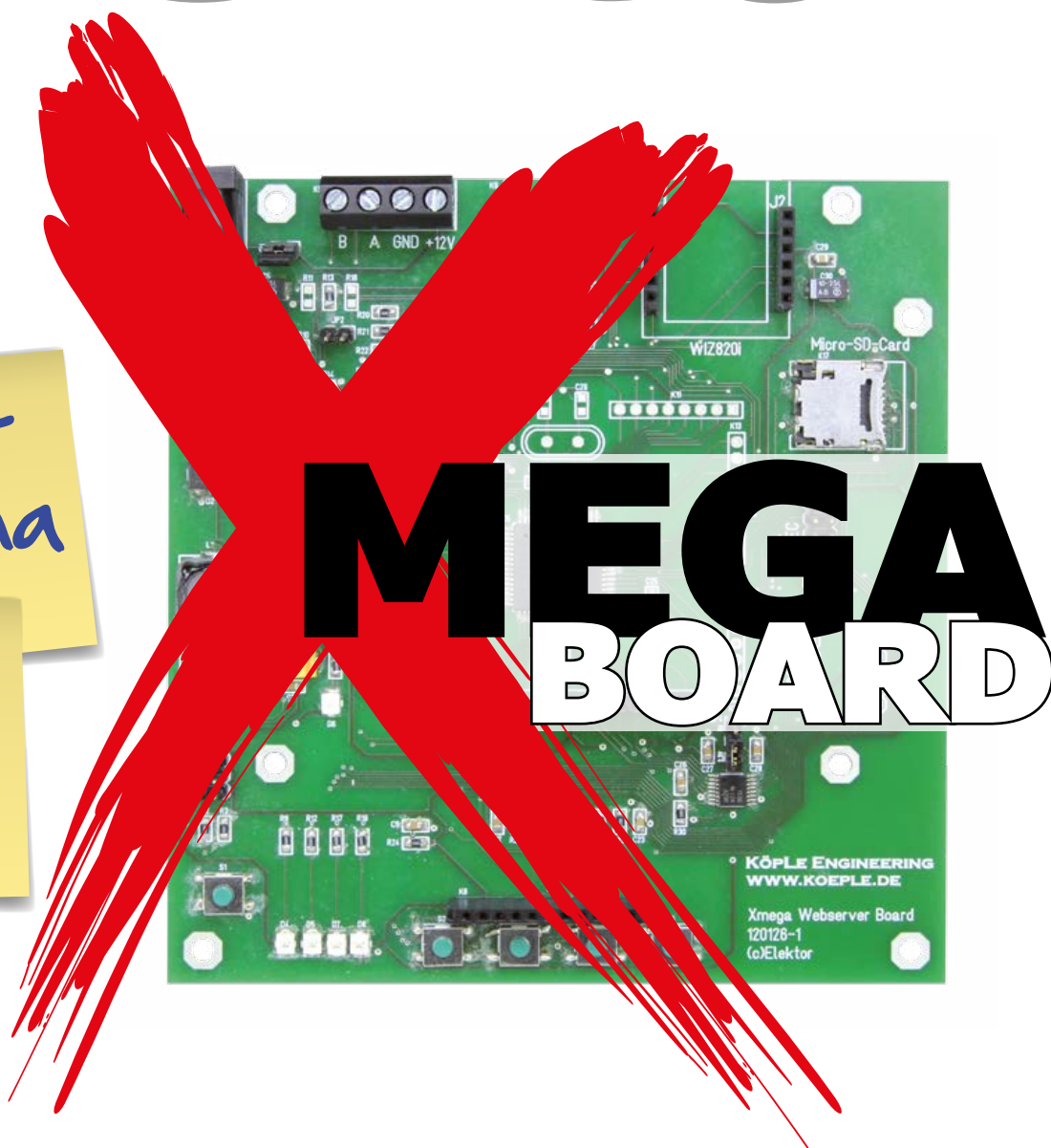


elektor



● **8 x 8 Dual-LED-Matrix** | Linux-Board mit LAN
Numitron-Arduino-Uhr | **Wasserpumpensteuerung**
Modulare Funkverbindung | Review: BeagleBone Black
Android-EKG ● **Neues aus dem Labor** ● **Retronik**



Entwickeln, Löten, Bestücken



Raspberry Pi Cam
HD-Foto- und
Videokamera

€ 28,30*

Raspberry Pi
Modell B 512 MB RAM



€ 38,95*

30 MHz, 2 CH Digital-
Speicheroszilloskop,
250 MSa/s



€ 470,00*

Big Beta-Reflow-Kit



€ 129,00*

ERSA Lötstation
i-CON NANO



€ 198,00*

*inkl. MwSt. und zzgl. Versandkosten

www.beta-eSTORE.com

Beta
LAYOUT
create : electronics

eSTORE[®] ist eine eingetragene Marke der Beta LAYOUT GmbH

PROTOTYPES

Beta LAYOUT

Entwerfen, Bestellen, Anfassen

3D-Druck online

mit neuester
Lasersinter-Technik



Ab sofort stellen wir alle 3D-Modelle
im Hi-Tech Lasersinterverfahren her.

Die Vorteile:

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

www.beta-prototypes.com

Beta
LAYOUT
create : electronics

THEY SAY THE FUTURE DOESN'T FIT IN A BOX.

Tuesday, January 7
through Friday, January 10, 2014
Las Vegas, Nevada • CESweb.org • [#CES2014](https://twitter.com/CES2014)

GOOD THING LAS VEGAS IS FAR FROM SQUARE.

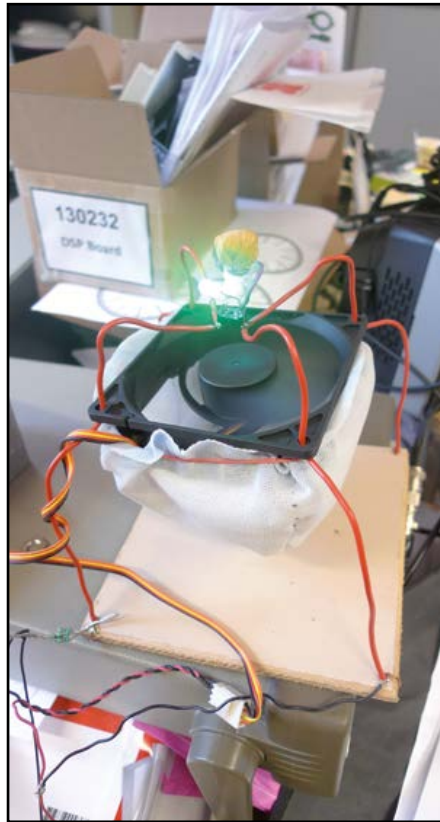
Over four days, those who shape the future gather in a city built on reinvention. Here, brands, markets and economies converge in what's far more than a tradeshow. And in 2014, there's more opportunity than ever to connect with those who matter. The only question now, why wouldn't you fit it in your future?

Register today at CESweb.org.



THE GLOBAL STAGE FOR INNOVATION

PRODUCED BY  CEA®



● Community

6 Impressum

10 Elektor World

Neues aus der Elektor-Community

● Industry

8 Aktuell

News / Korrektur

69 BeagleBone Black

Review des Raspberry-Pi-Rivalen.

● Labs

46 Starten mit dem LPC800 Mini-Kit

Das LPC800 Mini-Kit verfügt über einen 32-bit-ARM-Cortex-M0+-Controller in einem 8-poligen DIP-Ge-

häuse, einen Spannungsregler, zwei Tasten, eine LED und zwei kleine Prototyping-Bereiche. Wir geben etwas Starthilfe für angehende 32-bit-Controller-Entwickler.

50 DesignSpark: Ein einfaches Projekt

In diesem Teil unserer Serie zum kostenlosen CAD-Programm DesignSpark wird es praktisch: Am Beispiel eines Treibers für eine Zweifarben-LED kann man lernen, wie man die Editoren für Schaltplan und Platinen-Layout benutzt

● Projects

12 Xmega-Multifunktions-Board

Das hier vorgestellte Mikrocontroller-Board bietet sich vor allem für das Messen, Steuern und Regeln an. Dank eines aufsteckbaren TCP/IP-Moduls lassen sich Webserver-

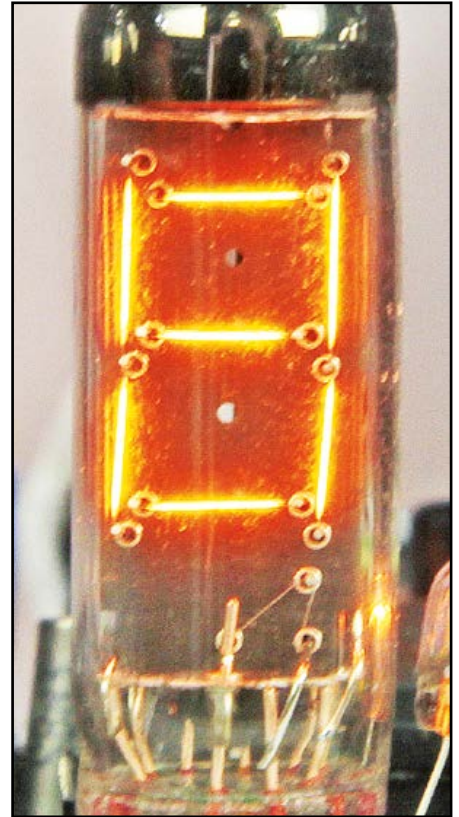
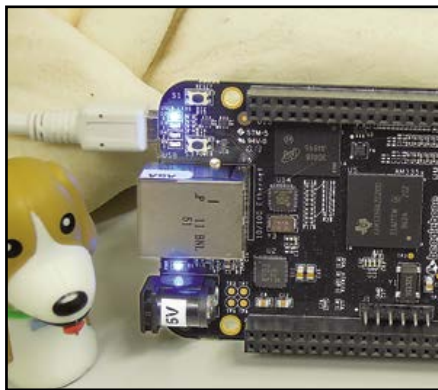
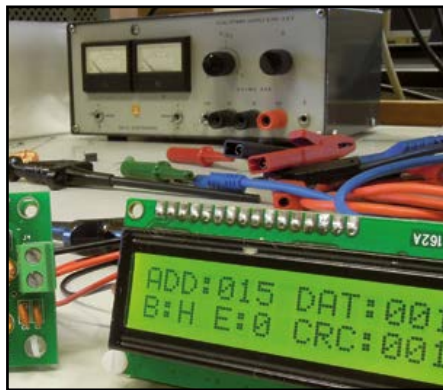
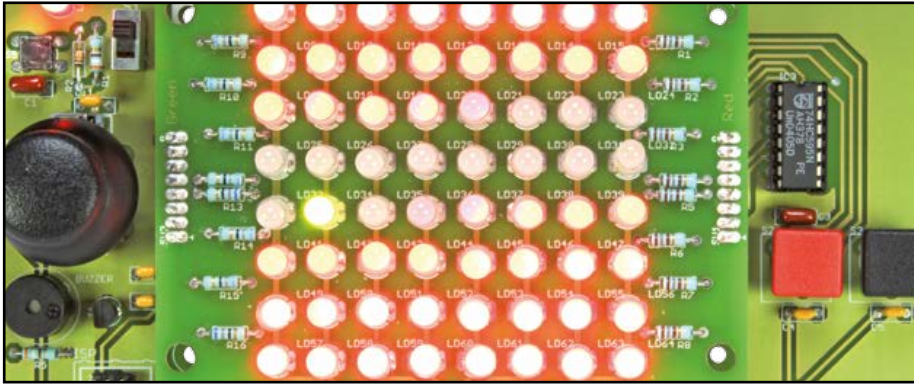
und andere Netzwerk-Anwendungen realisieren; eine Mikro-SD-Karte dient als Massenspeicher. Zur Bedienung stehen vier LEDs, vier Taster und ein (abnehmbares) Display zur Verfügung. Und auch mit Schnittstellen ist das Board reichlich ausgestattet.

24 Elektor-Linux-Board: Hier ist Version 2!

Seit rund eineinhalb Jahren kann man bei Elektor ein preiswertes und kompaktes Linux-Board kaufen – begleitet von einer Artikel-Serie, mit der auch Anfängern der Einstieg in die „Embedded Linux“-Welt gelingt. Nun ist es Zeit für eine Neuauflage des Boards.

28 8x8-Duo-LED-Matrix

Dieses Projekt nutzt eine elegante Methode, mit der eine Matrix sehr vieler Rot-Grün-LEDs mit nur einigen I/O-Leitungen eines Mikrocontrollers angesteuert werden kann. Auf dem Board ist auch ein Joystick



vorhanden, so dass sich kleine Spiele verwirklichen lassen.

36 Numitron-Uhr/Thermometer

Bei diesem Projekt wird Arduino aus seiner gewohnten Umgebung von Shields und Breadboards geholt und mit einem Elektor-typischen DIY-Projekt verbunden. Wir vereinen moderne Mikrocontroller-Technik mit Anzeige-Röhren aus der Sowjetunion der 50er Jahre, die aber auch heute noch ein attraktiver Blickfang sind.

42 Wasserpumpensteuerung

Die Wasserinstallation an Bord einer Yacht besteht aus Wasserleitungen plus einer mit 12 V betriebenen Pumpe, die gut und gerne 5...10 A Strom zieht. Etwas Elektronik sorgt dafür, dass die Pumpe genau passend zum Wasserbedarf angesteuert wird.

54 Modulare Funkverbindung mit Manchester-Code (2)

Im ersten Teil ging es um die Hard-

ware, jetzt kommt die Software an die Reihe. Wo gute Hardware und ein passendes Platinen-Layout wichtig für die korrekte Aussendung und den Empfang der HF-Signale ist, steht die Software bzw. Firmware für die Zuverlässigkeit der Datenübertragung mit Hilfe dieser Signale.

60 Elektor Android EKG (3)

Hier kommt der letzte Teil der kleinen Serie zu diesem interessanten Projekt. Schon ungeduldig erwartet von unseren Lesern wurden die Stückliste und Tipps zum Aufbau. Hinweise zur Inbetriebnahme und zur korrekten Einstellung komplettieren das Ganze.

72 D/A-Wandlung mit variabler Spannung

Sobald die erforderliche Auflösung 8 Bits übersteigt, werden die Wandler zwischen der analogen und der digitalen Welt deutlich teurer. Mit dieser Schaltung zeigen wir, wie

man ohne zusätzliche Kosten die Auflösung der gängigen preiswerten D/A-Wandler verbessern kann.

● Magazine

74 Retronik

Radiometer PHM22 / PHA928a

78 Hexadoku

Sudoku für Elektroniker

82 Vorschau

Nächsten Monat in Elektor

Aus technischen Gründen musste das Projekt „Windmesser“ auf die nächste Ausgabe verschoben werden.

Impressum

44. Jahrgang, Nr. 514 Oktober 2013
Erscheinungsweise: 10 x jährlich
(inkl. Doppelhefte Januar/Februar und Juli/August)

Verlag

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

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

Hauptsitz des Verlags

Elektor International Media
Allee 1
NL-6141 AV Limbricht

Anzeigen (verantwortlich):

Irmgard Ditgens
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. 43 ab 01.01.2013

Distribution:

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

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

© 2013 elektor international media b.v.

Druck: Senefelder Misset, Doetinchem (NL)

ISSN 0932-5468

8 bit (X)Mega, 32 bit Mini

In diesem Heft ist es endlich soweit, wir stellen unser Xmega-Board vor. Voraus gingen etliche Stunden an Design-Überlegungen; die erste Version haben wir noch einmal grundlegend überarbeitet. Den Löwenanteil der Entwicklungsarbeit – nämlich den Entwurf von Schaltplan und Platine – haben die Jungs von KöpLe Engineering übernommen, und ich finde, das Ergebnis kann sich sehen lassen. Doch zur Hardware gehört auch Software – schon mit dem ersten Prototypen habe ich unzählige Stunden zugebracht. Und das gerne auch am Wochenende, denn die Entwicklung mit dem Board macht wirklich Spaß. An dieser Stelle will ich noch das Display erwähnen, das KöpLe vorgeschlagen hat. Es zeigt auch sich schnell ändernde Werte gut ablesbar an und rechtfertigt so den höheren Preis gegenüber einem Standard-Display vollauf. Bei der zweiten Version des Boards mussten wir zur Ansteuerung der Peripherieblöcke teilweise andere Controllerpins verwenden. Und die LEDs leuchten nun bei einem High des jeweiligen Pins, was gebräuchlicher ist. Aber mit der Embedded Firmware Library dauerte es nur ein paar Minuten, bis sämtliche Software, die ich bisher für das Board erstellt hatte, an die neue Verdrahtung angepasst war. Mehr über das Xmega-Board lesen Sie ab Seite 12. Mikrocontrollerfans sollten sich auch den Artikel auf Seite 46 anschauen. Mein Kollege Clemens beschreibt Schritt für Schritt, wie man den LPC800 zum Leben erweckt. Das ist ein ARM-kompatibler 32-Bitter im DIL-Gehäuse, den der Hersteller zu einem wahren Kampfpfeis anbietet. Damit soll er Low-Cost-8-bit-Controllern Konkurrenz machen. Ich persönlich finde 8 Pins und einen Flashspeicher von 4 KB ziemlich limitiert, der Verzicht auf mehr lohnt sich erst bei größeren Stückzahlen. Und das Konzept funktioniert nur, wenn auch das Entwickeln ähnlich schnell wie bei einem 8-bit-Controller geht. Aber probieren Sie das doch selbst einmal aus!

Jens Nickel



Unser Team

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



Germany

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



United Kingdom

Wisse Hettinga
+31 46 4389428
w.hettinga@elektor.com



Netherlands

Harry Baggen
+31 46 4389429
h.baggen@elektor.nl



France

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



USA

Hugo Van haecke
+1 860-289-0800
h.vanhaecke@elektor.com



Spain

Eduardo Corral
+34 91 101 93 95
e.corral@elektor.es



Italy

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



Sweden

Wisse Hettinga
+31 46 4389428
w.hettinga@elektor.com



Brazil

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



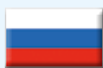
Portugal

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



India

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



Russia

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



Turkey

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



South Africa

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



China

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

Unser Netzwerk



VOICE COIL



audioXpress



Die Elektor-Community



Unsere Partner und Sponsoren



Batronix

www.batronix.com/go/27 23



Becker & Müller

www.becker-mueller.de 9



Beta Layout

www.beta-prototypes.com 2



CadSoft

www.cadsoft.de 27



CES International 2014

www.CESweb.org 3



Intelligent Soc s.l.

www.socutions.com 35



LeitOn

www.leiton.de 35



LinX Technologies

www.linxtechnologies.com 49



Pico

www.picotech.com/PS245 35



Reichelt

www.reichelt.de 84



Schaeffer AG

www.schaeffer-ag.de 45



WIZnet Europe

www.wiznet.eu 45

Sie möchten Partner werden?

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



elektorLive!

Am 12. Oktober 2013 ist es so weit: Der zweite Seminar- und Ausstellungstag „ElektorLive!“ findet in Hanau bei Frankfurt am Main statt. Aus dem attraktiven Programm können sich alle Teilnehmer wieder zwei Seminare herausuchen:

- Android-Grundlagen (eigene Apps entwickeln)
- LabVIEW für den Praktiker (Anschluss eigener Elektronik)
- Rapid SMD-Prototyping (PCB-Pool und Target 3001!)
- Röhrenverstärker (mit Menno van der Veen)
- AVR-Mikrocontroller (Tipps und Tricks)
- Schaltplan- und Layout-Design mit EAGLE v6
- LPKF ProConduct (Durchkontaktierungen ohne Chemie)
- Raspberry-Pi-Grundlagen (Einsatz in eigenen Projekten)
- Sensoren ans Internet (Entwickeln mit TCP/IP-Modulen)
- Arduino-Praxisprojekte (Programmieren und eigene Hardware)

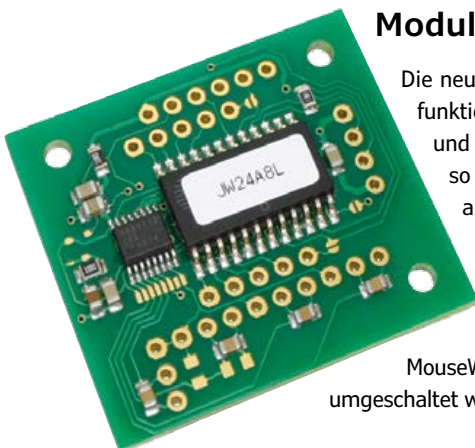
- ARM Cortex-M0-Mikrocontroller (Einstieg und Programmierung)
- FPGA-Erweiterungsboard (Für das Elektor FPGA-Board)
- Anwendungen mit Raspberry Pi und dem Elektor-Linux-Board

Auf der begleitenden Ausstellung kann man sich über neue Elektronik-Trends informieren. Bereits zugesagt haben Cadsoft („Eagle“), Eurocircuits, Arduino, Embedded Projects (Linux-Boards), LPKF, Beta-Layout, Tinkerforge, Segger Electronics, WizNet (Netzwerk-Module), RepRap (3D-Drucker), Ingenieurbüro Friedrich („Target3001!“), Matrix Multimedia („E-Blocks“) und die Berufsschule Leonberg.

Natürlich wird auch Elektor mit einem größeren Stand vertreten sein. Hier kann man mit Entwicklern aus dem Elektor-Labor fachsimpeln und sich neue Module und Boards anschauen.

Der Eintrittspreis beträgt für Abo-Mitglieder 29,50 Euro (49,50 Euro für Nicht-Mitglieder), Studenten, Azubis und Schüler zahlen nur 19,50 Euro.

www.elektor-live.de



Module für Joysticks und Mauseinsatz

Die neuen Module von Code Mercenaries enthalten die komplette Elektronik, um Joystick- oder Mausfunktionen am USB zu realisieren. Basierend auf den Bausteinen JoyWarrior24A8L, JoyWarrior24A10L und MouseWarrior24H8 können die Module Eingangssignale von 0 bis 5 V verarbeiten; sie erlauben so einen Anschluss der meisten Joysticktypen, oder auch die Erfassung von Messwerten anderer analoger Sensoren.

JoyWarrior24A8L-MOD bietet vier analoge Achsen mit jeweils 8 bit Auflösung, JoyWarrior24A10L-MOD drei Achsen mit jeweils 10 bit. Beide Module unterstützen jeweils 8 direkt oder 16 in einer 4x4-Matrix angeschlossene Tasten und haben vier Ausgänge, die z.B. zum Ansteuern von LEDs für die Anzeige von Betriebszuständen verwendet werden können.

MouseWarrior24H8-MOD ist ein Maus/Joystick-Hybrid, der jederzeit zwischen Maus- und Joystickbetrieb umgeschaltet werden kann. Er unterstützt drei Achsen mit jeweils 8 bit plus sechs Tasten, die Mausfunktion hat

eine abschaltbare Autozentrierung.

Für den Betrieb der Module ist keine Installation spezieller Treiber notwendig, da sie als Joystick bzw. Maus arbeiten und Systemtreiber verwenden. Die Module sind direkt einsatzbereit, USB, Achsen und Tasten können an den Lötäugen auf den Modulen angeschlossen werden.

Die Module sind unter den Bezeichnungen JW24A8L-MOD, JW24A10L-MOD, sowie MW24H8-MOD bei Code Mercenaries und bei Distributoren erhältlich.

www.codemercs.de

Raspberry Pi Projects

Die Elektronik-Plattform Element14 stellt mit „Raspberry Pi Projects“ ein neues Online-Portal vor, das Raspberry-Pi-Besitzern einen zentralen Anlaufpunkt zur Präsentation von Projekten und für die Weiterbildung bietet. Für (werdende) RaspPi-Entwickler gibt es außerdem ein „8 GB Raspberry Pi Paket“. Das neue Paket umfasst einen Raspberry Pi Typ B und



eine 8 GB fassende SD-Karte mit allen sechs offiziell von der Raspberry Pi Foundation freigegebenen Betriebssystemen. Die SD-Karte stellt außerdem Platz zur Verfügung, um weitere Raspberry-Pi-Projekte zu erstellen und zu speichern.

Für Anfänger sind online einfach verständliche Tutorials verfügbar, die durch den Aufbau des Raspberry Pis führen. Fortgeschrittene Anwender unterstützt

die Bibliothek bei der Realisierung ihrer Ideen und dem Erlangen der notwendigen Qualifikation für komplexere Projekte.

Mitglieder der element14-Community dürfen gerne zur Bibliothek beitragen und ihre eigenen Projekte in den Katalog eintragen sowie die Arbeiten anderer bewerten. Zu den ersten vorgestellten Projekten gehören:

- Eine Schritt-für-Schritt-Anleitung für Anfänger, die ausführlich darstellt, wie Raspberry Pi in einen vollständig funktionierenden Laptop verwandelt wird.
- Eine bebilderte Anleitung für fortgeschrittene Anwender, die erläutert, wie Raspberry Pi als Durchflussmesser für Wasser verwendet werden kann.
- Ein software-orientierter Leitfaden erklärt, wie sich eine Pumpkin-Pi-Kürbislaterne mit über das Internet gesteuerter Beleuchtung realisieren lässt.

www.element14.com/community/community/raspberrypi_projects

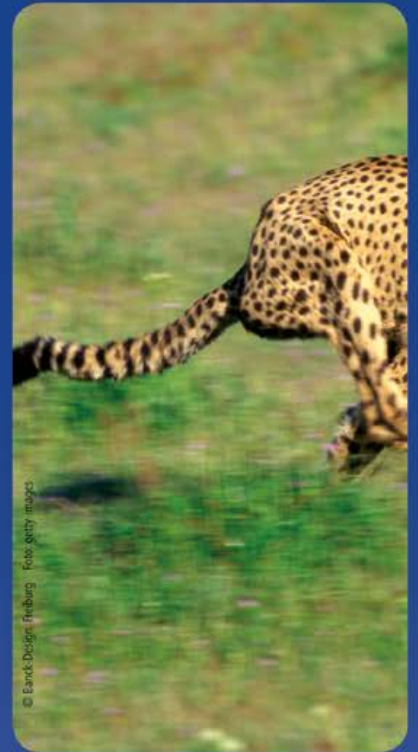
Updates und Korrekturen

Smartphone als Fernbedienung

Elektor 7-8/2013, S. 28 (120043)

Die Orientierung bei der Bestückung des Bluetooth-Moduls ist im Artikel falsch beschrieben. Fälschlicherweise steht dort: „Pin 1 ist dort, wo ein kleiner runder Punkt im Metall der Abschirmung (an der Seite, wo die Drahtantenne sitzt) erkennbar ist. Dieser Punkt sollte zu PC1 auf der Platine weisen.“

Richtig ist: Pin 1 ist im Bestückungsplan/Platinenaufdruck mit einem weißen Punkt innerhalb des Moduls markiert. Der kleine runde Punkt oder Kreis im Metall der Abschirmung muss zu PC1 weisen.



WOW! Schon da!

**punktgenau
plangenau
preisgenau!**

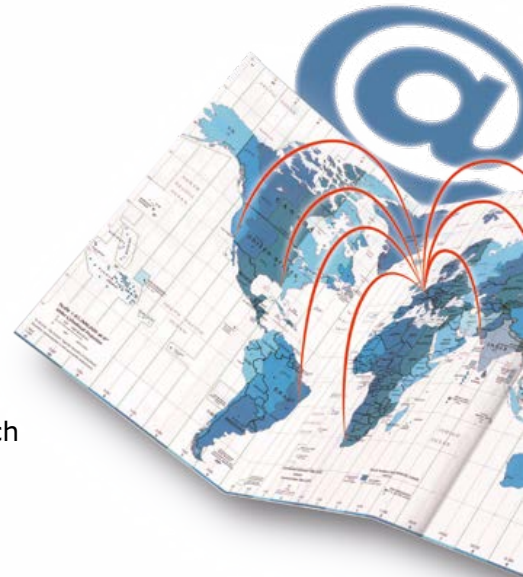
**Die schnellen
Leiterplatten-
Spezialisten
mit über
25 Jahren Erfahrung!**

**BECKER
MÜLLER** 
www.becker-mueller.de
Mit Online-Kalkulator!

Elektor World

Zusammengestellt von
Wisse Hettinga

Jeden Tag, jede Stunde, jede Minute, ja, in jedem Augenblick schaffen, optimieren, rekonstruieren und entwickeln Ingenieure und Enthusiasten neue Elektronik. Hauptsächlich zum Spaß, aber manchmal wird aus Spaß (beruflicher) Ernst...

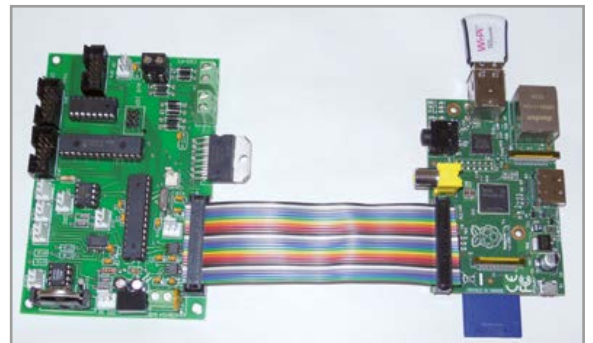


Iso-Pi Board

Als Brian Millier, Mitarbeiter des *Circuit Cellar*, Ende 2012 seinen ersten Raspberry Pi erhielt, begann er ein Projekt, das Grundlage seines zweiteiligen Artikels „Raspberry Pi I/O Board“ wurde und in CC August und September 2013 erschien. Millier, ein ehemaliger Messtechniker an der Dalhousie University in Halifax, Kanada, stellt eine eigene vielseitige I/O-Platine für den Single-Board-Computer vor. „Zu der Zeit, als ich meinen Raspberry Pi erhielt, hat einer der Entwickler des RPi ein I/O-Board namens Gertboard entworfen“, sagt Millier. „Ich glaube, meine Schaltung unterscheidet sich davon und weist auch einige Vorteile gegenüber dem Gertboard auf.“

Zum Beispiel, so Millier, „bietet es eine vollständige galvanische Trennung zwischen allen I/O-Devices und dem Raspberry Pi selbst.“ Sie können in der Septemбераusgabe des *Circuit Cellar* herausfinden, wie man Milliers Iso-Pi konfiguriert.

Ein kurzes Flachbandkabel verbindet den Raspberry Pi (rechts) und Milliers Iso-Pi-I/O-Board.



Xpressly für Audio



Die Historie von Audioelektronik in Elektor tritt mit dem Relaunch von *audioXpress* in eine neue Ära. Im Jahr 2011 erwarb Elektor *audioXpress*, *Voice Coil*, *Loudspeaker Industry Sourcebook*, *World Tube Directory*, ausgewählte Audio-Bücher und vieles mehr. In der Überzeugung, dass die Arbeit von Audio-Enthusiasten als Modell für die Industrie dienen könne, qualitativ hoch- und höchstwertige Produkte auf den Markt zu bringen, wurden diese Zeitschriften vor mehr als 35 Jahren in den USA von Edward T. Dell (1923-2013) gegründet. In dieser Zeit waren diese Titel nicht nur den DIY-Audiofans, sondern auch den Audioentwicklern der Industrie eine Quelle großartiger Artikel, Projekte, Tipps und Technologien. Ein neues Redaktionsteam von ausgewählten Autoren aus dem Elektor-Netzwerk arbeitet derzeit an einem Redesign der Titel in einem erweiterten Format. Die neue *AudioXpress* erreicht die globale Audioentwickler-Community und nicht zu vergessen die Forschungs- und Entwicklungsabteilungen der Industrie in vielen neuen Applikationsbereichen. „audioXpress restyled“ wird auf der kommenden AES Convention in New York (20.12.2013) vorgestellt, ein neues grafisches Layout sowohl für die Printausgabe als auch für alle digitalen Formate, einschließlich eines regelmäßigen Newsletters für (derzeit) über 30.000 Mitglieder. *AudioXpress* engagiert sich und diskutiert auch bei Twitter (@audioXP_editor) und Facebook (facebook.com/audioXpresscommunity) mit der globalen Audiogemeinde.

www.audioXpress.com



Der nächste Schritt: Arduino.next

Sicherlich stellt Arduino für viele junge Studenten eines der wichtigsten Gateways zur technologischen Welt dar. Ein Schlüssel zum Erfolg ist zweifellos die einfache Erstellung einer Applikation in wenigen Minuten. Sie brauchen dazu kein tiefgreifendes Wissen über Elektronik oder Programmierung. Fast alles steht im Internet - einfach etwas Code kopieren und einfügen, eine einfache Schaltung reproduzieren und schon ist die Anwendung startklar. Aber was ist, wenn Sie ein wenig tiefgründiger forschen wollen? Wie können Sie das Verhalten Ihrer Anwendung ändern? Was können Sie tun, damit sie in einer anderen Art und Weise arbeitet? Welche Schaltung brauchen Sie, um eine neue Funktion zu ermöglichen?

Wir haben uns all diese Fragen auch gestellt und die richtigen Antworten gegeben, einfach und leicht zu verstehen, um Ihnen zu helfen, den nächsten Schritt auf Ihrer Lieblings-Embedded-Plattform zu machen. Dieser Schritt heißt Arduino.next und erscheint demnächst im Netz - powered by Elektor.

Bleiben Sie dran an unseren Kommunikationskanälen!

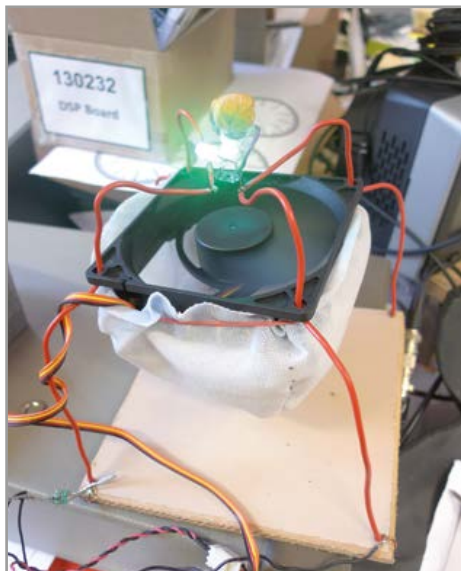
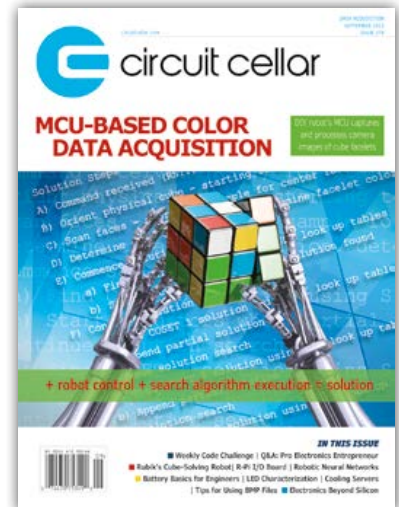
Folgen Sie uns auf Facebook, www.facebook.com/arduino.next und auf Twitter (@arduino.next)!



Circuit Cellar

Mit der September-Ausgabe 2013 enthüllt die Zeitschrift *Circuit Cellar* ein neues, frisches Layout und neue Inhalte für Elektrotechnik-Ingenieure, Akademiker und Embedded-Systemdesigner. Zusammen mit der Mannschaft von Elektor International Media liefert das CC-Team ein modernes, sauberes Layout, das das Studium und die Analyse von Fotografien und Schaltplänen einfacher und spannender gestaltet. Eingebaut in das Layout sind praktische direkte Links (auch QR-Codes) zu einer Vielzahl von Online-Ressourcen wie Quellcodes, Videos und Stücklisten.

Als neue Inhalte stellt CC zwei informative Kolumnen vor: Green Computing von Ayse Coskun (Ausgabe 278) und Programmable Logic in Practice von Colin O'Flynn (Ausgabe 279). Ein weiteres neues Feature ist CC World (S. 8). Analog zur Elektor Welt berichtet CC damit monatlich über interessante Themen wie die *CC Weekly Code Challenge* (<http://bit.ly/1brGEIU>). Das Team hofft, Sie von der erneuerten *Circuit Cellar* zu überzeugen. Um Artikel und Projektideen einzureichen, schreiben Sie einfach an editor@circuitcellar.com.



Tierfreundliche Mückenfalle

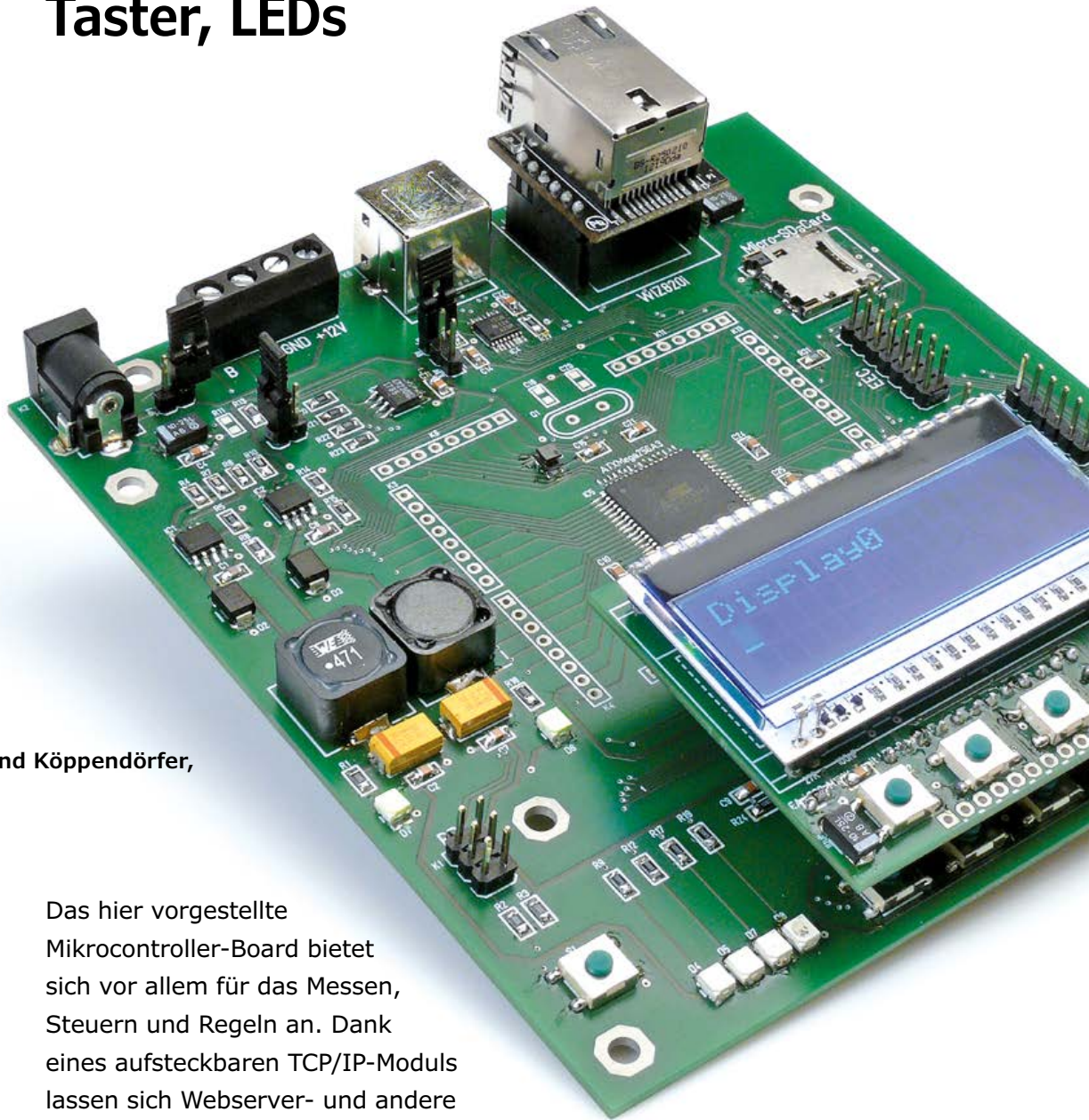
Nach den Überschwemmungen des Frühjahrs ist eine wahre Mückenplage über unser Land hereingebrochen. Für viele war der August ein wahrer Kampfmonat gegen Mücken, Gelsen, Schnaken, Bremsen oder wie das @#\$%'-s-Gezücht auch immer heißt, das hinter unserem Blut her ist.

Aurélien Moulin, unser Elektor-Labs-Auszubildender aus Frankreich, ist immer auf der Suche nach neuen Projekten. Er hat einen „ultimativen Mücken-Killer“ vorgeschlagen, der aus einer LED und einem alten Computer-Lüfter besteht. Als das Gelächter und die blöden Kommentare der versammelten Elektor-Crew abgeebbt waren, haben wir ihn gefragt, ob er seine Idee schon mal in der Praxis ausprobiert hat. Seine Antwort war schlechtweg „Yes“ (unsere französischen Auszubildenden sind immer todernst). Die Idee ist einfach: Die LEDs ziehen die Mücken an und der Ventilator saugt sie nach unten in ein kleines Netz. Aurélien bastelte einen einfachen Prototyp zusammen und zu unserer großen Überraschung landeten rund 120 Mücken (in Worten: hundertzwanzig!) in der Falle. Am ersten Abend!

Aber jetzt die 1-Million-Dollar-Frage! Alle Mücken im Netz waren zwar gefangen, aber noch am Leben. Frage an Sie: Wie konnten die Mücken die tödliche Geschwindigkeit der rotierenden Ventilatorblätter überleben?

Xmega-Multifunktions-Board

Display, SD-Karte, Netzwerk, RS485, Taster, LEDs



Von **Jens Nickel**

Entwicklung:

Achim Lengl und **Bernd Köppendörfer**,
KöPLe Engineering

Das hier vorgestellte Mikrocontroller-Board bietet sich vor allem für das Messen, Steuern und Regeln an. Dank eines aufsteckbaren TCP/IP-Moduls lassen sich Webserver- und andere Netzwerk-Anwendungen realisieren; eine Mikro-SD-Karte dient als Massenspeicher. Zur Bedienung stehen vier LEDs, vier Taster und ein (abnehmbares) Display zur Verfügung. Und auch mit Schnittstellen ist das Board reichlich ausgestattet.

Nicht erst seit der großen Resonanz auf die ElektorBus-Serie wissen wir, dass das Messen, Steuern und Regeln bei unseren Lesern hoch im Kurs steht. Natürlich kann man einen PC als Schaltzentrale einsetzen, doch dieser dürfte für viele Aufgaben zu groß, zu teuer und zu laut sein. Für andere Projekte ist das Elektor-Linux-Board eine gute Wahl, doch nicht jeder hat sich schon mit dem freien Betriebssystem angefreundet. Für viele Jobs völlig ausreichend ist ein 8-bit-Controller, zum Beispiel aus der beliebten AVR-Reihe.

Daneben benötigt man Schnittstellen, mit denen man sich auch über größere Entfernungen mit Sensoren und Aktoren „unterhalten“ kann. Eine gute Wahl sind hier RS485 und natürlich ein Netzwerkzugang. Sehr wertvoll ist auch ein SD-Karten-Anschluss für das Speichern von Messwerten. Für die Ein- und Ausgabe (und eventuell auch die Realisierung eines Menüs) gehören schließlich noch ein Text-Display, ein paar sinnvoll angeordnete Taster und LEDs auf das Board.

