

2/2015

ct Make: **Make:**

Schnell & einfach:
Dritte
Hände



Grundlagen und Bauprojekte

Elektronen- röhren



Make:

Tastatur-Upcycling
GPS mit Arduino
DIY-Mondmission

Laser-Upgrade für Fräsen
Analoge Beat-Maschine

ROBOTER

Konstruieren & Programmieren

2/2015

CH CHF 17,50
AT, Benelux,
IT, ES € 10,90

€ 9,90



Heise

www.make-magazin.de

Anzeige

Durchblick

Sie, liebe Leserinnen und Leser, haben es sicher schon bemerkt: Hier stellen sich die Neuen vor und haben am besten eine schöne Anekdote parat. Tatsächlich passieren die hier öfter. Man muss sie nur im Blick behalten, was eine deutlich größere Herausforderung ist.

Denn ich gebe zu: nach nicht mal drei Monaten im Make-Team ist mein Schreibtisch bereits ziemlich vollgestellt. Zerlegte Tastaturen, Widerstände auf Breadboards, diverse Kabel und eine Kiste mit Verpackungsmaterial, das mal ein Roboter werden soll. Wenigstens vor dem Bildschirm ist aber Platz. Bis ich neulich morgens ins Büro kam. Auf einmal standen da zwei Rollen Chips mit Butterbrotpapier. Das klärte sich zum Glück schnell mit einer E-Mail vom Chef. „Ist doch bald die Sonnenfinsternis, bau mal einen Sonnenbeobachter.“ Ich musste noch Alufolie kaufen, auch das ist mir im Arbeitsleben bisher nicht passiert.

Mit dem Durchblick war das bei dieser partiellen Sonnenfinsternis dann so eine Sache. Die speziellen Sonnenbrillen waren Tage vorher ausverkauft und meine erste Version des Beobachters geriet leider etwas unscharf. Das Loch in der Blende war zu groß. Immer noch besser als bei einer Anleitung aus dem Internet für den Schuhkarton. Dort fehlte die Lochblende ganz. Mein Kollege Peter König baute den Beobachter nach und fragte sich eine Weile, wie das denn funktionieren könne. Na ja, gar nicht! Nach der Erkenntnis war die Blende schnell mit etwas Alufolie nachgerüstet.

Am 20. März konnten wir mit den verschiedenen Konstruktionen schließlich die partielle Sonnenfinsternis beobachten. Wir hatten sogar eine Brille von 1999 – die aber bald von Kollegen im Haus ausgeliehen wurde und erst spät den Weg zurückfand. Besonders haben wir uns an dem Tag über Twitter gefreut, weil uns (@MakeMagazinDE) viele Follower Bilder von ihren Nachbauten und Eigenkreationen geschickt haben.

Nach diesem Spaß heißt es nun, sich der nächsten Herausforderung zu stellen: Ich werde meinen Schreibtisch aufräumen. Für mehr Durchblick!

Helga Hansen

Helga Hansen



Sagen Sie uns Ihre Meinung!
mail@make-magazin.de



ROBOTIK

Kompliziert, aber geil: Unser Schwerpunkt liegt diesmal auf der Robotik. Auf 46 Seiten haben wir vom kompletten DIY-Roboter über Bausätze bis hin zu einzelnen Komponenten alles getestet und nachgebaut, was das Robot-Herz braucht.

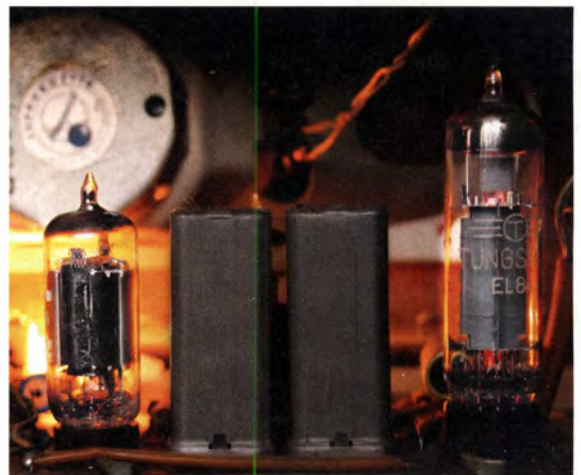
- 28 Roboter-Bausätze im Test
- 38 Roboter-Arm mit Objekt-Tracker steuern
- 44 Aktuelle Mikrocontroller-Boards
- 54 Abstands-Sensoren mit Arduino programmieren
- 62 Gesichtserkennung mit Python und Webcam
- 66 Einfaches Schwarmverhalten programmieren
- 74 Günstiger Roboter zum Nachbauen

Inhalt

ELEKTRONENRÖHREN

Röhren wecken häufig Assoziationen wie „altbacken“ und „unverständlich“. Nicht bei uns: Wir haben anschauliche Erklärungen und Anwendungsbeispiele.

110 Röhrentechnik für Einsteiger



- 3 Editorial
- 6 Leserforum
- 8 **Werkstattberichte:** FabLab-News und Szeneberichte
- 10 **DIY-Lasercutter:** Wie man den Laser aus dem Tetris-Projekt (Make 1/2015) baut
- 14 **Analoge Beatmaschine:** Elektronisches Schlagzeug aus passiven Bauteilen
- 18 **Part Time Scientists:** Mondmission aus privater Hand
- 26 **Lieblingswerkzeug:** DIY-Prüfstift
- 28 **Robotik-Kits:** Sieben Plattformen im Test
- 38 **Roboter-Arm** mit Objekt-Tracker steuern
- 44 **Mikrocontroller:** Übersicht über aktuelle Boards für die Robotik
- 54 **Abstands-Sensoren** mit Arduino programmieren
- 62 **Gesichtserkennung** mit Python und Webcam
- 66 **Schwarmverhalten:** Einfache Programmier-Anleitung

MENSCHEN

Manche Menschen machen Dinge, die die meisten für schlicht unmöglich halten. Wir haben Maker porträtiert, deren Produktivität und Unermüdllichkeit uns beeindruckt hat.

18 Part Time Scientists:
Makerchens Mondfahrt

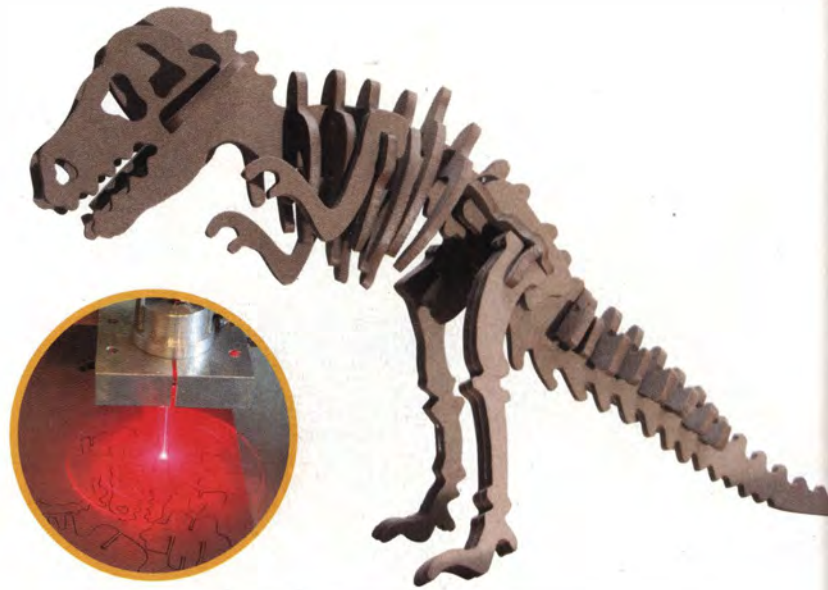
82 Mario Lukas: „Selbermachen kann jeder!“



PEW PEW: LASERFRÄSER

In unserem Artikel über einen Tetris-Nachbau mit LEDs aus dem letzten Heft erwähnte der Autor seinen Fräsenachbau mit Laser. Hier kommt der Baubericht.

10 Lasercutter: Laser trifft Dino

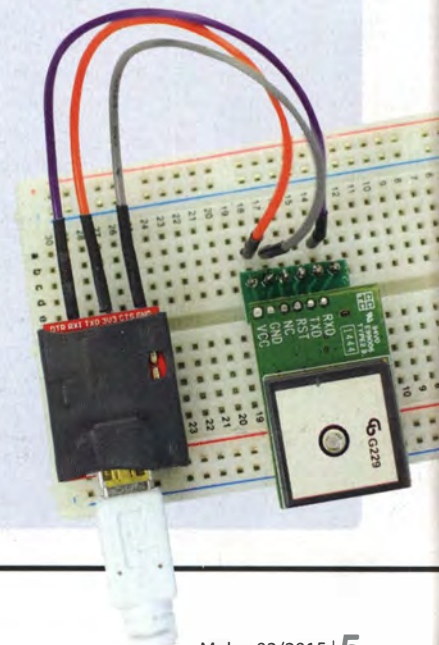


- 74** **DIY-Roboter:** Bauanleitung für günstigen Roboter
- 80** **Reingeschaut:** Gameboy Pocket
- 82** **Mario Lukas:** Multi-Bastelmaschine im Interview
- 90** **Community-Projekte:** 3D-Drucker Open Exposer und eine GPS-Mystery-Box
- 94** **GPS-Empfänger** mit Arduino
- 102** **Bauanleitung** für einfaches Helfendes-Händchen
- 106** **Was uns inspiriert**
- 110** **Grundlagen:** Elektronenröhren mit Baubeispielen
- 122** **Buchkritiken**
- 124** **Lieblingswerkzeug:** Transistortester Testy
- 126** **Wind messen** mit einem Anemometer
- 132** **Basteln:** Neues Leben für alte Tastaturen
- 138** **Batteriemonitor** zeigt Ladezustand von Bleiakkus an
- 146** **Superzeitlupe** mit GoPro
- 152** **Kurzvorstellungen:** HDMI-Displays mit Touch-Controller, HelvePic32, Chibitronics
- 154** Impressum

ELEKTRONIK BASTELN

Sie wollen mit der Make gleich in den Bastelkeller abtauchen? Wir haben schöne Elektronik-Projekte in petto:

- 14** Boom, Boom, Tschack: Elektronisches Schlagzeug
- 54** Sensoren am Arduino
- 94** GPS-Navigation mit Arduino
- 138** Batteriemonitor



LESERFORUM

Fehler in der Make 1/15

Tetris-Klon mit LED-Streifen,
Ausgabe 1/2015, S. 12

Mir ist ein Fehler in der aktuellen Make auf Seite 16 aufgefallen. Die Zeichnung der Verdrahtung des Arduinos mit den LED-Streifen ist falsch. Die Verbindungen vom Daten-Ausgang des einen LED-Streifens verlaufen in den +5-Volt-Eingang des nächsten Streifens. Diese sollten doch laut Anleitung in den Daten-Eingang des nächsten Streifens gelangen.

Thorsten Bergler

Sie haben Recht. Wir entschuldigen uns für den Fehler und haben im Heft-Forum auf unserer Webseite darauf hingewiesen. —dab

Nachbau Laser im Abflussrohr

Tetris-Klon mit LED-Streifen,
Ausgabe 1/2015, S. 12

Was steckt da für ein Laser in diesem PVC-Rohr? Gibt es da schon zugängliche Informationen, oder muss ich mich bis zum n+ixten Heft gedulden?

U. Kliegis

Da uns mehrere Anschreiben mit dieser Frage erreicht haben, haben wir den Autoren gebeten, hierfür noch einen kleinen Text zu verfassen. Sie finden den Artikel von Herrn Schmerold hierzu auf Seite 10. —dab

Großes Lob

Der individuelle Quadrokopter,
Ausgabe 3/2013, S. 54

Es liegt schon lange zurück, aber mit Ihrer Idee zum Quadrokopter-Bau haben Sie nicht

nur meine „großen“ Oberstufen-Schüler von der „Make“-Idee angesteckt und für eine sensationelle Abitur-Prüfung gesorgt. Mittlerweile verfügt die Schule über einen eigenen 3D-Drucker (i3-Berlin) sowie zahlreiche Arduino-Boards, die im Informatik-Unterricht der Oberstufe sowie in AGs zum Einsatz kommen. Daher möchte ich Ihnen auf diesem Wege einfach „Danke“ sagen. Danke für Ihre Ideen und Inspirationen aus der Make, die mittlerweile fester Bestandteil des Informatik-Unterrichts sind und meine Schüler und insbesondere auch Schülerinnen lang anhaltend begeistern!

Sascha Vorwerk

Drei-Finger-Regel links?

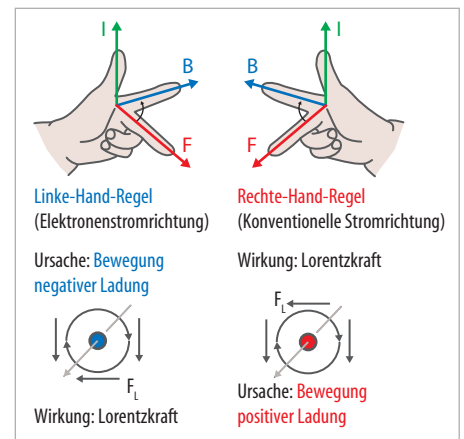
Die Gesetze der Anziehung,
Ausgabe 1/2015, S. 126

Beim ersten Durchblättern der Zeitschrift, ließ mich die Abbildung für den Artikel stützen. Aus meiner Schul- und Studienzeit kannte ich die Drei-Finger-Regel. Bei uns hieß sie aber immer explizit: Drei-Finger-Regel-der-rechten-Hand (analog zur Rechte-Faust-Regel). Die Abbildung zeigt eine linke Hand. Was war falsch? Ihre Autorin hat Recht und meine Erinnerung auch. Allerdings nutzen sie die in Deutschland wohl eher weniger benutzte Merkregel, die wohl eher im angloamerikanischen Raum bekannt sein dürfte. Vielleicht stammt auch der Artikel daher?

Florian Schäffer

Nein, der Artikel stammt direkt aus meiner Feder. Allerdings habe ich in der Recherche einige internationale Quellen verwendet, was wohl dazu führte, dass ich übersehen habe, dass in Deutschland die Rechte-Hand-Regel wesentlich gebräuchlicher ist. Mir persönlich ist

als Linkshänderin die Linke-Hand-Regel wesentlich eingängiger. Die Grafik unten macht die Unterschiede deutlich. —esk



Die linke Hand zeigt die physikalische Stromflussrichtung, die rechte Hand die technische.

Bierbrauen

Homebrew mal wörtlich,
Ausgabe 1/2015, S. 32

Ich bin privat ein Maker und beruflich Bierbrauer und habe mich natürlich besonders auf Euren Bier-Artikel gefreut. Einen kleinen Kritikpunkt habe ich: Das Aufspeisen zur Flaschengärung mit Kristallzucker zu erledigen ist nicht schön. Ich bin der Meinung, dass Zuckergabe zur Nachgärung zu einem sehr flachen Geschmack führt. Man kann auch einen kleinen Teil seiner Würze eingefrieren (je nach Stammwürze, Endvergärungsgrad und gewünschter Reszenz ca. 5–10 %) und diese dann direkt vor dem Abfüllen wieder zugeben. So kommt man um den Zucker herum. Geschmacklich recht interessant ist es übrigens, auch beim Aufspeisen und davor schon mit Honig zu arbeiten, aber besser nicht bei einem Pale Ale, da der Honig dann vom Hopfen erschlagen wird.

Sven Wagner

Das mit der Zuckergabe war nicht meine Idee, sondern steht so in der Brauanleitung des verwendeten Brausets. Ich werde Ihren Tipp aber garantiert einmal ausprobieren. Eine gute Idee, danke! —phs


Kontakt zur Redaktion

Leserbriefe schicken Sie bitte an
mail@make-magazin.de
oder direkt an den Redakteur.

Sie haben auch die Möglichkeit, in unseren Foren online über Themen und Artikel zu diskutieren:

www.make-magazin/forum

 www.facebook.com/makemagazin.de

 plus.google.com/+ct-HacksDe

 www.twitter.com/MakeMagazinDE



Manchmal bringen uns unsere eigenen Fehler zum Lachen. Vielen Dank für diesen Hinweis!

LED-Stripes nicht endlos

Tetris-Klon mit LED-Streifen, Ausgabe 1/2015, S. 12

Bezüglich Ihrer Anmerkung wegen der reparierten LED-Stripes im aktuellen Heft aus Seite 14 links: Nicht jeder Hersteller kann 5 m Folienplatinen in einem Stück produzieren. Daher werden die kurzen Stücke zu einem langen Stripe zusammengelötet.

Siegi Jank

Ich denke, ich habe mich im Artikel etwas missverständlich ausgedrückt. Mit „repariert“ wollte ich nicht auf ein Qualitätsproblem hinweisen, sondern zum Ausdruck bringen, dass „falsch

reparierte“ LED-Stripes ungeeignet sind für den LED-Bildschirm. Da die Streifen bei der „Reparatur“ überlappend angelötet werden, verändern sich die Abstände der LEDs. Dies führt dann zu einem unregelmäßigen Bild. „Falsch repariert“ war nicht abwertend gemeint – als LED-Stripe-Hersteller würde ich es ebenso machen.

Ulli Schmerold

Welche LED beim Upgrade für Taschenlampen?

LED-Upgrade für Taschenlampen, Ausgabe 1/2015, S. 38

Auf den Artikel „LED-Upgrade für Taschenlampen“ hab ich mich besonders gefreut. Eine Frage hätte ich aber dazu: Die meisten High-Power-LEDs, die ich gefunden habe, werden mit Spannungen um die 4 V betrieben und diejenigen, die mit mehr Spannung auskommen brauchen gleich 15 V oder mehr. Die im Artikel beschriebenen 6 V sind mir bei meiner „Schnell-Recherche“ bis jetzt nicht untergekommen. Könntet Ihr mir die genaue Typ-Bezeichnung und eine mögliche Bezugsquelle für genau die im Artikel verwendete LED nennen?

Benedikt Zettl

Nun, die von mir verbaute LED ist vom Segor: Bridgelux COB 3x3 Warmweiss 10 W/9–12 V.

Wichtig ist ja nur der Strom. Die Segor LED wird übrigens hier nur mit 700 mA betrieben. Man könnte natürlich auch 900 mA in der Software einstellen. LEDs mit 15 V sollten natürlich auch gehen. Überhaupt: Alle Bauteile lassen sich beim Segor bestellen.

Oliver Kraus

Richtigstellung

Kunst durch Technik, Ausgabe 1/2015, S. 50

Das Projekt „Orkest Ontdekker“ wurde nicht wie bei uns auf Seite 53 behauptet, vom Rotterdamer Staatstheater, sondern vom Rotterdamer Philharmonischen Orchester finanziert. —esk

C verloren

Arduino-Messgerät für Kondensatoren, Ausgabe 1/2015, S. 70

Im Kasten zur Kennlinie (S. 72) beschäftigen wir mit der Spannung zum Zeitpunkt $t = RC$. In der folgenden Formel ist uns das C dann abhandengekommen. Sie muss natürlich lauten:

$$U(t=RC) = U_0 \cdot \left(1 - \frac{1}{e}\right) = U_0 \cdot 0,632$$

—hch



Anzeige

FABLAB WÜRZBURG UMGEZOGEN

Der Space sitzt nun in der Veitshöchheimer Straße 14, gegenüber dem Kulturspeicher.

fablab-wuerzburg.de

MAKER- TERMINE

ThingsCon

8. & 9. Mai

Kulturbrauerei Berlin
thingscon.com

Chaos Cologne

16. & 17. Mai

KHM Köln
chaos.cologne

Maker Faire

Hannover

6. & 7. Juni

Hannover
Congress
Centrum
makerfaire.hannover.com

FabCon 3.D

10. & 11. Juni

Messe Erfurt
fabcon-germany.com

Pi and More 7

20. Juni

Uni Trier
piandmore.de

Diese und weitere Termine stehen auch laufend aktualisiert in unserem Kalender auf der Make-Website unter: www.heise.de/make/kalender

Veranstalten Sie selbst?

Tragen Sie Ihren Termin in unseren Kalender ein oder schicken Sie uns eine E-Mail an

mail@make-magazin.de



Ein richtiges Labor mit Abzug, Prägemaschine und Gashähnen. Hier ist noch Platz für mehr Geräte.

BASTELN, BACKEN, BRAUEN – AB JETZT AUCH IM HILDESHEIMER LABOR

Hildesheim hat einen neuen Ort für Kunst, Kultur und Technik

Hoch oben, im 4. Stock mit Blick über die Stadt, sitzt seit Mitte Februar das Freie Labor Hildesheim – in einem richtigen Labor im Technologiezentrum in der Nordstadt. Mit zwei Abzügen, einer Notdusche und Gashähnen an den Wänden. Anfang März feierte der Verein die Eröffnungsparty mit rund 50 Mitgliedern und Interessierten aus der Region.

Zunächst sahen die zwei Räume und drei Abstellkammern noch etwas spartanisch aus. Im Laufe des Abends kamen mit den Gästen aber auch die ersten Geräte. Auch wer etwas lernen wollte, kam auf seine Kosten. „Seid ihr schon bei Open-

StreetMap?“ führte zum ersten Workshop und den WLAN-Schlüssel gab es später als QR-Code. Ein Garderobenunglück sorgte für kurzzeitige Aufregung, es gab aber keine Verletzten.

Mit ins Labor eingezogen sind die Mitglieder des bisherigen Hackerspaces Hickerspace mit ihren Projekten. Bereits vorhanden sind derzeit eine Prägemaschine, Kappsäge und ein kleiner 3D-Drucker. Als erstes neues Projekt soll in Hildesheim Freifunk etabliert werden. Auch für musikalische Projekte und einen Workshop zum Thema Bierbrauen gibt es Interessierte.

—hch

► www.freieslabor.org

MAKERSPACE AUF DER RE:PUBLICA 2015

Global Innovation Gathering bringt
Maker aus aller Welt nach Berlin

Mitten auf der Internetkonferenz re:publica in Berlin, die dieses Jahr vom 5. bis 7. Mai stattfindet, wird zum dritten Mal in Folge ein kleiner Makerspace mit 3D-Druckern und LötKolben stehen. Das Global Innovation Gathering (GIG) stellt Technik-Projekte vor allem aus Asien, Afrika und Lateinamerika vor.

Am GIG-Makerspace können die Konferenzteilnehmer beispielsweise zusammen mit Tarek Ahmed vom FabLab Egypt in Gizeh einen Joule-Dieb bauen (siehe Make 1/15, Seite 44) oder von Mwangi Roy Ombatti vom FabLab an der Uni Nairobi lernen, wie man einen 3D-Drucker aus Elektronikschrott baut. Andere Workshops drehen sich um 3D-Druck und Elektronik in Textilien.

