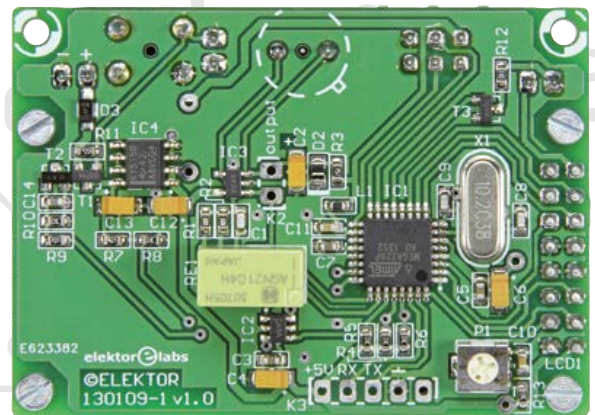
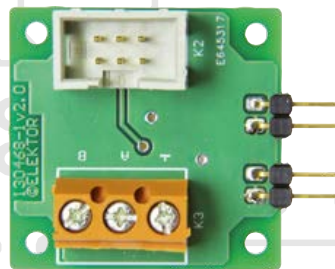
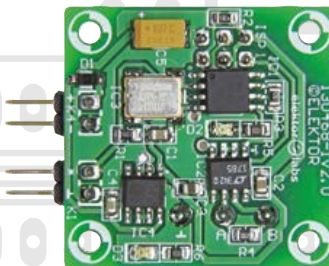
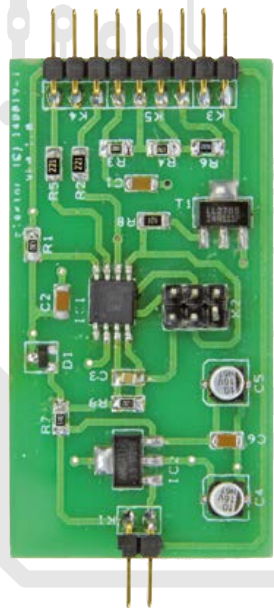
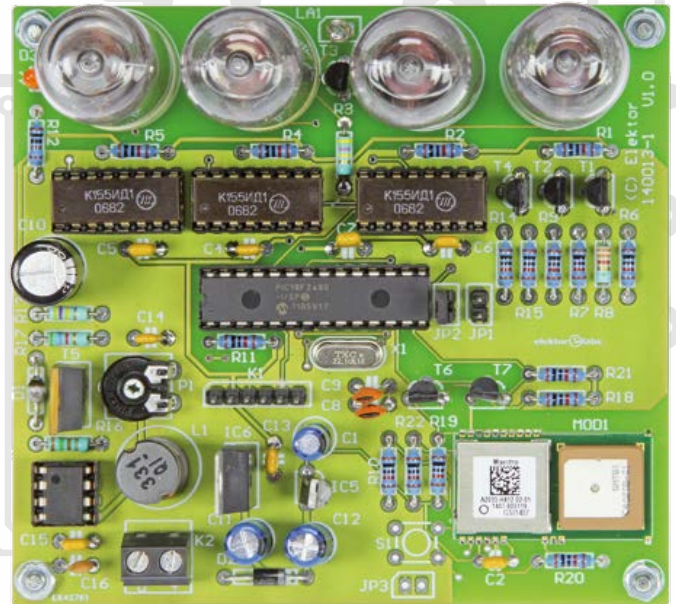
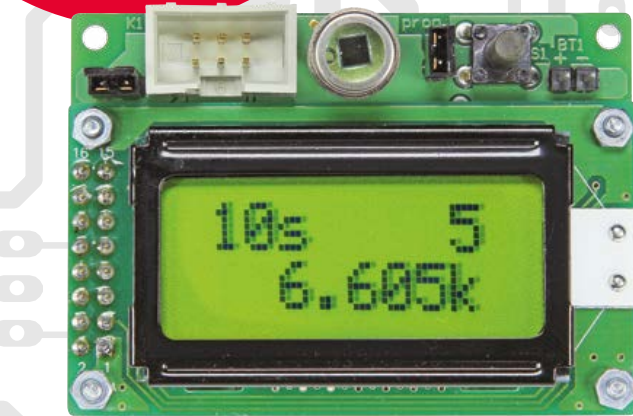


Neu: 8 Seiten mehr!

magazine

Oktober 2014 | Nr. 526

e!ektor



- Luxmeter 1 lx bis 100 klx | **Temp-Sensor-Board mit RS485**
- Dimmer für LED-Taschenlampen** | Nixie-Uhr mit GPS | Projekt: Video über Glasfaser
- Platinen-Trafos | Gyrator-Ferritantenne | **Mikrocontroller für Einsteiger: SPI**
- Bauteiltipp: RTC-Module** ● Einer für alles: Programmier Atmel-ICE
- Retronik: Programmierbarer IC-Tester



Elektor 3D-Drucker

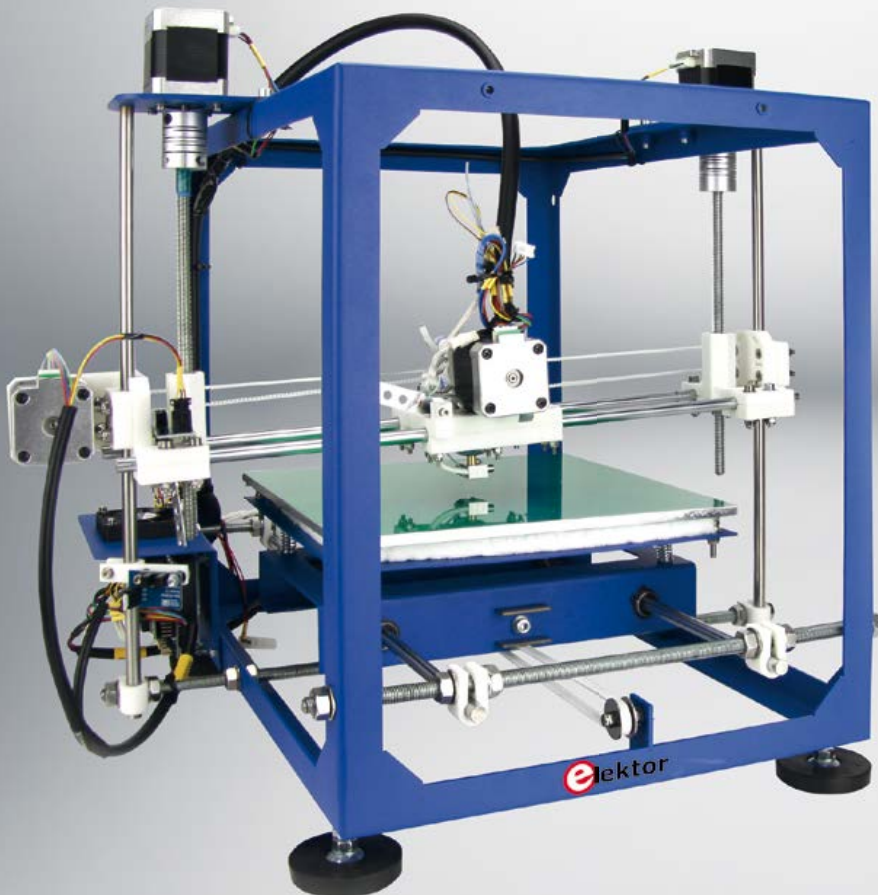
Drucken und Modellieren in der dritten Dimension

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

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

Technische Daten:

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



Lieferumfang:

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

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

Die Lieferung innerhalb Deutschlands erfolgt frei Haus.

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

eC-reflow-mate

Professioneller SMD-Reflow-Ofen für perfekt gelötete Platinen

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

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



Technische Daten:

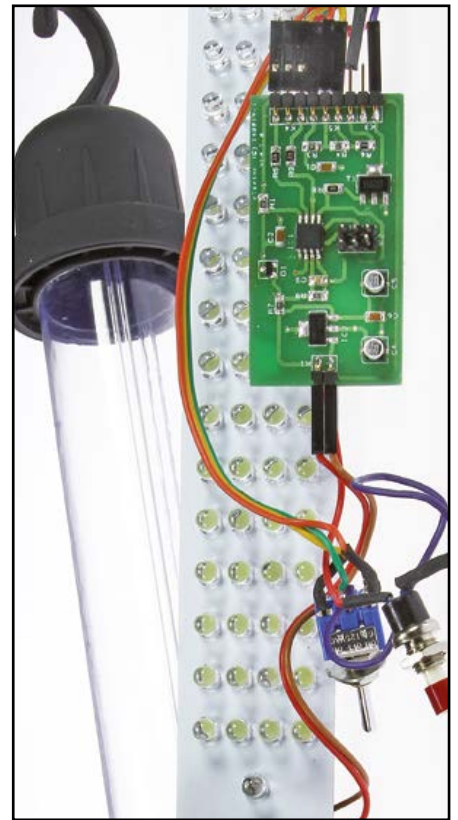
- Betriebsspannung: 230 V/50 Hz
- Leistung: 3000 Watt
- Gewicht: 29 kg
- Abmessungen (B x H x T): 62,0 x 24,5 x 52,0 cm
- Heizung: Infrarot-Strahler kombiniert mit Umluft
- Bedienung: Funktionstaster und LCD am Ofen oder via USB mit PC-Programm
- Temperaturbereich: 25...300 °C (300 °C Spitzenwert und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

Besonderheiten:

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

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

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



● Labs

44 Ausprobiert: Atmel ICE

Auf der letzten Embedded World stellte Atmel einen neuen Low-Cost-Programmer/Debugger vor, der sich gleichermaßen für die AVR-Controller inklusive Xmega wie auch die ARM-Cortex-Controller des Herstellers eignet. Wir haben das kleine Kästchen ausprobiert.

52 Seltsame Bauteile (8)

Das Magnetron

54 DesignSpark Tipps & Tricks

In dieser Folge gibt es keine Handarbeit, vielmehr kann man dem Autorouter beim Arbeiten zusehen...

57 Bauteiltipp: RTC-Module

Die aktuelle RTC-Familie des Herstellers Micro Crystal bedient mit energiesparenden, sehr genauen oder besonders kleinen Modulen die unterschiedlichsten Ansprüche.

Der typische 32.768-Hz-Uhrenquarz ist immer schon integriert.

60 (Los)gelöst

● Projects

10 Luxmeter 1 lx bis 100 klx

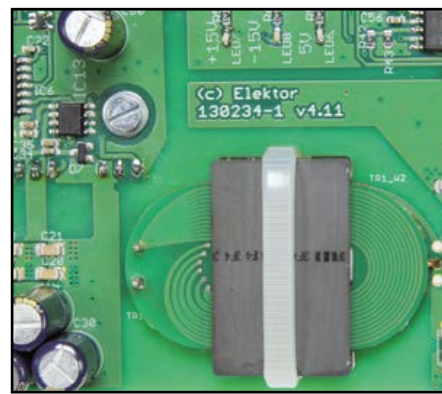
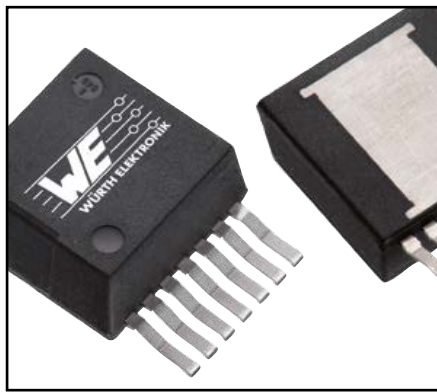
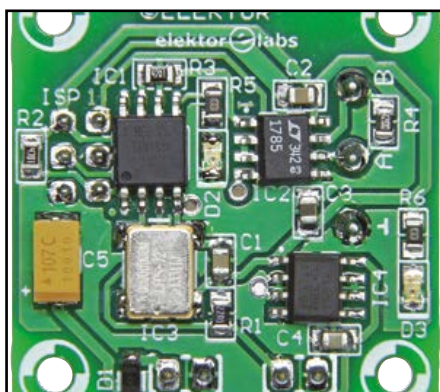
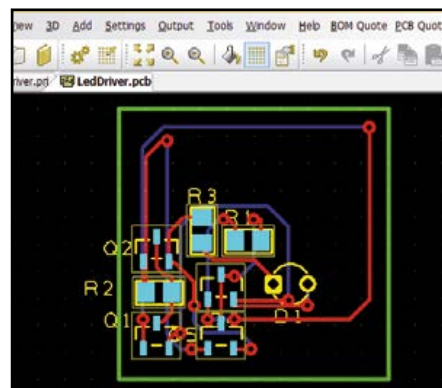
Mit einem Luxmeter kann man die Herstellerangaben von Halogen-, Energiespar- und LED-Lampen überprüfen; aber auch messen, wie hell es am Arbeitsplatz ist. Eine Fotodiode, ein 18-bit-ADC und ein ATmega328 sind die wesentlichen Zutaten unseres Luxmeters. Es verfügt über eine automatische Umschaltung der Messbereiche und lässt sich über eine serielle Schnittstelle kalibrieren.

16 Temp-Sensor-Board

Für viele Automatisierungsaufgaben benötigt man Temperatursensoren. Per RS485 lassen sich die Werte zuverlässig auch über längere Strecken übertragen. Unser kompaktes Temp-Sensor-Board ist mit einem ATtiny-Controller und einem RS485-Treiber ausgestattet; es können mehrere Temperatursensoren parallel angeschlossen werden. Dazu bieten wir eine Standardfirmware an, die zur Übermittlung der Messwerte das ElektorBus-Protokoll nutzt. PC-Software zur Anzeige der Daten rundet das Projekt ab.

24 Dimmer für LED-Taschenlampen

Die einstmals unscheinbaren LEDs haben sich zu lichtstarken und effizienten Leuchtmitteln gemauert. Doch die strahlende Helligkeit kann hinderlich sein, wenn eine leistungsstarke LED-Taschenlampe ihre Blendwirkung entfaltet. Der Dimmer mindert diese Erscheinung



durch Pulsbreitenmodulation des LED-Stroms. Außerdem kann die Taschenlampe auf Knopfdruck blinken.

28 Nixie-Uhr mit GPS

Das sanfte Glimmen nostalgischer Ziffernanzeigeröhren gibt dieser Uhr einen extravaganten, nicht alltäglichen Touch. Doch die Uhr ist auch sehr modern, denn GPS-Signale sorgen für eine hohe Genauigkeit.

34 Mikrocontroller für Einsteiger (6)

Beim Seriellen Peripherie-Bus (SPI) laufen die eigentlichen Daten Bit für Bit über eine Leitung; zum Beispiel von einem Mikrocontroller zu einem Display, einem EEPROM oder einer SD-Karte. Da man meist auch Daten lesen will, braucht man eine zweite Leitung, auf der Bits zurückflitzen können. Aber dann kommt noch etwas hinzu, nämlich eine Taktleitung. Unnötig sind dann genaue zeitliche Vereinbarungen zwischen den Gesprächspartnern.

Mit den drei Leitungen wird die Übertragung ziemlich robust.

62 Video über Glasfaser

Mit einer Handvoll preisgünstiger Standardbauteile werden wir versuchen, ein Videosignal praktisch ohne Verluste und Verzerrungen über eine lange Glasfaserstrecke zu übertragen.

70 Platinenintegration magnetischer Bauelemente

Spulen und Trafos sind nicht aus der Elektronik wegzudenken. Auch im Projekt „Professionelles Labornetzteil“ aus Elektor September 2014 war ein Trafo erforderlich. Seine Wicklungen bestanden aus Leiterbahnen einer Platine. Dieser Beitrag beschäftigt sich mit der speziellen Bauweise solcher magnetischer Komponenten und ihrer Simulation.

76 Gyrator-Ferritantenne

Um Kondensatoren und Spulen zu simulieren, kann man Gyratoren

verwenden. Dabei werden Gyratoren als Impedanzkonverter benutzt, was in diesem Artikel beschrieben wird. Danach zeigen wir, wie man mit einem Gyrator eine abstimmbare Ferritantenne realisieren kann.

● Magazine

6 Impressum

8 Updates und Mailbox

82 Retronik

BK Precision BK560: Programmierbarer IC-Tester

86 Hexadoku

Sudoku für Elektroniker

90 Vorschau

Nächsten Monat in Elektor

Impressum

45. Jahrgang, Nr. 526 Oktober 2014
Erscheinungsweise: 10 x jährlich
(inkl. Doppelhefte Januar/Februar und Juli/August)

Verlag

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

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

Hauptsitz des Verlags

Elektor International Media
Allee 1, NL-6141 AV Limbricht

Anzeigen:

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

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

Es gilt die Anzeigenpreisliste Nr. 44 ab 01.01.2014

Distribution:

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

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

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

Board-Sammlung

Wie viele andere Kreative brauchen auch wir Elektronik-Redakteure immer wieder etwas Inspiration, um unser schönes Handwerk ausüben zu können. Der Arbeitsplatz meines Kollegen Jan Buiting ist von Messgeräten mit wunderlichen Skalen und sah-nig laufenden Drehreglern umgeben, und hin und wieder stellt er sich ein Netzgerät auf seinen Tisch, mit dem man problemlos mittelschwere Abbrucharbeiten durchführen könnte. Bei mir sind es natürlich Mikrocontrollerboards. Obwohl recht ordnungsliebend veranlagt, lasse ich einen Teil der stetig wachsenden Sammlung absichtlich auf meinem Tisch liegen. Denn wenn man so ein kleines Teil von Zeit zu Zeit in die Hand nimmt – etwa beim Telefonieren – dann kommen oft die besten Ideen für neue Projekte. Ein paar neue „Spielzeuge“ finden sich seit ein paar Wochen auf meinem Tisch: Ein Programmer/Debugger Atmel ICE, das Temp-Sensor-Board aus unserem Labor sowie ein interessantes Entwicklungs-Board mit ARM-Cortex-M0-Prozessor. Der Atmel ICE (Review auf S. 44) ist bei mir auch im Praxis-einsatz, hierfür habe ich mir Atmel Studio 6.2 heruntergeladen (ich warte übrigens gespannt auf ein Päckchen von Benedikt Sauter, das den brandneuen Open-Source-Programmer USBprog 5.0 enthalten soll, siehe kommende Ausgabe). Nach dem Schreiben dieser Zeilen werde ich mit dem ICE die allerneueste Firmware in unser Temp-Sensor-Board flashen, die von unserem freien Mitarbeiter Mauk von der Laan stammt. Mauk hat sich ein eigenes kleines Betriebssystem ausgedacht, das sogar in einen ATtiny passt (siehe S. 16). Und das ARM-Cortex-M0-Board? Wird die Basis werden unserer geplanten Serie: „Von 8- auf 32-bit: ARM-Controller für Einsteiger“. Mehr wird noch nicht verraten, aber im Dezemberheft geht es los!



Jens Nickel

Chefredakteur Elektor

Unser Team

Chefredakteur:

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

Ständige Mitarbeiter:

Dr. Thomas Scherer, Rolf Gerstendorf

Leserservice:

Ralf Schmiedel

Korrekturen:

Malte Fischer

Internationale Redaktion:

Harry Baggen, Jan Buiting, Denis Meyer

Elektor-Labor:

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

Grafik & Layout:

Giel Dols



Germany

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



United Kingdom

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



Netherlands

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



France

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



USA

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



Spain

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



Italy

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



Sweden

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



Brazil

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



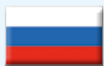
Portugal

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



India

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



Russia

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



Turkey

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



South Africa

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



China

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

Unser Netzwerk



VOICE COIL



audioXpress



Die Elektor-Community



Unsere Partner und Sponsoren



Batronix

www.batronix.com/go/39 69



Beta Layout

www.beta-prototypes.com 19



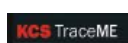
CadSoft

www.cadsoft.de 23



Eurocircuits/PCBservice

www.elektorpcbservice.com 61



KCS

www.trace.me 92



LeitOn

www.leiton.de 9



LPKF

www.lpkf.de/prototyping 67



Messe München

www.electronica.de 75



Pico

www.picotech.com/PS363 9



Reichelt

www.reichelt.de 85, 87



Schaeffer AG

www.schaeffer-ag.de 9

Sie möchten Partner werden?

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

Updates & Ergänzungen

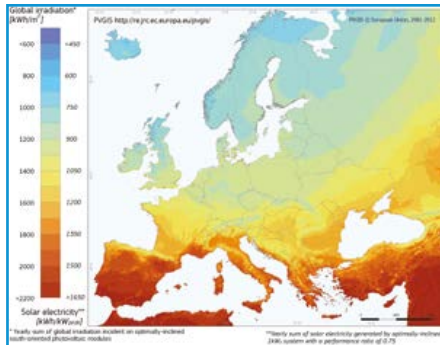
Photovoltaiksysteme berechnen und realisieren

Elektor 01/2014, S. 30 (130404)

Es haben sich ein paar Fehler bei der Übersetzung/Übertragung auf die Verhältnisse in Deutschland eingeschlichen:

In der Tabelle 1 wird der Energiebedarf pro Tag berechnet. Hier wurde vereinfacht von Energiebedarf gesprochen, physikalisch korrekt müsste hier

(bis auf die letzte Zeile) der Begriff Energie durch Ladung ersetzt werden, oder - da es sich um einen Akku handelt - durch den Begriff Kapazität. Die tägliche Sonneneinstrahlung von Berlin beträgt 1,08 kWh/m²/Tag und nicht 0,4 kWh/m²/Tag.



Korrektur der Formeln auf S. 33 unten:

$$\frac{\text{Leistung des Solarmoduls in Watt} = \text{Energiebedarf während 24h}}{\text{Tägliche Sonneneinstrahlung} \times \text{Verlustfaktor}}$$

Minimal erforderliche Nennleistung des Solarmoduls bei einem Energiebedarf von 27,36 Wh/Tag:

$$P = \frac{27,36 \text{ Wh}}{1,08 \times 0,7} = 36,19 \text{ W}$$

Anmerkung: Wenn man die Werte mit Einheiten korrekt in die Formel einsetzt, kommt eine kleine Fläche heraus, die man mit 1000 W/m² multiplizieren muss, um die Nennleistung zu erhalten. Die Nennleistung eines Solarmoduls bezieht sich nämlich auf die maximale Leistung der Sonneneinstrahlung von rund 1000 W/m².

Mein erstes Shield

Elektor 07/2014, S. 32 (140009)

Das fertig aufgebaute Modul 140009-91 wird mit einem LCD-Display vom Typ EA DIPS082-HN (ohne Hintergrundbeleuchtung) geliefert, und nicht mit einem LCD-Display vom Typ EA DIPS082-HNLED. Auch im Artikel 140064-I (Mikrocontrollerkurs) wird erwähnt, dass das Display auf dem Shield ein LED-Backlight hat. Das ist beim Komplettmodul nicht der Fall.

Platino-Experimentier-Netzteil

Elektor 04/2014, S. 20 (130406)

In der Stückliste der Platine 130406-1 ist leider ein Fehler. Damit das Projekt mit der aktuellen Firmware funktioniert, muss ein MCP-4261-103E/P und nicht ein MCP42010-E/P (wie in der Stückliste genannt) verwendet werden. Die zwei Chips sind pin-kompatibel, aber nicht software-kompatibel. An einem Software-Update (um auch den MCP42010-E/P verwenden zu können) wird gearbeitet.

Mailbox

Zinkpest

Radio Atwater Kent Modell 30 (1926),
Elektor 05/2014, S. 80 (130426)

Sehr geehrte Damen und Herren,
zum lesenswerten Artikel von Gerhard Fonte eine Anmerkung. Die Erklärung für die zerstörte Riemenscheibe „ein Extremfall von Materialermüdung, da das metallische Material gesintert statt gegossen oder gedreht war“ ist falsch.

Es handelt sich hier um die sogenannte „Zinkpest“, welche z.B. bei Sammlern alter Metallmodelle gut bekannt ist. Betroffen sind insbesondere Zink-Aluminium-Legierungen, wie sie für den Druckguss üblich sind, welche geringe Verunreinigungen der Elemente Blei, Zinn, Cadmium oder Thallium aufweisen. Diese Verunreinigungen reichern sich beim Erstarren des Materials im Herstellungsprozess an den Grenzen der einzelnen Metallkristalle an. Eine feuchte Umgebung



führt dann oft zu einem selektiven Korrosionsangriff entlang der Korngrenzen, der die Zerstörung des Teils zur Folge hat. Einmal begonnen, ist dieser Vorgang nicht aufzuhalten.

Bei den heute üblichen Druckguss-Legierungen tritt dieser Schaden nicht mehr auf.

Helmut Steiner



**Gratis
Frontplatten
Designer**

SIE DESIGNEN – WIR FERTIGEN

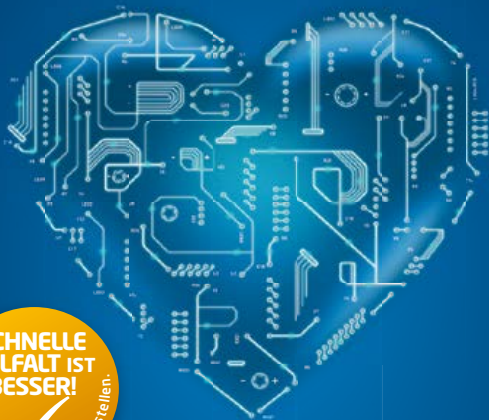
Frontplatten in Profiqualität

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

WIR LIEBEN, WAS WIR MACHEN.

LEITERPLATTEN MIT LEIB UND SEELE.



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

LEITON
RECHNEN SIE MIT BESTEM SERVICE

Mit ganzem Herzen bei der Sache. Die Fertigung hochwertiger Leiterplatten betreiben wir tagtäglich mit größtem Engagement und Leidenschaft. Herzblut inklusive. Damit haben wir uns nach der Gründung 2004 in kürzester Zeit am Markt etabliert. Heute ist das dynamische Team mit langjähriger Branchenerfahrung flexibel und servicestark genug, um auch Ihre ausgefallenen Wünsche erfüllen zu können. **Wir lieben was wir tun** und freuen uns jeden Tag erneut über die Aufgaben, die Sie uns stellen. Lassen Sie sich von einer großen Auswahl an Lösungen begeistern. Was Sie nicht selbst online kalkulieren können, lösen wir gerne **persönlich für Sie**. Das bedeutet: Sie können bei LeitOn immer mit bestem Service rechnen.

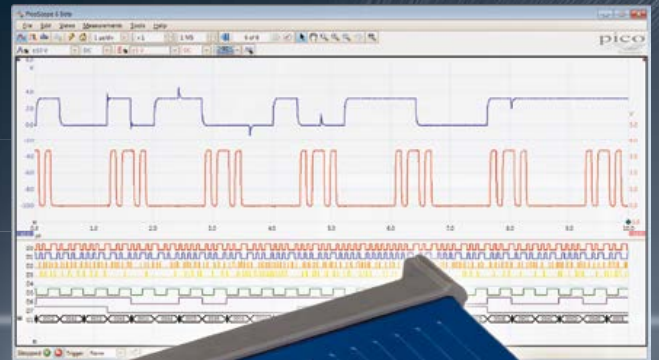
www.leiton.de

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

MIXED-SIGNAL OSZILLOSKOPE

**4 ANALOG- +
16 DIGITAL-KANÄLE**

**SCHNELLES DEBUGGING
KOMPLEXER, GEMISCHT-
ANALOGER/DIGITALER DESIGNS**



- **USB 3.0**
- **ULTRA-TIEFER SPEICHER**
- **SEGMENTIERTER SPEICHER**
- **SCHNELLE TRIGGER**

STANDARDMÄßIG INKL.: AUTOMATISCHE MESSUNGEN,
SPEKTRUM-ANALYSATOR, SDK, ERWEITERTE TRIGGER,
FARBIGES NACHLEUCHTEN, SERIELLES DECODING
(CAN, LIN, RS232, I²C, I²S, FLEXRAY, SPI), MASKENTEST,
MATHEMATIK-KANÄLE. KOSTENFREIE UPDATES.

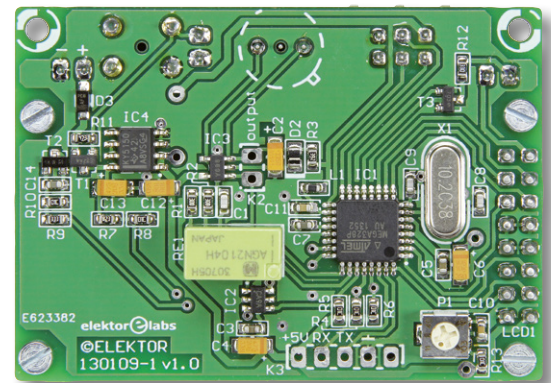
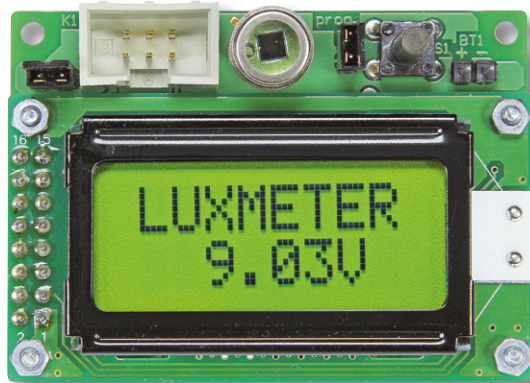
5 JAHRE GEWÄHRLEISTUNG

	3204D MSO	3205D MSO	3206D MSO	3404D MSO	3405D MSO	3406D MSO
Kanäle	2 analog, 16 digital			4 analog, 16 digital		
Bandbreite	60 MHz	100 MHz	200 MHz	60 MHz	100 MHz	200 MHz
Puffer-Speicher	128 MS	256 MS	512 MS	128 MS	256 MS	512 MS
Max. Sampling-Rate	1 GS/s					
Signalgenerator	Funktionsgenerator + Arbiträr-Signalgenerator					
Digital-Eingänge	100 MHz max. Frequenz, 500 MS/s max. Sampling-Rate					

www.picotech.com/PS363

Luxmeter

1 lx bis 100 klx



Von
Karl-Anton Dichtel
(D)

Die EU hat mittlerweile ja fast alle herkömmlichen Glühlampen verboten. Übrig bleiben Halogen-, „Energiespar-“ und LED-Lampen, die tatsächlich einen besseren Wirkungsgrad haben. Will man aber wissen, um wieviel besser, dann muss man sich erst einmal auf die Herstellerangaben verlassen. Doch Vertrauen ist gut – selbst messen ist besser. Unser Luxmeter hilft!

Gerade bei den „billigen“ LED-Lampen fernöstlicher Provenienz, die teilweise deutlich günstiger als Markenware angeboten werden, ist Vorsicht sicherlich nicht die schlechteste Strategie. Unter den aktuell 500.000 Angeboten bei einem bekannten Auktionshaus sind z.B. gleich fünf 3-W-Spots für 1 Euro. Behauptet wird, dass diese einen Lichtstrom von 210 lm abgeben. Das klingt zwar realistisch, aber stimmt das auch? Um solche Behauptungen nicht einfach glauben zu müssen, benötigt man ein Messgerät, das Helligkeit zuverlässig erfassen kann. So ein Messgerät ist nicht nur zum generellen Lampen-Check interessant. Man kann damit gut überprüfen, ob es an einem bestimmten Arbeitsplatz hell genug ist, denn (nicht nur) in der EU gibt es auch dafür Vorschriften.

Bei Lampen hat ein Luxmeter noch einen weiteren Nutzen: Für bestimmte Zwecke werden aus vielerlei Gründen gerne Halogenlampen eingesetzt. Und diese neigen bei zunehmender Betriebsdauer zum Rückgang der Helligkeit, unter anderem durch Metaldampfniederschlag am Glaskolben. Irgendwann profitiert man dann eher von der Heizleistung als vom Licht. Selbst Energiesparlampen

werden mit der Zeit dunkler. Eine Messung hilft bei der Entscheidung, wann man das ineffiziente Leuchtmittel austauschen sollte.

Licht messen

Das zentrale Element eines jeden Messgeräts ist sein Sensor, der die gewünschte physikalische Größe möglichst proportional in eine elektronisch leicht zu messende Spannung umwandelt. Heutzutage ist zwar schon häufig etliche Elektronik in den Sensor integriert und dieser liefert dann gleich digitale Werte. Doch der klassische Sensor für Licht ist eine Fotodiode. Neben Exemplaren für die Übertragung von IR-Signalen gibt es auch spezielle, genauere Ausführungen für Messzwecke. Der hier eingesetzte Typ BPW21R ist solch ein Exemplar, das passenderweise auch mit stabilem TO-5-Metallgehäuse mit Glasfenster (**Bild 1**) hergestellt wird.

So eine hermetisch versiegelte Fotodiode ist mit einer aktiven Fläche von 7,5 mm² sehr groß und daher sehr lichtempfindlich. Außerdem ist der Sensor sehr linear – siehe Datenblatt [1]. **Bild 2** illustriert die geradezu unglaubliche Linearität des Zusammenhangs von Kurzschlussstrom und

Beleuchtungsstärke. Die doppelt logarithmische Darstellung ist notwendig, da der Sensor einen extrem großen Helligkeitsbereich von gut sieben Größenordnungen abdeckt! Obwohl es sich bei diesem Typ um ein klassisches Bauelement handelt, können sich die technischen Daten nach wie vor sehen lassen, sodass dieser Lichtsensor gerade für den Selbstbau eines Luxmeters ideal ist.

Eine Fotodiode liefert einen Strom, welcher der Beleuchtungsstärke in Lux und nicht dem für Lampen typischerweise angegebenen Lichtstrom in Lumen proportional ist. Da $1 \text{ lx} = 1 \text{ lm/m}^2$ ist, lässt sich die eine Größe prinzipiell leicht in die andere umrechnen, doch hängt die von einer Lichtquelle bestrahlte Fläche einerseits vom Abstand und andererseits vom Abstrahlwinkel ab. Dank Internet muss man hierzu kein Blatt Papier mehr zücken und Formeln umstellen: Auf diversen Webseiten (z.B. [2]) gibt es fertige Rechner, in die man nur noch die Werte eingeben muss, um wahlweise Lux in Lumen und umgekehrt zu transformieren.

Messverfahren...

Wie schon erwähnt liefert die Fotodiode einen so genannten „Kurzschlussstrom“. Der Zusammenhang mit der Beleuchtungsstärke ist nur dann sehr linear, wenn die Diode quasi kurzgeschlossen wird bzw. die den Strom erfassende Elektronik einen sehr geringen Eingangswiderstand aufweist. Nun einfach aber brutal z.B. einen $1\text{-}\Omega$ -Widerstand parallel zum Sensor zu schalten und die abfallende Spannung zu messen ist wenig zielführend, da die Spannung bei 1 lx nur etwa 9 nV betragen würde. Die ADCs gängiger Mikrocontroller – und die Messung mit solchen Controllern ist die Methode der Wahl – sind für so niedrige Spannungen nicht geeignet. Mit der 10-bit-Auflösung eines typischen AVR-Controllers ergibt sich bei einer Referenzspannung von 1 V eine Auflösung von rund 1 mV – das sind fünf Größenordnungen mehr als der Endausschlag des kleinsten Messbereichs! Auch eine zwischengeschaltete Spannungsverstärkung um den Faktor $100.000.000$ (um bei 1 lx auf etwa 1 V zu kommen) ist illusorisch. Denn damit ergäben sich fast pures Rauschen und andere Komplikationen. So geht es also nicht.

Die Schaltung in **Bild 3** zeigt, wie es geht: Der Sensor D1 liegt zwischen den beiden Eingängen eines Opamps (IC3), dessen nichtinvertierender Eingang mit R3/D2 auf eine Hilfsspannung von

Technische Daten

- Luxmeter mit 6 Messbereichen:
 1 lx , 10 lx , 100 lx , 1 klx , 10 klx und 100 klx
- Kleinster Messwert im 1-lx -Bereich: $0,01 \text{ lx}$
- Automatische Umschaltung der Messbereiche
- Automatische Abschaltung nach einer Minute
- Ausgabe der Messergebnisse per serieller Schnittstelle
- Nur handelsübliche Bauteile
- Geringe Stromaufnahme
- Kalibrierbar über serielle Schnittstelle (Terminalprogramm)

etwa $0,65 \text{ V}$ gelegt wird. Wird D1 beleuchtet, möchte sie einen Strom liefern, weswegen die Spannung am invertierenden Eingang von IC3 sinken würde, wäre da nicht die Gegenkopplung über R1 bzw. R2. Folglich wird die Spannung am Ausgang des Opamps so viel positiver, dass der über den jeweiligen Gegenkopplungswiderstand fließende Strom den von D1 voll kompensiert und die Spannung zwischen den Eingängen von IC3 0 V beträgt. D1 ist also praktisch kurzgeschlossen und operiert damit äußerst linear. Um möglichst präzise zu messen, wird für IC3 kein Standard-Opamp, sondern mit dem Typ MCP6061T von Microchip ein Exemplar gewählt, das mit winzigen Eingangsströmen im pA -Bereich und einer Offsetspannung von nur $150 \text{ }\mu\text{V}$ die Messung nicht beeinträchtigt. An K2 liegt zu Kontrollzwecken das von IC3 gewandelte Sensorsignal an.

...und Schaltung

Bei 1 lx liegen (RE1B in Stellung R1) am Ausgang von IC3 etwa $1,8 \text{ mV}$, da $U = 9 \text{ nA} * 200 \text{ k}\Omega$. Das ist für den ADC eines normalen Mikrocontrollers immer noch zu wenig. Hier könnte man noch verstärken oder – wie es sich für ein präzi-

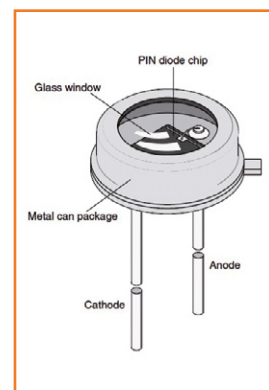


Bild 1.
Schematische Darstellung des BPW21R-Gehäuses.

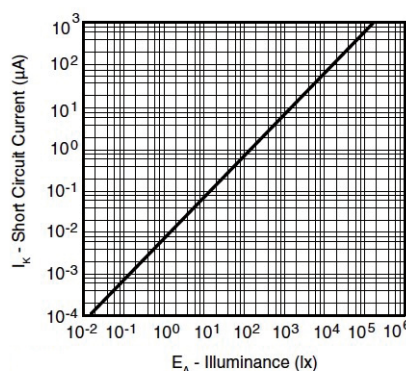


Bild 2.
Zusammenhang zwischen Kurzschlussstrom I_K und Beleuchtungsstärke E_A einer BPW21R.

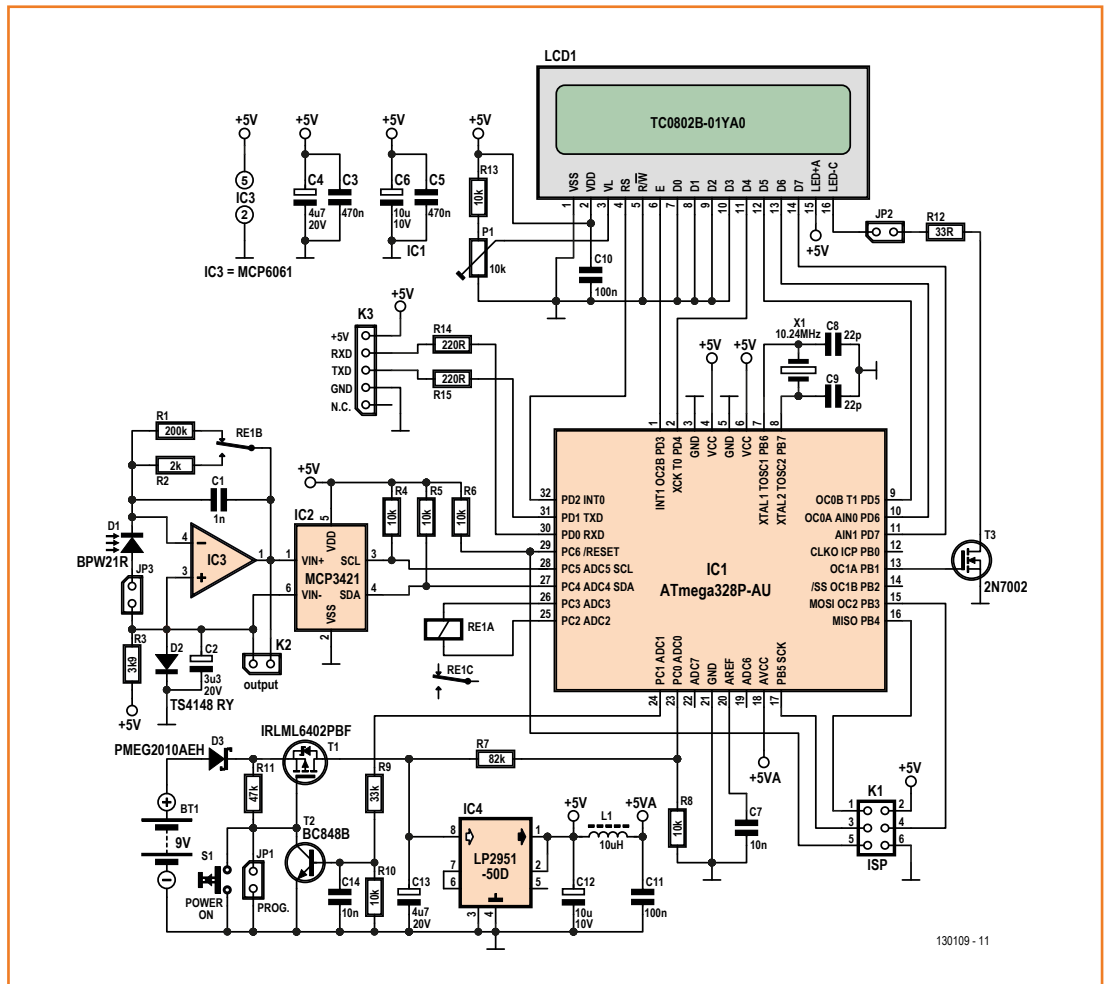


Bild 3.
Die Schaltung des
Luxmeters besteht im
wesentlichen aus Sensor,
Signalverstärker, Präzisions-
ADC, Mikrocontroller und
Display.

ses Messgerät geziemt – gleich Nägel mit Köpfen machen: Statt des im verwendeten ATmega328 integrierten So-lala-ADCs wurde mit IC2 ein extra AD-Umsetzer vorgezogen, der über eine integrierte hochqualitative Vorverstärkung und eine Auflösung von 18 bit verfügt. Letzteres klingt nach Overkill und es werden auch „nur“ 15 bit genutzt, aber diese Konfiguration hat Gründe. Ein zentraler Punkt ist dabei der Messbereich: Dem menschlichen Auge entsprechend reicht dieser von 1 lx (= Vollmond) bis 100 klx (= volles Sonnenlicht). Dies entspricht vollen fünf Größenordnungen und verlangt einen extremen Dynamikumumfang der Elektronik. Dieser Anforderung wird wie folgt begegnet: Die Umschaltung zwischen R1 und R2 ändert den Pegel um den Faktor 100. Die nachfolgende Schaltung muss also nur noch mit einem Dynamik-Umfang von 1:1.000 klar kommen. Da die interne Verstärkung von IC2 zusätzlich noch zwischen einfach und achtfach umgeschaltet werden kann, erreicht man mit dem

resultierenden Dynamikumumfang genaue Messungen im kompletten Messbereich. Die Details der Umschaltung zwischen den Verstärkungen und Messbereichen kann man dem Code der Firmware entnehmen. Die Genauigkeit der Elektronik liegt jedenfalls bei etwa 0,5 %.

IC2 wird über I²C vom Mikrocontroller IC1 angesteuert. Hierüber werden die Daten ausgelesen und IC2 konfiguriert. IC1 steuert außerdem auch das Relais mit zwei Ausgängen und kann auf diese Weise den Strom durch die Wicklung umkehren. Die direkte Ansteuerung ist möglich, da RE1 mit etwa 24 mA Strom auskommt. RE1 ist ein so genanntes bistabiles Relais, das nur kurz Strom in der einen Richtung benötigt, um umzuschalten und dann stromlos in dieser Position zu verharren. Zurückgeschaltet wird einfach durch einen kurzen umgekehrten Stromimpuls. Diese Methode spart Batterien. Ein Relaiskontakt ist an dieser Stelle jedem MOSFET vorzuziehen.

IC1 hat noch drei weitere Aufgaben: Einige seiner Pins steuern das LCD mit Nibbles und ein weiterer Pin schaltet über T3 während einer Messung bei Bedarf die Hintergrundbeleuchtung ein. Zwei Pins übernehmen den seriellen Datentransfer via TTL-Pegel. Der Header K3 ist dabei so belegt, dass er zum USB/Seriell-Konverter BOB von Elektor passt. So erhält man bei Bedarf eine serielle Schnittstelle via USB.

Eine weitere Funktion von IC1 ist, dass nach Einschalten mit dem Taster S1 die Stromversorgung über Pin 24 und T2/T1 für eine Minute gehalten wird. Anschließend schaltet sich das Messgerät Batterien schonend selbst aus.

Noch ein Wort zur Stromversorgung: Da so ein Messgerät praktischerweise batteriegespeist ist, lohnen sich Energiesparmaßnahmen. Eine davon ist die schon erwähnte Steuerung der Hintergrundbeleuchtung des LCDs via T3 (bzw. Abschaltung über JP2). Die Beleuchtung wird nur automatisch aktiviert und an die Umgebungshelligkeit angepasst, wenn letztere unter 64 lx liegt. Diese clevere Maßnahme alleine spart im Hellen schon etwa 20 mA, was angesichts der sonstigen Stromaufnahme von 14 mA während der Messung nicht wenig ist. Zur Erzeugung stabilisierter 5 V wurde zudem kein energieverschwendender Standardregler, sondern mit IC4 ein spezieller Low-Power/Low-Drop-Regler eingesetzt. Der Regler funktioniert noch bis hinab zu einer Batteriespannung von etwa 5,5 V. Darunter bricht seine Ausgangsspannung ein. Unter 5 V fällt das Display aus. L1 filtert die Versorgung für den Analogteil von IC1 zusätzlich.

