



elektor

LEARN • DESIGN • SHARE



VFD-Shield
für Arduino



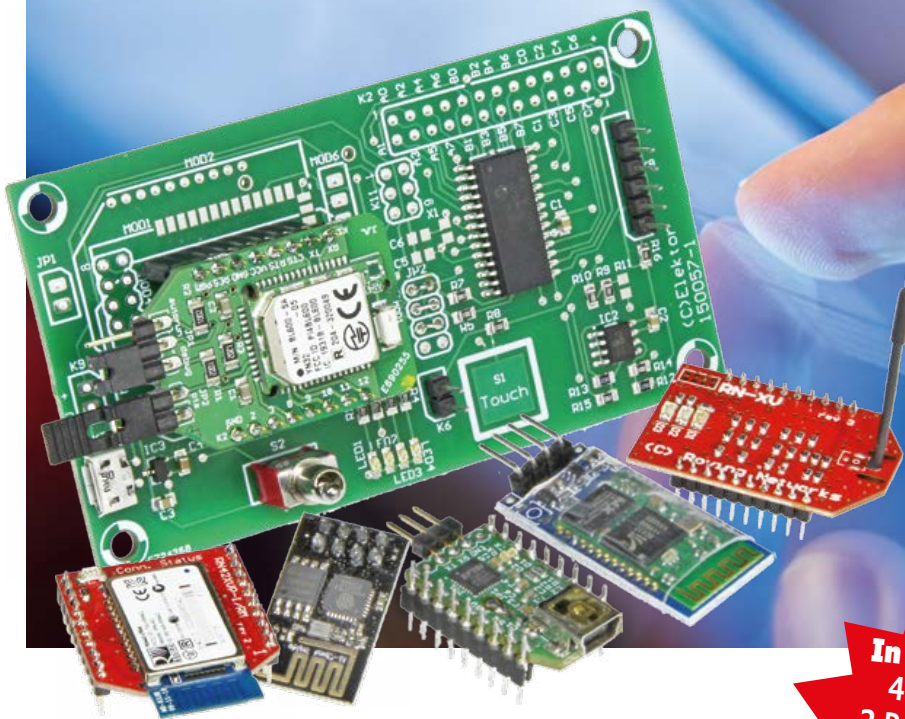
AC/DC-
Leistungsmesser



Batterie-Ersatz
mit USB

Android I/O-Board

Eigene Elektronik steuern mit
Android-Smartphones und -Tablets



ARM-Controller für Einsteiger (7): Funktionen für Fortgeschrittene • **Scripting mit EAGLE** • **Assembler Crashkurs:** Elektronischer Würfel • **Bluetooth Low Energy Modul:**

Events • **Prototyping mit Bricks und Bricklets** • **Web-Recherche:** SMD-Codierungen • **LED-Treiber mit großem Ausgangsspannungsbereich** • **Bemerkenswerte Bauteile:** Koaxialverbinder • **Gerber-Dateien aus dem Elektor-Labor**

In dieser Ausgabe:
4 Labor-Projekte
2 Programmierkurse
1 CAD-Kurs
1 Review
und noch viel mehr!

KCS TraceME

2G 3G 4G LBS

LoRa™ BLE M2M

Iridium Sensor



iBeacon™

Glomass GPRS

RF GPS

Bluetooth®

SMS

Internet of Things



LoRa™ für das Internet of Things

KCS hat ihre erfolgreiche Produktlinie TraceME mit einem leistungsfähigen Modul erweitert, das für weltweite Mobilität im Zeitalter des Internet of Things sorgt. Die neueste Entwicklung des GPS/GPRS-Track-and-Trace-Moduls der TraceME-Reihe kombiniert auf Basisstationen basierende Positionierung mit der Technologie LoRa™.

Diese Technologie macht „smart objects“ noch smarter, da LoRa™ energiesparende Weitbereichs-Kommunikation für viele (M2M-) Anwendungen ermöglicht. Es wird dabei GPRS/SMS sowie optional 3G, WLAN, Bluetooth LE, ANT/ANT+ und iBeacon™ providerunabhängig mit existierenden Netzwerken und mobilen Apps. Weitere Varianten im oberen und mittleren sowie Budget-Segment werden in Bälde folgen.

Diebstahlsschutz mit HF-Technik

Die TraceME-Produktlinie von KCS bietet nun ein intelligentes, ortsbasiertes Positionssystem als Diebstahlsschutz für den häuslichen Bereich. Es handelt sich um eine HF-Lösung mit einem intelligenten Algorithmus zur Messung der Laufzeit der gesendeten proprietären Signale. Besonders hervorzuheben sind: Geringe Abmessungen (nur 46x21x6,5mm), niedriges Gewicht (nur 7 g für die voll bestückte Platine) und eine Lebensdauer der Batterie von mehr als 10 Jahren. Ein „Listen before talk“-Algorithmus macht es praktisch unmöglich das Modul zu lokalisieren, das wertvolle Fahrzeuge oder andere Vermögenswerte schützt. Das Modul unterstützt GPRS/SMS und optional 3G, WLAN, Bluetooth LE, ANT/ANT+ sowie iBeacon und erlaubt die einfache Integration in bestehende Funknetzwerke und mobile Apps.

www.Trace.ME

Alle erwähnten Markenzeichen gehören den jeweiligen Eigentümern

Impressum

46. Jahrgang, Nr. 537 September 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



Neulich auf dem Open Air

Köln ist nicht umsonst eine meiner Lieblingsstädte: Hier gibt es Veranstaltungen, auf denen man gleichzeitig gute *Elektronische* Musik hören und interessante Leute treffen kann, deren Schulabschluss schon einige Jährchen zurück liegt. Klar, dass mitunter auch die Frage fällt: „Und womit verdienst du so dein Geld?“. Ganz stolz darf ich dann berichten, dass ich als Zeitschriften-Redakteur mit Sicherheit einen der schönsten Berufe der Welt ausübe. „Ja – und bei welcher Zeitschrift bist du?“

Natürlich freue ich mich dann, wenn nicht gerade zurückgefragt wird: „äh, Electro?“. Und gerade auf Partys, wo sich viele Musik- und Medien-Schaffende herumtreiben, habe ich oft Glück. Denn einige dieser Kreativen haben zumindest in ihren Jugendjahren so manchen Elektor-Bausatz zusammengelötet. Und mein Gegenüber freut sich dann, dass es uns tatsächlich noch gibt. Neugierig lasse ich mir dann auch dessen „Produkte“ zeigen. Zum Beispiel dieses hier: www.fantastic5.tv. Wirklich schön gemacht, schauen Sie mal rein (gern habe ich hiermit etwas Werbung für euch gemacht, Gunther ;-)). Natürlich ist da auch ein kleiner Wermutstropfen dabei: Mit den ständig wachsenden Möglichkeiten, die uns die Computertechnik bietet, haben eben viele unserer ehemaligen Leser das Hobby gewechselt und mitunter zum Beruf gemacht. Umso schöner ist es jetzt zu sehen, dass viele junge Programmier-Nerds zur Elektronik zurückkehren und sich plötzlich für LEDs, Sensorschaltungen und das Lötén interessieren. Alle frischgebackenen Maker heißen wir herzlich willkommen – auf allen Kanälen. Denn neben interessanten Zeitschriftenartikeln wollen wir in Zukunft auch viel mehr lehrreiche Webseiten und natürlich Videos produzieren, in denen wir zeigen, wie Elektronik funktioniert. Bleiben Sie dran!

Jens Nickel
Chefredakteur Elektor



Unser Team

Chefredakteur:	Jens Nickel (v.i.S.d.P.) (redaktion@elektor.de)
Ständige Mitarbeiter:	Dr. Thomas Scherer, Rolf Gerstendorf
Leserservice:	Ralf Schmiedel
Korrekturen:	Malte Fischer
Internationale Redaktion:	Harry Baggen, Jan Buiting, 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
- 24 News
- 26 Business Report
LED-Treiber mit großem
Ausgangsspannungsbereich
- 74 Elektor World News
- 82 Hexadoku
Sudoku für Elektroniker

LEARN

DESIGN

SHARE

- 8 Willkommen bei LEARN
- 9 Bemerkenswerte Bauteile
Koaxialverbinder
- 10 Von 8 auf 32 bit:
ARM-Controller für Einsteiger (7)
Funktionen für Fortgeschrittene
- 16 EAGLE Tipps & Tricks (1)
EAGLE mit ULPs individualisieren
- 18 Assembler Crashkurs (2)
Mini-Dev-Board und elektronischer Würfel

LEARN

DESIGN

SHARE

- 29 Willkommen bei DESIGN
- 32 Android I/O-Board (1)
Eigene Elektronik mit Android-Smartphones
und -Tablets steuern
- 40 Batterie-Ersatz mit USB
Ersetzt die AAA-Zelle
- 42 AC/DC-Leistungsmesser
Hohe Genauigkeit, weiter Messbereich
- 50 e-BoB BL600 (6)
Low Energy, Events für Timer,
Datenübertragung und Verbindung
- 56 VFD-Shield für Arduino
Mit verschiedenen Programmbeispielen

VFD-SHIELD FÜR ARDUINO

MIT INSTRUKTIVEN PROGRAMMBEISPIELEN

VFD bedeutet Vakuum-Fluoreszenz-Display: Hier sind es vier VFD-Siebensegment-Ziffernröhren aus russischer Fertigung, die von einem Arduino Uno gesteuert werden. Für eine stilgemäße LED-Beleuchtung ist ebenfalls gesorgt. Zum Repertoire der Programmbeispiele gehören eine Demo, eine Digitaluhr, ein Voltmeter und ein Thermometer.

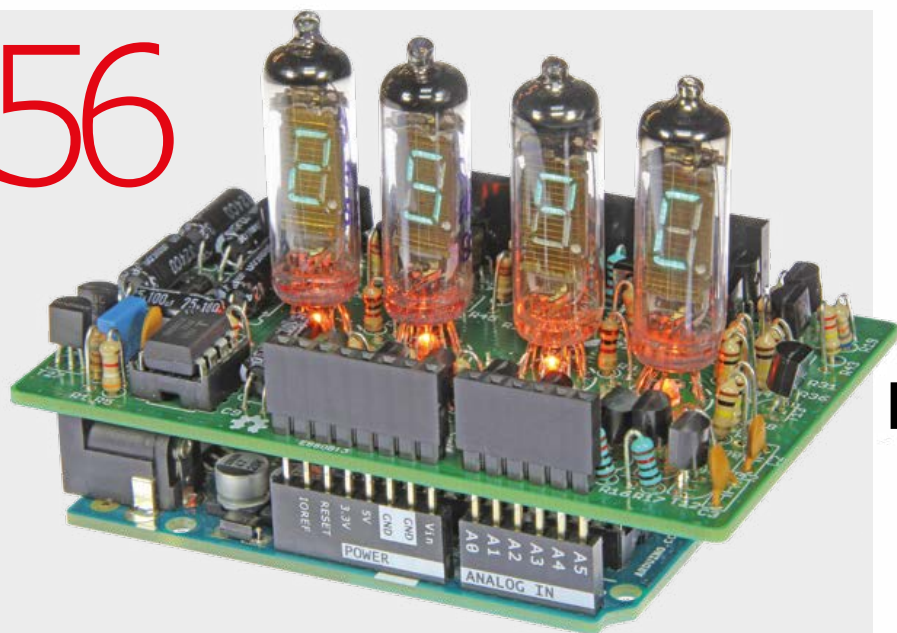


42 AC/DC- LEISTUNGSMESSE

HOHE GENAUIGKEIT,
WEITER MESSBEREICH

Wenn Messungen an Geräten durchgeführt werden, die mit dem Stromnetz in Verbindung stehen, ist die galvanische Trennung des Messsystems ein Gebot der Sicherheit. Dieser Leistungsmesser, der auch Spannungen und Ströme misst, erfüllt die Vorgaben der Schutzklasse II. Ein Arduino Uno und das bekannte Extension-Shield von Elektor sind für die Ausgabe der Messergebnisse auf einem LC-Display zuständig. Auch ein Multimeter oder ein Oszilloskop sind anschließbar.

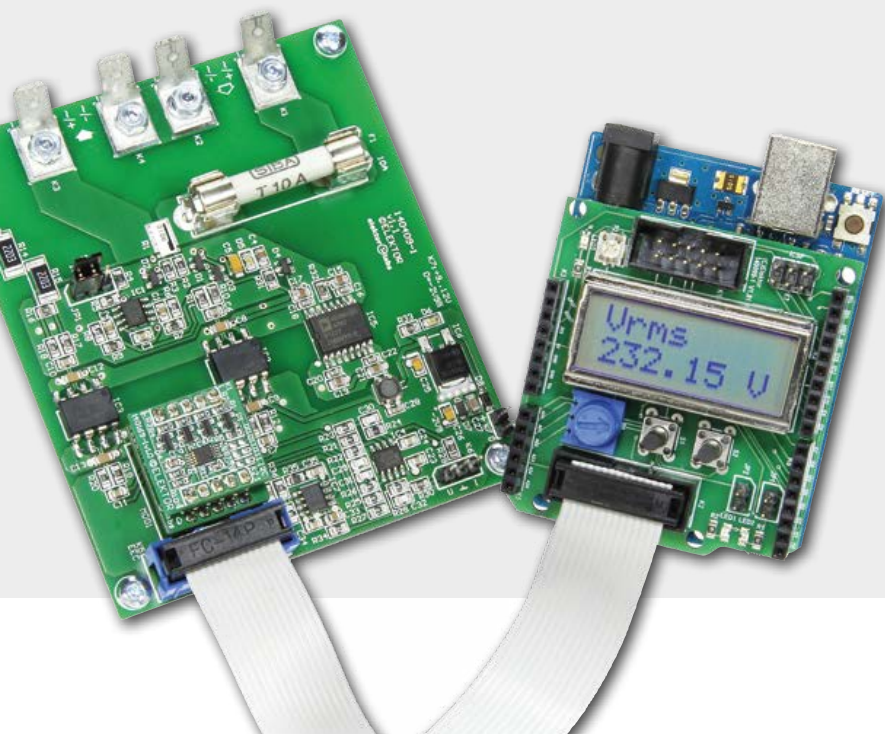
56



ANDROID I/O-BOARD

EIGENE ELEKTRONIK MIT ANDROID-SMARTPHONES
UND -TABLETS STEuern

32



LEARN

DESIGN

SHARE

61 Willkommen bei SHARE

62 Elektronische Bauklötze

Rapid (e-)Prototyping mit Bricks und Bricklets
von TinkerForge

66 Projekt 2.0

Korrekturen, Updates, Erweiterungen

68 Elektor-Labs.com

70 Aus dem Labor

Gerber-Dateien aus dem Elektor-Labor

72 Web-Recherche

SMD-Codierungen

Vorschau

Audio-Messbox MEASSY

Ein kompaktes Gerät für Messungen an Lautsprechern, das mit einem hochwertigen Mikrofonverstärker mit Phantom-speisung und einer kleinen Audio-Endstufe ausgestattet ist.

Software Defined Radio mit Red Pitaya

Hier zeigt sich die Vielseitigkeit des signalverarbeitenden Red Pitaya Boards: Mit einer zusätzlichen kleinen Vorstufe und der passenden Software lassen sich UKW-Sendungen in Stereo empfangen. Sogar RDS wird decodiert!

Kompakter 60-W-Audioverstärker

Die diskret aufgebaute Audio-Endstufe vereint gleich mehrere Vorteile. Sie ist kompakt, enthält ausschließlich gängige Bauteile, ist einfach nachzubauen und bietet dennoch einen hervorragenden Klang.

Änderungen vorbehalten.

Elektor Oktober 2015 erscheint am 23. September 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.

57

Länder

246817

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

29

Autoren

477

Literatur

233521

Monatliche Besucher

10:33

04. AUGUST 2015

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, 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

**Treten Sie dem weltweit größten
Elektroniker-Netzwerk bei!**



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

GOLD

127,50 €/Jahr

- ✓ 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/gold-mitglied

GREEN

92,50 €/Jahr

Empfehlung

- ✗ 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

GRATIS

- ✗ 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/newsletter

Willkommen bei **LEARN**



Von **Jens Nickel**

Auf der Maker Faire

Wenn Sie diese Zeilen lesen, liegt die Maker Faire in Hannover schon einige Zeit zurück. Hier an dieser Stelle aber noch ein Dankeschön an die Kollegen von der „Make“: Es war wirklich ein tolles Event, sogar schönes Wetter hattet ihr organisiert! Besonders

die Begeisterung der (ganz) jungen Elektronik machte Laune. Die acht Lötstationen unseres Workshops (Foto) wurden erst dann kalt, als auch der letzte „Elektronische Würfel“ bestückt war. Interessante neue Boards und Projekte gab es ebenfalls zu sehen. Aufgefallen sind mir zum Beispiel die Hardware-Module von Brick ‚R‘ Knowledge (www.brickrknowledge.de), die sich besonders gut (aber nicht nur) für die Ausbildung eignen. Ein spannendes Projekt ist auch die „SenseBox“. Zu diesem (IoT-)Sensorkit gehört eine Internetplattform zum Austausch von gesammelten Daten wie

Luftdruck, Temperatur, Luftfeuchte und Lichtstrahlung. Inzwischen gibt es zwei Versionen des Kits, eine für den Ausbildungsbereich und eine für DIYer.



Programmieren lernen (2)

Wie schon zum „(Hoch-)Schulboard“ haben mich auch zum Editorial „Programmieren lernen“ etliche Mails erreicht, vielen Dank dafür! Unter anderem machte mich ein Leser auf das „Project Oberon“ von Niklaus Wirth aufmerksam. Von der FPGA-basierten Hardware hatte ich bis dahin wirklich noch nie etwas gehört (siehe zum Beispiel www.xilinx.com/publications/archives/xcell/Xcell91.pdf, ab Seite 30). Eine andere Empfehlung lautete, ein

Launchpad von TI (wirklich günstig!) zu verwenden. Ein weiterer Leser sucht (wie ich) ein Board, das mit C# programmierbar ist, um für den PC und den Embedded-Bereich nur noch eine Programmiersprache verwenden zu müssen. Vielleicht kann uns beiden ja demnächst mit einem Raspberry Pi 2 geholfen werden, auf dem Windows läuft. Ich bin schon gespannt.

Da war doch was

Und noch eine weitere Folge aus der Serie, für die mir das Material nicht ausgehen will: „Doofe Fehler, die einem beim Entwickeln aufhalten“. Endlich wollte ich die MIDI-Checker-Anwendung aus dem letzten Heft vom Arduino Uno auf unser Xmega-Webserver-Board portieren. Meinen EFL-Konfigurator hatte ich dazu noch aufgebohrt, so dass er mir auf Knopfdruck ein schönes Xmega-Projekt mit allen benötigten Files anlegte, und auch der hardware-unabhängige Quellcode des MIDI-Checkers war dann schon drin. Jetzt müsste ich nur noch die Nummer der UART-Schnittstelle im Code ändern, an die das MIDI-Modul angeschlossen wurde. Das Kompilieren

verlief fehlerfrei, doch was war das? Die Power-LED auf dem MIDI-Modul blieb dunkel, als ich es an das Xmega-Board anschloss. Ich suchte nach einem zweiten Board, hier funktionierte es. Ein Blick in den Schaltplan offenbarte, warum: Auf dem ersten Board fehlte ein Jumper, den ich in der Zwischenzeit einfach nicht mehr „auf dem Schirm“ hatte. Gelernt habe ich diesmal, dass man immer noch mal in die Doku gucken sollte, bevor man Hardware erneut in Gang setzt. Auch wenn man meint, ein Board schon aus dem Effeff zu kennen. ◀

(150319)

Koaxialverbinder

Bemerkenswerte Bauteile

Von **Neil Gruending** (Kanada)

Nicht zufällig sind Koax- oder koaxiale Steckverbinder ideal für HF-Anwendungen: Der heiße Anschluss in der Mitte, drum herum der Masseanschluss und alles so bemessen, dass der Verbinder genau die richtige Impedanz für das HF-Signal hat. Im Laufe der Jahre hat sich eine Vielzahl von Steckertypen etabliert; lassen Sie uns einen Blick auf einige klassische Typen und ihre Verwendung werfen.

Unerwarteterweise ist der häufigste und uns vertrauteste Koax-Verbinder derjenige, der seit vielen Jahrzehnten für audio-visuelle (A/V) Anwendungen verwendet wird: der RCA- oder Cinch-Stecker (**Bild 1**) mit einer Bandbreite von etwa 10 MHz. Das ist mehr als genug, um analoge Audio- und Videosignale in Heimkinos zu übertragen.

F-Stecker und F-Buchsen werden für Kabel- oder Satelliten-TV und Internet eingesetzt (**Bild 2**). Die Bandbreite ist viel höher, bis zu 1 GHz in manchen Fällen, und so perfekt für die höheren Frequenzen, wie sie zum Beispiel ein LNB liefert. Aber es gibt mehr koaxiale Steckverbinder als die in Ihrem Heimkino.

Einer der ersten Koaxialverbinder wurde in den 1930er Jahren entwickelt und wird normalerweise als UHF- oder (Amphenol) SO-239-Typ bezeichnet. Diese Verbinder sind sehr robust und auch für relativ hohe Leistungen geeignet, wie sie im Amateur- und CB-Funk auftreten, ideal für Antennenanschlüsse. Die obere Frequenzgrenze für die UHF-Steckverbinder liegt bei etwa 200 MHz, bei höheren Frequenzen ist die Impedanz allerdings nicht linear. Das „U“ in der Typenbezeichnung ist allerdings falsch, da UHF eigentlich erst bei 300 MHz beginnt.

Für höhere Frequenzen gibt es BNC-Steckverbinder (Bayonet Neill-Concelman) (**Bild 3**), die über zwei Bajonette verfügen und mit einer Vierteldrehung verriegelt werden. Ursprünglich wurden sie für das Militär entwickelt, doch heute sind BNC-Anschlüsse weit verbreitet und in vielen Anwendungen mit einer Bandbreite von bis zu 2 GHz zu finden. Eine Anwendung, die wir alle kennen, ist ihr Einsatz

an Laborgeräten wie Signalgeneratoren und Oszilloskopen.

Es gibt auch eine Gewindeversion der BNC-Verbinder, die TNC (Threaded Neill-Concelman) genannt wird. Durch das Gewinde werden die Verriegelungsschlitze vermieden, die den Frequenzbereich der BNC-Verbinder einengen. TNC kommt so auf eine obere Grenzfrequenz von etwa 11 GHz. TNC wird allgemein als HF-Antennenanschluss verwendet. BNC- und TNC-Steckverbinder gibt es mit Impedanzen von 50 Ω und 75 Ω . Es ist wichtig, den richtigen Stecker für Ihre Anwendung zu verwenden, da sie leicht unterschiedliche Anschlussmaße aufweisen. Normalerweise kann man sie auseinanderhalten, da 75- Ω -Steckverbinder weniger Dielektrikum besitzen als die 50- Ω -Modelle, aber verlassen sollte man sich nicht darauf.

Eine weitere häufige Variante von Gewinde-BNC ist der N-Verbinder (**Bild 4**), der 1940 von Paul Neill bei den Bell Labs entwickelt wurde. Ein N-Stecker ist wesentlich größer als BNC, aber auch für viel höhere Leistungen geeignet. Auch diese Verbinder gibt es in 50- Ω - und 75- Ω -Versionen, aber sie sind nicht kompatibel. Ein 50- Ω -Stecker in einer 75- Ω -Buchse würde die Buchse beschädigen.

Dann gibt es noch den bekannten SMA-Verbinder (**Bild 5**), die in den 1960er Jahren erfunden wurde. Er ist viel kleiner als ein BNC und besitzt daher einen weiteren Frequenzbereich bis etwa 18 GHz. Er ist nicht für signifikante Leistungen oder sehr viele Steckzyklen ausgelegt, aber für feste Installationen mehr als ausreichend.

Dies ist nur eine kleine Auswahl aus der Unzahl verschiedener Koaxial-Steckverbinder, die man heutzutage kaufen kann. Die klassischen, hier beschriebenen Koaxialverbinder sind seit vielen Jahren und Jahrzehnten auf dem Markt, was für ihr gelungenes Design und ihre Zuverlässigkeit spricht. ◀

(150341)



Steuern Sie weitere Bemerkenswerte Bauteile hinzu: Mailen Sie an neil@gruending.net.



1



2



3



4



5

Funktionen für Fortgeschrittene

Von 8 auf 32 bit: ARM-Controller für Einsteiger (7)

Aktivieren des Brown-out-Detektors

Wie Sie es von den 8-bit-Mikrocontrollern her kennen, sorgt auch bei ARM-Controllern der sogenannte Brown Out Detector (kurz BOD) dafür, dass bei einer Unterschreitung einer vom Entwickler festgelegten Spannungsgrenze entweder ein Interrupt oder ein Reset der MCU ausgelöst wird. Da diese Spannungsschwelle meist unter der empfohlenen Versorgungsspannung des Mikrocontrollers liegt, stellt der BOD eine Schutzfunktion dar, die bei einem Einbruch der Versorgungsspannung zum Beispiel den Mikrocontroller früh genug herunterfährt, bevor es deswegen zu unerwünschten Effekten wie Datenverlust kommt.

Beim SAM D20 als typischen ARM-Cortex-M0+-Vertreter ist es nicht anders. Er besitzt gleich zwei BODs, da der Mikrocontroller intern ja mit verschiedenen Versorgungsspannungen arbeitet. Es gibt einen BOD12, der schon ab Werk konfiguriert ist. Er arbeitet immer, wenn die MCU in Betrieb ist und überwacht die 1,2-V-Versorgungsspannung der CPU, die vom internen Spannungsregler bereitgestellt wird. Für uns interessanter ist der BOD33 für die 3,3-V-Versorgungsspannung an VDDANA. Dieser BOD kann entweder ständig im *Continuous Mode* die Spannung überwachen oder aber im *Sample Mode* bei jeder festgelegten Takt-Flanke eine Einzelmessung durchführen und danach in den Schlaf fallen. Der Takt wird vom internen Prescaler von der internen 1-kHz-Quelle je nach Einstellung durch bis zu 65536 geteilt. Auch eine Hysterese, die meist zwischen 35 mV und 170 mV liegt, ist einstellbar. Nun aber zum dazugehörigen Programm „Test of the BOD-module“, das den BOD33 richtig konfiguriert und einstellt. In **Listing 1** sehen Sie, dass nach dem üblichen Muster erst eine Konfigurations-Struktur (`config_bod`) erzeugt und mit Voreinstellungen gefüllt wird. Ausgewählte Einstellungen werden in den Struktur-Variablen geändert, die Struktur an den BOD33 übergeben sowie der Brown Out Detector letztendlich aktiviert. Die Funktion ist hier so eingestellt: Continuous Mode, die Reaktion beim Erreichen der Schwellspannung ist Reset, die Schwellspannung ist 0x27 und die Hysterese mit den beiden Schwellspannungen V_{BOD+} und V_{BOD-} ist eingeschaltet (**Tabelle 1**). Wenn Sie die Hysterese-Funktion nicht nutzen, gelten die Werte V_{BOD} im zweiten Teil der Tabelle (**Bild 1**). In der Spalte BOD33-Level sehen Sie mögliche Konfigurati-

Listing 1.

Die einfache Konfigurations-Funktion des BOD33

```
void configure_bod(void)
{
    struct bod_config config_bod;
    bod_get_config_defaults(&config_bod);

    config_bod.mode           = BOD_MODE_CONTINUOUS;
    //config_bod.mode         = BOD_ACTION_NONE;
    config_bod.action         = BOD_ACTION_RESET;
    config_bod.level          = 0x27;
    config_bod.hysteresis     = true;

    bod_set_config(BOD_BOD33, &config_bod);
    bod_enable(BOD_BOD33);
}
```

onous Mode die Spannung überwachen oder aber im *Sample Mode* bei jeder festgelegten Takt-Flanke eine Einzelmessung durchführen und danach in den Schlaf fallen. Der Takt wird vom internen Prescaler von der internen 1-kHz-Quelle je nach Einstellung durch bis zu 65536 geteilt. Auch eine Hysterese, die meist zwischen 35 mV und 170 mV liegt, ist einstellbar. Nun aber zum dazugehörigen Programm „Test of the BOD-module“, das den BOD33 richtig konfiguriert und einstellt. In **Listing 1** sehen Sie, dass nach dem üblichen Muster erst eine Konfigurations-Struktur (`config_bod`) erzeugt und mit Voreinstellungen gefüllt wird. Ausgewählte Einstellungen werden in den Struktur-Variablen geändert, die Struktur an den BOD33 übergeben sowie der Brown Out Detector letztendlich aktiviert. Die Funktion ist hier so eingestellt: Continuous Mode, die Reaktion beim Erreichen der Schwellspannung ist Reset, die Schwellspannung ist 0x27 und die Hysterese mit den beiden Schwellspannungen V_{BOD+} und V_{BOD-} ist eingeschaltet (**Tabelle 1**). Wenn Sie die Hysterese-Funktion nicht nutzen, gelten die Werte V_{BOD} im zweiten Teil der Tabelle (**Bild 1**). In der Spalte BOD33-Level sehen Sie mögliche Konfigurati-

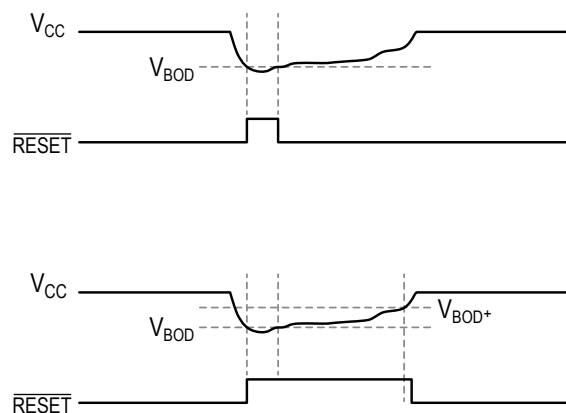


Bild 1. So funktioniert der BOD – oben ohne und unten mit Hysterese (Blockschaltbilder und Screenshots: Atmel).

Von Viacheslav Gromov (D)

In den bereits erschienenen Folgen haben wir alle wichtigen Peripherieelemente und die dazugehörigen ASF-Bibliotheken behandelt. In dieser Folge geht es um Funktionen für Fortgeschrittene. Die Projekte sind mehr oder weniger voneinander unabhängig, sodass Sie sich aussuchen können, was Sie lesen und ausprobieren möchten und was nicht. Außerdem werden in dieser vorletzten Folge Tipps und Tricks verraten. Ein Extra-Projekt zum Event-System finden Sie auf www.elektormagazine.de/articles.

onswerte für die Spannungsschwellen. Wir nutzen hier 0x27 (dezimal 39), was wiederum einer V_{BOD+} von typisch 2,84 V und einer V_{BOD-} von typisch 2,77 V entspricht. Durch das Verändern dieses Wertes im Programm können Sie also die Schwellenspannungen ändern.

In der Main-Funktion wird lediglich nach der system_init-Funktion die gerade besprochene BOD-Konfigurations-Funktion aufgerufen. Die Endlosschleife ist leer. Hier können Sie nun Ihren Code einfügen, der BOD33 ist nun aktiv und führt einen Reset durch, wenn die Versorgungsspannung die eingestellte Schwelle unterschreitet.

Sie können die BOD-Konfigurationen je nach Projekt ändern. Wenn Sie zum Beispiel statt

```
config_bod.action = BOD_ACTION_RESET;
```

den (im Code auskommentierten) Befehl

```
config_bod.action = BOD_ACTION_NONE;
```

in der Konfigurations-Funktion verwenden, können Sie in Ihrem Programm mit dem Befehl `bod_is_detected(BOD_BOD33)`, der true ausgibt, wenn der BOD33 bis zu diesem Zeitpunkt ein Unterschreiten der Spannungsschwelle registriert hat, „manuell“ den BOD pollen und die gewünschten Maßnahmen bei einer Spannungs-Unterschreitung durchführen. Diesen Detect-Flag müssen Sie dann bis zum nächsten Aufrufen des Befehls mit

Selected Modules

- ▶ Generic board support (driver)
- ▶ BOD - Brown Out Detector (driver)
- ▶ PORT - GPIO Pin Control (driver)
- ▶ SYSTEM - Core System Driver (driver)

Bild 2. Nur die BOD-Bibliothek wurde im ASF-Wizard eingebunden.

`bod_clear_detected(BOD_BOD33)` zurücksetzen.

Nun können Sie das Projekt auf das Board übertragen. Allerdings werden Sie davon gar nichts zu sehen bekommen, denn erst, wenn die Spannungsschwelle unterschritten wird und Sie die Endlosschleife mit Code gefüllt haben, können Sie den BOD in Aktion sehen.

Im ASF Wizard wurde für dieses Programm nur die BOD-Bibliothek eingebunden (**Bild 2**).

Nähere Informationen zum BOD33 finden Sie im Datenblatt des SAM D20 ab Seite 143 und unter <http://goo.gl/yVtRvK>.

Tabelle 1. Die Spannungsschwellen des BOD33 mit und ohne Hysteresis (Quelle: Atmel).

Symbol	BOD33-Level	Bedingung	Min.	Typ.	Max.
V_{BOD+}	6	Hysteresis ON	--	1,715 V	1,745 V
	7		--	1,75 V	1,779 V
	39		--	2,84 V	2,92 V
	48		--	3,2 V	3,3 V
V_{BOD-} oder V_{BOD}	6	Hysteresis ON oder	1,62 V	1,64 V	1,67 V
	7		1,64 V	1,675 V	1,71 V
	39	Hysteresis OFF	2,72 V	2,77 V	2,81 V
	48		3,0 V	3,07 V	3,2 V

Schlaf- und Reset-Funktionen des SAM D20

In der in jedem neuen Projekt vorhandenen System-Driver-Bibliothek befinden sich auch Befehle für die Schlaf- und Reset-Funktionen des SAM D20. Aus dieser Bibliothek kennen Sie zum Beispiel schon den Befehl `system_init()` zur Initialisierung der ganzen MCU. Weitere wichtige Befehle sind meist zuständig für die Schlaf- und Reset-Funktionen des SAM D20:

Die Funktion `system_reset()` setzt beim Aufruf die ganze MCU zurück. Nur die Peripherieelemente, die nicht zurückgesetzt werden können/sollen (wie die RTC) sind davon nicht betroffen.

Die Funktion `system_get_reset_cause()` gibt beim Aufruf eine symbolische Konstante (einen Wert) zurück, der angibt, durch was der letzte Reset ausgelöst wurde (zum Beispiel Software-Reset oder BOD-Reset).

`System_set_sleepmode()` legt den Schlafmodus für den nächsten Aufruf des Befehls `system_sleep()` fest, was je nach Modus mehr oder weniger Strom spart. Der einzige Parameter für diesen Befehl ist die symbolische Konstante für den gewünschten Schlafmodus (beispielsweise `SYSTEM_SLEEPMODE_`

`IDLE0`). Welche Peripherieelemente in den einzelnen Schlafmodi weiterhin arbeiten und welche nicht, ist in **Tabelle 1** zusammengefasst.

Die Funktion `system_voltage_reference_enable()` benötigt als Parameter nur die symbolische Konstante für die Referenz-Spannungsquelle (zum Beispiel `SYSTEM_VOLTAGE_REFERENCE_TEMPSENSE` für die Spannungs-Referenz des internen Temperatur-Sensors), die mit diesem Befehl aktiviert wird. Zum Deaktivieren einer Spannungs-Referenz gibt es den Befehl `system_voltage_reference_disable()`, mit dem gleichen Parameter.

Der Befehl `system_get_device_id()` gibt einen 32-bit-Integer-Wert zurück. Dieser Wert stellt die einzigartige ID der MCU dar, die MCUs gleichen Typs unterscheidet. Die Device-ID ist bei den meisten MCUs der ARM-Cortex-M0+-Familie vorhanden.

Nähere Informationen dazu finden Sie unter www.atmel.com/Images/Atmel-42120-SAM-D20-System-Driver-SYSTEM_Application-Note_AT03260.pdf.

Tabelle 1. Übersicht der in den verschiedenen Sleep-Modi (de)aktivierten Peripherie- und System-Elemente (Quelle: Atmel).

Sleepmode	CPU clock	AHB clock	APB clocks	Clock sources	System clock	32KHz	Regulator mode	RAM mode
IDLE 0	Stop	Run	Run	Run	Run	Run	Normal	Normal
IDLE 1	Stop	Stop	Run	Run	Run	Run	Normal	Normal
IDLE 2	Stop	Stop	Stop	Run	Run	Run	Normal	Normal
STANDBY	Stop	Stop	Stop	Stop	Stop	Stop	Low Power	Source / Drain biasing

Tipps und Tricks rund um den SAM D20

Stromverbrauch der Peripherie

Falls Sie längere Zeit nicht mehr in das Datenblatt geschaut haben, sollten Sie es jetzt tun! In Januar 2015 wurde nämlich das „große“ Datenblatt erneuert, wodurch manche Informationen besser verständlich oder gar neue Informationen (etwa in Form von Tabellen) vorhanden sind. Allerdings sind jetzt die Seiten etwas anders nummeriert.

Unabhängig davon ist es (da Sie nun alle Peripherieelemente kennen) vorteilhaft, das Kapitel 32 auf Seite 571 (früher Seite 570) mit den elektrischen Charakteristika der MCU und einzelner Elemente kurz durchzusehen. Dort finden Sie unter anderem die zulässigen Versorgungsspannungen und Ströme. Unbedingt anschauen sollten Sie sich Tabelle 32-8, mit dem Stromverbrauch der MCU in verschiedenen Schlafmodi, und die Tabelle 32-10 mit dem Strombedarf der einzelnen Peripherieelemente. In Low-Power-Projekten sind diese Informationen eine große Hilfe und Sie können genau bestimmen, welche Peripherieelementen Sie wirklich brauchen und wann Sie den Mikrocontroller schlafen legen können.

FreeRTOS

Auch auf dem vergleichsweise kleinen ARM Cortex-M0+-Controller SAM D20 kann ein Real Time Operating System (kurz

RTOS), also ein Betriebssystem laufen, natürlich ohne jegliche graphische Oberfläche. Weit verbreitet in der M-Familie ist FreeRTOS, das auch im ASF eingebunden ist. Es gibt vier Beispielprogramme dafür, die Sie sich bei Interesse auch anschauen sollten. FreeRTOS ist ein unabhängiges und plattformübergreifendes System, das vom Hersteller lediglich an den jeweiligen Mikrocontroller und seine Peripherie angepasst werden muss. Der Vorteil dieser Betriebssysteme ist der Aufbau und der hohe Abstraktionslevel. Sie können damit schnell und effizient große Projekte umsetzen, ohne sich mit jedem Bit und Byte in den Registern auseinanderzusetzen zu müssen. Der Einstieg fällt wegen der (Task-)Struktur aber erst einmal nicht so leicht. Mehr Informationen zu FreeRTOS und einen Quick-Start-Guide finden Sie unter www.freertos.org.

Weitere interessante Programme

Manche Bibliotheken wurden in diesem Kurs absichtlich nicht behandelt, da sie auf Erweiterungs-Boards basieren. Trotzdem lohnt sich ein Blick auf ihren Aufbau.

Schauen Sie sich einmal die Beispielprogramme für SD-Karten mit verschiedenen Datei-Systemen, für die Flash-Bausteine von Atmel und für die Kalibrierung der 8-MHz-Taktquelle an. Sie müssen nicht alles verstehen, nur die Grundzüge und

eingebundenen Bibliotheken nachvollziehen können. Schauen Sie auch in die dazugehörige API-Erklärung.

Hilfe und Tipps bei der Programmierung

Im Forum wurden verschiedene Fehler bei der Kommunikation zwischen dem SAM D20-Board und Atmel Studio gemeldet. Wenn Sie auch irgendwelche Probleme beim Erkennen des Boards oder dem Updaten der Firmware haben, lohnt sich ein Blick auf die Support-Seite von Atmel (**Bild 1**) unter www.atmel.com/design-support. Dort können Sie nach Ihrem Problem und einer Lösung suchen; falls Sie nichts finden, können Sie die Atmel-Experten zu Rate ziehen.

Wenn wir gerade beim Atmel Studio 6 sind: Der darin integrierte Compiler für die SAM-Familie ist ausgesprochen gutmütig und intelligent. Wenn Sie zum Beispiel in einem Application Manual lesen, dass der Parameter eines bestimmten Befehls ein Pointer auf die Struktur sein muss und Sie lediglich die Struktur selbst als Parameter eintippen, erkennt dies das Studio und akzeptiert die Eingabe, weist Sie aber auch unter „Warnings“ nach dem Kompilieren darauf hin. Deswegen sind manchmal sogar in den Beispielprogrammen von Atmel Strukturen statt Pointer auf Strukturen zu finden. Das gleiche gilt zum Beispiel auch für Funktionsprototypen, diese werden

meist gar nicht benötigt. Diese Gutmütigkeit des Compilers im Studio macht dem Entwickler das Leben leichter. Noch ein Tipp: Falls beim Kompilieren plötzlich und unerwartet ein Fehler des Make-Files angezeigt wird, kompilieren Sie das Programm einfach erneut - das klappt meist.

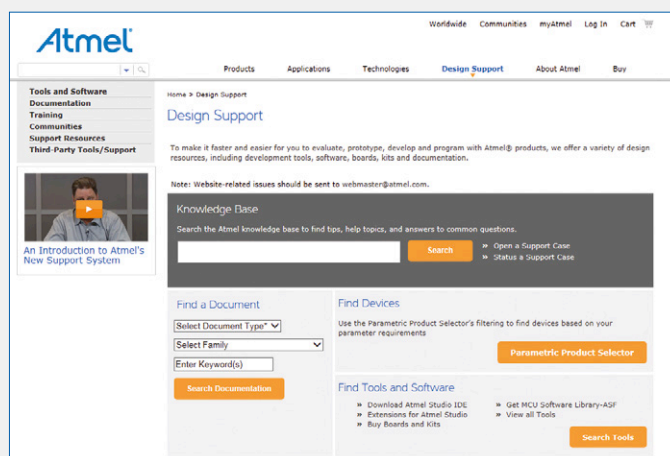


Bild 1. So sieht der Design Support auf der Atmel-Seite aus.

Der Peripheral Access Controller

Dieser Cortex-M-typische Controller kann den Schreibzugriff auf bestimmte Register ausgewählter Peripherieelemente blockieren. Dies ist von Peripherieelement zu Peripherieelement sehr unterschiedlich, meist jedoch betrifft es Control-Register. Durch diese Blockierung des Peripheral Access Controllers (PAC) kann die CPU keine Werte in diesen Registern ändern. So werden ungewollte Zugriffe und dadurch Fehler vermieden. Die ASF-Bibliothek dazu ist sehr einfach. Das Projekt „PAC Test1“ demonstriert den typischen Ablauf einer Blockierung und deren Aufhebung. Wie im **Listing 1** zu sehen, besteht das Programm lediglich aus der Main-Funktion. Darin werden nach den Initialisierungs-Befehlen die Register des Ports durch den PAC mit dem Befehl

```
system_peripheral_lock(SYSTEM_PERIPHERAL_ID(PORT),
    ~SYSTEM_PERIPHERAL_ID(PORT));
```

blockiert. Dieser enthält nur die ID des jeweiligen Peripherieelementes und die mit ~ invertierte ID. Diese invertierte ID stellt einfach eine Schutzfunktion dar, damit es nicht so einfach ist, mit dem Befehl etwas versehentlich zu blockieren. Die Endlosschleife lässt die LED0 auf dem Board blinken. Vor dem Ändern des Bits im Port-Register (durch den uns schon bekannten Toggle-Befehl) wird die Blockierung aufgehoben und danach wieder aktiviert. Damit während dieser Zeit kein Interrupt auftritt, der unter Umständen den Port manipuliert, werden alle Interrupts vor dem Aufheben der Blockierung mit `system_interrupt_enter_critical_section();` deaktiviert und mit `system_interrupt_leave_critical_section();` nach dem Blockieren wieder aktiviert.

Der Befehl zum Aufheben der Blockierung `system_peripheral_`

Listing 1. Ein Teil der Main-Datei mit dem Code zum Toggeln der LED.

```
system_peripheral_lock(SYSTEM_PERIPHERAL_ID(PORT), ~SYSTEM_PERIPHERAL_ID(PORT));
while(true){
    system_interrupt_enter_critical_section();
    system_peripheral_unlock(SYSTEM_PERIPHERAL_ID(PORT), ~SYSTEM_PERIPHERAL_ID(PORT));
    port_pin_toggle_output_level(LED_0_PIN);
    system_peripheral_lock(SYSTEM_PERIPHERAL_ID(PORT), ~SYSTEM_PERIPHERAL_ID(PORT));
    system_interrupt_leave_critical_section();

    delay_ms(500);
}
```

unlock() benötigt genau die gleichen Parameter wie der Blockier-Befehl.

Delay_ms(500) sorgt für die Blink-Periode von 1 s. Um dieses Programm auszuprobieren, müssen Sie es lediglich auf das Board übertragen. Alle eingebundenen Bibliotheken im ASF Wizard sind in **Bild 1** zu sehen.

Nähere Informationen zum PAC finden Sie im Datenblatt ab Seite 34 und unter <http://goo.gl/kyuzWc>.

Selected Modules

- ▷ Generic board support (driver)
- ▷ Delay routines (service) systick
- ▷ PAC - Peripheral Access Controller (driver)
- ▷ PORT - GPIO Pin Control (driver)
- ▷ SYSTEM - Core System Driver (driver)

Bild 1. Diese Bibliotheken wurden in unser Projekt eingebunden. Die Delay- und die PAC-Bibliothek kamen zu den Standard-Bibliotheken hinzu.

Non-Volatile Memory (NVM)

NVM heißt nichts anderes als nichtflüchtiger Speicher. In diesem Speicherbereich, der wie ein EEPROM strukturiert ist, kann die MCU Daten ablegen, die nach dem Ausschalten erhalten bleiben sollen. Wie in **Bild 1** zu sehen, ist ein Teil des NVMs für den Bootloader und für die internen Register (zum Beispiel für die Kalibrierung der Oszillatoren) reserviert. Den mittleren Bereich kann man aber in seinem Programm nutzen.

Mehr zum Aufbau und dem Ausrechnen der Adressen finden Sie unter www.atmel.com/Images/Atmel-42114-SAM-D20-Non-Volatile-Memory-Driver-NVM_Application_Note_AT03247.pdf.

An dieser Stelle soll kurz das Projekt „First test with NVM“ vorgestellt werden, das die drei wichtigsten Befehle behandelt: Löschen, Schreiben und Lesen. Eine Zeichenkette wird nach einem Löschen des NV-Speichers gespeichert und beim Betätigen von SW0 beliebig oft an der UART-Schnittstelle ausgegeben. Im Code sieht man nach dem Einbinden von asf.h, den Funktionsprototypen sowie den Deklarationen von Arrays die Konfigurations-Funktion für den U(S)ART. Es folgt die kleine Kon-

Listing 1. So sieht die kompakte NVM-Konfigurations-Funktion aus.

```
void configure_nvm(void)
{
    struct nvm_config config_nvm;
    nvm_get_config_defaults(&config_nvm);
    config_nvm.manual_page_write = false;
    nvm_set_config(&config_nvm);
}
```

figurations-Funktion für NVM, bei der die erzeugte Konfigurations-Struktur zunächst mit Voreinstellungen gefüllt und dann in die Struktur-Variable manual_page_write false geschrieben wird, damit der Speicher im automatischen Schreib-Modus arbeitet. **Listing 1** zeigt, wie die vorgenommenen Einstellungen an den NVM übergeben werden. In **Listing 2** ist der Kern des Programms zu sehen. Hier wird das Array page_buffer mit 64 Werten (die Speichergröße einer Page) von 0...63 in einer For-Schleife gefüllt. Nun wird zuerst zur Sicherheit diese Page mit dem Befehl

```
while(nvm_erase_row(100 * NVMCTRL_ROW_PAGES *
    NVMCTRL_PAGE_SIZE) != STATUS_OK){}
```

gelöscht. Die While-Schleife wartet, bis der Befehl vollständig ausgeführt ist. Als Parameter braucht man die errechnete Page-Adresse (siehe Application Note). Und schon kann das Array page_buffer mit der Funktion

```
while(nvm_write_buffer(100 * NVMCTRL_ROW_PAGES *
    NVMCTRL_PAGE_SIZE, page_buffer, NVMCTRL_PAGE_SIZE) != STATUS_OK){}
```

ins NVM geschrieben werden, die sich ebenfalls in einer While-Schleife befindet und die errechnete Page-Adresse, das Array und die Page-Größe als Parameter benötigt. In der Endlosschleife wird mit dem Lese-Befehl der NVM-Inhalt in das Array page_buffer_receive eingelesen. Dieser Befehl benötigt die gleichen Parameter wie der Schreib-Befehl und steckt ebenfalls in einer While-Schleife. Mit den nach-

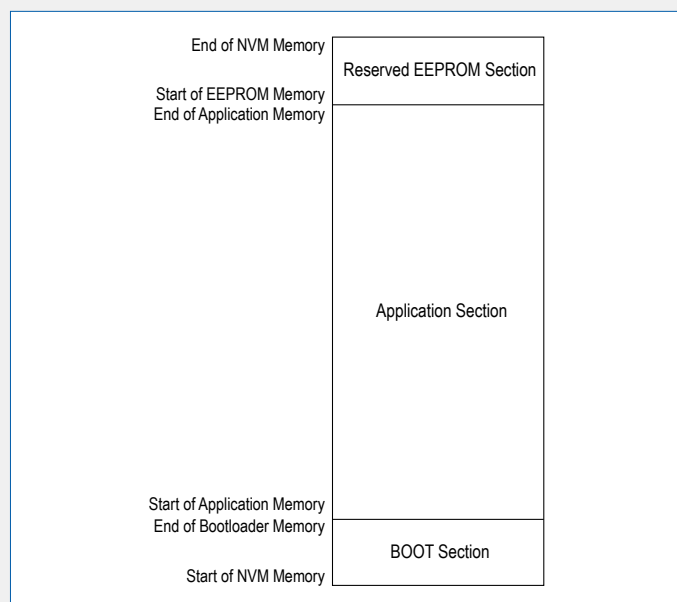


Bild 1. So sieht die Speicherverteilung des NVMs aus. Nur der mittlere Bereich ist für uns interessant.