Xmega als Basis

Anhand dieser kleinen Liste haben wir das vorliegende Board konzipiert; **Bild 1** zeigt ein Blockdiagramm. Das Ingenieurbüro KöpLe [1] brachte weitere Ideen ein und übernahm das Design von Schaltung und Platine. Das Endergebnis ist über Elektor als fertig bestücktes und getestetes Board und als Platine erhältlich [2], siehe **Bild 2**.

Beim Controller haben wir auf viel Flash geachtet, wobei wir hier vor allem Webserver-Anwendungen im Auge hatten. Wir entschieden uns gegen den beliebten ATmega2560 (der zum Beispiel auf den großen Arduino-Boards verbaut ist) und für einen ATXmega256A3. Dieser besitzt 256 kB Flash und 16 kB RAM, bringt aber noch ein paar schöne neue Features wie zum Beispiel das fortgeschrittene Event-System mit [3]. Es ist damit zum Beispiel auf einfache Weise möglich, Level-Änderungen an jedem der GPIO-Pins zu zählen. Für komplexere Anwendungen sind auch Interrupts gestufter Priorität ein nützliches Feature. Am Anfang etwas nachteilig scheint es, dass die Register nicht mehr weitgehend kompatibel zu den traditionellen ATmegas sind. Um

Eigenschaften

- ATXmega256A3 mit 256 kB Flash und 16 kB SRAM
- 4 Taster und 4 LEDs
- Display-Modul aufsteckbar, Textdisplay (3x16) mit LED-Backlight
- RS485-Treiber, Schraubklemmen für A, B, 12 V und GND (ElektorBus)
- Stiftleiste zum Anschluss von USB/TTL-Kabel von FTDI bestückbar
- Stiftleiste für USB/TTL-Konverter BOB bestückbar
- weitere UART-Pins nach außen geführt, Mini-DIN-Buchse bestückbar
- Zugang zu fast allen Controllerpins über (optionale) Stiftleisten
- Scharnier für Mikro-SD-Karte, Ansteuerung über SPI
- Sockel für Netzwerk-Modul WIZ820io, über Elektor erhältlich (130076-91)
- Embedded Extension Connector mit 3x ADC, 2x GPIO, SPI, I2C auf 2x7-Stiftleiste, diverse Erweiterungsplatinen über Elektor erhältlich
- Platine passend für Gehäuse Hammond 1598REGY und RS 220-995
- Programmierbar mit günstigem AVRISP-Programmer und kostenlosem Atmel Studio
- Softwarebibliothek in C für alle Peripherie-Einheiten kostenlos zum Download

diverse Stündchen Datenblatt-Studium kommen daher auch erfahrene AVR-Programmierer nicht herum. Schneller zu eigenem Code kommt man mit den UART-, SPI- und weiteren Treibern des Herstellers, die man ebenso wie die Entwick-

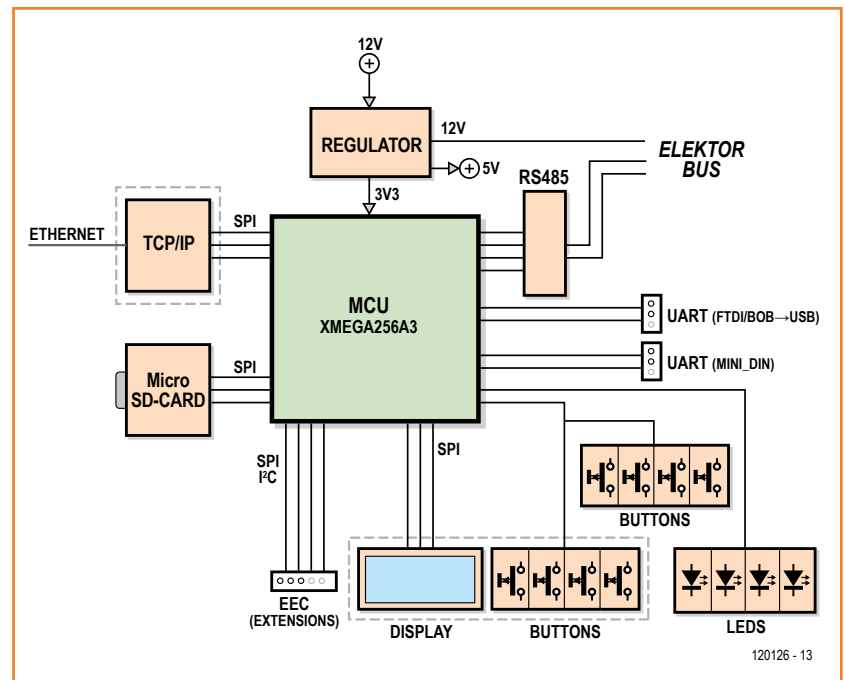


Bild 1. Blockdiagramm des Elektor-Xmega-Boards.

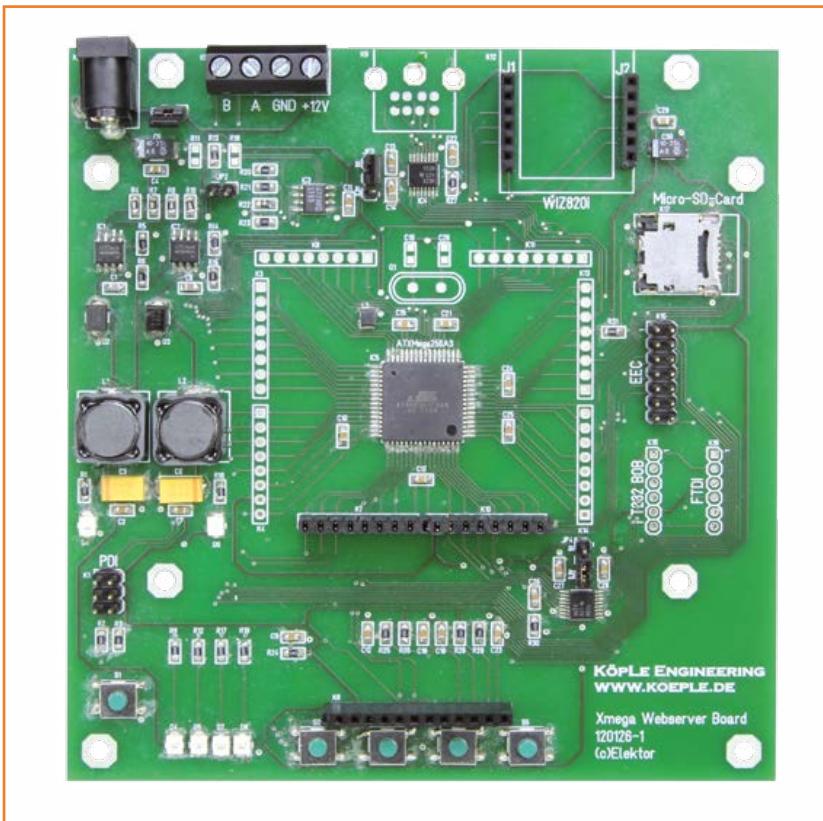


Bild 2.
Das Elektor-Xmega-Board ist bestückt und getestet erhältlich. Es kann noch mit Zusatzmodulen und Steckverbindern aufgerüstet werden.

lungsumgebung Atmel Studio 6 kostenlos aus dem Internet herunterladen kann. Doch wir bei Elektor machen das Ganze noch einfacher: Mit einem API (Programmier-Interface) für den Controller und einem Treiber-File für das ganze Board, das alle Peripherieeinheiten abdeckt. Mehr dazu siehe unten.

Stromversorgung

Der Xmega benötigt eine Betriebsspannung von 3,3 V, was ein wenig der Trend in der Elektronikwelt ist. Es war dennoch von Anfang an klar, dass wir mit 3,3 V und 5 V zwei Spannungen auf dem Board bereitstellen werden, allein schon für die Kommunikation mit herkömmlicher 5-V-Elektronik. Die Stromversorgung rund um den energiesparenden Schaltregler MC34063A [4] ist daher doppelt ausgeführt, siehe den Schaltplan in **Bild 3**. Zur Dimensionierung der Beschaltung gibt es übrigens ein hilfreiches Tool im Internet [5]. Eingangsseitig führen wir den beiden ICs jeweils 12 V zu. Diese Spannung lässt sich entweder über Schraubklemmen (siehe unten) oder eine Klinkenbuchse (Polung beachten!) extern einspeisen, was man mit JP1 auswählt.

K1 ist der sechspolige PDI-Steckverbinder zur

In-Circuit-Programmierung des Controllers. Die Steckerbelegung ist etwas anders als beim bekannten ISP-Anschluss für die ATmega-Serie, dennoch kann man hier den bekannten und recht günstigen Programmierer AVRISP mkII einsetzen [6], der selbständig auch die 3,3-V-Betriebsspannung des Xmega erkennt. Ein Reset-Taster fehlt ebenfalls nicht.

Schnittstellen

Für die Kommunikation von Board zu Board ist die gute alte serielle UART-Schnittstelle immer noch die beliebteste. Die Programmierung ist einfach dank der controllerinternen Hardware-UARTs und es gibt viele Konverter, unter anderem für RS485 und USB. Unser Controller besitzt gleich sechs UART-Units, von denen wir auf dem Board drei direkt nutzen.

Die Signale des UARTs PC2/PC3 leiten wir einem RS485-Treiberchip zu; dessen A- und B-Leitungen sind wiederum auf Schraubklemmen geführt. Da auch die Masse an einer Schraubklemme zugänglich ist, lässt sich das Board sehr einfach mit anderen RS485/ElektorBus-Platinen verbinden. Zum Beispiel kann man hier über drei Leitungen unseren bekannten RS485/USB-Konverter [7] anschließen und sich dann wunderbar mit einem (auch weit entfernten) PC unterhalten; über das Andropod-Modul [8] geht das genauso mit einem Tablet oder Smartphone. Das Board lässt sich darüber hinaus mittels einer vierten Schraubklemme über eine 12-V-Leitung (ElektorBus) versorgen.

Über die zwei Digitalsignale DE und /RE aktiviert man das Senden und Empfangen des RS485-Chips. Mit JP2 kann man einen Abschlusswiderstand von 120 Ω an den Bus legen. Die optionalen Widerstände R11 und R18 ziehen die Leitungen A und B auf definierte Pegel, falls alle Knoten inaktiv sind. Das vermindert die Empfindlichkeit gegenüber Einstreuungen; bei unseren bisherigen Bus-Experimenten sind wir aber gut ohne das „Biasing“ ausgekommen.

Einen weiteren UART haben wir auf einen Mini-DIN-Stecker geführt, neben vier GPIO-Signalen. Der Mini-DIN-Stecker ist optional selbst zu bestücken, dann kann man über ein Kabel mit anderer Elektronik kommunizieren. Auf dem Andropod-Modul ist übrigens der gleiche Mini-DIN-Stecker vorgesehen. Um das Ganze perfekt zu machen, lässt sich mittels JP3 wählen, ob die Signale 3,3-V- oder 5-V-kompatibel sein sollen. Ein dritter UART (PD2/PD3) ist für den Anschluss

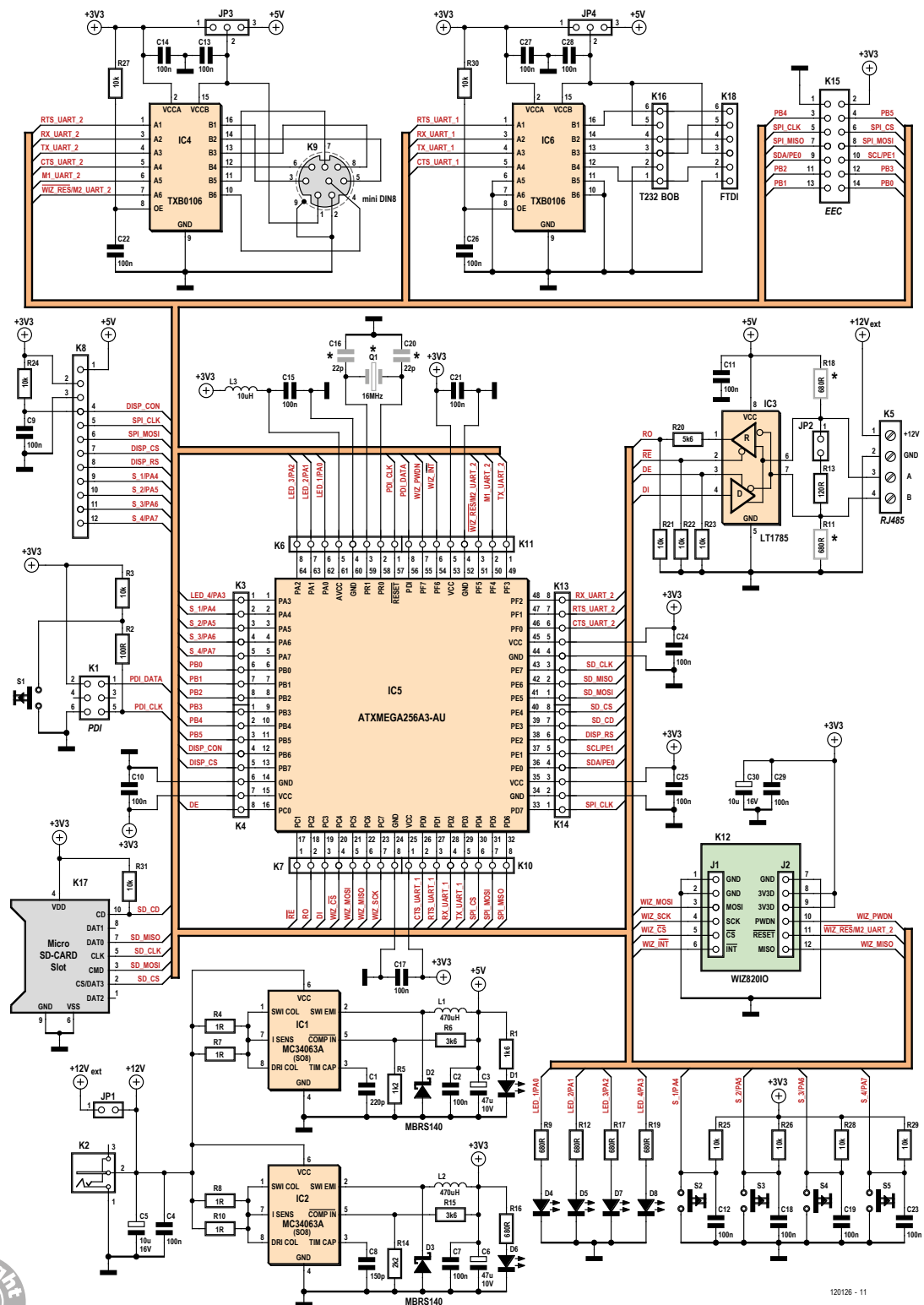
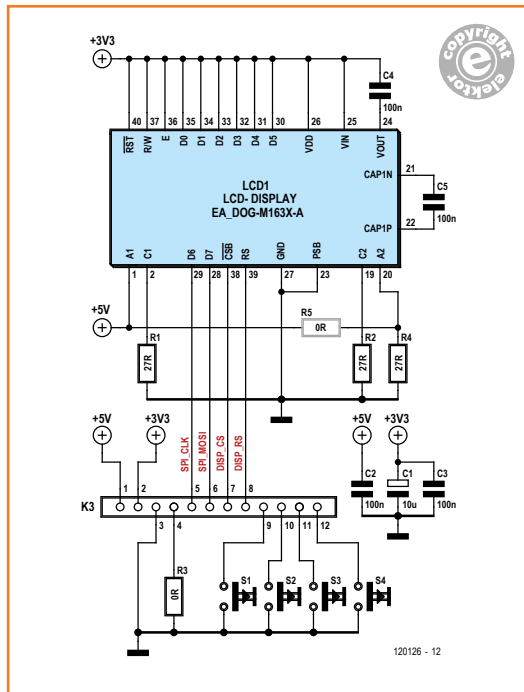


Bild 3. Schaltplan des Xmega-Boards. Die Pins des Controllers sind gut genutzt, unter anderem werden drei SPI-, drei UART- und eine I2C-Unit verwendet.

Bild 4.
Schaltplan des Display-
Moduls. Mit den vier Tastern
lässt sich eine Menü-
Führung verwirklichen.



tinien, welche das Team von Embedded Projects für das Linux-Board konzipiert hat. Bei Elektor erhältlich sind eine Erweiterung mit Display, Portexpander und Echtzeituhr [9], ein Relais-Board [10], eine Schrittmotorsteuerungsplatine, ein Temperatursensor und mehr (siehe den Artikel im letzten Heft [11] und [12]).

Zum Testen, Debuggen und Erweitern kann man rings um den Controller noch weitere Stift- oder Buchsenleisten bestücken, hierdurch bekommt man Zugang zu nahezu sämtlichen Pins des Xmega. Beim vom Elektor gelieferten Board sind an der unteren Kante Buchsenleisten eingelötet, damit das unten beschriebene Display-Modul sicheren Halt bekommt; elektrisch hat diese Verbindung keine Funktion.

UI-Interface

Wenn man das Board als Schaltzentrale in MSR-Anwendungen einsetzt, dann ist ein Display sehr wertvoll. Nicht jeder wird aber ein Display benötigen; wir haben es deshalb als eigenes

... einfach Webserver- und andere Netzwerk-Anwendungen realisieren ...

eines USB/UART-Konverters vorgesehen. Hier lassen wir unseren Lesern die Wahl zwischen einem USB/TTL-Adapterkabel von FTDI und der kleinen USB/TTL-Konverterplatine BOB, beide sind bei Elektor erhältlich [2]. Auch hier sorgt ein Pegelwandler für Kompatibilität sowohl mit 3,3-V- als auch mit 5-V-Elektronik; die gewünschte Spannung ist mit JP4 einzustellen. Passende Stiftleisten bestückt man bei Bedarf, entweder gerade oder gewinkelt.

Erweiterung

Unser Board ist ja schon sehr gut ausgestattet. Dennoch mag es Anwendungen geben, die nach einer Erweiterung der Board-Fähigkeiten durch weitere ICs verlangen. Diese ICs werden in den meisten Fällen über SPI und I2C angesteuert, so dass wir die entsprechenden Controller-Pins ebenfalls über eine Stiftleiste nach außen führen wollten. Dabei hatten wir Idee, dieselbe Pinbelegung zu verwenden, wie sie Benedikt Sauter beim Elektor-Linux-Board genutzt hat. An den 14-poligen „Embedded Extension Connector (EEC)“ passen nun auch alle Erweiterungspla-

Modul ausgeführt, das auf das Controllerboard aufsteckbar ist. Dies gibt dem Anwender auch die Möglichkeit, die Anzeige an anderer Stelle zu platzieren, etwa außen an einem Gehäuse. Der Steckverbinder K8 zur Verbindung mit dem Display besitzt 12 Pins, drei davon sind für die Masse und die beiden Betriebsspannungen reserviert. Das Display selbst wird über SPI angesteuert; zu den drei SPI-Leitungen MOSI, SCK und CS kommt hier noch eine vierte Leitung RS, die bestimmt, ob ein Kommando- oder ein Datenbyte zum Display geschickt wird. Zu Details sei hier auf das Datenblatt [13] hingewiesen.

Falls das Display-Modul auf das Controllerboard aufgesteckt ist, wird ein weiterer Pin am Stecker auf Masse gelegt. Dieser Steckerpin 4 ist mit Portpin PB 6 des Controllers verbunden. Falls man PB 6 per Software auf Eingang schaltet und den internen Pullup aktiviert, so lässt sich über den Level „Low“ oder „High“ abfragen, ob das Display aufgesteckt ist oder nicht.

Vier weitere Leitungen sind den vier Tastern gewidmet, die sich ebenfalls auf dem Displaymodul befinden (**Bild 4**). Um ein Menü realisie-

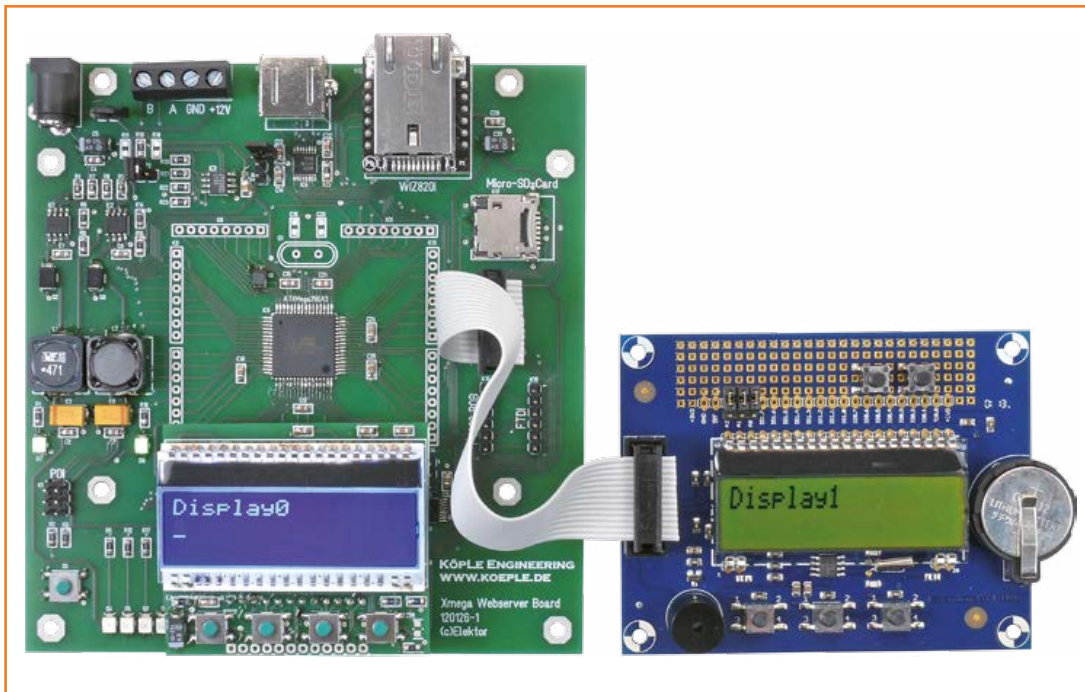


Bild 5.
An den „Embedded Extension Connector“ lassen sich diverse Erweiterungsboards anschließen.

ren zu können, wurden diese direkt unterhalb des Displays angeordnet.

Diejenigen Anwender, die das Displaymodul nicht einsetzen, müssen trotzdem nicht auf Taster verzichten. Wir haben die Taster nämlich „gedoppelt“, direkt auf dem Board befinden sich nochmals vier davon, die an denselben Portpins des Controllers hängen. Die Kondensatoren C12, C18, C19 und C23 dienen zur Entprellung.

Last but not least seien in diesem Abschnitt die LEDs erwähnt, die zum Debuggen und für einfache Anzeigen unverzichtbar sind; wir haben dem Board gleich vier Stück spendiert.

SD-Karte und Netzwerk

Das SD-Karten-Scharnier nimmt Mikro-SD-Karten auf; sie dienen als „Massenspeicher“ für unser Board. Die SD-Karte wird hier im sogenannten SPI-Mode angesteuert; die vier Leitungen MISO, MOSI, SCK und CS sind an eine SPI-Hardware-Unit (Port E) des Xmegas angeschlossen. Genauso wie für die anderen Peripherie-Einheiten geben wir auch hier Unterstützung in Form einer kleinen Bibliothek, zumindest für Rohdaten (Karte nur auf dem Board beschreib- und wieder lesbar). Wer selbst eine Ansteuerung programmieren will, findet eine gute Einführung in das Thema unter [14]. Der CD-Pin am SD-Karten-Scharnier wird auf Masse gezogen, wenn eine Karte eingelegt

ist, er ist mit Portpin PE3 des Xmegas verbunden. Highlight des Boards ist sicherlich der Steckplatz für das Netzwerk-Modul „Wiz820io“, das ebenfalls über Elektor (unter der Nummer 130076-91) erhältlich ist [2]. Es handelt sich dabei um ein eigenes kleines Prozessorboard mit integriertem TCP/IP-Stack; unser Xmega muss sich um diese Protokollebenen des Netzwerkverkehrs also nicht kümmern. Der Xmega teilt dem kleinen Modul nur mit, ob ein Socket (bestimmte IP-Adresse und Port) geöffnet werden soll, ob Zeichen gesendet werden sollen und mehr. Genauso können empfangene Zeichen von einem Socket abgerufen werden. Die Kommunikation zwischen Xmega und Netzwerkmodul läuft dabei ebenfalls über SPI; auf unserem Board wird die SPI-Unit an Port C genutzt. Hier sei wieder – um den Rahmen des Artikels nicht zu sprengen – auf die Dokumentation des Herstellers WizNet hingewiesen [15]. Die Koreaner bieten auch Treiber in C für eine Reihe von Controllern an, die dem Anwendungscode (also zum Beispiel einem Webserver) Funktionen wie `SocketOpen(...)` zur Verfügung stellen. Wir haben diesen Treiber nach unten hin an das Board und den Xmega angepasst und nach oben hin noch um ein paar einfache Funktionen erweitert. Zum Einsatz des Boards im Heimnetzwerk und im Internet ist demnächst ein eigener Artikel geplant.

Listing 1. Demo mit LEDs, Tastern und Display.

```
int main(void)
{
    Controller_Init();
    Board_Init();
    //Extension_Init();

    ApplicationSetup();

    while(1)
    {
        ApplicationLoop();
    }
};

void ApplicationSetup(void)
{
    LEDButton_LibrarySetup(ButtonEventCallback);

    Display_LibrarySetup();
    Display_WriteString(0, 0, "Display0");
    //Display_WriteString(1, 0, "Display1");
}

void ApplicationLoop()
{
    ButtonPollAll();
}

void ButtonEventCallback(uint8 BlockType, uint8 BlockNumber,
uint8 ButtonPosition, uint8 Event)
{
    //Buzzer(BuzzerBlockFirstIndex, 1000,
BUZZER_TONEMODE_RAMP);

    if (Event == EVENT_BUTTON_PRESSED)
    {
        ToggleLED(0, 0);
        Display_WriteNumber(0, 1, BlockNumber);
        Display_WriteNumber(0, 2, ButtonPosition);
    }
}
```

Software

Die bereits vorliegende ElektorBus-Bibliothek, eine kleine Display-Bibliothek von KoepLe und der WizNet-Treiber waren ein Geburtshelfer für die „Embedded Firmware Library“, über die wir bereits in zwei Elektor-Artikeln berichtet haben [16][17]. Da die Arbeit an diesem Framework vor allem auf dem Xmega-Webserver-Board vorangetrieben wurde, sind wir nun in der glücklichen Lage, für alle Peripherie-Blöcke auf dem Board Bibliotheksmodule anbieten zu können.

Unter [2] und [18] kann die aktuelle EFL-Codebasis heruntergeladen werden, welche die einzelnen Codemodule sowie eine Demo-Anwendung für unser Board enthält. Die Controller-API für den Xmega befindet sich wie immer in einem Filepaar, das den Namen ControllerEFL.h/.c trägt. Die beiden Files liegen in diesem Fall im Unterordner Xmega256A3. Es werden Funktionen zum Setzen und Lesen der digitalen Aus-/Eingänge, zur Abfrage der ADC-Pins, zum Senden und Empfangen über eine UART-Einheit und vieles mehr bereitgestellt, was ein Studium des Datenblattes überflüssig macht.

Das Boardfile enthält Code, der ebendiese Controllerfunktionen aufruft. Für höhere Schichten der EFL werden Low-Level-Funktionen zum Ansprechen der Peripherieblöcke bereitgestellt. Wie zum Beispiel:

```
void Display_SendByte(uint8
DisplayBlockIndex, uint8 ByteToSend,
uint8 DATABYTE_COMMANDBYTE)
```

Diese Funktion verschickt ein Byte über diejenige SPI-Unit, an die das Display angeschlossen ist (in der internen Peripherieblock-Tabelle steht ein Verweis auf diese Einheit). Gleichzeitig wird das RS-Digitalsignal zur Unterscheidung eines Befehls- oder Datenbytes bedient, die Funktion kann hierzu nachschauen, an welchen Controllerpin RS angeschlossen ist. Vor den höheren Schichten wird die Verdrahtung auf dem Board verborgen; die Display-Bibliothek und die eigentliche Applikation können hardware-unabhängig programmiert werden. Es kann dem Anwender sogar egal sein, ob das Display seriell über SPI oder parallel über vier Datenleitungen angesteuert wird, es wird nur immer die Funktion Display_SendByte() aufgerufen.

Ähnliche Low-Level-Funktionen sind im Board-File für die Ansteuerung der SD-Karte und des Netzwerkmoduls vorgesehen. Eine Dokumenta-

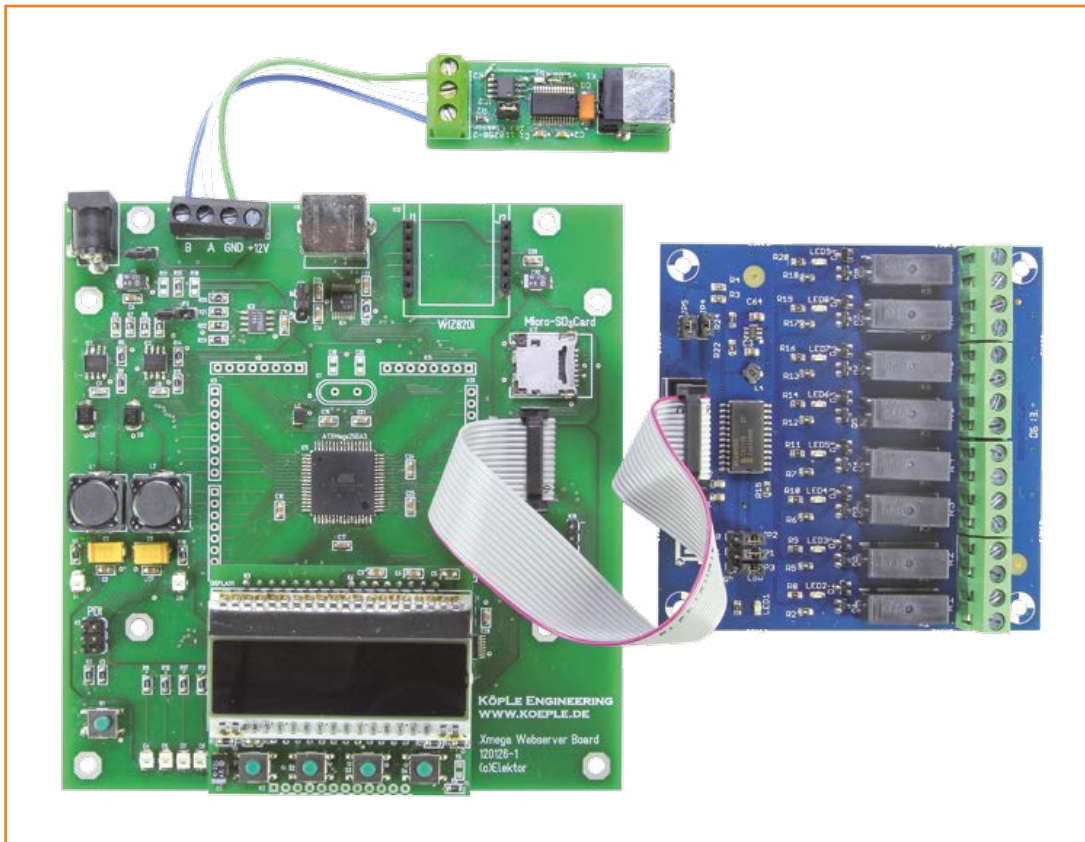


Bild 6.
Zusammen mit acht Relais
entsteht eine mächtige
Schaltzentrale. Hier mit
einer zusätzlichen PC-
Steuerung über RS485.

tion der Funktionen (erstellt mit Doxygen) findet man in der Codebasis. Im Ordner „Manuals“ liegt ein Zusatzdokument in Deutsch, Englisch und Französisch, das in die Interna der EFL einführt.

Das erste Programm

Die hardware-unabhängige Display-Bibliothek findet man in der Codebasis im Ordner „Libraries“ (DisplayEFL.h/c).

Im Anwendungsprogramm wird die Bibliothek mit dem Befehl

```
Display_LibrarySetup();
```

initialisiert. Danach kann man bis zu vier Displays, die sich auf dem Controller-Board oder auch einer Erweiterung befinden können, über eine Nummer 0..3 ansprechen. Dies können wir testen mit der Demo-Applikation „XmegaDemo“. Wir klicken uns in der Codebasis im Ordner „Applications“ bis hinunter zum File „XmegaDemo.atsln“. Ein Doppelklick öffnet das Projekt in Atmel Studio 6. Wenn wir das zugehörige Hex-File in den Xmega flashen, müsste in der ersten Zeile des Displays die Zeile „Display0“ erscheinen. Betätigt man

einen der Buttons, wird dessen Nummer auf dem Display angezeigt, außerdem toggelt die erste LED auf dem Board.

Listing 1 zeigt den Quellcode. In der ApplicationSetup-Funktion initialisieren wir neben der Display-Library auch die LEDButton-Library, dabei teilen wir der Bibliothek mit, welche Funktion in unserer Applikation angesprochen werden soll, wenn ein Taster betätigt wurde.

Damit die Taster regelmäßig abgefragt werden, muss die ApplicationLoop-Funktion die Codezeile `ButtonPollAll();` enthalten.

In der Funktion `ButtonEventCallback(...)` befindet sich der Code, der bei einem Taster-Ereignis ausgeführt wird. In der Variablen `ButtonPosition` steht die Nummer des Buttons auf dem Xmega-Webserver-Board (0..3). Die Variable `Event` kann die Werte `EVENT_BUTTON_PRESSED (= 1)` oder `EVENT_BUTTON_RELEASED (= 2)` annehmen. Man kann seine Anwendung also unterschiedlich auf das Drücken und Loslassen des Tasters reagieren lassen.

Erweiterung

Wer ein Linux-Extension-Board [9] sein eigen

Über die Entwickler

Bernd Köppendörfer und Achim Lengl schlossen 2009 ihr Studium zum Diplom-Ingenieur der Elektro- und Informationstechnik an der „Georg-Simon-Ohm Hochschule“ in Nürnberg ab. 2010 gründeten sie ihr eigenes Unternehmen „KöpLe Engineering GbR“ mit Sitz in Oberasbach. Seitdem unterstützen sie andere Firmen als Berater und übernehmen Entwicklungsaufträge im Bereich der analogen und digitalen Schaltungstechnik; angefangen bei einfachen Baugruppen bis hin zur FPGA-Implementierung für komplexe Bildverarbeitungssysteme in Echtzeit.

nennt, sollte dieses einmal mit einem Flachbandkabel an den „Embedded Extension Connector“ anschließen (siehe **Bild 5**). Zu diesem Extension-Board gehört ein eigenes Filepaar namens „ExtensionEFL.h/.c“, das die nötigen Low-Level-Funktionen für die Peripherieblöcke auf der

Listing 2. Steuerung über RS485/UART.

```
int main(void)
{
    Controller_Init();
    Board_Init();
    Extension_Init();

    ApplicationSetup();

    while(1)
    {
        ApplicationLoop();
    }
};

void ApplicationSetup(void)
{
    UARTInterface_LibrarySetup();
    UARTInterface_SetBaudrate(0, 38400);

    BlockProtocol_LibrarySetup(UARTInterface_Send, 0,
    UARTInterface_GetRingbuffer(0));
}

void ApplicationLoop()
{
    BlockProtocol_Engine();
}
```

Erweiterungsplatine mitbringt. Die Files sind schon im Projekt enthalten, wir müssen nur noch die Kommentarzeichen vor der Zeile `Extension_Init()` weglöschen. Das machen wir dann gleich auch mit den anderen auskommentierten Zeilen im Code des Hauptfiles.

Nach Compilieren und Flashen des Programms in den Xmega werden nun auch die drei Taster des Erweiterungsboards abgefragt. Über die Variable `BlockNumber` könnten wir in der Funktion `ButtonEventCallback(...)` unterscheiden, in welchem Tasterblock das Ereignis auftrat (0 = Buttons auf dem Controllerboard, 1 = Erweiterungsboard).

Man sieht, dass es für die Applikation keinen Unterschied macht, ob sich die Taster bzw. das Display auf dem Controllerboard oder der Erweiterungsplatine befinden. Das ist insofern bemerkenswert, als die Taster auf dem Linux-Extension-Board gar nicht über die digitalen Eingänge, sondern über die analogen Eingänge des Controllers abgefragt werden (siehe den Abschnitt „Virtualisierung“ im EFL-Zusatzdokument).

Schalten und Walten

Wie schon in einigen Artikeln zuvor gezeigt, verbinden wir uns nun über RS485 und den bekannten RS485/USB-Konverter [6] mit einem PC. An den EEC-Steckverbinder schließen wir das im letzten Heft vorgestellte Relais-Board [10] an; das Ganze müsste dann wie in **Bild 6** aussehen.

Als Anwendung rufen wir diesmal „XmegaRelay.atsIn“ auf. Die eigentliche Anwendung im Hauptfile ist erstaunlich kurz, wie **Listing 2** zeigt. Was in der `ApplicationSetup`- und der `ApplicationLoop`-Funktion passiert, ist ausführlich im EFL-Artikel des Juni-Hefts erklärt [17]. Neben der Vorbereitung der UART-Interfaces auf dem Board (für RS485 und FTDI/BOB) initialisieren wir auch eine Library, die für ein einfaches Steuerungs-Protokoll namens `BlockProtocol` zuständig ist.

In der `ApplicationLoop`-Funktion fragen wir mit der Zeile

```
BlockProtocol_Engine();
```

ab, ob neue Befehle vom PC aus beim Board eingegangen sind.

Nach Flashen des zugehörigen Hex-Files brauchen wir nur noch ein Terminalprogramm auf dem PC zu öffnen sowie den passenden COM-Port und die Datenrate 38400 Baud einzustellen.

Gegebenenfalls muss man das Terminalprogramm der Wahl noch so konfigurieren, dass bei Betätigung der Enter-Taste die vorher eingegebenen Zeichen verschickt werden, gefolgt von einem CR (ASCII 13).

Nach Eingabe von

R 0 0 + <ENTER>

sollte das erste Relais anziehen,

R 0 0 - <ENTER>

hat den gegenteiligen Effekt. Die übrigen sieben Relais steuert man mit R 0 x ... an (x = 1..7). Wer keinen USB/RS485-Konverter zur Hand hat, kann sich auch über ein FTDI-Kabel oder den BOB mit dem PC verbinden. Statt des ersten

Stückliste

Widerstände:

(SMD, 0805)
 R1 = 1k6
 R2 = 100 Ω
 R3,R21..R31 = 10 k
 R4,R7,R8,R10 = 1 Ω
 R5 = 1k2
 R6,R15 = 3k6
 R9,R12,R16,R17,R19 = 680 Ω
 R11,R18 = 680 Ω (optional)
 R13 = 120 Ω
 R14 = 2k2
 R20 = 5k6

Kondensatoren:

C1 = 220 p (0805)
 C2,C4,C7,C9..C15,C17,C18,C19,C21..C29 = 100 n (0805)
 C3,C6 = 47 μ (Tantal, 10 V, SMD-D/E)
 C5,C30 = 10 μ (Tantal, 16 V, SMD-C)
 C8 = 150 p (0805)
 C16,C20 = 22 pF (optional)

Induktivitäten:

L1,L2 = 470 μ H (Ferrit, PIS4728)
 L3 = 10 μ H (LQH3C)

Halbleiter:

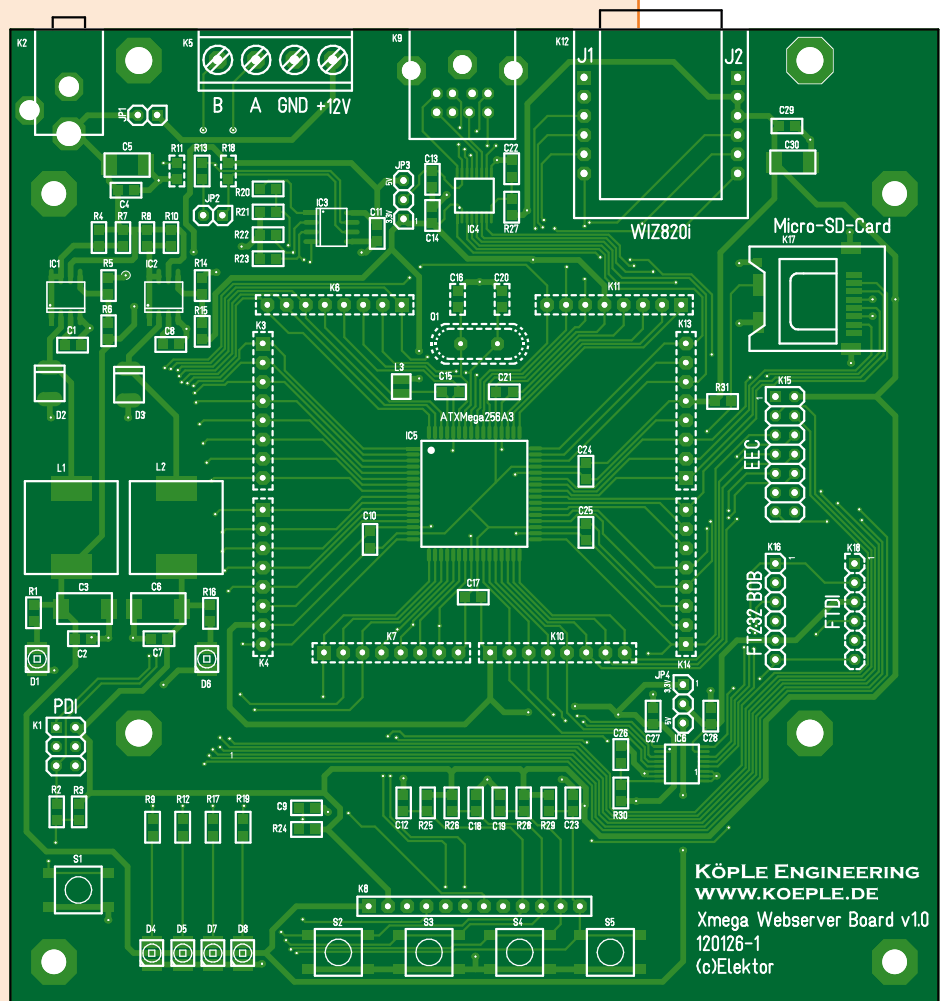
D1,D4..D8 = LED LG T67K (PLCC2)
 D2,D3 = MBRS140 Schottky-Diode
 IC1,IC2 = MC34063A Schaltregler (SO8)
 IC3 = LT1785CS8 RS485-Treiber (SO8)
 IC4,IC6 = TXB0106 Pegelwandler (TSSOP16)
 IC5 = ATXmega256A3-AU (TQFP64)

Außerdem:

JP1,JP2 = Stiftleiste 1x2, mit Jumper, RM 2,54
 JP3,JP4 = Stiftleiste 1x3, mit Jumper, RM 2,54
 K1 = Stiftleiste 2x3, RM 2,54
 K2 = 2,5-mm-Klinkenbuchse für Lötmontage
 K5 = Schraubklemme 4-polig, RM 5,08, für Lötmontage
 K8 = Buchsenleiste 1x12, RM 2,54
 K9 = 8-pol. Mini-DIN-Buchse (optional)
 K12 = 2x Buchsenleiste 1x6, RM 2,54, für Modul Wiz820io
 K15 = Stiftleiste 2x7, RM 2,54
 K16 = Stiftleiste 1x6, RM 2,54, für USB/TTL-Konverter BOB (optional)
 K17 = Scharnier-Sockel für Mikro-SD-Karte

K18 = Stiftleiste 1x6, RM 2,54, für USB/TTL-Kabel von FTDI (optional)
 Q1 = Quarz 16 MHz (optional)
 S1,S2,S3,S4,S5 = SMD-Kurzhubtaster
 Platine 120126-1

Oder
 120126-91 Controllerboard fertig bestückt und getestet



Stückliste Display-Modul

Widerstände:

(SMD, 0805)
R1,R2,R4 = 27 Ω
R3 = 0 Ω
R5 = 0 Ω (optional)

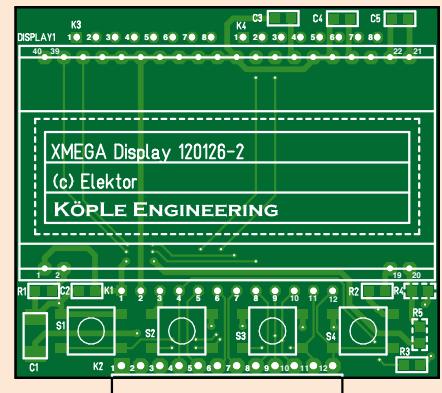
Kondensatoren:

C1 = 10 μ (Tantal, 16 V, SMD-C)
C2..C5 = 100 n (0805)

Außerdem:

DISPLAY1 = Display EA-DOG-M163X-A mit
LED-Backlight LED55X31
+ Buchsenleiste 1x20, 1x2, 1x2 für Montage
K1 = Stiftleiste 1x12, RM 2,54
K2 = Stiftleiste 1x12, RM 2,54 (optional)
K3,K4 = Stiftleiste 1x8, RM 2,54
S1,S2,S3,S4 = SMD-Kurzhubtaster

Platine 120126-2



Oder

120126-92 Display-Modul fertig bestückt und getestet

muss nun der zweite UARTInterface-Block auf dem Board verwendet werden, der die Nummer #1 trägt. Es ist lediglich eine einzige Codezeile im Programm zu ändern:

```
BlockProtocol_LibrarySetup(UARTInterface_
Send, 0, UARTInterface_GetRingbuffer(0));
```

in

```
BlockProtocol_LibrarySetup(UARTInterface_
Send, 1, UARTInterface_GetRingbuffer(1));
```

Und schon laufen die Nachrichten über den anderen UART-Kanal. Übrigens könnten wir die Kommandos auch über TCP/IP empfangen mit

```
BlockProtocol_LibrarySetup(IPInterface_
Send, 0, IPInterface_GetRingbuffer(0));
```

Das probieren wir im zweiten Artikel zu diesem Board aus!

