA Plastik 750 g (rot 3 mm)
- 1x PLA Plastik 750 g (gelb-grün 3 mm)
- 1x PLA Plastik 750 g (blau 3 mm)
- Software-CD inkl. Druckcode für ein Druckobjekt

Preis: 1.199,00 Euro
(inkl. MwSt.)

Die Lieferung innerhalb Deutschlands erfolgt frei Haus.

Weitere Infos & Bestellung unter www.elektor.de/3d-drucker

Lost Something? Never Again!

**Keep track of your most valued people
and possessions anywhere, anytime.**

Fear of losing your car? Are you worried about your children's safety on the streets? Perhaps your forgetful grandparents wander off and are unable to find their way back home. We want you to know that things like this no longer have to be of your concern.

Inside The Technology

One of the key features of the KCS TraceME modules is tracking and tracing extremely small to enormous things. From bird tracking to fleets, the TraceME will not let you down.

Equipped with a state-of-the-art GPS receiver, the TraceME modules provide most reliable and accurate navigational data, enabling you to remotely track & trace anyone or anything you value.

Remotely Controlling Operations

With the many onboard sensors, the TraceME modules allow you to be in total control of whatever operation you may

be carrying out, wherever you are. Ranging from altitude and G-Force sensors to thermometers and RF tags, only to name a few, the TraceME really is suitable for nearly any job.

The Possibilities?

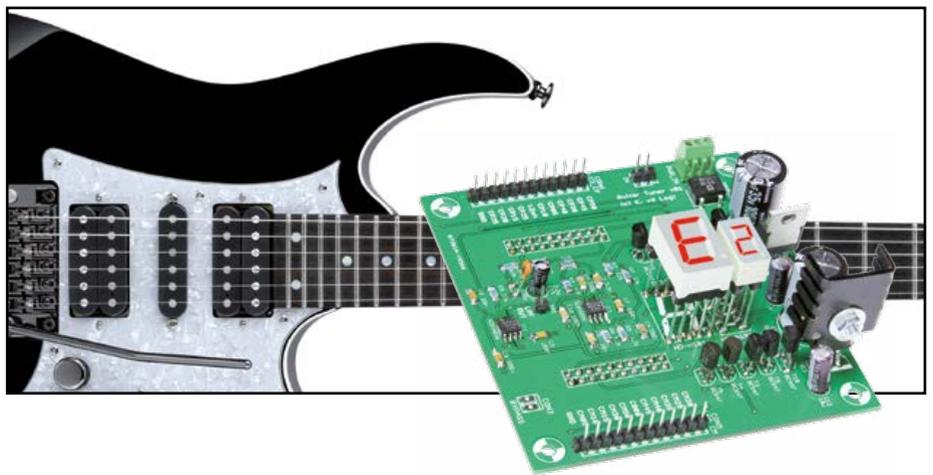
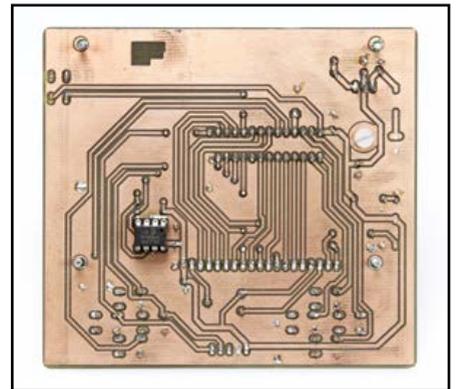
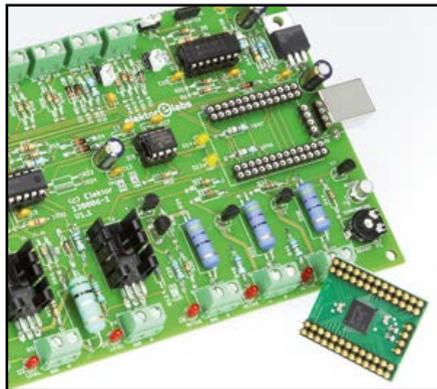
In short, the possibilities of TraceME are limited to your very own imagination!



We have a worldwide distribution and support network

www.trace.me

Email: trade@kcs-trade.com | Fax: +31-(0)20-5248130
Kuipershaven 22, 3311 AL Dordrecht, The Netherlands



● Industry

- 9 **Gewinnspiel**
Neues Digitaloszilloskop von Rohde & Schwarz



● Labs

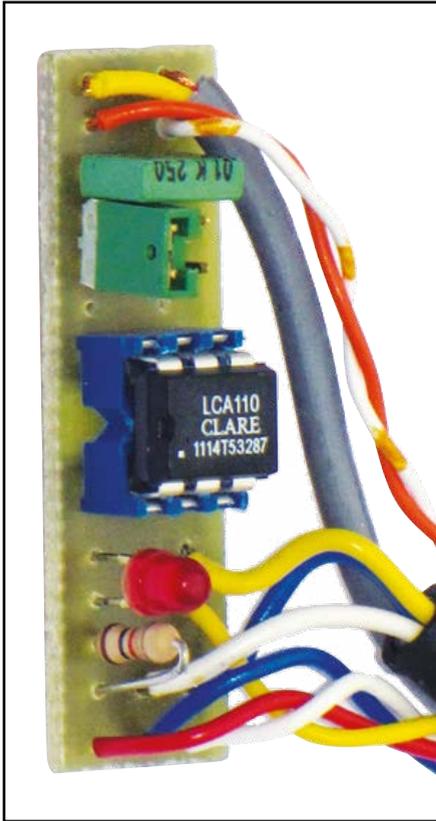
- 67 **Seltene Bauteile**
Tunnel-Dioden
- 68 **DesignSpark Tipps & Tricks**
Das Layout von Bussen
- 70 **It's Tag Time!**
Auf elektor-labs.com nach Uhr-Projekten suchen – und mehr!
- 71 **Arduino, il salvatore**
Arduino als Entwicklungs-Tool

● Projects

- 10 **IO-Warrior Erweiterungs-Board**
Alte PCs oder Notebooks müssen nicht im Keller verstauben! Die Ressourcen kann man sinnvoll nutzen: Mit einer universellen Interfacekarte, die mit unterschiedlichen Ein- und Ausgängen ausgestattet ist, wird der Computer zur Mess-Steuer-Regel-Station. Ein IO-Warrior-Modul auf dem Board übernimmt die USB-Kommunikation. Dazu gibt's Quellcode für das kostenlose Visual Studio Express.
- 22 **Gitarren-Tuner**
Gitarristen stimmen ihre Instrumente nach Gehör, oder sie nutzen elektronische Hilfen. Die Grundfrequenz eines Gitarrensignals lässt sich mit der FFT bestimmen, doch dem sind in der Praxis Grenzen gesetzt. Hier wird eine andere Methode der Frequenzanalyse angewandt, die von einer Autokor-

relationsfunktion Gebrauch macht. Kern der Hardware ist ein schneller Mikrocontroller ARM Cortex M4.

- 32 **Mein erstes Shield :-)**
Auch in der Welt der Mikrocontroller ist „Learning by Doing“ ein guter Weg. Der besonders günstige und einsteigerfreundliche Arduino Uno bringt leider so gut wie keine Peripherie mit. Wir haben daher ein kompaktes Shield entwickelt, das Anfängern mit einem Text-Display, LEDs und Tastern eine gute Basis für erste Schritte bietet. Und wer sich schon etwas auskennt, freut sich über zwei Erweiterungssteckverbinder. Hier lassen sich Relais-, Funk- und viele weitere Module anschließen.
- 38 **Mikrocontroller für Einsteiger (4)**
Ernsthafte Mikrocontroller-Anwendungen besitzen meist geeignete Schnittstellen zwischen Mensch und Maschine. Ein kleines Display, ein paar Taster, ein Einstell-Poti, ein



paar LEDs - schon kann man richtig schöne Geräte entwickeln.

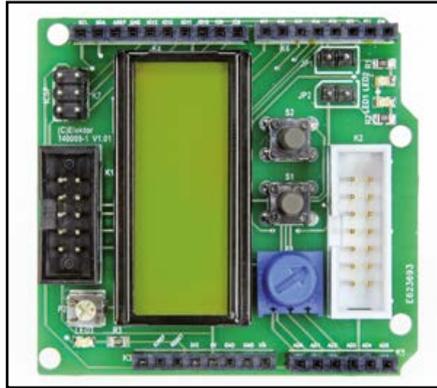
46 PIC-Strahlungsmesser

Der „Verbesserte Strahlungsmesser“ vom November 2011 ist ein praktikables und gleichzeitig kostengünstiges Gerät zum Messen radioaktiver Strahlung. Dort wird das mit einer Fotodiode aufgefangene Signal von einem ATmega88 verarbeitet. Diese Aufgabe kann aber auch von einem anderen Mikrocontroller übernommen werden - bei erweitertem Funktionsumfang.

52 SAME:

Chip-8 Videogame-Emulator

Die virtuelle Plattform Chip-8 existiert seit mehr als vier Jahrzehnten. Der Clou ist, dass kleine Spiele unabhängig von der Hardware programmiert werden können. Eine Virtuelle Maschine emuliert die Funktionen eines Betriebssystems sowie einen Prozessor, der real nicht existiert. In diesem Beitrag



wird das Konzept mit Hilfe eines PSoCs und eines Grafikdisplays umgesetzt.

62 RC Speed Control

DIY bedeutet, eine PWM-basierte Drehzahlsteuerung für RC-Gleichstrommotoren mit nichts als einem PIC, einem integrierten MOS-FET-Treiber und ein paar diskreten Bauteilen aufzubauen.

72 Chipkarten über USB auslesen

In Elektor gab es schon häufig Projekte, in denen ein PC einen I2C-Baustein ansteuerte. Meist wurde dazu die Intelligenz eines Mikrocontrollers als Vermittler zwischen PC- und I2C-Schnittstelle genutzt. Doch es geht auch einfacher, wie an diesem Lesegerät für Chipkarten gezeigt wird. Wir benötigen nur ein paar Bauteile und ein wenig „dirty engineering“!

75 DIY-LED-Taschenlampe

78 Rettung für die Lötstation

79 Vegane Mausefalle

84 Ultraschall-Entfernungsmesser

88 Signal für Wasserwaage

89 Steckdosentester mit Duo-LEDs

90 Ziege, Wolf und Kohlkopf

94 IR-Tester mit Solarzelle

95 Ersatz für 780X

96 Punktdisplay-Treiber

98 Maulwurfscheuche



100 NiCd-Akku-Regenerierer

101 Sicherungswächter

102 Mehr Strom für USB-Lithium-Lader

103 Master-Slave-Steckdosenleiste

104 Schnittstellen-Isolator

107 Kapazitive Erkennung von Flüssigkeiten

110 Optisches Theremin mit myDAQ und LabVIEW

114 Cube Clock

118 Review: Digilent „Analog Discovery“

● Magazine

6 Impressum

124 Retronik

Elektor Computerskop (1986)

127 Hexadoku

Sudoku für Elektroniker

130 Vorschau

Nächsten Monat in Elektor

Impressum

45. Jahrgang, Nr. 523/524 Juli/August 2014
Erscheinungsweise: 10 x jährlich
(inkl. Doppelhefte Januar/Februar und Juli/August)

Verlag

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

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

Hauptsitz des Verlags

Elektor International Media
Allee 1, NL-6141 AV Limbricht

Anzeigen:

Margriet Debeij (verantwortlich)
Tel. 02 41/88 909-13 / Fax 02 41/88 909-77
Mobil: +31 6 510 530 39
E-Mail: m.debeij@elektor.de

Julia Grotenrath

Tel. 02 41/88 909-16 / Fax 02 41/88 909-77
E-Mail: j.grotenrath@elektor.de

Es gilt die Anzeigenpreisliste Nr. 44 ab 01.01.2014

Distribution:

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

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

© 2014 elektor international media b.v.
Druck: Senefelder Misset, Doetinchem (NL)
ISSN 0932-5468

Ihr Projekt in Elektor

In den letzten Wochen haben wir Redakteure nicht nur dieses Doppelheft auf die Beine gestellt, sondern auch einen neuen „Workflow“ für Projekt-Vorschläge. Mit dem webbasierten System wollen wir vor allem die Zeit verkürzen, in der Sie Bescheid bekommen, ob wir Ihr Projekt in unserer Zeitschrift veröffentlichen können. Auch alle neuen Vorschläge auf unserer Projekt-Site www.elektor-labs.com bekommen wir nun in einer konferenztauglichen Übersicht angezeigt. Wenn wir



dort eine interessante Schaltung sehen, die noch etwas Reife-Zeit benötigt, setzen wir den Link auf „Wiedervorlage“. Ein solches Projekt wollen wir dann auch kommentieren, stimulieren und – wenn es unsere Zeit erlaubt – mit eigenen Entwicklungsideen unterstützen.

Jeder Projektvorschlag wird stets von einem der vier Länder-Redakteure begleitet. Mein erfahrenster Kollege Harry Baggen ist erste Wahl bei allen Messtechnik- und Audio-Projekten. *Editor EN* Jan Buiting ist prädestiniert für alles, was mit Stromversorgung, Motoren und HF zu tun hat. Denis Meyer ist unser erster Mann für Musik- und Medizinelektronik und alle sehr großen Projekte (mit und ohne Mechanik). Und ich werde mich auf eingesandte Mikrocontroller-Boards, Haustechnik-Projekte und Software-Artikel stürzen. Dabei werden wir uns natürlich auch Rat bei den Kollegen aus dem Labor holen, bevor wir die Vorschläge gemeinsam diskutieren.

Falls Sie also ein Projekt oder eine Schaltung aus den genannten Bereichen entwickelt haben (oder etwas ganz anderes!), dann zögern Sie nicht, und lassen Sie uns das wissen – via Elektor-Labs.com oder per E-Mail. Auch alle Entwickler, die kein Abo haben, möchten wir ausdrücklich zum Mitmachen auffordern (einen Gast-Zugang zur Labs-Website richten wir auf Anforderung gerne ein). Besonders willkommen sind vor allem kleine Schaltungen – zum Beispiel für unser nächstes Doppelheft.

Bleiben Sie uns treu!

Jens Nickel

Chefredakteur Elektor

Unser Team

Chefredakteur:

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

Ständige Mitarbeiter:

Dr. Thomas Scherer, Rolf Gerstendorf

Korrekturen:

Malte Fischer

Internationale Redaktion:

Harry Baggen, Jan Buiting, Denis Meyer

Elektor-Labor:

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

Grafik & Layout:

Giel Dols



Germany

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



United Kingdom

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



Netherlands

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



France

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



USA

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



Spain

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



Italy

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



Sweden

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



Brazil

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



Portugal

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



India

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



Russia

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



Turkey

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



South Africa

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



China

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

Unser Netzwerk



VOICE COIL



audio X PRESS



Die Elektor-Community



Unsere Partner und Sponsoren



Batronix

www.batronix.com/go/35 77



Beta Layout

www.pcb-pool.com/brd-to-3D 31



M. Braun Elektronik

www.mbraun-elektronik.de 87



Code Mercenaries

www.codemercs.de 87



KCS

www.trace.me 3



LeitOn

www.leton.de 51



LPKF

www.lpkf.de/prototyping 35



PEAK-System

www.peak-system.com 51



Pico Technology

www.picotech.com/PS254 113



Reichelt

www.reichelt.de 132



Schaeffer AG

www.schaeffer-ag.de 109

Sie möchten Partner werden?

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

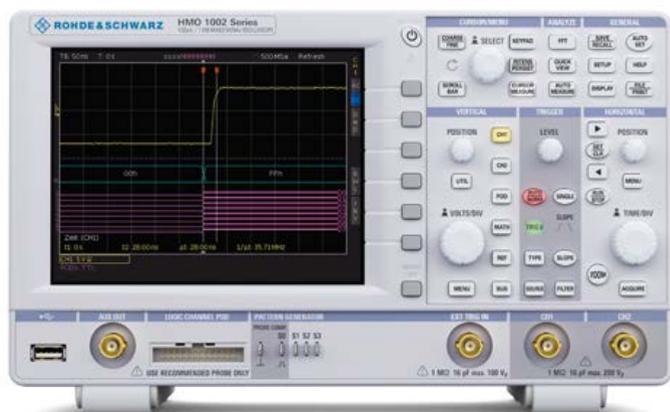
Von den Machern von Elektor!



Oder frei Haus unter www.elektor.de/roehren-10 bestellen!

Neues Digitaloszilloskop von Rohde & Schwarz

Hohe Qualität - niedriger Preis!



Das digitale Oszilloskop R&S®HMO1002 von **Rohde & Schwarz** ist mit Bandbreiten von 50 MHz, 70 MHz und 100 MHz erhältlich und zeichnet sich durch eine hohe Waveform-Update-Rate (die Geschwindigkeit, mit der das Oszilloskop Wellenformen auf dem erfassen kann) und vertikale Empfindlichkeit aus. Das lüfterlose Gerät überzeugt dabei mit einer Sampling-Rate von 1 GSamples/s und einer Speichertiefe von 1 MSample. Wie alle Oszilloskope der R&S®HMO-Serie verfügt das R&S®HMO1002 serienmäßig über eine Mixed-Signal-Funktionalität.

Hervorzuheben ist auch die Vielfalt von Zusatzfunktionen, die das Oszilloskop für ein breites Spektrum von Anwendern interessant macht: vom Schüler über den Service-Techniker bis zum Embedded-Entwickler.

Ein eingebautes, dreistelliges *Digitalvoltmeter* erleichtert vor allem Servicetechnikern die Arbeit. So lassen sich mit dem R&S®HMO1002 Spannungsmessungen auf beiden Analogkanälen mit jeweils zwei Messwerten gleichzeitig durchführen.

Für den edukativen Bereich ist der *Funktionsgenerator* sehr interessant, der alle grundlegenden Signalarten mit Frequenzen bis 50 kHz erzeugt. Schüler, Auszubildende oder Studenten erlernen dabei, die unterschiedlichsten Messaufgaben zu lösen. Im „Education Mode“ lassen sich sogar Komfortfunktionen abschalten.

Für Embedded-Anwender ist der integrierte *Pattern-Generator* von Interesse, der Protokolltelegramme mit Geschwindigkeiten von bis zu 50 Mbit/s erzeugt. So haben Entwickler die Möglichkeit, neben fest vorgefertigten Nachrichten für die unterstützten seriellen Protokolle individuelle Signalmuster frei zu programmieren. Im R&S®HMO1002 ist eine hardware-unterstützte Signaltriggerung und -dekodierung mit den gängigsten Protokollen (I2C, SPI, UART, CAN oder LIN) integriert. Die *Serielle Busanalyse* ist optional erhältlich, aber jederzeit freischaltbar.

Rohde & Schwarz verlost zehn R&S®HMO1002 mit der höchsten Bandbreite unter den Elektor-Lesern! Registrieren Sie sich noch heute unter www.elektor.de/hmo1002 und gewinnen Sie das von Hameg Instruments entwickelte brandneue Mitglied der Scope-of-the-Art-Familie!

Dank der 128k Messpunkte hält das R&S®HMO1002 bei seinen *FFT-Analysefunktionen* mit deutlich größeren Oszilloskopen mit. Die Darstellung von Zeitsignal, Messfenster und Analysebereich der FFT sowie dem Ergebnis auf einem Bildschirm erleichtert das Ausmessen der Spektren.

Der separat erhältliche, aktive Logikastkopf R&S®HMO3508 ist nicht an ein Gerät gebunden und kann mit allen Oszilloskopen der R&S®HMO-Serie verwendet werden. Das R&S®HMO1002 bietet damit eine Zeitbereichs-, Logik-, Protokoll- und Frequenz-Analyse in einem Gerät und ist ein weiteres Mitglied der Rohde & Schwarz-Familie „Scope of the Art“. Die unverbindlichen Richtpreise liegen zwischen 798 € plus MwSt. für das günstigste Modell mit 50 MHz Bandbreite und 998 € plus MwSt. für das Gerät mit der höchsten Bandbreite von 100 MHz.

Hauptmerkmale

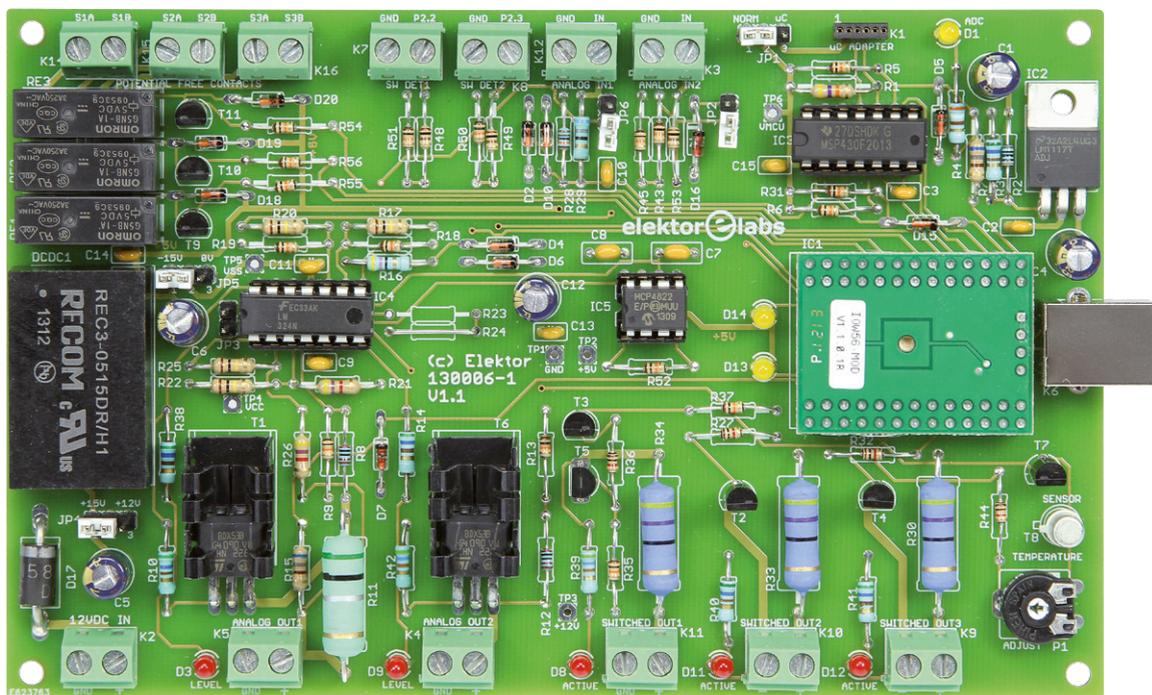
- Modelle mit 50 MHz, 70 MHz und 100 MHz Bandbreite
- 1 GSamples/s Sampling Rate, 1 MSample Speichertiefe
- Hohe vertikale Empfindlichkeit bis zu 1 mV/Div
- Schnelle Erfassungsraten bei der Signalfehlersuche mit 10.000 Waveforms/s
- Große Auswahl an Auto-Messfunktionen
- Quickview: alle wichtigen Signalparameter auf Knopfdruck
- Mixed-Signal-Funktionalität standardmäßig vorhanden
- Zielgerichtet auf Signalereignisse triggern
- Vielseitige Busanalyseoptionen zur Isolierung spezifischer Datenpakete
- FFT: der einfache Weg zur spektralen Analyse mit 128k Messpunkten
- Digitales Voltmeter für gleichzeitiges Messen auf beiden Analogkanälen
- Das richtige Signal zur Hand: Mustergenerator bis 50 Mbit/s oder Funktionsgenerator bis 50 kHz

IO-Warrior Erweiterungs-Board

Messen, Steuern und Regeln mit dem PC

Von
Franz Peter Zantis

Alte PCs oder Notebooks müssen nicht als Elektroschrott enden oder im Keller verstauben! Die Ressourcen kann man sinnvoll nutzen: Mit einer universellen Interfacekarte wird der Computer zur Mess-Steuer-Regel-Station. Ein IO-Warrior-Modul auf dem Board übernimmt die USB-Kommunikation. Dazu gibt's Quellcode für das kostenlose Visual Studio Express.



Es gibt viele MSR-Interfacekarten zu kaufen, alle mit spezifischen Vor- und Nachteilen. Die hier vorgestellte Interfacekarte kann gleich mehrmals punkten. Die Ansteuerung mit bekannten Programmiersprachen wie VisualBasic, C# oder C++ ist einfach, Sensoren und Verbraucher können ohne Spezialstecker direkt über Kabel-Anschlussklemmen angeschlossen werden, Modifikationen und unterschiedlichste Anwendungen sind mit einer immer gleichen Hardware besonders schnell erstellt.

Vorüberlegungen

Es gibt einige Grundbedingungen im Pflichtenheft dieser Interfacekarte. Die Interfacekarte soll das gängige Europakartenformat haben und über den an allen PCs vorhandenen **USB** angeschlossen werden. Zur USB-Protokollverarbeitung wird ein Chip namens IO-Warrior56 der deutschen Firma Code Mercenaries eingesetzt, der diese Arbeit ohne Programmieraufwand übernimmt. Der Baustein bringt eine Vielzahl von Funktionen mit, die sich leicht per Visual-Basic-Programm ansprechen lassen [6].

Es müssen **Ausgänge** zur Verfügung stehen, an die Verbraucher für Kleinspannungen bis 12 V direkt und ohne weitere Hardware angeschlossen und mit Energie versorgt werden können. Dazu muss eine externe Stromversorgung (Steckernetzteil) anschließbar sein. Es wurden zwei Ausgänge mit per Programmcode einstellbarer Ausgangsspannung zwischen 0 V und +12 V vorgesehen. Außerdem sind drei Schaltausgänge vorhanden, von denen einer gegen Masse schaltet. Man kann Verbraucher direkt oder auch über Relais anschließen. 230-V-Verbraucher lassen sich mit drei weiteren, potentialfreien Schaltkontakten schalten über Funkschalter/Funksteckdosen.

Über zwei **Eingänge** können Schalterstellungen erfasst werden; zwei analoge Spannungen lassen sich mit einem A/D-Wandler messen. Darüber hinaus verfügt die Interfacekarte über eine eigene Temperaturerfassung mit einem alten Silizium-Transistor im Metallgehäuse als Sensor.

Umsetzung

Diese Vorüberlegungen münden in einer Schaltung, wie sie in **Bild 1** zu sehen ist. Um mit der Interfacekarte zu arbeiten, wird eine Programmiersprache benötigt, im folgenden VisualBasic. Das *Microsoft Visual Studio Express*, das VB zusammen mit C# oder C++ enthält, kann kostenlos von der Microsoft-Homepage [1] heruntergeladen und verwendet werden. Die Programmierung gelingt sowohl unter Windows als auch unter Linux. Einen Einstieg in die Programmierung mit VisualBasic erhält man zum Beispiel in [2] und [3].

Den USB-Chip gibt es in zwei Varianten, als SMD im MLFP56-Gehäuse und als fertiges Modul (**Bild 2**), an dem direkt ein USB-Kabel angeschlossen werden kann. Damit wir uns nicht mit den 56 Beinchen des SMD herumschlagen müssen, verwenden wir das Modul. Es wird einfach kopfüber in eine Fassung gesteckt.

Die Stromversorgung der Interfacekarte erfolgt über die USB-Verbindung und einen auf dem Board befindlichen DC/DC-Wandler, der aus den +5 V des USB eine symmetrische Spannung von ± 15 V zur Versorgung der Operationsverstärker erzeugt. Genau genommen ist die negative Spannung nicht unbedingt erforderlich, da der eingesetzte Operationsverstärker LM324 den Ausgang bis auf wenige Millivolt an GND einstellen kann. Mit den Jumpfern JP4 und JP5 kann man die Spannungsversorgung auf ± 15 V oder auf +12 V/GND einstellen. Im zweiten Fall kann man an den analogen Ausgängen aber höchst-

tens noch +9 V erreichen, da die Ausgänge der Operationsverstärker etwa 3 V Abstand zur positiven Versorgung wahren. Dafür würde aber der DC/DC-Wandler entfallen. Zwei gelbe LEDs (D13/D14) an den Pins des IO-Warriors können als Betriebszustandsanzeigen genutzt werden.

Insgesamt ist die Interfaceplatine sehr übersichtlich und vor allem nachbaufreundlich gestaltet. SMDs haben wir gar nicht erst verwendet, so dass der Aufbau auch nicht so erfahrenen Löttern gelingen sollte. Alle ICs bis auf den Regler werden in Fassungen gesetzt, auch das USB-Modul. Wichtig ist aber, dass hier Buchsenleisten allerhöchster Qualität zum Einsatz kommen müssen, um sämtlichen Kontaktschwierigkeiten von vornherein aus dem Weg zu gehen.

Die Leerplatine 130006-1 ist im Elektor-Shop [4] zu haben. Auch das IO-Warrior-Modul (130006-91) kann man unkompliziert dort bestellen, ebenso den als A/D-Wandler verwendeten programmierten Mikrocontroller von TI (130006-41). Zusätzlich steht dort ein Software-Paket 130006-11 zum Download bereit, das neben Bibliotheks-Modulen auch Beispielcode und die Beschreibung eines Experiments zur Temperaturmessung enthält.

Schaltausgänge

Sind die Bestückungsarbeiten abgeschlossen, gibt es noch einiges vorzubereiten, bevor man mit dem Programmieren beginnen kann. Zum IO-Warrior gehört die Programmibliothek `iowiki.t56.dll`. Sie stellt das Bindeglied zwischen der selbst erstellten PC-Software und dem IO-Modul dar (im Download [4]). Damit der Zugriff auf diese Datei funktioniert, hinterlegt man sie entweder in einem bekannten Pfad (Systempfad) des Betriebssystems oder im VisualBasic-Projekt. Letzteres bietet den Vorteil, dass man das fertige Programm in einem Ordner auf einen anderen Computer kopieren und dort ohne weitere Maßnahmen durch Doppelklick starten kann. **Bild 3** zeigt, wo die Datei innerhalb eines VisualBasic-Projektes (hier „Motorsteuerung“) hingehört. Dann sollte man ein VisualBasic-Modul in das Projekt einbinden, das Deklarationen von Funktionen und Subroutinen enthält und beim Programmieren sehr viel Recherche- und Tipparbeit erspart. Im **Bild 4** sind die Programmierressourcen für das Projekt „FanControl“ zu sehen. Wie man sieht, ist das `Module1.vb` im Projekt enthalten.

Um festzustellen, ob der IO-Warrior mit dem

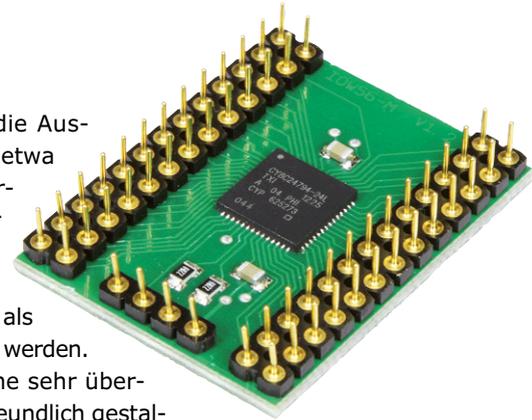


Bild 2. Als Modul ist der IO-Warrior gut handhabbar.

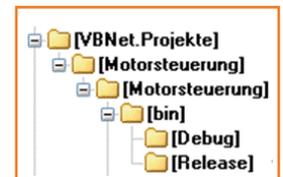


Bild 3. Struktur eines VB-Projekts.

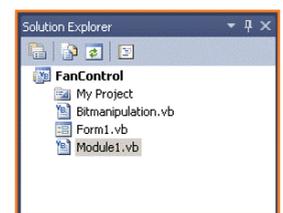
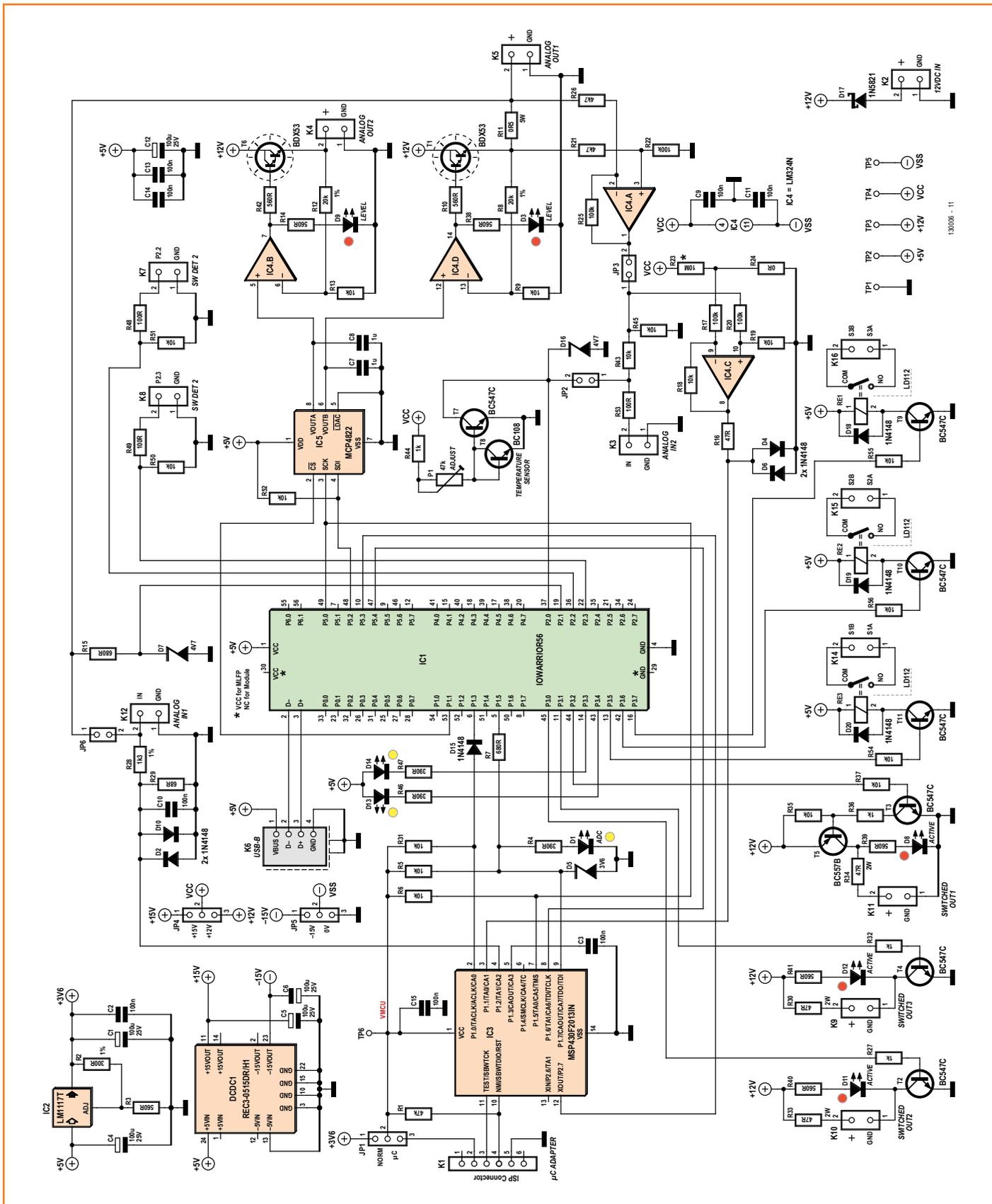


Bild 4. Einbindung des `Module1.vb`.



130008 - 11

Tabelle 1. Bedeutung der Parameter der Funktion IowKitWrite()

intret	eine Integer-Variable, die den Rückgabewert aufnimmt; dieser enthält nach der Ausführung die Anzahl der übertragenen Bytes
IOhandle	die Handle-Nummer, die mit Hilfe IowKitOpenDevice() bereits ermittelt wurde (siehe weiter oben)
0	hier wird an dieser Stelle im Standard-I/O-Betrieb immer eine Null eingetragen
I0data(0)	das vorbereitete Array mit dem Index ab dem die Übertragung starten soll
8	die Anzahl zu übertragenen Bytes; immer 8

Computer verbunden ist, verwendet man die Funktion `IowKitOpenDevice`. Mit dieser Funktion wird der Zugang zum IO-Warrior geöffnet und gleichzeitig die Handle-Nummer ermittelt. Ist diese ungleich 0, wurde der Baustein gefunden, bei 0 nicht. Eine Fehlermeldung sollte den Anwender darüber informieren (**Listing 1**). Auch während der Programmausführung sollte man ab und zu diese Funktion aufrufen, um sicherzustellen, dass die Hardware noch angeschlossen ist. Wird die USB-Verbindung kurzzeitig unterbrochen, funktioniert die Kommunikation zwischen Hard- und Software nicht mehr. Auch in diesem Fall muss zunächst `IowKitOpenDevice` aufgerufen werden, um die Verbindung wieder herzustellen. Man könnte den Aufruf etwa hinter einen Button mit der Aufschrift „Connect Hardware“ legen. Damit sind die Vorbereitungen beendet und das Programmieren kann beginnen. Zuerst testen wir die Schaltausgänge. Der IO-Warrior bietet sechs Ports (Port 0...5) mit jeweils acht I/O-Pins (jeweils Px.0 bis Px.7) sowie den Port 6 mit zwei I/O-Pins. Die Pins P5.0...P5.4 sind für den SPI-Bus reserviert.

Für das Schalten von Verbrauchern sowie der LEDs D13 und D14 ist Port 3 vorgesehen. Ein erstes Projekt soll die beiden LEDs ansteuern, um zu erläutern, wie die einzelnen I/O-Pins des IO-Warriors von Visual Basic angesteuert werden. Der IO-Warrior kennt einen Standard I/O-Betrieb und einen Spezialbetrieb (SPI/I²C). Über die grundsätzliche Arbeitsweise des SPI informiert [5]. Zur Ansteuerung der I/O-Pins wird der Standardbetrieb verwendet. Die betreffenden LEDs sind mit der 5-V-Versorgungsspannung des USB und über 390-Ohm-Widerstände mit P3.3 und P3.4 verbunden.

Bild 1. Die Schaltung der Interfacekarte mit vielen Ein- und Ausgängen.

Ein Byte-Array bestimmt, welche Pins eines Ports mit High- oder Low-Level belegt werden. Grundsätzlich werden immer alle Zustände der Pins in einem 8-Bit-Byte an den IO-Warrior gesendet. Zusätzlich wird das erste Byte des Arrays für die Kennung des Normalmodus verwendet. Insgesamt muss das Array also acht Bytes fassen können. Bei der Zählweise ab 0 sieht die Deklaration also wie folgt aus:

```
Dim I0data(7) As Byte
```

Je nach Aufgabenstellung kann man die Bytes vorbesetzen. Die LEDs sollen zunächst nicht leuchten, die zugehörigen Pins müssen auf logisch high gelegt werden. Das Bitmuster ist dann MSB00011000. Die Übergabe kann als Dezimalzahl 24 oder als Hexadezimalzahl 18 (in BASIC &H18) erfolgen. Die Vorbesetzung des Arrays stellt sich dann wie in **Listing 2** dar. Es fällt auf, dass Port1 (`I0data(2)`) mit lauter Einsen besetzt ist. Dieser Port ist für ChipSelect (CS) der am SPI-Bus angeschlossenen Bausteine vorgesehen. CS wird durch einen Sprung von High nach Low ausgelöst.

Das Array wird nun zum IO-Warrior gesendet. Dazu steht die Funktion `IowKitWrite` zur Verfügung:

```
intret = IowKitWrite(IOhandle, 0,
I0data(0), 8)
```

Die Bedeutung der Parameter sind in **Tabelle 1** erläutert.

Um die LEDs einzuschalten, müssen die Pins 3.3 und 3.4 auf Masse (GND) gelegt werden. Sie sollen beim Klicken auf einen Start-Button 100 Mal im Sekundenrhythmus blinken. Dazu ist eine Schleife mit fest vorgegebener Anzahl von Durchläufen zu programmieren. Das Programm dazu könnte wie in **Listing 3** aussehen.

So werden auch alle anderen Schaltausgänge angesteuert. Um einen an OUT1...OUT3 angeschlosse-

Listing 1. Die Funktion IowKitOpenDevice.

```
IOhandle = IowKitOpenDevice()
  If IOhandle = 0 Then
    MessageBox.Show("No hardware found!", "ERROR", MessageBoxButtons.OK, _
      MessageBoxIcon.Error, MessageBoxDefaultButton.Button1)
  End
Else
```

- Code der ausgeführt werden soll, wenn die Hardware erkannt wurde -

```
End If
```

Listing 2. Array für die Vorbelegung der Ports.

```
IOdata(0) = 0      'Byte für Parametrisierung im Normalmodus immer 0
IOdata(1) = 0      'Byte für Port0
IOdata(2) = 255    'Byte für Port1; alle auf 1, da dieser Port für Chip-Select
verwendet wird
IOdata(3) = 0      'Byte für Port2
IOdata(4) = &H18   'Byte für Port3; Anschlüsse der LEDs D13 und D14 auf logisch
high
IOdata(5) = 0      'Byte für Port4
IOdata(6) = 0      'Byte für Port5
IOdata(7) = 0      'Byte für Port6
```

Listing 3. 100 Mal auf Knopfdruck blinken.

```
Dim IOdata(7) As Byte 'Array mit den Portzuständen
Dim IOHandle As Integer 'Behälter für die Handle-Nummer
Dim intret As Integer 'Behälter für Integer-Rückgabewert
Dim n As Integer      'Zählvariable

Private Sub BtnStart_Click(sender As System.Object, e As System.EventArgs) _
Handles BtnStart.Click
  IOhandle = IowKitOpenDevice()
  If IOhandle = 0 Then
    MessageBox.Show("No hardware found!", "ERROR", MessageBoxButtons.OK, _
      MessageBoxIcon.Error, MessageBoxDefaultButton.Button1)

    End 'Programmabbruch falls keine Hardware angeschlossen ist
  Else
    For n = 1 to 100 'For-Next-Schleife
      IOdata(4) = &H18 'LEDs D13 und D14 auf +5V: LEDs sind dunkel
      intret = IowKitWrite(IOhandle, 0, IOdata(0), 8)
      System.Threading.Thread.Sleep(1000) '1000 ms warten

      IOdata(4) = 0 'alle Pins von Port3 auf GND: LEDs leuchten
```

```

intret = IowKitWrite(IOhandle, 0, IOdata(0), 8)
System.Threading.Thread.Sleep(1000) '1000 ms warten
Next n
End If
End Sub

```

Listing 4. Fernbedienung einer Funksteckdose.

```

Dim IOdata(7) As Byte 'Array mit den Portzuständen
Dim IOHandle As Integer 'Behälter für die Handle-Nummer
Dim intret As Integer 'Behälter für Integer-Rückgabewert

Private Sub BtnStart_Click(sender As System.Object, e As System.EventArgs) _
Handles BtnStart.Click
IOHandle = IowKitOpenDevice()
If IOHandle = 0 Then
    MessageBox.Show("No hardware found!", "ERROR", MessageBoxButtons.OK, _
    MessageBoxIcon.Error, MessageBoxDefaultButton.Button1)

    End 'Programmabbruch falls keine Hardware angeschlossen ist
Else
IOdata(4) = &H20 'P3.5 auf logisch 1; Relais zieht an
intret = IowKitWrite(IOhandle, 0, IOdata(0), 8)
System.Threading.Thread.Sleep(500) '500 ms warten
IOdata(4) = 0 'alle Pins von Port3 auf GND; Relais fällt ab
intret = IowKitWrite(IOhandle, 0, IOdata(0), 8)
End If
End Sub

```

nen Verbraucher einzuschalten, müssen P3.0, P3.1 und P3.2 jeweils logisch 1 führen. OUT1 schaltet gegen Masse (GND), OUT2 und OUT3 gegen +12 V. Die Schottky-Diode D17 dient als Verpolschutz. Um die potentialfreien Kontakte K14, K15 und K16 zu schließen, sind ebenfalls logische Einsen an P3.5, P3.6 und P3.7 erforderlich. Mit einem Funkschalter und Funksteckdosen lassen sich interessante Dinge anstellen, ohne dass man selbst oder das Interface mit Netzspannung in Berührung kommt. Dazu öffnet man die Fernbedienung und verbindet ihre Taster mit den potentialfreien Schaltkontakten der Interfaceplatine. Die Kontakte schließen beim Klicken auf den Start-Button nur für einen kurzen Moment (500 ms, siehe **Listing 4**).

Analogausgänge

An den Klemmen K4 und K5 (Analog OUT1 und Analog OUT2) können mit einem doppelten Digital/Analog-Wandler (IC5, MCP4822) mit SPI-Bus

beliebige Spannungen zwischen 0 V und +12 V erzeugt werden. Gesendet wird ein 16-bit-Integerwert (Bit 0...15), wobei die unteren 12 Bit (Bit 0...11) den gewünschten analogen Wert enthalten und die oberen 4 Bit (Bit 12...15) die Einstellparameter (**Tabelle 2**).

Der Ausgang VOUTA ist mit dem Operationsverstärker IC4.B verbunden, der die Spannung von VOUTA um den Faktor 3 verstärkt. Der Rückkopplungswiderstand R12 ist mit dem Spannungsausgang K4 verbunden. Damit kann der Operationsverstärker die durch die BE-Strecke von T6 verursachten Verluste ausgleichen. An K4 kann man also immer die 3-fache Spannung von VOUTA messen. Das sind maximal $4,096 \text{ V} \times 3 = 12,288 \text{ V}$. Um diesen Wert an K4 zu erreichen, muss die über K2 eingespeiste Spannung deutlich höher (+15V) sein, da an T6 die Kollektor-Emitter-Sättigungsspannung U_{CEsat} (bis zu 2 V) abfällt.

Tabelle 2. Einstellungsbits für den D/A-Wandler

Bit 15	0: die Daten werden an DAC A gesendet (VOUTA im Schaltbild)
	1: die Daten werden an DAC B gesendet (VOUTB im Schaltbild)
Bit 14	dieses Bit ist nicht relevant und kann ignoriert werden
Bit 13	0: Spannungsbereich am Ausgang ist 0...4096 mV
	1: Spannungsbereich am Ausgang ist 0...2048 mV
Bit 12	0: Output-Buffer ist nicht aktiv (der zuletzt gesendete Wert wird nicht gespeichert)
	1: Output-Buffer ist aktiv (der zuletzt gesendete Wert wird gespeichert)

An den Transistoren können hohe Verlustleistungen auftreten, es wurden aber nur kleine Kühlkörper vorgesehen. Je kleiner die Ausgangsspannung und je größer der Ausgangstrom, desto größer ist die Verlustleistung. Von der an K2 eingespeisten Spannung gehen 400 mV an D17 verloren, der Rest steht an den Kollektoren der beiden Transistoren an. Die LEDs D9 und D3 deuten die Ausgangsspannung mit ihrer Helligkeit an. Zudem sind die Ausgänge nicht kurzschlussfest. Es ist also sinnvoll, eine abgesicherte Stromversorgung einzusetzen. Insgesamt sollte man die Ausgänge stets mit Bedacht gebrauchen.

Die Ansteuerung der Ausgänge erfolgt durch Übergabe einer Zahl von 0 bis 4096 an den jeweiligen D/A-Wandler (VOUTA oder VOUTB). Die gesendete Zahl multipliziert mit 3 ergibt die Ausgangsspannung in Millivolt. Die Auflösung beträgt somit 3 mV. Es sind bei der Programmierung fünf Punkte zu berücksichtigen:

16-bit-Integerzahl vorbereiten

(4 Steuerbits + 12-Bit-Wert für die gewünschte Spannung). Eine praktikable Methode ist es, den gewünschten Spannungswert in die Variable zu laden und anschließend die oberen vier Steuerbits einzustellen. Die Übertragung vom USB-Chip über SPI funktioniert nur byteweise, so dass die Integerzahl in Highbyte und Lowbyte zerlegt werden muss. Das Modul `Bitmanipulation.vb` im Downloadpaket [4] enthält Funktionen, mit denen einzelne Bits innerhalb eines Byte gesetzt, gelöscht, gewechselt oder abgefragt werden können und außerdem eine Funktion zur Zerlegung von vorzeichenlosen 16-bit-Integerwerten in Highbyte und Lowbyte.

An Ausgang K4 (ANALOG OUT2) soll eine Spannung von 3,6 V erscheinen. Der Programmcode muss einen Wert von $3600 \text{ mV} / 3 = 1200 \text{ mV}$ an den D/A-Wandler senden (**Listing 5**). Nach Ausführung stehen Highbyte und Lowbyte für den D/A-Wandler im Array `msb_lsb` bereit.

SPI-Bus einstellen

Jetzt muss der SPI-Bus initialisiert werden. Die Taktfrequenz und einige andere Parameter, die Takt und Daten betreffen, werden eingestellt, alle Initialisierungsdaten in ein Byte-Array von 64 Byte abgelegt und dann zum Chip gesendet (**Listing 6**).

Da der SPI-Bus verwendet wird, muss der IO-Warrior in der Spezial-Betriebsart arbeiten. Deshalb sind die Parameter der Funktion `IowKitWrite` anders als bei der Einstellung der normalen I/O-Ports. `IOhandle` hat die gleiche Bedeutung, dann folgt eine 1, die den Spezialmodus einstellt, dann das Array mit dem Index, ab dem übertragen werden soll. Die 64 bezieht sich auf die Anzahl der zu übertragenden Bytes.

Eine Bemerkung noch zu `w56init(4)`: Der USB-Chip wird mit 5 V versorgt, der als A/D-Wandler arbeitende Mikrocontroller (siehe unten) über IC2 mit 3,6 V. Damit der Mikrocontroller nicht überfordert wird, müssen die SPI-Leitungen durch Deaktivieren der Pullup-Widerstände im IO-Warrior von der Spannungsversorgung des USB-Chips getrennt werden.

ChipSelect aktivieren

Anschließend wird der D/A-Wandler über Chip-Select (Pin 2 des Wandlers an P1.1 des IO-Warriors) aktiviert (**Listing 7**).

16-bit-Integerzahl senden

Vor der Übertragung werden auch die zu übertragenden Daten in ein Byte-Array mit 64 Plätzen abgelegt. Es sind weitere drei Einstellbytes notwendig, darunter die Anzahl der zu sendenden Bytes (**Listing 8**). Anschließend kann man an K4 tatsächlich 3,6 V messen.

ChipSelect deaktivieren

Formsache, aber dennoch notwendig ist es, den Wandler über CS wieder abzuschalten (**Listing 9**). Sinnvollerweise würde man den Code in eine

Listing 5. Bereitstellung der Werte für den D/A-Wandler.

```

Dim curspgch2 As UShort 'Variable zur Aufnahme des gewünschten Spannungswertes (16-Bit-Integer)
Dim vmsblsb(1) As Byte 'Array für die Aufnahme des Highbyte und Lowbyte
Dim curspgch2 As UShort = 1200 'gewünschte Ausgangsspannung in mV geteilt durch 3
vmsblsb = findHbyteLbyte(curspgch2)'Zerlegen in Highbyte und Lowbyte mit Hilfe von "Bitmanipulation.vb"

'jetzt folgt die Einstellung der Steuerbits für den DAU gemäß Tabelle 3.3
msblsb(1) = ClearBit(msblsb(1), 7) 'Bit15: die Daten werden an DACA gesendet
msblsb(1) = ClearBit(msblsb(1), 6) 'Bit14: ist nicht relevant
msblsb(1) = ClearBit(msblsb(1), 5) 'Bit13: Ausgangsspannungsbereich auf 0...4096 eingestellt
msblsb(1) = SetBit(msblsb(1), 4) 'Bit12: 1=OutputBuffer aktiv

```

Listing 6. Initialisierung des SPI für den D/A-Wandler.

```

Dim w56init(63) As Byte 'Byte-Array mit 64 Plätzen
'SPI-Initialisierung
w56init(0) = &H8 'SPI-Modus einstellen
w56init(1) = &H1 'SPI einschalten
w56init(2) = &H0 'SPI-Einstellungen passend für die Kommunikation mit DAU MCP4822
w56init(3) = 119 'Taktfrequenz 24MHz / (119+1) = 200 kHz
w56init(4) = &H1 'Abschalten der Pullup-Widerstaende
retval = IowKitWrite(IOhandle, 1, w56init(0), 64) 'Senden der Initialisierungsdaten an den IO-Warrior

```

Listing 7. CS für den Wandler.

```

'ChipSelect von IC5 (P1.1) von High nach Low schalten
w56data(2) = ClearBit(w56data(2), 1)
retval = IowKitWrite(IOhandle, 0, w56data(0), 8)

```

Listing 8. Senden der Daten zum Wandler.

```

Dim w56SPIData(63) As Byte 'Byte-Array mit 64 Plätzen

'Senden der Daten über SPI an den DAU
w56SPIData(0) = &H9 'bei SPI immer &H9
w56SPIData(1) = &H2 'Anzahl der zu sendenden Byte; hier 2 (Highbyte und Lowbyte)
w56SPIData(2) = &H0 'Flags hier auf 0
w56SPIData(3) = vmsblsb(1) 'Einladen des Highbyte in das zu sendende Array
w56SPIData(4) = vmsblsb(0) 'Einladen des Lowbyte in das zu sendende Array
retval = IowKitWrite(IOhandle, 1, w56SPIData(0), 64)

```

Listing 9. Abschalten des Wandlers.

```

'ChipSelect IC5 (P1.1) wieder zurück auf High setzen
w56data(2) = SetBit(w56data(2), 1)
retval = IowKitWrite(IOhandle, 0, w56data(0), 8)

```

Funktion schreiben, der man dann lediglich die gewünschte Ausgangsspannung übergibt.

Die Kondensatoren C7 und C8 an den Ausgängen des D/A-Wandlers verhindern, dass irgendwelche Spikes an ANALOG OUT2 erscheinen. Soll sich die Ausgangsspannung schnell ändern können, müssen diese Kondensatoren aber verkleinert werden (z.B. auf 100 nF). Die Softwarebeispiele steuern ANALOG OUT2, genau so geht es mit ANALOG OUT1 (K5). Dazu muss man lediglich das 15. Bit der vier Steuerbytes des D/A-Wandlers ändern.

Lesen von Schaltzuständen

K7 und K8 sind über Schutzwiderstände direkt mit P2.2 und P2.3 verbunden. Man kann ermitteln, ob an diesen Klemmen High- oder Low-Signal anliegt. Bei einer 5-V-Versorgungsspannung erkennt der IO-Warrior sicher ein High-Signal, wenn die Spannung am Port 3,25 V übersteigt, eventuell reichen schon 2,25 V.

Ebenso kann man über P2.1 feststellen, ob eine Spannung an K5 anliegt, wenn diese mindestens High-Level erreicht. Das kann nützlich sein, wenn man K5 nur für Ausgangsspannungen größer als 4 V benutzt. Der Widerstand R15 und die Diode D7 schützen den Eingang P2.1 des USB-Chips vor Überspannung.

Zum Lesen der Eingangszustände werden die betreffenden Pins zunächst hochohmig (auf High) geschaltet. Jeder Pin ist intern über einen Widerstand (4...8 kΩ) mit +Vcc (5 V) und dem Kollektor eines Transistors gekoppelt, dessen Emitter auf GND liegt. Der Transistor wird hochohmig geschaltet und der Pegel per Programmcode festgestellt. Dazu steht unter anderem die Funktion `IowKitReadNonBlocking` zur Verfügung (die anderen Read-Funktionen können in [3] oder [5] nachgelesen werden). Zunächst werden die benötigten Variablen deklariert, dann alle Pins des Port 2 auf High geschaltet, schließlich die Read-Funktion aufgerufen (**Listing 10**).

Das Ergebnis befindet sich nun im Array `w56data`.

Durch einfaches Auslesen und Vergleichen im Programmcode kann man feststellen, welcher Pin auf High und welcher auf Low liegt. Es werden zwar immer die Zustände aller Pins eingelesen, hier ist aber nur Port 2 für die Detektion von Low/High-Zuständen vorgesehen. Es reicht also, wenn man nur das Byte in `w56data(3)` untersucht.

Erfassen von Analogwerten

Zur Erfassung von analogen Werten verfügt die Interfacekarte über einen Mikrocontroller (MSP430F2013), der als 16-bit-Analog/Digital-Wandler programmiert ist. Die Abtastfrequenz beträgt 488 Hz. Diese Vorgehensweise erlaubt die Modifizierung der Eigenschaften durch eine Änderung der Firmware dieses Controllers. Dies soll aber hier kein Thema sein.

Der A/D-Wandler besitzt zwei Eingänge (Pin 3 und Pin 4). Die analoge Eingangsspannung schwankt zwischen 0 und 600 mV entsprechend $0 \dots 2^{16}$ (0...65535). Pin 3 ist mit dem Ausgang des Operationsverstärkers IC4C verbunden. R16, D4 und D6 schützen den Eingang des A/D-Wandlers. Spannungen über 700 mV können also nicht auftreten. R16 ist so niederohmig, dass er schützt, ohne den zu messenden analogen Wert nennenswert zu beeinflussen. IC4C ist als Differenzverstärker beschaltet, wobei aber zunächst der invertierende Eingang auf GND liegt. Man könnte die Widerstände R23 und R24 verändern und eine Gleichspannung anlegen, die von der am positiven Eingang anliegenden zu messenden Spannung abgezogen wird, um beispielsweise auf die 600 mV Vollaussteuerung zu kommen. Die Verstärkung beträgt 0,1. Man kann den nichtinvertierenden Eingang des Differenzverstärkers auf verschiedene Weise verwenden – je nach Jumperstellung JP2 und JP3.

Um den Ausgangsstrom von ANALOG OUT1 (K5) zu messen, ist JP3 gebrückt und JP2 nicht. IC4A ist wie IC4C als Differenzverstärker beschaltet, der die Spannung am Shunt-Widerstand R11 misst. Fließt ein Strom von 500 mA, erscheinen am Ausgang von IC4A 5 V. Mit IC4C werden

Listing 10. Lesen von Eingangszuständen.

```
Dim retVal As Integer
Dim w56data(7) As Byte
w56data(0) = 0 'Kennung für den "Simple-Mode"
w56data(3) = 255 'alle Pins von Port2 auf 1 (im hochohmigen Zustand)
retVal = IowKitReadNonBlocking(IoHandle, 0, w56data(0), 8)
```

daraus dann 500 mV, die der Wandler verarbeiten kann. Der Messbereich lässt sich leicht durch Ändern der Widerstände R21 und R26 anpassen. Setzt man 10 k Ω ein, erscheinen am Ausgang von IC4.A bei 500 mA nur noch 2,5 V beziehungsweise am Ausgang von IC4.C 250 mV.

Um eine analoge Spannung über ANALOG IN2 zu ermitteln, muss JP3 offen sein. Die Analogspannung wird über R43 und R45 halbiert und dann über IC4.C durch 10 geteilt. Legt man an K3 10 V an, werden daraus am Ausgang von IC4.C die gewünschten 500 mV. R53 schützt T7, wenn

Stückliste

Widerstände:

R1 = 47k
 R2 = 300 Ω , 1%
 R3,R10,R14,R38,R39,R40,R41,R42 = 560 Ω
 R4,R46,R47 = 390 Ω
 R5,R6,R9,R13,R18,R19,R31,R35,R37,R43,R45,R50,R51,R52,R54,R55,R56 = 10 k
 R7,R15 = 680 Ω
 R8,R12 = 20 k
 R11 = 0 Ω 5, 5 W
 R16 = 47 Ω
 R17,R20,R22,R25 = 100 k
 R21,R26 = 4k7
 R23 = 10 M (nicht bestückt)
 R24 = 0 Ω (Drahtbrücke)
 R27,R32,R36,R44 = 1 k
 R28 = 1k3, 1%
 R29 = 68 Ω
 R30,R33,R34 = 47 Ω , 3 W
 R48,R49,R53 = 100 Ω
 P1 = 47-k-Trimmpoti

Kondensatoren:

C1,C4,C5,C6,C12 = 100 μ , 25 V
 C2,C3,C9,C10,C11,C13,C14,C15 = 100 nF
 C7,C8 = 1 μ keramisch

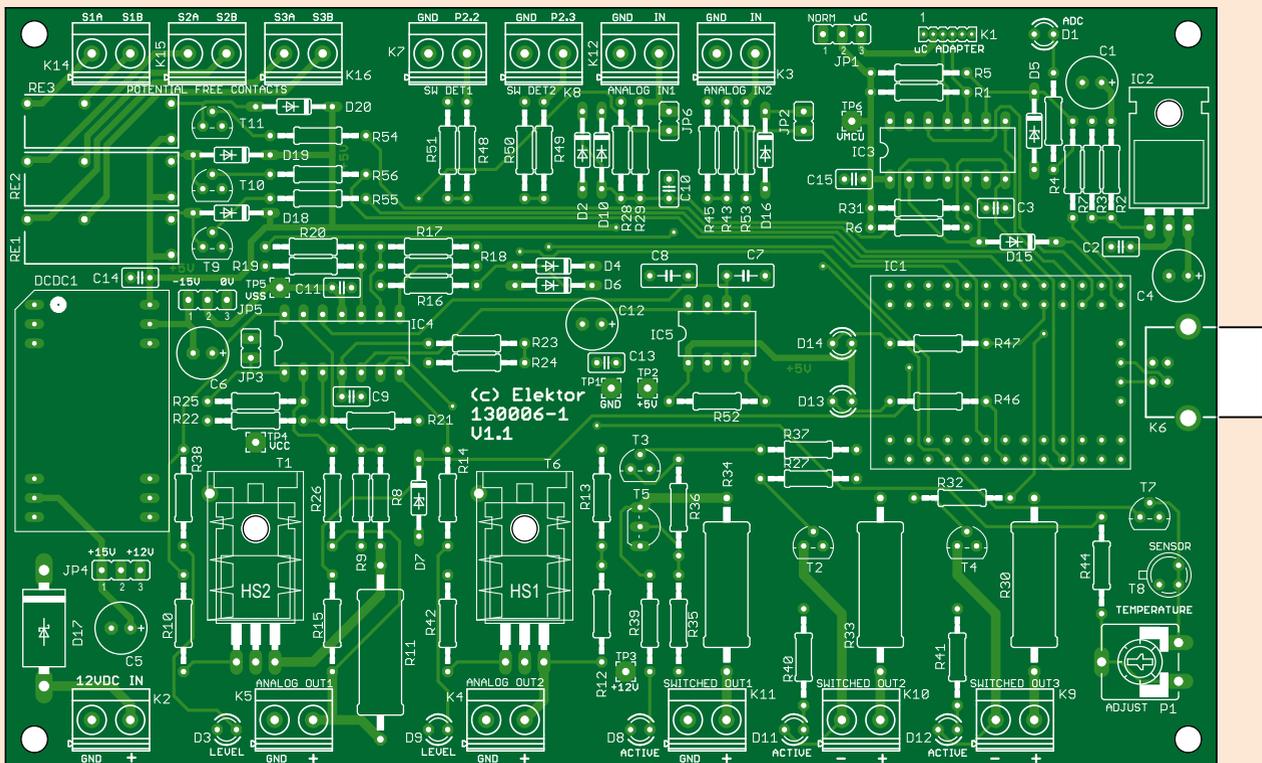
Halbleiter:

D1,D13,D14 = LED gelb, 3 mm
 D2,D4,D6,D10,D15,D18,D19,D20 = 1N4148
 D3,D8,D9,D11,D12 = LED rot, 3 mm
 D5 = Z-Diode 3V6
 D7,D16 = Z-Diode 4V7
 D17 = 1N5821
 T1,T6 = BDX53B
 T2,T3,T4,T7,T9,T10,T11 = BC547C
 T5 = BC557B
 T8 = BC108
 IC1 = IO-Warrior56-MOD (Version 1.1.0.1 oder höher)
 IC2 = LM1117T-ADJ
 IC3 = MSP430F2013IN
 IC4 = LM324N
 IC5 = MCP4822-E/P

DCDC1 = AM3N-0515D oder REC3-0515DR/H1

Außerdem:

JP1,JP4,JP5 = Stiftleiste 1x3, RM 2,54 mm
 K1 = Stiftleiste 1x6, RM 1,27 mm
 K6 = USB-Buchse, Typ B, gewinkelt
 JP2, JP3, JP6 = Stiftleiste 1x2, RM 2,54 mm + Jumper
 K2,K3,K4,K5,K7,K8,K9,K10,K11,K12,K14,K15,K16 = 2x1-Schraubklemme
 IC-Fassung DIP8
 IC-Fassung DIP14
 2x Buchsenleiste 2x13, RM 2,54 mm, für IC1, höchste Qualität
 1x Buchsenleiste 1x4, RM 2,54 mm, für IC1, höchste Qualität
 Kühlkörper für TO220, 21 K/W
 RE1,RE2,RE3 = 5-V-Relais SPST-NO
 Jumper
 Platine 130006-1 Version 1.1



man eine Spannung an K3 anlegt und versehentlich JP2 steckt.

Zur Messung der Spannung an ANALOG OUT1 (K5) muss JP6 gesteckt sein. Mit R15 und D7 kann man feststellen, ob an diesem Ausgang eine Spannung >4 V anliegt. Über den Jumper JP6 lässt sich der genaue Wert ermitteln. Bei gebrücktem JP6 gelangt die Spannung auf den Spannungsteiler R28/R29. Das Teilverhältnis ist 1:20. Liegen an K5 10 V an, dann erscheint parallel zu R29 eine Spannung von 500 mV, die nun am zweiten Eingang des Wandlers (Pin 4) anliegt. Auch dieser Eingang ist mit zwei Dioden (D2 und D10) gegen Überspannung gesichert.

Von außen zugeführte Analogspannungen können bei geöffnetem Jumper JP6 über ANALOG IN1 (K12) gemessen werden. Wie bei ANALOG IN2 (K3) beträgt das Teilverhältnis bis zum Eingang des Wandlers 1:20.

Es gibt also zahlreiche Möglichkeiten, analoge Werte zu erfassen. Der Wandler liefert pro Abfrage immer beide Analogwerte hintereinander – den Wert an Pin 3 und den an Pin 4. Die Übertragung erfolgt auch hier über den SPI-Bus. Der Pin 2 des Wandlers fungiert als ChipSelect.