Listing 2. Der wichtigste Ausschnitt aus der Main-Funktion.

```

for (uint32_t i = 0; i < NVCTRL_PAGE_SIZE; i++) {
    page_buffer[i] = i;
}
while(nvm_erase_row(100 * NVCTRL_ROW_PAGES * NVCTRL_PAGE_SIZE) != STATUS_OK){}
while(nvm_write_buffer(100 * NVCTRL_ROW_PAGES * NVCTRL_PAGE_SIZE, page_buffer, NVCTRL_PAGE_SIZE) != STATUS_OK){}
while (true) {
    while(port_pin_get_input_level(BUTTON_0_PIN)){}
    while(nvm_read_buffer(100 * NVCTRL_ROW_PAGES * NVCTRL_PAGE_SIZE, page_buffer_receive, NVCTRL_PAGE_SIZE) != STATUS_OK){}
    while(usart_write_buffer_wait(&usart_instance, &page_buffer_receive, 64) != STATUS_OK){}
    while(!port_pin_get_input_level(BUTTON_0_PIN)){}
}

```

folgenden Befehlen wird das Array an der U(S)ART-Schnittstelle ausgegeben:

```

while(nvm_read_buffer(100 * NVCTRL_ROW_PAGES * NVCTRL_PAGE_SIZE, page_buffer_receive, NVCTRL_PAGE_SIZE) != STATUS_OK){}
while(usart_write_buffer_wait(&usart_instance, &page_buffer_receive, 64) != STATUS_OK){}

```

Die beiden Befehle sind von zwei While-Schleifen umgeben, die den SW0-Pegel abfragen. So werden beide Befehle nur beim Betätigen von SW0 ausgeführt. Übertragen Sie das Programm auf das Board und öffnen Sie das Terminal-Programm mit den Einstellungen wie in **Bild 2**. Drücken Sie SW0, um die im NVM gespeicherten Werte auszulesen.

In dieses Projekt wurde die NVM- und die U(S)ART-Bibliothek in der Polled-Variante eingebunden.

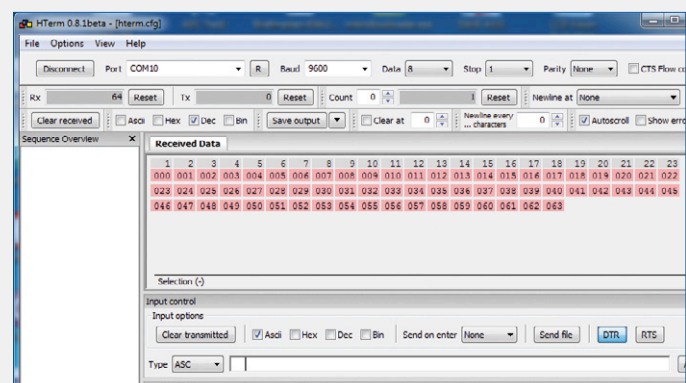


Bild 2. Hier sind die Terminal-Einstellungen und die Ausgabewerte nach dem Betätigen von SW0 zu erkennen.

Nähere Informationen finden Sie im Datenblatt ab Seite 25 und unter <http://goo.gl/afKJj>.

Interrupt- und Clock-Befehle

In den Application Notes unter www.atmel.com/Images/Atmel-42122-SAM-D20-System-Interrupt-Driver_Application-Note_AT03261.pdf und www.atmel.com/Images/Atmel-42119-SAM-D20-Clock-Management-Driver-CLOCK_Application-Note_AT03259.pdf finden Sie zahlreiche Befehle für Interrupt-, Clock- und viele andere System-Zwecke, die Sie sich unbedingt anschauen sollten. Hier eine stichpunktartige Auswahl:

System_interrupt_set_priority() braucht als Parameter nur den Interrupt-Vektor des benötigten Interrupts (als symbolische Konstante) und den Prioritätslevel von 0...3 (ebenfalls in Form von symbolischen Konstanten). Mit diesem Befehl kann man Prioritäten von Interrupts setzen beziehungsweise ändern. **System_interrupt_get_priority()** benötigt lediglich den Interrupt-Vektor und gibt dagegen den Prioritätslevel des dazugehörigen Interrupts aus.

System_interrupt_get_active() braucht keinerlei Parameter und gibt den Interrupt-Vektor des gerade stattfindenden

Interrupts aus. Dieser Befehl kann zum Beispiel in einer ISR aufgerufen werden.

Mit Befehlen wie **system_clock_source_osc32k_set_config()**, die eine Konfigurations-Struktur als Parameter benötigen, kann man die Taktquellen während des Programmablaufs nach dem typischen ASF-Muster mit Einstellungen in den Konfigurations-Strukturen aktivieren, deaktivieren und konfigurieren.

Mit **system_clock_source_get_hz()** kann man herausfinden, welche Taktquelle gerade welche Frequenz liefert. Als Parameter wird nur die Taktquelle (die symbolische Konstante dafür) benötigt. **System_apb_clock_source_get_hz()** ist ähnlich und gibt die aktuelle Frequenz des Taktsignals vom Peripherie-Bus (APB) aus.

Mit Befehlen wie **system_cpu_clock_set_divider()** stellt man für die entsprechenden Taktquellen die Teiler der Taktfrequenz ein. Der Teiler wird dem Befehl in Form einer symbolischen Konstante als Parameter mitgegeben. ◀

(150324)

EAGLE Tipps & Tricks (1)

EAGLE mit ULPs individualisieren



Von **Neil Gruending** (Kanada)

Die meisten unserer Leser dürften bereits mit den Grundlagen der Layout-Software EAGLE vertraut sein. Daher soll es in dieser Mini-Serie mehr um Spezialitäten gehen. In dieser Folge zeigen wir, wie sich User Language Programs erstellen lassen. Oft ist es einfacher, eine neue Programmiersprache über Beispiele zu lernen. In dieser Folge soll es um die Skript-Umgebung von EAGLE gehen und dabei kommt natürlich noch ein kleines Beispielprogramm vor.

Programmieren in EAGLE

Eine der Besonderheiten von EAGLE ist, dass man eigene Programme zur Steuerung dieser Software schreiben kann. Die Skriptsprache ist deutlich mächtiger als bei vielen anderen Layout-Paketen, die nur ein paar Regeln beherrschen.

Das Control Panel in **Bild 1** ist der Ort, wo man die Programme ablaufen lassen kann. EAGLE inspiziert standardmäßig das Verzeichnis „C:\EAGLE-7.3.0\ulp“ nach Dateien mit der Extension „.ulp“ und listet die Funde auf, damit man sie per Rechtsklick auf Schaltungen, Platinen oder Libraries loslassen kann.

EAGLE verfügt auch über Skripts, die sich etwas von ULPs unterscheiden. Skripte entsprechen eher eine Art Makro-Sprache, etwa zum Im- oder Export von Dateien sowie zu Konfiguration von EAGLE. Ein ULP ist ein Programm mit deutlich mehr Möglichkeiten. Weitere Informationen zu Skripts in EAGLE entnimmt man am besten dem EAGLE-Handbuch.

ULPs – eine Einführung

Bei ULPs handelt es sich um in ULP-Syntax gehaltene Textdateien mit der Extension „.ulp“. Im PDF des EAGLE-Handbuchs ist leider keine ULP-Dokumentation enthalten. Diese Datei lässt sich samt Online-Hilfe von der CadSoft-Webseite [1] herunterladen. Doch wer je in C programmiert hat, wird sich bei der

ULP-Syntax schnell zuhause fühlen, da die Ähnlichkeit groß ist. Man kann ein ULP per Rechtsklick im Control Panel öffnen und dort auch editieren, denn dann öffnet sich der EAGLE-Texteditor (siehe **Bild 2**). Der einfache Editor ist zwar nicht schlecht, doch ich ziehe farbiges Syntax-Highlighting vor. Glücklicherweise gibt es eine Syntax-Datei [3] für Notepad++ [4]. Man kann die ULP-Syntax-Datei in Notepad++ importieren, indem man im Sprachen-Menü den Eintrag „Define your language“ wählt. Dann öffnet sich das Fenster für die „User Defined Language“. Man klickt nun auf Import und wählt die ULP-Syntax-XML-Datei aus, die man zuvor von der Webseite heruntergeladen hat. Wenn man nun Notepad++ neu startet, sieht man im Sprachen-Menü den Eintrag „EAGLE ULP“. Wenn man jetzt eine ULP-Datei öffnet, erscheint diese mit Syntax-Highlighting wie in **Bild 3**, vorausgesetzt man stellt den Font passend ein und deaktiviert die automatische Rechtschreibkorrektur.

ULP-Datei-Format

Nachdem nun der Texteditor richtig konfiguriert ist, kann man das ULP-Beispiel von **Bild 4** genauer unter die Lupe nehmen. Die ULP-Datei lässt sich bequem von [2] downloaden. Wenn man sie bei einer Schaltung oder einer Platine ablaufen lässt, dann wird eine Datei namens „example1.txt“ im EAGLE-Installationsverzeichnis (bei mir „C:\EAGLE-7.3.0“) mit dem Inhalt „1+2=3“ erstellt.

Die erste Zeile des Beispiels beginnt mit „#require“. Hier wird EAGLE mitgeteilt, dass dieses Programm die Version 7.0 von EAGLE oder neuer benötigt. Diese Angabe ist zwar nicht zwingend erforderlich, aber ich verwende sie üblicherweise, falls Funktionen enthalten sind, die nur bei bestimmten EAGLE-Versionen verfügbar sind.

In der nächsten Zeile befindet sich die Deklaration „#usage“. Hier wird festgelegt, was beim Klick auf das Programm im Control Panel angezeigt wird (siehe **Bild 5**). Die weiteren Strings in Anführungszeichen enthalten den Text für „#usage“. Der String „\n“ am Ende der ersten Zeile ist die Anweisung, die Anzeige in einer neuen Zeile fortzusetzen. Man kann „#usage“ einzeilig oder nach Bedarf auch mehrzeilig ausführen.

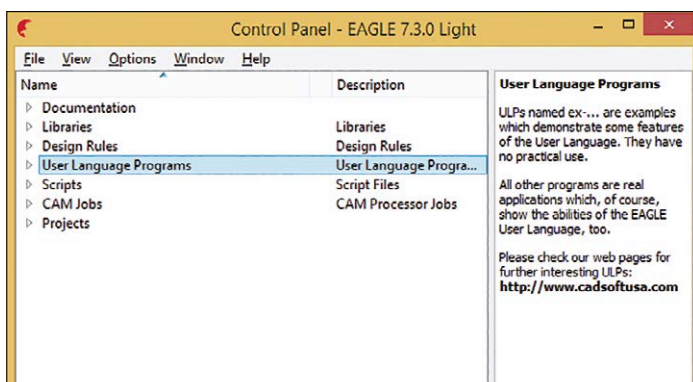


Bild 1. Control Panel.

Sowohl „#require“ als auch „#usage“ sind Befehle, die EAGLE mitteilen, wie der Rest des ULPs zu interpretieren ist. Solche Anweisungen sind leicht am führenden „#“ zu erkennen, ganz wie die Direktiven für einen C-Präprozessor.

Die nächsten Zeilen sind nützliche Kommentare, die von EAGLE bei der Programmausführung ignoriert werden. Eine einzelne Kommentarzeile wird von zwei Slashes „//“ eingeleitet. Mehrzeilige Kommentare rahmt man mit einem beginnenden „/*“ und einem abschließendem „*/“ ein. Man kann mehrzeilige Kommentare aber nicht verschachteln. Das erste Auftauchen von „*/“ beendet den kompletten Kommentar.

Der erste Teil einer ULP-Datei besteht aus Deklarationen von benötigten Variablen oder Funktionen. Im Beispiel werden zwei Strings mit den Namen „one“ und „two“ definiert. Eine Variablen-Deklaration beginnt mit einer der folgenden Variablentypen:

- **int**: eine Integer-Zahl (Bereich -2.147.483.647 bis +2.147.483.647)
- **char**: ein ASCII-Zeichen oder eine Integer-Zahl im Bereich 0 bis 255
- **real**: eine Fließkomma-Zahl
- **string**: eine Zeichenkette, in der Text abgelegt werden kann

Auf den Variablentyp folgt der Variablenname. Wie bei anderen Programmiersprachen auch muss er mit einem Buchstaben oder einem Unterstrich anfangen. Ziffern sind erst nach dem ersten Zeichen erlaubt. Namen wie „variable1“ sind gültig. Wichtig ist: Bei ULP-Variablen ist Groß- und Kleinschreibung relevant. „ab“ und „Ab“ sind also zwei verschiedene Variablen. Im Beispiel wird eine Variable pro Zeile deklariert, aber man kann auch mehrere auf einmal deklarieren, indem man sie wie in „int a, b, c;“ durch Kommas trennt. Im Beispiel ist auch gezeigt, wie man sie durch Initialisierung nach dem Gleichheitszeichen „=“ mit einem Startwert besetzt.

Weiter wird die Funktion „add“ demonstriert. Zwei Integer-Werte werden addiert und die Summe als Integer ausgegeben.

Im letzten Teil der ULP-Datei wird die Programmausführung gestartet. Im Beispiel werden hier die beiden Zahlen „1“ und „2“ addiert und das Resultat in der neuen Variable „c“ abgelegt. Anschließend folgt eine zusammengesetzte Funktion auf Basis der Funktion „output“. Solche Befehle sind mit geschweiften Klammern versehen, mit denen der entsprechende Bereich festgelegt wird. Hier wird per „output“ mit der Funktion „printf“ in eine Datei mit der Bezeichnung „example1.txt“ geschrieben.

Fazit

In dieser Folge ging es um ULPs und ihren grundlegenden Aufbau. In der nächsten Folge werden komplexere Beispiele beleuchtet, die auch Daten von der Schaltung oder der Platine einlesen können. ◀

(150331)

Weblinks

- [1] www.cadsoft.de/downloads/documentation
- [2] www.elektormagazine.de/150331
- [3] www.cadsoft.de/downloads/file/notepad_userdefinedlang.zip
- [4] <http://notepad-plus-plus.org>

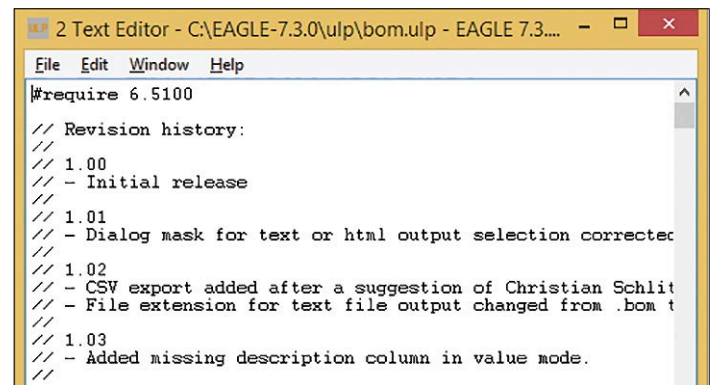


Bild 2. Texteditor von EAGLE.

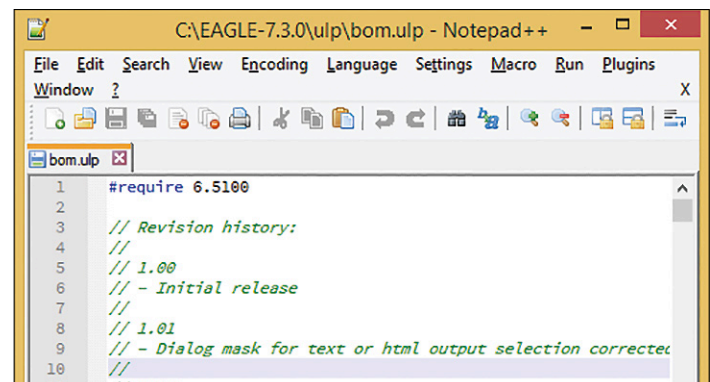


Bild 3. Texteditor Notepad++.

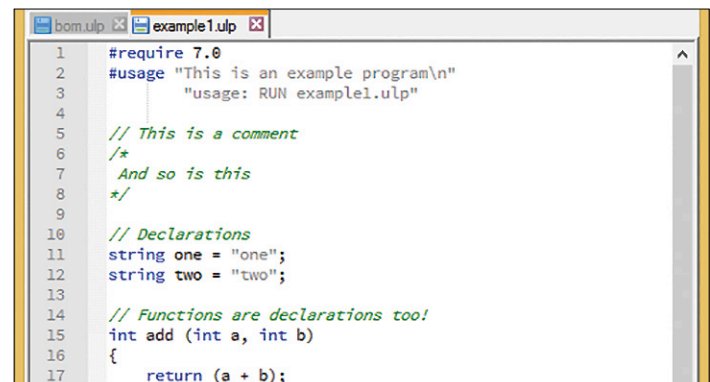


Bild 4. Beispiel-ULP.

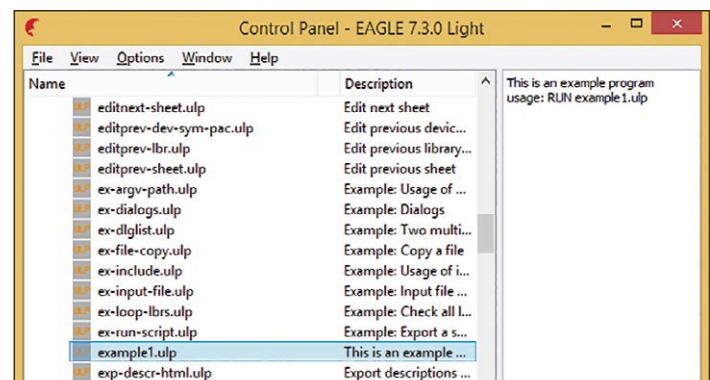
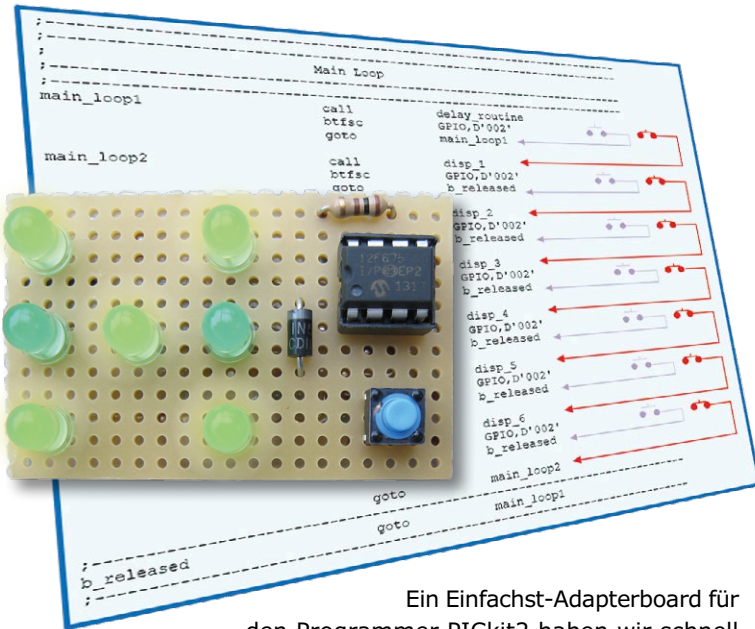


Bild 5. Beispiel-ULP im Control Panel.

Assembler Crashkurs (2)

Mini-Dev-Board und elektronischer Würfel

Von **Miroslav Cina** miroslav.cina@t-online.de



Ein Einfachst-Adapterboard für den Programmer PICKit2 haben wir schnell aufgebaut. So ein Board macht die Programmierung passender Mikrocontroller sehr einfach: Den Chip einstecken, Board an den Programmer anstecken, die Programmer-Software starten, damit die fertig assemblierte Firmware übertragen und fertig. Man kann mit wenig Aufwand Adapterplatinen für Controller unterschiedlichster Pin-Anzahl bauen, denn jeder Mikrocontroller von Microchip besitzt die nötigen Programmier-Pins: Vpp, ICSPDAT und ICSPCLK. Bei Adapterplatinen geht es also nur darum, diese Pins (und zusätzlich noch Vcc und GND) mit den gleichnamigen Pins des PICKit-Programmers zu verbinden. Für den hier verwendeten 8-Pin-Controller PIC12F675 baut man also eine achtpolige Variante wie in **Bild 1**. Man sieht: Einfacher geht es kaum. Man braucht dafür ein Stückchen Lochrasterplatine, eine achtpolige IC-Fassung und eine sechspolige Stiftleiste. Selbst auf die LED nebst Vorwiderstand kann verzichtet werden, doch ist die optische Signalisierung der Versorgung des Controllers mit Strom ein schönes Feature.

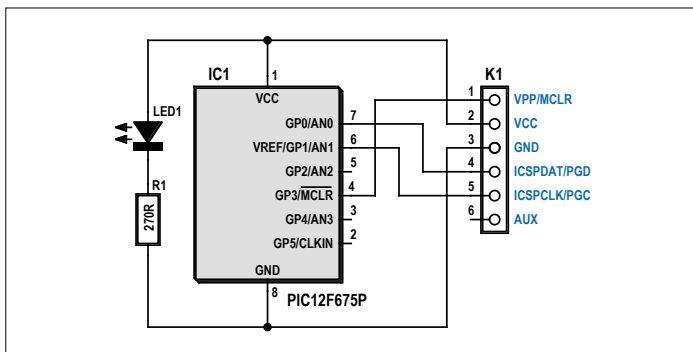


Bild 1. Schaltung der Adapterplatine als Minimalversion eines Entwicklungsboards.

Im ersten Teil haben wir die Basics der Assembler-Programmierung und die Hardware-Grundlagen des verwendeten PIC-Controllers kennengelernt. Weiter geht es mit einem selbstgestrickten Mini-Entwicklungs-Board und zusätzlichen Assembler-Befehlen. Als Demo-Anwendung werden wir einen elektronischen Würfel programmieren.

Bild 2 zeigt, wie die fertige Adapterplatine an den Programmer gesteckt wird.

Register-Details

Bevor wir zu weiteren Assemblerbefehlen kommen, die wir für unsere Demo-Anwendung benötigen, werfen wir noch einmal einen kurzen Blick auf die wichtigsten Register.

Config-Register

Fast alle Register des Mikrocontrollers sind im SRAM abgelegt, das Konfigurationsregister ist aber eine Ausnahme. Das Register hat insgesamt 14 Bits, wovon an dieser Stelle zunächst nur vier Bits relevant sind.

Die drei LSBs (**L**east **S**ignificant **B**its = niedrigstwertige Bits) „FOSC2:FOSC0“ sind für die Konfiguration des Oszillators vorgesehen. Im Kurs werden die folgenden zwei Einstellungen dieser drei Bits verwendet.

010: Quarzgesteuerter HF-Oszillator. In diesem Modus wird der extern gesteuerte Oszillator verwendet. Die Frequenz wird mit einem externen Quarz zwischen den Pins 2 (GP5) und 3 (GP4) festgelegt. Man kann so das mögliche Tempo mit einem Takt bis maximal 20 MHz ausreizen. Da zwei

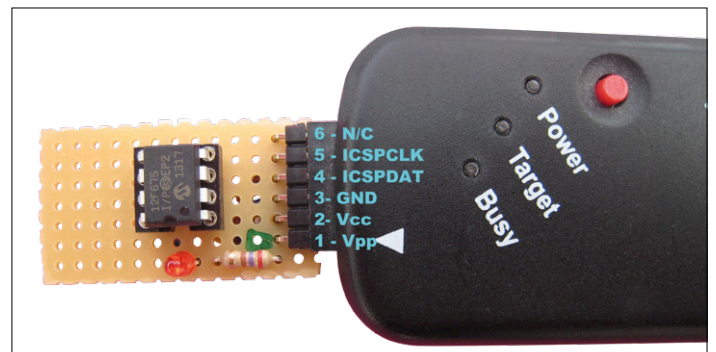


Bild 2. Fertig aufgebaute Adapterplatine, angeschlossen an den PICKit2-Programmer.

IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

Bild 3. Die Bedeutung der Bits des STATUS-Registers.

Pins durch den Quarz belegt sind, bleiben aber nur noch vier von sechs möglichen als I/Os verfügbar.

100: Interner Oszillator. In diesem Modus ist der interne Oszillator mit einem Takt von 4 MHz aktiviert. Der Takt ist nicht so genau wie bei einer Quarzsteuerung, aber mit einer Toleranz von $\pm 1\%$ für viele Anwendungen noch geeignet. Der große Vorteil ist: Man benötigt dafür keine extra Bauelemente und GP4 plus GP5 stehen als allgemeine I/Os zur Verfügung.

Bit 5 (MCLRE) des Konfigurationsregisters steuert die Funktion von Pin 4. Falls MCLRE = 1 ist, fungiert Pin 4 als „Master Clear“ – also als Reset-Pin. Bei MCLRE = 0 kann Pin 4 unter Verzicht auf die Reset-Funktion als Eingang genutzt werden.

Status-Register

Im STATUS-Register finden sich Informationen über durchgeführte arithmetische Operationen. In **Bild 3** sieht man die Bedeutung der einzelnen Bits des Registers.

Die beiden MSBs (**M**ost **S**ignificant **B**its = höchstwertige Bits) IRP und RP1 sind reserviert und sollen beim Schreiben immer „0“ sein. Sie werden beim PIC12F675 schlicht nicht genutzt. Bit 5 ist RP0, das „Register Bank Select Bit“. Es dient (wie schon im ersten Teil beschrieben) zur Auswahl der Speicherbank. Bei einem Wert von „0“ wird bei allen nachfolgenden Befehlen auf die Speicherplätze 00h...7Fh zugegriffen – alternativ bei einem Wert von „1“ auf 80h...FFh.

Die beiden Bits TO und PD verraten, wie der letzte Reset zustande kam (Näheres siehe Datenblatt [1]).

Bit 2 Z (= Zero) ist aktiviert, wenn das Ergebnis der letzten Operation Null ist – es muss dabei nicht unbedingt eine arithmetische Operation gewesen sein.

Die beiden LSBs zeigen beim Addieren und Subtrahieren aufgetretene Überläufe (Carry-Zustände) an.

Option-Register

Über das Option-Register lassen sich weitere Einstellungen tätigen. In unserer Demo-Anwendung (siehe unten) findet sich ein Beispiel: Ein Löschen von Bit 7 aktiviert die Pull-up-Funktion für die Eingänge.

Speicherorganisation

Für die Programmierung in Assembler ist es erforderlich, sich mit der Speicherorganisation des verwendeten Mikrocontrollers auszukennen. Der Speicher ist typischerweise (und so auch hier) in drei Bereiche aufgeteilt:

- Programmspeicher (Flash)
- Datenspeicher (SRAM)
- EEPROM

Der Programmspeicher dient in der Regel - und beim verwendeten PIC12F675 sogar ausschließlich - zur Ablage der Software. Ein Wort im Programmspeicher ist hier 14 bit breit. Für

die Programmierung ist diese Information zwar nicht so wichtig, doch macht dies klar, dass der Programmspeicher eines PIC12F1675 eben nicht 1024 **Bytes** (zu je 8 bit) sondern 1024 **Worte** (zu je 14 bit) umfasst.

Im Datenspeicher finden sich alle für Prozessor und Anwendung spezifischen Register. Mit Ausnahme der vordefinierten Prozessor-Register ist der Inhalt des Datenspeichers nach dem Einschalten undefiniert. Sein Inhalt geht bei Spannungsverlust verloren – es handelt sich eben um flüchtigen SRAM. Der Datenspeicher ist anders als der Programmspeicher ganz klassisch in Bytes zu je 8 bit organisiert. Die detaillierte Darstellung der Memory-Map findet sich im Datenblatt [1] ab Seite 9. Merken sollte man sich hier, dass die 64 Byte Anwendungsspeicher im Adressbereich 20h...5Fh liegen.

Im ebenfalls byte-organisierten EEPROM kann man per Software Werte ablegen, die auch einen Reset oder einen Neustart überleben, da es sich um nichtflüchtigen Speicher handelt. Allzu häufiges Beschreiben sollte allerdings vermieden werden.

Weitere Befehle

Nun folgt eine Beschreibung der Befehle, die zusätzlich zu den bereits im ersten Kursteil besprochenen Befehlen für unser Demo-Projekt benötigt werden. Übrigens findet sich in jedem Datenblatt zu jedem Mikrocontroller von Microchip eine komplette Kompaktliste aller Befehle [1].

Unsere Beschreibung beginnt mit der Vervollständigung eines Befehls, der schon im ersten Teil benutzt wurde.

DECFSZ

Im ersten Teil wurde die Funktion des Befehls (= **DEC**rement **F** and **S**kip if **Z**ero) für Schleifen besprochen. Außer Loops kann man mit DECFSZ aber auch allgemeine Verzweigungen im Programmlauf erreichen (eine Schleife ist eigentlich nur ein Spezialfall einer Verzweigung). **Bild 4** veranschaulicht das Ablaufdiagramm des Befehls.

Vorausgesetzt wird, dass eine Variable „X“ auf dem Speicherplatz 20h zu finden ist. Eine komplette Verzweigung benötigt drei Befehle (siehe **Bild 5**). Durch Verwendung des Arguments

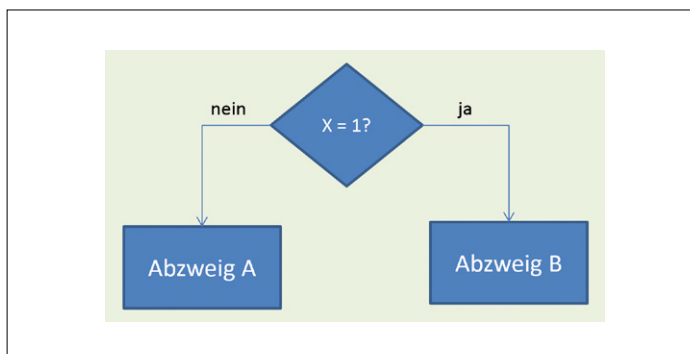


Bild 4. Ablaufdiagramm einer bedingten Verzweigung.

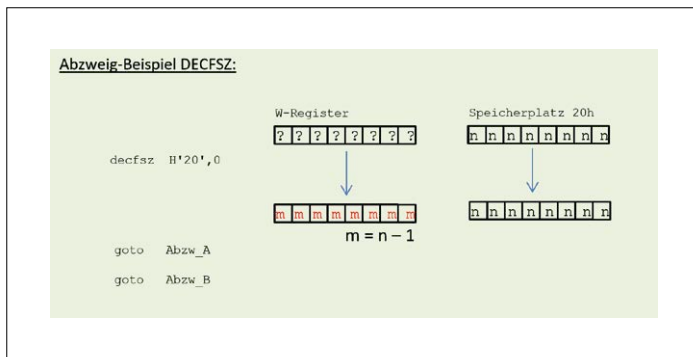


Bild 5. Code-Beispiel einer bedingten Verzweigung.

„0“ bei DECFSZ wird das Ergebnis der Subtraktion (Wert in „X“ minus 1) in das W-Register geschrieben. Falls das Ergebnis 00h ist, wird der nächste Befehl übersprungen und der dritte Befehl „goto Abzw_B“ ausgeführt. Das passiert nur dann, wenn der Wert von „X“ vor der Dekrementierung 01h ist. In allen anderen Fällen landet man bei „Abzw_A“.

INCFSZ

Der Befehl DECFSZ wurde nun ausführlich beschrieben. INCFSZ ist bis auf eine Inkrementierung statt der Dekrementierung identisch.

BTFSS und BTFSC

Um Verzweigungen geht es auch bei den beiden Befehlen BTFSS (= **B**it **T**est **F** and **S**kip if **S**et) und BTFSC (= **B**it **T**est **F** and **S**kip if **C**lear). Mit ihnen kann man den Status einzelner Bits eines Speicherplatzes abfragen und entsprechend verzweigen. Die Syntax des Befehls ist:

`btfss f,b`

bzw.

`btfsc f,b`

Das Argument „f“ hat einen Wert von 00h...7Fh und stellt die Adresse des Speicherplatzes dar. Argument „b“ ist eine

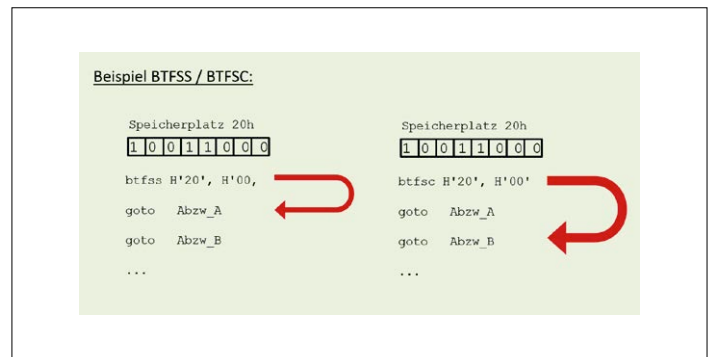


Bild 6. Schleifen-Beispiel mit den Befehlen BTFSS / BTFSC.

Zahl von 00h...07h und definiert das zu prüfende Bit. Diese Befehle ändern weder den Inhalt des W-Registers noch des Speicherplatzes.

Im Beispiel von **Bild 6** ist auf dem Speicherplatz 20h ein Wert von 98h (= 10011000b) abgelegt. Auf der linken Seite prüft der Befehl BTFSS, ob der Wert von Bit 00h den Wert „1“ hat. Falls ja wird der nächste Befehl „goto Abzw_A“ übersprungen. Weil der Wert von Bit 0 aber „0“ ist, wird nicht übersprungen und stattdessen der Befehl „goto Abzw_A“ ausgeführt.

Auf der rechten Seite sieht man die Wirkung des Befehls BTFSC. Hier wird übersprungen, wenn Bit 0 den Wert „0“ hat – was bei dem Wert an Speicherplatz 20h zutrifft. Der Befehl „goto Abzw_A“ wird also übersprungen und das Programm mit dem Befehl „goto Abzw_B“ fortgesetzt.

ADDWF

Mit diesem Befehl wird der Inhalt des W-Registers zum Inhalt eines Speicherplatzes addiert.

Die Syntax ist:

`addwf f,d`

Das Argument „f“ steht für die Zieladresse und hat einen Wert von 00h...7Fh. Das Argument „d“ kann „0“ oder „1“ sein. Es

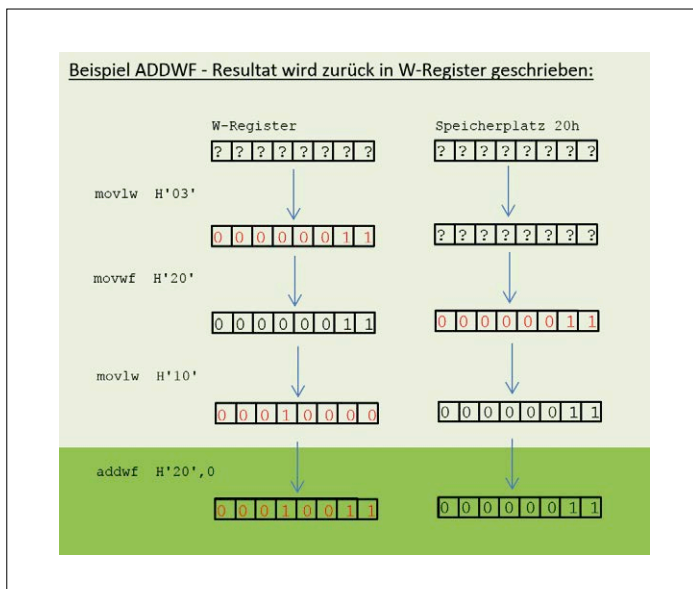


Bild 7. Code-Beispiel ADDWF: Ergebnis ins W-Register.

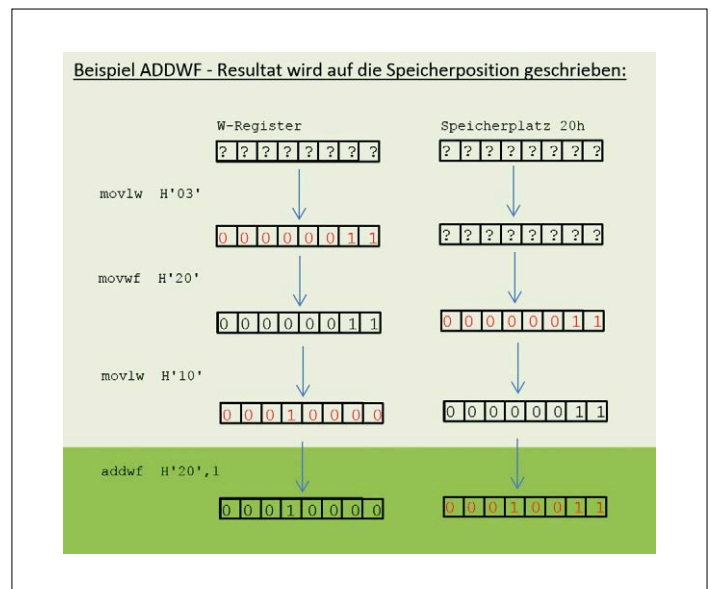


Bild 8. Code-Beispiel ADDWF: Ergebnis in SRAM-Speicherplatz.

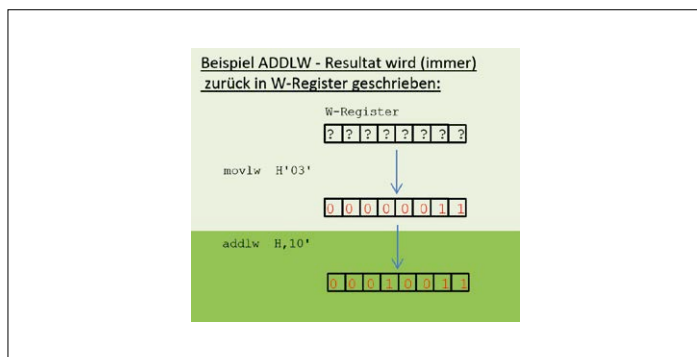


Bild 9. Addition einer Konstante zum W-Register.

definiert, wohin das Ergebnis gespeichert wird. Bei $d = 0$ wird das Resultat im W-Register gesichert, und der Speicherplatz bleibt unberührt. Bei $d = 1$ wird die Speicherposition mit dem Ergebnis überschrieben.

Im folgenden Beispiel wird die Summe von $3 + 16$ (dezimal) berechnet. Erst wird die „3“ auf dem Speicherplatz 20h abgelegt, danach wird „16“ (10h) ins W-Register geschrieben und zum Schluss die Addition ausgeführt. Das Ergebnis wird entweder ins W-Register (**Bild 7**) oder in Speicherposition 20h (**Bild 8**) geschrieben.

ANDWF

Bezüglich Syntax und Handhabung ist dieser Befehl identisch mit ADDWF. Der Unterschied liegt in der Funktion: ANDWF führt bitweise die logische Operation AND durch.

ADDLW

Ähnlich wie der Befehl ADDWF führt auch ADDLW eine Addition durch. ADDWF addiert allerdings einen Speicherwert zum W-Register, während ADDLW eine Konstante zum W-Register addiert. Die Syntax ist:

`addlw k`

Das Argument „k“ hat einen Wert von 00h...FFh und ist die Konstante, die zum Inhalt des W-Registers addiert wird. Das Ergebnis wird ins W-Register gesichert.

Im Beispiel von **Bild 9** wird zunächst der Wert 03h ins W-Register geladen und danach wird die Konstante 10h addiert.

ANDLW

Dieser Befehl ist bis auf die AND-Operation statt der Addition identisch mit ADDLW.

CLRF und CLRW

Der Befehl CLRF tauchte schon in Folge 1 auf (eine bestimmte Speicherposition wird auf 00h gesetzt). Analog dazu setzt der CLRW Befehl das W-Register auf 00h.

Die Syntax ist:

`clrf f`

bzw.

`clr w`

INCF und DECF

Mit diesen Befehlen kann man inkrementieren (INCF; +1) oder dekrementieren (DECF; -1). Die Syntax ist:

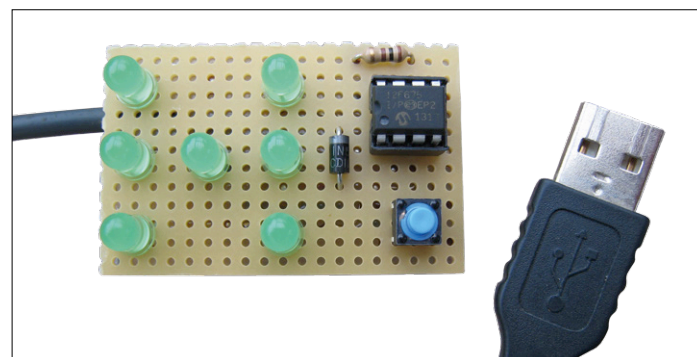


Bild 10. Prototyp des elektronischen Würfels.

`incf f,d`

bzw.

`decf f,d`

Die Argumente „f“ und „d“ haben die gleiche Bedeutung wie beim Befehl ADDWF. Das Ergebnis wird abhängig von „d“ in das W-Register oder an die ursprüngliche Speicherposition geschrieben.

Elektronischer Würfel

Die Realisierung eines Elektronischen Würfels ist nicht viel komplizierter als das obligatorische Blinken einer LED. Da die Schaltung der Einfachheit halber von einem USB-Port eines PCs versorgt wird, sieht man in **Bild 10** einen USB-Anschluss. Sonst hat der Würfel mit der USB-Schnittstelle nichts am Hut. Man kann ihn auch einfach mit einem Netzteil oder mit drei Primärzellen oder Akkus in Serie versorgen.

Schaut man sich einen richtigen Würfel genau an, dann sieht man, dass die Punkte darauf auf genau sieben verschiedenen Positionen liegen. Die Darstellung der einzelnen Zahlen folgt dem Schema in **Bild 11**. Wenn man diese Punkte durch LEDs darstellen lässt, dann braucht man dazu genau vier Ausgänge eines Mikrocontrollers. Wenn Sie jetzt staunen, hier die Aufklärung: Die LEDs 6 und 7, 2 und 3 sowie 4 und 5 sind immer nur gemeinsam an oder aus. Diese Paare werden daher jeweils gemeinsam von einem GPIO-Portpin angesteuert. LED1 erhält eine Steuerleitung alleine. Damit LED1 nicht heller als die anderen leuchtet, wird ihr R1 vorgeschaltet. Der direkte Anschluss von LEDs an GPIO-Pins (siehe die Schaltung von **Bild 12**) ist zwar nicht ganz regelkonform, jedoch aufgrund der niedrigen Ströme kein wirkliches Problem. Um etwas Energie zu sparen, werden die 5 V des USB-Ports mit D1 um ca. 0,7 V auf 4,3 V verringert. Falls man zur Versorgung drei NiMH-Akkus mit $3 \times 1,2$ V verwendet, kann man die Diode weglassen. Zum Würfeln drückt man einmal kurz auf Taster S1.

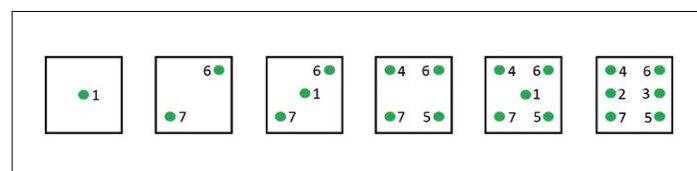


Bild 11. Darstellung der Zahlenwerte bei einem Würfel.

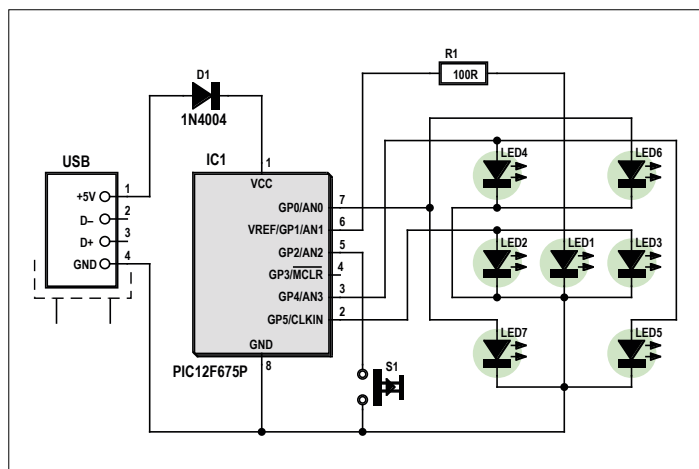


Bild 12. Die Schaltung des elektronischen Würfels.

Ein kleiner Hinweis noch: Aufgrund der höheren Durchlassspannung funktionieren in dieser Schaltung keine blauen oder weißen LEDs.

Würfel-Firmware

Bevor man in die Tasten haut, empfiehlt sich die genaue Spezifizierung dessen, was die Software machen soll. Die Spezifikation für den Würfel sieht so aus:

Nach dem Start (Reset) soll der Würfel eine Demo anzeigen. Die Demo läuft so ab, dass solange immer wieder nacheinander alle Zahlen (1...6) dargestellt werden, bis der Taster S1 gedrückt wird.

Die Würfel-Logik wird als Endlos-Schleife implementiert:

Wenn S1 gedrückt ist, werden die Werte 1 bis 6 immer wieder sehr schnell durchgezählt und dabei angezeigt. Durch die Trägheit des Auges sieht es so aus, als würde der Wert 7 angezeigt (alle LEDs an).

Die schnelle Zählung endet, wenn S1 losgelassen wird. Der aktuelle Zählerstand wird dann dauerhaft angezeigt. Jetzt kann man loslegen.

Die Firmware für den Würfel ist in zwei Teile aufgeteilt: Würfeln und Anzeigen. Selbstverständlich muss nach einem Reset oder Neustart zunächst einiges eingestellt werden.

Im Code von **Bild 13** sieht man diese Konfiguration bei Markierung 1. Eingestellt werden der interne Oszillator und GP3 als Input – folglich kein MCLR.

Bei Markierung 2 werden die Variablen TIMER1 und TIMER2 an den Speicherpositionen 20h und 21h definiert.

Bei den Markierungen 3 und 4 wird mit Speicherbank 1 gearbeitet, weswegen zunächst Bit RP0 des Status-Registers auf „1“ gesetzt wird.

Bei Markierung 3 sorgen die ersten zwei Befehle (`movlw B'11001100'` und `movwf TRSIO`) dafür, dass GP0, GP1, GP4 und GP5 als Ausgänge konfiguriert werden, indem die zugehörigen Bits des TRISIO-Registers auf „0“ gesetzt werden. An diese vier Pins sind die LEDs angeschlossen. GP2 wird als Eingang konfiguriert, denn hiermit wird der Taster-Status eingelesen. Hierzu wird Bit 2 des TRISIO-Registers auf „1“ gesetzt. Der Befehl „`clrf ANSEL`“ schaltet die analoge Funktionalität aus. Bei Markierung 4 wird ein externer Pull-up-Widerstand für den Taster überflüssig gemacht. Der Mikrocontroller PIC12F675 stellt nämlich so genannte „Weak Pull-ups“ für alle als Eingang konfigurierte GPIO-Pins (außer GP3) zur Verfügung. Hierzu muss Bit 7 des Option-Registers auf „0“ gesetzt werden. Im Programm erledigt dies der Befehl „`bcf OPTION_REG,D'007'`“. Anschließend legt man mit den Bits des WPU-Registers fest, für welche Pins die internen Pull-ups wirklich aktiv werden sollen. Wenn das Bit x des WPU-Registers auf 1 gesetzt wird, ist für den zugehörigen GPx-Pin der Pull-up-Widerstand aktiv. Die internen Pull-ups sind nicht grundsätzlich aktiv, da durch sie bis zu 400 μ A zusätzlich an Strom verbraucht werden kann. Wo es darauf ankommt, kann man dann einen hochohmigeren externen Pull-up oder bei Bedarf sogar einen externen Pull-down einsetzen.

Nach der Initialisierung springt das Programm in die Demo-Schleife (**Bild 14**). Die Implementierung ist an dieser Stelle sehr einfach – alles passiert in der Subroutine „start_effect“, die hier einfach aufgerufen wird. Der Demo-Teil wird mit einem Druck auf S1 beendet.

Demo-Routine

Die Demo-Routine in **Bild 15** ist sehr überschaubar. Man erkennt sechs nahezu identische Programmteile für jede Zahl. Für die Darstellung der Zahlen kommt eine Subroutine zum Einsatz: Unterprogramm „disp_1“ sorgt dafür, dass die entsprechende LED für die „1“ leuchtet. Unterprogramm „disp_2“ zeigt die Zahl „2“ an und so weiter.

