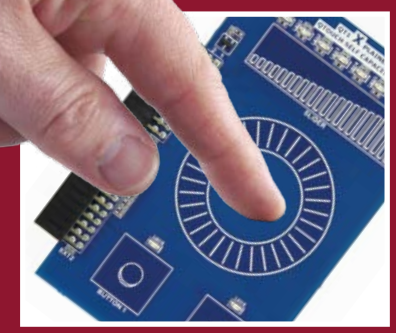
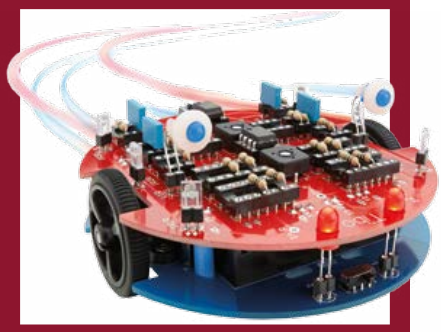
November 2015 | Nr. 539
www.elektormagazine.de

elektor

LEARN | DESIGN | SHARE



The Invisible Touch
ARM-Controller für Einsteiger (Teil 8)



Analoger Roboter
Von der Idee zum Bausatz

Drohnen im Anflug
Drei interessante Konstruktionen

DDS-Funktionsgenerator

Sinus, Rechteck und Dreieck bis 10 MHz

In dieser Ausgabe:
6 Projekte
1 Programmierkurs
1 CAD-Kurs
2 Web-Recherchen
und noch viel mehr!

T-Board mit ARM-Controller • **Seriell/USB-Konverter** • **Digitales Zoetrop** • **EAGLE Tipps & Tricks:** Beispielprogramme • **Android I/O-Board:** Datenlogger • **Retronik:** Schallband-Rekorder Tefifon • **Bemerkenswerte Bauteile:** Fluoreszenzanzeigen • **Camelback-Pegelindikator** • **Tipps und Tricks:** Von Lesern für Leser • **Mehr Tempo bei Embedded-Systemen**

G3078 E

(D) € 7,90 | CHF 14,80 | (A, B, L) € 8,90

4 198630 307906 11

DAS ORIGINAL SEIT 1994
PCB-POOL®
Beta LAYOUT

PCB-Prototypen & kleine Serien

Edelstahl SMD-Schablone
bei jeder Prototyp-Bestellung inklusive

PCB-POOL® ist eine eingetragene Marke der

www.pcb-pool.com

Beta LAYOUT
create:electronics

BECKER MÜLLER

WOW!
Schon da!

Zu Ihrem Vorteil aktualisiert!
Online-Kalkulator

30
JAHRE
LEITERPLATTEN
SPEZIALISTEN

- PUNKTGENAU
- PLANGENAU
- PREISGENAU

Becker & Müller
Schaltungsdruck GmbH
Tel.: +49 (0)7832 9180-0
www.becker-mueller.de

Schaeffer AG

SIE DESIGNEN – WIR FERTIGEN

Frontplatten in Profiqualität

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

www.schaeffer-ag.de

Impressum

46. Jahrgang, Nr. 539 November 2015
Erscheinungsweise: 10 x jährlich
(inkl. Doppelhefte Januar/Februar und Juli/August)

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

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

Hauptsitz des Verlags
Elektor International Media
Allee 1, NL-6141 AV Limbricht

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

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

Es gilt die Anzeigenpreisliste Nr. 45 ab 01.01.2015

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

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

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

Ihr Projekt

Auf Messen habe ich Gelegenheit, einige meiner Leser auch einmal persönlich kennenzulernen. Neben inhaltlichen Wünschen höre ich dann auch oft von meinem Gegenüber, dass in dessen Bastelkeller (neudeutsch auch Homelab genannt) gerade etwas Interessantes liegt, das aber leider noch nicht ganz fertig ist. „Wenn man doch nur mehr Zeit hätte…“
Dieses Problem kenne ich selbst nur zu gut. Aber ist es nicht schade, wenn eine tolle Idee verstaubt, weil noch eine attraktive Funktion fehlt oder nicht getestet ist? Auf unserer Website www.elektor-labs.com sind auch unfertige Projekte, bloße Skizzen und sogar Wünsche und Ideen willkommen. Nicht nur einmal haben wir eine solche Idee in unserem Labor zu einem fertigen Produkt weiterentwickelt. Geradezu das Metier meiner Kollegen im Labor ist das Finishen eines Entwurfs, wobei hier in erster Linie das Routen einer Platine zu nennen ist. Ein abgekündigtes Bauteil, ein Aufbau auf dem Steckbrett oder irgendetwas, das Ihnen noch unschön vorkommt, sollte Sie keinesfalls abhalten, uns Ihre Idee zu posten.
Natürlich heißen wir auch alle Schaltungen und Ideen willkommen, die uns per E-Mail erreichen. Mit einer Veröffentlichung auf unserer Projekt-Website haben Sie allerdings die Chance, dass sich direkt gleichgesinnte Leser melden, die eventuell sogar an einem ähnlichen Entwurf arbeiten oder gearbeitet haben. Mögliche Probleme kann man dann gemeinsam angehen.
Wir haben übrigens schon Pläne in der Schublade für weitere Funktionen von elektor-labs.com, mit denen eine Zusammenarbeit unserer Leser noch einfacher wird.
Bleiben Sie dran!

Jens Nickel
Chefredakteur Elektor

Unser Team

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

Inhalt

46. Jahrgang – Nr. 539 November 2015

3 **Impressum**

6 **Das Elektor-Netzwerk**

22 **News**

24 **Business Report**
Mehr Tempo bei Embedded-Systemen mit USB 3.0

71 **Elektor World News**

82 **Hexadoku**
Sudoku für Elektroniker

LEARN DESIGN SHARE

8 **Willkommen bei LEARN**

9 **Tipps und Tricks**
Von Lesern für Leser

10 **Von 8 auf 32 bit: ARM-Controller für Einsteiger (8)**
The invisible touch!

19 **Bemerkenswerte Bauteile**
Fluoreszenzanzeigen

20 **EAGLE Tipps & Tricks (3)**
Beispielprogramme

LEARN DESIGN SHARE

28 **Willkommen bei DESIGN**

29 **Seriell nach USB**
Mit dem FT232H

32 **Android I/O-Board (3)**
Konzept und Funktionen des Datenloggers

39 **Digitales Zoetrop**
Uraltes Prinzip, neue Technik

42 **DDS-Funktionsgenerator**
Sinus, Rechteck und Dreieck bis 10 MHz

50 **Analoge Roboter**
Von der Idee zum Bausatz

54 **T-Board mit ARM**
32-bit-Power mit Cortex-M0+

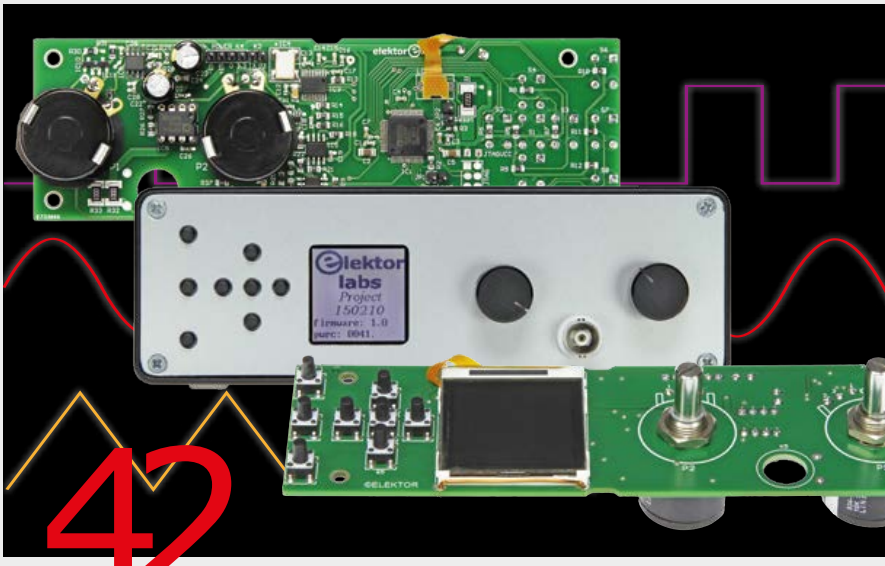
62 **Camelback-Pegelindikator**
Immer Wasser im Tank!

29

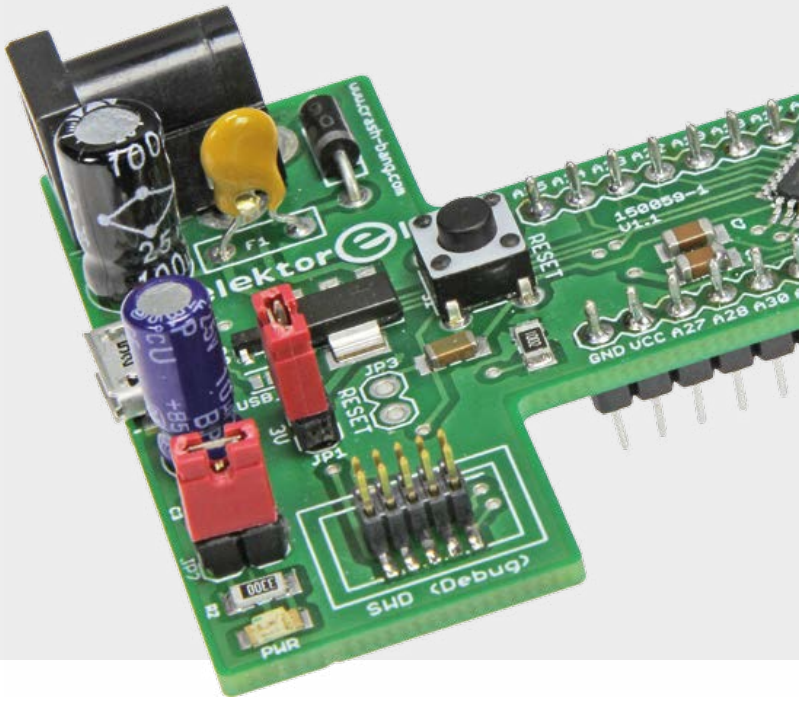
Seriell nach USB

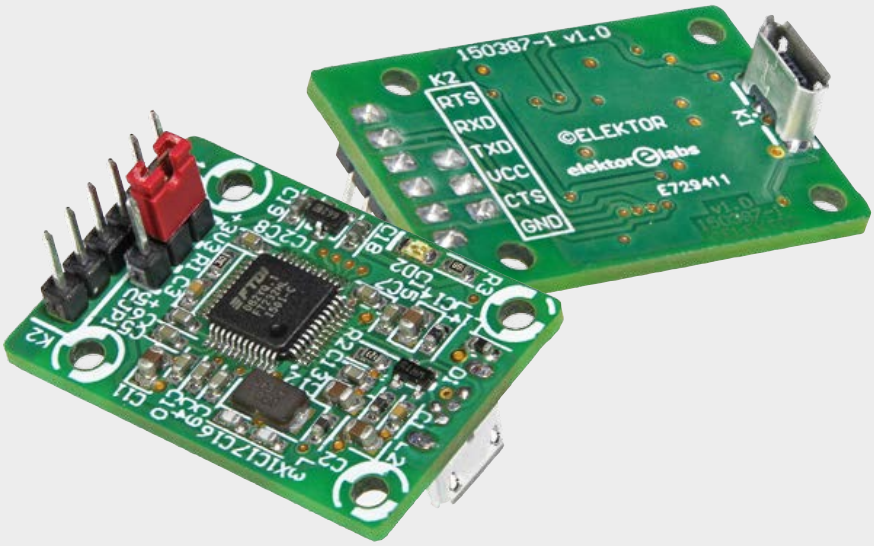
Mit dem FT232H

Der altgewohnte serielle Port ist noch immer eine viel genutzte Schnittstelle zwischen PC und Peripherie. Hier stellen wir eine kompakte USB/Seriell-Konverter-Implementierung mit dem FT232H von FTDI vor, die kompatibel mit USB 2.0 (high-speed) ist!



42

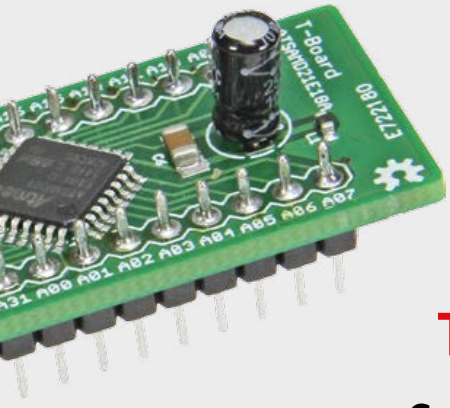




DDS-Funktionsgenerator

Sinus, Rechteck und Dreieck bis 10 MHz

DDS-Chips sind leicht erhältlich, so dass der Analogteil eines Weitbereichs-Funktionsgenerators viel einfacher ausfällt als früher. Man muss lediglich ein IC auswählen, eine geeignete Ausgangsschaltung und einen Mikrocontroller hinzufügen, sich eine Benutzeroberfläche ausdenken und ein wenig programmieren. Oder doch nicht?



54

T-Board mit ARM

Cortex-M0+ 32-bit-Power

Die vor einem guten Jahr vorgestellte Familie der T-Boards, mittlerweile eine echte Institution, erhält Zuwachs: Wir stellen das jüngste (und stärkste) Mitglied der Familie vor – erkunden Sie die Welt der 32-bit-ARM-Controller mit dem ARMierten T-Board!

elektor magazine

LEARN | DESIGN | SHARE

70 Willkommen bei SHARE

72 Aus dem Labor
Bockige Buck-Converter

74 Retronik
Schallband-Rekorder Tefifon

78 Web-Recherche (1)
Mit Kickstarter & Co.

80 Web-Recherche (2)
Drohnen im Anflug!

Den im letzten Heft angekündigten Artikel „Windows auf dem Raspberry Pi“ haben wir auf die nächste Ausgabe verschoben.

Speziell für unsere Gold-Mitglieder: Mit diesem Heft erhalten Sie die neueste Ausgabe von ElektorBusiness mit dem Thema Elektronik-Produktion, die wir anlässlich der Messe **productronica** erstellt haben. Green-Mitglieder können die Ausgabe herunterladen unter www.elektormagazine.de/articles.

Vorschau

Aktive Frequenzweiche

Diese aktive Frequenzweiche ist universell einsetzbar, beispielsweise in aktiven Lautsprecher-Systemen.

Kühlschrank-Thermostat mit Android I/O-Board

Das Ersetzen eines Kühlschrank-Thermostats dürfte für einen Elektor-Leser keine große Hürde darstellen, doch man muss erst mal einen neuen, passenden finden. Eine schöne Anwendung für unser Android I/O-Board: Hiermit lässt sich ein universeller Thermostat realisieren, der natürlich noch eine Menge Komfort-Funktionen bietet.

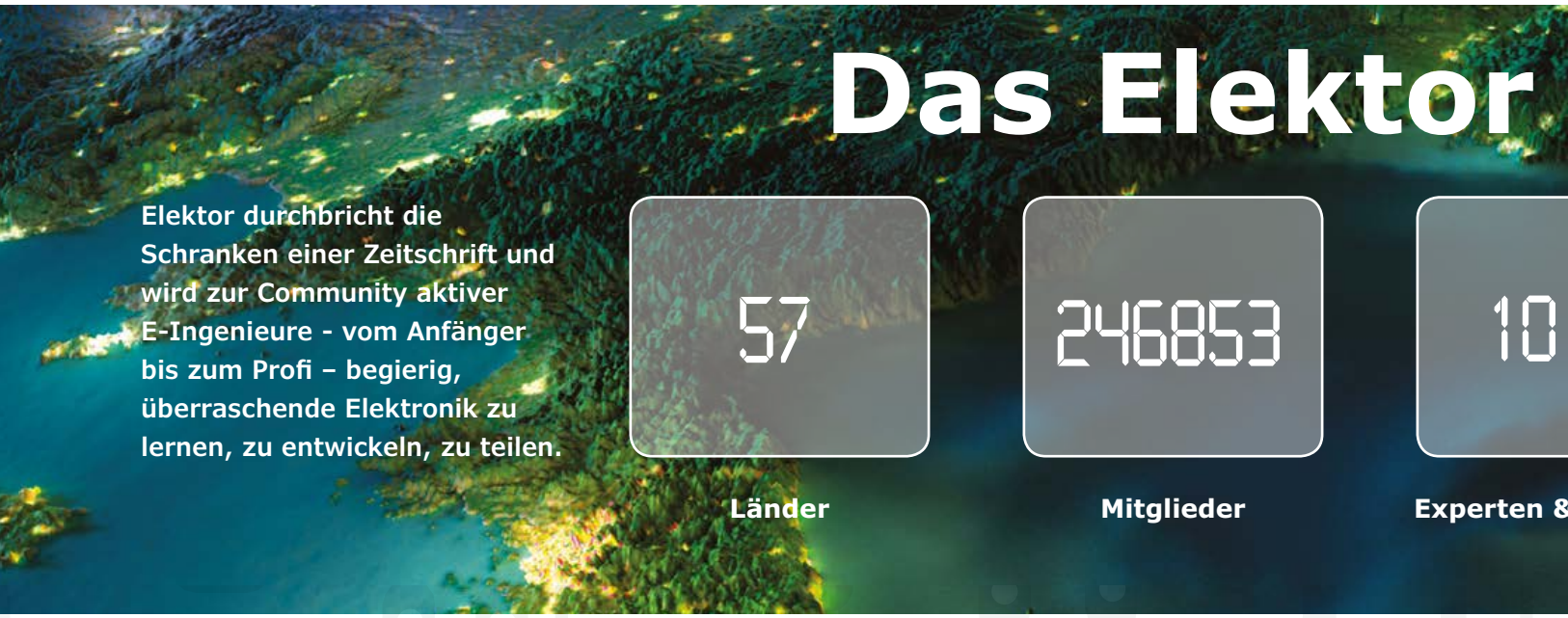
Weitere Themen im nächsten Heft:

- Lampensteuerung mit Bluetooth Low Energy
- LED-Ball
- Lader für Knopfzellen
- Und vieles mehr!

Änderungen vorbehalten.

Elektor Dezember 2015 erscheint am 19. November 2015.

Verkaufsstellen findet man unter www.pressekaufen.de.



Das Elektor

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

57 Länder

246853 Mitglieder

10 Experten &

Elektor.POST

Die wöchentliche E-Inspiration

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

www.elektor.de/newsletter

Elektor – das Magazin

Fast 1024 Seiten überraschende Elektronik im Jahr

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

www.elektormagazine.de

Elektor•TV

Auch auf der Mattscheibe

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

www.elektor.tv

Elektor-Platinenservice

Boards für Sie...

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

www.elektorpcbservice.com

Das Elektor-Labor

Lernen, entwickeln, teilen!

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

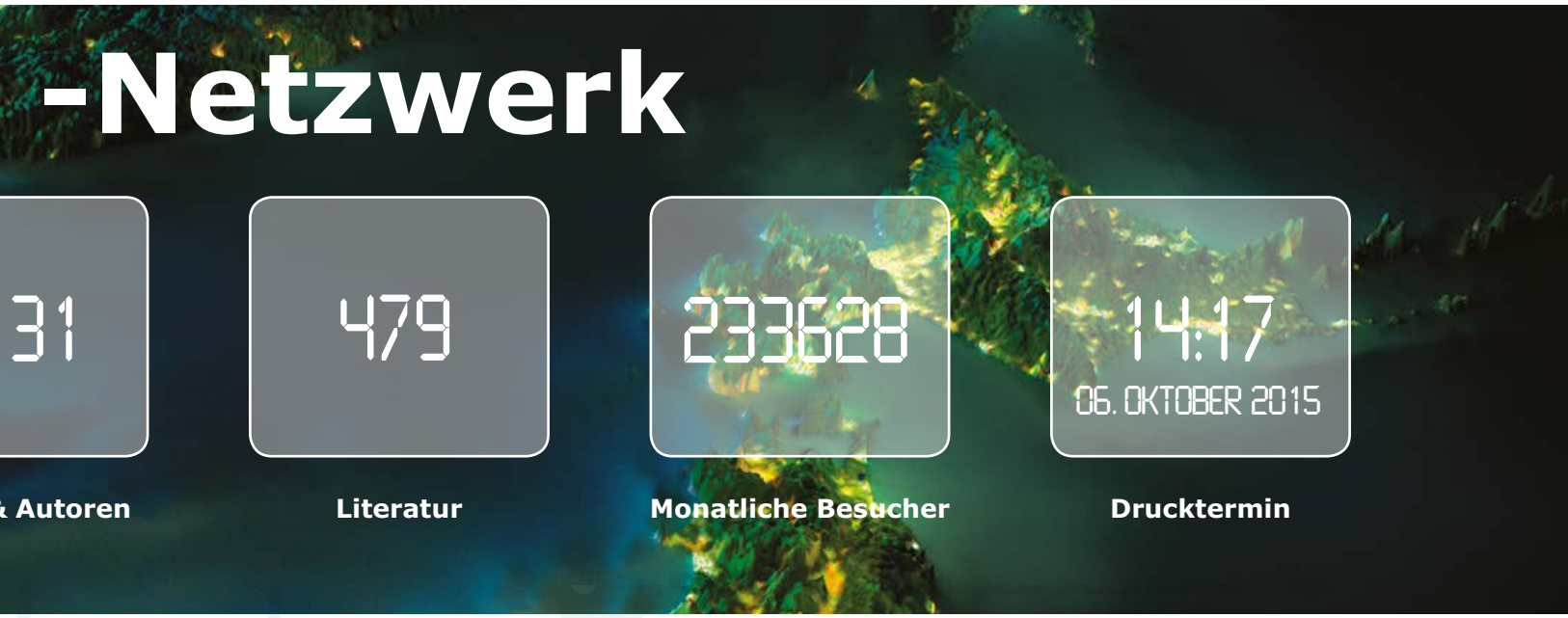
www.elektor-labs.com

Elektor Academy

Reiten Sie auf der Lernkurve!

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

www.elektor.de/events



Elektor-Community

Werden Sie GREEN oder GOLD-Mitglied!

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

www.elektor.de/mitgliedschaft

Elektor E-Shop

Füllen Sie Ihren Warenkorb!

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

www.elektor.de

Elektor-Bücher & DVDs

Powerpacks mit E-Informationen

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

www.elektor.de

Treten Sie dem weltweit größten Elektroniker-Netzwerk bei!

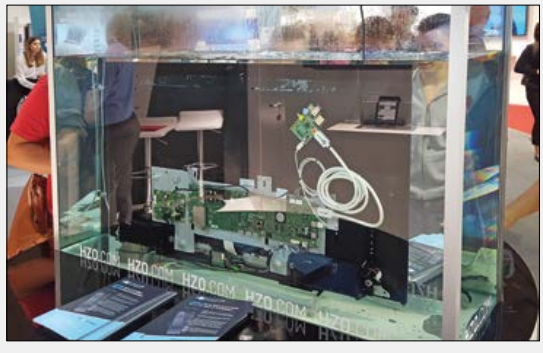
GOLD 127,50 €/Jahr	GREEN 92,50 €/Jahr Empfehlung	GRATIS
✔ 10x Elektor jährlich (Print)	✖ 10x Elektor jährlich (Print)	✖ 10x Elektor jährlich (Print)
✔ 10x Elektor jährlich (Digital)	✔ 10x Elektor jährlich (Digital)	✖ 10x Elektor jährlich (Digital)
✔ Zugang zum Elektor-Archiv	✔ Zugang zum Elektor-Archiv	✖ Zugang zum Elektor-Archiv
✔ 10% Rabatt auf Shop-Produkte	✔ 10% Rabatt auf Shop-Produkte	✖ 10% Rabatt auf Shop-Produkte
✔ Zugang zu Elektor-Labs	✔ Zugang zu Elektor-Labs	✖ Zugang zu Elektor-Labs
✔ Elektor-Jahrgangs-DVD	✔ Elektor-Jahrgangs-DVD	✖ Elektor-Jahrgangs-DVD
✔ Exklusive Top-Angebote	✔ Exklusive Top-Angebote	✔ Exklusive Top-Angebote
✔ 26 neue Projekte jährlich	✔ 26 neue Projekte jährlich	✔ 26 neue Projekte jährlich
www.elektor.de/gold-mitglied	www.elektor.de/green-mitglied	www.elektor.de/newsletter

Willkommen bei **LEARN**

Von **Jens Nickel**

Die Internationale Funkausstellung IFA wurde ja bereits von vielen anderen Zeitschriften im wahrsten Sinne des Wortes erschöpfend abgehandelt. Von meinem letzten Besuch dieser Messe – das ist zugegebenermaßen schon einige Jahre her – hatte ich riesige Hallen mit riesigen Fernsehern in Erinnerung; davor sehr viele sehr junge Frauen und Männer, deren technischer Sachverstand nicht allzu weit in die Tiefe reichte. Rein redaktionell würde es wohl auch in 2015 nicht allzu viel zu holen geben – so dachte ich vor meiner Fahrt nach Berlin.
Wieder was gelernt: Zwar gab es die riesigen Hallen immer noch; angereichert worden war das Ganze mit (selbstverständlich intelligenten) Kaffeemaschinen, Staubsaugern und Barbecue-Grills. In der unscheinbaren Halle 11.1 herrschte dagegen eine andere Atmosphäre. Ein Tag reichte nicht, um sich all die interessanten Elektronik-Neuheiten anzusehen, die von Klein- und Kleinstunternehmen entwickelt worden waren. Berlin war eben wie immer eine Reise wert!

Elektronik ganz wasserdicht zu bekommen, das schafft das Start-Up HZO mit einem hauchdünnen Kunststoffüberzug. Am Stand bewies das nicht nur der Raspi im Aquarium, sondern auch ein Smartphone, das in ein Wasserbecken getaucht wurde. Alles bleibt weiter bedienbar: Ein iPhone als Unterwasserkamera einzusetzen, ist problemlos möglich. Der Trick: Die Elektronik wird aus dem Gehäuse genommen, dann beschichtet und am Ende wieder eingebaut. So ein Service ist (leider) zu teuer, um echte Marktchancen zu haben. Gesucht werden stattdessen Partnerfirmen, die ihre Elektronik ab Werk in Massen beschichten lassen (www.hzo.com).



„Panono" heißt der rundum mit 36 Kameras bestückte Ball, über den schon im wöchentlichen Elektor-Newsletter zu lesen war. Jetzt steht das Berliner Start-Up vor dem Sprung in den Massenmarkt. Die erzeugten Panoramabilder mit 108 Megapixeln können auf dem PC, auf dem Tablet oder mit einer VR-Brille betrachtet werden (www.panono.com).

Die klobigen Brillen gab es überall auf der IFA zu sehen. Das Fraunhofer IIS bezog sogar einen Kopfhörer mit in das Rundum-Erlebnis ein (www.fraunhofer-cingo.com/de.html). Die (frei im Raum platzierbaren) Tonquellen behalten wie angenagelt ihren Platz im Bild, auch wenn man den Kopf bewegt. Als Demo wurde ein virtuelles Kino vorgeführt, auf dem ein realer Film lief. Die Boxen dieses gar nicht vorhandenen Kinosaals sorgten für echten Surround-Sound. Geht es noch cooler für einen Heimkino-Fan wie mich?

Das Tablet als „Second Screen" kennt man schon, etwa bei einer Fußball-Übertragung. Doch die Ultra-HD-Zoom-App des Fraunhofer HHI kann sogar automatisch einem der Spieler folgen (in diesem Fall Bastian Schweinsteiger). Beim Bier in der „dritten Halbzeit" ist man nun tatsächlich *der* Experte, wie beweglich (oder lauffaul) der Lieblings-Akteur auf dem Platz war.

Edukativ

Am gemeinsamen IFA-Stand des VDE, ZVEI und ZVEH wurde etwas für den Elektronik-Nachwuchs getan: Ein automatisiertes Modellhaus gehörte ebenso zum Konzept wie ein Löt-Workshop und eine Vorführung von Lego-Robotern (www.zvei.de).

(150475)

Tipps und Tricks

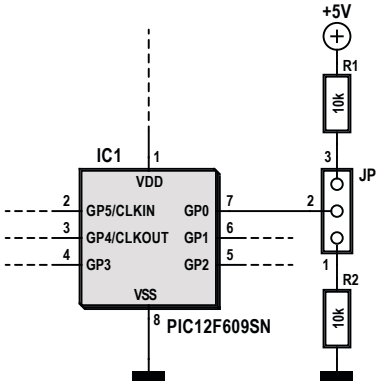
Von Lesern für Leser

Hier kommt wieder eine clevere Lösung, die das Elektronikerleben leichter macht.

Wieder mal zu wenig Portleitungen?

Von Dr. Martin Oppermann

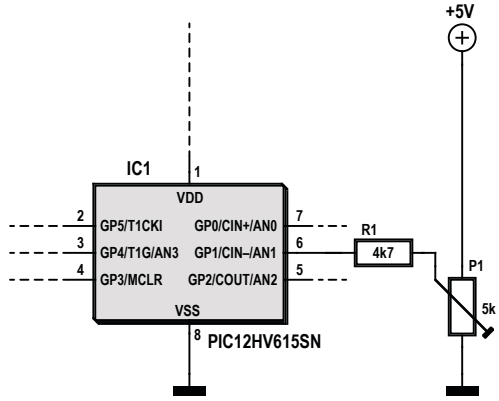
Um eine Mikroprozessorschaltung zu konfigurieren, werden häufig Jumper eingesetzt. Wenn nur zwei mögliche Zustände eingestellt werden müssen, so ist das einfach: Ein (geschlossener oder offener) Jumper an einem Portpin bestimmt den Wert. Müssen aber drei Zustände unterschieden werden, würde dies einen weiteren Portpin kosten. Ein Applikationsingenieur von Microchip hat mir dazu vor vielen Jahren einen Tipp gegeben, der auf jeden Fall bei PICs funktioniert; die drei Einstellungen werden dabei durch die Zustände „High“, „Low“ oder „offen“ eines einzigen Portpins definiert (Bild). Um den Pin abzufragen, geht man folgendermaßen vor:



- *Setze den Portpin auf Ausgang.*
- *Schreibe eine „1“ auf den Portpin.*
- *Setze den Portpin auf Eingang.*
- *Lese den Portpin aus und speichere den Wert.*
- *Setze den Portpin auf Ausgang.*
- *Schreibe eine „0“ auf den Portpin.*
- *Setze den Portpin auf Eingang.*
- *Lese den Portpin aus und addiere den Wert zum vorher gespeicherten Wert.*

Ist das Ergebnis nach der Addition „0“, dann liegt der Portpin auf Low (0 + 0). Ist das Ergebnis nach der Addition „2“ (1 + 1), dann liegt der Portpin auf High. Ist aber das Ergebnis der Addition „1“ („1“ + „0“), so ist der Portpin offen (floating), was dem dritten Zustand entspricht (um die Stromaufnahme nicht hochzutreiben, wird ein offener Portpin schnell wieder auf „Ausgang“ und einen definierten Pegel gesetzt).

Möglich machen das die immer vorhandenen parasitären Kapazitäten, der hohe Eingangswiderstand der Controllereingänge und die geringe Zahl der nötigen Befehle. Der Code ist abgearbeitet, ehe die parasitären Kapazitäten entladen sind.



Man kann die Methode sogar bei einem analogen Eingang einsetzen. Ich habe das bei einem Temperaturregler ausgenutzt (siehe Bild). Ist ein Poti am Analogport angeschlossen, so wird die Solltemperatur mit diesem Poti vorgegeben; ist der Portpin GP1 frei (bzw. P1 nicht bestückt), so gilt ein Festwert für die Solltemperatur.

Die Abfrage des Portpins GP1 erfolgt auf die gleiche Weise wie im vorhergehenden Beispiel. Ist ein Poti angeschlossen, so liegt irgendeine feste Spannung zwischen 0 V und 5 V am Portpin GP1 (je nach Stellung des Potis), der während der Abfrage als digitaler Eingang deklariert ist. Die Schmitt-Trigger-Eigenschaft der digitalen I/Os sorgt dafür, dass dabei immer ein „H“ oder ein „L“ eingelesen wird. Nur wenn der Eingang offen ist, bekommt man das Ergebnis „1“ nach der Addition der beiden zurückgelesenen Werte. ⏮

(150479)

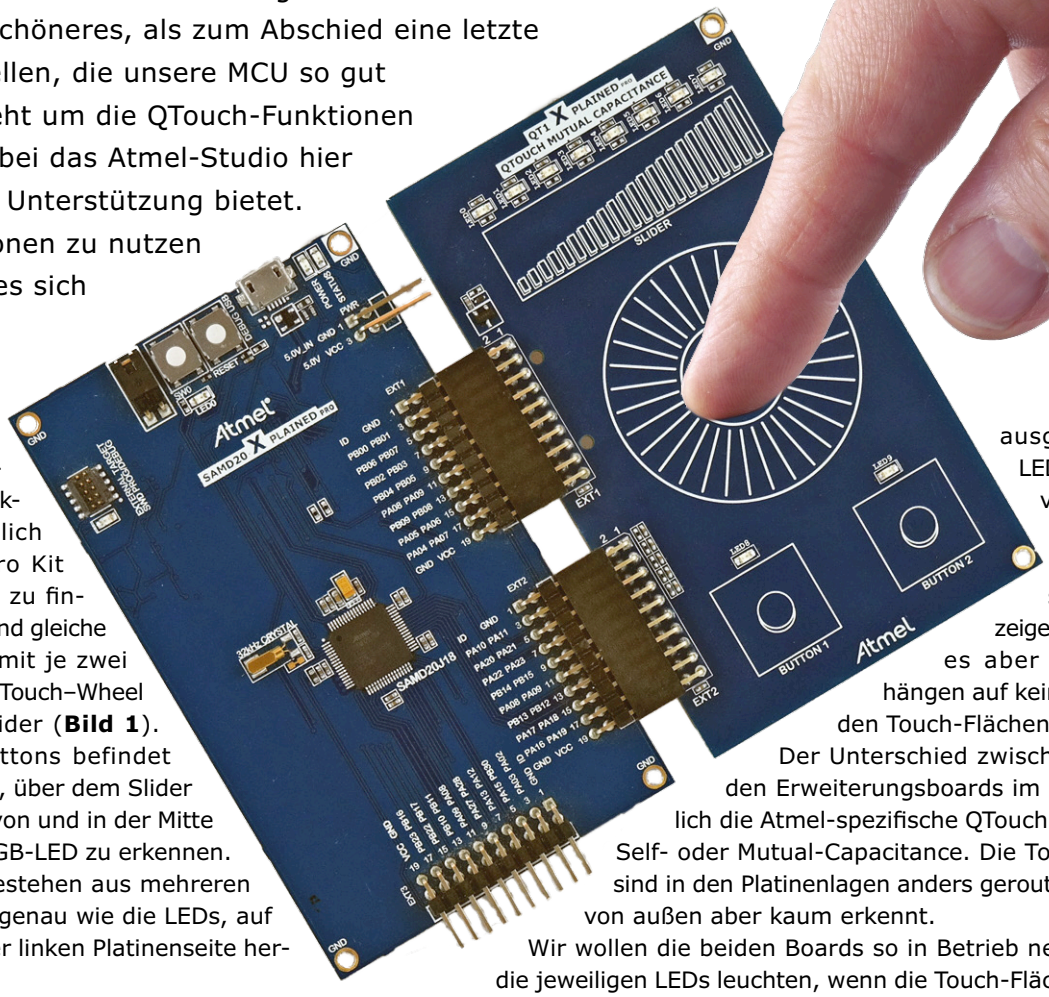
Sie haben selbst eine clevere Lösung für etwas wirklich Fummeliges? Wenden ein Bauteil oder Werkzeug auf ungewöhnliche Weise an? Haben eine Idee, wie man ein Problem einfacher oder besser angehen könnte, als das bisher gelöst wurde? Schreiben Sie uns – für jeden Tipp, den wir veröffentlichen, loben wir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

The invisible touch!

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

Von **Viacheslav Gromov** (D)

Endspurt, der letzte Teil des ARM-Programmierkurses steht an! Was gibt es Schöneres, als zum Abschied eine letzte Funktion vorzustellen, die unsere MCU so gut beherrscht? Es geht um die QTouch-Funktionen des SAM D20, wobei das Atmel-Studio hier eine tolle visuelle Unterstützung bietet. Die Touch-Funktionen zu nutzen ist einfacher, als es sich anhört.



Für diesen Kursteil brauchen wir ausnahmsweise kein Steckbrett, sondern lediglich das QT1 Xplained Pro Kit von Atmel [8]. Darin zu finden sind zwei annähernd gleiche Erweiterungsboards mit je zwei Touch-Buttons, einem Touch–Wheel und einem Touch–Slider (**Bild 1**). Über den beiden Buttons befindet sich je eine gelbe LED, über dem Slider gibt es gleich acht davon und in der Mitte des Wheels ist eine RGB-LED zu erkennen. Alle Touch-Flächen bestehen aus mehreren Elektroden und sind, genau wie die LEDs, auf Buchsenleisten auf der linken Platinenseite herausgeführt. Die LEDs kann man verwenden, um den Touch-Zustand anzuzeigen, man muss es aber nicht – sie hängen auf keine Weise mit den Touch-Flächen zusammen. Der Unterschied zwischen den beiden Erweiterungsboards im Kit ist lediglich die Atmel-spezifische QTouch-Technologie Self- oder Mutual-Capacitance. Die Touch-Flächen sind in den Platinenlagen anders geroutet, was man von außen aber kaum erkennt.

Bild 1. Das QT1 Xplained Pro Kit mit den beiden Erweiterungsboards.

Wir wollen die beiden Boards so in Betrieb nehmen, dass die jeweiligen LEDs leuchten, wenn die Touch-Flächen betätigt werden. Beim „Drehen“ des Touch-Wheels soll die RGB-LED proportional zum Winkel ihre Farbe ändern. Dazu müssen wir zunächst die beiden kostenlosen Tools QTouch-Composer und QTouch-Library im Atmel Studio 6 herunterladen und installieren. Dann können wir Schritt für Schritt und mit viel visueller Unterstützung das Projekt mit dem Mutual-Capacitance-Board realisieren. Auf ein ähnliches Programm für das Self-Capacitance-Board wird ebenfalls eingegangen. Zuvor werden wir uns aber erst die Hardware und die Touch-Technologien ansehen.

Peripheral Touch Controller

Der Peripheral Touch Controller (PTC) ist oft bei den SAM D- und den AVR Xmega-Mikrocontrollern vorhanden. Er führt die Touch-Messungen nahezu unabhängig von der CPU durch. Wenn dieses Peripherieelement aktiviert ist, läuft es parallel und wertet die Touch-Flächen kontinuierlich aus. Die Unterschiede der beiden Technologien, Mutual-Capacitance und Self-Capacitance werden im **Kasten** erläutert. Generell arbeitet dieses Peri-

pherieelement zusammen mit der Software als eine *State-Machine*.

Der grundsätzliche Aufbau dieses nicht ganz so unkomplizierten Elements bei unserem SAM D20 ist im **Bild 2** für beide Modi dargestellt. Man erkennt unter anderem den *X Line Driver* und *Input Control*, die die Pin-Belegungen konfigurieren und die Multiplexer ansteuern. Der *X Line Driver* ist in der Self-Capacitance-Konfiguration nicht vorhanden, da es pro Touch-Elektrode nur einen Y-Kanal gibt. Danach folgen die eigentlichen Elemente zur Auswertung und zur Aufbereitung des Signals. Man erkennt einen digital verstellbaren Widerstand (Potentiometer), der vor allem den Rauschpegel abschwächen soll, und einen Ausgangsverstärker. Es gibt noch weit mehr Einstellungen und Funktionen des PTCs, die aber hier wegen des beschränkten Platzes nicht beschrieben werden können. Sie werden detailliert in der Dokumentation [1] und im QTouch-Project-Builder (siehe unten) erklärt.

Das Tolle am PTC ist aber, dass er die wichtigsten Einstellungen beim Start selbst vornimmt – er kalibriert sich in einigen Parametern selbst, was dem Benutzer wiederum viele Einstellungen erspart.

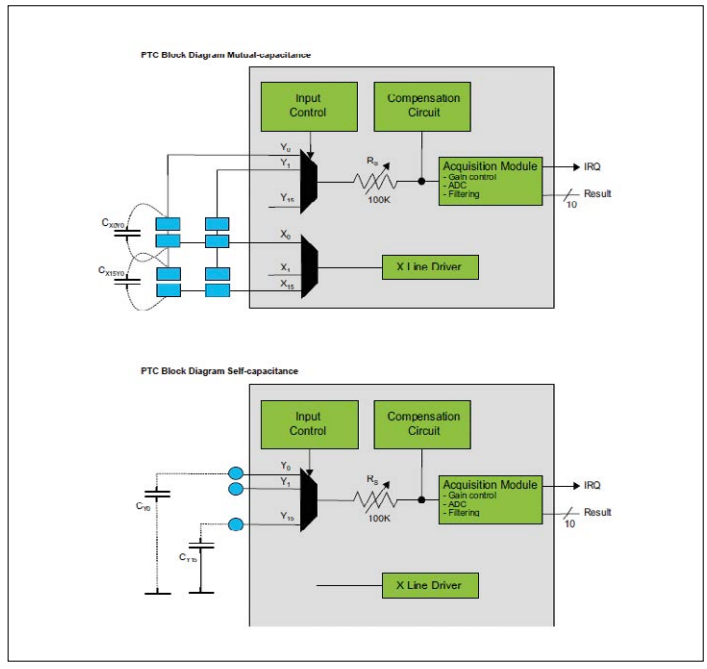


Bild 2. Prinzipieller Aufbau des Peripheral Touch Controllers in den Modi Self Capacitance und Mutual Capacitance (Blockschaltbilder und Screenshots: Atmel).

Installation der QTouch-Tools

Beschäftigen wir uns nun mit der Software. Zunächst sollen die beiden schon genannten Software-Tools im Atmel Studio 6 installiert werden. Dazu öffnet man das Studio und wählt unter „Tools“ den „Extension Manager…“, der Ihnen schon von den Updates des ASF bekannt sein dürfte. Man gibt, wie im **Bild 3** zu sehen, oben rechts „QTouch“ als Suchbegriff in das Suchfeld ein, um nach den beiden Tools zu suchen. Dabei werden Ihnen verschiedene Versionen angeboten. Installieren Sie bitte stets die aktuelle Version, zum Redaktionsschluss war es die Version 5.6. Bei neueren Versionen können die Installationsschritte von den hier beschriebenen abweichen.

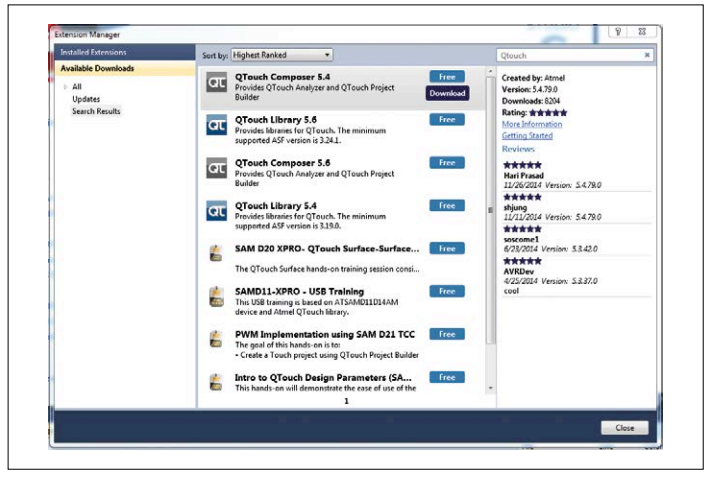


Bild 3. Bei der Suche nach den beiden Tools werden Sie schnell fündig.

Beginnen Sie mit dem QTouch-Composer. Wählen Sie ihn aus und drücken Sie auf „Download“. Es öffnet sich ein Pop-Up-Fenster, in dem Sie sich anmelden müssen (**Bild 4**). Sie können die Daten verwenden, mit denen Sie sich schon beim Herunterladen des Studios (im ersten Teil des Kurses) registriert haben. Wenn Sie sich beim Herunterladen des Studios lediglich als Gast bei myAtmel angemeldet haben, müssen Sie die Registrierung jetzt nachholen, indem Sie auf „New User Click here.“ klicken. Wenn Sie „Remember me“ auswählen, müssen Sie Ihre Daten bei der Installation der QTouch-Library nicht noch einmal eintippen.

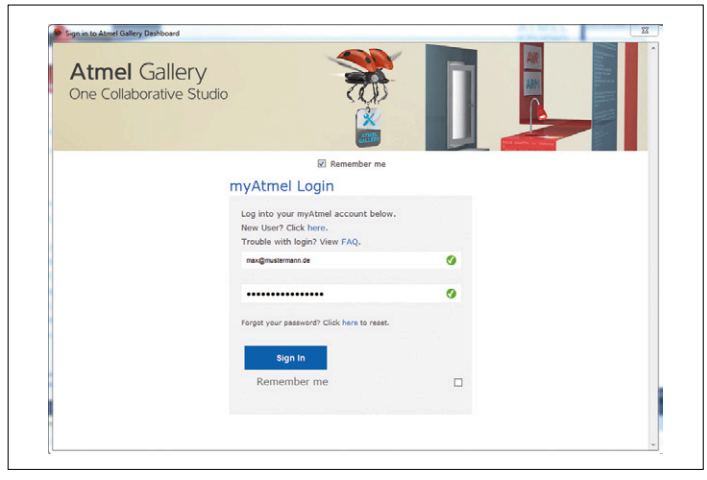


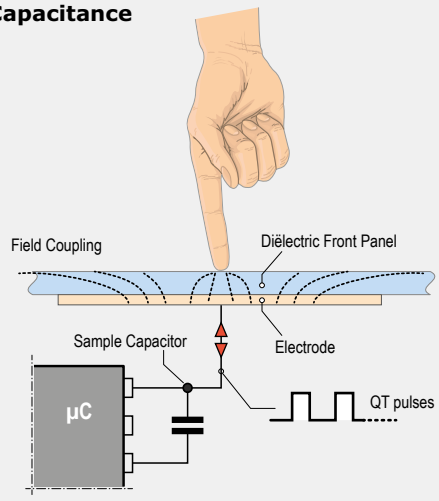
Bild 4. Dieses Anmeldefenster sehen Sie bestimmt nicht das erste Mal.

Nach der Anmeldung lädt der Extension-Manager sofort den Composer herunter und installiert ihn. Zwischendurch können Sicherheitsmeldungen Ihres Betriebssystems wie in **Bild 5** erscheinen. Da das Tool vertrauenswürdig ist, können Sie alle Änderungen am Computer zulassen. Nach dem Download, aber vor der Installation, erscheint noch eine Meldung, bei der bereits ausgewählt ist, dass das Tool ins Atmel-Studio eingebunden werden soll. Sie müssen lediglich mit „Install“ der Lizenzvereinbarung zustimmen, bevor der Installationsvorgang

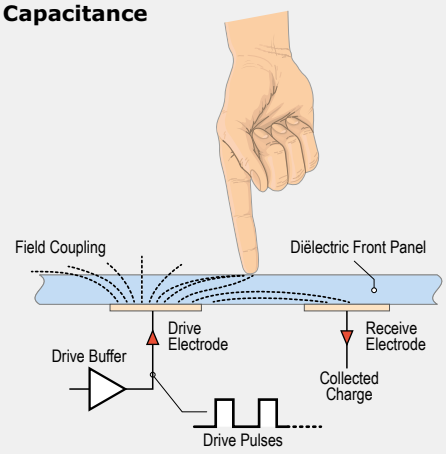
Self Capacitance vs. Mutual Capacitance

Die beiden grundlegenden QTouch-Technologien von Atmel, für die es übrigens auch verschiedene Touch-ICs gibt, werden hier grob unter die Lupe genommen und besonders die Unterschiede zwischen beiden herausgearbeitet.

Self Capacitance



Mutal Capacitance



Die Self-Capacitance-Technologie ist sehr weit verbreitet und sehr einfach, dafür aber mit einigen Nachteilen behaftet. Um zum Beispiel einen Button zu implementieren, muss lediglich eine Kupfer- als Touchfläche dienen, die mit einem Pin des Mikrocontrollers verbunden ist. Der Mikrocontroller kann nun die Kapazität zwischen der Touchfläche und der Erde, mit der der Mensch immer (mehr oder weniger) verbunden ist, messen und so eine Berührung registrieren.

Diese Touch-Technologie benötigt also keine aufwändigen Platinenlayouts und lediglich einen Mikrocontroller-Pin. Sie ist einfach und universell einsetzbar, allerdings relativ störanfällig, weshalb oft eine Abschirmung um die Touchflächen gebraucht wird und der Mensch ausreichend geerdet sein muss.

Wenn man sich das Layout der Self-Capacitance-Platine des QT1-Kits anschaut, bemerkt man schnell, dass die Touchflächen lediglich jeweils an einem Pin hängen und als einfache Kupferfläche (dunkelgrün) realisiert sind. Auch ist eine Platinenlage zur Abschirmung beziehungsweise Erdung (hellgrün) vorhanden.

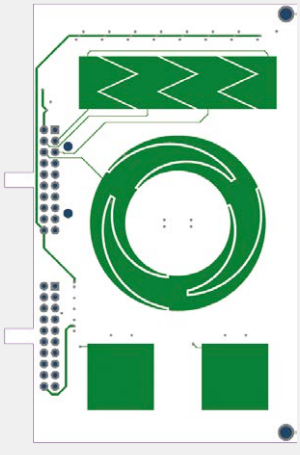
Mehr dazu unter [6] oder im Datenblatt [5] auf Seite 569.

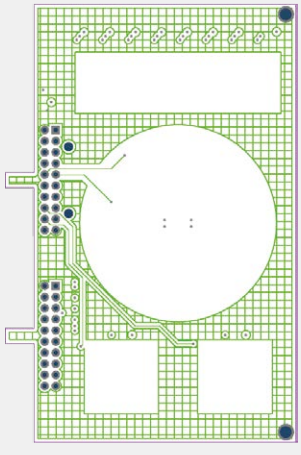
Die Mutual-Capacitance-Technologie von Atmel ist vor allem sehr zuverlässig und flexibel. Eine Touch-Fläche besteht aus zwei Kupferflächen beziehungsweise –elektroden, einer Y- und einer X-Elektrode. Die X-Elektrode sendet dabei ein (periodisches) Signal, die Y-Elektrode empfängt es. Eine Berührung des Fingers beeinflusst die Kapazität zwischen beiden Elektroden. Diese Änderung wird ermittelt. Für einen Button werden zwei MCU-Pins gebraucht, aber man kann diese Technologie in einer Matrix-Schaltung anwenden, sodass die X- und Y-Kanäle mit einer Reihe von Touchflächen gleichzeitig verbunden sind.

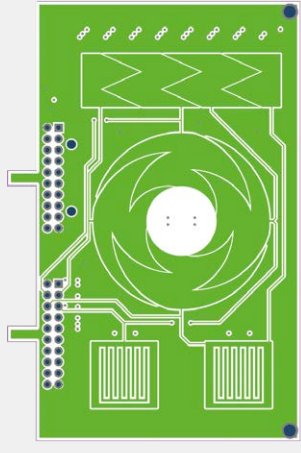
Diese Touch-Technologie benötigt zwar ein aufwändigeres Layout und mehrere Mikrocontroller-Pins für einen einzelnen Touch-Button, bei mehreren Touchflächen wie bei Buttons oder Wheels kann aber diese Technologie durch die Matrix-Schaltung punkten, da sie verglichen mit der Self-Capacitance-Technologie sogar weniger Pins braucht. Auch ist diese Technologie nicht so störanfällig und kann zum Beispiel gut auf Glasscheiben realisiert werden. Sogar ein dünner Wasserfilm (Feuchtigkeit) oder Temperaturschwankungen machen hier kaum etwas aus.

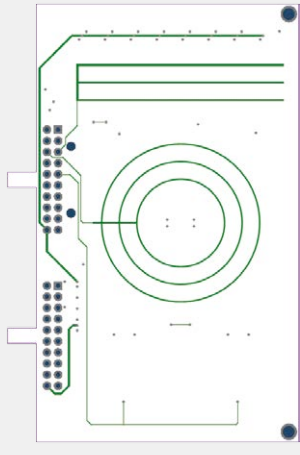
Wenn man sich das Layout der Mutual-Capacitance-Platine des QT1-Kits anschaut, bemerkt man schnell, wie die X- (hellgrün) und Y-Flächen (dunkelgrün) angeordnet sind.

