

JETZT GANZ NEU!

elektor

LEARN • DESIGN • SHARE



CODEBENDER
Arduino-Coding im
Web-Browser



NICHT VERGESSEN!
Für Medizin u.a.:
Uhr mit 9 Alarmen



SCHLAU LÜFTEN
Luftfeuchte regeln
mit ATmega328

USB-nach- Multi-Protokoll- Konverter

Für UART, SPI, I²C, ...



Diese Ausgabe:
3 Labor-Projekte
1 Programmierkurs
5 neue Module & Platinen
3 Leser-Projekte
1 Review und
noch viel mehr!

UART/RS232-Datenlogger mit isolierten Eingängen • AM-Sender mit Arduino • **ARM-Controller für Einsteiger (3): Rund um die Timer** • **Ins IoT mit WunderBar** • Einfacher Transistortester • **Preiswerter LED-Hochleistungstreiber** • **Bluetooth Low Energy Modul: Programm editieren, kompilieren und übertragen** • **Reingeschaut: Im Innern des Apfels** • Web-Recherche: Optimal messen mit dem Oszilloskop • **Embedded World 2015** • **Seltene Bauteile: Der Gyator**



4 198630 307906

Wo ist Ihr Fahrzeug? Ihr Paket? Ihre Maschine? Finden Sie es mit TraceME!

Das neue Modell TM-203

TM-203 ist die neueste Ergänzung unserer TraceME-Produkt-Reihe.



Besondere Features:

- Sehr kleiner GPS-Tracker mit allen Features (87 x 22 x 7 mm)
- Wiegt nur 15,5 g, inkl. GPS-, Mobilfunk- und HF-Antennen
- Besonders niedriger Stromverbrauch, bis hinab zu 3 μ A
- Positionsbestimmung via LBS sogar innerhalb von Gebäuden
- Portabel oder extern anschließbar zur Integration der Sensoren
- On-Board-Sensoren für Kompass, Temperatur und Beschleunigung
- Weltweite Abdeckung: Quad-Band-GSM/GPRS - Glonass/GPS - WLAN
- Entspricht zusätzlichen Standards von Versicherern (z.B. SCM und Insert)
- Entspricht den Sicherheits-Standards der Automobil-Industrie (E-mark)



Wir möchten Sie gerne zu einem Besuch auf unserer Webseite einladen. Lassen Sie sich von den Möglichkeiten inspirieren, die KCS-TraceME-Produkte für Ihr Projekt bringen können. Ob Sie an Personal Tracking, der Ortsbestimmung von Fahrzeugen, dem Objektschutz oder fortschrittlichen M2M-Lösungen interessiert sind – KCS TraceME hat die Lösung für Sie!

Wir bieten GPS-Tracker mit GPS/GPRS und erweiterter I/O. Damit können Sie eine breite Palette an Objekten, Maschinen oder Fahrzeugen aus der Ferne verfolgen oder aufspüren. KCS TraceME bietet aktuell die flexibelsten und anpassungsfähigsten Track-n-Trace-Lösungen. Unsere Kunden setzen TraceME-Hardware für die unterschiedlichsten Aufgaben ein. Wir bieten echte Allround-Lösungen, die andere Produkte in den meisten Aspekten übertreffen.

- KCS-TraceME-Produkte gibt es in vielen Ausführungen (Key fob, OWM, Alu- und kundenspezifische Gehäuse)
- GPRS – UMTS – SMS – GSM – WLAN – RFID – E-Mail
- Voll konfigurierbar mit Administrations-Tool
- Event-gesteuert, kann kundenspezifischen Code verarbeiten, fernkonfigurierbar
- Verbindet M2M mit jedem anderen Gerät
- G-Schock-Erkennung, Temperatur, Drehung, Bewegung, iButton, Handy, Kamera
- Die kompletten und detaillierten Optionen finden Sie auf unserer Webseite!

Sprechen Sie mit uns! Wir diskutieren gerne mit Ihnen über Ihre Ideen!

Anfragen von OEMs & Distributoren willkommen!

www.Trace.ME

Alle erwähnten Markenzeichen gehören den jeweiligen Eigentümern

Impressum

46. Jahrgang, Nr. 532 April 2015
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: margriet.debeij@eimworld.com

Julia Grotenrath

Tel. 02 41/88 909-16 / Fax 02 41/88 909-77
E-Mail: julia.grotenrath@eimworld.com

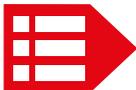
Es gilt die Anzeigenpreisliste Nr. 45 ab
01.01.2015

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.

© 2015 elektor international media b.v.
Druck: Senefelder Misset, Doetinchem (NL)
ISSN 0932-5468



Der Pico-Browser

Letzte Woche habe ich zum ersten Mal etwas in unserem Shop bestellt: Ein Raspberry Pi 2 mit Gehäuse durfte es sein. Ich wollte mich sowieso endlich mit dem RasPi beschäftigen; die Aussicht, dass ich hier irgendwann Windows zum Laufen bekommen könnte, lockte mich nun doppelt. Mein

Skript-Interpreter, den ich oft benutze, um PC-Software mit HTML-basierten Benutzeroberflächen und Datenbankabfragen zu realisieren, eignet sich auch gut dafür, Infos aus dem Internet abzurufen. Schauen Sie mal das Foto auf Seite 9 an: Unser Arduino-Shield zeigt hier die neueste Schlagzeile an, die mein PC von „Spiegel Online“ geholt hat (mehr darüber in einer der nächsten Ausgaben). Ein Windows-RasPi wäre ein billiger und stromsparender „Skript-Sklave“, um für mich Webdienste zu verrichten.

Ich bin gerade dabei, meinen Interpreter auf Android zu portieren, so dass alles 1:1 auch auf meinem Tablet und Smartphone läuft (ich denke über eine App für mich und meine Kollegen nach, mit der wir auf Messen unsere Termine verwalten und alle neuen Ideen gleich in unsere Datenbank eintragen können). Android und Java laufen auch auf dem Raspberry Pi, aber beides hat wohl seine Tücken. Ich habe gelesen, dass Java für die kleineren Zielprozessoren kein Webview-Element mitbringt, und das brauche ich, um meine HTML-basierten (und daher hardware-unabhängigen) User-Interfaces darzustellen. Noch viel cooler wäre es allerdings, ich könnte HTML-Formulare auch auf einem Embedded-Display anzeigen (bitte mit Touch-Funktion!).

Es gibt ja zuhauf Grafikbibliotheken und Grafikbeschreibungssprachen für Mikrocontroller. Ich brauche einen ressourcen-sparenden „Pico-Browser“, der meine auf der SD-Karte abgelegten HTML-Oberflächen auf einem Grafik-Display rendert. Bisher bin ich noch nicht zu einer Recherche gekommen, ob jemand so etwas im Open-Source- oder Low-Cost-Bereich anbietet. Oder haben wir hier eine Marktlücke entdeckt?

Jens Nickel

Chefredakteur Elektor



Unser Team

Chefredakteur:	Jens Nickel (v.i.S.d.P.) (redaktion@elektor.de)
Ständige Mitarbeiter:	Dr. Thomas Scherer, Rolf Gerstendorf
Leserservice:	Ralf Schmiedel
Korrekturen:	Malte Fischer
Internationale Redaktion:	Harry Baggen, Jan Buiting, Jaime González-Arintero, Denis Meyer
Elektor-Labor:	Thijs Beckers, Ton Giesberts, Luc Lemmens, Jan Visser, Clemens Valens
Grafik & Layout:	Giel Dols

- 3 Impressum
- 6 Das Elektor-Netzwerk
- 30 Embedded World 2015
- 32 **Business Report**
Preiswerter LED-Hochleistungstreiber
- 88 **Elektor World News**
- 90 **Hexadoku**

LEARN

DESIGN

SHARE

- 9 **Learn**
Nicht vergessen!
- 10 **Von 8 auf 32 bit:**
ARM-Controller für Einsteiger (3)
Rund um die Timer
- 18 **DesignSpark Tipps & Tricks**
Multilayer-Platinen
- 20 **IoT ausgepackt**
Einfach WunderBar
- 23 **Tipps und Tricks**
Von Lesern für Leser
- 24 **Codebender**
Arduino im Internet
- 28 **Seltene Bauteile**
Integrierter Gyrator TCA580
- 29 **Einfacher Transistortester**

LEARN

DESIGN

SHARE

- 35 **Design**
- 36 **Willkommen im Elektor-Labor**
- 38 **UART/RS232-Datenlogger**
Serielle Datenströme überwachen
- 50 **Lüftungssystem für den Keller**
Platino regelt Luftfeuchte
- 56 **Medikamenten-Alarm**
Neun Erinnerungen täglich

UART/RS-232- Datenlogger

Serielle Datenströme überwachen

Der UART kommt (immer noch) gern zum Einsatz, wenn Geräte Daten austauschen sollen. Die Schnittstelle ist sehr einfach zu verwenden, dazu gibt es eine Vielzahl von Konverter-Chips. Wer bei Entwicklung und Test aber feststellt, dass die Kommunikation noch hakt, muss sich auf eine knifflige Fehlersuche einstellen. Unser UART-Datenlogger macht das Leben leichter – er registriert Bytes in beiden Richtungen millisekundengenau. Dank flexibler Anschlussmöglichkeiten und einer galvanischen Trennung lässt er sich leicht in serielle Verbindungen mit TTL- oder RS232-Pegeln einschleifen.

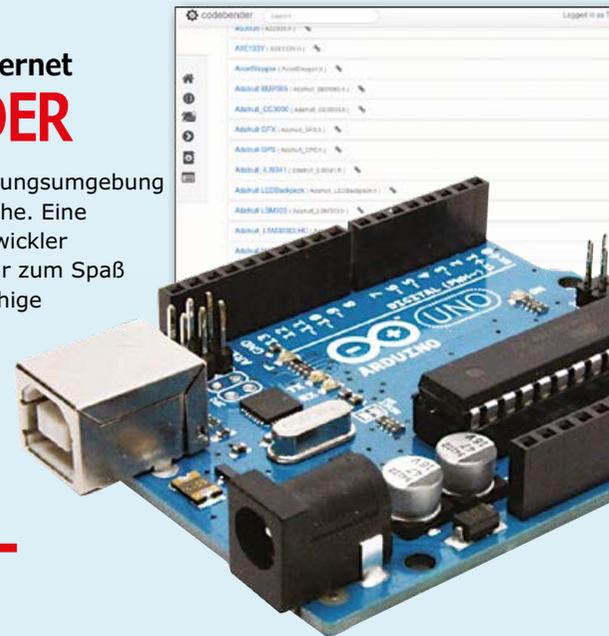


38

Arduino im Internet **CODEBENDER**

Massimo Banzis Entwicklungsumgebung ist nicht jedermanns Sache. Eine Gruppe griechischer Entwickler konstruierte 2012 – mehr zum Spaß – eine im Browser lauffähige Arduino-IDE, die sich im Laufe der Zeit zu einer ernstzunehmenden Alternative entwickelt hat.

24



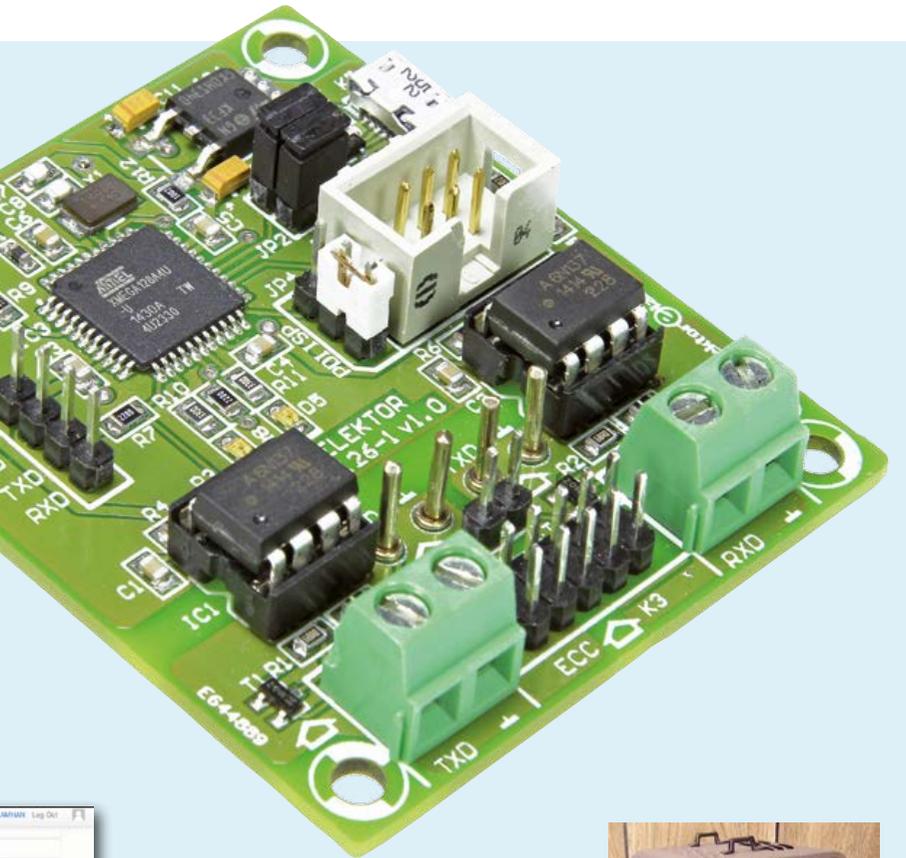
AM-Sender mit **Arduino** Induktiv auf Mittelwelle

Alte Röhrenradios haben einen ganz besonderen Reiz. Sie sehen nicht nur schön aus, sondern klingen auch noch gut und wecken oft angenehme Erinnerungen. Nur leider existieren immer weniger Rundfunkstationen, die noch auf den klassischen AM-Bereichen senden. Ein eigener AM-Sender muss her!



67

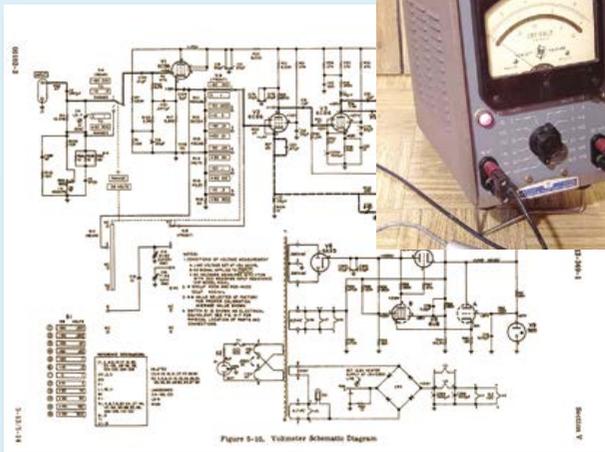
JETZT GANZ NEU!



- 62 e-BoB BL600 (2)**
Programm editieren, kompilieren und übertragen
- 67 AM-Sender mit Arduino**
Induktiv auf Mittelwelle
- 70 Serieller USB-nach-Multi-Protokoll-Konverter**
Für UART, SPI, I²C, ...!

LEARN	DESIGN	SHARE
-------	--------	-------

- 77 Share**
- 78 Retronik**
Restaurierung des Röhrenvoltmeters HP 400H (2)
- 82 Aus dem Labor**
Im Innern des Apfels
- 85 Elektor-labs.com**
Was ist heiß auf .Labs ...
- 86 Web-Recherche**
Messen mit dem Oszilloskop



Retronik

Restaurierung eines Röhrenvoltmeters (2)

78

Vorschau

- NFC-Gateway**
Ein ATmega328 macht's möglich: Das NFC-Tag-IC unseres Gateways kann man ganz einfach über ein UART-Interface beschreiben und mit einem NFC-fähigen Telefon auslesen. Und umgekehrt!
- Widerstandsmessung mit Arduino**
Eine neue Anwendung für unser Gespann aus Arduino, Extension-Shield und Bascom: Diesmal werden wir Widerstände genau messen und die Signale eines Feuchtesensors auswerten.
- Programmierbare musikalische Türklingel**
Diese Schaltung besteht aus zwei Teilen, die drahtlos kommunizieren: Dem Knopf und der Klingel. Über einen DIP-Switch lassen sich verschiedene Melodien auswählen.

Änderungen vorbehalten.
Elektor Mai 2015 erscheint am ??. April 2015.
 Verkaufsstellen findet man unter www.pressekaufen.de.

Das Elektor

Elektor durchbricht die Schranken einer Zeitschrift und wird zur Community aktiver E-Ingenieure - vom Anfänger bis zum Profi - begierig, überraschende Elektronik zu lernen, zu entwickeln, zu teilen.

56

Länder

246497

Mitglieder

10

Experten &



Elektor.POST

Die wöchentliche E-Inspiration

Niemals monostabil, bietet der wöchentliche Newsletter Elektor.POST das Triggersignal für ein Wochenende voller Tech-Klatsch und Tech-Talk. Und jede zweite Woche ein Extra-Projekt.

www.elektor.de/newsletter



Elektor – das Magazin

Fast 1024 Seiten überraschende Elektronik im Jahr

Wenn Sie lieber Elektronik beherrschen als sich von ihr beherrschen lassen, lesen Sie Elektors Flaggschiff, die vom internationalen Redaktionsteam zusammengestellte Zeitschrift. Ob online oder auf Papier, jede Ausgabe ist vollgepackt mit Elektronik!

www.elektor-magazine.de



Elektor TV

Auch auf der Mattscheibe

Unsere Filmkulissen sind noch nicht kinotauglich - aber Sie können sich auf einen Kameraschwenk verlassen, wenn irgendetwas zu summen, zu booten, zu blinken oder zu qualmen beginnt im Labor oder irgendeiner Veranstaltung, die unsere Moderatoren für video-kompatibel halten. Schauen Sie ab jetzt Elektor TV!

www.youtube.com/ElektorIM



Elektor-Platinenservice

Boards für Sie...

Vergessen Sie Chemikalien und bestellen Sie eine fertige Leiterplatte, um darauf Ihr Projekt zu verwirklichen. Schnelle Lieferzeiten, hervorragende Qualität, weltweiter Versand.

www.elektorpcbservice.com



Das Elektor-Labor

Lernen, entwickeln, teilen!

Das Technik-Kreativzentrum von Elektor (hier geht es von der Schaltungs-skizze bis zum Platinenlayout) steht Ihnen weit offen und ist über eine eigene Webseite erreichbar. Elektor-Labs.com ist dort, wo Projekte, groß, klein, analog, digital, new- oder old-school skizziert, gebaut, diskutiert, getestet und für den Nachbau und Ihre Anwendung optimiert werden.

www.elektor-labs.com



Elektor Academy

Reiten Sie auf der Lernkurve!

Webinare, Seminare, Kurse, Präsentationen, Workshops, Vorträge, innerbetriebliche Schulungen, DVDs und Demos sind nur einige der Methoden von Elektor, um das Wort der Elektronik im Hobby- und im Profi-Bereich zu verbreiten.

www.elektor.de/events

-Netzwerk



Autoren Literatur Monatliche Besucher Drucktermin



Elektor-Community

Werden Sie GREEN oder GOLD-Mitglied!

Die Mitgliedschaft in der Elektor-Community ist der sicherste Weg, klassische Elektronik sowie die Welt der Mikrocontroller zu genießen, ob Sie nun Anfänger oder Profi sind. Mit direktem Zugang zu Elektor-Labs.com, Foren, Rabatten, dem wöchentlichen Newsletter, Online-Projekten im 14-Tage-Rhythmus, Artikel-Archiven, Suchmaschinen und Hintergrundberichten. Mit der GREEN- und GOLD-Mitgliedschaft sitzen Sie stets in der ersten Reihe. Werden Sie GREEN-Mitglied, wenn Sie die Zeitschrift von vorne bis hinten nur online erhalten möchten oder GOLD-Mitglied, wenn Sie zusätzlich noch in der gedruckten Ausgabe der Zeitschrift blättern wollen.

www.elektor.de/mitgliedschaft



Elektor E-Shop

Füllen Sie Ihren Warenkorb!

Produkte und Dienstleistungen aus dem Elektor-Labor, der Zeitschrift und ausgewählten Geschäftspartnern von Elektor finden Sie hier. Der hell erleuchtete Online-Shop steht Ihnen durchgehend 24/7/365 offen und bietet Bestell- und Zahlungsmöglichkeiten für Kunden auf der ganzen Welt: Ali Babas Höhle der Elektronik!

www.elektor.de

Jetzt Mitglied werden!



Elektor-Bücher & DVDs

Powerpacks mit E-Informationen

Es ist schwer, einen Bereich der Elektronik zu finden, der nicht durch unsere Bücher und DVDs kompetent behandelt wird. Vom Nachschlagewerk zum Programmierkurs, von 8 bit zu ARM, von der Antenne zur Zenerdiode; es ist alles da.

www.elektor.de

GREEN	GOLD	GRATIS
92,50 €/Jahr	127,50 €/Jahr	
<ul style="list-style-type: none"> ✖ 10x Elektor jährlich (Print) ✔ 10x Elektor jährlich (Digital) ✔ Zugang zum Elektor-Archiv ✔ 10% Rabatt auf Shop-Produkte ✔ Zugang zu Elektor-Labs ✔ Elektor-Jahrgangs-DVD ✔ Exklusive Top-Angebote ✔ 26 neue Projekte jährlich 	<ul style="list-style-type: none"> ✔ 10x Elektor jährlich (Print) ✔ 10x Elektor jährlich (Digital) ✔ Zugang zum Elektor-Archiv ✔ 10% Rabatt auf Shop-Produkte ✔ Zugang zu Elektor-Labs ✔ Elektor-Jahrgangs-DVD ✔ Exklusive Top-Angebote ✔ 26 neue Projekte jährlich 	<ul style="list-style-type: none"> ✖ 10x Elektor jährlich (Print) ✖ 10x Elektor jährlich (Digital) ✖ Zugang zum Elektor-Archiv ✖ 10% Rabatt auf Shop-Produkte ✖ Zugang zu Elektor-Labs ✖ Elektor-Jahrgangs-DVD ✔ Exklusive Top-Angebote ✔ 26 neue Projekte jährlich
www.elektor.de/green-mitglied	www.elektor.de/gold-mitglied	www.elektor.de/newsletter

Frontplatten

- UV- und Digitaldruck
- Lasergravur
- Befestigungsbolzen

Kostenlose Panel-designersoftware
TARGET 300!

UV-Druck auf Acryl
und Aluminium

NEU!



www.front-panel.de

Beta
LAYOUT
create : electronics

Version 7.2

jetzt
erhältlich

**MIT BEWÄHRTEM
Lizenzmodell**


**CS
EAGLE**



www.cadsoft.de

**Designen Sie
mit Conrad!**

Über 5.000 Entwicklungskits und Prototyping-Zubehör aus der Halbleiter-Technologie!

Innovative, leistungsstarke Komponenten und
Schnellstart-Kits für Rapid Prototyping:

- ✓ TI Tiva, C2000, Hercules & MSP430 LaunchPad Kits
- ✓ STM32 Discovery Kits & Nucleo Boards
- ✓ Freescale Kinetis Freedom Platforms
- ✓ Microchip PIC-Kits & Tools
- ✓ Embedded Artists NXP LPCpresso Platforms
- ✓ MikroElektronika Compilers, Programmers, Debuggers & Kits
- ✓ 4D Systems TFT & OLED Anzeige-Modul
- ✓ UDOO Board & RioTboard mini-PCs
- ✓ FlowPaw Programmers Learning Kit
- ✓ WunderBar IoT Sensors & Cloud Starter-Kit

Technik + Service = Conrad
www.conrad.de

CONRAD
Business Supplies

Willkommen bei **LEARN**

SHARE

DESIGN

LEARN



Von **Jens Nickel**

Nicht vergessen!

Als Verantwortlicher für den Artikel „UART/RS232-Datenlogger“ in diesem Heft wollte ich den Logger natürlich auch gern einmal selbst ausprobieren; daher lieh ich mir den Prototypen meines Kollegen Ton Giesberts aus unserem Labor. Ernüchterung folgte, als ich das Board zum ersten Mal an meinen Rechner anschloss. Denn Windows 7 meldete „...not recognized...“. Das war merkwürdig, denn eigentlich sollte mein Betriebssystem jetzt herausfinden können, ob der nötige Treiber schon auf meinem Rechner installiert war oder nicht. Ich googelte nach „USB-CDC“ (dem USB-Modus für die Verbindung des Mikrocont-

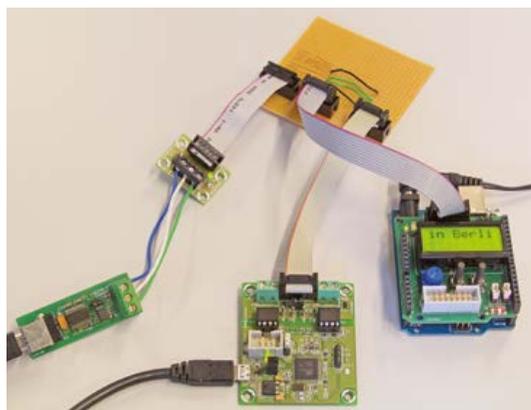
rollers mit dem PC) sowie möglichen bekannten Problemen, die den Xmega und/oder Windows 7 betrafen. Leider konnte ich nichts finden, das Licht ins Dunkel brachte.

Am nächsten Tag brachte ich das Board zu Ton ins Labor. „Nein Ton, diesmal habe ich nicht rumgespielt...“ (Ich hatte vor einiger Zeit an das „Temp-Sensor-Board“ die Betriebsspannung an die falsche Stiftleiste angeschlossen. Ich könnte schwören, dass ich das Ding so verkabelt auf den Tisch bekommen hatte, aber lassen wir das jetzt...).

Wir schlossen das Board an Tons Computer an und siehe da: „...not recognized...“. Ton konnte sich das überhaupt nicht erklären. Doch dann kam er auf die Idee, ein anderes USB-Kabel zu benutzen. Und hurra ... es lief!

Tja, auch das gehört zum Lernen dazu: Man darf nie vergessen, auch einmal an wirklich blöde Fehler zu denken. Keine instabile Firmware, kein Problem bei Atmel, kein unglückliches Zusammentreffen widriger Umstände bei meinem Betriebssystem - es war schlicht mein USB-Kabel futsch.

Als Vorkämpfer bei uns, was modulare Systeme angeht, wollte ich den nun funktionierenden Datenlogger gleich mit einer ECC-Verbindung ausprobieren und hatte auch versprochen, hierzu einen kleinen Kasten für den Artikel beizusteuern.



Auf meinem Tisch liegt ja immer ein Arduino Uno mit Extension-Shield, an den (über den ECC) ein RS485-Konverter angeschlossen ist; per RS485 und einen RS485/USB-Konverter geht es dann weiter zum PC. Die Kollegen hatten mir inzwischen eine kleine Y-Verteilerplatine für ECC-Signale gelötet und so konnte ich den Datenlogger leicht in die Verbindung zwischen Extension-Shield und RS485-Konverter einschleifen (siehe Foto). Für einen ersten Test spielte ich meine Steuerungs-Software auf, mit der man

über einfache ASCII-Befehle von einem Terminalprogramm aus („L 0 0 +<CR>“) zum Beispiel LEDs ein- und ausschalten konnte. Ich öffnete gespannt eine zweite Instanz von HTerm und konfigurierte den Datenlogger auf die richtige Baudrate. Danach schickte ich mit dem ersten Terminalprogramm Zeichen an den Arduino. Und sah, wie der Datenlogger im zweiten Terminalfenster ausgab:

```
OK_{cr}
```

Also die Antwort des Arduino auf den Befehl, den Pin zu schalten. Den ASCII-Befehl selbst bekam ich aber nicht angezeigt. Warum funktionierte das Mitloggen nur in eine Richtung? Hatte der Kollege beim Löten des Y etwas falsch gemacht? Testweise steckte ich den ECC-Stecker verdreht auf den Datenlogger. Nun bekam ich den Befehl des PCs angezeigt („L_0_0_{cr}“), aber nicht mehr die Antwort des Arduino. Also stimmt etwas mit dem Datenlogger selbst nicht. Etwa eine kalte Lötstelle? Ich sah mir das Board genau an und entdeckte, dass der Jumper fehlte, mit dem die Massen der beiden Signale zusammengeschaltet werden konnten. Wie der Schaltplan des Datenloggers klar zeigt, wird die ECC-Masse ja nur an einen Kanal weitergegeben.

Hatte ich nicht einmal (in einem ElektorBus-Artikel) einen Textkasten „Grounding for Dummies“ geschrieben?

Vergiss das Grounden nicht, du Dummie!

Rund um die Timer



Von 8 auf 32 bit: ARM-Controller für Einsteiger (3)

Von Viacheslav Gromov (D)

Das richtige Timing ist auch in der digitalen Welt sehr wichtig. Deswegen erkunden wir in dieser Kurs-Folge die Timer des SAM D20, vom Watchdog Timer bis hin zum Real Time Counter. Nebenbei werden weitere Themen wie Takterzeugung und External Interrupts behandelt, wie immer praxisnah anhand von kleinen Codebeispielen. So bleiben Sie garantiert Ihrer Zeit voraus ;-)

In der letzten Folge ging es unter anderem um die Kommunikationseinheit SERCOM, wobei wir noch einige Schnittstellen nicht ausprobiert haben. Bevor wir damit weiter machen, müssen wir uns noch die verschiedenen Timer des Mikrocontrollers anschauen. Diese werden wir noch in den nächsten Folgen gut gebrauchen können.

Die Listings der nachfolgenden Projekte finden Sie in der Textdatei „Listings“ im Download [1] zu diesem Kursteil. Natürlich sind auch die vollständigen Atmel-Studio-Projekte in dem Zip-Archiv untergebracht. Die für diesen Kursteil benötigte Hardware ist die gleiche wie beim letzten Mal, allerdings sind die Taster und LEDs etwas anders mit dem Board verbunden. Schließen Sie also erst einmal die in **Bild 1** gezeigte Schaltung am SAM D20 Xplained Pro Board an, dann kann es losgehen!

Der WDT, die Taktversorgung und der EIC

Der **Watchdog-Timer (WDT)** ist einer der einfachsten Timer des SAM D20. Wir können deshalb mit ihm leicht in die Materie einsteigen, obwohl er, wie **Bild 2** zeigt, mehr Funktionen aufweist als der WDT eines herkömmlichen 8-bit-Mikrocontrollers. Ein Watchdog-Timer hat die Aufgabe, ein Programm neu zu starten, falls es nicht im gewünschten Zeitrahmen ausgeführt wird (also festhängt). Wie Sie im Blockdiagramm erkennen können, gibt es ein Zählregister COUNT, das hier bis 32.768 zählen kann, und darunter weitere Register, mit denen der aktuelle Wert des Zählers verglichen wird. Bei einer Übereinstimmung wird ein Early Warning Interrupt oder sogar ein Reset des Mikrocontrollers ausgelöst. Das Programm sollte also immer zur rechten Zeit den WDT zurücksetzen, damit der Zähler nicht überläuft und kein Neustart (Reset) erzwungen wird. Beim SAM D20 kennt der WDT zwei Modi:

- Beim normalen Modus stellt man einen Überlauf-Wert von 8...16.384 ein, bei dem ein Reset ausgeführt und der Zähler zurückgesetzt wird. Im gleichen Bereich lässt sich auch der Early Warning Interrupt einstellen, der kurz vor dem Überlauf einen Warn-Interrupt auslöst.
- Im Window-Modus kann das Programm den Zähler nur in einem bestimmten Zeitfenster zurücksetzen. Wenn das Programm den WDT-Zähler außerhalb des Zeitfensters zurücksetzt, wird der Mikrocontroller neu gestartet. Für das „offene“ Zeitfenster kann man einen Zählerstand von 8...16384 wählen. Von diesem Zählerstand an beginnt dann das geschlossene Fenster, das ebenfalls 8 bis 16384 Zählertakte groß sein kann.

Getaktet wird der WDT von einem der acht Generatoren des Generic Clock Controllers (GCLK) über einen Multiplexer. Die Struktur dieses Controllers ist in **Bild 3** dargestellt. Er besteht aus acht Generatoren, die wiederum von vielen externen und internen Oszillatoren oder sogar anderen Generatoren getaktet werden können. Jeder der Generatoren besitzt unter anderem einen Vorteiler (Prescaler) zur Teilung des Oszillatortakts. Der EIC ist ein **External Interrupt Controller (Bild 4)**, der bei unserem SAM D20 16 Pins unterstützt. Der Controller filtert digital das Signal an den gewählten Pins und wertet es aus. Er kann damit, je nach Konfiguration, bei einer fallenden oder steigenden Flanke am jeweiligen Pin Interrupts auslösen und zum Beispiel die MCU aus dem Schlaf aufwecken. Neben weiteren Optionen ist auch die Detektion eines Pulses möglich. Der EIC unterstützt noch einen zusätzlichen Pin, der nicht-maskierte Interrupts auslösen kann, die dann mit Hilfe des **Nested Vectored Interrupt Controllers (NVIC)** sogar andere Interrupts (je nach Priorität) unterbrechen können.

Erste Schritte mit dem WDT

Jetzt wollen wir den WDT Timer im normalen Modus ausprobieren. Da er relativ einfach aufgebaut ist, eignet er sich perfekt für das erste Timer-Projekt. Keine Bange, dank der gut gemachten Bibliotheken des schon bekannten ASFs fällt die Ansteuerung eines Timers grundsätzlich leicht.

Das Programm „First Project with WDT“ aus dem Zip-Archiv ist sehr einfach zu verstehen. Ganz oben, nach der Deklaration der Variablen und der Funktionsprototypen, finden sich drei Callback-Funktionen. Die erste namens `watchdog_early_warning_callback` wird beim Early Warning Interrupt des WDT

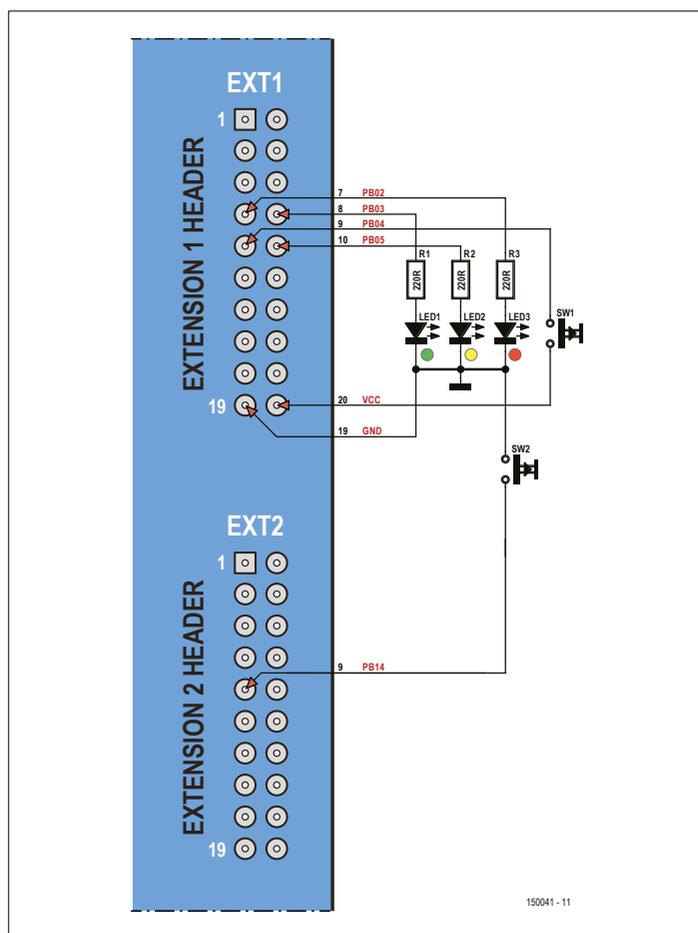


Bild 1. Die gegenüber dem letzten Kursteil leicht geänderte Schaltung.

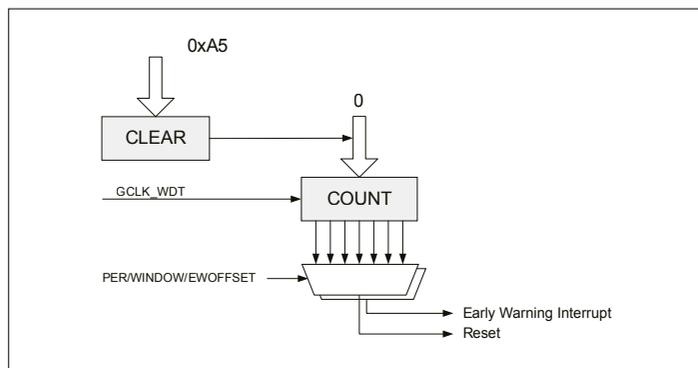


Bild 2. Ein Blockschaltbild des Watchdog-Timers (alle Screenshots und Blockdiagramme: Atmel).

aufgerufen und toggelt lediglich mit einem Port-Befehl die LED0 auf dem Board.

Die nächste Funktion lautet `extint4_detection_callback` und wird durch Taster SW1 am Interrupt-Pin 4 aufgerufen. Darin wird der Bool-Variablen `i` der Wert 1 zugewiesen. Die letzte Callback-Funktion `extint14_detection_callback` wird durch Taster SW2 (Interrupt-Pin 14) aufgerufen. Sie weist `i` den Wert 0 zu. Danach folgen die beiden wichtigen Konfigurations-Funktionen des WDTs und des EICs, die auch im **Listing 1** zu finden sind.

Die Konfiguration des WDTs läuft nach dem bekannten Muster ab: Zuerst wird eine Konfigurations-Struktur, in diesem Fall `config_wdt` erzeugt und mit `wdt_get_config_defaults(&config_wdt)` mit Voreinstellungen initialisiert. Es folgen die spezifischen Einstellungen der Variablen der Konfigurations-Struktur. Zuerst wird die Variable `always_on` mit `false` gefüllt (bei `true` wäre der WDT erst beim nächsten Einschalten wieder bereit, konfiguriert zu werden, zuvor wäre er immer aktiv und könnte nur zurückgesetzt werden). Dann wird mit der Struktur-Variablen `clock_source` die Taktquelle ausgewählt (GCLK-Generator 4) und der Überlauf- beziehungsweise der Early-Warning-Wert eingestellt, hier auf 8.192 respektive 4.096 Taktzyklen.

Da der Early-Warning-Wert genau die Hälfte des Überlauf-Werts beträgt, blinkt die LED später gleichmäßig. Am Ende dieser Funktion werden die Werte in der Konfigurations-Struktur mit dem Befehl `wdt_set_config(&config_wdt)` an den Watchdog Timer übergeben.

Nach ähnlichem Muster konfiguriert die nächste Funktion den EIC. Zuerst wird auch hier eine Konfigurations-Struktur namens `config_extint_chan` erstellt und mit Voreinstellungen beladen. Es folgen zwei sehr ähnliche Code-Blöcke. Im ersten werden mit den Variablen der Konfigurations-Struktur die Pin-, Multiplexer-, Pull-Widerstand- und Erkennungs-Kriterien für den Pin PB04 eingestellt, an den der Taster SW1 angeschlossen ist. Da dieser Taster high-aktiv ist, sorgt ein Pull-down-Widerstand für den Low-Pegel im Ruhezustand. Das Erkennungs-Kriterium muss eine steigende Flanke (von low auf high) sein. Am Ende dieses Blocks müssen wieder die vorgenommenen Einstellungen mit `extint_chan_set_config(4, &config_extint_chan)` an den External Interrupt Controller übergeben werden. Dieser Funktion müssen nur die Nummer des externen Interrupts (hier 4) und ein Pointer auf die Konfigurations-Struktur übergeben werden. Der zweite Code-Block zur Konfiguration des EICs ist genauso aufgebaut, nur betrifft er den Pin PB14 (externer Interrupt 14) mit dem angeschlossenen Taster SW2. Und da dieser Pin genau umgekehrt arbeitet, also low-aktiv ist, muss ein Pull-up-Widerstand konfiguriert werden und der EIC nicht auf die steigende, sondern auf die fallende Flanke reagieren.

Es folgen nur noch die Funktionen, die im Programm einmalig aufgerufen werden und die Interrupt-Funktionen (Callbacks) des WDT beziehungsweise des EICs registrieren und aktivieren. Die erste Funktion heißt `configure_wdt_callbacks` und enthält nur zwei Zeilen Code:

```
wdt_register_callback(watchdog_early_warning_callback, WDT_CALLBACK_EARLY_WARNING);
wdt_enable_callback(WDT_CALLBACK_EARLY_WARNING);
```

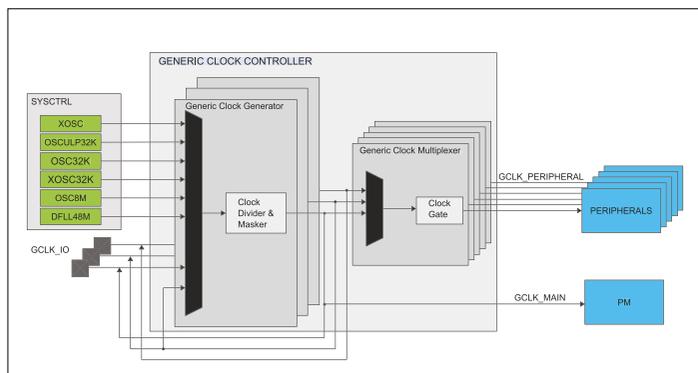


Bild 3. Blockschaltbild des Generic Clock Controllers.

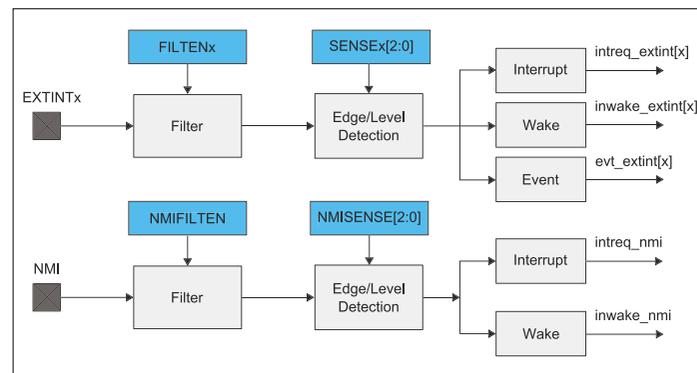


Bild 4. Das Blockschaltbild zeigt einen External Interrupt Controller.

Der erste Befehl, der die Callback-Funktion `watchdog_early_warning_callback` registriert, benötigt als zweiten Parameter nur das Ereignis, bei der die Funktion aufgerufen werden soll. Der zweite Befehl, der den zuvor registrierten Callback aktiviert, enthält nur die Art des Ereignisses.

Die Funktion für die Registrierung und das Aktivieren der EIC-Callbacks sieht ähnlich aus, bearbeitet aber gleich zwei Interrupts (EXTINT4 und EXTINT14). Die darin enthaltenen

Befehle ähneln denen der letzten Funktion, sie benötigen immer die Nummer des gewünschten externen Interrupts. In der Main-Datei werden nur noch die Konfigurations-Funktionen aufgerufen und die LED0 auf den low-Pegel gesetzt. In der Endlosschleife wird ständig abgefragt, ob `i = 1` ist. Wenn ja, wird der Watchdog-Timer mit `wdt_reset_count()` zurückgesetzt. Werfen Sie noch vor dem Kompilieren dieses Programms einen Blick auf die Datei `conf_clocks.h` (unter `src/config`). Die aufge-

Listing 1. Die Konfigurations-Funktionen des WDT und des EIC.

```
void configure_wdt(void)
{
    struct wdt_conf config_wdt;
    wdt_get_config_defaults(&config_wdt);
    config_wdt.always_on = false;
    config_wdt.clock_source      = GCLK_GENERATOR_4;
    config_wdt.timeout_period    = WDT_PERIOD_8192CLK;
    config_wdt.early_warning_period = WDT_PERIOD_4096CLK;
    wdt_set_config(&config_wdt);
}

void configure_extint_channels(void)
{
    struct extint_chan_conf config_extint_chan;
    extint_chan_get_config_defaults(&config_extint_chan);

    config_extint_chan.gpio_pin = PIN_PB04A_EIC_EXTINT4;
    config_extint_chan.gpio_pin_mux = MUX_PA04A_EIC_EXTINT4;
    config_extint_chan.gpio_pin_pull = EXTINT_PULL_DOWN;
    config_extint_chan.detection_criteria = EXTINT_DETECT_RISING;
    extint_chan_set_config(4, &config_extint_chan);
    config_extint_chan.gpio_pin = PIN_PB14A_EIC_EXTINT14;
    config_extint_chan.gpio_pin_mux = MUX_PA14A_EIC_EXTINT14;
    config_extint_chan.gpio_pin_pull = EXTINT_PULL_UP;
    config_extint_chan.detection_criteria = EXTINT_DETECT_FALLING;
    extint_chan_set_config(14, &config_extint_chan);
}
```

Listing 2. Die Einstellungen des GLCK Generators Nummer 4.

```
# define CONF_CLOCK_GCLK_4_ENABLE      true
# define CONF_CLOCK_GCLK_4_RUN_IN_STANDBY false
# define CONF_CLOCK_GCLK_4_CLOCK_SOURCE SYSTEM_CLOCK_SOURCE_OSC8M
# define CONF_CLOCK_GCLK_4_PRESCALER   1024
# define CONF_CLOCK_GCLK_4_OUTPUT_ENABLE false
```

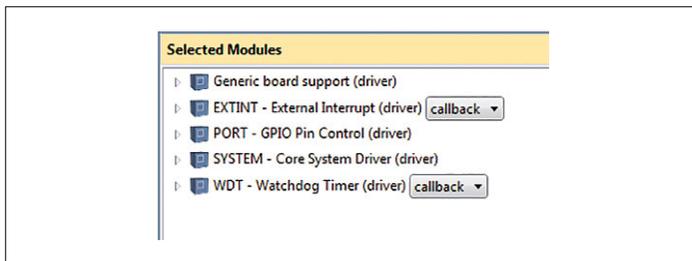


Bild 5. Alle notwendigen Bibliotheken für das Watchdog-Projekt.

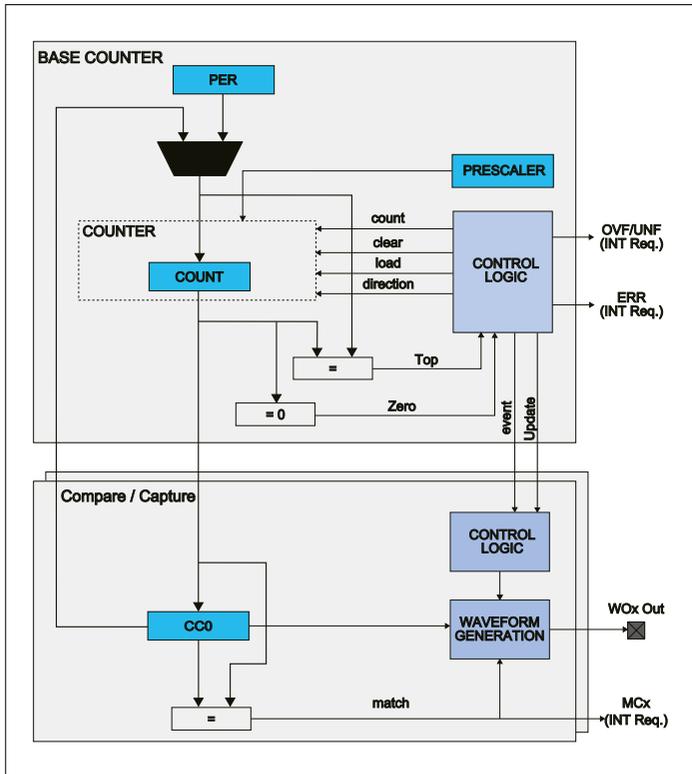


Bild 6. So ist ein Timer/Counter aufgebaut.

fürten Takt-Einstellungen können Sie leicht ändern. Für dieses Projekt wurden die Einstellungen des GCLK-Generators 4 geändert, der den WDT taktet (**Listing 2**). Der Prescaler wurde auf 1024 eingestellt, damit der WDT-Zähler möglichst langsam hochgezählt. `CONF_CLOCK_GCLK_4_ENABLE` wurde auf `true` gesetzt, um den Generator 4 zu aktivieren. Nebenbei sehen Sie im Listing auch, welchen Oszillator dieser Generator nutzt, nämlich den 8-MHz-Hauptoszillator, der auch die CPU taktet. Stellen Sie sicher, dass alle benötigten Bibliotheken mit dem ASF-Wizard in das Projekt eingebunden wurden (**Bild 5**). Jetzt können Sie endlich dieses Programm mit dem Studio kompilieren und auf das Board übertragen (Start without Debugging). Sie sehen, dass die LED0 auf dem Board blinkt, da der Early

Warning Interrupt sie zunächst aktiviert und der Reset beim Überlauf des WDT sie dann wieder auf low legt (LED dunkel). Wenn Sie aber Taster SW1 betätigen, wird ein externer Interrupt ausgelöst, der `i` auf 1 setzt. Der WDT wird jetzt permanent in der Endlosschleife zurückgesetzt, sodass er keine Interrupts mehr auslösen kann und die LED ihren letzten Zustand beibehält. Ein Druck auf SW2 bewirkt, dass über einen externen Interrupt die Variable `i` wieder auf 0 gesetzt wird. Der WDT wird nicht mehr zurückgesetzt und die LED0 beginnt wieder zu blinken [2][3].

Der Timer/Counter

Unser SAM D20 verfügt über acht Standard-Timer/Counter (kurz: TC), die sehr vielseitig nutzbar sind. Wie schon in der ersten Folge beschrieben, kann man einen Timer als 8-bit-Timer, als 16-bit-Timer oder zwei zusammen als 32-bit-Timer konfigurieren. Wie Sie es bestimmt schon von 8-bit-Mikrocontrollern her kennen, kann ein solcher Allzweck-Timer Taktzyklen zählen (und so ein Zeitintervall bestimmen) und bei einem Überlauf oder durch den Vergleich mit einem Compare-Register einen Interrupt auslösen. An entsprechenden Pins kann auch ein PWM-Signal erzeugt werden. Die TCs des SAM D20 können das natürlich ebenfalls und noch vieles mehr. Der Aufbau eines TCs des Cortex-M0+-Mikrocontrollers ist in **Bild 6** grob dargestellt. Der Zähler-Block oben besitzt mehrere Vergleichseinheiten unten. Bei einer Übereinstimmung des Zählwertes mit dem Wert des Compare-Registers wird der Pegel an den zum Timer gehörenden Ausgängen beziehungsweise Kanälen (z.B. WO0 und WO1) verändert oder ein Interrupt ausgelöst. Natürlich kann der Timer sowohl auf- als auch abwärts zählen und sich selbst beim Überlauf zurücksetzen. Der Timer ist nicht zuletzt wegen dieser Vergleichs-Blöcke sehr flexibel. Er kann auf unterschiedliche Arten PWM-Signale erzeugen, Zeit beziehungsweise Taktzyklen messen, Ereignisse zählen, beliebige Rechteck-Signale erzeugen und vieles mehr. Die Timer arbeiten, wenn gewünscht, auch im Schlafmodus des Mikrocontrollers oder während des Debuggens. Der Timer verfügt, wie man im Blockdiagramm erkennt, über einen eigenen Vorteiler (Prescaler) und kann seinen Takt von einem beliebigen GCLK-Generator erhalten.

Das erste Timer-Projekt

Nun wollen wir den Timer im Zähl-Modus mit der obigen Schaltung ausprobieren. Das Projekt schaltet die LEDs um, wenn der Zähler den Wert in den Compare-Registern erreicht oder überläuft. Öffnen Sie bitte das Projekt „The first project with the counter“ im Zip-Archiv. Wenn Sie einen Blick in die Main-Datei werfen, sehen Sie, dass im Programm als erstes die symbolischen Konstanten für die Pins definiert werden, an denen die drei LEDs der Steckbrett-Schaltung angeschlossen sind. Danach folgen die Funktionsprototypen, bevor eine Struktur des Typs `tc_module` namens `tc_instance` erzeugt wird, unter der später der TC aufgerufen werden kann.

Weblinks

- [1] www.elektor-magazine.de/150041
- [2] www.atmel.com/Images/Atmel-42124-SAM-D20-Watchdog-Driver-WDT_Application-Note_AT03264.pdf
- [3] www.atmel.com/Images/Atmel-42112-SAM-D20-External-Interrupt-Driver-EXTINT_Application-Note_AT03246.pdf
- [4] www.atmel.com/Images/Atmel-42123-SAM-D20-D21-Timer-Counter-Driver-TC_Application-Note_AT03263.pdf

Listing 3. Die Konfigurations-Funktion des TCs.

```

void configure_tc(void)
{
    struct tc_config config_tc;
    tc_get_config_defaults(&config_tc);
    config_tc.counter_size = TC_COUNTER_SIZE_8BIT;
    config_tc.clock_source = GCLK_GENERATOR_1;
    config_tc.clock_prescaler = TC_CLOCK_PRESCALER_DIV1024;
    config_tc.counter_8_bit.period = 99;
    config_tc.counter_8_bit.compare_capture_channel[0] = 33;
    config_tc.counter_8_bit.compare_capture_channel[1] = 66;
    tc_init(&tc_instance, TC0, &config_tc);
    tc_enable(&tc_instance);
}

```

Dann kommt auch schon die erste, bereits bekannte Konfigurations-Funktion, in der die drei Pins der LEDs als Ausgänge konfiguriert werden. Es folgen die drei Callback-Funktionen des Timers: `tc_callback_on_channel0`, `tc_callback_on_channel1` und `tc_callback_on_overflow`. Wie ihre Namen andeuten, wird die erste Funktion bei einer Übereinstimmung des TC-Zählers mit dem Compare-Register 0 aufgerufen, die zweite bei einer Übereinstimmung mit dem Compare-Register 1 und die dritte bei einem Überlauf des TC-Zählers. Jede der genannten Funktionen toggelt mit `port_pin_toggle_output_level(LEDx)` jeweils eine der drei LEDs, wobei x für G(reen), Y(ellow) oder R(ed) steht.

Auf die Callback-Funktionen folgt die Konfigurations-Funktion des TCs, wie sie in **Listing 3** zu sehen ist. Nach dem Erzeugen und dem Initialisieren mit Voreinstellungen wird innerhalb der Konfigurations-Struktur zuerst der Modus (8 bit), die Taktquelle (GLCK-Generator 1) und der Prescaler-Wert (1024) eingestellt. Schließlich werden nur noch die Werte der beiden

Compare-Register und des Überlaufs eingestellt. Dabei achten wir auf gleiche Abstände (33, 66, 99), damit das Blinkmuster der LEDs auch gleichmäßig ist. Die vorgenommenen Einstellungen werden dem Timer/Counter0 mit `tc_init(&tc_instance, TC0, &config_tc)` übergeben; schließlich wird der TC mit `tc_enable(&tc_instance)` aktiviert.

Die Funktion `configure_tc_callbacks` registriert lediglich noch die Callback-Funktionen für verschiedene Ereignisse und aktiviert diese gleich danach. In der Main-Datei werden nur noch die Konfigurations-Funktionen einmalig aufgerufen. Die Endlosschleife bleibt leer. In der Datei `conf_clocks.h` müssen wir den GLCK-Generator 1 über `CONF_CLOCK_GCLK_1_ENABLE = true` aktivieren und den externen 32-kHz-Oszillator, von dem dieser Generator den Takt erhält, ebenfalls mit einem `true` bei `CONF_CLOCK_XOSC32K_ENABLE` in Betrieb nehmen. Natürlich mussten auch hier die nötigen Bibliotheken mit dem ASF-Wizard eingebunden werden (**Bild 7**). Nun können Sie das Projekt kompilieren und beobachten, wie die LEDs nacheinander ein- und dann wieder ausgeschaltet werden.