Nachdem eine Zahl angezeigt wird, wartet die Software kurz bis zur nächsten Zahl. Dafür sorgt die „delay_routine“ – die schon im ersten Teil des Kurses verwendet wurde. Interessant an dieser Stelle ist der Befehl „btfs GPIO,D'002“. Er sorgt dafür, dass bei offenem Taster der Befehl „return“ übersprungen und die nächste Zahl angezeigt wird. Bei gedrücktem Taster wird nicht mehr übersprungen, denn der Eingang liefert eine „0“. Das Programm wird mit „return“ fortgesetzt. Das sorgt wieder dafür, dass die Subroutine „start effect“ und damit die

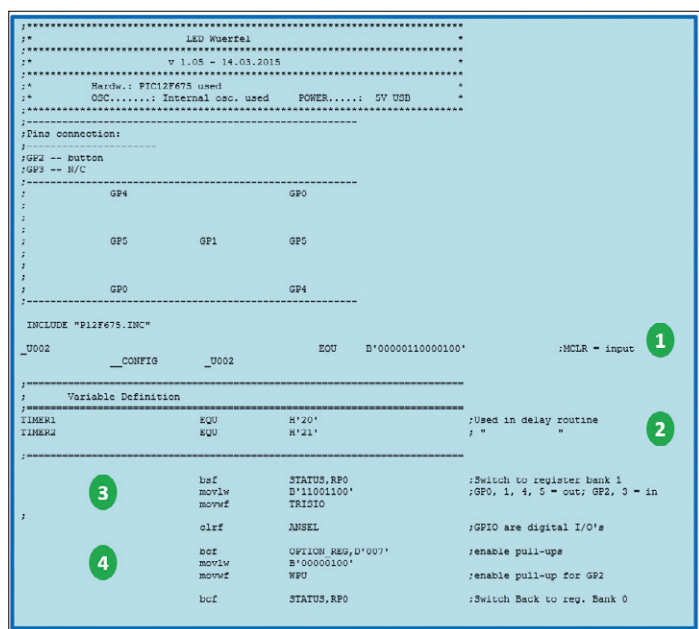


Bild 13. Würfel-Code: Initialisierungen.

www.elektormagazine.de September 2015 23

4Discovery: Intelligentes resistives Touch-Display mit Diablo16-Prozessor

Bei 4Discovery handelt es sich um ein intelligentes Display-Modul mit einem 3,5"-TFT-Panel mit 480 x 320 Pixeln, das mit resistiver Touch-Technik und dem leistungsfähigen Mikrocontroller Diablo16 ausgestattet ist. Das Modul kann autonom betrieben und sehr einfach mit der IDE 4D Workshop4 programmiert werden. 4Discovery eignet sich für die einfache Wandmontage in einer Standarddose für Lichtschalter, und ist deshalb für Anwendungen in der Heim-Automatisierung als Steuerung geeignet.

4Discovery verfügt über einen Slot für microSD-Karten, 16 MB Flash-Speicher, eine Echtzeituhr, ein Zweidraht-RS485-Interface (als Master oder Slave mit zusätzlicher Changeo-



ver-Leitung), optionales WLAN, einen optionalen Verschlüsselungs-Chip für sichere Daten-

übertragung sowie über ein Schaltnetzteil mit weitem Eingangsspannungsbereich.

Die Intelligenz stammt von einem Diablo16-Mikrocontroller, der den autonomen Betrieb ermöglicht und mit der IDE 4D Workshop4 programmiert wird. Workshop4 unterstützt die schnelle Realisierung grafischer Lösungen, da diese IDE speziell für die Grafik-Prozessoren von 4D Systems maßgeschneidert ist.

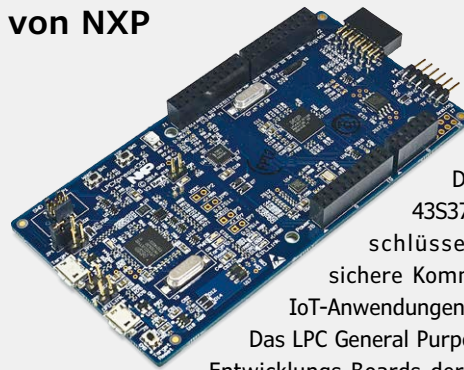
4Discovery wird mit und ohne WLAN und in verschiedenen Starter-Kits angeboten. 4Discovery ist direkt von 4D Systems oder über lokale Distributoren erhältlich. Die Software-Tools stehen kostenlos per Download zur Verfügung.

www.4dsystems.com.au (150273-3)

Neue LPC-Xpresso-Entwicklungs-Boards von NXP

Farnell element14 hat drei neue Boards von NXP aus der LPC-Xpresso-Familie in sein Portfolio aufgenommen.

Die neuen LPC-Xpresso-Boards der Version V3 sind Teil der gut eingeführten Toolchain auf der Basis von LPC-Mikrocontrollern mit ARM-Cortex-M-Cores. Die Boards arbeiten mit dem meisten existierenden Xpresso-Zubehör sowie mit Arduino-kompatiblen Shields zusammen, wodurch die Funktionen stark erweitert und für Anwendungen wie Motorsteuerungen, Leistungsmanagement, RFID-Leser, Embedded-Audio, Industrieautomation und Messtechnik angepasst werden können. Das ARM-Cortex-Entwicklungs-Board LPC Xpresso 18S37 ist eine gute Prototypenbasis für die Mikrocontroller der Reihe LPC1800. Sein ARM-Cortex-M3-Core läuft mit einem Takt von bis zu 180 MHz. Sowohl LPC Xpresso 4337 als auch 43S37 bilden eine Prototypengrundlage für die Mikrocontroller-Familie LPC4300. Die Dual-ARM-Cores (M4F und M0+)



können mit einem Takt von bis zu 208 MHz betrieben werden.

Die Typen 18S37 und 43S37 enthalten eine AES-Verschlüsselungs-Hardware für sichere Kommunikation, die z.B. in IoT-Anwendungen notwendig sein kann.

Das LPC General Purpose Shield kann mit den Entwicklungs-Boards der Versionen V2, V3, und MAX zusammenarbeiten und bietet gebräuchliche Peripherie wie ein LCD, LEDs oder einen Joystick mit fünf Positionen sowie Beschleunigungs- und Temperatursensoren.

www.element14.com/community/docs/DOC-74693 (150273-4)

Drahtlose Stromversorgung für Wearables

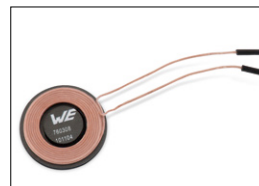
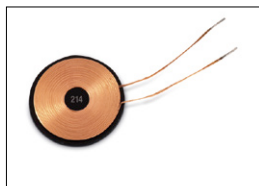
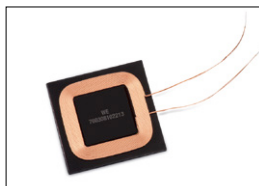
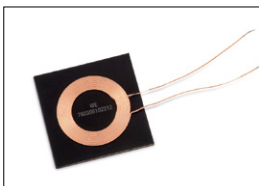
Würth Elektronik, der bekannte Hersteller von induktiven elektronischen und elektromechanischen Bauelementen, bietet ab sofort Wireless-Power-Spulen für Wearables an. Da Geräte wie Fitness-Sensoren und Smartwatches üblicherweise gegen Umwelteinflüsse gekapselt sind, durch die Vielfalt der integrierten Sensorik (GPS, Funk) aber häufig geladen werden müssen, ist eine Energieübertragung ohne Kabel und Stecker sinnvoll. Würth Elektronik bietet mit einer Erweiterung seiner bewährten Wireless-Power-Transfer-Produktfamilie zwei neue Sen-

der- und drei Empfängerspulen an, die optimal für Lösungen zum Laden von Wearables und Mobilgeräten geeignet sind. Die Spulen sind an die bereits existierenden Standardchipsätze für den Qi-Standard des WPC angepasst. Bestehende Lösungen können daher einfach durch den Austausch der Spulen eingesetzt werden.

Die Empfängerspulen WE-WPCC 760308102212 und WE-WPCC 760308102213 zeichnen sich durch ihre sehr geringe Höhe von gerade einmal 0,64 mm aus und sind damit optimal für Wearables mit Platz für

größere Spulenflächen (29 x 29 mm), aber einer starken Limitierung der Bauhöhe. WE-WPCC 760308102212 bietet darüber hinaus durch einen breiten Ferritüberstand eine ausgezeichnete Schirmung – wichtig für Anwendungen, die sensibel gegenüber elektromagnetischer Einstrahlung sind.

WE-WPCC 760308102213 ist durch die Wahl des Induktivitätswertes ein Dual-Standardempfänger. Mit dem entsprechenden Empfängerchip ist die Spule in der Lage, auf Sendern zu laden, die entweder dem Qi-Standard des WPC oder dem PMA-Standard entsprechen. Die dritte Empfängerspule WE-WPCC



760308101214 ist mit 0,8 mm etwas dicker, hat dafür aber nur einen Durchmesser von 19 mm. Durch ihre kompakte Bauform ist sie bestens geeignet für die kabellose Energieübertragung in Geräten mit begrenztem Platzbedarf. Sie erzielt sehr gute Übertragungswerte im Zusammenspiel mit den neuen Sen-

derspulen WE-WPCC 760308101104 und WE-WPCC 760308101105. Optimal zu diesen Empfängerspulen passen die neuen Wireless-Power-Charging-Sendespulen WE-WPCC 760308101104 und WE-WPCC 760308101105 für passende Ladestationen. Beide Spulen haben einen Außendurchmesser von nur 21 mm, nutzen hochwertiges Ferrit zur

Schirmung und Litzendraht, um den Gleichstromwiderstand klein zu halten. Sie unterscheiden sich durch ihren Induktivitätswert. Alle Wireless-Power-Spulen sind bei Würth Elektronik ab Lager verfügbar; Muster sind kostenlos erhältlich.

www.we-online.de (150335-6)

Kostenloses PicoScope-Upgrade mit 16 seriellen Schnittstellen

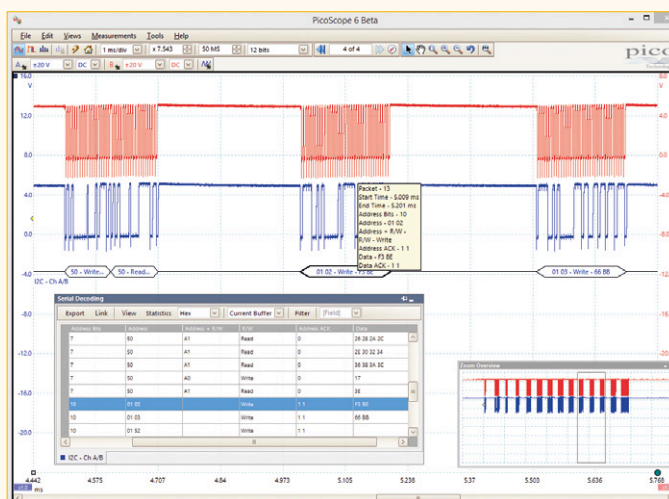
Aufgrund der zunehmenden Wichtigkeit serieller Kommunikation in der Elektronik ist in die neueste Version der Software für PicoScope nun die Fähigkeit zur Dekodierung und Analyse von 16 seriellen Protokollen integriert. Dank des umfangreichen Speichers in den meisten PicoScope-Versionen erlaubt die neue Software enorm leistungsfähiges Debugging und professionelle Fehleranalyse ohne Zusatzkosten. PicoScope R6.11.4 beta kann ab sofort kostenlos heruntergeladen werden und funktioniert mit jedem PicoScope.

Unterstützte Protokolle nach Anwendungsbereich:

- **Automotive:** CAN, LIN, FlexRay und SENT (Fast und Slow);
- **Avionics:** ARINC 429;
- **Computer & Netzwerk:** RS232/UART, USB (LS, FS, HS), PS/2 sowie Ethernet 100BASE-TX und 10BASE-T;
- **Embedded Systeme:** I²C, I²S, SPI und 1-Wire;
- **Beleuchtung:** DMX 512;
- **Modellbahn:** DCC.

Das serielle Decoding Display von PicoScope kann auch die Bus-Wellenform anzeigen und mit analogen und digitalen Quellen korreliert werden. Diese Anordnung erleichtert die Fehlersuche bei Datenfehlern, die durch Störungen, fehlerhafte Treiber oder nicht angepasste Impedanzen entstehen können.

Die alternative Tabellenansicht listet jedes Paket oder jeden Frame mitsamt Timing, den dekodierten Daten und optional mit Spannungsmessungen auf. Diese Tabelle kann man filtern und nach jedem Feld durchsuchen sowie die Daten für eine Tabellenkalkulation exportieren.



Man kann sogar für Menschen gut verständliche Namen für spezielle Adressen oder Datenwerte mit Hilfe eines „link file“ anlegen.

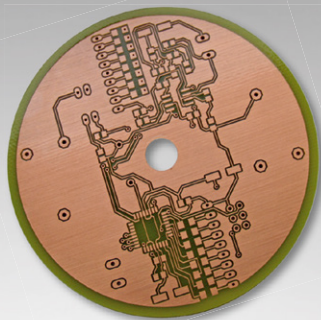
PicoScope kann viele serielle Busse gleichzeitig überwachen und das mit jeder Mischung an Protokollen, begrenzt lediglich durch die Anzahl verfügbarer Kanäle (bei MSO-Modellen maximal 20).

PicoScope R6.11.4 beta kann ab sofort kostenlos von www.picotech.com/downloads heruntergeladen werden.

Weitere Details zu den seriellen Protokollen finden sich unter www.picotech.com/library/oscilloscopes/serial-bus-decoding-protocol-analysis.

www.picotech.com (150335-1)

Anzeige



(M)ein kleiner Freund im Elektroniklabor

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

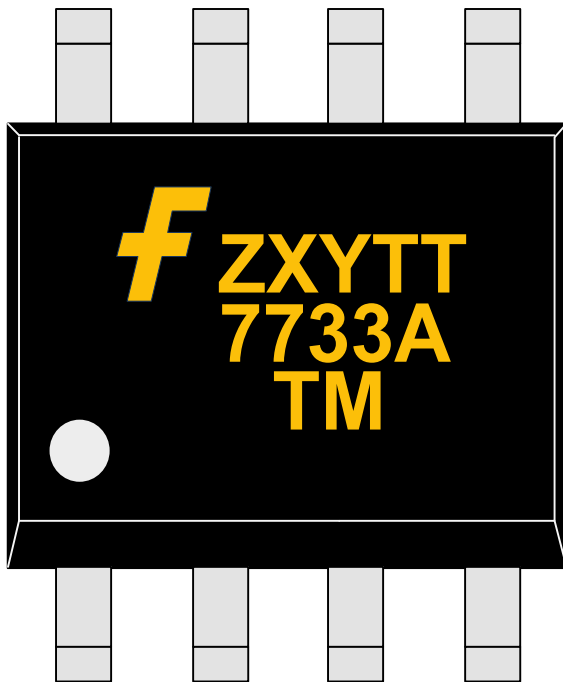
LPKF ProtoMat E33 – klein, präzise, wirtschaftlich

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

productronica: 10. – 13.11.2015, Halle B3, Stand 303



LPKF
Laser & Electronics



LED-Treiber

mit großem Ausgangsspannungsbereich

Von Inki Park



Seit LED-Lampen so vielfältig eingesetzt werden, gibt es sie in unterschiedlichsten Formen und Größen und für alle möglichen Fassungen. Konsequenterweise sollten LED-Treiber daher so konstruiert sein, dass sie für unterschiedliche LED-Typen und auch für spezielle Anwendungen geeignet sind. Selbstverständlich muss man dabei den Power Factor und die Netzstörungen im Blick behalten.

Die Versorgungsspannung der LEDs einer LED-Lampe hängt in erster Linie von der Anzahl in Serie geschalteter LEDs ab und in zweiter Linie von deren Temperatur. Der Strom durch die LEDs sollte dabei unter allen Umständen konstant bleiben, da die Helligkeit direkt proportional zum Strom ist. Gleichzeitig ist ein hoher PF (Power Factor) für die Schaltungsentwicklung genau so wichtig wie geringe Netzstörungen. Wenn dann der Treiber noch mit einem großen Bereich der Ausgangsspannung aufwarten kann, ist er mit vielen LED-Konfigurationen kompatibel. In diesem Artikel wird daher ein integrierter PWM-Controller vorgestellt, der mit einer fortschrittlichen PSR-Technik (Primary-Side Regulation) arbeitet. Resultat ist ein Design-Leitfaden für einen einstufigen sogenannten Flyback-Konverter bzw. Sperrwandler, der einen großen Ausgangsspannungsbereich aufweist.

Prinzip der Primärseitenregelung

Auch mit PSR erreicht man bei LED-Treibern die Übereinstimmung mit internationalen Regularien wie etwa dem Energy Star® für Halbleiter-Lampen (SSL, Solid-State Lighting). Bei der PSR steuert man den Ausgangsstrom nur mit primärseitigen Informationen und doch präzise. Dadurch vermeidet man die Verluste, die z.B. bei der Messung des sekundären Stroms per Shunt entstehen, und man spart gleichzeitig den Aufwand für eine sekundäre Feedback-Schleife. Daher passt

die Treiber-Schaltung auch für kleinere Lampen, die sich zur Nachrüstung bestehender Beleuchtungen eignen. Außerdem werden die gesetzlichen Grenzwerte für SSLs auch ohne erhöhte Kosten eingehalten. Eine passende Lösung ist der PWM-PSR-Controller FL7733 von Fairchild. Dieser Chip vereinfacht die Einhaltung der Grenzwerte und spart externe Bauteile ein. Das IC FL7733 bietet eine hochpräzise Regelung des Ausgangstroms auf der Basis der Veränderung der Induktivität des Trafos, Informationen über Ein- und Ausgangsspannung sowie leistungsfähige Schutzmaßnahmen zur Steigerung der Zuverlässigkeit.

Modus I

Während der Einschaltzeit t_{ON} des MOSFETs Q liegt die Eingangsspannung V_{IN} voll an der Primärwicklung L_m des Trafos. Dabei steigt dann kontinuierlich der Strom I_{DS} durch den MOSFET von null bis zu seinem Spitzenwert $I_{DS,PK}$, wie in **Bild 1** zu sehen. Währenddessen wird Energie aus dem Netz entnommen und im Trafo magnetisch gespeichert.

Modus II

Wenn Q ausgeschaltet wird, sorgt die gespeicherte Energie durch die Sekundärwicklung dafür, dass der Gleichrichter D durchschaltet. Während die Diode leitet, liegt an der Sekundärwicklung die Ausgangsspannung V_{OUT} plus die Flussspannung der Diode V_F an, und der Dio-

denstrom I_D fällt kontinuierlich von seinem Spitzenwert $I_{DS,PK} \times N_p/N_s$ auf Null. Am Ende der Entladungszeit t_{DIS} ist alle im Trafo gespeicherte Energie zum Ausgang übertragen worden.

Modus III

Wenn der Diodenstrom den Wert Null erreicht, zeigt die Spannung V_A aufgrund der Resonanz der Induktivität der Primärwicklung L_m mit der effektiven kapazitiven Last des MOSFETs an der Hilfswicklung des Trafos Oszillationen.

Den Ausgangsstrom kann man aufgrund des Spitzenwerts des Drain-Stroms und der Entladezeit des Trafos abschätzen, da der mittlere Ausgangsstrom ja dem mittleren Diodenstrom in der Leitphase entspricht. Der Spitzenwert des Drain-Stroms entspricht dem Spannungsabfall V_{CS} am Shunt R_s . Die Entladezeit wird über die aus der Hilfswicklung abgeleiteten Spannung V_s vom t_{DIS} -Detektor des ICs gemessen. Mit $I_{DS,PK}$, t_{DIS} und den Informationen über die Schaltzeiten des ICs wird mit Hilfe des neuartigen Berechnungsverfahrens TRUECURRENT® der Ausgangsstrom wie folgt bestimmt:

$$I_o = \frac{1}{2} \times \frac{t_{dis}}{t_s} \times V_{CS} \times \frac{N_p}{N_s} \times \frac{1}{R_s} \quad (1)$$

$$\frac{t_{dis}}{t_s} \times V_{CS} = 0,25 \quad (2)$$

Zwecks eines hohen PF sollte bei einer Flyback-Topologie der DCM (Discontinuous Conduction Mode) eingehalten werden. Um dies über einen großen Bereich der Ausgangsspannung zu gewährleisten, wird die Schaltfrequenz linear an die Ausgangsspannung angepasst. Die Höhe der Ausgangsspannung wird mit Hilfe der über einen Spannungsteiler von der Hilfswicklung abgeleiteten Spannung am Pin V_s gemessen (siehe **Bild 2**). Wenn die Ausgangsspannung sinkt, wird die Einschaltzeit der sekundären Diode erhöht, und die DCM-Steuerung verlängert die Schaltzeit, was einen sicheren DCM-Betrieb über einen großen Ausgangsspannungsbereich gewährleistet.

LED-Treiber mit großem Ausgangsspannungsbereich

In diesem Abschnitt geht es um das Design eines einstufigen Flyback-LED-Treibers mit dem IC FL7733. Für diesen Zweck wurde ein LED-Treiber mit einer Leistung von 50 W als Beispiel entworfen. Die technischen Anforderungen sind:

- Eingangsspannungsbereich:
90...277 V_{AC} bei 50...60 Hz
- Nennausgangsspannung und -strom:
50 V; 1 A
- Ausgangsspannungsbereich:
12...50 V
- Minimaler Wirkungsgrad: 88 %
- Schaltfrequenz: 65 kHz
- Maximale Einschaltzeit: 40 %

Die Anzahl der Primärwindungen bestimmt man nach Faradays Induktionsgesetz. $N_{p,min}$ ist durch den Spitzenwert der minimalen Spannung über der Primärwicklung und die maximale Einschaltzeit festgelegt. Zur Vermeidung einer Kernsättigung ergibt sich die minimale Windungszahl der Primärwicklung zu:

$$N_{p,min} = \frac{V_{IN,min,pk} \times t_{on}}{B_{sat} \times A_e} \quad (3)$$

wobei A_e der Querschnittsfläche des Kerns in mm² und B_{sat} der Sättigungsflussdichte in Tesla entspricht.

$$V_{CS,pk} = I_{SW,pk} \times R_s \quad (4)$$

Laut Gleichung 3 ergibt sich das Primär/Sekundär-Windungsverhältnis mit Widerstand R_s und Ausgangsstrom I_0 zu:

$$n_{PS} = \frac{I_0 \times R_s}{0,125} \quad (5)$$

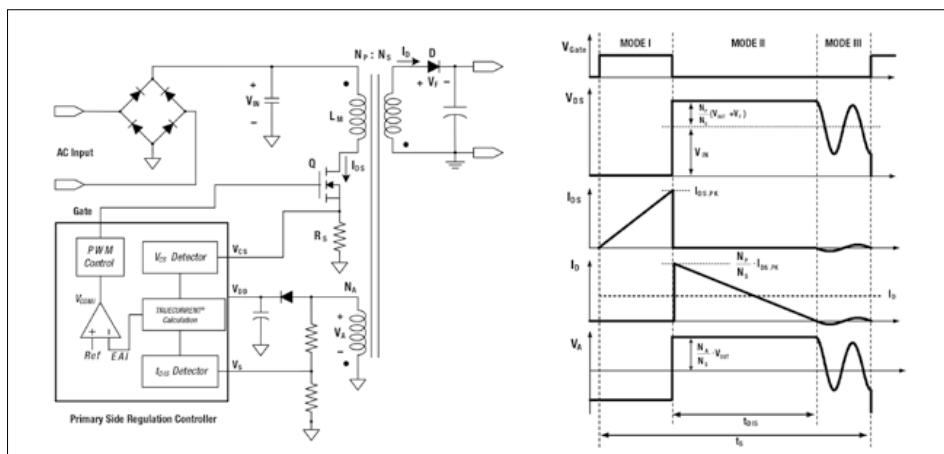


Bild 1. Primär geregelter Sperrwandler samt zugehöriger Signalverläufe.

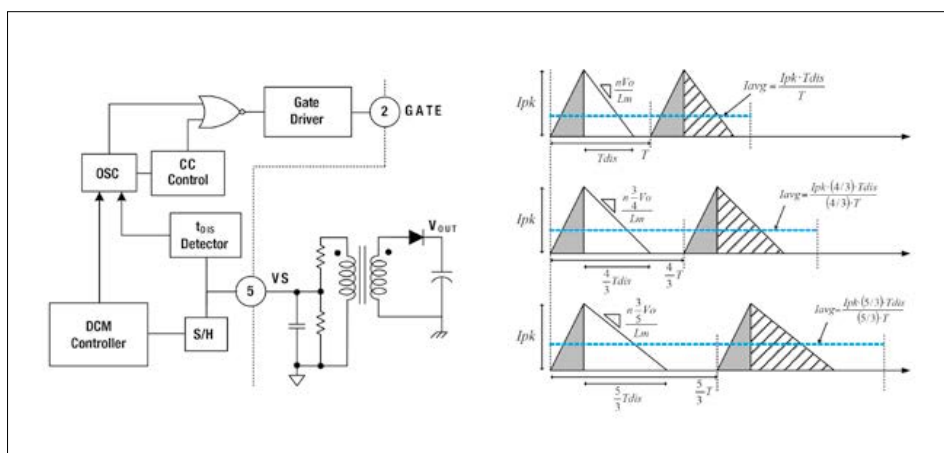


Bild 2. Prinzip der DCM-Steuerung.

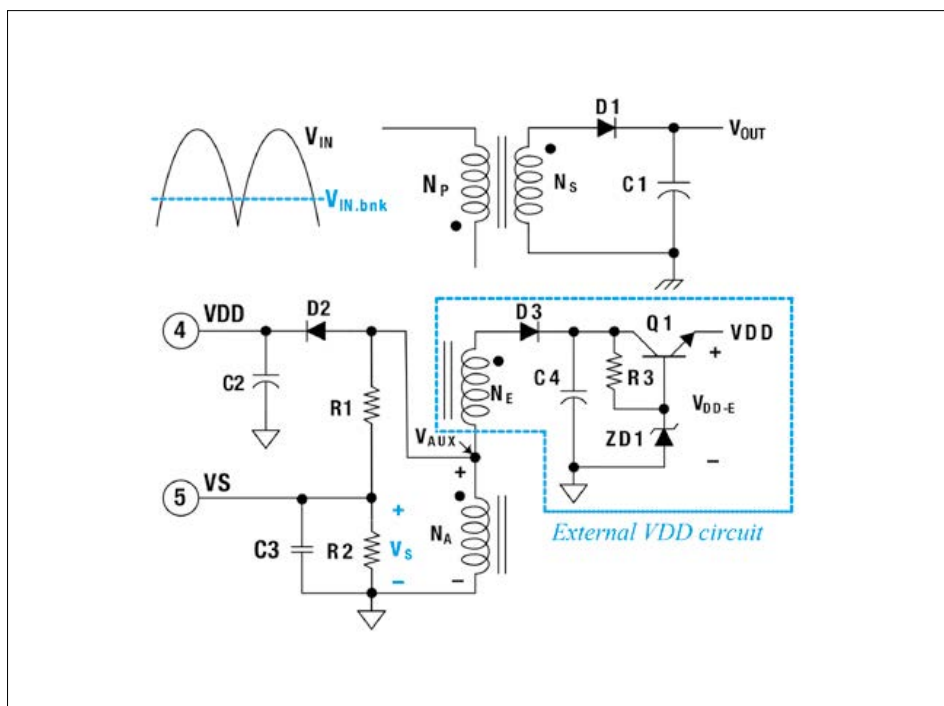


Bild 3. Erzeugung von V_s für einen großen Ausgangsspannungsbereich.

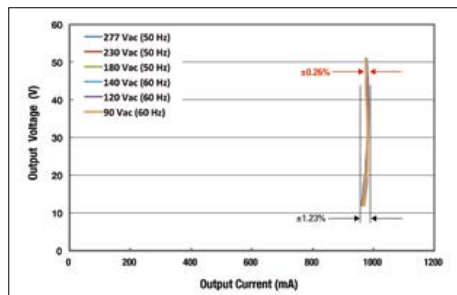


Bild 4. Stabilität des Konstantstroms in Abhängigkeit von der Ein- und Ausgangsspannung.

$$n_{AS} = \frac{V_{DD, OVP}}{V_{O, OVP}} = \frac{23}{V_{O, OVP}} \quad (6)$$

$$n_{AP} = \frac{n_{AS}}{n_{PS}} \quad (7)$$

Da die Sättigungsflussdichte mit der Temperatur steigt, muss man die Eigenschaften bei höheren Temperaturen berücksichtigen, wenn man den Treiber in ein geschlossenes Gehäuse einbaut.

Als erste Überlegung für die Berechnung von R1 und R2 wählt man die Spannung $V_S = 2,7 \text{ V}$, um den Betrieb mit hoher Frequenz bei Nennleistung zu gewährleisten. Die zweite Überlegung betrifft die Austastung von V_S . Die Ausgangsspannung wird ja über die Hilfswicklung und über einen Spannungsteiler am Pin V_S gemessen (siehe **Bild 3**). Mit einem einstufigen Sperrwandler ohne entsprechendem Kondensator kann bei niedriger Netzspannung die Spannung der Hilfswicklung nicht so geklemmt werden, dass sie der Ausgangsspannung entspricht, da ein kleiner Strom durch L_m einen Fehler bei V_S verursacht. Die Frequenz sinkt in der Nähe des Nulldurchgangs der Netzspannung rapide, was zum sichtbaren Flackern des LED-Lichts führen kann. Um über die ganze Sinuskurve mit konstanter Frequenz arbeiten zu können, wird die Auswertung von V_S bei Spannungen unter $V_{IN, bmk}$ ausgetastet. Die dritte Überlegung zielt auf den Wert von V_S , der irgendwo zwischen 0,6 und 3 V liegen sollte, um das Triggern von SLP und V_S OVP bei einem großen Ausgangsspannungsbereich zu vermeiden.

Da die Betriebsspannung V_{DD} des FL7733 im Bereich von 8,75...23 V liegen muss, wird die Ansteuerung des MOSFETs via U_{VLO}

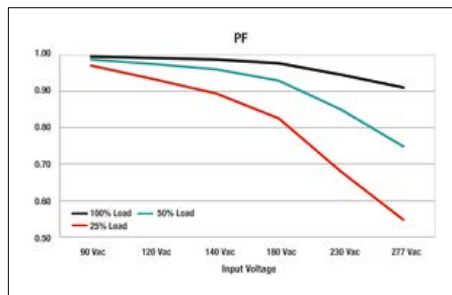


Bild 5. PF mit Eingangsspannung und Last als Parameter.

abgeschaltet, wenn die Ausgangsspannung niedriger als $V_{OUT-UVLO}$ ($8,75 \times N_S / N_A$) werden sollte. Daher sollte V_{DD} bei Ausgangsspannungen von 12...50 V bereitgestellt werden, ohne das U_{VLO} getriggert wird. V_{DD} kann auch über die Zusatzwicklung N_E und zusätzliche Bauteile wie in Bild 3 gewährleistet werden. N_E sollte dabei so ausgelegt werden, dass V_{DD} ohne Triggern von U_{VLO} auch bei minimaler Ausgangsspannung $V_{min, OUT}$ bereitgestellt werden kann. N_E errechnet sich zu:

$$N_E > \frac{(8,75 + V_{CE, Q1} + V_{F, D3})}{(V_{F, D1} + V_{min, OUT})} \times N_S - N_A \quad (8)$$

wobei $V_{CE, Q1}$ der Kollektor-Emitter-Sättigungsspannung von Q1 und $V_{F, D3}$ der Flussspannung von D3 entspricht. $V_{F, D1}$ steht für die Flussspannung von D1 bei minimaler Ausgangsspannung.

$$R1 = n_{AP} \times \frac{V_{IN, bmk}}{I_{VS, bmk}} \quad (9)$$

wobei $V_{IN, bmk}$ dem Pegel der Netzspannung und $I_{VS, bmk}$ dem Strom für das Blanking (Austasten) von V_S entspricht.

$$R1 = \frac{R2 \times 2,7}{(V_{OUT} + V_{F, D1}) n_{as} - 2,7} \quad (10)$$

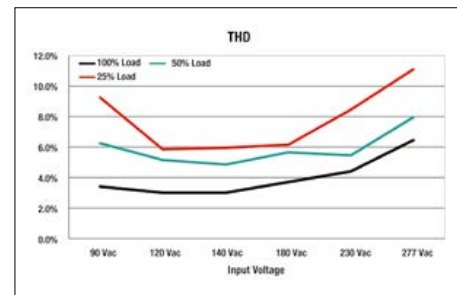


Bild 6. Verzerrungen (THD) mit Eingangsspannung und Last als Parameter.

Um die Gültigkeit der hier durchgeführten Überlegungen zu überprüfen, wurde die beschriebene Schaltung aufgebaut und getestet. **Bild 4** zeigt die Änderung des Konstantstroms über die ganzen Bereiche der Netz- und der Ausgangsspannung. Die Stromabweichung ist dabei abhängig von der Netzspannung kleiner als $\pm 0,26 \%$ und abhängig von sowohl der Netz- als auch der Ausgangsspannung mit $\pm 1,23 \%$ immer noch sehr gut. Die **Bilder 5** und **6** zeigen den PF und die Verzerrungen der Netzspannung (THD) bei verschiedenen Spannungs- und Lastbedingungen.

Fazit

Die hier vorgestellten Design-Überlegungen für einen LED-Treiber mit großem Ausgangsspannungsbereich haben zum Ziel, die Flexibilität und die Kompatibilität bezüglich unterschiedlicher Spezifikationen und Charakteristiken des LED-Betriebs zu erhöhen. Der PSR-Controller FL7733 von Fairchild liefert dabei eine gute Leistung bei hohem PF und niedrigen Störpegeln sowie einen sehr stabilen Ausgangsstrom über einen weiten Bereich der Ausgangsspannung. Die hier vorgestellte Schaltung eignet sich zum Betrieb unterschiedlichster LED-Konfigurationen bei niedrigen Kosten. **IK**

(150370)

Über den Autor

Inki Park arbeitet seit 2010 als Applikations-Ingenieur bei Fairchild Semiconductor. Im Rahmen seiner Tätigkeit entwickelt er primär geregelte Controller für LED-Treiber und Hochleistungsschaltnetzteile mit PF-Korrektur. Vor seiner Tätigkeit bei Fairchild war er schon sechs Jahre mit der Entwicklung diverser Netzteile und LED-Treiber befasst. Inki Park hat die Abschlüsse Master of Science und Bachelor of Science als Elektronik-Ingenieur von der Kangwon National University in Kangwon in Südkorea.



Willkommen bei DESIGN

Von **Clemens Valens**, Elektor Labor



Lizenz für ... für wen oder was nun eigentlich?

Eines der verworrensten Dinge in der Elektronik ist die Lizenzierung. Wenn ein Gerät oder Produkt geschlossen ist (und die meisten kommerziellen Dinge sind es), dann kann man davon ausgehen, dass es von internationalen Handels- und Urheberrechten geschützt ist. Diese sind ziemlich kompliziert zu verstehen (von einem Laien schon gar nicht), aber es gibt eine detaillierte Rechtsprechung, was man darf und was man nicht darf.

Wenn Sie jedoch eine Entwicklung als open source freigeben wollen, sei es Hardware, Software oder beides, wird es richtig kompliziert. Die Probleme liegen in der Natur der Open-Source-Bewegung selbst. Eine typische Open-Source-Lizenz erlaubt dem Lizenznehmer, das lizenzierte Werk abzuwandeln, was oft zu einer Anzahl von „abgeleiteten“ Designs führt, alle ein wenig anders. Der gleiche Effekt

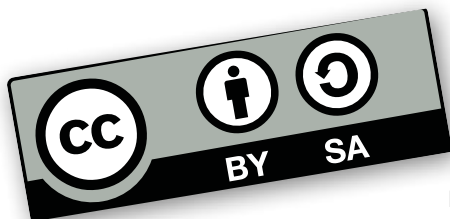
trifft auf die Open-Source-Lizenzen selbst zu. Inzwischen gibt es so viele verschiedene Open-Source-Lizenzen im Netz, dass es schwer, wenn nicht unmöglich ist, gerade die richtige für sich selbst zu finden.

Die Lizenzen sind juristische Dokumente, die vor allen Gerichten dieser Welt Bestand haben soll(t)en. Sie sind nicht nur sehr schwer zu verstehen, sondern entwickeln sich auch beständig weiter. Und alles, was man will, ist doch nur, eine eigene Entwicklung zu teilen, indem man eine Open-Source-Lizenz draufklebt. Stattdessen versinkt man in einem Sumpf von Juristen-Kauderwelsch...

Und man kann das Ganze nicht einfach abtun mit: „Ist umsonst, macht damit, was ihr wollt, ich hafte für nichts und niemanden!“. Wenn jemand etwas Illegales mit Ihrer Arbeit anstellt, könnte das von einem Gericht als zu leichtfertig befunden werden.



Nutzlose und unvollständige Liste aller Lizenzen (in alphabetischer Reihenfolge)



AAL, AFL-3.0, AGPL-3.0, Apache-2.0, APL-1.0, APSL-2.0, Artistic-2.0, BSD-2-Clause, BSD-3-Clause, BSL-1.0, CATOSL-1.1, CC BY, CC BY-NC, CC BY-NC-ND, CC BY-ND, CC BY-SA, CC-BY-NC-SA, CDDL-1.0, CECILL-2.1, CERN OHL, CNRI-Python, CPAL-1.0, CUA-OPL-1.0, ECL-2.0, EFL-2.0, Entessa, EPL-1.0, EUDatagrid, EUPL-1.1, Fair, Frameworkx-1.0, GPL-2.0, GPL-3.0, HPND, IPA, IPL-1.0, ISC, LGPL-2.1, LGPL-3.0, LPL-1.02, LPPL-1.3c, MirOS, MIT, Motosoto, MPL-2.0, MS-PL, MS-RL, Multics, NASA-1.3, Naumen, NCSA, NGPL, Nokia, POSL-3.0, NTP, OCLC-2.0, OFL-1.1, OGTSL, OSL-3.0, PHP-3.0, PostgreSQL, Python-2.0, QPL-1.0, RPL-1.5, RPSL-1.0, RSCPL, SimPL-2.0, Sleepycat, SPL-1.0, TAPR, UPL, VSL-1.0, W3C, Watcom-1.0, WXwindows, Xnet, Zlib, ZPL-2.0. ◀

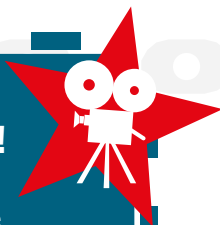
(150332-1)

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.

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 gekritzelten 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 Jugendlerner oder Student: Eine Veröffentlichung in Elektor ist DER Booster für einen tollen Job!



Unsere Geschichte

Die Ursprünge des Elektor-Labors gehen zurück auf die frühen 1970er-Jahre. Lötten 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.

Unsere Einrichtungen

Wir sind prächtig in drei geräumigen Zimmern im Elektor-Castle untergebracht, wo wir vergeblich versuchen, unsere Computer-Schreibtische frei von Lötspritzern 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.

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.



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.

Unsere Webinare

Die gesprächigeren 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.

elektor^elabs

Sharing Electronics Projects

Home Proposals In Progress Finished

Search

Log In

Laden Sie Ihre eigenen Projekte hoch!

Auf unserer Webseite Elektor-Labs.com können Sie Ihre eigenen Projekte vorstellen und mit tausenden anderer Elektroniker weltweit teilen. Sie können auf dieser Seite auch den Aktivitäten anderer Elektroniker folgen, sie kommentieren und so den Projekten zur Fertigstellung verhelfen. Und das schönste ist, dass die besten Projekte in unserem „echten“ Elektor-Labor nachbearbeitet, getestet und schließlich in der Elektor-Zeitschrift veröffentlicht werden.

Wer kann, wer darf?

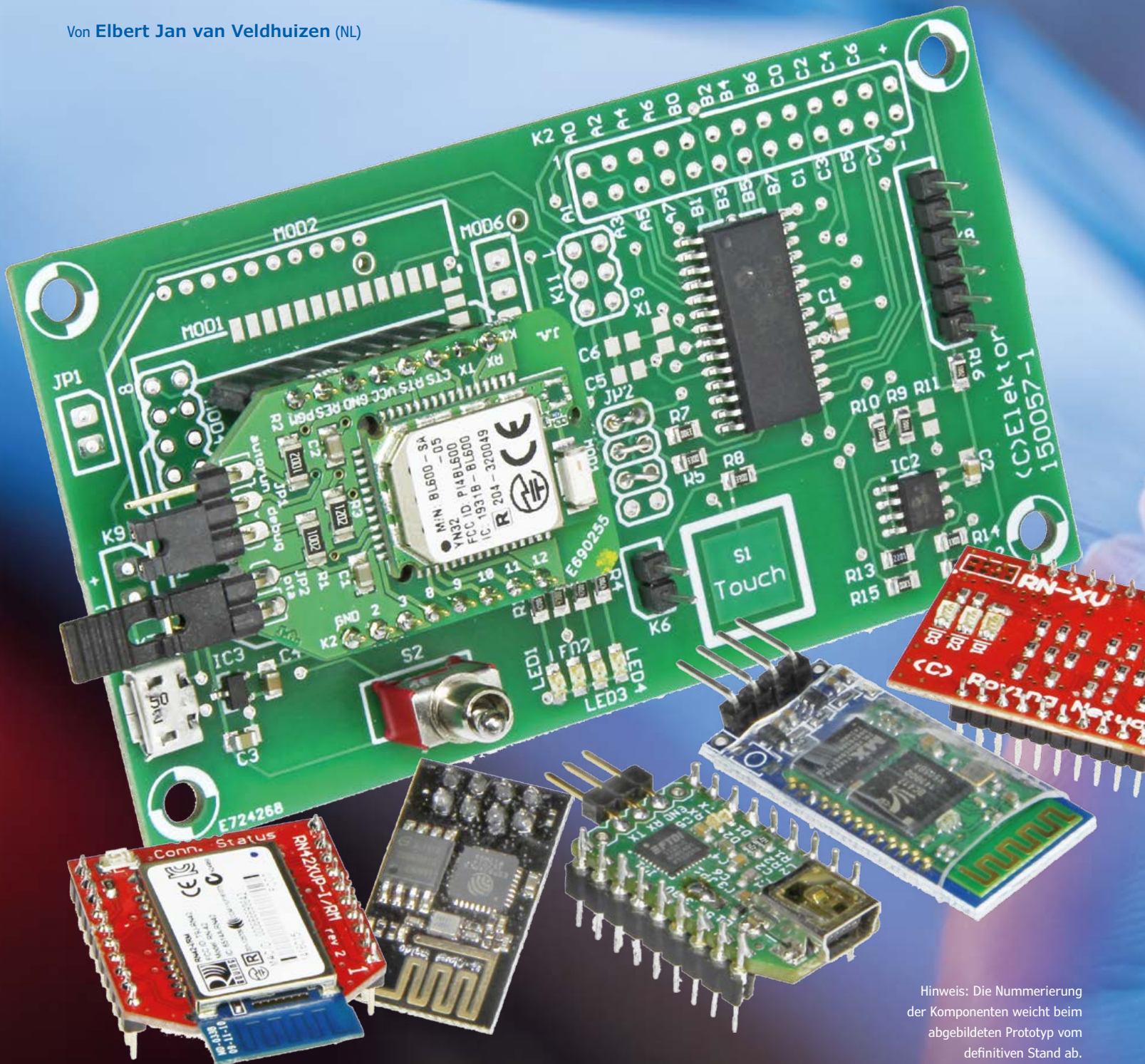
Jeder kann Projekte auf Elektor-Labs.com ansehen. Aber nur Mitglieder dürfen sich auf der Webseite einloggen, um dort eigene Projekte vorzustellen und andere zu kommentieren.

Möchten Sie, dass auch Ihr Projekt in vier Sprachen in der Zeitschrift Elektor veröffentlicht und von zehntausenden Elektronikern auf der ganzen Welt studiert wird? Dann werden Sie grünes oder goldenes Mitglied (www.elektor.de/mitgliedschaft) und loggen Sie sich als Mitglied unserer Elektor-Community ein!

Android I/O-Board (1)

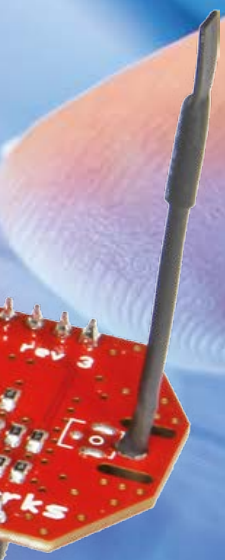
Steuern Sie eigene Elektronik mit Ihrem Android-Smartphone oder -Tablet

Von Elbert Jan van Veldhuizen (NL)



Hinweis: Die Nummerierung der Komponenten weicht beim abgebildeten Prototyp vom definitiven Stand ab.

Um eigene Elektronik anzusteuern, sind Android-Geräte nahezu ideal: Hohe Rechenleistung, Touchscreen und drahtlose Kommunikation sind vereint mit niedrigen Kosten. Mit dem hier vorgestellten I/O-Board und der zugehörigen App kann das Android-Smartphone oder -Tablet auf 22 I/O-Leitungen schalten, messen, Motoren steuern, analoge Signale akquirieren oder generieren und vieles mehr. Im ersten Teil dieses Beitrags beschreiben wir die Hardware.



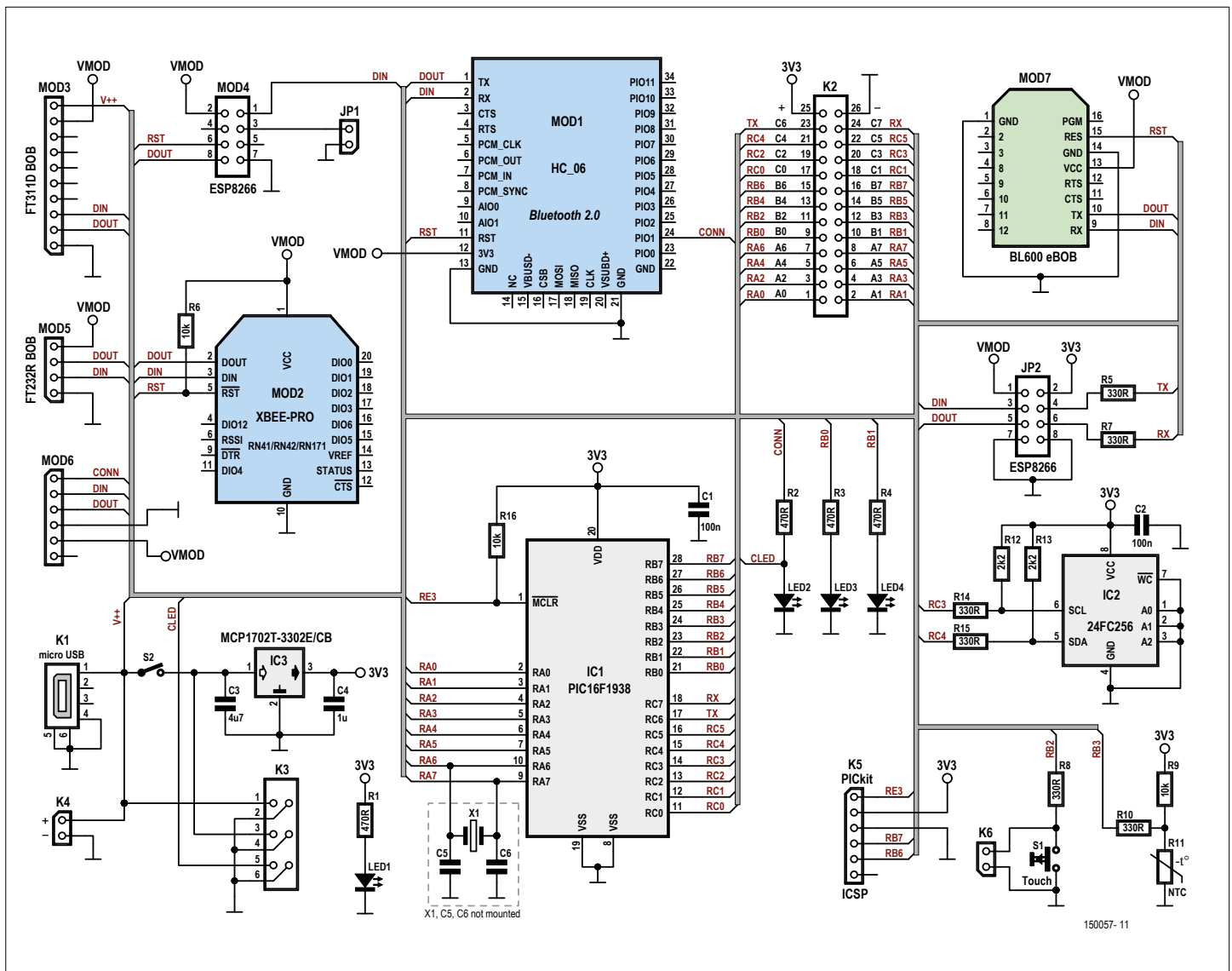


Bild 1. Der erste Anschein täuscht: Auf dem Board wird nur ein einziges Kommunikationsmodul nach Wahl verbaut.

Tabelle 1. Eigenschaften des Android I/O-Boards.

Funktion	Anzahl Pins (max. 22) / Wert	Anmerkung
Digitale Eingänge	22	8 Pins mit optionalen Pullups
Digitale Ausgänge	22	Ausgangsstrom 25 mA
ADCs	11	Auflösung 10 bit, ungefähr 20 kSamples/s
Kapazitiver Berührtaster	8	7 Pins Überlappung mit den ADC-Pins
PWM	8	generiert durch Software, 4 · 2 in Bridge-Mode
Counter	16-bit-Counter	bis 8 MHz
Datenrate	100 Instruktionen/Sekunde	Verbindung 9600 Baud
Datenspeicher (Flash, I2C)	32 Kbyte	mit 25FC256
Extras auf dem Board	4 LEDs, 1 NTC, 1 Berührtaster	
Betriebsspannung	4...13 V unstabilisiert 3,3 V stabilisiert	
Strombedarf (ohne Last)	Max. 30 mA (Bluetooth) Max. 250 mA (WiFi) Max. 5 mA (USB) und Strom für USB-Modul	
Anschluss Stromversorgung	Micro-USB-Buchse, Platinenstecker	

Das Board kann sich über Bluetooth, WLAN oder USB mit einem Android-Smartphone oder -Tablet verbinden.