—phs

► www.re-publica.de



Auf dem Makerspace der re:publica kann man sich der Technik auch gestaltend nähern.

Bild: Gregor Fischer/republica. Lizenz: cc-by-sa 2.0

OFFENE WERKSTÄTTEN LADEN EIN

Workshops im gesamten Bundesgebiet

Die offenen Werkstätten in Deutschland öffnen ihre Türen vom 8. bis 10. Mai noch ein wenig mehr: An rund 25 Orten gibt es an diesen drei Tagen (Freitag bis Sonntag) Workshops, Vorträge sowie Repair-Cafés, bei denen man defektes Hab und Gut zur Reparatur mitbringen kann.

Vor allem handwerkliche Fähigkeiten kann man an den Aktionstagen erlernen: Die Werkstätten bieten vom Nähen und Siebdrucken über das Töpfern bis zum Schmieden und Schreinern zahlreiche Workshops an, an einigen Orten wie FabLabs kommen noch Elektronik und 3D-Druck hinzu. Der Verbund Offener Werkstätten hat online eine vollständige Liste der an den Aktionstagen teilnehmenden Orte und was man dort erleben kann.

—phs

► www.offene-werkstaetten.org/seite/aktionstage-2015



FABLAB NEUMARKT STARTET

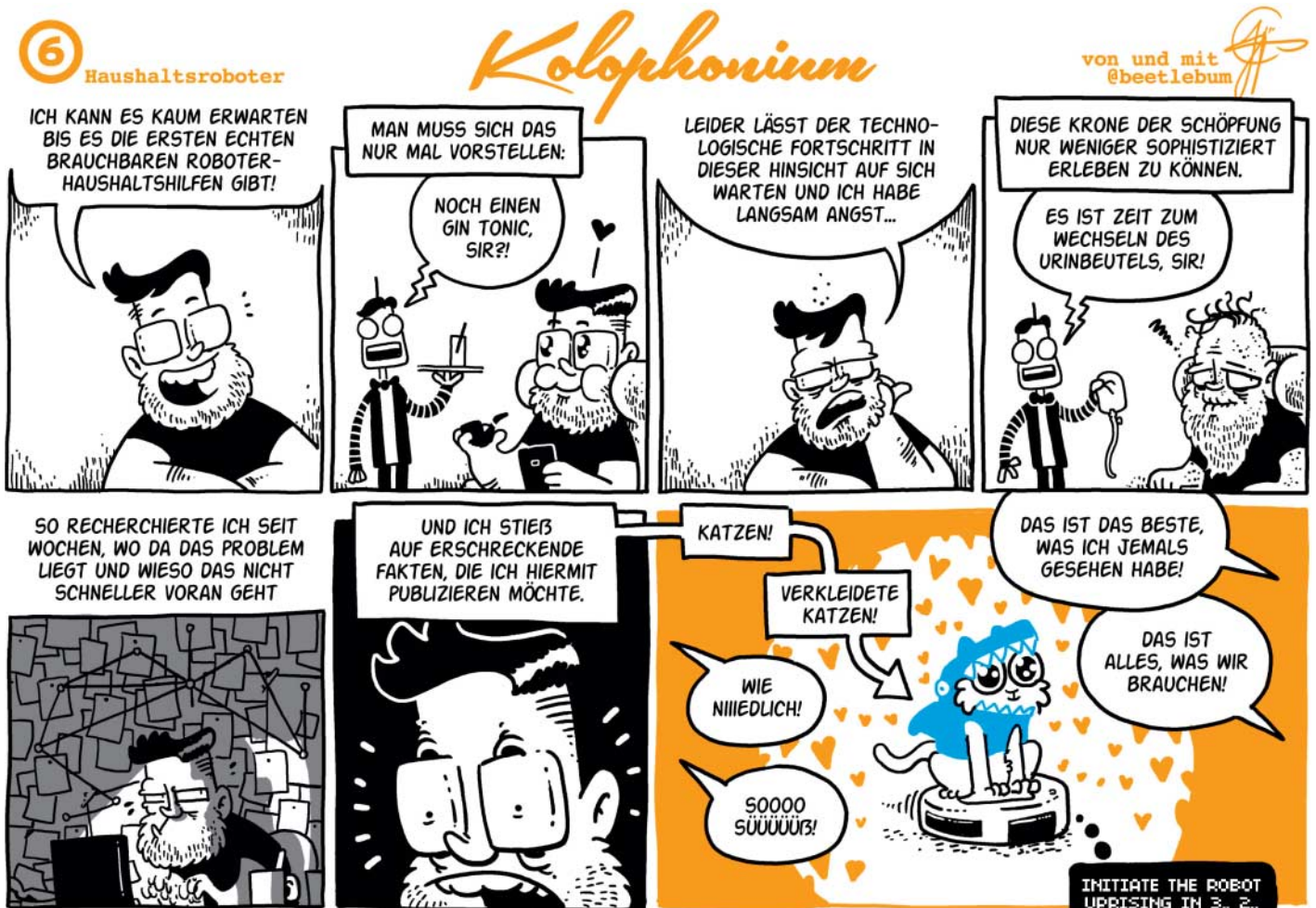
In der Oberpfalz soll ein neues FabLab entstehen. Maschinen gibt es schon, derzeit suchen die Initiatoren nach Räumen und wollen einen Verein gründen.

fablab-neumarkt.de

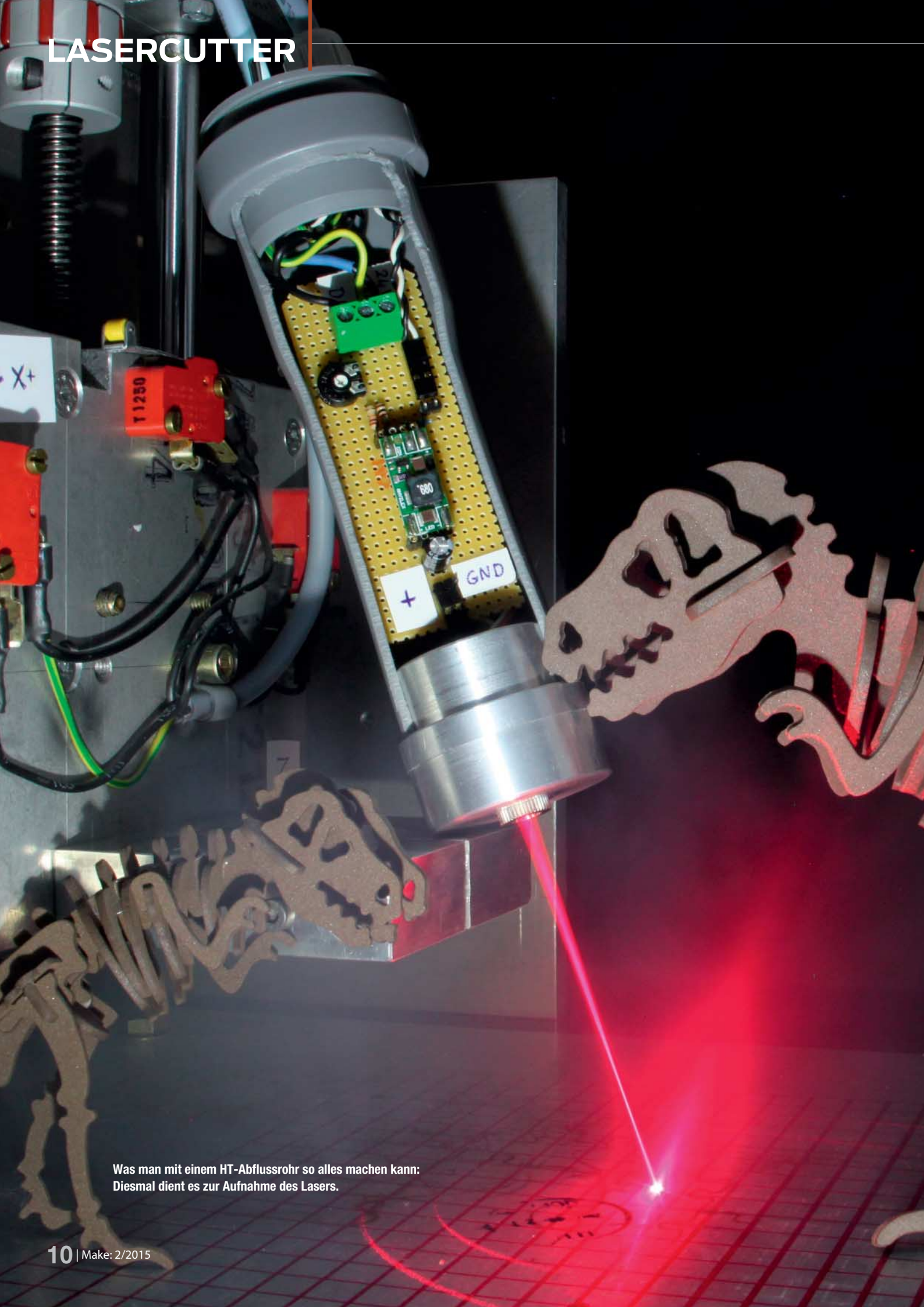
MAKERSPACE LEIPZIG ERÖFFNET

Am 11. April lud das Team zum ersten Mal in die neuen, eigenen Räume in der Bitterfelder Straße 5, wo ab sofort Workshops stattfinden werden.

makerspace-leipzig.de



LASERCUTTER



Was man mit einem HT-Abflussrohr so alles machen kann:
Diesmal dient es zur Aufnahme des Lasers.

Laser trifft Dino

In der letzten Ausgabe stellten wir einen Selbstbau-LED-Bildschirm vor. Dabei fertigten wir eine Moosgummiblende mit einem Eigenbau-Abflussrohr-Laser an. Dies erweckte das Interesse unserer Leser, die in der Redaktion nach Bauplänen, Parametern oder Unterlagen zu diesem Laser anfragten. Selbstverständlich wollen wir Ihnen Selbiges nicht vorenthalten.

von Ulrich Schmerold

Als in der c't Hacks Ausgabe 1/2013 die Eigenbau-Portalfräse vorgestellt wurde, war es für mich sofort klar, dass ich auch so ein Teil benötige. Auf die Frage meiner Frau: „Wofür brauchst du die?“, wusste ich aber leider auch keine schlüssige Antwort. Und als ich das Teil dann fertig aufgebaut mit einem lautstarken Fräsmotor im gemeinsamen Arbeitszimmer in Betrieb nahm, war es nun gänzlich aus mit dem Verständnis meiner Gattin.

Es musste also eine geräuschlose – oder zumindest geräuscharme – Alternative her.

Und wenn dann damit ein Material aus dem Bastelvorrat meiner Frau bearbeitet werden könnte, wäre es doch gelacht, wenn ich nicht im Arbeitszimmer bleiben dürfte. Was liegt da näher, als ein absolut geräuschloser Laser, der Moosgummi bearbeitet?

Diese Alternative liefert sicherlich auch anderen Lesern gute Argumente. Als Kernstück unseres Lasers verwenden wir eine rote Open-Can-Laserdiode, die bis zu 300 mW optischer Leistung liefern kann. Bei ihr lässt sich der Laserpunkt auf einen winzigen Punkt fokussieren und kann somit perfekt zum Schneiden von Moosgummi eingesetzt werden. Wir entdeckten sie im Internet bei InsaneWare (www.insaneware.de). Die Diode kann dort einzeln für 21,90 Euro oder vorinstalliert im Kollimatorgehäuse für 34,90 Euro erworben werden.

Die Lebensdauer einer Laserdiode hängt vor allem von der Stromstärke ab, mit der man sie betreibt. In DVD-Brennern werden sie abhängig von der Brenngeschwindigkeit meist mit weniger als 150 mW betrieben. Sie halten zwar auch höhere Ströme aus, altern dadurch

Kurzinfo

Zeitaufwand:
2 Stunden

Kosten:
60 Euro

Löten:
leichte Lötarbeiten

Schwierigkeitsgrad

 leicht schwer

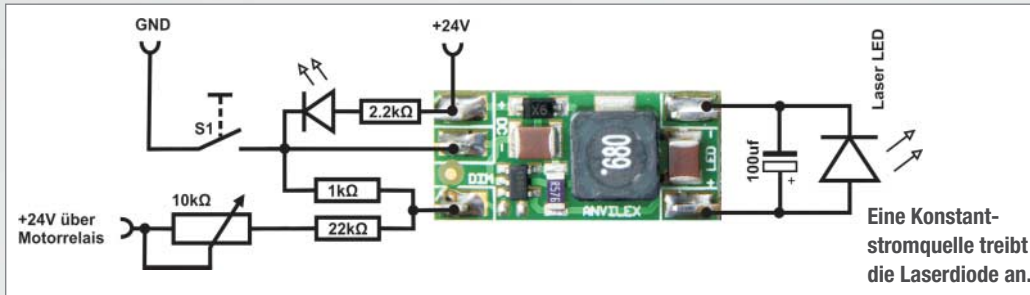
Einkaufsliste

- » Laserdiode im Kollimatorgehäuse
- » Konstantstromquelle
- » HT-40-Abflussrohr
- » Alu-Drehteil (siehe Kasten)
- » Platinenrest
- » Widerstände 1k, 2k2, 22k
- » Elko 100 µF
- » Wärmeleitpaste

ACHTUNG



Der in diesem Beitrag vorgestellte Laser ist ein Klasse-3B-Laser und sicher kein Kinderspielzeug. Er ist mehr als 300-mal so stark wie ein normaler Laserpointer. Ein direkter Blick in den Strahl verursacht unweigerlich eine ernstzunehmende und möglicherweise irreparable Verletzung des Auges. Das diffuse Streulicht ist zwar kurzzeitig nicht gefährlich, kann aber dennoch zu Reizungen des Auges führen. Denken Sie unbedingt auch an Personen, die sich im gleichen Raum aufhalten. Wir empfehlen das Tragen einer für rote Laser (658nm) geeigneten Laserschutzbrille.



aber wesentlich schneller. Um Moosgummi sauber zu schneiden, müssen wir ihr jedoch mindestens 250 mW abverlangen.

Die dabei entstehende Wärme muss bei dieser Leistung aber unbedingt abgeführt werden. Die Kühlung übernimmt in unserem Modell das 43-mm-Alu-Drehteil, welches genau in die Aufnahme unserer CNC-Fräsmaschine passt. Die Aluteile der CNC-Maschine führen dann die Wärme weiter ab. Zur besseren Wärmeableitung wird das Kollimatorgehäuse mit Wärmeleitpaste in das Alu-Drehteil eingebaut. Unser so gekühlter Laser arbeitet auch nach über 100 Stunden Betrieb immer noch wie am ersten Tag.

Treiberschaltung

Als Treiberstufe verwenden wir die Konstantstromquelle (KSQ) Anvilex CCS2-350-HV48, die wir für 5,50 Euro bei Pollin (www.pollin.de) erworben haben. Sie ist für eine Betriebsspannung von 6–50 V ausgelegt und kann somit direkt über die Spannungsversorgung unserer CNC-Maschine betrieben werden. Da es sich dabei um ein sogenanntes StepDown-Modul, also um einen Schaltregler und nicht um einen Linearregler handelt, wird das Modul kaum warm und bedarf keiner Kühlung. Die Stromstärke ist bei diesem Modul auf 350 mA eingestellt. Prinzipiell lässt sich dieses Modul zwar über den DIM-Anschluss per PWM steuern, da die Leistung der Laserdiode zum Bearbeiten aber gerade mal so ausreicht, haben wir uns für einen festen Wert entschieden. Die Schnitttiefe steuern wir über die Vorschubgeschwindigkeit der CNC-Maschine.

Wird ein StepDown-Modul mit der Stromversorgung verbunden, fließt kurzzeitig ein

sehr hoher Strom. Deshalb ist es nicht empfehlenswert, den Laser durch direktes Ein- und Ausschalten der Modul-Stromversorgung zu steuern. Dabei würde das Relais der CNC-Maschine unnötig schnell verschleiben. Aus diesem Grund steuern wir den Laser über den DIM-Anschluss.

Ist der DIM-Anschluss nicht belegt, so fließen 350 mA über die Laserdiode; ziehen wir ihn mit einem 1-kOhm-Widerstand gegen Masse, wird die Laserdiode ausgeschaltet. Zum Einschalten legen wir über den 22-kOhm-Widerstand und das 10-kOhm-Poti die 24 V Betriebsspannung an. Die drei Widerstände arbeiten dann als Spannungsteiler und legen (je nach Potistellung) 0,7–1,1 V an den DIM-Anschluss. So lässt sich mit dem Poti die Stromstärke am Ausgang zwischen 250 und circa 330 mA regeln.

Ein Elko am LED-Ausgang zum Glätten von Spannungsspitzen, eine LED nebst 2,2k Vorwiderstand zum Anzeigen der Betriebsbereitschaft und ein Hauptschalter in der GND-Leitung komplettieren die Schaltung.

Als Gehäuse für die Treiberelektronik dient ein 40-mm-HT-Abflussrohr. Im Deckel sind der Hauptschalter und die LED untergebracht. Die Gewindeschraube (rechts unten im Bild) fixiert das Kollimatorgehäuse. Als Testaufgabe ließen wir unseren Laser ein Modell des Tyrannosaurus Rex aus einer 2 mm dicken Moosgummiplatte schneiden. Die dafür benötigten Dateien und eine NC-Datei zum leichteren Einrichten des Lasers stellen wir Ihnen wie immer zum Download bereit (siehe Link).



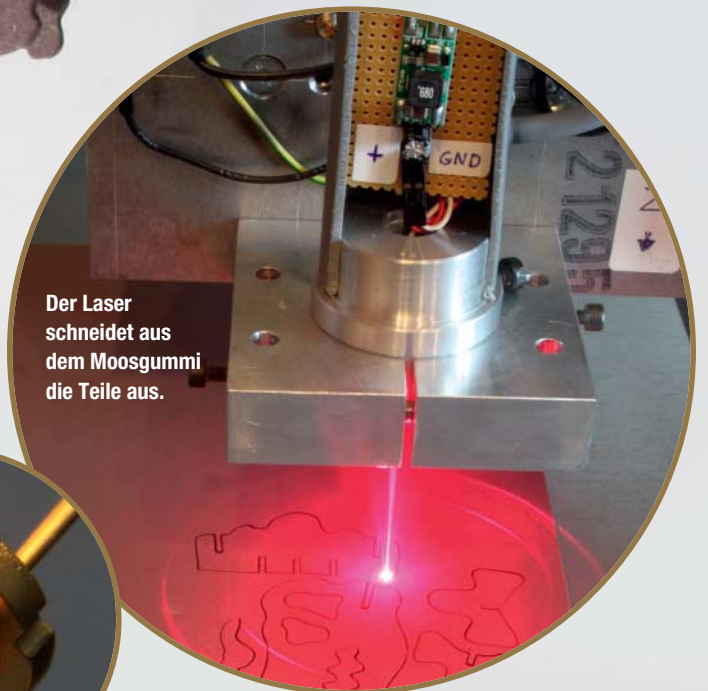
Links und Foren
make-magazin.de/xb4t



GEEIGNETES MATERIAL

Bei der Schneidleistung eines Lasers kommt es nicht zuletzt auf die Farbe des Materials an. In Tests ließen sich mit unserem roten Laser brauner, grauer, grüner, schwarzer und blauer Moosgummi sehr gut bearbeiten – weißer und

gelber wiederum so gut wie gar nicht. Es kommt darauf an, wie gut die Farbe das rote Laserlicht absorbiert beziehungsweise reflektiert. Bei perfekter Fokussierung lässt sich mit unserem Laser auch Holz gravieren.



Der Laser schneidet aus dem Moosgummi die Teile aus.



Open-Can-Laserdioden haben kein Gehäuse, welches den empfindlichen Laserchip schützt. Dadurch erzielen sie aber auch eine höhere optische Leistung.

ACHTUNG, HEISSER TIPP

Verbinden Sie niemals eine Laserdiode mit einem StepDown-Modul, das bereits unter Spannung steht. Fließt kein Strom am Ausgang des Moduls, so liegt am Anschluss für die LED fast die gesamte Betriebsspannung an. Schließt man nun die Laser-LED an, so wird diese augenblicklich zerstört.

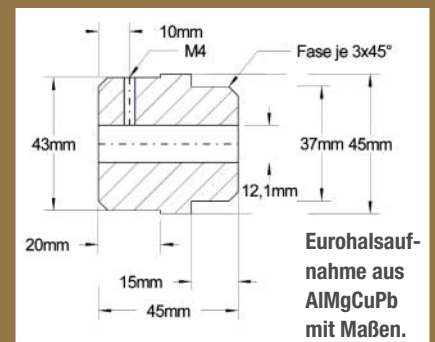
Wir wollten einmal messen, wie viel Strom über die LED fließt. Dazu schalteten wir ein Amperemeter zwischen LED und Treiber. Beim Umschalten des Amperemeters auf einen anderen Bereich wurde die LED kurz von der Treiberstufe getrennt und wieder verbunden. Die

Konstantstromquelle hob die Ausgangsspannung kurz an. Dies bedeutete den Tod unserer ersten Laserdiode. Zum Messen des eingestellten Stroms schließen Sie einfach den Ausgang über ein Amperemeter kurz.

ALU-DREHTEIL

Nicht von der Stange ist das Alu-Drehteil, genauer gesagt die Eurohalsaufnahme, die genau in die Aufnahme der CNC-Fräse passt. Wer sich das Teil selber drehen möchte, findet die Maße hier im Bild.

Wer keine Drehbank besitzt, kann es auch für 9,80 Euro bei Insaneware kaufen. Die Ulrichswerkstätten Schwabmünchen (Werkstatt für behinderte Menschen, WfbM) fertigen die Teile und geben sie weiter. Der Erlös finanziert die Gehälter der Beschäftigten. —dab



Boom Boom Tschak

Von verzerrten Kickdrums bis zu verspielten Chiptunesounds: Mit einer Handvoll passiver Bauelemente lassen sich schnell eigene Klänge kreieren. Kombiniert man mehrere Sounds, ergibt das eine analoge Beatmaschine.

von Florian Fusco

Kurzinfo



Zeitaufwand:
ein Wochenende



Kosten:
5 bis 30 Euro



Löten:
Grundkenntnisse

Schwierigkeitsgrad

leicht schwer

Einkaufsliste

- » diverse Widerstände
- » Dioden: 1N914
- » Elkos: 1 μ F, 10 μ F
- » diverse Kondensatoren
- » Transistoren: BC549B, BC559B
- » Taster oder Game-Controller
- » Stromversorgung 9 Volt (Netzteil oder Batterie)
- » Klinkenbuchse 6,3 mm
- » Kabel (Draht, Ethernetkabel)
- » optional: Gehäuse



Über den Monoausgang lässt sich die Rhythmuskiste problemlos mit einem Bass- oder Gitarrenverstärker betreiben. Im durchsichtigen Case weckt das vermeintliche Chaos der Bauelemente die Neugierde der Partygäste. Um so besser, denn der zweite Controller fordert zum Mitmachen auf.