Listing 4. Die Konfigurations-Funktion des TCs für eine PWM-Erzeugung.

```

void configure_tc(void)
{
    struct tc_config config_tc;

    tc_get_config_defaults(&config_tc);

    config_tc.counter_size = TC_COUNTER_SIZE_16BIT;
    config_tc.wave_generation = TC_WAVE_GENERATION_NORMAL_PWM;

    config_tc.counter_16_bit.compare_capture_channel[0] = 0xFFFF;
    config_tc.pwm_channel[0].enabled = true;
    config_tc.pwm_channel[0].pin_out = EXT1_PWM_0_PIN;
    config_tc.pwm_channel[0].pin_mux = EXT1_PWM_0_MUX;
    config_tc.counter_16_bit.compare_capture_channel[1] = 0xFFFF;
    config_tc.pwm_channel[1].enabled = true;
    config_tc.pwm_channel[1].pin_out = EXT1_PWM_1_PIN;
    config_tc.pwm_channel[1].pin_mux = EXT1_PWM_1_MUX;

    tc_init(&tc_instance, EXT1_PWM_MODULE, &config_tc);

    tc_enable(&tc_instance);
}

```

PWM-Signale erzeugen

Wir wollen nun kurz auf die PWM-Erzeugung eingehen. Da eine Callback-Variante bei diesem Projekt wenig Sinn hat, wurde die Polled-Variante der TC-Bibliothek ausgewählt. Die Delay-Bibliothek wird hier auch gebraucht (**Bild 8**). Das Programm soll die grüne und die rote LED an den Ausgängen (WO0/WO1) des TC6 gleichzeitig dimmen. Die Konfigurations-Funktion des TCS sieht dabei so aus wie in **Listing 4**. Nach der Erzeugung und der Initialisierung der Konfigurations-Struktur wird hier zuerst die Größe des Zählers (16 bit) und der Modus des TC (normales PWM) ausgewählt. Danach werden die beiden Kanäle eingestellt. Beide besitzen einen Überlauf-Wert von 0xFFFF (65.535) und werden jeweils mit dem dazugehörigen Output-Pin verbunden.

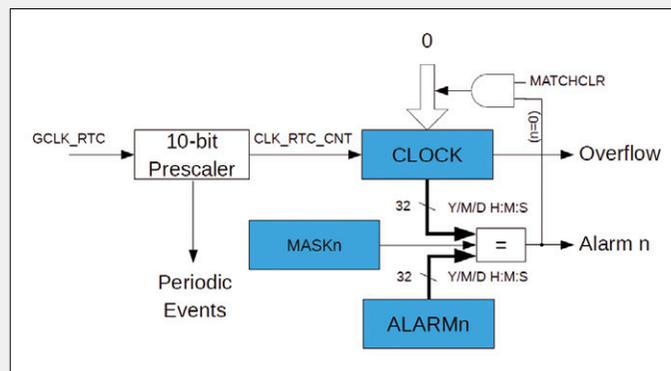
Dann wird nur noch TC6 (der hinter EXT1_PWM_MODULE steckt) mit den Einstellungen durch `tc_init(&tc_instance, EXT1_PWM_MODULE, &config_tc)` beladen und mit `tc_enable(&tc_instance)` aktiviert. Unter `tc_instance` kann TC6 dann später aufgerufen werden.

In der Endlosschleife der Main-Funktion werden die beiden Compare-Register der PWM-Kanäle mit dem Befehl `tc_set_compare_value(&tc_instance, TC_COMPARE_CAPTURE_CHANNEL_0, Wert)` in einer For-Schleife fortwährend geändert. Ein Register wird erhöht, das andere verringert. Je höher der Compare-Wert, desto höher ist auch das Tastverhältnis und damit die Helligkeit der LED. 65.535 ist der höchstmögliche und 0 der niedrigste Wert dieser Compare-Register. Die zweite For-Schleife in der

Der Real-Time Counter (RTC)

Der Hauptcontroller des SAM D20 Xplained Pro Boards hat wie die meisten 32-bit-Mikrocontroller einen Real Time Counter an Bord, im Gegensatz zu 8-bit-MCUs.

Dieser RTC ist ein 32-bit-Zähler mit einem 10-bit-Vorteiler, der von verschiedensten Quellen mit einem Takt versorgt werden kann. Er kann entweder im Counter-Modus über sehr lange Zeitperioden zählen oder, wie bei einer üblichen RTC, im Calendar-Modus das Datum und die Zeit erzeugen. Im Counter-Modus ist wie bei einem TC auch ein Vergleich des aktuellen Zählwerts mit einem Compare-Wert möglich. Im Calendar-Modus lässt sich an einem bestimmten Datum zu einer bestimmten Zeit im 24h- oder 12h-Format ein Interrupt auszulösen (RTC-Alarm). Soll der RTC im Calendar-Modus betrieben werden, muss er im Hertz-Rhythmus getaktet werden. Zwei einfache Programme im Zip-Archiv, „The first project with RTC Counter“ und „The first project with the RTC calendar mode“, demonstrieren den Umgang mit den beiden Modi des RTCs. Beide funktionieren mit der obigen Schaltung. Das erste Programm ist ein mit dem RTC realisierter Wechselblinker: Die grüne und rote LED leuchten alternierend zur gelben. Zuerst werden die Pins, welche die LEDs auf dem Steckbrett ansteuern, als Ausgänge konfiguriert und PB05 gleich auf high gesetzt. Dann müssen nur noch die drei Pins gleichzeitig getoggelt werden. Das passiert auch in der Callback-Funktion des RTCs, die periodisch bei einem Überlauf des Zählers aufgerufen wird. Die Konfigurations-Funktion hat das bekannte Muster (**Listing 5** oben). In der Mitte stehen wichtige Befehle. Der eine stellt den Prescaler auf 1 ein und der andere macht aus dem 32-bit-RTC einen 16-bit-Zähler. Nach dieser Funktion folgt die Callback-Registrierung und -Aktivierung sowie das Aufrufen der Konfigurations-Funktionen in der Main-Funktion. Der letzte Befehl der Main-Funktion vor der leeren Endlosschleife lautet `rtc_count_set_period(&rtc_instance, 1000)`. Die 1000 darin ist der Überlauf-Wert des RTCs. Der Wechselblinker wechselt seinen Zustand im Sekundentakt, da der RTC vom GCLK-Generator 2



einen vorgeteilten Takt von einem Kilohertz erhält und genau nach 1000 Taktperioden überläuft.

Das zweite Projekt zeigt den RTC im Calendar-Modus. Hier wird nur die grüne LED auf dem Steckbrett benutzt. Ein RTC-Alarm bewirkt, dass sie sekundlich ihren Zustand ändert, also blinkt. Dazu muss im Callback des RTC-Alarms die LED umgeschaltet und der nächste Alarm auf ein Intervall von einer Sekunde gesetzt werden. Die Konfigurations-Funktion des RTCs sieht aus wie in **Listing 5** unten.

In der Main-Funktion wird der Alarm0, der den benötigten Callback auslöst, auf den 01.04.2015 00:00:01 gesetzt. Viel weiter unten, nach der Funktion zur Registrierung und Aktivierung des Alarm0-Callbacks und dem Aufruf der zahlreichen Konfigurations-Funktionen, wird die RTC-Start-Zeit auf den 31.03.2015 23:59:59 gesetzt. Der erste Alarm wird also zwei Sekunden nach dem Start des Programms ausgelöst; dabei wechselt der Zustand der grünen LED. Danach wird der Alarm jede Sekunde ausgelöst, so dass die LED jede Sekunde ihren Zustand wechselt. Der GCLK-Generator 2 muss bei beiden Projekten manuell in der `conf_clocks.h`-Datei aktiviert werden. Die RTC-Bibliotheken benötigt man in der Callback-Variante; sie werden wie immer mit dem ASF-Wizard den Projekten hinzugefügt.

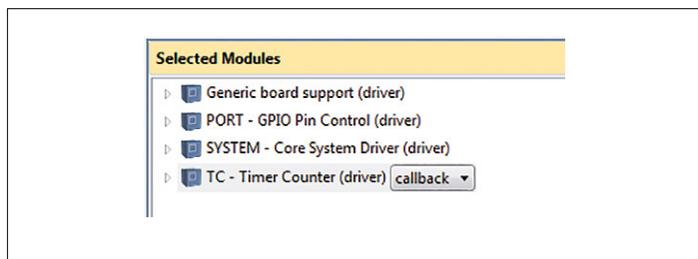


Bild 7. Alle Bibliotheken für das Timer-Projekt im ASF-Wizard.

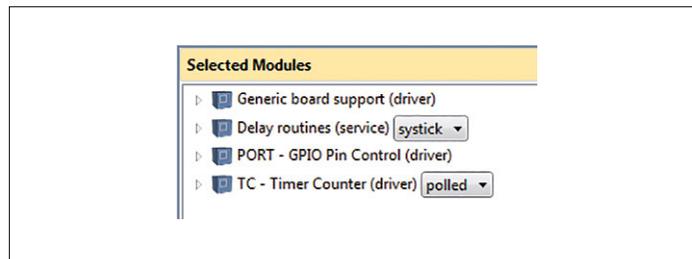


Bild 8. Die notwendigen Bibliotheken für das PWM-Projekt.

Endlosschleife bewirkt genau das Gegenteil – es wird langsam die grüne LED abgedimmt und die rote LED heller gemacht. Da nichts anderes eingestellt ist, erhält TC6 seinen Takt von Haupt-Generator 0, der nicht extra aktiviert werden muss (er versorgt ja schließlich auch die CPU).

Kompilieren Sie also das Programm und beobachten Sie die beiden LEDs [4].

Nun ist dieser ziemlich anspruchsvolle Teil des Kurses schon zu Ende. Sie können jetzt einige der kompliziertesten Peripherieelemente der MCU benutzen. Später, wenn das Event-System dazukommt, lernen wir noch eine neue Funktion des Timers kennen. Bis dahin aber erstmal viel Spaß beim Ausprobieren und Modifizieren der vorgestellten Funktionen! ◀

(150041)

Listing 5. Der Real-Time Counter (RTC).

```
void configure_rtc_count(void)
{
    struct rtc_count_config config_rtc_count;
    rtc_count_get_config_defaults(&config_rtc_count);

    config_rtc_count.prescaler      = RTC_COUNT_PRESCALER_DIV_1;
    config_rtc_count.mode           = RTC_COUNT_MODE_16BIT;
    rtc_count_init(&rtc_instance, RTC, &config_rtc_count);
    rtc_count_enable(&rtc_instance);
}
...

void configure_rtc_calendar(void) //configure the RTC
{
    struct rtc_calendar_config config_rtc_calendar;
    rtc_calendar_get_config_defaults(&config_rtc_calendar);
    config_rtc_calendar.clock_24h = true;

    alarm.time.year      = 2015; //first alarm date
    alarm.time.month     = 4;
    alarm.time.day       = 1;
    alarm.time.hour      = 0;
    alarm.time.minute    = 0;
    alarm.time.second    = 1;

    config_rtc_calendar.alarm[0].time = alarm.time;
    config_rtc_calendar.alarm[0].mask = RTC_CALENDAR_ALARM_MASK_YEAR;

    rtc_calendar_init(&rtc_instance, RTC, &config_rtc_calendar);
    rtc_calendar_enable(&rtc_instance);
}
```

DesignSpark Tipps & Tricks

Multilayer-Platinen

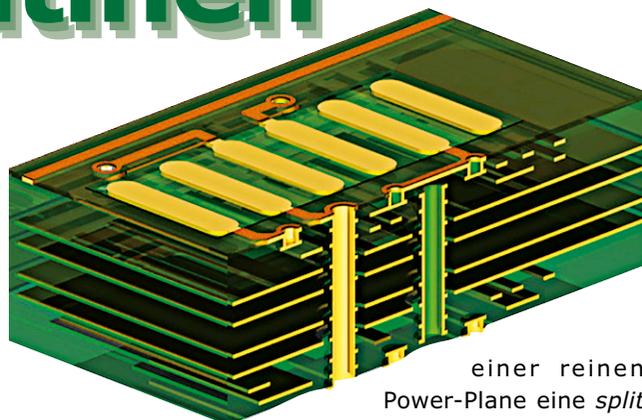
Von **Neil Gruending** (Kanada)

Bislang haben wir uns nur mit doppelseitigen Platinen beschäftigt, doch DesignSpark kann deutlich mehr als nur zwei Ebenen!

Der Einfachheit halber und weil das für „normale“ Projekte ausreicht, kamen in dieser Serie nur Platinen mit zwei Ebenen von Leiterbahnen zum Zuge. Darauf ist DesignSpark aber nicht beschränkt. Wie schon erwähnt sind vier und mehr Ebenen kein Problem. Hier geht es nun um verschiedene Ansätze für Multilayer-Platinen.

Die Schichtung der Ebenen

Der erste Schritt bei einer Multilayer-Platine ist es, sich darüber klar zu werden, welche Arten von Ebenen benötigt werden. Typische Ebenen sind: Signal-Layer, „power plane“ oder „split power plane“ für die Versorgung oder aber ein „mixed signal plane“, bei der so gut wie alles vorkommt. Auf einer *Signal-Layer* sind zwar alle Arten von Signalen, aber eben keine Versorgungsleitungen vorhanden. Bei der *Power-Plane* ist normalerweise lediglich eine große Kupferfläche mit Aussparungen vorhanden, z.B. eine Massefläche. Bei einer *split Power-Plane* wird mehr als eine Versorgungsspannung bzw. mehr als nur die Masse geroutet. Bei einer *mixed Signal-Plane* handelt es sich in der Regel um eine Power-Plane, auf der auch einige, aber eher wenige Leiterbahnen für Signale zu finden sind. DesignSpark unterstützt selbstverständlich alle diese Layer-Typen. Wenn man bei einer Signal-Layer eben zu den Bahnen für Signale dann doch noch einige Versorgungsleitungen verlegt, wird diese Lage eben zur *mixed Signal-Layer*. Genauso hat man statt



einer reinen Power-Plane eine *split Power-Plane*, wenn man weitere Versorgungsleitungen (oder Flächen) darauf anlegt. Eine richtige Power-Plane hat allerdings nur ein einziges Netz, und normalerweise kann man nur dann diese Ebene zur Power-Plane machen.

Layer-Festlegung

Der einfachste Weg zur Konfiguration der Layer bei DesignSpark ist der „New PCB Wizard“ (siehe **Bild 1**). Hier kann man gleich die Anzahl gewünschter Layer für die Platine festlegen. Die oberste Ebene ist dann *Layer 1*. Anschließend legt man die Layer für die Power-Planes fest. Wenn man mehr als vier Ebenen verwendet, dann ist man froh, dass DesignSpark spezielle Stackup-Optionen bietet. Bei einer vierlagigen Platine empfehlen sich z.B. die Layer 2 und 3 als Power-Planes. Wenn der Wizard seine Arbeit getan hat, kann man immer noch an der Layer-Schichtung und ihren Eigenschaften herumschrauben. Hierzu geht man in die Design-Einstellungen (Settings → Design Technology). Unter dem Layers-Tab werden alle Layer aufgelistet. Dort kann man sie editieren oder gar Layer zusätzlich anlegen. **Bild 2** zeigt ein Beispiel, bei dem per Wizard eine Platine mit acht Ebenen angelegt wurde. Layer 2 und 3 sind die Power-Planes. In DesignSpark ist eine Power-Plane eine Ebene mit dem Attribut „Power Plane“ – man

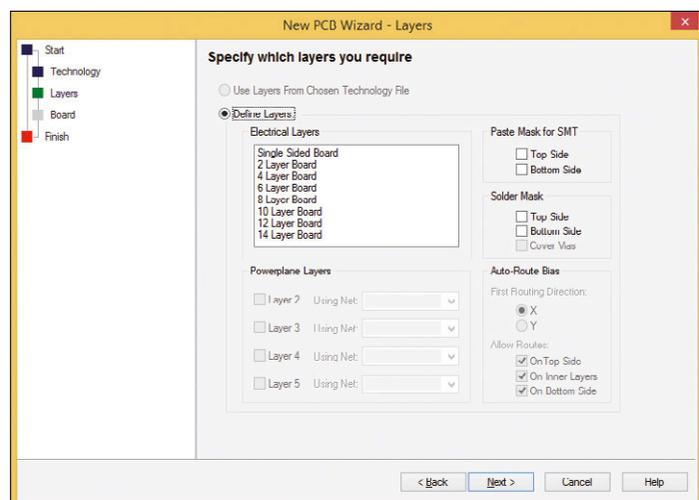


Bild 1. New PCB Wizard.

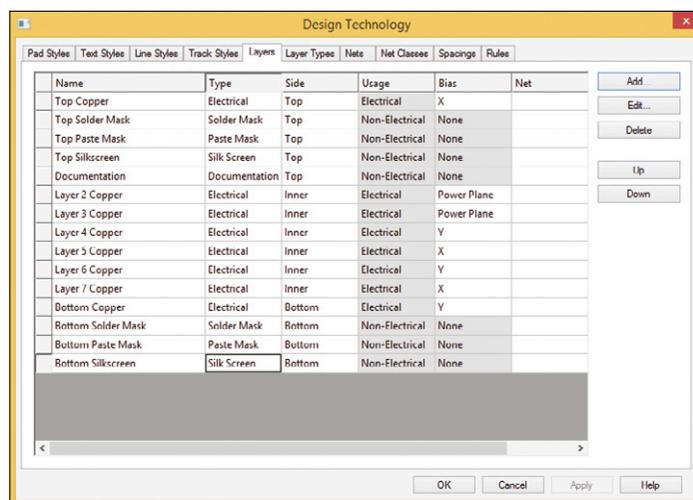


Bild 2. Layer-Tab im Dialog „Design Technology“.

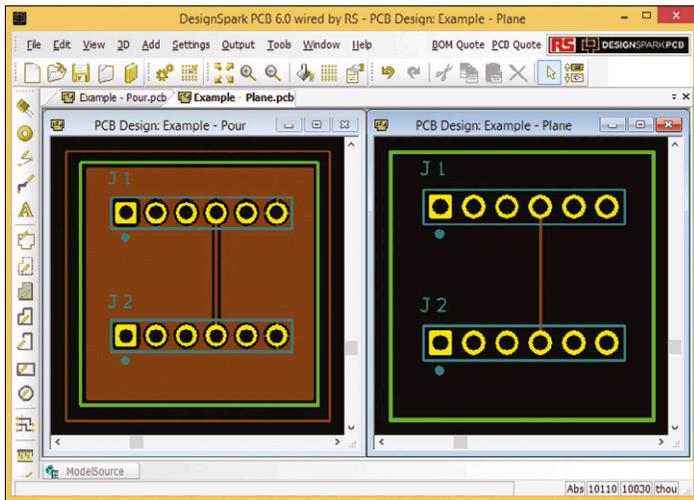
Advertorial für **DESIGNSPARK**

Bild 3. Pours vs. Planes.

sollte dieser Ebene normalerweise genau ein Netz zuweisen. An dieser Stelle kann man den verschiedenen Layern Labels wie „Top Signal“ oder „Ground Plane“ zuweisen, sodass ihre Bedeutung beim Routen eindeutig ist. Normalerweise vermeidet man das Verlegen von Leiterbahnen auf einer „Plane“-Ebene, da dies ja eine Fläche bezeichnet. In DesignSpark ist aber auch das kein Problem, wenn man es denn möchte. Diese Bahnen werden zwar genau so angezeigt wie andere auf Signal-Layern, doch beim Export von Gerber-Dateien sorgt DesignSpark dafür, dass um sie herum die Versorgungsflächen fließen.

Pours vs. Planes

Wie schon erwähnt ziehe ich so genannte „copper pours“, also mit Kupfer gefüllte Flächen auf normalen Layern den reinen Planes vor, die als komplette Flächen Masse oder eine Spannung liefern. Die Pours bedeuten etwas mehr Arbeit. Im Beispiel von **Bild 3** ist die linke Platine via Pour und die rechte mit einer Power-Plane realisiert. Formalelektrisch gibt es keinen Unterschied. Der Gerber-Export ist sogar fast gleich. Der eigentliche Unterschied liegt darin, dass auf der linken Seite gleich in DesignSpark deutlich wird, wie die Platine aussieht. Die rechte Variante benötigt zur Beurteilung der Power-Planes den Import der Gerber-Daten in ein anderes Programm. Pours sind zudem extrem flexibel, wenn es um eine Spannungsversorgung via Planes geht. Wenn man z.B. eine kleine Fläche mit 1,8 V in einer 3,3-V-Plane benötigt, warnt DesignSpark zwar kurz, lässt dies dann aber zu. Die bessere Methode ist allerdings, schlicht auf einer Ebene zwei Pours zu verwenden - je eine mit 1,8 V und 3,3 V - und die Füllreihenfolge so festzulegen, dass z.B. die Fläche mit 3,3 V die Fläche mit 1,8 V umfließt.

Gerber-Export

Glücklicherweise gibt es beim Gerber-Export gegenüber doppelseitigen Platinen kaum Unterschiede. **Bild 4** zeigt den Dialog „Output Manufacturing Plots“ für die Beispiel-Platine mit acht Lagen. Im Settings-Tab sieht man die ganzen Einstellungen für

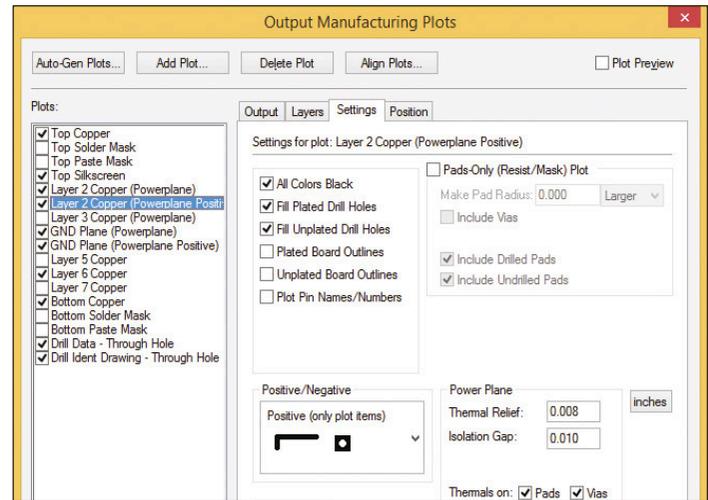


Bild 4. Gerber Output Options.

alle Layer. Normalerweise kann man diese unangetastet lassen. Hier könnte man noch die Isolationsabstände und die „thermal reliefs“ (die kleinen Stege um Pads) für Power-Planes festlegen. Es fällt auf, dass nicht alle Layer ausgewählt sind. DesignSpark hat nur die Layer berücksichtigt, die in diesem Beispiel auch verwendet wurden. „Layer 2 Copper“ und „GND Plane“ sind mit „(Powerplane)“ bzw. „(Powerplane Positive)“ ausgezeichnet. Normalerweise werden Power-Planes als Negativ von Planes gezeichnet. Da ich aber auf diesen beiden Layern auch Leiterbahnen platziert habe, funktioniert deren Darstellung nicht als Negativ. DesignSpark muss hier also das Positiv der Leiterbahn und der Pads ausgeben. Wenn man so beides kombiniert, erhält man die Ausgabe von **Bild 5**. Die schwarzen Stellen auf der Platine entsprechen der Plane, während die Leiterbahn violett gezeichnet ist. Die Isolation zwischen Plane und Leiterbahn ist dunkelblau. Eine solche Isolation umrahmt auch die Platinenkanten.

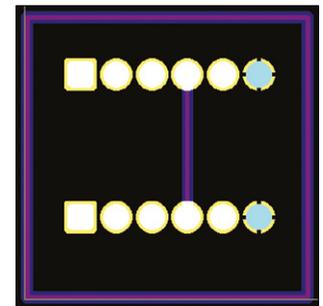


Bild 5. Gerber-Ausgabe der Massefläche.

Bevor man die Gerber-Dateien exportiert, sollte man ein extra Verzeichnis via „Options...“ festlegen, in dem die exportierten Dateien landen. Wenn man dann auf „Run“ klickt, werden diese Dateien dorthin statt in das Projekt-Verzeichnis gelegt.

Fazit

In dieser Folge ging es darum, wie man die verschiedenen Ebenen einer Multilayer-Platine erzeugt. Es war gleichzeitig das Schlusskapitel dieser mehrteiligen Anleitung zu DesignSpark PCB. Ich hoffe, dass Ihnen diese Beiträge gefallen haben und dass meine Anregungen bei Ihrem nächsten Projekt hilfreich sind! ◀

(150023)

Einfach WunderBar!

IoT ausgepackt

Von **Jaime González-Arintero** (Elektor)



Bild 1. Auspackzeremonie. Das andere Schoko-Gehäuse ist ein per 3D-Druck gefertigter Prototyp, den wir vor kurzem von relayr erhielten.

Das IoT (Internet of Things) entwickelt sich zum Zankapfel der Elektronikwelt. Die (Internet-)Medien haben ihren Anteil am aktuellen *Hype* und haben damit meiner Meinung nach nicht unerheblich zur Verwirrung der Öffentlichkeit beigetragen. Die Frage aber ist, ob das sogenannte ubiquitäre Computing tatsächlich die Welt verändert oder ob es lediglich darum geht, kleine Luxusprobleme entwickelter Gesellschaften zu lösen.

Statt den Hype weiter zu steigern ist es wohl sinnvoller, praktisch und konkret zu werden. Es sind nämlich schon etliche IoT-Devkits verfügbar. Beim WunderBar handelt es sich um etwas ganz Spezielles: Es stammt nämlich nicht von einem der Platzhirsche, sondern ist ein Produkt des Berliner Startups **relayr** [1].

Wunder?

Sieht nett aus, na und? Was ist „WunderBar“ eigentlich genau? Ein Starter-Kit für das Internet der Dinge. Wenn Sie da fragend die Augenbraue heben, geht es Ihnen nicht anders als mir, als ich diese „Erklärung“ zum ersten Mal hörte. Das war letztes Jahr, anlässlich der **electronica** in München. Dort wunderten sich viele Besucher, was für merkwürdige Schokoriegel überall am Elektor-Stand herumlagen. Also versprach ich, dass ich zur Aufklärung der geheimnisvollen Aura beitragen werde, die dieses Kit umgibt.

Bevor man sich in den Details verliert, zunächst ein Überblick: Das Kit besteht aus einem Master-Modul und einem Satz von sechs Sensor-Modulen (dazu später mehr). Im Prinzip also eine Master/Slave-Anordnung. Ein Sensor-Modul erfasst Werte und schickt die Daten via BLE (Bluetooth Low Energy) zum Master. Das Master-Modul wiederum stellt via WLAN eine Verbindung mit dem Internet her und überträgt die Sensordaten in eine Cloud. Alle von den Sensoren eingesammelten Daten landen also in der Cloud und können dort später von anderen Geräten mit

entsprechender Berechtigung abgeholt und verwertet werden. Man kann aber auch einen bestimmten Sensor „öffentlich“ schalten (hierzu später mehr). Wichtig zu wissen ist, dass man bei der Abfrage von Sensor-Werten eben nicht direkt ein Sensor-Modul abfragt, sondern „lediglich“ Werte aus der Cloud einliest, die zuvor vom Sensor- über das Master-Modul dort abgelegt wurden.

Geheim?

Ganz und gar nicht, im Gegenteil: Open Source. Beim WunderBar handelt es sich um ein schwer *software-lastiges* Entwicklungs-Kit, das durch einzigartige Code-Beispiele fasziniert. Hinzu kommen die SDKs und Libraries sowie die umfangreiche Dokumentation. Nicht nur die Soft-, sondern auch die Hardware ist Open Source. Jeder hat Zugang zu Allem und kann alles an seine Bedürfnisse anpassen. Da Elektroniker auf harte Ware und Silizium stehen, lohnt sich ein Blick auf [2].

Die Verpackung des WunderBar ähnelt einer Pralinschachtel. Schon das Auspacken triggert das Kind im Manne (oder der Frau), denn es fühlt sich an wie Weihnachten. Der aufgeklappte Deckel gibt den Blick frei auf so etwas wie Kunststoffgehäuse in Schokoladenform, daneben liegen dann noch ein ordinäres USB-Kabel, ein Lithium-Akku mit 3,7 V bei 130 mAh für das Master-Modul, fünf Knopfzellen des Typs CR2032 für die Sensor-Module, Klebeband-Stücke für die Module und natürlich die sieben Boards (**Bild 1**). Alle Module befinden sich auf einer

einigen, perforierten Platine, die man vor Benutzung auseinander bricht. Zuvor sollte man allerdings das Master-Modul samt angeschlossenem Akku via USB mit einem PC etc. verbinden. Dabei werden dann auch alle Sensor-Module mit versorgt, selbst wenn sie noch ohne eigene Knopfzelle sind. Nach dem Brechen sorgen die Nasen der Perforation dafür, dass die Module in den Gehäusen halten.

Auf dem Master-Modul versieht ein 32-bit-ARM-Controller des Typs Cortex-M4 MK24 aus der Kinetis-Serie von Freescale seinen Dienst. Der Chip verfügt über 1 MB Flash-Speicher für Programme sowie über viel Peripherie und Schnittstellen (inkl. SPI, I²C und USART) in einem extrem dünnen XFBGA-Gehäuse, das man von Hand weder bestücken noch entlöten könnte. Für die Verbindung mit dem Internet sorgt das WLAN-Modul GS1500 von GainSpan. BLE ist mit dem SoC nRF51822 von Nordic Semiconductor realisiert, der im 48-Pin-QFN-Gehäuse daherkommt. Hierdurch wird der ARM-Controller von der Aufgabe entlastet, mit den Sensor-Modulen zu kommunizieren. Auch auf den Sensor-Modulen findet sich das gleiche Nordic-SoC, es spielt aber hier die Hauptrolle. Die sechs Sensor-Module können unterschiedliche Größen erfassen. Stichworte sind Licht/Farbe & Distanz, Accelerometer/Gyroskop, Temperatur/Luftfeuchtigkeit, Lautstärke und IR-Sender (oha!). Das letzte Modul besitzt eine vierpolige Verbindung zum Anschluss anderer Elektronik. Damit sind die beiden letzten Module keine eigentlichen Sensoren, aber die damit möglichen Anwendungen sind umso interessanter. Alle Module verfügen über einen achtpoligen Debug-Port und einige Zusatz-Pins für den Anschluss von Erweiterungen.



Bild 2. Wirklich jeder wird Sie fragen, was das denn für komische Dinger sind.

Onboarding

Die Basis-Konfiguration von WunderBar erfolgt in einem „onboarding“ genannten Prozess. Die schrittweise Anleitung findet sich auf der Webseite von relayr [3] und ist super einfach nachzuvollziehen. Man benötigt lediglich die „relayr Manager App“ [4] auf seinem Smartphone (Android oder iOS) plus

Wirklich jeder wird Sie fragen, was das denn für komische Dinger sind.

ein mit Strom versorgtes Master-Modul. Nachdem die kleinen Hürden mit dem WLAN genommen sind, synchronisiert sich die App mit dem Master-Modul. Anschließend werden vorhandene Sensor-Module gescant. Bevor es damit losgeht, wird man gebeten, die Bluetooth-Verbindung des Smartphones zu aktivieren. Damit kann jedes Modul individuell und direkt per BLE angesprochen werden (dies kann in der App aktiviert werden). Wenn das nicht nötig ist, kann man Bluetooth wieder deaktivieren und das Kit wird dennoch perfekt weiter arbeiten. Nach diesem „onboarding“ kann man die einzelnen Module von der Gesamtplatine brechen, mit Knopfzellen bestücken und sie dann in die kleinen schokoartigen Gehäuse drücken (Bild 2). Jetzt kann man mit den Sensoren herumspielen und sie für beliebige eigene Zwecke einsetzen. Bild 3 zeigt eines dieser Stückchen e-Schokolade auf einem Fensterbrett im Elektor Castle. In diesem Fall war es der Sensor für Temperatur/Luftfeuchtigkeit. Die Daten im Screenshot meines Smartphones beweisen, dass es sich wirklich um die Niederlande handelt. Als frierender Spanier habe ich doch das Recht, das zu beklagen, oder?

Apps und nochmehr Apps

Wie schon erwähnt ist es die Software, die das Besondere von WunderBar ausmacht. Es gibt nicht nur eines, sondern

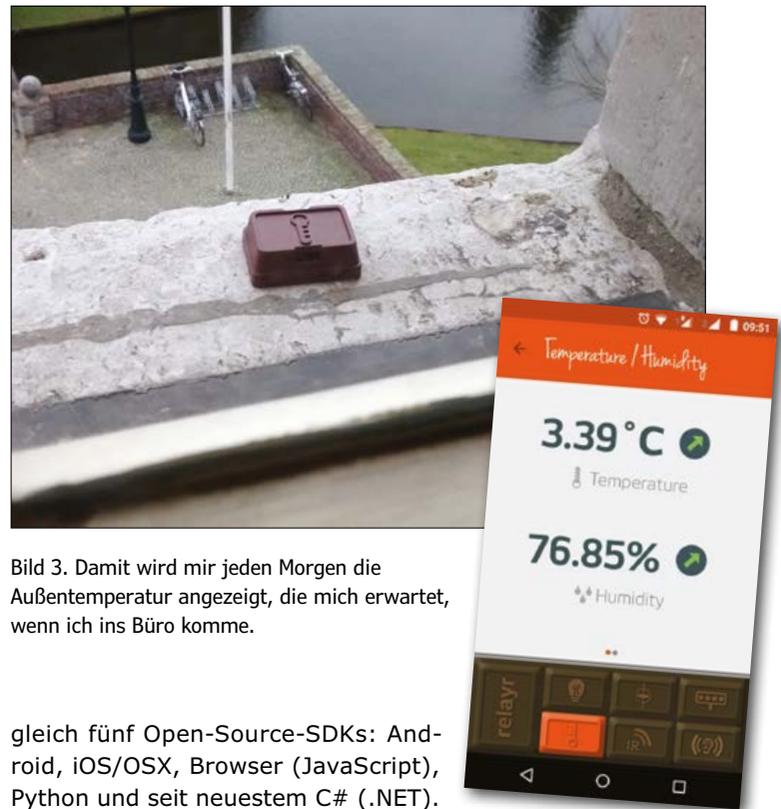


Bild 3. Damit wird mir jeden Morgen die Außentemperatur angezeigt, die mich erwartet, wenn ich ins Büro komme.

gleich fünf Open-Source-SDKs: Android, iOS/OSX, Browser (JavaScript), Python und seit neuestem C# (.NET). Um loszulegen benötigt man einen API-Key, den man da bekommt, wo auch alles andere her ist [5]. Hier kann man auch den aktuellen WunderBar-Status einsehen und die Werte

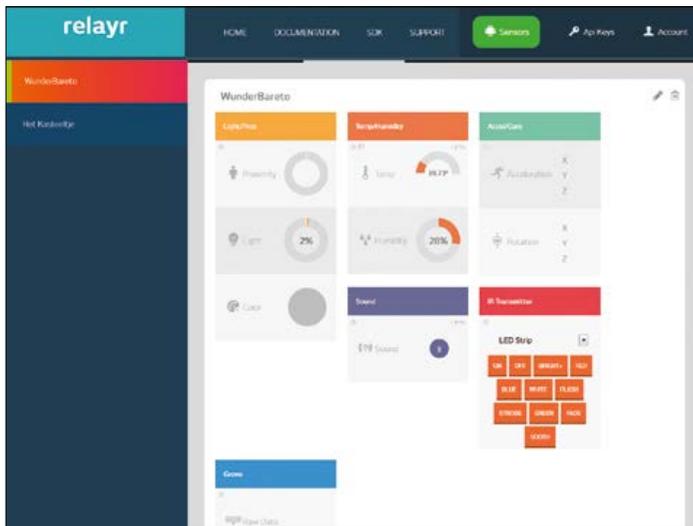


Bild 4. Im Dashboard kann man unter „Sensors -> My Devices“ den Status aller WunderBars betrachten, die mit dem eigenen Account verknüpft sind.

der Sensoren abfragen. Zwecks entfernter Tests habe ich ein Exemplar im Büro und eines zuhause platziert, wie man auf der linken Seite der Anzeige sehen kann (Bild 4). Man kann Sensorwerte auch mit einem Script öffentlich zugänglich machen. Ohne dieses bleiben die Messungen privat und nur dem Besitzer via Dashboard oder App einsehbar. Außerdem kann man die Messrate für jeden einzelnen Sensor einstellen. Es macht ja wenig Sinn, die Temperatur jede Sekunde abzufragen.



Bild 5. Meine Süßigkeiten, mein lebenswichtiges Wörterbuch Spanisch/Niederländisch und meine kleine, anachronistische Cloud in Form einer Diskette (was ist das?) auf dem Schreibtisch. Natürlich könnte ich mein Büro abschließen und den Schlüssel mitnehmen, aber dann könnte ich ja gar keine Bonbon-Diebe unter meinen Kollegen überführen :D.

App „TellMeWhen“ erlaubt die Erstellung von Regeln, die von bestimmten konfigurierbaren Messwerten getriggert werden. Zum Beispiel wenn ich verhindern will, dass liebe Kollegen sich in meiner Abwesenheit an meinen Goodies vergreifen (wozu es genug Gelegenheiten gibt, denn ich hasse den Morgen). In diesem Fall platziere ich den Licht/Nährungs-Sensor im Büro (Bild 5), was mich sofort darüber informiert, wenn jemand die Tür öffnet – egal wo ich gerade bin. Ertappt!

Fazit

Was davon zu halten ist? Die Zeit wird es zeigen. Zu Anfang war ich ja skeptisch gegenüber dem IoT-Hype, als man dem Begriff gar nicht mehr entkommen konnte, aber dennoch eher nutzlose Anwendungen präsentiert wurden. Mittlerweile bin ich aber zuversichtlicher, dass das Internet zu den Dingen kommen wird. Wie bei anderen neuen Technologien wird dies zunächst langsam und eher graduell geschehen. Die Anwen-

Schokolade bei Elektor.TV

Schon vor einiger Zeit haben wir WunderBar bei Elektor.TV vorgestellt. Wer die Auspack-Zeremonie miterleben will, begeben sich zu unserem Kanal [6]. Und wer immer in Sachen Elektor-Labor auf dem Laufenden bleiben möchte, sollte nicht vergessen sich zu registrieren.



dungsmöglichkeiten in der Industrie und im praktischen Leben sind unzählige. Man stelle sich nur einmal vor, dass man sofortigen Zugriff auf die Daten von tausenden Sensoren hat.

Hinzu kommt die Möglichkeit zum Zugriff auf historische Daten zu jeder Zeit an jedem Ort der Welt. Dabei geht es selbstverständlich um viel umfassendere Projekte als die simplen Anwendungen, die hier zur Sprache kamen. Moment mal: Verdammt, da hat doch tatsächlich jemand meine Bürotür geöffnet! ◀

(140474)

Weblinks

- [1] <http://relayr.io>
- [2] <http://po.st/wunderbarhw>
- [3] <https://relayr.io/wunderbar>
- [4] <http://po.st/wunderbarapp>
- [5] <https://developer.relayr.io>
- [6] <http://po.st/wunderbarunboxing>

Tipps und Tricks

Von Lesern für Leser

Zusammengestellt von **Burkhard Kainka**

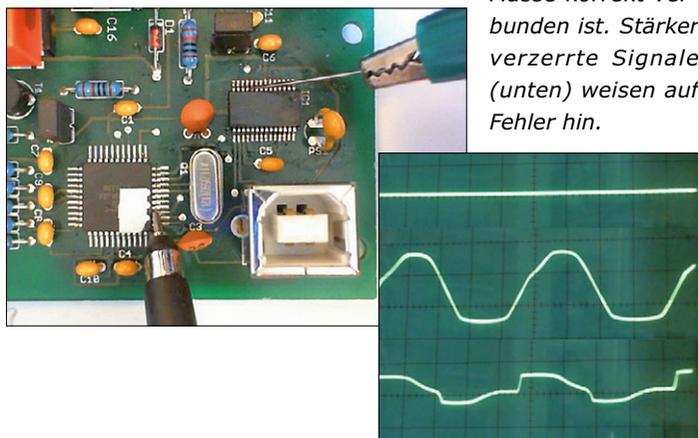
Hier sind wieder clevere Lösungen, die das Elektronikerleben leichter machen.



Oszilloskop als Durchgangsprüfer

Von Burkhard Kainka

Bei der Fehlersuche auf Platinen mit SMD-Bestückung hat man es mit einem Ohmmeter nicht leicht. Hier eine neue Methode, bei der ein Sinusgenerator und ein Oszilloskop verwendet werden. Mit der Oszi-Messspitze berührt man einen fraglichen Pin. Mit der Nadel am Signalgenerator streicht man über die Pins eines anderen ICs, wo eine Verbindung bestehen sollte. Man sieht dann sogar aus dem Augenwinkel: Keine Verbindung, gute Verbindung oder Unterbrechung bzw. schlechte Verbindung. Das Oszillogramm in der Mitte zeigt eine gute Verbindung. Der unten verbeulte Sinus kommt von den CMOS-Schutzdioden und ist ein gutes Zeichen. Wichtig ist es bei dieser Methode, dass auch die Masse korrekt verbunden ist. Stärker verzerrte Signale (unten) weisen auf Fehler hin.



mit dem Hautwiderstand! Man berührt mit den Fingern den Pluspol der Batterie und die Anode der LED, während die Kathode direkt an den Minuspol gehalten wird. Das Foto zeigt, dass es funktioniert. Bei einem Hautwiderstand von rund 1 M Ω beträgt der LED-Strom einige μ A.



AVR-Port toggeln

Tipps von Andreas Riedenauer (Inteltek Mitte GmbH)

Wenn man in das PIN-Register schreibt, toggelt der zugehörige Portpin. Ein Vorteil ist, dass das sehr schnell geht. Der Screenshot zeigt einen Test in Bascom mit dem ATmega328 des Arduino Uno (es funktioniert!). Und beim ATtiny13 geht es auch, getestet mit einem Blink-Programm.

```
'LEDblink.BAS
$regfile = "m324pdef.dat"           'ATmega328p
$crystal = 16000000                 '16 MHz

Config PORTB = Output

'13.2.2 Toggling the Pin
'Writing a logic one to PINxn toggles the value of PORTxn.
'Independent on the value of DDRxn.
'Note That The Sbi Instruction Can Be Used To Toggle One Single Bit In A Port.

Do
  PINB.5 = 1                        'BS toggeln
  Waitms 500
Loop

Do
  PORTE.5 = 1                        'LED on
  Waitms 500                          '500 ms
  PORTE.5 = 0                        'LED off
  Waitms 500                          '500 ms
Loop
```

Erst nach längerem Suchen entdeckt man es auch im Datenblatt (Beispiel ATtiny13): The Port Input Pins I/O location is read only, while the Data Register and the Data Direction Register are read/write. However, writing a logic one to a bit in the PINx Register, will result in a toggle in the corresponding bit in the Data Register. ◀

(150027)

Einfacher Batterieprüfer

Nach einer Idee von Eckhard Koch

Die grünen LEDs neuester Bauart kommen je bekanntlich schon mit Strömen unter einem Mikroampere aus. Das eröffnet ganz neue Möglichkeiten. Ein Batterieprüfer kommt jetzt ohne zusätzlichen Widerstand aus. Es funktioniert einfach



Sie haben selbst eine clevere Lösung für etwas wirklich Fummeliges? Wenden ein Bauteil oder Werkzeug auf ungewöhnliche Weise an? Haben eine Idee, wie man ein Problem einfacher oder besser angehen könnte, als das bisher gelöst wurde? Schreiben Sie uns – für jeden Tipp, den wir veröffentlichen, loben wir 40 Euro aus!





Codebender Arduino im Internet

Von **Tam Hanna** (D)

Massimo Banzis Entwicklungsumgebung ist nicht jedermanns Sache. Eine Gruppe griechischer Entwickler konstruierte 2012 – mehr zum Spaß – eine im Browser lauffähige Arduino-IDE, die sich im Laufe der Zeit zu einer ernstzunehmenden Alternative entwickelt hat. Dank Cloudspeicher muss man sich um Datenverluste keine Sorgen mehr machen. Partnerschaften mit Modulherstellern sorgen dafür, dass die meisten Shield-Libraries mit einem Klick ins Projekt wandern.

Der PIC-Serie von Microchip gebührt die Ehre, weltweit Horden von Elektronikern mit der Programmierung von Mikrocontrollern vertraut gemacht zu haben. Massimo Banzis Erfolg mit Arduino lag unter anderem daran, dass das für Nicht-Elektroniker schwierige Hantieren mit Chips und Brennern ersatzlos entfiel. Ein Arduino wird per USB mit dem Rechner verbunden, der Rest ist „Magie“. Mit der Online-Entwicklungsumgebung Codebender [cb] geht das Übertragen von Firmware genauso einfach. Unter Windows und Linux laufende Versionen von Chrome und Firefox lassen sich mit einem Browser-Plugin versehen, das die Kommunikation zwischen Arduino-Board und Computer erledigt (siehe **Bild 1**). Allerdings werden längst nicht alle Arduino-Boards unterstützt (siehe Kasten). Wer Codebender nutzen will, muss sich registrieren. Ein Einrichtungsassistent führt durch alle notwendigen Schritte, einschließlich der Installation des Plugins. Danach kann man bereits das erste Codebeispiel (Blink.ino) auf das Arduino-Board laden.

Eine Frage der IDE

Die Oberfläche von Codebender präsentiert sich wie in **Bild 2** gezeigt. Der rechte Teil des Bildschirms wird vom Codeeditor eingenommen, der auf dem von Cloud9 entwickelten ACE-Editor basiert. Er verhält sich – im Großen und Ganzen – wie Eclipse und Co., bietet aber im Vergleich zur Arduino-IDE wesentlich intelligentere Ergänzungsvorschläge (IntelliSense).

Der Kompilationsworkflow ist – teilweise – an der Arduino-IDE angelehnt. Klicken Sie auf den Button „Verify Code“, um den im Editor befindlichen Code probeweise zu kompilieren. Im Erfolgsfall blendet das Programm die Größe des Kompilats ein, das sich sodann durch Anklicken von „Run on Arduino“ auf das Board kopieren lässt.

Die darüber befindliche Statusleiste ermöglicht das Wechseln zwischen mehreren Sketchdateien. Der nach unten zeigende Pfeil erlaubt das Herunterladen der gezippten .ino-Datei und der in den Controller geschriebenen .hex-Datei, um diese lokal

auf dem eigenen Computer abzuspeichern. Im Moment gibt es keinen Weg zum Herunterladen der .elf-Datei: die Analyse der Kompilate wird dadurch erschwert. Chrome blendet beim Anklicken der Download-Option zudem eine Fehlermeldung über nicht gespeicherte Informationen ein, die aber ohne praktische Bedeutung ist.

Über den auf der rechten Seite des Bildschirms eingeblendeten blauen Pfeil lässt sich das Sozial-Feature der IDE aktivieren. Es erlaubt das Teilen des Quellcodes per Twitter, Facebook und LinkedIn.

Fehler ausräuchern

Der bekannte AVR-GCC liefert Fehlermeldungen, deren Interpretation ein tiefergehendes Verständnis von C/C++ voraussetzt. Der modulare Compiler Clang (siehe Kasten), der von Codebender verwendet wird, ist dagegen für seine einsteigerfreundlichen Hinweise bekannt.

Ein Beispiel sehen Sie in **Listing 1**, der Code enthält drei Fehler. Nach dem Anklicken des „Verify Code“-Buttons wirft Codebender einen detailreichen Fehlerbericht aus, der die betroffenen Passagen sogar mit Tilden markiert. Die offizielle Arduino-IDE zeigt sich im Vergleich weit weniger gesprächsbereit.

Populärwissenschaftliche Magazine postulieren immer wieder, dass Codebender auch die eigentliche Kompilation mit Clang erledigt. Dies ist insofern falsch, als es noch kein zuverlässiges Backend gibt, das die Intermediärsprache LLVM für den AVR übersetzt. Ein vor längerer Zeit veröffentlichter Blogbeitrag [TSIP] erlaubt die Erstellung des in **Bild 3** gezeigten Workflows, der die Kompilation im Detail beschreibt.

Kompilate sind trotzdem nicht 1:1 austauschbar. Version 1.5.7 der Arduino-IDE brauchte zur Realisierung des weiter oben gezeigten Programms 1030 Byte Codespeicher, während das von Codebender realisierte Kompilat 1082 Bytes einnahm. Bei komplexeren Beispielen fällt die Situation noch gravierender aus: ein in der Arduino-IDE 3052 Byte großes Produkt präsentiert sich in Codebender als 3256 Bytes groß. Komplette „leere“ Projekte sind in Codebender ebenfalls um einige Bytes größer – die Bootloader dürften sich (marginal) unterscheiden.

Codierte zu mehr

Prozessrechner der Ausführung Banzi haben sich im Laufe der letzten Jahre als Quasistandard im Maker- und semiprofessionellen Bereich etabliert. Die Marktbreite sorgt für eine Vielzahl von Erweiterungen, die das Ansprechen diverser Bauteile erleichtern. Diese werden im Allgemeinen als Library bezeichnet – Codebender bietet mehrere hundert davon an. Die Nutzung von vorgefertigten Bibliotheken beginnt mit dem Besuchen der in **Bild 4** gezeigten Liste [Lib]. Um die Dateien zu verwenden, genügt es, sie über das angezeigte Include-Statement in das Projekt einzubinden.

Mach' die Wolke

Codebender erlaubt die Verwendung eigener Bibliotheken: neben dem klassischen Aspekt des Recyclings können Sie die

Eine Frage des Prozessors

Codebender unterstützt sowohl Arduino Uno als auch Leonardo, hat aber keine Ahnung von komplexeren Systemen wie dem Yun oder dem auf ARM-Technologie basierenden Due.

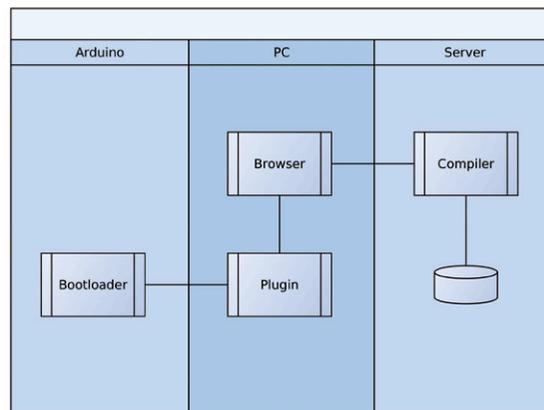


Bild 1. Codebender läuft im Standardbrowser. Für das Übertragen von Firmware auf ein Arduinoboard wird ein Plugin benötigt, das es für Firefox und Chrome gibt.

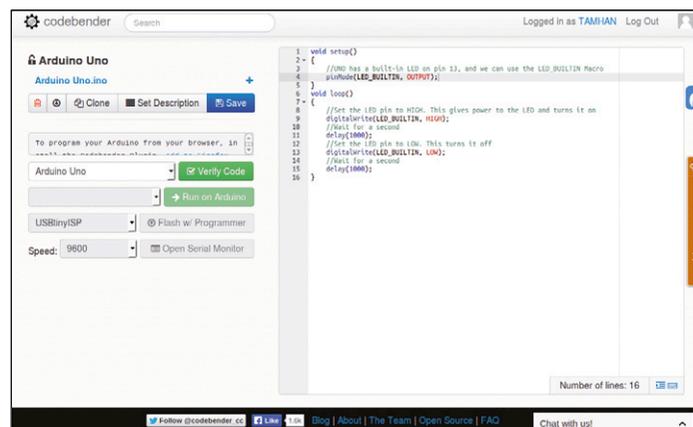


Bild 2. Der Codeeditor ist das zentrale Element der Online-Anwendung.

Listing 1. Ein kleines Beispiel mit 3 Fehlern.

```

void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
}
void loop()
{
  digitalWrite(LED_BUILTIN, HEIH);
  delaI(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000)
}
  
```

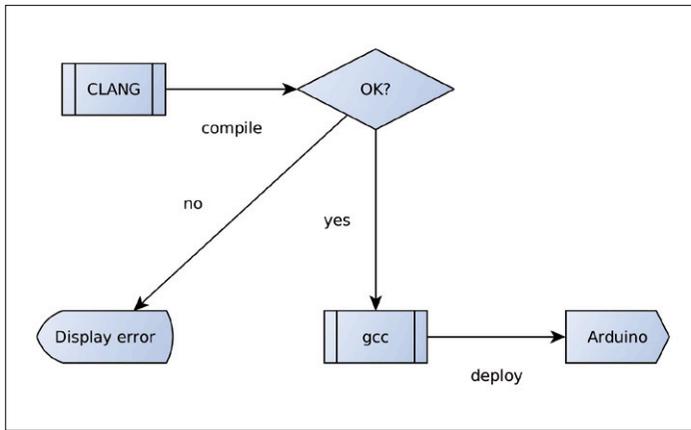


Bild 3. Zur Fehlererkennung im Quellcode wird das modulare Compiler-Frontend Clang genutzt. Das hiermit erstellte Zwischenprodukt wird allerdings nicht weiterverwendet.

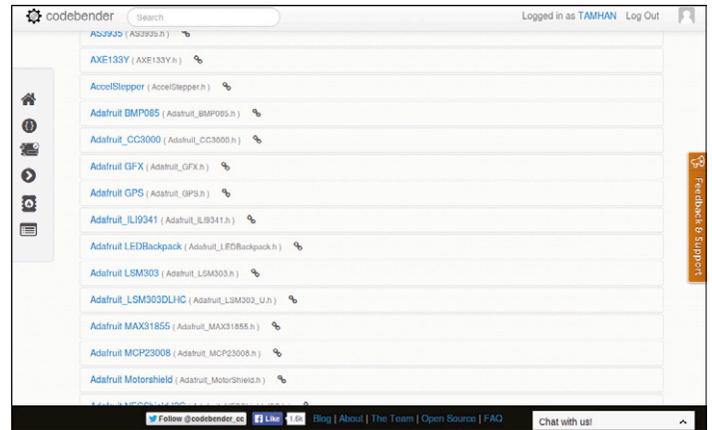


Bild 4. Liste der Bibliotheken, die man einfach in eigene Arduino-Programme einbinden kann.

Clang und LLVM

Clang ist ein seit Jahrzehnten bestehendes Compiler-Projekt, das eine Alternative zum hinreichend bekannten GCC-Compiler darstellt. Es erstellt Code für eine als LLVM bezeichnete virtuelle Maschine, der im nächsten Schritt statisch oder dynamisch in Code für das eigentliche Zielsystem konvertiert wird.

Der Vorteil dieser Architektur besteht darin, dass sich der Compilerentwickler keine Gedanken über das Zielsystem machen muss. Er beschränkt sich darauf, seine Sprache in

möglichst optimalen LLVM-Code umzuwandeln (Frontend). Die Adaptierung von LLVM in echten Maschinencode erfolgt durch darauf spezialisierte Entwickler, die von der Hochsprache keine Ahnung haben müssen (Backend).

Im Fall von Codebender liegt ein Problem darin, dass es – im Moment – noch keine nutzbare Backend-Implementierung für AVR-Controller gibt. Codebender nutzt Clang somit nur zur Kompilation in Intermediärcode – wenn dieser Prozess durchläuft, so gilt das Programm als fehlerfrei und kann an GCC weitergegeben werden.