Der A/D-Wandler erfasst zwar Analogwerte mit 488 Hz, liefert aber nur einen Wert auf Anfrage. Danach wechselt D1 den Zustand, so dass man

Listing 11. Initialisierung des SPI für den A/D-Wandler.

```
'SPI-Initialisierung
w56init(0) = &H8      'SPI-Modus einstellen
w56init(1) = &H1      'SPI einschalten
w56init(2) = &H4      'SPI-Einstellungen passend für die Kommunikation mit µC
MSP430F2013
w56init(3) = 119      'Taktfrequenz 24MHz / (119+1) = 200 kHz
w56init(4) = &H1      'Abschalten der Pullup-Widerstände
retval = IowKitWrite(IOhandle, 1, w56init(0), 64) 'Senden der
Initialisierungsdaten an den IO-Warrior
```

Listing 12. CS des A/D-Wandlers aktivieren.

```
'ChipSelect des ADU aktivieren
IOdata(2) = ClearBit(IOdata(2), 3)
retval = IowKitWrite(IOhandle, 0, IOdata(0), 8)
```

Listing 13. Übertragung der vier Bytes.

```
ldata(0) = &H9 'Indikator das SPI verwendet werden soll
ldata(1) = &H4 'Anzahl der zu übertragenden Bytes
ldata(2) = &H0 'Flags
retval = IowKitWrite(IOhandle, 1, ldata(0), 64)
retval = IowKitRead(IOhandle, 1, ldata(0), 64)
valuePin4 = ldata(2) * 256 + ldata(3) 'Wert an Pin 4 des ADU
valuePin3 = ldata(4) * 256 + ldata(5) 'Wert an Pin 3 des ADU
```

Listing 14. Abschalten des Wandlers.

```
'ChipSelect des ADU zurücksetzen (auf High-Level)
IOdata(2) = SetBit(IOdata(2), 3)
retval = IowKitWrite(IOhandle, 0, IOdata(0), 8)
```

Buchtipps

VisualBasic gehört dank seiner schnellen Erlernbarkeit und der einfachen Lesbarkeit zu den am meisten verbreiteten Programmiersprachen. Hier in diesem Artikel kann die Arbeit mit VB-Express nur angerissen werden; im neuen Elektor-Buch von F.P. Zantis: *VB-Express und die Hardware* (ISBN 978-3-89576-270-3) steht Experimenten mit VisualBasic Express und dem hier gezeigten universellen Interface-Board viel mehr Raum zur Verfügung. Das Buch ist für in der Elektronik bewanderte Einsteiger in VisualBasic.NET gedacht. Nach einer allgemeinen Einführung gibt es Kapitel über hardwarenahe Programmierung und den Einsatz von VisualBasic zum Messen, Steuern und Regeln - natürlich unter Berücksichtigung grafischer Oberflächen (GUI), die dem Ganzen ein ansprechendes Äußeres verleihen.



feststellen kann, dass der Wandler arbeitet. R7 und D5 sind notwendig, da beide Chips mit unterschiedlichen Versorgungsspannungen arbeiten.

Um die Daten des A/D-Wandlers über SPI zu erfassen, geht man wieder schrittweise vor:

SPI-Bus einstellen

Punkt 1 ähnelt dem Punkt 2 bei der Ansteuerung der Analogausgänge, einziger Unterschied ist `w56init(2)`, das jetzt den Wert 4 enthalten muss (**Listing 11**).

ChipSelect aktivieren

Punkt 2 unterscheidet sich nur durch den anzustuernden Pin1.3 (**Listing 12**).

Zwei 16-bit-Integerzahlen einlesen

Es werden zwei 16-bit-Werte (an Pin 4 und an Pin 3) entgegengenommen, insgesamt also 4 Bytes. Typisch bei SPI ist, dass zunächst ein Schreibvorgang ausgelöst werden muss, anschließend dann ein Lesevorgang. Zum Schluss stehen die vier Bytes in den Array-Plätzen 2...5. Sie müssen dann noch zu 16-bit-Integerwerten zusammengefügt werden (**Listing 13**).

Die entgegengenommenen Werte müssen noch richtig interpretiert werden. Liegen 600 mV an, ist der A/D-Wandler voll ausgesteuert. Alle 16 Bits haben den Wert 1. Dies entspricht $2^{16} = 65536$. Die Spannung in Millivolt am Pin4 des Wandlers beträgt

$$U_{\text{Pin4}} = 600 \text{ mV} \times (\text{Wert an Pin 4}) / 65536$$

Das Gleiche gilt für Pin 3. Um den tatsächlich an der Eingangsklemme liegenden Wert zu bestimmen, müssen natürlich noch die Spannungsteiler und Verstärker berücksichtigt werden.

ChipSelect deaktivieren

Schließlich wird ChipSelect deaktiviert (**Listing 14**).

Damit ist die Erfassung eines Datensatzes abgeschlossen.

Im Downloadpaket ist noch ein Dokument mit Programmcode zu finden, das die Anwendung der Temperatursensoren auf dem Board beschreibt.

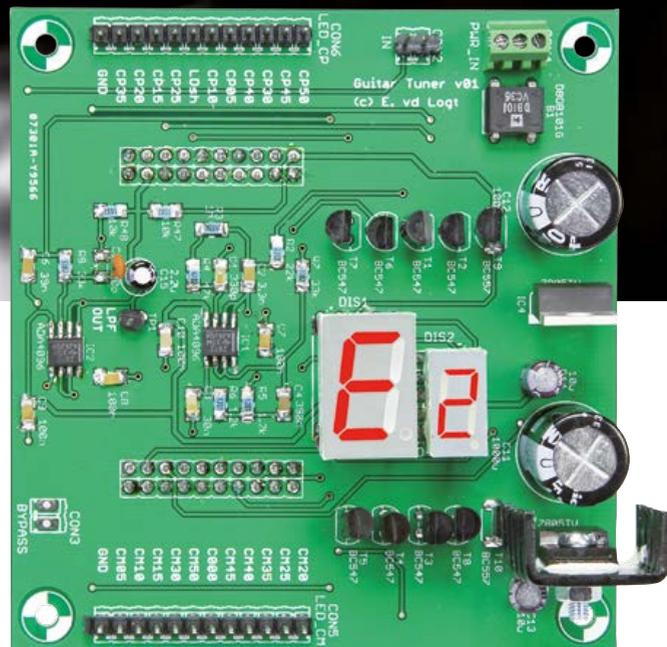
(130006)

Literatur und Weblinks:

- [1] www.microsoft.com/de-de/download/details.aspx?id=40787
- [2] Kurt Diedrich: VB Express 2010: Grundlagen für Visual Basic-Einsteiger und VB6-Umsteiger - ISBN 978-3895762697
- [3] Franz Peter Zantis: VB-Express und die Hardware - ISBN 978-3-89576-270-3
- [4] www.elektor-magazine.de/130006
- [5] Franz Peter Zantis: The SPI-Bus in general and in a special application
Präsentation auf der SEI-Tagung des Helmholtz-Zentrums am 11. März 2013 im Forschungszentrum Jülich -
<https://indico.desy.de/conferenceDisplay.py?ovw=True&confId=7078>
- [6] Code Mercenaries: IO-Warrior56 – Generic universal I/O-Controller for USB, V1.0.2,
September 22nd 2011 for chip version V1.1.0.1 - http://elmicro.com/files/codemercs/iowarrior56_datasheet.pdf

Gitarren-Tuner

Effizient und schnell



Die definitive Platine weicht in einigen Details vom hier dargestellten Prototyp ab.

Von
Emile van de Logt
(NL)

Gitarren stimmen ihre Instrumente nach Gehör, oder sie nutzen elektronische Hilfen. Ein solches Gerät selbst zu bauen, macht Spaß, gleichzeitig wird der Einblick in die Zusammenhänge vertieft. Die Grundfrequenz eines Gitarrensignals lässt sich mit der FFT bestimmen, doch dem sind in der Praxis Grenzen gesetzt. Hier wird eine andere Methode der Frequenzanalyse angewandt, die von einer Autokorrelationsfunktion Gebrauch macht. Kern der Hardware ist ein schneller Mikrocontroller ARM Cortex M4.

Ein im Fachhandel erworbenes, aber wenig zufriedenstellendes Gitarren-Stimmgerät war der Anlass für dieses Projekt. Das Bestimmen der Grundfrequenz eines Gitarrensignals kann doch nicht schwierig sein, und auch die Visualisierung

der Abweichungen von der Sollfrequenz erscheint leicht machbar. Die Idee nimmt Gestalt an, als ein *Launchpad Evaluation Board* TM4C123GXL [1] von Texas Instruments (**Bild 1**) in das Blickfeld gerät. Auf dem Board befindet sich ein Mik-

rocontroller vom Typ ARM Cortex M4, der nicht nur reichlich mit Eingangs- und Ausgangsleitungen bestückt ist, sondern auch Fähigkeiten eines Digitalen Signalprozessors (DSP) mitbringt. Ein unerwartet leistungsstarkes Board für erstaunlich wenig Geld: Erhältlich für 13 US-Dollar plus Versandkosten, unmittelbar bei Texas Instruments. In diesem Beitrag wollen wir zuerst einen Blick auf das Board werfen, um einen ungefähren Überblick über die enormen Möglichkeiten zu bekommen. Wir wollen das Entwicklungsziel und das dorthin führende Konzept im Auge behalten und auch zeigen, dass erfolgversprechende Wege in eine Sackgasse münden können. Deshalb erwähnen wir auch den ersten, mit der FFT arbeitenden Ansatz, danach beschreiben wir die definitive Realisierung.

TM4C123G Launchpad Evaluation Kit

Texas Instruments hat sich durch die qualifizierten Dokumentationen und Hilfen für seine Produkte einen Namen gemacht. Für dieses Entwicklungsboard sind ein *Wiki* mit *Workbook* [2], Demonstrationen und sogar Video-Clips verfügbar. Die Entwicklungsumgebung (*Integrated Development Environment, IDE*) heißt *Code Composer Studio (CCS)*, sie lässt keine Wünsche offen. Es geht los, nachdem wenige Handgriffe getan sind und das Board über USB an einen PC angeschlossen ist. Die Demonstrationen sind vorbildlich gestaltet, die Erklärungen sind klar und verständlich, sie werden von Beispielen und Video-Clips unterstützt. Zum Lieferumfang gehört ein vollständiges *Application Programming Interface (API)* mit dem Namen *Tiva*, das auf sämtliche I/O-Ports zugreifen kann.

Auf dem Entwicklungsboard befindet sich ein Mikrocontroller ARM Cortex M4, Typenbezeichnung TM4C123GH6PM, die maximale Taktfrequenz beträgt 80 MHz. Eine Fließkomma-Einheit (*Floating Point Unit, FPU*) mit DSP-Fähigkeiten, 256 KB Flash-Speicher, 32 KB SRAM, 2 KB EEPROM, acht UARTs, vier wandlungsfähige serielle Schnittstellen, vier I²C-Module, CAN, USB, zwei 12-bit-ADCs und eine Fülle von I/O-Ports sind die wichtigsten Highlights. Insbesondere die 12-bit-ADCs, die DSP-Fähigkeiten und die I/O-Peripherie kommen der Realisierung eines Gitarren-Tuners stark entgegen.

Erster Anlauf:

Fast Fourier Transformation (FFT)

Voller Enthusiasmus ob der beeindruckenden Fähigkeiten wurde begonnen, auf dem Board eine

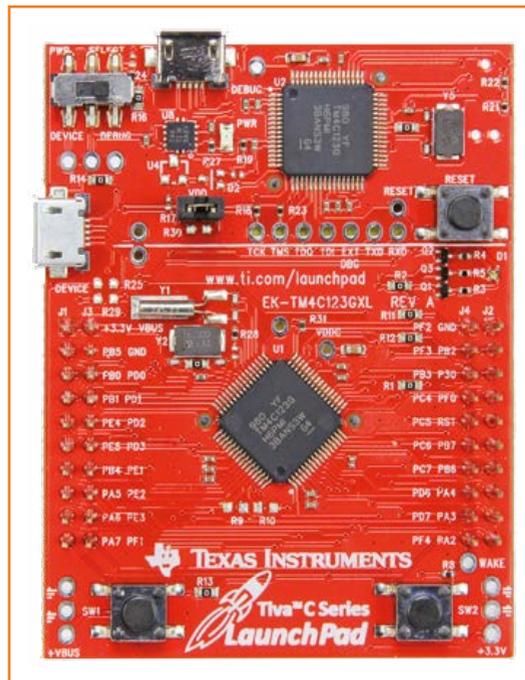


Bild 1. Das Entwicklungsboard TM4C123GXL von Texas Instruments hat einen Mikrocontroller ARM Cortex M4 an Bord (Foto: TI).

FFT zu implementieren. Für diesen Zweck existieren diverse, allgemein zugängliche Routinen, die unmittelbar übernommen werden konnten. Doch bei der Implementierung tauchte ein Problem auf: Das Spektrum eines Gitarrensignals hat eine Bandbreite von ungefähr 2 kHz. Nach dem Shannon-Theorem [3] muss die Abtastfrequenz mindestens 4 kHz betragen. Um eine FFT mit der

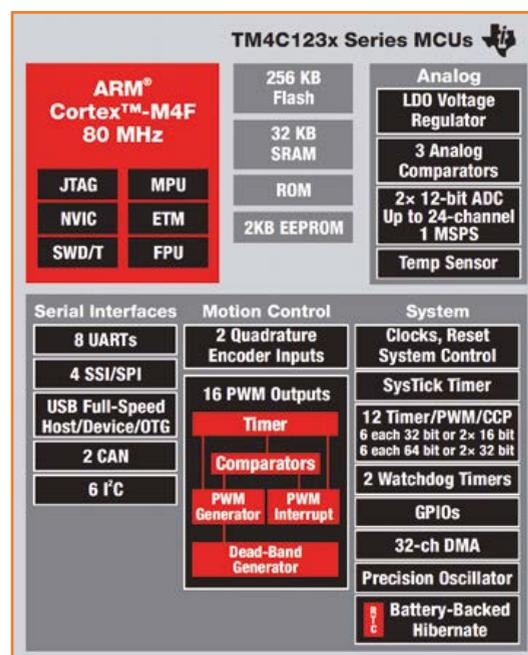


Bild 2. Schematische Übersicht der Funktionen und Anschlüsse des Entwicklungsboards TM4C123GXL (Grafik: TI).

Yin intern: Autokorrelation und parabolische Interpolation

Es ist nicht die Fourier-Transformation, die der Yin-Algorithmus benutzt, die Grundlage ist eine andere vielverwendete Methode der digitalen Signalverarbeitung: Die Autokorrelationsfunktion (siehe [5]). Darunter versteht man die Punkt-für-Punkt-Multiplikation eines Signals mit einer zeitverschobenen Version des gleichen Signals.

Angenommen das Eingangssignal ist weißes Rauschen im Wertebereich $-1...+1$. Wenn ein einzelner Wert mit einem anderen, zeitverschobenen Wert multipliziert wird, liegt auch das Ergebnis zwischen -1 und $+1$. Werden die Werte addiert und ist die Anzahl der Werte genügend groß, geht die Summe der Werte gegen 0. Das liegt daran, dass sich die Werte letztendlich gegenseitig aufheben.

Bei einem nicht zufälligen Signal wie dem Signal einer Gitarre verhält sich dies anders. Wenn hier die gleiche Operation ausgeführt wird, ist der typische Wert größer als 0, er hängt vom Maß der Verschiebung ab. Der Wert erreicht sein Maximum, wenn der verschobene Wert eine Periode später mit dem originalen Signal zusammenfällt.

Von dieser Gesetzmäßigkeit macht der Yin-Algorithmus Gebrauch.

In Schritten der Sample-Frequenz, hier $1/6000$ s, wird das Signal verschoben und mit sich selbst multipliziert. Anschließend wird betrachtet, für welchen Wert der Verschiebung die Autokorrelation am größten ist. Dies ist ein Maß für die wichtigste Frequenz im Signal.

Der Yin-Algorithmus wendet noch einige Verfeinerungen an, um die Frequenzbestimmung genauer zu gestalten, beispielsweise die parabolische Interpolation. Angenommen der Yin-Algorithmus ergibt, dass die Autokorrelation bei einem Signal ihr Maximum hat, das um 16 Samples verschoben ist. Das entspricht der Frequenz $6000/16$ Hz = 375 Hz. Die nächste Frequenz, bei der dies ebenfalls zutrifft, ist $6000/15$ Hz = 400 Hz. Diese Frequenzen sind stets gleich einem Vielfachen der Sample-Periode. Die gesuchte Frequenz liegt dann zwischen diesen beiden Werten. Mit der parabolischen Interpolation wird die gesuchte Frequenz zwischen diesen Punkten gefunden, was die Genauigkeit spürbar verbessert.

Auflösung 0,5 Hz zu realisieren, müssen 8192 Werte verarbeitet werden. Die Leistungstärke des Mikrocontrollers TM4C123G wäre dem zwar voll gewachsen, doch der Umfang des benötigten Speichers würde die Grenze des 32 KB großen SRAMs sprengen. Mit vielen Abstrichen wurde schließlich die Abtastfrequenz 1 kHz realisiert, damit war das Spektrum nur noch 500 Hz breit. Eine FFT mit 2048 Werten passte gerade noch in den Speicher.

Da schnell deutlich wurde, dass diese Herangehensweise nicht zum Ziel führt, wurde auf das Schreiben weiterer Software-Komponenten verzichtet.

Zweiter Anlauf: Autokorrelationsfunktionen und Pitch-Detection

Die Literatur-Recherchen zu Methoden, die das Zerlegen beliebiger Signale in ihre spektralen Bestandteile zum Ziel haben, führten schnell zu der bekannten, mathematisch-wissenschaftlichen Abhandlung, die bei [4] nachzulesen ist. Darin wird beschrieben, wie durch eine Autokorrelationsfunktion [5] die Grundfrequenz mit hinreichender Genauigkeit ermittelt werden kann. Den mathematischen Hintergrund dieser Methode, die *Yin* genannt wird, erläutert diese wissenschaftliche Arbeit ebenfalls. Der nebenstehende Textkasten enthält einige weiterführende Informationen. Auf der Grundlage von Yin schrieb Joren Six [6] eine Bibliothek für C++, sie wurde von Ashok Fernandez in eine Bibliothek für C portiert. Dieses Beispiel zeigt, dass eine theoretische Abhandlung durchaus ohne Umweg in eine praxisnahe Anwendung einfließen kann. Die C-Bibliothek von Fernandez wurde nahezu unverändert für den Gitarren-Tuner übernommen. Die Problemlösung, mit der dies möglich war, ist ein Merkmal der exzellenten Arbeit.

Noten und Frequenzen

Für das Verständnis des Gitarren-Tuners sind Grundkenntnisse zu Noten, Tonleitern und Frequenzen unabdingbar. Das Stimmen von Musikinstrumenten beginnt in der Regel mit dem so genannten Kammerton A, Frequenz 440 Hz, die übrigen Noten einer Tonleiter heißen H (Englisch B), C, D, E, F und G. Die Intervalle zwischen diesen Noten sind entweder Halbton- oder Ganztonschritte. Zwischen H und C sowie E und F liegen Halbtonschritte, während die übrigen Noten einen ganzen Ton voneinander entfernt sind.

Die 12 aufeinander folgenden Halbtöne von C bis H bilden eine Oktave, ihre Bezeichnungen lauten C, C#, D, D#, E, F, F#, G, G#, A, A# und H, wobei ein # (Kreuz) bedeutet, dass die Note einen halben Ton höher als die gleiche Note ohne # liegt. Auf die Note H folgt die Note C der darüber liegenden Oktave. Die Oktaven werden üblicherweise von den tiefen nach den hohen Oktaven durchnummeriert. Da der Kammerton A, 440 Hz, zur vierten Oktave gehört, hat er die Bezeichnung A4. Daraus folgt, dass der Ton A der nächsten Oktave mit A5 bezeichnet wird, er hat die doppelte Frequenz 880 Hz.

Die sechs Saiten einer Gitarre sind wie folgt gestimmt: E2-A2-D3-G3-H3-E4. Offen schwingt die untere E-Saite (E2) mit der Frequenz 82,407 Hz,

während die Frequenz der oberen E-Saite (E4) offen 329,628 Hz beträgt. Wenn die hohe E-Saite am 12. Bund gegriffen wird, ertönt der Ton E5 mit der Frequenz 659,255 Hz, am 24. Bund gegriffen ertönt E6, Frequenz 1318,51 Hz. Diese Frequenz legt die obere Grenze des vom Gitarren-Tuner erfassten Spektrums fest. Die untere Grenze orientiert sich an der viersaitigen Bassgitarre, sie ist verglichen mit einer normalen Gitarre um eine Oktave niedriger gestimmt. Die Tonfolge der offenen Saiten lautet E1-A1-D2-G2, wobei E1 die Frequenz 41,203 Hz hat. Auf einer fünfsaitigen Bassgitarre befindet sich zusätzlich eine H-Saite mit dem Ton H0, Frequenz 30,868 Hz. Diese Frequenz legt die untere Grenze des Spektrums fest, das vom Gitarren-Tuner erfasst wird.

Die Frequenzen der Halbtöne wachsen nicht linear, sie steigen exponentiell an, und zwar für zwei benachbarte Halbtöne um den Faktor der zwölften Wurzel aus 2. Zum Beispiel hat der Halbton A#4, der über dem Ton A4 mit der Frequenz 440 Hz liegt, die Frequenz $440 \cdot 2^{1/12} = 440 \cdot 1,059463 = 466,164$ Hz. Dieser Gesetzmäßigkeit folgt **Tabelle 1**, sie enthält die Frequenzwerte, geschrieben als C-Code.

Anforderungen an den Gitarren-Tuner

Bevor es mit dem Projekt weitergeht, sind hier die wichtigsten Eigenschaften aufsummiert, die der Gitarren-Tuner besitzen soll:

- **Schnell:** 10 Aktualisierungen in der Sekunde.
- **Genau:** Fehlertoleranz beim Ermitteln der Grundfrequenz unter 5 %.

Der Autor

Emile van de Logt ist im niederländischen Rotterdam als Lehrbeauftragter in den Bereichen Allgemeine Elektrotechnik und Versorgungstechnik im Gesundheitswesen tätig. Nach dem Studium der Elektrotechnik an der TU Eindhoven erweiterte er sein Wissen, indem er sich dem Fach Management Sciences zuwandte. Van de Logt ist leidenschaftlicher Praktiker, er war maßgeblich an den Elektor-Workshops Embedded C und FPGA-VHDL beteiligt. Das Bauen von Röhrenverstärkern und Gitarren-Effektgeräten steht bei ihm hoch im Kurs. Eine Nebenbeschäftigung ist das Hobby des Bierbrauens. Das von ihm entwickelte, voll automatisierte Brausystem unterstreicht seine Kreativität.



- **Anzeige:** Darstellen der Note oder Halbnote (#) sowie Nummer der Oktave.
- **Differenz:** Darstellen der relativen Abweichung, Bereich -50...+50 % mit 21 LEDs (jeder Schritt 5 %).
- **Lichteffekt:** Lauflicht in Richtung mittlerer LED bei korrekter Stimmung.
- **Flat-Tuning:** Wenn alle Saiten einen oder mehrere Noten tiefer gestimmt sind, muss eine Korrektur (0...-6) möglich sein, die Anzeige soll dann normal arbeiten.
- **Bereich:** Er muss vom niedrigsten Ton einer Bassgitarre (H0 bei fünf Saiten) bis zum höchsten Ton einer normalen Gitarre (E6) reichen. Damit ist der Gitarren-Tuner auch für viele andere Musikinstrumente einsetzbar.

```
const float note_freq[MAX_NOTES] = { 20.602, 21.827, 23.125, 24.500, 25.957, 27.500, 29.135, 30.868,
  32.703, 34.648, 36.708, 38.891, 41.203, 43.654, 46.249, 48.999, 51.913, 55.000, 58.270, 61.735,
  65.406, 69.296, 73.416, 77.782, 82.407, 87.307, 92.499, 97.999, 103.826, 110.000, 116.541, 123.471,
  130.813, 138.591, 146.832, 155.563, 164.814, 174.614, 184.997, 195.998, 207.652, 220.000, 233.082, 246.942,
  261.626, 277.183, 293.665, 311.127, 329.628, 349.228, 369.994, 391.995, 415.305, 440.000, 466.164, 493.883,
  523.251, 554.365, 587.330, 622.254, 659.255, 698.456, 739.989, 783.991, 830.609, 880.000, 932.328, 987.767,
  1046.502, 1108.731, 1174.659, 1244.508, 1318.510, 1396.913 };
```

```
const char note_name[MAX_NOTES][4] = {"E0", "F0", "F#0", "G0", "G#0", "A0", "A#0", "B0",
  "C1", "C#1", "D1", "D#1", "E1", "F1", "F#1", "G1", "G#1", "A1", "A#1", "B1",
  "C2", "C#2", "D2", "D#2", "E2", "F2", "F#2", "G2", "G#2", "A2", "A#2", "B2",
  "C3", "C#3", "D3", "D#3", "E3", "F3", "F#3", "G3", "G#3", "A3", "A#3", "B3",
  "C4", "C#4", "D4", "D#4", "E4", "F4", "F#4", "G4", "G#4", "A4", "A#4", "B4",
  "C5", "C#5", "D5", "D#5", "E5", "F5", "F#5", "G5", "G#5", "A5", "A#5", "B5",
  "C6", "C#6", "D6", "D#6", "E6", "F6"};
```

Tabelle 1.

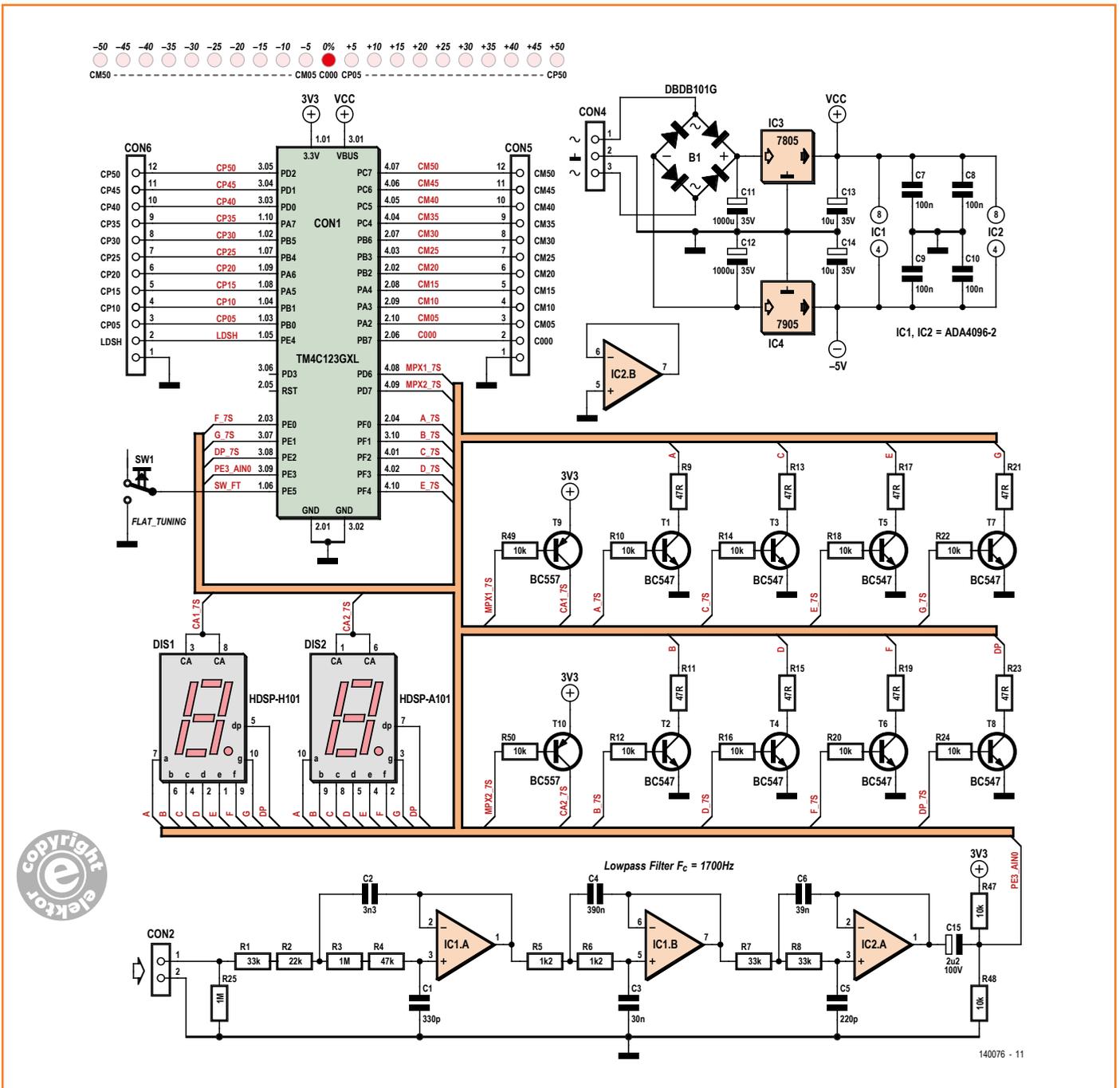
Bild 3.
Die Schaltung besteht aus relativ wenigen Komponenten, da viele Funktionen bereits auf dem Entwicklungsboard vorhanden sind.

Beschreibung des Entwurfs

Beim Gitarren-Tuner greifen die Funktionen der Hardware (**Bild 3**) und der Software eng ineinander. Deshalb folgt hier eine zusammenfassende Beschreibung, die beides einschließt.

Es wird empfohlen, die Software bereits jetzt herunterzuladen [8] und zur Hand zu haben. Hier sind die wichtigsten Details:

- Das Signal der Gitarre wird einem Tiefpass 6. Ordnung mit der Eckfrequenz 1700 Hz zugeführt. Der Tiefpass wurde mit dem *WEBENCH Filter Designer* von Texas Instruments entworfen. TI schreibt den eigenen Opamp-Typ ADA4096 vor, ein Standard-Opamp wie der LM833 leistet jedoch gleiche Dienste.
- Das Ausgangssignal des Tiefpasses wird Ein-



gang AIN0 (ADC-Eingang 0) des Mikrocontrollers zugeführt.

- Die ADC-Taktfrequenz beträgt 125 kHz. Es wird eine 16-fache Mittelwertbildung angewandt, so dass der Durchschnitt aus 16 Samples den Wert eines Sample ergibt. Die erforderliche Zeit beträgt 128 μ s. Die Einstellungen werden in der Routine `init_timer0_for_adc()` vorgenommen.
- Der Yin-Algorithmus benötigt mindestens zwei vollständige Perioden des Gitarrensignals, um die Grundfrequenz zu bestimmen. Bei der tiefsten Note (H0, 30,868 Hz) entspricht dies ungefähr 65 ms. Die Aktualisierung findet 10 Mal in der Sekunde statt, also in Abständen von 100 ms. Damit kann der Yin-Algorithmus sogar Frequenzen bis 20 Hz (Note E0) erkennen.
- Die Sample-Frequenz F_s wird auf 6000 Hz eingestellt, so dass in 100 ms 600 Samples generiert werden. Dies geschieht mit der Interrupt-Routine `ADC0_SS3_IRQHandler()`. Die Samples werden in einem Ringpuffer abgelegt. Ein Ringpuffer ist ein spezieller Puffer, in den ein Prozess Werte schreibt, während ein anderer Prozess Werte liest.
- Nach Generieren von 600 Samples wird ein Signal (Variable `sampled_new`) für die Schleife `while` in `main()` gesetzt. Die 600 Samples werden aus dem Ringpuffer in den Eingangspuffer des Yin-Algorithmus übertragen, danach wird der Yin-Algorithmus aufgerufen. Der Yin-Algorithmus berechnet die Grundfrequenz (Variable `pitch`) des Gitarrensignals.
- Die Routine `find_nearest_note()` ermittelt, welche Note (Variable `nnote`) aus der abgelegten Tabelle dem übergebenen Wert am nächsten kommt. Ferner berechnet die Routine die Abweichung in Prozent (Variable `cents`).

- Beide Variablen werden der Interrupt-Routine `Timer1_IRQHandler()` übergeben, die mit 200 Hz läuft. Die Routine setzt die Bezeichnung der Note und die Nummer der Oktave auf das Siebensegment-Display. Das Display wird von der Routine `display_note()` im Multiplex-Betrieb gesteuert. Außerdem wird die relative Abweichung von der Note mit 21 LEDs visualisiert, dies ist Aufgabe der Routine `display_cents()`. Basis ist ein Zustandsdiagramm, es ist auch für das Lauflicht zuständig, wenn die Saite korrekt gestimmt ist.

Was noch übrig bleibt, sind die Definitionen für die diversen I/O-Signale auf dem Entwicklungsboard. Nahezu alle I/O-Pins auf dem Board werden benutzt, ergänzende Hardware wird nicht benötigt. Die Pin-Belegung ist in **Tabelle 2** zusammengefasst, sie geht auch aus der Software hervor. Die Bezeichnungen in roter Farbe sind die Standard-Bezeichnungen der Pins, die anderen Bezeichnungen beschreiben die Funktionen, die sie beim Gitarren-Tuner haben:

- CP05...CP50: Anschlüsse der 10 LEDs für die Abweichung in positiver Richtung.
- CM05...CM50: Anschlüsse der 10 LEDs für die Abweichung in negativer Richtung.
- C000: Anschluss der LED in der Mitte, sie signalisiert ein korrektes Stimmen.
- Flat-Tuning: Eingang für den Taster, mit dem die Tiefe des Flat-Tunings eingestellt wird. Beim ersten Druck erscheint die aktuelle Einstellung, anschließend wird der Wert bis -6 gesenkt, um anschließend auf 0 zurückzuspringen. Ein Beispiel: Flat-Tuning ist -1, auf der Gitarre ist dann die E-Saite als D# gestimmt. Wenn die Saite angeschlagen wird, erscheint trotzdem E auf dem Display.

	+3V3	VBUS	+5V
CP30	PB5	GND	GND
CP05	PB0	PD0	CP40
CP10	PB1	PD1	CP45
LED-sharp	PE4	PD2	CP50
Flat-tuning	PE5	PD3	
CP25	PB4	PE1	G_7seg
CP15	PA5	PE2	DP_7seg
CP20	PA6	PE3	AIN0
CP35	PA7	PF0	B_7seg

C_7seg	PF2	GND	
D_7seg	PF3	PB2	CM20
CM25	PB3	PE0	F_7seg
CM35	PC4	PF0	A_7seg
CM40	PC5	RESET	
CM45	PC6	PB7	C000
CM50	PC7	PB6	CM30
mpx1_7seg	PD6	PA4	CM15
mpx2_7seg	PD7	PA3	CM10
E_7seg	PF4	PA2	CM05

Tabelle 2.

Bild 4.
Die Platine für den Gitarren-Tuner. Das Entwicklungsboard TM4C123GXL wird auf der Rückseite aufgesteckt.

Nur wenn Flat-Tuning auf 0 steht, wird D# angezeigt.

- LED-Kreuz (englisch: *sharp*): Anschluss der LED, die aufleuchtet, wenn eine halbe Note (#) erkannt wurde.
- mpx1_7seg und mpx2_7seg: Über diese Pins werden die Siebensegment-Displays im Multiplex-Betrieb gesteuert.
- Die xx_7seg-Signale liegen auf den Niveaus, die zum Steuern der Siebensegment-Displays nötig sind.

Aufbauen des Gitarren-Tuners

Für das Installieren der Software stehen zwei Varianten zur Wahl:

Erweitert: Laden Sie das *Code Composer Studio (CCS) Version 6* sowie das *Tivaware API* herunter [10] und installieren Sie beides. Mit dem schon erwähnten *Getting Started Workbook* wird dies problemlos möglich sein, in den ersten Kapiteln wird die Installation ausführlich beschrieben. Anschließend können Sie das Projekt *Gitarren-Tuner* von der Elektor-Website [8] herunterladen

Stückliste

Widerstände:

(alle SMD1206)

R1,R7,R8 = 33 k

R2 = 22 k

R3,R25 = 1 M

R4 = 47 k

R5,R6 = 1k2

R9,R11,R13,R15,R17,R19,R21,R23 = 47 Ω

R10,R12,R14,R16,R18,R20,R22,R24,R47,
R48,R49,R50 = 10 k

Kondensatoren:

(C1...C10: SMD1206)

C1 = 330 p

C2 = 3n3

C3 = 30 n (z.B. Mouser
581-12061C303JAT2A)

C4 = 390 n/16 V

C5 = 220 p/16 V

C6 = 39 n/50 V

C7,C8,C9,C10 = 100 n

C11,C12 = 1000 μ /35 V stehend (Raster 5
mm, \varnothing 12,5 mm)

C13,C14 = 10 μ /35 V stehend (Raster 2,54
mm, \varnothing 5 mm)

C15 = 2,2 μ /100 V stehend

Halbleiter:

B1 = Brückengleichrichter DB101

T1..T8 = BC547

T9,T10 = BC557

IC1,IC2 = ADA4096-2 oder LM833 (Du-
al-Opamp, SOIC-8)

IC3 = 7805

IC4 = 7905

21 LEDs rot 3 mm, Low-current (für CP05...
CP50, C000, CM05...CM50)

1 LED rot 5 mm, Low-current (für LDSH)

Außerdem:

Entwicklungsboard TM4C123GXL (Texas
Instruments)

DIS1 = 7-Segment-Display rot, gemeinsame
Anode, 0,56", 13 mcd@10 mA (z. B. Avago
HDSP-H101)

DIS2 = 7-Segment-Display rot, gemeinsame
Anode, 0,3", 10 mcd@10 mA (z. B. Avago
HDSP-A101)

CON2 = Stiftkontaktleiste 2-polig, Raster
2,54 mm

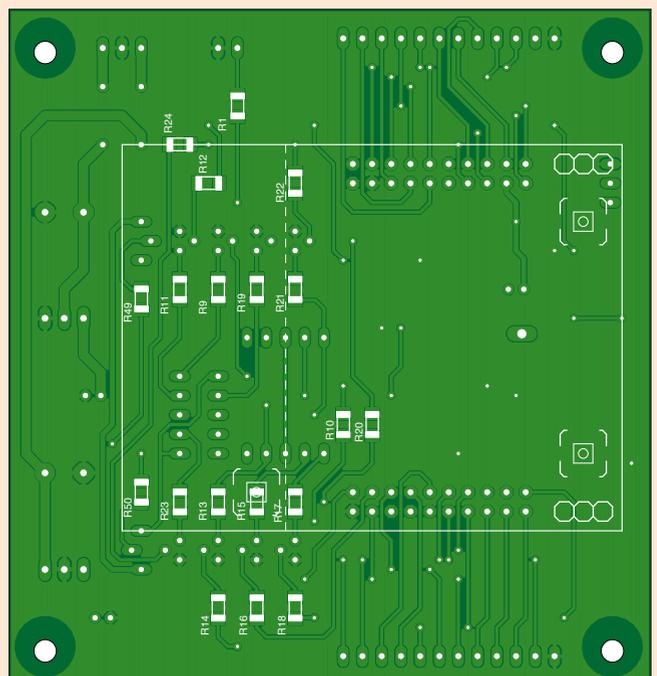
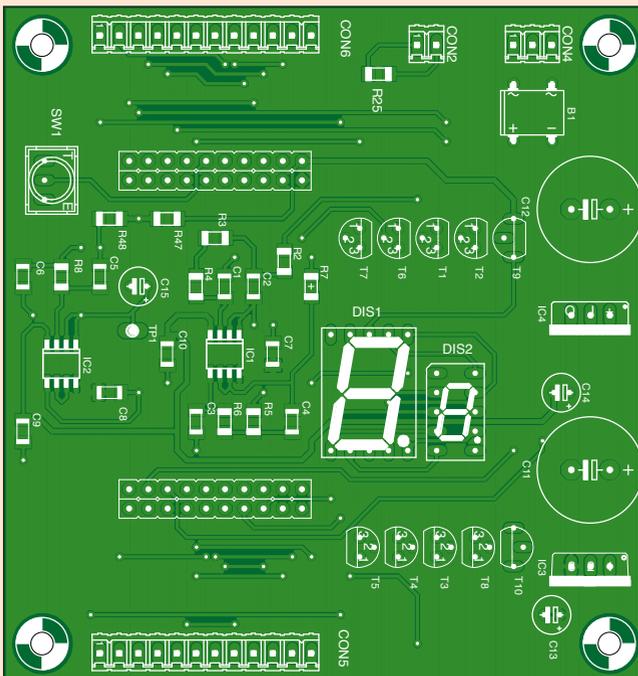
CON4 = Stiftkontaktleiste 3-polig, Raster
2,54 mm

CON5,CON6 = Kabelschraubklemme 12-po-
lig, Raster 2,54 mm

SW1 = Drucktaster mit Arbeitskontakt
2 Stück Buchsenleiste 2 · 10-polig für das
Entwicklungsboard, Raster 2,54 mm

3 Stück Klinkenbuchse 6,3 mm für
Pultmontage

Platinenlayout 140076-1, Download von [8]



und in den *Workspace* des CCS übernehmen. Mit dem CCS lässt sich die Software anpassen und eine neue .bin-Datei generieren. Verbinden Sie das Entwicklungsboard TM4C123GXL über USB mit dem PC und klicken Sie auf die Schaltfläche *Debug*. Die modifizierte Software wird in das Entwicklungsboard geladen.

Minimal: Diese Variante ist dafür gedacht, dass das Entwicklungsboard TM4C123GXL nur mit der Software des Gitarren-Tuners programmiert werden soll, während für die CCS-Entwicklungsumgebung kein Bedarf besteht. Installieren Sie dann nur das Programm *LM Flash Programmer* [11] und verwenden Sie die Datei *Guitar_Tuner_Yin.bin* aus dem Ordner *Debug* des Projekts, um diese Datei in das Entwicklungsboard zu laden. Die Prozedur ist ebenfalls in der *Workbook*-Sammlung beschrieben.

Das Entwicklungsboard TM4C123GXL kann unmittelbar bei Texas Instruments bestellt werden, auch bei Distributoren wie Farnell ist es erhältlich. Wegen der Kompatibilität mit anderen Boards dieses Herstellers sind einige Port-Pins miteinander verbunden. Dies trifft für PB7 und PD1 zu, die Verbindung stellt 0-Ω-Widerstand R10 her. Ebenso ist PB6 mit PD0 verbunden, hier ist der 0-Ω-Widerstand R9 die Brücke. Die Wider-

stände befinden sich unter dem Mikrocontroller und über dem Textlogo von TI, sie sind als R9 und R10 gekennzeichnet. Entfernen Sie diese Widerstände vorsichtig mit einem LötKolben.

Die Platine für den Gitarren-Tuner (**Bild 4**) ist als „Shield“ ausgeführt, das auf das Entwicklungs-

board TM4C123GXL aufgesteckt wird. Auf der Platine befinden sich der Tiefpass, die Stromversorgung sowie die Anschlüsse für die LEDs und die Siebensegment-Displays. Vorhanden sind einige SMD-Komponenten, sie lassen sich von Hand mit einem feinen LötKolben löten. Ein Heißluft-Lötgerät ist hier von Vorteil. Beginnen Sie mit dem Aufbau der Stromversorgung auf dem Shield. Die Spannung des Netztrafos kann 2 · 6 V oder 2 · 9 V betragen, die Leistung muss nicht hoch sein. Prüfen Sie nach dem Aufbau, ob stabile Spannungen +5 V und -5 V anliegen. Schalten Sie die Stromversorgung ab und montieren Sie die übrigen Komponenten. Die Siebensegment-Displays können auf der Platine aufliegen oder erhöht angebracht werden. Montieren Sie die 2 · 10-poligen Buchsenleisten auf der Unterseite des Shields, so dass dort das Entwicklungsboard TM4C123GXL eingesteckt werden kann.

An den Klemmen CON5 und CON6 werden die 21 LEDs angeschlossen. Der Anschluss der mittleren LED ist Kontakt 2 von CON5 (C000), rechts liegen die Kontakte für die LEDs CP5...CP50 (+5...+50 %), links die Kontakte für die LEDs CM5...CM50 (-5...-50 %). Außerdem gibt es die Kreuz-LED



stände befinden sich unter dem Mikrocontroller und über dem Textlogo von TI, sie sind als R9 und R10 gekennzeichnet. Entfernen Sie diese Widerstände vorsichtig mit einem LötKolben.

(*sharp*), die halbe Noten anzeigt, ihr Anschluss ist Kontakt 2 von CON6. Die gemeinsame Masse für alle LEDs ist Kontakt 1 der Klemmenleiste. Prüfen Sie nach dem Aufbau die Funktion des Tiefpasses, indem Sie ein sinusfö-

Bild 5. Platinenrückseite mit aufgestecktem Entwicklungsboard. Das Foto zeigt einen frühen Prototypen, an dem noch Änderungen vorgenommen wurden.



Bild 6.
Foto des Geräts, das der Autor regelmäßig auf der Bühne einsetzt.

miges Signal (500 mV_{SS}) an CON2 legen. Das Ausgangssignal des Filters erscheint an Testpunkt TP1 (LPF_OUT). Signale unter 1700 Hz werden durchgelassen, höherfrequente Signale werden unterdrückt.

Wenn dieser Test erfolgreich verlief, können das Entwicklungsboard TM4C123GXL und das Shield zusammengesteckt werden (Board unten, Shield oben, USB-Buchse in Richtung C11/C12/IC3/IC4). Nach Einschalten der Stromversorgung prüfen Sie, ob der Dezimalpunkt des kleinen Siebensegment-Displays mit der Frequenz 1 Hz blinkt. Drücken Sie den Flat-Tuning-Taster und prüfen Sie, ob das Display die Ziffern -0...-6 anzeigt. Setzen Sie Flat-Tuning auf 0, verbinden Sie die Gitarre mit dem Aufbau und schlagen Sie eine Saite an.

Auf dem Display müssen nun die Note und die Oktavnummer erscheinen, gleichzeitig leuchten bestimmte LEDs auf. Stimmen Sie die Gitarre, so dass die mittlere LED (C000) leuchtet. Das im Elektor-Labor getestete Gerät erkannte sogar die Note D5 (hohe E-Saite, 22. Bund). Mit einer Bassgitarre wurden noch keine Tests durchgeführt, doch die Signale eines Funktionsgenerators wurden bis herab auf 20 Hz problemlos erkannt. Für den Einbau verwendete der Autor ein Gehäuse von Hammond, Typenbezeichnung 515-0950, und stattete es mit einer Frontplatte aus. Die freie Gestaltung der Frontplatte ist möglich, weil die LEDs nicht auf der Platine montiert, sondern an CON5 und CON6 angeschlossen sind.

(140076)gd

Weblinks und Literatur

- [1] www.ti.com/tool/EK-TM4C123GXL
- [2] http://processors.wiki.ti.com/index.php/Getting_Started_with_the_TIVA%E2%84%A2_C-Series_TM4C123G_LaunchPad
- [3] <http://de.wikipedia.org/wiki/Nyquist-Shannon-Abtasttheorem>
- [4] YIN, A fundamental frequency estimator for speech and music, 2002, Alain de Cheveigne & Hideki Kawahara
- [5] <http://de.wikipedia.org/wiki/Autokorrelation>
- [6] <https://github.com/JorenSix/Pidato>
- [7] <https://github.com/ashokfernandez/Yin-Pitch-Tracking>
- [8] www.elektor-magazine.de/140076
- [9] www.ti.com/lscds/ti/analog/webench/webench-filters.page
- [10] www.ti.com/tool/ccstudio
- [11] www.ti.com/tool/lmflashprogrammer

25 Jahre **Beta**

LAYOUT

create : electronics

Geschenkt:

3D-Ansicht



3D-PDF & Step-Datei



3D-Kollisionsprüfung



25 Jahre **Beta**

LAYOUT



www.pcb-pool.com/brd-to-3D

Learn to program

Spielerisch programmieren lernen – ohne Tastatur

NEU!

Learn to program ist ein einfaches Lernspiel für Kinder und Schüler, das im Stil der Anfangsjahre des Computerzeitalters das Programmieren mit nur vier Tasten erlaubt. Auf dem Spielbrett befinden sich alle nötigen Ein-/Ausgabeeinheiten wie etwa eine LED-Ampel, ein Helligkeitssensor und ein Piezopiepser. Damit lassen sich verschiedene Programme entwickeln, wie zum Beispiel eine einfache Eieruhr, ein Wecker (der morgens wie ein Hahn „Kikeriki“ ruft) oder auch einfache Reaktionsspiele. Die Batterieversorgung macht die Programmierung netzunabhängig.



Für viele Elektroniker ist es selbstverständlich, Programme zu schreiben, einfache Regelungen zu implementieren oder Steuerungen zu realisieren. Nur ist einem dabei oft nicht klar, was für ein langer Weg es war, um soweit zu kommen. Mit diesem edukativen Spiel kann man das Mysterium des Programmierens Kindern anschaulich erklären.

Eigenschaften des Boards:

- 16 verschiedene Befehle • 8 Eingabetaster • Tonausgabe (Piezopiepser) • LED-Ampel • 4 weitere LEDs • Lichtsensor

Inhalt der Lernbox:

- Bestückte Mikrocontrollerplatine (Spielbrett)
- 73-seitiges A6-Handbuch

€ 44,95

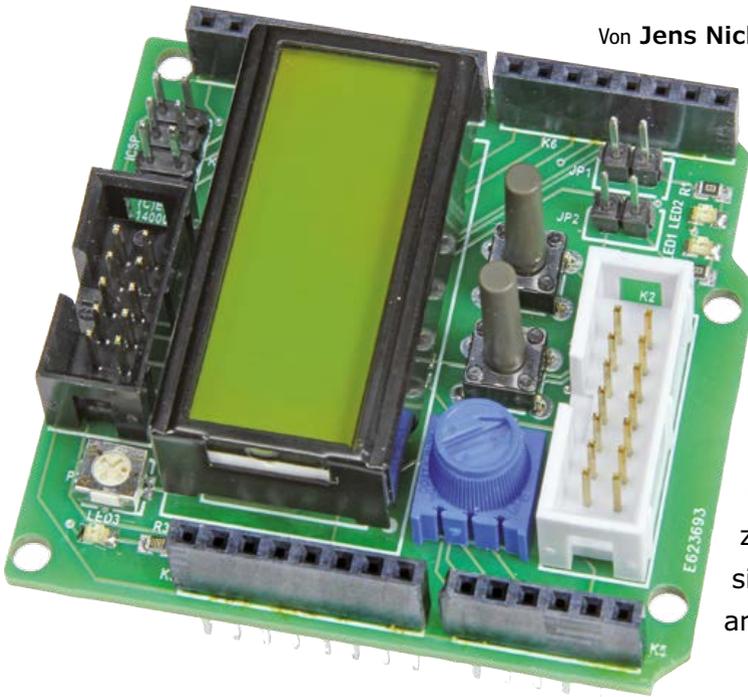
CHF 56,95

Weitere Infos & Bestellung unter www.elektor.de/learn-to-program

Mein erstes Shield :-)

LEDs, Taster, Display und mehr

Von **Jens Nickel**



Auch in der Welt der Mikrocontroller ist „Learning by Doing“ ein guter Weg. Der besonders günstige und einsteigerfreundliche Arduino Uno bringt leider so gut wie keine Peripherie mit. Wir haben daher ein kompaktes Shield entwickelt, das Anfängern mit einem Text-Display, LEDs und Tastern eine gute Basis für erste Schritte bietet. Und wer sich schon etwas auskennt, freut sich über zwei Erweiterungssteckverbinder. Hier lassen sich Relais-, Funk- und viele weitere Module anschließen.

Man mag es begrüßen oder bedauern: Besonders günstige, in Massen hergestellte Mikrocontrollerboards haben die Elektronikwelt verändert. Wer schnell zu einem funktionierenden (Demo-)Projekt kommen will oder muss, das nicht allzu viel Rechenleistung verlangt, greift zu einem „Arduino Uno“, von dem man noch mindestens einen irgendwo herumliegen hat. Besonders Einsteiger profitieren von der kostenlosen Arduino-Entwicklungsumgebung, zum Flashen des Programms ist nicht mehr als ein USB-Kabel nötig. Wer lieber Basic mag, kann von Bascom aus ebenfalls mit dem Bootloader arbeiten. Doch auch komplexe Programme in C/C++ sind möglich; mit dem kostenlosen AVR-Studio und einem kleinen Programmer wie dem AVR-ISP mk2.

Clevere Aufsteckplatinen

Ein Arduino Uno bietet freilich noch nicht allzu viel Peripherie an; lediglich eine LED haben die Entwickler dem Board spendiert. Außerdem kann man über USB auf die UART-Schnittstelle des Uno-Mikrocontrollers (Atmel ATmega328P) zugreifen. Allerdings sind die meisten Pins dieses Chips auf

zwei Reihen von Buchsenleisten geführt, und hier setzt das Konzept der „Shields“ an. Stattet man eine weitere Platine mit passend angeordneten Stiftleisten aus, so lässt sich diese fest auf das Controllerboard aufstecken. Auf dem Shield kann man dann die Peripherie unterbringen, vom Display über Sensoren bis zu Schnittstellen wie Bluetooth und WLAN. Wegen der großen Verbreitung gerade des kleinen Arduino-Uno-Boards gibt es inzwischen Hunderte von kommerziell erhältlichen Shields.

Erste Schritte

Viele unserer Leser interessieren sich für einen Einstieg in die Welt der Mikrocontroller, in Bascom und/oder C. Was läge näher, als hier einen Arduino Uno einzusetzen? Um das Programmieren eines Controllers richtig kennenzulernen, sollte man sich selbst einmal mit den wichtigsten Schnittstellen dieser Chips auseinandergesetzt haben. Und das geht natürlich am besten anhand von Peripherie, welche diese Schnittstellen nutzt. Da es aber kaum ein Shield gibt, das all die nötige Peripherie mitbringt und trotzdem klein und günstig ist, haben wir uns zur Entwicklung

eines eigenen Shields entschlossen. Es enthält ein Display, zwei User-LEDs, zwei Taster und ein Poti (**Bild 1**). So kann man nach dem Aufstecken des Shields gleich loslegen. Für größere Projekte sind zwei Steckverbinder gedacht, an die sich wiederum andere Erweiterungsboards über Flachkabel anschließen lassen. Bereits erhältlich sind zum Beispiel eine Relaisplatine und ein Funkmodul.

Das Shield ist im Elektor-Shop unter der Nummer 140009-91 zu haben [1]. Wer das kleine Board als Programmieranfänger nutzen möchte und in erster Linie Demo- und Beispiel-Anwendungen ausprobiert und abwandelt, muss ab hier eigentlich nicht weiterlesen. An anderer Stelle in diesem Heft zeigen wir erste kleine Anwendungen für unser Shield, die in Bascom geschrieben sind. Diese eignen sich auch prima zum Test. Für alle C-Freunde – und solche die es werden wollen – hier ein Versprechen: C-Software wird folgen, denn wir werden das Shield ab jetzt noch öfter in Elektor-Projekten einsetzen.

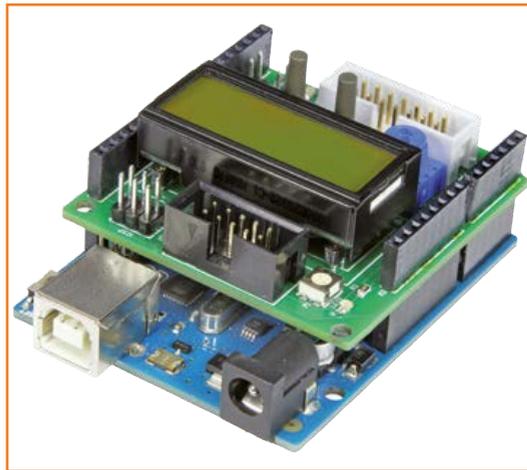


Bild 1.
Das Shield ist hier auf einem Arduino Uno zu sehen, kann aber auch auf anderen 5-V-Arduino-Boards zum Einsatz kommen.

den Kasten „Spannungen“). An den 3,3-V-Pin haben wir auf dem Shield eine Power-LED angeschlossen; bei aufgestecktem Shield zeigt diese also an, ob der Arduino versorgt ist. Außerdem stehen auf dem Shield zwei User-LEDs bereit, die über die Arduino-Pins IO10 und AD2

Buchsenleisten

Die Buchsenleisten des Arduino Uno wurden auf dem Shield „gedoppelt“. Für die dort zugänglichen Signale gibt es einen Arduino-Standard, der - unabhängig vom Mikrocontroller auf dem jeweiligen Arduino-Board - eingehalten werden muss. Die einzelnen Buchsen werden manchmal auch als Arduino-Pins bezeichnet. In der **Tabelle** führen wir die Arduino-Pins und die entsprechenden Controller-Pins des ATmega328 auf.

Wegen des Displays und der Taster ist es zwar nicht möglich, auf das Elektor-Shield ein weiteres Shield „huckepack“ zu setzen. Doch bleiben alle Arduino-Pins auch bei aufgestecktem Elektor-Shield zugänglich; in den meisten Fällen wird die Funktion auch nicht durch das aufgesteckte Shield beeinflusst, wie man im Schaltplan in **Bild 2** erkennen kann. Dies gilt zum Beispiel für die analogen Eingänge AD0 und AD1, für die serielle Schnittstelle RX/TX und die digitalen Ein-/Ausgänge IO8 und IO9. Bei gezogenen Jumpers JP1 und JP2 und bei abgestecktem Display werden weitere Ein-/Ausgänge frei.

LEDs, Taster, Poti

Auf Arduino-Boards werden grundsätzlich zwei Spannungen von 5 V und 3,3 V generiert; diese stehen an einer der Buchsenleisten bereit und können von Shields genutzt werden (siehe auch

Tabelle 1.
Arduino-Pins, Controller-Pins und Funktionen auf dem Shield.

Arduino-Pin	ATmega328	Funktion
SCL (*)	PC5	EEC-SCL
SDA (*)	PC4	EEC-SDA
IO13/SCK	PB5	ISP-SCK
IO12/MISO	PB4	ISP-MISO
IO11/PWM/MOSI	PB3	ISP-MOSI
IO10/PWM	PB2	LED2
IO9/PWM	PB1	ECC-GPIOB
IO8	PB0	ECC-GPIOA
IO7	PD7	LCD-D7
IO6	PD6	LCD-D6
IO5	PD5	LCD-D5
IO4	PD4	LCD-D4
IO3	PD3	LCD-E
IO2	PD2	LCD-RS
IO1/TX	PD1	ECC-TX
IO0/RX	PD0	ECC-RX
AD0	PC0	S1
AD1	PC1	S2
AD2	PC2	LED1
AD3	PC3	P1
AD4	PC4	EEC-SDA
AD5	PC5	EEC-SCL

(*) beim Arduino Uno mit AD4 und AD5 verbunden

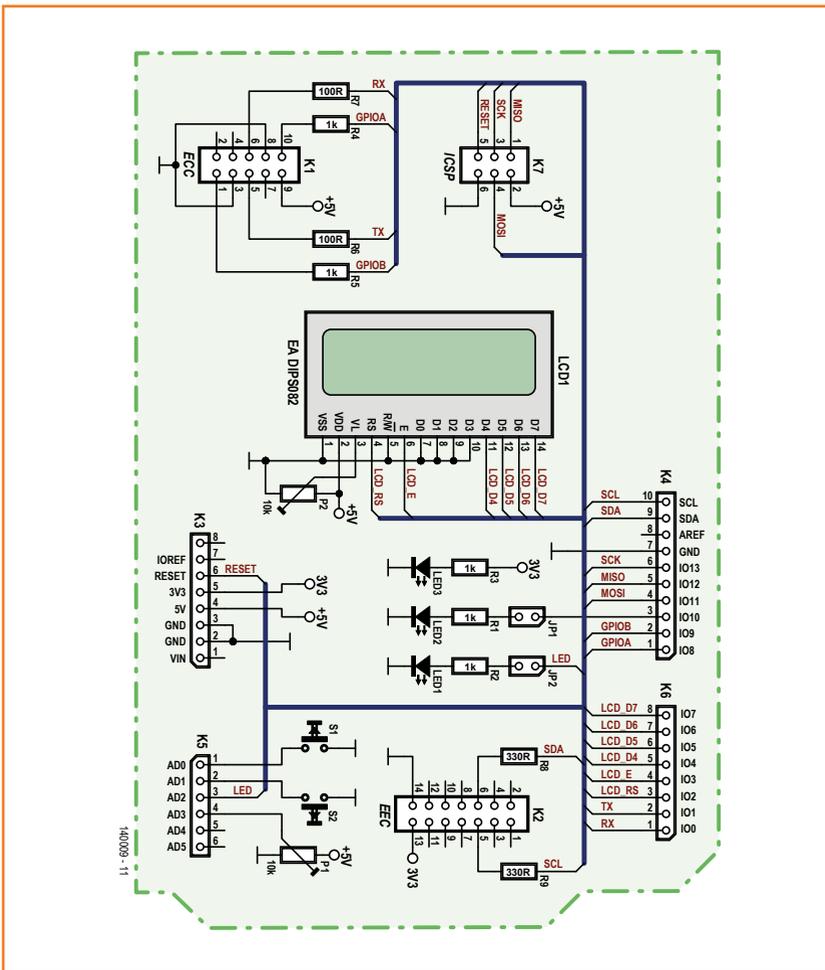
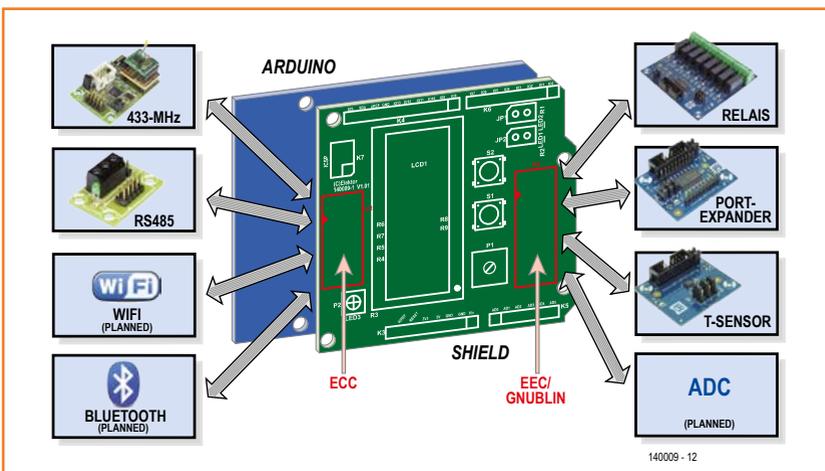


Bild 2. Schaltplan des Shields. Die Arduino-Signale sind auch auf dem Shield an Buchsenleisten zugänglich.

Bild 3. An die beiden Erweiterungssteckverbinder lassen sich die unterschiedlichsten Module anschließen.



angesteuert werden. AD2 ist eigentlich als analoger Eingang vorgesehen; beim Arduino-Uno-Board ist dieser Pin aber mit dem Controller-Portpin PC2 verbunden, der leicht per Software als digitaler Ausgang konfiguriert werden kann. Darüber hinaus lässt sich die LED1 an AD2 als Fotosensor verwenden (siehe Mikrocontroller-Kursfolge in diesem Heft). Der Pin IO10 muss nach dem Arduino-Standard ein PWM-Signal ausgeben können, dies gibt uns die Möglichkeit, das Tastverhältnis über die Helligkeit der LED2 zu visualisieren.

Für Benutzereingaben stehen zwei Taster und ein Poti zur Verfügung, die an die Eingänge AD0, AD1 und AD3 angeschlossen sind. Damit AD0 und AD1 weiterhin auch als analoge Eingänge verwendet werden können, haben wir sowohl auf Pullup-Widerstände als auch auf eine Hardware-Entprellung verzichtet. Beim ATmega328 lassen sich aber ganz leicht interne Pullup-Widerstände zuschalten, sowohl in C als auch in Bascom.

Das Poti ist mit dem Analogeingang AD3 verbunden. Wenn man – wie im Normalfall – den ADC des ATmega328 auf eine Referenzspannung von 5 V konfiguriert, ergibt sich ein Einstellbereich von 0..1023 (10 bit).

Display

Das alphanumerische Display von Electronic Assembly [2] ist abnehmbar; es stellt zwei Reihen mit 8 Zeichen dar, was für einfache Anzeigen ausreicht. Da sich die zwei Taster direkt daneben befinden, ist sogar eine einfache Menü-Steuerung realisierbar. Wir haben uns für ein Display mit 4-bit-Ansteuerung entschieden, womit neben den Signalen E und RS weitere vier Leitungen zum Display geführt werden müssen. Das ist ein kleiner Nachteil gegenüber einer Ansteuerung per SPI, dafür ist die Software-Unterstützung besser. Bascom stellt dann zum Beispiel Spezialbefehle zum Beschreiben der Anzeige mit Text zur Verfügung. Der Controller des Displays ist selbstverständlich HD44780-kompatibel, wodurch sich verbreitete C-Bibliotheken nutzen lassen. Der Adress-Offset für die zweite Zeile beträgt 40_{hex} [3].

UART-Steckverbinder

K1 ist ein voll beschalteter *Embedded Communication Connector* (ECC), den wir ausführlich in einem früheren Artikel vorgestellt haben [4]. Nach der Spezifikation sind hier die UART-Signale TX/RX sowie zwei digitale Ein-/Ausgänge zugänglich, mit High-Pegeln von 5 V. Außerdem liegen an einem Pin ständig 5 V an, um angeschlossene

Stückliste

Widerstände:

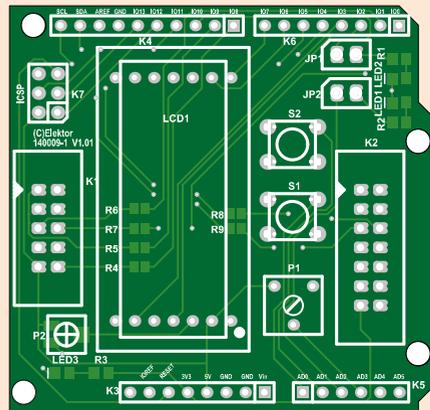
R1,R2,R3,R4,R5 = 1 k
 R6,R7 = 100 Ω
 R8,R9 = 330 Ω
 P1 = Trimmer 10 k mit Einstellknopf
 P2 = Trimmer 10 k (SMD Vishay TS53YJ103MR10)

Halbleiter:

LED1,LED2 = Low-Current-LED rot (0805)
 LED3 = Low-Current-LED grün (0805)

Außerdem:

S1,S2 = Taster
 K1 = 2x5-Wannenstecker (RM 2,54 mm)
 K2 = 2x7-Wannenstecker (RM 2,54 mm)
 K3,K4,K5,K6 = Arduino-Shield Steckverbinderleiste (Adafruit ID 85)
 K7 = 2x3-Stiftleiste (RM 2,54 mm)
 JP1,JP2 = 1x2-Stiftleiste (RM 2,54 mm) mit Jumper
 LCD1 = LCD 2 x 8 Zeichen, mit Hintergrundbeleuchtung (Electronic Assembly DIPS082-HNLED)



2x 7x1-Präzisions-Buchsenleiste für LCD1 (TE connectivity 1814655-7)
 Platine 140009-1 [1]
 oder
 Bestücktes Board 140009-91 [1]

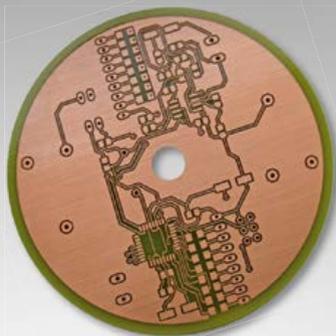
Peripherie zu versorgen. Bei dieser Peripherie handelt es sich um kleine Module, welche das eigene Projekt um verschiedene Schnittstellen mit der Außenwelt erweitern (siehe **Bild 3**). Bisher sind im Elektor-Labor ein RS485-Modul [4] und ein Funkmodul entstanden. Mit letzterem kann unser Arduino über 433 MHz drahtlos senden und empfangen [5]. Alle Signalpins des ECC (TX/RX/GPIOA/GPIOB) sind über Widerstände mit den entsprechenden

Arduino-Pins verbunden, was auf einen Vorschlag von Burkhard Kainka zurückgeht. Man erreicht so eine gewisse Experimentierfreundlichkeit: Der Uno-Controller geht nicht gleich kaputt, wenn einmal etwas mehr als 5 V an den Pins liegen.