Dieses elektronische Schlagzeug hat nicht nur einen individuellen Sound: Man kann es außerdem spielen, ohne hinzuschauen.



Wie leicht es doch Gitarristen haben: Entwickelt sich die Party zum Wohnzimmerkonzert oder zum Küchenjam, sind sie bestens ausgestattet. Eine akustische Gitarre lässt sich mühelos transportieren und respektiert eine gewisse Dezibelgrenze.

Schlagzeuger stehen hier vor einem echten Problem, denn eine akustische Trommel ist oft zu laut. Geht man den elektronischen Weg, kann man die Lautstärke regeln, handlich ist so ein Setup aber meistens nicht: Ein Drumpad-Controller mit Laptop, Sound-Interface und zugehörigen Kabeln – das artet schon wieder in Arbeit aus. Etwas praktischer zum Finger-Trommeln sind da noch Stand-alone-Geräte wie die Akai MPC, dem klassischen Hip-Hop-Drumcomputer.

Der Nachteil bei allen Klangerzeugern mit Drumpads ist allerdings, dass man beim Spielen ständig auf das Interface schauen muss. Wer Gitarre spielt, kann dagegen Blickkontakt zu anderen Musikern oder zum Publikum aufnehmen.

Unsere digitale Retro-Beatmaschine mit ihrem partyfreundlichem Interface behebt dieses Manko. Die Drum-Sounds kann man sich durch die Auswahl der Bauteile selbst zusammenstellen.

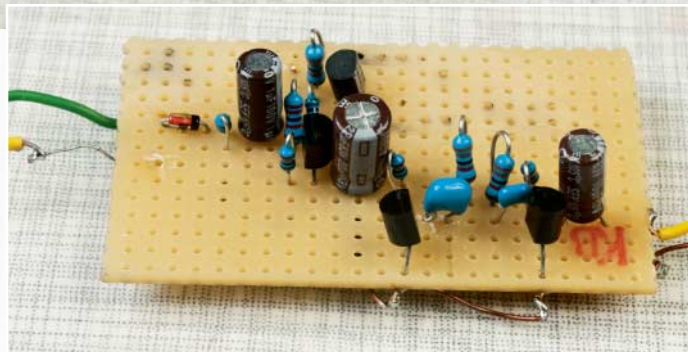
Sounds diskret zusammenbauen

Für unser Vorhaben ist die Schaltung „Mini Space Rockers“ des Synthesizer-Bastlers Eric Archer bestens geeignet. Der anfängerfreundliche Schaltkreis kommt mit wenigen diskreten Bauelementen aus und bietet eine breite Palette an Sounds. Für jeden einzelnen Drumsound von Boing bis Blip baut man dieselbe Grundschaltung auf und beeinflusst dann über die Wahl dreier Kondensatorwerte die Art und Dauer des Klangs.

Für die ersten Klangversuche, die man am besten auf einem Steckboard aufbaut, kann man die Soundvariationen mit den Kondensatorwerten aus der unten abgebildeten Tabelle ausprobieren. In den Links zum Artikel finden sich noch mehr brauchbare Kombinationen. Diese sollen jedoch nur als Ansatz bei der Klangfindung dienen: Richtig Spaß macht es zum Beispiel, die maximale Tonlänge mit verschiedenen Werten für C3 auszutesten.

Sound-Vorschläge

Sound	C1	C2	C3
Tekno Boom	1 μ F	220 nF	100 μ F
Pac Man	200 nF	100 nF	1 μ F
Lazer Tag	10 μ F	22 nF	22 μ F
Space Hawk	470 nF	220 nF	1 μ F
Mid Tom	100 nF	470 nF	4,7 μ F

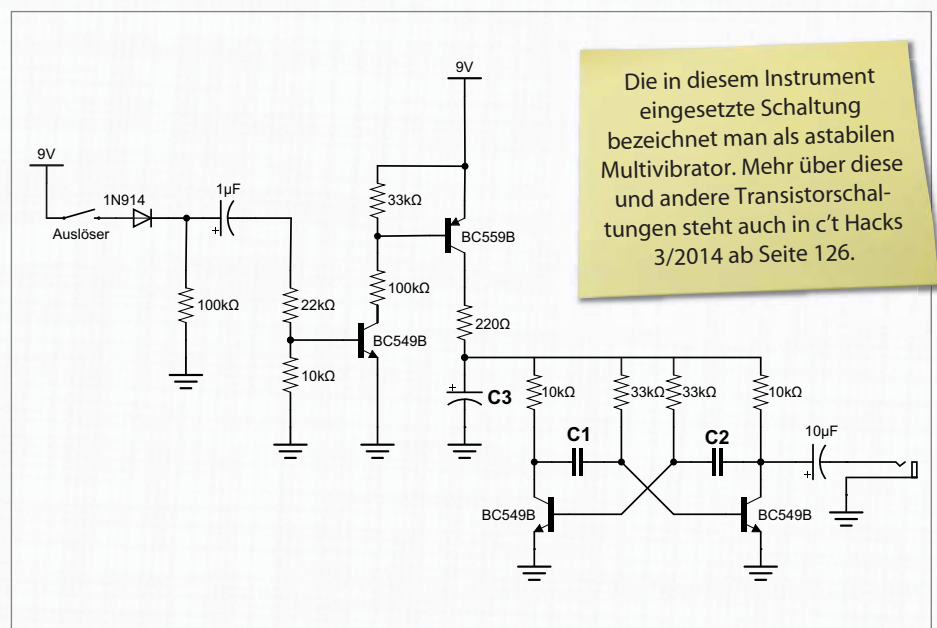


So sieht die Schaltung für einen Sound aus: Mini Space Rockers in der Lochraster- und der Freestyle-Version.

Archer empfiehlt hier zwar einen Kondensatorwert zwischen 1 und 10 μ F, doch die Merkwürdigkeit mancher Sounds offenbart sich erst bei extra langen „Sweeps“ ab 100 μ F. Es lohnt sich also auf jeden Fall, etwas Zeit zum Experimentieren einzuplanen.

Fest und luftig verdrahtet

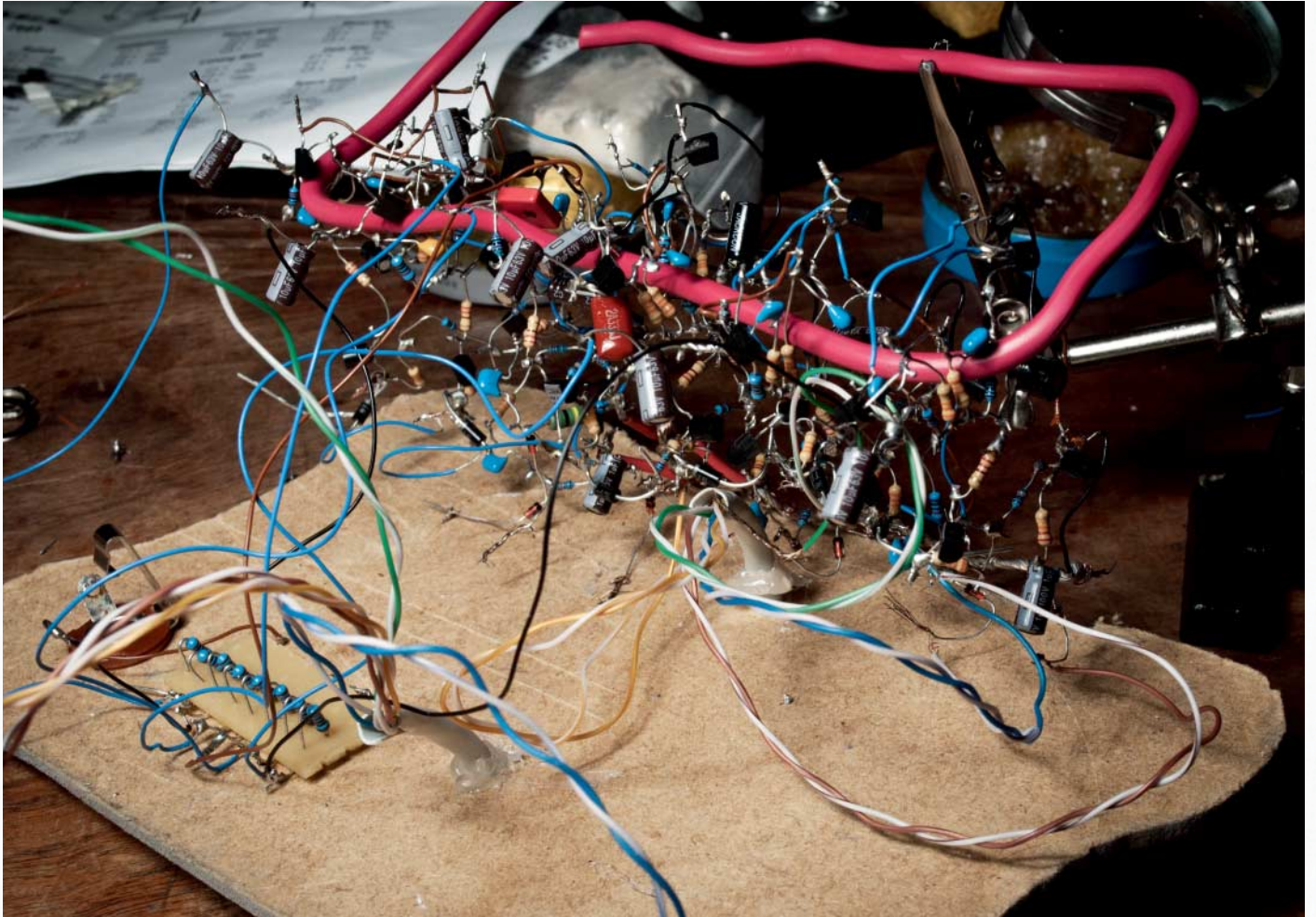
Ist die Klangfindungsphase beendet, sollte man die gewünschten Sounds fest verdrahtet aufbauen und sich auf diese Weise ein eigenes elektronisches Drumkit zusammen-



Die in diesem Instrument eingesetzte Schaltung bezeichnet man als astabilen Multivibrator. Mehr über diese und andere Transistorschaltungen steht auch in c't Hacks 3/2014 ab Seite 126.

Der Schaltplan zu Eric Archers „Mini Space Rockers“.

C1 und C2 (0,0022 μ F bis 0,22 μ F) definieren den Klang, C3 die Tonlänge (1 μ F bis 10 μ F).



Ja, das funktioniert. Wer wie hier acht Exemplare des Sound-Schaltkreises aufbaut, sollte allerdings nicht den Überblick verlieren.

stellen. Dazu kann man sich eigene Platinen herstellen (lassen) oder die kleinen Schaltungen auf Lochrasterplatine aufbauen.

Eine dritte Variante ist, die Klangmaschine einfach frei fliegend aufzubauen: Pro Sound sind es zwanzig Bauteile, was beim Basteln noch gerade so überschaubar ist.

Umso überraschter wird das Publikum später sein, wenn aus dem Drahtverhau Klänge entstehen.

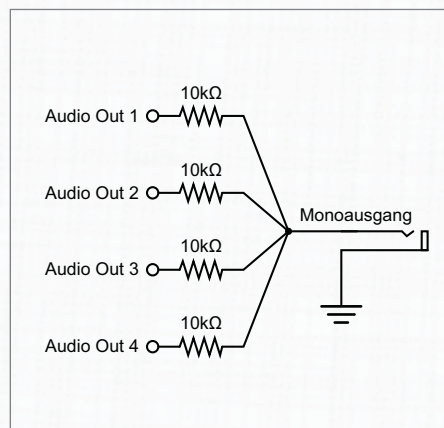
Wenn man sich für diesen Weg entscheidet, sollte man zwei dickere Drähte für die Stromversorgung nehmen und dazwischen jede Einzelschaltung verlöten. Damit bringt

man etwas Ordnung in den Aufbau und kann die Klangerzeuger einfach mit einer Stromquelle speisen.

Im Mono-Supersound

Um die Sounds in einem einzigen Monoausgang zusammenzufassen, benötigt man einen einfachen Passivmischer. Der ist schnell aus ein paar Widerständen hergestellt. Dabei sollte man sich nicht vom einfachen Aufbau dieses Soundmischers täuschen lassen: In der professionellen Musikproduktion werden Passivmischer eingesetzt, wenn der ursprüngliche Klang einer Soundquelle beim Mischen nicht verfälscht werden soll.

Bei diesem Arbeitsschritt ist ein bisschen Prototyping nötig, denn einige Space-Rocker-Sounds fallen lauter aus als andere. Um einen Drumsound in der Mischung leiser zu machen, tauscht man den entsprechenden Widerstand gegen einen höheren Wert aus. Alternativ kann man hier auch einstellbare Widerstände, also Potentiometer verwenden.



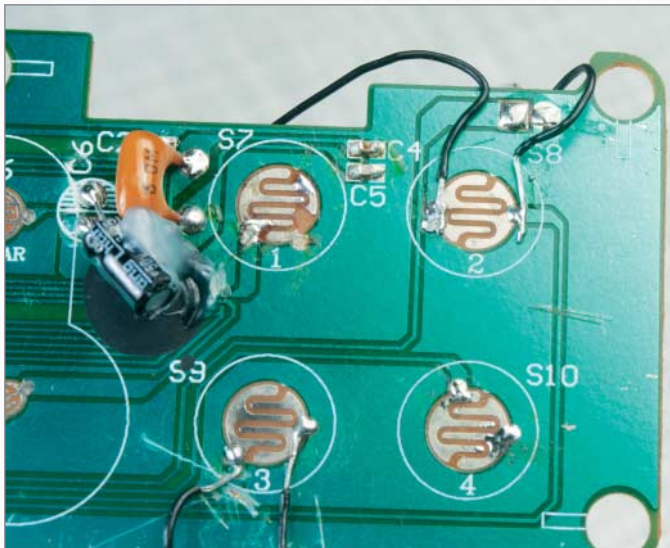
Passiv und effektiv: Ein Passivmischer macht den Beatmaker zur Monomaschine.

Sounds spielen



1 Wie spielt man diese Sounds?

Zum Auslösen muss man nur einen Taster drücken, ist also sehr flexibel, was das Schlagzeug-Interface angeht. Hier ist eine Idee: Wie wäre es mit einem ausgedienten Gamepad? Es liegt gut in der Hand und wer gerne spielt, ist daran gewöhnt, die Knöpfe zu bedienen, ohne hinzusehen. Außerdem erinnern die analogen Drumsounds an die Klänge der 80er Jahre.



2 Neue Verbindungen schaffen

Um einen Controllerbutton in einen mechanischen Taster umzuwandeln, trennt man mit einem Cutter die Verbindungen zum Rest der Platine auf. Dann lötet man an beide Seiten des Kontakts je ein Kabel. Die Lötstellen sollten flach sein, damit man das Gehäuse wieder schließen kann.



3 Ein Schlagzeug für zwei

Da wir schon dabei sind, den Beatmaker auf Spielekonsole zu stylen, warum nicht gleich einen 2-Player-Modus einbauen? Im Beispiel wurden jeweils vier Sounds auf zwei Gamepads verteilt. Als Zuleitung für die Auslöser eignet sich herkömmliches Ethernetkabel. —*phs*



Links und Foren
make-magazin.de/xmfz

Anzeige

Wir sind dann mal auf dem Mond

Jede Stufe ihrer Hardware-Entwicklung beginnt mit der Erkenntnis, dass es unmöglich sei, es zu schaffen, sagt Projektinitiator Robert Böhme über die Part Time Scientists. Sie wollen für 20 Millionen Euro zum Mond fliegen, dort herumfahren und Bilder zur Erde übertragen. Noch Fragen? Ja. Viele!

von Anika Kehrer



KALTSTART

Arne Reiners überlegt konzentriert, wie alt er eigentlich ist. Der Programmierer aus Bremen ist Gründungsmitglied der Part Time Scientists und der generelle Plan B der Gruppe – weil er für alles eine Lösung weiß. Und das, obwohl er weder eine Ausbildung noch ein Studium absolviert und – wie Projektkollegen erzählen – noch nie einen Lebenslauf geschrieben hat. „Der wird einfach irgendwie angesprochen und eingestellt, weil jeder, der ihn kennt, weiß, was er kann“. Neben ihm sitzt Karsten Becker, Doktorand der technischen Informatik im Bereich FPGA an der Uni Hamburg-Harburg. Woran genau die beiden bei den Part Time Scientists arbeiten, an dieser Frage scheiterte ich beharrlich an diesem Tisch irgendwo auf dem 31c3. Irgendwann fällt das auf. Karsten Becker springt dann kichernd in die Bresche: „Arne zu fragen, was er ganz konkret macht, ist eigentlich die falsche Frage, weil er einer der wenigen ist, die wirklich alles können.“

Reiners hat zum Beispiel zu Beginn des Projekts mal die Bahnberechnung abgeschätzt, also wie viel Energie man für wie viel Gewicht bei welchen einwirkenden Kräften benötigt, um auf den Mond zu kommen. „Das ist eigentlich gar nicht so schwer“, meint er. „Alles, was man wissen muss, steht in der Wikipedia. Man muss nur wissen, was man davon braucht.“ Später hat sich herausgestellt, dass seine Schätzung

nur um ein paar Prozentpunkte von dem Ergebnis abweicht, das ein Profi errechnet.

Die beiden erklären ihren Mond-Rover so: Es gibt vier Räder, jedes mit vier Freiheitsgraden: vorwärts/rückwärts, links/rechts, hoch/runter und rund um die eigene Achse. Dafür addiert man Rotationsvektoren auf Richtungsvektoren und Tangenten der Radmitte. Natürlich müssen die Bewegungen der einzeln ansteuerbaren Räder auch von der Geschwindigkeit her zueinander passen. Aber Becker findet: „Das ist nicht so schwierig. Mit dem Algorithmus, den ich mir ausgedacht habe, ist das trivial.“ Ein im Bauch des Rovers sitzender FPGA nimmt dann die Richtungsanweisungen entgegen und gibt sie an die Motorentreiber weiter. Vier Treiber sind für die Vorwärts- und Lenkbewegungen je zweier Räder zuständig. Zu dem Mond-Rover gehört auch noch ein Lander, der auf der Spitze einer Rakete sitzen wird, alles unter den Armen trägt und – idealiter – sanft absetzt. Dieser Lander verfügt über den Uplink zur Erde (via Laser), über den die Fernsteuersignale von der Erde kommen, denen der Rover auf dem Mond gehorcht, und über den die Bilddaten zurück an die Erde gehen.

So sieht ein Kaltstart zum Verständnis der Part Time Scientists aus. Arne Reiners und Karsten Becker sind zwei der weltweit rund 70 Projektmitglieder und Teil des sechs- bis zwölfköpfigen Kernteams.



Bild: Anika Kehrer, rffinem.de

Karsten Becker promoviert in technischer Informatik. Seine Arbeit am Mond-Rover – dessen Bauch er hier gerade zuklappt, weil die Kabel so unschön herausgingen – begeistert seinen Professor, weswegen er ihn in seine bezahlte Arbeit an der Uni integrieren kann.



Bild: Anika Kehrer, rffinem.de

Arne Reiners ist Software-Entwickler. Aber eigentlich, sagen seine Kollegen, ist er eine Art Geheimwaffe, denn er könne alles. Im Moment beschäftigt er sich vor allem mit der Bildverarbeitung des Kamerakopfes.

DIE PART TIME SCIENTISTS IN KÜRZE

Die Part Time Scientists sind private Freizeit-Raumfahrer. Oder genauer: Sie setzen alles daran, Raumfahrer zu werden. Gefunden hat sich die Gruppe für die Teilnahme an einem Wettbewerb zur Förderung der privaten Raumfahrt, dem Google Lunar XPrice. Google will hiermit die Entwicklung von kostengünstigen Raumfahrt-Projekten voranbringen. Insgesamt 30 Millionen US-Dollar sind an Preisgeldern ausgeschrieben. Die Finanzierung der teilnehmenden Projekte darf nur zu höchstens 10 Prozent aus staatlichen Mitteln bestehen. Die Aufgabe ist es, eine Raumfahrt-Mission zu entwickeln, die ein unbemanntes Fahrzeug auf dem Mond absetzt, dort 500 Meter fährt und davon Bilder auf die Erde übermittelt.

Die Mission der Part Time Scientists besteht aus drei Teilen. Teil eins ist der Transport in die niedrige Erdumlaufbahn (Low Earth Orbit, LEO). Dafür will die Gruppe den Service eines Raketenstart-Dienstleisters in Anspruch nehmen. Ein möglicher Start wäre mit dem Dnepr LV (Launch Vehicle) vom Yasny Kosmodrom in Russland. Dann beginnen Teil zwei und drei, die die Gruppe im Eigenbau vornimmt: In Teil zwei übernimmt ein Landemodul die Navigation um den Mond und die Landung. Teil drei ist die erfolgreiche Fahrt und Bildübermittlung des Rovers vom Mond

Eine Dnepr beim Start – der Launch-Teil ihrer Mission kostet die Part Time Scientists etwa 20 Millionen US-Dollar.



Bild: ISC Kosmotras

MOIN, WER WILL MIT ZUM MOND?

Im Jahr 2007 sind die Part Time Scientists zusammen mit 32 weiteren Teams für die Teilnahme am Google Lunar XPrice angetreten. Zurzeit sind noch 12 Teams im Rennen – sie sind das einzige verbleibende aus Deutschland. Und noch etwas hat sich verändert: Bei den Part Time Scientists macht nicht mehr nur das Ziel, sondern auch das Erreichte staunen.

Da ist zum einen die finanzielle Dimension. Allein der aktuelle Rover besteht aus Material im Wert von rund 250 000 Euro. Dann ist da die Kragenweite der Kooperationspartner, die, erzählt das Team nicht ohne Stolz, alle von sich aus auf das Projekt zugekommen sind: Universitäten wie die TU Wien oder die TU Hamburg-Harburg, die relevante Ressourcen beisteuern, erscheinen ja noch naheliegend. Aber DLR, ESA und NASA klingen doch schon etwas anders. Solche Organisationen sind ja im Privatsektor bisher gar nicht anzutreffen.