(120126)

Weblinks

[1] www.koeple.de

[2] www.elektor.de/120126

[3] www.atmel.com/Images/Atmel-8331-8-and-16-bit-AVR-Microcontroller-XMEGA-AU_Manual.pdf

[4] www.onsemi.com/pub_link/Collateral/MC34063A-D.PDF

[5] www.a-z-e.de/index.php?file=mc34063.php

[6] www.atmel.com/tools/AVRISPMKII.aspx

[7] www.elektor.de/110258

[8] www.elektor.de/110405

[9] www.elektor.de/120596

[10] www.elektor.de/130157

[11] www.elektor.de/130212

[12] www.elektor.de/gnublin

[13] www.lcd-module.de/pdf/doma/dog-m.pdf

[14] http://elm-chan.org/docs/mmc/mmc_e.html

[15] www.wiznet.co.kr/WIZ820IO

[16] www.elektor.de/120668

[17] www.elektor.de/130154

[18] www.elektor-labs.com/efl

NEU

LabVIEW 2

Arrays und serielle Daten

Der zweite Band der LabVIEW-Lehrbuchreihe beschäftigt sich u. a. mit Arrays, Cluster und den seriellen VISA-Funktionen. Als Erstes werden vier neue zusammengesetzte Datentypen (Enum, Ring, Array, Cluster) vorgestellt und deren Verwendung wird anhand zahlreicher praktischer Beispiele und Übungen erläutert.

Danach wird es praktisch: Ein 8051er-Mikrocontrollersystem dient als Datenquelle und -senke für verschiedene LabVIEW-VIs, die einerseits beliebige Messdaten empfangen, auswerten und die Ergebnisse numerisch oder graphisch auf dem Rechner-Monitor darstellen. Andererseits können sie auch Steueranweisungen an das Mikrocontrollersystem ausgeben.

Der gesamte Datentransfer zwischen Mikrocontroller und LabVIEW-VI läuft dabei über serielle COM-Schnittstellen oder mittels USB/Seriell-Adapter über USB-Ports.

Da alle VIs universell aufgebaut und ausführlich erklärt werden, lassen sie sich problemlos mit anderen Mikrocontrollerboards (AVR-, PIC-, ARM-, Arduino-, Raspberry PI-Systeme, E-blocks) kombinieren, sofern diese über eine serielle Schnittstelle verfügen.

248 Seiten (kart.) · Format 17 x 23,5 cm · ISBN 978-3-89576-274-1

€ 34,80 · CHF 43,20



Weitere Infos & Bestellung unter www.elektor.de/labview2

BATRONIX

FÜHRENDE PROGRAMMIER- UND MESSLÖSUNGEN



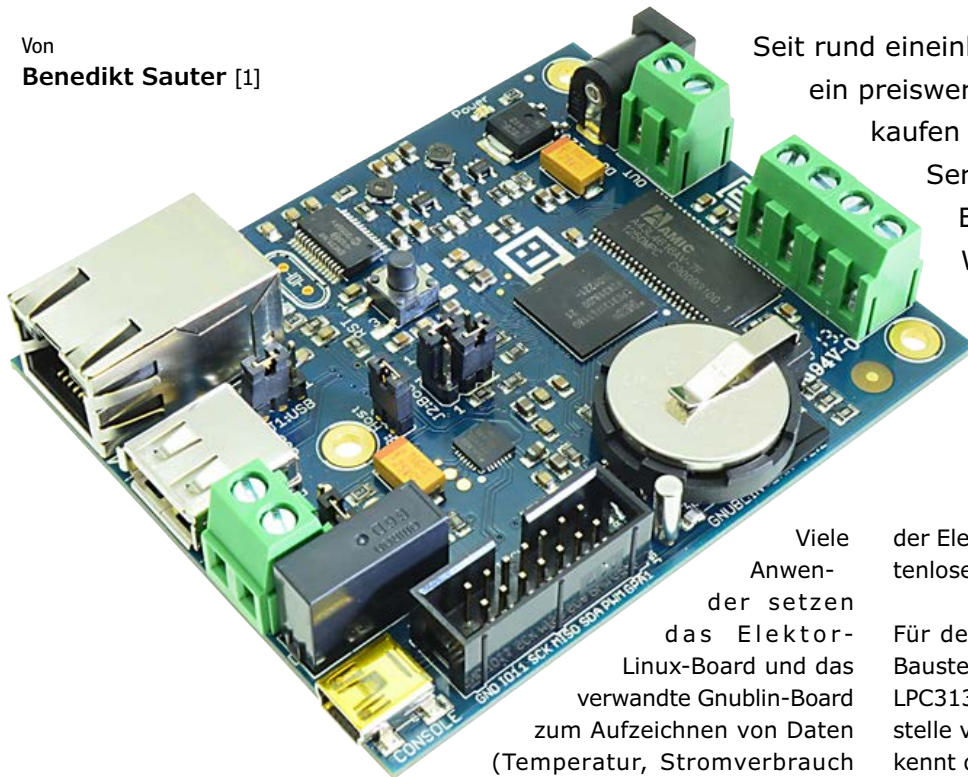
Jetzt
Angebote
nutzen!

ELEKTOR SONDERPREIS AKTION

Rigol Oszilloskope, Heißluft- & Lötstationen und UNI-T Multimeter zum Sonderpreis! Nur für Elektor Leser und bis zum 15.11.2013.
Jetzt Angebote nutzen: www.batronix.com/go/27

Elektor-Linux-Board: Hier ist Version 2! Jetzt mit LAN und Echtzeituhr

Von
Benedikt Sauter [1]



Seit rund eineinhalb Jahren kann man bei Elektor ein preiswertes und kompaktes Linux-Board kaufen – begleitet von einer Artikel-Serie, mit der auch Anfängern der Einstieg in die „Embedded Linux“-Welt gelingt. Nun ist es Zeit für eine Neuauflage des Boards, in die auch Feedback aus der Community eingeflossen ist.

Viele Anwender setzen das Elektor-Linux-Board und das verwandte Gnublin-Board zum Aufzeichnen von Daten (Temperatur, Stromverbrauch u.a.) ein, die dann über ein Netzwerk abgefragt werden. Da lag es nahe, auf dem neuen Board einen Netzwerkzugang und eine Echtzeituhr zu integrieren (Bild 1). Darüber hinaus wurden weitere Vorschläge von Usern umgesetzt. So sind beispielsweise die Befestigungslöcher vergrößert worden, um das Board mit größeren Schrauben befestigen zu können.

Zusätzliche Chips

Die Grundsaltung des Boards hat sich gegenüber der letzten Version [2] nicht verändert. Neben dem Prozessor befinden sich 32 MB RAM, ein USB/Seriell-Adapter (CP2102) und die Stromversorgung auf dem Board. Wie auch beim ersten Elektor-Linux-Board lassen sich am 14-poligen Gnublin/EEC-Steckverbinder eine Reihe von Erweiterungsplatinen anschließen [3][4], die man ebenfalls über Elektor beziehen kann [5]. Schaltplan und Platine im Eagle-Format kann man von

der Elektor-Website downloaden [6], einen kostenlosen Eagle-Viewer findet man unter [7].

Für den Netzwerkzugang musste ein eigener Baustein hinzugefügt werden, da der Prozessor LPC3131 über keine integrierte Netzwerkschnittstelle verfügt. Den ENC28J60 von Microchip [8] kennt der eine oder andere Leser sicher bereits als Netzwerkadapter für einfache 8-bit-Prozessoren. Er wird über SPI und eine zusätzliche Interruptleitung angebunden. Im Kernelarchiv existiert für diesen Controller schon lange ein passender Treiber.

Bei der Echtzeituhr (RTC) fiel die Wahl auf den Baustein MCP7940 [9]. Dieser benötigt einen externen Quarz. Wenn man eine Knopfbatte-rie als Puffer einsetzt, dann kann die RTC auch im stromlosen Zustand des Boards die Uhrzeit behalten.

Netzwerk konfigurieren

Um den LAN-Chip ENC28J60 auf dem Board verwenden zu können, muss als erstes mit ...

```
modprobe enc28j60 irq_pin=12 cs_pin=19
```

... der Treiber geladen werden. Wenn das geklappt hat, sollte man als Ausgabe von ...

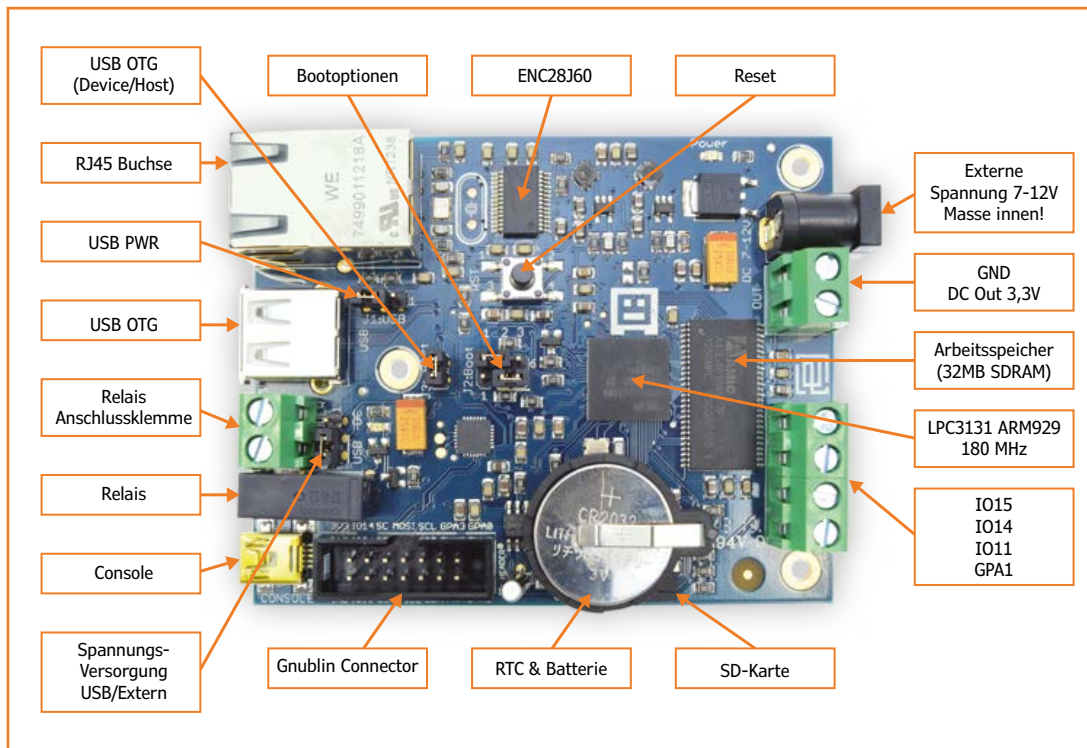


Bild 1.
Das „GnuBlin/Elektor-Linux-Board LAN“ verfügt über einen Netzwerkzugang und eine Echtzeituhr.

```
ifconfig -a
```

... unter anderem das „eth0“-Interface sehen. Um vom DHCP-Server im eigenen Netzwerk eine IP-Adresse zu beziehen, gibt man folgenden Befehl ein:

```
dhclient eth0
```

Nun sollte ein Test folgen, indem ein beliebiger PC im eigenen Netz bzw. ein Server im Internet gepingt wird:

```
ping google.de
```

Man erhält nun zum Beispiel eine Ausgabe wie in Bild 2. Beendet wird das Programm ping wie immer mit Strg + c.

Treiber automatisch laden

Soll der enc28j60-Treiber automatisch beim Systemstart geladen werden, muss ...
enc28j60 irq_pin=12 cs_pin=19

... in die Datei „/etc/modules“ im Dateisystem des Boards eingefügt werden. Dies macht man am einfachsten von der Konsole aus. Der Befehl ...

```
echo "enc28j60 irq_pin=12 cs_pin=19" >> /etc/modules
```

... hängt die Zeile am Ende der Datei automatisch an. Optional kann man die Datei auch mit dem bekannten Texteditor nano bearbeiten:

```
nano /etc/modules
```

Feste MAC-Adresse

Wenn das Board nach jedem Neustart eine andere IP-Adresse vom DHCP-Server bekommt, liegt das daran, dass der ENC28J60-Chip über keine feste MAC-Adresse verfügt und bei jedem Laden des Treibers eine andere Adresse ausgibt. Um das umgehen zu können, kann man eine feste MAC-Adresse vorgeben. Dazu wird in der Datei „/etc/network/interfaces“ die folgende Zeile eingetragen:

```
hwaddress ether MAC-ADRESSE
```

Als MAC-Adresse kann man z.B. diejenige auswählen, die das LAN-Modul beim ersten Laden des Treibers bekommen hat. Diese wird mit ...

```
ifconfig
```

... angezeigt, und zwar als „Hwaddr“:

```
eth0      Link encap:Ethernet  Hwaddr
ba:07:1b:0c:64:0
```

Echtzeituhr

Um die Echtzeituhr zu setzen, genügt ein kurzer Befehl:

```
gnublin-rtc -s "2013/01/20 11:23:12"
```

Möchte man die Uhrzeit auslesen, macht man dies mit:

```
gnublin-rtc -g
```

Und will man, dass die Uhrzeit aus der RTC von Linux übernommen wird, dann geht das wie folgt:

```
gnublin-rtc -x
```

Möchte man die Systemzeit bei jedem Start nach der RTC stellen, so sind ein paar kleine Veränderungen nötig. Zuerst wird der folgende Text in die Datei „/etc/rc.local“ eingefügt, vor die Zeile „exit 0“:

```
echo mcp7940 0x6f > /sys/bus/i2c/devices/
i2c-1/new_device
echo "Now setting the date and time."
sleep 1
hwclock --hctosys
```

Anschließend wird ein Eintrag in der Datei „/etc/modules“ hinzugefügt:

```
rtc-mcp7940
```

Zuletzt wird noch das hwclock-Shellsript deaktiviert, da es mit diesem Typ von Echtzeituhr manchmal zu Komplikationen führt:

```
update-rc.d hwclock remove && update-rc.d
hwclock.sh remove
```

Nun stellt das Board bei jedem Start die Systemzeit nach der Hardware-Uhr. Weitere Infos zu diesem Thema findet man im Gnublin-Wiki [10].

Ausblick

Es gibt eine neue Version des Gnublin-Installers [11], mit dem man sich auf einem Linux-PC eine SD-Karte mit Bootloader, Kernel und Dateisystem erstellen kann. Die neue Version bietet eine Auswahl zwischen einem 8-MB- und einem 32-MB-Image.

Daneben gab es noch eine Änderung am Dateisystem. Es wird nun EXT4 verwendet, das fast 100 % ausfallsicher bei einem plötzlichen Stromausfall ist. Den sehr zeitintensiven Dateisystem-Check (bei der bisher fälligen „Reparatur“ der SD-Karte) kann man sich damit ersparen.

Mehr dazu in einem weiteren Artikel in der nächsten Ausgabe.

(130214)

Weblinks

[1] sauter@embedded-projects.net

[2] www.elektor.de/120181

[3] www.elektor.de/120596

[4] www.elektor.de/130212

[5] www.elektor.de/gnublin

[6] www.elektor.de/130214

[7] www.cadsoft.de/download-eagle/freeware

[8] <http://ww1.microchip.com/downloads/en/devicedoc/39662a.pdf>

[9] <http://ww1.microchip.com/downloads/en/DeviceDoc/22266D.pdf>

[10] http://en.gnublin.org/index.php/RTC_DS1307

[11] http://en.gnublin.org/index.php/Gnublin_Installer

Bild 2.
Ausgabe bei einem
erfolgreichen Ping: Wir sind
im Netz!

```
PING google.de (173.194.69.94) 56(84) bytes of data:
64 bytes from bk-in-f94.1e100.net (173.194.69.94): icmp_req=1 ttl=49 time=44.3 ms
64 bytes from bk-in-f94.1e100.net (173.194.69.94): icmp_req=2 ttl=49 time=44.7 ms
64 bytes from bk-in-f94.1e100.net (173.194.69.94): icmp_req=3 ttl=49 time=43.3 ms
64 bytes from bk-in-f94.1e100.net (173.194.69.94): icmp_req=4 ttl=49 time=43.4 ms
--- google.de ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 43.308/43.968/44.784/0.673 ms
```


HOLEN SIE SICH DIE NEUESTE VERSION!

Neue Funktionen der Version 6.4

- Simulation Ihres EAGLE Schaltplans in LTspice IV
- Anzeige und Suchfunktion für Attribute im ADD- und REPLACE-Dialog
- Import von Designdaten aus P-CAD, Altium und Protel über das Zwischenformat ACCEL ASCII
- Verbesserte Benutzerführung und Voreinstellungen (Tooltips, Shortcuts)

EAGLE V6



www.cadsoft.de

25 Jahre CadSoft



NEU

45 Experimente mit Hard- und Software für Elektroniker



Der Raspberry Pi stellt einen sehr preiswerten, aber doch vollständigen Computer dar, an den auf einfache Weise verschiedenste Elektronik angeschlossen werden kann. Dieses Buch geht auf eine der Stärken des Raspberry Pi ein: die Kombination von Elektronik und Software.

Dieses Buch beschreibt 45 spannende und interessante Projekte, wie zum Beispiel ein Wechselblinklicht, eine Motorregelung, ein digitales Thermometer, ein Lichtmesser. Aber auch kompliziertere Projekte wie eine Motor-Geschwindigkeitsregelung, ein Webserver mit CGI (Common Gateway Interface) und Client-Server-Programme werden vorgestellt. Sie können dieses Buch als Projektbuch verwenden und die Projekte nachbauen, um sie dann in der Praxis einzusetzen. Durch die ausführliche Beschreibung mit Schaltplänen und Fotos gestaltet sich der Aufbau auf dem Steckbrett recht einfach.

Dieses Buch eignet sich auch gut als Lehrbuch. Bei jedem Projekt wird erklärt, warum es auf diese Art und Weise ausgeführt ist. Sie lernen viel über den Raspberry Pi, Python und die verwendeten Bauteile, so dass Sie selbst die Projekte anpassen, nach eigenen Wünschen erweitern oder mehrere Projekte miteinander kombinieren können.

Software-SD-Karte und Hardware-Starterkit zum Buch sind optional bei Elektor erhältlich.

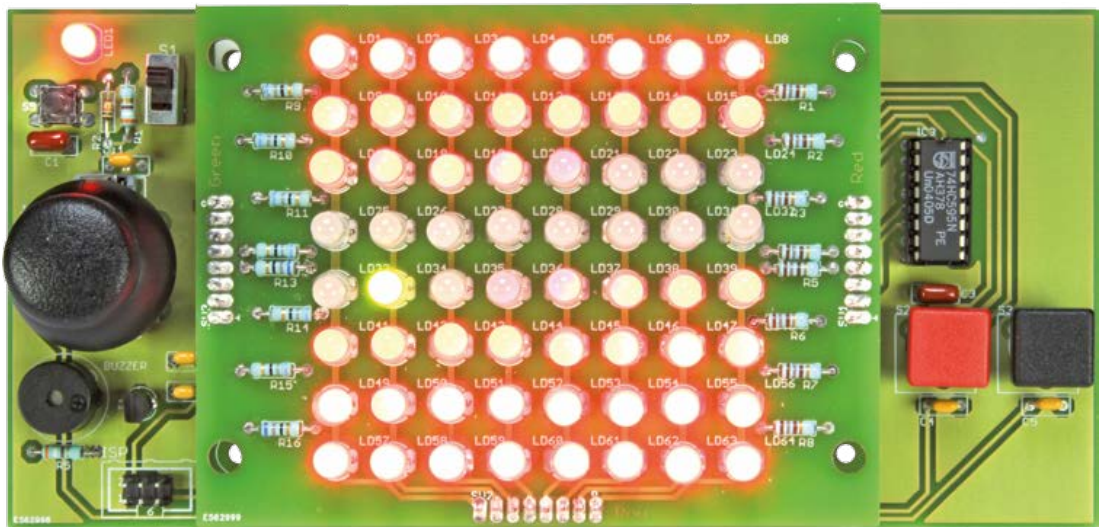
272 Seiten (kart.) • Format 17 x 23,5 cm • ISBN 978-3-89576-273-4

€ 39,80 • CHF 49,40

elektor

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

8x8-Duo-LED-Matrix Mit ATmega328P



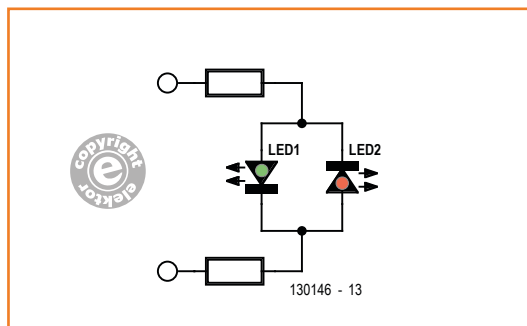
Von
Ruben van Leeuwen
und **Cederique Prevoo**
(Niederlande)

Dieser Artikel beschreibt eine etwas andere Methode, wie eine Matrix sehr vieler LEDs mit nur einigen I/O-Leitungen eines Mikrocontrollers angesteuert werden kann. Als beispielhafte Applikation wurde ein Spiel entwickelt, bei dem man die LEDs mit Hilfe eines Joysticks steuert.

Eine LED-Matrix ist ein guter Ausgangspunkt, wenn man auf dem Gebiet der LED-Ansteuerung Erfahrungen sammeln will. Da man es dabei schnell mit einer großen Zahl von LEDs zu tun bekommt, stößt man auch genauso schnell auf das Problem, dass der Mikrocontroller zu wenig I/O-Leitungen besitzt, um jede LED individuell anzusteuern. Deshalb ist ein Multiplexing erforder-

lich. Wir verwenden hier die weniger bekannte Methode des Charlieplexings (nach Charlie Allen von Maxim, der sie 1995 erfunden hat). Beim normalen Multiplexen werden die Zeilen der Matrix mit dem Steuersignal angesteuert (low oder high) und die Spalten reihum im Takt (der Multiplexfrequenz) ein- und ausgeschaltet. Beim Charlieplexing dagegen wechseln die Funktionen der Zeilen- und Spaltensignale, so dass viel mehr LEDs mit der gleichen I/O-Anzahl gesteuert werden können. **Bild 1** zeigt den grundsätzlichen Aufbau. An Stelle von einer LED werden nun zwei antiparallele LEDs mit den Matrixknoten verbunden. Durch die Änderung der Pegel des Spalten- oder Zeilensignals kann man eine der beiden LEDs aktivieren oder beide dunkel lassen, wenn Zeile und Spalte gleichzeitig low oder high sind.

Bild 1.
Prinzipieller Aufbau beim
Charlieplexing.



Matrix

Die hier verwendete Matrix besitzt 8 x 8 Duo-LEDs, wobei der rote und der grüne Chip im

LED-Gehäuse wie in **Bild 2** angeschlossen sind. Diese LED hat drei Anschlüsse, zwei Anoden und eine gemeinsame Kathode. Den Anschluss für den roten Chip findet man auf der abgeflachten Seite des Gehäuses. Die Schaltung der Matrix, die auf einer von der Ansteuerelektronik separaten Platine untergebracht ist, zeigt **Bild 3**.

Ansteuerelektronik

Um die Ansteuerung kümmert sich ein ATmega328P, ein 8-bit-AVR-Mikrocontroller mit 32 KB programmierbarem Flash-Speicher, 1024 Bytes EEPROM und 2 KB SRAM. In dieser Anwendung ist der interne Oszillator des ATmega ausreichend, auf einen externen Quarz kann verzichtet werden. Der Controller steuert drei 8-bit-Schieberegister 74HC595 an, eines für die grünen LEDs,

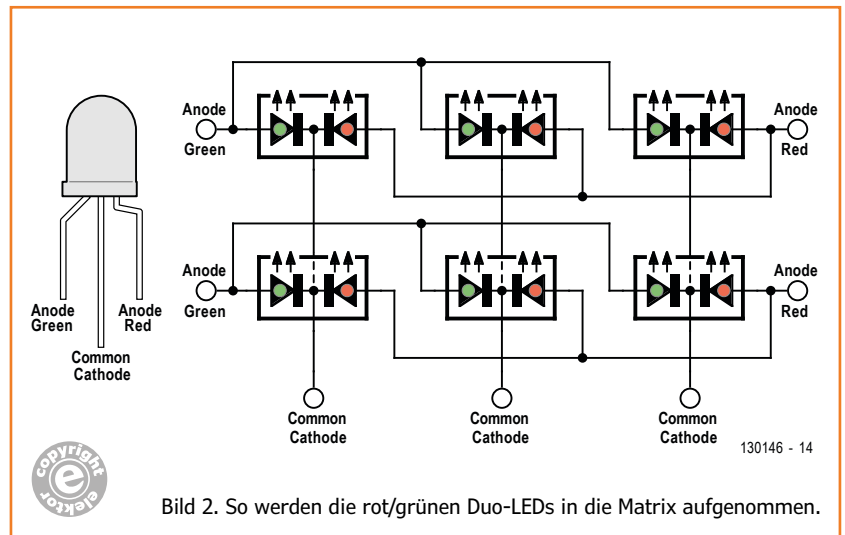


Bild 2. So werden die rot/grünen Duo-LEDs in die Matrix aufgenommen.

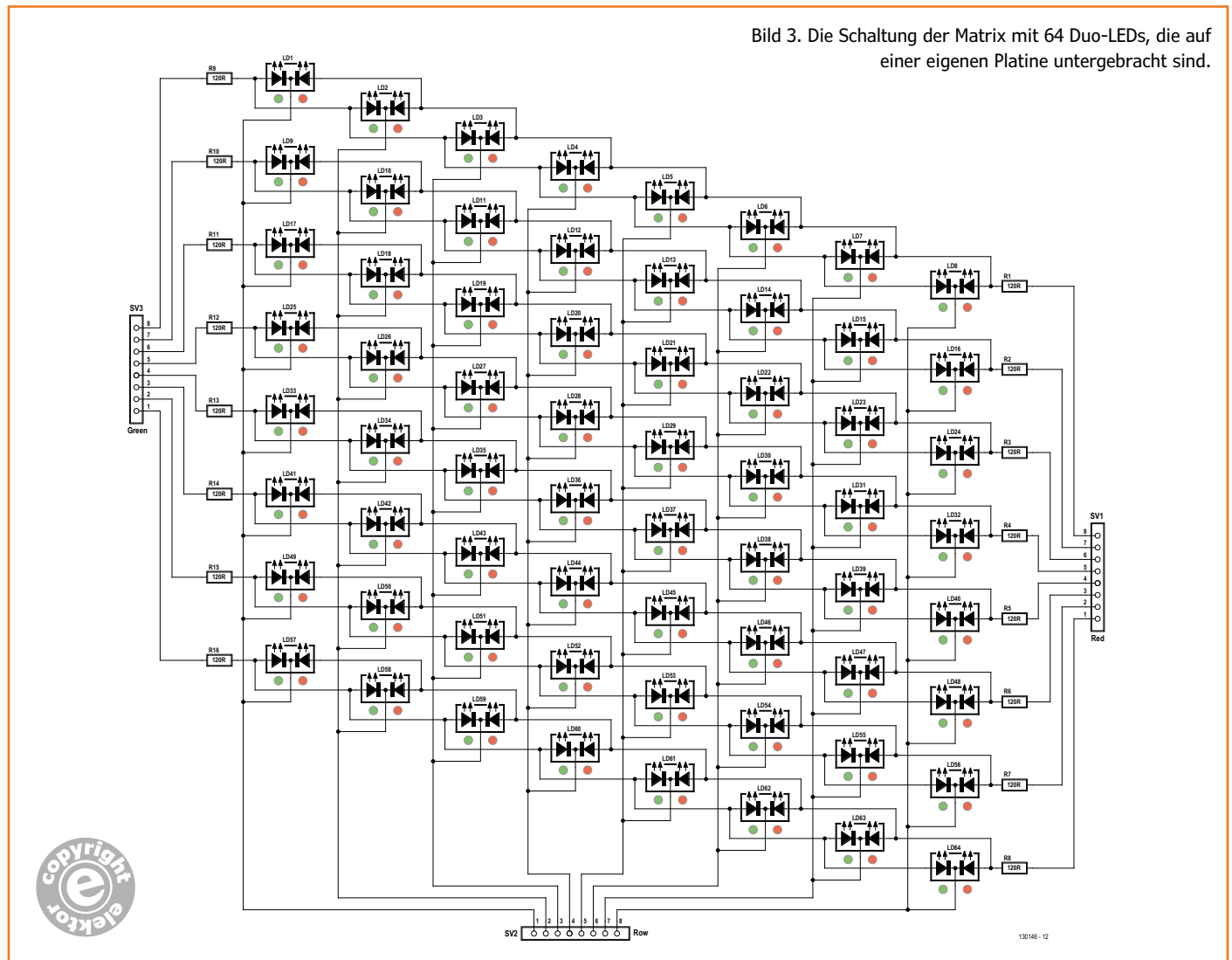
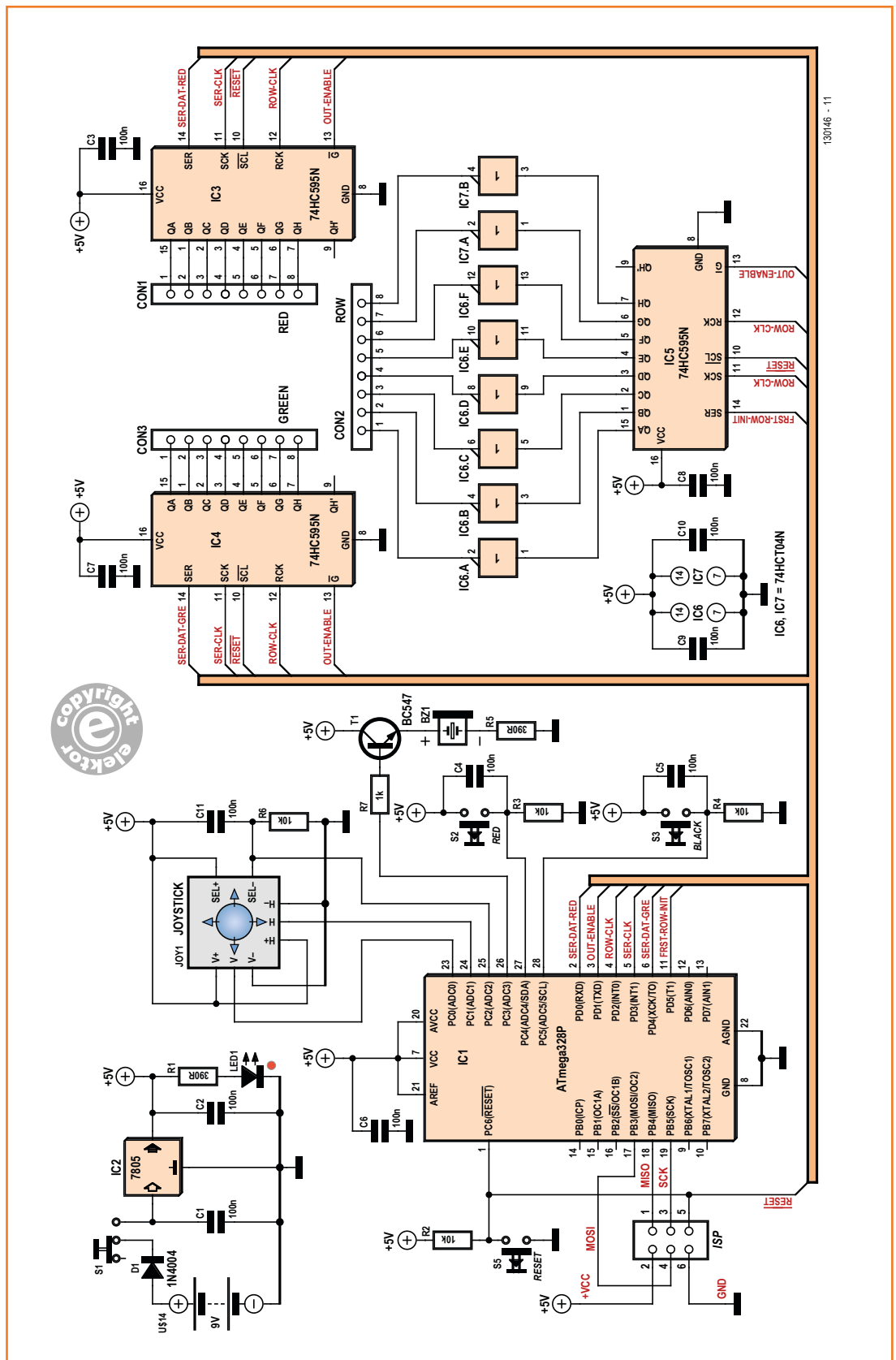


Bild 3. Die Schaltung der Matrix mit 64 Duo-LEDs, die auf einer eigenen Platine untergebracht sind.

Bild 4.
Die Ansteuerelektronik besteht hauptsächlich aus einem ATmega-Mikrocontroller und drei Schieberegistern.



eines für die roten und eines für die gemeinsamen Kathoden (IC3, IC4 und IC5 in **Bild 4**). Mit dieser Anordnung sparen wir uns eine Unmenge von I/O-Leitungen, die nun für andere Zwecke zur Verfügung stehen.

Der Mikrocontroller schickt immer 8 Bits zu den Schieberegistern für die beiden Farben und legt damit die Information für jede Reihe fest. Wenn diese 8 Bits versandt sind, wird ein Impuls zu den Output-Enable-Eingängen der beiden „Farbregister“ und zum Common-Cathode-Schieberegister geschickt, um eine vollständige Zeile aufleuchten zu lassen.

Ein Low bedeutet dabei eine aktive Zeile, ein High eine nicht-aktive Zeile. Das High sorgt dafür, dass keine Spannung über den LEDs liegt und die LEDs demzufolge nicht aufleuchten. Bei aktiven Reihen liegt eine Spannung von 5 V über den angeschlossenen LEDs, so dass selbige leuchten. Dieser Vorgang findet acht Mal pro „Frame“ statt, damit alle Zeilen angesteuert werden. Ein „Frame“ steht in diesem Fall für ein gesamtes Bild mit 8 x 8 roten und grünen LEDs.

Die Frequenz, mit der ein Frame durchlaufen wird, liegt über 60 Hz, damit das langsame menschliche Auge glaubt, alle LEDs würden gleichzeitig aufleuchten (anstelle von Reihe für Reihe). Die Wiederholfrequenz von 60 Hz ist gleichbedeutend mit

Tabelle 1. Port-Anschlüsse

Eingänge:	Mikrocontroller-Port
Schwarze Taste	PC5
Rote Taste	PC4
Joystick horizontale Achse	PC0
Joystick vertikale Achse	PC1
Ausgänge:	Mikrocontroller-Port
Buzzer	PC3
Erste Reihe Impuls	PD5
Serielle Daten rot	PD0
Serielle Daten grün	PD4
Serieller Takt	PD3
Reihen-Taktv	PD2

einer Framelänge von 0,017 s.

Die Zuordnung der Portleitungen zeigt **Tabelle 1**. Dabei ist der Joystick an PC0 und PC1 angeschlossen. Diese Portpins sind nämlich mit einem internen A/D-Wandler verbunden, so dass die Stellung des Sticks leicht analog auszulesen ist. Außerdem besitzt die Schaltung einen Gleichstromsummer, der Piepstöne generiert. Der Summer hat den Vorteil eines eingebauten Oszillators und den Nachteil,

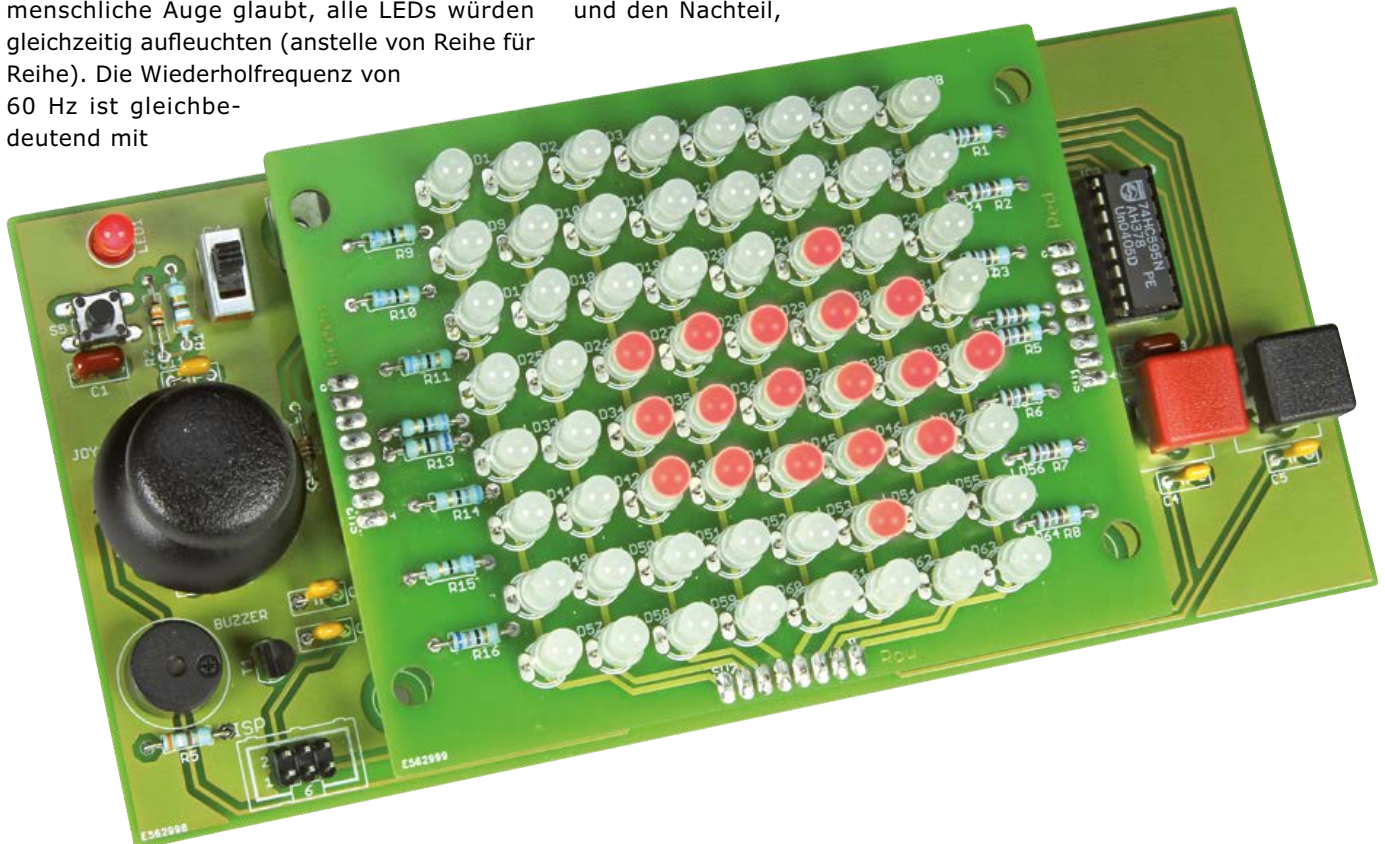
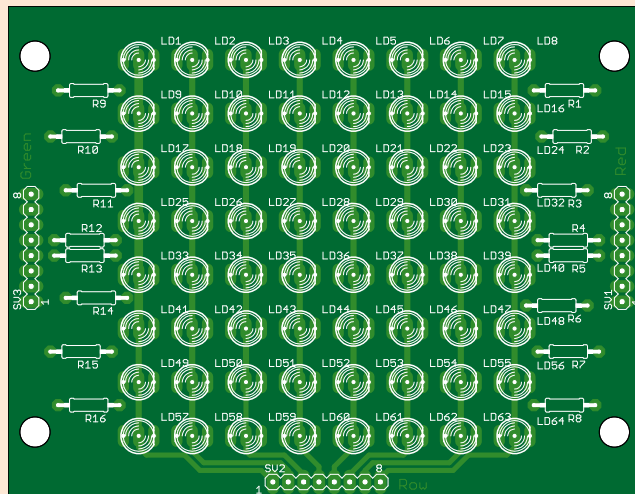
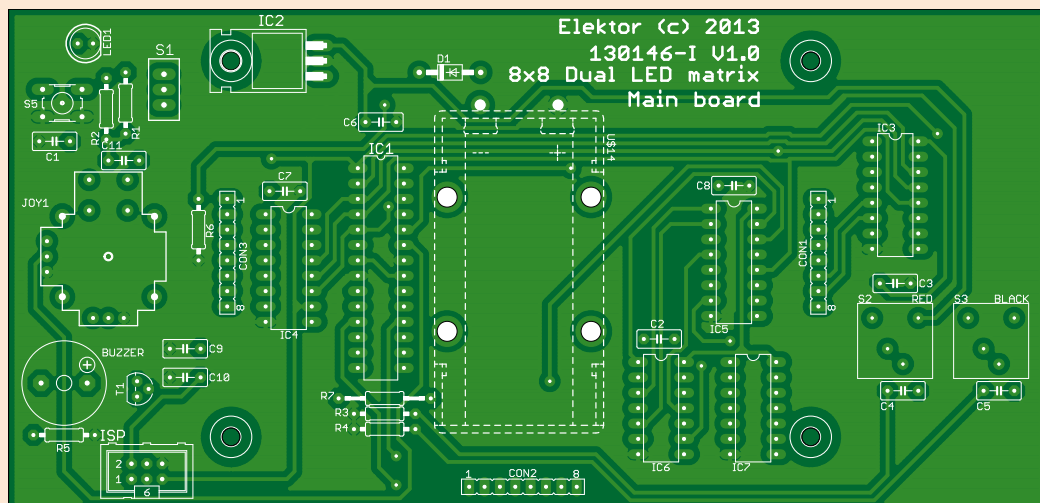


Bild 5.
Die Platine für die LED-Matrix. Achten Sie darauf, dass die LEDs gerade und in der gleichen Höhe montiert werden.