Aufbau

Bei der Bestückung der Platine braucht man eine ruhige Hand, eine feine Lötspitze, eine feine Pinzette, Lötpaste (oder dünnes Lötzinn) und eine Lupe, denn auch wenn es sich beim Sensor um ein gutes altes bedrahtetes Bauteil handelt, sind sonst fast durchweg SMD-Bauteile zu sehen – bei Widerständen und Kondensatoren meistens in der besonders „beliebten“ Ausführung 0603. Anfängern sei daher ausdrücklich nahegelegt, sich bei Lötversuchen durch einen erfahreneren Zeitgenossen unterstützen zu lassen. **Bild 4** zeigt die Oberseite der Platine eines Prototypen – das müsste als „Abschreckung“ genügen. Eine Lupe hilft bei der Suche nach Kurzschlüssen zwischen den Pins der SMD-ICs, die sich mit Entlötlitze und Flussmittel (bei den Pin-Abständen der hier

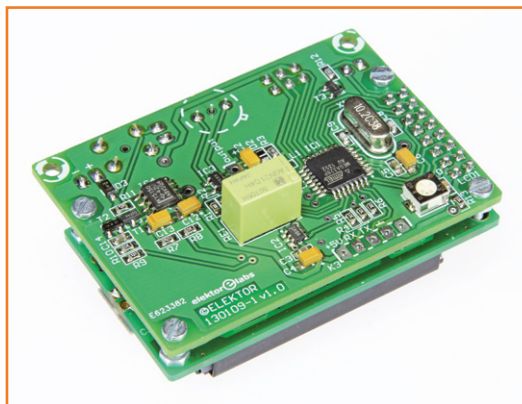


Bild 4.
Die Oberseite der bestückten Prototypen-Platine. Hier befinden sich (fast) alle Bauteile – auch die „schön kleinen“ SMDs vom Typ 0603.

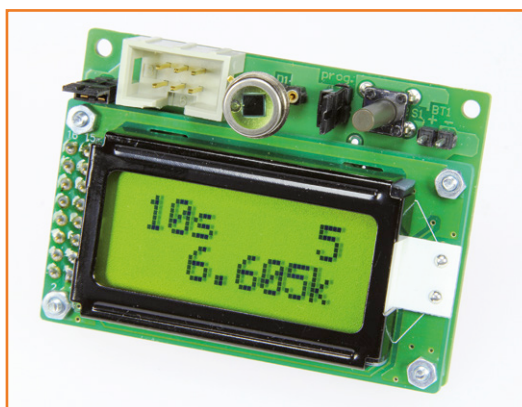


Bild 5.
Platine mit montiertem Display-Modul. Oben sieht man die Fotodiode und den Taster zum Einschalten.

verbauten ICs noch recht gut) beseitigen lassen. Das Display kommt auf die andere Seite der Platine (siehe **Bild 5**). Will man das LCD-Modul abnehmbar montieren, so kann man die Verbindung mit Pfostenbuchsen und langen Pfostensteckern realisieren. Wer darauf keinen Wert legt, kann aber auch einfach 16 passende Drahtstücke nehmen. Wichtig ist dabei, dass das Display genügend Abstand hat, damit keine ungewollten elektrischen Verbindungen zwischen der Display-Unterseite und der Platine entstehen können.

Firmware

Die Software für IC1 wurde mit der kostenlosen IDE AVR Studio (v4.18) in C geschrieben und ist samt fertiger Hex-Datei auf der Elektor-Webseite zu diesem Projekt [4] kostenlos verfügbar. Vom 10,24-MHz-Takt wird zunächst ein 5-ms-Takt für den Ansteuer-Impuls des Relais abgeleitet und daraus dann ein 0,5-s-Takt für die Ablaufsteuerung. Der Code für die LCD-Ansteuerung ist einem Beitrag der Webseite mikrocontroller.net [5] entnommen und dort ausgiebig kommentiert. Der I²C-Programmteil ist ein Recycling eines

Projekts für eine Wetterstation, weshalb sich an dieser Stelle noch ungenutzte Code-Reste für einen Dreh-Encoder etc. finden. Wer an weiteren Details interessiert ist, sei auf die Kommentare im Code verwiesen.

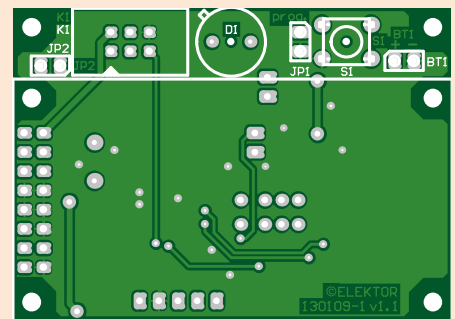
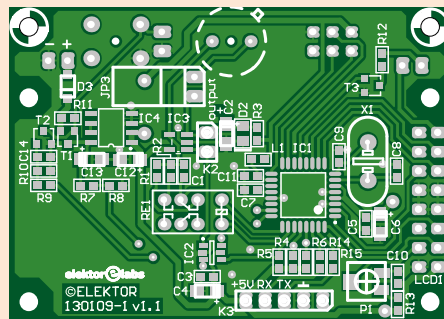
Zum „Brennen“ von IC1 dient die standardisierte sechspolige ISP-Schnittstelle K1. Damit der Mikrocontroller auch während des Brennvorgangs versorgt wird, muss eine Batterie angeschlossen und JP1 gesteckt sein. Alternativ kann man natürlich auch so lange S1 betätigen. Nach dem Brennen startet die Firmware zum ersten Mal.

Da alle Bytes im EEPROM-Speicher eines neuen Mikrocontrollers den Wert \$FF aufweisen und beim erneuten Brennen auch wieder auf diesen

Wert zurück gesetzt werden, kann der Mikrocontroller einfach feststellen, ob das Messgerät zum ersten Mal startet. Wird ein Wert von FF_{hex} in zwei EEPROM-Bytes gefunden, wird vorläufig ein Korrekturfaktor von 1 in diese beiden Bytes geschrieben. Dies entspricht einer Umrechnung von 9 nA in 1 lx. Eine genauere Kalibration kann später noch via serielle Schnittstelle vorgenommen werden.

Als nächster Schritt wird dann der Offset von Opamp und ADC ohne Fotodiode gemessen (daher zuvor JP3 abziehen), im Display angezeigt und zu Korrekturzwecken im EEPROM abgelegt.

Ab jetzt wird das Messgerät jede halbe Sekunde



Stückliste

Widerstände:

(1 %, 0,1 W, SMD 0603)
 R1 = 200 k
 R2 = 2 k
 R3 = 3k9
 R4..R6,R8,R10,R13 = 10 k
 R7 = 82 k
 R9 = 33 k
 R11 = 47 k, 5 %
 R12 = 33 Ω, 5 %, 0,2 W
 R14,R15 = 220 Ω, 5 %, 0,2 W
 P1 = 10k, 20 %, SMD-Trimpoti (Vishay TS53YJ103MR10)

Kondensatoren:

(10 %, 10 V, SMD 0603)
 C1 = 1 n, 5 %, 50 V, COG/NPO
 C2 = 3μ3, 20 V, SMD Case A, Tantal
 C3,C5 = 470 n
 C4,C13 = 4μ7, 20 V, SMD Case A, Tantal
 C6,C12 = 10 μ, SMD Case A, Tantal
 C7,C14 = 10 n, 50 V
 C8,C9 = 22 p, 50 V, COG/NPO
 C10,C11 = 100 n, 25 V

Spule:

L1 = 10 μH, 20 %, 250 mA, 1,05 Ω, SMD 0603

Halbleiter:

D1 = BPW21R, TO-5
 D2 = TS4148 RY, SMD 0805
 D3 = PMEG2010AEH, SMD SOD-123F
 T1 = IRLML6402PBF, SMD SOT-23
 T2 = BC848B, SMD SOT-23
 T3 = 2N7002, SMD SOT-23
 IC1 = ATmega328P-AU, SMD TQFP 32A
 IC2 = MCP3421A0T-E/CH, SMD SOT-23-6
 IC3 = MCP6061T-E/OT, SMD SOT-23-5
 IC4 = LP2951-50D, SMD SO-8

Außerdem

K1 = 2x3-Stiftleiste, RM 2,54 mm
 BT1,JP1,JP2,K2 = 2x1-Stiftleiste, RM 2,54 mm (K2 optional)
 JP3 = 2x1-Stiftleiste, RM 2,54 mm, abgewinkelt
 K3 = 5x1-Stiftleiste, RM 2,54 mm, abgewinkelt
 JP1,JP2,JP3 = Jumper, RM 2,54 mm
 S1 = Taster, 6x6 mm, vertikal, Schließer
 BT1 = Clip für 9-V-Batterie mit Litzen und optional Buchsen für Stiftleisten
 RE1 = Relais, bistabil, doppelpolig, Spule 4,5 V / 202,5 Ω, Kontakte 1 A / 110 VDC (Panasonic AGN2104H)
 X1 = Quarz, 10,24 MHz, 18 pF, HC-49/4H
 LCD1 = LCD-Modul, 2 x 8 Zeichen, hintergrundbeleuchtet, 58 x 32 mm, TC0802B-01YA0 (Elektor 120061-75)
 Platine EPS 130109-1

einen neuen Messwert anzeigen und jede Sekunde an der seriellen Schnittstelle einen neuen Wert bereitstellen. Nach den eigentlichen Messwerten wird seriell jeweils noch ein Semikolon als Trennzeichen übertragen, damit man mit einem PC empfangene Daten einfach in eine Tabellenkalkulation kopieren kann. Nach Ablauf einer Minute schaltet die Firmware das Messgerät wieder aus. Beim nächsten Start durch Druck auf S1 ist das EEPROM nicht mit FF_{hex} belegt und es geht direkt zur Messung der Helligkeit. Hierzu muss dann allerdings JP3 gesteckt sein.

Sonstiges

Nach einem Druck auf S1 wird zunächst die Batteriespannung gemessen und angezeigt. Nach einer Sekunde wird dann die Beleuchtungsstärke gemessen. In Bild 5 kann man gut sehen, wie die Messergebnisse angezeigt werden: In der oberen Zeile des Displays stehen links die abgelaufenen Sekunden und rechts der Messbereich. Das Luxmeter schaltet automatisch zwischen seinen sechs Messbereichen um. Rechts oben sieht man deshalb eine „5“ zum Zeichen, dass der Messbereich 10 klx gewählt wurde, was zum Messergebnis „6.605k“ (Lux) in der unteren Zeile passt. Die Datenübertragungsrate der seriellen Schnittstelle ist mit 38,4 kbd ein guter Kompromiss zwischen Tempo und Störsicherheit.

Um zu wissen, wie sich das selbstgebaute Luxmeter schlägt, wurde es mit einem kommerziellen Produkt der Firma Testo verglichen. Im Vergleich mit dem Modell Testo 540 gab es auch ohne Kalibration nur begrenzte Unterschiede im Bereich von 3...4 % der Messwerte. Bei schwacher Beleuchtung zeigte es 108 lx statt 105 lx an und bei hellem Licht 65 klx statt 63 klx. Im Büro (bei Licht durch Leuchtstofflampen) wurden die Unterschiede größer und erreichten mit 1.115 lx statt 1.600 lx durchaus 30 %. Das kann an der eher warmen Lichtfarbe gelegen haben oder auch daran, dass der Sensor des Testo 540 durch eine kleine Milchkugelpuppe abgedeckt ist – wobei unklar bleibt, welche Messung die bessere ist.

Auf jeden Fall bleibt festzuhalten, dass man mit diesem selbstgebauten Luxmeter ein vielfältig einsetzbares Messgerät erhält, das vor allem bei zusätzlicher Kalibration keinen Vergleich mit kommerziellen Produkten scheuen muss.

(130109)

Kalibrierung

Für eine Kalibrierung kann man entweder eine Referenzlichtquelle bei bekanntem Abstand oder ein anderes Lichtmessgerät verwenden. Ergibt eine Messung als extremes Beispiel, dass das Luxmeter statt realer 410 lx mit 820 lx das Doppelte anzeigt, ergäbe sich ein Korrekturfaktor von 0,5. Da der Korrekturfaktor fünfstellig als ASCII-String seriell übertragen wird, ergibt sich hieraus ein Korrekturausdruck von „05000“.

Ein anderes, realistischeres Beispiel: Bei einer Anzeige des Luxmeters von nur 400 lx ergäbe sich ein Korrekturfaktor von 410 lx / 400 lx = 1,025. Der zugehörige ASCII-String wäre dann „10250“.

Ist das Luxmeter eingeschaltet und per serieller Schnittstelle mit einem PC verbunden, auf dem die sekundlich laufenden Daten im Fenster eines Terminal-Programms zu sehen sind, kann man loslegen: Die Firmware des Luxmeters wartet nämlich im Hintergrund auf die Eingabe eines einzigen Zeichens. Ist dieses Zeichen „1“, schaltet die Firmware in den Modus zur Eingabe des Korrekturfaktors. Hierzu sieht man im Terminal-Programm zunächst die Ausgabe von:

```
Mode 1: Calibration - Input 5 digits or ESC with #
actual calibration factor = 10000
Input Calibration:
```

Jetzt kann man den zuvor berechneten String für den neuen Korrekturfaktor eingeben. Nach Eingabe des fünften Zeichens geht es mit den normalen Messungen weiter. Der Screenshot zeigt einen Fensterausschnitt des Programms HTerm.

Damit der neue Korrekturfaktor übernommen wird, muss man den Controller reseten. Dies geht z.B. über eine kurze Verbindung der Pins 5 und 6 von K1. Die Prozedur kann fast beliebig oft vorgenommen werden. Der Korrekturfaktor bleibt so lange erhalten, bis ein neuer eingegeben oder eine neue Firmware in den Controller gebrannt wird.

Time(s)	Range	Lux
1	10	15
2	10	25
3	10	30
4	10	35
5	10	40
6	10	45
7	10	50
8	10	55

```

Mode 1: Calibration - Input 5 digits or ESC with #
actual calibration factor = 1024000
Input Calibration: 05
Input Cal. Factor = 00001000
Input Cal. Factor = 00010000
Input Cal. Factor = 00100000
Input Cal. Factor = 01000000
Input Cal. Factor = 10000000
New Calibration Factor = 10000000

24:00:00 10 150.12 10000000
25:00:00 10 150.63 10000000
26:00:00 10 151.62 10000000
27:00:00 10 150.22 10000000
28:00:00 10 159.11 10000000
29:00:00 10 160.62 10000000
30:00:00 10 160.72 10000000
31:00:00 10 157.11 10000000
32:00:00 10 157.62 10000000
33:00:00 10 161.12 10000000
34:00:00 10 166.22 10000000
35:00:00 10 164.52 10000000
36:00:00 10 127.82 10000000

```

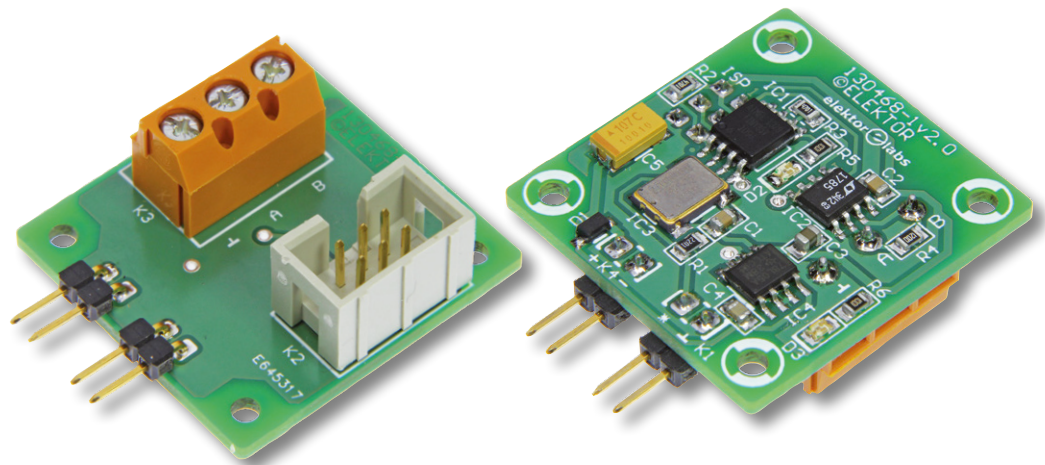
Weblinks

- [1] BPW21R-Datenblatt: www.vishay.com/docs/81519/bpw21r.pdf
- [2] Lux in Lumen umrechnen: www.leds.de/Lumen-zu-Lux/
- [3] Pegelkonverter: www.elektor.de/ft232r-usb-serial-bridge-bob-110553-91
- [4] Projekt-Webseite: www.elektor-magazine.de/130109
- [5] www.mikrocontroller.net/articles/AVR-GCC-Tutorial/LCD-Ansteuerung

Temp-Sensor-Board Mit RS485-Interface

Entwicklung:
André Goldberg,
Mauk van der Laan
und **Ton Giesberts**

Text: **Jens Nickel**



Für viele Automatisierungsaufgaben benötigt man Temperatursensoren. Per RS485 lassen sich die Werte zuverlässig auch über längere Strecken übertragen. Unser kompaktes Temp-Sensor-Board ist mit einem ATtiny-Controller und einem RS485-Treiber ausgestattet; es können mehrere Temperatursensoren parallel angeschlossen werden. Dazu bieten wir eine Standardfirmware an, die zur Übermittlung der Messwerte das ElektorBus-Protokoll nutzt. PC-Software zur Anzeige der Daten rundet das Projekt ab.

Die ElektorBus-Artikelserie in 2011 rief bei vielen Lesern großes Interesse hervor. Wir bekamen Hunderte von Mails, die eigene Entwicklungen beschrieben oder wertvolle Hinweise enthielten. Einer der fleißigsten ElektorBus-Fans war André Goldberg, der eine Raumtemperatur-Überwachung und -Steuerung realisierte. Ein wichtiger Bestandteil waren dabei ElektorBus-Knoten mit Temperatursensoren, die Messwerte an einen Rechner übermittelten.

Der Charme des ElektorBus-Protokolls ist seine Einfachheit: Jede ElektorBus-Nachricht ist grundsätzlich 16 Bytes lang. Als „Nutzlast“ kann ein Sensor-Board zum Beispiel vier Ganzzahlen (-1023 bis 1023) an eine Steuerzentrale übermitteln, wie die Werte von vier angeschlossenen Temperatursensoren. Mit einer Nachricht in umgekehrter Richtung lässt sich der Sensor-Knoten zum Beispiel anweisen, ab jetzt Messwerte in Grad Celsius beziehungsweise Grad Fahrenheit zu übertragen; auch ein Konfigurieren eines Messintervalls ist möglich. Für all das sieht das

ElektorBus-Protokoll entsprechende Kommandos vor [1].

Zur Übertragung der Bytes wird RS485 genutzt, das im Halbduplex-Verfahren zwei Leitungen benötigt, auch eine Masseleitung sollte vorhanden sein (siehe hierzu [2]). RS485 ist prinzipiell recht unempfindlich gegen Störungen. Die moderate ElektorBus-Taktrate von 9600 Baud trägt ebenfalls dazu bei, dass lange Leitungen (über 30 m) möglich werden.

Mini-Busknoten

In Elektor haben wir schon einige Schaltungen und Platinen von ElektorBus-Knoten veröffentlicht [3][4]. Für manche Anwendungen sind die Boards aber zu groß. André Goldberg machte sich daher Gedanken, wie eine möglichst kompakte Platine aussehen könnte. Der Platinen-Entwurf, den er unserem Labor zukommen ließ, misst nur 18 mm x 26 mm. Integriert ist ein ATtiny45-Controller im SMD-Gehäuse, der sechs GPIO-Pins zur Verfügung stellt. Auch der RS485-Treiberchip ist selbst-

verständlich ein SMD-Bauteil. Zum Anschluss der Busleitungen sind Lötinseln vorhanden. Die ISP-Programmierspins und zwei GPIO-Pins des Controllers wurden ebenfalls nur an Löt pads geführt, um Platz zu sparen. Auf einen externen Oszillator hat André verzichtet. Als Temperatursensor wählte er den 1-Wire-Temperatursensor DS18S20. Dieser belegt nur einen der GPIO-Pins, da der Sensor den ermittelten Temperaturwert über ein asynchrones Protokoll zum Mikrocontroller übermittelt, also nur eine Datenleitung und keine Taktleitung nötig ist (siehe Datenblatt [5]). Über die Datenleitung kann man die Sensoren auch stromversorgen. Es lassen sich sogar mehrere der Eindraht-Sensoren parallel an denselben Portpin anschließen. Jeder der Sensoren besitzt hierzu eine vom Hersteller eingebrannte 64-bit-ID. Über diese ID kann der Mikrocontroller den Temperaturwert eines bestimmten Sensors gezielt auslesen. Beim DS18S20 sind das 9-bit-Werte, die Auflösung beträgt dabei 0,5 °C.

Herausforderungen

Aufgrund des vorgeschlagenen Hardware-Designs ergaben sich verschiedene Anforderungen an die Software:

- Der ATtiny45 besitzt keinen Hardware-UART, mit dem der RS485-Treiberchip angesteuert werden könnte. Vielmehr muss man den UART (Senden und Empfangen) per Software

realisieren; es lassen sich dann grundsätzlich zwei beliebige GPIO-Pins des Controllers verwenden.

- Man benötigt eine Bibliothek zum Auslesen der Eindraht-Sensoren. Für das Lesen und Schreiben der Daten wird ein weiterer GPIO-Pin benutzt. Da aber mehrere Sensoren an der gleichen Leitung hängen können, müssen dem Controller die IDs der einzelnen Sensoren bekannt sein, um die Ergebnisse zuzuordnen.
- Der ATtiny besitzt nicht genügend Programmspeicher, um zusätzlich die bereits in Elektor vorgestellte ElektorBus-Protokoll-Bibliothek verwenden zu können [6][7]. Die nötigen Teile des Protokolls muss man also von Hand implementieren.

André Goldberg entschied sich dafür, zwei Open-Source-Bibliotheken für den Software-UART und den 1-Wire-Bus zu verwenden. Die Adressen der einzelnen T-Sensoren wurden erst (mit einer anderen Firmware) ausgelesen und dann hartcodiert in die Software eingefügt. Um vier Temperaturwerte in Grad Celsius zu übermitteln, multiplizierte das Programm die Werte erst mit 10 und codierte sie dann als vier Ganzzahlen gemäß ElektorBus-Protokoll. Auch die übrigen Bytes der ElektorBus-Nachricht wurden in der Software „manuell“ zusammengestellt.

Stückliste

Widerstände (0805):

R1 = 2,2 k
R2,R3 = 4,7 k
R4 = 120 Ω
R5,R6 = 1 k

Kondensatoren:

C1,C2,C4 = 100 n, 25 V, 10 %, X7R, SMD 0805
C3 = 10 µ, 25 V, 10 %, X5R, SMD 0805
C5 = 100 µ, 16 V, 20 %, Tantal, SMD Case F

Halbleiter:

D1 = PMEG2010AEH
D2 = LED gelb, SMD 0805
D3 = LED grün, SMD 0805
IC1 = ATtiny85-20SU, SMD SO-8S2
IC2 = LT1785, SMD SO-8
IC3 = 8 MHz Quarzoszillator, 5 x 7 mm, SMD (LF SPX0019079)
IC4 = 78L05

Außerdem:

K1,K4 = 2x1-Stiftleiste, Rastermaß 2,54 mm, rechtwinklig
K2 = 2x3-Stiftleiste oder 2x3-Wannenstecker, Rastermaß 2,54 mm
K3 = 3x1-Schraubklemme, Rastermaß 5,08 mm
PCB 130468-1 v2.0

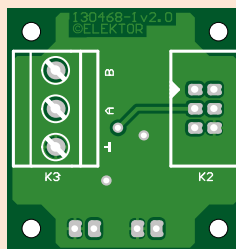
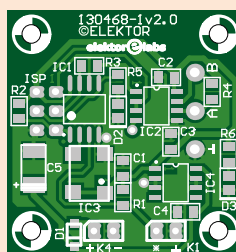


Bild 1.
Die einseitige Platine misst nur rund 29 x 32 mm. Die Steckverbinder werden von unten bestückt.

Schaltung

Im Elektor-Labor dachte mein Kollege Ton Giesberts alsbald über eine „elektorisierte“ Platine nach. Wir entschieden uns, für alle Anschlüsse Stiftleisten beziehungsweise Schraubklemmen vorzusehen, die aber nur bei Bedarf eingelötet werden. Auch einen Quarzoszillator haben wir integriert, was die Übertragung sicherer macht, gerade wenn es im harten Praxiseinsatz zu großen Temperaturschwankungen kommt. Ton gelang es, die Platine dennoch klein zu halten. Rund 31 x 32 mm misst das Ergebnis, das man in **Bild 1** sieht [8]. Die Platine ist doppelseitig; die ElektorBus-Schraubklemmen, die 2x3-Stiftleiste zum Programmieren, die beiden gewinkelten 1x2-Stiftleisten zum Anschluss der Eindraht-Sensoren und der Stromversorgung werden von unten bestückt. Die SMD-Bauteile lassen sich noch von Hand löten. Unserem freien Mitarbeiter Mauk van der Laan gelang dann noch eine entscheidende Verbesserung der Schaltung. Statt zwei der sechs GPIO-Pins für das Senden und Empfangen des Software-UARTs zu verbraten, benötigen wir nun nur noch einen Pin, der wechselweise zum Senden und zum Empfangen genutzt wird. Denn wegen der Halbduplex-Kommunikation auf dem RS485-Bus darf ein Busknoten ja nie gleichzeitig zuhören und sprechen; Kollisionen der Nachrichten müssen vermieden werden. Vorteil: Jetzt wurde ein GPIO-Pin frei, an den wir eine LED anschließen konnten, die Zustände signalisiert.

Den Schaltplan sieht man in **Bild 2**. Zentrales Bauteil ist der ATtiny85 [9], der mehr Flashspei-

cher als der ATtiny45 mitbringt. Alle sechs Portpins werden verwendet, teilweise mehrfach. An PB4 werden die Eindraht-Temperatursensoren angeschlossen (über den Steckverbinder K1). PB3 wird für den integrierten SMD-Quarzoszillator verwendet. Die SPI-Schnittstelle PB0..PB2 und der Resetpin PB5 realisieren zusammen das ISP-Interface, mit der sich der kleine AVR-Controller programmieren lässt. Im Betrieb wird über PB0 die Status-LED angesteuert.

Der Pegel an PB1 dient zum Umschalten des RS485-Transceivers zwischen „Senden“ (high) und „Empfangen“ (low). PB2 ist der oben erwähnte Pin, über den die UART-Signale versendet beziehungsweise empfangen werden. An K3 werden die RS485-Leitungen angeschlossen. Der Abschlusswiderstand von 120 Ω kann aufgelötet oder weggelassen werden.

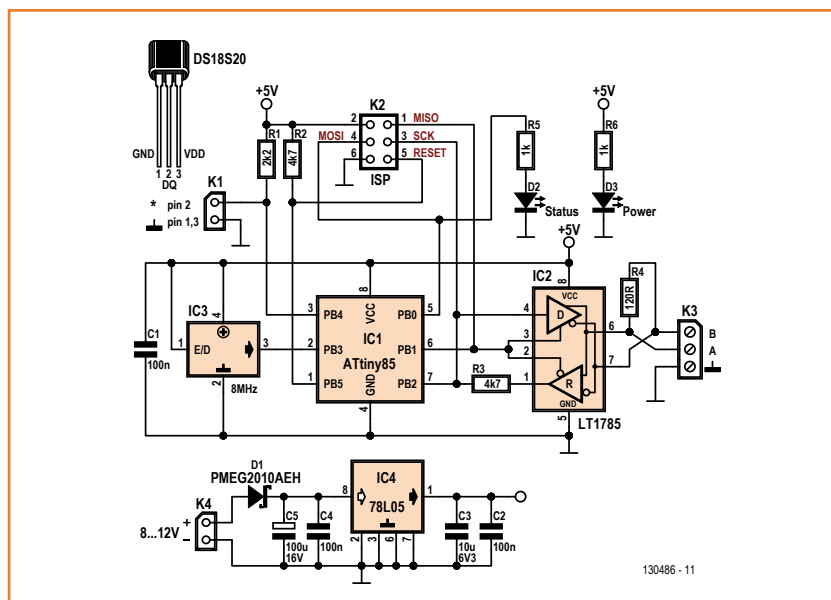
Konfiguration

Für den ATtiny85 bieten wir eine Standardfirmware zum Herunterladen an (natürlich wie immer kostenlos und open-source) [8]. Mauk van der Laan hat hier eine Software-UART-Library, eine 1-Wire-Library (eine angepasste Arduino-Bibliothek) und ein kleines ElektorBus-Interface integriert. Gesteuert wird das Ganze von einer Art Betriebssystem, das ein rudimentäres Multitasking beherrscht (siehe Kasten). Für fortgeschrittene Anwender lohnt es sich auf jeden Fall, einmal den kommentierten C++-Code zu studieren.

Bis zu vier T-Sensoren DS18S20 lassen sich an K1 parallel anschließen; dabei ist laut Datenblatt der VDD-Pin jeweils mit Masse zu verbinden, weil wir die Sensoren ja über die Datenleitung versorgen. Für die IDs der einzelnen Temperatursensoren benutzt die Software vier Speicherplätze (*Slots* 1..4) im EEPROM des ATtiny. Wenn man die Stromversorgung anschließt, ermittelt der Controller die IDs aller Sensoren am 1-Draht-Bus. Danach vergleicht er diese mit den bereits gespeicherten IDs. Wird eine bekannte ID nicht mehr gefunden, dann gibt die Software den entsprechenden Speicherplatz frei. Falls eine ID nicht bekannt ist, wird diese im ersten der noch freien Speicherplätze abgelegt.

Damit ist klar, wie man beim Konfigurieren des Systems vorgehen muss. Man schließt zuerst nur einen der 1-Wire-Sensoren an und fährt das Board durch Anlegen der Betriebsspannung hoch. Die ID wird nun im 1. Speicherplatz abgelegt. Am besten beschriftet man den Sensor mit einem Aufkleber (mit „0“ oder „1“, je nachdem, ob man

Bild 2.
Schaltplan rund um den
ATTiny85. Für den Software-
UART wird nur ein Pin
benötigt (PB2), da nie
gleichzeitig empfangen und
gesendet wird.



Basic- oder C-Programmierer ist ;-)). Danach klemmt man die Betriebsspannung ab, schließt den zweiten Sensor an K1 an und fährt erneut hoch. Nachdem die ID des zweiten Sensors ausgelesen und gespeichert wurde, fährt man mit dem dritten und vierten Sensor fort.

Der Fortschritt bei der Konfiguration wird recht hübsch durch die Status-LED visualisiert. Beim Start der Software blinkt die LED schnell, danach leuchtet sie vier Mal entweder kurz oder etwas länger auf. „Kurz“ steht für einen unbenutzten Speicherplatz, „Lang“ für eine bereits detektierte ID-Adresse. Hat man einen T-Sensor angeschlossen, bei dem man nicht mehr weiß, zu welchem Speicherplatz er gehört, kann man diesen abziehen, das Board hochfahren und die entsprechende Speicherstelle löschen lassen. Danach wiederholt man den oben beschriebenen Konfigurationsvorgang.

ElektorBus-Interface

Die vier T-Sensoren repräsentieren vier *Subnodes* (Unterknoten). Unsere Standardfirmware sendet

BIT	7	6	5	4	3	2	1	0	BYTE
	0	0	SET 1	0	1	C2	C1	C0	ADDRESS
	1	1	1	0	0	0	0	0	COMMAND
	0								FIRST
	0								SECOND

110610 - 13

in Intervallen die vier ermittelten Temperaturwerte innerhalb einer ElektorBus-Message aus. Hierfür werden die vier *Channels* der Nachricht genutzt; wir können auf jedem Kanal jeweils einen Wert von -1023...+1023 übermitteln [1]. Jeder Channel nimmt zwei Bytes im Nutzerdatenbereich der Nachricht ein (Bytes 6...13, von 0 an gezählt). Die Senderadresse (Bytes 4 und 5) ist bei unserem Knoten fest auf „5“ eingestellt. Die Bytes 2 und 3 enthalten die Empfängeradresse, diese ist bei unserer Standardfirmware

Bild 3.

Aufbau der Nachricht zum Setzen des Mess-Intervalls auf dem Temp-Sensor-Board (gezeigt sind nur die Bytes 6..9). *Interval Scale* ist 9, was der Schlüssel für Zehntelsekunden ist. Mit den Channelbits könnte man für jeden einzelnen Temp-Sensor ein eigenes Intervall setzen, hiervon wird in der Standardfirmware aber kein Gebrauch gemacht.

Anzeige

PROTOTYPES

Beta LAYOUT

3D-Druck für Elektronik

Gehäuse-Prototypen & Dummies



Sofort Online kalkulieren und bestellen

Wir stellen alle 3D-Modelle im Hi-Tech Lasersinterverfahren her.

Die Vorteile:

- Höhere Präzision
- Glattere Oberflächen
- Feinere Strukturen
- Komplexe Modelle sind realisierbar
- Flexibel bei Wandstärken von nur 0,4 mm - 2 mm

www.beta-prototypes.com

25 Jahre **Beta** LAYOUT
create: electronics

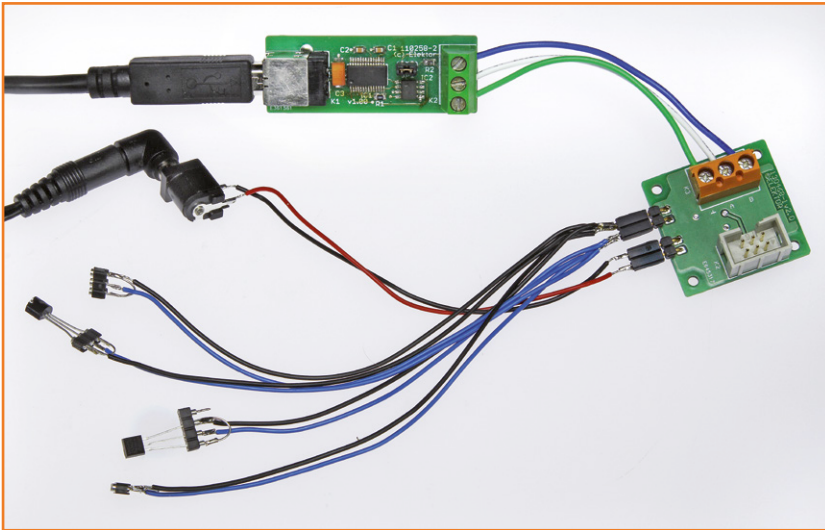
Bild 4.

Das Sensorboard sendet seine Messwerte über RS485 zu einem PC.

stets „10“. Nicht benutzt werden die Bytes 14 und 15 (Checksumme/CRC).

Die Standardfirmware ist für eine 1:1-Kommunikation eingerichtet, aber nicht für einen Betrieb mehrerer Sensorboards an einem Bus. Im sogenannten *DirectMode* werden Kollisionen einfach dadurch vermieden, dass wir ein definiertes Sen-

deschema haben. Der Sensorknoten sendet die Temperaturwerte in einem festgelegten Intervall an den *Master* (vorzugsweise eine Steuersoftware auf dem PC). Falls der Master ein Kommando zum Sensorknoten schicken muss, wartet er, bis eine Message mit Messwerten eintrifft, und versendet seine eigene Nachricht unmittelbar danach. Bei der normalen Busgeschwindigkeit von 9600 Baud sind Messintervalle bis hinunter zu 100 ms machbar; die Wandlungszeit der T-Sensoren liegt aber mit 750 ms (max.) weit darüber, daher dürften hier Intervalle von mehreren Sekunden sinnvoll sein. Die von Mauk erstellte Standardfirmware versendet die Temperaturwerte der identifizierten T-Sensoren in einem Default-Intervall von 1 s. Dabei werden die Temperaturwerte in Grad Celsius mit 10 multipliziert. Gleichzeitig horcht der Sensorknoten auf Kommandos zum Umstellen des Intervalls: 5 s und 500 ms sind einstellbar (siehe **Bild 3**). Außerdem lässt sich die Temperatur-Einheit von der Steuerung aus umschalten auf Grad Fahrenheit (und zurück auf Grad Celsius). Wenn der Sensorknoten eine solche Nachricht empfängt, übermittelt er nach dem nächsten Intervall in einer eigenen Nachricht den neuen Istzustand



Multitasking auf dem ATtiny

Die Standardfirmware wurde von Mauk van der Laan programmiert, einem selbstständigen Software- und Elektronik-Entwickler. Mauks Software ist modular aufgebaut, ein selbst entwickeltes Betriebssystem auf der Basis von Zustandsmaschinen sorgt für das korrekte Timing der einzelnen Tasks = Aufgaben (siehe unten).

Das File *Onewire.h* enthält die 1-Wire-Library. *OneWireTask.cpp/h* ist die Zustandsmaschine, die für die Detektion der IDs und das Auslesen der Temperaturwerte zuständig ist. Den Software-UART-Treiber *ElektorBusSw1w.cpp/h* für die RS485-Halbduplex-Kommunikation hat Mauk von einer Open-Source-RS232-Lib abgeleitet. Das ElektorBus-Interface *ElektorBus.cpp/h* kann aber auch mit einem Hardware-UART zusammenarbeiten. In *config.h* ist eingestellt, dass *ElektorBus.cpp* in der Standardfirmware auf die Software-UART-Schnittstelle zugreift:

```
#define ELEKTORBUS_DRIVER_INCLUDE "ElektorBusSw1w.h"
```

ElektorBusTask.cpp/h ist die Zustandsmaschine, die das Bus-Interface steuert.

Das Betriebssystem von Mauk van der Laan verwirklicht ein

sogenanntes kooperatives Multitasking. Im Gegensatz zum preemptiven Multitasking wird ein Task hier niemals von außen per Interrupt unterbrochen, sondern gibt die Kontrolle freiwillig wieder an einen Scheduler ab. Man benötigt also keine Sperr-Mechanismen, um ein Unterbrechen an einer ungeeigneten Stelle zu unterbinden. Im Gegensatz zu ausgewachsenen Embedded-Betriebssystemen wie FreeRTOS verfügen die Tasks hier auch über keinen eigenen Stack. Vielmehr ist jedem Task nur eine Zustandsvariable zugeordnet, der Speicherplatzbedarf ist also minimal. Daher läuft das Betriebssystem auch auf einem ATtiny mit 8 K Flash und 2 K RAM.

Die einzelnen Tasks werden mit Hilfe von Zustandsmaschinen (State Machines) implementiert. Der gesamte Code ist in einem Switch-Case-Ausdruck enthalten; je nach dem Wert einer Zustandsvariablen wird in den jeweiligen Codeabschnitt verzweigt. Am Ende ruft der Task die Methode *nextState()* auf und gibt die Kontrolle wieder an eine Zentralinstanz ab (die „Runtime“). Alternativ kann der Task auch *nextDelay()* aufrufen, mit einer Angabe einer Verzögerung in Millisekunden. Damit fragt der Task an, eine Zeitlang nicht mehr aufgerufen zu werden (Delay).

der Einheit und Skalierung (eine Nachricht mit Messwerten wird hierfür ausgelassen). Danach geht es mit Messwerten weiter. Da die Angaben in Grad Fahrenheit den Wert „102“ übersteigen könnten, konfiguriert der Sensor selbsttätig die Skalierung von -1 (Zehntelgrad) auf 0 (ganze Grad) um, wenn er eine Nachricht zum Umschalten auf Grad Fahrenheit erhält.

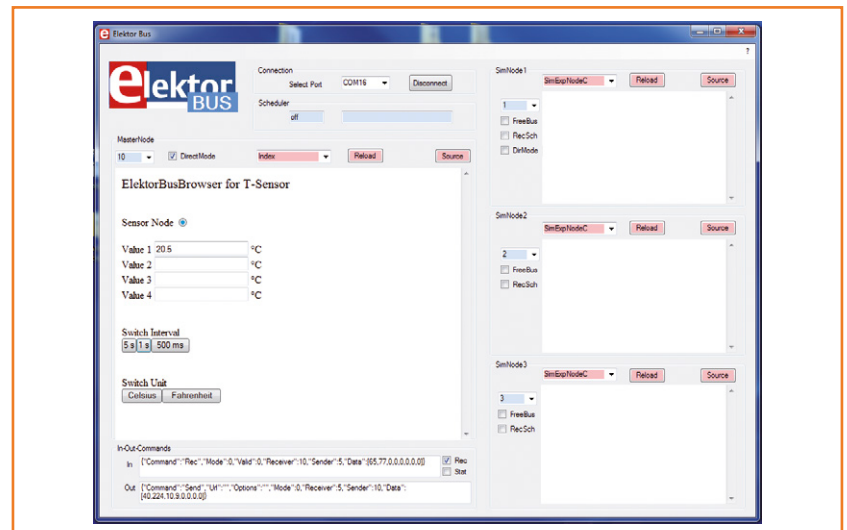
PC-Software

Ein Terminalprogramm eignet sich für einen ersten Test, zumindest sieht man, ob das Board im Sekundenintervall Messwerte versendet. Wir haben aber auch eine maßgeschneiderte Demo-Software für den PC entwickelt. Die Oberfläche ist wie immer in HTML/Javascript geschrieben. Im Download-File [8] findet man die Software ElektorBusBrowser.exe und den Ordner UIBus, den man auf den Desktop zieht.

Das Sensorboard verbinden wir über RS485 mit dem RS485/USB-Konverter von Elektor [3] und diesen wieder per USB mit dem PC (**Bild 4**). Nach dem Start des ElektorBusBrowsers stellt man oben die Nummer des COM-Ports ein, den der RS485/

USB-Konverter belegt, und drückt den Button „Connect“. Bei hochgefahrenem und konfiguriertem Sensorknoten müsste man jetzt schon die Messwerte angezeigt bekommen. Kommandos können wir aber erst zurückschicken, wenn wir einen Haken in der Checkbox „DirectMode“ setzen.

Bild 5.
Das PC-Userinterface basiert wie immer auf HTML. Hier lassen sich das Messintervall und die Einheit der Temperaturwerte ändern.



Der Scheduler in der Runtime verwaltet eine Liste von laufenden Tasks, die alle von der Klasse Task abgeleitet sind. Der Scheduler ruft reihum die execute()-Methode jedes Tasks auf (falls der Task kein Delay angefragt hatte, das noch nicht abgelaufen ist). Jeder Task läuft nur für eine kurze Zeit (ein paar Hundert Taktzyklen), so dass sich „gefühl“ ein echtes Multitasking ergibt.

Das Beispiel zeigt den Code eines Tasks, der eine LED blinken lässt:

```
class BlinkTask : public UserTask {

    enum States {Idle, BlinkingOn, BlinkingOff};

public:
    void start() {
        nextState(BlinkingOn);
    }
    void stop() {
        nextState(Idler);
    }
}
```

```
private:
    void turnOn();
    void turnOff();
    virtual void execute();
    virtual byte getTaskId() { return BLINKTASK_ID; }
};

void BlinkTask::execute() {
    switch(state) {
        case BlinkingOn: // turn the LED on and wait 500 ms
            turnOn();
            nextDelay(BlinkingOff, 500);
            return;
        case BlinkingOff: // Turn the LED off and wait 500 ms
            turnOff();
            nextDelay(BlinkingOn, 500);
            return;
        default: // Invalid state, give up
            panic(1);
    }
}
```

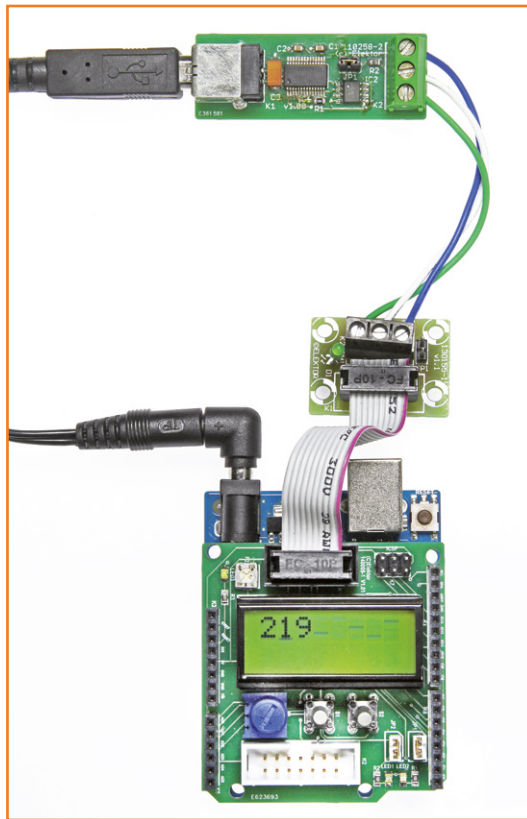


Bild 6.
Simulation mit einem
Arduino Uno und dem
Elektor Extension-Shield.
Am ElektorBus verhält sich
dieser Knoten genauso wie
das Temp-Sensor-Board.

Das simpel gehaltene User-Interface dürfte selbst-erklärend sein (**Bild 5**). Die Beschriftung „°C“ und „°F“ neben den Temperaturwerten wechselt erst, wenn eine Nachricht des Sensorknotens angekommen ist, dass die Einheit umgeschaltet wurde. Natürlich wird auch die neue Skalierung berücksichtigt. Den HTML/Javascript-Code kann man sich wie immer ansehen, wenn man den Knopf „Source“ betätigt.

Wer die Android-Bridge-Platine Andropod besitzt, kann das Temp-Sensor-Board auch mit einem Android-Smartphone oder -Tablet koppeln [10]. Die HTML/Javascript-Oberfläche funktioniert 1:1 auch im ElektorBusBrowserForAndropod, den man kostenlos bei Google Play herunterladen kann.

Simulation

Da beim Entwickeln der PC-Steuerung die Standardfirmware für das Board noch nicht zur Verfügung stand, wurde ein T-Sensorknoten auf einem Arduino-Uno-Board mit aufgesetztem Elektor Extension-Shield [11] und angeschlossenen RS485-Modul [12] simuliert (siehe **Bild 6**). Die „Temperaturwerte“ lassen sich mit dem Poti ändern, sie werden in Zehntelgrad Celsius auf dem Display angezeigt.

Der Arduino Uno sendet den simulierten Wert auf dem Channel 0 an den PC, natürlich im festgelegten Intervall. Auch auf die Steuerkommandos des PCs reagiert er genauso wie unser Temp-Sensorboard mit der Standardfirmware. Den Quellcode für den ATmega328 stellen wir ebenfalls im Download [8] zur Verfügung, er ist auf Basis der EFL [7] realisiert und nutzt die ElektorBus-Library. Im Ordner „Hardware“ des Atmel-Studio-Projekts befinden sich Codefiles für den Arduino Uno, das Elektor Extension-Shield und das RS485-ECC-Modul. Diese machen die darüber liegenden Software-Schichten hardwareunabhängig und sind selbst wiederum unabhängig voneinander. Weitere Software für das neue Shield gibt es in den nächsten Ausgaben.