Im Fokus des mehrteiligen Beitrags steht ein I/O-Board zusammen mit einer App, die gemeinsam die Signale auf 22 I/O-Leitungen über unkomplizierte Kommandos steuern und verwalten. Das Board kommuniziert wahlweise über Bluetooth, WLAN oder USB mit dem Android-Gerät. Die I/O-Leitungen sind nicht nur als digitale Eingänge oder Ausgänge nutzbar, sondern auch für ADCs, kapazitive Berührtaster, Pulsbreitenmodulatoren oder Counter. Die Firmware auf dem Board lässt sich modifizieren, so dass die Funktionalität des Mikrocontrollers für Real-time-Anwendungen erweitert werden kann. Für diesen Zweck ist im Mikrocontroller ein Bootloader integriert. Zum Steuern des Android-I/O-Board entwickelte der Autor Java-Classes (Libraries), die konsistente Methodennamen haben. Der Anwender muss sich nicht mehr in die Details vertiefen, er kann sich sofort dem Programmieren der App zuwenden.

Die Themen im ersten Teil des Beitrags sind die Platine und die Funktionalität des Mikrocontrollers. In den nachfolgenden

Teilen wird es um die Bluetooth-, WLAN- und USB-Modalitäten gehen, und ferner wird die App vorgestellt, die das Android-I/O-Board steuert. Abschließend werden die Möglichkeiten als Datenlogger betrachtet, und es wird gezeigt, wie sich die Firmware anpassen lässt und mit dem Bootloader in den Mikrocontroller geladen wird.

Bevor wir die Hardware aus der Nähe betrachten, hier zuerst eine Kurzübersicht über die Eigenschaften: Auf dem Android-I/O-Board sind 22 I/O-Leitungen verfügbar. Außerdem befinden sich auf der Platine ein gedruckter kapazitiver Sensor-Taster, vier LEDs und ein NTC. Damit bietet sich das Board auch zum Experimentieren und Lernen an. In **Tabelle 1** sind die Eigenschaften sowie die Optionen für die Betriebsspannung übersichtlich zusammengefasst.

Die Hardware

Bild 1 zeigt den Schaltplan des Android-I/O-Boards. Der erste Eindruck beim Betrachten täuscht: Für die drahtlose

Kommunikation mit dem Android-Gerät wird nur ein einziges Modul verbaut. Der Vollständigkeit halber wurden jedoch die drei wahlweise montierbaren Modultypen gemeinsam in das Bild aufgenommen. Auf das Board passen die gängigsten Bluetooth-Module wie das Elektor BL600 e-BoB (mit Bluetooth V4, Elektor 140270-91), ein Breakout-Board mit dem verbreiteten Modul HC-06 (mit Bluetooth V2), ein Platinensteckverbinder-Board mit dem HC-06, das RN41 von Microchip sowie ein Modul in Zigbee-Form mit dem HC-06. Unter den WLAN-Modulen passt sowohl das RN171-XV (Zigbee-Form) als auch das ESP8266 (Platinensteckverbinder). Ferner sind Steckverbinder auf der Platine montierbar, die zum Anschließen des Elektor Android Break-out-Boards (130516-91) und des Elektor FT232R USB/Serial Bridge/BoBs (110553-91) vorgesehen sind. Beide Typen stellen über ein USB-OTG-Kabel eine USB-Verbindung mit einem PC, Tablet oder Smartphone her.

Alle genannten Module übertragen die vom Android-Gerät oder PC kommen-

Anzeige



HAMMOND MANUFACTURING

Gehäuse für Raspberry Pi, Arduino und viele andere Einplatinencomputer

- Gehäuse
- Plattform

+ 44 1256 812812
sales@hammondmfg.eu



Sie haben die Wahl.

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



/1593HAM.htm

/1593HAMEGG.htm

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 Oberruberg

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

Wollen Sie sich auch ein Jahr lang in unserer Zeitschrift präsentieren? Bitte kontaktieren Sie dann Frau Grotenrath (julia.grotenrath@eimworld.com)

Stückliste

Widerstände:

R1,R2,R3,R4 = 470 Ω , 1 %, SMD 0805
 R5,R7,R8,R10,R14,R15 = 330 Ω , 1 %, SMD 0805
 R6,R9,R16 = 10 k, 1 %, SMD 0805
 R11 = NTC 10 k, SMD 0805 (z. B. Vishay NTCS0805E3103JHT)
 R12,R13 = 2k2, 1 %, SMD 0805

Kondensatoren:

C1,C2 = 100 n/25 V, 10 %, SMD 0805
 C3 = 4 μ F/25 V, 10 %, SMD 0805
 C4 = 1 μ F/25 V, 10 %, SMD 0805

Halbleiter:

LED1...LED4 = LED grün, Low-current, SMD 0805
 IC1 = PIC16F1938-I/SO, SOIC28 (programmiert, EPS 150057-41)
 IC2 = 24FC256-I/SN, SOIC8
 IC3 = MCP1702T-3302E/CB, SOT23A

Außerdem:

K1 = USB 2.0 Micro-Steckbuchse Typ B, SMD
 K2 = Stiftkontaktleiste 2 · 13 Pins, Raster 2,54 mm
 K3 = Stiftkontaktleiste 2 · 3 Pins, Raster 2,54 mm
 K4 = Stiftkontaktleiste 2 Pins, Raster 2,54 mm
 K5 = Stiftkontaktleiste 6 Pins, Raster 2,54 mm
 JP2 = Jumperleiste 2 · 4 Pins, Raster 2,54 mm
 S2 = Umschalter, Raster 2,54 mm
 Platine 150057-1
 oder
 Platine mit montierten SMDs: 150057-91

Verbindungsmodule

(nur ein Modul wird montiert!):

Mod1 = HC-06/-05 Bluetooth Break-out-Modul
Wird auf die Platine gelötet
 Mod2 = HC-06/-05 Bluetooth ZigBee Shaped Modul
 Mod2 = Bluetooth-Modul RN41XVC
 Mod2 = WLAN-Modul RN171 (mit Drahtantenne)
Montage auf zwei Buchsenleisten 10 Pins, Raster 2 mm
 Mod3 = Android Break-out-Board (Elektor 130516-91)
Montage auf Buchsenleiste 10 Pins, Raster 2,5 mm
 Mod4 = WLAN-Modul ESP8266 (Anschluss 2 · 4 Pins)
Montage auf Buchsenleiste 2 · 3 Pins, Raster 2,5 mm
 Mod5 = FT232R USB/Serial Bridge BoB (Elektor 110553-91)
Montage auf Buchsenleiste 3 Pins, Raster 2,5 mm (Stiftleiste 3 Pins auf BoB)
 Mod6 = HC-06 Modul (6-Pin-Version)
Montage auf Buchsenleiste 6 Pins, Raster 2,5 mm
 Mod7 = BL600 E-BoB (Elektor 140270-91)
Montage auf zwei Buchsenleisten 8 Pins, Raster 2,5 mm

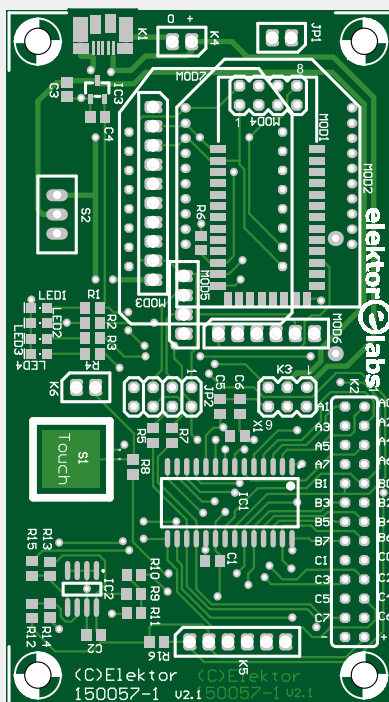


Bild 2. Sieben Möglichkeiten gibt es, auf der Platine des Android I/O-Boards ein Kommunikationsmodul zu montieren.

den Daten in seriellem Format. Die Daten werden dem Mikrocontroller PIC16F1938 übergeben, der insgesamt 24 I/O-Leitungen bereitstellt. Die Firmware des Mikrocontrollers, die später näher betrachtet wird, interpretiert die Kommandos für das Konfigurieren und Steuern der I/O-Leitungen, oder sie liest analoge oder digitale Signale ein.

Die I/O-Leitungen des Mikrocontrollers sind über eine zweireihige Stiftkontaktleiste (Pfostenverbinder) zugänglich. Zwei Leitungen werden von der seriellen Datenübertragung zum Modul belegt. Die Schaltung im Detail: Die Betriebsspannung gelangt zum Spannungsregler IC3 entweder über die Micro-USB-Buchse K1 oder zwei Lötstützpunkte, auf die eine zweipolige Stiftkontaktleiste gelötet werden kann (K4). Der Spannungsregler MCP1702-33 ist ein Low-drop-Typ für die Spannung 3,3 V, dies ist die Betriebsspannung des Bluetooth- oder WiFi-Moduls sowie der übrigen Schaltung. Mit Schalter S2 kann die Betriebsspannung abgeschaltet werden. Falls der Schalter entbehrlich ist, werden die zugehörigen Platinenlötunkte miteinander verbunden. Der Mikrocontroller (IC1) ist ein PIC16F1938 von Microchip im 28-poligen SOT-Gehäuse. Die Verbindung zwischen dem Mikrocontroller und dem Bluetooth-, WiFi- oder USB-Modul verläuft über die Jumper-Steckleiste JP2, die zum Debuggen benötigt wird, sowie über die Schutzwiderstände R5 und R7. Sämtliche I/O-Leitungen sind auf Steckverbinder K2 geführt.

Zum Speichern von Daten, beispielsweise als Datenlogger, befindet sich ein 32 KB großer, serieller I²C-Flash-Speicher (IC2) auf dem Board. Die I²C-Leitungen des Datenspeichers sind mit den Schutzwiderständen R14 und R15 sowie den Pull-up-Widerständen R12 und R13 beschaltet. Ferner ist ein NTC (R11) für Temperaturmessungen vorhanden, er bildet mit Widerstand R9 einen Spannungsteiler. Die LEDs LED1...LED4 zeigen diverse Signale an. LED1 leuchtet, wenn die Betriebsspannung eingeschaltet ist, LED2 ist nur in Betrieb, wenn das HC-06-Breakout-Modul montiert ist. LED3 und LED4 können vom Anwender genutzt werden, wobei LED3 auch die Aktivierung des Bootloaders signalisiert. Taster S1 ist der kapazitive Sensor in Gestalt einer kupfernen Insel auf der Platine. Parallel zu S1 liegen die Kontakte von Steckverbinder K6, er dient zum Aktivieren des Bootloaders.

Tabelle 2. Funktionen der Steckverbinder und Jumper.

K1	Micro-USB-Steckbuchse für die Stromversorgung
K2	Stiftleiste 2·13 Pins mit 24 I/O-Leitungen des PICs, 3,3 V und Masse
K3	Stiftleiste 2·3 Pins zum Zusammenschalten mehrerer I/O-Boards
K4	Stiftleiste 2 Pins für die unstabilisierte Stromversorgung (4...13 V)
K5	Stiftleiste 6 Pins zum Programmieren des Mikrocontrollers mit dem PICKit 2 oder 3
K6	Stiftleiste 2 Pins zum Starten des Bootloaders von Hand oder über Verbindungsleitung
JP1	Stiftleiste 2 Pins mit Jumper zum Anpassen der Firmware des Moduls ESP8266
JP2	Stiftleiste 2·4 Pins mit Jumpern zum Konfigurieren der Signal- und Stromversorgungsleitungen der übrigen Module
MOD1	Steckverbinder zum wahlweisen Anschließen des Bluetooth-, WLAN- oder USB-Moduls
...	
MOD7	(siehe Stückliste)

Auf dem Board sind für die Taktung des Mikrocontrollers auch ein Quarz (X1) mit den zugehörigen Kondensatoren (C5, C6) montierbar. Der in diesem Beitrag beschriebene Bootloader unterstützt diese Variante nicht, hierzu ist ein alternativer Bootlader erforderlich, bei dem das Konfigurationswort angepasst ist. In diesem Fall können die Leitungen A6 und A7 des Steckverbinders K2 nicht mehr von der Firmware benutzt werden.

Universelle Platine

Der Aufdruck auf der Platine (**Bild 2**) erscheint wegen der universellen Modulanschlüsse auf den ersten Blick reichlich unübersichtlich, doch bei näherem Hinsehen wird schnell deutlich, wo sich die Anschlüsse für das Modul der Wahl (MOD1...MOD6) befinden. Es sei noch einmal darauf hingewiesen, dass nur ein Modul zu montieren ist, die übrigen Anschlüsse bleiben unbenutzt. Mit Ausnahme der Steckverbinder und des Schalters S2 haben alle Bauteile SMD-Bauform, was die Montage von Hand nicht gerade einfach gestaltet. Deshalb ist im Elektor-Shop eine Platine erhältlich (Elektor 150057-91, siehe [1]), auf der alle SMD-Bauteile einschließlich Mikrocontroller und die Steckverbinder bereits montiert sind. Natürlich ist auch die leere Platine für die Bestückung in eigener Regie lieferbar, doch da die Module erst im nächsten Teil des Beitrags besprochen werden, ist diese Arbeit nicht eilig.

Auf der Platine befinden sich diverse Steckverbinder, Anschlüsse und Jumper, die unterschiedliche Funktionen haben. In **Tabelle 2** sind sie zusammen mit einer kurzen Beschreibung aufgelistet.

Jumperleiste JP2 schaltet die Signale Rx und Tx zwischen dem Mikrocontroller und dem montierten Modul durch. Normalerweise sind Pin 3 mit Pin 4 und Pin 5 mit Pin 6 verbunden. Die Pins 7 und 8 (GND) haben Verbindung über die Platine, was vorteilhaft ist, wenn ein externes Modul an JP2 angeschlossen wird. Auf Pin 2 liegt die Betriebsspannung 3,3 V vom Spannungsregler, während Pin 1 mit dem Betriebsspannungsanschluss des Moduls verbunden ist. Für die Stromversorgung des Moduls gibt es zwei Konfigurationen: Das Modul bezieht die Betriebsspannung 3,3 V von Spannungsregler IC3. Dies trifft für die Bluetooth- und WLAN-Module zu. Die Pins 1 und 2 müssen miteinander verbunden werden.

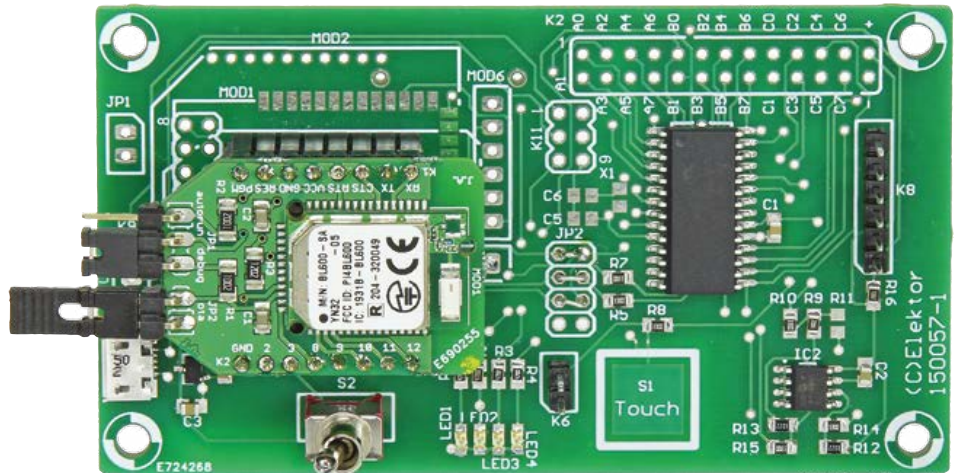


Bild 3. Ansicht der aufgebauten Platine, montiert ist hier das BL600 e-BoB von Elektor.

Tabelle 3. Optionale Funktionen der I/O-Leitungen.

Leitung	Eingang – hohe Impedanz (0)	Eingang – interner Pullup (1)	Ausgang (2)	ADC (3)	Cap Sense (4)	PWM (5)	Sonderfunktion (6)
A0	X		X	X			
A1	X		X	X			
A2	X		X	X			
A3	X		X	X			
A4	X		X		X		
A5	X		X	X	X		
A6	X1		X			X	
A7	X1		X			X	PWM-Brücke mit A6
B0	X	X	X	X	X	X	
B1	X	X	X	X	X	X	PWM-Brücke mit B0
B2	X	X	X	X	X		
B3	X	X	X	X	X		
B4	X1	X1	X	X	X		
B5	X1	X1	X	X	X		
B6	X	X	X			X	
B7	X	X	X			X	PWM-Brücke mit B6
C0	X1		X				Counter
C1	X1		X			X	
C2	X1		X			X	PWM-Brücke mit C1
C3	X1						I2C: SCL
C4	X						I2C: SDA
C5	X		X				
C6	TX (zum Bluetooth/WLAN/USB-Modul)						
C7	RX (vom Bluetooth/WLAN/USB-Modul)						
S							Datenlogger
X							Uhr
Y							Version/Seriennummer
Z							Globale Einstellungen

¹ Bitänderungen können automatisch übernommen werden

Bei Einsatz des FT232R USB/Serial Bridge BoBs (Elektor 110553-91) oder des Android Break-out-Boards (Elektor 130516-91) werden diese Boards über den USB-Anschluss mit der Betriebsspannung 5 V versorgt. Die Pins 1 und 2 dürfen auf keinen Fall miteinander verbunden werden!

Kommandos

Zwar ist der Betrieb des Android I/O-Boards in dieser Projektphase noch nicht möglich, trotzdem sollen hier die Steuerkommandos kurz gestreift werden.

Wenn die Verbindung mit dem Bluetooth-, WLAN- oder USB-Modul hergestellt ist, besteht eine transparente serielle Datenverbindung zwischen dem Android-Gerät und dem Mikrocontroller auf dem Android I/O-Board. Eine App kann dann Kommandos zum Mikrocontroller senden. Die Firmware des Mikrocontrollers interpretiert die Kommandos und steuert die I/O-Leitungen. Die I/O-Leitungen sind in Anlehnung an die Leitungen des Mikrocontrollers bezeichnet: A0...A7, B0...B7 und C0...C7 (siehe **Tabelle 3**). Die Kommandos sind in vier Gruppen unterteilt: **W** für *Write*, eine Leitung wird auf einen Wert gelegt. Format: *W Leitung Wert*. Beispiel: *w a3 1* legt I/O-Leitung A3 auf hohe Spannung.

R für *Read*, der Wert einer Leitung wird gelesen. Format: *R Leitung Wert*. Beispiel: *r b4* liest den Wert von I/O-Leitung B4. Der Mikrocontroller antwortet im Format *R Leitung Wert*, in diesem Fall mit *R B4 1*.

S für *Settings*, hiermit wird die Funktion einer Leitung festgelegt. Format: *S Leitung Funktion*, wobei *Funktion* ein Zahlenwert ist. Eine 0 steht für einen hochohmigen Eingang, eine 1 ergibt einen Eingang mit internem Pullup und bei 2 wird die Leitung als Ausgang konfiguriert. Zum ADC-Eingang wird die Leitung bei 3, mit 4 wird sie zum Kapazitätssensor-Eingang und bei 5 zum PWM-Ausgang. Die 6 ist für eine spezielle Funktion reserviert. Beispiel: Das Kommando *s b3 3* schaltet Leitung B3 als ADC-Eingang.

Den Kommandos können Parameter angefügt werden, dies geschieht wie folgt: Zu jeder Leitung gehören drei 8-bit Statusregister, die durch 1, 2 oder 3 hinter dem Namen der Leitung aufgerufen werden. Zum Beispiel bedeutet *s b32 8*, dass Statusregister 2 von Leitung B3 auf 8 gesetzt werden soll. Statusregister 0 ist die Leitung selbst, *s b3 3* ist folglich identisch mit *s b30 3*.

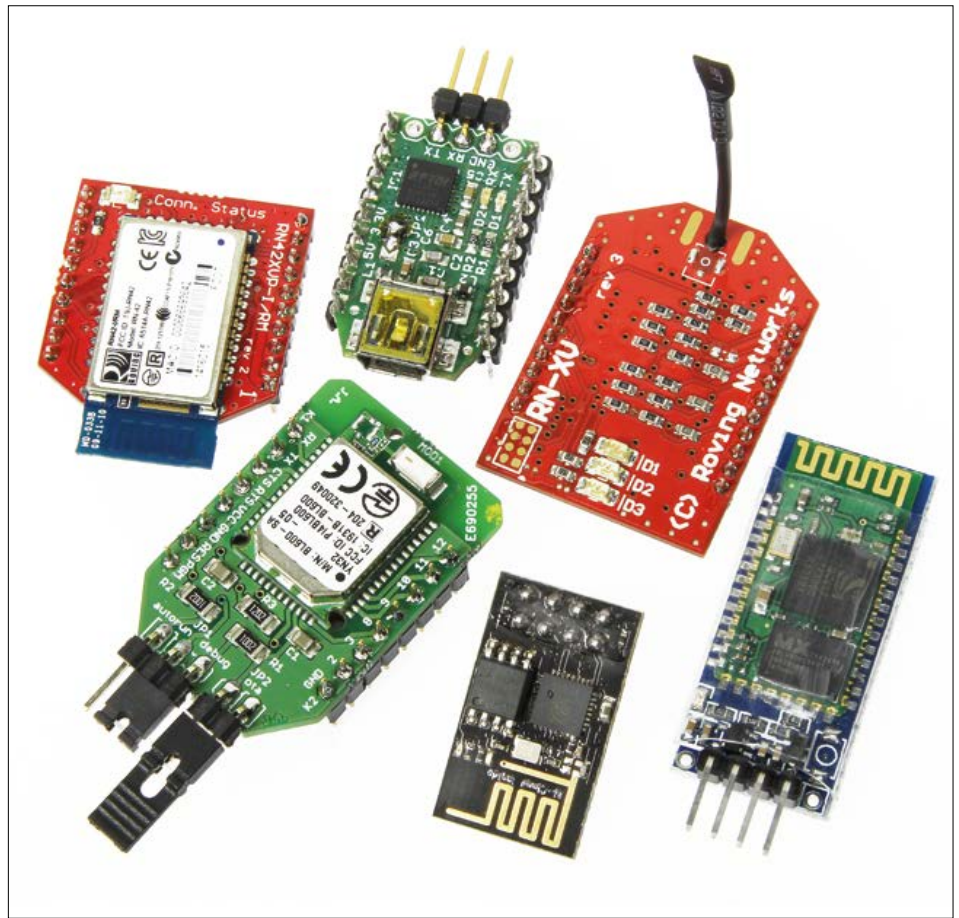


Bild 4. Auswahl einiger Kommunikationsmodule, die auf der Platine montiert werden können.

G für *Get*, damit wird eine Leitung gelesen. Format: *G Leitung*. Der Mikrocontroller antwortet mit *G Leitung Wert*. Beispiel: *g b3* ergibt *G B30 3*.

Die Kommandos und Leitungsamen unterscheiden nicht zwischen Groß- und Kleinschreibung. Beim Debuggen, wobei der Mikrocontroller die Eingaben echot, erweist sich die Kleinschreibung der Kommandos als vorteilhaft. Damit ist sofort ersichtlich, ob Texte vom Android-Gerät (kleine Buchstaben) oder vom Android I/O-Board (große Buchstaben) herrühren. In dem ergänzenden Dokument 150057-W werden sämtliche Funktionen

ausführlich erklärt. Das Dokument kann frei von der Projektseite [1] heruntergeladen werden.

Im zweiten Teil des Beitrags werden wir die Bluetooth-, WLAN- und USB-Module besprechen sowie eine App betrachten, die das Android I/O-Board zum Belichten und Ätzen von Platinen einsetzt. Wir werden grundlegende Classes besprechen, über die das Android I/O-Board von eigenen Apps gesteuert werden kann.

(150057)gd

Weblinks

[1] www.elektormagazine.de/150057



Webinar zum Android I/O-board

Am 17. September wird der Autor dieses Projekts ein Webinar zu dem von ihm entwickelten Board präsentieren. Nicht verpassen! Die Teilnahme ist gratis, Sie müssen sich lediglich registrieren unter:

www.elektor.de/webinar.

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

eSTORE[®]
Beta LAYOUT

Entwickeln, Bestücken, Löten

PCB-Design & Reflow-Workshop
Von der Schaltplanidee zur fertigen Elektronikbaugruppe
8. – 9. Oktober 2015 | 65527 Niedernhausen
www.pcb-pool.com/workshop

**Sommerangebot
Reflow-Controller V2**

€ 109,00*



Angebot gültig bis 30.9.15, solange
Vorrat reicht.

Lotpaste (bleifrei)

€ 21,40*



**ERSA Lötstation
i-CON NANO**

€ 198,00*



**ESD-Arbeitsplatz-
Set**

€ 58,00*



Big Beta-Reflow-Kit

€ 129,00*



* inkl. MwSt. und zzgl. Versandkosten

www.beta-eSTORE.com

Beta
LAYOUT
create electronics



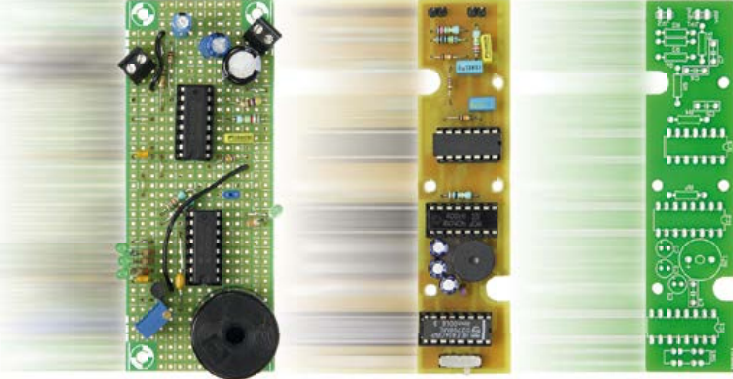
elektor • PCB Service

Platinen – Prototypen – Multilayer – Kleinserien

➡ Kostengünstig!

➡ Top-Qualität!

➡ Zuverlässig!



LEARN
DESIGN
SHARE

Powered by



- **PCB proto:** Ideal für Privatleute, die schnell und günstig maximal 2 Leiterplatten nach vordefinierten Spezifikationen benötigen.
- **STANDARD pool:** Diese Option ist für Firmen konzipiert, die ihre Kleinserie nach den am häufigsten verwendeten Spezifikationen produzieren lassen wollen.
- **RF pool:** Wenn Ihre Entwicklung sehr anspruchsvolle Spezifikationen erfordert, ist 100-µm-Technologie die beste Wahl.
- **IMS pool:** Bei dieser Option werden Aluminiumkern-Leiterplatten verwendet, um eine hohe Wärmeabfuhr zu gewährleisten.



Bestellen Sie Ihre Platinen jetzt unter:

www.elektorPCBservice.de

Batterie-Ersatz mit USB

Ersetzt die AAA-Zelle

Von **Danny Winkler** (D)

Alle Batterien sind sich in einem Punkt gleich: Irgendwann sind sie leer. Auch mit Akkus ist das Problem nicht ganz vom Tisch, denn die muss man immer wieder aufladen. Da ist es unbequem, wenn man ein Gerät dauerhaft stationär einsetzen will, das



für mobilen Betrieb gedacht ist und über keine Buchse zum Anschluss eines Netzteils verfügt. Doch wo Elektronik ist, ist auch eine Lösung nicht weit: Eine Pseudo-Batterie mit USB-Anschluss gibt dauerhaft Strom.

Mehrfach schon hat sich Danny Winkler über seinen kleinen MP3-Player/Recorder geärgert: Immer wenn er damit gerade einen interessanten Song aufnehmen wollte, war die Batterie leer. Ein Anschluss für ein Netzteil war nicht vorgesehen. Das ist Pech. Aber wie das Leben so spielt, wollte er eines schönen Tages doch einmal mit der Zeit und damit den bedroh-

teten Bauteilen fremdgehen. Mit diesen verflüchtigen kleinen SMDs müsste man sich einmal auseinandersetzen. Und just an dieser Stelle fiel ihm das Problem mit dem portablen MP3-Player wieder ein. „Fein!“ dachte er sich, denn so konnte er zwei Fliegen mit einem Lötcolben erschlagen. Er machte sich ans Werk und entwickelte eine kleine Schaltung, die eine Batterie-

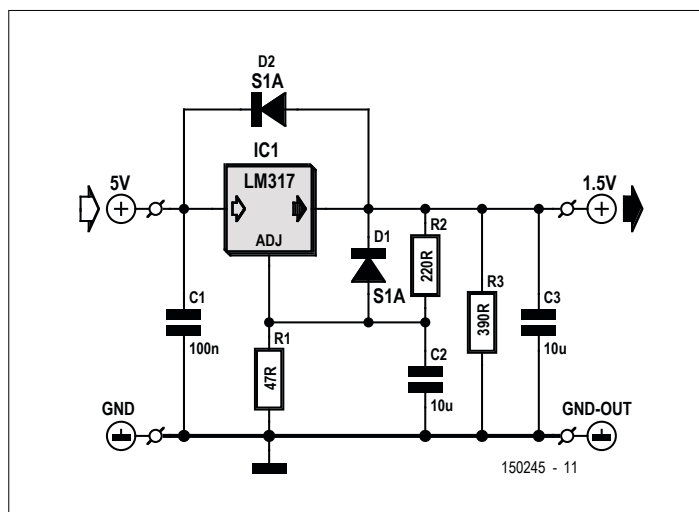


Bild 1. Schaltung des elektronischen Batterie-Ersatzes.

Stückliste

Widerstände:

(Standard: 5 %, 100 mW, SMD 0805)

R1 = 47 Ω

R2 = 220 Ω

R3 = 390 Ω

Kondensatoren:

(Standard: 10 %, Vielschicht, SMD 0805)

C1 = 100 n

C2, C3 = 10 μ

Halbleiter:

D1, D2 = S1A, DO-214AC oder SMB

IC1 = LM317, SOT-223

Außerdem:

Platine 150245-1 v1.0

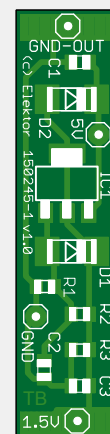


Bild 2. Der Bestückungsaufdruck der Platine im AAA-Format.

rie vom Typ AAA ersetzen und den Strom von einem Netzteil beziehen sollte. Mit SMDs, so war ihm sofort klar, musste es möglich sein, die Platine so klein zu bekommen, dass man sie als Ersatz anstelle einer Batterie in das Batteriefach stecken konnte. Und der Erfolg gab ihm recht.

AAA-Ersatzschaltung

Der ersten Überlegung, dass die Platine so geformt sein sollte, dass sie genau eine AAA-Batterie ersetzt, folgte die zweite. Ein preiswertes USB-Ladegerät für Smartphone & Co. sollte sich sehr gut zur Versorgung der Schaltung eignen. So ein kleines Netzteil kostet nicht viel und liefert 5,2 V mit einer Belastbarkeit von mindestens 1 A. Nötig wäre also nur noch ein Spannungsregler, der die 5,2 V auf primärzellen-typische 1,5 V bringt. Et voilà: Die Schaltung von

Bild 1 hat sich quasi von selbst gezeichnet!

Da Danny schon viele gute Erfahrungen mit der „großen Version“ des einstellbaren Spannungsreglers LM317 gemacht hatte, wollte er es dieses Mal mit der Miniaturversion im SOT223-Gehäuse probieren. Mit R1 und R2 wird die Ausgangsspannung nach der Formel $U_{out} = 1,25 V \times (1 + R1/R2)$ auf 1,5 V eingestellt. Ein paar Kondensatoren sorgen für stabilen Betrieb und die beiden Dioden schützen vor unangemessenen Spannungen an den Pins des Spannungsreglers. Das war schon alles.

Pseudo-Batterie

Nun blieb nur noch das Design einer Platine, die das Ersetzen einer AAA-Batterie möglich macht. **Bild 2** zeigt den Bestückungsaufdruck. Sowohl das Platinenlayout als auch die CAD-Daten kann man von der Elektor-Website [2] herunterladen. Fertig bestückt sieht der Prototyp so aus wie in **Bild 3** und **Bild 4**. Auf der Platine ist noch genug Luft und die SMD-Bauteile sind mit dem Format 0805 noch weit weg von einem staubkorn-großen Aggregatzustand, bei dem man nicht mehr Atmen darf, bevor das Teil mit Lötzinn festgepinnt ist. Es dürfen sich also wie Danny selbst gerne SMD-Verweigerer und Hardcore-Fans von Bauteilen mit Anschlussdrähten an dieses Projekt herantrauen. Man muss hierfür auch nicht zwingend SMD-Lötpaste und ähnliche Utensilien besorgen. Diese Platine kann man auch mit normalem Lötzinn bearbeiten. Lediglich eine Pinzette ist zu empfehlen. An die Enden der Platine lötet man kleine Stückchen dünnes Kupferblech, welche die Kontaktflächen der Batterie simulieren (**Bild 5**).

Nach der Bestückung und Kontrolle der Lötstellen sowie der Polung der Dioden sollte man die Ausgangsspannung messen, bevor man einen Test am (noch) lebenden portablen Gerät riskiert. Nutzt man als Netzteil eines dieser preiswerten USB-Ladegeräte, dann kann man schlicht den nicht benötigten Stecker am entsprechenden Ende des USB-Kabels abzwicken und die rote und schwarze Leitung passend mit dem 5-V- und Masse-Pin auf der Platine verbinden.

(150245)

Weblinks

[1] Datenblatt LM317: www.ti.com/lit/ds/symlink/lm117.pdf

[2] www.elektormagazine.de/150245

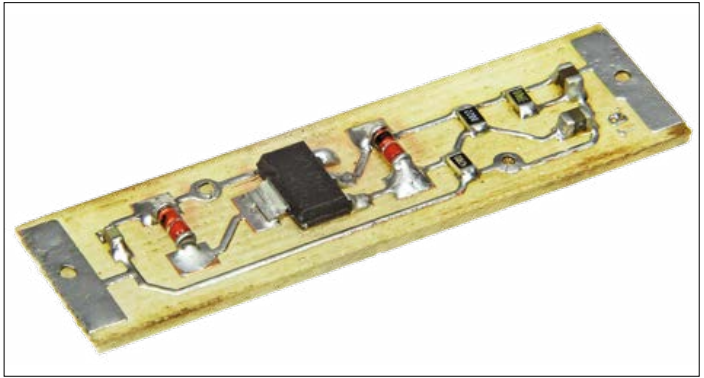


Bild 3. Der fertig bestückte Prototyp.

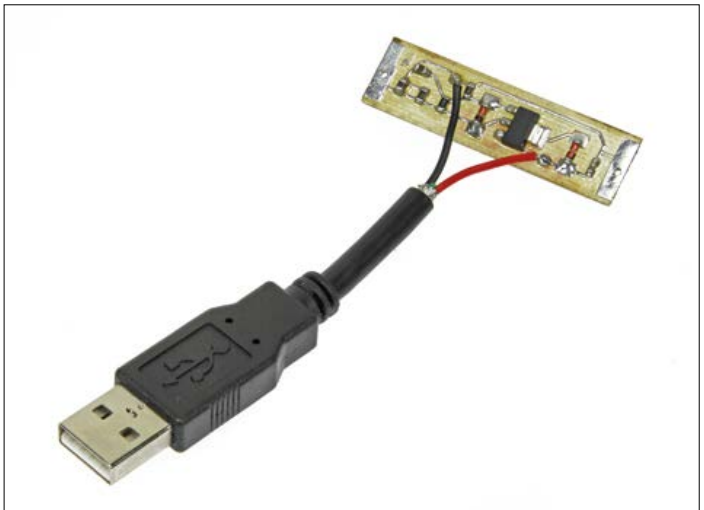


Bild 4. Prototyp mit angelötetem USB-Stecker.

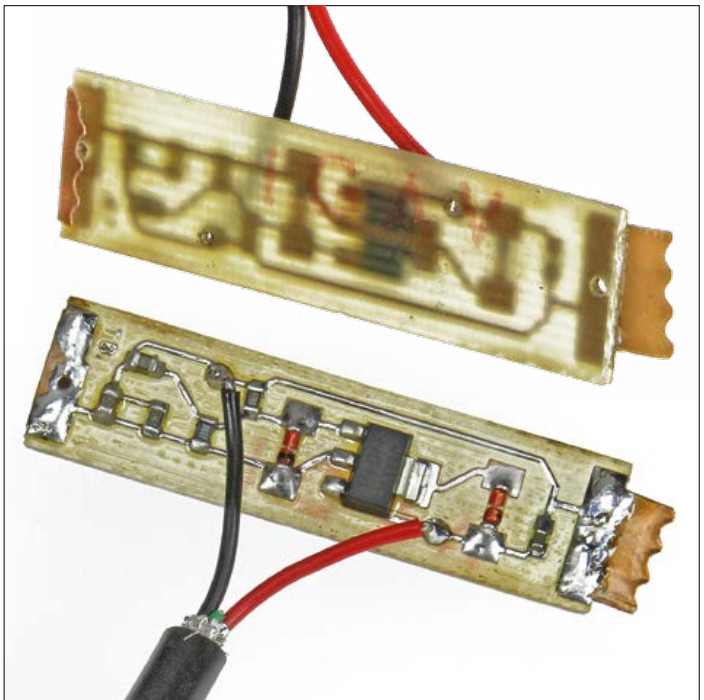
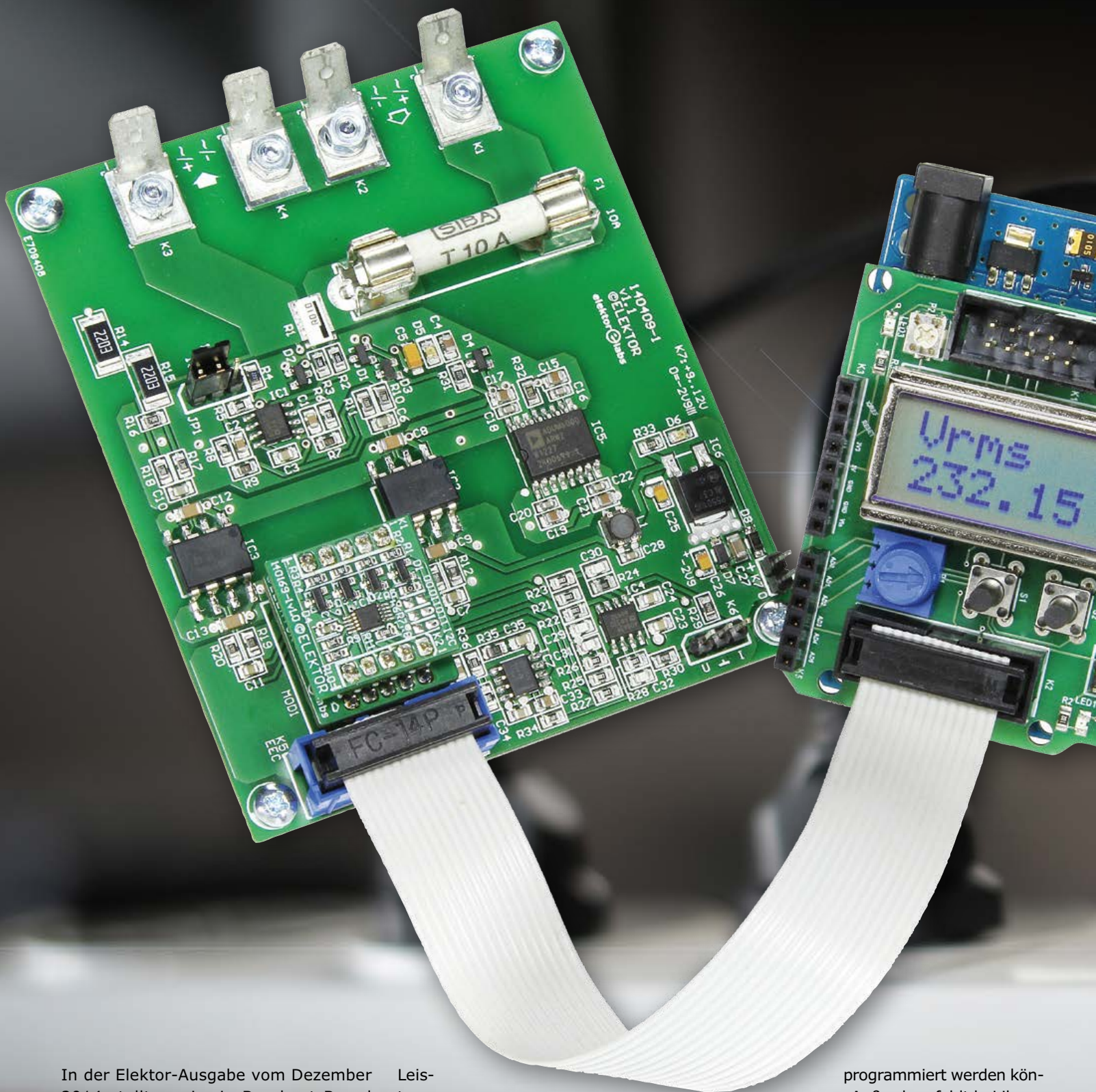


Bild 5. Zwei angelötete Stückchen Kupferblech stellen den Kontakt im Batteriefach sicher.



In der Elektor-Ausgabe vom Dezember 2014 stellten wir ein Breakout-Board (BoB) mit dem ADS1115 vor [1], einem 16-bit-ADC mit vier Kanälen und I^2C . Der ADS1115 war vorher bereits beim ELF-Empfänger [2] und beim 16-bit ADC-Modul [3] zum Einsatz gekommen. Da dieser Chip im winzigen MSOP-Gehäuse nun als anwenderfreundliches BoB-Modul verfügbar ist, haben wir mit ihm einen

Leistungsmesser entworfen. Die niedrige Abtastfrequenz des ADS1115, die bei 860 Hz ihre Grenze hat, beschränkt den Einsatz auf niedrigfrequente Messsysteme. Leistungsmesser mit ähnlichen Eigenschaften sind zwar preisgünstig auf dem Markt, doch dort handelt es sich um geschlossene Systeme, die nicht modifiziert oder

programmiert werden können. Außerdem fehlt bei ihnen ein Ausgang, an den ein Multimeter oder ein Oszilloskop anschließbar ist. Unser hier vorgestellter Leistungsmesser besteht aus einer Messplatine, darauf haben das BoB mit dem ADS1115 und eine zugehörige Filterplatine ihren Platz. Die Messplatine übernimmt gleichzeitig die sichere galvanische Trennung

AC/DC- Leistungsmesser

Hohe Genauigkeit, weiter Messbereich

Hardware: **Ton Giesberts** (Elektor-Labor) Software: **Luc Lemmens** (Elektor-Labor)

Wenn Messungen an Geräten durchgeführt werden, die mit dem Stromnetz in Verbindung stehen, ist die galvanische Trennung des Messsystems ein Gebot der Sicherheit. Dieser Leistungsmesser, der auch Spannungen und Ströme misst, erfüllt die Vorgaben der Schutzklasse II. Ein Arduino Uno und das bekannte Extension-Shield von Elektor sind für die Ausgabe der Messergebnisse auf einem LC-Display zuständig. Auch ein Multimeter oder ein Oszilloskop sind anschließbar.

zwischen den zu messenden Größen auf der einen Seite und den Messausgängen auf der anderen. Die Signale auf der sekundären Seite werden vom ADS1115 digitalisiert und anschließend von einem Arduino Uno mit Elektor Extension-Shield verarbeitet und auf das Display gesetzt. Dort werden bei Wechselspannung die Effektivwerte von Leistung, Spannung und Strom sowie die Scheinleistung

und der Leistungsfaktor angezeigt. Bei Gleichspannung erscheinen naturgemäß nur Leistung, Spannung und Strom in der Anzeige.

Galvanische Trennung

Leistungsmesser sind meistens im Einsatz, um Messungen an Geräten durchzuführen, die an Netzspannung betrieben werden. Hier erfordert die elektri-

sche Sicherheit, dass der Stromkreis der Messgröße von dem Teil galvanisch getrennt ist, der die Messdaten ausgibt oder weiterverarbeitet. Häufig wird dies mit diversen Optokopplern und Opamps realisiert, der Aufwand an Schaltungskomponenten ist hoch. Eine Alternative sind so genannte Isoliervverstärker, zum Beispiel der Typ AMC1100 von Texas Instruments. Bei diesem *Fully-differential Iso-*

Eigenschaften

- Drei Messbereiche: 10 A, 1,77 A und 0,177 A
- Eingangsspannung max. 500 V_{AC}/300 V_{DC}
- Niedrigste messbare Leistung: 0,1 W
- Getrennte, isolierte Messgänge für Spannung und Strom
- Isolation zwischen primärer und sekundärer Seite nach Schutzklasse II
- Verarbeitung der Messwerte mit Arduino Uno
- Anzeige der Messwerte auf zweizeiligem LC-Display
- Anzeige von Spannung, Strom, Effektivleistung, Scheinleistung und Leistungsfaktor bei Wechselspannung
- Anzeige von Spannung, Strom und Leistung bei Gleichspannung

lation Amplifier beträgt die Mindestbandbreite 60 kHz, die Isolationsspannung beträgt 4250 V_{Spitze}, was hier der maximalen Arbeitsspannung 1200 V entspricht. Mit diesem Opamp und einem überlegten Platinenlayout ist die Schutzisolation der Klasse II erzielbar, so dass das System

elektrisch sicher ist. Der Isolierverstärker befindet sich in einem SMD-Gehäuse *Gullwing-8*, es ist einem DIP-8-Gehäuse mit nach außen gebogenen Pins ähnlich. Ein Messsystem mit diesem Baustein benötigt für die Eingangsseite und die Ausgangsseite zwei voneinander unab-

hängige Stromversorgungen. Standardisierte Versorgungsmodule, die unstabilisierte Spannungen liefern, sind für diesen Zweck brauchbar. Allerdings sind solche Module meistens nicht für Schutzklasse II ausgelegt, so dass weitere Komponenten, beispielsweise ein separater Spannungsregler benötigt werden. Eine vorteilhaftere Lösung ist der Einsatz eines Bausteins, der eigens für diesen Zweck entwickelt wurde. Unsere Wahl fiel auf den ADuM6000 von Analog Devices, ein isolierender DC/DC-Wandler mit der Ausgangsspannung 5 V oder 3,3 V. Die Spitzenisolationsspannung V_{IORM} beträgt 846 V_p.

Schaltung

Ein Strom wird gemessen, indem der Spannungsabfall an einem Widerstand gemessen wird, der im Stromkreis liegt.

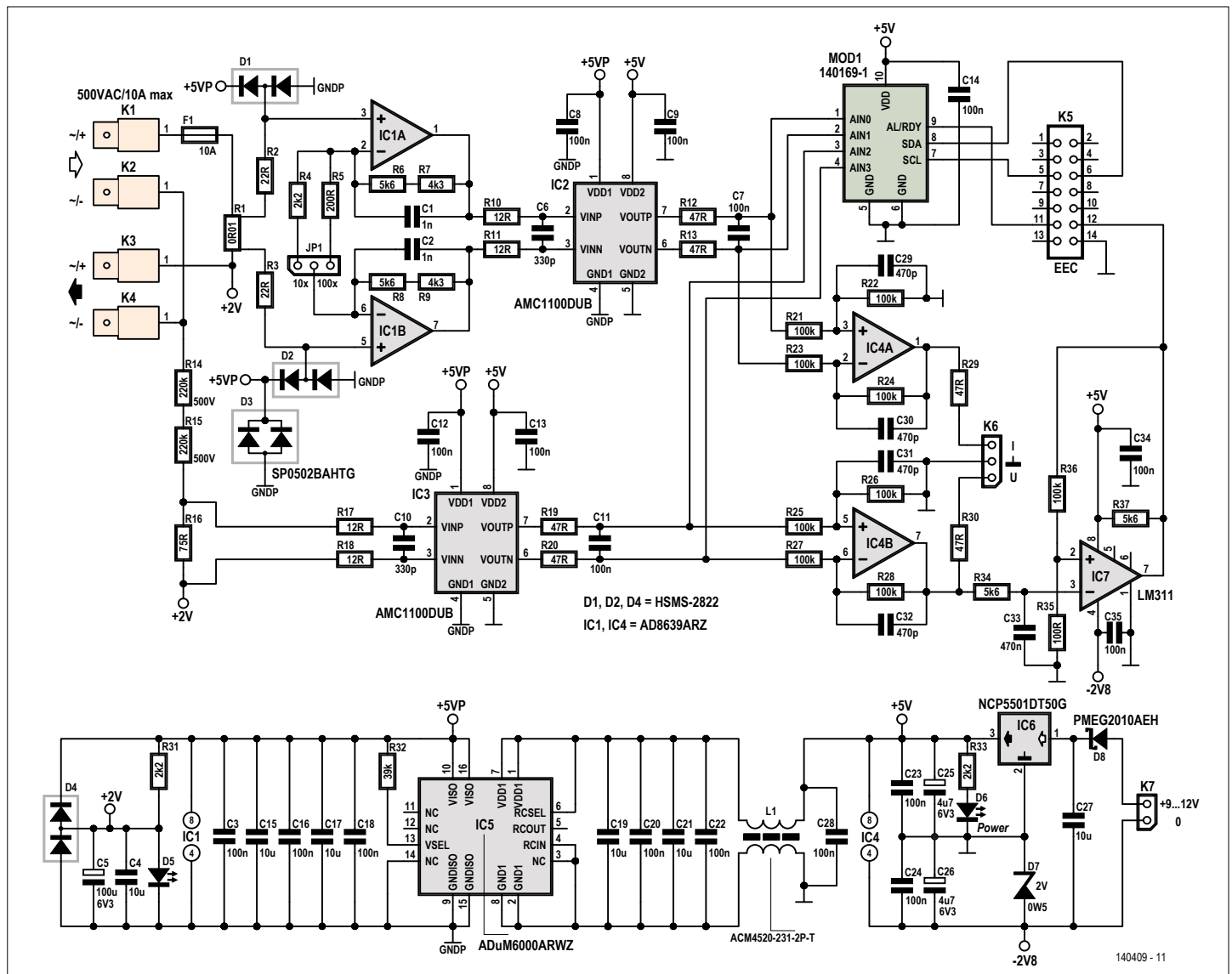


Bild 1. Der Leistungsmesser arbeitet mit zwei AMC1100 für das Messen von Spannung und Strom.