Platine mit 80 % der wahren Größe abgebildet.

Bild 6.
Die Hauptplatine ist so gestaltet, dass die Matrixplatine in der Mitte aufgesteckt werden kann.



Stückliste

Hauptplatine

Widerstände:

R1, R5 = 390 Ω
R2, R3, R4, R6 = 10 k
R7 = 1 k
JOY1 = Joystick mit zwei Achsen, 2x10 k + Feuerknopf

Kondensatoren:

C1...C11 = 100 n, RM 5 mm

Halbleiter:

D1 = 1N4004
LED1 = LED rot, 5 mm
T1 = BC547
IC1 = ATmega328P-PU (programmiert: EPS 130146-41)
IC2 = 7805
IC3, IC4, IC5 = 74HC595
IC6, IC7 = 74HCT04

Außerdem:

Buzzer = Gleichspannungssumme 30 mA/5 V
ISP = 2x3-poliger Pfostenverbinder mit Kragen
S1 = Schiebeschalter mit Arbeitskontakt
S2 = Drucktaste mit Arbeitskontakt, rote Kappe
S3 = Drucktaste mit Arbeitskontakt, schwarze Kappe
S5 = Mini-Drucktaste mit Arbeitskontakt
CON1, CON2, CON3 = 8-polige Stiftleiste
9-V-Blockbatterie mit Batteriehalter
Platine 130146-1, siehe [1]

Display-Platine

R1...R16 = 120 Ω
LD1...LD64 = Duo-LED grün/rot, I_F = 10 mA (z.B. Conrad 156269-89)
SV1, SV2, SV3 = 8-polige Stiftleiste
Platine 130146-2, siehe [1]

dass man seine Frequenz nicht ändern kann. Das Schaltbild zeigt außerdem einen ISP-Anschluss zum schnellen (Re-)Programmieren des Controllers.

Für die Stabilisierung wird ein simpler 7805-Spannungsregler eingesetzt. Die Stromaufnahme inklusive LEDs liegt bei maximal 50 mA, die Eingangsspannung (Steckernetzteil) darf zwischen 7 V_{DC} und 12 V_{DC} liegen. Will man nur ab und an mit der Matrix spielen, reicht auch eine 9-V-Blockbatterie.

Für die Schaltung gibt es zwei separate Platinen, eine für die LED-Matrix (**Bild 5**) und eine für die Steuerschaltung (**Bild 6**). Da ausschließlich bedrahtete Bauteile eingesetzt werden, sind keinerlei Probleme beim Bestücken zu befürchten. Für die ICs verwendet man am besten Sockel. Mit vier Schrauben und Muttern kann auf der Lötseite der großen Platine ein Halter für eine 9-V-Blockbatterie montiert werden. Größte Schwierigkeit ist es, die LEDs alle in Reihe und Glied und auch gleich in der Höhe zu montieren. Als Hilfsmittel kann ein Stückchen Karton dienen, das vor dem Löten zwischen die LED-Beinchen geschoben wird.

Programm

Im Programm gibt es ein paar Stellen, die einer näheren Betrachtung würdig sind: die serielle Datenansteuerung der Matrix, das analoge Lesen des Joysticks und das Spiel, das hier quasi als Demoanwendung genutzt wird.

Vom 74HC595-Schieberegister werden drei Eingänge verwendet, nämlich Serial-Data, Serial-Clock und Output-Enable. Der Serial-Data-Eingang sorgt bei jedem Taktimpuls dafür, dass die aktuellen 8 Bits im Schieberegister um eine Stelle nach oben geschoben werden, so dass das LSB entfällt und der aktuelle logische Pegel am Serial-Data-Eingang als MSB hinzugefügt wird. Wenn nach dem Einlesen von acht neuen Bits der Output-Enable-Eingang aktiviert wird, werden alle Bit-Werte in den Pufferspeicher des Schieberegisters verschoben.

Pro Frame wird ein Array von acht Zeichen verarbeitet und zerlegt, so dass dann (pro Zeile) deutlich ist, welche LEDs leuchten müssen. Das Programm kontrolliert erst, ob es sich um die erste Zeile handelt (in diesem Fall wird ein Impuls zum Common-Cathode-Schieberegister geschickt). Danach wird jedes Bit einer Zeile auf den aktiven Zustand hin überprüft; eine „1“ steht dabei für aktiv und eine „0“ für inaktiv. Wenn ein Bit als aktiv erkannt ist, wird der Serial-Data-Eingang

Listing 1.

```
if(RowNmbr!=0) //check if the current Row is not the first
{
    //Set First row out to high = PD5 (DDRD 0b00100000)
    PORTD&=0b11011111;
}
else if(RowNmbr==0)
{
    //Set First row out to low = PD5 (DDRD 0b00100000)
    PORTD|=0b00100000;
}

//give a row clock pulse
PORTD|=0b00000100;
PORTD&=0b11111011;

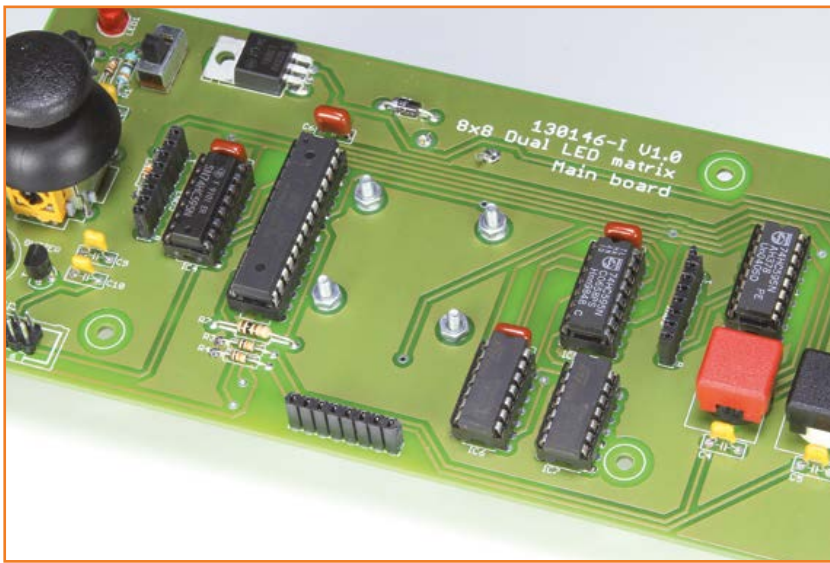
for(I=0;I<8;I++) //repeat the routine 8 times to get all the bits
{

    //get the right bit value by bit shifting the bit
    if(((RedRowData>>(7-I))%2)==1)
    {
        //set the serial Red output high = PD0 (DDRD 0b00000001)
        PORTD|=0b00000001;
    }
    else
    {
        //set the serial Red output low = PD0 (DDRD 0b00000001)
        PORTD&=0b11111110;
    }

    //get the right bit value by bit shifting the bit
    if(((GreenRowData>>(7-I))%2)==1)
    {
        //set the serial Green output high = PD4 (DDRD 0b00010000)
        PORTD|=0b00010000;
    }
    else
    {
        //set the serial Green output low = PD4 (DDRD 0b00010000)
        PORTD&=0b11101111;
    }

    //give a serial clock pulse = PD3 (DDRD 0b00001000)

    PORTD|=0b00001000;
    PORTD&=0b11111011;
}
```



der entsprechenden Farbe zusammen mit dem Serial-Clock-Eingang aktiviert, wenn das Bit dagegen inaktiv ist, wird nur der Serial-Clock-Eingang angesteuert. Ein Teil des Programmcodes der

seriellen Ansteuerung ist in **Listing 1** zu sehen. Zum Einlesen des Stands der analogen Potis des Joysticks werden Routinen aus der `avr/io`-Bibliothek verwendet (AVR Studio 5). Die globalen Variablen für das analoge Auslesen sind in der Header-Datei `program.h` deklariert:

- PortJoyV – Der aktuelle vertikale Wert des Joysticks
- PortJoyH – Der aktuelle horizontale Wert des Joysticks

In `ProgramInit.cpp` ist der Code aus **Listing 2** zu finden.

Die Referenzspannung der A/D-Wandlung ist auf 5 V eingestellt, was der Betriebsspannung A_{VCC} des Controllers entspricht. Der „prescale div factor“ sorgt dafür, dass die Wandlung mit der korrekten Geschwindigkeit stattfindet. Mit dem Code in der Datei `DataIn.cpp` wird anschließend die Stellung pro Joystick-Achse eingelesen.

Das Beispielprogramm für dieses Projekt ist kostenlos unter [1] verfügbar. Es handelt sich um ein Spiel, in dem man die grüne LED mit dem Joystick so steuern muss, dass sie nicht mit der roten LED kollidiert.

Nach einer Intro-Animation startet das Spiel, wenn man auf die rote Taste drückt. Der erste „Level“, der in der Datei `LoadLevel.cpp` zu finden ist, wird dann in ein Array von 117 Zeichen geladen. Dies ist ein Level mit einer Höhe von 8 bit (also der gesamten Höhe der LED-Matrix) und einer Breite von 117 bit. Die gegenwärtige Position der grünen LED wird in einer Charakter-Variablen gespeichert und kann durch Bewegen des Joysticks nach oben oder unten verändert werden. Nach jedem Frame wird die Position der grünen LED bitweise mit der nächsten Zeile des Levels AND-verknüpft. Stimmen diese überein, wird die Schleife beendet und das Spiel beginnt von neuem auf dem letzten gültigen Level. Dazu gibt es eine Testfunktion: Wenn man nach dem Spielen oder direkt nach dem Anlegen der Betriebsspannung den schwarzen Knopf gedrückt hält, kann man mit dem Joystick einen Block von 4 LEDs über die Matrix bewegen. Wir wünschen viel Spaß!

(130146)

Weblink

[1] www.elektor.de/130146

Listing 2.

```
//analog reading setup
ADMUX=0b01000000; // For Aref=AVcc;
ADCSRA=0b10000110; // prescale div factor = 64
```

Listing 3.

```
//Reading the analog signal of the joystick Vertical-Axis

//Reads vertical joystick (port 0)
ADMUX&=0b11111100;

//Start Single conversion
ADCSRA|=0b01000000; //ADSC = high

//Wait for conversion to complete
while(!(ADCSRA&0b00010000));

//Clear ADIF by writing "one" to it
ADCSRA|=(1<<ADIF);
ADCSRA&=0b10111111;

PortJoyV=ADC;
```

UNSER WICHTIGSTES WERKZEUG

LEITERPLATTEN WEITER GEDACHT.



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

LEITON
RECHNEN SIE MIT BESTEM SERVICE

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

www.leiton.de

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

**GLAUBST DU
DASS VOICE-ÜBER IP
ANWENDUNGEN
SCHWIERIG SIND?**



solutions

MIT SOLUTIONS
EBEN WIR KONNTEN
DAS MACHEN



ISL0A11E/SE

Beschallungssystem über IP
basierten schmalbandigen

ISL0A12E/SE

Beschallungssystem über IP
basierten Breitband

ISL0A15E/SE

IP Wechselsprechanlage
P2P für Schmal- und
Breitbandtelekommunikation

Transparent und sicher
verfügbare Lösungen

www.solutions.com



pico
Technology

PicoScope® 6000 Series

- Hochleistungs-PC-Oszilloskope
 - 500 MHz mit 5 GS/s
- Einem branchenführenden
2 Gigasample-Pufferspeicher
- Funktionsgenerator oder
Generator für
anwenderdefinierte Wellenformen
- USB SuperSpeed-Schnittstelle



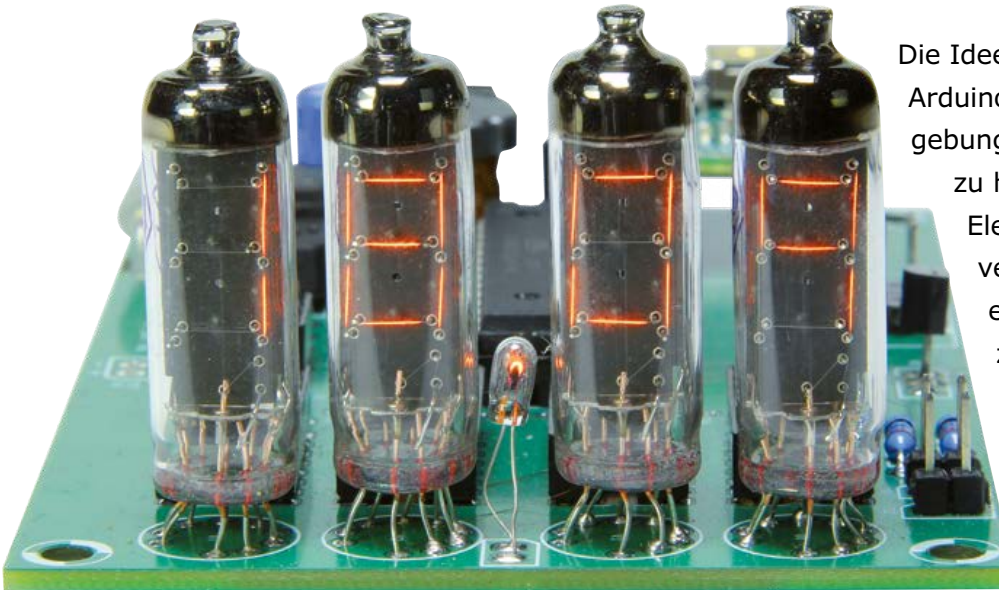
PicoScope	6402C	6402D	6403C	6403D	6404C	6404D
Channels	4					
Bandbreite	250 MHz		350 MHz		500 MHz	
Sampling Rate	5 GS/s					
AWG oder Funktionsgenerator	FG	AWG	FG	AWG	FG	AWG
Speicher	256 MS	512 MS	512 MS	1 GS	1 GS	2 GS
Preis	£1995	£2495	£2995	£3495	£3995	£4495
Kompatibilität	USB 2.0 & 3.0, Windows XP, Windows 7 & Windows 8					

ALLE MODELLE WERDEN MIT TASTKÖPFEN, KOMPLETTER SOFTWARE UND 5 JAHRE GARANTIE DELIEFERT. DIE SOFTWARE ENTHÄLT: MESSUNGEN, SPEKTRUM-ANALYZER, KOMPLETTES SDK, ERWEITERTER TRIGGER, FARB-PERSISTENZ, SERIELLE DEKODIERUNG (CAN, LIN, RS232, I2C, I2S, SPI), MASKEN, MATHEM. KANÄLE - ALLES ALS STANDARD. KOSTENLOSE UPDATES.

www.picotech.com/PS245

Numitron-Uhr/ Thermometer

Entwickelt mit der Arduino-Plattform



Die Idee zu diesem Projekt war es, Arduino aus seiner gewohnten Umgebung von Shields und Breadboards zu holen und einmal mit einem Elektor-typischen DIY-Projekt zu verbinden. So konnten wir auch eingefleischten AVR-Hackern zeigen, dass diese scheinbar so verschiedenen Welten gut zusammenpassen. Das Projekt vereint heutige Mikrocontroller-Technik mit Röhren aus der Sowjetunion der 50er Jahre.

Von **Paul Court** (UK)

Eine Uhr mit traditionellen 7-Segment-LED-Anzeigen ist ja doch etwas langweilig. Viel interessanter ist es, wenn man zur Ziffernanzeige Numitron-Röhren aus der Sowjetunion verwendet. Numitrons sind Verwandte der Nixie-Röhren, die schon mehrfach in Elektor-Projekten verwendet wurden, laufen aber mit etwa 4 V anstatt mehr als 150 V und eignen sich daher besser für Experimente. Genau wie die Nixies sind Numitrons recht

leicht zum Beispiel über eBay zu sehr günstigen Preisen erhältlich.

Die Numitron-Uhr/Thermometer ist auf einer einzigen Platine aufgebaut. Das Projekt *basiert* auf Arduino UNO, das heißt, das UNO-Board und die IDE wurden nur als Entwicklungs-Tools benutzt. Im fertigen Projekt gibt es aber weder Shields noch Module noch PlugIns - nur einen ATmega328 mit peripherer Hardware.

Eigenschaften

- Vier IV-9 Numitron-Röhren
- Anzeige Datum/Zeit/Jahr/Sekunden/Temperatur
- Mikrocontroller ATmega328
- Entwickelt auf Arduino-Plattform
- Einstellbarer Schaltregler für optimale Numitron-Helligkeit
- Stromversorgung über Steckernetzteil 9...12 VDC, 500 mA
- Keine Hochspannung
- Entwicklungssoftware komplett kostenlos zugänglich
- Design entwickelt durch Elektor.Labs

Und noch ein „tron“: Numitron!

Dekatron, Klystron, Thyatron, Magnetron, Trochotron, was kommt als nächstes? Man landet unweigerlich auf www.radiomuseum.org, wenn man sich für dieses rätselhafte Suffix „tron“ aus der dunklen Vor-Internet-Zeit interessiert. Man muss weder das große Latinum oder Graecum besitzen, um zu ergründen, dass *Numi*-tron etwas mit *Nummern* zu tun haben muss. Und tatsächlich, eine Numitron kann die Ziffern 0 bis 9 und ein paar Buchstaben mit sieben diskreten Segmenten (Glühfäden, Filament) darstellen. Bei

vielen Numitrons gibt es eine Art Komma (decimal symbol, ds) als achttes Segment in der rechten unteren Ecke. Die Segmente besitzen eine gemeinsame positive Versorgungsleitung und ihre anderen Enden sind als Drähte und nicht als Stifte herausgeführt. Wie viele Nixies sind Numitrons Draht-Röhren.

Obwohl Kinder des Kalten Krieges, verbreiten Numitrons ausgesprochen warmes Licht. Sie können auch nach Jahrzehnten noch NIB (new in box) aka NOS (new old stock) von russischen und vor allem ukrainischen Internet-Händlern erworben werden. Der Numitron-Typ IV-9, den unser Entwickler Luc Lemmens für den Aufbau im Elektor-Lab auswählte, wurde über Ebay in Russland bestellt. Sie erreichten ihn auch wohlbehalten trotz der sehr rudimentären und auch amüsanten Verpackungstechnologie des mit fünf Sternen bewerteten Händlers (wie das Foto in **Bild 1**, das unser Retronik-Redakteur geschossen hat, beweist).

Die wichtigsten technischen Spezifikationen des IV-9-Numitron sind in **Tabelle 1** angegeben. Wie wir bereits beim Nixie-VU-Meter mit der IN-9 [1] be- und anmerkten, sind die „Design-Daten“ dieser UdSSR-Röhren (1) stark interpretationsbedüftig (2) sehr (sehr!) tolerant und (3) in kyrillisch.

Schaltungsbeschreibung

Bei einem Blick auf die Schaltung in **Bild 2** kann man schon staunen, dass die Numitrons nicht gemultiplext werden, sondern jede der vier Digits der Uhr einen eigenen Treiber SN74LS47D besitzt. Es ist nämlich unmöglich, die IV-9 zu multiplexen, da die Wolfram-Glühfäden viel zu träge sind und entweder flackern oder zu schwach leuchten würden.

Jede Röhre besitzt also einen eigenen Treiber und vier BCD-Treiberleitungen mit Ausnahme vom höchsten Digit V1, das nur zwei benötigt. Bei der Verwendung von vier 74LS47 ohne Latch oder Multiplexer werden natürlich mehr Pins des ATmega „verbraucht“. Jede Röhre hat ihre eigenen BCD-Leitungen und kann so direkt durch Toggeln der Controlleranschlüsse aktiviert werden. Die Schaltung verfügt über zwei Versorgungsspannungen, +5 V von einem gewöhnlichen 78L05-Regler (IC1) und 4,00 V (einstellbar mit P1) von einem kleinen Switch-Mode-Regler rund um den bekannten MC34063 (IC2). Der mögliche Bereich der Glühfadenspannung der IV-9 liegt bei

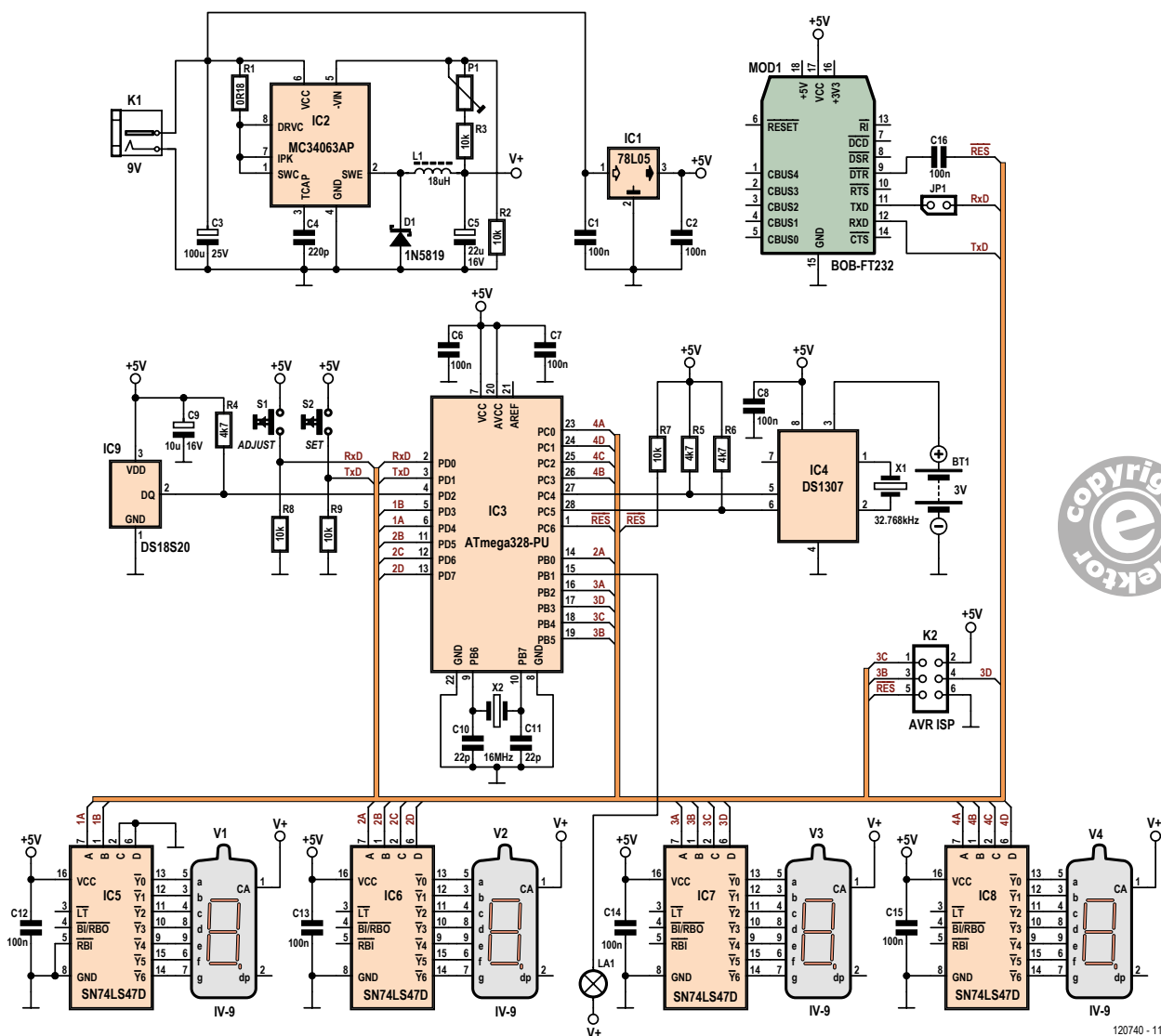


Bild 1.
Aber hallo, da hat uns eine Reihe von Numitrons aus dem Dot-ua-Land erreicht. Zusammen mit einem kostenlosen guten Buch. Bezahlt in US-\$ mit Kreditkarte und vom Zoll abgefertigt kamen die kleinen Röhrrchen den ganzen Weg aus der Ukraine zu uns.

Bild 2.
Die Schaltung der Numitron-
Uhr/Thermometer. Beachten
Sie, dass jede der Numitrons
V1...V4 individuell und nicht
gemultiplex angesteuert
wird.

Tabelle 1. Wichtigste Daten der IV-9

	Typ	IV-9 (IB-9)
	Marke	Reflector/ Sovtec
	Ersatztyp	keiner
	Angezeigte Symbole	7 Segmente, ds
	Symbol-Höhe	10 mm (0,400 ")
	Abmessungen	11 mm (0,433 ") Ø. x 35 mm (1,38 ")
	Anschlussdraht-Durchmesser	0,5 mm (0,020 ")
	Typ. (max.?) Versorgungsspannung	4,5 V
	Typ. Strom pro Segment	19,5 mA
	Basis	TO-100, Pin 1 vorne
	Fassung	Njet. Anschlussdrähte, keine Fassung erforderlich



copyright
elektor

120740 - 11

3,15...4,50 V und Sie können die Helligkeit mit P1 so einstellen, wie es Ihnen am besten passt. Jedes Filament zieht etwa 19,5 mA, die ganze Uhr bei einer unregelmäßigen Eingangsspannung von 7,5 V circa 500 mA.

Die internen Versorgungsspannungen wurden auf 5 V und 4 V (3,15...4,50 V) und nicht auf 3,3 V festgelegt, weil

die höhere Spannung es dem Controller erlaubt, auf 16 MHz zu arbeiten;

3,3 V als V+ zu niedrig ist, da die Open-collector-Ausgänge der 74LS47D nicht ganz auf 0 V gehen, wenn sie aktiv sind.

Das Herzstück der Uhr ist ein ATmega328P, wie er auch in der UNO- und der Duemilanove-Reihe des Arduino verwendet wird. Wenn Sie einen Controller mit vorinstalliertem Arduino-Bootloader verwenden möchten, sollten Sie die Duemilanove-Version bevorzugen. Die Programmierung des ATmega328P ist möglich über den 6-poligen Standard-ISP-Header und das AVR-Studio oder über den optionalen Elektor-FT232-BOB (USB) und die Arduino-IDE-Software (V1.0 oder höher) mit dem vorprogrammierten Duemilanove-Bootloader. Die Arduino IDE weiß, dass sie ein Standard „Arduino Duemilanove ATmega328“-Board anspricht.

Die Echtzeituhr (RTC) ist ein DS1307, der Temperatur-Sensor ein DS18B20. Für beide Bauteile gibt es viel Unterstützung in Arduino- und AVR-Communities. Ihr „One-Wire“-Datenformat wird vom Controller über die beiden letzten verfügbaren Portpins PD2 (Temperatur) und PC4 (RTC) gelesen.

Die ISP-Schnittstelle ist K2, der ATmega-Chip kann hier programmiert werden, ohne dass man ihn dafür aus der Schaltung holen müsste (In-System-Programmierung). MOD1 (Elektor-BOB-FT232) ist optional. Der Zusatz bietet USB-Konnektivität und ist recht nützlich, wenn Sie an der Uhren-Software innerhalb Ihrer Arduino-Umgebung arbeiten wollen.

Die Software

Quell- und Hex-Dateien für den ATmega328 stehen auf der Elektor-Website [1] zur Verfügung. Bereits programmierte ICs sind ebenfalls erhältlich (Elektor 120740-41). Der Code besteht aus einer Schleife, die sich alle 500 ms ($\frac{1}{2}$ s) wiederholt. Das Indikator-Lämpchen La1 blinkt entsprechend einmal pro Sekunde. Die Verzögerung gibt der RTC Zeit, sich vom Auslesen zu erho-

len. Gewährt man der Echtzeituhr nämlich diese Erholzeit nicht, dann erhält man nur Datenmüll. La1 ist übrigens ein Miniatur-Glühbirnchen, wie es in Puppenhäusern oder Modelleisenbahnen verwendet wird.

Zu Beginn jedes Zyklus überprüft der Code, ob die SET-Taste (S2) gedrückt ist. Wenn nicht, schaltet er die Indikator-Lampe ein. Dann wird die RTC gelesen, der Wert für Stunden und Minuten in Zehner und Einer umgerechnet und an die richtige Numitron gesendet.

Zur Vereinfachung des Codes besitzt jede Ziffer eine eigene Routine. Alle diese Unterprogramme arbeiten ähnlich; mit Ausnahme der Zehnerstunden, die nur zwei binäre Bits kennt. Mit einem Rechen-Trick (bitweise AND) wandeln wir dann die Dezimalzahl in eine binäre passend für die 74LS47 um. Mehr Informationen über diese bitweisen Operationen sind unter [2] zu finden. Auch wenn es ein wenig langatmig wirkt, ist es eigentlich sehr einfach und eine wirklich nützliche Methode, um Dezimalstellen in einzelne binäre Digits zu konvertieren.

Wir haben zwei zusätzliche interessante Funktionen integriert: Zum ersten wird alle 10 s und 30 s der Temperatursensor ausgelesen und der Wert in Grad Celsius dargestellt. Wie oben beschrieben werden auch hier mod- und bitweise Rechenoperationen zum Lesen, Umrechnen in binäre Werte und Darstellen über die 74LS47 verwendet. Ausnahme ist hier die niedrigste Zahl, bei der eine dezimale „10“ zum korrespondierenden 74LS47 (V4) gesendet wird, so dass ein „c“ für Celsius im Display erscheint. Die Anzeige lautet dann beispielsweise „22.5c“.

Das zweite Gimmick, die showdate()-Routine, wird alle 50 s aufgerufen. Dann nämlich wird nacheinander das Datum, der Monat und das Jahr angezeigt. Der Code ist sehr einfach und ich bin sicher, dass Sie andere Funktionen für die Uhr (Eieruhr?) erfinden können. Also ran an das Projekt! Zeigen Sie uns Ihren Code, Updates, Tweaks oder öffentlichen Sie Ihre Ideen und Anregungen bei Elektor.Labs [4].

Aufbau und Einsatz

Das SMD-freie Projekt sollte ohne Probleme zu Hause, in der Schule oder nebenbei im Labor aufgebaut werden können. Die Platine in **Bild 3** enthält alle in der Stückliste genannten Bauteile. **Bild 4** zeigt die komplett bestückte, sauber aufgebaute Platine. Nach dem Einschalten sollte

Stückliste

Widerstände

R1 = 0Ω18, 3 W
R2,R3,R7...R9 = 10 k
R4...R6 = 4k7
P1 = 4k7 Trimpoti, Einstellung von oben

Kondensatoren

C1,C2,C6...C8,C12...C16 = 100 n
C3 = 100 µ, 25 V radial

C4 = 220 p
C5 = 22 µ, 25 V radial
C9 = 10 µ, 50 V radial
C10,C11 = 22 p

Induktivität

L1 = 18 µ Drossel, 3,4 A, 0Ω036 (Panasonic Typ ELC10D180E)

Halbleiter

D1 = 1N5819
IC1 = 78L05
IC2 = MC34063
IC3 = ATmega328-PU, programmiert, Elektor 120740-41 [2]
IC4 = DS1307
IC5,IC6,IC7,IC8 = 74LS47
IC9 = DS1820

Außerdem

V1...V4 = IV-9 Numitron-Röhre
JP1 = 2-polige Stiftleiste mit Jumper, 0,1 ''
X1 = 32,768 kHz Quarz
X2 = 16 MHz Quarz
MOD1 = BOB-FT232R (optional), Elektor 110553-91
Bt1 = Batterie CR2032
S1,S2 = Drucktaster, SPST, für Chassismontage
K1 = Klinkenbuchse für Platinenmontage, 2,1 mm
K2 = 2x3-polige Stiftleiste mit Kragen
La1 = 5 V 300 mW, Subminiatur-Glühlampe
Platine 120740-1 [2]

Bild 3.
Die doppelseitige Platine, die vom Elektor-Labor für dieses Projekt entworfen wurde.

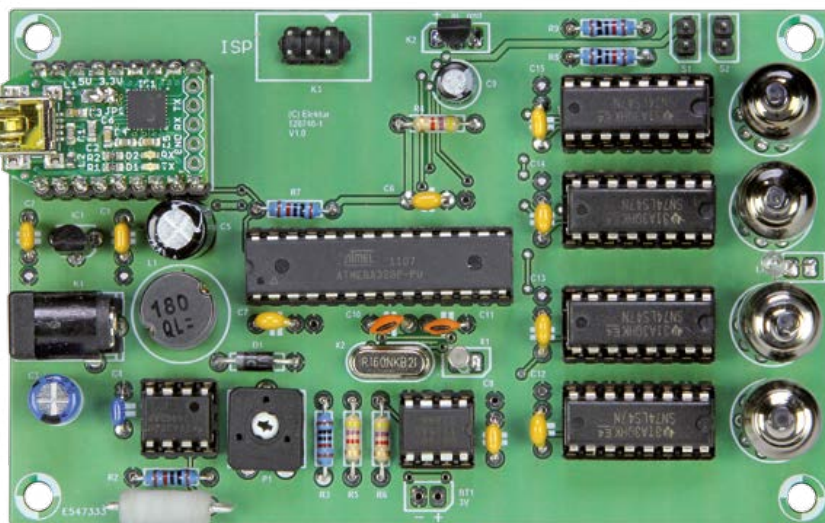
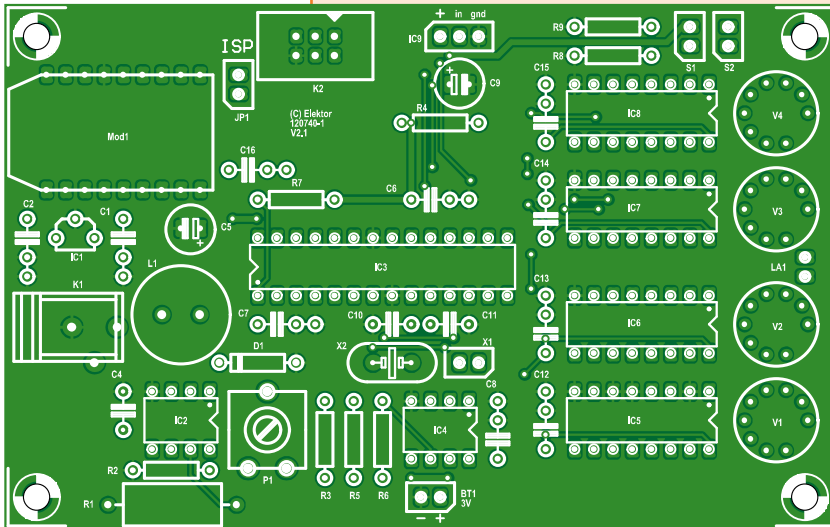


Bild 4.
Die aufgebaute und getestete Schaltung ist klar zum Einsatz!

man zunächst Trimpoti P1 so einstellen, das an V+ genau 4,00 V liegen.
Um die Uhr einzustellen, muss Jumper JP1 entfernt (Run-Modus) werden. Ein Druck auf die SET-Taste S2 ruft die setclock()-Routine auf. Zunächst werden die Minuten angezeigt. Jeder Druck auf

die ADJUST-Taste S1 erhöht die Anzeige um 1. Hält man die Taste gedrückt, erhöht sich der Wert im Halbsekundentakt automatisch. Kleine Verzögerungen im Code entprellen die Tasten. Sobald die Minuten korrekt eingestellt sind, drücken Sie auf SET und die Stundeneinstellung ist gewählt, danach folgen der Tag, der Monat und schließlich das Jahr. Nach Abschluss dieses Zyklus werden die neuen Werte zur RTC geschrieben und der Code kehrt zur Haupt-Schleife zurück.

(120740)

Webinks

- [1] Nixie VU Meter, Elektor November 2012, www.elektor.de/110744
- [2] www.elektor.de/120740
- [3] http://playground.arduino.cc/Code/BitMath#bitwise_and
- [4] www.elektor-projects.com/project/numitron-arduino-clock-and-thermometer-120740.12460.html

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

Am 12. Oktober 2013 findet im Congress Park Hanau die zweite 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:

- Einführung in LabVIEW
- Elektronik-Apps mit Android
- Raspberry Pi-Anwendungen
- Röhrenverstärker (mit Menno van der Veen)
- Einstieg in die AVR-Programmierung
- Layout-Design mit EAGLE 6
- ARM Cortex-M3 Mikrocontroller
- Entwicklung eines FPGA-Erweiterungsboards
- Praxisprojekte mit Arduino
- Rapid SMD-Prototyping

Datum:

Samstag, 12. Oktober 2013

Zeit:

09:00 Uhr – 17:00 Uhr

Ort:

Congress Park Hanau

Eintritt:

29,50 € für Elektor-Mitglieder

49,50 € für Nicht-Mitglieder

19,50 € für Schüler/Studenten

Der Eintritt berechtigt zur Teilnahme an 2 Seminaren.

Tagesablauf:

09:00 Uhr – 17:00 Uhr: Ausstellung

09:30 Uhr – 11:30 Uhr: 1. Seminar-Runde

12:30 Uhr – 14:00 Uhr: 2. Seminar-Runde

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



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



Wasserpumpensteuerung Für Yachten, Wohnwagen etc.



Von **Paul Cordonnier**
(BE)

Wer schon längere Zeit auf See verbracht oder längere Liegezeiten in Häfen erlebt hat, der hat sicher schon mit den Widrigkeiten der Pumpen zur Wasserverteilung oder deshalb gar mit freundlichen Handwerkern zu tun gehabt.

Im Prinzip besteht die Wasserinstallation an Bord aus Wasserleitungen plus einer mit 12 V betriebenen Pumpe, die gut und gerne 5...10 A Strom zieht. Diese Pumpe wird üblicherweise im digitalen An/Aus-Modus zusammen mit einem Druckkessel betrieben. Letzterer sorgt durch den Druckausgleich für einen konstanteren Wasserdurchfluss. Das Ganze bringt Komfort, aber auch einige Probleme mit sich. Man denke nur einmal an den Platzbedarf und die notwendige Wartung. Neuere Pumpenmodelle können diesen Druckausgleichsbehälter überflüssig machen. Etwas Elektronik sorgt dafür, dass die Pumpe genau passend zum Wasserbedarf angesteuert wird.

Das hat fast nur Vorteile: Kein Kessel, die Pumpe läuft leichter, braucht weniger Energie und der Wasserdurchfluss wird sanft geregelt. Das „fast“ bezieht sich auf die Kosten. Die neuartigen Pumpen mit ins Gehäuse integrierter Elektronik sind praktisch doppelt so teuer wie die durch sie ersetzten einfacheren Varianten.

Der Autor ist zwar kein Elektronik-Ingenieur, doch dafür ein Elektroniker aus Leidenschaft. Er suchte also nach einer einfachen und preiswerteren Lösung – und fand sie auch. Jetzt teilt er sie mit den Elektor-Lesern ;-).

Zunächst zu S1 in der Schaltung von **Bild 1**: Auch wenn er aussieht wie ein gewöhnlicher Schalter, ist es doch ein Druckschalter, der dann **ausschaltet**, wenn der Druck einer Flüssigkeit einen bestimmten Wert übersteigt (oft ist solch ein Schalter Bestandteil des Pumpen-Systems). Sein Job in dieser Schaltung ist, den Motor samt Elektronik zu aktivieren, wenn der Druck fällt.

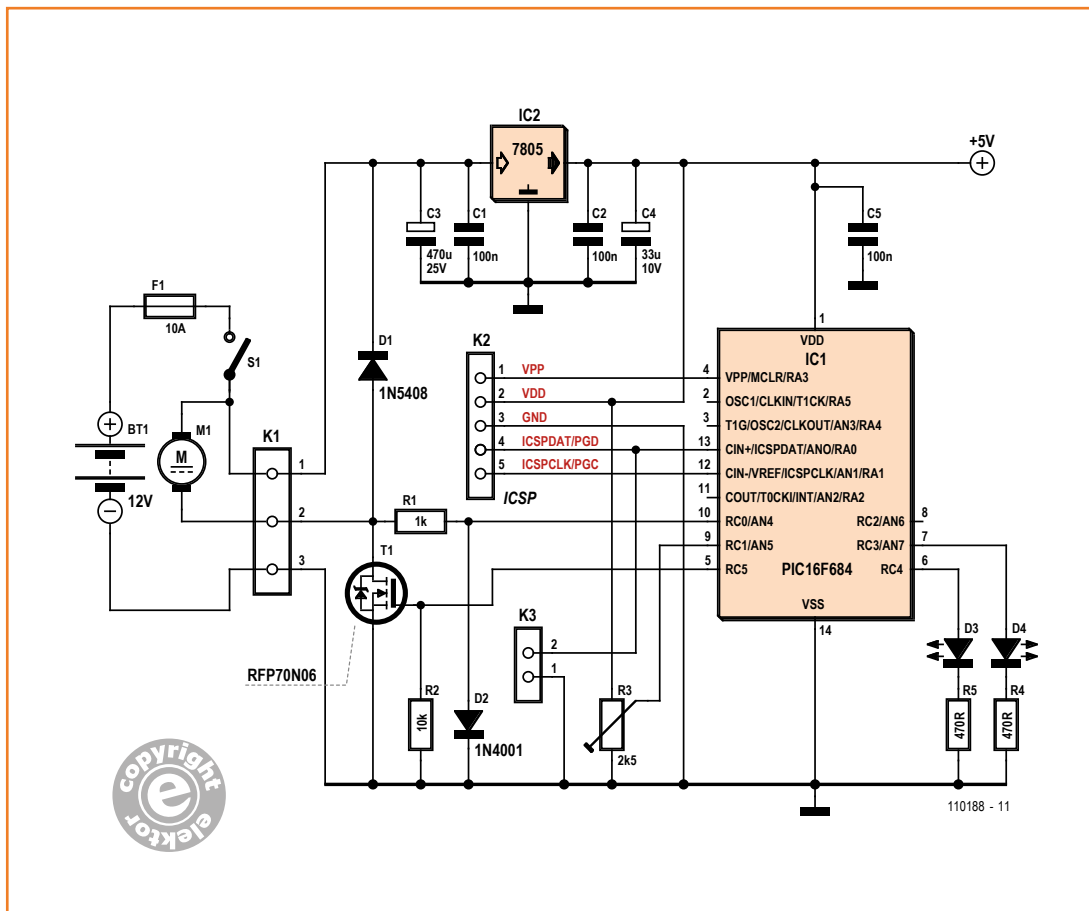


Bild 1.
Der Druck, bei dem S1
auslöst, ist an einigen
Pumpen-Modellen mittels
einer Schraube einzustellen.