Natürlich ist man nicht auf die Standardfirmware und den ElektorBus festgelegt: Mit eigener Software lassen sich auch selbst entwickelte Protokolle verwirklichen. Und man kann an K1 andere Sensoren, Schalter und sogar Aktoren anschließen; am Portpin PB4 lassen sich problemlos digitale Signale ausgeben und auch Spannungen messen (R1 entfernen). Falls Sie mit dem Board ein eigenes Projekt entwickelt haben, dann stellen Sie es doch auf unserer Plattform www.elektor-labs.com vor oder senden Sie eine Mail an die Redaktion!

(130468)

Weblinks

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

[2] www.elektor-magazine.de/110225

[3] www.elektor-magazine.de/110258

[4] www.elektor-magazine.de/110727

[5] <http://datasheets.maximintegrated.com/en/ds/DS18S20.pdf>

[6] www.elektor-magazine.de/120582

[7] www.elektor-magazine.de/120668

[8] www.elektor-magazine.de/130468

[9] www.atmel.com/images/atmel-2586-avr-8-bit-microcontroller-attiny25-attiny45-attiny85_datasheet.pdf

[10] www.elektor-magazine.de/110405

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

[12] www.elektor-magazine.de/130155

The new EAGLE has landed!

Version 7

ab sofort erhältlich

Mehr Informationen auf www.cadsoft.de



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



Einsteigerseminar: MSP430-Mikrocontroller

Am 18. Oktober bietet der Elektor-Verlag in Aachen ein Einsteigerseminar zur Mikrocontroller-Serie MSP430 von Texas Instruments an. Durch die geringe Leistungsaufnahme ist die MSP430er-Serie vor allem für alle Anwendungen im Bereich Personal Healthcare und Wearable Sensors bestens geeignet. Das Seminar richtet sich an Einsteiger der Mikrocontroller-Programmierung, die bereits Programmiererfahrung beispielsweise aus dem PC-Umfeld mitbringen, aber noch keine Erfahrung in hardware-naher Programmierung haben. Durch eine Reihe von Hands-On-Übungen am Mikrocontroller wird im Laufe des Seminars schrittweise ein Mikrocontroller-basierter optischer Herzratensensor entwickelt.

Veranstaltungsort/-termin: Aachen, 18. Oktober 2014

Referenten: Marcus Köny und Thomas Schlebusch

Teilnahmegebühr: 399,00 Euro (inkl. MwSt.)

3-tägiges Seminar: Embedded Linux in Theorie und Praxis – ein Crashkurs

Ziel des Kurses ist es, Ihnen grundlegende Embedded-Linux-Konzepte sowie die Handhabung von Linux zu vermitteln. Was sind z. B. Vor- und Nachteile? Sie werden Ihren eigenen Bootloader und Kernel cross-kompilieren, diverse Programme auf einem PC erstellen/cross-kompilieren und auf einem eingebetteten System ausführen und debuggen. Eine Kombination aus Theorie und praktischen Übungen wird es Ihnen ermöglichen, das neu erworbene Wissen bei Eigenentwicklungen einzusetzen. Nach dem Kurs sind Sie wahrscheinlich noch kein Embedded-Linux-Experte, aber hoffentlich in der Lage sein, sich selbstständig zurechtzufinden.

Veranstaltungsort/-termin: Hanau, 21. bis 23. Oktober 2014

Referent: Robert Berger

Teilnahmegebühr: 1.899,00 Euro (inkl. MwSt.)

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

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

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

Dimmer für LED-Taschenlampen

Von
Pascal Rondane
(F)

Die einstmals unscheinbaren LEDs haben sich zu lichtstarken und effizienten Leuchtmitteln gemausert. Doch die strahlende Helligkeit kann hinderlich sein, wenn eine leistungsstarke LED-Taschenlampe bei absoluter Dunkelheit ihre Blendwirkung entfaltet. Der Dimmer mindert diese Erscheinung durch Pulsbreitenmodulation des LED-Stroms. Außerdem kann die Taschenlampe auf Knopfdruck blinken.

Lassen Sie sich nicht blenden!

Mit dem LED-Dimmer lässt sich die Helligkeit leistungsstarker LED-Taschenlampen an die Beleuchtung der Umgebung anpassen, außerdem ist eine Blinksteuerung implementiert. Eine blinkende Taschenlampe kann die Aufmerksamkeit motorisierter Verkehrsteilnehmer auf den Fußgänger lenken, der sich auf dunkler Landstraße entlang des Fahrbahnrandes bewegt. Ein kleiner Mikrocontroller übernimmt hier zwei Aufgaben: Zum einen steuert er den Strom der

Taschenlampen-LEDs entweder nach dem Prinzip der Pulsbreitenmodulation (PWM) oder im Blinkmodus, zum anderen überwacht und signalisiert er die Restkapazität der Stromquelle.

Der Dimmer eignet sich nicht nur für den Einsatz in LED-Taschenlampen, sondern auch für andere LED-Beleuchtungssysteme mit vergleichbaren Spannungen und Strömen. Als Stromquellen sind Batterien, Akkus oder auch Netzteile mit Spannungen bis etwa 15 V verwendbar. In der vorgestellten Schaltung hat die Batterie die Nennspannung 7,2 V.

Eigenschaften

Betriebsspannung: 5 V ... 15 V
Strombedarf (ohne LED D2): 3,9 mA
Ausgangsstrom: 1 A
Akku-Schwellenspannungen (änderbar):
- Akku geladen: 7 V
- Akku fast leer: 6 V
- Akku leer (Lampe aus): 5,4 V
Platinenabmessungen: 60 · 30 mm

Mikrocontroller

Die Dimmerschaltung, dargestellt in **Bild 1**, arbeitet mit einem Mikrocontroller ATtiny45 (IC1), der nur acht Anschlüsse besitzt und in der Schaltung (*in circuit*) programmiert wird. Rechts im Bild befindet sich durchbrochen umrahmt die Schaltung der LED-Taschenlampe. Die Lichtstärke der LEDs wird mit S2 eingestellt, ein Umschalttaster, der selbsttätig in die neutrale Mittelstellung zurück-

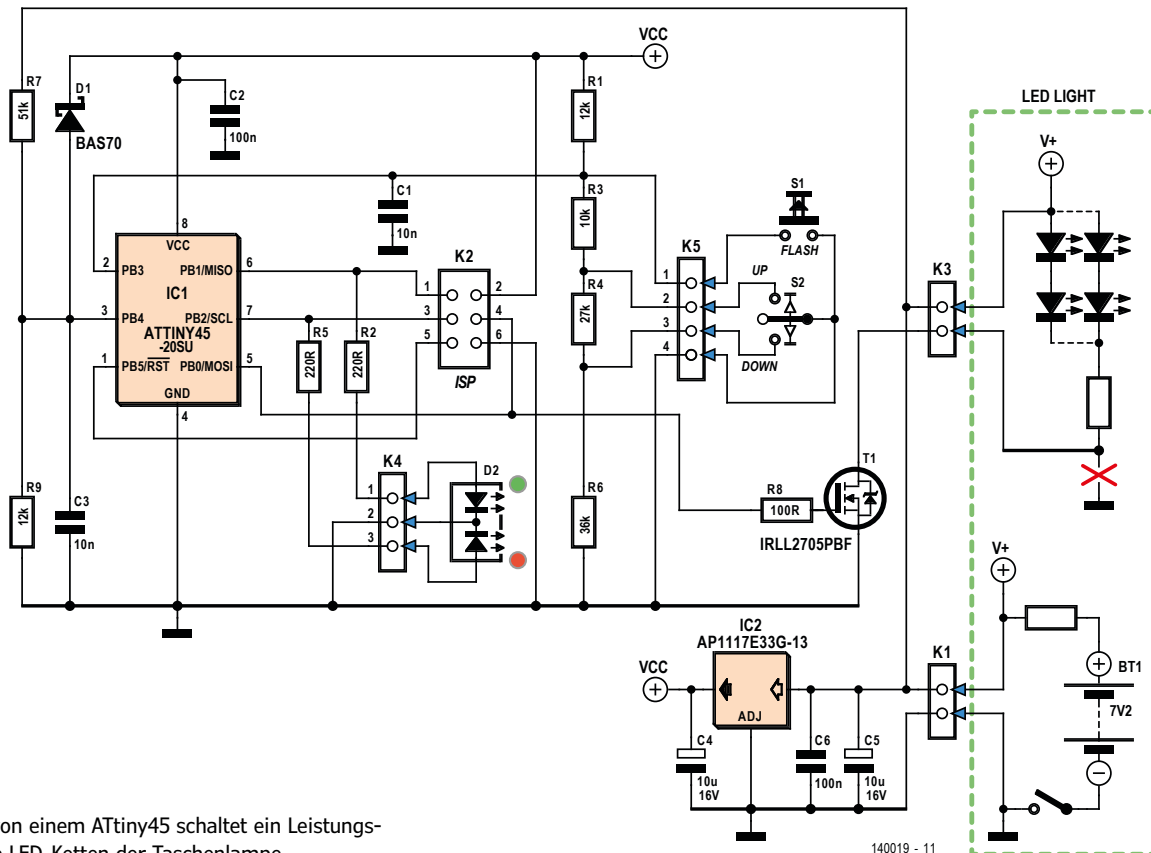


Bild 1.

Gesteuert von einem ATtiny45 schaltet ein Leistungs-MOSFET die LED-Ketten der Taschenlampe.

kehrt. Beim Druck in eine Richtung sinkt die Helligkeit, beim Druck in die andere Richtung nimmt die Helligkeit zu. Wenn Schalter S1 geschlossen wird, beginnt die Taschenlampe zu blinken.

Da die Anzahl der Anschlüsse beim Mikrocontroller ATtiny45 begrenzt ist, werden ihm die Schaltzustände der Taster als analoge Signale übergeben. Die Widerstände R1, R3, R4 und R6 bilden einen Spannungsteiler, so dass am Verbindungspunkt R1/R3 eine Spannung liegt, die von den Schalterstellungen abhängt. Die Spannung wird Eingang PB3 (Pin 2) des Mikrocontrollers zugeführt. Das Mikrocontroller-Programm wertet die Höhe der Spannung wie folgt aus:

- 2,8 V: S1 und S2 offen, keine Aktion
- 2,5 V: S2 UP, Helligkeit steigern
- 1,5 V: S2 DOWN, Helligkeit mindern
- 0 V: S1 geschlossen, Blinken

Die Lichtstärke der LEDs wird durch das Tastverhältnis eines pulsformigen Signals gesteuert (PWM-Signal), das vom Programm generiert wird. Dieses Signal stellt der Mikrocontroller an Anschluss

PB0 (Pin 5) bereit. Das PWM-Signal steuert Leistungs-MOSFET T1, er schaltet die LED-Ströme. Beim Start leuchten die LEDs mit halber Lichtstärke, so dass die Blendwirkung bereits reduziert ist. An Eingang PB4 (Pin 3) des Mikrocontrollers liegt über Spannungsteiler R7/R9 eine Spannung, die ein Maß für die Restkapazität der Stromquelle ist. Das Programm bewertet diese Spannung über den internen AD-Wandler, das Ergebnis wird von einer Mehrfarben-LED (D2) signalisiert. Die LED-Farben haben folgende Bedeutungen:

- **Grün** = Gut bis ausreichend,
- **Orange** = Nahezu leer,
- **Rot** = Leer.

Wenn die Kapazität erschöpft ist, schaltet der Mikrocontroller die LEDs der Taschenlampe vollständig ab. Wiederaufladbare Akkus werden durch diese Funktion vor Tiefentladung geschützt. Sobald der Akku geladen ist, wird der normale Betriebszustand selbsttätig wieder hergestellt. Bei der Batterie- oder Akkuspannung 8 V liegen an Eingang PB4 (Pin 3) des ATtiny45 1,5 V,



Bild 2.

Auf der Platine lassen sich die SMDs von Hand montieren, die Platine passt in die meisten gebräuchlichen Stab-Taschenlampen.

Stückliste

Widerstände (SMD1206, 1 %, 1/4 W):

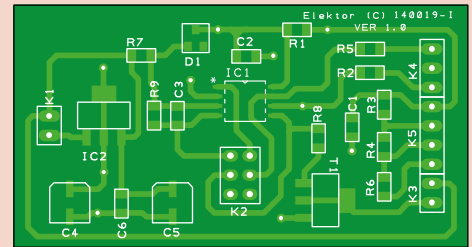
R1,R9 = 12 k
R2,R5 = 220 Ω
R3 = 10 k
R4 = 27 k
R6 = 36 k
R7 = 51 k
R8 = 100 Ω

Kondensatoren (SMD1206):

C1 = 10 n*
C2,C6 = 100 n
C3 = 10 n
C4,C5 = 10 μ , 16 V rad.

Halbleiter:

IC1 = ATtiny45-20SU (SOIC8), programmiert
erhältlich (140019-41)
IC2 = AP1117E33G-13 (SOT-223)
D1 = BAS70* (SOT-23)
T1 = IRL2705PBF (SOT-223)



Außerdem:

K1,K3 = Stiftkontaktleiste 2-polig, gewinkelt
K2 = Stiftkontaktleiste 2 x 3-polig
K4 = Stiftkontaktleiste 3-polig, gewinkelt
K5 = Stiftkontaktleiste 4-polig, gewinkelt
S1 = Taster (Öffner)
S2 = Tast-Umschalter On-Off-On
Platine 140019-1

* siehe Text

```
'#####  
'# Battery voltage test (Voltage hysteresis = 0V)  
'#####  
  
Select Case Etat_led 'Voltage hysteresis = 0V  
  
Case 1 : 'Operating voltage => OK  
  
Tension_batt = Tension_batt_basse - Tension_hysteresis '6V-0V = 6.0V  
If Adcval < Tension_batt Then 'ADC measurement and switching to low battery mode  
Etat_led = 2  
End If  
  
Case 2 : 'Low battery  
  
Tension_batt = Tension_batt_dechargee - Tension_hysteresis '5,5V-0V = 5.5V  
If Adcval < Tension_batt Then 'ADC measurement and go to discharged battery  
' mode - cut the load  
Etat_led = 3  
End If  
Tension_batt = Tension_batt_chargee + Tension_hysteresis  
If Adcval >= Tension_batt Then  
Etat_led = 1  
End If  
  
Case 3 : 'If low battery/ cutoff LED  
Tension_batt = Tension_batt_chargee + Tension_hysteresis '7,2V + 0V = 7.2V  
If Adcval >= Tension_batt Then 'ADC measurement and return  
' "OK mode "if battery is charged  
Etat_led = 1  
End If  
  
End Select
```

Dieser Programmteil überwacht die Betriebsspannung.

LED D2 hat die Farbe Grün. Die Spannung 6 V wird von R7/R9 auf 1,16 V herabgeteilt, LED D2 wechselt nach Orange. Sinkt die Spannung unter 5,4 V, bleibt am Spannungsteiler R7/R9 nur noch 1 V übrig. LED D2 signalisiert dies durch die Farbe Rot, die LEDs der Taschenlampe verlöschen. Da bei mobilen Verbrauchern der sparsame Umgang mit der Energie Priorität hat, blinkt LED D2 in den genannten Farben, sie leuchtet nicht kontinuierlich.

Die Schottky-Diode D1, beispielsweise eine BAS70, schützt den Eingang PB4 des Mikrocontrollers vor eventuellen Spannungsspitzen. Dem Prellen des Umschalttasters S2 wirkt Kondensator C1 entgegen. Falls der Wert 10 nF bei einem stark prellenden Schalter nicht ausreicht, kann der Wert auf 100 nF erhöht werden. Das Prellen macht sich dadurch bemerkbar, dass sich die Lichtstärke beim Betätigen von S2 sprunghaft ändert.

IC2, ein Spannungsregler vom Typ AP1117, liefert die Betriebsspannung 3,3 V für den Mikrocontroller ATtiny45. Da die Nennspannung der LEDs 7,2 V beträgt, übernimmt Leistungs-MOSFET T1 neben seiner Funktion als Schalter auch das Anpassen der unterschiedlichen Betriebsspannungen.

Software

Das Programm für den ATtiny45 wurde in Bascom-AVR geschrieben, es steht auf der Projektseite [1] zum Download bereit. Eventuelle Modifikationen sind auch mit der Demo-Version von Bascom möglich.

Zuerst werden die Variablen für die Spannungen deklariert, die den Schaltern zugeordnet sind, gefolgt von den Schwellenwerten der Betriebsspannungsüberwachung. Anschließend werden Timer 0 für die PWM und Timer 1 für das Blinken der LED D2 geladen. In der Hauptschleife verbringt der Mikrocontroller die meiste Zeit mit dem Abfragen der Schalterzustände. Ferner stehen im Programm die Routine, welche die Mehrfarben-LED D2 abhängig vom Zustand der Stromquelle steuert, sowie die Routine für den AD-Wandler, mit dem die Schalterpositionen ermittelt werden.

Im Blinkmodus, wenn Schalter S1 geschlossen ist, hat das PWM-Signal maximale Pulsbreite, so dass die Taschenlampe mit voller Stärke blinkt. Der Blinkmodus wird verzögert aktiviert, damit ein versehentliches kurzes Betätigen von S1 wirkungslos bleibt.

Der Quellcode des Programms ist umfassend kommentiert, Modifikationen sind unkompliziert möglich. Beispielsweise ist die Dimmgeschwindigkeit steigerbar, indem die Variable `Pas` herabgesetzt wird, und mit niedrigeren Werten für `Tempo_bp_on_value` reagiert die Taschenlampe schneller auf das Drücken der Taster. Eine andere Modifikation betrifft den Einschaltwert der Helligkeit. Bei `Pwm = 100` startet die Taschenlampe mit maximaler Lichtstärke.

Auf- und Einbau

Für den LED-Taschenlampendimmer wurde eine Platine entworfen, das Layout ist in **Bild 2** wiedergegeben. Die Platine ist mit SMDs bestückt, da sie aufgebaut zusätzlich in den zylinderförmigen Teil des Taschenlampengehäuses passen muss. Der Einbau gestaltet sich nicht immer so komplikationslos wie erhofft, denn manche Taschenlampengehäuse sind geklebt oder verschweißt. Manchmal ist das Öffnen nicht möglich, ohne das Gehäuse leicht zu beschädigen.

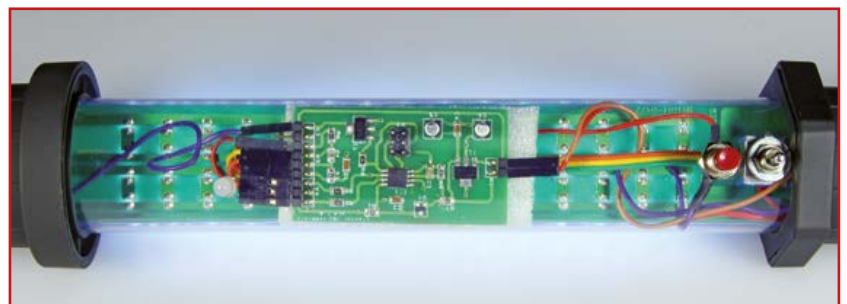
Nachdem diese Hürde genommen ist, lässt sich die Platine meistens bequem im Gehäuse unterbringen. Die Masseleitung der LED-Bänke (meistens 2 · 30 LEDs) wird aufgetrennt, angedeutet durch das rote „X“ im Schaltbild. Zwischen dem Fußpunkt der LED-Bänke und K3 muss eine Leitung verlegt werden, die dann mit MOSFET T1 verbunden ist. Der andere Anschluss führt von K3 zum gemeinsamen Plusanschluss der LED-Bänke und der Stromquelle. Dann folgen die Verbindungen von Plus und Minus des Akkus nach K1, und schließlich sind noch die Taster S1 und S2 mit K5 zu verdrahten. Für die Montage der Taster werden dazu passende Durchbrüche in das Lampengehäuse gebohrt.

(140019)gd

Weblinks

[1] www.elektor-magazine.de/140019

Bild 3. Ideal ist eine Taschenlampe, die sich unkompliziert öffnen lässt und genügend freien Raum für den Platineneinbau bietet.



Nixie-Uhr mit GPS

Genaue Zeit aus dem Weltraum

Von **Willem Tak** (NL)



Das sanfte Glimmen nostalgischer Ziffernanzeigeröhren gibt dieser Uhr einen extravaganteren, nicht alltäglichen Touch. Doch das ist es nicht allein: Hier sind Klassik und Moderne gemeinsam stilvoll in Szene gesetzt. Die Nixie-Röhren sind ein Erbe aus der Pionierzeit der Digitaltechnik, die GPS-Satelliten, die um die Erde kreisen, sind eine Errungenschaft der Gegenwart.

Nixie-Röhren ziehen fast jeden Elektroniker in ihren Bann. Die warmroten Ziffern, die sich, so scheint es, im Glaskolben vor und zurück bewegen, sind einzigartig und unverwechselbar. In diesem Beitrag stellen wir eine Digitaluhr mit vier Nixie-Röhren vor, sie zeigen Stunden und Minuten an. Die präzise Zeit bezieht unsere Nixie-Uhr nicht vom Normalzeitsender DCF77, sondern von dem am Himmel installierten GPS-System. Für den Empfang der Satellitensignale hatte der Autor das Modul EM-406 (oder EM-411) von GlobalSat vorgesehen. Wegen der leichteren Beschaffbarkeit wurde es gegen das Modul

A2035H von Maestro getauscht, das bereits auf dem FPGA-Erweiterungsboard zum Einsatz kam. Die Empfangsmodule von GlobalSat sind ohne Firmware-Änderungen einsatzfähig, sie erfordern jedoch Modifikationen der Platine. Alle genannten Typen sind so leistungstark, dass sie auch in Gebäuden und unter schwierigen Empfangsbedingungen ihren Einsatzzweck erfüllen.

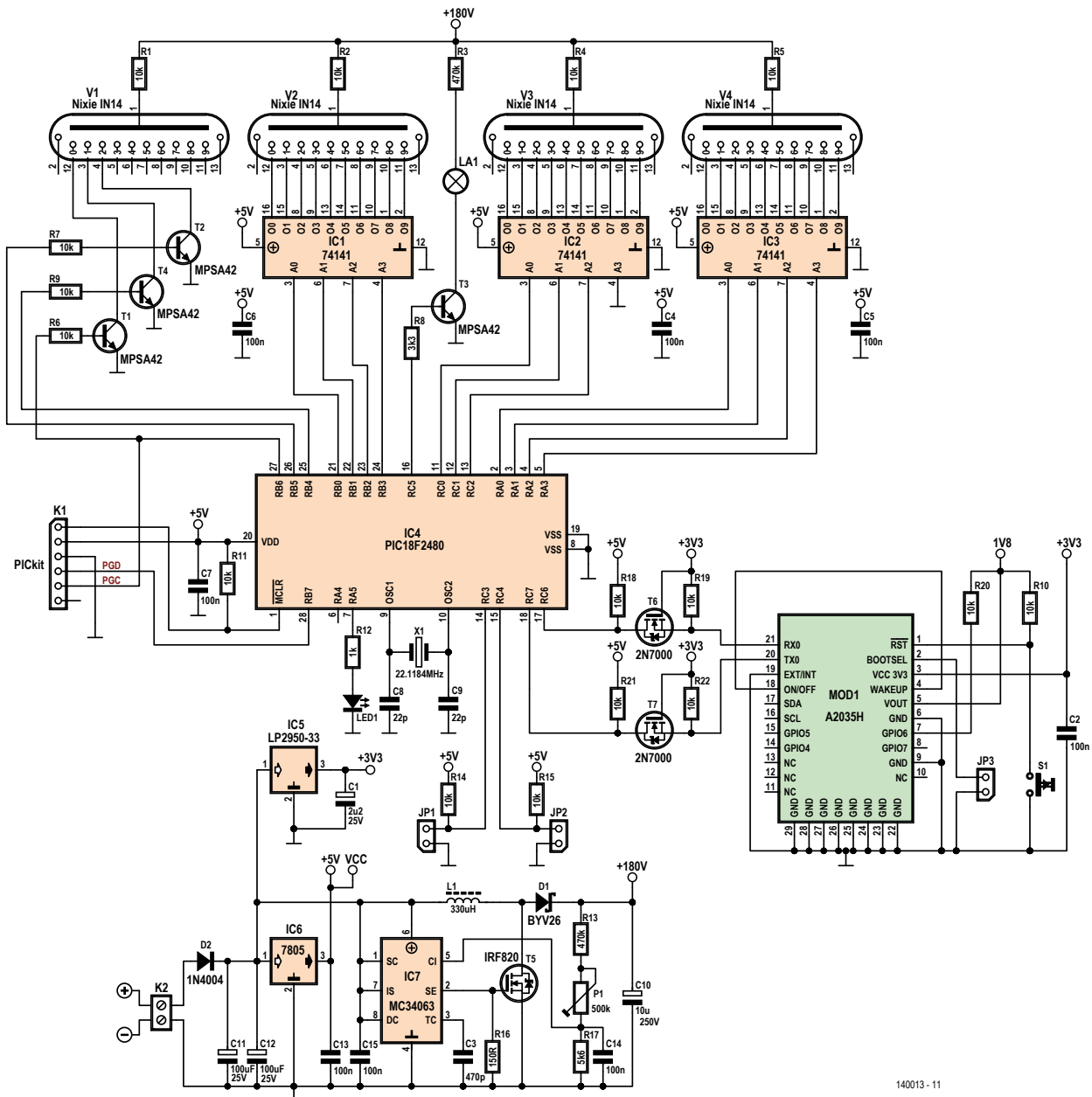
Die Schaltung

Blickfang dieser Uhr sind die vier Nixie-Ziffernanzeigeröhren, sie sind in **Bild 1** mit V1...V4 bezeichnet. Nixie-Röhren waren bereits die Glanz-

stücke anderer Elektor-Projekte (siehe [1] und [2]), so dass an dieser Stelle eine Kurzbeschreibung genügen soll. Eine Nixie-Röhre ist eine so genannte Kaltkathodenröhre, sie arbeitet nach dem Prinzip der Glimmlampe. Für jede anzuzeigende Ziffer ist im Glaskolben eine separate Kathode mit den Konturen der Ziffer eingebaut. Beim Anlegen einer Spannung in der Größenordnung 100...250 V umgibt eine glimmende Leuchtschicht die Kathode. Die hohe Spannung wird hier

von einem Aufwärts-Spannungswandler MC34063 (IC7) aus der Betriebsspannung 12 V erzeugt. Gemeinsam mit einer schnellen Diode (D1) und einer Induktivität (L1) lädt der Spannungswandler einen Hochvolt-Elko (C10) auf etwa 180 V auf. Ein Teil der Spannung wird zum Anschluss 5 des Spannungswandlers IC7 rückgeführt, so dass der Regelkreis geschlossen ist. Die Rückführung darf unter keinen Umständen unterbrochen werden, anderenfalls kann die Spannung auf Werte

Bild 1.
Ein Mikrocontroller aus der PIC-Familie bereitet die GPS-Zeit auf und steuert über Treiber vier Nixie-Röhren.



140013 - 11

hochlaufen, die über der Spannungsfestigkeit des Elkos liegen. Vom Elko gelangt die Spannung 180 V über die Widerstände R1, R2, R4 und R5 zu den Anoden der Nixie-Röhren. Die optimalen Anodenströme hängen vom Röhren-Typ ab, sie sind durch Anpassen der Widerstände variierbar. In der Uhr, die versuchsweise im Elektor-Labor aufgebaut wurde, waren Nixie-Röhren russischer Herkunft des Typs IN-16 verbaut. Diese russischen Nixie-Röhren sind gegenwärtig im Fachhandel und auf Internet-Plattformen preiswert erhältlich.

Zum Steuern von Nixie-Ziffernanzeigeröhren war seinerzeit der Baustein 74141 der TTL-Standard-

22,1184 MHz, passend zur Ausgabegeschwindigkeit 4800 Baud des GPS-Moduls (MOD1). Der PIC steuert ferner eine optionale LED (LED1), die nach dem Reset kurzzeitig blinkt und anschließend jeden korrekt empfangenen GPS-String mit einem kurzem Aufleuchten begleitet. Falls ein Empfangsproblem auftritt, leuchtet die LED kontinuierlich. Die In-Circuit-Programmierung des Mikrocontrollers ist über die Schnittstelle K1 möglich, geeignet ist beispielsweise der PIC-Programmer „PICKit 2“.

Von Jumper JP1 hängt ab, ob die Nixie-Röhren eine führende Null darstellen (entweder „00.00“ oder „0.00“), während JP2 festlegt, ob der Dezimalpunkt blinkt oder kontinuierlich leuchtet.



reihe auf dem Markt. Zwar wurde dieses IC inzwischen zur seltenen Rarität, doch mit dem russischen Ersatztyp K155ID1 bietet sich eine Alternative an. Zum Steuern der Röhre V1 (Stunden-Zehner) ist kein IC nötig, die Ziffern „0“, „1“ und „2“ werden von den Transistoren T1, T2 und T4 gesteuert. Zwischen den Ziffernpaaren der Stunden und Minuten hat das von T3 gesteuerte Neonlämpchen LA1 seinen Platz. Im Uhrbetrieb kann das Lämpchen (wählbar mit JP2) einmal in der Sekunde blinken oder kontinuierlich leuchten.

Die vom GPS-Modul kommenden Zeitinformationen werden von einem Mikrocontroller PIC18F2480 (IC4) ausgewertet und in Steuersignale für die Nixie-Röhren umgesetzt. Der Mikrocontroller-Quarz (X1) schwingt mit der Frequenz

Im GPS-Modul A2035H sind sämtliche Funktionen einschließlich der GPS-Antenne integriert, die zum Empfang der Satellitensignale nötig sind. Die externen Komponenten beschränken sich auf zwei Widerstände und einen Kondensator. Taster S1 und Jumper JP3 sind nur erforderlich, wenn die Firmware des GPS-Moduls A2035H einem Update unterzogen werden soll. Da die Betriebsspannung des GPS-Moduls 3,3 V beträgt, der PIC18F2480 jedoch an 5 V arbeitet, müssen die Pegel der Datensignale konvertiert werden. Diese Aufgabe übernehmen die FETs T6 und T7 zusammen mit den Widerständen R18/R19 und R21/R22.

Spannungsregler IC6, ein 7805, leitet aus der an K2 angelegten Versorgungsspannung die Betriebsspannung 5 V ab, während IC5, ein LP2950-33, das GPS-Modul mit 3,3 V versorgt. Als Stromversorgung ist ein Steckernetzteil geeignet, das eine Gleichspannung im Bereich 9...15 V abgibt.

Die PIC-Software

Wie bei den meisten Projekten stehen der Quell- und Hexcode auf der Projektseite im Web zum kostenlosen Download bereit, und auf Wunsch liefert der Elektor-Shop einen bereits programmierten Mikrocontroller ins Haus (siehe [3]). Das Programm ist so konzipiert, dass es auch Uhren mit Sekundenanzeige (sechs Ziffern) steuern kann. Bei unserer Nixie-Uhr sind zwar nur vier Ziffern vorgesehen, doch auf der Seite der Software steht einer Erweiterung nichts im Weg. An den Mikrocontroller übergibt das GPS-Modul einen String, der mit GPRMC bezeichnet ist. Bei der Initialisierung wird das GPS-Modul so konfiguriert, dass nur dieser String einmal in der Sekunde zum Mikrocontroller gelangt. Die ande-

ren Strings haben hier keine Bedeutung, sie sind deaktiviert. Eine Routine prüft zyklisch, ob im Puffer des seriellen Eingangs ein Zeichen steht. Wenn nach dem Start der Datenstrom ausbleibt, führt der Watchdog des Mikrocontrollers nach etwa einer Minute einen Reset herbei. Während dieses Betriebszustands leuchtet die LED1 kontinuierlich. Auch nachdem ein gültiger String empfangen wurde, wird die Abfrage des Eingangspuffers fortgesetzt. Ein kontinuierliches Leuchten von LED1 weist auch jetzt auf ein Problem hin. Werden Daten empfangen, prüft eine Routine, ob es sich um einen GPRMC-String handelt, was stets zutreffen muss. Die empfangenen Zeichen werden bis zum Empfang des Zeichens CR in einen Puffer geschoben. Dann wird die Prüfsumme des String berechnet und mit der im String enthaltenen Prüfsumme verglichen. Wenn der berechnete Wert mit dem empfangenen Wert übereinstimmt, gelten die Daten als gültig.

Anschließend ermittelt eine Routine, welcher Zeitversatz zur empfangenen Zeit addiert werden muss, um die korrekte Sommer- oder Winterzeit (MEZ oder MESZ) zu erhalten. Das GPS-System setzt hierzu ein Flag. Der Algorithmus berechnet die Anzahl der Stunden seit Jahresanfang und entnimmt einer Tabelle, ob für die laufende Stunde die Sommer- oder Winterzeit gilt. Die Tabelle enthält Daten bis Ende 2020.

Der letzte Schritt ist das Umsetzen der vom GPS-Modul im ASCII-Format kommenden Informationen in separate Variablen für Stunden, Minuten und Sekunden. Nebenbei bemerkt: Auch eine ASCII/BCD-Konvertierung ist implementiert, sie dient aber lediglich der Kompatibilität mit anderen Uhren des Autors.

Bei regulärem GPS-Empfang steht jetzt die Zeitinformation für die Anzeige bereit. Der Empfang der GPS-Strings ist jedoch nicht immer verlässlich, er hängt von den Empfangsbedingungen ab. Zwar ist das GPS-Modul ein ausgefeiltes, hochempfindliches System, trotzdem kann gelegentlich ein String fehlen. Deshalb muss noch ein potentieller Anzeigefehler beseitigt werden, der auffällt, falls die Zeit sechsstellig (einschließlich Sekunden) angezeigt wird. Bei vierstelliger Anzeige (nur Stunden und Minuten) bleibt dieser Fehler meistens im Verborgenen. Wenn ein String fehlt, bleiben die Sekunden für die Dauer dieses String stehen, um anschließend um zwei Sekunden zu springen. Aus diesem Grund erscheint

nicht unmittelbar die GPS-Zeit auf dem Display, sondern die Zeit einer zusätzlich implementierten Echtzeituhr (Real Time Clock, RTC). Die RTC ist mit einem Timer realisiert, der im Abstand von 50 ms Interrupts auslöst, sie wird von der GPS-Zeit synchronisiert. Im Programm kommen Variable wie PIC_HR und GPS_HR, PIC_MIN und GPS_MIN sowie PIC_SEC und GPS_SEC vor, was auf die unterschiedlichen Variablen der RTC und des GPS hindeutet. Das 50 ms breite, pulsformige Signal der RTC kann an Pin 6 (RA4) des Mikrocontrollers abgegriffen werden, ein Oszilloskop kann das Signal sichtbar machen.

Die RTC startet mit „00.00“ (oder „00.00.00“), sie kann nicht gestellt werden. Für die Synchronisa-



tion mit der aktuellen Zeit ist der Empfang mindestens eines korrekten GPS-String notwendig. Um eine zuverlässige Zeitanzeige zu gewährleisten, wird jeder eintreffende GPS-String einer umfassenden Prüfung unterzogen. Für gültig befundene Strings werden als Zeiten in einen Puffer geschoben, der insgesamt zehn Zeiten speichert. Zu jeder Zeit gehören drei Bytes, sie stellen die Anzahl der Sekunden dar, die seit Mitternacht vergangen sind. Der Puffer verhält sich wie ein Schieberegister: Eine empfangene Zeit schiebt alle Zeiten um einen Platz vorwärts, so dass die älteste Zeit aus dem Puffer fällt. Wenn der vom GPS-Modul kommende Datenstrom kontinuierlich und störungsfrei verläuft, beträgt die Differenz zwischen der ältesten und neuesten Zeit immer neun Sekunden. Ist dieses Kriterium

erfüllt, wird ein Flag gesetzt, das GPS_STABLE heißt. Das Flag gibt dann die Synchronisation der RTC-Zeit mit der GPS-Zeit frei.

Die Synchronisation wird 30 Sekunden nach jeder vollen Stunde durchgeführt. Hat die RTC-Zeit diese Zeit erreicht, generiert die Interrupt-Routine einen Auftrag zur Synchronisation. Der genannte Zeitpunkt wurde gewählt, damit sich eventuelle Korrekturen in der Anzeige nur auf die Sekunden, nicht jedoch auf die Minuten und Stunden auswirken.

Trotz der beschriebenen Mechanismen arbeitet die Synchronisation noch nicht perfekt. Während der

Startphase tritt gelegentlich der Fehler auf, dass das GPS-Modul Zeiten liefert, die von der korrekten Zeit drastisch abweichen, jedoch trotzdem das Kriterium der Gültigkeit erfüllen. Die RTC-Zeit wird falsch synchronisiert, und es kann bis zu einer Stunde dauern, bis der Fehler korrigiert ist. Um dies zu verhindern und gleichzeitig ohne ungenaue Interrupt-Schleifen auszukommen, wird auch die RTC-Zeit in ein aus drei Bytes bestehendes Format gewandelt. Solange das GPS-Signal stabil ist, werden die Sekunden der GPS-Zeit und die Sekunden der RTC-Zeit voneinander subtrahiert. Beträgt die Differenz weniger als drei Sekunden, wird die Synchronisation frei gegeben.

Stückliste

Widerstände:

R1,R2,R4,R5,R6,R7,R9,
R10,R11,R14,R15,R18,
R19,R20,R21,R22 = 10 k
R3,R13 = 470 k
R8 = 3k3
R12 = 1 k
R16 = 150 Ω
R17 = 5k6
P1 = 500 k Trimpoti
legend

Kondensatoren:

C1 = 2 μ 2/25 V,
Raster 2 mm
C2,C4,C5,C6,C7,C13,C14,
C15 = 100 n
C3 = 470 p, Y5P,
Raster 2,5 mm
C8,C9 = 22 p, C0G/NP0,
Raster 2,5 mm
C10 = 10 μ /250 V,
stehend, Raster
5 mm (Panasonic
ECA2EHG100)
C11,C12 = 100 μ /25 V,
stehend, Raster 3,5 mm

Induktivität:

L1 = 330 μ H/900 mA,
stehend ($\varnothing$ 10 mm, Höhe 15 mm)

Halbleiter:

D1 = BYV26 (Ultrafast Diode, 600 V/1 A)
D2 = 1N4007
LED1 = LED 3 mm, rot
T1,T2,T3,T4 = MPSA42 (300 V/500 mA)
T5 = IRF820 (N-MOSFET, 500 V/2,5 A)
T6,T7 = 2N7000 (N-MOSFET, 60 V/200 mA)
IC1,IC2,IC3 = K155ID1 (74141)
IC4 = PIC18F2480-I/SP (programmiert,
EPS 140013-41)
IC5 = LP2950-33LPE3
IC6 = MC7805
IC7 = MC34063

Außerdem:

V1,V2,V3,V4 = IN-14 (Nixie-Ziffernanzeigeröhre)
LA1 = Glühlampe mit Anschlussdrähten
X1 = Quarz 22,1184 MHz
MOD1 = A2035H, GPS-Modul mit integrierter Antenne
(Maestro Wireless Solutions, Farnell 2281693)
JP1,JP2,JP3 = Stiftkontaktleiste 2-polig mit Jumper
(JP3 optional)
K1 = Stiftkontaktleiste 6-polig, Raster 2,54 mm
K2 = Kabelschraubklemme 2-polig, Raster 5,08 mm
S1 = Drucktaster für Platinenmontage, 6 $\times$ 6 mm
(optional)
Platine 140013-1, siehe [1]

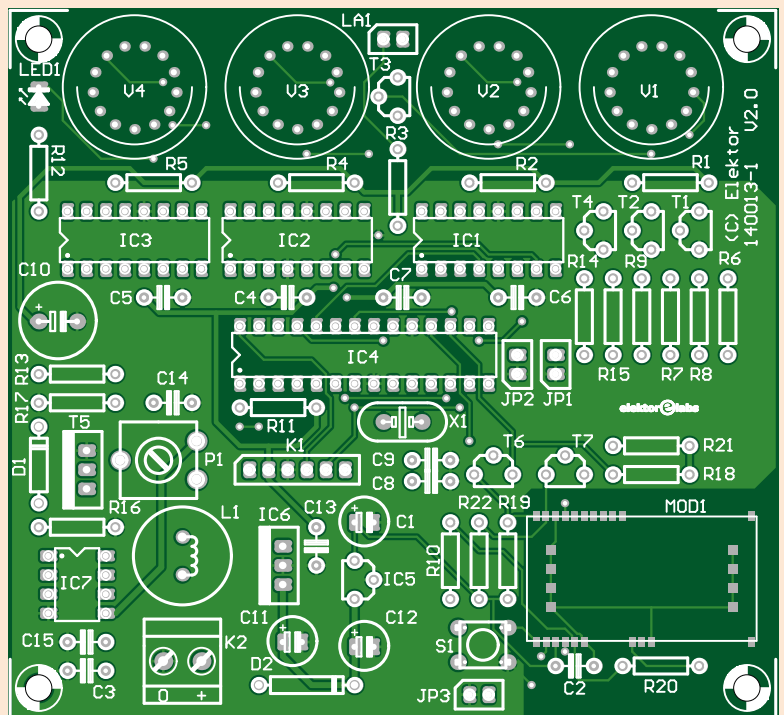


Bild 2.
Auf dieser Platine hat die Nixie-Uhr einschließlich GPS-Modul ihren Platz. Was noch fehlt, ist ein Steckernetzteil.

Diese Methode wirkt zwar improvisiert, trotzdem hat die Praxis gezeigt, dass die Nixie-Uhr auch im Langzeitbetrieb zuverlässig arbeitet. Das Neonlämpchen zwischen den Nixie-Röhren der Stunden und Minuten ist ein Indiz für die Ganggenauigkeit. Wenn das Lämpchen blinkt oder kontinuierlich leuchtet, ist die Zeit synchronisiert. Falls das Lämpchen verloschen ist, wird die Uhr von Interrupts des Mikrocontrollers gesteuert. Die Ursache ist das Ausbleiben eines gültigen GPS-Strings über eine lange Periode.

Aufbau

Die Platine für die Nixie-Uhr ist in **Bild 2** wiedergegeben, sie kann über die Projektseite [3] im Elektor-Shop bestellt werden. Das Layout steht auf der Projektseite zum kostenfreien Download bereit. Bis auf das GPS-Modul sind nur bedrahtete Komponenten vorhanden, die Montage dieser Bauelemente ist nicht schwierig. Das GPS-Modul ist mit kleinen Lötkontakflächen ausgestattet, die mit den zugehörigen Kupferinseln der Platine verlötet werden müssen. Mit einem feinen LötKolben, einer ruhigen Hand und etwas Geduld ist auch dies schnell machbar. Auf der Unterseite des GPS-Moduls befinden sich diverse Masseflächen, die nur mit einem Lötöfen gelötet werden können. Da ein Lötöfen nicht überall verfügbar ist, haben wir die Masseflächen versuchsweise nicht angelötet. Das GPS-Modul arbeitete trotzdem absolut perfekt.

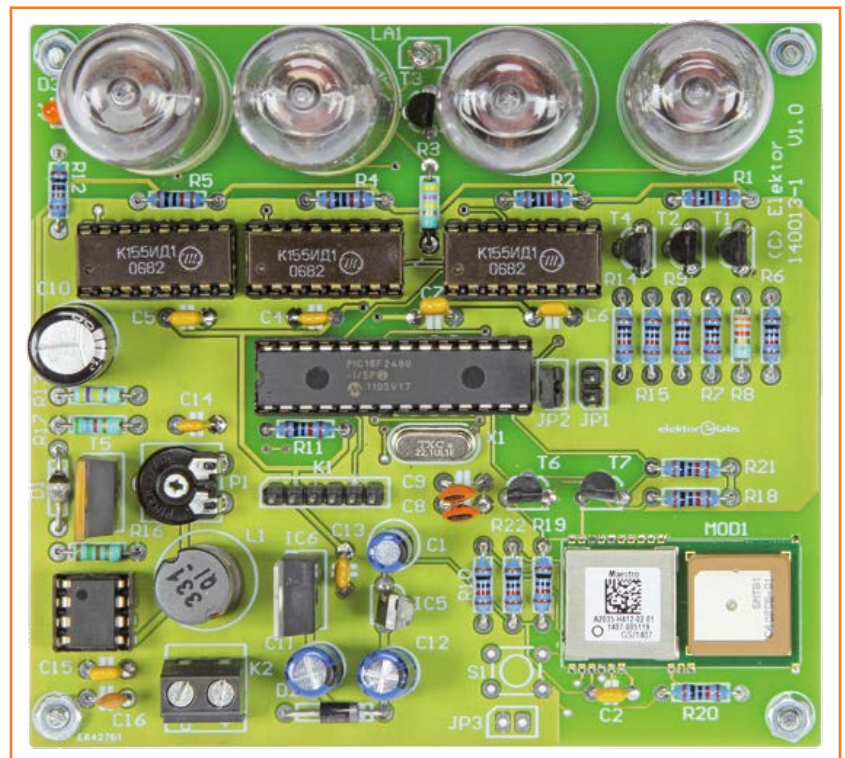
Die Bauhöhe der Bauelemente bestimmt die Reihenfolge ihrer Montage. Zuerst werden die niedrigen Bauelemente montiert, dann folgen die hohen Teile. Die Spannungsregler kommen ohne Kühlkörper aus. Besonderer Vorsicht bedürfen die Nixie-Röhren, denn sie sind zerbrechlich, und die Anschlussdrähte haben einen niedrigen Querschnitt. Für diesen Röhren-Typ sind Fassungen im Handel, die Fassungen erhöhen die Standfestigkeit auf der Platine. Das Einfädeln der Anschlussdrähte in die Platinenbohrungen ist weniger mühsam, wenn die Drähte treppenförmig gekürzt werden (Achtung: Nicht zu kurz!). Nach dem Einfädeln wird zunächst nur ein Draht angelötet, dann wird die Röhre exakt senkrecht ausgerichtet, und erst wenn das geschehen ist, folgen die übrigen Drähte.

Nachdem die Platine fertiggestellt ist, muss nur noch die Stromversorgung angeschlossen werden. Das kann zum Beispiel ein Steckernetzteil

sein, das die Gleichspannung 12 V liefert und mit 1 A belastbar ist.

Der Einbau der Platine in ein stilgerechtes Gehäuse, aus dem die Nixie-Röhren bis zum Fuß herausragen, vollendet das Nixie-Uhr-Design. Das Gehäuse ist gleichzeitig ein Berührungsschutz gegen die hohe Anodenspannung der Nixie-Röhren. Wir wünschen Ihnen stets guten GPS-Empfang und viel Freude mit Ihrer nicht alltäglichen Nixie-Uhr!