Praktisch jedes Multimeter arbeitet nach dieser simplen Methode. In der Schaltung des Leistungsmessers (**Bild 1**) ist R1 der im Stromkreis liegende Widerstand. Außerdem liegt im Stromkreis die Sicherung F1, sie trennt ihn bei Überschreiten des Nennstroms 10 A. Die Eingangsklemmen K1 und K2 sind mit der Spannungsquelle verbunden, K3 und K4 führen zur Last.

Die Eingangs- und Ausgangsklemmen sind als Flachstecker ausgeführt, sie werden mit der Platine verschraubt. Die Flachstecker gewährleisten zuverlässige mechanische und elektrische Verbindungen zu Leitungen mit dem erforderlichen Querschnitt.

Damit der Stromkreis möglichst wenig beeinflusst wird, muss der Messwiderstand einen niedrigen Wert haben. Andererseits muss der Spannungsabfall genügend hoch sein, um den niedrigsten zu messenden Strom messen zu können. Wenn notwendig, kann ein Opamp die Messspannung verstärken. In der Schaltung übernimmt diese Aufgabe die Stufe mit IC1A und IC1B, die Verstärkung ist einstellbar. IC1A/B ist ein AD8639, ein zweifacher *Rail-to-rail Output Operational Amplifier*. Die typische Eingangs-offsetspannung beträgt 3 μV (maximal 9 μV), die maximale Offsetdriftspannung liegt bei nur 0,06 μV . Mit diesen Werten ist der AD8639 für die vorgesehene Anwendung wie maßgeschneidert. Ferner beträgt das Verstärkungs-Bandbreiten-Produkt 1,35 MHz, so dass die Bandbreite bei dieser Anwendung nicht unter 10 kHz liegt, was mehr als ausreicht.

Der AD8639 ist als Instrumentenverstärker konfiguriert, wobei die differentielle Verstärkung $1 + 2 \cdot (R6 + R7)/RG$ ist (mit $RG = R4$ oder $R5$). Ohne Jumper JP1 ist die Verstärkung gleich 1, sie ist das Verhältnis der Spannung zwischen den Ausgängen von IC1A und IC1B zu der Spannung an Messwiderstand R1. Mit Jumper JP1 ist eine Verstärkung von 10 oder 100 wählbar. Die wählbaren Verstärkungen ergeben gemeinsam mit der Verstärkung von IC2 drei Strombereiche: 17,7 A (Verstärkung 1, durch F1 auf 10 A begrenzt), 1,77 A (Verstärkung 10) und 0,177 A (Verstärkung 100) bei sinusförmigen Strömen. Die maximalen RMS-Werte bei Strömen mit hohen Scheitelfaktoren liegen niedriger. Abhängig vom Rauschen bewegt sich die niedrigste messbare Leis-

tung unter 0,1 W (Messwert ADC auf -50 dB voll ausgesteuert).

Damit die Spannung an Messwiderstand R1 nicht aus dem Gleichtaktbereich von IC1A/B gerät, liegt R1 einseitig an +2 V. Die Spannung +2 V fällt auf der isolierten Seite von IC5 an LED D5 ab. Gleichzeitig ist D5 auf der isolierten Seite die Betriebsspannungsanzeige.

Die Eingangsspannung des nachfolgenden Isolierverstärkers AMC1100 (IC2) darf differentiell bis 250 mV betragen. Die Verstärkung beträgt 8, so dass die Ausgangsspannung 2 V erreichen kann. Der AMC1100 hat einen differentiellen Eingang mit geschalteter Kapazität, das Ausgangsrauschen ist pauschal mit 3,1 mV_{rms} angegeben. Gilt dieser Wert über die gesamte Bandbreite, dann liegt der Rauschpegel niedriger, wenn die Bandbreite bereits vor dem ADC begrenzt wird. Aus diesem Grund haben wir eine separate Filterplatine mit vier Sektionen entworfen, sie hat ihren Platz vor dem ADC (siehe Kasten „Filtermodul“). Wenn der ADC unmittelbar mit den Ausgängen von IC2 verbunden wäre, würde das Signal-Rausch-Verhältnis 645 betragen. Durch Begrenzen der Eingangsbandbreite des ADC steigt dieser Wert um den Faktor $\sqrt{60}$, er beträgt folglich rund 5000. Das bedeutet, dass niedrigere am Messwiderstand abfallende Spannungen gemessen werden können.

Die Spannung, die an den Eingangsklemmen K1 und K2 liegt, wird von einem zweiten AMC1100 (IC3) gemessen. Der Spannungsteiler R14...R16 setzt die Spannung ungefähr um den Faktor 5900 herab, so dass bei der Netzspannung 230 V an den Eingängen von IC3 etwa 39 mV liegen. R16 ist einseitig mit der Referenzspannung +2 V verbunden. Die Ausgänge von IC2 und IC3 führen zu den vier Eingängen des Breakout-Boards (BoB) mit dem ADS1115.

Der Leistungsmesser kann auch als isolierende Brücke dienen, wenn hohe Spannungen mit dem Multimeter oder Oszilloskop gemessen werden sollen. Die differentiellen Ausgänge von IC2 und IC3 sind dafür wenig geeignet, ein asymmetrischer Ausgang ist für diesen Zweck vorteilhafter. Deshalb setzt der Zweifach-Opamp IC4 (ebenfalls ein AD8639) die symmetrischen Ausgangssignale der beiden Isolierverstärker in asymmetrische Äquivalente gleicher Höhe um. Der Gleichtaktoffset (typisch 2,55 V bei $V_{DD} = 5\text{ V}$), der an den

Ausgängen des AMC1100 liegt, wird auf fast 0 V, jedoch nicht exakt 0 V reduziert. Abhängig von der Differenz der Gleichspannungen an den getrennten Ausgängen des AMC1100 kann an den Ausgängen von IC4 eine Offsetspannung von einigen Millivolt ($< 12\text{ mV}$) auftreten.

Der Gleichtakt-Eingangsspannungsbereich des AD8639 verläuft von -0,1 V bis 3 V bei der asymmetrischen Betriebsspannung +5 V. Damit die Ausgangsspannungen der beiden AMC1100 in diesem Bereich verarbeitet werden können, setzen die Spannungsteiler R21/R22 und R25/R26 die Gleichspannungen 2,55 V an den Ausgängen beider AMC1100 auf 1,275 V herab. Die maximale differentielle Ausgangsspannung, die am AMC1100 gemessen werden konnte, lag bei ungefähr 2,5 V (1,25 V Spitze). Das bedeutet, dass die maximale Spannung an den Eingängen von IC4 den Wert 1,9 V nicht übersteigen kann (1,275 V_{DC} plus halbe Spitzenwechselspannung 0,625 V). Die Spannung liegt mit einigem Abstand im Gleichtakt-Eingangsspannungsbereich des AD8639.

Zum Erkennen der Nulldurchgänge ist Komparator IC7, ein LM311, hinzugefügt. Der LM311 besitzt einen Open-Kollektor-Ausgang. Wenn er an einer quasi-symmetrischen Betriebsspannung arbeitet (+5 V/-2,8 V), sind die Nulldurchgänge ohne viel Aufwand detektierbar. R37 ist der Pullup-Widerstand des Ausgangs, R35 und R36 bewirken eine Mitkopplung, so dass die Nulldurchgänge zuverlässig erkannt werden. Der Komparator-Ausgang liegt an Pin 12 von K5. Mit optimaler Präzision werden die Nulldurchgänge bei negativen Impulsen erkannt. Der Ausgang von IC7 ist dann noch auf niedriger Spannung, eine Hysteresis tritt nicht auf. Der nicht invertierende Komparator-Eingang liegt dann noch auf Masse-Niveau.

Zur Weiterleitung der Informationen an den Arduino ist ein EEC-Steckverbinder (K5) vorgesehen. Über diesen Steckverbinder gelangen die I²C-Signale des A/D-Wandlers sowie einige weitere Signale zu K2 des Arduino Extension-Shields, das auf einen Arduino Uno montiert ist. Die Betriebsspannung +5 V der sekundären Seite kommt von Low-drop-Spannungsregler IC6. Die meisten ICs der Schaltung arbeiten mit dieser Spannung. IC4 benötigt jedoch auch eine negative Betriebsspannung, damit seine Aus-

gangswechselspannung frei von einem DC-Offset ist, und außerdem sollen auch Gleichspannungen gemessen werden. Deshalb muss an K6 auch eine negative Spannung verfügbar sein. Die negative Spannung wird erzeugt, indem Zenerdiode D7 (2 V) in der Masse-Leitung des Low-drop-Spannungsreglers IC6 eine virtuelle Masse bildet.

Die isolierte Betriebsspannung auf der Eingangsseite der beiden AMC1100 sowie des Eingangsverstärkers IC1 wird von

IC5, einem ADuM6000, bereitgestellt. Der interne Spannungsregler ist auf +5 V eingestellt, indem die Anschlüsse V_{SEL} und V_{ISO} miteinander verbunden sind. Gleichtaktdrossel L1, die mögliche Störsignale dämpft, ist zwar mit 3 A überdimensioniert, sie weist jedoch einen niedrigen Reihenwiderstand von $2 \cdot 50 \text{ m}\Omega$ auf.

Messen mit Arduino

Da die maximale Abtastfrequenz des A/D-Wandlers im freilaufenden Betrieb

nur 860 Hz beträgt und in diesem Baustein nur ein einziger physischer A/D-Wandler für vier Eingänge integriert ist, können Strom und Spannung mit maximaler Abtastfrequenz nicht gleichzeitig gemessen werden. Der A/D-Wandler müsste über den I²C-Bus fortlaufend umgeschaltet werden, so dass zwischen beiden Messungen unzulässige Verzögerungen auftreten würden (ausführlicher geht der Beitrag „Nur noch einige Zeilen Software...“ auf dieses Thema ein, der in

Stückliste Messplatine

Widerstände:

R1 = 0 Ω 201, 2 W, 1 % (Ohmite FC4L64R010FER)
 R2,R3 = 22 Ω , 0,25 W, 5 %, SMD 0805
 R4,R31,R33 = 2k20, 0,125 W, 1 %, SMD 0805
 R5 = 200 Ω , 0,125 W, 1 %, SMD 0805
 R6,R8,R34,R37 = 5k6, 0,125 W, 1 %, SMD 0805
 R7,R9 = 4k3, 0,125 W, 1 %, SMD 0805
 R10,R11,R17,R18 = 12 Ω , 0,125 W, 1 %, SMD 0805
 R12,R13,R19,R20,R29,R30 = 47 Ω , 0,125 W, 1 %, SMD 0805
 R14,R15 = 220 k, 1,5 W, 1 %, 500 V, SMD 2512
 R16 = 75 Ω , 0,125 W, 1 %, SMD 0805
 R21...R28,R36 = 100 k, 0,125 W, 1 %, SMD 0805
 R32 = 39 k, 0,125 W, 5 %, SMD 0805
 R35 = 100 Ω , 0,125 W, 1 %, SMD 0805

Kondensatoren:

C1,C2 = 1 n/50 V, 5 %, SMD 0805, COG/NP0
 C3,C7,C11,C14,C23,C24,C34,C35 = 100 n/50 V, 10 %, SMD 0805, X7R
 C4,C15,C17,C19,C21 = 10 μ /10 V, 10 %, SMD 0805, X7R
 C5 = 100 μ /6,3 V, 20 %, SMD Case A (1206), Tantal
 C6,C10 = 330 p/50 V, 5 %, SMD 0805, COG/NP0
 C8,C9,C12,C13 = 100 n/50 V, 10 %, SMD 1206, X7R
 C16,C18,C20,C22,C28 = 100 n/25 V, 10 %, SMD 0603, X7R
 C25,C26 = 4 μ 7/6,3 V, 10 %, SMD Case R (0805), Tantal
 C27 = 10 μ /25 V, +80/-20 %, SMD 1206, Y5V
 C29...C32 = 470 p/50 V, 5 %, SMD 0805, COG/NP0
 C33 = 470 n/16 V, 10 %, SMD 0805, X7R

Induktivität:

L1 = ACM4520-231-2P-T, SMD, Gleichtaktdrossel 3 A, $2 \cdot 50 \text{ m}\Omega$, 230 Ω @100 MHz

Halbleiter:

D1,D2,D4 = HSMS-2822-TR1G, SMD SOT-23
 D3 = SP0502BAHTG, SMD SOT-23
 D5,D6 = LED grün, SMD 0805
 D7 = BZT52C2V0-7-F, SMD SOD-123 (Zenerdiode 2 V/0,5 W)
 D8 = PMEG2010AEH, 20 V/1 A, SMD SOD-123F
 IC1,IC4 = AD8639ARZ, SMD SOIC-8
 IC2,IC3 = AMC1100DUB, SMD Gullwing-8 (SOP-8)
 IC5 = ADuM6000ARWZ, SMD RW-16 (SOIC_W-16)
 IC6 = NCP5501DT50G, SMD DPAK3
 IC7 = LM311D, SMD SOIC-8

Außerdem:

K1...K4 = Faston-Stecker 6,35 mm, für Schraubbefestigung 3,3 mm
 K5 = 2x7-Stiftleiste, Raster 2,54 mm
 K6,JP1 = 1x3-Stiftleiste, Raster 2,54 mm
 K7 = 1x2-Stiftleiste, Raster 2,54 mm
 MOD1 = 2x5-SIL-Fassung, Raster 2,54 mm
 F1 = Sicherungshalter für Platinenmontage, 32 A/600 V (6,3 $\cdot$ 32 mm)
 F1 = Sicherung 10 A träge, 500 V_{AC}/300 V_{DC}, 6,3 $\cdot$ 32 mm
 Modul 140169-91 (ADS1115-BOB)
 Filterplatine 140169-2
 Platine 140409-1

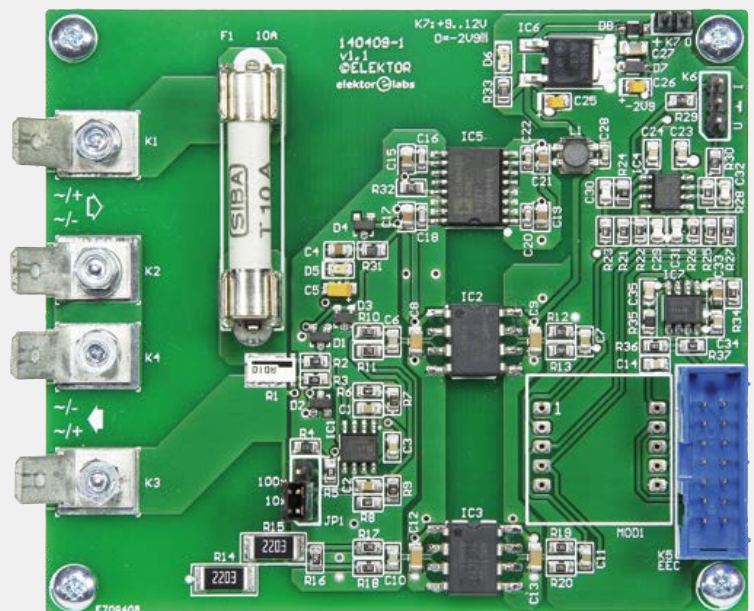
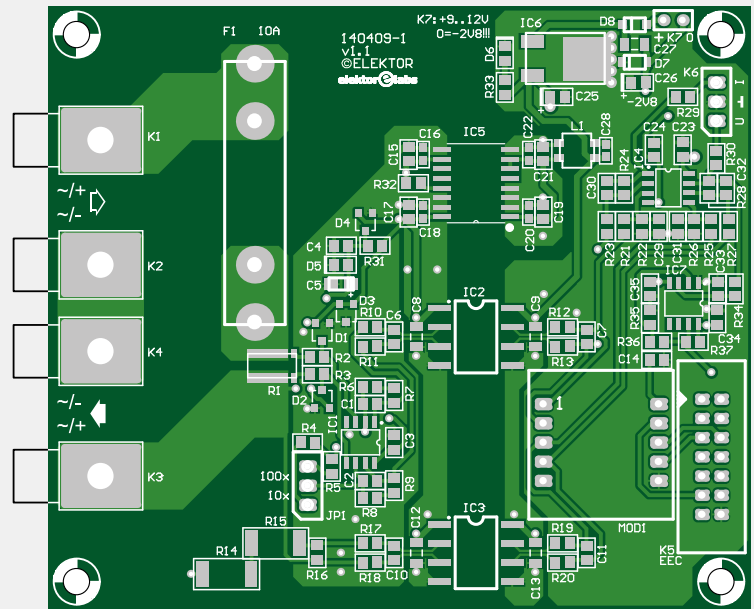


Bild 2. Auf der Messplatine befindet sich unter den drei ICs eine breite, kupferfreie Zone, sie trennt galvanisch die primäre von der sekundären Seite.

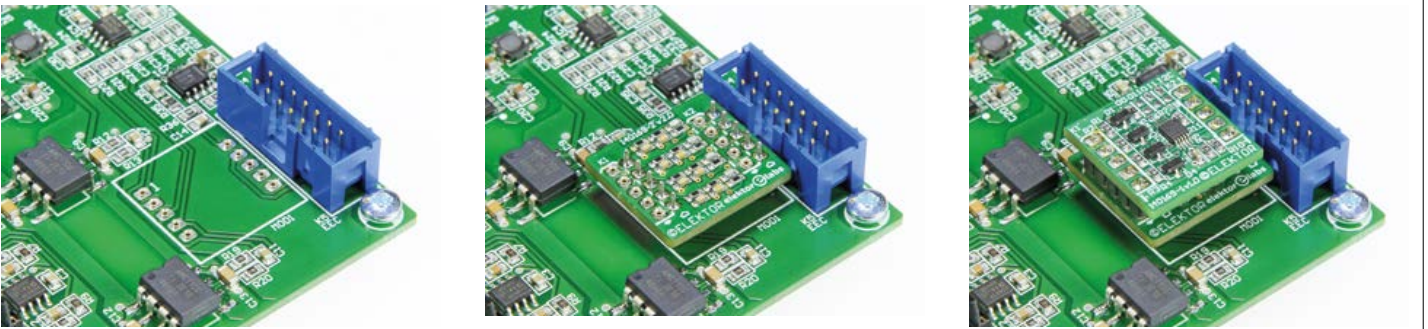


Bild 3. Die Platine mit dem A/D-Wandler und die Filterplatine werden übereinander auf den Steckverbinder MOD1 der Messplatine gesteckt.

der Juli/August-Ausgabe 2015 erschienen ist.) Deshalb werden zuerst 172 Spannungsmessungen durchgeführt, gefolgt von ebenso vielen Strommessungen. Um die Phasenbeziehung zwischen Spannung und Strom zu definieren, werden beide Messreihen in einem Nulldurchgang der Spannung gestartet. Komparator IC7 ist mit Arduino-Pin 9 (Portpin B1) verbunden, der Arduino detektiert den Nulldurchgang. Fehlt dieses Synchronisationssignal, schaltet die Software auf DC-Messung um, sie kehrt selbsttätig zum AC-Betrieb zurück, sobald das Synchronisationssignal wieder vorhanden ist.

Die Software für diesen Leistungsmesser, ein „Sketch“ für den Arduino Uno, kann frei von der Projektseite [4] heruntergeladen werden. Folgende Größen werden von der Software gemessen beziehungsweise berechnet (siehe auch [5]):

Für Wechselspannung:

$$P_{VA} = \frac{\sum_{i=1}^N V[i] \times I[i]}{N}$$

$$V_{RMS} = \sqrt{\left(\sum_{i=1}^N V^2[i] \right) / N}$$

$$I_{RMS} = \sqrt{\left(\sum_{i=1}^N I^2[i] \right) / N}$$

$$P_{RMS} = V_{RMS} \times I_{RMS}$$

$$P\text{-factor} = \frac{P_{RMS}}{P_{VA}}$$

Für Gleichspannung:

$$V = \left(\sum_{i=1}^N V[i] \right) / N$$

$$I = \left(\sum_{i=1}^N I[i] \right) / N$$

$$P = V \times I$$

Die Messwerte werden, wie **Bild 4** zeigt, zyklisch auf dem LC-Display ausgegeben.

Aufbau

Die Messplatine (140409-1) ist relativ großflächig gestaltet, sie ist in **Bild 2** dargestellt. Bestückt ist die Platine fast nur mit SMDs, so dass der Aufbau für Ungeübte zwar schwierig, aber nicht unmöglich ist. Zuerst werden die SMDs montiert, dann folgen die Kontaktleisten und der Sicherungshalter. Die Flachstecker werden mit Gewindeschrauben, Zahnscheiben und Muttern möglichst fest mit der Platine verschraubt. Das Breakout-Board 140169-1 mit dem ADS1115 ist bestückt und getestet im Elektor-Store erhältlich. Die zugehörige Filterplatine 140169-2 muss in eigener Regie aufgebaut werden, dort befinden sich SMDs auf beiden Seiten. Auf der Oberseite sind zwei 5-polige Buchsenleisten für die Verbindung mit dem ADS-BoB montiert, die Unterseite ist der Platz für zwei 5-polige Stiftleisten zur Befestigung auf der Messplatine. Die beiden Platinen werden aufeinander gesteckt (Filterplatine unten) und anschließend in den mit MOD1 bezeichneten Steckverbinder eingesetzt. Die Verbindung zwischen dem Extension-Shield sowie der Messplatine stellt ein 14-adriges Flachkabel her,

an dem Flachkabel-Steckverbinder montiert sind.

Achtung! Die Messplatine muss absolut berührsicher in ein schutzisolierendes Gehäuse eingebaut werden!

Auf einer Gehäuseseite können die Flachstecker nach außen ragen, oder sicherer, die Flachstecker werden mit Bananenchsen oder Euro-Einbausteckern verbunden. Auf der gegenüber liegenden Gehäuseseite wird eine Aussparung zum Durchführen des Flachkabels angebracht. Hier können auch Messbuchsen montiert werden, die mit den Messausgängen von K6 verbunden sind, sowie eine Stromversorgungsbuchse, verbunden mit K7. Auf dem Extension-Shield sind zwei zusätzliche Leitungen erforderlich, damit alle benötigten Signale über den dort vorhandenen EEC-Steckverbinder laufen. Zu legen sind isolierte Leitungen von K2-11 nach K4-1 und von K2-12 nach K4-2.

Die Stromversorgung des Leistungsmessers kann ein Steckernetzteil übernehmen, das 9...12 V Gleichspannung liefert und mit mindestens 200 mA belastbar ist. Der Arduino Uno wird von seinem eigenen Netzteil mit Strom versorgt.

Einstellungen

Die Verstärkung und der Offset der analogen Eingangsstufe auf der Messplatine sind für Spannungs- und Strommessungen über die Software kalibrierbar. Die eingestellten Werte werden im internen EEPROM des Arduino Uno abgelegt, so dass die Kalibrierung im Prinzip nur einmal vorzunehmen ist.

Nach dem Reset prüft die Software, ob bereits ein Wert im zugehörigen Speicherplatz steht (ein Wert ungleich 255). Wenn

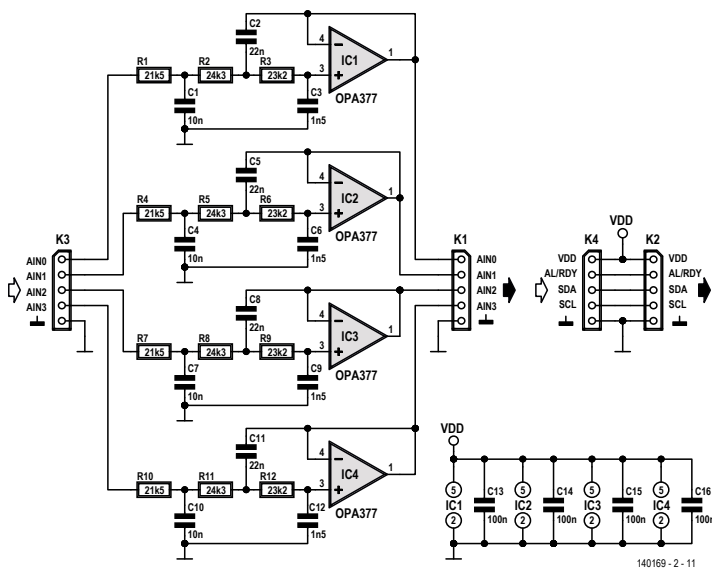
Filtermodul



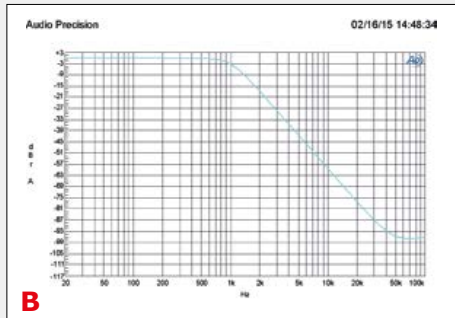
Wie im Haupttext beschrieben, lässt sich die Empfindlichkeit des Leistungsmessers erhöhen, indem die Bandbreite am Eingang des ADC-Moduls mit einem Filter begrenzt wird. Zu diesem Zweck wurde ein Filtermodul entworfen, das aus vier aktiven Filtern dritter Ordnung besteht. Die Filter sind so dimensioniert, dass die Grenzfrequenz 1 kHz beträgt. Dieser Wert erscheint hoch, doch er wurde gewählt, um Phasendrehungen der Messsignale bei 50 Hz oder 60 Hz sowie der ersten Harmonischen möglichst gering zu halten.

Bei den Filtern (**Bild A**) handelt es sich um eine Sallen-und-Key-Konfiguration mit Butterworth-Charakteristik dritter Ordnung (siehe Frequenzgang in **Bild B**). Die Opamps, die zum Einsatz kommen, sind vom Typ OPA377. Dies ist ein Rail-to-rail-Opamp mit CMOS-Eingängen und exzellenten Rauscheigenschaften. Der Eingangsvorstrom beträgt typisch nur 0,2 pA, so dass die Filterwiderstände keinen zusätzlichen Offset verursachen. Die Temperaturabhängigkeit von Eingangs-Offset und Offset-Drift beträgt typisch 0,25 $\mu\text{V}/^\circ\text{C}$ beziehungsweise 0,32 $\mu\text{V}/^\circ\text{C}$.

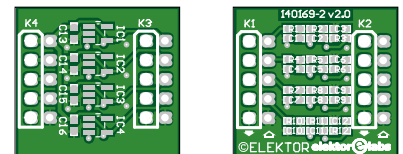
Die Platine ist so gestaltet, dass sie exakt unter das BoB mit dem ADS1115 passt. Um dies zu realisieren, sind beide Seiten mit SMDs bestückt (**Bild C**). Auf der Oberseite befinden sich zwei Buchsenleisten, beispielsweise SIL-IC-Fassungen, und auf der Unterseite zwei Stiftkontaktleisten mit runden, dünnen Stiften, so dass ein Zusammenstecken möglich ist. Wegen der zweiseitigen SMD-Bestückung dürfte der Aufbau nicht ganz einfach sein, doch dies ist der Preis für das Platinenformat.



A



B



C

Stückliste Filtermodul

Widerstände:

R1,R4,R7,R10 = 21k5, 1 %, 0,1 W, SMD 0603
 R2,R5,R8,R11 = 24k3, 1 %, 0,1 W, SMD 0603
 R3,R6,R9,R12 = 23k2, 1 %, 0,1 W, SMD 0603

Kondensatoren:

C1,C4,C7,C10 = 10 n/50 V, 5 %, COG/NPO oder X7R, SMD 0603
 C2,C5,C8,C11 = 22 n/50 V, 5 %, COG/NPO oder X7R, SMD 0603
 C3,C6,C9,C12 = 1n5/50 V, 5 %, COG/NPO oder X7R, SMD 0603
 C13,C14,C15,C16 = 100 n/50 V, 10 %, X7R, SMD 0603

Halbleiter:

IC1,IC2,IC3,IC4 = OPA377AIDBVT

Außerdem:

K1,K2 = 5x1-Buchsenleiste, Raster 2,54 mm
 K3,K4 = 5x1-Stiftleiste, Raster 2,54 mm
 Platine 140169-2

noch kein Wert gespeichert wurde, startet die Kalibrieroutine, anderenfalls startet die Messung. Die Kalibrierung kann später jederzeit wiederholt werden, wenn beim Einschalten des Leistungsmessers der Taster S1 gedrückt gehalten wird.

Während des Einstellvorgangs wird bei jedem Schritt zuerst die Art der Einstellung angezeigt, nach Drücken von Taster S1 kann die betreffende Einstellung vorgenommen werden.

Die erste Einstellgröße ist der Offset der Spannungsmessung (*0 V input*). Bei offenem Eingang K1/K2 wird mit den Tastern S1 und S2 auf dem Experimentier-Shield die Anzeige des LC-Displays auf 0 V eingestellt. Wenn die Taster fünf Sekunden lang nicht bedient werden, schaltet die Software zum Einstellen der Verstärkung bei Spannungsmessung weiter (*input V*). Dazu wird eine bekannte, hohe Gleichspannung (mindestens 50 V) zusammen mit einem präzisen Multimeter an den Eingang K1/K2 gelegt. Jetzt müssen die Taster S1 und S2 so bedient werden, dass die Anzeige auf dem LC-Display mit dem Wert auf dem Multimeter übereinstimmt.

Nach weiteren fünf Sekunden ohne Aktion folgt das Einstellen des Strom-Offsets (*0 A input*). Dabei bleibt der Eingang K1/K2 offen, und der Offset wird wie bei der Spannungsmessung mit den Tastern S1 und S2 auf 0 gebracht. Schließlich bleibt noch das Einstellen der Verstärkung bei Strommessung (*input I*). Dazu muss eine Gleichspannungsquelle mit dem Eingang K1/K2 verbunden werden, und an den Ausgang K3/K4 wird eine Last in Reihe mit einem präzisen Strommesser angeschaltet. Jetzt lässt sich die Verstärkung bei Strommessung in gleicher Weise wie die Verstärkung bei Spannungsmessung einstellen. ◀

(140409)gd

Weblinks

- [1] www.elektormagazine.de/140169
- [2] www.elektormagazine.de/140035
- [3] www.elektormagazine.de/130485
- [4] www.elektormagazine.de/140409
- [5] <http://openenergymonitor.org/emon/buildingblocks/ac-power-arduino-maths>

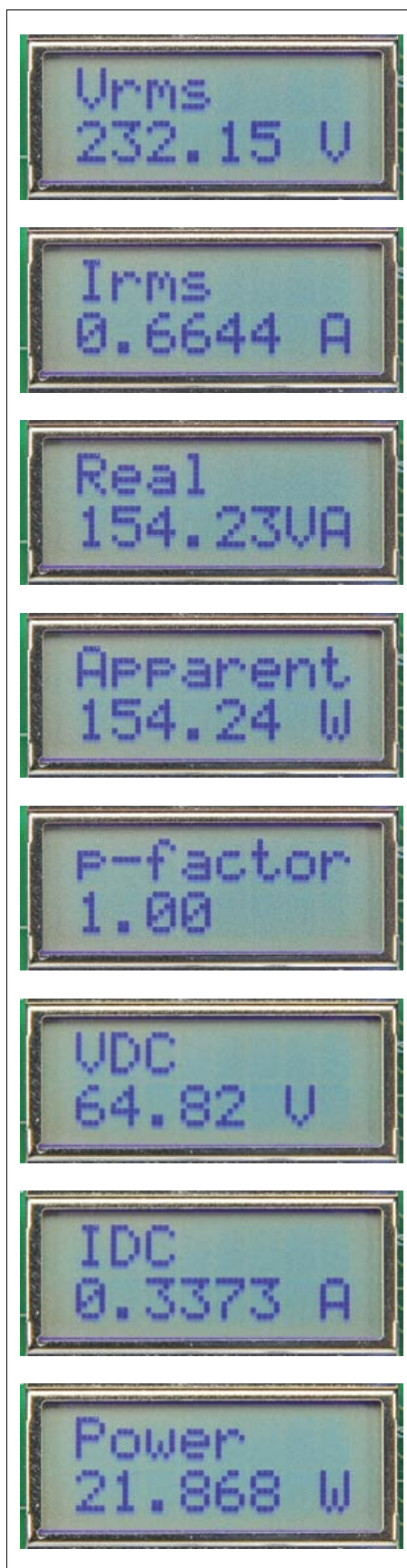


Bild 4. Die Messwerte erscheinen der Reihe nach auf dem Display. AC: Spannung, Strom, Wirkleistung, Scheinleistung, Leistungsfaktor. DC: Spannung, Strom, Leistung.

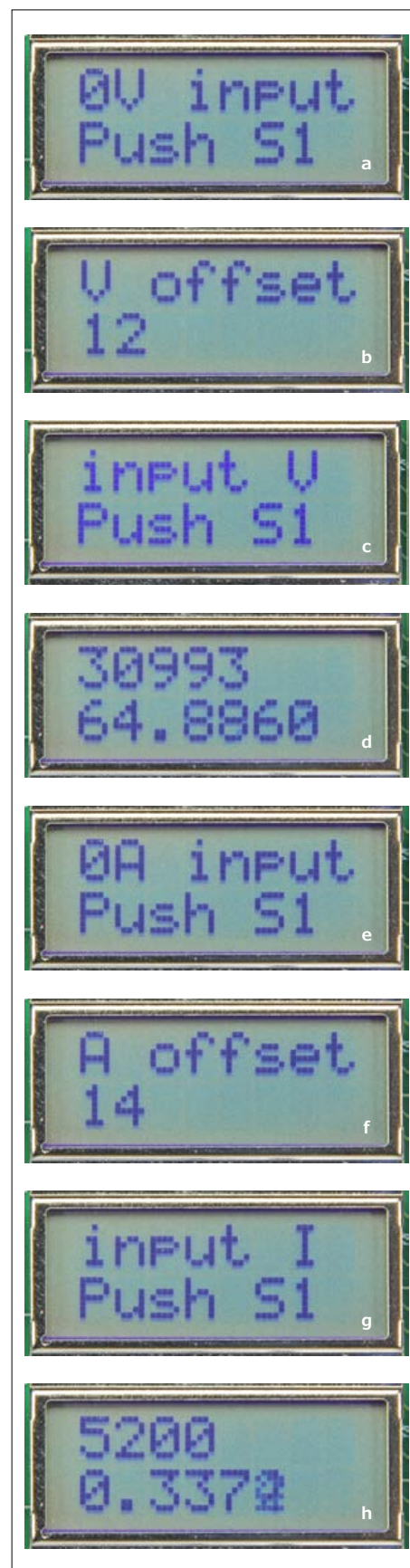
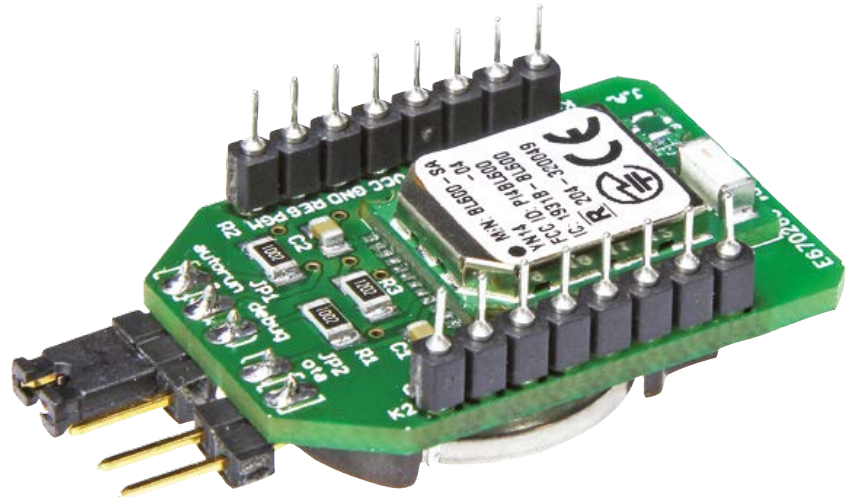


Bild 5. Das LC-Display bei der Kalibrierung: Offset für Spannungsmessung (a/b), Verstärkung für Spannungsmessung (c/d), Offset für Strommessung (e/f), Verstärkung für Strommessung (g/h).

e-BoB BL600

Teil 6

- Low Energy
- Events für Timer
- Datenübertragung und Verbindung



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

In den vorangegangenen Artikeln haben wir zusammen mit Ihnen die Programmiergrundlagen unseres e-BoB BL600 herausgearbeitet und die Inbetriebnahme gezeigt. Der sechste und letzte Teil dieser Serie beschäftigt sich mit der geringen Stromaufnahme (*Low Energy*), die einer der größten Vorteile des Moduls ist!

Low Energy, 5 μ A

Im Tiefschlaf-Modus beträgt die Stromaufnahme zwar nur 0,4 μ A, aber der e-BoB kann nur durch Reset oder Zustandsänderung an einem Eingang geweckt werden. Wir verwenden also den Überwachungsmodus mit ca. 5 μ A Stromaufnahme, so dass das Modul durch unser Programm geweckt werden kann, z.B. mit Hilfe der Timer.

Es wird nur eine einzelne LED eingesetzt, um dieses näher zu illustrieren. Falls Sie weitere Komponenten an Ihren BL600 anschließen, kann es nötig sein, das Programm anzupassen. Um die geringe Stromaufnahme ganz auffällig zu zeigen betreiben wir das Modul abwechselnd im Normal- und im Überwachungsmodus. Schließen Sie ein Amperemeter wie im Schaltbild gezeigt an: sobald die LED erlischt, fließen 5 μ A Strom.

Der Sourcecode des hier vorgestellten Testprogramms *bl600-6.sb* steht bei Elektor zum Download zur Verfügung [1]. Falls Sie Änderungen einbringen wollen, sehen Sie in den vorangegangenen Artikeln nach, die sämtliche Informationen hierzu bieten. Die einzelnen Schritte:

- **Bluetooth anhalten:** Bluetooth wird durch die Funktion *BleAvertStop* angehalten und mit *BleAdvertStart* wieder gestartet. Dieser Wechsel geschieht mittels der Timer (s. Absatz *Timer* für die Beschreibung des Codes in **Rot** und **Grün**).
- **LED aus:** Alles, was an das Modul angeschlossen wird,

zieht Strom. Also wird unsere LED durch die Funktion *GPIOWrite* in den Low-Zustand versetzt. Wir lassen sie leuchten, wenn Bluetooth gestartet wird und löschen sie wieder beim Anhalten (mit den Timer-Funktionen, s. unten).

- **UART angehalten im Modus NON DEBUG:** Der serielle Port des Moduls zieht ebenfalls Strom. Wir halten ihn an mit der Funktion *UartClose*, die Pins TX und RTS werden als Ausgänge mit *High*- bzw. *Low*-Pegel definiert. Zwei Punkte müssen berücksichtigt werden, um Fehler zu vermeiden:
 - Falls wir den seriellen Port beim Start des Programms (im Programm MAIN) schließen, ist der Puffer des Ports noch gefüllt. Um ihn zu leeren warten wir 1000 ms ab (willkürlicher Wert), bevor wir ihn schließen. Hierzu nutzen wir den Timer, Code in **Lila**.
 - Sobald der serielle Port einmal geschlossen ist, überträgt der Handler *HandlerLoop* die per Bluetooth empfangenen Daten an den seriellen Port und umgekehrt (mit der Funktion *BleVspUartBridge*). Wir kommentieren diese Funktion aus und erstellen unseren eigenen Code in **Orange** (s. *Events der Datenübertragung*).
- **Messen:** Der bei manchen Amperemetern festzustellende Innenwiderstand kann die Messung in Wach-Modus beeinflussen. Bei einem Wert von z.B. 100 Ω ergibt sich ein Strom von 10 mA (bei leuchtender LED), der Spannungs-

abfall an den Anschlüssen des Messgerätes beträgt 1 V und das Modul funktioniert nicht mehr... Verwenden Sie besser den Messbereich 2 mA als den Bereich 200 μ A.

Der Timer-Event

Der Timer, vorgestellt im Artikel über das Lauflicht [4], wird eingesetzt, um ein Wechselspiel von Stopp und Start von Bluetooth zu erzeugen. Dies codiert man so:

- Deklaration des Handlers: *OnEvent*
EVTMR1 Call FuncTimer1
- die Funktion: *Function FuncTimer1()*
- Timer-Auslösung: *TIMERSTART(1,4000,0)*

Um die geringe Stromaufnahme des Moduls mit einem Amperemeter anzeigen zu können, aktivieren wir Bluetooth (LED an) für 25 ms (Funktion *bleadvertstop*, Code in **Rot**) und halten dann Bluetooth für 4000 ms an (LED aus) (Funktion *bleadvertstart*, Code in **Grün**).

```
n = Bleadvertstart(0,Adrt$,25,4000,0)
```

Der Timer wird auch eingesetzt, um das Schließen des UART-Ports zu verzögern.

Die Events der Datenübertragung

Die über Bluetooth bzw. den seriellen Port empfangenen Daten werden durch einen Handler bearbeitet, genauso, wie wir es z.B. gemacht haben, um die Steuerdaten für die Dreifarb-LED von Ihrem Smartphone abzufangen [4]. Hier ergibt sich jedoch ein Problem, denn wir haben den UART-Port deaktiviert indem wir den Code *BleVspUartBridge* unterdrückt (auskommentiert) haben. Zu Kontrollzwecken wird der via Bluetooth empfangene Text um ein Zeichen versetzt wieder über Bluetooth zurückgesandt (ich komme später darauf zurück).

Das Verbindungs-Event

In den vorangegangenen Artikeln haben wir bereits das Event *EVBLEMSG* angezapft, um die Events zu Verbindungsauf- und -abbau behandeln zu können. Im Thermometer-Programm aus dem Artikel von Januar/Februar 2015 [1] war der erste Schritt nach Verbindungsaufbau die Übernahme des NTC-Wertes, der zweite der Versand an das Smartphone und der letzte bestand darin, das Modul wieder zu trennen. Nun leuchtet eine LED an Ausgang 2 des Moduls, solange es verbunden ist; sie erlischt bei Trennung der Verbindung.

Der Handler *EVBLEMSG* für die Bluetooth-Messages wird aus der Bibliothek *cli.manager.sblib* kopiert und in *MyHandlerBleMsg* umbenannt (Code in **Blau**). Vergessen Sie nicht, Port 2 im Hauptprogramm als Ausgang zu initialisieren.

Erstellung unseres Programms

Auch das Programm *upass.vsp.sb* wird als Grundlage für unser Programm Low Energy wiederverwendet, es wurde bereits in den anderen Artikeln für die UART-Dienste [1][4][6] genutzt. Es wird überarbeitet und dient zur Behandlung der vom Modul empfangenen Daten. Hierzu benennen wir es um in *BL600-6.sb* und ändern zwei Parameter in dieser ersten Version ab:

- **ENABLE_DEBUG_PRINTS: 0** (*Debug-Modus inaktiv*)

- **DEVICENAME: „JA_TEST“**
- Prüfen Sie die Funktion Ihres Programms indem Sie es zum e-BoB BL600 übertragen. Es wird automatisch gestartet, sonst geben Sie den Befehl *AT+RUN* ein [3]. Prüfen Sie nun die Kommunikation Ihres Programms mit dem Programm *SERIAL* aus dem *TOOLKIT* auf Ihrem Smartphone [2].

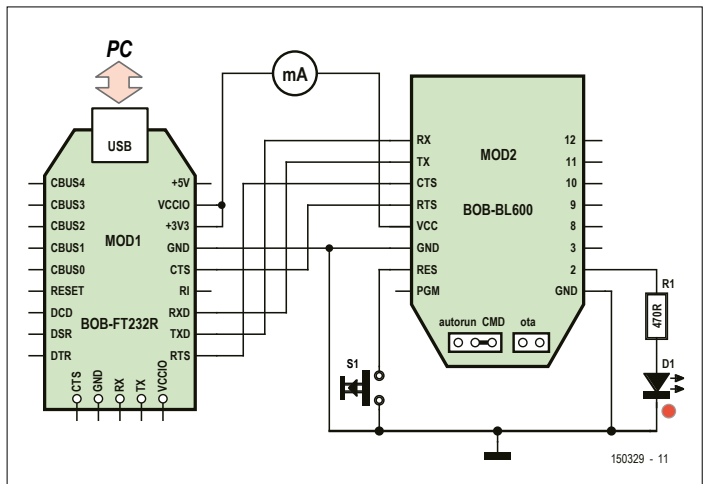


Bild 1. Wir setzen – wie es schon Gewohnheit ist – den BoB FT232 von Elektor [8] ein, um mit dem BL600 zu kommunizieren.

Stückliste:

Widerstände:

R1 = 470 Ω

Halbleiter:

D1 = LED 3 mm (Farbe nach Wahl)

Verschiedenes:

S1 = Taster

MOD1 = e-BoB FT232 bestückt 110553-91 [7]

MOD2 = e-BoB BL600 bestückt 140270-91 [7]
oder Platine 140270-1 [7]

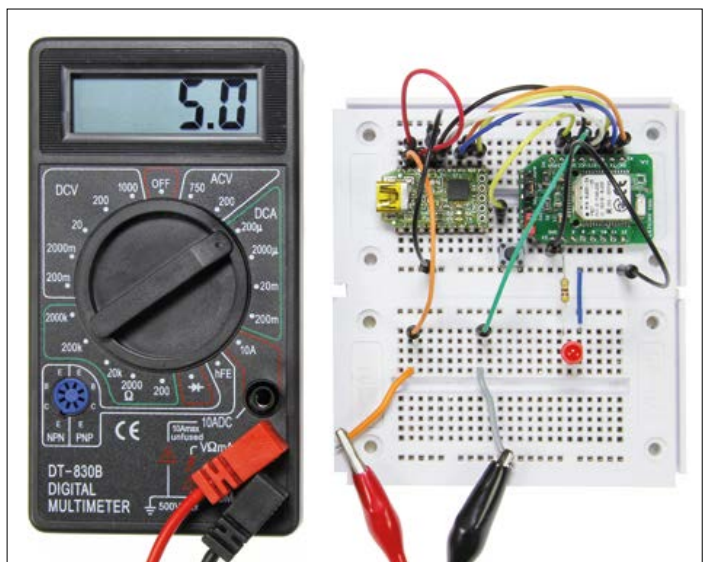


Bild 2. Foto des Versuchsaufbaus auf einem Steckbrett.

Die beiden TIMER:

• Code in **Grün**:

- Deklaration des Handlers: *OnEvent EVTMR1 Call FuncTimer1*
- die Funktion: *Function FuncTimer1()*
- Aktivieren der LED: *GpioWrite(2,1)*
- Start von Bluetooth: *BleAdvertStart*
- Auslösen des nächsten Timers: *TIMERSTART(2,25,0)*

• Code in **Rot**:

- Deklaration des Handlers: *OnEvent EVTMR2 Call FuncTimer2*
- die Funktion: *Function FuncTimer2()*
- Löschen der LED: *GpioWrite(2,0)*
- Bluetooth-Stopp: *BleAdvertStop*
- Auslösen den nächsten Timers: *TIMERSTART(1,4000,0)*

Die Funktion *BleVspUartBridge* muss deaktiviert werden, wenn wir den UART abschalten und dadurch weniger verbrauchen

wollen. Zum Prüfen der korrekten Funktion des Moduls senden wir den empfangenen Text mit einem Versatz von einem Zeichen zurück: a wird zu b, d zu e... (Code in **Orange**).

- Der Code der Funktion *HandlerLoop* findet sich in der Bibliothek *cli.upass.vsp.sbib*. Um Dubletten zu vermeiden, die zu Fehlern bei der Kompilierung führen, wird sie umbenannt in **MyHandlerLoop**.
- *MyHandlerLoop* wird in folgenden Fällen aufgerufen:
 - Datenempfang am UART des Moduls.
 - Datenempfang am Bluetooth-Port.
 - Der Bluetooth-Buffer ist leer.
 - Der UART-Buffer ist leer.
- Die Funktion *BleVspUartBridge* bildet eine Schleife zwischen Smartphone und UART-Port. Diese Codezeile wird auskommentiert.
- Funktion *BleVSpRead* erlaubt das Lesen von Daten über Bluetooth. Achtung, das herstellerbedingte Limit liegt bei 20 Zeichen.
- Mit der Funktion *STRPOS* kann man ermitteln, ob der empfangene Text mit *carriage return* endet.