Wenn der Druck wieder erreicht wurde, soll er alles wieder deaktivieren.

Die Pumpe M1 wird von IC1 mit Hilfe des MOSFETs T1 angesteuert. Das Gate von T1 wird hierzu vom Mikrocontroller mit einem PWM-Signal mit 1 kHz Takt gefüttert. T1 ist in Serie mit der negativen Leitung der Pumpe geschaltet. Der PIC bzw. ein darin enthaltener Komparator vergleicht die am Drain/Source-Widerstand abfallende Spannung mit einem Referenzwert (einstellbar mit Trimmer R3) und steuert die PWM so, dass die Spannung an Pin 10 (nichtinvertierender Eingang) möglichst gleich wie die Referenz an Pin 9 (invertierender Eingang) wird. Steigt der Strom durch den Motor, wird das Tastverhältnis reduziert und umgekehrt. Der PWM-Bereich ist zwischen 55 % (= 140/255) und 98 % (= 250/255) limitiert.

Für Debugging-Zwecke schickt der PIC den aktuellen Wert der PWM direkt über eine serielle Schnittstelle mit 2400 Baud in die Welt hinaus. Wenn kein Wasser entnommen wird, dann steigt der Druck bis zu einer bestimmten Schwelle, bei

der S1 geöffnet wird. Pumpe und Elektronik werden dann stromlos. D2 begrenzt zusammen mit dem Vorwiderstand R1 die hohen Spannungen, um den Eingang des PICs zu schützen. LED D3 wird vom im PIC enthaltenen Komparator und D4 von der Software gesteuert. K3 ist zum Debuggen gedacht und kann auch weggelassen werden.

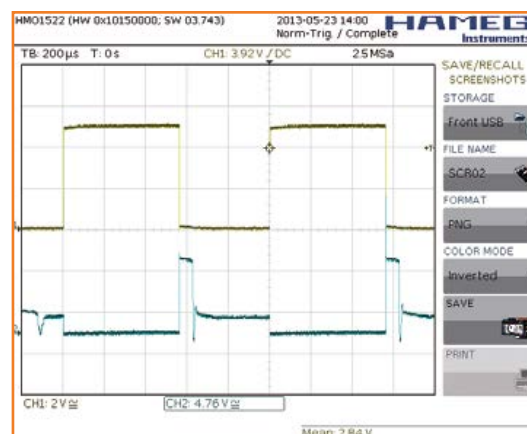


Bild 2.
Signal an Gate (gelb) und
Drain (blau) von T1.



Stückliste

Widerstände:

(0,25 W)
 R1 = 1 k
 R2 = 10 k
 R3 = 2k5, Trimpoti, groß,
 liegend
 R4,R5 = 470 Ω

Kondensatoren:

C1, C2 = 100 n, RM 2,5 mm
 C5 = 100 n, RM 5 mm
 C4 = 33 μ /10 V, radial, RM
 2 mm
 C3 = 470 μ /25 V, radial, RM
 5 mm

Halbleiter:

D1 = 1N5408
 D2 = 1N4001
 D3,D4 = LED rot, 5 mm
 T1 = RFP70N06, N-MOSFET
 70 A 60 V
 IC1 = PIC16F684-I/P, programmiert erhältlich*
 IC2 = 7805

Außerdem:

K1 = 3-pol. Klemme für Platinenmontage, RM 2/10"
 K2 = Stiftleiste 1x5, RM 1/10"

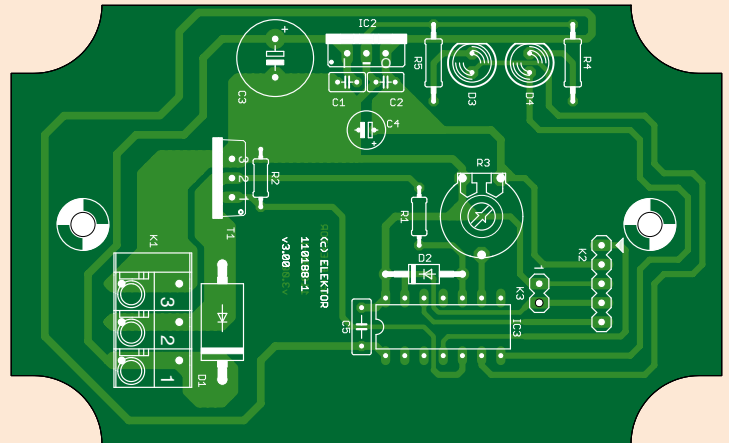


Bild 3. Das spezielle Layout der Platine ist auf ein wasserdichtes Gehäuse zugeschnitten

K3 = nicht bestückt*
 14-polige DIL-IC-Fassung
 Verwendete Pumpe: Jabsco PAR-MAX 7,2 l/min,
 12 V/3 A
 Platine 110188-1

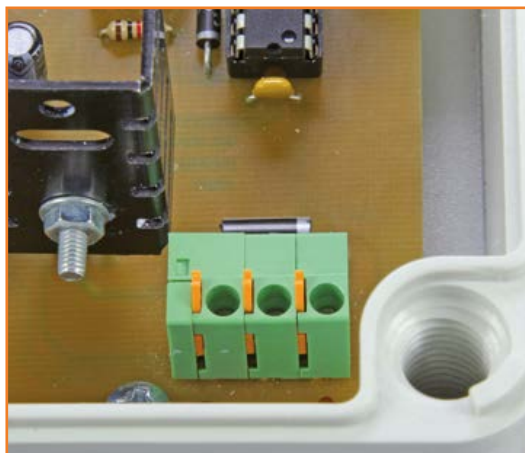
* siehe Text

Die Steuerung wird von einem in PICBasic Pro geschriebenen Programm erledigt. Die Software ist sehr einfach gehalten, da der PIC ja nur diese eine Aufgabe hat. Der Source-Code kann wie immer kostenlos von der zugehörigen Elektor-Webseite [1] herunter geladen werden. Neben einer Platine ist auch ein fertig programmierter PIC über den Elektor-Service erhältlich. Das Einschleifen der Regelung in die negative Versorgungsleitung einer Pumpe sollte bei kaum

einer Installation Probleme bereiten. Worauf man allerdings achten sollte ist, dass man es hier mit einer potentiell feuchten Umgebung zu tun hat. Das erklärt auch die spezielle Form der Platine. Sie ist nämlich passend für ein spezielles Gehäuse (PICCOLO ABS B 65 von Fibox) maßgeschneidert worden, das einen ausreichenden Feuchtigkeitsschutz (IP 66/67) bietet: Der Inhalt ist gegen einen Wasserstrahl und sogar gegen Eintauchen bis zu 1 m Tiefe geschützt.

Die vom Autor verwendete Pumpe fördert 7,2 l/min bei 3 A an 12 V. Ein richtiger Seemann bevorzugt für K1 übrigens Federkraftklemmen statt der üblichen Schraubklemmen, da letztere sich bei Vibrationen lösen können. Ahoi!

(110188)



Weblink

[1] www.elektor.de/110188

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: 34,90€
zzgl. Ust./Versand

Schaeffer AG · Nahmitzer Damm 32 · D-12277 Berlin · Tel +49 (0)30 8058695-0
Fax +49 (0)30 8058695-33 · Web info@schaeffer-ag.de · www.schaeffer-ag.de

IOcP4IoT

Internet Offload co-Processor
für das Internet-der-Dinge



TCP/IP
+ MAC
+ PHY

W7200-S2E



2x UART
Max. 921 kb/s
TTL out für RS232
und RS422 / RS485

W7100A-S2E



WIZ107SR

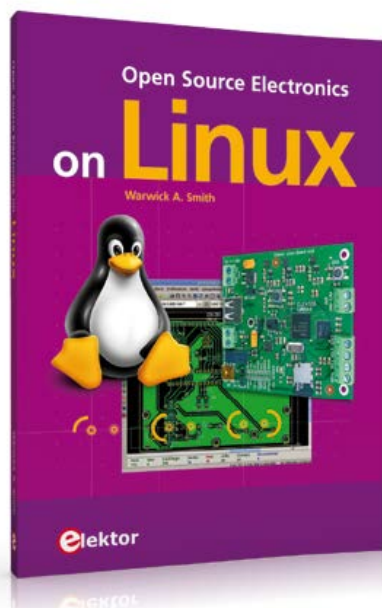
Serial-zu-Ethernet
Chips & Module

1x UART to Ethernet
max. 230 kb/s
TTL und RS232
oder RS422 und RS485

WIZnet Europe GmbH
Ludwig-Erhard-Str.30-34 / 65760 Eschborn
www.wiznet.eu +49-6196-9540-270 info@wiznet.eu

Open Source Electronics on Linux

Neues Fachbuch in englischer Originalsprache!



Wenn Sie sich schon immer einmal mit dem stetig wachsenden Angebot an Open-Source-Software für Elektronik befassen wollten, dann ist dieses Buch genau das, was Sie suchen. Der Autor Warwick A. Smith führt Sie unter Einsatz von Linux durch die Welt der Open Source Hard- und Software. Er erläutert, wie man EDA-Tools und Software nutzt, die kostenlos ist und im Internet auf Ihren Download wartet.

Dieses neue englischsprachige Buch zeigt, wie sich Elektroniker zu Ingenieuren entwickeln können, die ihre Projekte mit aktueller Technik entwickeln und bauen. Dies ist sowohl rein unter Linux als auch in Kombination mit anderen Betriebssystemen möglich.

Alle im Buch beschriebenen Hardware-Projekte können auf einseitigen Platinen aufgebaut werden, die man auch selbst herstellen kann. Man kann aber auch mit den entsprechenden Layout-Dateien bei einem Dienstleister professionelle Platinen fertigen lassen. Es sind keine speziellen oder schwer erhältlichen Bauteile notwendig.

271 Seiten (kart.) • Format 17 x 23,5 cm • ISBN 978-1-907920-19-6

€ 34,50 • CHF 42,80

Starten mit dem LPC800 Mini Kit!



Kurz vor dem Sommer bot Elektor.Labs seinen Mitgliedern die Möglichkeit, ein kleines Mikrocontroller-Board kostenlos zu erhalten. Das Board war ein LPC800 Mini-Kit mit einem Mikrocontroller LPC810 32-bit ARM Cortex-M0+ in einem 8-poligen DIP-Gehäuse, einem Spannungsregler, zwei Tasten, einer LED und zwei kleinen Prototyping-Bereichen. Die Kampagne war ein großer Erfolg, die Boards waren nach etwa 45 Minuten weg...

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

Wenn Sie unter den Glücklichen waren, die eines dieser Boards ergattern konnten, hatten Sie den ganzen Sommer Zeit, damit rumzuexperimentieren. Doch wir wissen, wie das so läuft: Sobald Sie das Kit erhalten, öffnen Sie die Box, werfen einen Blick darauf, schließen es am PC an, um die LED blinken zu sehen und dann legen Sie es „für später“ auf Ihren Schreibtisch. Draußen lacht die Sonne und drinnen gerät das Kit in Vergessenheit. Damit das Kit nicht endgültig in der Versenkung verschwindet, hier eine kurze Anleitung zum Einstieg mit diesem Kit. Um mit dem Mini-Kit zu spielen, hat man folgende Dinge nötig:

- Ein Mini-Kit;
- Einen PC mit einer 3,3-V-Logikpegel-kompatiblen seriellen Schnittstelle;
- Eine 5-V-Stromversorgung;
- Einen Serial-Wire-Debugger (SWD) (optional).

Den ersten Punkt können Sie natürlich schon mal abhaken, für den zweiten benötigen Sie entweder einen PC, an den ein MAX3232-Adapter, betrieben mit 3,3 V, am seriellen Port angeschlossen ist oder viel einfacher, weil es auch den dritten Punkt abdeckt, ein 3,3-V-FTDI-Kabel (Elektor 080213-

72). Sie können auch unsere USB/Seriell-Brücke „BOB“ (Elektor 110553-91) verwenden, die flexibler ist, aber es braucht einen heißen LötKolben, weil Sie den Lötjumper auf die 3,3-V-Position setzen müssen. Sowohl das FTDI-Kabel als auch das BOB liefert die 5-V-Spannung, um das Mini-Kit zu versorgen.

Das Mini-Kit besitzt einen Verbinder, der mit dem 3,3-V-FTDI-Kabel kompatibel ist, ich aber bevorzuge einen BOB, montiert auf einem kleinen Adapter-Platinchen, auf dem ich einige Jumper platzieren kann, die mir erlauben, Datenleitungen des seriellen Ports zu trennen, ohne die Spannungsversorgung unterbrechen zu müssen. Das ist nützlich für Experimente, da einige Funktionen des Controllers Pins benutzen, die auch zum seriellen Port gehören. Es ist auch möglich, die VCC-Spannung (0 V, 3,3 V oder 5 V) zu wählen. Wenn Sie diese Adapterschaltung verwenden, stecken Sie einen Jumper auf Pin 1 und Pin 2 von JP1, um 5 V als VCC zu wählen. Dies ist notwendig, damit der Spannungsregler des Mini-Kits arbeiten kann. Die serielle Schnittstelle ist für die Programmierung des Mikrocontrollers notwendig – echtes Debugging ist nicht möglich. Wenn Sie den Serial-Wire-Debugger (SWD) besitzen, können Sie ihn statt der seriellen Schnittstelle zur Programmierung verwenden. Leider habe ich keinen SWD,

so dass ich auf die (unten genannte) serielle Schnittstelle angewiesen bin.

Wenn Sie die Hardware beieinander haben, können Sie mit dem Einrichten der Software loslegen. Alles, was Sie brauchen, ist kostenlos, Sie müssen es nur downloaden:

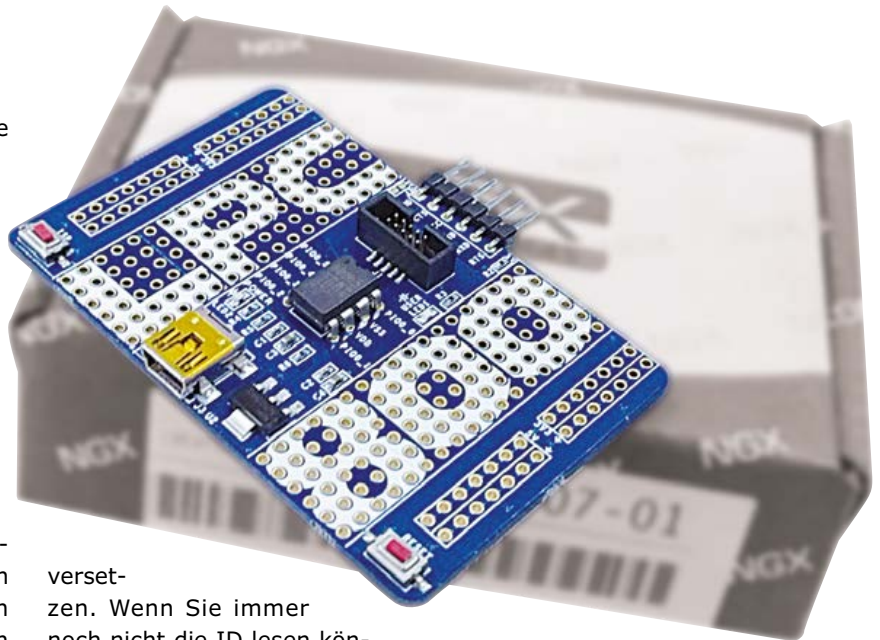
- LPCXpresso IDE (groß, erfordert Registrierung) [1];
- Flash Magic [2];
- LPC800 Mini Kit Basiscode [3].

Unnötig zu sagen, dass Sie die neusten Versionen aller Tools und Bibliotheken downloaden müssen. Installieren Sie die ersten beiden Werkzeuge (wer einen SWD verwendet, kann auf Flash-Magic verzichten). Entpacken Sie den Basiscode irgendwo hin, am besten direkt in den Projekt-Ordner.

Magic Flash ermöglicht die Kommunikation zwischen Mini-Kit und PC. Verbinden Sie das Board mit der seriellen Schnittstelle (über BOB, FTDI-Kabel oder einen eigenen Pegelwandler) und schalten sie es ein. Die Versorgung kann auch über den USB erfolgen (die Datenleitungen sind hier nicht angeschlossen). Die Power-LED auf der Platine leuchtet auf und die User-LED blinkt, wenn das Board noch nagelneu ist.

Drücken Sie die Taste ISP und halten Sie sie unten, während Sie die Reset-Taste drücken. Die blinkende LED verlischt. Starten Sie Magic Flash und wählen Sie den richtigen Controller (LPC810M021FN8) und den richtigen COM-Port aus. Die „Baud Rate“ darf offiziell 115.200 Baud betragen, ich hatte aber Probleme bei dieser Geschwindigkeit und habe deshalb mit 38.400 Baud gearbeitet. „Interface“ sollte „None (ISP)“ sein und das „Oscillator“-Feld frei bleiben (**Bild 2**). Im „ISP“-Menü wählen Sie „Read Device Signature...“. Ein Fenster öffnet sich und wenn alles gut geht, sollte es mit Daten gefüllt sein. Wenn Sie eine „Autobaud“-Fehlermeldung erhalten, funktioniert das Kabel nicht richtig oder der Controller ist nicht im ISP-Modus. Überprüfen Sie das Kabel und probieren einmal eine andere Geschwindigkeit.

Vielleicht ist (aufgrund früherer Experimente) der Reset-Pin gesperrt. In diesem Fall ziehen Sie den Netzstecker oder das serielle Kabel, dann halten Sie die ISP-Taste gedrückt und schließen Power (oder das Kabel) wieder an. Mit diesem Trick können Sie immer den Controller in den ISP-Modus



versetzen. Wenn Sie immer noch nicht die ID lesen können, haben Sie ein Problem mit der Verbindung. Nun ist es an der Zeit, die LPCXpresso Entwicklungsumgebung zu starten. Wenn Sie nach dem Speicherort für die Arbeitsdatei gefragt werden,

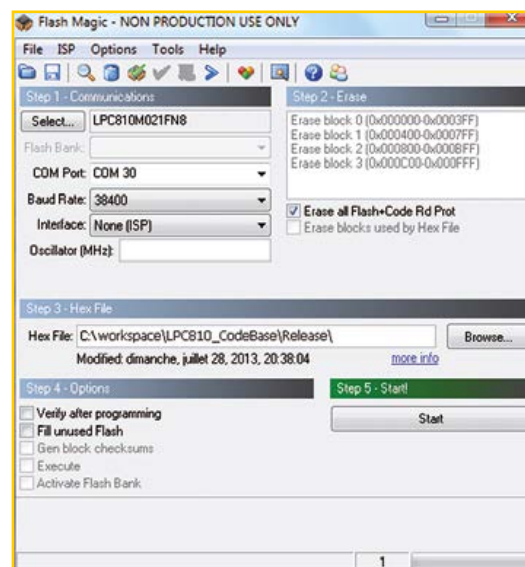
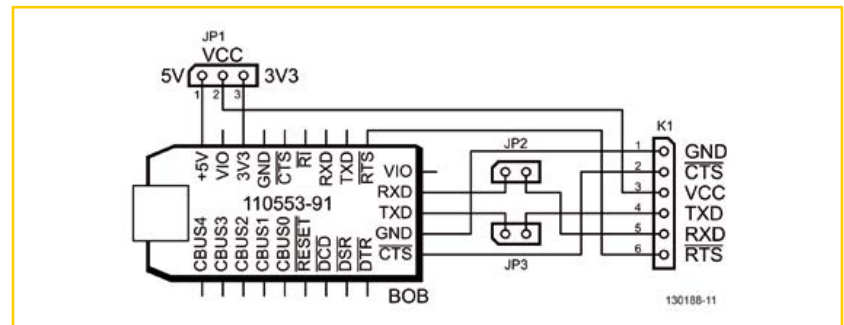


Bild 1.
Das LPC800 Mini-Kit wird hier über den Elektor-BOB programmiert.

Bild 2.
Flash-Magic mit meinen Einstellungen.

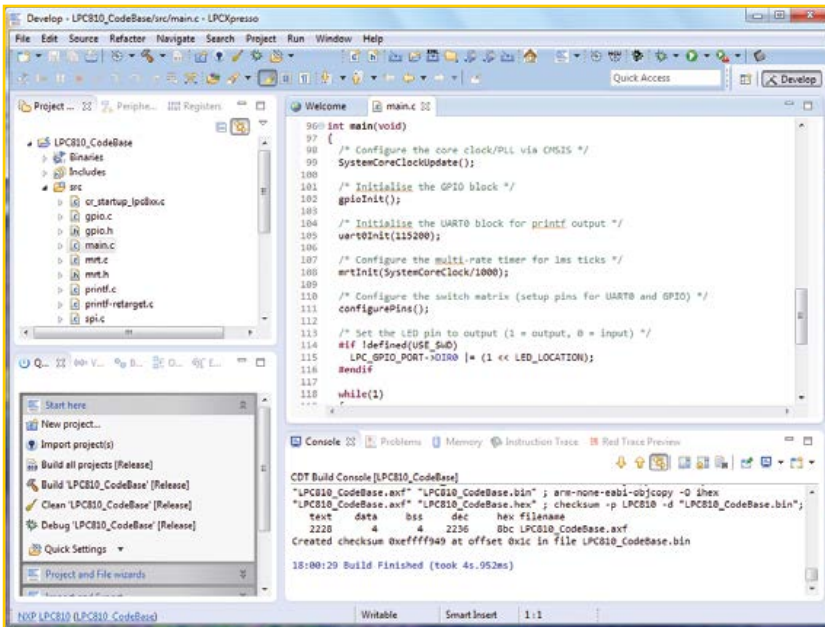


Bild 3. Die LPCXpresso-Entwicklungsumgebung zeigt einen erfolgreichen Build des LPC800_CodeBase-Projekts.

verweisen Sie auf einen Ordner, den Sie zur Speicherung Ihrer Projekte verwenden. Merken Sie sich den Pfad, da Sie ihn später benötigen werden. Die IDE braucht eine Weile, um zu starten, aber wenn sie endlich fertig ist, bietet sie ein schnelles Zugangsmenü namens „start here“ mit den wichtigsten Funktionen (und einigen häufig verwendeten mehr wie „new project“ oder „build & debug“). Hier finden Sie auch eine Möglichkeit, um Beispielprojekte zu importieren. Klicken

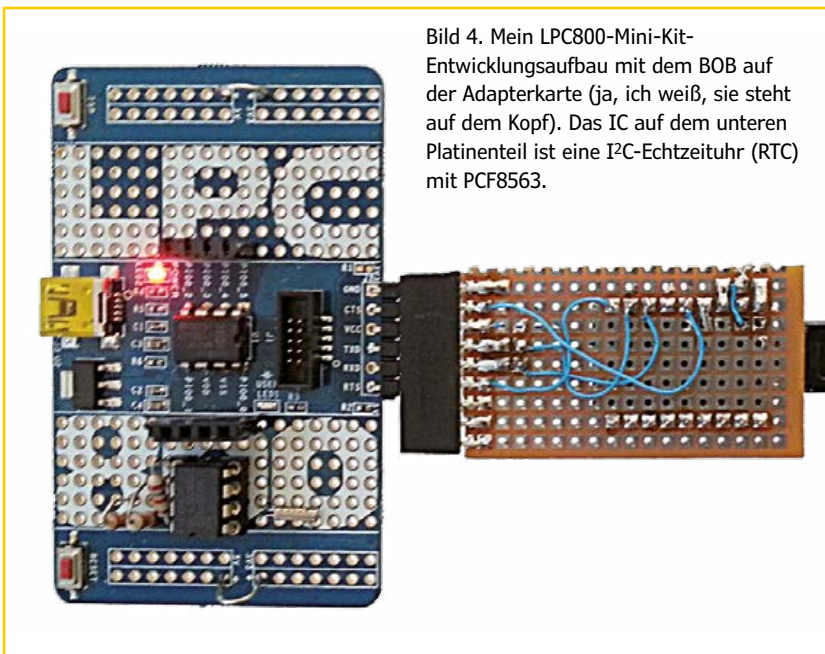


Bild 4. Mein LPC800-Mini-Kit-Entwicklungsaufbau mit dem BOB auf der Adapterkarte (ja, ich weiß, sie steht auf dem Kopf). Das IC auf dem unteren Platinenteil ist eine I²C-Echtzeituhr (RTC) mit PCF8563.

www.elektor-labs.com

Sie auf den Link „Import project(s)“, um den Import-Dialog zu öffnen. Dann klicken Sie auf den „Browse...“-Knopf rechts neben dem „Root directory“-Feld (die Code-Basis sollte schon entpackt sein). Navigieren Sie zum LPC800-Mini-Kit-Code-Basis-Ordner, klicken Sie „OK“ und „Next“. Stellen Sie sicher, dass „LPC810_CodeBase project“ gewählt ist, bevor Sie auf „Finish“ drücken. Sie sehen nun ein Projekt namens „LPC810_CodeBase“ im „Project Explorer“-Fenster. Wählen Sie es aus und starten es vom „Start here“-Menü aus. Beachten Sie die Mitteilungen, die über das „Console“-Fenster scrollen, es sollten keine Fehler oder Warnungen (**Bild 3**) erscheinen. Wenn dies doch geschieht, klicken Sie auf „Problems“, um weitere Informationen zu erhalten. Ein Doppelklick auf eine Zeile in diesem Fenster zeigt Ihnen den problematischen Code.

Nach einem erfolgreichen Build erhalten Sie eine HEX-Datei im „Release“-Ordner des Projekts. Klicken Sie auf „Browse“ in Flash Magic, um zur HEX-Datei zu navigieren. Starten Sie den ISP-Modus des Mini-Kits und drücken auf „Start“. Wenn alles gut geht, wird nun der Controller mit der neuen HEX-Datei programmiert. Drücken Sie die Reset-Taste auf dem Board, um das Programm zu starten.

Sie sind nun bereit, eigene Projekte zu entwickeln. Die Elektor.Labs Website [4] bietet ein paar fix und fertige Projekte für dieses Board. Wenn Sie etwas Bemerkenswertes, Nützliches, Interessantes oder was auch immer mit dem Mini-Kit entwickeln, möchten wir unbedingt davon wissen. Um uns zu informieren, posten Sie einfach Ihre LPC800-Projekte auf Elektor.Labs.

(130188-1)

Weblinks und Quellen

- [1] LPCXpresso IDE: <http://lpcxpresso.code-red-tech.com/LPCXpresso>
- [2] Flash Magic: www.flashmagictool.com
- [3] LPC810 Code Base: <http://lpcware.com/lpc800-mini-kit>
- [4] Elektor.Labs: www.elektor-labs.com

Tipps & Tricks

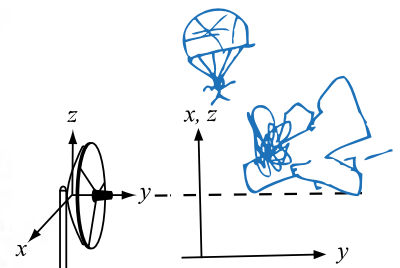
Die LPCXpresso IDE basiert auf Eclipse, einem beliebten - wenn auch meiner bescheidenen Meinung nach absolut schrecklichen - Tool. Ich empfehle daher dem unerfahrenen Benutzer, die Beispiel-Projekte als Ausgangspunkt für eigene Projekte zu verwenden. Sie können mit rechten Mausklicks ein Projekt per „copy & paste“ in das „Projekt-Explorer-Fenster“ aufnehmen, da können Sie sicher sein, dass die Einstellungen korrekt sind.

Projekte, die von Grund auf neu erstellt werden, generieren keine HEX-Datei. Um dies zu korrigieren, kopieren Sie die Einstellungen aus dem Beispiel-Projekt: Dazu wählen Sie das Beispiel-Projekt, im Menü klicken Sie auf „Project“ und dann „Properties“. Erweitern Sie „C/C++ Build“, klicken Sie auf „Settings“ und dann den „Build Steps“-Tab. Kopieren Sie den Inhalt des „Command“-Feldes des „Post-build“-Bereichs in Notepad oder einem anderen Texteditor, so dass sie ihn nicht während der folgenden Schritte verlieren. Schließen Sie nun den „Properties“-Dialog. Wählen Sie das neue Projekt und wiederholen Sie die obigen Schritte, um zu den Post-Build-Schritten zurückzukehren, dieses Mal aber mit dem neuen Projekt. Ersetzen Sie die Command-Zeile der Post-Build-Schritte durch die vorhin kopierte Zeile. Klicken Sie auf „OK“, um die Einstellungen zu speichern. Wiederholen Sie diesen Vorgang für jede Konfiguration, die Sie erstellt haben. Nur für „Release“ kompilieren, anderenfalls werden Sie schnell an die Grenzen des Programmspeichers stoßen (LPC810 hat nur 4 KB). Natürlich, wenn Sie einen SWD haben, möchten Sie vielleicht Debugversionen erstellen, aber Anwender von Magic Flash ziehen daraus keine Vorteile.

Das Hinzufügen von vorhandenen Quellcode-Dateien zu einem Projekt ist ziemlich einfach: „File“ -> „Import...“ -> „Filesystem“ selektieren, „Next“ klicken, zum Speicherort der Datei navigieren, die Datei, die Sie im Auge haben, überprüfen und dann auf „Finish“ klicken. Die Entwicklungsumgebung erinnert sich nicht an den zuletzt gewählten Pfad, so dass unter Umständen eine Menge Mausklicks erforderlich sind, um solche Dinge zu erledigen. Aber Sie können die Datei auch direkt mit Ihrem gewohnten Filemanager in den Projektordner kopieren. Nach dem Kopieren der Dateien drücken Sie „F5“, damit die neuen Dateien in Ihrem Projekt angezeigt werden.

Die Reset-Funktion des Controllers lässt sich von dem Pin trennen, so dass dieser Anschluss später für andere Zwecke eingesetzt werden kann. Die Reset-Taste auf der Platine besitzt dann keine Funktion mehr und es wird schwieriger, in den ISP-Modus zu wechseln. Für diesen Fall sollten Sie den ISP-Taster drücken und gedrückt halten, während Sie die Schaltung von der Stromversorgung trennen und wieder einschalten. Durch diesen Trick kann man den Controller immer in den ISP-Modus versetzen. Dies aktiviert auch wieder das SWD-Interface, sollten Sie es vorher abgeschaltet haben.

6. Draw the radiation pattern for the following parabolic antenna:



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



DesignSpark Tipps & Tricks

Ein einfaches Projekt

Von **Neil Gruending**
(Kanada)

Im letzten Beitrag dieser Serie ging es um das Setup und die Verwendung von Libraries in DesignSpark. Jetzt wird es praktisch: Am Beispiel eines Treibers für eine Zweifarben-LED kann man lernen, wie man die Editoren für Schaltplan und Platinen-Layout benutzt. Es gibt viele Methoden, eine zweifarbige LED anzusteuern. In diesem Beitrag lässt eine H-Brücke eine der beiden Farben leuchten.

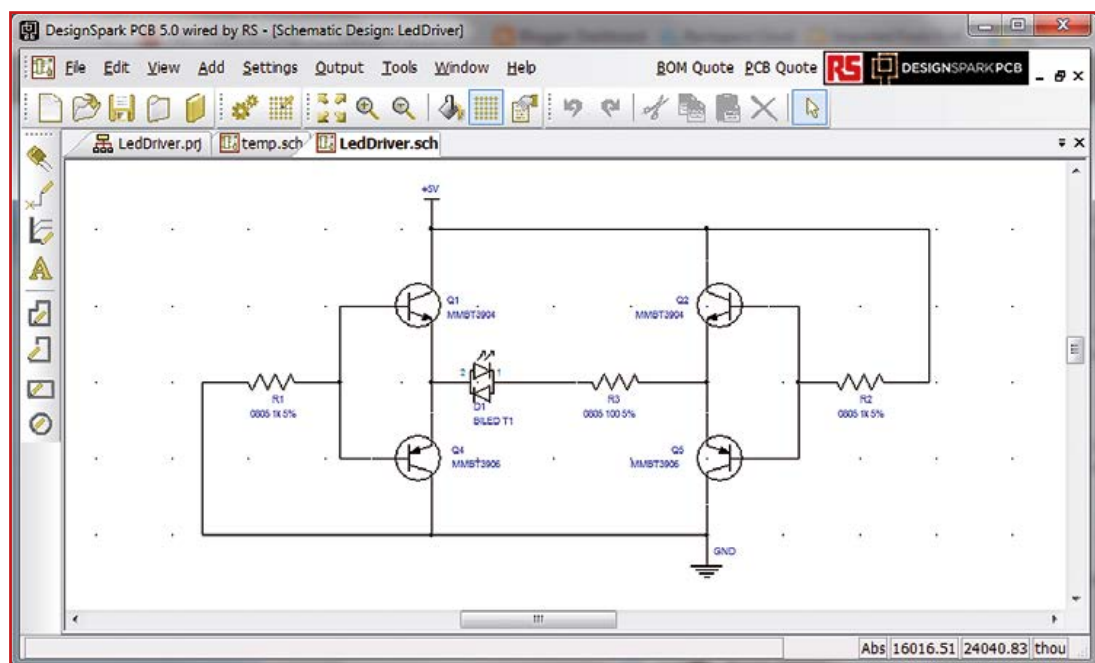


Bild 1.
Schaltung des Treibers für
eine zweifarbige LED.

Schaltplan zeichnen

Zunächst erzeugt man eine Projekt-Datei, damit Schaltplan und Platine miteinander gekoppelt sind. Hierzu wählt man den Menü-Befehl „File>New“. Dann fügt man einen Schaltplan (Schematic Design) - wieder mit dem Befehl „File>New“ - zum Projekt hinzu. Man sollte darauf achten, einen Haken bei „Add To Open Projekt“ zu setzen. An diesem Punkt kann man auch das verwendete „Technology File“ auswählen, das schon früher erläutert wurde. **Bild 1** zeigt die Schaltung.

Für Schaltpläne gibt es auch eine Einführung unter [1], die erklärt, wie man Bauteile hinzufügt und die Schaltung editiert.

Das Verschieben der Textfelder von Bauteilen funktioniert bei DesignSpark anders als bei ande-

ren CAD-Programmen, denn alle Felder werden als Block bewegt. In der Beispielschaltung sind bei den Transistoren Bauteilnummern und Typenbezeichnungen sichtbar. Egal auf welches Feld man klickt - beide werden optisch hervorgehoben und können nur als Gruppe verschoben werden. Dies kann beim Spiegeln von Bauteilen („flip“ in DesignSpark) Probleme machen, da dann die Textausrichtung falsch sein kann. Möchte man rechtsbündige Felder wieder linksbündig haben, ist das einfach: Ein Rechtsklick bringt ein Menü zum Vorschein, wo man dies via „Properties“ ändern kann. Unter dem dortigen Tab „Text“ gibt es die Option „Alignment“, bei der sich zwischen linker, rechter und mittiger Ausrichtung auswählen lässt.

Wichtig zu wissen ist auch, dass die Symbole für

die Spannungsversorgung und Masse in DesignSpark als Bauteile behandelt werden. Die Standard-Symbole befinden sich in der Schema-Library von DesignSpark, doch kann man auch seine eigene Library mit Symbolen erstellen. Ebenfalls wichtig ist, dass bei der Verbindung eines Power-Symbols mit einem Netz eine Warnung auftaucht, dass dies das Netz umbenennt.

Das LED-Bauteil

Für Transistoren und Widerstände wurden die vorhandenen Schaltplan-Symbole und Platinen-Layouts der DesignSpark-Libraries verwendet. Doch bei der LED haben wir das existierende Symbol von DesignSpark modifiziert und ein eigenes Platinen-Layout dafür erstellt.

Der Footprint-Wizard erleichtert die Erstellung eines eigenen Bauteil-Layouts beträchtlich. Zum Wizard kommt man, indem man die Library, die das Layout enthalten soll, mit dem Library-Manager öffnet und auf „Wizard...“ klickt. Der Wizard will dann einige Fragen beantwortet haben. Von den generischen Bauteil-Vorschlägen sollte man den ähnlichsten wählen, damit anschließend wenig editiert werden muss. Bei der LED wählte ich ein axiales Bauteil mit dem Rastermaß 1/10“ (2,54 mm). Hier musste ich lediglich noch die Bauteilform (Silkscreen) editieren und die Polarität von Pin 1 markieren.

Layout-Vorbereitung

Noch etwas Vorarbeit: Passend zur Schaltung wird eine Platinen-Datei mit dem Befehl „Tools->Translate To PCB“ erstellt, was den „New PCB Wizard“ startet. Ziel ist eine zweiseitige, metrische Platine mit einer Kantenlänge von 20 mm. Wenn man den Wizard anweist, die Bauteile außerhalb der Platine zu platzieren, erhält man so etwas wie in **Bild 2**.

Die Bauteile wollte ich im 0,25-mm-Raster platzieren, weshalb zunächst das Raster auf dieses Maß geändert wurde. **Bild 3** zeigt die Platine mit fertig platzierten Bauelementen.

Vor dem Routing der Leiterbahnen noch ein Wort zum verwendeten Raster beim Verlegen der Leiterbahnen. DesignSpark hat keinen interaktiven Autorouter. Daher sollte man das „Routing-Grid“ auf die Breite der Leiterbahnen einstellen. Wenn sich nun zwei Bahnen berühren, haben sie einen Abstand von 0 mm – doch wenn ein Abstand zwischen zwei Bahnen (sichtbar) ist, beträgt er mindestens die Breite einer Bahn. Das funktioniert, weil die Mitte der Bahnen einrastet und nicht die

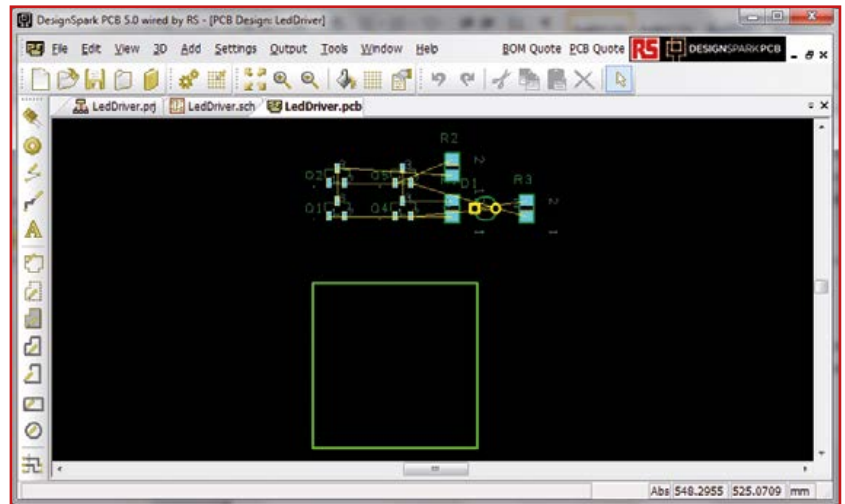


Bild 2. Die Bauteile sind zunächst außerhalb der Platine platziert

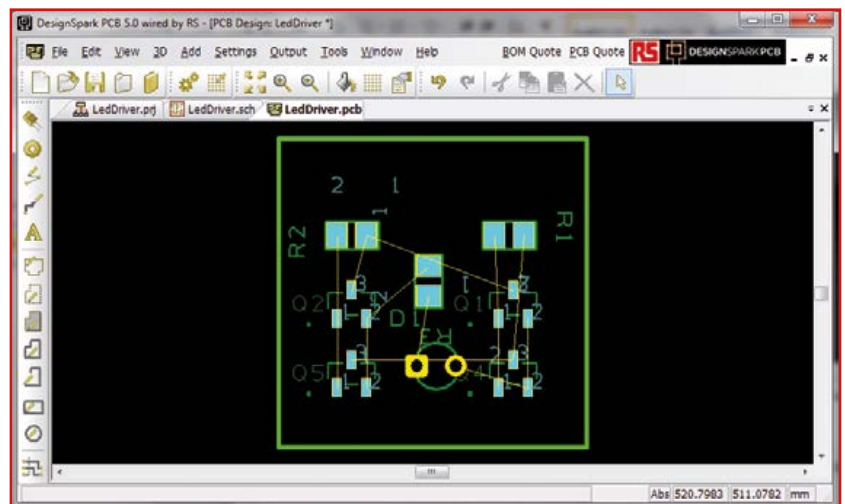


Bild 3. Die Bauteile sind nun im vorgegebenen Raster auf der Platine platziert

Kanten. Bei 0,2 mm breiten Bahnen und einem 0,2 mm Raster ist der minimale Abstand also 0,2 mm. Der Nachteil dieser Technik ist, dass die Leiterbahnbreiten nur Vielfache des Rasters sein sollten. Bei einem Raster von 0,2 mm wäre eine Breite von 0,6 mm okay, nicht aber ein „krummer“ Wert wie 0,35 mm.

Man sollte außerdem für jede verwendete Breite einen „Style“ in „Settings->Design Technology...“ anlegen und dann „Track Styles“ auswählen. Das vereinfacht den Umgang mit verschiedenen Bahnbreiten deutlich. Man kann so gerade bei kom-

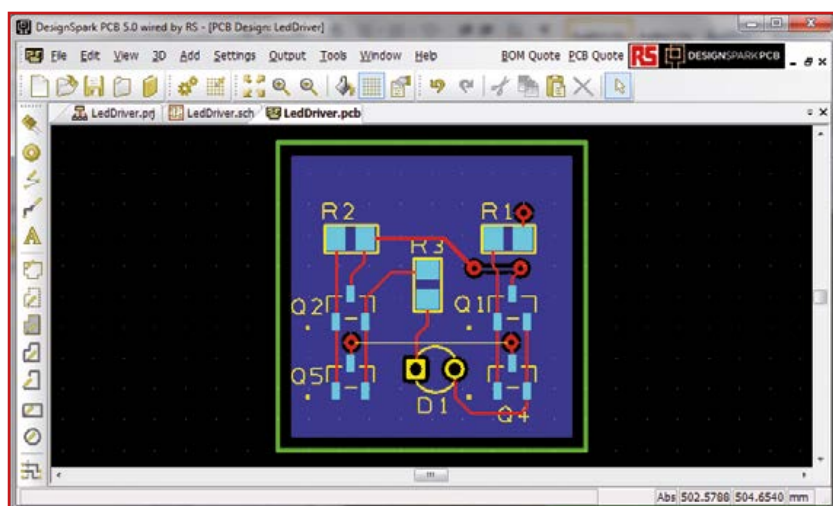


Bild 4. So sieht die fertig layoutete Platine aus.

plexeren Schaltungen einfach alle Bahnen einer bestimmten Breite ändern, indem man schlicht den Style ändert. Den Style der aktuellen Bahn kann man durch Eingabe von „s“ beim Routen ändern.