Mehr dazu unter [7] oder im Datenblatt [5] auf Seite 568.









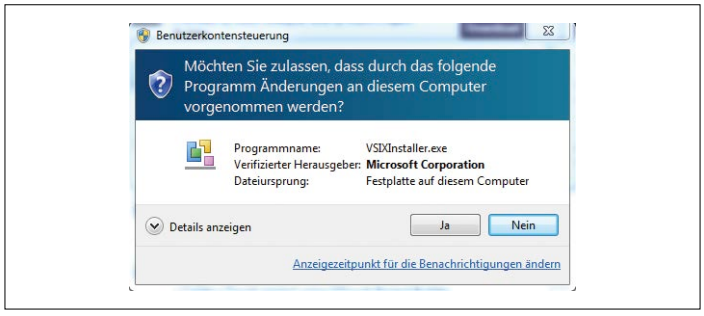


Bild 5. Je nach ausgewählter Sicherheitsstufe kann diese Windows-Meldung auftauchen.

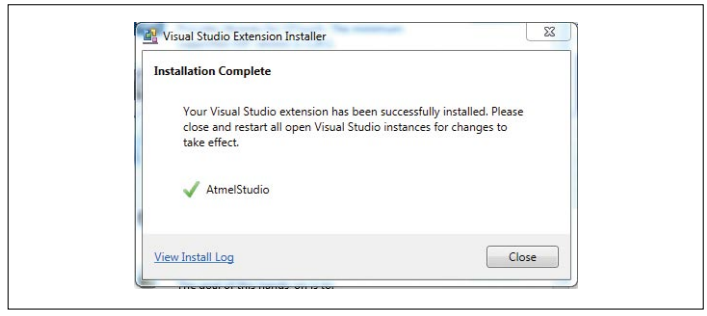


Bild 6. Sie haben es geschafft, den QTouch-Composer erfolgreich zu installieren. Jetzt nur noch die QTouch-Library…

startet. Nach kurzer Zeit erhalten Sie hoffentlich die Meldung „Installation Complete“ (**Bild 6**), die Sie mit „Close“ schließen müssen. Nun weist der Extension-Manager darauf hin, dass das Studio neu gestartet werden muss, um das Tool automatisch und endgültig ins Atmel-Studio einzubinden („Restart Now“). Nach dem Neustart öffnen Sie wieder den Extension-Manager, suchen nach „QTouch“ und erfahren in den Suchergebnissen durch zwei Symbole beim QTouch-Composer, dass er heruntergeladen und installiert wurde.
Der gleiche Installationsvorgang wird nun mit der QTouch-Library durchgeführt. Nach dem erneuten Neustart des Studios erscheint die QTouch-Startseite (**Bild 7**) mit verschiedenen QTouch-Libraries, zum Beispiel für besondere Safety-Funktionen oder für Mikrocontroller, die über keinen Touch-Controller als Hardware verfügen. Falls nicht, müssen Sie diese Startseite oben mit dem Icon direkt unter „Help“ manuell aufrufen. Nun sind alle nötigen Software-Tools installiert.

Das erste Touch-Projekt

Jetzt können wir das erste Projekt starten. Wir wollen das Mutual-Capacitance-Board mit allen Touchflächen in Betrieb nehmen. Dabei soll bei einer Berührung der Buttons die jeweilige LED aufleuchten und beim Touchslider gleich mehrere, je nach der Berührungs-Position. Das Wheel wird „virtuell“ in drei gleiche Winkel von 120° geteilt, für die jeweils eine der LEDs der RGB-LED aufleuchten soll.

Die LEDs kann man mit den sattsam bekannten Port-Befehlen ansteuern, aber wie setzt man eine Touch-Erkennung mit dem QTouch-Composer und der QTouch-Bibliothek in Betrieb? Der Composer wird den Code für die Touch-Funktionen generieren, wenn wir vorher die benötigten Touch-Elemente, die daran angeschlossenen Pins der MCU und die gewünschten Einstellungen des PTCs visuell auswählen. Dazu ruft man den Projekt-Wizard unter *File/New/Project…* auf. Dort müssen Sie nun statt des üblichen Projekts „C ASF Board Project“ das „GCC C QTouch Executable Project“ auswählen und ihm einen Namen wie zum Beispiel „First test with QT1 MC“ vergeben (**Bild 8**). Wenn Sie nun auf „OK“ klicken, erscheint der QTouch-Project-Builder. Dort wählen Sie die normale QTouch-Bibliothek mit „Create QTouch Library Project“. Über die anderen Libraries können Sie sich später informieren. Im QTouch-Project-Builder müs-

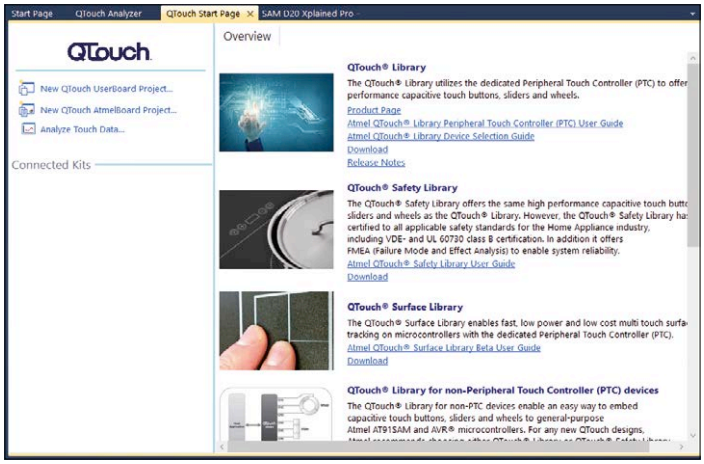


Bild 7. Freundliche Begrüßung vom QTouch-Composer.

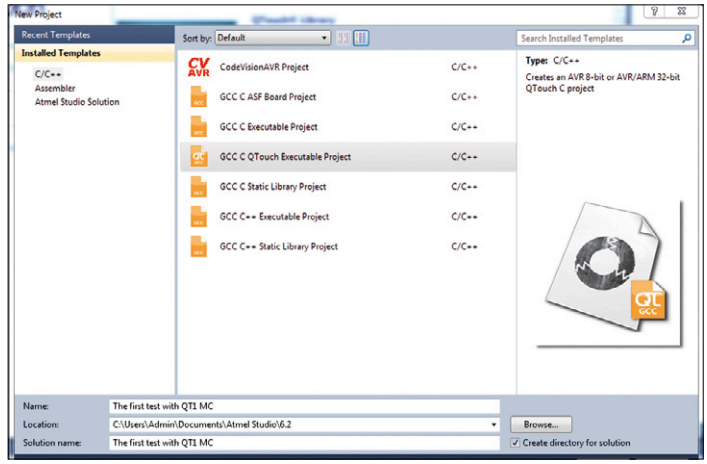


Bild 8. Den Project Wizard kennen Sie schon, nur müssen Sie dieses Mal ein QTouch-Projekt erzeugen.

Das neue Atmel Studio 7 ist da!

Freuen Sie sich, das neue Studio 7 ist (in einer Beta-Version) da! Sie können es unter **http://atmel-studio.s3-website-us-west-2.amazonaws.com/#** herunterladen und schon ausprobieren.
Das neue Studio setzt noch mehr auf visuelle Tools, Sie werden sich bestimmt schnell daran gewöhnen.

sen Sie nun Schritt für Schritt die für das Projekt benötigten Informationen eingeben. Als erstes werden die auf dem Board vorhandenen Touch-Elemente durch Doppelklick auf das Element in das Arbeitsfeld (QTouch Kit) aufgenommen (**Bild 9**). Beginnen Sie mit den beiden Buttons, danach folgen der Slider und am Ende das Wheel. Es erscheint ein kleines Fenster, in dem Sie nach den Maßen gefragt werden (**Bild 10**). Geben Sie dort jeweils die Maße der Touch-Elemente an, die in **Tabelle 1** zu finden sind (diese werden lediglich für die visuelle Darstellung benötigt). Die Elemente müssen aber nicht in derselben Anordnung wie auf dem Board platziert werden - nur annähernd, für eine bessere Übersicht.

Nachdem alle gewünschten Elemente in der Auswahl stehen, öffnen Sie die nächste Registerkarte „Device Selection“ des Project-Builders. Hier wird die von uns verwendete MCU „SAMD20J18“ ausgewählt (**Bild 11**). Nach der Auswahl wird

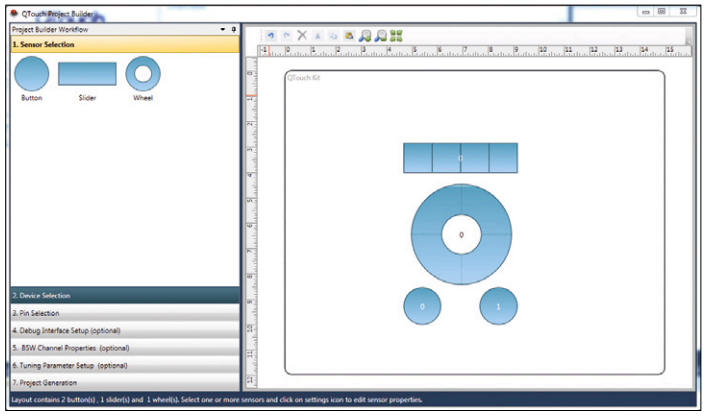


Bild 9. Der grafische QTouch-Project-Generator möchte zuerst wissen, welche Touch-Elemente Sie auswerten wollen.

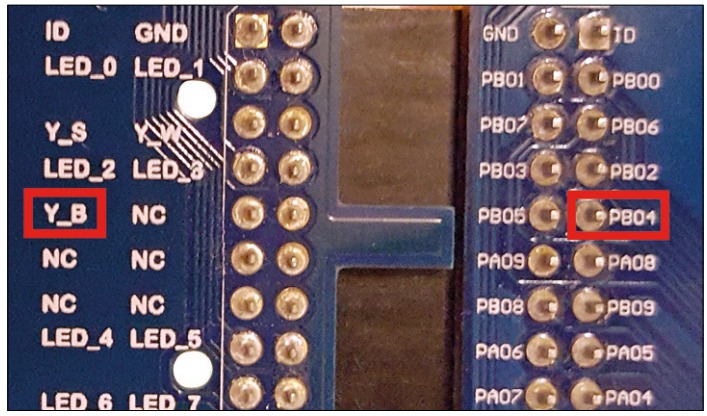


Bild 13. So können Sie auf der Rückseite der beiden miteinander verbundenen Boards ablesen, welche Touch-Elektrode an welchem Pin liegt.

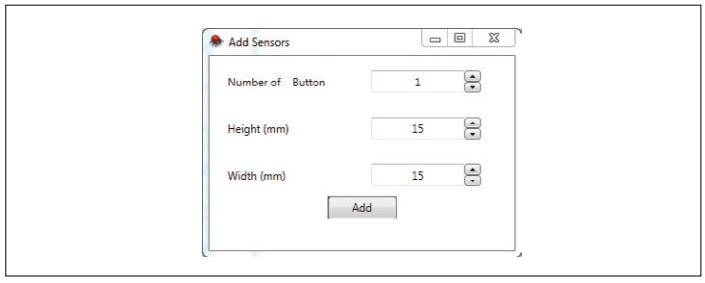


Bild 10. Die Maße spielen bei kleinen Projekten kaum eine Rolle, doch bei größeren wird es dann viel übersichtlicher.

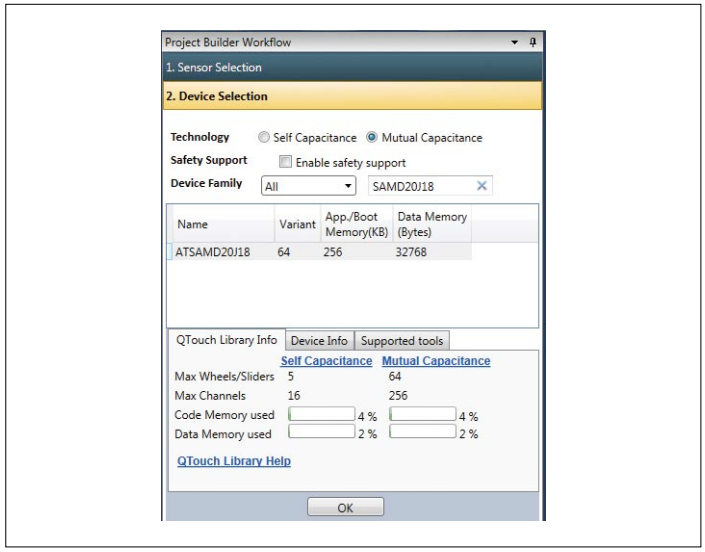


Bild 11. Hier können Sie unten links weitere Eigenschaften der MCU einsehen.

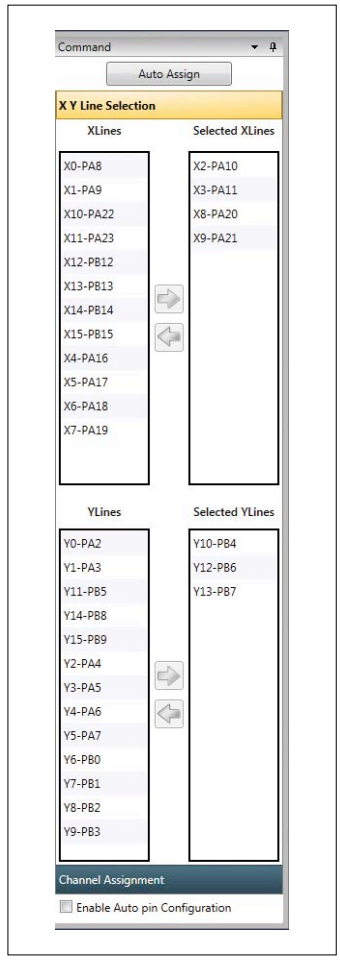


Bild 14. Rechts müssen Sie die Auswahl der benutzten Kanäle sorgfältig eintragen und ...

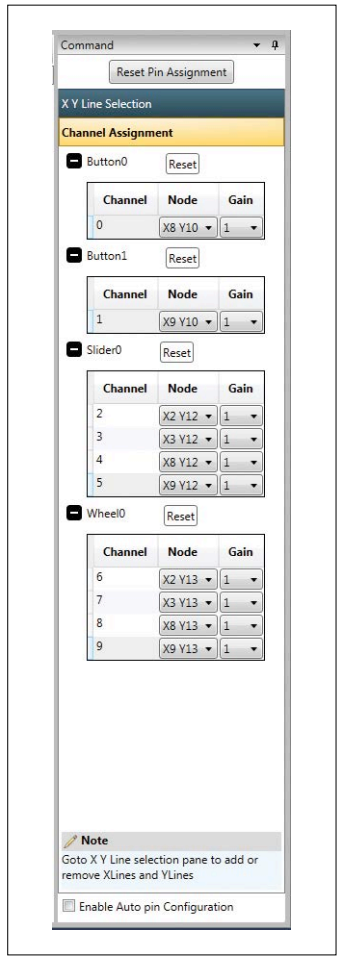


Bild 15. ... nun die Zuordnungen zu einzelnen Touch-Elektroden.

unten der für die Auswertung der Touch-Elemente in den beiden Touch-Modi erforderliche Speicherplatz gezeigt und auch, wie viele Kanäle beziehungsweise Touch-Elemente mit dieser MCU jeweils implementiert werden können. Bei Mutual-Capacitance können dank der Matrix-Schaltung viel mehr Touch-Elemente ausgewertet werden. Wir arbeiten in diesem Projekt mit dem Mutual-Capacitance-Board und kreuzen ganz oben den Radio-Button entsprechend an.

Mit einem „OK“ gelangt man zur nächsten Registerkarte mit der Pin-Auswahl. Hier werden in zwei Schritten die Pins (Kanäle) mit den einzelnen Elektroden verbunden. Zuerst müssen Sie die benötigten Kanäle auswählen und sie dann den einzelnen Elektroden zuweisen. Davor sollten Sie aber auf „Reset Pin Assignment“ klicken, damit die vorgeschlagenen Pin-Zuordnungen des Studios gelöscht werden. Danach können Sie den Tabellen in **Bild 12** oder dem Benutzerhandbuch [2] entnehmen,

Tabelle 1. Ungefähre Maße der Touch-Elemente des QT1 Xplained Pro Mutual Capacitance Boards

Buttons	15 mm x 15 mm
Slider	45 mm x 12 mm
Wheel	8 mm $Radius_{in}$ x 20 mm $Radius_{out}$

Tabelle 2. Ungefähre Maße der Touch-Elemente des QT1 Xplained Pro Self Capacitance Boards

Buttons	15 mm x 15 mm
Slider	45 mm x 12 mm
Wheel	12 mm $Radius_{in}$ x 20 mm $Radius_{out}$

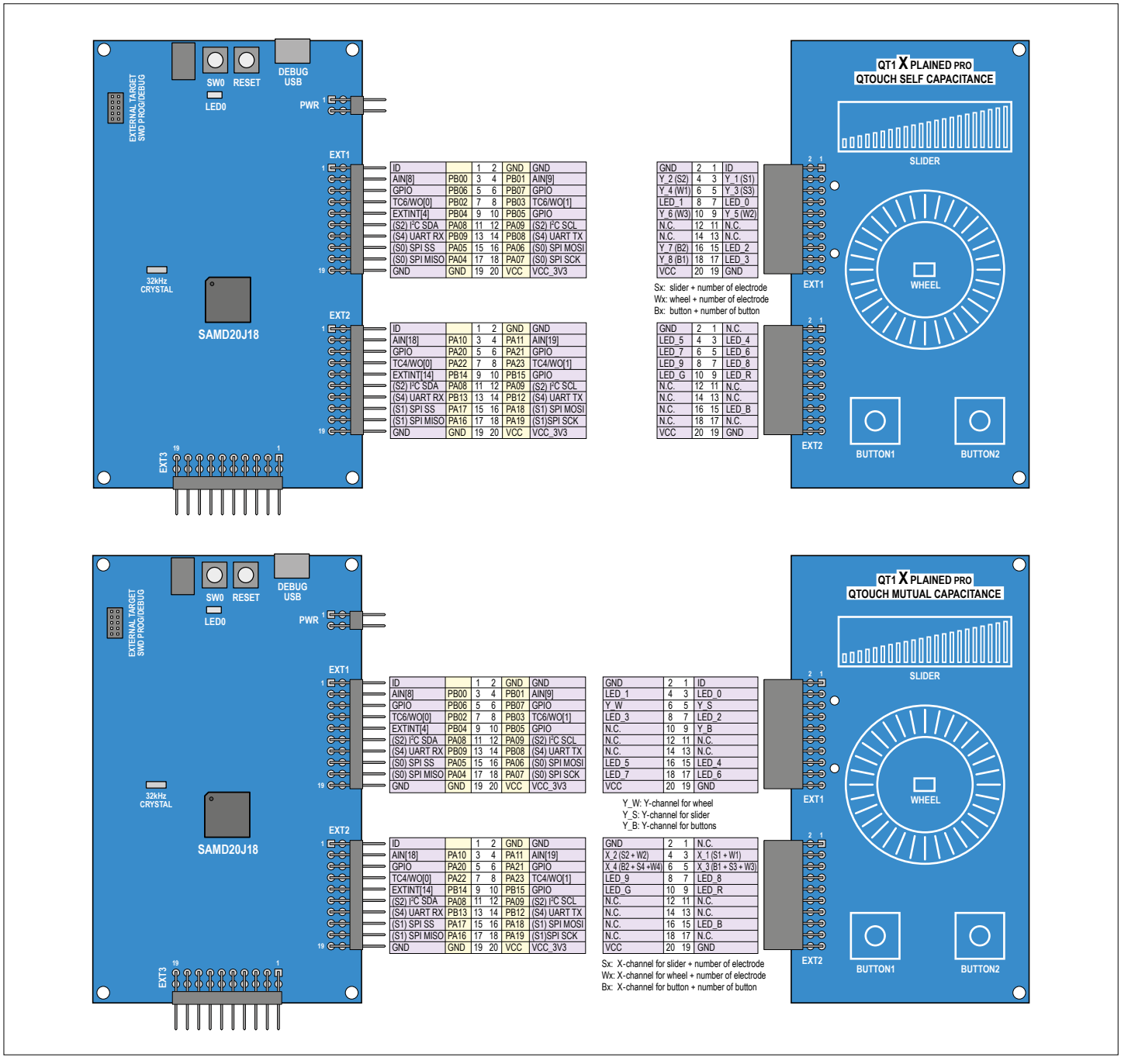


Bild 12. In den Tabellen sind die Anschlüsse der beiden QT1 Xplained Pro Boards aufgeführt.

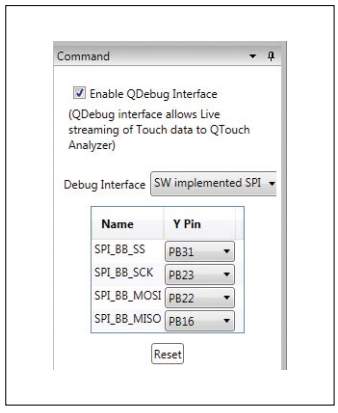


Bild 16. So sind die Interface-Pins richtig zugeordnet.

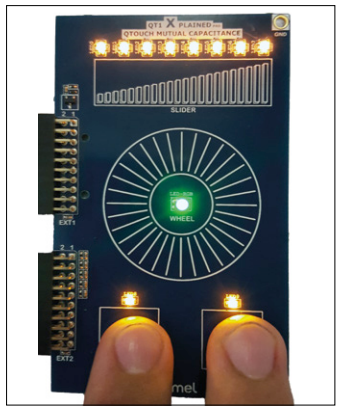


Bild 17. Die LEDs bei Wheel und Slider leuchten auch, nachdem die Touch-Elemente losgelassen wurden. Dies kann man bei Bedarf ändern.

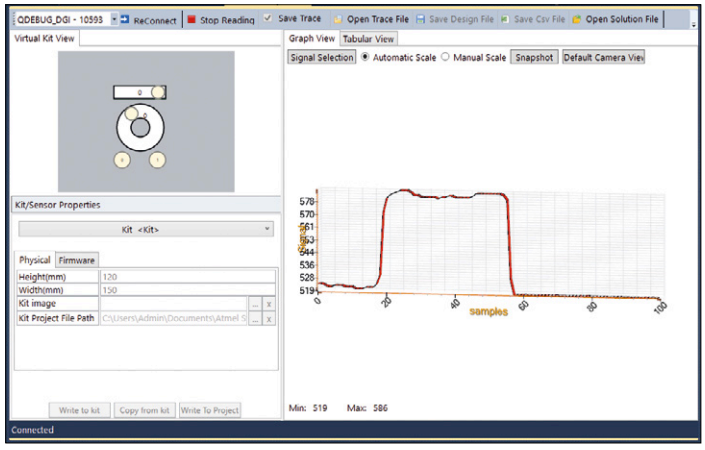
Listing 1. Symbolische Konstanten und benötigte Variablen.

```
#define LED0 PIN_PB00
#define LED1 PIN_PB01
#define LED2 PIN_PB02
#define LED3 PIN_PB03
#define LED4 PIN_PA05
#define LED5 PIN_PA06
#define LED6 PIN_PA04
#define LED7 PIN_PA07
#define LED8 PIN_PA22
#define LED9 PIN_PA23
#define LEDR PIN_PB14
#define LEDG PIN_PB15
#define LEDB PIN_PA17

void configure_port_pins(void);

bool sensor_state_button_1, sensor_state_button_2;
uint8_t sensor_state_wheel, sensor_state_slider;
```

Bild 18. Signalverlauf bei kurzzeitiger Betätigung von Button 1.



Listing 2. Port-Konfigurationsfunktion für die benötigten LEDs.

```
void configure_port_pins(void)
{
  struct port_config config_port_pin;
  port_get_config_defaults(&config_port_pin);

  config_port_pin.direction = PORT_PIN_DIR_OUTPUT;
  port_group_set_config(&PORTB, 49167, &config_port_pin);
  port_group_set_config(&PORTA, 12714224, &config_port_pin);
}
```

welchen Kanälen auf dem QT1-MC-Board welche Touch-Elektroden zugeordnet sind. Auf der Platinenrückseite der miteinander verbundenen Boards kann man die an die Touch-Elektroden angeschlossenen MCU-Pins ablesen (**Bild 13**). Die im Atmel Studio genannten Kombinationen von Pin und Kanalname entsprechen nicht denen des QT1-MC-Boards. PA10 wird zum Beispiel im Studio als X2 bezeichnet, damit ist aber der Kanal X_1 unseres Boards verbunden. Nachdem alle Kanäle ausgewählt sind (**Bild 14**), müssen Sie in der nächsten Registerkarte „Channel Assignment“ rechts die Atmel-Studio-Kanalnamen für die Auswahl verwenden (**Bild 15**). Für die erste Elektrode des Sliders wählt man zum Beispiel X2 Y12. Nach der Auswahl können Sie zur nächsten Registerkarte gehen, um das Debug-Interface für den später sehr nützlichen QTouch-Analyzer mit den entsprechenden EDBG-Pins zu konfigurieren. So kann das Tool später durch den EDBG die aktuellen QTouch-Daten auslesen. Die einzelnen Interface-Pins finden Sie im User Guide [3]; Sie müssen sie wie in **Bild 16** gezeigt im QTouch-Project-Builder auswählen.

Auf den nächsten beiden Registerkarten können Einstellungen für den PTC vorgenommen werden. Zunächst bleibt alles bei den Standardeinstellungen. In der siebten und der letzten Registerkarte sehen Sie nun eine Zusammenfassung aller im Project-Builder vorgenommenen Einstellungen, die lediglich mit „Generate Project“ bestätigt werden müssen, um das Projekt zu generieren. Dies kann eine Zeit dauern, doch danach bekommen Sie das fertig generierte Projekt zu Gesicht. Die Dateien **touch_api_ptc.h**, **touch.c** und **touch.h** sind die interessantesten und enthalten die API. Sie sollten auf jeden Fall einen Blick darauf werfen.

Nun zum Code! Die Touch-Erkennung ist schon im Hintergrund realisiert, da gibt es nichts daran zu rütteln. Lediglich die Hauptfunktion, das Aufleuchten der LEDs, muss verwirklicht werden. Dazu müssen nach dem Einbinden von **asf.h** zunächst wie im **Listing 1** gezeigt am Anfang der Main-Datei symbolische Konstanten für die einzelnen QT1-LED-Pins, ein Funktionsprototyp für die spätere Konfigurations-Funktion der Pins und schließlich später benötigte Variablen deklariert werden.

Bevor wir zum Kern des Programms kommen, muss nach allen anderen Konfigurations-Funktionen, die in der main-Datei automatisch für QTouch generiert wurden, die Konfigurations-Funktion für die LED-Pins eingefügt werden (**Listing 2**). Darin werden alle LED-Pins als Ausgänge konfiguriert. Leider liegen die LED-Pins an zwei verschiedenen Ports, sodass immer beide

Weblinks

[1] www.atmel.com/images/atmel-42195-qtouch-general-library-peripheral-touch-controller_user-guide.pdf
[2] www.atmel.com/Images/Atmel-42193-QT1-Xplained-Pro_User-Guide.pdf
[3] www.atmel.com/Images/Atmel-42102-SAMD20-Xplained-Pro_User-Guide.pdf
[4] www.elektormagazine.de/150368
[5] www.atmel.com/Images/Atmel-42129-SAM-D20_Datasheet.pdf
[6] www.atmel.com/products/TouchSolutions/bsw/qtouch.aspx
[7] www.atmel.com/products/TouchSolutions/bsw/qmatrix.aspx
[8] https://hbe-shop.de/Art-2399861-ATMEL-ATQT1-XPRO-XPLAINED-PRO-EXTENSION-BOARD-SAMD20-MCU

nacheinander konfiguriert und angesteuert werden müssen. Alle Werte und Masken bei den Port-Befehlen im ganzen Programm lassen sich stets ganz einfach (wie bei 8-bit-Mikrocontrollern) nach der Wertigkeit der einzelnen Bits berechnen. Sie können die Werte auch binär oder hexadezimal angeben. Diese Konfigurations-Funktion muss gleich nach dem ganzen generierten Code und vor der Endlosschleife aufgerufen werden:

```
configure_port_pins();
```

In der Main-Datei müssen Sie ganz am Ende nach dem generierten QTouch-Code und den Kommentaren den Code in **Listing 3** einfügen. Am Anfang werden hier die Zustände einzelner Touch-Elemente mit `GET_MUTLCAP_SENSOR_STATE(x)` und `GET_MUTLCAP_ROTOR_SLIDER_POSITION(x)` zunächst abgefragt und in den gewünschten Variablen abgespeichert. Das erste Makro kann Buttons abfragen, das zweite Wheels und Slider. Die Makros benötigen lediglich die Sensor-Nummer (Buttons: 0 und 1; Slider: 0 und Wheel: 1). Sie geben bei Buttons einen booleschen und bei Wheels und Slidern einen 8-bit-Wert aus. Die Werte werden in den schon erwähnten Variablen gespeichert. Der Wert der Slider wird hier aber gleichzeitig durch 256/8 = 32 und der Wert des Wheels durch 256/3 ≈ 85 geteilt, so dass wir acht bzw. drei LEDs ansteuern können.

Die beiden LEDs 8 und 9 werden mit den folgenden Zeilen entsprechend dem (wegen der Beschaltung) umgekehrten Touch-Wert der Buttons geschaltet:

Anzeige

```
port_pin_set_output_level(LED8, !sensor_state_button_1);
port_pin_set_output_level(LED9, !sensor_state_button_2);
```

Es folgen nur noch zwei einfache Switch-Abfragen. Die erste schaltet die LEDs 0...7 je nach Zustand des Slider-Werts in `sensor_state_slider`, die zweite schaltet die RGB-LED je nach Wheel-Position in `sensor_state_wheel`.

Listing 3. Das Hauptprogramm mit der Auswertung der Touchflächen und dem entsprechenden Schalten der LEDs.

```
sensor_state_button_1 = GET_MUTLCAP_SENSOR_STATE(0);
sensor_state_button_2 = GET_MUTLCAP_SENSOR_STATE(1);
sensor_state_slider = GET_MUTLCAP_ROTOR_SLIDER_POSITION(0) / 32;
sensor_state_wheel = GET_MUTLCAP_ROTOR_SLIDER_POSITION(1) / 85;
port_pin_set_output_level(LED8, !sensor_state_button_1);
port_pin_set_output_level(LED9, !sensor_state_button_2);
switch (sensor_state_slider)
{
  case 0:
  port_group_set_output_level(&PORTB, 15, 14);
  port_group_set_output_level(&PORTA, 240, 240);
  break;

case 1:
  port_group_set_output_level(&PORTB, 15, 12);
  port_group_set_output_level(&PORTA, 240, 240);
  break;

case 2:
  port_group_set_output_level(&PORTB, 15, 8);
  port_group_set_output_level(&PORTA, 240, 240);
  break;

...

}

switch (sensor_state_wheel)
{
  case 0:
  port_pin_set_output_level(LEDR, 0);
  port_pin_set_output_level(LEDG, 1);
  port_pin_set_output_level(LEDB, 1);
  break;

case 1:
  port_pin_set_output_level(LEDR, 1);
  port_pin_set_output_level(LEDG, 0);
  port_pin_set_output_level(LEDB, 1);
  break;

case 2:
  port_pin_set_output_level(LEDR, 1);
  port_pin_set_output_level(LEDG, 1);
  port_pin_set_output_level(LEDB, 0);
  break;
}
```

Das Projekt kann jetzt kompiliert werden. Gegebenenfalls erscheint eine Fehlermeldung, dass die Dateien `power.h` und `reset.h` nicht gefunden werden können. Dann müssen Sie `#include <power.h>` und `#include <reset.h>` in der Datei `asf.h` jeweils mit `//` auskommentieren. Spätestens dann sollten Sie aber das Board in Betrieb setzen und das Programm wie im **Bild 17** ausprobieren können.

Zusätzlich können Sie auch das QTouch-Analyzer-Tool mit dem dazugehörigen Icon in Atmel Studio starten. In dem Fenster wählen Sie oben links Ihr Board aus, dann klicken Sie auf „Connect“. Ein weiteres Fenster mit den Einstellungen öffnet sich, bei dem Sie lediglich auf „OK“ klicken müssen. Nun ist das Tool durch den EDBG mit der MCU verbunden, um die Touch-Daten abzurufen. Sie können oben in der Mitte unter „Signal Selection“ die Werte, die Sie beobachten wollen, auswählen. Machen Sie zum Beispiel einen Haken bei „Signal“ unter „Button0“ und starten die Messung mit „Start Reading“ oben links. Das relative Touch-Signal vom Button 1 lässt sich im Live-Diagramm (Graph View) oder in der Tabelle (Tabular View) beobachten (**Bild 18**). Mit diesem Tool können Sie ein Stück weit in das Herz des PTCs schauen und im Nachhinein vielleicht noch einige Einstellungen im QTouch-Project-Builder, den Sie jederzeit mit Rechtsklick auf Ihr Projekt im Solution-Explorer aufrufen können, perfektionieren. Das fertige Projekt gibt es auch unter [4] als Download.

Self-Capacitance

Das gleiche Projekt können Sie nun auch mit dem Self-Capacitance–Board realisieren, wobei die Software sich kaum ändert (die Befehle sind ähnlich, eine Hilfe sind auch die automatisch generierten Kommentare in der Endlosschleife). Beim QTouch-Project-Builder ist die Umsetzung aber etwas einfacher, da es jetzt ja keine zwei Kanäle gibt, an die die Touchflächen angeschlossen sind.

Erstellen Sie mit Hilfe des QTouch-Project-Builders ein neues Projekt namens „First test with QT1 SC“, wobei Sie sich dieses Mal für Self-Capacitance entscheiden und den Touch-Elektroden nur einen Pin (Y-Kanal) zuordnen müssen. Der Code sieht sehr ähnlich aus, jedoch müssen Sie vor allem die Touch-Befehle ändern und die LED-Pins anpassen. Alternativ können Sie sich aber auch direkt das fertige Projekt unter [4] anschauen.

In jedem Ende liegt ein Anfang!

Wir hoffen sehr, dass Ihnen dieser Kurs behilflich war, die nötigen Grundlagen der ARM-Familie zu erlernen und dass Sie nun in der Lage sind, eigene Projekte umzusetzen. Nun sollen Sie zu Wort kommen, denn schließlich gibt es immer Raum für Verbesserungen. Wie hat Ihnen der Kurs gefallen? Was können wir nächstes Mal besser machen? Oder wollen Sie uns Ihr eigenes ARM-Projekt vorstellen? Schreiben Sie uns an: redaktion@elektor.de! ⏮

(150368)

Fluoreszenzanzeigen

Bemerkenswerte Bauteile

Von Neil Gruending (Kanada)

In früheren Ausgaben habe ich schon das eine oder andere Wort zu Röhren verloren. In dieser Tradition stehen auch Fluoreszenzanzeigen (englisch VFDs = Vacuum Fluorescent Displays), da sie ebenfalls in diese Kategorie fallen. Fast jeder kennt sie und normalerweise macht man sich kaum Gedanken darüber, denn sie funktionieren einfach problemlos und das jahrelang.

Die ersten Siebensegment-VFDs wurden schon in den 1960ern als stromsparender und langlebiger Ersatz für die damaligen Nixie-Röhren entwickelt. Seither gab es mit LEDs und LCDs neuere Display-Kategorien, die VFDs aus vielen Anwendungen verdrängt haben. Aber nicht ganz, denn man findet sie immer noch in Messgeräten und auch in anderer Elektronik, wo eine hohe Lebensdauer und ein großer Ablesewinkel wichtig sind. VFDs gibt es sogar im Auto, da sie robust sind und hohe Temperaturschwankungen aushalten.

Das Erkennungsmerkmal für VFDs ist ihr typisches blaugrünes Leuchten (siehe **Bild 1**), selbst wenn es sie auch in anderen Farben gibt. VFDs sind im Prinzip Trioden mit Kathodenheizung und einem Steuergitter (manchmal auch mehreren) sowie einer phosphorbeschichteten Anode. **Bild 2** zeigt das Prinzip; es stammt von [1]. Die Heizung generiert Elektronen, die durch die Anodenspannung zur Anode beschleunigt werden, wenn sie nicht durch eine negative Gitterspannung abgeblockt werden. Das Auftreffen auf der Anode bringt dann die Phosphorbeschichtung zum Leuchten.

Der Triodenaufbau eines VFD bedeutet, dass zu seiner Ansteuerung gegenüber anderen Displays etwas komplexere Elektronik erforderlich ist. Eine Herausforderung ist, die Spannung über der Heizung konstant zu halten, damit das Display überall gleich hell ist. Zwar könnte man die Heizung mit Gleichstrom betreiben, doch der Spannungsabfall entlang des Heizfadens würde die Helligkeit ungleichmäßig machen. Daher ist das Standardverfahren eine Heizung mit Wechselspannung. Typischerweise wird hier ein

Bild 1. Display des Messgeräts Fluke 8808A.

Trafo genommen, dessen Mittelanzapfung geerdet wird, sodass die Spannungsdifferenzen gerade bei größeren Displays verringert werden.

Gitter und Anoden werden dann natürlich mit Gleichspannungen versorgt. Diese liegen oft deutlich höher, als das für CMOS-ICs gut wäre. Bei manchen kleinen VFDs gibt es nur einen Gitteranschluss, und dafür sind alle Anoden herausgeführt. Solche „statischen" Displays sind besonders einfach anzusteuern. Da das Gitter immer eingeschaltet ist, werden geringere Anodenspannungen für eine ausreichende Helligkeit benötigt.

Die meisten VFDs aber sind deutlich komplexer und benötigen eine Multiplex- oder gar eine aktive Ansteuerung. Beim Multiplexen sind die Anodensegmente einzelner Ziffern miteinander verbunden und die Ziffern werden über je ein Gitter nacheinander kurz aktiviert. Beim Multiplexverfahren ist daher jede Ziffer für einen Bruchteil der Zeit eingeschaltet, was gegenüber statischen Displays höhere Gitter- und Anodenspannungen erfordert, um genügend Helligkeit zu erzeugen. Es gibt aber sogar CMOS-kompatible VFD-Module, und zudem sind spezielle VFD-Treiber-ICs erhältlich. Von daher ist die Hürde gar nicht so hoch,

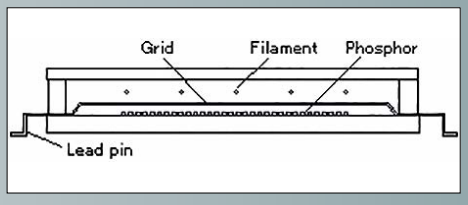


Bild 2. Prinzipaufbau eines VFDs.

wenn Sie bei Ihrem nächsten Projekt einmal ein VFD einsetzen wollen. ⏮

(150484)

Weblink

[1] www.noritake-elec.com/display/vfd_operation.html

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

EAGLE Tipps & Tricks (3)

Beispielprogramme

Von **Neil Gruending** (Kanada)

In dieser dritten Folge geht es um Feinheiten der automatisch erstellten Stückliste.

Nachdem klar ist, was man bei Eagle unter einem ULP (User Language Program) versteht, geht es jetzt darum, die Ausgabe der Stückliste durch Modifikation des BOM-Scripts in einem eigenen Format zu ermöglichen. Die kurz als „Englisches Elektor-Format“ bezeichnete Ausgabe sieht so aus wie in **Listing 1**:

```
R1 = 2.26MΩ 1%, 0.063W, 0603
R2,R6 =  2.43MΩ 1%, 0603
C1,C2 = 22nF, 25V, 0603
C3,C4 = 10pF, 25V, 0603
IC2 = SN74LVC2G04DBVR
```

Alle Referenzen (Bauteilnummern) werden gruppiert und bei der Ausgabe durch Kommas getrennt. Gefolgt vom Gleichheitszeichen kommen dann Bauteilewerte, deren Teile ebenfalls durch Kommas getrennt werden. Und los geht’s!

Hinzufügen des Elektor-Knopfs

Als Erstes braucht es eine Elektor-Option beim Ausgangsformat in der Dialogbox (siehe **Listing 2**):

```
dlgGroup(tr ("List type"))
{
  dlgRadioButton(tr ("&Parts"), ListType)
                              GeneratePartList ();
  dlgRadioButton(tr ("&Values"), ListType)
                            GenerateValueList ();
  dlgRadioButton(tr ("&Elektor"), ListType)
                          GenerateElektorList ();
  dlgCheckBox(tr ("List &attributes"),
                                  UseAttributes) {
    if (!UseAttributes) {
      NumParts = 0;
    }
    CollectPartData(CurrentVariant);
    GenerateList();
  }
}
```

Hierzu wird der Dialog-Code des BOM-ULP insofern modifiziert, als ein sogenannter „radio button“ mit der Bezeichnung Elektor in der Gruppe „List type“ hinzugefügt wird. Ein solcher Button besteht aus einem kleinen, anklickbaren Kreis. Sind mehrere solcher Buttons gruppiert wie bei Listing 2, dann sorgt Eagle dafür, dass nur ein einziger davon aktiviert sein kann. Der erste *dlgRadioButton*-Parameter ist der daneben anzuzeigende Text. Wenn im Text ein &-Zeichen auftaucht, wird der nächste Buchstabe unterstrichen dargestellt. Der Anwender kann diesen Knopf dann mit der Tastatur per Alt-Taste + dem unterstrichenen Buchstaben auswählen. Der zweite Parameter ist ein Integer-Wert, der speichert, welcher der gruppierten Knöpfe gerade ausgewählt ist. Der folgende optionale Befehl

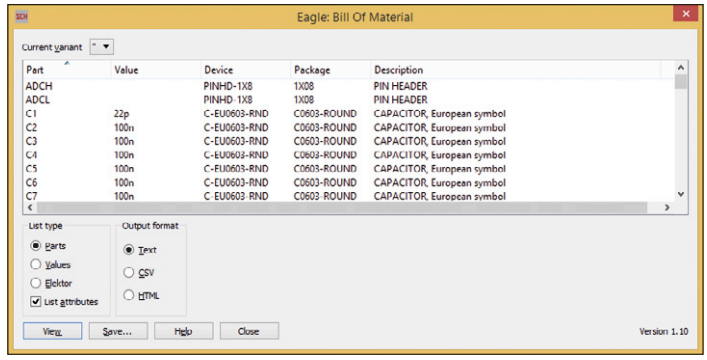


Bild 1. Die neue Option „Elektor“ im Stücklisten-Dialog.

wird bei der Auswahl dieses Knopfs ausgeführt.
Hier wird ein Knopf mit dem Text „Elektor“ erzeugt. Der ausgewählte Knopf steht in der globalen Variable ListType. Ist „Parts“ ausgewählt, steht in ListType eine „0“ und bei „Elektor“ eine „2“. Ist Elektor ausgewählt, wird anschließend die Funktion *GenerateElektorList* ausgeführt, wodurch der angezeigte Text ein Update erfährt, doch dies wird später implementiert. Die „List attributes“ des *dlgCheckBox*-Objekts zeigen, wie man einen Funktions-Block in Klammern statt eines einzigen Funktionsaufrufs verwendet. Wenn jetzt das BOM-Script ausgeführt wird, erscheint die Elektor-Option wie in Bild 1 gezeigt, auch wenn noch nichts passiert, wenn man darauf klickt.

Erstellen der GenerateElektorList

Das BOM-ULP verwendet die globale Variable *Lines* zur Speicherung von Text für die erzeugte Stückliste. Der Inhalt von *Lines* wird dann zum Update des angezeigten Texts und zur Ausgabe der Stückliste in eine Datei genutzt. Man muss also lediglich die Variable *Lines* mit der Funktion *GenerateElektorList* bearbeiten, und das BOM-ULP wird die Arbeit erledigen.
Bild 1 zeigt die Ausgabe der Stückliste, bei der die Bauteile zwar pro Typus aufeinander folgen, aber für jedes Bauteil eine Zeile mit den Eigenschaften verwendet wird. Das entspricht nicht so ganz dem Elektor-Format, aber ist schon nahe dran. Es fehlt die Gruppierung der gleichen Bauteile in der gleichen Zeile. Da dies durch die Funktion *GenerateValueList* erledigt wird, nimmt man sie am besten als Grundlage für die eigene Funktion *GenerateElektorList*, und modifiziert sie wie in **Listing 3**:

```
void GenerateElektorList (void)
{
  int NumLines = 0;
  int Index [];
  string attr[], s, s_val;
```

```
  if (UseAttributes) s = strjoin(AttributesList, ',');

  Lines[NumLines] = tr ("Parts = Value, Description,
                                    Device, Package");
  if (s != "") Lines[NumLines] += ", " + s;
  NumLines++;
  sort (NumParts, Index, PartValue, PartDevice,
               PartPackage, PartAttributes, PartName,
                                        PartHeadline);
  for (int n1 = 0, n2 = 0; ++ n2 <= NumParts; )
  {
    int i1 = Index [n1];
    strsplit (attr, PartAttributes[i1], Separator);
    if (UseAttributes) s = strjoin(attr, ',');
    s_val = attr[i1];
    if (n2 < NumParts)
    {
      int i2 = Index [n2];
      strsplit (attr, PartAttributes[i2], Separator);
      if (PartValue[i1] == PartValue[i2] &&
          PartDevice[i1] == PartDevice[i2] &&
          PartAttributes[i1] == PartAttributes[i2])
        continue;
    }
    Lines[NumLines] = "";
    for (;;)
    {
      Lines[NumLines] += PartName[i1];
      if (++n1 < n2)
      {
        i1 = Index [n1];
        Lines[NumLines] += ", ";
      }
      else
        break;
    }

    Lines[NumLines] += " = " + PartValue[i1] +
                       ", " + PartHeadline[i1] +
                       ", " + PartDevice[i1] +
                       ", " + PartPackage[i1] +
                       ", " + s;

    NumLines ++;
  }
  Lines[NumLines] = "";
}
```

Der erste Teil der Funktion holt alle vom User definierten Attribut-Namen, die in *AttributeList* als Array abgelegt sind, und kombiniert sie zu einem durch Kommas getrennten String. Diese extra Bezeichnungen werden dann zur ersten Zeile *Line[0]* hinzugefügt. *Line[0]* ist etwas Besonderes, da damit die Spalten-Labels für den Text definiert werden. Hier aber gibt es lediglich eine Spalte mit den kompletten Daten.

Nun werden alle Teildaten so sortiert, dass alle Bauteile nach Typ und identische Bauteile nebeneinander in Arrays abgelegt sind. Der Code durchläuft zur Gruppierung gleicher Bauteile alle Bauteile in den Arrays, und updated dann die nächste Zeile mit den Bauteile-Infos. Zuerst werden die Bauteilbezeichnungen (*PartName*) hinzugefügt und falls mehrere vorhanden sind, diese durch Kommas getrennt. Dann kommt das Gleichheitszeichen gefolgt vom Bauteile-Wert (*PartValue*), der Bauteile-Beschreibung (*PartHeadline*), der Bezeichnung der Library (*PartDevice*), dem „Footprint“ der Platine (*PartPackage*) und dann kommen noch besondere Attribute, die schon zuvor durch Kombination der durch Kommas getrennten *PartAttributes* erzeugt wurden. Ab hier würde das Elektor-Format im BOM-ULP schon funktionieren, aber es würden zu viele Kommas ausgegeben, wenn einige der vorgesehenen Felder leer sind. Das ist besonders dann der Fall, wenn keine User-Attribute vorhanden sind, da jede Zeile mit einem Komma abgeschlossen wird. Dies wird gefixt, indem der Code, der Bauteilfelder zu Zeilen addiert, durch den Code von **Listing 4** ersetzt wird:

```
    // add part fields and skip empty ones
    Lines[NumLines] += " = ";
    int fieldCount = 0;
    if (PartValue[i1] != "")
    {
        Lines[NumLines] += PartValue[i1];
        fieldCount++;
    }
    // this if statement is repeated for each field
    if (PartHeadline[i1] != "")
    {
        if (fieldCount > 0)
        {
            Lines[NumLines] += ", ";
        }
        Lines[NumLines] += PartHeadline[i1];
        fieldCount++;
    }
    // the rest of the fields
```

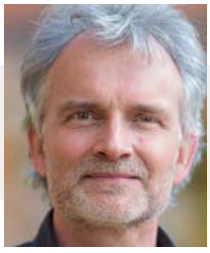
Der neue Code überspringt leere Felder und fügt nur Kommas hinzu, wenn mehr als ein Feld ausgegeben wird. Die Anzahl ausgegebener Felder steckt in fieldCount.
Mit diesen Änderungen ist die Ergänzung des BOM-ULP um das Elektor-Format abgeschlossen. Wenn man das Arduino-Beispiel-Projekt von EAGLE damit traktiert, ergibt sich die Ausgabe von **Listing 5**:

```
JP1, JP2, JP3, JP4 = PIN HEADER, PINHD-1X1, 1X01
C8, C11 = 100n, CAPACITOR, European symbol, C-EUC0603,
   C0603
RN1, RN5 = 10K, Array Chip Resistor, 4R-NCAT16, CAT16
Y1, Y2 = 16MHz, RESONATORMU, RESONATOR
Q1, Q2 = 16MHz, CRYSTAL, XTAL/S, QS
R1, R2 = 1M, RESISTOR, European symbol, R-EU_R0603,
   R0603-ROUND
RN3, RN4 = 1k, Array Chip Resistor, 4R-NCAY16, CAY16
C10 = 1u, CAPACITOR, European symbol, C-EU0603-RND,
   C0603-ROUND
RN2 = 22R, Array Chip Resistor, 4R-NCAY16, CAY16
```

Fazit

In dieser Folge wurde demonstriert, wie man eigene ULPs für EAGLE erstellt, indem man schlicht das Standard-BOM-ULP nutzt und so modifiziert, dass die Stückliste auch im Elektor-Format ausgegeben werden kann. Bei EAGLE befinden sich noch viele andere, gut gemachte ULPs im Lieferumfang. Bevor man also für eigene Vorhaben das Rad neu erfindet, empfiehlt sich ein Blick in die Dokumentation, und die Erweiterung ist oft einfacher, als man denken würde. Übrigens kann man viel über das ULP-Konzept von EAGLE lernen, wenn man das über den Elektor-Verlag erhältliche Handbuch „THE EAGLE COMPANION“ von Mitchell Duncan zu Rate zieht. ⏮

(150477)



Im ElektorBusiness-Bereich in Elektor finden Sie Artikel, News und andere Beiträge von Elektronik-Unternehmen und -Institutionen. Für Sie ausgewählt von Jan Buiting (Chefredakteur ElektorBusiness).
Beiträge nimmt er gerne entgegen unter newsdesk@elektor.com

IoT Thread Solution

Die Kombination des Thread-Stacks von Silicon Labs, der EM35xx Funk-SoC-Plattform sowie Hardware- und Software-Tools unterstützt Entwickler bei der nahtlosen Migration von ZigBee nach Thread via OTA-Upgrades (Over-The-Air). Die Hardware- und Software-Roadmap von Silicon Labs ermöglicht Multiprotokoll- und Multiband- (2,4 GHz plus Sub-GHz) Verbindungen für das IoT.

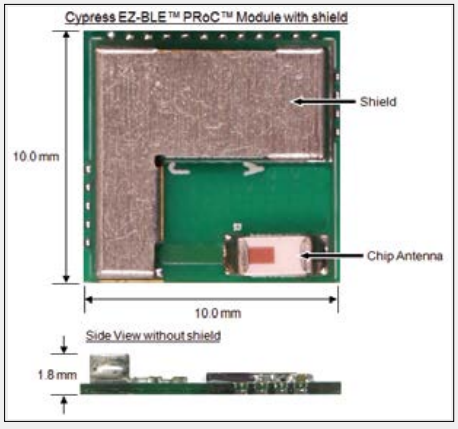
Thread bietet einen einfachen, sicheren und skalierbaren Weg zur Verbindung hunderter Home-Devices untereinander und mit dem Internet. Thread-Software ermöglicht ein selbstheilendes, IPv6-basiertes Mesh-Netzwerk, das bis auf über 250 Nodes ohne *single point of failure* skaliert. Das Protokoll unterstützt auch sogenannte „sleepy modes" und ermöglicht so den jahrelangen, energiesparenden Betrieb mit einer einzigen Batterie. Anwender können zum Netzwerk sehr einfach weitere Knoten per Smartphone oder Browser hinzufügen. Der Thread-Stack von Silicon Labs bietet eine übliche Ende-zu-Ende-Verschlüsselung der teilnehmenden Netzwerkknoten und unterstützt die Verschlüsselung aller Netzwerktransaktionen gemäß AES-128.