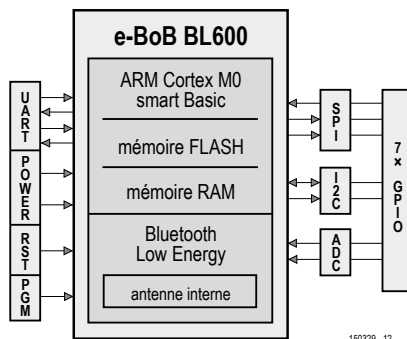
e-BoB BL600

Übersicht der Features des BL600-Moduls:

- Interfaces UART, I²C, SPI
- 28 Mehrzweck-Ein-/Ausgänge (GPIO)
- 6 Analog-Eingänge (ADC mit 10 bit)
- Stromaufnahme: 5 µA im Betriebs-, 0,4 µA im Sleep-Modus
- 10 mA während des Sendens
- einfache Programmierung in der Sprache smartBASIC
- Schaltkreis nRF51822 von *Nordic Semiconductor*.
- geringe Abmessungen: 19 x 12,5 x 3 mm
- Reichweite im offenen Feld: bis zu 20 m

Bluetooth in BASIC!

Die Programmiersprache smartBASIC (Event-orientiert) des BL600-SA vereinfacht die Integration von Bluetooth in Ihren Applikationen, weil nicht nur die Verwaltung der direkt an das Modul angeschlossenen Sensoren erleichtert wird, sondern auch die Weitergabe von Werten, die an irgendeinem Bluetooth v4.0-Empfänger anfallen (Smartphone, Tablet, Rechner, Interfaces...).



Kenndaten der Adapterkarte e-BoB von Elektor mit dem BL600-SA :

Port K1	serieller Port zum Laden der Programme in den BL600
	Anschlüsse Stromversorgung (3,3 V)
	RESET-Eingang.
	PGM-Anschluss (für eventuelle Programm-Updates des Moduls)
Port K2 7 Logik-Ein-/Ausgänge	2 analoge Eingänge mit je 10 bit (Pins 2 und 3)
	I2C-Port (Pins 8 und 9)
	SPI-Port (Pins 10, 11 und 12)
Jumper JP1	cmd: AT-Kommandos
	autorun
Jumper JP2	OTA (Over The Air)

Das Android-Programm

Auf der Website des Herstellers [7] des BL600 finden Sie den Sourcecode der App *Toolkit* für Android (und auch iOS), die folgende Funktionen umfasst:

- BPM (Blutdruck)
- HRM (Herzrhythmus)
- *Proximity*
- HTM (medizinisches Thermometer)
- *Serial* (UART)
- OTA (*Over The Air*)
- *Batch*

▶ Weitere Projekte für den BL600 warten schon in den Labors der Autorin und im Elektor-Labor.

- Um den Versatz von einem Zeichen bei den zurückzusendenden Texten zu erzeugen, verwenden wir die beiden Funktionen *StrGetChr* (Umwandeln eines Zeichens an angegebener Position in einem String in ein Byte) und *StrSetChr* (Setzen eines Zeichens an angegebener Position durch vorgegebenes Byte).
- Und schließlich wird der Text via Bluetooth mit Funktion *BleVspWrite* versandt.

Das Programm versendet keine Daten mehr über den UART, wir können diesen Port also schließen (nach willkürlich festgelegter Wartezeit von 1000 ms, Code in **Lila**):

- Deklaration des Handlers: *OnEvent EVTMR0 Call*

FuncClose

- die Funktion: *Function FuncClose()*
- Schließen des Ports: *UartClose()*
- Pins TX und RTS werden zum Ausgang mit High- bzw. Low-Pegel: *GPIOSetFunc*
- Auslösen des Timers im Programm MAIN: *TIMERSTART(0,1000,0)*

Zur Überwachung des Verbindungsstatus dient die LED: an bei Verbindung, aus bei Trennung (in Wirklichkeit blinkt sie in Abhängigkeit von TIMER1 und TIMER2). Der Code in **Blau** der Funktion *HandlerBleMsg* stammt aus der Bibliothek *cli.upass.vsp.sbib*:

Wir halten für Sie zum Download [6] den extrem einfach gehaltenen Sourcecode einer App für Android-Smartphones bereit, die nur die UART-Routinen nutzt.

Dazu haben wir *Android Studio* eingesetzt, verfügbar unter Windows, MAC OS und Linux [7]. Zur Erinnerung, der Hersteller hat eine Bibliothek *laird_library_ver.0.18.1.1.jar* entwickelt, um die Entwicklung von Android-Apps für klassisches Bluetooth und *Bluetooth Low Energy* zu beschleunigen.

Die in dieser Artikelserie beschriebenen Funktionen:

Ein-/Ausgänge:

- Parametrisierung des BL600-Anschlusses (nFunction = 1 oder 2):
`rc = GPIOSETFUNC(nSigNum, nFunction, nSubFunc)`
- Lesen eines Wertes 0 oder 1: `rc = GPIOREAD(nSigNum)`
- Schreiben (nNewValue = 0 oder 1): `rc = GPIOWRITE(nSigNum, nNewValue)`

Analog-Digital-Wandler:

- Parametrisierung des BL600-Anschlusses (nFunction = 3):
`rc = GPIOSETFUNC(nSigNum, nFunction, nSubFunc)`
- Lesen des Wertes (0 à 1023): `rc = GPIOREAD(nSigNum)`

I²C-Port:

- Öffnen des Ports: `rc = I2COpen(nClockHz, nCfgFlags, nHande)`
- Schreiben eines 8-bit-Wertes:
`rc = I2CWriteReg8(nSlaveAddr, nRegAddr, nRegValue)`
- Lesen eines 8-bit-Wertes:
`rc = I2CReadReg8(nSlaveAddr, nRegAddr, nRegValue)`
- Schließen des Ports: `I2CClose(handle)`

SPI-Port:

- Öffnen des Ports: `rc = SpiOpen(nMode, nClockHz, nCfgFlags, nHande)`
- Lesen / Schreiben: `rc = SpiReadWrite(stWrite$, stRead$)`
- Schließen des Ports: `SpiClose(handle)`

Tools zum Kompilieren und Übertragen

Alle Tools und Beispiele stehen zum Download bereit [7]. Wenn Sie einen Account auf der Website des Herstellers einrichten möchten, geben Sie „Elektor reader“ ein. Dann genügt es, die Datei *Firmware Files version 1.5.70.0 - Revision 5* herunterzuladen und zu entpacken, welche Folgendes enthält:

- Programmbeispiele im Ordner *smartBASIC_Sample_Apps*
- das Programm *UwTerminal.exe* dans im Ordner *smartBASIC_Sample_Apps*
- die Bibliothek im Ordner *smartBASIC_Sample_Apps/lib*
- Beispiele (*UserManualExampleCode*)
- die spezifische Konfiguration für Notepad++ für *smartBASIC* (*smartBASIC(notepad++).xml*)

Häufige Befehle

- AT I 0: Revisions-Nummer des BL600
- AT I 3: Version des Programmpakets des BL600
- AT+DIR: Liste der Programme auf dem BL600
- ATZ: Reset des BL600
- AT&F 1: Löschen des Speichers und Neustart des BL600
- AT+RUN „xxxx“: Ausführen des Programms xxxx

- Wir haben den Code der Funktion *HandlerBleMsg* der Bibliothek *cli.upass.vsp.sblib* entnommen. Um auch hier wieder Dubletten und Kompilationsfehler zu vermeiden wird sie umbenannt in ***MyHandlerBleMsg***.
- Bei Verbindungsaufbau *BLE_EVBLEMSGID_CONNECT* :
 - Anhalten der TIMER: *TIMERCANCEL*
 - LED anschalten: *GpioWrite(2,1)*
- Bei Trennung der Verbindung *BLE_EVBLEMSGID_DISCONNECT* :
 - LED ausschalten: *GpioWrite(2,0)*
 - (Wieder-)Auslösen des ersten Timers: *TIMERSTART(1,10,0)*

Hiermit haben wir alle notwendigen Elemente beisammen, um Ihr Programm *BL600-6.sb* zu erstellen. Nachdem man die geringe Stromaufnahme verifiziert hat, empfiehlt es sich für eine bessere Verbindung zu Ihrem Smartphone, die Parameter von *TIMER1* von 4000 ms auf z.B. 300 ms zu ändern: *TIMERSTART(1,300,0)*. Nun ist der Moment gekommen, sich an die

Programmierung Ihres *eigenen* BL600 zu wagen, auf Ihrem *eigenen* e-BoB, für Ihre *eigene* Android-App; alle notwendigen Informationen finden sich im Artikel des letzten Monats [6].

Ein offenes Ende...

Ihr *eigenes* Programm – das ist das richtige Resümee dieser Serie, die mit dem Projekt Bluetooth-Thermometer begonnen hat und die wir jetzt mit doppeltem Stolz beenden: unsere Arbeit, die Inbetriebnahme dieses Moduls so einfach wie möglich zu gestalten, wurde sowohl von der Elektor-Leserschaft als auch vom Hersteller des BL600-SA anerkannt und geschätzt. Weitere Projekte für den BL600 warten schon in den Labors der Autorin und im Elektor-Labor. Außerdem hat der BL600 einen großen Bruder, den BL620. Das ist ein weiteres Bluetooth-Modul, aber mit *Master*-Fähigkeit, das die Kommunikation noch einmal sehr vereinfacht. Mehr in Kürze in Ihrem Elektor-Magazin... ◀

(150329)

```
//*****
// Laird Technologies (c) 2013
// Jennifer AUBINAIS (c) 2015
// ++++++

//*****
// Definitions
//*****
#define AUTO_STARTUP          1
//Set this to 0 to disable all debugging messages
#define ENABLE_DEBUG_PRINTS   0
#define DEVICENAME            "JA_TEST"
#define DEVICENAME_WRITABLE   1
**** code here ****

//*****
// Library Import
//*****
#include "lib\cli.upass.vsp.sblib"
//*****
// Global Variable Declarations
//*****
DIM text$

//*****
// Function and Subroutine definitions
//*****

//=====
// CLOSE UART
//=====
Function FuncClose()
  DIM rc
  UartClose()
  rc = GPIOSetFunc(21,2,1) '// TX
  rc = GPIOSetFunc(23,2,0) '// RTS
  TIMERSTART(1,10,0)
ENDFUNC 1
```

```
//=====
// Receive data
//=====
function MyHandlerLoop()
  //BleVspUartBridge()
  DIM n, rc, tempo$, tx$
  DIM pos, return$
  // Wait return from received data
  tx$ = "0D"
  return$ = StrDehexize$(tx$)
  tempo$ = ""
  n = BleVspRead(tempo$,20)
  text$ = text$ + tempo$
  pos = STRPOS(text$,return$,0)
  IF ( pos >= 0 ) THEN
    DIM i, x
    pos = pos - 1
    FOR i = 0 TO pos
      x = StrGetChr(text$,i)
      rc = StrSetChr(text$,x+1,i)
    NEXT
    rc = BleVspWrite(text$)
    text$ = ""
  ENDIF
endfunc 1

//=====
// TIMER 1
//=====
FUNCTION FuncTimer1()
  dim rc, Adr$
  Adr$ = ""
  // led on
  GpioWrite(2,1)
  rc = bleadvertstart(0,Adr$,25,25,0)
  TIMERSTART(2,25,0)
ENDFUNC 1
```

Links

- [1] Elektor Januar/Februar 2015
www.elektormagazine.de/140190
Drahtloses Thermometer mit Bluetooth Low Energy
Fernanzeige der Temperatur auf Ihrem Smartphone
- [2] Elektor März 2015
www.elektormagazine.de/140270
e-BoB BL600: Modul für Bluetooth Low Energy | Teil 1:
Eine Plattform für die drahtlose Kommunikation
- [3] Elektor April 2015
www.elektormagazine.de/150014
e-BoB BL600 | Teil 2:
Ein Programm editieren, kompilieren und übertragen mit
dem Bluetooth Low Energy-Modul

- [4] Elektor Mai 2015
www.elektormagazine.de/150129
e-BoB BL600 | Teil 3:
Das BLE-Modul in smartBASIC programmieren
- [5] Elektor Juni 2015
www.elektormagazine.de/150130
e-BoB BL600: Bluetooth Low Energy | Teil 4:
Der I²C-Port mit Temperatur-Sensor
- [6] Elektor Juli/August 2015
www.elektormagazine.de/150272
e-BoB BL600 | Teil 5:
SPI-Port mit DA-Wandler und Android-App
- [7] https://laird-ews-support.desk.com/?b_id=1945
- [8] www.elektor.de/ft232r-usb-serial-bridge-bob-110553-91

```
'//=====
'// TIMER 2
'//=====
FUNCTION FuncTimer2()
    // led off
    GpioWrite(2,0)
    rc = bleadvertstop()
    TIMERSTART(1,4000,0)
ENDFUNC 1
'//=====
'// This handler is called when there is a BLE
message
'//=====
function MyHandlerBleMsg(BYVAL nMsgId AS INTEGER,
BYVAL nCtx AS INTEGER) as integer
'// Inform libraries
ConnMgrOnBleMsg(nMsgId,nCtx)
AdvMgrOnBleMsg(nMsgId,nCtx)
select nMsgId
    case BLE_EVBLEMSGID_CONNECT
        TIMERCANCEL(2)
        TIMERCANCEL(1)
        DbgMsgVal(" --- Connect : ",nCtx)
        ShowConnParms(nCtx)
        // set at High
        GpioWrite(2,1)
    case BLE_EVBLEMSGID_DISCONNECT
        DbgMsgVal(" --- Disconnect : ",nCtx)
        // set at Low
        GpioWrite(2,0)
        TIMERSTART(1,10,0)
        ***** code here *****
endselect
endfunc 1
'-----
'// TIME OUT = nothing
'-----
```

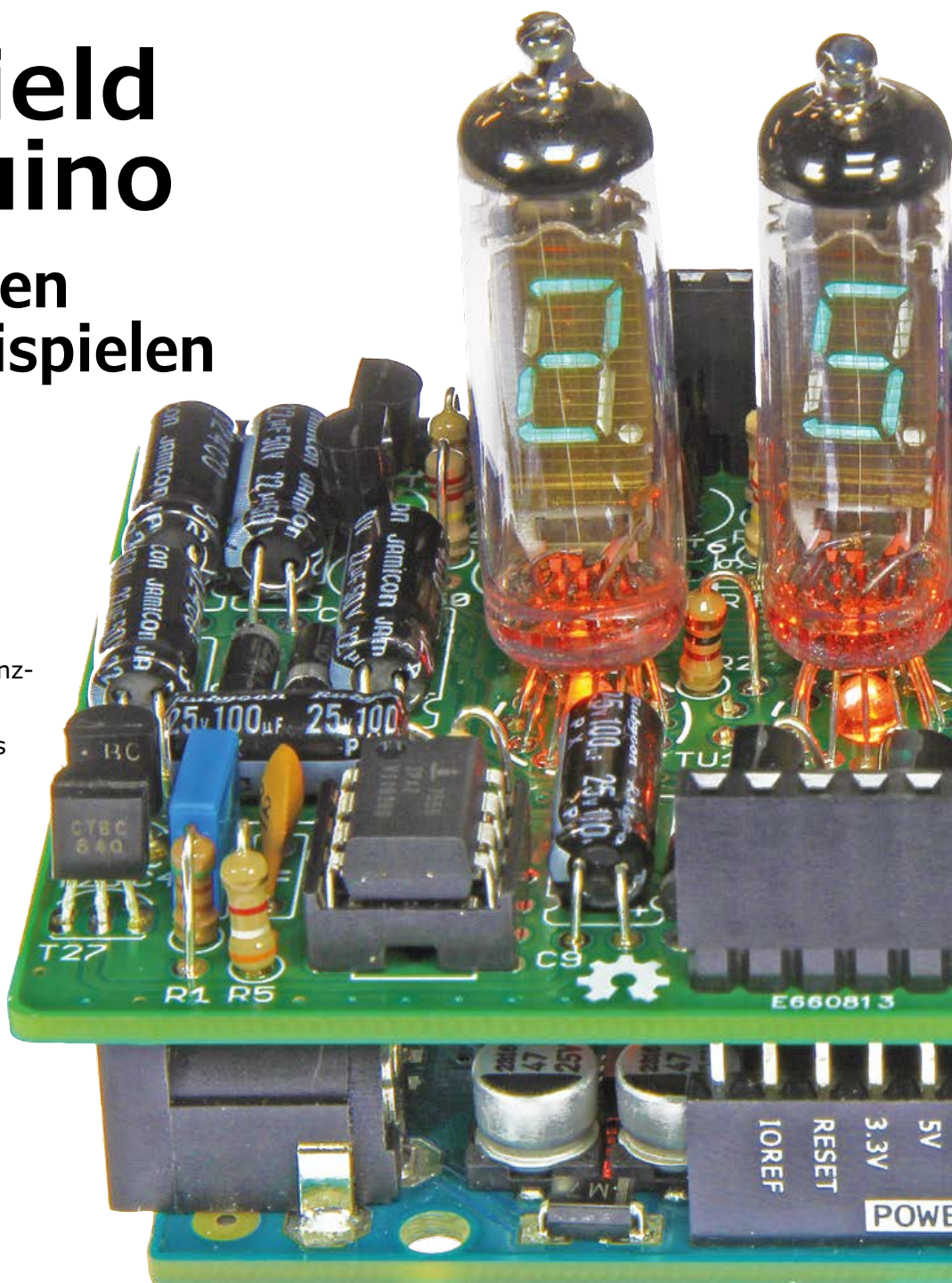
```
Function MyHandlerTimOut() as integer
    'NOTHING
EndFunc 1
'//*****
'// Handler definitions
'//*****
ONEVENT EVTMR1 CALL FuncTimer1
ONEVENT EVTMR2 CALL FuncTimer2
OnEvent EVTMR0 Call FuncClose
OnEvent EVBLEMSG call MyHandlerBleMsg //
EVBLEMSG indicate when significant BLE event occurs.
OnEvent EVBLE_ADV_TIMEOUT Call MyHandlerTimOut
//all events have the same handler
OnEvent EVVSPRX call MyHandlerLoop //
EVVSPRX is thrown when VSP is open and data has
arrived
OnEvent EVUARTRX call MyHandlerLoop //
EVUARTRX = data has arrived at the UART interface
OnEvent EVVSPTXEMPTY call MyHandlerLoop
OnEvent EVUARTTXEMPTY call MyHandlerLoop
'//*****
'// main
'//*****
text$ = ""
// pin 2 output at low
rc = GPIOSetFunc(2,2,0)
rc = bleadvertstop()
IF (ENABLE_DEBUG_PRINTS == 1) THEN
    UartRsp(0)
    TIMERSTART(2,10,0)
ELSE
    PRINT "low Energy"
    TIMERSTART(0,1000,0)
ENDIF
WaitEvent
```

VFD-Shield für Arduino

Mit instruktiven Programmbeispielen

Von **Ilse Joostens** (Belgien)

VFD bedeutet Vakuum-Fluoreszenz-Display: Hier sind es vier VFD-Siebensegment-Ziffernröhren aus russischer Fertigung, die von einem Arduino Uno gesteuert werden. Für eine stilgemäße LED-Beleuchtung ist ebenfalls gesorgt. Zum Repertoire der Programmbeispiele gehören eine Demo, eine Digitaluhr, ein Voltmeter und ein Thermometer.



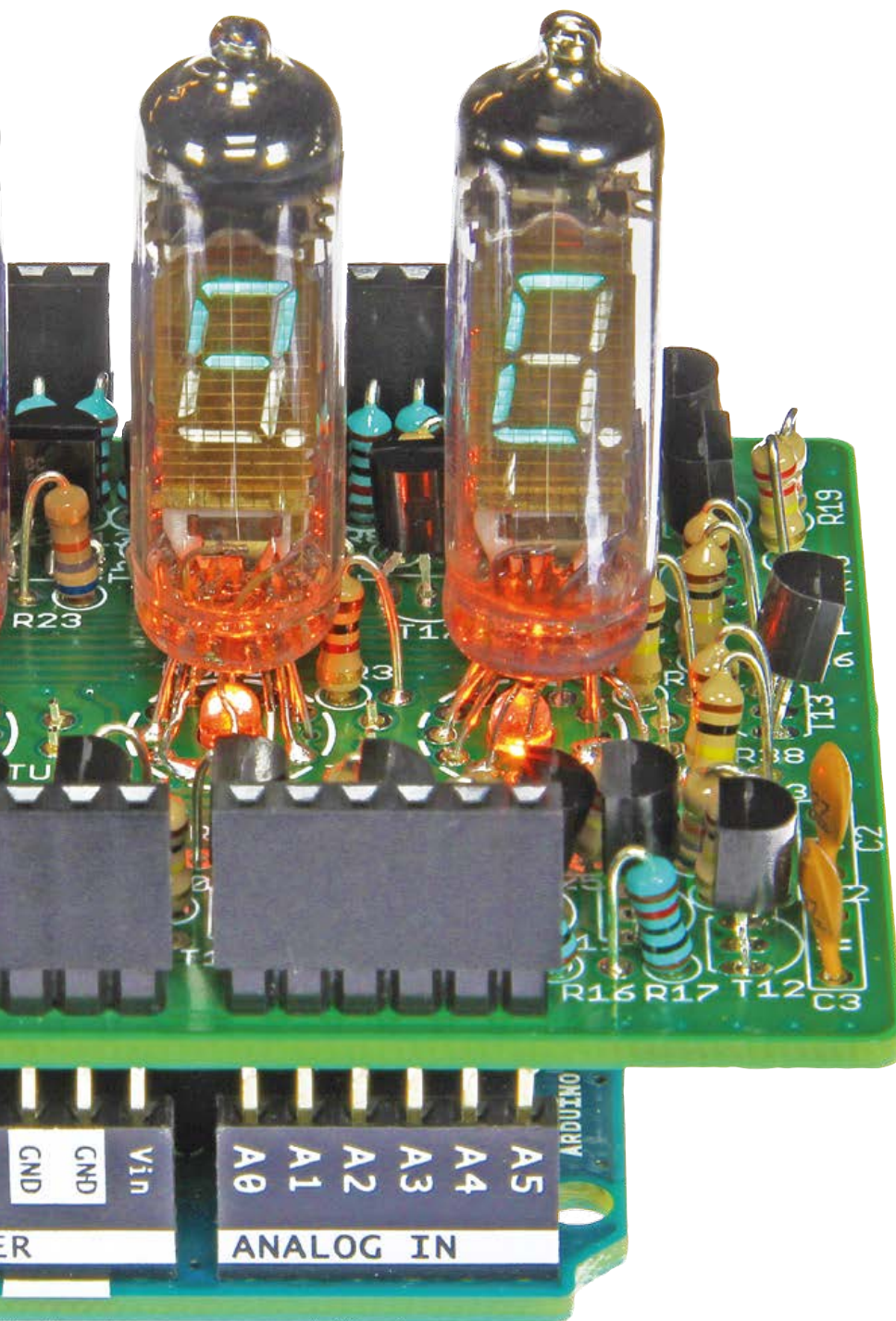
Bei den Freunden nostalgischer Elektronik sind Nixie-Röhren schon lange beliebt, das beweisen die zahllosen Bauvorschlge, die im Internet kursieren. Auch Elektor hatte sich in der Vergangenheit schon mehrfach dieses Themas angenommen. Anders als bei jenen Projekten werden diesmal nicht Nixie-Röhren, sondern sogenannte VFD-Röhren eingesetzt. Das Prinzip, das ihnen zugrunde liegt, ist mit den Nixie-Röhren nicht vergleichbar. VFD-Röhren und Nixie-Röhren sind gewissermaßen Konkurrenten, wobei die VFD-Röhren an Beliebtheit aufholen.

Dieses Arduino-Shield soll auch dem Newcomer Spaß bereiten, deshalb gehören exotische Bauteile und SMDs nicht dazu. Es kann sogar sein, dass die meisten Bauteile bereits im Werschrank lagern und auf ihren Einsatz warten. Außerdem treten hier keine gefährlichen Spannungen auf, die höchste Spannung liegt bei 35 V.

Prinzip der VFD-Röhre

Die VFD-Röhre IV-3 (**Bild 1**) ist eine Ziffernanzeigeröhre mit der Ziffernhöhe 8 mm. In dem von Gasen evakuierten Glaskolben sind der Glühfaden, das Git-

ter und die Ziffernsegmente einschließlich Dezimalpunkt eingeschmolzen. Die leuchtenden Balken und der Punkt befinden sich auf keramischen Trägern, auf denen ein Leuchtstoff aufgetragen ist. Wenn Strom durch den Glühfaden fließt, steigt die Temperatur, so dass Elektronen frei austreten können. Der Glühfaden ist mit einem Alkalimetall-oxid beschichtet, dadurch setzt dieser Vorgang schon bei relativ niedrigen Temperaturen ein. Weil der Glühfaden fast unsichtbar glimmt, kann er beim Betrachten der Ziffer nicht stören.



Eigenschaften

- Kompatibel mit Arduino-Entwicklungsboards
- Betriebsspannung 12 V stabilisiert, Versorgung vom Arduino-Board
- Einsatzmöglichkeiten: Uhr, Thermometer, Voltmeter, Zähler, Punkte-Anzeige, ...
- Mehrere Beispiel-Sketches für Arduino verfügbar

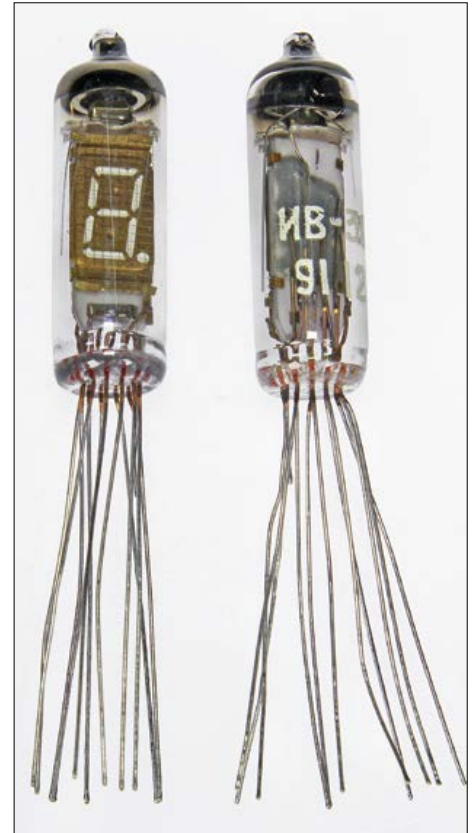


Bild 1. Vorder- und Rückansicht der russischen VFD-Röhre IV-3.

Wird an ein Ziffernsegment positive Spannung gelegt, dann werden die freien Elektronen in Richtung des Ziffernsegments beschleunigt. Beim Aufprallen auf den Leuchtstoff wird Energie frei, die den Leuchtstoff zum Leuchten anregt. Das Prinzip ist vergleichbar mit der klassischen Bildröhre, doch anders als dort leuchtet der Leuchtstoff deutlich sichtbar schon bei niedriger Elektronengeschwindigkeit. Die Anodenspannung muss nur etwa 30 V betragen, während bei der Bildröhre mehrere Kilovolt erforderlich sind. Das Gitter, das zwischen Glühfaden und

Segment angeordnet ist, dient zum Steuern des Elektronenstroms. Liegt am Gitter negative Spannung, dann kann der Elektronenstrom nicht in Richtung der Ziffernsegmente fließen, unabhängig von der Spannung an den Segmenten. Über das Gitter lassen sich folglich alle Segmente und der Dezimalpunkt gleichzeitig sperren. Die VFD-Röhre verhält sich wie eine Röhren-Triode, sie könnte sogar Signale verstärken (siehe [1]). Doch das ist hier nicht die Absicht, hier werden die Ziffernröhren mit den Gittern gemeinsam im Multiplexbetrieb gesteuert, was die

Schaltungstechnik vereinfacht.

Erwähnt sei noch, dass die dunkelgrau bis silbrig schimmernde Schicht oben in der VFD-Röhre der so genannte Getter ist, ein chemisch reaktiver Stoff, der das Vakuum aufrecht erhält, indem er vagabundierende Gasmoleküle bindet.

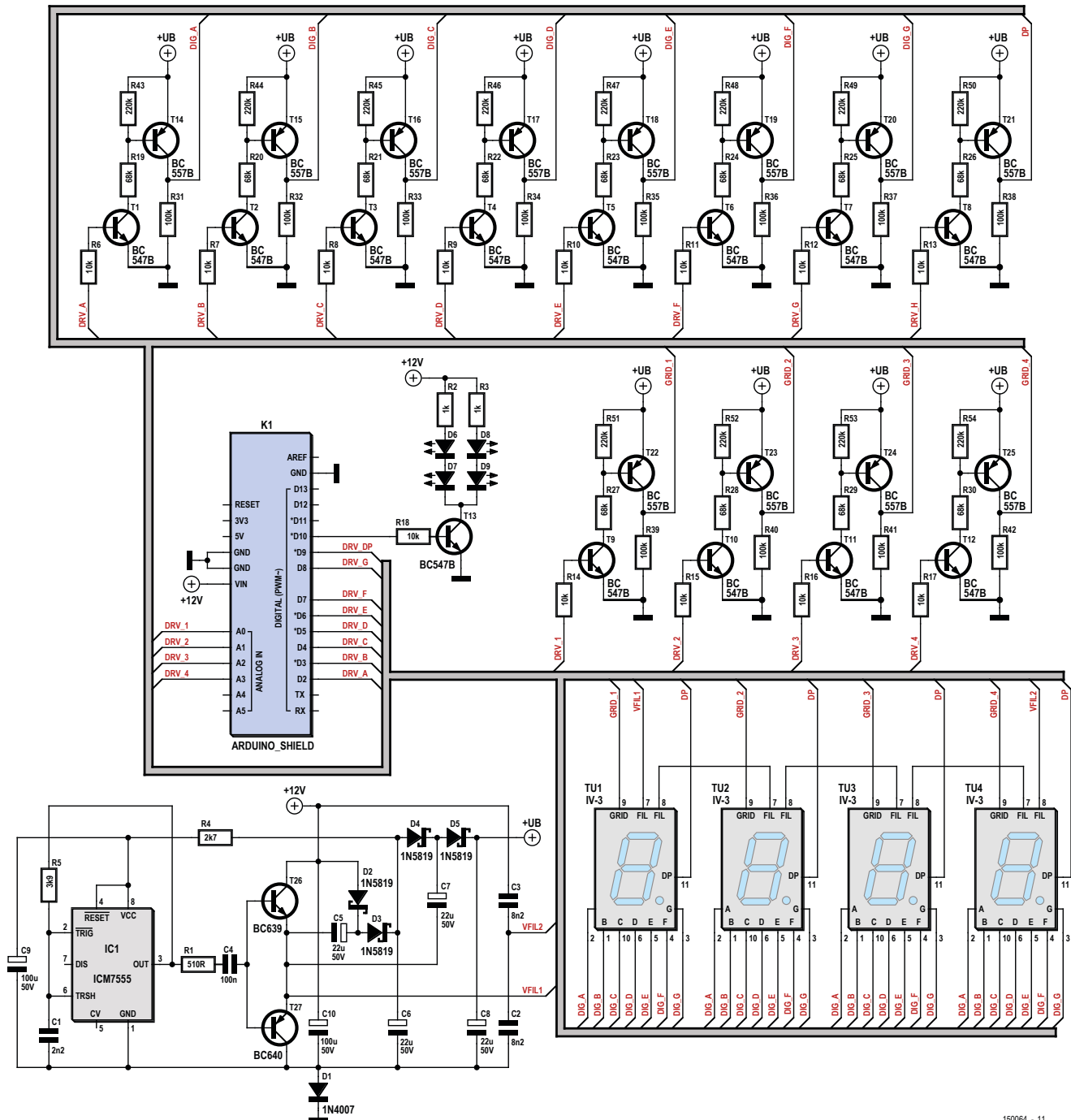
Shield-Schaltung

In der Schaltung des Shields (**Bild 2**) sind vier Funktionen erkennbar: Ein Spannungswandler, der die Betriebsspannung 12 V auf 32 V umsetzt, die Stromversorgung für die Röhren-Glühfäden, die Trei-

ber für die Segmente und Gitter der Ziffernröhren sowie ein Schalttransistor und vier LEDs für die Hintergrundbeleuchtung. Der Spannungswandler ist ein Verdreifacher, er ist mit einem 7555 (IC1), einer Pufferstufe und diversen Kondensatoren sowie Schottky-Dioden aufgebaut. Theoretisch beträgt die Ausgangsspannung 36 V, doch in der Praxis werden nur etwa 32 V erreicht, zumal Diode D1 in der Masseleitung einen zusätzlichen

Spannungsabfall verursacht. Die Spannung 32 V ist für die Ziffernröhren des Typs IV-3 ausreichend. Der 7555 ist als Rechteckoszillator geschaltet, die Frequenz hängt von C1 und R5 ab. Da der Ausgangsstrom nur geringe Werte annehmen kann, ist die Push-Pull-Stufe mit T26 und T27 nachgeschaltet. Die Höhe der Ausgangsspannung des 7555 reicht allerdings nicht aus, um T26 vollständig in die Sättigung zu

steuern. Deshalb wird die Betriebsspannung des 7555 über R4 der verdoppelten Betriebsspannung an Verbindungspunkt D3/D4/C6 entnommen, wegen des Spannungsabfalls an R4 beträgt sie nur ungefähr 14 V. Dadurch übersteigt die Basisspannung von T26 die Kollektorspannung, so dass die interne Basis-Kollektor-Diode leitet und die vom 7555 kommende Ausgangsspannung begrenzt. Um dem entgegen zu wirken, sind R1



150064 - 11

Bild 2. Die Schaltung besteht aus den Treibern zum Steuern der Ziffernröhren sowie einem Spannungsverdreifacher für die Anodenspannung.



Nostalgie, diesmal nicht mit Nixies, sondern mit VFD-Röhren

und C4 hinzugefügt. Außerdem sorgt C4 für eine hohe Basisstrom-Spitze in Richtung T26, jedes Mal wenn der Ausgang des 7555 von niedriger auf hohe Spannung schaltet. Die Kondensatoren C5...C8 des Spannungsverdreifachers werden beschleunigt geladen, was zur Folge hat, dass der Wirkungsgrad des Spannungswandlers nicht unwesentlich steigt. Als Nebenvorteil müssen T26 und T27 nicht gekühlt werden, auch nicht bei hoher Belastung der Ausgangsspannung. Der Spannungsabfall an Diode D1 hat zur Folge, dass die Stromversorgung bei ungefähr 0,7 V über Massepotential abschaltet. Wenn die Spannung an einem Segment oder Gitter der Ziffernröhren nach Masse gezogen wird, ist die Spannung bezogen auf den Glühfaden negativ. Die Segmente bleiben dann vollständig dunkel.

Die Glühfäden der Ziffernröhren sind in Reihe geschaltet, was den Schaltungsaufwand mindert. Ferner liegt nicht Gleichspannung, sondern Wechselspannung an den Glühfäden, damit das Spannungsgelände der Glühfäden keine erkennbaren Helligkeitsunterschiede zwischen den Segmenten und den Ziffernröhren verursacht. Die Glühfäden liegen auf der einen Seite an den gemeinsamen Emittern von T26 und T27, auf der anderen an C2 und C3. Weil C2 und C3 Gleichstromanteile sperren, fließt nur der Wechselstromanteil durch die Glühfäden. Der Strom, der hier fließt, hängt von den Impedanzen dieser Kondensatoren ab. Deshalb sind die Werte von C2 und C3 sowie die Werte der frequenzbestimmenden Komponenten R5 und C1 einigermaßen kritisch.

Die Segmente und Gitter der Ziffernröhren werden von NPN/PNP-Transistorpaaren gesteuert. Bei hohem Ausgangssignal des Arduino (5 V) schaltet der PNP-Transistor die Spannung 32 V auf das zugehörige Segment oder Gitter. Ist das Signal niedrig, wird das Segment oder Gitter über einen 100-k-Widerstand nach Masse gezogen. Die Segmente und Dezimalpunkte der vier Ziffernröhren sind parallel geschaltet, so dass für die Multiplex-Steuerung 12 Datenleitungen genügen. Die I²C-Leitungen SDA und SCL des Arduino werden nicht belegt, so dass über I²C beispiels-

weise ein Uhrenbaustein (RTC) mit dem Shield gekoppelt werden kann.

Software

Für das Shield sind mehrere Programmbeispiele verfügbar: Eine Demo, eine Uhr, ein Voltmeter und ein Thermometer. Die Programme, so genannte Arduino-Sketches, können als ZIP-Datei gepackt von der Projektseite [2] heruntergeladen werden. Die Quellcodes enthalten ausführliche Hinweise zur Funktion und Arbeitsweise der Programme sowie gegebenenfalls zu den benötigten Bibliotheken.

Das Programm „Demo“ setzt eine Animation auf die Ziffernröhren, gleichzeitig variieren die Hintergrund-LEDs ihre Leuchtintensität. Mit diesem Programm lässt sich der korrekte Aufbau der Hardware überprüfen.

Das Programm „Uhr“ zeigt die Zeit und das Datum an. Da der Arduino Uno nicht mit einer Echtzeituhr ausgestattet ist, müssen Zeit und Datum über den seriellen Datenanschluss eingegeben werden. Mit dem *Serial Monitor* der Arduino IDE ist dies schnell getan.

Das Voltmeter misst Spannungen an Pin A5 des Arduino Uno und gibt die Messwerte auf den Ziffernanzeigeröhren aus. Der Messbereich erstreckt sich von 0 V bis 5 V, angezeigt als 0.00 bis 5.00.

Das Programm „Thermometer“ liest die Temperatur aus einem Eindraht-Thermosensor und zeigt den Messwert in °C an. Unterstützt werden die Sensoren DS18B20, DS18S20 und DS1820. Das Programm sucht periodisch nach einem unterstützten Sensor. Sobald ein Sensor erkannt wird, folgen das Auslesen und Anzeigen des Werts. Der Eindraht-Thermosensor kann bei laufendem Programm getauscht werden.

Alle Programmbeispiele enthalten ein Steuermodul für die Ziffernanzeigeröhren. Die Steuerung arbeitet mit Interrupts, sie benutzt eine ISR (Interrupt Service Routine). Die Ziffernröhren werden nacheinander in zyklischer Folge eingeschaltet, so dass gleichzeitig immer nur eine Ziffernröhre aktiv ist. Die Steuerung besteht aus folgenden Schritten:

- Abschalten der Gitter und Segmente der aktiven Röhre,

- Verzögerung 40 µs, um „Ghosting“ zu vermeiden,
- Aktivieren der nächsten Röhre im Zyklus,
- Schreiben der Ziffer, Pins 2...9, Anpassen der Segmente,
- Einschalten des Gitters der aktiven Röhre.

Falls die Zeit vom Ausschalten einer Ziffernröhre bis zum Einschalten der nächsten Röhre zu kurz ist, können die Segmente der ersten Röhre nachglimmen, eine Erscheinung die „Ghosting“ genannt wird. Die Wartezeit 40 µs wirkt dem Ghosting entgegen, sie wurde experimentell gefunden. Die tatsächliche Zeit zwischen dem Umschalten der Gitter ist etwas länger, da noch anderer Programmcode ausgeführt wird.

Die Multiplex-Frequenz der Anzeige liegt bei 250 Hz. Jede Ziffernröhre ist 62,5 Mal in der Sekunde für knapp 4 ms aktiv geschaltet. Bei dieser Frequenz erscheint die Anzeige dem menschlichen Auge flimmerfrei.

Die anzuzeigenden Muster sind in einer Datenstruktur gespeichert (tube_list), ein Array aus vier Elementen, ein Element für jede Ziffernröhre. Jedes Element enthält ein Flag für den Dezimalpunkt und einen Index in einer Tabelle mit den Mustern für die Segmente.

Im Programm „Demo“ ist eine PWM-Steuerung für die Hintergrund-LEDs enthalten. Ein PWM-Zyklus besteht aus acht Schritten. Die Schrittfrequenz beträgt 500 Hz, die PWM-Frequenz 62,5 Hz. Bei dieser Frequenz leuchten die LEDs für das menschliche Auge flimmerfrei, das gilt auch für den niedrigsten Duty-Cycle 1 : 8. Die ISR ist mit Timer 1 verknüpft. Die „Programmiersprache“ des Arduino kennt keine Elemente für den Umgang mit Timern. Der Quellcode der Programme benutzt deshalb AVR Libc, um Timer 1 und die ISR in C zu implementieren. Die Arduino-Programmierungsumgebung macht übrigens ebenfalls von Libc Gebrauch.

Aufbau

Die für das VFD-Shield entworfene Platine ist in **Bild 3** wiedergegeben. Die Platine wird ausschließlich mit bedrahteten Komponenten aufgebaut, so dass SMD-Lötverfahren nicht erforderlich sind.

Für das VFD-Shield ist ein Baupaket einschließlich der vier Ziffernröhren im Elektor-Shop erhältlich. Der Aufbau wird ausführlich in einer separaten, englischsprachigen Bauanleitung beschrieben, sie kann ebenfalls von der Projektseite [2] heruntergeladen werden. An dieser Stelle können wir nur das Wichtigste streifen. Alle Bauelemente werden ausnahmslos auf der Platineseite mit dem Bestückungsaufdruck montiert. Den Anfang machen die Widerstände, Dioden, Kondensatoren und die IC-Fassung. Die Widerstände sind stehend zu montieren, dazu wird ein Anschlussdraht zur Gegenseite gebogen. Die Elkos liegen auf der Platine, nachdem die Anschlussdrähte rechtwinklig umgebogen sind. Wir empfehlen, die Stiftkontaktleisten zunächst lose auf die Platine zu stecken und die Konstruktion provisorisch auf den Arduino Uno zu setzen. Bei dieser Passprobe werden zwei oder drei Pins zur Fixierung angelötet, danach wird der Uno abgezogen und die übrigen Pins werden verlötet.

Die nächsten Bauelemente sind die Transistoren, gefolgt von den vier LEDs der Hintergrundbeleuchtung. Die Platinenbohrungen für die LED-Anschlüsse befinden

sich unter den Ziffernröhren. Die LED-Anschlüsse sind etwas gegeneinander versetzt. Nachdem die LED-Anschlussdrähte in die korrekte Form gebogen sind, so dass sie in die Bohrungen passen, können sie angelötet werden.

Zuletzt folgt die Montage der Ziffernröhren. An den Röhren befinden sich lange Anschlussdrähte, sie werden durch die zugehörigen Platinenbohrungen gesteckt. Falls das Einfädeln nicht gelingen will, hat sich das spiralförmige Kürzen der Drähte bewährt, so dass jeder Draht unterschiedliche Länge hat. Die Anschlussdrähte werden so weit in die Platine gesteckt, dass die Unterkante der Röhre die Oberkante der Stiftkontaktleisten etwas überragt. Wenn alle Anschlüsse eingefädelt wurden und die Röhren ausgerichtet sind, wird zunächst nur ein Draht jeder Röhre angelötet. Die Röhren müssen jetzt so justiert werden, dass sie exakt senkrecht auf der Platine und parallel zueinander stehen. Erst wenn das Gesamtbild stimmt, werden die übrigen Anschlussdrähte angelötet. Danach sind die Positionen der Röhren nicht mehr korrigierbar. Zum Schluss bleibt nur noch, den 7555 in die Fassung zu stecken.

Laden Sie die Software von der Projektseite [2] herunter, und spielen Sie den Demo-Sketch über die USB-Verbindung auf den Arduino. Trennen Sie den Arduino vom PC und decken Sie die USB-Buchse auf dem Arduino mit einem Stück Isolierband ab. Damit wird verhindert, dass die USB-Buchse die Lötstellen auf dem VFD-Shield berührt. Stecken Sie nun das VFD-Shield auf den Arduino. Schließen Sie ein Steckernetzteil an den Arduino an, das die stabilisierte Spannung 12 V abgibt und mit mindestens 300 mA belastbar ist.

Wenn Ihnen beim Aufbauen kein Fehler unterlaufen ist und sie jetzt das Steckernetzteil mit dem Stromnetz verbinden, zählen die Ziffernanzeigeröhren in einer Endlosschleife, und die Helligkeit der LEDs variiert langsam. Laden Sie die übrigen Programmbeispiele in den Arduino, probieren Sie die Beispiele aus und kreieren Sie eine neue Anwendung. Viel Spaß! ◀

(150064)gd

Weblinks

[1] www.youtube.com/watch?v=aiszksJs9C8

[2] www.elektormagazine.de/150064

Stückliste

Widerstände:

R1 = 510 Ω
 R2,R3 = 1 k
 R4 = 2k7
 R5 = 3k9
 R6...R18 = 10 k
 R19...R30 = 68 k
 R31...R42 = 100 k
 R43...R54 = 220 k

Kondensatoren:

C1 = 2n2
 C2,C3 = 8n2
 C4 = 100 n
 C5...C8 = 22 μ /50 V stehend
 C9,C10 = 100 μ /25 V stehend

Halbleiter:

D1 = 1N4007
 D2...D5 = 1N5819 Schottky-Diode
 D6...D9 = LED 3 mm, Farbe nach Wunsch
 T1...T13 = BC547B
 T14...T25 = BC557B
 T26 = BC639
 T27 = BC640
 IC1 = ICM7555

Außerdem:

TU1...TU4 = IV3(a) Siebensegment-VFD-Röhre
 DIP-Fassung 8-polig für IC1
 1 Satz Steckverbinder, Arduino-kompatibel,
 (1·6 Pins, 2·8 Pins, 1·10 Pins)
 Baupaket (inklusive 4 VFD-Röhren):
 150064-71

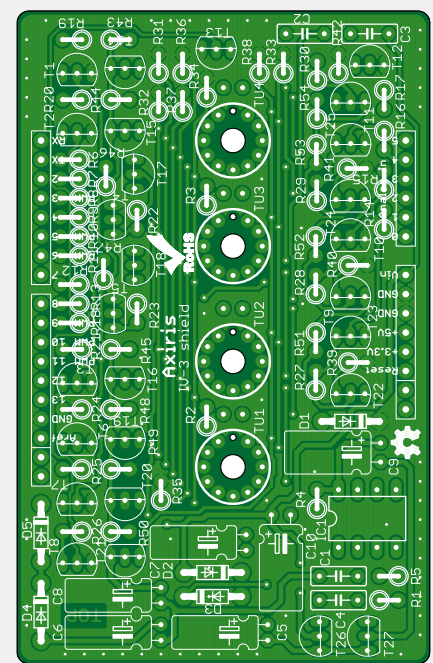
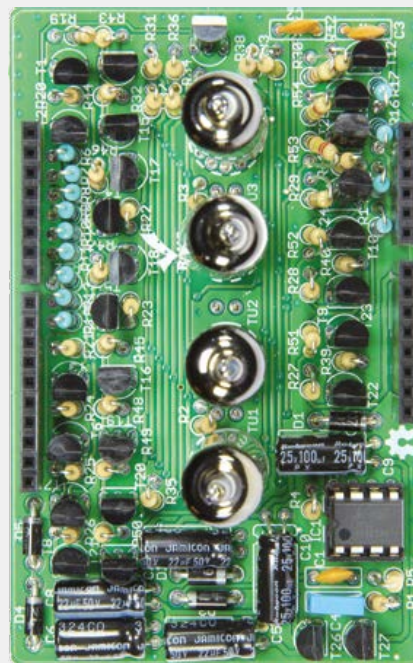


Bild 3. Auf der Platine werden ausschließlich bedrahtete Bauelemente verbaut.

Willkommen bei SHARE



Von **Jaime González-Arintero** – jaime.glez.arintero@eimworld.com

Play Hard

Achtung, Philosophie! Letzte Woche sah ich eine kurze Dokumentation, die die Gründe erkunden wollte, warum Menschen Spiele spielen. Die Schlussfolgerung war mehr oder weniger, dass das Leben selbst ein Spiel ist, aber das Ziel dieses Spiels nicht jedem klar erscheint. Anscheinend müssen

wir aber solche Ziele haben, und wenn keine Ziele offensichtlich sind, so müssen wir sie individuell selber setzen.

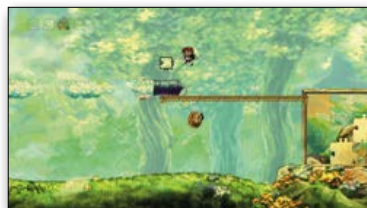
Wie die meisten Menschen meiner Generation habe ich Videospiele gespielt, als ich ein Kind war. Ich fand das Gymnasium nicht sehr amüsant, aber Videospiele schon. Damals waren Videospiele ziemlich teuer, so dass man sie normalerweise mit Freunden „teilte“. Computerspiele wurden gehackt und herumgereicht und es war kein Wunder, dass sie mit immer anspruchsvolleren Kopierschutz-Techniken auf den Markt kamen. Einen Internet-Zugang besaßen damals nur die wenigsten, aber man konnte immer die Bibliothek besuchen, um in Foren und Chats Anleitungen zu finden, wie man den neuesten Kopierschutz knacken könnte. Wieder zu Hause, versuchte man sich offline an verschiedenen Dingen. Es war ein „You vs. The Game“-Spiel. Ich bin nicht besonders stolz darauf, gebe aber zu, es getan zu haben.

Heutzutage zahle ich gerne für alles, auf Papier oder digital. Doch damals waren wir noch Kinder. Es war herausfordernd und schwierig. Und es hat Spaß gemacht!