Das Gleiche gilt für Durchkontaktierungen (Vias). Man konfiguriert deren Styles auf dem Tab „Pads Styles“ im Fenster „Design Technology“. Empfehlenswert ist die Erstellung des Styles „Signal-Via“ und anderer benötigter Styles. Mein Signal-Via hat ein Loch von 0,45 mm und ein Pad von 0,95 mm. Den Style eines Vias kann man per Rechtsklick beim Routen über das Menü „Change Via Style“ ändern.

Mit dem Menü „Settings->Defaults“ stellt man die Standardwerte für Bahnen und Vias ein. Bei mir gab es allerdings Schwierigkeiten beim Wiedererkennen dieser Einstellungen, sogar nach einem Neustart.

Auf der DesignSpark-Webseite [2] gibt es mehr Informationen über das Setup von Platinen und der Bauteilplatzierung.

Layout

Jetzt kann man loslegen und die Platine layouten. Das Resultat ist in **Bild 4** zu sehen.

Alle Bahnen für Signale wurden auf der Oberseite (Top Layer) angebracht. Auf der Unterseite wurde mit einem Polygon eine Massefläche erzeugt. Beim Routen des Boards ist es wichtig, einen Doppelklick auf die „rat's nest“-Verbindungsline beim Verlegen einer Bahn zu machen. Dabei kann man mit einem Rechtsklick die Behandlung der Kurven ändern, in dem man eine andere Seg-

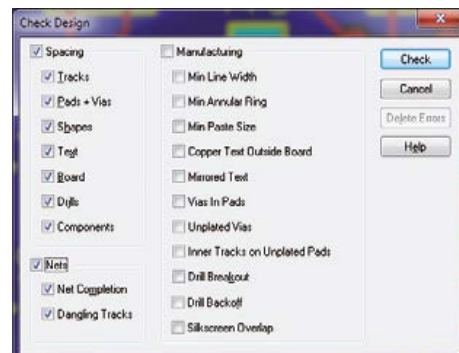


Bild 5. Der „Design Rule Check“-Dialog.

ment-Methode auswählt.

Bei genauem Hinsehen erkennt man eine „rat's nest“-Linie zwischen Q4 und Q5 für die Masseverbindung. DesignSpark ist also der Ansicht, dass die beiden Transistoren nicht mit Masse verbunden sind. Glücklicherweise gibt es einen DRC (Design Rule Check) im Tools-Menü, mit dem man alle Verbindungen prüfen kann — siehe **Bild 5**. Wenn man in diesem Dialog auf „Check“ klickt, wird das komplette Layout dahingehend verifiziert, ob alle ausgewählten Design-Kriterien eingehalten werden. Anschließend wird ein Fehler-Report ausgegeben. Die Fehler sind im Layout markiert und wenn man mit der Maus darüber fährt, erscheint die zugehörige Fehlermeldung. Alle Clearance-Regeln sind unter dem Tab „Spacing“ im Menü „Settings->Design-Technology...“ eingestellt, wo in einer Tabelle alle Clearances (Abstände) zwischen den verschiedenen Objekten aufgeführt sind. Auch für das Routing gibt es eine Anleitung unter [3].

In der nächsten Folge werden BOM- und Gerber-Dateien erzeugt, mit denen man die Platine herstellen (lassen) kann.

(130230)

Weblinks

- [1] www.designspark.com/eng/tutorial/schematics-entry
- [2] www.designspark.com/tutorial/pcb-setup-placing-components
- [3] www.designspark.com/tutorial/pcb-routing

Seminar ‚Arduino - Programmierung und Projektentwicklung‘

Nach einer kurzen Einführung und der Inbetriebnahme des Arduino-Boards erfolgt eine systematische Einführung in verschiedene Themengebiete.

Dabei wird neben den erforderlichen theoretischen Grundlagen stets größter Wert auf eine praxisorientierte Ausrichtung gelegt. So werden wichtige Techniken wie AD-Wandlung, Timer oder Interrupts anhand von Praxisprojekten ausführlich erläutert.

Den Abschluss des Seminars bildet eine Einführung in die eigenständige Entwicklung von Projekten und Systemen. Der Seminarteilnehmer wird damit in die Lage versetzt, auch komplexe eigene Ideen in praxistaugliche Geräte umzusetzen.

Veranstaltungsort/-termin: Zürich (CH), 07.11.2013
Referent: Dr. Günter Spanner
Teilnahmegebühr: € 349,00 Euro (inkl. MwSt)



Altium Seminar ‚Leiterplattenentwicklung – Bibliotheken intelligent aufbauen‘

Im Seminar werden verschiedene Aspekte erarbeitet, die bei der Anlage von Bibliotheken zu beachten sind. Das fängt ganz klassisch bei der Struktur an, die besagt, welche Komponenten wo abgelegt werden, damit der Entwickler später eine Chance hat, die Bauteile zu finden. Aber auch die Belange der Fertigung und des Einkaufs werden in einer guten Datensammlung berücksichtigt. Heute sind diese sicher keine Optionen, sondern feste Anforderungen.

Nicht zuletzt werden aber auch die Möglichkeiten eines aktuellen EDA Tools am Beispiel von Altium Designer beleuchtet. Hier geht es sowohl um generelle Bibliothekskonzepte, aber auch um die Realisierungsmöglichkeiten.

Veranstaltungsort/-termin: Hanau, 21.11.2013
Referent: Dipl. Ing. Christian Keller
Teilnahmegebühr: € 299,00 Euro (inkl. MwSt)

50% Sonderrabatt für Elektor-Abomitmglieder: 149,00 € (inkl.MwSt)

3-tägiges Seminar ‚Embedded Linux in Theorie und Praxis‘

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

Veranstaltungsort/-termin: Zürich (CH), 09.12 – 11.12.2013
Referent: Robert Berger
Teilnahmegebühr: € 1.898,00 € (inkl. MwSt.)



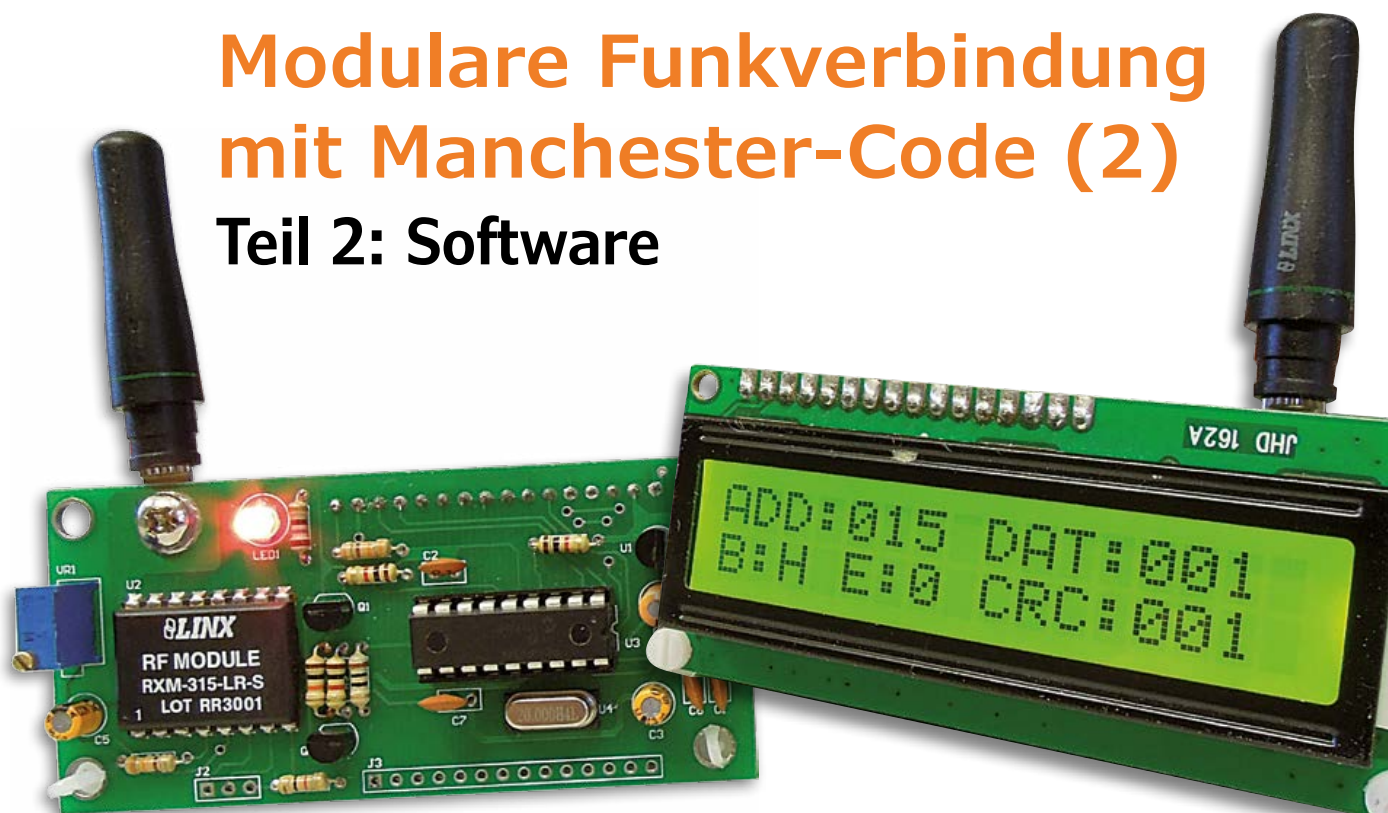
2-tägiges Seminar: Linux Debugging – ein Crash-Kurs

Diese 2-tägige Schulung beinhaltet praktische Übungen kominiert mit Theorie, um einige grundlegende Konzepte der Debugging / Tracing / Profilieren Embedded GNU / Linux zu illustrieren. Hands-on-Sessions werden auf Ubuntu durchgeführt. Der gesamte Workshop ist konzipiert, um Sie schnell „up-to-speed“ zu bringen. Erfinden Sie das Rad nicht neu, sondern lernen Sie von einem erfahrenen Trainer und nehmen Sie dieses Wissen mit nach Hause, um diese neu erworbene Fähigkeit bei eigenen Entwicklungen effektiv zu nutzen.

Veranstaltungsort/-termine: München, 12.12. + 13.12.2013
Referent: Robert Berger
Teilnahmegebühr: € 995,00 € (inkl. MwSt.)

Modulare Funkverbindung mit Manchester-Code (2)

Teil 2: Software



Im ersten Teil ging es um die Hardware, jetzt kommt die Software an die Reihe. Wo gute Hardware und ein passendes Platinen-Layout wichtig für die korrekte Aussendung und den Empfang der HF-Signale ist, steht die Software bzw. Firmware für die Zuverlässigkeit der Datenübertragung mit Hilfe dieser Signale.

Von **Marcelo Maggi**
(USA)

Wie schon im ersten Teil erläutert, kommen die HF-Module ohne integrierte Intelligenz aus. Eine wirklich smarte Datenübertragung per Funk benötigt daher zusätzlich einen Mikrocontroller, der mit schlauder Software gefüttert ist. Der Begriff Intelligenz bezieht sich in diesem Kontext auf die Auswahl eines guten Codierungs-Protokolls, der Übertragungsgeschwindigkeit, der Datenorganisation und den Fehlererkennungs- und Fehlerkorrekturmechanismen.

Die Software ist modular strukturiert. Essentielle Funktionen sind als Treiber organisiert, die unverändert bleiben. Der Anwender bleibt aber komplett frei darin, seine eigenen Routinen im Hauptprogramm zu implementieren. Ein Aufruf des TX-Treibers verschickt die Daten. Analog dazu übernimmt auf der Empfängerseite der RX-Treiber alle komplexen Aufgaben und gibt dem Hauptprogramm lediglich die Nutzdaten zurück.

Manchester-Code

Die Manchester-Codierung gilt als einfaches aber dennoch zuverlässiges Verfahren, Daten sicher über eine Funkstrecke zu transportieren. Tests zeigten, dass bei diesem Projekt bis zu 5 kb/s bei guter Stabilität drin sind, weshalb dies den Maximalwert darstellt. Langsamere Geschwindigkeiten erlauben mehr Flexibilität. Die zu transportierenden Daten werden in einer Makrostruktur mit der Bezeichnung „frame“ verpackt. In diesem steckt auch ein Byte zur Fehlerkorrektur. Das Prinzip ist einfach: Jedes Bit wird durch einen Übergang statt durch einen logischen Pegel repräsentiert. **Bild 1** zeigt das Prinzip der Manchester-Codierung und die beiden Konventionen zur Bit-Repräsentation. Hier wird die Definition nach IEEE 802.3 eingesetzt: Eine logische „1“ entspricht einem Low/High-Übergang und eine logische „0“ dem

Gegenteil. Die Vorteile dieser Codierung liegen auf der Hand:

- Der Takt ist im codierten Signal enthalten und unabhängig von den Daten. Er kann problemlos nach Empfang extrahiert werden.
- Da hier das HF-Signal im Rhythmus des Nutzsignals ein- und ausgeschaltet wird, reduziert sich die mittlere Sendeleistung. Das ist nicht nur für portable Anwendungen gut, sondern hilft auch, die Grenzen der erlaubten Sendeleistung im ISM-Band einzuhalten. Durch die höhere Spitzenleistung ist auch die Reichweite größer.

Die Linx-Module sind für bis zu 10 kb/s spezifiziert. Trotzdem wird hier nur mit maximal 5 kb/s gesendet. Dies ist der Codierung geschuldet, denn jedes Bit hat ja zwei logische Pegel und eine einfache binäre Übertragung würde die Bitrate verdoppeln. Aufgrund der Vorteile der Manchester-Codierung wird die Übertragungsrate zugunsten von Einfachheit und Zuverlässigkeit reduziert. Mit 5 kb/s wurden zuverlässige Funkverbindungen über 180 m (getestet) erreicht.

Geringere Geschwindigkeiten ($\leq 2,5$ kb/s) sind bei höheren Störpegeln oder größeren Entfernungen sinnvoll. Linx behauptet, dass unter optimalen Bedingungen bis knapp 1 km möglich sind.

Datenformat

Zu übertragende Daten bestehen aus drei 1-Byte-Elementen: Adresse, Daten und CRC. Das Adress-Byte steht für den Empfänger, an den sich die Sendung richtet. Bei einem Byte könnten 256 Empfänger adressiert werden. In den meisten Fällen würden damit Bits verschwendet. Effizienter ist es, in diesem Byte den Empfänger und die Funktion zu codieren:

- Das obere Nibble ist die Adresse (16 Empfänger).
- Das untere Nibble ist die Funktion im Empfänger (16 Optionen).

Man kann so z.B. Empfänger Nr. 5 mitteilen, mit den kommenden Daten die Funktion Nr. 9 auszuführen. Damit kann dann etwa ein Servo angesprochen werden, dessen Sollposition in den Daten enthalten ist.

Im Beispiel-Code wird lediglich die Adresse mit dem Maximalwert 15 verwendet. Der Empfänger

Eigenschaften

- Ein Sender und mehrere Empfänger möglich.
- Reichweite bis 180 m.
- Datenrate zwischen 2,5 und 5 kbit/s umschaltbar.
- Vorbereitet für Fehlerkorrektur.
- Platinen-Layout und Code sind kostenlos unter [2] verfügbar.

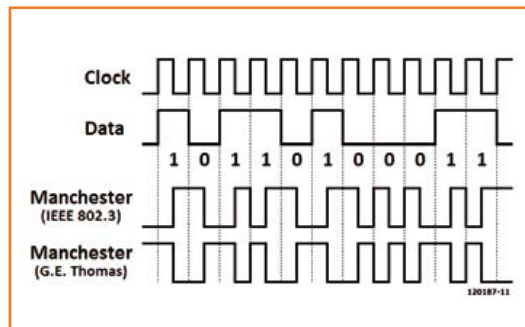


Bild 1.
Prinzip der Manchester-Codierung plus Bit-Konventionen.

betrachtet die ausgelesene Adresse 15 als gültig. Die Wahl von 15 ist kein Zufall: Binär entspricht ein Byte „15“ der Bitfolge „00001111“, was sehr gut mit einem Oszilloskop erkannt werden kann. In Manchester-Codierung entspricht so ein Byte der Sequenz „1010101001010101“.

Im Daten-Byte steckt die zu übertragende Information. Das kann ein fixer Wert für eine bestimmte Aufgabe sein. Auch ein variabler Wert wie die Ausgabe eines A/D-Konverters ist möglich. Im Beispiel-Code werden unterschiedliche Daten in Abhängigkeit vom Zustand von Pin B3 (RB3/CCP1, Pin 9) übertragen: Bei „low“ enthält das Daten-Byte den Wert „1“ und bei „high“ ändert sich dieser Wert in Abhängigkeit von einem internen Timer zwischen „0“ und „1“. Am Empfänger kann man damit die LED D1 ansteuern und so blinken lassen. Mit dieser einfachen Methode kann man die Funkstrecke testen. Sie zeigt außerdem, wie man das Daten-Byte verwenden kann.

Trotz der Einfachheit des Beispiel-Codes empfiehlt es sich, dessen Möglichkeiten zu studieren. Bei geschickter Nutzung des Adress-Bytes kann man auch längere Informationen als ein Byte übertragen. Man könnte das untere Nibble zur Indizierung des übertragenen Bytes verwenden, so dass der Empfänger die empfangenen Daten sauber reorganisieren kann. Damit wird die

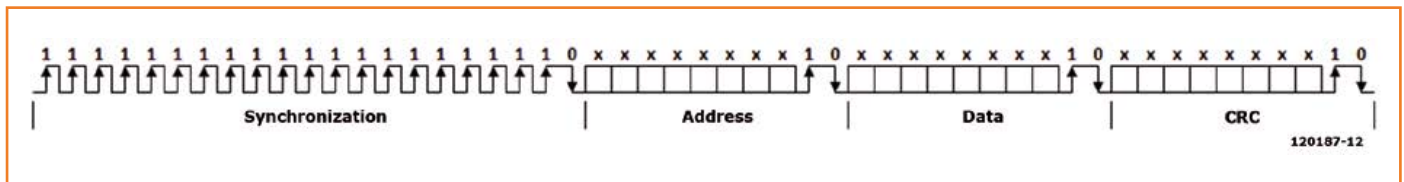


Bild 2.
Komplette Frame-Struktur.

Funkstrecke effizienter, als wenn man lediglich ein Byte pro Frame übertragen würde. Das Beispiel demonstriert nur die grundsätzlichen Möglichkeiten. Mit geringen Änderungen kann man die Übertragungsleistung erhöhen.

CRC-Daten (Cyclic Redundancy Check) enthalten Informationen zur Validierung der empfangenen Daten. Damit kann man z.B. ein fehlendes Bit rekonstruieren. Die verschiedenen Verfahren zur Berechnung eines CRC-Bytes überschreiten den Rahmen dieses Artikels, aber soviel steht fest: Eine stabile Übertragungsstrecke benötigt so ein Verfahren. Aus diesem Grund enthält der Beispiel-Code ein simples Verfahren: die Daten werden einfach in das Fehlererkennungs-Byte kopiert. Auf der Empfangsseite ist bei Fehlererkennung ungleich Daten-Byte klar, dass etwas schief gegangen ist. Leider ergibt sich daraus kein Weg, den Fehler zu korrigieren.

Frame-Aufbau

Nun müssen die drei Elemente in einem Frame als Makrostruktur zusammengefasst werden. Da jede Funkverbindung mit Störungen und Rauschen zu tun hat, muss der Empfänger erkennen können, wo ein Frame anfängt und ob er gültig ist. Daher muss vor jeden Frame eine Phase zur Frame-Synchronisation geschaltet werden. Bei dieser Synchronisationssequenz handelt es sich um eine unveränderliche Bitfolge, die durch keine Kombination der drei Elemente eines Frames erreicht werden kann.

Die Sequenz lautet „111111111111111110“. Der eigentliche Trick ist nun, dass jedes Frame-Byte von einem Abstandshalter aus der kurzen Sequenz „10“ gefolgt wird. Ein potentieller Konflikt von Frame und Synchronisation, wenn sowohl Adresse als auch das Datenbyte „11111111“ enthält und CRC = „11110XXX“, wird so vermieden, denn die Frame-Sequenz lautet dann „11111111 10 11111111 10 11110XXX 10“. Die lange Folge von Einsen wird durch den Abstandshalter unterbrochen. Es kann also keine Verwechslung von Synchronisation und Frame geben. **Bild 2** zeigt einen kompletten Frame in Manchester-Codierung.

Datenrate

Die Übertragungsgeschwindigkeit wird durch den Sender (TX) bestimmt. Sie ist in der Firmware definiert und kann über den Pin RB2/TX/CK (Pin 8) von IC2 via Pin 11 von K2 umgeschaltet werden. Bei „low“ wird mit 2,5 kbit/s und bei „high“ mit 5 kbit/s gesendet. Der Empfänger erkennt die Geschwindigkeit automatisch. Da die Erkennung durch eine Zeitmessung auf Basis des verwendeten Takts erfolgt, ist es sehr wichtig, einen Quarz mit exakt 20 MHz zu verwenden.

Damit die Funkverbindung aufrechterhalten wird, muss der Empfänger regelmäßig Frames senden. Einzelne Frames oder Folgen von Frames mit Abständen von mehr als 10 ms werden u.U. nicht richtig erkannt. Der Empfänger muss nämlich erst aufwachen und die Verstärkung richtig einstellen, um die Signale richtig zu empfangen und zu demodulieren. Nach einer Pause von mehr als 10 ms ist der Empfänger nicht mehr bereit für einen einlaufenden Frame.

Die simpelste Lösung dafür ist, Frames periodisch mit Abständen von < 10 ms zu senden. Das funktioniert, bringt aber zwei Nachteile mit sich: Der Mikrocontroller darf keine Tasks verarbeiten, die länger als 10 ms sind.

Es kann in Ihrem Land illegal sein, zu oft und zu viel im gegebenen Frequenzband zu senden. Aber auch das kann man lösen: Man verwende einen Timer im Hauptprogramm und verschicke zwei oder drei Frames am Stück mit einem Abstand von < 10 ms. Dann kann man den Empfänger auch für einige Sekunden inaktiv sein lassen.

Hinweis: Der von [1] downloadbare Beispiel-Code zeigt lediglich, wie die Funkverbindung arbeitet und ermöglicht einen Funktionstest. Das Hauptprogramm sollte nicht einfach unverändert in einer richtigen Anwendung eingesetzt werden. Außerdem empfiehlt es sich, die nationalen Regularien für das ISM-Band zu kennen und zu berücksichtigen, damit Sie Ihre Funkstrecke legal betreiben.

RX-Treiber

Die Frames werden durch den Treiber des Empfängers decodiert. Im Treiber steckt die komplette Komplexität, weshalb sein Code gut kommentiert ist. Vom Hauptprogramm aus ruft man lediglich die Funktion „mc_rx“ auf, die dann die restlichen Funktionen zur Bitmanipulation aufruft.

Zuerst wird die Datenrate extrahiert. Die Funktion „baud_detect“ misst die Länge von zwei halben Bits. Alle diese Aktivitäten werden von den steigenden Flanken des einlaufenden Bit-Stroms ausgelöst. Das erste Bit eines Frames ist eine „1“ – in Manchester-Codierung eine ansteigende Flanke. Wenn dadurch ein Interrupt ausgelöst wird, ist die erste Hälfte dieses Bits schon vorbei. Von daher erfasst „baud_detect“ de facto die zweite Hälfte des ersten Bits und die erste Hälfte des zweiten Bits.

Wäre es nicht ausreichend, nur eine Hälfte eines Bits zu messen, weil dies ja symmetrisch ist? Die Antwort ist jein: Wenn man kontinuierlich Daten sendet, bleibt der Empfänger aktiv, und die Gleichspannungspegel sind stabil. In diesem Fall sind die empfangenen Bits tatsächlich symmetrisch. Doch bei Pausen zwischen den Frames von ≥ 10 ms ist der Empfänger nicht richtig eingestellt, wenn das erste Bit des neuen Frames ankommt. Als Folge davon ist das erste Bit des neuen Frames dann nicht symmetrisch, wie **Bild 3** zeigt. Auch wenn die Verbindung funktionieren sollte, wenn man kein ganzes Bit als Basis der Datenrate vermisst, könnte dabei doch ein Wert herauskommen, der bei der Detektion der Bits gerade von längeren Sequenzen zu Fehlern führt. Im Beispiel-Code betragen die Pausen nur 1 ms, sodass die Stabilität kein Problem sein sollte. Doch die Funktion „baud_detect“ soll auch in extremeren Fällen gute Resultate liefern.

Nach der ersten Hälfte des zweiten Bits enthält „timer0“ die Bit-Dauer in internen Einheiten. Die Umrechnung ist einfach:

- Ein 20-MHz-Quarz führt zu einem 5-MHz-Befehlstakt.
- Die Periodendauer ist $1 / 5 \text{ MHz} = 200 \text{ ns}$.
- Der Wert von „timer0“ inkrementiert alle $200 \text{ ns} \times 16 = 3,2 \text{ } \mu\text{s}$.
- Ein Bit dauert $1 \text{ b} / 5 \text{ kb/s} = 200 \text{ } \mu\text{s}$.
- Nach einem Bit hat „timer0“ rechnerisch den Wert $200 \text{ } \mu\text{s} / 3,2 \text{ } \mu\text{s} = 62,5$.

Als 8-bit-Zähler kann „timer0“ nur einen Wert von 62 oder

63 haben. Das ist aber nicht wichtig, denn Funksignale haben sowieso mit Jitter zu kämpfen, wodurch die Flanken der Bits entsprechend variieren. Der Wert von „timer0“ wird also um den Mittelwert von 62,5 pendeln. Das Programm betrachtet Werte im Bereich von 55 bis 70 als gültig. Wenn „timer0“ in diesem Bereich liegt, wird angenommen, dass die Bit-Dauer $200 \text{ } \mu\text{s}$ beträgt und ein halbes Bit eben $100 \text{ } \mu\text{s}$ benötigt. Genau das steht in der Variablen „semiperiod“. Bei 2,5 kb/s enthält „semiperiod“ den Wert 200. Der gültige Bereich für „timer0“ reicht dann von 118 bis 133. Nachdem die Bit-Dauer ermittelt wurde, ist die Bit-Detektion einfach: Einlesen des Signals und warten auf eine Pegeländerung, nächsten Pegel erfassen und vergleichen. Bei jedem Bit gibt es ja einen Pegelwechsel in der Mitte des Bits. Bevor man das nächste Bit bestimmt, wird ein Intervall „semiandjitter“ gewartet, um sicherzustellen, dass wirklich das nächste Bit und nicht fälschlicherweise der Wert vom ausklingenden aktuellen Bit ausgewertet wird. Wenn „status 1 > status 2“ (fallende Flanke, „in1“ und „in2“ im Code), dann hat das empfangene Bit den Wert „0“ – anderenfalls den Wert „1“. Anschließend wird dieses Bit zur 32-bit-Variablen „three_byte_rx“ hinzugefügt, die im Anschluss um eine Stelle nach links geschiftet wird, damit sie für das nächste Bit bereit ist. Nachdem die Abstandshalter sauber eliminiert sind, enthält „three_byte_rx“ die drei Bytes für Adresse, Daten und CRC. Das überzählige Byte ist leer.

Solange dabei kein Fehler aufgetreten ist, wird „three_byte_rx“ mit der Funktion „ad_da_cr_rx“ in die Variable „frame_rx“ übertragen, damit die Werte via Funktion „mc_rx“ zur Verfügung stehen. Anschließend wird die Variable durch Multiplikation mit 256 um acht Stellen nach links geschiftet und der Wert von „semiperiod“ hinzuaddiert. Nun ist „frame_rx“ bereit, zum Hauptprogramm zurückgegeben zu werden, wo die empfangenen Frame-Werte dann in „frame“ stehen. Der Fehlerfall wird dem Hauptprogramm dadurch signalisiert, dass die Variable „frame_rx“ leer zurückgegeben wird.

Noch ein Hinweis: Wie auch bei der Software für den Sender empfiehlt es sich, bei eigenen Anpassungen darauf zu achten, möglichst keine schon vergebenen Variablennamen zu verwenden, um schwer zu fixende Konflikte zu vermeiden.

TX-Firmware

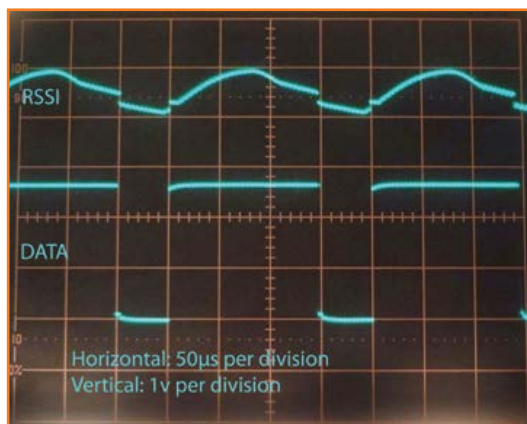
Die fundamentalen Routinen für Sender und Empfänger sind wie bei PC-Hardware-Treibern in einer separaten Datei gruppiert. Diese Routinen enthalten den komplexen Teil des Codes und können unverändert verwendet werden. Im Hauptprogramm kann man mit wenigen Code-Zeilen die zuvor beschriebenen Funktionen realisieren, mit denen man überprüfen kann, ob die Funkverbindung funktioniert. An dieser Stelle kann man auch eigenen Code einfügen.

Alle Software wurde in C mit dem CCS-Compiler erstellt. Die Kommentare im Code stellen zusätzliche Informationen zur Verfügung. Der TX-Code ist in drei Dateien aufgeteilt:

- Manchester_Link_TX.c: das Hauptprogramm
- Manchester_Link_TX.h: Setup für den PIC
- MAN_TX.c: Treiber für den Sender

Das Hauptprogramm ist recht einfach: Nach den initialen Definitionen kommt eine Endlosschleife.

Bild 3.
Oszillogramm eines
empfangenen Signals, bei
dem die Frames mit Pausen
von 10 ms gesendet wurden.



Hier wird Pin B2 zur Auswahl der Datenrate eingelesen. Die Adresse ist 15, Pin B3 wird zur Auswahl der zu sendenden Daten eingelesen und CRC ist eine Kopie der Daten. Die Sende-Funktion wird mit vier Parametern aufgerufen:

```
mc_tx(baud, address, data, crc);
```

Die letzte Zeile besteht lediglich aus einer Pause von 1 ms, bevor das Ganze von vorne beginnt. Der Sende-Treiber ist der Kern dieser Software. Er bleibt gleich und ist unabhängig vom Code des Anwenders. Er muss im Programm ganz am Anfang nach dem PIC-Setup eingefügt werden. Der Treiber enthält alles, um ein Frame in Manchester-Codierung aus den vier Variablen des Hauptprogramms zu generieren. Aus der Bit-Rate wird die Zeit „semiperiod“ für ein halbes Bit berechnet, so dass eine Manchester-Codierung leicht erledigt werden kann. Die Frame-Synchronisation wird durch zwanzigfachen Aufruf der Funktion „one“ und dem einfachen Aufruf von „zero“ erzeugt. Jedes Byte wird beginnend beim MSB bitweise analysiert. Abhängig vom Ergebnis wird die passende Funktion „one“ oder „zero“ aufgerufen und der Abstandshalter nach einem Byte eingefügt. Das war es auch schon. Der komplette Frame samt den beschriebenen Feinheiten wird dann mit der eingestellten Datenrate seriell via Pin B0 ausgegeben. Bei Standard-Frames sind also keinerlei Änderungen notwendig. Auch für mehr Bytes in einem Frame halten sich die notwendigen Änderungen in Grenzen. Zur Anzeige der codierten Information (Adresse, Daten und CRC) ist ein LCD-vorgesehen.

RX-Software

Der Code für den Empfänger ist etwas komplexer als sein Gegenstück für den Sender. Aber

keine Angst: Die Komplexität steckt im Treiber, der im Kasten „RX-Treiber“ erläutert wird. Auch hier gibt es drei Dateien:

- Manchester_Link_RX.c: das Hauptprogramm
- Manchester_Link_RX.h: Setup für den PIC
- MAN_RX.c: Treiber für den Empfänger

Das Hauptprogramm ist ebenfalls einfach aufgebaut. Nach den LCD-Prozeduren kommen die Standard-Definitionen, dann die ISR (Interrupt Service Routine):

```
void detection_isr();
```

Sie ist wichtig, da der empfangene Datenstrom an Pin B0 (RB0/INT, Pin 6) anliegt und die steigenden Flanken einen externen Interrupt auslösen. Bei jedem Interrupt wird dann sofort die ISR angesprungen.

Eine andere wichtige Zeile ist die Einstellung von „timer0“:

```
setup_timer_0(RTCC_INTERR|RTCC...)
```

Hier wird „timer0“ als 8-bit-Timer definiert, der alle 16 Befehls-Takte inkrementiert wird. Mit diesem Timer wird die Datenrate durch Messung der Länge eines Bits bestimmt.

Da Interrupts verwendet werden, müssen diese auch freigegeben werden:

```
enable_interrupts(GLOBAL);
```

Anschließend kommt die Endlosschleife, in der die ankommenden Daten überwacht werden. In dieser Schleife kann nach Belieben weiterer Code eingefügt werden, solange dadurch die Interrupts nicht deaktiviert werden. Einige Aktionen wie die Ausgabe auf dem LCD könnten die Interrupts beeinträchtigen. Von daher müssen diese Aktivitäten direkt in der ISR erledigt werden, nachdem die Daten empfangen und validiert wurden. Der Empfänger wartet die ganze Zeit auf ein empfangbares Signal.

Bei einer steigenden Flanke wird die ISR angesprochen, in der die Interrupts deaktiviert werden und dann die Empfangsfunktion aufgerufen wird:

```
frame=mc_rx(frame);
```

Diese Funktion steckt im Empfänger-Treiber. Sie gibt die 32-bit-Variable „frame“ zurück. Sie ist elegant und schlank: Bei Signalempfang wird die Funktion „mc_rx“ aufgerufen und „frame“ enthält dann die übertragenen Daten.

Als 32-bit-Variable enthält „frame“ die vier Bytes „address“, „data“, „CRC“ und „halftime“ des empfangenen Frames. Die Bytes werden wie folgt aus „frame“ extrahiert. Zunächst wird einfach



Bild 4.
Elektor-Prototypen mit LCD.

„frame“ in eine 8-bit-Variable kopiert, so dass nur die acht LSBs übrig sind:

```
halftime=frame;
```

Somit enthält „halftime“ das vierte Frame-Byte. Nun wird der Inhalt von „frame“ um 8 bit nach rechts geschiftet und der Inhalt wieder in eine 8-bit-Variable kopiert, um das dritte Byte zu erhalten. Nach zwei weiteren 8-bit-Shifts sind alle Bytes extrahiert.

Die Variable „halftime“ enthält die halbe Dauer eines Bits. Aus diesem Wert wird die Datenrate berechnet. Die restlichen Befehle demonstrieren die grundlegende Verarbeitung der empfangenen Daten. Bei einem Fehler in einem Frame gibt die Funktion eine „0“ zurück („halftime“ = 0)s. Damit wird das Fehler-Flag gesetzt.

Wenn „address“ = 15 ist und kein Fehler erkannt wurde, dann werden die Daten dazu verwendet, die LED zu steuern. CRC wird in diesem Beispiel nicht verwendet.

Das LCD zeigt kontinuierlich die empfangenen Daten an (Adresse, Daten und CRC), nachdem der Interrupt „enabled“ ist. Ohne Datenempfang wird auf dem LCD der Fehler-Status als „1“ und der Bit-Status als „L“ angezeigt.

Verbesserungen

Obwohl die Funkstrecke gut arbeitet und sich bei diversen Tests als sehr stabil erwiesen hat, gibt es noch genug Möglichkeiten für Verbesserungen und Anpassungen wie z.B. die Übertragung von mehr als einem Daten-Byte pro Frame. Wenn man mehr Daten-Bytes in einem Frame transportiert, muss man die vorgesehene Anzahl kennen, damit man kein Byte verliert. In diesem Fall ist eine 32-bit-Variable zu klein und eine „struc-

ture“ eignet sich besser als Speicherort für einen Frame. Es sind sehr viele Anwendungen denkbar und diese grundlegende Lösung ist als eine Art Baustein für größere eigene Projekte gedacht.

Bild 4 zeigt die im Elektor-Labor aufgebauten Prototypen. Das LCD zeigt „address = 15“, „data = 1“, „CRC = 1“, „B = H“ (hohe Datenrate = 5 kb/s) und „E = 0“ (kein Fehler).

Schon die grundlegende Anwendung zeigt das Potential dieser Module. Und falls die Funkverbindung Probleme machen sollte, sucht man an erster Stelle im Code.

An diesem Punkt ist der zweite und letzte Teil dieses Projekts abgeschlossen. Sollten Sie Fragen, Bedenken oder Anregungen zur Hardware oder Software haben, dann zögern Sie nicht und nehmen über [3] Kontakt auf oder statten dem Autor einen Besuch auf seiner Webseite [4] ab. Dateien für das Platinen-Layout und den Code stehen kostenlos unter [2] zum Download zur Verfügung.

(120187)

Nachlese zu Teil 1 (Elektor September 2013)

Im ersten Teil wurde zu Bild 5 angemerkt, dass die Sendeleistung vom Wert von R3 abhängt. Richtig ist, dass sie vom Wert von R1 abhängt.

Weblinks

- [1] www.elektor.de/120187
- [2] www.elektor.de/120049
- [3] www.elektor-projects.com/120049
- [4] www.magusporta.com

Elektor Android EKG

Teil 3 Bau, Einstellungen und Betrieb

Hier kommt der letzte Teil der kleinen Serie zu diesem Projekt. Schon ungeduldig erwartet von unseren Lesern wurden die Stückliste und Tipps zum Aufbau. Hinweise zur Inbetriebnahme und zur korrekten Einstellung komplettieren das Ganze.

Von **Marcel Cremmel** (F), in Zusammenarbeit mit **Raymond Vermeulen** (Elektor-Labor)

Aufbauen leicht gemacht!

Der Bau des Elektor-Kardioskops (auch EKG-Interface genannt) ist nicht schwierig, wenn die bereits montierte und getestete Platine zum Einsatz kommt. Die aufgebaute Platine ist über den Elektor-PCB-Service [8] erhältlich. Die Schaltung, die als Schaltbild im ersten Teil des Beitrags [9] auf zwei Heftseiten verteilt ist, nimmt eine Platinenfläche von nur 10 · 6 cm ein (**Bild 19**). Die Platine ist maßgeschneidert für ein Gehäuse des Herstellers *Pactec* (siehe Stückliste), Befestigungsschrauben sind nicht erforderlich. Im Gehäuse befindet sich ein Batteriefach, Platine und Batteriehalter müssen nur noch miteinander verbunden werden.

SMD-Lötspezialisten können die ebenfalls lieferbare leere Platine in eigener Regie bestücken. Allerdings muss nicht nur das nötige Lötgerät vorhanden sein, fundierte Kenntnisse und Erfahrungen in dieser Löttechnik sind unabdingbar. Als Gleichspannungswandler (IC12) ist nur der Typ MCP1640BT geeignet, denn er produziert weniger Rauschen als andere Varianten. Dagegen ist das Bluetooth-Modul (Bild 4c in Teil 1) nicht an einen bestimmten Typ gebunden, solange das Modul das SPP-Protokoll unterstützt. Der verwendete Typ RN42 von *Roving Networks* hat die Eigenschaft, dass er mit einem Taster (S3) rücksetzbar ist. Die Schaltungsfunktionen lassen sich bequemer testen, wenn der Ferrit-Chip L3 zunächst nicht montiert oder gegebenenfalls demontiert wird. Die Kontaktleiste auf der Platine ist nur notwendig, wenn der Mikrocontroller später „in circuit“ neu programmiert werden soll.



Geeignete Tools sind zum Beispiel das PICKit2 oder PICKit3 von Microchip.

Die Funktionen der Drucktaster und LEDs sind auf der Frontplatte aufgedruckt (**Bild 20**): S1 stoppt den Betrieb und setzt den Mikrocontroller zurück, S2 löst den Start aus und S3 führt einen Reset des Bluetooth-Moduls herbei. Wenn Daten zum Android-Gerät übertragen werden, leuchtet D3 (TX) auf, und D4 zeigt den Status des Bluetooth-Moduls an. Die Elektroden und die zugehörigen Steckverbindungen müssen selbst angefertigt werden, wir kommen später darauf zurück.

Nach diesem unkomplizierten Zusammenbau kann das System gestartet werden. Wenn die Batterien eingelegt sind und S2 gedrückt wird, muss D4 langsam (2 Hz) blinken. Dies bedeutet, dass das Bluetooth-Modul „on air“ ist, es kann nun von anderen Bluetooth-Geräten erkannt werden.

Wir danken an dieser Stelle Aurélien Moulin, Praktikant im Elektor-Labor, für seinen unermüdlichen Einsatz beim Test und Debuggen der ersten Android-EKG-Prototypen.