EEC/Gnublin

An K2 finden Erweiterungsmodule nach dem Gnublin/EEC-Standard Anschluss (EEC = *Embedded Extension Connector*). Der Steckverbinder ist

Anzeige



(M)ein kleiner Freund im Elektroniklabor

- Kompakt
- Einfach zu bedienen
- 33 000 U/min Spindel
- Ein- und doppelseitige Leiterplatten

LPKF ProtoMat E33 – klein, präzise, wirtschaftlich

Kaum größer als ein DIN A3-Blatt: LPKF Qualität zum Einstiegspreis zum Fräsen, Bohren und Trennen von Leiterplatten und Gravieren von Frontplatten. Erfahren Sie mehr: www.lpkf.de/prototyping

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



LPKF
 Laser & Electronics

Spannungen

Da es inzwischen unzählige Shields gibt, hat sich die Anordnung und die Pinbelegung der Arduino-Uno-Buchsenleisten zu einer Art Quasi-Standard für Erweiterungs-Steckverbinder gemausert. Immer mehr Mikrocontrollerboards – auch mit 32-bit-Controllern – sind damit ausgestattet. In unserem ersten „Business Special Mikrocontroller“, das alle deutschen Abo-Mitglieder kostenlos erhalten haben, ist eine kleine Übersicht zu finden [7].

Allerdings benötigen einige dieser Controller eine Betriebsspannung von 3,3 V (und nicht 5 V wie beim Arduino Uno), auch die Ein- und Ausgänge sind

dann 3,3-V-kompatibel. Im Idealfall kann ein Shield sowohl auf 5-V- als auch auf 3,3-V-Controllerboards aufgesteckt werden und stellt sich automatisch auf die Betriebsspannung des Controllers ein. Daher sind an den Arduino-Buchsenleisten nicht nur grundsätzlich 5 V und 3,3 V zu finden, sondern auch ein IOREF-Pin, an dem die Betriebsspannung des Controllers anliegt.

Ursprünglich geplant hatten wir ein Shield, das sich für alle Arten von Controllerboards eignet, und an den zwei Erweiterungssteckverbindern ECC und EEC trotzdem definiert mit 5-V- beziehungsweise 3,3-V-Peripherie

kompatibel ist. Dann muss man aber zwingend mehrere Pegelwandler-ICs einsetzen, die das Shield technisch und preislich aufwendig machen. Den Plan haben wir noch nicht verworfen, im Labor existieren schon mehrere Prototypen. Speziell für den Mikrocontroller-Anfängerkurs von Burkhard Kainka und kleinere Projekte haben wir uns aber entschlossen, ein weniger aufwendiges Shield zu konzipieren, das ohne Pegelwandler auskommt. An den teilweise beschalteten EEC lässt sich trotzdem 3,3-V-Peripherie anschließen; das Shield kann mit Einschränkungen sogar auf einem 3,3-V-Controllerboard zum Einsatz kommen.

nicht voll beschaltet, weil der Standard 3,3-V-Pegel voraussetzt und wir aus Kostengründen auf Pegelwandler verzichtet haben (siehe auch Kasten „Spannungen“). Die meisten der Gnublin-Module, die von der Firma Embedded Projects entwickelt wurden und auch bei Elektor erhältlich sind, werden aber nur über die zwei I2C-Leitungen SDA und SCL angesteuert. Es ist eine vorteilhafte Eigenart des I2C-Busses, dass problemlos ein 5-V-Master (wie unser ATmega328) mit einem 3,3-V-Slave (auf dem Gnublin-Modul) verbunden werden kann. Die beiden Leitungen werden von Pullup-Widerständen (die sich in diesem Fall auf den Erweiterungsmodulen befinden) auf High gehalten und dann vom Master beziehungsweise Slave aktiv auf Low gezogen. R8 und R9 dämpfen Reflexionen bei längeren Leitungen und bieten einen gewissen Schutz gegen Störimpulse. Im Elektor-Shop erhältlich sind zum Beispiel ein

Board mit acht Relais, eine Port-Erweiterung mit 16 Ein-/Ausgängen, ein Temperatursensor und mehr [6].

Programmierung

Zu Experimentier- und Lehrzwecken sind nun alle wesentlichen Funktionen unseres ATmegas (oder eines anderen 5-V-Controllers auf einem Arduino-Board) abgedeckt: Digitale Aus- und Eingänge, analoge Eingänge, parallele Schnittstelle (4 bit), serielle UART-Schnittstelle und I2C. Fehlt eigentlich nur noch SPI. Die entsprechenden Pins haben wir auf K7 geführt, eine 2x3-Stiftleiste. Über diesen Anschluss kann der ATmega328P auf dem Arduino Uno auch programmiert werden, mit Hilfe eines einfachen Programmers wie zum Beispiel des AVR-ISP mk2. Damit wird der Arduino-Bootloader überschrieben, den man mit der Arduino-IDE aber wieder aufspielen kann.

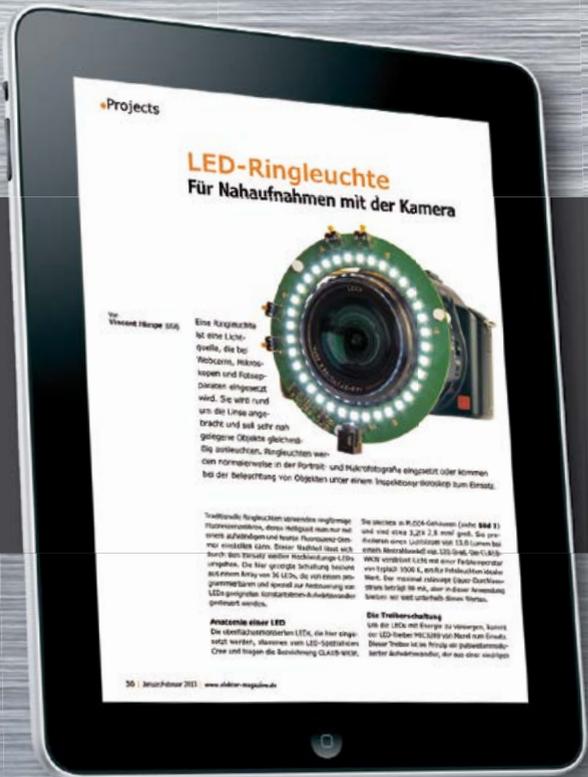
(140009)

Weblinks

- [1] www.elektor-magazine.de/140009
- [2] www.lcd-module.de/pdf/doma/dips082.pdf
- [3] www.datasheetcatalog.com/datasheets_pdf/S/T/7/0/ST7066.shtml
- [4] Elektor März 2014, www.elektor-magazine.de/130155
- [5] Elektor April 2014, www.elektor-magazine.de/130023
- [6] www.elektor.de/devtools/gnublin-linux
- [7] www.elektor-magazine.com/de/extra-downloads/specials.html



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



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

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

UMWELTSCHONEND – GÜNSTIG – GREEN

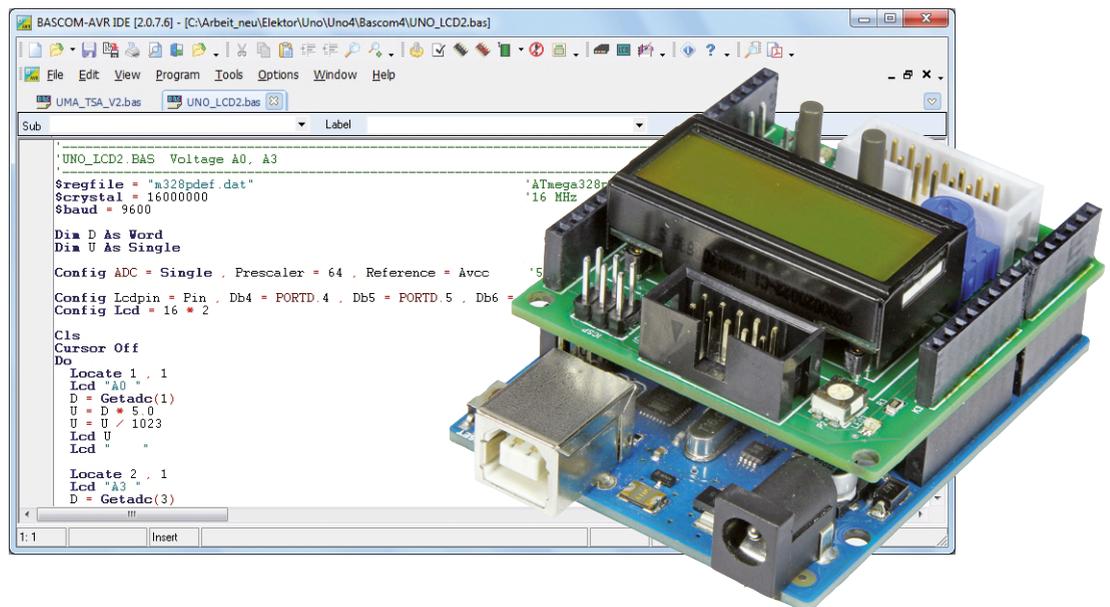
Möchten Sie Elektor lieber im elektronischen Format beziehen? Dann ist die neue GREEN-Mitgliedschaft ideal für Sie! Die GREEN-Mitgliedschaft bietet (abgesehen von den 10 Printausgaben) alle Leistungen und Vorteile der GOLD-Mitgliedschaft.



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

Mikrocontroller für Einsteiger (4)

Benutzer-Schnittstellen



Von **Burkhard Kainka**

Ernsthafte Mikrocontroller-Anwendungen besitzen meist geeignete Schnittstellen zwischen Mensch und Maschine. Ein kleines Display, ein paar Taster, ein Einstell-Poti, ein paar LEDs - schon kann man richtig schöne Geräte entwickeln. Gleich loslegen können wir mit dem von Elektor entwickelten Arduino-Shield, das ebenfalls in dieser Ausgabe beschrieben wird.

Also erstmal ein Display ansteuern! Da gibt es ganz verschiedene Typen, grafische Displays und Textzeilen-Displays, die letzteren von ganz klein (1 x 8 Zeichen) bis ganz groß (4 x 20 Zeichen). Klein ist fein, und oft braucht man nicht viele Zeichen darzustellen, deshalb wird auf dem neuen Shield ein kleines zweizeiliges Textdisplay mit jeweils acht Zeichen pro Zeile verwendet. Sowas wie „U=3,3V“ und „I=8,2mA“ passt jedenfalls schon mal rein, was will der Elektroniker mehr!

Anschluss des LCD

Wie das Display auf dem Shield an den ATmega328 auf dem Arduino angeschlossen ist zeigt **Bild 1**. Zugleich sieht man hier auch die anderen Peripherieelemente, zwei LEDs, zwei Tas-

ter und ein Poti. Wer möchte kann all dies auf einer Steckplatine nachbauen. Dabei kommt es nicht einmal darauf an, genau das gleiche Display zu nehmen. Ein Typ mit 2 x 16 Zeichen tut es genauso. Wer es einfach haben will, bestellt das neue Shield über die Website zu diesem Artikel [1], wo man wie immer auch einen Downloadordner mit den hier gezeigten Programmen findet. Das Display belegt sechs Leitungen am Port D, genauer D2 bis D7. Ganz praktisch, denn es bleiben genau noch zwei für die serielle Schnittstelle übrig, D0 (RXD) und D1 (TXD). Die Ports B und C bleiben frei für andere schöne Sachen. Alle Standard-Displays dieser Art können wahlweise im 8-bit-Betrieb verwendet werden, das heißt die Daten zum Display werden auf acht

parallelen Leitungen übertragen. Um Portpins am Controller zu sparen, hat sich der 4-bit-Betrieb allgemein durchgesetzt. Die Datenbytes werden dabei in zwei Schritten zum Display übertragen. Will man z.B. ein großes A schreiben, dann wird entsprechend der ASCII-Tabelle ein Byte &H41 (= &B01000001, dezimal 65) geschickt. Tatsächlich wird erst &B0001 geschickt und dann &B0100. Aber solche Feinheiten und die ganze Initialisierung des Displays muss man gar nicht so genau kennen, denn Bascom nimmt einem die wesentliche Arbeit ab. Schreiben Sie einfach Lcd „A“ in Ihr Programm, dann wird das A im Display erscheinen.

Das erste Programmbeispiel (**Listing 1**) zeigt, wie es geht. Mit dem Config-Befehl muss man dem Compiler erstmal sagen, welche Leitung des Displays an welchem Portpin liegt. Mit Config Lcd = 16 * 2 wird das Display dann initialisiert. Da gibt es verschiedene Formate zur Auswahl. Die Größe 8 * 2 wird nicht unterstützt, denn auch in diesem kleinen LCD ist ein Display-Controller für 16 x 2 Zeichen verbaut. In der Praxis ist das

kein Problem, denn wenn man versehentlich mal bis zu 16 Zeichen in eine Zeile schreibt, läuft das Programm fehlerlos weiter. Der Programmierer sieht die abgeschnittene Zeile aber auf dem Display und wird dann überlegen, wie noch ein paar Zeichen eingespart werden können.

Nach der Initialisierung sollte man erstmal alles löschen mit Cls (Clear Screen). Was immer dann geschrieben wird, landet in der oberen Zeile und beginnt am linken Rand. Hier ist es das Wort „Elektor“, kurz und knapp, sieben Zeichen. Eine weitere Lcd-Ausgabe würde jetzt an der Position 8 in der gleichen Zeile weiterschreiben. Es soll aber jetzt mal die zweite Zeile verwendet werden. Und dafür gibt es den Locate-Befehl. Mit Locate 2 , 1 wird das folgende an die linke Position in der zweiten Zeile geschrieben. Und hier geben wir im Programm eine aufsteigende Zahl aus. Der Zähler N erhöht sich einmal pro Sekunde, sodass das Ganze als Zeitmesser durchgehen kann. (Seit dem Start des Programms habe ich für diese Zeilen 1200 Sekunden gebraucht, 20 Minuten, ganz schön langsam heute.)

Listing 1: Text- und Zahlenausgaben.

```

'-----
'UNO_LCD1.BAS  Text Output
'-----

$regfile = "m328pdef.dat" 'ATmega328p
$crystal = 16000000      '16 MHz
$baud = 9600

Dim N As Word

Config Lcdpin = Pin , Db4 = Portd.4 ,
Db5 = Portd.5 , Db6 = Portd.6 , Db7 =
Portd.7 , E = Portd.3 , Rs = Portd.2
Config Lcd = 16 * 2

Cls
Lcd "Elektor"
'Cursor Off
Do
  Locate 2 , 1
  Lcd N
  N = N + 1
  Waitms 1000
Loop
    
```

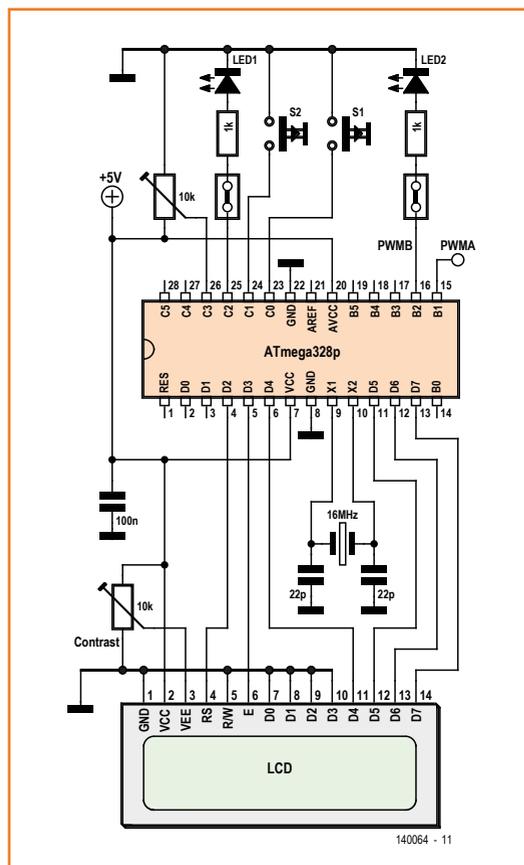


Bild 1. Prinzipschaltbild: So sind die LEDs, die Taster und das Display auf dem Elektor-Shield an den ATmega328 des „Arduino Uno“ angeschlossen.

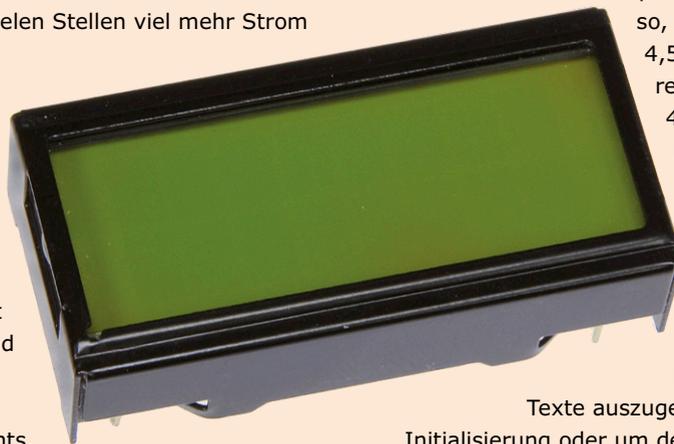
In diesem ersten Beispielprogramm gibt es zwei Lcd-Befehle, und man sollte sich klar machen, dass sie völlig unterschiedliche Dinge tun. Die reine Textausgabe kopiert nur die Zeichen, die der Programmierer in Anführungszeichen geliefert hat. Der Befehl Lcd N dagegen wandelt einen Zahlenwert in der Variablen N in einen Text um und überträgt diesen an das Display. Hier ist es eine Ganzzahl, in anderen Fällen kann es eine Realzahl mit Nachkommastellen sein. Bascom entscheidet erstmal allein, was zu tun ist; das ist sehr bequem für den Anwender. In anderen Sprachen hat der Programmierer an dieser Stelle mehr zu tun.

Hinter der Zahlenausgabe sehen Sie einen Cursor (dargestellt durch einen Unterstrich) der anzeigt, wohin das nächste Zeichen geschrieben werden würde, wenn eines käme. Das mag ja ganz sinnvoll sein, wenn man gerade etwas eintippt. Aber in unserem Fall ist der Cursor eher störend. Aber dafür gibt es ja den Befehl Cursor Off, der im Listing noch auskommentiert ist. Entfernen Sie das Kommentarzeichen, kompilieren Sie das Programm neu und laden sie es in den Controller, schon ist der Cursor weg. An dieser Stelle sei nochmal daran erinnert, wie leicht man sich weitere Informationen in der Bascom-Hilfe anzeigen lassen kann. Man bewegt einfach den Cursor (diesmal den am

Liquid Crystal Display

Flüssigkristallanzeigen (Liquid Crystal Displays, LCDs) sind sehr beliebt, weil sie wenig Strom brauchen und trotzdem gut abgelesen werden können. Man kann sie mit oder ohne Hintergrundbeleuchtung betreiben. Auf dem Elektor-Shield ist die Beleuchtung übrigens immer an. Trotzdem würde eine LED-Anzeige mit so vielen Stellen viel mehr Strom brauchen.

Ein LC-Display ist deshalb so sparsam, weil kein Licht erzeugt werden muss, sondern vorhandenes Licht gesteuert durchgelassen wird oder nicht. Zwischen zwei Glasscheiben befindet sich der flüssige Kristall und dreht abhängig von einer angelegten Spannung die Polarisationssebene des Lichts. Auf dem Glas befindet sich grundsätzlich eine Polarisationsfolie. Wenn man ein kaputtes LCD besitzt, sollte man diese Folie einmal abziehen, denn sie erlaubt interessante optische Experimente [2][3]. Die eigentliche Ansteuerung des LC-Displays benötigt Rechteckspannungen ohne Gleichspannungsanteil zwischen den einzelnen Segmenten. Dafür gibt es spezielle CMOS-Schaltungen, aber auch einige Mikrocontroller können das. In einem üblichen Text-LCD befindet sich aber ein spezieller Controller, der die eigentliche Arbeit erledigt. Der Urgroßvater aller LCD-Controller war der HD44780. Und auch heute noch sind die meisten Controller HD44780-kompatibel. Damit sind einige Steuerkommandos festgelegt und auch die Anschlüsse eines Displays. Im Normalfall hat man 14 Leitungen, manchmal kommen noch zwei für die Hintergrundbeleuchtung hinzu. Das Display



auf dem Elektor-Shield braucht nur 14 Leitungen; die Hintergrundbeleuchtung ist fest verdrahtet.

Zu den Leitungen im Einzelnen: GND und Vcc sind klar = Masse und +5V. Vee liegt an einem Kontrastpoti. Die optimale Einstellung ist bei vielen LCDs so, dass zwischen Vee und Vcc etwa 4,5 V liegen sollten. In vielen Fällen reicht auch ein Festwiderstand von 470 Ω bis 1 k Ω zwischen Vee und Gnd, dann kann man das Poti sparen.

RS ist ein Eingang, über den das Display in zwei Modi versetzt werden kann, nämlich zum Empfang von Daten- (RS=1) oder Steuerkommandos (RS=0). Die Daten braucht man, um

Texte auszugeben, die Steuerkommandos zur Initialisierung oder um den Cursor zu positionieren. R/W schaltet zwischen Schreiben und Lesen um. Lesen muss man eigentlich nur um festzustellen, ob das Display schon bereit für die nächsten Daten ist. Aber eine kleine Wartezeit tut es auch, deshalb verzichtet man oft auf die Leserichtung und legt R/W fest an GND. E ist der Enable-Eingang. Er wird gebraucht, um dem Display-Controller anzuzeigen, dass nun gültige Daten am Datenbus (den Leitungen D0 bis D7) liegen, die dann übernommen werden sollen. Nach einer High-Low-Flanke übernimmt der Controller jeweils die aktuellen Daten. Im 8-bit-Modus legt man alle acht Datenbits an und erzeugt dann eine negative Flanke an E. Im 4-bit-Modus passiert das Ganze zwei Mal; hierfür werden nur die oberen vier Leitungen (D4 bis D7) benötigt. Erst kommen die unteren (niederwertigen) Bits, dann die oberen. Ob Daten im 4- oder 8-bit-Modus kommen, erfährt das Display bei der Initialisierung.

PC-Bildschirm) auf das entsprechende Schlüsselwort und drückt F1. Und schon erfährt man mehr. In diesem Fall zum Beispiel, dass man auch einen blinkenden Cursor erzeugen kann.

Ein Zweikanal-Voltmeter

Ein kleines Doppel-Voltmeter gefällig? **Listing 2** zeigt eine von vielen möglichen Lösungen. Wie man die vom AD-Wandler gelieferten Daten in eine Spannung umrechnet wurde ja schon in der letzten Folge gezeigt. Hier geht es nun um die Ausgabe auf dem Display. Links in jeder Zeile wird erstmal ein Text geschrieben, der sagt, um welchen Eingang es sich handelt. Mit dabei ist ein Leerzeichen, das den Messwert etwas absetzt. Und nach der Ausgabe des (Realzahl-)Messwerts werden noch ein paar Leerzeichen nachgeschoben. Ohne sie könnte folgendes passieren: Sie haben zuletzt 4,859 V gemessen und gehen jetzt auf 5,0 V. Bascom gibt auch tatsächlich 5,0 aus, aber dahinter stehen noch Reste der letzten Ausgabe. Sie sehen also 5,059 V und wundern sich sehr. Damit das nicht passiert wird hier grundsätzlich alles gelöscht, was noch hinter der Ausgabe stehen könnte. Dass dabei die nicht vorhandenen Zeichen einer größeren Anzeige in Anspruch genommen werden, wird billigend in Kauf genommen.

Gemessen wird hier übrigens an ADC0 und ADC3. An ADC0 kann man ein Messobjekt anschließen, wie immer am besten über einen Schutzwiderstand (**Bild 2**). Und an ADC3 ist bereits etwas angeschlossen: Das Poti auf dem Elektor-Shield. Da kann man eine Spannung zwischen Null und 5 V einstellen.

Alle Arduino-Pins sind auch auf dem Shield auf Buchsenleisten herausgeführt. Also legen Sie mal einen dicken Elko bis maximal 470 µF parallel zum Poti an GND und ADC3 (irgendein Elko liegt sicher sowieso schon auf Ihrem Tisch), dann haben Sie ein Tiefpassfilter und können im Display beobachten wie die Spannung hinterherkriecht, wenn Sie am Poti etwas verändern. Warum nicht mehr als 470 µF? Sie könnten ja den Elko in aller Ruhe auf 5 V aufladen und dann das Poti schnell in Richtung Null stellen. Ein entsprechend großer Entladestrom würde dann das Poti quälen.

Helligkeit messen

Jetzt ändern Sie das Programm einmal so ab, dass statt ADC0 der Kanal ADC2 angezeigt wird (**Listing 3**). Hier ist auf dem Shield die Leuchtdiode LED1 über einen Jumper und einen Vorwi-

derstand von 1 kΩ angeschlossen, denn der Portpin kann ja wahlweise auch als Ausgang genutzt werden. Diesmal aber bleibt alles schön hochohmig. Und da zeigt sich, dass eine ganz normale LED zugleich auch eine Fotodiode ist, die in diesem Fall sogar als Fotoelement arbeitet. Mit einer hellen Taschenlampe bestrahlt liefert die LED eine Spannung von bis zu 1,5 V, dreimal so

Listing 2: Anzeige von Spannungen.

```

'-----
'UNO_LCD2.BAS  Voltage A0, A3
'-----
$regfile = "m328pdef.dat" 'ATmega328p
$crystal = 16000000        '16 MHz
$baud = 9600

Dim D As Word
Dim U As Single

Config Adc = Single , Prescaler = 64
, Reference = Avcc      '5 V

Config Lcdpin = Pin , Db4 = Portd.4 ,
Db5 = Portd.5 , Db6 = Portd.6 , Db7 =
Portd.7 , E = Portd.3 , Rs = Portd.2
Config Lcd = 16 * 2

Cls
Cursor Off
Do
  Locate 1 , 1
  Lcd "A0 "
  D = Getadc(1)           'A0
  U = D * 5.0
  U = U / 1023
  Lcd U
  Lcd "  "

  Locate 2 , 1           'Pot
  Lcd "A3 "
  D = Getadc(3)
  U = D * 5.0
  U = U / 1023
  Lcd U
  Lcd "  "
  Waitms 1000
Loop

```


werden, also Rechtecksignale, deren Puls/Pausen-Verhältnis man einstellen kann. So lassen sich z.B. Helligkeiten steuern; ganz ähnlich wie bei einem Dimmer am Lichtnetz, der ja auch nichts anderes macht, als ganz schnell ein- und auszuschalten. Und genau das soll jetzt auch hier geschehen, der PWM-Ausgang soll die Helligkeit einer LED steuern.

Die PWM-Ausgabe kann eine Einstellbreite von maximal 10 bit haben, aber man kann auch 8 bit einstellen. 10 bit bietet sich an, weil es gerade zur Auflösung des AD-Wandlers passt. Der Zähler zählt dabei jeweils von 0 bis 1023. Man kann nun programmieren, dass er zum Beispiel bei einem Zählerwert von 511 die Ausgangsspannung abschaltet (dann ergibt sich ein PWM-Signal mit etwa 50 % Tastverhältnis). Hierfür wird der Zählerwert regelmäßig in einer Compare-Einheit

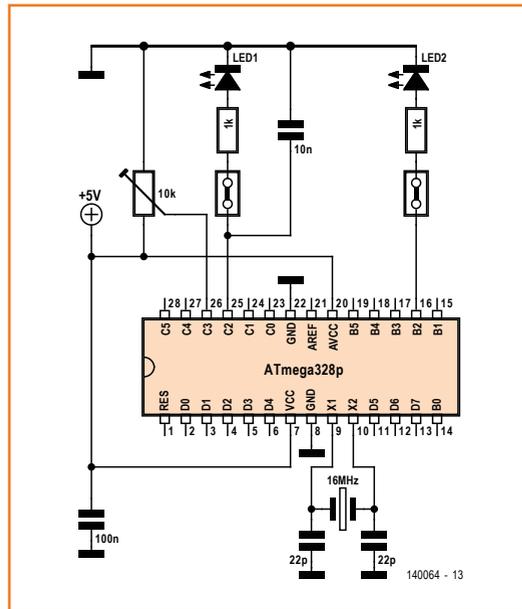


Bild 3.
LED als Lichtsensor.

Listing 4: PWM-Steuerung.

```

'-----
'UNO_LCD3.BAS  PWM
'-----
$regfile = "m328pdef.dat" 'ATmega328p
$crystal = 16000000      '16 MHz
$baud = 9600

S1 Alias Pinc.0
S2 Alias Pinc.1

Dim D As Word
Dim U As Single

Config Adc = Single , Prescaler = 64 ,
Reference = Avcc      '5 V

Config Timer1 = Pwm , Prescale = 1 ,
Pwm = 10 , Compare A Pwm = Clear Up ,
Compare B
Pwm = Clear Up

Config Lcdpin = Pin , Db4 = Portd.4 ,
Db5 = Portd.5 , Db6 = Portd.6 , Db7 =
Portd.7 , E = Portd.3 , Rs = Portd.2
Config Lcd = 16 * 2

Portc.0 = 1          'Pullup
Portc.1 = 1          'Pullup
Config Portb = Output
Cls
Cursor Off
Do
  Locate 1 , 1
  Lcd "PWMA="
  A = Getadc(3)
  Pwm1a = A
  Lcd A
  Lcd " "

  If S1 = 0 Then
    If B > 0 Then B = B - 1
  End If
  If S2 = 0 Then
    If B < 1023 Then B = B + 1
  End If
  Pwm1b = B

  Locate 2 , 1
  Lcd "PWMB="
  Lcd B
  Lcd " "
  Waitms 100
Loop
    
```

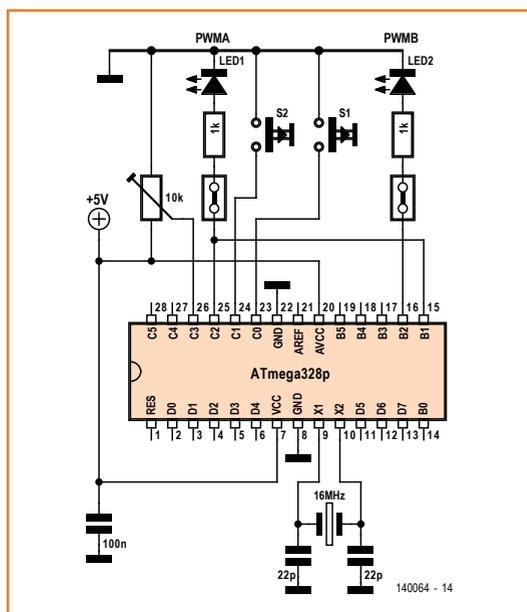


Bild 4.
Zwei LEDs an zwei PWM-Ausgängen.

(compare = vergleichen) mit dem programmierten Wert verglichen. Tatsächlich gibt es zwei solcher Compare-Einheiten, sodass man mit dem gleichen Timer zwei unabhängige PWM-Signale erzeugen kann. PWM1A kommt am Portpin PB1 heraus und PWM1B an PB2. Da war doch was? Ach ja, an PB2 liegt die LED 2 über einen Vorwiderstand von 1 k Ω . Zufall? Nein, Absicht. Denn nun kann man die Helligkeit der LED per PWM steuern.

Das Programm in **Listing 4** verwendet den PWM-Ausgang A im Zusammenhang mit dem AD-Wandler an ADC3 (Poti). Der Wert, den man am Poti einstellt, wird als Vergleichswert für das PWM-Signal verwendet. Stellt man das Poti in die Mitte, kommt ein symmetrisches Rechtecksignal heraus. Alles zwischen 0 und 1023 ist möglich. Was gerade ausgegeben wird, erscheint auch auf dem LCD. Wer sich das Signal ansehen will, kann ein Oszilloskop an PB1 (= Arduino-Pin 9) halten. Welche Frequenz hat eigentlich das PWM-Signal? Es hängt natürlich davon ab, mit welchem Takt der Timer angesteuert wird. Das kann die Quarzfrequenz 16 MHz des Arduino sein oder auch ein davon (über einen Vorteiler) abgeleiteter Teil. In diesem Fall steht in der Initialisierung die Anweisung Prescale = 1, es wird also die volle Taktfrequenz verwendet. Daraus ergibt sich eine PWM-Frequenz von $16000 \text{ kHz} / 1024 = 15,625 \text{ kHz}$ - denkt man. Aber tatsächlich ist es nur die halbe Frequenz, also ca. 7,8 kHz. Das hat damit zu tun, dass Bascom die Betriebsart

„Phase Correct PWM“ verwendet, im Gegensatz zur „Fast PWM“. Näheres dazu erfährt man im Datenblatt des Controllers, aber das ist recht umfangreich. Bascom dagegen spart Entwickler-Zeit, weil für solche Fälle immer schon eine sinnvolle Vorentscheidung getroffen wurde. Bascom funktioniert auch, wenn man noch nicht alle Details überblickt.

Die LED 2 liegt am zweiten PWM-Ausgang (PB2). Die Ausgabe wird in unserem Programm über die beiden Tasten S1 und S2 gesteuert. Damit das Programm gut lesbar ist, wurden die Namen der Tasten mit einem Alias-Befehl den Porteingängen zugeordnet. Wenn man im Programm also „If S1 = 0 Then“ schreibt, dann weiß Bascom, dass eigentlich „If Pinc.1 = 0“ gemeint ist, also der Zustand des Portpins PC1 abgefragt werden soll. Wichtig ist auch noch, dass die Taster gegen GND angeschlossen sind und deshalb einen Pullup benötigen. Normalerweise liest man dann den Ruhezustand S1 = 1. Nur wenn jemand auf die Taste drückt wird S1 = 0. Und so wird die Sache ganz einfach. Mit einem Druck auf S1 verkleinern Sie die PWM-Ausgabe (aber nur wenn sie noch größer als 0 ist) und mit S2 vergrößern Sie die Ausgabe (aber nur wenn sie noch kleiner als 1023 ist). Die ganze Sache funktioniert in Einzelschritten (wenn man kurz drückt), aber auch mit laufender Veränderung (zehn Schritte pro Sekunde) bei einem Dauerdruck. Auf dem LCD sehen Sie jeweils, was gerade ausgegeben wird.

Tastenabfragen

Das Programm zeigt ein sehr einfaches Beispiel für die Verwendung von Tasten und entsprechenden Programmverzweigungen. Doch da gibt es natürlich noch viele andere Möglichkeiten. Für eine gegebene Aufgabe die optimale Bedienung zu entwickeln, das ist eine spannende Aufgabe. Mit zwei Tastern und einem Poti kann man sehr viele unterschiedliche Dinge einstellen. Mit wenigen Tasten lässt sich viel erreichen.

Das Programm verwendet alles, was das Elektor-Shield zu bieten hat. Das LCD zeigt die beiden aktuellen PWM-Werte an, das Poti steuert den Ausgang PWMA, die beiden Taster steuern PWMB und damit die LED 2. Wirklich alles? Nein, die LED 1 ist noch arbeitslos. Also stecken Sie einen Draht in PB1 (Arduino Pin 9) und ADC2 (Arduino A2), damit liegt PWMA über 1 k Ω und

einen Jumper an der LED 1 (**Bild 4**). Nun können Sie über das Poti die Helligkeit der LED steuern. Zwei LED-Helligkeiten, die unabhängig voneinander eingestellt werden können, das ergibt gleich ein kleines Geschicklichkeitsspiel. Eine Person stellt mit den Tasten irgendeine Helligkeit an der LED 2 ein. Dann wird das LCD abgedeckt. Die zweite Person muss nun mit dem Poti die gleiche Helligkeit an der LED 1 einstellen. Das LCD enthüllt dann, wie gut das geklappt hat. Dann werden die Rollen getauscht. Alle Abweichungen

werden notiert. Wer insgesamt besser eingestellt hat, ist der Sieger und wird zum Meister mit dem unfehlbaren Blick ernannt.

(140064)

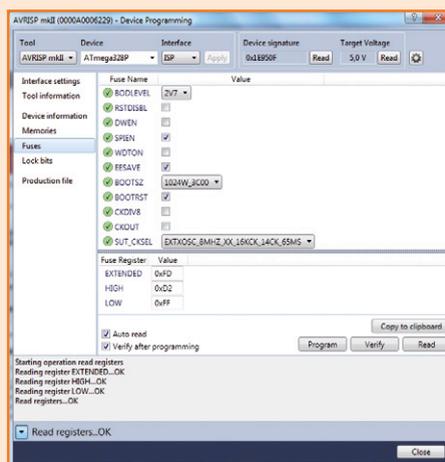
Weblinks

- [1] www.elektor-magazine.de/140064
- [2] <http://b-kainka.de/bastel49.htm>
- [3] <http://de.wikipedia.org/wiki/Polarisationsfilter>

Der MCS-Bootloader

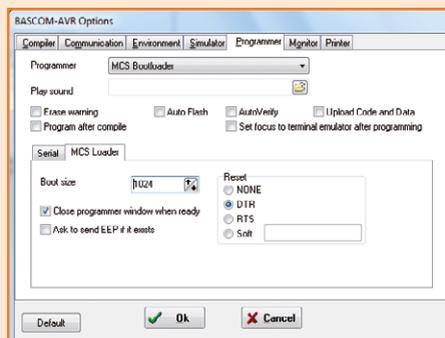
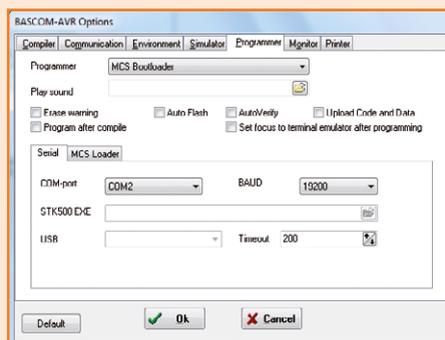
Bisher wurden zwei Wege vorgestellt, auf denen der Arduino programmiert werden kann, über den Arduino-Bootloader oder über die ISP-Schnittstelle, die übrigens auch am Elektor-Shield herausgeführt ist. Mit einem externen Brenner kann man hier auch einen anderen Bootloader aufspielen.

Bascom kennt den MCS-Bootloader, der sich an verschiedene Controller anpassen lässt. Eine schöne Sache, weil man dann verschiedene Controllerboards (auch eigene Hardware) mit demselben Bootloader ausstatten kann. Man findet den Quelltext unter den mitgelieferten Bascom-Beispielen. Nur der verwendete Controller, die Taktfrequenz und die gewünschte Baudrate müssen noch angepasst werden. Tatsächlich war der Mega328P noch nicht dabei, aber die Anpassung war einfach. Der fertig angepasste Bootloader mit Quelltext und Hex-File kann bei Elektor heruntergeladen werden [1]. Die Files heißen BootLoaderUno_16.bas (16 steht für 16 MHz) bzw. BootLoaderUno_16.hex. Achtung, die Fuses müssen etwas anders (siehe 1. Screenshot, AVR Studio) stehen als beim Arduino-Bootloader, weil mehr Speicherplatz benötigt wird. Der Bootbereich ist nun 1024 Words groß und beginnt bei



&H3C00. Wenn der MCS-Bootloader verwendet werden soll, muss Bascom entsprechend eingestellt werden. Wichtig ist auch hier wieder die korrekte COM-Schnittstelle und die richtige Baudrate, jetzt 19200 (2. Screenshot).

Und da gibt es unter dem Reiter MCS-Loader noch eine wichtige Einstellung. Wir erinnern uns, dass die Daten zwar über USB zum Arduino kommen, aber eigentlich eine serielle RS232-Schnittstelle simuliert wird. Deren Signal DTR wird bei allen Arduino-Systemen als Reset-Leitung verwendet: Eine negative Flanke löst einen Reset aus. Und dann kommt der Bootloader zum Zuge. Der Controller beginnt nämlich erstmal mit dem Programm, das bei der Boot-Adresse abgelegt ist (&H3C00). Dort wird dann auf bestimmte Aktionen des PCs gewartet. Entweder kommt ein neues Programm angefliegen oder nicht, danach wird an den Anfang des Programmspeichers bei Adresse Null verzweigt. Bei jedem Reset gibt es einen neuen Versuch. Deshalb kann Bascom über einen Pegelwechsel an DTR einen Sprung in den Bootloader erzwingen. Und der Anwender hat es ganz bequem, er muss nämlich nicht selbst auf die Reset-Taste drücken.



PIC-Strahlungsmesser

Von Reinier Ott (NL)



Alternative mit mehr Funktionen

Der „Verbesserte Strahlungsmesser“ vom November 2011 ist ein praktisches und gleichzeitig kostengünstiges Gerät zum Messen radioaktiver Strahlung. Dort wird das mit einer Fotodiode aufgefangene Signal von einem ATmega88 verarbeitet. Dass diese Aufgabe auch von einem anderen Mikrocontroller bei erweitertem Funktionsumfang übernommen werden kann, beweist der vorliegende Beitrag.

Im November 2011 stellte Autor Burkhard Kainka einen Strahlungsmesser vor, der Alpha-, Beta- und Gammastrahlung mit einer Fotodiode misst. Die Idee fand ich so spannend, dass ich den zugehörigen Bausatz im Elektor-Shop erwarb. Mit dem aufgebauten Strahlungsmessgerät habe ich ausgiebig experimentiert. Meine Idee war, die Funktionen des ATmega88 auf einen Mikrocontroller aus der PIC-Familie zu übertragen. Die Sensor-Platine habe ich mit dem Unterschied übernommen, dass die Fotodiode BPW34 zweifach vorhanden ist. Die Software für den PIC16F88 habe ich in Flowcode geschrieben. Dabei nutzte ich die Gelegenheit, weitere Funktionen hinzuzufügen.

Schaltung

Der Vorverstärker, aufgebaut auf der Platine 110538-2, ist mit der ursprünglichen Version identisch. In der Schaltung mit dem Mikrocontroller PIC16F88, die **Bild 1** zeigt, ist er nicht darge-

stellt. Der PIC16F88 wird extern mit 19,6608 MHz getaktet, intern leitet der Vorteiler eines Timers aus dem Taktsignal ein präzises Sekundensignal ab. Das Ausgangssignal des Vorverstärkers gelangt über Steckverbinder J1 und Widerstand R2 zum Anschluss RA0 des Mikrocontrollers. Dieser Anschluss ist als analoger Eingang konfiguriert. Zenerdiode D1 (5,1 V) schützt den Eingang vor Überspannungen. Die Ausgangsspannung des Vorverstärkers liegt zwar niemals über +5 V, doch falls andere Sensoren angeschlossen werden, kann dieser Schutz für den Mikrocontroller lebenswichtig sein.

Ebenso wie beim Strahlungsmesser vom November 2011 wird das Messergebnis auf einem alpha-numerischen LC-Display mit 2 x 16 Zeichen ausgegeben. Gesteuert wird das LC-Display über die Leitungen RB0...RB5 des Mikrocontrollers. Der Display-Kontrast lässt sich mit Trimpoti P2 einstellen.

Die Bedienelemente beschränken sich neben Ausschalter SW1 auf einen Modus- und einen Reset-Taster (BT1 und BT2). Der Steckverbinder J2 ist hinzugefügt, damit der Mikrocontroller in der Schaltung programmiert werden kann. Vor dem Programmieren müssen die Jumper JP1 und JP2 entfernt werden. Mit JP2 wird die Ausgangsleitung des Spannungsreglers IC3 unterbrochen, anderenfalls wäre der Spannungsregler mit der Betriebsspannung des Programmers kurzgeschlossen. Wenn Jumper JP1 gesteckt ist, lässt sich mit Taster BT2 der Reset des Mikrocontrollers herbeiführen. Im Programmier/Debug-Modus muss JP1 entfernt werden, da anderenfalls das Programmiersignal MCLR von R5 und C6 beeinflusst wird. Das kann zur Folge haben, dass der Mikrocontroller nicht programmiert werden kann.

Verglichen mit der Schaltung vom November 2011 ist neu, dass ein NF-Verstärker (IC2) und ein Lautsprecher den Messwert in ein akustisches Signal umsetzen, während eine LED (LD1) den Messwert visualisiert. Ausgang RA2 des Mikrocontrollers steuert sowohl den Lautsprecher als auch die LED.

Die Hintergrundbeleuchtung des LC-Displays wird mit Jumper JP3 zuschaltbar. Der 9-V-Batterie wird dann ein vergleichsweise hoher Strom entnommen. Hier besteht die Gefahr, dass die Schaltung infolge von Instabilitäten des Spannungsreglers IC3 (78L05) zu schwingen beginnt. Ohne Hintergrundbeleuchtung beträgt die Stromaufnahme ungefähr 20 mA, so dass eine 9-V-Alkaline-Batterie die Energie für etwa acht Betriebsstunden liefern kann.

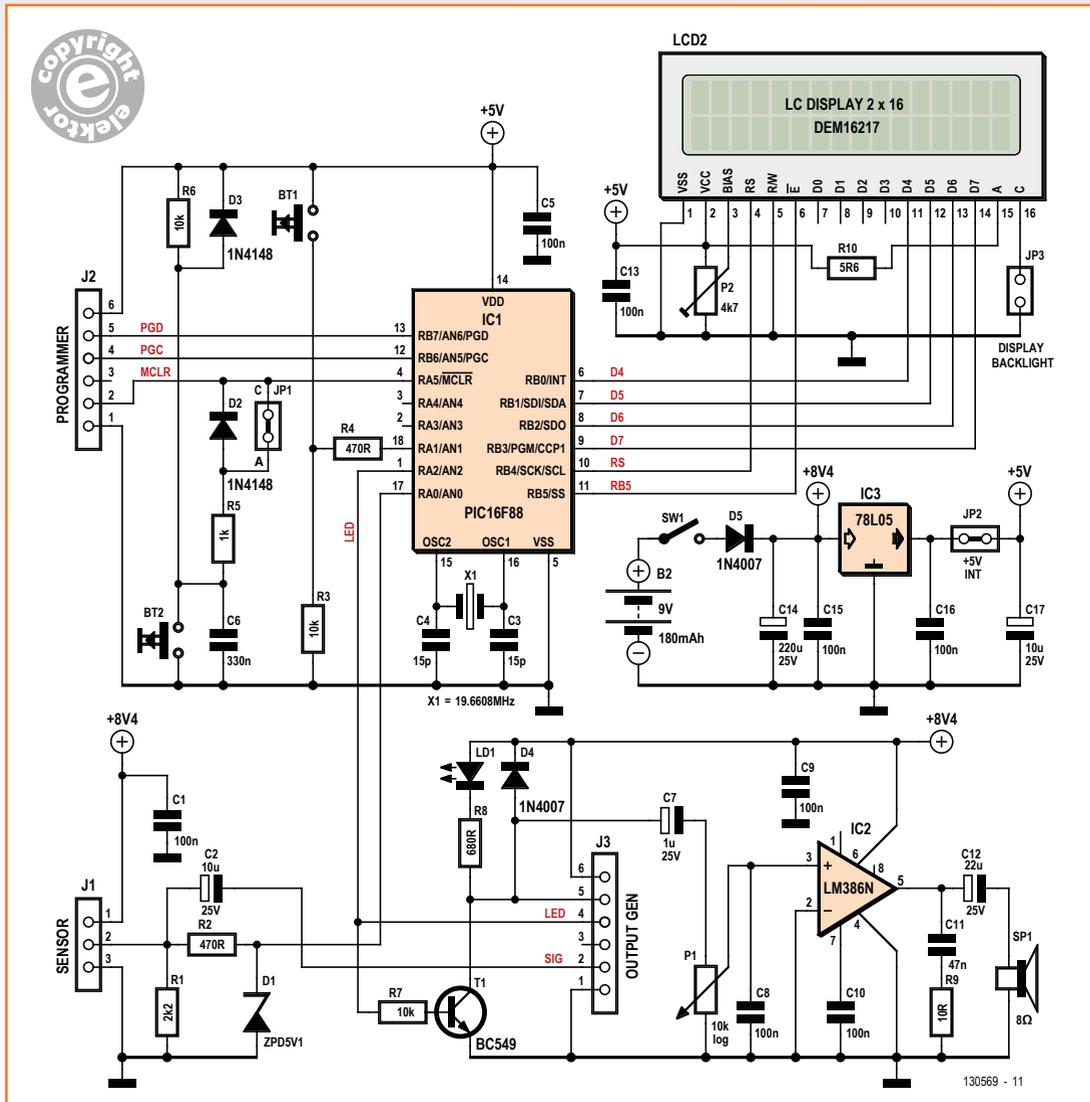


Bild 1. Schaltung des Strahlungsmessers mit dem Mikrocontroller PIC16F88.

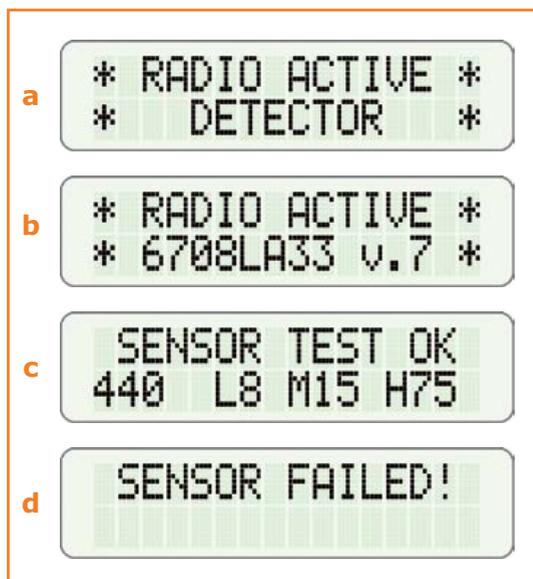


Bild 2.
Nach dem Start gibt das LC-Display diverse Meldungen aus.

Arbeitsweise

Nach dem Einschalten der Betriebsspannung oder dem Reset des Mikrocontrollers erscheinen auf dem LC-Display die Bezeichnung des Systems (**Bild 2a**), gefolgt von Daten des Autors sowie der Firmware-Version (**Bild 2b**). Zeitgleich bestimmt der Mikrocontroller den Rauschlevel (wie bei der ersten Version des Strahlungsmessers), anschließend werden folgende Daten ausgegeben (**Bild 2c**):

- Rauschlevel in bit, hier 440 (10-bit-ADC, Bereich 0...1023).
- Wert L, *Low Threshold*, hier 8. Dieser Wert wird zum Rauschlevel addiert, um die höchste Empfindlichkeit einzustellen.
- Wert M, *Moderate Threshold*, hier 15,



Bild 3.
Messmodus (a),
Intensitätsmodus (b) und
Einstellen der Startwerte (c).

für eine mittlere Empfindlichkeit des Messsystems.

- Wert H, *High Threshold* für das Messen hochenergetischer Strahlung.

Die Werte L, M und H sind über Taster-Kombinationen jederzeit neu definierbar, sie werden wie bei der ersten Version des Strahlungsmessers im EEPROM des Mikrocontrollers abgelegt.

Falls der Sensor nicht angeschlossen oder defekt ist, erscheint die Fehlermeldung, die **Bild 2d** zeigt. Zusätzlich signalisiert die rote LED den Fehler, indem sie kontinuierlich aufleuchtet.

Nach diesem etwa fünf Sekunden dauernden Programmvorspann startet der Messbetrieb. Auf dem LC-Display werden Daten ausgegeben, wie sie in **Bild 3a** wiedergegeben sind. In der oberen Zeile erscheinen die Strahlungswerte L (*Low*), M (*Moderate*) und H (*High*) als Anzahl der gezählten Impulse. Bei jedem Impuls blinkt die rote LED kurz auf, und aus dem Lautsprecher ertönt ein charakteristisches „Tick“. Die Lautstärke lässt sich mit dem Potentiometer P1 einstellen.

Die zweite Zeile zeigt die verstrichene Zeit seit Beginn der Messung sowie die Anzahl der in einer Minute gemessenen Low-Impulse an. Nach Ablauf einer Minute wird der Wert aktualisiert, die Anzeige wird im Sekunden-Rhythmus aufgefrischt.

Bedienung Drucktaster

Ein Reset des Mikrocontrollers kann mit Drucktaster BT2 herbeigeführt werden. Nach dem Loslassen startet der vorstehend beschriebene Vorspann. Wird innerhalb der ersten drei Sekunden ein beliebiger anderer Taster gedrückt (Mode oder BT1), verzweigt das Mikrocontroller-Programm in den so genannten Pulsgeschwindigkeitsmodus (Impulse in der Sekunde), der zum Messen von Strahlungsintensitäten vorgesehen ist (**Bild 3b**). In der ersten Zeile wird die Anzahl der (im empfindlichsten Bereich = L) innerhalb einer Sekunde gemessenen Impulse ausgegeben. Die zweite Zeile stellt diesen Wert in Form eines aus Rauten bestehenden, längenveränderlichen Balkens dar.

Strahlungslevel programmieren

Die drei Strahlungslevel sind programmierbar, indem zuerst Taster BT1 (Modus) gedrückt und gedrückt gehalten wird. Danach wird BT2 (Reset)

gedrückt, BT1 wird etwa drei Sekunden nach Beginn des Vorspanns auf dem Display losgelassen. Beim Drücken von Taster BT1 werden die drei Level L, M und H im Sekundenabstand erhöht, nach Loslassen des Tasters bleiben die Werte stehen (siehe **Bild 3c**). Wenn die übrigen Level auf die gleiche Weise eingestellt sind, erscheint die Meldung, dass die neuen Level gespeichert wurden. Die neuen Werte werden nach einem Reset wie in Bild 2c angezeigt.

Software

Die Software wurde vollständig in Flowcode 4 entwickelt. Für leichte Lesbarkeit sorgt der modulare Aufbau in Makros (Subroutinen). Das Hauptprogramm besteht aus drei Teilen (Modi). Die Initialisierung ist zuständig für das Bestimmen des mittleren Rausch-Niveaus, ähnlich wie beim Strahlungsmesser vom November 2011. Hier kommt jedoch ein 10-bit-ADC zum Einsatz, so dass die Auflösung des Signal-Rausch-Verhältnisses verbessert ist. Die Praxis hat gezeigt, dass der Wert des Rauschsignals meistens nahe 440 liegt. Der Wert entscheidet auch darüber, ob das vom Sensor kommende Signal als gültig betrachtet wird. Der Schwellenwert ist 100, wenn er darunter liegt (fehlender oder defekter Sensor), wird auf dem LC-Display eine Fehlermeldung ausgegeben. Das Programm kann dann nicht in die Messschleife verzweigen.

Im ersten Programmteil werden auch die 10-bit-Schwellenmesswerte L, M und H festgelegt. Diese Werte werden aus dem mittleren Rausch-Level berechnet (ebenfalls ein 10-bit-Wert). Hinzuaddiert werden Korrekturwerte, die für jeden Level als individuelle 8-bit-Werte im EEPROM des Mikrocontrollers abgelegt sind. Hier entspricht dem niedrigsten Wert die höchste Sensor-Empfindlichkeit. Im ersten Programmteil kann der Benutzer den gewünschten Programmmodus wählen:

- Regulärer Impuls-Zählmodus für die drei Strahlungsbereiche,
- Strahlungsintensität (Impulse in der Sekunde) mit Balkenanzeige,

- Programmiermodus für die drei Empfindlichkeiten L, M und H.

Nachdem der erste Programmteil durchlaufen ist, verzweigt der Mikrocontroller in das Hauptprogramm. Für eine schnelle Messfolge ist wichtig, dass die zum Eingangssignal gehörende Abtastschleife möglichst kurz ist. Auch schnell aufeinander folgende Impulse müssen zuverlässig erkannt werden.

Versuche haben gezeigt, dass an genügend starken radioaktiven Präparaten, zum Beispiel an den Leuchtziffern eines alten mechanischen Weckers, Messwerte von mehr als 6000 Impulse



Bild 4. Hardware des Strahlungsmessers, eingebaut in ein speziell angefertigtes Gehäuse.

in der Minute auftreten. Man kann annehmen, dass diese Messwerte noch verlässlich sind; die Messschleife wird oft genug durchlaufen, so dass keine Impulse verloren gehen.

Ohne den Geschwindigkeitsverlust, verursacht durch die Messwertausgabe auf dem LC-Display, liegt die Messschleifen-Frequenz noch viel höher (ungefähr bei 9...10 kHz).

Das analoge Signal gelangt zum Eingang RA0 des Mikrocontrollers, es wird in einen digitalen

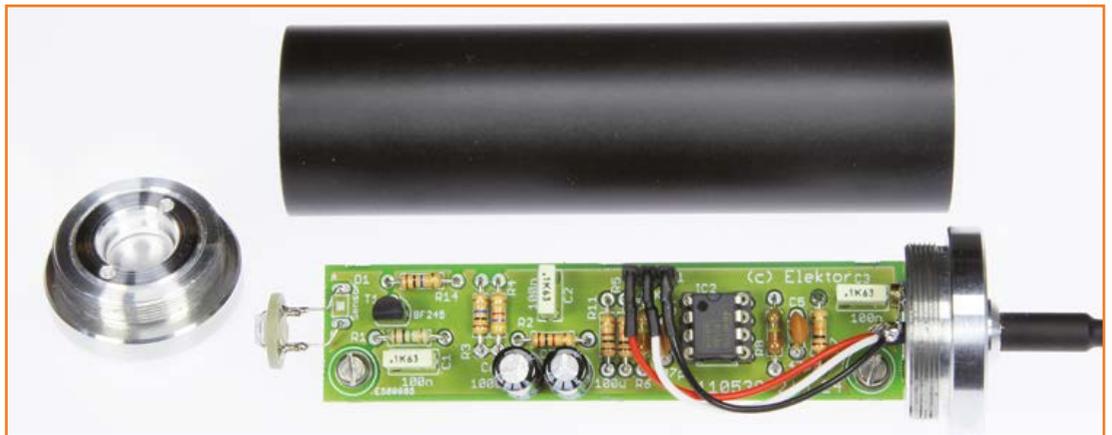


Bild 5.
Auch das Gehäuse des Messkopfs entstand in der eigenen Werkstatt.

10-bit-Wert gewandelt (Definition INT in Flowcode). Zuerst wird der Wert mit dem Wert des Rausch-Levels verglichen, zu dem der Wert L addiert ist (Low Signal Threshold Level). Dieser Wert wurde im ersten Programmteil definiert. Erst wenn ein Signal auftritt, das den Vergleichswert übersteigt, wird das Signal auch mit den Werten M und H verglichen. Die Reihenfolge der Schritte wurde so gewählt, dass der Mikrocontroller in der Messschleife möglichst wenige Programmzeilen abarbeiten muss. Nach Erkennen eines Sensorsignals wird die zugehörige Variable fortlaufend inkrementiert, der Wert wird im Sekundenabstand auf dem LC-Display angezeigt. Bei diesem Verfahren ist zu beachten, dass H-Signale auch als M- und L-Signale und M-Signale auch als L-Signale interpretiert werden. Falls ein L-Signal erkannt wird, kann es möglicherweise so stark sein, dass es auch als M- oder sogar H-Signal betrachtet wird. Das Programm wird von einem Timer-Interrupt gesteuert, der auf die Frequenz 75 Hz eingestellt ist. Daraus werden die auszugebende Zeit sowie die auf eine Sekunde festgelegte Aktualisierungs-

frequenz des Displays abgeleitet. Das Programm ist so strukturiert, dass die Messgeschwindigkeit möglichst wenig von den LC-Display-Anzeigeroutinen beeinflusst wird. In dieser Phase führt der Mikrocontroller auch die Zwischenberechnungen durch. Einige Programmzeilen in C, eingebaut in die Wiedergabe-Makros von Flowcode, lassen die Zeit im Format hh:mm:ss erscheinen.

Wie schon erwähnt, ist im Programm ein Modus zur Anzeige der Impulsgeschwindigkeit implementiert. Die Messwerte werden durch einen Balken (*Gauge*) visualisiert, die Balkenlänge ist ein Maß für die gemessene Strahlungsintensität. Diese Anzeige arbeitet (ungefähr) logarithmisch, damit sie trotz der geringen Auflösung des 16-Zeichen-LC-Displays aussagekräftig ist. Da Flowcode für den PIC16F88 nur wenige mathematische Funktionen bereithält, wurde ein unkomplizierter, in C geschriebener Abschnitt implementiert (*supplement code* unter *project options*):

```
short GaugeValue(int p)
// Position of Gauge bar
{
    short y;
    if (p>1000) y=16; else
        if (p>500) y=15; else
            if (p>250) y=13; else
                if (p>120) y=12; else
                    {
                        y=p/10;
                    }
    return (y);
}
```

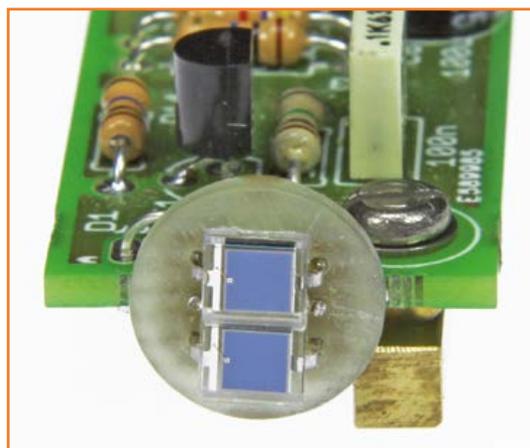


Bild 6.
Der Sensor besteht aus zwei parallel geschalteten Fotodioden BPW34.

Ganz am Rand ist noch anzumerken, dass der

erste Programmteil auch „eingebrennte“ Daten enthält. Dort kann ein persönlicher Code stehen, beispielsweise die Postleitzahl und Hausnummer des Gerätebesitzers. Diese Daten erscheinen beim Programmstart auf dem LC-Display. In der Version, die von [2] heruntergeladen werden kann, ist im letzten Block des ersten Flowcode-Makros der String 1234AB56 v0 eingetragen. Dort kann auch jede andere Zeichenkombination stehen.

Konstruktion

Die Sensor-Platine mit dem Sensor (siehe **Bild 5**) wird in einem hermetisch lichtdichten Gehäuse untergebracht. Das aus Alufolie bestehende Fenster in der Gehäusefront lässt sich schnell auswechseln, es wurde mit einem Federring festgeklemmt. Versuche mit unterschiedlichen Fenstermaterialien sind unkompliziert durchführbar. Solche Experimente können recht aufschlussreich sein, auch handelsübliche Alufolien für den Haushalt haben längst nicht immer identische Eigenschaften.

Bei der Montage ist darauf zu achten, dass die beiden Fotodioden leicht gegen die Alufolie drücken. Wie schon im Beitrag vom November 2011 beschrieben, wird dadurch einem unerwünschten, manchmal recht hartnäckigen Mikrofonie-Effekt entgegengewirkt. Die gegen das Alufolien-Fenster federnde Befestigung der Platine kann den Effekt nicht vollständig verhindern. Bei der im Bild gezeigten Konstruktion half eine dünne Lage Silikon-Paste zwischen der Alufolie und den Fotodioden ein gutes Stück weiter. Die Silikon-Schicht bewirkt eine mechanische Dämpfung, sie muss möglichst dünn aufgetragen werden, da sie die Strahlungsdurchlässigkeit mindert. Ferner hat sich bewährt, die Platine an sämtlichen Kanten im Gehäuse zu fixieren, insbesondere wenn der Messkopf als separate Einheit ausgeführt wird.

Aus **Bild 6** ist ersichtlich, dass zwei Fotodioden vorhanden sind. Da sie parallel geschaltet sind, wird die Sensor-Oberfläche verdoppelt. Im Beitrag vom November 2011 wurde diese Überlegung mit der Begründung verworfen, dass sich auch die parasitären Kapazitäten mindestens verdoppeln. Die inzwischen gesammelten Erfahrungen haben jedoch gezeigt, dass dieser nachteilige Effekt in Kauf genommen werden kann. Die gesteigerte Empfindlichkeit ist wichtiger, zumal meistens nur relativ schwache Strahlungen gemessen werden. Dem kommt auch entgegen, dass die Messfrequenz vergleichsweise niedrig ist (siehe Beschreibung der Software), weil ein Mikrocontroller die Messdaten verarbeitet.

(130569)gd

Weblinks

[1] www.elektor-magazine.de/110538

[2] www.elektor-magazine.de/130569

You CAN get it...

Hardware & Software für CAN-Bus-Anwendungen



PCAN-USB

CAN-Adapter für den USB-Port. Optional auch mit galvanischer Trennung erhältlich. Inklusive Programmierschnittstelle und CAN-Monitor für Windows.

ab 195 €

PCAN-Router

Frei programmierbarer CAN-Router mit 2 High-Speed-CAN-Kanälen. Mit D-Sub- oder Phoenix-Anschlussteckern im Aluprofilgehäuse erhältlich.

ab 200 €

www.peak-system.com

Otto-Röhm-Str. 69, 64293 Darmstadt, Germany
Tel.: +49 6151 8173-20 - Fax: +49 6151 8173-29
info@peak-system.com

PEAK
System

Alle Preise verstehen sich zzgl. MwSt., Porto und Verpackung.
Irrtümer und technische Änderungen vorbehalten.

UNSER WICHTIGSTES WERKZEUG

LEITERPLATTEN WEITER GEDACHT.



**SCHNELLE
VIELFALT IST
BESSER!**
Online kalkulieren. Online bestellen.

LEITON
RECHNEN SIE MIT BESTEM SERVICE

Wir denken weiter. Als lösungsorientiertes Unternehmen möchten wir unsere anspruchsvollen Kunden immer wieder mit neuen Ideen und sinnvollen Innovationen begeistern. **Als ein führender Online-Anbieter von Leiterplatten** bieten wir erstklassige Leistungen in den Bereichen **Service, Fairness, Technologie, Auswahl, Komfort, Geschwindigkeit und Zuverlässigkeit.** Unsere Webseite bietet eine enorme Auswahl an Lösungen in der Online-Kalkulation sowie weiterführendes Wissen, Informationen und Entwicklerwerkzeuge. Und sollte einmal etwas sehr Ausgefallenes gefordert sein, finden wir gerne Ihre **persönliche Lösung.** Sie können bei LeitOn immer mit bestem Service rechnen.

www.leiton.de

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

SAME: Chip-8 Videogame-Emulator

Single Arcade Machine Emulator: Virtuelle Maschine auf PSoC



Von **Camille Jourdan Gay** (Japan, ccamyy@gmail.com)
und **Luc Lemmens** (Elektor-Labor)

Wir wollen unsere Leser nicht animieren, kostbare Zeit mit Videospiele zu verbringen. Hier geht es vor allem um die Kunst des Programmierens. Doch wo steht geschrieben, dass das Eine das Andere ausschließt?