Fehlerbericht von Codebender:

```

Arduino Uno.ino:7:28: error: use of undeclared identifier 'HEIH'
    digitalWrite(LED_BUILTIN, HEIH);
                               ^
Arduino Uno.ino:8:2: error: use of undeclared identifier 'delaI'; did you mean 'delay'?
    delaI(1000);
    ^~~~~
    delay
hardware/arduino/cores/arduino/Arduino.h:107:6: note: 'delay' declared here
void delay(unsigned long);
    ^
Arduino Uno.ino:10:13: error: expected ';' after expression
    delay(1000)
                ^
                ;
  
```

Fehlerbericht der offiziellen Arduino-IDE:

```

sketch_jan24a.ino: In function 'void loop()':
sketch_jan24a.ino:7:28: error: 'HEIH' was not declared in this scope
sketch_jan24a.ino:8:12: error: 'delaI' was not declared in this scope
sketch_jan24a.ino:11:1: error: expected ';' before '}' token
  
```

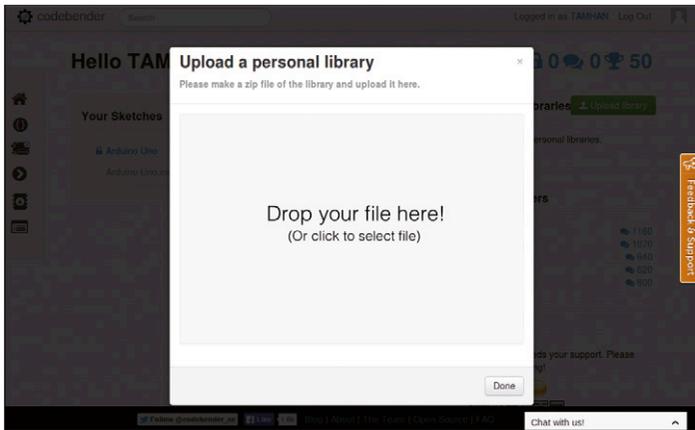


Bild 5. Codebender erlaubt das Hochladen eigener Libraries in die „Wolke“.

normalerweise nur aus einer Datei bestehenden Sketches auf diese Art und Weise „modularisieren“.

Dazu müssen Sie sich auf CodeBender.cc einloggen, um die in **Bild 5** gezeigte Startseite auf den Bildschirm zu holen. Der oben rechts eingeblendete Dialog „Personal Libraries“ erlaubt das Hochladen von .zip-Archiven, die aus .c- und .h-Dateien bestehen.

Von Ihnen hochgeladene Bibliotheken lassen sich ab diesem Zeitpunkt wie normale Bibliotheken verwenden. Im Fall einer Namenskollision bevorzugt Codebender die von Ihnen hochgeladene Version gegenüber der im Framework enthaltenen Variante.

Code herbei

Zu guter Letzt sei an dieser Stelle darauf hingewiesen, dass der Quellcode von Codebender offengelegt ist. Die Entwickler nutzen GitHub zur Teilung des Codes – öffnen Sie [github] in ihrem Webbrowser, um die diversen Module auf ihren Rechner herunterzuladen.

Von Seiten der Entwickler werden Änderungswünsche über den unter [IT] angebotenen Issue Tracker entgegengenommen, in dem Nutzer Ideen anmelden und für ihre Implementierung stimmen können. Zudem ist es über das am unteren Ende der Webseite befindliche Chatfenster möglich, direkt Kontakt zu einem diensthabenden Entwickler aufzunehmen.

Im Moment gibt es noch keine fixen Verträge, die die Verwaltung von nicht-öffentlichen Projekten erlauben. Wer unter support@codebender.cc nachfragt, kann sein Interesse bekunden: es wäre durchaus vorstellbar, dass Betatester schon jetzt einige der an sich nur für zahlende Kunden vorgesehenen Funktionen ausprobieren können.

Fazit

Codebender setzt die Speicherung des Codes in der Cloud voraus: von Haus aus gibt es im Moment nicht einmal eine Möglichkeit, Projekte als privat zu markieren. Zudem sind die Kompilate durch die Bank um 5 bis 15 % umfangreicher.

An quelloffenen Produkten arbeitende Entwickler sollten einen Umstieg ins Auge fassen: die verbesserte IntelliSense-Funktion, der intelligentere Compiler und das automatische Hosting des Codes sind Gold wert.

Anbieter von Bibliotheken und Shields sollten aus eigennützi-

► Wünsche der Community sind willkommen

gen Gründen über eine Partnerschaft nachdenken: die Griechen sind aufgrund des Library-Systems stolze Besitzer einer riesigen Menge von Applikationscode. Eventuelle Probleme bei API-Änderungen lassen sich damit im Voraus abschätzen; denn Codebender kann für zahlende Kunden ermitteln, wie viel Prozent des Codes dann nicht mehr kompilierbar wird. ◀

(140519)

Weblink

[cb] www.codebender.cc

[TSIP] <http://blog.tzikis.com/?p=454>

[Lib] <https://codebender.cc/libraries>

[IT] <http://feedback.codebender.cc/forums/165703-general>

[github] <https://github.com/codebendercc>

— Anzeige



HAMMOND
MANUFACTURING®

Gehäuse für
Raspberry Pi, Arduino und
viele andere
Einplatinencomputer

- Gehäuse
- Plattform

+ 44 1256 812812

sales@hammondmfg.eu /1593HAM.htm



Sie haben die Wahl.

- Gehäuse für jeden Rundumschutz
- Plattform für jeden Rundumzugriff
- Designspezifische Ausführungen für alle beliebten Modelle
- Alle Details unter hammondmfg.com



/1593HAMEGG.htm

Integrierter Gyrtator TCA580

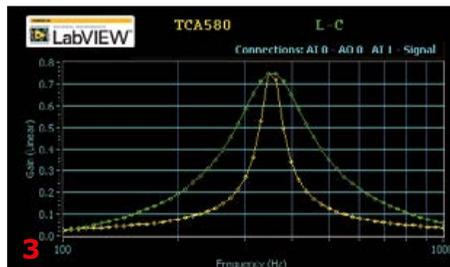
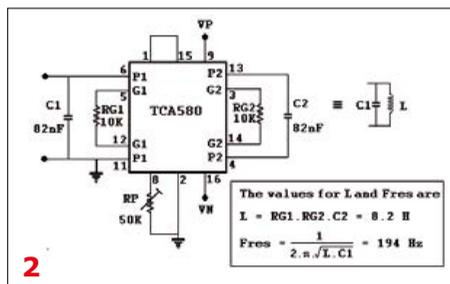
Seltene Bauteile (14)

Von **David Ashton** (Australien)

Vor einiger Zeit bin ich bei der Demontage eines alten Mobilfunkgeräts auf das mir unbekannt IC TCA580 gestoßen. Laut Datenblatt handelt es sich um ein Gyrtator-IC. Ich erinnerte mich sofort an meine Einführung zu Gyrtatoren in Elektor September 1975 (ich habe das Heft noch!) mit dem Titel „Gyrtatoren, warum und wie“ und ein paar Drum-Simulatoren als Anwendungsbeispiel. Ein Gyrtator simuliert eine Induktivität. Wie war das noch mal in der Wechselspannungstheorie? Die Ströme in einer Spule und einem Kondensator sind in Gegenphase. Ein Gyrtator beobachtet den Strom in einem Kondensator an einer Wechselspannungsquelle und fügt der Quelle einen größeren Strom in Gegenphase hinzu. Schwupps – schon haben wir eine Induktivität. Der Originalartikel geht mehr ins Detail und wird auch ein wenig mathematisch – er steht für Sie als kostenloser Download unter [1] bereit.

Der TCA580

Der TCA580 (**Bild 1**) ist ein integrierter Gyrtator von Philips/Signetics, der Mitte der 1970er Jahre in Telekommunikationsschaltungen verwendet wurde. Mit dem IC, einem Kondensator und zwei Widerständen kann man eine virtuelle Induktivität mit einem hohen und stabilen Wert bis zu 1 Megahenry (MH) und sehr hoher Güte ($Q > 1000$) schaffen. Wenn Sie einen weiteren Kondensator hinzufügen, haben Sie schon eine abgestimmte Schaltung, die sich als Notch-Filter oder Frequenzdetektor bis zu 10 kHz einsetzen lässt. Eine Schaltung mit einer echten Induktivität wäre weitaus größer, teurer und ungenauer. Heutzutage würde man eher aktive Filter, PLLs oder digitale Signalprozessoren (DSP) einsetzen, um diese Aufgaben zu erfüllen, aber in den 70er Jahren war ein Gyrtator ein wirtschaftlicher und effektiver Weg, um zu einer solchen Lösung zu gelangen. Informationen über den TCA580 sind



schwer zu bekommen. Es gibt ein dreiseitiges Datenblatt im Netz [2], aber es ist nicht detailliert und echte Anwendungen fehlen auch. In der Schaltung befindet sich ein 50-k Ω -Trimpoti, aber es wird nicht erklärt, wozu es benötigt wird (es dient der Balance und der Minimierung von Verzerrungen). Ich fand auch einen Verweis auf eine Applikation in einem Philips-Fachbuch von 1977 mit dem Titel *TCA580 Gyrtator IC. A Replacement for Wound Inductors in Low Frequency Cir-*

cuits, aber ich konnte den Artikel nirgendwo downloaden oder gar das Buch kaufen. Dennoch bietet die Schaltung im Datenblatt (**Bild 2**) genügend Informationen, um einen Schwingkreis auf einem Steckbrett aufzubauen. Zunächst habe ich für C1 und C2 47-nF-Kondensatoren eingesetzt und die Schaltung in Betrieb setzen können. Dann wollte ich die Schaltung mit einem diskreten LC-Schwingkreis (einer Schalenkern-Induktivität von rund 14,4 H und einem 15-nF-Kondensator) mit etwa der gleichen Resonanzfrequenz von rund 350 Hz vergleichen und habe einen 160-nF-Gyrtator-Kondensator (C2) und den 15-nF-Resonanzkondensator aus dem LC-Schwingkreis für C1 eingesetzt. Ich habe für die Messungen noch einen 100-k Ω -Widerstand zwischen die Quelle und den Gyrtator/Schwingkreis geschaltet.

TCA580 versus echte Spule

Nachdem ich es geschafft habe, beide Schaltungen auf die gleiche Frequenz abzustimmen, konnte ich die Frequenzgänge (TCA580 gelb, LC grün) in **Bild 3** erstellen. Die TCA580-Kurve ist deutlich steiler und enger. Zudem dürfte selbst mit den großen Bauteilen aus der damaligen Zeit die Schaltung wesentlich weniger Platz in Anspruch nehmen als die diskrete. Sie wäre ideal zur Filterung, Erkennung und Erzeugung von Pilottönen. TCA580-ICs sind bei eBay noch erhältlich. Der Preis liegt bei knapp 10 Euro, ein stolzer Preis für ein DIL-IC. Ich habe ein paar dieser ICs aus alten Geräten ausgeschlachtet. Obwohl ich eigentlich nicht weiß, wozu ich sie heute einsetzen könnte, war es doch ganz nett, mit diesen High-Tech-ICs der 70er-Jahre zu experimentieren. ◀

(150024)

Weblinks

[1] www.elektor-magazine.de/150024

[2] www.datasheetarchive.com/dlmain/Databooks-4/Book618-6582.pdf

Einfacher Transistortester

Mit LED oder Piezo-Buzzer

Von **Hans-Norbert Gerbig** (D)

Dieser einfache Transistortester kann die Funktion von NPN/PNP-Transistoren und JFET/(V)MOS-FETs prüfen sowie die Anschlussfolge bestimmen.

Die drei Anschlüsse eines Transistors oder FETs führen zu sechs Kombinationsmöglichkeiten für die Beschaltung, von denen nur eine richtig ist. Diese kleine Schaltung erlaubt eine rasche und eindeutige Anschlussbestimmung mit gleichzeitiger Funktionsprüfung. Sie enthält einen Transistor, der mit dem Prüfling einen Multivibrator bildet. Zum Anschluss des zu testenden Transistors stehen fünf nebeneinander liegende Buchsen zur Verfügung:

E/S – B/G – C/D – E/S – B/G

Folgende Anschlussmöglichkeiten lassen sich damit testen:

- Bipolare Transistoren: EBC / BCE / CEB und rückwärts: BEC / ECB / CBE.
- Unipolare Transistoren (FETs): SGD / GDS / DSG und zurück: GSD / SDG / DGS.

Bei richtiger Anschlussfolge, denn nur so kann der Multivibrator schwingen, blinkt die High-Efficiency-LED in **Bild 1**. Werden die Anschlüsse E und C vertauscht, kann es sein, dass die LED auch blinkt, aber etwas schneller. Denn manche bipolaren Transistoren funktionieren auch - mit Einschränkung - wenn die Anschlüsse von Emitter und Kollektor vertauscht werden. Bei Junction-FETs (Sperrschicht-FETs) mit symmetrischem Aufbau wird nur das Gate mit Sicherheit angezeigt, die Anschlüsse S und D sind bei diesen Typen austauschbar.

Der Lastwiderstand des Prüflings ist als Spannungsteiler ausgeführt (halbe Betriebsspannung, R3/R4). Deshalb kann mit einem einfachen Umschalter (S1) von N(PN) nach P(NP) gewechselt werden.

Wichtig beim Testen: Nur die blinkende LED gibt eine eindeutige Auskunft. Kein Leuchten oder Dauerleuchten bedeutet eine falsche Anschlussfolge oder eine Fehlfunktion des Test-Transistors (kaputt!). Es kann auch sein, dass der Prüfling gar kein Transistor ist, aber z.B. ein Spannungsregler, Thyristor, Triac etc. In der Version 2 der Schaltung (**Bild 2**)

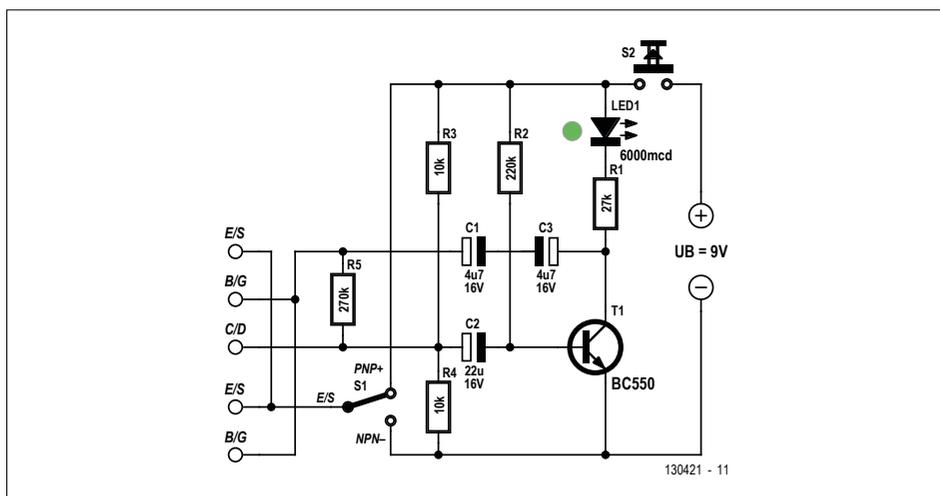


Bild 1. Version mit LED.

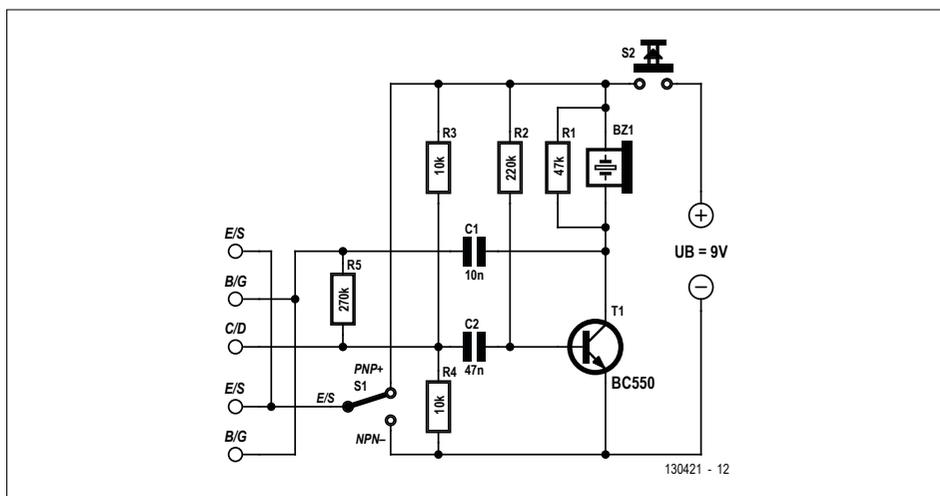


Bild 2. Version mit Piezo-Buzzer.

ist die LED durch einen Piezo-Summer ersetzt. Um einen hörbaren Sound zu generieren, haben die frequenzbestimmenden Kondensatoren viel kleinere Werte als in Bild 1. Ein tiefer Ton bedeutet, dass der Prüfling funktioniert, kein Ton bedeutet entweder eine falsche Anschlussfolge oder eine Fehlfunktion des Test-Transistors.

Mit dem Taster kann man den Transistor-Tester kurz aktivieren, wenn ein Prüfling angeschlossen ist. Die Versorgungsspannung wird durch eine 9-V-Blockbatterie geliefert.

Die Schaltung kann einfach auf einem Stück Experimentierplatine aufgebaut werden. ◀

(130421)

Embedded World 2015

902 Aussteller markieren für die jährlich in Nürnberg stattfindende Messe einen neuen Rekord. Viele Hersteller präsentierten wieder neue Produkte aus den Bereichen Mikrocontroller, Entwicklungs-Tools und Software; große Klammer war in diesem Jahr natürlich das Internet der Dinge (IoT). Hier ein kleiner Überblick.

Von Viacheslav Gromov (D)

Schon nach den ersten 100 Metern „Spaziergang“ über die Messe war das Thema „Internet of Things“ nicht zu übersehen. Neue Ideen und Entwicklungen auf diesem Gebiet gab es fast an jedem Stand zu sehen – sowohl bei den großen und sehr bekannten als auch bei den kleinen Unternehmen, die teilweise sogar auf das Internet der Dinge spezialisiert sind. Nach den nächsten 100 Metern bekommt man dann ein Gefühl für zwei wichtige Trends: Bluetooth Low Energy (auch oft nur BLE oder Bluetooth 4.x genannt) und 32-bit-Mikrocontroller mit dem neuen ARM Cortex-M7-Kern. Es gab kaum einen großen Halbleiterhersteller im Embedded-Bereich, der nicht zumindest für eine dieser beiden Technologien ein neues Produkt entwickelt hatte.

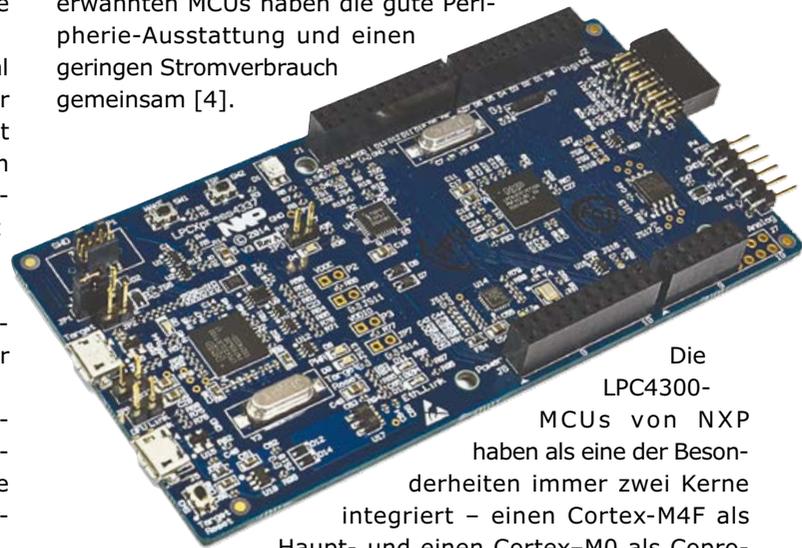
Bluetooth Low Energy zielt, wie schon der Name vermuten lässt, auf einen sehr geringen Stromverbrauch, wobei deswegen aber Abstriche sowohl bei der Reichweite als auch bei der Übertragungsgeschwindigkeit gemacht werden müssen [1]. Der ARM Cortex-M7-Kern ist ebenfalls auf geringen Stromverbrauch getrimmt und stellt nun den leistungsfähigsten Cortex-M-Kern dar. Er ist deswegen auch besser ausgestattet als seine Brüder, zum Beispiel besitzt er mehrere Cache-Speicher. Innerhalb der Cortex-M-Familie ist er aber abwärtskompatibel [2].

Der Drahtlos-Spezialist u-blox zeigte neue Module, die gleich mehrere Kurzsteckenfunk-Technologien beherrschen. Die Module der EMMY-W1-Serie können WiFi, Bluetooth und NFC und sind überwiegend für Automotive-Anwendungen gedacht. Aber das ist noch nicht alles: Bluetooth wird sowohl mit dem klassischen als auch mit dem BLE-Standard unterstützt und die WiFi-Kommunikation ist sowohl auf

2,4 GHz als auch 5 GHz möglich. Der Stromverbrauch von unter einem Watt ist ein sehr guter Wert [3].

Atmel zeigte auch viele Neuheiten, zum Beispiel den SAM L21. Das ist ein leistungsfähiger Ultra-Low-Power-Mikrocontroller, der dem SAM D20 aus dem Elektor

ARM-Kurs ähnlich ist, allerdings nur 200 nA im tiefsten Schlaf verbraucht. Für den neuen Controller gibt es ein Xplained-Pro-Board. Mit einem Cortex-M7-Kern ist die neue Controller-Familie SAM V (V70 und V71), die Familie SAM E (E70) sowie die SAM S-Familie (S70) ausgestattet. Für den SAM V70 und den V71 stehen „Xplained Ultra Boards“ (siehe Foto) zur Verfügung. Alle erwähnten MCUs haben die gute Peripherie-Ausstattung und einen geringen Stromverbrauch gemeinsam [4].

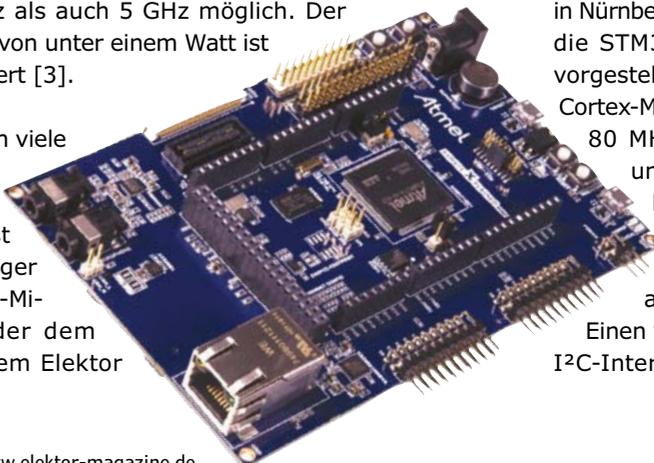
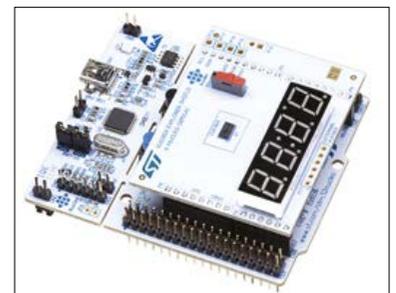


Die LPC4300-MCUs von NXP haben als eine der Besonderheiten immer zwei Kerne integriert – einen Cortex-M4F als Haupt- und einen Cortex-M0 als Coprozessor, die sich beide mit bis zu 204 MHz takten lassen. Die Kerne können unabhängig voneinander aktiv sein – das sorgt für viel Flexibilität. Eine besondere Sub-Familie sind die 43Sxx-Controller, diese bieten diverse Möglichkeiten zur Verschlüsselung. Das abgebildete LPCXpresso43S37-Board ist dank vieler Erweiterungsmöglichkeiten hier ein guter Einstieg [5].

STMicroelectronics setzt ebenfalls auf das Ultra-low-power-Konzept. Der Halbleiterhersteller hat in Nürnberg unter anderem die STM32L4-MCU-Serie vorgestellt, die mit einem Cortex-M4-Kern mit bis zu 80 MHz Taktfrequenz

und reichlich Peripherie ausgestattet ist. Die Stromverbrauchswerte mit zum Beispiel 30 nA (!) im Shutdown-Modus (ohne den RTC) sprechen für sich. Dazu wird es zahlreiche Entwicklungs-Boards geben, unter anderem ein Nucleo-Board.

Einen winzigen Entfernung- und Umgebungslichtsensor mit I²C-Interface (VL6180X) zeigte STMicroelectronics auch auf





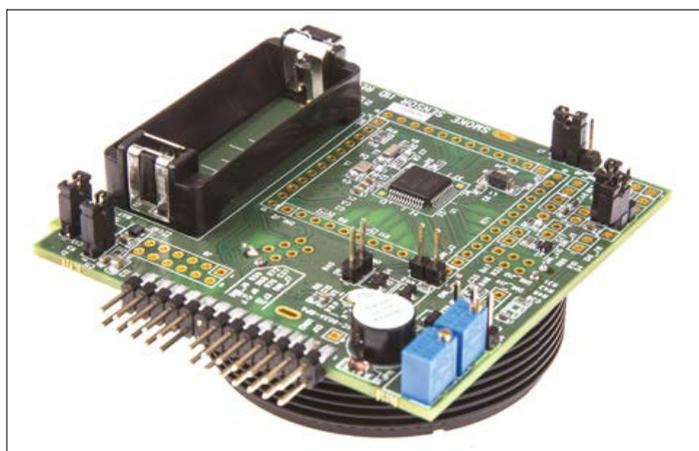
dem Stand. Die Entfernung (bis über 10 cm) misst er dabei mit Hilfe eines IR-Signals, welches von einem davor befindlichen Gegenstand reflektiert wird. Das Umgebungslicht kann der Sensor in Lux messen. Für diesen Sensor gibt es eine Erweiterungsplatine für ST-Nucleo-Boards (Foto) und ein entsprechendes Visualisierungs-Tool für erste Versuche [6].

Renesas hatte eine neue Sub-Familie der 16-bit-RL78-Mikrocontrollerreihe namens RL78/I1D nach Nürnberg mitgebracht, die vor allem auf das Anwendungsgebiet Sensorik bzw. Detektoren zielt. Die Controller sind hierfür mit viel Peripherie ausgestattet und natürlich auch beim Stromverbrauch sparsam. Es lassen sich zum Beispiel Rauchmelder, Glasbruchmelder, CO₂-Melder und vieles mehr realisieren. Damit es an Sensoren

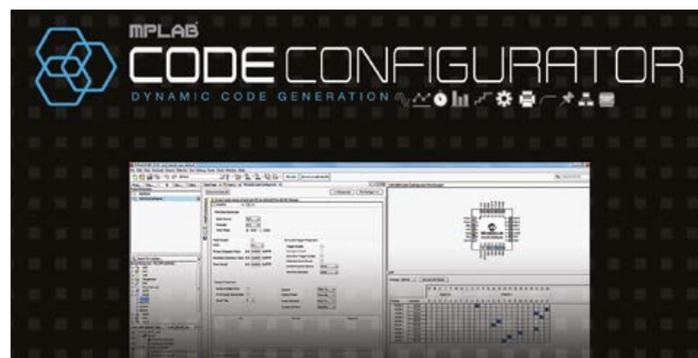
Natürlich wurden auf der Embedded World auch viele neue und überarbeitete Software-Tools vorgestellt. Der Chip-Hersteller Microchip zeigte ein neues Code Configuration Plug-In für die MPLAB X IDE. Dieser Code-Generator erstellt den Initialisierungs- sowie Treiber-Code für die vom Nutzer ausgewählten Peripherieelemente mit den benötigten Einstellungen automatisch. Das Plug-In unterstützt bisher mehr als 50 16-bit-PIC-MCUs sowie mehr als 140 8-bit-PIC-MCUs [8].

Mit dem MM7150 wurde zudem ein neues Bewegungssensor-Modul auf dem Stand gezeigt. Es kombiniert einen 3-achsigen Beschleunigungsmesser, einen 3-achsigen Magnetfeldsensor und ein 3-achsiges Gyroskop von Bosch. Der verbaute SSC7150 „Motion Coprozessor“ sorgt für eine Filterung und Zusammenführung der Rohdaten. Positions- und Bewegungsdaten kann ein Host-Controller dann über I2C abrufen.

(150152)



bzw. Detektoren für den Einstieg und für die Entwicklungsphase nicht mangelt, hat Renesas viele nützliche Erweiterungsplatinen für das (später erscheinende) Entwicklungs-Board entwickelt. Ein Board mit Rauch-Sensor ist auf dem Foto zu sehen [7].



Weblinks

- [1] www.bluetooth.com/Pages/Bluetooth-Smart.aspx
- [2] www.arm.com/products/processors/cortex-m/cortex-m7-processor.php
- [3] www.u-blox.com/en/bluetooth-wifi-multiradio-modules/multiradio-wi-fi-bluetooth-modules/emmy-w1.html
- [4] www.atmel.com/about/news/release.aspx?reference=tcm:26-65219
- [5] www.nxp.com/products/microcontrollers/product_series/lpc4300/series/LPC43SXX.html
- [6] www.st.com/web/catalog/tools/FM116/SC627/PF260896?icmp=pf260896_pron_p3609p_sep2014
- [7] www.renesas.com/products/mpumcu/rl78/rl78i1x/rl78i1d/index.jsp
- [8] www.microchip.com/pagehandler/en_us/devtools/code_configurator/home.html

Preiswerter 0,5-A-/33-V-LED-Hochleistungstreiber

Von **Valentin Kulikov** (FuturoLighting)

Wir beschreiben die Bauteilauswahl und einige Entwicklungskriterien für den Bau eines einfachen Konstantstrom-Treibermoduls mit schnellem PWM-Eingang. Geeignet ist dieser für die Ansteuerung von Medium- und High-Power-LEDs mit einem einstellbaren Ausgangsstrom in Schritten von 0,1...0,5 A, bei einer Spannung von 8...33 V.

Die Schaltung in **Bild 1** zeigt das LED-Treibermodul mit einem von Taiwan Semiconductor hergestellten Abwärtsregler-IC TS19376 im SOT89-5-Gehäuse. Der TS19376 verfügt über eine Hysterese-Regelung, die eine Effizienz von über 90 % ermöglicht, ohne dass eine Kompensation nötig wäre. Der Ausgangsstrom wird durch die parallel geschalteten Widerstände R1, R2, R3 mit einem Verhältnis von 0,13 Ω /A eingestellt. Die Grundsätze der Hysterese-Regelung werden in [1] zusammengefasst: Der interne Schalter des TS19376-Treibers verbindet die Eingangsspannung über die Induktivität L1 mit der Last. Der Strom durch die Induktivität steigt linear an und wird als Spannungsabfall über R1||R2||R3 überwacht. Bei einer Hysterese von beispielsweise 15 % (19,5 mV) bewirkt der Strom diese Spannung:

$$130 + U_{\text{csn_hys}} = 149,5 \text{ [mV]}$$

Der integrierte Schalter schaltet ab und der Strom durch die Spule und D1 nimmt linear ab bis auf:

$$130 - U_{\text{csn_hys}} = 110,5 \text{ [mV]}$$

Dann schließt der Schalter wieder und der Prozess wiederholt sich zyklisch, so wie in **Bild 2** dargestellt. Die Schaltfrequenz wird vom Ausgangsstrom (I_{LED}), der Versorgungsspannung (U_{cc}), der Ausgangsspannung und dem Wert von L1 bestimmt.

Dimmen mit PWM

Der durchschnittliche LED-Strom wird von einem PWM-Signal gesteuert. PWM ist populär und kann leicht von einem Mikrocontroller oder auch „diskret“ erzeugt werden, etwa mit einem Timer 555 (**Bild 3**). Das PWM-Signal wird an den Eingang des PWM-Moduls gelegt, wobei wie bei CMOS üblich <0,3 V als low und >2 V als high angesehen werden. Der TS19376 akzeptiert relativ hohe PWM-Frequenzen, so dass auch bei schnellen

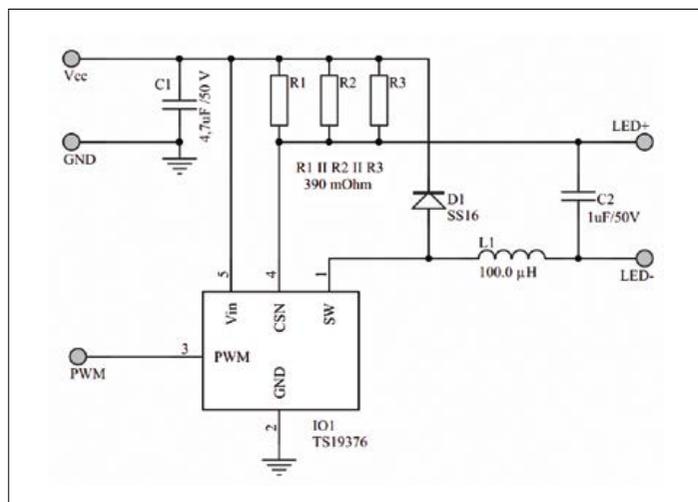


Bild 1. Schaltung des LED-Treibers.

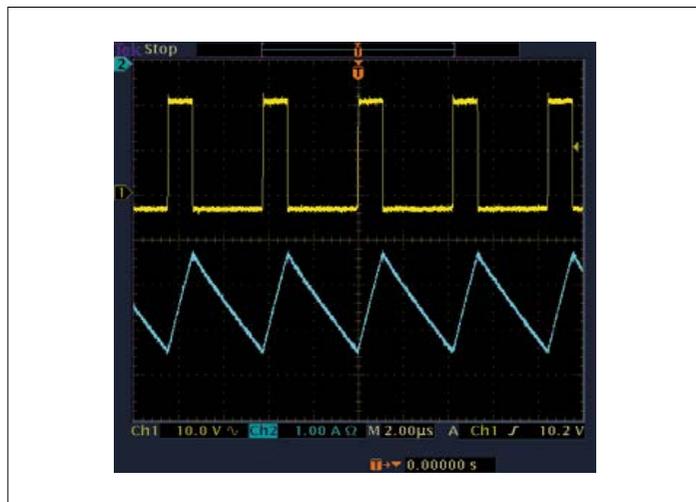


Bild 2. Wellenformen von Strom und Spannung am Schaltknoten SW (Oszilloskop-Masse an U_{cc}).

PWM-Signalen mit mehr als 8-bit-Auflösung keine Probleme entstehen. Da der PWM-Eingang mit einem Pullup-Widerstand ausgestattet ist, erhalten die LEDs ohne angelegtes PWM-Signal den maximalen Strom. Die empfohlene PWM-Frequenz sollte höher als 100 Hz sein, um sichtbares Flimmern zu vermeiden.

In der Praxis

Der TS19376 benötigt eine gewöhnliche Kupferfläche auf der Rückseite der Leiterplatte als Kühlkörper und ist thermisch via Durchkontaktierungen mit der Oberseite verbunden. Ein Low-ESR-Eingangskondensator ist erforderlich, um Stromspitzen während des Schaltens zu unterdrücken. Der empfohlene Wert für C1 beträgt 4,7...100 μF mit X7R, X5R oder besser als dielektrischem Material. C1 sollte so nah wie möglich an den Versorgungsanschlüssen des ICs angebracht werden.

Der optimale Bereich für die Induktivität L1 ist 47...120 μH . Eine geringe Induktivität ist für höhere Ströme geeignet, höhere Induktivitäten für niedrigere Ströme, wo die Schaltverzögerung eliminiert ist. Die Platzierung der anderen Bauteile sollte dem Prinzip der „kleinsten Schaltschleife“ folgen, um eine minimale EMI zu erhalten. Der Beginn der Spulenwicklung sollte mit dem Schaltknoten verbunden sein (SW-Pad).

D1 soll niedrige Sättigungsströme bei maximaler Betriebstemperatur und geringem t_{rr} gewährleisten. Die Durchlassspannung von D1 hat Einfluss auf die Effizienz. Je niedriger U_f ist, desto höher die Effizienz und geringer die Verlustleistung.

Der maximale Vorwärtsdiodenstrom sollte mindestens 30 % über I_{LED} liegen. Hier wird eine SS16 (1 A/60 V) von Taiwan Semiconductor eingesetzt. Der Kondensator C2 reduziert die Welligkeit des Ausgangsstroms. Höhere Kapazitäten führen zu geringerer Welligkeit (wichtig bei niedrigen PWM-Frequenzen). Der TS19376 besitzt eine thermische Abschaltung. Sobald die Temperatur 150°C erreicht, schaltet der Treiber ab, bis die Temperatur wieder auf 115°C gesunken ist. Dieser Schutz verhindert ein Verbrennen der Modulplatine. Das Treibermodul kann mit zweiseitigem wärmeleitendem Band (zum Beispiel Bergquist Bond Ply) am Kühlkörper befestigt werden. Es ist möglich, das Treiber-Modul mit EMI-Filter und Verpolschutz (etwa ein p-MOS-Schalter) auszustatten, aber das hängt von der konkreten Anwendung ab. Das Treibermodul ist auf einer doppelseitigen, 16x16 mm² kleinen FR4-Platine mit einer Dicke von 1 mm aufgebaut.

Fazit

Der hier beschriebene LED-Treiber macht sich in zahlreichen Anwendungen nützlich, in denen eine kräftige einstellbare Kon-

Technische Daten:

- Topologie: Abwärtsregler
- Regelung: Hysterese
- Eingangsspannung: 8...33 VDC
- Ausgangsstrom: 100...500 mA
- Schaltfrequenz: 1 MHz max.
- Stromeinstellung: 0,13 Ω/A
- Maße: 16 x 16 x 5,5 mm
- Gewicht: 1,6 g
- Thermische Abschaltung
- Überstromschutz
- Dimmen: PWM bis 20 kHz

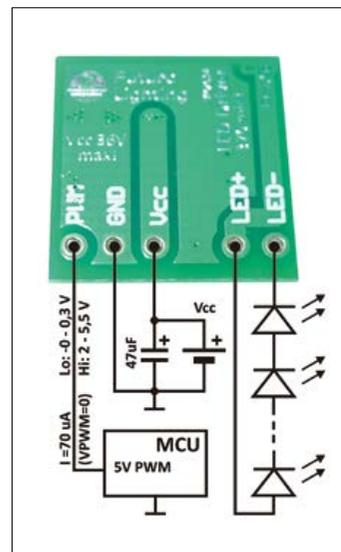


Bild 3. Anschluss des LED-Treibermoduls.

stantstromquelle gefragt ist, wie beim Betrieb von Medium- und High-Power-LEDs oder beim Laden von Akkus. Die mögliche Anzahl der LEDs in der Kette wird von der Betriebsspannung (U_{cc}) bestimmt. Wie in den Diagrammen in **Bild 4** zu sehen ist, führt es zu einem höheren Wirkungsgrad, wenn die Spannung (U_{LED}) der LED-Kette in der Nähe von U_{cc} liegt. So sind drei LEDs in Reihe (mit U_{LEDF} von etwa 3 V) eine gute Wahl für eine Betriebsspannung von $U_{cc} = 12 \text{ V}$. Alle Messungen wurden automatisiert bei Raumtemperatur vorgenommen.

Das LED-Treiber-Modul mit wählbarem Ausgangsstrom von 0,1...0,5 A ist erhältlich bei FuturoLighting [2]. Der TS19376 und die Diode SS16 sind in Stückzahlen bei Microdis Electronics [3] erhältlich, einem autorisierten Distributor von Taiwan Semiconductor.

Hier möchte ich Herrn Bilik von Würth Elektronik und Herrn Reguli von Microdis Electronics für die tolle Unterstützung bei diesem Projekt danken. ◀

(150048)

Weblinks

- [1] www.taiwansemi.com/en/
- [2] www.fulit.eu/store
- [3] www.microdis.net

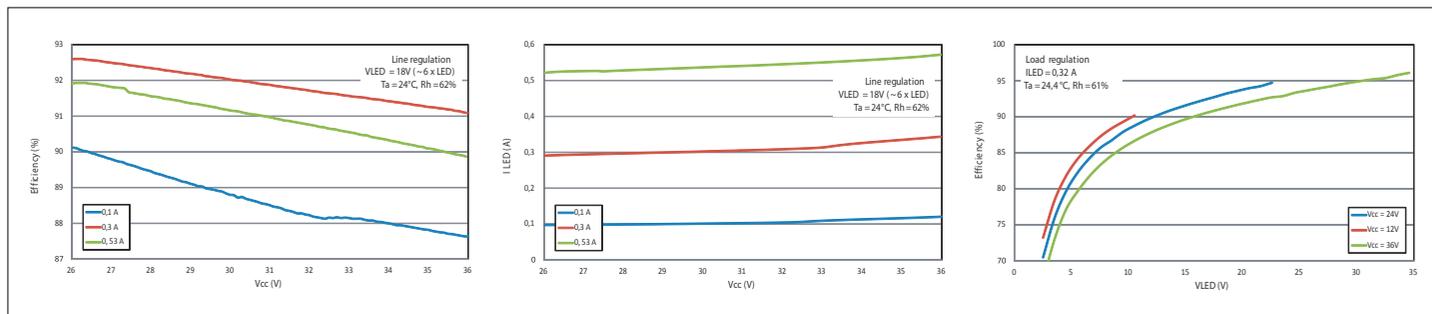


Bild 4. Der LED-Treiber unter verschiedenen Bedingungen auf dem Prüfstand.

GERADE WENN'S MAL HEISS HERGEHT.

LEITERPLATTEN AUS ALUMINIUM ONLINE BESTELLEN.



LEITON
RECHNEN SIE MIT BESTEM SERVICE

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

www.leiton.de

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

Schaeffer
AG



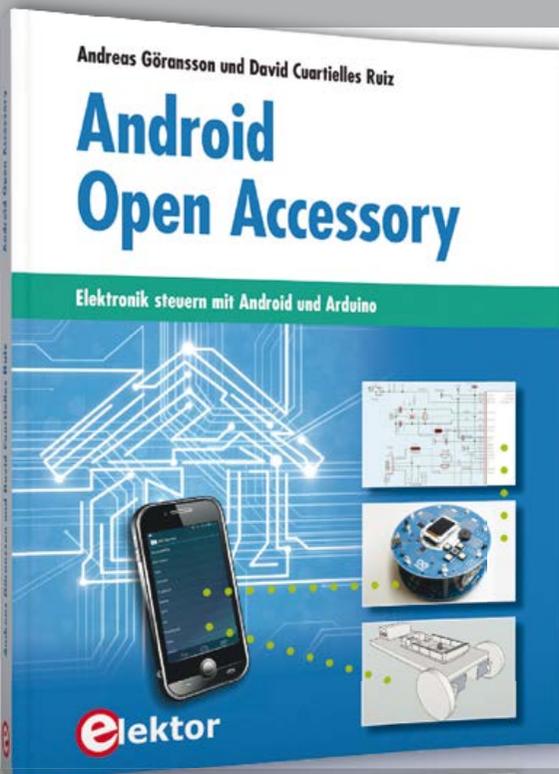
Gratis Frontplatten Designer

SIE DESIGNEN – WIR FERTIGEN

Frontplatten in Profiqualität

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

www.schaeffer-ag.de



Elektronik steuern mit Android und Arduino

Android Open Accessory

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

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

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



Mitgliederpreis: € 42,00 • CHF 44.95

www.elektor.de/android-open-accessory

LEARN



DESIGN

SHARE

Willkommen bei DESIGN

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



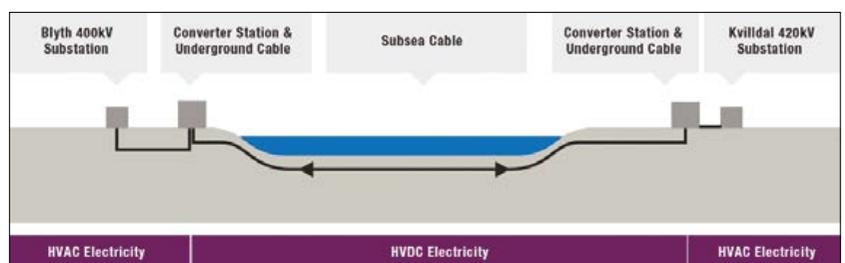
Das Paradoxon der Elektronikentwicklung

Einer der schwierigsten Aspekte beim Entwurf elektronischer Schaltungen ist die Auswahl der Bauteile, um ein bestimmtes Ziel zu erreichen. War die Wahl zu Nikola Teslas und Thomas Edisons Zeiten auf Widerstände, Kondensatoren und Induktivitäten (und später Dioden) beschränkt, so haben wir heute bei Zehntausenden von Bauteilen und tagtäglichen Neuerungen buchstäblich die Qual der Wahl. Für fast jede denkbare Funktion gibt es elektronische Bauteile und dies von verschiedenen Herstellern in einer Vielzahl von Gehäusen und Klassen, was die Wahl noch schwieriger macht. Selbst wenn Sie das ideale Bauteil für Ihre Anwendung ausgraben, das nur in einer Version von einem Hersteller existiert, wird Ihr Chef Sie anweisen, es nicht zu benutzen, weil es zu riskant wäre, ein Bauteil zu verwenden, das nur von einer Quelle angeboten wird. Oder Ihr Arbeitgeber möchte nicht, dass Sie Ihrem Inventar ein neues Bauteil zufügen. Also, nutzen Sie, was Sie finden können, was Sie beschaffen können, was Ihnen erlaubt wird zu nutzen, was <hier eigene Bedingungen eintragen>. Dann, wenn Sie fast fertig sind und die Platine schon bestellt wurde, entdecken Sie, dass

es viel einfacher, billiger und schneller gewesen wäre, den AW50M3 von Acme Corp. zu verwenden. Und was jetzt? Wenn Sie Glück haben, ist es zu spät, um etwas zu ändern. Wenn nicht, dürfen Sie Ihren Entwurf ändern. Wie kommen Sie raus aus diesem Dilemma? Gar nicht! Planung ist wie die Besteigung einer M.C.-Escher-Treppe: Man kommt niemals an.

Darf ich mein Land einstöpseln?

Im Jahr 2020 soll das weltweit längste Stromkabel fertiggestellt werden. Über 700 Kilometer soll die NSN-Verbundleitung von Kvittdal in Norwegen nach Blyth in Großbritannien reichen. Das 515-kVDC-Hochspannungs-Seekabel, das 1...3 m tief im Meeresboden vergraben wird, soll satte 1,4 GW für etwa 700.000 Durchschnittshaushalte transportieren. Die geschätzten Kosten des Projekts liegen bei 1,5...2,0 Milliarden Euro, was in etwa 1,5 €/W entspricht. Solche Leitungen, die die Stromnetze zweier Länder verbinden, werden mittlerweile weltweit geplant. So ist auch eine 1.000 km lange Leitung zwischen Großbritannien und Island angedacht. Damit wäre Großbritannien und das restliche Europa an Islands reiche Quellen von Erdwärme, Wasserkraft und Windenergie angeschlossen.



Willkommen im Elektor-Labor

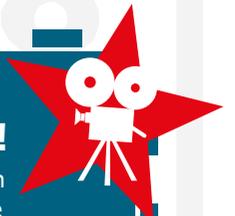
Das Elektor-Labor ist der Ort, wo Projekte - groß, klein, analog, digital, new- oder old-school - skizziert, gebaut, diskutiert, getestet und für den Nachbau und Ihre Anwendung optimiert werden.

Unsere Geschichte

Die Ursprünge des Elektor-Labors gehen zurück auf die frühen 1970er-Jahre. Löten und Schreiben war damals noch ein Ein-Mann-Schreibtisch-Job. Über die Jahre waren Mitarbeiter des Labors nicht nur Zeugen der Ankunft der Transistoren, der ICs, der Mikrocontroller und der SMDs, sondern sie setzten diese Bauteile auch sofort in der Praxis ein.

Unser Angebot: Werden Sie berühmt!

Die meisten Ingenieure und angehenden Autoren sind einfach zu bescheiden. Sie sehen nicht den Reiz und die Schönheit einer auf einen Bierdeckel gekritzelt Idee, die später zu Hause ausgearbeitet wird. Lassen Sie sich vom Elektor-Labor helfen, ihr Projekt zu perfektionieren, überlassen Sie den Redakteuren Text und Grafik und ernten die Früchte Ihrer Arbeit in Form Ihres Namens, gedruckt in der Design-Rubrik von Elektor. Sicher, wir freuen uns, mit Ihnen über eine Vergütung zu verhandeln, aber Ruhm und Ehre im Land der Elektronik erlangen Sie vor allem, wenn Ihr Name der langen Liste der erfolgreichen E-Autoren hinzugefügt wird. Unsere „Wie werde ich berühmt“-Formel gilt auch für Buchautoren, Blogger und Video-Regisseure. Ob Jugendlischer oder Student: Eine Veröffentlichung in Elektor ist DER Booster für einen tollen Job!



Unsere Einrichtungen

Wir sind prächtig in drei geräumigen Zimmern im Elektor-Castle untergebracht, wo wir vergeblich versuchen, unsere Computer-Schreibtische frei von Lötspitzern und Prototyp-Verkabelungen zu halten. Wir haben Wasser, Netzspannung und Kaffee in der Nähe. Platinenfräsen, Prototypenmontage, SMD-Nachbearbeitung, Audio-Tests, Pizzabacken und mechanische Arbeiten sind in die umgebauten Keller verlagert.

Elektor-Labs.com – unsere Website

Verwenden Sie diesen hoch-bidirektionalen Online-Port, um Ihr Projekt vorzustellen. Folgen Sie den Aktivitäten anderer Leser. Erfahren Sie mehr über andere coole Projekte, erstellen Sie Beiträge und tauschen Sie sich mit anderen Enthusiasten aus. Bemerkenswert aufregende Projekte eignen sich für das Post-Engineering im realen Elektor-Labor und vielleicht zur glorreichen Veröffentlichung in der Zeitschrift Elektor. Stellen Sie Ihr Projekt auf unserer Website vor und Sie haben die Chance, es auch in gedruckter Form zu sehen.



354

Project Proposals

42

Projects in Progress

154

Projects Finished

550

Projects Total

Unsere Produkte

Unsere Produkte sind in der Elektor-Zeitschrift, auf Elektor-Labs.com und im Elektor-Shop zu sehen. Das Sortiment umfasst Notizen für die Redaktion, Fotos von Prototypen, Platinen inklusive SMD-Vorbestückung, Layout-Dateien, Projekt-Software, programmierte Bauteile, Semi-Kits, Tools, Module, Videos und weitere Infos.

Unsere Webinare

Die geschätzteren unserer Labor-Ingenieure machen nicht beim Test von Prototypen halt, sie sind froh, Probleme der, Einblicke in die, aktuelle Informationen über die und Entwicklungstricks aus der Elektronik vor der Live-Kamera auf Elektor-TV zu diskutieren. Die Webinare aus dem Elektor-Labor sind kostenlos zu erreichen und extrem unterhaltsam. Sie werden in Elektor.POST angekündigt und live aus dem Elektor-Castle in den Niederlanden übertragen. Machen Sie mit!

Unsere Experten und Entwickler

Neben den erfahrenen Support-Mitarbeitern und qualifizierten Ingenieuren mit einer Gesamtberufserfahrung von etwa 200 Jahren hat das Labor Zugang zu einem riesigen Netzwerk von Experten für eine Beratung, kritische Ratschläge und Unterstützung bei speziellen Aufgaben.

Unser Maßstab

Alle Projekte und Produkte, die unser Labor verlassen, erfüllen einen hohen technischen Standard. In der Praxis werden Prototypen von Projekten, die in der Zeitschrift mit Labor-Projekt gekennzeichnet sind, einer strengen Prüfung mit unseren zertifizierten, kalibrierten Testgeräten unterworfen. Stücklisten und Schaltpläne müssen perfekt passen, Kits sind auf Vollständigkeit geprüft. Wir sind ROHS-kompatibel, bleifrei und entsprechen in unserem Labor den Normen für elektrische Sicherheit. Sollten trotz aller Sorgfalt technische Fehler erkannt werden, publizieren wir dies unmittelbar.

UART/RS232-Datenlogger

Serielle Datenströme überwachen

Von **Thomas Schlott**, **Francisco Ramirez** und **Dr. Thomas Scherer**

Der UART kommt (immer noch) gern zum Einsatz, wenn Geräte Daten austauschen sollen. Die Schnittstelle ist sehr einfach zu verwenden, dazu gibt es eine Vielzahl von Konverter-Chips. Wer bei Entwicklung und Test aber feststellt, dass die Kommunikation noch hakt, muss sich auf eine knifflige Fehlersuche einstellen. Unser UART-Datenlogger macht das Leben leichter – er registriert Bytes in beiden Richtungen millisekundengenau. Dank flexibler Anschlussmöglichkeiten und einer galvanischen Trennung lässt er sich leicht in serielle Verbindungen mit TTL- oder RS232-Pegeln einschleifen.

Bei Fehlfunktionen in elektronischen Systemen kann man sich einen Wolf suchen, wenn die Ursachen selten oder

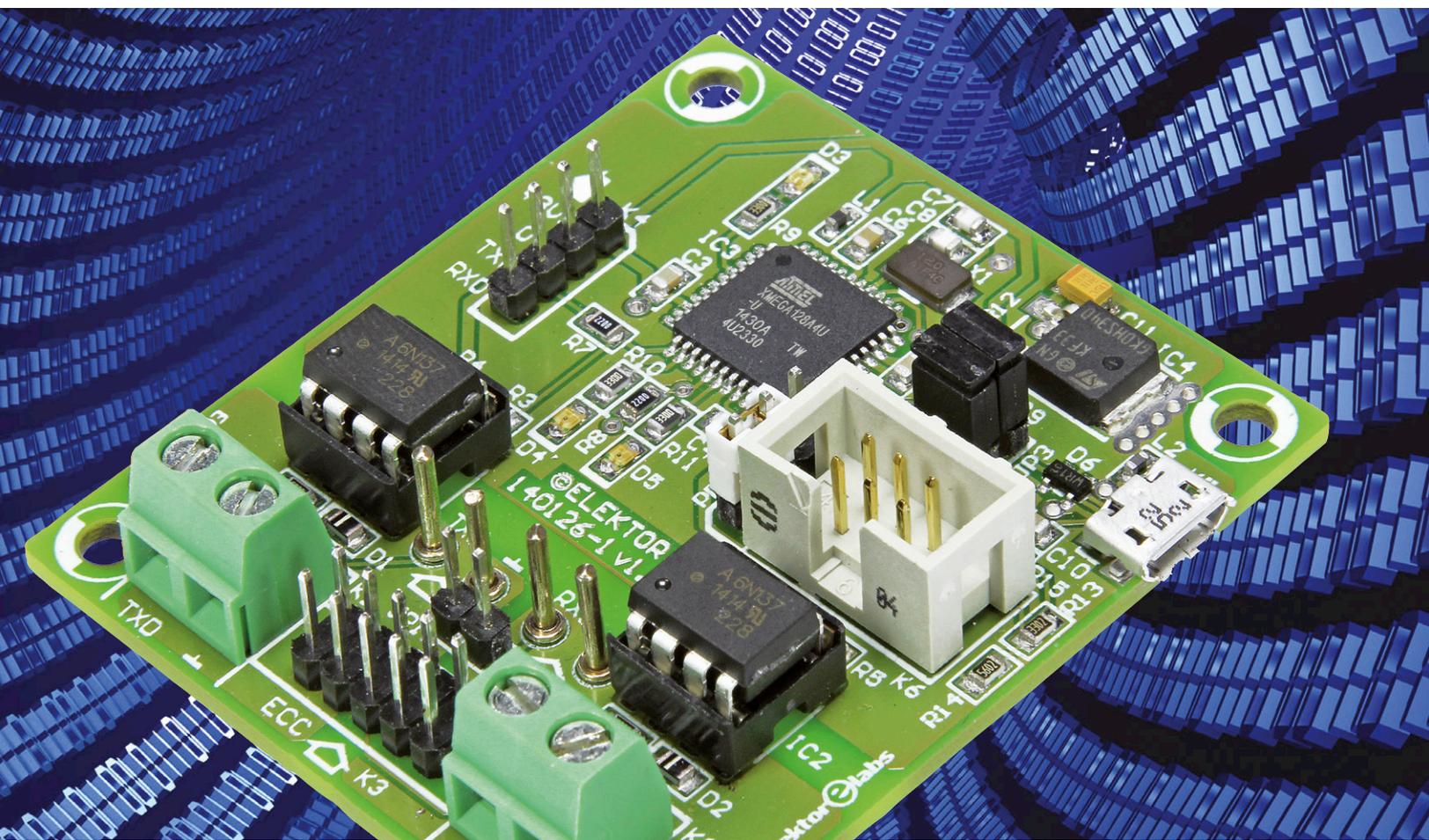
nicht reproduzierbar auftreten. Und wenn sie auftreten, dann sind die „schuldigen“ Daten längst schon durch die Leitung

gerauscht und haben sich im digitalen Nirwana verflüchtigt. Um im Nachhinein zu klären, was los war, muss man die verfügbaren Daten mitschreiben und sie dabei tunlichst mit einem Zeitstempel versehen.