Stückliste

Widerstände (0603, 1 %):

R1, R13, R15, R18, R19, R20, R33, R34, R54 = 10 k 0,25 W
 R2, R12 = 3k3
 R3 = 523 k
 R4 = 300 k
 R6 = 150 Ω
 R7...R11, R23 = 1 k
 R14, R53, R55, R56, R59...R62 = 100 k
 R5, R16, R17, R35 = 1 M
 R21, R36, R37, R38 = 330 k
 R22 = 100 Ω
 R24, R65 = 390 k
 R25, R29 = 47 M 5 %
 R26, R30 = 10 M
 R27, R31 = 2M2
 R28, R32 = 470 k
 R39, R40, R49...R52 = 47 k
 R41, R42, R45, R46 = 28k7
 R43, R44 = 1M4
 R47, R48 = 45k3
 R58 = 9k1
 R63, R64 = 27 Ω
 P1, P2 = Trimpoti 5 k, 20 % (Vishay TS53YJ502MR10)
 P3 = Trimpoti 2 k, 20 % (Vishay TS53YJ202MR10)

Kondensatoren (0603):

C1, C13 = 33 μ F / 6V3, Tantal (Case A)
 C2, C4, C7, C9, C12, C14 = 10 μ F / 6V3, X5R
 C3, C5, C6, C8, C10, C15, C21, C22, C23, C24, C39, C40, C41, C42, C43, C44 = 100 nF / 25 V, X7R
 C11, C16, C25, C26, C31...C38 = 1 μ F / 10 V, X5R
 C17...C20 = 1 nF / 50 V, X7R
 C27, C29 = 470 pF / 50 V, NPO, 5 %
 C28, C30 = 47 nF / 25 V, X7R, 5 %
 * Wert von C18...C20 ist 1 nF und nicht 100 nF wie im Schaltplan [9]

Induktivitäten:

L1 = 4,7 μ H 20 %, 0,5 A (Würth 744032004)
 L2...L9 = Ferrit-Chip, 30 Ω @ 100 MHz (Murata BLM18PG330SN1D)

Halbleiter:

D1, D2 = BAV99S
 D3, D4 = LED rot (PLCC-4)
 T1 = NMOSFET PSMN6R5-25YLC
 IC1 = PIC24FJ32GA002-I/SS (EPS 120107-41)
 IC2 = TPS60403DBVT
 IC3...IC7 = TLC2252AIDRG4
 IC8 = LMC6482AIMX/NOPB
 IC9 = DG4053 AEQ-T1-E3
 IC10, IC11 = CD74HC4052PW
 IC12 = MCP1640BT-I/CHY
 IC13 = LTC1981ES5#TRMPBF

Außerdem:

K1 = Buchsenleiste abgewinkelt, 5-polig 1-reihig, Raster 2,54 mm
 K2 = Stiftleiste 6-polig 1-reihig, Raster 2,54 mm
 K3 = Stiftleiste abgewinkelt 2-polig, Raster 2,54 mm
 MOD1 = Bluetooth-Modul RN42 (Roving Networks)
 S1, S2, S3 = Drucktaster (Omron SPNO B3FS-1052) mit Kappe (Omron B32-2010)
 Gehäuse Pactec PPL-2AA
 Platine, aufgebaut und getestet: 120107-91 oder Platine, leer: 120107-1

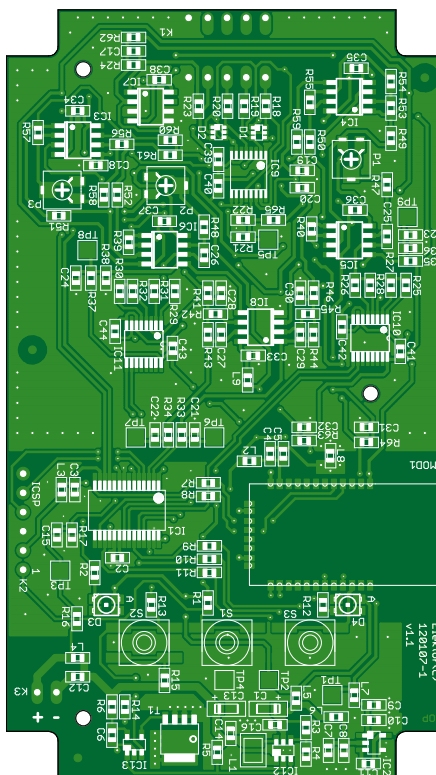


Bild 19.
Aufgebaute Platine für das Elektor-Kardioskop. Schaltungstechnisch ist wichtig, dass analoger und digitaler Teil konsequent voneinander getrennt sind.

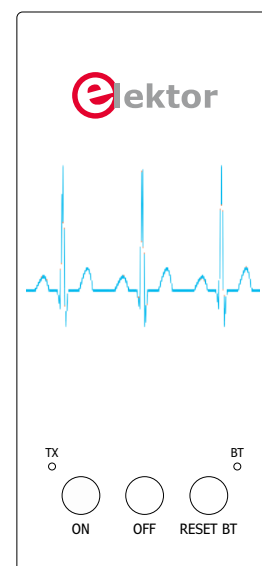


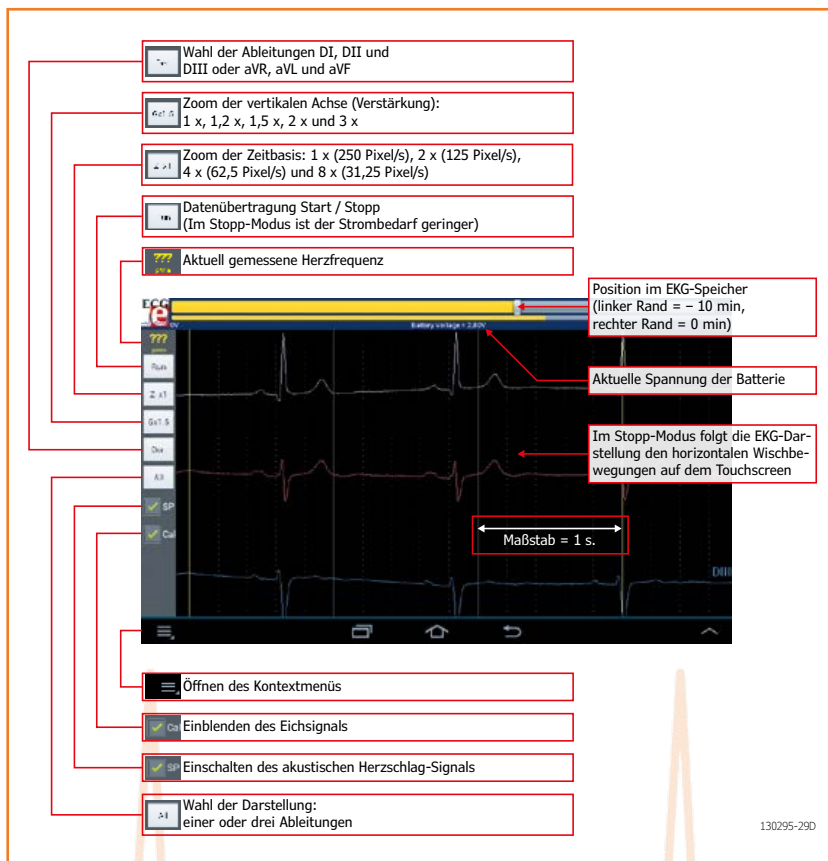
Bild 20.
Auf der Frontplatte befinden sich drei Taster und zwei LEDs.

Bild 21.

Auswahlliste für die Bluetooth-Verbindung zwischen dem Elektor-Kardioskop und dem Android-Gerät.



Bild 22. So wird das EKG auf dem Touchscreen des Android-Geräts dargestellt.



Android-Applikation

Der Installationsablauf ist Ihnen sicher geläufig: Laden Sie die gepackte App von [10] herunter, legen Sie die App im *Root* Ihres Android-Geräts ab und selektieren Sie die App im Dateimanager. Die Installation kann nur starten, wenn Sie dem Android-Betriebssystem im Menü Sicherheit (*Security*) erlaubt haben, so genannte *Non-Market-Apps* zu installieren. Eventuell richten Sie zusätzlich eine Verknüpfung ein. Falls Bluetooth auf Ihrem Android-Gerät abgeschaltet ist, gibt die App eine Warnmeldung aus. Ohne Bluetooth kann die App ihre Aufgabe natürlich nicht erfüllen. Das Kardioskop hat nun noch keine Bluetooth-Verbindung mit dem Android-Gerät. Um die Verbindung herzustellen, öffnen Sie das Menü und wählen unter den *Paired BT Devices* Ihr Bluetooth-Modul aus (**Bild 21**). Sobald die Verbindung hergestellt ist, muss die EKG-Darstellung in Bewegung kommen. Jetzt muss auch die Status-LED aufleuchten. Bei der ersten Inbetriebnahme müssen Sie eine PIN eingeben, sie lautet „1234“. Danach kennt Ihr Android-Gerät das Elektor-Kardioskop, die PIN wird nicht mehr abgefragt. Falls mehrere Bluetooth-Module des Typs RN42 in Funkreichweite sind, hilft die MAC-Adresse, das Modul auszuwählen. Auf dem Android-Gerät erscheinen die MAC-Adressen in der Liste der Bluetooth-Geräte. Das Kardioskop schaltet selbstständig ab, wenn innerhalb von fünf Minuten keine Bluetooth-Verbindung zustande gekommen ist. Die Funktionsbeschreibungen in **Bild 22** geben Auskunft über die Menüstruktur und die Software-Funktionen des Elektor-Kardioskops. Ferner können Sie auf Youtube einen Videoclip betrachten [11], er existiert allerdings vorläufig nur in französischer Sprache.

Gleichtaktunterdrückung

Der nächste Schritt ist das Einstellen der Gleichtaktunterdrückung (CMRR, *Common Mode Rejection Ratio*). Dazu brauchen Sie einen Funktionsgenerator sowie die Hilfskonstruktion aus **Bild 23**. Links im Foto sieht man eine BNC-Buchse, unmittelbar angelötet an eine 5-polige Stiftkontaktleiste. Stecken Sie die Stiftkontaktleiste in K1 auf der Kardioskop-Platine. Wenn sich die Taster unten befinden, muss Stift 1 rechts liegen. Die Stifte 1..3 sind die Eingänge der Signale RA, LA und LL, sie sind mit dem Signalanschluss der BNC-Buchse verbunden. Die Masse der BNC-Buchse liegt an Stift 4, während Stift 5 in der Luft hängt.

Stellen Sie den Funktionsgenerator so ein, dass er ein Sinussignal mit der Frequenz 50 Hz und der Amplitude 1 V abgibt. Schalten Sie das Kardioskop ein, starten Sie auf dem Android-Gerät die App ANDROECG, stellen Sie die Bluetooth-Verbindung her, und stellen Sie das Display auf die Signale DI und DII bei maximaler Verstärkung ein. Drehen Sie P1 und P2 so, dass das Signal möglichst Null wird. Bei der Verstärkung 10 darf das Signal kaum noch sichtbar sein (**Bild 24**). Wenn ein niederfrequenter Funktionsgenerator nicht zur Verfügung steht, können Sie sich wie folgt behelfen:

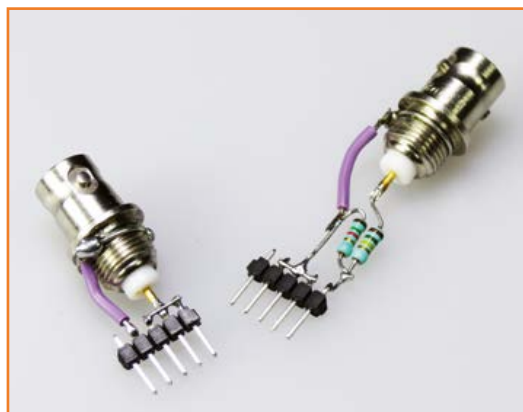


Bild 23.
Diese Hilfskonstruktionen dienen zum Einstellen der Gleichaktunterdrückung und der Verstärkung.

- Berühren Sie mit einem Finger einer Hand die Masse (GND) der Kardioskop-Platine.
- Berühren Sie mit dem Finger der anderen Hand den gemeinsamen Schaltungspunkt von RA, LA und LL. Das Ergebnis ist ein Common-Mode-Signal mit der Frequenz 50 Hz, das Ihr Körper von den umgebenden Netzspannungsleitungen auffängt.
- Stellen Sie das Kardioskop auf DI und DII ein.
- Mit einer „dritten Hand“ drehen Sie P1 und P2 so, dass das Signal möglichst klein ist.

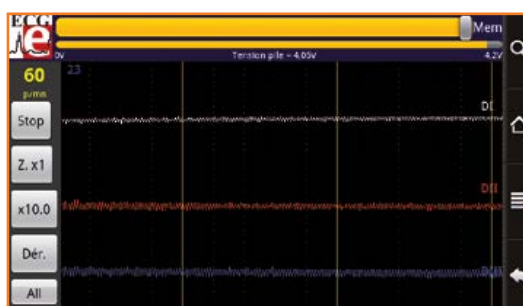


Bild 24. Die Amplitude des 50-Hz-Störsignals muss minimiert werden.

Verstärkung

Die Gesamt-Verstärkung in beiden Zweigen (über IC5a und IC5b) muss exakt gleich sein, denn aus DI und DII werden alle übrigen Ableitungen berechnet (siehe „Abbild biologischer Vorgänge“ in Teil 1). Das zum Einstellen benötigte Testsignal muss dem natürlichen, schwachen EKG-Signal ($\approx 1,4 \text{ mV}_{SS}$) möglichst ähnlich sein. Ein solches Signal entsteht, wenn das Signal des Funktionsgenerators durch den Spannungsteiler rechts in **Bild 23** abgeschwächt wird. Wie die Widerstände $1,5 \text{ k}\Omega$ und $150 \text{ k}\Omega$ geschaltet werden, geht aus **Bild 25** hervor. Diese Hilfskonstruktion stecken Sie auf K1 der Kardioskop-Platine auf, daran schließen Sie den Funktionsgenerator an. Auch hier muss sich Stift 1 rechts befinden, wenn die Taster nach unten zeigen. Stellen Sie den Funktionsgenerator auf ein Sinussignal 1 Hz, 140 mV_{SS} ein. Falls noch nicht vorhanden, bauen Sie die Verbindung mit dem Android-Gerät auf. Dort stellen Sie DI, DII und DIII mit der Verstärkung 2 ein, wie die erste Abbildung in **Bild 26** zeigt. Drehen Sie P3 so, dass DIII möglichst klein ist, denn die App berechnet $DIII - DI$, was bei $DI = DII$ gleich Null sein muss. Anschließend maximieren Sie die vertikale Spreizung mit ausschließlich DIII im Bild und minimieren DIII noch

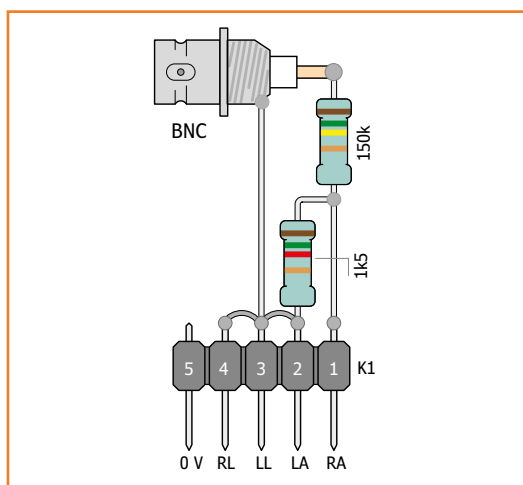


Bild 25.
Schaltung der Hilfskonstruktion rechts in Bild 23. Der Spannungsteiler schwächt das Signal des Funktionsgenerators um den Faktor 100 ab.

WICHTIGER HINWEIS

Das Produzieren und Anwenden medizintechnischer Geräte ist internationalen und nationalen Regelungen unterworfen [14]. Da das Elektor-Kardioskop kein Prüfsiegel trägt, ist es für den professionellen Einsatz nicht zugelassen. Das Gerät wurde für den experimentellen und edukativen Einsatz entwickelt. Das Elektor-Kardioskop darf ausschließlich an Batterien betrieben werden, es erfüllt die Bedingungen der Sicherheitsklasse III.

Weder der Verlag noch der Autor können eine Verantwortung für die Anwendung des Elektor-Kardioskop übernehmen.

Um die Verstärkung einzustellen, sollte die Amplitude des Testsignals möglichst groß sein, ohne dass der Verstärker in die Sättigung gerät. Vor der Kalibrierung sollten Sie dies prüfen; das Sinus-Signal darf im Terminal nicht verzerrt erscheinen (vor allem bei DI). Bei 1,4 mV (dem berechneten Wert für die passende Amplitude) dürfte es aber in jedem Fall noch Spielraum geben.

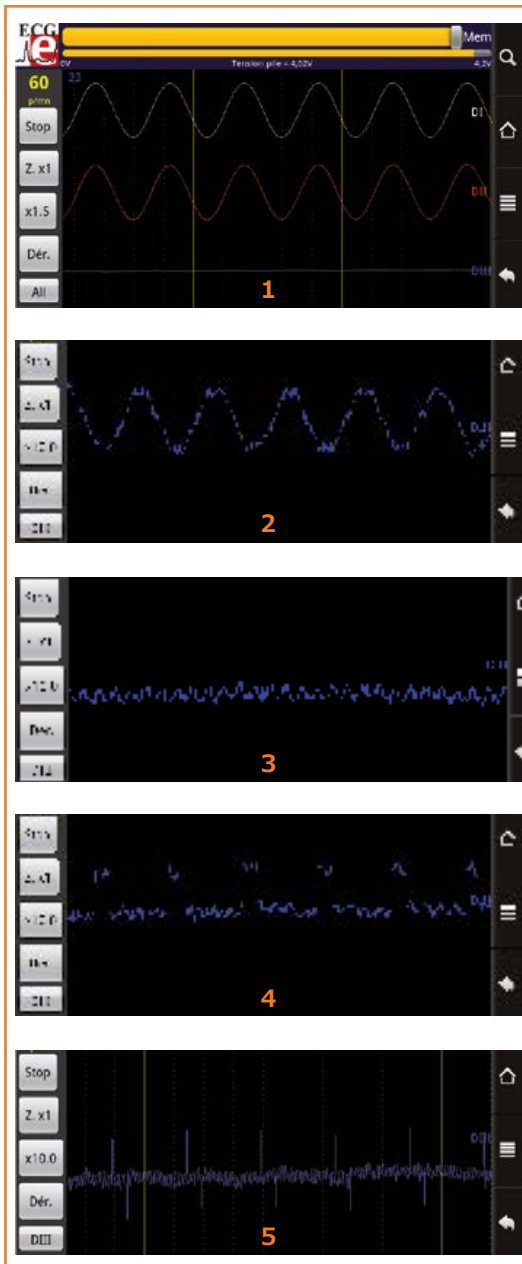


Bild 26. Die Screenshots helfen beim Einstellen der Verstärkungen von IC3 und IC4 (DI und DII). Signal DIII muss auf ein Minimum eingestellt werden

Bild 27. Ist die Checkbox „Cal“ aktiviert, wird im 10-Sekunden-Abstand das Kalibriersignal (2 Hz, 1 mV) eingeblendet.



einmal mit P3. Die zweite und dritte Abbildung in Bild 26 zeigen DIII vor und nach der Einstellung. Falls ein Funktionsgenerator nicht zur Verfügung steht, können Sie wie folgt vorgehen: Wenn Sie die Checkbox „Cal“ aktivieren, liegt am Eingang des Kardioskop ein periodisches Kalibriersignal (**Bild 27**, siehe auch Bild 9 in Teil 2). Sobald das Kalibriersignal erscheint, drehen Sie P3 so, dass dieses Signal möglichst klein ist. Die vierte und fünfte Abbildung in Bild 26 geben das Signal vor und nach der Einstellung wieder.

Damit ist das Elektor-Kardioskop einsatzbereit. Sie können nun die Android-App mit ihren diversen Schaltflächen ausprobieren (Bild 22), Sie können die Verstärkung und die Zeitbasis variieren sowie EKGs speichern und aus dem Speicher abrufen. Für das Sichtbarmachen menschlicher Herzsignale fehlen jedoch noch die Elektroden - und natürlich ein Proband!

Elektroden

Die elektrischen Signale, die am Körper mit Elektroden abgegriffen werden, sind um ein Vielfaches schwächer als durchschnittliche Signale. Die Qualität eines EKG hängt nicht nur von den Elektroden und ihren Kabeln ab, die Orte am Körper und das Präparieren der Hautpartien sind kaum weniger wichtig. Wenn die Taster des Kardioskop zum Probanden weisen, befindet sich die Masse der Elektroden-Anschlüsse auf der linken Seite (**Bild 28**). Das korrekte Anbringen am Körper erleichtern die standardisierten Farben der Leitungen und ihrer Anschlüsse:

- Rot = Rechter Arm, RA, *Right Arm*
- Schwarz = Rechtes Bein, RL, *Right Leg*
- Gelb = Linker Arm, LA, *Left Arm*
- Grün = Linkes Bein, LL, *Left Leg*

Elektroden aus der professionellen Medizintechnik, insbesondere Klammer-Elektroden, sind ziemlich kostspielig. Es gibt spezielle Varianten für Untersuchungen an Kindern, sie haben noch höhere Preise. Eine Alternative besteht darin, die Elektroden in eigener Regie anzufertigen. Dabei ist jedoch zu bedenken, dass die Elektroden und ihre Zuleitungen hervorragende Antennen für Störsignale aller Art bilden. Um die Störsignale auf ein praktikables Maß herabzudrücken, müssen die Kabel abgeschirmt sein. Die Kabelabschirmungen dürfen nur auf der Seite des Kardioskop mit Masse verbunden sein. An der Elektrode muss die Abschirmung isoliert werden, sie darf weder mit der Elektrode noch mit der Haut des Probanden in Berührung kommen. Abgeschirmte

Audiokabel sind elektrisch eine gute Wahl, sie halten jedoch der mechanischen Beanspruchung nicht lange stand. Die farbigen „Tie Wraps“ in Bild 28 tragen zur mechanischen Stabilität bei. Zusammen mit den Gegenstücken zu Bananensteckern (4 mm) sind die meisten professionellen oder halbprofessionellen Elektroden einsatzfähig (**Bild 29**). Ferner existieren EKG-Druckknopf-Adapter, Stecker-Adapter [12] sowie Bananenstecker mit Schrauben (RS Components 641-8053). Als Elektroden sind nickelhaltige Münzen aus der Zeit vor dem Euro wie beispielsweise die 1-DM-Münze gut geeignet. Das Anlöten einer Bananenstecker-Buchse ist zwar schwierig, doch wie **Bild 30** zeigt, nicht unmöglich. Die so entstandenen Elektroden können mit elastischen Klebe- oder Klettbandern an ihren Plätzen gehalten werden. Geeignete Orte sind der Puls an den Handgelenken und der Knöchel an den Füßen. Elektrisch leitendes Gel kann helfen, das Kontaktpotential und das Rauschen zu reduzieren, was der Signalqualität entgegenkommt.

Speichern und Abrufen

Der Speicher des Elektor-Kardioskop verhält sich wie ein Ringpuffer. Die Ableitungen DI, DII und DIII werden in einem zehn Minuten langen Zeitintervall aufgenommen oder wiedergegeben. Bei der Aufnahme werden die ältesten Werte von den aktuellen Werten überschrieben. Ein Pointer zeigt immer auf die letzten Abtastwerte der analogen Eingangssignale. Wegen der digitalen Speicherung stimmen die Aufnahmen exakt mit den Originalen überein. Die letzten Abtastwerte erscheinen immer rechts im Bild (siehe Bild 17 in Teil 2). Aufnahme und Wiedergabe werden über das Menü gestartet. Vor der Aufnahme muss ein individueller Dateiname vergeben werden (siehe Bild 18 im zweiten Teil des Beitrags [13]).

Weitere Funktionen

Wenn die Checkbox „Cal“ aktiviert ist, wird das EKG-Signal periodisch von einem Kalibriersignal unterbrochen (Bild 27). Die Amplitude des Eichsignals beträgt genau 1 mV. Das Kalibriersignal dient als Referenz für das aktuell gemessene EKG-Signal.

In der Android-App ist ein Algorithmus implementiert, der den Herzschlag (die Phase R) erkennt und die Herzfrequenz berechnet. Die Herzfrequenz wird oben links im Display angezeigt. Wenn die Herzfrequenz nicht berechnet werden kann, bleibt die Anzeige leer. Bei jedem erkannten

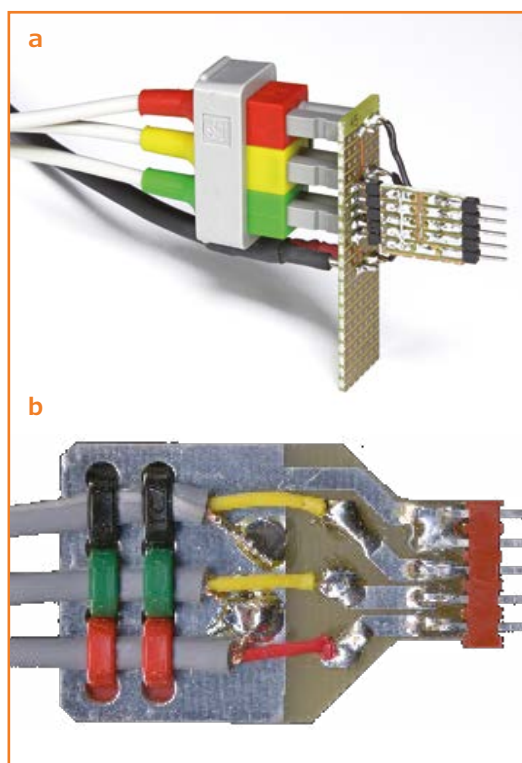


Bild 28. Dieser selbstgebaute Adapter dient zum Anschließen professioneller Elektrodenkabel. Er wurde für das Gameboy-EKG-Projekt aus dem Jahr 2006 konstruiert, das nur drei Elektroden hatte.

Herzschlag ertönt ein kurzes akustisches Signal (Beep). Falls das akustische Signal als störend empfunden wird, kann es über die Checkbox „SP“ abgeschaltet werden.



Bild 29. Professionelles EKG-Zubehör gehört nicht zu den Billigprodukten, die Qualität rechtfertigt jedoch die vergleichsweise hohen Preise.



Bild 30. Die gute alte D-Mark und auch ein Franc sind noch nicht wertlos. Mit aufgelöteter Bananenstecker-Buchse wird die Münze zur Elektrode für das Elektor-Kardioskop.

Zukünftige Funktionen

Ein Projekt wie das Elektor-Kardioskop ist niemals vollendet. Bei Redaktionsschluss existierte lediglich die App für das Betriebssystem Android. Diese App ist *Open Source*, was bedeutet, dass sie frei auf andere Plattformen wie iPhone, Mac oder PC portierbar ist.

Wenn Sie eine Version geschrieben haben, die Sie veröffentlichen möchten, bitten wir um eine kurze Nachricht. Inzwischen entwickeln wir unsere Android-App weiter, wir haben vorläufig folgende Ziele:

Digitale (durch Software realisierte) Bandsperre für 50 Hz oder 60 Hz, Speichern der EKGs in einer Cloud über eine API (*Application Programming Interface*) von Google.

Ferner entwickelt der Autor eine Applikation für Windows, die EKGs auf einer SD-Speicherkarte ablegt. Das Ziel ist hier, gespeicherte EKGs über eine Bluetooth-Schnittstelle unmittelbar betrachten zu können.

Für die fernere Zukunft ist eine aktive HTML-Seite geplant, die in eine individuelle Homepage eingebunden werden kann. Dort wird das EKG wiedergegeben, das vom Elektor-Kardioskop hochgeladen wird. Ein behandelnder Hausarzt oder Kardiologe kann das EKG über das Internet betrachten und bei Unregelmäßigkeiten die notwendigen Maßnahmen einleiten.

(130295)gd



Marcel Cremmel ist als Hochschullehrer in den Fachbereichen Elektrotechnik und Elektronik am Lycée Louis Couffignal in Straßburg (Frankreich) tätig. <http://electronique.marcel.free.fr/> marcel.cremmel.llc@free.fr

Weblinks

- [8] Einsatzfertiges Modul:
www.elektor.de/120107
- [9] Elektor Android EKG Teil 1, Elektor Juli/August 2013, S. 10, www.elektor.de/120107
- [10] Applikation ANDROECG.apk
www.elektor.de/130295
- [11] Videoclip (französischsprachig)
www.youtube.com/watch?v=wCGXKk-V7EJ4
Eine englischsprachige Version ist in Vorbereitung. Kommentare erreichen den Autor unter d.meyer@elektor.fr über die französischsprachige Seite.
- [12] Elektroden: www.praxisdienst.de
- [13] Elektor Android EKG Teil 2, Elektor September 2013, S. 32, www.elektor.de/130227

- [14] Sicherheitsnormen und -regelungen
<http://homeusemedicaldevices.com/hum.html>

Klick-Adapter für Bananenstecker

<http://goo.gl/5WWYFu> oder
www.praxisdienst.com/en/Diagnosis/Specialised+diagnosis/ECG+devices+and+accessories/oxid+oxid/Press+stud+adapter+for+ECG+red.html

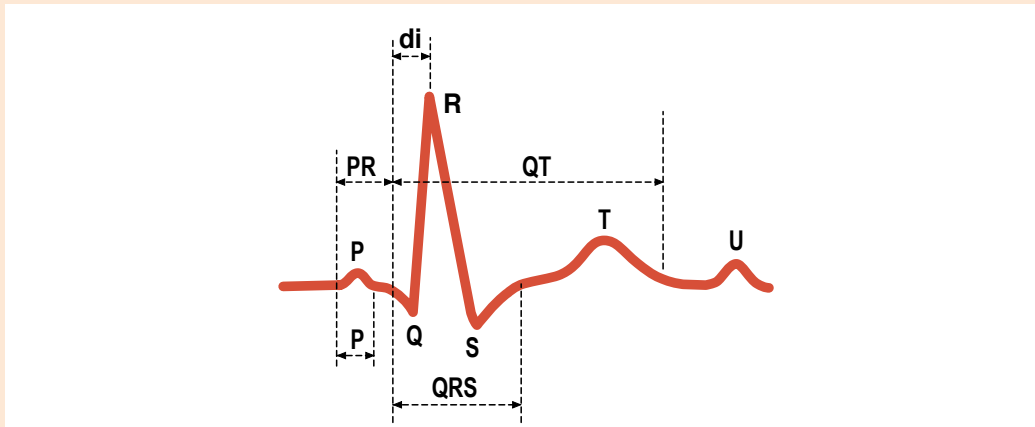
Klammer-Elektroden

<http://goo.gl/bZIKXf> oder
<http://www.praxisdienst.com/en/Diagnosis/Specialised+diagnosis/ECG+devices+and+accessories/oxid+oxid/Limbs+Clip+electrode+Adult+red.html>

Elektrokardiogramm

Rhythmus und Frequenz

Die Herzfrequenz eines Menschen in Ruhe liegt normalerweise zwischen 50 und 100 Schlägen in der Minute (*beats per minute, bpm*). Herzfrequenzen unter 50 bpm werden als *Bradycardie* bezeichnet, bei Herzfrequenzen über 100 bpm sprechen die Kardiologen von *Tachycardie*.



P-Schwingung: Depolarisation der Vorkammern

Die P-Schwingung dauert in der Regel 0,1 s oder kürzer, die Amplitude beträgt 0,25 mV oder weniger. Das entspricht einem Viertel des Kalibriersignals, das im Elektor-Kardioskop implementiert ist. Auf den Ableitungen DII, DIII und aVF erreicht die P-Schwingung stets ihr Maximum, sie ist auf DI und DII immer positiv, auf aVR ist sie immer negativ.

PQ-Segment

Das PQ-Segment ist das Intervall vom Beginn der P-Schwingung bis zum Beginn der Q-Spitze. Dieses Intervall dauert normalerweise 0,12...0,20 s, das ist die Zeit, in der das Blut aus der Vorkammer in die Hauptkammer strömt. Bei körperlichen Anstrengungen verkürzt sich das Intervall. PQ-Zeiten unter 0,12 s können auf Herzanomalien hindeuten.

Amplitude des QRS-Komplexes

In frontalen Ableitungen kann die Amplitude stark variieren. Die Amplitude liegt im normalen Bereich, wenn sie 0,5 V oder mehr beträgt. Das entspricht der Hälfte des im Elektor-Kardioskop zuschaltbaren Kalibriersignals.

Länge des QRS-Komplexes

Der QRS-Komplex dauert im Mittel 0,08 s, er darf höchstens 0,12 s lang sein. Bei höheren Werten sind Herzrhythmus-Störungen nicht auszuschließen.

Repolarisierung der Herzkammern: ST-Segment, T- und U-Schwingung

ST-Segment heißt der Abschnitt vom Ende des QRS-Komplexes bis zur T-Schwingung.

Die Form der T-Schwingung ist asymmetrisch, der Anstieg ist weniger steil als der Abfall. Die Richtung stimmt mit dem QRS-Komplex überein. Auf den Ableitungen DI, DII, DIII und aVF ist die T-Schwingung normalerweise positiv. Auf Ableitung DIII kann die T-Schwingung auch negativ sein.

Die U-Schwingung folgt der T-Schwingung, sie hat die gleiche Richtung, jedoch eine schwächere Amplitude. Welche Signifikanz die U-Schwingung hat, wird noch von der Fachwelt diskutiert.

Das QT-Intervall, vom Beginn des QRS-Komplexes bis zum Ende der T-Schwingung, variiert als Funktion der Herzfrequenz. Wenn die Herzfrequenz 60 bpm beträgt, ist das QT-Intervall ungefähr 0,4 s lang.

The background of the advertisement is a tablet displaying the cover of Elektor magazine. The cover features the magazine's logo, which consists of a red stylized 'e' inside a circle, followed by the word 'elektor' in a large, bold, sans-serif font. Below the title, there are several headlines in German, including 'Für LEDs, Moto...', 'Netzteile-Recycling', 'Elektronik einfach steuern', 'Fallsender für Flugmodelle', 'Resonanzmeter', 'FPGAs pro...', and 'Neues aus dem Labor'. There is also a small graphic of a bar chart with red, green, and blue bars. The magazine cover is partially obscured by three text boxes.

**Lesen Sie jetzt Elektor
auf Ihrem Tablet!**

**Einzelheft: nur 7,99 €
Doppelheft: nur 14,99 €**

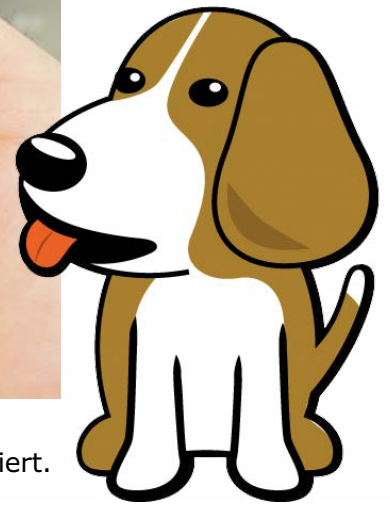
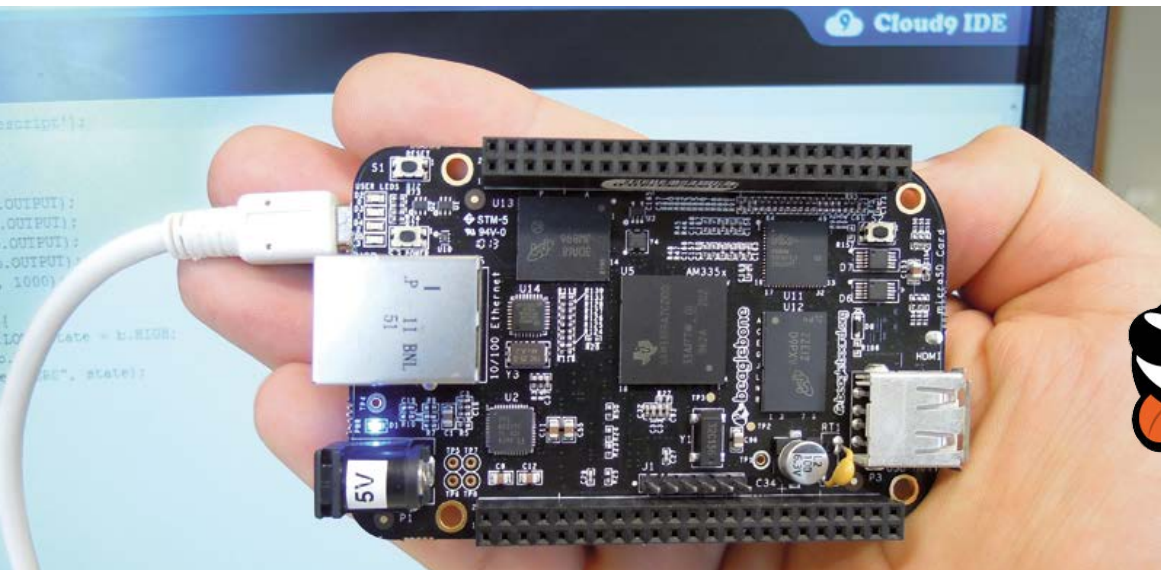
**Elektor-Mitglieder erhalten
jede App-Ausgabe kostenlos!**

Elektor-App herunterladen:

- www.elektor.de/ios-app
- www.elektor.de/android-app

BeagleBone Black

Der Raspberry-Pi-Rivale



BeagleBone Black: Eine neue Plattform wird dem Embedded-Publikum präsentiert. Gerade ist sie (endlich) auf unserem Labortisch gelandet. Auf den ersten Blick eine vielversprechende Plattform mit leistungsfähiger Hardware und einer Menge Potenzial. Wird Raspberry Pi nun zum Hundefutter?

Vor einiger Zeit wurde BeagleBone Black (BBB) vorgestellt. Dieser „1GHz 512MB Open-Hardware-Embedded-Computer mit On-board 2GB eMMC Flash“ ist der Nachfolger des erfolgreichen kreditkartengroßen „BeagleBone“-Linux-Computers. BBB verfügt über einen Sitara AM3359A-ZCZ100 ARM Cortex-A8 Prozessor mit 2.000 MIPS, eine SGX530 Grafik-Engine mit 20 MPolygone/s, 512 MB DDR3L SDRAM mit 800 MHz, 10/100 Mbit Ethernet, ein USB 2.0 Host- und Client-Interface, µSD-Slot, µHDMI, 65 digitale und 7 analoge Eingänge, 8 PWMs, 4 Timer, 4 bis 5 serielle UARTs, 2 I²C-Ports sowie 2 SPI-Ports. Und er kostet mit rund 45 € sogar nur etwa die Hälfte seines Vorgängers BeagleBone. Damit stellt BBB eine ernsthafte Konkurrenz zum Raspberry Pi dar.

In unserer IT-Abteilung haben wir ein schönes Fernsehgerät mit HDMI-Eingang beschlagnahmt, einen Sony KDL-32EX650. Dann haben wir ein µHDMI-nach-HDMI-Kabel bestellt, eine Maus

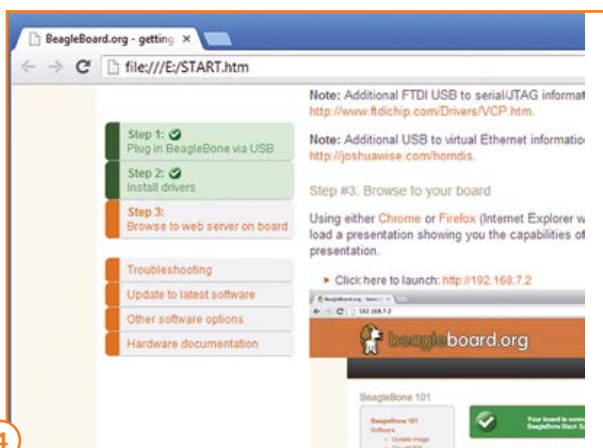
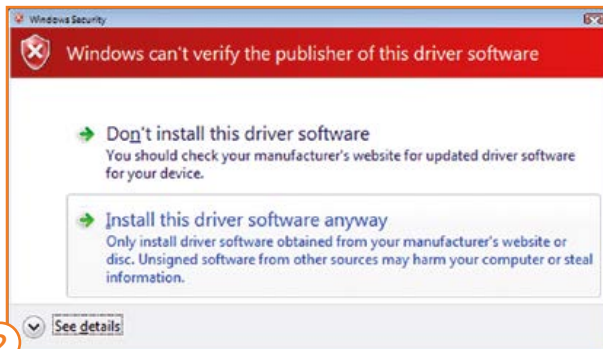
gefangen und alles eingestöpselt. Bequemerweise ist unser TV auch mit einem USB-Host-Port ausgestattet, über den wir den BBB versorgen. BBB wird derzeit mit der Linux-Distribution Ångström auf dem internen eMMC-Flash-Speicher geliefert und bootet direkt in das Graphical User Interface (GUI). Auspacken und im Stand-alone-Modus loslegen, dies hat BBB dem Raspberry Pi voraus, wo man erst einmal das Betriebssystem auf die SD-Karte kopieren muss.

Sicher, Sie können den BBB auch an ihren Windows-PC anschließen, um eine Anwendung zu entwickeln. Das geht so einfach wie's Brezelbacken. An einer freien USB-Buchse eingesteckt, startet Windows-Autoplay. Lassen Sie sich die Dateien anzeigen und starten Sie *START.htm*, wie es auf dem kleinen Beipackzettel der BBB-Box steht.

Der Web-Browser wird geöffnet und die Treiber für den BBB werden von der (Offline-)Webseite

Von **Thijs Beckers**
(Elektor Redaktion)

•Projects



(1) installiert. Alle benötigten Dateien befinden sich auf dem Board, der wie ein USB-Stick eingesteckt wird. Eine Internet-Verbindung ist nicht notwendig. Trotz der vier (!) Sicherheitswarnungen (2) war es überhaupt kein Problem, die Treiber zu installieren (3). Auf der Website können Sie den Fortschritt der Installation verfolgen (4). Jetzt sind Sie bereit, sich mit dem BBB-Webserver mit der URL 192.168.7.2 (Chrome oder Firefox) zu verbinden.

Die Entwicklung einer eigenen Anwendung ist tatsächlich simpel. Die Cloud9 IDE (5) mit der Adresse 192.168.7.2:3000 (in der Adresszeile des Web-Browsers eingeben oder auf den Link „BBB Home Page“ klicken) läuft im Web-Browser und muss nicht auf dem PC installiert sein. Sie nutzt die BoneScript-JavaScript-Bibliothek, die verdeutlicht, wie man dem BBB „physikalische“ Aufgaben überträgt. Einige Beispiele ebnen den Weg für die Entwicklung eigener Anwendungen. Da die Lernkurve ziemlich steil nach oben führt, ist es recht nützlich, wenn man schon einmal in C oder Java programmiert hat. Mehrere Shields... (Entschuldigung!) Capes stehen zur Verfügung. Grundsätzlich handelt es sich um „Tochterboards“, die an die Steckverbinder des BBB angeschlossen werden und Funktionalität hinzufügen wie 3D-Drucker, CAN-Bus, 7“-LCD-Touchscreen, VGA, Kamera, Audio, Radar und vieles mehr. Viele der alten Capes für den (ersten) BeagleBone sind kompatibel mit dem BBB. Nur muss man darauf achten, ob die BBB-Linux-Distribution ein Update benötigt. Wir wollten unser BeagleBone LCD7-Cape rev A2 anschließen, das vom BeagleBone Kamera Demo Kit noch bei uns herumlag und nach [1] kompatibel mit BBB sein sollte, mussten aber sicherstellen, dass wir die Ängström-Version vom 20.06.2013 hatten, um zu verhindern, dass der BBB beschädigt (!) wird. Unsere Version datierte vom 06.06.2013, so dass wir ein Update benötigten.