Dass in diesem Beitrag nur wenig Programm-Code steht, widerspricht zwar der Überschrift, doch das ist unvermeidlich. An dieser Stelle haben wir nicht den Platz, die Details erschöpfend zu behandeln. Stattdessen haben wir ein Online-Dokument erstellt, das von der Projektseite heruntergeladen werden kann [2]. Dort ist der Programm-Code vollständig gelistet, klar strukturiert und ausführlich kommentiert. Der Quell-Code gehört selbstverständlich auch dazu.

Chip-8 anno 2014?

Die virtuelle Plattform Chip-8 [1] existiert seit mehr als vier Jahrzehnten. In den 1970er und 1980er Jahren war sie recht verbreitet, insbesondere wenn es um Videospiele ging. Der Clou von Chip-8 war (und ist), dass Spiele unabhängig von der Hardware programmiert werden können. Das Konzept der *Virtuellen Maschine* ist auch heute

noch aktuell, es wird beispielsweise bei *Java* angewandt. Die Software-Plattform imitiert und simuliert, oder besser gesagt, sie *emuliert* die Funktionen eines Betriebssystems sowie einen Prozessor, der real nicht existiert. Chip-8-Videospiele werden für virtuelle Plattformen geschrieben. Die Videospiele laufen auch auf Hardware, welche dafür ursprünglich nicht vorgesehen war. Vorausgesetzt wird lediglich, dass auf dem Zielsystem eine Virtuelle Maschine lauffähig ist. Dies gilt auch für die eingebetteten Systeme, die in jedem Haushalt vielfach vertreten sind. Dazu gehören der Computer ebenso wie das Smartphone, doch auch die Spülmaschine oder der intelligente Kühlschrank lassen sich als Gameboy zweckentfremden! Der Typ des Mikrocontrollers ist bedeutungslos, wenn darauf lediglich ein einfacher Interpreter laufen soll. Auf der Zutatenliste stehen hier nur der Mikrocontroller, ein Display, einige Schalter oder Taster sowie ein Interpreter für den Chip-8-Code. Im Internet ist das Angebot für Chip-8-Spiele unüberschaubar.

Dieses Projekt ist ein Lehrstück, das den Anwender spielerisch in die Elemente der System-Architektur einführen möchte. System-Komponenten wie Register, Stack, Spei-

cher mit Adressraum und der Programmzähler bleiben keine Unbekannten. Die Eckpfeiler der Architektur werden transparent...

Der Kern unserer virtuellen Maschine hat mit realen Mikrocontrollern viel gemeinsam. Er setzt sich aus einem Prozessor, den Registern, dem Stack, dem Speicher und zwei Timern zusammen. Die virtuelle Maschine kann 35 Instruktionen ausführen, die 16 bit breit sind und hier *Opcodes* heißen. Bereits mit diesem knappen Befehlssatz lassen sich Videospiele programmieren. Real werden noch einige Taster für die Bedienung, ein anspruchsloses grafisches Display und eventuell ein Beeper benötigt. Ein populäres, aus dem Internet herunterladbares Chip-8-Videospiel ist Tetris, dort *Tetris.ch8* genannt. Der Chip-8-Code wird von einem *Chip-8-Interpreter* abgearbeitet. Chip-8-Interpreter werden in einer Sprache geschrieben, die für den Mikrocontroller des Zielsystems spezifisch ist. Der Interpreter übersetzt die Chip-8-Opcodes in Kommandos, die der Mikrocontroller ausführen kann. Als Beispiel sei der Opcode 00E0 im Quellcode eines Spiels genannt, der das Löschen eines einzeiligen Displays bewirkt. Meistens sind auf dem Display des Zielsystems mehrere Zeilen vorhanden, so dass das Kommando an die reale Situation angepasst werden muss.

Der Speicher des Systems, auf dem der Interpreter läuft, ist zweigeteilt. Einerseits hat der Interpreter eine eigene Arbeitsumgebung für die Verwaltung der Variablen, als Schnittstelle zur Hardware und ähnliche Aufgaben. Auf der anderen Seite müssen Ressourcen für die virtuelle Chip-8-Architektur bereitgestellt werden. Dazu gehören virtuelle Variablen, der *Program Counter* und der *Stack Pointer*.

Der Chip-8-Code ist in einem Byte-Array gespeichert. Der Interpreter liest den Chip-8-Code und übersetzt ihn in den Code des Mikrocontrollers. Ein Übersetzungsalgorithmus lässt sich in C mit einer Switch-Case-Konstruktion implementieren:

```
while(1)
{
    opcode = gameMemory[programCounter];
    switch(opcode)
    {
        case 0001: ExecuteOpcode0001();
            break;
        case 0002: ExecuteOpcode0002();
            break;
    }
}
```

Implementierung und Ressourcenverbrauch

Der Chip-8-Standard definiert nur die Architektur der Anwendung, bei der Implementierung hat der Programmierer viele Freiheiten. Ein Beispiel ist die Berechnung des 16-bit-Opcodes aus zwei 8-bit-Werten, die im virtuellen Arbeitsspeicher liegen. Der Programmierer hat hier folgende Alternativen: Entweder er definiert ein Makro:

```
#define OPCODE ((READ_PROGRAM_MEMORY(PC) << 8) |
READ_PROGRAM_MEMORY(PC+1))
```

oder er definiert eine Variable im Arbeitsspeicher:

```
short OPCODE = ((READ_PROGRAM_MEMORY(PC) << 8) |
READ_PROGRAM_MEMORY(PC+1))
```

Die erste Alternative benötigt bei wiederholten Aufrufen mehr Zeit, spart aber realen Speicher, weil keine Variable definiert werden muss.

```
//...
}
programCounter++;
}
```

Einige Opcodes wie ADD und SUB sind fast buchstäblich übersetzbar. Andere Funktionen erfordern einen höheren Aufwand, beispielsweise das Darstellen von Grafik auf dem Display oder das Aktivieren des Zufallsgenerators, wie es bei Spielen häufig vorkommt.

Unser Interpreter ist quelloffene Software, sie wird ausführlich in dem 31-seitigen Dokument beschrieben, das von [2] heruntergeladen werden kann. Dort wird im Detail erklärt, wie die Opcodes interpretiert und übersetzt werden. Wir empfehlen, sich die Zeit zu nehmen, die zum Durcharbeiten des Dokuments nötig ist.

An dieser Stelle wollen wir uns auf das Realisieren eines Embedded-Systems konzentrieren, auf dem Chip-8-Spiele lauffähig sind. Wir mussten uns auf eine bestimmte Hardware festlegen, doch das Grundprinzip ist auf andere Hardware mit vergleichbarer Konstellation der Ein- und Ausgänge übertragbar.

Das oben genannte Dokument gibt genaue Auskunft über die Konfiguration des Mikrocontrollers, zum Beispiel über Eingänge und Timer, sowie über die internen Abläufe des Interpreters. Unsere Herangehensweise soll lediglich Anregungen und Impulse geben, auch hier gilt, dass viele Wege zum Ziel führen. Die Hürden, die dabei genommen werden müssen, dürften jedoch ähnlich sein. Der ökonomische Umgang mit dem Speicher und die begrenzte Anzahl der Eingänge sind nur zwei von ihnen.

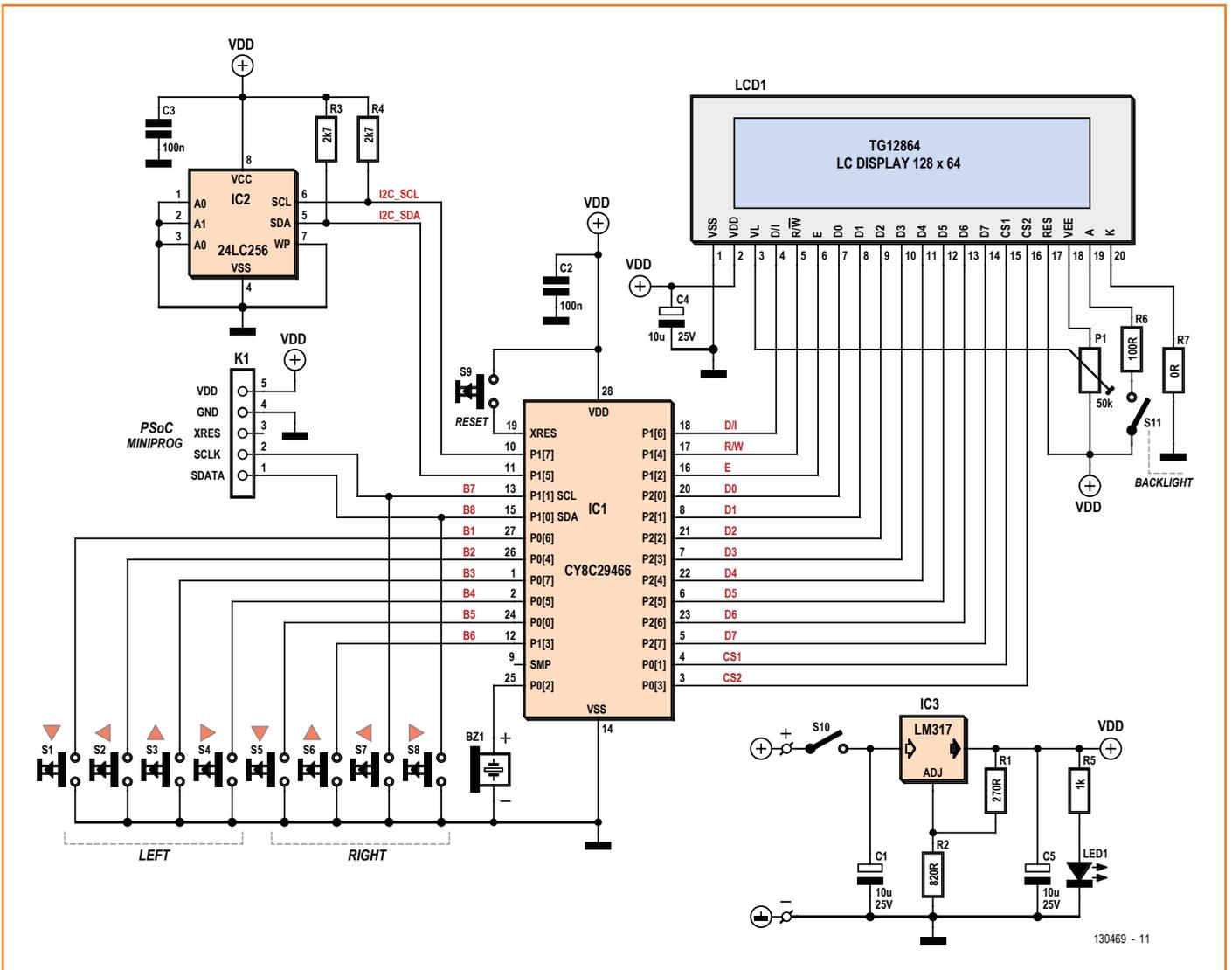


Bild 1.
Die Hardware sieht unscheinbar aus, doch der Schein trügt. Im PSoC ist ein Arsenal programmierbarer Module versammelt.

Reale Hardware

Wir wollen nun auflisten, was wir real an Hardware brauchen:

- Grafisches Display 64 x 32 Pixel oder größer,
- 16 Taster (für die meisten Spiele genügen aber drei oder vier Taster)
- akustischer Signalgeber, es sei denn, das Spiel ist tonlos,
- Mikrocontroller, er muss für die virtuelle Chip-8-Maschine genügend leistungsstark sein:
 - 35 Befehle mit 16 bit (Opcodes),
 - 4 KB Arbeitsspeicher,
 - 16-bit-Stack, 16 Ebenen, für die Rückkehradressen,
 - 16 Register, 8 bit, für allgemeine Anwendungen,

- 3 Register, 8 bit, für besondere Zwecke (Speicher-Pointer, Programmzähler und Stack-Pointer),
- 2 Timer/Counter, 8 bit, ein *Delay Timer* und ein *Sound Timer*.

Beim Autor ist ein System im Einsatz, auf dem auch eine neue Version von Chip-8 läuft. Die Version hat den Namen *Super Chip (S-Chip)*, sie kann größere Displays mit 128 x 64 Pixeln steuern, die Anzahl der Codes und Register ist erweitert. Von außen betrachtet ist die Hardware nicht sehr umfangreich (**Bild 1**). Außer den Tastern, dem LC-Display und einem Spannungsregler sind nur zwei ICs vorhanden: Der PSoC 29466, der hier die Funktion des Mikrocontrollers übernimmt, sowie ein I²C-EEPROM. Und dann noch der Beeper, er gehört natürlich auch dazu.

Eine spielerische Übung für den Umgang mit Registern, Stack, Speicher-Adressräumen, Programmzähler...

Zunächst hatten wir geplant, nur ein einziges Spiel im Speicher des Mikrocontrollers unterzubringen. Daraus wurden acht Spiele, die nun ein extra EEPROM erfordern. Auf dem Display muss eine Auswahlliste erscheinen, und da mehr als ein Spiel wählbar ist, kam ein Reset-Taster hinzu. Bei einer Version mit nur einem Spiel kann der Reset-Taster entfallen.

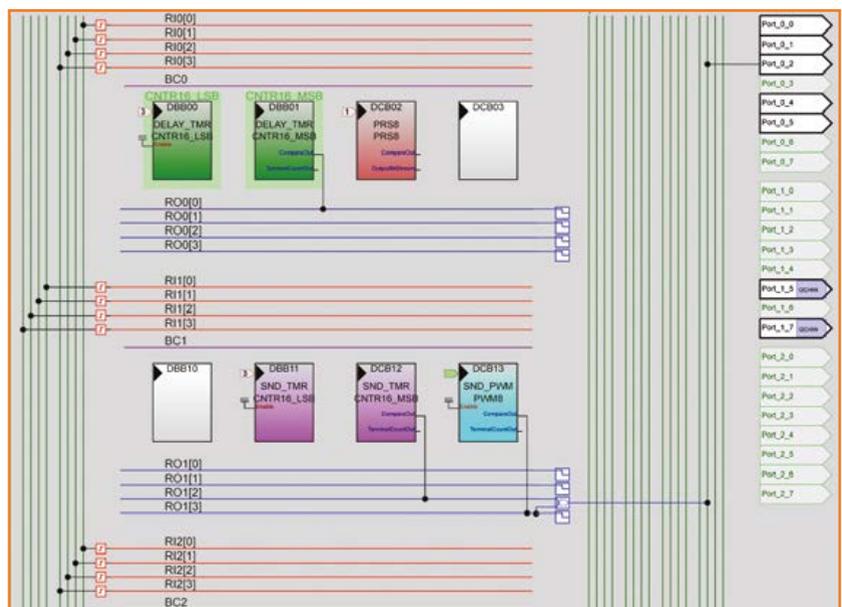
Ein PSoC ist ein *Programmable System on Chip*, ein hoch integriertes System mit einem Mikrocontroller und diverser digitaler und analoger Peripherie wie Zähler, Wandler, Filter und Verstärker auf einem einzigen Chip. Alle Komponenten sind auf der Ebene der Software konfigurierbar und miteinander kombinierbar. Für den Chip-8 Game Emulator fiel unsere Wahl auf den PSoC 29466 von Cypress. Ausschlaggebend waren der Preis von 30 € für den unvermeidbaren Programmierer und nur 5 € für den Chip. Natürlich war auch wichtig, dass dieser Typ das an Bord hat, das wir uns hier wünschen: Zwei Timer, ein PWM-Generator für die akustischen Signale und nicht zuletzt einen Zufallsgenerator. Externe Pullup-Widerstände für die Taster entfallen, denn auch sie sind integriert, konfigurierbar als Pullup oder Pulldown. Darauf kommen wir noch zurück.

Die 8-bit-Timer/Counter haben wir mit mehreren „Modulen“ des PSoCs realisiert. Die Module lassen sich konfigurieren, indem bestimmte Werte in zugehörige Register gesetzt werden. Noch wesentlich einfacher geht dies mit einem Tool, das *PSoC Designer* heißt, eine grafische Oberfläche, die kaum Wünsche offen lässt. Im PSoC Designer können die Module beliebig arrangiert werden, die Modul-Verbindungen sind frei konfigurierbar, auch das Zwischenschalten kombinatorischer Logik-Elemente ist möglich. Mit wenigen Mausklicks lassen sich Ports konfigurieren oder globale Variablen definieren, beispielsweise für die Taktfrequenz des Mikrocontrollers. Der PSoC Designer generiert selbstständig die Software-Funktionen, zum Beispiel `Timer1_Start()`, `Timer1_Stop()`. Die Portpins der Taster sind als Eingänge mit

internen Pullup-Widerständen konfiguriert, während alle übrigen Portpins die Funktion von Ausgängen haben. In **Bild 2** ist wiedergegeben, wie dies im PSoC Designer aussieht. Die Timer haben eindeutige, unverwechselbare Namen: `DELAY_TMR` und `SND_TMR`. `SND_TMR` zählt rückwärts, der akustische Signalgeber `Bz1` ist aktiv, bis der Zähler auf Null steht. Mit dem Port `ET` und `SND_PWM` wurde ein PWM-Generator realisiert; er erzeugt ein 4-kHz-Signal, das von `Bz1` hörbar gemacht wird. `PRS8` bezeichnet den Zufallsgenerator, den Chip-8 zum Ausführen des Opcodes `RND` benötigt. Nachfolgend sind noch einige Besonderheiten zusammengefasst, die den PSoC betreffen:

Taster: Den Status der Taster `S1...S8` liest eine Funktion, die eine Variable zurückgibt. In der Variablen sind die Bits entsprechend dem Status der Taster gesetzt oder rückgesetzt. Wenn die internen Port-Widerstände als Pullup konfiguriert sind, müsste das Bit beim Loslassen des Tasters von 0 nach 1 zurückkehren. Das geschieht jedoch nicht, das Bit bleibt auf 0 stehen. Abhilfe ist mit einem Kunstgriff möglich: Wenn der Sta-

Bild 2. Konfigurieren der Module im PSoC Designer.



Page	Lines	Column	Address(0~63)	Data
1st page(X=0)	Line 0	0 1 1 1 0 0	0 0 1 0 0 0	DB0(LSB)
	Line 1	1 0 0 0 1 0	0 0 1 1 0 0	DB1
	Line 2	1 0 0 0 1 0	0 0 1 0 1 0	DB2
	Line 3	1 0 0 0 1 0	0 0 1 0 1 0	DB3

	Line 6	1 0 0 0 1 0	1 1 1 0 0 0	DB5
	Line 7	0 0 0 0 0 0	0 0 0 0 0 0	DB7(MSB)
2nd page(X=1)	Line 8	1 1 1 1 0 0	0 1 1 1 0 0	DB0(LSB)
	Line 9	1 0 0 0 1 0	0 1 0 0 1 0	DB1
	Line 10	1 0 0 0 1 0	0 1 0 0 1 0	DB2

	Line 13	1 0 0 0 1 0	0 1 0 0 1 0	DB4
	Line 14	1 0 0 0 1 0	0 1 0 0 1 0	DB5
	Line 15	1 1 1 1 0 0	0 1 1 1 0 0	DB6
8th page(X=7)	Line 56	1 0 0 0 1 0	0 0 0 0 0 0	DB0(LSB)

	Line 61	1 0 0 0 1 0	1 0 0 1 0 0	DB4
	Line 62	1 0 0 0 1 0	0 1 1 0 1 0	DB5
	Line 63	0 0 0 0 0 0	0 1 1 0 1 0	DB6

	Line 63	0 0 0 0 0 0	0 0 0 0 0 0	DB7(MSB)

Bild 3. Der Aufbau eines Zeichens auf dem Display gleicht in Chip-8 einem Puzzle.

tus gelesen werden soll, wird jedes Mal vorher der Pullup gesetzt.

Display: Für das Ansteuern von LC-Displays existieren typabhängig zahllose Varianten. Wir haben uns für ein LC-Display mit dem Chipset KS0108 entschieden, bei dem paralleles *Bit-banging* angewandt wird. Vom Signal an Eingang D/I hängt ab, ob die Signale als Kommandos oder

Daten interpretiert werden, Signal R/W schaltet zwischen Lesen und Schreiben um, und mit CS1 und CS2 werden der linke und rechte Teil des Displays selektiert ($128 \times 64 = 2 \times 64 \times 64$). Eingang EN (von *Enable*) dient zur Synchronisation. Die Daten, die gelesen oder geschrieben werden sollen, liegen an den acht Ausgängen von Port 2. Die Dokumentation [2] enthält ein Makro, mit dem dieser 8-bit-Port als Eingang (HIGHZ) oder Ausgang (STRONG) genutzt werden kann.

Timer: Der PSoC 29466 hat drei Oszillatoren an Bord, VC1, VC2 und VC3. Dabei wird der System-Takt von 24 MHz heruntergeteilt. Hier sind die Teilfaktoren so gewählt, dass VC1 = 1,5 MHz, VC2 = 93 kHz und VC3 = 360 Hz sind. Dies ist gleichzeitig die niedrigste Frequenz, die der mit 24 MHz getaktete PSoC direkt generieren kann. Da Chip-8 auch den Takt 60 Hz benötigt, wird VC3 mit einem virtuellen Timer durch 6 geteilt. Auch dies wird in der Dokumentation ausführlich beschrieben. Ferner enthält die Dokumentation Erläuterungen zum Zusammenspiel zwischen dem PSoC, dem Zähler-Ausgang und dem PWM-Modul.

Der Interpreter

Primäre Aufgabe des Interpreters ist das Emulieren der Chip-8-Architektur im Speicher des Mikrocontrollers. Die benötigten Register, insgesamt 19, belegen einige Bytes, dies ist kein Problem. Für das Spiel müssen jedoch 4 KB verfügbar sein, während der Arbeitsspeicher des PSoCs nur 2 KB umfasst. Wie lässt sich dieses Problem lösen? Wir verwenden einen Software-Kunstgriff, den wir hier jedoch nur global beschreiben können. Der Kunstgriff läuft darauf hinaus, dass wir eine spezifische Fähigkeit unseres PSoCs nutzen. Mikrocontroller haben diese Eigenschaft normalerweise nicht. Unser PSoC lässt sich mit I²C-Kommandos dynamisch neu programmieren, ähnlich wie ein EEPROM. Auf diesem Weg können wir auch ein neues Spiel in den PSoC laden, ohne den gesamten PSoC neu programmieren zu müssen. Auch auf dieses Thema geht unsere Dokumentation ein. Nachdem die Speicherarchitektur feststeht, können wir uns dem Interpreter zuwenden. Aus dem nachstehenden Listing wird deutlich, dass es sich um eine Endlosschleife handelt. In der Schleife wird der Opcode gelesen, der Opcode wird übersetzt und mit einem *switch/case*-Konstrukt sofort ausgeführt. Wenn das erfolgreich geschehen ist, wird der Programmzähler um den Wert 2 erhöht. Ein nicht erfolgreicher Verlauf

Spiele im EEPROM

Weil dies ist kein Gameboy ist, darf man keine Spiele nach aktuellem Stand der Spiele-Kultur erwarten. Ein typisches Chip-8-Spiel ist „Pong“: Dort muss der Spieler auf dem Display einem virtuellen Tennisball folgen, was nicht wesentlich leichter als auf einem realen Tennis-Parcours ist.

Das EEPROM ist in 4 KB große Blöcke unterteilt, dies entspricht dem Arbeitsspeicher von Chip-8 mit einem Spiel in jedem Block. Die maximale Größe eines Chip-8-Spiels beträgt 3584 Byte. Übrig bleiben 512 Byte für einen Header, davon sind 64 Byte belegt: 2 Byte für eine Pseudo-Checksumme, 1 Byte für die Spielgeschwindigkeit, 16 Byte für die Taster-Konfiguration, die übrigen Bytes für den Titel des Spiels. In das EEPROM mit 32 KB passen acht Spiele. Die meisten Chip-8-Spiele haben jedoch nur einen Umfang von 256 Byte (1 KB für S-Chip). Im EEPROM ist folglich noch Platz vorhanden.

Nach der Initialisierung liest der PSoC die Header im EEPROM und gibt die Titel auf dem Display aus. Wenn der Spieler sein Spiel gewählt hat, werden das Spiel und die zugehörige Taster-Konfiguration geladen, zusammen mit den Parametern für die Spielgeschwindigkeit.

ist ein Ereignis, das vorrangig behandelt werden muss.

Der Interpreter hat außerdem noch einige Nebenaufgaben: Abfragen der Taster, Fehlerbehandlung (zum Beispiel *Stack Overflow* oder Fehler beim Speicherzugriff). Dazu kommen allgemeine Aktivitäten, wie das Abarbeiten von Verzögerungsschleifen, damit das Spiel nicht zu schnell läuft. Dass der Programmzähler um 2, nicht um 1 erhöht wird, liegt an den 16 bit breiten Opcodes. Die Opcodes befinden sich in einer Tabelle, die nur 8 bit breit ist.

Nach diesen Vorüberlegungen gestaltet sich der Interpreter überraschend einfach:

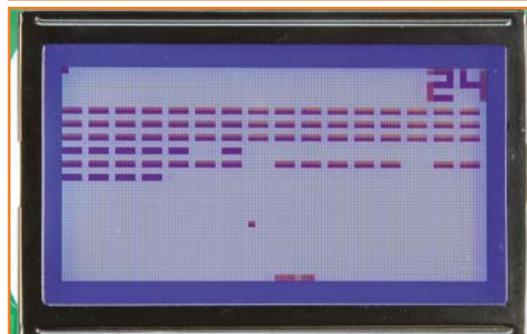
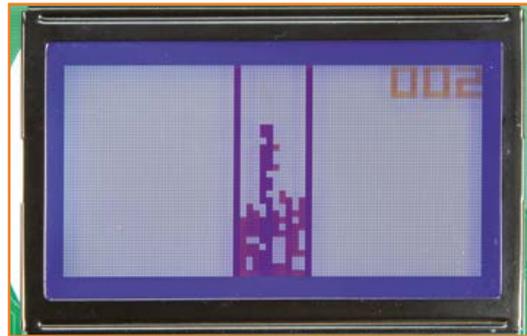
```
while(1)
{
    //Clear the flags
    OS_FLAGS = 0;
    ERROR = ERROR_NO_ERROR;
    input_read(&buttons);
    //fetch the opcode (16 bits)
    OPCODE = ((READ_PROGRAM_MEMORY(PC) <<
8) | READ_PROGRAM_MEMORY(PC+1));

    switch((OPCODE >> 12) & 0xF) //we
decode the opcode by groups of 4 bits
    {
        /*OPCODE DECODING AND EXECUTION*/
    }
    if(ERROR)
    {
        /*ERROR HANDLING*/
    }

    if(!((OS_FLAGS >>
PC_NOT_INCREMENT_FLAG) & 1))
    {
        PC += 2; // Increment the PC except
if told not to
    }
}
```

Die 35 Opcodes gehören einer der folgenden Kategorien an:

- *Mathematische Operatoren* wie ADD und SUB, sie werden von den meisten Mikrocontrollern unterstützt.
- *Sprung-Befehle* wie JMP und CALL bedürfen besonderer Aufmerksamkeit, denn sie



beziehen sich natürlich auf die virtuelle Speicher-Architektur.

- *Speicher-Kommandos* sind bei Mikrocontrollern mit genügend Speicher unkritisch, doch wenn der Speicherplatz wie bei der Chip-8-Architektur knapp ist, muss der eine oder andere Kunstgriff angewandt werden.
- *Hardware-Kommandos* sind für die Hardware spezifisch, sie unterscheiden sich abhängig von der Hardware-Konfiguration.

Auch zu diesem Thema enthält die Online-Dokumentation weiterführende Informationen, ebenso wie zu den Opcodes der Timer, den Tastern und den Zufallszahlen. An dieser Stelle wollen wir kurz die grafischen Operatoren von Chip-8 betrachten. Gehen wir davon aus, dass das Standard-Format eines grafischen Displays eine Matrix mit 64 x 32 Pixeln ist. Unser Display besitzt 128 x 64 Pixel, so dass ein Chip-8-Pixel aus vier realen

Pixeln besteht. Pixel 0,0 ist das Pixel in der Ecke links oben. Grafische Zeichen lassen sich nur mit einigem Aufwand auf dem Display bewegen, insbesondere weil sich die Logik des Displays von der Chip-8-Architektur elementar unterscheidet. Chip-8 schreibt *Zeilen* mit 8 Pixeln auf das Display, während das Display *Spalten* mit 8 Pixeln auf die Bildfläche setzt. Chip-8 schreibt Pixel bezogen auf eine vorgegebene Koordinate in Gruppen mit fester Breite (8 bit) und variabler Höhe. Unser Display mit 128 x 64 Pixeln besteht aus zwei Bildflächen zu 64 x 64 Pixeln, die in 8 Seiten mit 64 Spalten unterteilt sind (**Bild 3**). Deshalb müssen wir das Display in Spalten mit 8 Pixeln beschreiben, in eins der 2 x 8 x 64 Felder. Für diese Gruppe von Pixeln bedeutet dies, dass eine feste Höhe vorgegeben ist. Die Koordinaten liegen ebenfalls fest (Zeilen 0, 8, 16, 24, 32, 40, 48 und 56), und jede Schreiboperation hat Einfluss auf benachbarte Pixel. Augenscheinlich sind hier

Bild 4.
Die doppelseitige Platine kann auch einseitig ausgeführt werden.

Stückliste

Widerstände:

R1 = 270 Ω, 250 mW, 5 %
R2 = 820 Ω, 250 mW, 5 %
R3,R4 = 2k7, 250 mW, 5 %
R5 = 1 k, 250 mW, 5 %
R6 = 100 Ω, 250 mW, 5 %
R7 = Drahtbrücke (siehe Text)
P1 = 47 k, Trimpoti

C2,C3 = 100 n, 50 V, 20 %

Halbleiter:

LED1 = LED 3 mm, rot
IC1 = PSoc CY8C29466, EPS 130469-41
IC2 = EEPROM 24LC256
IC3 = LM317

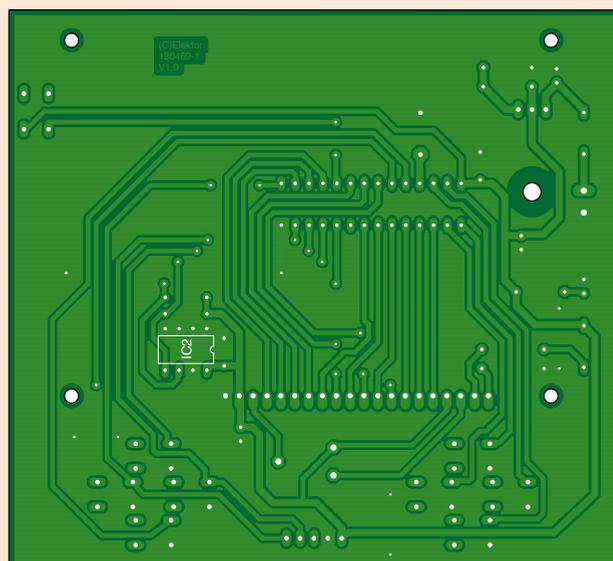
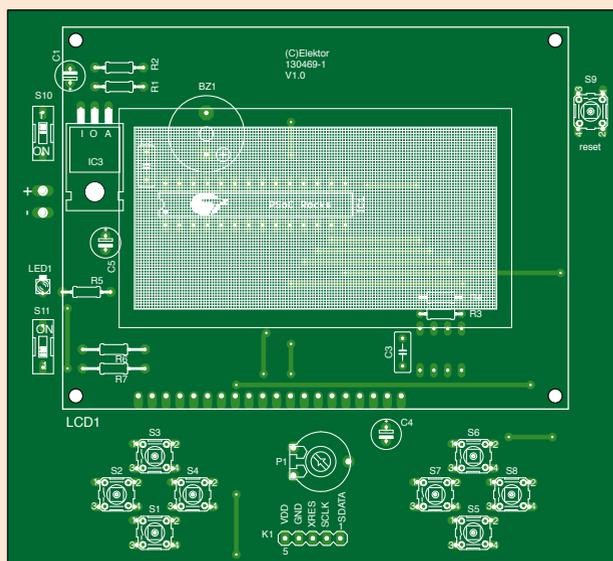
S10, S11 = Umschalter 1-polig
DIP-Fassung 8-polig für IC2, siehe Text
LCD1 = LCD-Display, grafisch 128 x 64 Pixel, z. B. Vatronix TG12864B-03 oder MCD128064A6W, siehe Text
Bz1 = Beeper 5 V

Kondensatoren:

C1,C4,C5 = 10 μ, 25 V stehend

Außerdem:

S1...S9 = Microswitch-Taster für Platinenmontage, 24 V, 50 mA, 6 · 6 mm



einige Kunstgriffe und Verrenkungen nötig, um dies alles zu bewerkstelligen. In der Online-Dokumentation werden auch diese Feinheiten erklärt.

Der Aufbau

Nachdem wir uns in der virtuellen Welt bewegt haben, wenden wir uns jetzt den konkreten Gegebenheiten zu. Die Platine (**Bild 4**) ist zweiseitig, doch wenn einige Drahtbrücken in Kauf genommen werden, ist auch eine einseitige Variante machbar. Zu montieren sind ausschließlich bedrahtete Komponenten. Das LC-Display mit 128 x 64 Pixeln und dem Chipset KS0108 ist ein gängiger Typ, die Belegung der Anschlüsse kann jedoch unterschiedlich sein. Wir hatten das LC-Display Vatronix TG12864B-02 geordert, doch geliefert wurde der Typ TG12864B-03. Bei diesem Typ sind die Anschlüsse der Anode und Kathode für die Hintergrundbeleuchtung (Pin 19 und 20) miteinander vertauscht. Der 0-Ω-Widerstand R7 erleichtert das Anpassen an die verfügbare Version. Wenn erforderlich werden R6 und die Drahtbrücke R7 gekreuzt, so dass Anschluss 19 an Masse und Anschluss 20 an der Betriebsspannung liegen. Der Distributor Farnell bietet das Display MC128064A6W des Herstellers Midas an. Auch auf ebay gibt es sicher das eine oder andere günstige Angebot. Die Hintergrundbeleuchtung ist nicht obligatorisch, dieser Spiel-Komfort belastet die Energiequelle zusätzlich mit 20 mA.

Wie die Fotos zeigen, wird das EEPROM auf der Lötseite der Platine montiert. Das hat den Vorteil, dass das EEPROM zum Ändern des Inhalts bequem herausgenommen werden kann. Allerdings muss dafür eine Fassung konstruiert werden, wie sie **Bild 5** zeigt.

Der Spannungsregler IC3 darf nicht auf der Platine aufliegen, denn an der metallischen Fahne des LM317 liegen +5 V. Bei der doppelseitigen Platine, die das Foto zeigt, muss zwischen Spannungsregler-Gehäuse und Kupferfläche ein Abstand eingehalten werden. Einer isolierten Montage wie in **Bild 6** steht natürlich nichts entgegen. Der PSoC ist über ein PSoC MiniProg-Interface an K1 und den PSoC Designer programmierbar. Unsere Version des PSoC Designers war die Version 5.4. Während der Programmierung erhält das System Betriebsspannung über das Interface, die eigene Stromversorgung wird nicht benötigt.

Firmware

Elektor stellt den Quellcode zum Programmieren von Spielen in das EEPROM sowie den Code

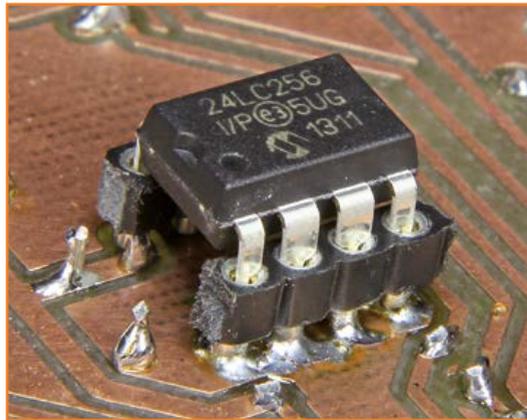


Bild 5. Das EEPROM wird in eine Fassung aus vierpoligen Buchsenleisten gesetzt, die auf der Lötseite montiert sind.



Bild 6. Auf der Bestückungsseite muss zwischen Spannungsregler und Kupferfläche ein Zwischenraum eingehalten werden.

für den Chip-8-Emulator zur Verfügung, nicht jedoch die Spiele! Der programmierte PSoC, der im Elektor-Shop erworben werden kann, enthält nur die Firmware. Spiele für Chip-8 sind jedoch im Internet zahlreich vertreten [3]. Manche Spiele sind kostenlose Public Domain, andere Spiele kosten einen kleinen Obolus. Das Urheberrecht an Spielen ist auch nach 40 Jahren noch nicht erloschen. Zum Programmieren des Flash-Speichers im PSoC muss der MiniProg Programmer von Cypress an K1 angeschlossen werden.



Bild 7. Liste der Spiele, die der PSoC hier im EEPROM vorfindet.

Ein Wort des Autors

Nach meinem Ingenieurs-Studium an der INSA Lyon (Frankreich) und einem Jahr an der Universität Tokio (Japan) bin ich im Rahmen eines Austauschprogramms bei Ichikoh Industries beschäftigt, ein Hersteller für Hightech-Systeme der Kfz-Beleuchtung. Schwerpunkte meiner Tätigkeiten in Tokio und Paris waren die Elektronik, die Bildverarbeitung und die Software-Entwicklung.

Der Chip-8 Video Game Emulator beruht auf meinen Ideen. Das Projekt habe ich in meiner Freizeit entwickelt, erste Anstöße gab die Hardware, die ich in Akihabara vorfand. Akihabara ist ein gigantischer Elektronik-Marktplatz in Tokio, geradezu ein Pilgerort für Elektor-Leser. Mein Dank gilt Matthieu Denoual, Yoshio Mita sowie Luc Lemmens für die tatkräftige Unterstützung.



Für das Laden eines Spiels gibt es drei Optionen:

- Das Spiel wird mit einem geeigneten EEPROM-Programmer in das EEPROM geschrieben.
- Das Spiel wird mit dem Tool [9] geladen, das wir für diesen Zweck geschrieben haben. Anschließend kann das EEPROM mit dem Emulator-Board *geflasht* werden.
- Das Spiel wird unmittelbar in den Flash-Speicher des PSoC geladen. In diesem Fall müssen die Default-Werte einiger Variablen der Interpreters angepasst werden. Der Code muss in der Tabelle PROGRAM_MEMORY stehen, direkt hinter den Fonts (global.c).

Zum Schluss seien noch die drei Situationen genannt, die der Interpreter abhandeln kann:

- EEPROM nicht vorhanden: Das Spiel im Speicher des PSoC wird gestartet, es muss zuvor aus dem EEPROM oder mit dem PSoC-Programmer geladen sein.
- EEPROM ohne Header vorhanden: Das Spiel im EEPROM wird mit den Default-Einstellungen gestartet.
- EEPROM mit Headern vorhanden: Das Menü erscheint auf dem Display, das ausgewählte Spiel wird gestartet.

(130469)gd

Weblinks

- [1] **Allgemeine Beschreibung Chip-8:** <http://de.wikipedia.org/wiki/CHIP-8>
- [2] **Dokumentation Chip-8-Emulator** (in Englisch): www.elektor-magazine.de/130469
- [3] **Programm-Code der Spiele mit Dokumentation von David Winter:** www.pong-story.com/chip8/
- [4] **Befehlssatz Chip-8:** <http://devernay.free.fr/hacks/chip8/C8TECH10.HTM>
- [5] **Informationen zu Super-Chip und Beispiel eines Interpreters** (in Französisch): <http://blogs.wefrag.com/mrhelmut/2012/01/14/recette-creer-son-premier-emulateur/>
- [6] **Bibliotheken (Fonts) zum Programmieren von LC-Displays:** <http://en.radzio.dxp.pl>
- [7] **Wandler-Utility zum Umsetzen von Game-Code in Hex-Arrays:** www.fourmilab.ch/xd
- [8] **Der PSoC im Elektor-Shop:** www.elektor.de/psoc_chip8
- [9] **Anleitung zum Programmieren von EEPROMs:** www.elektor-magazine.de/130469 (EEPROM tutorial.pdf)

Professionelle Hard- & Software zum Sonderpreis!

Exklusiv für Studenten!

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

LabVIEW

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



Circuit Design Suite

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



myDAQ

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



Studentenversionen:

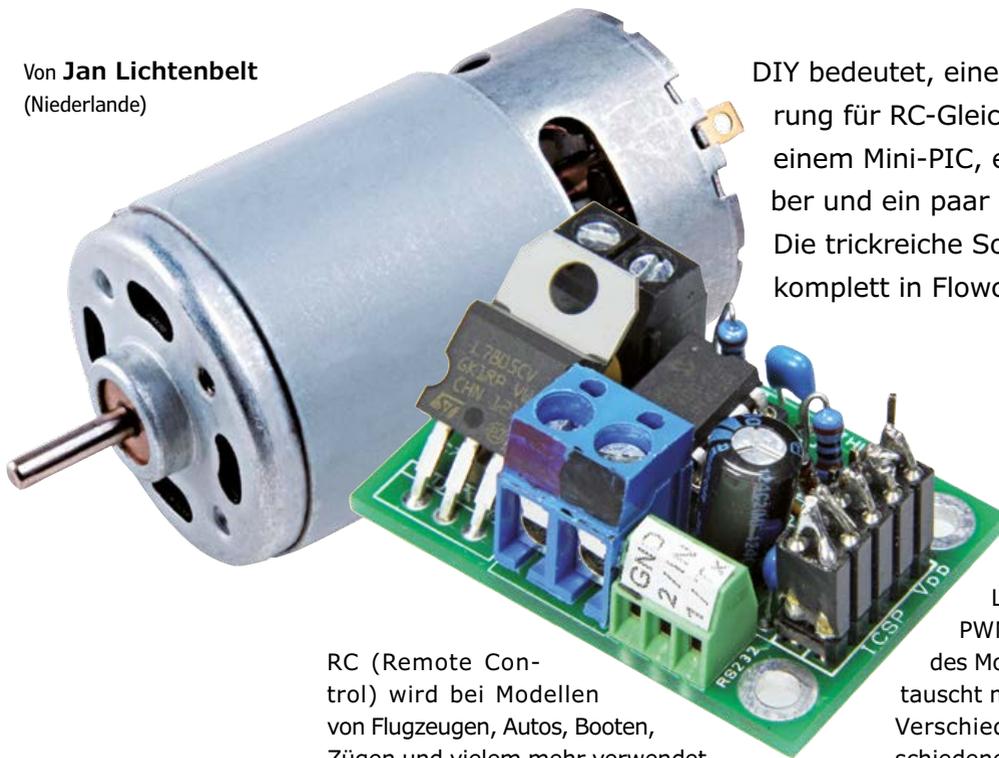
- Der Preis von **LabVIEW** und **Circuit Design Suite** beträgt jeweils nur 23,95 Euro.
- Das **myDAQ Education-Kit** bestehend aus *myDAQ* + 3 Software-DVDs (*LabVIEW*, *Circuit Design Suite* und *DIAdem*) kostet nur 174,95 Euro.

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

RC Speed Control für Gleichstrommotoren

Klein und vielseitig

Von **Jan Lichtenbelt**
(Niederlande)



RC (Remote Control) wird bei Modellen von Flugzeugen, Autos, Booten, Zügen und vielem mehr verwendet. Das Prinzip ist einfach: Ein RC-Sender sendet ein Signal zu einem RC-Empfänger im Modell. Die Position des oder der Steuerknüppel der Fernbedienung bestimmt die Länge eines Impulses von 1 ms bis 2 ms Länge mit einer Wiederholrate von 50 Hz. Während dieser Impuls direkt zur Steuerung von Servomotoren verwendet werden kann, muss er für die Drehzahleinstellung von DC-Motoren in Vorwärts- (Flugzeuge und Rennwagen-Modelle) oder Vorwärts/Rückwärts-Bewegung (Modellbahn, Schiff) „übersetzt“ werden. Es gibt kommerzielle RC-Empfänger mit pulsweitenmodulierten Ausgangssignalen (PWM), um die Drehzahl eines Gleichstromelektromotors zu steuern. Dieser Artikel beschreibt, wie Sie eine solche PWM-Drehzahlregelung auf einer bemerkenswert kleinen Platine realisieren können. Ein Mikrocontroller kann das RC-Tastverhältnis (5...10 %) in ein PWM-Signal von 0...100 % sowohl in Vorwärts- als auch in Rückwärtsrichtung übersetzen. Dazu sind zwei PWM-Signale

DIY bedeutet, eine PWM-basierte Drehzahlsteuerung für RC-Gleichstrommotoren mit nichts als einem Mini-PIC, einem integrierten MOSFET-Treiber und ein paar diskreten Bauteilen aufzubauen. Die trickreiche Software für das Projekt wurde komplett in Flowcode entwickelt.

und eine MOSFET-Vollbrücke (wie in **Bild 1** zu sehen) erforderlich. Wenn der eine Brückeneingang auf Low gehalten wird und der andere das PWM-Signal empfängt, wird die Drehzahl des Motors in Vorwärtsrichtung geregelt; vertauscht man die Signale, in Rückwärtsrichtung. Verschiedene Arten von Modellen haben verschiedene Anforderungen: Flugzeuge und Race Cars müssen sehr schnell reagieren, während Modellbahnen und Schiffe nur langsam natürlich wirken. Fällt der Empfänger aus, soll die Modellbahn weiter fahren, das Rennauto aber so schnell wie möglich stoppen. Und was soll nach einem Überlaststrom oder einer Übertemperatur des Brückentreibers oder des Motors passieren?

Zunächst sollte die Schaltung des Brückentreibers ausschließlich aus bedrahteten und diskreten Bauteilen bestehen, mit der Ausnahme eines kleinen Brückentreiber-ICs, das aber leicht von Hand gelötet werden konnte. Die erste Version der Schaltung mit vier MOSFETs funktionierte aber nicht zuverlässig. So wurde das Brückentreiber-IC ADP3624ARDZ gewählt, das einen Spitzenstrom von 4 A liefern kann. Dies ist völlig ausreichend für die direkte Ansteuerung eines Modellbahnmotors (Spur I) mit typischen Stromanforderungen von 0,3...1 A. Dieser Brückentreiber arbeitet an einer Versorgungsspannung von 4,5...18 V,

besitzt zwei nichtinvertierende Eingänge und verfügt über eine Übertemperatur-Warnung. Das IC steckt in einem SOIC-Gehäuse mit einem Pinabstand von 0,05" (1,27 mm), was gerade noch weit genug ist, um die Pins mit ruhiger Hand zu verlöten und eventuelle Lötbrücken nachher mit Entlötlitze zu entfernen.

Hardware

Bild 2 zeigt den einfachen Aufbau der Schaltung. Der Microchip-Mikrocontroller PIC12F1840 empfängt die 1...2 ms langen Impulse des RC-Empfängers und übersetzt sie in PWM-Signale. Diese Signale werden zum MOSFET-Brückentreiber geführt, an dessen Ausgängen der Motor direkt angeschlossen ist. Widerstand R6 ist ein „Dummy“, der sicherstellen soll, dass die Schaltung auch funktioniert, wenn der Motor nicht angeschlossen ist. C1 muss so nah wie möglich am Treiber-IC, C3 ebenso nah am Motor angebracht sein. Um die MOSFETs als Schalter verwenden zu können, muss die Gatespannung mindestens 4,5 V betragen. Da bei Stromspitzen (beim Umschalten) die Spannung leicht unter diesen Wert fallen kann, sorgt ein „dicker“ Keramikkon-

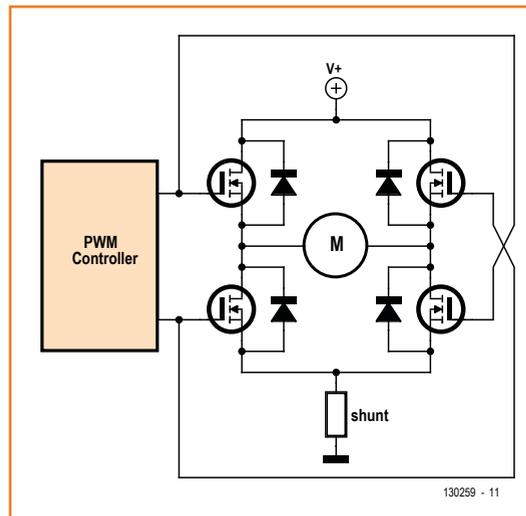


Bild 1. Die Vollbrücke steuert die Motorgeschwindigkeit und Drehrichtung. Der Shunt ist optional.

densator (C7) für einen Energiepuffer. Auch dieser Kondensator sollte sich so nah wie möglich am IC befinden. Der handelsübliche 10- μ F-Kondensator im SMD-Format ist auch für Heimwerker sehr leicht lötlbar.

Der Brückentreiber IC2 gibt eine Temperaturwarnung an Pin 8 (high aktiv) aus. Dieses Sig-

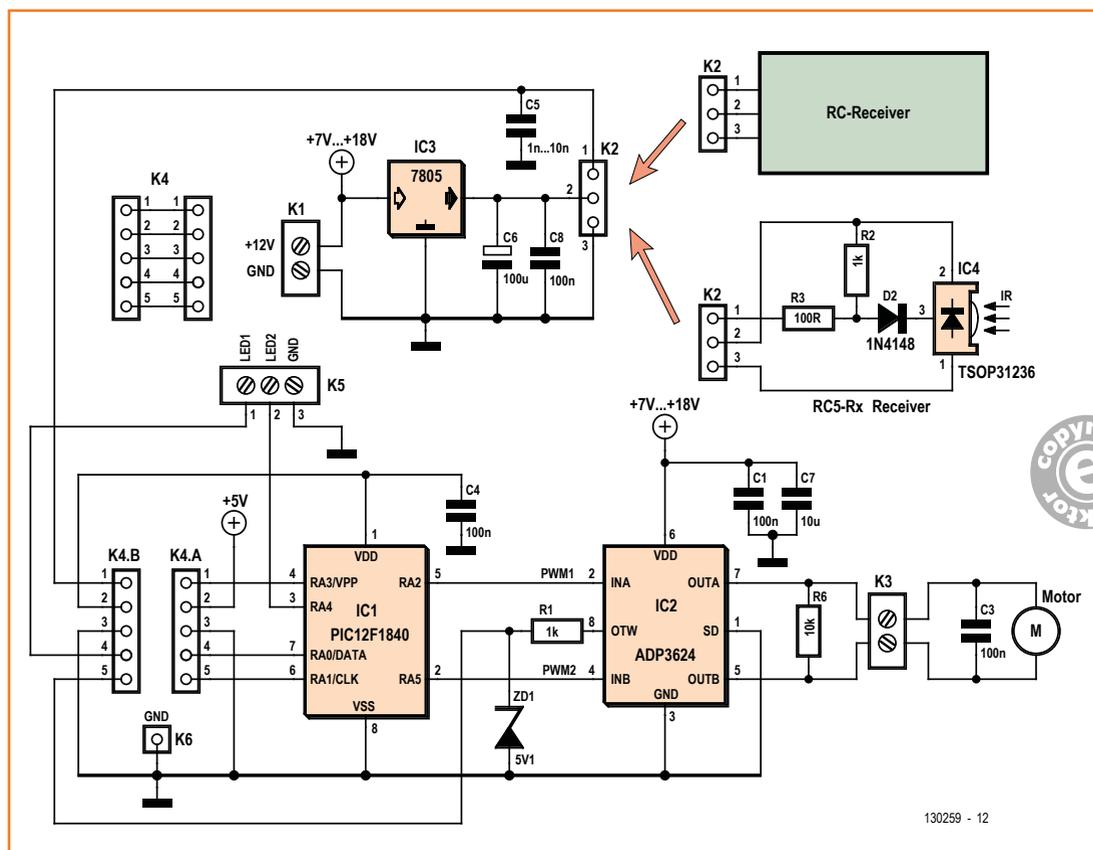


Bild 2. Der Schaltplan der RC Speed Control.

nal löst im PIC eine Interrupt-Prozedur aus. R1 und die Zenerdiode D1 schützen den PIC vor eventuellen Überspannungen. Der Shutdown-Anschluss SD von IC2 wird nicht verwendet, weil die Abschaltung von der Firmware des PICs vorgenommen wird.

Wenn der RC-Sender in der Nähe der Platine platziert wird, können Hochfrequenzstörungen (circa 40 MHz beim Autor) zum Eingang des Mikrocontrollers gelangen. Ein Kondensator von 1...10 nF (C5) am Eingang des RC-Signals (K2a) verringert diese Störungen.

Die Spannungsversorgung (+7...18 VDC) ist unkompliziert. Ein Spannungsregler IC3 (L7805) und die Kondensatoren C6 (10 µF) und C8 (100 nF) sorgen für eine saubere +5-V-Versorgung. C4 ist ein HF-Entkopplungskondensator für den Mikrocontroller. Der Regler mit seinem maximalen Strom von 1 A kann auch den RC-Empfänger mit +5 V versorgen und ebenso zusätzliche Servos, die an diesem Empfänger angeschlossen sind. Geht das Signal des RC-Senders verloren, neigen bestimmte RC-Empfänger aber zu unkontrollierten Impulsen. Einige Servos quittieren dies mit Oszillationen und erzeugen starke Stromimpulse, so dass eventuell die Kontrolle durch den Mikrocontroller verloren geht. Wenn dies geschieht, trennen Sie die Servos von der Schaltung und prüfen Sie, ob die Geschwindigkeitsregelung ordnungsgemäß funktioniert. Versuchen Sie auch einen anderen Funkkanal, um die Funktion zu verbessern. Schließen Sie die Servos wieder an und testen Sie erneut. Wenn die Probleme weiterhin bestehen, können ein paar zusätzliche Kondensatoren helfen oder ein eigener Spannungsregler für die Servos eingesetzt werden. Um Überspannungen zu vermeiden, wenn RC-Signale verloren gehen, sei dringend empfohlen, den RC-Sender **vor** dem Mikrocontroller einzuschalten und nicht nur den RC-Sender, sondern immer auch den Mikrocontroller auszuschalten. Verwenden Sie stets den besten Funkkanal, der zur Verfügung steht.

Der Mikrocontroller PIC 18F1840 verwendet seinen internen 16-MHz-Oszillator. Das Master-Clear-Signal gibt der Controller sich selbst. Ein 2-µs-Timer wird verwendet, um die RC-Impulse von 1...2 ms Länge abzutasten. Die Präzision ist eigentlich höher als für ein Byte (8 bit) nötig ist, mit dem der PWM-Ausgang gesteuert wird. Wesentliche Daten sind dauerhaft im EEPROM gespeichert und verfügbar, wenn sie vom Programm benötigt werden. Die Flanken der Impulse

des RC-Signals an Pin 4 (Port A3) von IC1 bewirken Interrupts. Obwohl der RC-Empfänger mit 5 V betrieben wird, erkennt der TTL-kompatible Eingang A3 des PICs auch Impulse von nur etwa 3 V. Das Übertemperatursignal des Brückentreibers ist an Pin 6 (Port A1) angeschlossen und bildet einen Teil eines Komparators mit der PIC-internen Referenzspannung von 2,5 V. Die PWM-Frequenz von etwa 40 kHz wird durch den Timer2 erzeugt, was innerhalb der Spezifikationen des Brückentreibers (20...100 kHz) liegt.

Haben Sie sich schon über K4 gewundert? Im Normalbetrieb sind die rechte und linke Seite verbunden und leiten alle Signale zum Controller, aber die rechte Seite K4a stellt auch eine ICSP-Schnittstelle (In-Circuit Serial Programming) für die drei PIC-Pins data, clock und Vpp (13,8 V) dar. Zusätzlich werden +5 V und Masse des ICSP-Interfaces zum Controller geführt.

Anstelle der RC-Impulse können auch IR-Impulse einer RC5-Fernbedienung zu diesem Portpin geführt werden. Dies ist erforderlich, um die Eingabeparameter des Mikrocontrollers zu ändern. Ein 36-kHz-IR-Empfänger TSOP31236 erledigt das komplette Signalhandling. Am Ausgang des TSOP ist der Pullup-Widerstand R2 angeschlossen. Über Diode D2 (1N4148) kann der TSOP den Ausgang auf Low ziehen. R3 (100 Ω) schützt den Eingang des Mikrocontrollers.

An K5 können zwei LEDs angeschlossen werden, um die Drehrichtung anzuzeigen. Der maximale Ausgangsstrom des PICs von jeweils 25 mA ist dafür ausreichend. Die LEDs spielen auch eine andere Rolle: Wenn der Empfang abbricht, flackern die LEDs, bei Übertemperatur verlöschen beide. Die LED1 an Pin 7 signalisiert auch eine RC5-Kommunikation mit einem anderen Mikrocontroller, der auf einem LCD die eingestellten Startparameter und verschiedene Betriebsdaten anzeigt. Verwenden Sie einen PIC16F628A oder den gleichwertigen PIC16F1847 (wie in Elektor beschrieben [1]), ohne den HF-Sender. Die Software für den RC5-Rx-Controller Microchip PIC16F1847 ist Teil dieses Artikels.

Um das RC5-Tx-Signal an Pin 5a-1 zu aktivieren, muss Pin K5a-2 beim Einschalten über einen 1-kΩ-Widerstand auf High gelegt werden sein. Im Normalbetrieb liegt dieser Anschluss über einen Strombegrenzungswiderstand auf GND.

Software

Die Software eignet sich für die Geschwindigkeitssteuerung, passend für viele Anwendungen.

Die beiden möglichen Startprozeduren dauern nur ein paar Sekunden:

- Stellen Sie den Steuerknüppel auf Maximum. Die Vorwärts-LED leuchtet auf. Sobald sie verlischt, legen Sie den Steuerknüppel in die Minimalstellung. Die Rückwärts-LED wird hell. Wenn sie verlischt, stellen Sie den Steuerknüppel in die neutrale Position zurück. Dies kann eine beliebige Position, Mittelstellung oder Mindeststellung sein. Es ist wichtig, dass Sie beide LEDs gleichzeitig beobachten können. Minimal- und Maximalposition werden im EEPROM gespeichert.
- Stellen Sie den Steuerknüppel auf Neutral. Schalten Sie den Strom für den Mikrocontroller ein und lassen Sie den Steuerknüppel für mindestens 3 s in dieser Stellung. Die minimalen und maximalen Einstellungen werden aus dem EEPROM gelesen.

Die Mittelposition kann auch ein wenig über dem Minimum liegen; dies erlaubt bei Race Cars das Bremsen mit rückwärts drehenden Reifen. Testen Sie aber bei der ersten Inbetriebnahme, ob die Drehrichtung stimmt.

Normalerweise behält der Motor seine letzte Einstellung bei, wenn der RC-Empfänger aus der Reichweite des Senders gerät. Über die unten beschriebene RC5-Kommunikation kann dieses Verhalten geändert werden: Der Motor stoppt, sobald das Signal verloren geht und beide LEDs blinken.

Nur wenn die Mittel- der Minimalstellung entspricht, kann der Motor nur in der „Vorwärts“-Richtung drehen. Dies ist bei Race Cars und Flugmodellen sinnvoll. In diesem Fall stoppt das Programm die PWM, wenn das Auto oder Flugzeug für mehr als 140 ms (sieben RC-Impulse) außerhalb der Reichweite des Senders gerät. Es wird dringend empfohlen, dieses Verhalten regelmäßig zu testen.

Ein PWM-Signal von 0 % von 500 ms von Dauer verhindert (zu) hohe Ströme aufgrund einer plötzlichen Änderung der Drehrichtung. Jeder Motor benötigt zum Anlaufen einen gewissen PWM-Prozentsatz. Das Programm verwendet dazu einen PWM-Mindestwert von 30 %. Die Übertemperaturwarnung des Brückentreibers stoppt den Motor und lässt dank der eingebauten Hysterese des ICs ein Wiederanfahren erst nach dem Abkühlen zu. Beide LEDs bleiben dunkel.

Es wurde nötig, das RC-Signal in der Software

Stückliste

Widerstände:

R1 = 1 k
R2 = 1 k (für IR-Receiver)
R3 = 100 Ω (für IR-Receiver)
R6 = 10 k

Kondensatoren:

C1, C4, C8 = 100 n
C3 = 100 n (auf den Motoranschlüssen)
C5 = 1...10 n
C6 = 100 μ /25 V
C7 = 10 μ /25 V, keramisch, SMD (z.B. AVX-18123D106KAT2A bei Farnell 1327761)

Halbleiter:

ZD1 = Z-Diode 5,1 V
D2 = 1N4148 (für IR-Receiver)
IC1 = PIC12F1840
IC2 = ADP3624ARDZ oder ADP3624ARDZ-RL (SOIC)
IC3 = L7805CV
TSOP31236 (36-kHz-Version), angeschlossen an K2

Außerdem:

K1, K3 = 2-polige Schraubklemme, RM 5 mm
K2 = 3-polige Stiftleiste, RM 0,1"
K4 = 2x5-polige Stiftleiste, RM 0,1"
K5 = 3-polige Schraubklemme, RM 5 mm
K6 = 1-polige Stiftleiste (Lötnagel)
Platine 130259-1

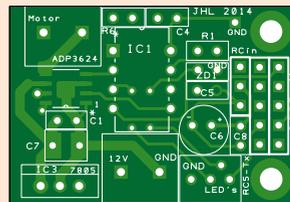


Bild 3.

Das nur 38,1 x 26,4 mm² große Platinenlayout mit dem ADP3624 Brückentreiber, der sich um das Signalhandling zwischen PIC und Motor kümmert.

etwas zu filtern (besonders wenn sich mehrere Sender in der Umgebung befanden). Es wird ein gleitender Durchschnittswert mit einer RC-Zeit (50 %) von 60 ms (drei RC-Impulse) verwendet. Insgesamt zwölf RC-Impulse fließen in den gleitenden Durchschnitt ein, also etwa $12 \times 20 = 240$ ms. Dies alles resultiert in einer geringen Filtersteilheit in Verbindung mit einer schnellen Reaktion, wie es insbesondere für Flugmodell-Anwendungen notwendig ist. Einzelheiten, Updates und die Flowcode-Quellcode-Dateien finden Sie unter [2].

Kommunikation

Das Ändern der Parameter erfordert eine Kommunikation mit dem Mikrocontroller. Auf eine RS232-Schnittstelle wurde aufgrund der schlechten Synchronisation verzichtet, da bei Verwen-

derung einer RS232-Rx-Leitung nur ein Start-Bit zur Verfügung steht. Stattdessen kommt eine IR-Kommunikation mit RC5-Protokoll zum Einsatz, das die Hälfte jedes Bits zur Synchronisation verwendet. Fernbedienungen mit RC5-Protokoll lassen sich in den meisten Haushalten finden. Der Gerätecode spielt keine Rolle, nur die sechs Tastenbefehle werden ausgewertet [4].

Die RC5-Rx-Signale werden dem PIC über den gleichen Anschluss zugeführt, wie ihn auch die RC-Impulse benutzen. Entfernen Sie die RC-Empfänger und schließen Sie stattdessen den IR-Empfänger an. Der RC5-Eingang wird aufgrund der Tatsache erkannt, dass er auf High gezogen wird, während der Eingang bei RC-Impulsen standardmäßig Low ist. Die Software ist sehr flexibel und erfasst nicht nur die „offiziellen“ RC5-Bits mit einer Spanne von $2 \times 889 \mu\text{s}$, sondern auch viel schnellere Impulse.

Die RC5-Tx-Ausgabe ist am LED-Anschluss (Port A0, Pin 7) verfügbar, wenn der andere LED-Anschluss Port A4 (Pin 3) des PICs über einen $1\text{-k}\Omega$ -Widerstand auf High gelegt wird. +5 V ist zum Beispiel an Pin 2 von K4a/b erhältlich. RC5-Tx unterscheidet sich vom offiziellen Protokoll:

- Das Protokoll ist RC6 sehr nahe, es erlaubt die Übertragung von Bytes (8 bit) oder Integers (16 bit). RC6 bedeutet: 2 Start-Bits, 1 Byte und 3 Zahlen.
- Wir verwenden eine Bit-Periodendauer von $2 \times 255 \mu\text{s}$, die maximale Frequenz, die der unter [1] beschriebene HF-Sender verwenden kann. Jede Übertragung mit $2+3+8 \times 16 = 58$ bit dauert somit

$$58 \times 0,5 = 29 \text{ ms.}$$

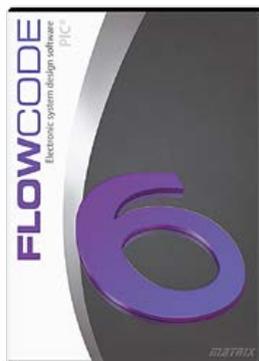
Dies erfordert eine spezielle RC5-Empfangssoftware für den PIC 16F1839 mit LCD, die im freien Download zu diesem Artikel gefunden werden kann [4].

Die Platine

Alle Komponenten (bedrahtet und SMD) werden auf der Oberseite der kleinen Platine (**Bild 3**) angebracht. Löten Sie den Brückentreiber zunächst an einem Pin fest, dann an den anderen und entfernen das überschüssige Lötzinn mit Entlötlitze. Pin 1 des ADP3624 wird vom grauen Punkt auf dem Gehäuse markiert. An der Unterseite des ICs gibt es eine Fläche für den thermischen Kontakt zur Platine. An dieser Stelle müssen Sie den Schutzlack auf der Platine entfernen! Als nächstes wird der $10\text{-}\mu\text{F}$ -SMD-Kondensator C7 angebracht. Pin 1 des PIC-Mikrocontrollers ist links oben, auf der rechten Seite des PICs befinden sich die Bauteile von oben nach unten: R1, Z-Diode D1 (GND rechts), C5 und C6. Rechts daneben sind K3 (3-poliger-RC-Eingang) und C8 zu finden. Auf der rechten Seite der Platine liegt der 2×5 -polige ICSP-Anschluss K4. Der LED-Anschluss K5 hat drei zusätzliche Anschlussmöglichkeiten am Platinenrand. Der Masseanschluss K6 auf der oberen rechten Seite ist optional, aber praktisch bei der Überprüfung der bestückten Platine.

Schauen Sie sich einmal das Video auf YouTube [3] an, wo die Schaltung getestet wird, allerdings noch mit RS232-Kommunikation und dem EB006-Board für die ICSP-Programmierung.

(130259)



Die Firmware für dieses Projekt wurde vollständig im Flowcode 6 für PIC entwickelt (erhältlich bei Elektor, www.elektor.de).