Technische Daten

- Zwei optisch isolierte serielle Eingänge für bidirektionale Verbindungen
- Masse der Eingänge trennbar
- Datenübertragung an PC via USB
- Direkte Stromversorgung über USB
- User-Interface über Kommandos per Terminal
- Baudraten bis 230 kBd
- Speichert bis zu 900 Datensätze
- Datenlogging bis 2:20 (h:mm)
- Zeitliche Auflösung 1 ms

Das ist aber noch nicht alles: Sind zwei Geräte per serieller Schnittstelle miteinander verbunden, kommt es auf die Bytes in beiden Richtungen an. Ein serieller Datenlogger braucht daher in der Praxis gleich zwei Datenkanäle, damit er solch eine bidirektionale serielle Verbindung (full duplex) komplett überwachen kann. Und da das reine Aufzeichnen ja



nicht genügt, sondern man anschließend die Daten auch inspizieren will, braucht es auch noch eine Verbindungsmöglichkeit zu einem PC – am besten modern via USB. Damit sind die wesentlichen Anforderungen an einen seriellen Datenlogger gesammelt. Jetzt muss man diese nur noch in Hardware gießen, mit Firmware beseelen und schon hat man ein preiswertes und ungemein nützliches kleines Hilfsmittel zur Hand.

Hardware & Ressourcen

Keine Frage: Die aufgezählten Funktionen schreien geradezu nach einem Mikrocontroller, der Daten mit zwei seriellen Eingängen einliest, sie mit einem Zeitstempel versieht, dann intern in einem Puffer ablegt und an einen PC ausgibt. Die Anforderungen an die Hardware sind also nicht so groß: Benötigt werden zwei U(S)ARTs, ein Timer, USB, ein genauer Takt und noch ein paar GPIOs. Ein moderner AVR-Controller mit „eingebautem USB“ ist geradezu prädestiniert für diesen Zweck – ganz besonders, wenn er über genug internen Speicher verfügt. Die Wahl fiel daher zielsicher

auf den Typ ATxmega128a4u [1], den es im noch gut manuell lötbaren SMD-Gehäuse mit 44 Pins gibt.

Gemessen am Können ist die Schaltung des kompletten Loggers von **Bild 1** recht bescheiden. Damit es keinen Stress mit unterschiedlichen Massepotentialen gibt, sind die beiden seriellen Eingänge individuell über besonders schnelle „digitale“ Optokoppler des Typs 6N137 von der Elektronik des Loggers samt angeschlossener PC isoliert. In Tests haben sich diese Optokoppler bis hin zu Baudraten von 230 kBd als stabil erwiesen. Die erhöhten Datenraten eines FTDI-Kabels von 460 oder gar 920 kBd werden von der Firmware nicht unterstützt. Die beiden FETs T1 und T2 erlauben (als Stromquellen geschaltet) Eingangsspiegel von bis zu 30 V bei einem Strom von etwa 3 mA. Die Eingangssignale RXD und TXD kann man über fliegende Drähte von der zu untersuchenden Verbindung oder dem zu untersuchenden Interface an die Schraubklemmen K1 und K2 führen. K3 dient dazu, eine UART-Verbindung zu untersuchen,

die über einen *Embedded Communication Connector* (ECC) läuft. Beispielsweise also zwischen dem Elektor Extension-Shield und einem der bereits in Elektor vorgestellten Konverter (RS485, 433-MHz-Funk, USB). Mehr dazu im Kasten.

Ein gesteckter JP1 sorgt für eine Verbindung der Massen beider Eingänge. Will man keine bidirektionale, sondern stattdessen zwei unterschiedliche unidirektionale serielle Schnittstellen überwachen, kann man JP1 abziehen; ansonsten sollte JP1 gesteckt werden. Die seriellen Daten können an K4 potentialgetrennt und signalgeformt abgegriffen werden, zum Beispiel mit einem Oszilloskop-Tastkopf.

Die seriellen Signale werden vom Controller IC3 über seine Pins 12 und 16 eingelesen. IC3 verfügt über genug Pins, um mit D3...D5 gleich drei LEDs zur Anzeige der Betriebszustände steuern zu können. Wenn es um das Loggen serieller Daten geht, ist mit X1 ein Quarz unabdingbar. Auf der rechten Seite von Bild 1 sieht man mit K6 einen sechspoligen Programmieranschluss, der mit JP4 zwischen den

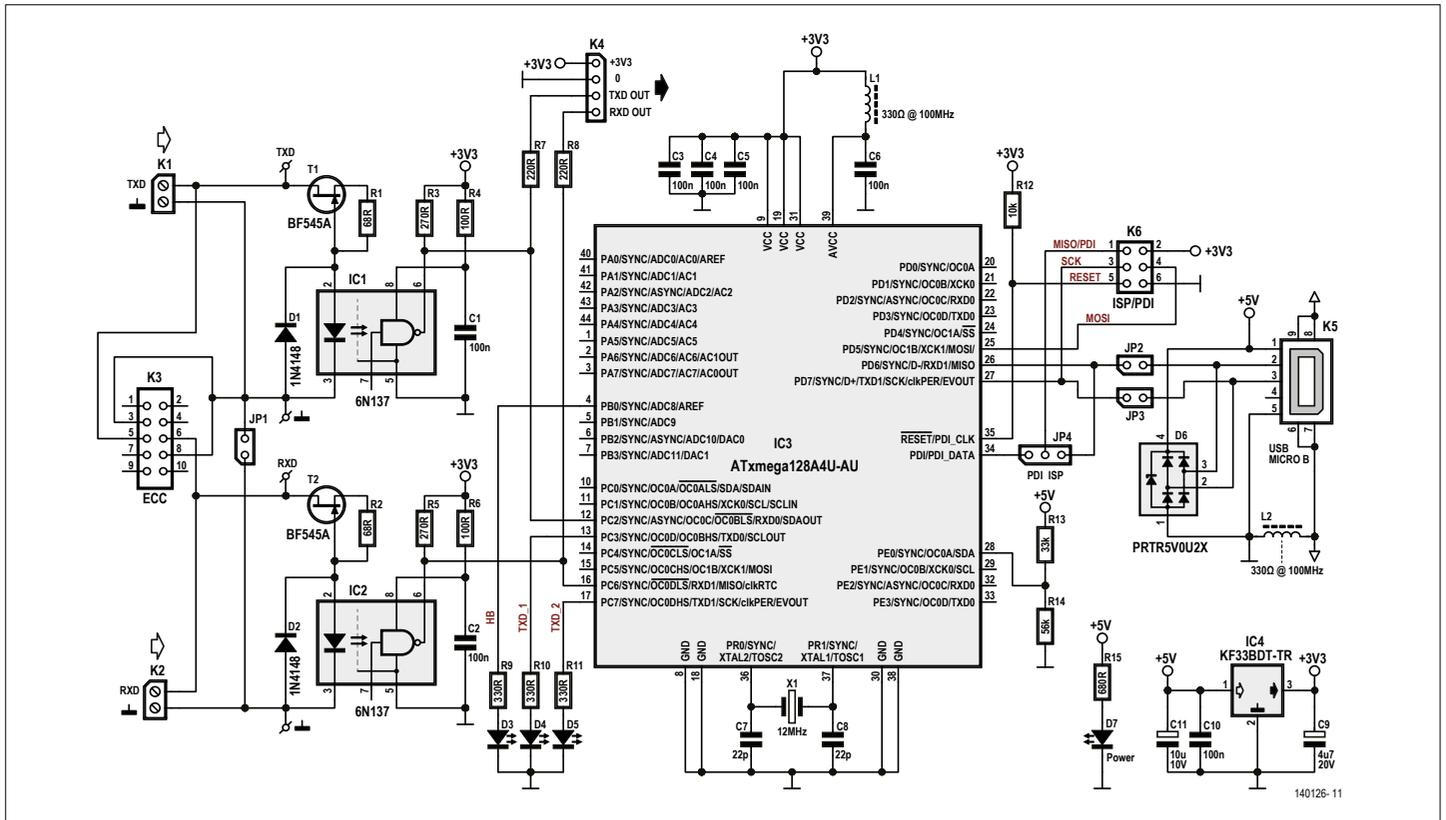


Bild 1. Schaltung des UART/RS232-Datenloggers.

Modi ISP und PDI umschaltbar ist. Die seriellen Signale der USB-Buchse K5 sind direkt mit den Pins 26 und 27 von IC3 verbunden. Die Diodenmatrix D6 schützt die Elektronik vor Über- und negativen Spannungen. Die beiden Jumper JP2 und JP3 müssen bei der Programmierung von IC3 unbedingt abgezogen werden.

Zeitstempel

Der Zeitstempel belegt intern 23 bit. Dies ermöglicht eine Auflösung im Millisekundenraster über eine Zeitspanne von 2 Stunden und 20 Minuten. Das Zeitstempelformat ist „a:bb:cc:ddd“, wobei:

a: Stunden, 0...2
 bb: Minuten, 00...59
 cc: Sekunden, 00...59
 ddd: Millisekunden, 000..999

Zusammen mit einem Daten-Byte belegt ein Datensatz daher 4 Bytes. In die 8 kB SRAM des Controllers passen so neben den Variablen und anderen Daten des Programms immerhin 900 komplette Datensätze.

Anschließend sollte man nicht vergessen, sie wieder aufzustecken!

Bei den Xmega-Controllern handelt es sich um eine Weiterentwicklung der bekannten ATmega-Serie von Atmel. Sie können deutlich mehr und sind schneller, vertragen aber keine 5-V-Betriebsspannung. Um den Datenlogger als kleines Modul direkt via USB zu versorgen, ist daher eine Umsetzung auf 3,3 V notwendig, die mit IC4 vorgenommen wird. LED D7 leuchtet, wenn der Logger an einen (eingeschalteten) PC angeschlossen ist. Über den Spannungsteiler aus R13 und R14 wird die 5-V-Spannungsversorgung an den analogen Eingang Pin 28 von IC3 gelegt. Der Datenlogger kann also auch überwachen, ob die 5-V-Versorgung des PCs Zicken macht oder sonstige Störungen aufweist. In der gegenwärtigen Version der Firmware ist dieses Feature allerdings nicht implementiert.

Firmware & Funktionen

Damit aus einer Mikrocontrollerschaltung etwas Nützliches wird, müssen die konkreten Funktionen mit Software realisiert werden. Die Software wurde mit Atmel Studio 6.2 entwickelt; das Original

stammt so wie auch die Idee von unserem Leser Thomas Schlott. Unser freier Mitarbeiter Francisco Ramirez hat die Software noch optimiert, unter anderem die USB-CDC-Verbindung implementiert (siehe unten). Bei einer USB-CDC-Kommunikation sieht es für das PC-Betriebssystem so aus, als ob der Mikrocontroller über eine klassische serielle Schnittstelle angebunden wäre (Virtual COM Port). Dann kann man ein normales Terminalprogramm verwenden, um über USB Zeichen zu empfangen und zu senden.

Die Software steht wie immer kostenlos als Source- und Hex-Code auf der Elektor-Webseite zu diesem Projekt [3] zum Download zur Verfügung. Die Funktionen stecken in kommentierten und daher gut nachvollziehbaren Code-Dateien (*.c) und ihre Definitionen in entsprechenden Header-Dateien (*.h). Die konkreten Aufgaben der Firmware sind:

- Interruptgesteuertes Schreiben von seriell einlaufenden Daten in den Pufferspeicher.
- Interruptgesteuerte Verarbeitung von Kommandos via USB.
- Die Main-Funktion der Firmware hat

zwei Teile:

- Initialisierung und Vorbereitung des Speichers sowie von Variablen und Modulen wie GPIO, TIMER, USART, Takt, USB, EEPROM, Flash-Speicher und RAM. Im RAM stecken die Variablen, im Flash-Speicher der Code plus der vordefinierten Strings und im EEPROM stehen die letzten Einstellungen.
- Start der Hauptschleife.
- Eine so genannte Heart-Beat-LED (D3) blinkt zum Zeichen, dass die Elektronik aktiv ist.
- Die LEDs D4 (Kanal 1) und D5 (Kanal 2) zeigen an, ob auf dem jeweiligen Kanal Daten einlaufen.
- Zur Anzeige der Daten auf einer Konsole werden diese samt Kanal-Info und Zeitstempel aus dem Speicher ausgelesen und via USB ausgegeben.
- Die Einstellungen betreffen zum Beispiel das Datenformat der Ausgabe. Die Modi können vom Anwender über Kommandos (siehe Kasten „Befehle“) verändert werden.

Software-Feinheiten

Beim Nachrüsten der USB-Schnittstelle musste Francisco Ramirez auf das Atmel Software Framework ASF [4] zurückgreifen, das einen USB-Stack für die USB-CDC-Kommunikation enthält. Man sollte besser nicht versuchen, das ASF zu entwirren, um etwa die benötigten Teile von Hand in das eigene Projekt zu übernehmen. Besser nutzt man das ASF dann gleich als Basis des eigenen Projekts. Man beginnt mit einem frischen Projekt („User-Board“-Template) und lässt den ASF-Wizard von Atmel Studio die nötigen Module zusammenstellen. In diesem Projekt waren das die folgenden:

- Generic Board Support
- GPIO – General Purpose Input/Output
- IOPORT – Input/Output Port Controller
- System Clock Control – XMEGA-A1U/A3U/A4U/B/C-Implementierung
- TC – Timer Counter
- USB-Device CDC

Leider ist das ASF gerade am Anfang nicht ganz einfach zu verstehen, denn es gibt zwar viel vorgefertigter Code und eine Fülle von Dokumentation, aber die ist über ziemlich viele Dateien ver-

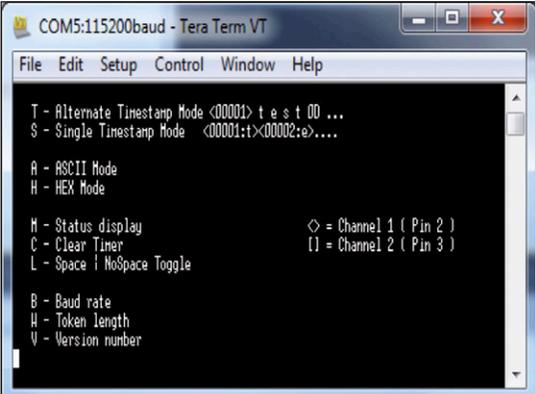
Konfiguration mit einfachen Befehlen

teilt. Nur als Beispiel sei erwähnt, dass die Einstellung des System-Takts im ASF anders funktioniert, als man das vielleicht gewohnt ist. Es scheint hier nötig zu sein,

den Takt für jedes Peripherie-Modul des Controllers eigenständig zu definieren. Die Verwendung des internen 32-MHz-Oszillators für USB und als Systemtakt miss-

Befehle

Zur Konfiguration relevanter Parameter durch den Anwender können mit einem gewöhnlichen Terminal-Programm wie HTerm, TeraTerm etc. Befehle an den Datenlogger geschickt werden. Das User-Interface umfasst folgende Befehle:



- S: Single Time Stamp Mode – gibt jedes eingelesene Zeichen in einer neuen Zeile mit eigenem Zeitstempel aus.
- T: Alternate Time Stamp Mode – gibt einen Zeitstempel aus, gefolgt von den hereinkommenden Zeichen. Ein neuer Zeitstempel (in einer neuen Zeile) wird erst ausgegeben, wenn auf dem anderen Kanal Zeichen empfangen werden.
- A: ASCII Mode – eingelesene Daten werden als ASCII-Zeichen ausgegeben.
- H: HEX Mode – eingelesene Daten werden im Hex-Format ausgegeben.
- E: Execute Line Feed – das Zeichen <LF> wird unmaskiert an das Terminalprogramm weitergegeben (und führt dort bei entsprechender Einstellung zu einem Zeilenumbruch).
- M: Show Status – zeigt den aktuellen Status diverser Einstellungen.
- C: Clear Timer – setzt den Zeitstempel zurück.
- L: Space / No Space Toggle – legt fest, ob zwischen zwei Zeichen ein Leerzeichen eingefügt wird oder nicht.
- B: Baud Rate – Auswahl der Baudrate für die seriellen Schnittstellen aus einer Liste.
- W: Token length – Auswahl der Bit-Länge eines RS232-Tokens.
- I: Inversion – invertiert die Pegel. Für jeden Kanal einzeln einstellbar; ein nochmaliges Betätigen des Befehls führt wieder zum Ausgangszustand. Beim Mitloggen einer RS232-Verbindung müssen beide Kanäle auf „not inverted“ stehen, bei TTL-UART auf „inverted“.
- V: Version number – informiert über die verwendete Software-Version.

Bei den Befehlen, die einen Parameter erfordern, gibt man zuerst das Befehlszeichen ein. Danach bekommt man eine Auswahlliste angezeigt.

Wichtig ist noch zu wissen, dass unterschiedliche Klammern genutzt werden, um die Daten beider Kanäle zu unterscheiden:

- <>: Channel 1 – spitze Klammern um Zeitstempel und die empfangenen Daten von Kanal 1.
- []: Channel 2 – rechteckige Klammern um Zeitstempel und die empfangenen Daten von Kanal 2.

lang. Der externe 12-MHz-Quarz konnte dann aber problemlos als Taktquelle eingebunden werden. Seine Erfahrungen mit dem ASF beschreibt Francisco Ramirez auf unserer Website Elektor-Labs.com [5].

Aufbau und Betrieb

Für den RS232-Datenlogger wurde eine kleine doppelseitige Platine entwickelt (siehe **Bild 2**), die fertig bestückt in **Bild 3** gezeigt ist. Da bis auf die Optokoppler, Stiftleisten, Buchsen und Klemmen alle Bauteile SMD-Form haben, muss man beim Löten sorgfältig sein und zunächst IC3 sowie IC4 und dann erst die Widerstände und Kondensatoren bestücken. Da es sich bei Letzteren fast durchweg um das SMD-Format 0805 handelt, dürfte man noch gut ohne Lupe auskommen. Lediglich bei den Dioden, den LEDs und den Tantal-Elkos gilt es, auf die korrekte Polung zu achten.

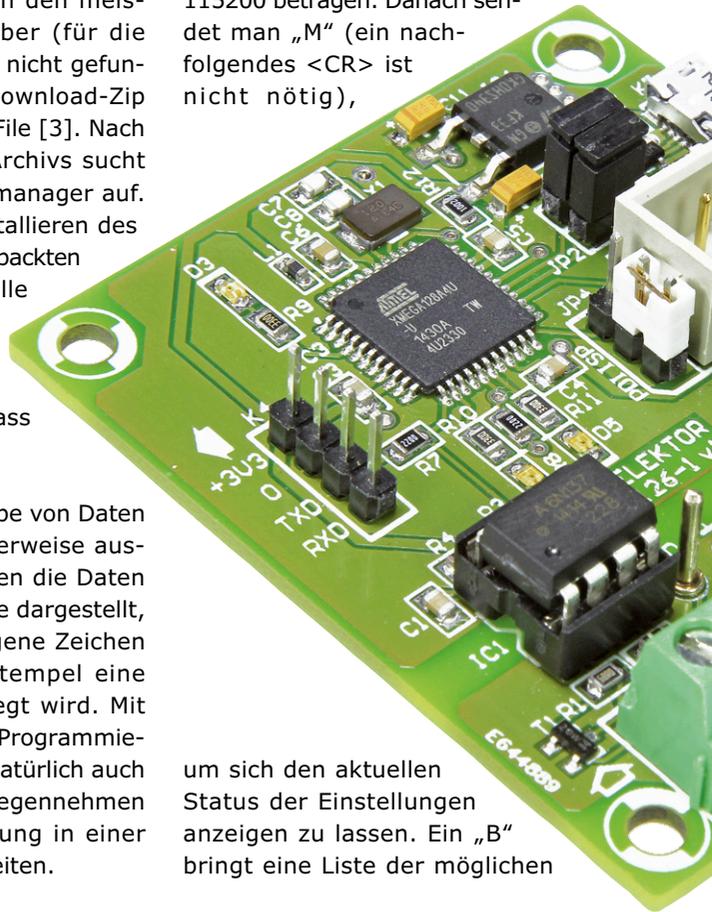
Ob die Eingangsstufe mit den Optokopplern funktioniert, kann man dank K4 sehr gut mit einem Oszilloskop prüfen, wenn man die jeweiligen Eingänge RXD und TXD mit seriellen Signalen speist. Wer über keinen passenden Programmierer verfügt, der kann im Elektor-Shop [3] problemlos einen fertig programmierten Mikrocontroller beziehen. Wer nicht löten möchte, der kann dort sogar eine fertig

bestückte und somit direkt einsatzbereite Platine erwerben.

Beim Anstecken des Datenloggers an den PC meldet Windows in den meisten Fällen, dass der Treiber (für die USB-CDC-Kommunikation) nicht gefunden werden konnte. Im Download-Zip findet man das nötige .inf-File [3]. Nach dem Auspacken des Zip-Archivs sucht man das Board im Geräte manager auf. Im Dialog zum (Nach-)Installieren des Treibers gibt man den ausgepackten Download-Ordner als Quelle an. Nach erfolgter Installation erscheint der Datenlogger als Gerät des Typs „Communication Device Class ASF Example“.

Bild 4 zeigt, wie die Ausgabe von Daten in einem Terminal typischerweise aussieht. In diesem Fall werden die Daten von Kanal 2 im Single-Mode dargestellt, bei der um jedes empfangene Zeichen samt individuellem Zeitstempel eine rechteckige Klammer gelegt wird. Mit etwas Erfahrung in der PC-Programmierung kann man die Daten natürlich auch mit eigener Software entgegennehmen und etwa für die Darstellung in einer Tabellenkalkulation aufbereiten.

Am besten stellt man das Terminalprogramm so ein, dass das Zeichen <LF> zu einem Zeilenumbruch führt. Die Baudrate (für die PC-Kommunikation!) muss 115200 betragen. Danach sendet man „M“ (ein nachfolgendes <CR> ist nicht nötig),



um sich den aktuellen Status der Einstellungen anzeigen zu lassen. Ein „B“ bringt eine Liste der möglichen

Stückliste

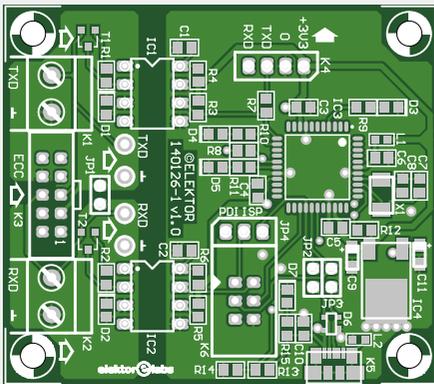


Bild 2. Bestückungsplan der Platine.

Widerstände:

(Standard: 5 %, 125 mW, SMD 0805)
 R1,R2 = 68 Ω, 1 %
 R3,R5 = 270 Ω
 R4,R6 = 100 Ω
 R7,R8 = 220 Ω
 R9..R11 = 330 Ω
 R12 = 10 k
 R13 = 33 k
 R14 = 56 k

R15 = 680 Ω

Kondensatoren:

(Standard: 10 %, 50 V, SMD 0805)
 C1..C6,C10 = 100 n, X7R
 C7,C8 = 22 p, COG/NPO 5 %
 C9 = 4μ7, 20 V, Tantal, SMD A
 C11 = 10 μ, 10 V, Tantal, SMD A

Spulen/Filter:

L1,L2 = 330 Ω @100MHz, 0,08 Ω, 1,7 A, SMD 0603

Halbleiter:

D1,D2 = TS4148 RY
 D3..D5 = LED, gelb, SMD 0805
 D6 = PRTR5V0U2X
 D7 = LED, grün, SMD 0805
 T1,T2 = BF545A, SMD SOT23
 IC1,IC2 = 6N137, DIP8
 IC3 = ATmega1284U-AU
 IC4 = KF33BDT-TR

Außerdem:

K1,K2 = 2-pol. Schraubklemme, Platinenmontage, RM 2/10“
 K3 = 2x5-Stiftleiste, RM 1/10“
 K4 = 1x4-Stiftleiste, RM 1/10“
 K5 = Micro-USB-Buchse, Typ B, SMD
 K6 = 2x3-pol. Wannenstecker, RM 1/10“
 JP1..JP3 = 2x1-Stiftleiste, RM 1/10“

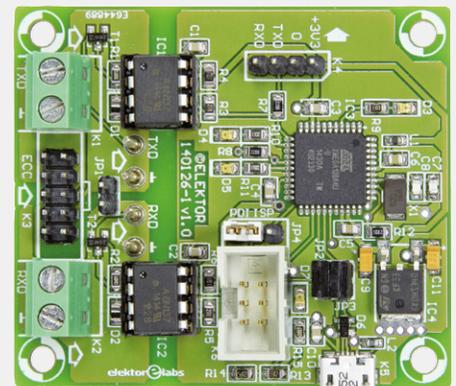


Bild 3. Die fertig bestückte Platine des UART/RS232-Datenloggers (Labor-Prototyp).

JP4 = 1x3-Stiftleiste, RM 1/10“
 JP1..JP4 = 2-pol. Jumper, RM 1/10“
 PC1..PC4 = Lötnagel, 1,3 mm
 X1 = Quarz, 12 MHz, 18 pF, SMD, 5x3,2 mm (Abracon ABM3-12.000MHZ-D2Y-T)
 Platine # 140126-1 v1.0 [3]
 Programmierter Mikrocontroller # 140126-41 [3]
 Fertig aufgebautes Modul # 140126-91 [3]

Baudraten für die mitzuloggende Verbindung auf den Schirm. Man wählt nun eine Baudrate aus durch Senden von „0“ bis „F“. Mit „I“ gefolgt von „1“ oder „2“ muss man einstellen, ob die Pegel auf dem Kanal 1 bzw. 2 invertiert werden sollen oder nicht. Bei einer RS232-Verbindung müssen die Kanäle Ch1 und Ch2 auf „not inverted“ stehen, bei TTL-UART auf „inverted“. Falls die serielle Kommunikation mit den üblichen 8 Datenbits (8N1) abläuft, müsste man spätestens jetzt mitgeloggte Zeichen im Terminalpro-



gramm angezeigt bekommen. Die anderen Befehle betreffen hauptsächlich das Format der Ausgabe; eine Zusammenstellung findet man im Kasten „Befehle“. ◀

(140126)

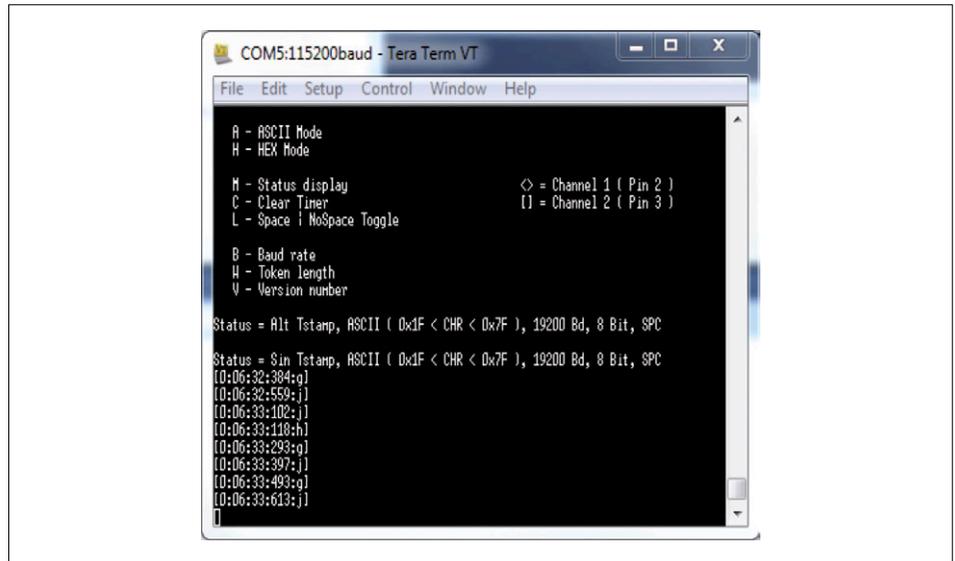
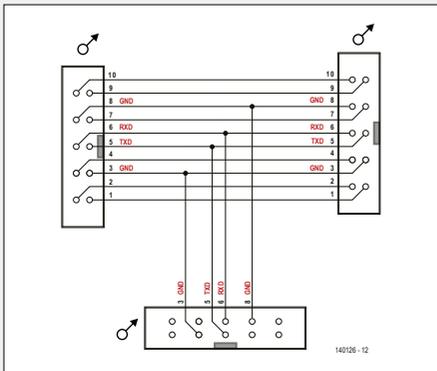


Bild 4. Ausgabe von Daten auf Kanal 2 in Tera Term.

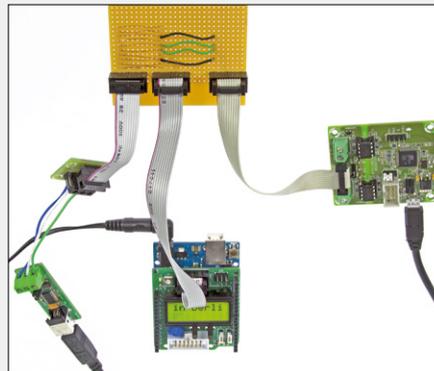
Weblinks

- [1] www.atmel.com/devices/ATXMEGA128A4U.aspx
- [2] www.elektor-magazine.de/130155
- [3] www.elektor-magazine.de/140126
- [4] www.atmel.com/tools/avrsoftwareframework.aspx?tab=overview
- [5] www.elektor-labs.com/contribution/from-the-lab-new-software.14381.html
- [6] www.elektor-magazine.de/140009
- [7] www.elektor-magazine.de/130023

Daten loggen mit dem ECC



Der von Elektor spezifizierte *Embedded Communication Connector* (ECC) gibt TTL-UART-Signale über zwei Pins eines 2x5-poligen Steckverbinders weiter. Das Elektor Extension-Shield für den Arduino Uno [6] ist bereits mit einem solchen Steckverbinder ausgerüstet, in Kürze wird ein Adapter für das SAM-D20-Board aus dem ARM-Kurs folgen. An den Steckverbinder sind über ein Flachkabel



verschiedene Kommunikations-Module anschließbar, vorgestellt wurden bereits ein RS485-Konverter [2], ein 433-MHz-Funkmodul [7] und ein UART/USB-Konverter (in diesem Heft). Ein NFC-Gateway und weitere Module werden folgen.

Will man nur mitloggen, was beispielsweise der Arduino Uno ausgibt, dann kann man das Extension-Shield

direkt über ein Flachkabel an den Datenlogger anschließen. Falls man bidirektional mitloggen will, dann benötigt man einen Y-Verteiler für die Signale, welcher sinnvollerweise mit drei ECC-Steckverbindern ausgestattet ist. Eine entsprechende Platine wird gerade in unserem Labor entworfen. Für erste Experimente haben die Kollegen die hier gezeigte Schaltung auf einer Lochrasterplatine aufgebaut. Das Foto zeigt das Mitloggen von Daten, die zwischen dem Arduino Uno und einem PC ausgetauscht werden (siehe dazu auch die „Learn“-Einleitungsseite in diesem Heft). Der Y-Verteiler samt Datenlogger wurde hierzu in die Verbindung zwischen dem Shield und dem RS485-Konverter eingeschleift. Die ECC-Masse wird bei offenem Jumper JP1 nur an einen Kanal des Datenloggers weitergegeben (siehe Bild 1). JP1 muss also gesteckt werden.



Willkommen in Ihrem **E-SHOP**

ELEKTOR EMPFIEHLT

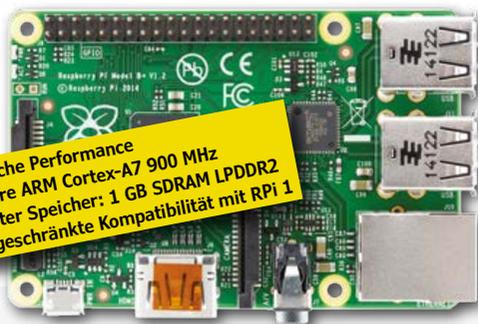
Eines Tages musste er ja kommen, der Allround-Computer – mit Internetzugang, HD-Grafik, der bequem in die Hand passt. Und das Ganze zu einem unglaublich erschwinglichen Preis! Seit drei Jahren erleben wir eine Überraschung nach der nächsten. Anzeigen über angeblich wundertätige neue Produkte bekommen wir tagtäglich nachgeschmissen, aber die völlig überraschende Ankündigung des RPi 2 (Anfang Februar) war ein echter Adrenalinstoß. Jetzt kann ich endlich einen alten Traum realisieren: für meinen guten alten Analog-Synthesizer aus den 80er-Jahren mit seinen Dutzenden von Modulen (immer unter Spannung) ist nun die ultimative Musiker-Maschine-Schnittstelle machbar. Endlich lassen sich überlegene analoge Synthesetechnik und moderne numerische Verfahren miteinander verbinden, endlich werden Grafik-Interfaces und spontanes Speichern



selbst der komplexesten Klang-Konfigurationen möglich.

Denis Meyer
(Chefredakteur Elektor Frankreich)

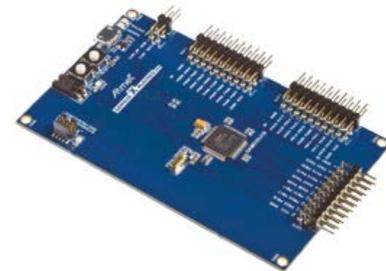
www.elektor.de/rpi-2



- Sechsfache Performance
Quadcore ARM Cortex-A7 900 MHz
- Doppelter Speicher: 1 GB SDRAM LPDDR2
- Uneingeschränkte Kompatibilität mit RPi 1

Elektor-Bestseller

1. Atmel SAM D20 Xplained Pro
www.elektor.de/samd20-board



2. Raspberry Pi 2 (Mod. B)
www.elektor.de/rpi-2

3. Elektor-DVD 2014
www.elektor.de/dvd-2014

4. EveryCircuit (Jahreslizenz)
www.elektor.de/everycircuit

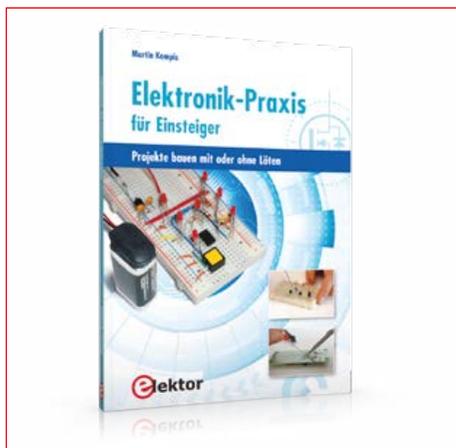
5. Praxiskurs
AVR-XMEGA-Mikrocontroller
www.elektor.de/kurs-avr-xmega

6. Raspberry Pi (Mod. A+)
www.elektor.de/rpi-a-plus

7. Mikrocontroller verstehen und anwenden
www.elektor.de/mcu-verstehen

8. Elektor-DVD 2000-2009
www.elektor.de/2000-2009

Elektronik-Praxis für Einsteiger

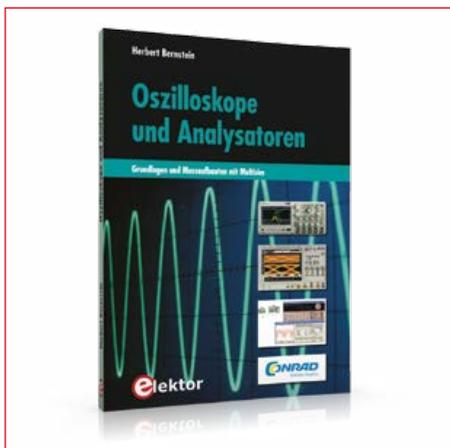


Wenn Sie nachbausichere Schaltungen suchen, die mit wenigen elektronischen Bauteilen auskommen, problemlos funktionieren und Spaß machen – dann liegen Sie mit diesem Buch richtig. Hier finden Sie beispielsweise Radioempfänger für UKW, LW und MW, eine akustische Illusion, ein Mini-Oszilloskop zur grafischen Anzeige von Spannungsverläufen, elektronische Orgeln, einen elektronischen Würfel mit Mikrocontroller und weitere funktionssichere Schaltungen.

Mitgliederpreis: € 32,80 • CHF 34.95

www.elektor.de/elektronik-praxis-einsteiger

Oszilloskope und Analytoren



Dieses Fachbuch beinhaltet alles über analoge Oszilloskope, digitale Speicheroszilloskope, Logikanalysatoren, Bode-Plotter, Spektrum- und Netzwerkanalysatoren mit praxisorientierten Fakten. Der Autor hat auch für die komplexen Vorgänge der elektronischen Messtechnik praktische kurze Erklärungen und zeigt viele Anwendungen aus der Messpraxis. Im Wesentlichen basiert das gesamte Buch auf der Simulation mit Multisim.

Mitgliederpreis: € 42,00 • CHF 44.95

www.elektor.de/oszilloskope-und-analysatoren

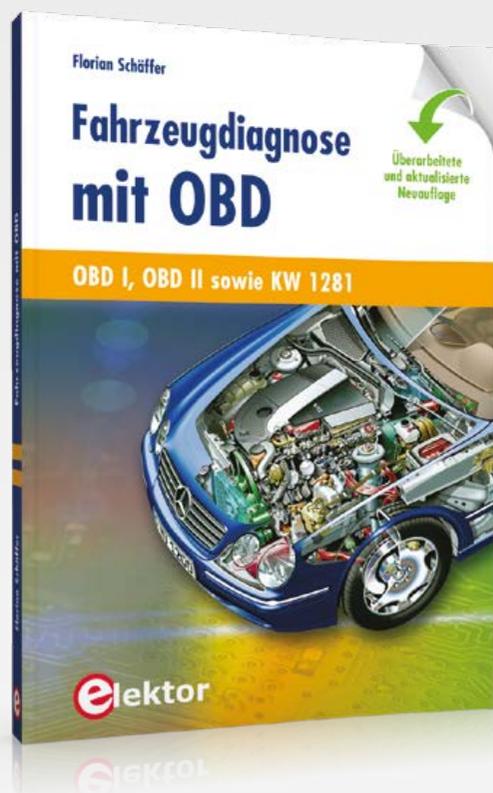
USBprog 5.0



Der USBprog 5.0 OpenOCD ist ein Programmier- und Debugadapter, der komplett ohne extra Installation von zusätzlicher Software auf dem PC (egal ob Windows, Mac oder Linux) auskommt. Im einfachsten Fall bedient man alles per Webbrowser. Soll der Programmierer aus einer Batchdatei oder einem Makefile angesteuert werden, gibt es dafür ein einfaches Kommandozeilentool.

Mitgliederpreis: € 98,90 • CHF 103.95

www.elektor.de/usbprog5



Streikt Ihr Auto schon wieder und Sie befürchten, dass die nächste teure Reparatur fällig wird? Vielleicht ist es aber auch nur ein kleiner Fehler, den Sie selber beheben können. Doch woher sollen Sie das bei einem modernen Fahrzeug mit all der Elektronik unter der Motorhaube wissen? Hier kann die fahrzeugeigene Diagnosefunktion helfen, den Fehler zu finden und Reparaturkosten zu senken, sodass Sie nicht bei jedem Aufblinken der Warnlampen gleich in die Werkstatt müssen. Nur mit einem geeigneten Interface zum Auslesen der Fehlercodes und zahlreichen Messwerten der elektronischen Sensoren ist bei modernen Autos überhaupt noch eine Fehlerdiagnose möglich. Neben der praxisorientierten Beschreibung der heutigen Diagnosemöglichkeiten für den ambitionierten Autofreak beschreibt das Buch den Selbstbau eines preiswerten Diagnose-Interface und welche Fertiglösungen es auf dem Markt gibt.

Fahrzeugdiagnose mit OBD

OBD I, OBD II sowie KW 1281
Überarbeitete und aktualisierte Neuauflage!

Analog Circuit Design 1-3

Mitglieder sparen doppelt!

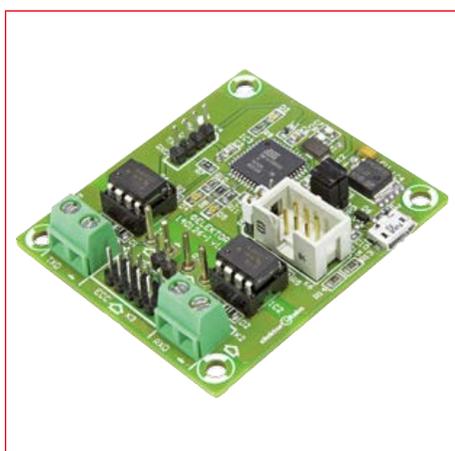
Arduino Sensor-Kit (37 in 1)

Vollständig Arduino-kompatibel!



Mitgliederpreis (frei Haus): € 37,50 • CHF 39.95
www.elektor.de/fahrzeugdiagnose-mit-obd

UART/RS232-Datenlogger

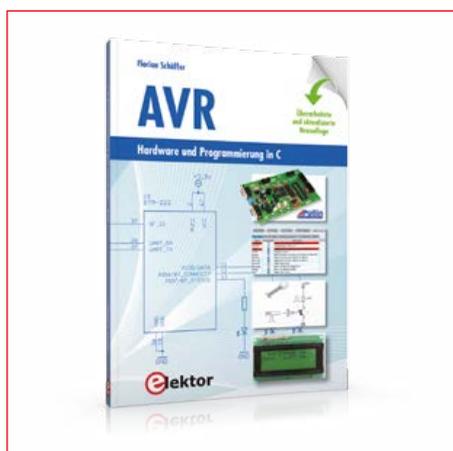


Wer bei Entwicklung und Test feststellt, dass die Kommunikation noch hakt, muss sich auf eine knifflige Fehlersuche einstellen. Unser UART-Datenlogger macht das Leben leichter – er registriert Bytes in beiden Richtungen millisekundengenau. Dank flexibler Anschlussmöglichkeiten und einer galvanischen Trennung lässt er sich leicht in serielle Verbindungen mit TTL- oder RS232-Pegeln einschleifen.

 Mitgliederpreis: € 31,46 • CHF 33.95

www.elektor.de/140126-91

AVR

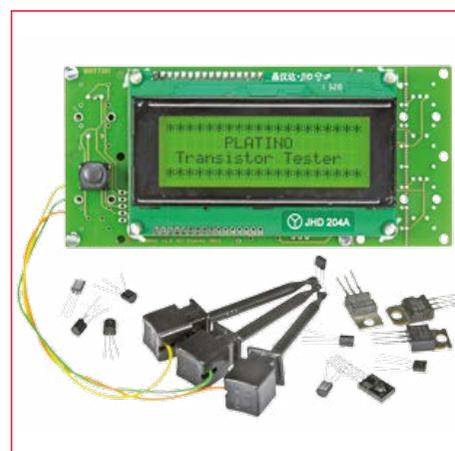


Der Autor führt Einsteiger und auch Fortgeschrittene gekonnt und professionell in eine hochinteressante Thematik ein. Auch wer seine Elektronik- und Programmierkenntnisse weiter ausbauen und vertiefen möchte, hat dazu gute Möglichkeiten. Nach Einführung und Vorstellung der notwendigen Entwicklungsumgebung werden Projekte vorgestellt, die schrittweise zum Ziel führen.

 Mitgliederpreis: € 34,80 • CHF 36.95

www.elektor.de/avr-buch

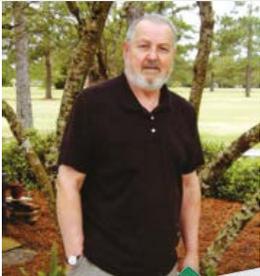
Platino-Transistortester (Kit)



Dieser einfache Transistortester auf Platino-Basis testet zweifelhafte und/oder nicht gekennzeichnete Transistoren auf Funktion und ermittelt den Verstärkungsfaktor, damit auch Transistorpärchen ausgemessen werden können. Das Gerät ist für NPN- und PNP-Bipolartransistoren sowie für n- und p-Kanal-MOSFETs geeignet.

 Mitgliederpreis: € 62,95 • CHF 67,57

www.elektor.de/platino-transistortester-kit



Von Gene Sellier

Ich habe die T-Boards gekauft, um auf eine Arduino-Platine verzichten zu können – vor allem in den Fällen, wo ein ATtiny-Mikrocontroller völlig ausreichend ist. Mir gefällt vor allem das T-Board 8, da der 8-Pin ATtiny die Anzahl der benötigten Teile deutlich reduziert. Die T-Boards sind wirklich ideal, wenn man mit Steckbrettern arbeitet. Ich verwende WinAVR IDE und einen USBtinyISP-Programmer von Adafruit, um meine Schaltungen zu entwickeln.

Für die Regelung eines 12-V-Netzteil Lüfters sind zum Beispiel nur ein ATtiny 45, ein Thermistor, ein Widerstand und ein MOSFET erforderlich. Der Mikrocontroller liest die Temperatur im Innern des Gehäuses und stellt die Lüfterdrehzahl ein. Dafür benötigt man nur zwei Pins und ein paar Zeilen Code. Da mir noch zusätzliche Pins zur Verfügung standen, beschloss ich eine LED (rot/grün) mit dazugehörigen Widerständen an der Frontplatte anzubringen, um die Temperatur im Gehäuse visuell anzuzeigen. Schließlich baute ich die Schaltung auf einer kleinen Experimentierplatine (2 x 4 cm) auf. Das T-Board hat den gesamten Prozess vereinfacht und beschleunigt, da man viel weniger Verbindungsdrähte benötigt.

Jetzt bestellen unter www.elektor.de/t-board-8

Wenn Sie uns einen ähnlichen Bericht über ein Elektor-Produkt schicken, der in Elektor veröffentlicht wird, erhalten Sie als Dankeschön einen 100-EUR-Gutscheincode, den Sie im Elektor-Shop einlösen können.

Mehr Infos gibt's unter www.elektor.de/review.



ARDUINO
DAY 2015
WEEK! 24.-30. MÄRZ

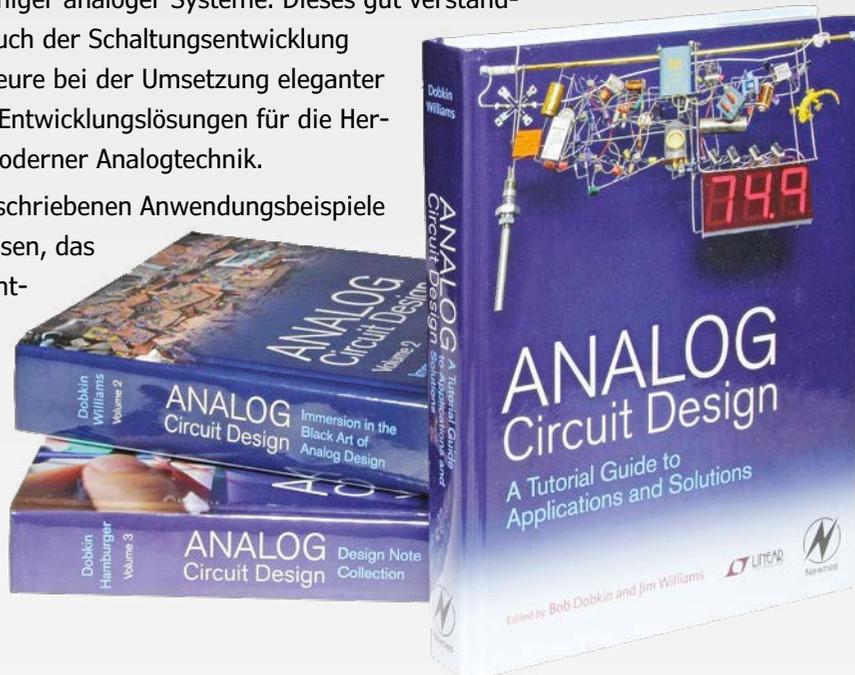


TOLLE ARDUINO-DEALS
@ELEKTOR.DE



Die Entwicklung analoger Elektronik wird immer wichtiger. Mit der Zunahme vielfältiger digitaler Systeme, von funkbasierter Kommunikation sowie komplexer Systeme für Industrie und den Bereich Automotive stellen sich dem Entwickler besondere Herausforderungen beim Design leistungsfähiger analoger Systeme. Dieses gut verständliche Grundlagenbuch der Schaltungsentwicklung unterstützt Ingenieure bei der Umsetzung eleganter und zielführender Entwicklungslösungen für die Herausforderungen moderner Analogtechnik.

Die ausführlich beschriebenen Anwendungsbeispiele vermitteln das Wissen, das für die moderne Entwicklungspraxis unabdingbar ist.



Fahrzeugdiagnose mit OBD

OBD I, OBD II sowie KW 1281 Überarbeitete und aktualisierte Neuauflage!

Analog Circuit Design 1-3

Mitglieder sparen doppelt!

Arduino Sensor-Kit (37 in 1)

Vollständig Arduino-kompatibel!



Mitgliederpreis: € 161,10 • CHF 169,50
www.elektor.de/analog-circuit-design

Mikrocontroller verstehen und anwenden



Mit diesem Buch erweitert der Leser seine Mikrocontroller-Kenntnisse auf Grund eigener Erfahrungen und Erfolgsergebnisse und wird dazu noch ganz nebenbei in die Welt des Arduino und seiner Entwicklungsumgebung eingeführt. Am Ende dieses 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.

Mitgliederpreis: € 42,00 • CHF 44,95

www.elektor.de/mcu-verstehen-anwenden

Mystery Product



www.elektor.de/mystery

Raspberry Pi 2 (Mod. B)



Die kleinste und billigste Version des sehr erfolgreichen Einplatinenrechners Raspberry Pi wurde komplett überarbeitet. Die neue Version mit dem Zusatz „+“ zeichnet sich durch geringeren Stromverbrauch, bessere Tonqualität, mehr GPIO-Pins und zusätzliche USB-Anschlüsse aus. Die Gesamtgröße der Platine ist ebenfalls geschrumpft.

Mitgliederpreis: € 38,66 • CHF 40,95

www.elektor.de/rpi-2

- ✓ **Serieller USB-nach-Multi-Protokoll-Konverter**
 Platine: 130542-1
 Modul: 130542-91
- ✓ **UART/RS232-Datenlogger**
 Platine: 140126-1
 Progr. Controller: 140126-41
 Modul: 140126-91
- ✓ **Lüftungssystem für den Keller**
 Platine: 100892-1
 Progr. Controller: 100892-41
 Modul: 140154-91
- ✓ **e-Bob BL600**
 Platine: 140270-1
 Modul: 140270-91

www.elektor.de/neu-im-heft

elektor•post: Neuigkeiten & Angebote aus erster Hand



Über 100.000 Elektronik-Profis, ambitionierte Entwickler und anspruchsvolle Anwender lesen regelmäßig unseren Gratis-Newsletter „Elektor.POST“ – die wöchentliche Online-Ausgabe von Elektor! Unsere Web-Redakteure halten Sie jeden Freitagmorgen mit neuesten und interessanten Meldungen, Downloads, Tipps & Trends aus der Welt der Elektronik auf dem Laufenden. Des Weiteren werden Sie schnell und umfassend über aktuelle Elektor-Projekte sowie über das umfangreiche Elektor-Sortiment und spezielle Produkt-Angebote als Erster informiert.

- Alle 2 Wochen ein Schaltungsprojekt frisch aus dem Labor (Wert je 2,50 €)
- Top-Angebote aus dem Elektor-Shop
- Gratis-PDF-Bundle im Wert von 40 €

JETZT ANMELDEN UND VORTEILE SICHERN:
www.elektor.de/newsletter

Funkempfängerkompodium



Wollten Sie schon immer wissen, wie sich die klassische Funkempfängertechnik fortentwickelt hat? Wie funktionieren professionelle Funkempfänger heute und was können sie leisten? Welche Empfangssysteme und Techniken stehen heute zur Verfügung? Möchten Sie auch ausgefallene Anwendungen von Empfängern kennenlernen und wissen, wie ein SDR nun wirklich funktioniert und was der letzte Stand der entsprechenden Technik kann? In diesem Buch findet man die Antworten!

Mitgliederpreis: € 49,00 • CHF 51.95

www.elektor.de/funkempfangerkompodium

Praxiskurs AVR-XMEGA-Mikrocontroller

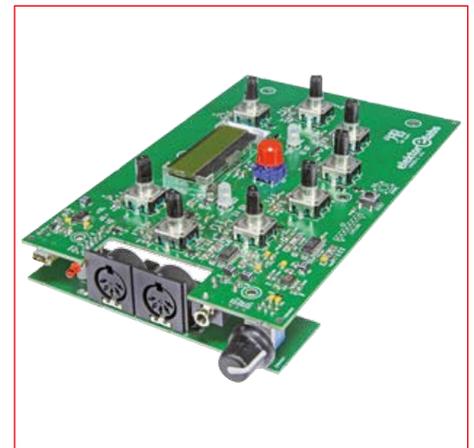


Die Mikrokontrollertechnologie ist eines der wichtigsten Gebiete der modernen Elektronik. Mikrocontroller haben sich in den letzten Jahren in allen Bereichen der modernen Technik etabliert. Der vorliegende Kurs gibt eine umfassende Einführung in die faszinierende Welt der Controllertechnik.

Mitgliederpreis: € 39,80 • CHF 41.95

www.elektor.de/praxiskurs-avr-xmega

J²B-Synthesizer



Der J²B-Synthesizer ist ein monophoner 9-bit-Synthesizer, der voll via MIDI steuerbar ist und originale Klänge mit Biss produziert. Er ist an das ATmega16 von Soulsby Synthesizers angelehnt, basiert allerdings auf dem leistungsfähigen Controller ARM-Cortex-M0+ des Typs LPC1347 mit 32-bit. Das Projekt ist zu 100% Open Source und Open Hardware.