Und schließlich ist da noch die zeitliche Dimension. Das Projekt dauert seine bisherigen sechs Jahre nicht deswegen, weil es herumschlufft, sondern weil es enorm aufwendig ist und in großen Teilen nebenberuflich entsteht (alle anderen Teams des Wettbewerbs arbeiten hauptberuflich und sind – nach Aussage der

Part Time Scientists – auch nicht weiter). Es besteht aus offenbar ziemlich fähigen, erstaunlich unbeirrten, nach eigener Aussage ein bisschen verrückten und „einfach netten“ Mitstreitern. Auf der Bühne des Chaos Communication Congress (CCC), wo sie seit je her über ihren Fortschritt berichten, wirken sie wie große Jungs, die begeistert Unsinn reden.

Karsten Becker und der IT-Freelancer Robert Böhme, welcher das Projekt überhaupt erst angezettelt hat, stehen hier auch Ende 2014 auf der Bühne. Sie haben nämlich eine Mitfahrgelegenheit zum Mond für Bastlerprojekte anzubieten.

Es ist merkwürdig zu erleben, wie sich auch bei einem so unerhörten Projekt die Dimensionen normalisieren, wenn man erst mal mit den Leuten spricht. Nach dem Vortrag erklärt Böhme, dass es in der Raumfahrt schon immer viel unnütze Masse gegeben habe, die man zum Beispiel allein dafür brauche, etwas auszubalancieren. Als Füllmaterial sei in der Vergangenheit zum Beispiel Blei zum Einsatz gekommen. „Das tut weh“, sagt er, „wenn man bedenkt, dass ein Kilogramm Richtung Mond ungefähr 1,2 Millionen Dollar kostet.“ Noch etwas überrumpelt fragt man, wieso sie etwas so Teures kostenlos

anbieten. Robert Böhme antwortet in seiner sehr natürlichen, schnell denkenden und schnell redenden Art: „Auch wenn es blöd klingt – warum denn nicht? Was mir wichtig ist: Ich mache das nicht aus Profitgründen. Sondern ich denke, dass wir hier die Möglichkeit haben, die Raumfahrt und



Bild: Anika Kehrer, rtfinem.de

Robert Böhme denkt und redet schnell. Wenn er sich in einer Sache sicher ist, scheut er sich nicht, sie umzusetzen. Der Raumfahrt-Fan wurde im Jahr 2008 mit 23 Jahren Initiator und Namensgeber der Part Time Scientists.

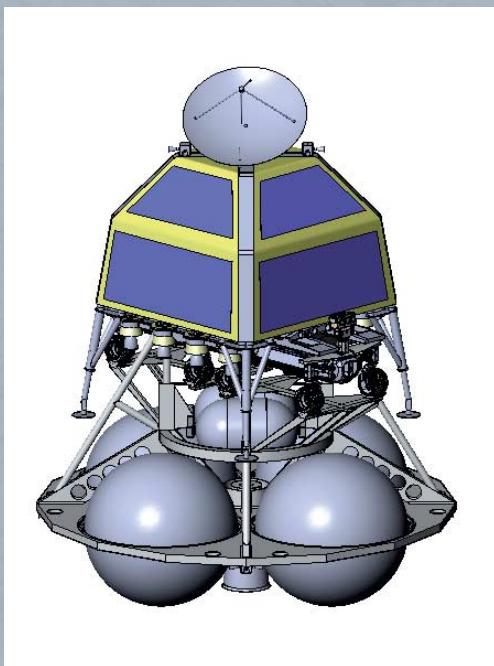


Bild: Alex Adler, Part Time Scientists

Dieser Entwurf des Landemoduls entstand 2013. Es soll zwei Rover auf den Mond bringen. Seit 2014 schraubt das studentische Space Team der TU Wien als Partner der Part Time Scientists daran, es in die Tat umzusetzen.

die Menschen einen großen Schritt weiterzubringen. Ich sehe dieses Potenzial.“

Das Projekt verzichtet allerdings nicht zur Gänze auf die Einnahmemöglichkeiten, die sich durch den Shuttle-Service bietet. Von den zwölf Cube-Sats, die nach aktuellem Planungsstand mit auf den Mond fliegen sollen, sind nach Angaben des Projekts zwei für die Xprice-Foundation reserviert und zwei weitere für Sponsoren. Den Rest verkaufen sie. Was einer kostet, wollen sie nicht verraten, das sei „verhandelbar“. Auf den CCC-Vortrag hin hat sich übrigens die NASA bei ihnen gemeldet. Sie würde gern einen Samen mitsenden, um ihn beim Wachsen zu beobachten. Das haben sie bisher zwar schon auf der ISS machen können, aber noch nie auf dem Mond.

MITFAHRGELEGENHEIT ZUM MOND

Folgende Voraussetzungen müssen die Projekte erfüllen:

- Die Artefakte müssen Strahlung und starke Vibration aushalten und im Vakuum arbeiten können.
- Das Making muss unter 1,33 Kilogramm bleiben.
- Der Experimentaufbau rechtzeitig zum avisierten Launch im Jahr 2016 fertig sein.

Die Bewerbung ist bis Ende April 2015 unter ptscientists.com/go/space möglich.

Anzeige

VOM IRC-CHAT ZUM 4000-EURO-DUMMY

Vor dem Google Lunar Xprice, kurz GLXP, gab es die Idee nicht, stellt Robert Böhme klar („Wer kommt schon darauf, was zum Mond zu schicken?“). Doch der Name Part Time Scientist ist älter als das Projekt: Er wurzelt darin, dass Böhme früher über eigene Physik-Experimente bloggte, woraufhin ihn Kollegen Part Time Scientist taufte. Mit einem bestimmten Kern von Leuten – viele davon sind heutige Part Time Scientists – hing er über die Jahre im IRC (Internet Relay Chat) zusammen. Gegen Ende 2008 erfuhr er vom GLXP. Er fragte sich, erzählt er heute, ob es überhaupt Sinn ergebe, sich damit zu beschäftigen. Kann es möglich sein, privat zum Mond zu fliegen? Diese Frage stellte er den anderen – und veranstaltete damit unbemerkt ein Kickoff.

Wenn das Team von seinen Anfängen berichtet, überschlagen sich die Erinnerungen („Wir haben, ganz ehrlich, überhaupt nicht überblickt, was da auf uns zukommt. Nicht mal ansatzweise“). Arne Reiners erinnert sich: „Zu Anfang hat man nicht gemerkt, dass da ein Projekt draus wird. Man hat sich darüber unterhalten, was möglich ist und was nicht geht. Zu Anfang haben wir auch nicht Software ge-

schrieben, sondern ein Konzept, wie man da überhaupt hinkommt. Häufiger Gedanke war, man müsste eine eigene Rakete bauen. Dann gab es noch den Vorschlag, eine Rakete an einen Ballon zu hängen. Anfangs gab es viele Vorschläge, die unrealistisch waren. Da musste man aussortieren, und am Schluss blieb eine relativ konventionelle Mission über.“

„Damals waren wir völlig überfordert“, schüttelt Böhme lachend den Kopf. Auf dem LinuxTag 2009 haben sie die Pressemitteilung rausgeschickt: Sie als deutsches Team wollen zum Mond. Man habe sehen können, in welcher Zeitzone die Pressemitteilung gerade herumging: Erst 300 Mails aus Indien, dann 500 Mails aus Hongkong. Der nächste Schritt war der erste CCC-Vortrag Ende 2009. Böhme fasst sich an den Kopf: „Wenn ich jetzt daran denke ... Das war ein Drei-Stunden-Vortrag! Und es ist keiner weggerannt. Die Leute waren total begeistert. Wir haben gesagt: Leute, wir finden es total super, zum Mond zu fliegen. So und so würden wir das machen, und das sind die Probleme. Das Schlimmste war unsere Präsentation. Unser Englisch ...“ Robert Böhme lacht sich tot. „Man muss sich ja vorstellen, gleich den ersten Vortrag vor

tausend Leuten, und dann noch auf Englisch! Ist schon 'ne dämliche Idee. Und dann noch drei Stunden!“

Ernst wird es, als es um hehre Ziele geht. „Es geht nicht nur darum, eine Mission zum Mond zu machen“, sagt Böhme. „Es geht auch darum, dass dabei etwas entsteht, das alle weiterbringt.“ Als sie angefangen haben, erzählt er, gingen sie so handgemacht vor, wie man sich nur vorstellen kann: Sie befragten das Web. „Du gibst in Google ein: Motor für Space-Einsatz. Und du findest: nichts. Oder du findest eine Forschungsarbeit, wo mal irgendjemand darüber nachgedacht hat. Oder du findest einen Motor, den das DLR für die ISS entwickelt hat, und fragst dich, kann man das kaufen? Also fragst du beim DLR nach, und die sagen: Wat? Wir sind doch keine Firma, wir sind eine Forschungseinrichtung, natürlich kannst du bei uns nichts kaufen! Und wenn wir das verkaufen würden, dann würde das mindestens so viel kosten, wie wir an Entwicklung da reingesteckt haben, das sind ein paar Millionen.“ Und genau das sei das Dilemma, in dem man mit der Raumfahrt stecke, sagt Böhme: „Alles ist schweineteuer, du findest keine Preise und es gibt keine Anbieter.“

„Wir waren kreativ, was die EMV angeht“: Ein SPI-Verbindungskabel zu umwickeln (links), hatte nicht gereicht, um die elektromagnetischen Überstrahlungen in den Griff zu bekommen. Also versuchten sie es mit einem Ethernet-Adapter (rechts; „Das hat überraschend gut funktioniert“). Inzwischen ist diese Verbindung obsolet, indem ein FPGA direkt zu den Motorentreibern auf das Board gewandert ist.

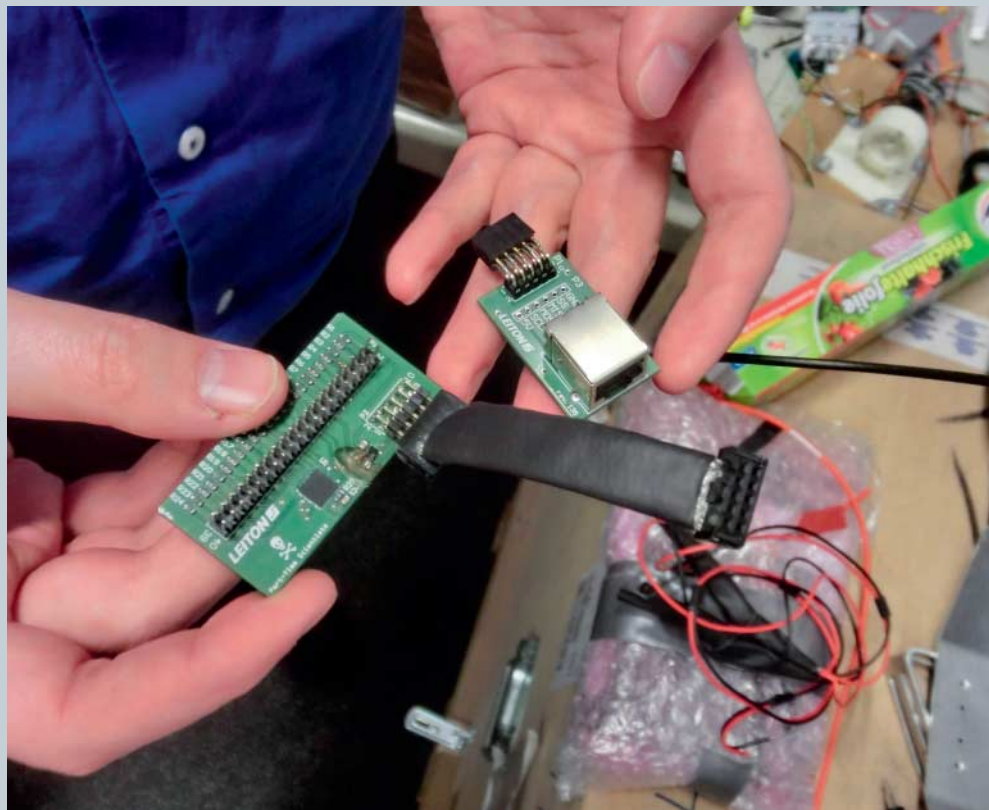


Bild: Anika Kehrner, rtfinem.de

ANS EINGEMACHTE

Karsten Beckers Büro an der TU Hamburg-Harburg sieht aus wie Kraut und Rüben. (Er sei eben einer der angewandten Informatiker, entschuldigt er sich, die anderen Büros sähen nicht so aus.) Das meiste, was hier herumliegt, geht auf das Konto der Part Time Scientists: Diverse Versionen der Platinen, des Rovers Innereien. In der Mitte des Raumes verdrängt eine taillenhohe massive Holzkiste die Luft. Darauf hockt, leidlich zerrupft, etwas aus Alu mit profilierten Rädern so groß wie Honigmelonen, das entfernt wie ein Mini-Quad aussieht, wo sich ein Kind reinsetzen und den ganzen Tag fröhlich durch brockige Erdfelder düsen könnte. Anders betrachtet steht da Hardware im Wert von 250 000 Euro und sechs Jahren Entwicklungsarbeit. Auf dem Mond wird es fahren, und zwar ferngesteuert von der Erde aus. (Ohne Kind.)

Die Hardware des Projekts besteht aus einem Mond-Rover, einer hochauflösenden Kamera und einem Landemodul. Rover und Kamera sind originäre Eigenentwicklungen des Projekts. (Die Rakete, die all das transportiert, baut das Team ja ausnahmsweise nicht selbst.) Das Landemodul ist im Frühjahr 2014 als Entwurf in die Hände des studentischen Space Teams an der TU Wien gewandert, das als Kooperationspartner zeichnet, weil bei den Part Time Scientists die Ressourcen nicht gereicht haben.

Eine Besonderheit des Rovers, erklärt Becker, sei die Fähigkeit zum seitlichen Fahren, falls die Sonneneinstrahlung quer

zur Fahrtrichtung verläuft. Seine Energie bezieht der Rover auf dem Mond nämlich aus einem Solarpanel, das er auf dem Rücken trägt. Das Panel ist auch verantwortlich für die Größe des Gefährts. Seit etwa 2012 sind sie bei der Version R3 gelandet. Sollten sie feststellen, dass das Panel wieder nicht reicht, kann das fatale Folgen haben: „Wir müssten das Panel vergrößern“, erklärt Becker, „also den Rover vergrößern. Und dann müssten wir die nächstteurere Rakete kaufen. Die kostet dann statt 20 Millionen 100 Millionen.“

Im Moment frickelt das Team an der Elektronik. Daher ist auch der Kamerakopf gerade nicht da (er ist bei Arne Reiners). Für die Videoverarbeitung evaluiert das Team gerade einen Nvidia Tegra K1, mit dem sie die Daten im H.265-Codec übertragen könnten (im Moment arbeiten sie mit JPEG 2000). Durch die höhere Kompression könnten die Kameras ihre Aufnahmen in fast voller Auflösung übertragen. Außerdem hat sich das Team dafür entschieden, den Rover-Bauch mit einer Backplane zu durchziehen, auf die die einzelnen Platinen aufgesteckt werden anstatt sie mit Kabeln zu verbinden. Auf die Art können sie einfach eine Platine auswechseln, wenn zum Beispiel ein Motortreiber kaputt geht. Auf die Frage, ob die feste Montage via Steckkarten nicht anfälliger für die Vibrationen sei, sagt Becker, dass bei Kabeln dafür andere Probleme auftreten: Da lockern sich dann die Stecker. Die Platinen hingegen würden fest verschraubt.

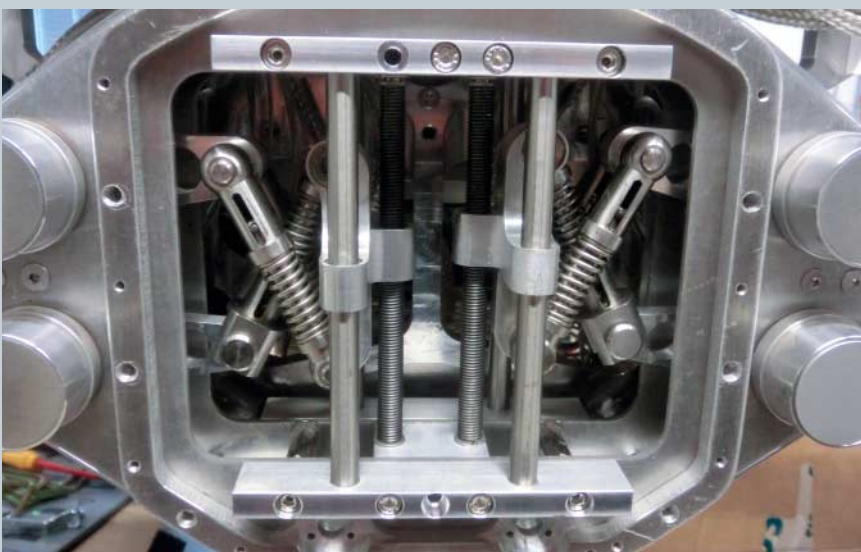


Bild: Anika Kehlner, rtfmem.de

An dieser offenen kurzen Seite des Rovers sind die Rad-Federungen zu sehen. Die beiden aufrechten Schrauben in der Mitte sind hingegen dafür da, den Rover in der Höhe verstellen zu können.

Anzeige

GEWONNEN! VORERST.

Das Setup des Rovers, wie er da bei Karsten Becker in Hamburg steht, ist genau das, mit dem ihn das Team im Dezember 2014 auf Teneriffa hat herumfahren lassen. Denn im Januar 2015 standen bei dem Google Lunar Xprice einige Meilenstein-Preise an. Insgesamt wurden sechs Millionen US-Dollar ausgeschüttet. Die Part Time Scientists hatten alle drei Komponenten eingereicht – Rover, Kamera und Lander. Mit Rover und Kamera schafften sie es Anfang 2014 in den Kreis von nur fünf Teams, die sich testen lassen durften. Bei Bestehen winkten 750 000 US-Dollar – 500 000 für die Kategorie Rover, 250 000 für die Kamera (für das Landingmodul hätte es eine Million gegeben, aber na ja).

In mehreren Tests ging es im Verlauf des Jahres 2014 also darum zu zeigen, dass das Gefährt missionstauglich ist und dass die Kamera die geforderten Bilder liefert. Für die notwendigen Prüfungen in

den Bereichen Vibration, Thermal-Vakuum, Fahren und Bildqualität nutzten die Maker unter den Augen von Juroren Labore beim DLR und beim DFKI. Und der letzte Test war der auf Teneriffa, bei dem es um die Bildqualität ging.

Im Februar 2015 war das Ergebnis da. Sie haben es tatsächlich geschafft. Sowohl die Kamera als auch der Rover hatten die Jury überzeugt: Nach über fünf Jahren Arbeit halten die Part Time Scientists mit dem Erlangen der Meilenstein-Preise den Beweis in den Händen, dass sie sich auf einem wirklich guten Weg befinden. Insgesamt zeigten bei dieser Etappe fünf Gewinnerteams, dass sie die Mission mit einiger Wahrscheinlichkeit erfolgreich durchführen können. Karsten Becker hält es aber für unwahrscheinlich, dass alle fünf tatsächlich auf dem Mond landen: Die Finanzierung ist die große Crux. Auch bei den Part Time Scientists.



Bild: Anika Kehrer, rtfinem.de

Der Übungsrover dient dazu, die Steueralgorithmen zu prüfen (und Gäste damit fahren zu lassen).



Bild: Alex Adler, Part Time Scientists

Die zwei älteren Versionen des Rovers: Links der R1 ist ein Dummy, den das Projekt 2009 als erstes vorstellte. Er ist etwas größer als eine Schuhschachtel und war noch ohne Solarpanel geplant. Rechts der R2 von 2010 hat bereits das Solarpanel, das auf dem Mond als Energiequelle dient. Version R3 ist noch größer geworden, weil das Solarpanel größer werden musste.

ENDSPURT MARKE NEUANFANG

Es ist ja schon verrückt, was das alles kostet. Auch mit 750 000 US-Dollarn kommt man da nicht weit. Doch Robert Böhme ist sich sicher: „Geld-Optionen gibt es oft. Wir müssen sie nur passend nutzen.“ Hinter den Part Time Scientists steht zu diesem Zweck eine GmbH, die schon sehr früh bei Projektstart gegründet wurde. Alle Team-



Mitglieder schließen mit dieser Firma einen Volontärsvertrag, der dem Projekt Nutzungsrechte an ihren Entwicklungen sichert, sie auf gegebenenfalls anfallende Geheimhaltungsklauseln verpflichtet und der ihnen im Gegenzug einen Teil des Preisgeldes zuspricht. Aber Böhme führt viele Sponsorengespräche, um das Projekt vom Preisgeld unabhängig zu machen. „Wir versuchen gerade, uns als Technologieunternehmen zu platzieren“, sagt er. Ihm ist das sehr wichtig. Denn: „Denn nur so ist das Ganze nachhaltig.“

Professionalisierung steht auf dem Programm. Karsten Becker sagt es so: „Wir sind vom Herzen zwar Bastler. Aber wir professionalisieren uns immer weiter. Denn nur so können wir es schaffen, Sponsoren zu größeren Geldbeträgen zu überreden.“ Und Robert Böhme greift auf seine Gründungsvision zurück. „Wir wollen anderen zeigen, dass es machbar ist. Und wir wollen mit dieser Dienstleistung Leuten helfen, das auch zu tun.“

Mit dem Meilenstein-Gewinn gehen die Part Time Scientists jetzt zwei entscheidende Schritte. Erstens: Leute wer-

den angestellt. Zu den ersten beiden gehört Karsten Becker (der sein Chaos an der TU Hamburg daher gerade in Kisten verpackt und hofft, seine Dissertation trotzdem noch irgendwie fertig schreiben zu können). Zweitens: Das Projekt erhält eine Niederlassung. Der Berliner Robert Böhme tut sich in der Hauptstadt dafür gerade händeringend nach Lab-Räumen um, und schlägt dabei die Hände über dem Kopf zusammen, weil ihm irgendwelche Start-ups ständig alles wegschnappen.

Seit der Projektgründung hat sich an der Gruppe aus Freizeit-Ingenieuren eines geändert: Sie wollen ihr bisher Erreichtes auf eine geschäftliche Grundlage stellen, um anderen zu zeigen, dass private Raumfahrt geht. Im Kontrast zur Tragweite ihres Vorhabens – sowohl technisch als auch wirtschaftlich – steht vor allem ihr ungezwungenes Auftreten. Hoffentlich werden sie es nie ablegen. —esk



Links und Foren
make-magazin.de/xg34



Die Radhalterungen sind 3D-gedruckt aus Aluminium. Sie haben eine Kabelführung integriert. Der Quickhack aus Holz soll zukünftig auch durch Alu ersetzt werden: Er hindert die Räder daran, sich komplett im Kreis zu drehen, damit sich die weiterführenden Kabel im Fall eines Firmware-Versagens nicht aufwickeln und im Inneren des Rovers vom Bord abreißen.