(140013)gd

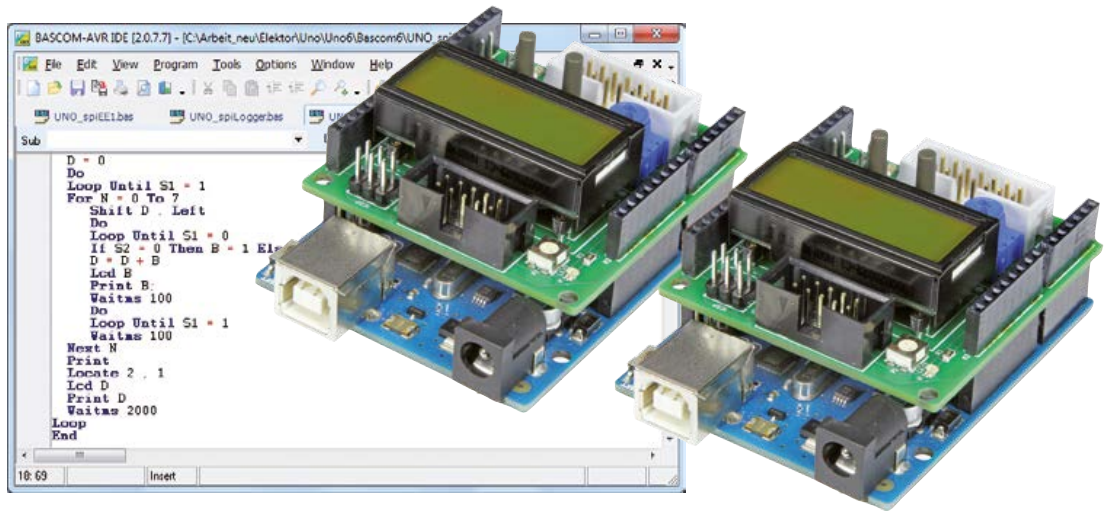


Weblinks

- [1] Sputnik-Zeitmaschine:
www.elektor-magazine.de/050018
- [2] Nixie-Thermometer/Hygrometer:
www.elektor-magazine.de/110321
- [3] www.elektor-magazine.de/140013

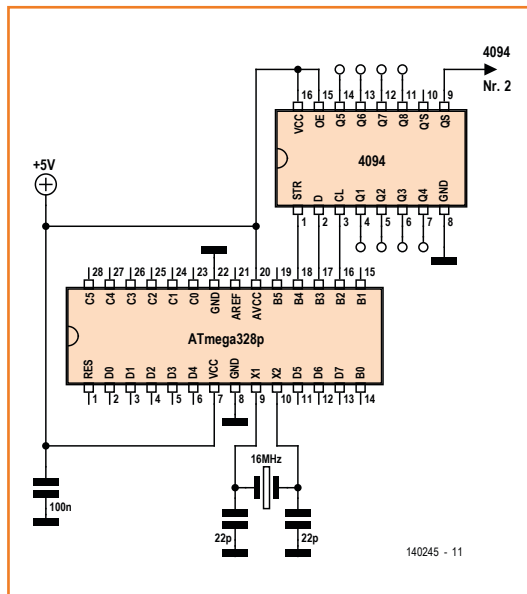
Mikrocontroller für Einsteiger (6)

Die SPI-Schnittstelle



Von **Burkhard Kainka**

Informationen seriell übertragen, immer schön eine Informationseinheit nach der anderen, das ist eigentlich der Normalfall. Wir kennen das von Sprache, Telefon und Telegrafie, Schreiben und Lesen. In vielen Fällen reicht dann ein einzelner Draht zur Übertragung der Informationen. Mit einer Taktleitung wird das Ganze aber noch zuverlässiger!



Beim Seriellen Peripherie-Bus (SPI) laufen die eigentlichen Daten Bit für Bit über eine Leitung; zum Beispiel von einem Mikrocontroller zu einem Display, einem EEPROM oder einer SD-Karte. Da man meist auch Daten lesen will, braucht man eine zweite Leitung, auf der Bits zurückflitzen können. Aber dann kommt noch etwas hinzu, nämlich eine Taktleitung. Das Taktsignal zeigt immer eindeutig, wann das nächste Bit auf der Leitung steht. Unnötig sind dann genaue zeitliche Vereinbarungen, die bei beiden Gesprächspartnern durch einen Timer überwacht werden müssten. Mit den drei Leitungen wird die Übertragung ziemlich robust.

Porterweiterung mit Schieberegister

Der erste Versuch zeigt noch keine echte SPI-Schnittstelle, aber etwas ganz ähnliches. Schieberegister gab es schon lange vor dem ers-

Bild 1.
Anschluss eines
Schieberegisters 4094.

ten Mikrocontroller, und sie sind gut geeignet, um das Prinzip einer seriellen Übertragung zu verstehen. Außerdem lassen sie sich sinnvoll zusammen mit einem Mikrocontroller verwenden. Portleitungen sind immer knapp, besonders beim Arduino. Eine Porterweiterung mit einem Schieberegister kann da helfen. Mit einem 8-bit-Schieberegister 4094 benötigt man drei Leitungen und erhält im Endergebnis acht neue Ausgänge. Oder 16, wenn man ein zweites Schieberegister anschließt, oder auch 80 Ausgänge, wenn man zehn Bausteine hintereinander schaltet. Das ist eine besonders preiswerte Methode, wenn man viele Ausgänge braucht.

Bild 1 zeigt den Anschluss an den Uno. Das 8-bit-Schieberegister hat einen Takteingang CL und einen Dateneingang D. Man legt nacheinander, beginnend mit dem höchstwertigen Bit, die Daten an den D-Eingang und erzeugt für jedes Bit einen positiven Taktimpuls an CL. Die Daten werden dann Schritt für Schritt durch die einzelnen Flipflops des Registers geschoben. Zusätzlich gibt es noch den Strobe-Eingang Str. Mit einem Strobe-Impuls werden alle Daten, die sich gerade im Schieberegister befinden, an die D-Flipflops der Ausgänge durchgeschaltet. Man könnte den Strobe-Eingang zwar auch fest an Vcc legen, aber dann würden alle Zwischenergebnisse eines Schiebevorgangs nach außen sichtbar. Mit einem Strobe-Impuls am Ende bekommt man dagegen nur das fertige Ergebnis zu sehen.

Die Programmierung (**Listing 1**) ist einfach. Um ein Byte D auszugeben wird zunächst einmal dessen höchstwertiges Bit in die Bit-Variablen B kopiert ($B = D.7$) und von da an den entsprechenden Port gelegt. Dann wird an CL ein positiver Taktimpuls mit einer Länge von einer Millisekunde erzeugt. Eine Mikrosekunde würde auch reichen, aber die verlangsamte Ausgabe dient dazu, dass man sich alles bequem mit dem Oszilloskop ansehen kann. Nach dem Taktimpuls sorgt ein Schiebebefehl (Shift D, Left) dafür, dass alle Bits in D um eine Stelle nach links geschoben werden. Damit kommt als nächstes das ehemalige Bit 6 an den Ausgang. Und so weiter, bis alle acht Bits verschoben sind. Am Ende folgt noch der Strobe-Impuls, und schon liegen alle acht Bits an den Ausgängen des 4094.

Damit sich während der Programmausführung auch etwas an den Ausgängen des Schieberegisters ändert, wird im Code das zu übertragende Datenbyte laufend hochgezählt. Der aktu-

elle Stand wird jeweils auf dem LCD (wenn das Elektor Extension-Shield verwendet wird) und am Terminal ausgegeben. So kann man das Ergebnis leicht mit den Ausgangszuständen des Schieberegisters vergleichen.

Listing 1. Ausgabe über ein Schieberegister.

```
'-----
'UNO_shift.BAS  Shift Register 4094
'-----

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

Dim Dat As Byte
Dim D As Byte
Dim N As Byte
Dim B As Bit

Sr Alias Portb.4      '4094 pin 1
Da Alias Portb.3      '4094 pin 2
Cl Alias Portb.2      '4094 pin 3
Config Portb = Output

...

Dat = 0
Do
  Cls
  Lcd Dat
  Lcd " "
  Print Dat
  D = Dat
  For N = 1 To 8
    B = D.7
    Da = B
    Waitms 1
    Cl = 1
    Waitms 1
    Cl = 0
    Waitms 1
    Shift D, Left
  Next N
  Sr = 1
  Waitms 1
  Sr = 0
  Waitms 1
  Dat = Dat + 1
  Waitms 500
Loop
End
```

Wenn man mehr als acht Ausgänge benötigt, dann hilft der Qs-Ausgang des 4094. Alle Daten die vorn hineingeschoben werden erscheinen acht Taktimpulse später an Qs. Hier kann nun der D-Eingang des nächsten Schieberegisters angeschlossen werden. So lassen sich beliebig viele 4094 hintereinanderschalten, wobei Clock und Strobe jeweils parallel geschaltet werden. Das Programm muss dann entsprechend erweitert werden. Zuerst schiebt man alle 16 (oder auch 80) Datenbits heraus, dann gibt es den gemeinsamen Strobe-Impuls.

Manuelle Datenübertragung

Die eigentlichen Daten wandern zwar über eine einzelne Leitung, aber man braucht bei einer getakteten seriellen Schnittstelle (wie hier beim

Schieberegister oder beim SPI-Bus) zusätzlich mindestens noch eine Taktleitung. Ein Vergleich mit der Morsetelegrafie zeigt den Unterschied. Dort muss man sich auf eine Geschwindigkeit einigen. Und es dürfen keine Unterbrechungen mitten in einer Informationseinheit kommen. Wenn ein Funker zum Beispiel ein X senden will (- . . -) und mittendrin möchte er sich am Kopf kratzen, dann werden daraus die beiden Zeichen N (-.) und A (.-). Genauso ist es bei einer asynchronen seriellen Schnittstelle, wo eine Baudrate vereinbart werden muss. Wenn die Übertragung eines Bytes gestartet wurde, müssen alle Bits in einem exakten Zeitrahmen gesendet werden. Beim SPI dagegen sind die Zeiten unkritisch, hier sind beliebige Verzögerungen erlaubt. Mit dem zusätzlichen Taktsignal wird die Übertragung völ-

Listing 2. SPI-Master und Slave ganz langsam.

```
'-----
'UNO_spi1.BAS  Shift in/out
'-----
$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600

Dim D As Byte
Dim B As Bit
Dim N As Byte

S1 Alias Pinc.0
Portc.0 = 1
S2 Alias Pinc.1
Portc.1 = 1
Led1 Alias Portc.2
Ddrc.2 = 1
Led2 Alias Portb.2
Ddrb.2 = 1

...

Do
  D = Rnd(255)
  Cls
  Lcd D
  Print D
  Locate 2 , 1
  For N = 0 To 7
    B = D.7
    Lcd B
    Print B;
    Led2 = B
    Waitms 300

    Led1 = 1
    Waitms 200
    Led1 = 0
    Waitms 500
    Shift D , Left

  Next N
  Led1 = 0
  Led2 = 0
  Waitms 2000
  Cls
  Print
  D = 0
  Do
    Loop Until S1 = 1
    For N = 0 To 7
      Shift D , Left
      Do
        Loop Until S1 = 0
        If S2 = 0 Then B = 1 Else B = 0
        D = D + B
        Lcd B
        Print B;
        Waitms 100
      Do
        Loop Until S1 = 1
        Waitms 100
    Next N
    Print
    Locate 2 , 1
    Lcd D
    Print D
    Waitms 2000
  Loop
End
```

lig unabhängig von der Geschwindigkeit. Ob es nun ein Bit pro Minute oder eine Million Bits pro Sekunde sein sollen, die Übertragung funktioniert. Dabei gibt es immer einen SPI-Master und einen SPI-Slave. Daten können in beide Richtungen laufen, aber der Master gibt immer den Takt vor. Das Verfahren soll nun einmal im Handbetrieb ausprobiert werden, wobei der Anwender zum Master wird. Durch Betätigen der beiden Tasten (**Bild 2**) kann man ein Byte übertragen. Das ist zwar nicht üblich, hilft aber die Sache genau zu verstehen. Man muss nur diese beiden „Morse-tasten“ bedienen, eine für die Daten und eine für den Takt. Sonst muss man nichts Neues lernen, denn den Aufbau eines Bytes kennt jeder ja schon. Und das geht so: Zuerst überträgt man das Bit 7. Wenn es eine 1 ist, betätigt man S2, sonst nicht. Und dann drückt man einmal kurz auf S1 ohne S2 zu verändern. Die Empfangsstation (der Slave, in diesem Fall der Arduino) weiß dann, wann er das Bit von der Leitung lesen soll. Dann folgen Bit 6, Bit 5 usw., bis Bit 0.

Das Programm in **Listing 2** zeigt beide Datenrichtungen. Zuerst ist der Controller der Master, Sie sind der Slave. Es wird ein zufälliges Byte gesendet, wobei Ihnen die LED1 das Taktsignal zeigt und LED2 die Daten. Mit einiger Konzentration kann man das übertragene Byte lesen. Einfach ist das allerdings nicht, deshalb wird das Byte auch im LCD verraten und zusätzlich ans Terminal geschickt. Außerdem erscheinen die einzelnen Bits nacheinander auf dem LCD und im Terminal.

83

01010011

Und dann wechseln die Rollen. Nun sind Sie der Master und haben die Aufgabe, genau dieses Byte wieder zurück zu senden. Und da zeigt es sich, dass die beliebige Geschwindigkeit ein großer Vorteil ist. Sie können nämlich in aller Ruhe überlegen, welches Bit als nächstes kommt. Sie sollen z. B. eine 100 senden. Das Bit 7 hat ja eine Wertigkeit von 128 und ist demnach nicht dabei. Also eine Null. Dann kommt aber das Bit 6 (1×64), und Sie rechnen aus, dass nun noch 36 übrig sind. Also 1×32 , da sind es noch 4. Es folgen also zwei Nullbits (0×16 und 0×8), dann kommt eine 1 ($\times 4$) und noch zwei Nullen. Sie haben die Binärzahl 01100100 gesendet und jedes Bit mit einem Taktimpuls bestätigt. War vielleicht nicht ganz einfach, aber der Controller liest es ganz

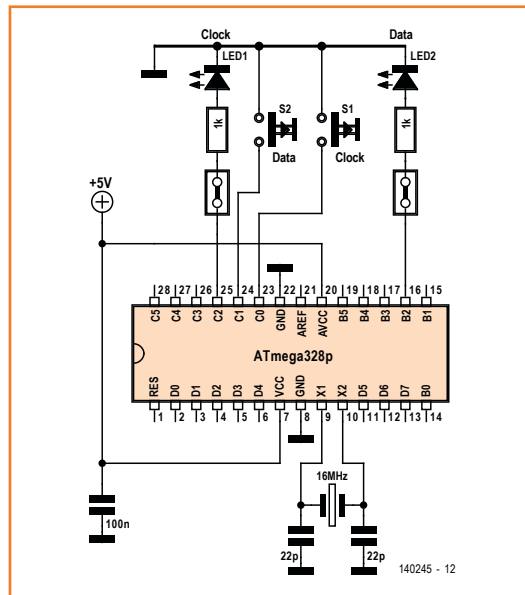


Bild 2.
Manuelle Ein- und Ausgabe.

locker mit. Das Ganze ist ein Konzentrationsspiel, und wenn der Controller das korrekte Ergebnis im LCD zeigt, haben Sie gewonnen.

Ein genauer Blick auf das Listing 2 zeigt, dass die gelesenen Bits invertiert werden. Das wurde so gemacht, weil eine gedrückte Taste ein Nullbit ergibt. Damit der Anwender nicht umdenken muss, wird das Ergebnis beim Lesen umgedreht. Interessant ist auch die Erzeugung einer Pseudo-Zufallszahl mit $D = \text{Rnd} (255)$. Tatsächlich wird immer dieselbe Zahlenfolge erzeugt. Aber die Bascom-Hilfe gibt Hinweise, was man da noch tun könnte.

Von Controller zu Controller

In diesem Beispiel werden Daten per SPI von einem Controller zu einem anderen geschickt. Diesmal sollen es 10-bit-Daten des AD-Wandlers sein. Da zeigt sich ein weiterer Vorteil des SPI, die Datenbreite ist flexibel. Ob es nun 8, 10, 12 oder 16 bit sind, das Verfahren ist immer das gleiche. Tatsächlich hätte man mit einer asynchronen seriellen Schnittstelle über TXD und RXD weniger Aufwand, wenn es nur darum geht, zwei Controller zu verbinden. Der SPI-Bus spielt dagegen seine Vorteile bei der Ansteuerung externer Hardware richtig aus. Der Versuch dient also in erster Linie der Verdeutlichung des Übertragungsverfahrens. Diesmal kommt wie schon beim Schieberegister 4094 eine dritte Leitung zum Einsatz, die Chip-Select-Leitung /CS. Der Strich / steht dafür, dass diese Leitung im aktiven Zustand low ist. Die

Listing 3. Der SPI-Master.

```

'-----
'UNO_spi2.BAS  SPI Master
'-----
$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600
Dim B As Bit
Dim Dout As Word
Dim N As Byte
Dim I As Byte

Sck Alias Portb.5
Ddrb.5 = 1
Mosi Alias Portb.3
Ddrb.3 = 1
Cs Alias Portb.2
Ddrb.2 = 1

Cs = 1
Mosi = 0
Sck = 0

Config Adc = Single , Prescaler = 32 ,
Reference = Avcc
Start Adc

Cls
Cursor Off

Waitms 200
Do
    Dout = Getadc(3)    'Pot
    Locate 1 , 1
    Lcd Dout
    Lcd "    "
    Cs = 0
    Waitms 20
    For N = 1 To 10
        Mosi = Dout.9
        Waitms 1
        Sck = 1
        Waitms 1
        Sck = 0
        Waitms 1
        Shift Dout , Left
    Next N
    Cs = 1
    Waitms 100
Loop
End

```

Listing 4. Der SPI-Slave.

```

'-----
'UNO_spi3.BAS  SPI Slave
'-----
$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600

Dim Addr As Byte
Dim B As Bit
Dim Dout As Word
Dim Din As Word
Dim N As Byte
Dim I As Byte

S1 Alias Pinc.0
Portc.0 = 1
S2 Alias Pinc.1
Portc.1 = 1
Sck Alias Pinb.5
Portb.5 = 1
Mosi Alias Pinb.3
Portb.3 = 1
Cs Alias Pinb.2

Portb.2 = 1
...

Do
    Do
        Loop Until Cs = 0
        Din = 0
        For N = 1 To 10
            Shift Din , Left
            Do
                Loop Until Sck = 1
                Din = Din + Mosi
            Do
                Loop Until Sck = 0
        Next N
        Do
            Loop Until Cs = 1
            Locate 1 , 1
            Lcd Din
            Lcd "    "
            Print Din
        Loop
    End

```

CS-Leitung kann eingesetzt werden, um mehrere Slaves an einen Master zu hängen. Sie teilen sich dann die Datenleitung und die Taktleitung, bekommen aber jeweils eine eigene CS-Leitung. Wenn diese low ist, dann weiß der entsprechende Slave, dass er gemeint ist. Die Sache hat aber auch noch einen zweiten Nutzen. Wenn der Slave erst mit Verspätung eingeschaltet wurde, könnte es zu einem Missverständnis kommen, welche Bits gerade übertragen werden. Wenn er aber erstmal abwartet, bis seine CS-Leitung von high auf low gezogen wird (fallende Flanke), dann weiß er, dass die Übertragung beginnt. Und sollte einmal ein Störimpuls als Taktsignal gelesen werden, sind zwar die folgenden Daten nur noch Müll, beim nächsten Zugriff ist aber wieder alles korrekt. Der ATmega328 verwendet den SPI-Bus auch zu seiner eigenen Programmierung über ein externes Programmiergerät. Am sechspoligen Programmierstecker auf dem Arduino und dem Elektor Extension-Shield (ICSP in **Bild 3**) findet man daher die Taktleitung SCK (Serial Clock, B5), die Datenleitung zum Schreiben MOSI (Master out Slave in, B3) und die Datenleitung zum Lesen MISO (Master in Slave out, B4). Eine CS-Leitung gibt es nicht, aber Reset tritt an ihre Stelle, denn der Programmiervorgang findet bei heruntergezogener Reset-Leitung statt. Diese Leitungen sollen bei uns nun in genau der gleichen Funktion verwendet werden. Das hat den Vorteil, dass man bei Gelegenheit auch die Hardware-SPI-Einheit

Für den Slave (**Listing 4**) sind die drei Leitungen Eingänge. Er wartet jeweils auf bestimmte Flanken an CS und SCK und liest dann ein Bit an MOSI ein. Weil hier alles per Software erledigt wird, muss jede Flanke in einer Do-Loop-Schleife abgewartet werden. Dafür wird etwas Zeit benötigt, sodass die Übertragung langsamer sein muss als mit einem Hardware-SPI. Die empfangenen Daten werden auf dem Display und im Terminal angezeigt. Dreht man am Poti des Masters, dann wird die Änderung beim Slave sichtbar.

Es gibt zahlreiche Bausteine mit SPI-Anschluss, vom AD-Wandler über Speicher bis zum Display-

treiber. Häufig eingesetzt und preiswert erhältlich sind serielle EEPROMs von Microchip und anderen Firmen. Der 25LC512 (nicht zu verwechseln mit dem 24C512 mit I2C-Bus) bietet 64 KB, und wenn das interne EEPROM des Mega328 mit nur einem Kilobyte mal nicht reicht, kommt dieser Chip gerade recht. Im Hobbybereich sind die I2C-EEPROMs weiter verbreitet, im professionellen Bereich dagegen die SPI-Typen, weil sie eine besonders große Betriebssicherheit bieten.

Bild 4 zeigt die Verbindung mit dem Uno. Hier

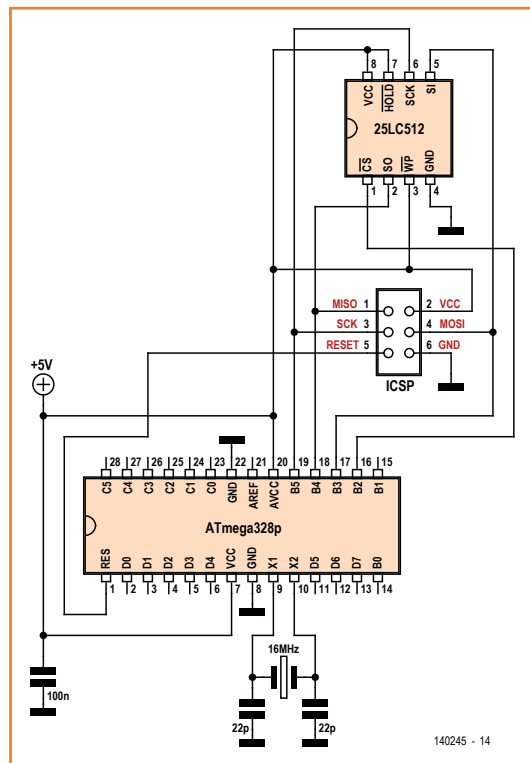


Bild 4.
Anschluss eines seriellen
EEPROMs.

werden wieder die Anschlüsse der originalen SPI-Schnittstelle (2x3-Stiftleiste) verwendet. Daher bietet es sich an, das IC samt Sockel an eine kleine sechspolige Pfostenbuchse zu löten und damit ein bequemes einsetzbares Speichermodul zu bauen. Nur eine weitere Leitung muss extern angeschlossen werden. Es handelt sich um die Chip-Select-Leitung, die wieder an B2 gelegt wird, weil die am Stecker vorhandene Reset-Leitung nicht verwendet werden kann.

Diesmal gibt es zwei Datenleitungen. MOSI (Master Out Slave In) ist die Ausgabeleitung und geht an den seriellen Eingang SI des EEPROMs, während über MISO (Master In Slave Out) Daten von SO gelesen werden. Der Mikrocontroller ist immer der Master und legt den Takt an SCK. Passend zu dieser Aufgabe haben wir ein Unterprogramm (eine Sub, siehe **Kasten**) gebildet, das die Daten in beiden Richtungen in einem Rutsch überträgt (**Listing 5**). Vor dem Aufruf muss man Daten in der globalen Variablen Dout bereitstellen, nach dem Aufruf erhält man gelesene Daten in Din zurück. Tatsächlich gibt es Situationen, in denen nur geschrieben wird, aber dann werden üblicherweise Nullbits in die andere Richtung gesendet. Was man nun wann und wie übertragen muss geht aus dem relativ komplexen Datenblatt des 25LC512 hervor. Der Speicherchip erhält zunächst (nachdem CS heruntergezogen wurde) einfache Byte-Kommandos, die festlegen, welche Aktion ausgeführt werden soll. Um den Speicher auszu-lesen muss man ein Kommando 3 senden, gefolgt von zwei Adressbytes (dem Highbyte und dem Lowbyte der Adresse). Danach können beliebig viele Datenbytes ausgelesen werden, wobei die Adresse automatisch erhöht wird (vgl. **Listing 6**). Das Programm zeigt die laufende Adresse und die ausgelesenen Datenbytes an. Ein fabrikneues oder ein vollständig gelöscht EEPROM liefert immer nur den Wert 255. Also sollen nun Daten programmiert werden. Dafür gibt es das Bytekommando 2. Aber es ist eine zusätzliche Vorbereitung nötig. Das Schreiben muss durch ein Kommando 6 freigeschaltet werden (**Listing 7**). Ob das gelungen ist kann man im Statusregister des EEPROMs nachsehen, das mit einem Kommando 5 ausgelesen wird. Jede Aktion für sich wird nur wirksam, wenn man am Anfang CS herunterzieht und am Ende wieder hochlegt. Wenn nun der Wert 2 des Statusregisters angezeigt wird, dann ist das IC für Schreibvorgänge freigeschaltet. Umständlich? Ja, aber besonders fehlersicher! Nun dürfen also Daten geschrieben werden. Aber

Listing 5. Schreiben und Lesen über MOSI und MISO.

```
Sub Spioutin
    Din = 0
    For N = 0 To 7
        Shift Din , Left
        Din = Din + Miso
        If Dout.7 = 1 Then Mosi = 1 Else Mosi = 0
        Waitus 3
        Sck = 1
        Waitus 2
        Sck = 0
        Waitus 2
        Shift Dout , Left
    Next N
End Sub
```

Listing 6. Auslesen des EEPROMs.

```

Cs = 0
Dout = 3          'read
Spioutin
Dout = 0          'A8...A15
Spioutin
Dout = 0          'A0...A7
Spioutin
I = 0
Do
  Locate 1 , 1
  Lcd I
  Print I;
  Print " ";
  I = I + 1
  Spioutin
  Locate 2 , 1
  Lcd Din
  Print Din
  Waitms 200
Loop

```

auch da ist einiges zu beachten. Der Speicherbereich ist in Seiten gegliedert, die beim 25LC512 eine Größe von 128 Bytes haben. Man darf maximal 128 Bytes bzw. Daten bis zur nächsten Seitengrenze übertragen. Dann schaltet man CS hoch und lässt dem EEPROM genügend Zeit für den eigentlichen Programmiervorgang. Laut Datenblatt reicht dazu eine Programmierzeit von 5 ms. Überschreibt man die Seitengrenze (ich hab's probiert), dann gibt es Chaos. Man findet dann ganz andere Daten im Speicher als man hineinschreiben wollte. Das Beispielprogramm schreibt daher ganz brav 128 Bytes in die erste

Listing 7: Freischalten der Schreibzugriffe.

```

Cs = 0
Dout = 6          'write enable
Spioutin
Cs = 1
Waitms 20
Cs = 0
Dout = 5          'read status
Spioutin
Spioutin
Cs = 1
Locate 1 , 1      'status
Lcd Din
Print Din

```

Unterprogramme

Ein Unterprogramm heißt in Bascom Sub und wird dann eingesetzt, wenn ein Programmteil immer wieder von verschiedenen Stellen des Programms aus aufgerufen werden soll. Am Anfang muss die Sub aber deklariert werden. Der Name des Unterprogramms ist dem Compiler dann bekannt; das Unterprogramm kann nun über diesen Namen wie eine Bascom-Funktion aufgerufen werden.

```

Declare Sub Spioutin()
...
Dout = 6
Spioutin
...

Sub Spioutin
  Din = 0
...
  Shift Dout , Left
End Sub

```

Bei den Variablen, die das Unterprogramm bearbeitet, muss man immer genau aufpassen. In diesem Fall arbeitet die Sub Spioutin nur mit sogenannten globalen Variablen, die schon am Anfang des Hauptprogramms dimensioniert wurden und daher im gesamten Programm gültig sind. Man könnte es aber auch anders machen und die Daten mit dem Aufruf der Sub übergeben:

```
Declare Sub Spioutin (Byteout as Byte)
```

Die Variable Byteout gilt dann nicht global, sondern nur lokal innerhalb der Sub. Der Aufruf dieser Sub erfolgt zum Beispiel mit...

```
Spioutin 6
```

... was gegenüber dem oben gezeigten Aufruf eine Zeile einspart. Man kann Subs auch mehrere Variablen gemeinsam beim Aufruf übergeben, wie die bascom-eigene Sub Spiout zeigt:

```
Spiout Dout , 1
```

Bei den ersten Programmierversuchen ist es meist weniger fehlerträchtig, wenn man in eigenen Subs nur globale Variablen verwendet. Für den fortgeschrittenen Programmierer ist dagegen die optimale Datenübergabe ein wichtiges Thema.

Listing 8. Verwendung der Software-SPI.

```

Config Spi = Soft , Din = Pinb.4 , Dout = Portb.3 ,
Ss = None , Clock = Portb.5 , Mode = 0
Spiinit

Cs Alias Portb.2
Ddrb.2 = 1
Cs = 1

    Cs = 0
    Dout = 4          'write disable
    Spiout Dout , 1
    Cs = 1
    Waitms 1
    Cs = 0
    Dout = 3          'read
    Spiout Dout , 1
    Dout = 0          'A8...A15
    Spiout Dout , 1

Dout = 0          'A0...A7
Spiout Dout , 1
I = 0
Do
    Locate 1 , 1
    Lcd I
    Lcd " "
    Spiin Din , 1
    Locate 2 , 1
    Lcd Din
    Lcd " "
    Print I ;
    Print " ";
    Print Din
    Waitms 200
    I = I + 1
    If I >= 65535 Then Exit Do
Loop

```

Seite von Adresse 0 bis Adresse 127, und zwar eine aufsteigende Zahlenfolge, ebenfalls von 0 bis 127. Und genau diese Daten findet man dann auch beim Auslesen.

Das gesamte Programm UNO_spiEE1.bas kann wie immer von der Elektor-Homepage [1] geladen werden. Es führt die folgenden Aktionen nach-

einander aus:

- Kommando 6, Write Enable
- Kommando 5, Statusregister auslesen, Anzeige für eine Sekunde
- Kommando 2, 128 Bytes schreiben ab Adresse 0
- Kommando 3, Auslesen ab Adresse 0, Endlosschleife

Listing 9. Datenspeicherung im Timer-Interrupt.

```

Tim0_isr:
    '4000 µs
    Timer0 = 6
    Ticks = Ticks + 1
    If Ticks = 250 Then
        Ticks = 0
        U = Getadc(4)
        U = U / 4
        Addr = Seconds
        Addr = Addr And 127
        If Addr = 0 Then          'start of page
            Cs = 0
            Dout = 6              'write enable
            Spiout Dout , 1
            Cs = 1
            Waitus 100
            Cs = 0
            Dout = 2              'write
            Spiout Dout , 1
            Waitus 100
            Cs = 0

            Dout = 3
            Addr = High(seconds)
            Dout = Addr          'A8...A15
            Spiout Dout , 1
            Addr = Low(seconds)
            Dout = Addr          'A0...A7
            Spiout Dout , 1
        End If
        Dout = U
        Spiout Dout , 1
        Addr = Seconds
        Addr = Addr And 127
        If Addr = 127 Then      'End of page
            Cs = 1
        End If
        Print Seconds
        Seconds = Seconds + 1
        If Seconds = 0 Then Seconds = 65535
    End If
    Return

```

Ein Datenlogger

Eine praktische Anwendung für den 64 KB großen Speicher ist ein Datenlogger. Messdaten vom Analogeingang ADC4 sollen einmal pro Sekunde aufgenommen und gespeichert werden. Der Speicher ist dann nach etwa 18 Stunden voll.

Man muss nicht immer alles selbst programmieren, denn Bascom bietet ja für viele Fälle bereits fertige Funktionen. In diesem Fall kann eine SPI-Schnittstelle wahlweise als Software-Schnittstelle an beliebigen Portpins eingerichtet werden oder als Hardware-Schnittstelle an genau den dafür vorgesehenen Pins des Controllers. Die Hardware-SPI ist besonders schnell und wird z.B. gern für die Ansteuerung grafischer Displays verwendet. Allerdings gibt es da fürchterlich viele Parameter, für die man von Fall zu Fall ein tiefgehendes Studium des Mega328-Datenblatts braucht. Mit der Software-SPI-Funktion geht es etwas einfacher, und sie bietet ebenfalls schon eine recht hohe Übertragungsgeschwindigkeit. Eine von Grund auf selbst geschriebene SPI-Prozedur ist zwar auch nicht zu verachten, weil man da sehr anschaulich die Impulsdigramme der Datenblätter nachprogrammieren kann. Aber die Software-SPI ist noch bequemer und schneller, deshalb soll sie hier verwendet werden.

Die Konfiguration der Schnittstelle legt fest, welche Leitungen verwendet werden sollen. Für Din, Dout und Clock nutzen wir die altbekannten Leitungen MISO, MOSI und SCK (wie immer am ICSP-Stecker zugänglich). Die SS-Leitung entspricht der CS-Leitung, sie soll hier nicht automatisch von Bascom bedient werden, weil beim EEPROM meist viele Daten während einer einzigen aktiven CS-Phase übertragen werden müssen. Diese Leitung (Portpin B2) wird also weiterhin

„per Hand“ bedient. Wichtig ist noch die Einstellung Mode = 0, denn es gibt vier verschiedene SPI-Modi.

Der Programmausschnitt in **Listing 8** zeigt die Verwendung der Software-SPI beim Auslesen des seriellen EEPROMs. Mit Spiout Dout, 1 sendet man genau ein Byte, das mit der Variablen Dout übergeben wird. Umgekehrt liest man mit Spiin Din, 1 ein Byte, das danach in Din liegt. Insgesamt liest das Programm alle Daten aus dem EEPROM und zeigt den Inhalt auf dem Display und im Terminal.

Das Gesamtprogramm UNO_spiLogger.bas kann wie immer von der Elektor-Homepage heruntergeladen [1] werden. Es ist zu umfangreich, um es hier komplett abzdrukken. Ein Druck auf S1 startet eine Messreihe. Sie kann jederzeit mit S2 unterbrochen werden, um die gespeicherten Daten auszulesen.

Für eine Zeitsteuerung während der Datenerfassung wird ein Timer-Interrupt eingesetzt (Ausschnitt in **Listing 9**). Einmal pro Sekunde wird eine Spannung an ADC4 gemessen und abgespeichert. Dabei wird die Datenblockgröße des EEPROMs von 128 Bytes beachtet. Jeweils am Anfang eines Blocks wird ein Schreibzugriff gestartet und die aktuelle Adresse übertragen. Danach folgen 128 Bytes. Am Ende des Blocks wird die CS-Leitung hochgezogen, damit das serielle EEPROM den ganzen Block speichern kann. Da die CS-Leitung an Port PB2 liegt, wird gleichzeitig die LED2 auf den Shield eingeschaltet. Man sieht daher jeweils am Aufblitzen der LED, wann ein Block abgespeichert wird.

(140245)

Weblinks

[1] www.elektor-magazine.de/140245

Tipps zum Arduino-Programmer in Bascom

Viele Leser verwenden den originalen Arduino-Bootloader und sind auf folgendes Problem gestoßen: Wenn ein Programm eine serielle Ausgabe enthält, funktioniert der Arduino-Programmer in Bascom beim nächsten Mal nicht mehr so wie er sollte und hängt sich fest. Einige Leser haben herausgefunden, dass man das Problem lösen kann, wenn man zwischendurch die Arduino-IDE verwendet; es reicht hier z.B. das Blink-Programm zu übertragen.

Aber es geht auch einfacher: Nach dem Start des Programmers in Bascom drückt man mindestens dreimal im Abstand von etwa einer Sekunde kurz auf die Reset-Taste des Uno. Dann funktioniert die Programmierung auch in den Fällen, wo das letzte Programm eine Printausgabe enthielt. Einen intensiven Erfahrungsaustausch dazu findet man im Elektor-Forum „BASCOM-AVR-Kurs“ (www.elektor-magazine.de/forum).

Wer übrigens den MCS-Bootloader im Uno verwendet, hat dieses Problem nicht. Eine weitere Alternative ist die Verwendung eines externen Brenners.

Einer für alles

Der neue Debugger/Programmer Atmel-ICE



Von
Wolfram Pioch
(D)

Auf der letzten Embedded World stellte Atmel einen neuen Low-Cost-Programmer/Debugger vor, der sich gleichermaßen für die AVR-Controller inklusive Xmega wie auch die ARM-Cortex-Controller des Herstellers eignet. Wir haben das kleine Kästchen ausprobiert.

Bild 1.
In der Basic-Variante wird nur ein Kabel mitgeliefert, das einen 10-poligen und einen 6-poligen Steckverbinder bietet (ersterer im 1,27-mm-Raster).

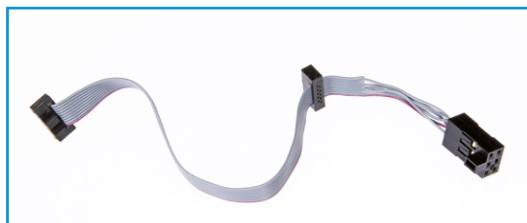
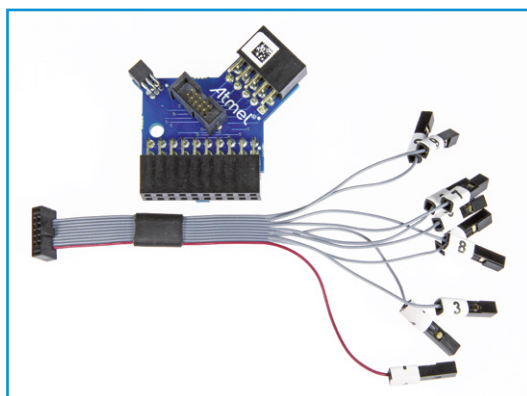


Bild 2.
Das „Full Kit“ enthält noch ein Squid-Kabel mit Einzelbuchsen und einen sternförmigen Adapter.



Der Atmel-ICE (ICE = In Circuit Emulator) ist der neueste Programmer/Debugger von Atmel. Er kann seine Verwandtschaft zum AVR-Debugger JTAGICE3 nicht leugnen. Die Gehäusegröße ist die gleiche (85x 55x 16 mm), darüber hinaus ist dieselbe 10-polige Buchse im 50-mil-Raster vorhanden. Per Adapter-Kabel geht es von hier aus weiter zu den üblichen Programmiersteckern, die man auf den verschiedenen AVR-Entwicklungs- und Prototyping-Boards findet.

Doch der Atmel-ICE kann noch mehr als nur ATmega/Xmega/ATtiny-Controller zu debuggen. Bei seiner Entwicklungssoftware „AVR Studio“ hat der Hersteller ja bereits erfolgreich gezeigt, wie sich eine Unterstützung für die herstellereigenen ARM-Controller integrieren lässt (daher heißt die IDE jetzt auch nicht mehr „AVR Studio“ sondern „Atmel Studio“).

Auch beim Atmel-ICE gilt die Maxime „One fits for all“. Eine zweite 10-polige Buchse („SAM“) ist für das Programmieren und Debuggen von ARM-Con-

trollern gedacht, wie zum Beispiel dem SAM3X8E (ARM Cortex M3) auf dem Arduino-Due-Board. Wie ein Vergleich der technischen Daten mit bereits etablierten Atmel-Debuggern zeigt (siehe Tabelle), muss man beim neuen Atmel-ICE aber Abstriche in punkto Geschwindigkeit machen. So liegt die maximale Taktrate beim Programmieren von ARM-Controllern deutlich unter der des Debuggers SAM-ICE.

Adapter

Das Tool gibt es in drei Varianten: Als Board ohne Gehäuse, als Basic-Variante im Gehäuse und mit einem mitgelieferten Kabel sowie darüber hinaus als „Adapter-Kit“. Wir haben uns die letzte Variante angesehen; der deutsche Atmel-Distributor Ineltek [1] stellte uns freundlicherweise eines der ersten ausgelieferten Tools zur Verfügung. Beim Elektronikversender Reichelt ist das Adapter-Kit für knapp 120 Euro zu haben.

In der Basic-Variante wird nur ein Flachbandkabel mitgeliefert (**Bild 1**). Es besitzt zwei 10-polige-Buchsen im 50-mil-/1,27-mm-Raster. Eine solche Buchse kann man zum Beispiel auf die Debug-Stiftleiste des Arduino Due stecken. Am Ende des Kabels sind aber zusätzlich sechs Leitungen auf eine 6-polige Buchse im üblichen 100-mil-/2,54-mm-Raster geführt, diese lässt sich auf die 6-poligen Stiftleisten vieler ATmega- (SPI) und Xmega-Boards (PDI) stecken.

Im Adapter-Kit sind noch ein weiteres Kabel mit zehn Einzel-Buchsen (Squid Cable) und eine Adapterplatine in Sternform enthalten (**Bild 2**). Letztere besitzt eine 20-polige Buchse im 2,54-mm-Raster für eine SAM-JTAG- bzw. SWD-Schnittstelle (wie beim SAM-ICE).

Außerdem ist hier die bekannte 10-polige JTAG-Buchse für AVR im 2,54-mm-Raster zu finden. Darüber hinaus steht noch eine 6-polige Buchse mit den ISP-, PDI-, debugWire- und aWire-Signalen zur Verfügung; allerdings in einem Raster von 1,27 mm. Das wird all diejenigen freuen, die kompakte Controller-Platinen entwickeln; denn jetzt lässt sich eine platzsparende 6-polige Stiftleiste im Raster von 1,27 mm als Programmier-/Debug-Anschluss verwenden.

Eine Übersicht über alle Möglichkeiten des neuen Tools findet man unter [2].

An den PC wird der Debugger/Programmer wie üblich über ein USB-Kabel angeschlossen. Die zugehörige Mikro-USB-Buchse brach bei unserem Testexemplar leider recht schnell von der Platine, was eine DIY-Reparatur nötig machte (siehe diese

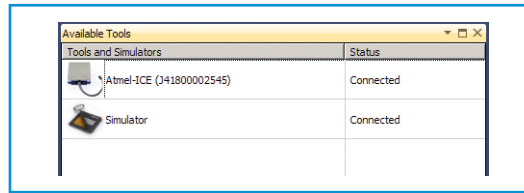


Bild 3.
Tool-Auswahl.

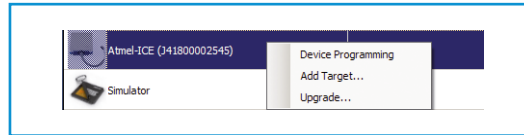


Bild 4.
Firmware-Version prüfen mit „Upgrade“.

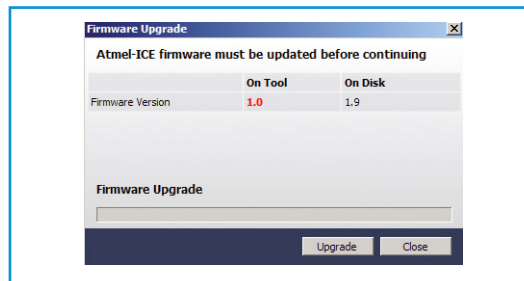


Bild 5.
Hier ist die Firmware-Version veraltet.

Ausgabe). Leider ist das ein verbreitetes Problem von Mikro-USB-Buchsen, die meist lediglich aufgelötet werden. Vielleicht entschließt sich Atmel ja bei einer zweiten Version zum Einsatz einer „normalen“ USB-B-Buchse.

Software

Der neue Atmel-ICE wird erst ab dem Atmel Studio 6.2 unterstützt. Diese Software muss vor der ersten Verwendung auf dem PC installiert sein [3]. Nach dem Einstecken des USB-Steckers in den Rechner wird das Tool automatisch erkannt und der USB-Treiber installiert. Im Studio 6.2 sieht man nun zuerst einmal nach, ob der ICE von der Software erkannt wurde.

Unter „View“ → „Available Atmel Tools“ öffnet sich das Fenster in **Bild 3**. Das Arbeiten mit dem ICE setzt die neueste Firmware-Version voraus. Nach einem Klick mit der rechten Maustaste erscheint ein Popup-Menü, in welchem man „Upgrade...“ auswählt (**Bild 4**). Wenn die Firmware des ICE nicht aktuell ist, erscheint die Meldung zum Firmware-Update in **Bild 5**.

Vor dem Debuggen eines Projekts sollte man zuerst einmal nachsehen, ob der ICE mit dem Prozessor spricht. Den ICE haben wir zunächst einmal als Programmer von drei verschiedenen Prozessoren getestet.

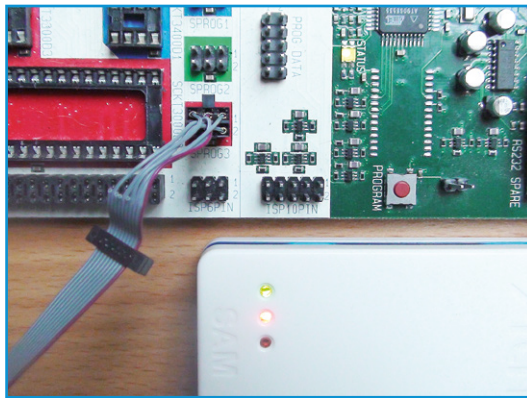


Bild 6.
ATmega32 programmieren
auf dem STK500-Board.

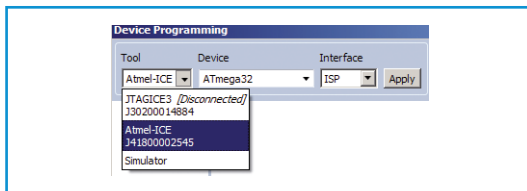


Bild 7.
Zuerst wählt man den
Atmel-ICE aus ...

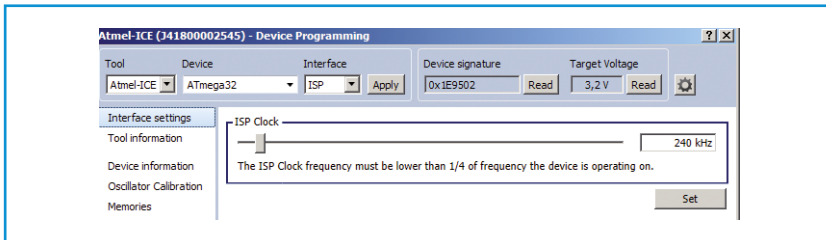


Bild 8.
... dann den Prozessor und
das Interface.