Mitgliederpreis: € 112,45 • CHF 118.95

www.elektor.de/j2b-synthesizer



Arduino Sensor Kit

Dieses umfangreiche Sensor-Kit ist vollständig Arduino-kompatibel. Das Paket enthält alles, was Sie brauchen, um noch mehr aus dem Arduino herauszuholen. Die 37 Sensormodule werden in einer praktischen Plastikbox mit separaten Fächern geliefert.



Fahrzeugdiagnose mit OBD

OBD I, OBD II sowie KW 1281 Überarbeitete und aktualisierte Neuauflage!

Analog Circuit Design 1-3

Mitglieder sparen doppelt!

Arduino Sensor-Kit (37 in 1)

Vollständig Arduino-kompatibel!



Mitgliederpreis: € 49,46 • CHF 51.95

www.elektor.de/arduino-sensor-kit

Sie sind einzigartig!

THANK YOU.

Jetzt sind wir dran:

Wir wissen Ihre Treue zu schätzen.

Exklusiv nur für MITGLIEDER!





Intelligentes Lüftungssystem regelt Luftfeuchte

Von **Luc Lemmens** (Elektor-Lab) Nach einer Idee von **Danny Winkler**

Feuchte Keller sind ein weit verbreitetes Phänomen. Meistens sind sie die Folge von Kondenswasser, das bei niedrigen Temperaturen in unbeheizten Räumen entsteht. Muffige Gerüche durch Schimmel, Korrosion an Metallrohren, das Faulen von Holz sowie Schäden an lagerndem Inventar sind oft die Folgen. Der ständige Austausch der Raumluft durch regelmäßiges Lüften kann vorbeugend wirken oder das Problem beseitigen. Doch leider reicht es nur selten aus, gelegentlich Fenster oder Türen zu öffnen. Das Lüftungssystem, das wir hier vorstellen, wendet eine effizientere Methode an. Es kann sogar dazu beitragen, dass Kellerräume bewohnbar werden.

Keller und Dachböden sind Räumlichkeiten im Haus, in denen sich viele Menschen möglichst nur kurz aufhalten. Seit der Mensch in Häusern lebt, sind sie vornehmlich dafür bestimmt, Gebrauchsgüter und Vorräte zu lagern. Im Lauf der Zeit stapelt sich das, was nicht mehr gebraucht wird, aber vermeintlich zu wertvoll ist, um es zu entsorgen. Beheizbar sind Keller- und Bodenräume in der Regel nicht, Partykeller und Hobbyräume sind, sofern vorhanden, eine Ausnahme.

Während Dachböden durch die darunterliegenden warmen Geschosse und die Sonneneinstrahlung auch im Winter noch mäßig auf Temperatur gehalten werden, herrschen in Kellern niedrige Temperaturen vor. Wegen dieser Eigenschaft, die auf die Lage unter Erdniveau zurückzuführen ist, bieten sich Kellerräume auch zum Lagern und Frischhalten von Lebensmitteln an. Allerdings besteht heute im Zeitalter der Kühlschränke und Tiefkühlgeräte weniger Bedarf als in früheren Zeiten. Die

niedrigen Kellertemperaturen begünstigen die Feuchtigkeit, sie haben zur Folge, dass ein Teil des Wasserdampfs kondensiert, der in nachströmender warmer Luft gebunden ist. Das Wasser schlägt sich an kaltem Mauerwerk als feuchter Film oder als Tropfen nieder. Diese Erscheinung macht sich im Sommer auch an Kaltwasserleitungen bemerkbar, die herabhängenden Tropfen deuten selten auf Haarrisse im Rohr oder undichte Rohrverbindungen hin.

Der Wasserhaushalt in Räumen unter Erdniveau ist von komplexer Natur, denn auch eindringendes Grundwasser bei schadhafter Bausisolierung und unfachmännische Regenableitung im umgebenden Erdboden können dazu beitragen, dass Kellerräume feucht sind. Das gilt insbesondere für Altbauten, moderne Neubauten sind dank einschlägiger Normen und Bauvorschriften in der Regel nicht betroffen. Die Bekämpfung äußerer Einwirkungen wollen wir hier außer Betracht lassen, denn sie sind Sache der Baufachleute und Bausanierer. Hier geht es um die Feuchtigkeit, die durch mangelnde Lüftung entsteht, einschließlich ihrer Folgen: Gesundheitsgefährdende Schimmelbildung, korrodierende Metallteile, verfaulendes Holz sowie üble, modrige Gerüche. Ein Lüftungssystem kann Abhilfe schaffen, wenn es überlegt konzipiert und realisiert ist.

Leserentwurf

Auf unserer Entwickler-Website elektor-labs.com stellte ein Elektor-Leser ein Mikrocontroller-System vor [1], das die Luftfeuchte seines Kellers auf optimalem Niveau hält. Zwei Sensoren messen die Luftfeuchten und Temperaturen sowohl im Keller als auch außerhalb des Hauses. Die Messwerte steuern das Öffnen eines Kellerfensters und das Zuschalten eines elektrischen Lüfters. Der Leser berichtet, dass das System das Problem der feuchten Kellerräume gelöst hat.

Wir haben beschlossen, das Leserprojekt aufzugreifen und für die Veröffentlichung in Elektor aufzubereiten. Zum originalen Entwurf gehört keine Platine, doch statt ein weiteres Mikrocontroller-Board mit LCD, Bedientastern und Stromversorgung zu entwerfen, haben wir unser Platino-Board als Basis genommen. Die Software haben wir mit der Arduino-IDE neu geschrieben, so dass es nicht schwierig sein dürfte, örtlich notwendige oder gewünschte Modifikationen vorzunehmen. Die Elektronik, die den Fensteröffner und den Lüfter steuert, sowie die Anbindung der Sensoren werden wir in einem nachfolgenden Beitrag beschreiben.

Im originalen Entwurf des Lesers kamen zwei kostenträchtige Feuchte- und Temperatur-Sensoren zum Einsatz, wir haben sie gegen preiswerte ChipCap2-Sensoren ausgetauscht. Unser CC2-eBoB wurde bereits in der Elektor-Ausgabe vom Januar/Februar dieses Jahres beschrieben. Die Aufsteckplatine (siehe **Bild 1**)

Eigenschaften

- Zwei ChipCap2-Sensoren zum Messen der relativen Luftfeuchten (Genauigkeit 2 %) und Temperaturen (Toleranz $\pm 0,6$ °C)
- Vierzeiliges alphanumerisches LC-Display, 4 · 20 Zeichen
- Betriebsspannung 8...12 V_{DC}
- Software für Mikrocontroller ATmega328 (wie Arduino)
- Bedienung über Dreh-Encoder mit integriertem Drucktaster
- Steuerung eines elektrisch angetriebenen Fensteröffners
- Steuerung eines elektrischen Lüfters oder Lufttrockners

erlaubt die Montage des kombinierten Feuchte- und Temperatur-Sensors CC2D35 von Amphenol mit herkömmlichem Lötwerkzeug.

Das Messsystem...

Eigentlich geht es kaum einfacher: Während ein Sensor die Temperatur und Luftfeuchte im Keller misst, erfasst ein zweiter Sensor die gleichen Messgrößen im Freien. Die Werte der relativen Luftfeuchten sind allein noch nicht aussagefähig, auf die Temperaturen kommt es ebenfalls an. Der Grund ist das Gesetz der Physik, dass warme Luft mehr Wasserdampf als kalte Luft aufnehmen kann. So hat zum Beispiel das Lüften des Kellers bei glühender Sommerhitze zur Folge, dass aufgeheizte Luft in den Keller strömt und ein Teil des gebundenen Wasserdampfs wegen der niedrigen Kellertemperaturen zu Wasser kondensiert. Aus der relativen Luftfeuchte und der Temperatur lässt sich die absolute Luftfeuchte errechnen, sie ist ein Maß für die Feuchtekonzentration in Gramm Wasser pro Kubikmeter Luft. Dies ist die Größe, die für die Lüftungsregelung benötigt wird (siehe Kasten). Wie bei anderen Strömungsvorgängen breitet sich auch die Luftfeuchte von höheren zu niedrigen Konzentrationen aus. Wenn die absoluten Luftfeuchten sowohl im Keller als auch im Freien bekannt sind, lässt sich entscheiden, wann das Fenster geöffnet und der Lüfter eingeschaltet werden müssen.

...und das Prinzip der Regelung

Auch das ist nicht kompliziert: Wenn die Feuchtekonzentration im Keller höher als im Freien ist, muss die Lüftung eingeschaltet werden, in jedem anderen Fall bleibt sie ausgeschaltet. Das klingt zwar simpel, doch es gibt einige Klippen. Herrscht draußen strenger Frost, ist es nicht sinnvoll, das Kellerfenster über Stunden geöffnet zu halten. Abgesehen vom Energieverlust im Haus können Wasserleitungen einfrie-

ren und platzen. Ferner lässt sich nicht ausschließen, dass ein Sensor irgendwann einmal ausfällt. In diesem Sonderfall darf das System nicht unkontrolliert in Betrieb bleiben. Zwar würde es eine Weile dauern, bis die relative Luftfeuchte im Keller einen Mindestwert unterschreitet, doch auch diese Situation darf möglichst nicht eintreten. Aus vorstehenden Überlegungen ergeben sich für das Öffnen des Fensters und das Einschalten des Lüfters folgende Kriterien:

1. Ein Sensor defekt: Fenster geschlossen, Lüfter ausgeschaltet.
2. Innentemperatur < Frostgrenze: Fenster geschlossen, Lüfter ausgeschaltet.
3. Relative Luftfeuchte innen zu niedrig: Fenster offen, Lüfter ausgeschaltet.
4. Absolute Luftfeuchte innen > außen: Fenster offen, Lüfter eingeschaltet.
5. Absolute Luftfeuchte innen < außen: Fenster geschlossen, Lüfter ausgeschaltet.

Die Nummerierung der Kriterien gibt die Prioritäten an, wobei Situation 1 die höchste Priorität hat. Die Betriebszustände von Fenster und Lüfter werden im Abstand von zehn Minuten aufgrund der zuletzt gemessenen Werte gesteuert, das



Bild 1. Die Adapterplatine erleichtert den Umgang mit dem ChipCap2-Sensor.

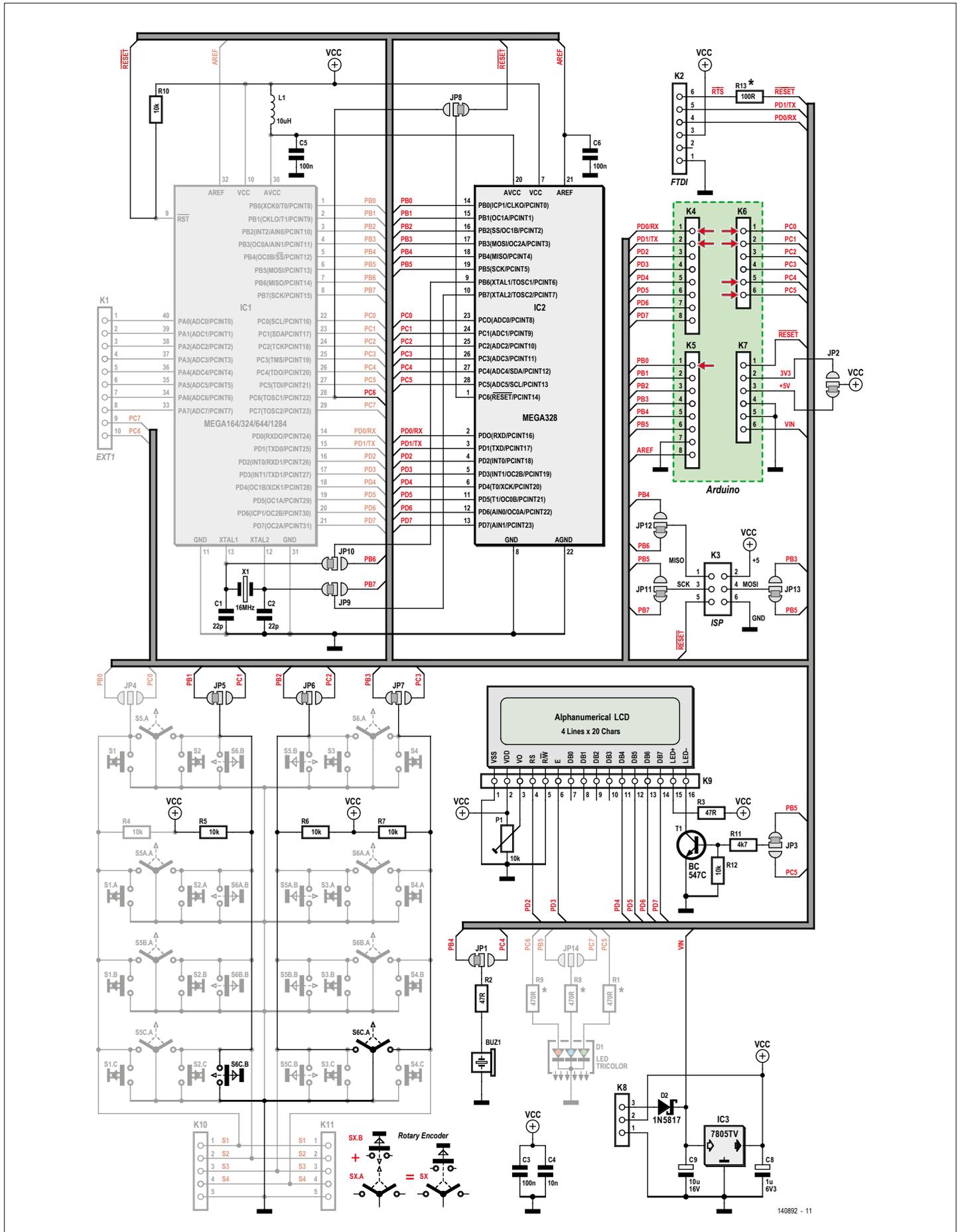


Bild 2. Gesamtschaltung des Platino. Die Bauteile und Brücken des Kellerlüftungssystems sind hervorgehoben.

System kommt deshalb ohne Hysterese aus. Der lange zeitliche Abstand verhindert, dass Schaltvorgänge unaufhörlich aufeinander folgen, was nicht nur permanent Geräusche verursacht, sondern auch unnötig Energie kostet.

Platino

Ein Platino-Board (siehe Elektor Oktober 2011) ist die Steuer- und Schaltzentrale des Systems. Auf diesem Board befindet sich ein ATmega328 mit Arduino-Bootlader, in der Arduino-Entwicklungsumgebung verhält sich das Board wie ein Arduino Uno. **Bild 2** zeigt die vollständige Schaltung des Platino, wobei die hier benötigten Komponenten farbig hervorgehoben sind. Aus der Schaltung geht hervor, welche Lötbrücken auf der Platine offen oder geschlossen sein müssen. Die Stückliste fasst die Komponenten zusammen. Natürlich gehören auch die Feuchte- und Temperatur-Sensoren zur Hardware. Wie schon erwähnt, kommen zwei Exemplare des CC2-eBoB zum Einsatz (Elektor-Shop 140154-91). Die darauf befindlichen ChipCap2-Sensoren werden an den I²C-Bus des Systems gelegt. Beide Exemplare haben im Lieferzustand die I²C-Adresse 0x28. Damit sie am gleichen I²C-Bus betrieben werden können, muss die Adresse eines Sensors geändert werden. Das lässt sich mit einem kurzen Sketch (Kurzprogramm) bewerkstelligen, jedoch erst, wenn der Platino aufgebaut ist.

Montage und I²C-Bus

Die Montage auf dem Platino-Board ist anhand von Schaltung und Stückliste nicht schwierig. Den Anfang machen die Lötbrücken und Jumper JP1...JP14, in **Tabelle 1** sind sie zusammengefasst. JP11...JP13 werden nur benötigt, falls der ATmega328 mit einem AVR-ISP-Adapter programmiert werden soll. Wenn der Arduino-Bootlader benutzt wird, stellt ein FTDI-Adapterkabel mit USB-Seriell-Wandler, Spannung 5 V, die Verbindung mit K2 des Platino her. Das Adapterkabel ist auch zum Testen und Weiterentwickeln der Firmware nützlich. Während der Test- und Entwicklungsphase kann das Board über das Kabel aus dem USB-Port mit Strom versorgt werden.

Induktivität L1 hat nur den Zweck, die analoge Betriebsspannung des ATmega328 zu entkoppeln. Da hier vom A/D-Wandler des ATmega328 kein Gebrauch gemacht wird, ist L1 durch eine Drahtbrücke ersetzbar.

Tabelle 1. Lötbrücken und Jumper auf dem Platino-Board.

Jumper:		Jumper:	
JP1	PB4 (Tongeber)	JP8	reset
JP2	5V	JP9	XTAL2
JP3	PB5 (Beleuchtung)	JP10	XTAL1
JP4	-	JP11	PB7
JP5	PB1	JP12	PB6
JP6	PB2	JP13	PB5
JP7	PB3	JP14	-

Wie schon angemerkt, muss die I²C-Adresse eines Sensors geändert werden, denn beide Sensoren liegen am gleichen I²C-Bus. Zu diesem Zweck haben wir einen Sketch geschrieben, den Sie wie folgt einsetzen:

Verbinden Sie den CC2-eBoB mit dem Platino, die I²C-Anschlüsse sind die Pins 5 und 6 (SDA und SCL). Der Sensor-Anschluss V_{DD} wird vorübergehend an den digitalen Ausgang PB0 (Pin 1 von K5) des Platino gelegt. Das Ändern der I²C-Adresse ist nur kurzzeitig nach Einschalten der Sensor-Betriebsspannung möglich. Nach Verbinden mit dem Platino-Ausgang PB0 kann der Sketch die Sensor-Betriebsspannung selbstständig einschalten. Der Masseanschluss GND wird mit Pin 4 von K7 verbunden. Starten Sie die Arduino-Entwicklungsumgebung auf Ihrem PC und laden Sie den Sketch *CC2A_set*

I2C_address.ino hinein, Sie können ihn von der Projektseite des CC2-eBoB [2] herunterladen. In dem Sketch sind zwei Konstanten definiert:

1. CURRENT_I2C_ADDRESS = 0x28 (Standardwert, hier Innen-Sensor)
2. NEW_I2C_ADDRESS = 0x22 (hier Außen-Sensor)

Abweichende Adressen im Bereich 0x00...0x7F sind ebenfalls erlaubt, allerdings muss dann der Quellcode des Sketchs modifiziert werden. Im Quellcode sind die Adressen der Sensoren durch die Konstanten *Inside_sensor* und *Outside_sensor* definiert.

Verbinden Sie, falls noch nicht geschehen, das Platino-Board über das FTDI-Adapterkabel mit dem PC, kompilieren Sie *CC2A_set_I2C_address.ino* und laden Sie

Relative und absolute Luftfeuchte

Dem Kellerlüftungssystem muss die absolute Luftfeuchte bekannt sein, um den Lüfter und den Fensteröffner zu steuern. Die meisten Feuchte-Sensoren messen jedoch die relative Luftfeuchte. Die relative Luftfeuchte lässt sich in die absolute Luftfeuchte umrechnen, wenn mindestens die Temperatur bekannt ist. Zwar sind noch weitere Größen beteiligt, doch mit den folgenden Beziehungen ist eine gute Näherung möglich:

$$\text{Dampfdruck [hPa]} = F/100 \cdot 6,1078 \cdot 10^{((a \cdot T)/(b+T))}$$

$$\text{Absolute Luftfeuchte [g/m}^3\text{]} = 216,7 \cdot \text{Dampfdruck}/(273,15+T)$$

Hierbei sind:

$$a = 7,5 \text{ (Konstante)}$$

$$b = 237,5 \text{ (Konstante)}$$

$$T = \text{Temperatur in } ^\circ\text{C}$$

$$F = \text{Relative Luftfeuchte [\%]}$$

Quelle: www.wettermail.de/wetter/feuchte.html

In der Firmware werden diese Beziehungen in der Funktion *Calc_Abs_Hum* des Arduino-Sketchs angewendet. Die Ergebnisse anderer existierender Näherungsmethoden weichen nur unwesentlich ab. Für das Lüftungssystem gilt, dass „absolut“ eigentlich auch „relativ“ ist: Exakte Werte sind nicht elementar wichtig, solange eindeutig ist, ob die Luft im Keller oder die Luft im Freien die höhere Feuchtekonzentration aufweist.

Tabelle 2. Anschlüsse und Funktionen der Steuerplatine.

Arduino	ATmega328	Funktion
Fensteröffner:		
Digital 0	PD0	High beim Öffnen
Digital 1	PD1	High beim Schließen
Analog 0	PC0	Endschalter, Low in Endstellung <i>Fenster offen</i>
Analog1	PC1	Endschalter, Low in Endstellung <i>Fenster geschlossen</i>
LCD:		
Digital 2	PD2	RS
Digital 3	PD3	Enable
Digital 4	PD4	Data D4
Digital 5	PD5	Data D5
Digital 6	PD6	Data D6
Digital 7	PD7	Data D7
Analog 5	PB5	Beleuchtung Ein/Aus
Dreh-Encoder:		
Digital 9	PB1	Taster
Digital 10	PB2	Phase A
Digital 11	PB3	Phase B
Lüfter:		
Digital 8	PB0	Ein/Aus
Tongeber:		
Analog 4	PB4	Ein/Aus

die dem Platino das Erreichen der Endpositionen signalisieren.

Firmware

Das Lüftungssystem kann nun einem ersten Test unterzogen werden. Laden Sie den Sketch `_140432_Control_Unit.ino` von der Projektseite [3] herunter, laden Sie ihn in die Arduino-Entwicklungsumgebung und von dort in den Platino. Nach Einschalten des Lüftungssystems wird zuerst das Fenster geschlossen, der Lüfter bleibt abgeschaltet. Mit dem Dreh-Encoder können diverse Einstellungen vorgenommen werden, so dass sich das Lüftungssystem an örtliche Gegebenheiten und Erfordernisse anpassen lässt. Die Einstellungen werden im EEPROM des ATmega328 permanent gespeichert, sie bleiben auch nach einem Stromausfall erhalten. Bei der ersten Inbetriebnahme, wenn das EEPROM noch leer ist, stellt der ATmega328 vorgegebene Standardwerte für folgende Größen ein:

- Temperatur, bei der die Frostsicherung einsetzt. Der vorgegebene Wert ist +3 °C, er ist im Bereich +3 °C...+10 °C einstellbar.
- Mindestwert der relativen Luftfeuchte. Wird dieser Wert unterschritten, aktiviert das System den Lüfter. Der vorgegebene Wert ist 30 %, er ist im Bereich 20 %...80 % einstellbar.

Die Einstellungen lassen sich während des regulären Betriebs nach Drücken des Tasters im Dreh-Encoder ändern. Sobald das Menü mit den Einstellwerten im Display erscheint, schließt das Fenster und der Lüfter stoppt. Das Lüftungssystem ist jetzt inaktiv, es kann abgeschaltet werden, während das Fenster geschlossen ist. Das ist zum Beispiel dann sinnvoll, wenn das Fenster und der Fensterrahmen geputzt werden sollen. Nachdem die Einstellungen beendet sind, geht das System wieder in Betrieb.

Außer den genannten Einstellungen lässt sich das System auch an unterschiedliche Fensteröffner anpassen, im Handel sind diverse Typen und Ausführungen erhältlich. Die zum Lüftungssystem gehörende Elektromechnik kann natürlich auch in eigener Regie konstruiert und gebaut werden. Die Einstellungen für den Fensteröffner lassen sich nur ändern, wenn beim Einschalten des Systems der Dreh-Encoder-Taster gedrückt gehalten wird. Nachdem die Einstellungen beendet sind,

die Firmware in den Platino. Danach ist die I²C-Adresse geändert, so dass beide CC2-eBoBs parallel am gleichen I²C-Bus arbeiten können.

Verbinden Sie die Anschlüsse der Betriebsspannung vorläufig mit Pin 3 (+5 V) und Pin 4 (GND) von K7, die I²C-Leitungen SDA und SCL legen Sie an Pin 5 (SDA) und Pin 6 (SCL) von K6. Auf dem Erweiterungsboard, das zu diesem Projekt gehört und in einem nachfolgenden Beitrag beschrieben wird, sind eigene

Anschlüsse vorhanden. Auch die Treiberschaltungen für den Lüfter und den Fensteröffner werden Thema des nachfolgenden Beitrags sein. Einen Vorgeschmack geben **Bild 3** und **Tabelle 2**, aus ihnen geht hervor, welche Leitungen des Platinos den Systemfunktionen zugeordnet sind. Das Schalten des Lüfters (Ein/Aus) erfordert nur einen digitalen Ausgang. Der Fensteröffner belegt zwei Leitungen für den Motorantrieb und zwei weitere Leitungen für die (optionalen) Schalter,

Stückliste Platino

Widerstände (250 mW/5 %):

R2,R3 = 47 Ω
R5,R6,R7,R10,R12 = 10 k
R11 = 4k7
R13 = 100 Ω
P1 = Trimpoti 10 k

Kondensatoren:

C1,C2 = 22 p keramisch
C3,C5,C6 = 100 n
C4 = 10 n
C8 = 1 µ/50 V stehend, Raster 2,5 mm
C9 = 10 µ/50 V stehend, Raster 2,5 mm

Induktivität:

L1 = 10 µH oder Drahtbrücke

Halbleiter:

D2 = 1N5817 (Schottky-Diode)
T1 = BC547
IC2 = ATmega328 (programmiert mit Arduino-Bootlader, 100892-41)
IC3 = 7805

Außerdem:

LCD1 = LC-Display alphanumerisch, 4 · 20 Zeichen mit Beleuchtung (z. B. Midas MC4200B6W-SPR)
X1 = Quarz 16 MHz
S6 = Dreh-Encoder mit Drucktaster (z. B. Alps EC12E2424407)
BUZ1 = Akustischer Signalgeber 12 mm
Platino-Board 100892-1
2 Stück CC2-eBoB, Elektor Store 140154-91

startet der reguläre Betrieb. Die relativen Luftfeuchten und die Temperaturen werden kontinuierlich gemessen und auf dem LC-Display ausgegeben. Dort erscheinen auch die daraus errechneten Feuchtekonzentrationen.

Tritt der Fall auf, dass ein Sensor nicht mit einer gültigen Antwort reagiert, wenn der ATmega328 den Status über den I²C-Bus abfragt, wird auf dem LC-Display eine Fehlermeldung angezeigt. Aus der Fehlermeldung geht hervor, welcher Sensor die Störung verursacht. Außerdem macht ein akustisches Alarmsignal auf den Fehlerzustand aufmerksam.

Während des regulären Betriebs werden die Stellung des Fensteröffners und der Status des Lüfters (wenn nötig) im Abstand von zehn Minuten an die aktuellen Messwerte angepasst. Das LC-Display gibt, wie **Bild 4** zeigt, Auskunft über den Betriebszustand.

Zum Schluss

Wir haben bisher nur den Teil des Lüftungssystems betrachtet, der den Fensteröffner und den Lüfter abhängig von den Luftfeuchten auf der Ebene des Mikrocontrollers steuert. Ein folgender Beitrag wird die elektrische Anbindung des Lüfters und des Fensteröffners an die Steuerung zum Inhalt haben. Die mechanische Konstruktion des Fensteröffners wird besprochen, und außerdem veröffentlichen wir eine Platine, die das Bindeglied zum Platino darstellt.

Obwohl das vorgestellte Lüftungssystem seine Bewährungsprobe in der Praxis bestanden hat, kann das System keine Wunder vollbringen und muffige Kellerräume nicht in lauschige Salons verwandeln. Der Eintrag auf unserer Entwickler-Website elektor-labs.com [1] lässt erkennen, dass unser Leser noch damit beschäftigt ist, das System zu optimieren und auszubauen. Unser Lüftungssystem kann wirksame Hilfestellung leisten, um den Feuchtehaushalt in baulich intakten Kellerräumen zu verbessern. Bei Problemen mit der Durchlässigkeit der Kellerräume...

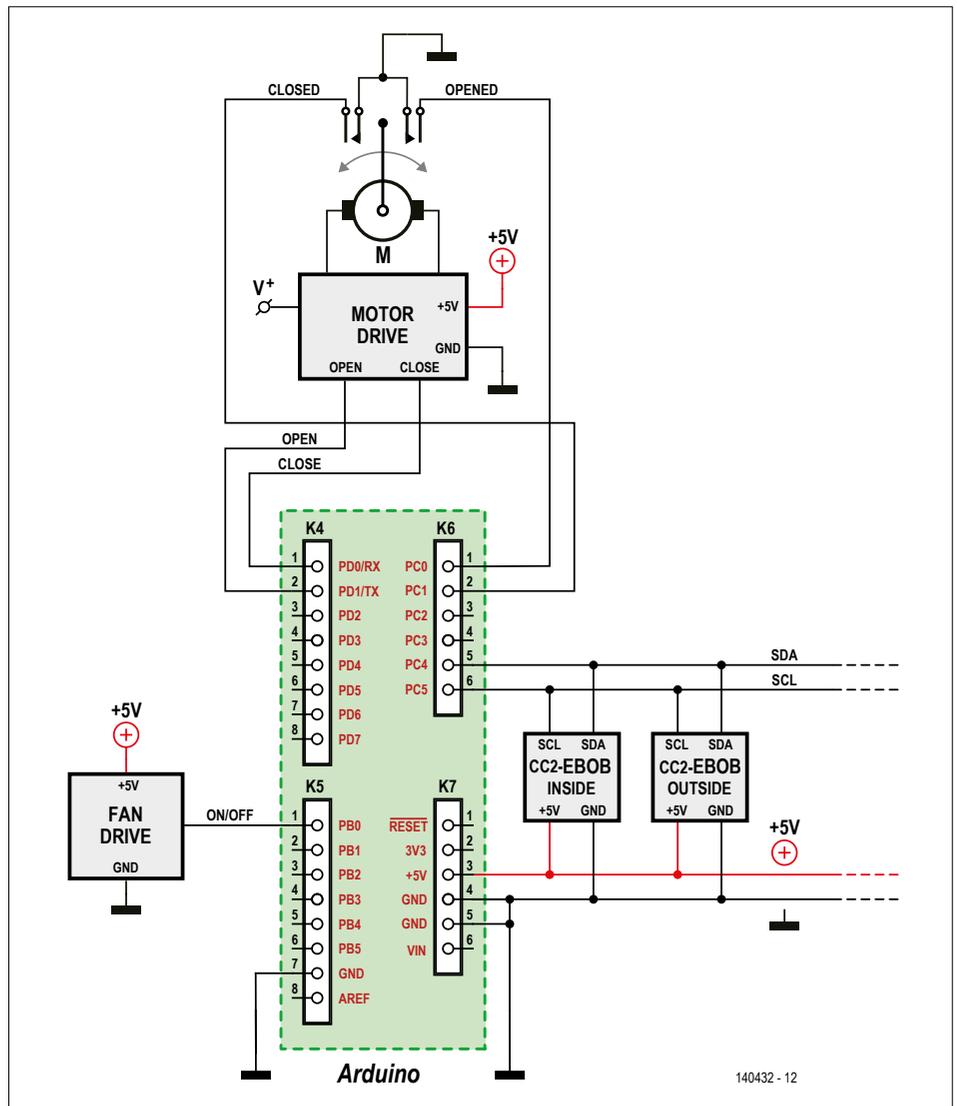


Bild 3. Belegung der Platino-Anschlüsse durch die Leitungen der externen Systemkomponenten.

lerwände oder des Fundaments müssen immer Baufachleute zu Rat gezogen werden, die sich in diesem Metier auskennen. Dies ist nicht nur zur Erhaltung der

Bausubstanz notwendig, sondern auch mit Blick auf die Gesundheit der Hausbewohner. ◀

(140432)gd

Weblinks

- [1] Leserentwurf: www.elektor-labs.com/project/feuchtegesteuerte-kellerlftung-humidity-basement-ventilation-140154.13770.html
- [2] Projektseite CC2-eBoB: www.elektor-magazine.de/140154
- [3] Projektseite zu diesem Beitrag: www.elektor-magazine.de/140432



Bild 4. Werte und Meldungen, die auf dem LC-Display ausgegeben werden.



Medikamenten-Alarm

Neun Erinnerungen täglich

Von Martien Schot (NL)

Wer regelmäßig ärztlich verordnete Medikamente nehmen muss, macht irgendwann die Erfahrung, dass die oft lebenswichtige Einnahme schon mal in Vergessenheit gerät. Gegen die Vergesslichkeit hilft diese Gedächtnisstütze, ein Alarmsystem, das flexible Einstellmöglichkeiten bietet und am gleichen Tag zu neun Zeiten Alarm schlagen kann.

Ein Wecker, wie er in wohl jedem Schlafzimmer steht, weckt den Schläfer meistens nur einmal am Tag. Wenn mehrere Medikamente zu unterschiedlichen Tageszeiten einzunehmen sind, ist ein solcher Wecker überfordert. Hinzu kommt, dass die meisten Wecker nicht an unterschiedlichen Wochentagen zu unterschiedlichen Zeiten Alarm geben können. Der Medikamenten-Wecker ist, etwas Lötfernung

vorausgesetzt, schnell aufgebaut. Ein Display zeigt Zeiten und Funktionen an, bedient wird der Wecker mit zwei Drucktastern. Die Uhrzeit muss nicht gestellt werden, denn der Zeitzeichensender DCF77 liefert sie frei Haus.

Hardware

Wie **Bild 1** zeigt, hält sich die Anzahl der Komponenten in Grenzen. Alarmzen-

trale ist ein Mikrocontroller ATmega328 (IC2) zusammen mit einem zweizeiligen Display, Typ LCD oder OLED. Beide Ausführungen sind Standard-Typen mit 2 · 16 Zeichen, die Abmessungen betragen 80 mm · 36 mm, die Anschlüsse befinden sich oben. Beim LCD-Typ wird der Kontrast mit Trimpoti P1 eingestellt, beim OLED-Typ entfällt diese Prozedur. Getaktet wird der Mikrocontroller

mit einem 16-MHz-Quarz.

Ein Tonendverstärker LM386 (IC3) verstärkt das Alarmsignal, das der Mikrocontroller generiert. Die Lautstärke lässt sich mit dem Trimpoti P2 einstellen. Am Endverstärker ist ein 8-Ω-Minilautsprecher angeschlossen, X2 stellt die Verbindung her.

Damit die interne Uhr des Alarmsystems weder vor- noch nachgeht, empfängt sie die Zeit vom Zeitzeichensender DCF77. X3 ist der Anschluss für ein handelsübliches DCF77-Empfangsmodul.

Die Zeiten und Funktionen werden mit den Drucktastern S1 und S2 eingestellt. S1 (Mode) ist für die Alarmzeiten zuständig, S2 (AL/Nav) hat drei Funktionen: Einstellen der Alarmzeiten (Navigieren), Abschalten eines laufenden Alarms und Aktivieren des DCF77-Monitors (siehe Firmware-Beschreibung).

Nach Drücken des Tasters S1 (Mode) erscheint in der unteren Display-Zeile das Alarmmenü. Die aktivierte Funktion blinkt, zuerst ist die Alarmnummer aktiv. Die nächsten Funktionen sind der Alarmtag, die Alarmzeit sowie die Änderungsanzeige, ein senkrechter Balken. Die Details gehen aus **Bild 2** und der Bedienungsanleitung hervor.

Ein Steckernetzteil, das die stabilisierte Gleichspannung 5 V liefert, versorgt das Alarmsystem mit Strom. Energiesparende Schaltnetzteile in Steckerform sind zurzeit im Handel preiswert erhältlich. Vor versehentlicher Verpolung schützt Diode D1.

Firmware

Die Firmware wurde mit einem Arduino Uno und der dazugehörigen IDE entwickelt. **Tabelle 1** gibt an, wie die I/O-Leitungen des Arduino den Anschlüssen des ATmega328 zugeordnet sind.

Die Firmware besteht aus folgenden, in der Schleife *void loop* aufgerufenen Modulen:

- *InsideClock*, für die interne Uhr und den Vergleich mit den Alarmzeiten.
- *DCF_receiver*, der Zeitzeichenempfang mit der Decodierung.
- *ModeButton*, über eine Kontaktentprellung wird die Funktion *Alarmedit* aufgerufen, der Alarm 1 wird in der unteren Display-Zeile angezeigt.
- *NavigateButton*, wird nach einer Kontaktentprellung aufgerufen, wenn die Funktion *Alarmedit* eingeschaltet ist.
- *AlarmHandler*, wird nur aufgerufen, wenn ein Alarm aktiv ist.

Eigenschaften

- Neun täglich programmierbare Alarmzeiten
- Als Alarmtag sind die Wochentage oder alle Tage wählbar
- Alarmzeiten lassen sich in Intervallen von 15 Minuten einstellen
- Alarmzeiten werden im EEPROM des ATmega328 gespeichert
- Alarme werden für 15 Minuten im Abstand von 3 Minuten wiederholt
- Gleichlauf mit der DCF77-Zeit durch handelsübliches Empfangsmodul
- Monitor-Funktion für das empfangene DCF77-Signal
- Zweifache Prüfung der empfangenen Zeit auf Gültigkeit
- Nachbaufreundlich, ausschließlich konventionelle Bauelemente

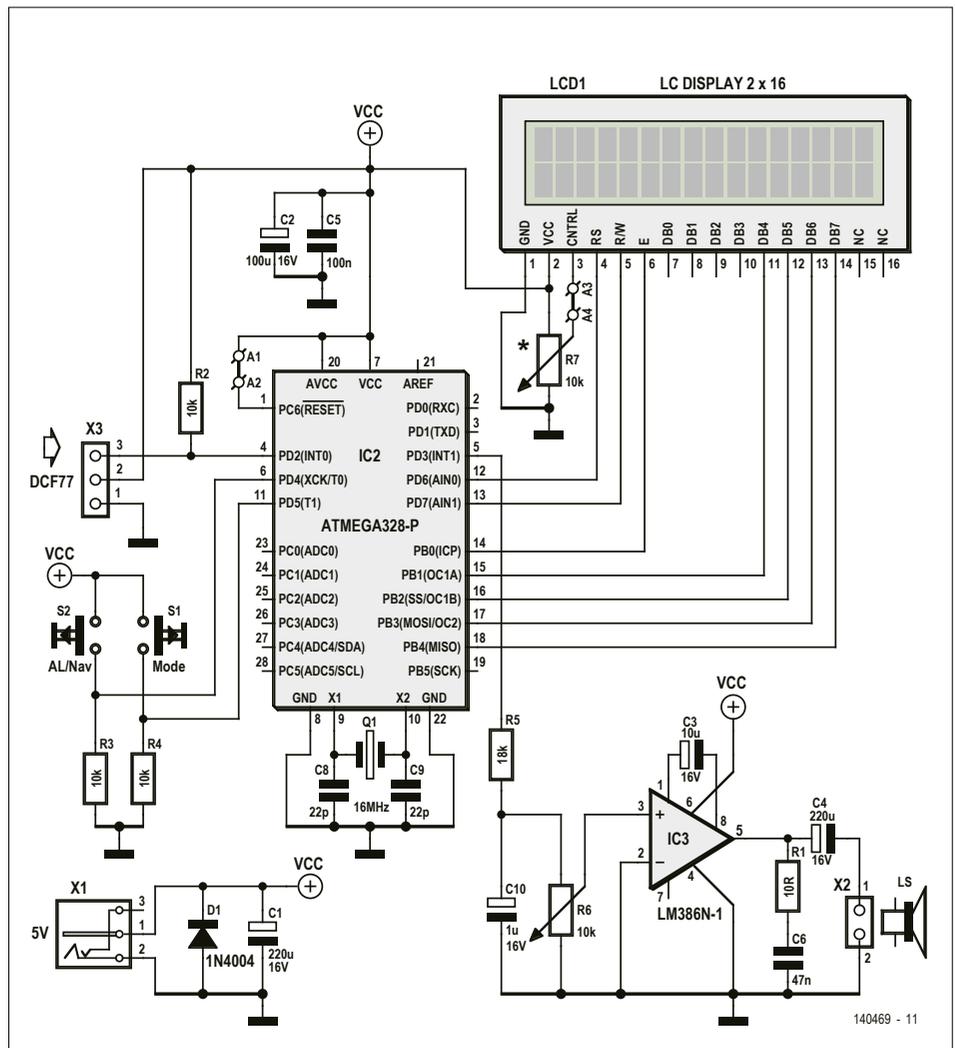


Bild 1. Alarmzentrale ist ein Mikrocontroller ATmega328, ein zweizeiliges Display und ein Tonendverstärker sind die Beigaben.

Listing 1.

```
int TotalMinutes = (WkDay*1440)+(Hour*60)+Minute;
if (TotalMinutes==ExpectTotal)CorrectReceive=true;
else CorrectReceive=false;
if (TotalMinutes==11519) ExpectTotal=1440;
else ExpectTotal= ++TotalMinutes ;
```



Bild 2. Nach Drücken des Tasters S1 erscheint in der unteren Display-Zeile das Alarm-Menü.

Bedienungsanweisung

Warten Sie nach dem Einschalten, bis das Display die korrekte Zeit anzeigt. Stellen Sie die Alarmzeiten wie folgt ein:

- Drücken Sie Taster *Mode* (S1). In der unteren Display-Zeile erscheint folgender Text:
#1 XXXXXX 00:00 |

#1 = Alarmnummer

Wählen Sie Alarm 1 bis 9 aus (aktiver Alarm blinkt).

XXXXXX Alarmtag (jetzt ausgeschaltet).

Hier sind wählbar: Immer, Montag, Dienstag, Mittwoch, Donnerstag, Freitag, Samstag, Sonntag und XXXXXX.

00:00 = Alarmzeit

Einstellen in drei Schritten: 10er Stunden, 1er Stunden sowie Minuten im Viertelstunden-Abstand.

| Vertikaler Balken: Zeigt an, dass die blinkende Alarmnummer mit Taster S2 geändert werden kann.

- Drücken Sie Taster *Mode* (S1) erneut. Der vertikale Balken geht in einen horizontalen Balken über. Jetzt können Sie mit Taster *AL/Nav* (S2) zum nächsten Feld XXXXXX springen, so dass diese Anzeige blinkt. Wenn Sie den Taster *Mode* (S1) noch einmal drücken, erscheint der Balken wieder in horizontaler Stellung. Sie können nun beispielsweise Immer wählen, was bedeutet, dass der Alarm täglich ausgeführt wird.
- Auf analoge Weise stellen Sie auch die Alarmzeit 00:00 ein.
- Am Zeilenende wird ein blinkender Unterstrich angezeigt. Wenn der eingegebene Alarmauftrag korrekt ist, können Sie ihn im EEPROM speichern. Dazu drücken Sie noch einmal den Taster *Mode* (S1). Anschließend verlässt das Alarmsystem den Eingabemodus.
- Falls Sie einen weiteren Alarmauftrag eingeben möchten oder der eingegebene Alarmauftrag nicht korrekt ist, drücken Sie anstelle des *Mode*-Tasters (S1) den Taster *AL/Nav* (S2). Anschließend können Sie den vorigen Alarmauftrag ändern oder einen neuen Alarmauftrag eingeben.
- Wenn Sie einen eingegebenen Alarmauftrag ändern möchten, geschieht dies auf die gleiche Weise.

Tasterfunktionen

- Taster *Mode* (S1) dient ausschließlich zum Einstellen der Alarmzeiten.
- Taster *AL/Nav* (S2) hat drei Funktionen:
- Navigieren, wie oben beschrieben.
- Abbrechen eines ausgelösten Alarms.
- Einschalten der DCF77-Monitor-Funktion.

InsideClock generiert die Sekunden durch die Funktion *millis()*, daraus werden die Minuten, Stunden und Wochentage abgeleitet.

DCF_receiver wandelt die Impulslängen des DCF-Signals in binäre logische Signale um und prüft die Parität. Aus dem binären Zeitsignal werden dezimale Werte gebildet. Sobald ein Zyklus vollständig ist, wird die zu erwartende Zeit des folgenden Zyklus berechnet. Die berechnete Zeit ist die Summe der Minuten, die seit Wochenbeginn (Montag 00:00 Uhr) vergangen sind. Aus **Listing 1** geht hervor, wie die Berechnung abläuft. Wenn die zuvor berechnete Summe mit der empfangenen Summe übereinstimmt, ist die Zeit gültig. Dann wird die interne Uhr synchronisiert, und die Zeit erscheint im Display in der unteren Zeile. **Bild 3** zeigt das Display mit der Anzeige von Zeit, Wochentag und Datum.

In *ModeButton* geschieht folgendes:

- *Alarmedit* wird eingeschaltet. Während *Alarmedit* läuft, wechseln ein horizontaler und ein vertikaler Balken in Zeichenposition 15 einander ab.
- *Alarmedit* wird abgeschaltet, der Balken in Zeichenposition 15 verschwindet.

NavigateButton ruft *Right* (horizontaler Balken) und *UP* (vertikaler Balken) auf. In **Listing 2** ist die Routine *Right* wiedergegeben.

Das zweidimensionale Array *Alarms[9][4]* speichert die Alarmdaten, es besteht aus neun Datensätzen mit vier Feldern. Im Listing sind dies Datensatz 0...8 und Feld 0...3.

Jeder Datensatz enthält ein vollständiges Alarmschema:

- Feld 0 = Alarmtag, 0 = Täglich, 1...7 = Montag...Sonntag, 8 = XXXXXX.
- Feld 1 = 10er Stunden, 0, 1 oder 2.
- Feld 2 = 1er Stunden, 0...9.
- Feld 3 = Viertelstunden, 0 = 00, 1 = 15, 2 = 30, 3 = 45.

Da die Felder 1, 2 oder 6 Zeichen enthalten, wird die Standard-Blinkfunktion für ein Zeichen im Display nicht genutzt. Stattdessen lässt eine selbst geschriebene Routine alle Zeichen gemeinsam blinken, siehe *voidCharBlink* in **Listing 3**.

Listing 2.

```

void RIGHT()
{++CurPosH;
 lcd.setCursor(CurPosH,1);
 switch (CurPosH)
 {case 0: CurPosH= 1; break; //goto 1=Alarmnumber
 case 2: lcd.setCursor(1,1);lcd.print(Ala_Num+1);
 CharakStr=AlarmDaySt[Alarms[Ala_Num][0]];CurPosH=3;break; // DOW
 case 4: lcd.setCursor(3,1);lcd.print(AlarmDaySt[Alarms[Ala_Num][0]]);
 CharakStr=String(Alarms[Ala_Num][1]);CurPosH=10; break; // ten minutes
 case 11:lcd.setCursor(10,1);lcd.print(String(Alarms[Ala_Num][1]));
 CharakStr=String(Alarms[Ala_Num][2]);CurPosH=11; break; //minutes
 case 12:lcd.setCursor(11,1);lcd.print(String(Alarms[Ala_Num][2]));
 CharakStr=QuarterStr[Alarms[Ala_Num][3]];CurPosH=13; break; //Quarters
 case 14:lcd.setCursor(13,1);lcd.print(QuarterStr[Alarms[Ala_Num][3]]);
 CharakStr=" ";CurPosH=15; break; //write data
 case 16:CurPosH= 1;CharakStr=String(Ala_Num+1); break;//goto Alarmnumber again
 }
 lcd.setCursor(CurPosH,1);
 }
}

```

Das Vergleichen der Alarmzeiten mit der aktuellen Zeit wird durch ein spezielles Format vereinfacht (**Listing 4**). Das Format ist nicht identisch mit dem Format, das zum Prüfen der empfangenen DCF77-Zeit benutzt wird. Beim hier verwendeten Format ist die aktuelle Zeit (*Real-Time*) gleich der Summe der Minuten, zu der die Stunden mit dem Faktor 100 und die Wochentage mit dem Faktor 10.000 addiert werden. Das Format der Summe einer Alarmzeit (*AlarmSum[i]*) ist gleich dem Format der aktuellen Zeit.

Die Wochentage liegen bei der aktuellen Zeit im Bereich 0...7, während die Wochentage der Alarmzeiten die Werte 0...8 annehmen können. Da der Wert des realen Wochentags nicht größer als 7 werden kann, wird der Alarm bei Wahl des Werts XXXXXX (8) in Feld 0 niemals ausgeführt.

Wird der Alarm-Modus Immer (0, täglich) gewählt, bleibt die Alarmsumme stets unter 2360 (24 Stunden). In diesem Fall wird der Wert des aktuellen Wochentags zur Alarmsumme addiert, so dass sie mit der Summe der aktuellen Zeit verglichen werden kann.

Der *AlarmHandler* wird aufgerufen, wenn in Listing 4 (unterste Zeile) *Alarm=true* ausgeführt wird. Jetzt zeigt die untere Display-Zeile den Alarm an. Aus dem Lautsprecher ertönt für 15 Minuten eine



Bild 3. Im Normalbetrieb zeigt das Display die aktuelle Zeit und das aktuelle Datum an. Die Software ist in vier Sprachen verfügbar.

Listing 3.

```

void CharBlink()
{if (millis()-LastOne > 400)// 0,4 sec
 {LastOne=millis();
 switch (CharakStr.length())
 {case 6:if(Charak==" ")Charak=CharakStr;
 else Charak=" ";// 6 characters days
 break;
 case 2:if(Charak==" ")Charak=CharakStr;
 else Charak=" ";// 2 characters quarters
 break;
 case 1:if(Charak==" ")Charak=CharakStr;
 else Charak=" ";// 1 character number, minutes
 break; // ten minutes
 }
 lcd.print(Charak);lcd.setCursor(CurPosH,1);
 }
}

```

Listing 4.

```

if (IntSec == 0 && AlarmEdit == false)
{RealTime=IntMin+(IntHour*100)+(IntWkDay*10000L);
  for (byte i = 0; i<9; i++)// 9 alarms
  { if(AlarmSum[i]< 2460) CompareSum=AlarmSum[i]+(IntWkDay*10000L);
    else CompareSum=AlarmSum[i];
    if (CompareSum==RealTime){Alarm=true;ParticuAla=i;TriggerTime=RealTime;}
  }
}
}

```

Melodie, sie wird in dieser Zeit im Drei-Minuten-Abstand wiederholt, bis Stopp-Taster S2 betätigt wird.

Für die Firmware wurde die EEPROM-Library des Arduino genutzt, die sich im

Libraries-Ordner des Arduino befindet. Wenn die Firmware zum ersten Mal startet, ist das EEPROM des ATmega328 noch leer, alle Speicherplätze enthalten 0xFF. In die ersten 36 Byte werden neun Alarme

geladen. In Feld 0 steht ausschließlich der Wert 8, so dass diese Alarme nicht ausgeführt werden. In die übrigen Felder wird der Wert 0 gesetzt (siehe Listing 5). Alle Alarme haben die gleiche Notation

Stückliste

Widerstände: (0,25 W):

R1 = 10 Ω
 R2...R4 = 10 k
 R5 = 18 k
 R6,R7 = Trimpoti 10 k, liegend
 A1-A2 = Drahtbrücke
 A3-A4 = Drahtbrücke

Kondensatoren:

C1,C4 = 220 μ /16 V stehend
 C2 = 100 μ /16 V stehend
 C3 = 10 μ /16 V stehend
 C5 = 100 n
 C6 = 47 n
 C7 = 10 n
 C8,C9 = 22 p
 C10 = 1 μ /16 V Tantal

Halbleiter:

IC2 = ATmega328P
 IC3 = LM386
 D1 = 1N4004

Außerdem:

Display = LCD 2 · 16 Zeichen oder OLED-Display (z. B. Winstar WEH01602 oder EA W162-X3LG)
 Q1 = Quarz 16 MHz
 X1 = Stromversorgungsbuchse, Mittelstift 2 mm, für Platinenmontage
 X2 = Kabelschraubklemme 2-polig, Raster 5,08 mm
 X3 = Kabelschraubklemme 3-polig, Raster 5,08 mm
 S1,S2 = Drucktaster (z. B. Conrad 700046)
 DCF-Empfangsmodul (z. B. Conrad 641138)

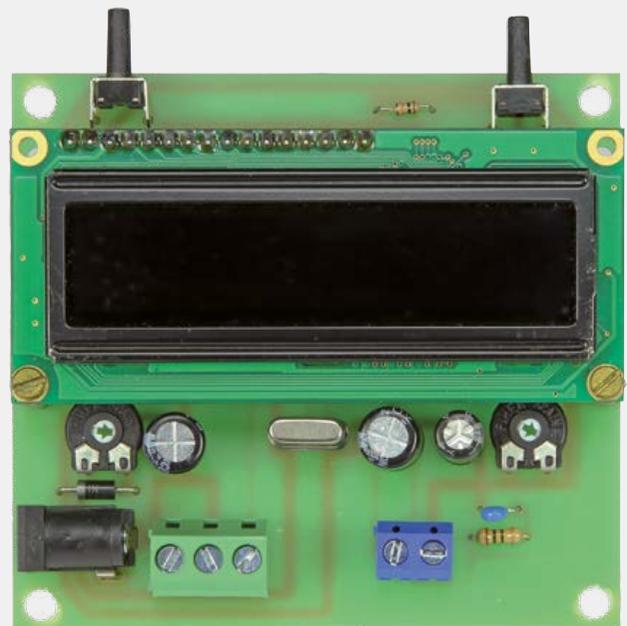
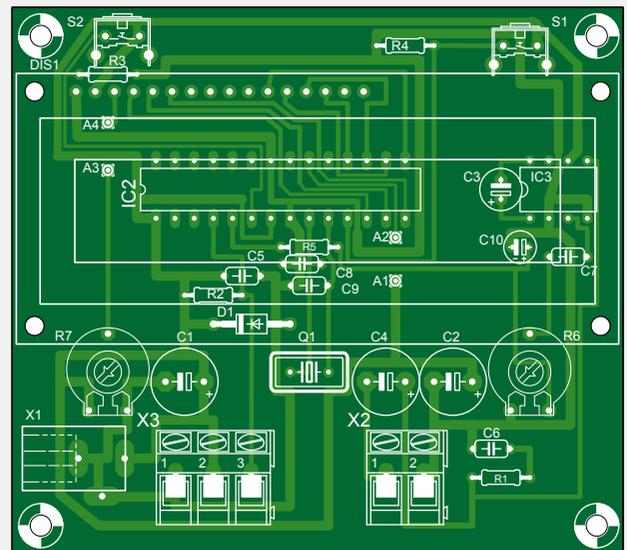


Bild 4. Auf der vom Autor entworfenen, einseitigen Platine befinden sich die Drahtbrücken A1-A2 und A3-A4.

wie Alarm 1: # 1 X X X X X X 00 : 00. Ferner ist in der Firmware ein *Monitor* implementiert, der Aufschluss über das empfangene DCF77-Signal gibt. Die Funktion startet, wenn Taster S2 (AL/Nav) betätigt wird. Auf dem Display erscheinen Informationen wie P13 0=100 ms oder P23 1=200 ms. Darin ist Pxx die fortlaufende Nummer des empfangenen Impulses, gefolgt vom logischen Wert (0 oder 1) und der Impulslänge in Millisekunden. Die Impulslänge kann Toleranzen bis 10 % aufweisen. Bei einem zweiten Druck auf S2 kehrt die Anzeige zum normalen Betrieb zurück.

Für die Display-Steuerung wird von der Library *LCD_OLED_FourBit* Gebrauch gemacht, sie steuert LCDs oder OLEDs. Bevor der ATmega328 programmiert werden kann, muss im Ordner *Libraries* ein Unterordner mit dem Namen *LCD_OLED_FourBit* angelegt werden. Darin müssen folgende Dateien stehen:

- LCD_OLED_FourBit.h
- LCD_OLED_FourBit.cpp
- keywords.txt

Diese Dateien können (zusammen mit der Software in vier Sprachen) von der zum Medikamenten-Alarm gehörenden Elektor-Projektseite [3] heruntergeladen werden. In dem noch anzulegenden Ordner *HelloLcdOledChar* wird die Datei *HelloLcdOledChar.ino* gespeichert.