Die meisten Spiele (und auch kommerzielle Software) konnten mit einer geklonten CD oder einem „gemounteten“ virtuellen Laufwerk (mit Software wie DAEMON Tools) gehackt werden. Andere Spiele benötigten mehr Schritte, etwa das Überschreiben von Registern, Dateien, DLLs mit Key-Generatoren. Dies war weithin bekannt und man konnte leicht Schritt-für-Schritt-Anleitungen in suspekten ZIP-Dateien irgendwo herunterladen. Sie mussten nur den Anweisungen zu folgen, und voilà: Das Spiel läuft.

Manchmal waren die Dinge komplizierter: Ein Schlüssel wurde intern vom Installer auf Basis der Computer-ID oder einer anderen verfügbaren Seriennummer lokal erzeugt und direkt in die Exe-Datei eingetragen. Mit etwas Glück und wenn die Anwendung selbst nicht verschlüsselt war, konnte man einem solchen Schlüssel mit einem Hex-Editor auf die Spur kommen. Auch wenn man nur den Schritten folgte, die jemand anderes detailliert herausgefunden hatte – es war doch ein Gefühl des ultimativen Hackens (oder Crackens, sollte ich sagen).

Der lustige Teil der Geschichte ist, dass ich einmal Spiele erfolgreich gecrackt habe, so dass sie ordnungsgemäß auf meiner Offline-Maschine funktionierten, ich sie aber nur selten gespielt habe. Das Cracken war aufregender als das Spiel selber, es ans Laufen zu bekommen, war das eigentliche Spiel.



Braid ist ein Meisterwerk, das ungenutzte Teile meines Gehirns aktivierte. Konzepte wie die Laufzeitverzögerung wurden nie so schön dargestellt.



Wenn Sie *Portal* nicht kennen, sollten Sie es kennenlernen. Es ist ein wunderbar kniffliges Spiel, das Sie jeden Morgen anders in den Spiegel schauen lässt.



In *Fez* gibt es Puzzles zu lösen und den Weg im Wechsel zwischen 2D- und 3D-Perspektive zu finden. Haben Sie schon gemacht? Ok, aber nicht so!

Play Harder

Aber hey, was ich eigentlich sagen wollte ist: Es war nicht völlig umsonst! Ich wollte meinen kollegialen Computerfreaks schon immer mal ein paar Gaming-Titel empfehlen. Wenn Sie immer noch gerne Videospiele spielen und Sie bereit für etwas Gehirntuning sind: Ich habe einige liebgewonnenen Games in einer kurzen Liste zusammengestellt. Nein, es handelt sich nicht um „The Incredible Machine“ oder bloße Puzzle-Spiele, die wie Schaltungs- oder Platinendesigns aussehen. Wenn Sie weitere Screenshots, größere Bilder, aktive Links sehen möchten ... oder einfach diesen Artikel kommentieren und mit Freunden teilen wollen, dann navigieren Sie zu [\[http://po.st/playharder\]](http://po.st/playharder). ◀

Elektronische Bauklötze

Rapid (e-)Prototyping mit Bricks und Bricklets von TinkerForge

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

Der Bau von Prototypen ist auch für komplexe Steuerungen schnell erledigt, wenn man vorgefertigte Module verwendet. Heute existieren aber so viele modulare Systeme, dass es schwer ist, den Überblick zu bewahren. Möglicherweise ist der Ansatz mit Bricks & Bricklets genau das Passende für Sie? Hier unsere Erfahrungen damit.



Das Test-System für diesen Artikel bestand aus dem Motortreiber-Modul *DC Brick* und den beiden Bricklets *Pushbutton* und *Rotary Poti* sowie dem *RED Brick*, einem *Master Brick* und der *Ethernet Extension*.

Rapid Prototyping scheint ein boomender Markt zu sein, wenn man die Anzahl von Tools betrachtet, die es schon gibt oder die gerade auf den Markt drängen. Dabei stehen preiswerte Ansätze für das Hobby & Lernen (wie Arduino) neben hochprofessionellen Lösungen (entwickelt von den Größen der Industrie für die Industrie). Es ist also wahrscheinlich, dass eines dieser Systeme genau zu Ihren Anforderungen passt. Alle Lösungen haben schließlich das gleiche Ziel: die Vereinfachung des Prototypenbaus!

Eines der neuesten Prototypen-Systeme auf meinem Tisch stammt von TinkerForge [1], einer aufstrebenden deutschen Firma. Auch wenn ich nicht leicht zu beeindrucken bin, muss

ich doch zugeben, dass mich die Qualität der vor mir liegenden Module beeindruckt hat.

Wer, wie, was?

Die Module von TinkerForge sind eine Art elektronische Bausteine, die man aufeinander stecken und zu einem komplexen System verdrahten kann. Die Module selbst machen erst einmal gar nichts. Es ist Sache des Anwenders, sie mit Code zum Leben zu erwecken. Als Software dient eine PC-Applikation, die in vielen Sprachen (von C/C++ usw.) und sogar LabVIEW oder Mathematica geschrieben werden kann. Die Module gibt es in der Form von Bricks, Bricklets, Erweiterungen und Zubehör. Ein

Brick besteht aus einem Mikrocontroller-Modul mit Spezialteil (z.B. Motortreiber). Ein Bricklet enthält entweder einen Sensor (z.B. ein Poti) oder einen Aktor (z.B. eine LED). Ein typisches System besteht also aus mindestens einem Brick, zu dem man Bricklets hinzufügen kann. Die PC-Applikation kommuniziert mit dem Brick, schickt Befehle zu ihm und holt Daten von ihm. Ein Brick kommuniziert mit angeschlossenen Bricklets.

Ein PC ist obligatorisch, denn auf dem PC läuft die Anwendungs-Software, nicht auf dem Brick. Außer man investiert in einen so genannten RED Brick; der kann nämlich den PC ersetzen. Dazu später mehr.

Ein Brick kann mit einer begrenzten Anzahl an Bricklets Verbindung aufnehmen – wieviel genau, das hängt vom Brick ab. Bricks können aufeinander gesteckt werden, um noch mehr Bricklets anzusprechen. Bricklets hingegen können nicht gestapelt werden. Sie werden mit Kabeln angeschlossen.

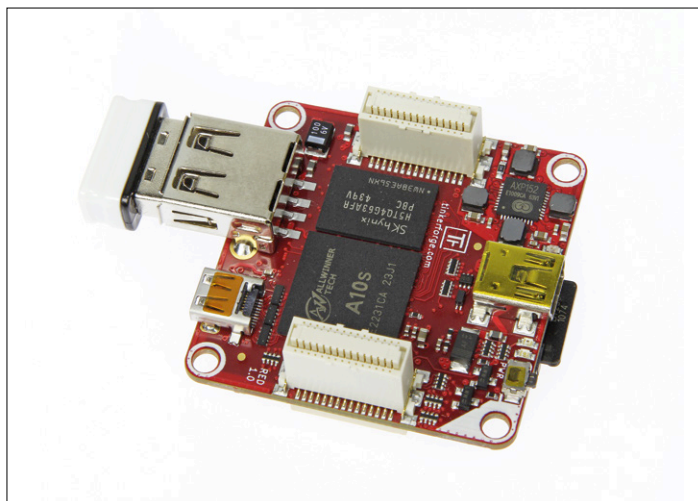
Erweiterungsmodule erlauben z.B. die Verbindung mit Brick-Stapeln und deren Steuerung über ein Netzwerk. Das ist nützlich,

Weshalb? Warum?

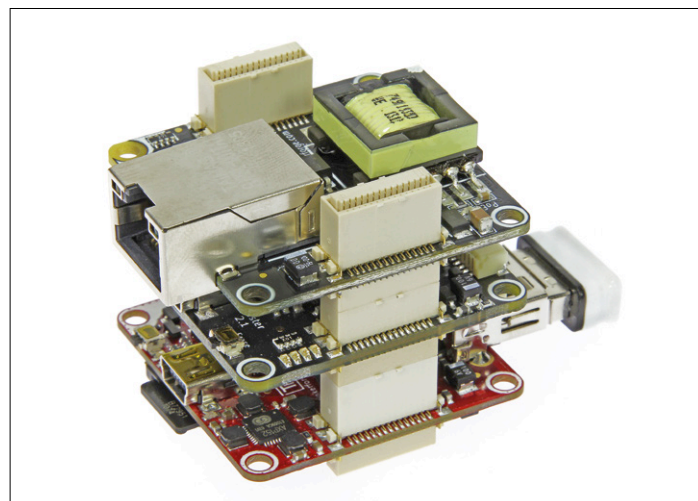
Das klärt sich, wenn man mit den Bausteinen herumspielt. Nachfolgend meine Erfahrungen.

Zunächst las ich das Tutorial *First Steps* von der Website. Es arbeitet mit dem DC Brick – ich also auch. Bei diesem Brick handelt es sich nicht um ein Netzteil für einen Brick-Stapel, wie man meinen könnte, sondern um einen Brick mit einem Treiber für DC-Motoren. Die Verbindung zum PC erfolgt über einen Mini-USB-Anschluss. Der DC-Motor kommt an die (nicht gerade leichtgängigen) Federkontakte. Anschließend muss man noch Software installieren:

Brick Daemon (brickd) - ein kleines Hintergrund-Programm, das die Kommunikation zwischen Brick und Software auf dem PC abhandelt
Brick Viewer (brickv) - ein prima Tool zur Konfiguration des Systems, das unter anderem zeigt, welches Bricklet mit welchem Brick verbunden ist. Es bietet auch Interfaces zur Steuerung von Bricks und Bricklets.



RED Brick mit WLAN-Dongle und SD-Karte.



Stapel: RED Brick (unten), Master Brick (mittig) und Ethernet-Extension (oben).

wenn das zu steuernde System entfernt oder nicht zugänglich ist. Unter Zubehör laufen Steckverbinder, Muttern und Schrauben, Displays etc. - was man eben so zur Komplettierung eines Systems benötigt. Diese Teile kann man auch aus anderen Quellen beziehen.

Wofür?

TinkerForge-Module eignen sich für Projekte, die irgendwie Elektronik benötigen, wo aber die Zeit, das Können oder die Lust fehlt, eigene Elektronik zu entwickeln. Solche Leute nennt man neuerdings „Maker“ oder eben „Tinkerer“ - daher der Name TinkerForge. Auch wenn man als Anwender nicht unbedingt tiefes Elektronikwissen haben muss, sollte man doch halbwegs programmieren können. Auf alle Fälle gibt es detaillierte Tutorials für den Einstieg.

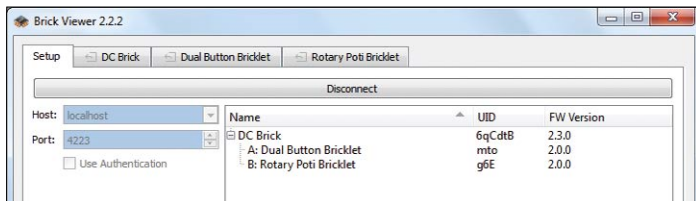
Mögliche Programmiersprachen sind C/C++, C#, Delphi/Lazarus, Java, JavaScript, LabVIEW, Mathematica, MATLAB/Octave, Perl, PHP, Python, Ruby, Shell und Visual Basic .NET. C/C++ eignet sich für iOS, C# für Windows-Phones und Java für Android. Es dürfte kaum Produkte geben, die mehr Sprachen unterstützen.

Der Brick-Dämon startet automatisch, sobald ein Brick angeschlossen wird. Den Brick Viewer muss man manuell starten. Ein Klick auf *Connect* und die Verbindung zum Brick steht. Nun sollten die Bricks im Baumdiagramm auftauchen, samt einem Reiter für das User-Interface einer Demo-Software. Im Viewer kann man die UID des Bricks sehen. Man braucht sie, wenn man dafür eigene Software schreiben will. Mit ihr adressiert man ein Objekt des Systems.

Mit dem User-Interface im Viewer konnte ich die Geschwindigkeit und die Drehrichtung eines kleinen Motors einstellen. Nun probiert man aus, ob man das gleiche Verhalten auch mit eigener Software erreichen kann. Im Folgenden wird vorausgesetzt, dass Sie die Grundlagen der gewählten Programmiersprache ausreichend beherrschen, und dass notwendige Compiler oder Interpreter bereits installiert sind.

Los geht es mit der Installation der API-Bindings der jeweiligen Programmiersprache. Detaillierte Anleitungen befinden sich für jede Sprache auf der TF-Webseite. Meine Wahl war Python 2.7, obwohl auch Python 3 möglich ist.

Um zu testen ob alles läuft, lädt man sich ein paar Beispiele



Der Brick Viewer zeigt die Verbindung des DC Bricks mit zwei Bricklets. Ihre UIDs benötigt man für die Programmierung.

für die jeweilige Sprache herunter. Da meine Kombination aus DC Brick und Python bestand, wählte ich das Skript `example_Konfiguration.py`. Man muss es noch editieren und an die eigenen Bedingungen anpassen. Hierzu benötigt man den Brick Viewer. Man öffnet das Skript in einem Editor und passt die Parameter HOST, PORT und UID an die im Viewer zu sehenden Werte an. HOST und PORT stimmen vermutlich schon – nicht aber die UID, denn diese ist individuell pro Brick. Also ändern und die Datei sichern. Wenn man dann das Skript laufen lässt, sollte sich der Motor drehen. Bei einem Druck auf die Eingabetaste stoppt das Skript, aber der Motor läuft weiter. Man kann ihn manuell anhalten, indem man folgende Python-Befehle ausführt:

```
>>> ipcon = IPConnection()
>>> dc = DC("6qCdtB", ipcon)
>>> ipcon.connect("localhost", 4223)
>>> dc.set_velocity(0)
```

In den ersten drei Zeilen stehen die Schritte, die zum Aufbau einer Verbindung zu einem Brick nötig sind (es geht um die Parameter HOST, PORT und UID). Die letzte Zeile stoppt den Motor. Zum Abschluss muss die Verbindung wieder getrennt werden:

```
>>> ipcon.disconnect()
```

Noch ein Hinweis zum Experimentieren mit dem DC Brick: Man muss den Motor erst aktivieren, bevor irgendetwas passieren kann. Hierzu dient der Befehl `dc.enable()`. Wenn Sie an diesem Punkt angelangt sind, können Sie auch die restlichen Produkte von TinkerForge ausprobieren.

Bricklets

Es gibt ganz schön viele Bricklets, und zukünftig wird es sogar noch mehr geben. Beispiele sind analoge Ein- und Ausgänge, Barometer, Entfernungsmesser, GPS, IO16, Joystick und viele andere Bricklets. Wie schon gesagt, können Bricklets nur und ausschließlich an Bricks angeschlossen werden. Eine Ausnahme ist der RED Brick: Er akzeptiert keine Bricklets. An einen Master-Brick können bis zu vier Bricklets angeschlossen werden, ein DC Brick verträgt nur zwei. Die Verbindung zwischen Brick und Bricklet erfolgt über ein zehnpoliges Kabel, das nicht so einfach selbst zu machen ist. Man kauft es besser fertig. Es gibt sie in unterschiedlichen Längen. Bevor Bricklets angeschlossen oder entfernt werden, sollte man die Brick-Stromversorgung trennen. Ich koppelte die Bricklets „Rotary Poti“ (schönes Denglisch :-)) und „Dual Button“ an meinen DC Brick. Nach Einschalten konnte man die Bricklets im Viewer unter dem DC Brick sehen. Für jedes Bricklet gibt es einen eigenen Reiter. Im Poti-Reiter sieht man eine Grafik mit der Stellung (in Grad) und beim ande-

ren Bricklet kann man im Reiter die LEDs einschalten und die Funktion konfigurieren. Sehr nett das.

Mit Hilfe der Beispiele ist es sehr einfach, eine Anwendung zu erstellen, mit der man die Motorgeschwindigkeit über den Poti einstellen und den Motor über die beiden Taster ein- und ausschalten kann. Die LEDs im Taster geben ein optisches Feedback. Merkwürdigerweise funktionierten dann aber bei meinem Programm die Taster genau verkehrt herum (oder ich habe einen Fehler eingebaut). Mein Beispiel-Skript kann man von der Webseite zu diesem Artikel [2] downloaden.

Erweiterungen

Mit Bricks und Bricklets kann man beispielsweise einen Motor steuern. Das Ganze wird dann an einen PC angeschlossen, auf dem der Brick-Dämon und die eigentliche Applikation laufen. Doch wie steuert man Hardware, die nicht in der Nähe des PCs platziert ist? Hier kommen Erweiterungen in Form der Tinkerforge-Extensions zum Zuge. Diese Module können mit zwei Stacks kommunizieren, und am PC sieht das dann aus, als wäre die Hardware direkt angeschlossen. Momentan gibt es vier Extensions: Ethernet, WLAN (Wi-Fi), RS485 und Chibi (ein leider nicht sehr bekannter Open-Source-Funk-Standard nach 802.15.4).

Extensions benötigen aber einen Master Brick – ein DC Brick reicht nicht. Doch das ist kein prinzipielles, sondern nur ein Kostenproblem. Für die Fernsteuerung eines DC Bricks via *Ethernet Extension* braucht man also noch einen zusätzlichen Master Brick. Ich kombinierte also die Ethernet Extension mit einem Master Brick und auch wenn alles gut aussah, so blieb mir doch verborgen, wie das Ganze funktioniert. Laut TF-Webseite kann man ein solches System sogar per Smartphone oder Tablet ganz ohne Dämon-Software steuern – bloß wird nicht angegeben, wie das gehen soll. Es gibt keine Doku dazu und auch keine Beispiele.

Mit einem kleinen Trick fand ich die IP-Adresse der Extension im DHCP-Mode heraus, bei dem ein Server die Adresse vergibt. Wenn man die Extension kurz auf „Static IP“ umschaltet, werden die aktuellen Werte angezeigt. Man notiert diese und schaltet zurück auf DHCP, ohne die Konfiguration zwischendurch zu sichern.

RED Brick

Dass ein Brick einen PC benötigt, auf dem die Applikation läuft, ist für manche Zwecke ungeeignet. Deshalb gibt es es von TinkerForge den RED = *Rapid Embedded Development* Brick. Es handelt sich dabei um einen kleinen Linux-Computer auf der Basis von Debian, der Applikationen der meisten unterstützten Programmiersprachen akzeptiert. Ausgeschlossen sind nur LabVIEW, Mathematica und MATLAB. Das Schreiben, Debugging und Testen der Applikation geschieht auch hier auf dem PC. Wenn sie fertig ist, wird sie auf den RED Brick übertragen und dort ohne Änderungen ausgeführt. Die Programmausführung kann man sogar planen (beim Booten; jede Stunde etc.) und überwachen. Es ist auch möglich, mehrere Programme gleichzeitig auszuführen.

Wenn Sie sich je mit dem Raspberry Pi oder einem anderen Embedded-Linux-System beschäftigt haben, dann kennen Sie die Komplexität samt der Frustration bei der Konfiguration. TinkerForge hat das Ganze vereinfacht, indem für den RED Brick eine umfangreiche Konfiguration via Brick Viewer eingebaut

wurde. Dank dieses GUI ist es simpel, eine WLAN-Verbindung aufzusetzen (man muss nur einen USB-WLAN-Dongle anstecken). Man sieht damit gut, was sich bei diesem Brick tut. Für Mausverächter gibt es natürlich auch die Kommandozeile, mit der Linux-Befehle eingegeben werden können. Man kann das System auch runterfahren oder neu starten, aber an einer weniger offensichtlichen Stelle: Mit dem System-Taster in der rechten oberen Ecke des Tabs für den RED Brick. Jedes Linux-Board sollte so einfach zu konfigurieren sein!

Wenn der RED Brick via WLAN oder Ethernet mit dem Netzwerk verbunden ist und die IP-Adresse klar ist, dann kann man sich in das Web-Interface des RED Brick einloggen und die hier dargebotenen Infos zu den installierten Programmen betrachten. Man kann sich natürlich auch ein eigenes Interface schreiben, denn HTML/JavaScript, Python und PHP werden unterstützt. Der RED Brick verfügt auch über ein HDMI-Interface. Wenn man ein passendes Display anschließt, erscheint beim Booten eine GUI. Meine Testversion wurde mit einem Touchscreen geliefert, doch das Steuern per Berührung funktionierte nicht, obwohl die LED auf dem Display signalisierte, dass alles korrekt angeschlossen war. Möglicherweise war das eine Frage der Konfiguration irgendwo.

Community

Wenn man anfängt, sich mit Bricks und Bricklets zu beschäftigen, empfiehlt sich der regelmäßige Besuch des Diskussionsforums [3] (in Deutsch und Englisch), denn hier werden Software- und Hardware-Updates angekündigt. Im Forum ist viel los. Als ich mit diesem Artikel begann, lud ich den Brick Viewer 2.2.2 herunter. Zwei Monate später gab es schon das Update auf die Version 2.2.5 plus eines für die Brick-Firmware.

Open Source

Alles was mit Bricks, Bricklets und Extensions zu tun hat, ist open source. Daher kommt man auch an fast allen Code heran und hat Zugriff auf alle Schaltpläne.

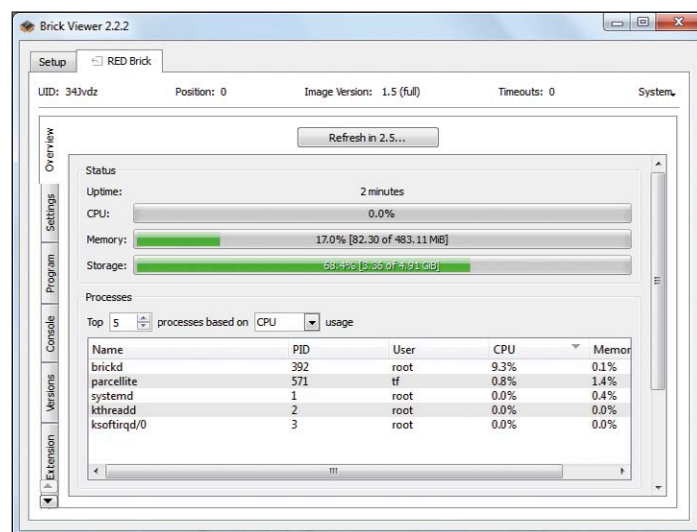
Fazit

Da nun klar ist, wie TinkerForge-Module operieren, stellt sich die Frage aller Fragen: Ist diese Lösung besser als andere? Die Antwort hängt von der Anwendung ab. Ich selbst war von diesem Rapid Prototyping System sehr beeindruckt. Es unterstützt so viele Programmiersprachen, dass jeder, der im Laufe seines Lebens einmal programmiert hat, mit Bricks & Co. zurechtkommen sollte. Außerdem gibt es eine umfangreiche gut geschriebene Dokumentation und viele Beispiele. Selbstverständlich sehe ich auch Verbesserungsmöglichkeiten, zum Beispiel beim Viewer. Und wenn es um besondere Fähigkeiten für das eigene Projekt geht, kann es eng werden. Doch da helfen dann das Forum bzw. andere User und selbst vom TinkerForge-Team gibt es Unterstützung. Und wie schon gesagt: Die Qualität der Hardware ist wirklich exzellent. ◀

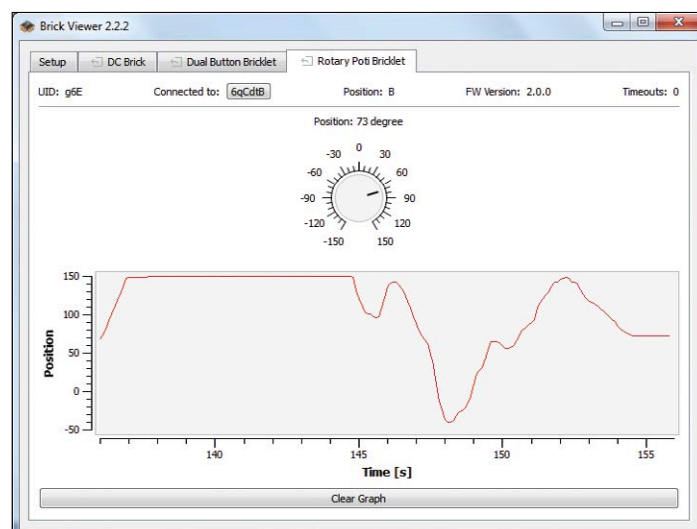
(140553)

Weblinks

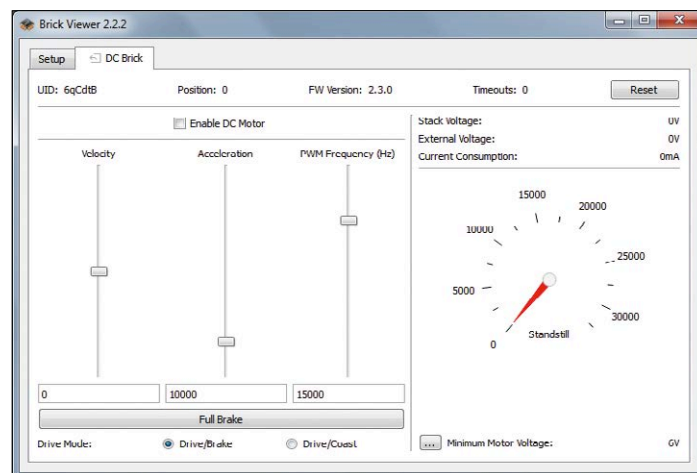
- [1] www.tinkerforge.com
- [2] www.elektormagazine.de/140553
- [3] www.tinkerunity.org/forum/index.php



Das Interface zur Linux-Konfiguration für den RED Brick im Viewer. In der Liste aktiver Programme sieht man „brickd“, den Brick-Dämon.



Hier sieht man die Position des Poti-Bricklets.



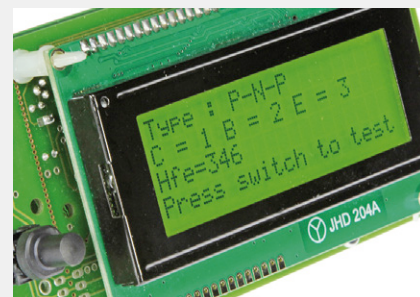
Das Interface für den DC Brick im Viewer erlaubt die Einstellung von Geschwindigkeit und Drehrichtung des Motors.

Platino-Transistortester

Elektor 3/2015, S. 60...65 (130544)

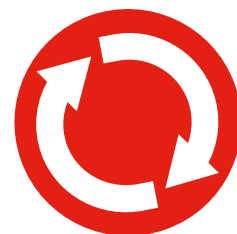
NACHTRAG. Im Artikel wurden nur knappe Angaben zur Funktion von Transistor T1 gemacht. Nach dem Aufbau laut Schaltung hat dieses Bauteil keine besondere Bedeutung für den Anwender. T1 dient zur Selbstkalibrierung via Portleitung PA3. Die konkrete Funktion ist im Sourcecode ersichtlich, der in BASCOM-AVR geschrieben wurde. Es wird auch nicht näher auf die Frequenz des Quarz-Oszillators auf dem Platino-Board eingegangen. Die Frequenz ist 8 MHz (steht ebenfalls im Sourcecode).

[www.elektormagazine.de/130544]



Projekt 2.0

Korrekturen, Updates und Nachträge zu veröffentlichten Artikeln



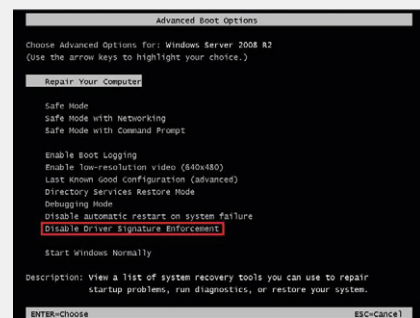
Tippfehler, richtige Irrtümer und Nachträge – dieses Mal sogar historische Verwechslungen! Nach Murphy geht bekanntlich schief, was schief gehen kann. Auch wenn wir stets sehr bemüht sind, einwandfreie Schaltungen zu präsentieren, ist der Fehlerteufel damit nicht ganz einverstanden. Glücklicherweise haben wir Leser mit Adleraugen (also Sie!), die alles merken. Ein großes Dankeschön für Ihre aufmerksamen Beiträge!

UART/RS232-Datenlogger

Elektor 4/2015, S. 38...43 (140126)

UPDATE. Unter Windows 8 und neueren Versionen kann es notwendig sein, dass man den USB-Treiber für den Datenlogger installieren muss. Schön, aber DSE, das „Driver Signature Enforcement“ von Windows, zickt dann, da es nicht signierte Treiber blockiert. Als Work-Around sollte man DSE deaktivieren, was aber gar nicht so einfach ist. Glücklicherweise gibt es dafür viele Anleitungen im Internet, die das Vorgehen erläutern. Berücksichtigen Sie also den nachfolgend verlinkten Patch. Mit deaktiviertem DSE wird der Treiber problemlos installiert.

[<http://po.st/disableDSE>]



J2B-Synthesizer: Eine offene digitale Musikplattform

Elektor 1/2015, S. 44...55 (140182)

UPDATE. Der Leiter des Elektor-Labors Clemens Valens hat die Software des erfolgreichen J2B-Synthesizers erweitert. Jetzt sind drei neue Firmware-Versionen verfügbar. Eine davon verwandelt den Synthesizer in eine (rudimentäre) Drum Machine. Es handelt sich dabei um Portierungen von Soulsbys Atmegadrum, Atcyclotron und Delayertron. Zusammen mit der originalen Atmegatron-Firmware verwandelt sich das Instrument nun in gleich vier besondere Synthesizer. Man kann die Archiv-Datei mit allen vier Firmware-Files von der originalen Projektseite bei Elektor.Labs oder über den nachfolgenden Link herunterladen.

[<http://po.st/J2Bv3>]



USB-Hub/UART-Konverter mit RS232/422/485

Elektor 11/2014, S. 22...32 (140033)

KORREKTUR. In der abgedruckten Schaltung wurden bei IC2, dem Spannungsregler LM2937, die Pins 2 und 3 vertauscht. Richtig ist: 1 = Eingang, 2 = GND und 3 = Ausgang. Die verbesserte Schaltung kann via nachfolgendem Link heruntergeladen werden. Der Fehler ist in den Platinen nicht enthalten und auch nicht in den anderen Dateien.

[<http://po.st/140033schematic>]



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

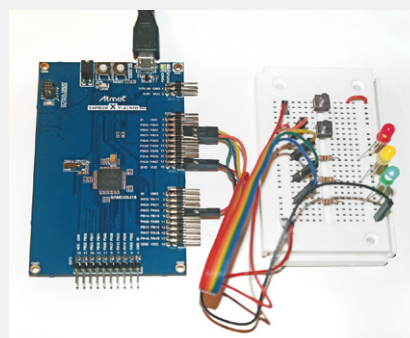
Elektor 4/2015, S. 10...17 (150041)

KORREKTUR. Ein simpler Buchstabe kann bekanntlich einen großen Unterschied machen. So auch hier, bei Listing 1: In den Konfigurationsfunktionen für WDT und EIC finden sich zwei identische Befehle (beide im zweiten Teil des Codes, mit wenigen Zeilen dazwischen). Leider ist bei beiden der Buchstabe für den Port falsch: „A“ ist falsch. Richtig ist „B“. Die Korrektur lautet daher:

```
config_extint_chan_gpio_pin_mux = MUX_PBxxA_EIC_EXTINTx;
```

Die „x“ stehen dabei für Pin-Nummer und Kanal-Nummer. Statt die Zeilen zu suchen und händisch zu korrigieren können Sie sich auch ganz einfach den korrigierten Code von der Webseite zu diesem Artikel holen. Dieser Fehler wurde vom Autor entdeckt.

[www.elektormagazine.de/150041]



Serieller USB-nach-Multi-Protokoll-Konverter

Elektor 4/2015, S. 70...76 (130542)

KORREKTUR. Im Kasten „Was ist überhaupt eine serielle Schnittstelle?“ steht: „Für die Verbindung der Terminals mit dem Mainframe-Rechner über eine gewisse Distanz wurde schließlich der bekannte RS232-Standard eingeführt.“

Tatsächlich aber wurde RS232-C ursprünglich als Interface zwischen DTE (Data Terminal Equipment, also Endgeräten wie Terminals, Mainframes, Servers etc.) und DCE (Data Kommunikation Equipment, also Datensendern wie z.B. Modems) eingeführt. In der Folge war es damit dann auch möglich, zwei DTEs direkt über ein Kabel mit gekreuzten Datenleitungen zu verbinden. Ein besonderes Dankeschön für diese historische Richtigstellung geht an Ernst Stippl aus Wien.



AVR-Port toggeln (Rubrik Tipps & Tricks)

Elektor 4/2015, S. 23 (150027)

NACHTRAG. Der Trick ist schon älter, denn schon manche der ersten AVR-Mikrocontroller vor 2005 konnten das. Wenn man im Datenblatt zum ATtiny13 schürft, kann man ihn bei „I/O Ports“ unter „Ports as General Digital I/O“ im Abschnitt „Toggling the Pin“ (vor dem Abschnitt „Reading the Pin Value“) finden. Die Pin-Register finden sich für gewöhnlich unter den ersten 32 I/O-Registern (Adresse < 0x20), weshalb man sie auf Bit-Ebene ansprechen kann. Der Ausdruck „PINB.5=1“ im Code wird von BASCOM in einen SBI-Befehl verwandelt. Wenn der Flash-Speicher eines AVR-Controllers größer als 8 KB ist, demnach mit einer zwei Worte breiten IVT (Interrupt Vector Table), ist es z.B. möglich, Ports direkt zu toggeln, z.B. durch Interrupt in der IVT. Ist der Pin als Eingang konfiguriert, kann der interne Pullup-Widerstand durch die gleiche Prozedur aktiviert und deaktiviert werden. Dennoch konnten in einigen AVR-Serien (in der Zeit von 2007 bis 2010) die Pullups nicht über die Data Direction Register oder die Port Register gesetzt werden, doch verfügten diese über eigene Pullup Enable Register.

Wenn ein Pin wegen einer anderen Funktion (wie z.B. Reset) nicht direkt genutzt werden kann, bleibt er doch zugänglich (funktioniert zum Beispiel mit ATtiny4/5/9/10). Man kann dann immer noch bitweise XOR-Operationen durchführen, um z.B. ein Parity-Bit zu berechnen oder zu testen.

Ausführlichen Dank an Stefan Rosenthal für diese Ausführungen!

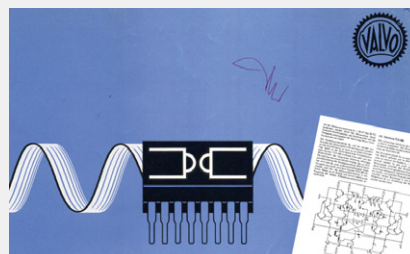


Integrierter Gyrator TCA580

Elektor 4/2015, S. 28 (150024)

NACHTRAG. Christoph Kessler, ein aufmerksamer Leser, war im Besitz nicht nur der deutschen Application Note für das IC TCA580, sondern auch vom zugehörigen fünfseitigen Datenblatt. Er hat diesen Fund eingescannt und uns freundlicherweise zukommen lassen. Veröffentlichungsdatum war das Jahr 1974 – richtig nostalgisch also. Beide Dateien warten unter dem nachfolgenden Link auf ihren Download. Herzlichen Dank!

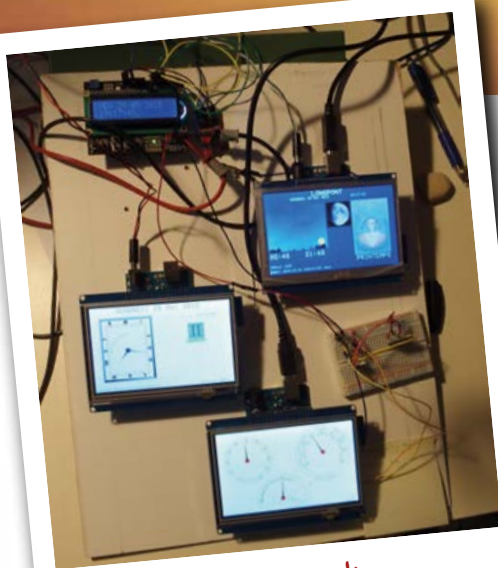
[<http://po.st/TCA580DE>]



Das war ein heißer, heißer Sommer

Im Labor, mindestens ...

Auch wenn wir nicht wissen, ob der diesjährige Sommer auf der ganzen Welt so schön warm war wie bei uns, herrschte, was die LötKolben und die Mitglieder von Elektor-Labs betrifft, eine Bullenhitze. Um das zu beweisen, kommt hier eine Übersicht der Projekte, die auf Elektor-Labs.com entwickelt wurden und schon prächtig aussehen. Deshalb stellt sich die Frage: Wird der Herbst noch heißer? Wer weiß!



<http://po.st/astroclock>

Zeitzeichen

Diese astronomische Uhr bietet eine Unmenge an Informationen auf der Grundlage aktueller Positionskoordinaten und zeigt alles auf nicht weniger als vier Displays! Greenwich-Zeit, Sternzeichen, Zeit, Tag, Ort, Jahreszeit, Luftfeuchtigkeit, Temperatur, Luftdruck ... die Uhr ändert selbst ihr Aussehen je nach Tageszeit. Kontroll-Freaks - diese Uhr ist was für euch!

On the drums: Unser neuer analoger Schlagzeuger

Klar, Software für die Musikproduktion ist bequem. Aber Schieberegler auf dem Bildschirm haben halt nicht die Haptik von echten Knöpfen, mit denen man Beats einstellt. Diese Beatbox besitzt einen Sequencer mit 16..32 Steps und eine zusätzliche MIDI-Steuerung. Die zwölf verschiedenen Sounds werden von einer analogen Quelle generiert. Ein Gefühl der Nostalgie der 80er Jahre?



<http://po.st/beatbox>



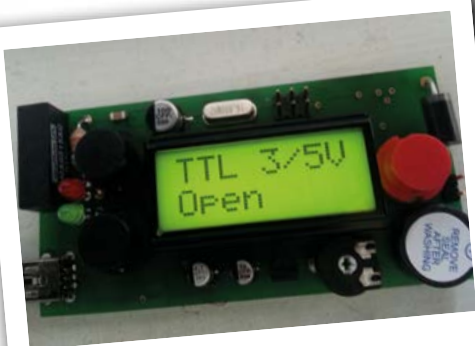
<http://po.st/levelsensor>

Vertraut meinem Pegel!

Sind Sie auf der Suche nach einem vertrauenswürdigen Füllstandssensor zur Pegelüberwachung in einem Brunnen oder Flüssigkeitstank? Dieser Entwurf besteht aus einem Signalgeber und drei „Empfangsstationen“, an die Sonden (hoch, niedrig, und kritisches Niveau/Alarm) angeschlossen werden. Die Schaltung kann auch einen Pumpensumpf überwachen, um eine Pumpe automatisch zu aktivieren und so eine Kellerflutung zu vermeiden. Die Schaltung ist skalierbar; der erste Teil kann auch alleine arbeiten.

Geh nie über deine Grenzen, USB!

Diese USB-Ports muss man wirklich immer unter Kontrolle haben... und dafür eignet sich ein Gerät wie dieses hervorragend. Es misst und begrenzt den Strom zweier Ports ganz nach den Wünschen beziehungsweise Einstellungen des Benutzers. So wird ein zu hoher Strom sicher vermieden. Echtzeit-Werte werden auf einem Display angezeigt, stehen aber auch über einen FTDI USB-zu-TTL-5V-Port für weitere Analysen zur Verfügung.

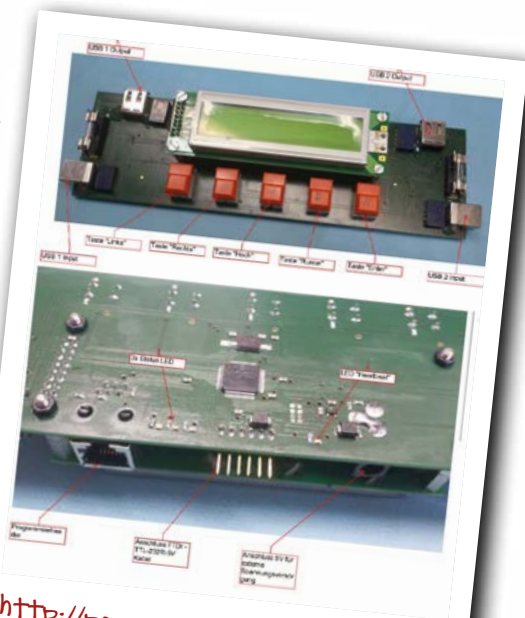


<http://po.st/19inabox>

Ein elektronisches Schweizermesser

Wenn Sie mehrere Funk-
rollen ... warum entwerfen

Sie nicht ein Multi-Tool, das sie alle umfasst? Jawohl, das ist der Vorteil kundenspezifischer Elektronik! Erinnern Sie sich an das 13-in-a-box-Projekt (1/2013)? Nun, den 13 wurden jetzt sechs neue Funktionen hinzugefügt, ein Summer, Spannungsmessung, Servosteuerung, DCF77-Dekodierung, Neopixel-LED-Steuerung und DS18B20-Temperaturmessung. Und das Helferlein wächst und wächst und wächst ...



<http://po.st/USBcontrol>

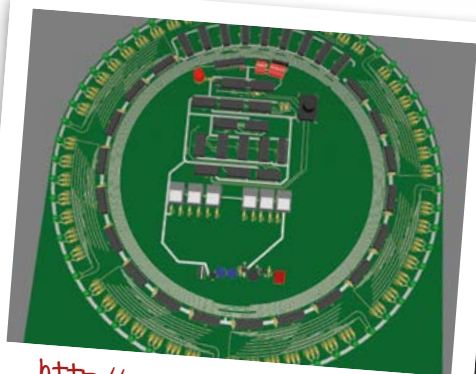
All You Need Is ... 8 Transistoren

Ein AM-Radio zu bauen bringt einen Haufen Spaß. Dieses Projekt beschreibt die Montage und den Test eines superheterodyn AM-Radios, kurz Superhets: Es ist ein hervorragender Einstieg in die hochfrequente Welt! Die ersten vier Transistoren umfassen die Empfangsstufen inklusive Lokaloszillator LO, die anderen vier bilden einen einfachen dreistufigen Verstärker. Die Schaltung

arbeitet mit 6 V, so dass vier AA-Batterien zur Versorgung ausreichen. Damit ist das Radio sogar portabel!



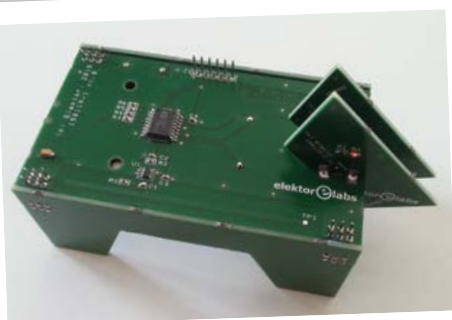
<http://po.st/8Qradio>



<http://po.st/TTLclock>

Von der digitalen Uhr, die analog sein wollte

Dies ist die bewegende Geschichte einer digitalen Uhr, die gerne analog sein wollte, aber nur TTL-ICs in ihrem Uhrkasten hatte. Statt der erwarteten Stunden- und Minutenzeiger verfügt sie über einen Kreis von LEDs. Ein Controller? Nein danke, zu leicht, zu Mainstream! Aber das heißt nicht, dass die Uhr nicht vom Benutzer einstellbar wäre. Denn mit einigen DIP-Schaltern kann das LED-Muster systematisch verändert werden. Eine Digitaluhr ohne Mikrocontroller und mit einer analogen Seele .. 100% cool!



<http://po.st/hacktor>

Die den Mond anbellen

Erinnern Sie sich an Hacktor, die Platine, die sich in einen Hund verwandelt? Nun kann dieses Tierchen auch bellen! Hacktor verfügt über einen QTouch-Näherungssensor von Atmel, der erkennt, wenn jemand in der Nähe ist. Ein Audio-Wiedergabe-IC lässt dann ein Bellen im Lautsprecher hören, der sich im Bauch des Hundes versteckt. Und die LED-Augen blitzen gefährlich. Natürlich wurde Hacktor geschaffen, um zu hacken. Das Board kann über einen Onboard-SPI-Anschluss programmiert und einfach in eigene Projekte eingebunden werden. Außerdem: Hacktor sieht echt gut aus auf Ihrem Schreibtisch, finden Sie nicht? ◀

Gerber-Dateien aus dem Elektor-Labor

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

Versuchsweise sind einige „Gerber“-Platinendateien im Elektor-Shop erhältlich. Was kann man damit machen und wie funktioniert das Ganze?

Gerber-Dateien sind eine Gruppe von Dokumenten, die beschreibt, wie eine Platine hergestellt werden soll. Sie enthält die verschiedenen Layer der Leiterplatte, die untere und obere Kupferlage, die Siebdruck-Lage, die Lötmaske, den Bohrplan und Informationen zu allen weiteren denkbaren Layern. Die Daten werden in offenen binären 2D-Vektorgrafiken gespeichert. Derzeit ist „Extended Gerber“ oder „RS-274X“ Industrie-Standard. Es existiert auch ein „Standard Gerber“-Format, das allerdings abgekündigt ist und deshalb vermieden werden sollte.

Das Gerber-Dateiformat wurde ursprünglich von Gerber Systems, einer Abteilung der von Joseph Gerber gegründeten Firma Gerber Scientific, in den 1960er und 1970er Jahren entwickelt, um einen Vektor-Foto-Plotter für die Leiterplattenindustrie zu betreiben. Da Gerber seinerzeit Marktführer war, wurde das Format ein De-facto-Standard für die Branche. Und das ist immer noch so.

Mit der Entscheidung, allen Elektor-Lesern über den Online-Shop (www.elektor.com) Zugang zu einem Teil der Dateien des Elektor-Labors zu gewähren, hoffen wir, Sie anzuregen, noch mehr Elektor-Projekte zu realisieren und damit die boomende

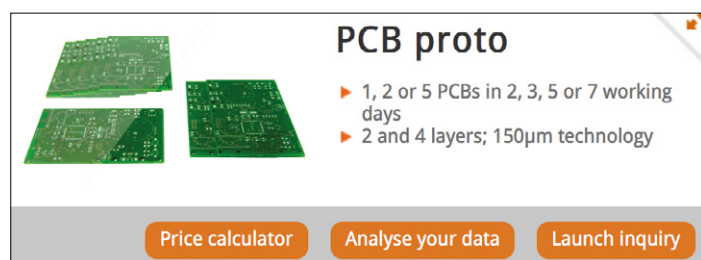
Maker-Community zu unterstützen. Mit den Gerber-Dateien aus unserem Shop können Sie nun Elektor-Platinen bei einem Platinenhersteller Ihrer Wahl herstellen lassen, problemlos und professionell. Schließlich verwenden wir den gleichen Satz von Dateien für unsere eigenen Prototypen und unsere Produktion. Wichtig zu beachten ist: Der neue Gerber-Datei-Service umfasst nicht alle in Elektor veröffentlichten Projekte und wird dies auch in Zukunft nicht tun. Die Auswahl der Platinen nimmt das Labor vor. Jetzt, da ich dies schreibe, wird der Service gerade ausgearbeitet. Am besten, Sie suchen nach „Gerber“ auf der Shop-Webseite, dann werden Sie eine aktuelle Liste mit allen lieferbaren Dateipaketen erhalten.

Da es sehr viele Platinenhersteller gibt, ist es unmöglich aufzuzeigen, wie eine Bestellung jeweils funktioniert. Exemplarisch leiten wir Sie durch den Bestellvorgang bei unserem Haus- und Hoflieferanten Eurocircuits. Bei Ihrem bevorzugten Hersteller sollte es ähnlich ablaufen, auch wenn die Dienste von Eurocircuits recht umfangreich sind und manche Optionen Ihnen möglicherweise nicht bei anderen Firmen zur Verfügung stehen. ◀

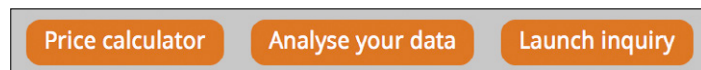
(150342)



Schritt 1. Gehen Sie zu eurocircuits.de und wählen Sie „Kalkulator“.



Schritt 2. Auf der nächsten Seite wählen Sie „Analyse your data“. Für Einzelanfertigungen können Sie auch „PCB proto“ oder „NAKED proto“ nutzen.



Schritt 3. Damit Eurocircuits Ihre Gerber-Dateien mit Ihren persönlichen Daten (wie Lieferadresse und Rechnungsstellung) kombinieren kann, müssen Sie sich registrieren oder (wenn Sie bereits registriert sind) anmelden.

Analyse your data

PCB name *

Upload data file ?

Purchase reference *

Article reference

Project reference *

Delivery term

Quantity *

* Mandatory field.

Schritt 4. Nach der Anmeldung gelangen Sie zu einem Formular, in dem Sie Ihr Projekt und die Platinen-Eigenschaften angeben und Ihre in einer ZIP-Datei komprimierten Gerberdaten hochladen.

Price calculator

Shopping basket

Checkout items

Shopping basket – To start the manufacturing process, select an item(s) and "Proceed to checkout" !

Proceed to checkout

Modify

View details

Files

Delete

History

Download PDF offer

Edit administrative details

Ask a question

Number

Item name

Service

Status

Created di

Search

Items ready for checkout

☐	PCB Visualizer®	Offer	Number	Type	Status	Item name	Service	Quantity	Delivery days	Unit price	Net price	Stencil
☒	PCB Visualizer®		B0433027	PCB	Ready to checkout	120272-1	PCB Proto	2	7 Working days	33.68	67.36	-

You can order all jobs with status 'Ready for Checkout' immediately, even when the PCB Visualizer process is still running or failed.

Items in analysis

No record(s) found.

Visualizer link in [Visualizer] column lets you view PCB layers visually, during processing you will not be able to modify your data files. .