Zuerst war ein ATmega32 auf dem STK500-Entwicklungsboard an der Reihe (Bild 6). Über das Basic-Kabel wird der Atmel-ICE mit der prozessorabhängigen Stiftleiste des STK500 verbunden. Das Flachbandkabel wird so herangeführt, dass die rot markierte Ader 1 am 10-poligen Kabel auf der Seite der Markierung „1“ der STK-Stiftleiste ist. Die Nase des 6-poligen Steckverbinders

zeigt nach oben.

Im Atmel Studio wird der Programmer über „Available Atmel Tools“ → „Atmel ICE“ → „Device-Programming“ gestartet. Alternativ kann man auch das Icon mit dem Chip und dem Blitz in der Toolbar anklicken.

In dem sich öffnenden Fenster „Device Programming“ lässt sich der ICE unter „Tools“ auswählen (Bild 7). Als Device wird der gewünschte Prozessor ausgewählt (ATmega32) und als Interface „ISP“ eingestellt. Danach muss der Button „Apply“ angeklickt werden (Bild 8).

Anschließend wird unter „Interface Settings“ der ISP-Takt von rund 240 kHz eingestellt; dieser darf nicht höher sein als ein Viertel der Prozessor-Taktfrequenz. Hier sollte man nicht vergessen, dass ein fabrikneuer Prozessor standardmäßig mit dem internen RC-Oszillator und 1 MHz läuft. 240 kHz Taktfrequenz gehen aber immer. Wenn später über die Fuses ein höherer Prozessortakt konfiguriert wurde, kann man natürlich auch den ISP-Takt entsprechend höher wählen.

Als erster Kontakt zum Prozessor wird „Read“ unter „Device Signatur“ angeklickt (Bild 9). Wenn man keine Fehlermeldung bekommt, hat man den richtigen Prozessor ausgewählt.

Xmega

Nun ist ein Xmega192A3 an der Reihe, auf einem STK600-Board. Diesmal soll die PDI-Schnittstelle verwendet werden. Der Steckverbinder wird mit der Nase zum Prozessor auf die SPI/PDI-Stiftleiste aufgesetzt (Bild 10). Die Handhabung des Programmers ist die gleiche wie beim SPI-Beispiel, nur wird diesmal als Interface „PDI“ ausgewählt. Die PDI-Taktfrequenz kann auf das Maximum von 7,5 MHz gesetzt werden.

Der Xmega lässt sich aber auch über JTAG pro-

Daten der wichtigsten Atmel-Debugger

	AVR ONE!	JTAGICE3	Atmel-ICE	SAM-ICE
Interface	SPI,PDI,TPI debugWire JTAG	SPI,PDI,TPI debugWire aWire JTAG	SPI,PDI,TPI debugWire aWire JTAG/SWD	 JTAG/SWD
SPI-Clock	32 kHz ... 32 MHz	8 kHz ... 1,8 MHz	8 kHz ... 5 MHz	
JTAG-Clock	32 kHz ... 32 MHz	32 kHz ... 15 MHz	32 kHz ... 7,5 MHz	
PDI-Clock	32 kHz ... 32 MHz	32 kHz ... 10 MHz	32 kHz ... 7,5 MHz	
dW Baudrate	k.A.	4 kbit/s... 0,5 Mbit/s	4 kbit/s ... 0,5 Mbit/s	
aWire Baudrate	k.A.	7,5 kbit/s ... 7,5 Mbit/s	7,5 kbit/s ... 7 Mbit/s	
SWD-CLOCK			32 kHz ... 2 MHz	0 ... 8 MHz

grammieren, hierfür wird der Adapter des Atmel-ICE mit der Bauteilseite zum Prozessor auf die JTAG-Buchse des STK600 gesetzt. Als Interface wählen wir diesmal „JTAG“; es ist ebenfalls die maximale Taktfrequenz von 7,5 MHz nutzbar (**Bild 11**).

Zum Schluss ein Test mit dem Arduino Due (ATSAM3X8E). Das Basis-Kabel wird an die 10-polige Debug-Schnittstelle so angeschlossen (wie im **Bild 12** gezeigt), dass die gekennzeichnete Ader 1 des Kabels zur Platinenmitte zeigt. Nun müssen wir natürlich die SAM-Buchse des ICE nutzen. Benutzt man aus Versehen den AVR-Anschluss, dann geht aber nichts kaputt. Doch nur beim richtigen Anschluss leuchtet die grüne LED des Debuggers. Der Anschluss des ICE ist übrigens auch bei einem aufgesetzten Arduino-Shield problemlos möglich.

In den Interface-Settings des Programmers ist als Prozessor nun der ATSAM3X8E des Arduino Due einzutragen. Achtung! Als Interface muss „SWD“ (= Serial Wire Debug) gewählt werden, „JTAG“ funktioniert nicht. Die maximale Taktfrequenz ist (leider nur) 2 MHz.

Debuggen

Nun steht dem Debuggen eines kompletten AVR- oder ARM-Projekts nichts mehr im Wege. Hier soll die neue SAM-Schnittstelle getestet werden. Es wird empfohlen, als erstes Projekt ein passendes Beispiel des Atmel Studios auszusuchen und dieses dann den eigenen Bedürfnissen anzupassen. Das Vorgehen soll hier am Beispiel des Arduino-Due-Boards gezeigt werden. Hier kommt kein Arduino-Sketch zum Einsatz, sondern ein „richtiges“ C-Programm. In der nächsten Ausgabe zeigen wir jedoch, wie man mit dem Atmel Studio auch Arduino-Sketches entwickeln und sogar debuggen kann.

Zuerst wählen wir „File“ → „New“ → „Example project“. Es öffnet sich ein Fenster, in welchem man als Device-Familie „SAM, 32 Bit“ und in der Textbox Category „Applications“ wählt. In der dann erscheinenden Auswahl wird „Getting Started Application on SAM - Arduino Due/X“ angeklickt (**Bild 13**). Weiter unten kann dann der Projektname und der Speicherort geändert werden. Wir lassen es bei dem vorgeschlagenen Namen und wählen das Verzeichnis „D:\ATMEL ICE“. Nach dem Bestätigen mit „OK“ müssen die Lizenzbedingungen anerkannt werden, danach

meldet Atmel Studio, dass ein Projekt angelegt wird.

Nun erscheint rechts der Solution-Explorer (**Bild 14**), der den Aufbau des Projekts zeigt. Ein Klick auf den Eintrag „main.c“ öffnet die dazugehörige Datei.

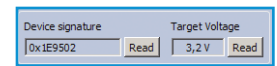
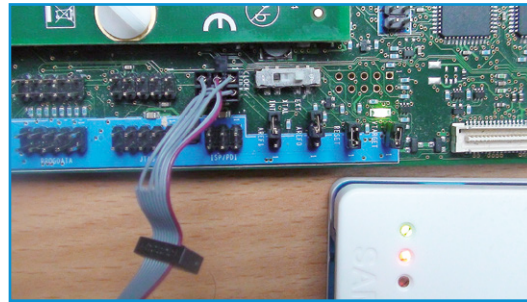


Bild 9.
Wenn man die Device Signatur auslesen kann, ist man auf der sicheren Seite.

Bild 10.
Xmega programmieren über PDI.

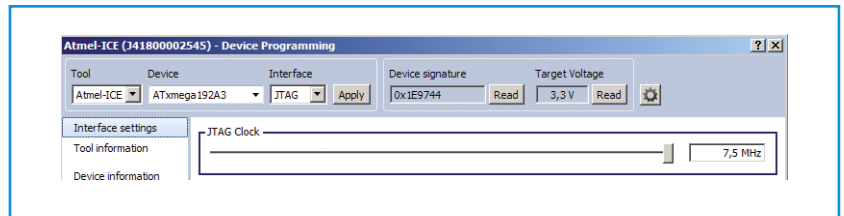


Bild 11.
Auch über JTAG lässt sich der Xmega programmieren.

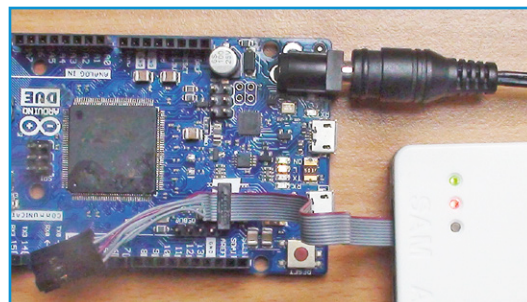


Bild 12.
Arduino Due am Atmel-ICE.

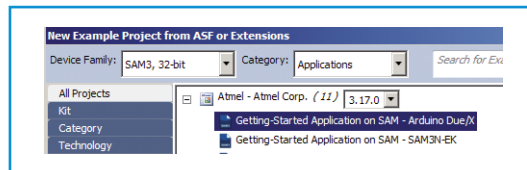


Bild 13.
Beispiel-Applikation wählen.

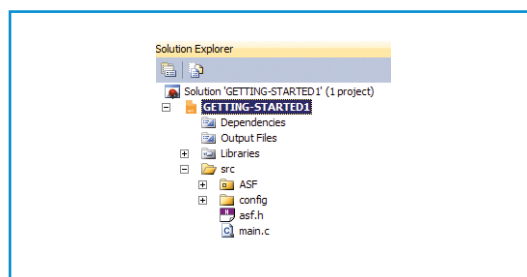


Bild 14.
Files der Anwendung im Solution Explorer.

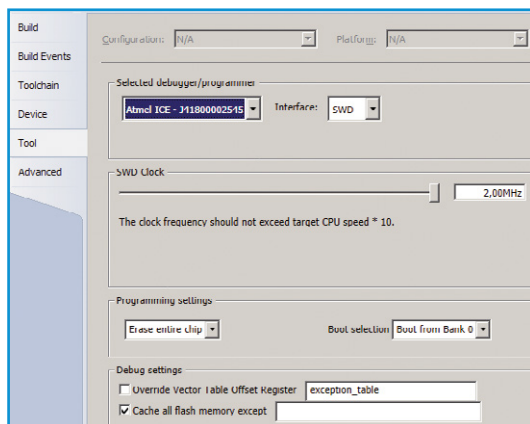


Bild 15.
Im Eigenschafts-Menü der Anwendung muss man unter „Tool“ den Debugger und das Interface auswählen.

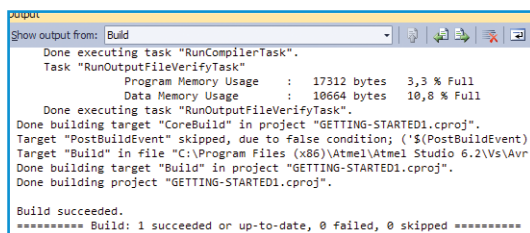


Bild 16.
Output-Fenster.

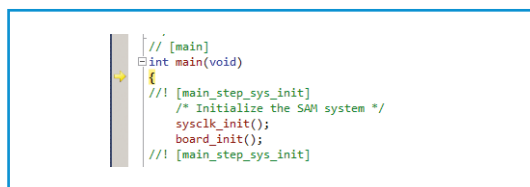


Bild 17.
Mit ALT-F5 hält das Programm in der ersten Zeile an.

Ein Klick mit der rechten Maustaste auf den Projektnamen „GETTING-STARTED1“ oder im Hauptmenü „Project“ → „GETTING-STARTED1 Properties“.. (ALT-F7) öffnet die Eigenschaftenseite des Projekts.

In der Karte „Device“ kann der Prozessor geändert werden. In der Registerkarte „Tool“ müssen wir den Debugger (Atmel ICE) und das Interface (SWD) auswählen (**Bild 15**). Der Haken bei „Cache all flash memory except“ muss nur dann entfernt werden, wenn zur Laufzeit des Programms der Flashspeicher verändert wird (z.B. durch ein Bootloader-Programm) und man dies beobachten möchte. Andernfalls sieht man im Memory-Fenster immer nur den Zustand des Flashspeichers zum Zeitpunkt des Programmstarts.

Alle anderen Einstellungen, auch in den übrigen Fenstern, brauchen erst einmal nicht geändert zu werden. Die Einstellungen sollten gespeichert werden, indem man im Hauptmenü „File“ → „Save Selected Items (Ctrl+S)“ oder „File“ → „Save All“

(Ctrl+Shift+S) ausführt.

Mit „Build“ → „Solution“ (oder F7) wird das Programm kompiliert. Da wir noch nichts geändert haben, sollte es fehlerfrei sein, was das Output-Fenster bestätigt (**Bild 16**). Nun ist es soweit. Mit ALT+F5 wird die ELF-Datei, die man sich im Solution-Explorer unter „output“ ansehen kann, in den Prozessor geladen und das Programm bleibt auf der ersten Programmzeile in „main“ stehen (**Bild 17**). Dagegen wird mit F5 das Programm nach dem Hochladen sofort gestartet: Eine LED blinkt bis zum Abbruch des Programms vor sich hin.

Im Hauptmenü sind unter „Debug“ alle Debug-Funktionen aufgeführt. Hier kann man zum nächsten Breakpoint springen und dergleichen mehr.

Ein wichtiger Tipp zum Schluss: Wenn das Ganze mal gar nicht funktionieren will, hilft oft ein komplettes „Erase Chip“ (Registerkarte „Memories“ im Programmer-Fenster). Vor allem wenn vorher ein anderes Programm im Prozessor geladen war.

Fazit

Mit dem Atmel-ICE bekommt man zu einem moderaten Preis gleichzeitig die Funktionen eines AVR- und SAM-Debuggers. Allerdings muss man gegenüber den etablierten Debuggern Abstriche bei der Geschwindigkeit machen. Bei kleiner bis mittlerer Projektgröße fällt die geringere Geschwindigkeit kaum ins Gewicht. Doch bei größeren Dateien und dauerhafter Softwareentwicklung weiß man die kürzere Programmierzeit eines AVROne! beziehungsweise eines SAM-ICE zu schätzen.

(140275)

Weblinks

- [1] www.ineltek.de
- [2] www.atmel.com/webdoc/atmelice/index.html
- [3] www.atmel.com/tools/atmelstudio.aspx

Learn to program

Spielerisch programmieren lernen – ohne Tastatur

NEU!

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



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

Eigenschaften des Boards:

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

Inhalt der Lernbox:

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

€ 44,95
CHF 56,95

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

Android Open Accessory

Elektronik steuern mit Android und Arduino



NEU!

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

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

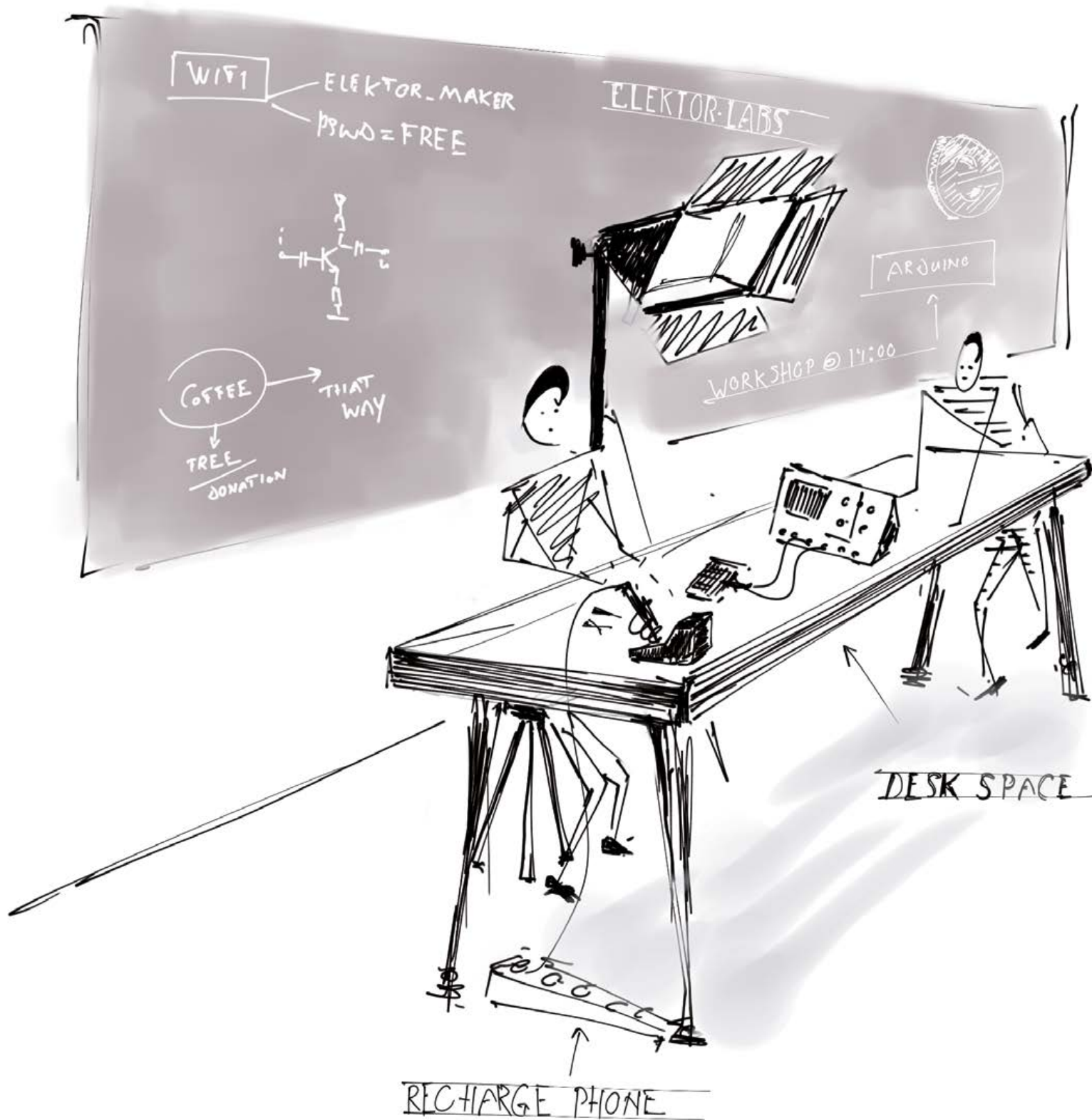
Die Programmbeispiele aus diesem Buch stehen auf der Elektor-Website zum Gratis-Download bereit.

380 Seiten (kart.) • Format 17 x 23,5 cm
ISBN 978-3-89576-279-6

€ 42,00
CHF 52,95

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

ELEKTOR-MAKER-S

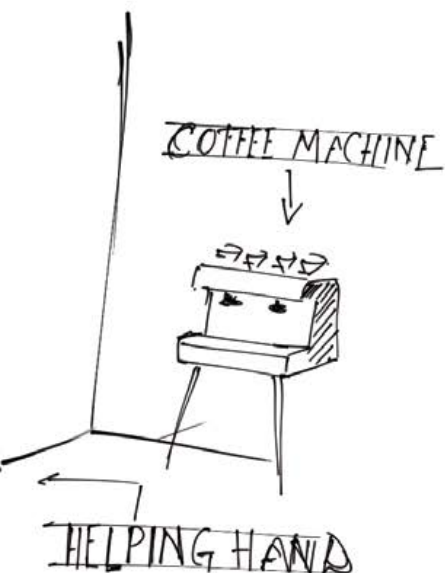


SPACE

Elektor Maker Space

@electronica 2014, München

Von **Wisse Hettinga** (Elektor-Labor)



Bestimmt ist Ihnen schon mal der Begriff „Maker“ zu Ohren gekommen und auch die damit verbundene Revolution sagt Ihnen was. Von tollen Sachen zu lesen ist eine Sache, doch so etwas auf der weltgrößten Elektronikmesse selbst zu probieren ist noch viel cooler! Man kann dort relaxen, das Smartphone aufladen, fachsimpeln, kaffeetrinken und richtig mit Elektronik in Kontakt sein. Man kann alles mitbringen, was man auspacken, (ent)löten, zeigen, messen, austesten oder sogar verarduinoen und raspberrieren möchte. Fühlen Sie sich eingeladen, Dinge ganz praktisch zu elektorisieren!

Die Entwickler und Ingenieure von Elektor stehen bereit, um Sie bei elektronischen Problemen zu unterstützen und sperrige Elektronik vor Ort zum Laufen zu bringen. Das Ganze in deutsch, englisch, niederländisch, spanisch und C++ ;-).

Für dieses spezielle Event stellt das Elektor-Labor zur Verfügung: Arbeitsplatz, Werkzeug, Messgeräte, 3D-Drucker und kostenloses WLAN. Nicht zu verachten: Mini-Workshops, Techtalk, Kaffee und jede Menge Steckdosen zum Aufladen von Handys, Tablets, Laptops und anderen Apparaten.

Navigation zum Elektor Maker Space: Halle A6, Stand 380

Lassen Sie sich von Elektor auf den neuesten Stand bringen – registrieren Sie sich für unseren Newsletter rechts unten auf www.elektor.de.

WER:

SIE UND ELEKTOR INTERNATIONAL MEDIA

WAS:

ELEKTOR MAKER SPACE

WANN:

11.-14. NOVEMBER 2014

WO:

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



Das Magnetron

Seltsame Bauteile (8)

Von **Neil Gruending**
(Kanada)

Da ich mich bislang hauptsächlich über Halbleiter ausgelassen habe, wird es Zeit, etwas Abwechslung ins Spiel zu bringen: In dieser Folge geht es daher um das Magnetron. Eigentlich gehen Sie häufig damit um. In der Küche. In jeder „Mikrowelle“ steckt nämlich ein Magnetron. Man nutzt dieses auch zum Trocknen und Erwärmen in der Halbleiter-Industrie und darüber hinaus. Bleibt noch zu klären, wie ein Magnetron funktioniert.

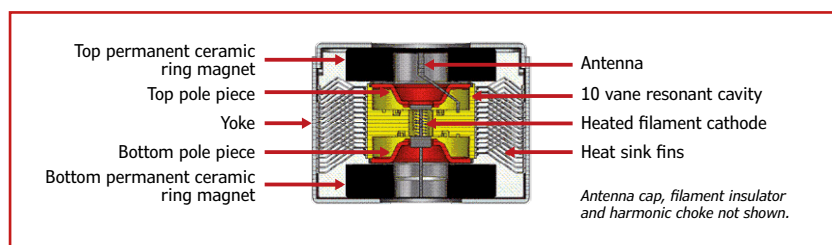
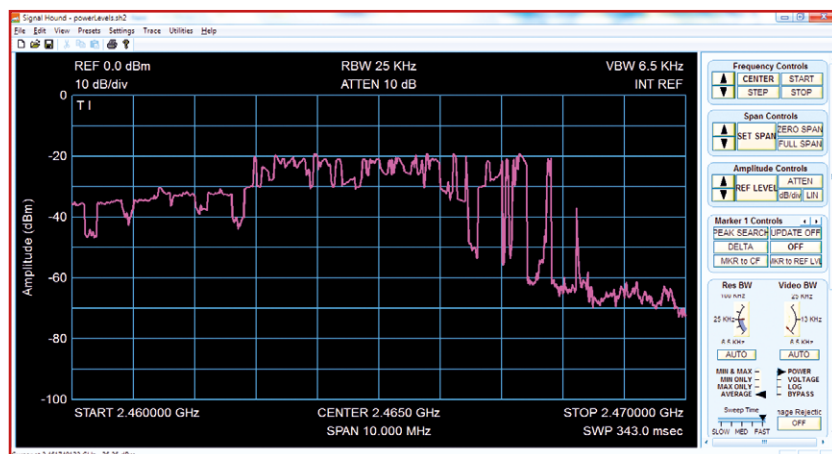


Bild 1.
Interner Aufbau eines
Magnetrons [1].

Bild 1 zeigt den inneren Aufbau. Durch die Heizung der Kathode in der Mitte gelangen Elektronen in den Hohlraum. Normalerweise würden diese direkt zur Anode fließen und das Resultat wäre „nur“ eine Röhrendiode. Doch die beiden Permanentmagneten oben und unten zwingen die Elektronen auf Kreisbahnen und beschleunigen sie, bis sie die Antennen-Anode erreichen und das Magnetron verlassen. Es gibt an den Enden des Magnetrons je einen Hohlraum, dessen Abstimmung die Frequenz der Wellen bestimmt, die von der Antenne abgestrahlt werden.

Bild 2.
Mikrowellenspektrum eines
Magnetrons.



Eine der ersten Anwendungen für Magnetrons war der Einsatz als Strahlungsquelle für Radarsysteme. Magnetrons sind nach wie vor sehr effektive Generatoren für Strahlungsimpulse hoher Leistung, wie sie für Radar benötigt werden. Eines der dabei auftretenden Probleme ist allerdings, dass die Frequenz von der Temperatur und der Ausgangsimpedanz beeinflusst wird. Außerdem ist die Strahlung eines Magnetrons nicht sehr schmalbandig, weswegen ein Radarempfänger nicht sehr frequenzselektiv sein darf, was seine Empfindlichkeit beeinträchtigt. Doch Ausgangsleistung und Wirkungsgrad gleichen die Nachteile mehr als aus. Kommerzielle Magnetrons benötigen zum Betrieb Hochspannung im Bereich 4...10 kV. In der häuslichen Mikrowelle stecken kleinere Exemplare mit Spannungen im Bereich 1...2 kV, die noch luftgekühlt sein können (daher das hörbare Gebläse). Industrielle Magnetrons in der Leistungsklasse ab 3 kW benötigen hingegen eine Wasserkühlung. Ausrangierte Mikrowellenöfen sind eine preiswerte Quelle, um Magnetrons für eigene Experimente zu erhalten. An dieser Stelle ist eine Warnung mehr als angebracht: Die Mikrowellenstrahlung ist ernsthaft gefährlich! Auch die Versorgungsspannung mit ≥ 1.000 V kann tödliche Folgen haben. Das von mir ausgebaute Exemplar wurde zunächst vermessen: In **Bild 2** sieht man die Frequenzdrift über die Zeit. Die fiel heftiger aus, als ich angenommen hatte. Die Frequenz änderte sich so schnell, dass der Spectrum-Analyzer mit der Echtzeiterfassung überfordert war. Von daher nutzte ich das Peak-Detection-Verfahren zur Ermittlung der Frequenzvariation. Bei einem Test legte das Magnetron mit 2,465 GHz los und driftete dann für 30 s. Die Linie bei -20 dB zeigt die Frequenzabweichung. Die Kurve wäre bei einem längeren Test sicher noch viel flacher ausgefallen. Bei Experimenten sollte man auch die HF-Leistung gut abführen können. Ich selbst verwendete einen versiegelten Wellenleiter mit einer wassergekühlten Last. Nochmals: Wenn man mit einem Magnetron herumspielt, ist Sicherheit das Allerwichtigste! (140241)

[1] www.cst.com/Applications/Article/Magnetron-And-Microwave-Oven-Design-To-Solve-Wi-Fi-Interference-Issues

Professionelle Hard- & Software zum Sonderpreis!

Exklusiv für Studenten!

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

LabVIEW

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



Circuit Design Suite

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



myDAQ

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



Studentenversionen:

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

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



DesignSpark Tipps & Tricks

Leiterbahnen ziehen sich von Geisterhand

Von **Neil Gruending**
(Kanada)

In dieser Folge gibt es keine Handarbeit, vielmehr kann man dem Autorouter beim Arbeiten zusehen.

Wenn Sie noch nie mit einem Autorouter gearbeitet haben, dann bekommen Sie heute einen in die Finger. Nach einer Autorouter-Einführung geht es weiter mit dessen Konfiguration und seinem praktischen Einsatz.

Was genau ist ein Autorouter?

Bei der Platinenentwicklung kann ein Autorouter alle oder ein Teil der Leiterbahnen nach vordefinierten Regeln verlegen. Wie die meisten Autorouter nutzt auch der von DesignSpark eine formabhängige Multipass-Technik, was die Wahrscheinlichkeit erhöht, dass er die komplette Platine routen kann. Netzbasierte Autorouter können schon mit einer Leiterbahn ziemlich viel blockieren und daher beim Routen komplexer Platinen zu argen Schwierigkeiten führen. Ersteres Verfahren hingegen berücksichtigt die tatsächlichen Formen der Leiterbahnen und versucht, diese so dicht wie möglich zu verlegen. Das Multipass-Verfahren bedeutet, dass der Autorouter das Verlegen solange ändert und ausprobiert, bis alle Routing-Fehler beseitigt sind oder die Anzahl zuvor festgelegter Durchgänge überschritten wird. Den Autorouter von DesignSpark konfiguriert man mit dem Dialogfenster „Route All Nets“ (siehe

Bild 1), das man im Menü via *Tools* → *Auto Route Nets* → *All Nets* erreicht. Autorouter suchen den einfachsten und ökonomischsten Pfad für ein Netz. Man kann über weitere Regeln beeinflussen, wie diese Pfade berechnet werden. Ganz wie bei den Tools zur automatischen Bauteileplatzierung kann man den Autorouter auf verschiedene Art und Weise verwenden, aber die Konfiguration erfolgt in diesem einen Dialog.

Eine der wichtigsten Regeln betrifft das Verhalten bezüglich vorhandener Leiterbahnen beim Start des Autorouters. Als Standard werden sie komplett entfernt, bevor ein neues Routing beginnt. Wenn man das nicht will, muss man die Regel „Keep Preroutes“ und „Keep Fixed Routes“ anwenden. Eine „Fixed Route“ ist ein Abschnitt einer Leiterbahn, die in ihren Eigenschaften als „Fixed“ für den Autorouter konfiguriert wird. DesignSpark markiert die von Hand verlegten Leiterbahnen automatisch als „Fixed“. Ein fixierter Leiterbahnabschnitt wird vom Autorouter auf keinen Fall angetastet. Eine „Preroute“ ist eine Leiterbahn, die vom Autorouter beim Start zwar in Ruhe gelassen wird, doch bei den weiteren Durchläufen darf er sie verändern. Dies aktiviert man mit der Option „Keep Preroutes“.

Eine weitere wichtige Regel ist „Miter Track“. Als Standard werden die Leiterbahnen mit rechten Winkeln (90°) verlegt. Mit dieser Regel wird der Autorouter angewiesen, nach dem Ende des Routings so viele rechte Winkel wie möglich aufzulösen.

Der Zeitbedarf und der Aufwand, den der Autorouter zum Routing benötigt, wird über die Regeln „Max Effort“ und „Passes“ festgelegt. Der Autorouter bemüht sich intensiver und länger, wenn man den Wert „Max Effort“ erhöht. Aber normalerweise wird man an dieser Schraube nicht viel drehen, da ein Autorouter nach Aufgabe bei einem nächsten Durchgang in der Regel bessere Resultate bringt. In „Passes“ legt man fest, wie viele Durchgänge der Autorouter maximal durchführen soll. Bei Werten >1 routet der Autorouter

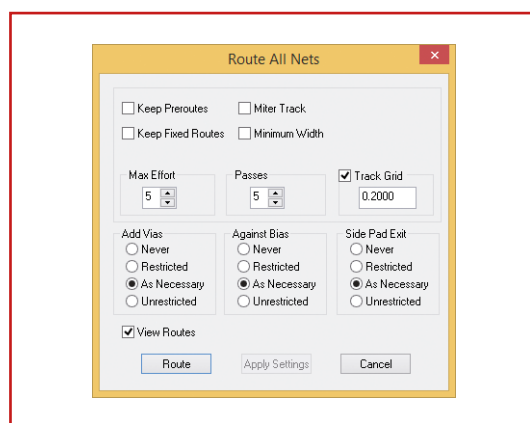


Bild 1.
Der Dialog „Route All Nets“.

so, dass er Fehler wie Überlappungen zunächst akzeptiert und dann versucht, diese in weiteren Durchgängen aufzulösen.

Mit der Regel „Against Bias“ entscheidet man, ob die Bias-Regeln für diese Platine gelten sollen. Man kann dies für jede Kupferlage für X oder Y festlegen oder ganz abschalten. Ein X-Bias instruiert den Autorouter, so viele Bahnen wie möglich horizontal zu führen. Beim Y-Bias geht es um die vertikale Orientierung. Normalerweise verwendet man keinen Bias, da dies dem Autorouter die meiste Flexibilität erlaubt.

Meiner Ansicht nach empfiehlt sich das Aktivieren von „Track Grid“ samt Festlegung auf das beim Platzieren der Bauteile verwendete Raster. Der Autorouter kommt auch ganz gut ohne Raster aus, aber spätere Änderungen von Hand gelingen viel leichter, wenn sich die erzeugten Leiterbahnen ans Raster halten. Empfehlenswert ist es auch, „Add Vias“ und „Side Pad Exit“ auf „As Necessary“ zu setzen.

Verwendung des Autorouters

Wie schon erwähnt, kann man den Autorouter auf verschiedene Art und Weise einsetzen. Der Befehl zum Routen einer kompletten Platine ist *Tools → Auto Route Nets → All Nets*. **Bild 2** zeigt die Platine des LED-Treibers aus der letzten Folge dieser Reihe. Sie wollte unbedingt automatisch geroutet werden. Hinterher muss man zwar etwas nachbessern, doch das ist beim automatischen Routing normal. Doch woher weiß der Autorouter, welche Bahnen er wie routen soll?

Als Voreinstellung nutzt er seine eigenen Werte für Vias (Durchkontaktierungen) und Leiterbahnen. Man kann diese aber durch eigene Netzklassen überstimmen (siehe **Bild 3**). Eine so genannte „net class“ bestimmt, wie DesignSpark Netze in Gruppen organisiert, sodass festgelegte Einstellungen für alle Netze dieser Gruppe gelten. Als Standard wird eine Klasse namens „Signal“ angelegt, in der alle Netze enthalten sind. Selbstverständlich kann man aber so viele Klassen wie gewünscht anlegen. Netz-Klassen editiert und legt man unter dem Tab „Net Classes“ an. Die einzelnen Netze assoziiert man via Tab „Nets“ mit ihrer Klasse. In diesem Fall habe ich die Netzklasse dazu verdonnert, den Bahntyp „Signal“ und den Via-Typ „SignalVia“ zu verwenden.

DesignSpark bietet auch noch andere Modi: Das

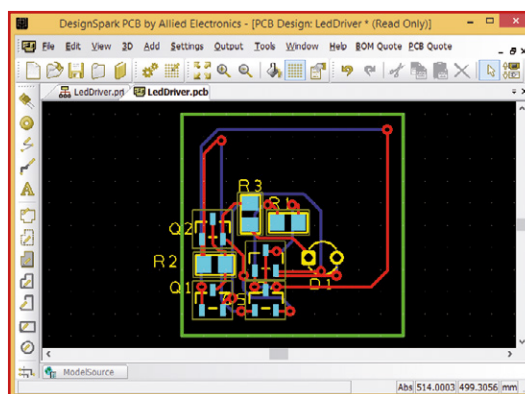


Bild 2.
Geroutete LED-Treiber-Platine.

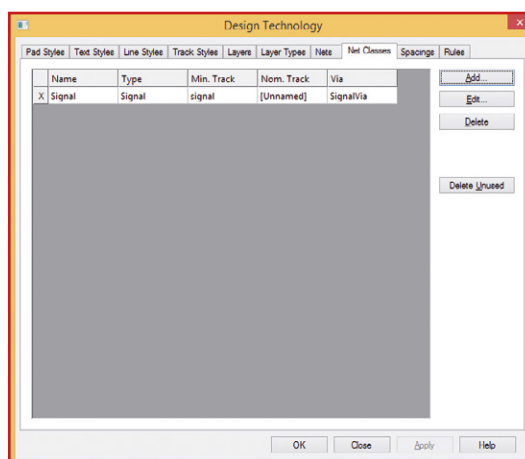


Bild 3.
Der Netz-Klassen-Dialog.

Routen individueller Netze oder Netzklassen. Eine der wichtigsten Fähigkeiten eines Autorouters ist meiner Ansicht nach, die Verbindungen der Bauteile mit der Stromversorgung kurz zu halten und nahe gelegene Vias zu verwenden, wenn die Platine über Stromversorgungs-Layer verfügt. Wenn man das den Autorouter erledigen lässt, spart man schon eine Menge Zeit. Auch die interaktive Verwendung des Autorouters ist möglich. Hierzu selektiert man per Rechtsklick Netze oder Bauteile und wählt dann „Auto Route“ aus. Dann werden nur die hierzu nötigen Leiterbahnen automatisch verlegt.

Fazit

In dieser Folge ging es um den Autorouter von DesignSpark. Dieses Tool kann sehr viel Zeit sparen. Normalerweise route ich Platinen von Hand, doch der Autorouter von DesignSpark ist so flexibel, dass er mir ein ganzes Stück Arbeit abnimmt. In der nächsten Folge geht es dann um weitere Fähigkeiten von DesignSpark.

(140242)

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



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

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

UMWELTSCHONEND – GÜNSTIG – GREEN

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



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

RTC-Module von Micro Crystal

Klein, komplett und genau

Nicht jede mit Uhrzeit operierende Elektronik kann die aktuelle Zeit via Internet von einem Time-Server oder per Funk von einem Zeitzeichensender beziehen. Und selbst wenn, dann besteht immer noch der Bedarf, die Zeit auch ohne Internetverbindung oder ohne Empfang vorzuhalten. Hier für gibt es spezielle ICs unter der Bezeichnung RTC bzw. Echtzeituhr. Ein RTC-Klassiker wie der Typ DS1302 benötigt aber einen Extra-Quarz und hat kein standardisiertes Interface. Moderne RTCs hingegen sind fix und fertige Lösungen und haben alles Notwendige schon an Bord.

Die aktuelle RTC-Familie des Herstellers „Micro Crystal“ bedient mit energiesparenden, sehr genauen oder besonders kleinen Modulen die unterschiedlichsten Ansprüche. Hervorzuheben ist, dass der typische 32.768-Hz-Uhrenquarz immer schon integriert ist. Und da es sich (ungewöhnlich für Halbleiter-Bauteile) um einen Hersteller aus der Schweiz handelt, darf man zu Recht einige Expertise auf dem Gebiet der Zeitmessung erwarten. Micro Crystal bietet nicht nur diverse Quarze und RTCs an, sondern auch passende Evaluation-Boards.

Wie man am Beispiel von **Bild 1** sehen kann, eignen sie sich wegen der auf Stiftleisten herausgeführten Signale gut für Experimente und Messungen.

Die RTC-Familie

Micro Crystal stellt RTC-Module mit I²C- oder SPI-Interface mit hohem Übertragungstakt her, wobei die Anzahl an Typen mit I²C-Interface überwiegt. Bei diesen Typen sind dank weniger Datenleitungen auch weniger Pins erforderlich. Die RTC-Module werden in drei verschiedenen Gehäusen geliefert (**Bild 2**): Das C2-Gehäuse ist mit 5,0 x 3,2 x 1,2 mm das größte und hat ein Fenster, durch das man den Quarz sehen kann. Das mittlere Format ist mit 3,7 x 2,5 x 0,9 mm das C3-Gehäuse. Bei den Modulen im nur 3,2 x 1,5 x 0,8 mm messenden C7-Gehäuse handelt es sich um die zur Zeit kleinsten RTC-Module der Welt. Manche RTC-Typen werden in unterschiedlichen Gehäusen angeboten, aber nicht alle Typen in allen Gehäusen.

Prinzipiell sind alle Module ähnlich aufgebaut, verfügen aber je nach Typ über unterschiedliche Eigenschaften, andere Register oder abweichende Peripherie. Die Typen unterscheiden sich unter

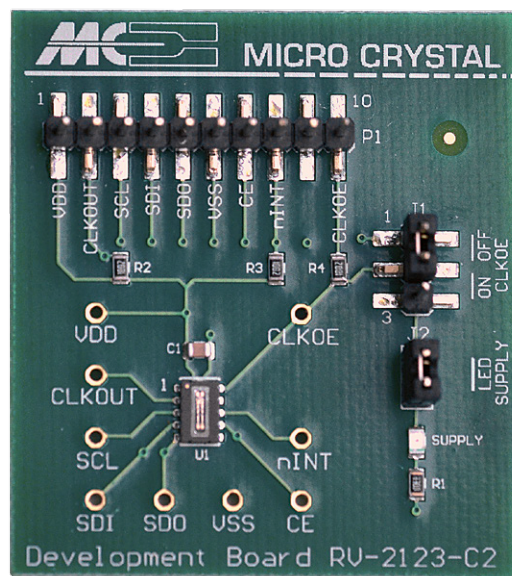


Bild 1.
Ein typisches
Evaluation-Board
(Bild: Richard Endres).

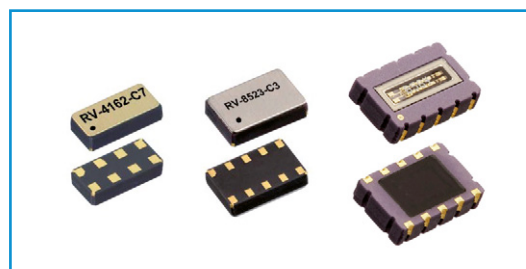


Bild 2.
Größenvergleich der
verschiedenen RTC-Gehäuse.
Von links nach rechts: C7, C3
und C2.

anderem in den Punkten Stromverbrauch und Genauigkeit. **Tabelle 1** gibt eine Übersicht über diese Unterschiede. In der Spalte „Spannung“ bezieht sich die Minimalspannung auf den Wert, bei dem reines „Time Keeping“ noch funktioniert. Das Interface oder der eingebaute Temperatursensor stehen erst bei höheren Spannungen zur Verfügung. Spezialfunktionen bezüglich der Zeitgenauigkeit wurden nicht berücksichtigt.

Das Innenleben

Die internen Funktionseinheiten der RTC-Module werden nachfolgend am Beispiel des Typs RV-3029 erläutert, der über ein I²C-Interface verfügt. In der Blockschaltung von **Bild 3** erkennt man den schon erwähnten Temperatursensor. Dieser dient weniger zur Erfassung der Umgebungstemperatur, sondern ist zur Temperaturkompensation der Quarzfrequenz erforderlich. Hierfür wurden vom Hersteller in speziellen Registern an den jeweiligen Quarz angepasste Korrekturinformationen abgelegt. Diese können zwar vom Anwender überschrieben werden, doch sollte man hierzu genau wissen, was man tut.

Links oben befindet sich der Quarzoszillator, auf den der korrigierende Teiler folgt, der direkt das Sekunden-Register inkrementiert. Alle Registerwerte sind im BCD-Format gehalten. Im Stunden-Register gibt es noch ein Bit für den 12/24-h-Modus. Der Block *Output Control* liefert abhängig von CLKOE einen Takt am Ausgang CLKOUT. Die ausgegebene Frequenz reicht von 1...32.768 Hz und wird durch die Bits FD1 und FD0 im EEPROM-Control-Register festgelegt. Am /INT-Pin erscheint bei einem Interrupt eine logische „0“. Ein Interrupt kann vom Timer, dem

Alarm, der *Power Control* oder dem „Self-Recovery System“ ausgelöst werden. Letzteres passiert, wenn sich die interne State Machine aufhängen sollte. Es gibt zwei Timer-Register, jeweils eines für das LSB und das MSB. Es handelt sich dabei um einen 16-bit-Abwärts-Zähler, dessen Takt in den Control-Registern festgelegt wird. Hier kann auch angegeben werden, ob der Timer nach dem (möglichen) Interrupt beim Endwert „0“ automatisch wieder gefüllt wird. Die meisten Interrupts müssen in den Control-Registern freigegeben werden. Alternativ gibt es auch Flags, die anzeigen, ob das Ereignis schon stattgefunden hat. Diese müssen nach jedem Ereignis wieder zurückgesetzt werden, selbst wenn die Interrupts nicht aktiviert sind. Dazu kommen Flags, die anzeigen, welches Ereignis ein Interrupt ausgelöst hat. Diese müssen nach einem Interrupt wieder zurückgesetzt werden, damit auch der INT-Pin wieder auf eine logische „1“ resettet wird.

Beim Block *Power Control* handelt es sich um ein ausgeklügeltes System, welches bestimmte Spannungspegel erkennt und abhängig davon z.B. den internen Widerstand zwischen V_{DD} und V_{BACKUP} schaltet. Dadurch wird ein optional an V_{BACKUP} angeschlossener Gold Cap wieder aufgeladen. Wenn die Versorgungsspannung unter 2,1 V (V_{LOW1}) fällt, werden CLKOUT, der Temperatursensor und die Schreibmöglichkeit in die Register deaktiviert. Das I²C-Interface bleibt zwar aktiv, aber mit reduzierter Geschwindigkeit.

Booten und Betrieb

Nach dem Anlegen der Versorgungsspannung und dem folgenden Power-on-Reset sind die meisten Funktionen wie der Alarm oder der Timer inak-

Tabelle 1. Daten der verschiedenen RTC-Module.