Aufbau

Das Layout der Platine, die der Autor entworfen hat, ist in **Bild 4** wiedergegeben. Auf der Projektseite [3] steht das Layout zum Download bereit. Die einseitige Platine hat die Abmessungen 7,2 cm · 8,1 cm. Einige Bauelemente haben ihren Platz unter dem Display, hier ist bei der Montage die Höhe der Bauelemente zu beachten. Die Drucktaster sind Ausführungen mit hohen Bedienknöpfen. Wenn die Platine in einem Gehäuse unterge-

Listing 5.

```
byte Adress=0;//EEPROM init 36 bytes
if (EEPROM.read(Adress)== 0xFF)/(“empty=FF”)
{for (byte i = 0; i < 9; i++)
{for (byte j = 0; j < 4; j++)
{if (j == 0) EEPROM.write(Adress, 8);
else EEPROM.write(Adress, 0);
++Adress;
}
}
}
```

Tabelle 1. I/O-Leitungen des Arduino und des ATmega328

ARDUINO	6	7	8	9	10	11	12
ATmega328	12	13	14	15	16	17	18
ARDUINO DCF77PIN = 2	ATmega328-Pin = 4						
ARDUINO SPEAKPIN = 3	ATmega328-Pin = 5						
ARDUINO NAVIGBUT = 4	ATmega328-Pin = 6						
ARDUINO MODEBUTT = 5	ATmega328-Pin = 11						

bracht wird, können die Knöpfe durch die Oberseite ragen. Der Taster für die Funktion *Mode* hat eine niedrigere Höhe, um ein versehentliches Abschalten des Alarms zu verhindern.

Das Display wird über kurze Leitungen oder Steckverbinder-Leisten mit der Platine verbunden. Geeignet sind zweizeilige Standard-LCDs mit 2 · 16 Zeichen oder OLEDs im gleichen Format, zum Beispiel der Typ WEH01602 von Winstar. Die Firmware steuert gleichermaßen LEDs oder OLEDs, die Anschlüsse sind identisch. Für den Schallaustritt des Lautsprechers werden an einem passenden Ort im

Gehäuse einige Bohrungen angebracht, der Lautsprecher wird dahinter montiert und mit X2 auf der Platine verbunden. Das DCF77-Empfangsmodul muss für einen ungestörten Empfang in einem separaten Kleingehäuse untergebracht werden. Die Verbindung mit X3 auf der Platine stellt ein etwa 50 cm langes, dreidrahtiges Kabel her. Das Stromversorgungskabel, das zum Steckernetzteil führt, darf sich nicht in unmittelbarer Nähe des Empfangsmoduls befinden. Auch TV-Geräte, Computer und Leuchtstofflampen können Empfangstörungen verursachen. ◀

(140469)gd

Weblinks

- [1] Zeitzeichensignal DCF77: <http://de.wikipedia.org/wiki/DCF77>
- [2] Anschlussbelegung ATmega328: www.hobbytronics.co.uk/arduino-atmega328-pinout
- [3] Projektseite: www.elektor-magazine.de/140469

— Anzeige

SMD NAGLER

Ihr Spezialist für Bestückung von
Prototypen und Kleinserien

SMD-Nagler
Abraham-Wolf-Str. 42
70597 Stuttgart

Tel. 0711 12390019
Fax 0711 7653146
kontakt@smd-nagler.de
www.smd-nagler.de



PCB JOKER

**LEITERPLATTEN
DISCOUNT
FÜR PRAGMATIKER**

100% Made in Germany
www.pcb-joker.com
PCB Joker GmbH



LC Design

Ihr Partner
für
Displayapplikationen

LC Design
Johann-Knecht-Str.28
63785 Oberrburg

Tel. 06022 614430
Fax 06022 614431
Mail: info@lc-design.de
Website: www.lc-design.de

e-BoB BL600

(Teil 2)

Ein Programm editieren, kompilieren und übertragen mit dem BLE-Modul

Von **Jennifer Aubinais** (Paris) elektor@aubinais.net

Wenn man den Vorhersagen Glauben schenken darf, dann wird man in naher Zukunft nicht mehr nur vom *Internet der Dinge*, sondern vom *Internet aller Dinge* sprechen. Drahtlose Kommunikation und vernetzte Objekte mit geringer Stromaufnahme werden sich auf breiter Front durchsetzen. Dies ist genau der Zweck dieser Schaltung, die wie ein Mini-Computer eingesetzt werden kann und ganz einfach in smartBasic programmiert wird.

Im ersten Teil dieser Artikelserie über den e-BoB BL600 [1] haben wir die Hardware vorgestellt: auf der einen Seite den UART-Port, um mit anderen Objekten zu kommunizieren, hier ein PC. Auf der anderen Seite gibt es sieben Ein-/Ausgänge, die frei eingesetzt werden können: zwei ADC-Eingänge, der I²C- und der SPI-Port. Außerdem wurde gezeigt, wie man mittels der Funktion OTA (*Over The Air*) drahtlos ein erstes Programm von einem Android-Phone in unseren e-Bob lädt.



Jetzt folgt ein erstes Anwendungsprogramm. Damit machen wir uns vertraut mit den Tools zum Editieren, Kompilieren und Übertragen. Wir gehen in kleinen und kleinsten Schritten vor, um uns an die vielfältigen Möglichkeiten des Moduls heranzutasten und zu gewöhnen. Im dritten Artikel, im nächsten Heft, behandeln wir die Event-Programmierung, mit der dann größere Projekte erstellt werden können.

Editieren

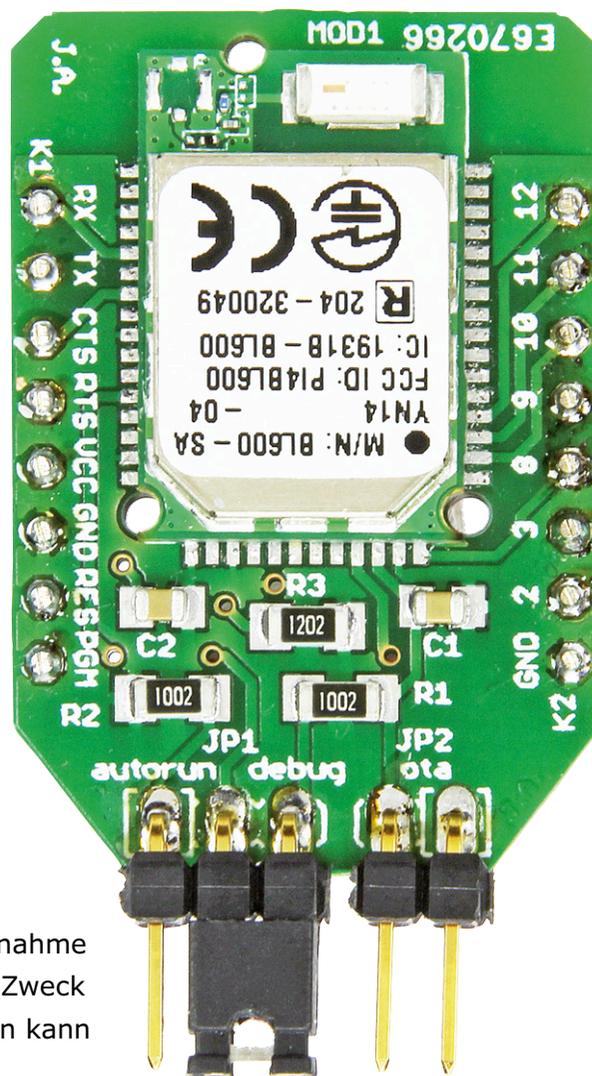
Wir beginnen mit dem Download der aktuellen Software für den BL600 (*Firmware Files version 1.5.70.0 – Revision 5* [2]) und eines Editors. Laird Technologies empfiehlt Notepad++ (**Bild 1**), es kennt die Syntax der Sprache smartBASIC.

Um sich im Abschnitt *SOFTWARE DOWNLOADS* der Laird Technologies Website zu registrieren, klicken Sie auf *If you need credentials, please click here* und gedulden Sie sich einige Tage. Laden Sie

nach Verbindungsaufbau *Firmware Files version 1.5.70.0 – Revision 5* q herunter und entpacken Sie den Ordner. Die Software enthält (**Bild 2**):

- Beispielprogramme im Verzeichnis *smartBASIC_Sample_Apps*
- das Programm *UwTerminal.exe* im Verzeichnis *smartBASIC_Sample_Apps*
- eine Programmbibliothek im Verzeichnis *smartBASIC_Sample_Apps/lib*
- weitere Beispiele (*UserManualExampleCode*)
- die spezifische Konfiguration von Notepad++ für smartBASIC (*smartBASIC(notepad++).xml*)

Starten Sie Notepad++ nach erfolgter Installation. Wählen Sie im Menü *language* (**Bild 4**) und klicken Sie auf *Define your language*. Klicken Sie auf *import*, wählen Sie die Datei *smartBASIC(notepad++)*



pad++).xml im ZIP-Ordner und starten Sie Notepad++ neu. Wählen Sie schließlich *language* (**Bild 5**), klicken Sie auf *User language*, wählen Sie *smartBasic* und schließen Sie das Programm.

Notepad++ ist jetzt so konfiguriert, dass die Syntax von smartBasic farbig dargestellt wird. Probieren Sie dies z.B. mit dem Programm *LedChaser.sb* (**Bild 6**) aus, auf das wir am Ende des Artikels zurückkommen. Wir werden aber nicht dieses Programm einsetzen, um im nächsten Absatz das Kompilieren zu lernen. Hierfür wird das Programm *upass* verwendet, das schon im ersten Artikel zum BL600 benutzt wurde.

Kompilieren

Bevor wir fortfahren sei noch einmal festgehalten, dass der e-BoB BL600 über ein UART/USB-Interface mit dem PC verbunden sein muss. Hierfür empfehle ich den e-BoB FT232, den ersten aus der Serie der Elektor-e-BoBs. Ein einfaches und sehr effizientes Seriell-USB-Interface, vorgestellt im Septemberheft 2011 [4]. Dieses Tandem aus den beiden e-BoBs wird in **Bild 14** und der nebenstehenden Abbildung gezeigt. Wir kommen auf diese Anordnung bei der Besprechung unserer Beispiel-Anwendung, dem Lauflicht, zurück. Wir schreiten nun zum Kompilieren. Hierfür benötigen wir auch noch das Programm *UwTerminal.exe* aus dem von Laird heruntergeladenen Paket.

Am Schluss des Artikels vom Vormonat verblieb unser e-BoB BL600 im *AutoRun*-Modus, in welchem er das vorher geladene UART-Programm ausführt. Zum Kompilieren und Laden muss sich Jumper JP1 in Stellung *cmd* befinden (in der Vorgängerversion des e-BoBs *debug* genannt), während JP2 (*ota*) nicht gesteckt ist. Initialisieren Sie das Modul (*reset*), sei es durch Tastendruck oder durch kurzes Unterbrechen der Spannungsversorgung. Starten Sie *UwTerminal.exe*, wählen Sie den passenden COM-Port des e-BoBs FT232 und eine Datenrate von 9600 Baud (**Bild 7**).

Drücken Sie *Enter* während der schwarze Bildschirm erscheint (**Bild 8**). Auch wenn es möglich ist, alles gleichzeitig zu tun, gehen wir doch schrittweise vor und beginnen zu kompilieren. Rechtsklicken Sie auf den schwarzen Hintergrund. Es öffnet sich ein Fenster. Wählen Sie *XCom-pile* (**Bild 9**) sowie die Datei *\$autorun\$.upass.vsp.sb*. Jetzt öffnet sich ein Com-



Bild 2. Diese Webseite von Laird wird Ihnen als Stammgast bald ein vertrauter Anblick sein.

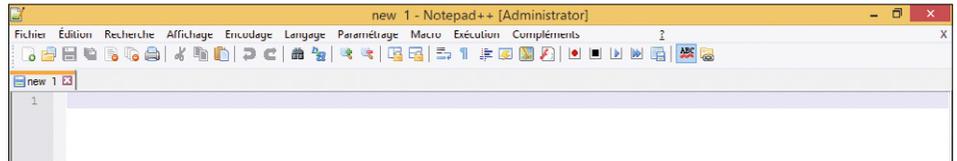


Bild 3. Der Editor Notepad++, mit dem die Apps geschrieben werden.

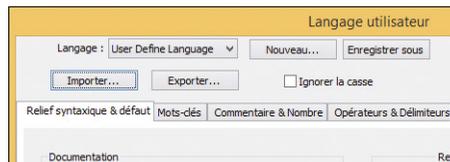


Bild 4. Damit Notepad++ die smartBASIC-Syntax lernt...

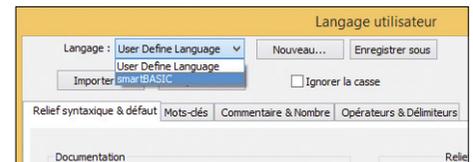


Bild 5. ...genügt es, die zugehörige Konfigurationsdatei von Laird zu laden.

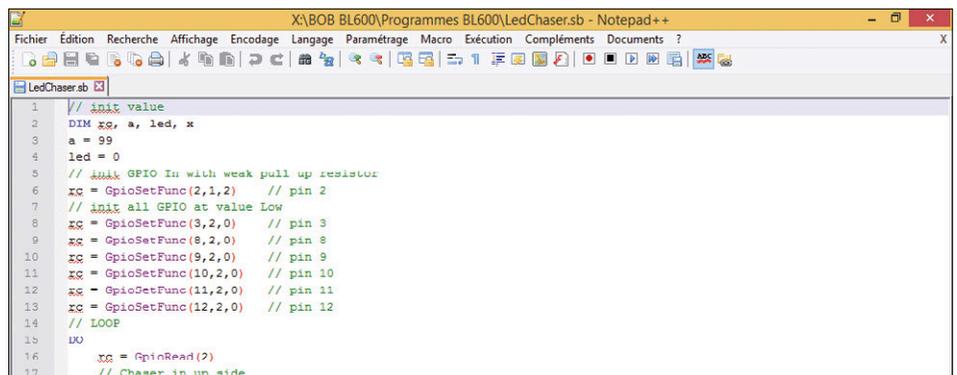


Bild 6. Das Editor-Fenster mit dem Code (komplettes Listing am Ende des Artikels).

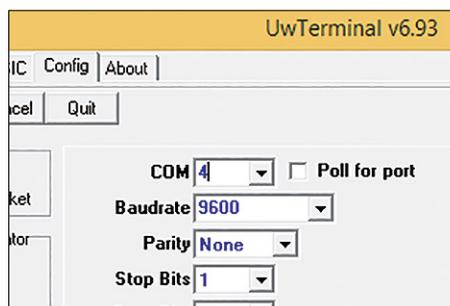


Bild 7. Einstellungen des Kommunikationsprogramms *UwTerminal*.



Bild 8. Die beiden 00 auf dem schwarzen Schirm von *UwTerminal* sind ein gutes Zeichen!

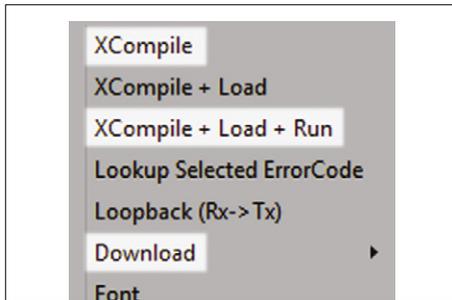


Bild 9. Menü Xcompile | Download | XCompile + Load + Run.

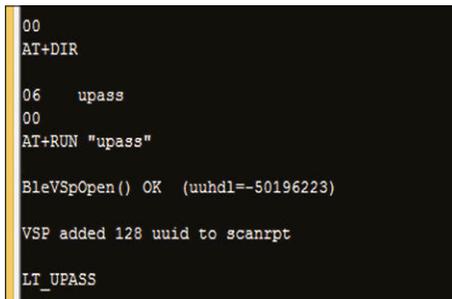


Bild 10. Das Ergebnis der Kompilierung wird im UwTerminal-Fenster angezeigt.

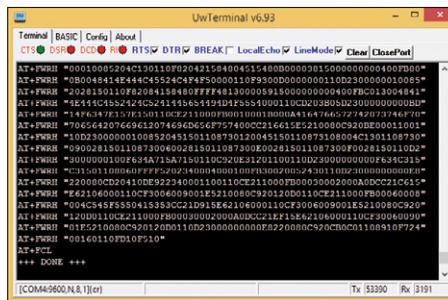


Bild 11. Die Übertragung des kompilierten Programms endet mit der Anzeige +++ DONE +++.



Bild 12. AT+DIR zeigt die Programmliste und AT+RUN startet Ihr Programm.

piler-Fenster und das Ergebnis der Übersetzung wird angezeigt (**Bild 10**). Es wird die kompilierte Datei \$autorun\$.upass.vsp.uwc erzeugt, welche auch zu den Downloads zum vorhergehenden Artikel gehörte [1]. Sie haben sie jetzt selbst neu erstellt und bald, wenn Sie mit der beschriebenen Vorgehensweise vertraut genug sind, können Sie sie nach Wunsch verändern.

Übertragen

Hier wird nicht BLE *Over The Air* eingesetzt, um das eben kompilierte Programm in den e-BoB BL600 zu übertragen. Dieser ist ja per UART mit dem PC verbunden, so geht es schneller. Außerdem ist dies für mich der beste Weg, die einzelnen Schritte aufzuzeigen, mit denen Sie eigene Programme schreiben können... und Ihre eigene Fehler machen können (so lernt man am besten). Jumper JP1 wird auf *cmd* gesteckt und JP2 (OTA) fehlt. Ist das Programm *UwTerminal.exe* auf dem PC gestartet, so genügt es, *ENTER* zu drücken, um zu prüfen, ob der e-BoB BL600 korrekt mit *00* auf dem schwarzen Schirm von *UwTerminal* antwortet. Falls nötig, muss der e-BOB BL600 initialisiert werden (*Reset*).

Falls Sie bereits ein Programm auf den BL600 geladen haben, wie z.B. im ersten Artikel beschrieben (*Over The Air*), dann muss jetzt der Programmspeicher durch Eingabe des Befehls *AT&F 1* gelöscht werden, der das Modul auch neu startet. Wählen Sie *Download* (**Bild 9**), dann *BASIC*, danach *Load Precompiled BASIC* und schließlich das kompilierte Programm *upass.vsp.uwc*. Die Übertragung des Programms endet mit der Message *+++ DONE +++* (**Bild 11**).

Geben Sie *AT+DIR* ein, um eine Liste der geladenen Programme zu erhalten. Zum Start des Programms geben Sie *AT+RUN "upass"* ein (**Bild 12**).

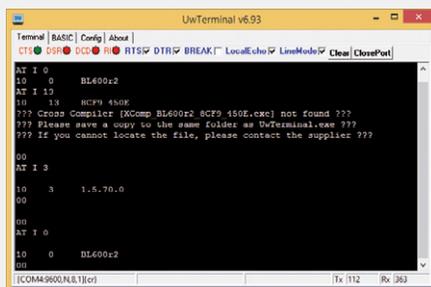
Wir haben *UwTerminal.exe* eingesetzt, um das Programm zu kompilieren, zu übertragen und auszuführen. Diese drei Schritte können in einem zusammengefasst werden, dann ist im Menü zu wählen: *XCompile + Load + Run* (**Bild 9**). Bisher benutzte Befehle:

- AT I 0: Revisionsnummer des BL600
- AT I 3: Programmversion des BL600
- AT+DIR: Liste der geladenen Programme im BL600
- ATZ: Reset des BL600

Die Möglichkeiten des Moduls sind vielfältig. Zum Eingewöhnen gehen wir in kleinen und kleinsten Schritten vor...

Error: Cross Compiler [XComp_BL600.....exe] not found ?

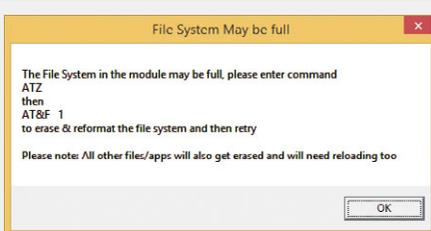
Dieser Fehler tritt auf, wenn das Programm (*Firmware*) Ihres BL600 nicht zu dem von der Laird-Seite heruntergeladenen Software-Paket passt. Mit diesen Befehlen stellt man die Version fest:



AT I 3 für die Firmware
AT I 0 für die Revision

Sobald Firmware und Version des BL600 identifiziert sind laden Sie das entsprechende Paket von der Seite von Laird Technologies [2]. Wiederholen Sie dann das oben beschriebene Kompilieren.

Error: The File System in the module may be full



Dieser Fehler tritt auf, wenn zu wenig Speicher für das neue Programm vorhanden ist. Löschen Sie den Speicher durch Eingabe der Befehle *ATZ* und *AT&F 1* und laden Sie das smartBASIC-Programm erneut.

- AT&F 1: Löschen des Speichers und Neustart des BL600
- AT+RUN "xxxx": Ausführen des Programms xxxx

Steuern des drahtlosen Lauflichts

Der BL600 steuert ein kleines Lauflicht aus 6 LEDs, das bei Knopfdruck die Laufrichtung ändert. Es werden einfache Logik-Ein-/Ausgänge genutzt, komplexere Fähigkeiten, die der BL600 aber beherrscht, wie AD-Wandlung oder Impulsbreitensteuerung, bleiben außen vor.

Die serielle Verbindung zwischen dem e-BoB FT232 und dem neuen e-BoB BL600 wurde bereits im letzten Artikel beschrieben. Zur Erinnerung: die Versorgungsspannung beträgt 3,3 V. Die Ausgänge des BL600 steuern die LEDs über je einen Vorwiderstand zur Strombegrenzung von 470 Ω an.

Bei der BASIC-Programmierung konzentrieren wir uns hier auf die Funktionen zur Ansteuerung der Universal-I/Os (GPIO), so wie im **Listing** dargestellt.

```
GPIOSETFUNC(nSigNum, nFunction,
nSubFunc)
```

Diese Funktion legt mit dem zweiten Argument einen mit dem ersten Argument definierten Pin als Ein- bzw. Ausgang fest.

a. Konfiguration als Logik-Eingang

Der Taster ist an Pin GPIO 2 angeschlossen und wird durch einen internen Pull-Up auf High gezogen:

```
rc = GpioSetFunc(2,1,2)
```

- **nSigNum** = 2: Pin GPIO 2
- **nFunction** = 1: Port als Eingang
- **nSubFunc** = 2: interner Pull-Up aktiv

b. Konfiguration als Logik-Ausgang

Jede LED wird über einen Strombegrenzungswiderstand an einen der Ausgänge (3, 8 bis 12) gelegt. Konfiguration von z.B. Pin 12:

```
rc = GpioSetFunc(12,2,0)
```

- **nSigNum** = 12: Pin GPIO 12
 - **nFunction** = 2: Port als Ausgang
 - **nSubFunc** = 0: Ausgang low
- rc** ist der Return-Code (= 0x0000 bei Erfolg).

```
GPIOREAD(nSigNum)
```

Lesen eines durch das Argument festgelegten GPIO-Eingangs. Der Taster liegt an Pin GPIO 2, konfiguriert als Eingang mit internem Pull-Up. Dieser Eingang wird

Listing LedChaser.sb

```
// init value
DIM rc, a, led, x
a = 99
led = 0
// init GPIO In with weak pull up resistor
rc = GpioSetFunc(2,1,2) // pin 2
// init all GPIO at value Low
rc = GpioSetFunc(3,2,0) // pin 3
rc = GpioSetFunc(8,2,0) // pin 8
rc = GpioSetFunc(9,2,0) // pin 9
rc = GpioSetFunc(10,2,0) // pin 10
rc = GpioSetFunc(11,2,0) // pin 11
rc = GpioSetFunc(12,2,0) // pin 12
// LOOP
DO
  rc = GpioRead(2)
  // Chaser in up side
  IF (rc == 1) THEN
    IF (led == 0) THEN : GpioWrite(12,0) : GpioWrite(3,1) : ENDIF
    IF (led == 1) THEN : GpioWrite(3,0) : GpioWrite(8,1) : ENDIF
    IF (led == 2) THEN : GpioWrite(8,0) : GpioWrite(9,1) : ENDIF
    IF (led == 3) THEN : GpioWrite(9,0) : GpioWrite(10,1) : ENDIF
    IF (led == 4) THEN : GpioWrite(10,0) : GpioWrite(11,1) : ENDIF
    IF (led == 5) THEN : GpioWrite(11,0) : GpioWrite(12,1) : ENDIF
    led = led + 1
  IF ( led == 6) THEN : led = 0 : ENDIF
ELSE
  // Chaser in down side
  IF (led == 6) THEN : GpioWrite(3,0) : GpioWrite(12,1) : ENDIF
  IF (led == 5) THEN : GpioWrite(12,0) : GpioWrite(11,1) : ENDIF
  IF (led == 4) THEN : GpioWrite(11,0) : GpioWrite(10,1) : ENDIF
  IF (led == 3) THEN : GpioWrite(10,0) : GpioWrite(9,1) : ENDIF
  IF (led == 2) THEN : GpioWrite(9,0) : GpioWrite(8,1) : ENDIF
  IF (led == 1) THEN : GpioWrite(8,0) : GpioWrite(3,1) : ENDIF
  led = led - 1
  IF ( led == 0) THEN : led = 6 : ENDIF
ENDIF
// tempo : speed
for x = 0 to 2000
  next
DOWHILE (a != 0)
```

Weblinks:

- [1] e-BoB BL600, Elektor März 2015, www.elektor-magazine.de/140270
- [2] https://laird-ews-support.desk.com/?b_id=1945
- [3] <http://notepad-plus-plus.org/>
- [4] www.elektor-magazine.de/110553
- [5] www.youtube.com/watch?v=SxwaVIOKkk8
- [6] Drahtloses Thermometer mit Bluetooth Low Energy, Elektor 1-2/2015, www.elektor-magazine.de/140190

wie folgt gelesen:

`rc = GpioRead(2)`

- `nSigNum = 2` : Pin GPIO 2
 - `rc` liefert den Wert an Pin 2: 0 oder 1
- Hinweis:** Falls der GPIO-Eingang als Analogeingang konfiguriert ist, liefert `rc` die Eingangsspannung an diesem Pin. Der AD-Wandler wird in einem nächsten Artikel behandelt.

`GPIOWRITE(nSigNum, nNewValue)`

Schreiben auf einen durch das erste Argument bestimmten GPIO-Ausgang (falls die Angabe ungültig ist geschieht nichts).

Pins 3 und 8 bis 12 sind als Ausgänge konfiguriert. Im Programm werden diese auf 0 bzw. 1 gesetzt:

`GpioWrite(12,0)`

- `nSigNum = 12` : Pin GPIO 12
 - `nNewValue = 0` : setze auf low
- Hinweis:** GPIO-Pins können auf Impulsbreitensteuerung konfiguriert werden (Pulse Width Modulation). Diese Funktion ist beim BL600 auf zwei Pins nach Wahl beschränkt. Der Wert von `nNewValue` liegt zwischen 0 und N, wobei N als Maximum zu einer Einschaltdauer von 100 % führt. In einem folgenden Artikel werden wir diese Möglichkeit des BL600-SA-Moduls nutzen und als Beispielanwendung ein kleines Fahrzeug über Bluetooth fernsteuern [5].

Am Ende des zweiten Artikels, der diesem kleinen Modul gewidmet ist, darf ich an dieser Stelle dem ganzen Team von Laird Technologies für die große Hilfe danken und hoffe, dass wir uns nächsten Monat wiederfinden, um den BL600 zu programmieren. ◀

(150014)

Themenauswahl für die nächsten Folgen dieser Serie rund um den e-Bob BL600:

- Handler/Events
- das Programm Red Green Blue
- Low Energy, 5 μ A
- der I²C-Port
- der SPI-Port
- Bluetooth-Kommunikation
- Erläuterungen zum Programm des Drahtlos-Thermometers
- Wie schreibt man ein Android-Programm
- Wie schreibt man ein iOS-Programm

(hmm... die Apple-Lizenz ist leider nicht umsonst)

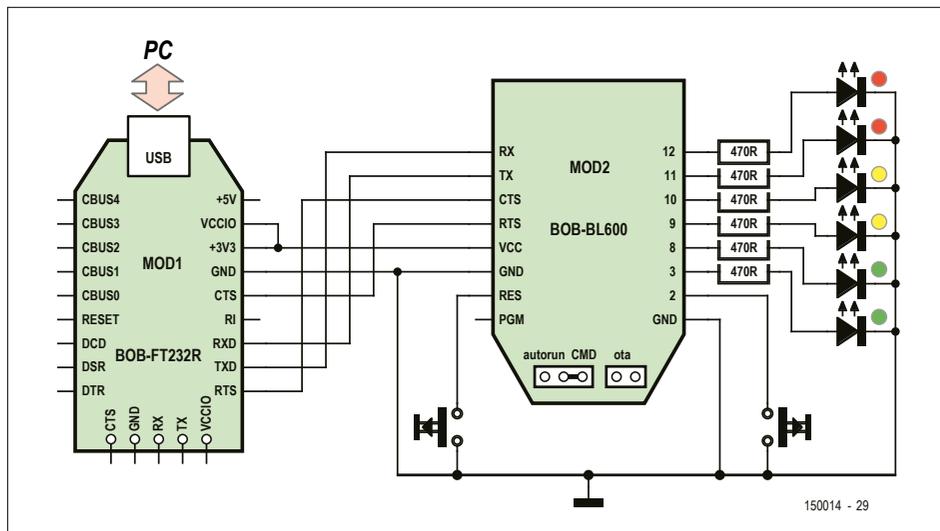


Bild 14. Der e-BoB BL600 kann autonom arbeiten, aber ich setze ein Tandem mit dem USB-Seriell-Adapter e-BoB FT232 ein, um ihn vom PC aus zu programmieren.

Stückliste

Widerstände :

R1 bis R6 = 470 Ω

Halbleiter:

D1 bis D6 = LED 3 mm (Farbe nach Wahl)

Verschiedenes:

K1, K2 = Taster

MOD1 = e-BoB FT232 bestückt 110553-91 (www.elektor.de)

MOD2 = e-BoB BL600 bestückt 140270-91 (www.elektor.de)

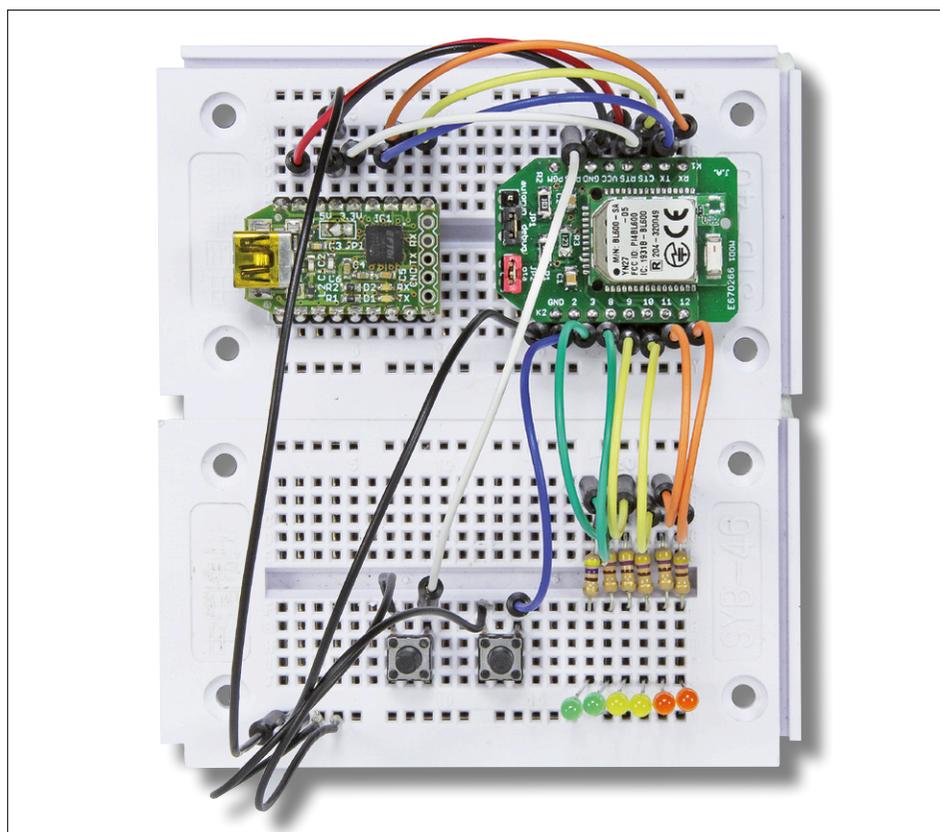


Bild 15. Das Lauflicht, aufgebaut auf einem Steckbrett.

AM-Sender mit Arduino

Induktiv auf Mittelwelle

Von
Burkhard Kainka
(D)

Alte Röhrenradios haben einen ganz besonderen Reiz. Sie sehen nicht nur schön aus, sondern klingen auch noch gut und wecken oft angenehme Erinnerungen. Deshalb gibt es viele Fans, die solche Radios sammeln, reparieren oder liebevoll restaurieren. Nur leider existieren immer weniger Rundfunkstationen, die noch auf den klassischen AM-Bereichen senden.

Bei mir im Arbeitszimmer steht so ein altes Röhrenradio von 1957, das auch noch regelmäßig eingeschaltet wird. Und zwar auf Mittelwelle. Die Amplitudenmodulation auf Mittelwelle hat zwar nur eine begrenzte Bandbreite, aber das bringt diesen besonderen, warmen Klang. Und am Abend hört man zahlreiche euro-

sind doch allgemein nicht gern gesehen. Es gibt aber eine legale Lösung. Man muss das Signal induktiv übertragen und hat dann eine geringe Reichweite; so läuft man kaum Gefahr, den Nachbarn zu stören. Da gibt es die „Allgemeinzuteilung von Frequenzen im Frequenzbereich 9 .. 30.000 kHz für induktive Funkan-

wendungen“ (Vfg. 4/2010 geändert mit Vfg. 4/2014). Für den Frequenzbereich 148,50 kHz bis 5000 kHz muss man im Abstand von 10 Metern einen Grenzwert von 15 dB μ A/m einhalten. wendet und den HF-Strom auf 0,3 mA begrenzt, dann beträgt die magnetische Feldstärke im Abstand 10 m nur noch -15 dB μ A/m. Damit ist die Sache legal. In der Praxis heißt das, dass die „Sendeanterie“ z.B. unter dem Tisch oder an der Wand in der Nähe des Radios befestigt wird. Innerhalb der Drahtschleife hat man guten Empfang, außerhalb nimmt die Feldstärke schnell ab. Man darf auch zwei Windungen aus derselben Drahtlänge bilden, dann ist die Feldstärke in einem kleineren Bereich noch größer und nimmt in der Ferne noch schneller ab.

► Nur leider existieren immer weniger Rundfunkstationen, die noch auf den klassischen AM-Bereichen senden

päische Stationen. Nur leider werden es immer weniger. Viele Ortssender haben schon abgeschaltet und sogar die BBC hat sich von der Mittelwelle verabschiedet. Das ruft geradezu nach einer neuen Nutzung der frei gewordenen Frequenzen. Ein eigener AM-Sender muss her! Damit kann ich nun BBC per Internetradio empfangen und auf Mittelwelle zu Gehör bringen, so wie in den alten Tagen.

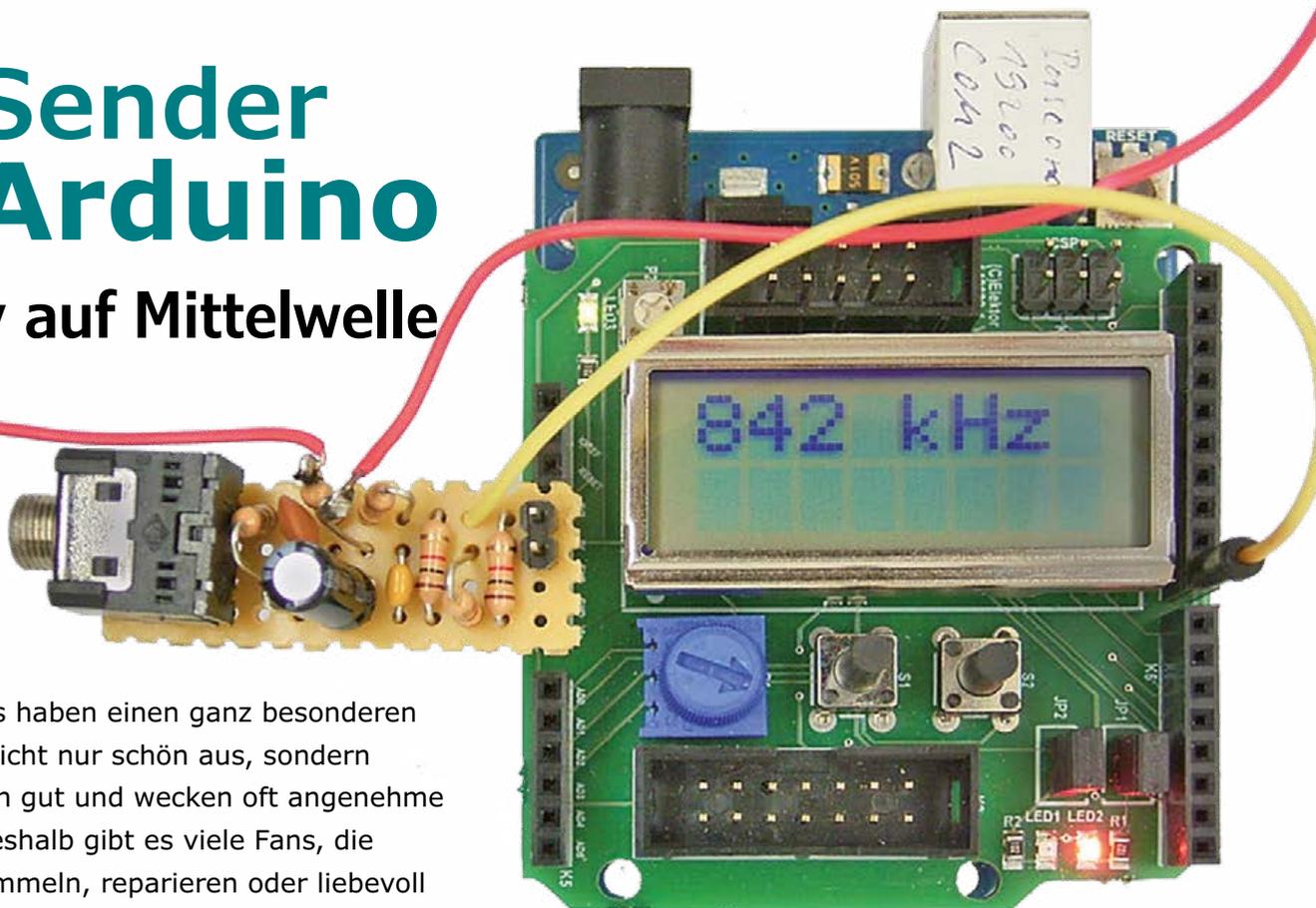
wendungen“ (Vfg. 4/2010 geändert mit Vfg. 4/2014). Für den Frequenzbereich 148,50 kHz bis 5000 kHz muss man im Abstand von 10 Metern einen Grenzwert von 15 dB μ A/m einhalten.

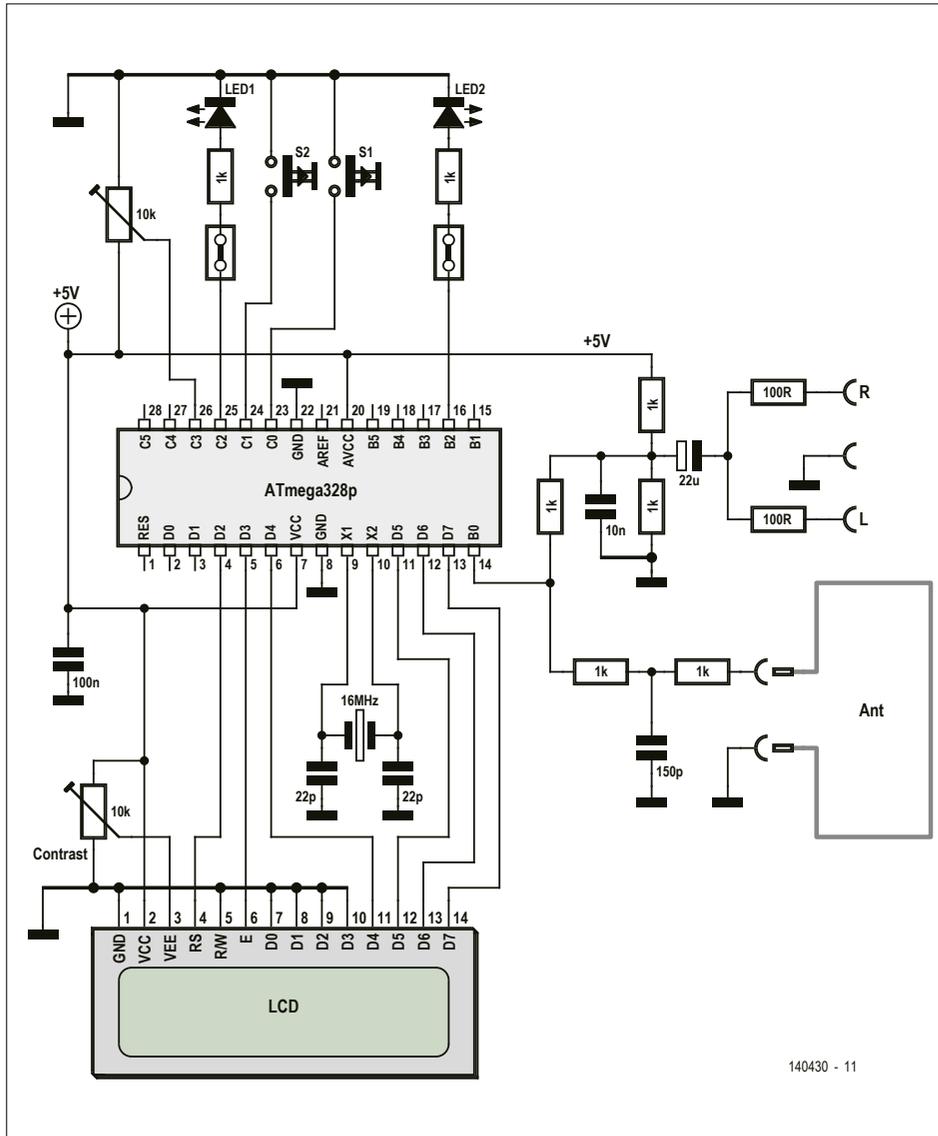
Die magnetische Feldstärke ist zwar nicht leicht zu messen, aber man kann sie aus dem Antennenstrom in einer induktiven Schleife berechnen: Wenn man eine Drahtschleife mit $r = 1$ m (also Durchmesser 2 m) und nur einer Windung ver-

Also einen AM-Sender bauen, wie ging das noch? Eine Senderöhre wurde mit dem Trägersignal voll durchgesteuert (Klasse-C-Endstufe), das NF-Signal modulierte deren Anodenspannung. Das funktioniert bei kleiner Sendeleistung auch mit einem Transistor, egal ob bipolar oder FET. Jetzt müsste man nur noch ein quarzstabiles HF-Signal passender Frequenz haben.

HF-Quelle Mikrocontroller

Und jetzt kommt der Mikrocontroller ins Spiel. Der Arduino Uno hat ja schon einen





Quarz, aus dem man durch programmierte Teiler Frequenzen im Mittelwellenbereich ableiten kann. Das ist eine einfache Lösung und hat den kleinen Nachteil, dass man nicht im üblichen 9-kHz-Raster (10 kHz in USA) sendet. Aber eine passende freie Frequenz wird sich trotzdem leicht finden lassen. Eine kleine Bedieneinheit mit Frequenzanzeige können wir mit dem Elektor Extension-Shield realisieren [1].

Und es kommt noch besser: Sogar die passende Sende-Endstufe hat der Uno schon an Board! Jeder beliebige Portpin kann dazu verwendet werden, wenn man einen kleinen Trick anwendet. Man steuert nicht das Portregister an, sondern das Datenrichtungsregister, zum Beispiel für den Anschluss PB0. Mit $PORTB.0 = 0$ und $DDRB.0 = 1$ (Ausgang) hat man einen niederohmigen Low-Zustand. Schaltet man dann $DDRB.0 = 0$ (normalerweise macht man das, wenn man digitale Signale einlesen will), wird der Port hochohmig. Diese beiden Zustände entsprechen genau einem FET mit offenem Drain. Wir können nun an PB0 eine beliebige „Drain“-Spannung anlegen (wenn diese nur im Bereich 0 V bis 5 V bleibt). Mit $DDRB.0 = 1$ wird diese auf Masse gezogen, etwa im Takt des HF-Signals. Und natürlich kann diese Drain-Spannung sogar moduliert werden, zum Beispiel mit einem NF-Signal.

Bild 1. Schaltplan des AM-Senders. Alle Bauteile auf der linken Seite befinden sich auf dem Arduino Uno und dem Elektor Extension-Shield.

Damit haben wir alles, was wir brauchen! Wir müssen nur noch ein paar Feinheiten erledigen und ein wenig programmieren.

Schaltung

Der Kopfhörerausgang einer typischen Signalquelle (CD-Player, PC-Soundkarte usw.) bringt üblicherweise bis zu 1 Veff, also fast 3 Vss. Vielmehr dürfte es ohnehin nicht sein, weil die Endstufe im Bereich 0...5 V angesteuert werden muss. Deshalb kommt man ganz ohne Modulationsverstärker aus. Ein paar Widerstände und Kondensatoren genügen, um das NF-Signal zu konditionieren (**Bild 1**, rechts). Der Schaltplan zeigt links den ATmega328 des Arduino Uno und die Bauteile, die man auf dem Extension-Shield findet. Natürlich kann man sich das Ganze auch auf einem Steckbrett aufbauen.

Am Antennenausgang gibt es ein einfaches Tiefpassfilter zur Dämpfung von

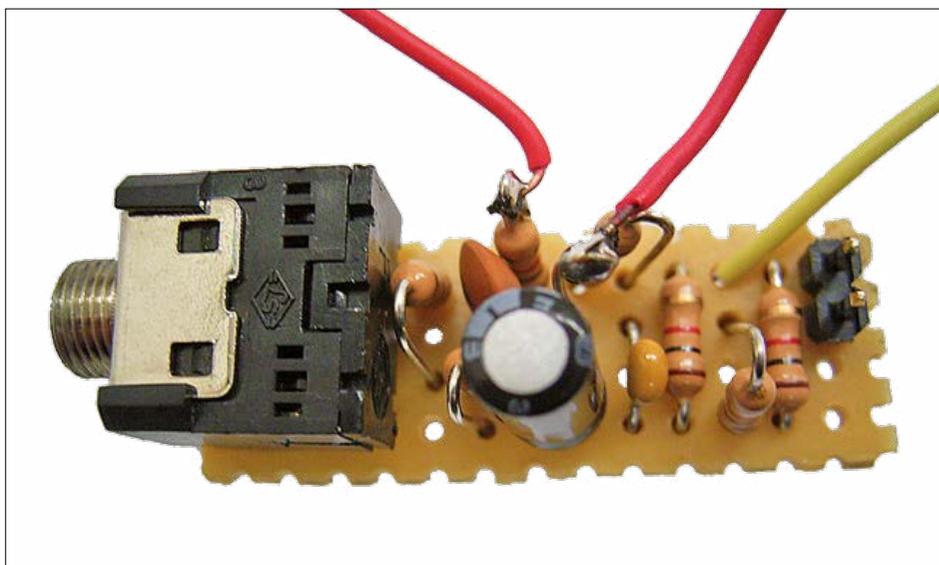


Bild 2. Die zusätzlichen Bauteile auf einer kleinen Lochrasterplatine.

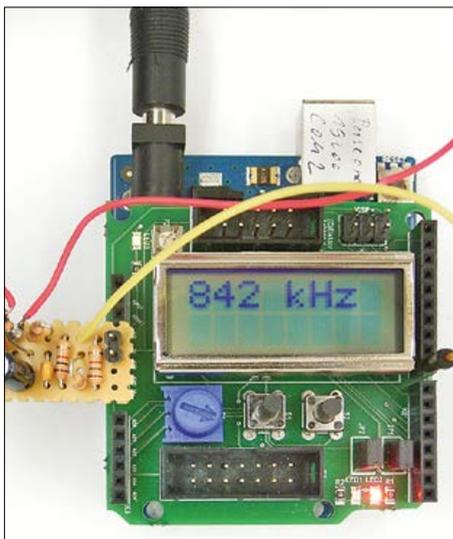
Listing 1. Erzeugung des HF-Signals.

```

D11:
Do      ' /11, 1454 kHz
  Ddrb.0 = 1
  nop   '+4 nop
  nop
  nop
  Ddrb.0 = 0
  nop
Loop

D10:
Do      ' /10, 1,6 MHz
  Ddrb.0 = 1
  nop   '+3 nop
  nop
  nop
  Ddrb.0 = 0
Loop

```



Oberwellen. Messungen und theoretische Überlegungen zeigen, dass etwas weniger als 0,3 mA durch die Antennenschleife fließen. Die Richtlinien werden damit sicher eingehalten, solange die Gesamtlänge des Drahtes für die Antennenschleife nicht größer als 3 m ist. Für die ersten Versuche reicht aber auch schon eine sehr viel kleinere Drahtschleife (mit z.B. nur 20 cm Draht).

Mit einem üblichen Soundkartenausgang (etwa 3 V_{ss}) ergibt sich ein Modulationsgrad von ca. 50 %, d.h. eine Übermodulation ist praktisch ausgeschlossen. Da das AM-Signal nur mono überträgt,

werden der linke und der rechte Kanal zusammengefasst. Insgesamt ergibt sich eine absolut saubere Modulation ohne irgendwelche Verzerrungen. Über ein Röhrenradio abgespielt erhält man damit genau den schönen warmen Klang aus alten Tagen.

Software

Um die Trägerfrequenz zu erzeugen, schalten wir einfach ganz schnell das entsprechende DDRB-Bit ein und aus. In einer einfachen Do-Loop-Schleife (**Listing 1**), die in Bascom programmiert wird, benötigt der ATmega328 sieben Taktzyklen. Drei zusätzliche NOPs, dann wird die Taktfrequenz von 16 MHz durch 10 geteilt. Damit sendet man am oberen Rand der Mittelwelle bei 1,6 MHz.

Fügt man einen weiteren NOP-Befehl ein, teilt die Schleife durch 11 und liefert 1454 kHz. Und so weiter. In unserem Programm, das man sich wie immer von der Elektor-Website [2] herunterladen kann, bekommt jede Frequenz ihre eigene Schleife; jeweils mit einem Label, das den jeweiligen Teilerfaktor ausdrückt. Die kleinste Frequenz von 500 kHz wird mit D32 erzeugt (16 MHz geteilt durch 32). Die NOPs sind übrigens so verteilt, dass ein möglichst gleiches Puls-/Pausenverhältnis entsteht. Ein symmetrisches Rechteck hat nämlich von Natur aus die wenigsten Oberwellen.

Insgesamt können 22 verschiedene Mittelwellenfrequenzen ausgewählt werden. Dabei liegen die Frequenzen am unteren Ende mit 500 kHz, 516 kHz, 533 kHz usw. näher beieinander als am oberen Ende.

Bedienung

Einmal in der HF-Schleife gibt es kein zurück. Zwar könnte man sich irgendeine Lösung mit einem Interrupt ausdenken. Einfacher ist aber folgender Bedienablauf (**Listing 2**): Zuerst wählt man mit dem Poti eine passende Frequenz aus, die auch im Display angezeigt wird. Dann drückt man auf S1 und startet damit den Sender. Zugleich wird auch die LED2 eingeschaltet und zeigt damit die Aktivität. Die zuletzt gewählte Frequenz bleibt einfach im Display stehen. Irgendwelche Änderungen am Poti wirken sich nicht mehr aus, die Frequenz steht stabil. Das ist genau wie bei den Röhren-Großsendern vergangener Tage, auch da konnte man nicht einfach im laufenden Betrieb an der Frequenz drehen. Will man doch noch die

Listing 2. Auswahl der Sendefrequenz.

```

Do
  U = Getadc(3)
  D = U / 46
  D = D + 10
  F = 16000 / D
  Locate 1 , 1
  Lcd F
  Lcd " kHz "
  Waitms 500
  If S1 = 0 Then
    Led2 = 1
    If D = 10 Then Goto D10
    If D = 11 Then Goto D11
    If D = 12 Then Goto D12
    If D = 13 Then Goto D13
    If D = 14 Then Goto D14
    If D = 15 Then Goto D15
    If D = 16 Then Goto D16
    If D = 17 Then Goto D17
    If D = 18 Then Goto D18
    If D = 19 Then Goto D19
    If D = 20 Then Goto D20
    If D = 21 Then Goto D21
    If D = 22 Then Goto D22
    If D = 23 Then Goto D23
    If D = 24 Then Goto D24
    If D = 25 Then Goto D25
    If D = 26 Then Goto D26
    If D = 27 Then Goto D27
    If D = 28 Then Goto D28
    If D = 29 Then Goto D29
    If D = 30 Then Goto D30
    If D = 31 Then Goto D31
    If D = 32 Then Goto D32
  End If
Loop

```

Frequenz wechseln, drückt man zuerst auf Reset und stoppt damit den Sender. Nun kann eine neue Frequenz eingestellt und mit S1 aktiviert werden. ◀

(140430)

Weblinks

- [1] www.elektor-magazine.de/140009
- [2] www.elektor-magazine.de/140430

Serieller USB-nach-Multi-Protokoll-Konverter

Für UART, SPI, I²C, ...!

Von **Ton Giesberts** und **Clemens Valens** (Elektor Labs)

Oft muss ein Computer mit einem Gerät über eine bestimmte Art einer seriellen Schnittstelle kommunizieren, sei es TTL-UART, RS232, RS485, I²C oder SPI. Wie bekommen wir das hin, wenn alles, was der PC zur Verfügung hat, eine Reihe von USB-Anschlüssen ist?

Die Verwendung der seriellen Schnittstelle (**Kasten** „Was ist überhaupt eine serielle Schnittstelle?“) hat in Elektor eine lange Tradition. Die Schaltungslösungen, Projekte und Workarounds in Elektor sind Legion wie der **Kasten** „Eine lange Geschichte“ beweist. Sie basieren alle auf bestimmten USB-zu-Seriell-Wandlern. Aber was ist mit den anderen Arten von seriellen Ports?

Elektroniker treffen immer häufiger auf Komponenten mit I²C- und SPI-Schnittstellen. Dies sind zwar auch serielle Ports und es scheint logisch, dass ein USB-Wandler auch diese Kommunikationsstandards unterstützen sollte. Und tatsächlich, unser Serial-BoB [6] kann dies dank des Bit-bang-Modus des FT232R, doch das Protokoll muss dann von der Software ausgeführt werden, was den Lauf der Dinge doch stark verlangsamt. Aber es gibt eine bessere und einfachere Lösung, um das Ziel zu erreichen.