Silicon Labs bietet leicht verständliche Entwicklungs- und Debugging-Tools um die Einführung von Thread-kompatiblen Produkten zu beschleunigen. Der AppBuilder von Silicon Labs vereinfacht die Entwicklung von IP-basierten Netzwerkanwendungen und erlaubt Entwicklern die einfache Konfiguration von Thread-Anwendungen mit Hilfe des Application Frameworks von Silicon Labs, mit dem der Anwendungs-Code durch einfache Aufrufe und Plugins isoliert wird. Auf diese Weise wird die erstellte Software sehr portabel und kann bei vielen der unterstützten Funk-SoCs verwendet werden. Von Silicon Labs gibt es außerdem auch noch ein Desktop Network Analyzer Tool, das anders als andere Sniffer jede Aktivität im Funknetzwerk anzeigt.

www.silabs.com/thread

EZ-BLE-PRoC-Modul

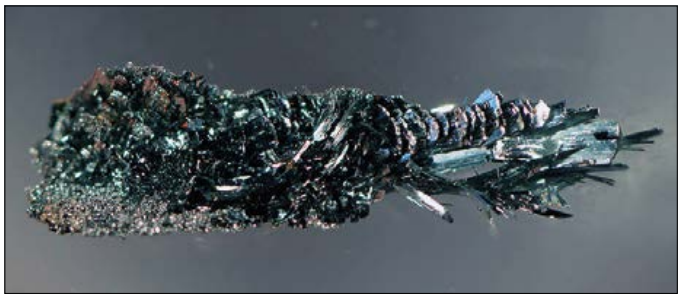
Das Modul EZ-BLE PRoC von Cypress Semiconductor integriert die Programmierbarkeit und den ARM-Cortex-M0-Core von PRoC BLE, zwei Quarze, eine Chip-Antenne, eine Metallabschirmung und passive Bauteile in einem kompakten Format von 10 x 10 x 1,8 mm. Anwender können ihre damit ausgestatteten Produkte nach dem QDID (Cypress's Qualification Design Identification) 67366 mit dem Bluetooth-Logo und einer einzigartigen Seriennummer der Bluetooth SIG versehen. Zusätzlich entspricht das Modul den Funk-Standards der USA, Kanada, Japan, Korea und Europa. Das Modul erspart Kunden durchschnittlich Kosten in der Höhe von annähernd 200.000 $ für Entwicklung, Tests und Zertifizierung.



Cypress hat den Bluetooth Low Energy Protokoll-Stack abstrahiert und eine Profil-Konfiguration in seine lizenzkostenfreie, GUI-basierte BLE-Komponente integriert, die einfach in eigenen Anwendungen verwendet werden kann, die mit Hilfe der IDE PSoC Creator von Cypress erstellt werden. Diese IDE erlaubt die komplette Systementwicklung mit einem einzigen Tool.
Anwendungsdetails der BLE-Komponente sind samt Beispielen zu den unterstützten Bluetooth-Low-Energy-Profilen sowie vielen anderen Mixed-Signal-Anwendungen in PSoC Creator enthalten. Das BLE Pioneer Development Kit gibt schon für 49 $ einen einfachen Einstieg in die Welt der Bluetooth Low Energy Devices von Cypress und entspricht in den Abmessungen dem schon bekannten PSoC4 Pioneer Kit. Das Development Kit enthält einen USB Bluetooth Low Energy Dongle der sich mit dem CySmart Master Emulation Tool pairt und so einen Windows-PC in eine Test- und Debug-Umgebung für Bluetooth Low Energy verwandelt. Das Modul EZ-BLE PRoC kann schnell und einfach mit dem EZ-BLE PRoC Module Evaluation Board evaluiert werden, das in das BLE Pioneer Development Kit eingesteckt werden kann.

www.cypress.com

Graphen bekommt Konkurrenz



Graphen ist dieses nur eine Atomlage dicke Schichtmaterial, das mit dem Nobel-Preis von 2010 über Nacht berühmt wurde. Doch jetzt gibt es Konkurrenz: Solch dünne Lagen können auch mit Phosphor erzeugt werden. Chemiker der Technischen Universität München (TUM) entwickelten nun ein Halbleiter-Material, bei dem einzelne Phosphor-Atome durch Arsen ersetzt sind. Im Rahmen einer internationalen Kooperation bauten sie daraus zusammen mit amerikanischen Kollegen erstmals Feldeffekt-Transistoren.

Seit vielen Jahrzehnten ist Silizium die Basis der modernen Elektronik. Bisher konnte die Silizium-Technik für immer kleinere Geräte immer kleinere Transistoren herstellen, doch die Größe von Silizium-Transistoren stößt langsam an ihre physikalische Grenze. Silizium ist darüber hinaus hart und spröde, doch die Konsumenten hätten gerne flexible Geräte, Geräte, die sich in Kleidung einarbeiten lassen und vieles mehr. All dies hat einen Wettlauf um neue Materialien ausgelöst, die Silizium eines Tages ersetzen könnten.

Ein solches Material könnte Arsen enthaltender schwarzer Phosphor sein. Wie das Graphen, das aus einer einzigen Lage von Kohlenstoffatomen besteht, bildet es dünnste Schichten. Die Bandbreite seiner Anwendungen reicht von Transistoren über Sensoren bis hin zu mechanisch-flexiblen Halbleiter-Bauteilen. Anders als beim Graphen, dessen elektronisches Verhalten dem von Metallen ähnelt, verhält es sich wie ein Halbleiter.

In einer Kooperation der Technischen Universität München und der Universität Regensburg auf deutscher Seite sowie der amerikanischen Universitäten University of Southern California (USC) und Yale wurden nun erstmals Feldeffekt-Transistoren aus Arsen enthaltendem schwarzem Phosphor hergestellt. Die Verbindungen synthetisierte Marianne Köpf im Labor des Fachgebiets für Synthese und Charakterisierung innovativer Materialien an der TU München. Feldeffekt-Transistoren wurden in der Gruppe um Professor Zhou und Dr. Liu gebaut und vermessen.

Die von ihr entwickelte neue Methode ermöglicht es, schwarzes Arsen-Phosphor ohne hohen Druck zu synthetisieren. Das erfordert weniger Energie und ist billiger. Über den Arsengehalt kann die Lücke zwischen Valenz- und Leitungsband präzise eingestellt werden.

Bei einem Arsengehalt von 83 Prozent hat das Material eine Bandlücke von nur noch 0,15 Elektronenvolt. Aus einem solchen Material können Sensoren aufgebaut werden, die Wellenlängen im langwelligen Infrarot detektieren. In diesem Bereich arbeiten beispielsweise LiDAR-Sensoren (unter anderem in Autos als Abstandssensoren). Eine andere Anwendung ist die Messung von Staubteilchen und Spurengasen in der Umweltmesstechnik.

www.acinnomat.ch.tum.de

Anzeige

elektorevents

Seminare, Workshops und Messen

◇ **Tagesseminar: CAN-Grundlagen**
20.10.2015
(Troisdorf)

◇ **2-tägiges Seminar: Windows programmieren in C# für Elektroniker**
03.11.2015 – 04.11.2015
(Regensburg)

◇ **Elektor auf der embedded world 2016 (Halle 4A / Stand 518)**
23.02.2015 – 25.02.2016
(Nürnberg)

Mehr Infos finden Sie unter: **www.elektor.de/events**

Als etabliertes Unternehmen im Bereich der Aerosolmesstechnik entwickeln, produzieren und vertreiben wir seit mehr als 30 Jahren weltweit Messgeräte und Prüfanlagen. Die Schwerpunkte unserer Produktpalette sind Messgeräte zur Beurteilung von Brandmeldern, Partikelfiltern und Schüttgütern.

Zur Verstärkung unseres Teams suchen wir eine/n

Programmierer-/in

Ihre Aufgaben
- Anpassung und Erweiterung von Echtzeitprogrammen auf Mikrocontrollern
- Programmierung von Auswertesoftware für PCs

Wir erwarten
- Kenntnisse in Echtzeitprogrammierung, z.B. PEARL
- Kenntnisse in Regelungstechnik und dessen softwaretechnischer Umsetzung
- Wünschenswert sind Kenntnisse in C und LabView

Wir bieten Ihnen
- Eine Position mit Selbstständigkeit und Herausforderung
- Ein dynamisches Arbeitsumfeld mit Teamgeist und Eigenverantwortung

Interessiert?
Dann freuen wir uns auf Ihre Rückmeldung

LORENZ Meßgerätebau

Lorenz Messgerätebau GmbH & Co. KG
Im Siechenanger 5
37191 Katlenburg-Lindau
Tel. 05556-995597-0, Fax 995597-9
bewerbung@lorenz-messgeraetebau.de

Mehr Tempo bei Embedded-Systemen mit USB 3.0

Von **Abhishek Gupta** (Cypress Semiconductor)

HD- und UHD-Videokameras (Ultra High Definition) profitieren von den Vorteilen einer standardisierten seriellen Verbindung wie USB 3.0. Moderne Exemplare generieren nämlich ein hohes Aufkommen an Echtzeit-Daten, die lokal erfasst und vorverarbeitet, aber dann mit hohem Tempo an weitere Geräte wie Überwachungsmonitore und Speichersysteme übertragen werden müssen. Schon eine 5-Megapixel-Kamera kann bei 24 fps eine Datenrate von 2,4 Gbit/s generieren. Geräte mit Schnittstellen wie USB 2.0 oder WLAN nach 802.11n sind schlicht nicht dazu in der Lage, mit solchen Datenmengen umzugehen. Zu langsame Datenpfade können außerdem auch an anderer Stelle zu erhöhten Kosten bei Embedded-Systemen führen, da dann größere Frame-Buffer und größere lokale Speicher erforderlich werden. Dies kann die Gesamtkosten und den Entwicklungsaufwand deutlich erhöhen. USB 3.0 ist für moderne Hochleistungs-Systeme die Schnittstelle der Wahl. Seine Architektur erlaubt die Kommunikation mit älteren „legacy devices“ und verfügt mit SuperSpeed zudem über Datenraten von bis zu 5 Gbit/s. Außerdem kann eine USB-3.0-Schnittstelle gegenüber dem Vorgänger USB 2.0 den dreifachen Strom bieten. Wegen der breiten industriellen Akzeptanz ist USB 3.0 die richtige Wahl für interne und externe Datenpfade, denn hier kommen hohe Leistung, geringer Stromverbrauch und niedrige Kosten zusammen.

SoCs für Kameras sind sehr spezifisch an die Datenerfassung und deren Vorverarbeitung bei hochauflösenden Bildsensoren angepasst. Sie nutzen dabei On-Chip-DSPs für die Bild- und Videoverarbeitung. Die grundlegende Auslegung zeigt **Bild 1**: Ein SDXC-Interface dient zur lokalen Speicherung der Video- bzw. Bilddaten auf handelsüblichen SD-Karten in Mini- oder Mikro-Ausführung. Ein HDMI-Port erlaubt den direkten Video-Transfer an externe Peripherie wie Monitore. Der SoC muss sich gegebenenfalls auch noch um das LCD des Systems kümmern. USB 2.0 und WLAN dienen zur Übertragung der Daten an weitere externe Geräte. Leider kommen diese SoC-Interfaces mit den anfallenden Datenraten nicht mit. Eigene SoC-Entwicklungen führen aber aufgrund der hohen Komplexität solcher SoCs zu langen Entwicklungszeiten.

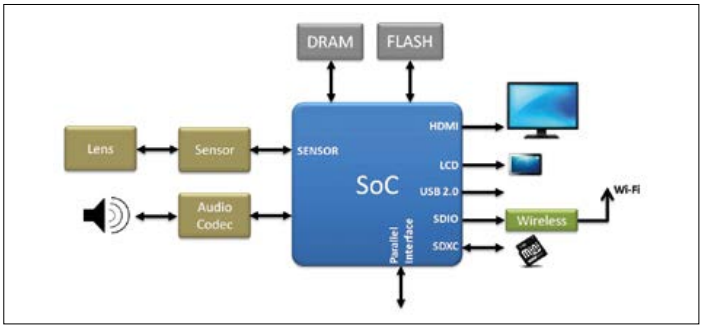


Bild 1. Prinzipaufbau der Elektronik einer Videokamera.

Tabelle 1. Dauer der Datenübertragung für ein 30 Minuten dauerndes HD-Video in 1080p (H.264).

Interface	Transfer Rate	Transfer Time
Wi-Fi 802.11n Optimized	300 Mbit/s	8 Minuten
Wi-Fi 802.11n Typical	80 Mbit/s	30 Minuten
Wi-Fi 802.11g	54 Mbit/s	45 Minuten
Wi-Fi 802.11b	11 Mbit/s	3:39 Minuten
USB 2.0 Optimized	320 Mbit/s	7:30 Minuten
USB 2.0 Typical	200 Mbit/s	12 Minuten

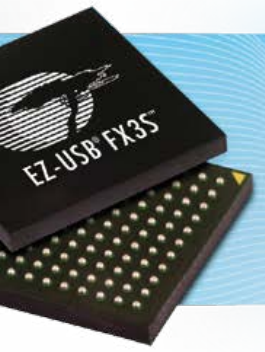
Bandbreite

Produkte wie kommerzielle Überwachungskameras verfügen nur über einen begrenzten internen Speicher. Sie verlassen sich stattdessen auf schnelle Datenverbindungen über verkabelte Netzwerke oder WLAN, um die Video-Streams zuverlässig auf einem lokalen Server abzulegen. Im Consumer-Bereich sind Kameras hingegen häufiger mit SD-Karten-Slots ausgestattet. Solche Kameras sind aber üblicherweise nicht mit einem Bildschirm ausgestattet, mit dem sich die aufgezeichneten Daten betrachten ließen. Das Material wird hier an externe Geräte nicht nur zu Sicherung, sondern auch zur Inspektion übertragen. Diese externen Geräte sind in der Regel hierfür mit WLAN, USB oder Bluetooth ausgestattet.

Die WLAN-Verbindung hängt stark von der Qualität eines schnellen Funknetzwerks ab. Selbst ein gutes WLAN nach 802.11n unterstützt maximal eine Datenrate von 300 Mbit/s.

USB 2.0 kommt auf einen effektiven Durchsatz von 280...320 Mbit/s (= 35...40 MB/s). Tatsächlich aber bieten die meisten Geräte lediglich 200...240 Mbit/s. Der dauerhafte USB-2.0-Durchsatz hängt stark von den Treibern und von der jeweiligen Plattform ab. **Tabelle 1** zeigt die Übertragungsdauern bei verschiedenen Übertragungsraten für ein 30 Minuten langes HD-Video (1080p in H.264), was sich auf eine Datenmenge von rund 18 GB summiert. In einigen Fällen dauert es länger das Video zu streamen als es zu betrachten! Kamera-Hersteller begegnen diesem Problem mit zwei Methoden: Entwicklung eines eigenen ASIC der nächsten Generation oder warten, bis bessere ASICs verfügbar sind.

Beide Varianten können zu enormen Entwicklungsverzögerungen von bis zu zwei Jahren führen. Die bessere Alternative ist es also, auf die schon vorhandene USB-3.0-Technik zu setzen. SuperSpeed bietet gegenüber USB 2.0 einen um den Faktor 10 höheren Durchsatz, womit sich diese Technik ideal zur Datenübertragung von Videodaten eignet. Fertige SuperSpeed-Controller wie der EZ-USB® FX3STM [1] von Cypress enthalten neben den USB-3.0-Funktionen auch einen ARM9-Core und bieten sogar gleich zwei Speicherschnittstellen (konfigurier-



Als serielle Hochgeschwindigkeitsverbindung hat sich USB 3.0 (SuperSpeed USB) mittlerweile bei PCs und Peripherie weithin durchgesetzt. Auch bei der Entwicklung von Embedded-Systemen stoßen die hohen Datenraten von USB 3.0 auf zunehmendes Interesse. Man kann damit preiswerte und doch schnelle Verbindungen zu anderen Geräten aufbauen sowie dank hoher Integration dennoch den Energiebedarf, den Bauteile- und den Entwicklungsaufwand in Grenzen halten.

bar als SDIO 3.0 oder eMMC 4.41), wie in **Bild 2** zu sehen ist.

Einsatz des FX3STM

Ein FX3S-Chip kann sehr leicht in eine Videokamera integriert werden. Man koppelt einfach dank des programmierbaren GPIF-Interfaces (**Bild 3**) den vorhandenen SoC mit dem FX3S. Das GPIF-II-Interface des FX3S ist ein frei konfigurierbares paralleles Interface zum Anschluss externer ASICs, SoCs oder FPGAs. In dieser Konfiguration verhält sich ein FX3S samt SD-Karten fast wie ein USB-Speicherstick für das SoC. Die komprimierten Videodaten können dann wie bisher abgelegt werden, ohne dass Änderungen an der zugehörigen Software nötig wären. Diese Implementierung bietet etliche Architekturvorteile. Zunächst verfügt man über eine risikolose Methode der Integration einer USB-3.0-Schnittstelle. Außerdem entlastet diese Lösung das SoC vom Management von SD-Karten, sodass mehr Rechenleistung für andere Aufgaben zur Verfügung steht. Die GPIF-II-Schnittstelle ist dabei als gemultiplexter 16 bit breiter Adress/Daten-Bus mit einem Takt von 100 MHz konfiguriert. Die erlaubt die 1,8-fache Schreibrate (Bild 3, Pfeil 1) relativ zu gängigen Implementierungen. Das doppelte SD-Interface des FX3S kann bei zwei Karten sogar als RAID 0 konfiguriert werden. Dies steigert nicht nur den Durchsatz, sondern verdoppelt auch die Speicherkapazität. Die Datenrate beim Lesen (Bild 2, Pfeil 2) erreicht bis zu 720 Mbit/s. Ein 18 GB großes Video ist damit in weniger als vier Minuten übertragen. Auch das ist ein deutlicher Performance-Gewinn. Dank der höheren Speicherkapazität kann zudem länger oder mit höherer Auflösung aufgezeichnet werden, denn die Zukunft gehört 4K-Videos. Von Vorteil ist, dass der zusätzliche Controller die Betriebsdauer nicht wesentlich beeinflusst. Ein FX3S benötigt lediglich etwa 97 mW zusätzlich. Bei einer Akku-Kapazität von typischen 1.160 mAH (bei 3,7 V) benötigt ein FX3S zusätzlich nur etwa 14 mAH (95 % Wirkungsgrad des Schaltreglers) bei einem 30 Minuten dauernden Video. Der Akku wird also nur mit rund 1,2 % seiner Kapazität zusätzlich belastet. Der Betrieb mit einer zweiten SD-Karte hoher Kapazität wird bei einem Leistungsbedarf von etwa 270 mW den Akku um weitere 3,3 % seiner Kapazität belasten.

Fazit

USB 3.0 ist die passende Alternative für Embedded-Systeme, die schnelle Datenübertragungen benötigen. Entwickler können natürlich auf die SoCs der nächsten Generation warten oder das Risiko eingehen und solche SoCs selbst entwickeln. Einfacher, schneller und schon verfügbar aber sind spezialisierte Controller wie der FX3S von Cypress, der den Datentransfer (ohne großen Aufwand und ohne die Akkulaufzeit bedeutend einzuschränken) um den Faktor 2 bis 10 gegenüber aktueller Technik beschleunigen kann. ⏮

(150466)

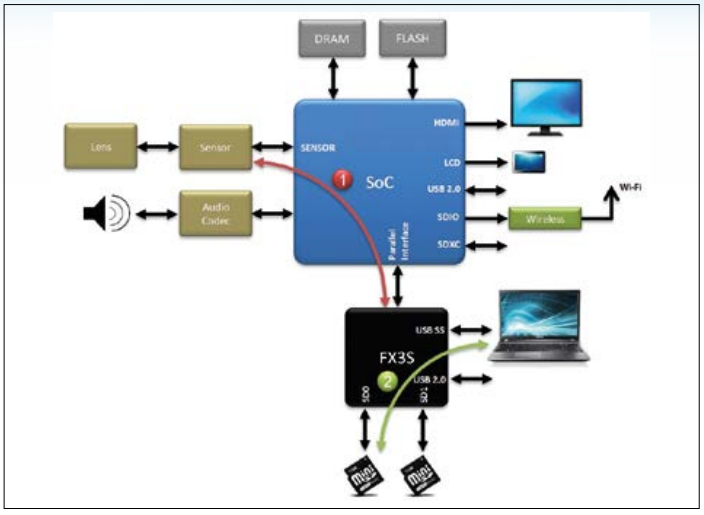


Bild 2. Blockschaltung des FX3S SuperSpeed-USB-3.0-Controllers von Cypress.

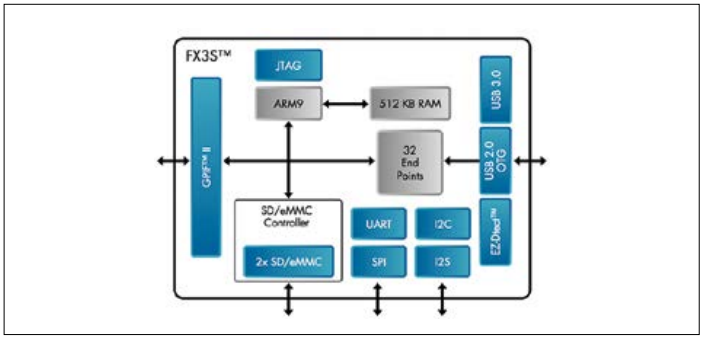


Bild 3. Optimierte Architektur durch Hinzufügung eines FX3S.

Weblink

[1] FX3STM SuperSpeed USB Controller: www.cypress.com/?docID=45991

Über den Autor

Abhishek Gupta ist Business Analyst bei Cypress Semiconductor. Er hat einen Abschluss als Ingenieur in Elektronik & Kommunikation der Maharishi Dayanand University in Haryana (Indien). Zuvor war er bei Agilent Technologies als Logistik-Koordinator (RoHS-Spezialist) tätig. Seine Mail-Adresse ist agni@cypress.com.

Willkommen im Elektor-Labor

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

Unser Angebot: Werden Sie berühmt!

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

Unsere Geschichte

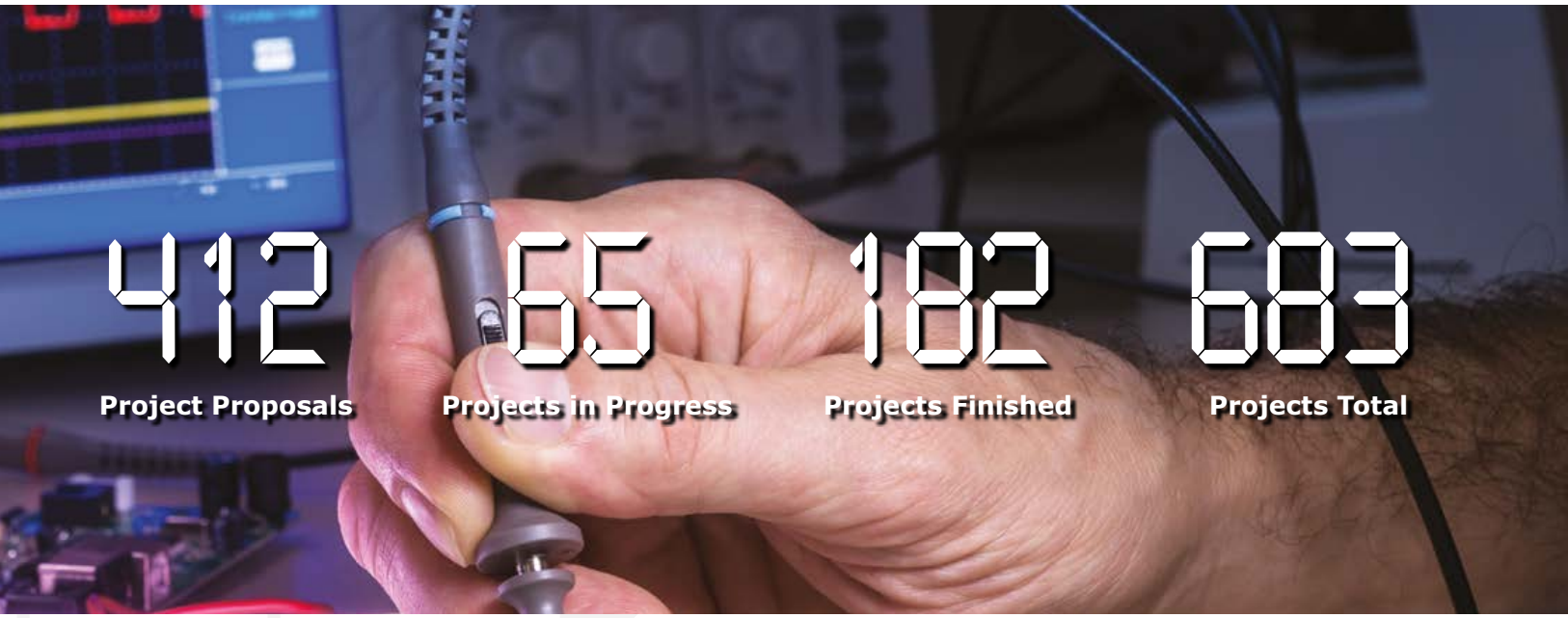
Die Ursprünge des Elektor-Labors gehen zurück auf die frühen 1970er-Jahre. Löten und Schreiben war damals noch ein Ein-Mann-Schreibtisch-Job. Über die Jahre waren Mitarbeiter des Labors nicht nur Zeugen der Ankunft der Transistoren, der ICs, der Mikrocontroller und der SMDs, sondern sie setzten diese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

Unsere Einrichtungen

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

Unsere Produkte

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



Unser Maßstab

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

Unsere Webinare

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

Unsere Experten und Entwickler

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

elektor labs

Sharing Electronics Projects

Search

Home Proposals ➡ In Progress ➡ Finished

Log in

Laden Sie Ihre eigenen Projekte hoch!

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

Wer kann, wer darf?

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

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

Willkommen bei DESIGN

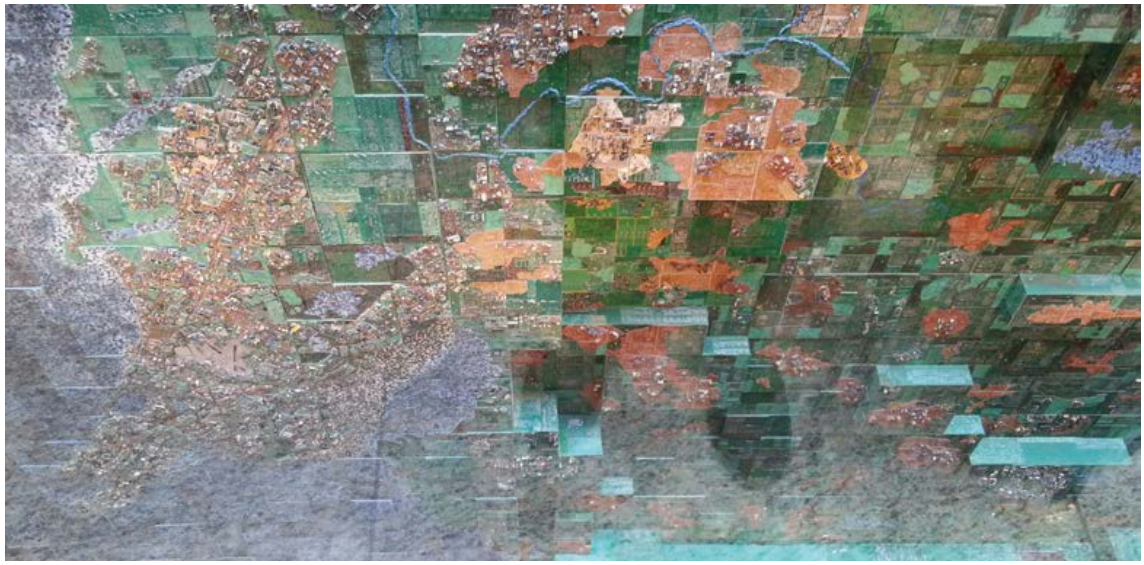
Von **Clemens Valens,** Elektor Labor

Strebe immer nach dem Besten!

Neulich bin ich von Paris nach Mumbai geflogen. Kurz vor dem Boarding in Paris Charles de Gaulle (CDG) sah ich ein Schild an der Tür: CDG hatte den Titel des „World's Most Improved Airport 2015" in einem Wettbewerb errungen. Seltsam, dachte ich. Zunächst einmal, weil ich mehrmals im Jahr von/nach CDG fliege und ich Verbesserungen überhaupt nicht bemerkt habe. Auch brauche ich immer noch eine Stunde vom Eingang zum Gate und die Flüge sind überdies noch verspätet. Ich war wirklich sehr überrascht, nicht, weil vor dem Gate statt eines Flugzeugs ein Stadtbus mit dem Schild „Mumbai" stand, sondern vor allem von dem Streben, oder genauer gesagt, dem Mangel daran, den dieser Titel offensichtlich ausstrahlt.
Stellen Sie sich ein Produkt vor, ein Computer-Betriebssystem beispielsweise, das wirklich schrecklich ist, aber das Sie trotzdem irgendwie vermarkten müssen. Dann können Sie es irgendwie ändern und es als „am meisten verbessertes Produkt der Welt" anpreisen. Bedeutet dies, dass das Produkt nun ein gutes Produkt ist? Nein, es kann immer noch lausig sein, aber vielleicht ein bisschen weniger lausig als zuvor.
Die Auszeichnung ist völlig frei von jeder Bedeutung! Ein gutes Produkt muss immer nach dem Titel „Bestes Produkt der Welt" streben. Nur, wenn Sie nach dem Besten streben, können Sie ein Qualitätsprodukt verkaufen. Wenn nur mehr Unternehmen dies beherzigen würden, wäre die Welt ein besserer Ort.
P. S. Habe ich schon erwähnt, dass dieser „2015 World Airport Award" in Paris vergeben wurde? Ein Schelm, der Böses dabei denkt...

www.worldairportawards.com

Die Kunst der Elektronik



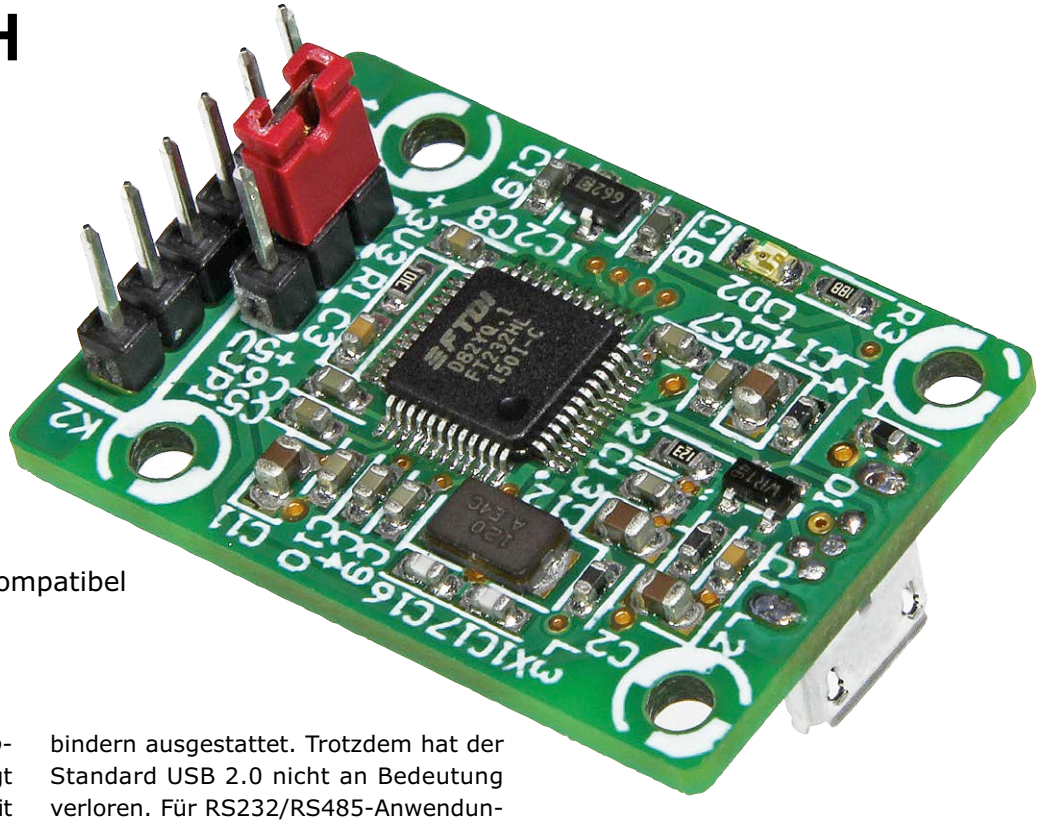
Der Mumbai Chhatrapati Shivaji International Airport (BOM) wurde 2015 zum „drittverbesserten" Flughafen der Welt gewählt. Ich war dort und kann der Wahl dieses neu erbauten Flughafens nur beipflichten. Eigentlich wäre auch eine höhere oder sogar die erste Position für diesen schönen und modernen Airport gerechtfertigt, wo sogar Flüssigkeiten kein Problem in den Sicherheitskontrollen darstellen.
Bei einem Spaziergang durch die ruhigen, mit Kunstwerken aus ganz Indien dekorierten Flure entdeckte ich ein riesiges Mosaik, das eine Karte von Mumbai und dessen Hinterland darstellte. Bei näherer Betrachtung stellte ich fest, dass es ausschließlich aus Leiterplatten, elektronischen Bauteilen und Drähten bestand. Stellen Sie sich ein Meer aus Aluminiumkondensatoren, Flüsse aus Draht und Felder aus bestücktem FR4 vor! Wenn Sie jemals dorthin gelangen, werfen Sie einen Blick darauf, es befindet sich am Gate 85/86. Das Foto ist wegen des Panzerglases ein wenig verschwommen. ⏮

(150476)

Seriell nach USB

Mit dem FT232H

Von **Ton Giesberts** (Elektor-Labor) und **Thijs Beckers** (Redaktion NL)



Der altgewohnte serielle Port ist noch immer eine viel genutzte Schnittstelle zwischen PC und Peripherie. Hier stellen wir eine kompakte USB/Seriell-Konverter-Implementierung mit dem FT232H von FTDI vor, die kompatibel mit USB 2.0 (high-speed) ist!

Nach dem *Seriellen USB-nach-Multi-Protokoll-Konverter* vom April 2015 [1] folgt hier eine kompakte Version, ebenfalls mit dem FT232H von FTDI, die als Modul für die Montage an einer Frontplatte oder Rückwand eines Gerätegehäuses konzipiert ist. Die USB-Buchse hat ihren Platz auf der Platinenunterseite. Der Einsatz des Moduls bietet sich überall dort an, wo ein Projekt unkompliziert durch einen USB-Anschluss ergänzt werden soll. Bei der Entwicklung lag das Hauptaugenmerk auf minimierten Abmessungen, die kleine Platine misst nur 22 · 33 mm.

Full Speed oder High Speed?

Zwar ist USB 3.0 schon vor einiger Zeit erschienen, und einige Geräte neuester Generation sind bereits mit dem Nachfolger USB 3.1 und USB-Typ-C-Steckverbindern ausgestattet. Trotzdem hat der Standard USB 2.0 nicht an Bedeutung verloren. Für RS232/RS485-Anwendun-

Eigenschaften

- USB 2.0 High-Speed
- (Pin-)Kompatibel mit FTDI-USB/TTL-UART-Kabel
- Abgewinkelte USB-Buchse für platzsparende Montage
- Betriebsspannung am Ausgang wahlweise 3,3 V oder 5 V
- Wird nach Installation der VCP-Treiber [3] als COM-Port erkannt

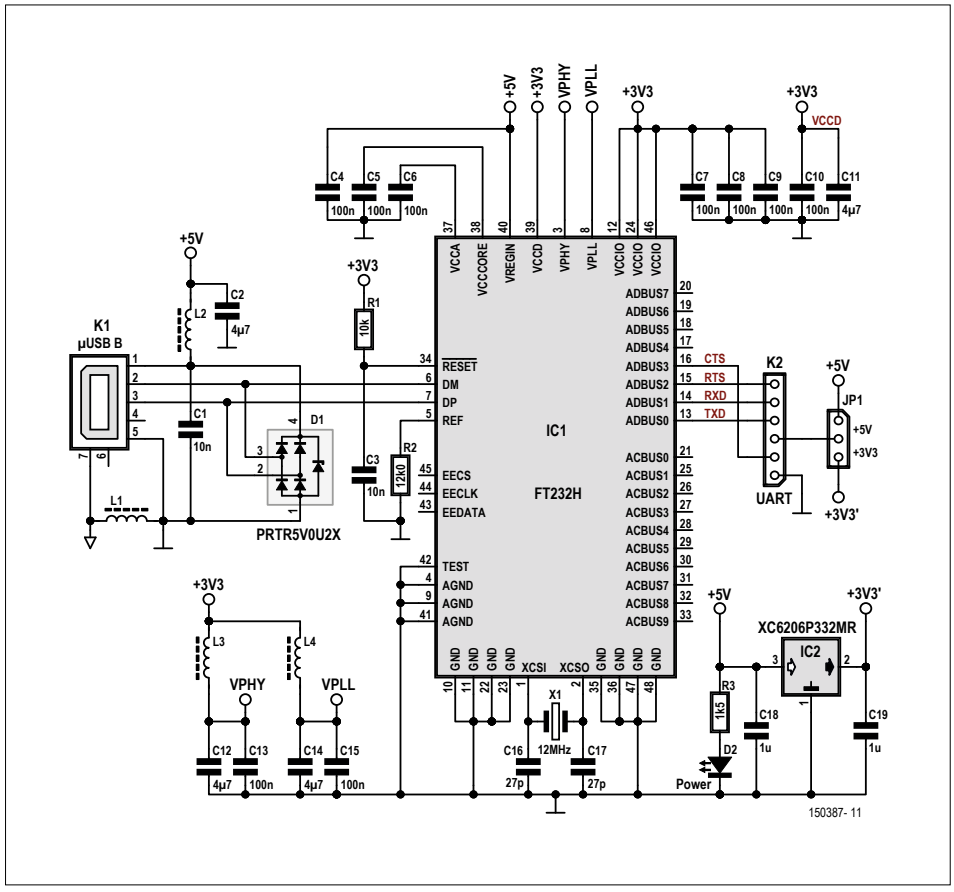


Bild 1. Die Schaltung lehnt sich an den USB-nach-Multi-Protokoll-Konverter vom April 2015 an.

MPSSE

Die *Multi-Protocol Synchronous Serial Engine* (MPSSE) ist ein flexibles Interface zwischen synchronen seriellen Schnittstellen und USB-Ports. Synchrone Verbindungen erfordern physisch drei Leitungen: Daten, Takt und Masse. Der Takt wird meistens vom *Master* generiert, die übrigen Geräte am Bus synchronisieren ihren Takt mit diesem Signal. Als Beispiele seien der SPI-, der I²C- und der JTAG-Bus genannt. Der UART, mit dem der serielle Port eines Mikrocontrollers arbeitet, ist ein Beispiel für eine asynchrone Schnittstelle. Hier sind das Datensignal und das Taktsignal miteinander kombiniert, das Taktsignal muss aus dem kombinierten Signal rekonstruiert werden. Die MPSSE unterstützt die Kommunikation mit mehreren Kategorien synchron arbeitender Schnittstellen. Das Datenformat und die Taktsynchronisation lassen sich unterschiedlich konfigurieren, die Übertragungsgeschwindigkeit kann bis zu 30 MB/s betragen. Auch der parallele Datentransport ist möglich, beispielsweise im Modus FT1248. Bei dieser Art der Datenübertragung im Halb-Duplex-Betrieb stehen Bandbreite und Übertragungsgeschwindigkeit in gegenseitiger Abhängigkeit. Auch synchrone und asynchrone FIFO-Betriebsarten wie zum Beispiel FT245 liegen im Bereich des Machbaren. Das so genannte „Bitbanging“ ist ebenfalls eingeschlossen.

Stückliste

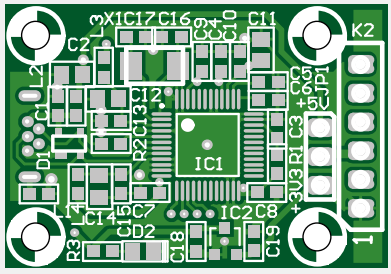
Widerstände: (0,1 W, 1 %, SMD 0603)
R1 = 10 k
R2 = 12 k
R3 = 1k5

Kondensatoren: (SMD 0603, sofern nicht anders angegeben)
C1,C3 = 10 n, 50 V, 10 %, X7R
C2,C11,C12,C14 = 4µ7, 10 V, 10 %, X7R, SMD 0805
C4...C10,C13,C15 = 100 n, 50 V, 10 %, X7R
C16,C17 = 27 p, 50 V, 5 %, C0G/NP0
C18,C19 = 1 µ, 6,3 V, 20 %, X5R

Induktivitäten:
L1...L4 = 600 Ω @ 100 MHz, 25 %, 0,15 Ω, 1A3, SMD 0603

Halbleiter:
D1 = PRTR5V0U2X (SMD SOT-143B)
D2 = LED grün (SMD 0805)
IC1 = FT232HL (SMD LQFP-48)
IC2 = XC6206P332MR (SMD SOT-23)

Außerdem:
K1 = USB-2.0-Micro-Buchse Typ B, Through-hole, stehend
K2 = SIL-Buchsenleiste 6-polig, RM 2,54 mm
JP1 = Stiftleiste 3-polig mit Jumper, RM 2,54 mm
X1 = Quarz 12 MHz, 18 pF, SMD, 5 mm · 3,2 mm
Platine 150387-



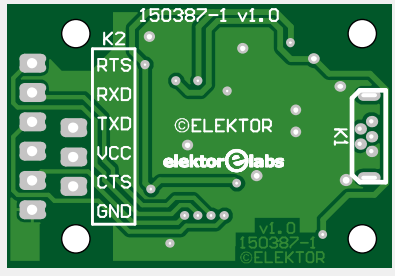
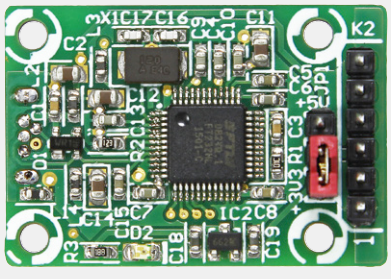
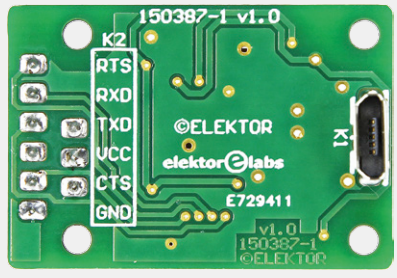


Bild 2. Auf der minimierten Platine wird fast jeder Quadratmillimeter genutzt. (hier afgebeeld op 150% van de ware grootte).





gen reicht die Geschwindigkeit absolut aus. Der Wandler-Baustein FT232**H** (Suffix „H“) erfüllt den Standard USB 2.0 *High Speed* (480 Mbit/s), während der FT232**R** (Suffix „R“) lediglich dem Standard USB 2.0 *Full Speed* (12 Mbit/s) genügt. Die Funktionserweiterungen der H-Version sind in der *Multi-Protocol Synchronous Serial Engine (MPSSE)* festgeschrieben, nebenstehend erklären wir, was es mit **MPSSE** auf sich hat.

> Gewoon, omdat het high-speed is

Die Funktionserweiterungen bleiben bei diesem Projekt außen vor. Falls sie für einen vorgesehenen Einsatzzweck unverzichtbar sind, ist der oben erwähnte *Serielle USB-nach-Multi-Protokoll-Konverter* eine geeignete Lösung. Übrigens ist die Baustein-Version FT232H nicht nur schneller als die Version FT232R, sie ist zurzeit sogar preiswerter.

Schaltungsneuauflage

Beim Betrachten der Schaltung in **Bild 1** fallen naturgemäß weitreichende Ähnlichkeiten mit dem vorangegangenen Projekt auf. Spannungsregler IC2, unten rechts im Bild, stellt die Betriebsspannung 3,3 V für den FT232H bereit. Die mit „D1“ bezeichneten Dioden bilden einen integrierten Überspannungsschutz, er schützt die Eingänge des FT232H vor elektrostatischen Entladungen bis 8 kV. Am ~~RESET~~-Eingang liegt über R1 die Spannung 3,3 V, wie im Datenblatt vorgeschrieben. Ebenfalls vorgeschrieben ist der Widerstand vor dem Stromreferenz-Eingang REF, er ist über R2 (12 kΩ, 1 %) mit Masse verbunden. Obligatorisch ist auch Quarz X1 (12 MHz, ±30 ppm) mit den zugehörigen Kondensatoren C16 und C17. Die Induktivitäten L1 und L2 mindern gemeinsam mit C1 das hochfrequente Rauschen, das der Betriebsspannung und der Abschirmung überlagert ist. LED D2 ist die Einschaltkontrolle, sie leuchtet während des Betriebs auf.

Die USB-2.0-Micro-Buchse K1 wird auf der Unterseite der Platine montiert. Die seriellen UART-Signale gelangen zum FT232H über die 6-polige Buchsenleiste

Tabelle 1. Anschlussbelegung K2 (FTDI-Kabel).

Pin	Funktion	Farbe
1	GND	Schwarz
2	CTS (AD3)	Braun
3	VCC	Rot
4	TX (AD0)	Orange
5	RX (AD1)	Gelb
6	RTS (AD2)	Grün

K2. Die Kontaktbelegung von K2 ist mit dem Elektor-FTDI-Kabel [2] kompatibel, sie ist in **Tabelle 1** aufgelistet. Mit Jumper JP1 kann die Betriebsspannung, die an Pin 3 von K2 verfügbar ist, wahlweise an 3,3 V oder 5 V gelegt werden (siehe **Tabelle 2**). Ebenso wie beim vorangegangenen Projekt liegt Spannungsregler IC2 unmittelbar an der gemeinsamen 3,3-V-Leitung, die 3,3-V-Leitung des FT232H wird entlastet.

Tabelle 2. Jumper JP1.

Pin	Funktion
1-2	K2: Pin 3 liegt an 5 V
2-3	K2: Pin 3 liegt an 3,3 V

Das EEPROM, das dabei half, den *Seriellen USB-nach-Multi-Protokoll-Konverter* zu konfigurieren, wurde hier eingespart. Der FT232H kann deshalb nicht über *FTProg* konfiguriert werden, er arbeitet ausschließlich im Default-Modus als UART.

Praktisches

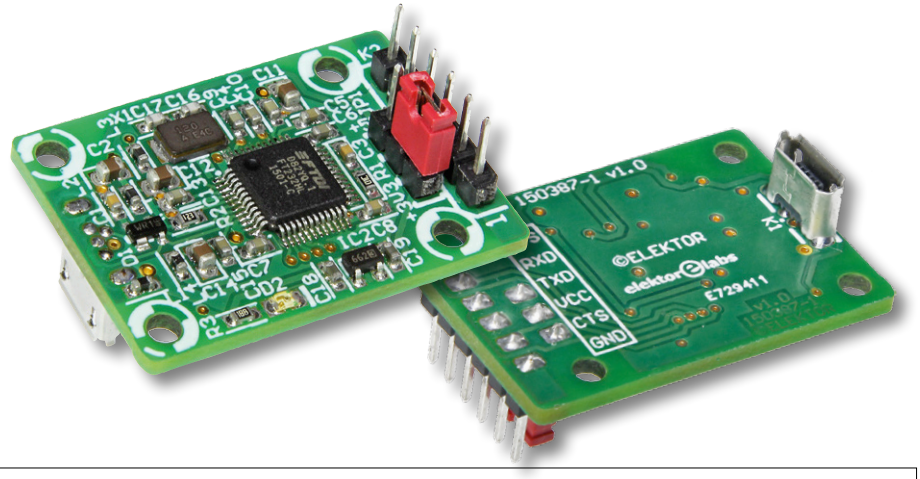
Die Mini-Platine (**Bild 2**) lässt sich mit vier M2-Schrauben am Einbauort befestigen. Da die USB-Buchse auf der Unterseite der Platine senkrecht steht, ist die Montage auch an vergleichsweise starken Frontplatten oder Gehäuserückwänden möglich.

Beim Verbinden mit einem PC wird der FT232H als UART erkannt, vorausgesetzt, dass zuvor die VCP-Treiber (*Virtual COM Port*, [3]) installiert wurden. In **Bild 3** ist wiedergegeben, wie dies unter Windows 7 auf dem Bildschirm aussieht. Falls bei der Installation Probleme auftauchen, bietet das Internet vielfältige Hilfe an. Zu den thematisch informativen Websites gehört beispielsweise die Website von Adafruit [4]. ⏮

(150387)gd

Cypress CY7C65211

FTDI ist nicht der einzige Hersteller, der sich mit Bausteinen für serielle Schnittstellen beschäftigt. Ein Produkt von Cypress ist der CY7C65211, Cypress bezeichnet ihn als *USB-Serial-Single-Channel (UART/I²C/SPI) Bridge with CapSense® and BCD*. Es spricht vieles dafür, diesen Chip in eigene Entwicklungen einzubeziehen: Das integrierte EEPROM erübrigt einen externen Datenspeicher, über eine API lassen sich die Bus-Betriebsarten umschalten (bei FTDI nur mit dem Konfigurationstool), unterstützt werden serielle Ports mit Spannungen im Bereich 2,2...5 V. Dass hier trotzdem der FT232H von FTDI zum Einsatz kommt, liegt an der Eignung für USB 2.0 High-Speed. Hinzu kommt, dass der FT232H im Lieferzustand als UART konfiguriert ist, er ist deshalb sofort einsetzbar.



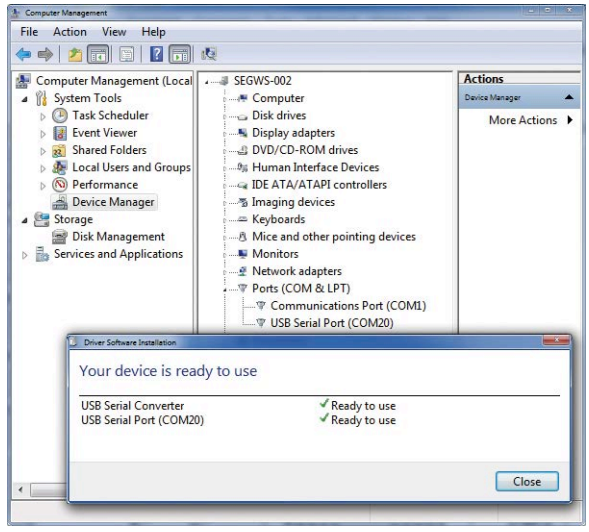


Bild 3. Nach der Installation der VCP-Treiber wird der FT232H als COM-Port erkannt.

Weblinks

[1] Serieller USB-nach-Multi-Protokoll-Konverter, Elektor April 2015: www.elektormagazine.de/130542.

[2] FTDI-Kabel: Elektor-Store 080213-71 (5-V-Version), 080213-72 (3,3-V-Version), siehe www.elektor.de.

[3] Treiber-Download bei FTDI: www.ftdichip.com.

[4] Hilfe zur Installation bei Adafruit: https://learn.adafruit.com/adafruit-ft232h-breakout.

Android I/O-Board (3)

Konzept und Funktionalität des Datenloggers

Von **Elbert Jan van Veldhuizen** (NL)

In der dritten Folge beschreiben wir, wie das Android I/O-Board als multifunktionaler Datenlogger eingesetzt werden kann. Die Daten werden in einem seriellen, 32 KB großen Flash-Speicher auf dem Board gespeichert. Mit der Datenlogger-App, die auf dem Android-Gerät läuft, ist die Bedienung übersichtlich und unkompliziert. Weitere Themen sind diesmal der Bootloader und die Firmware.

Der nicht flüchtige Flash-Speicher befähigt das Android I/O-Board, Messdaten unabhängig von anderen Systemen zu speichern. Wenn der Logger die Daten von beispielsweise zwei analogen Kanälen einmal in der Minute liest, reicht die Speicherkapazität für mehr als eine Woche aus.