Schritt 5. Nach dem Absenden Ihrer Daten klicken Sie auf „Continue“ und gelangen zu einer Übersicht der übermittelten Daten. Eurocircuits bietet Ihnen umfangreiche Vorschau-Optionen für Ihre Leiterplatten mit dem „PCB-Visualizer“. Mit diesem Tool können Sie eventuellen Problemchen oder Schwierigkeiten auf die Schliche kommen, die beim Produktionsprozess auftreten könnten. Nun aktivieren Sie die Entwürfe, die produziert werden sollen (Sie können mehrere Boards in einer Bestellung zusammenfassen und diese vor dem Auschecken überprüfen) und klicken „Proceed to checkout“.

Price calculator

Shopping basket

Checkout items

Checkout items

Back

Submit

Add new address

Delivery address

Elektor International Media - 1, AlleeKasteel Limbricht, Limbricht - 6141 AV, Netherlands

Invoice address

11, Postbus, Susteren - 6114 ZG, Netherlands

Elektor International Media

Elektor International Media BV

Schritt 6. Schließlich sollten Sie die Versand- und Rechnungsadresse überprüfen, dann klicken Sie auf „submit“. Ihre professionell hergestellte Platine sollte sich bald auf dem Weg zu Ihnen befinden!

The screenshot shows the PCB Visualizer v1.3-113-150701 interface. It is divided into several sections:

- Customer data:** Includes fields for PCB proto, delivery format (Single PCB), PCB quantity (2), delivery term (7 working days), PCB width (90.00 mm), PCB height (90.00 mm), eC-registration compatible PCB, and Board name (120272-1).
- Material:** Includes fields for Board thickness (1.55 mm), Material Tg (145-150 °C), Outer layer copper foil (18 µm end +/-), and Technology (PCB PDKure).
- PCB definition:** Includes fields for Top soldermask (Green), Bottom soldermask (Green), Top legend (White), Bottom legend (None), and Surface finish (Any lead free).
- Board buildup:** Shows a top view and a bird's eye view of the PCB layout.
- Summary:** Provides a detailed overview of the order, including Service (PCB proto), Delivery term (7 working days), Estimated shipment date (13-07-2015), Quantity (2 PCBs), Board surface / Order surface (0.81 dm² / 1.62 dm²), Prices (Net € 33.68, Total boards € 67.36), Express transport (€ 0.00), VAT 0.00%, and Total gross (€ 67.36).
- Alternatives:** Offers three alternative configurations for 2, 5, and 10 PCBs, each with a 7 working day delivery term and a net price of € 33.68, € 88.81, and € 124.55 respectively.

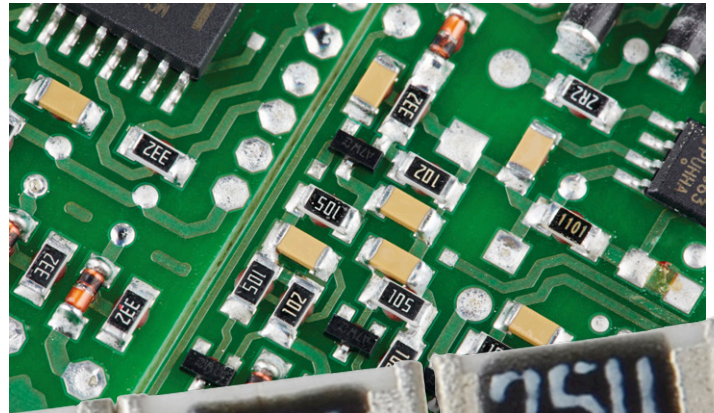
Der „PCB Visualizer“ zeigt Produktionsdetails und lässt Sie Fehler vor der Auftragserteilung erkennen und korrigieren.

SMD-Codierungen

Typenkunde für Winzlinge

Von **Harry Baggen** (Redaktion NL)

Die Miniaturisierung elektronischer Bauelemente brachte mit sich, dass anstelle konventioneller Typenbezeichnungen kurze, kryptische Codes auf winzigen Gehäusen aufgedruckt sind. Wenn es darum geht, diese Codes zu deuten, naht Hilfe aus dem World Wide Web. Wir haben uns einige Seiten angeschaut, die bei der Entschlüsselung nützlich und klärend sein können.



In elektronischen Systemen sind Bauelemente in SMD-Bauform allgegenwärtig, denn verglichen mit konventioneller Schaltungstechnik sparen sie nicht nur enorm viel Raum, sie senken auch drastisch die Produktionskosten. Bedrahtete Komponenten bietet der Markt zwar immer noch vielfältig an, doch es ist unübersehbar, dass dieses Angebot schrumpft. Allmählich gehen Halbleiterhersteller dazu über, neue Produkte ausschließlich als SMDs zu entwickeln und überkommene Bauformen aus dem Programm zu nehmen. Dem Elektronik-Spezialisten im Entwicklungslabor und dem Service-Techniker in der Werkstatt bleibt keine andere Wahl: Auch sie müssen sich mit den SMDs arrangieren, falls das nicht längst schon geschehen ist.

Als Anfang der neunziger Jahre die SMDs Einzug in die automatisierte Fertigung der Industrie hielten, hatte mancher Elektor-Leser die Vorahnung, dass das Ende seiner handwerklichen Ambitionen in der Elektronik naht. Doch das war weit gefehlt: Mit der nötigen Übung und Geduld, mit der obligatorischen Pinzette, mit angepasstem Lötgerät und leistungsstarker Lupe sind auch die Zwerge unter den Bauelementen (meistens noch) problemlos bezwingbar.

Bauelemente müssen identifizierbar sein, auch als einzelnes Exemplar, doch die Fläche für den Typenaufdruck ist auf den winzigen Gehäusen stark eingeschränkt. Die Industrie hat deshalb einen verkürzten Typencode eingeführt, der aus nur wenigen alphanumerischen Zeichen besteht. Das Problem der Umsetzung in den Klartext tritt bei verpackten SMDs kaum auf, denn die Umhüllung trägt meistens auch die herkömmliche Typenbezeichnung. Bei unsortierten oder auf Platinen verbauten SMDs ist jedoch der Typencode das einzige äußere Identifizierungsmerkmal.

Zum Glück ist Hilfe so nahe wie das World Wide Web. Wir stellen nachfolgend einige Webseiten vor, die wir zum Thema gefunden haben.

Widerstände und Kondensatoren

SMD-Widerstände tragen drei- oder vierstellige Codes, dies hängt von der Toleranz ihres Werts ab. Die einfachste Systematik ist an den Farbcode bedrahteter Widerstände angelehnt, die Farben sind durch numerische Äquivalente ersetzt. Die vorderen zwei oder drei Ziffern sind die signifikanten Werte, die Endziffer steht für den dezimalen Multiplikator. Widerstände mit der Toleranz 1 % werden nach Standard EIA-96 mit drei Zeichen kodiert, der Multiplikator ist hier mit einem Buchstaben angegeben. Die Website *Your Hobby-Hour* [1] enthält eine kurze Erklärung, eine Tabelle mit den EIA-96-Werten sowie einen *SMD resistor code calculator*, der den SMD-Aufdruck online in den zugehörigen Widerstandswert übersetzt. Auf der Website *resistorguide.com* [2] fanden wir eine ähnlich kompakt

EIA-96 system



Significant values **1**

Multiply factor **2**

Resistance = 243·**100**

1	Code	Value	Code	Value	Code	Value
01	100	17	147	33	215	
02	102	18	150	34	221	
03	105	19	154	35	226	
04	107	20	158	36	232	
05	110	21	162	37	237	
06	113	22	165	38	243	
07	115	23	169	39	249	
08	118	24	174	40	255	
09	121	25	178	41	261	
10	124	26	182	42	267	
11	127	27	187	43	274	
12	130	28	191	44	280	
13	133	29	196	45	287	
14	137	30	200	46	294	
15	140	31	205	47	301	
16	143	32	210	48	309	

2	Code	Multiply
Z	0.001	
Y/R	0.01	
X/S	0.1	
A	1	
B/H	10	
D	100	
E	1000	
F	10000	

www.resistorguide.com
your guide to the world of resistors

SMD-Codierungen:

Keine Geheimsprache, sondern Kennzeichnung von Bauelementen in Kurzformat.

gehaltene Beschreibung, ergänzt durch einen Video-Clip, ersetzt die Zusammenhänge audiovisuell in Szene.

Kondensatoren lassen sich weniger leicht identifizieren, hier kommt es auf die physische Größe an. Auf Typen hoher Kapazität wie Elkos ist meistens genügend Platz vorhanden, um die Kapazität und Arbeitsspannung im Klartext anzugeben. Kleine Kondensatoren werden häufig in gleicher Weise wie Widerstände mit drei Ziffern kodiert (siehe [3]). Gelegentlich fehlt jedoch jeder Aufdruck, was übrigens auch für kleine Induktivitäten gilt. Montierte SMDs müssen dann entlötet und gemessen werden, bei Kondensatoren bleibt allerdings die Arbeitsspannung unbekannt. Wenn die messtechnische Bestimmung eines Kondensators nicht möglich ist, kann die Gehäusefarbe (siehe [4]) einen groben Hinweis auf den Kapazitätsbereich geben.

Halbleiter

Bei nicht komplexen Halbleiter-SMDs mit zwei, drei oder auch mehr Anschlüssen (Dioden, Transistoren, Opamps, Spannungsregler, Reset-Schaltungen) ist das Identifizieren des Typs eine Kategorie schwieriger. Auf vielen SMDs dieser Gruppe sind zwei oder drei Zeichen in hervorgehobener Schrift aufgedruckt, begleitet von Zeichen in untergeordneter Schrift, die das Produktionsdatum und die Charge angeben. Da auf dem Gehäuse nur wenige Zeichen Platz haben, die Typenvielfalt aber immens ist, kann die gleiche Zeichenkombination unterschiedliche Bedeutungen haben. Auf diversen Websites stehen Übersichten und Tabellen bereit, die zu zahllosen Kombinationen die möglichen Typenbezeichnungen mit den zugehörigen Herstellern angeben. In nicht seltenen Fällen lassen sich der gleichen Zeichenkombination 20 oder mehr Halbleitertypen zuordnen. Dann muss die Bauform in ihren Details mit den Datenblättern verglichen werden, und die Schaltung, in der die Komponente verbaut ist, wird zu Rückschlüssen herangezogen. Die nachfolgend genannten Websites können bei solchen Puzzlespielen eine gute Hilfestellung leisten.

The SMD Codebook [5] ist ein 80 Seiten umfassendes PDF-Dokument, das außer einführenden Erklärungen diverse Code-Tabellen enthält. Zu jedem SMD-Code sind der mögliche Typ, der Hersteller, die Gehäusebauart sowie einige wichtige Eigenschaf-

Codes beginning with 'C'					
Code	Device	Manufacturer	Base	Package	Leaded Equivalent/Data
C	BB565	Sie	I	SCD80	uhf varicap 2-20pF
C white	BAT165	Sie	I	-	40V 750mA sw Schottky
C	KV1832E	Tok	I	URD	uhf varicap 4-17pF
C0	HSMS-2820	HP	C	SOT23	HP2835 schottky
C0	HSMS-282B	HP	C	SOT23	HP2835 schottky
C1	HSMS-2821	HP	K	SOT23	HP2835 schottky
C1	BCW29	Phi	N	SOT23	BC178A
C1p	BCW29	Phi	N	SOT23	BC178A
C1t	BCW29	Phi	N	SOT23	BC178A
C1	BFQ51C	Phi	CX	SOT173	pnp BFR90A complement
C2	BCW30	Phi	N	SOT23	BC178B
C2p	BCW30	Phi	N	SOT23	BC178B
C2t	BCW30	Phi	N	SOT23	BC178B
C2	BFQ32C	Phi	CX	SOT173	pnp 4.5GHz 15V 100mA
C2	HSMS-2822	HP	D	SOT23	dual HP2835 schottky
C2	HSMS-282C	HP	D	SOT23	dual HP2835 schottky
C2	SST112	Tem	F	SOT23	J112 analog sw n-ch jfet
C2A	ZDC833A	Zet	B	SOT23	dual cc 28V varicap 15pF @2V
C3	HSMS-2823	HP	A	SOT23	dual HP2835 schottky
C3	BFQ23C	Phi	CX	SOT173	pnp complement BFP91A
C3	SMBT4126	Sie	N	-	2N4126
C3	SST113	Tem	F	SOT23	J113 analog sw jfet
C4	BCW29R	Phi	R	SOT23R	BC178A
C4	HSMS-2824	HP	B	SOT23	dual HP2835 schottky
C5	MMBA811C5	Mot	N	-	2N5086 pnp hfe 135-270

ten angegeben. Dem *SMD Codebook* sind wir auf mehreren unterschiedlichen Websites begegnet. Eine HTML-Version nennt sich *GM4PMK's SMD Codebook* [6], wir haben sie nicht vollständig verglichen, doch inhaltlich ist sie vermutlich identisch. Eine nützliche Hilfe kann auch die SMD-Webseite von *S-Manuals* [7] sein. Unter *Marking SMD* fanden wir eine Tabelle, in der sämtliche zweistelligen Kombinationen alphanumerischer Zeichen gelistet sind, die den Anfang eines SMD-Codes bilden können. Die Links, die mit den Kombinationen verknüpft sind, führen zu Listen der in Frage kommenden Typen. Auch hier sind stets die Typenbezeichnungen, die Bauformen und die Hersteller angegeben. Ein weiterer Mausklick führt unmittelbar zum Datenblatt des potentiellen Typs.

Bei komplexeren SMD-Halbleitern mit zwei Dutzend oder mehr Anschlüssen ist so viel Mühe nicht notwendig, denn dort reicht der Platz gewöhnlich aus, um die Typenbezeichnung im Klartext aufzudrucken.

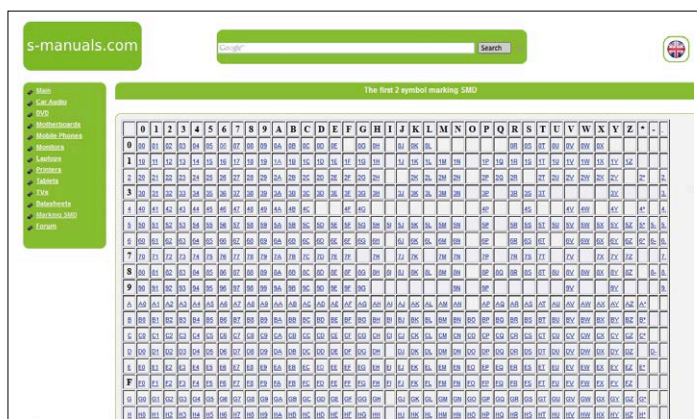
Wir wünschen Ihnen viel Intuition und Spürsinn bei der Identifizierung unbekannter SMDs!

(150339)gd

Korrektur: In der Web-Recherche des Maihefts („Stöbern und staunen bei Hackaday“) ist beim Projekt „Using the Red Pitaya as an SDR“ ein falscher Entwickler angegeben worden. Das Projekt wurde durch Pavel Demin entwickelt.

Weblinks

- [1] www.hobby-hour.com/electronics/smdcalc.php
- [2] www.resistorguide.com/resistor-smd-code/
- [3] www.radio-electronics.com/info/data/capacitor/smd_capacitor.php
- [4] https://de.wikipedia.org/wiki/Surface-mounted_device
- [5] www.sos.sk/pdf/SMD_Catalog.pdf
- [6] www.marsport.org.uk/smd/mainframe.htm
- [7] www.s-manuals.com/smd



Zusammengestellt von **Beatriz Sousa**

Noch logistischer

Wir hätten nie erwartet, dass der Elektor-Shop ein solcher Erfolg werden würde, aber er wurde es! Schon die Crazy-Xmas-Aktion war toll, aber die Cool Summer Deals stellten alles in den Schatten. Der Erfolg veranlasste uns, Lager und Logistik-Abteilung ein wenig aufzupeppen, so dass wir ab sofort schneller und problemloser liefern können. Wir entschuldigen uns, wenn Sie Opfer einer Verzögerung bei der Bearbeitung Ihrer Bestellung bei einer unserer sehr beliebten Werbeaktionen geworden sind.



Robotcamp

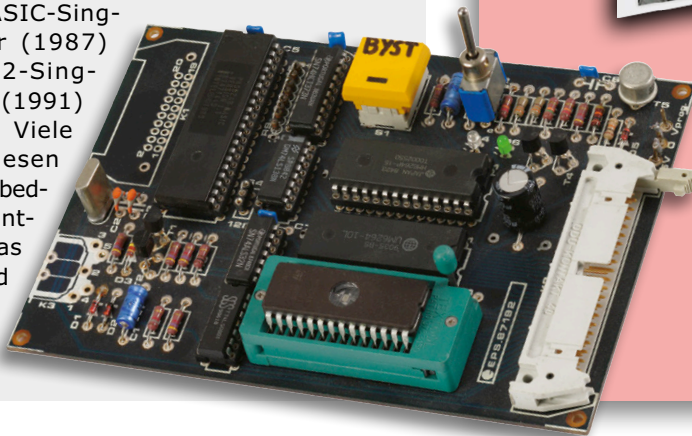
Kürzlich veranstaltete der Elektor-Experte Bart Huyskens einen Roboter-Workshop für Kinder in seinem Heimatland Belgien. Mit einigen Boards aus dem Elektor-Labor und Barts unnachahmlich begeisternder Methode, Elektronik zu vermitteln, hat er sicher die Kinder für ihr Leben als Elektroniker und als (Junior-)Elektor-Mitglieder gewonnen ;-).



READ ONLY MEMORY

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

Intels 8052AH-BASIC-Mikrocontroller war der Arduino der achtziger Jahre des vorigen Jahrhunderts. Elektor-Projekte wie der 8052AH-BASIC-Single-Board-Computer (1987) und der 8032/8052-Single-Board-Computer (1991) waren sehr beliebt. Viele Leser haben mit diesen Boards ihre ersten Embedded-Programmierkenntnisse gesammelt. Das 8052AH-BASIC-Board war im März 2007 Thema der Rubrik „Retronik“.



LEUTE • CJ Abate von **CircuitCellar** war (sehr!) behilflich, die Elektor-Kampagne Cool Summer großen Erfolg werden zu lassen • Muhammed Söküt, im deutschen Team verantwortlich für und -Produkte in themenverwandten, deutschsprachigen Communities an • Client Managerin Unternehmen dazu ein, „special reports“ in der nächsten Ausgabe von **Elektor Business** zu

Elektor Select Pages

schriften-Artikel bietet, gibt es noch viel mehr zu entdecken. Zum Beispiel die Elektor Select Pages.

Select Pages sind Webseiten, auf denen wir Informationen und Publikationen zu einem bestimmten Thema zusammenstellen. Diese Themen variieren, es sind natürlich auch beliebte Themen wie Arduino oder Raspberry Pi darunter. Select Pages sollen dem Leser auch ermöglichen, bestimmte Inhalte an einem bestimmten Ort zu finden, zum Beispiel all die digitalen Extra-Projekte oder die besten Elektor.TV-Sendungen. Andere Select Pages kombinieren bestimmte Themen, die zusammen einen Wissensbereich abdecken, wie zum Beispiel unsere (englischsprachige) Select Page „Power and Energy“, die nicht nur über erneuerbare Energien berichtet, sondern auch Batterietechnologien zum Thema hat. Es ist eine völlig neue und effiziente Möglichkeit, zu browsen, zu surfen und das Beste von Elektor zu entdecken! Sie können einen Blick auf eine unserer Select Pages werfen:

www.elektormagazine.com/select/energy.

Die Select Pages werden durch Tags (Schlagwörter) angesteuert. Klickt man auf einen Tag, der mit einer Select Page verknüpft ist, erscheint oben unter dem Textfeld der Suchmaschine ein Kästchen mit einem kleinen Ketten-Icon und einem Link zur Select Page. Außerdem gibt es in jedem Artikel oder jedem News-Beitrag, der Teil einer Select Page ist, ein kleines Banner, das direkt zur Select Page führt. Das ist einfach und bequem zu bedienen.

Wir sind offen für Ihre Anregungen! Senden Sie Ihre Vorschläge zu Themen, Fachgebieten und Inhalten, die wir für Sie auswählen sollen, an webmaster@eimworld.com.

Deals auch in den USA und Kanada zu einem unserer Mitglieder, bietet Elektor-Mitgliedschaften Margriet Debeij lädt deutsche und ausländische veröffentlichen.

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: **Jason Long**

Alter: **37**

Ausbildung: **Elektroingenieur (B.ScEE)**

Publikationen: **Sehr viele Online-Tutorials**

Training: **PIC, ARM, ANT+, BLE, LabVIEW, Low-power-Embedded-Entwicklungen**

Wer ist Jason Long?

Ich bin Kanadier und verliebt in Embedded-Systeme. Ich habe dieser Leidenschaft in den letzten 15 Jahren Ausdruck verliehen, indem ich ehrenamtlich andere Menschen unterrichtet habe. Seit 2010 arbeite ich bei Engenuics Technologies.

Welches wird die Schlüsseltechnologie in der Elektronik werden?

Netzwerke, die mit Nahfeldkommunikation (NFC) und Funkprotokollen wie ANT+ und BLE funktionieren. Dies wird das Internet der Dinge explodieren lassen. Embedded-Entwickler, die sich mit Hardware- und Firmware-Design auskennen, werden stark gefragt sein.

Was war DIE Entwicklung in der Elektronik?

In den letzten Jahren ist die ARM-Cortex-Familie enorm wichtig gewesen, vor allem mit Bausteinen wie dem Nordic nRF51. Er enthält alles, was man für Low-Power-Wireless benötigt (ich arbeite nicht für Nordic, aber ich liebe dieses Teil!).

Was unterscheidet Kanada von den USA, was die E-Innovation betrifft?

Vielleicht wird die Innovation in Kanada durch die Leidenschaft der Menschen angetrieben, während in den USA der Kapitalismus die Triebfeder ist? Aber ehrlich, Innovation entsteht überall aus dem Wunsch heraus, die Menschen durch großartige Produkte und Services zu verbinden.

Angenommen, wir schenken Ihnen 500 Dollar, um damit im Elektor-Shop zu shoppen. Was würden Sie kaufen und warum?

Ganz oben auf meiner Liste stehen Multikopter. Ihre Crazyfly 2.0 sieht ziemlich süß aus...

Welchen Entwickler, lebendig oder verstorben, bewundern Sie am meisten? Und warum?

Ich bin vielleicht ein bisschen komisch, weil ich mich nie mit bestimmten Personen identifiziere. Die Menschen, die ich bewundere, sind die Bastler und die Ingenieure mit Leidenschaft und Engagement, die Sachen herausfinden und die Dinge entwickeln wollen.

Auf welches Ihrer Projekte sind Sie am meisten stolz? Und warum?

Ich habe unsere Razor/Blade-Entwicklungsplattform über viele Jahre entwickelt und ich denke, sie ist wirklich solide. Sie füllt eine Lücke für Ingenieure, die aus der Hobby-Welt oder der Forschung zur „harten“ Produktgestaltung in der Industrie wechseln. ◀

(150344)



Willkommen in Ihrem **E-SHOP**

ELEKTOR EMPFIEHLT



CrazyFlie 2.0 (Kombi-Set)

Bei CrazyFlie 2.0 handelt es sich um einen kleinen Open-Source-Quadrocopter für den Einstieg, dessen Beherrschung sehr schnell erlernt werden kann. Trotzdem ist es am Anfang durch die Abmessungen (14 cm zwischen den Rotorenden) eine



Herausforderung, das Ding stabil zu fliegen. Aber wenn man einmal die Bedienung mit Smartphone oder Tablet intus hat, besteht Suchtgefahr. Mit einer Höchstgeschwindigkeit von immerhin 30 km/h muss man die Steuerung schon beherrschen, denn das ist wirklich schnell – vor allem in geschlossenen Räumen! CrazyFlie ist das Produkt dreier schwedischer Entwickler. Sie haben für dieses Projekt auch eine extra Support-Webseite aufgesetzt: www.bitcraze.io. Alle Software und auch ein komplettes Image einer virtuellen Maschine kann man dort herunterladen. Alle Parameter- und Log-Einstellungen kann man selbst anpassen. Wer dann wirklich tiefer einsteigen will, kann auch Hand an den Code legen.

Thijs Beckers (Elektor-Labor)

www.elektor.de/crazyflie-2-combo

Elektor-Bestseller

1. Red Pitaya V1.1
www.elektor.de/red-pitaya-instrument



2. Multikopter-Sonderheft
www.elektor.de/multikopter
3. Raspberry Pi 2 (Mod. B)
www.elektor.de/rpi-2
4. Arduino entdecken
www.elektor.de/arduino-entdecken
5. Arduino Sensor-Kit
www.elektor.de/arduino-sensor-kit
6. Raspberry Pi steuert Modelleisenbahn
www.elektor.de/rpi-eisenbahn
7. Elektor-DVD 2014
www.elektor.de/dvd2014
8. SmartScope
www.elektor.de/smartscope

Fahrzeugdiagnose mit OBD



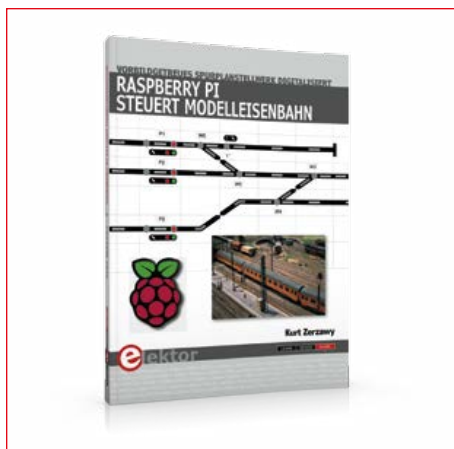
Streikt Ihr Auto schon wieder und Sie befürchten, dass die nächste teure Reparatur fällig wird? Vielleicht ist es aber auch nur ein kleiner Fehler, den Sie selber beheben können. Doch woher sollen Sie das bei einem modernen Fahrzeug mit all der Elektronik unter der Motorhaube wissen? Hier kann die fahrzeugeigene Diagnosefunktion helfen, den Fehler zu finden und Reparaturkosten zu senken, sodass Sie nicht bei jedem Aufblinken der Warnlampen gleich in die Werkstatt müssen.



Mitgliederpreis: 42,00 €

www.elektor.de/obd-buch

Raspberry Pi steuert Modelleisenbahn



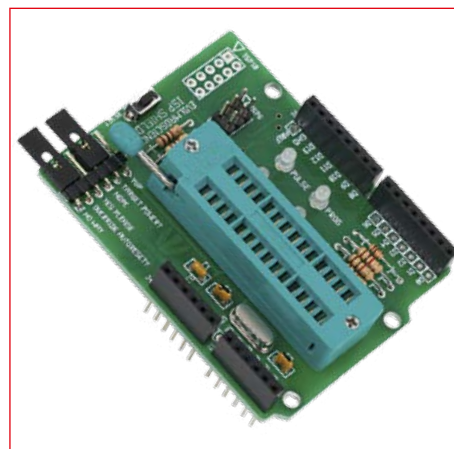
In diesem Buch findet der ambitionierte Modelleisenbahner die genaue Beschreibung, wie er sich für die eigene Anlage ein vorbildgetreues Druckstellenstellwerk mit Computer-Steuerung bauen kann. Es schließt die Lücke zwischen den früher beliebten Stellwerken auf Relaisbasis und den Möglichkeiten, die kleine Rechner wie der RPi bieten. Sie lernen alle für die Planung und den Aufbau notwendigen Schritte.



Mitgliederpreis: 34,80 €

www.elektor.de/rpi-eisenbahn

ISP-Shield für Arduino



Dieser Bausatz enthält eine Platine und alle benötigten Bauteile, um ein ISP-Shield (In-System Programmer) für Arduino-Boards zu bauen. Mit diesem Shield kann man dann auch andere Bootloader in nackte AVR-Controller brennen und das direkt aus der Arduino-Programmierungsumgebung. Der Shield ist mit einem 28-poligen ZIF-IC-Sockel für das bequeme Platzieren zu programmierender Mikrocontroller (ATmega168/328) versehen.

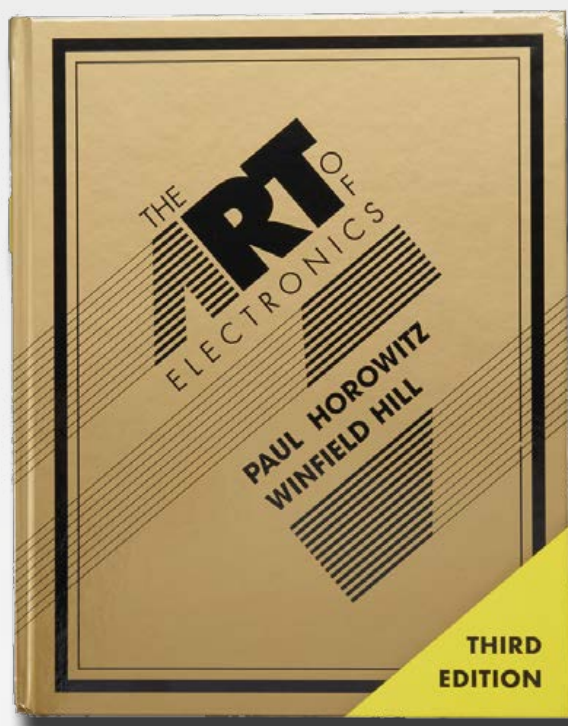


Mitgliederpreis: 17,96 €

www.elektor.de/isp-shield-arduino



Neueste Auflage!



The Art of Electronics

Unverzichtbares Nachschlagewerk für Studenten, Forscher und Entwickler

Endlich ist die sorgfältig überarbeitete und auf den neuesten Stand gebrachte 3. Auflage des erfolgreichen Bands *The Art of Electronics* verfügbar. Dieses Buch gilt gemeinhin als die Grundlage schlechthin für die Entwicklung elektronischer Schaltungen.

Zusätzlich zur erneuerten und breiteren Abhandlung vieler Punkte enthält die dritte Ausgabe 90 Oszilloskop-Screenshots, welche das Verhalten und die Funktion von Schaltungen beschreiben. Darüber hinaus illustrieren dutzende Diagramme gemessene Daten, die ansonsten tief in Datenblättern versteckt sind oder überhaupt ganz fehlen. Diese Infos sind beim Entwickeln von Schaltungen genauso hilfreich wie die etwa 80 Tabellen, in denen rund 1.650 aktive Bauelemente aufgeführt werden.

Elektor Labs Löt-Starterkit

Arduino-Bundle

(Buch + Hardware-Starterkit)



Mitgliederpreis: 80,10 € (frei Haus)

www.elektor.de/art-of-electronics

Offizielles Gehäuse für Raspberry Pi

Praxiskurs AVR-XMEGA-Mikrocontroller

RFID-Starterkit für Arduino Uno



Der geistige Vater des Raspberry Pi Eben Upton und seine Kollegen von der Raspberry Pi Foundation haben sehr viel Zeit in die Entwicklung eines schicken und funktionalen Gehäuses für das Modell 2 und B+ gesteckt. Bei dem aus 5 Kunststoffteilen bestehenden Gehäuse sind alle Anschlüsse frei verfügbar. Außerdem lassen sich die Teile der Hülle entfernen, damit der Raspberry Pi mit anderen Geräten verbunden werden kann.



Mitgliederpreis: 8,96 €

www.elektor.de/official-case-rpi



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 €

www.elektor.de/kurs-avr-xmega



Dieses Kit bietet eine sehr reichliche Sammlung von Boards etc. für so gut wie alle Arduino-Projekte. Nicht nur eine RFID-Erweiterung ist vorhanden, sondern umfangreiche Elektronik zum Messen, Datenerfassen und zur Steuerung.



Mitgliederpreis: 44,96 €

www.elektor.de/rfid-starterkit-arduino-uno



Von Luc Henderieckx

Hardware und technische Daten von Red Pitaya sind richtig beeindruckend. Hinzu kommt noch umfangreiche Software, mit der das Messgerät zum Leben erwacht. Mit dieser Software nimmt man nicht nur Messungen vor, sondern kann damit auch die entsprechenden Einstellungen vornehmen.

Da stellte sich die Frage, ob Elektor nicht das Programmieren passender Applikationen starten und in einem

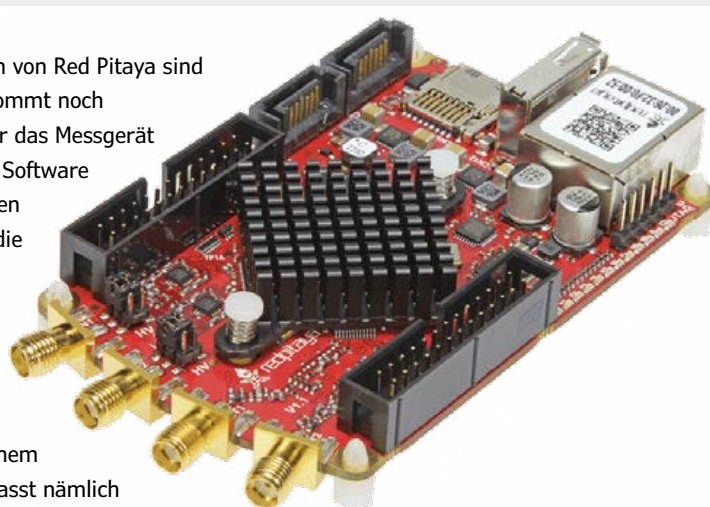
Kursus erklären sollte, wie Red Pitaya zu programmieren ist? Das passt nämlich genau zur Ausrichtung von Elektor! Und tatsächlich sind eine Menge passender Artikel zu Mikrocontrollern in Elektor zu finden und diese Hardware kommt dem Gedanken an Selbstbau-Elektronik sehr nahe.

Mehr Infos finden Sie unter

www.elektor.de/red-pitaya

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.



SMARTSCOPE ist das weltweit erste
Open-Source-USB-Speicheroszilloskop für Smartphone, Tablet und PC mit



Mitgliederpreis:

229,00 € **206,10 €**

Nur für kurze Zeit: Jetzt bestellen und 10-EUR-Wertgutschein GRATIS sichern!

www.elektor.de/smartscope



Elektor Labs Löt-Starterkit



Dieses Löt-Kit bildet die Basis-Ausrüstung für jeden, der sich mit Elektronik näher beschäftigen möchte. Die digitale Lötstation erlaubt eine Temperatureinstellung im Bereich von 160 bis 480 °C. Die Lötspitze ist netzgetrennt, denn der Kolben wird mit 24 V Gleichspannung betrieben. Auf der Station wird sowohl die eingestellte als auch die aktuelle Temperatur der Lötspitze angezeigt. Das Digitalvoltmeter (DVM) DT-830B misst Gleich- und Wechselspannung, Strom und Widerstand.

The Art of Electronics

Unverzichtbares Nachschlagewerk für Studenten, Forscher und Entwickler

Elektor Labs Löt-Starterkit

Arduino-Bundle

(Buch + Hardware-Starterkit)



Mitgliederpreis: 134,96 €

www.elektor.de/elss

Multicopter selber bauen

Mystery Product

LEGO-LED-Board



Das Buch wendet sich an eine technisch versierte Zielgruppe, die eigene Flugmodelle bauen möchte. Fachbegriffe werden verständlich erklärt und der Leser muss weder Luftfahrer noch Diplom-Elektroniker sein. Neben der Herstellung eigener Flugmodelle wird die gesamte Technik für Multicopter und damit verbundene Anwendungen wie FPV (First Person View) und Luftaufnahmen beschrieben



Mitgliederpreis: 34,90 €

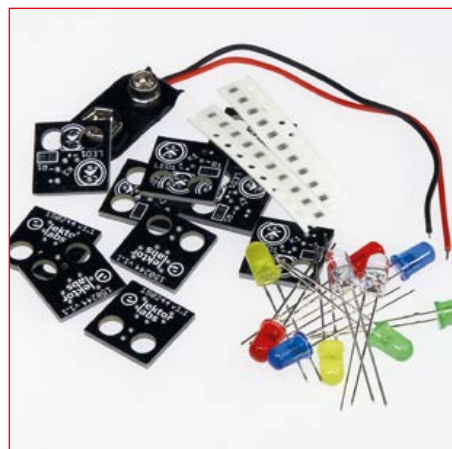
www.elektor.de/multicopter-selber-bauen



We dare you ...



www.elektor.de/mystery



Beim LEGO-LED-Board handelt es sich um eine kleine Platine, die physisch kompatibel mit den kleinen Kunststoffklötzen aus Dänemark ist. Sie beherbergen eine LED, einen Vorwiderstand und optional Anschlusspins. Damit kann man tolle Lichteffekte bei seinen LEGO-Bauten erreichen. Das Kit ist voll anpassbar. Es enthält zehn kleine Platinen, zehn verschieden farbige LEDs und je zehn Vorwiderstände zu 390 Ω und 820 Ω. Ein Batterie-Clip für eine 9-V-Batterie befindet sich im Lieferumfang.



Mitgliederpreis: 8,96 €

www.elektor.de/legoed

Neue ELEKTOR-SEMINARE

- ✓ **01.09.2015 – 02.09.2015**
2-tägiges Seminar
EMV-Stromversorgung
(Gmunden, Österreich)
- ✓ **24.09.2015 – 25.09.2015**
2-tägiges Seminar
FPGA
(Regensburg)
- ✓ **20.10.2015**
Tagesseminar
CAN-Grundlagen
(Troisdorf)

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

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

Trans Röhrenverstärker



Ist es möglich, Fehler in Röhren und Ausgangsstromen spürbar zu kompensieren? Im vorigen Jahrhundert wurden sehr viele Maßnahmen zum Erreichen dieses Ziels entwickelt. Eine dieser Methoden ist in Vergessenheit geraten: Transkonduktanz, was soviel bedeutet wie „Umsetzung von Strom in Spannung“ und umgekehrt. Autor Menno van der Veen hat diese Methode wieder zum Leben erweckt und gab ihr den Arbeitstitel „Trans“.



Mitgliederpreis: 29,80 €

www.elektor.de/trans-roehren

Internet of Things



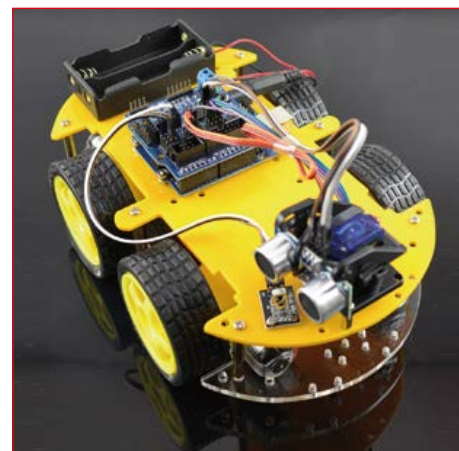
Das Internet of Things (Internet der Dinge) ist eine unumkehrbare Entwicklung. Wir möchten gerne alles im Haus mit unserem Smartphone oder Tablet erledigen – von Facebook bis Fernsehen, Lampen steuern oder die Heizungstemperatur einstellen. In diesem Buch stellen wir 35 interessante und nützliche Projekte vor, die demonstrieren, wie Sie selbst ein Internet-of-Things-System anlegen können.



Mitgliederpreis: 39,80 €

www.elektor.de/iot-buch

Robot Car Kit für Arduino mit Bluetooth



Dieses intelligente Auto mit Vierradantrieb kann über ein Android-Mobilgerät via Bluetooth oder Infrarotlicht (mitgeliefert) gesteuert werden. Die Intelligenz stammt von einem Arduino Uno, weshalb das Auto Hindernisse umfahren oder einer Route folgen kann. Das Ganze ist eine Entwicklungsplattform für autonomes Fahren, das sich nicht nur für Studienzwecke, sondern auch rein zum Spaß eignet.



Mitgliederpreis: 89,96 €

www.elektor.de/robot-car-kit-arduino



Mit diesem Bundle (bestehend aus dem neuen Buch „Arduino entdecken“, Hardware-Bausatz und Arduino Prototyping Shield) können Sie alle im Buch beschriebenen 45 Arduino-Projekte, beispielsweise ein stiller Alarm, ein Menschensensor, ein Lichtmessgerät, die Motorsteuerung und die Bedienung über das Internet oder eine Fernsteuerung (mit Funkverbindung) sofort aufbauen.

Im Gegensatz zu vielen Gratis-Projekten aus dem Internet sind alle Projekte in diesem Buch in der Praxis getestet und funktionieren garantiert!

The Art of Electronics

Unverzichtbares Nachschlagewerk für Studenten, Forscher und Entwickler

Elektor Labs Löt-Starterkit

Arduino-Bundle

(Buch + Hardware-Starterkit)



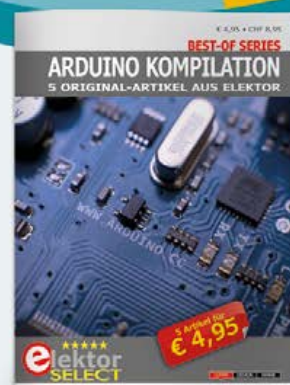
Mitgliederpreis: 56,71 €

www.elektor.de/arduino-entdecken-kit

MADE IN ITALY
DAS GESAMTE OFFIZIELLE
ARDUINO-SORTIMENT



GRATIS E-BOOK
BEI JEDER
ARDUINO-BESTELLUNG



JETZT bei ELEKTOR!

Hexadoku

Sudoku für Elektroniker

Der Urlaub gehört für die meisten unserer Leser der Vergangenheit an – nun ist es wieder Zeit für den Ernst des Lebens. Doch muss das ja nicht heißen, ständig zu arbeiten: Für etwas Entspannung haben wir wieder eines unserer schönen Hexadokus vorbereitet. Wie immer kann man auch etwas gewinnen. Hier warten fünf Elektor-Gutscheine ...

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 Gutscheinen im Wert von 50 Euro gewinnen!



Einsenden

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

Elektor Redaktion
Süsterfeldstr. 25
52072 Aachen

Fax: 0241 / 88 909-77

E-Mail: hexadoku@elektor.de

Als Betreff bitte nur die Ziffern der Lösung angeben!

Einsendeschluss ist der
30. September 2015.

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

Die richtige Lösung ist: **458EF**

Einen Elektor-Wertgutschein über je 50 € haben gewonnen:

Gerard Lodder, Karl-Josef Wernet, Olivier Chaufouraux, Pascal Jordil und Dirk Petig.

Herzlichen Glückwunsch!

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

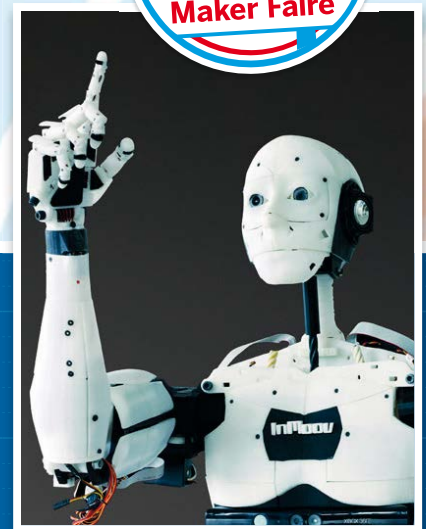
F	5	7	3	6	A	9	C	2	8	B	D	4	1	0	E
B	0	1	9	E	F	D	2	3	7	4	A	8	5	6	C
2	4	8	E	0	1	3	7	9	5	6	C	B	D	A	F
6	A	C	D	B	4	5	8	E	F	0	1	2	7	9	3
0	F	4	7	C	2	1	6	A	D	8	9	E	3	5	B
5	E	D	1	9	7	8	0	6	B	F	3	A	4	C	2
3	2	9	8	4	B	A	5	C	E	1	7	F	6	D	0
A	6	B	C	D	3	E	F	5	0	2	4	7	8	1	9
C	7	E	4	8	D	F	1	B	9	3	5	0	A	2	6
1	D	F	5	2	9	0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

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

Maker Faire® Berlin

FAMILIEN-FESTIVAL FÜR INSPIRATION,
KREATIVITÄT & INNOVATION

DER GROSSE MAKER-TREFFPUNKT.



3. & 4. Oktober 2015
Postbahnhof Berlin

präsentiert von:

Make:
make-magazin.de

Goldschmied:



Silberschmied:



Kupferschmied:



Kooperationspartner:



Medienpartner:



Stand: 24.07.2015

Tickets und weitere Informationen unter

WWW.MAKERFAIRE.BERLIN

JETZT ABONNIEREN!

Newsletter

Erhalten Sie wöchentlich
frische Informationen über

- Neuheiten
- beste Angebote
- Preisreduzierungen



TEMPERATUR
NÄHERUNGSSCHALTER
WEGAUFNEHMER
FEUCHTE
DURCHFLUSS
DRUCK
HALL-EFFEKT
ULTRASCHALL
KRAFT

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

Hall-Effekt-Sensoren

Radiometric-Sensor, Honeywell linear

- Betriebsspannung: 4,5 ... 10,5 VDC
- Stromaufnahme: 8,7 mA (@ 5 VDC)
- Betriebstemperatur: -40 ... +150 °C
- Linearität: 1% typ.
- Antwortzeit: 3 µs

Gehäuse: TO-92,
Ausgang: 0,2 V / 1,5 mA



Bestellnummer	Magnetic-Range	Empfindlichkeit
SS 496 A1	±84 mT	2,5 mV/G
SS 495 A	±67 mT	3,125 mV/G
SS 495 A1	±84 mT	3,125 mV/G

Radiometric-Sensor, Honeywell linear, SMD

- Betriebsspannung: 2,7 ... 6,5 VDC
- Stromaufnahme: 10 mA
- Betriebstemperatur: -40 ... +100 °C
- Linearität: 1% typ.
- Antwortzeit: 3 µs

Gehäuse: SOT-89
Ausgang: 1,0 V / 1,5 mA



Bestellnummer	Magnetic-Range	Empfindlichkeit
SS 59 ET	±65 mT	1,0 mV/G

Hall-Effektsensor, Honeywell digital, SMD

Temperaturkompensierte
digitale Halleffektsensoren

unipolar, Gehäuse: SOT-89
Ausgang: 0,4 V / 20 mA



Bestellnummer	Magnetic-Range
SS 543 AT	1,40 7,5 ... 18,0 mT
SS 549 AT	2,65 23,5 ... 39,0 mT

Hallsensoren SIEMENS



Gehäuse: P-SSO-3-2

Bestellnummer	Magnetic-Range	
TLE 4905L	0,60 ±17 mT	unipolar
TLE 4935L	0,63 ±20 mT	bipolar/latch

Induktiver Näherungsschalter

CONTRINEX

Exzellente induktive Sensoren für das berührungslose Erfassen von metallischen Teilen im Automatisierungsalldag. Ideal für den Einsatz in anspruchsvollen Umgebungsbedingungen.



Auszug aus unserem Sortiment.
Das komplette Programm finden Sie unter: <http://rch.it/kj>



Schließer PNP

Bestellnummer	Ø x L	Schaltabstand	Einbau
DWAD 623 03	66,80 Ø 3,0 x 22 mm	1,0 mm	bündig
DWAD 623 04	51,40 Ø 4,0 x 25 mm	1,5 mm	bündig
DWAD 623 M5	51,40 M5 x 25 mm	1,5 mm	bündig
DWAD 503 M12	47,90 M12 x 50 mm	6,0 mm	quasi-bündig
DWAS 623 M8 001	32,90 M8 x 45 mm	2,0 mm	quasi-bündig
DWAS 613 M8 001	30,80 M8 x 45 mm	2,5 mm	nicht bündig
DWAS 513 M12	47,80 M12 x 60 mm	10,0 mm	nicht bündig

Lichtschranke CONTRINEX Ø x L 18 x 50 x 50 mm



Einweg-Lichtschranke, 2 m Kabel

Bestellnummer	Schaltabstand	Schaltungsart
LLK 5050 000	23,30 15000 mm	Sender
LLK 5050 003	35,99 15000 mm	antivalent, PNP

Reflexions-Lichtschranke, 2 m Kabel

LLK 5050 103	42,40 800 mm	antivalent, PNP
--------------	--------------	-----------------

Distanzsensoren

SHARP

Sehr hohe Zuverlässigkeit
sowie höhere Präzision
gegenüber herkömmlichen
Sensoren.

Typen: GP2Y0A

Bestellnummer	Reichweite (cm)	LxBxT (mm)
GP2-0215	6,70 20 ... 150	29,5 x 13,0 x 21,6
GP2-0430	5,10 4 ... 30	37,0 x 18,9 x 13,5
GP2-1080	5,65 10 ... 80	29,5 x 13,0 x 21,6

Bestellnummer	
DMC01-SC150	2,15 3-polig Datenkabel



Feuchtesensor 0 ... 100% rF, TO 39

Digitaler Feuchtesensor
mit IC-Interface im druckfesten
TO39-Gehäuse (bis 16 bar),
geeignet für Taupunktmessungen.

B+B
SENSORS



Bestellnummer	
HYT 939	29,90 TO 39

Drucksensoren, UsV 4,75 - 5,25

freescale
semiconductor

Bestellnummer	PR kPa	Sens mV/kPa	Lin %
MPX 5010DP	11,60 0 - 10	450	±5
MPX 5050DP	14,90 0 - 50	90	±2,5
MPX 5100DP	14,90 0 - 100	45	±2,5
MPX 5500DP	9,35 0 - 500	9	±2,5



Ultraschallsensoren

Ultraschall-Keramik-Sender
und Empfänger für 40 kHz



Ø 9,9 mm, H 7,1 mm

Bestellnummer	
MUS-40E	3,05 Empfänger
MUS-40S	3,05 Sender