FT232H

Eine Lösung bietet der FT232H. Obwohl die Typenbezeichnung der des FT232R ähnelt, ist der H-Chip im Inneren völlig unterschiedlich aufgebaut. Und obwohl beide USB 2.0-kompatibel sind, ist beim R-Chip nur „Full Speed“ mit 12 MBit/s möglich, während der H-Chip „High-Speed“ mit 480 Mbit/s erlaubt. Weiterhin ist der R-Typ ein USB-UART-Wandler mit

einigen zusätzlichen Bit-Bang-Optionen, während der H-Typ als *Hi-Speed USB to Multipurpose UART/FIFO IC* bezeichnet wird. Folglich ist der H-Typ nicht nur ein UART, sondern kann auch I²C und SPI sowie als i-Tüpfelchen auch als parallele Schnittstelle fungieren. All dies wird durch die so genannte *Multi-Protocol Synchronous Serial Engine* (MPSSE) ermöglicht.

MPSSE

Nach der Hersteller-Dokumentation bietet MPSSE eine flexible Möglichkeit, synchrone serielle Geräte mit einem USB-Anschluss zu verbinden. Beachten Sie das Wort synchron! Synchrone serielle Kommunikationsprotokolle verwenden mindestens zwei Leitungen (einen Bus), eine für Daten und eine für das Taktsignal. In solchen Systemen sorgt ein Gerät, in der Regel der Master, für den Takt. Der Empfang und die Übertragung von Datenbits bei den am Bus angeschlossenen Geräten werden von diesem Takt synchronisiert. Dieser Ansatz vermeidet die Notwendigkeit eines hochgenauen Timings, weil alle Geräte nach dem gleichen Takt tanzen. In asynchronen Kommunikationssystemen hingegen wird das Taktsignal mit den Daten kombiniert, so dass nur eine Busleitung erforderlich ist. Allerdings erfordert die Extraktion des Takts aus dem Signal in jeder Richtung ein präzises Timing. Um eine Analogie zu verwenden: In einem

synchrone System hören alle Teilnehmer auf die Kirchturmuhr des Dorfes, in asynchronen Systemen hat jeder seine eigene Armbanduhr. Protokolle wie SPI, I²C und JTAG sind Beispiele für synchrone Kommunikation, die serielle Schnittstelle einer MCU und seine UARTs arbeiten dagegen asynchron.

Da MPSSE viele Protokolle erlaubt, ist die synchrone Kommunikation mit vielen Geräten via SPI, I²C und JTAG möglich, um nur die beliebtesten zu nennen. Datenformatierung und Taktsynchronisation sind so flexibel konfigurierbar, dass fast jede mögliche Anforderung mit Geschwindigkeiten von bis zu 30 Mbit/s erfüllt werden kann. Und obwohl es die Bezeichnung MPSSE nicht explizit erwähnt, wird auch die asynchrone serielle Kommunikation unterstützt, allerdings nur mit einem Protokoll.

Das MPSSE-Modul des FT232H verfügt nicht nur über serielle Kommunikationsprotokolle, es kann auch in einer Vielzahl von Variationen parallel synchron kommunizieren. Eine ist der sogenannte FT1248-Modus, ein Interface mit einem 1, 2, 4 oder 8 bit breiten bidirektionalen Datenbus. Auf den ersten Blick scheint die MPSSE auch den berühmten (manche sagen, berühmten) Multiplex-Adress- und Datenbus der 8048/8051-Controller emulieren zu können, aber Informatio-

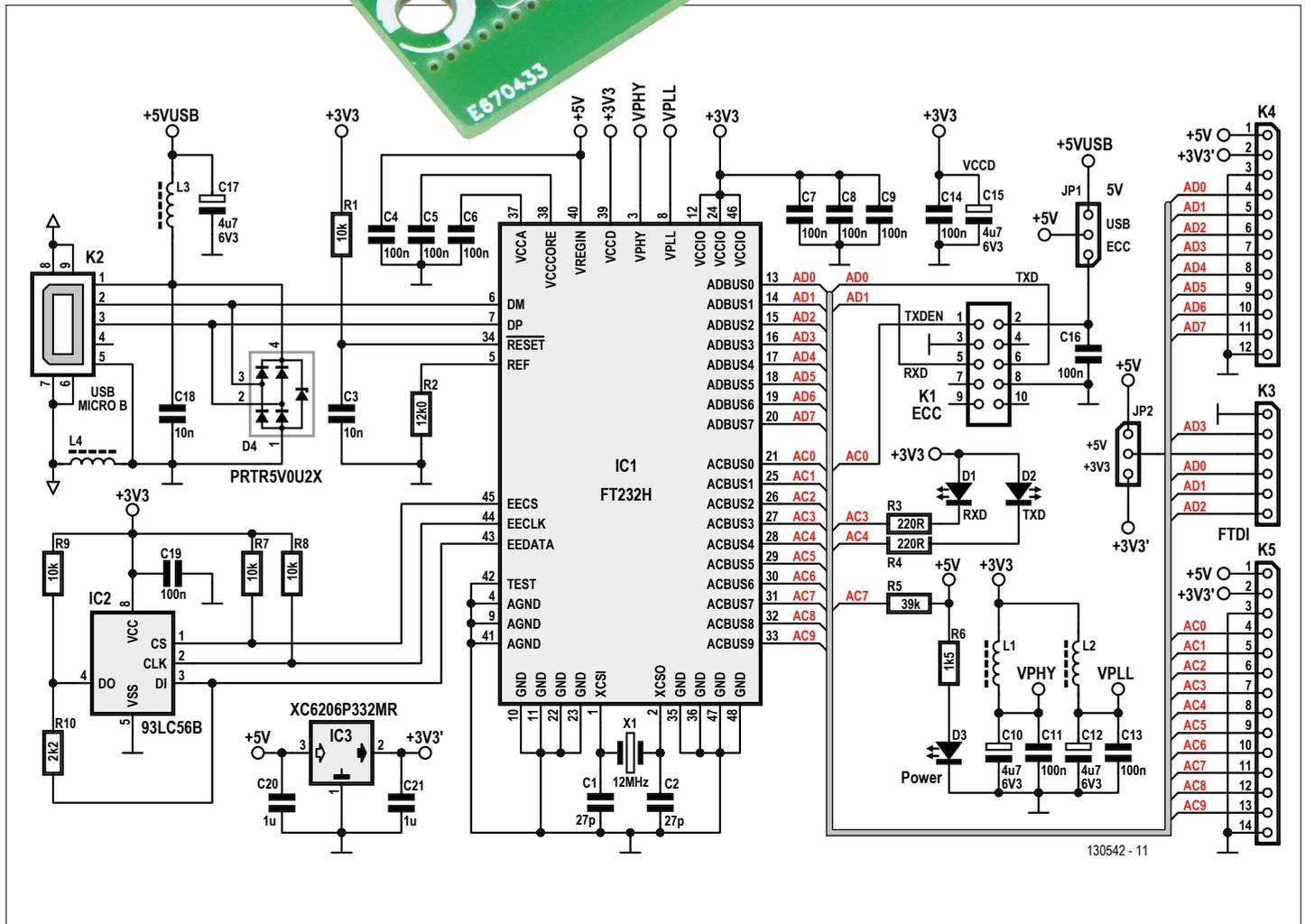
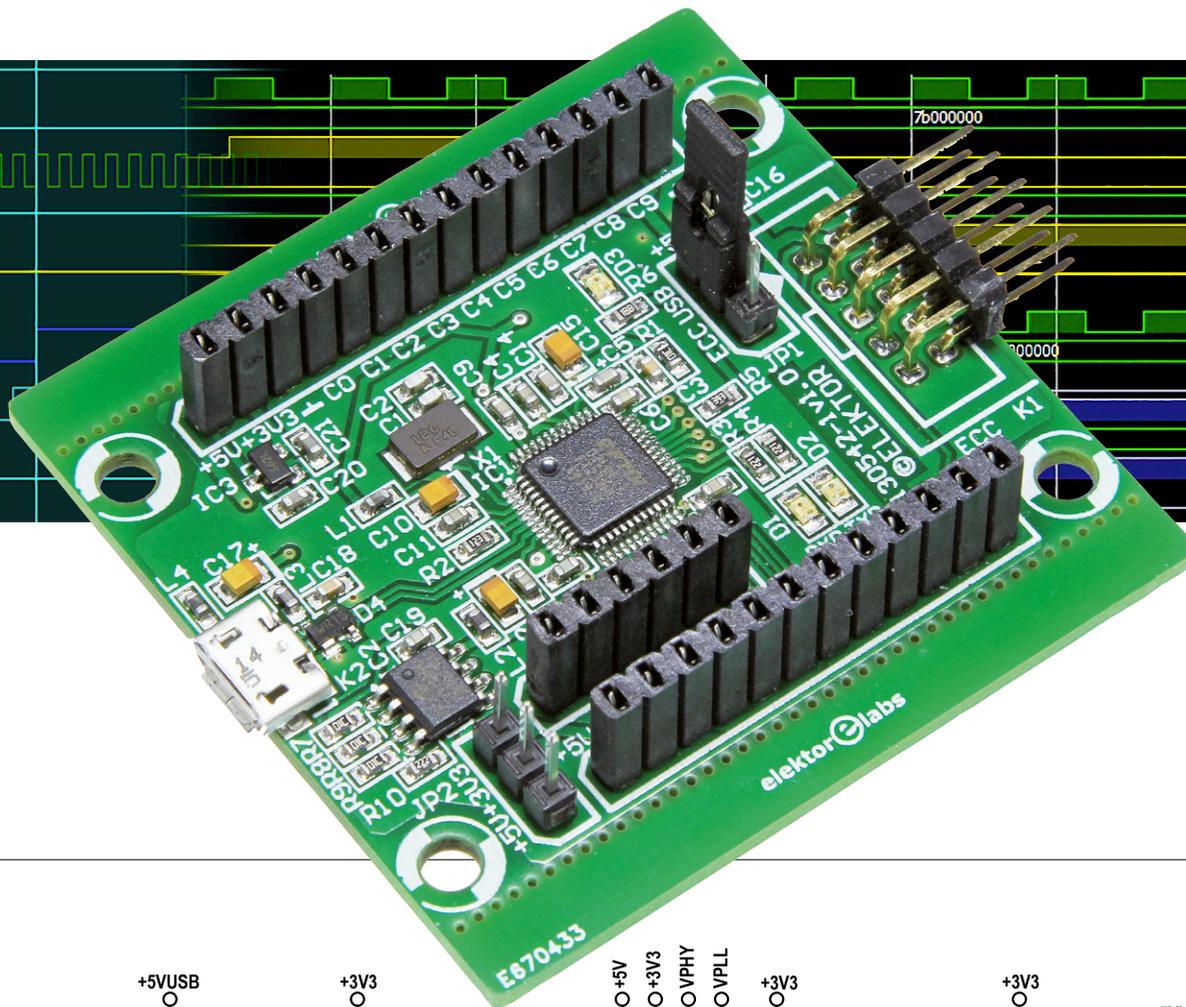


Bild 1. Schaltung des FT232H-Boards.

Nur FTDI?

In diesem Artikel erwähnen wir nur FTDI-Produkte. Als USB-to-Serial-Konverter werden in Elektor meist Produkte von FTDI eingesetzt. Stellt denn nur FTDI solche ICs her? Keineswegs! Mehrere andere Halbleiterhersteller produzieren ebenfalls USB-Interface-Chips. Hier ein paar Beispiele:

- Der berühmte PL2303 von Prolific.
- Cypress war immer sehr aktiv im Umfeld des USB. Der FX2LP ist sehr bekannt, der CY7C65211 ähnelt dem FT232H, ist aber nur für Full-Speed-USB geeignet.

- Texas Instruments hat als weltweit größter Halbleiterhersteller natürlich auch einige Produkte in petto. Bekannt ist der TUSB3410 mit einem 8052 im Inneren.
- Nanjing QinHeng Electronics Co. aus China produziert verschiedene USB-Interface-Chips. Der CH340 ist in extrem billigen EEPROM-Programmierern aus dem Internet zu finden.
- Microchip bietet den MCP2200 an, einen USB-zu-Seriell-Konverter. Es scheint, dass es sich tatsächlich um einen PIC18F14K50 beim Ausführen eines USB-Stacks handelt.

nen darüber sind nicht leicht zu finden. FT245-artige synchrone und asynchrone FIFO-Modi sind ebenfalls denkbar (der FT245 ist ein Wandler des gleichen Herstellers; von USB zu einer parallelen FIFO-Schnittstelle). Neben diesen ganzen seriellen und parallelen Protokollen sind zusätzliche GPIO-Signale verfügbar und auch „freihändiges“ Bit-banging wird unterstützt.

Derzeit gibt es die MPSSE in den vier FTDI-ICs FT2232D, FT2232H, FT4232H (die erste Ziffer gibt jeweils die Anzahl der UARTs an) und FT232H (ein UART). Beim FT232H bietet das MPSSE-Modul die meisten Protokolle.

In die Schaltung

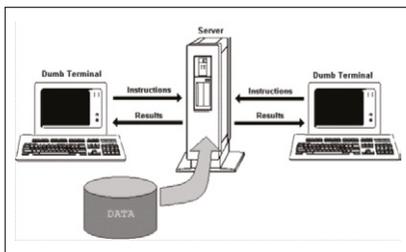
Wir haben uns deshalb entschieden, eine Mehrzweck-Platine für den FT232H zu entwerfen. Die ausgesprochen einfache Schaltung ist in **Bild 1** zu sehen. Die meisten Bauteile sind Kondensatoren und Ferrite mit der Aufgabe, Signale und Spannungen zu filtern.

Es handelt sich um eine Art von Break-Out-Board (BoB), da alle interessanten Anschlüsse auf die Buchsenleisten K4 und K5 herausgeführt sind. Daneben gibt es noch zwei weitere Anschlüsse auf dem Board: K1 ist der *Embedded Communication Connector* (ECC), der auch auf einigen anderen Boards von Elektor vor-

handen ist. Steckt man den Jumper an JP1 in Position ECC, so kann das Board über K1 versorgt werden, wenn Sie den USB-Port nicht weiter belasten wollen. K3 ist wie ein FTDI-Kabel belegt, so dass das FT232H-Board auch ein solches Kabel ersetzen kann. Sie sollten sich aber bewusst sein, dass Sie es an allen relevanten Verbindern mit einem Signalpegel von 3,3 V und nicht mit 5 V zu tun haben. Glücklicherweise ist der Chip 5-V-tolerant, so dass Sie die Schaltung auch in 5-V-Systemen verwenden können. Mit JP2 können Sie die Spannung am

Was ist überhaupt eine serielle Schnittstelle?

Früher, als Computer noch teuer und groß waren, erschien es sinnvoll und wirtschaftlich, nur einen großen Computer (den Mainframe) in den Keller zu stellen und daran mehrere Terminals für die Nutzer anzuschließen. Das Terminal selbst war nichts anderes als eine Tastatur und ein Bildschirm (daher „Dumb-Terminal“). Für die Verbindung der Terminals mit dem Mainframe-Rechner über eine gewisse Distanz wurde schließlich der bekannte RS232-Standard eingeführt.



Diese Norm legt eine Schnittstelle fest, mit bis zu 25 Anschlüssen inklusive einer Ader zum Senden (TXD) und einer zum Empfang von Daten (RXD). Heute, etwa fünfzig Jahre

später, ist diese Schnittstelle so gut wie verschwunden. Alles, was davon übrig geblieben ist, ist eine 3-polige Schnittstelle mit den Signalen RXD, TXD und GND. Auch die RS232-Spannungen sind durch TTL- (+5 V) oder noch niedrigere Spannungen ersetzt. Aus diesem Grund ist einem Mikrocontroller keine direkte Verbindung zu einer echten RS232-Schnittstelle möglich.

Warum heißt es überhaupt serielle Schnittstelle? Das hat mit der Art und Weise zu tun, wie Computer mit Informationen

umgehen. In einem Computer Informationen durch Bits dargestellt. Die Informationseinheit Bit kann entweder eins oder null sein. Um komplexere Inhalte darzustellen, verwendet der Computer Sequenzen von Bits. Irgendwann war man der Meinung, dass eine parallele Gruppierung von 8 Bits in einem sogenannten Byte eine gute Idee wäre. Informationen können so acht Mal schneller transportiert werden. Dazu ist ein Port mit acht Leitungen (abgesehen von der Masseleitung) erforderlich. Ein solcher Port übermittle Bits parallel und wurde deshalb als Parallel-Port bezeichnet. Wenn Sie aber ein Byte über nur eine Leitung senden wollen, müssen Sie die Bits eines Bytes sukzessive, eins nach dem anderen, übermitteln. Es geschieht „in Serie“, so dass der Ausdruck serieller Port gebräuchlich wurde.

Seit der Geburt der ersten seriellen Schnittstelle ist viel Wasser den Rhein hinunter geflossen. Wie alles, was Ingenieuren in die Finger gerät, wurde die serielle Schnittstelle vielfach modifiziert, so dass heute viele Arten üblich sind: SPI, I²C, USB, JTAG, I²S, Ethernet, SATI, SWD und unzählige mehr, alles sind serielle Anschlüsse, die die Informationen Bit für Bit transportieren.



VCC-Anschluss des „FTDI-Kabels“ wählen. Bei einem echten FTDI-Kabel sind dies immer 5 V, aber in 3,3-V-Systemen ist es meist praktischer (und sicherer), wenn die Spannung an diesem Pin auf 3,3 V beschränkt ist. Spannungsregler IC3 stellt eine Spannung von 3,3 V an K3 zur Verfügung, damit die interne 3,3-V-Quelle im FT232H nicht unnötig belastet wird. Drei LEDs auf der Platine zeigen an, was auf dem Board gerade so passiert. Die LED D3 leuchtet, wenn das Board mit Strom versorgt wird. Wenn die Schaltung als „normaler“ USB-zu-Seriell-Konverter verwendet wird, leuchten die LEDs D1 und D2 dann, wenn etwas auf der RX- oder TX-Leitung geschieht. In anderen Modi haben sie keine vordefinierte Funktion. Um die volle Leistung des FT232H abrufen zu können, ist ein externes EEPROM notwendig. Im EEPROM werden Konfigurationsdaten wie die USB-VID des Herstellers (Vendor ID) und die PID (Product ID) gespeichert, die Art der Stromversorgung (bus-powered oder mit eigener Stromversorgung) und die Funktion der ACBUS-Pins. Andere Parameter steuern den Hardware-Interface-Modus. Zögern Sie nicht, das Datenblatt für alle möglichen Optionen zu konsultieren. Erwähnt werden muss hier der USB-Suspend-Modus, ein Low-Power-Zustand, bei dem der Chip keinen Strom aus dem USB-Anschluss bezieht. R5 spielt hierbei eine wichtige Rolle. Wenn der Modus freigegeben ist, bewirkt ein logisches Low an ACBUS7 den Eintritt in den USB-Suspend-Modus. Die Standard-Einstellung im EEPROM für ACBUS7 ist: Eingang mit Pull-Down-Widerstand. Ein nicht angeschlossener ACBUS7 versetzt den Chip in den Schlaf. R5 verbindet ACBUS7 aber mit +5 V, um zu verhindern, dass dies automatisch geschieht. Der Wert von R5 wird im Datenblatt vorgeschlagen. Obwohl der FT232H auch sehr gut ohne EEPROM arbeitet, wäre er in seinen Fähigkeiten stark begrenzt. Wenn man ein EEPROM am Chip anschließt, wird es mit Standardwerten geladen. Um diese Werte zu ändern, können Sie das FTDI-Hilfsprogramm FT Prog verwenden. Dazu gleich mehr. Widerstand R2 (1%) und Quarz X1 (± 30 ppm) sind als Präzisions-Bauteile obligatorisch für eine ordnungsgemäße Funktion von IC1. Die Lastkondensatoren für den Quarz, C1 und C2, stellen sicher, dass der On-Chip-Oszillator richtig funktioniert. Der Wert dieser zwei Kondensatoren ist

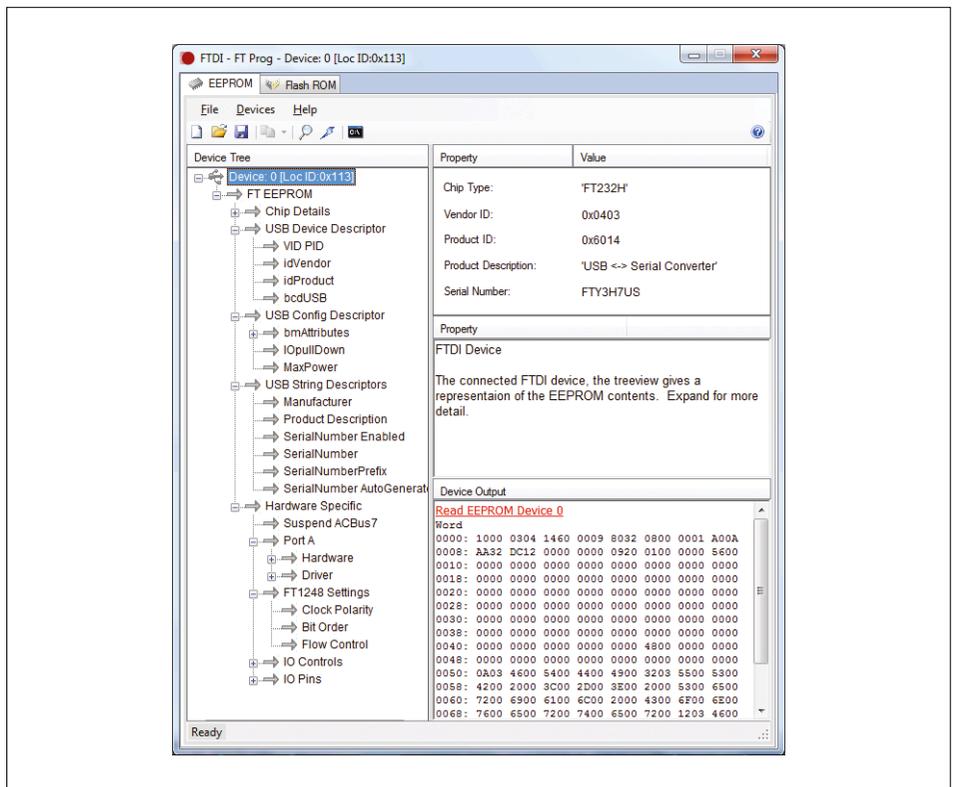


Bild 2. Glückwunsch! FT Prog hat den FT232H gefunden.

eigentlich abhängig vom Quarz, aber mit den angegebenen Werten klappt es in den meisten Fällen.

K2 ist der USB-Anschluss. Es ist ein Micro-USB, so dass Sie eventuell Ihr Handy-Ladekabel verwenden können. Diode D4 schützt die USB-Dateneingänge des Chips gegen potenziell gefährliche Überspannungen und Rauschen. L3, L4 und C18

sorgen für einen gewissen Schutz gegen Hochfrequenzrauschen auf der Versorgungsleitung und der Abschirmung.

FT PROG

Wie bereits erwähnt, steht ein spezielles Tool namens FT Prog zur Verfügung, um den FT232H (IC1) zu konfigurieren und das EEPROM (IC2) zu programmieren

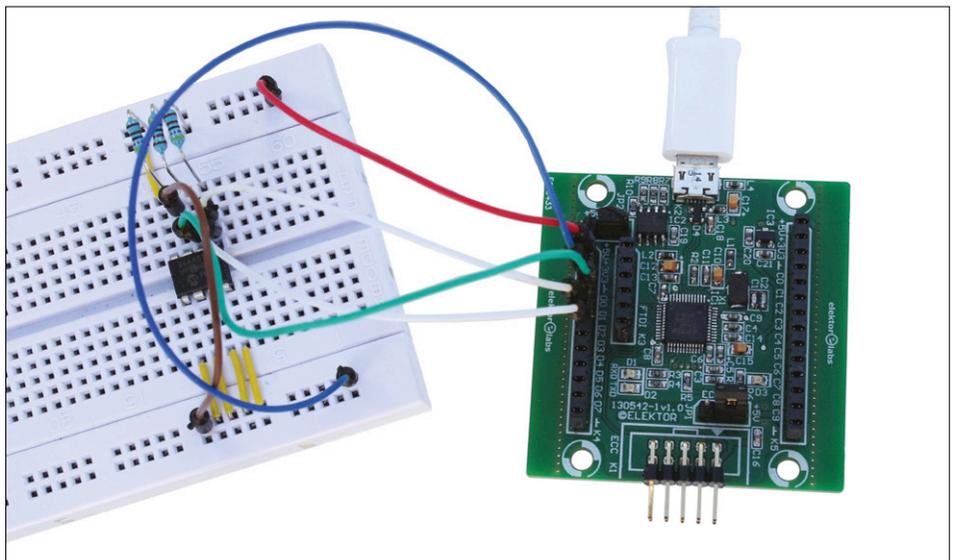


Bild 3. Ein I²C-EEPROM auf dem Steckbrett ist an das FT232H-Board angeschlossen. Beachten Sie die beiden verbundenen Leitungen von AD1 und AD2 zum SDA-Pin.

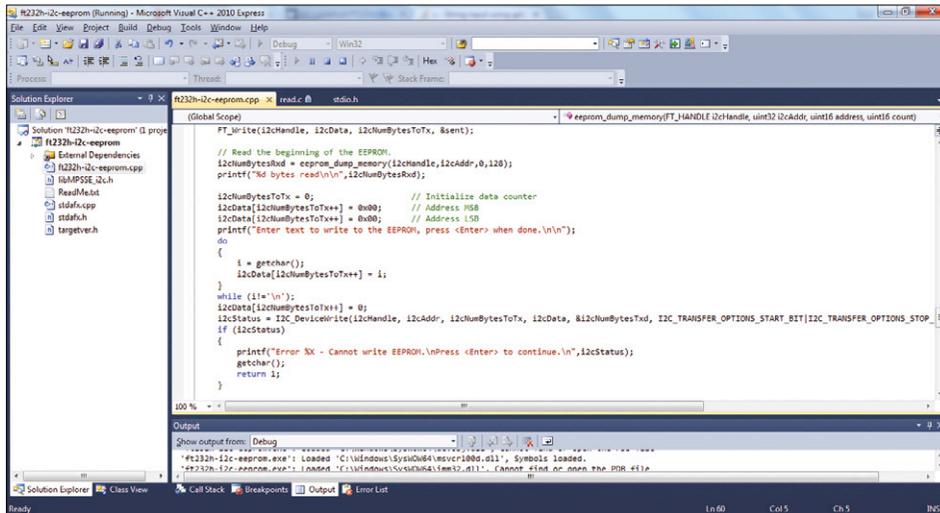


Bild 4. Der Codeschnipsel zeigt, wie Daten zum I²C-EEPROM auf dem Steckbrett geschrieben werden.

Tabelle 1. Jumper JP1

Verbindung	Funktion
1-2	Stromversorgung von K1 (ECC)
2-3	Stromversorgung vom USB

Tabelle 2. Jumper JP2

Verbindung	Funktion
1-2	K3 Pin 3 ist 5 V
2-3	K3 Pin 3 ist 3.3 V

Stückliste

Widerstände:

Alle SMD 0603, 0,1W
 R1,R7,R8,R9 = 10 k
 R2 = 12 k, 1%
 R3, R4 = 220 Ω
 R5 = 39 k
 R6 = 1k5
 R10 = 2k2

Kondensatoren:

Alle SMD 0603
 C1,C2 = 27 p, 50 V, COG/NPO
 C3,C18 = 10 n, 50 V, X7R
 C4-C9,C11,C13,C14,C16,C19 = 100 n, 50 V,
 X7R
 C10,C12,C15,C17 = 4μ7, 6V3V, Tantal,
 SMD-Gehäuse R (0805)
 C20,C21 = 1 μ, 6V3V, X5R

Spulen

L1...L4 = Ferritperle, 600 Ω bei 100 MHz,
 25%, 0Ω15 bei DC, 1A3 (SMD 0603)

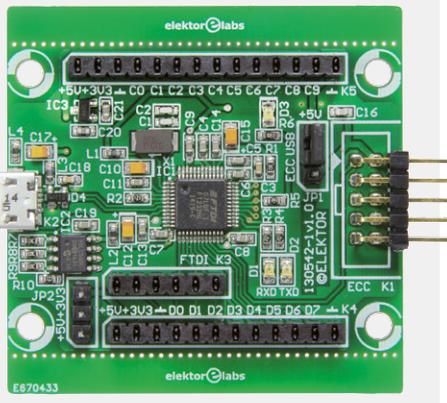
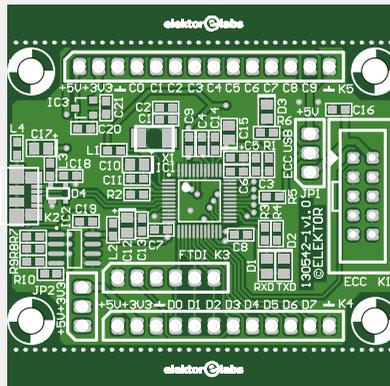
Halbleiter

D1,D2,D3 = LED, grün (SMD 0805)
 D4 = PRTR5V0U2X (SOT-143B)
 IC1 = FT232HL (LQFP-48)
 IC2 = 93LC56B-I/SN (SOIC-8)
 IC3 = XC6206P332MR (SOT-23)

Außerdem

K1 = 2x5-polige Stiftleiste, gewinkelt
 K2 = Micro-USB-Buchse Typ B, gerade, SMD
 K3 = 6-polige Buchsenleiste, 0,1"
 K4 = 12-polige Buchsenleiste, 0,1"
 K5 = 14-polige Buchsenleiste, 0,1"
 JP1, JP2 = 3-polige Stiftleiste, 0,1"
 JP1, JP2 = Jumper, 0,1"
 X1 = 12 MHz Quarz, 18 pF C_L,
 SMD 5 x 3,2 mm

Leerplatine 130542-1 v1.0
 Gehäuse Hammond 1455C802
 Platine teilbestückt 130542-91



und/oder zu lesen. Sie können FT Prog von der FTDI-Website herunterladen. Die Benutzung ist einfach: Sie schließen das Board am PC an, starten FT Prog und klicken auf den „Scan and Parse“-Knopf (das Vergrößerungsglas). Wenn alles gutgeht, sollten Sie so etwas wie in **Bild 2** sehen. Sie können die verschiedenen Optionen expandieren, bestimmte Funktionen aktivieren oder deaktivieren und Pins und Interfaces des asynchronen Teils des Chips und den USB-Treiber konfigurieren. Der MPSSE-Teil, etwa die Aktivierung und Nutzung des I²C- oder SPI-Ports, wird durch die Software gesteuert und muss nicht in FT Prog eingestellt werden.

Die interessanten Dinge, die den UART betreffen, finden sich unter „Hardware Specific“. Port A bezieht sich auf ADBUSx und ADx im Schaltbild. Sie können die Schnittstelle für diesen Port wählen (belassen Sie es am besten beim Standard-UART) und den Treiber, der durch das Betriebssystem geladen wird. Der VCP-Treiber ist voreingestellt, Sie benötigen ihn für traditionelle serielle Schnittstellenanwendungen.

Unter „IO Controls“ finden Sie den Port, der im Schaltbild mit ACBUSx und ACx bezeichnet ist (Port C). Dieser Port wird für Steuersignale verwendet, wie „TX enable“ (für RS485 und RS422) und für LED-Anzeigen. Sie müssen hier ein wenig vorsichtig sein, denn die zehn Pins sind nicht völlig identisch, sondern weisen unterschiedliche Möglichkeiten auf.

Dann etwas Verwirrendes unter „IO Pins“: Sie finden hier die beiden Ports A und C, aber jetzt als Gruppen AD und AC. Hier können Sie die Slew Rate und den Eingangstyp auswählen und die Stromabgabe für diese Anschlüsse einstellen. Wenn Sie mit Ihrer Konfiguration zufrieden sind, können Sie die Einstellungen im EEPROM ablegen. Dazu klicken Sie auf den Knopf „Program Devices“ (das Blitzsymbol). Für unsere Experimente mit der seriellen Schnittstelle, synchron oder asynchron, können Sie alles so lassen, wie es voreingestellt ist.

Programmbeispiele

Sie haben es gesehen, der FT232H ist ein recht kompliziertes IC mit umfangreichen Einstellmöglichkeiten. Damit der Anwender all dies auch nutzen kann, stellt FTDI mehrere Application Notes [1] mit Programmierbeispielen inklusive Quellcode zur Verfügung. Die interessantesten sind ANs 180, 188 und 190. Obwohl sich die



Bild 6. Das FT232H-Board passt exakt in das Aluminium-Gehäuse von Hammond. K1 schaut auf der anderen Seite heraus.

DI-Treiber deinstallieren. Hierzu müssen Sie die Box „Delete the driver software for this device“ anklicken. Trennen Sie dann das FT232H-Board und stöpseln es wieder ein, so installiert das Betriebssystem die FTDI-Treiber erneut.

Basierend auf der Python-GPIO-Bibliothek haben wir ein Beispiel für GPIO-Programmierung erstellt, ein Lauflicht mit 16 LEDs von AD0 bis AC7. An dieser Stelle noch eine Bemerkung: AC8 und AC9 sind nicht zugänglich. Wenn man das Datenblatt unter die Lupe nimmt, so stellt man fest,

dass diesen Pins keine Funktion zugewiesen ist. Die einzige Möglichkeit, sie zu nutzen, ist es, ihnen mit FT Prog eine Funktion zu geben.

Eine letzte Sache zu Python: Um Frustration zu vermeiden, verwenden Sie besser Python 2.7.x (anstelle von 3.4.x).

Schlusswort

Da wir das FT232H-Board für ein sehr nützliches Werkzeug halten, haben wir beschlossen, die Platine etwas größer zu machen, damit sie gut in ein Standard-

gehäuse passt (**Bild 6**). Das Aluminiumgehäuse von Hammond ist allerdings zu lang, es kann aber leicht auf die richtige Länge gestutzt werden. Sie müssen dann nur noch die ABS-Deckel mit den entsprechenden Ausbrüchen versehen. Die FT232H-Platine gleitet einfach in dieses Gehäuse und stellt so einen stabilen USB-zu-Seriell-Konverter dar. Sollte Ihnen die Platine als zu groß erscheinen, können Sie sie auch an der gepunkteten Linie abschneiden. ◀

(130542)

Weblinks und Literatur

- [1] FT232H: www.ftdichip.com/Products/ICs/FT232H.htm
- [2] Universelles Audio-DSP-Board, Elektor Januar/Februar 2014, www.elektor-magazine.de/130232
- [3] Adafruit FT232H Python GPIO library: <https://learn.adafruit.com/adafruit-ft232h-breakout/>
- [4] Diese Kabel gibt es immer noch: 080213-71 (5 V-Version) und 080213-72 (3,3 V-Version) bei www.elektor.de
- [5] USB/RS232-Adapter für USB/TTL-Schnittstellenkabel 080470-1 auf www.elektor.de
- [6] USB-FT232R Breakout-Board, Elektor September 2011, www.elektor-magazine.de/110553. Aufgebautes Modul 110553-91 auf www.elektor.de
- [7] USB-Hub/UART-Konverter - mit RS232 und RS422/RS485, Elektor November 2014, www.elektor-magazine.de/140033. Aufgebaute Schaltung 140033-91 auf www.elektor.com

Willkommen bei **SHARE**



Von **Jaime González-Arintero**

jaime.glez.arintero@eimworld.com

A.C.A.B. (All Clocks Are Boring)

Ich würde ja gerne glauben, dass Sie eifrig durch diese Ausgabe blättern, um die beste aller Rubriken zu finden: SHARE. Okay, das ist wahrscheinlich nicht der Fall. Es gehörte sich so, dass wir Ihnen in der letzten Ausgabe unsere neuen Rubriken erst

einmal bekannt gemacht haben. Jetzt aber *können* wir endlich „nerdy“ miteinander diskutieren.

Der Einfachheit halber verlassen sich die meisten ICs - und überhaupt jede Schaltung ab einer gewissen Komplexität - darauf, dass eine Uhr tickt: Ein CLK-Signal regelt die Sache schon. Alle Ereignisse geschehen nur, wenn eine Uhr sie diktiert. Dieses ungeschriebene Gesetz klingt *logisch*, für die meisten von uns Elektronikern ist es nahezu ein Tabu, taktlos auf einen Takt zu verzichten. Sind Taktgeber heutzutage etwa ein Hindernis für den Fortschritt?

Lassen Sie uns das Konzept veranschaulichen. Stellen Sie sich eine große Firma vor; mit perfekt aufeinander abgestimmten Arbeitern in miteinander verzahnten Prozessen. Einige Mitarbeiter mögen schneller (und intelligenter) sein als andere, aber sie sind gezwungen, ihre Arbeit zu verlangsamen und/oder zu warten, bis der Rest des Teams seine Aufgabe beendet hat. Genau so müssen einige Teams auf andere warten, bis alles ordnungsgemäß übergeben wurde. Stellen Sie sich jetzt ein Start-up-Unternehmen vor, bestehend aus einem kleinen Team, das auf der

Arbeitsgeschwindigkeit einer Arbeitskraft oder eines Teams beruht, ganz ohne strenge Synchronisation. Am Ende wird der Gewinn des Riesenunternehmens nicht unbedingt die Gewinne des Startups übersteigen.

Asynchrone Mikrocontroller bieten einige Vorteile gegenüber ihren „Slave-to-the-clock“-Genossen: extrem geringer Stromverbrauch, nahezu keine elektromagnetischen Störungen und ein schneller und effizienterer Betrieb. Außerdem benötigt in asynchroner Betriebsart die Peripherie buchstäblich null Leistung, wenn sie nicht verwendet wird. Das gleiche gilt für die CPU im Ruhezustand. Ein taktloser Chip ist auch fähig, den Betrieb zu beschleunigen oder zu verlangsamen, wenn die äußeren Bedingungen dies erfordern. Diese Vorteile sind nicht zu verachten, wenn man bedenkt, dass einige der innovativsten Projekte auf einem Clockless-Ansatz beruhen. Zum Beispiel arbeiten die neuromorphen SyNAPSE-Chips des amerikanischen Rüstungsproduzenten DARPA, von denen einige die höchste jemals erreichte Transistorzahl aufweisen, im asynchronen Modus.

Einige Entwickler glauben, dass die Zukunft den Verzicht auf eine Taktung mit sich bringt. Allerdings haben Neinsager auch gute Gründe. Die meisten Entwicklungsumgebungen sind nicht für asynchrone Logik bestimmt, zudem sind Ingenieure in der Regel nicht darin geschult, die Dinge zu verlangsamen, so dass der Entwicklungsprozess erst einmal sehr zeitaufwändig sein dürfte. Auch innerhalb der akademischen Welt wird, abgesehen von einigen Arbeitsgruppen, das Thema als fast esoterisch angesehen. Und bei einer Schnittstelle zwischen synchronen und asynchronen Chips ist es wie in der Politik: Es ist besser, nicht darüber zu sprechen!

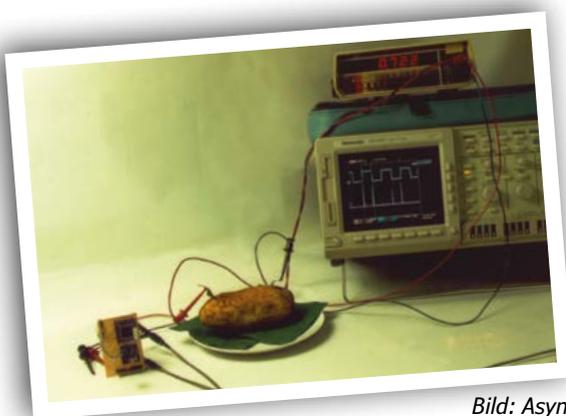


Bild: Asynchronous VLSI Group, Caltech

Echte **Kartoffel**chips

Es ist bekannt, dass Clockless-Chips schneller laufen und viel effizienter arbeiten als ihre synchronen Kollegen. Bereits 1989 zeigte der Ingenieur Mika Nyström aus der Asynchronous VLSI Group am California Institute of Technology, wie ihr CAM (Caltech Asynchronous Mikroprozessor, die weltweit erste asynchrone MCU) mit rund 300 kHz lief und dabei eine ganz eigentümliche Stromversorgung nutzte: zwei in Reihe geschaltete Kartoffeln, die etwa 0,9 V produzierten.

Restaurierung des Röhrenvoltmeters HP 400H (2)

Von Chuck Hansen (USA)

Dies ist die Fortsetzung von Teil 1, der in Elektor 3/2015 veröffentlicht wurde. Der nachfolgende Text bezieht sich auf die hier nochmals abgebildete Schaltung.

Erste Tests

V9 ersetzte ich mit einer Regler-Röhre des Typs NOS 5651A und schaltete dann mein HP 400H ein.

Dabei erhöhte ich mit meinem einstellbaren Wechselspannungsnetzgerät B-K 1655 langsam über 15 Minuten die Versorgungsspannung. Die geregelte Schiene

B+ erreichte dann die vorgesehenen +245 V, aber die Spannungen an den Anoden und diversen Gittern der vier Ver-

stärkeröhren V2...V5 waren immer noch zwischen 15 und 37 % zu hoch.

Dann fiel auf, dass die Heizung für V1...V4 mit nur 10,5 V zu niedrig eingestellt war. Mit R66 versuchte ich diese Spannung auf normgerechte 12,6 V (bei einer Netzspannung von 115 V) einzustellen, doch selbst bei Rechtsanschlag waren nur maximal 11,5 V möglich. Die beiden in Serie geschalteten Sekundärwicklungen des Trafos aber lieferten korrekte 13,6 V. Bevor sich nun CR3 stinkend (und giftig) verabschieden konnte, wollte ich das Bauteil durch einen Silizium-Brückengleichrichter ersetzen. Ich war mir sicher, dass dann ein zusätzlicher Serienwiderstand erforderlich war, da der Spannungsabfall bei Siliziumgleichrichtern üblicherweise kleiner als der von gealterten Selenexemplaren ausfällt. Erst dann konnte man den Arbeitspunkt bzw. die korrekten Spannungen an den Röhren mit R119 einstellen. Als nächsten Schritt ersetzte ich auf Verdacht den Vierfach-Elko C30 (4 x 20 µF / 450 V) durch ein Exemplar mit einer

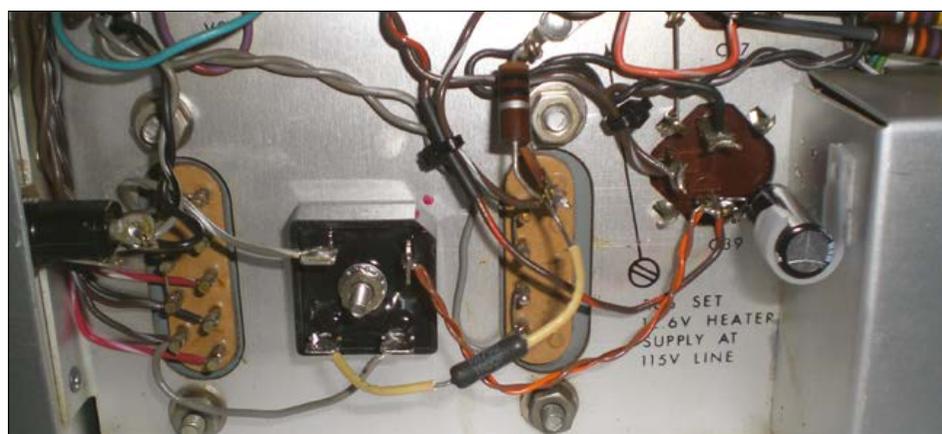
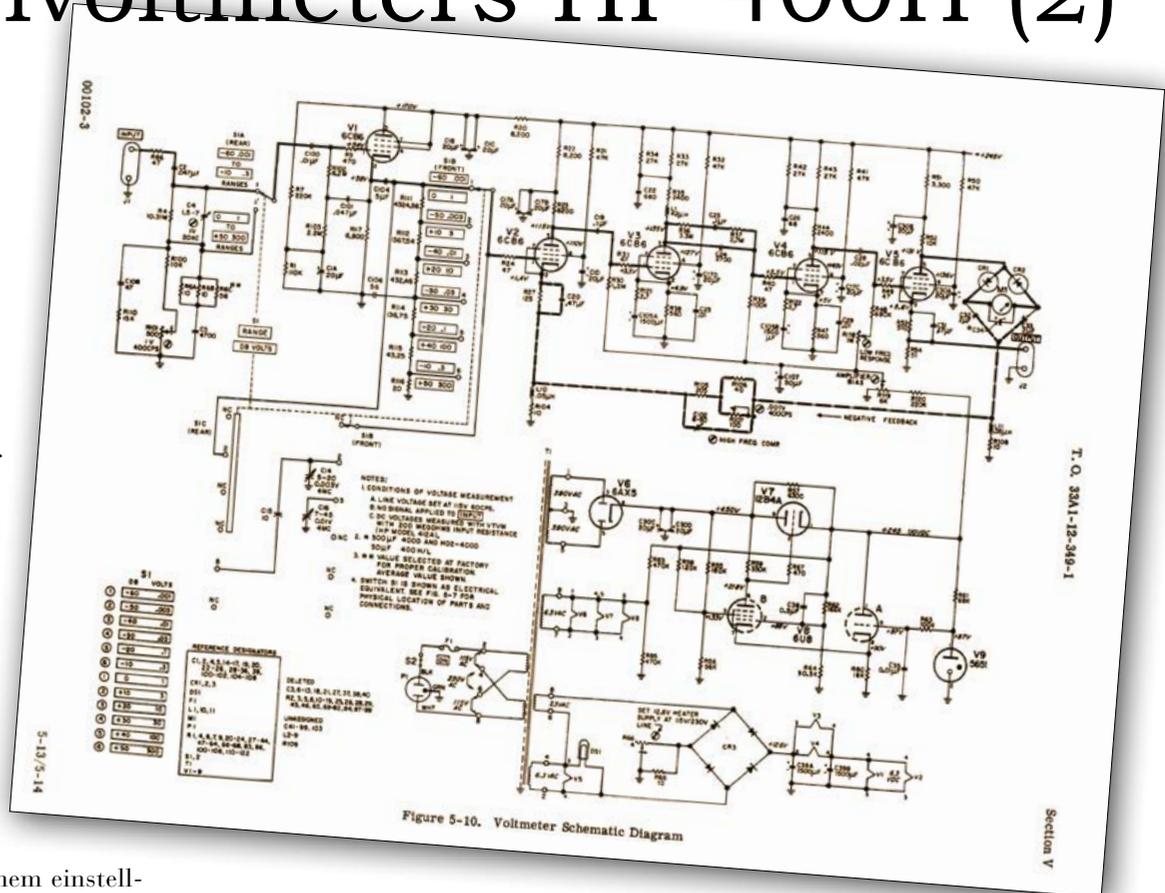


Bild 7. CR3 ersetzt durch einen Silizium-Gleichrichter. Hinzu kamen ein 1-Ω-Widerstand und parallel zu C30C ein radialer Elko mit 1.500 µF (befestigt an der Abschirmung von V1 und V2).

EST[®] 2004

www.elektor.tv



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).

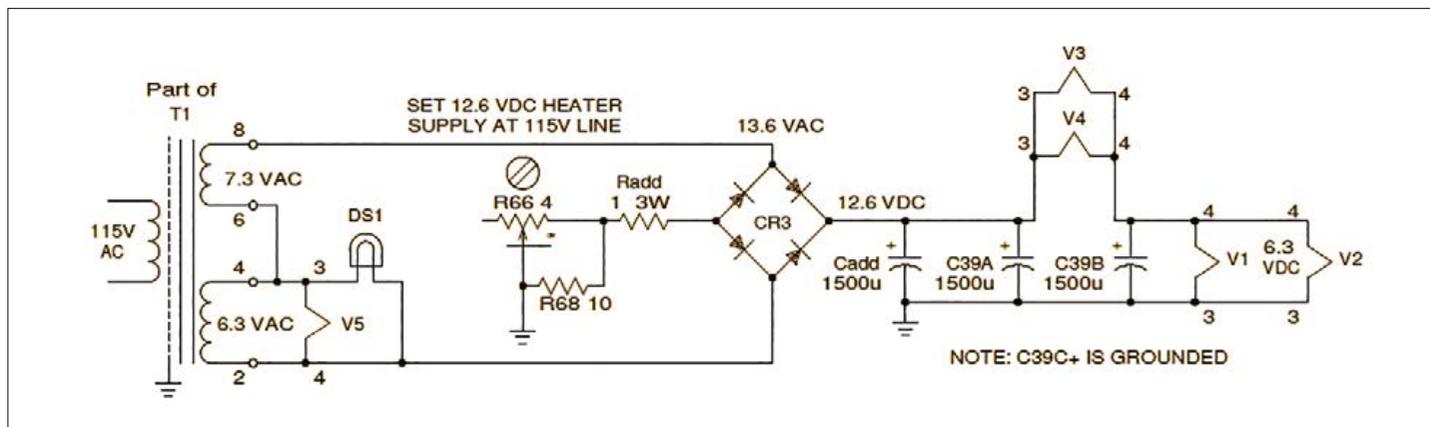


Bild 8. Schaltung der modifizierten Versorgung der Heizung mit Gleichspannung.

Spannungsfestigkeit von 475 V. Hier liegt nämlich die hohe, noch unstabilierte Anodenspannung an, und dieser Elko hatte beim Versagen der Regelung sicher am meisten gelitten.

Weitere Tests

Jetzt wurde CR3 durch eine Silizium-Brücke mit 50 V / 15 A ersetzt. Der Spannungsabfall an der Selen-Brücke lag zuvor fast bei 2,5 V. Also schleifte ich eine Leistungs-Widerstandsdekade in den negativen Anschluss der Brücke ein. Beim Einschalten zeigte sich auf dem Oszilloskop ein relativ hoher Brummanteil der Heizgleichspannung. C39 hatte wohl Kapazität verloren. Unglücklicherweise hatte der Lieferant für antike Elektronik AES (tubesandmore.com) keinen Ersatztyp im Online-Katalog.

Also schaltete ich einen diskreten 1.500- μ F-Kondensator parallel zu C39A. Der Zusatz-Elko wurde mit einem modifizierten Kunststoff-Clip von Burndy links an der Abschirmung von V1 und

V2 befestigt (Bild 7). Die Widerstandsdekade ergab einen Wert von 1 Ω für die korrekte Heizung mit 12,6 V bei Mittelstellung von R66. Folglich kam anstelle der Dekade der Drahtwiderstand RS2B von Dale mit 1 Ω / 3 W zum Einsatz.

Um die Wechselspannungsanschlüsse des neuen Brückengleichrichters zu erreichen, musste ich die graue und die grau/weiße Leitung vom Trafo verlängern. Als nächstes musste ich noch die braun-orange Leitung vom Plusanschluss des Gleichrichters zur Heizung der Röhren aufsplitten. Da ich keine braun-orange Litze zum Anschluss von C39A an den Gleichrichter hatte, verdrillte ich zur Beibehaltung des Farb-Codes eine braune AWG-22-Tel fonlitze mit einer orangen in AWG-24.

Bild 8 zeigt die Schaltung der geänderten Versorgung der Heizung mit Gleichspannung.

Bei kurzgeschlossenem Eingang bleibt die Anzeige des HP 400H in den beiden niedrigsten Bereichen bei 0,001 V und 0,003 V. Es sollte eine geringere Nullab-

weichung in diesem Bereich möglich sein. Es gibt zwei Elkos mit 50 μ F und 5,6 V von Sprague: C34 über dem Drehspulinstrument und C107 bei der Arbeitspunkteinstellung des Verstärkers. Hierfür bestellte ich Ersatzelkos mit 47 μ F / 16 V von Vishay bei Mouser, die für 105 $^{\circ}$ C ausgelegt sind.

Nun ging es dem zu hohen Nullwert an den Kragen. Nach dem Ersatz von C34 (Bild 9) ergab sich im 1 mV-Bereich eine Nullanzeige von 0,18 mV (offenes Gehäuse und kurzgeschlossener Eingang). Ich überprüfte den alten Elko mit meinem LC-Meter Sencore LC102: Es zeigte einen Wert von nur 25 μ F bei 5,4 V an. Außerdem war der ESR mit 15,7 Ω sehr hoch. Die dielektrische Absorption lag bei 31 %, und mit einem Leckstrom von über 700 μ A markierte das LC-Meter diesen Elko als defekt. Das Ding war kein richtiger Kondensator mehr.

Die Gitterspannungen waren an allen 6CB6-Röhren zu hoch. Also wurde auch C107 ersetzt. Leider war C107 unter vie-

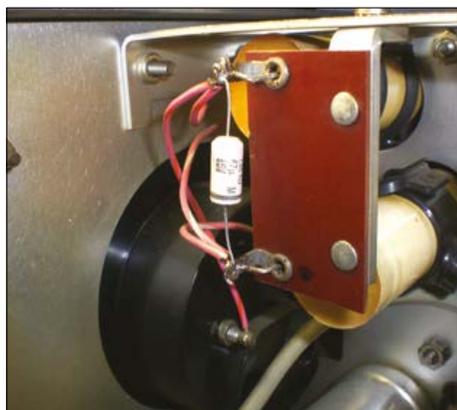


Bild 9. Ersatz für C34, parallel zu den Klemmen des Instruments.



Bild 10. C107 (silbernes Gehäuse) unter anderen Bauteilen und einer schwarzen Drahtbrücke.



Bild 11. Der Ersatz für C107 wird über C23 eingebaut.

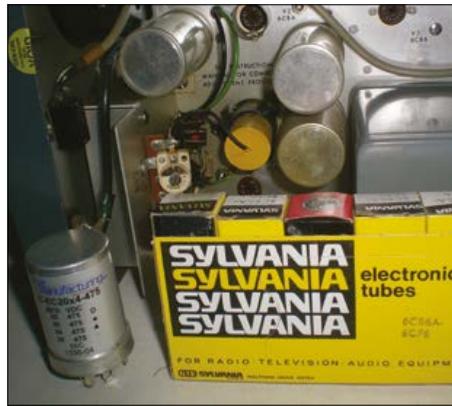


Tabelle 1.
Spannungsfehler beim HP 400H

Bereich	f	V _{test}	Fehler
300 V	60 Hz	300 V	0%
		270 V	0%
		30 V	-3.3%
100 V	60 Hz	100 V	0%
		90 V	0%
		10 V	5.0%
30 V	60 Hz	30 V	0%
		27 V	0%
		3 V	0%
10 V	60 Hz	10 V	0.8%
		9 V	0.22%
		1 V	4.0%
3 V	1 kHz	3.0 V	0%
		2.7 V	0%
		0.3 V	1.3%
1 V	1 kHz	1.0 V	0.5%
		0.9 V	0.6%
		0.1 V	1.6%
0.3 V	1 kHz	0.30 V	2.3%
		0.27 V	2.5%
		0.03 V	4.3%
0.1 V	1 kHz	0.10 V	2.0%
		0.09 V	2.2%
		0.01 V	8.0%
0.03 V	1 kHz	0.030 V	1.7%
		0.027 V	2.2%
		0.003 V	23%
0.01 V	1 kHz	0.010 V	3.0%
		0.009 V	5.9%
		0.001 V	28%
0.003 V	1 kHz	0.0030 V	2.3%
		0.0027 V	4.1%
		(siehe Text)	
0.001 V	1 kHz	(siehe Text)	

Bild 12. Der neue C30 samt 6CB6A-Röhren in Schachteln.