Weblinks

[1] Modulare Funkverbindung mit Manchester-Code, Elektor September/Okttober 2013:

www.elektor-magazine.de/120049 und www.elektor-magazine.de/120187

Hinweis: PIC16F628A wurde durch einen äquivalenten PIC16F1847 ersetzt. Programmierung der Mikrocontroller mit Vpp an Pin A5, Vdd, GND, Clock-Pin B6 und Data-Pin B7. Pin B0 ist der RC5/RC6-Eingang (Rx) oder Ausgang (Tx). Die LCD-Daten DB4...DB7 sind mit den IC-Pins A0...A3 verbunden, RS mit B4 und E mit B5. Normalerweise verbindet ein Widerstand von $1\text{ k}\Omega$ VDD und MCRL.

[2] Jan Lichtenbelt: Remote Speed Control of DC Motors, Matrix Multimedia Artikel MX040:

www.matrixmultimedia.com/article.php?a=677 und Updates www.matrixmultimedia.com/mmforums/viewtopic.php?f=36&t=11613

[3] Erster Test der Platine auf YouTube (Version mit RS232, nun von RC5 ersetzt):

<http://youtu.be/TSJCCADYKsE>

[4] Zusätzliche Dokumentation und Software: www.elektor-magazine.de/130259

Tunnel-Dioden

Seltsame Bauteile (7)

Mein Labor ist immer eine Quelle der Inspiration. Neulich fiel mir mein Tektronix-Oszilloskop ins Auge. Ich habe bei diesen älteren Geräten immer bewundert, dass sie solch exzellente Leistungen mit diskreten Komponenten erreichen konnten. Etwas, was mich stets fasziniert hat, waren die Tunnelioden in einigen der Trigger-Schaltungen. Diese Bauteile sind heute nicht mehr üblich, auch wenn man sie (nach harter Suche) noch käuflich erstehen kann.

Warum hat Tektronix Tunnelioden in Oszilloskop-Trigger-Schaltungen verwendet? Wahrscheinlich, weil Tunnelioden schnell schalten und auch als eine Art Flip-Flop arbeiten können. So setzt das Oszilloskop Tektronix 465 ein paar Tunnelioden ein, um einen Auslöseimpuls für die Logik-Schaltungen zu erzeugen. Eine Diode wird verwendet, um den Trigger-Impuls vorzubereiten, eine andere, um ihn zu „zünden“ [1]. Die Dioden sind erforderlich, damit die Schaltung definierte Betriebszustände für die Triggerlogik einnehmen kann.

Bild 1 zeigt eine Tunneliode in einer vereinfachten Vorspannungsschaltung. Tunnelioden weisen einen pn-Übergang mit sehr enger Verarmungszone auf, um den Tunneleffekt zu maximieren. Dies bedeutet, dass die Diode im Durchlassbereich beginnt, mit sehr wenig Vorspannung zu leiten. Der Durchlassstrom nimmt zu und erreicht eine Spitze (i_1) und dann wieder ein Minimum (i_2), wenn die Vorspannung wie in **Bild 2** weiter erhöht wird. Dieses Verhalten gibt der Tunneliode ihre negative Widerstandscharakteristik, so dass sie gut für einen Verstärker oder Oszillator verwendet werden kann. Einer der Schlüsselparameter der Tunneliode ist ihr Spitzenstromwert. Schauen wir uns Bild 1 noch einmal an. Die 10-mA-Diode wird mit 9,9 mA betrieben, also gerade am Spitzendurchlassstrom, wenn V_{trigger} hochimpedant ist. Die Diode wird in den Punkt des minimalen Durchlassstroms geschoben, wenn V_{trigger} ausreichend Strom zur Diode hinzufügt. Diese Triggercharakteristik zeigt auch, wie man eine Tunneliode testen kann. Wenn Sie den Widerstand mit einem DMM messen und es einen hohen Wert anzeigt, dann ist die Diode defekt. Ein niedriger Widerstandswert bedeutet, dass sie in

Ordnung sein kann, aber um dies sicherzustellen, müssen Sie noch überprüfen, ob die Diode zwischen dem Spitzen- und dem minimalen Stromwert umschaltet. In einigen kritischen Schaltungen muss auch überprüft werden, ob diese Werte einer Langzeitdrift unterliegen, ein bekanntes Problem solcher Bauteile. Sollten sich die Werte verändern, so könnte ein Neuabgleich und eine Kalibrierung der Schaltung erforderlich sein.

Tunnelioden sind ziemlich interessante Bauteile, die in den 1960er Jahren wirklich gebraucht wurden. Sie haben sich als nächste große Entdeckung nach dem Halbleitertransistor entpuppt. Heutzutage sind sie ziemlich schwer zu finden, aber unverzichtbar, wenn sie alte Elektronik restaurieren. Wenn Sie mehr über Tunnelioden und deren Anwendungsbereiche erfahren wollen, finden Sie auf den Tektronix-Oszilloskop-Fanseiten [2] viele Anregungen.

(140057)

Weblinks

- [1] <http://w140.com/mmm/tek-465-late.pdf>
 [2] http://w140.com/tekwiki/wiki/Tunnel_diodes

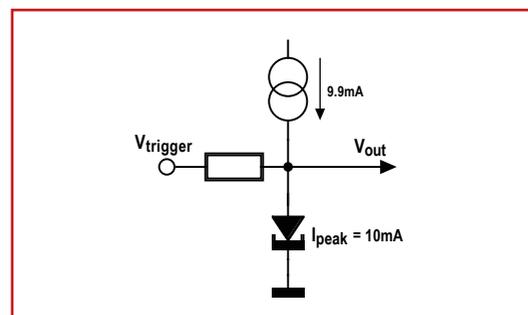


Bild 1.
Vereinfachte
Anwendungsschaltung einer
Tunneliode.

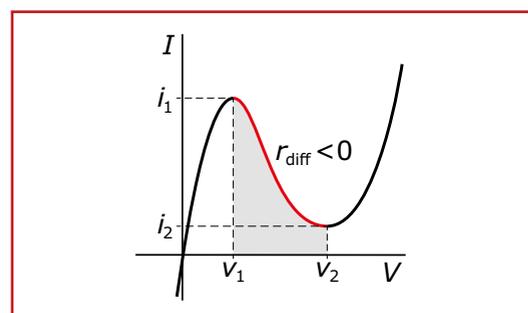


Bild 2.
Strom/
Spannungscharakteristik
einer Tunneliode.



DesignSpark Tipps & Tricks

Das Layout von Bussen

Von **Neil Gruending**
(Kanada)

In dieser Folge geht es darum, wie man in DesignSpark mit einem Bus umgeht.

In der letzten Folge ging es um mehrseitige Schaltungen und die Verwendung des Tools „Cross Probe“ in komplexeren Projekten in DesignSpark. Hieran wird nun mit den Eigenheiten von Bussen angeknüpft.

Bus-Typen

Ein Bus ist mittlerweile ein häufiges Strukturelement in Schaltungen, da man damit sehr schön

Signale grafisch gruppieren kann, wodurch der Schaltplan übersichtlicher und besser lesbar wird. Üblicherweise wird ein Bus als breitere Linie gezeichnet, bei der die abzweigenden Signale von und zum Bus mit Netznamen beschriftet sind (siehe **Bild 1**). Die fette blaue Linie symbolisiert den Bus und die abzweigenden Signale sind mit Data0...Data5 beschriftet. Nun zu beiden Bus-Typen „open“ bzw. „closed“ in DesignSpark:

Ein Bus vom Typ „open“ kümmert sich nicht wirklich um die Verbindungen. Es ist schlicht eine fette Linie im Schaltplan. Die Verbindungen mit ihm nutzen schlicht die Namen der Netzleitungen, die automatisch bei der Verbindung erscheinen. **Bild 2** zeigt ein Beispiel. Wenn man die Widerstände mit dem Bus verbindet, erzeugt DesignSpark automatisch um 45 ° abgewinkelte Verbindungen am Bus-Ende und zeigt die Namen der Netze an. Wenn man einen solchen Bus löscht, verschwinden auch alle Verbindungen. Man kann die Bezeichnungen durch Klick aufs Netz und Auswahl von „Change Net“ anpassen.

Ein Bus vom Typ „closed“ ist etwas intelligenter, denn er fungiert als Container für alle Bus-Netze und man kann nur genau diese Netze mit ihm verbinden. Man kann einen Bus von „open“ zu „closed“ ändern, indem man per Rechtsklick das Fenster mit den Properties aufruft. Über die Schaltfläche „Add Net“ öffnet sich ein Fenster (siehe **Bild 3**). Hier wird eine Liste der vorhandenen Netze angezeigt. Im Beispiel sind das die Netze N0001...N0004, die man so zum Bus hinzufügen kann. Benötigt man eine neue Gruppe von Netzbezeichnungen mit Zahlen am Ende (wie für einen Adress-Bus), dann nutzt man die Option „New Nets“. Bei einem 16-bit-Adress-Bus werden z.B. die Namen A0...A15 erzeugt, wenn man auf „Add To List“ klickt.

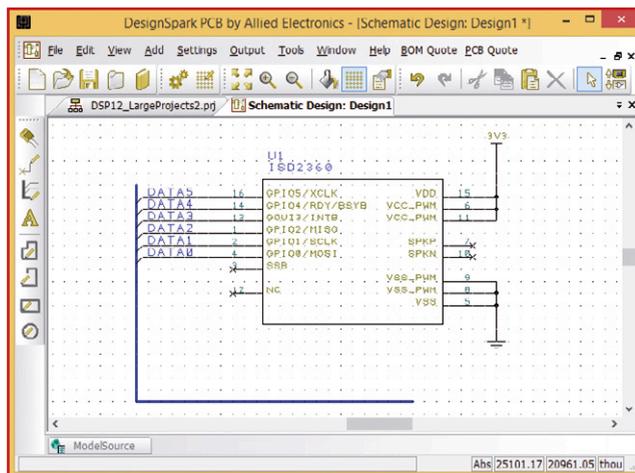


Bild 1.
Bus-Beispiel.

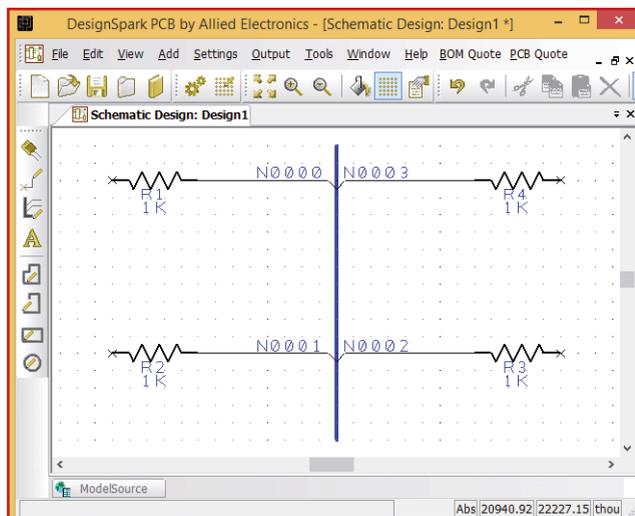


Bild 2.
Der „Open Bus“.

Open vs. Closed

Die Frage ist, welchen Bus-Typ man verwenden sollte, ist nicht einfach beantwortbar. Beide haben ihre Vorzüge. Ich ziehe den Typ „closed“ für größere Busse wie einen Datenbus vor. Für kleinere Ausführungen wie den SPI-Bus präferiere ich die offene Variante.

Einen Closed-Bus kann man kopieren. Die Kopie enthält alle Bus-Assoziationen, was für größere Busse sehr nützlich sein kann. Nur beim Kopieren zwischen unterschiedlichen Schaltplan-Seiten ist Vorsicht geboten, denn die Bus-Netze folgen den gleichen Regeln wie bei einer normalen Schaltung. Man sollte daher sicherstellen, dass Bus-Netze anders benannt sind, als es die Konvention Nxxxx vorgibt. Glücklicherweise wird man von DesignSpark beim Bus-Einfügen gefragt, ob man gleiche Netze verbinden will, wenn Netze gleichen Namens schon in der Schaltung vorhanden sind. Normalerweise erlaubt man deren automatische Verbindung.

Ein weiterer Vorteil ist der mögliche Einsatz des Tools „Cross Probe“. Wenn man es auf einen Bus anwendet, kann man gut sehen, wo überall in der Schaltung Busse verwendet werden, die mit den gleichen Netzen assoziiert sind. Dadurch kann man sehr schön sehen, wo überall in einer Schaltung ein Bus verwendet wird.

Man kann auch einen offenen Bus kopieren, doch als Kopie erhält man lediglich die Bus-Form. Das „Cross Probing“ funktioniert auch hier, allerdings nicht bus-weit, sondern nur bezogen auf einzelne Netze.

Den Bus nehmen

Wenn man in einer Schaltung einen Bus hinzufügt, dann erhält man eine fette Linie, zu der man Verbindungen ziehen kann. Es werden automatisch um 45 ° abgeschrägte Verbindungen zum Bus erzeugt. Man kann die Größe der Verbindung mit dem Bus über das Fenster „Bus Defaults“ bei den Einstellungen konfigurieren. Mit Hilfe der F-Taste kann man sogar die Richtung der Verbindung wechseln.

Ein Nachteil von Bussen ist, dass man leicht eine Verbindung vergisst und so „tote Leitungen“ produziert. DesignSpark kann so etwas leider nicht überprüfen. Auch der DRC (Design Rule Check) enthält keine passende Regel. Man kann das aber manuell checken. Am einfachsten öffnet man dazu die Projekt-Datei und startet den „Generic Net-

list“-Report via Menü Output → Reports. Das sollte man direkt mit der Projekt-Datei machen, denn vom Schaltplan aus werden nur die Netze dieser Seite geprüft. **Bild 4** zeigt das Ergebnis für einen Closed-Bus mit den Netzen D0...D3 und einem offenen Bus mit den Netzen A0...A1, die mit den Widerständen R1...R7 verbunden sind. Im Report sieht man, dass die Netze D2, D3 und A1 nur eine Verbindung haben. Man wird auch darauf hingewiesen, dass Netz D0 drei Verbindungen aufweist, was – je nach Schaltung – ebenfalls ein Fehler sein kann.

Fazit

Die beiden Bus-Typen von DesignSpark sind sehr einfach anzuwenden. Beide haben ihre eigenen Stärken. In der Online-Hilfe finden sich noch weitere Details zu ihrer Implementierung und Anwendung. In der nächsten Folge geht es dann um speziellere Fähigkeiten von DesignSpark PCB.

(140058)



Bild 3. Hinzufügen von Netzen zum Bus.

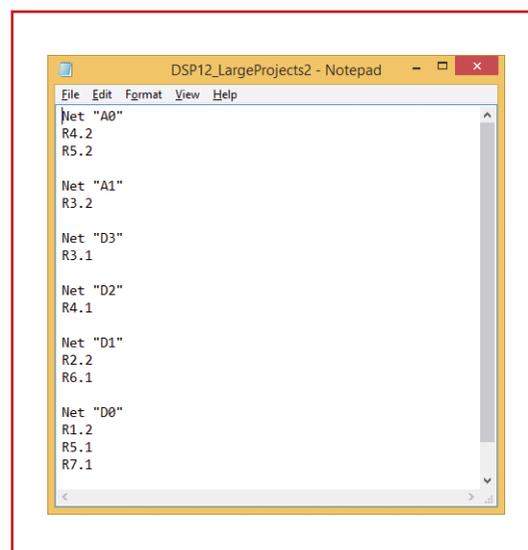
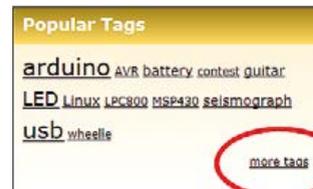


Bild 4. Report mit Netzlisten.

It's Tag Time!

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

Die Menschheit hat sich schon immer mit der Natur der Zeit (und wie sie zu messen wäre) beschäftigt. Auch das Elektor-Labor hat in dieser Beziehung einiges Bemerkenswertes geleistet. Ein Überblick...



Das Problem der Zeitmessung wurde 1967 gelöst, als die Generalkonferenz für Maße und Gewichte (CGPM, conférence générale des poids et mesures) die Sekunde neu definiert hat. Eine Sekunde ist so lang wie das 9192631770-fache der Periodendauer der dem Übergang zwischen den beiden Hyperfeinstrukturniveaus des Grundzustandes von Atomen des Caesium-Isotops 133 entsprechenden Strahlung - entschuldigen Sie bitte, ich war nicht ganz sicher und musste es kurz bei Wikipedia nachsehen.

Die CGPM hätte natürlich genau so gut das 9.000.000.000-fache oder einen anderen runden Wert wählen können, aber sie tat es nicht. Die neue Definition sollte so nahe wie möglich beim historischen Wert der Sekunde liegen. Allerdings ist das letzte Wort noch nicht gesprochen: Im Jahr 2012 haben die Physiker David Wineland und Serge Haroche den Physik-Nobelpreis für das Einfangen einzelner Ionen erhalten, ein zirkusreifes Kunststück, das - wie Experten vorhersagen - zu einer neuen Definition der Sekunde führen wird.

Uhren waren schon immer ein beliebtes Thema in Elektor und im Elektor-Labor wurden schon unzählige Projekte aufgebaut. Diese Projekte können in ein paar Kategorien unterteilt werden:

Hübsche Objekte: In dieser Kategorie finden wir Uhren, die gebaut wurden, um gut auszusehen. Hier finden wir Nixie-Designs, die Rheinturm-, Berlin- und Propeller-Uhr und andere kreative Zeitanzeige-Schaltungen.

Präzisionsinstrumente: Einige Leute benötigen Uhren mit Millisekunden-Genauigkeit oder mit Langzeitgenauigkeit. Diese basieren auf GPS und Funk (DCF77, France Inter, WWF, MSF), aber auch auf präzisen Echtzeituhr-Chips (RTC).

Programmierprojekte: Leute, die ausprobieren möchten, wie ein FPGA, ein Mikrocontroller oder gar ein PC programmiert wird, können sich an einer Uhr die Zähne ausbeißen. Das Thema Zeitmessung reicht von ganz einfach (Sekunden zählen) bis ganz schwierig (ewiger Kalender).

Add-on-Module: Kleine Projekte, die vor allem zeigen, wie man einen RTC-Chip verwendet und andere Projekte mit einer Zeitbasis ausstattet.

Nun, da Sie wissen, dass alle diese Projekte da draußen auf Sie warten, wie kann man sie finden? Das „Tag“ bringt es an den Tag! Natürlich können Sie die gesamte Elektor-Labs-Website durchsuchen, aber viel einfacher ist es, den Link „more tags“ zu verwenden, um sie alle zu finden, aus dem Dunkel zu treiben und ewig zu binden. Dann klicken Sie auf ein Tag, das Sie interessiert, und werden mit einer Liste von Projekten belohnt, die mit dem Tag in Zusammenhang steht. Ein Beispiel: Alle Uhren-Projekte finden Sie unter [1].

(140060)

Weblink

[1] www.elektor-labs.com/tags/time

Arduino, il salvatore

Wenn Sie nicht gerade die letzten zehn Jahre im Koma verbracht haben, dürften Sie von Arduino gehört haben. Jeder redet darüber und entwickelt Programme (Sketches) und Erweiterungsplatinen (Shields) für Arduino. Auch wir haben schon Dutzende von Elektor-Seiten mit Beiträgen mit oder über Arduino gefüllt. Hier ist noch eine!

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



Dieser Artikel ist jedoch ein wenig anders, denn es geht nicht um ein Arduino-Projekt, sondern um die Verwendung von Arduino als Werkzeug. Ich brauche Arduino oft, weil er so einfach zu programmieren und anzuwenden ist. Ich brauche ihn manchmal sogar als Taschenrechner, weil es schneller ist, ein Arduino-Programm (Sketch) zu schreiben als eine PC-Anwendung. Um Ihnen ein Beispiel zu geben: Wir mussten im Elektor-Labor den Prototypen des MIDI-Channel-Analyzers testen, der in einer der nächsten Ausgaben veröffentlicht wird [1]. Meine Kollegin Hedwig Hennekens hatte ein schönes MIDI-Keyboard von zu Hause mitgebracht, aber wir konnten damit den Analysator nicht richtig testen, weil wir nicht genau wussten, was dieses Instrument ausgab. Wir hatten ein klassisches Henne/Ei-Problem vor uns: wir brauchten den MIDI-Analysator, um das Verhalten des MIDI-Keyboards zu überprüfen, das den MIDI-Analysator überprüfen sollte. So riefen wir Arduino zu Hilfe. Es dauerte weniger als fünf Minuten, um ein Sketch zu schreiben, das jede Sekunde ein *Note-On* und *Note-Off* an einen MIDI-Kanal (inkrementierend) schickte. Ton Giesberts, Veteran im Elektor-La-

bor, fügte einen Transistor an die serielle Schnittstelle des Arduinos an, um sie MIDI-kompatibel zu machen (das DATA-Signal muss invertiert werden) und verband ihn mit dem Device-under-Test (DUT). Es hätte sogar ohne Transistor funktioniert, wenn wir einen seriellen Port in der Software emuliert hätten, aber das hätte auch mehr (Programmier-)Arbeit bereitet.

Wir brauchten den Arduino-Tester dann nur noch einzuschalten, uns zurückzulehnen und zu beobachten, wie der Test verlief. Dies zeigte mir das wahre Gesicht des Arduino, das Gesicht eines Retters (it.: *il salvatore*) aus aussichtslosen Situationen bei der Entwicklung und dem Test elektronischer Schaltungen. Ein einfach zu bedienendes und dennoch „mächtiges“ Multifunktionswerkzeug.

Also, wenn Sie jetzt fragen, was dieser Artikel soll, denken Sie daran, wenn Sie partout nicht herausfinden können, warum ihr beinahe fertiges, aber irgendwie nicht richtig funktionierendes Super-dupa-Gadget nicht so will, wie es soll.

(140059)

Weblink

[1] www.elektor-labs.com/node/3380

Chipkarten über USB auslesen

USB emuliert UART emuliert I²C

Von Florian Schäffer



In Elektor gab es schon häufig Projekte, in denen ein PC einen I²C-Baustein ansteuerte. Meist wurde dazu die Intelligenz eines Mikrocontrollers als Vermittler zwischen PC- und I²C-Schnittstelle genutzt. Doch es geht auch einfacher, wie an diesem Lesegerät für Chipkarten gezeigt wird. Wir benötigen nur ein paar Bauteile und ein wenig „dirty engineering“!

Es gibt zwei Arten von Chipkarten im gängigen Kreditkartenformat: die mit einem Prozessor und die ohne. Speicher-Chipkarten sind nur mit einem einfachen EEPROM (electrical erasable and programmable read-only memory) vom Typ 24Cxx [1] unter den acht Kontaktflächen bestückt (**Bild 1**). Diese Bausteine sind mit verschiedenen Speichergrößen von 128 Byte (1 kbit) bis 2048 Byte (16 kbit) verfügbar. Angesteuert werden sie über einen Zweidrahtbus, der je nach Hersteller als I²C oder Two-Wire-Interface (TWI) bezeichnet wird.

Solche Smartcards finden vor allem dort Anwendung, wo relativ kleine Datenmengen besonders preiswert gespeichert werden sollen. USB-Sticks bieten zwar viel mehr Speicherplatz, kosten aber auch deutlich mehr als eine Smartcard, die schon

für weniger als 1 € zu haben ist [2]. Zudem werden beim USB-Stick lediglich 100.000 Schreibzyklen garantiert, während die 24Cxx EEPROMS zehnmal so viel vertragen und die Daten auch 100 Jahre anstatt nur 10 Jahre erhalten bleiben. Außerdem sind die Speicherkarten einfacher zu handhaben als andere kleine Speicherlösungen wie SD-Karten, was vor allem bei täglicher Nutzung wichtig sein kann. Sie passen einfach gut ins Portemonnaie, wenn man sie gerade nicht braucht. Anzutreffen sind die kleinen Karten etwa bei der Zeiterfassung und Zutrittskontrolle, in Kantinen zur Kundenidentifizierung und in Sportcentern zur Erfassung der Trainingsparameter [6].

I²C aus dem USB

Selbstverständlich gibt es am Markt eine große Auswahl an Lesegeräten für Smartcards. Gebrauchte Geräte sind sogar schon für wenige Euro zu bekommen. Das Problem bei den älteren Geräten ist aber, dass diese oft eine PS/2-, eine serielle oder gar eine parallele Schnittstelle benötigen. Bei modernen PCs und vor allem an Laptops sind solche Schnittstellen aber nicht unbedingt mehr vorhanden.



Bild 1.
Achtpolige Kontaktfläche einer Chipkarte.

Bei vielen Lesegeräten mit USB-Anschluss bereiten dagegen Treiber Schwierigkeiten, weil sie nicht mehr zu aktuellen Windows-Versionen kompatibel sind. Auch die Software kann zum Problem werden. Sie muss in der Lage sein, mit dem Lesegerät zu kommunizieren, die Daten der Karte auszulesen und gegebenenfalls auch speichern zu können. So gibt es reichhaltig Programme, die die Daten einer elektronischen Gesundheitskarte (eGK) auslesen können, die aber nicht in der Lage sind, reine (prozessorlose) Speicherkarten zu lesen und die Daten als Binärdatei zu sichern. Längere Recherchen haben ergeben, dass es keine geeignete Freeware gibt und die einzige kommerzielle (das heißt, kostenpflichtige) Anwendung das Programm *Smartcard Commander* ist [3]. Für das angestrebte Projekt wäre es aber smart, wenn das Programm nicht nur die Daten auswerten, sondern auch selber lesen und schreiben könnte, so dass wir mit einer minimalen externen Hardware auskommen.

Die Lösung des Dilemmas wäre, die I²C-Kommunikation direkt aus dem Hut, nein, dem USB zu zaubern. Wir müssten hierzu nur zwei Leitungen zur Smartcard gemäß I²C-Protokoll von unserer PC-Software aus bedienen. Als Hardware könnten wir für wenig Geld einen USB-Seriell-Adapter (**Bild 2**) erstellen und leicht modifizieren. Die Treiber der USB-Adapter stellen auf dem PC einen virtuellen COM-Port zur Verfügung, der für unsere PC-Software wie eine echte Schnittstelle aussieht. Die USB-Seriell-Adapter stellen zwar keine Signale mit RS232-Pegel (± 12 V) zur Verfügung, sondern, da der USB eben mit +5 V arbeitet, nur TTL-Pegel. Aber das ist genau das, was wir für I²C brauchen!

Kenner der RS232-Schnittstelle werden jetzt anmerken, dass dieses Projekt schlichtweg nicht zu realisieren ist, da die serielle Schnittstelle des PCs es nicht erlaubt, per Software die „normalen“ Datenleitungen RxD und TxD per Bitbanging auf Low oder High zu setzen beziehungsweise zu lesen. Das geht aber mit den Leitungen CTS, RTS und DTR! Da wir nur zwei digitale Ausgänge (Takt und Daten) und einen digitalen Eingang (Daten) benötigen, können wir diese RS232-Signale für unsere Zwecke „missbrauchen“. Es ist nicht regelgerecht, aber es funktioniert. Voraussetzung ist, dass der USB-Seriell-Adapter die drei Signale auch anbietet.

Beinahe keine Schaltung

Eine Smartcard hat üblicherweise eine Kontaktfläche mit (je nach Hersteller) sechs oder acht Kontak-



Bild 2.
Für wenige Euro zu haben:
ein USB-Seriell-Adapter.

ten. Beide Versionen sind kompatibel. Die beiden unteren Kontakte C4 und C8 werden zurzeit ohnehin nur von kontaktlosen Karten für den Anschluss der RFID-Antenne genutzt. Laut **Tabelle 1** besitzt ein EEPROM nur fünf Anschlüsse (der VPP-Kontakt wird nicht mehr genutzt), von denen die Schaltung in **Bild 3** gar nur vier kontaktiert: die Versorgungsspannung (VCC und GND) sowie den I²C-Bus mit Datentakt SCL und Daten SDA.

Die Schaltung verdient diese Bezeichnung auch fast gar nicht und dient lediglich dazu, die unidirektionalen Signalleitungen CTS (clear to send zum PC) und RTS (request to send vom PC) der Schnittstelle zu der bidirektionalen I²C-Datenleitung SDA zu verknüpfen. Der Transistor in der PC-Sendeleitung, der zur Auswertung des ACK-Signals der Karte erforderlich ist, invertiert die Signalpegel. Dies muss später im Programm berücksichtigt werden. R2 ist der bei der I²C-Datenleitung erforderliche Pullup-Widerstand. Eigentlich wäre auch bei der Taktleitung ein solcher Pullup erforderlich, aber da diese Leitung

Tabelle 1. Belegung der achtpoligen Kontaktfläche

Kontakt	Name	Funktion
C1	VCC	5 V
C2	RST	Reset (hier nicht genutzt)
C3	CLK	Takt SCL
C4*	NC	keine
C5	GND	Masse
C6	VPP	EEPROM-Programmierspannung (hier nicht genutzt)
C7	I/O	Daten SDA
C8*	NC	keine

* bei sechspoligen Kontaktflächen nicht vorhanden

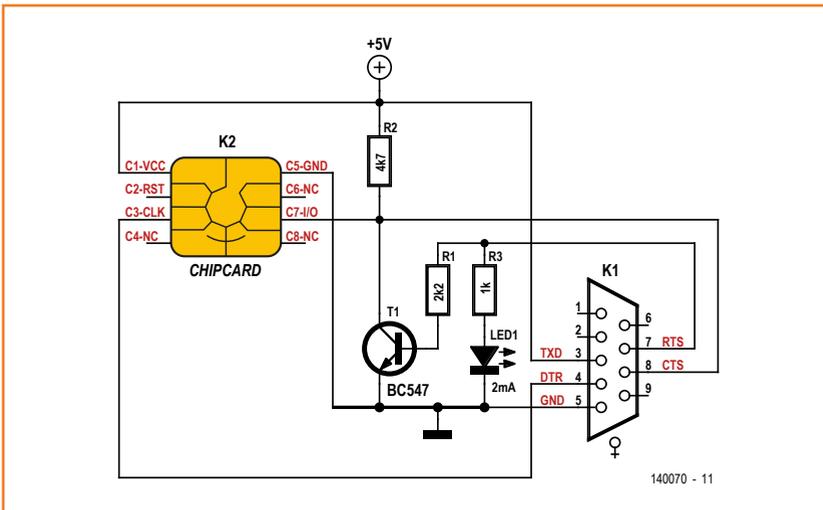


Bild 3.
Die Mini-Schaltung des Kartenlesers.

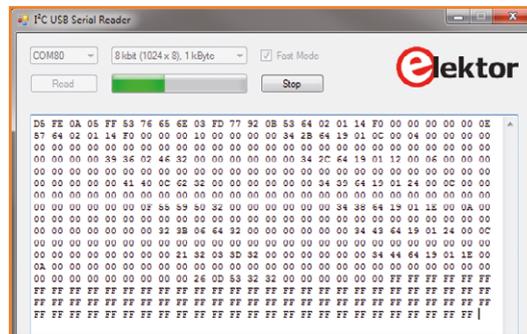


Bild 4.
So erscheinen die Daten im Fenster des PC-Programms.

von der Software auf Low und High gelegt wird, können wir ruhigen Gewissens darauf verzichten. Ein wenig „dirty“ halt. Die Sendeleitung TxD wird dazu missbraucht, die für den Speicherchip notwendige Spannungsversorgung von +5 V bereitzustellen: Sie wird in der Software mit einem Break-Befehl permanent auf High gesetzt. Die LED signalisiert den Datenaustausch. Theoretisch wäre es auch noch möglich, den (hier nicht eingezeichneten) Schaltkontakt des Kartenhalters zu verwenden, um auszuwerten, ob eine Karte eingesteckt wurde. Wie im Titelbild zu sehen, kann die Schaltung in Minutenschnelle auf einem Stückchen Lochraster aufgebaut werden. Ein Kartenhalter ist für einen Euro erhältlich [4]. Die Schaltung besitzt zwar eine neunpolige Sub-D-Buchse, funktioniert aber nur an einem USB-Adapter und darf nicht an einer echten RS232-Schnittstelle betrieben werden!

Bitbanging

Das I²C-Protokoll ist relativ simpel. Der Master erzeugt den Takt und generiert eine Start-Bedingung, bei der während einer High-Phase von

SCL die SDA-Leitung von High auf Low wechselt. Anschließend startet die Aufforderung für den Zugriff auf eine Speicheradresse. Dieser scheinbare Schreibbefehl besteht aus der Geräteadresse (bei Speicherkarten immer Null) und der Speicheradresse. Anschließend wird noch mal die Geräteadresse mit gesetztem Lese-Bit gesendet. Daraufhin übermittelt der Slave die gewünschten Daten. Mit dem ACK-Bit signalisiert der Slave, dass die Daten vollständig gesendet wurden. Wird das Bit vom Master unterdrückt, indem er die Leitung während des entsprechenden Taktes auf Masse zieht, wechselt der Slave in den sequentiellen Lesemodus und sendet solange fortlaufend den Inhalt der weiteren Speicheradressen, bis der Master das ACK-Bit freigibt. Anschließend sendet der Master noch einen Stoppbefehl. Eine Besonderheit bei Speicherchips ist, dass nicht erkannt wird, wann die letzte Speicherzelle ausgelesen wurde. Beim sequentiellen Lesen wird am Ende einfach mit der ersten Speicheradresse fortgefahren.

I²C bietet den Vorteil, dass ein Master den Bustakt vorgibt und sich alle anderen Geräte (Slaves) daran zu orientieren haben. Solange nur der PC Master sein kann und die angeschlossenen Geräte keine Anforderungen an eine Mindesttakttrate haben, kann auch ein PC unter Windows, das ja keine Echtzeitanwendungen erlaubt, den Takt erzeugen. Der Takt ist nicht hundertprozentig regelmäßig und weit von den üblichen 100...400 kbit/s entfernt. Für die meisten Anwendungen ist dies aber unerheblich, da die Slaves tolerant sind und sich einfach an die Taktvorgabe halten.

Im frei verfügbaren Visual Studio Express [5] wurde ein einfaches Programm in Visual Basic geschrieben, das eine Speicherkarte auslesen kann (Bild 4). Den VB-Code können Sie kostenfrei unter [7] herunterladen. Das Programm ist möglichst einfach gehalten und kann natürlich noch erweitert werden (zum Beispiel zum Auslesen eines einstellbaren Speicherbereiches). Da der normale Timer von Visual Studio nicht präzise genug ist, wird der etwas unbekannte Multimedia-Timer genutzt. Auf einem durchschnittlichen System sollte selbst die kürzeste Intervallzeit von 1 ms keine Probleme bereiten, da genügend Zeit zwischen zwei Ticks verbleibt, um die notwendigen Anweisungen auszuführen. Das Programm zählt dabei einfach die erzeugten Taktflanken und ändert zum entsprechenden Zeitpunkt den Pegel an der Datenleitung RTS beziehungsweise fragt

den Zustand an der Dateneingangsleitung CTS ab. Neben dem hier gezeigten Auslesen von Kartendaten sind viele weitere Möglichkeiten denkbar. Selbstverständlich könnten auch Daten auf der Speicherkarte durch Schreiben geändert werden. Auch die Auswertung von Sensoren wie dem Temperatursensor DS1621 mit integriertem A/D-Wandler und I²C-Schnittstelle ist mög-

lich. Viele LCD-Module können über I²C angesteuert werden, so dass ein Statusdisplay für den PC (zum Beispiel zur Anzeige von Tweets) schnell realisiert werden kann. Denken Sie aber daran, dass sich der Client keinesfalls als Master aufspielen darf.

(140070)

Weblinks:

- [1] ATMEL Datenblatt AT24Cxx: www.atmel.com/Images/doc0180.pdf
- [2] Bezugsquelle Smartcards: www.reichelt.de/Chipkarten/CHIPKARTE-256B/3/index.html?&ACTION=3&LA=2&ARTICLE=37015&GROUPID=5967&artnr=CHIPKARTE+256B
- [3] Smartcard Commander: www.chipdrive.de
- [4] Bezugsquelle Chipkartenhalter: www.tme.eu/de/katalog/?search=Smart+Card&id_category=31
- [5] Visual Studio Express: www.visualstudio.com/downloads/download-visual-studio-vs#d-express-windows-desktop
- [6] Projekt zur Auswertung von Trainingsdaten des Autors: www.blafusel.de/misc/miha_zirkel.html
- [7] www.elektor-magazine.de/140070

DIY-LED-Taschenlampe

Besser als fertig gekauft

Frage: Warum selbst bauen, wenn man etwas auch preiswert fix & fertig kaufen kann?

Antwort: Erstens weil es Spaß macht und zweitens weil es besser als China-Billigware ist.

Technische Daten

- Effiziente Selbstbau-LED-Taschenlampe
- Betrieb mit Spannungen von 1 bis 3 V
- Wirkungsgrad von bis zu 78,8 %
- Helligkeit bleibt bis hinab zu 1,2 V stabil

Als *homo electronicus* ist es dem Autor zutiefst zuwider, etwas Elektronisches einfach so zu kaufen, wenn er es auch selber besser machen kann.



Von
Wolfgang Schmidt
(D)

Kleine LED-Taschenlampen gibt es massenhaft überall. Bei ebay.de werden zur Zeit gut 17.000 solcher LEDs mit Gehäuse und Batterie für Preise ab einem Euro angeboten – inklu-

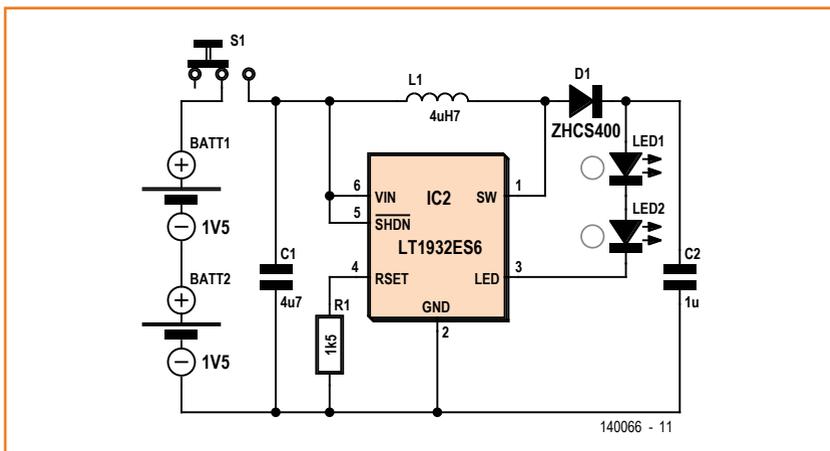


Bild 1.
Die Schaltung des Taschenlampen-Elektronik ist einfach: Außer dem Konverter-IC und den LEDs sind nur fünf weitere Bauteile erforderlich.

sive Porto für den Versand aus China!
Der Gedanke, dass man so etwas selbst preiswerter bauen könne als es die perfektionierte chinesische Massenproduktion kann, der ist mehr als illusorisch. Aber die Sache etwas besser zu machen dürfte durchaus drin sein, denn bei diesen Preisen muss logischerweise dermaßen gespart werden, dass sicher nicht die technisch beste Lösung dabei herauskommt. Preis und Qualität tendieren nunmal dazu, ziemlich negativ zu korrelieren. Und wenn man etwas Besseres möchte, dann muss man schon mehr Geld in die Hand nehmen: LED-Taschenlampen von Markenherstellern sind richtig teuer.

Eine Alternative ist der Selbstbau, der neben dem angenehmen Gefühl, etwas selbst gemacht zu haben, den Zugriff auf gute Technik erlaubt. Die hier beschriebene LED-Taschenlampe im Selbstbau überzeugt durch einen hohen Wirkungsgrad und die Ausnutzung auch noch des letzten Tropfens an Energie, der in den Batterien steckt.



Das Konzept

Es dürfte jedem klar sein, der schon einmal mit LEDs gearbeitet hat, dass es zu einer LED-Lampe mehr als LED plus Batterie braucht. Eine LED nämlich wird aufgrund ihres niedrigen differentiellen Widerstands nicht mit einer konstanten Spannung, sondern mit einem konstanten Strom betrieben. Ergo braucht es auch mehr als nur einen Vorwiderstand. LED-Taschenlampen sind also immer komplexer als die mit Glühlämpchen, die ja zum direkten Anschluss an eine Batterie hergestellt werden.

Dafür ist der Selbstbau von kleinen LED-Taschenlampen mechanisch einfacher als der von Glühlampenausführungen. Das gilt zumindest dann, wenn die Helligkeit von bedrahteten LEDs ausreicht; denn die gibt es mittlerweile nicht nur mit hoher Lichtausbeute, sondern mit einer so engen Strahlführung (20° und weniger), dass die Notwendigkeit eines Parabolspiegels zur Fokussierung entfällt. Man benötigt also zumindest eine LED - der Autor entschied sich für zwei in Serie - sowie eine Elektronik, die batteriegespeist einen passenden Konstantstrom durch die LEDs fließen lässt. Um möglichst geringe Verluste zu erreichen, benutzt man heutzutage geschaltete Konstantstromquellen. Und damit man nicht allzu viele Batterien in Reihe schalten muss, wenn man die für zwei LEDs in Serie nötige Spannung von etwa 7 V erreichen will, ist ein sogenannter Step-Up-Konverter erforderlich.

Wie es der Zufall will, gibt es von Linear Technology mit dem LT1932 ein IC, das bei hohem Wirkungsgrad genau diese Anforderungen erfüllt und zu akzeptablem Preis im Handel erhältlich ist. **Bild 1** zeigt, dass der Autor dieses IC ganz konventionell beschaltet hat. Es benötigt nur fünf externe Bauteile. Mit R1 wird der Strom durch die LEDs eingestellt. Beim angegebenen Wert von 1,5 kΩ liegt er bei den LED-typischen 15..20 mA.

Lampenbau

Damit der Nachbau schön einfach wird, wurde eine Platine (siehe **Bild 2**) entwickelt, auf der alle Bauteile, die beiden LEDs und sogar der Batteriehalter Platz finden. Die Platine selbst als auch deren Layout-Dateien sind über die Elektor-Webseite zu diesem Artikel [1] erhältlich. Da es sich beim IC schon um ein SMD handelt, wurden die restlichen fünf Bauelemente ebenfalls als oberflächenmontierbare Variante vorgesehen. Dies hat den Vorteil, dass die Platine klein ausfällt. Bei so wenig Bauteilen sollte die Bestückung auch für

weniger Geübte kein Problem sein. Die beiden LEDs werden abgewinkelt auf die Platine gelötet. Zur Befestigung bohrt man passende 5-mm-Löcher in das Kunststoffgehäuse und steckt die LEDs soweit durch, dass zumindest die Kalotten der LEDs ordentlich herausgucken. Die Platine kann dann einfach mit doppelseitigem Klebeband im Gehäuse fixiert werden.

Das vorgeschlagene Gehäuse kann zwei Micro-Zellen (Typ AAA) beherbergen. Bei Primärzellen kommt man so auf eine Nennspannung von 3 V. Das IC hält den Strom durch die LEDs auch bei sehr niedrigen Spannungen voll aufrecht. Bei 1,2 V fließen immer noch 15 mA! Die Elektronik holt also auch noch das letzte Tröpfchen Energie aus jeder Batterie und kann sogar mit eigentlich schon leeren Batterien noch Licht produzieren. Erst bei unter 1 V wird es dunkel. Dieser Vorteil hat allerdings den Nachteil, dass bei der Verwendung von Akkus diese tiefentladen werden könnten, denn 0,5 V tun weder einer NiCd- noch einer NiMH-Zelle auf Dauer gut.

Im Prinzip wäre damit fast schon alles gesagt. Vielleicht noch folgende Hinweise: Es empfiehlt sich, LEDs mit so vielen Candela zu kaufen, wie man kriegen kann. Die vom Autor verwendeten Exemplare kosten nicht viel und sind ganz schön hell. Manche dieser LED-Typen verkraften auch dauerhaft Ströme von 30 mA. Will man das ausnutzen, kann man für R1 einen Wert von 750 Ω verwenden. Das IC kann auch mehr als zwei LEDs in Serie treiben. Bei vier LEDs in Serie erhöht sich aber die minimale Versorgungsspannung fast auf 2 V. Wichtig ist noch, dass man den Schalter S1 nur an der Stelle einschleifen sollte, an der er eingezeichnet ist. Unterbricht man nämlich die LED-Kette direkt, steigt die Leerlaufspannung auf das 15- bis 20-fache der Batteriespannung, was für das IC gefährlich werden kann. Der Autor war auch so gründlich und hat den Wirkungsgrad seines Prototypen gemessen. Bei 3 V erreichte er mit 78,8 % ziemlich genau den Wert, der im Datenblatt für eine Spule mit 4,7 μ H angegeben ist. Selbst bei der niedrigen Versorgungsspannung von 1,2 V lag der Wirkungsgrad immer noch bei guten 69,3 %.

(140066)

Weblink: [1] www.elektor-magazine.de/140066

Stückliste

Widerstände:

R1 = 1k5, 0805

Kondensatoren:

C1 = 4 μ 7/6,3V, Vielschicht, 0805

C2 = 1 μ /16V, Vielschicht, 0805

Spulen:

L1 = 4 μ 7, Würth Elektronik Induktivität
1812 Case SMD

Halbleiter:

D1 = ZHS400, Schottky
LED1, LED2 = LED, weiß, 5 mm
(z.B. Reichelt 5-25000 WS)
IC1 = LT1932ES6

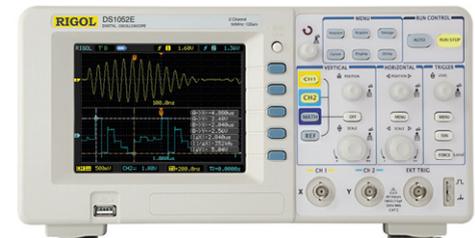
Außerdem:

S1 = Schalter für Platinenmontage, stehend, EG1213
Batteriehalter für 2 Micro-Zellen (AAA)
Gehäuse (z.B. Hammond 15930)
Platine # 140066-1



Bild 2. Der Bestückungsplan der vom Elektor-Labor entwickelten Platine.

UNSCHLAGBAR beim Preis-Leistungsverhältnis.



Rigol DS1000E Oszilloskope

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

ab € 284,41
inkl. MwSt. und Versand



Rigol DS1000Z Oszilloskope

4 Kanäle, 70/100 MHz, 1 GSa/s Abtastrate,
12 Millionen Messpunkte Speicher, USB, LAN,
professionelle Mess- & Analysefunktionen,
optional mit eingebautem Funktionsgenerator,
3 Jahre Garantie

ab € 535,50
inkl. MwSt. und Versand

Machen Sie Ihr **LEBEN** leichter.
Führende **LABORTECHNIK**
mit **BATRONIX** Zufriedenheits-
garantie

- ✓ Rechnungskauf
100% sicher und schnell. Erst
nach Erhalt der Ware zahlen.
- ✓ Große Auswahl ab Lager
- ✓ 30 Tage testen
- ✓ Geld zurück Garantie

Jetzt Angebote nutzen:
www.batronix.com/go/35

NEU

Rettung für die Lötstation

Alte Technik repariert und verbessert

Von Ingo Burret

Viele Jahre haben sie wertvolle Dienste im Labor geleistet, die legendären Magnastat-Lötstationen der Firma Weller. Doch langsam kommen sie in die Jahre und klagen über das eine oder andere Zipperlein. Heiße Liebe rostet nicht, sie verbrennt! Doch ist das ein Grund, das Schätzchen zu entsorgen?



Während heutzutage selbst in der einfachsten Lötstation eine ausgeklügelte Regelelektronik steckt, wurde das Problem der Temperaturregelung bei den berühmten türkisfarbenen Lötstationen WTCP 51-LT, WTCP-S, WTCP 50, WTCP 51 [1] (und älter) der Firma Weller magnetomechanisch gelöst.

Bild 1 zeigt den inneren Aufbau des Lötkolbens. Die Lötspitze (1) ist fest mit einer Metallkappe (2) verbunden, die den Temperaturfühler dieses Systems darstellt. Die Kappe ist ferromagnetisch und zieht im kalten Zustand einen Dauermagneten (3) an. Über eine Schubstange wird die Bewegung auf eine Kontaktbrücke (4) übertragen, die den Heizstrom (5) einschaltet.

Die Zahl auf der ferromagnetischen Kappe ist ein Code für die Regeltemperatur (von 260°C bis 480°C). Steigt die Temperatur beim Aufheizen gegen diesen Regelwert an, so verliert der Magnastat durch den Curie-Effekt [2] schlagartig seine ferromagnetischen Eigenschaften. Er kann den Dauermagneten nicht mehr festhalten. Der Magnet fällt ab und betätigt den Schalter, wodurch die Stromzufuhr zum Heizelement unterbrochen wird. Kühlt sich die Spitze etwas ab, so zieht der Temperaturfühler den Dauermagneten wieder an und das Spiel beginnt von neuem. Dieses Kippverhalten des Magnastaten ist sehr stabil. Es unterliegt keinem Verschleiß durch Alterung oder Materialermüdung.

Die Curie-Werte der Lötspitzen weisen zudem eine sehr geringe Streuung auf. Ein weiterer Vorteil dieser Anordnung ist, dass beim Auswechseln der Spitze der Kolben ausgeschaltet ist und der Heizkörper bei fehlender Spitze nicht durchbrennen kann.

Heiße Kontakte

Ein oft auftretendes Problem ist aber, dass der Magnetschalter festbrennt. Dann muss man klopfen und schütteln, in der Hoffnung, dass der Schalter wieder abfällt. Die Lötstation ist ja sonst ganz ok (man denke nur an die Unzahl von Lötspitzen in verschiedenen Formen und mit unterschiedlichen Temperaturwerten in der Schublade). Und man hat sie auch über die Jahre liebgewonnen, so dass man sie wegen eines solch geringen Makels nicht auf den Müll schmeißen mag. Zwar kann man den Magnetschalter ersetzen (es gibt dieses Ersatzteil als eine Einheit mit Schubstange und Permanentmagneten). Aber das ist teuer; und wenn man feststellt, dass die Heizwicklung auch noch durchgebrannt ist, ist es doppelt ärgerlich.

Also hat sich der Autor Gedanken gemacht, wie man den Magnetschalter ohne großen Aufwand entlasten kann. Da über die zweiadrige Zuleitung eine Wechsellastspannung von 24 V den Heizkörper versorgt (eine dritte Leitung ist für den

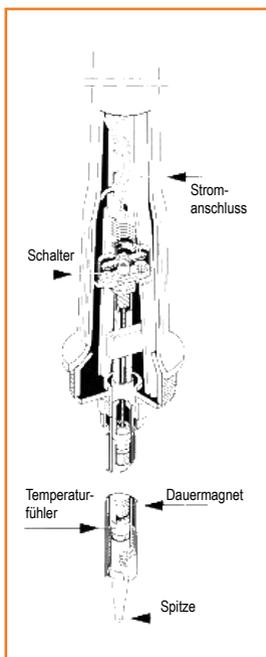


Bild 1. Aufrisszeichnung des Magnastat-Lötkolbens (Zeichnung: Weller).

Potentialausgleich vorgesehen), fallen ein Mosfet oder ein bipolarer Transistor für eine einfache Lösung aus. Relais wären zu groß für den Einbau in den LötKolben. Die Lösung ist ein Triac, der klein und fast so zu beschalten ist wie ein Transistor (**Bild 2**). Die „Elektronik“ umfasst noch einen Widerstand, der den Zündstrom des Triacs und damit den Strom durch den Schalter auf ein erträgliches Maß reduziert.

Der Triac passt gut auf einen trapezförmigen Pertinaxabschnitt, der im Griff des LötKolbens (**Bild 3**) untergebracht wird. So hält der Magnetschalter ewig, bis dass der Tod euch scheidet...

(130008)

Weblinks

- [1] www.weller.de/products/product.php?pid=431
- [2] <https://de.wikipedia.org/wiki/Curie-Temperatur>

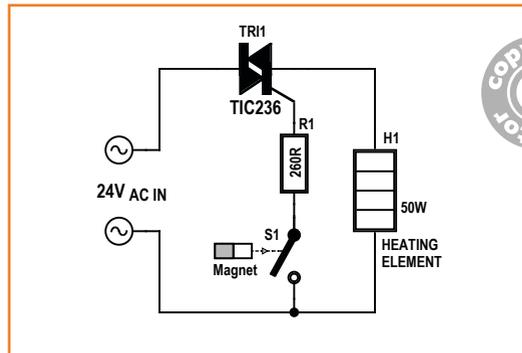


Bild 2. Ein Triac und ein Widerstand sorgen für einen geringen Strom durch den Magnetschalter.



Bild 3. Der Triac kann mit dem Widerstand leicht auf einem Pertinaxplättchen im Griff des LötKolbens angebracht werden.

Vegane Mausefalle

Tierschonendes Kammerjagen

Von **Manfred Haemmerle** (Österreich)

Mäuse sind eigentlich possierliche Tierchen, können aber im Haus großen Schaden anrichten und gesundheitsgefährdende Absonderungen hinterlassen. Deshalb bleibt keine Wahl: Wer Mäuse im Haus hat, muss sie fangen und nach draußen befördern. Dead or alive!



Vor zwei Jahren passierte das Maleur: In einer Schlagfalle steckte eine Maus, die durch den eigentlich tödlichen Schlag nur schwer verletzt

herumzappelte, aber nicht ins Jenseits befördert worden war. Mir blieb nichts anderes übrig, als die arme Maus mit der Axt... Na, lassen wir das

tig für den PIC, bei Alkalizellen senkt die Verpolschutzdiode D2 die Spannung auf für den Controller gerade noch zulässige 5,4 V. Auch D3 stellt einen Verpolschutz dar, allerdings dürfte die Diode, wenn sie zum Einsatz kommt, schnell den Wärmetod sterben.

Rechts oben ist der Anschluss für den Magneten zu sehen (K2), der beispielsweise aus einer elektrischen Klingel (deren Klöppel als Stift dient) oder wie in **Bild 3** zu sehen aus einem elektrischen Toaster stammt. Vielleicht findet sich ja auch ein kleiner Hubmagnet in der Bastelkiste. Der stromhungrige Magnet (er benötigt kurzzeitig mehr als 1 A) wird direkt aus U_{bat} an K1 mit Energie versorgt. Weil die Akkus mit der Zeit an Kraft verlieren, habe ich die beiden großen 4.700- μF -Kondensatoren C3 und C4 als Energiespeicher parallel zu den Batterien geschaltet. Dieser garantiert, dass der Magnet auch noch bei fast entladenen Batterien anzieht. Der Magnet wird über den Darlingtont transistor T2 angesteuert, der immerhin 1,25 A schalten muss. Hier kommt ein TIP112 zum Einsatz.

Die Infrarot-Lichtschanke besteht aus einer IR-Leuchtdiode und einem auf der anderen Seite des Korbes angebrachten IR-Fototransistor T1 (OP803SL) als Empfänger. Sowohl die LED als auch T1 sind über Transistoren (T3 als Verstärker, T4 als Pegelinverter) mit dem Mikrocontroller gekoppelt.

Ein PIC12F683 ist das Herzstück der Schaltung. Alle Funktionen werden mit Zeitschleifen realisiert. Jede Zehntelsekunde wird an GP2 des Controllers ein nur wenige Mikrosekunden langer Impuls ausgegeben. Die Software kontrolliert, ob dieser Impuls bei GP0 auch wieder zurückkommt. Geschieht dies (nach mehrfachem Fehlercheck) nicht, wird über GP1 der Magnet ausgelöst.

Wenn der Jumper JP1 gesteckt ist und die Kontroll-LED (LED2) langsam blinkt (alle 2 s), ist die Falle scharf. Blinkt sie dagegen nervös und rasch, signalisiert sie, dass zu viel Umgebungslicht auf den Fototransistor fällt. Diese Kontrollsignale würden aber selbst eine durchschnittlich intelligente Maus stören (ganz abgesehen vom zusätzlichen Stromverbrauch). Daher wird diese Funktion nur nach dem Einschalten zur Kontrolle benutzt, danach wird JP1 gezogen.

Der normale Stromverbrauch liegt im Schnitt bei nur 660 μA . Damit kann ein voller Batterie-



Bild 3.
Hier wird ein Magnet aus einem Toaster verwendet. Die Freilaufdiode ist direkt an der Induktivität angebracht.

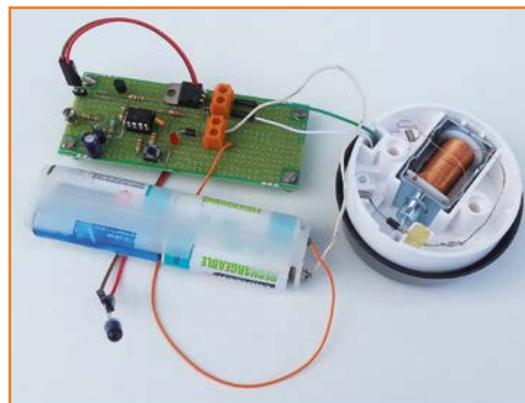


Bild 4.
Der elektronische Teil der Mausefalle auf einem Stück Lochraster, aufgebaut im Elektor-Labor.

satz die Schaltung viele Tage aktiv halten. Ist diese aber dann in die Falle gegangen, geht der PIC12F683 in den Sleep-Mode und verbraucht nur mehr circa 40 μA . Die Batterien werden damit soweit geschont, dass ein Ausschalten der Elektronik eigentlich nicht nötig wäre. Aber es kann nicht Sinn der Sache sein, die Maus verhungern zu lassen, und laut Tierschutzgesetz müssen Lebendfallen ohnehin täglich kontrolliert werden. Um die Falle wieder zu aktivieren, reicht es aus, die Reset-Taste zu betätigen. Wird die Falle nicht benötigt, schaltet man die Elektronik mit dem Hauptschalter S2 ganz ab (oder entfernt die Batterien).

Das gut dokumentierte C-Programm ist mit dem CCS-C-Compiler geschrieben, in der MPLAB-Umgebung kompiliert und mit einem PICKit3 auf den Controller übertragen worden. Das war völlig unproblematisch. Die Controller-Firmware kann als Quellcode und Hex (genau wie das Platinenlayout) unter [2] kostenfrei heruntergeladen werden. Die Schaltung wurde im Elektor-Labor auf einer Lochrasterplatte (**Bild 4**) aufgebaut, es wurde aber auch eine „gedruckte“ Schaltung entworfen

Stückliste

Widerstände:

R1,R3,R5,R8 = 10 k
R2 = 150 Ω
R4,R6 = 1 k
R7,R9 = 3k3

Kondensatoren:

C1 = 100 n
C2 = 10 μ/50 V, RM2 mm
C3,C4 = 4.700 μ/50 V, RM10 mm
C5 = 220 n

Induktivitäten:

L = Hubmagnet (siehe Text)

Halbleiter:

D1...D3 = 1N4007
LED1 = IR-LED, 940 nm

LED2 = LED rot

T1 = IR-Fototransistor OP803SL oder PBY62

T2 = TIP112

T3 = BC547C

T4 = BC557C

IC1 = PIC12F683-E/P, programmiert
Elektor 140011-41 [2]

Außerdem:

S1 = Taster

S2 = Kleiner Schalter 1x an

K1,K2 = Schraubklemme RM 5,08 mm

JP1,K3,K4,K5,K6 = 2x1-Stiftleiste

Batteriehalterung für vier Zellen und
Batterien/Akkus

1 Jumper

Platine 140011-1 [2]

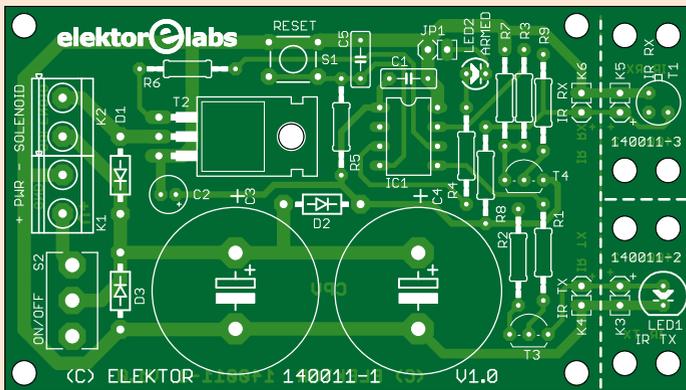


Bild 5. Das Layout der Mausefallen-Platine mit den beiden Abschnitten für die Lichtschranke.

(Bild 5). Zunächst werden die kleinen Abschnitte für die IR-Diode und den Fototransistor abgetrennt und die bestückte Hauptplatine auf der Rückseite des Käfigs angebracht, um die zweite Öffnung zu verdecken. Der Fototransistor wird nun an der einen, die IR-LED an der anderen Seite des Käfigs so befestigt, dass die eintretende Maus die Lichtschranke auslöst. Für den Anschluss von Batteriepack und Magnet sind Schraubklemmen vorgesehen. Die IR-LED und der Fototransistor werden, so wie in Bild 6 zu sehen, einfach in ein Paar Reihenbuchsen gesteckt. Die Freilaufdiode D1 parallel zur Spule ist direkt am Magneten befestigt.

Prompter Erfolg

Kurze Zeit nach Fertigstellung der Mausefalle zeigte sie bei einem realen Test ihr Können. Eine Maus war wieder einmal als Gratismieter im Keller eingezogen. Jetzt galt es, die Konstruktion zu testen. Im hintersten Winkel der Mausefalle wurden einige selbstgebackene Nussplätzchen abgelegt in der Hoffnung, der Neumieter könne ihnen nicht widerstehen.

Nur einen Tag später saß ein kleines, niedliches Mäuschen in der Falle. Allein schon ihretwegen hat sich der Bau der Falle gelohnt. Und als im folgenden Winter eine ganze Mausefamilie mit acht Mitgliedern nach und nach in der Falle landete, und wir sie alle ein Stück weit von unserem Haus entfernt im Wald aussetzen konnten, war der Beweis erbracht, dass es nicht nötig war, ungebetene Hausmäuse umzubringen.

(140011)

Weblinks:

[1] www.mausefallendorf.de/museum/

[2] www.elektor-magazine.de/140011

[3] www.abacom-online.de/html/lochmaster.html
(kostenlose Demo)

Bild 6. Die IR-LED wird über ein langes Kabel angeschlossen, das zur anderen Seite der Mausefalle verlegt wird.

eC-reflow-mate

Professioneller SMD-Reflow-Ofen für perfekt gelötete Platinen

Der "eC-reflow-mate" ist die perfekte Lösung, wenn es um das Löten von Platinen für Prototypen und Kleinserien mit SMD-Komponenten geht. Er hat einen großzügig bemessenen Innenraum, so dass mehrere Standard-Platinen gleichzeitig gelötet werden können. Zwei fest eingebaute Temperatursensoren und die neu konstruierten, nicht linearen Infrarot-Heizstäbe sorgen dafür, dass die Wärme im Innenraum homogen verteilt wird. Mit einem dritten, beweglichen Sensor kann die Temperatur der Platinenoberfläche oder die einer ausgewählten Komponente in die Steuerung einbezogen werden. Das Geschehen im Ofen lässt sich durch die Glasfront der Schublade jederzeit beobachten. Die Konstruktion des "eC-reflow-mate" hält auch intensiver Dauerbeanspruchung stand.

Der "eC-reflow-mate" ist ein kompromisslos konstruierter, universeller SMD-Lötofen – unverzichtbar für alle, die SMD-bestückte Platinen in Einzelanfertigungen oder Kleinserien entwickeln und produzieren.



Technische Daten:

- Betriebsspannung: 230 V/50 Hz
- Leistung: 3000 Watt
- Gewicht: 29 kg
- Abmessungen (B x H x T): 62,0 x 24,5 x 52,0 cm
- Heizung: Infrarot-Strahler kombiniert mit Umluft
- Bedienung: Funktionstaster und LCD am Ofen oder via USB mit PC-Programm
- Temperaturbereich: 25...300 °C (300 °C Spitzenwert und 260 °C beim Löten)
- Effektive Platinengröße: max. 350 x 250 mm
- Temperatursensoren: 2 feste interne Sensoren, 1 externer Sensor (im Lieferumfang)
- Mehrsprachige Menüführung

Besonderheiten:

- Infrarot-Strahler für homogene Temperaturverteilung
- Schublade fährt nach Lötprozess-Ende selbsttätig aus
- Fenster in der Gerätefront zur ständigen Sichtkontrolle

Preis:
2.995,00 Euro
(zzgl. MwSt. und Versand)

Weitere Infos & Bestellung unter www.elektor.de/ec-reflow-mate

Ultraschall-Entfernungsmesser

Präzise Messung bis zu vier Metern



Von Rolf Gerstendorf

Okay, ein Zollstock würde es auch tun. Aber bequemer und zeitgemäßer ist zweifellos eine „berührungslose“ Entfernungsmessung mit Ultraschall. Der Aufwand ist gering, die Kosten auch. Möglich wird dies durch einen intelligenten und dennoch preisgünstigen Ultraschallsensor aus Malaysia.

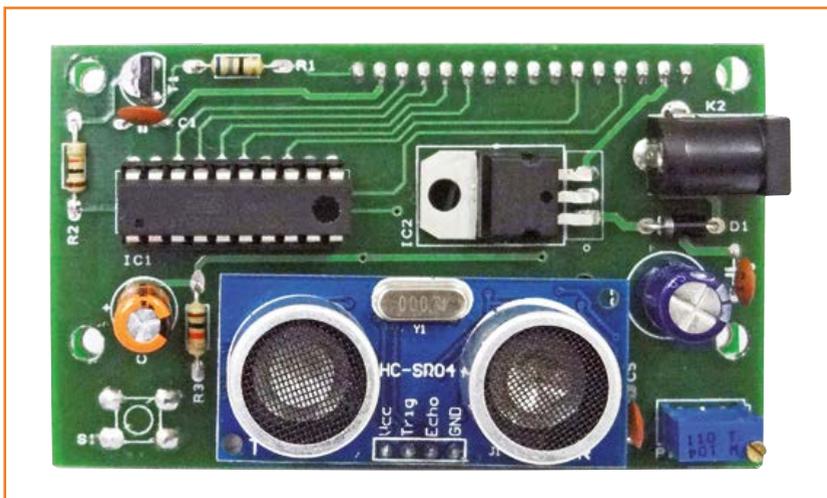
Der US-Sensor HC-SR04 wurde von der Firma Cytron Technology [1] entwickelt. Dieses Unternehmen ist auf Zubehör für Robotik-Projekte im edukativen Bereich spezialisiert. Der Sensor besteht nicht nur aus Ultraschall-Sender und -Empfänger, sondern bietet auch ausreichende „Intelligenz“, um ein einfaches Interfacing mit einem Mikrocontroller oder μ C-Board wie etwa dem Arduino zu ermöglichen. Doch nicht nur aus diesem Grund, sondern auch dank seines nied-

rigen Preises von etwa 4 € ist der HC-SR04 in Robotik-Kreisen sehr beliebt.