In einem anderen Modus werden die Messdaten statistisch ausgewertet, gespeichert werden die Ergebnisse. Angenommen die Messungen folgen zeitlich im Abstand von 0,9 s. Aus 4096 Messwerten werden einmal in der Stunde der Mittelwert, die Standardabweichung, das Minimum und das Maximum ermittelt und im Speicher abgelegt. Falls es sich um Temperaturen handelt, genügen in der Regel diese Informationen als Kriterien für Entscheidungen, die vom Temperaturverlauf abhängen. Die Daten eines analogen Kanal sind dann über mehr als neun Monate erfassbar.

Der Datenlogger ist in der Lage, die Daten von acht Kanälen gleichzeitig zu protokollieren, generiert zum Beispiel von vier ADCs, dem Counter, einem CPS und zwei digitalen Eingängen. Für die Datenausgabe können alle Einstellungen vorgenommen werden, die im ersten Teil des Beitrags beschrieben sind. Zum Beispiel ist das Rauschen des ADC-Signals reduzierbar, indem ein Wert aus mehreren Messungen gebildet wird, oder die Referenzspannung wird in die Messungen einbezogen.

Die Firmware kann das System anhalten, wenn der Speicher belegt ist, oder der Speicherinhalt wird zyklisch überschrieben. Der Datenlogger kann nach einem Stromausfall den Betrieb an der Abbruchstelle nahtlos fortsetzen.

Gesteuert wird der Datenlogger über die Statusregister S0, S1 und S3.

Ein Eingangssignal wird protokolliert, wenn im zugehörigen Statusregister 1 das Bit 7 gesetzt ist. Wie bereits beschrieben, sind auch andere Einstellungen möglich. Um beispielsweise Eingang B2 als ADC mit der Referenzspannung 2,048 V zu loggen, muss zuvor das Kommando *s b2 5#s b21 129* gegeben werden. Das Loggen startet, sobald der Wert 1 in das Statusregister S3 geschrieben wird. In diesem Fall wird Bit 0 des Statusregisters S0 gesetzt. Um den Vorgang zu stoppen, kann in S3 der Wert 0 gesetzt werden. Bit 0 von S0 wird dann zurückgesetzt. Der Datenlogger setzt seine Arbeit fort, wenn entweder der Wert 2 oder der Wert 3 nach S3 geschrieben werden. Beim Wert 2 schließen die nächsten Daten im aktuellen 64-Byte-Block unmittelbar an. Nach einem Mikrocontroller-Reset ist dies stets der Speicheranfang. Beim Wert 3 in Statusregister S3 beginnt der Datenlogger mit einem leeren 64-Byte-Block.

Über die Bits 5...7 ist die Periode einstellbar, mit der die Daten gespeichert werden. Die Speicherperiode kann 10 ms, 100 ms, 1 s, 1 min, 15 min, 1 h oder 1 d lang sein. Die Bits 3 und 4 legen fest, wie viele Messungen durchgeführt werden, es können 1, 64, 4096 oder 4194304 sein. Wenn Bit 7 des Statusregisters S1 gesetzt ist, blinkt die LED an B0 bei jeder Messung.

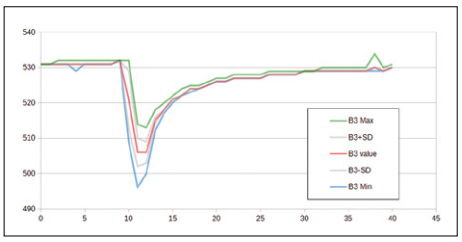
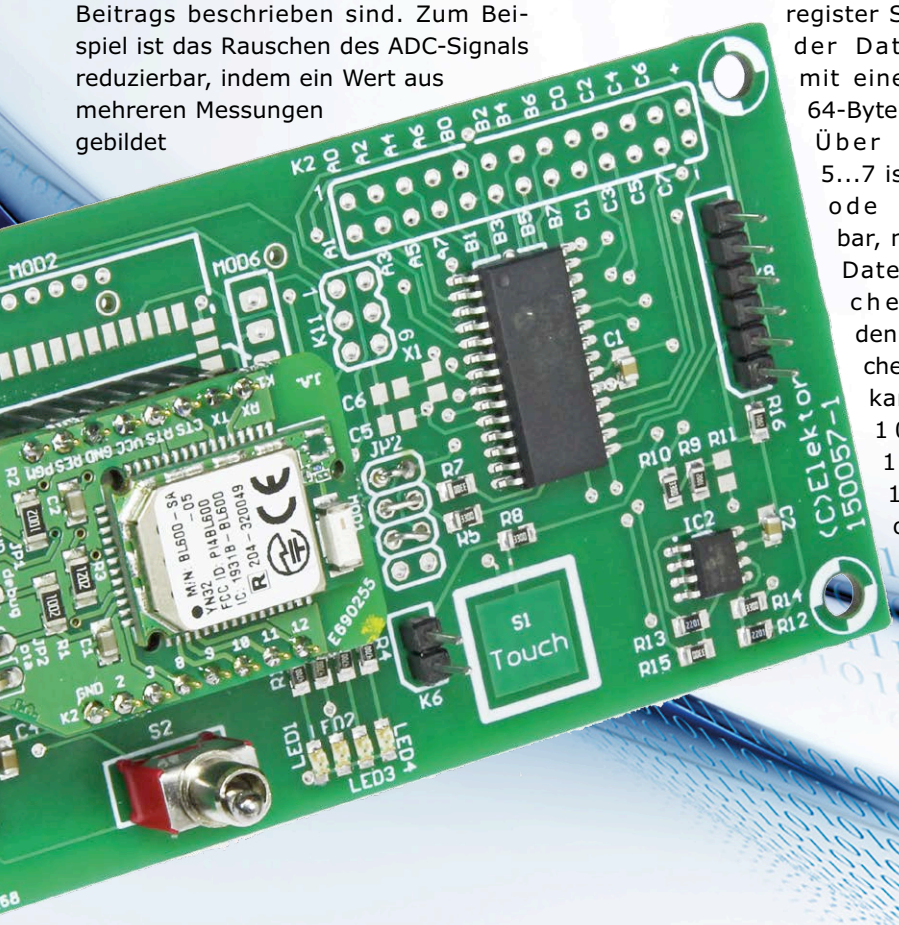


Bild 1. Grafische Darstellung errechneter statistischer Werte.

Speichern kann der Datenlogger den kumulierten Wert (Summe der Messwerte) oder den arithmetischen Mittelwert (Summe der Messwerte dividiert durch Anzahl der Messungen). Außerdem sind der minimale und maximale Wert speicherbar (siehe **Bild 1**). Aus den Messwerten kann ferner die Standardabweichung berechnet werden, auch dieser Wert wird im Speicher abgelegt. Der Modus wird für alle Eingänge gemeinsam mit den ersten fünf Bit des Statusregisters S1 festgelegt. Eine besondere Einstellung gilt für den Fall, dass der Messwert gleich 1 ist (digitales Signal oder CPS). Der Mittelwert ist dann als Integer wenig aussagefähig. Wenn Bit 0 und 1 von S1 gesetzt sind, wird bei 1-bit-Werten die Summe der Messwerte gespeichert, bei Werten mit mehreren Bits dagegen der Mittelwert.

Das Speichern des Mittelwerts, des höchsten und niedrigsten Werts und der Standardabweichung ist nicht immer effizient. Bei normalverteilten Werten sind der höchste und niedrigste Wert meistens gleich dem Zweifachen der Standardabweichung plus und minus dem Mittelwert. Wenn die Werte normal verteilt sind, genügt es, den Mittelwert und die Standardabweichung zu speichern. Soll beispielsweise der Temperaturverlauf eines Kühlschranks



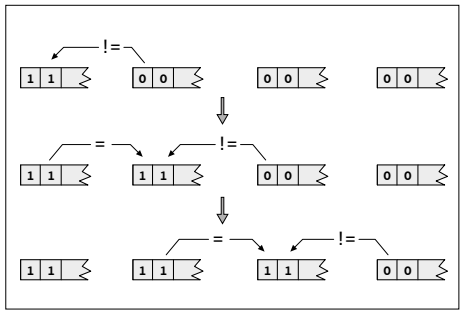


Bild 2. Die ersten beiden Bits in der RunID.

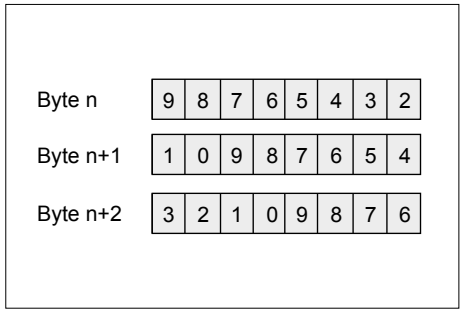


Bild 3. Stapeln der Bits, hier die Bits 9...0 einer ADC-Messung.

geloggt werden, wobei Spitzen bei den hohen Temperaturen zu erwarten sind, genügen der mittlere und der höchste Wert. Daraus ist auch die Dauer der Spitzen ableitbar. Breite Spitzen weisen auf die Nähe des Mittelwerts zum Maximum hin.

Nicht alle Einstellungen führen zu sinnvollen Ergebnissen. Ein Grund ist die Anzahl der Bits, aus denen die Variablen bestehen. Die Summe der Messungen kann höchstens die Breite 32 bit haben (4294967295), beim Minimum und Maximum sind die Werte auf 16 bit (65535) begrenzt. Wenn der ADC des Datenloggers lediglich den Mittelwert oder die Summe speichert, können vom ADC bis zu 4194304 Messungen vorgenommen werden. Die Standardabweichung lässt sich berechnen, indem das Quadrat der Summe aus den Messwerten gebildet wird, sie werden in einem 32-bit-Register gespeichert. Wenn es um die Standardabweichung eines 10-bit-ADCs geht, sind bis zu 64 Messungen ausführbar. Dann muss beim Messen mit dem ADC der Mittelwert weiterverarbeitet werden (Statusregister 1).

Zu beachten sind auch die Zeiten, die das Datenloggen beansprucht. Das Speichern eines 64-Byte-Blocks dauert 5 ms, das Berechnen der Standardabweichung dauert pro Kanal 2,5 ms, der Messvorgang ist dagegen im Bruchteil einer Millisekunde abgeschlossen. Wenn beispielsweise die Schreibperiode für vier Kanäle auf 10 ms eingestellt ist, mit 64 Messungen und Standardabweichung, beträgt die effektive Geschwindigkeit weniger als die Hälfte, unter starkem Jitter. Auch der Jitter ist auslesbar.

Ist über Bit 2 des Statusregisters S0 der automatische Neustart eingestellt, müssen diese Einstellung und die Einstellungen der Leitungen im EEPROM des Mikrocontrollers gespeichert werden (Kommando 2 in das Statusregister Z3). Anderenfalls gehen die Einstellungen beim Neustart des Mikrocontrollers verloren. Ferner muss Bit 2 des Statusregisters Z gesetzt sein, so dass die Werte tatsächlich gespeichert werden.

Die Datenlogger-App liest unmittelbar aus dem seriellen Flash-Speicher die dort stehenden Daten. Es ist deshalb wichtig, die Struktur zu kennen, mit der die Daten gespeichert sind.

Serieller Flash-Speicher

Der Speicherbaustein 24FC256 verfügt über eine I²C-Schnittstelle, so dass zwei Leitungen zur Anbindung an den Mikrocontroller genügen. Die Leitungen C3 und C4 arbeiten intern mit offenen Kollektoren, die Leitungen werden entweder nach Masse gezogen oder von Pullup-Widerständen auf die Betriebsspannung gelegt. Dadurch werden Kurzschlüsse vermieden, falls zwei Bausteine gleichzeitig schreibend auf den I²C-Bus zugreifen. Das Signal auf der Datenleitung wird mit der Vorderflanke des Taktsignals übernommen, die Bus-Geschwindigkeit beträgt maximal 1 Mbit/s.

Beim Ansteuern des seriellen Flash-Speichers ist zu berücksichtigen, dass die Anzahl der Schreib- und Löschzyklen nicht unendlich ist. Soll lediglich ein einzelnes Byte geschrieben werden, wird eine 64 Byte lange Reihe zuerst lokal gespeichert. Anschließend wird die gesamte Reihe gelöscht, und die Daten werden neu geschrieben. Wenn die 64 Byte einer Reihe byteweise gespeichert werden, sind 64 Löschvorgänge der gesamten Reihe nötig. Der 24FC256 überlebt laut Hersteller mindestens 1 Million Löschvorgänge. Daraus folgt, dass dieser Baustein effektiv nur ungefähr 15000 Mal beschreibbar ist. Eine schonendere Methode besteht darin, dass der Mikrocontroller die 64 Byte einer Reihe puffert, um sie anschließend in einem einzigen Vorgang in den Speicher zu schreiben. Die Lebensdauer des 24FC256 wird dadurch um den Faktor 64 verlängert.

Der Flash-Speicher wird geschont, indem die aktuelle Schreibposition nicht im Flash-Speicher, sondern im RAM des Mikrocontrollers festgehalten wird. Allerdings entsteht beim Ausfall der Betriebsspannung ein Problem, denn die Information über die Schreibposition geht verloren. Der Autor hat das Problem mit einem Kunstgriff gelöst: Um die Information in den Daten unterzubringen, werden die ersten Bits am Anfang jedes 64-Byte-Blocks als so genannte Run-Identifier (runID) eingesetzt. Das Protokoll schreibt vor, dass das erste Bit des 64-Byte-Blocks identisch mit dem zweiten Bit des 64-Byte-Blocks davor ist, während das zweite Bit den inversen Wert des ersten Bits im 64-Byte-Block danach hat. In **Bild 2** werden die Zusammenhänge veranschaulicht. Das Ganze läuft darauf hinaus, dass bei den zusammengehörigen Daten das Bit 6 des 64-Byte-Blocks und das Bit 7 des folgenden Blocks identisch sind, während bei den vorangehenden Daten die Bits voneinander abweichen. Die Methode macht die Wiederaufnahme der Datenspeicherung an der ursprünglichen Position möglich.

Außerdem haben die Bits 3...5 die Funktion von Flags. Wenn die Speicherkapazität überschritten wurde und der zweite (oder ein weiterer) Zyklus begonnen hat, wird dies von Bit 3 angezeigt. Bit 4 markiert den Block, bei dem ein Neustart stattfand, so dass dort eine Inkonsistenz der Daten möglich ist. Falls der Mikrocontroller ausgelastet war und eine Messung nicht im regulären Zeitraster ausgeführt werden konnte, ist Bit 5 gesetzt. Dies kann wichtig sein, da die Messwerte ohne Zeitstempel gespeichert werden. Normalerweise ist davon auszugehen, dass die Messwerte im gewählten Intervall aufeinander folgen. Wenn Bit 5 gesetzt ist, gab es eine Unregelmäßigkeit, in diesem Fall ist Jitter aufgetreten. Rückgesetzt wird Bit 5 nur, wenn der Datenlogger neu startet.

Die Daten werden bitweise kettenartig hintereinander gehängt. Sollen beispielsweise zwei 10-bit-Werte geschrieben werden, speichert das erste Byte die oberen 8 Bits, die beiden verbleibenden Bits übernehmen die beiden oberen Bits des folgenden Bytes. Vom nächsten 10-bit-Wert werden die oberen 6 Bits in die unte-

ren 6 Bits des zweiten Bytes geschrieben, und die übrigen 4 Bits haben ihren Platz in den oberen 4 Bits des dritten Bytes, und so weiter. In **Bild 3** ist dargestellt, wie die Bits verteilt sind. Jeder Messwert, der gespeichert wird, kann bis zu 63 Bytes lang sein. Theoretisch lässt sich das Android I/O-Board so konfigurieren, dass ein Datensatz länger als 63 Bytes sein kann. In diesem Fall generiert das Board eine Fehlermeldung, indem Bit 1 des Status-Bytes gesetzt wird, lesbar mit „R C4“. Der Betrieb des Datenloggers stoppt.

Die App kann die Daten aus dem seriellen Flash-Speicher nur verarbeiten, wenn die zu den einzelnen Kanälen gehörenden Bitlängen bekannt sind. Diese Daten sind im ersten 64-Byte-Block des Speichers abgelegt. Dort steht, welche Leitungen gelesen werden, welche Daten zu speichern sind und mit wie vielen Bits dies zu geschehen hat. In **Tabelle 1** ist der Inhalt dieses Headers zusammengefasst.

Der Flash-Speicher kann mit dem Kommando „Read“ und der Adresse im Register C3 gelesen werden (Beispiel: r c3 128). Der Mikrocontroller gibt dann 32 Bytes zurück, die genau in den Schreibpuffer passen. Bei mehr als 32 Bytes blockiert der Mikrocontroller und folglich das Datenlogging, bis die Daten über die 9600-Baud-Verbindung übertragen sind. Die App muss bis zum Ende der Übertragung warten, bevor sie die nächsten 32 Byte anfordert. Damit wird dem Jitter und anderen Unregelmäßigkeiten bei den Messungen vorgebeugt.

Über die Adressen 32768 und 32800 wird nicht der serielle Flash-Speicher, sondern der aktuelle Puffer gelesen. Insbesondere bei langen Perioden, beispielsweise 1 h für eine Messung, wobei der Pufferinhalt vielleicht einmal pro Tag gespeichert wird, kann die App trotzdem auf die aktuelle Messung zugreifen.

Durch Lesen des Registers C4 ist der Status der Messung abfragbar. Im hexadezimalen Format werden der Status, die Position im Flash-Speicher (in 64-Byte-Blöcken), die Position in Bytes und Bits im Puffer, die Anzahl der noch auszuführenden Messungen bis zur nächsten Schreibaktion in den Puffer sowie die Anzahl der Millisekunden bis zur folgenden Messung zurückgegeben. Die wichtigsten Bits des Status sind Bit 0 (Datenlogging aktiv), Bit 2 (Jitter) und Bit 4 (der Speicher wurde einmal oder mehrfach zyklisch durchlaufen).

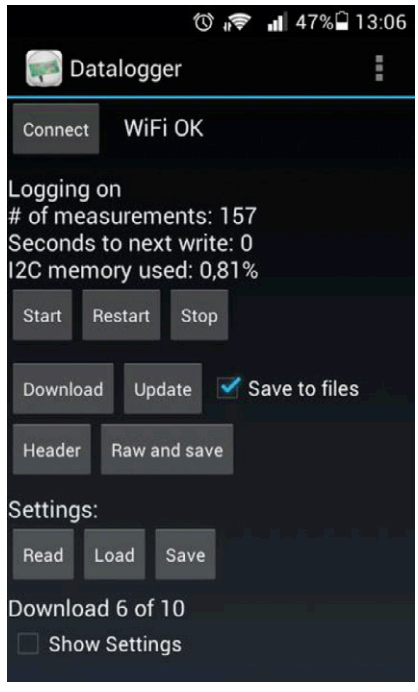


Bild 4. Die für den Datenlogger entwickelte App auf dem Bildschirm.

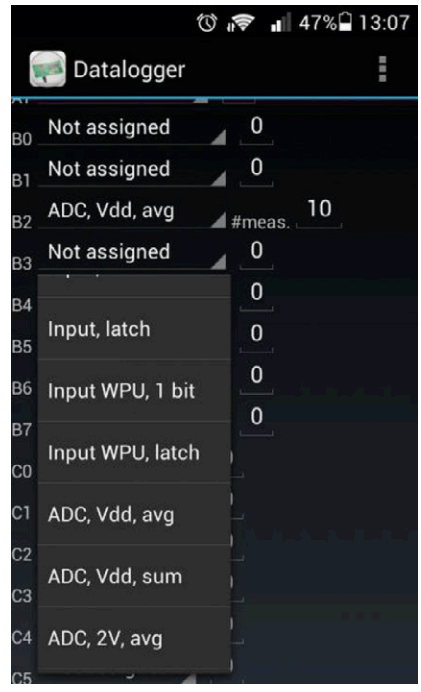


Bild 5. Das Dropdown-Menü enthält für jede Leitung die verfügbaren Optionen.

Datalogger-App

Mit der App, die für das Android I/O-Board geschrieben wurde (**Bild 4**), werden die Einstellungen des Datenloggers vorgenommen: Frequenz, Anzahl der Messungen, zyklisches Überschreiben des Speichers bei Überlauf, Datenlogger aktiv sofort beim Programmstart. Ferner ist wählbar, welche Daten gespeichert werden: Summe oder Mittelwert, höchster und niedrigster Wert sowie Standardabweichung. Hier werden auch die zu den diversen Leitungen gehörenden Funktionen festgelegt. In einem Dropdown-Menü (Android-Terminologie: „Spinner“) sind die verfügbaren Optionen für jede Leitung gelistet (**Bild 5**). Zu bestimmten Optionen gehören weitere Parameter, beispielsweise zum ADC die Anzahl der Messungen und zum CPS der Schwellenwert. Diese Parameter sind ebenfalls einstellbar.

Der Datenlogger kann aus der App gestartet und gestoppt werden. Nach einem Stopp ist die Wiederaufnahme des Betriebs (Restart) möglich, sofern

Tabelle 1. Aufbau des Headers.

Byte	Funktion (niedriges Byte LSB)
0	RunID
1	Status S0-Register
2	Status S1-Register
3	Anzahl zusätzlicher Bits
4,5	Anzahl der Bits insgesamt pro Messung (alle Kanäle)
6,7,8,9	Anzahl der Messungen pro Schreiboperation
10,11,12,13	Anzahl der Millisekunden zwischen den Messungen
16	Pin-Nummer des ersten Kanals (255: kein Pin)
17	Anzahl der Bits des ersten Kanals
...	
30	Pin-Nummer des achten Kanals (255: kein Pin)
31	Anzahl der Bits des achten Kanals

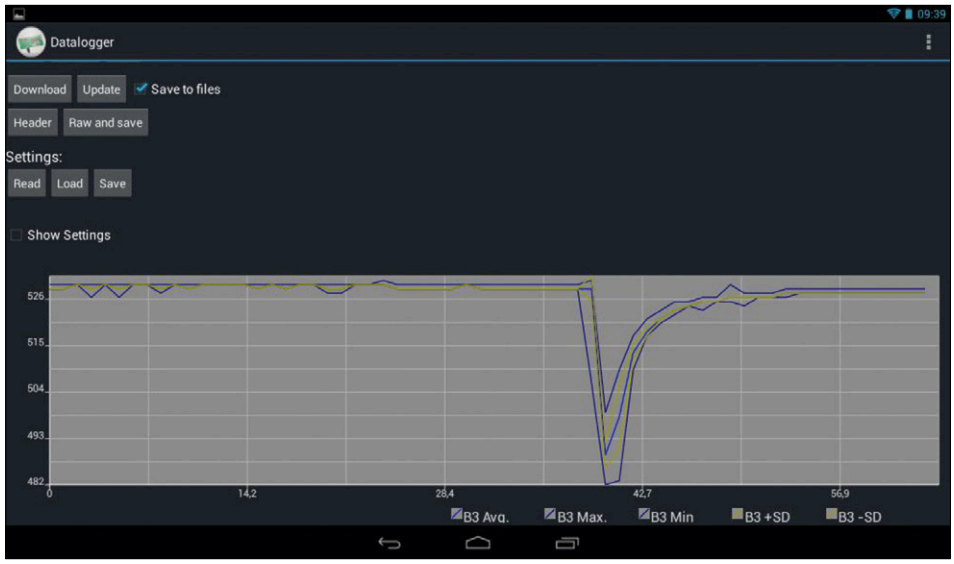


Bild 6. Darstellen der Messdaten in einer Grafik.

Funktionen und Parameter nicht geändert wurden. Geänderte Parameter werden bei einem Restart nicht gespeichert. Die App gibt auch Auskunft über den Status des Datenloggers, beispielsweise über den Einschaltzustand, den Speicherplatz, der belegt wurde, und die Zeit, die bis zum nächsten Messen und Speichern verbleibt. Der Download der Daten ist vollständig oder inkrementell möglich. Beim inkrementellen Download werden nur hinzugekommene Daten gelesen. Die Daten werden in einer Grafik dargestellt und optional im CSV-Format ausgegeben. CSV-Dateien können mit Microsoft Excel oder Libre Office Calc weiterverarbeitet werden. Ferner wird eine Datei erzeugt, die protokollierende Informationen zum Datensatz enthält. Dort steht beispielsweise, welche Leitungen gelesen wurden und wie die Parameter gewählt waren. Die Informationsdateien werden im Ordner „Datalogger“ auf dem Android-Gerät abgelegt. Weitere Möglichkeiten sind das Lesen ausschließlich des Header-Blocks sowie der Download und das Speichern der unbearbeiteten Daten aus dem Flash-Speicher.

Der Aufbau dieser App ist mit der im zweiten Teil des Beitrags vorgestellten App vergleichbar, die das Erwärmen eines Ätzbads steuerte. In „onCreate“ werden die grafischen Elemente, die Hintergrundschleife und der Message-Handler gestartet. Die „Spinner“ für die Leitungsfunktionen sind jedoch nicht in den XML-Dateien definiert, sondern in Java in einer Schleife untergebracht. Parallel dazu steckt das gesamte Layout in einem „ScrollView“, so dass gleichzeitig ein Scrollen möglich ist. Die Kommandos für die Kommunikation mit dem Android I/O-Board sind mit den Kommandos der App für das Ätzbad im zweiten Teil identisch.

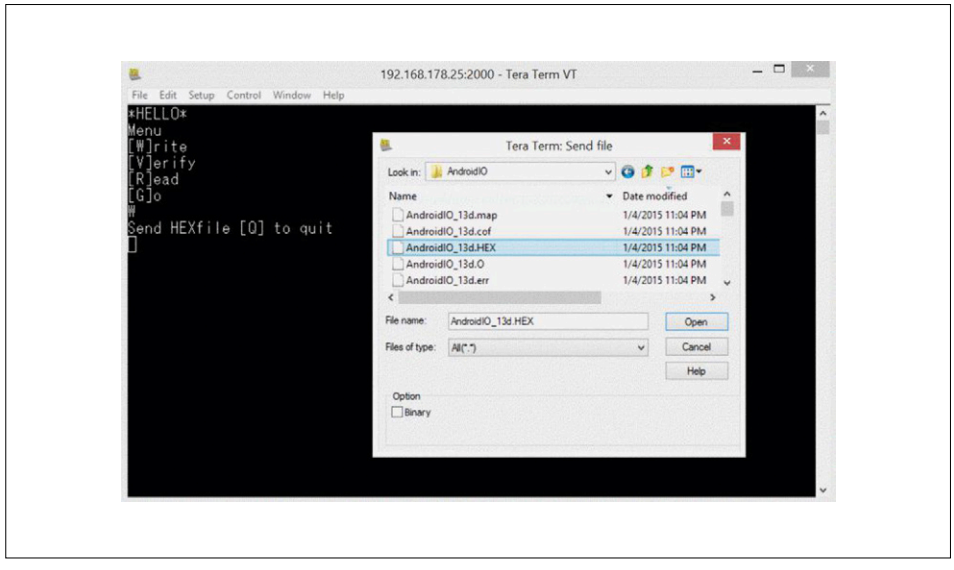


Bild 7. Upload einer Datei zum Android I/O-Board vom PC mit Tera Term.

Auch diese App entbehrt nicht einer gewissen Komplexität. Sie entsteht hauptsächlich durch das asynchrone Übertragen von Kommandos zum Android I/O-Board sowie das Empfangen von Kommandos. Das bedeutet, dass ein Teil der Funktionalität in einem eigenen Thread stattfindet, gesteuert von den Variablen, die als Flags dienen (eventuell gesetzt von einem „Clicklistener“). In Gegenrichtung werden mit dem Message-Handler Informationen zum Steuern der grafischen Elemente übergeben. Dadurch ist die Funktionalität, die eigentlich eine Einheit bilden sollte, auf zwei oder drei Positionen im Programmcode verteilt. Die Übersichtlichkeit ist nicht optimal, Programme in Java sind aber dafür bekannt.

In dieser App ist ein Puffer mit Verzögerungszeiten eingebaut, die zum Steuern der Kommandos dienen. So werden verschiedene Leitungsfunktionen mit 10 ms Verzögerung gesetzt, damit die Puffer des Android I/O-Boards nicht überlaufen und Kommandos nicht überschlagen werden. Diese Funktionen können auch in andere Apps übernommen werden, die der Anwender schreibt. Ferner werden im Flash-Speicher aufeinander folgende Blöcke einmal in 120 ms gelesen.

Beim Programmieren einer App sollte stets geprüft werden, ob passende und verwendbare Bibliotheken existieren. Zum Darstellen der Grafik hat der Autor die Open-source-Bibliothek AndroidPlot in die App eingebunden. Wenn der Quellcode in Eclipse editiert werden soll, muss die Datei *androidplot-core-x.x.x.jar* aus dem Internet [1] heruntergeladen werden, diese Datei muss im Ordner */lib* stehen. AndroidPlot stellt Grafiken visuell ansprechend dar, einschließlich Achsen und Legende, und ferner ist ein Zoom möglich (siehe **Bild 6**). Weitere Informationen sind auf der Website von AndroidPlot [2] zu finden.

Bootloader

Der Autor hat anlässlich dieses Projekts einen Bootloader für den Mikrocontroller

entwickelt. Der Bootloader ist in einem gesicherten Bereich des Speichers untergebracht, so dass er von der Firmware nicht überschrieben werden kann. Mit dem Bootloader lässt sich auf einfache Weise alternative oder modifizierte Firmware testen.

Der gesicherte Speicherbereich des Mikrocontrollers umfasst nur 512 Instruktionen. Um den Bootloader unterbringen zu können, mussten verschiedene Kunstgriffe angewendet werden. So besteht beispielsweise der Text aus zwei Zeichen für ein 14-bit-Word, und die Variablen sind ebenso wie der Code so gruppiert, dass möglichst wenig Bank-Select-Statements benötigt werden.

Der Bootloader wird aktiviert, wenn Leitung B2 des Mikrocontrollers während des Systemstarts über einen Taster oder Jumper K6 an Masse liegt. In diesem Fall leuchtet LED3 auf, sie wird über Leitung RB0 gesteuert. Auf einem Terminalprogramm, das auf dem PC läuft und mit dem Board über Bluetooth, WiFi oder USB kommuniziert, erscheint ein Menü (**Bild 7**). Um Programmcode in den Mikrocontroller zu laden, ist auf der Tastatur das „W“ (Write) zu drücken. Der Bootlader fordert dann die zu ladende Hex-Datei an. Beim Terminalprogramm TeraTerm Pro wird die Hex-Datei mit *file/send file...* ausgewählt und hochgeladen. Da die Geschwindigkeit nur 9600 Baud beträgt, kann dies abhängig von der Dateigröße mehrere Minuten dauern. Mit „Q“ (Quit) ist bei allen Menü-Optionen die Rückkehr zum Hauptmenü möglich.

Durch Drücken von „V“ (Verify) lässt sich kontrollieren, ob die Hex-Datei im Mikrocontroller angekommen ist. Mit „L“ (List) wird die Ausgabe des Speicherinhalts im Adressbereich 0x0200...0x3FFF gestartet. Falls mehr als vier leere Speicherplätze (Wert 0x3FFF) aufeinander folgen, wird erst wieder der nächste belegte Speicherplatz ausgegeben. Zum Abschluss erscheint die letzte Adresse.

Um die Firmware zu starten, muss „R“ (Run) gedrückt werden. Die Rückkehr zum Bootloader ist möglich, wenn in der Firmware die Werte 20 oder 21 nach Z3 geschrieben werden. Im ersten Fall startet der Bootloader nur, wenn beim Systemstart der Bootloader-Taster gedrückt ist, im zweiten Fall startet der Bootloader unmittelbar. Der Sprung zum Bootloader ist auch möglich, indem der Großbuchstabe „Q“ im Debug-Modus gesendet wird. Alternative Firmware des Anwenders kann nach 0x0000 oder 0x0019 springen. In beiden Fällen muss zuvor der Interrupt deaktiviert werden, PCLATH und BSR sind auf 0 zu setzen.

Anpassen der Firmware

Der Anwender kann die Firmware anpassen, und es gibt Gründe, weshalb er dies tun sollte. Wenn die gewählten Grundeinstellungen in den Code eingefügt werden, müssen sie nicht erst zum Android I/O-Board übertragen und im EEPROM gespeichert werden. Ferner ist für besondere I^2C- und SPI-Bausteine spezifische Firmware erforderlich, und die Software könnte auch bestimmte Formate (zum Beispiel IR-Fernbedienung, 433-MHz-Module) unterstützen.

Das Anpassen der Firmware setzt Kenntnisse des Microchip 14-bit-Assemblers voraus. Unter der MPLAB IDE oder MPLAB X lässt sich die Firmware modifizieren, kompilieren und testen. Die benötigten Source-Dateien (AndroidIO.asm und ppmacros.asm) stehen auf der Elektor-Projektseite [3] zum Download bereit.

Die Struktur der Firmware ist in **Bild 8** dargestellt, sie besteht aus zwei Schleifen. Die Interrupt-Routine wird aufgerufen, um in Echtzeit die PWM zu generieren, den CPS zu messen, den Counter zu lesen und die zeitrelevanten Variablen zu aktualisieren. Dazu gehören die Flags des ADC, der Taktgeber, der Watchdog und der Zeitgeber. Während des Interrupts wird kein Text über den seriellen Port ausgegeben, dies geschieht ausschließlich in der Hauptschleife. Dadurch wird vermieden, dass die Interrupt-Routine blockiert. Die Hauptschleife schreibt in einen Puffer. Wenn der UART bereit ist, überträgt die Interrupt-Routine den Pufferinhalt zeichenweise zum seriellen Port (Android-Gerät). In der Hauptschleife befindet sich ein Parser für die Kommandos und (sofern aktiviert) für das Erfassen von Änderungen (Input und CPS) oder die periodische Anzeige des ADC-Werts und des Zeitgebers. Die Hauptschleife steuert auch den Datenlogger.

Weitere Hinweise zum Anpassen der Firmware, zum Beispiel zum Implementieren der Standard-Einstellungen, sind unmittelbar im Quellcode vermerkt.

Andere Systeme

In diesem Beitrag stand das Steuern des Android I/O-Boards durch ein Android-Gerät im Mittelpunkt. Das Board ist aber auch zusammen mit einem PC oder Laptop unter Windows oder Linux oder mit Systemen wie dem Raspberry Pi einsetzbar.

Der Autor nutzte während eines Urlaubs ein Android I/O-Board mit WLAN, um die Außentemperatur zu erfassen und diese Information mit einem Raspberry Pi als E-Mail zu versenden. Die Basis ist ein simples Bash-Script, das einmal in der Viertelstunde über das Telnet-Kommando Kontakt mit dem Android I/O-Board aufnimmt, das Ergebnis in Grad Celsius umsetzt und den Wert einer Datei hinzufügt. Die Daten werden vier Mal am Tag mit GnuPlot in eine GIF-Grafik übertragen, die Grafik wird als Mail verschickt (**Bild 9**). Das Script ist in **Listing 1** wiedergegeben, dort wird vorausgesetzt, dass die Programme BC, Gnuplot und Mutt installiert sind.

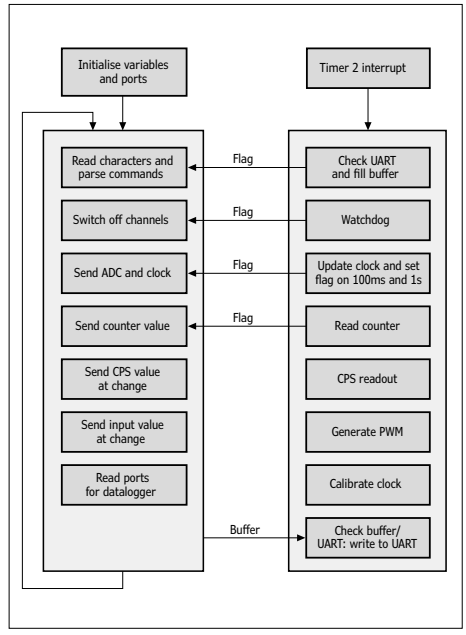


Bild 8. Struktur der Firmware mit Hauptschleife und Interrupt-Routine.

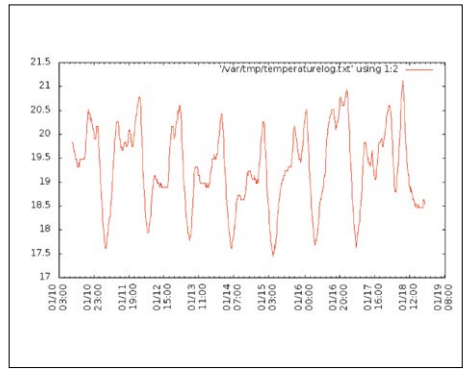


Bild 9. Die Temperaturkurve wird von einem Raspberry Pi als Mail verschickt.

Listing 1. Script zum periodischen Messen der Temperatur und Darstellen als Grafik.

```
#!/bin/bash
IOcmmd () { sleep 5; echo; echo s b3 3#; sleep 1; echo s b32 10#; sleep 1; echo r b3#;sleep 3;}
txtfile='/var/tmp/temperaturelog.txt'
giffile='/var/tmp/temperaturelog.gif'
loop=0
beta="3900"
r00='echo "10000*e(-$beta/298)"|bc -l'
while true
do
  adcvalue1='IOcmmd | telnet 192.168.178.27 2000 | grep "R B3" |tail -n 1 | awk '{ printf $3 }''
  adcvalue=$(if [ "$adcvalue1" -gt "1" ] 2>/dev/null; then echo $adcvalue1; else echo 1; fi)
  rntc='echo "10000/((1024/$adcvalue)-1)"|bc -l'
  temp='echo "$beta/l($rntc/$r00)-273" | bc -l'
  printf "%s %.2f\n" 'date +%Y-%m-%d_%H:%M' $temp >> $txtfile
  if [ "'echo $loop%24 |bc'" -eq "0" ]; then
    xlines='cat $txtfile |wc -l'
    nofxtic='echo $xlines*900/10|bc'
    printf "set terminal gif size 800,480\nset output \'$giffile\'\nset timefmt \"%%Y-%%m-%%d_%%H:%%M:%%S\"\
     nset bmargin 4\nset xdata time\nset xtics rotate 270\nset xtics $nofxtic\nplot \'$txtfile' using 1:2
     with lines\n" |gnuplot
    echo Temperature |mutt -s "Temperature" xxx@gmail.com -a $giffile
  fi
  loop='echo "$loop+1"|bc'
  sleep 900
done
```

Das Android I/O-Board ist über die Jumperkontakte JP2 auch unmittelbar am seriellen Port des Raspberry Pi anschließbar, so dass sich das Kommunikationsmodul erübrigt. Dann hat das Board die Funktion eines I/O-Extenders für den Raspberry Pi. Die Anschlüsse 6 (GND), 8 (TXD) und 10 (RXD) des Raspberry Pi werden mit den ersten drei Anschlüssen auf der rechten Seite von JP2 verbunden (TXD mit R und RXD mit T). Bei Bedarf kann auch die 3,3-V-Leitung des Raspberry Pi mit dem vierten Anschluss verbunden werden, so dass der Raspberry Pi die Betriebsspannung des Android I/O-Boards liefert. Der zulässige Strom darf natürlich nicht überschritten werden.

Die Software kann den seriellen Port mit */dev/ttyAMA0* bedienen, die Baudrate ist mit *stty -F /dev/ttyAMA0 speed 9600* einstellbar. Da der serielle Port normalerweise zum Austausch von Logging-Informationen verwendet wird, muss diese Funktion hier deaktiviert werden. Dies lässt sich durch Anpassen der Dateien */etc/inittab* und */boot/cmdline.txt* bewerkstelligen. In der ersten Datei muss eine Raute (#) gesetzt werden:

```
#T0:23:respawn:/sbin/getty -L
ttyAMA0 115200 vt100
```

In der zweiten Datei ist die Zeile

```
console=ttyAMA0,115200
kgdboc=ttyAMA0,115200
```

zu löschen.

(150310)gd

Weblinks

[1] http://androidplot.com/download/

[2] http://androidplot.com/

[3] www.elektormagazine.de/150310

[4] http://ww1.microchip.com/downloads/en/DeviceDoc/80000501F.pdf

Bug im PIC16F1938

In die ersten beiden Versionen des PIC16F1938 (Revision A und B) hat sich ein Fehler eingeschlichen, so dass der ADC bei Taktfrequenzen ab 16 MHz nicht korrekt gelesen werden kann. Auch der Absturz des Mikrocontrollers ist möglich (siehe [4]). Die Versionsnummer ist auslesbar über das Statusregister Z3, Kommando 11.
Es ist ein Workaround verfügbar, der diesen Fehler umgeht. Dazu muss während des Lesens der Interrupt deaktiviert werden, und die Taktfrequenz ist temporär abzusenken. Allerdings kann dies Jitter im PWM-Signal und im Counter zur Folge haben. Aus diesem Grund existieren zwei Versionen der Firmware: Die reguläre Version für die Revision C (und höher) des PIC 16F1938 sowie die Version mit dem Workaround, erkennbar an „_AB“ in der Bezeichnung.

Digitales Zoetrop

Uraltes Prinzip, neue Technik

Von **Willem van Dreumel** (NL)

Womit spielen Ihre Kinder, wenn das Wetter schlecht ist? Es müssen nicht immer die Apps auf dem Smartphone sein. Versuchen Sie es mal mit einem Zoetrop! Vielleicht bedeutet Ihnen dieser Begriff nichts, vielleicht löst er Erinnerungen aus. Egal, dieses kleine Projekt ist eine erprobte Medizin gegen familiäre Langeweile, nicht nur an herbstlich trüben Tagen.

Leser der jungen Generation sind ihm vielleicht noch nie begegnet, die älteren Leser kennen es vielleicht aus frühen Jahren: Das Zoetrop [1], in gewissem Sinn ist es der Vorläufer der Nipkow-Scheibe [2] und der bewegten Bilder auf der Leinwand. Das Prinzip ist verblüffend einfach. Bilder oder Grafiken, die aufeinander folgende Bewegungsphasen eines Objekts darstellen, wechseln einander in schneller Folge ab. Beim alten Original des Zoetrop wird dies erreicht, indem die Bilder auf der Innenseite einer rotierenden Trommel angebracht sind. Die Trommel ist oben offen, und in der Trommelwand sind in gleichmäßigen Abständen Lichtschlitze eingelassen. Die Anzahl der Schlitze ist mit der Anzahl der Bilder identisch.

Digitale Technik

Auf dem Stand der heutigen Zeit ist die Steuerung in **Bild 1**. Die Schaltung lässt einen Schrittmotor rotieren, der eine runde Scheibe dreht, Durchmesser 10...15 cm. Auf der Scheibe sind radial die Momentaufnahmen des Bewegungsablaufs angebracht. Dazu synchron blitzt eine leistungsstarke LED auf. Der Betrachter sieht wegen der Trägheit des menschlichen Auges einen stabilen Bewegungsablauf. Durch den Einsatz eines Mikrocontrollers, hier ein ATtiny85, kommt die Steuerung mit nur sechs Bauteilen aus. Mit dem **Arduino-Sketch**, der hier gelistet ist, generiert der Mikrocontroller die Steuersignale für den Schrittmotor und die LED. Die LED leuchtet bei jedem 20. Schritt für 1 ms auf.

Eine hilfreiche Anleitung zum Programmieren des ATtiny85 (und anderen) findet sich auf der Website von High-Low Tech [3]. Da das Programm als Arduino-Sketch geschrieben ist, kann auch ein Arduino-Board die Steuerung übernehmen. In diesem Fall ist auf die Nummern der Ports zu achten, über die der Schrittmotor und die LEDs gesteuert werden. Die Port-Nummern müssen an das Arduino-Board angepasst werden.

Motor und LED

Die Steuerung des bipolaren Schrittmotors ist unkompliziert, denn der integrierte Baustein L293D hat alles Nötige an Bord.

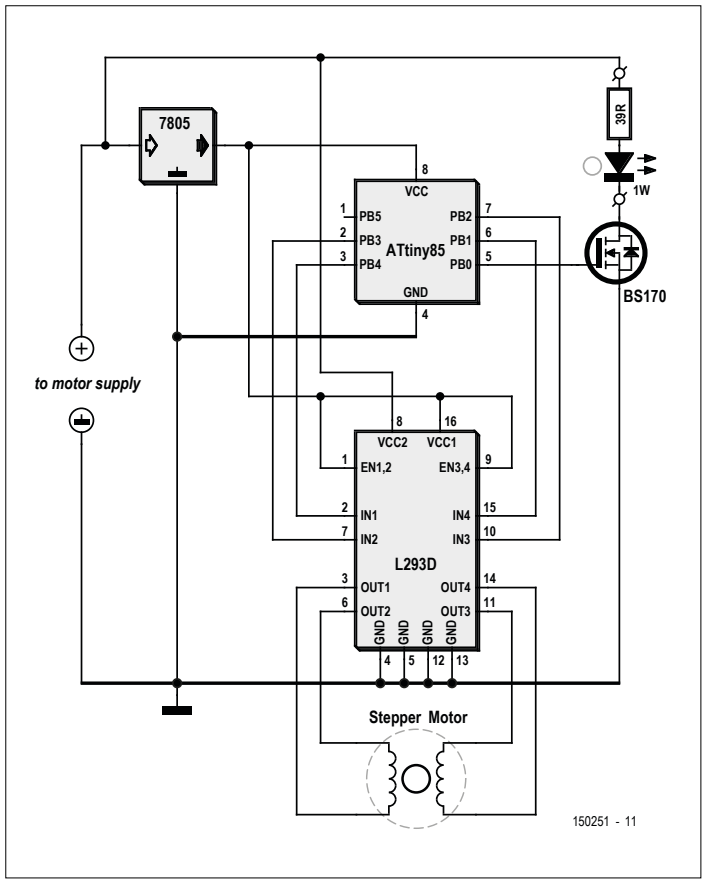


Bild 1. Die Steuerung besteht aus nur sechs Bauteilen.

Er wird über vier digitale Ausgänge des ATtiny85 gesteuert. Jedes Mal wenn 20 Schritte durchlaufen sind, schaltet der BS170 durch, so dass die LED aufblinkt. Als LED ist ein leistungsstarkes Exemplar, zum Beispiel ein 1-W-Typ, am besten geeignet. Das Umgebungslicht soll den sichtbaren Bewegungseffekt möglichst wenig stören. Die Betriebsspannung für den Schrittmotor und die LED ist in Grenzen wählbar. Allerdings muss der Widerstand angepasst werden, der den LED-Strom begrenzt. Bei der Betriebsspannung 12 V beträgt der Wert ungefähr 33 Ω; da der Duty-Cycle nur 5 % beträgt, genügt ein 0,25-W-Widerstand. Aus der unten angegebenen **Widerstandsberechnung** geht hervor, wie der Widerstand berechnet wird. Damit bewegt sich der LED-Strom auf der sicheren Seite.

Miniplatine

Die kleine Platine in **Bild 2** bietet Platz für vier der sechs Bauteile. Die LED, der Vorwiderstand und auch der Schrittmotor sind aus praktischen Gründen an anderen Orten untergebracht. Die LED muss die Scheibe beleuchten, sie wird vor der Scheibe montiert, der Vorwiderstand kann aus einzelnen Widerständen unterschiedlich zusammengestellt sein, und der Schrittmotor wäre montiert auf der Platine nicht sehr praktisch. In Bild 2 ist angegeben, wo die LED und der Vorwiderstand angeschlossen werden. Die Verbindungspunkte, die sich rechts davon befinden, sind die Anschlüsse der Betriebsspannung. Natürlich sind die Polaritäten der LED und der Betriebsspannung zu beachten. Der L293D verträgt bis 36 V, beim 78L05 ist 35 V die Obergrenze der Eingangsspannung. Weil beide ICs bei Spannungen dieser Größenordnung viel Energie in Wärme umsetzen, ist von hohen Spannungen abzuraten. Die meisten Schrittmotor-Typen arbeiten zuverlässig an einem Bruchteil dieser Spannungen.

Praktisches

Die Schrittweite gängiger Schrittmotoren beträgt 1,8 Grad. Die LED blitzt jedesmal nach 20 Schritten auf, was einem Drehwinkel von 36 Grad entspricht. Eine volle Umdrehung hat bekanntlich 360 Grad, so dass 10 aufeinander folgende Bewegungsphasen auf der Scheibe stehen müssen. Mit der nächsten Umdrehung beginnt die Szene von vorn. Die Anzahl kann auch ein Vielfaches von 10 betragen, doch dann müssen die einzelnen Phasen entsprechend geringere Breiten haben. Falls der Schrittmotor nach dem Einschalten rückwärts dreht, muss der Steckverbinder gewendet werden.

Wenn die kreisförmigen Vorlagen mit den Bewegungsphasen den Außendurchmesser 12 cm haben, können sie auf ausrangierte CDs geklebt werden. Mit den üblichen Text- und Zeichenprogrammen lässt sich der Vorlagenausdruck auf dieses Maß skalieren. Auf die Schrittmotorachse wird zweckmäßigerweise ein Flansch montiert, der die CD trägt. Damit lassen sich

Widerstandsberechnung

Für das Berechnen des Strombegrenzungswiderstands werden folgende Eigenschaften der LED angenommen:

Die LED ist ein 1-W-Typ

- Bei 3,0 V fließen 200 mA durch die LED. Die LED hat die Leistung P=U·I= 0,6 W.
- Bei 3,3 V fließen 350 mA durch die LED. Die LED hat die Leistung P=U·I= 1,15 W.

Der Strom kann folglich im Bereich 200...350 mA liegen. Beim Strom 350 mA und einem Duty-Cycle von 5 % gilt für den Strombegrenzungswiderstand:

$$R = \frac{U}{I} = \frac{(12-3{,}3)V}{0{,}350\,A} = 25\,\Omega$$ und

$$P_{diss} = U \times I \times 0{,}05 = (12-3{,}3)V \times 0{,}35\,A \times 0{,}05 = 0{,}15\,W$$

Beim Strom 200 mA ergibt die Rechnung:

$$R = \frac{U}{I} = \frac{(12-3{,}0)V}{0{,}20\,A} = 45\,\Omega$$ und

$$P_{diss} = U \times I \times 0{,}05 = (12-3{,}0)V \times 0{,}20\,A \times 0{,}05 = 0{,}09\,W$$

Für den Strom ist ein Wert zweckmäßig, der etwa 10...20 % unter dem Höchststrom liegt. Dadurch wird die Verlustleistung sowohl des Strombegrenzungswiderstands als auch der LED herabgesetzt, was die Lebensdauer der LED verlängert. Wenn der Wert rund 33 Ω beträgt, ist dies eine richtige Wahl.

Arduino-Sketch für das Digitale Zoetrop [5]

```
//L293D Zoetrop
int L0 = 1;  //Motorwicklung Rechts
int L1 = 2;
int L3 = 3;  //Motorwicklung Links
int L4 = 4;  //Zeit zwischen den Schritten
int flits = 0; //LED
int t = 1; //Blitzdauer
void setup() {
pinMode(L0, OUTPUT);
pinMode(L1, OUTPUT);
pinMode(L3, OUTPUT);
pinMode(L4, OUTPUT);
pinMode(flits, OUTPUT);}
void loop(){
for(int x=0;x<20;x++){  //abhängig von der
      Schrittweite
digitalWrite(L3, HIGH); digitalWrite(L4, LOW);
      digitalWrite(L0, HIGH); digitalWrite(L1, LOW);
delay(t);
digitalWrite(L3, HIGH); digitalWrite(L4, LOW);
      digitalWrite(L0, LOW); digitalWrite(L1, HIGH);
delay(t);
digitalWrite(L3, LOW); digitalWrite(L4, HIGH);
      digitalWrite(L0, LOW); digitalWrite(L1, HIGH);
delay(t);
digitalWrite(L3, LOW); digitalWrite(L4, HIGH);
      digitalWrite(L0, HIGH); digitalWrite(L1, LOW);
delay(t);
}
digitalWrite(flits, HIGH);
delay(1);
digitalWrite(flits, LOW);
}
```

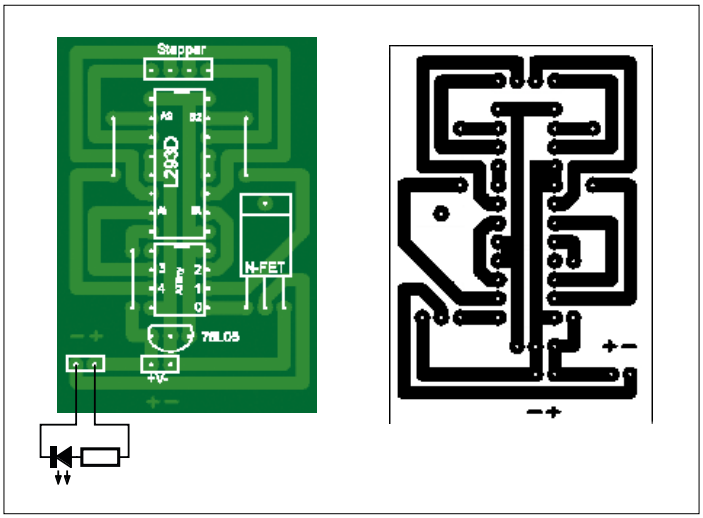


Bild 2. Layout der Platine für das Digitale Zoetrop.

die Scheiben wechseln, ohne dass der Schrittmotor gestoppt werden muss.