Bild 13. Originaler C30 von unten mit benachbarten Bauteilen.

len anderen diskreten Bauteilen und einem schwarzen Jumper begraben (Bild 10). Ich musste also einen Weg finden, möglichst wenige Bauteile zu entfernen. Am einfachsten war, den alten C107 abzuzwickeln und den Ersatz über dem schwarzen Filmkondensator C23 von Sprague einzulöten (Bild 11).

Die Überprüfung des alten Elkos ergab 51,6 μF bei einem ESR von 3,9 Ω . Mit dem Leckstrom von 17 μA und einer Absorption von 12 % wäre er auch nach Meinung des LC-Meters noch akzeptabel gewesen. Der neue Vishay-Kondensator war mit 45,1 μF und 3,7 Ω (unter der Grenze von 5,4 Ω) ja nicht wirklich besser. Mit R119 war es dann möglich, die richtigen Gitterspannungen einzustellen. Doch die Anode von V3 hatte eine Spannung von 154 V statt 121 V. Nach dem Tausch von V3 und V4 hatte die neue V4 die erhöhte Spannung (152 V statt 119 V). Da mindestens eine Röhre Probleme machte, bestellte ich neue 6CB6-Pentoden für V1...V5.

Bei der weiteren Überprüfung der Elektronik fiel mir auf, dass sich zuvor schon jemand an diesem Gerät vergriffen hat. Es gab einige kalte Lötstellen – ein klarer

Hinweis darauf, dass der Lötter über nicht allzu viel technische Erfahrung verfügt hatte. Weiter fiel auf:

1. Bei C24 war ein Glimmer-Kondensator mit 1800 μF / 500 V von Sangamo vorgesehen (meine spätere Schaltung meinte: 2700 μF). Er wurde durch ein oranges Folien-C von Sprague mit 2,5 nF / 600 V ersetzt.
2. R27 sollte ein Drahtwiderstand mit 125 Ω und $\frac{1}{2}$ W von Dale sein. Er wurde durch das 3-W-Exemplar CS-2 mit 125 Ω von Dale ersetzt. In meinem Dale-Katalog von 1991 gab es leider keine Infos zu dieser CS-Serie. Er gleicht aber völlig der mir bekannten RS-Serie.
3. C105 war ausgelaufen, sichtbar am Chassis. Er wurde durch ein identisches Exemplar mit neuem Datum von 1957 ersetzt. Die anderen Cs sind von 1952. Beim Ersatz von C105 wurden die Drähte von R122 zu stark gekürzt. Sie wurden mit kleinen verzinneten Drahtstückchen verlängert.
4. Die Kunststoff-Hülle der Abschirmung von C30 ist leicht braun. Dies ist ein Indiz für erhöhte Temperatur nach Ausfall von V9. Zunächst wechselte ich die fünf 6CB5A-Röhren aus, und dann

Verwendete Ersatzteile

Von AES:

- C30 = 4 x 20 μF / 475 V, Elko im Alubecher, C-EC20X4-475
- V1..V5 = Pentode 6CB6-A/6CF6
- V9 = Regler-Röhre 5651

Von Mouser:

- CR3 = Silizium-Brückengleichrichter, z.B. 12 A / 100 V, 512-GBPC1201
- C34, C107 = 47 μF / 16 V, 105 °C, axial, 594-2222-138-25479
- Cadd = 1.500 μF / 16 V, 105 °C, radial, 667-EEU-FR1C152LB
- Radd = 1 Ω , 3 W, 1 %, Drahtwiderstand, 71-RS2B-1.0



Bild 14. C30 ausgebaut.



Bild 15. Neu eingebauter C30 mit benachbarten Bauteilen.



Bild 16. Ersetzter C30 von oben.

wurde für C30 das neuere Exemplar mit 475 V bestückt. Das Vorgehen ist in den **Bildern 12 bis 16** veranschaulicht.

Die Anodenspannungen stimmten nun, doch die von V4 und V5 waren immer noch etwas hoch. Gegenseitiger Röhrentausch änderte daran nichts. Der Serienwiderstand für das Schirmgitter von V5 betrug statt 47 k Ω tatsächlich 52 k Ω . Bei geschlossenem Gehäuse und kurzgeschlossenen Eingängen wurden im 1-mV-Bereich nun 0,017 mV angezeigt.

Genauigkeit

Nach den Reparaturen wurde mein HP 400 H anhand des HP-Manuals mit meinem parallel geschalteten Fluke DMM 8050A kalibriert, das selbst wiederum im Juni 2014 mit dem „AC voltage calibrator 5200“ von Fluke kalibriert worden war. Als Signalquelle kam mein HP-339A „distortion test set oscillator“ zum Einsatz. Zuerst wurde der Frequenzgang mit einem Sinus-Signal von 1,0 V_{eff} zwischen 10 Hz und 50 kHz überprüft. Hierzu wurde das HP 400H in den 1-V-Bereich gesetzt. Die Anzeige betrug 0,95 V bei 10 Hz und 0,96 V bei 20 Hz. Richtig flach war es zwischen 50 Hz und 20 kHz. Bei 50 kHz ergab sich ein Wert von 1,005 V. Letzteres war auch die Frequenzgrenze für mein DMM des Typs 8050A. Für den Test der oberen Frequenzgrenze des HP 400H von 4 MHz fehlten mir die Messgeräte.

Als nächster Schritt wurde die Genauigkeit der Spannungsmessung überprüft, und zwar bei 10, 90 und 100 % des Vollausschlags der zwölf Messbereiche. Hierzu wurde der Signalgenerator HP-339A auf 1 kHz eingestellt und seine Ausgangsspannung von 2,7 mV_{eff} bis auf 3 V_{eff} erhöht. Da der kleinstmögliche Pegel des HP-339A 2,5 mV beträgt, konnte der

1-mV-Bereich des HP 400H nicht überprüft werden. Insgesamt war zu sehen, dass sich der Fehler bei 10 % mit sinkendem Pegel erhöhte. Dennoch blieb das Messgerät bei 90 und 100 % der Skala im 1-V-Bereich immer noch innerhalb von ± 1 %. Unterhalb dieser Pegel wird das Messergebnis wohl durch Rauschen beeinträchtigt. Die Testresultate sind in **Tabelle 1** zusammengefasst.

Im Manual zum HP 400H wird das Einschleifen eines „cheater plug“ in die Netzleitung vorgeschlagen, der zur Erhöhung der Genauigkeit die Erdung unterbricht. Solche Tricks finde ich nicht gut, da nicht bekannt ist, wie gut die Isolation des Gehäuses eines solchen 50 Jahre alten Geräts faktisch noch ist. Die maximale Ausgangsspannung des HP-339A beträgt 6,75 V_{eff}. Für die Messbereiche ab 10 V des HP 400H nutzte ich daher mein von 0 bis 325 V einstellbares 60-Hz-Netzteil. Die Fehler bei 10 % lagen unter ± 5 %. Bei 90 und 100 % war die Genauigkeit wieder viel besser (siehe Tabelle 1).

Unerledigtes

Eigentlich sollte noch der verschlissene

Ledergriff ersetzt werden. Bei AES gab es früher guten ledernen Ersatz. Mittlerweile werden leider nur noch gummierte Griffe für Gitarrenverstärker mit Logos angeboten.

Mit C1 und C17 gibt es noch zwei weitere Quad-Elkos mit 20 μ F und 450 V, die ersetzt werden sollten.

Es würde mich sehr freuen, wenn ich einen Ersatz für die beiden anderen Triple-Elkos mit 3 x 1500 μ F bei 15 V sowie für C34 und C107 finden könnte. Ein ähnliches Problem stellte sich bei der Renovierung meines alten Verstärkers vom Typ Scott 222C (*audioXpress* 5/2006 bis 9/2006). AES bietet zwar Elkos für hohe Spannungen, doch für den Quad-Elko mit 4 x 75 μ F / 75 V für die Gitter der Ausgangsröhre (C207...C210) gab es nichts. Ich musste diesen mit diskreten Elkos mit 100 μ F / 100 V und einer Temperaturfestigkeit von 105 °C ersetzen

Alles in allem war ich aber zufrieden mit der Leistung des HP 400H. Die Restaurierung war erfolgreich und auch interessant. Man konnte dabei lernen, wie man früher elektronische Probleme gelöst hat. ◀

(150034)

Der Autor



die Jazz-Gitarre. Sehr gern hackt er Gitarrenverstärker und setzt alte Audio-Messtechnik wieder in Betrieb.

Chuck Hansen ist ein Elektrotechnik-Ingenieur und hält auf seinem Fachgebiet fünf Patente. Er arbeitet als Berater in der Luftfahrtindustrie. Chuck hat bereits zwei Bücher für Audio-Fans geschrieben und über 260 (!) Zeitschriftenartikel verfasst. Geräte mit Röhrentechnik baut er seit seiner Zeit auf dem College. Zu seinen Hobbys zählt er das Segeln und

Gitarrenverstärker und setzt alte Audio-

Im Innern des Apfels

Die meisten der heutigen Apple-Produkte lassen sich nicht von jedermann leicht öffnen. Wir Elektroniker hassen solche Black Boxes, denn wir wollen schließlich wissen, wofür wir unser Geld ausgeben. Haben wir uns nicht schon als Jugendliche über Radios und TVs hergemacht, ohne uns von irgendwelchen 20 kV schrecken zu lassen? Warum sollte uns jetzt eine versiegelte Kunststoffkiste abhalten, den Dingen auf den Grund zu gehen?

Von **Thijs Beckers** (Elektor Labs)

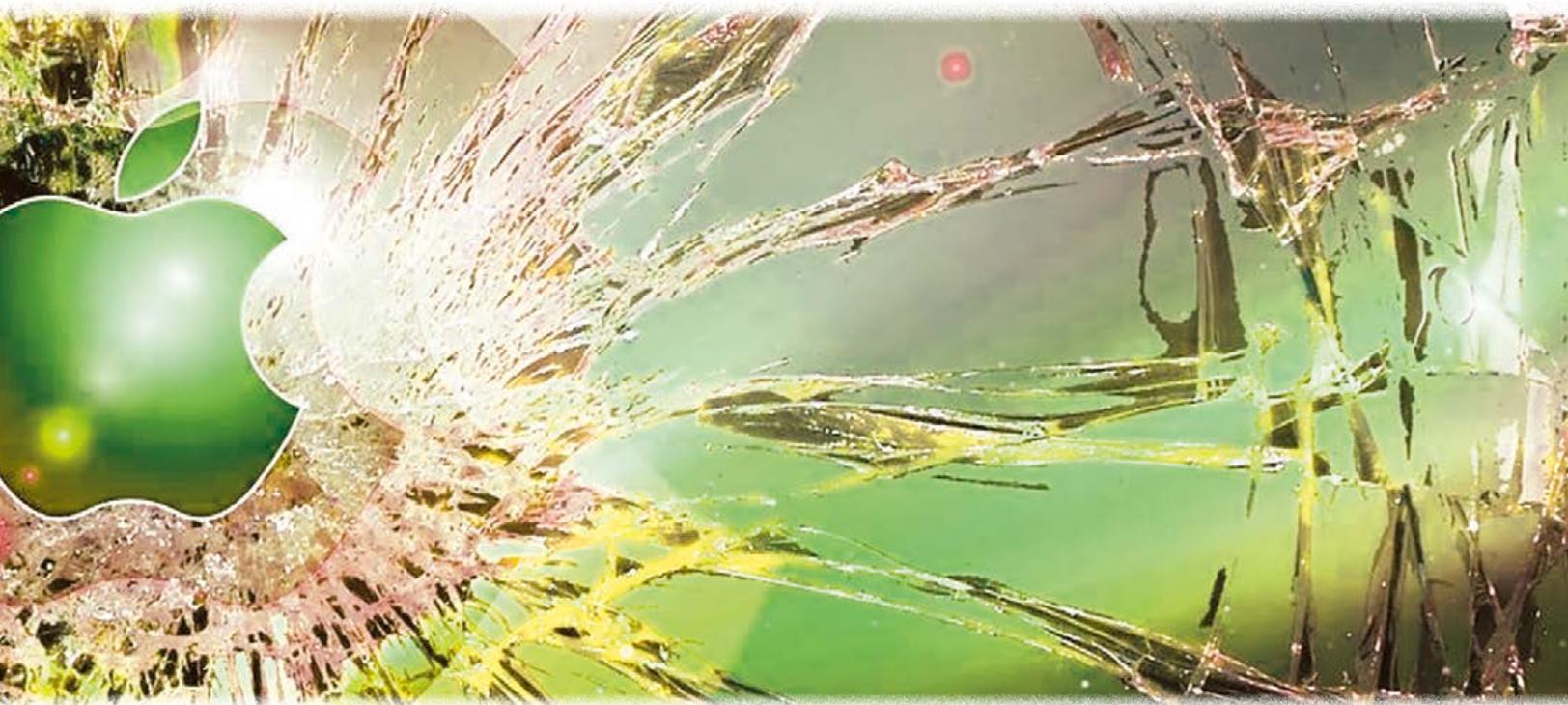
Wenn Sie ein wenig wie wir im Elektor-Labor geartet sind, dann sind Sie auch am Inneren elektr(on)ischer Geräte interessiert, vor allem, wenn diese kaputt sind und Sie den Drang Ihrer heilenden Hände verspüren. Nun, hier blicken wir in einen 90 Watt Apple Cinema HD Display Power Adapter mit einer vermuteten Funktionsstörung.

Ein erster Blick auf den Stromlieferanten zeigt eine Naht an den Seiten, an der man die obere und untere Hälfte des Kunststoff-Schalengehäuses leicht trennen könnte. Konjunktiv! Denn die beiden Schalen sind mit einem Schmelzkleber miteinander verbunden. Doch für unsere Dremel stellt das kein Problem dar! Aber Vorsicht: Einige Kom-



ponenten sind sehr nah am Gehäuse montiert und können leicht zerstört werden, wenn die Schleifscheibe ein wenig zu weit in den Kunststoff taucht.

Die Oberschale ist gelöst, 1:0 für uns! Doch bedarf es noch ein wenig „Überzeugungskraft“. Die chinesischen Hersteller wer-



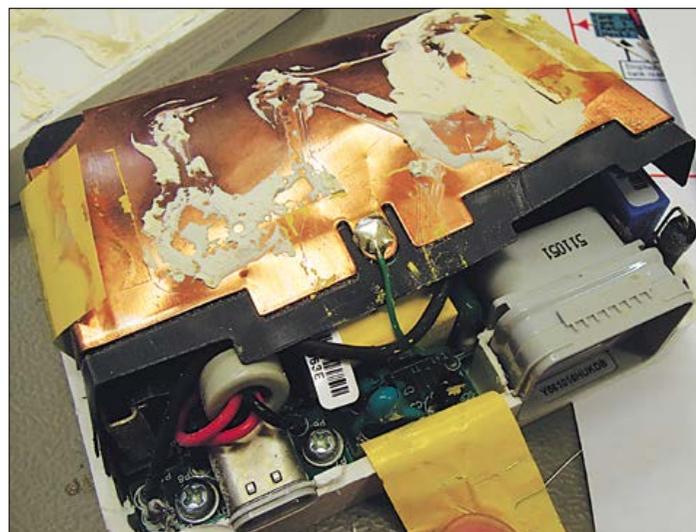
den von Apple bezahlt, um mit einer Art Kitt die Innereien der Dose unseren Blicken zu entziehen. Ein Ruck – und es steht Apple: 0, Elektor: 2.

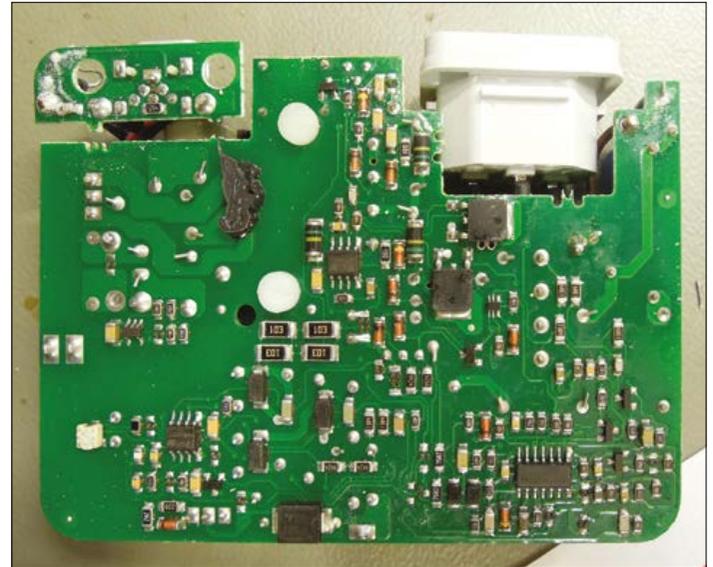
Jetzt blicken wir auf eine große Kupferfolie, die um die ganze Schaltung gefaltet ist und als Abschirmung fungiert. An die Folie ist ein Draht gelötet, der die Abschirmung mit der Platine verbindet. Dieses Kabel wird durchtrennt, die Abschirmung auseinander gefaltet und es steht 3:0, das Spiel ist gelaufen! Nun schauen wir uns erst einmal gründlich um. Wir stellen fest, dass die Platine nicht mit dem Gehäuse verschraubt ist, sondern nur „strack“ hinein passt. Wir sehen auch, dass sich der Apple-Display-Stromanschluss und der Netzanschluss jeweils auf einer kleinen separaten Platine befinden. Das ist eigentlich ziemlich clever.

Dies sind die einzigen Teile, die nicht direkt mit der Hauptplatine verlötet sind, sondern auf separaten Platinen sitzen, die auch nicht an der Hauptplatine, sondern am Gehäuse befestigt

sind. Es sind auch die einzigen Teile, die einer mechanischen Belastung ausgesetzt sein können. So lässt sich leicht vermeiden, dass unsachgemäße Bedienung (brutales Herausreißen des Kabels) die Hauptplatine beschädigen kann.

Die Schaltung selbst ist ein stinknormales Standardschaltenteil. Es gibt einige Filter-Induktivitäten und Kapazitäten am AC-Eingang, einen Brückengleichrichter und einen dicken Kondensator, um eine hohe Gleichspannung zu erzeugen, aus der der Transformator seine Kraft bezieht. Dann kommen zwei Leistungs-MOSFETs für den schaltenden Teil, gefolgt vom Transformator. Die MOSFETs werden von Hochspannungs-Halbbrückentreibern mit Oszillator L6571B von ST Microelectronics angesteuert. Auf der Sekundärseite des Trafos sind ein Gleichrichter (mit Kühlkörper) und schließlich die Ausgangsfilterkondensatoren angebracht. Die gesamte Schaltung wird von einem STMicro-IC mit nicht zu entzifferndem Aufdruck gesteuert. Und





natürlich enthält die Schaltung ordentlich passives „Hühnerfutter“ (Widerstände und Kondensatoren).

Es gibt auch eine recht umfangreiche Schaltung zur Leistungsfaktorkorrektur (PFC). Der Transformator, der nicht vollständig in das gelbe Tape eingewickelt ist, ist Teil dieser Schaltung.

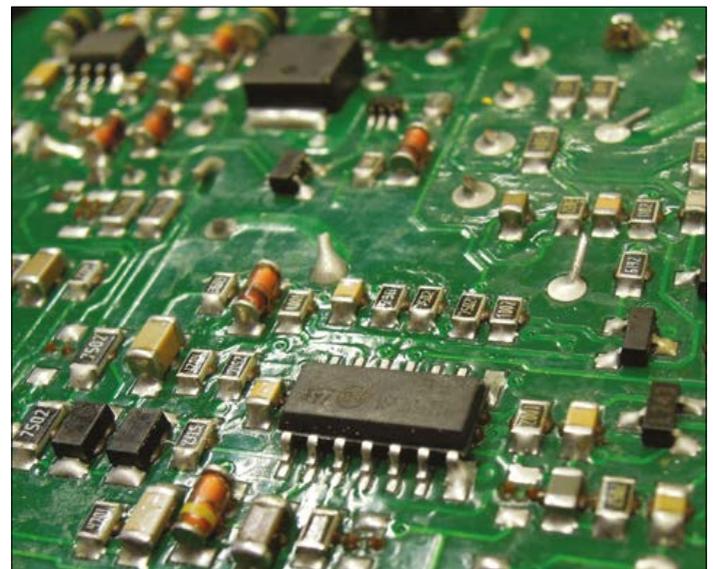
Ein Blick auf die Kühlmaßnahmen offenbart nur zwei Metallstreifen auf beiden Seiten der Platine. Ein Streifen kühlt die Gleichrichterdiode für die Ausgangsschaltung, die andere das, was wir für einen Schalt-FET der PFC-Schaltung halten, die mit einem L6561D von ST Microelectronics aufgebaut ist. Um die Typenbezeichnung des FETs lesen zu können, hätten wir einen Transformator auslöten müssen, wozu wir aber keine wirkliche Lust verspürten.

Die eigentlichen Schalt-MOSFETs sind STD5NM50 (500 V; 0,7 Ω ; 7,5 A) in DPAK-Gehäusen (wieder) von STMicro. Sie sind nicht auf einem Kühlkörper montiert, sondern als SMDs direkt auf die Platine gelötet. Kupferflächen zur Kühlung? Pustekuchen! Dies muss schon eine äußerst effiziente Schaltung sein, da sie in der Lage ist, 90 Watt Leistung einer 100...240-V-Netz-

wechslung in eine Ausgangsspannung von ± 24 VDC zu verwandeln, ohne irgendwo Wärme ableiten zu müssen. Die Schaltung ist sogar eingesperrt in ein geschlossenes Kunststoffgehäuse. Geplante Obsoleszenz?

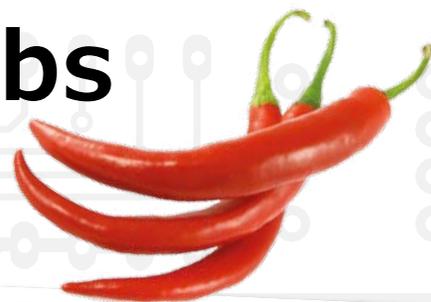
Es stellte sich aber nun doch heraus, dass das Netzteil gut funktioniert und die korrekte Spannung von ± 24 V lieferte. Im Internet haben wir herausgefunden, dass das Apple Cinema Display verdächtigt wird, zu viel Strom aus dem Netzteil zu ziehen, wenn die Energiesparlampen der Hintergrundbeleuchtung in die Jahre gekommen sind und so den Überstromschutz auslösen und das Gerät abschalten. Zwar könnte der Einsatz des schwereren 130-Watt-Netzteils helfen, aber das Problem wäre nicht gelöst, sondern nur vertagt. Eine traurige Bilanz! Da ein neues Cinema-Display bereits bestellt war, um das sieben Jahre tagtäglich genutzte alte Gerät zu ersetzen, haben wir davon abgesehen, auch dort einen Blick hinein zu werfen. Dennoch war es eine schöne Übung und wir wissen jetzt, was in einem Apple Cinema HD Display Power Adapter steckt. ◀

(150022)



Was ist heiß auf .Labs ...außer den Lötkolben?...

Hier eine Auswahl der coolen, großartigen, lustigen und interessanten Projekte auf www.elektor-labs.com.



Knockin' on Mozart's door

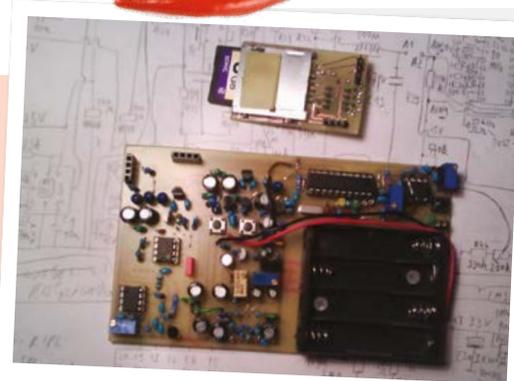
Ist es dies, woran Ihre Gäste denken, wenn sie bei Ihnen klingeln? Mit einem klassischen PIC16F873-Mikrocontroller unterhält die programmierbare musikalische Türklingel Ihre Besucher mit maßgeschneiderten Melodien, während sie warten. Sie können den Zutritt auch verweigern, bevor die Melodie beendet ist. Und für eine bohrlochfreie Installation arbeitet der Klingelknopf (ein Touch-Sensor) sogar drahtlos.



<http://po.st/doorbell>

Auch Zufälle sind nur Berechnung

In der Elektronik folgen die Dinge (auch wenn etwas nicht funktioniert) immer bestimmten (vorhersehbaren) Mustern. Auch in Computer-Algorithmen gibt es keinen echten Zufall (sehen wir einmal vom seltsamen Verhalten einiger Programme ab), sondern nur eine Pseudo-Zufälligkeit. Ein echter Zufallszahlengenerator nimmt das Lawinenrauschen einer Zenerdiode, reinigt das Rauschen, reduziert die Freiheitsgrade und speichert das Ergebnis dann als eine binäre Sequenz auf einer SD-Karte. Glauben Sie nicht, dass dies wirklich zufällig ist? Fragen Sie MATLAB!



<http://po.st/random>

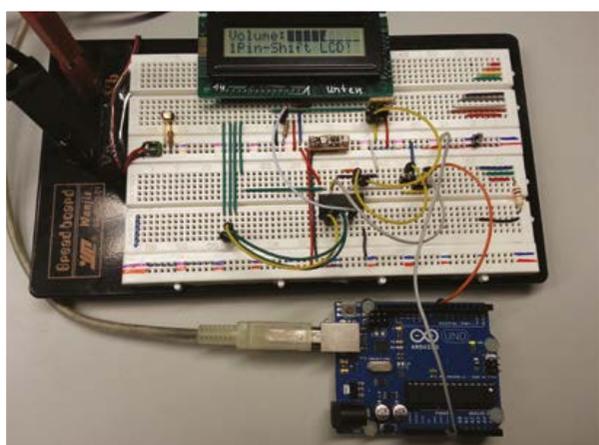
The network is down!

Schon ein wenig geheimnisvoll ... Das Bildchen bei dem Projekt erinnert an ein Spiel oder an einen dieser Computer aus den 60er-Jahre-Science-Fiction-Filmen, den man etwas fragen kann und der antwortet, als hätte er einen Besenstiel in der Röhre. Das Bild ist tatsächlich so unscharf, als ob es mit einer Überwachungskamera aufgenommen worden wäre. Nach der Beschreibung hat das Projekt aber die Aufgabe, den Status Ihres Computer-Netzwerks zu überwachen. Und es geht ziemlich ernst zu, wenn eine Meldung auf Ihrem Handy erscheint, dass ein Problem aufgetreten ist und um was es sich dreht. Wirklich cool wäre es, wenn man dann zurückfragen könnte: „Und, warum hast du es noch nicht gelöst?“

The only one

„Verdammt, mir gehen die I/O-Pins aus ...“
Wahrscheinlich kennen Sie diesen Gedanken, wenn Sie plötzlich feststellen, dass Sie über zu wenig GPIO-Ports am Mikrocontroller verfügen, um die neue Funktion zu implementieren, die der Technische Leiter doch so gerne noch hätte. Sie

können nun auf einen anderen Controller mit mehr Ports migrieren – oder eine Art von Seriell-zu-Parallel-Port-Expander hinzufügen. Dieses Projekt geht noch einen Schritt weiter und reduziert das serielle Interface auf eine Leitung. Sie können alle üblichen alphanumerischen LCDs mit nur einem Port-Pin, einem 74HC595-Schieberegister und ein paar passiven Bauteilen steuern. Das Ganze ist vielleicht nicht so schnell wie eine 8-bit-Parallelschnittstelle, aber gut genug für viele Low-Speed-Anwendungen.



<http://po.st/onewire>



<http://po.st/PINGcube>

Optimal messen mit dem Oszilloskop

Informatives von den Herstellern

Von **Harry Baggen** (Redaktion NL)

Oszilloskope sind leistungsstarke, enorm vielseitige Messgeräte, insbesondere wenn der Anwender damit umzugehen weiß. Zu diesem Thema hält das Internet eine unüberschaubare Flut an Informationen bereit, angeführt von den Webauftritten namhafter Hersteller. Wir haben nachgeschaut, was die Hersteller neben der Präsentation ihrer Produkte an Informativem zu bieten haben.



Zum Inventar jedes Praktikers, der sich mit der Entwicklung oder Wartung elektronischer Systeme befasst, gehören bestimmte, unverzichtbare Arbeitsgeräte. An vorderster Stelle rangiert das Multimeter, dicht gefolgt vom Labornetzgerät und dem Oszilloskop. In vergangenen Zeiten zählten Oszilloskope zu den kostenintensiven Investitionen, doch der technische Fortschritt hat dazu beigetragen, dass sie erschwinglich wurden. Oszilloskop-Vorsätze für den PC sind schon ab etwa 200 € erhältlich, und eigenständige Zweikanal-Oszilloskope, die nicht auf den PC aufsetzen, gibt es dank fernöstlicher Regsamkeit bereits ab etwa 500 €.

Wer zum ersten Mal mit einem Oszilloskop umgeht, tut gut daran, sich mit den Funktionen und Besonderheiten des Geräts eingehend vertraut zu machen. Zu wissen, wie das Oszilloskop das Messsignal verarbeitet, ist nicht nur hilfreich, sondern auch notwendig. Was schon während der Ära analoger Oszilloskope galt, ist im digitalen Zeitalter nicht überholt.

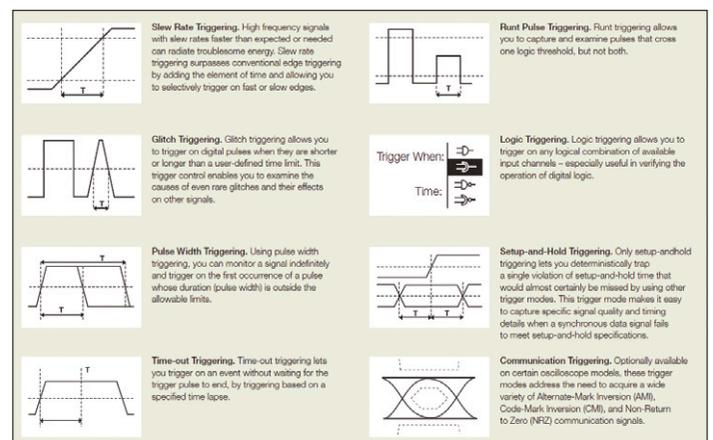
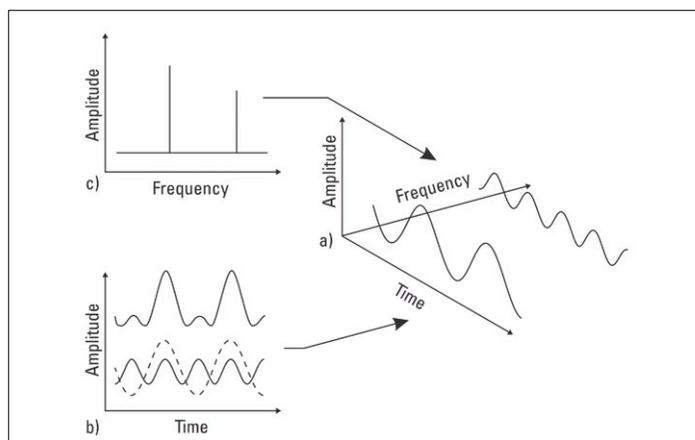
Gemeinsam mit einem Entwickler des Elektor-Labors betrachtete

ich kürzlich ein Signal, das uns zunächst unlösbare Rätsel aufgab. An einem Messpunkt unserer Schaltung hatten wir ein Signal mit total anderer Signalform erwartet. Ursache für das vermeintliche Mysterium waren Interferenzen zwischen dem Messsignal und der Abtastfrequenz des Oszilloskops. Das Messsignal enthielt Anteile, deren Frequenzen mehr oder weniger schnell variierten.

Grund- und Spezialwissen zur Arbeitsweise analoger und digitaler Oszilloskope sowie Anleitungen, Hintergründe, Tipps und Tricks sind im Web reichlich vertreten. Insbesondere die großen Messgeräte-Hersteller bieten als Beigabe zu ihrer Produktwerbung solides und verlässliches Know-how an, das kostenlos und frei Haus verfügbar ist.

Keysight

Der kalifornische Hersteller Keysight, vormals zu Agilent gehörig, hat eine umfangreiche Informationsdatenbank ins Netz gestellt. Der Link *Technical Support* [1] öffnet den Weg zu Tausenden Dokumenten, die nicht nur Datenblätter und Handbücher zu eigenen Produkten umfassen. Die zweiteilige Applikationsschrift



Fundamentals of Signal Analysis ist eine Einführung in die Grundlagen der Signalanalyse mit Oszilloskopen. Der erste Teil [2] beschäftigt sich mit dem Messen im Zeit- und Frequenzbereich, die Prinzipien werden anschaulich an Parallelen aus der Mechanik erklärt. Im zweiten Teil [3] folgt eine Vertiefung des Stoffs, Schwerpunkte sind die FFT sowie das Sampling und Aliasing (**Bild 1**). Das Verständnis dieser Begriffe ist für den Einblick in die Funktionsweise digitaler Oszilloskope von essentieller Bedeutung.

Die Datenbank von Keysight enthält eine Fülle weiterer Dokumente, die es lohnen, einen Blick hineinzuzwerfen. Als Beispiele seien die Veröffentlichungen genannt, die unter dem Label *8 Hints* erschienen sind. Die *8 Hints For Successful Impedance Measurement* [4] und die *8 Hints for Making Better Measurements Using Analog RF Signal Generators* [5] sind nur zwei Highlights unter vielen.

Tektronix

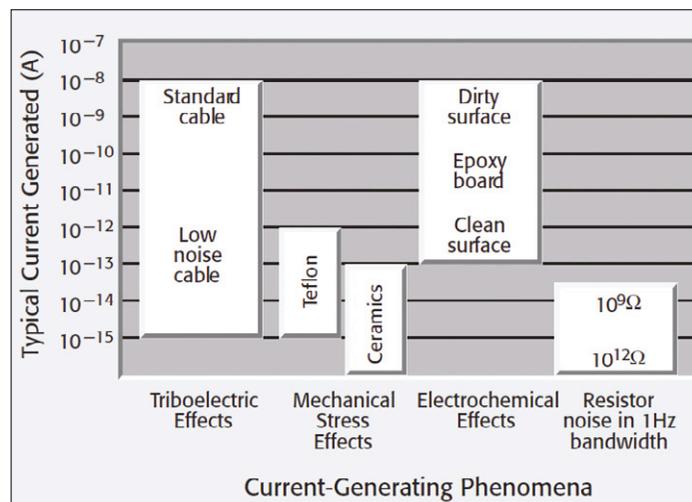
Dieser global agierende Hersteller hat seinen Handbüchern und Applikationsschriften umfangreichen Raum im Web gewidmet [6]. Zuerst sei hier der Titel *XYZs of Oscilloscopes* [7] genannt, ein 59 Seiten umfassendes eBook, das den Leser in kompaktem Stil in die Welt der Oszilloskope führt. Am Anfang steht eine unkomplizierte Darstellung der gängigsten Signalformen. Dann folgt eine Einführung in die Arbeitsweise und den Einsatz heutiger Oszilloskope, und natürlich werden auch die Geheimnisse der Triggerung gelüftet (**Bild 2**). Bevor Dokumente heruntergeladen werden können, erwartet Tektronix eine verbindliche Registrierung mit einer gültigen Email-Adresse. Doch diese kleine Mühe lohnt sich allemal.

Das 60-seitige Dokument *ABC of Probes* [8] hat ein Zubehöriteil zum Thema, dessen Bedeutung oft unterschätzt wird. Diese Broschüre beschreibt, wie Oszilloskop-Tastköpfe aufgebaut sind und welche Typen existieren. Die Rückwirkungen auf die zu messenden Signale werden ebenso behandelt wie die Wahl des optimalen Tastkopfs abhängig von den Messbedingungen. Über die Website von Tektronix gelangten wir zu Keithley, ein Unternehmen, das inzwischen zu Tektronix gehört. Bei Keithley fanden wir ein bemerkenswertes, 240 Seiten starkes eBook, in dem es allerdings nur mittelbar um Oszilloskope geht. Der Titel *Low Level Measurements Handbook* spricht für sich, hier steht generell das Messen kleiner Signale im Mittelpunkt (**Bild 3**). Dieses eBook ist ein wirklicher Tipp, es hat unsere unbedingte Empfehlung. Auch Keithley ist daran gelegen, vor dem Download die Email-Adresse des Besuchers zu erfahren.

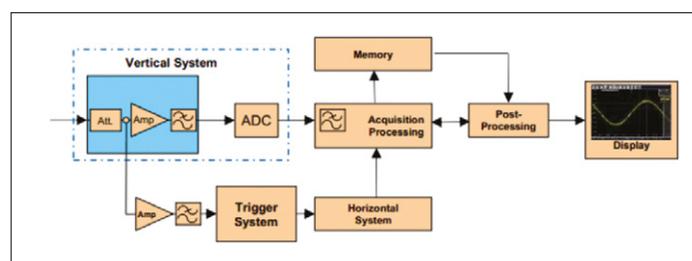
Außerdem

Viel informatives Material zum Thema Oszilloskope halten auch andere große Messtechnik-Hersteller bereit. Bei Teledyne LeCroy fanden wir eine aufschlussreiche Abhandlung über den Einsatz von Tastköpfen, das *Probing Tutorial* [10]. Vornehmlich wird hier betrachtet, wie die Eigenschaften des Tastkopfs die Messergebnisse beeinflussen oder sogar verfälschen können. Ein Dokument, das sich an den weiter fortgeschrittenen Praktiker wendet, ist der Aufsatz *Why Differential?* [11]. Das Dokument erklärt, weshalb differentielle Messverfahren den üblichen Methoden überlegen sind.

Bevor wir zum Schluss kommen, darf das Traditionsunternehmen Rohde & Schwarz nicht unerwähnt bleiben. Auch dort fanden wir umfangreiche Lektüre, die unter dem Titel *Oscilloscope*



3



4

Fundamentals [12] Grundwissen zur Technik und Anwendung von Oszilloskopen vermittelt. Das in sich abgerundete Dokument führt den Leser vom allgemeinen Blockschema (**Bild 4**) bis hin zum Verständnis hochspezieseller Funktionen und Anwendungen. Ein Kapitel zu Oszilloskop-Tastköpfen fehlt ebenso wenig wie ein Blick auf die geschichtliche Entwicklung der Oszilloskope.

Unser Exkurs ins Web hat gezeigt, dass unzählige technische Informationen solider Qualität auf ihren Download warten. Die bekannten Messtechnik-Hersteller stehen mit ihrem Namen dafür ein, dass die Inhalte sachlich bis ins Detail korrekt sind. Was im Web nicht immer selbstverständlich ist... ◀

(150032)gd

Weblinks

- [1] www.keysight.com/main/facet.jspx
- [2] <http://literature.cdn.keysight.com/litweb/pdf/5988-6765EN.pdf>
- [3] <http://literature.cdn.keysight.com/litweb/pdf/5988-6774EN.pdf>
- [4] <http://literature.cdn.keysight.com/litweb/pdf/5968-1947E.pdf>
- [5] <http://literature.cdn.keysight.com/litweb/pdf/5967-5661E.pdf>
- [6] www.tek.com/learning/oscilloscope-tutorial
- [7] <http://info.tek.com/www-xyzs-of-oscilloscopes-primer.html>
- [8] <http://info.tek.com/www-abcs-of-probes-primer.html>
- [9] <http://info.tek.com/KI-Low-Level-Measurements-Handbook-LP.html>
- [10] http://cdn.teledynelecroy.com/files/appnotes/lecroy_probing_tutorial_appnote016.pdf
- [11] http://cdn.teledynelecroy.com/files/whitepapers/wp_differential_measurements.pdf
- [12] www.rohde-schwarz-usa.com/rs/rohdeschwarz/images/Oscilloscope-Fundamentals_v1.1.pdf

Zusammengestellt von **Beatriz Sousa**

Elektor verhilft E-Talenten zu neuen Jobs!

Nach neuesten Erkenntnissen der *Harvard Management Review* sind 25 % der Spitzen-Geschäftsführer weltweit Ingenieure. Die Unternehmen sind immer mehr bestrebt, Elektronik-Talente einzusetzen. Deshalb führt Elektor einen neuen Mitgliederservice ein. Wir stellen Unternehmen vor, die Ingenieure einstellen möchten, um unseren Mitgliedern und Lesern eine unabhängige Bewertung zu ermöglichen. E-Talente, die sich davon angesprochen fühlen, richten bitte eine kurze E-Mail-Anfrage an unser Serviceteam. Elektor ist dann in der Lage, zwischen beiden Parteien zu vermitteln. Also: Wenn Ihr Unternehmen plant, E-Menschen einzustellen, kontaktieren Sie bitte unsere Anzeigenabteilung.

elektor•talents

len möchten, um unseren Mitgliedern und Lesern eine unabhängige Bewertung zu ermöglichen. E-Talente, die sich davon angesprochen fühlen, richten bitte eine kurze E-Mail-Anfrage an unser Serviceteam. Elektor ist dann in der Lage, zwischen beiden Parteien zu vermitteln. Also: Wenn Ihr Unternehmen plant, E-Menschen einzustellen, kontaktieren Sie bitte unsere Anzeigenabteilung.

Danke für Ihren Besuch!

Elektor war mit einem großen Team auf der **Embedded World** vom 24. bis zum 26. Februar in Nürnberg vertreten. Wie versprochen war unser Stand noch größer und aufregender als bei der Messe im Jahr zuvor; mitgebracht hatten wir einige T-Boards, die Propelleruhr, unser FPGA-Board, das Arduino Extension-Shield und viele andere Produkte. Viele Mitglieder nutzten die Chance, mit uns zu fachsimpeln, auch über kommende Projekte. Sie erhielten ein kostenloses Exemplar unseres neuen *Business Special* sowie ein LPC800-Entwicklungskit von NXP. Wir hoffen, Sie bald wieder zu sehen!

READ ONLY MEMORY

Unser Magazin kann auf eine lange Geschichte zurückblicken. In diesem Kasten zeigen wir stolz Vergangenes aus alten Tagen.

Laptops für alle!

Bereits im Jahr 1980 hat Elektor den Junior-Computer veröffentlicht, einen PC, den jedermann und jedefrau mit einem Lötkolben und etwas Geschick zu Hause nachbauen konnte.

Der Junior wurde als Bausatz mit eigens von Elektor produzierten Platinen verkauft. Er basierte auf einem 6502-Prozessor und konnte mit einem Floppy-Laufwerk erweitert werden.

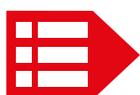
Viele Menschen haben damals den Junior gebaut!



Sonderhefte, frisch aus der Redaktion



Das erste Elektor-Special handelt von Drohnen. Der Chefredakteur Dre de Man und sein Team von Spezialisten stellt alles, was man über die rasant wachsende Drohnentechnologie wissen muss, in diesem Heft zusammen. Das Drohnen-Sonderheft erscheint passend zur Flugsaison am ersten Tag des Sommers.



LEUTE Der Autor Bert van Dam stellt sein Buch über Intels Edison fertig • Die Versandabteilung, während der Raspberry-Pi-Kampagne den Rekord an Postsendungen/Tag • Johan Bergqvist Kompendiums The Bottle Builder • Thom Wolf hat seine Arbeit als Kundenmanager in den Politis ist neuer Einkaufsleiter von Elektor • Jan Schrader unterstützt das deutsche Team

Ingenieure sind an vielen Themen interessiert: 3D-Druck, Software und Programmierung, Sicherheit, LEDs, Robotik... Deshalb erstellt die Elektor-Redaktion neue Sonderhefte zu „heißen“ Themen.



Achten Sie auf Ihre Mailbox:

Mitglieder erhalten die Chance, das Drohnen-Sonderheft mit 10 % Mitgliederrabatt zu kaufen. Und wenn Sie Vorschläge für neue Specials haben, lassen Sie es uns wissen.

namentlich Jan Aspers und Theo, brachen beendet Band 1 seines dicken Röhrenverstärker-Niederlanden und Belgien aufgenommen • Alex bei der Veröffentlichung weiterer Specials.

EXPERTENPROFIL

Elektor arbeitet mit mehr als 1.000 Experten und Autoren bei der Produktion von Büchern, Artikeln, DVDs, Webinaren und Live-Events zusammen. In jeder Ausgabe wollen wir einen von ihnen ins Rampenlicht stellen...



Name: **Dr. Günter Spanner**

Alter: **49**

Titel: **Doktor der Physik**

Publikationen: **Fachartikel in vielen Fachzeitschriften. Mehrere Patente auf dem Gebiet der biomedizinischen Technik. Mehr als zehn Bücher und Publikationen auf dem Gebiet der**

Sensortechnik und Mikroelektronik.

Tätigkeit: **Lehrbeauftragter an mehreren Instituten und Fachhochschulen in Europa.**

Wer ist Günter Spanner?

Nach meiner Promotion an der Technischen Universität München bin ich zum Corporate Research Center der ABB in der Schweiz sowie zur ETH Zürich als wissenschaftlicher Mitarbeiter gewechselt. Bei Siemens fungierte ich als wissenschaftlicher Experte auf dem Gebiet der Faseroptik-Telekommunikation und der Sensoranwendungen. Derzeit lebe ich in einem kleinen Dorf zwischen München und den Alpen.

Was sind Ihre Berufserfahrungen?

Ich habe vor allem in der Lichtwellenleiter-Anwendung, Lasertechnik und optischen Sensorik Erfahrung. Bei Siemens war ich für die Systementwicklung, Produktionstechnik und Compliance verantwortlich. Daneben leitete ich mehrere Entwicklungsgruppen in Forschung und Entwicklung.

Was wird in der Elektronik die nächste Schlüssel-Entwicklung sein?

Ich denke, die Bio-Elektronik. Derzeit sind zwei große Projekte im Gang, BRAIN in den USA und das Human Brain Project in Europa. Hoffentlich werden sie zu technologischen Durchbrüchen in der Bio- und neuronalen Elektronik führen.

Über welche Themen werden Sie in Zukunft schreiben?

Im Moment arbeite ich an Mikrocontrollern, Sensorik und Physical Computing. Der nächste große Schritt wird ein umfassender Kurs über die ATXmega-Controller sein.

Angenommen, Elektor gibt Ihnen 100 €.

Was würden Sie kaufen? Und warum?

Ich würde das Geld in einige drahtlose IoT-Geräte stecken. Diese kommende Internet-der-Dinge-Technologie hat enormes Potential für neue Anwendungen in der Home-Automation sowie in vielen Industriebereichen.

Wer ist Ihrer Meinung nach der größte Konkurrent des Arduino?

Der Raspberry Pi. Für etwa den gleichen Preis erhalten Sie gleich ein kleines komplettes Mainboard inklusive einer Reihe von I/O-Ports.

Hexadoku

Sudoku für Elektroniker

Haben Sie diese Elektor schon ausgelesen, von A bis Z? Oder haben Sie zum Entwerfen, Löten und Programmieren gerade keine besondere Lust? In jedem Fall ist unser monatliches Hexadoku wieder eine willkommene Ablenkung. Geben Sie Ihr Bestes und Sie haben die Chance, einen von fünf Elektor Buchgutscheinen zu gewinnen!

Die Regeln dieses Rätsels sind ganz einfach zu verstehen: Bei einem Hexadoku werden die Hexadezimalzahlen 0 bis F verwendet, was für Elektroniker und Programmierer ja durchaus passend ist.

Füllen Sie das Diagramm mit seinen 16 x 16 Kästchen so aus, dass alle Hexadezimalzahlen von 0 bis F (also 0 bis 9 und A

bis F) in jeder Reihe, jeder Spalte und in jedem Fach mit 4 x 4 Kästchen (markiert durch die dickeren schwarzen Linien) **genau einmal** vorkommen. Einige Zahlen sind bereits eingetragen, was die Ausgangssituation des Rätsels bestimmt. Wer das Rätsel löst - sprich die Zahlen in den grauen Kästchen herausfindet - kann einen von fünf Wertgutscheinen 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-Verlag GmbH
Redaktion
Süsterfeldstr. 25
52072 Aachen
Fax 0241 / 88 909-77
E-Mail: hexadoku@elektor.de

Als Betreff bitte nur die Ziffern der Lösung angeben!

Einsendeschluss ist der 30. April 2015.

Die Gewinner des Hexadokus aus der Januar/Februar-Ausgabe stehen fest!

Die richtige Lösung ist: **3146A**

Einen Elektor-Wertgutschein über je 50 € haben gewonnen:

Yngve Bordoy, Harrie Rouwhorst, Hartwig Wander, Jairo Rotava und Jan Verdois.

Herzlichen Glückwunsch!

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

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

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

Professionelle Qualität zu attraktiven Preisen!



- ✓ über 45 Jahre Erfahrung
- ✓ schneller 24-Stunden-Versand
- ✓ über 50.000 Produkte am Lager
- ✓ kein Mindermengenzuschlag



Jetzt kostenlos anfordern!

Der Neue ist da!
Katalog 06|2015

Oder gleich reinschauen:
online blättern!



<http://rch.lt/15-1>



- mehr als 50.000 Artikel aus Elektronik & IT
- über 1.300 Seiten
- mehr als 5.000 Neuheiten

<http://rch.lt/kat>

Telefonisch anfordern unter:
04422 955-333

Löt- und Entlötstation



60/80 W

- ☑ ESD geschützt
- ☑ keramisches Heizelement
- ☑ schnelle und präzise Temperaturregelung



159,95
STATION ZD-987



Tischlupenleuchte mit 48 LEDs

A+

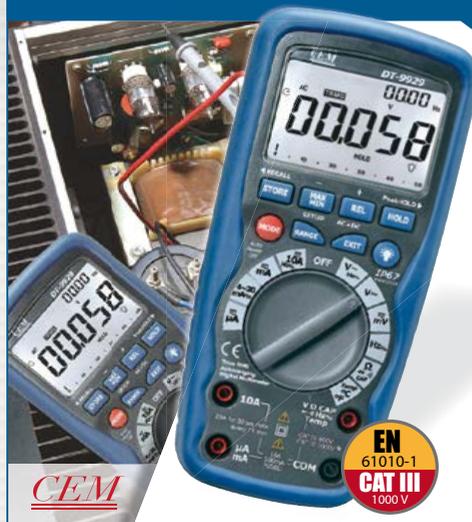
- 1,75 fache Vergrößerung
- 3 Dioptrien
- fester, stabiler Standfuß



27,95

LL ZD-2012B-2H

TRMS-Digital-Multimeter mit Wireless-USB



Robustes, wasserdichtes Universalmessgerät für präzise Messungen.

- Grundgenauigkeit: 0,06%
- Anzeigebereich: 40.000 Counts
- Echteffektivwertmessung (TrueRMS)
- umfangreiche Messfunktionen sowie Zusatzfunktionen
- Daten-Speicher- und Lesefunktion (9999 Speicher)
- PC-Anschluss per Wireless-USB (inkl. Software)
- Messkategorie: CAT III 1000 V, CAT IV 600 V



154,00
CEM DT-9939

Labor-Kompaktnetzgerät mit USB-Schnittstelle

- ☑ komplett digitalisierte Steuerung
- ☑ Überlastschutz
- ☑ große LED-Anzeige für Ausgangswerte
- ☑ Konstantspannungs- und Konstantstrombetrieb

- Ausgang: 0 - 30 V DC / 0 - 5 A
- Feineinstellung von Spannung Strom
- geringe Restwelligkeit
- USB-/RS-232-Schnittstelle für PC-Fernsteuerung



89,00
KD3005P

Preise in € inkl. MwSt., zzgl. Versandkosten, Stand: 02.03.2015
reichelt elektronik GmbH & Co. KG, Elektronikring 1, 26452 Sande

Jetzt bestellen!

@reichelt_el reichelt.de

www.reichelt.de

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



Newsletter
Jetzt abonnieren & gewinnen
<http://rch.lt/news>

Alles drin: HMO Complete

HAMEG®
Instruments
A Rohde & Schwarz Company

Noch bis zum 30.06.2015: Sparen Sie bis zu 40% beim Kauf eines HMO Complete-Bundles, bestehend aus einem HMO3000 wahlweise mit zwei oder vier Kanälen, 500 MHz Bandbreite, allen Software-Optionen und zwei Digital-Tastköpfen.

Neugierig?
www.hameg.com

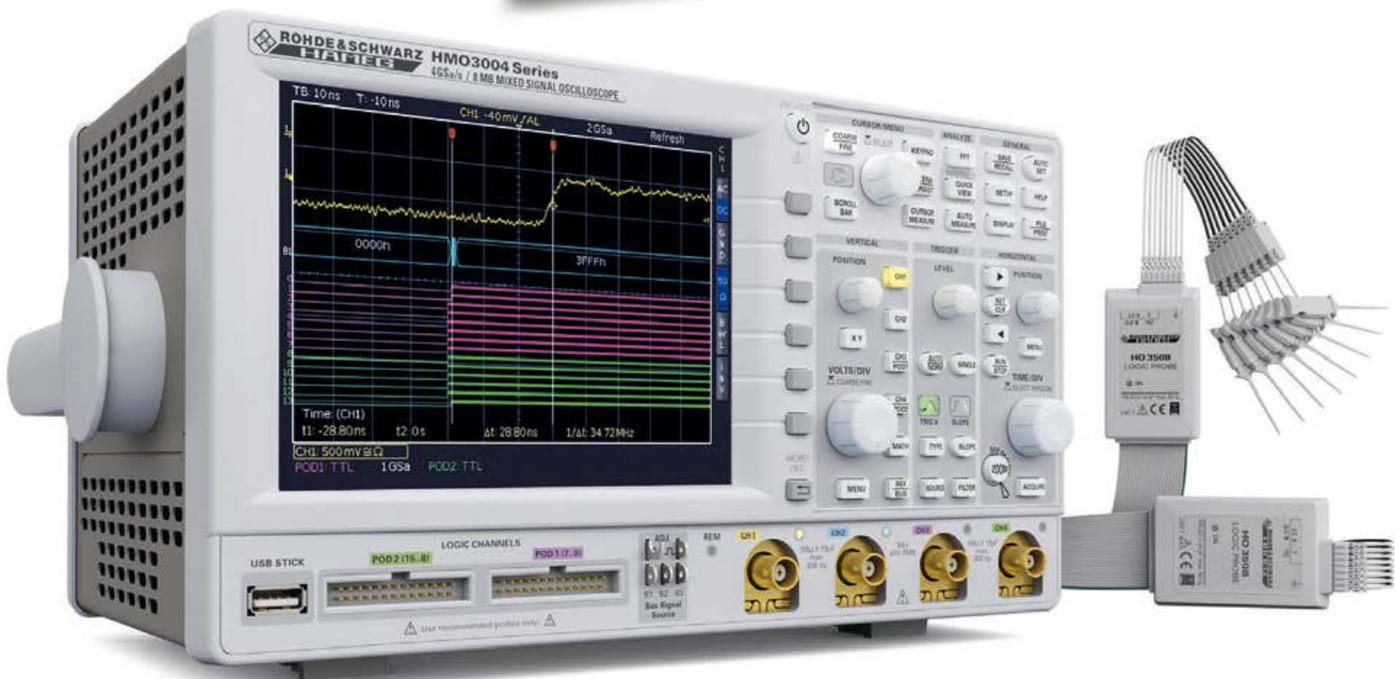
Noch bis zum
30.6.
2015

inklusive H0010
Busanalyse für I²C, SPI, UART

inklusive H0012
Busanalyse für CAN/LIN

inklusive H0014
Segmentierter Speicher

inklusive H03516
2x 8-Kanal Digital-Tastköpfe



[f](#) [+](#) [YouTube](#) [#HAMEGcom](#)