Der Sensor besteht aus einer 45x20 mm² großen Platine mit Sender und Empfänger (und einem 4-MHz-Quarz) auf der einen Seite (**Bild 1**) und den Signalverarbeitungs- und Controller-ICs auf der anderen. Das Datenblatt gibt für den HC-SR04 eine Reichweite von 2 cm bis 4 m bei einer Auflösung von 0,3 cm an. In einem Abstrahlwinkel von $\pm 22,5^\circ$ ist eine gute Empfindlichkeit des Sensors gegeben. Cytron verspricht im Datenblatt, dass „der Betrieb auch durch Sonnenlicht und bei schwarzen Materialien nicht beeinträchtigt wird“ (bei der Reflexion an schallschluckenden Materialien wie Kleidung sieht es allerdings anders aus).

Der HC-SR04 arbeitet mit einer Betriebsspannung von 5 V und zieht im Ruhezustand nur 2 mA; während des Betriebs sind es 15 mA. Die Spannungsversorgung erfordert zwei der vier Anschlusspins des Sensors, die anderen beiden sind der Triggereingang (an Pin 14 des Controllers) und der Echo-Ausgang (Pin 6) in TTL-Logik. Das Timing des Sensors ist in **Bild 2** zu sehen. Auf einen mindestens 10 μ s langen Impuls am Triggereingang erzeugt der Sensor sukzessive acht Bursts

Bild 1.
Die Vorderseite des
Ultraschall-Sensors.



von 40-kHz-Signalen, in den Burstpausen wartet der Empfänger auf die reflektierten Signale. Die Firmware des Sensors mittelt die Verzögerungen aus (und sorgt nebenbei auch noch für ein wenig Fehlerkorrektur) und erzeugt schließlich am Echo-Ausgang einen Impuls, dessen Länge im Verhältnis zur Verzögerung und damit zum Abstand zwischen Sender und reflektierendem Objekt steht. Als Faustregel gilt, dass die die Impulsdauer in Mikrosekunden geteilt durch 58 der Distanz in Zentimetern entspricht (und durch 148 in Zoll). So liegt die Impulsdauer bei der angegebenen Messspanne zwischen 150 μ s und 25 ms. Wird kein Hindernis aufgespürt, erscheint ein 38 ms langer Impuls am Ausgang.

Noch mehr Intelligenz

Wie in der Schaltung in **Bild 3** zu sehen und es auch kaum anders zu erwarten ist, besteht die Schaltung des US-Entfernungsmessers (abgesehen vom Sensor) fast nur aus einem kleinen Mikrocontroller und einem an Port B angeschlossenen LC-Display. Die „Eingabeeinheit“ ist ein simpler Taster mit Mehrfachfunktion. Ansonsten ist nur ein Trimpoti zur Einstellung des Kont-

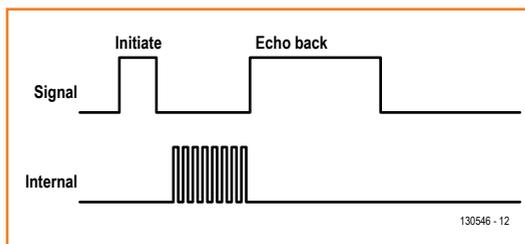


Bild 2. Timing-Diagramm der Sensor-Logik.

rasts sowie eine Spannungsstabilisierung mit dem altvertrauten 7805 und einer Verpolschutzdiode D1 zu sehen.

Die Software wurde zunächst für den ATtiny2313-Mikrocontroller mit der BASCOM AVR-Entwicklungsumgebung entwickelt. Diese erste Version kann die gemessene Entfernung nur in Millimetern ausgeben und besitzt keine „Luxus“-Eigenschaften. Wer möchte, kann diese Beta-Software von [2] herunterladen. Dann wurde die Software erweitert: Sie kann das Messergebnis auf Wunsch in Millimetern, Zentimetern, Zoll, Fuß und Yard (wer immer dies braucht....) mit einer Auflösung von zwei Nachkommastellen auf das 16X2-LC-Display zaubern und erlaubt

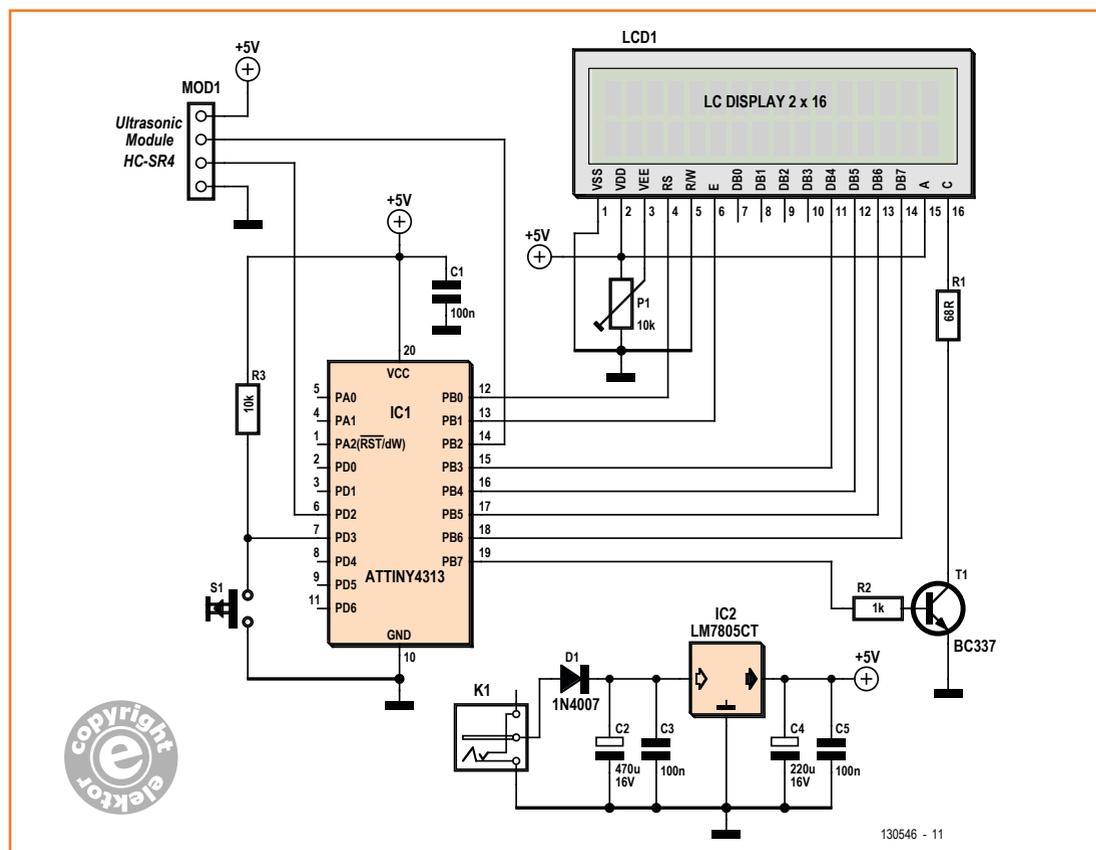


Bild 3. Ein Controller und ein Display. Größtenteils harmlos!



auch eine relative Abstandsmessung. Für diese Full-Version war der 2313 zu klein, als Alternative wurde nun der etwas größere, aber pinkompatible ATtiny4313 gewählt.

Welche Version auch immer, nach der Initialisierung des Controllers zu Beginn der gut kommentierten Software erzeugt die **Hauptschleife Dist** im Sekundentakt einen Triggerimpuls an Port B2 des Controllers, um den Sensor zu starten und einen Messzyklus zu initiieren.

Die **Interrupt-Routine Isr_lbl**, die auf eine steigende Flanke am Interrupteingang0 (PD2) reagiert, ermittelt anschließend die Länge des Ausgangsimpulses des Sensors und übergibt diesen Wert der Hauptroutine, die wiederum den Wert in das gewünschte Format umrechnet und auf dem LC-Display ausgibt. Das Timing beruht auf einer Quarzfrequenz von 8 MHz.

Daneben gibt es noch eine **Setup-Funktion**, in

der zwei Betriebsarten eingestellt werden können: Im **Calibrate-Modus** wird die aktuell gemessene Distanz zum Hindernis als Nullwert für die Messung des Sensors festgelegt. Der **Default-Modus** setzt alle Skalierungen zurück und legt die Sensoroberfläche als Nullpunkt fest.

Die Funktion **Timer 1** hat lediglich die Aufgabe, die Hintergrundbeleuchtung des LCDs zu aktivieren. Ein Druck auf den Taster schaltet die Beleuchtung ein, nach acht Sekunden verlischt die Beleuchtung aus Energiespargründen wieder, wenn die Taste nicht nochmal gedrückt wird.

Aufbau und Betrieb

Da die Elektronik auf dem LCD-Modul und dem Sensorplatinchen fertig bestückt ist, bleibt wenig Lötarbeit auf der Platine in **Bild 4** zu verrichten. Der Controller darf sogar in eine Fassung gesteckt werden (soviel Platz muss sein!), der Spannungsregler wird flachgelegt.

Das Projekt wurde mit Designspark [2] entwickelt, die Software für den Controller (Full-Version) kann kostenfrei von [3] heruntergeladen werden. Ganz stressfrei wird es, wenn man sich im Elektor-Shop den fertig programmierten Controller ATtiny4313 besorgt. Auch die (leere) Controller-Platine ist fertig geätzt zu haben [3].

Die Controllerplatine wird von Display- und Sensorplatine in die Mitte genommen (Titelbild). Das Display kann man mit kurzen Drahtstückchen

Stückliste

Widerstände:

R1 = 68 Ohm
 R2 = 1 k
 R3 = 10 k
 P1 = Mehrgang-Trimmpoti 10 k (Bourns 3296W-1-103LF)

Kondensatoren:

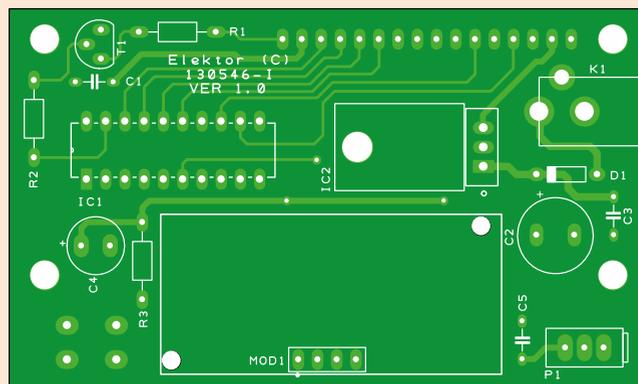
C1, C3, C5 = 100 n
 C2 = 470 µ/16 V, RM 10
 C4 = 220 µ/16 V, RM 7,5

Halbleiter:

D1 = 1N4007
 T1 = BC337-40
 IC1 = ATtiny4313-20PU, programmiert
 Elektor 130546-41 (siehe Text)
 IC2 = LM7805

Außerdem:

LCD1 = LCD-Modul 2x16 (Fordata FDCC1602N-FLYYBW51SE)



MOD1 = 4-poliger SIP-Verbinder passend zu US-Sensor HS-SR04
 K1 = Hohlstift-Buchse
 S1 = Taster (Alps SKHHLA010)
 Platine 130546-1

Bild 4.
 Die Controllerplatine ist in ein paar Minuten bestückt.

anschießen (oder man verwendet Reihenstecker/fassungen in passender Länge), für den Sensor ist eine vierpolige SIP-Buchse vorgesehen. Im Datenblatt des HC-SR04 wird ausdrücklich (in rot!) darauf hingewiesen, dass der Massepin **vor** Anlegen des VCC-Potentials angeschlossen sein muss. Das bedeutet: Erst den Sensor aufstecken und dann die Betriebsspannung einschalten oder - um sicher zu gehen - den VCC-Pin des Sensors um ein paar Millimeter kürzen (voreilende Masse). Dann kann man nichts falsch machen!

Beim Einschalten ist die Messung in Millimeter voreingestellt. Um dies zu ändern, drückt man kurz auf den Taster, die Einheit wechselt nach Zentimeter, Zoll, Fuß und Yard und wieder zurück zu Millimeter. Ein langer Druck auf den Taster versetzt die Schaltung in den Setup-Modus. In diesem Modus gibt es zwei Einstellungen: Man kann die Sensorvorderseite als Nullpunkt definieren (default) oder man setzt seinen eigenen Nullpunkt (calibrate). Im zweiten Fall misst der

Sensor die Distanz zu einem Hindernis, und mit einem langen Druck auf den Taster setzt der Controller (im Calibrate-Modus) diese Distanz als Nullpunkt. Mit einem kurzen Tastendruck kann man zwischen den Modi wechseln, ein langer Druck wählt die jeweilige Option aus.

(130546)

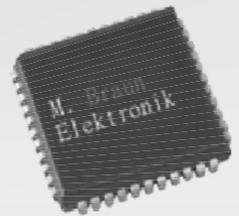
Weblinks

- [1] www.cytron.com.my/
- [2] www.rs-online.com/designspark/designshare/eng/projects/184/view/stage/design/
- [3] www.elektor-magazine.de/130546

Anzeige

Bestückung & Fertigung

M. Braun Elektronik



Tel.: 07457/731564

www.mbraun-elektronik.de

info@mbraun-elektronik.de

IO-Warrior von Code Mercenaries Anschluss an den USB

- I2C und SPI Interface
- bis zu 50 I/O Pins
- LCD Support
- LED- und Tasten-Matrix
- Verfügbar als Chip, Kit und Dongle
- Support für Windows, Linux und MacOS

Besuchen Sie uns auf der



electronics

11. - 14.11.2014

Halle A4, Stand 131/2

www.codemercenaries.de

Tel: +49(0)3379/20509-20

Signal für Wasserwaage

Von **Wouter Eisema**
(NL)

Eine Wasserwaage, die ein optisches und akustisches Signal ausgibt, wenn die Blase der Libelle in der Mitte steht, ist eine feine Sache. Der Signalgeber ist nicht nur nützlich, wenn Werbetransparente, großformatige Gemälde oder auch Küchenschranktüren aufgehängt werden müssen.

Er kann auch eine Hilfe für seh- oder hörbehinderte Menschen sein. Andere Einsätze, bei denen ein Blick auf die Wasserwaage nicht möglich ist, beispielsweise in der Fotografie, sind ebenfalls denkbar. Verglichen mit käuflichen optischen oder akustischen Wasserwaagen dürfte die hier vorgestellte Lösung wesentlich kosten-

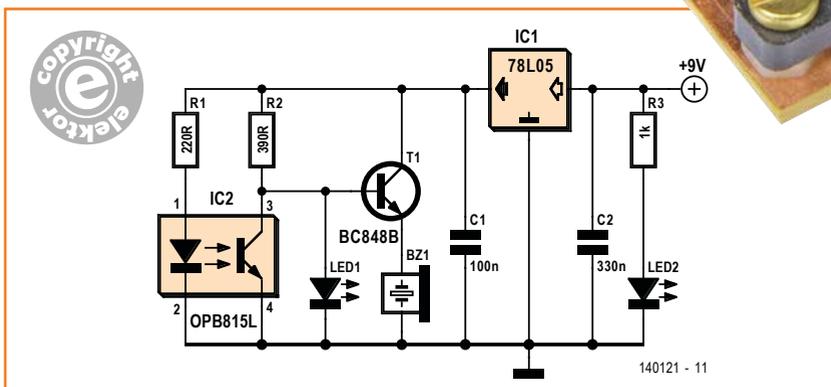


günstiger sein. Die Idee ist verblüffend einfach: Eine Infrarot-Gabellichtschranke erkennt den Stand der Blase in der Libelle. Der Abstand zwischen den Gabelschenkeln muss so breit sein, dass eine Libelle mit den Abmessungen 30 x 10 x 10 mm dazwischen passt. Der Fototransistor der Lichtschranke lässt die LED1 aufleuchten, gleichzeitig steuert er über T1 den akustischen Signalgeber BZ1.

Solange das Infrarot-Licht der Gabellichtschranke die Flüssigkeit der Libelle durchdringt, fällt auf den Fototransistor so viel Licht, dass er leitet. LED1 und BZ1 sind jetzt nicht aktiv. Steht jedoch die Blase in der Mitte der Libelle, wird das Infrarot-Licht so breit gestreut, dass der Fototransistor sperrt. LED1 und BZ1 signalisieren diesen Zustand.

Eine 9-V-Batterie liefert die Betriebsspannung, sie wird von einem Spannungsregler auf 5 V herabgesetzt. Wenn die Wasserwaage mit dem Signalgeber nur gelegentlich im Einsatz ist, hat die Batterie eine lange Lebensdauer. Die LED2 soll daran erinnern, dass das Gerät nach Gebrauch ausgeschaltet werden muss.

Der Autor hat eine eigene Platine für die Schaltung entworfen, die wenigen Bauelemente sind aber auch schnell auf ein Stück Lötspitzen-Platine montiert. Auf der originalen Platine haben einige Bauelemente SMD-Format. Die Libelle wird waagrecht zwischen die Schenkel der Gabellichtschranke geklemmt und mit Sekundenkleber fixiert.



Stückliste

Widerstände:

- R1 = 220 Ω (SMD 1206)
- R2 = 390 Ω (SMD 1206)
- R3 = 1 k (SMD 1206)

Kondensatoren:

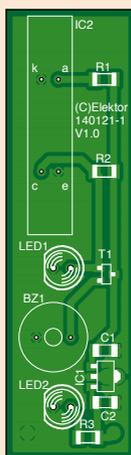
- C1 = 100 n (SMD 1206)
- C2 = 330 n (SMD 1206)

Halbleiter:

- LED1 = LED weiß, 5 mm
- LED2 = LED rot, 5 mm
- T1 = BC848B (SOT-23)
- IC1 = 78L05 (SOT-89)
- IC2 = IR-Gabellichtschranke, breiter Spalt, z. B. Optek OPB815L

Außerdem:

- BZ1 = Piezo-Tongebler aktiv, kleine Ausführung
- Libelle, kleine Ausführung, passend zur Gabellichtschranke
- 9-V-Batterie mit Anschlussclip
- Platinenlayout 140121-1, Download von [1]





In einem passenden Gehäuse finden der Schaltungsaufbau, ein Ausschalter und die 9-V-Batterie ihren Platz. Die Platine muss im Gehäuse so montiert werden, dass ihre Position nachträglich justierbar ist. Außen auf dem Gehäuse wird eine zweite Libelle angebracht, sie dient zur Kontrolle und als Referenz für die Libelle in der Gabellichtschranke.

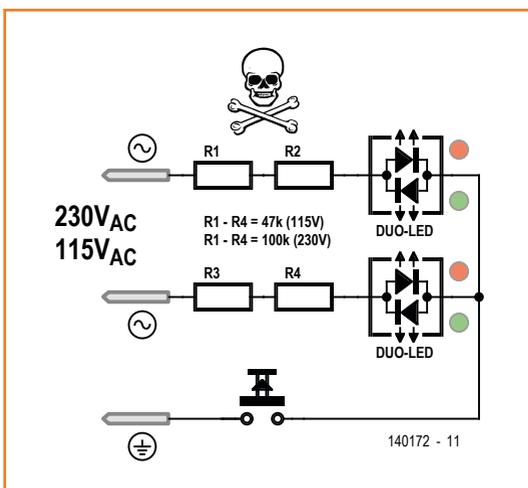
Das kleine Gerät ist bereits für sich allein eine Wasserwaage, es kann aber auch auf eine konventionelle Wasserwaage geklemmt werden und dort ergänzend als Signalgeber dienen.

(140121)gd

Weblink

www.elektor-magazine.de/140121

Steckdosentester mit Duo-LEDs



wegen der Spannungsfestigkeit), zwei Duo-LEDs und einem Taster.

Von
Hans-Norbert Gerbig
(D)

Das Ganze kann man einfach in einen Netzstecker mit Schutzkontakt einbauen. Achten Sie darauf, dass alles berührsicher eingebaut wird und der Typ des Tasters für 230 V geeignet ist.

Wenn der Steckdosentester an die Netzsteckdose angeschlossen wird, leuchten beide Duo-LEDs auf. Das aber nur wenn tatsächlich ein Strom von der Phase zum Nullleiter fließt. Die Farben der LEDs haben keine Bedeutung, es werden immer beide einzelnen LEDs leuchten, so dass sich eine orange Farbe ergibt. Wenn man Duo-LEDs verwendet, spart man sich aber zusätzliche Dioden, die anti-parallel geschaltet werden müssten.

Ein Phasenprüfer zeigt nur an, ob eine Spannung anwesend ist; an der Netzsteckdose findet er noch den Kontakt mit der Phase („statische Prüfung“). Dieser kleine Tester kann mehr. Er besteht aus nur vier Widerständen (jeweils zwei in Serie

Sobald der Taster betätigt wird, darf nur noch eine der Duo-LEDs leuchten, zum Zeichen, dass der Schutzleiter richtig angeschlossen ist. Die leuchtende Duo-LED zeigt die Phase an.

(140172)

Ziege, Wolf und Kohlkopf

Gute Rätsel sterben nie!



Ziege, Wolf und Kohlkopf ist eine Denkaufgabe, die der berühmte Abt von Tour, Alkuin, Lehrer und Berater von Karl dem Großen, eronnen und in seinem Werk *Propositiones ad acuendos iuvenes* [1] unsterblich gemacht hat.

Von **Mark Donners**
(Niederlande)

Wie der Titel dieses Werks schon andeutet, sollen solche Rätsel der geistigen Ertüchtigung der Jugend dienen, selbst wenn selbige schon etwas in die Jahre gekommen ist. Die Aufgaben in diesem Buch trainieren auf spielerische Weise Strategieentwicklung, Argumentation, kombinatorische Logik und potenzielle Problemanalyse. Das Rätsel Ziege, Wolf und Kohlkopf ist seit mehr als 1000 Jahren bekannt, nicht nur bei uns, sondern in ganz Europa!

Stellen Sie sich vor...

Stellen Sie sich vor, ein Bauer kauft auf dem Rätsel-Markt eine ausgewachsene Ziege, einen lebenden Wolf und einen großen grünen Kohlkopf, wie es ja im späten 8. Jahrhundert auch häufig vorkam. Nun steht er am Ufer des Flusses, an dem sein Bauernhof liegt. Leider am anderen Ufer. Sein Nachen ist aber zu klein, um die erstandenen Waren gleichzeitig über den Fluss nach

Hause zu bringen. Er kann immer nur eine(n) im Boot mitnehmen.

An dieser Stelle sollte man TV-Zappern, MP3-Hörern und Mausschubsern erklären, dass es sich bei einem Wolf (*canis lupus*) um ein Tier handelt, das, wenn am Flussufer unbeaufsichtigt, eine Ziege (*capra hircus*) durchaus nicht als Abendessen verschmähen würde. Aber die Ziege ist auch kein Kind von Traurigkeit, sie würde den Kohlkopf verschlingen, bevor der Bauer auch nur das Paddel ins Wasser getaucht hätte. Obwohl bei Wikipedia nichts zu diesem Thema zu finden war, darf angenommen werden, dass Wölfe keinen Kohl mögen und dass sich ein Kohlkopf weder am Wolf noch an der Ziege vergreifen würde. Und um es direkt klarzustellen: Weder Ziegen, Kohlköpfe noch Wölfe sind fähig, ein Boot erfolgreich zum anderen Flussufer zu rudern.

In der Vergangenheit wurde das Spiel im Stehen auf dem Schulhof und durch lautstarkes Argumen-

tieren gelöst. Später folgte ein grafischer Ansatz auf einer Tafel oder mit Buntstiften. Heute kann das Spiel mit intelligentem Silizium und hunderttausenden unsichtbar umhertollenden Bits gespielt werden.

Elektronik

Ein Blick auf die Schaltung in **Bild 1** zeigt nicht, dass - vorausgesetzt, die Stromversorgung ist über S8 und IC4 eingeschaltet - die im Mikrocontroller ATtiny2313 gespeicherte Firmware zuerst in einer Initialisierungsschleife überprüft, ob die LEDs in der Schaltung betriebsbereit sind, also richtig montiert und verlötet. Danach kann das Spiel beginnen: Zweimal drei LEDs stellen Ziege, Kohl und Wolf dar, die LED-Reihe im blauen Fluss deutet die Bemühungen des Bauern beim Skullen an. Das Ziel ist offensichtlich: Ziege, Kohl und Wolf müssen auf die andere Seite des Flusses, ohne dass irgendwem Schaden widerfährt. Die LEDs für Wolf, Kohl und Ziege werden vom Mikrocontroller über Strombegrenzungswiderstände angesteuert und deuten an, an welchem Flussufer sich wer aktuell befindet. So ist dem

Wolf beispielsweise das LED-Paar LED9 (linkes Ufer) und LED11 (rechtes Ufer) zugeordnet. Die Flussüberquerung deutet die Reihe LED1...LED8 an, die der ATtiny über seine Portleitungen PB0, PB1 und PB2 und einen 1-aus-8-Demultiplexer 74HC138 (IC2) steuert.

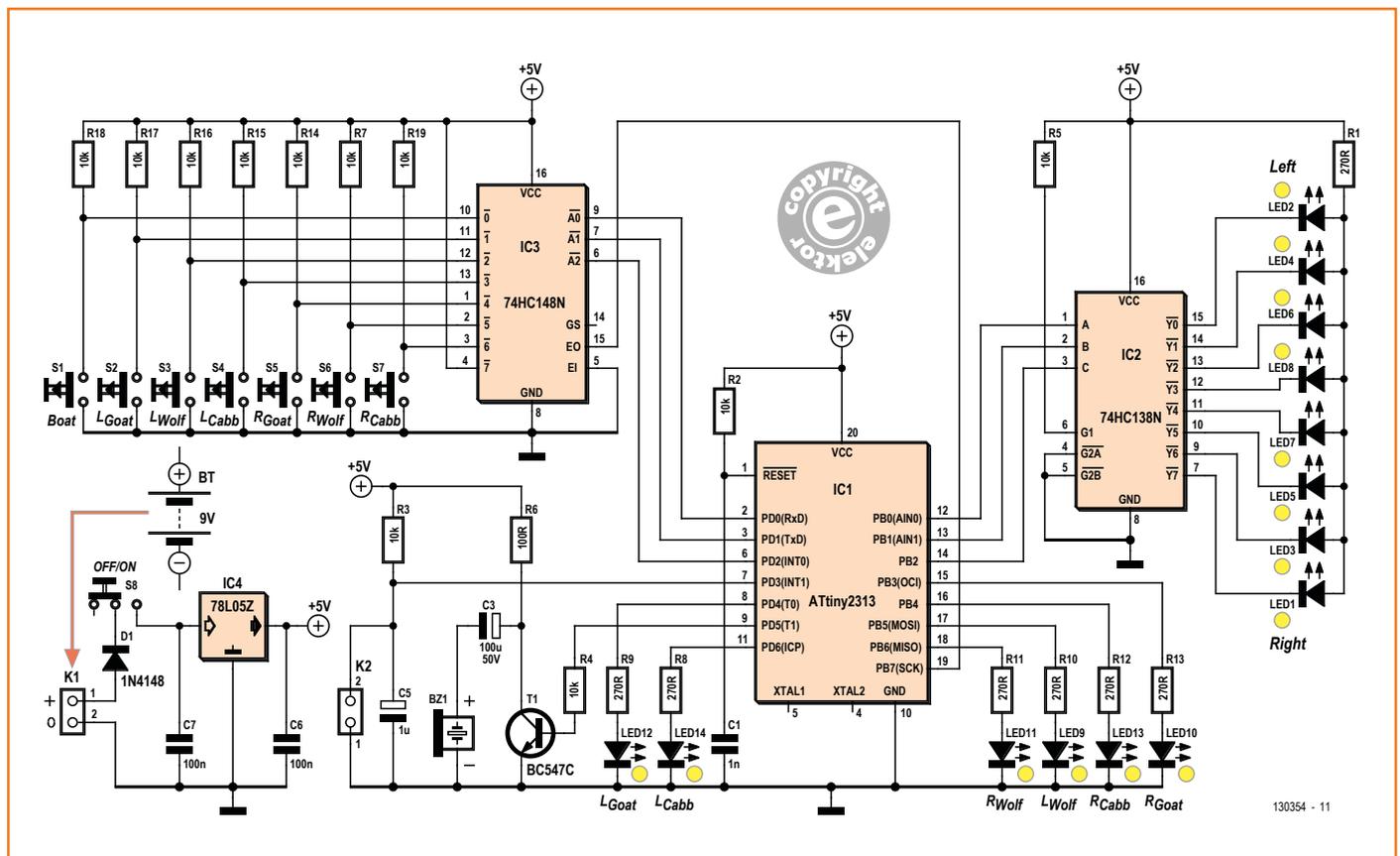
In dieser Anwendung werden keine hohen Anforderungen an die Geschwindigkeit und die Takt-Präzision des Mikrocontrollers gestellt, so dass der ATtiny2313 bei etwa 8 MHz mit seinem internen Oszillator arbeiten kann.

Aufbauen und programmieren

Der Bestückungsplan für die doppelseitige Platine ist in **Bild 2** zu sehen. Das Projekt ist garantiert SMD-frei und auch ein Kabelbaum ist nicht auszumachen, sind doch alle LEDs, Schalter und auch der Summer auf der Platine untergebracht. Lediglich für den Anschluss einer 9-V-Batterie oder einer anderen Energiequelle ist ein Kabel erforderlich.

Bevor man den Mikrocontroller in die Schaltung einsetzt, muss er natürlich programmiert werden. Dies aber nur, falls Sie nicht einen fertig program-

Bild 1. Das Spiel wird mit einem ATtiny2313-Mikrocontroller und einer Reihe von Drucktastern und LEDs gespielt.



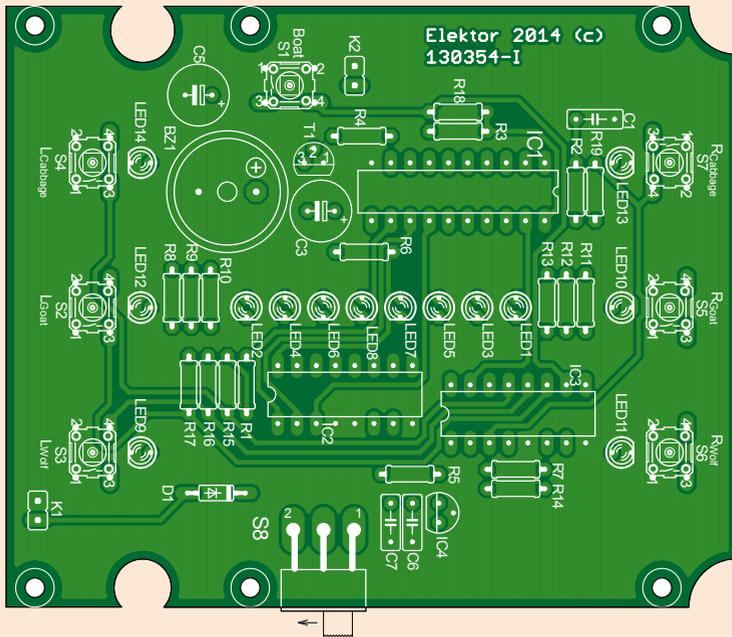


Bild 2. Bestückungsplan der Platine für das Spiel.

mierten Controller verwenden wollen, wie er bei Elektor (130354-41) erhältlich ist.

K2 wurde nur während der Projektentwicklung benötigt und hat nun keine Funktion mehr. Ein Vorschlag für eine künstlerische Gehäusegestaltung ist in **Bild 3** zu sehen. Was oben als rechtes und linkes Ufer benannt wurde, entspricht hier Süden und Norden!

Spielen

Entweder ein Tier oder der Kohlkopf kann zum Bootfahren über den Fluss in die gewünschte Richtung mit den Tastern S2...S7 ausgewählt werden. Wenn Sie den Landwirt ohne Fracht über den Fluss rudern wollen, drücken Sie die Taste S1 am Boot.

Wenn aber am Flussufer die Ziege auf dem Kohl herumkaut oder der Wolf geifernd die Ziege zerreißt, blinken die LEDs von Täter und Opfer und deuten damit an, dass die Situation „einer ernsthaften Überprüfung“ unterzogen werden muss. Auch ist ein trauriger Ton zu hören und das Spiel kann (muss) von vorne beginnen.

Stückliste

Widerstände:

(5%, 0,25 W)
 R1,R8...R13 = 270 Ω
 R2...R5,R7,R14...R19 = 10 k
 R6 = 100 Ω

Kondensatoren:

C1 = 1 n
 C6,C7 = 100 n
 C3 = 100 μ, 50 V radial
 C5 = 1 μ, 50 V radial

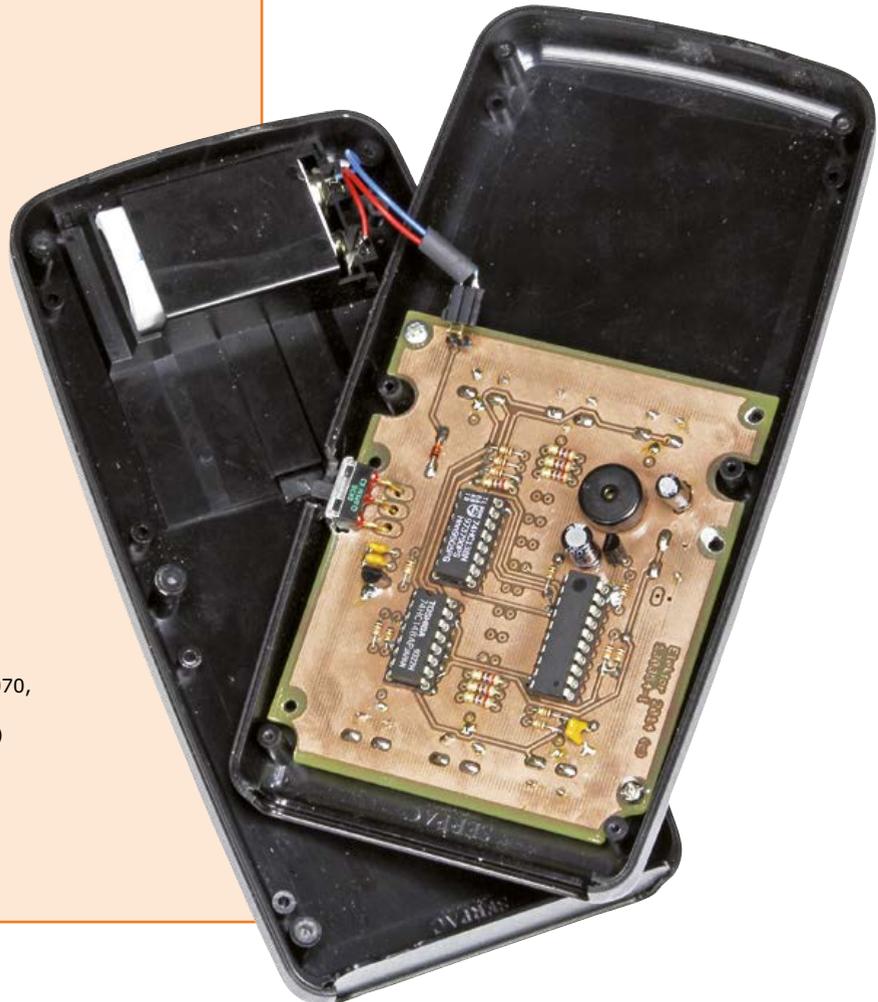
Halbleiter:

D1 = 1N4148
 LED1...LED14 = LED, 3 mm
 T1 = BC547
 IC1 = ATtiny2313-20PU, programmiert, Elektor 130354-41
 IC2 = 74HC138N
 IC3 = SN74HC148
 IC4 = MC78L05Z

Außerdem:

BZ1 = Buzzer, Kingstate (15027377)
 S1...S7 = Drucktaster für Platinenmontage, Omron B3F-1070, 6x6x9,5 mm (959698)
 S8 = Schiebeschalter, SPDT, für Platinenmontage (674357)
 K1 = 2-polige Stiftleiste
 K2 = nicht verwendet, optional
 9-V-Batterieclip
 Platine 130354-1

(Farnell-Bestellnummern in runden Klammern)



Listing 1. Auszug aus dem Programm

```
#include <avr/io.h>
#define F_CPU 8000000UL
#include <util/delay.h>

#define LGoat PD4
#define LCabb PD6
#define LWolf PB5
#define RGoat PB3
#define RCabb PB4
#define RWolf PB6
#define BuzzerPD5

#define statLGoat 0b00000001
#define statLCabb 0b00000010
#define statLWolf 0b00000100
#define statRGoat 0b00001000
#define statRCabb 0b00010000
#define statRWolf 0b00100000
#define statLBoat 0b01000000
#define statRBoat 0b10000000

/*
Inputs are as follows:

PIND&7 (0b00000111)

7 = S1 (Boat)
6 = S2 (LGoat)
5 = S3 (LWolf)
4 = S4 (LCabb)
3 = S5 (RGoat)
2 = S6 (RWolf)
1 = S7 (RCabb)
0 = not connected

*/
void setLeds(void);
uint8_t readInput(void);
void boatStart(void);
void boatContinue(void);
void playSound(uint8_t Sequence);
```

Wenn man aber Tiere und Gemüse unbeschadet auf das andere Ufer geschafft hat, ertönt eine triumphierende Melodie aus dem Summer BZ1. Auch der Bauer wird sehr ausgelassen und feiert seinen Erfolg mit ein paar Flussüberquerungen. Wir befürchten zwar, dass es dabei zu unwesentlichen Zwischenfällen kommen könnte - aber egal! Das C-Programm für den ATtiny steht zum kostenlosen Download auf der Elektor-Webseite bereit [2]. Ein Extrakt der Software finden Sie in **Listing 1**.

(130354)

Weblinks

- [1] www.instag.geschichte.uni-mainz.de/137.php
- [2] www.elektor-magazine.de/130354

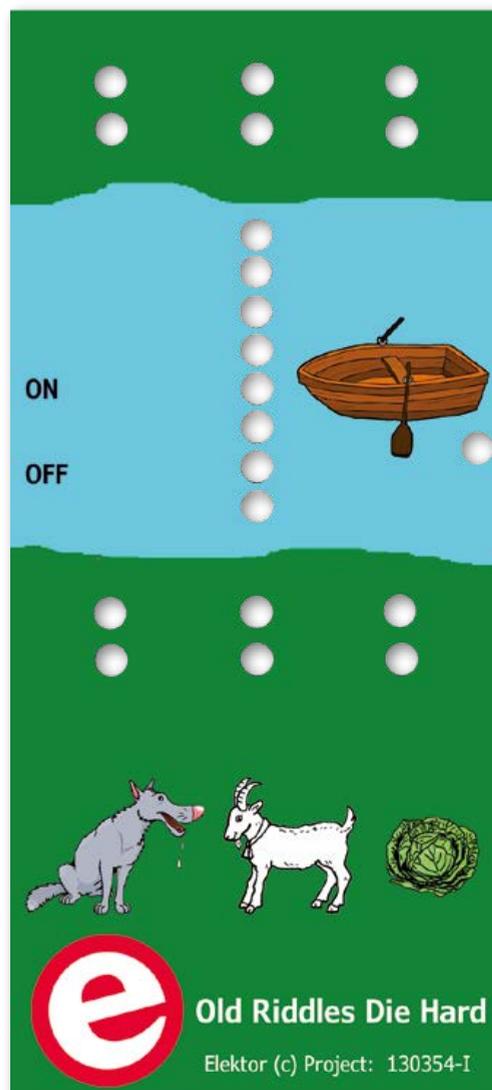
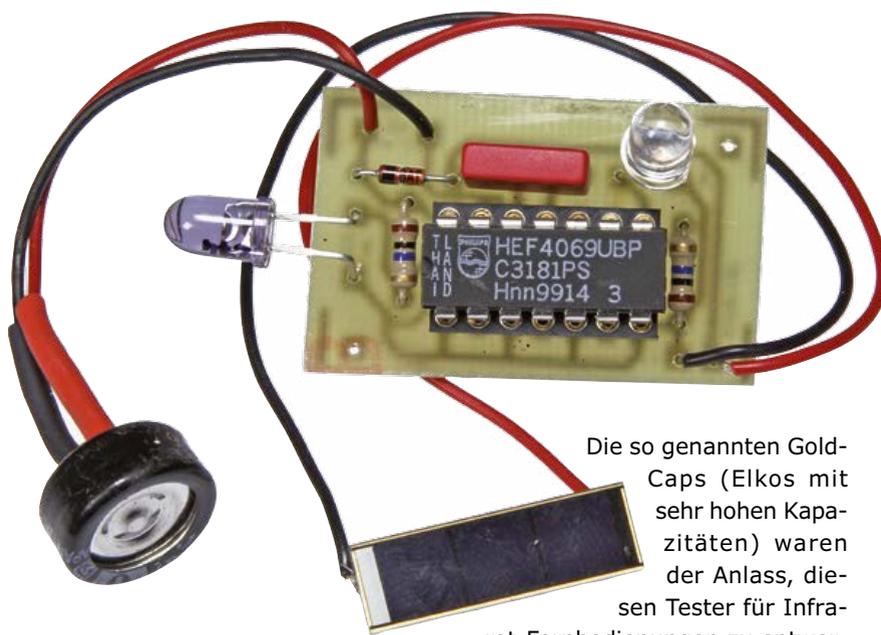


Bild 3.
Vorschlag für eine „Frontplatte“, passend für ein Gehäuse von Serpac H759V.

IR-Tester mit Solarzelle



Die so genannten Gold-Caps (Elkos mit sehr hohen Kapazitäten) waren der Anlass, diesen Tester für Infrarot-Fernbedienungen zu entwerfen.

Von **Wouter Eisema** (NL)

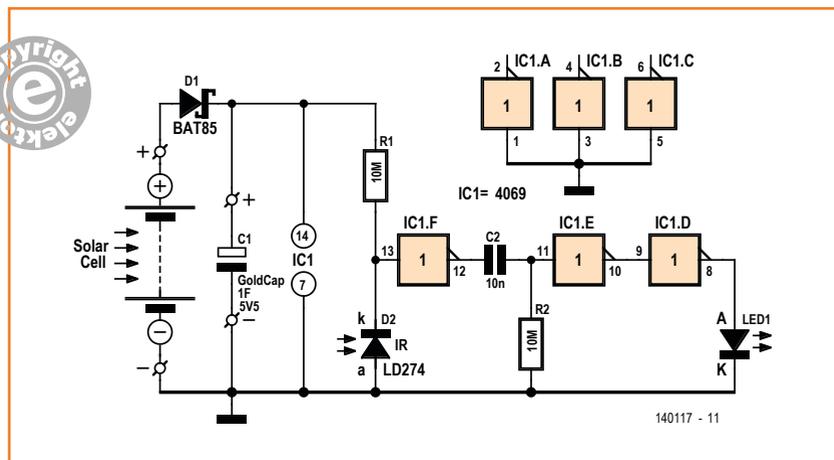
Sicher gibt es auch einfachere Methoden, IR-Fernbedienungen zu testen, doch die hier beschriebene Variante entbehrt nicht einer gewissen Originalität.

Der Strombedarf der Schaltung ist so gering, dass sie an einem GoldCap-Elko betrieben werden kann, der von einer Solarzelle geladen wird. Ein Ausschalter ist entbehrlich. Die Solarzelle lädt über Diode D1 den GoldCap-Elko C1, er hat bei bescheidenen Abmessungen die Kapazität 1 Farad! Eine Ladestrombegrenzung braucht

dieser Super-Kondensator nicht. Diode D1 verhindert, dass sich der Elko bei nur spärlichem Lichteinfall über die Solarzelle entlädt. Die Spannungsverluste an der Diode werden durch Einsatz einer Schottky-Diode BAT85 niedrig gehalten. Der Empfänger der Infrarot-Signale ist eine Infrarot-LED. Infrarot-LEDs können nicht nur infrarotes Licht emittieren, sie geben eine Spannung ab, wenn infrarotes Licht darauf fällt. Dazu muss die IR-LED (hier eine LD274) über einen Widerstand in Sperrrichtung mit der Betriebsspannung verbunden werden. Wenn das modulierte IR-Licht der zu testenden Fernbedienung auf die IR-LED fällt, ändert sich die Spannung am Eingang von CMOS-Inverter IC1F im Rhythmus des empfangenen Infrarot-Signals. Das am Inverter-Ausgang erscheinende Signal durchläuft einen Hochpass, bestehend aus C2 und R2. Der Hochpass schwächt niederfrequente Signale, so dass LED1 am Ausgang von IC1D durch kurze Impulse gesteuert wird. Auf einen Strombegrenzungswiderstand kann verzichtet werden, weil der Ausgang von Inverter IC1D bereits den LED-Strom begrenzt. Von den sechs Invertern des verwendeten CMOS-IC 4069 sind nur drei Inverter in Gebrauch. Die Eingänge der nicht benutzten Inverter sind an Masse gelegt.

Mit diesem Infrarot-Tester lassen sich die meisten Infrarot-Fernbedienungen auf Funktion testen. Solange ein beliebiger Taster der Fernbedienung gedrückt wird, leuchtet LED1 auf. Der Abstand zwischen der Infrarot-Lichtquelle und der IR-Sensor-LED darf nicht größer als etwa 20 cm sein. Auch ohne Fernbedienung ist die Funktion kontrollierbar. Wenn helles Tageslicht auf die IR-Sensor-LED fällt und diese dann plötzlich abgedunkelt wird, muss die Anzeige-LED (LED1) kurz aufleuchten.

Für den IR-Fernbedienungstester wurde eine kleine Platine entworfen, die Schaltung kann aber auch auf einer universellen Experimentierplatine oder einem Steckbrett aufgebaut werden. Die Solarzelle und der GoldCap-Elko werden mit der Platine über kurze Leitungen verbunden. Mit SMD-Bauelementen ist ein besonders kompakter Aufbau möglich. Als Schlüsselanhänger konstru-



iert und in ein transparentes Gehäuse eingebaut, ist er stets zu Hand.

(140117)gd

Weblink

www.elektor-magazine.de/140117

Stückliste

Widerstände:

R1,R2 = 10 M

Kondensatoren:

C1 = 1 F/5,5 V GoldCap

C2 = 10 n

Halbleiter:

LED1 = LED rot 5 mm, Low-current

D1 = BAT85

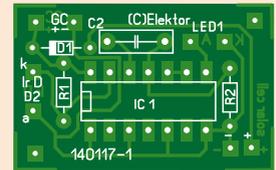
D2 = LD274 (Infrarot-LED)

IC1 = 4069

Außerdem:

Solarzelle ca. 22 x 7 mm, z B. Conrad 160839

Platinenlayout 140117-1, Download von [1]



Ersatz für 780X

Zu einem Spottpreis kaufte der Autor einen größeren Posten des Spannungsreglers LM317 ein. An einstellbaren Spannungsreglern bestand zwar kein konkreter Bedarf, in den hauseigenen Entwicklungen kamen nur Festspannungsregler des Typs 780X vor. Um den LM317 verwenden zu können, hätten die bereits produzierten Platinen modifiziert werden müssen. Deshalb wurden die Spannungsregler LM317 mit einem kleinen Stück Lötpunktraster-Platine, zwei oder drei Widerständen, einem Elko und drei Kontaktstiften zu Festspannungsreglern der Reihe 780X umfunktioniert.

Ein nebenstehendes Bild zeigt die Standardschaltung des LM317, dort ist

$$V_{out} = 1,25 \cdot (1 + R2/R1).$$

In der Tabelle sind für den LM317 die Widerstandswerte bezogen auf gängige Ausgangsspannungen von Festspannungsreglern angegeben. In der Spalte für R2 stehen die exakten Werte. Meistens genügt es, den nächstgelegenen Wert der E96-Reihe zu nehmen. Höhere Genauigkeiten lassen sich erreichen, wenn der exakte Wert durch Reihenschaltung von zwei Widerständen angenähert wird.

Dem Autor ging es insbesondere darum, dass die Konstruktion einschließlich der Anschlüsse mit den Spannungsreglern der 780X-Reihe übereinstimmt. Der Elko muss nicht unbedingt vorhanden sein. Wenn der Raum für den Ersatz knapp bemessen ist, kann er weggelassen werden.

(140174)gd

Von **Johnny Verhoeven** (NL)

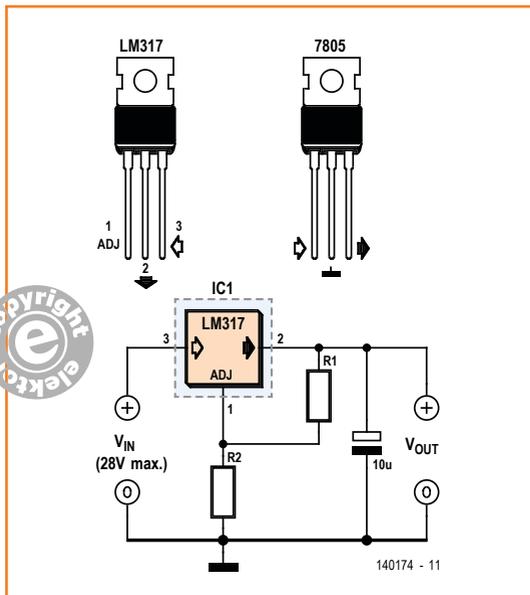


Tabelle 1. Widerstandswerte für gängige Ausgangsspannungen.

U _{out}	R1	R2
5 V	470 Ω	1410 Ω
6 V		1786 Ω
9 V		2914 Ω
12 V		4042 Ω
15 V		5170 Ω

Punktdisplay-Treiber

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

Manchmal ist es sinnvoll, den Dynamikbereich eines Signals in wenigen groben Teilbereichen darzustellen. Als Beispiel sei ein Indikator genannt, der nicht die genaue Drehzahl eines Motors anzeigt, sondern nur ein paar Werte wie *Aus*, *Langsam*, *Mittel*, *Schnell* und *Zu schnell*.



Solche Signale können leicht durch Mittelung (das heißt, einer Tiefpassfilterung) der Amplitude, Frequenz oder Pulsbreite des zu beobachtenden Signals gewonnen werden.

Für solche Schaltungen bietet sich der immer noch sehr beliebte LM3914 mit seiner Punkt-Skala an (auch als NTE1549 erhältlich, außerdem gibt es noch die Geschwister LM3915 und LM3916). Alternativ lässt sich ein Mikrocontroller mit einem analogen Eingang und entsprechender LED-Ansteuerung verwenden. Der LM3914 ist zwar einfach anzuwenden, erlaubt aber kaum eine Kontrolle über die Quantisierung des Eingangssignals. Ein Mikrocontroller gibt Ihnen zwar alle Flexibilität, die Sie sich wünschen können, aber Sie müssen ein Programm schreiben und dies im Controller unterbringen. Die hier vorgestellte Schaltung ermöglicht eine beliebige Quantisierung des Eingangssignals und muss nicht programmiert werden, sieht man von der Berechnung einiger Widerstandswerte ab.

Die Schaltung, die sich wie ein LM3914 mit Punktdisplay (dot display) verhält, ist unkompliziert. Ein Spannungsteiler ist so dimensioniert, dass alle in Frage kommenden Bereiche abgedeckt sind. Jeder Spannungsteiler-Ausgang wird mit dem Eingangssignal verglichen; und wenn das Eingangssignal die Referenzspannung überschreitet, wird der dazu gehörende Komparatorausgang high. Das ist alles Standard, aber noch nicht gut genug, da die LED-Skala jetzt nicht im Punkt-, sondern im Balken-Modus arbeitet.

Soll aber stets nur eine LED leuchten, bedarf es

weiterer Bemühungen. Die höherwertigen Komparatoren müssen nicht nur die LED ansteuern, sondern gleichzeitig dafür sorgen, dass die niedrigwertigeren Komparatoren irgendwie gesperrt oder deaktiviert werden. In dieser Schaltung steuert deshalb jeder Komparator auch einen PNP-Transistor an. Also, wie funktioniert das? Wenn zwei aufeinanderfolgende Komparatorausgänge low sind, sehen die Basis und der Emitter des Transistors zwischen ihnen die gleiche Spannung. Folglich sperrt der Transistor und die am Kollektor angeschlossene LED bleibt dunkel. Geht der untere Komparator-Ausgang auf High (und der obere bleibt Low), schaltet der Transistor durch, da die Basis/Emitter-Spannung hoch genug ist, und die angeschlossene LED leuchtet. Wenn nun der obere Komparator-Ausgang High wird, sind auch die Potentiale an Basis und Emitter wieder gleich, so dass der Transistor wieder sperrt und die LED verlöscht. Kurz gesagt, kann eine LED nur leuchten, wenn ihr eigener Komparator high ist und der darüber liegende low. Es gibt einen Haken in der Schaltung: Er steckt im Komparator-Ausgang oder, um genauer zu sein, in seiner Impedanz. Ich habe es mit dem Single-Supply-Rail-to-Rail (SS-R2R) Operationsverstärker TS924 probiert und alles funktionierte, wie es sollte. Ich fand allerdings heraus, dass dieser OpAmp obsolet und sein Ersatz im DIP-Gehäuse (TS924A) schwer zu bekommen ist. Also ersetze ich den Opamp durch einen anderen SS-R2R, den

LMC6464, den ich gerade zur Hand hatte. Zu meiner Überraschung leuchtete nun aber nicht nur eine LED, sondern auch die LED direkt darunter. Nicht hell, aber mehr als genug, um bemerkt zu werden. Die Ursache war wohl die relativ hohe Impedanz des LMC6464-Ausgangssignals, so dass die Spannung in die Knie ging, als der Ausgang ordentlich Strom (für die LED und den Transistor) zu liefern hatte. Deshalb war der High-Pegel gar nicht mehr so high, so dass am Transistor ein wenig Basis/Emitter-Spannung lag und er ein wenig durchschaltete.

Ein Blick in die Datenblätter der beiden Operationsverstärker erklärte dann alles. Der TS924A ist ein ziemlich cooler Operationsverstärker, den ich wärmstens empfehlen kann. Er hält sich auch mit so niederohmigen Lasten wie 600 Ω in seinen R2R-Grenzen, während der LMC6464 für eine Last von 25 k Ω oder höher ausgelegt ist.

Okay, der LMC6464 war keine gute Idee, aber wie wäre es mit dem guten alten LM324? Mit diesem Chip funktionierte die Schaltung fast perfekt, auch wenn die LEDs etwas weniger hell erstrahlten, da der High-Ausgangspegel rund 1,5 V unter der Versorgungsspannung des LM324 lag. Dies schuf noch ein Problem für die unterste LED (*Aus*), weil ich den Emitter des untersten Transistors an +5 V angeschlossen hatte. Das war zu viel für den Opamp-Ausgang, um die LED dauerhaft leuchten zu lassen. Zwei Dioden in Reihe mit dem Emitter lösten dieses Problem.

Die Eingangsstufe der Schaltung ist durch Experimentieren entstanden. Ich habe die Widerstandswerte so gewählt, wie es mir gerade in den Kram passte. Alles ist hier von *Ihrem* Eingangssignal abhängig. Ich habe es mit einem 3-V-/22-kHz-PWM-Signal ausprobiert und erhielt hervorragende Ergebnisse mit C1 = 220 nF und R15 = 2,2 k Ω . Die Platine wurde so ausgelegt, dass Elkos bis zu 10 μ F problemlos eingesetzt werden können. Dickere Elkos kann man über R15 und R14 schwebend montieren.

An P1 lassen sich die Ausgangspegel des Spannungsteilers skalieren. Beachten Sie, dass die Skala durch die identischen Widerstandswerte von R5 bis R8 linear ist, aber dies ist keineswegs zwingend. Anstelle der LEDs können Sie eine Stiftleiste montieren, um mit der Schaltung beispielsweise Relais zu schalten. Vergessen Sie aber in einem solchen Fall nicht die erforderliche Pufferung (um die oben beschriebenen Probleme zu vermeiden), etwa mit einem ULN2003.

(140111)

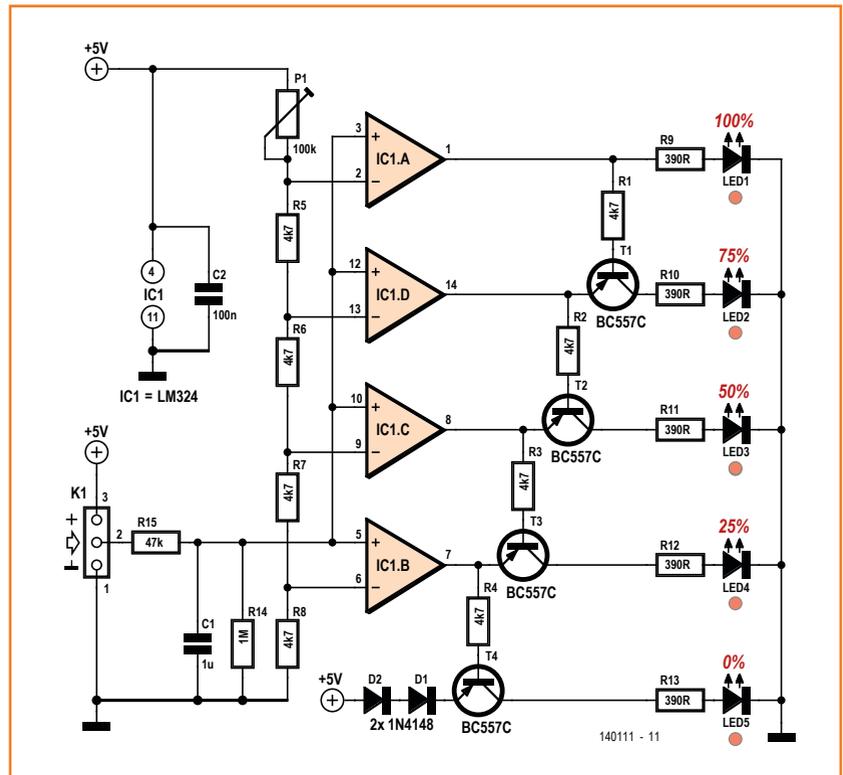


Bild 1.
Schaltung des
Punktdisplay-Treibers.

Weblink

- [1] www.elektor-labs.com/node/4013
- [2] www.elektor-magazine.de/140111

Stückliste

Widerstände:

(0,25 W)
R1,R2,R3,R4,R5,R6,R7,R8 = 4k7
R9,R10,R11,R12,R13 = 390 Ω
R14 = 1 M
R15 = 47 k
P1 = 100 k, Trimpoti

Kondensatoren:

C1 = 1 μ
C2 = 100 n

Halbleiter:

D1,D2 = 1N4148
IC1 = LM324
LED1..LED5 = LED, rot, 3 mm
T1..T4 = BC557C

Außerdem:

K1 = 3-poliger Pfostenstecker, RM 0,1"
Platine 140111-1 [2]

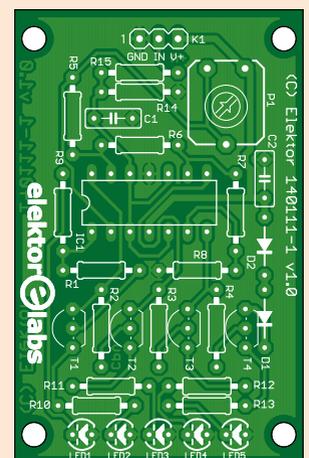
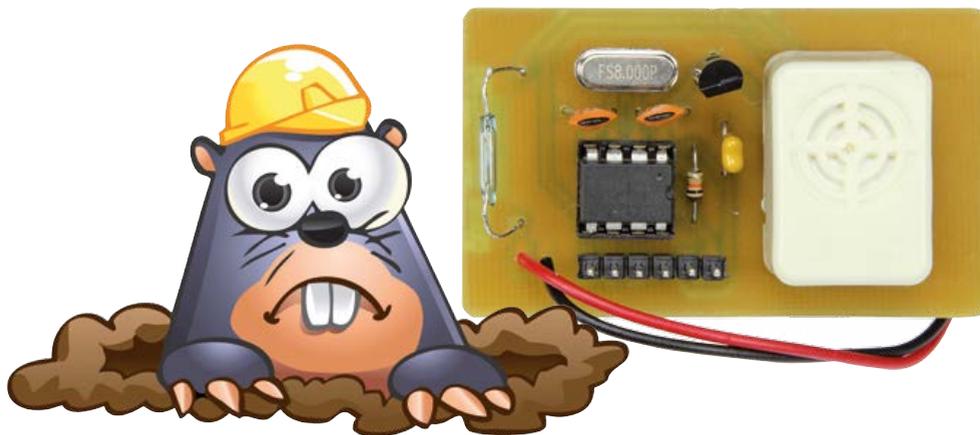


Bild 2.
Auf dieser Platine lässt sich der
Punktdisplay-Treiber bequem
aufbauen.

Maulwurfscheuche

Von **Jaap van Rijswijk**
(Südafrika)

Ein geplagter Gartenfreund trug an den Autor die Bitte heran, eine elektronische Maulwurfscheuche zu ersinnen, die ihren Namen wirklich verdient. Nun, auf Shopping-Plattformen im Web werden sie feil gehalten, die elektronischen Abwehrwaffen gegen die wühlwütigen Graber. Auch Beschreibungen der Technik sind nicht weit. Elektronische Maulwurfscheuchen geben im zeitlichen Abstand von 30 bis



40 Sekunden akustische Signale ab, die Frequenz liegt durchweg bei 400 Hz. Doch vertrauenserweckend erscheinen die käuflichen Produkte nicht, zumal in Rezensionen etwa die Hälfte der Anwender den Geräten Wirkungslosigkeit bescheinigt.

Die meisten Anti-Wühl-Systeme arbeiten mit dem Timer 555, einem Veteran unter den ICs. Dem wollte der Autor ein zeitgemäßeres Konzept entgegensetzen: Eine Maulwurfscheuche mit minimiertem Energiebedarf, die betrieben an einer Batterie, mehrere Monate ihrer Aufgabe nachkommt. Ein unkomplizierter Mikrocontroller aus der PIC-Familie wurde zum Zeitgeber gekürt, und was den Schallgeber betrifft, fiel die Wahl auf einen lautstarken, aktiven Beeper. Der integrierte Tongenerator erzeugt Signale mit einer Frequenz um 400 Hz, der Schalldruck des abgestrahlten akustischen Signals ist beträchtlich. Als Mikrocontroller kam der PIC12F675 gerade recht, der zufällig noch auf Lager war.

Die Firmware des Mikrocontrollers schaltet den Beeper im Abstand von 40 s für ungefähr 1 s ein, was einen gemeinen Maulwurf in einem Maß aufschreckt, dass er irritiert das Weite sucht. Mit einer durchschnittlichen 9-V-Batterie, Kapazität 550 mAh, ist die Maulwurfscheuche annähernd zwei Monate in Betrieb.

Zur Schaltung gehören der Mikrocontroller (IC1), der Beeper (SG1), ein Spannungsregler (IC2) und

wenige weitere Bauelemente. Quarz X1 sowie C1 und C2 sind die externen Komponenten des Taktoszillators, die Frequenz beträgt 8 MHz. Der interne Taktoszillator des PIC bleibt unbenutzt, weil sein relativ hoher Strombedarf die Batterie unnötig belasten würde. Widerstand R1 an Anschluss MCLR führt beim Einschalten einen Reset herbei. Spannungsregler IC2, ein LP2950, ist ein energiesparender Typ mit einem Ruhestrom von 75 μ A. Kondensator C3 entkoppelt die Betriebsspannung. Eingeschaltet wird die Maulwurfscheuche über den Reed-Kontakt S1. Bleibt noch die sechspolige Kontaktleiste SV1, sie ist die Schnittstelle zum Programmieren des Mikrocontrollers. Ein geeigneter Programmer ist der PICKit3.

Die Elektronik muss in ein feuchtigkeitsdichtes Gehäuse eingebaut werden, zum Beispiel in ein Marmeladenglas mit Schraubverschluss. Wenn ein Magnet am Gehäuse nahe dem Reedkontakt angebracht und mit Klebeband fixiert wird, ist die Maulwurfscheuche aktiv. Die Konstruktion wird dort im Erdboden versenkt, wo der Maulwurf mutmaßlich sein Unwesen treibt. Nach dem Einschalten gibt der Beeper für die Dauer von 1 s den 400-Hz-Ton ab. Es folgt eine Pause von 2,5

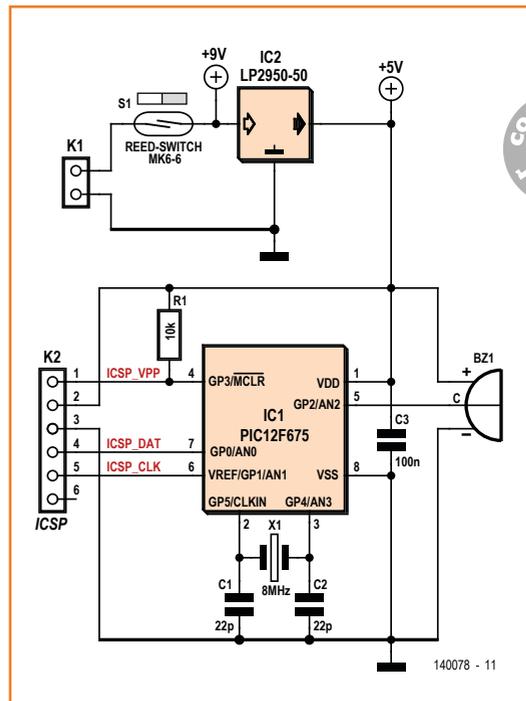
Minuten, genügend Zeit, die Maulwurfscheuche im Erdboden zu versenken. Anschließend tritt der Beeper im Abstand von 40 s für ungefähr 1 s akustisch in Aktion.

Die Firmware nutzt den internen Watchdog-Timer (WDT, Basis-Timeout 18 ms), um den Mikrocontroller mit Timeout-Interrupts in Intervallen kurzzeitig aus dem Sleep-Mode zu wecken. In diesem Zustand beträgt der Strombedarf nur etwa 150 μ A. Der WDT ist als 1:128-Vorteiler konfiguriert, so dass der Timeout im Abstand von 2,3 s auftritt. Ein Zähler registriert die Anzahl der generierten Interrupts. Nach 16 Interrupts wird Ausgang GP2 (Pin 5) für rund 800 ms auf hohe Spannung gelegt.

Für die Schaltung wurde eine Platine entworfen, auf der die Bauteile einschließlich Beeper ihren Platz haben. Der Schaltungsaufbau ist frei von Tücken, alle Bauelemente sind bedrahtet. Der Reed-Kontakt wird in einem kleinen Abstand zur Platine angelötet. Dann können die Anschlüsse gebogen werden, so dass sich der Reed-Kontakt am Platinenrand befindet. Die Firmware und das Platinenlayout stehen auf der Projektseite [1] zum Download bereit. Bereits programmiert ist der Mikrocontroller im Elektor-Shop erhältlich. Wie schon erwähnt, wird die Schaltung mit der Batterie in einem verschiebbaren Glas oder einem dichten Kunststoffbehälter untergebracht. Der Reed-Kontakt muss an der Innenwand anliegen, so dass er auf den von außen angelegten Magnet reagieren kann. Übrigens ist auch ein Reed-Kontakt verwendbar, der im Ruhezustand geschlossen ist. Dann muss der Magnet am Behälter anliegen, solange kein Bedarf für die Dienste der Maulwurfscheuche besteht.