Ihre Sprösslinge können Foto- oder Zeichentrick-Szenen in eigener Regie anfertigen, was nicht nur die Kreativität fördert, sondern den Beteiligten auch viel Spaß bereitet. Natürlich spricht nichts dagegen, mehrere CDs auf beiden Seiten zu bekleben und eine kleine Sammlung aufzubauen. Als Beispiel für eine Kollektion ansprechender Grafiken sei eine Webseite von Bekkahwalker [4] genannt. Mehr geeignete Grafiken finden Sie im Netz, wenn Sie nach „Zoetrope“ oder „Zoetrope Disk“ suchen, dies sind die englischsprachigen Begriffe.

(150251)gd

Weblinks

[1] Zoetrop auf Wikipedia: https://de.wikipedia.org/wiki/Zoetrop

[2] Nipkow-Scheibe auf Wikipedia: https://de.wikipedia.org/wiki/Nipkow-Scheibe

[3] Programmieren des ATtiny (englischsprachig): http://highlowtech.org/?p=1695

[4] Sammlung von Grafiken, geeignet für das Digitale Zoetrop: www.bekkahwalker.net/MediaArcheology/disks.html

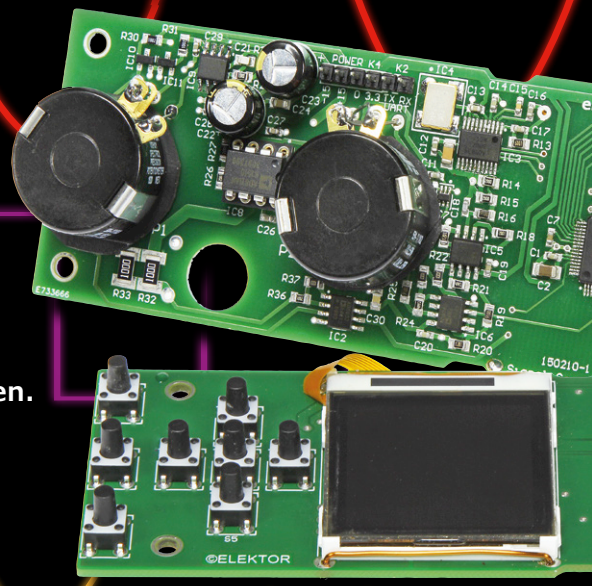
[5] Arduino-Sketch: www.elektormagazine.de/150251

DDS-Funktionsgenerator

Sinus, Rechteck und Dreieck bis 10 MHz

Von **Theodorou Gerasimos** (Griechenland)

Überarbeitung des Designs: **Ton Giesberts** (Elektor-Labor)



DDS-Chips sind leicht erhältlich, so dass der Analogteil eines Weitbereichs-Funktionsgenerators viel einfacher ausfällt als früher. Man muss lediglich ein IC auswählen, eine geeignete Ausgangsschaltung und einen Mikrocontroller hinzufügen, sich eine Benutzeroberfläche ausdenken und ein wenig programmieren. Oder doch nicht?

Vor einigen Jahren brauchte ich einen Funktionsgenerator für mein Heimlabor. Auf der Arbeit hatte ich es mit einigen teuren kommerziellen Modellen zu tun, so dass mir die Idee kam, eines von ihnen zu kaufen. Doch keines war wirklich das, was ich wollte - zu komplex für einfache Anforderungen - und so entschied ich mich, einen eigenen Funktionsgenerator zu entwerfen und zu bauen. Das Gerät sollte auf DDS (Direct Digital Synthesis) basieren, aber nicht nur die Bauteilauswahl stellte sich als ein Problem heraus.

Meine Bauteile, deine Bauteile?

Nachdem ich verschiedene Techniken verglichen hatte, ließ ich mich auf eine Architektur, basierend auf der Direkten Digitalen Synthese (DDS), für mein Projekt ein. DDS verwendet einen digitalen Oszillator mit Präzisionsquarz, um Sinuswellen zu erzeugen, hochpräzise und bis zu sehr hohen Frequenzen.

Es sollte ein Mikrocontroller von Analog Devices zum Einsatz kommen. Dieser Hersteller, der berühmt ist für seine Digitalen Signalprozessoren (DSP), bietet eine Familie von 32-bit-ARM-Controllern, die durch eine reichhaltige und lesbare Dokumentation unterstützt werden. Analog Devices ist auch spezialisiert auf DDS-Chips, Operationsverstärker und andere analoge ICs, so dass schnell klar war, dass ich für das Projekt im Wesentlichen auf AD-Chips zurückgreifen würde. Der freundliche Labormuster-Service von AD hat mich schnell mit dem Nötigsten versorgt, ohne mir dafür eine Menge Geld aus der Tasche zu ziehen. Als **MCU** wählte ich den **ADuC7024BSTZ62**, ein Mitglied der Familie „Precision Analog Microcontroller“ in einem 64-poligen Gehäuse. Der Controller enthält einen ARM7TDMI-Core und läuft mit einem Takt von 44 MHz. Diese MCUs tragen die Bezeichnung „analog“, weil sie über analoge Ein- und Ausgänge (ADC und DAC) und einen Analog-Komparator verfügen. Natürlich sind auch PWM, Timer und serielle Schnittstellen (SPI, UART, I²C) vorhanden. Unser Mikrocontroller verfügt über 8 KB RAM und 62 KB Flash-Speicher, der in-circuit über eine serielle Schnittstelle programmiert werden kann. Einige MCUs dieser Familie verwenden I²C für die Programmierung des Flash-Speichers. Achten Sie deshalb darauf, dass Sie genau den angegebenen Typ bestellen und nicht einen mit einer etwas anderen Teilenummer.

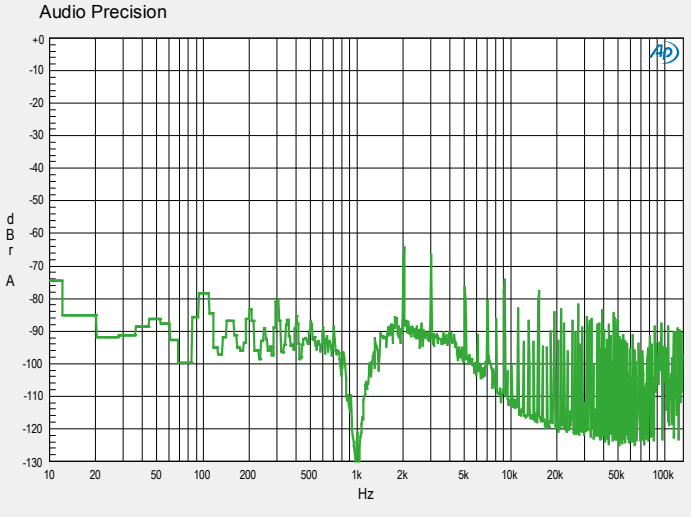
Der **DDS-Chip** für das Projekt ist der beliebte **AD9834**. Die maximale Frequenz des externen Oszillators beträgt 75 MHz, so dass eine maximale Ausgangsfrequenz von 37,5 MHz (die Hälfte der Taktfrequenz) garantiert ist. Der Nachteil eines solch schnellen Taktsignals ist eine Frequenzauflösung von nur 0,28 Hz aufgrund des 28-bit-Integer-Frequenzteilers im Chip. Das klingt nicht nach einem gewaltigen Problem, aber hat doch einen Fehler von fast 0,6 % bei 20 Hz zur Folge. Um dies zu verbessern, kann die Taktfrequenz herabgesetzt werden, allerdings auf Kosten der maximal erreichbaren Ausgangsfrequenz. Bei 1 MHz Taktrate liegt die Auflösung bei 0,004 Hz, was einem Fehler von 0,005% bei 20 Hz entspricht. Als maximale Frequenz werden dann aber auch nur magere 500 kHz erreicht. In diesem Projekt soll die DDS-Taktfrequenz deshalb 75 MHz betragen, um ein relativ sauberes Signal bis zu 10 MHz zu gewährleisten.

Die Benutzerschnittstelle ist wichtiger Bestandteil eines jeden Messgeräts. Ich entschied mich für ein Grafik-**Display** und eine Reihe von Tastern, um einfach durch übersichtliche Menüs und Optionen navigieren zu können. Als Anzeige kommt ein billiges Ersatzdisplay für ein Handy des Typs Nokia 6100 zu Einsatz, das in der Open-source-Community gut dokumentiert ist. Leider existiert dieses Display in

zwei Varianten (Epson und Philips), die nicht vollständig kompatibel sind. Noch ungünstiger ist es, dass man beim Kauf die beiden Typen erst einmal nicht unterscheiden kann. Um diese Misslichkeit zu umgehen, habe ich zwei Softwarevarianten geschrieben, für jeden Displaytyp eine. Zwei Multiturn-Potentiometer sind für die schnelle, aber dennoch genaue Einstellung der Amplitude des Ausgangssignals und des DC-Offsets vorhanden. Natürlich spielt auch die Optik eine Rolle: Der Generator findet sein Zuhause in einem schönen Hammond-**Aluminium-Gehäuse**.

Eigenschaften

- Direkte Digitale Synthese (DDS) mit analogem Frontend
- Frequenzbereich: 1...10 MHz
- Frequenz-Auflösung: 0,28 Hz
- Ausgangsspannung: 0...15 Vss
- THD+N (100 kΩ Last, B>500 kHz):
  - 1 V, 1 kHz: 0,12% (0,09% für B = 22 kHz)
  - 5 V, 1 kHz: 0,1% (0,09% für B = 22 kHz)
  - 1 V, 10 kHz: 0,1% (0,09% für B = 80 kHz)
  - 5 V, 10 kHz: 0,09% (0,08% für B = 80 kHz)
  - 1 V, 100 kHz: 0,1%
  - 5 V, 100 kHz: 0,08%
- S/N (bezogen auf 1 V): 72 dB
- Maximale Ausgangsspannung (10 MΩ Last):
  - Sinus: 16 Vss
  - Dreieck: 16 Vss
  - Rechteck: 18 Vss
- DC-Offsetspannungsbereich: –10 bis +10 V
- Ausgangsimpedanz: 50 Ω
- Tastverhältnis (Rechteck): 1...99%
- Rise- und Fall-time (80%, Rechteck): 100 ns
- Sweep-Modus
- Leistungsaufnahme: 3 VA



Ein FFT-Schrieb eines 1-kHz-Sinussignals mit 1 V Spannung bis 130 kHz und unterdrückter Grundwelle. Die Verzerrungen (THD+N) werden vor allem von den Harmonischen des Signals verursacht. Einige Rückstände der Netzfrequenz (50 Hz) sind sichtbar.

Mehr Bilder und Diagramme auf
www.elektor-labs.com/150210

Die Schaltung

Bild 1 zeigt die Schaltung des DDS-Funktionsgenerators. Der Schaltungsteil, der für die Signalerzeugung zuständig ist, erscheint oben im Bild, unten sind der Mikrocontroller und das User-Interface zu sehen.

Die passiven Bauteile um den DDS-Chip IC3 werden wie vom Hersteller empfohlen eingesetzt. Das Ausgangssignal durchläuft zunächst das RC-Netzwerk R16/C18, bevor es von IC5.A verstärkt wird. Der Tiefpass mag simpel erscheinen (und ist es auch), der Einsatz eines Filters höherer Ordnung erwies sich, wie wir später noch sehen werden, als nicht notwendig.

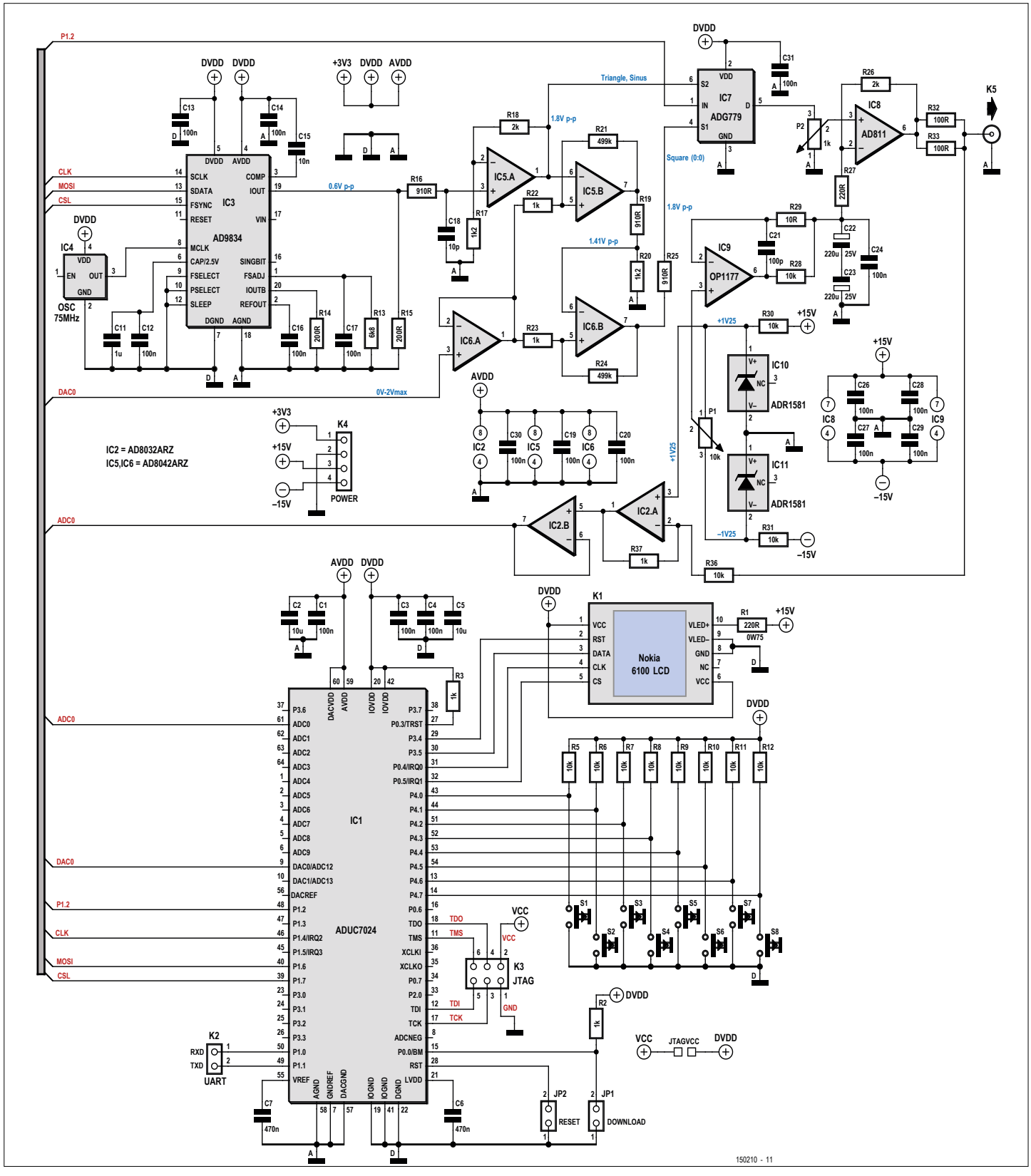


Bild 1. Die Hauptplatine des Funktionsgenerators enthält abgesehen vom Netzteil sämtliche Bauteile.

Das Ausgangssignal von IC5.A folgt zwei Pfaden. Der obere Pfad ist Sinus- und Dreiecksignalen vorbehalten (IC3 kann beides). Sie liegen an S2 von IC7, einem CMOS-Wechselschalter mit sehr guter Isolation zwischen seinen beiden Eingängen S1 und S2. Der untere Pfad wird von Rechtecksignalen verwendet. Sie gelangen an den anderen Eingang S1 von IC7. Das Rechtecksignal wird vom Dreieck abgeleitet. Dieses durchläuft IC5.B und IC6.B, die als recht schnelle Komparatoren (mit richtig bemessenen Hysteresen) geschaltet sind. Die Referenzspannung für die Komparatoren stammt von einem der Analogausgänge (DAC0) des Mikrocontrollers und wird von IC6.A gepuffert. Dies ermöglicht eine präzise Steuerung des Tastverhältnisses des Rechtecksignals.

Am Ausgang des Wechselschalters ist das Abschwächer-Potentiometer P2 angeschlossen. Das Signal wird danach vom High-Speed-Video-Operationsverstärker IC8 verstärkt und liegt dann auf der Ausgangsbuchse. Dieser Verstärker ist schnell (2500 V/ms Anstiegsgeschwindigkeit) und auch in der Lage, niederohmige Lasten zu treiben. Der AD811 steckt übrigens nicht ohne Grund in einem DIP-Gehäuse: Nach einem tragischen Unfall im rauen Laboralltag (Überlastung) ist es viel leichter, ihn gegen einen neuen Opamp auszutauschen als dies bei einem SMD der Fall wäre.

Der Spitze-Spitze-Pegel oder die Amplitude des Ausgangssignals wird an P2 eingestellt. Aufgrund der parasitären Kapazitäten und Induktivitäten (die sich wie ein Tiefpassfilter auswirken) des Potentiometers ist die Qualität des Potis essentiell für die Qualität des Ausgangssignals. Die Grenzfrequenz des Potis ist abhängig von der Anzahl der Gänge und seinem Widerstand. Wenn der Generator vor allem für saubere Hochfrequenz-Sinuswellen verwendet wird, ist ein 10-Gang-Potentiometer mit einem Widerstand von 1 kΩ oder mehr eine gute Wahl. Sind Sie eher an Impulsen mit steilen Flanken interessiert, dann setzen Sie ein 5-Gang-Poti mit einem Widerstand von 200 Ω oder weniger ein (**Bild 2**).

Da wir schon einmal beim Thema parasitäre Filter sind, sollten Sie einen genauen Blick auf die Signalwege werfen. Alle Bauteile in den Signalwegen, beginnend am Ausgang von IC3, einschließlich der Platine und ihres Layouts, haben eine begrenzte Bandbreite und tragen zu einem gewissen Grad zur Tiefpassfilterung des Ausgangssignals bei. Deshalb ist es in Ordnung, nur ein RC-Filter erster Ordnung (nämlich R16/C18) zu verwenden. Mit einem Filter hoher Ordnung wäre ein dritter Pfad für die Dreiecksignale erforderlich, was den Entwurf viel komplexer gestaltet hätte.

Die Offsetspannung des Ausgangssignals wird durch die beiden Spannungsreferenzen IC10 und IC11 erzeugt, an Potentiometer P1 eingestellt und vom High-precision-Opamp IC9 gepuffert. Diese Gleichspannung wird am invertierenden Eingang von Opamp IC8 dem Ausgangssignal hinzugefügt.

Das Ausgangssignal wird abgeschwächt und vom Gleichspannungsoffset befreit (IC2) auch einem der analogen Eingänge des Mikrocontrollers (ADC0) zugeführt, damit der Signalpegel überwacht und auf dem Display angezeigt werden kann (siehe [2] für Messungen).

Die Schaltung rund um den Mikrocontroller IC1 bedarf kaum einer Erklärung: acht Drucktaster, ein Grafik-LCD, ein paar Entkopplungskondensatoren und einige Stiftleisten, das ist alles. Der Mikrocontroller wird von seinem internen Oszillator getaktet.

Der JTAG-Steckverbinder wird zum Programmieren und Debuggen der Software benötigt. Die serielle Schnittstelle ist auch notwendig, wenn der MCU-Bootloader zum Programmieren verwendet werden muss. Wenn JP1 gesteckt ist, startet die

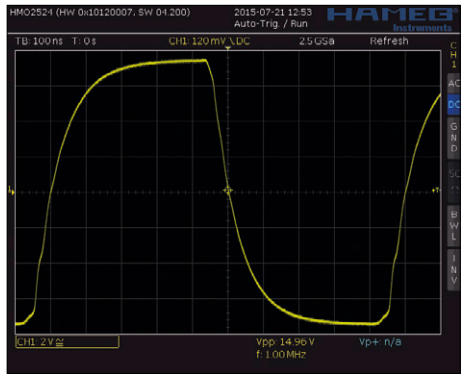


Bild 2. Eine Periode eines 1,00-MHz-Rechtecksignals mit einem Tastverhältnis von 50 %. Die Ausgangsspannung ist mit P2, einem 10-fach-Poti (1 kΩ), fast auf das Maximum eingestellt.

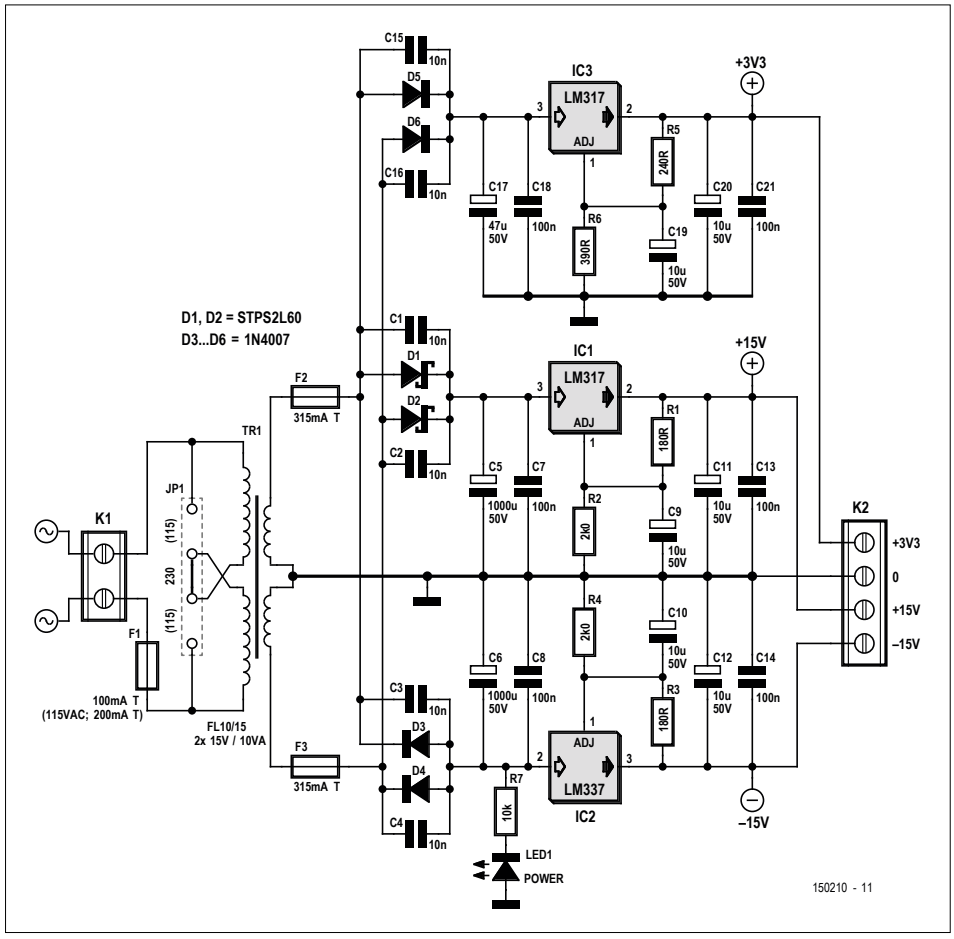


Bild 3. Das Netzteil für den Funktionsgenerator besteht aus einem Haufen LM3x7-Spannungsreglern. Mit einem Trick wurde die Stromaufnahme des 3,3-V-Teils reduziert.

MCU im Bootloader-Modus.
Die Stromversorgung (**Bild 3**) ist klassisch mit mehreren LM317 und einem LM337 aufgebaut. Zusammen bilden sie eine stabile ±15-V-Versorgung für den Analogteil und liefern saubere +3,3 V für den digitalen Rest der Schaltung. Im Gegensatz zu der Hauptplatine, die fast vollständig mit SMDs bestückt ist, sind alle Bauteile des Netzteils bedrahtet und stammen aus der Liste Elektor.Labs Preferred Parts (ELPP).

Der einzige Trafo besitzt zwei Sekundärwicklungen á 15 V/5 VA. 10 VA sind völlig ausreichend für den Betrieb des Funktionsgenerators. Um den Leistungsverlust des 3,3-V-Reglers zu minimieren, wurde ein separater Gleichrichter (D5/D6) mit einem relativ kleinen Siebkondensator verwendet (C17/C18). Die relativ hohe Brummspannung reduziert die Verlustleistung in Regler IC3 ein wenig.
Für D1 und D2 werden Schottky-Dioden verwendet. Bei maximaler Belastung wäre der Spannungsabfall an „normalen“ Dioden so hoch, dass zusammen mit der Welligkeit über C5 die Eingangsspannung von IC1 gefährlich nahe an ihren erlaubten Minimalwert sinken könnte. Bei 0,5 A beträgt die Durchlassspannung der Schottky-Diode weniger als 0,45 V, während über einer klassischen 1N4007 fast doppelt so viel abfällt. Die Hauptplatine wurde mit besonderer Sorgfalt entworfen. Das Ausgangssignal wird von jeglicher Beeinflussung durch digitale Hochge-

Stückliste Generator-Platine

Widerstände:
Alle 1%, 0,125 W, SMD 0805
R1 = 220 Ω, 1%, 0,75W, SMD 2010
R2,R22,R23,R37 = 1 k
R3 = 1 k, 1%, 0,1W, SMD 0603
R5,R6,R7,R8,R9,R10,R11,R12,R30,R31,R36 = 10 k
R13 = 6k8
R14,R15 = 200 Ω
R16,R19, R25 = 910 Ω
R17,R20 = 1k2
R18,R26 = 2k0
R21,R24 = 499 k
R27 = 220 Ω
R28 = 10 Ω
R29 = 10 k, 1%, 0,1W, SMD 0603
R32,R33 = 100 Ω, 1%, 0,75W, SMD 2010
P1 = 10 k, 2W, 10-fach Potentiometer
P2 = 1 k, 2W, 10-fach-Potentiometer

Kondensatoren:
Alle SMD 0603, wenn nicht anders angegeben
C1,C3,C4,C12,C13,C14,C16,C17,C19,C20,C26, C27,C28,C29,C31 = 100 n, 50 V, X7R
C2,C5 = 10 µ, 16 V, X7R, SMD 1206
C6,C7 = 470 n, 25 V, X7R
C11 = 1 µ, 16 V, X7R, SMD 1206
C15 = 10 n, 50 V, X7R
C18 = 10 p, 50 V, C0G/NP0
C21 = 100 p, 100 V, C0G/NP0, SMD 0805
C22,C23 = 220 µ, 25 V, radial, 3,5 mm Raster, Durchmesser 8 mm max.
C24,C30 = 100 n, 50 V, X7R, SMD 0805

Halbleiter
IC1 = ADUC7024BSTZ62, LQFP-64, programmiert*
IC2 = AD8032ARZ, SOIC-8
IC3 = AD9834BRUZ, TSSOP-20
IC4 = FXO-HC736R-75, 7 x 5 mm
IC5, IC6 = AD8042ARZ, SOIC-8
IC7 = ADG779BKSG-REEL7, 6-Lead SC-70
IC8 = AD811ANZ, DIP-8
IC9 = OP1177ARZ, SOIC-8
IC10, IC11 = ADR1581ARTZ-REEL7, SOT-23-3

Sonstiges
K1 = Fassung, 0,5 mm, Raster 1,5 mm, 10-polig, DF23C-10DS-0.5V(51), Hirose (HRS)
K2, JP1, JP2 = 1x2-poliger Pfostenverbinder, gerade, Raster 0,1’’
K3 = 2x3-poliger Pfostenverbinder, gerade, Raster 0,1’’
K4 = 1x4-poliger Pfostenverbinder, gerade, Raster 0,1’’
K5 = BNC-Buchse für Chassismontage, isoliert, gerade (50 Ω)
S1-S8 = 6 mm taktiler Taster, Stiftlänge 4,9 mm, 24V/0,05 A, SPST-NO
8-polige DIP-Fassung für IC8
Grafisches LCD (Ersatzteil für Nokia 6100)

Aluminiumgehäuse, Hammond 1455T1601, 165 x 160 x 51,5 mm
Optional: EMI/EMC-Filter zum Einbau, IEC, 250 VAC/4 A
2 Knöpfe, schwarz, 16 mm, für 0,25”-Achse
Platine 150210-1 v1.11 (www.elektor.de)

* Nicht erhältlich

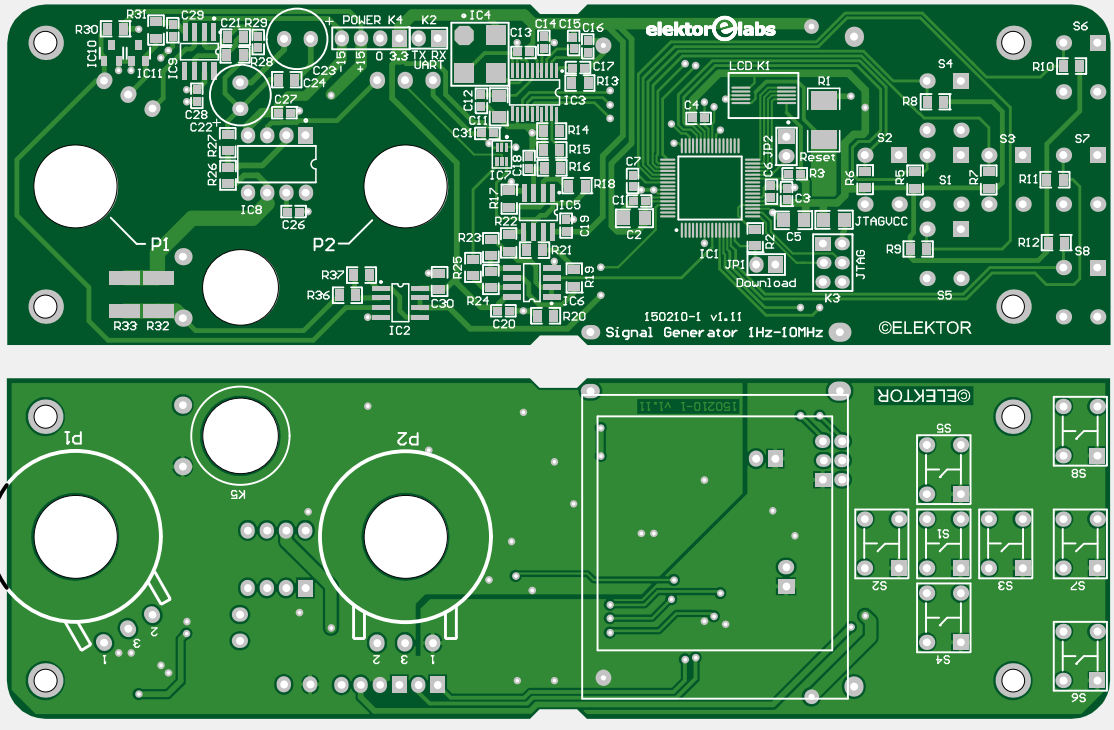


Bild 4a. Bestückungsseite der Generatorplatine (doppelseitig mit SMDs bestückt).

schwindigkeits-Steuersignale befreit. Die meisten Bauteile sind SMDs, aber dennoch sollte die Bestückung recht unproblematisch sein, wenn Sie bei den Lötarbeiten eine (Leucht-)Lupe verwenden.

Die Software

Der Quellcode wurde in C mit der bekannten µVision-IDE von Keil geschrieben. Ein Echtzeit-Betriebssystem (RTOS) wurde nicht verwendet, um die Dinge einfach zu halten.

Der Code besteht aus nur ein paar Dateien, einigen Header-Dateien mit Definitionen und Funktionsprototypen und anderen, die die Funktionen selbst enthalten. Zwei von ihnen unterstützen die beiden unterschiedlichen Grafik-LCDs. Die Datei „main.c“ enthält den gesamten Signalgenerator-Code. Die Datei „init.s“ ist in Assembler geschrieben und enthält die Funktionen, die den Mikrocontroller initialisieren.

Der LCD-Treiber basiert zum Teil auf Open-Source-Code, den ich im Internet fand, zum Teil habe ich ihn auch selber geschrieben. Da das Display nicht an einen Hardware-SPI-Port angeschlossen ist, wird das Kommunikationsprotokoll emuliert und über GPIO-Pins „gebitbangt“. Die Funktionen `WriteLcdCommand` und `WriteLcdData` kümmern sich darum. Die meiste Mühe machte der Entwurf zweier Schriften, einer großen und einer kleinen.

Die Tastatur wird durch die Main-Endlosschleife abgefragt. Um den Code kurz und IDE-unabhängig zu halten, habe ich darauf verzichtet, mit Fließkomma-Arithmetik und Mathematik-Bibliotheken zu arbeiten. Alle Quelldateien wurden im sogenannten ARM–thumb-Code (16-bit-Code) kompiliert, nicht so sehr, um die Größe der Hex-Datei zu minimieren, sondern weil der Flash-Speicher des Mikrocontrollers 16 bit breit organisiert ist und ein 16-bit-Code schneller ausgeführt werden kann. Die Gesamtgröße der Exe-Datei liegt bei 8 KB, dazu kommen 20 KB, um den Startbildschirm beim Einschalten zu erzeugen. Es waren keine Tricks nötig, um all dies in der MCU unterzubringen, der Flash-Speicher ist groß genug für den Zweck. Das Software-Archiv für dieses Projekt kann von [1] heruntergeladen werden.

Aufbau

Die beiden Platinen für dieses Projekt sind zusammen mit den Stücklisten in **Bild 4** zu sehen. Wie bereits erwähnt, ist die Montage der Hauptplatine nicht allzu

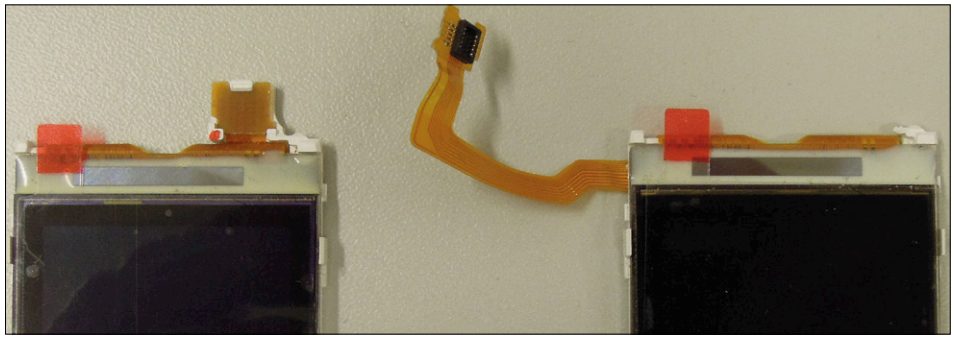


Bild 5. Links ein Display aus einem Handy einer bekannten skandinavischen Marke, rechts das gleiche Display, aber für die Montage in unserem Generator vorbereitet.

Stückliste Netzteil

Widerstände (1%, 0,6 W)
R1,R3 = 180 Ω
R2,R4 = 2,0 k
R5 = 240 Ω
R6 = 390 Ω
R7 = 10 k, 5%, 0,25 W

Kondensatoren
C1,C2,C3,C4,C15,C16 = 10 n, 50 V, Y5V, Raster 0,2’’
C5 = 1000 µ, 50 V, Raster 5 mm oder 7,5 mm, Durchmesser 16 mm
C6 = 470 µ, 50 V, Raster 5 mm oder 7,5 mm, Durchmesser 13 mm
C7,C8,C13,C14,C18,C21 = 100 n, 50 V, X7R, Raster 0,2’’
C9,C10,C11,C12,C19,C20 = 10 µ, 50 V, Raster 2 mm, Durchmesser 6,3 mm max.
C17 = 47 µF, 50 V, Raster 2,5 mm oder 3,5 mm, Durchmesser 8 mm max.

Halbleiter
D1,D2 = STPS2L60, DO-41-Gehäuse
D3,D4,D5,D6 = 1N4007, DO-41-Gehäuse
IC1,IC3 = LM317, TO-220-Gehäuse
IC2 = LM337, TO-220-Gehäuse
LED1 = LED, grün, 3 mm

Außerdem
K1 = 2-polige Platinenanschlussklemme, 7,5 mm, 500 V
K2 = 4-polige Platinenanschlussklemme, 5 mm, 250 V
TR1 = Trafo 2x115 V prim./2x15 V sek., 10VA, zum Beispiel Block FL 10/15
F1 = Sicherung, 100 mA träge, 250 V, 20x5mm
F2, F3 = Sicherung, 315 mA träge, 250 V, 20x5mm
Sicherungshalter für F1, F2, F3, 20x5mm, 500 V, 10 A
plus Abdeckungen für Sicherungshalter
JP1 = Drahtbrücke
Platine 150210-2 v1.1 (www.elektor.de)

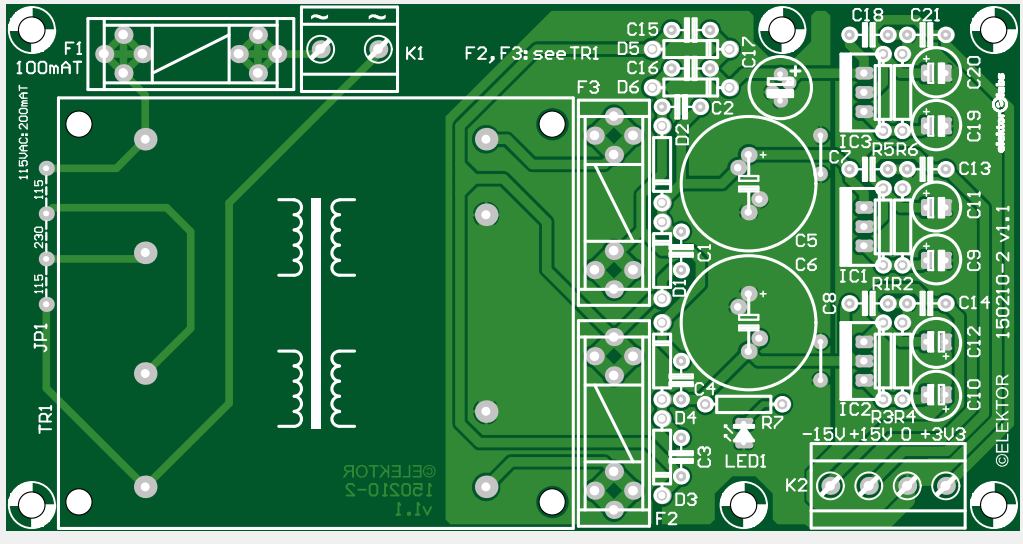


Bild 4b. Bestückungsseite des Netzteils (einseitig für bedrahtete Bauteile).

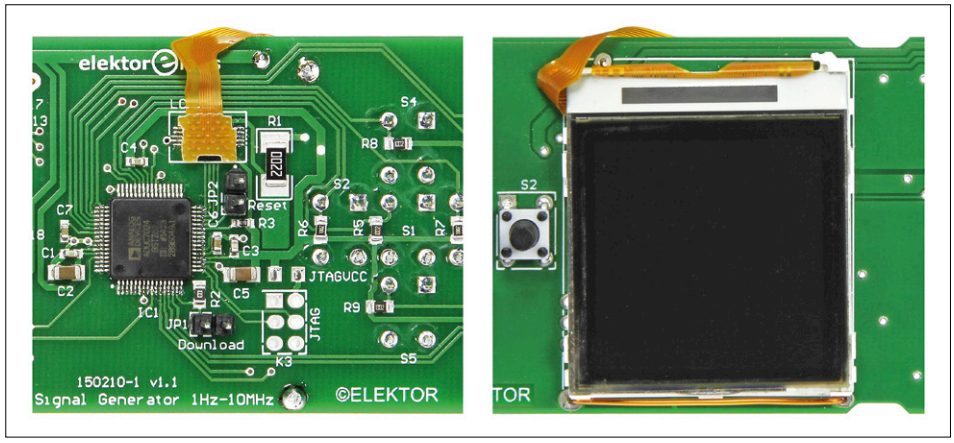


Bild 6. So wird das Display mit doppelseitigem Klebeband auf der Hauptplatine befestigt.

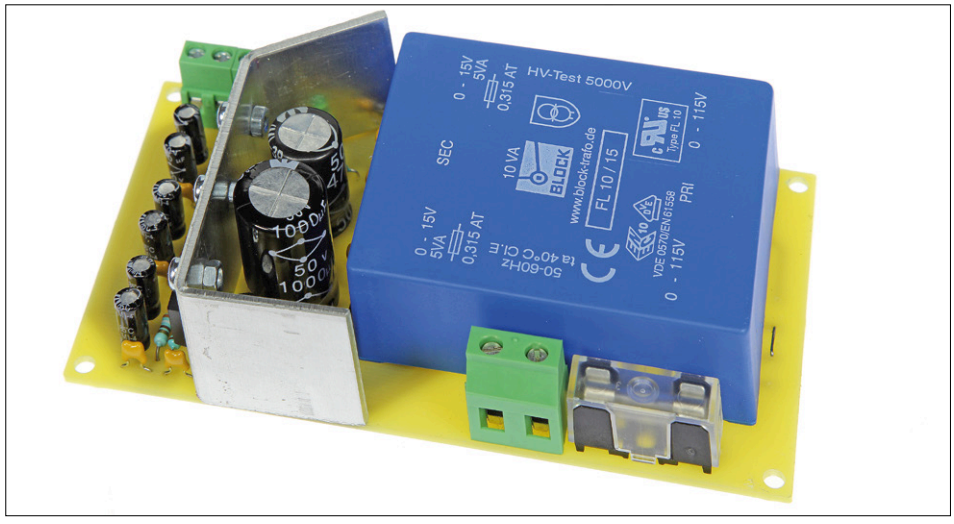


Bild 7. Die Netzteilplatine mit selbstgebasteltem Kühlkörper. Beachten Sie die Drahtverbindung rechts neben dem Trafo zur Auswahl der Netzspannung (230 V in diesem Fall).

Bild 8. Der Prototyp, den Jan Visser im Elektor-Labor mit ~~einer 100.000-Euro-CNC-Fräse~~ Bohrmaschine, Laubsäge, Präzision und Geduld hergestellt hat.

schwierig. Übrigens: Die Drucktaster und das LCD werden auf der Unterseite der Platine angebracht.

Beim LCD muss das flexible Flachbandkabel vorsichtig aus dem Kunststoffträger (**Bild 5**) gelöst und über den Rand der Platine gefaltet werden, damit es den Verbinder K1 erreichen kann (**Bild 6**). Das Display wird mit doppelseitigem Klebeband auf der Platinenrückseite fixiert.

Die beiden Potentiometer werden so durch die Platine gesteckt, dass die Achse aus der Unterseite hervorragt. Im Elektor-Labor verwenden wir bei der Befestigung aus einem alten Fahrradschlauch selbstgemachte Gummischeiben, damit die Potis fest sitzen und nicht verrutschen können. 10 mm Lochdurchmesser und ein Außendurchmesser von etwas weniger als 22 mm sind ok. Bei P2 verhindert die Gummischeibe auch Schaden an einer Kupferbahn auf der Oberseite der Platine.

Beim Netzteil haben wir einen Trafo mit zwei Primärwicklungen verwendet, damit unsere überseeischen Kollegen auch etwas von dem Funktionsgenerator haben. Bei 230 V ist deswegen eine Drahtbrücke bei JP1 (die mittlere der drei gestrichelten Linien) erforderlich, für 115 V Netzspannung werden zwei Drahtbrücken an den anderen Linien angebracht. Installieren Sie nie alle drei Drahtbrücken! Vergessen Sie auch nicht, die korrekte Primärsicherung einzusetzen: 100 mA (träge) für 230 VAC und 200 mA (träge) für 115 VAC.

Die drei Spannungsregler benötigen einen Kühlkörper, den wir aus einem 2 mm dicken Aluminiumblech (**Bild 7**) hergestellt haben. Vergessen Sie nicht die elektrische Isolierung (Glimmerscheibe und Kunststoffbuchse) der Regler. Die M3-Schrauben sollten etwa 6 mm lang sein. Eine U-Scheibe zwischen dem Kopf der Schraube und der Kunststoffbuchse wird empfohlen. Oft ist die Kunststoffbuchse ein bisschen zu lang, schneiden Sie sie mit einem scharfen Messer auf die richtige Länge, bevor Sie die Regler montieren.

Wenn Sie sich entscheiden, das gleiche Gehäuse wie wir (**Bild 8**) zu verwenden, dann können Sie eine Mechanik-Zeichnung von der Elektor-Website herunterladen [1], die Ihnen Details für das Fräsen der Front und Rückseite zeigt.

Die BNC-Buchse K5 ist isoliert und wird auf der Frontplatte montiert. Nach der Befestigung der Hauptplatine an der Frontplatte (wir haben die Schrauben an der Rückseite der Frontplatte einfach

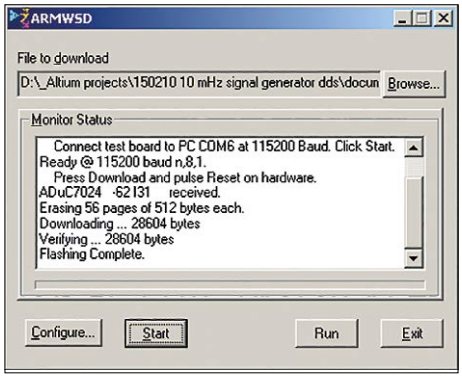


Bild 9. Der Screenshot zeigt das Firmware-Programmiertool ARMWSD in Aktion.

festgeklebt) wird die BNC-Buchse mit kurzen Drähtchen mit der Platine verbunden.

Programmierung

Bevor der Funktionsgenerator funktionieren kann, muss er natürlich mit der richtigen Firmware programmiert werden. Wegen der beiden möglichen LCD-Konfigurationen gibt es auch zwei Firmware-Versionen auf unserer Website [1]. Da man nicht vorhersagen kann, um welchen Anzeigentyp es sich handelt, muss man es halt ausprobieren.

Es gibt zwei Möglichkeiten zur Programmierung der MCU: JTAG oder Bootloader. Bei der ersten Option ist ein JTAG-Adapter erforderlich, dafür erhalten Sie auch direkt ein Debug-Interface. Bekannte JTAG-Adapter sind J-Link von Segger und ULINK von Keil. Die Standard-JTAG-Schnittstelle besitzt 20 Pins, aber sie funktioniert auch mit nur sechs Pins (K3).

Die zweite Option ist die serielle Portschnittstelle (K2), allerdings ausschließlich zum Programmieren. Debugging ist über diese Schnittstelle nicht möglich. Ein passendes Kabel kann von Analog Devices bezogen werden, aber es ist sehr einfach, ein serielles TTL-to-USB-Kabel selbst zu machen.

Ein serielles Software-Programmiertool (ARMWSD.exe) steht kostenlos auf der Website von Analog Devices zur Verfügung. Wählen Sie zunächst den entsprechenden seriellen USB-Port und laden Sie die Hex-Datei. Das Programm befiehlt dann *„Press Download and pulse Reset on hardware“* (**Bild 9**). Aus diesem Grund tragen die beiden Jumper JP1 und JP2 auf der Platine die Labels „Download“ und „Reset“. Wenn ein Jumper beim Einschalten auf „Download“ steckt, arbeitet der Mikrocontroller im Bootloader-Modus und das Display bleibt dunkel. Vergessen Sie also nicht, den Jumper nach der Programmierung zu entfernen.

(150210)

Produkte im ELEKTOR-Shop

- 150210-1 – Hauptplatine, unbestückt
- 150210-2 – Netzteilplatine, unbestückt
- 150210-91 – Hauptplatine, bestückt

Den Mikrocontroller IC1 können wir leider weder leer noch programmiert liefern!

Weblinks

[1] www.elektormagazine.de/150210

[2] www.elektor-labs.com/150210

Benutzerhandbuch

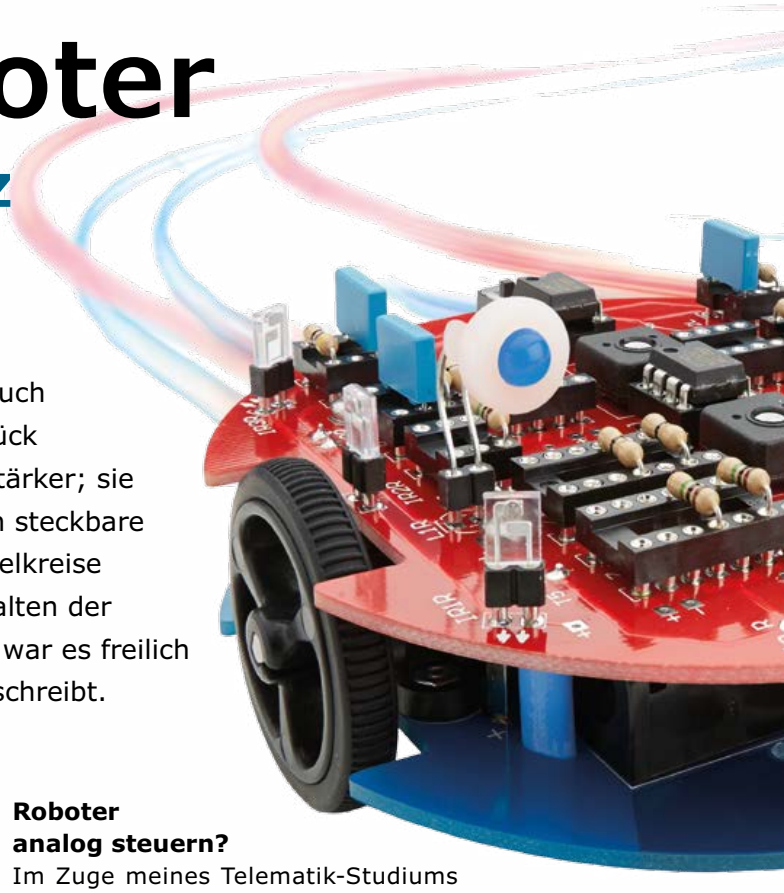
PB	Funktion	PB	Funktion
S1	Setzen	S5	–Minus
S2	Rechts	S6	Modus
S3	Links	S7	Sweep
S4	+Plus	S8	Kalibrierung

- **Wellenform** – *Modus* drücken, um zwischen Sinus, Rechteck und Dreieck umzuschalten.
- **Tastverhältnise** – Das Tastverhältnis kann nur beim Rechteck geändert werden. Dazu *Modus* drücken, um das Rechtecksignal zu wählen. Das Tastverhältnis wird unten im Display angezeigt und kann mit *Plus* und *Minus* eingestellt werden. Die Digits dürfen dabei **nicht** blinken!
- **Frequenz** – *Setzen* drücken. Eine Ziffer beginnt zu blinken. Mit den Tasten *Plus und* Minus wird der gewünschte Wert eingestellt. Mit *Links* und *Rechts* navigiert man zwischen den Ziffern. Wenn eine Ziffer richtig eingestellt ist, drückt man *Setzen*.
- **Amplitude** – Einstellen an P2. Beachten Sie, dass das Einstellen der Amplitude Auswirkungen auf die Offsetspannung hat. Ziehen Sie [2] für detaillierte Messungen zu Rate.
- **DC-Offset** – Einstellen an P1.
- **Frequenz-Sweep** – *Sweep* drücken, um das Menü zu öffnen. Die niedrigste Ziffer der Startfrequenz blinkt. Mit den Tasten *Plus und* Minus wird der gewünschte Wert der blinkenden Ziffer eingestellt. Mit Links und Rechts navigiert man zwischen den Ziffern. Wenn eine Ziffer richtig eingestellt ist, drückt man *Setzen*. Auf ähnliche Weise bestimmt man die Stoppfrequenz, die Sweep-Zeit („msec“) und den Sweep-Modus (logarithmisch oder linear). Mit einem Druck auf *Setzen* startet man das Sweep-Signal, was durch „sweep run“ angezeigt wird. *Set* erneut drücken, um das Sweepen zu beenden. Jetzt kann man neue Werte einstellen oder durch einen Druck auf *Setzen* das Sweep-Menü verlassen.
- **Kontrast** – Auf *Kalibrieren* drücken, um das Menü zu öffnen, in dem der Kontrast des LCDs eingestellt wird. Mit *Setzen* navigiert man zu der Kontrast-Option, mit *Plus* und *Minus* stellt man den gewünschten Wert ein. Um in das Hauptmenü zurückzukehren, drückt man erneut *Kalibrieren*.
- **Kalibrieren der Spannungspegel** – Schließen Sie ein Oszilloskop am Generator-Ausgang an und stellen Sie dessen Ausgangspegel auf 5 V_{ss} ein. Mit einem Druck auf *Kalibrierung* wird das Menü geöffnet, mit *Measurements* der Kalibriervorgang gestartet. Wenn Sie sich jetzt vertippen, ist die einzige Möglichkeit, den Netzstecker zu ziehen. Drehen Sie an P1, um das Ausgangssignal auf das Minimum von 0,00 V einzustellen und drücken Sie dann *Setzen*. Dann stellen Sie mit P1 den maximalen Pegel von 12,00 V ein und drücken erneut auf *Setzen*. Eine Mitteilung erscheint, dass die Kalibrierung vollständig ist und mit einem erneuten Druck auf *Kalibrierung* abgeschlossen werden kann.
- **Kalibrieren der Frequenz** – Schließen Sie ein genaues Frequenzmeter am Generator-Ausgang an. Mit einem Druck auf *Kalibrierung* wird das Kalibrierungs-Menü geöffnet, mit *Frequency* der Kalibriervorgang gestartet. Wenn Sie sich jetzt vertippen, müssen Sie wieder den Netzstecker ziehen. Stellen Sie die Ausgangsfrequenz mit den Tasten *Plus* und *Minus* so ein, dass die Ausgangsfrequenz 100.000 Hz beträgt und drücken Sie dann *Setzen.* Eine Mitteilung erscheint, dass die Kalibrierung vollständig ist und mit einem erneuten Druck auf *Kalibrierung* abgeschlossen werden kann.