Bild: Anika Kehrer, rtfinem.de

DIY-Prüfstift

von Burkhard Fleischer

Für mobile Experimente wünschte ich mir Anfang der 80er Jahre einen Minicomputer, der batteriegespeist unabhängig von der Steckdose wäre. In einem Elektronikatalog fand ich einen „Einplatinencomputer“. In meiner Naivität bestellte ich ihn und war maßlos enttäuscht, als ich einen nackten Chip in der Hand hielt. Was sollte ich mit ihm anfangen?

Da stieß ich auf das Buch von Rolf-Dieter Klein „Mikrocomputersysteme“ von 1984 (Franzis-Verlag). „Selbstbau, Programmierung, Anwendung“ versprach mir der Autor im Untertitel. Sollte ich hier Hilfe bei meinem Anliegen finden? Doch das Buch war leider für Experten geschrieben, nicht für Anfänger wie mich. Trotz dieser Frustration fand ich in dem Buch einen Tipp, der für mein gesamtes Elektronik-Bastelleben bedeutsam war. Auf Seite 14 sah ich die Anleitung eines Prüfstiftes für TTL-Schaltungen, der schnell verlötet war.

Mit dem Prüfstift kann ich Pegelzustände in Schaltungen mit 5-Volt-Stromversorgung anzeigen. Dabei bietet das Gerät unterschiedliche Informationen: Je nach Pegel leuchtet entweder die LED „L“ oder „H“. Bleiben beide Anzeigen dunkel, so sind die entsprechenden Ein- oder Ausgänge unbeschaltet. Bei der Anzeige „L“ könnte auch eine Wechselspannung anliegen. Um dies zu prüfen, drückt man den Taster TA1. Liegt ein „L“-Signal vor, leuchtet weiter nur die eine LED. Bei Wechselspannung leuchten beide LEDs.

Dies kleine und unscheinbare Gerät war über die Jahre meiner Elektronikbasteleien ein treuer Begleiter. Besonders bei der Signalverfolgung und Fehlersuche hat es mir viel Zeit und Ärger erspart. Sicherlich, es gibt inzwischen Prüfstifte und -klemmen, die mehr Funktionen und Informationen bieten. Aber gerade die Einfachheit und Mobilität meines alten Prüfstiftes machen ihn zu einem alltäglichen, unkomplizierten Werkzeug. —hch

Prüfspitze

umfunktionierte dicke Stopfnadel

Kunststoffkasten

enthielt vorher Unterlegscheiben

Taster TA1

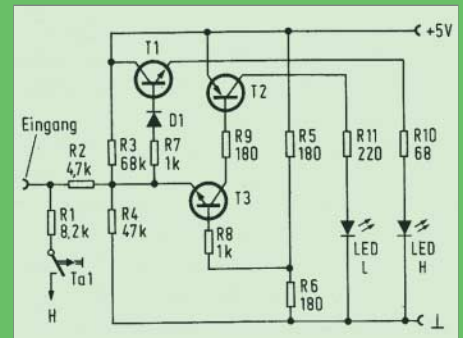
Krokodil-
klemmen

für die Strom-
versorgung

LED H

LED L

Grundlage des Prüfstifts:
die Transistorschaltung
(T1, T3: BC 107B, T2: BC 177B)



Dein Lieblingswerkzeug

Welches Werkzeug macht Dir das Leben leichter? Schick uns Deine Idee an:

mail@make-magazin.de

Anzeige

Robotik-Kits

Mit Robotern kommt im wahrsten Sinne des Wortes Bewegung in die Software-Entwicklung, was für viele Anfänger den Einstieg in die Programmiererei erleichtert. Wir haben uns sieben Systeme von günstig bis teuer und von einfach bis luxuriös angeschaut.

von Daniel Bachfeld



Das Angebot an Robotern ist schwer überschaubar, auch weil viele Hersteller ihre Produkte als Roboter bezeichnen, obwohl sie nur ferngesteuert sind und gar keine Eigenintelligenz besitzen. Abgesehen von Roboterarm haben wir uns nur mobile Modelle angeschaut, denen man selbst ein Verhalten einprogrammieren kann.

Je mehr Sensoren ein Roboter hat, auf umso mehr „Reize“ kann er reagieren. Die Klassiker sind etwa, einer Linie auf dem Boden nachzufahren, Lichtquellen zu suchen oder einfach herumzufahren und Hindernissen auszuweichen. Darum herum lassen sich dann komplexere Verhalten programmieren, beispielsweise das Durchqueren von Labyrinth (Wandfolger) oder die Kartierung von Räumen samt Tischen, Stühlen und Schränken. Für letzteres sind beispielsweise neben einem Sensor zur Messung von Entfernungen zu Hindernissen auch Rad-Encoder nützlich, mit denen man von der Rotation der Räder auf die gefahrene Strecke schließen kann (Odometrie). Mit einem Kompass oder einem Winkelratensensor ließe sich die Ausrichtung in Räumen ermitteln.

Mit Kameras und Objekterkennungssoftware kann ein Roboter nicht nur feststellen, dass etwas vor ihm steht, sondern auch, was vor ihm steht. Mit Lautsprecher, Mikrofon oder Display sind dann sogar die Grundlagen für eine Interaktion mit der Umwelt möglich. Doch ehe man so weit ist, muss man sich erst einmal in die kleinen Aufgaben einarbeiten.

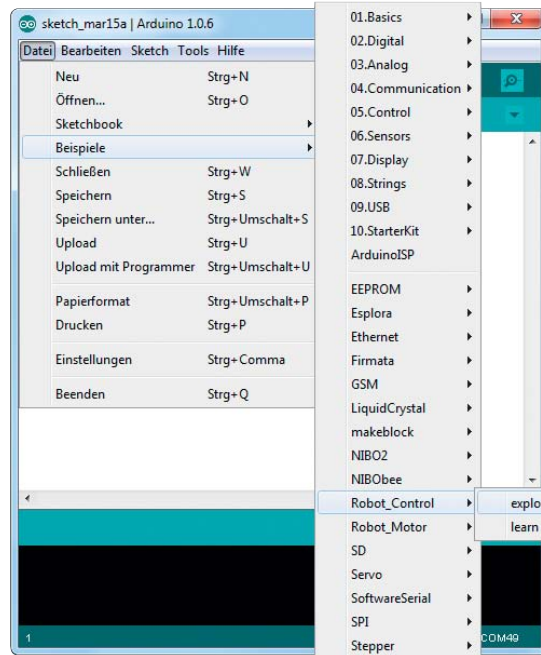
Einige Systeme wie Fischertechnik und Lego sind dafür didaktisch recht gut aufgebaut und leiten den Einsteiger durch den Aufbau der Modelle bis zu ersten Programmieraufgaben. Für Lego gibt es eine Fülle an deutscher Literatur und Tutorials. Andere Systeme, die etwa auf Arduino beruhen, setzen voraus, dass man sich das notwendige Rüstzeug zum Programmieren woanders besorgt. Unsere folgende kleine Auswahl soll bei der Einschätzung helfen, was zu den eigenen Zielen und Fähigkeiten passt.

Arduino Robot

Der Begriff Arduino beschreibt nicht einfach nur eine Platine, sondern ein ganzes Konzept des gleichnamigen Herstellers. Arduino steht für den einfachen Umgang mit offener Hardware und einen schnellen Einstieg in die Programmierung. Mit dieser Prämisse ist auch der Arduino Robot konzipiert.

Arduino Robot

Prozessor	ATmega32u4, 16 MHz
ROM, RAM	32 KByte, 2,5 KByte
Bedientasten	5
Display	✓
Sound	✓
Motoren	2
Bodensensoren	5
Hindernissensoren	–
Webseite	arduino.cc
Preis	225 €



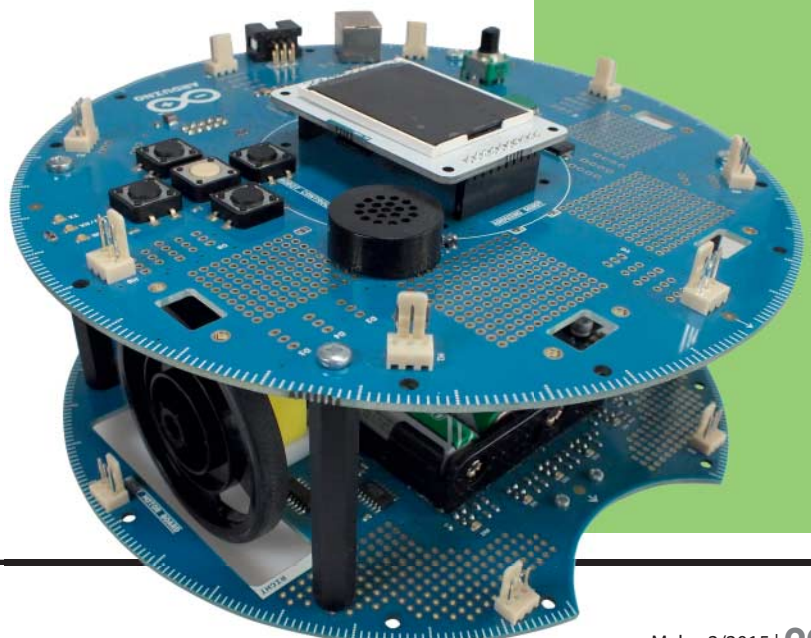
In der Arduino-IDE sind bereits Beispiele für den Arduino Roboter enthalten. Zu sehen sind hier auch die Einträge für den NiboBee und Makeblock.

R01_Logo
R02_Line_Follow
R03_Disco_Bot
R04_Compass
R05_Inputs
R06_Wheel_Calibration
R07_Runaway_Robot
R08_Remote_Control
R09_Picture_Browser
R10_Rescue
R11_Hello_User

Wer das Programmieren eines Arduino zu langweilig fand, das Selber-Nachrüsten mit Sensoren und Motoren plus Chassisbau aber scheute, findet in diesem Roboter einen prima Einstieg in das Thema. Der Roboter besteht aus zwei großen, übereinander gestapelten Platinen. Die untere Platine ist der „Motor Controller“, die vom oberen „Robot Controller“ Steuerbefehle entgegennimmt. Neben den zwei Motoren sind auf der unteren Platine fünf zum Boden gerichtete IR-Sensoren verbaut. Damit kann man beispielsweise (schwarze) Linien auf dem Boden erkennen und ihnen folgen.

Das Control Board beruht auf einem Arduino Leonardo, also einem 8-bittigen ATmega32u4 mit 16 MHz und 32 KByte Flash. Ihm sind ein Lautsprecher, 5 Buttons, ein LC-Display mit 160 × 120 Pixel, ein elektronischer Kompass sowie ein zusätzlicher

Eigentlich besteht der Roboter aus zwei Arduinos: oben der „Robo Controller“, unten der „Motor Controller“.



ROBOTER-PLATTFORMEN

EEPROM mit 512 kBit zur Seite gestellt. Das Display hat einen Slot für Micro-SD-Karten, auf dem man Grafiken und Bilder ablegen kann. Acht dreipolige Pfosten sind im Abstand von 45 Grad kreisförmig auf dem Board angebracht, an die sich zusätzliche IR-Empfänger oder Ultraschallabstandssensoren anschließen lassen.

Der Roboter wird von 4 Akku-Zellen angetrieben, die sich dank der Board-eigenen Ladeschaltung ohne separates Ladegerät wieder füllen lassen: Einfach das beigelegte 9-V-Netzteil anschließen und gut. Der Ladechip hat sich bei einem unserer Testexemplare allerdings sehr stark erhitzt, was Zweifel an der richtigen Dimensionierung aufkommen ließen.

Mangels Odometrie und nicht exakt gleich laufender Motoren fuhr unser Modell bei gleichen Motorstellgrößen statt geradeaus immer nur Rechtskurven. Nach einer Kalibrierung mittels Trimmer fuhr es leidlich geradeaus. Bei Drehungen auf der Stelle neigten die recht dünnen und auf der Motorwelle

iernden Rädchen zum Durchdrehen. Immerhin nutzen die in der Arduino-IDE (ab Version 1.0.5) angebotenen Programmierbeispiele den Kompass, um winkelgenaue Drehungen anzustreben. Mit der vorgefertigten Roboterbibliothek reduziert sich der Aufwand dafür auf eine einzige Zeile: `Robot.turn(90);`

Alles in allem ist der Roboter für Einsteiger eine Empfehlung, auch weil er durch seine Erweiterbarkeit für fortgeschrittene Projekte eine gute Basis bietet. Die fehlende Odometrie ist jedoch ebenso ein Manko wie ein fehlender Abstandssensor, insbesondere bei einem Preis von 225 Euro. Die Dokumentation ist nur online und nur auf Englisch verfügbar.

LEGO EV3

Den ersten Roboter-Baukasten der Mindstorms-Serie brachte Lego 1998 mit dem RCX auf den Markt, damals noch mit einem 8-Bit-Prozessor von Renesas, der mit 16 MHz lief. Der 2013 erschienene EV3 hat einen ARM9-Prozessor mit 300 MHz, 64 MByte RAM und 16 MByte Flash. Wie bei NXT gibt es vier Anschlüsse für Motoren und vier für Sensoren. Zusätzlich hat der EV3-Brick einen Slot für MicroSD-Karten, auf denen man ein eigenes Betriebssystem sowie andere Programmiersprachen installieren kann. Beim Vorgängermodell NXT musste man dafür noch das interne Flash neu programmieren, um etwa mit dem alternativen Java-basierten Mindstorms-Betriebssystem LeJOS arbeiten zu können. Der EV3 bootet stattdessen auf Wunsch von der SD-Karte.

Der Brick hat einen USB-Host-Port, an dem man beispielsweise weitere Peripherie für Robotikexperimente anschließen kann. Der Brick unterstützt auch angeschlossene WLAN-Dongles (wie den EW-7811UN von Edimax und andere). Bluetooth ist in den Brick bereits integriert, eine Steuerung per App auf dem Smartphone ist vorbereitet. Über einen USB-Client-Port kann man ihn mit dem PC verbinden, um eigene Programme hochzuladen.

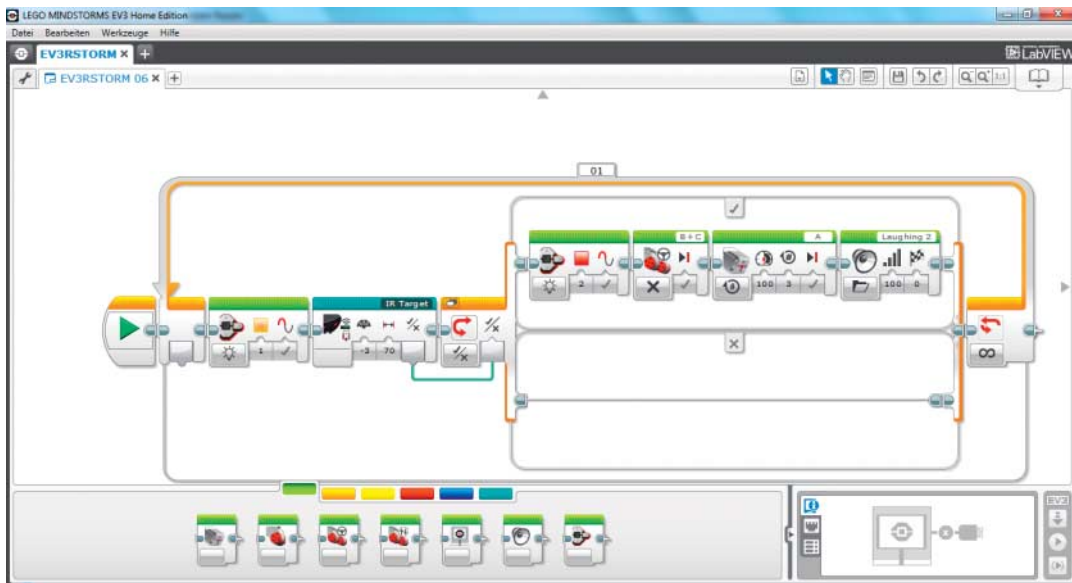
Zum Anschluss an den Brick sind dem Kasten in der Consumer-Variante drei Motoren mit Rad-Encodern (Odometrie), ein Farbsensor, ein Berührungssensor, ein IR-Sensor und eine IR-Fernbedienung zur Steuerung beigelegt. Zusammen mit mehr als 500 Lego-Technics-Teilen werden daraus autonome Roboter.

LEGO Mindstorms EV3

Prozessor	ARM9, 300 MHz
ROM, RAM	16 MByte, 64 MByte
Bedientasten	6
Display	✓
Sound	✓
Motoren	3
Bodensensoren	–
Hindernissensoren	1
Webseite	mindstorms.lego.com
Preis	ab 300 €

Dieses Lego-Modell verschießt auf Wunsch Kugeln.





Die quietschbunten Blöcke sind Funktionsblöcke für Motoren, Sensoren, Schleifen und Abfragen.

Der IR-Sensor kann je nach Anwendung Hindernisse erkennen oder als Empfänger der Fernbedienung fungieren beziehungsweise sie als Ziel verfolgen.

Auf der Brick läuft Linux, das die Anwenderprogramme ausführt. Selbige erstellt man in der EV3-Entwicklungsumgebung durch Zusammenschieben grafischer Elemente. Die Software liegt auf CD bei und ist in Deutsch. In der Software sind auch die Anleitungen zum Bau diverser Modelle integriert, die es jeweils in verschiedenen Ausbaustufen gibt.

Im Test haben wir den EV3Storm gebaut (siehe Bild), der auf „Ketten“ fährt und Kugeln verschießt. Die Anleitung führt Lego-typisch alle Schritte detailliert auf. Nach zwei Stunden Basteln stand das Modell auf den Beinen. Die Anwendung erstellt man in einer Oberfläche, auf der man bunte Blöcke mit Motor- und Sensorfunktionen zusammenschiebt. Das ist für Einsteiger ohne jegliches Vorwissen möglicherweise intuitiv. Menschen mit etwas Programmierkenntnissen dürften daran wenig Spaß haben. Glücklicherweise gibt es mit den C-ähnlichen Sprachen NQC und NXC gute Alternativen.

Insgesamt ist der EV3 hübsch anzusehen, kann fahren, schießen und ausweichen, allerdings fragt man sich, wofür der Brick eigentlich so viel Rechenleistung braucht. Im Vergleich zum NXT kann er eigentlich nicht viel mehr. Bei einem Basis-Preis von offiziell 350 Euro (Straßenpreis 300 Euro) kommt man ins Grübeln.

Interessant wird es, wenn man als fortgeschrittener Anwender die alternativen Programmiermöglichkeiten des EV3 nutzt. Dann eröffnen sich ganz neue Anwendungen, die die Kapazitäten der Brick auch voll ausschöpfen können. Lego-Fans bieten beispielsweise die Debian-Portierung ev3dev an, die bereits Unterstützung für Webcams und vieles mehr mitbringt.

Fischertechnik Robo TXT Discovery

War Fischertechnik in den siebziger und achtziger Jahren des vergangenen Jahrhunderts noch in vielen Kinderzimmern anzutreffen, muss man heutzutage selbst in Spielwarenabteilungen schon genauer hinschauen, um die Kästen zu finden. Dabei produziert Fischertechnik Robotikbaukästen schon seit der goldenen Ära der Heimcomputer.

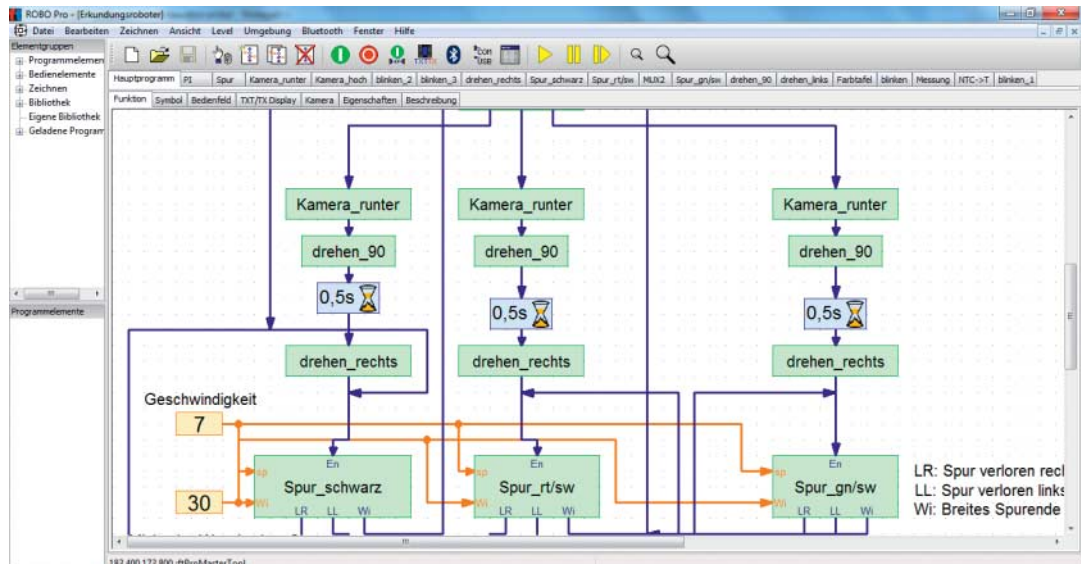
Der aktuelle Kasten „Robotics TXT Discovery Set“ beeindruckt mit seinem Controller, der gleich zwei Prozessoren beherbergt: einen ARM-Cortex-A8 mit 600 MHz und einen ARM Cortex M3. Mit 128 MByte RAM und 64 MByte Flash sind die Speicher vergleichsweise üppig ausgestattet. Mit seinem farbigem 2,4"-Touch-Display hat der Robotikkasten ein Alleinstellungsmerkmal. Auch bei den Kommunikationsschnittstellen hat Fischertechnik nicht gespart. Ein kombiniertes Funkmodul bietet Bluetooth 4.0 und WLAN 802.11 b/g/n. Eine Mini-USB-Buchse dient zum Anschluss an einen PC; an einen USB-Host-Port schließt man beispielsweise die zum Lieferumfang gehörende USB-Kamera an. Das auf dem Controller laufende Linux hat den passenden Treiber schon dabei.

Die Kamera dient unter anderem zur Erkennung von Objekten und der Linienverfolgung – auch die Bildübertragung per WLAN auf einen PC ist möglich. Zur Programmierung liefert der Hersteller spezielle Bildverarbeitungsroutinen mit.