Ein Update auf die neueste Linux-Distribution durchzuführen, ist zwar einfach, aber irgendwie unsicher, da es keine Rückmeldung über den Prozess gibt. Wir aktualisierten unseren BBB mit dem microSD-Kartenverfahren: Zunächst das neueste Image von [2] laden, mit 7-zip [3] das Image extrahieren und mit Diskimager [4] das extrahierte Bild auf eine microSD-Karte (mindestens 4 GB) schreiben. BBB mit einem 5 V/1 A-Steckernetzteil versorgen und S2 gedrückt halten, bis ein oder mehrere User-LEDs aufleuchten (microSD-Karte eingelegt, Ethernet- und USB-Geräte ausgesteckt!). Wenn Sie diesen Anweisungen *genau* (!) folgen, wird die Onboard-eMMC mit dem Image, das auf der microSD-Karte ist, geflasht. Es gibt keine Rückmeldung über das (HDMI-) Display, nur das User-LED-Benutzer-Array flackert (aber das macht es auch, wenn BBB an einen USB-Port angeschlossen ist...). Wenn das Update endlich fertig ist (es dauert circa 40 Minuten), startet der BBB mit der neuen Version (6, jawohl, da vorne ist BBB!).

Dann haben wir den BBB an das LCD-7-Cape angeschlossen, eingeschaltet und gewartet, dass das GUI erscheint. Aber nichts war zu sehen, nichts passierte auf der Mattscheibe, bis wir die noch angeschlossenen Kamera- und Wetter-Capes vom LCD abkoppelten. Nun war alles makellos (7), inklusive Kalibrierungssequenz für den Touchscreen. Wir haben uns mal über die kompatiblen Capes schlau gemacht und es stellte sich heraus, dass sowohl das Kamera- als auch das Wetter-Cape (noch) nicht kompatibel mit der BBB waren.

Hoppla! Zum Glück wurde nichts beschädigt, alles war gut. Unser erster Eindruck vom Board ist sehr positiv. Was man für nur 45 € bekommt, ist erstaunlich. Obwohl RPi wohl besser geeignet ist, um einen eigenen Heimkino-Media-Player zu realisieren, scheint BBB stärker und bietet eine Menge I/Os und alle Konnektivität, die man sich nur wünschen kann. Obwohl man einen USB-Hub anschließen muss, um mehr als ein USB-Gerät (zum Beispiel Tastatur und Maus) zu betreiben, da BBB mit nur einem USB-Host-Anschluss aufwarten kann.

Zusammenfassend: Raspberry Pi ist für Anfänger gut geeignet und hat zurzeit die größere Fangemeinde, BeagleBone Black ist aber eine ernsthafte(re) Konkurrenz. Ein wenig schwieriger zu meistern, aber die Hardware-Vorteile gegenüber dem RPi sind erheblich. Und die User-Community [5] hilft Ihnen gerne, wenn Sie Probleme mit dem BBB haben.

Übrigens: Demnächst ist ein Adapter-Cape im Elektor-Shop erhältlich, das selbst wieder GnuBlin/EEC-Steckverbinder bietet [6]. Auf diese Weise lassen sich die GnuBlin-Linux-Erweiterungsboards [7] an den BBB anschließen.

(130279)

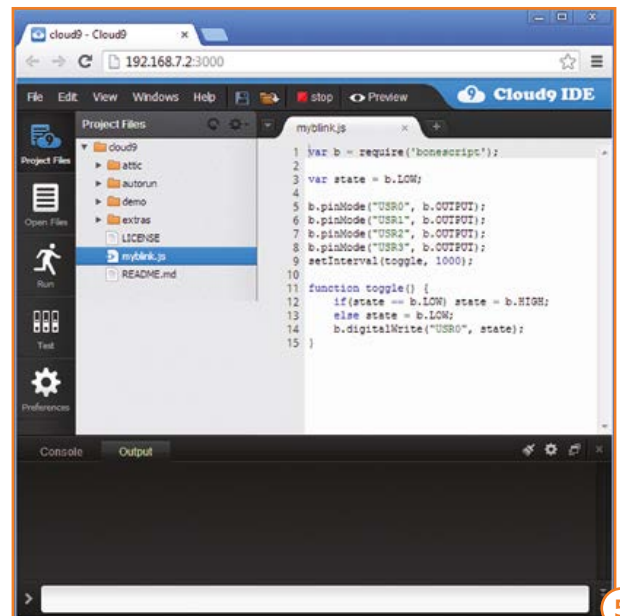
Weblinks

- [1] <http://beagleboardtoys.info>
- [2] <http://beagleboard.org/latest-images>
- [3] www.7-zip.org
- [4] <https://wiki.ubuntu.com/Win32DiskImager>
- [5] <http://beagleboard.org/Community>
- [6] www.elektor.de/gnublin
- [7] www.elektor.de/130212

Was möchten Sie mit dem BeagleBone Black tun?

Möchten Sie ein Media-Center bauen, wie es so viele mit dem Raspberry Pi getan haben? Oder schwebt Ihnen ein kostengünstiges Embedded-Entwicklungs-Board vor? Oder ein Sicherheitssystem in Ihrem Auto oder dann doch ein mobiles Media-System? Oder vielleicht ein einfacher Temperatur- und Feuchte-Monitor für Ihre Tomatenplantage (fernablesbar natürlich), oder realisieren Sie endlich das Hirn eines Roboters? Vielleicht haben Sie schon eine Anwendung entwickelt und möchten es mit unserer Community teilen? Wenn Sie glauben, dass Ihre raffinierte 45-€-Mini-Computer-Anwendung auch für andere von Interesse sein könnte, zögern Sie nicht, uns zu kontaktieren und mit etwas Glück wird Ihr BBB-Design in Kürze mehr als 250.000 Elektronik-Begeisterten bekannt sein...

Ausgangspunkt: www.elektor-labs.com.



5



6



7

D/A-Wandlung mit variabler Spannung

8-bit-Wandlung verfeinert

Von
Olivier Schrevens (F)

Sobald die erforderliche Auflösung 8 Bits übersteigt, werden die Wandler zwischen der analogen und der digitalen Welt deutlich teurer. Mit dieser Schaltung zeigen wir, wie man ohne zusätzliche Kosten die Auflösung der gängigen preiswerten D/A-Wandler verbessern kann. Der Ausgang des einen Wandlers dient dabei als programmierbare Spannungsreferenz des anderen Wandlers.

Die Idee zu dieser Schaltung entstand aus dem Wunsch heraus, mittels Mikrocontroller über D/A-Wandler mit einem Widerstand am Ausgang verschiedene Typen von Drehspulinstrumenten ansteuern zu können. Bereits der erste Schaltungsentwurf funktionierte an einem ATmega8. Bei sehr präzisen Instrumenten reichte allerdings die Anzahl der verwendeten Bits nicht aus, um einen Vollausschlag der Anzeigenadel zu erreichen. Um nicht immer den Ausgangswiderstand an das jeweilige Drehspulinstrument anpassen zu müssen, entwarf der Autor die hier beschriebene Schaltung, welche die verfügbare Auflösung der 8-bit-Wandler verbessert. Der Kniff besteht darin, die Referenzspannung zu ändern, um andere Ausgangsspannungsbereiche zu erhalten.

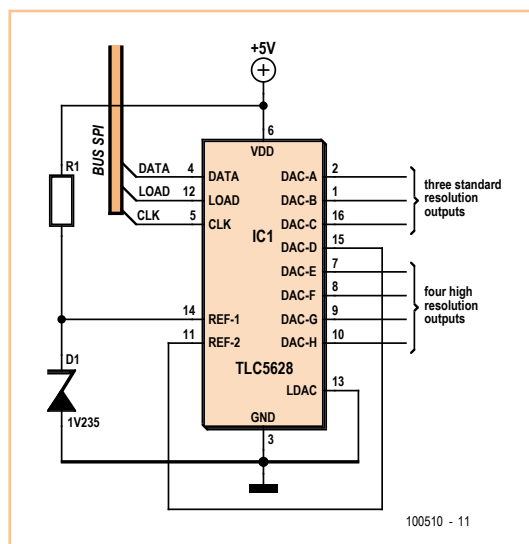
Das Hauptkriterium bei der Wahl der D/A-Wandler war ein Ausgangsstrom von mindestens 1,5 mA, um den Einsatz von externen Operationsverstärkern zu vermeiden. Für einen leichteren Aufbau war außerdem ein DIL-Gehäuse wichtig. Die Wahl fiel auf den TLC5628 von Texas Instruments. Bei diesem Bauteil handelt es sich um einen achtfachen 8-bit-Wandler in einem DIP16-Gehäuse, der pro Kanal 2 mA liefern kann. Jeder Kanal enthält einen 256-stufigen Spannungsteiler. Der TLC5628 bietet die hier entscheidende Option, den Ausgangsspannungsbereich zu verdoppeln (Bild 1).

Der Baustein lässt sich von gängigen Mikrocontrollern sehr einfach über einen seriellen SPI-Bus steuern. Das Befehlsformat besteht aus acht Datenbits, drei Auswahlbits für jeden der acht integrierten Wandler und schließlich dem Auswahlbit für den Ausgangsspannungsbereich (einfach oder doppelt). Die SPI-Daten werden für alle Wandler nacheinander ausgegeben und können in einem Register (latch) gespeichert werden. Durch Takten des LDAC-Eingangs werden dann alle Daten gleichzeitig übernommen. Diese Option wird hier nicht benutzt. Mehr Details ergeben sich aus dem Datenblatt zum Baustein.

Wie das Schaltbild zeigt, verfügt der TLC5628 über zwei getrennte Referenzspannungs-Eingänge (Pin 11 und Pin 14) für je eine Gruppe von vier Wandlern.

Die externe Referenzspannung für die ersten vier Wandler – hier 1,235 V – liefert D1 (LM385). Sie bildet mit R1 einen Spannungsteiler, dessen anderes Ende an Masse liegt. Jeder Abgriff des

Bild 1.
So beschaltet erlaubt es der TLC5628 gleichzeitig – mit 256 Konversionsschritten – drei Spannungsbereiche abzudecken (0 V bis 1,235 V, 0 V bis 2,47 V und 0 V bis 4,94 V).



Teilers liefert also eine Spannung, die direkt aus der Referenzspannung folgt.

Diese Besonderheit macht es möglich, einen Ausgang des ersten D/A-Wandlers als Referenzspannung für die zweite Wandlergruppe zu verwenden. Mit der so programmierbaren Referenzspannung lässt sich die Auflösung der zweiten Wandlergruppe erhöhen.

Die drei anderen Ausgänge der ersten Gruppe können wie bisher mit der normalen 8-bit-Auflösung genutzt werden. In Kombination mit der Option, die Ausgangsspannung jedes Wandlers des TLC5628 zu verdoppeln, deckt die 8-bit-Auflösung ausgehend von der ursprünglichen Referenz 1,235 V schließlich drei Spannungsbereiche ab. Kombiniert man also das Bit RNG (range) des Kanals, der die Referenzspannung liefert, und dasjenige des gerade benutzten Kanals, erhält man die folgenden drei Bereiche:

- 0 V bis 1,235 V
- 0 V bis 2,47 V
- 0 V bis 4,94 V

Dies entspricht durchaus einer Erhöhung der Auflösung. Wir verfügen jetzt über sieben analoge Ausgänge: Vier Ausgänge mit „hoher Auflösung“ und drei Ausgänge mit der normalen 8-bit-Auflösung.

Andere Wandler-Typen verfügen über ähnliche Eigenschaften wie der hier verwendete Baustein und sind daher ebenfalls für diese Schaltung geeignet. Kompatibel sind beispielsweise die Wandler TLV5628, TLC5620, TLV5620 sowie auch der TLV5604. Allerdings hat das Kaskadieren zweier Wandlers den Nachteil, dass sich die jeweiligen Ungenauigkeiten summieren können.

Durch seine Architektur erscheint dieser Wandler eher wie ein programmierbares Potentiometer als ein klassischer D/A-Wandler mit R2R-Netzwerk. Das Testprogramm [1] liefert in einer Schleife für vier Sekunden jede der drei Maximalspannungen an Ausgang E (Pin 7). Der SPI-Bus arbeitet in Modus 3 und wird mit 11,0592 MHz/4 getaktet. Eine Anzeige über den Ablauf des Programms wird über den seriellen Port ausgegeben (38400 Baud, 8N1).

(100510)

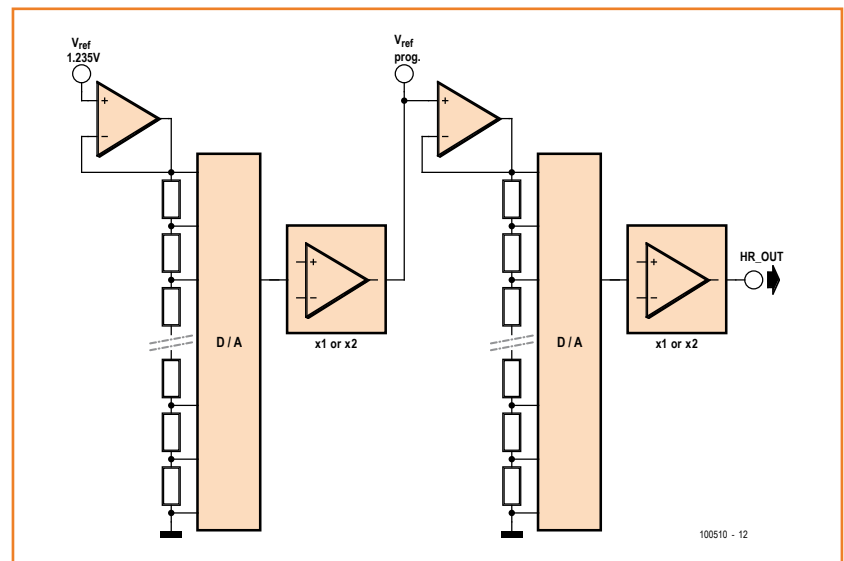
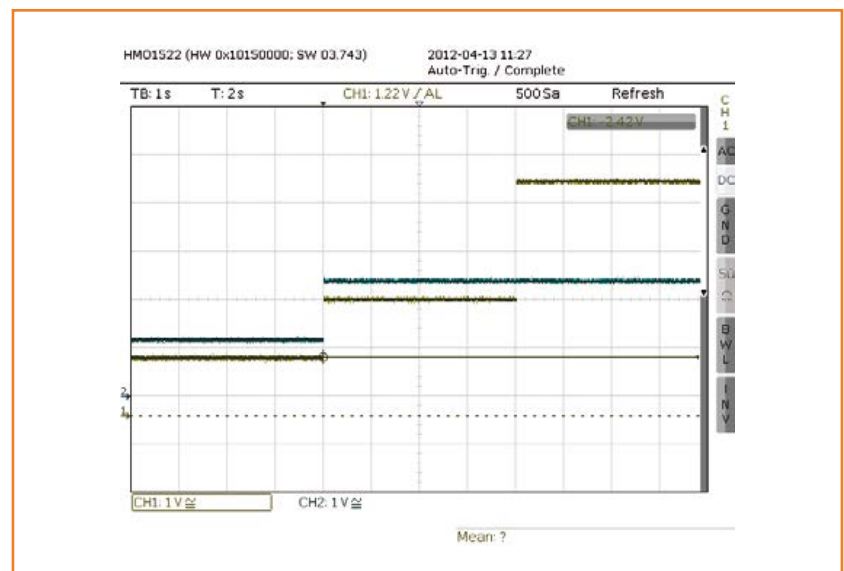


Bild 2.
Das Prinzip der D/A-Wandler-Kaskade:
Der Ausgang des ersten Wandlers dient als Referenzspannung für den zweiten Wandler.

Bild 3.
Ergebnis des Testprogramms, V_{ref2} und V_{out} werden angezeigt.
Links $V_{out} = 1,235 \text{ V}$ und $V_{ref2} = 1,235 \text{ V}$, Verstärkung = 1
Mitte $V_{out} = 2,47 \text{ V}$ und $V_{ref2} = 2,47 \text{ V}$, Verstärkung = 1
Rechts $V_{out} = 4,94 \text{ V}$ und $V_{ref2} = 2,47 \text{ V}$, Verstärkung = 2



[1] www.elektor.de/100510



Radiometer PHM22 / PHA928a zur Bestimmung von pH, O₂ & CO₂ im Blut

**Bitte warten
Sie auf Ihre
Blutwerte...**

Von **Seppo Lindeman**
(Finnland)

Ist das Instrument etwa lindgrün? Nein. Eher blassgraugrün. Dann muss es von der Firma Radiometer aus Kopenhagen sein, denn das war deren (einzige) Lieblingsfarbe während der über 50 Jahre im 20. Jahrhundert, in denen hochqualitative Messgeräte hergestellt wurden. Das erste Mal kam ich 1961 mit Radiometer in Berührung, als ich meinen ersten Job bei den Helvar-Werken für TV- und Radiogeräte in Helsinki antrat. Da gab es nämlich viele grüne Röhrenvoltmeter und Signalgeneratoren von Radiometer. Sie waren leicht zu finden, da sie sich gerne in der Nähe

von Steckdosen zusammenrotteten. Kurze Zeit später fiel mir dann auf, dass alle Geräte von Radiometer dieses schicke Grün trugen. Ob elektronisches Messgerät oder medizinische Analyse-Apparatur, das war egal.

Es war die Zeit, in der:

- Transistoren begannen eine Konkurrenz für Röhren zu sein und noch niemand etwas von ESD (ElectroStatic Discharge) gehört hatte,
- Germanium-Transistoren häufig einen mysteriösen Tod starben und
- Röhren-Fans gerne diesen Witz erzählten,



1



2



3

dass der einzige Vorteil eines Transistors gegenüber einer Röhre der sei, dass defekte Transistoren weniger Platz in der Mülltonne brauchten...

Das waren noch Zeiten!

Graugrünes Arsenal

Die hier beschriebene Apparatur für die Messung der Blutwerte $\text{pH}/\text{O}_2/\text{CO}_2$ bestand nicht aus einem einzelnen Messgerät — sie war aus mehreren Teilen zusammengesetzt. Die ersten Geräte stammen aus den 1950er Jahren. Die hier beschriebene Technik hingegen wurde hauptsächlich in den 1960ern hergestellt. Die Instrumente gab es in verschiedenen Ausführungen als „Mobilgerät“ (auf Rollwagen montiert) oder für den stationären Betrieb. Die Konstellation von **Bild 1** besteht aus dem Blut-pH-Meter des Typs PHM22t mit Glas- und Kalomelelektroden, einer Mikroelektroden-Einheit, dem Sauerstoffmonitor PHA928a mit pO_2 -Elektrode und der thermostatisch geregelten Zelle D616 sowie einem Tonometer, zwei Verdampfern aus Glas, einem Thermostaten des Typs VTS13, einem „Meter Scale Expander“ und zwei Gasflaschen.

pH-Meter 22

Die Zentraleinheit der ganzen Apparatur ist sicher das PHM22t von 1966, das in **Bild 2** präsentiert wird. Drin stecken fünf Röhren (siehe **Bild 3**). Die zugehörige Schaltung ist teilweise in **Bild 4** zu sehen. Man erkennt grob sieben Sektionen: Eingangsverstärker, Chopper, AC-Verstärker, Demodulator, Meter, Netzteil und der Spannungsteiler zur Kompensation. Die drei .jpg-Dateien der kompletten Schaltung stehen unter [2] zum Download bereit.

Die Ausgangsspannung der Glas-Elektrode liegt bei 61,54 mV/pH bei 37 °C. Die Auflösung beträgt 0,001 pH entsprechend 61 μV . Bei diesen niedrigen Spannungen ist es ziemlich schwer, einen Röhren-DC-Verstärker stabil zu bekommen. Zudem ist der Ausgangsstrom der Glas-Elektrode so gering, dass der Verstärker eine Eingangsimpedanz im Bereich von 50...500 M Ω haben muss. **Bild 5** zeigt das Funktionsprinzip des älteren Radiometer-pH-Meters PHM12. Die Röhren bilden hier einen Verstärker mit hohem Verstärkungsfaktor. Außerdem gibt es da noch ein Weston-Normalelement. Vor einer Messung sollte man den Test-Knopf betätigen und die Position der Nadel des analogen Instruments auf der Skala markieren, um die Drift des Verstärkers zu kompensieren.

PHM22t-Innenleben

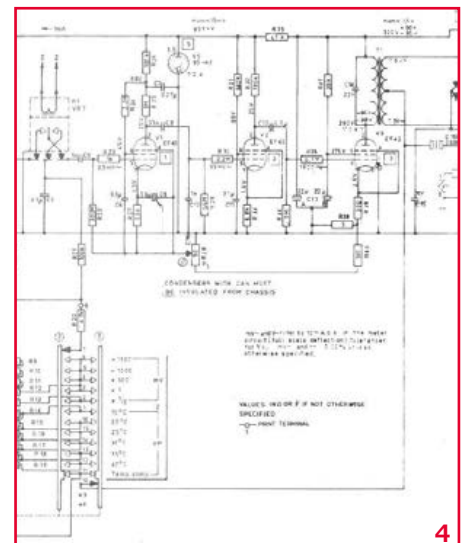
Der Eingangsverstärker ist chopper-stabilisiert. Der Chopper wandelt das DC-Signal am Eingang in ein AC-Signal, das einfacher und ohne Drift verstärkt werden kann. Der Chopper ähnelt dem Brown-Konverter, der in der Retronik-Ausgabe vom Mai 2013 [1] abgebildet wurde. Der originale Chopper war übrigens mechanisch. Später wurde er durch einen Foto-Chopper (VR7) ersetzt, denn bei einer mit 50 Hz vor sich hin brummenden Mechanik kann man sicher sein, dass sie wegen Kontaktproblemen in wenigen Jahren ausfällt. Der AC-Verstärker hat drei Röhrenstufen mit Gegenkopplung von der Kathode der Ausgangsstufe zur Kathode der Eingangsstufe. Der Ausgang des AC-Verstärkers ist per Trafo an den Demodulator gekoppelt. Der Demodulator wandelt das AC-Signal in ein DC-Signal, das recht proportional zur Eingangsspannung des pH-Meters ist (siehe **Bild 6**).

Einen interessanten Aspekt gibt es noch beim Gleichrichter V1, der ersten und in Bild 4 nicht gezeigten Röhre: Ihre Heizspannung ist niedriger als 6,3 V, da mit R48 ein 3- Ω -Widerstand in Serie liegt. Der Grund für die niedrigere Spannung ist, dass sich die Eingangsimpedanz erhöht, wenn die Kathode weniger heiß ist. Negativ daran ist, dass diese Maßnahme durch vorzeitige Veränderungen der Kathode die Lebensdauer der Röhre verkürzt. Als ich die Rückseite des Geräts öffnete, fiel mir gleich die eine weniger hell leuchtende Röhre auf. Zuerst dachte ich sogar, sie wäre defekt.

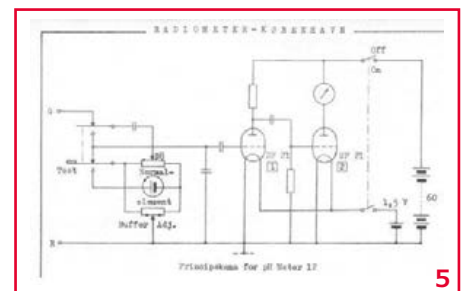
Die Betriebs-Gleichspannung des Verstärkers stammt von einem Vollweg-Röhrengleichrichter. Die Eingangsstufe wird mit den 85 V einer Spannungsstabilisator-Röhre betrieben.

Sauerstoffmonitor PHA928a

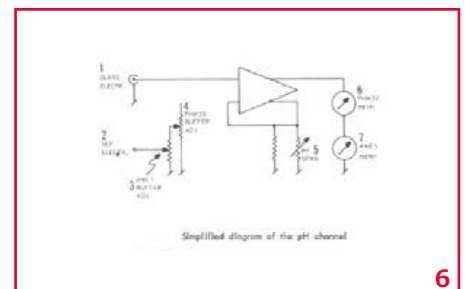
Der Sauerstoffmonitor ist passiv aufgebaut. Er enthält lediglich Feedback-Schaltungen, die



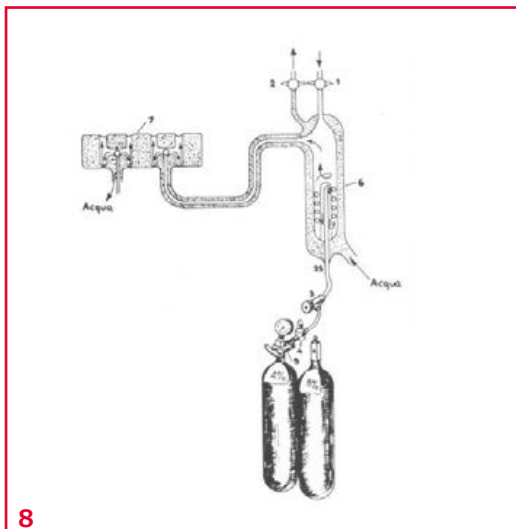
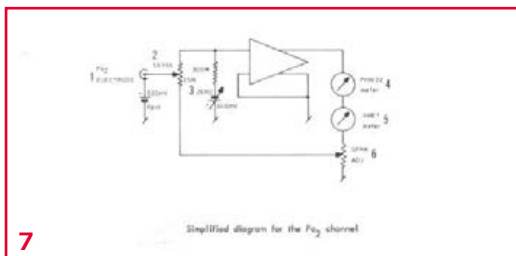
4



5



6



EST^D 2004

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

zwischen Verstärker und Meter des PHM22 sowie dem externen Messgerät eingeschleift werden.

Die pO_2 -Elektrode wird mit einer Polarisationsspannung von 630 mV betrieben, die aus einer Quecksilberzelle mit 1,35 V stammt. Ihre Beschaltung ist niedrigohmig ausgelegt, bei geringen Strömen. **Bild 7** zeigt die vereinfachte Schaltung des pO_2 -Kanals. Die Bilder 6 und 7 zeigen, wie ein einzelnes pH-Meter für zwei verschiedene Messungen verwendet werden kann.

Kohlendioxidwerte (pCO_2)

Eine pCO_2 -Elektrode gibt es hier nicht. CO_2 -Werte im Blut werden indirekt durch drei Blutproben gemessen. Die erste Messung bestimmt den pH-Wert. Die beiden weiteren Proben kommen in getrennten Kammern des Tonometers, wo sie noch mit 4 % bzw. 8 % CO_2 -Gas beaufschlagt werden (siehe **Bild 8**). Nach etwa vier Minuten Ausgleichszeit wird der pH-Wert beider Proben gemessen. Nun kann pCO_2 der ersten Probe nach dem Verfahren von Astrup (et al.) durch die Anwendung von Siggaard-Andersons Nomogramm der Korrelation von arteriellem pH-Wert und pCO_2 bestimmt werden [3]. Mit diesem Nomogramm kann man noch andere Dinge über den menschlichen Körper herausfinden, doch

das dürfte weit über das Ziel dieses Artikels hinausführen.

Mikroelektroden

Der wichtigste Grund ungenauer Messungen ist die Drift. Die Elektroden sind empfindlich für Temperatureffekte und statische Aufladung, da sie eine hohe Impedanz aufweisen. Die Blutproben hinterlassen Substanzen am Glas und beeinträchtigen zu einem gewissen Grad auch die Polypropylen-Membranen der Elektroden. Auch die Elektronik driftet. Daher sind alle Elektroden mit Glas ummantelt und von Wasser umflossen, dessen Temperatur per Thermostat geregelt wird. Das Wasser ist leicht salzhaltig und damit leitend, was eine Art flüssige Abschirmung der Elektroden ergibt – ganz wie Metall bei empfindlichen Verstärkern (siehe **Bild 9**).

Da sich die Empfindlichkeit der Elektroden mit der Zeit verändert, muss immer wieder kalibriert werden. Hierzu dienen Referenzlösungen für den pH-Wert sowie freie und gesättigte Lösungen für pO_2 .

50 Jahre später

Heute haben die Laboranten in der Hämatologie eine ganze Batterie an Referenzen zur Verfügung. Zudem gibt es eine intensive Qualitätskontrolle. In den 1950er und 1960er Jahren, als die Radiometer-Kombination aus PHM22/PHA928a eingesetzt wurde, gab es dies alles noch nicht. Aus einer Klinik in Helsinki wird folgende Geschichte einer Qualitätskontrolle auf Retronik-Art erzählt: Wenn die Apparatur aus PHM22/PHA928a zu viele fragwürdige Resultate lieferte, blätterte ein Labor-Mitarbeiter durch die neue Rolodex, telefonierte nach einer bestimmten jungen Reinigungskraft und wartete, bis sie erschien. Diese junge Frau war stämmig und praktisch immer gesund. Eine Krankenschwester entnahm dieser „Referenzperson“ etwas Blut, das dann analysiert wurde. Wenn pH-, pO_2 - und pCO_2 -Werte plausibel waren, wurde die Blutanalysetechnik für gut befunden. Gut genug zumindest für die meisten Patienten.

(130132)

Weblinks

1. *Der Fluch des Sammelns*, Elektor Mai 2013. www.elektor.de/120753
2. www.elektor.de/130132
3. www.anaesthesia.med.usyd.edu.au/resources/lectures/acidbase_mjb/description.html

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



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

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

UMWELTSCHONEND – GÜNSTIG – GREEN

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



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

Hexadoku Sudoku für Elektroniker

Es wird jetzt draußen wieder kühler und der Herbst kündigt sich an. So ein schon etwas dunklerer Abend eignet sich natürlich vorzüglich, um es sich mit einem Hexadoku gemütlich zu machen. Und wenn man die Lösung herausgefunden hat, kann man sogar noch etwas gewinnen. Wir wünschen viel Rätselspaß!

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 **Eurocircuits/Elektor-PCB-Service-Gutschein im Wert von 100 €** und drei **Elektor-Bücher-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 2013!

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

Die richtige Lösung ist: **3B4CD**.

Der Eurocircuits/Elektor-PCB-Service-Gutschein im Wert von 100 € geht an: József Nagy aus Baracs (Ungarn).

Einen Elektor-Gutschein über je 50 € haben gewonnen: Mary Chang, Jacqueline Deletombe und Olavi Parkka.

Herzlichen Glückwunsch!

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

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

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

Von den Machern von Elektor!

elektor

5 | 2014

[D] 17,50 €

[A] 20,00 €

[L] 20,00 €

[B] 20,00 €

CHF 30,00

LEDs 5

Special Project

Leuchtdioden in
Theorie, Anwendung
und Selbstbauprojekten

- LED-Kommunikator
- RSS-Nachrichten-Ticker
- Beleuchtungs-Steuerung
- RGB-LED-Effekte mit NXP-ARM
- Schaltungen, Tipps und Kniffe
- Treiberströme: Höher als erlaubt
- Knowhow: Matrix-Schaltungen
- Dimmungstechniken im Vergleich
- Ein IC treibt LEDs, regelt Solarzellen und lädt Batterien
- Ansteuerung von LEDs mit gepulstem Überstrom



**Jetzt neu
am Kiosk!**

Oder frei Haus unter www.elektor.de/leds5 bestellen!



Grundlagen für Visual Basic-Einsteiger und VB6-Umsteiger

1 VB Express 2010

Dieses Buch unterstützt den Anwender bei den ersten Schritten mit Visual Basic, in dem es sich auf die Werkzeuge der Toolbox und deren Eigenschaften konzentriert, die zum Schreiben praktisch verwertbarer Programme notwendig sind. Zu jedem Thema findet der Leser ausführlich kommentierte Beispielprogramme, die er selbst ausprobieren kann und die sich auf das Mindeste beschränken, was zum Starten der Software notwendig ist.

284 Seiten (kart.) • ISBN 978-3-89576-269-7
€ 34,80 • CHF 43,20

LCD-Graphik II, verkettete Strukturen II und die Fädelsprache LAX

2 AVR-Programmierung 4

In diesem neuen vierten Band der erfolgreichen Buchreihe zur Programmierung von AVR-Mikrocontrollern wird die LCD-Graphik aus Band 3 weiterentwickelt. Hinzu kommen das Füllen von Polygonen, die Zuordnung von Pixelkoordinaten zu Graphikobjekten und

die Verwendung des Displays als Textfenster. Aufbauend auf der Darstellung der inneren Mechanik von Fädelsprachen im vorigen Band wird außerdem die Fädelsprache LAX vorgestellt und implementiert.

320 Seiten (kart.) • ISBN 978-3-89576-232-1
€ 46,00 • CHF 57,10

Schaltungsprojekte für Profis

3 Arduino

Für den großen Erfolg der Arduino-Plattform lassen sich zwei Ursachen finden. Zum einen wird durch das fertige Board der Einstieg in die Hardware enorm erleichtert; der zweite Erfolgsfaktor ist die kostenlos verfügbare Programmieroberfläche. Unterstützt wird der Arduino-Anwender durch eine Fülle von Software-Bibliotheken. Die täglich wachsende Flut von Libraries stellt den Einsteiger vor erste Probleme. Nach einfachen Einführungsbeispielen ist der weitere Weg nicht mehr klar erkennbar, weil oft detaillierte Projektbeschreibungen fehlen. Hier setzt dieses Buch an. Systematisch werden Projekte vorgestellt, die in verschiedene Themengebiete einführen. Dabei wird neben den erforderlichen theoretischen Grundlagen

stets größter Wert auf eine praxisorientierte Ausrichtung gelegt.

270 Seiten (kart.) • ISBN 978-3-89576-257-4
€ 39,80 • CHF 49,40

Apps programmieren – Schritt für Schritt

4 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



Bestücke und getestete Platine

5 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

30 Projekte in C für Fortgeschrittene

6 ARM-Mikrocontroller 2

Die im Buch beschriebenen Projekte mit dem mbed-Board sind für Einsteiger in C und ARM-Mikrocontroller ausgelegt. Der mbed NXP LPC1768 nutzt Cloud-Technologie, ein revolutionäres Konzept in der Software-Entwicklung.

Es bedeutet, dass man keinerlei Software auf seinem PC installieren muss, um den mbed zu programmieren. Das Einzige, was Sie brauchen, ist ein

Webbrowser mit Internetzugang und einen freien USB-Anschluss an Ihrem PC.

243 Seiten (kart.) • ISBN 978-3-89576-271-0
€ 39,80 • CHF 49,40

Kompletter Elektor-Jahrgang 2012 auf DVD

7 Elektor-DVD 2012

Die neue Elektor-Jahrgangs-DVD enthält alle Artikel des Jahrgangs 2012. Sie verfügt über eine sehr übersichtlich gestaltete Benutzeroberfläche. Mit der Elektor-DVD 2012 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 978-90-5381-273-0
€ 27,50 • CHF 34,10

Taschenbuch für Elektronik und Elektrotechnik

8 Formelsammlung

Diese „Formelsammlung“ beinhaltet alle wichtigen Details für Ingenieure, Techniker, Meister und Fach-

arbeiter in der Elektrotechnik und Elektronik, die in Forschung, Entwicklung und Service tätig sind. Die logische Gliederung in zehn Kapitel vereinfacht das Nachschlagen und Aufsuchen der gewünschten Themen. In den einzelnen Kapiteln finden Sie immer die notwendigen mathematischen und physikalischen Formeln sowie die wichtigsten Tabellen.

271 Seiten (kart.) • ISBN 978-3-89576-251-2
€ 29,80 • CHF 37,00

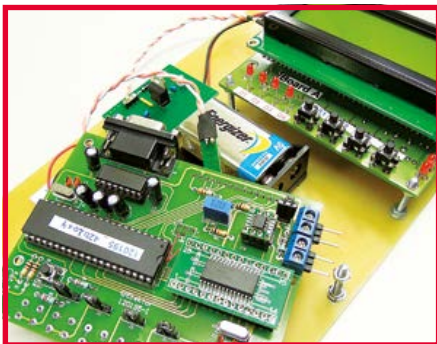
Weitere Informationen zu unseren Produkten sowie das gesamte Verlagssortiment finden Sie auf der Elektor-Website:

www.elektor.de/shop

Elektor-Verlag GmbH
Süsterfeldstr. 25
52072 Aachen

Tel. +49 (0)241 88 909-0
Fax +49 (0)241 88 909-77
E-Mail: bestellung@elektor.de

●Nächsten Monat in Elektor



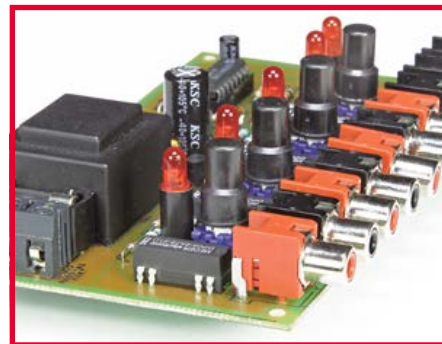
CAN-Tester

Moderne Motorfahrzeuge sind rollende Netzwerke. Für den Datentransport der an Bord installierten Steuerungen und Regelungen hat sich inzwischen der CAN-Bus (Controller Area Network) etabliert, ein Bus-System, das für den Betrieb in störbehafteter Umgebung ausgelegt ist. Da der CAN-Bus recht komplex ist, lassen sich Fehler nicht immer leicht aufspüren. Hier hilft unser CAN-Tester weiter, er gibt genauen Aufschluss über das Geschehen auf diesem Bus.



Mehrkanal-Temperaturlogger

Möchten Sie die Messwerte mehrerer digitaler Temperatursensoren DS18B20 gleichzeitig erfassen und aufzeichnen? Mit dem Sechskanal-Temperaturlogger und der eingebauten Echtzeituhr ist das kein Problem. Der Temperaturlogger wird über ein Tastertfeld und ein vierzeiliges alphanumerisches Display bedient. Speichermedium ist eine herkömmliche SD-Speicherkarte, so dass der Auswertung der Messdaten auf dem PC nichts im Weg steht.



Audio-Umschalter

Dies ist ein zuverlässiger und unkomplizierter Umschalter für Audio-Signale, der die installierte Anlage ergänzt oder den defekten Eingangswahlschalter einer sonst noch intakten Audio-Anlage ersetzt. Der Audio-Umschalter wählt eine von bis zu vier Eingangssignalquellen aus; ferner lassen sich zwei Signalquellen zum Ausgang durchschalten, ohne dass diese Signalquellen miteinander kurzgeschlossen werden. Alle Bauelemente sind bedrahtet, SMDs kommen hier nicht vor.

Änderungen vorbehalten! Elektor November 2013 erscheint am 23. Oktober 2013. Verkaufsstellen findet man unter: www.pressekaufen.de.

Rund um die Uhr und
sieben Tage die Woche

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

elektorlabs
Sharing Electronics Projects

Home News Proposals In Progress Finished

elektor
PCB Service
powered by Eurocircuits
www.elektorpcbservice.com

Proposals
Active Popular
EDITOR'S CHOICE
OTA-Overdrive with genuine Germanium-Sound (130311...)
5,135 views ★★★★★

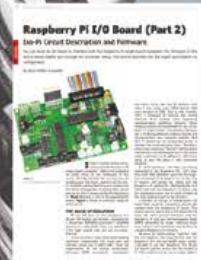
In Progress
Active Popular
Efficient Solenoid Valve (130258-1)
1,599 views ★★★★★

Finished
Active Popular
POST
Elektor.POST - No. 15 (The Hypnotizer)
535 views ★★★★★

About Elektor.LABS
Elektor Labs

Create a Project
Create a new project or enter a proposal
Get help, feedback & votes from other visitors, and maybe you will get Elektorized too!

Not a member?
You want to post a project but you are not a member? You can! Click here to send a description of your project including a circuit diagram and a



We're celebrating a new look — and the next 25 years.



25% OFF CIRCUIT CELLAR

Whether it's **programming advice** or **design applications**, you can rely on *Circuit Cellar* for solutions to all your electronics challenges. **Raspberry Pi**, **embedded Linux**, **low-power design**, **memory footprint reduction** and more! Become a member, and see how the hottest new technologies are put to the test.

JOIN TODAY!

www.circuitcellar.com/sepN13

- ✓ über 40 Jahre Erfahrung
- ✓ schneller 24-Std.-Versand
- ✓ über 45.000 Produkte am Lager
- ✓ kein Mindermengenaufschlag

JETZT ENTDECKEN!

„Eine Hi-tech-Himbeere, unendlich viele Möglichkeiten“

Raspberry Pi - eine kreditkartengroße Platine mit schier endlos vielen Einsatzmöglichkeiten entdecken Sie den Mini-PC auf unserer Raspberry-Themenseite.

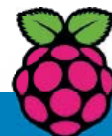
Raspberry-Pi Special-Themenseite



<http://rch.it/b9>

„Raspberry Pi“-Mini-PCs

Top-Erweiterungen für Ihr Projekt!



HD-Kamera für den Raspberry Pi

- 5 Megapixel • Foto: 2592 x 1944 Pixel
- Video: 1080p (Full HD), bis zu 30 Bilder/s
- Anschluss: über 15-poliges Folienbandkabel an die serielle MIPI-Kameraschnittstelle

29,95

RASPBerry PI CAM



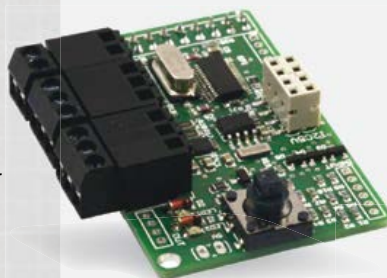
FULL-HD

Erweiterungsboard für den Raspberry Pi

RasPiComm ist eine winzige Eingangs-/Ausgang-Erweiterungsplatine

- serielle Kommunikationschnittstellen
- batteriegestützte Echtzeituhr
- 5-V-tolerante Digital-E/As
- Daumen-Joystick

RASPBerry COMM **64,95**



NEU IM PROGRAMM

USB zu TTL serieller UART-Wandler, 1 m

Das TTL-232R-RPi-Kabel serielle Datenübertragung und unterstützt Datenraten von 300 Bit/s bis 3 Mbit/s bei einem 3,3-V-TTL-Pegel.

- 5V sichere TTL-Eingänge

TTL 232R RPi **21,90**



MIXEN & WÄHLEN SIE IHRE EIGENE FARBKOMBINATION

„Raspberry Pi“-Gehäuse

Kein schrauben, kein kleben ...

- Alle Ausschnitte vorgeschritten
- Es sind keine Modifikationen erforderlich!
- Mit Gummifüßen
- Mit LED-Lichtleiter
- Hochglanz-Oberfläche

Wählen Sie ein Ober- und ein Unterteil aus unserer Auswahl von 10 Farben in unserem Online-Shop!



Beispiel Oberteil:
CB RPIOT RT rot **6,70**

Beispiel Unterteil:
CB RPIUT RT rot **5,95**

Embedded Pi Board

3-fach Plattform für:

- ☒ Raspberry Pi
- ☒ Arduino
- ☒ 32-Bit Arm Cortex

inkl. 4 Kunststoffabstandshalter und Flachbandkabel (26-/10-pol.)

RASPBerry IBOARD **36,90**



Jetzt bestellen! www.reichelt.de

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

reichelt elektronik
Ihr kompetenter Partner für

Bauelemente • Stromversorgung • Messtechnik • Werkstattbedarf
Haus- & Sicherheitstechnik • Netzwerk- & PC-Technik • TV-Technik