Analoge Roboter

Von der Idee zum Bausatz

Von **Tino Werner**



Der Roboter-Bausatz „tibo“ soll vor allem Jugendliche für Technik und das Selbermachen begeistern. Doch auch ältere Semester kommen auf ihre Kosten: Das Herzstück dieser ungewöhnlichen Roboter ist ein Operationsverstärker; sie benötigen weder Mikrocontroller noch Software. Durch steckbare Widerstände und Kondensatoren werden einfache Regelkreise aufgebaut, welche das – erstaunlich komplexe – Verhalten der Roboter festlegen. Von der Idee zum fertigen Bausatz war es freilich ein weiter und steiniger Weg, wie der Erfinder hier beschreibt.

Autonom agierende Maschinen zu bauen, im Besonderen mobile Roboter, die eigenständig auf ihre Umwelt reagieren, begeistert mich schon seit meiner Kindheit. Damals baute ich meine Projekte mit Fischertechnik, doch als ich 1993 eine Roboterspinne konstruieren wollte, die sich mit geringem mechatronischem Aufwand möglichst natürlich bewegt, stieß ich in punkto Stabilität schnell an die Grenzen dieses Baukastensystems. Daher entschied ich mich, den Rumpf der Spinne aus Holz zu fräsen und die Beine aus Aluminiumkomponenten aufzubauen. Auch das Getriebe für die Antriebe war eine Eigenkonstruktion.

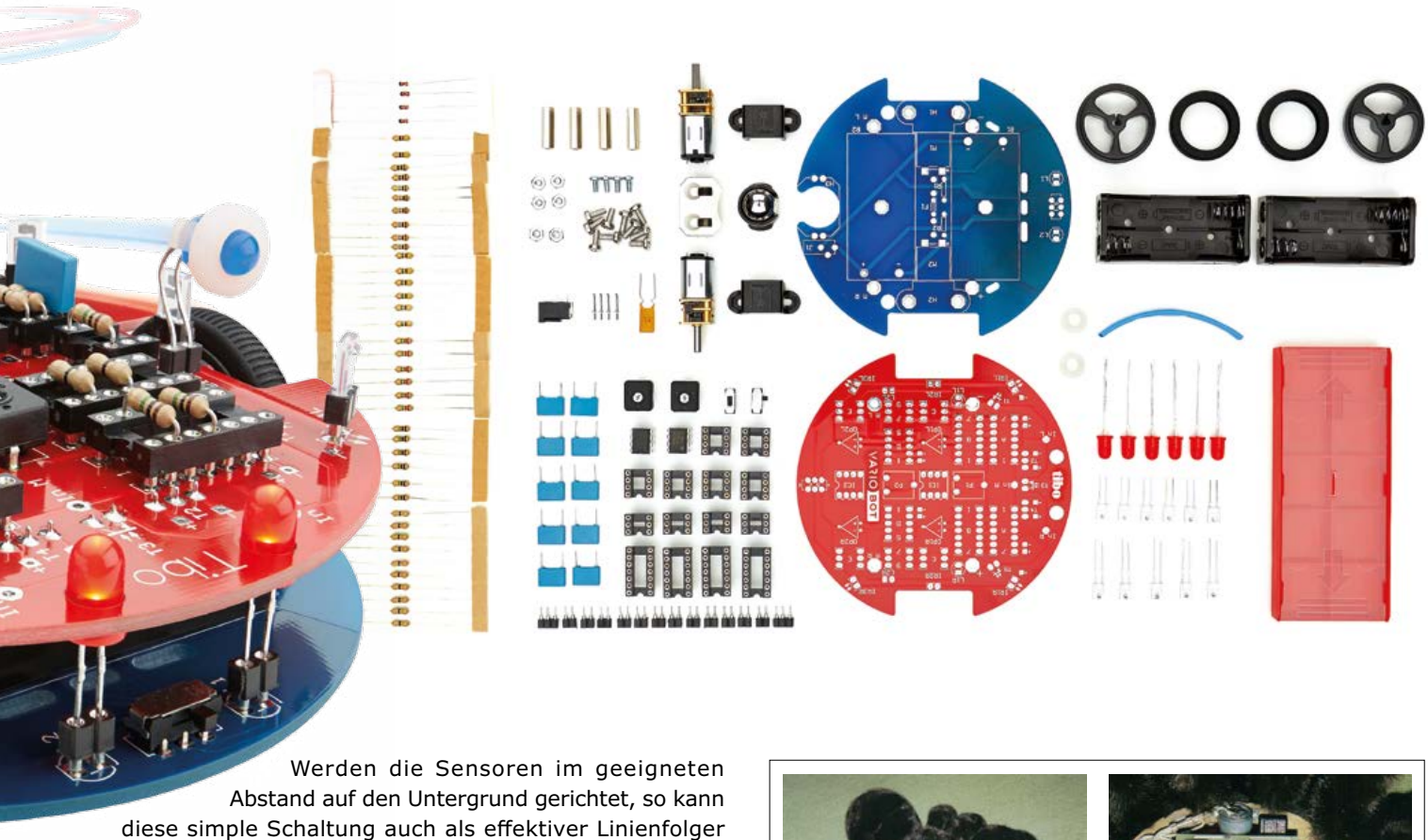
Ich hatte zwar Grundwissen über Mikrocontroller, aber da es damals noch keinen Arduino, keinen Raspberry Pi oder dergleichen gab, schien mir eine festverdrahtete Logikschaltung einfacher umzusetzen. Die Steuereinheit konnte ich an meiner Schule im Rahmen einer Projektarbeit entwickeln. Mit Hilfe eines Binärzählers generierte ich die pseudo-zufälligen Bewegungen der Spinne. Ohne es vorhersagen zu können, änderte sie die Laufrichtung oder blieb für unterschiedlich lange Zeit stehen. Zudem bewegte sie ihre Fühler und den Hinterleib – auch dies scheinbar zufällig. Eine Sensorik mit Infrarotlichtschranken spürte Hindernisse auf: Bevor die Spinne mit einem Gegenstand kollidierte, blieb sie kurz stehen, trat zurück und drehte im Stand, um danach in eine andere Richtung weiter zu laufen. Durch die Kombination von Funktion und „zufälligen Entscheidungen“ entstand beinahe der Eindruck, als hätte man es mit einem intelligenten Lebewesen zu tun. Meine Mutter half mir dabei, ein Fell zu nähen, das dem Roboter das Aussehen einer riesigen Vogelspinne verlieh. Das Modell (**Bild 1**) hatte eine Spannweite von etwa 60 cm!
Bei dieser und den folgenden Entwicklungen galt mein Interesse nicht so sehr dem Erfüllen von klar definierten Funktionen, wie es in der Robotik üblicherweise der Fall ist. Vielmehr beschäftigte mich die Frage, wann wir Menschen das biologische Verhalten eines Wesens als lebendig empfinden und wie ein solches Verhalten technisch effizient nachgebildet werden kann.

Roboter analog steuern?

Im Zuge meines Telematik-Studiums lernte ich 1999 die Braitenberg-Vehikel [1] kennen. Diese Experimente beschreiben, wie bei einfachen Vehikeln durch direkte Verbindungen zwischen Sensoren und Antriebsmotoren unerwartet komplexe Verhaltensweisen entstehen können. Ich war fasziniert von der Einfachheit und der Effektivität dieses Ansatzes, so dass ich damals begann, zahlreiche Varianten der Sensorintegration und analogen Signalverarbeitung anhand von simplen fahrenden Robotern mit Lichtsensoren zu entwickeln und zu optimieren.

Die wesentliche Innovation dabei war das Zusammenfassen von in Reihe liegenden Sensoren zu Sensorgruppen. Diese liefern nicht wie üblich absolute Messwerte innerhalb eines begrenzten Arbeitsbereichs, man erhält vielmehr aus n (2 oder mehr) Lichtsensoren bis zu n - 1 relative Helligkeitswerte, die zur Navigation und Hinderniserkennung herangezogen werden können. Auch unser visuelles System verfügt über eine relative Helligkeitswahrnehmung: Die Wahrnehmung eines Objektes bezieht sich immer auf die Helligkeit der Umgebung, was durch zahlreiche optische Täuschungen eindrucksvoll nachvollzogen werden kann.

Die einfache Transistorschaltung in **Bild 2** zeigt das Prinzip einer solchen Steuerung für einen Roboter mit Differenzialantrieb, bei der Sensor S1 nach rechts und S2 nach links gerichtet ist. Ein solcher Roboter navigiert sensibel in Abhängigkeit der Helligkeitsverhältnisse der Umgebung, ohne dabei auf einen bestimmten Helligkeitsbereich beschränkt zu sein. Werden dabei die Sensoren so angebracht, dass kein direktes Licht auf sie trifft, so weicht das Gefährt auch Hindernissen aus, da diese den Lichteinfall des entsprechenden Sensors reduzieren. Bringt man zwischen S1 und S2 einen weiteren Sensor an, so stoppt das Vehikel zuverlässig vor frontalen Hindernissen.



Werden die Sensoren im geeigneten Abstand auf den Untergrund gerichtet, so kann diese simple Schaltung auch als effektiver Linienfolger eingesetzt werden. Soll das Vehikel einer dunklen Linie folgen, müssen lediglich die Sensoren S1 und S2 vertauscht werden. Für einen Nachbau dieser Schaltung eignen sich Darlington-Transistoren, die eine ausreichend große Verstärkung aufweisen, um kleine Gleichstrommotoren direkt ansteuern zu können.

Opamps als Gehirnzellen

Ich hatte mich jedoch von Beginn an dazu entschlossen, Operationsverstärker (Opamps) als analoge „Gehirnzellen“ für meine autonomen Gefährten zu nutzen. Leistungs-Opamps können nämlich nicht nur hochohmige analoge Sensorsignale verarbeiten, sondern auch Motoren bidirektional ansteuern. Damals verwendete ich den Dual-Power-Opamp L272, der einen Ausgangsstrom von bis zu 1 A liefert und mit einer Versorgungsspannung von nur 4 V auskommt. Bei dieser geringen Spannung ist jedoch der relativ eingeschränkte Aussteuerbereich von Nachteil.

Eine „direkte Verbindung“ zwischen Sensoren und Motoren ist im einfachsten Fall als Impedanzwandler (Spannungsfolger) realisierbar, bei dem das Ausgangssignal auf den invertierenden Eingang zurückgeführt wird. Bei einer symmetrischen Spannungsversorgung werden die Motoren mit jeweils nur einem Opamp in beide Richtungen ausgesteuert, wodurch diese Braitenberg-Vehikel auch in der Lage sind, rückwärts zu fahren.

Ein Opamp kann aber nicht nur linear verstärken, sondern auch invertieren, (gewichtet) addieren oder subtrahieren, logarithmieren oder integrieren, mehrere Signale als Komparator vergleichen oder als Schwellenwertschalter dienen. So wie diese Rechenoperationen bei den früheren Analogrechnern per Steckfeld programmiert wurden, nutzte auch ich von Beginn an Steckplätze in Form von IC-Fassungen, um die Opamps variabel zu beschalten und so die Funktionen und Parameter der Steuerschaltungen anzupassen. In ihrer ersten und einfachsten Ausführung bestanden diese kleinen Vehikel lediglich aus zwei Antrieben, einem kleinen schwenkbaren Stützrad, den Akkus und einer DIP-Fassung, auf dem ein Dual-Opamp sowie ein paar austauschbare Widerstände oder Kondensato-

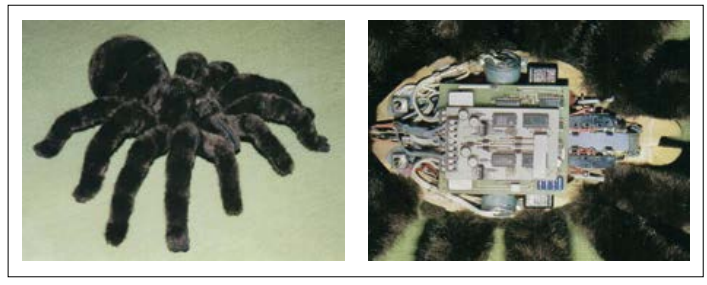


Bild 1. Aussehen und Innenleben meiner ersten Roboterspinne.

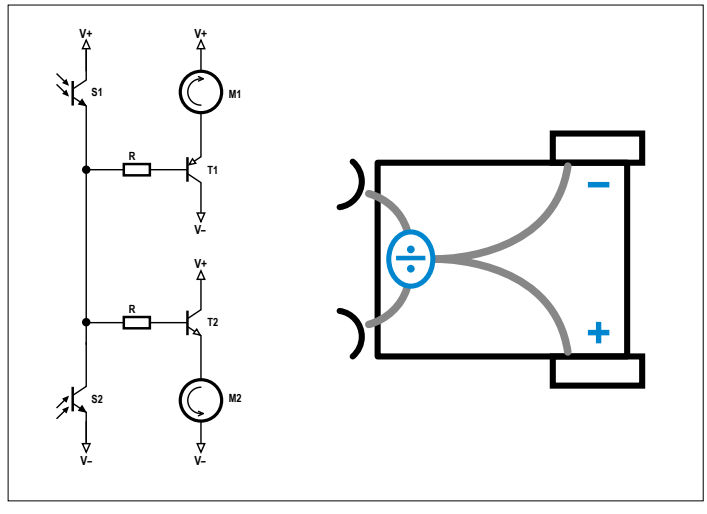


Bild 2. Einfache Transistorschaltung für einen Roboter mit Differenzialantrieb.

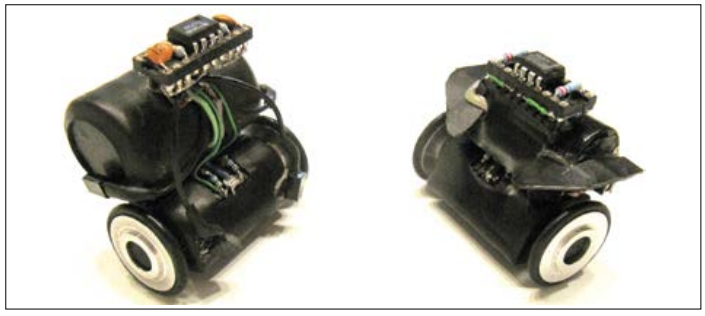


Bild 3. Die ersten einfachen analogen Robotermodelle.

ren Platz fanden (**Bild 3**).

Herausfordernde Serienentwicklung

Als ich mich 2013, nach acht Jahren in der Forschung, dazu entschlossen hatte, mich mit meinem Hobby selbstständig zu machen, die Ideen zu patentieren und eine neue Generation von Robotern zu etablieren, hatte ich kaum eine Vorstellung davon, was auf mich zukommen würde. Eine der größten Herausforderungen war es, die analoge Experimentierplattform so vielseitig und gleichzeitig so einfach wie möglich zu gestalten. Nach langem Tüfteln gelang es schließlich, das symmetrische Stecksystem auf der roten Steuerplatine (**Bild 4**) übersichtlich abzubilden, sodass der jeweilige Schaltplan mit dem entsprechenden Layout des Top-Layers übereinstimmt und das Experimentieren auch ohne Anleitung möglich wurde. Auf einer zweiten, darunter liegenden blauen Platine (rechts) befinden sich die Stromversorgung und die Antriebe des tibo-Roboters. Auch die Entwicklung einer gut verständlichen Aufbauanleitung und eines neuen didaktischen Konzepts für die Experimentieranleitung von tibo nahm enorm viel Zeit in Anspruch. Hierbei werden die Zusammenhänge von Sensorsignalen und Ausgangsgrößen mit Balkendiagrammen veranschaulicht, über die ein grundlegendes funktionelles Verständnis auch ohne Formeln ermöglicht wird. **Bild 5** zeigt ein Beispiel aus der Experimentieranleitung für einen Linienfolger.

Bei der Auswahl der Bauteile legte ich großen Wert auf einen robusten, zuverlässigen und komfortablen Aufbau des Bausatzes. Kabelbinder oder Klebeverbindungen, wie sie bei billigen Bausätzen oft verwendet werden, kamen für mich nicht in Frage. Den Aufwand für die Auswahl und den kosteneffizienten Einkauf aller Bauteile bei den verschiedenen Händlern hatte ich dabei aber völlig unterschätzt.

Allein die Recherche und Beschaffung eines geeigneten Operationsverstärkers dauerten viele Wochen. Da für den Lötbausatz nur ein IC mit DIP-Gehäuse in Frage kam, war die Auswahl bereits stark eingeschränkt. Der Opamp sollte nicht nur die Motoren ansteuern, sondern auch die Infrarot-LEDs. Um diese nicht zu überlasten, musste deren Strom begrenzt werden. Da tibo seine Energie aus vier AAA-Zellen (Akkus oder Batterien) mit einer variablen Versorgungsspannung von 4...6 V (symmetrisch ±2 V bis ±3 V) beziehen sollte, kam eine einfache

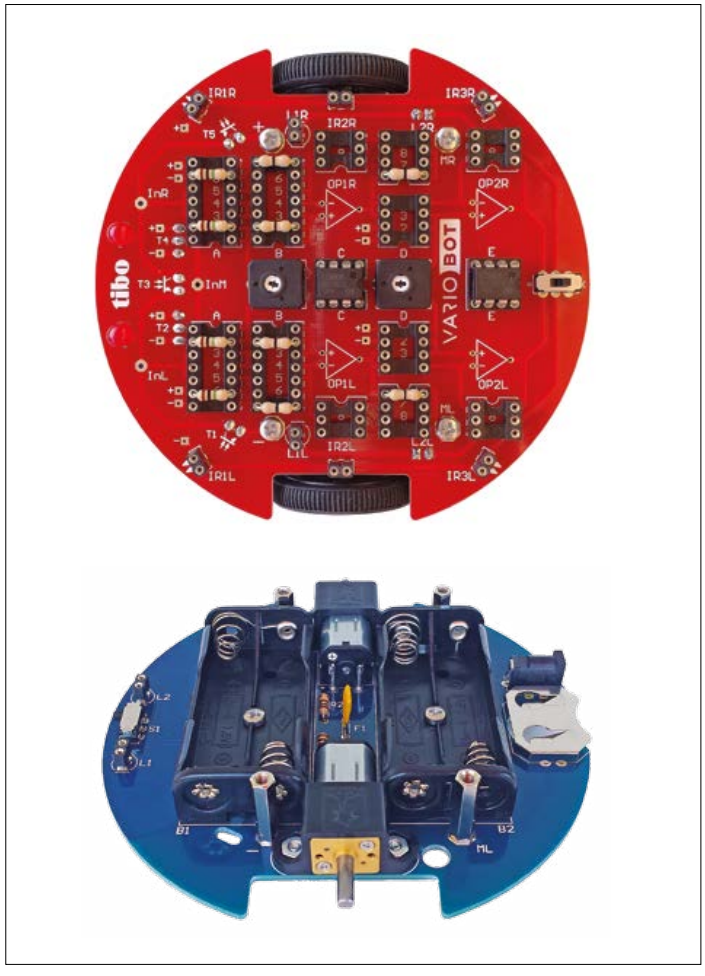


Bild 4. Steuerplatine (oben) und Leistungsplatine (unten) des tibo-Roboters.

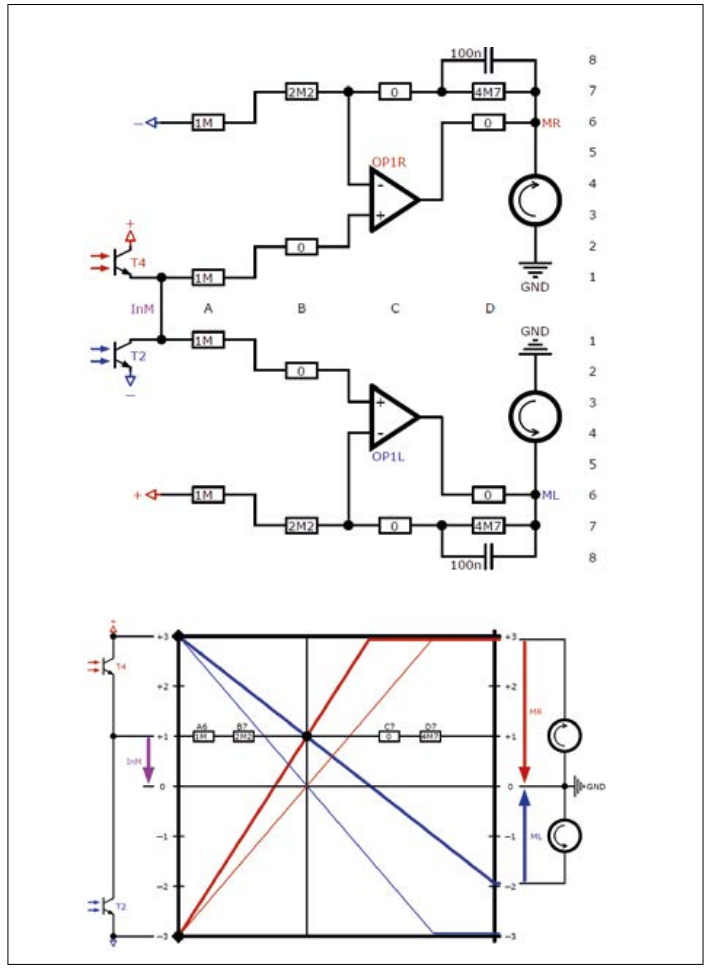


Bild 5. Ein Beispiel aus der Experimentieranleitung.

Lösung mit Serienwiderstand nicht in Frage. Um die Schaltung des Lötbausatzes möglichst einfach und überschaubar zu halten, sollten auch keine zusätzlichen ICs wie Spannungsregler verwendet werden. Die Auswahl fiel schließlich auf den Dual-Opamp TS922, der den großen Vorteil hat, über eine interne Strombegrenzung zu verfügen. Mit diesem Opamp können die IR-LEDs (IR908-7C-F) direkt angesteuert werden, ohne sie zu überlasten. Zudem macht sich die Strombegrenzung auch beim Betrieb der Motoren bezahlt. Hängt tibo etwa unbemerkt an einem Hindernis fest oder werden die Räder blockiert, so nehmen die Motoren dabei keinen Schaden.

Beschaffung

Erst nachdem die Entwicklung abgeschlossen und der Großteil der Opamps bereits bestellt war, erfuhr ich, dass der TS922IN im DIP8-Gehäuse vom Hersteller STMicroelectronics abgekündigt worden war. Die noch fehlenden 300 Stück für die erste Serie zu besorgen gestaltete sich als schier endlose Odyssee. Die bestellten ICs aus China erwiesen sich nämlich als Fälschung und lieferten statt 80 mA nur etwa 12 mA. Ich ließ mir an die 30 Angebote mit genauer Typenbezeichnung und Fotos der Bauteile schicken und machte über zehn Musterbestellungen - jedoch ohne Erfolg. Alle Muster hatten denselben Mangel. Nach langer Suche konnte die deutsche Firma TronicPool, die auf die Beschaffung von abgekündigten Bauteilen spezialisiert ist, die benötigte Menge schließlich doch noch besorgen.

Die Beschaffung der Metallgetriebemotoren (PGM-12F) gestaltete sich ähnlich langwierig. Diese mit 25 x 12 x 10 mm^3 sehr kompakten Motoren sind in einer großen Auswahl an Drehzahlen, Übersetzungsverhältnissen und Leistungen verfügbar. Sie sind zumeist für eine Spannung von 6 V ausgelegt, aber auch 3-V-Motoren (wie im tibo) sind erhältlich. Eine Bestellung bei einem europäischen Händler kam aufgrund der hohen Preise nicht in Betracht. Bei haushaltsüblichen Mengen dieser Motoren muss man mit etwa 15 € rechnen. Auf eBay, Aliexpress und Alibaba findet man zahlreiche Händler, die diese Motoren preiswerter anbieten. Hier ist jedoch Vorsicht geboten. Selbst eine zufriedenstellende Mustersendung ist keine Garantie dafür, dass man hinterher die gleiche Qualität erhält. Verzichtet man auf hohe Versandkosten, so muss man mit Lieferzeiten von einigen Wochen rechnen. Für den Roboterbausatz hatte ich sieben Motorentypen getestet. Die entscheidenden Kriterien waren eine geringe Stromaufnahme, eine geringe Anlaufspannung sowie ein ruhiges Laufverhalten. Einige der Motoren scheiterten jedoch bereits daran, dass ihre Welle eierte oder sie gar nicht erst anliefen. Für private Anwendungen und kleine Stückzahlen empfehle ich daher, diese Motoren bei europäischen Händlern zu beziehen, um kein Risiko hinsichtlich der Qualität einzugehen und sich Ärger mit Reklamationen zu ersparen.

Noch mehr Probleme...

Wer mit dem Gedanken spielt, aus seinem eigenen Hardwareprojekt ein Produkt zu entwickeln und dieses auch selbst zu vermarkten, dem sollte bewusst sein, dass es neben den technischen Aspekten noch eine Vielzahl an Herausforderungen gibt, die einiges an Zeit und Geld kosten.

Ich hatte mich im Vorfeld dazu entschieden, sowohl die Steuerung als auch die Kinematik der Roboterspinne zu patentieren. Dabei war es sehr aufschlussreich, ähnliche alternative Verfahren zu analysieren und den Kern der eigenen Erfindung herauszuarbeiten. Auch wenn ein Schutzrecht natürlich keine Voraussetzung für die Vermarktung eines innovativen Produktes ist, so rate ich in jedem Fall dazu, eine Patentrecherche anzustellen, um nicht Gefahr zu laufen, die Schutzrechte anderer zu verletzen und eine Klage zu riskieren. Das gleiche trifft auf eine eventuelle Markenanmeldung zu. Für beides gibt es Förderinitiativen wie SIGNO [2], die eine erstmalige Schutzrechtsanmeldung unterstützen.

Schließlich kommt man nicht darum herum, sich mit Gesetzesauflagen wie der CE-Prüfung, der Verpackungsverordnung und dem ElektroG (Gesetz über das Inverkehrbringen, die Rücknahme und die umweltverträgliche Entsorgung von Elektro- und Elektronikgeräten) auseinanderzusetzen. Im Zweifelsfall ist es ratsam, professionelle Hilfe in Anspruch zu nehmen, auch wenn man zunächst meint, alles selbst machen zu können. Um Erfahrungen zu sammeln, hatte ich beispielsweise die Konfektionierung der ersten 100 Bausätze selbst durchgeführt, jedoch dabei schnell gemerkt, dass man solche Arbeiten, die man nicht gut und gerne macht, besser an Profis abgeben sollte. Die weiteren Bausätze werden seitdem von einer ortsnahen Behindertenwerkstatt zuverlässig verpackt und geprüft.

Für viele Erfinder wird es möglicherweise der bessere Weg sein, einen etablierten Hersteller von seiner Idee zu überzeugen und die Zusammenarbeit etwa über eine Lizenzierung zu regeln. Dabei trägt man nicht das Risiko, das eine eigene Firmengründung mit sich bringt. Ohne sich um den Vertrieb der Produkte kümmern zu müssen, der einen ansonsten doch sehr beansprucht, kann man sich besser auf das nächste Projekt stürzen! ⏮

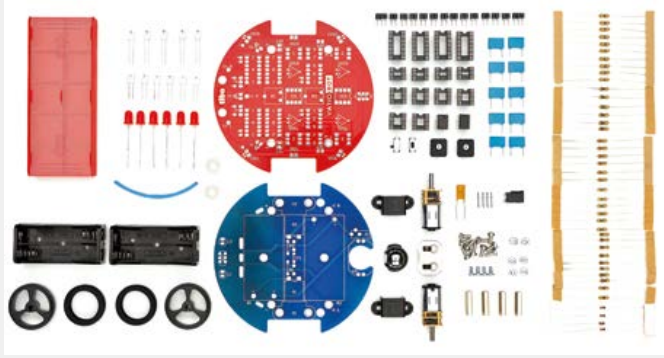
(140510)

[1] Braitenberg, Valentino: Vehicles: Experiments in Synthetic Psychology (MIT Press 1986)

[2] www.signo-deutschland.de

tibo im Elektor-Shop

Der Elektor-Shop ist eine hervorragende Plattform, um solch interessante Entwicklungen wie die tibo-Roboter weltweit zu vertreiben. Der Bausatz enthält alle erforderlichen elektronischen und mechanischen Komponenten, die zum Aufbau eines tibo-Roboters nötig sind. Unter www.elektor.de/tibo-robot-kit finden Sie nicht nur nähere Informationen zum Bausatz und zur Bestellung, sondern auch eine deutschsprachige, 40-seitige Bau- und Experimentieranleitung sowie ein Video, das den Aufbau des Roboters zeigt.



Der Bausatz kostet 89,95 €, Elektor-Mitglieder genießen einen Rabatt von 10%!

T-Board mit ARM

Cortex-M0+ 32-bit-Power

Idee/Entwicklung: **Viacheslav Gromov** (Deutschland)

Text/Co-Entwicklung: **Andrew Retallack** (Südafrika)

Platinenlayout (verantwortlich): **Luc Lemmens** (Elektor-Labor)

Die vor einem guten Jahr vorgestellte Familie der T-Boards, mittlerweile eine echte Institution, erhält Zuwachs: Wir stellen das jüngste (und stärkste) Mitglied der Familie vor – erkunden Sie die Welt der 32-bit-ARM-Controller mit dem ARMierten T-Board!

Obwohl 8-bit-MCUs die Brot- und Butter-Controller aller Hobby-Entwickler und „Maker"-Communities sind und bleiben werden, steigt doch das Interesse an 16- und 32-bit-Mikrocontrollern rasant. Ohne Zweifel ist die beliebteste Architektur für Einsteiger der ARM-Cortex-M, ein Kern, der in den Controllern fast aller namhaften Hersteller steckt. Die Popularität ist zum Teil auf die sinkenden Preise zurückzuführen (kleine Cortex-M0s sind schon für unter 1 € zu haben), zum anderen Teil auf die Forderung nach mehr Rechenleistung, Funktionen und I/O-Pins im Zuge der IoT-Entwicklung. Auch bei Elektor tragen wir dieser Entwicklung Rechnung, last but not least mit dem Kurs *ARM-Controller für Einsteiger*.

Zwar gibt es etliche ARM-Cortex-Entwicklungsboards, doch wir glauben, dass das T-Board eine perfekte Plattform für einen ARM-Cortex-M0+-Mikrocontroller darstellt. ARM-Chips gibt es in kleinen Gehäusen mit geringem Pin-Abstand und sie benötigen zudem eine Reihe von peripheren Komponenten, so dass sie nicht so „benutzerfreundlich" aus dem Regal kommen wie mancher 8-bit-Controller. Deshalb sollten Sie auf irgendeiner Art von Entwicklungsboard montiert sein, damit der nicht-professionelle Entwickler sie überhaupt ausprobieren und für Prototypen nutzen kann. Leider sind die „käuflichen" Entwicklungsplatinen nicht für den Einsatz auf Steckboards ausgelegt, sondern oft zu plump und zu schwer (und zu teuer), um sie in einer Prototyp-Umgebung zu verwenden. Dagegen das T-Board: speziell entwickelt, um direkt in ein Steckboard gesteckt zu werden. Alle Anschlüsse des Controllers sind herausgeführt und erreichbar. Wenn Sie jemals ein anderes T-Board verwendet haben, werden Sie wissen, wie schnell und einfach es ist, damit einen Prototyp zum Laufen zu bringen!

T-Board ARMiert

Herz des ARMierten T-Boards ist ein ATSAMD21E18A-Mikrocontroller von Atmel, ein ARM-Cortex-M0+. Kennen Sie nicht? Dann bestaunen Sie einmal folgende Eigenschaften:

- 8/16/32-bit-Timer/Counter
- 3 24-bit-Timer/Counter
- 32-bit-RTC mit Uhr/Kalenderfunktion
- USB-2.0-Interface (Host- und Device-Funktionalität)
- 4 serielle Interfaces (USART, I²C, SPI, LIN)
- 12-bit-DAC mit 20 Kanälen (Differenz- und Single-ended-Eingang; Hardware-Oversampling-Unterstützung)
- 10-bit-DAC
- Inter-IC-Sound-Interface (I²S)
- 2 Analog-Komparatoren
- 16 externe Interrupts
- 26 GPIO-Pins

- Peripheral Touch Controller: 10 x 6 Zeilen
- Maximale Frequenz: 48 MHz
- Versorgung 1,62...3,63 V

Was für eine Vielzahl von Funktionen, verglichen mit einem guten alten 8-bit-Controller! Wir glauben, dieser Controller ist perfekt geeignet für den Einsatz auf einem T-Board, zumal die Anzahl der Pins noch überschaubar ist. Es gibt einen anständigen 256-KB-Flash-Speicher und 32 KB SRAM, der Controller unterstützt USB und ist zudem sehr preiswert. Nun, das „Wie“ und „Warum“ des ARMierten T-Boards haben wir nun abgehandelt, vertiefen wir uns in die Details der Konstruktion.

Das T-Board-Design

Bild 1 zeigt die Schaltung des ARM-T-Boards. Genau wie bei den AVR-Varianten wird die Versorgungsspannung über die Niederspannungsbuchse K1 mit 2,1-mm-Stift (Plus) zugeführt. D1, eine 1N4007-Diode, schützt die Elektronik vor Verpolung, während eine selbst-zurücksetzende PTC-Sicherung (F1) gegen Überstrom von mehr als 0,5 A schützt. Das ist mehr als ausreichend, um das Board und einiges an angeschlossener Peripherie zu versorgen (jeder VDD-Pin kann ein Maximum von 92 mA ziehen). Die Kondensatoren C1...C3 glätten und filtern den Versorgungsstrom, der Low-Dropout-Spannungsregler IC2 regelt die Ausgangsspannung auf 3,3 V. Im Gegensatz zu AVRs vertragen ARM-Cortex-MCUs eine maximale Spannung von 3,8 V. D1 zeigt an, dass das Board mit Strom versorgt wird, R1 begrenzt den LED-Strom. Die Jumper JP1 und JP2 helfen, den Strombedarf zu reduzieren - mehr dazu weiter unten.

Sie werden die Mikro-USB-Buchse K5 bemerken. Die 5-V-Versorgung aus dem USB-Anschluss wird natürlich dem Spannungsregler zugeführt, damit die Spannung gefiltert und auf 3,3 V reduziert werden kann. Elektrostatische Entladung (ESD) ist immer eine Gefahr, wenn man Peripheriegeräte an den USB anschließt.

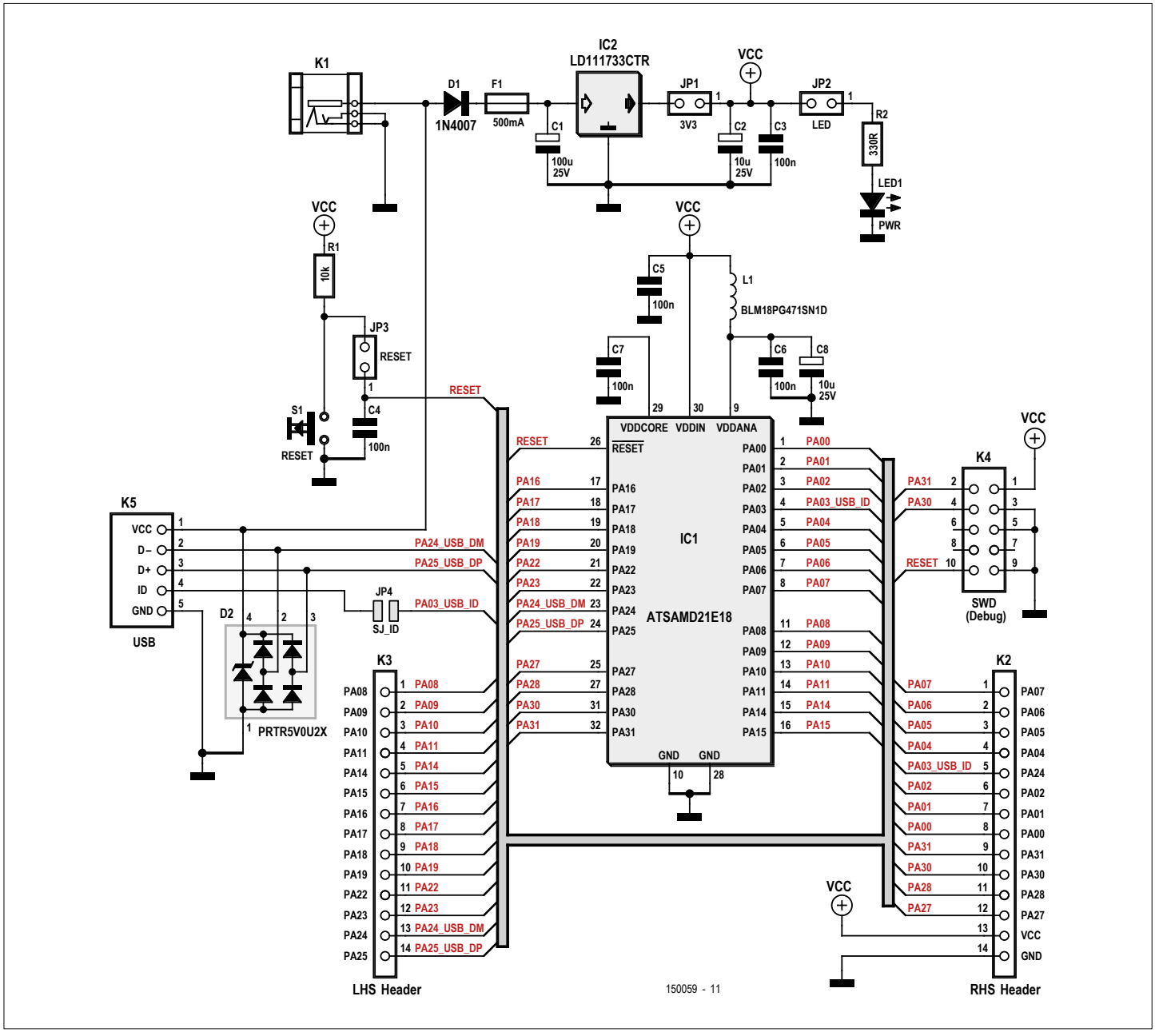
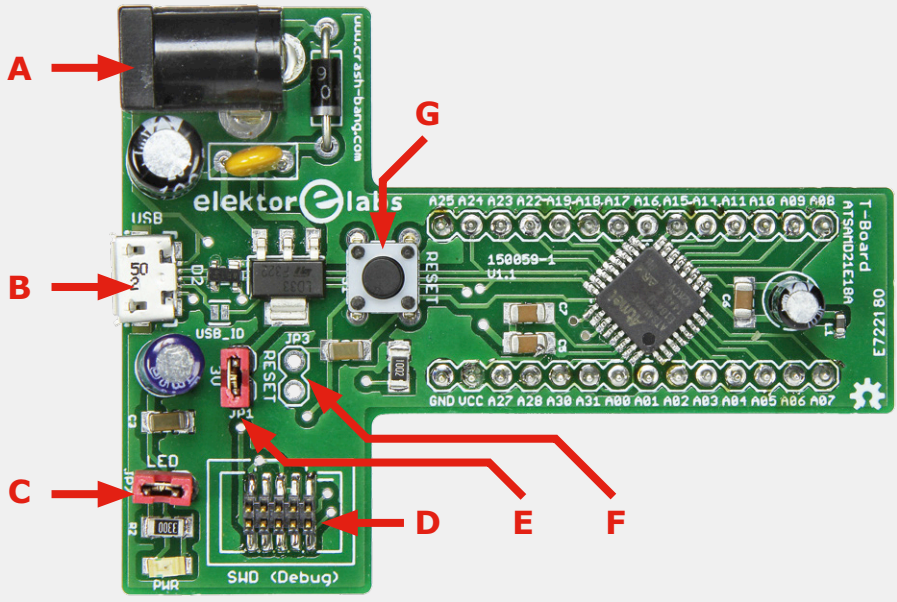


Bild 1. Schaltung des ARMierten T-Boards.

Physikalisches Layout des T-Boards

A. Spannungsversorgung: Niederspannungsbuchse mit 2,1-mm-Stift (plus), max. 9 V
B. Micro-USB-Buchse
C. LED-Jumper: Verbindet/trennt die Power-LED
D. SWD-Anschluss: 10-poliger Programmieranschluss (Serial Wire Debug)
E. Strommessung: Jumper ziehen, um die (interne) Stromaufnahme zu messen
F. Reset-Header: Ermöglicht ein abgesetztes Reset des Boards
G. Reset-Schalter



Deshalb schützt das ESD-Schutzdiodenarray D2 die Signalleitungen D- und D+ des USB. Das Array NXP PRTR5V0U2X, das wir für diesen Zweck ausgesucht haben, ist in einem winzigen SMD-Gehäuse untergebracht und speziell für Hochgeschwindigkeits-Datenleitungen geeignet. Schnallen Sie sich an: Der offizielle Name dieses Bauteils ist laut NXP *Ultra low capacitance double rail-to-rail ESD protection diode*. Der fünfte USB-Anschluss USB_ID, der für USB On the Go (OTG) verwendet wird, ist standardmäßig nicht mit dem Controller verbunden. Der Lötjumper jedoch ermöglicht es Ihnen, leicht eine Verbindung zu PA03 herzustellen. Mehr Infos zu diesem Thema finden Sie in einer Atmel Application Note [1].

In der Reset-Leistung gibt es einen 2-poligen Header, der ein „abgesetztes" Reset aus der Ferne ermöglicht. Programmierung und Debugging wird über einen 10-poligen Verbinder im 0,05-Zoll-Raster vorgenommen. Er trägt den Namen SWD (Serial Wire Debug), ist ein Teil einer JTAG-Schnittstelle und verwendet nur sechs Leitungen. Der 10-polige Header ist kompatibel zu Programmierern/Debuggern wie dem Atmel ICE.

Auf dem Board befinden sich externe Controller-Bauteile (einige Entkopplungskondensatoren wie im Datenblatt gefordert und eine Ferritperle L1), um Rauschen auf dem analogen Versorgungsspannungsanschluss VDDANA abzuschwächen. Schließlich sind alle I/O-Pins des Mikrocontrollers auf Stiftleisten geführt und damit zugänglich.

Es gibt keinen „festen" Quarz auf der Platine. So können Sie selber bestimmen, ob Sie einen Quarzoszillator im Bereich 0,4...32 MHz an PA14/PA15 oder einen 32,768-kHz-Uhrenquarz für RTC-Funktionen an PA00/PA01 einsetzen möchten.

Weniger Strom - mehr Möglichkeiten

In dem Artikel *T-Board 28: Übung in lowest Power* (Dezember 2014) berichtete der Autor über Möglichkeiten, den Stromverbrauch eines T-Board-28 mit einem ATmega328 zu messen und dann zu reduzieren. Die Fähigkeit, diese Art von Flexibilität und Kontrolle des Strom-

Stückliste

Widerstände:
SMD 1206, 1 %, 0,125 W
R1 = 10 k
R2 = 330 Ω

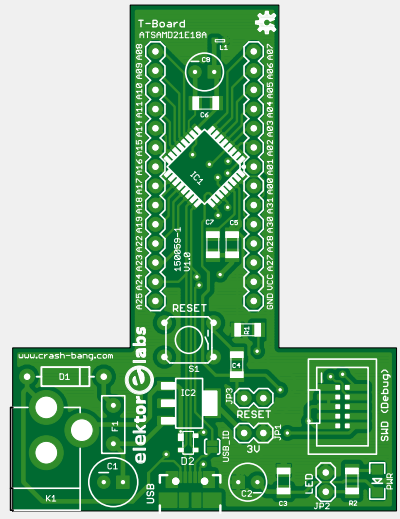
Kondensatoren:
C1 = 100 µ, 25 V, 20%, radial
C2,C8 = 10 µ, 25 V, 20%, radial
C3,C4,C5,C6,C7 = 100 n, 50 V, X7R, 1206

Induktivität:
L1 = BLM18PG471SN1D Ferritperle, 0,2 Ω/1 A, 470 Ω @ 100 MHz

Halbleiter:
D1 = 1N4007 (1000 V, 1 A)
LED1 = LED, rot, SMD 1206
IC1 = ATSAMD21E18A-AU
IC2 = LD1117S33CTR (3V3 LDO-Spannungsregler)
IC3 = PRTR5V0U2X (ESD-Schutzdiode)

Außerdem:
S1 = Taktiler Taster, 24 V, 50 mA, 6x6 mm²
JP1,JP2,JP3 = Jumper, 1x2, vertikal, 0,1''-Raster
F1 = Rücksetzbare PTC-Sicherung, 500 mA
K1 = Kleinspannungsbuchse mit 2,1-mm-Pin
K2,K3 = 14-polige Stiftreihe, SIL, 0,1''-Raster
K4 = 2x5-polige Stiftleiste, 0,05''-Raster
K5 = Micro-USB-Buchse (USB 2.0, Typ B, SMD)
2-polige Stiftleiste, 0,1''-Raster, für Jumper
Leerplatine 150059-1 im Elektor-Shop
Alternativ: fertig bestückte Platine 150059-91 im Elektor-Shop

Bild 2. Bestückungsplan des ARMierten T-Boards. Erschrecken Sie nicht ob der winzigen Bauteile, die Platine gibt es fix und fertig bestückt im Elektor-Shop zu kaufen!



bedarfs zu erreichen, war ein wichtiger Faktor bei der Gestaltung der T-Boards - und dies wurde bei der Gestaltung des ARMierten T-Boards nicht aufgegeben. Jumper JP1 ermöglicht es, eine Schaltung zur Strommessung anzuschließen und damit die Auswirkungen von Code-Änderungen und Optimierungen (in Hinsicht auf den Energiebedarf) zu beobachten. Auch bei den ursprünglichen AVR-T-Boards war das möglich, aber nun haben wir mit JP2 die Möglichkeit, die Power-LED von der Schaltung zu trennen, so dass der LED-Strom Ihre Messungen nicht verfälschen kann.

Das T-Board-Design

Wenn Sie sich schon mit T-Boards auseinandergesetzt haben, werden Ihnen die Eckpunkte des physikalischen Designs des ARM-T-Boards bekannt vorkommen. Der horizontale Querbalken des „T“ hat den Sinn, den Großteil der Bauteile aus dem Weg (Ihres Versuchsaufbaus) zu räumen, während auf dem vertikalen Balken alle Controlleranschlüsse herausgeführt und damit zugänglich sind und so leicht mit dem Steckboard verbunden werden können.

Das T-Board ist natürlich open-source, die Layout-Dateien [3] stehen kostenlos zum Download für Sie bereit. Wenn Sie die Ausrüstung besitzen und sich zutrauen, die Pins des ARM-Controllers im TQFP-Gehäuse mit 0,8-mm-Raster und die unsichtbaren Anschlüsse der USB-Buchse zu verlöten, können Sie sich mit der Stückliste und dem Layout in **Bild 2** ans Werk machen. Wenn nicht, so steht Ihnen ein komplett bestücktes und getestetes ARM-T-Board im Elektor-Shop zur Verfügung.

ARMiert und fertig

Genug zur Gestaltung, nun wollen wir das Board in Aktion erleben! Folgen wir dem Demo-Projekt Schritt für Schritt, um die Anwendung des Boards zu veranschaulichen. Um komplexere Projekte zu erkunden, empfehlen wir die Kurs-Reihe *ARM-Controller für Einsteiger* in den Elektor-Ausgaben Januar bis Oktober 2015. Arbeiten mit ARM-Mikrocontrollern ist auf jeden Fall komplizierter als mit AVRs. Wenn dies für Sie Neuland ist, benötigen Sie etwas Ausdauer und einige Zeit. Am besten ist es, wenn Sie den Kurs durcharbeiten. Wir haben aber hier nicht annähernd ähnlich genug Platz; unser Ziel ist es, Sie an das T-Board heranzuführen. Viacheslav Gromov, der dieses Projekt angeregt hat, will für kommende Elektor-Ausgaben noch mehr speziell auf das ARMierte T-Board ausgelegte Software schreiben und vorstellen. Dies im Hinterkopf behalten, und schon geht es los!

Schritt 1: Layout für das Steckbrett

Der erste Schritt ist es, das T-Board auf einem Steckbrett zu platzieren. Der Abschnitt mit der Spannungsversorgung soll über das Steckbrett hinausragen und die Stiftleisten an beiden Seiten des mittleren Kanals des Steckbretts aufgesteckt werden. Dann:

1. Stecken Sie eine Brücke zwischen den GND-Pin und der negativen Stromschiene des Boards.
2. Schließen Sie die Anode einer LED an Pin A07 und die Kathode an einer leeren Zeile des Steckboards an.
3. Setzen Sie einen Widerstand zwischen der Kathode und der negativen Stromschiene ein.

Bild 3 zeigt die aufgesteckte Platine mit allen Verbindungen. Es gibt in der gesamten Anordnung nur eine Drahtbrücke. Sauber!

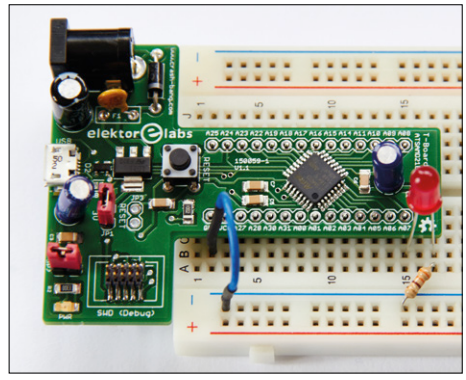


Bild 3. Das ARMierte T-Board betriebsbereit auf dem Steckbrett.

Schritt 2: Ein Projekt erstellen

Als nächstes wird das Projekt erstellt. Dazu verwenden wir das Atmel Studio 6.2 (siehe auch den **Kasten**: Alternative IDEs) und nutzen das Atmel Software Framework (ASF), unserer Meinung nach der schnellste Weg, um angesichts der Komplexität eines ARM-Mikrocontrollers ein Projekt zu verwirklichen. Auch im ARM-Kurs wurde das ASF verwendet, allerdings mit dem „SAMD20 Xplained Pro Board“ von Atmel.