Anders als bei den dedizierten Motor- und Sensoranschlüssen bei der EV3-Brick hat der Fischertechnik-Controller acht digital und analog ansteuerbare Universaleingänge, vier Zählgänge sowie vier PWM-Ausgänge. Der Anschluss der zwei mitgelieferten Motoren erfordert deshalb zwei Schritte: erstens Motoren an die PWM-Ausgänge und zweitens die Rad-Encoder an die Zählgänge

ROBOTER-PLATTFORMEN

Wie in den Anfangszeiten der Informatik: Die Anwendungen für den Robo TXT Discovery werden als Flussdiagramm entwickelt.



anschließen. Damit ist der Roboter Odometriefähig.

Wo wir gerade beim Thema Anschließen sind: Wie bei den ersten Fischertechnik-Elektronikkästen vor fast 50 Jahren kommen die Verbindungskabel nicht fertig konfektioniert, sondern müssen selbst mit Steckern versehen werden. In unserer Redaktion warf dies einen Schülerpraktikanten ob der Fummelarbeit kurzzeitig aus der Bahn. Also, eine Bitte an den Hersteller: Liefert die Kabel fertig.

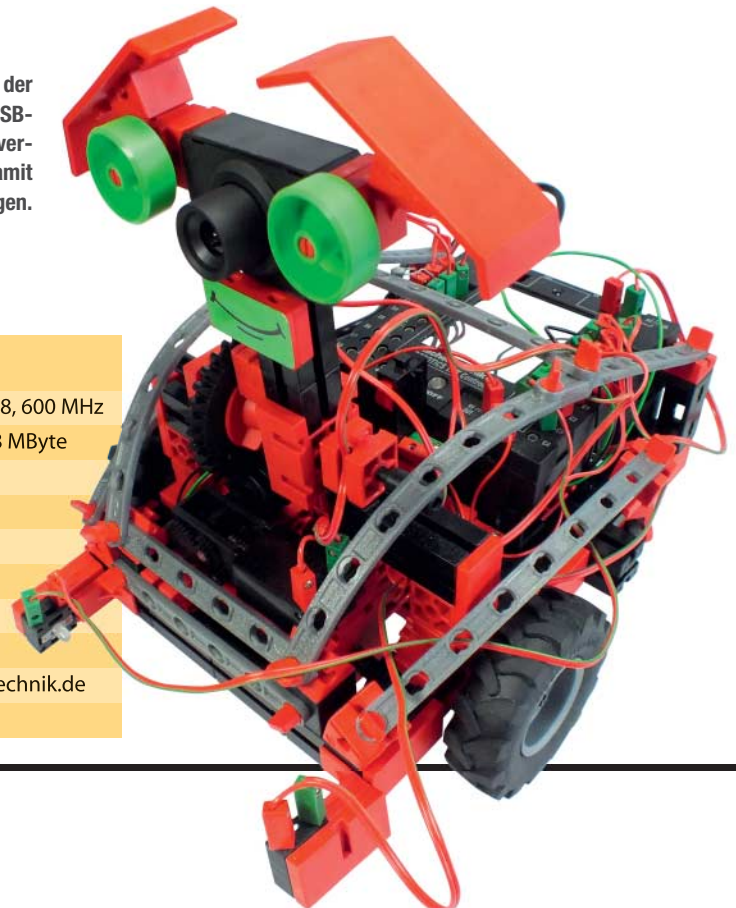
Neben den zwei Motoren mit Encodern hat Fischertechnik einen kleinen Motor, zwei LEDs, zwei Taster, einen Fototransistor und einen Temperatursensor in den Kasten gelegt. Im Test haben wir von den 14 Bauvorschlägen in der beiliegenden ausgedruckten (!) deutschen Anleitung den Erkundungsroboter nachgebaut. Die Montageschritte am Mo-

dell sind zwar isometrisch dargestellt. Bei einigen Schritten ist die Perspektive jedoch ungünstig, so dass wir mitunter ein paar Mal probieren mussten, um die Teile richtig zu platzieren. Der Roboter war nach rund zwei Stunden aufgebaut.

Der dem Kasten beiliegenden Software Robo Pro merkt man an, dass ihre Oberfläche bereits einige Jahre auf dem Buckel hat. Das tut der Bedienfreundlichkeit aber keinen Abbruch. Man verknüpft einzelne grafische Elemente für Motoren, Sensoren, Schleifen und Abfragen zu einem Flussdiagramm – fertig ist die Anwendung. Die Oberfläche und die Icons sehen in Legos IDE zwar schicker aus, wir fanden Robo Pro aber trotzdem eingängiger.

Der Kasten kostet alleine offiziell 370 Euro. Zusätzlich benötigt man ein Akku, wofür man nochmals 60 Euro über den Tisch reichen muss. Wir

Statt optischer Sensoren enthält der Fischertechnikkasten eine USB-Kamera. Zusammen mit der Bildverarbeitungssoftware kann man damit Objekte erkennen und Linien folgen.



Robo TXT Discovery

Prozessor	ARM Cortex A8, 600 MHz
ROM, RAM	64 MByte, 128 MByte
Bedientasten	Touchdisplay
Display	✓
Sound	✓
Motoren	3
Bodensensoren	-
Hindernissensoren	-
Webseite	www.fischertechnik.de
Preis	ab 350 €

haben aber bei Amazon auch Bundles mit Kasten und Akku für 350 Euro gesehen. Das ist zwar immer noch ein hoher Preis, dafür bekommt man aber auch einiges geboten. Weil Fischertechnik eher ein regionaler Anbieter ist, ist im Vergleich zu anderen Systemen die Unterstützung durch eine Community eher begrenzt. Eigentlich schade.

Robot-Arm

Mit seinen fünf Gelenken ist der gelb-schwarze Roboterarm schon nah dran an einem Industrieroboter – im Maßstab 5:1 und in Plastik. Der Arm kommt als

Der Roboterarm hat keinerlei Eigenintelligenz.



Robot-Arm

Prozessor	-
ROM, RAM	-
Bedientasten	6
Display	-
Sound	-
Motoren	5
Bodensensoren	-
Hindernissensoren	-
Webseite	www.owirobot.com
Preis	ab 70 €

Anzeige

ROBOTER-PLATTFORMEN

Bausatz, dessen Spritzgussteile man zunächst von den Stegen lösen und dann Gelenk für Gelenk zusammensetzen und verschrauben muss. In jedem Gelenk arbeitet ein Gleichstrommotor, dessen Plastikgetriebe man aus einzelnen Zahnrädern ebenfalls Stück für Stück zusammensetzt. Das ist alles dank der guten Anleitung recht kurzweilig.

Einzig die Verkabelung der einzelnen Motoren, die in der Basis zusammenlaufen, ist etwas fummelig. Dort wird auch die Hebelsteuerung angeschlossen, mit der man jedes einzelne Gelenk sowie den Greifer öffnen und schließen kann. Im Greifer ist zudem eine LED untergebracht, die man separat an- und ausschaltet. Da dem Roboterarm jegliche Eigenintelligenz fehlt, muss man die Hebel selber bedienen. Die Hebel haben aber keine lineare Funktion, geben also auch keine Position vor. Vielmehr schalten sie in den Hebel-Endpositionen den Motor in die eine oder andere Richtung an. In der Basis sind auch vier Mono-Batterien zur Stromversorgung untergebracht.

Mit dem USB-Zusatzmodul übernimmt der PC oder ein Raspberry Pi die Steuerung der Motoren. Dazu muss man die manuelle Steuerung ausbauen und das Modul einbauen. Für Windows liegt eine Software bei, mit der man den Arm per Mausklick steuert. Einen Vorschlag für eine eigene Anwendung zeigen wir auf Seite 38.

Was dem Arm fehlt, ist die Möglichkeit, die Stellung der einzelnen Motoren an einen PC zu senden. Damit wäre eine gezielte Bewegung möglich, statt nur durch Zeitvorgaben eine Position anzufahren.

Der Roboterarm und sein USB-Controller sind ein klassisches OEM-Produkt, das unter anderem als Vellemann-Produkt von Reichelt oder als Playtastic von Pearl angeboten wird.

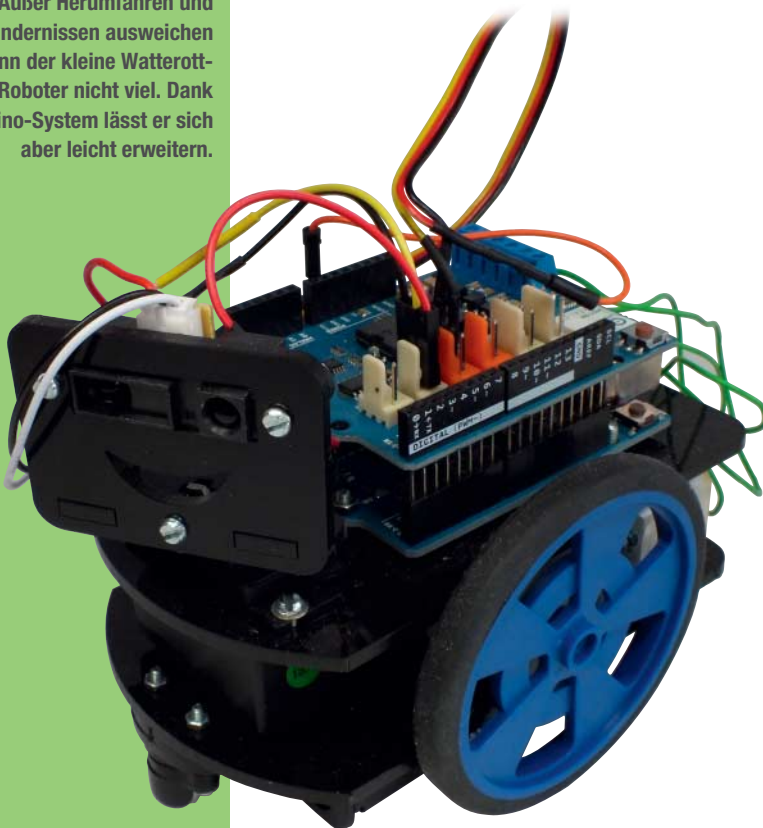
Watterott StarterKit Roboter V2

Ein Arduino mit Motor-Shield, zwei Getriebemotoren mit Rädern, ein Servo, Abstandssensor und ein Acryl-Chassis bilden den von Watterott angebotenen Mini-Roboter. Ursprünglich entstand er als ein Projekt des deutschen, mittlerweile eingestellten „Let’s make Robots“-Projekts. Der Roboter kommt in Einzelteilen und ist mit etwas Konzentration in rund anderthalb Stunden aufgebaut. Vorher sollte man die als PDF zum Download angebotene deutsche Aufbauanleitung jedoch einmal komplett durchlesen, um sich einen vollständigen Überblick darüber zu verschaffen, wo einige der sehr ähnlich aussehenden Acryl-Teile hinkommen.

An einigen Stellen ist das Eindrehen der Zylinderkopfschrauben in die Muttern etwas fummelig. Zu fest darf man die Schrauben auch nicht anziehen, sonst gibt es Risse im Acrylglas. Für erste Tests steht ein Arduino-Sketch zum Download bereit, der den Roboter vorwärts fahren lässt und den IR-Sensor auf dem Servo hin- und herschwenkt, um Hindernisse zu erkennen und ihnen auszuweichen. Das war’s dann leider auch schon, ab hier ist der Anwender auf sich allein gestellt. Blutigen Anfängern, die sich von Grund auf in den Aufbau des Roboters, seine Arbeitsweise und seine Programmierung einarbeiten wollen, sei das Buch von Markus Knapp „Roboter bauen mit Arduino“ aus dem Rheinwerk-Verlag – ehemals Galileo Press – empfohlen. Es widmet sich in wesentlichen Teilen genau diesem Roboter, liefert aber zusätzlich noch eine Einführung in die Elektronik.

Unser Roboter wollte nach dem Aufbau anfangs gar nicht fahren. Der Grund waren die Batterien: Selbst mit frischen Zink-Kohle-Batterien ruckte und rührte sich zunächst nichts. Mit frischen Alkali-Zellen lief er dann. Mangels gleichlaufender Motoren und fehlender Odometrie fuhr er allerdings nicht geradeaus. Alles in allem haut der Roboter einen nicht wirklich vom Hocker, da seine Fähigkeiten für immerhin 95 Euro doch sehr begrenzt sind. Mit zusätzlichen Shields ließe er sich jedoch aufpumpen.

Außer Herumfahren und Hindernissen ausweichen kann der kleine Watterott-Roboter nicht viel. Dank Arduino-System lässt er sich aber leicht erweitern.



StarterKit Roboter V2

Prozessor	ATmega328, 16 MHz
ROM, RAM	32 KByte, 2 KByte
Bedientasten	-
Display	-
Sound	-
Motoren	3
Bodensensoren	-
Hindernissensoren	1
Webseite	www.watterott.com
Preis	95 €

Nicai NiboBee

Der kleinen Roboter besteht buchstäblich nur aus Platinenmaterial und elektronischen Bauelementen. Selbst die Gehäuse zur Aufnahme der Motorgetriebe sind aus Platinen und müssen an den Kanten zusammengelötet werden. Sogar die Lager der Antriebswellen sind aus diesem Material, man muss sie rechtwinklig an der Unterseite festlöten. Anders als bei den anderen Roboterbausätzen und -kästen in diesem Artikel mit fertigen Steuer-Boards will beim NiboBee auch die Steuerplatine erst mal bestückt werden. Die deutsche Anleitung beschreibt das jedoch sehr genau, sodass kaum Fragen offen bleiben.

Der NiboBee setzt einiges an Lötferfahrung voraus, um ihn an drei Abenden zusammenzubauen. Dafür hat man aber anschließend einen Roboter, der durch die Lichtschranken im Getriebe nicht nur Odometrie-fähig ist, sondern auch Sensoren zum Folgen von Linien und Fühler zum Ausweichen von Hindernissen hat – und das Ganze für knapp über 50 Euro.

Da der NiboBee einen ATmega als Prozessor nutzt, kann man diverse freie Programmierertools nutzen. Neben dem AVRStudio bietet sich auch die Arduino-IDE an. Hersteller Nicai hat dafür sogar eine Roboterbibliothek und Beispiele erstellt, die sich nahtlos in die Arduino-IDE integrieren. Nervig ist allerdings, dass bei allen Beispielen die Wire-Bibliothek nicht inkludiert ist. Wir mussten im Testen zum Kompilieren jedes Mal `#include <Wire.h>` hinzufügen.

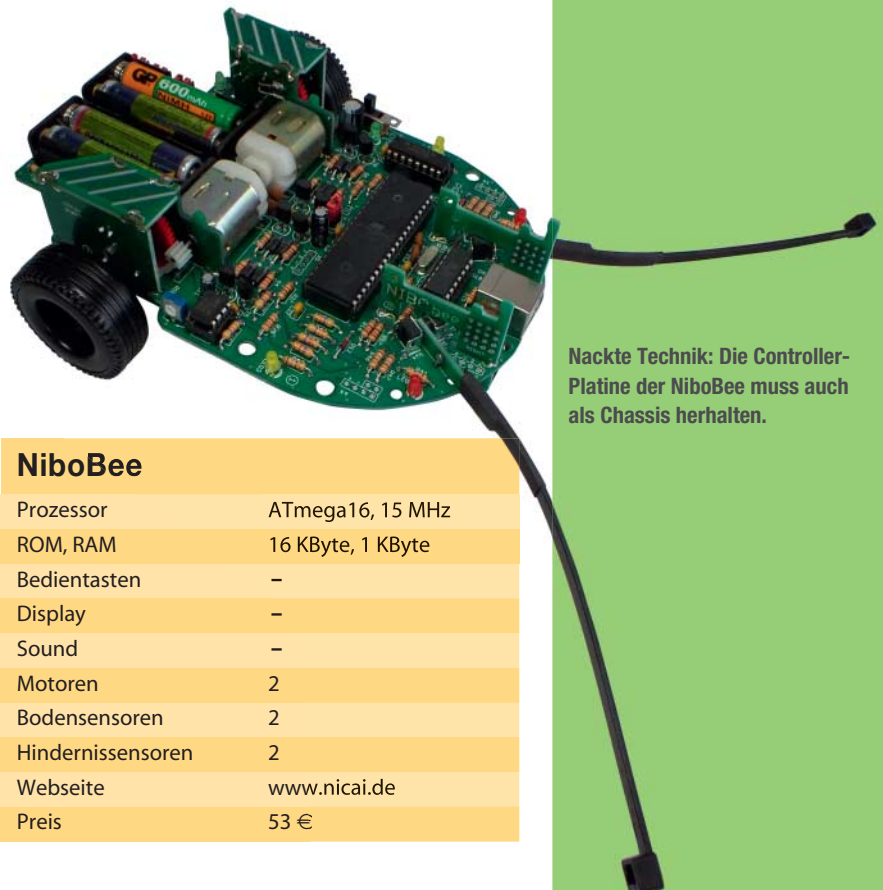
Zum Hochladen der Programme reicht ein USB-Kabel. Der Hersteller war so schlau, gleich einen Programmer (USBasp-kompatibel) mit in den Roboter zu integrieren. Zur Übertragung mit diesem Programmer gibt es in der Arduino-IDE die Funktion „Upload mit Programmer“ – also einfach die Shift-Taste halten und Upload klicken.

Über den Online-Compiler auf der Webseite robot.cc kann man auch Java-Programme für den NiboBee entwickeln. Die kompilierten Hex-Dateien muss man zunächst mit dem Browser herunterladen und anschließend mit dem Tool RoboDude auf den Roboter laden. Zum Programmieren muss der NiboBee immer Batterien enthalten und angeschaltet sein, die Stromversorgung beim Anschluss an den PC schaltet nämlich nicht den ATmega an.

In unserem Test waren die Getriebe sehr laut. Eines der Räder war etwas schwergängig, weil vermutlich das Getriebegehäuse nicht exakt ausgerichtet war. Hindernissen wich der Roboter nach Berührung aus. Der Hersteller bietet ein Erweiterungsset für 56 Euro, das ein Display und 4 Tasten zum Nachrüsten enthält. Über fünf Erweiterungsport lassen sich auch leicht eigene analoge Sensoren oder solche mit I2C-Schnittstelle nachrüsten.

Makeblock Starter Robot Kit

Alu statt Plastik ist die Devise des US-Herstellers Makeblock, der neben Arduino-Experimentier-Kits auch Roboter im Portfolio hat. Wir haben das Star-



Nackte Technik: Die Controller-Platine der NiboBee muss auch als Chassis herhalten.

NiboBee

Prozessor	ATmega16, 15 MHz
ROM, RAM	16 KByte, 1 KByte
Bedientasten	–
Display	–
Sound	–
Motoren	2
Bodensensoren	2
Hindernissensoren	2
Webseite	www.nicai.de
Preis	53 €

ter Robot Kit-Blue (Bluetooth Version) unter die Lupe genommen. Statt zu stecken (Lego) oder zu schieben (Fischertechnik) wird bei Makeblock geschraubt. Die blau eloxierten Alu-Profile, Räder und Platten sehen edel aus. Der Aufbau des Chassis und die Befestigung der zwei Motoren ist eigentlich simpel, zog sich jedoch aufgrund einer Abweichung der gelieferten Motoren zu den in der beigelegten Anleitung gezeigten etwas in die Länge. Auch die Verdrahtung der Motoren ist in der Praxis später anders als in der Anleitung gezeigt.

Auf das Chassis mit den fetten Raupenkettchen kommt der Ultraschallabstandssensor, ein Arduino-kompatibles Board und der Batteriehalter. Zum Verkabeln haben das Board und die Sensoren RJ25-Buchsen, die man mit den mitgelieferten vorkonfektionierten Kabel verbindet. Die Dokumentation zur Software-mäßigen Inbetriebnahme gibt es nur Online und auf englisch. Neben der Installation der Arduino-IDE muss man noch die Makeblock-Bibliothek manuell im richtigen Ordner installieren – für Anfänger ist das mitunter fehlerträchtig. Daneben ist unter Windows ein zusätzlicher Treiber für das Board notwendig.

Solange auf dem Starter Kit das Bluetooth-Modul angeschlossenem war, ließ sich in unserem Test jedoch kein Sketch auf das Arduino-Uno-kompatible Board laden. Erst wenn wir das Kabel vom BT-Modul abzogen, gelang der Upload. Mit dem Ultrasonic-Car-Sketch ging es dann flott zur Sache.

ROBOTER-PLATTFORMEN



Mit den kräftigen Getriebemotoren und den fetten Gummiketten entwickelt der Makeblock-Roboter einen überraschend guten Vortrieb.

Mit den Ketten und den starken Motoren überfuhr der Roboter alles, was im Weg war – es sei denn, er erfasste ein Hindernis per Sensor. Dann wich er aus. Mit Bluetooth-Modul und passendem Sketch steuert man per Smartphone-App den Roboter manuell.

Zwar hatten die Motoren einen guten Gleichlauf und das Gefährt damit einen guten Geradeauslauf. Mangels Odometrie kann der Roboter aber keine genauen Drehungen machen und auch keine vorgegebenen Entfernungen fahren. Motoren mit Rad-Encoder und passender Elektronik kosten 75 Euro zusätzlich. Weitere Sensoren, Treiber

Starter Robot Kit-Blue

Prozessor	ATmega238, 16 MHz
ROM, RAM	32 KByte, 2 KByte
Bedientasten	–
Display	–
Sound	✓
Motoren	2
Bodensensoren	–
Hindernissensoren	1
Webseite	www.makeblock.cc
Preis	149 €

und Alu-Teile gibt es einzeln. Ein Liniensensor kostet beispielsweise ab 20 Euro. Alles in allem ist Makeblock ein schönes modulares System und im Vergleich zu den anderen Kästen eine stabile Lösung gelungen, die auch mal ruhig draußen rumrollern kann, ohne dass bei rauem Gelände gleich Teile abfallen. Wer sich dafür interessiert, sollte statt des Starter Robot Kit Blue für 150 Euro besser das ohne Bluetooth für 50 Euro weniger wählen und dafür einige zusätzliche Sensoren kaufen. Zusätzlich hat Makeblock noch einen Roboterarm im Programm, der sich auf das Raupenchassis montieren lässt. —dab

↓ Links und Foren
make-magazin.de/x89s

AUSSER KONKURRENZ

Als Roboterfan hat es mir schon lange in den Fingern geuckt, mal mit Hasbros „Interactive R2D2 Astromech“ rumzuspielen und auf seine „Hackbarkeit“ zu untersuchen. Letztes Jahr war die Gelegenheit und der Preis auf Amazon günstig. Der Roboter ist ein reines Spielzeug und nicht im eigentlichen Sinne programmierbar. Immerhin reagiert er leidlich auf Sprachbefehle und tanzt dann beispielsweise, macht seine Taschenlampe an oder folgt einem durch die Wohnung.