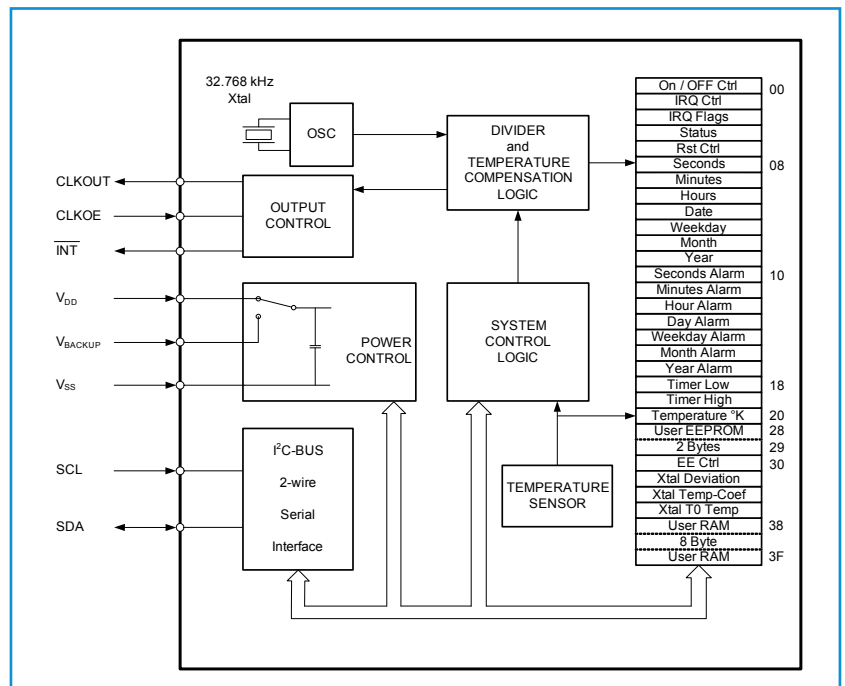
Typ-Gehäuse	Genauigkeit (25°C)	Interface	Spannung	Strom	Merkmale
RV4162-C7	±20 ppm	I ² C	1,0...4,4 V	350 nA	Submin.-Gehäuse
RV8523-C3	±20 ppm	I ² C	1,2...5,5 V	130 nA	Stromsparend
RV8564-C2/C3	±20 ppm	I ² C	1,2...5,5 V	250 nA	Standard
RV3029-C2/C3	±3 ppm	I ² C	1,3...5,5 V	800 nA	Sehr genau
RV1805-C3	±20 ppm	I ² C	1,5...3,6 V	50 nA	Sehr stromsparend
RV8803-C7	±10 ppm	I ² C	1,5...5,5 V	250 nA	Submin.-Gehäuse
RV2123-C2/C3	±20 ppm	SPI	1,1...5,5 V	130 nA	Stromsparend
RV049-C2/C3	±3 ppm	SPI	1,3...5,5 V	800 nA	Sehr genau

tiv. Der Hersteller empfiehlt, nach jedem Power-on-Reset auch einen System-Reset (mit Hilfe des SysR-Bits im Control-Reset-Register) durchzuführen. Danach sollte man via Control-Register festlegen, welche Ereignisse einen Interrupt auslösen können und welche Peripherie aktiv sein soll. Wichtig ist, dass ein 10-nF-Siebkondensator nahe am IC an die Leitungen zur Versorgungsspannung angeschlossen sein soll. Im sehr ausführlichen Application-Manual [1] findet man detaillierte Informationen zu Einsatz und Betrieb. (130580)

Weblink

[1] www.microcrystal.com/index.php/products/real-time-clocks

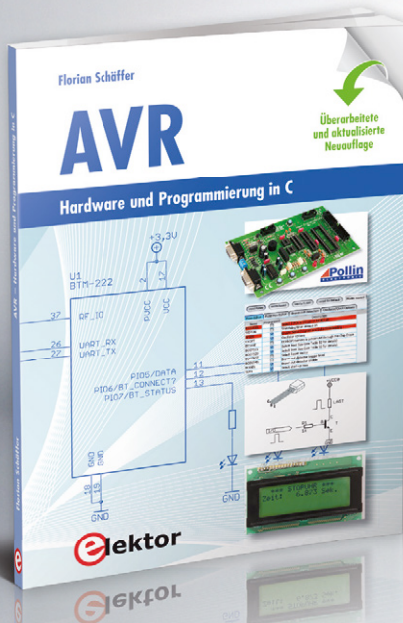
Bild 3. Das Blockschaltbild des RTC-Moduls RV-3029 (Quelle: Datenblatt).



Anzeige

AVR Hardware und Programmierung in C

Überarbeitete und aktualisierte Neuauflage



Der Autor führt Einsteiger und auch Fortgeschrittene gekonnt und professionell in eine hochinteressante Thematik ein. Auch wer seine Elektronik- und Programmierkenntnisse weiter ausbauen und vertiefen möchte, hat dazu gute Möglichkeiten. Die modernen und zeitgemäßen Atmel AVR-Prozessoren sowie die Programmierung in C sind in Kombination eine zukunftssichere Plattform für lange Zeit. Nach Einführung und Vorstellung der notwendigen Entwicklungsumgebung werden Projekte vorgestellt, die schrittweise zum Ziel führen. Für die meisten Projekte kommt das Atmel AVR-Evaluation-Board zum Einsatz – eine Experimentierplatine aus dem Hause Pollin Electronic. Das gewährleistet den reibungslosen Nachbau der vorgeschlagenen Projekte. Natürlich ist auch die Verwendung eigener Experimentierschaltungen möglich, denn ein erklärtes Ziel des Buches ist es, den Anwender zu selbständigem Arbeiten und Entwickeln zu befähigen.

Tipp: Unter www.elektor.de/avr-buch haben wir ein vorteilhaftes Buch/Board-Bundle für Sie geschnürt.

260 Seiten • Format 17 x 23,5 cm (kart.)
ISBN 978-3-89576-300-7

€ 34,80
CHF 43,95

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

(Los)gelöst

Von
Wolfram Pioch
(D)

Autor Wolfram Pioch hatte für sein Review (siehe diese Ausgabe) einen Programmer Atmel ICE geliehen bekommen. Leider lief die Inbetriebnahme nicht ganz problemlos. Er schrieb uns:

Als ich das Micro-USB-Kabel zum zweiten Mal in die Buchse stecken wollte, ging dies verdächtig leicht. Viel zu leicht. Der Kabelstecker ging in Leere, die Buchse war nicht mehr da, wo sie sein sollte. Dann kam der Schreck! Ein geliehenes Gerät behandelt man ja besonders sorgfältig, und nun war es schon kaputt...

Sch...! Aber wenn die Platine sowieso hinüber ist, kann ich auch einen Reparaturversuch wagen, dachte ich mir.

Also den Schraubenzieher genommen und den ICE aufgemacht. Nach dem Ausbau der Platine (**Bild 1**) zeigte sich der Schaden: Buchse natürlich ab, äußere Massepads auch ab, ein Leiterzug (der zweite von rechts) nicht mehr da. Auf den Massepads unter der Buchse befand sich kein Lötzinn. Dort war nur der Klebstoff von der Bestückung drauf. Jedenfalls war die Befestigung der Buchse nicht stark genug, um den Steckvorgang auszuhalten. Wären die übrig gebliebenen Leiterzüge richtig angelötet gewesen, hätten sie eigentlich auch abgerissen sein müssen...

Also wie nun reparieren? Die Chance war nicht sehr groß, die Buchse korrekt wieder anzulöten. Ganz

zu schweigen davon, dass sie auch noch halten müsste. Blieb noch die alte Methode, dünne

Kupferdrähte von 3-4 cm Länge an die Pins der Buchse zu löten und diese dann mit der Platine zu verbinden.

Es dauerte schon eine ganze Weile, bis das Werk fertig war. Die Abstände waren so gering, dass man selbst mit der dünnsten LötKolbenspitze schräg arbeiten musste. Ansonsten lötete sich der benachbarte Draht von der Wärme wieder ab. Mit den Drähten an der Buchse wurde dann ein sanfter Haltbarkeitstest ausgeführt. Zwei fielen gleich wieder ab. Dann begann die ganze Prozedur von vorn: alle fünf Drähte ablöten und von links nach rechts wieder anlöten. Als endlich kein Draht mehr abfiel, kamen die anderen Seiten der Drähte auf die Platine. Unter der Lupe wurde geguckt, ob kein Kurzschluss zu sehen war. Dann kam die Positionierung der Buchse in der ursprünglichen Lage. Dazu mussten die Drähte um 180° geknickt werden, wobei natürlich wieder einer abfiel. Schließlich konnte die Platine wieder in den Kunststoffrahmen des Gehäuses gebracht werden, wobei ... Sie ahnen es. Zum Schluss hatte ich dann schon Übung im Anlöten der Drähte und die Buchse konnte mit reichlich Heißkleber (mehr am Rahmen als auf der Platine) fixiert werden. Das Ganze sah dann mehr schlecht als recht aus (**Bild 2**). Ich habe bewusst darauf verzichtet, die Kupferdrähte ästhetisch auszurichten. Dabei wäre garantiert auch wieder einer abgefallen.

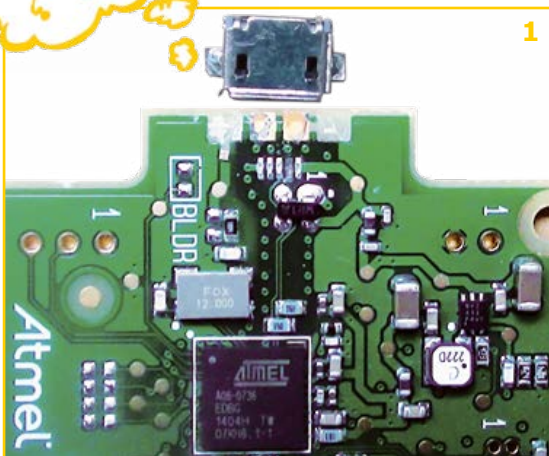
Nach dem (ganz vorsichtigen) Anstecken des USB-Kabels wurde dieses sicherheitshalber an einen USB-Hub statt direkt an den PC angeschlossen. Was geschah? Es glühte rot auf! Aber zum Glück nicht die Platine oder der Hub, sondern die rote Power-LED. Dann noch schnell das ICE-Kabel angeschlossen und ans Zielsystem gesteckt: Es lief als sei nichts geschehen.

Nach dem Einbau ins Gehäuse wurde das USB-Kabel noch mit Heißkleber am Gehäuse fixiert. Es kam noch ein Aufkleber drauf: „Abnehmen auf eigene Gefahr!“.

Wie lange das halten wird? Egal, es funktioniert jetzt schon länger als im Originalzustand.

Mir schwant, weshalb das Board des Emulators als „PCBA Kit“ einzeln erhältlich ist und in der Hilfe des Atmel Studio 6.2 erklärt wird, wie das Gehäuse geöffnet wird...

(140274)



Professional Quality
Trusted Service
Secure Ordering



Elektor PCB Service at a glance:

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

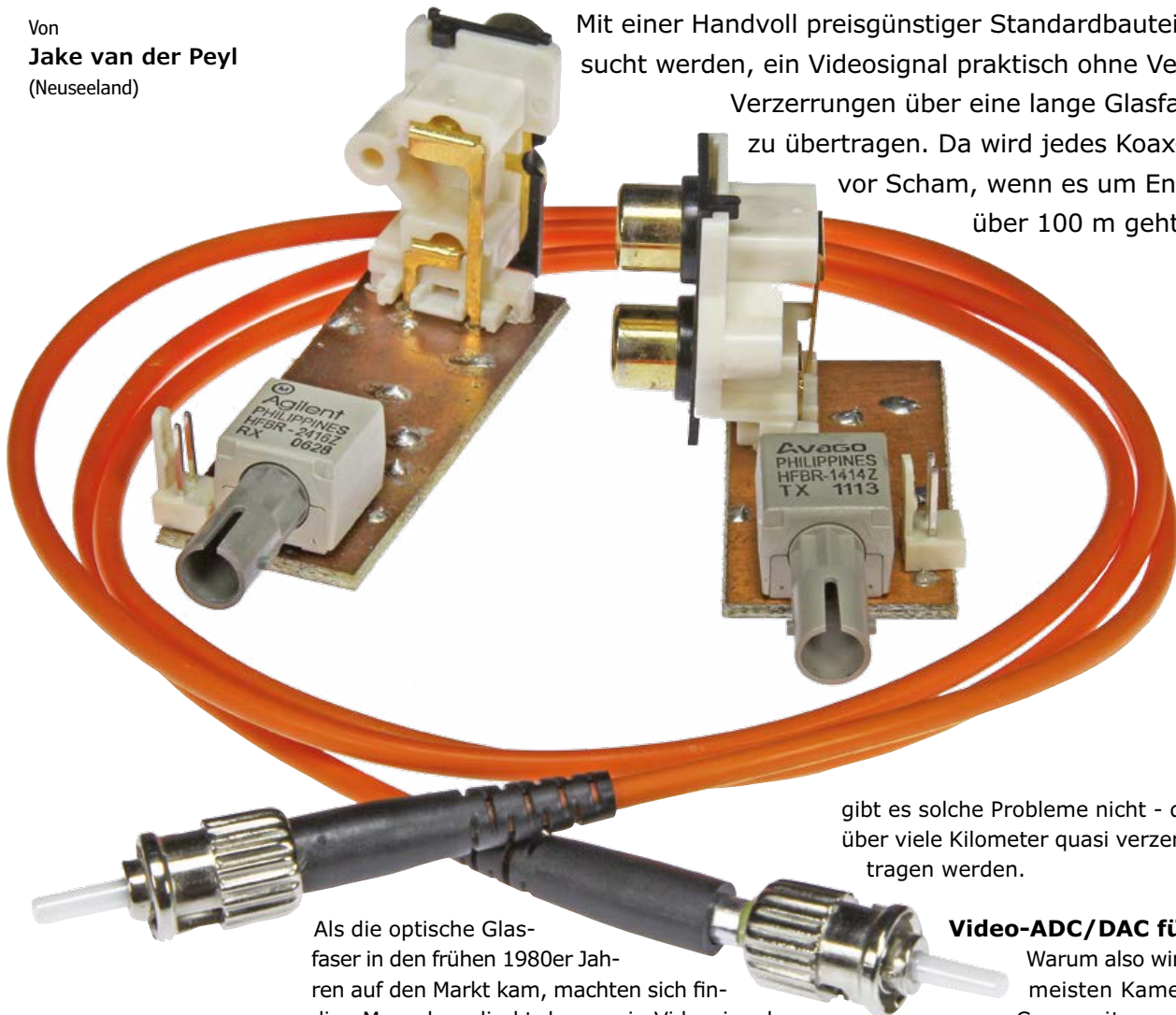
Delivery
from 2
working
days

Video über Glasfaser

Low-Cost-Lösung im Test

Von
Jake van der Peyl
(Neuseeland)

Mit einer Handvoll preisgünstiger Standardbauteile soll versucht werden, ein Videosignal praktisch ohne Verluste und Verzerrungen über eine lange Glasfaserstrecke zu übertragen. Da wird jedes Koax-Kabel rot vor Scham, wenn es um Entfernungen über 100 m geht.



gibt es solche Probleme nicht - das Signal kann über viele Kilometer quasi verzerrungsfrei übertragen werden.

Als die optische Glasfaser in den frühen 1980er Jahren auf den Markt kam, machten sich findige Menschen direkt daran, ein Videosignal über diesen neuen Link zu schicken. Wegen der stark nichtlinearen Eigenschaften der Übertragung durch Lichtwellenleiter musste das Signal aber zunächst digitalisiert werden.

Dünnes Kabel, dicke Vorteile

Das Senden eines Videosignals über ein Koax-Kabel ist recht kompliziert und endet in der Praxis bei Entfernungen von ungefähr 100 m. Und selbst unterhalb dieser Marke ist bei einem Composite-Videosignal eine deutlich schlechtere Farbwiedergabe auszumachen. Bei einem Glasfaserkabel

Video-ADC/DAC für Arme

Warum also wird Video bei den meisten Kameras über einen Composite- oder bestenfalls einen S-Video-Ausgang übertragen und nicht über ein Glasfaserkabel? Der Grund ist die Komplexität der Digitalisierung des Videosignals, die immer schnelle AD- und DA-Wandler erfordert. Diese Bauteile sind leider sehr teuer und außerdem noch sehr energiehungrig. Glücklicherweise gibt es einen einfachen und preisgünstigen Weg, um Video ohne AD-Wandler zu digitalisieren: PWM. Damit übersetzt man die Momentanspannung eines analogen Signals in Pulsbreiten. Die Impulsbreite ist eine endliche Zeit und kann in Sekunden oder Bruchteilen einer

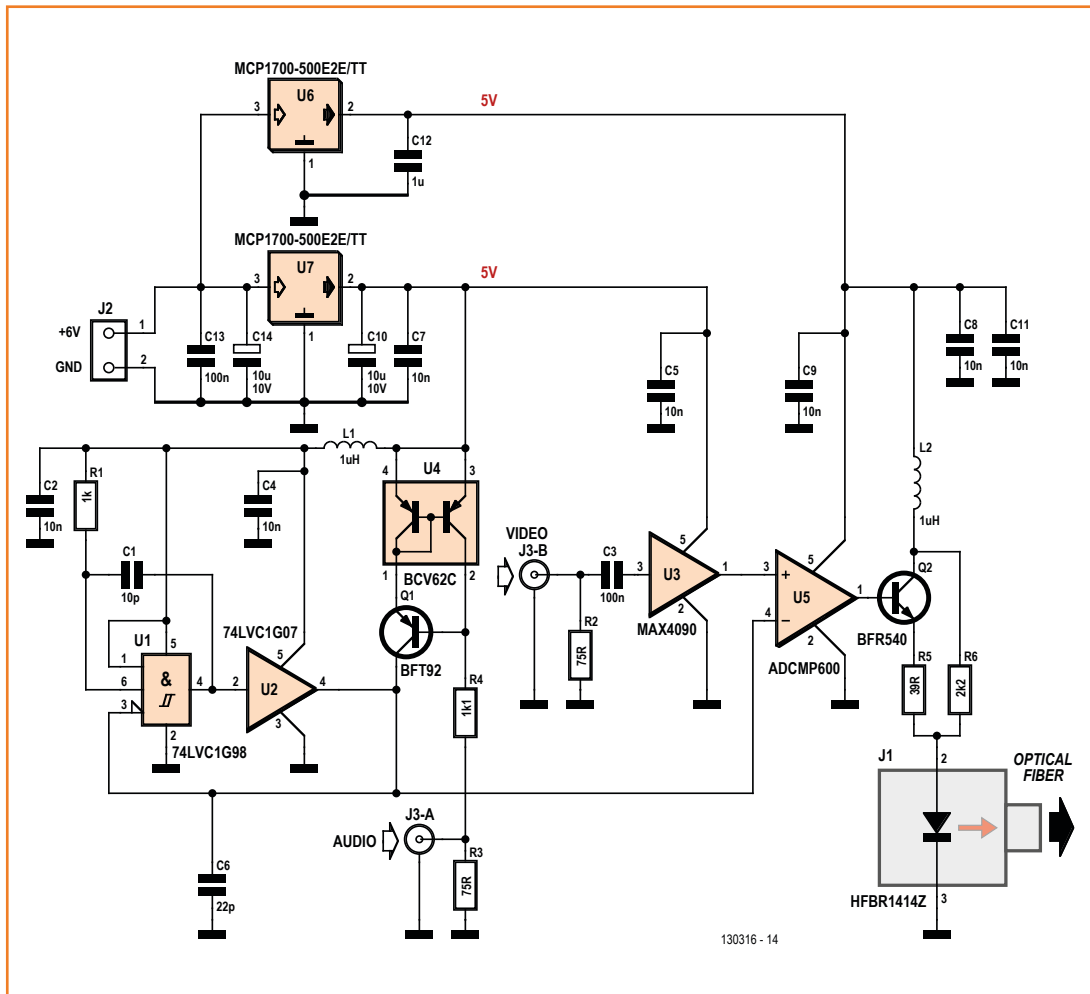


Bild 1.
Schaltung des Video-Senders
für Glasfaserkabel.

Sekunde ausgedrückt werden.

Das Nyquist-Abtasttheorem bestimmt, dass die Frequenz des PWM-Signals konstant und mindestens zweimal so hoch wie die maximale Bandbreite des zu konvertierenden Signals sein muss. Die Mindestbandbreite für Composite Video beträgt etwa 5,5 MHz, also würde die minimale PWM-Frequenz in der Größenordnung von 11 MHz liegen.

Wie sieht eine solch schnelle PWM-Schaltung aus? In der Theorie ganz einfach: Zuerst erzeugt man eine Dreiecksspannung, legt sie auf den Eingang eines Komparators und das Videosignal auf den anderen. Am Ausgang des Komparators erscheint dann das PWM-Signal. Voilà, das ist es!

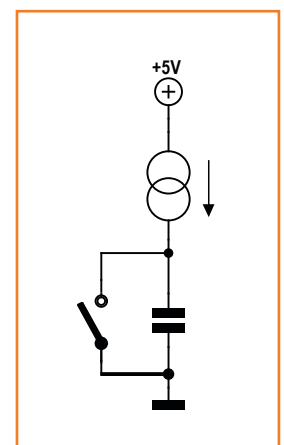
Der Sender

Die Schaltung des experimentellen Video-über-Glasfaser-Senders ist in **Bild 1** zu sehen. Einige der Konstruktionsprinzipien sol-

len weiter unten besprochen werden, zunächst aber zurück zur Dreiecksspannung. Natürlich muss sie sehr linear sein, und sich außerdem über den gesamten Spannungsbereich des Videosignals erstrecken. Ein Sonderfall der Dreiecksspannung ist die Sägezahnspannung, wie man sie in analogen Oszilloskopen verwendet. Ein Sägezahn ist viel einfacher zu produzieren als eine reine Dreiecksspannung. Die einfachste Form des Sägezahngenerators besteht aus einem Kondensator, einer Konstantstromquelle und einem Schalter - wie in **Bild 2**.

Aus praktischen Gründen wollen wir unsere Versorgungsspannung auf +5 V beschränken. Es ist auch einfacher, das Signal auf eine höhere Frequenz als 11 MHz zu verschieben, da es die Anforderungen an die Filter im Empfänger senkt. Außerdem dürfte diese Maßnahme auch die Qualität des Signals verbessern. Einigen wir uns auf 20 MHz (Periodenzeit 50 ns).

Bild 2.
Wir erzeugen einen
Sägezahn.



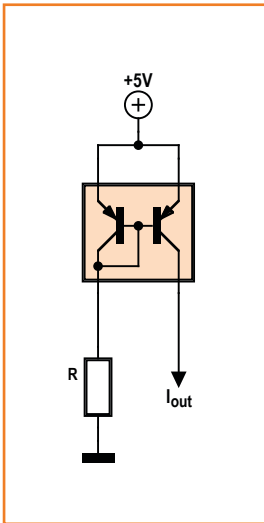


Bild 3. Die Konstantstromquelle „nach Lehrbuch“ weist für unsere Anwendung einige Nachteile auf.

Als Schalter können wir einen MOSFET oder einen bipolaren Transistor verwenden. Beide haben ihre Vor- und Nachteile: Ein MOSFET hat (zu) viel Gatekapazität und ein bipolarer Transistor braucht Zeit, um sich von der Sättigung zu erholen. Eine gute Lösung stellt die Open-Kollektor-Technologie dar. Die bevorzugte LVC-Logik ist sehr schnell, akzeptiert 5 V und kann hohe Ausgangsströme (32 mA) liefern. Zur Erzeugung des Schaltsignals wird ein konfigurierbares LVC-Schmitt-Trigger-Gatter eingesetzt.

Mit einer Versorgungsspannung von +5 V kann der Sägezahn 3 V erreichen, bis der Schmitt-Trigger schaltet. In dem Video-Sender sorgt die RC-Kombination R1/C1 für eine Pulsbreite von 10 ns, was ausreichend ist, um den 22-pF-Kondensator C6 gänzlich zu entladen. Der Kondensator hat $50 - 10 = 40$ ns Zeit, um sich von 0 V auf 3 V aufzuladen. Bei 3 V sollte auch die Konstantstromquelle nicht in Sättigung geraten. Die einfachste Stromquelle mit niedriger Sättigung besteht aus zwei PNP-Transistoren, die wie in **Bild 3** angeordnet sind. Sie weist aber mehrere gravierende Nachteile auf. Zum einen kann der h_{fe} der beiden Transistoren sehr unterschiedlich sein, selbst wenn ein Doppeltransistor wie der

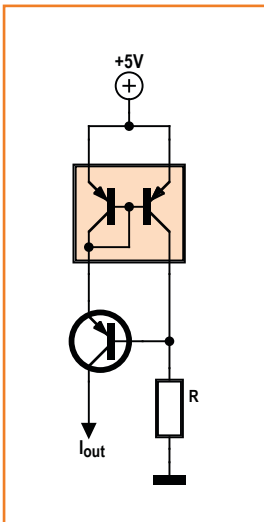


Bild 4. Durch den zusätzlichen HF-Transistor verhält sich der Wilson-Stromspiegel nahezu ideal.

BCV62C eingesetzt wird. Zweitens ist ihre Streukapazität viel zu hoch. Die Lösung ist eine Konfiguration mit einem zusätzlichen HF-Transistor wie in **Bild 4**, die Wilson-Stromspiegel genannt wird. Sie ist genauer und hat eine viel niedrigere Ausgangskapazität (etwa 3 pF). Mehr über den Wilson-Stromspiegel erfahren Sie bei Wikipedia [1]. Die Sättigungsspannung beträgt etwa 3,6 V, das ist deutlich höher als 3 V. Durch Anpassung von R (das sind R4 + R3 in unserem Sender) können wir die Sägezahn-Frequenz auf 20 MHz einstellen. Der Strom durch R beträgt:

$$I_R = U_{\text{sat}} / R$$

was für unsere Senderschaltung bedeutet:

$$I_R = 3,6 \text{ V} / (1100 \Omega + 75 \Omega) = 3,06 \text{ mA}$$

Idealerweise sollte dies gleich dem Ausgangsstrom der Konstantstromquelle sein. In unserer Schaltung fließen, wie eine Messung mit dem Multimeter über C6 belegt, 2,67 mA. Der Wert ist etwa 0,4 mA niedriger als erwartet. Dies liegt in der Differenz beider Transistoren begründet. Wir können nun die Summe aller zu C6 parallelen Kapazitäten (inklusive C6) berechnen. Wir verwenden die Formel zum Laden eines Kondensators mit einem Konstantstrom:

$$Q = I \times t = C \times U$$

$$C = I \times t / U$$

$$C = (2,67 \times 10^{-3} \times 40 \times 10^{-9}) / 3$$

$$C = 35,6 \text{ pF.}$$

Optischer Sender, optischer Empfänger

Der optische Sender verwendet den HFBR1414Z von Avago (ehemals Hewlett Packard), ein altertümlicher LED-Typ mit sehr schlechtem Wirkungsgrad. Es ist sehr schwierig, das Licht der LED so zu bündeln, dass eine angemessene „Menge“ Licht in die Faser eingekoppelt werden kann. Deshalb wird die LED auch mit 60 mA betrieben.

Optische Sender zur Kopplung einer Glasfaser werden kurz TOSA (für Transmitter Optical Sub-Assembly) genannt. Wenig überraschend heißt die Empfangseinheit dann ROSA.

Es ist ziemlich kompliziert, die Lichtquelle mit der Faser zu verbinden. Ein 3D-Manipulator wird verwendet, währenddessen wird ständig das Licht vermessen, das aus der Faser kommt. Danach wird der Verbinder an der richtigen Stelle befestigt.

Natürlich würde ein Laser einen viel besseren Job machen. Der Vertical Cavity Surface Emitting Laser (VCSEL) durchlief eine lange Entwicklungsperiode vor 1990 und ist heutzutage recht preiswert zu haben.

Also warum kein VCSEL hier? Weil all die billigen VCSELs in TOSAs namens LC-TOSA eingebaut sind. LC ist einer der etwa 20 Verbinder für optische Fasern auf dem Markt. Der LC-Stecker wurde speziell für High-Density-Patch-Panels entwickelt. Also wäre alles, was man tun müsste, die billigen LC-TOSA und LC-ROSA zu kaufen und LC-Patchkabel dazu. Sie werden feststellen, dass Sie sie zusammenstecken können (und sie passen!), aber es gibt nichts, um sie zusammen zu halten. Offensichtlich wird eine Kunststoffhalterung gebraucht, um sie zusammen zu halten, aber leider gibt

Das Glasfaserkabel

Materialien

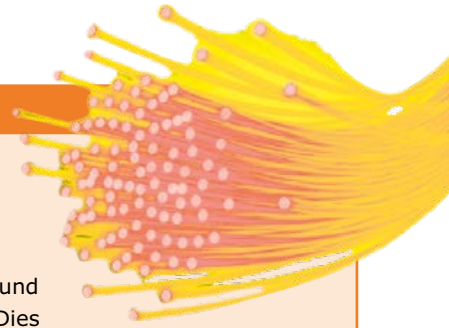
- **Quarzglas (Silica):** Silica ist Siliziumdioxid (SiO_2). Die Dämpfung bei 1300 nm beträgt nur 0,2 dB pro km. Das Silica muss sehr rein sein, was aber dank der Erfahrungen der Halbleiterindustrie kein Problem ist.
- **Kunststoff:** Plastic Optical Fiber (POF) verwendet Acryl (PMMA). Diese Kabel wurden früher meist als Patchkabel für kurze Strecken eingesetzt. Heutzutage werden perfluorierte Polymere verwendet, so dass eine hohe Bandbreite bei Entfernungen von 1 km und mehr erreicht werden kann.

Modi

- **Multimode:** Multimode-Lichtwellenleiter werden für kurze Distanzen meist innerhalb von Gebäuden verwendet. Der Name deutet darauf hin, dass das Licht unterschiedliche Distanzen in der Faser zurücklegt, was an den unterschiedlichen Reflexionswinkeln liegt. Das Licht wird in der Faser an der Grenze zwischen dem Kern und dem Kernmantel reflektiert. Dieses Problem

wird durch Verwendung der Gradientenfaser gemildert, bei welcher der Schritt zwischen den Brechungsindizes von Kern und Mantel allmählich verläuft. Dies erhöht den Nutzen von Multimode-Kabeln erheblich. Gradientenfasern werden auch in neuen „Kunststoff“-Kabeln (GIPOF) verwendet.

- **Single-Modus:** Im Single-Modus kennt das Licht nur einen Weg, um durch die Faser zu reisen. Der Faserkern ist viel dünner (8 Mikrometer) als die Ummantelung (125 Mikrometer). Bei 1300...1500 nm können lange Strecken bei hohen Bandbreiten erzielt werden. Dieses System hat aber viele Nachteile. Es ist viel schwieriger, aufgrund des kleinen Querschnitts genügend optische Leistung auf die Faser zu übertragen. Zudem sind die ROSA und TOSA wegen der erforderlichen Fertigungs- und Montagegenauigkeit weit schwieriger anzufertigen.



Die Summe aller zu C6 parallelen zusätzlichen Kapazitäten beträgt also 13,6 pF. C6 selbst besitzt eine niedrige Toleranz von 5 %, was mit COG/NPO-Dielektrikum durchaus möglich ist. Dieses Design ist durchaus reproduzierbar, jede Schaltung, die Sie aufbauen, arbeitet in der Nähe von 20 MHz. Zurück zu Bild 1. Das Videosignal erreicht zunächst einen Pufferverstärker. Dieser Verstärker stellt den Pegel des Videosignals so ein, dass der negative Teil (die Synchronisationsspitzen) bei 0,4 V

beginnt, ausreichend hoch über der niedrigsten Spannung des Sägezahns. Die höchste Spannung des Videosignals beträgt etwa 2 V, diesmal ausreichend niedrig unter der 3-V-Amplitude des Sägezahnsignals.

Der Komparator U5 ist sehr schnell und arbeitet mit einer Verzögerung von nur 4 ns. Die Verzögerungen sind für positive und negative Signalausschläge in etwa gleich, so dass die Verzerrung minimal ist. Wenn Sie dieses Signal einem

es niemanden, der ein solches Bauteil „lose“ anbietet. Die Hersteller von Patchfeldern verwenden diese Komponenten in einem Standard-Modul namens SFP (Small Form-factor Pluggable) und produzieren ihre eigenen Systeme, um sie zusammen zu halten. Meist sind diese Halteklammern Teil des SFP-Gehäuses.

Wir wollten kein SFP-Modul einsetzen - zu teuer, zu sperrig, zu unnötig. Vielleicht können freundliche Elektor-Leser ein solches Plastikteil ... drucken?

Moderne VCSEL arbeiten bei 6 mA und geben dabei eine höhere optische Ausgangsleistung ab als eine LED mit 60 mA. Avago produziert einen 850-nm-VCSEL mit einem ST-Stecker, wie wir ihn hier verwenden. Das Gerät ist aber etwa 10 Mal so teuer wie ein LC-TOSA. Der ST-Stecker ist auch viel

größer als ein LC-Anschluss und hat wie BNC eine Bajonett-Verriegelung. Der LC-Stecker sieht eher wie ein modularer RJ-45-Telefonanschluss aus (siehe Foto).



Fragen über Fragen

• *Wie linear ist die Übertragungsfunktion des Video- und des Audiokanals?*

Die Videoübertragung ist grundsätzlich linear, ausreichend für ihren Zweck. Wenn Sie eine viel niedrigere PWM-Frequenz wählen, können Sie wahrscheinlich eine Linearität von 0,1 % erzielen, aber die Temperaturdrift der Konstantstromquelle würde alles zunichtemachen. Die Soundübertragung ist grundsätzlich nichtlinear. Am Eingang wird die Periodenzeit linear moduliert, auf der Empfängerseite ist die Ausgangsspannung eine lineare Funktion der Frequenz, also des Kehrwerts. Bei geringfügigen Abweichungen wäre es „beinahe“ linear, aber aus den schon erläuterten Gründen habe ich eine Linearität von ± 10 % gewählt. Dies erscheint mir als ein guter Kompromiss.

• *Wie hoch kann man mit der PWM-Frequenz gehen?*

Es gibt keine theoretische Grenze. Ich denke, 40 MHz sind möglich.

• *Wer soll Video über Glasfaser verwenden?*

Schwer zu beantworten! Ich denke, jeder, der ein Composite-Videosignal über mehr als 100 m transportieren möchte. Das kann in großen Gebäuden, Fernüberwachungseinrichtungen oder an Orten erforderlich sein, wo kein WLAN möglich ist, wie zum Beispiel unter Wasser. Es könnte auch für Verkehrsüberwachungskameras an stark befahrenen Straßen eingesetzt werden.

Entwicklungsziele

Als ich in den 80ern zuerst über Video-Übertragung über Glasfaserkabel gelesen habe, war meine erste Reaktion: Was ist daran so kompliziert? Verwendet einfach eine schnelle PWM! Mein Beruf führte mich aber immer in andere „elektronische“ Richtungen, so dass ich das nie in der Praxis ausprobieren konnte. Erst jetzt, wo ich in „Altersteilzeit“ bin, habe ich endlich Zeit, mich damit ernsthaft zu beschäftigen. Natürlich habe ich vorher im Internet recherchiert, ob sich jemand anderes an diese Arbeit gemacht hatte, aber ich konnte nichts finden. Das Entwicklungsziel war sehr einfach: Bauen Sie ein Composite-Video-PWM-System, das die Qualität des Composite-Videos nicht sichtbar ändert. Ich habe meinen Camcorder als Videoquelle eingesetzt und auf meinem 40-Zoll-TV die feinen Details und Farben des Videosignals beobachtet. Soweit ich das beurteilen kann, gibt es keinen Unterschied in der Qualität. Also wurde das Ziel erreicht. Erst als ich die Schaltung aufbaute, merkte ich, dass ich auch den Sound frequenzmodulieren konnte. Ein Bonus! Ich benutze ein 100 MHz Tektronix TDS1012. Als ich den 20-MHz-Sägezahn untersuchte, verwendete ich ein kurzes Stück verzinnten Kupferdrahts mit einer Schleife als Sonde (Sondenadapter des armen Mannes). Der Sägezahn zeigte anfänglich eine gewisse Nichtlinearität, was aber im Videosignal nicht sichtbar wurde. Zunächst baute ich den Sägezahn-Generator. Es dauerte etwa vier Entwürfe, bis dieser gut funktionierte. Dann habe ich eine Platine mit Sägezahn-Generator, Komparator und Videofilter aufgebaut. Danach musste ich die optischen Teile und die Audiosignalverarbeitung hinzufügen, es wurde ein echtes Spaß-Projekt. Warum hat das vorher niemand versucht? Ich weiß es wirklich nicht. Vielleicht weil die Leute denken, dass alle besteigbaren Berge schon bestiegen sind...

Filter zuführen, um die 20-MHz-Komponente zu entfernen, erhalten Sie am Filterausgang wieder das originale Video-Signal.

Alles, was wir jetzt noch tun müssen, ist, das PWM-Signal in einen optischen Sender zu führen und es auf eine Glasfaser auszukoppeln. Auf der anderen Seite haben wir einen schnellen optischen Empfänger, der das optische Signal wieder in ein elektrisches umwandelt. LWL-Sender und-Empfänger haben Verzögerungen im Piko-sekundenbereich. Sie spielen für uns keine Rolle. Das optische Signal wird mit einer für diese Anwendung üblichen Infrarot-Wellenlänge von 850 nm übertragen.

Audio

Auch wenn man die PWM-Frequenz ein wenig variiert, arbeitet die Schaltung zufriedenstellend und überträgt das Videosignal störungsfrei. Es stellte sich heraus, dass man die Frequenz im Bereich von etwa ± 10 % ungestraft verändern kann, also von etwa 18 MHz bis 22 MHz. Das nutzen wir aus, um auch den Sound (Audio) zu übertragen, indem wir das Trägersignal frequenzmodulieren. Auf der Empfängerseite lässt sich das Audiosignal mit einer Phasenregelschleife (PLL) extrahieren. Die Audiofrequenz wird auf den von der Konstantstromquelle zugeführten Treiberstrom moduliert.

Dies ist auch der Grund dafür, dass die Spannungsversorgung stabil und genau sein muss. Wir benötigen am Eingang J2 nur eine geregelte Versorgungsspannung von 6 V. Die beiden 5-V-Regler U6 und U7 werden zusammen mit zwei LC-Filtern verwendet, um die Versorgungsschiene so sauber wie möglich zu halten. Jedes Rauschen oder Brummen wird nämlich als Frequenz moduliert und erscheint unweigerlich auf dem PLL-Ausgang des Empfängers. Aus diesem Grund verwenden wir auch eine Breitband-Frequenzmodulation. Die Versorgung der Konstantstromquelle muss möglichst unabhängig von der des Pulsweitenmodulators sein, weil dieser sonst die Form des Sägezahns beeinflussen würde. Sowohl Sender als auch Empfänger sind mit SMDs aufgebaut, die Platine muss auf der Unterseite eine Massefläche aufweisen. Auf einer Platine für diese Schaltung sind wegen der 20-MHz-Signale lange Leiterbahnen nicht erwünscht.

Der Empfänger

Schauen wir uns nun **Bild 5** an. Das optische Signal gelangt über J2 im Empfänger zur Schaltung.

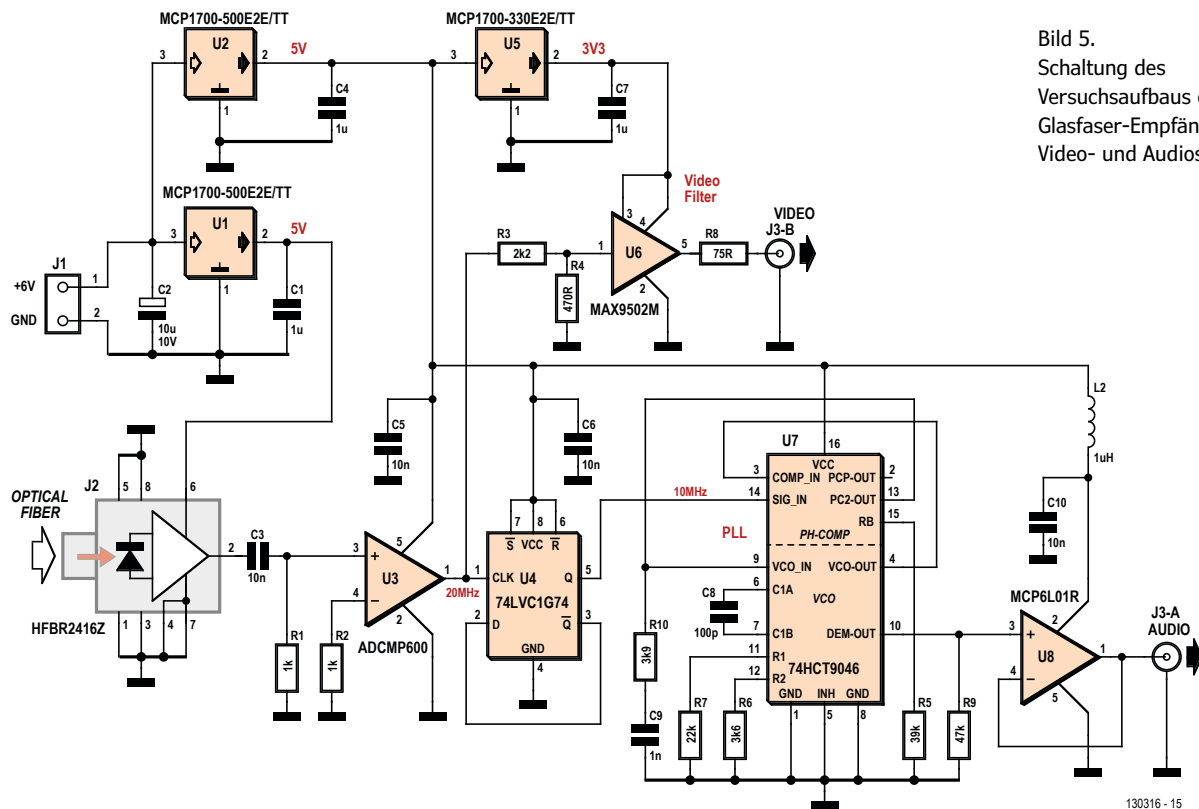
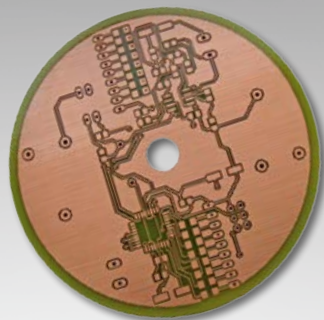


Bild 5.
Schaltung des
Versuchsaufbaus eines
Glasfaser-Empfängers für
Video- und Audiosignale.

Von dort geht es zum schnellen Komparator U3. Wenn Sie dessen Ausgangssignal mit dem am Ausgang des Komparators U5 im Sender vergleichen, werden Sie feststellen, dass sie identisch

sind und lediglich eine Verzögerung von ungefähr 8 ns + 5 ns je Glasfaserkabelmeter aufweisen. Das PWM-Signal wird dann zum Filter-IC U6 geführt. Der MAX9502M kann nicht mit 5 V betrie-

Anzeige



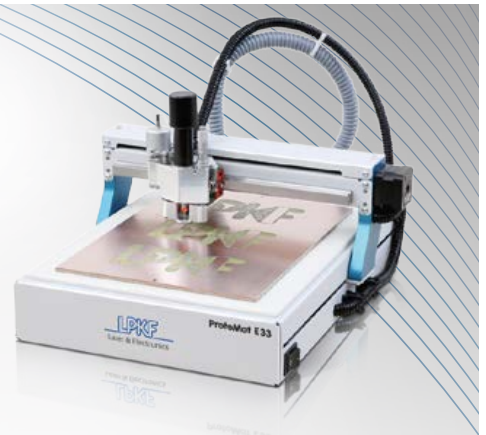
(M)ein kleiner Freund im Elektroniklabor

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

LPKF ProtoMat E33 – klein, präzise, wirtschaftlich

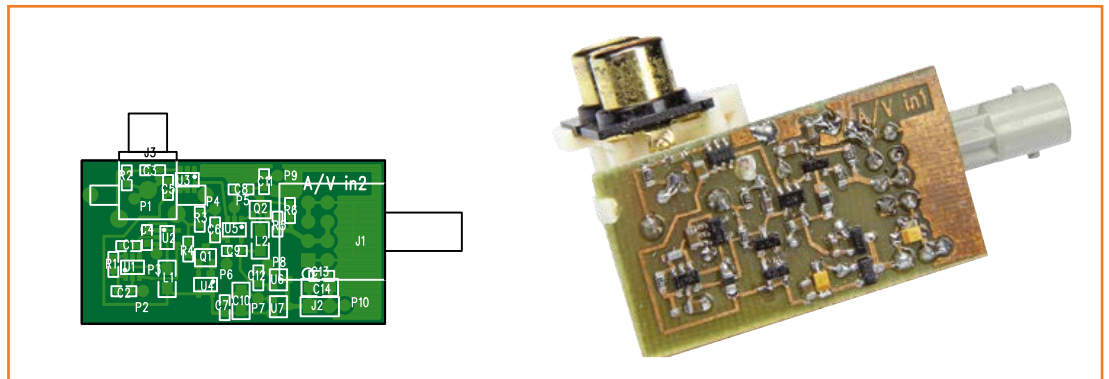
Kaum größer als ein DIN A3-Blatt: LPKF Qualität zum Einstiegspreis zum Fräsen, Bohren und Trennen von Leiterplatten und Gravieren von Frontplatten. Erfahren Sie mehr: www.lpkf.de/prototyping

electronica: 11. – 14.11.2014, Halle 2, Stand 419



LPKF
Laser & Electronics

Bild 6.
Die Sender-Platine des Autors wurde hier um 150 % vergrößert.



ben werden, er erhält seine eigene Versorgungsspannung vom 3,3-V-Regler U5. Der Filterbaustein ist ein Tiefpassfilter mit einer -3-dB-Grenzfrequenz von 7 MHz und einer Steilheit von 30 dB/Oktave. Ein solch gutes Filter ist wegen der starken 20-MHz-Komponente unumgänglich. Der MAX9502M besitzt eine Schaltung zur Wiederherstellung einer Gleichspannungskomponente, nur so am Rande erwähnt, wir gebrauchen sie hier nicht. Um Clipping am Ausgang zu vermeiden, wird das PWM-Signal vor dem Eingang von R3/R4 um den Faktor 5 abgeschwächt. Der MAX9502M verstärkt es wiederum um 12 dB. Kommt der MAX9502G mit 6 dB Verstärkung zum Einsatz, muss R4 auf 1 k Ω erhöht werden. Hinter U6 steht das originale Videosignal wieder zur Verfügung.

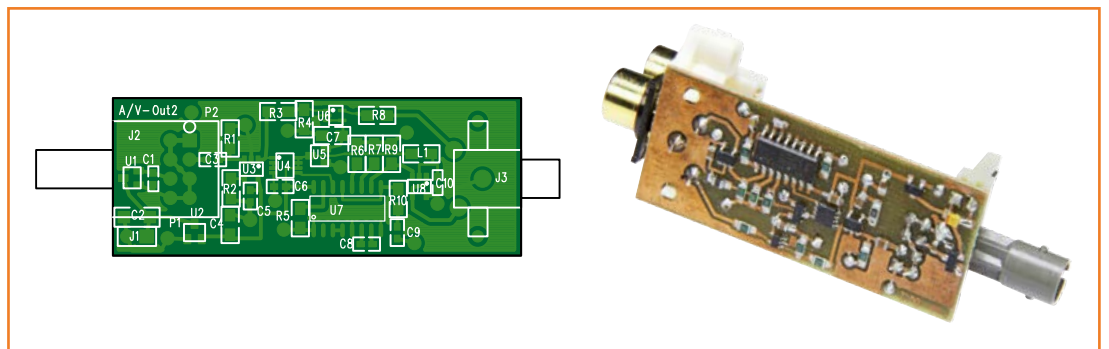
Die PLL-Schaltung U7 ist für den +5-V-Betrieb ausgelegt und auch U3 und U4 lassen sich mit dieser Spannung versorgen. Da die maximale Frequenz des 74HCT9046 (U7) „nur“ bei etwa 15 MHz liegt, muss die Frequenz des Videosignals zunächst vom D-Flip-Flop-Typ 74LVC1G74 (U4) halbiert werden. Mit 10 MHz ist die Frequenz für diese PLL immer noch recht hoch, aber es funktioniert zufriedenstellend.