Gehen Sie folgendermaßen vor:

Alternative IDEs

Wenn Sie ARM-Mikrocontroller verwenden, spielen Sie in der Champions League der Integrated Development Environments (IDE). Es gibt von einer Reihe von Experten eine Reihe von Meinungen, welche IDEs solide Werkzeuge sind und welche nicht, aber es ist nicht Aufgabe dieses Artikels, all diese recht komplexen Softwarepakete detailliert zu vergleichen. Um Ihnen jedoch einen Vorgeschmack zu geben, hier sind drei der beliebtesten IDEs:

Atmel Studio 6.2

Ich denke, dies ist die erste Wahl für diejenigen, die sich nicht professionell mit ARM-Mikrocontrollern beschäftigen. Die Software ist kostenlos, nicht in der Codegröße beschränkt und geschrieben und unterstützt durch Atmel. Damit ist gewährleistet, dass die IDE nicht nur die Controller selbst, sondern auch die Atmel-Programmer und Debugger unterstützt. *www.atmel.com/atmelstudio*

IAR Embedded Workbench für ARM

IAR bietet eine breite Palette von professionellen IDEs an - natürlich zu professionellen Preisen! Sie unterstützen AVR- und ARM-Mikrocontroller von Atmel sowie eine Reihe von ARM- und Nicht-ARM-MCUs anderer Hersteller. Wenn Sie Mikrocontroller unterschiedlicher Hersteller verwenden möchten, kann diese IDE eine gute Wahl sein. Es gibt code-beschränkte Versionen, die aber eher zum Ausprobieren geeignet sind als zur Realisierung echter (kleiner) Projekte. *www.iar.com/iar-embedded-workbench/arm*

Keil

Keil MDK-ARM ist eine IDE, die ARM selbst gehört. Es ist ein angesehenes Werkzeug zur Entwicklung von ARM-Projekten - vor allem für Professionals. Keil bietet allerdings eine Lite-Version des MDK-ARM-Tools an mit einer Codebegrenzung auf 32 KB. Das dürfte für die meisten Enthusiasten und Hobby-Programmierer mehr als genug sein. Wenn Sie zudem ARM-Mikrocontroller anderer Hersteller erkunden möchten, dann lohnt sich ein Blick auf: *www.keil.com*

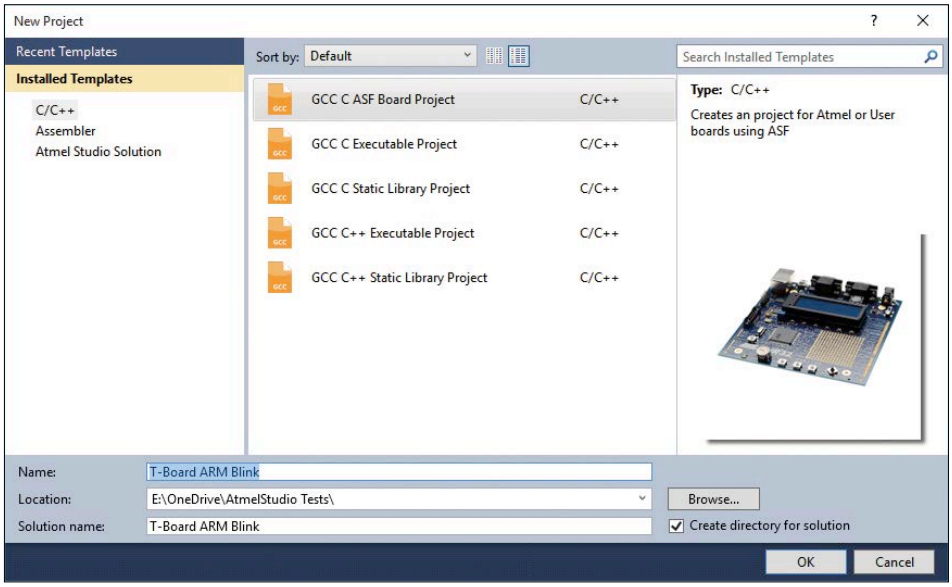


Bild 4. Grundeinstellungen für das GCC-ASF-Board-Projekt.

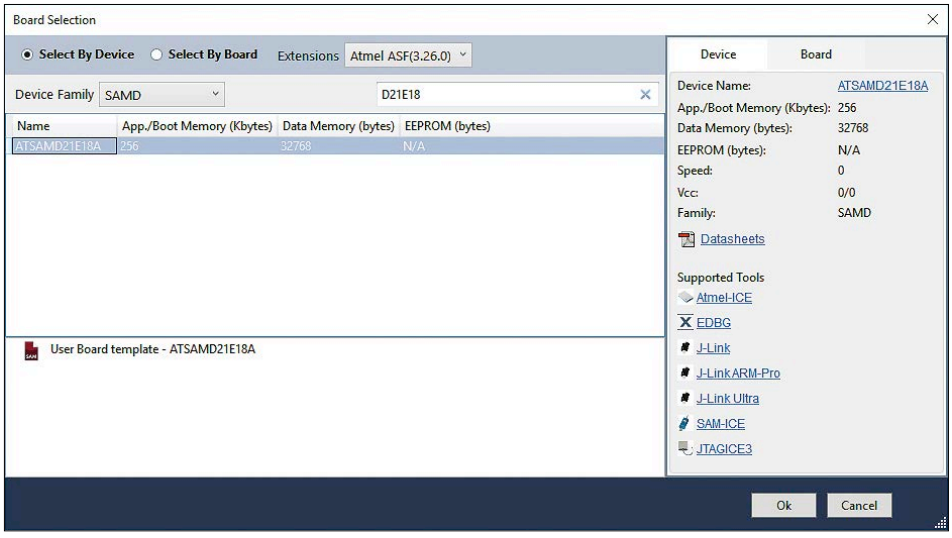


Bild 5. Auswahl des SAM-D-Controllers.

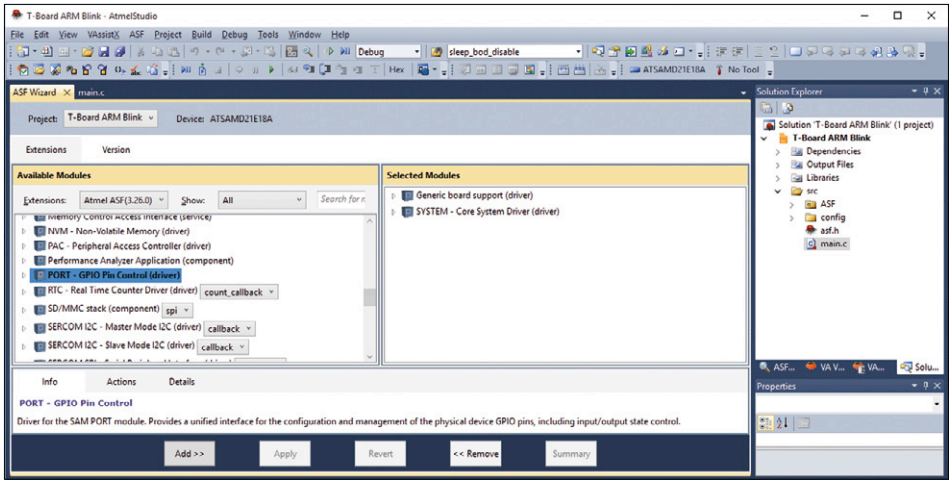


Bild 6. Hier wird der PORT-Treiber hinzugefügt.

1. Öffnen Sie das Atmel Studio und erstellen Sie ein neues Projekt. Wählen Sie *GCC ASF Board Project*, geben Sie dem Kind einen Namen und bestimmen Sie, wo es gespeichert werden soll (**Bild 4**).
2. Wählen Sie den Controller: Das ARM-T-Board ARM verwendet den *ATSAMD21E18A*. Da es keine Entwicklungsboards für diese Mikrocontroller von Atmel gibt, müssen Sie *User Board Template* wählen und auf OK klicken (**Bild 5**).
3. Ein leeres Projekt mit *main.c* unter dem *src*-Ordner im Solution Explorer ist nun erstellt.
4. Als Nächstes müssen wir die ASF PORT-Treiber zum Projekt hinzufügen. Diese bieten eine Reihe von Funktionen, die es uns erleichtern, auf die Controller-Anschlüsse zuzugreifen. Wählen Sie *ASF-Wizard* im *ASF*-Menü.
5. Im Popup-Dialogfeld (**Bild 6**) sind links die verfügbaren und rechts die gewählten Module zu sehen. Scrollen Sie auf der linken Seite nach unten, markieren Sie *PORT – GPIO Pin Control (driver)*, klicken Sie auf Add und dann Apply. Sie haben nun die PORT-Treiber zu Ihrem Projekt hinzugefügt.
6. Tragen Sie nun den Code aus **Listing 1** in das Modul *main.c* ein.

Schritt 3: Flash the Board!

Ihr Projekt ist nun bereit, kompiliert und auf das T-Board hochgeladen zu werden:

1. Überprüfen Sie das Projekt abschließend auf Fehler und kompilieren Sie es (F7).
2. Wenn Ihr Programmer nicht den ganzen Strom für das Projekt liefern soll (der Atmel-ICE, den ich benutze, kann dies nicht), müssen Sie eine externe Stromversorgung an der Gleichspannungsbuchse K1 anschließen. Wenn Sie einen JTAGICE3-Programmer verwenden wollen, sollten Sie die entsprechende Dokumentation zur Hand haben.
3. Schließen Sie den Programmer an den SWD-Header auf dem Board an und verbinden Sie den USB-Anschluss mit dem PC.
4. Wählen Sie den Programmer aus dem Menü *Tools/Device Programming.*
5. Laden Sie schließlich das Programm hoch: Wählen Sie *Start without Debugging* aus dem Menü *Debug*.
6. Sie sollten nun eine blinkende LED sehen!

Weblinks

[1] Atmel USB 2 GO: www.atmel.com/Images/Atmel_11201_USB-OTG-Like-Connector-Implementation_SAM9G-SAM9X-SAMA5D3_Application-Note.pdf
[2] T-Board 28: Übung in lowest Power: www.elektormagazine.de/140413
[3] Projektresource: www.elektormagazine.de/150059
[4] Atmel Studio: www.atmel.com/tools/atmelstudio.aspx
[5] www.der-hammer.info/terminal/

Die nächsten Schritte

Jetzt haben Sie das ARMierte T-Board in Aktion gesehen und möchten natürlich sofort mit weiteren Projekten loslegen. Nochmals: Wenn Sie unerfahren in Sachen ARM-Controller sind, so sei Ihnen dringend die Lektüre des Kurses *ARM-Controller für Einsteiger* empfohlen. Statt des Xplained-Pro- verwenden Sie dann eben das ARMierte T-Board. Dies reduziert die Anzahl der Steckbrücken, so dass die Projekte einfacher zu entwerfen und Fehler leichter zu finden sind.

In der Zwischenzeit präsentiert Ihnen der Co-Entwickler Viacheslav ein weiteres Kick-off-Programm, das er angesichts dieses „hardware-lastigen" Artikels geschrieben hat: siehe **Listing A** und den **Kasten**: *Mein/Dein/Unser erstes Projekt*. Wir hoffen, dass Ihnen das ARMierte T-Board viel Freude bereitet, als Lernwerkzeug und als ein Weg, Ihr nächstes Projekt schneller und effektiver zu entwickeln.

(150059)

Listing 1. Blinkende LED (im ARM-T-Board-Stil)

```
#include <asf.h>

#define LED_PIN PIN_PA07   //LED reliée à PA07

//prototypes de fonction
void configure_port_pins(void);


int main (void)
{
   system_init();   //routine pour initialiser le système, les horloges, etc.

   configure_port_pins();   //configuration des broches GPIO

   SysTick_Config(system_gclk_gen_get_hz(GCLK_GENERATOR_0));   //active l'interruption SysTick : lit la fréquence pour
définir l'interruption sur 1 s

   while (1)
   {
      //besoin d'aucun code ici puisque nous utilisons une interruption pour changer le niveau de la broche de la LED
   }
}


//fonction de configuration des broches de port
void configure_port_pins(void)
{
   struct port_config config_port_pin;      //structure utilisée pour stocker les paramètres
   port_get_config_defaults(&config_port_pin);   //lit la configuration actuelle des broches dans la structure
   config_port_pin.direction = PORT_PIN_DIR_OUTPUT;   //paramètre la structure pour indiquer que la broche est une sortie
   port_pin_set_config(LED_PIN, &config_port_pin);   //utilise la structure pour définir la direction de la broche de la LED
}


//gestionnaire d'interruption de l'interruption SysTick
void SysTick_Handler(void)
{
   port_pin_toggle_output_level(LED_PIN);   //change le niveau de la broche de la LED
}
```

Mein, dein, unser erstes Projekt

Von **Viacheslav Gromov**

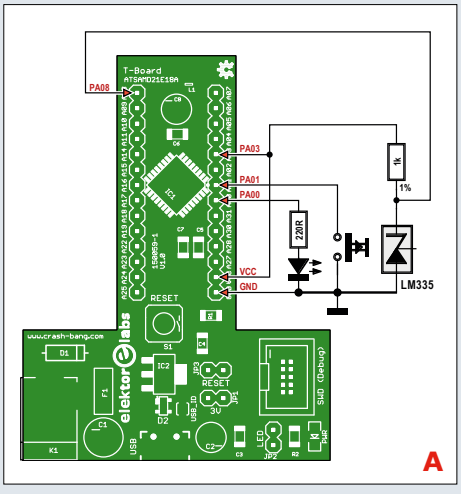
Lassen Sie uns ein einfaches Projekt realisieren, eine LED ein- und ausschalten, einen Taster abfragen und die Temperatur mit einem LM335 und dem ADC messen. Wir wollen alles schnörkellos mit einem Terminal-Emulator vom PC aus über USB steuern.

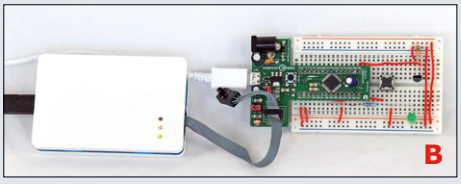
1. Die Hardware

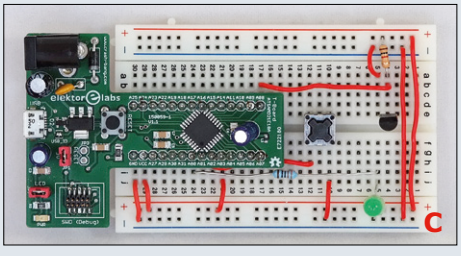
Bild A zeigt, wie die externen Bauteile an das ARMierte T-Board angeschlossen werden, **Bild B** die praktische Umsetzung. Sind alle Bauteile verbunden, schließen Sie das T-Board mit einem „USB-A-Stecker an USB Micro-B-Stecker"-Kabel am PC an. Dieser Kabeltyp wird oft zum Laden von Smartphones oder Tablets verwendet. An den SWD-Verbinder K4 wird ein Debugger (zum Beispiel Atmel ICE) und dieser selbstverständlich auch an den PC angeschlossen (**Bild C**).

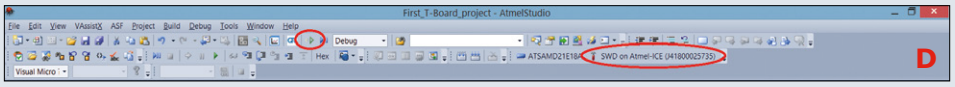
2. Die Software

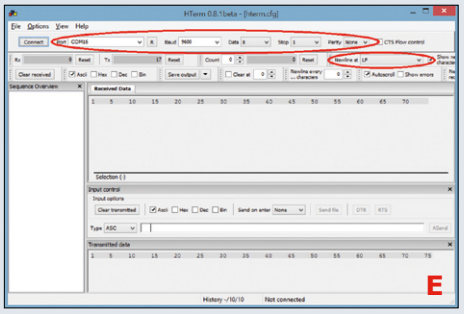
Zunächst besorgen Sie sich das das Atmel Studio-Projekt mit dem Namen „First_T-Board_project" von [3]. Dann öffnen Sie es im Atmel Studio 6, das Sie nach Registrierung oder als Gast von [4] herunterladen können (wenn Sie es nicht ohnehin schon installiert haben). Sind Sie mit der SAM-D-Controllerfamilie von Atmel und/oder dem Atmel-Studio nicht vertraut, sollten Sie einen intensiven Blick auf den Kurs *ARM-Controller für Einsteiger* werfen.

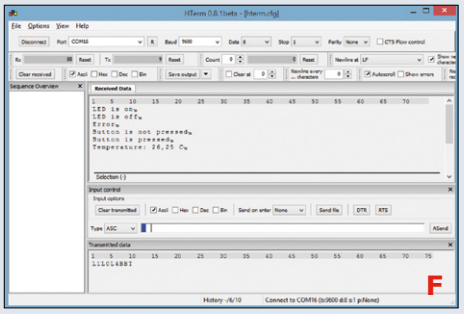


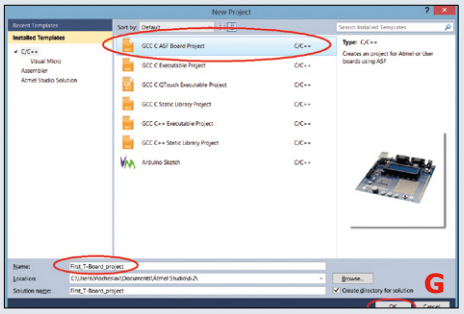


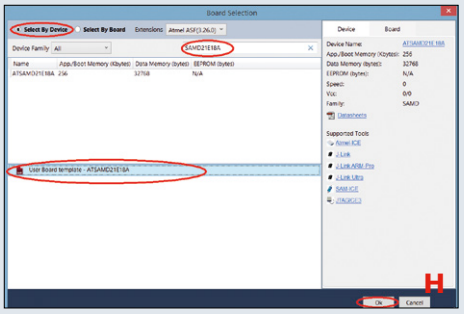












Das Programm beginnt mit der Deklaration der Variablen und Arrays, Funktionsprototypen und Konfigurationsfunktionen für die Peripherie. Danach werden in der Main-Funktion das System, die Interrupts und die Delay-Funktionen initialisiert und alle Konfigurationsfunktionen aufgerufen. **Listing A** zeigt die Endlosschleife mit dem wichtigsten Teil des Codes. Der gesamte Code steht in einer `switch`-Anweisung, die mit `udi_cdc_getc ()` ein ASCII-Zeichen über den USB empfängt und eine bestimmte Verzweigung veranlasst. **Tabelle A** zeigt alle Zeichen (Kombinationen), die wir später mit dem Terminalprogramm senden können. Für jede der drei Buchstaben gibt es einen `case`. Die Nummer des `case` ist kein Buchstabe, sondern dessen ASCII-Wert: 76 steht für „L", 66 für „B" und 84 für „T" (nur Großbuchstaben!).

Im ersten `case` (76) schaltet eine `else-if`-Anweisung die LED ein, wenn die Variable `number` 1 ist, und aus, wenn `number` gleich 0 ist. Die Variable `number` wird zuvor mit dem (zweiten) Zeichen gefüllt, das über den USB gesendet wurde. Als Rückmeldung wird „LED is on" und „LED is off" mit `udi_cdc_write_buf(&buffer, x)` über den USB gesendet. An Argumenten muss lediglich ein Pointer auf einen zu versendenden String im (Daten-)Puffer mitgegeben werden; die Anzahl der Zeichen wird durch `x` definiert. Hier allerdings setzen wir keinen Pointer, der Einfachheit halber betten wir einen Text wie „LED is

Tabelle A. Mini-Befehlssatz

L	**1**	„L1" (für LED) bewirkt, dass die LED eingeschaltet und eine Meldung „LED is on" empfangen wird. „L0" bedeutet, dass die LED ausgeschaltet und die Mitteilung „LED is off" empfangen wird. Sendet man ein „L" mit einer anderen Ziffer, so lautet die Meldung „Error".
	0	
B		„B" (für: Button) veranlasst das T-Board, die Meldung „Button is pressed" oder „Button is not pressed" auszugeben.
T		„T" (für: Temperatur) veranlasst das T-Board, einen String mit der aktuell gemessenen Temperatur x auszusenden: „Temperatur: x C".

on“ direkt in den Befehl ein: `udi_cdc_write_buf („LED is on\n „, 10);`. Dabei steht `\n` für eine neue Zeile. Wenn in der Variablen `number` ein anderer Wert als Eins oder Null vorgefunden wird, wird „Error“ an den PC gesendet.

Im nächsten `case` (66) überprüft eine `if-Anweisung` den Status des Tasters an Pin PA1 mit `port_pin_get_input_level(PIN_PA01)` und gibt entweder „Button is pressed“ oder „Button is not pressed“ über den USB aus.

Der letzte `case` (84) misst die Spannung am ADC-Kanal 16 (PA08), die vom LM335 geliefert wird, und berechnet aus dieser Spannung die Temperatur. Eine Zeichenkette „Temperature: x C“ wird über den USB geschickt.

3. Der erste Test

Probieren wir es aus! Wählen Sie Ihren Debugger oben rechts und starten den Kompilier- und Übertragungsvorgang mit einem Klick auf den grünen „Start Without Debugging“-Button (**Bild D**). Nachdem der Controller programmiert ist, wird das T-Board von Ihrem Computer als „Communication Device Class ASF example“ erkannt – genau, ein virtuelles serielles Interface. Vielleicht braucht Ihr Computer etwas Zeit, um die richtigen Treiber zu installieren, danach aber können Sie im Geräte-Manager unter „Anschlüsse (COM & LPT)“ überprüfen, welcher COM-Port Ihrem T-Board zugewiesen wurde. Öffnen Sie nun ein Terminal-Programm wie HTerm [5] und nehmen Sie die Einstellungen wie in **Bild E** vor:

- COM-Port-Nummer (siehe Gerätemanager)
- 9600 Baud
- 8 Datenbits
- 1 Stoppbit
- No Parity
- Newline bei „LF“
- DTR-Schaltfläche aktiv (im Bild unten rechts, hier noch ausgegraut)

Danach schließen Sie das Board an und testen das Ganze mit unseren Befehls-Großbuchstaben wie in **Bild F**. Herzlichen Glückwunsch, Ihr erstes Projekt mit dem ARMierten T-Board läuft!

4. Ein neues Projekt mit dem ARMierten T-Board und dem Atmel-Studio

Wie man ein neues Projekt gestaltet, wird im ersten Teil des ARM-Kurses beschrieben. Sie müssen genau so zunächst ein „GCC C ASF Board Project“ unter File/New/Project… wählen (**Bild G**), aber im nächsten Schritt, nachdem Sie „OK“ gedrückt haben, wie in **Bild H** unseren Controller ATSAMD21E18A (Select By Device). Nochmal ein Klick auf OK, schon ist das C-Projekt erzeugt. Jetzt können Sie die ASF-Bibliotheken im ASF-Wizard importieren und Ihren Code schreiben!

Listing A. Die Endlosschleife des Programms.

```
while(1){
switch (udi_cdc_getc())
{
	case 76: //if received "L"
	number = udi_cdc_getc(); //get the second character (a number)
	if (number == 49) //if received a one
	{
		port_pin_set_output_level(PIN_PA00, 1);		//put the LED on
		//send "LED is on" with a "new line" on USB
		udi_cdc_write_buf("LED is on\n", 10);
	}
	else if(number == 48) //if received a zero
	{
		port_pin_set_output_level(PIN_PA00, 0); //put the LED off
		//send "LED is off" with a "new line" on USB
		udi_cdc_write_buf("LED is off\n", 11);
	}
	else //if received a wrong number (not 1 or 0)
	{
	udi_cdc_write_buf("Error\n", 6); //send "Error" and a "new line" on USB
	}
	break;
	case 66: //if received "B"
	if(!port_pin_get_input_level(PIN_PA01)) //get the level on PA01 (button)
	{
		//if pressed, send "Button is pressed" and a "new line" on USB
		udi_cdc_write_buf("Button is pressed\n", 18);
	}
	else
	{
	//if not pressed, send "Button is not pressed" and a "new line" on USB
		udi_cdc_write_buf("Button is not pressed\n", 22);
	}
	break;
	case 84: //if received "T"
	adc_start_conversion(&adc_instance);  //start ADC-conversion
	//read and save the conversion result
	while(adc_read(&adc_instance, &data) == STATUS_BUSY){}
	//calculate the temperature
	temperature = (25 + (data * 0.000805 - 2.945) / 0.01) * 100;
	//reformat the result in a buffer
	sprintf(temperature_string, "%i", temperature);
	//send "Temperature:" with USB
	udi_cdc_write_buf("Temperature: ", 13);
	//send the temperature on USB
	for(i = 0; i < 4; i++){
	if((temperature >= 1000) && (i == 2)) udi_cdc_putc(44); //make a comma
	if((temperature < 1000) && (i == 1)) udi_cdc_putc(44);  //make a comma
	udi_cdc_putc(temperature_string[i]);  //send a digit on USB
	//send "C" and"new line" on USB at the end of transmission
	if(i == 3) udi_cdc_write_buf(" C\n", 3);
	}
	break;
	}
}
```

Camelback-Pegelindikator

Immer Wasser im Tank!

Von Laurent Labbe (Frankreich, China)

Mehr als GPS, Kreditkarte und schönes Wetter ist die Versorgung mit Trinkwasser die erste, wichtigste und essentielle Anforderung an jeden Backpacker oder Wanderer, der in Dschungel, Wüste, New York oder dem Outback überleben möchte. Daher wurden die „Camelback"-Rucksäcke erfunden, die einen zusammenfaltbaren Wassersack mit Schlauch und Mundstück enthalten. Leider ist schwer abzuschätzen, wieviel Wasser der Tank noch enthält. Mit unserer solarversorgten Pegelanzeige ist man aber stets informiert.

Dieses Projekt durchlief im Elektor-Labor das volle Programm. Das bedeutet, es wurde auf unserer Projekt-Website www.elektor-labs.com in der Rubrik *Proposals* gepostet, es wurde von den Elektor-Mitgliedern gesehen, diskutiert und bewertet. Das Projekt machte einen Schritt zu *In Progress* und einen weiteren zu *Finished,* und deshalb finden Sie es jetzt hier in der Zeitschrift.

Während der Phase *In Progress* wurde das Projekt von unserem Entwickler Luc Lemmens in enger Zusammenarbeit mit Projekt-Autor Laurent Labbe betreut. Luc entwarf für die Schaltung ein Elektor-typisches Platinenlayout. Mehr zur Entwicklungsgeschichte des Projekts finden Sie in den nachfolgenden Abschnitten.

Anlass und Überlegungen

Während vieler Mountainbike-Touren in der heißen Jahreszeit in und um meinen Wohnort herum habe ich einen „Kamelhöcker" für meine Trinkwasserversorgung getragen. Der Handelsname eines solchen Produkts ist übrigens Camelbak.

Während dieser Reisen habe ich oft vergeblich die Wassermenge im Tank richtig abzuschätzen versucht. Aber ich hatte während

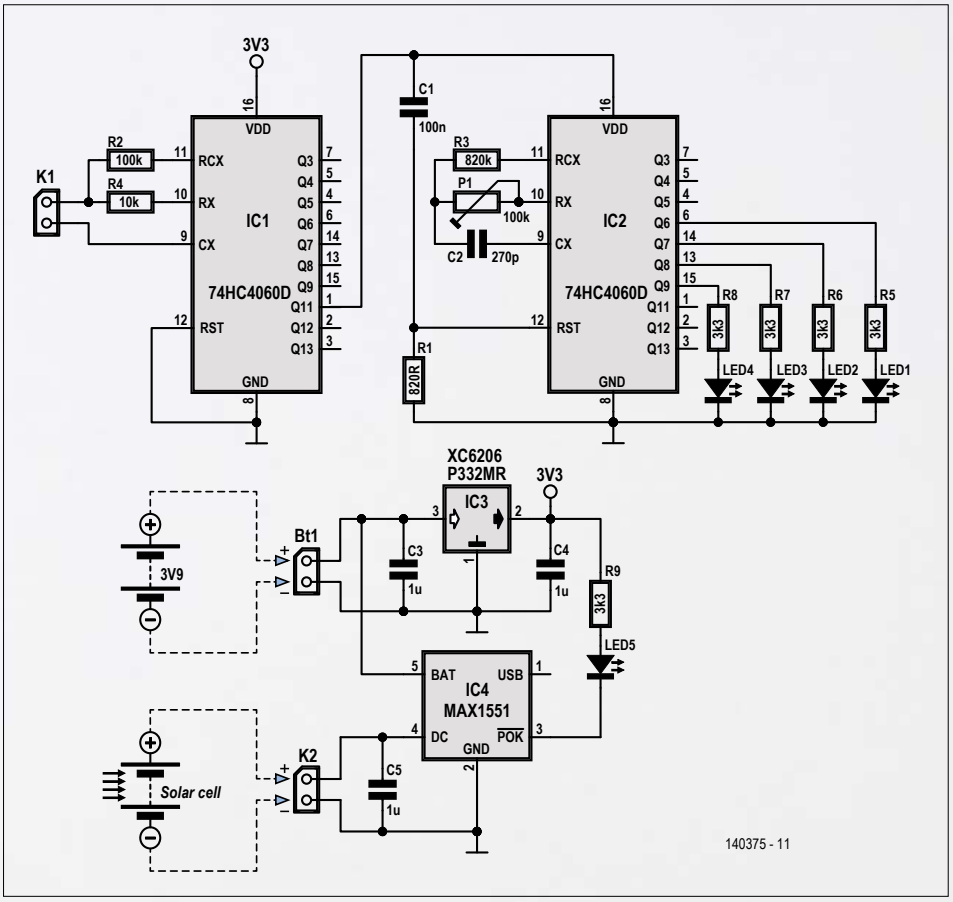


Bild 1. Schaltung des Camelback-Pegelindikators. Beachten Sie den Energiespartrick: Der erste 4060 versorgt den zweiten!

der Fahrten genug Zeit, mir Gedanken zu diesem Problem zu machen. Ich stellte mir einen Sensor im Wasserbeutel oder etwas Ähnliches vor, aber nichts überzeugte mich, bis ich auf den kapazitiven Flüssigkeitsdetektor im etwas anderen Regenmesser von Paul Cordonnier in Elektor April 2014 [1] stieß. Pauls Idee, den Sensor außerhalb des Wasserbehälters zu platzieren, gefiel mir.

Keine Fata Morgana

Ich begann mit ein paar Tests mit einem 74HC4060-Oszillator/Teiler mit 3,3-V-Versorgung (damit eine Lithium-Zelle zur Versorgung reichen würde). Ich habe eine Plastiktüte mit Wasser gefüllt und dann entleert. Mit Erfolg: Es konnte eine Variation der Frequenz

Stückliste

Widerstände
alle 5%, 0,125 W, 0805
R1 = 820 Ω
R2 = 100 k
R3 = 820 k
R4 = 10 k
R5,R6,R7,R8,R9 = 3k3
P1 = 100 k Trimmpoti, SMD

Kondensatoren
C1 = 100 n, 5%, 50 V, 0805
C2 = 270 p, 5%, 50 V, 0805
C3,C4,C5 = 1 µ, 10%, 16 V, 0805

Halbleiter
LED1...LED5 = LED, grün, low current, 0805
IC1,IC2 = 74HC4060 (SOIC-16)
IC3 = XC6206P332MR (SOT23-3)
IC4 = MAX1551

Außerdem
Solarpanel 5 V, 150 mW
4,2V-LiPo+-Batterie mit Halterung
Platine 140375-1 V1.0
2 selbstklebende Kupferstreifen, Länge 15...20 cm

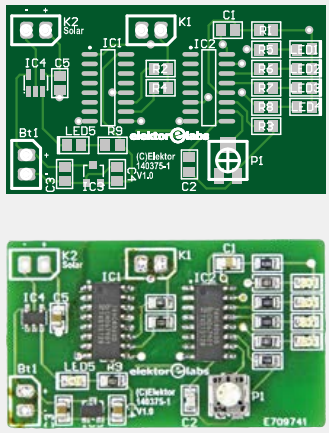


Bild 2. Wenn man es im Rucksack mitnehmen muss, zählt jedes Gramm und jeder Millimeter. Natürlich führt dann an SMDs kein Weg vorbei.

beobachtet werden! Um es einfacher zu machen, habe ich den 4060-Ausgang Q11 (Pin 1) verwendet, der eine Frequenz von etwa 150 Hz bei leerer und 60 Hz bei voller Plastiktüte erzeugt.

An dieser Stelle können wir einen Blick auf die Schaltung in **Bild 1** wagen. Wie können die „Daten“ des selbstgebauten kapazitiven Sensors an K1 angezeigt werden? Da die Frequenz an Q11 niedriger ist, wenn der Tank gefüllt, also die logische 1 effektiv länger ist, kam mir die Idee, dieses Signal zu verwenden, um ein anderes 4060-IC zu starten.

Dieser zweite Oszillator schwingt recht schnell mit ungefähr 30 kHz. Er startet, wenn die logische 1 an Q11 von IC1 beginnt und stoppt, wenn der Pegel am high-aktiven RST-Eingang des 4060 als logische 1 erkannt wird. In dieser Zeit läuft der Oszillator, je nach Wasserstand, bis zu den 128/256/512/1024-Teilerausgängen hoch. Die vier an diesen Ausgängen angeschlossenen LEDs geben also den Wasserstand im Kamelhöcker wieder. Diese Methode ist nicht nur elegant, sondern spart auch eine Menge Strom, denn IC2 ist nur während der High-Phasen von Q11-IC1 aktiv.

Es werden Hochleistungs-LEDs mit 3,3-kΩ-Widerständen (R5..R8) eingesetzt, so dass die Anzeige auch bei strahlendem Sonnenschein gut erkennbar ist. Der Sensor besteht aus zwei parallel verlaufenden, selbstklebenden Streifen

Bild 3. Der Prototyp des Autors baumelt am Tragriemen seines Rucksacks.

Bild 4. Der Wasserbehälter, wie man ihn in jedem Outdoor-Laden kaufen kann, mit dem aufgeklebten kapazitiven Sensor.

Bild 5. Der DIY-Sensor funktioniert genauso gut, wenn man ihn auf der anderen Seite des Tanks befestigt. Aber nicht innen!

Kupferfolie oder Kupferlaminat, die vertikal auf der Außenseite des Wassertanks befestigt sind.
Das Konzept klingt gut, aber lässt sich die Schaltung, insbesondere der Takt von IC2 an Trimmpoti P1 auch zuverlässig kalibrieren? Zugegeben, es ist ein bisschen schwierig, die richtige Einstellung für „leer“ und „voll“ zu finden. Bei meinem Prototypen habe ich es aber doch genau genug für meine Bedürfnisse hinbekommen. Ich habe die Schaltung in einem kleinen transparenten Gehäuse untergebracht – sehr klein und sehr leicht.
Später, als ich das Projekt auf www.elektor-labs.com postete, habe ich noch ein kleines Solarpanel hinzugefügt, um die LiPo-Batterie zu laden. Auch der Einsatz eines Mikrocontrollers zur automatischen Kalibrierung stand zur Debatte (die Idee wird derzeit intensiv verfolgt). Die erste Idee wurde schon verwirklicht, wie man im Schaltplan sieht. Ein 5-V-Panel, das 50 mA liefern kann, lädt über ein spezielles Lader-IC (MAX1551) die LiPo-Zelle. Der Gesamtverbrauch der Schaltung liegt bei ungefähr 3 mA. Die Batterie wird – wie ein zweistündiger Test-Ausritt ergab – auch bei nicht so idealen Bedingungen (bedeckter Himmel) immer im vollgeladenen Zustand gehalten. Die Klemmenspannung lag vor wie nach dem Trip bei 3,72 V! Obwohl die Schaltung damit lange Zeit autonom arbeiten kann, ist es keine schlechte Idee, die Solarzelle über eine (wasserdichte) USB-Buchse anzuschließen. Denn dann könnte man die Zelle bei Bedarf auch mit einem USB-Ladegerät (oder über einen USB-Hub) laden. Der Ladestrom ist durch den MAX1551 auf maximal 280 mA eingestellt.

Die Konstruktion

So klein die Platine und die Bauteile auch sein mögen, die Bestückung im Heimlabor mit einem kleinen Handlötkolben ist keine Hexerei, wenn man die Winzlinge auf der Leiterplatte zunächst sicher fixiert und dann mit Geduld und ruhiger Hand verlötet! Dies hat Luc Lemmens im Elektor-Labor bewiesen, wie **Bild 2** zeigt. Wie man die Platine, die Sensor-Kupferstreifen und das Gehäuse anfertigt und am Wassertank und am Rucksack befestigen kann, zeigen die **Bilder 3..5**. Ich habe alles in einer alten Dose für Reparaturzeug spritzwassergeschützt untergebracht. Wenn man ein dem Solarpanel angemessenes (wasserdichtes) Gehäuse auftreiben kann, wird die Angelegenheit noch kompakter und schlenkert auch nicht so am Rucksack herum.

Wer zuerst per pedes oder per pedalia mit einem funktionierenden Camelback-Pegelindikator im Elektor-Castle vorbeikommt [2], erhält zur Belohnung eine kostenlose Führung durch die Elektor-Laboratorien inklusive eines Besuchs des Brunnens im Keller der Burg. Der Wettbewerb endet am 30. November 2015, Mitarbeiter der Elektor International Media, ihrer Tochtergesellschaften und ihrer Lizenznehmer und/oder assoziierten Verlage dürfen nicht mitmachen. ⏮

(140375)

Weblinks

[1] Ein etwas anderer Regenmesser, www.elektormagazine.de/120554

[2] @ 51.016224, 5.838898; Mo-Do von 8.00 bis 17.00 Uhr, Fr 8.00 bis 13.00 Uhr. An der Tür klingeln!

Anzeige

SMD NAGLER
Ihr Spezialist für Bestückung von Prototypen und Kleinserien
SMD-Nagler
Abraham-Wolf-Str. 42
70597 Stutgart
Tel. 0711 12390019
Fax 0711 7653146
kontakt@smd-nagler.de
www.smd-nagler.de

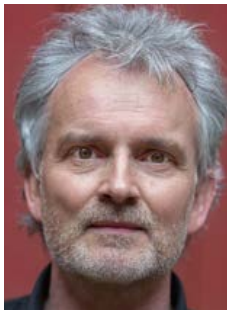
PCB JOKER
LEITERPLATTEN DISCOUNT FÜR PRAGMATIKER
100% Made in Germany
www.pcb-joker.com
PCB Joker GmbH

LC Design
Johann-Knecht-Str.28
63785 Obernburg
Tel. 06022 614430
Fax 06022 614431
Mail: info@lc-design.de
Website: www.lc-design.de

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

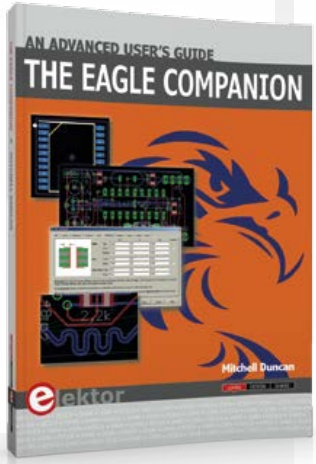
Willkommen in Ihrem E-SHOP

ELEKTOR EMPFIEHLT



THE EAGLE COMPANION

Auf über 600 Seiten verhilft der Autor Mitchell Duncan allen EAGLE-Anwendern zu tieferen Einblicken in das Programm samt seiner Funktionen und besonderen Fähigkeiten. Bislang war ich ja der Meinung, dass ich EAGLE im Prinzip verstanden hätte, aber dieses Buch überraschte mich dann doch mit leistungsfähigen Modulen, Befehlen und Funktionen, die EAGLE zum Star in der Platinen-CAD-Szene machen. Obwohl ich fast rund um die Uhr mit PCs samt Internet zu tun habe, ziehe ich vielfach vor, bestimmte Informationen in realen Büchern nachzuschlagen. Die eigentliche Stärke von EAGLE ist in seinen ULPs (User Language Programs) versteckt. Mit ULPs kann man EAGLE in ein ganz persönliches Programm verwandeln, das den eigenen Bedürfnissen und Vorlieben komplett entspricht. Dank Cadsoft gelang es Mitchell und mir zudem, das komplette offizielle ULP-Manual ungekürzt in dieses Buch zu integrieren. Wenn EAGLE jemals Ergänzung nötig hatte, dann in Form dieses Buchs. Aber Achtung: Es kommt per Paketpost!



Jan Buiting, Chefredakteur UK/US

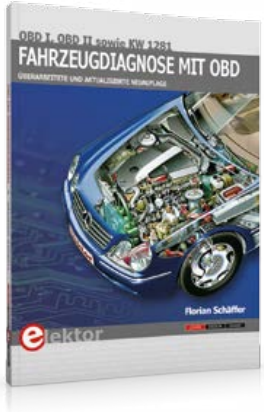
www.elektor.de/the-eagle-companion

Elektor-Bestseller

1. SmartScope
   www.elektor.de/smartscope

2. Raspberry Pi 2 (Mod. B)
   www.elektor.de/rpi-2
3. Multikopter
   www.elektor.de/multikopter
4. Arduino Sensor-Kit
   www.elektor.de/arduino-sensor-kit
5. PicoScope 2204A
   www.elektor.de/picoscope-2204a
6. Arduino entdecken
   www.elektor.de/arduino-entdecken
7. Red Pitaya V1.1
   www.elektor.de/red-pitaya-instrument
8. Microduino Basic Tutorial Kit
   www.elektor.de/microduino-kit

Fahrzeugdiagnose mit OBD

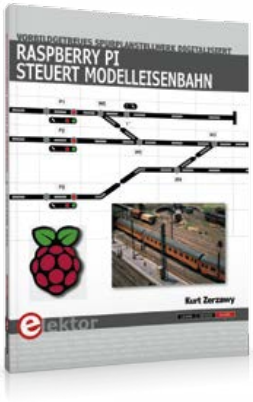


Streikt Ihr Auto schon wieder und Sie befürchten, dass die nächste teure Reparatur fällig wird? Vielleicht ist es aber auch nur ein kleiner Fehler, den Sie selber beheben können. Doch woher sollen Sie das bei einem modernen Fahrzeug mit all der Elektronik unter der Motorhaube wissen? Hier kann die fahrzeugeigene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

Mitgliederpreis: 42,00 €

www.elektor.de/obd-buch

Raspberry Pi steuert Modelleisenbahn



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

Mitgliederpreis: 34,80 €

www.elektor.de/rpi-eisenbahn

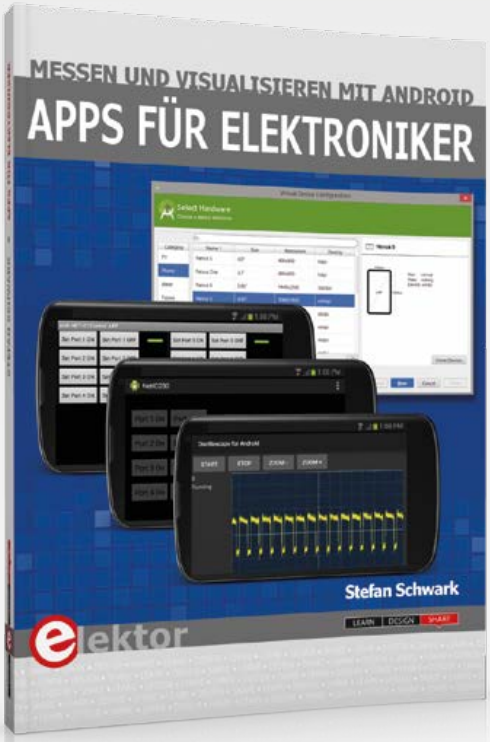
VFD-Shield für Arduino

Dieses Arduino-Shield enthält vier russische Siebensegment-Fluoreszenz-Anzeigen des Typs IV-3 mit den zugehörigen Treiberschaltungen. Vier 3-mm-LEDs sorgen für die Hintergrundbeleuchtung dieser Displays. Es gibt natürlich auch passende Software für den Arduino Uno, mit der man eine Uhr, einen Zähler oder ein Voltmeter realisieren kann. Das Ganze basiert selbstverständlich stilecht auf bedrahteten Bauteilen. SMDs müssen draußen bleiben.

Mitgliederpreis: 35,96 €

www.elektor.de/vfd-shield-for-arduino

Apps für Elektroniker



Apps für Smartphones gehören mittlerweile vollkommen selbstverständlich zum Alltag und sind in täglich wachsender Zahl in den entsprechenden Stores kostenlos oder für wenig Geld zu haben. Jeder Smartphone-Besitzer nutzt sie und passt sein Gerät damit seinen individuellen Bedürfnissen an. Leider sind die wenigsten Apps auf die Bedürfnisse von Elektronikern zugeschnitten, ändern kann man sie auch nicht und über die Möglichkeiten des Datenklaus durch unseriöse App-Anbieter mag man kaum nachdenken.
Gerade im Elektronikbereich stellt sich daher fast zwangsläufig die Frage, wie man eigene Projekte mit Hilfe seines Smartphones ansprechen und steuern kann oder wie sich die mit selbst gebauter Hardware gemessenen Daten auf den hochauflösenden Displays darstellen lassen.

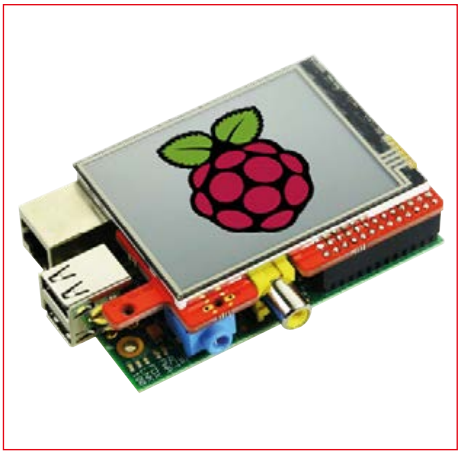
Dieses Buch veranschaulicht anhand verschiedener Beispiele, wie man eigene Apps programmieren kann, um damit gekaufte oder selbst gebaute Elektronik auf unterschiedlichen Wegen anzusprechen.

Mitgliederpreis: 29,80 € (frei Haus)
www.elektor.de/apps-fuer-elektroniker

Apps für Elektroniker
Messen und Visualisieren mit Android

PicoScope 2204A
Kompakt und leicht

Touchscreen für RPi



Aus dem befreundeten Watterott-Labor kommt ein Touch-Display für den Raspberry Pi, das wir ab sofort in unserem Online-Shop anbieten. Man kann damit das RPi-Modell B+ oder RPi 2 mit einem vollfarbigen 2,8"-Touchscreen mit 320 x 240 Pixeln ausrüsten. Dank des verfügbaren Framebuffer-Treibers ist das Display direkt einsatzbereit. Eine sehr nützliche Erweiterung für den RPi zu einem vernünftigen Preis!

Mitgliederpreis: 31,46 €

www.elektor.de/touch-display-for-raspberry-pi

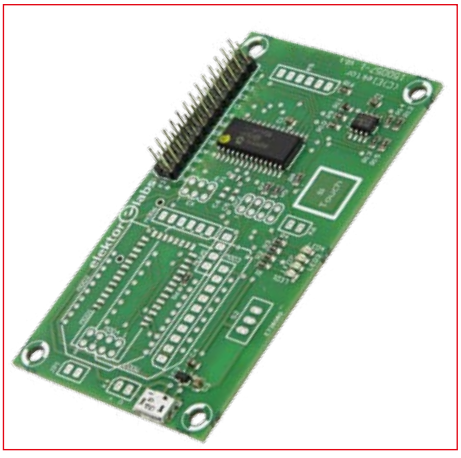
Praxiskurs AVR-XMEGA-Mikrocontroller

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

Mitgliederpreis: 39,80 €

www.elektor.de/kurs-avr-xmega

Android I/O-Board



Das Android I/O-Board erlaubt die Fernsteuerung der Pins eines PIC-Mikrocontrollers durch ein unter Android laufendes Gerät via WLAN, USB oder Bluetooth (Version 2.0 und 4.0). Die speziell hierfür entwickelte Java-Klassen (Libraries) zur Steuerung des Android I/O-Boards erleichtern die Entwicklung, da man sich so auf die eigentlichen Aufgaben des Projekts konzentrieren kann. Das Android I/O-Board basiert auf einem PIC16F1938-I/SO und bietet den Zugriff auf alle 25 I/O-Pins.

Mitgliederpreis: 22,46 €

www.elektor.de/android-i-o-board

Von **Rei Vilo**

Das Konzept hinter Microduino ist mir sehr sympathisch: Ein ganzes Ökosystem auf zwei zentralen Boards und schon am ersten Tag sind viele Shields verfügbar. Dank der sehr günstigen Preise kann man (statt Sensoren in vielen Stunden auf Platinen zu löten und das ganze irgendwie zu verkabeln) einfach die richtige Erweiterungsplatine für das Projekt aussuchen und das Ganze zu einem kompakten Stapel zusammenstecken.

Für klassische Arduino-Shields gibt es eine spezielle Erweiterungsplatine, die Pinkompatibilität zum Arduino Uno herstellt. Microduino bietet dabei ein Wiki mit einem eigenen Unterbereich, der dem Training gewidmet ist.

Microduino bietet kompakte und stapelbare Boards mit vielen Shields, tollen Tutorials sowie Hardware und Software, die komplett zu Arduino kompatibel ist. Eine perfekte Lösung für Projekte, bei denen es vor allem auf kleine Abmessungen ankommt.

Rei Vilo blogt zum Thema Embedded Computing auf embeddedcomputing.weebly.com.

Mehr Infos finden Sie unter
www.elektor.de/microduino-basic-tutorial-kit

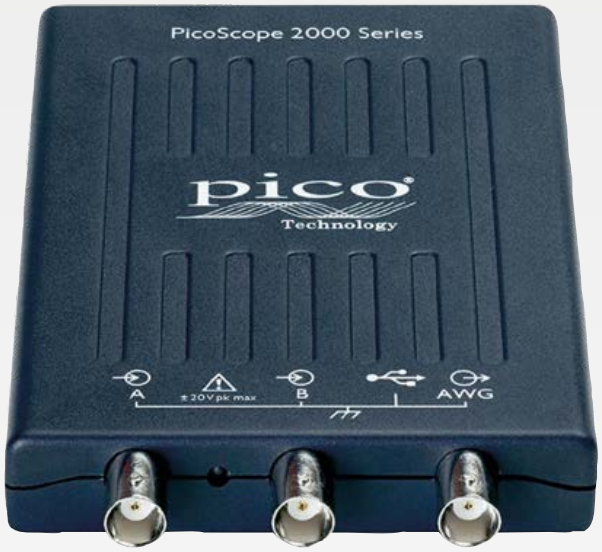
Wenn Sie uns einen ähnlichen Bericht über ein Elektor-Produkt schicken, der in Elektor veröffentlicht wird, erhalten Sie als Dankeschön einen 100-EUR-Gutscheincode, den Sie im Elektor-Shop einlösen können.
Mehr Infos gibt's unter www.elektor.de/review

PicoScope 2204A

Das PicoScope 2204A bietet Ihnen eine kompakte und leichte Alternative zu sperrigen Tisch-Geräten. Sie können jetzt ein Gerät mit 10 MHz und 100 MS/s problemlos in Ihrer Laptop-Tasche mitführen! Dieses Oszilloskop ist die ideale Lösung für Techniker im Außendienst und eignet sich hervorragend für zahlreiche Anwendungen wie die Forschung und Entwicklung, Prüfung, Ausbildung, Wartung und Überwachung sowie die Fehlersuche und Reparaturen.

Ebenfalls im Lieferumfang enthalten:

- **2 Tastköpfe**
- **USB-Kabel**
- **Software-CD**
- **Schnellstartanleitung**



Mitgliederpreis: 152,10 €

www.elektor.de/picoscope

Apps für Elektroniker

Messen und Visualisieren mit Android

PicoScope 2204A

Kompakt und leicht

Formelsammlung

Dieses Buch ist ein Nachschlagewerk mit praxisorientierten Fakten – kein Lehrbuch mit ausführlichen Erklärungen. Der Autor hat auch für komplexe Vorgänge praktische kurze Erklärungen, Näherungsformeln und Rechenbeispiele entwickelt, ohne die Darstellungen zu simplifizieren. Die logische Gliederung in zehn Kapitel vereinfacht das Nachschlagen und Aufsuchen der gewünschten Themen.

Mitgliederpreis: 29,80 €

www.elektor.de/formelsammlung

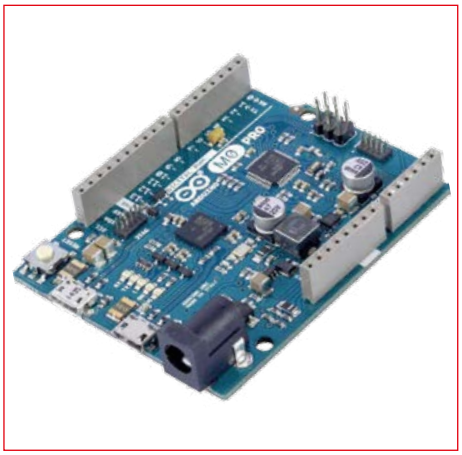
Trans Röhrenverstärker

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

Mitgliederpreis: 29,80 €

www.elektor.de/trans-roehren

Arduino M0 Pro



Arduino M0 Pro bietet eine einfache und doch leistungsfähige 32-bit-Erweiterung der Plattform Arduino Uno. Das Board basiert auf dem Mikrocontroller SAMD21 von Atmel, der auf einem 32-bit-ARM-Cortex-M0-Core aufsetzt. Die Power dieses Atmel-Controllers macht das Board außergewöhnlich flexibel. Außerdem ist M0 Pro eine ideale Lernplattform, mit der man erste Erfahrungen in der Entwicklung von 32-bit-Projekten sammeln kann.