Ein vielversprechender Einsatzort ist ein vom Maulwurf gegrabener Gang, beispielsweise unter einem Maulwurfhügel. Die Stelle, an der die Maulwurfscheuche vergraben ist, wird oberirdisch durch ein Fähnchen markiert. Anderenfalls kann es schwierig werden, die Batterie auszutauschen. Und das Ergebnis? Nach einer Betriebszeit von mehr als drei Monaten gab es Gewissheit, dass diese Maulwurfscheuche tatsächlich hält, was sie verspricht. Der Maulwurf hatte sein Tummelfeld an einen Ort außerhalb des eigenen Gartens verlegt. Dabei hatten die Bauelemente noch nicht einmal 5 € gekostet!

(140078)gd



Weblinks

[1] www.elektor-magazine.de/140078

Stückliste

Widerstände:

R1 = 10 k

Kondensatoren:

C1, C2 = 22 p

C3 = 100 n

Halbleiter:

IC1 = PIC12F675 -I/P (programmiert: 140078-41)

IC2 = LP2950-50

Außerdem:

X1 = Quarz 8 MHz

K1 = Stiftkontaktleiste 2-polig, Raster 2,54 mm

K2 = Stiftkontaktleiste 6-polig, Raster 2,54 mm

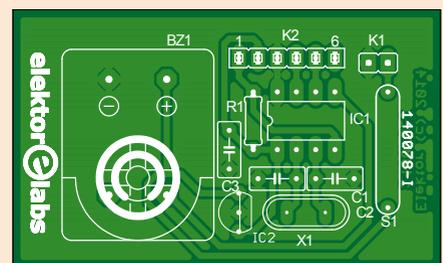
S1 = Reed-Kontakt, Schließer oder Öffner (siehe Text)

BZ1 = Beeper aktiv 400 Hz, 3 Anschlüsse, z. B. EMS-06-4TP (Reichelt)

9-V-Batterie mit Anschlussclip

Kleiner Permanentmagnet

Platine 140078-1, siehe [1]



NiCd-Akku-Regenerierer

Von **Ian Field** (UK)

Die hier vorgestellte Schaltung funktioniert sehr gut bei NiCd-Akkus. Mit NiMH-Akkus sind die Resultate nicht wirklich eindrucksvoll. Dennoch kann man damit auch Nickel-Metall-Hydrid-Akkus ein neues Leben einhauchen, wenn diese zu schwächeln beginnen.

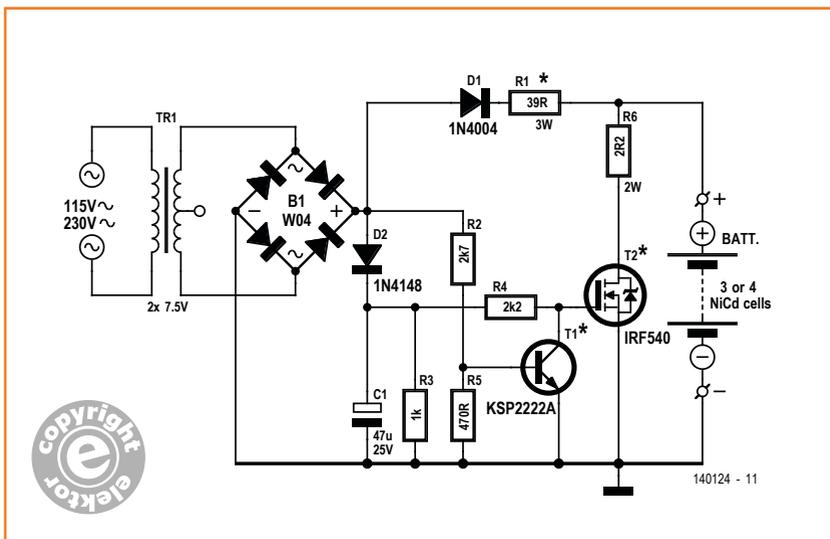


Vermutlich haben auch Sie noch ein paar NiCd-Akku vorrätig für die Fälle, wo kurzfristig hohe Ströme benötigt werden. Vielleicht haben auch Sie schon einmal überlegt, ein Schaltnetzteil in einen gepulsten Lader umzubauen? Diese Ladetechnik bringt aber nur bei NiCd-Zellen etwas. Bei NiMH-Akkus wirkt dieser Trick nur mäßig. Außerdem ist die Modifikation eines Schaltnetzteils nicht trivial und unter Umständen sogar gefährlich, selbst wenn man ein passendes Umbauobjekt zur Verfügung hat.

Die Grundlage für dieses Projekt war ein normaler Trafo aus einem ausgedienten NiCd-Ladegerät. Der Trafo hatte eine 15-V-Sekundärwicklung mit Mittelanzapfung. Damit wurde mit zwei Dioden eine Vollweggleichrichtung realisiert. Ein Satz Vorwiderstände erlaubte so im ursprünglichen „dummen“ Lader das parallele Laden von vier NiCd-Akkus auf einmal.

Jetzt aber wurde ein Brückengleichrichter verwendet, um die doppelte Spannung von nun 15 V zu erhalten, mit der man die Zellen in Serie schalten kann. In meinem Prototypen lade ich damit drei Zellen in Serie. Aber die Schaltung kann sicherlich auch ohne Änderung mit vier Zellen in Serie umgehen. Auch mit zwei Zellen wird es klappen, doch sollte man dann den Vorwiderstand R1 auf 56 Ω erhöhen.

Die Regenerierungstechnik für NiCd-Zellen besteht darin, dass man eigentlich mit einem normalen Konstantstrom lädt, der periodisch von nadelförmigen hohen Entladeströmen unterbrochen ist. Diese Methode sorgt dafür, dass sich beim Laden keine großen Kristalle an den Elektroden bilden, und die höhere Oberfläche kleinerer Partikel sorgt für eine höhere Energiedichte. Die Schaltung selbst ist eine ausgesprochene Minimalversion. Es wäre möglich, mehrere MOSFETs zellenspezifisch einzusetzen, aber dann wäre auch ein Batteriehalter nötig geworden, mit dem man jede Zelle individuell und isoliert von den



anderen anschließen kann. Außerdem hätte man so auch getrennte Strombegrenzungen etc. pro Zelle verwenden müssen. Darüber hinaus wäre dann wieder eine niedrigere Sekundärspannung des Trafos günstiger.

Die schmalen Entladeimpulse erzeugt der MOSFET T2. Im Prototyp wurde hierfür ein gebauchter SMD-Typ von einer alten Platine geklaut. Ströme von bis zu 70 A und 20...30 V sind für solche FETs üblich. Man kann natürlich auch einen neuen FET verwenden ;-). Der Typ IRF540 wäre z.B. geeignet, braucht dafür aber einen kleinen Kühlkörper. Der MOSFET wird von T1 angesteuert, mit dem eine Art Nulldurchgangsdetektor plus Inverter realisiert wurde. Immer wenn eine Halbwelle den Nullpunkt erreicht, fehlt die Basisspannung, um T1 durchgeschaltet zu halten. Genau dann entsteht ein Gate-Impuls an T2 und dieser entlädt die Zellen kurz mit höherem Strom über R6 – was die Akkus frisch hält.

Der Puffer C1 wird beim Abschalten schnell über R3 entladen, damit beim Abschalten oder bei Stromausfall T2 nicht lange durchgeschaltet bleibt und so den Akku leer zieht, was R6 überlasten würde. Die Speicherzeit von C1 reicht aber dazu aus, um im Nulldurchgang über R4 die nötige Gate-Spannung für T2 zu liefern.

D1 verhindert, dass die Spannung der Akkus die pulsierende Gleichspannung nicht auf 0 V kommen lässt und rückwärts über D2 ans Gate von T2 gelangt.

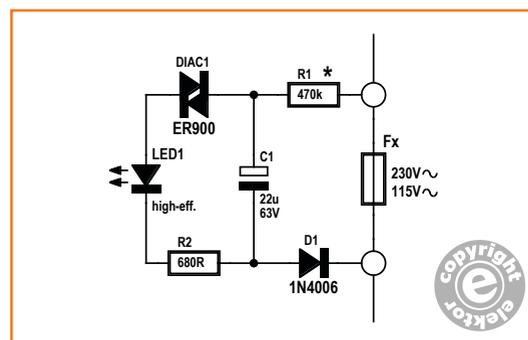
Der Typ KSP2222A für T1 ist eine Hersteller-Bezeichnung für den Typ PN2222A, was für die TO92-Kunststoff-Ausführung des Typs 2N2222A im TO18-Metallgehäuse steht. Ein BC337 oder der japanische Typ 2SC1815 tun es genau so gut. Selbst die gebräuchlicheren Typen 2SC945 oder BC547 funktionieren, werden aber möglicherweise etwas wärmer.

(140124)

Sicherungswächter

Diese kleine Schaltung mit nur sechs Komponenten zeigt durch Blinken einer LED an, wenn eine Netzsicherung ausgefallen ist.

Und das funktioniert so: Wenn die Netzsicherung in Ordnung ist und leitet, passiert natürlich nichts, weil der Sicherungswächter spannungslos ist. Wenn die Sicherung durchbrennt, lädt sich der Elko von 22 μ F durch die Netzspannung über den Vorwiderstand von 470 k Ω und die Diode 1N4006 auf. Solange die Kondensatorspannung noch unter der Zündspannung des DIACs liegt, ist dieser gesperrt. Erreicht die Spannung des Elkos nach einiger Zeit die Zündspannung des DIACs, wird dieser leitend und der Entladestrom fließt über die Leuchtdiode. Der Widerstand von 680 Ω begrenzt den Strom auf einen für die LED ungefährlichen Wert. Sinkt nun der durch den DIAC fließende Strom unter seinen Haltestromwert, sperrt dieser und der Elko kann sich wieder erneut aufladen. Der Zyklus beginnt dann von neuem. Die Schaltung kann einfach auf einem kleinen Stück Experimentierplatine aufgebaut werden. R1 muss ein Typ sein, der für mindestens 325 V geig-



net ist ($230\text{ V} \times \sqrt{2}$, ggf. zwei Rs in Serie schalten). Das Sicherungswächter-Modul wird dann parallel zu der zu überwachenden Netzsicherung geschaltet. Bei eingeschaltetem Gerät und intakter Sicherung ist die Schaltung kurzgeschlossen und damit inaktiv. Anders bei Sicherungsausfall: Die DIAC-Blinkschaltung erhält Strom, wird aktiv und die LED blinkt. Da die Schaltung nicht galvanisch vom Netz getrennt ist, muss sie berührungssicher eingebaut werden.

(130420)

Von
Hans-Norbert Gerbig
(D)

Mehr Strom für USB-Lithium-Lader

Von **Ian Field** (UK)

Mit einem Parallelregler wie dem TL431 kann man sicherstellen, dass eine Lithium-Akkuzelle nie auf mehr als z.B. 4,2 V geladen werden kann. Wenn nämlich die Maximalspannung erreicht wird, sorgt der TL431 dafür, dass aller Strom über einen Widerstand abgeleitet wird. Ein Wert von 10 Ω begrenzt diesen Strom auf einen Wert, den ein TL431 vertragen kann.

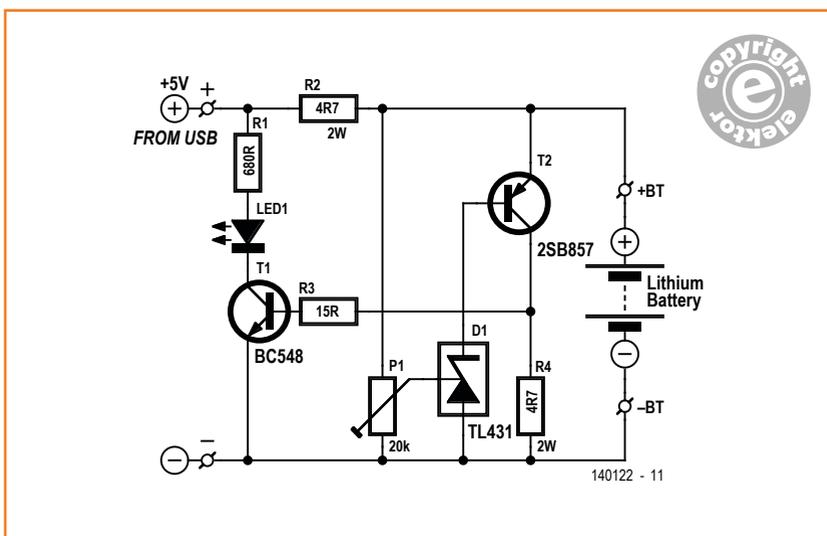


Eine andere Lösung ist, über einen PNP-Transistor eine Art Komplementär-Darlington mit dem TL431 zu realisieren. So kann man den Maximalstrom des TL431 von 100 mA im Prinzip mit der Stromverstärkung des

PNP-Transistors multiplizieren. Da nun der Maximalstrom des TL431 keine Begrenzung mehr darstellt, hat man es nur noch mit dem Maximalstrom zu tun, den ein korrekt arbeitender USB-Lader liefern kann – und dies sind 500 mA. Bezogen auf die niedrigste zulässige Spannung des Akkus ergibt sich der hierzu passende Widerstand zu 4,5 Ω . Der nächste Wert der E12-Reihe liegt bei 4,7 Ω .

In der Schaltung wurde für T2 der Typ 2SB857 eingesetzt, da er gerade zur Hand war. Es ist ein normaler PNP-Transistor mit $V_{CE(max)} = 50 V$, $I_{C(max)} = 4 A$ und $h_{fe(min)} = 60$. Man kann also praktisch jeden PNP-Typ im T220-Gehäuse mit ähnlichen Werten verwenden. Er braucht auch keinen Kühlkörper, denn sogar ohne Last (= mit maximalem Strom durch R4) wird er nur leicht warm.

Da nun die meiste Energie an R4 vernichtet wird, kann man an ihm eine Spannung abgreifen, die über T1 die LED1 aktiviert und so den Zustand: „Zelle voll!“ signalisiert. Ohne T2 und nur mit dem TL431 wäre das nicht so einfach möglich, denn ein Widerstand in der Anodenleitung des Reglers würde für einen schwankenden Arbeitspunkt sorgen. Der Widerstand in der Kollektorleitung von T2 aber hat keine Auswirkungen, da sein Einfluss vom TL431 ausgeregelt wird. Die Maximalspannung des Akkus lässt sich damit nach wie vor präzise mit P1 einstellen.



(140122)

Master-Slave-Steckdosenleiste

Mit nur einem einzigen Bauteil ist es möglich, eine Master-Slave-Steckdosenleiste zu bauen. Dafür brauchen wir nur einen Triac!

Das funktioniert so: Die volle Netzspannung von 230 V~ liegt an A1, der Master ist an der Steuerelektrode G (Gate) angeschlossen und der Slave an A2 eines Triacs (**Bild 1**).

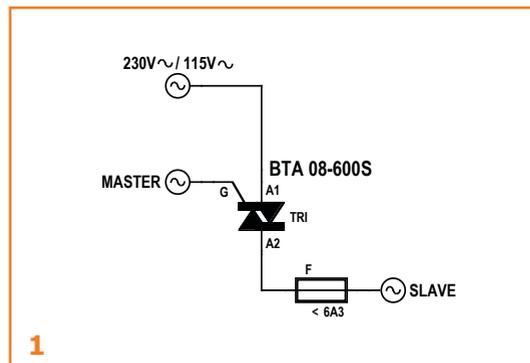
Sobald das Gerät am Master-Eingang eingeschaltet wird, läuft ein Strom von A1 nach G. Dadurch wird der Triac gezündet, womit das Gerät an A2 automatisch eingeschaltet wird. Sobald das Gerät am Master-Eingang ausgeschaltet wird, geht der Gate-Strom auf 0 zurück und alle Geräte werden ausgeschaltet.

Gate und A1 können vertauscht werden, dann liegen 230 V~ am Gate und A1 am Mastereingang. Der Effekt bleibt derselbe.

Bild 2 zeigt, wie man den Triac in einer Steckdosenleiste unterbringen kann, wobei drei Slave-Anschlüsse zur Verfügung stehen. Der Triac kann einfach auf eine kleine Experimentierplatine gelötet werden, zusammen mit einer dreipoligen Schraub-Anschlussklemme. In der Steckdosenleiste werden die Leitungen zwischen dem ersten Netzanschluss und den übrigen Anschlüssen getrennt; dazwischen wird dann das Triac-„Modul“ geschaltet (über einige kurze Kabel).

Zu empfehlen ist es, die Slave-Ausgänge mit einer T-Sicherung abzusichern. Da alle elektrischen Verbindungen Netzspannung führen und nicht galvanisch voneinander getrennt sind, muss die Schaltung berührungssicher eingebaut werden. Der Schutzleiter ist durchgängig von der Eingangs- bis zu den Ausgangssteckdosen zu verlegen.

Von
Hans-Norbert Gerbig
(D)



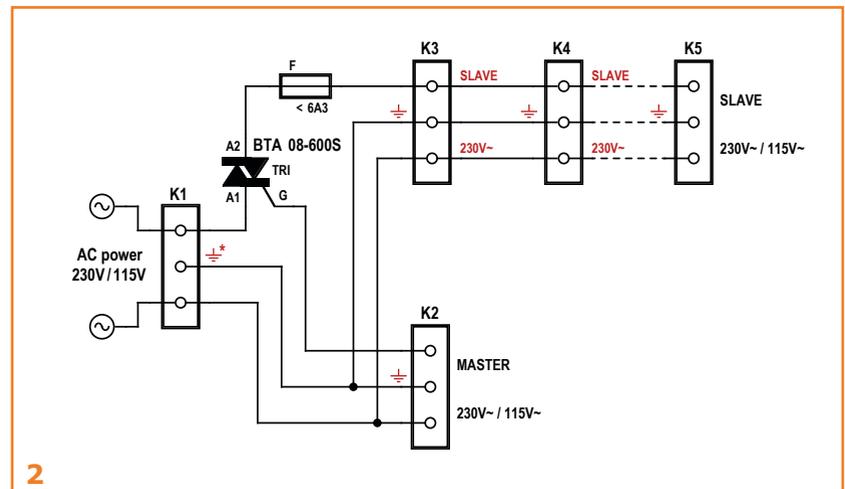
1

Der Typ des Triacs richtet sich nach dem gesamten Stromverbrauch des Masters und der Slaves. Der Triac-Typ BTA08-6008 (600 V/8 A) benötigt einen sehr geringen Zündstrom und eine sehr geringe Zündspannung.

(130419)

Anmerkung des Elektor-Labors:

Dies ist ein netter Schaltungstrick, den wir bisher noch nirgends gesehen haben. Es ist aber die Frage, ob ein Triac für diese Verwendung geeignet ist. Auf dem Chip des Triacs ist die Gate-Fläche relativ klein und eigentlich nicht geeignet für größere Ströme. Leider geben die Triac-Datenblätter von verschiedenen Herstellern nur selten Auskunft über den maximalen Gate-Strom. Bei einem Hersteller fanden wir einen Maximalwert von 500 mA. Es ist also wichtig, dass das Master-Gerät nicht zu viel Strom aufnimmt.



2

Schnittstellen-Isolator

Galvanische Trennung vom PC

Von
Christophe Bourrier
 F4EZC QRV@wanadoo.fr

Funkamateure koppeln Sender, Empfänger oder Transceiver gern mit einem PC, so dass diese Geräte über den PC gesteuert werden können. Der Schnittstellen-Isolator trennt die serielle Schnittstelle des PC galvanisch vom PTT- oder CW-Eingang eines Amateurfunk-Senders (PTT = *Push To Talk*, CW = *Continuous Wave*), denn auch hier gilt: Sicherheit geht vor!

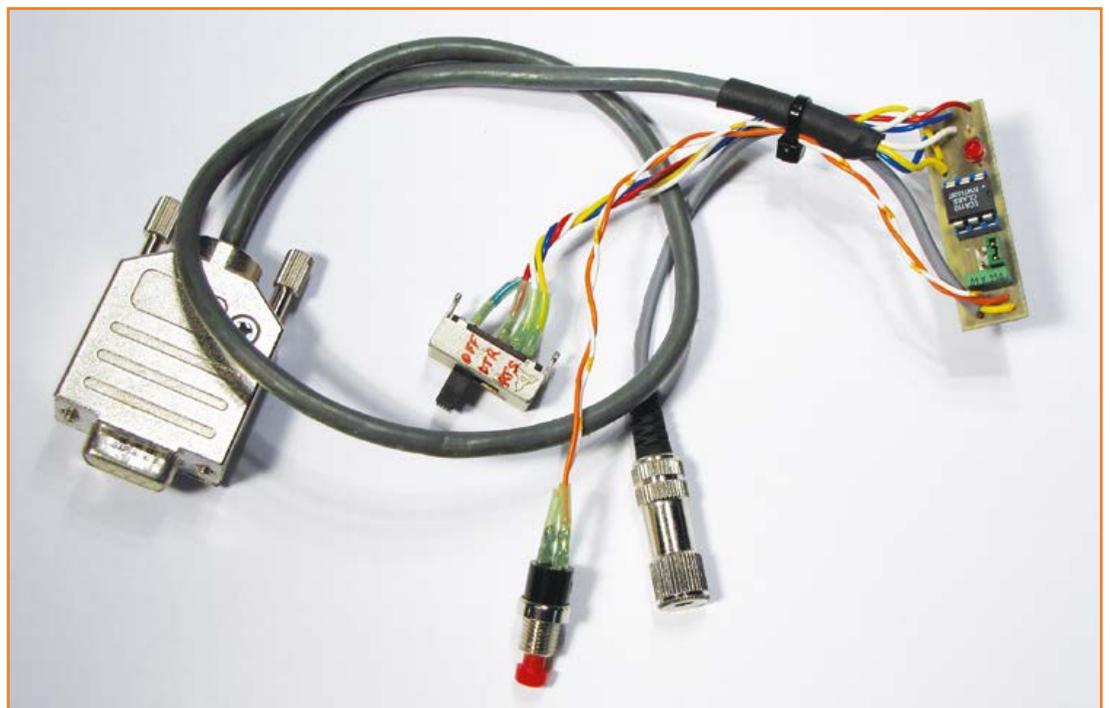
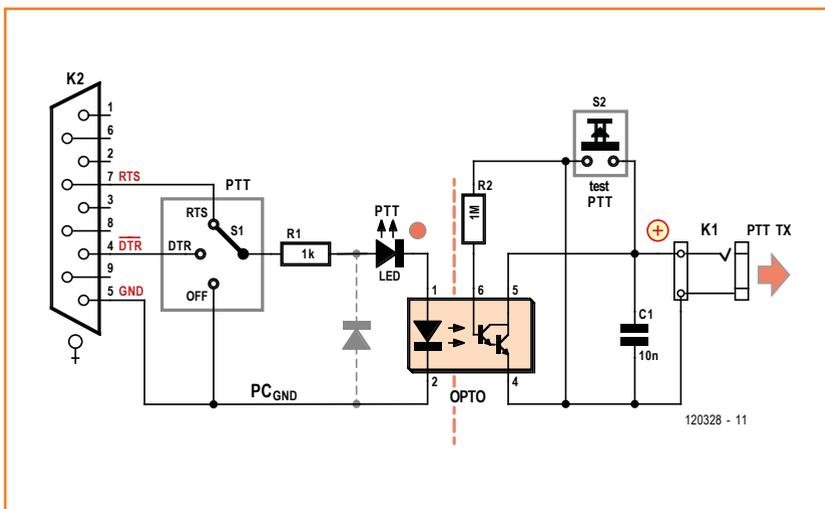


Bild 1.
 Der Optokoppler trennt den Sender galvanisch von der PC-Schnittstelle.



Ein PC erzeugt zahllose Störsignale, die im Sender Fehlfunktionen verursachen können, falls die Geräte über ein Kabel miteinander verbunden sind. Eine galvanische Trennung zwischen den Geräten mindert diese Gefahr drastisch. In anderer Richtung kann das HF-Signal des Senders nicht auf direktem Weg in den PC gelangen, um dort möglicherweise erheblichen Schaden anzurichten. Bei Versuchen, Fehler aufzuspüren und zu beseitigen, kann vieles schiefgehen: Eine falsche Einstellung, ein Antennenproblem, ein Kurzschluss in einem Koax-Stecker oder ein Kabelbruch, dies sind nur wenige potentielle Ursachen für ernste Schwierigkeiten. Wenn ein starker Sender seine hochfrequente Energie abstrahlt, während er mit der seriellen Schnittstelle eines am Stromnetz

Der Schnittstellen-Isolator trennt galvanisch Geräte vom PC.

betriebenen Laptops verbunden ist, können vagabundierende Ströme auftreten. Dabei können leistungsstarke hochfrequente Signale in den Laptop gelangen, die entweder nur die serielle Schnittstelle oder sogar die Hauptplatine beschädigen. Auch hier gilt die alte Regel: Vorbeugen ist besser als Verschrotten!

Der Schnittstellen-Isolator ist ein einfaches System mit nützlichen Eigenschaften: Zwei RS232-Varianten stehen zur Wahl, eine LED signalisiert den Betriebszustand und der Einsatz eines *optoMOS-Kopplers* ist optional. Steht der Schalter auf OFF, werden die Kommandos des PCs ignoriert.

Die Schaltung (**Bild 1**) folgt dem bewährten Muster, nur der Wahlschalter für die PC-Schnittstellen-Signale RTS/CTS und DTR/DSR ist nicht Standard. Die Signale lassen sich blockieren, zum Beispiel während der PC und das Betriebssystem hochfahren. In Reihe mit dem Wahlschalter liegen Widerstand R1, eine rote LED und der Eingang des Optokopplers. Die LED leuchtet auf, wenn der PC ein PTT- oder CW-Signal ausgibt.

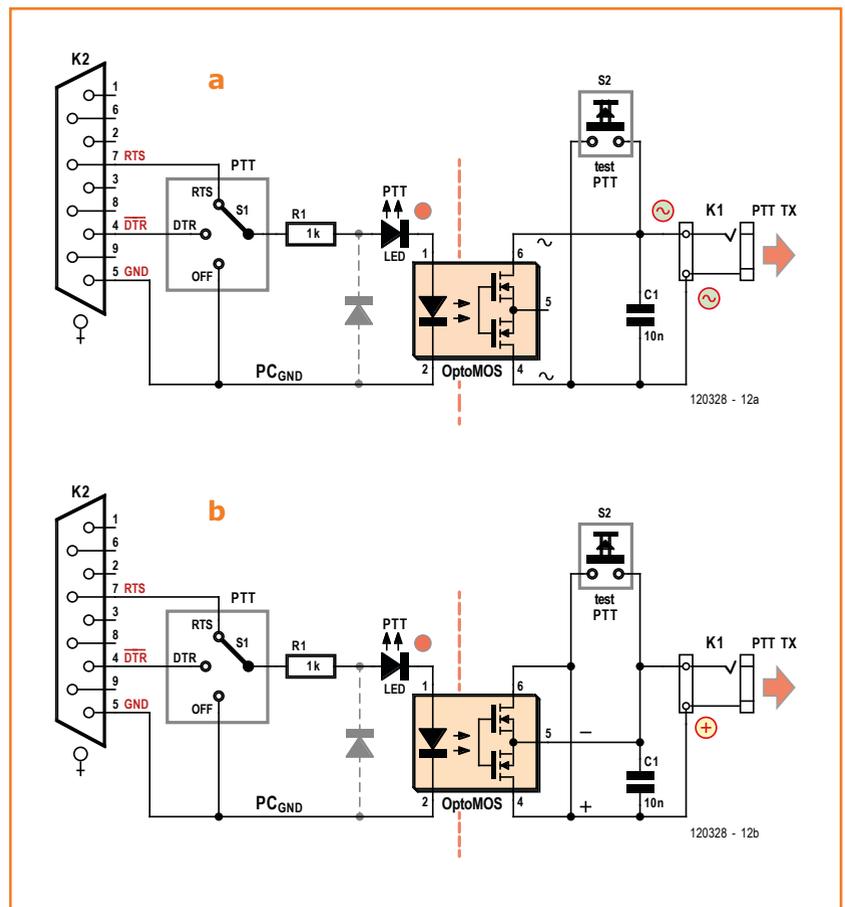
Als Optokoppler eignet sich ein kostengünstiger Typ mit Fototransistor, der Ausgang wird meistens von einem Darlington gesteuert. Hier hat die Emitter-Kollektor-Strecke eine Polarität, der Spannungsabfall beträgt mehr als 1 V. Für die meisten Anwendungen dürfte dies kein Problem sein. Der optionale Widerstand R2 sorgt dafür, dass die Basis korrekt eingestellt ist und unerwünschtes Schwingen vermieden wird. Kondensator C1 unterdrückt eine eventuelle Schwingneigung am Ausgang. Mit dem Drucktaster kann der Sender zu Testzwecken von Hand eingeschaltet werden, ohne Steuerung durch den PC.

Anstelle des Optokopplers ist auch ein Halbleiter-Relais mit MOSFET-Ausgang verwendbar. Diese Bauelemente werden von ihrem Hersteller Clare unter der Marke „optoMOS“ [1] vermarktet. In **Bild 2a** und **2b** sind optoMOS-Koppler in die Schaltung integriert. Die optoMOS-Typen haben den Vorteil, dass kein nennenswerter Spannungsabfall auftritt. Zwar schalten diese Bauelemente träger als klassische Optokoppler, doch das fällt hier nicht ins Gewicht. In Bild 2a ist eine für das Schalten von Wechselströmen geeignete Konfigu-

ration dargestellt. Der Ausgang hat keine Polarität, die Signale können positiv oder negativ sein. Bild 2 b zeigt die Konfiguration für das Schalten von Gleichströmen. Hier beträgt der Übergangswiderstand des MOS-Schalters weniger als 10 Ω , er ist niedriger als bei der Version für Wechselstrom. Zu beachten ist hier, dass die Polarität verglichen mit dem Fototransistor in Bild 1 vertauscht ist: Anschluss 4 ist positiv, Anschluss 5 ist negativ. Bei beiden Konfigurationen (2a und 2b) entfällt Widerstand R2.

Für die drei Varianten hat der Autor eine kleine Platine mit den Abmessungen 42,5 x 11 mm entworfen (**Bild 3**). Darauf sind die Eingangs- und die Ausgangsseite auch räumlich strikt voneinander getrennt. Für den Optokoppler mit Foto-

Bild 2. Koppler des Typs optoMOS können Gleich- und Wechselströme schalten.



Stückliste

Widerstände:

R1 = 1 k
R2 = 1 M (optional)

Kondensatoren:

C1 = 10 n

Halbleiter:

LED = LED rot
Opto (siehe Text) =
Version mit Fototransistor: 4N25, 4N26, 4N27,
4N32, 4N33, 4N37, CNY17-3, SL5500, TIL111
Version mit optoMOS: LCA110, OMA160 (CLARE)

Außerdem:

S1 = Umschalter, drei Stellungen
S2 = Drucktaster (optional)
K1 = Klinkenbuchse 3,5 mm, möglichst isoliert
K2 = Sub-D-Stecker 9-polig, weiblich
DIP-Fassung 6-polig

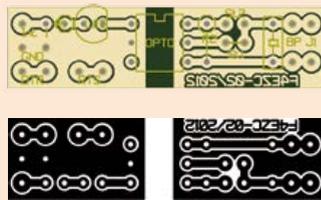


Bild 3.
Platine und Bestückung des
Schnittstellen-Isolators.

Bild 4.
Musteraufbau des Autors,
hier mit einem optoMOS-
Koppler.

transistor und optionalem Widerstand R2 muss Jumper JP1 gesteckt sein, JP2 bleibt offen. Der optoMOS in Wechselstrom-Konfiguration (2a) erfordert, dass JP1 offen bleibt und JP2 gesteckt ist, R2 entfällt. Bei der Variante für Gleichstrom

(2b) wird JP1 gesteckt, JP2 bleibt offen und R2 wird durch eine Drahtbrücke ersetzt.

Die Polarität der Klinkenbuchse K1 muss unbedingt beachtet werden. Wird K1 in ein Metallgehäuse eingebaut, ist diese Buchse zu isolieren. Seinem Namen wird der Schnittstellen-Isolator allerdings erst voll gerecht, wenn das Gehäuse aus nicht leitendem Kunststoff besteht. Der Optokoppler soll in einer Fassung montiert werden, so dass er leicht auswechselbar ist. Die in der Stückliste aufgeführten Typen wurden getestet und für gut befunden.

Anwendung

Schließen Sie den Sender bei der ersten Inbetriebnahme noch nicht an. Verbinden Sie den RS232-Eingang des Schnittstellen-Isolators mit dem PC, wenn nötig über einen USB/RS232-Wandler. Dann starten Sie Ihre *Radio-Application* für PTT oder CW auf dem PC. Wenn Ihr Programm die Wahl zwischen RTS und DTR zulässt, wählen Sie ein Signal von beiden aus. Der Wahlschalter des Schnittstellen-Isolators muss auf das gleiche Signal eingestellt sein. Sobald das Programm Kommandos sendet, muss die LED des Schnittstellen-Isolators aufleuchten. Sendet Ihr Programm CW, blinkt die LED im Rhythmus der Morsezeichen. In diesem Fall verbinden Sie K1 mit dem CW-Eingang des Senders.

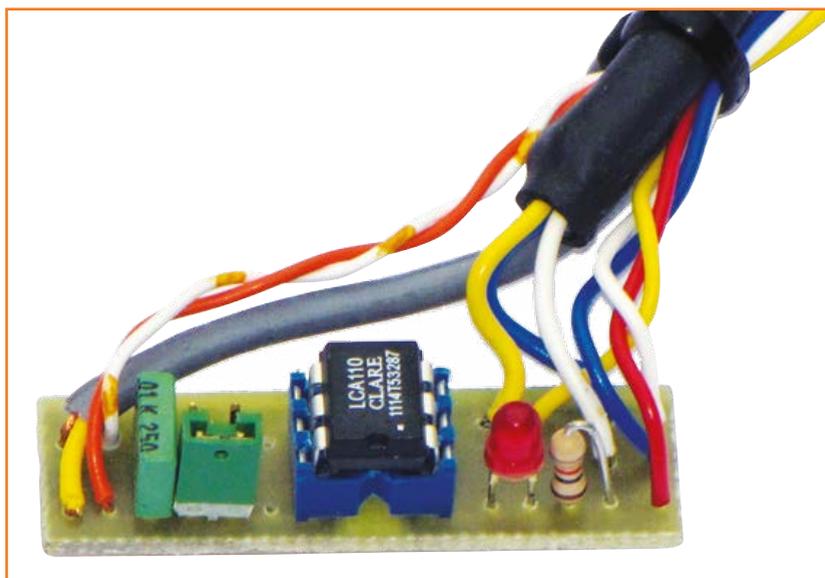
Steuert Ihre Software die Push-to-Talk-Funktion des Senders und schaltet das PTT-Signal den Sender ein, verbinden Sie K1 mit dem PTT-Eingang des Senders. Mit einem Druck auf Taster S2 können Sie prüfen, ob der Sender ordnungsgemäß auf das Steuersignal reagiert. Denken Sie daran, den Wahlschalter des Schnittstellen-Isolators auf OFF zu stellen, solange Sie mit dem PC den Sender nicht steuern wollen.

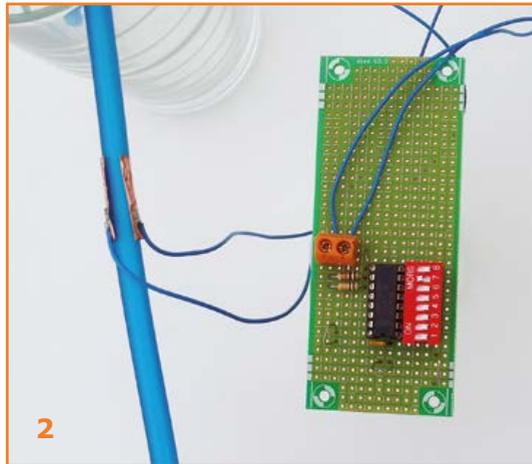
Zum Schluss noch ein letzter, aber wichtiger Hinweis: Wenn der Schnittstellen-Isolator an einer *normgemäßen* RS232-Schnittstelle betrieben wird, die mit Spannungen von ± 12 V arbeitet, muss die Diode 1N4148 montiert werden, die im Schaltbild durchbrochen gezeichnet ist. Von der Diode werden die LED und der Optokoppler vor Überspannung geschützt. Die Sperrspannung der meisten LEDs und der LEDs in Optokopplern beträgt nicht mehr als 5 V.

(120328)gd

Weblink

[1] Datenblatt LCA110: <http://goo.gl/DIgbqo>





Wasser im Kondensator

Beim Oszillator des Zähler/Teiler-ICs 4060 (**Bild 1**) sind die drei Pins RCX, RX und CX von außen zugänglich. Eben dort wird der Kondensator angeschlossen, den die beiden aufgeklebten Kupferstreifen bilden. Die Kapazität dieser Anordnung ist jeweils anders, wenn Wasser in der Leitung ist bzw. wenn nicht. Und mit dieser Kapazität ändert sich auch die Frequenz des Oszillators. Man braucht also nur die Impulsdauer

am Ausgang des Zählers messen, um zu wissen, was im Schlauch vor sich geht. Dies funktioniert allerdings nicht mit Leitungen aus Metall.

Die Frequenz des zur Auswertung genutzten Oszillator-Signals wird vorab noch durch den Teiler des 4060 gesenkt. Der 8-fach-DIP-Schalter dient zur Auswahl einer der verfügbaren Frequenzen. Aber Achtung: man darf immer nur einen Schalter gleichzeitig aktivieren. Das resultierende Rechtecksignal wird über K1 zu einem Arduino geführt und kann dort durch ein kleines Programm „analysiert“ werden (**Listing 1**).

Der Detektor liefert nur Werte, die zwischen zwei Extremen pendeln (An- oder Abwesenheit von Flüssigkeit). Die absoluten Werte haben keine Bedeutung, es zählt nur die gewählte Schaltschwelle auf halbem Wege.

Mit dem abgebildeten Sensor (**Bild 2**, der Schlauch hat 6 mm Außen- und 4 mm Innendurchmesser, die Kupferstreifen messen ca. 20 x 7 mm) erhält man einen Kapazitäts-Unterschied von etwa 1,5 %. Das ist nicht viel, aber ein Mikrocontroller kann damit umgehen.

Stückliste

Widerstände (1/4 W):

R1 = 100 k
R2 = 10 k

Kondensatoren:

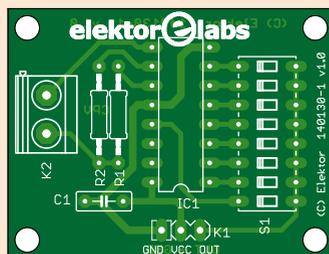
C1 = 100 n

Halbleiter:

IC1 = 74HC4060N

Außerdem:

K1 = Stiftleiste 3-polig
K2 = Schraubklemme, RM 5,08 mm
S1 = DIP-Schalter, 8-polig
Platine 140130-1-v1.0 [6]



Listing 1

```
int freqPin = 7; // probe input
int beepPin = 6; // buzzer
unsigned long duration;

void setup()
{
  Serial.begin(9600);
  pinMode(freqPin, INPUT);
  pinMode(beepPin, OUTPUT);
}

void loop()
{
  duration = pulseIn(freqPin, HIGH);
  Serial.println(duration);

  // delay(1000);
  if (duration > 3200)
  {
    tone(beepPin, 600, 100);
  }
}
```

Längere Streifen sollten die Empfindlichkeit erhöhen. Beim Prototypen des Autors, der mit Wasser und Diesel getestet wurde, treten Unterschiede von bis zu 10 % auf. Die Frequenz *ohne Wasser* von 166 Hz ergibt einen mit *pulseIn* ermittelten Wert von 3128, während die 151 Hz *mit Wasser* *pulseIn* = 3229 liefern. Als Schaltschwelle wurde 3178 gewählt.

Kurz und gut, wir sind noch im Experimentierstadium. Es gilt nicht nur, die Messung zu verbessern, es soll auch ein Verfahren gefunden wer-

den, das System periodisch zu kalibrieren, um ein mögliches Driften zu kompensieren.

Für dieses Projekt wurde eine Platine entworfen, diese steht zum Download bereit [6].

Nun sind Sie als Leser aufgerufen, uns allen Ihre Erfahrungen mitzuteilen [1][2].

(140130)

Weblinks

- [1] www.elektor-labs.com
- [2] Ein etwas anderer Regenschwimmer: www.elektor-magazine.de/120554
- [3] Multifunktionaler Touch-Wandschalter: www.elektor-magazine.de/130272
- [4] <http://goo.gl/TcCE94>
- [5] Pattern Lock: www.elektor-magazine.de/120579
- [6] www.elektor-magazine.de/140130

Anzeige

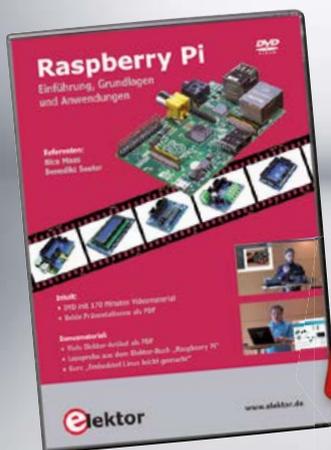
Workshop-DVD Raspberry Pi

Einführung, Grundlagen und Anwendungen

Konnten Sie aus zeitlichen Gründen nicht an unserem ElektorLive-Seminartag am 12.10.2013 in Hanau teilnehmen und haben somit auch die beiden populären Raspberry-Pi-Seminare verpasst!? Kein Problem! Mit dieser neuen DVD-Video sind Sie jetzt trotzdem live dabei!

IT-Systemelektroniker Nico Maas erklärt in diesem Einführungskurs, wo die Möglichkeiten, Stärken und Grenzen des 40-Euro-Rechners und die Vorteile gegenüber traditionellen Mikrocontroller-Anwendungen liegen. Entwickler Benedikt Sauter stellt Ihnen in seinem Seminar praktische Anwendungen vor, um den Raspberry Pi, das BeagleBone Black und das GNUBLIN/Elektor-Linux-Board in eigenen Projekten einsetzen zu können.

NEU!



Inhalt:

- 170 Minuten Videomaterial
- Beide Präsentationen als PDF
- Viele Elektor-Artikel zu RPi als PDF

ISBN 978-3-89576-295-6

€ 24,90
CHF 30,90

Weitere Infos & Bestellung unter www.elektor.de

Schaeffer AG



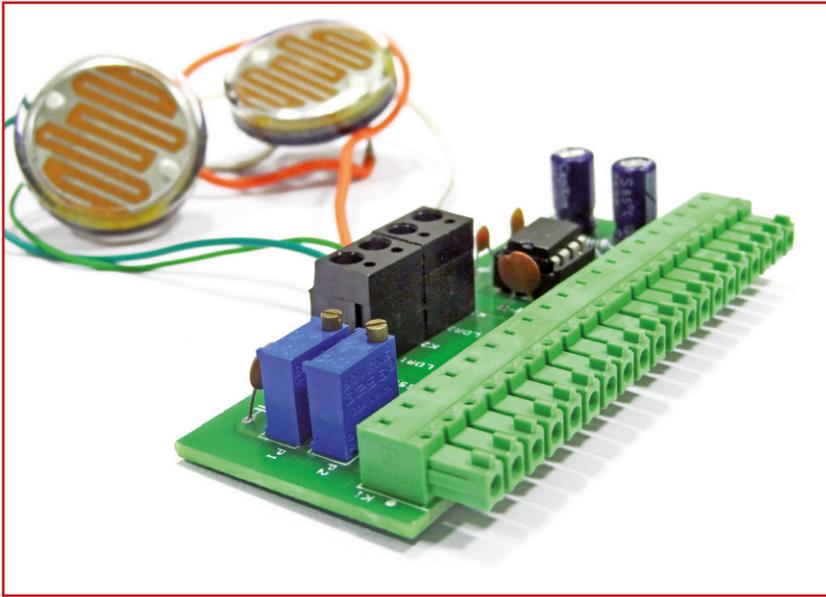
SIE DESIGNEN – WIR FERTIGEN

Frontplatten in Profiqualität

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

www.schaeffer-ag.de

Optisches Theremin mit myDAQ und LabVIEW



Von **Sunil Malekar** (Elektor Labor Indien)

MyDAQ und LabVIEW von National Instruments sind normalerweise eng mit edukativen und professionellen Anwendungen der Mess- und Regeltechnik verbunden. Hier aber kommt ein Elektor-Projekt mit einer spielerischen Herangehensweise an diese Entwicklungssoftware.

Das Theremin wurde in den später 1920er Jahren von dem russischen Musiker Lew Termen erfunden [1] und gilt als eines der ersten elektronischen Musikinstrumente. Es wird „dreidimensional“ durch Verändern der Position und der Geschwindigkeit der Hand des Spielers zwischen oder über zwei Antennen gespielt. Die Antennen, die leicht interagieren, bestimmen die Tonhöhe und Lautstärke eines Audioszillators. So ein Theremin ist erstaunlich aktuell: Matthew Bellamy von der Band *Muse* imitiert den sphärischen Klang auf dem XY-Pad seiner Manson M1D1 E-Gitarre mit Hilfe von Effektgeräten [2], Brian Jones von den Rolling Stones, Jean-Michel Jarre und Led Zeppelins Jimmy Page haben echte Instrumente verwendet, nicht zu vergessen Léon Theremin (wie sich Termen in den USA nannte) höchst daselbst mit Klavierbegleitung [3]. Und das Instrument, das Sie in „Good Vibrations“ der Beach Boys hören, ist ebenfalls ein Theremin-Derivat.

Seit nahezu 100 Jahren hat der Oszillator im Herzen des Theremins eine technologische Weiterentwicklung erfahren, von der Röhre zum DSP. In diesem Projekt verbinden wir einen solchen Oszil-

lator mit zwei Produkten von National Instruments (NI): einer myDAQ-Datenerfassungs-Hardware [4] und LabVIEW für die Steuerung.

Einkaufszettel

Bevor es losgehen kann mit dem Theremin, muss folgende Hardware besorgt werden:

- MyDAQ mit USB-Datenkabel,
- PC mit LabVIEW-Software,
- Platine Optical Theremin von Elektor,
- aktive (PC-)Lautsprecher.

Und auch ein Treiber namens ELVISmx ist noch erforderlich, der es möglich macht, dass LabVIEW-basierte Softwareinstrumente myDAQ steuern können.

Mein myDAQ

MyDAQ von National Instruments ist eine portable Datenerfassungseinheit (DAQ), die gut mit den virtuellen Geräten der berühmten LabVIEW-Software zur Messung und Analyse von realen Signalen zusammenarbeitet. MyDAQ eignet sich gut für die Erkundung der Elektronik im Allgemeinen und Sensorik im Speziellen. In Verbindung mit

LabVIEW kann man auf dem PC erfasste Signale analysieren und verarbeiten sowie einfache Prozesssteuerungen entwickeln. Das myDAQ-Gerät wird mit einem USB-Kabel für Datenverkehr und Stromversorgung am PC angeschlossen.

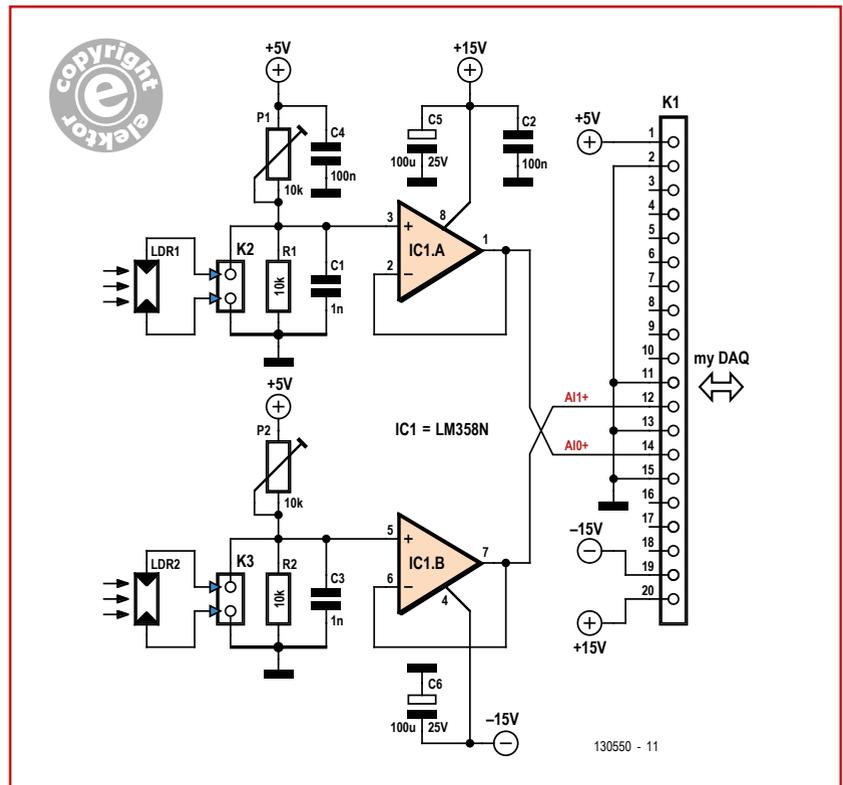
MyDAQ bietet eine Reihe von Ein- und Ausgängen sowie Audiofunktionen, eine Stromversorgung und die Funktionalität eines Digitalmultimeters in einem kompakten USB-Gerät. Die verfügbaren I/Os und die Stromversorgung sind hier unser wichtigstes Anliegen.

- **AI:** zwei analoge Eingangskanäle, die man entweder als ein Allzweck-Hochimpedanz-Differenzspannungseingang oder als Audioeingang konfigurieren kann;
- **AO:** wie AI, aber 2-mA-max-Differenzspannungs- oder Audio-Ausgang;
- **DIO:** 8 digitale Ein- und Ausgänge, jeweils als digitale, von der Software getaktete Allzweck-Ein- oder Ausgänge oder als Ausgang eines digitalen Zählers konfigurierbar.
- **Versorgungsspannungen:** -15 V, +15 V und +5 V.

Das Add-on-Board „Optical Theremin“

Das Add-on-Board von Elektor verbindet LDRs (lichtabhängige Widerstände) mit dem myDAQ-Gerät. Ein LDR auf der Platine bestimmt die Frequenz (Tonhöhe) und der andere die Lautstärke (Amplitude) des Ausgangssignals. Dazu benötigt das Board die Hilfe von LabVIEW und der „Elektor Optical Theremin VI“-Software. VI steht für Virtual Instruments.

Der Schaltplan zu dieser Platine ist in **Bild 1** zu sehen. Die Klemmenreihe K1 wird verwendet, um das Board mit dem myDAQ zu verbinden. Die Theremin-Platine liefert die Signale für myDAQ, das Datenerfassungsgerät die Versorgungsspannung für die Theremin-Platine. Der Dual-Operationsverstärker LM358 „puffert“ die durch das Licht verursachten Widerstandsänderungen der an K2 und K3 angeschlossenen LDRs. LDR1 an K2 steuert die Frequenz; mit P1 lässt sich ein Basis-Wert einstellen. LDR2 an K3 bestimmt die Lautstärke; auch hier gibt es einen (mit P2 einstellbaren) Basis-Wert. Die Operationsverstärker-Ausgänge sind mit den Analogeingängen des myDAQ verbunden. Da vom myDAQ ein Line-Audiosignal ausgegeben wird, müssen wir einen Verstärker oder aktive Lautsprecher an der Audio-Out-Buchse anschließen.



Ein Program für das Optische Theremin

Die Anwendungssoftware *VI Front Panel* bietet zwei grafische Darstellungen, um die Schwankungen von Lautstärke und Frequenz zu beobachten (**Bild 2**). Ein Zeigerinstrument wird für die Frequenzschwankungen benutzt, ein Schieberegler, um die Lautstärke anzuzeigen.

Die Analogeingänge A1 und A2 des myDAQ erfassen die Spannungssignale, die von LDR1 und

Bild 1. Schaltung des HID (human interface device), das Gesten in zwei analoge Signale übersetzt, die Tonhöhe und Amplitude eines Audio-Oszillators bestimmen.

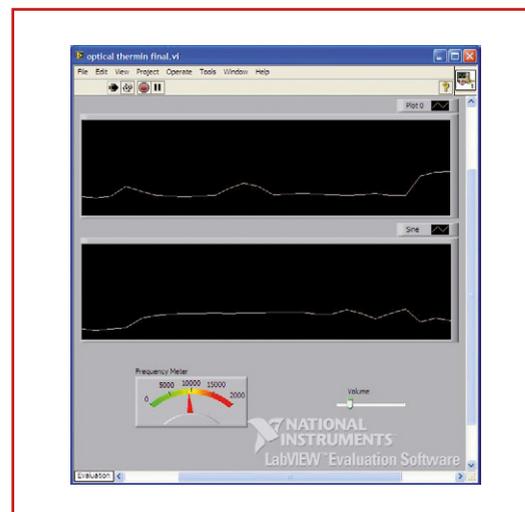


Bild 2. VI Front Panel mit Anzeige der Frequenz und der Amplitude.

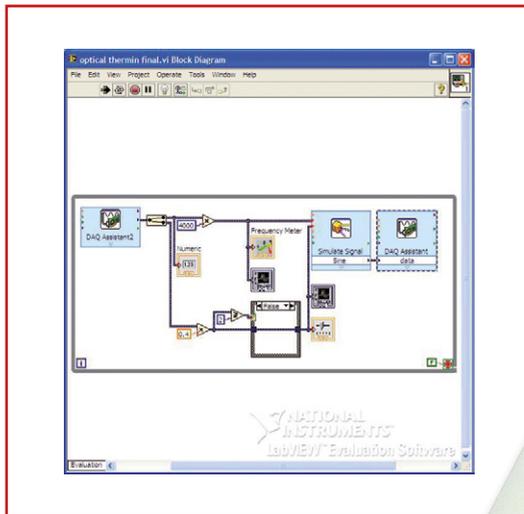
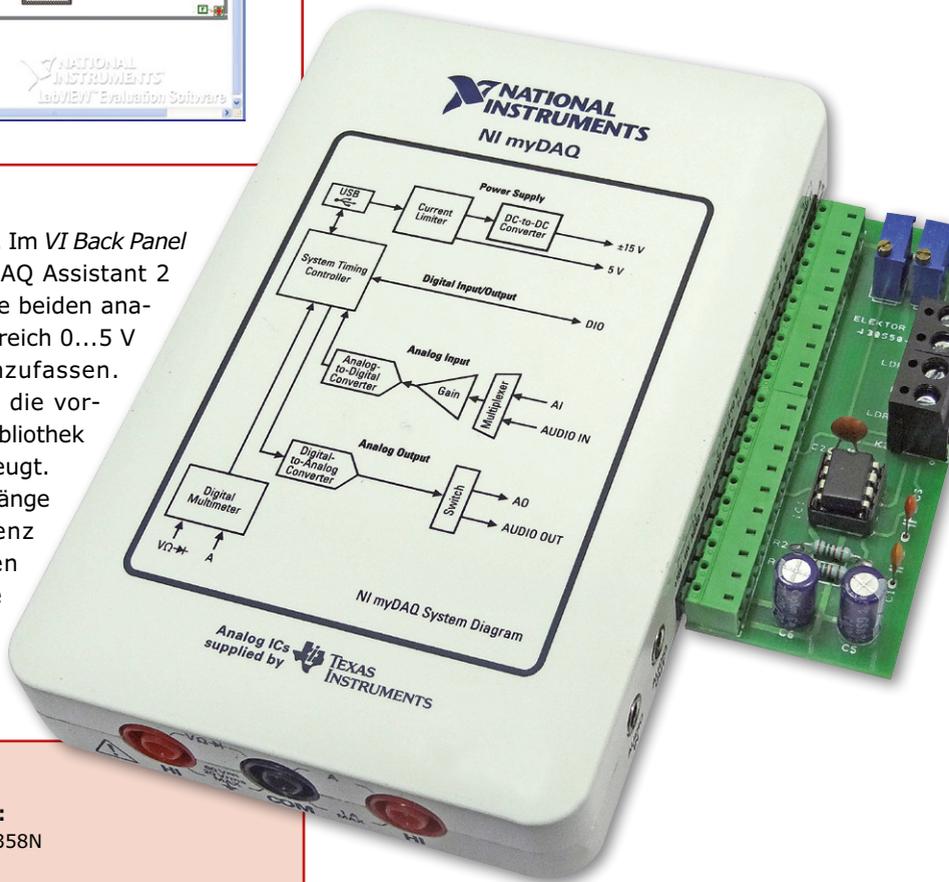


Bild 3.
VI Back Panel: das
Blockschaltbild des
Theremin in LabVIEW.

LDR2 variiert werden. Im VI Back Panel (Bild 3) wird der DAQ Assistant 2 verwendet, um diese beiden analogen Signale im Bereich 0...5 V zum VI zusammenzufassen. Der Ton wird durch die vordefinierte myDAQ-Bibliothek Simulate Signal erzeugt. Einer der Analogeingänge steuert die Frequenz des zu erzeugenden Tons. Die maximale Frequenz, die von myDAQ erzeugt

Bild 4.
Bestückungsaufdruck der
einfachen Platine, die an den
myDAQ gesteckt wird.



werden kann, ist 20 kHz, so dass der analoge Eingangswert mit dem Faktor 4000 multipliziert werden muss. Dieses Signal wird dann mit dem Grafikmodul sowie dem Frequenzzähler auf dem Front Panel verbunden.

Da die maximale Signalstärke des Generators $2 V_{pp}$ beträgt, muss das analoge Eingangssignal für die Lautstärke mit dem Faktor 0,4 mul-

tipliziert werden. Dieses Signal wird dann dem Grafikmodul und der Balkenanzeige auf dem VI Front Panel zugeführt.

Let's Theremin!

Haben Sie das Add-on-Board (welch eine Bezeichnung für eine solch einfache Platine!) in (Bild 4) erfolgreich aufgebaut und die Software für das Projekt von der Elektor-Webseite [5] heruntergeladen, dann gehen Sie wie folgt vor:

- Das Elektor Optical Theremin Board am myDAQ einstecken;

Stückliste

Widerstände:

R1,R2 = 10 k
P1,P2 = 10 k Mehrgang-Trimpoti

Kondensatoren:

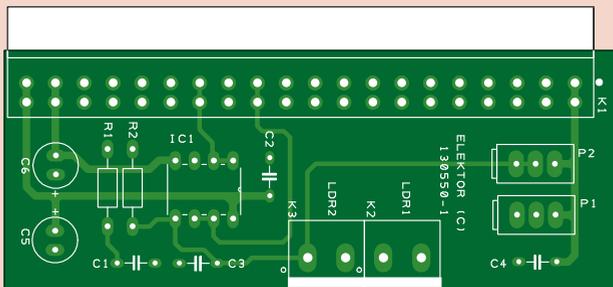
C1,C3 = 1 n
C2,C4 = 100 n
C5,C6 = 100 µ, 25 V radial

Halbleiter:

IC1 = LM358N

Außerdem:

K1 = 20-polige Platinenanschlussklemme, gewinkelt, RM 3,81 mm
K2,K3 = 2-polige Schraubklemme Platine 130550-1 [5]



PicoScope[®]

PC-OSZILLOSKOPE

Für jede Anwendung gibt es ein PicoScope

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

PICOSCOPE 6000-SERIE

Ultimative Leistung
inkl. USB 3.0



PICOSCOPE 5000-SERIE

Oszilloskope mit
flexibler Auflösung



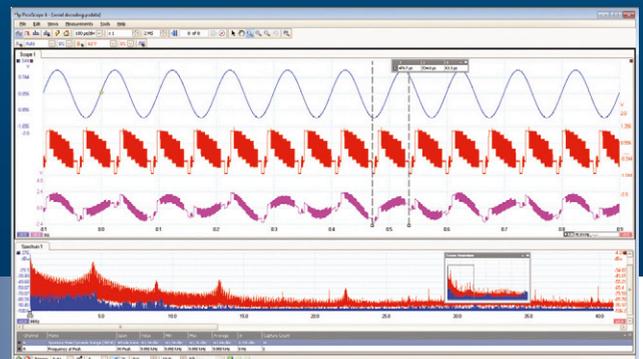
PICOSCOPE 4824

8 Kanäle,
hohe Auflösung



PICOSCOPE 3400-SERIE

Oszilloskope mit
hoher Leistung



PICOSCOPE SOFTWARE ENTHÄLT AUTOMATISCHE MESSUNGEN, SPEKTRUM-ANALYSATOR, SDK, ERWEITERTE TRIGGER, FARBIGES NACHLEUCHTEN, SERIELLES DECODING (CAN, LIN, RS232, I²C, PS, FLEXRAY, SPI), MASKENTEST, MATH-KANÄLE, ALLE ALS STANDARD-FUNKTIONEN, KOSTENFREIE UPDATES

www.picotech.com/PS254

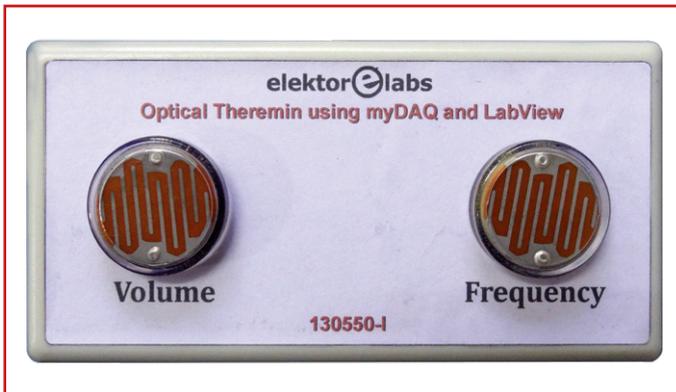


Bild 5. Frontansicht des Optical Theremin.

- den aktiven Lautsprecher oder einen Verstärker am myDAQ anschließen;
- den myDAQ über ein USB-Datenkabel an den PC anschließen;
- das VI-Programm aus dem Software-Ordner ausführen
- und mit einem Klick auf die PLAY-Taste starten;
- Ihre Mitbewohner darauf hinweisen, dass es im *Omikron Kappa Hostel* in der benachbarten Straße kostenloses WLAN und leckere Pizza gibt.
- Beeinflussen Sie das Licht, das auf die LDRs fällt, mit beiden Händen und machen Sie so Musik (oder das, was Sie dafür halten).

Anmerkung: Manchmal hilft es, bei Theremin-Etuden (Kopfhörer auf, danke!) eine Software einzusetzen, die in Echtzeit ihre Handbewegungen aufzeichnet.

(130550)

Weblinks

- [1] Geschichte des Theremin:
<http://de.wikipedia.org/wiki/Theremin>
- [2] Muse: „Invincible“ Live @ Wembley:
<http://youtu.be/t2KX0xBIEZI>
- [3] Léon spielt Theremin: <http://youtu.be/w5qf9O6c20o>
- [4] myDAQ und LabView:
www.elektor.de/mydaq-labview-multisim
- [5] Projektsoftware: www.elektor-magazine.de/130550

Cube Clock

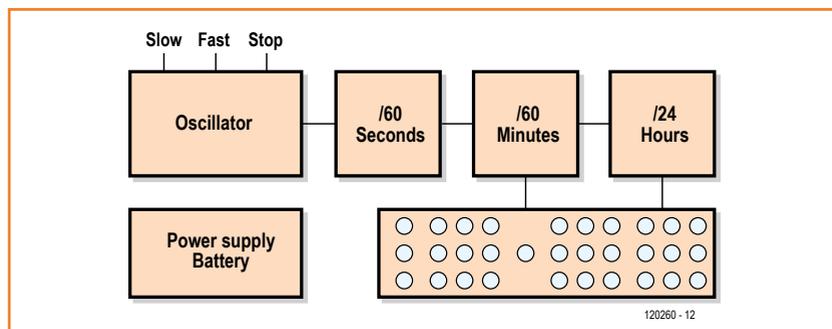
Digitaluhr im Würfel-Design

Von
Joost Waegebaert (B)



Wie muss ein Mikrocontroller programmiert werden, damit er über lange Zeit ein hochpräzises Zeitsignal liefert? Um diese Frage kreisten die Diskussionen auf mehreren Internet-Foren. Methoden und Verfahren wurden erwogen, für gut befunden oder wieder verworfen. Möglichst alles durch Software zu lösen und die Hardware zurückzudrängen, liegt im Trend der Zeit. Trotzdem gibt es noch Nischen, in denen die Hardware die Oberhand behalten hat. Zu dieser Kategorie gehört die Digitaluhr, die wir hier vorstellen. Aufgebaut mit Frequenzteilern klassischer Art, ist sie „Eyecatcher“ und Uhr zugleich. Außerdem ist sie extrem genügsam, Stromausfälle überbrückt sie mit einer unscheinbaren Batterie für erstaunlich lange Zeit.

Bild 1.
Funktionsschema der Uhr mit der Zeitanzeige in Würfel-Mustern.



Digitaluhren sind dankbare Objekte, wenn es darum geht, brachliegende Zeit- und Material-Ressourcen durch nützliche und lehrreiche Projekte zu nutzen. Den Anstoß zu dieser Uhr gaben Diskussionen auf Internet-Foren, die das Generieren eines hochpräzisen 1-Hz-Signals zum Thema hatten. Das Herz dieser Uhr besteht aus CMOS-Logik-ICs, ein Mikrocontroller ist nicht beteiligt.

Schema

Im Prinzip ist eine Digitaluhr eine Zählerkette, wobei der Zählerstand im Rhythmus der laufenden Sekunden um eins erhöht wird. Die Kette ist so gegliedert, dass Sekunden, Minuten und Stunden von separaten Zählern gezählt werden. Jeder Zähler steuert ein Display, das den Zählerstand anzeigt. In **Bild 1** sind die Funktionen blockschematisch dargestellt.