Angetrieben wird er von zwei Motoren in seinen Beinen; im mittleren Bein ist eine Rolle unterbracht. R2D2 kann seinen Dome – also die oberen Halbkugel – drehen und macht genau die Geräusche, die

man aus den Filmen kennt. Leider verhält sich der Droide auch wie in den Filmen – er ist eigentlich öfter bockig, als dass er auf die Befehle hört. Ab und zu steigert sich die Bockigkeit in vollständige Ignoranz – dann hilft nur ein leichter Klaps auf den Kopf und schon hört er wieder. Für 200 Euro ist das allerdings zu nervig.

Das Gehäuse bietet indes viel Raum zum Einbau eigener Spielereien. Einige Enthusiasten haben die Elektronik ausgebaut und einen Raspberry Pi nebst Webcam eingebaut. Der Pi übernimmt dann die Kontrolle und kann ebenfalls via Sprachingabe auf Befehle reagieren. Das ist auch mein nächstes Projekt für eine der kommenden Ausgaben.



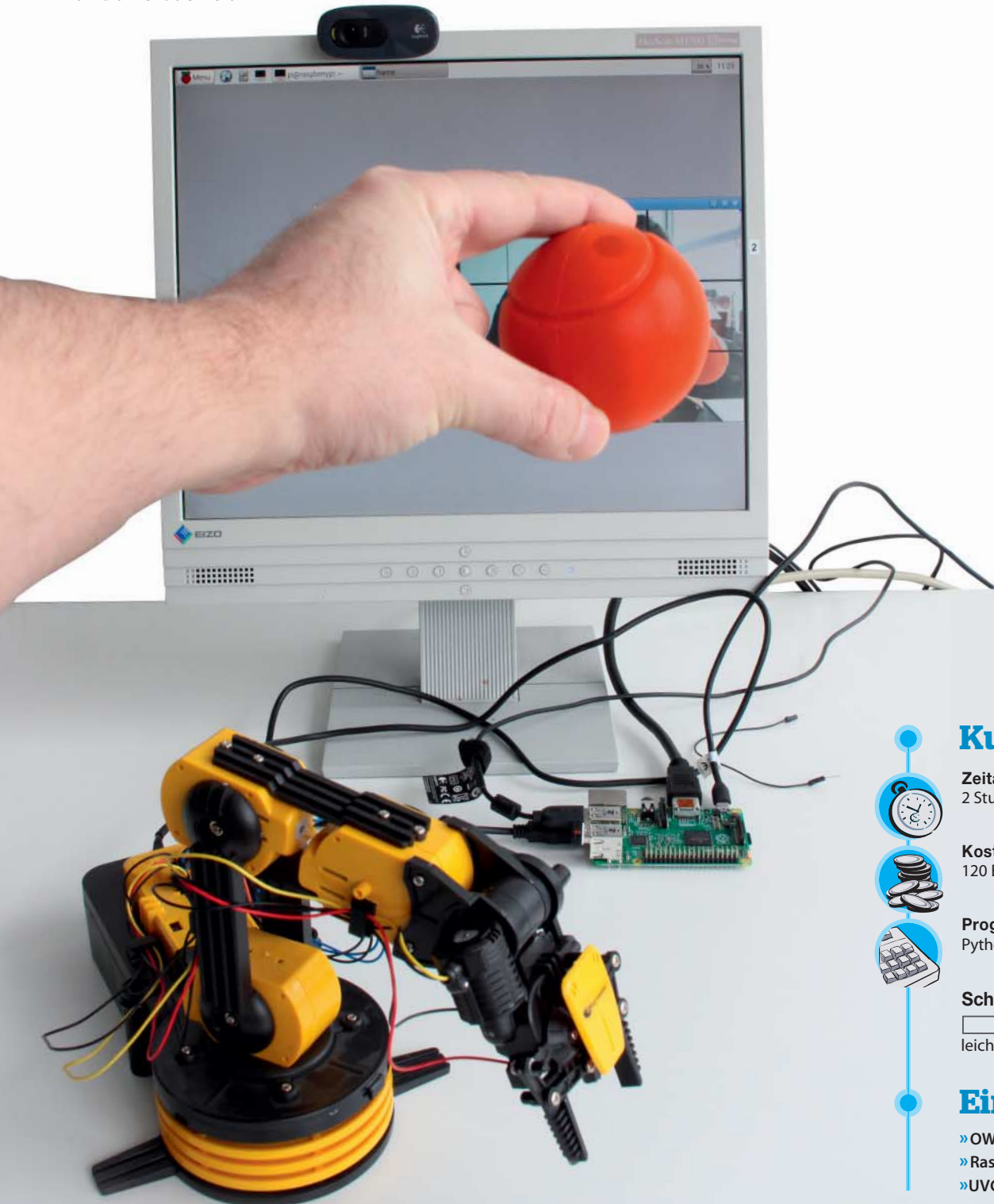
Sieht zwar super aus, bietet aber langfristig trotzdem keinen Spaß.

Anzeige

Am Ball bleiben

Mit Webcam und Bilderkennungssoftware wird der Raspberry Pi 2 zum Objekt-Tracker, der einen Roboterarm in verschiedene Richtungen fahren lässt. Das kann man als Geschicklichkeitsspiel verwenden oder mit wenigen Änderungen zur Zielverfolgung.

von Daniel Bachfeld



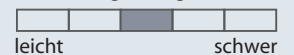
Kurzinfo

Zeitaufwand:
2 Stunden

Kosten:
120 Euro

Programmieren:
Python

Schwierigkeitsgrad



Einkaufsliste

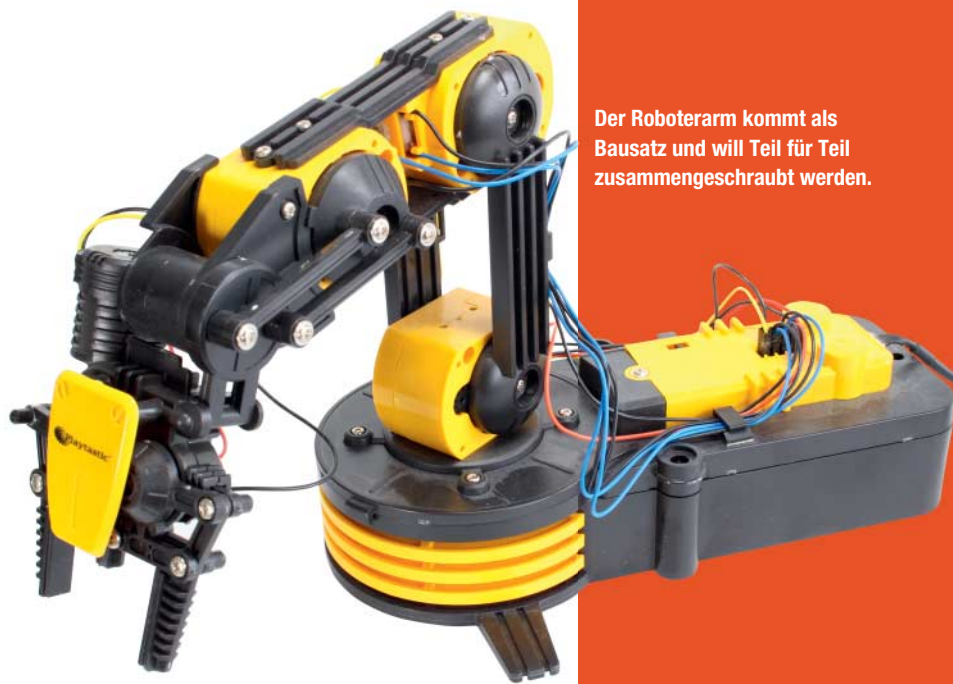
- » OWI-Roboterarm
- » Raspberry Pi 2
- » UVC-kompatible Webcam

Der Hersteller Robotikits stellt einen günstigen Roboterarm mit drei Gelenken, einer drehbaren Basis und einem Greifer für 30 US-Dollar her, der sich per Hebel manuell bedienen lässt. Die deutschen Distributoren Pearl und Reichelt haben den Arm als „Baukasten Roboter-Arm“ für 60 Euro im Angebot. Im Bundle mit einem USB-Modul lässt sich für 20 Euro mehr jeder einzelne Motor vom PC aus ansteuern. Die mitgelieferte Windows-Software ist sehr rudimentär – sie verlagert im Grunde nur die manuelle Steuerung von Bedienhebeln auf Buttons auf einer grafischen Oberfläche.

Dank der freien Bibliothek libusb kann man verhältnismäßig einfach eigene Befehle per USB verschicken und so den Arm mit selbstgestrickten Anwendungen steuern. Am einfachsten geht das unter Linux-Derivaten. Exemplarisch zeigen wir das am Beispiel des Raspberry Pi 2 und Raspbian (Wheezy). Dort ist die Bibliothek libusb bereits vorinstalliert. Prinzipiell funktioniert der hier gezeigte Weg aber auch auf vielen anderen Linux-Systemen. Vielleicht muss das eine oder andere Paket zusätzlich installiert werden.

Weil der Autor ein großer Fan von Python ist und der Pi ja nicht umsonst so heißt, wie er heißt, kommt Python zum Programmieren von libusb zum Einsatz. Praktischerweise steht sogar ein fertiger Wrapper dafür bereit, der die Python-Module zum Ansprechen der libusb mitbringt: PyUSB. Der Wrapper ist zwar über die Pi-Repositories verfügbar, allerdings funktionierte diese Version in unseren Tests nicht. Wir haben deshalb die Version aus dem Python Package Index (PyPI) installiert. Dazu muss man den Paketmanager pip installieren: `sudo apt-get install python-pip`. „pip“ ist ein rekursives Akronym und steht für „pip installs packages“. Mit `sudo pip install pyusb` installiert man dann PyUSB.

Unser Skript verwendet nur zwei (gewrappte) Funktionen von libusb: `usb.core.find` und `ctrl_transfer`.



Der Roboterarm kommt als Bausatz und will Teil für Teil zusammenschraubt werden.

Erstere dient dazu, den richtigen USB-Endpoint festzulegen und zu testen, ob er überhaupt angeschlossen ist: `RoboArm = usb.core.find(idVendor=0x1267, idProduct=0x0000)`.

Mit dem Befehl `ctrl_transfer()` kommuniziert man mit dem Endpoint, um Daten abzufragen oder Steuerdaten zu senden. Die Funktion erwartet die sechs Parameter (`requestType`, `request`, `value`, `index`, `byte`, `timeout`). Der Wert `requestType` gibt die Richtung des Requests an; `0x40` fürs Senden und `0xc0` fürs Empfangen. Welche

WRAPPER

Viele Software-Bibliotheken lassen sich nur mit C/C++ nutzen, was gerade für Programmieranfänger eine ziemlich hohe Hürde darstellt. Mit Wrappern lassen sich solche Bibliotheken auch von anderen Sprachen aus ansprechen. Sie fungieren als ein Adapter zwischen den verschiedenen Sprachen. Beispiele dafür sind die im Artikel benutzten Wrapper PyUSB und Python-OpenCV.

UVC-KOMPATIBLE WEBCAM

Nach wie vor sind Webcams unter Linux ein spezielles Thema. Während die Hersteller für ihre Produkte immer mindestens Windows-Treiber mitliefern, ist man unter Linux auf frei verfügbare Treiber angewiesen. Je verbreiteter ein Modell, desto höher die Wahrscheinlichkeit, dass es auch einen Treiber für Linux gibt.

Am wenigsten Probleme gibt es mit Webcams, die die USB Device Class Definition for Video Devices (UVC) erfüllen. Der dazu passende UVC-Treiber ist unter den meisten Linux-Systemen als Kernel-Modul verfügbar. Eine Liste von Webcams, die den UVC-Standard einhalten, gibt es

unter www.ideasonboard.org/uvc/#devices. Auch nicht-UVC-kompatible Kameras laufen mit dem Raspberry Pi, sofern der GSPCA-Treiber (Generic Software Package for Camera Adapters) sie unterstützt. Ob er das tut, sieht man nach dem Anstöpseln der Kamera und dem Eingeben des Befehls `dmesg` an einer Meldung am Ende der Ausgabe wie „`usbcore: registered new interface driver gspca`“. In unseren Tests gab es bei den GSPCA-Treibern allerdings in Zusammenspiel mit der auf dem Pi-Image vorinstallierten OpenCV-Version 2.4.1 Probleme bei der Bilddarstellung. Mit einer neueren OpenCV-Version verschwanden sie.

PYUSB-VERSIONEN

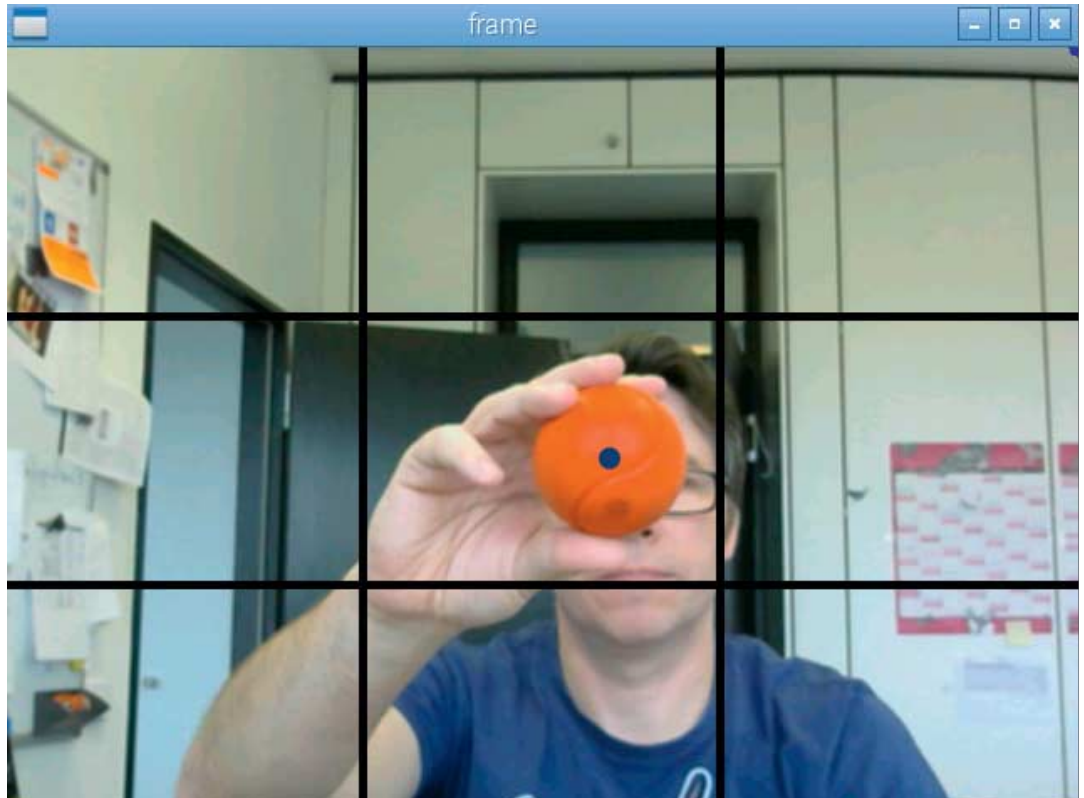
Mit der Raspbian-Version 2015-02-16-raspbian-wheezy war über pip bereits die PyUSB-Version 1.0.0b2 verfügbar, die ohne Fehler läuft. Auf anderen Linux-Systemen mussten wir zur Installation dieser Version die Option `--pre` angeben, also `sudo pip install --pre pyusb`. Andernfalls kommt es später beim Import der USB-Funktionen `usb.core` etc zu Fehlermeldungen.

OPENCV SELBER ÜBERSETZEN

Um OpenCV auf den neuesten Stand (Anfang März 2.4.11) zu bringen, muss man die Bibliotheken selbst übersetzen. Das hat den Vorteil, dass auch mit dem GSPCA-Treiber arbeitende Webcams besser funktionieren. Zusätzlich erhält man durch das Selbstübersetzen Bibliotheken, auf die man in der vorkompilierten Version keinen Zugriff hat. Da einige Funktionen patentiert sind, dürfen sie nämlich nur als Quellcode und nicht als fertige Pakete verteilt werden.

Ein Anleitung zum Kompilieren finden Sie auf unserer Webseite unter [make-magazin/x9e9](#)

Die Steuerung des Roboters hängt davon ab, in welchen Feld die Kamera den Ball erkennt.



Dass der Raspberry ein Objekt erkannt hat, sieht man am blauen Punkt auf dem Ball.

Greifer auf	Ellenbogen hoch	Greifer zu
Basis links	neutral	Basis rechts
Schulter hoch	Ellenbogen runter	Schulter runter

Steuerbefehle

Befehl	Bedeutung
MoveArm(1,[0,1,0])	Basis nach links
MoveArm(1,[0,2,0])	Basis nach rechts
MoveArm(1,[64,0,0])	Schulter hoch
MoveArm(1,[128,0,0])	Schulter runter
MoveArm(1,[16,0,0])	Ellenbogen hoch
MoveArm(1,[32,0,0])	Ellenbogen runter
MoveArm(1,[4,0,0])	Handgelenk hoch
MoveArm(1,[8,0,0])	Handgelenk runter
MoveArm(1,[2,0,0])	Greifer auf
MoveArm(1,[1,0,0])	Greifer zu
MoveArm(1,[0,0,1])	Licht an
MoveArm(1,[0,0,0])	Licht aus/Motoren aus

genauen Werte der Roboterarm erwartet, haben findige Maker durch Mitlesen der USB-Kommunikation herausgefunden: `ctrl_transfer(0x40, 6, 0x100, 0, [Befehl], 1000)`.

Der Parameter [Befehl] ist ein Array aus drei Bytes, deren Bits die Richtung der insgesamt 5 Motoren steuern: [32,0,0] fährt den Ellenbogen runter; [16,0,0] fährt in wieder hinauf, [0,0,0] stoppt alle Motoren. Will man mehrere Motoren gleichzeitig bewegen, so muss man die jeweiligen Werte per UND verknüpfen. Eine Auflistung der einzelnen Steuerbefehle ist im Kasten links zu finden.

Zur Vereinfachung haben wir im Skript eine Funktion definiert, die den Umgang mit dem Arm abstrahiert:

```
def MoveArm(Duration, ArmCmd):
    RoboArm.ctrl_transfer(0x40,6,0x100,0,ArmCmd,1000)
    time.sleep(Duration)
    ArmCmd=[0,0,0]
    RoboArm.ctrl_transfer(0x40,6,0x100,0,ArmCmd,1000)
```

Die Funktion erwartet als Parameter die Zeitspanne, die ein Motor angeschaltet werden soll, sowie den Motor-Parameter.

Verfolger

Damit lassen sich erste eigene Steueranwendungen programmieren. Den Roboter einfach nur über ein paar Befehle zu lenken, fanden wir jedoch ein bisschen langweilig. Deshalb haben wir das Ganze mit der freien Bilderkennungsbibliothek OpenCV (CV für

RAKETENWERFER

Neben dem Roboterarm lassen sich auch andere USB-Gadgets über libusb und PyUSB kontrollieren – man muss nur die richtigen Kommandos kennen. Auch beim Rocket Launcher des Herstellers Dream Cheeky haben Hacker einfach mit einem USB-Monitor-Programm die Kommunikation belauscht – oder die Windows-Treiber auseinandergeplückt. Der Raketenwerfer ist beispielsweise bei Amazon und anderen Anbietern ab 30 Euro verfügbar. Einen Link zu einem Steuer-Skript finden Sie unter den Links am Ende des Artikels.

Der Raketenwerfer lässt sich wie der Roboterarm per libusb und pyusb steuern.



Anzeige

Computer Vision) verknüpft. Eine Webcam nimmt ein Bild auf und versucht darin ein Objekt mit einer bestimmten Farbe zu erkennen, beispielsweise einen roten Ball. Je nachdem, in welchem Abschnitt sich der Ball befindet, bewegt sich der Roboterarm. Ist der Ball etwa links im Bild, dreht der Arm nach links. Ist der Ball oben, bewegt sich das Schultergelenk nach oben.

Das Paket OpenCV ist in der ziemlich veralteten Version 2.4.1 im Wheezy-Image bereits enthalten, reicht aber für unsere Zwecke. Um die Bibliothek mit Python nutzen zu können, muss man nur noch den Python-OpenCV-Wrapper nachinstallieren: `sudo apt-get install python-opencv`. Danach lädt man das unter dem Link am Ende des Artikels zum Download angebotene Python-Skript und startet es mit `python owidemo.py`.

Listing 1

```
1 import cv2
2 import numpy as np
3 import usb.core, usb.util, time
4
5 cap = cv2.VideoCapture(0)
6
7 while(1):
8     _,frame = cap.read()
9     frame = cv2.blur(frame,(3,3))
10    hsv = cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
11    thresh = cv2.inRange(hsv,np.array((10, 100, 100)),
12                        np.array((30, 255, 255))) # gelb
13    contours,hierarchy = cv2.findContours(thresh,
14                                        cv2.RETR_LIST,cv2.CHAIN_APPROX_SIMPLE)
15    max_area = 0
16    for cnt in contours:
17        area = cv2.contourArea(cnt)
18        if area > max_area:
19            max_area = area
20            best_cnt = cnt
21
22    if max_area > 0:
23        M = cv2.moments(best_cnt)
24        cx,cy = int(M['m10']/M['m00']),
25                int(M['m01']/M['m00'])
26        cv2.circle(frame,(cx,cy),5,255,-1)
27
28        print cx,cy,"\\n"
29
30        if cx < 213:
31            if cy >160 and cy < 320:
32                print "rechts\\n"
33                MoveArm(1,[0,2,0])
34                ...
35                ...
36
37    cv2.line(frame,(0,160),(650,160),2,4,0)
38    cv2.line(frame,(0,320),(650,320),2,4,0)
39    cv2.line(frame,(213,0),(213,480),2,4,0)
40    cv2.line(frame,(426,0),(426,480),2,4,0)
41    cv2.flip (frame, 1, frame)
42    cv2.imshow('frame',frame)
43
44    if cv2.waitKey(33)== 27:
45        break
```

Je nachdem was die Kamera sieht, kann sich der Roboterarm nun bereits in eine Richtung bewegen. Am besten macht man erste Versuche in einem hellen Raum mit einer hellen Wand. Hält man nun den roten Ball vor die Kamera, sollte sich der Roboterarm entsprechend der Ballposition bewegen.