Mitgliederpreis: 40,46 €

www.elektor.de/arduino-m0-pro

Willkommen bei **SHARE**

Von **Thijs Beckers** thijs.beckers@eimworld.com

Vielleicht waren Sie darauf eingestimmt, an dieser Stelle eine neue philosophische Betrachtung unseres spanischen Kollegen Jaime González-Arintero vorzufinden. Jaime ist inzwischen zum Ex-Kollegen avanciert, er hat die Elektor-Redaktion gegen ein neues spannendes Wirkungsfeld in der Multikulti-Hauptstadt Berlin eingetauscht.

Hier in unserem Team war Jaime der richtige Mann am richtigen Ort, er war ein kompetenter, zuverlässiger und allseits beliebter Kollege. Der Draht zu ihm, so hat er uns versprochen, wird nicht abreißen, denn auch auf vorgeschobenem Posten wird er uns redaktionell unterstützen.

Jaimes Markenzeichen war seine stets gute Laune, obwohl die Fristen bis zum Redaktionsschluss manchmal massiv drückten. Auch auf Messen und Events war Jaime in seinem Element, er stand Teilnehmern und Besuchern in den Sprachen Englisch, Deutsch, Niederländisch und natürlich in seiner Muttersprache Spanisch gern Rede und Antwort.

Auch wenn Jaime in unserem Team eine kaum zu schließende Lücke hinterlässt, wünschen wir ihm in seinem neuen Wirkungskreis *Alles Gute*, *All the best* und natürlich *Todo lo mejor*!

Die redaktionelle Betreuung der SHARE-Seiten habe nun ich übernommen, mein Name ist Thijs Beckers. Seit einem Monat bin ich auch Chefredakteur der niederländischen Elektor-Ausgabe. ⏮

(150481)gd

Ist es E-ethisch?

Von **Tessel Renzenbrink**

„Hände hoch, bitte: Wer von Ihnen denkt, Elektronik-Unternehmen sollten eine aktive Rolle spielen, wenn es um die *elektronische* Ethik unserer zukünftigen Welt geht?“

Diese Frage wurde auf der Jahreskonferenz einer nationalen Elektronik-Branchenorganisation gestellt. Was glauben Sie, wie viele Hände von den 300 Elektronik-Unternehmern im Kongressraum gehoben wurden? Nur fünf.

Mehr als 98% der Unternehmer und Entscheidungsträger glauben also, dass sie einer Vogel-Strauß-Politik folgen können, wenn es um Fragen geht wie: „Sollen wir einen Autofahrer sein Auto so programmieren lassen, dass es automatisch für einen Hund bremst in einer Situation, die gefährlich ist für den Fahrer selbst?“

Wir bei Elektor meinen: Das muss sich ändern!

Mindestens 290 Hände sollten beim nächsten Mal in die Höhe schießen! Weil wir Entwickler und Produzenten eine wesentliche Verantwortung tragen, wenn es um elektronik-bezogene Ethik geht. Und man muss es nicht leugnen, es geht auch ums Geschäft. Die ethischen Diskussionen werden auf die eine oder andere Art zu Innovationen führen, zu neuen Möglichkeiten, neuen Geschäftsfeldern. Und last but not least macht es auch Spaß, sich vorzustellen, wie Innovationen der Elektronik die Welt verändern könnten.

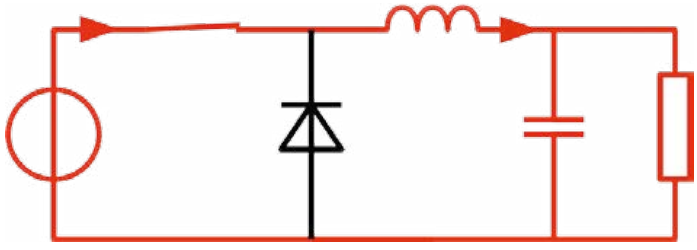
Deshalb starten wir im November die Rubrik *Elektor EEthik*, online und in gedruckter Form. Sie tritt an die Stelle unserer Rubrik *Elektor World*, die auch weiterhin regelmäßig erscheint, aber nicht mehr in einem festen Format.

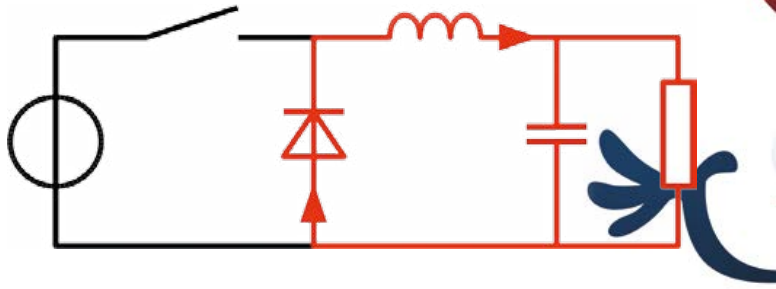
Haben Sie Fragen, Probleme oder Anmerkungen zu ethischen Elektronik-Themen? Dann nehmen Sie mit mir Kontakt auf: Tessel Renzenbrink (tessel.renzenbrink@eimworld.com). Ich werde bei Elektor ab sofort die Diskussionen über Ethik koordinieren. ⏮

(150491)

Bockige Buck-Converter

Das machen Sie immer ganz anders?





Von **Thijs Beckers** (Redaktion NL)

Hilfe! Diesen seltsamen Abwärtswandler bekommen wir im Labor einfach nicht unter Kontrolle. Was machen wir falsch? Wir wissen es nicht. Sie vielleicht?

„Im Nachhinein kann man einfach sagen, dass ich keinen LM2677T-ADJ hätte verwenden sollen."

So lautete der Kommentar unseres Kollegen im Labor, der mit dem Entwurf eines neuen Labornetzteils betraut war.

„Selbst ein fester Spannungsteiler für 30 V hat bei einer Impulsbelastung von mehreren Ampere dramatische Probleme."

Gehen wir einen Schritt zurück und beginnen von vorne, mit der Idee eines Labornetzgerätes mit folgenden Spezifikationen:

- Einstellbare Ausgangsspannung von 0...30 V
- Strom mindestens 3 A mit Begrenzung
- Stabil unter den gängigsten Laboranwendungen

Das ist eine ziemlich lockere Definition und gab den Entwicklern viel Freiraum. Und weil wir nicht umsonst Elektor heißen und unseren Ruf nicht mit kopierten Schaltungen und wenig originellen Ideen erworben haben, wählten wir natürlich nicht die einfachste 0815-Lösung. Und dann gibt es halt manchmal Überraschungen wie bei dieser Schaltung.
Bei der Suche nach interessanten ICs entdeckten wir den LM2677T-ADJ von Texas Instruments, der relativ einfach zu handhaben ist, eine hohe Effizienz und gute Spannungsstabilisierung bietet sowie bis zu 5 A Ausgangsstrom liefern kann. Alles in Ordnung, wir können loslegen!
So entstand die erste Schaltung, Version v0.01. Wir gingen von einer (einigermaßen) stabilisierten Spannung von 36 V (an K1) aus. Der P-Kanal-JFET T1 soll Ein- und Ausschalteffekten vorbeugen. Die Ausgangsspannung wird am Rückkopplungseingang eingestellt. Mit IC2B und IC2D ist eine aktive Regelung realisiert, die eine Strombegrenzung ermöglicht. IC2A misst mit Shunt-Widerstand R12 den Strom und zeigt ihn auf einem Instrument an. Für IC2 ist eine 5-V-Versorgung erforderlich. Darum kümmert sich IC3, ein LM317. C4, C5 und R10/C13 werden benötigt, um die Regelung unter Last zu stabilisieren. Zeit für einen Prototypen.
Ohne externe Last liegt am Ausgang unseres Prototyps eine Mindestspannung von etwa 0,3 V, bei einer Last von 40 Ω etwa 0,017 V. Nicht schlecht ohne negative Hilfsspannung. Der gemessene Wirkungsgrad von fast 94 % bei 30 V/5 A kann sich auch sehen lassen.

Der nächste Test, Impulsbelastung mit 5 A, Tastverhältnis von 50 %, verlief etwas weniger positiv: Die Ausgangsspannung klappte wie ein Kartenhaus in sich zusammen und war alles andere als stabil. Wir fanden keine Kombination von Spule und Ausgangskapazität für das Ausgangsfilter (L1/C10-C12), mit dem es klappte. 82 µH statt 22 µH änderte so gut wie nichts. Die einzige „Vorteil" war, dass nun die Spannung beim Ladeimpuls einfach nur zusammenbrach und nicht auch noch über die eingestellte Spannung hinausschoss. Eine größere Ausgangskapazität hatte bei niedrigen Spannungen einen positiven Effekt auf die Regelung bei Impulsbelastung, verursachte aber einen noch stärkeren Ausgangsspannungsabfall bei höheren Spannungen (>20 V). Bei 50 % Tastverhältnis war der Regler auch wieder etwas instabiler, was wiederum einen Einfluss auf die Strombegrenzung hatte.
Alle Massen miteinander zu verbinden verbesserte die Situation leicht. Eine große Massefläche auf der Oberseite ist wohl unverzichtbar (der Prototyp besaß eine einseitige Platine), aber dies dürfte das Regelverhalten auch nicht wesentlich verbessern. Simulationsberechnungen von WEBENCH (webench.ti.com) ergaben für unterschiedliche Ausgangsspannungen auch verschiedene Kombinationen für die Filterkomponenten...
Vielleicht ist der LM2677 zu empfindlich für eine Welligkeit auf dem Ausgang, wie sie eine schaltende Belastung verursacht? Experimente mit dem LC-Filter (82 µH und 220µF) an einer „normalen" DC-Stromversorgung und einer schaltenden Last

ergaben eine kleine Überraschung: Bei 1,32 kHz (etwa die Resonanzfrequenz) Impulsbelastung und 50 % Tastverhältnis fließt ein Gleichstrom durch die Spule, der laut Multimeter etwa halb so groß ist wie der Spitzenstrom. Seltsamerweise scheint es auf dem Oszilloskop etwas weniger zu sein. Aber es gibt auch eine beinahe sinusförmige Wechselstromkomponente, mit der Frequenz der Last natürlich.

In unserem Test schien der DC+AC-Peak eineinhalbmal größer zu sein als der maximale Strom, ohne dass die Last geschaltet wurde. Belastet man etwa ein Netzteil mit 3 A, so rechnet man bei einem Tastverhältnis von 50 % mit einem mittleren Strom von 1,5 A. Doch der Spitzenstrom liegt dann plötzlich bei fast 4,5 A, wenn die Last in der Resonanzfrequenz des Ausgangsfilters geschaltet wird. Der Regler in unserem Netzgerät muss also mindestens etwa 8 A liefern können, um einen Impuls von 5 A ohne Probleme in den Griff zu bekommen...

Das Internet konnte nichts zur Lösung des Problems beitragen. Doch wir dürften doch sicher nicht die ersten sein, die in eine solche Falle tappen?! Auf der anderen Seite, wie realitätsnah ist überhaupt eine solche Impulsbelastung, ist dies eine „gängige Laboranwendung“?

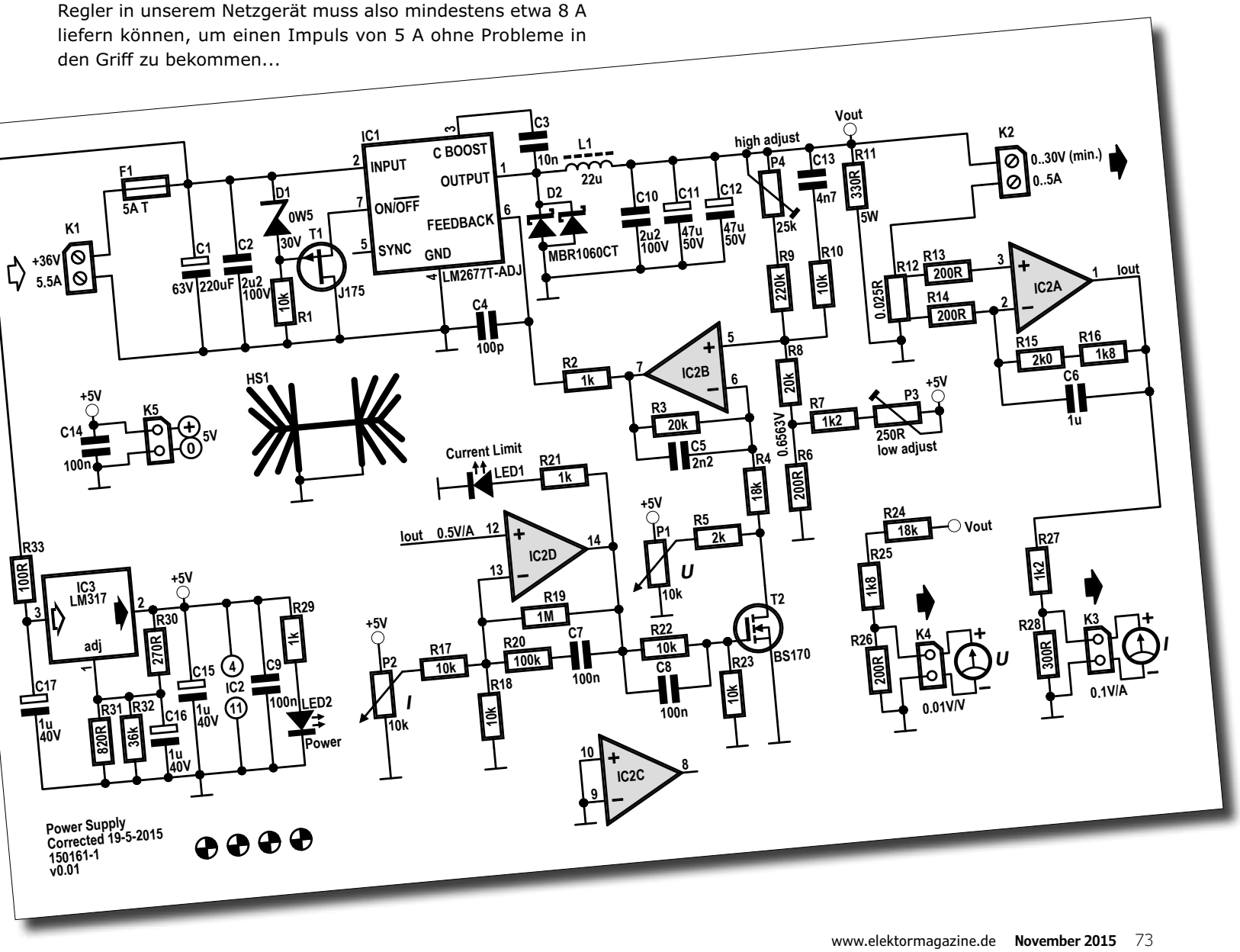
Wir könnten das Netzgerät natürlich auf einen niedrigeren Strom begrenzen, aber das wäre eine Schande, denn bei konstanter Belastung sind ohne weiteres 5 A möglich.

Ist das vielleicht ein typisches Verhalten für alle Abwärtswandler mit LC-Ausgangsfilter? Vielleicht wird das Ausgangsfilter in (Reihen-)Resonanz gebracht, weil das Problem dann auftaucht, wenn die Schaltfrequenz der Last in der Nähe der Resonanzfrequenz des Ausgangsfilters liegt? Vielleicht fließt irgendwo ein parasitärer Strom?

Was halten Sie von folgendem Gedanken? Die Schaltlast sorgt im Prinzip für Spannungs-Überschwingen, aber da der Regler versucht, die Ausgangsspannung konstant zu halten, kann die im LC-Kreis gespeicherte Energie nur in Form einer Stromspitze entweichen.

Teilen Sie uns Ihre Meinung mit! Fortsetzung folgt...

(150487)



Schallband-Rekorder Tefifon

Das Beste aus Köln nach Kölsch und 4711

Von **Peter Beil** (D)

Anfang der 1950er Jahre habe ich als Schüler im örtlichen Kino als Vorführer gejobbt. Für die damals übliche Begrüßungsmusik sorgte ein aus heutiger Sicht mehr als skurriles Gerät, dem diese Retronik-Folge gewidmet ist.

Selbst Elektroniker dürften heute kaum noch wissen, dass es einmal ein Zwischending zwischen Plattenspieler und Tonbandgerät gab. Dieser Apparat war tatsächlich ein Zwitter, denn er basierte auf dem sogenannten Schallband, einem Kunststoffband mit Rillen drin.

Das „Tefifon“ war ein solcher Schallbandspieler. Das Grundprinzip wurde bereits in den 1930er Jahren von Dr. Karl Daniel (**Bild 1**) entwickelt und war als Anrufbeantworter konzipiert. Als „Speichermedium“ diente ein flexibles Endlosband mit parallelen Schallrillen, die elektromechanisch abgetastet wurden (siehe **Bild 2**). Im Gegensatz zu späteren Modellen konnte dieses Gerät sogar aufzeichnen. Der für die damalige Zeit durchaus als Hightech-Gadget zu bezeichnende Apparat scheiterte aber an den gar gestrengen Reglementierungen der Reichspost.

Bild 1. Der Gründer der Firma Tefi, Dr. Karl Daniel (1905-1977).
Quelle: H. Jüttemann, „Das Tefifon“.

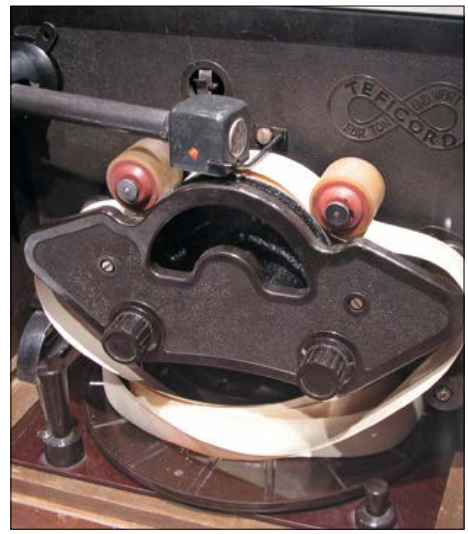


Bild 2. Das TefiCord ist ein Anrufbeantworter. Es konnte „Ton und Sprache“ aufzeichnen, scheiterte aber an den Restriktionen der Reichspost (*Quelle: Wikipedia.de).*

Die ersten Kassetten

Auf der Rundfunkausstellung von 1936 in Berlin wurde dann das erste „Tefiphon“ (damals noch mit „ph“) vorgestellt. Die Bandgeschwindigkeit betrug beeindruckende 45,6 cm/s und die Laufzeit unerhörte zwölf Stunden. In der Nachkriegszeit Anfang der 1950er Jahre war die Schelllackplatte mit ihren 78 U/Min. noch weit verbreitet. Hier bot die Schallband- Technik noch einige bedeutende Vorteile. Gegenüber der Platte mit ihrer geringen Spieldauer von drei bis vier Minuten, konnte das Nachkriegsmodell B 51 mit über einer Stunde Laufzeit punkten. Das verwendete Band war 16 mm breit und hatte 70 parallele Rillen, die beim Abspielen eine durchgehende Rille ergeben. Damit war es möglich, ganze Symphonien, Opern oder Operetten unterbrechungsfrei wiederzugeben. Die Auslenkung zur nächsten Rille geschah übrigens nicht wie bei der Schallplatte in einem Schritt, sondern in drei kleinen Stufen. Das weiche und flexible PVC-Trägermaterial knisterte und knackte fast überhaupt nicht. Mit den etwa vier Rillen pro Millimeter wurde übrigens bereits die Mikroschrift vorweggenommen. Auch das Prinzip der Kassettentechnik war neu. Um 1954 reduzierte man die Bandgeschwindigkeit auf 19 cm/s und erzielte mit den Kassetten eine Laufzeit von vier Stunden. Schon ab 1953 gab es zum Nachrüsten auch Modelle mit Magnetband-Kassetten, die aber wegen der hohen Kosten gegenüber einem „echten“ Tonbandgerät keine Verbreitung fanden.

Finessen des Typs KC 1

Im Laufe der Zeit wurde eine große Anzahl unterschiedlicher Ausführungen hergestellt, doch die Version mit der höchsten Auflage war wohl das Tefifon KC 1 (1956 bis 1959, diesmal mit „f“, siehe **Bild 3**). Dieses Audiogerät im Hammerschlag-Blechgehäuse kostete damals rund 150 DM (entspricht heute der Kaufkraft von etwa 700 €). Ein großer Vorzug gegenüber dem Vormodell war, dass die Tonabnehmermechanik wohl aus Servicegründen nach oben verlegt wurde.
Neben der Vier-Stunden-Kassette gab es auch eine kleinere Version mit nur einer Stunde Abspielzeit. Beim Aufsetzen auf das Gerät wurde der Wickelteller im Inneren angehoben, um Reibgeräusche zu vermeiden. Das Band wurde innen herausgezogen (**Bild 4**) und lag auf einem angetriebenen Teller, um ein Zuziehen und Klemmen des Wickels zu verhindern.

Bild 3. Das Modell KC 1 war das erfolgreichste Produkt der Tefi-Reihe.

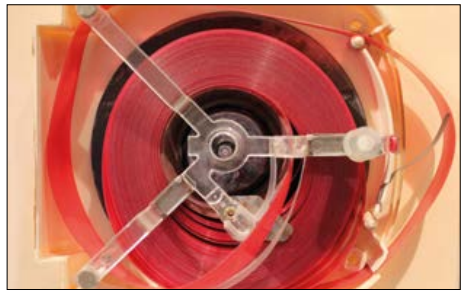


Bild 4. Der ausgetüftelte Aufbau einer Tefifon-Kassette.

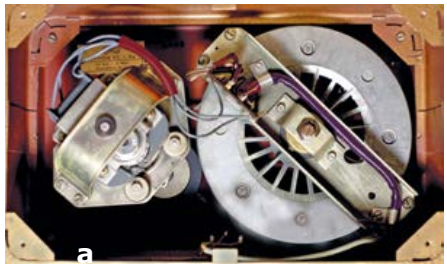


Bild 5. Einblicke in die Mechanik des Reibradantriebs des Modells KC 1 (5a) sowie den Riemenantrieb des Wickelkonus (5b).

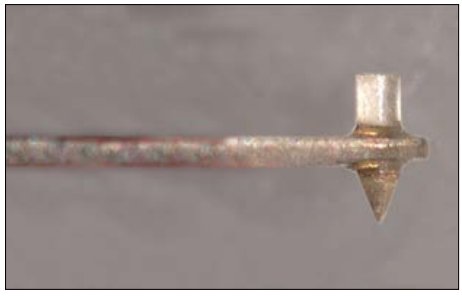


Bild 6. Makroaufnahme des KC-1-Kristalltonabnehmers.

Bild 7. Aufsetzdämpfung (weißer Pfeil).

Das Band musste man von Hand um eine gummierte Transportrolle legen, an die das Band im Betrieb mit einer weiteren Rolle angedrückt wurde. Über ein Reibrad trieb der Motor die zugehörige Schwungmasse an (**Bild 5a**) und zum Wickelkonus führte ein Gummiriemen (**Bild 5b**). Der Tonabnehmer bestand aus einem Kristall, der auf einen elastischen Metallträger montiert war (**Bild 6**). Er konnte durch eine vertikale Wippenmechanik an der gewünschten Höhe aufgesetzt werden. Die Wippe hatte die Form eines Parallelogramms und war mit einem Gegengewicht versehen. Diese Konstruktion vermied die typischen Skating-Effekte bzw. Probleme mit den Radien bei der Schallplatte. Bemerkenswert ist auch die mechanische Dämpfung, die für ein sanftes „Aufsetzen“ sorgt (**Bild 7**). Sogar heute, nach über 60 Jahren, funktioniert sie noch!

Technik und Nutzung

Im Grundgerät Tefifon KC 1 war kein Leistungsverstärker enthalten. Vorgesehen war, es an eines der damals üblichen

ESTD 2004
www.elektor.tv

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

Bild 8. Damals futuristisch: Balkenanzeige der Bandspur *(Quelle: Google Images)*.

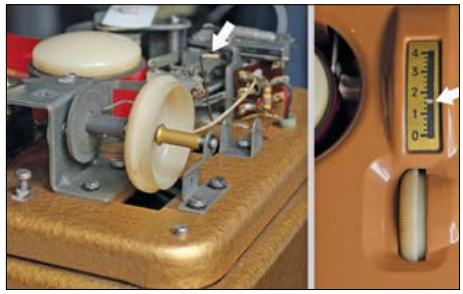


Bild 9. Spurauswahl und Anzeige.

Bild 10. Ab jetzt kann ein Tefifon auch Platten abspielen!

Bild 11. Der Dreiminutenadapter für das Tefifon. Richtiger Techno-Punk!

Radiogeräte anzuschließen. Es gab später sogar weitere Versionen mit eingebautem Radio, als Koffergerät, zum Einbau ins Auto, als Musikschrank oder TV-Kombination und sogar als Jukebox.

Bei all seinen Vorteilen hatte das Tefi-System aber auch einen großen Nachteil: Man konnte nicht gezielt auf bestimmte Musikstücke zugreifen. Es gab zwar eine recht genaue Anzeige für die Tonabnehmerhöhe, aber den richtigen Anfang zu treffen war reine Glückssache. In frühen Geräten gab es einen richtigen breiten „Lichtzeiger“ für die Tonabnehmer-Position (siehe **Bild 8**), doch später begnügte man sich mit einer mechanischen Anzeige.

Zur Spurauswahl schob man das Stellrad (**Bild 9**) nach rechts, um den Kristall vom Band abzuheben. Dann drehte man am Stellrad und verstellte damit die vertikale Position des Kristalls, die auf einer Skala abgelesen werden konnte. Alternativ dazu konnte man mit Hilfe einer optionalen Kabelfernbedienung magnetisch von Spur zu Spur springen.

Ein Detail am Rande: Die Herstellung der Bänder war sehr aufwändig. Zuerst wurden die Rillen geschnitten und dann von einer Matrize gepresst. Knackpunkt war die Stoßstelle des Bandes. Diese wurde von Hand geklebt und die Rillen nachgeprägt. Das war wohl auch der Grund, dass man auf die zunächst geplante Version als „Möbius-Band“ (einmal verschränkt) verzichtete, obwohl man dadurch beide Bandseiten nutzen und so die Kapazität hätte verdoppeln können.

Die Plattenspielererweiterung

Mit der Verbreitung der Single aus Vinyl gab es harte Konkurrenz. Alle berühmten Künstler waren damals bei den großen Plattenlabels unter Vertrag, und Letztere hatten kein Interesse an einem zweiten Tonträgersystem. Somit musste man sich beim Tefifon-Programm mit weniger bekannten Künstlern begnügen. Erst Anfang der 60er Jahre durfte das Philips-Repertoire verwendet werden. Außerdem erwies sich die lange Laufzeit der Kassette plötzlich als Pferdefuß. Die Single mit ihren Schlagern und Hits war modern und preiswert. Für ein Schallband musste man genug Repertoire haben, um dessen Länge füllen zu können.

Daher behalf man sich zunächst mit einem Trick: Eine Adapterkassette mit über ein Reibrad angetriebenem Plattenteller sowie ein nachrüstbarer Tonarm verwandelten das Tefifon in einen Plattenspieler (**Bild 10**). Um der Single und der LP Paroli zu bieten gab es später noch eine 18-Minuten-Kassette und sogar einen 3-Minuten-Adapter (**Bild 11**). Er musste von Hand per Stellschraube gespannt werden. Diese komplizierte Handhabung vermasselte den Erfolg.

Langsames Verschwinden

Das Tefifon konnte trotz Stereoausführung mit dem schnellen technischen Fortschritt nicht mithalten. Die Vinylplatte erreichte mit 40...50 dB einen besseren Störabstand. Beim Tefifon übertrugen sich Geräusche des Bandwickels und des Antriebs, weshalb hier nur etwa 35 dB möglich waren. Auch beim Gleichlauf hatte die Platte die Nase vorn.

Als dann in den 1960ern auch noch die Kompakt- oder Musik-Kassette auf den Markt kam, war das Ende der Schallband-Ära vorgezeichnet. Für Langzeitbeschallungen in Kaufhäusern, Bars und Kinos wurde trotzdem lange Zeit noch das Tefifon eingesetzt.

1965 wurde die Produktion eingestellt und das Tefi-Werk geschlossen. Restbestände übernahm das damalige Versandhaus Neckermann. Dr. Daniel hatte die Firma schon vorher verlassen und fertigte unter neuem Namen Schaltelemente für Kernkraftwerke.

In der Stadt Köln-Porz erinnert heute nichts mehr an den ehemaligen Standort des Tefi-Werks. Noch heute findet man im Internet eine Menge Infos. Etliche Liebhaber alter Audiotechnik haben Tefifon-Geräte restauriert, um ein Stück Audiogeschichte zu erhalten.

(150372)

Dank

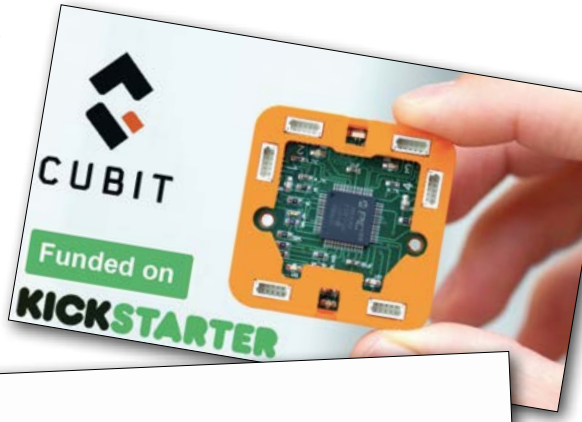
Der Autor bedankt sich beim „Klingenden Museum Riedenburg“ für die freundliche Unterstützung.

„Retronik erinnert an vergangene Technik und lässt Elektroniker schmunzeln.“

Sammeln Sie Geld!

Mit Kickstarter & Co.

Von **Harry Baggen**



Crowdfunding liegt im Trend der Zeit. Gehören Sie zur Gemeinde der innovativen Entwickler, haben Sie zukunftsträchtige Ideen, doch für die Umsetzung fehlt das Startkapital? Zeigen Sie der Welt, was Sie können, laden Sie die Welt ein, in Ihr Vorhaben zu investieren! Plattformen wie Kickstarter sind Ihre Arena.

Nicht mehr so freigiebig wie früher sind die Banken, wenn es um die Vergabe von Krediten geht, insbesondere an junge, aufstrebende Unternehmer. In diese Bresche sind Plattformen wie Kickstarter und Indiegogo gesprungen, sie konnten in den letzten Jahren einen enormen Run verzeichnen. Das Kalkül, das dahinter steht, ist simpel, aber effizient: Präsentieren Sie der globalen Internet-Community Ihr avisiertes Produkt oder Ihre Dienstleistung, geben Sie vor, wie viel Startkapital Sie brauchen. Die Internet-Community entscheidet, ob Ihr Vorhaben als investitionswürdig oder aber als floplastig erachtet wird. Generell kann jeder Internet-Nutzer zum Investor werden, oft ist auch die geringste Beteiligung willkommen. Doch es sind nicht nur couragierte Neugründer und Einzelkämpfer, die das Crowdfunding für sich entdeckt haben. Zunehmend gehen auch etablierte Unternehmen diesen Weg, um die Markteinführung innovativer Produkte zu finanzieren und die Risiken zu streuen. Es ist schon erstaunlich, wie kreativ die Mitmenschen sind und welche bemerkenswerten Ideen sie haben. Auch das Terrain der Elektronik gehört, wie könnte es anders sein, zu den Arealen der Crowdfunder. Ein beispielhaftes Highlight ist das „SmartScope“ der belgischen Ideenschmiede LabNation [1], das inzwischen sogar Eingang in den Elektor-Store gefunden hat. Vor rund zwei Jahren startete das Unternehmen eine Crowdfunding-Aktion auf Kickstarter. Es vergingen nur wenige Monate, bis ein Vielfaches des angesteuerten Startkapitals zusammengetragen war. Die Produktion eines ultimativen USB-Oszilloskops konnte beginnen. Wir haben uns einige weitere Produkte angeschaut, die auf Crowdfunding setzen, unsere Auswahl kann allerdings nur willkürlich sein.

In den Medien wurde bereits über den „Panono“ berichtet, einen Mini-Ball, dessen Kugelfläche mit Mini-Kameras gespickt ist [2]. Die Kameras sind in der Lage, die Umwelt visuell im Raumwinkel von 360° zu erfassen. Das Projekt hat den

Weblinks

[1] www.kickstarter.com/projects/751733865/smartscope-reinventing-the-oscilloscope
[2] www.indiegogo.com/projects/panono-panoramic-ball-camera
[3] www.indiegogo.com/projects/mars-by-crazybaby
[4] www.indiegogo.com/projects/zboard-2-the-most-advanced-electric-skateboard
[5] www.kickstarter.com/projects/749835103/hamshield-for-arduino-vhf-uhf-transceiver
[6] www.kickstarter.com/projects/1762626887/cubit-the-make-anything-platform

Machern inzwischen Gewinne in Millionenhöhe beschert.
Nicht nur dem Ohr, sondern auch dem Auge schmeichelt „Mars“, ein drahtloser Bluetooth-Lautsprecher für die Audio-Wiedergabe mobiler Player [3]. Mars schwebt gleich einem Raumschiff über seiner Basis. Dort lässt er sich nieder, wenn sein Akku Energie nachtanken muss, um anschließend wieder aufzusteigen.
Elektrisch angetriebene Skateboards sind in Mode, im Web stehen diverse Konzepte und Konstruktionen zur Diskussion. Überzeugt hat uns die Leistung des „ZBoard 2“ [4], das den Skater mit nur einer Akkuladung fast 40 km weit trägt und eine Geschwindigkeit bis 30 km/h erreicht.

Das „HamShield“ von Casey Halverson ist ein Mini-Transceiver für den Amateurfunk im VHF-Bereich, es wird auf einen Arduino aufgesteckt [5]. Der RF-Transceiver-Chip 1846S des in Hong Kong ansässigen Spezialisten Auctus verleiht diesem Shield mit einem Software-Defined-Radio-Kern bemerkenswerte Fähigkeiten.
Schließlich soll noch „Cubit“ erwähnt werden, ein modulares Entwicklungssystem mit visueller Programmierumgebung [6]. Laut seinen Erfindern ist das Entwickeln von Applikationen möglich, ohne eine Programmzeile schreiben oder ein Bauteil löten zu müssen. Ein vielversprechender Ansatz!

(150375)gd

Das Web ist Ihre Bühne!

Drohnen im Anflug!

Drei bemerkenswerte Konstruktionen

Von Harry Baggen (Redaktion NL)

Quadrocopter und ihre Ableger sind ein aktuelles Thema, das gegenwärtig nicht nur die junge Generation beschäftigt. Sogar gestandene Fachleute aus der Flugzeug- und Elektronik-Branche hat der Drohnen-Hype erfasst. Die mehrmotorigen Flugobjekte gibt es inzwischen in zahllosen Varianten. Wir haben uns im Netz umgesehen und manches Erstaunliche entdeckt.

Das Fliegen fasziniert die Menschheit seit ihren Ursprüngen. Es kommt deshalb nicht von ungefähr, dass sich der Modellflugzeugsport hoher Beliebtheit erfreut. Eine trendige Fortentwicklung dieser Beschäftigung sind die Quadrocopter, auch als „Drohnen“ bekannt. Die Palette reicht von Modellen aus dem Kinderspielzeugladen bis hin zu professionellen Fluggeräten, die aus luftigen Perspektiven Städte, Gebäude oder Landschaften fotografieren. Die unbemannten, militärisch eingesetzten Flugkörper, die ebenfalls „Drohnen“ genannt werden, wollen wir hier außer Betracht lassen.

Zivile Drohnen sind in vielen Größen, Ausstattungen und Preisklassen auf dem Markt. Dazu gehören auch einige Sonderformen, die mit ihren Konstruktionen, Ausstattungen oder Funktionen aus dem Gros der Masse herausragen.

Zusammenklappbar

Quadrocopter, insbesondere die größeren Modelle, sind vergleichsweise sperrig. Weil aus der Silhouette die zum Starten und Landen notwendigen Stützen herausragen, lassen sie sich oft nur mit höchster Vorsicht transportieren. Bei einigen Modellen können die Propeller demontiert werden, doch auch dann sind noch die Stützen im Weg. Einige Hersteller haben das Problem entschärft, indem sie clevere Zusammenklapp-Mechanismen ersonnen haben.

Die Mehrzahl der Konstruktionen ist dem gewöhnlichen Regenschirm nachempfunden, doch wie der Drohnen-Spezialist Droidworx an seinem Modell „Blu” zeigt, geht es auch anders. Dort wurde ein Quadrocopter entwickelt, der beim Transport wie eine kleine Blackbox aussieht [1]. In eingeklapptem Zustand passen die Stützen exakt in dafür vorgesehenen Nuten des Rumpfs, ebenso wie die Propeller, die außerdem ineinander gesteckt werden können. Auf der Vorderseite des Rumpfs sind das Bedienfeld und eine HD-Kamera angeordnet. Steuern lässt sich der Quadrocopter über eine Fernbedienung oder ein Android- oder iOS-Gerät. Zum Fotografieren von „Selfies“ (Selbstportraits) dient eine spezielle Einstellung, bei der die Drohne einen definierbaren Abstand zum steuernden Gerät einhält. Auf den ersten Blick wirkt diese Drohne mit ihrer fragilen Gestalt empfindlich gegen mechanische Beanspruchung. Doch Materialien wie Nylon und karbonverstärkter Kunststoff geben ihr eine erstaunliche Robustheit. Die nötige Rechenkraft verleiht der „Blu“ ein 32-bit-Mikrocontroller STM32F7, und außerdem ist ein schneller GPS-Empfänger an Bord. Die „Blu“ kostet rund 500 $, als Zubehör ist ein Bluetooth-Controller in Gestalt eines Armbands lieferbar.

Lange Leine

Welchen Sinn kann es haben, einen Quadrocopter an einer Leine spazieren zu führen? Jedenfalls wurde dieses Konzept beim Photokite Phi realisiert, ein Projekt, das über Crowdfunding auf Indiegogo an den Start ging [2]. Dieser Quadrocopter schwebt geführt an einer Schnur, er ist weder über eine Fernbedien-

ung noch ein Smartphone steuerbar. Die Reichweite der Schnur beträgt einige Meter. An ihrem Ende befindet ein Bedienteil, mit dem die Flughöhe und die Drehung gewählt werden können. Mit der Schnur ist das Manövrieren erstaunlich präzise möglich, was Fotoaufnahmen in Innenräumen aus großer Höhe, beispielsweise in Kirchen möglich macht. Die drahtgebundene Drohne sieht zuerst etwas befremdlich aus, doch in diesem Quadrocopter stecken einige wirklich originelle Ideen. Die mechanische Konstruktion ist pfiffig durchdacht, die klappbaren Stützen lassen sich mit einem Handgriff ausfahren und mit einem Drehknopf verriegeln. Die Halterung an der Unterseite nimmt eine Kamera vom Typ GoPro Hero auf, der Blickwinkel ist variierbar. Unterbringen lässt sich die ungewöhnliche Drohne in einem Behälter von der Größe einer Thermosflasche.

Fast wie Drachensteigen: Eine Drohne an der Leine.

Auf dem Wasser

Die AguaDrone ist ein Quadrocopter, der für einen verblüffenden, aber nahe liegenden Einsatzzweck konstruiert wurde. Leider kam beim Crowdfunding auf Kickstarter nicht genügend Startkapital zusammen, doch der Konstrukteur hofft, dem unkonventionellen Fluggerät auf anderem Weg zur Serienproduktion zu verhelfen. Die Drohnen-Spezialkonstruktion ist auf die Tätigkeiten von Anglern zugeschnitten. Der Angler kann die Drohne über das Gewässer rundfliegen lassen, sie kann auf der Wasseroberfläche landen. Das Sonarsystem auf der Unterseite der schwimmenden Drohne spürt Fische im Wasser auf. Wird ein größerer Fisch oder ein Fischschwarm entdeckt, überträgt das System die Position zum Smartphone des Anglers. Der Angler beordert die Drohne zurück und tauscht das Sonarsystem gegen das Angelsystem mit Angelleine und Angelhaken aus. Die Drohne fliegt zum Fischgrund zurück, lässt die Angelleine in das Wasser hinab - und dann „Petri Heil“! Ist das nicht der Traum jedes technikbegeisterten Anglers?

Mit Intel Edison

Die Eedu, die Easy Educational Drone Unit, ist ein Quadrocopter, der als Lehr- und Lernprojekt konzipiert wurde [3]. Die Mechanik lässt sich unkompliziert zusammenbauen und auseinandernehmen, die Elektronik ist eine Spielwiese für spannende Experimente. Auf der Grundplatine hat nicht nur ein schneller Mikrocontroller ARM Cortex M7 seinen Platz, sondern auch ein Intel-Edison-Modul. Ferner ist Raum vorhanden, um gängige Arduino-Shields zu montieren. Die so geschaffene Plattform bietet viel Freiheit für Hardware-Konfigurationen und Software-Applikationen. Das Programmieren gestaltet sich unkompliziert dank einer speziellen Entwicklungsumgebung für Roboter (RDE), genannt „Forge“. Damit können auch Newcomer ohne Umweg Programme entwickeln oder modifizieren. Leider wurde auch bei diesem Projekt das Ziel für das Startkapital bei Kickstarter nicht erreicht [4], so dass nun abzuwarten bleibt, ob die Drohne Eedu doch noch in Produktion geht.

Anti-Drohnen

Die Invasion der Drohnen hat nicht wenige Menschen verunsichert, sie fürchten um ihre Privatsphäre. Insbesondere in den USA war dieses Phänomen der Anlass, Drohnen-Abwehrsysteme unterschiedlichster Art zu konstruieren. Zum Beispiel kam ein Gerät höherer Preisklasse auf den Markt, das in der Lage ist, die Funkverbindungen von Drohnen folgenschwer zu stören. Eine noch rigorosere Methode ist die Anti-Drohnen-Patrone [5], die zielgerichtet abgeschossen, die „feindliche“ Drohne in einem künstlichen Hagelschauer zu Boden zwingt...

(150492)gd

Weblinks

[1] www.mydroidworx.org/blu.html
[2] www.indiegogo.com/projects/fotokite-phi-tethered-uav-for-aerial-filming
[3] http://skyworksas.com/eedu/
[4] www.kickstarter.com/projects/1528442751/first-water-proof-drone-with-sonar-fish-finder
[5] http://snakerivershootingproducts.com/

Hexadoku

Sudoku für Elektroniker

Deutschland gegen Frankreich heißt es ab jetzt bei unserem Hexadoku – der Rest der Welt spielt in einer anderen Liga. Wegen der unterschiedlichen Erscheinungstermine der Hefte mussten wir zwei Gruppen von Hexadoku-Ländern bilden, dabei gibt es jeweils drei Gutscheine zu gewinnen. Sonst ändert sich nichts - der Knobelspaß bleibt in jedem Fall der alte!

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

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

Einsenden

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

Elektor Redaktion
Süsterfeldstr. 25
52072 Aachen

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

Als Betreff bitte nur die Ziffern der Lösung angeben!

Einsendeschluss ist der 30. November 2015.

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

Die richtige Lösung ist: **6170C**

Einen Elektor-Wertgutschein über je 50 € haben gewonnen:

Marco Retti, Martin Heyer, Marc Moulin, Gerald Kreisler und Frédéric Hay.

Herzlichen Glückwunsch!

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

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

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

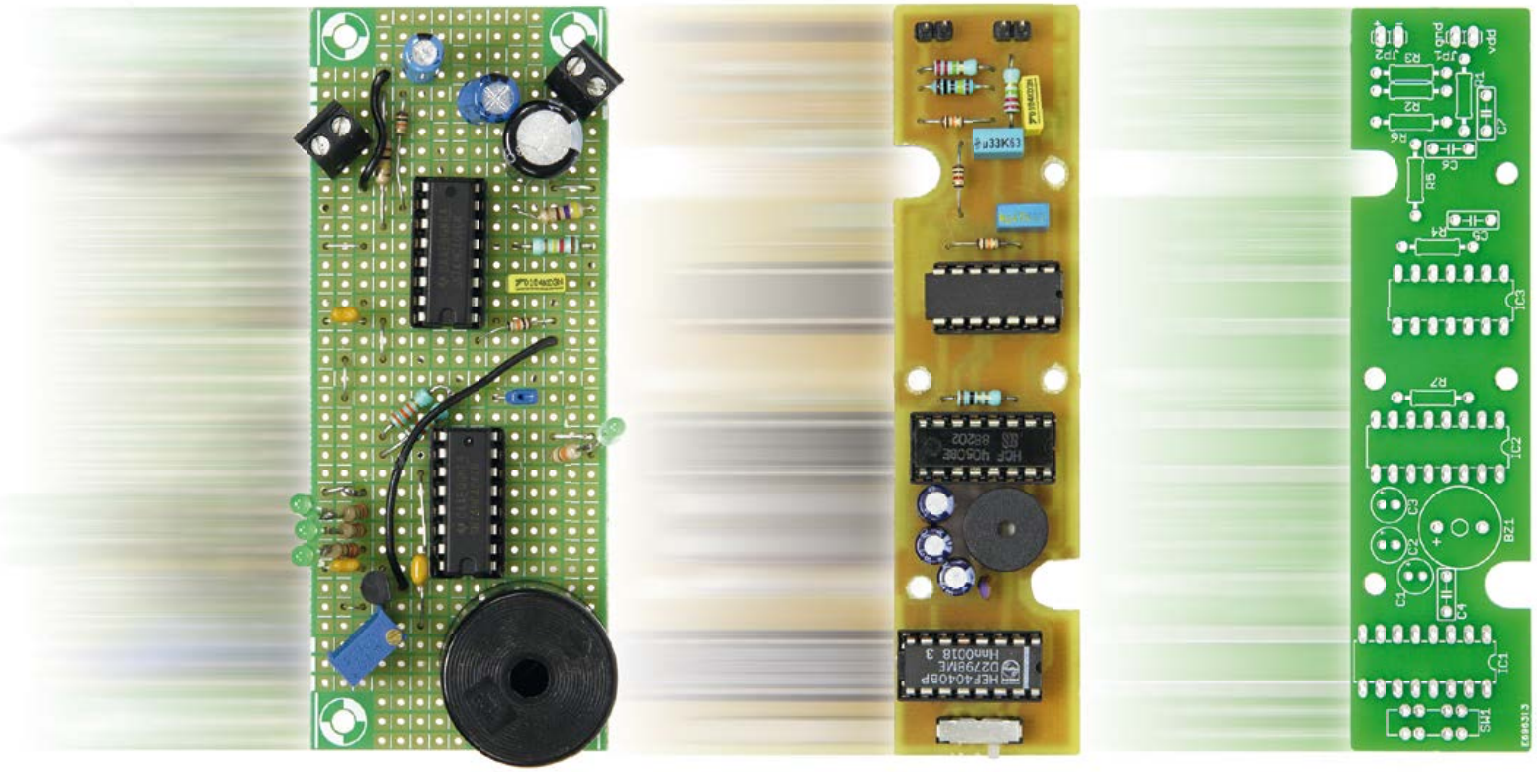
LEARN DESIGN SHARE

elektor PCB Service

Powered by EURO CIRCUITS

Platinen – Prototypen – Multilayer – Kleinserien

Kostengünstig! **Top-Qualität!** **Zuverlässig!**



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

Bestellen Sie Ihre Platinen jetzt unter:

www.elektorPCBservice.de

Professionelle Qualität zu attraktiven Preisen!

reichelt.de elektronik

Jetzt kostenlos anfordern!

Katalog 01|2016 »»»

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

2x 1. Platz

in der Kategorie
„Produktverfügbarkeit“
in den Produktbereichen
„Optoelektronik / Displays“
und „Passive Bauelemente“.

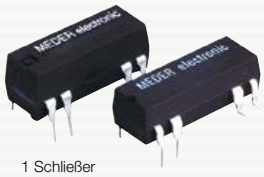
Elektronik Distributor des Jahres 2015

Reedrelais

Standex MEDER ELECTRONICS

Dual-In-Line

- mit Diode
- IC-Pin kompatibel
- umpresste DIP-Version
- niedrige Bauform
- 1 A



DIP 7212-D 5V	1,40	1 Schließer
DIP 7221-D 12V	2,55	2 Schließer

Single-In-Line

- hohe Spulenwiderstände
- Schleifenstromerkennung
- 1 Schließer

SIL 7271-L 5V	1,20	ohne Diode
SIL 7271-D 5V	1,50	mit Diode
SIL 7271-D 24V	1,50	mit Diode

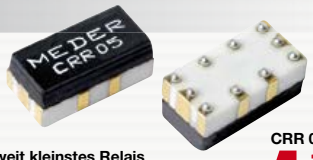
Reed-Kontakt

- 1 Schließer
- im Glaskörper
- 0,5 A
- Ø 2,2 mm

KSK 1A66

0,39

SMD



- weltweit kleinstes Relais
- 1 Schließer
- 0,5 A

CRR 05-1A

4,15

Schaltuhr steuern per Smartphone

finder

Dank der NFC (Near Field Communication) und der App „FINDER Toolbox“, die auf Google Play kostenlos zum Download bereitsteht, können Sie diese Zeitschaltuhren über ein NFC-fähiges Smartphone programmieren.

Lieferung ohne Smartphone!

ANDROID APP BEI Google play

- Programmierung über „Smart“ mit NFC oder „Classic“ mit dem Joy-Stick
- LCD-Anzeige mit Hintergrundbeleuchtung
- Joystick-Programmierung
- 1 Wechsler für 16 A
- Kontaktmaterial AgSnO2
- Anschlussspannung 230 V AC
- Tages-und Wochenprogramm mit 6 Jahren
- Gangreserve
- automatische Sommer- und Winterzeitumschaltung

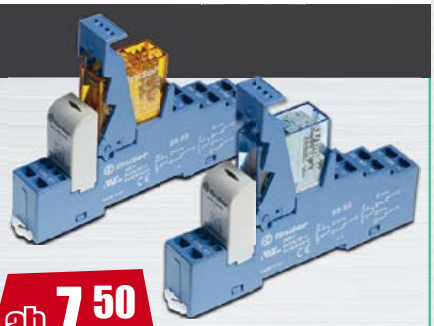
FIN 12.51 230V

39,95

Finder Koppelrelais

Fertig montierte Koppelrelais für den Verteilereinbau mit LED-EMV-Modul und (De-) Montagebügel.

FIN 48.31.8 24V	9,65	24 V AC	1 Wechsler	10 A
FIN 48.52.8 24V	7,50	24 V AC	2 Wechsler	8 A
FIN 48.31.8 230V	9,30	230 V AC	1 Wechsler	10 A
FIN 48.52.8 230V	9,50	230 V AC	2 Wechsler	8 A
FIN 48.72.9 24V	11,05	24 V DC	2 Wechsler	8 A
FIN 48.72.8 230V	13,95	230 V AC	2 Wechsler	8 A



ab **7,50**

Steckrelais

Steckrelais, 2x UM

- RM 5,0 mm
- 2 Wechsler, 250 V / 8 A

ab **2,25**

FIN 40.52.9 24V	2,25	24 V DC	2 Wechsler	8 A
FIN 40.52.9 12V	2,25	12 V DC	2 Wechsler	8 A
FIN 40.52.8 230V	4,20	230 V AC	2 Wechsler	8 A

JETZT ABONNIEREN!

Newsletter

Abbonieren und profitieren

- Neuheiten
- beste Angebote
- Preisreduzierungen

Jetzt bestellen! **www.reichelt.de**

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

Jetzt bestellen!

Tagespreise! Preisstand: 23.09.2015

Preise in € inkl. gesetzl. MwSt., zzgl. Versandkosten
reichelt elektronik, Elektronikring 1, 26452 Sande (D)

reichelt.de

@reichelt_el