Das D-Flip-Flop reagiert auf die positiven PWM-Signalfanken; die negativen Flanken sind ja moduliert und können nicht verwendet werden. Das achtpolige SO-8-Gehäuse des SN74LVC2G80DCT bietet gleich zwei Flip-Flops Platz. Damit ließe sich die PWM-Frequenz sogar auf 5 MHz teilen. Die Frequenzhalbierung wird erreicht, indem man den /Q-Ausgang zum D-Eingang zurückführt. Der Q-Ausgang des ersten Flip-Flops ließe sich mit dem Takteingang (CLK) des zweiten verbinden. Dies können Sie nutzen, wenn Sie herausfinden wollen, wie hoch Sie mit der PWM-Frequenz gehen können.

Die PLL 74HCT9046 wird von NXP (ehemals Philips Semiconductors) gefertigt. Es ist eine verbesserte Version der altbekannten PLL 4046. Die VCO-Abteilung hat zwei Optionen: Bei der ersten steuert man die Frequenz mit einem Widerstand, kann aber so nur ein kleines Ausgangssignal erzeugen. Die zweite Möglichkeit ist, eine Offset-Frequenz einzusetzen, um die Frequenz um die Grundfrequenz zu variieren. In unserem Fall liegt die Grundfrequenz bei 10 MHz.

Die Frequenz-Schwankung beträgt ± 1 MHz. Die 10-MHz-Frequenz wird mit R7 eingestellt, die

Bild 7.
Die Empfänger-Platine des Autors wurde hier um 150 % vergrößert.



± 1 -MHz-Abweichung mit R8. Es kann sein, dass R7 leicht angepasst werden muss, wenn die unmodulierte Frequenz um mehr als 0,5 MHz daneben liegt. Die Spannung am Audio-Ausgang sollte 2,5 V sein (ohne Audio- und Videoeingangssignal am Sender). Wenn Sie R6 von 3,6 k Ω auf 3,9 k Ω erhöhen, steigt die Ausgangsspannung der PLL und natürlich auch an J3-A um ungefähr 1 V. Es ist wichtig, dass die PWM-Frequenz möglichst genau 20 MHz beträgt, wenn man mehrere Empfänger hat und sie untereinander austauschen möchte. Um die Werte für C9 und R10 (die Rückkopplungsschleife) zu ermitteln, wurde ein Rechtecksignal an den Audioeingang auf der Senderplatine gelegt und das Unter- und Übersprechen am Audioausgang des Empfängers beobachtet.

Ein experimenteller Aufbau

Die Platinenlayouts für Sender und Empfänger, wie sie vom Autor gestaltet wurden und in **Bild 6** und **Bild 7** zu sehen sind, stellen eher eine Quelle der Inspiration dar als eine genaue Bauanleitung, an die man sich sklavisch halten muss. Die Layouts stehen unter [2] zum kostenlosen Download bereit. Die aufgebauten Boards in den Fotos sind frühe Prototypen und unterscheiden sich leicht von den finalen Entwürfen.

Pad-Größen, Bestückung und Leiterbahnführung sind sicherlich nicht optimal, aber die Prototypen haben funktioniert und konnten den Elektor-Mitarbeitern bei einem Besuch des Autors in der niederländischen Elektor-Zentrale erfolgreich vorgeführt werden.

Die Platinenlayouts wurden mit PADS von Mentor Graphics erstellt, die Filme wurden mit einem Laserdrucker gedruckt. Die Platten sind doppelseitig und mit Photoresist beschichtet. Sie wurden mit Ammoniumpersulfat bei 50 °C geätzt.

Um die Bauteile mit Lötpaste zu versehen und zu platzieren, wurde die Hilfe eines optischen Mikroskops in Anspruch genommen. Die Lötpaste wurde mit einer Spritze aufgetragen, dann die Bauteile vor dem Löten in einer Bratpfanne (mit Deckel) auf maximaler Hitze aufgeheizt. Nachdem man die Boards hineingesteckt hat, wird der Deckel lang genug angehoben, um beobachten zu können, wie das Lötzinn schmilzt. Dann holt man die Boards vorsichtig mit einer großen Spitzzange aus dem Feuer.

Montage und Überprüfung der beiden kleinen Platinen schließen das interessante Experiment zur preisgünstigen Übertragung von Video und Ton über Glasfaser ab, ohne dass man spezielle ICs oder professionelle Ausrüstung benötigen würde. Wir hoffen, dass es Sie zu eigenen Versuchen inspiriert...

(130316)

Weblinks

[1] Wilson-Stromspiegel:

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

<http://de.wikipedia.org/wiki/Stromspiegel>

[2] Platinenlayouts im Mentor Graphics PADS Format:

www.elektor-magazine.de/130316

[3] www.glasfaserinfo.de/

UNSCHLAGBAR beim Preis-Leistungsverhältnis.



Rigol DS1000E Oszilloskope

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

ab € 284,41
inkl. MwSt. und Versand



Rigol DS1000Z Oszilloskope

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

ab € 355,81
inkl. MwSt. und Versand

Machen Sie Ihr **LEBEN** leichter.
Führende **LABORTECHNIK**
mit **BATRONIX** Zufriedenheitsgarantie

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

Jetzt Angebote nutzen:
www.batronix.com/go/39

NEU

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

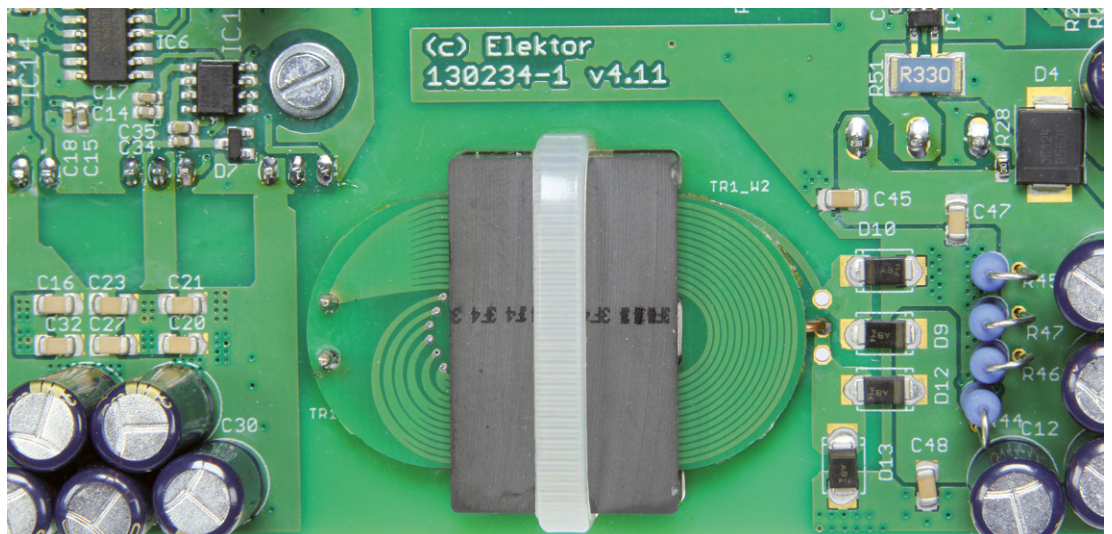
Tel.: +49 (0)4307/8243-20
service@batronix.com
www.batronix.com

Platinenintegration magnetischer Bauelemente

Prinzipien und Simulation

Von
Martin Christoph
(D)

Spulen und Trafos sind nicht aus der Elektronik wegzudenken. Auch im Projekt „Professionelles Labornetzteil“ aus Elektor September 2014 war ein Trafo erforderlich. Seine Wicklungen bestanden aus Leiterbahnen einer Platine. Dieser Beitrag beschäftigt sich mit der speziellen Bauweise solcher magnetischer Komponenten und ihrer Simulation.



Trafos und Induktivitäten gibt es in vielen Ausführungen und mit unterschiedlichen Eigenschaften. Oft wird jedoch ein Bauteil benötigt, das genau auf die Anwendung zurechtgeschnitten wurde. Gerade bei Trafos wird die Auswahl dann eng, weil es nicht nur auf die Leistungsdaten, sondern auch auf das Wicklungsverhältnis und auf die Betriebsfrequenz ankommt. Wenn dann noch Feinheiten wie die Streuinduktivität etc. eine Rolle spielen, bleibt meist nicht mehr viel übrig. Dann hilft nur noch das Selbermachen.

So genannte „leiterplattenintegrierte magnetische Bauelemente“ können genau wie ihre konventionellen Alternativen für die Anwendung maßgeschneidert werden. Leider bietet die Platinenintegration weniger Freiheiten in Form und Wicklungsanordnung, da man hier an die Struktur

und den Fertigungsprozess der Platine gebunden ist. Die Flachheit einer Platine ist für den Aufbau zudem eher ungünstig. Nachfolgend wird beschrieben, wie man dennoch durch optimale Gestaltung möglichst viel herausholen kann.

Die Vorteile

Wie schon erwähnt wurde beim professionellen Labornetzteil [1] ein spezieller Trafo benötigt. Die ersten Prototypen waren noch konventionell und wurden von Hand gewickelt. Schnell wurde klar, dass dieser Trafo die größte Hürde für den Nachbau des Projektes darstellt. Einerseits muss man den richtigen Draht haben und dann auch noch gleich wie beim Prototypen wickeln, um das gleiche Verhalten erzielen. Das ist schon der erste Vorteil des Aufbaus mit platinenintegrierten Wicklungen und auch der Hauptgrund für die

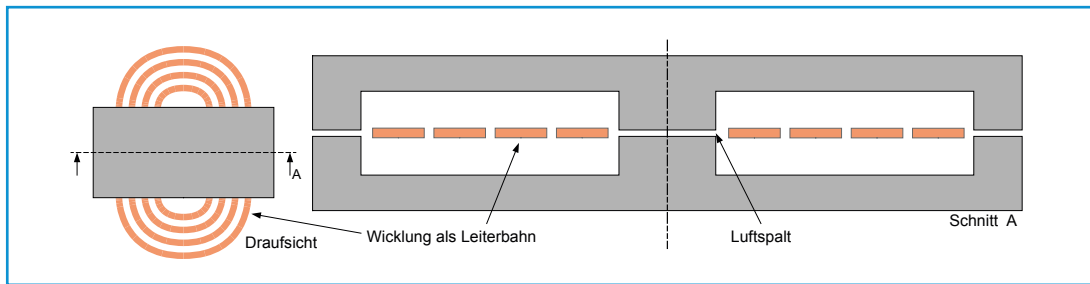


Bild 1.
Schematische Darstellung
einer platinenintegrierten
Spule.

Verwendung in diesem Projekt: Die Anordnung und somit die Eigenschaften sind beim Eigenbau nämlich sehr gut reproduzierbar.

Ein weiterer Vorteil können geringere Kosten sein. Das gilt sowohl für den industriellen Einsatz als auch für den Eigenbau. Gerade im kleinen Elektronik-Labor ist es von Vorteil, nicht ein ganzes Arsenal an Drähten und Litzen vorrätig halten zu müssen, sondern durch geschicktes Platinen-Design die gewünschte Funktion erzielen zu können. Ein zusätzlicher Aspekt sind die, bei entsprechendem Design, guten Isolationseigenschaften dieser Anordnung. Besonders gute Isolationen erreicht man mit einer Multilayer-Platine: Die Wicklungen sind in den inneren Lagen untergebracht und es werden nur „buried vias“ (verdeckte Durchkontaktierungen) verwendet. Die äußeren Lagen bleiben zur Vermeidung von Kriechstrecken an der Oberfläche unbenutzt und frei von Vias. Die Wicklungen sind dann in der Platine verpresst und somit sehr gut gegenüber dem Kern und anderen Wicklungen isoliert. Kommerzielle Produkte wie etwa 1,7-kV-IGBT-Treiber sind genau so aufgebaut.

Trafos mit platinenintegrierter Wicklung lassen sich mit sehr geringer Streuinduktivität konzipieren, da Primär- und Sekundärwicklung direkt übereinander angeordnet werden können. Die Kopplung beider Wicklungen ist somit sehr hoch. Eine geringe Streuinduktivität ist bei hohen Schaltfrequenzen von Vorteil.

Spezielle Eigenschaften

Zur Veranschaulichung des Prinzips zeigt **Bild 1** einen Querschnitt durch eine einfache Spule. Das Bauteil besteht aus zwei E-Kern-Hälften und einer aus Leiterbahnen realisierten Wicklung. Der Kern besitzt den für Spulen üblichen Luftspalt. Wegen der Symmetrie genügt die Betrachtung einer Hälfte.

In **Bild 2** oben ist der unerwünschte magnetische Streufluss einer Hälfte dargestellt, welcher

sich eben nicht über den Kern, sondern durch das Wickelfenster (in dem die Wicklungen sind) schließt. Er setzt sich aus unvermeidlichem Streufluss aus dem Kern heraus (besonders in der Nähe des Luftspalts) sowie dem Fluss benachbarter Windungen zusammen und durchdringt die Wicklungen in der Platine. Die untere Hälfte von Bild 2 zeigt, wie durch diesen Fluss durch die Leiterbahn in diese ein Strom eingeprägt wird, der an einer Seite entgegengesetzt zur anderen fließt. Dieser Strom ist dem eigentlichen, durch die Leiterbahn fließenden Strom überlagert und produziert in dieser Widerstandsverluste. Der überlagerte Strom steigt mit dem Streufluss und der Betriebsfrequenz, da die im Leiter induzierte Spannung von diesen Größen abhängt.

Die beschriebene Problematik sollte man nicht mit dem Skin-Effekt verwechseln, der lediglich die Stromverdrängung in einem Leiter mit steigender Frequenz beschreibt. Gemeint sind hier der Einfluss von Streufeldern auf den Leiter (siehe Bild 2) und der Proximity-Effekt, welcher die Beeinflussung eng benachbarter Leiter untereinander beschreibt. Der Skin-Effekt ist bei

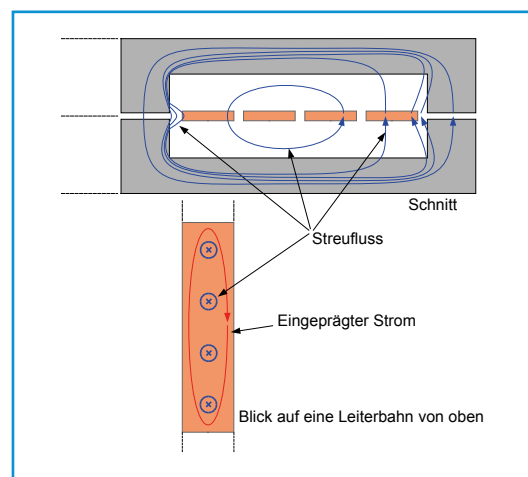
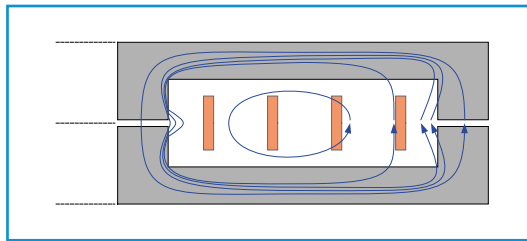


Bild 2.
Streufluss der Anordnung
und resultierende
eingeprägte Ströme.

Bild 3.
Anordnung für
eine minimale vom
Streufluss durchsetzte
Leiterbahnfläche.



platinenintegrierten Wicklungen weniger ausgeprägt, da Leiterbahnen sehr dünn und somit für hohe Frequenzen gut geeignet sind.

Diese Verluste sind umso größer, je größer die vom Fluss durchsetzte Leiterbahnfläche ist. Eigentlich wären schmale und hohe Leiterbahnen wie in **Bild 3** optimal. Dies würde allerdings eine sehr dicke Kupferschicht der Platine erfordern, die sehr teuer und kaum realisierbar wäre. Das Problem ist allenfalls optimierbar. Bei konventionell aufgebauten magnetischen Bauelementen lässt sich eine Wicklung nach Bild 3 sehr einfach durch die Verwendung von Folien erzielen.

FEM-Simulation

Um die Effekte in der Induktivität genauer zu untersuchen, bietet sich eine Simulation mittels FEM (**F**inite-**E**lemente-**M**ethode) an. Dies klingt zwar kompliziert, doch können einfache Anordnungen damit leicht dargestellt werden. Wie schon beschrieben ist die Anordnung symmetrisch. Es geht kein magnetischer Fluss über die Symmetrieachse. Man muss also nur eine Hälfte simulieren. Würden sich die Leiter in der Mitte befinden, würde sogar ein Viertel genügen. Nimmt man an, dass sich der relevante Teil unterhalb des Kernmaterials befindet, ergibt sich eine planare Anordnung mit definierter Tiefe. Der Schnitt durch den Kern lässt sich dann mit der Kerntiefe als Parameter simulieren. Hierfür eignet sich das frei verfügbare Programm FEMM, das man mitsamt der Dokumentation unter [2] herunterladen kann. Die folgende schrittweise Anleitung erleichtert das Nachvollziehen der Simulation. Die exakten Parameter sind gegenüber der allgemeinen Vorgehensweise nicht so wichtig.

Nach Download und Installation wird das Programm gestartet und via Menü *File* → *New* per Dialog ein neues „Magnetics Problem“ erstellt. Via Menü *Problem* gelangt man zum Dialog „Problem Definition“. Für den *Problem Type* wählt man „Planar“, als Einheit „Millimeter“ und bei der Frequenz z.B. 100 kHz. Als Tiefe ergeben

sich bei einem E32/6/20-Kern 20 mm.

Für die Geometrie gibt es fünf Schaltflächen in der Symbolleiste, mit denen man die Objekt-Art spezifizieren kann (*Node*, *Segment*, *Arc Segment*, *Label* oder *Group* – siehe auch das Menü *Operation*). Wählt man ein Objekt aus, kann man per Klick oder Tab-Taste (bei *Node*, *Label* und *Group*) neue Exemplare erzeugen, diese per Rechtsklick selektieren und mit der Leertaste die Eigenschaften dieser Objekte aufrufen. Die Bedienung ist etwas eigenwillig, funktioniert aber nach Eingewöhnung recht gut.

Zuerst wird ein Rand um das Problem konstruiert. Nimmt man an, dass kein magnetischer Fluss den Kern verlässt, ergeben sich verfälschte Resultate für den äußeren Luftspalt. Also umgibt man die Anordnung mit Luft. Ein Halbkreis mit entsprechenden Randbedingungen vermeidet die Definition eines unendlich großen Bereichs. Hierzu wählt man *Operate on nodes*, betätigt die Tabulator-Taste und gibt die Koordinaten „x = 0“ und „y = -20“ ein. Eine weitere Tab-Betätigung erzeugt einen *Node*, für den man „x = 0“ und „y = 20“ eingibt.

Danach wählt man *Operate on segments* und klickt zur Verbindung mit einer senkrechten Linie zunächst auf den unteren und dann auf den oberen Punkt. Nun wählt man *Operate on arc segments*, klickt wieder beide Punkte an, gibt im erscheinenden Dialog einen „Arc Angle“ von 180 ein und fertig ist der Halbkreis. Der weitere Aufbau der (halben) Geometrie erfolgt nach dem gleichen Verfahren.

Das Programm kann komplett durch ein eigenes Script von Matlab oder Octave gesteuert werden. Die Anleitung dazu findet sich im Handbuch.

Die Wicklung kann ähnlich wie in Bild 1 eingezeichnet werden. Für Experimente sind die genauen Abmessungen nicht so wichtig. Zur Info: Eine normale Leiterbahn ist 35 µm hoch. Mit den Angaben im Datenblatt zum E32/6/20-Kern [3] lässt sich die (halbe) Kerngeometrie leicht nachbauen. Als Luftspalt zwischen den Kernhälften wählt man z.B. 0,2 mm.

Über *Properties* → *Materials Library* lassen sich Materialien auswählen. Man kann sie vom linken Feld per „drag & drop“ nach rechts in die „Model Library“ ziehen. Für den Anfang reichen *Air*, *Copper* und *Soft magnetic ferrite*.

Mit *Properties* → *Circuits* legt man einen Stromkreis an. Mit einem *Add Property* des Dialogs definiert man einen Stromkreis namens „Win-

ding1“ in der Konfiguration *Series* mit einem Strom von „0.1 A“.

Jetzt werden in allen von Linien umschlossenen Bereichen *block labels* gesetzt und mit Rechtsklick selektiert. Per Leertaste erscheint der zugehörige Properties-Dialog, in dem man das entsprechende Material auswählt. Bei der Wicklung wird im Dialog noch bei *in circuit* die vorher angelegte „Winding1“ ausgewählt und bei *number of turns* eine 1 eingetragen.

Jetzt müssen nur noch die Randbedingungen für den zu Anfang erzeugten Halbkreis gesetzt werden. Aufgrund der Symmetrie gilt auf der senkrechten Linie durch die Mitte der Anordnung „A = 0“. Hierzu wird unter „Properties“ auf *Boundary* und auf *Add Property* geklickt. Bei *Name* wird z.B. A=0 eintragen und bei *BC Type* „Prescribed A“ gewählt. Die *Prescribed A parameters* bleiben alle „0“. Nach Schließen des Dialogs werden mit *Operate on segments* durch Rechtsklick die senkrechten Linien des Halbkreises selektiert. Nun die Leertaste drücken und im Dialog die erstellte Eigenschaft „A=0“ auswählen.

Der Halbkreisrand soll sich verhalten, als wäre hier unendlich viel Luft. Dies wird mit einer weiteren Randbedingung erreicht, die wie zuvor angelegt und zugewiesen wird. Allerdings wird hier als *BC Type* „Mixed“ gewählt und bei den Koeffizienten folgendes eingetragen:

$$c_1 = 0; \quad c_0 = \frac{1}{\mu_0 r_0} = 39788735,77$$

$$\text{bei } r_0 = 0,020 \text{ m}$$

Die Berechnung findet man im FEMM-Handbuch auf Seite 155.

Diese zweite Randbedingung wird dem Halbkreis mit *Operate on arc segments* durch Rechtsklick und Leertaste zugewiesen.

Jetzt ist alles fertig spezifiziert und man kann in der Symbolleiste auf *Run mesh generator* klicken. Wenn das geklappt hat, folgt ein Klick auf *Run analysis*. Bei Problemen ist die Geometrie vermutlich irgendwo nicht geschlossen oder einem Bereich wurde kein Material zugewiesen. Alle geschlossenen Bereiche innerhalb des Halbkreises müssen ein Material haben - auch die mit Luft gefüllten. Anschließend sollte sich ein Aufbau ähnlich dem von **Bild 4** zeigen.

Die FEMM-Dateien für die drei Anordnungen kann man unter [4] kostenlos downloaden.

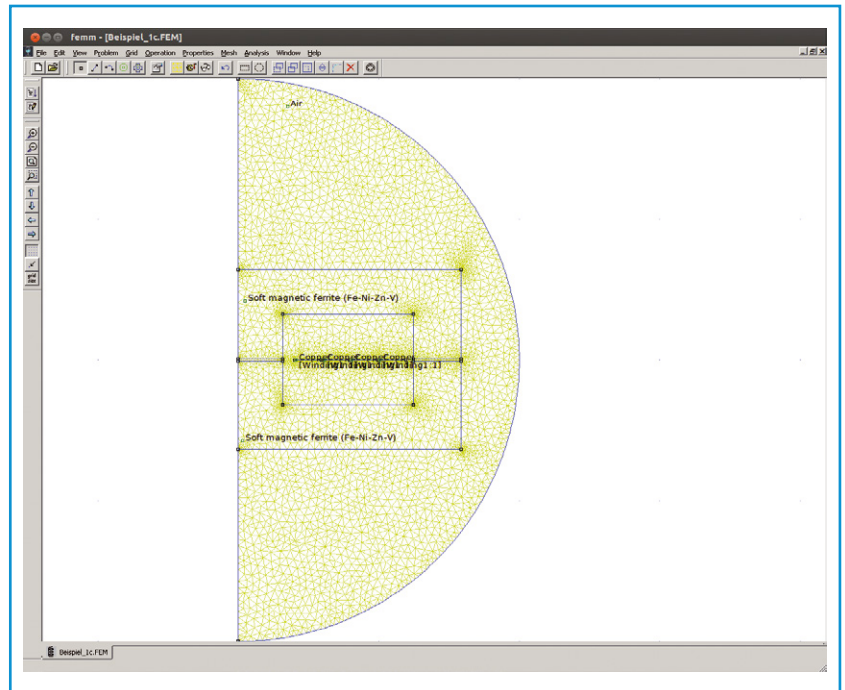


Bild 4.
Anordnung in FEMM.

Resultate

Jetzt kann man mit *View results* alle möglichen Ergebnisse inspizieren. Unter *View* → *Density plot* kann man sich z.B. mit $|Flux Density| (T)$ über die Flussdichte im Kern informieren. So kann man überprüfen, ob das Kernmaterial schon stellenweise in Sättigung geht. Wählt man $|Current Density| (MA/m^2)$, erhält man die Stromverteilung im Leiter. Man sieht hier, dass sie nicht gleichmäßig ist, sondern durch die beschriebenen Streufelder beeinflusst wird. Das zeigt sich besonders in der Nähe des Luftspalts. Bei Erhöhung der Frequenz zeigt sich ein noch stärkerer Effekt. **Bild 5**

Bild 5.
Simulation: Linien = magnetische Flusspfade; Farbe = Stromdichte.

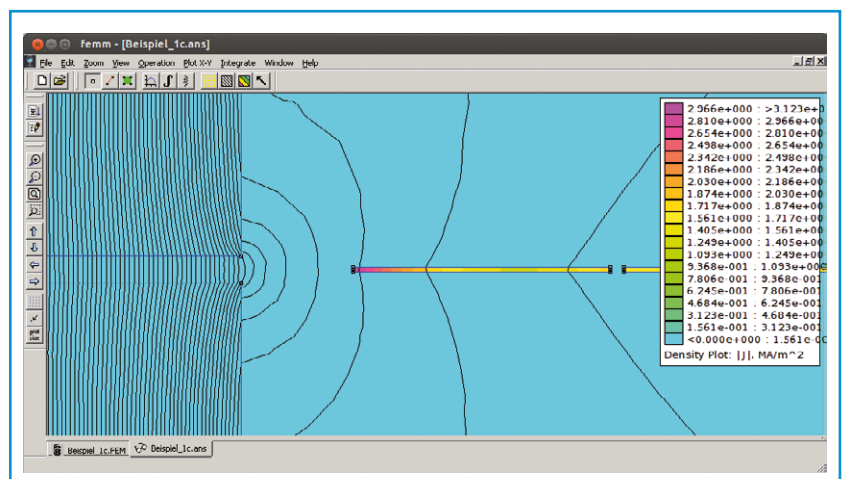


Bild 6.
Drei Leiterbahnanordnungen
mit gleichem Kupferbedarf.

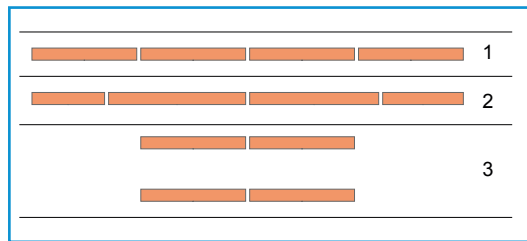


Tabelle 1. Vergleich der Verlustleistung.		
Anordnung	Verlustleistung	Gegenüber 1
1	129,634 mW	$\pm 0 \%$
2	126,359 mW	-2,5 %
3	114,508 mW	-11,7 %

zeigt das Ergebnis einer solchen Simulation im Leiter bei $f = 100 \text{ kHz}$.

Mit *Operation* → *Contours* lässt sich eine Linie durch die Anordnung legen und eine Größe auf dieser Line in einem X-Y-Plot darstellen. So kann

man z.B. eine Linie durch die Leiterbahnen legen und so die Stromverteilung anzeigen.

Auch sehr interessant ist *View Circuit Probs*. Hier werden Kenndaten der Anordnung wie Induktivität und Verlustleistung direkt angezeigt. Anhand der Verlustleistung lässt sich beurteilen, ob eine Anordnung besser ist als eine andere. **Tabelle 1** listet die Ergebnisse für die drei Leiteranordnungen von **Bild 6** auf, die alle den gleichen Gesamtkupferbedarf haben.

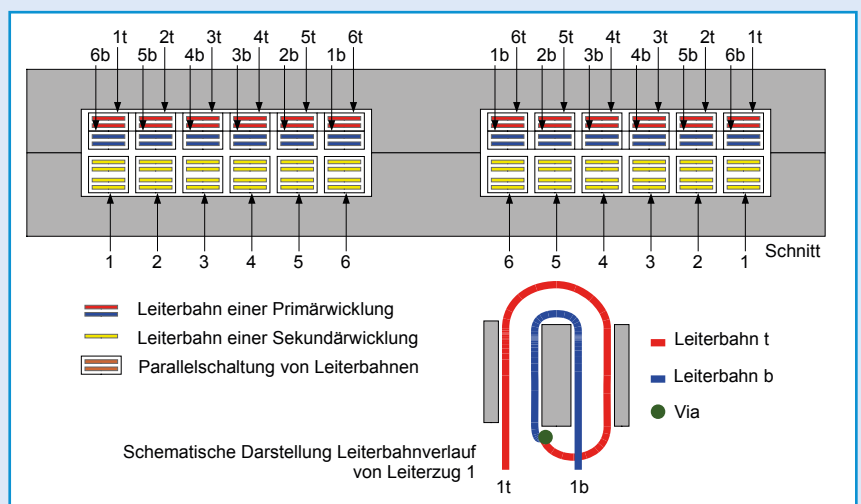
Es zeigt sich, dass hier die gleichmäßige Anordnung der Windungen auf einer Platinenlage die ungünstigste ist. Mit ungleichen Leiterbahnbreiten sinken die Verluste schon um 2,5 %. Noch besser ist die Verwendung von Ober- und Unterseite der Platine mit frei gelassenem Rand. Dann aber kommt man bei einer zweilagigen Platine nicht mehr vom Innern der Wicklung mit einer Leiterbahn nach außen. Doch die >10 % geringeren Verluste sollten die extra Drahtbrücke wohl wert sein.

Aufbau des platinenintegrierten Trafos im Labornetzteil

Der Aufbau des Trafos im Labornetzteil von Elektor September 2014 resultiert aus diversen Überlegungen und Erkenntnissen, wurde jedoch nicht in seiner Gesamtheit simuliert. Das Bild zeigt den Querschnitt durch das Wicklungsprinzip des Trafos. Es fällt auf, dass die Leiterbahnen massiv parallelgeschaltet wurden, um dem geforderten Strom überhaupt gewachsen zu sein. Die Sekundärwicklung (gelb) hat sechs Windungen und ist nicht breit in mehreren Lagen, sondern schmal in allen vier Lagen parallel ausgeführt. Somit ergeben sich hier vier parallele Leiterzüge, die ohne Vias geführt werden können und am Anfang und Ende parallel geschaltet sind. Schmale Leiterbahnen reduzieren zusätzlich den Einfluss von Streufeldern. Da alle vier Leiterzüge direkt übereinander liegen und somit die gleiche Fläche aufspannen, werden sie auch von fast gleichem Fluss durchsetzt, was in allen Leiterzügen die gleiche Spannung induziert und somit eine gleiche Stromaufteilung unter den Leiterzügen gewährleistet.

Für die Primärwicklung gilt: Durch das Vertauschen der einzelnen, an den Enden parallel geschalteten Leiterzüge hat jeder Leiterzug etwa die gleiche Gesamtfläche. Somit wird jeder Leiterzug von einem fast gleichen magnetischen Fluss durchsetzt und in ihm die gleiche Spannung induziert, was gleiche Ströme zur Folge hat. Da sich resistive Verluste quadratisch zum Strom verhalten, bringt eine gleichmäßige

Stromaufteilung die geringsten Verluste. Wer sich die Platine des Trafos genau anschaut, der erkennt, dass sich die einzeln getauschten Leiterzüge auf einer Windung nochmals aufteilen und wieder verbinden. Dies ist dem Designprozess geschuldet, da am Ende doch nicht genug Vias möglich waren. Elektrisch nicht optimal, aber die Realität ist eben oft nichtideal...



Fazit

Der Artikel gibt nur einen kleinen Einblick in die Optimierung magnetischer Bauteile. Es wird aber deutlich, dass schon kleine Design-Änderungen geringere Verluste ergeben. Es dürfte klar sein, dass die Platinenintegration nicht für alle Anwendungen gleich gut passt. Wenn man sie aber schon beim Gesamt-Design berücksichtigt, kann man durchaus zu guten Lösungen kommen. Für Induktivitäten sind Anwendungen mit hohem Gleichstromanteil (z.B. Schaltregler mit geringem Strom-Ripple oder Filter für Gleichspannung) vorteilhaft, da sich so kaum Wirbelströme auswirken. Mit hohem Wechselstromanteil ist Vorsicht beim Wicklungs-Design geboten. Bei hohen Streufeldern ist viel Kupfer in der Nähe des Luftspaltes nicht unbedingt klug. Bei Trafos ist in der Regel kein Luftspalt vorhanden, weshalb sich hier weniger negative Effekte zeigen. Die sehr geringe Streuinduktivität macht platinenintegrierte Trafos für höherfrequente Anwendungen interessant. Diese kom-

pakten Lösungen lassen sich zudem auch einfach aufbauen.

(130487)

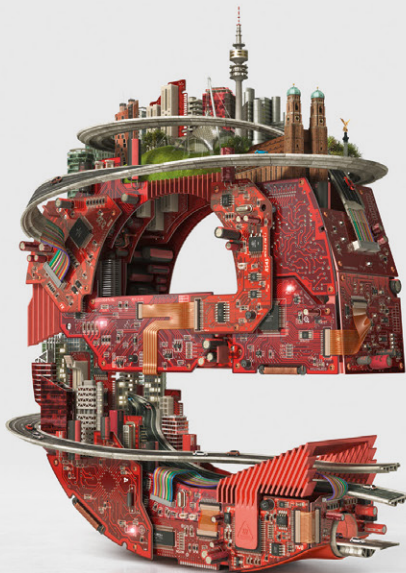
Weblink

- [1] www.elektor-magazine.de/130234
- [2] www.femm.info
- [3] www.farnell.com/datasheets/1632190.pdf
- [4] www.elektor-magazine.de/130487

Über den Autor:

Martin Christoph ist Dipl.-Ing. Elektrotechnik und hat über fünf Jahre im ISEA (Institut für Stromrichtertechnik und Elektrische Antriebe) der RWTH Aachen im Bereich Leistungselektronik gearbeitet.

Anzeige



Welcome to Planet e.

Das ganze Universum der Elektronik an einem Ort!

Tickets & Registrierung
www.electronica.de/tickets

26. Weltleitmesse für Komponenten,
Systeme und Anwendungen der Elektronik
Messe München
11.–14. November 2014
www.electronica.de

50 Jahre
electronica

 **electronica** 2014
inside tomorrow

Gyrator-Ferritantenne

Drehkondensator durch ein Poti ersetzen

Von
Martin Oßmann
(D)

Um Kondensatoren und Spulen zu simulieren, kann man Gyratoren verwenden. Dabei werden Gyratoren als Impedanzkonverter benutzt, und genau das werden wir im folgenden Artikel beschreiben. Danach zeigen wir, wie man mit einem Gyrator eine abstimmbare Ferritantenne realisieren kann.

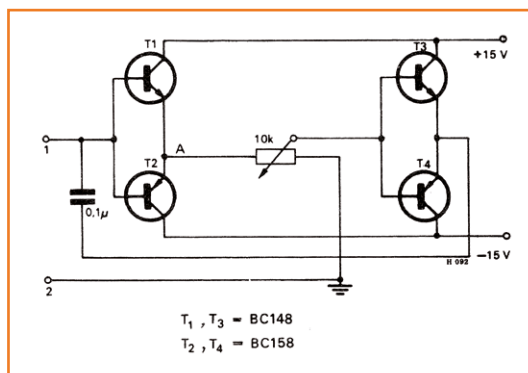


Bild 1.
Eine variable Kapazität bis 100 µF, publiziert in Elektor (November 1970).

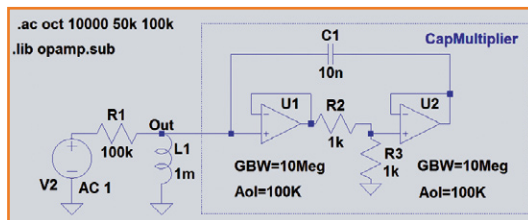
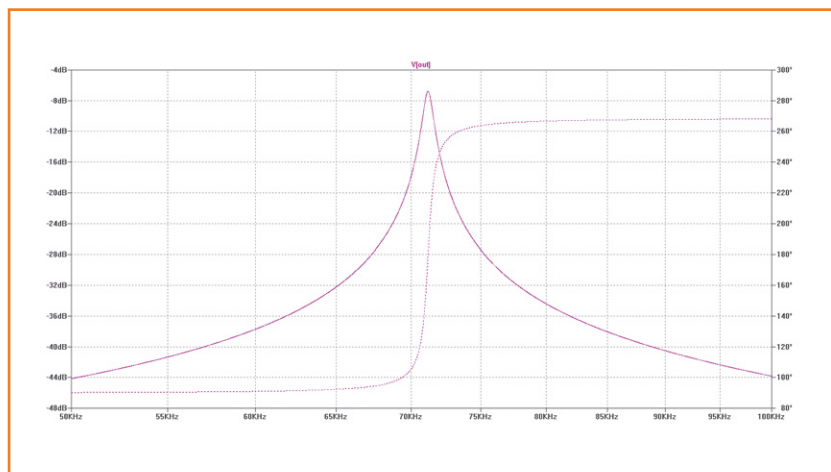


Bild 2.
Simulation des Schwingkreises mit variablem Kondensator.



Der Autor erinnert sich noch gut daran, dass er als Jugendlicher in einer Elektor-Ausgabe von 1970 die Schaltung einer variablen Kapazität entdeckte (**Bild 1**).

Ziemlich unmittelbar kam ihm die Idee, dass man so ja den Drehkondensator im Schwingkreis eines Mittelwelleneinkreislers ersetzen könnte, um eine Poti-Abstimmung zu realisieren. Probiert wurde das damals, geklappt hat es nicht. Aber der Traum vom Ersatz eines Drehkos durch ein Poti ist geblieben (auch wenn es Kapazitätsdioden gibt, aber keine mit ein paar nF).

Mit den beschränkten Mitteln eines bastelnden Schülers konnte ich damals nicht herausfinden, warum der Ersatz des Drehkos nicht klappte. Heute, mit besseren Messmitteln und einem soliden Verständnis analoger Schaltungstechnik, kann man nicht nur den damaligen Misserfolg verstehen, sondern (mit modernen Opamps) sogar den damaligen Traum einer Schwingkreisabstimmung per Poti realisieren. Die dabei gemachten Überlegungen und Erfahrungen zeigt der Artikel auf.

Kondensatorersatz

Um die Schaltung zu verstehen wurde sie in LTSpice simuliert. In **Bild 2** sieht man den Schaltplan. Mit einem Widerstand und einer Spule wird ein einfaches RLC-Filter mit einem Schwingkreis nachgebildet.

In **Bild 3** sieht man den Frequenzgang, wenn man das Kleinsignalverhalten mit einem AC-Sweep simuliert.

Soweit sieht alles eigentlich hoffnungsvoll aus. Man sieht schön die Resonanzkurve. Die Schal-

Bild 3. Frequenzgang des RLC-Filters nach Bild 2.

tung simuliert, bedingt durch den Spannungsteiler R2/R3, einen Kondensator von $0,5 \times 10 \text{ nF}$. Das ergibt mit $L = 1 \text{ mH}$ rechnerisch eine Resonanzfrequenz von ca. 71 kHz, die in der Simulation gut getroffen wird.

Baut man die Schaltung jedoch auf, wird man eine herbe Enttäuschung erleben. Anstatt als selektives Filter zu wirken, schwingt sie munter vor sich hin. Das kann man mit Hilfe der transienten Simulation auch in Spice nachbilden.

In **Bild 4** sieht man das exponentielle Anschwergen als Simulationsergebnis. Auf den ersten Blick kann das verwirrend erscheinen, da die beiden Opamps hintereinander ja nur eine Verstärkung von insgesamt $v = 0,5$ realisieren. Aber Obacht! Der Kondensator C1 und die Spule L1 bilden in der Rückkopplung einen Serienschwingkreis, der die Ausgangsspannung des Opamps U2 um den Faktor der Güte Q anhebt, so dass die Schwingbedingung erfüllt ist. In der Kleinsignalsimulation kann Spice das prinzipbedingt nicht erkennen. Diese Beobachtung müssen wir unbedingt im Auge behalten: Die Kleinsignalsimulation ist eventuell nur die halbe Wahrheit, wenn wir schwingfähige Gebilde aufbauen.

Gyrator mit Opamps

Um Kondensatoren und Spulen zu simulieren, kann man auch Gyratoren verwenden. Dabei werden Gyratoren als Impedanzkonverter benutzt, und genau das werden wir nachfolgend auch tun. Allgemeiner kann man Gyratoren in Vierpolschaltungen anwenden.

Gyratoren haben in Elektor eine lange Tradition, frühe Schaltungen finden sich schon in der ersten Elektor [2]. In **Bild 5** ist das übliche Schaltbild eines Gyrators aus Operationsverstärkern dargestellt. Die simulierte Impedanz z ergibt sich als:

$$z = \frac{z_1 z_3 z_5}{z_2 z_4}$$

Verwendet man als z_2 beispielsweise einen Kondensator, und für die restlichen Impedanzen Widerstände, kann man eine Induktivität simulieren. Es ist dann:

$$z = j\omega \frac{R_1 R_3 R_5 C_g}{R_4} = j\omega L_g$$

Daraus ergibt sich:

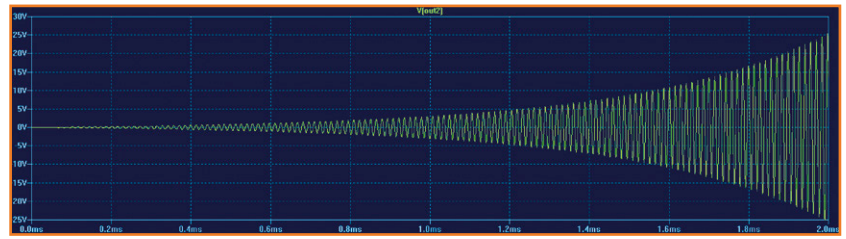


Bild 4. Die transiente Simulation zeigt die Instabilität.

$$L_g = K C_g \quad \text{mit} \quad K = \frac{R_1 R_3 R_5}{R_4}$$

Auch die Konfiguration in **Bild 6** ist ein Gyrator. Bei dieser Schaltung gilt:

$$z = \frac{z_1 z_3 z_5}{z_0 z_4}$$

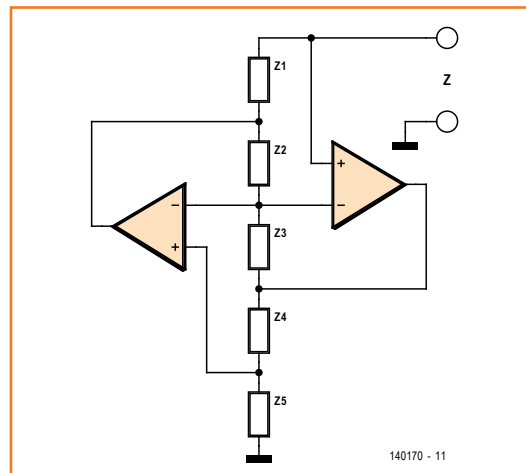


Bild 5. Gyrator mit Opamps.

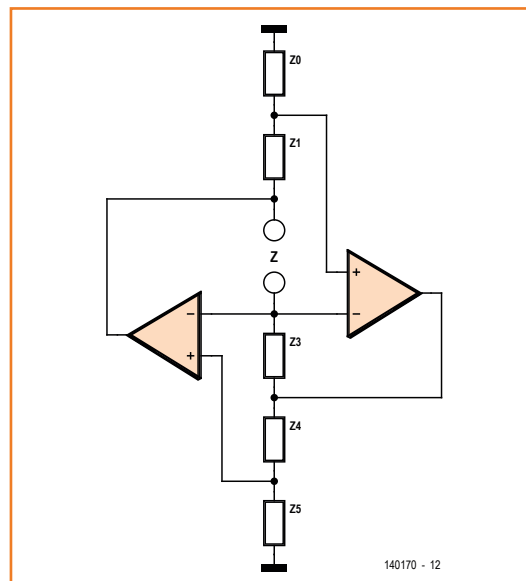


Bild 6. Alternative Gyratorschaltung.

Bild 7.
Aufbau eines RLC-Filters mit Gyrator.

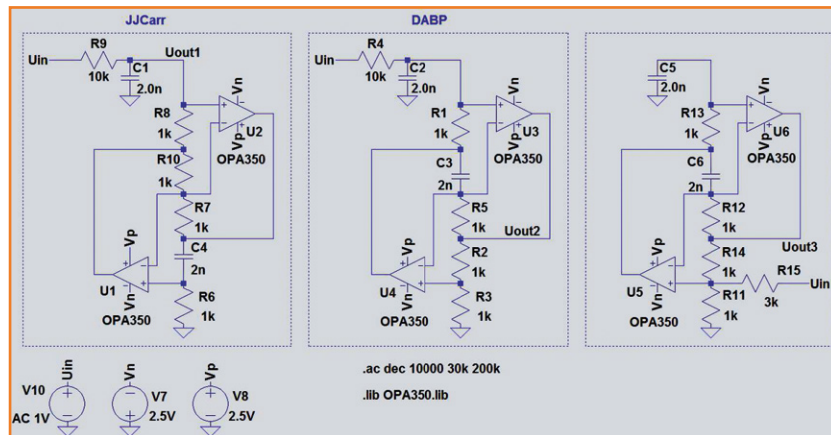
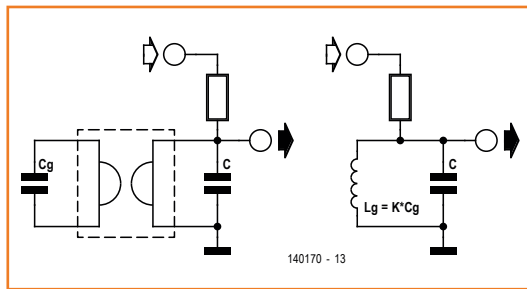
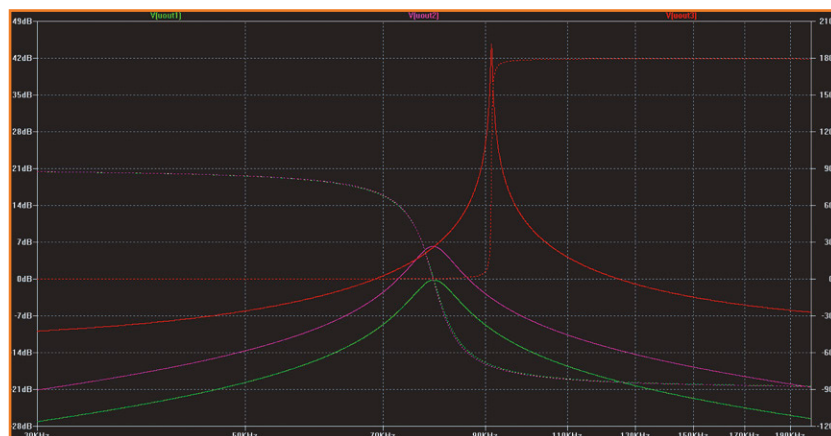


Bild 8.
Drei Gyrator-Varianten des RLC-Filters.

Mit einem weiteren Kondensator C und einem Widerstand kann man die Schaltung aus Bild 5 nun zu einem einfachen RLC-Bandpass (wie in **Bild 7** dargestellt) ergänzen. Links die Form mit Gyrator, rechts die äquivalente Schaltung. Die Gyratorkonstante K kann man nun mit den Widerständen einstellen. So lassen sich dann Kondensatoren oder Spulen simulieren, die mit einem Widerstand einstellbar sind.

Bild 9. Frequenzgänge der Schaltungen aus Bild 8 (Grün = Uout1, Lila = Uout2, Rot = Uout3).

Ändert man beispielsweise in Bild 7 die Gyratorkonstante, so ändert sich der Wert der simulierten



Induktivität L_g und damit die Resonanzfrequenz des aus C und L_g gebildeten Schwingkreises.

Simulation von Gyratorschaltungen

Filterschaltungen, die auf Gyrotoren aufbauen, sind natürlich altbekannt. In **Bild 8** sind drei Varianten als Simulation in LTspice dargestellt. Die linke Variante entspricht dem Filter in [3]. Der Gyrator simuliert eine Induktivität und damit wird ein LC-Schwingkreis aufgebaut, dem über einen Widerstand R9 die Eingangsspannung U_{in} zugeführt wird. Die Ausgangsspannung U_{out} wird am Parallelschwingkreis (hochohmig) entnommen. Üblicherweise folgt noch ein Impedanzwandler. Die Maximalverstärkung bei Resonanz ist 1 (0 dB) und die Güte wird entscheidend beispielsweise durch den Widerstand R9 bestimmt.

In der Mitte ist das sogenannte DABP (Dual Amplifier Band Pass) Filter aus [4] dargestellt. In der dortigen Darstellung ist dieses Filter gar nicht so leicht als Gyratorfilter wiederzuerkennen. Der Unterschied zur linken Schaltung ist die (niederohmige) Auskopplung am Ausgang des einen Opamps. Damit lassen sich auch Verstärkungen erreichen.

In der rechten Schaltung wird die Eingangsspannung U_{in} nicht mehr über einen Widerstand an den Schwingkreis geführt, sondern in den Gyrator selbst eingekoppelt. Diese Idee wird z.B. in [5] zur Kopplung von Gyratorfiltern angewandt. Damit wird die Güte nicht mehr durch den Einkoppelwiderstand erniedrigt. Man kann wesentlich höhere Güten und Verstärkungen erreichen. Im Beispiel erreicht man bei 90 kHz Mittenfrequenz ca. 300 Hz Bandbreite (3 dB), was immerhin einer Güte von 300 entspricht.

In **Bild 9** sind die Frequenzgangkurven für die drei Filterschaltungen wiedergegeben. Dass die rechte Schaltung eine höhere Resonanzfrequenz besitzt, liegt an R15, der kleinsignalmäßig für den Gyrator parallel zu R11 liegt. Durch eine Veränderung von R11 kann man das ausgleichen.

Praktische Erprobung

Nun soll die Schaltung aus Bild 8 (rechts) praktisch bei $f = 455$ kHz erprobt werden. Der zugehörige Schaltplan ist in **Bild 10** gezeigt. Es kommt ein Doppel-Opamp AD8042 mit einem Verstärkungs-Bandbreitenprodukt von $GBW = 160$ MHz zum Einsatz.

Für Gyrotorexperimente hat sich der Verfasser ein spezielles Steckbrett gelötet (**Bild 11**). Bei diesem können die einzelnen Impedanzen einfach

aufgesteckt werden, so dass man die verschiedenen Konfigurationen leicht durchspielen kann. In **Bild 12** sieht man den gemessenen Frequenzgang des Filters. Die Schaltung erreicht bei einer Mittenfrequenz von $f = 455 \text{ kHz}$ eine 3-dB-Bandbreite von 3 kHz. Die Güte liegt dabei bei circa $Q = 150$ und muss sich hinter LC-Filtern nicht verstecken. Sollte die Schaltung wegen parasitärer Kapazitäten zu Schwingungen neigen, hilft eine zusätzlich leichte Bedämpfung durch einen Parallelwiderstand zu C1. Damit ist gezeigt, dass man mit modernen Opamps durchaus Gyratorfilter hoher Güte im Mittelwellenbereich aufbauen kann. Es verbleibt noch unser ursprüngliches Ziel einer Ferritantenne, die mit einem Widerstand abstimbar ist.

Poti-abstimmbare Ferritantenne

Auf der Suche nach einem Drehko-Ersatz hatte der Verfasser den folgenden Gedanken: Wenn man mit einem Gyrator veränderbare Spulen und Widerstände realisieren kann, müsste man eigentlich auch eine abstimmbare Ferritantenne realisieren können. Eine erste Idee ist in **Bild 13** dargestellt.

Mit einer Festinduktivität L_g wird mit einem veränderlichen Gyrator die Schwingkreiskapazität C_g nachgebildet, die mit der Ferritantenne L_a einen Schwingkreis bildet. Aufgrund von Stabilitätsproblemen war diese Idee aber nicht umsetzbar. Eine Alternative besteht darin, mit dem Gyrator wieder eine veränderliche Induktivität L_g nachzubilden. Diese veränderliche Induktivität kann man dann zu der Ferritantenne parallel oder in Serie schalten, um die Resonanzfrequenz des Schwingkreises mit der Gyratorkonstanten K veränderlich zu machen. In **Bild 14** sind die Schaltungen dargestellt, links jeweils als Gyratorschaltung, rechts als äquivalente Schaltung.

Eine erste praktische Schaltung wurde für den DCF-Empfang dimensioniert. In **Bild 15** ist der Schaltplan dargestellt. Als Gyrator kommt die Variante nach Bild 6 zum Einsatz, wobei z_0 durch einen Kondensator C1 ersetzt wird. Die nachgebildete Induktivität bildet dann mit L_a und C2 einen Schwingkreis. Im Oszillogramm von **Bild 16** ist das DCF77-Signal mit den typischen Sekundenmarken erkennbar. Man konnte die Resonanzfrequenz mit R3 bequem einstellen und es wurde eine Bandbreite von 350 Hz erreicht. Damit ist die Umsetzbarkeit gezeigt.

In **Bild 17** ist eine andere Variante dargestellt. Diesmal liegt die nachgebildete Induktivität par-

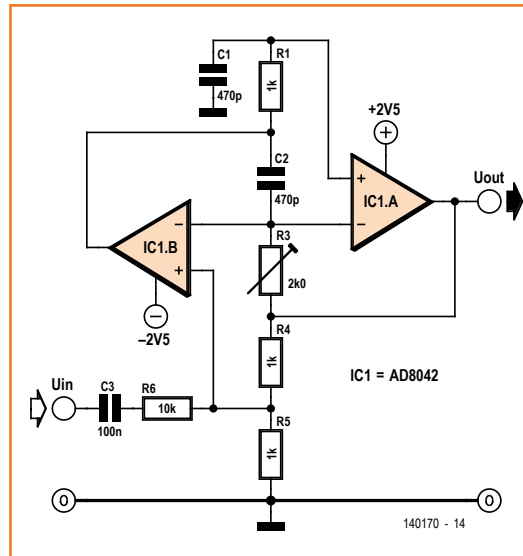


Bild 10.
Praktische Schaltung eines
455-kHz-RLC-Filters.

allel zu L_a und C1. Die Resonanzfrequenz ist zwischen 140 kHz und 200 kHz einstellbar. In **Bild 18** ist das empfangene Signal von TDF auf 162 kHz zu sehen. Auch die BBC auf 198 kHz konnte gut

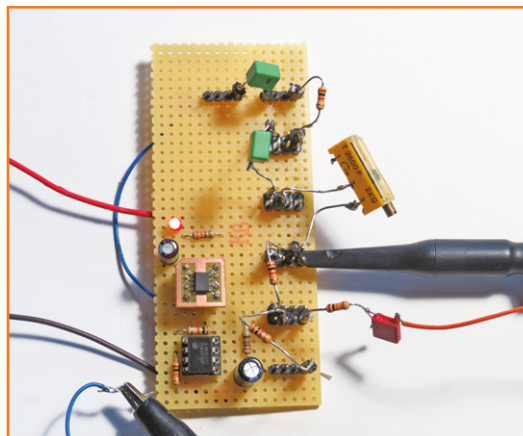


Bild 11.
Steckplatine für
Gyratorschaltungen.

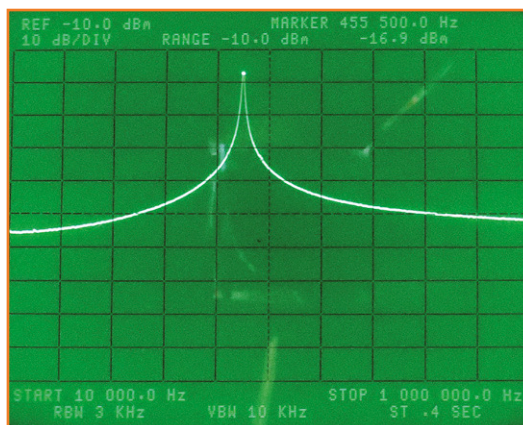


Bild 12.
Frequenzgang des 455-kHz-
Filters.

Bild 13.
Simulation einer
veränderlichen Kapazität.

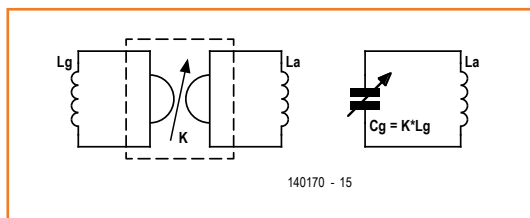


Bild 14.
Einsatz von veränderlichen
Induktivitäten.

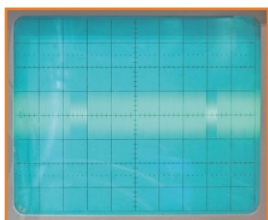
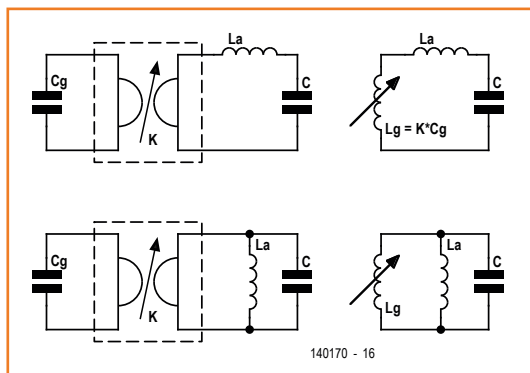


Bild 16.
DCF-Signal mit
Sekundenmarken.

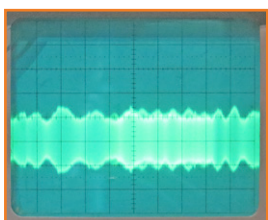


Bild 18.
Empfangenes TDF-Signal auf
162 kHz.

empfangen werden. Man erreicht eine Bandbreite von ca. 5 kHz. Mit diesen beiden Sendern konnten auch die Experimente aus dem SDR-Kursus [6] durchgeführt werden. Die erreichbare Güte hängt dabei entscheidend von den Opamp-Parametern ab. Insofern hilft vor dem Aufbau oft eine Simulation, welche die Stabilitätsfrage ebenfalls behandeln sollte.

Damit ist endlich das Ziel des Verfassers erreicht, den Drehkondensator bei Benutzung einer Ferritantenne durch ein Poti zu ersetzen.

(140170)

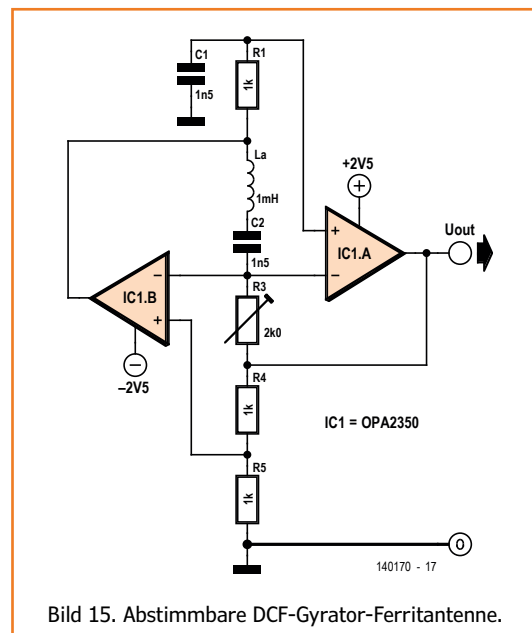


Bild 15. Abstimmbare DCF-Gyrator-Ferritantenne.

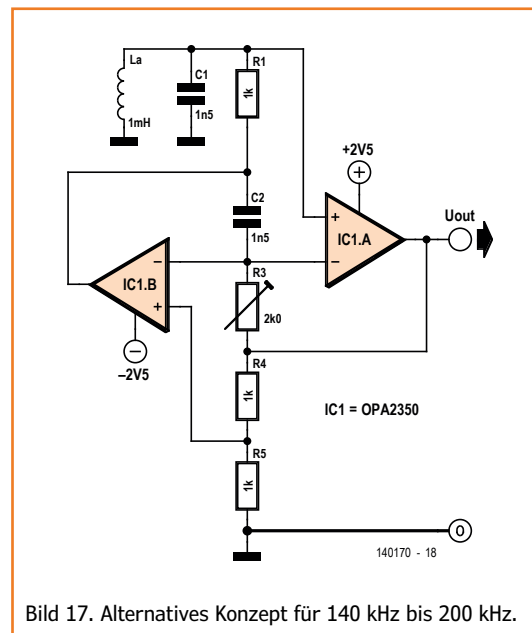


Bild 17. Alternatives Konzept für 140 kHz bis 200 kHz.

Literatur

- [1] Elektor November 1970, S. 40: variable Kapazität bis 100 uF
- [2] Elektor Mai 1970, S. 27: Der Gyrator, eine drahtlose Induktivität
- [3] J.J.Carr, Geheimnisse des HF-Schaltungsentwurfs, Elektor Verlag Aachen, S. 94
- [4] Linear Circuit Design Handbook, Analog Devices, Elsevier, S. 640
- [5] Lynch, T.H., The right gyrator trims the fat off active Filters, Electronics, July, 1977, S. 115
- [6] AVR Software Defined Radio, Elektor E-Book



1-tägiges Altium Elektor Seminar:

**„Tu Gutes und schreib darüber –
wie dokumentiere ich meine Leiterplattenentwicklung richtig“**

Altium

Ein nicht unerheblicher Teil der Arbeit eines Entwicklers wird von der Dokumentation dessen, was er macht, aufgefrisst. Muss das sein? Kann der Anteil dieser in der Regel anspruchsvollen, notwendigen aber auch nicht gerade kreativen Tätigkeit eventuell durch einfache Maßnahmen reduziert werden?

In diesem Seminar sollen gemeinsam Richtlinien und Vorlagen für eine gute Dokumentation unter Ausnutzung der Möglichkeiten eines State-of-the-Art EDA-Werkzeugs erarbeitet werden. Dabei werden sowohl der Schaltplan- als auch der Leiterplattenbereich berücksichtigt.

Die erarbeiteten Ergebnisse sind prinzipiell zwar unabhängig von der gewählten Entwicklungsplattform, die Umsetzung hingegen wird zugeschnitten auf die Möglichkeiten von Altium Designer erfolgen.

Veranstaltungsort/-termin: Köln, 14. Oktober 2014

Referent: Dipl. Ing. Christian Keller

Teilnahmegebühr: 149,00 Euro (inkl. MwSt.) für Elektor-Mitglieder (50% Rabatt!)

299,00 Euro (inkl. MwSt.) für Nicht-Mitglieder

Im Preis sind Mittagessen, Seminarunterlagen sowie Dokumentation und Aushändigung des Teilnahmezertifikats inbegriffen.

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

Elektor-Abomitmglieder erhalten 50% Rabatt auf den Seminar-Preis!

Elektor-Verlag GmbH | Süsterfeldstr. 25 | 52072 Aachen
Tel. +49 241 88 909-16

www.elektor.de
j.grotenrath@elektor.de

Workshop-DVD

Entwurf eines Röhrenverstärkers

NEU!

Wie entwirft man einen kompletten Röhrenverstärker mit nur einem aktiven, verstärkenden Element pro Kanal? Das war die Wettbewerbsaufgabe des European Triode Festivals, das 2012 in Berlin stattfand.

Röhren-Koryphäe Menno van der Veen bespricht in diesem Seminar seine sorgfältig ausgearbeiteten Entwürfe. Er geht dabei auf häufig auftretende Probleme bei Röhren und Trafos ein. Wie viel Ausgangsleistung kann man erwarten? Wie kann man Verzerrungen, die durch den Ausgangsstrom entstehen, minimieren? Wie findet man den besten Kompromiss zwischen maximaler Leistung und der Lebensdauer bei minimalen Verzerrungen? Ist Feedback nötig und warum?

Dieses Seminar gibt eine verkürzte, jedoch komplette Übersicht des einjährigen „TubeSociety“-Kurses wieder.



Inhalt:

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

ISBN 978-3-89576-284-0

€ 24,90
CHF 30,90

Weitere Infos & Bestellung unter www.elektor.de

Elektor-DVD 2013

Alle Artikel von 2013 auf DVD-ROM

NEU!

Die neue Elektor-Jahrgangs-DVD enthält alle Beiträge (mit Ausnahme aktueller Rubriken) des Jahrgangs 2013 der Elektor-Ausgaben in Deutschland, England, USA, Frankreich, Spanien und den Niederlanden. Sie verfügt über eine sehr übersichtlich gestaltete Benutzeroberfläche.

Mit der Elektor-DVD 2013 können Sie alle Platinenlayouts in perfekter Qualität drucken; diese Layouts mit einem Zeichenprogramm verändern; die Schnell-such-Funktion benutzen, mit der Sie in den einzelnen Artikeln oder im ganzen Jahrgang nach Wörtern, Bauteilen oder Titeln suchen können; Schaltbilder, Platinenlayouts, Illustrationen, Fotos und Texte exportieren.



Inhalt:

- Alle Elektor-Artikel des Jahrgangs 2013
- HTML-Benutzeroberfläche zum schnellen Auffinden des gesuchten Artikels

ISBN 978-90-5381-277-8

€ 27,50
CHF 34,10

Weitere Infos & Bestellung unter www.elektor.de

BK Precision BK560: Programmierbarer IC-Tester

Die glorreichen Tage von TTL- & CMOS-ICs



Von
Pascal Rondane
(F)

Wie bei *Hewlett Packard*, *Apple* und *Nirvana* begann auch die Erfolgs-Story der Firma *B&K Precision* in einer amerikanischen Garage. 1948 war das Jahr, in dem amerikanische Mittelschichtsfamilien begannen, Fernseher in größerem Umfang zu kaufen. In Chicago waren der Geschäftsmann Carl Korn und sein Partner Philippe Ban gerade sauer, weil es kaum Apparate zum Test elektronischer Bauteile für TV-Geräte gab. Also beschlossen sie, selbst welche zu bauen. Ihr Unternehmen nannten sie *Central Television Service Company*. Kurz darauf leiteten Korn und Ban eine profitable Firma, die Röhrentester und Regeneratoren für Bildröhren verkaufte. 1951 wurde die Firma in *B&K Precision Corp.* umgetauft und die Produktpalette im Bereich Testen und Messen verbreitert.

EST^d 2004

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

1961 wurde *B&K Precision* zur Tochter der Firma *DynaScan*. 60 Jahre später ist *B&K* ein global operierender Hersteller von besseren Messgeräten geworden. Interessanterweise firmiert sie auf ihrer eigenen Webseite unter *BK Precision*, *B&K Precision Corporation* und schließlich auch schlicht unter *BK*, gemäß der uramerikanischen Tradition, alles aufs Wesentliche zu konzentrieren und in Akronymen zu sprechen.

Herkunft

Bild 1 mit einer Anzeige für das Modell BK560 aus den 1980er Jahren gibt einen Eindruck von der damaligen Atmosphäre. Das hier besprochene Gerät habe ich im Jahre 2000 zusammen mit dem Transistortester BK502B anlässlich der Auflösung der Firma gekauft, bei der ich bis dahin arbeitete. Mein BK560 wurde noch jahrelang im Service für Drucker und Computer von IBM und Motorola sowie von anderer Industrielektronik eingesetzt. Heute können sich jüngere Servicetechniker gar nicht mehr vorstellen, wie die Technik damals ausgesehen hat. Der Terminus „embedded“ war noch nicht erfunden, doch gab es schon vielfach Mikroprozessoren wie die Typen Z80, 8052 und 68000, die von Unmengen an TTL- und CMOS-

Logik-ICs begleitet wurden. Dank unseres BK560 konnte mein Team sehr viele unterschiedliche ICs der Familien 74xx und 40xx testen.

Funktionen

Es gab zwei Modi: Das Testen von ICs „in-circuit“ oder „out-of-circuit“. Nebenbei: Mein BK560 produzierte so viel HF-Störstrahlung, dass der UKW-Empfang in seiner Nähe unmöglich war. Begriffe wie EMV wurden damals nicht ganz so ernst genommen.

Mein BK560 kannte 1.500 Bauteile – und der Speicher war erweiterbar, sodass man sogar als Anwender neue Bauteile hinzufügen konnte. Man konnte damit sehr gut Zähler und Flipflops, aber leider keine MMVs (74121, 74123, alles klar?) mit externen RC-Bauteilen oder Opamps testen. Ein In-Circuit-Test war nicht möglich, wenn ein Ausgang direkt mit einem Eingang verbunden war, was ja beim Flipflop oder in Taktschaltungen häufiger der Fall ist. Doch konnte man sich behelfen, indem man den Takt-Eingang abtrennte und die Learn-Methode nutzte. Auch hier gab es wieder zwei Modi:

1. Out-of-Circuit

Man gab auf der Tastatur des BK560 die gewünschte IC-Nummer ein, steckte das IC in den ZIF-Sockel und startete den Test. Auf dem Display wurde dann angezeigt, ob das untersuchte IC in Ordnung war oder nicht. Im letzten Fall wurde die Anzahl defekter Pins angezeigt.

2. In-Circuit

Hier gab es zwei Sub-Modi: Reines *In-Circuit* und *Learn*.

2a. In-Circuit. Man entfernt alle Taktquellen, schaltet die Stromversorgung des Boards mit dem IC ein und wählt am BK560 den richtigen IC-Typ. Nun kommen die Prüfspitzen an die IC-Pins und man startet den Test.

2b. Learn. Man entfernt alle Taktquellen, schaltet die Stromversorgung des Boards mit dem IC ein, wählt am BK560 den richtigen IC-Typ und startet den Test. Das Messgerät speichert die Messresultate für eine spätere Untersuchung und Vergleiche. In diesem Modus verliert man bei einem Stromausfall die Daten im Speicher, außer man hat einige EEPROMs der Organisation 16 Kbit (2 K x 8) in die Sockel an der Frontplatte des Messgeräts gesteckt (**Bild 2**).

Innenleben

Leider habe ich keinen Schaltplan des BK560. Man kann ihn aber für 35 \$ im Internet von einem BK-Vertragshändler [2] erwerben.



Wenn man ein BK560 öffnet, grüßen rechts drei doppelseitige Platinen. Auf der obersten Platine sitzt das flureszierende, grüne 20-Zeichen-Display, ein 8031-Controller, vier 2764-EPROMs und ein leerer Sockel für Software- oder Systemerweiterungen. **Bild 3** zeigt den Stapel Platinen.

Die meisten Steckverbinder sind mit Heißkleber fixiert. Noch zwei Details: In **Bild 4** sieht man ein 16-poliges Flachbandkabel mit einem Steckverbinder nur an einer Seite, das wohl das Routing der Platine vereinfachen sollte. Freut jeden Servicetechniker: **Bild 5** zeigt noch eine extrem preiswert gelöste Platinenaufhängung.

Auf der linken Seite gibt es noch eine Platine mit einem ZIF- bzw. Nullkraftsockel für die zu untersuchenden ICs und einen Pfostenstecker mit Auswurfhebeln zum Anschluss des Testkabels sowie vier Sockel für EEPROMs.

Im Netzteil stecken ein Mehrbereichs-Trafo (100...240 VAC für 50 oder 60 Hz) und einige der damals typischen Linearregler.



Test 1-2-24

Laut den technischen Unterlagen stehen insgesamt 24 bidirektionale Three-state-I/Os zum Anschluss an das zu testende IC zur Verfügung. Für die Tests von TTL-ICs wird eine Versorgungsspannung (V_{CC}) von 5,0 V angenommen. Eine logische „1“ entspricht $\geq 2,0$ V und eine logische „0“ $\leq 0,8$ V. Bei CMOS-ICs soll die Versorgungsspannung (V_{DD}) im Bereich 5...15 V liegen. Eine logische „1“ entspricht ≥ 70 % von V_{DD} und eine logische „0“ ≤ 30 % von V_{DD} (Hinweis: „ V_{DD} “ ist der richtige Terminus für die Versorgungsspannung von CMOS-Logik).

Auf der Rückseite findet sich eine 25-polige Sub-D-Buchse für RS232. Ich kann nur vermuten, dass

ein BK560 ursprünglich mit einer Floppy-Disk ausgeliefert wurde, auf der sich Software für weitere ICs befand, die nicht in der Datenbank im Speicher enthalten waren. Wahrscheinlich umfasste dieses Upgrade ein PC-Verbindungskabel, ein Handbuch und ein EEPROM für Updates. Das Software-Handbuch spricht von einem „IBM Computer“ oder einem Kompatiblen mit mindestens 250 KB Speicher, einem 5¼“-Diskettenlaufwerk und DOS 2.00 oder höher – eben voll 80er Jahre. Das Programm für den PC unterstützt das Kreieren von Test-Prozeduren für neue oder unbekannte ICs. Anschließend werden diese Prozeduren auf den BK560 übertragen und dort permanent in einem EEPROM abgelegt. Mit der Software kann man die Anzahl an Pins und die Technologie (TTL oder CMOS) festlegen. Bei Open-Collector-Ausgängen wird noch ein 10-k Ω -Pullup-Widerstand aktiviert. Ein sequentieller Test erlaubt die Überprüfung individueller Gatter und Logik-elemente in einem Logik-IC. Außerdem gibt es auch eine Reset/CLR- bzw. RST/CLR-Funktion für Zähler, Flipflops etc. Ich kann mich nicht erinnern, sie je genutzt zu haben.

Ich bin mir zwar nicht völlig sicher, aber höchstwahrscheinlich wurde mein BK560 noch in den USA hergestellt.

Fazit

Auch heute noch nutze ich diesen Tester aus den 1980ern gelegentlich für die Reparatur alter Platinen mit TTL- oder CMOS-Logik-ICs. Das Messgerät war mir viele Jahre eine große Hilfe. Das einzige in dieser Zeit auftauchende Problem war die nachlassende Kontaktqualität des Kabels für den In-Circuit-Test, da das externe Testen wohl gelegentlich mit zu viel Schwung vonstattenging. Ich stocherte etwas am Lack auf den Lötstellen der doppelseitigen Platine herum und verlötete etliche Pins hier und da neu. Ansonsten gab es keine ernsthaften Schwierigkeiten.

Meine originalen Dokumente zum BK560 sind in englischer und französischer Sprache vorhanden, aber sie enthalten leider keine Schaltung. Wer daran Interesse hat, der kann mich gerne über editor@elektor.com kontaktieren.

(140051)

Weblinks

[1] Webseite von BK Precision:
www.bkprecision.com

[2] BK560-Manuals: www.bkmanuals.com

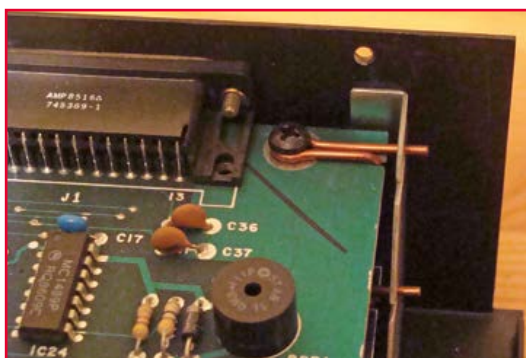
3



4



5





Kundenbewertungen



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

*Quelle: Shopauskunft.de (3. 6. 2014)

ALLES FÜR
WERKSTATT &
LABOR

ZU EINEM
TOP-PREIS!



KONTAKT CHEMIE

... und die Elektronik funktioniert.

Kontakt 60, 100ml Oxidlösender Kontaktreiniger

Entfernt störende Korrosionsschichten und Schmutz von Kontaktmaterialien

KONTAKT 2010

3,85



Druckluft 67, 400ml Unbrennbares Druckgas

Zum berührungslosen Entfernen von Staub in Feinmechanik, Optik und Elektronik

KONTAKT 334

14,20



Kälte 75, 200ml Standard-Kältespray

zur thermischen Fehlersuche in der Elektronik, zum Kälteschrumpfen von kleinen Teilen usw.

KONTAKT 316

8,35



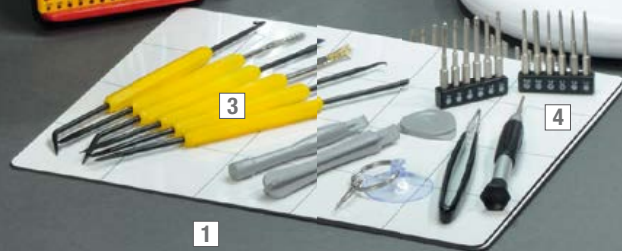
Praktische Helfer für den Elektroniker

LED Tisch-Lupenleuchte mit 100 mm Linse

- Glas-Linse mit 3 Dioptrien und Schutzkappe
- 1,75 fache Vergrößerung
- energiesparend durch aktuelle LED-Technik (nur ca. 5 Watt)
- Sofortstart unter 0,1 Sekunden
- blend- und schattenfreie Ausleuchtung
- fester, stabiler Standfuß

LL ZD-2012B-2H

29,95



1 Magnet-Arbeitsmatte rutschfest, 20 x 25 cm

- einfache Organisation von Kleinteilen
- reduziert die Montagezeit nach der Reparatur um bis zu 40%
- Schrauben und Muttern rollen nicht weg
- beschriftbar für leichte Zuordnung

MAGNETMATTE

5,95

2 Präzisions-Mini-Bits in praktischer Kassette

mit „easy-Grip“ Schraubendrehergriff und Verlängerung

MICRO BITSET 9,95

3 Löt- Hilfsmittel-Set 6-teiliges Platinebesteck

SOLDER AID TOOLS 4,95

4 Praktisches Werkzeugkit für die Wartung von Smartphones & Tablet-PCs

HANDY WZ KIT 14,95

Die „helfende Hand“ praktischer Platinenassistent

Ihre Hilfe bei filigranen Lötarbeiten, beim Verzinnen oder Kleben – immer wenn Sie beide Hände benötigen.



HALTER ZD-10D

2,95

Jetzt bestellen: www.reichelt.de

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

Katalog
06.1/2014

Kostenlos –
Jetzt anfordern!



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

Preisstand: 26. 08. 2014
Tagesaktuelle Preise:
www.reichelt.de

Hexadoku Sudoku für Elektroniker

Herbst... Zeit um draußen wieder alles aufzuräumen und drinnen etwas auszuruhen. Und wenn man dann auf einem bequemen Stuhl sitzt, macht es natürlich Laune, unser neues Hexadoku griffbereit zu haben. Machen Sie mit und versuchen Sie, die Zahlen in den grauen Kästchen herauszufinden. Wie immer gibt es schöne Bücher-Gutscheine zu gewinnen!

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

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

Wer das Rätsel löst - sprich die Zahlen in den grauen Kästchen herausfindet - kann einen von fünf Buchgutscheinen im Wert von 50 Euro gewinnen!

Einsenden

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

Elektor - Redaktion - Süsterfeldstr. 25 - 52072 Aachen

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

Als Betreff bitte nur die Ziffern der Lösung angeben!

Einsendeschluss ist der 31. Oktober 2014!

Die Gewinner des Hexadokus aus der Juli/August-Ausgabe stehen fest!

Die richtige Lösung ist: **CE234**.

Einen Elektor-Buchgutschein über je 50 € haben gewonnen:

Ciril Zalokar, Jacek Butowski, Pascal Schmitz, Chris Smith und Brian Wood.

Herzlichen Glückwunsch!

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

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

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

Professionelle Qualität zu attraktiven Preisen!

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

3D-Drucker

MAKERBOT® REPLICATOR®

Ausgefeilte 3D-Drucktechnik der 5. Generation

- ✓ 3,5"-LC-Display
- ✓ integrierte Kamera
- ✓ leicht wechselbarer Extruder
- ✓ 4 Anschlussmöglichkeiten:
USB-Stick, USB-Kabel, LAN-Kabel oder WLAN



Die wichtigsten Neuigkeiten der 5. Generation:

- vergrößerter Bauraum (+ 11%): 25.2 x 19.9 x 15.0 cm
- schnellere Druckzeiten
- App, Cloud, Ethernet und WLAN fähig
- LCD Bildschirm (ca. 9 cm)
- integrierte Kamera
- unterstützte Bauplattform-Justierung
- MakerBot® Replicator® Smart Extruder

REPLICATOR 5G

3190,-

Verbrauchsmaterial & Zubehör finden Sie online:



Ultimaker 2

Profi-Qualität für Prototypen und 3D Modelle.



- ✓ beheiztes Druckbett, neues Leveliersystem
- ✓ Verbesserter Extruder
- ✓ flüsterleiser 3D Druck
- ✓ Raspberry Pi WLAN-Druck

Entwickelt für hervorragende Leistung, Zuverlässigkeit, Geschwindigkeit und Genauigkeit!

Technische Daten:

- Druckgröße: max. 230 x 225 x 205 mm
- Schichtdicke: 0,02 mm
- Druckmaterial: PLA, ABS, LAYWOOD, LAYBRICK, Nylon, Flexible
- Software: Cura Open Source
- SD-Karten-Slot



2299,-
ULTIMAKER 2

Der Ultimaker² vollbringt
Höchstleistungen:
Druckgeschwindigkeit:
bis zu 300 mm/s
Schichtauflösung: 0,02 mm!

3D-Drucker-Bausatz

Velleman K8200 inkl. 1kg Filament

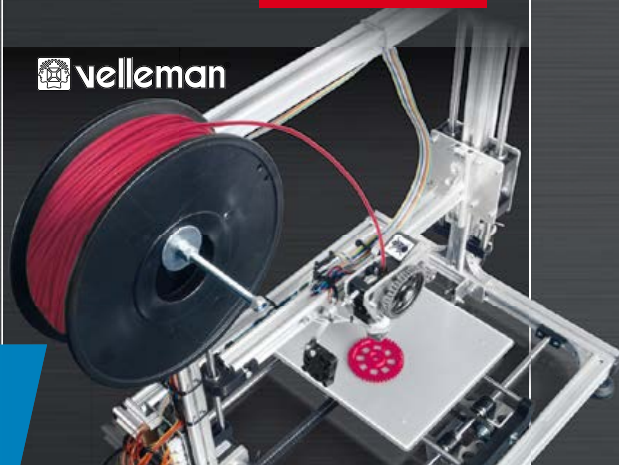
- ✓ kostengünstiges Einstiegsmodell
- ✓ extrem schnell & zuverlässig
- ✓ druckbare Objektgröße: 20 x 20 x 20 cm
- ✓ Schichtdicke: 0,2-0,25 mm

Träumen Sie davon Werkstücke und Modelle selbst zu fertigen? Sind Sie begeisterter Modellbau-Enthusiast und bauen Ihre Modelle gerne komplett in Eigenregie? Dann ist der kostengünstige 3D-Drucker von Velleman das perfekte Werkzeug.

% bisher
699,-

**1 kg PLA-Filament
GRATIS!** (schwarz)

499,-
3D PRINTER K8200



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

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

Katalog
06.1/2014!
Kostenlos –
Jetzt anfordern!



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

Preisstand: 26. 8. 2014
Tagesaktuelle Preise:
www.reichelt.de



Mit Visual Basic in die analoge Welt

1 VB-Express und die Hardware

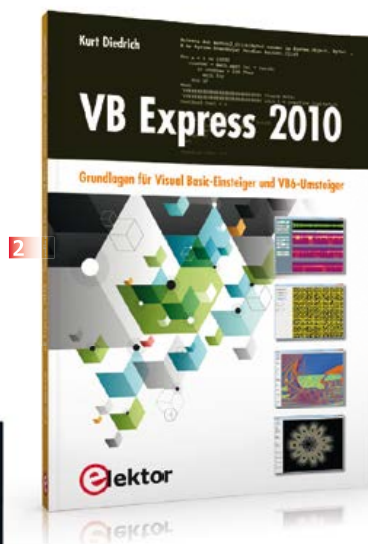
Visual Basic zählt nach wie vor zu den sehr weit verbreiteten Programmiersprachen. Seine Beliebtheit resultiert gerade für den Einsteiger aus der schnellen Erlernbarkeit und der einfachen Lesbarkeit des Programmcodes. Dieses Buch ist für Einsteiger in die Programmiersprache Visual Basic.NET gedacht,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

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

Grundlagen für Visual Basic-Einsteiger und VB6-Umsteiger

2 VB Express 2010

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



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

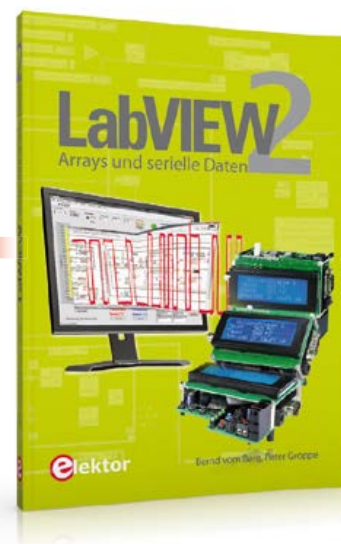
284 Seiten (kart.) • ISBN 978-3-89576-269-7
€ 34,80 • CHF 43,95

45 Experimente mit Hard- und Software für Elektroniker

3 Raspberry Pi

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

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



Arrays und serielle Daten

4 LabVIEW 2

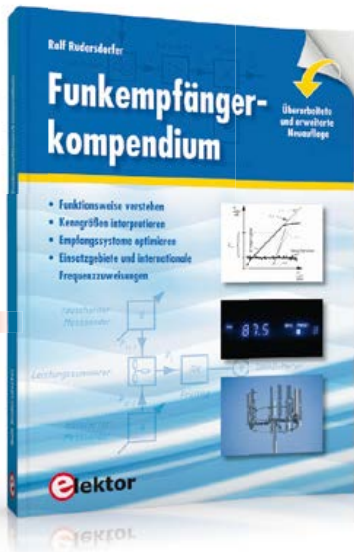
Der zweite Band der LabVIEW-Lehrbuchreihe beschäftigt sich u.a. mit Arrays, Cluster und den seriellen VISA-Funktionen. Als Erstes werden vier neue zusammengesetzte Datentypen (Enum, Ring, Array, Cluster) vorgestellt und deren Verwendung wird anhand zahlreicher praktischer Beispiele und Übungen erläutert.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

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

Der professionelle Ratgeber für Funkempfangstechnik

5 Funkempfänger-kompodium

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

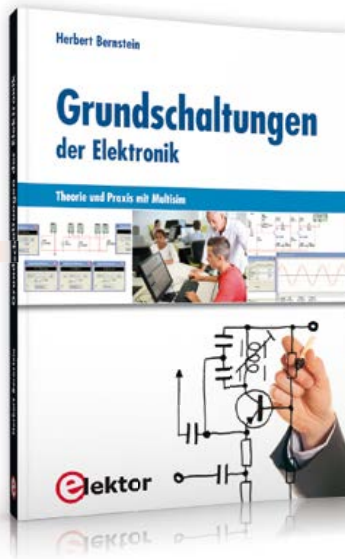


auch ausgefallene Anwendungen von Empfängern kennenlernen und wissen, wie ein Software Defined Radio (Digitalempfänger) nun wirklich funktioniert und was der letzte Stand der entsprechenden Technik kann? In diesem Buch findet man die Antworten!
397 Seiten (geb.) • ISBN 978-3-89576-276-5
€ 49,00 • CHF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

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

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

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



Schnell und einfach mit Arduino und Elektor-Shield

7 Mikrocontroller verstehen und anwenden

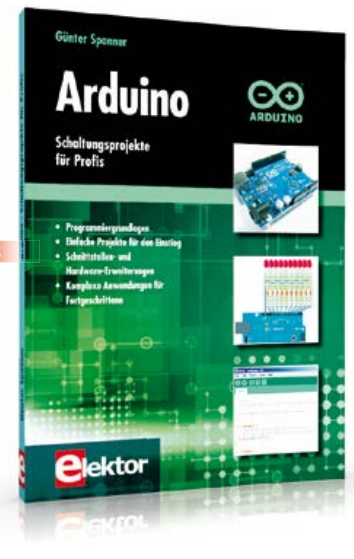
Mit diesem Buch erweitert der Anfänger auf dem Gebiet der Mikrocontroller, der Arduino-User bzw. -Enthusiast seine Mikrocontroller-Kenntnisse auf Grund eigener Erfahrungen und Erfolgserlebnisse und wird dazu noch ganz nebenbei in die Welt des Arduino und seiner Entwicklungsumgebung eingeführt.

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

Schaltungsprojekte für Profis

8 Arduino

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



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

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

Weitere Informationen zu unseren Produkten sowie das gesamte Verlagssortiment finden Sie auf der Elektor-Website:

www.elektor.de

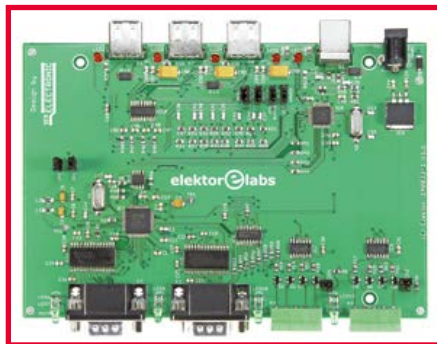
Elektor-Verlag GmbH
Süsterfeldstr. 25
52072 Aachen
Tel. +49 (0)241 88 909-0
Fax +49 (0)241 88 909-77
E-Mail: bestellung@elektor.de

●Nächsten Monat in Elektor



Labornetzgerät 40 V/2 A

In der Vergangenheit haben wir mehrere Labornetzteile veröffentlicht, mit ganz unterschiedlichen Eigenschaften. Die nächste Ausgabe bringt eine weitere Entwicklung aus dem Elektor-Labor: Von einem Mikrocontroller gesteuert kommt sowohl schaltende als auch analoge Technik zum Einsatz. Entwicklungsziel war diesmal, eine störfreie Ausgangsspannung, optimiertes Regelverhalten und exzellenten Wirkungsgrad miteinander zu vereinen.



USB-Hub mit RS232/RS485

Schon in der letzten Vorschau angekündigt, aber aus technischen Gründen auf die nächste Ausgabe verschoben: Unser USB-Hub ist universell ausgelegt. Drei USB-Ports werden von zwei voll duplex arbeitenden RS232-Schnittstellen und zwei RS422/485-Bussen unterstützt. Denn viele bewährte periphere Geräte, die durchaus noch auf dem Stand der Zeit sind, nutzen immer noch die serielle RS232-Schnittstelle zur Kommunikation.



Software-Module in C

Das Elektor Extension-Shield für den Arduino Uno bringt ein Display, LEDs und Taster mit, an Steckverbinder lassen sich Relais-, Funk- und weitere Module anschließen. Für das Shield und die Erweiterungen liefern wir nun C-Code auf Basis der EFL. Zu jedem Hardware-Modul gehört dabei auch ein Software-Modul; die Einzelteile muss man nur noch zusammenfügen, um zu einem funktionierenden Prototypen zu kommen.

Änderungen vorbehalten. Elektor November 2014 erscheint am 22. Oktober 2014. Verkaufsstellen findet man unter www.pressekaufen.de.

Rund um die Uhr und
sieben Tage die Woche

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

elektorlabs

Sharing Electronics Projects

Home Proposals In Progress Finished

Log in

Mobility supercharges everything. Live for Super. Mobility Week powered by CTA

9.9.2014 SANDS EXPO LAS VEGAS REGISTER

Join Elektor.Labs Now!

Choose your language to join:
English / Deutsch / Français / Nederlands

Create a Project

Create a new project or enter a proposal

Get help, feedback & votes from other visitors, and maybe you will get Elektorized too!

Not a member?

You want to post a project but you are not a member? You can! Click here to send a description of your project including a circuit diagram and a photograph for evaluation and maybe you will be granted free access!

Challenges

IR Remote Control Learning Dimmer or Heat Control [140279-1]

Author: Juan Canton | Project status: Proposal | 0 contributions | 3 members | 0 comments | 1341 views

IR Remote Control Learning Dimmer or Heat Control

This circuit has the following features:

- 1.- It works with main frequencies of 50 or 60 Hertz.
- 2.- It works with main voltages of 117 or 220 VAC.
- 3.- It is connected in series with the load and can replace any conventional switch without any modifications.
- 4.- It works with only one switch.
- 5.- It can control the brightness level of any heat element, incandescent or alogen lamp.
- 6.- It has the ability to learn up to four control codes of a common infrared control.

Circuit description:

look for the file description.zip in others files

Alle Elektor-Artikel der Jahre 2000 bis 2009 auf DVD!



NEU!

**Ein Muss
für jeden
Elektor-Leser!**

ISBN 978-3-89576-292-5
€ 89,00 • CHF 110,40

Jetzt unter www.elektor.de/2000-2009 bestellen!

Lost Something? Never Again!

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

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

Inside The Technology

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

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

Remotely Controlling Operations

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

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

The Possibilities?

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



We have a worldwide distribution and support network

www.trace.me

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