So funktioniert

Listing 1 zeigt den wichtigsten Ausschnitt des Python-Skripts. Die Zeilen 1 bis 3 importieren die verwendeten Bibliotheken. Zeile 5 legt das erste verfügbare Video-Device als Eingabe fest (/dev/video0).

Ab Zeile 7 startet die Hauptschleife. Zeile 8 liest einen Frame aus der Webcam ein. Das seltsame Konstrukt `_,frame` kommt zustande, weil die OpenCV-Funktion ein Tupel als Rückgabewert liefert, also zwei Werte statt einem. Da wir nur das Bild benötigen, landet der erste Parameter durch die Angabe `_` quasi im Mülleimer.

Das Bild wird in den Zeilen 9 und 10 aufgehellt und in den HSV-Farbraum konvertiert (hue-saturation-value), aus dem OpenCV farbrige Objekte leichter extrahieren kann. Zeile 11 legt den oberen und unteren Farbwert fest, den das Skript zum Suchen der Objekte benutzen soll. Im zum Download hinterlegten Skript haben wir Zeilen für rote, blaue und gelbe Objekte vordefiniert. Gegebenenfalls muss man die Schwellwerte an seine eigenen Objekte anpassen.

Die Hauptarbeit leistet die Funktion `cv2.findContours()` in Zeile 12. Sie durchsucht den aufgenommenen Frame nach Farbflächen und liefert eine Liste von Konturen zurück. Die Konturen enthalten wiederum jeweils eine Liste von Punkten als Umriss der gefundenen Fläche. Der Parameter `cv2.CHAIN_APPROX_SIMPLE` begrenzt die Zahl der Punkte auf einen ungefähren Umriss. Mit `cv2.CHAIN_APPROX_NONE` würden sämtliche Punkte des Umrisses zurückgeliefert, was recht speicherintensiv sein kann.

Mit den Zeilen 13 bis 18 wird die größte Kontur gefunden. Nur wenn es überhaupt eine Kontur mit der gewünschten Farbe gibt, verarbeitet das Skript die Befehle ab Zeile 21. `cv2.moments` liefert das sogenannte „Image Moment“ zurück, ein fiktiver Wert, der gewichtete Mittelwerte aus den Helligkeitswerten einzelner Pixel darstellt. Die Zeile 22 berechnet die x- und y-Koordinaten des „Centroid“ dieses Moments, was gewissermaßen den Schwerpunkt darstellt. An diese Koordinaten zeichnet `cv2.circle` einen Punkt.

Der Rest ist einfach: Je nach Position wird einer der Armbefehle aufgerufen. `cv2.line` zeichnet Linien zur Abgrenzung der Steuerfelder ins Bild, damit man weiß, wohin man den Ball halten muss. Der Befehl `cv2.flip()` in Zeile 36 spiegelt das Bild, sodass man sich und seine Bewegungen wie im Spiegel sieht und nicht seitenverkehrt agieren muss. Der Befehl `cv2.imshow()` zeichnet schließlich das Bild inklusive Punkt und Linien auf den Bildschirm. Gegebenenfalls bricht die While-Schleife in Zeile 40 ab, wenn die Escape-Taste gedrückt ist.

Mit ein bisschen Übung kann man den Arm relativ gut steuern. Wir haben den Aufbau bei einer Veranstaltung als Geschicklichkeitsspiel eingesetzt. Das Skript lässt sich mit wenigen Änderungen auch zum Steuern von Servomotoren einsetzen, beispielsweise um ein Ziel im Auge zu behalten. Denkbar ist eine Nachführung von Solar-Paneln, eine Kamera für Tieraufnahmen und vieles mehr. —*dab*

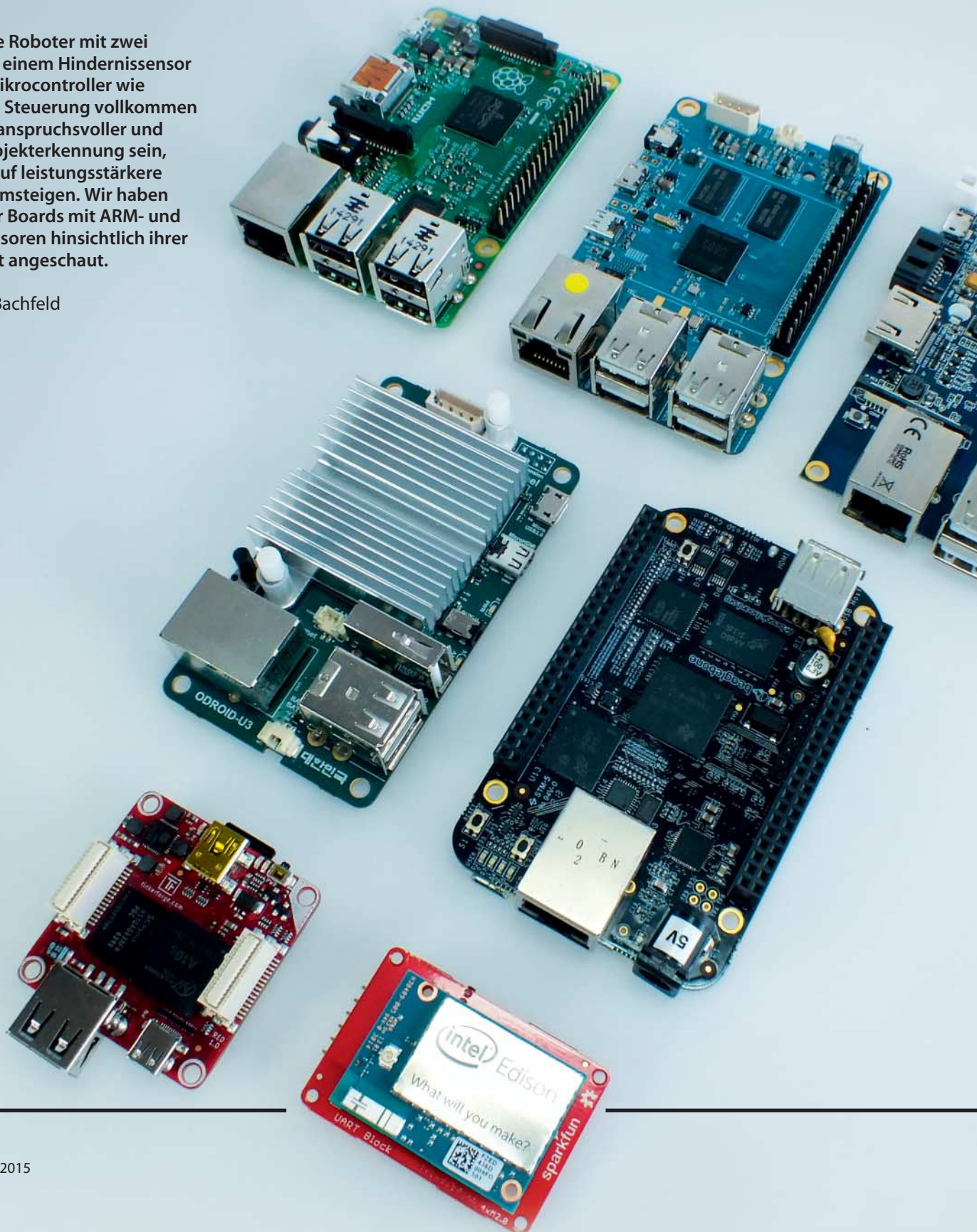
 [Links und Foren
make-magazin.de/x9e9](http://make-magazin.de/x9e9)

Anzeige

Bretter, die die Welt bedeuten

Für einfache Roboter mit zwei Rädern und einem Hindernissensor reicht ein Mikrocontroller wie Arduino zur Steuerung vollkommen aus. Soll es anspruchsvoller und etwa mit Objekterkennung sein, muss man auf leistungsstärkere Lösungen umsteigen. Wir haben uns ein paar Boards mit ARM- und Intel-Prozessoren hinsichtlich ihrer Tauglichkeit angeschaut.

von Daniel Bachfeld



Jeder fängt mal so an: Linienfolger, Hindernisausweicher und Lichtsucher sind typische erste Roboteranwendungen. Schnell kommt jedoch der Wunsch nach mehr auf: Ein Roboter, der sehen und erkennen kann, der Sprache versteht, selber sprechen und sogar gezielt durch die Wohnung navigieren kann. Dafür reichen ATmega- und PIC-Mikrocontroller jedoch nicht mehr aus. In Zeiten vor dem Raspberry Pi setzten Roboterbauer deshalb oft auf Netbooks mit Atom-Antrieb, weil die einen guten Kompromiss zwischen Größe, Leistungsfähigkeit und Stromverbrauch boten. Der Raspberry Pi war Anfang 2012 für Roboterbauer ein neuer Meilenstein: klein, sparsam und mit vielen Schnittstellen zum direkten Anschließen von Sensoren.

Seitdem kommen ständig neue Boards auf den Markt, die den Pi übertrumpfen und sogar leistungsfähiger als die früheren Atom-Netbooks sind. Damit sind auch komplexere Anwendungen wie das Robot Operating System (ROS) auf kleinen Robotern lauffähig. ROS ist eine riesige Sammlung fertiger Robotik-Funktionen, auf die der Entwickler in seiner Robotik-Anwendung zurückgreifen kann.

Unsere Übersicht verschiedener Boards soll bei der Auswahl für das eigene Robotik-Projekt helfen. Allerdings muss man sich beim Umstieg über eines im klaren sein: Alle Boards laufen mit einem Betriebs-

system – und zwar mit Linux. Wer damit auf Kriegsfuß steht, hat schlechte Karten und sollte besser auf Mini-PCs nach Intels NUC-Konzept oder Gigabytes Brix-Serie setzen.

Statt der bekannten Leistungsmesser wie Coremark und SysMark haben wir einen aus unserer Sicht aussagekräftigeren Benchmark eingesetzt. Ein in Python geschriebenes Skript versucht anhand der OpenCV-Bibliothek Gesichter in Bildern zu erkennen, die eine Webcam liefert. Um ein gefundenes Gesicht zeichnet das Skript einen Rahmen und stellt das Bild auf dem Display dar. Die Zahl der Frames pro Sekunde haben wir als Vergleichswert für die Boards genommen. Wie das Skript funktioniert, lesen Sie ab Seite 62

Neben der Leistungsfähigkeit ist die Zahl der unterstützten Schnittstellen und I/O-Funktionen wichtig. Die auf den Platinen aufgelöteten Chips sind in der Regel komplette System-on-a-Chip-Lösungen (SoC), die nicht nur den Prozessor beherbergen, sondern auch den Video-Chip, das Speichermanagement, Netzwerk, USB sowie zusätzliche Schnittstellen und eben die I/O-Funktionen. Je mehr das Board davon bereits mitbringt, desto weniger Zusatz-ICs muss man nachrüsten.

Nicht alle Funktionen sind auf eigene Pins nach außen geführt. Um physische Pins zu sparen, benutzen die SoCs einen Multiplexer, dessen Einstellungen festlegen, ob beispielsweise Pin 10 auf einer Pfostenleiste auf die ADC-, PWM-, UART- oder GPIO-Funktion im SoC geschaltet ist. Eine kurze Erklärung der Funktionen ist im Glossar unten zu finden.

Mittels Software lässt sich auf jedem Pin ein PWM-Signal erzeugen. Allerdings ist diese Form der Signalerzeugung nicht zuverlässig. Durch kleine Laufzeitunterschiede der

Software variiert die Breite der Pulse manchmal, was sich bei Servos durch Zittern bemerkbar macht. Bei echter Hardware-PWM gibt es eigene Prozessorregister, in die man die gewünschte PWM-Frequenz und die Pulsbreite hineinschreibt, und ein dedizierter Timer kümmert sich um die Signalerzeugung. In den Produkttabellen unten haben wir nur HW-PWMs vermerkt.

Sofern man die Boards in mobilen Robotern einsetzt, sollte man der Spannungsversorgung besonderes Augenmerk schenken. Insbesondere die Odroid-Boards und der BeagleBone Black reagieren mit seltsamen Fehlverhalten, wenn die Spannungsquelle nicht genügend Strom liefert. USB-Netzteile sind damit oft überfordert. Für den Einsatz in größeren Robotern bieten sich LiPo- oder Blei-Gel-Akkus mit Spannungswandlern an, die mindestens 2A zur Verfügung stellen.

Obwohl die Boards meist eine Versorgungsspannung von 5 Volt erwarten, arbeiten sie auf den GPIO-Ports nur mit 3,3 Volt, beim Edison sogar nur mit 1,8 Volt. Viele externe Sensoren und andere ICs erwarten jedoch einen Pegel bis 5 Volt. Zwar kann ein 3,3-Volt-Ausgang einen 5-Volt-Eingang ansteuern, ohne dass es zu Schäden kommt, umgekehrt geht das jedoch meist in die Hose. 5 Volt killen 3,3-Volt-Eingänge, es sei denn, der Eingang ist explizit als spannungstolerant für 5 Volt deklariert. Das ist aber bei keinem der hier vorgestellten Boards der Fall. Zum Koppeln nimmt man deshalb Pegelwandler (Level-Shifter), die eine Umsetzung der Spannungen vornehmen.

 [Links und Foren
make-magazin.de/xhjp](http://make-magazin.de/xhjp)

GLOSSAR

ADC: Analog Digital Converter. Damit lassen sich analoge Spannungen digitalisieren. Viele Sensoren liefern nur analoge Spannungen, die man auf diesem Wege weiterverarbeiten kann.

CAN: Controller Area Network ist ein serielles Bussystem (CAN_L, CAN_H), das in der Industrie und in Fahrzeugen zur Vernetzung von Steuereinheiten eingesetzt wird.

GPIO: General Purpose I/O bezeichnet die Ports, die je nach Anforderung als Ein- oder Ausgang konfiguriert werden können oder weitere Sonderaufgaben erfüllen.

I2C: Inter-Integrated Circuit bezeichnet einen Zweidraht-Bus (SDA und SCL) zur Datenübertragung von ICs untereinander.

IR: Infrarot-Empfänger. In Kombination mit einer Fernbedienung kann man das Board steuern.

PWM: Pulsweitenmodulation. Dient zum einen zur Positionssteuerung von Servo-Motoren und zum anderen zur Leistungssteuerung bei Gleichstrommotoren.

SPI: Serial Peripheral Interface (MISO, MOSI, SCLK) ist eine synchrone serielle Schnittstelle für Kurzstreckenverbindungen, die mit Taktraten bis in den MHz-Bereich arbeitet.

USB-Host: USB-Port zum Anschluss von Peripherie-Geräten wie Webcams, USB-Sticks, Eingabegeräten.

USB-OTG: On-the-Go-Port, ändert seine Funktion abhängig davon, woran er angeschlossen ist. Am Host-Port eines PC wird der Port zum Client-Port und gewährt beispielsweise Zugriff auf den Flash-Speicher. Ist über einen OTG-Adapter hingegen ein Gerät angeschlossen, schaltet der Port auf Host-Funktion.

UART: Universal Asynchronous Receiver Transmitter, die herkömmliche, vom PC bekannte, serielle Datenübertragung (RX, TX).

BANANA PI

Wie der Name vermuten lässt, platziert sich der Banana Pi als direkter Konkurrent zum Raspberry Pi. Dazu haben sich die Entwickler sowohl an der Größe der Platine als auch an der Anordnung der Anschlüsse orientiert. Zudem ist die Belegung der I/O-Pins auf der Steckerleistung nominell die gleiche, damit für den Pi verfügbare Erweiterungskarten zum Aufstecken und die dazugehörige Software auch auf dem Banana Pi funktionieren – und das trotz des anderen Chips. Soweit zur Theorie: In der Praxis funktioniert das nur mit den extra vom Hersteller für den Banana angepassten Images; beispielsweise für Raspbian. Darin ist eine Vermittlungsschicht enthalten, die die Umnummerierung zwischen den Pins des A20 und den originären Pi-Pins vornimmt.

Daneben steht noch Ubuntu, Fedora, OpenSuse, ArchLinux und Android zur Wahl. Zusätzlich gibt es für die bekannte WiringPi-Bibliothek, die den einfachen Zugriff auf die GPIOs aus C, Python und anderen Sprachen erlaubt, eine Portierung auf den Banana Pi (WiringBP).

Mit der Kraft der zwei Herzen und dem doppelten Speicher überflügelt der Banana Pi den alten Raspberry Pi hinsichtlich der Rechenleistung locker, der Facetracker läuft mit 10 bis 11 fps. Mit seinen vielen Schnittstellen ist er prima für Robotik-Anwendungen geeignet und unterstützt sogar den im Industriebereich eingesetzten CAN-Bus. Durch seinen SATA-Anschluss und dem Gigabit-Ethernet ist er auch als Medienserver hervorragend geeignet. Mit LeMedia steht eine Distribution zur Verfügung, die auf XBMC beruht. Die seit kurzem für rund 44 Euro angebotene Pro-Version des Banana Pi bringt noch einen integrierten WLAN- und Bluetooth-Adapter mit und hat 40 statt 26 I/O-Ports.

Banana Pi

Prozessor	A20 ARM Cortex-A7 Dual-Core, 1 GHz
Grafik	ARM Mali-400M P2 (OpenGL ES 2.0/1.1)
RAM	1 GByte DDR3
OS-Speicher	SD
Video	HDMI, CVBS, LVDS
Audio	HDMI, Klinke
Netzwerk	Gigabit Ethernet
SATA	1
USB-Host	2
USB-OTG	1
I/O	26
I/O-Spannung	3,3 V
5V-tolerant	-
PWM	2
I2C	2
UART	1
SPI	1
IR	1
ADC	2
CAN	1
Versorgung	USB, 5 V / 2 A
Webseite	www.lemaker.com
Preis	50 €



Die schwarze Buchse unten links neben dem HDMI-Port ist der SATA-Anschluss.

Der Banana Pi hat einen Slot für normale SD-Karten.



BEAGLEBONE BLACK

Das kurz BBB genannte Board richtet sich mit seinen vielen I/O-Funktionen in erster Linie an Elektroniker, die damit Steuerungen bauen möchten. Zwar ist es mittlerweile seit zwei Jahren auf dem Markt, davon gab es aber anderthalb Jahre Lieferengpässe. So hat das Board trotz seines vergleichsweise geringen Preises wenig Anhänger gefunden.

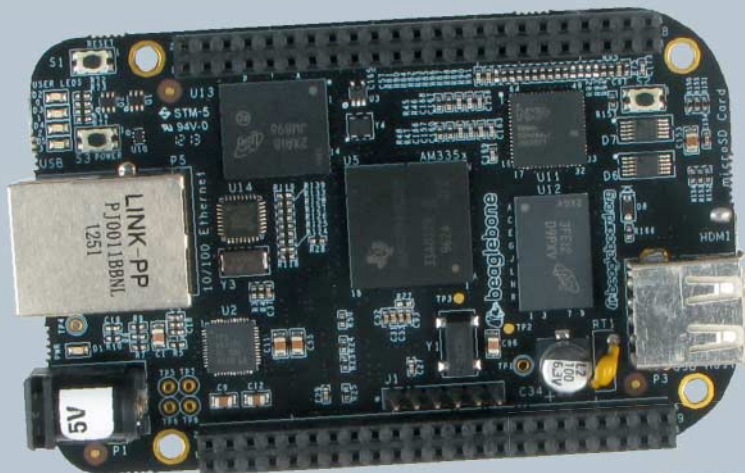
Der BBB hat von allen Boards die meisten I/O-Ports, die auf zwei Pfostenleisten mit je 46 Anschlüssen verteilt sind. Darauf lassen sich sogenannte Capes mit diversen Funktionen aufstecken, ähnlich den bekannten Shields bei Arduino. Apropos Arduino: Der BBB-Hersteller stellt mit Bonescript eine Node.js-Library zur Verfügung, mit der sich der BBB im Befehlsstil von Arduino programmieren lässt.

Eine Besonderheit des BBB sind seine zwei mit 200 MHz getakteten Programmable Real-Time Units (PRU) mit jeweils eigenem Speicher, Interrupts und I/O-Ports. Sie arbeiten wie 32-Bit-Mikrocontroller und können mit der ARM-CPU kommunizieren. Die Programmierung der PRUs ist aber eher etwas für professionelle Entwickler.

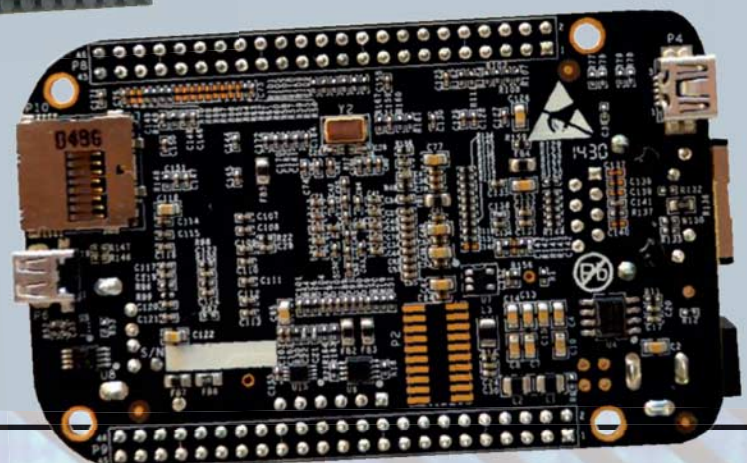
Das Board kann vom internen eMMC-Flash booten, in dem standardmäßig die Linux-Distribution Angstrom installiert ist. Wahlweise bootet das System von einer Micro-SD-Karte, auf der beispielsweise Ubuntu, Debian oder Android installiert ist. Im Test mit einem Ubuntu 14.04 mit LXDE-Desktop flackerte die ganze Zeit die Grafikdarstellung. Mit unserer UVC-kompatiblen Kamera hatten wir mit unserem Testskript keinen Erfolg. Mit einer GSPCA-Kamera kam der Facetracker auf immerhin 4 bis 5 fps.

BeagleBone Black

Prozessor	Sitara, ARM Cortex-A8, 1 GHz
Grafik	SGX530 3D
RAM	512 MByte DDR3L
OS-Speicher	MicroSD(UHS-1), eMMC4 GByte Onboard
Video	Micro-HDMI
Audio	HDMI
Netzwerk	10/100
SATA	-
USB-Host	1
USB-OTG	-
I/O	65
I/O-Spannung	3,3 V
5V-tolerant	-
PWM	8
I2C	2
UART	4
SPI	2
IR	-
ADC	7
CAN	2
Versorgung	Hohlbuchse, 5 V / 2 A
Webseite	www.beaglebone.org
Preis	50 €



Auf die schwarzen Pfostenleisten steckt man die Capes.



Der BeagleBone Black hat nur einen einzigen USB-Host-Port. