

eLektor

www.elektor.de

Arduino

Kombi-Kommunikations-Shield

WLAN / Bluetooth / USB für Platino und Arduino

+ **Embedded Linux**
Ein Blick in den Kernel

+ **Elektronik-Grundlagen**
Verstärker für NF-Signale

+ **Stromzähler im Netz**
Energieverbrauch erfassen mit Open-Source-Lösung

Strahlungsmesser II

Mehr Sensoren,
mehr Funktionen





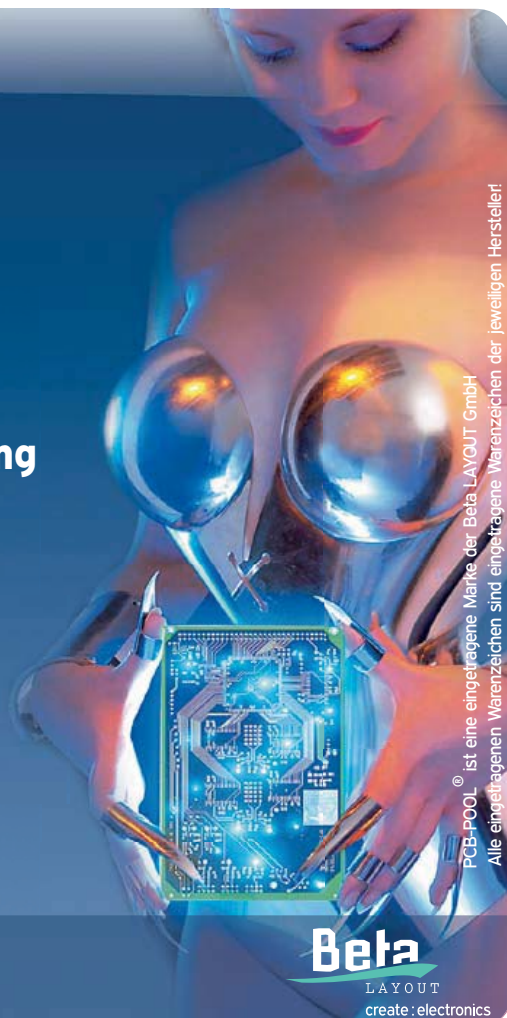
DAS ORIGINAL SEIT 1994
PCB-POOL
 Beta LAYOUT

FREE Stencil
 bei jeder PCB Prototyp-Bestellung



Easy-going
 17 akzeptierte Layoutformate

www.pcb-pool.com



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

Beta
 LAYOUT
 create: electronics



eSTORE
 Beta LAYOUT

Entwickeln, Bestücken und Löten



Arduino Mega (ATMega 1280-16AU)
 kompatibel

Reflow-Controller



**LED Wechselblinker
 SMD-Bausatz**



Big Beta-Reflow-Kit



Tool-Kit Extended



* inkl. MwSt. und zzgl. Versandkosten

www.beta-eSTORE.com

Beta
 LAYOUT
 create: electronics

ein SOCKEL für 30 click boards



Stecken Sie einfach
das **Click-Board™**
Ihrer Wahl in den
fortschrittlichen
neuen **mikroBUS-**

Sockel, und schon kann es los gehen.

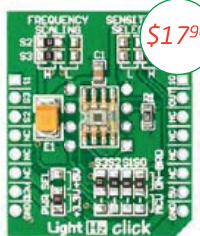
Ihre Hardware last sich so besonders
einfach um **neue Funktionen** erweitern.



\$49⁰⁰



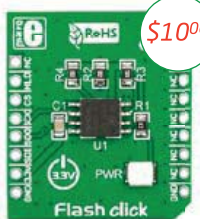
\$18⁰⁰



\$17⁹⁰



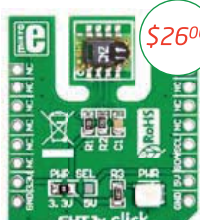
\$12⁰⁰



\$10⁰⁰



\$16⁹⁰



\$26⁰⁰



\$25⁹⁰



\$9⁰⁰



\$16⁰⁰



\$45⁰⁰



\$21⁰⁰



\$41⁹⁰



\$21⁰⁰



\$29⁰⁰



\$21⁹⁰



\$8⁰⁰



\$21⁰⁰



\$29⁰⁰



\$13⁹⁰



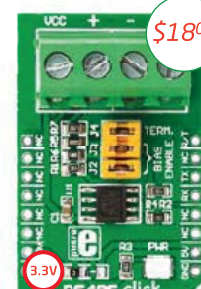
\$10⁰⁰



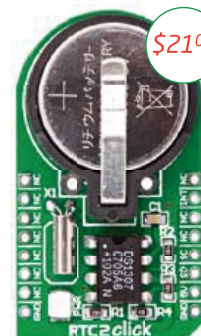
\$18⁰⁰



\$13⁹⁰



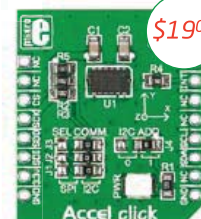
\$18⁰⁰



\$21⁰⁰



\$44⁰⁰



\$19⁰⁰



\$24⁰⁰



\$21⁰⁰



\$24⁰⁰



Elektronik im WWW

Immer mehr Geräte besitzen mittlerweile ein Netzwerk-Interface, mit dem ein Zugang ins WWW möglich wird. Einen Internet-Anschluss vorausgesetzt, kann man dann von überall auf der Welt mit dem Apparat Kontakt aufnehmen. Auch ihr persönliches Lieblingsprojekt benötigt vielleicht noch einen solchen Zugang? Sicherlich haben auch Sie hin und wieder ein besonders kreatives halbes Stündchen, in dem plötzlich tausend Ideen kommen, was man alles aus der Ferne steuern, messen und überwachen könnte! Wenn dem so ist, dann wollen wir Sie in diesem (und den kommenden) Heften gerne wieder einmal unterstützen. Unser „Multicom-Shield“ (Seite 12) kann mit einem WLAN-Modul ausgerüstet werden – einfacher als kabellos kommt man nicht ins Internet. Das Shield passt übrigens nicht nur (wie der Name vermuten lässt) auf Arduino-Boards, sondern auch auf die AVR-Platine „Platino“. Und es lässt sich auch einfach als „WLAN/Seriell-Konverter“ in Verbindung mit eigener Elektronik einsetzen. Der Artikel „Stromzähler im Netz“ (Seite 30) zeigt ein weiteres Beispiel der Fernüberwachung über das WWW. Wie immer kann man das Projekt 1:1 verwenden, aber auch modifizieren oder einfach Anregungen und Tricks für eigene Entwicklungen herausziehen. Die Folge der Serie „Arduino auf Kurs“ in diesem Heft sollten Sie sich ebenfalls anschauen: Falls einmal kein Internet verfügbar ist, wird hier gezeigt, wie man Daten per SMS versenden kann. Übrigens habe ich unseren Abo-Plus-Mitgliedern ja noch einen besonderen Service versprochen, der zum Fernsteuern von Elektronik genutzt werden kann. Wir werden eine Schnittstelle und etwas Speicher „in der Cloud“ bereitstellen; hiermit kann man Geräte (etwa die eigene Schaltung und ein Smartphone) miteinander kommunizieren lassen kann. Natürlich wird die benötigte Software wie immer „open source“ sein – so kann man sich solch einen Service auch selbst einrichten. Mehr unter www.elektor-projects.com!

Jens Nickel

6 Impressum

Who is who bei Elektor

8 Aktuelles

Nachrichten und Termine

12 Multicom-Shield

Das „Multicom-Shield“ stellt wahlweise eine WLAN- oder Bluetooth-Funktion zur Verfügung.

20 Embedded Linux leicht gemacht (4)

Am Beispiel eines USB/Seriell-Konverters zeigen wir, wie man einen Gerätetreiber selbst in den Kernel integriert.

30 Stromzähler im Netz

Unser Gateway besitzt vier S0-Eingänge für Stromzähler. Um die Daten zu speichern und anzuzeigen, wird auf eine Open-Source-Lösung zurückgegriffen.

36 Arduino auf Kurs (2)

Zusammen mit der spanischen Telefongesellschaft Telefonica hat der Autor ein Mobilfunk-Shield für Arduino-Boards entwickelt. Wir zeigen, wie man mit wenigen Codezeilen zum Beispiel Daten per SMS verschickt.

44 Labcenter

Plötzlicher Reset
E-Verschmutzung!
Lieblings-Bauteile
RS485-Massen

48 Bauelemente-Tipp

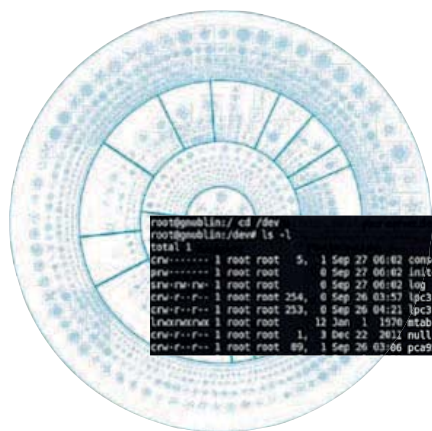
Wir stellen zwei Chips vor, die Energie aus der Umwelt auf unkonventionellen Wegen gewinnen.

50 Strahlungsmesser aufgebohrt

Spezielle Aufgaben lassen sich oft mit kleinen Änderungen des Basisprogramms lösen.

56 Motorrad-Alarm

Das Prinzip dieser Alarm-Anlage ist ganz einfach: Ist das Motorrad einmal auf dem Ständer abgestellt, geben drei Quecksilberschalter eine Lage-Information an einen Mikrocontroller.



INHALT

43. Jahrgang
Oktober 2012
Nr. 502

12 Multicom-Shield WLAN/Bluetooth/USB-Kombi-Shield für Platino und Arduino

Das „Multicom-Shield“ ist ein sehr vielseitiges Board. Bestückt mit einem WLAN-Modul kann es elektronische Geräte ins Internet bringen. Mit einem Bluetooth-Modul nimmt es Verbindung mit anderen bluetooth-fähigen Gerätschaften auf; per USB/Seriell-Konverter kann es auch mit älteren Apparaten seriell kommunizieren. Das Board passt auf ein Mikrocontroller-System mit arduino-kompatiblen Erweiterungs-Steckern, also auf Arduino-Boards oder „Platino“ von Elektor. Es lässt sich aber auch alleine als Erweiterungs-Modul für einen PC einsetzen.

20 Embedded Linux leicht gemacht (4) Ein Kernel-Überblick

In den letzten Folgen der Serie haben wir die Toolchain, den Kernel und Bootloader sowie ein Standard-Dateisystem zum Laufen gebracht. In unserer Artikelreihe soll dem Entwickler aber auch gezeigt werden, was „unter der Oberfläche“ passiert. Ein weiteres Thema dieser Folge sind Gerätetreiber, die man selbst in den Kernel integrieren kann. Denn auch im Embedded-Bereich nimmt Linux dem Programmierer viel Arbeit ab: für Netzwerk, USB, Bluetooth und mehr muss man nicht mehr aufwendig eigene C-Lösungen entwickeln.

30 Stromzähler im Netz Gateway für Stromzähler mit S0-Bus

Eine zentrale Aufgabe der Menschheit in naher Zukunft wird es sein, den CO₂-Ausstoß drastisch zu verringern. Zuvorderst muss dazu ermittelt werden, wieviel und an welcher Stelle elektrische Energie „verheizt“ wird. Was spricht dagegen, eine clevere Technologie und das WWW zu Hilfe zu rufen, um den Stromverbrauch zu reduzieren? Das hier vorgestellte Board besitzt vier S0-Eingänge für Stromzähler. Um die Daten zu speichern und anzuzeigen, wird auf eine Open-Source-Lösung zurückgegriffen.

50 Strahlungsmesser aufgebohrt Eigene Modifikationen erweitern die Möglichkeiten

Der Strahlungsmesser aus Elektor 11/2011 ist bei vielen Lesern im Einsatz. Das Gerät hat sich vor allem bei Langzeitmessungen und bei der Untersuchung schwach radioaktiver Proben bewährt. Das Messgerät ist als Bausatz erhältlich; der Mikrocontroller ATmega88 ist bereits mit einem Bootloader ausgestattet. Damit ist es einfach, eigene Programme in den Controller zu laden. Aber auch mit dem LötKolben lässt sich das eine oder andere sinnvoll verändern.

59 Entwicklungstipp
PIC-Programmer für Notfälle

60 SDR mit AVR (6)
In dieser abschließenden Folge der Serie schauen wir uns weitere Decodier-Verfahren an.

66 Stromwandler
Professionelle Stromwandler werden oft mit hoher Präzision gewickelt. Dieser Beitrag zeigt, wie man Stromwandler selbst aus handelsüblichen Teilen herstellen kann. Damit lassen sich dann sichere Messungen an Stromversorgungen vornehmen.

68 Transkonduktanz-Messverstärker
Ein Transkonduktanz-Verstärker ist ein Differenzverstärker, dessen Ausgangsstrom eine Funktion der Spannungsdifferenz zwischen den Eingängen ist. Mit diesem Prinzip kann ein vielseitiger Vorverstärker für den Einsatz in der Messtechnik aufgebaut werden.

74 Zurück zu den Wurzeln (8)
Eines der wichtigsten Einsatzgebiete des Transistors ist die Verstärkung von Tonsignalen.

78 PicoScope 2205-MSO unter der Lupe
Ein MSO (Mixed Signal Oscilloscope) kombiniert ein Oszilloskop mit einem Logic-Analyser. Wir testen das Einsteiger-MSO 2205 von Pico Technology.

80 Retronik
Signalgenerator TF801D/1 AM RF von Marconi

82 Hexadoku
Sudoku für Elektroniker

86 Vorschau
Nächsten Monat in Elektor

Aus technischen Gründen mussten wir den in der letzten Vorschau angekündigten Artikel „Der 7805 – neu erfunden!“ auf die nächste Ausgabe verschieben.

Unser Team

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

Unser Netzwerk



Internationale Teams

 Großbritannien Wisse Hettinga +31 46 4389428 w.hettinga@elektor.com	 Spanien Eduardo Corral +34 91 101 93 95 e.corral@elektor.es	 Indien Sunil D. Malekar +91 9833168815 ts@elektor.in
 USA Hugo Vanhaecke +1 860-875-2199 h.vanhaecke@elektor.com	 Italien Maurizio del Corso +39 2 66504755 m.delcorso@inware.it	 Russland Nataliya Melnikova +7 (965) 395 33 36 Elektor.Russia@gmail.com
 Deutschland Ferdinand te Walvaart +49 241 88 909-0 f.tewalvaart@elektor.de	 Schweden Wisse Hettinga +31 46 4389428 w.hettinga@elektor.com	 Türkei Zeynep Köksal +90 532 277 48 26 zkoks@beti.com.tr
 Frankreich Denis Meyer +31 46 4389435 d.meyer@elektor.fr	 Brasilien João Martins +55 11 4195 0363 joao.martins@editorialbolina.com	 Südafrika Johan Dijk +27 78 2330 694 / +31 6 109 31 926 J.Dijk@elektor.com
 Niederlande Harry Baggen +31 46 4389429 h.baggen@elektor.nl	 Portugal João Martins +351 21413-1600 joao.martins@editorialbolina.com	 China Cees Baay +86 21 6445 2811 CeesBaay@gmail.com

IMPRESSUM

43. Jahrgang, Nr. 502 Oktober 2012
Erscheinungsweise: 11 x jährlich (inkl. Doppelheft Juli/August)

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

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

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

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

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

Die Elektor Community

274932

Mitglieder in

83

Ländern...

Noch kein Mitglied?

www.elektor.de/community



Elektor-Newsletter E-weekly jetzt gratis abonnieren!

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

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



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



Unsere Partner und Sponsoren

	AudioXpress www.audioamateur.com	71		Linx Technologies www.linxtechnologies.com	11
	Beta Layout www.pcb-pool.com	2		LPKF www.lpkf.de/prototyping	9
	CES www.cesweb.org	33		MikroElektronika www.mikroe.com	3
	Eurocircuits www.elektorpcbservice.com	41		Reichelt www.reichelt.de	88
	Jackaltac www.jackaltac.com	73		Schaeffer AG www.schaeffer-ag.de	67
	LeitOn www.leiton.de	59			

Sie möchten Partner werden?

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

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

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

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



Seminartag für Elektronik-Profis und Hobbyisten

Am 20. Oktober 2012 veranstaltet Elektor den ersten Seminartag „ElektorLive!“ in Deutschland. Im Dortmunder Kongresszentrum Westfalenhallen haben Elektronik-Profis und Hobby-Elektroniker Gelegenheit, attraktive Seminare zu besuchen und sich auf einer Ausstellung über neue Produkte zu informieren.

Der Eintrittspreis beträgt für Abonnenten nur 25 Euro (45 Euro für Nicht-Abonnenten), Studenten, Azubis und Schüler zahlen in jedem Fall nur 20 Euro. Im Eintrittspreis inbegriffen sind zwei Seminare, die man bei der Anmeldung auswählt. Wie immer gilt: „Der frühe Vogel fängt den Wurm“, denn die Teilnehmerzahl ist bei allen Seminaren begrenzt. Hier ein kurzer Programm-Überblick:

Android-Grundlagen (Stefan Schwark)

Das Android-Framework wird auf vielen Geräten wie Smartphones und Tablet-Computern eingesetzt. Im Seminar werden die Grundlagen der Programmierung eigener Android-Apps dargestellt. Anhand von kurzen Programmen wird der Umgang mit der freien Entwicklungsumgebung „Eclipse“ und dem Android SDK erklärt.

Produktentwicklung mit Embedded Linux (Benedikt Sauter)

Embedded-Linux ist immer mehr in der Welt der klassischen Mikrocontroller-Entwicklung angekommen. Doch was bedeutet es in der Praxis, ein Produkt basierend auf Open-Source-Anwendungen zu bauen? In diesem Seminar werden die verschiedenen Aspekte (auch rechtliche und kommerzielle) betrachtet.

Transistor-Grundlagen (Burkhard Kainka)

Das Seminar behandelt die Funktion und den Einsatz von Transistoren in einfachen Grundsaltungen und Miniprojekten. Mit einem oder zwei einfachen NPN-Transistoren wie z.B. dem BC547 kann man schon die unterschiedlichsten Aufgaben lösen. Praktische Versuche und Messungen sollen live durchgeführt werden.

Entwickeln mit EAGLE Version 6 (Richard Hammerl)

EAGLE ist eine leistungsfähige und flexible Design-Software und vereint die Module Schaltplan-Editor, Layout-Editor und Autorouter. Die Software erlaubt Funktionserweiterungen wie Simulation, Datenimport und -export sowie die Definition von Befehlsabfolgen. Die neue Version bringt noch einmal vielfältige Verbesserungen mit. Natürlich erhält man hier auch Praxis-Tipps aus erster Hand.

Gassensoren (Fa. Unitronic)

Gassensor-Module bieten viele Vorteile. Die Module sind fertig kalibriert und leicht in eigene Projekte zu integrieren. Die Anwendungen reichen von simplen Schaltaufgaben beim Überschreiten bestimmter Gaskonzentrationen bis hin zur Protokollierung von Messdaten. Über neue Entwicklungen und die praktische Anwendung kann man sich in diesem Seminar informieren.

FPGA für Einsteiger (Prof. Dr. Peter Schulz)

Field Programmable Gate Arrays (FPGAs) sind aus der modernen Digitaltechnik nicht mehr wegzudenken. Das Seminar richtet sich an FPGA-Anfänger, die sich einen Überblick über diese interessante Technologie verschaffen wollen; Themen sind unter anderem Konfiguration, Hardwarebeschreibung und IP-Cores.

AVR-Xmega-Mikrocontroller (Andreas Riedenauer)

Die AVR-Chips gehören zu den beliebtesten Mikrocontrollern, nicht zuletzt, weil es leistungsfähige Entwicklungstools gratis gibt. Das Seminar stellt neue Controller und Entwicklungskits vor, Schwerpunkt ist der besonders schnelle ATXmega. Ein weiteres Thema wird die neue Version der Entwicklungssoftware „Atmel Studio 6“ sein.

Röhrenverstärker (Rainer zur Linde)

HiFi-Enthusiasten loben die klanglichen Eigenschaften der Röhren. Elektro-Gitaristen mögen das als musikalisch empfundene Verzerrungsverhalten der Bauteile. In diesem Seminar werden, neben der Vermittlung von Theorie, auch praktische Beispiele für den Entwurf und die Realisierung von Röhrenverstärkern nicht fehlen.

Elektronik steuern mit Android (Bernhard Wörndl-Aichriedler)

Mit Android lässt sich mit der Leistung des Smartphones eigene Elektronik ansteuern; so erhält man einen großen Touchscreen und obendrein noch WLAN, Bluetooth und Mobilfunk. Das ElektorLive!-Seminar ist als Crash-Kurs gedacht für alle, die ihr Android-Smartphone oder -Tablet in eigenen Projekten nutzen wollen.

Motorsteuerung mit ARM-Cortex-Controllern (Fa. Toshiba)

Die Cortex-M3-basierte M370-Prozessorfamilie von Toshiba ist für die energieeffiziente Ansteuerung von Motoren prädestiniert. Der Hersteller hat Tools entwickelt, welche die Inbetriebnahme eines neuen Motors ohne das Schreiben einer einzigen Codezeile erlaubt. Neben der Vermittlung von Hintergrundwissen steht in diesem Seminar auch eine Demonstration (mit einem BLDC-Motor und dem Toshiba Starterkit) auf dem Programm.

Elektor-Strahlungsmesser (Burkhard Kainka)

Das Seminar behandelt den Aufbau des Elektor-Strahlungsmessers, doch auch die Messpraxis kommt nicht zu kurz. Während des Seminars sollen typische Messungen durchgeführt und ausgewertet werden, mit unterschiedlichen Fotodioden und Transistoren als Alpha-, Beta- und Gamma-Sensoren.

Einführung in die OBD-II-Fahrzeugdiagnose (Florian Schäffer)

Die On-Board-Fahrzeugdiagnose gehört bei Arbeiten am Auto inzwischen genauso zum Handwerkszeug wie der Schraubenschlüssel. Mit einem OBD-II-Diagnosegerät kann man Fehlerursachen schnell



finden. Im Seminar werden Lösungen für den Selbstbau, aber auch fertige Geräte vorgestellt.

Zu den Ausstellern gehören:

Reichelt

Seit langen Jahren ist der Distributor aus Friesland allen Elektronikern ein Begriff, nicht zuletzt wegen der attraktiven Preise. Am Stand werden Arduino-Boards, eine Android-Entwicklungsplattform, Wärmebildkameras und weitere Produkte gezeigt.

Cadsoft

Auch über dieses Unternehmen muss man nicht mehr viele Worte verlieren: die CAD-Software „Eagle“ gehört zu den meistverwendeten Schaltungs- und Platinen-Entwurfsprogrammen. Am Stand kann man die neue Version 6 ausprobieren und mit Experten fachsimpeln.

Toshiba Europe

Die leistungsstarken ARM-Cortex-M3-Controller des Herstellers sind für Anwendungen wie die Ansteuerung von bürstenlosen Gleichstrommotoren prädestiniert. Am Stand sind Entwicklungs-Boards und -Software „in action“ zu sehen.

Unitronic

Schwerpunkt des Distributors sind die Kommunikation und die Sensorik. Am Stand werden interessante Lösungen aus den Bereichen Machine-To-Machine-Kommunikation, Wireless, Gassensorik und mehr gezeigt.

Trenz Electronic

Dieser Distributor ist auf Embedded-Lösungen spezialisiert. Am Stand kann man FPGA-Boards, Arduino-kompatible Boards, ein kompaktes Test- und Mess-Modul für analoge Schaltungen und mehr sehen. Von etlichen der Produkte gibt es preislich interessante Studentenversionen.

Eurocircuits

Die Herstellung von Platinen-Prototypen und -Kleinserien ist das Metier dieses Unternehmens. Am Stand sind unter anderem ein Reflow-Lötofen und ein Lötpastendrucker zu sehen. Ausprobieren kann man das neue Platinauftrags-Tool „PCB-Visualizer“. Tipps und Tricks von den Platinen-Profis gibt es gratis!

Embedded Projects

Ein kleiner, aber feiner Distributor, der Mikrocontroller- und sonstige Entwicklungsboards im Portfolio hat. Etliches wurde durch das junge Team selbst entwickelt. Geschäftsführer Benedikt Sauter ist vielen Elektor-Lesern durch Projekte wie „USBprog“ und „Elektor-Linux-Board“ bekannt geworden.

Elektor, Circuit Cellar und Audioexpress

Dass auch der Gastgeber und die Schwesterzeitschriften aus dem Verlag mit einem eigenen Stand vertreten sind, versteht sich von selbst. Am Stand sind interessante Boards und Module zu sehen. Auch neue Bücher und Sonderhefte kann man sich in Ruhe anschauen.

Mehr Infos & Anmeldung:
www.elektor-live.de

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.

www.lpkf.de/prototyping



European Microwave Week, Amsterdam
29.10. – 31.10.2012, Halle 3, Stand B425

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



First Step

NEU!

Erste Schritte mit dem Mikrocontroller

Sie interessieren sich als Auszubildender, Schüler, Student – oder einfach nur so – für Mikrocontroller-Technik? Mit dem neuen „First Step“-Paket haben Sie den Schlüssel und alle nötigen Werkzeuge für diese faszinierende Welt in der Hand! Das fertig bestückte und getestete „First Step“-Board und drei exakt darauf abgestimmte Arbeitshefte (plus Software-CD) machen die ersten Experimente mit einem Mikrocontroller zum Kinderspiel.

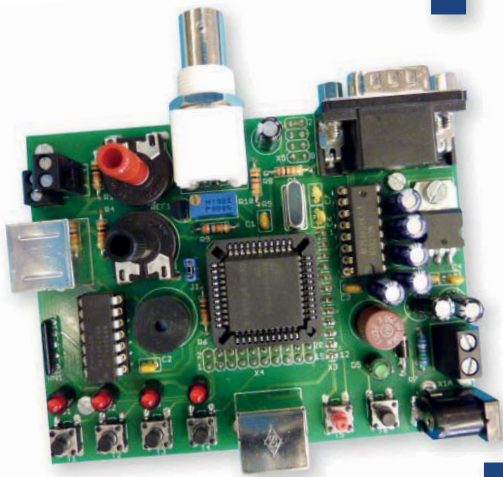
Bestandteile des „First Step“-Pakets:



→ 3 Arbeitshefte

(inkl. passendes DIN A4-Ringbuch)

- Beschreibung der Hardware
- Beschreibung der integrierten Entwicklungsumgebung IDE
- Einführung in die Programmiersprache „C“
- Zahlensysteme, Arithmetik, Variable
- Logische Operationen
- Digitale I/O-Ports
- A/D-Wandler, Timer/Counter



→ 1 „First Step“-Mikrocontroller-Board

- 8051er-Mikrocontroller: AT89C51CC03
- 2,5-V-Referenzspannungsgeber für A/D-Wandler: LT1009
- TTL/RS-232-Pegelwandler MAX232
- Treiber für LEDs und Piezo-Summer: 74HC04
- 4 Taster (Eingabe von binären Signalen)
- 4 LEDs (Ausgabe von binären Signalen)
- Piezo-Summer (Ausgabe von akustischen Signalen)
- BNC-Buchse (Ein-/Ausgabe von externen binären Signalen)
- 2 Potentiometer (Eingabe von analogen Signalen)
- 2 Mini-DIN-Buchsen und eine Doppelstock-Schraubklemme
- Karten-Format: 98 x 75 mm
- Spannungsversorgung: 9 V DC, max. 100 mA, Verpolungsschutzdiode und Miniaturversicherung

→ 1 CD-ROM mit Zusatzinfos

- Datenblätter
- Systemdokumentation
- Entwicklungsumgebung
- Beispielprogramme

Das gesamte „First Step“-Paket kostet nur 199,00 Euro.

Weitere Infos und Bestellung unter
www.elektor.de/first-step

News auf dieser Seite: Phil Knurhahn

Viren erzeugen Strom

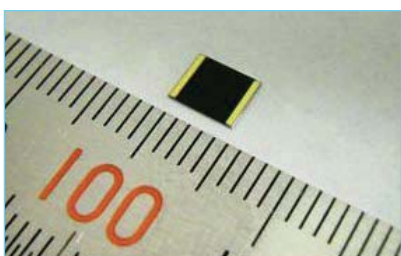
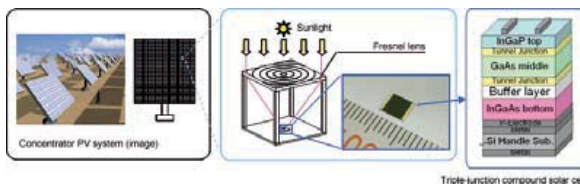
Wissenschaftler des Lawrence Berkeley National Laboratory haben harmlose Viren dazu bringen können, mechanischen Druck in elektrischen Strom zu wandeln. Wenn man den papierdünnen Generator in die Schuhsohle einbaut, dann würde der so erzeugte Strom bereits ausreichen, um ein kleines Flachdisplay (LCD) zu betreiben – Verbesserungen nicht ausgeschlossen. Es ist dies der erste biologische Generator überhaupt, der mechanoelektrische Energie produziert – echte Grundlagenforschung. Das Virus selbst ist eine M13-Bakteriophage, ein Kleinstlebewesen von 880 nm Länge und einem Durchmesser von 6,6 nm. Es ist mit 2.700 geladenen Proteinen beschichtet. Bakteriophagen sind aggressiv gegenüber Bakterien, aber nicht gegenüber Menschen. Weil sich Viren innerhalb einer Stunde millionenfach vermehren, sich dabei auch zu wohlgeordneten Filmen selbstständig organisieren, sind sie für die Forscher sehr interessant. Diese Selbstanordnung konnte unter einem Rasterkraft-Mikroskop beobachtet werden. Der gebaute Prototyp – er soll lediglich die Funktion nachweisen – besteht aus 20 Schichten übereinander, ist etwa 1 cm² groß und hat zwei Goldelektroden, an die das LCD angeschlossen wurde. Drückt man auf diese Biogenerator-Zelle, dann produziert sie etwa 6 nA an Strom bei einer Spannung von 400 mV (Foto: Berkeley Lab). Jetzt arbeitet man an Verbesserungen und es wird Zeit brauchen, bis das Prinzip einsatzfähig ist. Das Berkeley Lab gehört zum U.S. Department of Energy und wird von der Universität Berkeley gemanagt. Forscher des im Jahr 1931 gegründeten Labors haben bereits 13 Nobelpreise erhalten.

<http://newscenter.lbl.gov/news-releases/2012/05/13/electricity-from-viruses>



Solarzellen mit 43,5 % Wirkungsgrad

Die Theorie der Solarzellen besagt, dass die klassische photovoltaische Wandlung von Sonnenenergie in Strom bei einem Wirkungsgrad von etwa 33 % ihr Maximum erreicht. Wie also ist es Sharp gelungen, einen solchen Riesensprung zu machen und mit mehr als 43 % den höchsten jemals ermittelten Wirkungsgrad zu erreichen? Das geschieht über ein linsenbasiertes Dreifach-System, welches das Sonnenlicht fokussiert und in drei verschiedenen Schichten in Strom wandelt. Diese Schichten bestehen aus InGaP und GaAs, der dritte Wandler-Layer aus InGaAs (Grafik: Sharp). Das ist im Prinzip schon bekannt, aber Sharp hat sowohl die Elektrodenabstände optimiert als auch den elektrischen Widerstand der Zellen selbst. Bisher wurden solche extremen Solarzellen wegen der hohen Kosten nur in Weltraumsatelliten verwendet, wo man schon wegen des Startgewichts erhebliche Einschränkungen hat. Hier ist man jetzt auf dem Weg, solche Zellen in kleinem Maßstab in Massenfertigung zu realisieren und dann zu größeren Feldern zusammensetzen (Bild: Sharp). Das könnte den Weg bereiten für terrestrische Einsätze. Der Weltrekord wurde übrigens vom bekannten Fraunhofer-Institut für Solarenergie in Freiburg bestätigt.



<http://sharp-world.com/corporate/news/120531.html>

Algebra Quiz

1. Expand the following:
 $(2x + 3y)^2$ (5 points)

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

+2
Very funny

If only RF could be so easy.

Linx
TECHNOLOGIES

Wireless made simple®

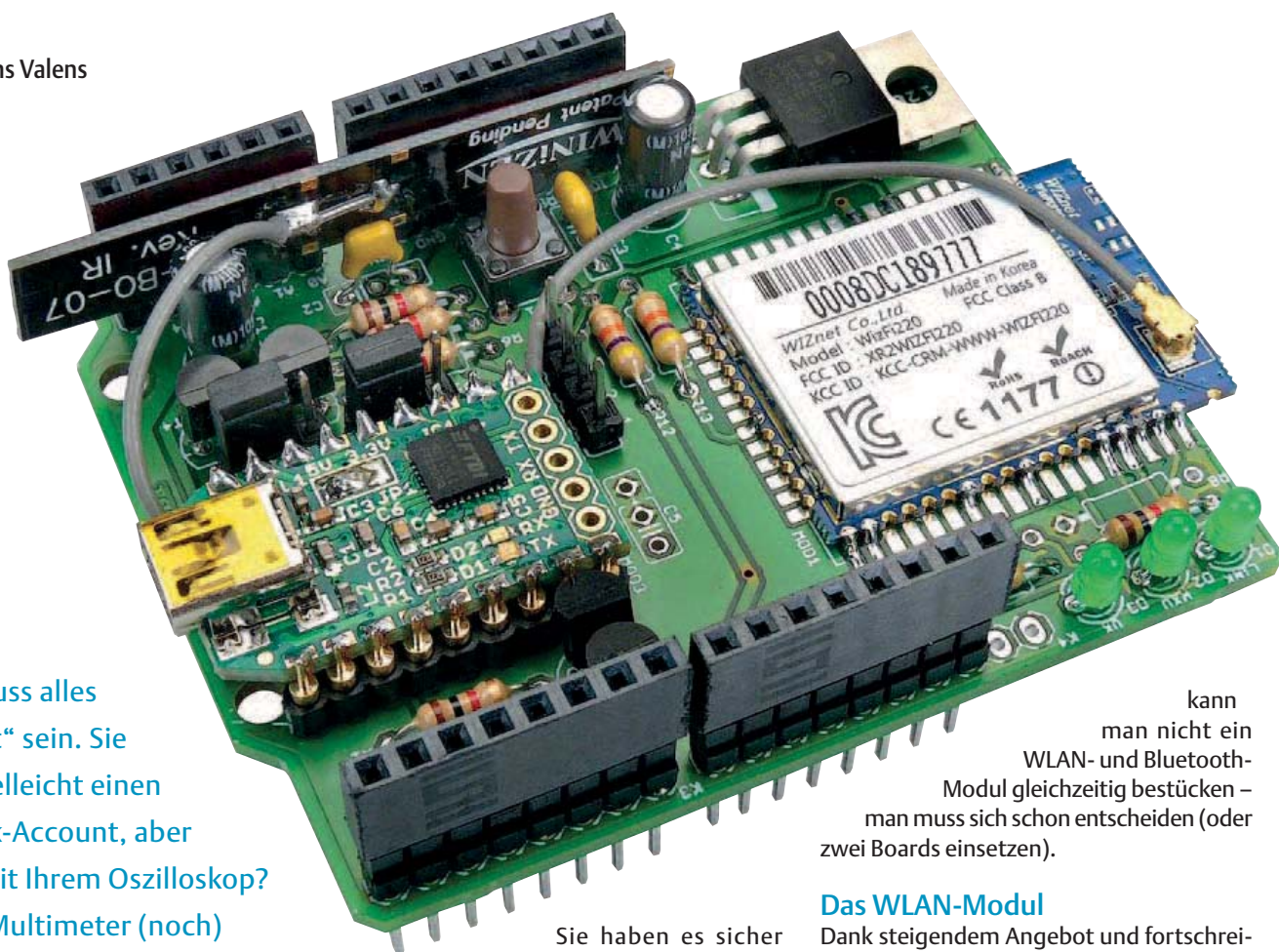
RF Modules
Remote Controls
Antennas
RF Connectors
Custom Designs

www.linxtechnologies.com

Multicom-Shield

WLAN/Bluetooth/USB-Kombi-Shield für Platino und Arduino

Von Clemens Valens



Heute muss alles „vernetzt“ sein. Sie haben vielleicht einen Facebook-Account, aber was ist mit Ihrem Oszilloskop? Darf Ihr Multimeter (noch) nicht twittern? Ist Ihr Lötkolben eingeloggt? Mag sein, dass Sie ein seltener selbstgenügsamer bzw. kaum kommunizierender Nerd sind, aber vielleicht dürsten Ihre Geräte im Labor nach sozialer Interaktion! Wie dem auch sei: Mit dem hier beschriebenen Add-on-Board kann man alles mit allem verbinden; und das per Kabel oder per Funk.

kann man nicht ein WLAN- und Bluetooth-Modul gleichzeitig bestücken – man muss sich schon entscheiden (oder zwei Boards einsetzen).

Sie haben es sicher bemerkt: Ich übertreibe ganz leicht ;-) Wirklich alles mit allem zu verbinden wäre ein extremer Anspruch. Dennoch ist das Board, um das es hier geht, wirklich sehr vielseitig. Bestückt mit einem WLAN-Modul kann es elektronische Geräte ins Internet bringen. Mit einem Bluetooth-Modul kann es Verbindung mit anderen Bluetooth-fähigen Gerätschaften aufnehmen. Per USB/Seriell-Konverter kann es auch mit älteren Apparaten seriell kommunizieren. Man kann das Board sehr einfach als Erweiterung eines Mikrocontroller-Systems mit arduino-kompatiblen Erweiterungs-Steckern (**Bild 1**) verwenden. So ein System ist zum Beispiel „Platino“ von Elektor [1] (**Bild 2**). Wenn man will, kann man es aber auch einfach so als Erweiterungs-Modul für einen normalen PC einsetzen. Allerdings

Das WLAN-Modul

Dank steigendem Angebot und fortschreitender Miniaturisierung sind WLAN-Module in den letzten Jahren geradezu unerschämpt preiswert geworden. Schwierig wird es aber an dem Punkt, wenn man von den Herstellern etwas über die Funktion und Benutzung erfahren möchte (solange man nicht sehr viele Module für sehr viel Geld kauft oder aber eine ebenfalls nicht billige maßgeschneiderte Software-Library für die Ziel-Hardware ordert). Glücklicherweise aber gibt sich die „fabriklose“ Halbleiterfirma WIZnet aus Korea weniger zugeknöpft. Direkt vom Hersteller bekam ich zwei Exemplare des Moduls WizFi220, und dieser Artikel ist die Folge. WizFi2x0-Module (es gibt auch den Typ WizFi210) funktionieren mit normalen APs (Access Points) nach 802.11b/g/n bei Geschwindigkeiten bis zu 11 Mbit/s nach

802.11b, WEP, WPA, WPA2-PSK und Sicherheitsstandards nach 802.11i werden unterstützt. Die Module eröffnen bemerkenswerte Möglichkeiten. Eine davon ist der Einsatz als WLAN-fähige serielle Schnittstelle. Ebenfalls interessant ist der Betrieb als (beschränkter) AP, was direkten Zugang zu Smartphones und Tablets unter Android oder Apples iOS erlaubt.

Das Modul WizFi220 misst 32 x 23,5 x 3 mm. Es verfügt schon über eine Chip-Antenne und eine Hirose-U.FL-Buchse zum Anschluss einer externen Antenne. Die kleinere Version WizFi210 braucht weniger Strom, bietet dafür aber auch weniger HF-Leistung und somit eine geringere Reichweite. Insgesamt 49 Pads bieten genug Möglichkeiten, das Modul irgendwie mit einer Platine zu verlöten. Die meistens Pads sind nicht für den Normal-Elektroniker, sondern für die Massenproduktion gedacht. Für Normalsterbliche reicht eine serielle Zweidrahtverbindung zum Senden und Empfangen von Daten sowie zur Konfiguration des Moduls. Wie häufig bei Modulen dieser Art kann auch ein WizFi2x0 seriell mit Hilfe von sogenannten AT- bzw. Hayes-Modem-Befehlen konfiguriert werden. Solche Kommandos sind kurze, auch für Menschen verständliche Strings, die mit dem Präfix „AT“ beginnen. Beim Einschalten ist das Modul schon im CMD-Mode, sodass man gleich mit der Konfiguration anfangen kann. Nach dem initialen Setup kann das Modul mit dem Befehl O(nline) bzw. „ATO“ in den Daten-Modus geschaltet werden. Um von hier aus wieder in den CMD-Mode zu schalten, genügen drei aufeinander folgende Pluszeichen „+++“.

Vorüberlegungen

Einige Konfigurationen wie etwa das Einstellen der Baudrate der seriellen Schnittstelle muss man ja kaum mehr als einmal durchführen, denn das Modul merkt sich diese Einstellungen. Die auch für Menschen lesbaren AT-Befehle sind nicht nur für das initiale Setup, sondern auch für Experimente praktisch, da man zur Kommunikation mit dem Modul nur einen PC mit Terminal-Programm benötigt. Ein USB/Seriell-Konverter ist also für Legacy-Free-PCs eine gute Investition. Viele Elektor-Leser werden

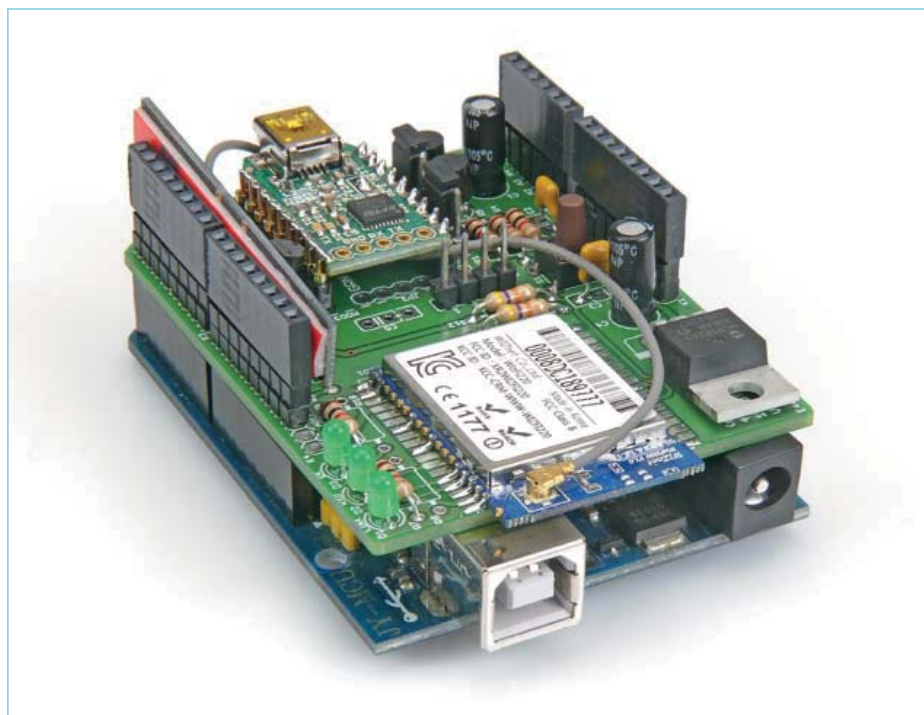


Bild 1. WLAN-Shield auf einem Arduino Uno.
R8 und R10 sind nicht bestückt.

zwar sicherlich schon so einen Konverter auf FTDI-Chip-Basis [2] mit TTL-Pegeln haben, doch ist ein integrierter Konverter auch bei diesem Board nicht schlecht, da es so keine Probleme mit einfacher Hardware wie Platino gibt, wo keine USB-Buchse zu finden ist. In der September-Ausgabe 2011 von Elektor wurde das USB-FT232R Breakout-Board [3]

vorgestellt, das ich einfach in dieses Projekt integriert habe. Mit etwas geschickter Verdrahtung kann man so den USB-Konverter mit dem WLAN-Modul oder dem Host-System (Arduino) verbinden. Man kann es auch einfach inaktivieren. Mein Projekt gestaltete sich also immer flexibler. Das WLAN-Modul benötigt 3,3 V zur Versor-

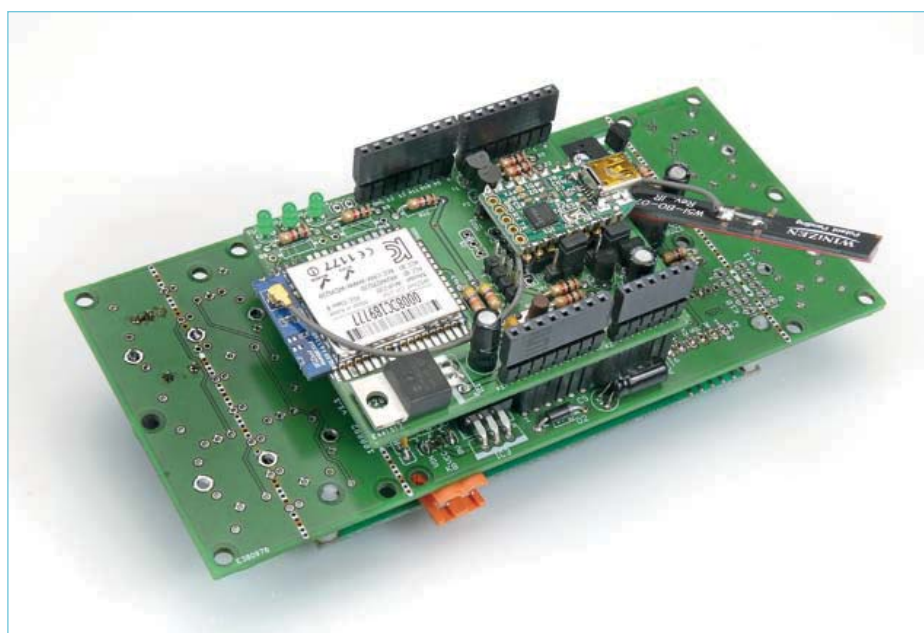


Bild 2. WLAN-Shield auf Platino. Ein alphanumerisches LCD mit 20x4 Zeichen ist ebenfalls angeschlossen.

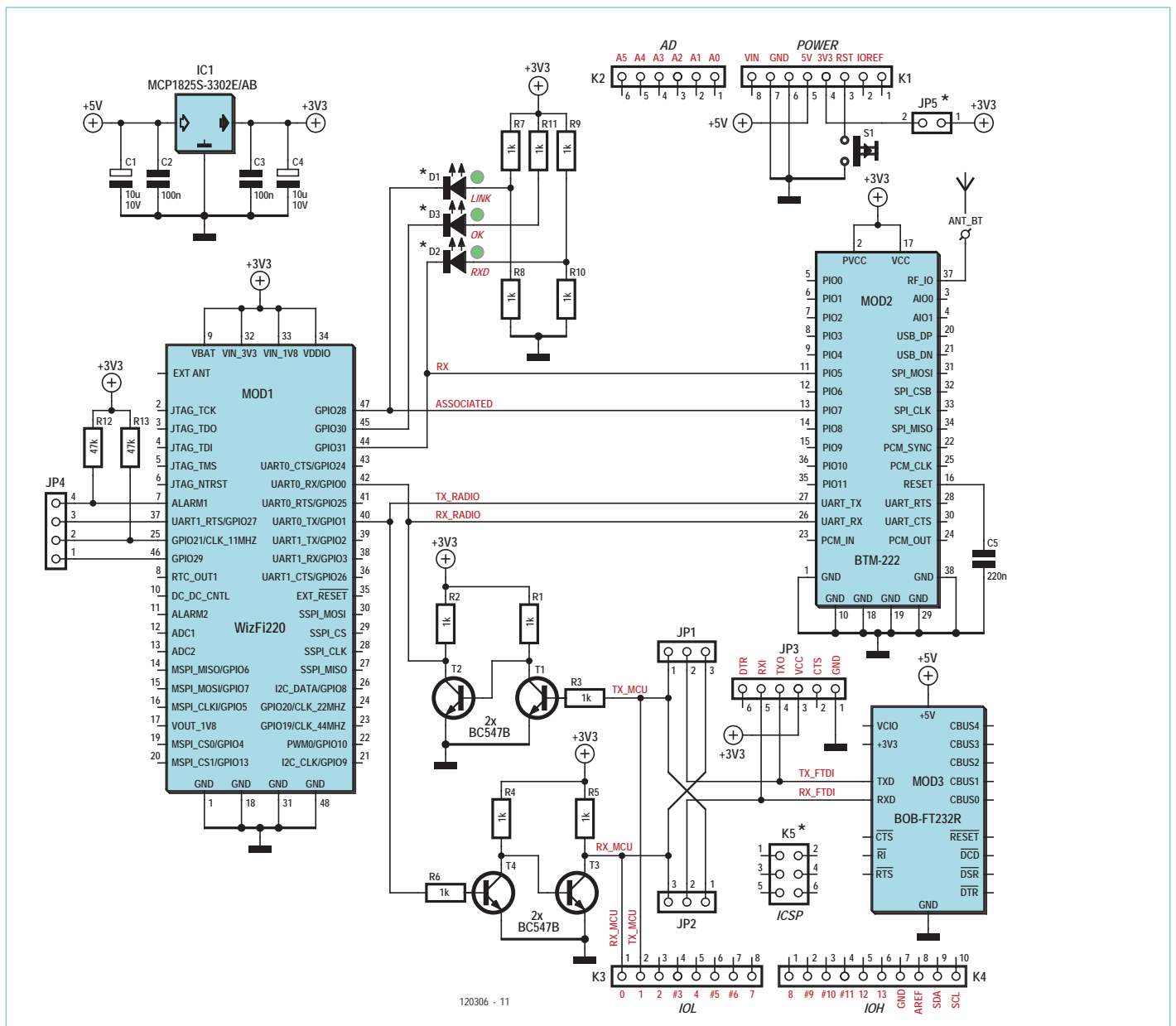


Bild 3. Komplette Schaltung des Multicom-Shields. Welche Teile wozu verwendet werden, ist im Text beschrieben.

gung. Es verträgt keine 5 V. Die Hauptanwendung ist ja, einen Mikrocontroller mit WLAN-Fähigkeiten auszustatten. Einige Controller kommen mit 3,3 V gut zurecht, andere aber wie Arduino oder Platino werden mit 5 V betrieben. Es könnten also Pegelkonverter notwendig sein. USB/Seriell-Konverter mit FTDI-Chip gibt es für beide Spannungen. Mit passenden Pegelkonvertern sind sogar beide Versionen verwendbar. Beim USB-FT232R Breakout-Board gibt es keine Probleme, denn es kommt mit beiden Pegeln zurecht. Auch wenn viele Systeme wie Arduino eine 3,3-V-Spannung anbieten, ist doch ein

extra Spannungsregler sinnvoll, denn ein WizFi220 kann im aktiven Modus leicht 250 mA aufnehmen, und in der Regel ist die 5-V-Schiene höher belastbar. Der bei Arduino Uno eingesetzte Spannungsregler LP2985-33 kann nämlich nur 150 mA liefern, was nicht ganz ausreicht. Bei Platino gibt es gleich gar keine 3,3 V. Und dann machen wir es doch gleich richtig: Über einen Jumper kann man die auf dem Multicom-Shield erzeugten 3,3 V auch dem Host-System zur Verfügung stellen. In diesem Abschnitt kann man bisher jedes Vorkommen von „WLAN“ durch „Bluetooth“ ersetzen und der Text macht immer

noch Sinn. Tatsächlich sind viele Bluetooth-Module einem WizFi-Modul sehr ähnlich: Sie können ebenfalls mit AT-Befehlen über eine serielle Schnittstelle konfiguriert werden und benötigen 3,3 V zum Betrieb. Sie haben ähnliche Abmessungen und Betriebsbedingungen. Es ist also nicht sehr schwierig, das Board um Bluetooth zu erweitern. Man muss lediglich ein brauchbares Bluetooth-Modul anstelle des WLAN-Moduls anschließen – genau das habe ich gemacht. Meine Wahl fiel auf das Modul BTM220 von Rayson [4], da ich es schon kannte und es sehr preiswert ist. Außerdem wurde es in Elektor-Projekten schon mehrfach einge-

Stückliste

Widerstände:

R1..R11 = 1k, 5%, 0,25W
R12, R13 = 47k, 5%, 0,25W

Kondensatoren:

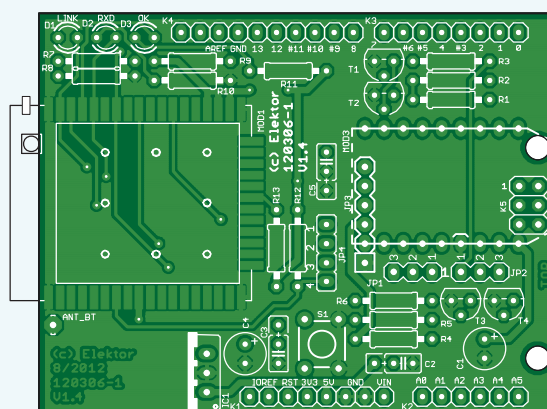
C2, C3 = 100n/50V, RM 5mm
C5 = 220n/50V, RM 5mm
C1, C4 = 10µ/35V, RM 2,5mm

Halbleiter:

D1..D3 = LED, grün, 3mm
T1..T4 = BC547C, TO-92
IC1 = MCP1825S-3302E/AB, 3,3-V-Spannungsregler, TO-220

Außerdem:

JP1 = 6-pol. Stiftleiste, RM 0,1", gerade
JP3, JP4 = 3-pol. Stiftleiste, RM 0,1", gerade, mit Jumper
JP2 = 4-pol. Stiftleiste, RM 0,1", gerade
K1, K3, K4 = 8-pol. Steckverbinder, RM 0,1",



stapelbar
K2 = 6-pol. Steckverbinder, RM 0,1", stapelbar
K5 = 6-pol. (2x3) Steckverbinder, RM 0,1",

stapelbar

S1 = Taster, SPNO, bedrahtet, 6x6mm
Mod1 = WIZnet WizFi2x0, WLAN-Modul
Mod2 = Rayson BTM22x, Bluetooth-Modul
Mod3 = USB-FT232R Breakout-Board, Elektor # 110553-91, oder USB/TTL-Konverterkabel, Elektor # 080213-71 (5V) bzw. 080213-71 (3,3V)
Platine # 120306-1 [1]

setzt [5][6].

Soviel zu den Grundlagen meines Projekts. Nun zur Schaltung in **Bild 3**. Mit diesen Vorüberlegungen im Hinterkopf bietet sie kaum Überraschungen.

Alles zusammen

Links befindet sich das WLAN- und rechts das Bluetooth-Modul. Sie teilen sich die RX- und TX-Leitung, weshalb man nicht beide zugleich bestücken kann. Also WLAN **oder** Bluetooth, aber nicht beides. Mit D1 und D2 werden auch zwei LEDs von beiden Modulen angesteuert. D3 wird lediglich vom WLAN-Modul genutzt. Vielleicht runzeln Sie bei R8 und R10 die Stirn. Das hätte ich auch besser zeichnen können, aber ich habe es absichtlich so merkwürdig gelassen, damit ich den Sinn erklären kann und muss: Das WLAN-Modul legt die Kathoden der LEDs auf Masse, während das Bluetooth-Modul deren Anoden auf +3V3 schaltet. Folglich gilt für den WLAN-Betrieb: **R8 und R10 entfallen**. Bei bestücktem Bluetooth-Modul werden

R8 und R10 bestückt, aber **R7 und R9 entfallen**. R11 und D3 sind dann funktionslos, wenn man sie nicht an einen anderen Pin des Moduls legt. Sie können auch weggelassen werden. Wichtig ist aber, dass bei Bluetooth im Vergleich zum Schaltplan **D1 und D2 andersherum gepolt** bestückt werden! Das Wi-Fi-Modul hat etliche Pins mit speziellen Funktionen. Ich brauchte sie nicht und habe daher ein paar davon via JP4 zugänglich gemacht. Damit man noch weitere Shields auf diesen stecken kann, habe ich diese Signale nicht auf die üblichen Erweiterungs-Ports K1 bis K4 gelegt. Wenn Sie also welche davon benötigen, können Sie diese selbst herausführen. Pin 25 (GPIO 21) erlaubt das Wiederherstellen der Default-Einstellungen. Zwei Impulse versetzen das Modul in den (begrenzten) AP-Modus, und drei Impulse schalten in den Ad-hoc-Mode. Pin 46 (GPIO 29) schaltet zwischen Command- (high) und Data-Mode (low) um. Pin 37 (GPIO 27) ist für das Upgrade der Firmware notwendig (high) und wird hof-

fentlich nie benötigt. Ein „low“ an Pin 7 (Alarm1) schließlich lässt das Modul aus dem Deep-Sleep-Mode aufwachen.

Die Pegel-Konverter aus R1...R3, T1, T2, R4...R6, T3 und T4 enthalten keine Geheimnisse und wurden in Elektor schon mehrfach eingesetzt. T1 und T4 erledigen die Pegelanpassung, invertieren aber das Signal. Daher wird dies durch je eine weitere Inversion mit T2 und T3 korrigiert. Die Werte der Widerstände sind unkritisch. Ich habe so viele 1-kΩ-Widerstände wie möglich verwendet. RX und TX kommen an JP1 und JP2. Diese 3-Pin-Jumper sind nicht ganz selbsterklärend. Doch wenn man den Leitungen folgt, sollte man deren Funktion verstehen können. Ist bei JP1 eine Brücke zwischen den Pins 1 und 2 gesteckt, dann wird TX des USB/Seriell-Konverters an RX des WLAN- oder Bluetooth-Moduls gelegt. Eine Brücke zwischen den Pins 2 und 3 von JP2 verbindet RX des USB/Seriell-Konverters mit TX des WLAN- oder Bluetooth-Moduls. In diesen Positionen kann das jeweilige Modul vom PC angespro-

Elektor-Projekte & Produkte

- Platino, Multitalentierte AVR-Platine (Oktober 2011); Platine # 100892-1
- USB-FT232R Breakout-Board (September 2011); Modul # 110553-91
- USB-TTL-Adapterkabel (Juni 2008); Kabel # 080213-71 (5 V) oder 080213-71 (3,3 V)

Listing 1.

Beispiel einer Konfiguration des WizFi2x0-Modul als serieller Server mit AT-Befehlen. Die Befehle sind **fett** und die Antworten des Moduls sind *kursiv* gedruckt. Kommentare sind (eingeklammert). Im WizFi2x0-User-Manual findet man noch weitere Befehle.

```

AT (aufwachen)
[OK]
AT+NSTAT=? (wie ist der Status?)
MAC=00:08:dc:18:97:76
WSTATE=NOT CONNECTED    MODE=NONE
BSSID=00:00:00:00:00:00  SSID=""  CHANNEL=NONE  SECURITY=NONE  RSSI=0
IP addr=0.0.0.0  SubNet=0.0.0.0  Gateway=0.0.0.0  DNS1=0.0.0.0  DNS2=0.0.0.0
RxCount=0 TxCount=0
[OK]
AT+WPAPSK=myssid,"mypassword" (setze SSID & Passwort)
Computing PSK from SSID and PassPhrase...

[OK]
AT+NDHCP=1 (erfrage IP-Adresse vom DHCP-Server)
[OK]
AT+WAUTO=0,myssid (automatische Verbindung mit Netzwerk myssid)
[OK]
AT+NAUTO=1,1,,8011 (Setup für Auto-Connect: Server, TCP, Port 8011)
[OK]
ATA (starte Auto-Connect)
      IP              SubNet          Gateway
      192.168.2.7: 255.255.255.0: 192.168.2.1
[OK]

```

Nun kann man z.B. mit Telnet eine Verbindung per WLAN zum Modul aufnehmen.

chen und konfiguriert werden. Man erhält aber keine serielle Verbindung zum Host-System, weshalb dieses dann besser nicht angeschlossen wird. Das Board wird in diesem Modus vom USB-Port versorgt. Brückt man aber die Pins 2 und 3 von JP1 und die Pins 1 und 2 von JP2, dann kann man per USB/Seriell-Konverter auf das Host-System zugreifen. In diesem Fall fungiert das Board als USB-Shield. Es sollte daher kein Funk-Modul bestückt sein, da dieses sonst die Kommunikation stören könnte. Ist bei JP1 und JP2 nichts gesteckt, ist das Board im Funk-Modus und der USB/Seriell-Konverter wird nicht genutzt. Man kann ihn aber bestückt lassen und bei Bedarf dessen RX und TX mit anderen Pins eines Erweiterungssteckers verbinden. Dies kann bei einer Arduino-Software-UART-Konfiguration oder in einem Host-System mit mehr als einem UART durchaus sinnvoll sein. Zum Schluss noch einige Details: Der Reset-

Taster ist hauptsächlich für Platino gedacht, da er dort fehlt. Er ist auch bei einem Arduino nützlich, wenn dort ein Shield den Reset-Knopf verdeckt, was auch vom Arduino-Board und dessen Revision abhängt. K5 ist nicht angeschlossen und nur deshalb auf dem Board, da man so auf Signale eines Arduino zugreifen kann.

Die Praxis

Nun stand die Transformation der Schaltung in eine Platine bzw. ein Shield an. Das von mir erstellte Layout kann in Form von Eagle-Dateien von der Webseite dieses Artikels [7] heruntergeladen werden. Die Platine entspricht in der Größe der von Arduino Uno. Sie ist allerdings komplett rechteckig. Alle Bauteile sind konventionelle bedrahtete Ausführungen. Die einzigen Ausnahmen sind das WLAN- und das Bluetooth-Modul, die als SMD ausgeführt sind. Der Lötbrücken-Jumper JP5 und das Bluetooth-Modul

befinden sich auf der unteren Lötseite der Platine. Alle anderen Bauteile befinden sich auf der oberen Bestückungsseite. Wenn man für K1...K4 nicht stapelbare Ausführungen verwendet, müssen diese auf die Lötseite, damit man die Platine auf ein Arduino- oder Platino-Board stecken kann.

JP1 und JP2 sind so angeordnet, dass ihre Steckbrücken gleich ausgerichtet werden – also beide entweder links oder rechts oder gar nicht gesteckt sind. Will man die Funk-Kommunikation des Host-Systems für Debugging-Zwecke beobachten, kann man JP2 gesteckt lassen.

An JP3 liegen RX und TX des USB/Seriell-Konverters. Man verwendet also entweder das USB-FT232R Breakout-Board oder aber einen externen Konverter mit TTL-Pegeln für RX und TX, doch nicht beides.

Bei durchweg mit 3,3 V operierenden Systemen können natürlich der Spannungsregler und die Pegel-Konverter entfallen. In



Bild 4. Die drei Hauptvarianten nebeneinander. Das Bluetooth-Modul beim mittleren Board ist nicht sichtbar, da es auf der Unterseite der Platine bestückt ist. Bei den Bildern handelt es sich um einen frühen Prototypen, dessen Platine der von Arduino Uno entspricht. Die endgültige Platine ist rechteckig.

diesem Fall sind zwei Drahtbrücken von R3/JP1 nach R2/T2 und von R6/MOD1 zu R5/T3 erforderlich. Die Widerstände R3, R2, R5 und R6 sind so angeordnet, dass sehr einfach kurze Drahtbrücken verwendet werden können. Außerdem muss JP5 auf der Lötseite der Platine gebrückt werden. Wie schon beschrieben ist die Polung der LEDs vom verwendeten Funk-Modul abhängig. Der Bestückungsdruck der Platine ist richtig, wenn das WLAN-Modul verwendet wird. Bei Bluetooth ist daher D1 und D2 „verkehrt herum“ zu polen. Und denken Sie daran: Entweder R7, R9 und R11 oder R8 und R10 bestücken! Der 3,3-V-Spannungsregler kann stehend oder liegend bestückt werden.

Bild 4 zeigt die drei Hauptvarianten, die prinzipiell modifizierbar sind:

- **WLAN-Shield.** Wenn man einen externen USB/Seriell-Konverter hat, muss man das abgebildete Konverter-Modul nicht bestücken. JP3 dient zum Anschluss des externen Converters. Alle LEDs sind wie im Bestückungsdruck vorgesehen bestückt. R8 und R10 entfallen. Pegel-Konverter sind überflüssig und je nach Host-System auch der 3,3-V-Spannungsregler.
- **Bluetooth-Shield** — ähnlich wie die WLAN-Variante. D3 ist funktionslos. R7, R9 und R11 können entfallen, aber R8 und R10 werden bestückt. Als Antenne genügt ein Stück isolierter Schalt Draht mit 31 mm Länge.

- **USB/Seriell-Shield** — benötigt wird nur ein bestückter USB/Seriell-Konverter ohne Pegel-Konverter. Mit einem Lötzinnlecks an JP1 wird der USB/Seriell-Konverter für 5-V- oder 3,3-V-Betrieb konfiguriert. Der 3,3-V-Spannungsregler und der Reset-Taster sind sinnvoll. Die LEDs kann man bei Bedarf nutzen, man muss sie dann aber mit den geeigneten Pins verbinden. Die Pins 2 und 3 von JP1 und die Pins 1 und 2 von JP2 werden gebrückt.

Test

Bei den Varianten 1. und 2. wird das Board über einen externen oder internen USB/Seriell-Konverter mit dem PC verbunden. Zunächst muss man den vom Betriebssystem des PCs festgelegten virtuellen COM-Port des USB/Seriell-Konverters identifizieren. Dazu müssen natürlich auch die passenden Treiber installiert sein. Für Chips von FTDI gibt es diese samt Dokumentation via www.ftdichip.com. Dann startet man ein Terminal-Programm. Das Modul WizFi220 hat eine Standard-Baudrate von 115,2 kbit/s und das Modul BTM220 eine von 19,2 kbit/s. Beide Module verwenden 8 Datenbits, kein Parity und kein Hardware-Handshake. Wenn man nun im Terminal „AT“ gefolgt von der Eingabetaste eintippt, dann sollte das Modul mit „[OK]“ oder „OK“ antworten. Klappt das, ist alles in Ordnung. **Listing 1** enthält ein funktionierendes WLAN-Beispiel (die Einstellung von SSID und Passwort des eigenen Netzwerks sollte nicht vergessen werden).

Beim USB/Seriell-Shield wird ähnlich vorgegangen. Der Test mit dem AT-Kommando ist allerdings nur möglich, wenn man ein angeschlossenes Host-System selbst entsprechend programmiert.

An dieser Stelle ein Dankeschön an Joachim Wülbeck von WIZnet Europe (www.wiznet.eu) für die Überlassung der WizFi220-Module samt WLAN-Antennen.

(120306)

Weblinks

- [1] Platino: www.elektor.de/100892
- [2] USB-TTL-Adapterkabel: www.elektor.de/080213
- [3] USB-FT232R Breakout-Board: www.elektor.de/110553
- [4] Experimente mit Rayson-Bluetooth-Modulen: <http://elektorembdedd.blogspot.com/2010/08/rayson-btm222-btm112-bluetooth-modules.html>
- [5] Bluetooth mit dem ATM18: www.elektor.de/080948
- [6] Diagnose mit Blauzahn: www.elektor.de/090918
- [7] Multicom-Shield: www.elektor.de/120306
- [8] Multicom-Shield bei Elektor Projects: www.elektor-projects.com/project/wi-fi-bluetooth-usb-shield-for-arduino-platino.12252.html

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

NEU

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

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

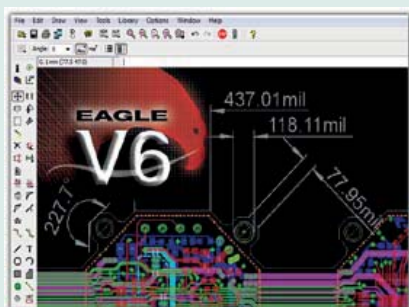


Embedded Linux in Theorie und Praxis (3-tägiges Seminar)

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

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

Eagle PCB und Design (2-tägiges Seminar)



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

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

Elektronik steuern mit Android (AndroPod)

NEU

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

Referent: B. Wörndl-Aichriedler – Teilnahmegebühr: 349,00 € (inkl. MwSt.)

Schaltungsdesign mit Mikrocontroller (2-tägiges Seminar)

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

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

Referent: Dipl.-Ing. Franz Peter Zantis – Teilnahmegebühr: 799,00 € (inkl. MwSt.)

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

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

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

Referent: Prof. Dr.-Ing. Francesco P. Volpe – Teilnahmegebühr: 1.299,00 € (inkl. MwSt.)

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

NEU



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

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

Sounddesign (1-tägiges Seminar)

Geräusche werden heute nicht mehr einfach akzeptiert so wie sie sind. Das Geräusch muss zum Gerät oder zur Maschine passen. Es vermittelt ein „Image“. Es soll angenehm sein und zum Gerät passen. Dabei spielt auch die Umgebung in der das Geräusch wahrgenommen wird eine Rolle. Bevor man allerdings Geräusche designen kann, müssen die notwendigen Grundlagen und die dazu verfügbaren technischen Verfahren bekannt sein.

Das Seminar gibt einen ersten Einblick in die Bedeutung sowie der korrekten Erfassung und Analyse der Geräusche. Erst wenn diese Techniken sicher beherrscht werden, ist es möglich, ein Geräusch nach eigenen Vorstellungen zu verändern – also zu designen.

Referent: Dipl.-Ing. Franz Peter Zantis – Teilnahmegebühr: 499,00 € (inkl. MwSt.)

Besuchen Sie uns auf unserem Seminar- & Ausstellungstag:

ElektorLive! 2012 – 20. Oktober 2012 in Dortmund



Workshops * Seminare * Kurse * Weiterbildungen

Top-Fachleute aus der Branche referieren
über ein faszinierendes Thema!

Embedded Linux in Theorie und Praxis

21.09.2012 Füssen
04. bis 06.12.2012 Hannover
20. bis 22.11.2012 Zürich

Seminar Schaltungsdesign mit Mikrokontroller

20. + 21.09.2012 Füssen
16. + 17.10.2012 München

Eagle PCB und Design

18. + 19.09.2012 Füssen
07. + 08.11.2012 Hanau

Elektronik steuern mit Android (AndroPod)

20.09.2012 Füssen

Seminar Sounddesign

27.09.2012 Dortmund

PIC-Mikrocontroller- Programmierung in C

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

Seminar Messen, Steuern, Regeln mit VB.NET

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

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



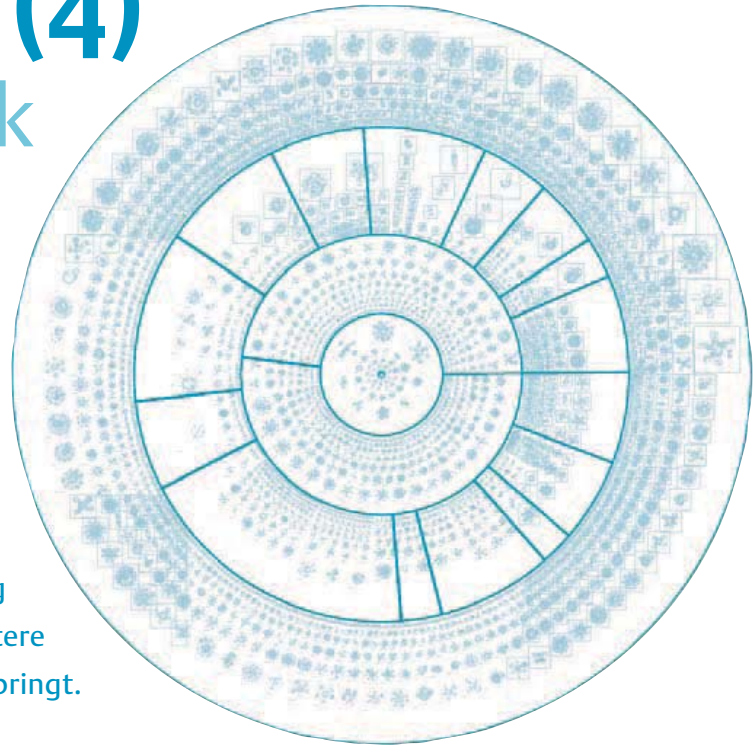
Weitere Infos & Anmeldung: www.elektor.de/events

Embedded Linux leicht gemacht (4)

Ein Kernel-Überblick

Von Benedikt Sauter [1]

GNU/Linux bietet Schnittstellen für Geräte und Anwendungen an, die man von modernen Desktop-Computern oder Servern kennt. Auch im Embedded-Bereich nimmt Linux dem Programmierer viel Arbeit ab: für Netzwerk, USB, Bluetooth und mehr muss man nicht mehr aufwendig eigene C-Lösungen entwickeln. Und es gibt noch weitere Vorzüge, die solch ein modernes Betriebssystem mitbringt.



In den letzten Folgen der Serie haben wir die Toolchain, den Kernel und Bootloader sowie ein Standard-Dateisystem zum Laufen gebracht. In unserer Artikelreihe soll dem Entwickler aber auch gezeigt werden, was „unter der Oberfläche“ passiert, zum Beispiel bei der Geräte- und Prozessverwaltung. Wir wollen uns in dieser Folge zuerst einmal anschauen, was das Betriebssystem im Vergleich zur klassischen, direkt auf der Hardware laufenden Firmware („Bare Metal“) an Mehrwert bietet.

Was ist überhaupt ein Betriebssystem? Im Wesentlichen ist das eine Software-Sammlung, die man benötigt, um eigene Anwendungen parallel mit anderen ablaufen lassen zu können. Der Übergang von „normaler“ Firmware zu einem echten Betriebssystem ist fließend. Da geht es los mit einem sauber konstruierten Hardware-Layer, der immer wieder benutzte Funktionen (zum Beispiel zur Speicherverwaltung) mitbringt. Und auch ein kleiner Scheduler, der einzelnen Programmen zyklisch Rechenzeit zuteilt, ist schon sehr „betriebssystem-like“.

Typisches Betriebssystem

Natürlich gibt es zum Thema eine Vielzahl von Büchern [2]. Uns soll hier ein grober Überblick über die wesentlichen Komponenten eines Betriebssystems genügen:

- Geräteverwaltung
- Prozessverwaltung
- Speicherverwaltung
- Dateiverwaltung
- Benutzer- und Rechteverwaltung

Im Embedded-Bereich ist wohl die Geräteverwaltung besonders bedeutsam. Das Betriebssystem muss die Peripherie und Hardware „abbilden“, so dass von der Anwendung aus ein einfacher Zugriff möglich ist. Dieser wird durch Gerätetreiber gesteuert. Das Betriebssystem regelt auch alles Nötige, wenn gleichzeitig mehrere Programme auf SD-Karten, Netzwerkschnittstellen usw. zugreifen wollen. Eine Anwendung ist in einem Betriebssystem (sofern man kein aufwändigeres Programmiermodell wählt) ein einzelner, selbständig ausführbarer Prozess. Ein Prozess hat einen eigenen virtuellen Adressraum, wo er keinen anderen Prozess stören kann. Zentral werden Prozesse in der Prozesstabelle [3] verwaltet, dort gehört zu jedem ein Prozesskontrollblock („Process Control Block“). In diesem findet man die typischen Speicherbereiche, wo Registerinhalte und andere wichtige Variablen zwischengespeichert werden:

- Befehlszähler
 - CPU-Register
 - Stack-Pointer
 - Zustand geöffneter Dateien
 - Zustand des Prozesses
- etc.

Starten wir unser Programm „Hello World“, dann legt der Kernel beim Laden der Anwendung einen neuen Prozess an. Es wird immer vom aktuellen Prozess aus ein neuer abgeleitet, wobei auch die Benutzerrechte zuerst einmal vererbt werden. Mit dem Befehl

```
ps -ejH
```

kann man sich die dadurch entstehende, baumartige Struktur anzei-

gen lassen. Man sieht, dass der erste vom Kernel aufgerufene Prozess immer „/sbin/init“ ist. Dieses Programm wird beim Start als allererstes vom Kernel aufgerufen.

Als Entwickler interessieren uns besonders die folgenden Features des Betriebssystems:

- Schnittstelle für Gerätetreiber
- Dateisystem für Programme und Daten
- Ausführen und Starten von Programmen
- Gemeinsame Nutzung von Bibliotheken durch verschiedene Programme
- Zugriff auf Ein- und Ausgabeschnittstellen
- Auslagern von Arbeitsspeicher auf einen nichtflüchtigen Speicher
- Speicherverwaltung für Anwendungen bzw. Prozesse
- Prozessverwaltung

Beginnen wir einmal mit der Geräteverwaltung.

Konfiguration des Kernels

Abhängig von der Anwendung werden verschiedene Schnittstellen und Peripherieblöcke des eingesetzten Prozessors benötigt. Der Linux-Kernel bietet durch seinen modularen Aufbau eine sehr feine Einstellmöglichkeit dessen an, was an Treibern und Mechanismen in den Kernel mit aufgenommen werden soll. Andere Teile kann man später dynamisch nachladen. Wie das alles funktioniert, schaut man sich am besten direkt im Quelltext des Kernels an. Dafür wechseln wir in den Quelltextbaum des Kernels...

```
cd ElektorLinuxBoardDownload_20120509
```

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

...und rufen dort das Menü des Kernels auf:

```
make menuconfig
```

Direkt danach öffnet sich das Hauptmenü (**Bild 1**). Sollte das Menü nicht erscheinen, kann es sein, dass die Bibliothek für das Anzeigen solcher Konsolen-Menüs fehlt. Diese kann man einfach auf unserem Ubuntu-System nachinstallieren:

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

Vom Hauptmenü des Kernels aus kann man alle Features und Gerätetreiber frei ein- und ausschalten. Für den Neuling ist es leider nicht immer ganz eindeutig, wo etwas im Menü zu finden ist, aber mit etwas Übung kennt man sich schnell aus. Anstelle das Menü zu verwenden, könnte man auch direkt im Ordner des Quelltextes (linux-2.6.33-lpc3131x) die Konfigurationsdatei „config“ (der Punkt bedeutet, dass es sich um eine versteckte Datei handelt) aufsuchen und mit einem Editor editieren. Aber bequemer geht alles

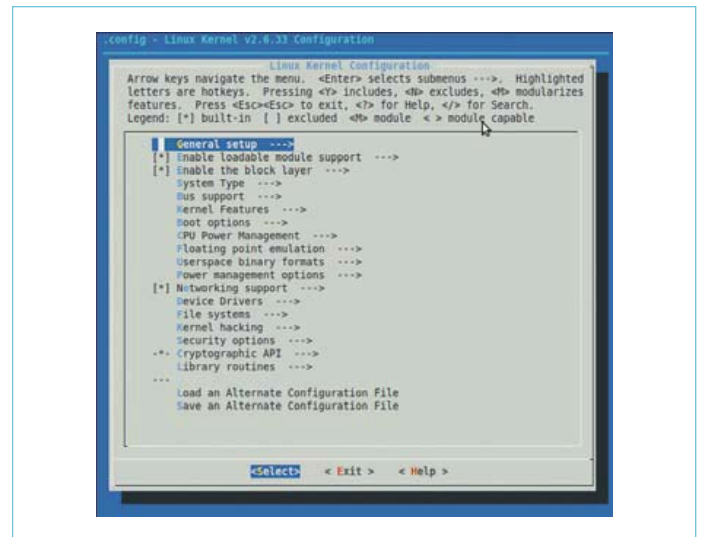


Bild 1. Hauptmenü des Kernels.

mit diesem Menü. Bevor man mit dem Bearbeiten der Konfiguration beginnt, sollte man sich aber eine Sicherheitskopie der aktuellen Konfiguration anlegen.

```
cp .config config_backup
```

Tipp: Viele Linux-Einsteiger fragen immer nach einer guten Dokumentation. Die beste Dokumentation für den Kernel ist im Kernel selbst zu finden. Zum einen kann man im Menü jederzeit auf „Hilfe“ klicken und Informationen zum jeweiligen Item abrufen, oder man stöbert einmal im Ordner „Documentation“ (direkt im Quelltextverzeichnis) nach dem passenden Thema.

In Bild 1 sieht man, dass die Menüpunkte nach Gruppen sortiert sind. Es geht los mit allgemeinen Einstellungen zum Kernel, und dann weiter über die Gerätetreiber und Dateisysteme bis hin zu Sonderfunktionen wie Verschlüsselung und Sicherheit.

General Setup

Im Menüpunkt „General Setup“ findet man grundlegende Kernel-Eigenschaften. So z.B. ob es einen Auslagerungsspeicher gibt, wie der Speicher verwaltet wird, wie die interne Kommunikation umgesetzt wird. Dazu kommen die Eigenschaften des Kernel-Images selbst, Optimierungen für kleine Systeme, Statistik, Tools und vieles mehr. Wenn man sich mit dem Linux-Kernel beschäftigt, findet man nach und nach immer mehr nützliche Features. Es gibt wohl auch nur wenige Menschen, die alle Funktionen kennen und verstehen!

Enable loadable module support

Das Gute an Linux ist, dass man den Kernel zur Laufzeit ganz einfach mit Modulen erweitern kann. Im Kernel-Menü erkennt man,

Neue Version des Elektor-Linux-Boards

Der Ansturm auf das Elektor-Linux-Board war wirklich gewaltig! Aufgrund der zahlreichen Bestellungen musste die nächste Produktions-Charge schneller aufgesetzt werden als erwartet. Leider hat der DRAM-Speicherchip im BGA-Gehäuse eine Lieferzeit von rund 20 Wochen. Daher fiel die Entscheidung, das Layout auf die Bauform TSOP54 umzustellen. Der neue Arbeitsspeicher ist genauso groß, benötigt aber anstatt 1,8 V eine Stromversorgung von 3,3 V. Das obere Bild zeigt die ursprüngliche Board-Version mit der BGA-Bauform des Arbeitsspeichers. Die Board-Version im mittleren Bild ist bereits auf einen DRAM-Chip im TSOP54-Gehäuse umgestellt. Ab Anfang Juli wird die Board-Version 1.1-blau ausgeliefert, bei welcher der nicht mehr benötigte 1,8-V-Spannungsregler entfallen ist. Diese aktuellste Version des Elektor-Linux-Boards ist leicht an der blauen Platinenfarbe zu erkennen. Den dazugehörigen Schaltplan haben wir hier abgebildet.

ob ein Feature bzw. Treiber fest im Kernel oder als Modul übersetzt ist; und zwar an der Markierung vor dem Menüeintrag. Ein „M“ bedeutet, dass es sich um ein Modul handelt. Das Sternchen steht für fest im Kernel integrierte Funktionen. Möchte man eine Vielzahl an verschiedenen Geräten unterstützen, würde der Kernel irgendwann unnötig groß werden. Dann ist es besser, die Gerätetreiber als Modul zu übersetzen und bei Bedarf entweder manuell oder automatisch zu laden.

Enable the block layer

Im Kernel unterscheidet man zwischen zwei Gerätearten: dem Character- und dem Block-Device. Bei einem Character-Device sendet und empfängt man Nachrichten bzw. Daten zeichenweise. Bei einem Block-Device sind es immer komplette Blöcke. Typische Character-Geräte sind Konsolen und UART-Schnittstellen. Typische Block-Geräte sind Festplatten oder Speicherkarten (wie MMC und SD). Benötigt man keine Block-Devices (kommt am ehesten bei Embedded-Systemen vor), kann man etwas Platz im Speicher sparen, wenn man den „Block Layer“ deaktiviert.

System Type

Einige Features des Kernels hängen direkt vom Prozessor-Typ ab. So kann man hier z.B. gezielt Caches oder Speicherverwaltungseinheiten ein- und ausschalten. In diesem Unterpunkt stellt man (sofern schon vorhanden) den vorliegenden Prozessor-Typ ein. Fehlt der eigene Prozessor, muss man sich ein Patch bzw. entsprechendes Board-Support-Package suchen (oder selbst schreiben).

Bus support

Typische PC-Busse kann man hier aktivieren. In unserem Fall steht nur die Option PCMCIA zur Verfügung.

Kernel Features

Sehr kernel-spezifisch: Man kann hier einstellen, wie die Binärschnittstelle für Programme aussieht oder auch solche Optionen wie die Anzeige der Auslastung anhand der Blinkfrequenz einer LED.

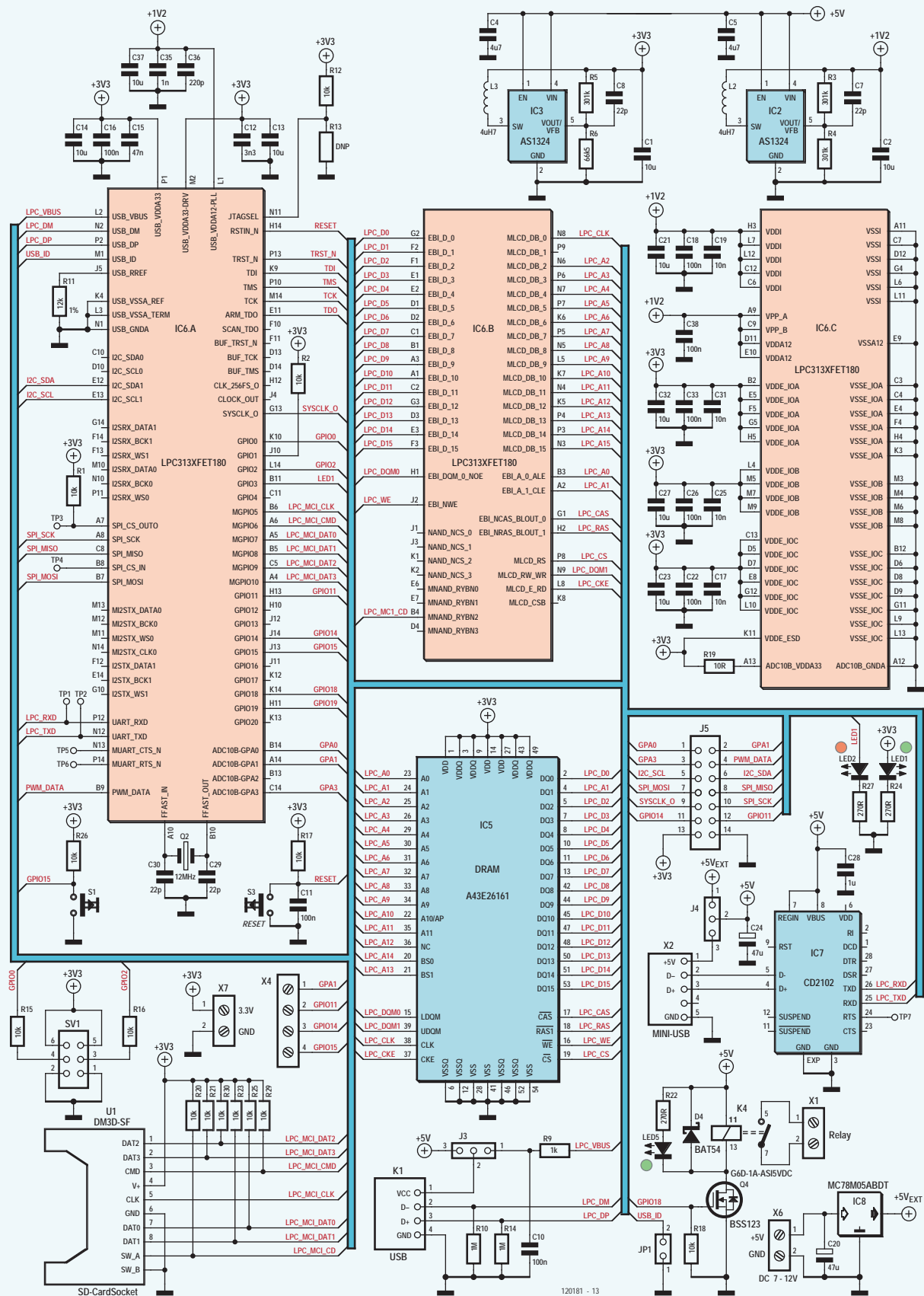
Boot options

Auf welche Weise der Kernel gestartet wird, hängt beim Elektor-Linux-Board im Wesentlichen vom Bootloader ab. Primäre Schnittstelle vom Bootloader zum Kernel ist die sogenannte „Kernel Commandline“. In dieser Zeile kann man dem Kernel Parameter übergeben, die später von diesem oder anderen Programmen ausgewertet werden können. In diesem Unterpunkt könnte man einstellen, dass eine andere, eigene Commandline eingesetzt werden soll. Oder man lässt den Kernel so übersetzen, dass man ihn direkt aus einem Flashspeicher starten lassen kann.

CPU Power Management

Moderne Prozessoren bieten immer mehr Unterstützung für ein gutes Power-Management an. Damit diese Features von Anwendungen genutzt werden können, muss sie der Kernel entsprechend implementiert haben und anbieten.





120181 - 13

Floating point emulation

Auch eine Option im Kernel-Menü: Entweder ist der Kernel selbst für die Fließkommaberechnung zuständig oder er kann auf eine Hardwareunterstützung zurückgreifen. Mindestens eine Möglichkeit muss ausgewählt werden, um das Ablaufen von Programmen zu ermöglichen.

Userspace binary formats

Programme bzw. Anwendungen werden mit der Toolchain übersetzt (siehe die vorangegangenen Teile der Serie). Die Toolchain erzeugt bei Linux gewöhnlich eine .elf-Datei (Executable and Linkable Format, entspricht in etwa dem .exe bei Windows) oder eine a.out-Datei. Der Kernel kann diese Formate lesen und daher die Anwendungen entsprechend starten.

Tipp: Unter Linux kann man ganz einfach mit dem Befehl `file` überprüfen, um welches Format es sich bei einer Datei mit einer beliebigen Endung handelt. Man bewegt sich mit `cd` in den entsprechenden Ordner und ruft dort `file` mit der zu testenden Datei als Parameter auf. Im Linux-Ordner liegt beispielsweise eine Datei „`vmlinux`“ herum. Der Aufruf von

```
file vmlinux
```

ergibt die Ausgabe:

```
vmlinux: ELF 32-bit LSB executable, ARM, version 1
(SYSV), statically linked, not stripped
```

Man sieht hier: Es ist eine ausführbare Datei für ARM-Prozessoren, die statisch gelinkt ist.

Power management options

Nicht nur der Prozessor bietet ein Power-Management an, auch Peripherie wie Arbeitsspeicher u.a. ist dafür vorbereitet, schlafen gelegt werden zu können. Das Betriebssystem legt vorher wichtige Daten im RAM ab. Beim Wecken werden diese Daten wieder aus dem Speicher geholt, um den alten Zustand wiederherzustellen. Das genaue Verhalten kann hier eingestellt werden.

Networking support

Jetzt kommen wir langsam zu den für uns Entwickler besonders interessanten Punkten. Unter „Networking Support“ werden alle Software-Stacks (Protokoll-Bibliotheken) für die Kommunikationsschnittstellen verwaltet. Stacks für z.B. Ethernet, TCP/IP, CAN, Infrarot, Amateur-Funk, Bluetooth, Wireless etc. können hier ein- und ausgeschaltet bzw. konfiguriert werden.

Device Drivers

Besonders wichtig für uns sind natürlich die Gerätetreiber für die Geräte. Unter diesem Menüpunkt findet man alles: vom einfachen I/O-Gerät bis hin zum modernsten Server-Highspeed-Bus. Im Quelltext des Kernels (Download unter [4]) befindet sich eine Vielzahl an USB-, I2C- und SPI-Geräten, auf die wir zurückgreifen können.

File systems

Das Dateisystem spielt in zwei Bereichen eine wesentliche Rolle. Zum einen muss das Betriebssystem selbst (auf der SD-Karte) in einem Dateisystem liegen, so dass der Kernel die für ihn wichtigen Daten lesen kann. Und zum anderen muss das Betriebssystem den Anwendungen Festplattenspeicher für Verzeichnisse und Dateien anbieten.

Kernel hacking

Diese Optionen dienen vor allem den Kernel-Entwicklern.

Security options

Linux bietet verschiedene Sicherheitsmechanismen an. Der bekannteste ist das klassische Benutzerrechte-System mit `Root` und den zusätzlichen Nutzern und Gruppen. Mit weiteren Methoden wie TPM-Modulen oder internen Sicherheitsmodellen können Anwendungen noch besser geschützt werden.

Cryptographic API

Kryptografie-Funktionen des Kernels.

Library routines

Funktionen für Prüfsummen-Verfahren und vieles andere mehr kann hier ausgewählt werden.

Möchte man schließlich die neue Kernel-Konfiguration abspeichern, kann man dies mit dem letzten Menüpunkt „Save an Alternate Configuration File“ machen. Nach Angabe des neuen Namens der Kernel-Konfiguration kann diese lokal im Quelltext des Kernels abgespeichert werden.

Wie man den Kernel übersetzt (`make zImage`) und auf die SD-Karte kopiert, wurde in der letzten Folge beschrieben [4]. Möchte man neue Module im Kernel nutzen, muss man zusätzlich diese übersetzen und installieren. Übersetzt werden die Module wie folgt:

```
make modules
```

Um die Module auf die SD-Karte zu kopieren, kann man entweder im folgenden Befehl direkt den Pfad zur eingehängten SD-Karte angeben, oder man installiert die Module nach „`/tmp`“ und kopiert den dort neu angelegten Ordner manuell als Benutzer `Root` bzw. mit `sudo` auf die SD-Karte. Man benötigt hier die `Root`-Rechte eigentlich nicht für das Kopieren. Vielmehr müssen die Daten so in das Dateisystem der SD-Karte geschrieben werden, dass der Kernel diese später als `Root` nutzen kann.

```
make INSTALL_MOD_PATH=/tmp modules_install
```

Nun wechseln wir in das Verzeichnis „`/tmp`“ und kopieren den Ordner „`lib`“ auf die SD-Karte:

```
cd /tmp
sudo cp -R lib /media/86b3be7-00f3-45e0-832e-1f48c2c3065e
```


Tabelle 1. Typische Gerätedateien im /dev/-Verzeichnis		
Gerätekategorie	Datei in /dev/	Character- oder Blockdevice
IDE-Festplatte	/dev/hda, /dev/hdb, ...	Block
SCSI-Festplatte	/dev/sda, /dev/sdb, ...	Block
RS232-Schnittstelle	/dev/ttyS0, /dev/ttyS1, ...	Character
Drucker	/dev/lp0, /dev/lp1	Character
Kamera	/dev/video0, /dev/video1	Block
Maus	/dev/input/mice	Character
RAM	/dev/ram	Block

Schnittstellen

Ursprünglich stammt Linux aus der PC-Welt. Dort hat man im Wesentlichen Schnittstellen wie IDE, SATA, PCI, AGP etc. In der Embedded-Welt liegt der Fokus auf den üblichen Mikrocontroller-Schnittstellen und natürlich USB. Zusammengefasst ist bei unserem Board vorhanden:

- USB (als Host und Device)
- UART (aktuell als Systemkonsole, kann aber auch als Schnittstelle für weitere Geräte verwendet werden)
- I2C (als Master)
- SPI (als Master)

Parallel bietet der Prozessor zudem Hardwarefunktionen wie:

- GPIO
- A/D-Wandler
- PWM-Ausgang

Nach und nach werden wir uns diese Schnittstellen genauer anschauen.

Gerätedateien

Um die Geräte aus den Anwendungen heraus ansprechen zu können, benötigt man eine Schnittstelle zum Treiber. Die Zugriffe auf Treiber werden unter den verschiedenen Betriebssystemen auf unterschiedliche Weise gelöst. Linux ist ein UNIX-Dateisystem, und unter UNIX ist das Konzept ganz einfach: Jedes Gerät wird als Datei behandelt.

Die Idee ist, dass man keine Spezial-Tools benötigt, um Geräte anzusteuern. Sondern dass man alles bequem mit kleinen, einfachen Programmen (von der Konsole aus) erledigen kann, über klassische Datei-Operationen. Wie das am Beispiel eines GPIO-Pins funktioniert, haben wir bereits im zweiten Artikel [5] anhand einer blinkenden LED gezeigt.

Im Wesentlichen gibt es zwei Befehle, mit denen man bereits viel erreichen kann: cat und echo. Der Befehl cat gibt den Inhalt einer Datei aus. Und mit echo kann man Daten in eine beliebige Datei schreiben; oder, genauer gesagt, man kann die Ausgabe von echo in eine beliebige Datei umlenken.

Möchte man nun zum Beispiel Daten auf eine Festplatte schreiben (direkt, nicht über das Dateisystem) oder Daten an der seriellen Schnittstelle ausgeben, kann man dies einfach mit echo machen: echo „Daten auf Festplatte“ > /dev/sda

oder

echo „Hallo Welt“ > /dev/ttyS0

Achtung: Wenn man testweise einmal direkt Daten auf eine Festplatte oder Speicherkarte schreiben möchte, sollte man dies unbedingt mit einem unbenutzten Speichermedium tun!

Bei den Gerätedateien muss man eigentlich nur zwischen Block- und Character-Devices unterscheiden. Eine Festplatte beispielsweise wird immer blockweise angesprochen, im Gegensatz zu einer RS232-Schnittstelle, welche ein klassisches Terminal ist, das Zeichen einzeln empfangen und verarbeiten kann. In der **Tabelle 1** sind typische Gerätedateien aufgelistet, wie sie in einem UNIX- oder Linux-System zu finden sind.

Eventuell hat sich der eine oder andere Leser bereits gefragt, was für ein Mechanismus hinter echo und cat steckt – wieso damit eigentlich ein Treiber etwas ausgibt bzw. annimmt. Nun, wenn alles eine Datei ist, dann kann man lesend oder schreibend auf diese Dateien zugreifen. Und das genau sind die Funktionen, die jeder Treiber anbieten muss: Read, Write und als dritte Ioctl (um Einstellungen am Gerät vornehmen zu können). Read und Write sind klassische Funktionen, um Dateien zu lesen oder zu schreiben. Das Vorgehen ist bei einem Gerät genau gleich wie bei einer Datei. Im ersten Schritt öffnet man das Gerät oder die Datei und erhält ein *Handle* (eine Zahl, welche die Datei oder das Gerät identifiziert). Mit Hilfe dieses *Handles* kann man dann über Read und Write mit dem Gerät oder der Datei arbeiten.

Kernel- und User-Space

Klassische Betriebssysteme – zu denen auch Linux gehört – sind meist so aufgebaut, dass es eine besondere Trennung zwischen einem Kernel-Speicherbereich und einem Speicherbereich für Anwendungen gibt. Eine Anwendung hat nicht die Möglichkeit, in den Kernel-Speicher zu greifen. Das ist eine weitere wichtige Information für den Entwickler, der sehr nahe am System arbeitet. In Linux gibt es den Kernelspace und den Userspace. Prinzipiell handelt es sich bei einem Gerätetreiber aber immer um eine Software, die im Kernelspace läuft. Wie kommt man dann überhaupt an empfangene Daten einer Schnittstelle?

Der Kernel bietet hierzu Funktionen wie copy_to_user [6], der Datenblöcke vom Kernelspace in den Userspace kopiert. Bei einem lesenden Zugriff auf eine Gerätedatei muss ein Treiber im Hintergrund diese Kernelfunktion aufrufen.

Integration eines Gerätetreibers

Jetzt wissen wir, wie ein Treiber im Kernel prinzipiell arbeitet. Doch

```

root@gnublin:~# cat /proc/devices
Character devices:
 1 mem
 4 /dev/vc/0
 4 tty
 4 ttyS
 5 /dev/tty
 5 /dev/console
 5 /dev/ptmx
 7 vcs
10 misc
13 input
21 sg
89 i2c
128 ptm
136 pts
153 spi
180 usb
189 usb_device
252 usbmon
253 lpc313x_pwm
254 lpc313x_adc

Block devices:
 1 ramdisk
259 blkext
 7 loop
 8 sd
65 sd
66 sd
67 sd
68 sd
69 sd
70 sd
71 sd
128 sd
129 sd
130 sd
131 sd
132 sd
133 sd
134 sd
135 sd
179 mmc

```

Bild 2. Liste der verfügbaren Treiber.

stellt sich jetzt noch die Frage, woher der Kernel die Information hat, dass z.B. mit `/dev/ttyS0` der Treiber für die serielle Schnittstelle angesprochen wird?

Hierfür gibt es sogenannte Major- und Minor-Nummern. Dies sind eindeutige Nummern, die in der Kernel-Dokumentation hinterlegt sind. Diese müssen dann auch im jeweiligen Treiber des Gerätes

```

root@gnublin:~# cd /dev
root@gnublin:~/dev# ls -l
total 1
crw-rw-rw- 1 root root 5, 1 Sep 27 06:02 console
prw-r--r-- 1 root root 0 Sep 27 06:02 initctl
srw-rw-rw- 1 root root 0 Sep 27 06:02 log
crw-r--r-- 1 root root 254, 0 Sep 26 03:57 lpc313x_adc
crw-r--r-- 1 root root 253, 0 Sep 26 04:21 lpc313x_pwm
lrwxrwxrwx 1 root root 12 Jan 1 1970 mtab -> /proc/mounts
crw-r--r-- 1 root root 1, 3 Dec 22 2011 null
crw-r--r-- 1 root root 89, 1 Sep 26 03:06 pca9555
crw-rw-rw- 1 root root 5, 2 Jan 1 1970 ptmx
drwxr-xr-x 2 root root 0 Jan 1 1970 pts
crw-r--r-- 1 root root 1, 9 Sep 26 04:28 random
brw-r--r-- 1 root root 8, 0 Sep 27 05:52 sda
brw-r--r-- 1 root root 8, 1 Sep 27 05:52 sda1
crw-rw-rw- 1 root root 5, 0 Jan 1 1970 tty
-rw-r--r-- 1 root root 24 Jan 1 1970 tty0
crw-rw-rw- 1 root root 4, 64 Sep 27 06:03 ttyS0
crw-r--r-- 1 root root 1, 9 Sep 26 04:30 urandom

```

Bild 3. Diese Dateien symbolisieren unsere Geräte.

stehen.

Welche Treiber mit welchen Major-Nummern aktuell verfügbar sind, kann man mit dem folgenden Aufruf ermitteln.

```
cat /proc/devices
```

Als Ausgabe erhält man eine Liste der Gerätetreiber, sortiert nach Character- und Blockdevices (**Bild 2**).

Anschließend kann man einmal den Ordner „`/dev`“ aufsuchen und `ls -l` eingeben, um den Inhalt dieses Ordners anzuzeigen (**Bild 3**). Der Ordner enthält eine Fülle von Dateien, die unsere Geräte symbolisieren. In der Ausgabe sieht man bei jeder Gerätedatei verschiedene Informationen. Ganz am Anfang steht ein „c“ oder „b“, was bedeutet, dass es sich um ein Block- oder Character-Device handelt. Anschließend sieht man die klassischen Rechte (dazu in einer späteren Folge der Serie mehr). Dann folgt der Benutzer, dem diese Datei gehört. Gleich dahinter sieht man zwei Zahlen, die durch ein Komma getrennt sind. Genau das sind die Major- und Minor-Nummern. Die Major-Nummern sind 1,4,5,8,89,253 und 254; die Minor-Nummern 0,1,2,3,9,12,24 und 64.

Greift man jetzt aus dem Userspace auf eine solche Datei zu, kann das Betriebssystem anhand der hinterlegten Nummer und der Information aus „`/proc/devices`“ genau feststellen, welcher Treiber gemeint ist.

Solche Gerätedateien kann man einfach mit dem Tool „`mknod`“ [7] anlegen. Das werden wir später noch gemeinsam machen.

USB-Seriell-Wandler integrieren

Es ist nun Zeit, die Theorie in die Praxis umzusetzen. Vom Linux-Board aus wollen wir einen einfachen USB/Seriell-Wandler (per USB) ansteuern (siehe **Bild 4**). Im Wesentlichen sind die folgenden Schritte zu machen:

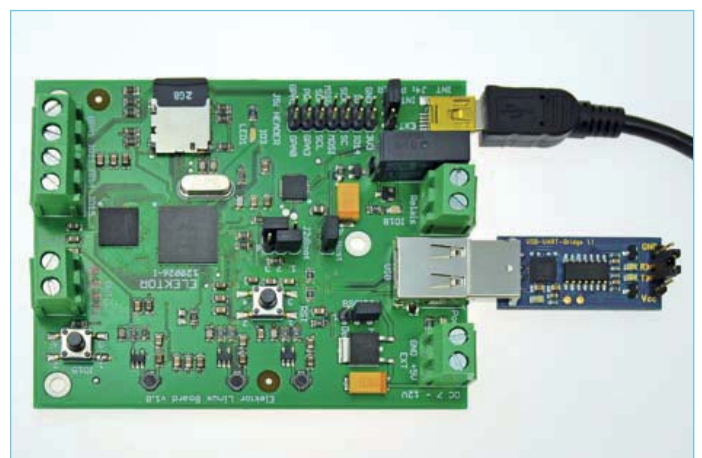


Bild 4. USB/Seriell-Wandler CP2102 an unserem Elektor-Linux-Board.

- Aktivierung des Treibers im Kernel
- Übersetzen des Kernels
- Kopieren des Kernels auf die SD-Karte
- Prüfen, ob der Treiber verfügbar ist
- Gerätedatei anlegen
- Das Gerät testen

Aktivierung des Treibers im Kernel

Wir wechseln zuerst in den Quelltextbaum von Linux:

```
cd ElektorLinuxBoardDownload_20120509/
```

```
cd linux-2.6.33-lpc313x/
```

```
./set.h
```

```
make menuconfig
```

Im ersten Schritt (**Bild 5**) wählt man den Menüpunkt „Device Drivers“ aus. Anschließend gibt es weiter unten einen Punkt „USB-support“ (**Bild 6**). In der Standardeinstellung ist hier ein „M“ anstelle wie im Screenshot ein „*“ zu finden. Mit der Leertaste kann man zwischen den drei Optionen: „M“ (als Modul), „*“ (fest in den Kernel) oder „nichts“ wählen. Wir wählen das „*“, um das Hantieren mit Modulen zu vermeiden.

Der nächste Menüpunkt ist der „USB Serial Converter support“; auch hier muss man aus dem „M“ ein „*“ machen (**Bild 7** und **Bild 8**). Mit der Enter-Taste kommt man in das Untermenü zur Auswahl des passenden Treibers (**Bild 9**). Viele Leser verwenden vermutlich Chips von FTDI oder den Controller CP210x von Silabs, daher setzen wir bei beiden Treibern ein „*“. Um die Änderung zu übernehmen, geht man überall auf „Exit“, bis man schließlich auf der letzten Seite das Speichern der neuen Konfigurationsdatei bestätigen muss.

Übersetzen des Kernels

Jetzt kann man den Übersetzungsvorgang starten:

```
make zImage
```

Als Ergebnis erhält man die Ausgabe in **Bild 10**.

Kopieren des Kernels auf die SD-Karte

Die SD-Karte aus dem Board steckt man nun in einen SD-Karten-Leser und schließt diesen an den PC an. Nun wird der Kernel auf die SD-Karte kopiert.

```
sudo cp arch/arm/boot/zImage /  
media/386b3be7-00f3-45e0-832e-1f48c2c3065e/
```

Nach dem Kopieren sollte man das Dateisystem manuell aushängen, um sicherzustellen, dass alle Blöcke sicher auf die SD-Karte übertragen wurden [4].

```
umount /media/386b3be7-00f3-45e0-832e-1f48c2c3065e
```

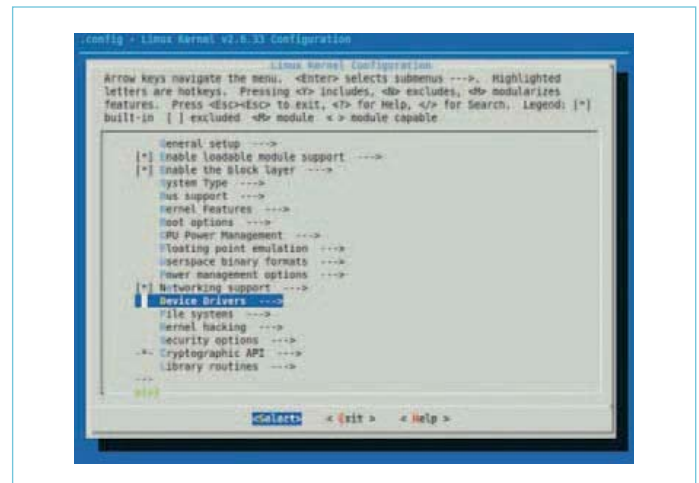


Bild 5. Menüpunkt „Device Drivers“.

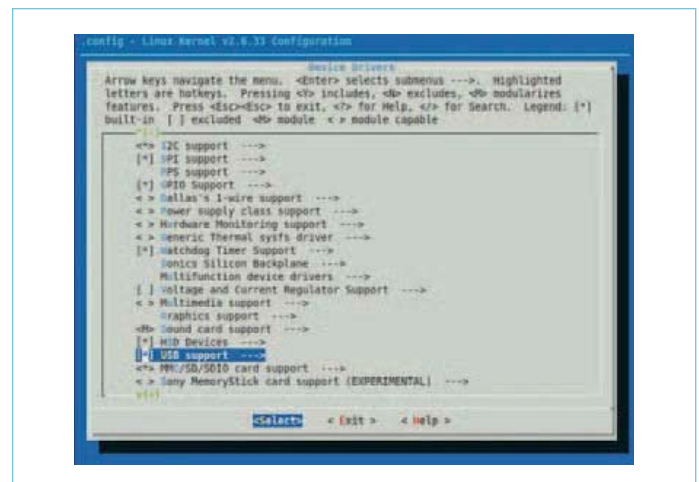


Bild 6. Menüpunkt „USB support“.

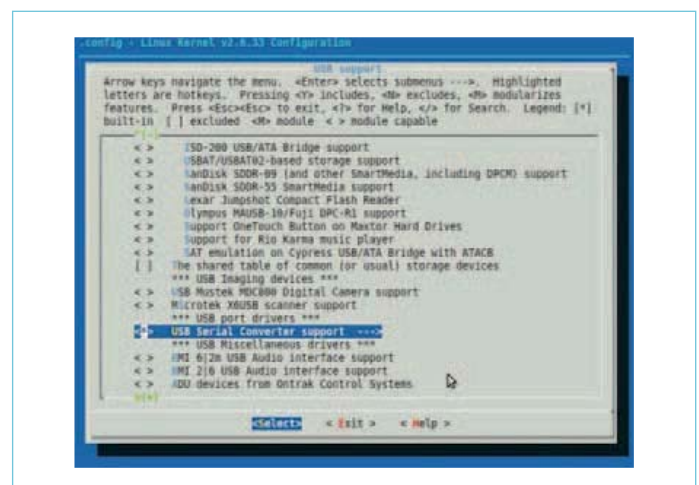


Bild 7. Auswahl „USB Serial Converter support“ als Bestandteil des Kernels.

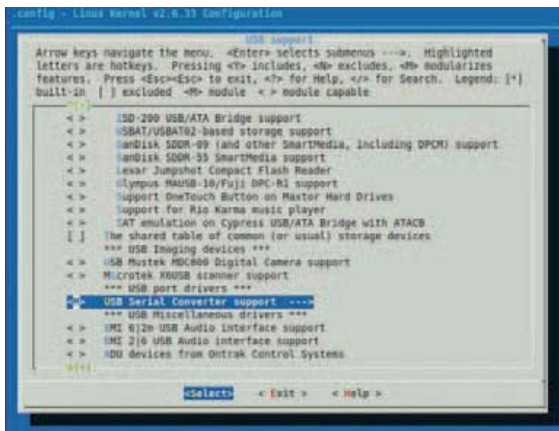


Bild 8. Auswahl „USB Serial Converter support“ als nachladbares Modul.

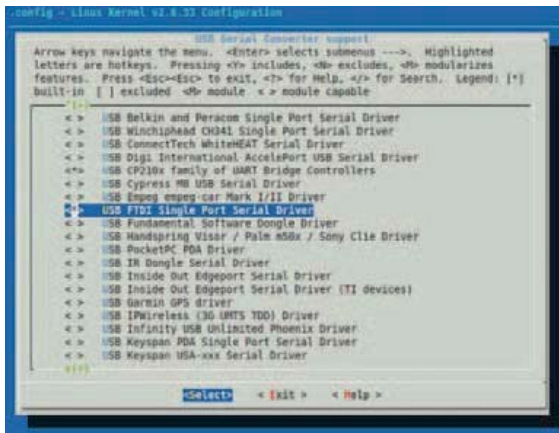


Bild 9. Hier wählt man den verwendeten USB/Seriell-Adapter aus.

Prüfen, ob der Treiber verfügbar ist

Jetzt kann man das Elektor-Linux-Board wie gewohnt starten. Wichtig ist dabei, dass der Jumper JP1 gesteckt ist! Dieser erzwingt beim Booten, dass die USB-Schnittstelle im Host-Modus gestartet wird. Außerdem muss das Gerät am USB-Anschluss vom Linux-Board aus mit Strom versorgt werden. Den Jumper 3 bringt man dazu in die Position 3-2.

Während des Bootens sieht man nun die erste Änderung. Es tauchen die beiden neuen Treiber auf:

cp210x: v0.09:Silicon Labs CP210x RS232 serial adaptor driver

USB Serial support registered for FTDI USB Serial Device

Gerätedatei anlegen

Nach dem Aufruf von `cat /proc/devices` sieht man jetzt einen neuen Eintrag: „188 ttyUSB“.

Für diese Major-Nummer muss man jetzt eine Gerätedatei anlegen:

```
mknod /dev/ttyUSB0 c 188 0
```

Das Gerät testen

Steckt man den USB/Seriell-Adapter jetzt an das Board an, dann erscheint folgende Ausgabe:

```
usb 1-1: New USB device found, idVendor=10c4, idProduct=ea60
```

```
usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
```

```
usb 1-1: Product: CP2102 USB to UART Bridge Controller
```

```
usb 1-1: Manufacturer: Silicon Labs
```

```
usb 1-1: SerialNumber: 0001
```

```
cp210x 1-1:1.0: cp210x converter detected
```

```
usb 1-1: reset full speed USB device using lpc-ehci and address 2
```

```
usb 1-1: cp210x converter now attached to ttyUSB0
```

In der letzten Zeile sieht man, dass das Anlegen der Gerätedatei geklappt hat. Der Kernel weist dem frisch erkannten Gerät die korrekte Gerätedatei zu.

Unser Linux-Betriebssystem bringt ein kleines Terminalprogramm namens „microcom“ mit. Mit Hilfe einer Brücke zwischen RX und

Weblinks

- [1] sauter@embedded-projects.net
- [2] http://en.wikipedia.org/wiki/Operating_Systems:_Design_and_Implementation
- [3] <http://de.wikipedia.org/wiki/Prozessstabelle>

- [4] www.elektor.de/120180
- [5] www.elektor.de/120146
- [6] www.gnugeneration.com/mirrors/kernel-api/r4299.html
- [7] <http://linuxwiki.de/mknod>

```
CC      arch/arm/boot/compressed/misc.o
In file included from arch/arm/boot/compressed/../../../../lib/decompress_inflate.c:10:0,
      from arch/arm/boot/compressed/misc.c:248:
arch/arm/boot/compressed/../../../../lib/zlib inflate/inflate.c: In function 'zlib fixedtables':
arch/arm/boot/compressed/../../../../lib/zlib inflate/inflate.c:100:1: warning: the frame size of 2176 bytes is larger than 1024 bytes
SHIPPED arch/arm/boot/compressed/lib1funcs.o
AS      arch/arm/boot/compressed/lib1funcs.o
LD      arch/arm/boot/compressed/vmlinux
OBJCOPY arch/arm/boot/zImage
Kernel: arch/arm/boot/zImage is ready
```

Bild 10. Ausgabe beim Übersetzen des (erweiterten) Kernels.

TX (Jumper) am USB/Seriell-Adapter kann man den Treiber hiermit testen. Hierzu rufen wir das Terminalprogramm mit folgenden Parametern auf:

```
microcom -s 9600 /dev/ttyUSB0
```

Gibt man jetzt Buchstaben ein, so werden diese zum Adapter geschickt und von dort zum Board zurückgesandt. Wenn der Treiber funktioniert, dann müssten diese Buchstaben jetzt in der Ausgabe erscheinen. Unterbricht man die Verbindung zwischen RX und TX am USB/Seriell-Adapter, dann erscheinen die eingegebenen Buchstaben nicht mehr. So sehen wir, dass der Gerätetreiber funktioniert. Mit STRG-x beendet man microcom wieder.

Auf die gleiche Art und Weise können wir jetzt LAN- und WLAN/USB-Adapter, USB-Soundkarten und viele weitere USB-Geräte verwenden.

Im nächsten Teil werden wir weitere Geräte und Schnittstellen anschauen und diese mit kleinen Versuchen testen!

(120181)

Labyrinth - die Lösung!

In der Sommerdoppelausgabe präsentierte Elektor außer dem traditionellen jährlichen Spezial-Hexadoku noch mehr Rätselhaftes. Verschlungene Wege waren zu gehen, denn hier wandelten *Elektroniker im Labyrinth!* Ein passives Netzwerk aus Dioden, Zenerdioden, Widerständen, Induktivitäten, Kondensatoren und Schaltern hatte drei Eingänge und einen Ausgang. Die Rätselfrage: An welchen Eingang muss die Spannung 20 V gelegt werden, damit am Ausgang Strom fließt?

Einige hundert Leser fanden den Weg durch den Irrgarten, sie fanden den richtigen Pfad, denn sie legten die Spannung an **Eingang 2**.

Den Preis,
eine **Electronic Workstation - Desktop** im Wert von **1000 €**
plus ein
Protostation Advanced Breadboard im Wert von **150 €**, beide
gesponsert von **Matrix Multimedia**, hat gewonnen:

Evan Wasserman aus Lakewood, New Jersey, USA.

Die Anzahl der richtigen Einsendungen, die dieser Rätselfreund schätzte, kommt der tatsächlichen Anzahl am nächsten.
Herzlichen Glückwunsch!

(120537)gd



Stromzähler im Netz

Gateway für Stromzähler mit S0-Bus

Eine zentrale Aufgabe der Menschheit in naher Zukunft wird es sein, den CO₂-Ausstoß drastisch zu verringern, um die globale Erwärmung so niedrig wie möglich zu halten. Jeder kann und muss sein Scherflein beitragen, aber um dies zu tun, muss zuvorderst ermittelt werden, wieviel und an welcher Stelle elektrische Energie „verheizt“ wird. Und was spricht dagegen, eine clevere Technologie und das WWW zu Hilfe zu rufen, um den Stromverbrauch zu reduzieren?



Von Søren Mikkelsen, unterstützt von Hans Henrik Skovgaard (Dänemark)

Jeder normale Haushalt nutzt einen Mix von Energiequellen (darunter auch und gerade elektrische Energie) und verfügt über einen Zähler für die elektrische Energie oder über Arbeit, die der Haushalt dem Stromnetz entnimmt. Wie die gezählte Energiemenge angezeigt wird, hängt von der Technologie des Zählers ab. Ältere elektromagnetische Typen besitzen einen mechanischen Zähler und können nur lokal von einem Menschen abgelesen werden, neuere Stromzähler erfassen die Verbrauchsdaten elektronisch und senden die ermittelten Werte automatisch zum Stromversorger.

Manchmal kann es nützlich sein, mit einem Sub-Stromzähler den Energiebedarf eines einzelnen Gerätes im Haushalt zu ermitteln, beispielsweise einer Wärmepumpe. So kann man feststellen, ob sie korrekt arbeitet.

S0-Interface

Es gibt verschiedene Arten, den „Verbrauch“ elektrischer Energie zu messen und die Daten weiter zu reichen. Beliebte und recht preiswert sind kommerzielle digitale Stromzähler mit S0-Schnittstelle, die man mit einem speziellen Gateway (S0 rein, Ethernet raus) kombinieren kann. Das S0-Interface ist in der DIN 43864 definiert. Sie beschreibt ein Strom-Interface zur Übertragung von Impulsen von einem Messgerät zu einem den Tarif oder Preis anzeigenden Gerät. Die maximale Spannung, die an einen S0-Ausgang gelegt werden kann, beträgt 27 V DC, der maximale Laststrom des Ausgangs beläuft sich auf 27 mA. Ein Impuls ist typisch etwa 30 ms lang. Das S0-Interface wird nicht nur von Strom-, sondern auch beispielsweise von Gas- oder Wasserzählern genutzt. Die verbrauchte Energie-, Wasser- oder Gasmenge wird durch die Anzahl der Impulse repräsentiert. Im Falle elektrischer Energie sind dies 1000 Impulse pro Kilowattstunde.

Wenn man einen solchen S0-Stromzähler zur Messung einzelner Verbraucher (oder Gruppen) in den Sicherungskasten einbauen möchte, sollte (muss) man einen autorisierten Elektriker zu Rate ziehen.

Manchmal sind schon digitale Stromzähler mit S0-Interface im Zählerkasten vorhanden. Wenn der Zähler ein LC-Display besitzt und/oder eine LED auf der Front blinkt, ist die Wahrscheinlichkeit hoch, dass es sich um ein Gerät mit S0-Interface handelt.

Speicherung der Verbrauchsdaten

Das Zählen der Impulse reicht allerdings nicht aus, man benötigt auch einen Ort, an dem die gesammelten Daten gespeichert werden. Dies kann entweder lokal beim Stromkonsumenten geschehen oder auf einem Server außerhalb des Haushalts (in einer Cloud). Die Datenmenge und die dazu benötigte Bandbreite zur Speicherung der Verbrauchsdaten ist nach heutigen Standards ein Klacks.

Wir haben eine Open-Source-Lösung gewählt, um die Leistungs-Verbrauchsdaten zu speichern und anzuzeigen. Sie stammt vom Openenergymonitor-Projekt [1] und bildet ein System, das auf PHP und einer MySQL-Datenbank basiert. Ange-

sichts der guten Verfügbarkeit dürfte man kaum eine bessere Lösung finden. Einerseits ist sie open source, das heißt, man hat vollen Zugriff auf den Quellcode. Andererseits muss man kein Experte in Web-Programmierung zu sein, um sie zu verwen-

den. Man kann dieses System gegen eine geringe Gebühr auch bei einem Hosted-Service-Provider nutzen (www.emoncms.org), aber auch auf einem eigenen Server oder jedem Web-Provider, der PHP und MySQL unterstützt.

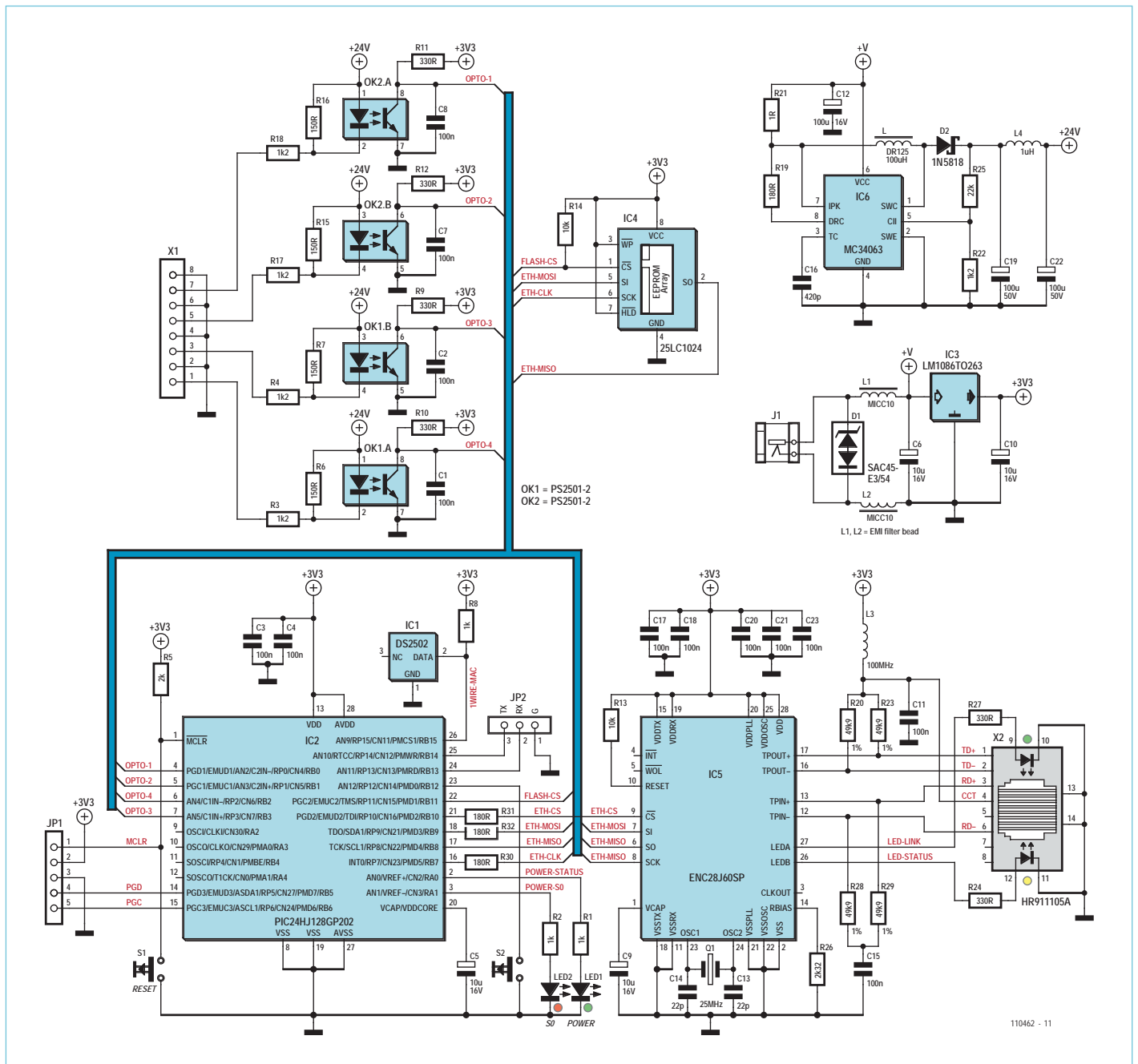


Bild 1. Die Schaltung des Puls0-Gateways für S0-kompatible Stromzählermodule weist vier S0-Eingänge auf.

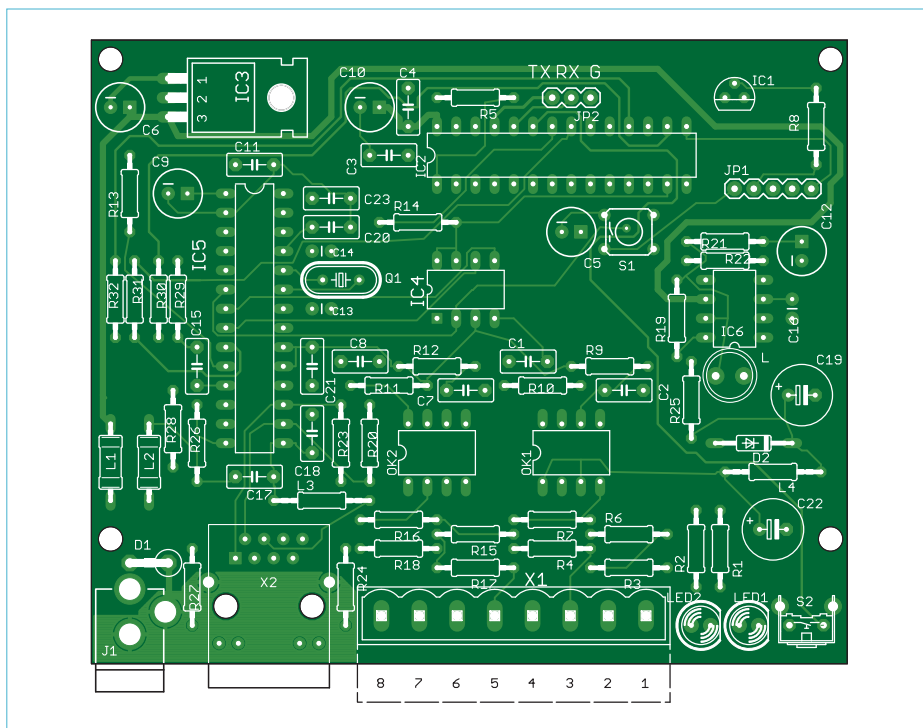


Bild 2. Die Platine für den „Stromzähler im Netz“ wird ausschließlich mit bedrahteten Bauteilen bestückt.

Das Puls0-Gateway für Stromzähler

Das Puls0-Gateway zeichnet sich aus durch vier S0-Eingänge:

- 5-V-Versorgung
- Ethernet-Verbindungsmöglichkeit
- Unterstützung des OpenEnergyMonitors EmonCMS
- Visuelle Anzeige der empfangenen Impulse

Das Schaltbild des Gateways ist in **Bild 1** zu sehen. Erstaunlicherweise werden – anders als in den meisten modernen Controllerdesigns – ausschließlich bedrahtete Bauteile und keine SMDs verwendet. Dies macht es auch Menschen mit dicken Daumen und müden Augen (den Autor eingeschlossen) möglich, die Schaltung ohne nennenswerte Probleme aufzubauen. Alle Prototypen der Schaltung wurden auf professionell gefertigten Platinen aufgebaut.

Das Puls0-Gateway besteht aus den Hardware-Blöcken Spannungsversorgung, lokaler Speicher, Mikrocontroller, S0-Interface und Ethernet-Interface. Werfen wir einen detaillierten Blick darauf:

Die **Spannungsversorgung** mit IC3 wandelt die Eingangsspannung in die benötigten 3,3 V, die vom lokalen Speicher, dem

Mikrocontroller und dem Ethernet-Interface genutzt werden. IC3 ist ein LM1086-3.3-Spannungsregler. Der Ethernet-Block benötigt immerhin bis zu 500 mA, so dass ein kleiner Kühlkörper für den Spannungsregler unverzichtbar ist.

Der **lokale Speicher** besteht aus IC4, einem seriellen (SPI-)EEPROM von Microchip. Er enthält nicht nur die lokalen Software-Einstellungen, sondern auch die Webseiten, die vom Anwender-Interface gebraucht werden. Ein EEPROM-Speicher von 1024 kbit ist groß genug, um die Webseiten inklusive Grafiken zu beherbergen. Natürlich ist der Speicherplatz zu klein für schicke Flash-basierte Konfigurationsseiten, aber mit einer JavaScript/CSS/HTML-basierten Lösung sollte man sich zufrieden geben.

Der **Mikrocontroller** IC2 stammt ebenfalls von Microchip. Es ist ein 16-bit-Controller mit der Bezeichnung PIC24HJ128GP202 (40 MIPS). Er besitzt 128 kB Flashspeicher und 8 kByte RAM und ist schnell genug für unser Webserver-Programm und andere TCP/IP-Funktionen.

Das **S0-Interface** um OK2A/B und OK1A/B benötigt eine höhere Betriebsspannung von etwa 24 V. Diese wird vom Step-up-Gleichspannungswandler IC6 erzeugt. Die Eingangsspannung bezieht der Wandler direkt aus dem Steckernetzteil des Gateways. Der Gleichspannungswandler besteht aus dem

bekannten und weltweit milliardenmal genutzten MC34063. Jedes S0-Interface ist vom Mikrocontroller durch einen Optokoppler isoliert.

Das **Ethernet-Interface** basiert auf IC5, einem Ethernet-Controller ENC28J60 von Microchip. Das Gateway verwendet einen Ethernet-Steckverbinder mit eingebautem Übertrager. Aber nicht alle solchen Ethernet-Verbinder sind geeignet. Es ist entscheidend, genau den hier gezeigten Typ zu verwenden, der ENC28J60 ist in dieser Hinsicht ziemlich pingelig. Zudem sind die verschiedenen anderen Verbinder auf dem Markt nicht pinkompatibel.

Konstruktion und Test

Bild 2 zeigt das Platinenlayout für das Puls0-Gateway. Es sind keine Schwierigkeiten beim Aufbau der Platine zu erwarten. Die Schaltung sollte funktionsblock-weise aufgebaut werden, beginnend mit der Spannungsversorgung. Dann sollte man sich davon überzeugen, dass jedes IC auch die richtige Versorgungsspannung erhält und keinerlei Kurzschlüsse vorliegen. Besonders sollte man bei der Montage des Spannungsreglers samt Kühlkörper darauf achten, dass keine Kurzschlüsse entstehen.

Wenn all dies sichergestellt ist, überprüft man die Spannung Vcap an Pin 14 des Ethernet-Controllers. Sie sollte bei rund 2,5 V liegen. Ist dies auch der Fall, kann man die Schaltung über ein Ethernet-Kabel und einem Netzwerkschalter mit einem aktiven Netzwerk verbinden. Die LEDs im Ethernet-Verbinder sollten nun blinken. Nun ist die Schaltung bereit zur Programmierung.

Software für das Puls0-Gateway

Ohne Software ist das Puls0-Gateway nicht von großem Nutzen. Die erforderliche Software wird in zwei Schritten in die Schaltung geladen. Zunächst lädt man die Firmware für den Mikrocontroller herunter, dann werden die Webseiten im EEPROM gespeichert. Alle Software-Bausteine des Projekts samt Platinenlayout (im Eagle-Format) können kostenlos von der Elektor-Website heruntergeladen werden [2].

Der Mikrocontroller PIC24HJ128GP202 lässt sich mit einem PICKIT3 (oder einem ähnlichen Programmieradapter) von Microchip programmieren. Vergessen Sie nicht, die



THE GLOBAL STAGE FOR INNOVATION



*Be the first to see
what tomorrow
will be
talking about.*

TUESDAY, JANUARY 8 — FRIDAY, JANUARY 11, 2013 * LAS VEGAS, NEVADA * REGISTER AT CESWEB.ORG



REGISTER NOW

Schaltung während des Programmiervorgangs „unter Strom“ zu halten. Damit das S0-Gateway korrekt arbeiten kann, müssen die Konfigurations-Webseiten im 1024-kbit-EEPROM gespeichert sein. Sie können mit der eingebauten Upload-Funktion der Gateway-Firmware hochgeladen werden. Diese Funktion ist über die URL <http://192.168.X.X/mpfsupload> erreichbar, wobei für 192.168.X.X die Adresse eingesetzt werden muss, die das Gateway im Netzwerk belegt.

Anwendung des Puls0-Gateways

Für ein erstes Setup verbindet man, wie in **Bild 3** gezeigt, den S0-Ausgang des Stromzählers mit dem Eingang S0-1 des Puls0-Gateways. Die Verbindung kann mit einem leichten Zweidrahtkabel vorgenommen werden, da die Spannung nur 24 V beträgt. Allerdings sollte die Kabellänge 20 m nicht überschreiten.

Dann verbindet man den Ethernet-Port des Gateways mit einem entsprechenden Anschluss am Router oder LAN-Switch, anschließend steckt man die 5-V-Spannungsversorgung in J1. Die grüne LED beginnt zu blinken. Jedes Mal, wenn ein Impuls vom Stromzähler eintrifft, blitzt die rote LED auf. Sollten keine Impulse detektiert werden, dreht man die Adern des S0-Kabels um (dies kann wegen einer Diode im Ausgang des S0-Interfaces erforderlich sein).

Lokalisieren des Web-Interfaces

Die Konfiguration des Puls0-Gateways wird über ein Web-Interface vorgenommen. Um das Web-Interface zu erreichen, muss dessen IP-Adresse bekannt sein. Es gibt zwei Wege, dieses Wissen zu erlangen:

- mit einer Discovery-Applikation (Wizard), um die IP-Adresse zu finden
- mit der Liste der DHCP-IP-Clients im Router, wenn dynamische IP-Adressen vergeben werden.

Konfiguration des Puls0-Gateways

Wenn die Verbindung zum Web-Interface des Puls0-Gateways einmal steht, müssen die Einstellungen für den OpenEnergyMonitor-EmonCMS-Server angewendet werden (siehe **Bild 4**).

Detaillierte Anweisungen, wie das EmonCMS-Datalogging-System einzustel-

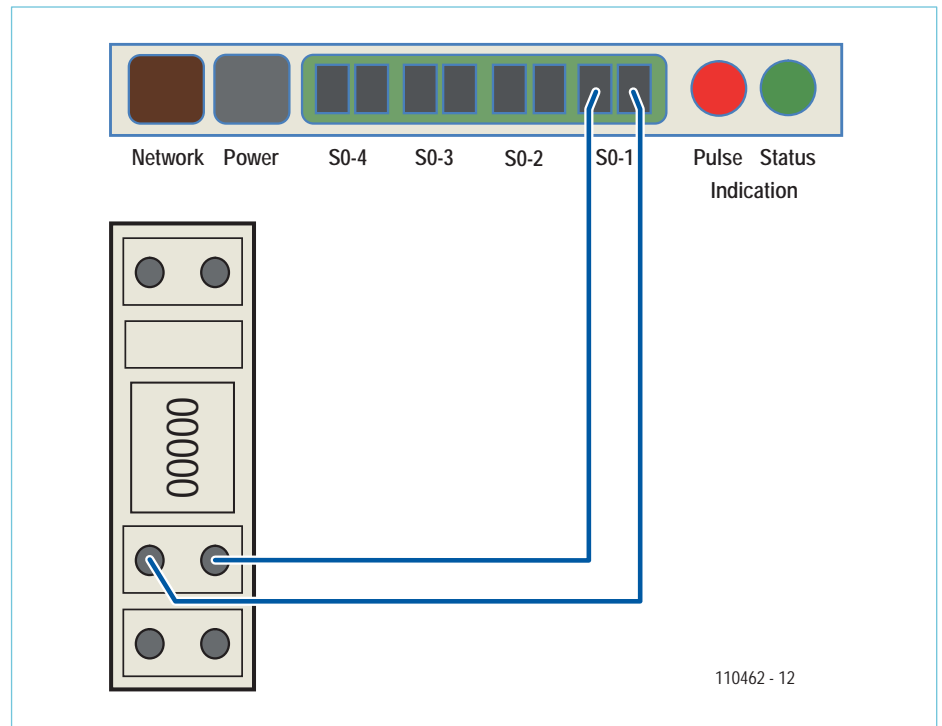


Bild 3. Simple Zweidrahtverbindung eines S0-kompatiblen Stromzählers mit dem Gateway.

len ist, findet man auf der Website von Openenergymonitor.

Zum Schluss

Obwohl das Projekt für die OpenEnergyMonitor-Plattform ausgelegt ist, lässt sich die Firmware so modifizieren, dass sie auch mit anderen Online-Datenspeicherdiensten funktioniert.

(110462)

Weblinks

- [1] www.openenergymonitor.org
- [2] www.elektor.com/110462



Bild 4. Konfiguration des OpenEnergyMonitor-Servers.

elektorLive! 2012

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

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

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

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

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

Geplante Seminare (Änderungen vorbehalten):

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

Datum:

Samstag, 20. Oktober 2012

Zeit:

09:00 Uhr – 17:00 Uhr

Ort:

Kongresszentrum
Westfalenhallen Dortmund

Eintritt:

25,00 € für Elektor-Abonnenten

45,00 € für Nicht-Abonnenten

20,00 € für Studenten/Schüler

Der Eintritt berechtigt zur Teilnahme an 2 Seminaren.

Tagesablauf:

09:00 Uhr – 17:00 Uhr: Ausstellung

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

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

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



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



Arduino auf Kurs (2)

Teil 2: Fernauslesen von Sensoren per Mobilfunk

Diesen Monat werden die Möglichkeiten der Datenkommunikation per Mobilfunk für eigene Projekte beleuchtet. Dabei werden Konzepte wie IoT (Internet of Things) oder M2M (Machine to Machine) gestreift. In kurzer Zeit kann man eine Menge erreichen!



Von David Cuartielles (Spanien)

Einkaufsliste

Zunächst eine Liste der Hardware, die man zum Nachvollziehen der hier beschriebenen Experimente benötigt:

- Arduino Uno,
- Arduino-GSM-Shield (zusammen mit Telefonica I+D konzipiert),
- SIM-Karte für Mobilfunk,
- Handy mit eigener SIM-Karte,
- TinkerKit-Shield,
- TinkerKit-LED-Modul (x3) und
- TinkerKit-Slider-Modul.

Die letzten drei Positionen können auch durch eine Experimentierplatine und einige Bauteile ersetzt werden. Doch da es hauptsächlich um die Funktion der GSM-Library für Arduino geht, habe ich mich für fertige Module entschieden. „GSM“ wird übrigens im nichtdeutschen Sprachraum oft synonym für Mobilfunk verwendet. Bevor Sie weiterlesen empfiehlt es sich, zunächst den in dieser Folge verwendeten Code herunterzuladen [5], denn aufgrund seiner Länge werden im Artikel nur Auszüge davon abgedruckt.

Einführung in M2M und IoT

Kurz gesagt handelt es sich bei IoT um ein Paradigma der Informatik, an dem in den letzten Jahren schon viel in den Forschungsabteilungen größerer Firmen und Universitäten auf der ganzen Welt gearbeitet wurde. Dabei wurde versucht, die Idee von der Vernetzung von allem und jedem weiter voran zu bringen. Von der Mikrowelle über das Auto bis zur Haustüre und zum Fahrrad sollen alle Geräte und Installationen des täglichen Bedarfs über das Internet Daten aus-

tauschen und abfragbar sein, um neue und bessere Wege und Lösungen zu erschließen.

Die schwedische Kommunikationsfirma Ericsson stellte letztes Jahr ihre Zukunftsvision in einem Memo [1] dar, wonach bis zum Jahre 2020 etwa 50 Milliarden Geräte „im Internet“ sein werden. Das würde für die entwickelten Regionen der Erde bedeuten, dass fast jedes Gerät im Haushalt einen Internetanschluss hat, vom Kühlschrank über Mikrowelle, Wecker, Fernseher und Backofen bis hin zum Aufzug. Genau so geht es dann den Maschinen und Apparaten draußen: Auch Autos, die Straßenbeleuchtung und Einkaufswagen sind vernetzt.

Der Terminus M2M betrifft die reine Verbindung zwischen Geräten, unabhängig ob per Funk, Kabel oder hybrid. In einer gewissen Weise ist M2M die Basis von IoT, da M2M die Infrastruktur dafür bietet, dass gesammelte Daten über das Internet an einen anderen Ort übertragen werden können. Gelegentlich wird M2M auch noch synonym zu Telemetrie verwendet, aber in den letzten Jahren hat sich die Bedeutung eher auf funkbasierte Kommunikation verdichtet, und zunehmend ist damit hauptsächlich die Kommunikation per Mobilfunk gemeint, die für Fernsteuerzwecke eingesetzt wird. Grob vereinfacht geht es bei IoT um den Internetanschluss, wobei dieser mit Hilfe von M2M und Mobilfunkfähigkeiten realisiert wird. Im Internet finden sich sehr viele Artikel und Videos mit dem gleichen Ansatz, was auch einer der Gründe dafür ist, dass der Mobilfunk bzw. GSM sich aktuell zum größten Netzwerk mausert.

Ich gehe davon aus, dass Sie diesen Artikel lesen, weil Sie vorhaben, ein Gerät zu vernetzen. Und wenn Sie sich die Frage stellen, was wohl das beste Verfahren ist, etwas zwecks Datenabruf irgendwo auf der Welt ans Internet anzuschließen, dann dürfte die Antwort klar sein: Das ist der Mobilfunk, da er (fast) überall funktioniert.

GSM/GPRS-Shield für Arduino

Mit Arduino kann man den Bau von Prototypen vereinfachen und trotzdem eine Menge dabei lernen. Von daher liegt die Frage nahe, wie sich diese Strategie für das IoT einsetzen lässt. Ein Weg wäre M2M, und hierzu wurde Kontakt mit dem *Physical Internet Laboratory* der Firma Telefonica aufgenommen.

Bei der Zusammenarbeit ging es um die Entwicklung eines offiziellen Arduino-Shields, mit dem eine Kommunikation per GSM/GPRS (auch unter 1G und 2G bekannt) realisiert werden kann. Es wurde eine Library erstellt, die das Verschicken von Texten (SMS) sowie das Anrufen oder aber das Veröffentlichen von Daten auf einer Webseite vereinfachen soll.

Das Ganze ist natürlich open source und die Software kann einfach für andere Shields (mit anderen AT-Befehlen) portiert werden. Mehr dazu später. Bei einem mit einer SIM-Karte (z.B. von Telefonica) bestückten Shield ist das Gerät zum SMS-Empfang und -Versand, zur Annahme und zum Durchführen von Anrufen sowie zur Verbindung mit dem Internet per GPRS in der Lage.

Hinweis: Bei diesem Shield gibt es **keine** Providersperre. Man kann also jede beliebige SIM-Karte verwenden, die physikalisch passt. Idealerweise nimmt man eine SIM-Karte mit Datendiensten. Wenn man nur per SMS kommunizieren will, empfiehlt sich die Verwendung einer Karte des gleichen Providers wie beim empfangenden Handy. Wenn Sie die beim Board mitgelieferte Karte verwenden wollen, folgen Sie bitte der beiliegenden Beschreibung zur Online-Aktivierung.

AT-Befehle

Modems sind die grundlegenden Geräte, mit denen Daten in Netzwerke kommen. Zur Steuerung von Modems hat die Firma Hayes [2] im Jahre 1981 die so genannten AT-Befehle eingeführt, mit denen zwischen Kommando- und Datenmodi umgeschaltet werden kann. Der Befehls-Modus erlaubt die Ausführung bestimmter Aktionen wie die Einstellung der Rufnummer, der Baudrate (Datengeschwindigkeit) etc.

Der von Hayes konzipierte Befehlssatz etablierte sich schnell als eine Art Industrie-Standard für die Kommunikation über serielle Kanäle. Dieser Standard wird auch heute noch in vielen Geräten eingesetzt. Befehle beginnen immer mit dem String „AT“ (= Attention) gefolgt von den Zeichen des jeweiligen Befehls samt Argumenten. Das gesteuerte Modem interpretiert diese AT-Befehle dann spezifisch. Im Daten-Modus überträgt dann das Modem lediglich Daten vom und ins Internet.

Beim GSM/GPRS-Shield ist ein Funkmodem von Quectel [3] vorgesehen, das ebenfalls mit AT-Befehlen gesteuert wird. Die meisten Befehle sind modemspezifisch. Die Library allerdings ist so konzipiert, dass sie sehr einfach an Modems anderer Hersteller angepasst werden kann. Der Hauptzweck der Library besteht in der Abstraktion von der Komplexität des Modems für die eigene Programmierung. Aus diesem Grund wird darauf nicht weiter eingegangen. Wenn man aber die GSM-Library im Debug-Mode verwenden will, stößt man auf viele dieser Befehle.

Installation der GSM-Library

Als dieser Artikel geschrieben wurde, hatte die Arduino-IDE die Version 1.0.1 und enthielt die hier besprochene GSM-Library noch nicht. Wenn Sie also im IDE-Menü „Sketch / Import Library“ keine Library mit der Bezeichnung „GSM3“ in der Liste stehen haben, dann müssen Sie die Library von der offiziellen Arduino-Webseite zu die-

sem Projekt [4] herunterladen und sie (so wie unter dem Link „Library“ angegeben) installieren. Das Hinzufügen einer neuen Library zur IDE erfolgt durch Erstellen eines Ordners mit dem Namen „libraries“ in Ihrem Sketchbook, in den dann der entpackte Inhalt der Datei von [4] hinein kommt. Anschließend wird die Arduino-IDE neu gestartet und die Library zeigt sich im oben beschriebenen Menü. Zusammen mit der Library wird nicht nur der Code zur Ausführung bestimmter Befehle, sondern zugleich ein Satz an Beispielen installiert. Diese beinhalten:

- Test der korrekten Funktion des Modems,
- Senden/Empfang von SMS,
- Annahme/Absetzen eines Anrufs,
- Öffnen einer TCP/IP-Verbindung samt Datentransfer über das Internet via GPRS,
- Versenden/Sammeln von Sensor-Daten von und zu Cosm (ein Online Datenservice) und
- Sammeln/Anzeigen von Twitter-Messages.

Der Zugriff auf diese Beispiele ist simpel: einfach per Menü zu „File / Examples / GSM3“ navigieren.

Statusüberprüfung

Vermutlich überprüfen Sie zuerst die korrekte Funktion der Hardware und checken, ob die eingesetzte SIM-Karte funktioniert. Hierzu öffnet man per Menü das Beispiel „File / Examples / GSM3 / Tools / TestModem“:

```
// libraries
#include <GSM3ShieldV1ModemVerification.h>

// modem verification object
GSM3ShieldV1ModemVerification modemTest;

// IMEI variable
String IMEI = "";

// serial monitor result messages
String oktext = "OK";
String errortext = "ERROR";

void setup()
{
    // initialize serial communications
    Serial.begin(9600);

    // start modem test (reset and check response)
    Serial.print("Starting modem test...");
    modemTest.begin();
    Serial.println(oktext);
}
```

```
void loop()
{
  // get modem IMEI
  Serial.print("Checking IMEI...");
  IMEI = modemTest.getIMEI();

  // check IMEI responded
  if(IMEI != NULL)
  {
    Serial.println(oktext);
    // show IMEI in serial monitor
    Serial.println("Modem's IMEI: " + IMEI);
    // reset modem for check booting
    Serial.print("Resetting modem...");
    modemTest.begin();
    // get and check IMEI one more time
    if(modemTest.getIMEI() != NULL)
    {
      Serial.println(oktext);
      Serial.println("TEST COMPLETE!");
    }
  }
  else
  {
    Serial.println(errortext);
  }
}
else
{
  Serial.println(errortext);
}
while(true);
}
```

Mit diesem Code kann man überprüfen, ob das Modem von Arduino erkannt wird. Hierzu wird eine Verbindung zum Funkmodem auf dem GSM-Shield aufgebaut und dann versucht, die IMEI (International Mobile Equipment Identity) zu lesen. Diese Nummer ist übrigens für jedes Gerät im Mobilfunknetz einzigartig und jedes Handy gibt sich mit dieser Nummer zu erkennen. Wenn das klappt, sollte man nach dem Laden des Beispiels auf Arduino Uno und dem Öffnen des „Serial Port Monitors“ etwas wie in **Bild 1** lesen können.

Wenn Ihr Modem die korrekten Lebenszeichen von sich gegeben hat, empfiehlt sich ein Test der Verbindung mit dem Mobilfunknetz. Hierzu öffnet man das Beispiel „File / Examples / GSM3 / Tools / TestGPRS“. Der Code-Abdruck erübrigt sich, da er sehr ähnlich zum vorhergehenden ist. Es wird überprüft, ob eine Verbindung mit dem GPRS-Netzwerk möglich ist, und auch, ob die Einstellungen korrekt sind. Zumindest mit der beigelegten SIM-Karte funktioniert das.

Hinweis: Mit dem GSM/GPRS-Shield nutzt man genau diese Technologie und eben kein 3G-Netz. 3G- und GSM-SIM-Karten sehen zwar gleich aus, aber bei manchen Betreibern wird kein 2G bzw. GPRS (mehr) angeboten.

Die Verbindung mit einem GPRS-Netz erfordert etwas mehr an Konfiguration als das bloße Versenden von Text via SMS. Es müssen vier Parameter konfiguriert werden:

- APN: Dies ist ein Akronym für Access Point Name, über den das Modem des Shields ins Internet kommt. Er ist spezifisch für den Netzbetreiber, über den man den Zugang zum Mobilfunk erhält.

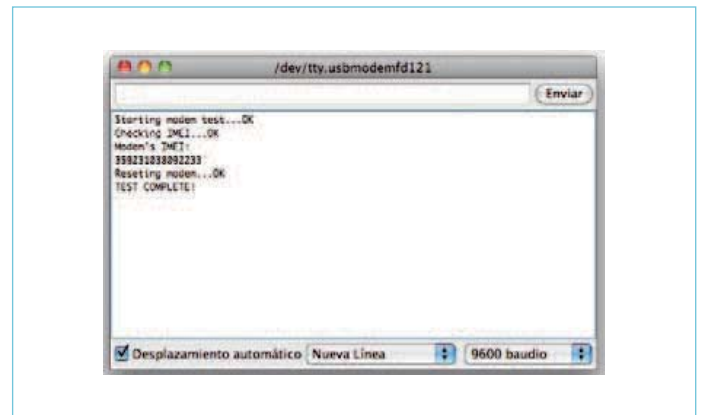


Bild 1. Positives Resultat des TestModem-Beispiels.

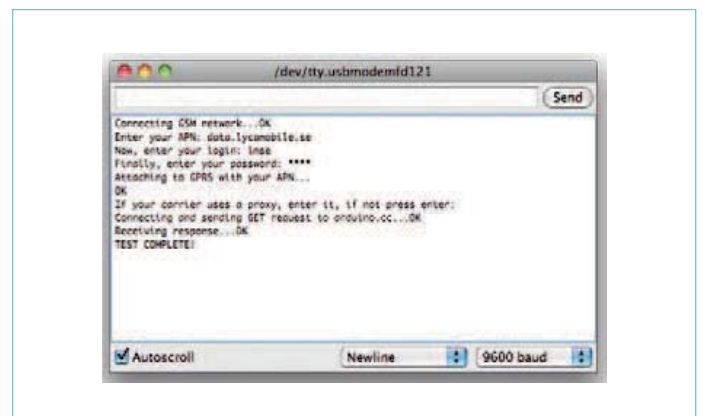


Bild 2. Konfiguration einer GPRS-Verbindung mit dem Serial-Monitor.

- login: Einige Betreiber erwarten ein Login.
- password: Einige Betreiber erwarten ein Passwort.
- proxy: Dies ist die Adresse eines Servers, der alle Kommunikation im Netzwerk des Betreibers kanalisiert. Dieser Parameter wird häufig nicht genutzt.

Das Beispiel erfragt alle für die Verbindung mit dem Netzwerk notwendigen Informationen.

Tipp: Denken Sie dran, die Option „Newline“ im Dropdown-Menü unten auf dem Serial-Port-Monitor zu aktivieren. In diesem Fall wird dann bei einem „Send“ im Monitor vom System ein EOL-Zeichen (End-Of-Line) eingefügt, das dem Modem signalisiert, dass hier die Information zu Ende ist.

Wenn es keine Probleme gibt, dann sollte das Fenster so wie in **Bild 2** aussehen. Die Meldungen bedeuten, dass das Modem sich mit Hilfe der SIM-Karte mit dem Internet verbinden kann. Doch selbst wenn keine Verbindung mit dem Internet möglich sein sollte, kann man immer noch Text in SMS-Form verschicken oder empfangen und Anrufe tätigen oder entgegennehmen.

Text per SMS

Das Senden oder Empfangen von Text könnte einfacher kaum sein. Die folgenden Code-Zeilen verschicken einen String an eine bestimmte Telefonnummer:

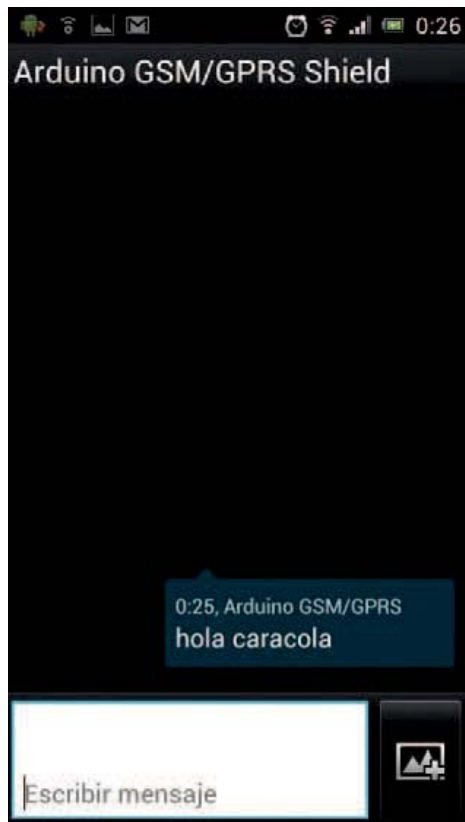


Bild 3. Screenshot der Textmeldung auf meinem Smartphone.

```
// send the message
sms.beginSMS("0755442200");
sms.print("hola caracola");
sms.endSMS();
```

Die angegebene Telefonnummer ist natürlich nur ein fiktives Beispiel. Das Ergebnis stellt sich auf dem Bildschirm meines Handys so wie in **Bild 3** dar.

Zur Library gehören Beispiele zum Senden und zum Empfang. Beispielsweise sendet der Code unter „*File / Examples / GSM3 / Tools / SendSMS*“ eine SMS an ein Handy. Zuerst wird im Serial-Port-Monitor nach der Ziel-Rufnummer und dann nach dem zu sendenden String gefragt. Einfach mal ausprobieren!

Fernsteuerung via Text (SMS)

Zur Fernsteuerung eines Geräts muss dieses Text empfangen können, die empfangenen Daten parsen und gegebenenfalls adäquat auf die Befehle reagieren. Für einen ersten Test eignet sich ein Prototyp mit je einer roten, grünen und blauen LED sehr gut. Statt LEDs könnte man natürlich auch Relais etc. ansteuern und so zum Beispiel auch Geräte aus der Ferne ein- und ausschalten. Das Programm liest hierfür das erste Zeichen der SMS und schaltet die entsprechende LED ein. R steht für die rote, G für die grüne und B für die blaue LED.

Wie Sie sehen können, ist das Programm sehr einfach. Es gibt keinen komplexen Parser, denn es wird sowieso nur das erste Zeichen der SMS abgefragt und ausgewertet. Anschließend wird die SMS mit dem Befehl `sms.flush()` aus dem Speicher gelöscht. Der folgende Code-Abschnitt ist Teil des Beispiels „*DecodeSMS*“ aus dem Download [5] zu diesem Artikel.

```
// If there are any SMSs available()
if (sms.available())
{
    Serial.println("Message received from:");

    // Get remote number
    sms.remoteNumber(remoteNumber, 20);
    Serial.println(remoteNumber);

    // This is just an example of message disposal
    // Messages starting with # should be discarded
    if(sms.peek()=='#')
    {
        Serial.println("Discarded SMS");
        sms.flush();
    }

    // Read message bytes and print them
    int count = 0;
    while(c=sms.read()) {
        if(!count) {
            digitalWrite(pinR, LOW);
            digitalWrite(pinG, LOW);
            digitalWrite(pinB, LOW);

            switch(c) {
                case 'R':
                    digitalWrite(pinR, HIGH);
                    break;
                case 'G':
                    digitalWrite(pinG, HIGH);
                    break;
                case 'B':
                    digitalWrite(pinB, HIGH);
                    break;
            }
        }
        count++;
    }

    Serial.println("\nEND OF MESSAGE");

    // delete message from modem memory
    sms.flush();
    Serial.println("MESSAGE DELETED");
}
```

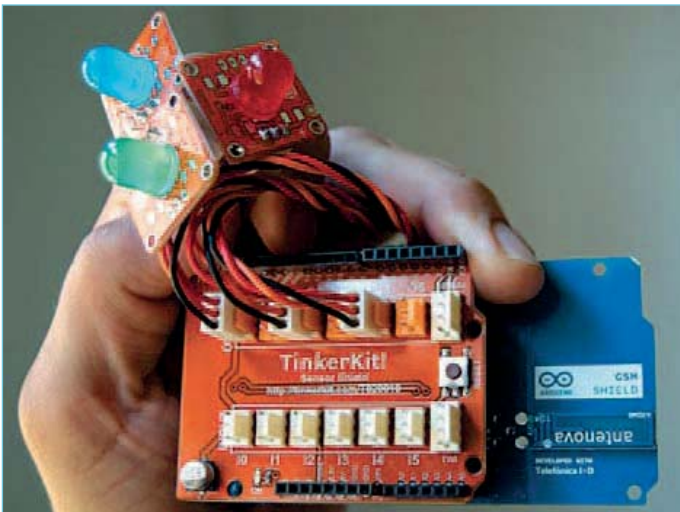



Bild 4. Arduino Uno + GSM/GPRS Shield + TinkerKit Shield + TinkerKit LED.

Bild 4 zeigt die bei diesem Projekt verwendete Hardware.

GPRS-Betrieb

Nun geht es um die Abfrage eines analogen Sensors – in diesem Fall um ein Poti – und die Übertragung der Daten an eine Webseite. Auf der Serverseite wird Apache und ein sehr simples PHP-Skript eingesetzt, das die vom Board gesendeten Daten empfängt. Ein Teil des Skripts erneuert dann den HTML-Code. In meinem Sprachgebrauch läuft dieses Verfahren unter „Online Memory Cell“. Das PHP-Skript kann Daten entgegennehmen und anzeigen. Man kann diese Webseite auch per Browser darstellen und so die Änderungen über der Zeit beobachten. Man benötigt die gleiche APN/login/pass-Kombination, die auch im vorherigen Beispiel eingesetzt wurde. Das Beispiel zeigt die Übertragung eines Bytes einmal in zehn Sekunden:

```
void loop()
{
  client.connect(server, 80);
  Serial.println("sending data...");
  client.print("GET ");
  client.print(path);
  client.print(query);
  client.print(analogRead(A0)); // take the value
                                // on A0 and send it
  client.println(" HTTP/1.1");
  client.print("HOST: ");
  client.println(server);
  client.println();
  client.stop();
  delay(10000);
}
```

Um die vom Arduino-Board geschickten Daten zu lesen, benötigt man einen Server für den PHP-Code. Der Server muss öffentlich zugänglich sein, damit das Arduino-Board darauf Informationen ablegen kann, die man per Browser eines beliebigen internetfähigen Geräts lesen kann.

Wichtig: Denken Sie daran, dass dieser Code reinen Demonstrationszwecken dient. Er ist keineswegs sicher. Von daher sollten Sie das PHP-Skript nach dem Ausprobieren wieder löschen.

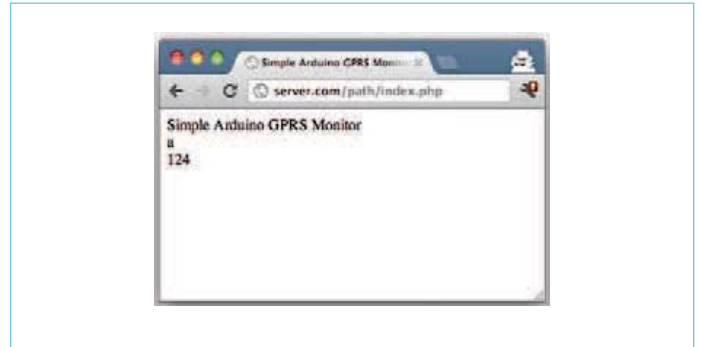


Bild 5. Vom GSM-Shield stammende Informationen im Browser.

Zuerst muss man den Code dieser „PHP-Memory-Cell“ [5] entpacken und die Datei dann in das Wurzelverzeichnis seines Http-Servers hochladen. Die Datei benötigt Lese/Schreib-Berechtigung, damit sie sich selbst modifizieren kann. Anschließend kann man die URL des Servers eingeben (beispielsweise: <http://server.com/path/index.php>) und mit einem Browser beobachten, wie die Information alle 10 s erneuert wird (siehe **Bild 5**).

Schlusswort

Hoffentlich hat diese kurze Einführung in die M2M-Kommunikation mit GSM/GPRS Ihr Interesse geweckt. Wenn man nicht gerade datenhungrige Anwendungen hat, dann kann man sehr viel per Mobilfunk erledigen. Solange die Hardware eingeschaltet ist, kann man auf diese Weise prima aus der Ferne Daten sammeln und Geräte steuern.

(120506)

Weblinks

- [1] **Memo der Firma Ericsson zu den 50 Milliarden Geräten im Netz:**
<http://www.ericsson.com/res/docs/whitepapers/wp-50-billions.pdf>
- [2] **Der AT-Befehlssatz:** <http://de.wikipedia.org/wiki/AT-Befehlssatz>
- [3] **Beschreibung der AT-Befehle von Quectel:**
<http://datasphere.eu/en?t=/documentManager/sfdoc.file.supply&fileID=1285079825955>
- [4] **Arduino-Webseite mit Informationen zum GSM-Shield:**
<http://labs.arduino.cc/GPRS/Index>
- [5] **Archiv-Datei mit allen Beispielen und Bildern:**
www.elektor.de/120506

Danksagung

An dieser Stelle möchte ich mich herzlich beim Team des Physical Internet Laboratory von Telefonica für ihre Unterstützung bei der Entwicklung dieses neuen Arduino-Shields bedanken. Mein spezieller Dank gebührt F. J. Zorzano für das nächtelange Debuggen von Code.

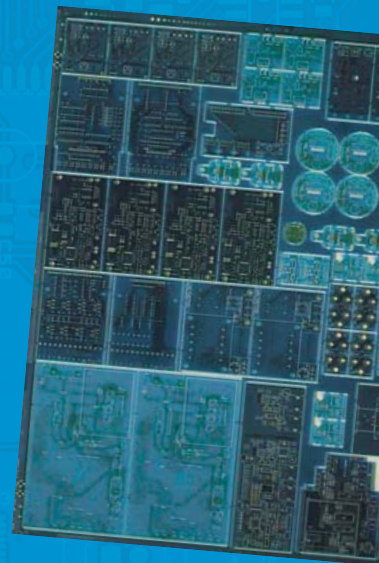
Die europäische Referenz für PCB Prototypen und Kleinserien

Kostensenkung durch leicht zu bedienende Online-Services

- Keine Einrichtungskosten
- Keine Mindestbestellwerte - ab der 1. Platine
- Sofortbestellung Online - ohne Vorkasse

Schnellere Produktion Ihres Layouts durch neue Mehrwertdienste

- PCB Visualizer - sofortiger Online Daten-Check
- Online Tipps für DFM (Design-For-Manufacturability)



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

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

Preise inkl. 19% MwSt und ohne Transportkosten

STANDARD pool – die größte Auswahl an Pooling Optionen in Europa

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

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

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

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

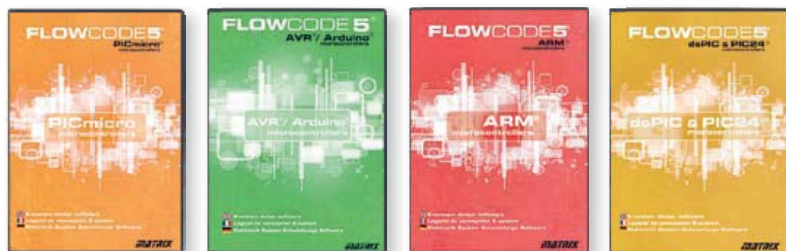
- Leiterplatten mit einlagig isoliertem Metallsubstrat
- ab 3 AT

On demand – Alle Optionen im Nicht-Pooling für Spezialanwendungen

- 1-16 Lagen bis 90µm-Technologie
- ab 2 AT

Entwickeln und Lernen

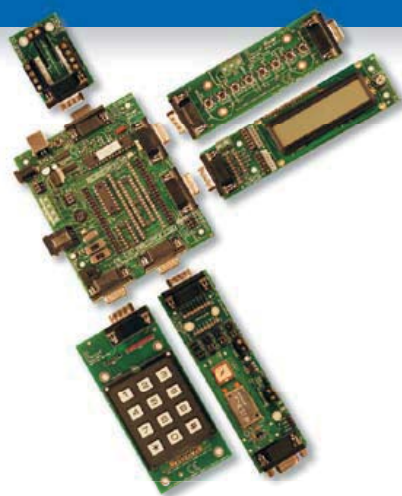
FLOWCODE5



Flowcode 5 ist eine der weltweit besten grafischen Programmiersprachen für Mikrocontroller (PIC, AVR, ARM und dsPIC/PIC24).

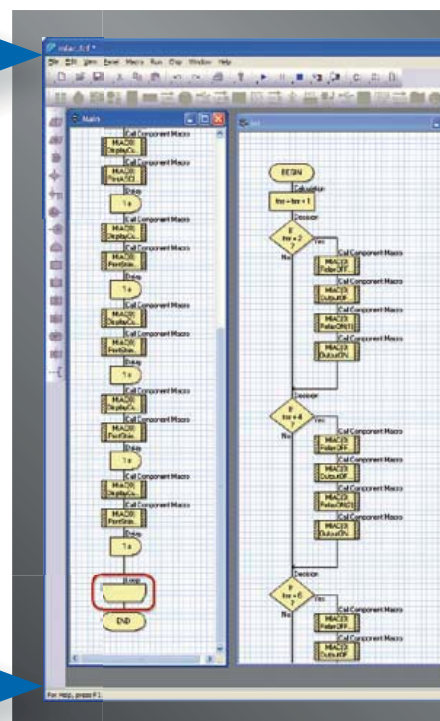
Der große Vorteil von Flowcode ist, dass man mit nur wenig (oder gar keiner) Programmiererfahrung in der Lage ist, komplexe elektronische Systeme in Minutenschnelle zu erstellen.

... für Elektronik



E-blocks sind kleine Schaltungen auf Platinen, die für sogenannte Embedded Systeme typische Elektronik enthalten. Es gibt mittlerweile mehr als 40 unterschiedliche Platinen. Die Module reichen von einfachen LED-Boards zu komplexeren Einheiten wie Programmieren, Bluetooth oder TCP/IP.

E-blocks können einfach zusammengesteckt werden, um damit eine große Bandbreite an Systemen zu Lernzwecken oder für die Ausbildung im Fach Elektronik zu realisieren. Außerdem ist Rapid Prototyping komplexer elektronischer Systeme möglich. Das Angebot wird ergänzt durch Sensoren, Software, Anwendungsinfos und Curricula.



... für Industrie-Steuerungen



Ein MIAC (**M**atrix **I**ndustrial **A**utomotive **C**ontroller) ist eine Steuerungseinheit für den industriellen Bereich, der die Steuerung einer breiten Palette von elektronischen Systemen im Bereich Sensorik, Überwachung und Automotive erlaubt. Intern arbeitet ein MIAC mit leistungsfähigen Mikrocontrollern der PIC18-Serie und verfügt über USB. Das Modul kann mit Flowcode, C oder Assembler programmiert werden. Flowcode ist zudem mit dem Industriestandard CAN-Bus ausgestattet, über welchen mehrere MIACs vernetzt werden können. Flowcode gehört zum Lieferumfang aller verfügbaren MIAC-Bundles.

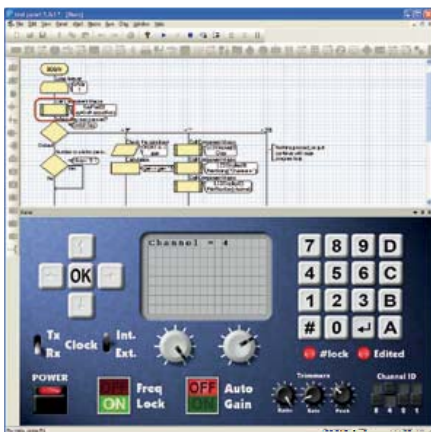
FlowKit

Das FlowKit-Modul ermöglicht In-Circuit-Debugging für Flowcode-Anwendungen in PIC- und AVR-Projekten:

- Start, Stopp, Pause und Schritt für Flowcode-Programme in Echtzeit
- Anzeige der Programm-Variablen
- Ändern von Variablenwerten
- In-Circuit-Debugging für Formula Flowcode Buggy, ECIO- und MIAC-Projekte



mit Flowcode 5 ...



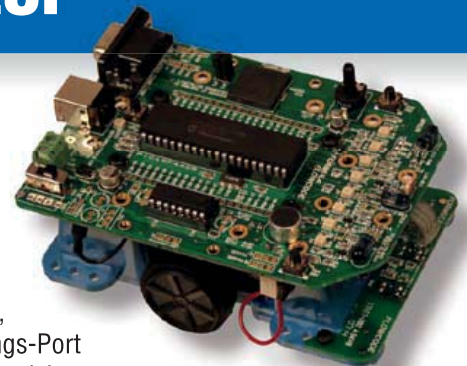
NEU in Flowcode 5:

- Neue C-Code-Ansicht und -Anpassung
- Verbesserte Simulation
- Funktion zum Suchen und Ersetzen
- Neue Variablen-Typen und Funktionen, Konstanten und Port-Variablen
- Automatische Projekt-Dokumentation
- Neuer Projekt-Explorer vereinfacht die Code-Erstellung
- Implementierung von Code-Bookmarks zur Programm-Navigation
- Zugriff auf mehr Chip-Funktionen durch komplettes Redesign des Interrupt-Systems
- Compilierungsfehler und Warnungen navigieren zu Icons
- Icon-Deaktivier-Funktion
- Verbesserte Annotationen
- Verbesserte Links zu Support-Medien

... für Roboter

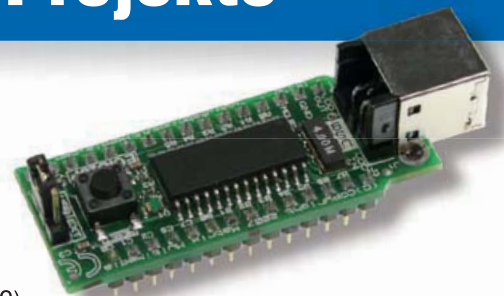
Beim Formula Flowcode Buggy handelt es sich um ein preiswertes Roboter-Fahrzeug für Lernzwecke und zum Einsatz in der Aus- und Weiterbildung. Entsprechend programmiert kann man damit auch auf Robotik-Events glänzen.

Das Vehikel lässt sich direkt via USB programmieren und ist mit Linien-Sensoren, Distanz-Sensoren, 8 LEDs, Mikrofon, Lautsprecher und einem E-blocks-Erweiterungs-Port ausgestattet. Die Lösung eignet sich für einen weiten Bereich an Robotik-Experimenten von der einfachen Linienverfolgung bis zum Entkommen aus einem Labyrinth. Via Erweiterungsport kann man Displays, Bluetooth- und Zigbee-Funk oder gar GPS anschließen.



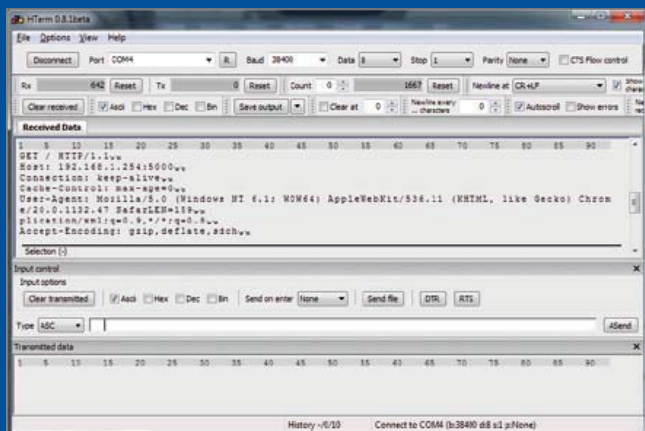
... für USB-Projekte

ECIO-Module enthalten leistungsfähige via USB programmierbare Mikrocontroller im Format von DIL-ICs mit 28 oder 40 Pins (0,6"). Technisch basieren sie auf Mikrocontrollern der PIC18- oder ARM7-Serien. ECIO-Module eignen sich perfekt für eigene Projekte wie auch für den Unterricht, da sich damit komplette Lösungen realisieren lassen. ECIO-Module können mit Flowcode, C oder Assembler programmiert werden. Neue USB-Routinen in Flowcode bieten sich zum extrem schnellen Prototypenaufbau für USB-Projekte an und unterstützen USB-HID, USB-Slave und USB-Serial-Bus (nur PIC). Eigene Projekte können durch integrierte ECIO-Module um USB-Programmierbarkeit ergänzt werden.



Weitere Produkte und Infos zu E-blocks finden Sie unter
www.elektor.de/eblocks

Plötzlicher Reset

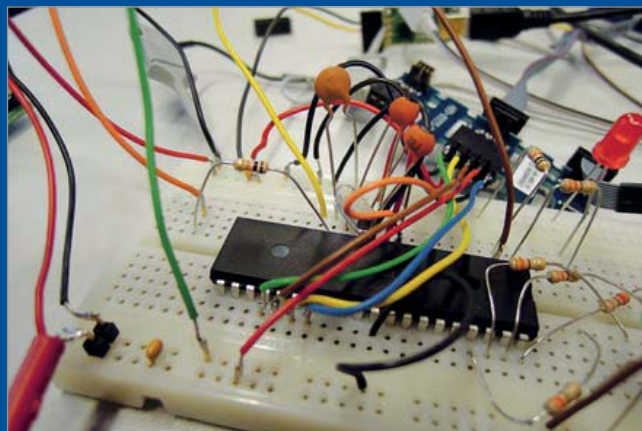


Von Koen Beckers (Elektor-Labor)

Vielleicht kennen Sie dieses Problem nur zu gut: Während der Kommunikation zwischen PC und AVR-Mikrocontroller in Ihrer neuen Schaltung gibt es plötzlich ohne Grund einen Reset beim Controller. Mir passierte das, als ich einen WLAN-fähigen LED-Streifen testete.

Nach dem Ausprobieren mehrerer Konfigurationen und etlichen Tests stellte ich fest, dass das Problem dann auftrat, wenn der UART des Controllers ein großes Datenpaket in Empfang nahm. Manchmal klappte alles, aber oft gab es einen Reset. Gelegentlich gab es direkt am Anfang der Datenübertragung einen Crash. Ganz selten war nach einem Reset sogar ein Power-Cycle notwendig.

Nach der Konsultation von Dr. Internet merkte ich, dass ich damit nicht ganz allein auf der Welt war. Und die Lösung schien simpel zu sein: Der UART braucht zusätzlich einen



10-kΩ-Serienwiderstand und einen Blockkondensator von 100 pF, was zusammen einen Tiefpass mit einer Grenzfrequenz von 159 kHz ergibt.

Da bei mir der Empfang gestört war, schleifte ich den Widerstand in die RX-Leitung ein und schaltete den Kondensator von RX nach Masse. Die korrekte Funktion überprüfte ich mit etwas Test-Code, der die empfangenen Daten wieder zurück zum PC schickte (siehe Screenshot). Die Kommunikation klappte hervorragend und ohne Crash des Mikrocontrollers.

Um ganz sicher zu sein, dass nicht mein Test-Code den Fehler beseitigt, entfernte ich den RC-Filter. Nun crashte der Controller sofort nach Beginn der Übertragung auch mit dem Test-Code. Und mit dem erneuten Einbau des Filters klappte wieder alles wunderbar. Von daher kann ich wohl mit Recht behaupten, dass dieser einfache Filter mein Problem gelöst hat. Möglicherweise löst so ein RC-Tiefpass Ihr Problem ebenfalls!

(120386)

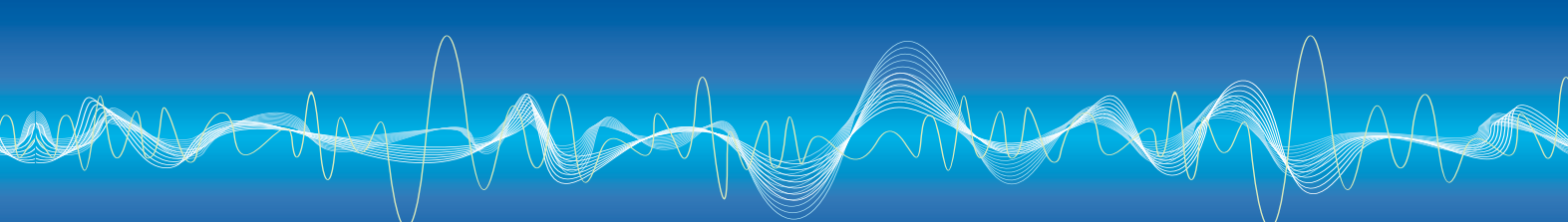
E-Verschmutzung!

Von Thijs Beckers (Elektor-Redaktion & Labor)

Tests und Messungen sind elementar im Labor-Alltag. Eher selten tauchen dabei wirklich hartnäckige Probleme auf. Beim Test eines DCF77-Empfängers für die bekannte AVR-SDR-Artikel-Serie (Teil 1 erschien im März 2012 [1]) verhakte sich Kollege Ton Giesberts in zunächst unerklärlichen Schwierigkeiten.

Es war nämlich so gut wie unmöglich, bei uns im Labor ein Funk-signal zu empfangen. Er probierte verschiedene Ecken innerhalb unseres Schlosses aus – ohne Erfolg. Zugegebenermaßen sind die Schlossmauern ziemlich massiv (ca. 80 cm dick), was Radiosignale durchaus abschwächen kann. Doch auch als die Antenne an einer 2 m langen Stange aus einem Fenster gehalten wurde und sich so definitiv außerhalb der Mauern befand, wurde es nicht besser.





Nun wurden WLAN-Router, PCs, Laptops, Leuchtstofflampen, die kleinen Stecker-Schaltnetzteile und auch DECT-Telefone als Störquellen verdächtig. Ton brachte ein altes Radio mit Langwellenbereich von zuhause mit und versuchte, innerhalb des Schlosses den Sender BBC4 zu empfangen. Damit wollte er die Stelle mit den wenigsten Störungen im Gebäude finden. Die Sendefrequenz von BBC4 beträgt 198 kHz. Und Sie ahnen es schon: Nirgendwo im Schloss war BBC4 zu empfangen. Es gab nur Rauschen und Knistern, E-Smog in Reinform also (wurde nicht vor kurzem ein E-Smog-Tester veröffentlicht? Doch: Genaueres zum TAPIR genannten Messgerät findet sich unter [2]). Ton stellte dann fest, dass er erst dann ein schwaches Signal empfangen konnte, als er mehr als 5 m von den Schlossmauern entfernt war.

Als er wieder mit eingeschaltetem Radio ins Schloss kam, bemerkte er plötzlich einige Interferenzen. Gerade da hatte er eine Leuchtstofflampe passiert, und es zeigte sich, dass diese just bei 198 kHz Störungen aussandte. So testete er die Sache nochmals an einem (sehr) frühen Montagmorgen, als sich noch kaum jemand im Schloss befand und die Lampen aus waren. Doch noch immer kamen zu viele Störungen aus Quellen wie WLAN, DECT, Drucker und PCs sowie anderer Elektronik, die nie ausgeschaltet wird.

So viel zum Testen empfindlicher Elektronik innerhalb des Schlosses. Wenn eine Schaltung die Tests in unserem Labor absolviert hat, dann muss sie ziemlich interferenzfest sein... Kollege Luc Lemmens stieß übrigens bei einer anderen Schaltung, die demnächst veröffentlicht wird, auf ähnliche Probleme. Er testete sie dann an einem anderen Ort: zuhause. Sobald er dort den Prototyp einschaltete, der übrigens auch über einen DCF77-Empfänger verfügte, funktionierte alles auf Anhieb! Ton löste sein Problem, indem er den Prototypen zum Test an einen geschätzten externen Autor schickte. Auch da funktionierte alles sofort!

Die Moral dieser Geschichte ist: Beim Test von Funkempfängern und ähnlichem Equipment sollte man sich nicht ärgern, wenn diese im eigenen Labor nicht so recht wollen, wie sie sollen. Möglicherweise funktionieren die Empfänger ja prima und werden nur vom E-Smog Ihrer anderen Geräte gestört!

(120504)

[1] www.elektor.de/100180

[2] www.elektor.de/120354

Lieblings-Bauteile

Von Thijs Beckers (Elektor-Redaktion & Labor)

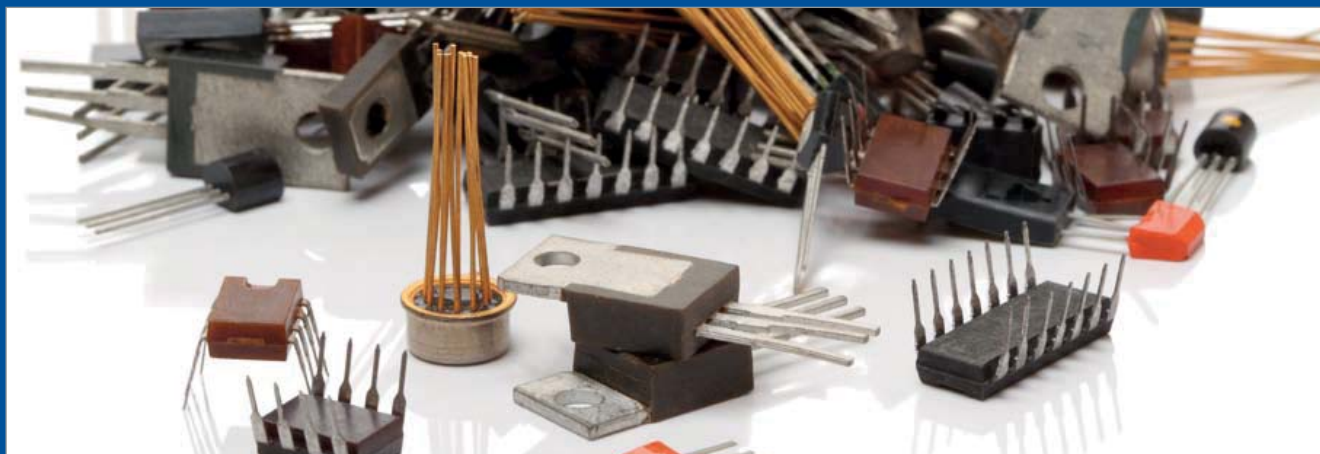
Als Elektroniker hat man es ja mit sehr vielen elektronischen und elektrischen Bauteilen zu tun. Manche sind neu und gerade erst auf den Markt gekommen – manche gibt es schon seit Jahrzehnten.

Mit der Zeit bevorzugt man einen speziellen Spannungsregler, eine Schottky-Diode, einen Mikrocontroller, einen Kondensator-Typ oder gar eine bestimmte Röhre. Und da gibt es durchaus Unterschiede zwischen Elektronikern und somit ganz sicher auch zwischen Elektor-Lesern. Von daher möchten wir Sie bit-

ten, uns zu erläutern, warum Sie gerade diesen einen Transistor immer wieder verwenden. Erzählen Sie uns etwas über Ihren bevorzugten Thyristor und Ihr Lieblings-IC. Gibt es etwas Spezielles am Opamp LMXYZ? Und warum nehmen Sie gerade jenen Controller für praktisch jeden Job?

Wir erwarten eine Flut an E-Mails im Eingangsordner von myfavoritepart@elektor.com und sind sehr gespannt auf ungewöhnliche Geschichten und denkwürdige Ereignisse im Zusammenhang mit einem ganz speziellen Bauteil!

(120503)



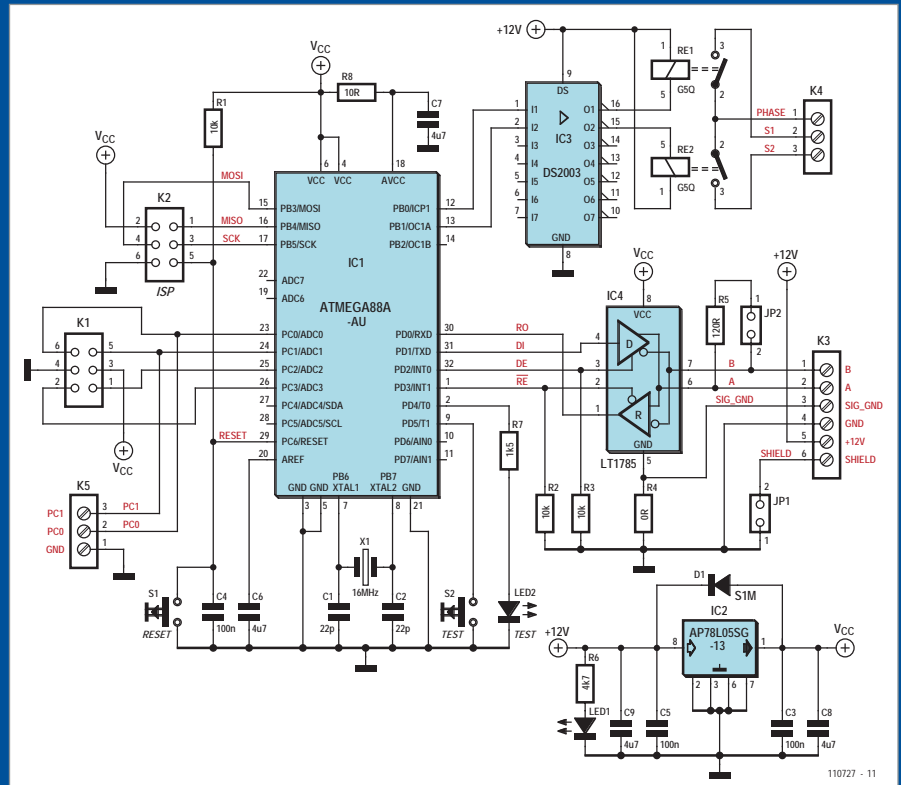
RS485-Massen

Von Jens Nickel

Für alle Automatisierungs-Fans im Allgemeinen und ElektorBus-Freunde im Besonderen gibt es gute Nachrichten: Demnächst kommt weitere Bus-Hardware, unter anderem ein größeres Xmega-Board mit Display und Webserver-Modul, das man als zentrale Steuerung einsetzen kann. Auch die C-Firmware-Bibliothek ist auf dem Weg. Mehr unter www.elektor-projects.com! Zwischendurch folgt hier ein kleines ElektorBus-„Labcenter“, bei dem man wieder etwas lernen kann. Beim Entwurf unserer RS485-Relais-Platine [1] hatten wir einen Extra-Zugang zum Masse-Anschluss des RS485-Treibers vorgesehen (SIG_GND = Signal Ground). Die Idee war, diese Extra-RS485-Treiber-Masse auf einer zusätzlichen Leitung von Knoten zu Knoten weiterzugeben, parallel zur ElektorBus-Masse zur Versorgung der Knoten (GND). Im Artikel hatten wir das erläutert:

Falls Lasten geschaltet werden, treten lokale Störungen auf der Bus-Masse auf. Und falls die Massepotentiale der Treiber zu sehr differieren, können im ungünstigsten Fall Bits aus dem Nichts entstehen. Eine Lösung könnte sein, die Signal-Masse nur an einer Stelle des Busses mit der Bus-Masse zu verbinden. Bei den anderen Busteilnehmern werden beide Massen dann beispielsweise durch einen 100-Ohm-Widerstand getrennt. Bisher haben wir das allerdings nicht getestet...

Hierzu erhielten wir eine Mail von Elektor-Leser Andreas Schoenberg, der sich seit etwa 25 Jahren beruflich (unter anderem) mit RS485-Bussystemen beschäftigt. Unser Wunsch, eine „Referenz-Masse“ haben zu wollen, die für alle Bustreiber immer auf dem gleichen Pegel liegt, sei verständlich. Doch darf beim LT1785 die Common-Mode-Spannung an A/B im Bereich von -7 bis +12 V (gegen Masse) liegen. Selbst wenn die Masse-Potentiale zweier Busknoten um 7 V differieren sollten, würden alle Buspegel noch korrekt interpretiert. Andreas Schoenberg weiter: „Statisch kann so etwas in der Schaltung nicht geschehen. Wenn auf der GND-Leitung zu irgendeinem Hochstrom-Busknoten 7 V abfallen, dann sicherlich genauso viel auf der Versorgungsleitung, d.h. insgesamt gingen dann 14 V verloren.“ Doch das könne ja gar nicht sein, da die Bus-Speisespannung nur 12 V beträgt. Und weiter: „Dynamisch sieht es sicherlich anders aus - wenn im Modul ein Relais anzieht, kann das zu einem kurzfristigen Spannungseinbruch führen. Falls dem so ist, sollte man diesem aber lokal mit entsprechenden Kapazitäten an den Versorgungspins von IC3 entgegenwirken. Ausreichende lokale Pufferung vorausgesetzt, bin ich daher der Meinung, dass in Ihrem System und bei den verwendeten Transceivern kein



Ground-Shift-Problem bestehen kann, das die Kommunikation stören könnte.“

Falls man aber tatsächlich einen Busknoten hätte, der so viel Strom zieht und/oder über so lange/dünne Leitungen angebunden ist, dass GND wegen eines Spannungsabfalls um beispielsweise 3 V über SIG_GND liegt, könnten andere Probleme auftreten. Der LT1785 sieht an seinem VCC-Pin ja immer eine Spannung, die um 5 V über dem lokalen GND-Potential liegt - er wird in diesem Fall also mit 8 V versorgt. Der DE-Pin wird aber über R3 auf GND gezogen, und die CPU kann ihn auch nicht weiter nach unten ziehen, da auch sie nur dieses Massepotential kennt. Für den Transceiver sind dies jedoch 3 V über dem eigenen Masseanschluss (SIG_GND), also immer High. Der Driver-Enable-Pin ist dann gar nicht mehr per Software beeinflussbar!

So weit die Theorie. Interessant wäre es nun zu wissen, welche Erfahrungen unsere Leser mit der Verkabelung der ElektorBus-Hardware und eventuellen Störungen gemacht haben. Wir selbst haben ja schon festgestellt, dass die Kommunikation sehr robust ist, auch mit langen, preiswerten Kabeln und vier Leitungen [2]. Welche Kabel haben Sie benutzt? Lassen Sie uns das wissen unter redaktion@elektor.de!

(120501)

[1] www.elektor.de/110727

[2] www.elektor.de/120198

Von den Machern von Elektor!

elektor

4 | 2013 [D] 17,50 € [A] 20,00 € [L] 20,00 € [B] 20,00 € CHF 30,00

Special Project LEDs 4

Leuchtdioden in Theorie und Praxis

Projekt

**Flash Box – RGB-Controller
für LED-Stripes**

Praxis

**Kühlkonzept zur Aktivluftkühlung
eines Hochleistungs-UV-LED-Moduls**

Theorie & Anwendung

**Optische Charakterisierung
von Leuchtdioden**

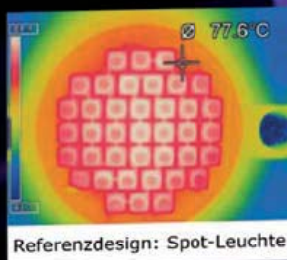
Schaltungstechnik

LED-Treiber für Scheinwerfer

News

LEDs, LED-Treiber und EVA-Boards

**Jetzt neu
am Kiosk!**



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

Bauelemente-Tipp

Von Raymond Vermeulen (Elektor-Labor)

Stellen Sie sich vor, dass Sie weit entfernt einen aktiven Sensor installiert haben, der nur selten in Aktion treten muss. Das Versorgen mit Energie über die Entfernung ist nicht praktisch, und auch eine Batterie vor Ort ist keine gute Lösung. Hier kommt das Energy Harvesting zum Zug, die Energiegewinnung aus der Umwelt! Neuzeitliche Sensoren gehen sparsam mit der Energie um, sie lassen sich wie viele moderne Mikrocontroller in einen stromsparenden Schlaf-Modus versetzen. Wenn der Energiebedarf im zeitlichen Mittel gering ist, kann eine Quelle einen Kondensator während eines langen Zeitintervalls aufladen. Die gesammelte Energie deckt den nur sporadisch auftretenden höheren Energiebedarf.

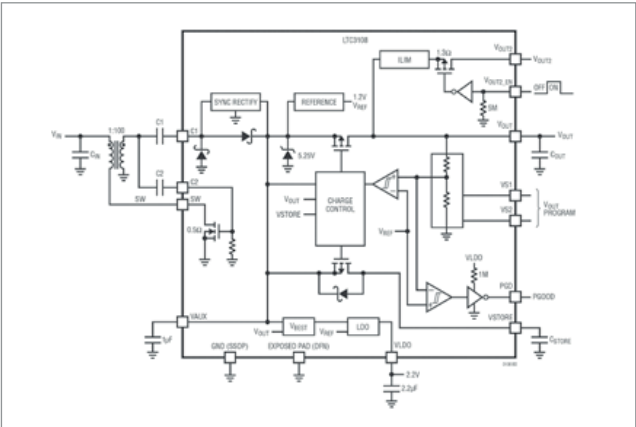
Hier stellen wir zwei Chips vor, die Energie aus der Umwelt gewinnen. Der LTC3108 wurde für den Einsatz vornehmlich mit Peltier-Elementen geschaffen, der LTC3588-1 ist eher für die Energiegewinnung zusammen mit Piezo-Elementen geeignet. Die erste oder zweite Variante könnte beispielsweise den Blitzsensor aus der letzten Folge einschließlich Mikrocontroller und HF-Empfänger mit Energie versorgen.

LTC3108

Dieser Chip kann Energie aus Quellen sammeln, die sehr niedrige Spannungen abgeben. Da die untere Grenze bei 20 mV liegt (!), genügen ein Peltier-Element und eine nur geringe Temperaturdifferenz. Im Datenblatt schlägt Hersteller Linear Technology weitere Energiequellen vor. Der LTC3108 arbeitet nach dem Prinzip des Aufwärts-Spannungswandlers, die Ausgangsspannung ist einstellbar.

Ferner befindet sich ein unabhängiger Low-drop-Spannungsregler (LDO) für 2,2 V auf dem Chip. Auch dieser Ausgang ist steuerbar, beispielsweise um in Zeitintervallen einen Verbraucher einzuschalten, der keinen Schlaf-Modus bietet. Erwähnt sei noch, dass der BQ25504 von TI eine Alternative für den LTC3108 sein kann.

Datenblatt LTC3108
<http://cds.linear.com/docs/Datasheet/3108fb.pdf>



Alle Elektor-Artikel der "80er-Jahre" auf DVD

NEU!



Jetzt unter www.elektor.de/80-89 bestellen!

Strahlungsmesser aufgebohrt

Eigene Modifikationen erweitern die Möglichkeiten

Von Burkhard Kainka (D)

Der Strahlungsmesser aus Elektor 11/2011 ist bei vielen Lesern im Einsatz. Das Gerät hat sich vor allem bei Langzeitmessungen und bei der Untersuchung schwach radioaktiver Proben bewährt. Es gibt jedoch immer noch etwas zu verbessern und zu verändern. Auch individuelle Lösungen lassen sich realisieren, wie wir an einigen Beispielen zeigen.



Im November 2011 haben wir eine neue Version des Elektor Strahlungsmessers vorgestellt [3]. Das Messgerät ist als Bausatz erhältlich [2]; der Mikrocontroller ATmega88 ist bereits mit einem Bootloader ausgestattet. Damit ist es einfach, eigene Programme in den Controller zu laden. Spezielle Aufgaben lassen sich oft mit kleinen Änderungen des Basisprogramms lösen. Hier sollen einige Beispiele gezeigt werden. Aber auch mit dem Lötkolben lässt sich das eine oder andere sinnvoll verändern. Diese Hardware-Modifikationen wollen wir hier zuerst vorstellen.

Kalibrierung

Diese Frage kam in unserem Leserforum öfter auf: Kann man den Strahlungsmesser in Mikrosievert pro Stunde ($\mu\text{Sv/h}$) eichen? Das ist gar nicht so einfach, denn es hängt von vielen Parametern ab. Der wichtigste ist die verwendete Abschirmung. Die Erfahrungen mit dem Strahlungsmesser haben gezeigt, dass es gar nicht so einfach ist, den Sensor absolut lichtdicht zu verpacken und gleichzeitig die zu untersuchenden Strahlen so wenig wie möglich abzuschirmen. Es gab dabei zwei Strategien: Zum einen kann man den Sensor zusammen mit der Probe in eine Blechdose stecken,

die gleichzeitig mit Masse verbunden sein muss. Zum anderen hat es sich bewährt, die Sensorplatine komplett in Alufolie einzupacken, die ebenfalls geerdet sein muss. Einige Leser hatten bei dieser Methode Probleme mit einem Mikrofonie-Effekt, weil die Fotodiode zusammen mit der Abschirmfolie eine Art Kondensatormikrofon bildet. Als Abhilfe kann man die Folie direkt auf den Sensor kleben. Auch selbstklebende, 0,035 mm dünne Kupferfolie hat sich bewährt, vor allem weil die Masseverbindung einfach angelötet werden kann. Mit einer BPW34 und Kupfer-Abschirmfolie wurde eine Messung durchgeführt. Die Strahlenquelle war ein alter Thorium-Lampen-Glühstrumpf in seiner Papierhülle. Wegen der großen Fläche dieser Quelle ergibt sich im Nahbereich ein annähernd homogenes Strahlungsfeld, d.h. die Messergebnisse sind nur wenig vom Abstand abhängig. Der Elektor Strahlungsmesser zeigte im Abstand von 1 cm etwa 40 Impulse pro Minute. Zum Vergleich wurde ein professioneller Geigerzähler mit einem Zählrohr für Gamma- und Betastrahlen ebenfalls im Abstand 1 cm eingesetzt. Dieser zeigte 800 Impulse pro Minute bzw. $20 \mu\text{Sv/h}$. Damit steht das (nicht-amtliche) Ergebnis fest:

$$1 \text{ Imp/min} = 0,5 \mu\text{Sv/h}$$

Prinzipiell ist das Gerät für die Messung der allgemeinen Strahlenbelastung geeignet, wenn man sich genügend Zeit lässt. Bei Langzeitmessungen sollte sogar ein deutlicher Unterschied bei steigender Sonnenaktivität messbar sein. Allerdings müssen die Messzeiträume dabei wesentlich länger sein als mit einem Geigerzähler, was die Sache manchmal etwas unpraktisch macht. Wer die Strahlenbelastung eines Salats misst, erhält im günstigen Fall irgendwann das Ergebnis, dass dieser unbedenklich ist. Aber dann ist der Salat schon verwelkt und schmeckt nicht mehr so gut.

Die Stärke des Geräts liegt nicht so sehr in der Messung der Umweltbelastung, sondern eher in der Untersuchung kleiner, schwach radioaktiver Proben. Und da ist es sogar einem Geigerzähler überlegen, weil man mit dem kleinen Sensor einen deutlicheren Unterschied zur Hintergrundstrahlung sieht.

AC-Kopplung

Gegen die Lichtempfindlichkeit gibt es auch noch ein elektronisches Mittel. Man muss einfach nur eine AC-Kopplung zwi-

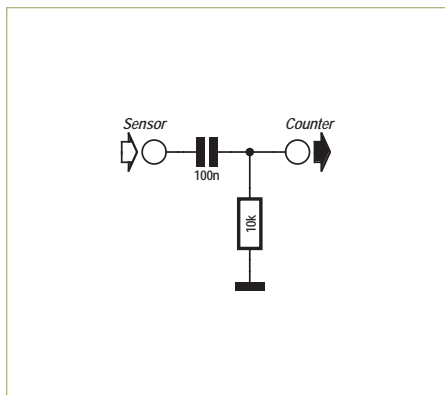


Bild 1. AC-Kopplung.

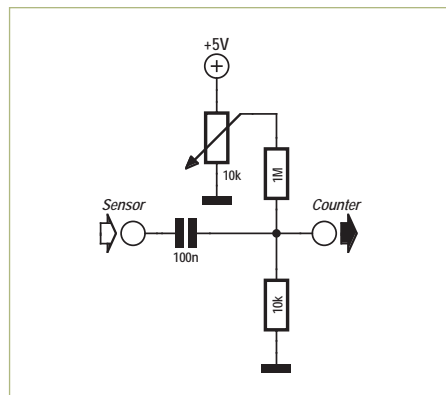


Bild 2. Einstellen der Auslöseschwelle.

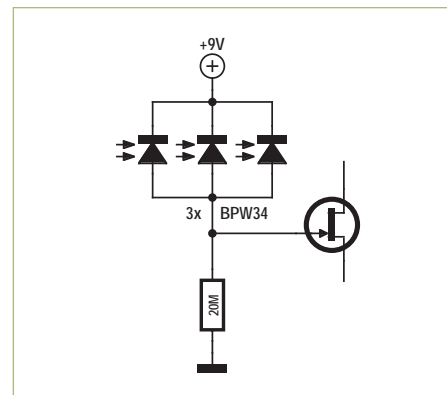


Bild 3. Drei parallele BPW34.

schen Sensorplatine und Zähler realisieren. Dazu reicht ein Kondensator von 100 nF und ein Widerstand von 10 k Ω (**Bild 1**). Die Schaltung wirkt als Hochpassfilter mit einer Grenzfrequenz von 160 Hz. Falls etwas gleichmäßiges Licht eindringt und den Arbeitspunkt des Sensorverstärkers allmählich verschiebt, ist das nun nicht mehr so schädlich. Auch das leichte Flackern einer Glühlampe mit seiner typischen Frequenz von 100 Hz wird etwas gedämpft. Dagegen ist gegen eine Leuchtstoffröhre kein Kraut gewachsen, weil sie scharfe Lichtflanken erzeugt. Insgesamt ist die Schaltung mit diesem Filter jedoch gutmütiger und deutlich weniger lichtempfindlich.

Bei der Gelegenheit kann man gleich noch ein Poti einbauen, mit dem die Auslöseschwelle beeinflusst werden kann (**Bild 2**). Diese Möglichkeit hat man zwar bereits mit der zugehörigen PC-Software, einige Leser verzichten aber auf die PC-Verbindung und wollen auch im freien Feld unterschiedli-

che Messungen durchführen. Wer gerade in einem Steinbruch herumkraxelt, hat ja nicht gern seinen teuren Laptop dabei. Die vorgeschlagene Schaltung setzt den mittleren Pegel um bis zu 50 mV herauf.

Per Default wurde der Software die Auslöseschwelle 10 mitgegeben, was ca. 50 mV entspricht. Der Controller misst beim Start die mittlere Spannung an seinem Eingang und addiert dazu diese Schwelle. Alle Impulse, die größer sind, werden gezählt. Mit der neuen AC-Kopplung würde der Controller als mittleren Pegel die Hälfte des Eingangsräuschens ermitteln und z.B. auf den Wert 5 kommen. Die Auslöseschwelle liegt dann bei 15. Mit dem neu hinzugekommenen Poti kann der Ruhepegel um bis zu 50 mV angehoben werden. Beim Start sollte das Poti am unteren Anschlag stehen. Später tastet man sich dann vorsichtig an die Rauschgrenze heran und findet so den Punkt größter Empfindlichkeit. Ein Druck auf den Null-Taster startet eine neue Messung mit dieser Einstellung.

ren, damit man die Hintergrundstrahlung einschätzen kann. **Bild 4** zeigt eine Leerlaufmessung mit drei parallelen BPW34. Die Auslöseschwelle war 5. Es wurden 40 Impulse unterschiedlicher Energie in einer Stunde registriert. Diese Werte lassen sich dann von den folgenden Messungen subtrahieren, wenn es um schwach radioaktive Proben geht.

Schwache Beta-Aktivität zeigt z.B. Kaliumchlorid, genauer das darin enthaltene Kalium-40. Noch leichter zu bekommen ist Pottasche (Kaliumkarbonat), die als Backmittel im Lebensmittelhandel (z.B. im Reformhaus) zu finden ist. Das Material wird in eine kleine Plastiktüte getan und neben den Sensor gelegt. **Bild 5** zeigt eine Messung über eine Stunde. Insgesamt wurden 251 Impulse registriert, also etwa sechsfach mehr als die Untergrundstrahlung.

Übrigens, Alkali-Batterien laufen ja angeblich niemals aus. Wenn aber doch, dann tritt konzentrierte Kalilauge aus. Diese bildet

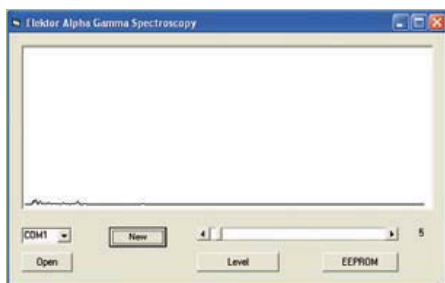


Bild 4. Leerlaufmessung mit drei parallelen BPW34.

Sensoren und Messergebnisse

Die Grundbestückung des Sensors ist eine einzelne PIN-Diode BPW34. Man kann aber sinnvoll mehrere parallel geschaltete BPW34 einsetzen (siehe **Bild 3**). Damit erhält man brauchbare Messwerte schon nach kürzerer Zeit. Man kann wahlweise nur die Impulse zählen oder sich über das Auswerteprogramm das Energiespektrum anzeigen lassen.

Vor der eigentlichen Messung sollt man jeweils eine Leerlaufmessung durchfüh-

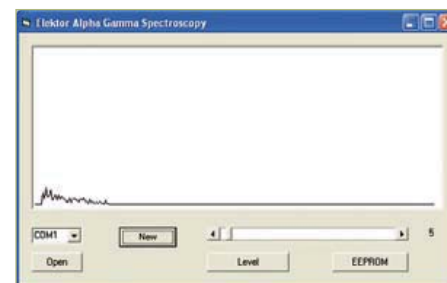


Bild 5. Betastrahlung von Pottasche.

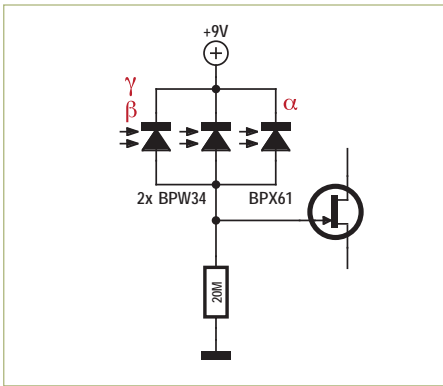


Bild 6. Gemischte Bestückung.

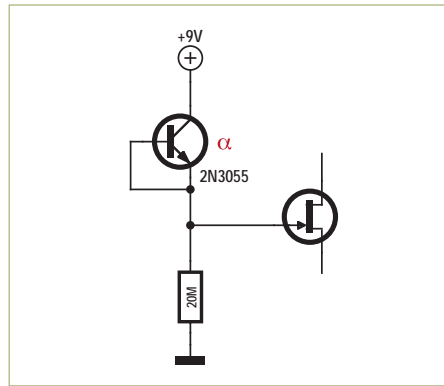


Bild 7. Leistungstransistor als Alpha-Sensor.

zusammen mit dem CO_2 aus der Luft Kaliumkarbonat. Man findet weiße Rückstände, die ebenfalls als schwache Teststrahler taugen. Eine Messung mit dem Elektor Strahlungsmesser hat das bestätigt. So ausgerüstet und mit höchster Empfindlichkeit, auch für Impulse geringer Energie, kann der Strahlungsmesser auch für Proben mit geringer Aktivität eingesetzt werden. Versuchen Sie es einmal mit ganz normaler Holzkohlenasche vom Grill. Die darin enthaltene Pottasche strahlt messbar. Es wurden ca. 150 Impulse in einer Stunde gefunden. Und sogar Zigarettenasche konnte als radioaktiv entlarvt werden. Es zeigten sich 80 Impulse in einer Stunde, also die doppelte Nullrate. Die Ursache liegt vermutlich hauptsächlich im enthaltenen Kalium. Aber auch Polonium als Folgeprodukt zerfallenen Urans könnte eine Rolle spielen, da es sich in der Tabakpflanze anreichert.

Alpha-Messung

Mit einer geöffneten BPX61 lassen sich auch Alpha-Teilchen messen. Ebenfalls sinnvoll ist die Kombination beider Diodentypen. So

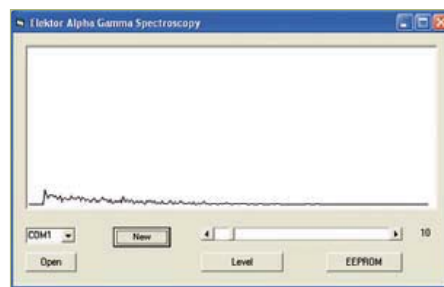


Bild 8. Messung mit einem 2N3055.

wurden z.B. zwei BPW34 für Gamma- und Betamessung mit einer BPX61 für Alphastrahlen kombiniert (Bild 6). Zur Überprüfung sollte man zuerst die schwache Betastrahlung von Kaliumsalz oder Pottasche messen und die Auslöschschwelle entsprechend einstellen. Wenn diese Messung gelingt, können andere schwierige Untersuchungen durchgeführt werden. Dazu gehören diverse Baumaterialien, deren Aktivität von Interesse ist. Alpha-

Strahlen verraten sich durch ihre höhere Energie. So findet man z.B. oft merkliche Strahlung an Gipskarton, weil der darin verwendete Gips aus der Rauchgasentschwefelung von Kohlekraftwerken stammt. Und die Kohle wiederum kann eine kleine Menge Uran und seine Zerfallsprodukte enthalten.

2N3055 als Alpha-Sensor

Wenn Alpha-Teilchen auf eine Silizium-Sperrschicht treffen, dann erzeugen sie ein relativ starkes Signal. Das gilt nicht nur für Fotodioden, sondern genauso für jede andere Diode und für jeden Transistor. Nur kommen die Alpha-Strahlen meist gar nicht erst bis in die Sperrschicht, weil sie durch das Gehäuse oder irgendeine Art von Schutzschicht abgefangen werden. Eine Ausnahme bilden manche Leistungstransistoren im TO3-Gehäuse. Entfernt man die Metallkappe, dann liegt der Silizium-Chip mit seiner relativ großen Fläche völlig frei. Bei einem 2N3055 ist das meist der Fall. Allerdings gibt es einige, die noch eine zusätzliche Schutzschicht besitzen. Man muss also etwas Glück haben. Bei einem neueren Transistor von ST hat es funktioniert (siehe Bild 7). Der Chip liegt völlig blank und ist ein exzellenter Alpha-Sensor. Gamma- und Betastrahlen liefern dagegen keine erkennbaren Signale. Bild 8 zeigt eine Messung an Uran-Pechblende. Mit einem AM241-Strahler aus einem Rauchmelder entstehen so viele Impulse, dass der Zähler nicht mehr alle erfassen kann.

Serielle Schnittstelle

Die Zählerplatine besitzt einen dreipoligen Anschluss, an denen man Zugang zur seriellen Schnittstelle des Controllers hat. Die Signale RXD und TXD liegen hier als TTL-Pegel an. Der Anschluss ist zum Elektor Breakout-Board (BOB) [1] mit dem FT232R kompatibel, so dass man einen PC über USB anschließen kann (virtuelle serielle Schnittstelle). Mit einer sehr einfachen Schaltung kann aber auch ein Gerät mit RS232-Interface angeschlossen werden. Es reichen zwei NPN-Transistoren, um die Pegel zu invertieren (siehe Bild 9). Mit dieser Pegelanpassung kann der Strahlungsmesser an einer normalen COM-Schnittstelle oder an einem üblichen USB/Seriell-Adapter betrieben werden.

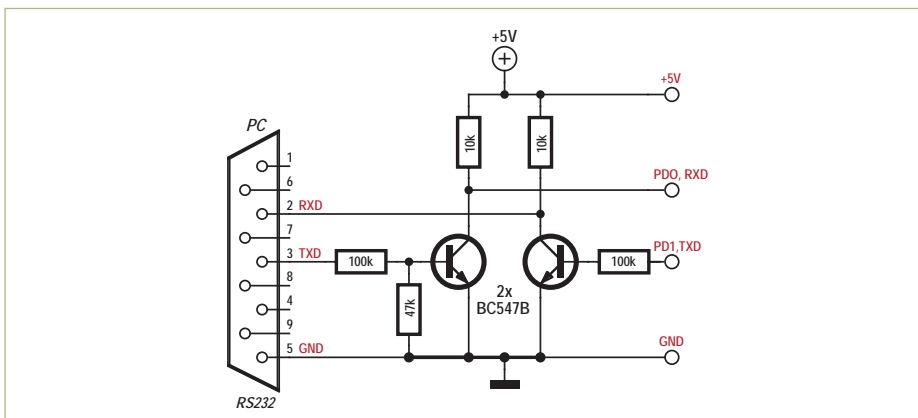


Bild 9. RS232-Pegelanpassung.

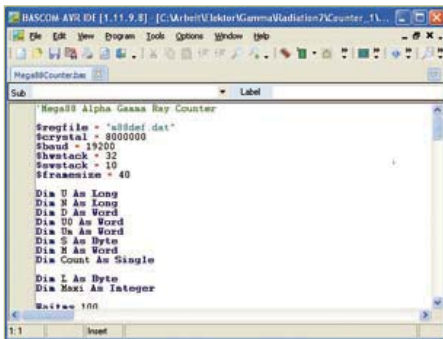


Bild 10. Die Bascom-Entwicklungsumgebung.

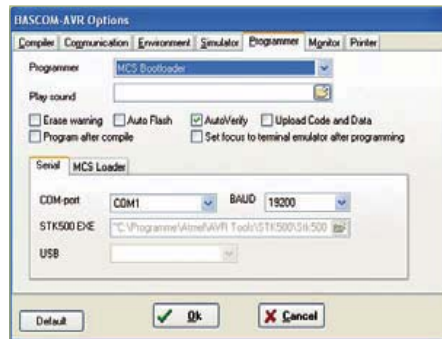


Bild 11. Auswahl des Programmers.

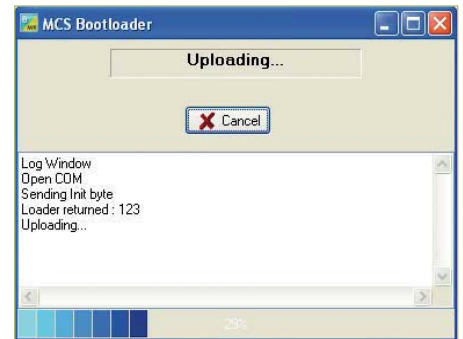


Bild 12. Erfolgreicher Brennvorgang.

Das zugehörige PC-Programm lässt sich einsetzen, um Spektren zu messen und die Schaltschwelle zu ändern. Und es gibt noch eine weitere nützliche Anwendung: Mit der seriellen Schnittstelle können Sie ein Programm-Update einspielen bzw. eigene Programme in den Controller brennen.

Programm-Upload

Vielleicht möchten Sie die Funktionalität des Zählers etwas abändern. Mit dem Bootloader ist das ganz einfach; und es funktioniert auch mit der Demoversion von Bascom (Bild 10). Laden Sie zuerst den Basic-Quelltext und compilieren Sie ihn nach eventuellen kleinen Änderungen.

Wählen Sie dann in BASCOM den passenden Programmer aus. Es ist in diesem Fall der MCS-Bootloader. Zusätzlich müssen die COM-Schnittstelle und die Baudrate 19200 stimmen (Bild 11).

Starten Sie nun den Programmer (= Bootloader), z.B. mit dem kleinen grünen Platinen-Symbol. Der MCS-Bootloader versucht nun, die Verbindung zum Chip aufzunehmen.

Die Verbindung funktioniert kurz nach einem Reset oder Neustart des Controllers. Schalten Sie deshalb das Gerät einmal aus und wieder ein. Damit startet der Programmiervorgang (Bild 12).

Nach dem Ende des Upload-Vorgangs startet der Controller ohne Verzögerung mit dem neuen Programm.

Software-Varianten

Auch ohne tiefere Basic-Kenntnisse kann jeder kleine Änderungen am Programm vor-

nehmen und die veränderte Firmware dann in den Controller schreiben. Dazu einige kleine Beispiele:

1. Konstante Auslöseschwelle (Listing 1)

Man möchte immer mit einer konstanten Auslöseschwelle von 5 starten. Dazu muss nur die Zeile „L = 5“ an passender Stelle in das Programm eingefügt werden. Außerdem soll die Startmeldung „Counter“ entfallen. Die entsprechende Zeile wird dann einfach mit einem Semikolon auskommentiert. Eine weitere Änderung ist im Listing 1 vorbereitet, aber dort noch auskommentiert. Mit der Zeile Um = 0 an der passenden Stelle wird der mittlere Level auf Null gesetzt, was

Listing 1: Änderungen in Counter_1

```
'Print "Counter"
Locate 1 , 1
'Lcd "Counter"
'Waitms 500
Readeeprom L , 1
If L = 255 Then L = 5
U = 0
For N = 1 To 1000
    D = Getadc(0)
    U = U + D
Next N
U = U / 1000
Um = U
'Um = 0
U0 = Um + L
N = 0
```

für die in Bild 1 vorgeschlagene AC-Kopplung besser geeignet ist.

2. Energieverteilung messen (Listing 2)

Ein zweites Beispiel soll Impulse in Abhängigkeit von ihrer Energie zählen. Es gibt drei Zähler, für alle Ereignisse (N), für solche mit Leveln über 40 (N2) und für die höchsten Impulse mit Leveln über 80 (N3). Alle drei Zählergebnisse werden in der oberen Zeile dargestellt. Damit bekommt man eine grobe Messung der Energieverteilung ohne einen angeschlossenen PC. Diese Möglichkeit ist ja schließlich der wesentliche Vorteil der Halbleitersensoren gegenüber einem Geigerzähler.

Listing 2: Änderungen in Counter_2

```
Do
    D = Getadc(0)
    If D > Maxi Then Maxi = D
Loop Until D < U0
Maxi = Maxi - Um
If Maxi > 255 Then Maxi = 255
N = N + 1
If Maxi > 40 Then N2 = N2 + 1
If Maxi > 80 Then N3 = N3 + 1

Locate 1 , 1
Lcd N
Lcd " "
Locate 1 , 8
Lcd N2
Locate 1 , 14
Lcd N3
```

Weitere Messobjekte

Immer wieder kommt die Frage auf, was man denn messen könnte, um die Funktion des Geräts zu überprüfen. Schon der Startartikel hat ja auf Uran-Mineralien, Leuchtzeiger und Kaliumchlorid hingewiesen. Inzwischen sind noch einige weitere Quellen als mögliche Kandidaten aufgefallen:

- Uranglas wird immer noch hergestellt und z.B. für Glasperlen verwendet, die stark fluoreszieren.
- Schutzgas-Schweißelektroden vom Typ WT20 bestehen aus Wolfram mit einem 2-prozentigen Zusatz von Thorium-Oxid. Die Elektroden sind schwache Quellen für Alpha-, Beta- und Gammastrahlen.
- Große Senderöhren verwenden in ihren direkt geheizten Kathoden ebenfalls Wolfram mit einem Zusatz von Thorium. Senderöhren liegen nicht gerade an jeder Straßenecke herum, aber das gleiche Material wurde auch in einem Magnetron aus einem Mikrowellenherd entdeckt. Die ausgebaute Magnetron-Kathode ist ein wirksamer Teststrahler.
- Thoriumsalze wurden früher auch in Leuchtstrümpfen für Gas- und Petroleumlampen verwendet. Neuere Glühstrümpfe verwenden alternative Materialien. Wer aber noch einen alten Glühstrumpf findet, hat damit meist einen äußerst kräftigen Teststrahler, der auch durchdringende Gammastrahlen erzeugt.
- Ionisations-Rauchmelder enthalten einen starken Alpha-Strahler mit 33 kBq Am241. Solche Rauchmelder sind in USA und Großbritannien gebräuchlich, im übrigen Europa jedoch kaum noch zu finden. Das Material sendet nur relativ schwache Gammastrahlen aus, sodass eine BPW34 nur ca. 30 Impulse pro Minute findet. Eine geöffnete BPX61 ist dagegen ein wirksamer Alpha-Detektor und liefert viele tausend Impulse pro Minute.

Jetzt sind Sie dran! Wie Sie sehen, können Sie auch mit kleinen Modifikationen die

Möglichkeiten des Strahlungsmessers noch beträchtlich erweitern. Der Bausatz [3] ist recht einfach aufzubauen, da nur bedrahtete Komponenten verwendet werden. Für eine einfache Verbindung mit dem PC ist ein Elektor BOB [1][2] empfehlenswert.

(120381)

Weblinks

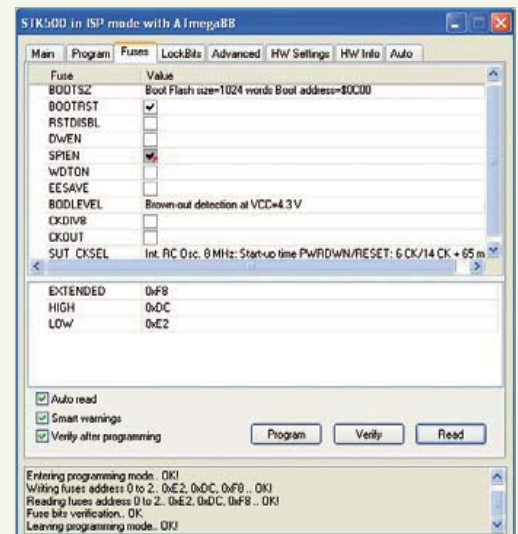
- [1] www.elektor.de/BOB
- [2] www.elektor.de/120381
- [3] www.elektor.de/110538

Bootloader selbst programmieren

Anwender, die den Controller selbst gebrannt haben, haben dies meist noch ohne den Bootloader getan. Es bietet sich an, den Bootloader nachträglich aufzuspielen. Das Bootloader-Programm ist unter den Beispielen des BASCOM-Entwicklungssystems zu finden und muss nur noch an den verwendeten Controller ATmega88 angepasst werden. Die angepasste Form befindet sich unter den Download-Dateien zu diesem Artikel [2]. Das compilierte File BootLoader88.hex muss nun mit einem ISP-Programmer wie z.B. dem STK500 in den Controller gebrannt werden. Bei dieser Gelegenheit wird übrigens das vorhandene Counter-Programm erst mal überschrieben. Sie sollten es am Ende mit dem Bootloader wieder aufspielen, nicht mit einem Hardware-Programmer, um den Bootloader nicht durch die neue ISP-Programmierung wieder zu löschen. Zuvor müssen aber die Fuses passend eingestellt werden (Bild). Wichtig ist die Einstellung und Aktivierung des Bootbereichs mit einer Größe von 1024 Words. Wenn alles korrekt programmiert wurde, kann nun das Counter-Programm direkt über die serielle Schnittstelle geflasht werden.

Zum genaueren Verständnis der Vorgänge sind die folgenden Hinweise nützlich: Wenn der Bootloader aus der Entwicklungsumgebung heraus angesprochen wird, sendet der PC eine endlose Serie von Bytes 123 und wartet darauf, dass der angeschlossene Controller mit demselben Byte antwortet. Danach werden alle nötigen Programmdateien über die serielle Schnittstelle gesendet und so in den Flash-Speicher übertragen.

Auf der Seite des Mikrocontrollers sieht die Sache so aus: Nach einem Neustart oder Reset verzweigt der Controller erst einmal in den Bootbereich, weil dies in den Fuses so festgelegt wurde. Das dort vorhandene Bootloader-Programm muss feststellen, ob gerade ein Bootvorgang geplant ist. Wenn nicht, verzweigt es an die Adresse Null und startet damit die vorhandene Firmware. Wenn es aber ein Byte 123 erkennt, sendet es dieses zurück und empfängt dann weitere Daten, um sie in den Programmspeicher zu übertragen. Nach beendeter Programmierung wird das neue Programm ab Adresse Null gestartet.



NEU

LabVIEW

Band 1: Einstieg in die Praxis

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

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

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

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

240 Seiten (kart.) · Format 17 x 23,5 cm · ISBN 978-3-89576-253-6

€ 34,80 · CHF 43,20



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

ARM-Mikrocontroller

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

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

Die Quelltexte zu den Beispielprogrammen stehen gratis unter www.elektor.de/arm-buch zum Download bereit. Das zum Buch gehörige Hardware-Starterkit kann unter www.elektor.de/arm-kit geordert werden.

**NEU!**

261 Seiten (kart.)

Format 17 x 23,5 cm

ISBN 978-3-89576-262-8

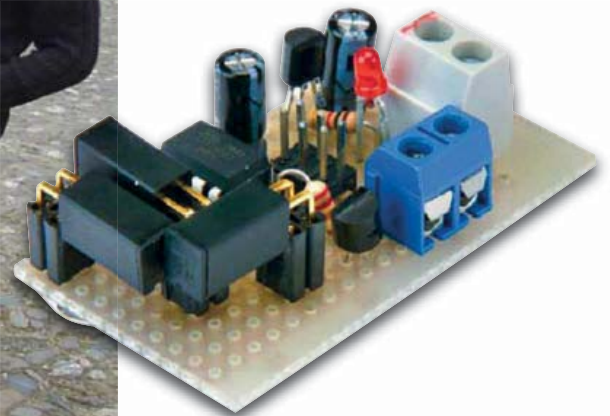
€ 39,80 · CHF 49,40

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

Motorrad-Alarm

Elektronischer Wächter

Von Pierre Rigai (F)



Das Prinzip dieser Alarm-Anlage ist ganz einfach: Ist das Motorrad einmal auf dem Ständer abgestellt, geben drei sinnvoll angeordnete Quecksilberschalter eine Lage-Information an einen Mikrocontroller, welche dieser beim Einschalten abspeichert. Damit wird der Alarm scharf geschaltet; jede Lageänderung des Motorrades verändert den Status dieser drei Schalter und der Alarm wird ausgelöst.

Meine Schaltung passt in ein ganz kleines Gehäuse, kaum so groß wie ein 1-Euro-Stück. Ich habe bereits zwei Räder mit dieser Alarmanlage ausgerüstet, eine Kawasaki 650 und eine Aprilia SX50. Elektor hat sie seinerseits bei einer Honda Goldwing GL1200 eingebaut, welche man auf den Fotos sieht. Aus Sorge um die Umwelt hat das Elektor-Labor es vorgezogen, meine Quecksilberschalter durch Kugelschalter zu ersetzen, weniger umweltschädlich und preiswert.

Beim Einschalten zeigt ein sehr kurzes Tonsignal den Beginn einer Wartezeit von 30 s an,

während derer das Motorrad noch bewegt werden kann. Nach Ablauf dieser Zeit wird die Position des Motorrades gespeichert. Dies wird durch ein zweites sehr kurzes Tonsignal angezeigt. Sobald der Mikrocontroller eine Änderung der Position nach einer weiteren Wartezeit von 30 s feststellt (zufällige Berührung), wird der Alarm für 30 s ausgelöst. Falls das Motorrad nach diesen 30 s nicht in der vorherigen Position ist, geht der Alarm weiter.

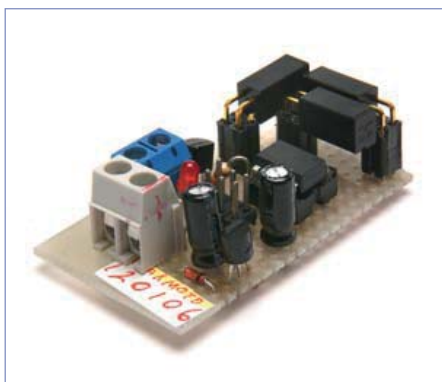
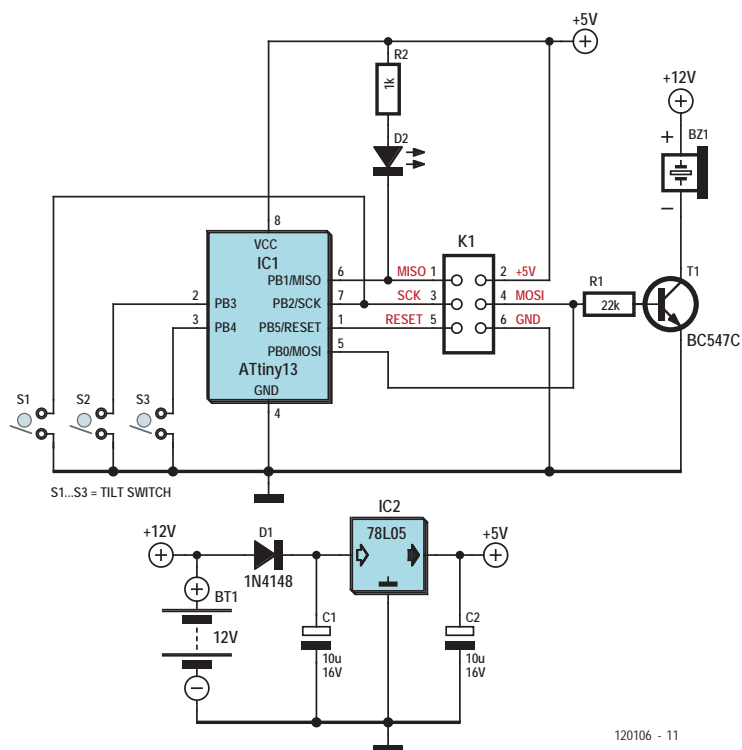
Der μC meines Prototyps ist ein ATtiny13 im DIP-Gehäuse, er wurde auf einer Experimentierplatine montiert. Um ihn zu program-

mieren, habe ich einen kleinen Verbinder von TE Connectivity (Nr. 7-215079-6) eingesetzt, der noch kleiner als ein HE10-Steckverbinder ist. Mit identischen Verbindern habe ich meine AVR-ISP-Programmierkabel ausgerüstet. Falls man auf eine weitere Miniaturisierung Wert legt, kann man den Controller erst programmieren und danach einlöten, was noch einmal Platz spart.

Wie die Fotos zeigen, sind zwei der drei Schalter mit abgewinkelten Kontakten (S1 bis S3) entgegengesetzt nebeneinander (und leicht in gegensätzliche Richtungen geneigt) montiert, um eine zuverlässige

Bewegungserkennung beim Aufrichten der Maschine vom Seitenständer aus zu erhalten. Der dritte Schalter wird längs dazu (im rechten Winkel zu den beiden anderen) angeordnet; dieser reagiert eher auf Stöße, wie sie typischerweise beim Einziehen eines Mittelständers auftreten. Die drei Schalter werden an die Ports PB2, PB3 und PB4 des μC angeschlossen (interne Pull-Ups des Attiny13 müssen aktiviert werden).

Die Schalter sind auf Sockeln montiert, um sie während des Programmierens des μC entfernen zu können. Hierdurch werden mögliche Konflikte zwischen internen und externen logischen Pegeln vermieden. In jedem Falle muss der Schalter an PB2 während des Programmierens offen sein, weil sonst das Signal SCK auf 0 gehalten und das



Programmieren unmöglich wird. Falls gewünscht, kann an Port PB1 eine LED (D2) angeschlossen werden, welche blinkt, solange die Anlage scharf ist. Ausgang PB0 steuert über T1 die „Sirene“. Dies ist kein passiver Piezo-Resonator, sondern ein aktiver Baustein mit eingebautem Oszillator, gespeist mit 12 V, welcher ein schrilles Pfeifen abgibt. Beim Pfeifen zieht er etwa 150 mA. Im Ruhezustand liegt der Verbrauch in der Größenordnung von 1,5 mA. In der jetzigen Version hat die Schaltung keine eigene Spannungsversorgung, die Bordspannung des Motorrads muss anliegen, während die Anlage scharf ist.

Für die Alarmanlage muss ein diskreter und schlecht zugänglicher Platz gefunden

Stückliste

Widerstände:

R1 = 22 k
R2 = 1 k

Kondensatoren:

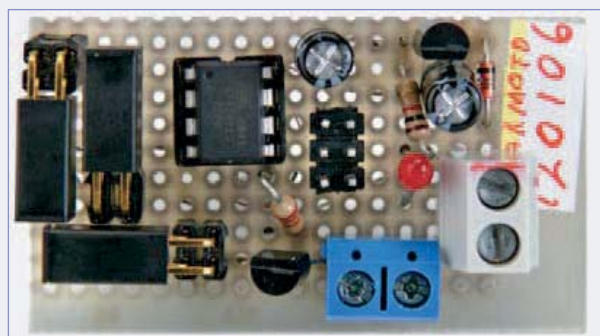
C1, C2 = 10 μV /16 V Elko, radial

Halbleiter:

D1 = 1N4148
T1 = BC547C
IC1 = ATtiny13-20PU (Atmel), DIP-8 (programmiert erhältlich, siehe www.elektor.de/120106)
IC2 = 78L05

Außerdem:

S1, S2, S3 = Kugel-Neigungsschalter mit gewinkelten Kontakten z.B. RBS040200



werden. Ich weiß genau, dass bei manchen Modellen wie der SX50 ein entschlossener Dieb sehr schnell den Sitz angehoben und die Anschlussdrähte gekappt hat! Schließlich muss man noch den Reset-Schalter (zwischen Pin 5 und 6 von K1) gut verstecken. Dieser ermöglicht es dem rechtmäßigen Eigentümer der überwachten Maschine, die Alarmanlage außer Betrieb zu nehmen.

Programm

Das sehr einfache Programm [1] wurde mit dem Tool CodeVisionAVR (herausgegeben von HP Infotech) in C geschrieben und enthält nur eine C-Datei (main.c). Kompiliert belegt es weniger als 50 % des Flash-Memorys des Tiny13. Dieser verwendet den internen 128-kHz-Oszillator, um den Stromverbrauch zu reduzieren. Für das, was er zu tun hat, ist es überflüssig, ihn mit 8 MHz zu takten! Die Beschreibung der Fuse-Bits ist in *main.c* enthalten.

Am Anfang der Datei finden sich die Typ-Deklarationen (*typedef*), einige *defines* und die globalen Variablen, welchen ich immer

ein *G_* voranstelle, um sie von lokalen Variablen zu unterscheiden. Globale Variablen sind die, welche sich das Hintergrundprogramm und die Interrupt-Routine von *timer 0* teilen. Letzterer wurde im CTC-Modus konfiguriert, um alle 100 ms einen Interrupt zu erzeugen. Diese 100-ms-Zeitreferenz dient gleichzeitig zum Herabzählen der „Rüttel-Intervalle“ und der Steuerung des LED-Blinkens.

Die LED blinkt beim Einschalten alle 500 ms für 100 ms und im Ruhezustand alle 5 s. Durch 10 geteilt dient diese Zeitbasis auch zur Festlegung der Intervalle im Sekundentakt.

Die Funktion `interrupt [TIM0_COMPA]` `void timer0_compa_isr(void)` ist die Interrupt-Routine von *timer 0* und erhält bzw. liefert wie jede Interrupt-Funktion keine Parameter. Sie verwendet zwei statische Variablen, die zwischen einem Interrupt und dem nächsten konstant sind. Schließlich gibt es noch die Funktion `void main(void)`. Dies ist natürlich das Haupt-

programm, welches seinerseits auch keine Parameter empfangen oder Werte zurückgeben kann!

Das Hauptprogramm verwendet nur eine Variable *memo_sw*, um den Zustand der Schalter zu speichern. Hier findet sich auch die Initialisierung des Mikrocontrollers, die Interrupt-Freigabe, der Piepston beim Einschalten und schließlich eine Endlosschleife. Dort wird festgestellt, ob Intervalle abgelaufen sind und außerdem der Status der Neigungsschalter fortwährend geprüft. Meine Kommentare [1] erläutern den Ablauf.

(120106)

[1] www.elektor.de/120106

Ausschnitt aus dem Quellcode: komplett downloadbar unter [1]

```

/*-----
I Funktion      : main                      I
-----
I Action        : main program              I
I In Param      : nada                      I
I Return        : nada                      I
I               :                           I
I               :                           I
-----*/
void main(void)
{
    byte memo_sw;                          // tilt status check

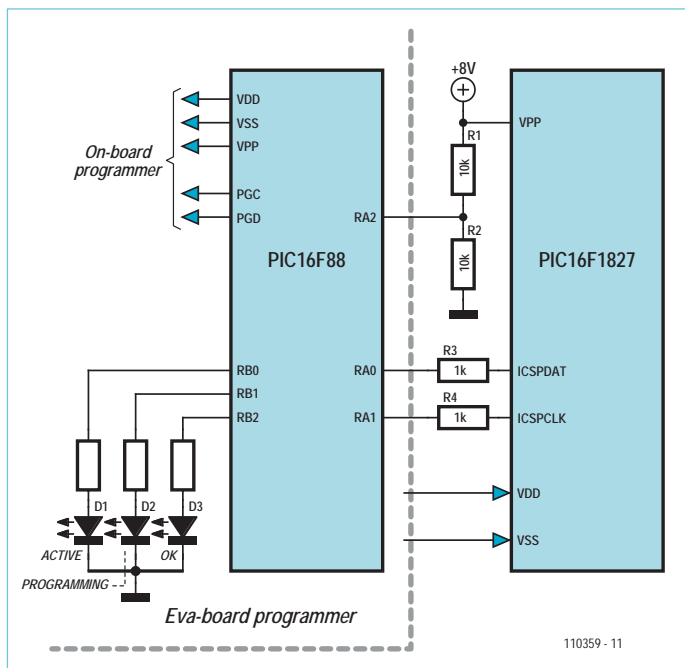
    // Crystal Oscillator division factor: 1
    #pragma optsize-
    CLKPR=0x80;
    CLKPR=0x00;
    #ifdef _OPTIMIZE_SIZE_
    #pragma optsize+

// Input/Output Ports initialization
// Port B initialization
// Func5=In Func4=In Func3=In Func2=In Func1=In Func0=Out
// State5=P State4=P State3=P State2=P State1=P State0=0
PORTB=0x3E;
DDRB=0x01;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 0,500 kHz
// Mode: CTC top=OCR0A
// OC0A output: Disconnected
// OC0B output: Disconnected
TCCR0A=0x02;
TCCR0B=0x04;
TCNT0=0x00;
OCR0A=0x31;
OCR0B=0x00;
#endif
}

```

PIC-Programmer für Notfälle



Gebrauchsanleitung

- Compilieren Sie das Programm, so dass die Hex-Datei für den PIC16F1827 erzeugt wird.
- Konvertieren Sie die Hex-Datei mit „ihex2pic“ [1] in das PIC-Format.
- Öffnen Sie in MPLAB die Datei SimpProg.asm und prüfen Sie, ob die korrekte Include-Datei eingebunden wird.
- Stecken Sie den PIC16F88 in das ausgeschaltete Entwicklungsboard und schließen Sie den PIC16F1827 wie in Bild 1 an.
- Programmieren Sie den PIC16F88 mit dem zum Entwicklungsboard gehörenden Programmer.
- Nach dem Programmieren leuchtet LED D1 auf. Die Programmierspannung 8 V des PIC16F1827 kann eingeschaltet werden.
- LED D2 und D3 leuchten ebenfalls auf. Warten Sie, bis LED D1 und D2 verlöschen. Wenn kein Fehler aufgetreten ist, leuchtet LED D3 weiterhin.
- Schalten Sie die Programmierspannung 8 V ab, setzen Sie das Entwicklungsboard außer Betrieb.

Nun ist der PIC16F1827 programmiert!

(110359)gd

[1] www.elektor.de/110359

Von Elbert Jan van Veldhuizen (NL)

Der Autor hatte folgendes Problem: Die neuen Mikrocontroller der PIC-Familie mit erweitertem Befehlssatz (wie beispielsweise der PIC16F1827) wurden von einem vorhandenen Programmer nicht erkannt, ein Software-Update für den Programmer ließ auf sich warten. Wenn ein PIC-Entwicklungsboard mit zugehörigem Programmer verfügbar ist, lassen sich diese Mikrocontroller trotzdem programmieren. Dazu sind lediglich vier Widerstände, eine externe 8-V-Spannungsquelle sowie ein zweiter PIC nötig. Allerdings muss der zweite PIC über das vorhandene Board programmierbar sein. Im hier berichteten Fall war der zweite Mikrocontroller ein PIC16F88. Für diesen Controller wurde ein kurzes, aus rund 500 Worten bestehendes Programm geschrieben, es kann von der Elektor-Website [1] heruntergeladen werden. Zuerst wurde aus der Hex-Datei, die für den PIC16F1827 bestimmt war, auf dem PC mit einem Konvertierungsprogramm eine Include-Datei für den PIC16F88 erzeugt. Diese Datei (ungefähr 3500 Worte) wurde in den freien Speicherplatz des PIC16F88 geladen und anschließend als Daten ausgegeben.

Die LEDs auf dem Entwicklungsboard zeigen den Status der Programmierung an. Wenn LED D1 aufleuchtet, ist der Controller bereit. Jetzt kann die Programmierspannung 8 V an den PIC16F1827 gelegt werden. Leuchten die LEDs D2 und D3 auf, ist dies das Zeichen dafür, dass die Programmierung läuft. Falls ein Programmierfehler auftritt, verlischt LED D3. LED D1 und D2 verlöschen, sobald die Programmierung abgeschlossen ist. Die Programmierspannung 8 V und das Entwicklungsboard können abgeschaltet werden, der PIC16F1827 ist jetzt programmiert.

Über die Widerstände R1 und R2 erkennt der PIC16F88, dass die Programmierspannung anliegt. Die Widerstände R3 und R4 sind eigentlich entbehrlich, sie haben hier nur schützende Funktion. Falls beide Controller infolge eines Programmfehlers gleichzeitig Daten ausgeben oder ein Controller falsch angeschlossen ist, werden Kurzschlüsse verhindert.

Anzeige

EINKAUFSSPASS AUCH FÜR MÄNNER.

GROSSES PRODUKTSPEKTRUM ONLINE KALKULIERBAR.





RECHNEN SIE MIT BESTEM SERVICE

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

www.leiton.de
Info-Hotline +49 (0)30 701 73 49 0

SDR mit AVR (6)

Decodierungsvarianten beim BBC-Empfang

Von Martin Ossmann (D)

Dass sich die beliebten AVR-Controller auch für die digitale Signalverarbeitung eignen, soll diese Serie zeigen. In dieser abschließenden Folge schauen wir uns ein paar weitere Decodier-Verfahren an. Das Ganze wie immer auch in der Praxis: diesmal gewinnen wir Zeitinformationen aus BBC-Signalen.



In der letzten Folge [5] ging es bereits um die Decodierung von Zeitzeichen-Signalen. Mit den dort beschriebenen digitalen IIR- und Matched-CIC-Filtern wurden dem deutschen DCF77, dem britischen MSF und dem französischen TDF162 die ersten codierten Zeitsignale entlockt. Nun geht es zum Abschluss um die Decodierung von BBC198 und des ehemaligen Senders BBC648. An diesen Beispielen werden noch zusätzliche Decodier-Verfahren und ihre Realisation erläutert.

Bit-Takt-Synchronisation

In der Nachrichtentechnik gibt es eine Reihe von Verfahren, bei welchen Daten synchron übertragen werden. Die Daten laufen dabei ununterbrochen mit einer bestimmten Taktrate durch das System. In einem Empfänger wird dann oft zuerst demoduliert. Im so gewonnenen Signal stecken dann die gesuchten Bits. Für die Experimente mit der Bit-Takt-Synchronisation benötigt man

außer dem einfachen Frontend (oder der bei Elektor erhältlichen Empfängerplatine) sowie der Empfänger-Software „EXP-RX-FM-125kHz-RDSlike-BitSync-V01.c“ den Signalgenerator mit 125-kHz-Serienschwingkreis plus die Sender-Software „EXP-TX-FM-125kHz-RDSlike-V01.c“. Mit letzterem kann man frequenzmodulierte Daten auf 125 kHz mit einer Datenrate von 31,25 bit/s generieren. An den DAC-Ausgängen werden das Eingangssignal und der Bittakt ausgegeben.

Zur Taktrückgewinnung wird das Signal mit einer Frequenz von $31,25 \text{ Hz} \cdot 16 = 500 \text{ Hz}$ abgetastet. Auf ein Bit entfallen also 16 Samples. Man könnte nun einfach versuchen, aus dem Datensignal mit einer PLL den 31,25-Hz-Bit-Takt zu rekonstruieren. Das funktioniert aber so nicht. Den Grund hierfür erkennt man, wenn man sich das Spektrum des Signals in **Bild 1** ansieht. Es ist nämlich gar keine Signalkomponente bei 31,25 Hz enthalten, auf welche die PLL einrasten könnte.

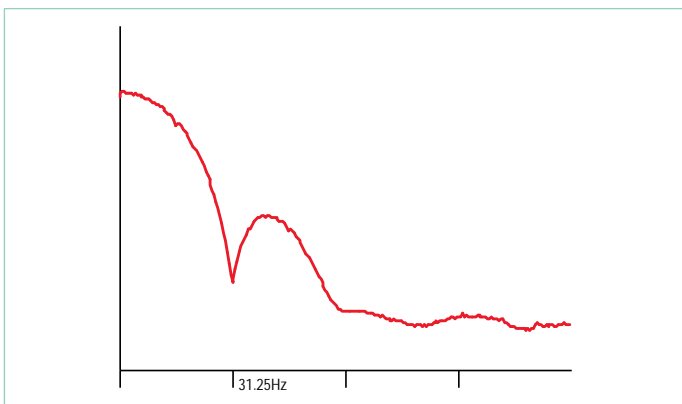


Bild 1. Spektrum des 31,25-Hz-Datensignals.

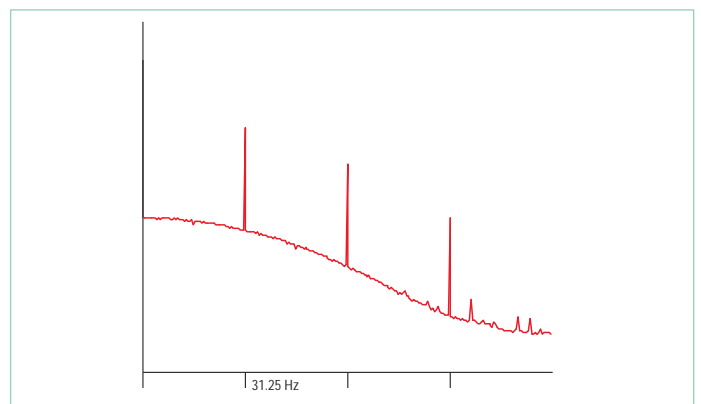


Bild 2. Spektrum des gleichgerichteten Datensignals.

Wenn man das Datensignal allerdings gleichrichtet, hat das Spektrum (wie in **Bild 2** zu sehen) eine Linie bei 31,25 Hz. Das Signal dieser Linie kann eine PLL dann zum Einrasten nutzen. Die Datenrückgewinnungs-PLL sieht dabei so aus wie in der Prinzipschaltung von **Bild 3**.

Diese PLL ähnelt der schon früher aufgebauten PLL zur Trägerrückgewinnung (siehe Bild 7 in der Folge 4 [4] dieser Artikelserie). Um die Bit-Taktfrequenz zu regeln wird wie folgt vorgegangen (siehe Programm „EXP-RX-FM-125kHz-RDSlike-BitSync-V01.c“): Die Routine `doSignalSample()` von **Listing 1** wird mit der 16-fachen Bit-Taktrate aufgerufen. Dabei wird der Wert der Variablen `Clk16` erhöht und bei Erreichen von 16 zurückgesetzt. Die Phase dieses Sägezahnsignals sollte nun so geregelt werden, dass die Mitte (`Clk16 = 8`) auch tatsächlich in der Mitte des gesendeten Bits liegt. Das Bit wird dann von Routine `doBitSample()` ermittelt und weiterverarbeitet. Die Regelung der Phase geschieht folgendermaßen: Im Normalfall zählt `Clk16` mit dem Datensignaltakt periodisch bis 16. Vom Ausgangssignal des Phasenvergleichers wird nun die Periode je nach Vorzeichen auf 15 oder 17 verändert, wodurch sich die Phase anpassen kann.

Um das korrekte Arbeiten der Takt-PLL kontrollieren zu können, wird am Ausgang DAC1 das demodulierte Signal ausgegeben. Am Ausgang DAC2 wird `Clk16` mit Peak bei `Clk16 = 8` (also im Sampling-Zeitpunkt) dargestellt. **Bild 4** zeigt dies beim Empfang von BBC auf 198 kHz (siehe **Bild 5**).

Early-Late-Synchronizer

Neben einer klassischen PLL kann man auch einen sogenannten „Early-Late-Gate-Synchronizer“ zur Taktrückgewinnung einsetzen. Ein Blick auf das zugehörige Augendiagramm in **Bild 6** erleichtert das Verständnis.

Auch hier geht es darum, das `Clk16`-Signal mit dem Bit-Takt so zu synchronisieren, dass das Datensignal in der Bit-Mitte abgetastet wird. Um dies zu erreichen misst man den Signalwert zu einem frühen Zeitpunkt E und zu einem späten Zeitpunkt L, beispielsweise beim dritten (E) und zwölften (L) Takt des Signals `Clk16`. Sind beide Messwerte gleich groß, liegen die Punkte E und L symmetrisch zur Mitte. Ist aber ein Wert größer als der andere, so ist das Auge quasi aus der Mitte herausgerutscht, und man muss die Phase von `Clk16` wieder korrigieren.

Der zugehörige Programmcode ist in **Listing 2** wiedergegeben. Die Differenz zwischen dem Wert bei E (early) und dem Wert bei L (late) wird in `earlyLateDelta` zehnfach aufsummiert. Anschließend wird mit diesem Wert die Periode von `Clk16` korrigiert. Der Code für diesen Synchronizer kann ebenfalls mit dem Programm „EXP-RX-FM-125kHz-RDSlike-BitSync-V01.c“ getestet werden.

Decodierung der Phasenmodulation bei BBC198

Nun kommt der praktische Einsatz der Bit-Synchronisation. Hierzu programmiert man den Empfänger so, dass er die von BBC auf 198 kHz gesendeten phasenmodulierten Daten empfangen kann. Der Aufbau des Signals ist in [7] genau beschrieben. Die Phase des 198-kHz-Trägers wird um $\pm 22,5^\circ$ moduliert. Die Datenrate ist 25 bit/s und es wird eine Manchesterkodierung (gelegentlich auch

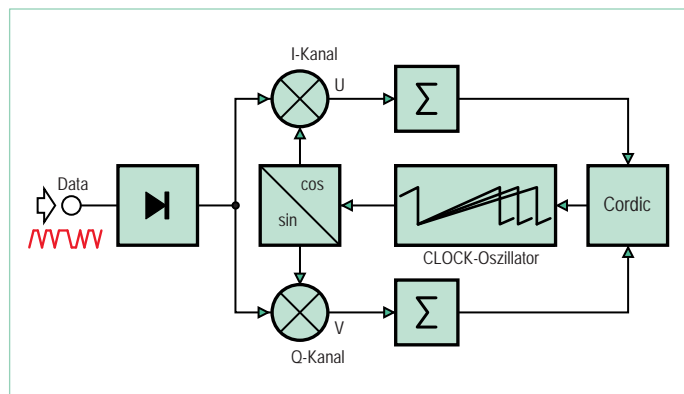


Bild 3. Prinzip der Takt-PLL.

Listing 1: Phasenregelung

```
void advanceCLK16() {
    Clk16++;
    if ( Clk16 >= ClockPeriod ) { Clk16=0 ; }
}

void doSignalSample(){
    int16_t Signal ;
    ATOMIC_BLOCK(ATOMIC_FORCEON) {
        Signal=FRQcicout ;
    }
    ClockPLL16(Signal) ;
    doBitSample(Signal) ;
    advanceCLK16() ;
}
```

Biphasen-Codierung genannt) mit Filterung verwendet. Dadurch wird die mittlere Phase des Trägers nicht geändert, so dass der Träger nach wie vor als hochstabile Referenzfrequenz verwendet werden kann.

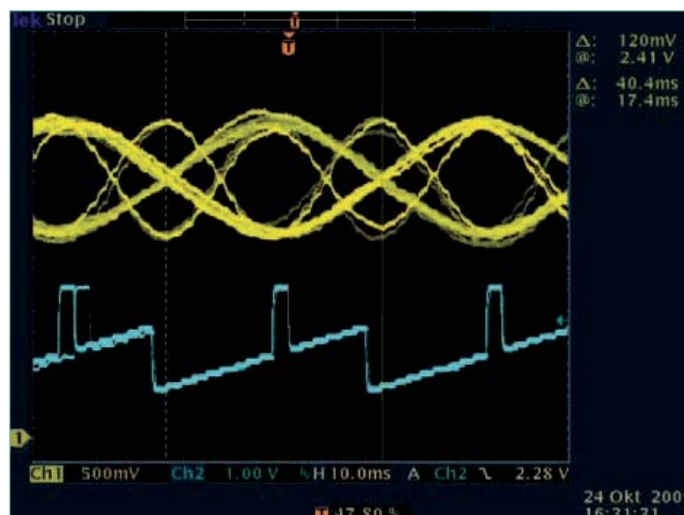


Bild 4. Augendiagramm von BBC198.

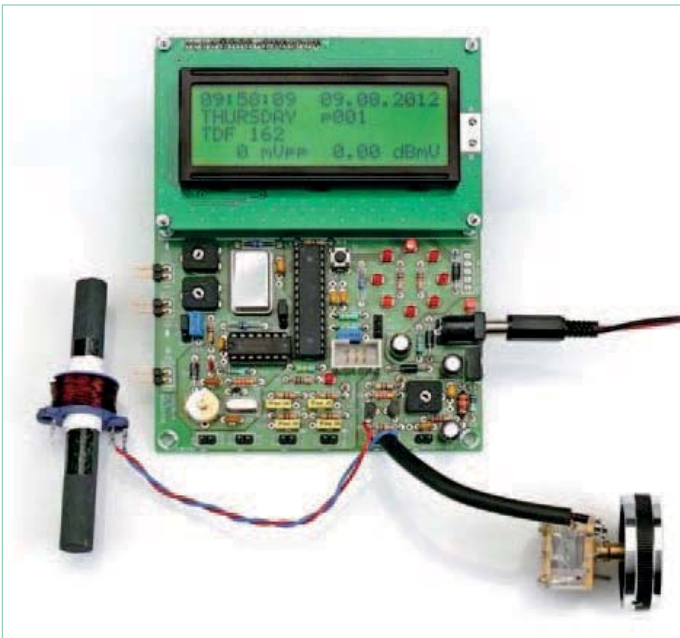


Bild 5. Kompletter Receiver für den Empfang von BBC198.

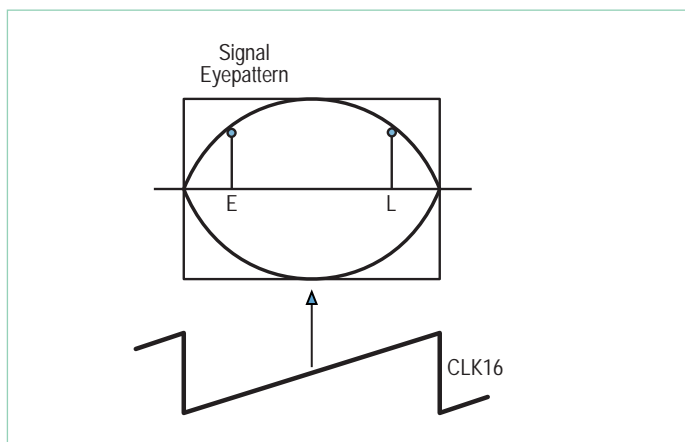


Bild 6. Augendiagramm des Early-Late-Synchronizers.

Listing 2: Early-Late-Synchronizer

```
void earlyLateSynchronizer(int16_t Signal){
    if ( Clk16==3 ) { earlyValue=abs(Signal) ; }
    if ( Clk16==12 ) { lateValue=abs(Signal) ; }
    if ( Clk16==8 ) {
        earlyLateDelta += lateValue-earlyValue;
        ClkCnt++;
        ClockPeriod=16 ;
        if ( ClkCnt==10){
            ClockPeriod += earlyLateDelta/16 ;
            earlyLateDelta=0 ;
            ClkCnt=0 ;
        }
    }
}
```

Die A/D-Wandlung wird mit einer Frequenz von 20 MHz / 2.500=8 kHz vorgenommen. Durch diese Abtastung wird der Träger von 198 kHz heruntergemischt auf 198 kHz – 25 * 8 kHz = -2kHz. Die „Zwischenfrequenz“ beträgt demnach -2 kHz und wird folglich mit vier Samples pro Periode abgetastet, wie das beim IQ-Abtastmischer auch gefordert ist. Das negative Vorzeichen bedeutet dabei, dass man die Seitenbänder gespiegelt bearbeitet. Nach der Aufteilung in I- und Q-Komponente werden beide durch ein erstes CIC-Filter zweiter Ordnung gefiltert und ein Downsampling um den Faktor $M_d = 16$ vorgenommen. Danach schließt sich ein weiterer CIC-Tiefpassfilter ohne Downsampling an. Aus diesen gefilterten I- und Q-Signalen wird nun mit dem CORDIC-Verfahren die momentane Amplitude und Phase ermittelt.

Eine Möglichkeit der Demodulation des Signals würde darin bestehen, eine PLL zur Trägersrückgewinnung einzusetzen und dann das NRZ-Signal zu decodieren. NRZ steht für „Non Return to Zero“, und hierunter versteht man darunter das ursprüngliche Datensignal, mit dem die Phasenmodulation vorgenommen wurde und das man ja als eigentliches Nutzsignal zurückgewinnen möchte. Zur Decodierung kann man jedoch den Aufbau eines PLL-Regelkreises vermeiden, indem man eine Phasenmodulation als Frequenzmodulation interpretiert. Diese Überlegung kann man anhand von **Bild 7** nachvollziehen.

Der theoretische NRZ-Phasenverlauf ist als Signalverlauf $p(t)$ dargestellt. Da die Phase nicht springen kann, sondern um Bandbreite zu sparen stetig langsam variiert wird, sieht der tatsächliche Phasenverlauf eher aus wie das Signal $q(t)$. Die Abtastung des demodulierten Signals erfolgt mit 400 Hz, so dass auf ein Bit wieder 16 Samples entfallen. Nun entspricht nachrichtentechnisch die Frequenz eines Signals der zeitlichen Änderung der Phase. Damit ergibt sich der zu $q(t)$ gehörige Frequenzverlauf als $f(t)$ durch Ableitung von $q(t)$ nach der Zeit. Wenn die Phase linear zunimmt, ergibt sich eine konstante positive Frequenz. Wenn die Phase mit der Zeit linear abnimmt, ergibt sich eine konstante negative Frequenz.

Durch erneute Tiefpassfilterung entsteht aus $f(t)$ das Signal $g(t)$, welches nun leicht auszuwerten ist. Man muss sich nur den Wert in der Mitte des Bits anschauen. Ist er positiv, wurde eine „1“ empfangen. Ist er negativ, dann entspricht dies „0“. Auf diese Weise wird deutlich, wie der Empfänger arbeitet. Durch Differenzbildung von aus dem Cordic-Verfahren kommenden Phasen-Samples ergibt sich die Momentanfrequenz. Diese wird wieder mit einem CIC-Filter tiefpassgefiltert, so dass das Signal $g(t)$ entsteht. Dieses Signal wiederum speist die Bit-Takt-Rückgewinnung. Damit ergeben sich die in Bild 8 von Folge 2 [2] der Artikelserie dargestellten Signalverläufe. In der weiteren Verarbeitung müssen die Datenblöcke erkannt und decodiert werden. Wie das funktioniert, wird später beschrieben. Vorerst zeigt **Listing 3** die auf 198 kHz von der BBC empfangenen Daten.

Block-Synchronisation

Nach der Extraktion der Bit-Folge aus dem Empfangssignal bleibt als letzte Aufgabe nur noch die Erkennung von Anfang und Ende der Datenblöcke. Hierfür ist die Block-Synchronisation zuständig. Die meisten Funkverfahren senden die Daten in Form von Blöcken

bekannter Länge. Schaltet man nun einen Empfänger ein, weiß dieser nicht, wo diese Blöcke beginnen. Um das zu erkennen, gibt es mehrere Verfahren. Gelegentlich wird dazu eine eindeutige Präambel gesendet, die sonst nicht in den Daten vorkommen kann. Dieses Verfahren wird z.B. bei RFIDs des Typs EM4102 eingesetzt. Eine Präambel verbraucht allerdings Platz, der die Nutzlast reduziert. Viele Verfahren basieren daher auf der sowieso schon vorhandenen Information der Fehlerkorrektur.

Fehlererkennung

Viele Datenübertragungsverfahren für Funkübermittlung hängen an einen Datenblock jeweils eine Bit-Folge als Prüfsumme an, mit deren Hilfe der Empfänger Fehler erkennen und ggf. auch korrigieren kann. Wie man zeigen kann, lässt sich mit dieser Zusatzinformation auch die Block-Synchronisation realisieren. Zuerst wird beschrieben, wie die Prüfsumme mit dem 125-kHz-Testsender (Software „EXP-TX-FM-125kHz-RDSlike-V01.c“) berechnet wird. Die Berechnung aus Listing 4 ist in Bild 8 als Prinzipschaltung dargestellt.

Zehn Prüf-Bits werden in einem mit XOR-Gattern rückgekoppelten Schieberegister namens Syndrom berechnet. Die 16 Daten-Bits werden aus einem Schieberegister herausgeschoben, und diese Bits steuern die neuen Bits, die im Syndrom-Schieberegister landen. Nach 16 Takten ist die CRC-Prüfsumme berechnet. Zu dieser Prüfsumme wird noch per XOR der Wert 198H addiert, was die Synchronisation erleichtert. Die 10 Prüf-Bits werden nach den Daten-Bits gesendet. Dieses Verfahren ist vom RDS-System [8] inspiriert. Wenn der Empfänger weiß, wo ein neuer Block beginnt, berechnet er das Syndrom - genau wie der Sender zuvor auch - für die folgenden 16 Bits und überprüft, ob dieses mit den auf die Daten folgenden 10 Bits übereinstimmt. Um den Anfang eines Blocks zu finden, geht man wie folgt vor: Bei der Ankunft eines neuen Bits prüft man, ob die vergangenen 26 Bits eine fehlerfreie Kombination aus 16 Daten- und 10 Prüf-Bits darstellen. Dieser Test müsste daher mit einem Abstand von 26 Bits ein „OK“ ergeben. Genau nach diesem Rhythmus sucht man also. Es passiert ziemlich selten, dass an einer anderen als der richtigen Position einmal ein „OK“ vorkommt. Der Block-Synchronisationsalgorithmus sucht nun einfach nach einem Muster, wo drei „OK“ im Abstand von je 26 Bits vorkommen. In diesem Fall geht er davon aus, dass er die richtige Stelle für den Blockanfang gefunden hat.

In Bild 9 ist schematisch dargestellt, wie die Prüf-Bits fortwährend berechnet werden. Die Syndrom-Berechnung kann auch als eine Art Filterung verstanden werden. Das eigentliche, rückgekoppelte Schieberegister ist das gleiche wie bei der Sendeschaltung. Einlaufende Bits werden verarbeitet, und wenn ein Bit aus dem 10+16-Bit-Schieberegister ausläuft, wird dies ebenfalls im Syndrom berücksichtigt. Die Software hat dann nur zu testen, ob im Syndrom-Schieberegister aktuell der Wert 198H steht, der ja bei der Erzeugung addiert wurde. Immer wenn dieser Wert gefunden wird, meldet die Software „OK“. Nun wird ein Zähler namens localTime immer von null bis 25 inkrementiert. Beim Stand von null ist gerade ein Block samt Prüf-Bits eingelaufen.

Zur Darstellung dieses Verhaltens wurde eine Variante des Empfängers so programmiert, dass mit dem einen PWM-DAC-Ausgang die aktuelle

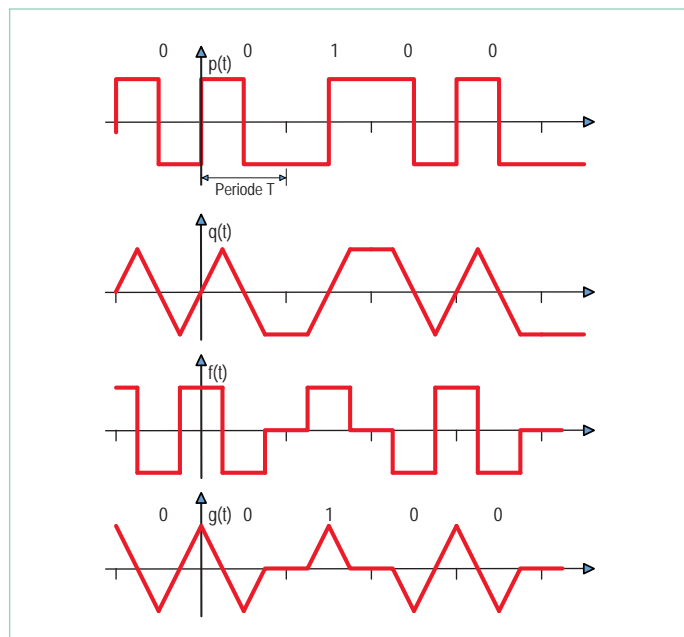


Bild 7. Phasenmodulation als Frequenzmodulation.

Listing 3: BBC auf 198 kHz: empfangene Daten

```
+E EB11 0391
+0 AAAA AAAA
+E E8F1 0391
+E EB09 0391
+E EB49 0391
+0 AAAA AAAA
+E EB31 0391
+E EB41 0391
+0 5475 2980 time= 18:38 week=07 day=02 tuesday
+E EB29 0391
```

Listing 4: CRC-Berechnung im Sender

```
SendBits(w,16) ;
Syndrom=0 ;
for (j=0 ; j<16 ; j++) {
    Syndrom=Syndrom<<1 ; // outgoing bit is at
                          // mask 0x0400

    if (w & 0x8000) {
        Syndrom ^= 0x0400 ; // exor into outgoing bit
    }
    w=w<<1 ;
    if ( Syndrom & 0x0400 ) {
        Syndrom ^= 0x1B9 ;
    }
} ;
Syndrom=(Syndrom^ofs)<<6 ;
SendBits(Syndrom,10) ;
```

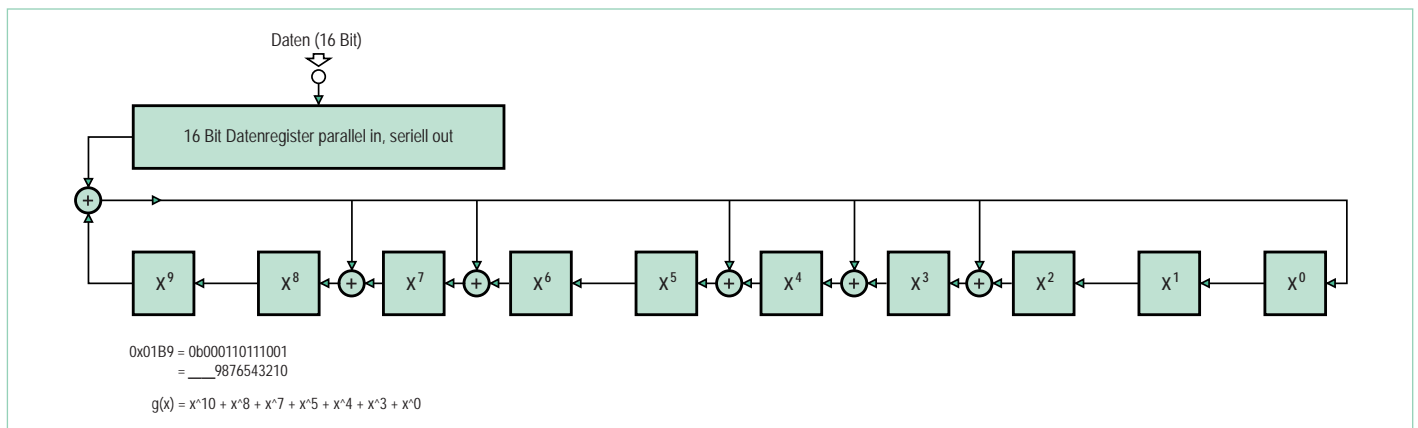



Bild 8. CRC1.

Zeit (localTime) als Sägezahn ausgegeben wird. Der zweite PWM-DAC-Ausgang gibt immer „high“ aus, wenn die Synchronisationsschaltung „OK“ detektiert. In **Bild 10** ist der Rhythmus schön zu sehen. Dargestellt ist auch eine Stelle, an der sich ein „falsches OK“ ereignet. Man muss allerdings ziemlich warten, bis dies einmal auftritt.

Tritt nach der Synchronisation die Entscheidung „OK“ im Zeitpunkt null auf, hat man einen fehlerfreien Block empfangen, dessen Daten dann ausgegeben werden. Damit ist der Datenempfänger komplett. Mit dem zuvor Beschriebenen ist die Synchronisation auf BBC bei 198 kHz umfassend erläutert. Dabei werden allerdings nicht die RDS-Parameter, sondern Blöcke aus 47 Daten-Bits und 13 Prüf-Bits verwendet. Die von BBC auf 198 kHz gesendeten Daten sind aus den Blöcken anhand der Dokumentation [7] relativ leicht zu ermitteln. Insbesondere die gesendete Uhrzeit steckt einfach codiert in einem Block.

Empfang von BBC648 AMSS mit Decodierung

Ursprünglich war als letztes Beispiel der Empfang von BBC auf 648 kHz geplant. Dieser Sender hat aber leider in der Zwischenzeit seine Ausstrahlung eingestellt, weswegen ein eigenes experimentelles Nachvollziehen des Verfahrens nicht mehr möglich ist. Trotzdem soll das Vorgehen kurz beschrieben werden, weil man anhand dieses Beispiels verstehen kann, wie AMSS-Daten gesendet werden. Bis vor kurzem hat die BBC noch auf 648 kHz im AMSS-Verfahren Daten gesendet (siehe die umfangreiche Literatur unter [10]). Unser

Frontend tastet mit 20 MHz / 1.875 = 10,666... kHz ab.

Damit ergibt sich als Zwischenfrequenz 648 kHz - 61 * 10,666... kHz = -2,666... kHz \approx 0,25 * 10,666... kHz.

Als Modulation wird Biphasen-Signalisierung mit Phasenmodulation von $\pm 20^\circ$ verwendet.

Bezüglich Modulation und Bit-Codierung entspricht dieses Verfahren (fast) genau dem des Empfangs auf 198 kHz. Lediglich die Parameter sind etwas verändert. Hier werden Blöcke aus jeweils 36 Nutz-Bits und 11-CRC-Prüf-Bits verwendet. Zwei solche Blöcke bilden jeweils eine Gruppe, und eine gewisse Anzahl von Gruppen zusammen bildet einen DRM-SDC-Block. Um diesen zu verstehen muss man einen Blick in die DRM-Spezifikation werfen. Unser einfacher Empfänger kann lediglich den Sendernamen extrahieren und anzeigen. Aber immerhin haben wir so einen kompletten Datenempfänger realisiert, der korrekt arbeitet.

In **Listing 5** sehen wir die empfangenen Daten. Die Ausgabe „2/7+“ bedeutet, dass die zweite von sieben Gruppen fehlerfrei empfangen wurde. Wenn alle sieben Gruppen fehlerfrei empfangen wurden, werden die Daten decodiert.

Schlussbemerkung

Damit ist die sechstellige Artikelserie über Signalverarbeitung und „Software Defined Radio“ mit AVR-Controllern am Ende angelangt. Es wurde eindrucksvoll demonstriert, dass man mit solch einfachen Mikrocontrollern ganz schön viel anstellen kann. In „richtigen“ SDR-

Listing 5: BBC auf 648 kHz: AMSS-SDC-Daten

```
[6 1]BBC WS [C B] [6 B] 0/7+ 1/7+ 2/7+ 3/7+ 4/7+ 5/7+ 6/7+ 7/7+!0007 ;
[6 1]BBC WS [C B] [6 B] 0/7+ 1/7+ 2/7+ 3/7+ 4/7+ 5/7+ 6/7+ 7/7+!0007 ;
[6 1]BBC WS [C B] [6 B] 0/7+ 1/7+ 2/7+ 3/7+ 4/7+ 5/7+ 6/7+ 7/7+!0007 ;
[6 1]BBC WS [C B] [6 B] 0/7+ 1/7+ 2/7+ 3/7+ 4/7+ 5/7+ 6/7+ 7/7+!0007 ;
[6 1]BBC WS [C B] [6 B] 0/7+ 1/7+ 2/7+ 3/7+ 4/7+ 5/7+ 6/7+ 7/7+!0007 ;
```

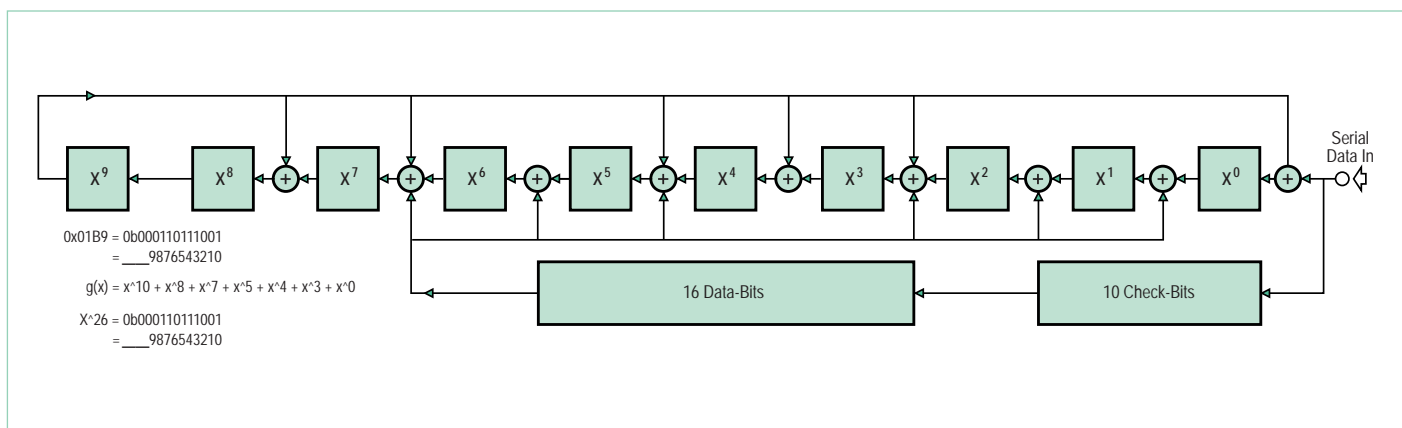


Bild 9. CRC2.

Empfängern kommen nämlich genau die gleichen Verfahren vor, wie sie hier vorgestellt wurden. Dort werden lediglich viel leistungsfähigere Frontends und DSPs eingesetzt.

(120392)

Weblinks

- [1] www.elektor.de/100180
 - [2] www.elektor.de/100181
 - [3] www.elektor.de/100182
 - [4] www.elektor.de/120088
 - [5] www.elektor.de/120089
 - [6] www.elektor.de/120392
 - [7] <http://downloads.bbc.co.uk/rd/pubs/reports/1984-19.pdf>
 - [8] <ftp://ftp.rds.org.uk/pub/acrobat/rbds1998.pdf>
 - [9] www.drm.org/
 - [10] www.ebu.ch/fr/technical/trev/trev_305-murphy.pdf
- www.broadcastpapers.com/whitepapers/ABUBBCamss2006.pdf?CFID=16508900&CFTOKEN=dac28b1a87e54d77-47F9A337-9A30-F5E3-667FAB2A9EA27223

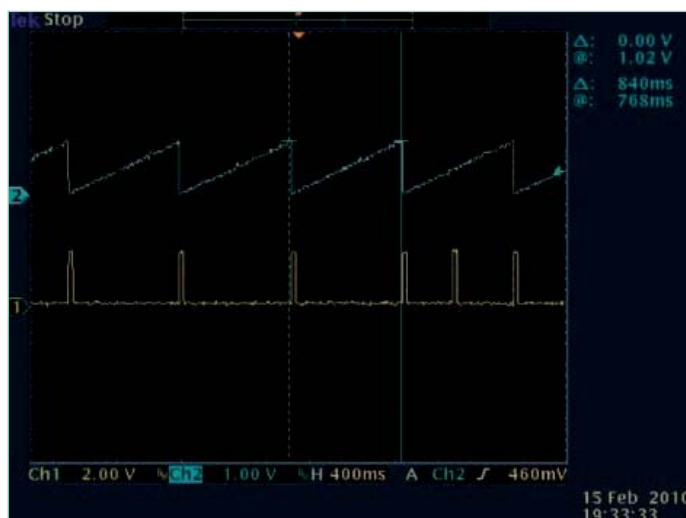


Bild 10. OK-Signale.

Elektor-Produkte und Support

- Signalgenerator (Kit mit Platine und allen Bauteilen 100180-71)
- Universeller Empfänger (Kit mit Platine und allen Bauteilen 100181-71)
- Aktive Ferritantenne (Kit mit Platine und allen Bauteilen 100182-71)
- Kombi-Kit aus allen drei Bauteilsätzen plus USB/TTL-Konverter

BOB FT232: 100182-72

- USB/TTL-Konverter BOB FT232, bestückt und getestet 110553-91
- Gratis Software-Download (Hex-Files und Source-Code)

Alle Produkte und Downloads sind über die Website zu diesem Artikel erhältlich: www.elektor.de/120392

Stromwandler

Professionelle Stromwandler werden oft mit Ringkernen hergestellt und mit hoher Präzision gewickelt. Dieser Beitrag zeigt, wie man Stromwandler selbst aus handelsüblichen Teilen herstellen kann. Damit lassen sich dann sichere Messungen an Stromversorgungen vornehmen.

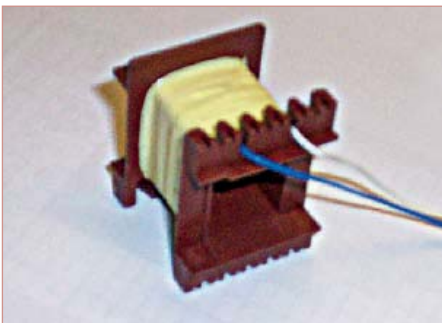


Bild 1. Stromwandler mit Sekundärwicklung auf Spulenkörper.

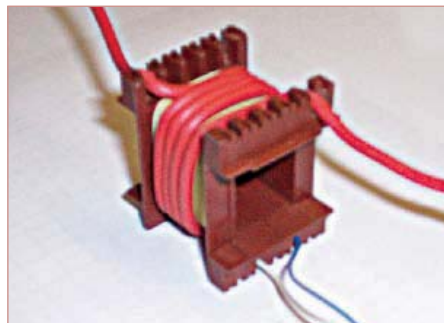


Bild 2. Stromwandler mit zusätzlicher Primärwicklung.

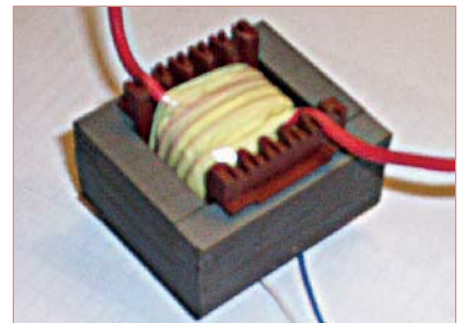


Bild 3. Fertig zusammengefügter Stromwandler.

Von Ed Dinning (UK)

Stromwandler sind kleine Trafos, die hohe Ströme in für Messungen brauchbarere Bereiche transformieren. Sie sind in der Regel zwischen Primär- und Sekundärwicklung isoliert und werden in der Leistungselektronik relativ häufig eingesetzt. Ihr großer Vorteil gegenüber moderneren Ansätzen wie Hall-sensoren oder Rogowski-Spulen ist, dass sie keine Stromversorgung brauchen und nicht nur Messinstrumente, sondern sogar Überstromrelais direkt treiben können.

Parameter

Für solche Wandler ist es üblich, die Leistung bzw. die mögliche Last zu spezifizieren. In professionellen Anwendungen liegt diese eher unter 10 VA. Die Last wird typischerweise in VA angegeben, da Strom und Spannung nicht phasengleich sein müssen; für die Zwecke dieses Beitrags spielt dies aber keine Rolle.

Vielfach genügt es, ein Messinstrument oder einen Komparator als Teil einer Schutzschaltung zu versorgen, wofür eine Belastbarkeit von 0,5 VA ausreicht. Möchte man einen Strom von 10 A messen und diesen

dazu auf 50 mA reduzieren, ergibt sich ein Verhältnis von 200:1. Bei 0,5 VA und 50 mA ergibt sich nach der Leistungsformel ($U = P / I$) eine Spannung von 10 V und nach dem ohmschen Gesetz ($R = U / I$) eine ohmsche Last von 200 Ω .

Nun geht es an die Berechnung der Sekundärwicklung. Für den obigen Wandler genügt eine Kernfläche von 300...400 mm². So einen Kern kann man preiswert kaufen oder aus alten Geräten recyceln. Je größer der Kern, desto weniger Windungen sind notwendig.

Viele kommerzielle Trafos werden anhand

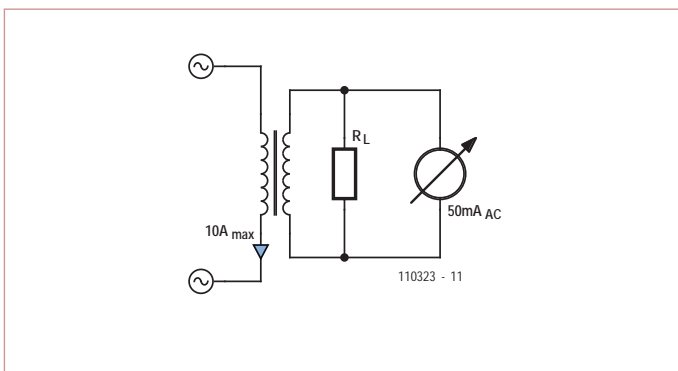


Bild 4. Testschaltung für den Eigenbau-Stromwandler.

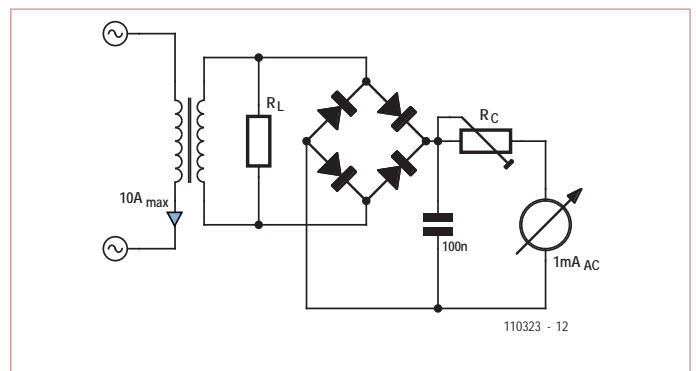


Bild 5. Stromwandler mit Drehspulinstrument.

ihrer EI-Zahl klassifiziert. „EI“ bezieht sich dabei auf E- und I-förmige Bleche. Für das Beispiel genügt ein EI48-Kern mit 20 mm Stapelhöhe. Die „48“ bezieht sich auf die Länge bzw. Höhe von E und I in mm. Der mittlere Balken des E ist 16 mm breit. Bei einem Blechstapel von 20 mm ergibt sich daher ein Kernquerschnitt von:

$$16 \text{ mm} \times 20 \text{ mm} = 320 \text{ mm}^2 = 0,000.32 \text{ m}^2$$

Mit der Trafo-Gleichung berechnen wir die Windungszahl:

$$N = E / (4,44 f B A_e)$$

wobei N = Windungszahl, E = Spannung, f = Frequenz, B = magnetische Flussdichte in Tesla und A_e = Kern-Querschnitt in m^2 .

Die Netzfrequenz beträgt 50 Hz (in den USA 60 Hz). Man betreibt solch einen Wandler üblicherweise bei der relativ niedrigen Flussdichte von 0,2 Tesla (im Gegensatz zu einem „richtigen“ Trafo, der abhängig vom Kernmaterial mit >1 Tesla operiert). Setzt man die Werte in die Gleichung ein, ergibt sich:

$$N = 10 / (4,44 \times 50 \times 0,2 \times 0,000.32) \\ \Rightarrow N = 704 \text{ Windungen (USA: } N = 586)$$

Jetzt könnte man auf 700 Windungen runden. Bei einem Übersetzungsverhältnis von 200:1 ergäbe sich so eine Primärwicklung mit 3,5 Windungen (USA: 2,93). Dies ist nicht praktikabel. Ein Windungsverhältnis von 800:4 (USA: 600:3) ist besser und ergibt eine geringere Flussdichte sowie eine höhere Genauigkeit. Soviele zur Theorie. Nun zur Praxis des Wickelns.

Wickeln

Die Sekundärwicklung soll 50 mA liefern können. Bei Kupferdraht rechnet man mit etwa 3 A/mm². Also ist ein Draht mit einem Durchmesser von 0,15 mm erforderlich. Eine gebräuchliche Drahtstärke ist z.B. 0,16 mm (#34 AWG), wobei die Lack-schicht schon eingerechnet ist. Der nutzbare Wicklungsquerschnitt ist 20 mm breit und 6 mm hoch. Dies erlaubt Lagen zu 125 Windungen. Somit werden 6,4 Lagen für die Sekundärwicklung erforderlich, die man auf 7 Lagen aufrundet, was einer Höhe von 1,2 mm entspricht. Für die Primär-

wicklung bleiben also noch die restlichen 4,8 mm Höhe übrig, siehe **Bild 1**.

Man beachte, dass die Primärwicklung aus Netzleitung gemacht wird, die mindestens 10 A verkraftet (siehe **Bild 2**). Ob Litze oder Voll Draht ist nicht so wichtig. Die wichtige Isolation zwischen Primär- und Sekundärseite ist schlicht die der Leitung. Die 4 Windungen werden einfach auf die Primärwicklung aufgebracht und von einem Stück Isolierband gesichert. Für Trafos spezifiziertes Polyester-Band eignet sich besonders gut für diesen Zweck. Anschließend werden E- plus I-Bleche mit abwechselnder Orientierung in den Spulenkörper eingebracht, sodass sich so gut wie kein Luftspalt ergibt (siehe **Bild 3**).

Der fertige Stromwandler kann nun in der Schaltung nach **Bild 4** ausprobiert werden. Wichtig ist, dass der Lastwiderstand R_L immer angeschlossen ist, da es andernfalls zu ziemlich unerwünschten Effekten kommen kann, da eine offene Sekundärwicklung bei einem Stromwandler zu hohen Spannungen führen könnte. Mit einem Multimeter kann man die sekundäre Wechselspannung messen. Für die Impedanz des Wechselstrombereichs gilt dann: je geringer, desto besser.

Beispiel

Ein praxisgerechter Einsatz für einen Stromwandler ist der direkte Anschluss eines Messinstruments via Gleichrichter wie in **Bild 5**. Ein typisches Drehspulinstrument hat bei einem Strom von 1 mA Vollausschlag. Wenn dies parallel zu R_L angeschlossen wird, fällt es kaum ins Gewicht.

Wie man in der Schaltung sehen kann, liegen bei einem primärseitigen Strom von 10 A auf der Sekundärseite 10 V an. Nach Gleichrichtung hat man dann eine Gleichspannung als Maß für den Strom am Ausgang des Brückengleichrichters.

Wenn man ein 1-mA-Meter an den Gleichrichter anschließen will, braucht man noch einen Serienwiderstand R_C . Nach Herrn Ohm ergibt sich dieser zu 10 k Ω minus dem Innenwiderstand des Messinstruments (typischerweise um die 100 Ω). Mit einem 10-k Ω -Trimpoti kann man die Anzeige schön kalibrieren.

Eventuelle Zeigerunruhe kann man übr-

gens mit Hilfe eines kleinen Sieb-Kondensators dämpfen. Eine Kapazität von 100 nF reicht aus. Größere Werte ergeben eine Spitzenwertanzeige mit einer längeren Rückkehrzeit.

Auch wenn die Sache gut kalibriert ist, wird sie doch in der Nähe des Nullpunkts ziemlich nichtlinear sein, was mit der Durchlasskennlinie der Dioden zu tun hat. Mit Schottky- oder Germanium- statt Siliziumdioden wird es linearer.

Das Prinzip eignet sich auch für Anwendungen bei höheren Frequenzen wie z.B. im Audibereich. Oberhalb von 1 kHz kann man für geringere Verluste Ferritkerne verwenden. Bei Ferriten sollte man allerdings zugunsten der Genauigkeit bei Flussdichten unter 25 mT bleiben.

(120193)

Anzeige



Schaeffer AG

FRONTPLATTEN & GEHÄUSE

Kostengünstige Einzelstücke und Kleinserien

Individuelle Frontplatten können mit dem Frontplatten Designer mühelos gestaltet werden. Der Frontplatten Designer wird kostenlos im Internet oder auf CD zur Verfügung gestellt.

- Automatische Preisberechnung
- Lieferung innerhalb von 5-8 Tagen
- 24-Stunden-Service bei Bedarf

Preisbeispiel: 34,93 € zzgl. USt./Versand

Schaeffer AG
Nahmitzer Damm 32
D-12277 Berlin
Tel +49 (0)30 8 05 86 95-0
Fax +49 (0)30 8 05 86 95-33
Web info@schaeffer-ag.de
www.schaeffer-ag.de

Transkonduktanz-Messverstärker

Spannung hinein, Strom heraus

Ein Transkonduktanz-Verstärker ist ein Differenzverstärker, dessen Ausgangsstrom eine Funktion der Spannungsdifferenz zwischen den Eingängen ist. Mit dem Operationsverstärker AD844 von Analog Devices lässt sich dieses Verhalten unkompliziert realisieren. Dieser Beitrag zeigt, wie mit einem Transkonduktanz-Verstärker ein vielseitiger Vorverstärker für den Einsatz in der Messtechnik aufgebaut werden kann.



Von Wim de Jager (NL)

Zwar wurde der Opamp AD844 in erster Linie für Anwendungen mit Stromrückkopplung entwickelt, doch wegen seiner unkonventionellen Struktur ist er auch „open-loop“ als gesteuerte Stromquelle einsetzbar. Diese Eigenschaft kommt dem Entwickler entgegen, wenn es darum geht, eine spannungsgesteuerte Stromquelle zu realisieren. Verstärker mit diesem „Transkonduktanz“ genannten Verhalten sind in der Messtechnik weit verbreitet.

Der AD844

In **Bild 1** ist das interne Funktionsschema des AD844 dargestellt. Der nicht-invertierende Eingang des Puffers B1 hat eine hohe Impedanz, die Impedanz des invertierenden Eingangs ist niedrig. Der von der Ein-

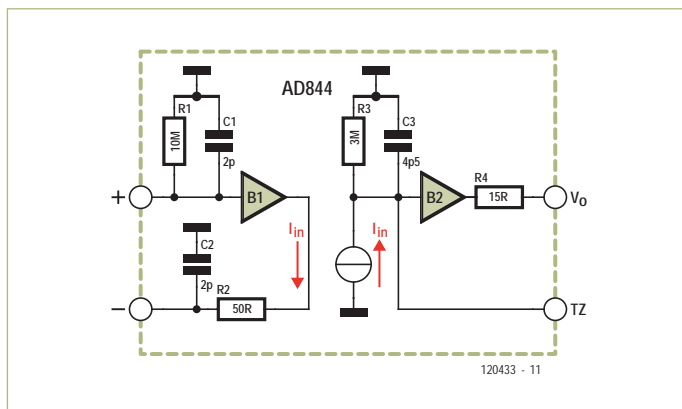


Bild 1. Internes Funktionsschema des AD844.

gangsstufe generierte Strom I_{in} fließt auch über Ausgang TZ, dies ist der Ausgang des Spannung-Strom-Umsetzers (Transkonduktor). Im nachfolgend beschriebenen Messvorverstärker steuert I_{in} ein angeschlossenes externes System. Über den Puffer B2 ist auch die an TZ liegende Spannung am Ausgang verfügbar. R1...R4 und C1...C3 stehen symbolhaft für das nicht-ideale Verhalten des Opamps. Die angegebenen Werte wurden dem Datenblatt von Analog Devices [1] entnommen. Wenn eine Spannungsquelle zwischen den Eingängen liegt, ist $I_{in}/V_{in} = 1/R2 = 20 \text{ mA/V}$. Für den unbelasteten Ausgang TZ gilt $A_o = R3/R2 = 60000$. Die Bandbreite folgt aus $R3 \parallel C3$, was mit den angegebenen Werten $f_{3dB} \approx 12 \text{ kHz}$ ergibt.

Messvorverstärker

Im Vorverstärker, dessen Schaltung **Bild 2** zeigt, ist eine Regelung für die Arbeitspunkt-Stromkompensation eingebaut. Eine Diodenschaltung schützt den nicht-invertierenden Eingang POS.IN vor Spannungsspitzen, eine Overload-Erkennung signalisiert das Auftreten von Überspannungen am Ausgang. Der invertierende Eingang NEG.IN ist bei allen hier betrachteten Anwendungen mit Masse verbunden. Das Eingangssignal liegt über R2 am nicht-invertierenden Eingang POS.IN. Die maximale Eingangsspannung beträgt $5 V_{DC}$, höhere Spannungen werden von D1...D4 zusammen mit R9 und R10 begrenzt. D2 und D3 sind einfache Dioden des Typs 1N4148, Schottky-Dioden würden hier wegen ihres relativ hohen Sperrstroms einen unerwünschten DC-Offset verursachen. Wenn R2 ein ¼-W-Typ ist, werden Spannungen bis $20 V_{DC}$ sicher begrenzt. Ein 2-W-Metallfilm-Widerstand verschiebt die obere Grenze auf $50 V_{DC}$. Hohe Leistungen für R2 wirken sich ebenso wie drahtgewinkelte Typen nachteilig auf das Verhalten im hochfrequenten Bereich aus. Wenn ein Koppelkondensator vorgeschaltet wird, verhindert die Begrenzung, dass die bei ungeladenem Kondensator auftretenden Stromspitzen den AD844 beschädigen. Mit $R3 = 1 \text{ k}\Omega$ beträgt die Transkonduktanz

1 mA/V. Der Widerstand R4 kompensiert den Einfluss von R_i . Wenn R4 den Wert 1 k Ω hat, fließen 5 mA bei 5 V_{DC}, was laut Datenblatt des AD844 die Obergrenze darstellt. So dimensioniert arbeitet der AD844 im optimalen Bereich.

Die Eingangsimpedanz hängt maßgeblich von R1 ab. Der Strom, der bei offenem Eingang POS.IN über den nicht invertierenden Eingang des Opamps fließt, verursacht an R1 einen Spannungsabfall von etwa 10 mV. Da der Offset des AD844 äußerst niedrig ist, kann bei bestimmten Anwendungen (AC-Kopplung oder hohe Quellimpedanz) eine Stromkompensation sinnvoll sein. Da die Stromrichtung wegen der Toleranzen in komplementären Schaltungen nicht festliegt, lässt sich mit Poti P1 die Spannung im Bereich von etwa -23...+23 mV einstellen. Die Widerstände R7 und R8 sorgen für niedrige Widerstände des Poti-Schleifers gegen Masse. Wenn an POS.IN eine Quelle mit niedriger Ausgangsimpedanz liegt, ist das Einstellen entbehrlich. Den internen Offset des AD844 kann die Schaltung mit P1 nicht kompensieren.

Ausgang TZ der gesteuerten Stromquelle ist gleichzeitig der Eingang des internen Puffers B2 (siehe Bild 1). TZ kann jede geeignete Komponente steuern, dies kann ein einzelner Halbleiter ebenso wie ein nachfolgendes System sein. Da der Transkonduktanz-Verstärker nicht rückgekoppelt ist, bestehen im Hinblick auf die Stabilität keine Einschränkungen. Die Impedanz Z_{TZ} verursacht zwar einen geringen Messfehler, doch bei den meisten Anwendungen dürfte dieser Fehler vernachlässigbar sein. Die Spannung an TZ ist über Puffer B2 als Ausgangsspannung V_o verfügbar.

Die Overload-Erkennung ist mit den Dioden D5...D10 aufgebaut. Ausgangsspannung V_o wird von der Schottky-Brücke D5...D8 gleichgerichtet, die gleichgerichtete Spannung steuert über die Zenerdiode D9 die LED D10. Wenn D10 kaum sichtbar aufleuchtet, liegen die Verzerrungen infolge Übersteuerung bei ungefähr 0,5 %. Deutlicheres Aufleuchten von D10 ist ein Zeichen dafür, dass das Signal festläuft. Die RC-Glieder R11/C2 und R12/C3 tragen zur Stabilität des AD844 bei, sie werden von Analog Devices im Datenblatt empfohlen.

Betriebsspannung

Die Betriebsspannung ± 15 V wird, wie in Bild 3 dargestellt, von einem Netztespaltung mit 12 V_{AC}/500 mA bezogen. Da die Last niedrig ist, läuft die Trafospaltung auf etwa 15 V_{AC} hoch. Durch zweifache Einweggleichrichtung entsteht eine symmetrische Betriebsspannung aus nur einer Trafo-Sekundärwicklung. Die Glättungselkos haben relativ hohe Kapazitäten, so dass die Welligkeit gering ist. Die Spannungsregler IC1 und IC2 stabilisieren die symmetrische Spannung, gleichzeitig schützen sie die Ausgänge vor Überlast und Kurzschlüssen (maximaler Strom 100 mA).

Realisierung

Der Versuchsaufbau, den das Foto in Bild 4 zeigt, befand sich in einem geschlossenen Alu-Gehäuse, die Schaltung wurde mit einer kleinen Platine aufgebaut. Die geschlossene, verzinnete Kupferfläche war der gemeinsame Massepunkt. Für die Eingänge und den Ausgang wurden BNC-Buchsen montiert, Ausgang TZ ist außerdem über Bananenbuchsen zugänglich. Die Fassung des AD844 wurde von einer so genannten Minimount-Platine gehalten. Die Dioden

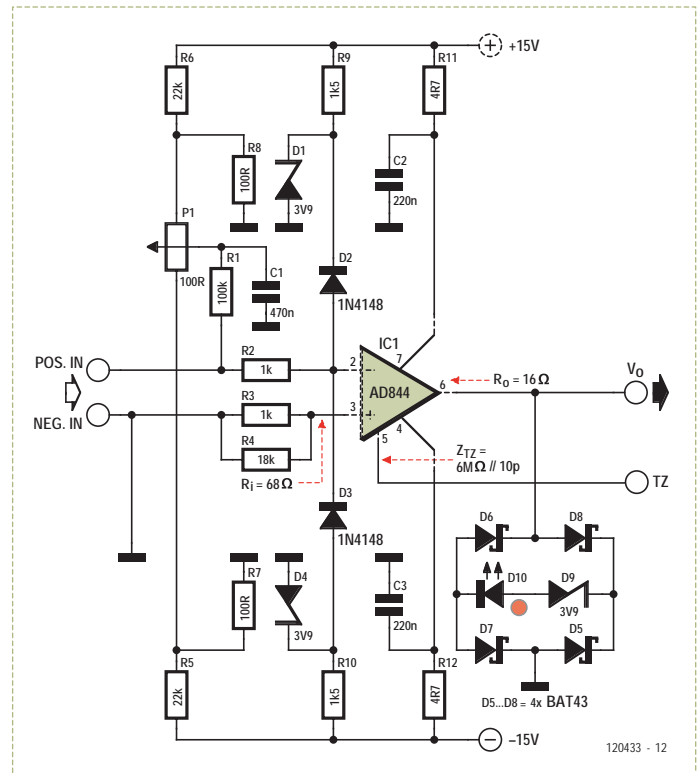


Bild 2. Ein Transkonduktanz-Verstärker mit dem AD844.

D5...D8 waren in der Fassung untergebracht. P1 war ein Mehrgang-Trimpoti mit 15 Umdrehungen, so dass die Stromkompensation exakt eingestellt werden konnte.

Die komplementären bipolaren Transistoren des AD844 zeichnen sich durch eine hohe Transitfrequenz f_T aus. Das Layout einer Platine muss so gestaltet werden, dass der Transkonduktanz-Verstärker nicht unkontrolliert schwingen kann. Gemeinsame Masseflächen sowie kurze Verbindungen zwischen den Komponenten sind gute Voraussetzungen für ein stabiles Verhalten.

Messungen am Transkonduktanz-Verstärker

Die Auswirkungen des internen Offsets beim AD844 wurden mit einem Mikroamperemeter gemessen, das mit Ausgang TZ

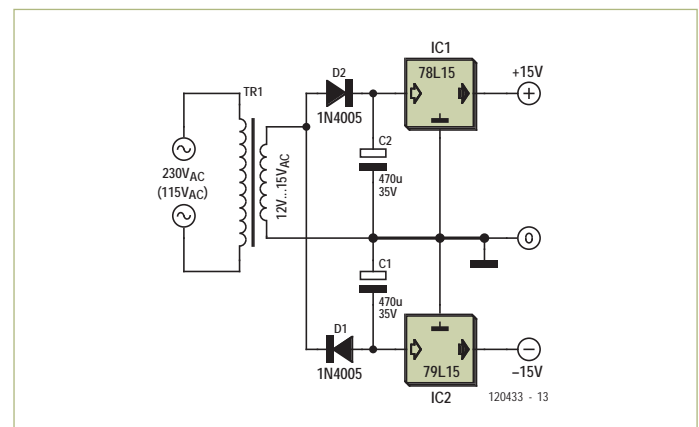


Bild 3. Durch doppelte Gleichrichtung entsteht eine symmetrische Betriebsspannung aus nur einer Trafo-Sekundärwicklung.



Bild 4. Blick auf den Versuchsaufbau des Autors.

bei kurzgeschlossenem Eingang verbunden war. Der gemessene Wert betrug nur 0,1 μA , was hauptsächlich darauf zurückzuführen ist, dass der Offset des AD844 lasergetrimmt ist. Falls notwendig, lässt sich mit der im Datenblatt dokumentierten Offset-Kompensation eine weitere Reduktion erreichen. Bei offenem Eingang POS.IN fällt an R1 infolge des Stroms, der im Nullpunkt über den nicht-invertierenden Eingang des Opamps fließt, eine Spannung von 10 mV ab. Der Strom über Ausgang TZ beträgt folglich 10 μA . Da dieser Wert den Offset deutlich übersteigt, wurde die Schaltung durch die schon beschriebene Ruhestrom-Kompensation ergänzt. Durch sorgfältiges Einstellen kann der Wert unter 0,1 μA gesenkt werden.

Das AC-Verhalten wurde an Ausgang TZ ohne Last mit einem Eingangssignal 0,5 mV bei 500 Hz gemessen, die Verstärkung betrug 6000. Wenn $1 \text{ mA/V} \cdot R_{TZ} = 6000$ ist, folgt daraus, dass $R_{TZ} = 6 \text{ M}\Omega$ beträgt. Die Frequenz der -3-dB-Schwelle liegt bei 2,5 kHz. Da $\omega \cdot R_{TZ} \cdot C_{TZ} = 1$ sein muss, ist $C_{TZ} = 10 \text{ pF}$. Der für R_i gemessene Wert lag bei 68 Ω , ferner war $R_o = 16 \Omega$.

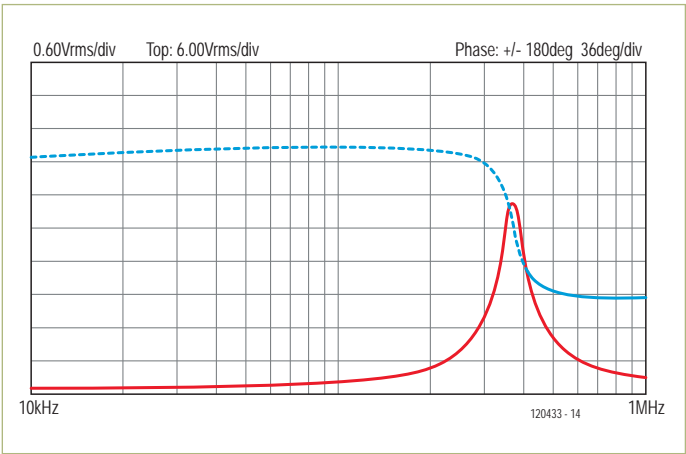


Bild 5. Impedanzmessung an einer Induktivität, zusammen mit einem USB-Oszilloskop, einem Funktionsgenerator und zugehöriger Software.

Messen von Kapazitäten und Induktivitäten

Um einen Kondensator zu messen, wird er mit Ausgang TZ verbunden. An Eingang POS.IN wird das Signal eines Sinusgenerators, beispielsweise mit der Spannung 500 mV gelegt. Die Sinus-Frequenz wird mit hohen Werten beginnend so weit abgesenkt, bis die Spannung an TZ mit der Eingangsspannung übereinstimmt. Bei der zugehörigen Frequenz ist die Verstärkung gleich 1, so dass die Kapazität aus $C = 1/\omega \cdot R$ berechnet werden kann. Ein Beispiel: Für einen Kondensator wurde die Verstärkung 1 bei der Frequenz 16 kHz gemessen. Da $R = R_3 = 1 \text{ k}\Omega$ ist, ergibt sich eine Kapazität von $C = 10 \text{ nF}$. Ein LCR-Meter des Typs Voltcraft 9036 zeigte bei diesem Kondensator die Kapazität 10,47 nF an.

Auch zum Messen von Induktivitäten hat sich die Signalspannung 500 mV als zweckmäßig erwiesen. Die Frequenzvariation in Richtung von niedrigen zu hohen Werten verhindert unnötige Übersteuerung. Bei einer Ringkerndrossel betrug die Frequenz 56 kHz, als die Verstärkung 1 erreicht war. Aus der Rechnung $L = R/\omega$ folgt der Wert $L = 2,84 \text{ mH}$. Das LCR-Meter von Voltcraft wies den Wert

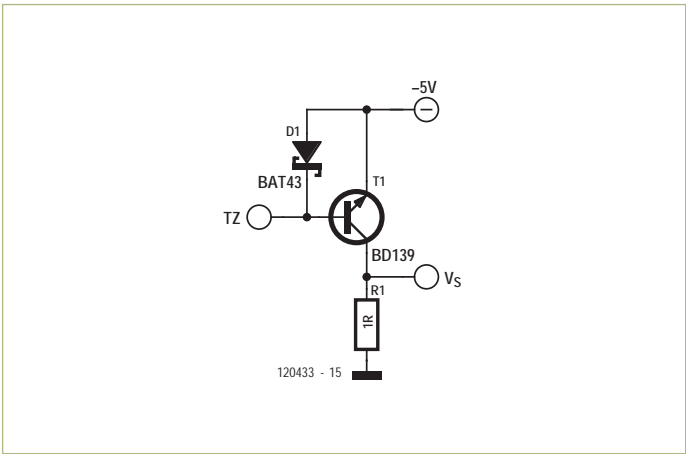


Bild 6. Schaltung zum Messen von NPN-Transistoren.

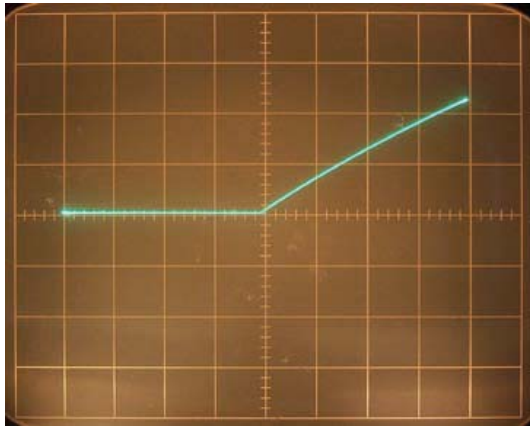


Bild 7. Das Foto entstand bei Messungen an der Schaltung aus Bild 6. Skalierung: Horizontal 1 mA/div, vertikal 0,2 A/div.

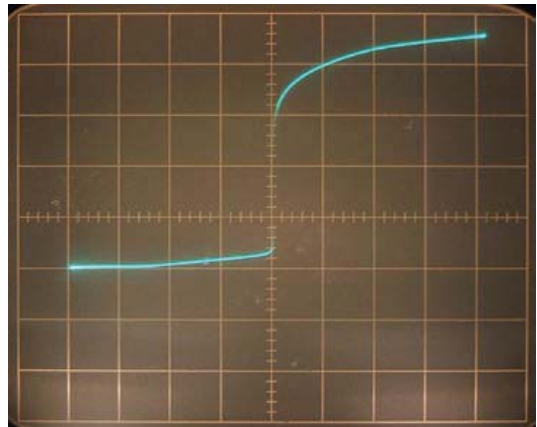


Bild 8. Die Charakteristik einer Zenerdiode. Skalierung: Horizontal 0,5 mA/div, vertikal 1 V/div.

2,86 mH aus.

Es ist auch möglich, die Wickelkapazität C_p aus der Frequenz zu berechnen, bei der Eigenresonanz auftritt. Die Ringkernndrossel zeigte Resonanzerscheinungen bei 360 kHz. Aus $C_p = 1/\omega_0^2 \cdot L$ folgt rechnerisch $C_p = 69$ pF. Von diesem Wert muss $C_{TZ} = 10$ pF subtrahiert werden, so dass $C_p = 59$ pF ist. Bei Eigenresonanz ist die Impedanz gleich dem parallelen Lastwiderstand, er kann durch Messen der Verstärkung bestimmt werden. Im betrachteten Fall lag die Verstärkung bei 38, woraus $R_p = 38 \cdot R_3 = 38$ k Ω folgt.

Impedanzmessungen mit Velleman PCSGU 250

Im USB-Oszilloskop PCSGU 250 von Velleman ist ein Funktionsgenerator integriert. Zusammen mit der zugehörigen Software ist der Funktionsgenerator auch als System-Analyser einsetzbar. Beispielsweise können Bode-Diagramme erstellt werden, die das Amplitudenverhältnis und die Phasenverschiebung bei der Übertragung als Funktion der Frequenz darstellen. Da der Transkonduktanz-Verstärker Spannungen in Ströme umsetzt, handelt es sich um eine Messung einer (komplexen) Impedanz. Ein solches Messsystem, das die Phase einbezieht, wird auch „Vektor-Impedanzmesser“ genannt. **Bild 5** gibt die Impedanz- und Phasencharakteristik der Ringkernndrossel wieder. Die Phasendrehung beträgt bis etwa 200 kHz recht genau 90°, was bei einer Induktivität nicht anders zu erwarten war. Bei 370 kHz tritt die Eigenresonanz in Erscheinung, so dass bei dieser Frequenz der parallele Lastwiderstand bestimmt werden kann. Da die Signalspannung 100 mV beträgt, beläuft sich die Skalierung für die Amplituden-(Y-)Achse auf 6 k Ω /div. Der hier für R_p gemessene Wert beträgt 34 k Ω .

Messungen an Halbleitern

Die Stromsteuerung ist auch zum Messen von Halbleiter-Charakteristiken nutzbar, zum Beispiel bei bipolaren Transistoren. Eine Messschaltung für NPN-Transistoren ist in **Bild 6** angegeben. Die Betriebsspannung -5 V auf der Seite des Emitters erlaubt, den Messwiderstand R1 an Masse zu legen. Die Spannung an R1, dessen Wert 1 Ω beträgt, ist ein Maß für den Kollektorstrom. Ausgang TZ des Transkonduktanz-Verstärkers wird mit der Basis verbunden, die Basissteuerung beträgt 1 mA/V. Schottky-Diode D1 schützt T1 vor zu hohen Sperrspannungen. Die β -Charakteristik wurde mit einem Oszilloskop gemessen, **Bild 7** zeigt das Ergebnis. Die vertikale Achse

gehört zur Spannung an R1, die horizontale Achse zur Spannung an Eingang POS.IN. Vertikal ist die Empfindlichkeit auf 0,2 V/div eingestellt, was dem Kollektorstrom 0,2 A/div entspricht. Die horizontale Empfindlichkeit beträgt 1 V/div, gleichbedeutend mit dem Basisstrom 1 mA/div. Da das Signal an V_S negativ ist, arbeitet das Oszilloskop im inversen Betrieb (nicht an jedem Oszilloskop vorhanden!), das Ergebnis erscheint daher positiv. Aus der Charakteristik geht hervor, dass für $I_C = 400$ mA die Stromverstärkung $\beta = 118$ ist. Das zweite Oszilloskop-Foto in **Bild 8** stammt von einer Messung an einer Zenerdiode. Der Strom durch die Zenerdiode wird auf der horizontalen Achse dargestellt. Die Skalierung beträgt 0,5 V/div, dies entspricht dem Zenerdioden-Strom 0,5 mA/div. Auf der vertikalen Achse ist die Spannung an der Zenerdiode mit der Skalierung 1 V/div aufgetragen. Beim Strom 2 mA beträgt die Zenerspannung 3,6 V, was mit dem Datenblatt des Herstellers übereinstimmt.

Vorverstärker

Ein DC-gekoppelter Vorverstärker lässt sich unkompliziert realisieren, indem an den Ausgang TZ ein Lastwiderstand gelegt wird. Ein Beispiel: Mit $R_L = 10$ k Ω ist die Verstärkung $R_L / R_3 = 10$. Die Bandbreite ist aus $R_L | C_{TZ}$ ableitbar, sie beträgt 1,6 MHz. Die rechnerischen Werte lassen sich durch Messungen verifizieren. Der Vorverstärker ist beispielsweise geeignet, die Empfindlichkeit eines Messsystems zu erhöhen.

Puffer

Der Ausgangspuffer kann auch als unabhängiger Spannungsfolger dienen, der Eingang ist TZ, während POS.IN an Masse gelegt wird. Die Bandbreite des Spannungsfolgers liegt bei 10 MHz. Die Eingangsimpedanz ist $Z_{TZ} = 6$ M Ω | 10 pF, die Ausgangsimpedanz ist $R_O = 16$ Ω . Die maximale Ausgangsspannung beträgt 5 V_{eff}, wobei die THD-Verzerrungen (1 kHz) nicht höher als 0,7 % sind. Der maximale Ausgangsstrom beläuft sich auf 80 mA_{DC}, der Ausgang ist kurzschlussfest.

(120433)gd

Weblink

[1] www.analog.com

• **Subscribe** to *audioXpress* magazine!

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

Then you should be **reading audioXpress!**

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

What will you find in *audioXpress*?

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

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

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

audioXpress



 <p>Entwicklung industrietauglicher Software und Hardware sowie Elektronik 03303/212166 oder www.jasys.de</p>	<p>Gravuren / Schilder / Frontblenden www.webgrav.de</p>	<p>LCDs und mehr www.lcd-store.de www.LC-Design.de www.crystalfontz.de</p>	<p>Leiterplatten für Pragmatiker!</p>  <p>www.pcb-joker.com</p>
<p>Alles Spule!</p> <p>Wir liefern und fertigen: Drähte, HF-Litzen, Ferrit- und Eisenpulverkerne, Spulenkörper, Isoliermaterial, Klebebänder, Tränklacke, Übertrager, RFID-Spulen, Sensor- und Aktorspulen, Prototypen, Kleinserien, Serien, Ersatzteile und vieles mehr.</p> <p>MM Menting Mikroelektronik www.spulen.com</p>	<p>W2micro www.w2micro.de</p> <ul style="list-style-type: none"> • diverse SMD-Adapter • RS232-TTL-Pegelwandler • Sonderelektronik (z.B. Audioverteiler) • Leiterplattenbestückung und Kabelkonfektionierung ab 1 Stück 	<p>LOETRONIC Embedded MP3 Module www.loetronic.com</p>	<p>www.anttronic.de ab 1 Stck. ANT TRONIC Leiterplatten zu TOP-Preisen!!</p>
<p>Auch Ihr Unternehmen ist eine Anzeige wert!</p>	<p>Bausätze zu ELEKTOR 1986 bis heute!</p> <p>Teilesätze, Platinen, programmierte Controller sowie Cds zu fast allen Elektor-Projekten vom Spezialist. Alle Elektor-Artikel zum Verlagspreis.</p> <p>Ihr zuverlässiger Partner für aktive und passive elektronische Bauteile und Komponenten:</p> <p>Geist Electronic-Versand GmbH Tel.: 07720/36673 Fax: 07720/36905 Mail: info@geist-electronic.de Shop: www.geist-electronic.de</p>	<p>SCOPES und mehr</p> <p>HAMEG® Instruments A Rohde & Schwarz Company</p> <p>MESSTECHNIK zum fairen Preis</p>	<p>Kleinrechner mit FPGA www.bomerenzprojekt.de</p>
<p>Günstige Platinen clever bestellen!</p>  <p>www.pcb-joker.com</p>	<p>NienTech</p> <p>SCHNITTSTELLENWANDLER von</p> <p>WLAN LAN USB nach</p> <p>RS485 RS422 RS232 TTY</p> <p>über virtuellen COM-Port ansprechbar</p> <p>www.NienTech.de</p>	<p>USB-ISOLATOR</p>  <p>Galvanische Trennung Störfreie Messungen Sicherheit und Schutz</p> <p>CESYS www.cesys.com Elektronik · Messtechnik · FPGA</p> <p>Baugruppenbestückung vom Prototypen bis zur Serie FS-ELECTRONIC.de</p>	<p>Ausgabe: Dezember 2012</p> <p>Anzeigenschluss: 30. Oktober 2012</p> <p>Erscheinungstermin: 21. November 2012</p>

www.elektor.de

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

Zurück zu den Wurzeln (8)

NF-Vorverstärker

Eines der wichtigsten Einsatzgebiete des Transistors ist die Verstärkung von Tonsignalen. Man spricht hier auch vom Niederfrequenz-(NF-)Verstärker. Egal ob als Mikrofon-Vorverstärker oder Teil einer Radioschaltung: Aus leise mach laut!

Von Burkhard Kainka (D)

Das Prinzip eines solchen (möglichst) linearen Verstärkers besteht darin, dass der kleine Basisstrom durch einen NF-Strom moduliert (vergrößert und verkleinert) wird, sodass der verstärkte Kollektorstrom entsprechend verstärkte NF-Signale enthält. Die Emitterschaltung ist dabei die meist eingesetzte Verstärkerschaltung. Der Name soll andeuten, dass hier der Emmitter an das gemeinsame Massepotential der Schaltung gelegt wird. Gegenüber der Masse variieren die Basisspannung und die Kollektorspannung, aber die Emitterspannung bleibt immer gleich, z.B. 0 V. In Schaltbildern verwendet man manchmal das Massezeichen statt einer durchgehenden Masseleitung. Die Masse ist meist mit dem Minuspol der Stromquelle verbunden.

So ein Verstärker sollte einen möglichst großen Aussteuerbereich haben, bevor Verzerrungen auftreten (und die schön gleichmäßige Verstärkung in die Knie geht). Dazu muss der Kollektor-Ruhestrom die Hälfte

des maximal möglichen Stroms betragen, den der Kollektorwiderstand und die Betriebsspannung festlegen. Dann nämlich lässt sich der Strom in beide Richtungen gleich weit verändern, ohne dass es zu einer Begrenzung kommt. Die Einstellung des richtigen Ruhestroms durch einen passenden Basiswiderstand ist aber nicht ganz einfach.

Sehen wir uns einmal **Bild 1** an, hier haben wir die Bauteile optimal dimensioniert, wenn der Transistor BC547B zufällig genau eine 300-fache Stromverstärkung aufweist. Der Basisstrom beträgt $I_B = U / R = (9 \text{ V} - 0,6 \text{ V}) / 560 \text{ k}\Omega = 15 \mu\text{A}$. Der Kollektorstrom wird damit $I_C = I_B * V = 15 \mu\text{A} * 300 = 4,5 \text{ mA}$. Am Kollektorwiderstand fällt damit eine Spannung $U = I * R = 4,5 \text{ mA} * 1 \text{ k}\Omega = 4,5 \text{ V}$ ab. Der Rest von 9 V, also ebenfalls 4,5 V, liegt noch zwischen Emmitter und Kollektor. Das ist der Idealfall, denn nun kann der Ausgang in beide Richtungen etwa gleich weit ausgesteuert werden. Die Spannung am Kollektor könnte also zwischen 0 V und 9 V schwanken und z.B.

einen reinen Sinuston mit maximaler Ausgangsspannung unverzerrt übertragen. Die Spannungsverstärkung ist in diesem Fall mehr als 100-fach.

Aber wenn Sie diese Schaltung mit mehreren Transistoren so aufbauen, werden Sie jedes Mal ein anders Ergebnis bekommen, weil die Stromverstärkung stark variiert. Im schlimmsten Fall ist der Verstärkungsfaktor eines Transistors mit 600 so groß, dass der Transistor bereits ganz durchgeschaltet ist. Dann wird das nichts mit den verzerrungsarmen Ausgangssignalen. Aber da gibt es ein geeignetes Gegenmittel: Die Gegenkopplung.

Gegenkopplung

Die einfachste Gegenkopplung besteht darin, dass der Basiswiderstand nicht mehr an die Betriebsspannung, sondern an den Kollektor gelegt wird (**Bild 2**). Die Faustregel lautet: $R_B = R_C * V$, wobei für V die mittlere zu erwartende Stromverstärkung eingesetzt wird. Für eine 300-fache Stromverstärkung und einen Kollektorwiderstand

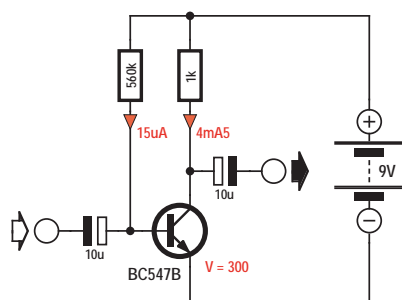


Bild 1. Einstellung des Arbeitspunktes bei genau 300-facher Verstärkung.

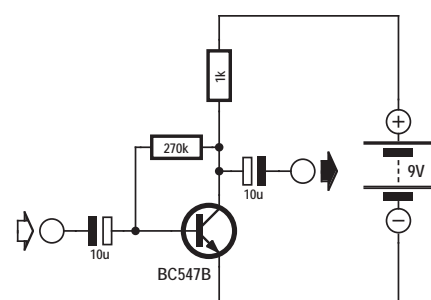
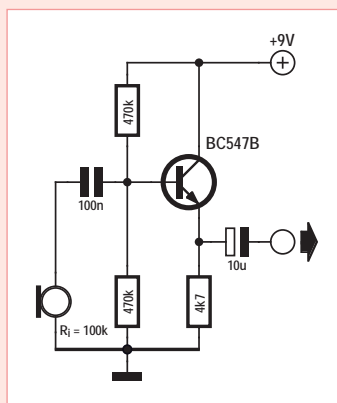


Bild 2. Arbeitspunkteinstellung mit Gegenkopplung.

Der Emitterfolger



Emitterstrom erheblich zu ändern. Die Emitterspannung liegt immer

Der Emitterfolger wird auch als Kollektorschaltung bezeichnet, denn der Transistor arbeitet mit konstanter Kollektorspannung. Der Verstärkerausgang liegt am Emitter. Jede Änderung der Eingangsspannung führt dazu, dass die Emitterspannung automatisch nachgeführt wird, weil schon eine sehr kleine Abweichung der Basis-Emitter-Spannung ausreicht, um den

etwa um 0,6 V tiefer als die Basisspannung.

Eine Änderung der Basisspannung um 1 V wird daher auch die Emitterspannung um fast 1 V ändern. Der genaue Wert könnte bei 0,99 V liegen, wenn für die Änderung des Kollektorstroms eine Änderung der Basis-Emitter-Spannung um 10 mV erforderlich war. Die Spannungsverstärkung ist also fast 1, d.h. die Eingangsspannung wird nicht verstärkt, wohl aber der Eingangsstrom. Der Vorteil des Emitterfolgers ist sein großer Eingangswiderstand. Während der Eingangswiderstand einer Emitterschaltung je nach Arbeitspunkt bei etwa 1 kΩ liegt, erreicht die Kollektorschaltung 100 kΩ und mehr. Man kann z.B. ein Kristallmikrofon oder einen Kristall-Tonabnehmer direkt anschließen. Mit dieser Schaltung lässt sich auch ein einfacher piezokeramischer Schallwandler als Mikrofon (auch für Körperschallwellen) einsetzen. So lässt sich z.B. der Pulsschlag abhören.

von 1 kΩ sollte also ein Basiswiderstand von etwa 300 kΩ verwendet werden.

Diesmal reagiert die Schaltung viel gutmütiger auf unterschiedliche Stromverstärkungen. Ein größerer Verstärkungsfaktor führt zu einem größeren Kollektorstrom und damit zu einem größeren Spannungsabfall am Kollektorwiderstand. Die Kollektor-Emitter-Spannung sinkt gleichzeitig, und damit auch die Spannung am Basiswiderstand, was wiederum zu einem verkleinerten Basisstrom führt. Im Endeffekt wird also ein größerer Verstärkungsfaktor durch einen kleineren Basisstrom teilweise ausgeglichen, sodass der Kollektorstrom weniger steigt. Man spricht hier von einer Gegenkopplung, weil der Vergrößerung des Kollektorstroms entgegengewirkt wird. Die Gegenkopplung führt hier auch zu einer geringeren Verstärkung, zugleich aber zu weniger Verzerrungen. Das Wichtigste ist

jedoch: Diesmal kann man irgendeinen NPN-Transistor aus der Bastelkiste greifen, die Schaltung funktioniert immer. Oder ingenieurmäßig korrekt ausgedrückt: Die Schaltung verhält sich tolerant gegenüber Exemplarstreuungen.

Eine noch genauere Einstellung des Arbeitspunktes erfordert etwas mehr Aufwand. Der Transistor erhält dafür einen Emittterwiderstand (**Bild 3**). Mit einem Basis-Spannungsteiler wird eine feste Teilspannung eingestellt, wobei der Strom durch den Spannungsteiler rund 10 Mal größer als der Basisstrom sein soll, damit keine Rückwirkungen durch unterschiedliche Basisströme auftreten. Die Emitterspannung stabilisiert sich hier selbst auf einen Wert, der um die Basis-Emitterspannung niedriger als die Basisspannung liegt. Dadurch ist der Emitterstrom stabil und damit auch der Kollektorstrom.

In der Beispielschaltung wird eine Basisspannung von 3 V vorgegeben. Die Emitterspannung stellt sich auf 2,4 V ein. Bei einem Emittterwiderstand von 1 kΩ ergibt sich ein Emittterstrom von 2,4 mA. Am Kollektorwiderstand von 1 kΩ fällt ebenfalls eine Spannung von 2,4 V ab. Es stellt sich also eine Kollektorspannung von 6,6 V bzw. eine Kollektor-Emittter-Spannung von 4,2 V ein.

Die Schaltung weist eine starke Spannungs-Gegenkopplung auf. Eine kleine Änderung der Emitterspannung wirkt sich direkt auf die Basis-Emittter-Spannung aus und führt aufgrund der steilen Basis-Kennlinie zu einer starken Änderung des Kollektor- und Emittterstroms. Da nur sehr kleine Änderungen der Basis-Emittter-Spannung nötig sind, stellt sich die Emitterspannung auf einen Wert ein, der immer um ca. 0,6 V unter der Basisspannung liegt. Die Funktion ähnelt der des sogenannten Emitterfolgers (siehe Kasten).

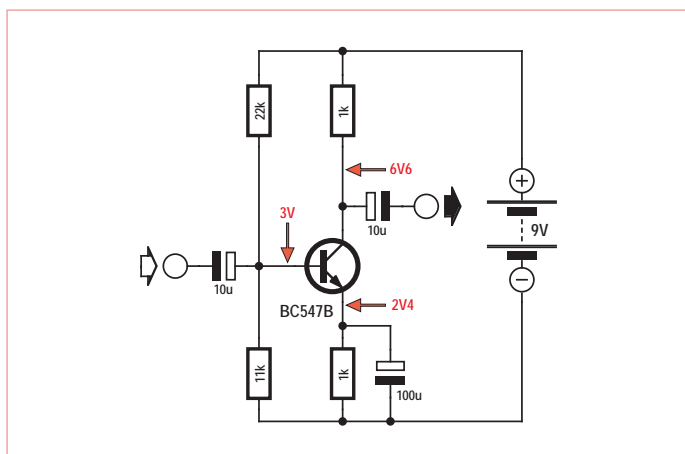


Bild 3. Stabilisierung des Arbeitspunktes.

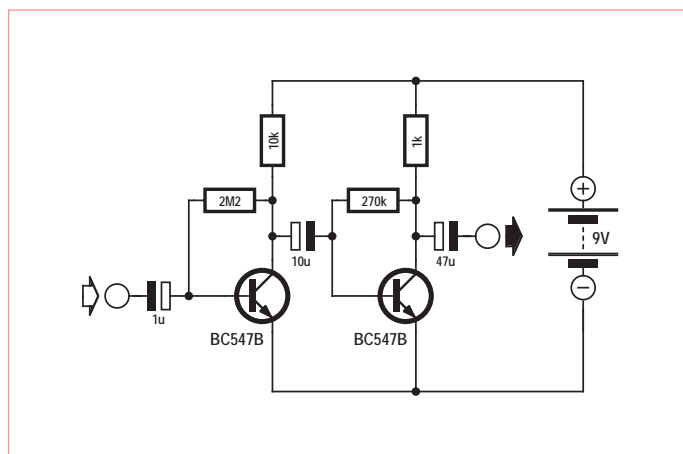
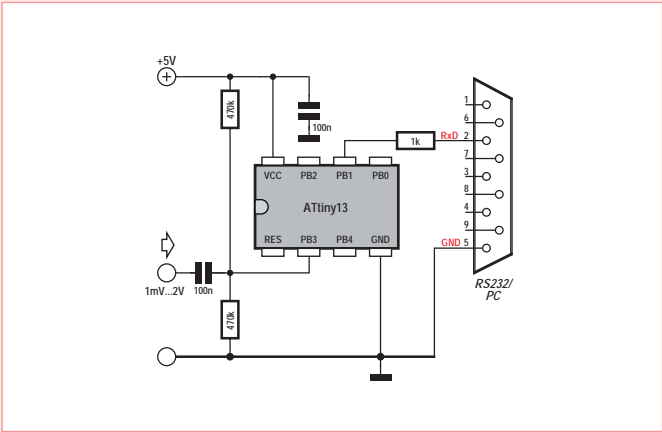


Bild 4. Ein zweistufiger NF-Verstärker.

Ein NF-Millivoltmeter

Für die Untersuchung der hier vorgestellten Verstärker kann man ein NF-Millivoltmeter gut gebrauchen. Eigentlich kann der A/D-Wandler im ATtiny13 nur Gleichspannungen messen. Aber durch das Hochlegen der mittleren Spannung auf 2,5 V und schnelle Messungen können auch Wechselspannungen untersucht werden. Das Ergebnis wird seriell an den PC gesendet und kann mit einem Terminalprogramm angeschaut werden.



Das Prinzip des Messprogramms ist einfach. Zuerst wird die Mittenspannung durch eine Mittelwertbildung bestimmt. Dann folgt eine schnelle Serie vieler Einzelmessungen, bei der die Mittenspannung subtrahiert und dann jeweils der Absolutwert gebildet wird. Eigentlich liegt die kleinste messbare Spannungsstufe des A/D-Wandlers bei 5 mV. Durch die Mittelung über viele Einzelmessungen kommt man dennoch bis herunter auf 1 mV. In der eigentlichen Mess-Schleife werden 2780 Messungen durchgeführt, verarbeitet und aufsummiert. Die Messpunkte liegen übrigens völlig asynchron zum Messsignal. Aber die große Anzahl der Messungen und der Zufall führen zum Ergebnis, solange das Eingangssignal etwa zwischen 50 Hz und 50 kHz bleibt.

Wie kommt es aber zu dieser ominösen 2780? Diese Zahl geht von einer Referenzspannung von genau 5 V aus und berücksichtigt schon den Unterschied zwischen arithmetischer Mittelung und echter Effektivmessung, sodass das Ergebnis im Fall eines Sinussignals tatsächliche Effektivwerte in mV_{eff} zeigt. Um genaue Effektivwerte zu ermitteln, müsste man eigentlich das Quadrat der Spannung aufsummieren und dann später aus dem Mittelwert die Wurzel ziehen. Damit wäre der ATtiny13 aber überfordert. Deshalb wird hier der arithmetische Mittelwert der Absolutspannung gebildet. Dieser ist

um rund 10 % zu niedrig, genauer gesagt um die Wurzel aus 2 geteilt durch pi/2, also 0,9003. Ein A/D-Schritt beträgt 5000 mV / 1023 = 4,8876 mV. Außerdem wird der Messwert einmal mit 8 multipliziert und dann durch 4096 geteilt, also effektiv durch 512 geteilt. Damit am Ende alles korrekt in mV angezeigt wird, muss man also mit 512 * 4,8876 / 0,9003 = 2780 multiplizieren; und das geht am einfachsten, wenn man 2780 Messwerte aufsummiert.

Das Ergebnis ist erstaunlich genau. Auch Werte wie 1 oder 2 mV_{eff} werden stabil angezeigt!

```
'Millivoltmeter 1 mVeff ... 2000 mVeff
$regfile = "attiny13.dat"
$crystal = 1200000
$hwstack = 8
$swstack = 4
$framesize = 4

Dim U1 As Integer
Dim U2 As Integer
Dim U3 As Long
Dim N As Integer

Config Adc = Single , Prescaler = Auto
Start Adc
Open "comb.1:9600,8,n,1,INVERTED" For Output As #1

Do
    U2 = 0
    For N = 1 To 64
        U1 = Getadc(3)
        U2 = U2 + U1
    Next N
    Shift U2 , Right , 3 ' /8
    U3 = 0 ' Nullpunkt
    For N = 1 To 2780
        U1 = Getadc(3)
        Shift U1 , Left , 3 ' *8
        U1 = U1 - U2
        U1 = Abs(U1)
        U3 = U3 + U1
    Next N
    Shift U3 , Right , 12 ' / 4096
    Print #1 , U3
Loop

End
```

Die starke Gegenkopplung führt jedoch im Prinzip zu einer sehr geringen Spannungsverstärkung. Damit die Verstärkung für NF-Signale hoch genug ist, wird die Gegenkopplung für Wechselströme aufgehoben, und zwar durch einen Emitterkondensator. Sein Wert richtet sich nach der tiefsten zu verarbeitenden Frequenz und sollte großzügig bemessen werden.

Zwei Stufen
Wenn die Verstärkung eines Transistors nicht reicht, muss ein mehrstufiger Verstärker eingesetzt werden. Bild 4 zeigt eine Version mit zwei Stufen und RC-Kopplung. In den meisten Fällen reicht die einfache Form der Gegenkopplung mit einem Widerstand zwischen Basis und Kollektor völlig aus, um einen geeigneten Arbeitspunkt einzustellen.

len. Die Schaltung zeigt ein weiteres Prinzip der Verstärkerentwicklung: Von links nach rechts wird die Schaltung immer niederohmiger. So erreicht man einen hohen Eingangswiderstand und einen geringen Ausgangswiderstand. Das reicht zwar noch nicht zum Anschluss eines Lautsprechers, aber mit einem Kopfhörer am Ausgang ist bereits so manche Aufgabe lösbar.

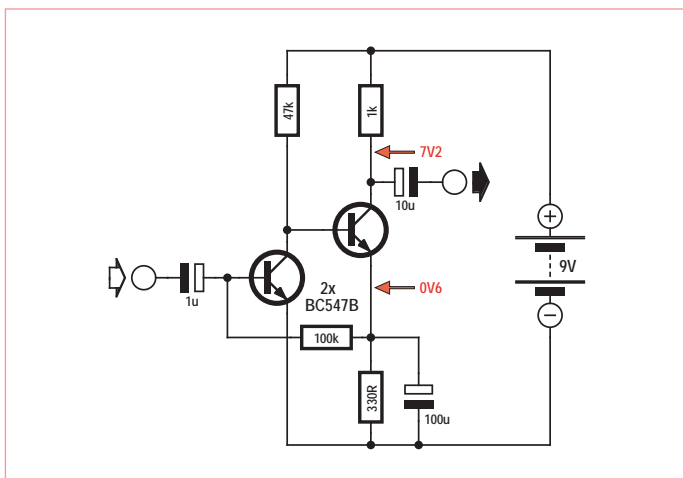


Bild 5. Ein zweistufiger NF-Verstärker mit Gleichstromkopplung.

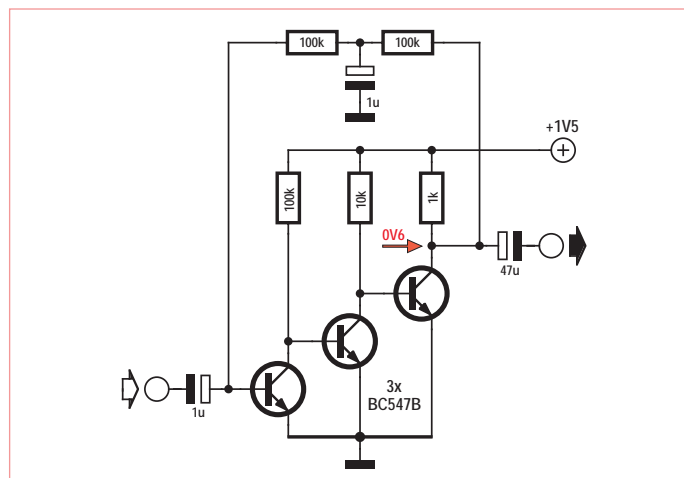


Bild 6. Ein dreistufiger, direkt gekoppelter Verstärker.

Gleichstromgekoppelte Stufen

Mehrstufige NF-Verstärker können mit geringerem Aufwand gebaut werden, wenn man auf die Kondensatorkopplung zwischen den einzelnen Stufen verzichtet und stattdessen eine Gleichstromkopplung verwendet. Der Kollektorwiderstand des ersten Transistors wird hier zugleich der Basiswiderstand des zweiten (Bild 5). Oft findet sich dabei auch eine einfache Möglichkeit der Arbeitspunkteinstellung, mit der gleich beide Transistoren passend mit Basisstrom versorgt werden.

Bild 5 zeigt einen zweistufigen, direkt gekoppelten Verstärker. Der Basisstrom des ersten Transistors wird vom Emitter der zweiten Stufe abgezweigt. Damit ergibt sich eine Gegenkopplung und Stabilisierung des Arbeitspunktes. An einem relativ kleinen Basiswiderstand von 100 k Ω kann der Spannungsabfall vernachlässigt werden, und damit muss die Emitterspannung am zweiten Transistor etwa 0,6 V betragen. Der Emitterwiderstand von 330 Ω bestimmt somit den Emitter- und Kollektorstrom der zweiten Stufe, der weitgehend unabhängig von der Betriebsspannung ist.

Die Gegenkopplung soll nicht zu einer Verringerung der Wechselspannungsverstärkung führen. Ein zusätzlicher Emitterkondensator sorgt deshalb dafür, dass nur Gleichstrom zurückgekoppelt wird. In der Praxis hat der Verstärker eine untere Grenzfrequenz, die mit einem größeren Emitterkondensator weiter herabgesetzt werden kann.

Drei Stufen

Noch einfacher liegen die Verhältnisse bei einem dreistufigen Verstärker, weil hier

eine Gegenkopplung direkt vom Ausgang auf den Eingang möglich ist (Bild 6). Jede Stufe dreht nämlich die Phase um 180 Grad. Bei zwei Stufen käme es zu einer Rückkopplung wie bei einem Flipflop. Bei einer oder drei Stufen ergibt sich jedoch eine Gegenkopplung, die den Arbeitspunkt stabilisiert.

Bild 6 zeigt eine Schaltung, die sich gut für kleine Betriebsspannungen ab etwa 1 V eignet. Hier gilt analog wie in Bild 2, dass eine größere Verstärkung der Transistoren zu einem größeren Kollektorstrom am dritten Transistor und damit zu einem größeren Spannungsabfall an dessen Kollektorwiderstand führt. Die Kollektor-Emitter-Spannung sinkt, und damit auch die Spannung, die über den zwei 100-k Ω -Widerständen der Gegenkopplung abfällt. Dies führt schließlich wie schon oben erklärt zu einem verkleinerten Basisstrom.