Oszillator

Das Kernstück jeder elektronischen Uhr ist ein stabiler Oszillator, das Element, das die Frequenz bestimmt, ist ein Quarz. Grundsätzlich kann die Frequenz frei gewählt werden, denn ein nachgeschalteter Frequenzteiler lässt sich immer so konfigurieren, dass am Ausgang ein Signal mit der Frequenz 1 Hz erscheint. Die Oszillatorfrequenz hat jedoch Einfluss auf die Energiebilanz. Den Quarzoszillatoren ist eigen, dass der Energiebedarf mit niedrigen Frequenzen sinkt. Die zweite Überlegung geht dahin, dass die Oszillatorfrequenz möglichst unkompliziert auf 1 Hz teilbar sein soll. Wenn die Oszillatorfrequenz eine Potenz der Zahl 2 ist, kann der Frequenzteiler aus einfachen, in Reihe geschalteten Flipflops aufgebaut werden. Vor allem für diesen Zweck werden Quarze mit der Frequenz 32,768 kHz

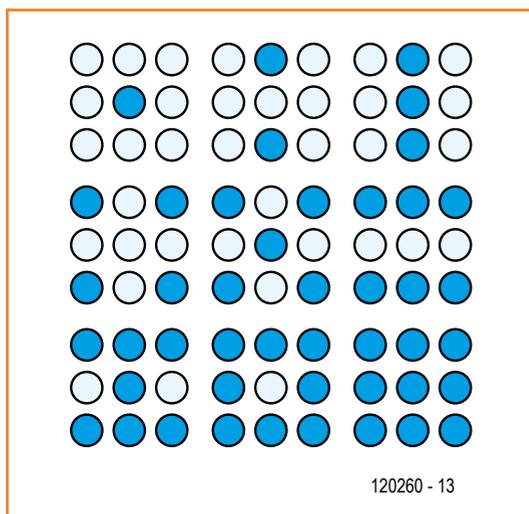


Bild 3.
Die Muster für die Ziffern 1...6 wurden dem mechanischen Spiele-Würfel entlehnt.

fen, denn auch die Temperatur hat Einfluss auf die Genauigkeit. Mit einem Frequenzmesser wird das Signal an Pin 5 von IC1 auf genau 32768,00 Hz eingestellt. Wenn kein Frequenzmesser verfügbar ist, genügt hilfsweise ein täglicher Vergleich mit einem Zeitnormal zu einer festen Tageszeit. Die Einstellung von C3 wird täglich entsprechend der Abweichung korrigiert. Nach etwa einer Woche dürfte die angestrebte Genauigkeit erreicht sein.

Zähler

Ein fast maßgeschneiderter Baustein für eine Digitaluhr ist der CMOS-Typ 4566. Darin sind zwei BCD-Zähler untergebracht, der erste zählt von 0 bis 9, der zweite wahlweise bis 5 oder 6. Wenn der zweite Zähler bis 6 zählt, sind der Sekunden- und Minuten-Zähler (IC2 und IC3) bereits realisiert.

Der Zähler, der 24 Stunden zählt, besteht aus den beiden BCD-Zählern, die in IC4 enthalten sind. Diese Zähler werden beim Zählerstand 24 über die Dioden D5 und D6 rückgesetzt. Im 4566 ist ein monostabiler Multivibrator integriert, er erzeugt hier aus dem Impuls am Verbindungspunkt mit R1 das 24-Stunden-Reset-Signal. Soll der Zähler nur 12 Stunden zählen, müssen D5 mit Ausgang Q1 von IC4A und D6 mit Ausgang Q2 von IC4B verbunden werden.

Display

Die Anzeige der Würfel-Uhr besteht aus LEDs, die wie die Punkte eines Spiele-Würfels angeordnet sind. Damit ist sie verwandt mit den Binäruhren, die die Zeit im Binär-Code visualisieren. Anders als der Binär-Code sind Würfel-Muster auch von solchen Menschen leicht zu deuten, die nur ent-

fernte Berührung mit der Digitaltechnik haben. Auf dem Würfel werden die Ziffern 0...9 dargestellt, indem die dazugehörige Anzahl LEDs aufleuchten. Die Muster, die **Bild 3** zeigt, stimmen für die Ziffern 1...6 mit den bekannten Würfel-Mustern überein. Die Muster für die Ziffern 7...9 kommen ebenso wie die 0 auf mechanischen Würfeln nicht vor. Bei der 0 bleiben alle LEDs dunkel.

Die LEDs werden von Stromquellen gesteuert, die diskret aufgebaut sind, beispielsweise mit Transistor T3 und Emitterwiderstand R11. Bezogen auf die Betriebsspannung +18 V ist die Basisspannung von T3 konstant. Wenn auch die Basisspannung bezogen auf den Emitter konstant ist, liegt an R11 eine konstante Spannung. Über den Emitter fließt ein konstanter Strom. Der durch die LED fließende Kollektorstrom hat den gleichen Wert wie der Emitterstrom abzüglich des Basisstroms. Da T3 ein Darlington-Transistor mit sehr niedrigem Basisstrom ist, kann der Basisstrom vernachlässigt werden.

Mit Trimpoti R18 lässt sich die Basisspannung und folglich die Helligkeit der LEDs einstellen. Der Basis-Emitter-Übergang von T2 in Reihe mit R18 bewirkt eine Temperaturkompensation der Stromquellen. Ändert sich die Basis-Emitter-Spannung beispielsweise von T3, wird diese Drift durch eine proportionale Änderung der Referenzspannung an der Basis kompensiert.

VR1, eine integrierte Spannungsquelle des Typs TL431, ist mit R16 und R17 auf die Spannung 3 V eingestellt. Die Basis-Emitter-Spannung der Darlingtonströme beträgt 1,4 V. Mit Trimpoti R18 lässt sich die Spannung an den Stromquellen-Widerständen, zum Beispiel an R11, im Bereich 0...1,6 V einstellen, was LED-Strömen bis 16 mA entspricht. Durch die Stromsteuerung leuchten alle LEDs der Display-Segmente mit gleicher Helligkeit, unabhängig von der Anzahl der in Reihe liegenden LEDs.

Die Segmente werden mit MOSFETs geschaltet, beispielsweise mit T8. Die MOSFETs übernehmen auch das Trennen der LEDs von der Uhr, falls die Netzspannung ausfällt. Für diese Aufgabe sind MOSFETs ideal, da bei statischem Betrieb nur Gate-Leckströme in der Größenordnung 10 nA fließen. Bipolare Transistoren können zwar auch perfekt schalten, sie brauchen jedoch stets Basisstrom, so dass sie für den Notbetrieb bei Netzausfall nicht geeignet sind. Die Wahl des MOSFET-Typs ist unkritisch, geeignet sind zum Beispiel der 2N7000, BSS100 oder BSS123.

Das Decodieren der binären Werte in die zugehörigen Würfel-Muster gestaltet sich einfach, wenn die LEDs der Reihenschaltungen geschickt kombiniert werden. Nur für Bit 3 der Minuten (Leitung M08) sind die zusätzlichen Dioden D8 und D9 nötig.

Wegen der Stromsteuerung kann beispielsweise jedes Digit in einer eigenen Farbe aufleuchten. Der Strom durch die LEDs bleibt konstant, unabhängig vom Spannungsabfall an verschiedenfarbigen LEDs. Unterschiedliche Wirkungsgrade lassen sich kompensieren, indem die Ströme über die Werte der Emitterwiderstände angepasst werden. Die Schaltung des Blocks mit M01...M08 wird für die übrigen Digits dreimal wiederholt. Für H11...H12 sind nur zwei Stromquellen nötig, da die höchste darzustellende Ziffer die 2 ist. Zwischen den Minuten- und Stunden-Digits zeigt eine blinkende Sekunden-LED den Betrieb der Uhr an. Diese LED wird unmittelbar vom 1-Hz-Signal gesteuert.

Stromversorgung

Das Netzteil liefert eine stabilisierte Spannung für den Oszillator und die Teilerkette sowie eine unstabilisierte Spannung für das Würfel-Display. Die Trafospaltung hängt von den Spannungsabfällen der LEDs ab, sie reichen von 1,7 V für rote LEDs bis 4 V für einige weiße LED-Typen. Die Display-Segmente mit vier LEDs in Reihe bestimmen die Spannung, die mindestens verfügbar sein muss. Die Spannungsabfälle an den Stromquellen (Spannung am Emitterwiderstand + Durchlassspannung der Darlingtons) gehen mit ungefähr 3 V in die Rechnung ein. Für vier weiße LEDs muss die Spannung $4 \cdot 4 \text{ V} + 3 \text{ V} = 19 \text{ V}$ betragen. Hinzu kommen noch 1,4 V, die an den Dioden der Brücke B1 abfallen. Das Ergebnis ist eine Trafospaltung von mindestens $(19 \text{ V}_- + 1,4 \text{ V}_-) / \sqrt{2} \approx 14,5 \text{ V}_\sim$.

Die Spannung für die Stromquellen muss nicht stabilisiert sein, denn der LED-Strom wird bereits von den Stromquellen konstant gehalten.

Der Spannungsregler LM2936 (IC5) unterscheidet sich von anderen Typen durch seinen ungewöhnlich niedrigen Ruhestrom. An der Spannung 9 V lag die gemessene Stromaufnahme von Oszillator und Zählern bei insgesamt 28 μA . Diesen Strom liefert der LM2936, dabei erhöht er die Bilanz um weitere 10 μA . Folglich beträgt der Strombedarf bei Ausfall der Netzspannung nur 38 μA , die Displays sind dann allerdings abgeschaltet. Die Dropout-Spannung des LM2936 liegt bei Strö-

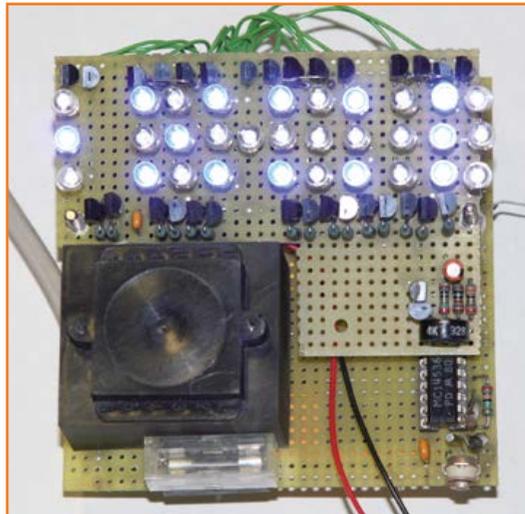


Bild 4.
Der Autor hat den Prototyp auf zwei Lötpunktraster-Platinen aufgebaut.

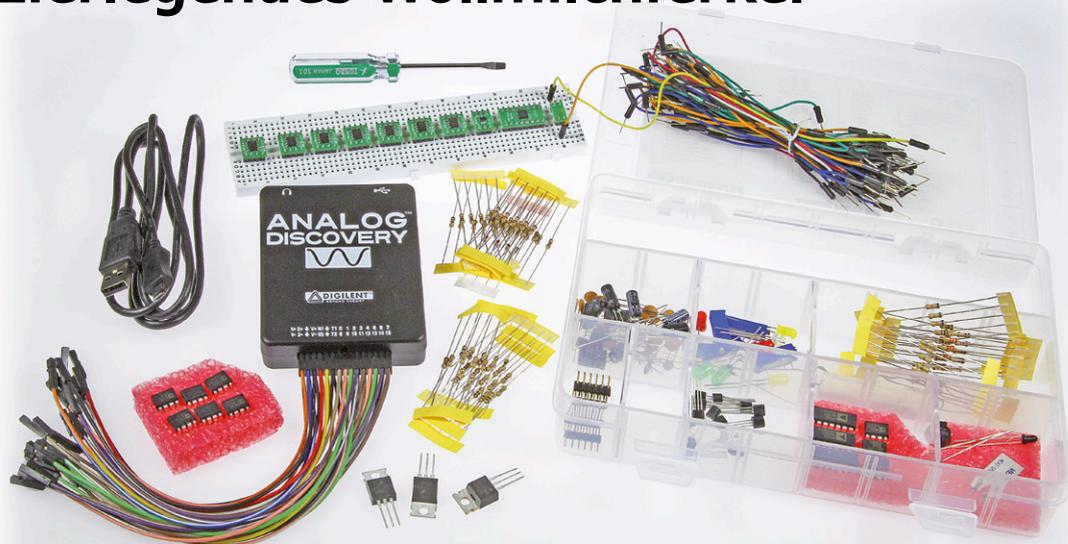
men dieser Größenordnung unter 0,1 V, so dass die Energie der Batterie optimal genutzt wird. Wenn durchschnittliche 9-V-Alkaline-Batterien bis 5,4 V entladen werden, haben sie eine Nennkapazität von 550 mAh. Für die Uhr bedeutet dies, dass die Batterie $0,55 \text{ Ah} / 0,000038 \text{ A} \approx 14470$ Stunden im Standby-Betrieb durchhält, was mehr als 1,5 Jahren entspricht. Auch ein Akku kann als Standby-Energiequelle dienen, wenn Diode D11 mit einem Widerstand für die Ladungserhaltung überbrückt wird. Ein Akku bringt jedoch keine wesentlichen Vorteile.

Zum Schluss

Die Cube Clock ist ein Beweis dafür, dass mit klassischen Logik-Bausteinen auf unkomplizierte Weise attraktive Projekte realisierbar sind. Insbesondere der extrem niedrige Energiebedarf dürfte einige Mikrocontroller-Experten überraschen. Als Erweiterung kann eine Dimmschaltung nützlich sein, die die Referenzspannung der Stromquellen an das Umgebungslicht anpasst. Auch ein netzunabhängiger Betrieb ist möglich, wenn der Trafo durch einen Aufwärtsregler ersetzt wird. Er versorgt die Stromquellen auf Knopfdruck aus der Batterie. Wenn zwei weitere Würfel-Displays hinzugefügt werden, können auch die an IC2 liegenden Sekunden-Informationen visualisiert werden. Die CMOS-Logik-Familie 4000 wird auch heute noch von mehreren namhaften Herstellern produziert, darunter sind TI, NXP und OnSemi. Herstellerabhängig können die Typenbezeichnungen unterschiedliche Präfixe tragen, zum Beispiel sind die Typen CD4566, HEF4566 und MC14566 miteinander kompatibel.

(120260)gd

Digilent „Analog Discovery“ Eierlegendes Wollmilchferkel



Von
Dr. Thomas Scherer
(D)

Ein USB-Oszilloskop mit ein paar Zusatzfunktionen und einer analogen Bandbreite von 5 MHz – fängt ein Elektroniker da nicht an zu gähnen? Doch dieses Vielzweck-Gerät hat es in sich, denn seine Zusatzfunktionen machen es zum universellen Kompaktlabor und prädestinieren es für Lernzwecke und analoge Experimente – nomen est omen!

„Richtige“ Oszilloskope mit Bildschirm und Co. kosten mittlerweile auch nicht mehr viel, bieten aber mit 25...40 MHz deutlich höhere Analogbandbreiten. Und Extra-Funktionen haben diese Resultate fernöstlicher Produktionskunst auch. Oder? Was also spricht für so eine kleine schwarze Plastik-Box von Digilent? Was kann sie besser als ein preiswertes Stand-Alone-Scope? Und 5 MHz – ist das nicht Spielzeug?

Eine Antwort vorweg: Wer lediglich ein preiswertes USB-Oszilloskop sucht, der ist mit dem Digilent-Produkt vielleicht wirklich nicht gut beraten. Studenten, Hobby-Labor-Aufrüstwillige und dezidierte Analog-Elektroniker sollten aber weiterlesen, denn der kleine Kasten mit den vielen Käbelchen kann mehr, als man vermuten würde.

Auspacken...

Auf der Webseite von Digilent, einem Hersteller von allerlei Zubehör und Modulen rund um Elek-

tronik, kann man für 239 \$ ein Vielzweck-Modul kaufen, mit dem man so ziemlich alles testen, messen und steuern kann, was im niederfrequenten Elektroniker-Alltag so anfällt. Dieser in Kooperation mit dem bekannten Halbleiter-Hersteller Analog Devices entwickelte kleine Apparat ist vollgestopft mit Funktionen (siehe **Technische Daten**) und daher eine prima Hardware-Grundlage für die Ausbildung von Elektronik-Ingenieuren im Grundstudium. Diese Zielgruppe erhält darüber hinaus auch noch spezielle Rabatte. Noch besser: Passend zum Gerät gibt es auch noch ein „Analog Parts Kit“ für 70 \$, das neben bekannten Opamps und CMOS-ICs im DIL-Gehäuse eine Menge an modernen SMD-ICs von Analog Devices auf kleinen Platinchen enthält, die man direkt auf ein mitgeliefertes Steckbrett stecken kann. Im Bundle gibt es das Kit mit einem Rabatt von 20 \$. **Bild 1** zeigt, was man bekommt, wenn man die Kombination von Analog Discovery + Parts

Technische Daten

Oszilloskop

- 2 Kanäle mit differentiellen Eingängen
- 5 MHz analoge Bandbreite
- 100 MS/s ADC-Abtastrate
- 14 bit ADC-Auflösung
- 1 M Ω / 20 pF Eingangsimpedanz
- Vertikal: 0,5 mV - 5 V pro Teilstrich, max. ± 25 V
- 16 KS Buffer
- Cross-Triggering mit den anderen Funktionen

Funktionsgenerator

- 2 Kanäle mit unipolaren Ausgängen
- 5 MHz analoge Bandbreite

- 100 MS/s DAC-Abtastrate
- 14 bit DAC-Auflösung
- Sinus/Dreieck/Rechteck etc.
- Selbstdefinierbare Kurvenformen
- Sweeps, Hüllkurven, AM- und FM-Modulation

Logic Analyzer

- 16 digitale Kanäle
- 100 MS/s Abtastrate
- 3,3 V Pegel, LVCMOS-kompatibel
- 16 K Pegelwechsel Buffer
- Trigger: Pin Change, Bus Pattern etc.
- Interpreter für SPI, I2C, UART, parallel

Sonstige Funktionen

- Voltmeter DC, AC mit True RMS
- Einstellbare Ausgangsspannung
- ± 5 V / 50 mA Ausgangsspannung
- Spectrum Analyzer mit vielen Modi
- Digitale IOs – alle 16 Pins einzeln steuerbar
- Digital Pattern Generator – bis 100 MS/s
- Network Analyzer – 1 Hz bis 10 MHz
- Datenübertragung und Versorgung via USB 2.0
- Waveform Generator – Signale auf 3,5-mm-Buchse
- Exportierbare Signaldaten
- MATLAB-Unterstützung

Kit bestellt hat. Öffnet man die grüne Box links, kommt neben einem USB-Kabel eine mit 82 x 63 x 19 mm recht kleine schwarze Plastik-Box zum Vorschein, aus der eine Menge farbiger Käbelchen herausragen (**Bild 2**). Ein paar lange, sechspolige Pin-Header gibt es gratis dazu. Auf der transparenten Sortiments-Box des Parts Kit sind alle enthaltenen Teile aufgelistet. Für jemand, der bauteile-mäßig nicht gerade üppig bestückt ist, bringt das Kit durchaus ein paar interessante Spezialitäten mit (**Bild 3**). Neben normalen Opamps, Transistoren, Widerständen, Trimpoties, Keramik-Cs und Elkos finden sich auch ungewöhnliche ICs wie eine Spannungsreferenz, ein Spannungs-Frequenz-Umsetzer, Komparatoren und etliche Sensoren zur Erfassung von Strom, Magnetfeldern, Beschleunigung, Schall, Temperatur, Vibration, Helligkeit und IR-Strahlung. Digilent hat auch an Kleinigkeiten gedacht: Neben dem praktischen Steckbrett liegen auch eine Menge Kabel mit Stiftenden und sogar ein kleiner Schraubenzieher bei. Wozu? Damit kann man die im Steckbrett steckenden Bauteile schön ohne Beschädigung wieder heraushebeln.

... und Ausprobieren

Da ganz bäume-schonend nicht ein einziges Blatt Papier beiliegt, gibt man zuerst den auf der Rückseite der Verpackung aufgedruckten Weblink [1] in einen Browser ein. Dort landet man nicht nur auf einer Bestellseite von Digilent, sondern kann von da auch die nötige Soft-

ware namens „Waveforms“ herunter laden, ohne die Hardware heutzutage ja keinen Sinn macht. Waveforms gibt es nicht nur für Windows-PCs, sondern auch in je einer Ausführung für i386-



Bild 1. Das kommt zum Vorschein, wenn man ein Bundle aus Analog Discovery + Parts Kit bestellt hat.

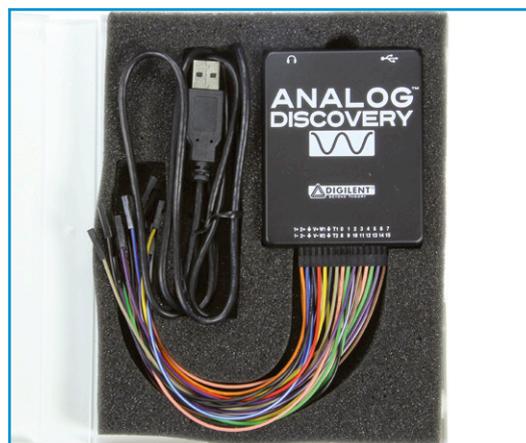


Bild 2. In der Verpackung von Analog Discovery steckt eine kleine Plastik-Box mit vielen Litzen und ein USB-Kabel.

und AMD64-Prozessoren unter Linux. Die Installation per Doppelklick ist trivial. Die Software sucht nach passender Hardware und verbindet sich dann. Auf meinem PC unter Windows 7 allerdings nicht sofort, sondern jedes Mal erst etwa 20 s nachdem Waveforms gestartet wurde. Dann funktioniert der Auswahl-Dialog von **Bild 4**, mit dem man die grundlegenden Funktionen (und die in den technischen Daten aufgeführten sonstigen Funktionen) starten kann. Auch die Spannung auf der USB-Leitung und der Stromverbrauch werden dort angegeben.

Bild 5 zeigt, wie das wohl wichtigste Messinstrument aussieht: das Oszilloskop. Es handelt sich um ein vollwertiges zweikanaliges Speicher-Oszilloskop mit allerhand netten Zusatz-Features. Gezeigt wird hier das Signal des Funktionsgenerators, der ein 200-kHz-Sinussignal erzeugt, das sowohl frequenz- als auch amplitudenmoduliert ist. Im unteren Teil des Fensters ist gleichzeitig das Spektrum dieses Signals abgebildet.

Bild 6 zeigt den arbiträren Funktionsgenera-

tor, der das vom Oszilloskop gemessene Signal erzeugt. Man sieht, dass alle Module unabhängig voneinander in separaten Fenstern betrieben werden können - ganz wie richtige Stand-Alone-Geräte. Die Einstell-Optionen demonstrieren, dass man dabei über viele Features verfügt. Die Möglichkeiten reichen durchaus an preiswerte Funktionsgeneratoren heran, die selbst aus fernöstlicher Produktion einen ähnlichen Preis haben wie das komplette Gerät von Digilent. Arbiträr darf sich das Generator-Modul deshalb nennen, weil man nicht nur die vorgefertigten Signalformen ausgeben kann, sondern auch Eigenkonstruktionen. Als Hilfsmittel hierfür genügt Excel.

Bild 7 zeigt das separate Modul des Spectrum Analyzers. Hier sind die typischen Nadeln eines 100-kHz-Dreieckssignals zu sehen (1f, 3f, 9f etc.), deren Amplituden mit steigender Frequenz abfallen. Die Einstellmöglichkeiten sind übersichtlich. Etwas störend ist, dass die Frequenzbereiche nicht stufenlos gewählt werden können, sondern fix vorgegeben sind. Es handelt sich halt nicht um einen High-End-Analyzer, doch für die einfachere analoge Praxis sind die Möglichkeiten sicher ausreichend. Zum Checken von Audiofiltern etc. reicht es allemal.

Hinzu kommen noch die „sonstigen Funktionen“, die in den technischen Daten aufgeführt sind. Ein Test von Schaltungen mit kleineren Mikrocontrollern ist zum Beispiel kein Problem. Man kann zwar keine Quarze höherer Frequenz ausmessen, aber ein Logic-Analyzer mit 16 digitalen Kanälen ist ja auch nicht schlecht. Damit verknüpft ist ein Mustergenerator, der parallele digitale Signale mit immerhin 100 MS/s ausgeben und dabei den kompletten Buffer für die erzeugten Sequenzen ausschöpfen kann. Besonders reizvoll ist, dass die digitalen IOs auch unabhängig voneinander als Ein- oder Ausgang geschaltet und zur Erzeugung und Anzeige digitaler Pegel genutzt werden können. Last not least sind die beiden analogen Ausgänge nicht nur mit entsprechenden Käbelchen auf der Vorderseite der Box verbunden, sondern auch mit einer 3,5-mm-Klinkenbuchse. Um eine Audioanlage zu testen, braucht es also nur minimalen Verkabelungsaufwand.

Experimente und Argumente

Die pure Multi-Mess-Box samt Bauteilen des Parts Kit würden im Unterricht und im Selbststudium nur wenig Sinn machen. Unter [1] sind daher fünf Einführungs-Videos und ein kompletter, passend zugeschnittener universitärer Elektronik-Grund-

Bild 3.
Das Parts Kit enthält nicht nur Standardbauteile, sondern auch interessante Sensoren, ein Steckbrett und andere Kleinigkeiten.

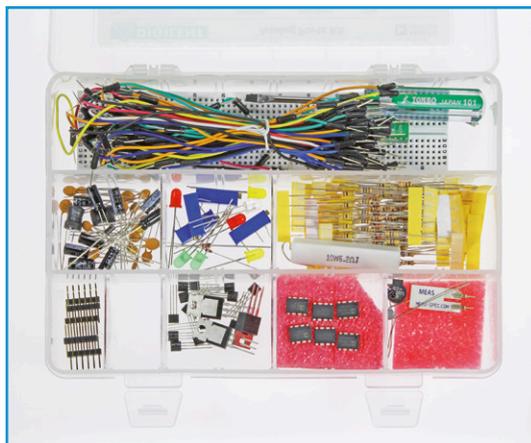
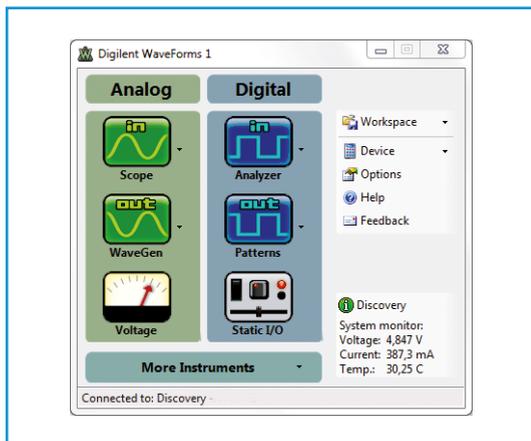


Bild 4.
Mit dem Auswahl-Dialog des Programms Waveforms startet man die Hauptfunktionen.



kurs enthalten. Dieser ist allerdings nichts für angehende Elektroniker, die mit Mathematik auf Kriegsfuß stehen.

Bleibt die Frage, warum so eine kleine eierlegende Wollmilchsau in bestimmten Bereichen auch für Nichtstudenten eine passende Ausrüstung für das kleine Elektronik-Labor darstellen kann. Analog Discovery bietet nämlich gegenüber preiswerten Stand-Alone-Oszilloskopen eine ganze Menge handfester Vorteile. Zuallererst wäre der Platzvorteil zu nennen: Einen PC hat man sowieso und das Ding ist wirklich klein und universell. Es ersetzt im Sub-Profi-Bereich eine ganze Menge einzelner Geräte, die nicht nur mehr Platz wegnehmen, sondern zusammen genommen auch deutlich teurer wären.

Außerdem wichtig ist der viel größere Bildschirm eines PCs. Selbst wenn ein preiswertes Stand-Alone-Oszilloskop mit 7"-Bildschirm und „sagenhaften“ 800 x 600 Pixeln beeindruckend möchte, kommt es doch bei weitem nicht an diese integ-

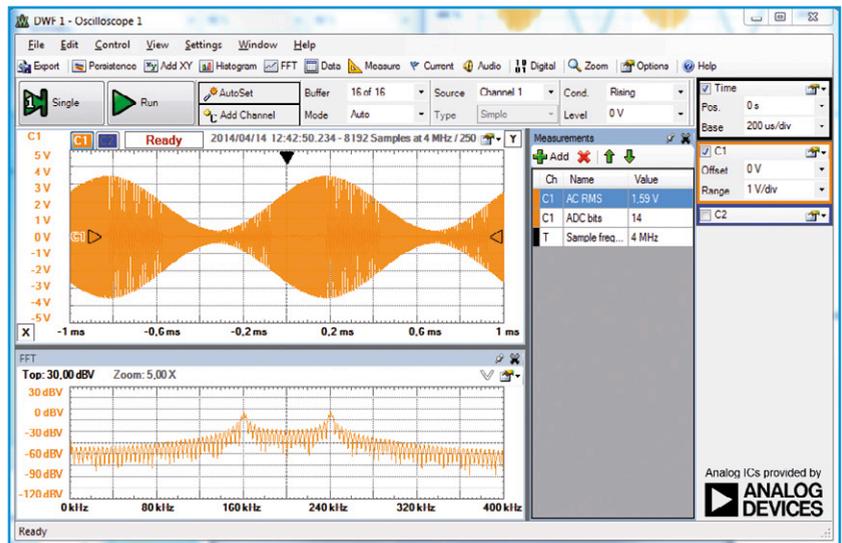


Bild 5. So sieht das Oszilloskop-Modul von Waveforms aus. Hier wird ein mit Analog Discovery erzeugtes, moduliertes Sinussignal samt Spektrum angezeigt.

Anzeige

Mikrocontroller verstehen und anwenden

Schnell und einfach mit Arduino und Elektor-Shield



Mit diesem Buch erweitert der Leser seine Mikrocontroller-Kenntnisse auf Grund eigener Erfahrungen und Erfolgserlebnisse und wird dazu noch ganz nebenbei in die Welt des Arduino und seiner Entwicklungsumgebung eingeführt. Am Ende dieses vergnüglichen und fast spielerischen Lehrgangs stellen Begriffe wie I/O, Speicherplatz, Interrupts, Kommunikationsstandards, A/D-Konverter (und vieles mehr) keine Geheimnisse mehr dar und der Leser ist in der Lage, auch andere Mikrocontroller zu programmieren. Mit anderen Worten: ein erstes Mikrocontroller-Buch mit Happy End.

Dieses Buch ist für Sie geeignet, wenn Sie Anfänger auf dem Gebiet der Mikrocontroller sind, als Arduino-User bzw. -Enthusiast Ihre Kenntnisse vertiefen möchten, Elektronik studieren oder als Lehrer inspiriert werden möchten. Alle gezeigten Programme können kostenlos von der Elektor-Website heruntergeladen werden.

350 Seiten (kart.) • Format 14 x 21 cm
ISBN 978-3-89576-296-3

€ 42,00
CHF 52,95

Weitere Infos & Bestellung unter www.elektor.de/mcu-verstehen-anwenden

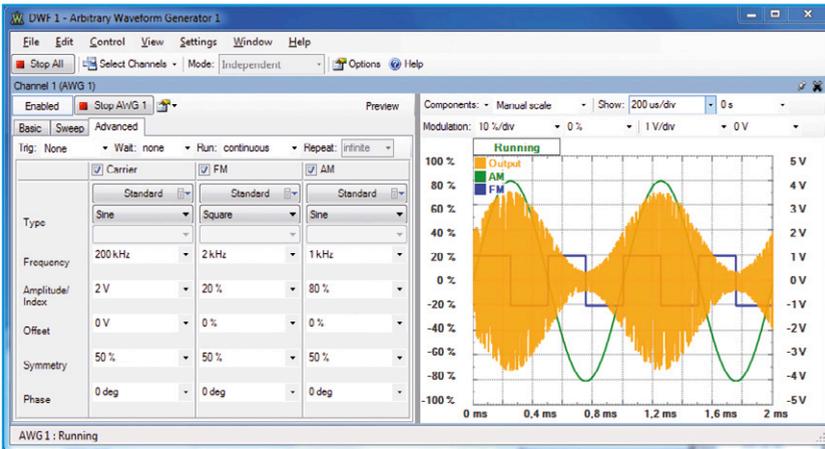


Bild 6. Der arbiträre Funktionsgenerator erzeugt das in Bild 5 gemessene 200-kHz-Sinus-Signal, das gleichzeitig frequenz- und amplitudenmoduliert wurde.

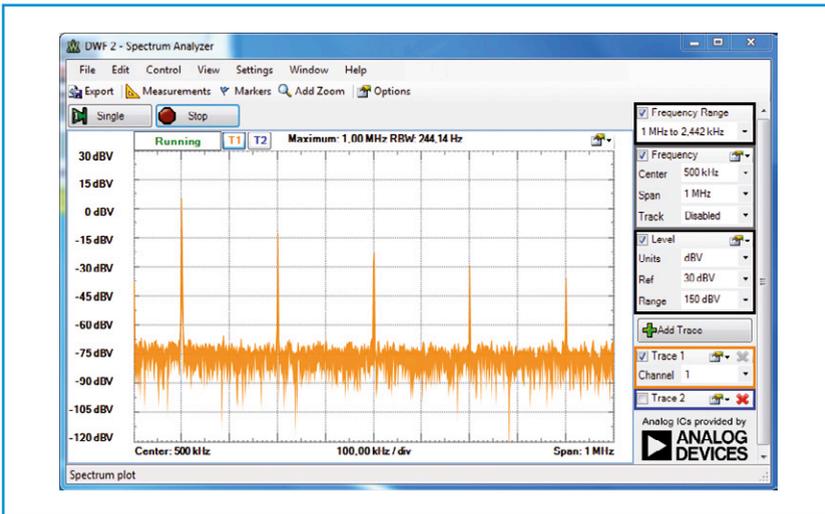
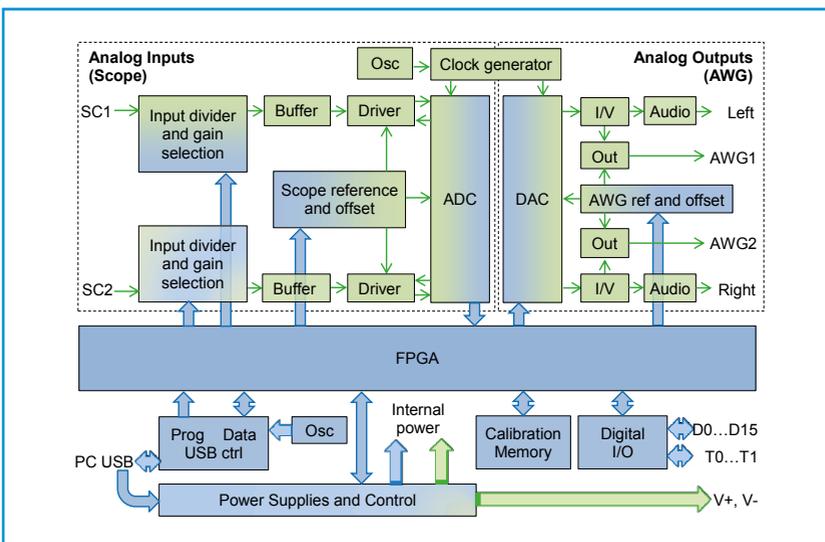


Bild 7. Der Spectrum Analyzer bietet nicht so viele Funktionen. Hier sieht man das Spektrum eines 100-kHz-Dreieckssignals.



rierte Lösung heran. So ein fertiges Fernost-Scope hat nämlich normalerweise ADCs mit einer Auflösung von nur 8 bit. Die vertikale Auslenkung eines Signals kommt daher nicht über 256 Pixel hinaus, außer mit vergrößernder Pixelvervielfachung. Da spielen die 14 bit von Analog Discovery schon in einer ganz anderen Liga. Zudem: Die Fenster des Waveforms-Programms können stufenlos größer oder kleiner gezogen werden. Zu guter Letzt ist die Software für preiswerte Stand-Alone-Sopes auch bei vorhandener USB- oder Netzanbindung oft eher rudimentär und alles andere als gut bedienbar. Waveforms hat hier ebenfalls die Nase vorn. Auch wenn ein Stand-Alone-Scope mehr Bandbreite hat: Mit Analog Discovery bekommt man einen arbiträren Funktionsgenerator und mehr geschenkt dazu. Weiter positiv: Unter [1] kann man sich ein PDF herunterladen, bei dem der komplette innere Aufbau von Analog Discovery detailliert beschrieben ist. Wo hat man sowas schon?

Wo Licht ist, ist auch Schatten: Der gravierendste Nachteil ist sicherlich, dass Analog Discovery nicht potentialgetrennt ist. Bei analogen Signalen spielt das bei Potentialdifferenzen von einigen Volt keine Rolle, da ja differentielle Eingänge vorhanden sind. Die Pegel der digitalen IOs etc. beziehen sich allerdings auf eine Masse, die direkt über USB mit der PC-Masse verbunden ist. Hier muss man also wissen, was man tut.

(120585)

Weblink

[1] Weiterführende Informationen, Anleitungen und Software:

www.digilentinc.com/AnalogDiscovery

Das interessante Gerät haben wir nun auch in den Elektor-Shop aufgenommen (www.elektor.de).

Bild 8. Blockschtung des Innenlebens von Analog Discovery. In einem PDF sind auch Schaltungsdetails enthalten.

Alle Elektor-Artikel der Jahre 2000 bis 2009 auf DVD!



NEU!

**Ein Muss
für jeden
Elektor-Leser!**

- Elektronisches Jahrgangs-Archiv, Artikel als PDF, schnelle Suchfunktion
- 110 Elektor-Hefte, über 4000 Artikel, bequeme Druckfunktion
- Ideen, Schaltungen und Projekte für Elektroniker im Beruf, in der Weiterbildung und der Freizeit

ISBN 978-3-89576-292-5
€ 89,00 • CHF 110,40

Jetzt unter www.elektor.de/2000-2009 bestellen!

Elektor Computerskop (1986)

Tschüss grüne Röhre



Von **Jan Buiting**
(Chefredakteur
Elektor-UK/US)

Damals im Jahre 1986, irgendwo im Elektor-Labor, kam ein angehender niederländischer

Ingenieur auf die Idee, eine Oszilloskop-Erweiterung für einen PC zu entwickeln. Nur wenige Monate zuvor hatte ich bei Elektor als Redakteur angefangen. Da ich mit einer elektrischen Schreibmaschine und einem Wörterbuch umzugehen wusste, fiel mir die Aufgabe zu, der Halbleiterabteilung von ITT im badischen Freiburg einen Brief zu schreiben und um ein paar Muster ihres neuesten Chips UVC3101 zu bitten. Mir wurde gesagt, es handle sich um einen „High-speed“-ADC und -DAC in

einem Chip, der „sehr schwer erhältlich“ wäre. Kurz darauf rief mich Reinhard W. Preuss an, der wohl dienstefrigste und vornehmste Pressesprecher, den ich in den ganzen Jahrzehnten seither im Bereich Elektronik kennenlernen durfte. Nur wenige Tage später waren die ICs da. Diese Lieferung ermöglichte es Rene van Linden, mit seinem Computerskop-Projekt zu beginnen und damit gleichzeitig seine Abschlussarbeit zu bestreiten.

Damals...

Damals war es sehr „in“, mit PCs anderes zu tun als zu kalkulieren, zu schreiben und Pac-Man zu spielen. Damals waren noch Maschinen wie MSX-Computer, Commodores C64 und Ataris ST aktuell, und es gab die ersten IBM-PC-Kompatiblen. Damals lag sowohl die Hardware als auch die Software offen und war zugänglich für alles Mögliche. Solange ein Computer einen Port hatte, dessen Stecker einigermaßen erhältlich war, schraubten Schüler, Ingenieure und Elektor-Entwickler daran herum und entwickelten passende Peripherie, die vom Hersteller nicht vor(her)gesehen war. Überall hörte man Ausdrücke wie POKE(16xxx)...

Beim Elektor-Computerskop ging es darum, einen PC die gemessenen Daten auf seinem Monitor anzeigen zu lassen. Paradoxaerweise entspricht ein Röhrenmonitor technisch ja weitgehend einem Oszilloskop, nur dass Ersterer größer ist und manchmal sogar farbig. Von daher war das kaum preiswerter als ein (gebrauchtes) Oszilloskop von Tektronix oder ein neues von Hameg.

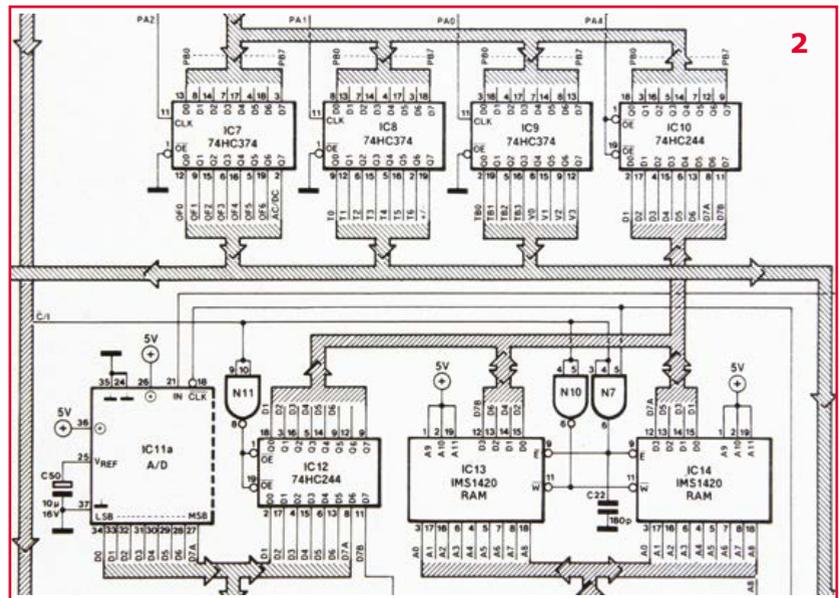
Große Projekte wurden damals gerne in zwei Teilen in Elektor veröffentlicht. Der erste Teil erschien im September 1986 und brachte die Schaltung, erklärte die Grundfunktionen und zeigte Screenshots einfacherer Wellenzüge. Im Oktober des gleichen Jahres ging es um den Bau, die Einstellungen und die Software. Das Computerskop schmückte den Titel der September-Ausgabe (**Bild 1**). Interessanterweise sind da gleich



zwei Computer abgebildet: Prominent im Vordergrund der BBC Micro und dahinter noch ein Commodore C64.

Die Schaltung

Das Computerskop brauchte unbedingt eine breite und schnelle Datenautobahn zum Speicher und zur Grafik des Computers. Computer = digital – doch die zu messenden Signale sind (meistens) analog. Also musste ein ADC dazwischen. Die Verbindung von Computer und Scope war keine Einbahnstraße, denn der Computer musste ja Einstellungen wie die AC/DC-Umschaltung oder den Eingangsabschwächer/Verstärker bedienen können. Hierzu wurden einige DIP-Relais eingesetzt. Das Computerskop hatte nämlich weder Knöpfe noch sonstige Bedienelemente. Das fertige Gerät im Gehäuse sieht sogar fast langweilig aus,

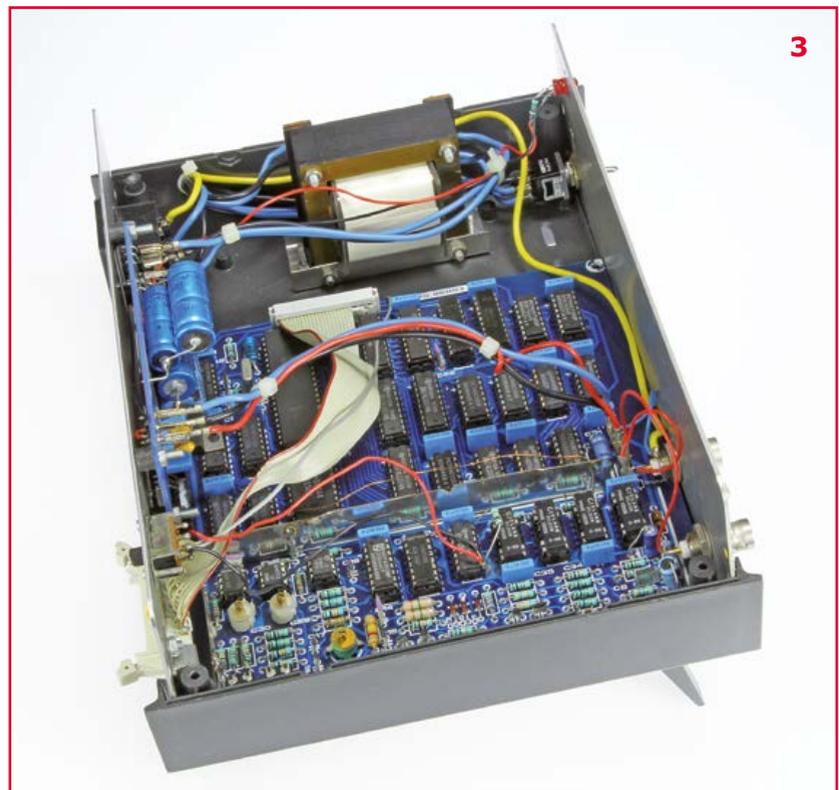


Kompatibel auf Papier mit Acorn Electron, BBC Micro, C64, MSX, IBM PC?

wenn man von der geschwungenen Welle auf der Frontplatte absieht.

Das Herz der Schaltung war das IC UVC3101 mit seiner „schnellen“ ADC-DAC-Kombination; es wurde vom kleinen RAM IMS1420 von Inmos unterstützt. Diese ICs waren damals selten und teuer. Gut, dass einige Händler Bausätze zusammenstellten und diese in Elektor anboten.

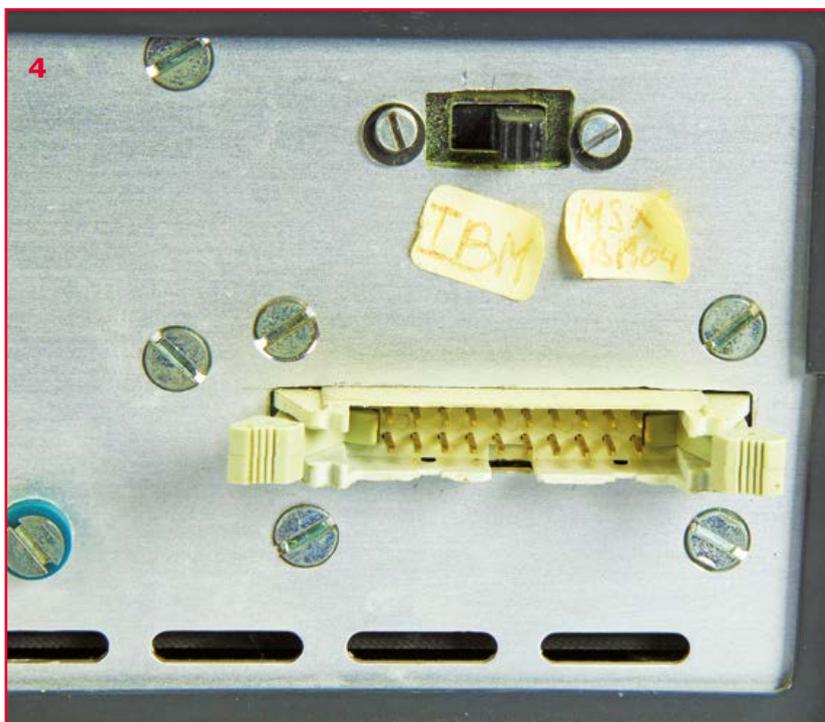
Der Schaltungsteil mit ADC und RAM ist in **Bild 2** zu sehen. Der (hier nicht sichtbare) DAC wurde bei Bedarf zur Erzeugung eines Offsets des Eingangssignals verwendet. Der ADC konvertierte mit einer Auflösung von 7 bit. Bit 8 war der Trigger-Indikator. In den Speicher passten 512 Byte Daten. Für das RAM gab es einen eigenen Adresszähler, der mit TTL-ICs der Reihe 74HC realisiert war. Damals gab es noch keine Mikrocontroller. Die Platine war vollgestopft mit 74HC-ICs, und hie und da kamen noch 74LS- und CD400-Chips dazu. Es mussten eben eine Menge Steuer- und Timing-Aufgaben erledigt werden, darunter auch die komplette Zeitbasis. Ich konnte 24 Logik-ICs zählen (**Bild 3**) – sowas nannte man damals spöttisch „Bauteilegrab“.



Die Elektronik wurde von einer extra Spannungsregler-Platine (EPS 9968-5) gespeist.

Eine kleine Modifikation erlaubte dem Gerät mittels eines kleinen Schiebeschalters auf der Rückseite (**Bild 4**) den Betrieb mit einem IBM-PC(-Kompatiblen).

Wer an den kompletten Artikeln von 1986 interessiert ist, der sei auf die 1980er DVD von Elektor verwiesen (www.elektor.de/80-89).



Diese Listings...

Wenn man die Platine zum Computerskop #86083 via EPS (Elektor-Platinen-Service) bestellte, gab es die Programme in Form von Listings auf Papier für Acorn Electron, BBC Micro, Commodore C64 und MS-Computer. Ich erinnere mich daran, wie diese Papierstapel bei uns im Lager neben den Platinen herumlagen und sich bei mir auf dem Schreibtisch die Bittschreiben aus der ganzen Welt häuften, man möge doch die Listings auf Diskette oder auch ohne Platine verschicken. Um Besitzern nicht bedachter Computer dabei zu helfen, eigene Software auf ihren Systemen zum Laufen zu bringen, wurde am Schluss des Oktober-Artikels noch ein grobes Flussdiagramm abgedruckt.

Love Story

Wenn man die Namen von zwei Elektor-Mitarbeitern zwischen der Oktober- und der November-Ausgabe vergleicht (dort auf Seite 13), fällt ein veränderter Nachname auf. Außer dass der frühere Praktikant R (für Rene) v(an) Linden seinen Ingenieurs-Abschluss mit guten Noten für das Computerskop erhalten hatte, hat sich im Laufe des Jahres 1985 ergeben, dass unsere allseits beliebte Sekretärin W (für Wilma) Wijnen für ihn mehr als nur eine Kollegin wurde.

Rettung

Elektor ist im Jahre 2006 umgezogen und wollte damit auch einen Teil der Vergangenheit loswerden. Techniker und Redakteure wurden angehalten, ihre alten Büros zu verlassen und einen kompletten Neustart im Elektor Castle hinzulegen. An einem Freitag-Abend wurde das alte Labor in Beek (NL) unter Aufsicht des Chefs von Freiwilligen rigoros gesäubert. Tonnenweise Material verschwand in einem x m³ großen Müll-Container, der in der Einfahrt direkt unter den Fenstern des Labors positioniert war. Es gab Bier und Pizza zu Belohnung der Helfer und zur Feier des nunmehr fast leeren Gebäudes.

Von den etwa 20 Laborprojekten und Prototypen, die ich vor dieser Operation Augias retten konnte, waren die meisten „fertig“, d.h. es war ein Gehäuse drumrum. Es gelang mir, diese Teile vor der Zerstörung zu bewahren. Schön war, dass einige Kollegen ebenfalls zu solchen Rettungsaktionen schritten. Heute sind die wenigen überlebenden Geräte dieses wilden Freitags in die Retronik-Sammlung integriert, die jetzt zum „Bestand der Firma“ avancierte. Man kann sie im Netz unter www.elektor-labs.com/attic bestaunen. Dort im Regal, im Fach oberhalb des „Mugen Hybrid Amp“, befindet sich der endgültige Standort des Elektor Computerskops von 1986.

(140034)

EST^D 2004

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

Hexadoku Sudoku für Elektroniker

Auch in der Sommer-Ausgabe darf ein Hexadoku natürlich nicht fehlen, das zwischen dem Studium all der großen und kleinen Schaltungen für etwas Abwechslung sorgt. Wie immer warten die grauen Kästchen darauf, mit den richtigen Ziffern gefüllt zu werden. Machen Sie mit und gewinnen Sie mit etwas Glück einen von fünf Elektor-Buchgutscheinen!

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

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

Wer das Rätsel löst - sprich die Zahlen in den grauen Kästchen herausfindet - kann einen von fünf Buchgutscheinen im Wert von 50 Euro gewinnen!

Einsenden

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

Elektor – Redaktion – Süsterfeldstr. 25 – 52072 Aachen

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

Als Betreff bitte nur die Ziffern der Lösung angeben!

Einsendeschluss ist der 31. August 2014!

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

Die richtige Lösung ist: **18047**.

Einen Elektor-Buchgutschein über je 50 € haben gewonnen:

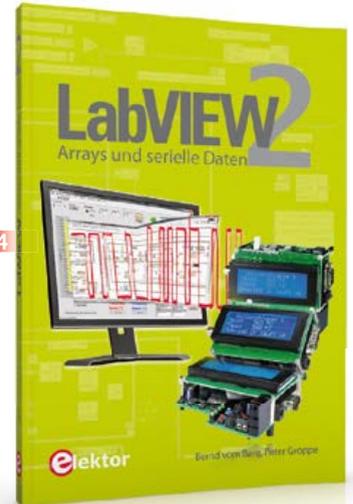
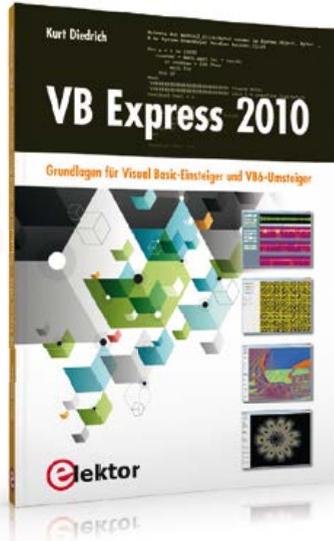
Alex Lo Furno, Rainer Klein, Luc Vandormael, Thomas Eriksen und Doug Brown.

Herzlichen Glückwunsch!

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

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

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



NEU

Theorie und Praxis mit Multisim
1 Grundsaltungen der Elektronik

Dieses Buch ist ein Nachschlagewerk über Elektronik mit praxisorientierten Fakten und ausführlichen Erklärungen. Der Autor hat selbst für komplexe Vorgänge oder Formeln praktische kurze Erklärungen und Näherungsrechnungen entwickelt, ohne die Darstellungen zu simplifizieren. Als Ausgangspunkt wurde das Simulationsprogramm Multisim gewählt, das zahlreiche Bauelemente und umfangreiche Messinstrumente zur Verfügung stellt.
360 Seiten (kart.) • ISBN 978-3-89576-286-4
€ 44,00 • CHF 54,60

Grundlagen für Visual Basic-Einsteiger und VB6-Umsteiger
2 VB Express 2010

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

der Leser ausführlich kommentierte Beispielprogramme, die er selbst ausprobieren kann und die sich auf das Mindeste beschränken, was zum Starten der Software notwendig ist.
284 Seiten (kart.) • ISBN 978-3-89576-269-7
€ 34,80 • CHF 43,20

45 Experimente mit Hard- und Software für Elektroniker
3 Raspberry Pi

Dieses Buch beschreibt 45 spannende und interessante Projekte mit Raspberry Pi, wie zum Beispiel ein Wechselblinklicht, eine Motorregelung, Erzeugen und Verarbeiten analoger Signale, ein digitales Thermometer, ein Lichtmesser. Aber auch kompliziertere Projekte wie eine Motor-Geschwindigkeitsregelung, ein Webserver mit CGI (Common Gateway Interface) und Client-Server-Programme werden vorgestellt. Sie können dieses Buch als Projektbuch verwenden und die Projekte nachbauen, um sie dann in der Praxis einzusetzen.
271 Seiten (kart.) • ISBN 978-3-89576-273-4
€ 39,80 • CHF 49,40

Arrays und serielle Daten
4 LabVIEW 2

Der zweite Band der LabVIEW-Lehrbuchreihe beschäftigt sich u.a. mit Arrays, Cluster und den seriellen VISA-Funktionen. Als Erstes werden vier neue zusammengesetzte Datentypen (Enum, Ring, Array, Cluster) vorgestellt und deren Verwendung wird anhand zahlreicher praktischer Beispiele und Übungen erläutert. Danach wird es praktisch: Ein 8051er-Mikrocontrollersystem dient dabei als Datenquelle und -senke für verschiedene LabVIEW-VIs.
248 Seiten (kart.) • ISBN 978-3-89576-274-1
€ 34,80 • CHF 43,20

Der professionelle Ratgeber für Funkempfangstechnik
5 Funkempfänger-kompodium

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



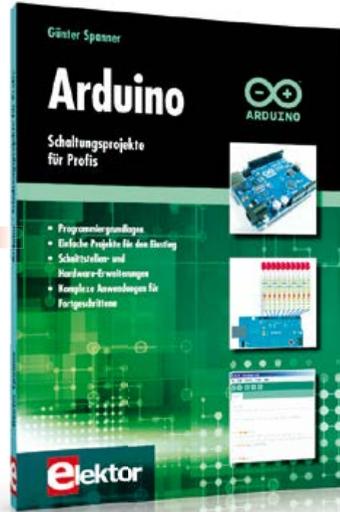
5

auch ausgefallene Anwendungen von Empfängern kennenlernen und wissen, wie ein Software Defined Radio (Digitalempfänger) nun wirklich funktioniert und was der letzte Stand der entsprechenden Technik kann? In diesem Buch findet man die Antworten!
397 Seiten (geb.) • ISBN 978-3-89576-276-5
€ 49,00 • CHF 60,80

Schaltungsprojekte für Profis

6 Arduino

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



6

stets größter Wert auf eine praxisorientierte Ausrichtung gelegt.

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

91 spannende Geschichten elektronischer Antiquitäten

7 Retronik

Dieses Buch ist eine Zusammenstellung von 91 Retronik-Beiträgen, die zwischen 2004 und 2013 veröffentlicht wurden. Bei diesen Geschichten geht es um alte Messgeräte, prähistorische Computer, längst vergessene Bauteile und populäre Projekte aus älteren Elektor-Ausgaben. Alle Geschichten haben Eines gemeinsam: Der Kenner schmunzelt, staunt und schwelgt nostalgisch in längst vergangenen elektronischen Zeitaltern.
230 Seiten (kart.) • ISBN 978-3-89576-291-8
€ 32,80 • CHF 40,70

30 Projekte in C für Fortgeschrittene

8 ARM-Mikrocontroller 2

Die im Buch beschriebenen Projekte mit dem mbed-Board sind für Einsteiger in C und ARM-Mikrocon-



7



8

troller ausgelegt. Der mbed NXP LPC1768 nutzt Cloud-Technologie, ein revolutionäres Konzept in der Software-Entwicklung.

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

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

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

www.elektor.de

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

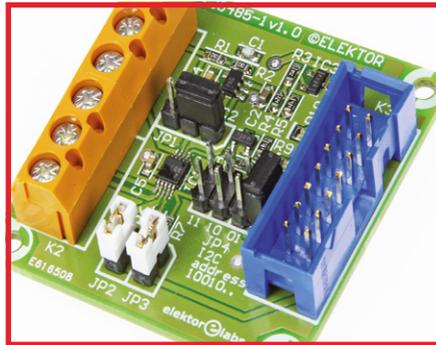
•Nächsten Monat in Elektor



Schaltendes Labornetzgerät

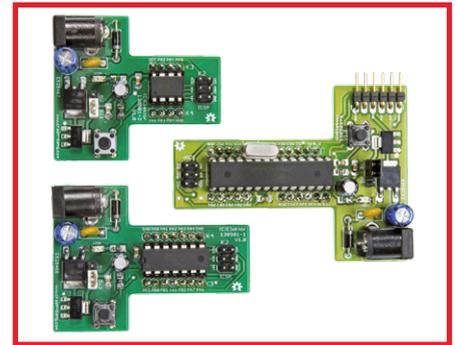
Bereits in der letzten Ausgabe angekündigt, doch wegen „Überlänge“ auf den September verschoben:

Ein Labornetzgerät mit innovativem Flachtransformator, bei dem Drahtwicklungen durch Platinenbahnen ersetzt wurden. Bis 30 V und 1 A kann dieses Netzgerät liefern. Zwei LED-Displays zeigen die Werte von Spannung und Strom an, und bei Gefahr trennt ein Relais den Ausgang von der Last.



16-bit-ADC-Modul

Für Datenlogger-Anwendungen wurde im Elektor-Labor ein ADC-Modul mit dem ADS1115 von TI entwickelt. Auf der Platine befindet sich ein Gnulbin/EEC-Steckverbinder, so dass das Modul unmittelbar mit dem Elektor-Linux-Board, dem multifunktionalen Xmega-Board und dem neuen Elektor Arduino-Shield gekoppelt werden kann. Eine universell verwendbare C-Bibliothek macht die Integration in eigene Projekte einfach.



Prototyping mit T-Boards

Steckboards sind für das Entwickeln von Mikrocontroller-Systemen nicht immer das Non-Plus-Ultra. Schnell entsteht ein Dickicht aus Verbindungsleitungen, das den Durchblick erschwert. T-Boards erleichtern den Entwickler-Alltag, denn dort ist die grundlegende Hardware des auserkorenen Mikrocontrollers bereits aufgelötet. So ein T-Board lässt sich auf ein Steckboard setzen, aber auch einfach in ein Platinen-Projekt integrieren.

Änderungen vorbehalten. Elektor September 2014 erscheint am 20. August 2014. Verkaufsstellen findet man unter www.pressekaufen.de.

Rund um die Uhr und
sieben Tage die Woche

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

elektor@labs
Sharing Electronics Projects

Home Proposals In Progress Finished

elektor PCB Service www.elektorpcbservice.com
Quick - Cheap - Secure

Need some stuff?
Visit the **Elektor Labs** store!

Proposals In Progress Finished

Active Popular

Q&A #3
YU NO ASK ABOUT
electronics?

Create a Project
Create a new project or enter a proposal
Get help, feedback & votes from other visitors,
and maybe you will get Elektorized too!



**Lesen Sie jetzt Elektor
auf Ihrem Tablet!**

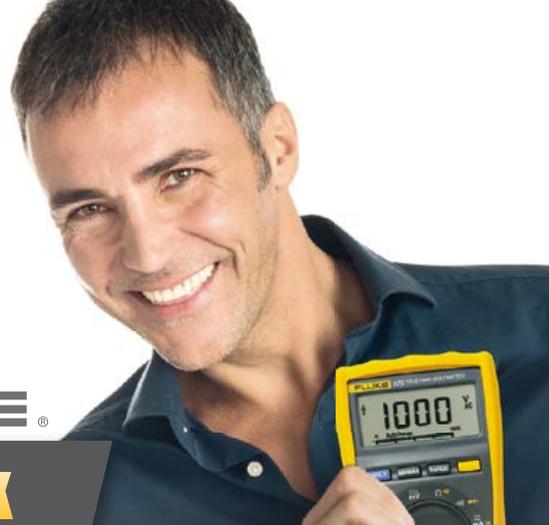
**Einzelheft: nur 7,99 €
Doppelheft: nur 14,99 €**

**Elektor-Mitglieder erhalten
jede App-Ausgabe kostenlos!**

• Netzteil-Recycling : Elektronik einfach steuern
Preisender für Flugmodelle : Akkulader ohne Mikrocontroller
Resonanzmeter : FPGAs pro
• Neues aus dem Labor •

Elektor-App herunterladen:

- www.elektor.de/ios-app
- www.elektor.de/android-app



Professionelle Qualität zu attraktiven Preisen!

- ✓ 45 Jahre Erfahrung
- ✓ über 50.000 Produkte am Lager
- ✓ schneller 24-Std.-Versand
- ✓ kein Mindermengenzuschlag

FLUKE®

Profi-Messtechnik

für Industrie, Handwerk und Beruf

Kundenbewertungen



Rund 97% unserer Kunden sind vom reichelt-Service überzeugt!
* Quelle: Shopauskunft.de (3. 6. 2014)



Das FLUKE-Sortiment

Visuelles IR-Thermometer

Punktgenaue Temperaturmessung kombiniert mit den Vorteilen einer Wärmebildkamera



- ✓ Messbereich: -10 bis 250 °C
- ✓ Genauigkeit: ±2 °C oder ±2 %
- ✓ Gesichtsfeld: 28° x 28°

- integrierte Digitalkamera
- 5 Überblendungsmodi mit Wärmekarte
- intelligent & vollautomatisch
- professionelle Berichtserstellung mit SmartView® Software
- automatische Aufspürfunktion für heiße/kalte Messpunkte
- Alarm- & Zeitrafferfunktionen
- inkl. umfangreichem Zubehör

FLUKE VT04

749,00

TRMS-Digital-Multimeter

Beeindruckend leistungsfähig, äußerst anwenderfreundlich, hochgenau, sicher und sehr zuverlässig – das Fluke 175 setzt neue Maßstäbe!

- manuelle und automatische Bereichswahl
- Frequenz-, Kapazitäts-, Widerstandsmessung, Durchgangsprüfung, Diodentest
- Min/Max/Mittelwert-Aufzeichnung
- Smoothing-Modus für stabilere Messwerte
- EN 61010-1, CAT III 1000 V, CAT IV 600 V
- inkl. Batterie, Messleitungen und Anleitung



Top-Seller!

FLUKE®

- ✓ Echtheffektivmessung von Spannung und Strom
- ✓ Grundgenauigkeit: 0,1%
- ✓ 6.000 Digits Auflösung

FLUKE 175

199,00

WearGuard™ zeigt Beschädigungen der Messleitungen

Extrem strapazierfähige Zugentlastung

TwistGuard™ Messleitungssatz

FLUKE®

- einstellbare Messspitzenlänge: 4-19 mm
- zugelassen für: CAT II 1000 V, CAT III 1000 V, CAT IV 600 V, max. 10 A
- Verschmutzungsgrad 2

FLUKE TL175 26,85



NEU! FLUKE 6500-2

Für schnellere und einfachere Messungen nach VDE 0701/0702 an ortsveränderlichen Geräten.

Prüfgerät für Geräte und Betriebsmittel

- umfangreiche Messfunktionen
- Eintastenbedienung und Tastatur
- voreingestellte Werte für gut/schlecht sowie Automatik-Testsequenzen
- großes Grafik-Display mit Hintergrundbeleuchtung
- Einzelnetzsteckdose & separate IEC-Steckdose
- USB-Port zur Speicherung für Datenübertragung
- Inkl. Anleitungs-CD, Messleitung, Prüfspitze, Kokodilklemme, Netzkabel, USB-Stick, USB-Kabel



FLUKE 6500-2 1729,00

FLUKE 6500-2 Kit
Die Kit-Version enthält zusätzlich einen erweiterten Testadapter, einen Barcodeleser, DMS 0702/PAT Software und Aufkleber für Bewertungen und Barcode-Anwendungen. (Lieferung ohne Notebook)

FLUKE 6500-2 KIT 2584,00

FLUKE®

Aktuellste Mess- und Sicherheitstechnologie



Spannungs- & Durchgangsprüfer

- LCD-/LED-Anzeige, beleuchtet
- integrierte Taschenlampe
- Gleich-/Wechselspannung: 6-690 V
- Durchgangstest mit Signal & Vibration
- Drehfeldanzeiger, Frequenzmessung, Widerstandsmessung
- Funktion zur Prüfung von RCDs
- einpolige Phasenprüfung
- IP-Spezifikation: IP64

EN 61010-1 CAT III 690 V

EN 61010-1 CAT IV 600 V

FLUKE T150 VDE

144,00



Jetzt bestellen: www.reichelt.de

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

Katalog 06/2014

Kostenlos – Jetzt anfordern!