Die Kollektor-Emitter-Spannung beträgt hier jeweils nur etwa 0,6 V, so dass in jeder Stufe nur eine relativ geringe NF-Verstärkung möglich ist. Doch das wird durch die hohe Anzahl der Verstärkerstufen mehr als ausgeglichen.

Der Gegenkopplungszweig der Schaltung enthält ein Tiefpassfilter mit einem Kondensator gegen Masse (Bypass-Kondensator), der die Gegenkopplung für höhere Frequenzen aufhebt. Er ist erforderlich, um die hohe NF-Verstärkung realisieren zu können. Die Schaltung eignet sich z.B. als empfindlicher Mikrofon-Vorverstärker oder als NF-Verstärker in einfachen Empfängern. Ein Kopfhörer kann direkt statt des Kollektorwiderstands der dritten Stufe eingesetzt

werden. Die Schaltung wurde übrigens im HF-Schnüffler „Tapir“ aus Elektor 7/8 2012 verwendet (www.elektor.de/120354).

(120008)

Quiz-Lösung aus dem Juniheft

Der richtige Lösungscode lautet „CDG“. Hier die Erklärung:

Lösung 1:

An der LED fällt rund 1,8 V ab, U_{CE} am Transistor ist ca. 0,1 V. Es bleiben 7,1 V übrig. 1 k Ω und 470 Ω bilden einen Spannungsteiler mit rund 3,1:1. $7,1 \text{ V} / 3,1 = 2,3 \text{ V}$ (Lösung C).

Lösung 2:

Damit der linke Transistor zu leiten beginnt, muss seine Basisspannung um ca. 0,6 V über der Emitterspannung liegen, also bei $2,3 \text{ V} + 0,6 \text{ V} = 2,9 \text{ V}$. Die richtige Lösung ist D.

Lösung 3:

Wenn der rechte Transistor ganz ausgeschaltet ist und der linke ganz an, findet man am Emitterwiderstand ca. $9 \text{ V} / 11 = 0,8 \text{ V}$. Dazu kommt eine Basis-Emitter-Spannung von 0,6 V, die der linke Transistor braucht, um genügend zu leiten. Der untere Umschaltunkt liegt daher bei ca. 1,4 V (G).

PicoScope 2205-MSO unter der Lupe

Von Thijs Beckers und Jan Buiting (Elektor International)



Ein so genanntes MSO (Mixed Signal Oscilloscope) kombiniert ein Oszilloskop mit einem Logic-Analyser, deren Anzeigen bei Bedarf auf einem Bildschirm erscheinen. Als das Einsteiger-MSO 2205 von Pico Technology [1] auf unter 400 € fiel, beschlossen wir, ein Muster zu besorgen und es im Elektor-Labor zu testen.

Wir erhielten das komplette Kit mit dem eigentlichen Gerät, einem Paar passiver x1/x10-Prüfspitzen sowie einem Logik-Kabel mit Test-Clips. Das MSO 2205 entspricht

der typischen Aufmachung der PicoScope-Geräte: ein blaues ABS-Gehäuse mit zwei BNC-Buchsen an der Frontplatte. Der Logic-Analyser hat 16 Eingänge. Die zugehörigen gelben Litzen entspringen einem gemeinsamen 20-poligen Stecker für die Frontplatte und haben am anderen Ende individuelle kleine Buchsen. An diese kann man dann nach Bedarf kleine rote Clips anstecken (maximal 16). Außerdem gibt es noch vier schwarze Masseleitungen mit passenden Clips.

Zur Stromversorgung des 2205 genügt der USB-Anschluss eines PCs. An einem passiven Hub arbeitet das Gerät allerdings nicht. Zwar ist es auch mit USB 1.1 kompatibel, doch USB 2.0 ist empfohlen. Auf der Rückseite findet sich neben der USB-Buchse noch eine mit „AWG“ bezeichnete BNC-Buchse, die das Signal des Tongenerators liefert.

Software: PicoScope 6

Interessant an der PicoScope-Serie ist, dass alle Varianten die gleiche Software verwenden. Im Moment ist das PicoScope 6. Die Software stellt sich selbst auf die angeschlossene Hardware ein. Die auf der mitgelieferten CD enthaltene Version war PicoScope Version 6.6.14. Laut Hersteller-Webseite war sie aktuell und somit kein Update nötig. Wir installierten die Software und klickten routinemäßig auf „check for update“. Prompt wurde uns Version 6.6.13 als „Update“ angeboten! Kaum zu glauben, dass im Internetzeitalter eine CD aktueller als das Netz sein kann...

Die Installation verlief problemlos (Windows XP SP2, Vista oder 7 wird vorausgesetzt). Nach dem Anschluss der Hardware klickt man auf den Wizard „New Hardware Found“ (nicht via Windows Update). Anschließend ist das Gerät sofort betriebsbereit. Beim ersten Start ist Kanal A aktiviert und man kann direkt die ersten Messungen vornehmen. Selbstverständlich lässt sich die

Startkonfiguration jederzeit ändern.

Alle Elektor-Redakteure haben mehr als einen Bildschirm am PC. Das brachte uns auf die Idee, anstelle eines Funktionsgenerators aus dem Elektor-Labor ein zweites PicoScope (Modell 3206B) mit seinem AWG-Ausgang als Signalquelle für das MSO 2205 zu verwenden. Das geht sehr einfach an einem einzigen PC, indem man die Software zweimal startet, das 3206B einem Bildschirm zuordnet und dem 2205 einen weiteren (siehe **Bild 1**). Einziger Nachteil: Man sieht nicht sofort, welcher Bildschirm jetzt welchem Gerät zugeordnet ist. Man muss ins Hilfenü und dort auf „About PicoScope 6“ klicken, um Informationen über die Software und die damit verbundene Hardware zu bekommen. Dann kann es losgehen.

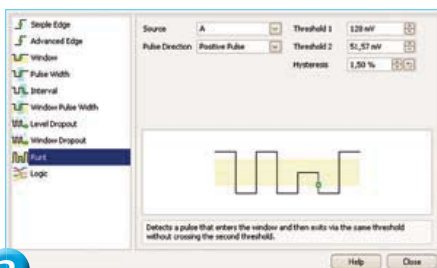
Gefällt uns

Die beiden analogen Eingänge von Kanal A und B produzierten eine sehr stabile Anzeige über den ganzen Frequenzbereich. Alle gewohnten Einstellungen sind verfügbar: Trigger (mit vielen Optionen, siehe **Bild 2**), Auswahl der Prüfspitze, Achsenkalibrierung, Tiefpass-Filter, XY-Modus und anderes mehr. Alles funktioniert prima. Die Software kann in den Spektrum-Modus mit einer Bandbreite von 25 MHz umgeschaltet werden. Dabei kann man überall im Spektrum zoomen.

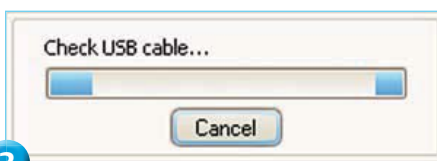
Für ein MSO in dieser Preisklasse ist ein integrierter Funktionsgenerator für frei definierbare Wellenformen (Arbitrary Wave Generator) schon ein super Extra. Auch die Frequenz ist bis zu 100 kHz frei einstellbar. Es existieren etliche voreingestellte Kurvenformen: Sinus, Rechteck, Dreieck, Sinus(x)/x, weißes Rauschen und eben ein freier Modus, bei dem man ohne Begrenzungen seine eigene Kurvenform mit einer Auflösung von bis zu 16.384 Samples zeichnen kann.



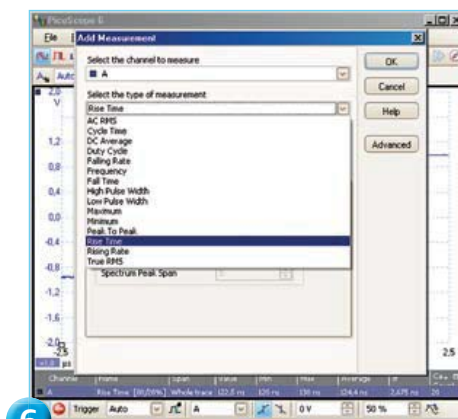
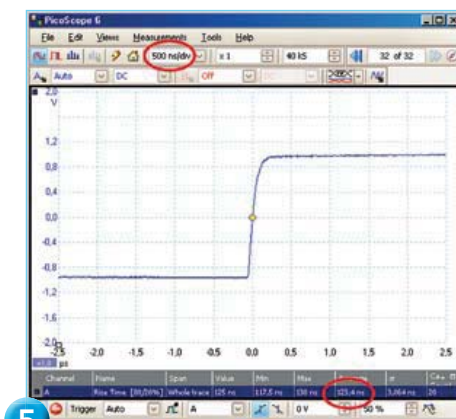
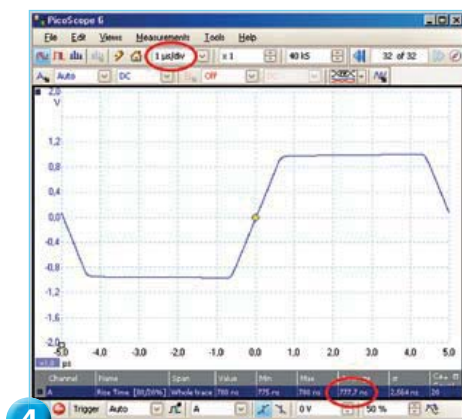
1



2



3



Bei diesen eindrucksvollen Analogfunktionen vergisst man fast die 16 digitalen Eingänge. Man kann sehr einfach auswählen, welche der digitalen Eingänge aktiviert werden sollen. Außerdem ist pro acht Eingänge ein Schwellwert definierbar. Man kann Eingänge auch als Bus organisieren. Die digitalen können zusammen mit den analogen Signalen angezeigt werden. Die Größe des Anzeigenfensters lässt sich nach Wunsch verändern.

Die Software verhielt sich sehr stabil. Selbst das Abziehen des USB-Kabels im Betrieb hat keinen Crash zur Folge. Stattdessen wird man freundlich mit „Check USB Cable“ auf diesen Fehler hingewiesen (**Bild 3**). Steckt man die Hardware wieder ein, macht die Software an dem Punkt weiter, wo sie aufgehört hat.

Ein besonders hilfreiches Feature der Software ist ihre Fähigkeit zur Anzeige aufgezeichneter Kurvenzüge. Bis zu 10.000 Bildschirminhalte rückwärts reicht das Gedächtnis. Man kann recht einfach in diesem großen Puffer blättern.

CPU-Last

Wenn man Kurvenzüge in Echtzeit auf dem PC-Bildschirm anzeigen will, dann ist dazu schon eine gewisse Rechenleistung erforderlich. Unser Test-PC enthielt eine nicht so aktuelle Pentium-Dual-Core-CPU vom Typ E5400 mit 2,7 GHz sowie 4 GB RAM. Die CPU-Last variierte von praktisch 0 % (ohne Messungen) bis hin zu heftigen 60 % (beide analoge Kanäle mit einer Rate von 30 Messungen pro Sekunde). Recht nichtlinear war der Zusammenhang mit der eingestellten Zeitachse: Während 100 ms/Teilstrich nur etwa 5 % der CPU kosteten, waren es bei 200 ms/Teilstrich und mehr plötzlich bis zu 55 % - außer für 1000 s/Teilstrich (!), wo die CPU-Last erstaunlicherweise auf nahezu 0 % sank. Bei kürzeren Zeiten ($\leq 2 \mu\text{s}$ /Teilstrich)

wuchs die CPU-Last wieder bis auf 20...25 %. Im Überlagerungsmodus, bei dem mehrere Kurvenzüge übereinander dargestellt werden, waren Spitzen bis zu 70 % zu sehen. Der Spektrum-Modus ist mit Last-Spitzen bis zu 40 % weniger auffällig.

Gefällt uns weniger

Auch wenn wie bei fast allen Oszilloskopen ein X/Y-Modus verfügbar ist, so ist es doch nicht einfach, hier die Darstellung zu erhalten, die man erwarten würde. Ein Test mit einem im Entstehen begriffenen Elektor-Projekt strapazierte unsere Geduld massiv, als wir durch alle möglichen Zeit-Einstellungen scrollen mussten, um zur gewünschten Darstellung zu kommen. Anders als bei unseren Versuchen mit einem preiswerten LeCroy WaveAce 224 DSO funktionierte es zwar, aber idealerweise sollte man direkt sehen können, was gerade passiert, statt sich wie wild durch alle möglichen Einstellungen hangeln zu müssen, bis das gewünschte Resultat erscheint. Probleme mit dem X/Y-Modus scheint es mehr oder weniger bei allen digitalen Oszilloskopen zu geben, und vermutlich muss man damit einfach leben.

Eine zweite Schwäche ergab sich bezüglich der Leistung des Arbitrary Wave Generators. Von etwa 10 kHz aufwärts zeigt sich ein deutlicher Jitter im erzeugten Signal. Die so erzeugten Kurven resultierten in ziemlich instabilen Anzeigen auf dem Oszilloskop-Schirm – völlig unabhängig von der Amplitude und am auffälligsten bei Rechteck und selbst generierten Wellenzügen. Es sah so aus, als hätte das Trigger-System Probleme, doch wenn wir den gleichen Eingang mit einem identischen Signal eines „richtigen“ externen Generators fütterten, war die Anzeige absolut stabil. Laut Pico Technology liegt dies an Quantisierungsphänomenen bei einer festen Abtastrate, da das exakte Timing der Flanken dynamisch ange-

passt werden müsste, damit die Kurve über mehrere Kurvenzüge im Mittel die gleiche Form hat.

Eine weitere Sache bei unserem Test war die Anstiegszeit, die anscheinend mit der Aquiszeit zusammenhängt. Beispiele zeigen die **Bilder 4** und **5**. Bei $1 \mu\text{s}$ /Teilstrich wurde die Anstiegszeit eines Signals mit 778 ns angegeben. Umgeschaltet auf 500 ns/Teilstrich ergab das gleiche Signal aber eine Anstiegszeit von 123 ns. Letzterer Wert war korrekt, sofern man den Angaben eines zum Vergleich herangezogenen LeCroy-Oszilloskops trauen kann. Pico Technology erklärt hierzu: Zwecks einer verbesserten Darstellung der Kurven wurde die Auflösung der Software auf 12 bit erhöht. Dies begrenzt dann die Slew-Rate und macht aus einem scharfen Anstieg einen exponentiellen. Wenn die Verbesserung ausgeschaltet ist, ergibt der Betrieb mit nunmehr 8-bit-Auflösung keine Unterschiede mehr in der Anstiegszeit.

Übrigens verwendeten wir zur Erfassung der Anstiegszeit ein nettes kleines Utility mit vielen Messmöglichkeiten (siehe **Bild 6**).

Wertung

Unserer Meinung nach hat das neue Familienmitglied der Pico-Serie wirklich viel fürs Geld zu bieten. Das Kit ist sehr praktisch, da alle benötigten Kabel und die Software dabei sind, so dass man direkt loslegen kann. Man braucht zwar einen relativ leistungsfähigen PC, aber so einer ist mittlerweile Standard. Eine Warnung noch: Man muss schon wissen, was man messen möchte, sonst trifft man gerade bei Messgeräten leicht eine suboptimale Kaufentscheidung.

(120091)

Weblink

[1] www.picotech.com

Signalgenerator TF801D/1 AM RF von Marconi

„Heute wird so etwas nicht mehr gebaut!“

Von Stephan Germann (Schweiz)

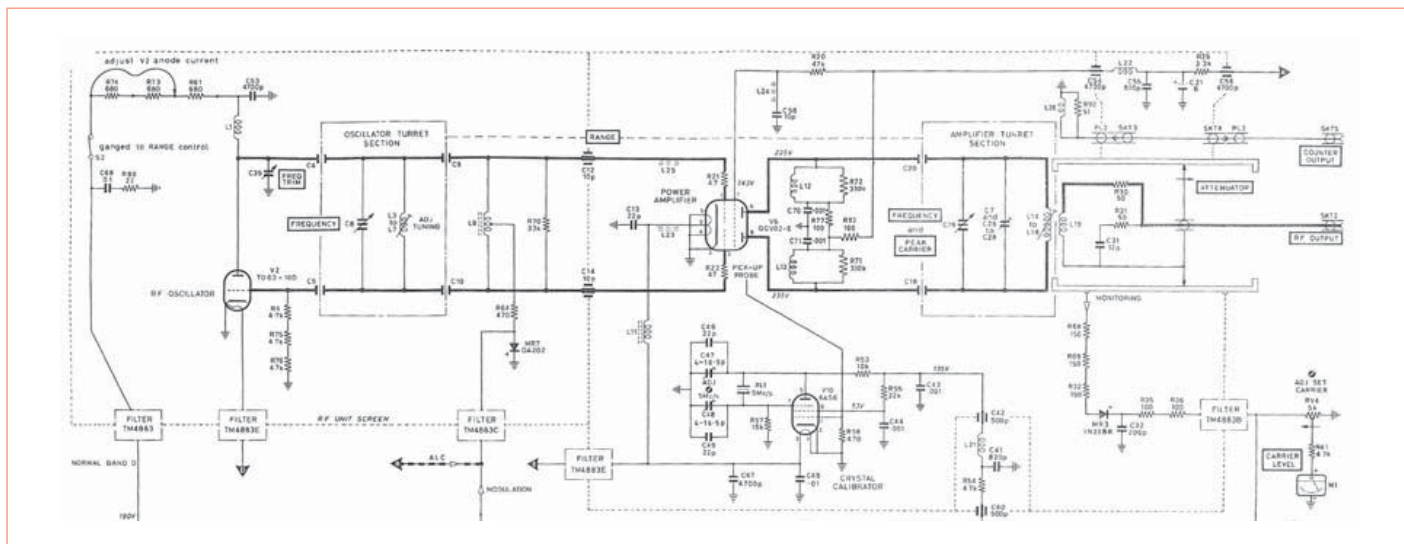
Der HF-Generator TF801D/1 von Marconi Instruments erzeugt Signale mit Frequenzen von 10...470 MHz. Das ist für ein Gerät von 1960 ziemlich gut. Und schon aufgrund seiner aufwändigen mechanischen Konstruktion war es damals recht teuer. Ich bekam diesen Generator vor 15 Jahren als Zugabe zum Kauf eines Spektrum-Analysers. Seitdem hat er mir stets treu beim Bau und der Reparatur von Transceivern für Funkamateure gedient. Die Daten des Geräts sind selbst noch nach heutigen Kriterien gut. Der einzige Schwachpunkt ist die Frequenzstabilität, die selbstverständlich nicht mit modernen DDS-Generatoren mithalten kann. Mit etwas Geduld und Erfahrung kann man aber sogar SSB-Empfänger für das 2-m-Band überprüfen.

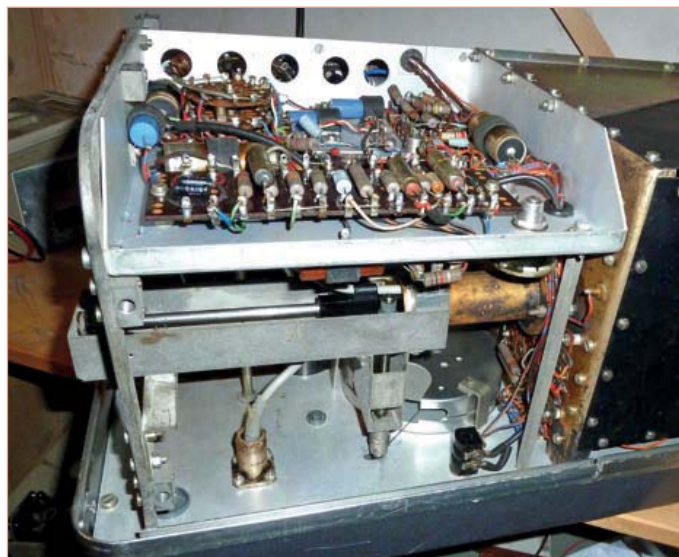
Ich nutze den Generator für die Überprüfung und den Abgleich von Funkamateurer-Apparaten sowie für die Fehlersuche und den Abgleich selbstgebauter Antennen, Filter oder Anpassungsnetzwerke. Dank des großen Frequenzbereichs werden die Bänder mit Wellenlängen von 30 m bis 70 cm abgedeckt. Auch alle üblichen Zwischenfrequenzen liegen in diesem Bereich. Der enorme Bereich der Ausgangsspannung von 100 nV bis hin zu 1 V erlaubt sowohl die Überprüfung empfindlicher Eingangsstufen als auch der Treiberstufen von Sendern. Selbst für passive Messbrücken reicht der Pegel aus. Dank Röhrentechnik ist das Gerät sehr robust. Die Präzision der Ausgangsspannung und das völlige Fehlen von HF-Lecks finde ich nach wie vor toll. Auch die Bedienung ist im Vergleich zu moderner Technik sehr bequem. Tatsächlich ist die Ergonomie besser als bei

manch moderner Einstellerei mit verschachtelten Menüs oder mehrfach belegten Drehknöpfen. Hier hat nämlich jede Funktion ihren eigenen Knopf; und überflüssige Funktionen gibt es einfach nicht.

Mit Ausnahme eines defekten Elkos waren keine weiteren Reparaturen notwendig, was sicherlich auf die hohe Qualität der eingesetzten Bauteile zurückzuführen ist. Sogar die Potis sind versiegelte Drahtausführungen. Für Modulationstiefe und HF-Ausgangspegel gibt es getrennte Drehspulinstrumente. Die Bedienelemente sind großzügig über die Frontplatte verteilt. Alle Knöpfe und Hebel sind von ergonomischer Größe. Die Frequenz-Skala ist so groß und genau, dass selbst im UHF-Bereich ein Fehler von weniger als 100 kHz ohne weitere Messgeräte möglich ist. Der verwendete Antrieb funktioniert präzise und auch eine Feineinstellung ist vorgesehen.

Der mechanische Aufbau des TF801D ist extrem stabil. Allein schon das enorme Gewicht von 30 kg zeugt von der massiven Technik. Da der Generator in Großbritannien hergestellt wurde, sind durchweg nichtmetrische, imperiale Schrauben und Bolzen zu finden. Der innere Aufbau basiert auf einem stabilen Chassis, auf das der Modulator, die Steuerung und das Netzteil montiert sind. Daran befestigt ist auch der HF-Teil in einem HF-dichten Gehäuse aus versilberten Kupferblechen. Das externe HF-Gehäuse enthält den Abstimm-Mechanismus und die HF-Filter für die Zuführung der Steuerspannung. Die HF-Elektronik steckt in einem HF-dichten Gehäuse, das in zwei Teile für den Oszillator und die Endstufen unterteilt ist. Dank dieses Aufbaus zeigt sich die HF wirklich nur am Ausgang. An anderen Stellen im Gerät ist sie nicht nachweisbar. Das ist kein schlech-





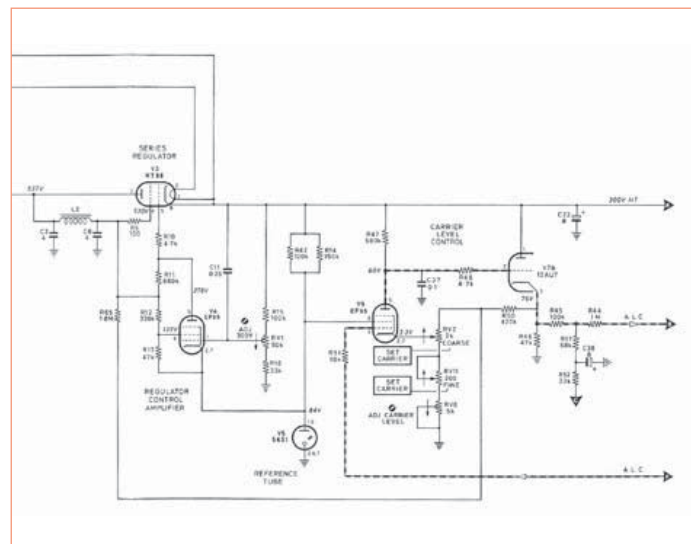
ter Wert, wenn man bedenkt, dass die Ausgangsstufe immerhin 100 mW an HF-Leistung bringt.

Die Elektronik ist rund um zehn Röhren aufgebaut. Hinzu kommen einige Halbleiter-Dioden und mit einem OC71 ein einziger Germanium-Transistor. Teile der Schaltung wurden hier abgedruckt, doch für eine vollständige Wiedergabe ist sie zu umfangreich. Das Netzteil besteht aus einem Trafo, der Vollweg-Gleichrichterröhre 5R4GY und einer Stabilisierung der Spannung mit der Referenz-Röhre 5651, der Steuerröhre EF95 (6AK5W) und der Audio-Ausgangsröhre KT66 als Leistungsregler. Die Heizspannung der Oszillator-Röhre stabilisiert ein Transduktor (eine Art gekoppelte Drossel, mit der man Wechselströme einstellen kann).

Das HF-Signal wird „direkt“ erzeugt. Der Oszillator schwingt einfach auf der gewünschten Frequenz und sein Ausgangssignal wird von einer HF-Tetrode gepuffert. Der Oszillator besteht aus der UHF-Triode TD03-100 und fünf umschaltbaren Spulen für die Frequenzbereiche. Er wird mit Hilfe eines einstellbaren Kondensators abgestimmt. Es ist sogar ein Kühlkörper an der Anode der Triode angebracht.

Das Oszillatorsignal treibt direkt eine symmetrische Push-Pull-Ausgangsstufe mit der UHF-Doppeltriode QQV02-6. Diese Röhre wird im Klasse-A-Betrieb gefahren, was minimale Verzerrungen (wenig „Harmonische“) und eine hohe Verstärkung garantiert. Mit Hilfe der Gleichspannung an den Steuergittern kann die Verstärkung eingestellt oder bei Bedarf die Amplitude moduliert werden. Dank dieser Verstärkungseinstellung ist der Ausgangspegel unabhängig von Frequenz und Belastung, was das Messen an und den Abgleich von Filtern sehr vereinfacht.

Die Anodenbeschaltung des Verstärkers enthält fünf umschaltbare Spulen. Zusammen mit einem zweiten einstellbaren Kondensator fungieren die Spulen als synchron einstellbare Resonanzschwingkreise. Um Fehler zu vermeiden kann der zweite Abstimmkondensator gegenüber dem ersten leicht verstimmt werden – und zwar mechanisch, was aufwändig ist. Die lineare Verstärkung der Push-Pull-Ausgangsstufe reduziert harmonische Verzerrungen auf ein Minimum. Das Ausgangssignal wird induktiv in ein Widerstandnetzwerk ausgekoppelt, das als Abschwächer einen kalibrierten Bereich von 135 dB aufweist. Resultat ist eine frequenzunabhängige, hochstabile Ausgangsimpedanz von 50 Ω .



Der Umschalter für die Frequenzbereiche mit fünf Stufen ist ebenfalls erwähnenswert. Er schaltet nämlich nicht direkt von einer Stufe auf die nächste, sondern benötigt zum Weiterschalten immer eine volle Umdrehung. Der Schalter ist sogar kontaktlos, denn die Verbindung mit den Spulen ist kapazitiv ausgeführt. Kontaktprobleme, welche die Güte der Schwingkreise beeinflussen könnten, sind dadurch komplett eliminiert.

Das Handbuch beschreibt nicht nur den täglichen Einsatz, sondern auch sehr detailliert Wartung und Reparatur – und zwar sowohl mechanisch (z.B. Ersatz des Bowdenzugs) als auch elektronisch. Ein Auszug zum Schmunzeln: „Zuerst schaut man nach angebrannten Widerständen, um den fehlerhaften Schaltungsteil zu lokalisieren.“

(120169)

Retronik ist eine monatliche Rubrik, die antiker Elektronik und legendären ELEKTOR-Schaltungen ihre Referenz erweist. Beiträge, Vorschläge und Anfragen telegraphieren Sie bitte an editor@elektor.com.

Hexadoku

Sudoku für Elektroniker

Im Moment gerade keine Lust aufs Entwickeln, Programmieren und Löten? Dann wird es wieder Zeit für eine Runde Rätselspaß. Wie jeden Monat kommt hier unsere ganz besondere Sudoku-Variante. Füllen Sie überall die richtigen Hexadezimal-Ziffern ein und schicken Sie uns diejenigen in den grauen Kästchen zu. Hier warten vier schöne Elektor-Gutscheine!

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

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

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

Mitmachen und gewinnen!

Unter allen internationalen Einsendern mit der richtigen Lösung verlosen wir

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

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

Einsenden

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

Elektor – Redaktion Süsterfeldstr. 25 52072 Aachen

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

Als Betreff bitte nur die Ziffern der Lösung angeben!

Einsendeschluss ist der 31. Oktober 2012!

Die Gewinner des Penta-Hexadokus aus der Sommerausgabe stehen fest!

Die richtige Lösung ist: D89276.

Der Elektor-Gutschein über 100 € geht an: Norbert Breuker aus Münster.

Einen Elektor-Gutschein über je 50 € haben gewonnen: Jean-Marie Bellet, Donald Hoogendorp und Mike Parker.

Herzlichen Glückwunsch!

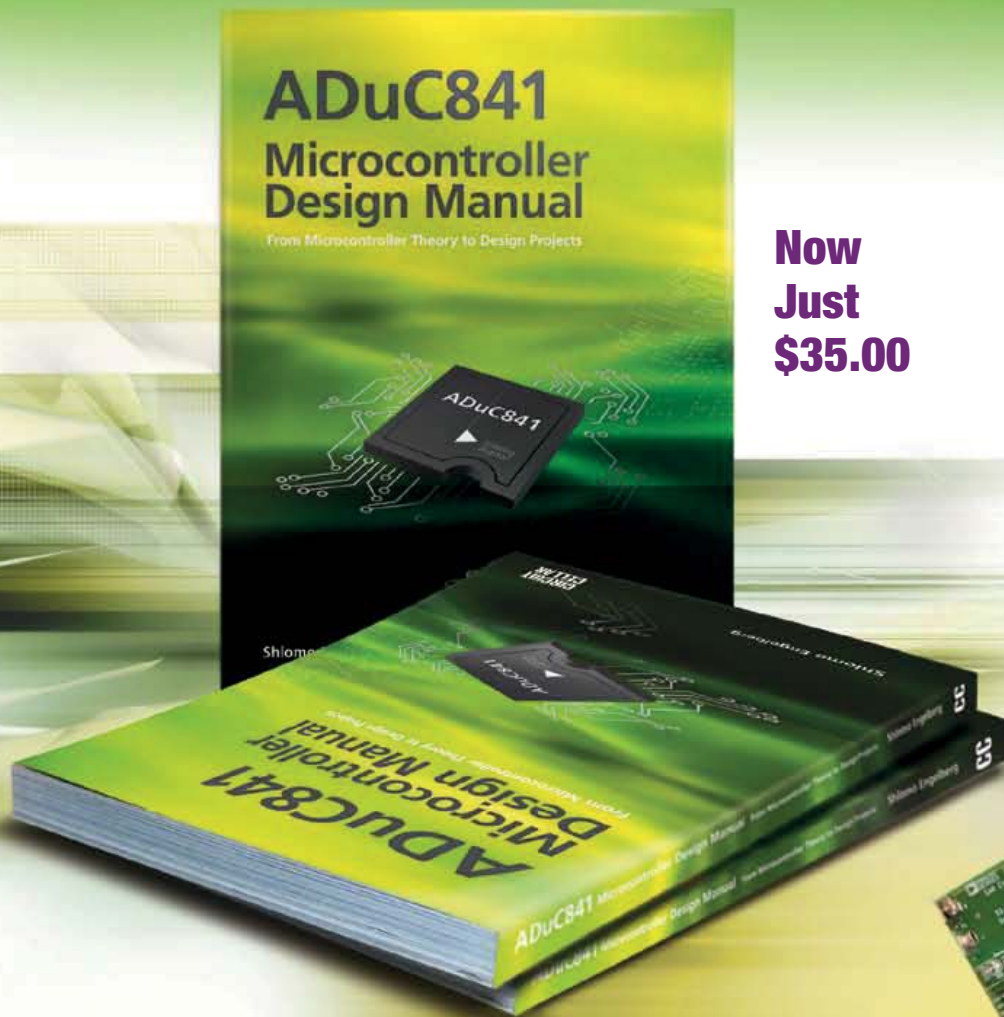
	9			5	F	6	C					D	1		
5				1		E	B	4							0
	C		D	0	7	B		F		9		E			
0		8	4	C		3	A						6	9	
4			6					D		3		C			A
	F	5	A				D		9	1	B	4			
		D	0	E		4				C	5	F		8	
		1			F				7		D			5	6
D	3		9		1				A				4		
	0		E	D	C			2		F	1	7			
			C	A	E	6		9				2	8	D	
1			8		4		5					9			B
	8	F					A	D		4	5	0		C	
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

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

CIRCUIT CELLAR

ADuC841 Microcontroller Design Manual: From Microcontroller Theory to Design Projects

If you've ever wanted to design and program with the ADuC841 microcontroller, or other microcontrollers in the 8051 family, this is the book for you. With introductory and advanced labs, you'll soon master the many ways to use a microcontroller. Perfect for academics!

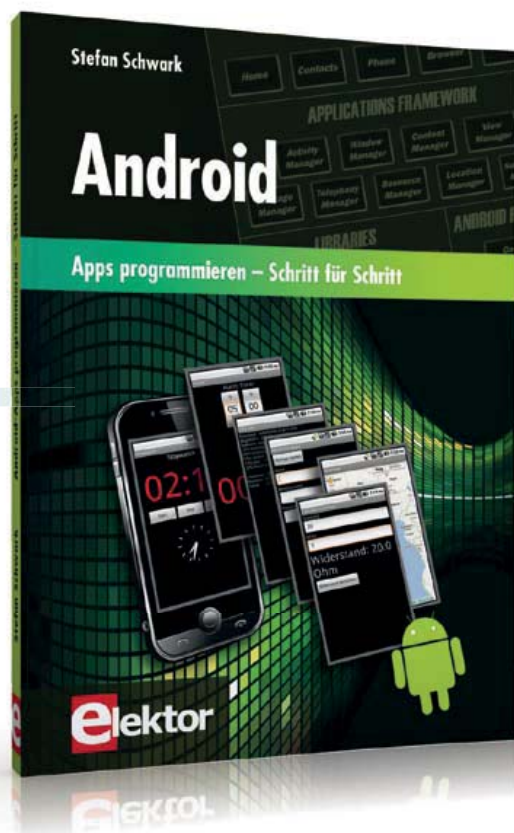


**Now
Just
\$35.00**



Buy it today!

www.cc-webshop.com



Apps programmieren – Schritt für Schritt

1 Android

Smartphones und Tablet-Computer mit dem Betriebssystem Android finden immer weitere Verbreitung. Die Anzahl der Anwendungsprogramme – die sogenannten Applikationen oder kurz Apps – mit denen sich die Geräte individuell an die Vorlieben und Wünsche ihrer Benutzer anpassen lassen, steigt täglich an. Man ist bei der Individualisierung seines Smartphones aber nicht auf fix und fertige Applikationen beschränkt. Es ist einfacher als man denkt, Android-Geräte selber zu programmieren und eigene Apps zu schreiben. Dieses Buch bietet eine Einführung in die Programmierung von Apps auf Android-Geräten. Es erklärt leicht nachvollziehbar die Funktionsweise des Android-Systems und Schritt für Schritt die Programmierung von Applikationen.

256 Seiten (kart.) • ISBN 978-3-89576-252-9
€ 34,80 • CHF 43,20

Kompletter Elektor-Jahrgang 2011 auf DVD

Elektor-DVD 2011

2 Die neue Elektor-Jahrgangs-DVD enthält alle Artikel des Jahrgangs 2011. Sie verfügt über eine sehr übersichtlich gestaltete Benutzeroberfläche. Mit der



Elektor-DVD 2011 können Sie Platinenlayouts in perfekter Qualität drucken; diese Layouts mit einem Zeichenprogramm verändern; die Schnellsuchfunktion benutzen, mit der Sie in den einzelnen Artikeln oder im ganzen Jahrgang nach Wörtern, Bauteilen oder Titeln suchen können; Schaltbilder, Platinenlayouts, Illustrationen, Fotos und Texte exportieren.

ISBN 978-90-5381-276-1 • € 27,50 • CHF 34,10

Theorie und Praxis mit WinFACT und Multisim

3 Regelungstechnik

Die heutige Regelungstechnik hat Verknüpfungspunkte mit fast jedem technischen Gebiet. Ihre Anwendungen reichen von der Elektrotechnik über die Antriebstechnik und den Maschinenbau bis hin zur Verfahrenstechnik. Will man nun die Regelungstechnik anhand der fachlichen Regeln dieser einzelnen Gebiete erklären, so müsste man von einem Regelungstechniker verlangen, jedes Fachgebiet, in dem er Regelungen vornehmen will, fundiert zu beherrschen. Dies ist aber bei dem heutigen Stand der Technik nicht möglich. Bei der Regelung einer Antriebsaufgabe, einer Druck- oder einer Temperaturregelung tauchen Gemeinsamkeiten auf, die man



mit einer einheitlichen Vorgehensweise beschreiben kann. Die Grundgesetze der Regelungstechnik gelten in gleicher Weise für alle Regelkreise, ganz unabhängig davon, wie verschieden sie im Einzelnen auch apparativ aufgebaut sein mögen. Dieses Buch richtet sich an den Praktiker, der gründlicher in die Regelungstechnik eindringen möchte, auf ausschweifende theoretische Exkursionen in die Mathematik aber gerne verzichten kann.

365 Seiten (kart.) • ISBN 978-3-89576-240-6
€ 49,00 • CHF 60,80

Bestücke und getestete Platine

4 Elektor-Linux-Board

Linux läuft heutzutage auf den unterschiedlichsten Geräten – sogar in Kaffeemaschinen. Es gibt daher viele Elektroniker, die an Linux als Basis für eigene Controller-Projekte interessiert sind. Eine Hürde ist jedoch die scheinbar hohe Komplexität, außerdem sind Entwicklungsboards oft recht teuer. Mit diesem kompakten Modul, das bereits für modernste Embedded-Projekte fertig bestückt ausgestattet ist, gelingt der Linux-Einstieg ideal und preiswert zugleich.

Art.-Nr. 120026-91
€ 64,95 • CHF 80,60



Band 1: 35 Einsteiger-Projekte in C

5 ARM-Mikrocontroller

Die Projekte in diesem Buch sind für Einsteiger in C und ARM-Mikrocontroller ausgelegt. Das heißt nicht, dass diese Projekte einfach sind. Sie sind aber einfach zu verstehen. Es wird beispielsweise die USB-Verbindung zur Kommunikation benutzt, eine Methode, die im mbed-Board so einfach integriert ist, dass sie sich auch für ein Einsteiger-Buch eignet. Der mbed NXP LPC1768 nutzt Cloud-Technologie, ein revolutionäres Konzept in der Software-Entwicklung. Es bedeutet, dass man keinerlei Software auf seinem PC installieren muss, um den mbed zu programmieren. Das Einzige, was Sie brauchen, ist ein Webbrowser mit Internetzugang und einen freien USB-Anschluss an Ihrem PC.

261 Seiten (kart.) • ISBN 978-3-89576-262-8

€ 39,80 • CHF 49,40

Von 0 und 1 zum FPGA

6 Digitale Logik selbst entwickeln

Dieses Buch nimmt Sie mit auf eine Entdeckungsreise in die Welt der digitalen Elektronik. Nach dem Aufbau einer soliden Wissensbasis hierüber verlagert

sich der Schwerpunkt zur programmierbaren Logik. Wie lassen sich mit vorhandenen Bausteinen logische Systeme aufbauen und wie koppelt man sie sicher und störungsfrei an die analoge Außenwelt? Das sind Fragen, die das Buch beantwortet.

603 Seiten (kart.) • ISBN 978-3-89576-254-3

€ 49,00 • CHF 60,80

Bausatz mit allen Bauteilen inklusive Platine(n)

7 TAPIR – E-Smog-Detektor

Der TAPIR (Totally Archaic but Practical Interceptor of Radiation) spürt "strahlendes Missverhalten" elektronischer Geräte in Ihrer Umgebung auf. Trotz der einfachen Schaltung handelt es sich hierbei um einen ultrasensitiven E-Smog-Detektor, der jede Quelle eines elektrischen oder – mit einer entsprechenden

Antenne versehen – magnetischen Feldes aufspürt und dies akustisch signalisiert.

Art.-Nr. 120354-71

€ 14,95 • CHF 18,60

Kreative Lösungen

aus allen Bereichen der Elektronik

8 311 Schaltungen

Das mittlerweile zwölfte Buch aus Elektors erfolgreicher „Dreihunderter“-Schaltungsreihe bietet ein weiteres Mal neue Konzepte sowie einen (fast) unerschöpflichen Fundus zu allen Bereichen der Elektronik: Audio & Video, Spiel & Hobby, Haus & Hof, Prozessor & Controller, Messen & Testen, PC & Peripherie, Stromversorgung & Ladetechnik sowie zu Themen, die sich nicht katalogisieren lassen.

544 Seiten (kart.) • ISBN 978-3-89576-255-0

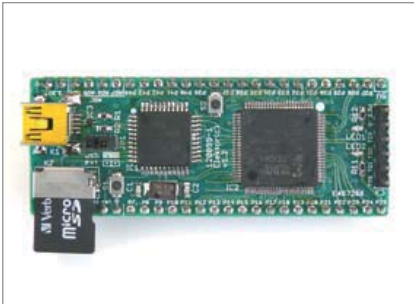
€ 36,80 • CHF 45,70

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

www.elektor.de

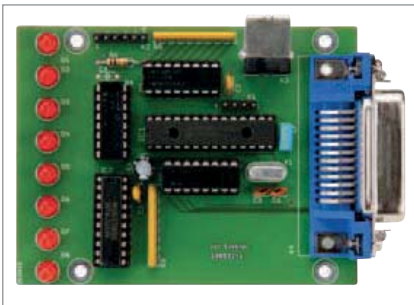
Elektor-Verlag GmbH
Süsterfeldstr. 25
52072 Aachen
Tel. +49 (0)241 88 909-0
Fax +49 (0)241 88 909-77
E-Mail: bestellung@elektor.de





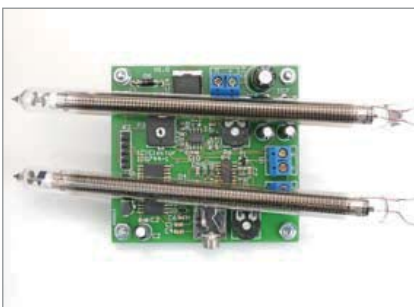
FPGA-Board

Wollten Sie nicht schon lange in die Welt der FPGAs vordringen, aber die Hürde des ersten Schritts war zu hoch? Das ist Vergangenheit, denn Elektor veröffentlicht ein FPGA-Entwicklungsboard, das nicht nur das Einarbeiten wesentlich erleichtert, sondern auch kostengünstig und vielseitig erweiterbar ist. Ein Mikrocontroller stellt über USB die Verbindung mit dem PC her, die Konfigurationsdaten des FPGAs werden auf einer microSD-Karte auf dem Board gespeichert. Ein Tool überträgt die Daten vom PC auf die microSD-Karte. Von dort werden sie in den FPGA geladen, sobald das Board gestartet oder rückgesetzt wird.



GPIB-nach-USB-Konverter

Der GPIB (General Purpose Interface Bus) ist ein 24-poliges Parallel-Interface, das 1984 für den Anschluss von Sensoren und programmierbaren Messgeräten an einen Computer eingeführt wurde. Verschiedene Anbieter verkaufen wiederaufbereitete und kalibrierte, hochwertige und robuste GPIB-Messgeräte von Herstellern wie HP, Tektronix und Marconi oft für weniger als 100 €. Im nächsten Heft stellen wir einen für rund 40 € realisierbaren Wandler vor, mit dem man solches GPIB-Equipment über USB an einen PC anschließen kann.



VU-Meter mit Nixies

Nach Uhren, Thermometern und einem Hygrometer mit Nixie-Anzeigeröhren haben wir für die nächste Elektor-Ausgabe ein VU-Meter vorbereitet, das zum Beispiel einen Röhren-Audioverstärker stilgerecht aufwerten kann. Dem Entwickler ging es weniger um absolut exakte Messungen von Ausgangsleistungen; an erster Stelle stand hier das trendige Design. Ein PSoC CY8C27443 von Cypress steuert zwei Nixieröhren des russischen Typs IN-9, ein Blickfang für das Auge des Betrachters!

Elektor November 2012 erscheint am 24. Oktober 2012.

Elektor gibt es im Bahnhofsbuchhandel, Elektronik-Fachhandel, an ausgewählten Kiosken und garantiert beim Presse-Fachhändler. Sie können Elektor auch direkt bei www.elektor.de bestellen.

Änderungen vorbehalten!

www.elektor.de www.elektor.de www.elektor.de www.elektor.de www.elektor.de www.elektor.de

Elektor im Netz

Alle Magazin-Artikel ab 1996 können einzeln im PDF-Format heruntergeladen werden (gegen Elektor-Credits). Zu jedem Artikel findet man eine kurze Zusammenfassung, technische Daten und eine Stückliste (sofern vorhanden), was bei der Projektauswahl hilft. Dazu kommen eventuelle Updates/Berichtigungen und die Downloads zum Artikel (z.B. Software und Zusatzinfos). Auch die zum Projekt gehörenden Produkte wie Platinen, programmierte Controller und mehr lassen sich hier bestellen.

Im Elektor-Shop findet man alle anderen Angebote von Elektor, wie CD-ROMs, DVDs, Kits, Module, Software und Bücher. Mit der Suchfunktion können Sie die ganze Site nach Stichwörtern durchforsten.

Ebenfalls unter www.elektor.de:

- News aus der Elektronikwelt
- Leserforum
- Download des E-Magazins
- Spezialangebote (zeitlich begrenzt)
- FAQs, Inhaltsverzeichnisse und Kontaktformular



SPECIAL: SAVE 50% SPECIAL: SAVE 50% SPECIAL: SAVE 50%

Celebrate Circuit Cellar's **25th Anniversary**



\$25 Print or Digital :: **\$50** Combo

Celebrate *Circuit Cellar's* 25th year of bringing readers insightful analysis of embedded electronics technology.

Visit www.circuitcellar.com/el912 to take advantage of these great deals.

BONUS OFFER! BONUS OFFER! BONUS OFFER! BONUS OFFER!

Sign up today and you'll also receive the **Special 25th Anniversary Edition** with your subscription!





- ✓ über 40 Jahre Erfahrung
- ✓ mehr als 40.000 Produkte auf Lager
- ✓ kein Mindermengenaufschlag
- ✓ schneller 24-Std.-Versand



+49 (0)4422 955-333



+49 (0)4422 955-360

**Professional quality
@ discount prices!**

Bauelemente
Werkstattbedarf
Netzwerk- & PC-Technik

Steckverbinder

Sat- und TV-Technik
Stromversorgung

RADIALL 

QR-Code mit
dem Smartphone
scannen oder
Kurzlink nutzen:



<http://rch.it/50>

Mit unserem neuen Premiumpartner bieten wir Ihnen Steckverbinderlösungen in den Bereichen BNC, SMA, SMB, QMA, SMP-MAX, MCX und MML ...

BNC-Steckverbinder

Radiall's erschwingliche „True“ BNC 75 Ohm Koaxial Steckverbinder Serie HDTV kann Datenraten von bis zu 3 Gbps oder höher verarbeiten.

- Steckkompatibel mit Standard 50 oder 75 Ohm BNC Produkten



R142 077 702

1,93

SMA-Steckverbinder

Die RADIALL Edelstahl SMA-Stecker sind für Anwendungen, bei denen Zuverlässigkeit, Langlebigkeit, Robustheit und hohe Frequenzen sehr wichtig sind, entworfen worden.



R125 072

3,99

SMB-Steckverbinder

Die SMB Snap-on-Subminiatur-Koaxial-Steckverbinder sorgen für eine schnelle und zuverlässige Verbindung bei hoher Packungsdichte für Anwendungen bis 4 GHz.



R114 238

3,11

QMA-Steckverbinder

QMA-Verbinder haben die gleiche hohe elektrische Leistungen wie die SMA-Serie mit einer einfacheren und schnelleren Montage. Die QMA-Serie ist eine kosteneffektive Lösung für die neue Generation von Basisstationen. Die QMA-Serie ist für DC bis 18 GHz ausgelegt.



R123 172

4,16

MCX-Steckverbinder

Die MCX-Serie ist für eine Optimierung Ihrer Applikation geeignet, da sie gut 30 % kleiner ist als die SMB-Serie. Die 50 Ohm Ausführungen eignen sich für Wireless-Anwendungen sowie für den Radio-Funkkommunikationsbereich.



R113A 182

1,60

MML-Steckverbinder

Die MML-Serie eignet sich insbesondere für die Anforderungen in den Bereichen

- Cell Relay
- WLAN Access Points
- GPS
- mobile Endgeräte



R302 302

0,29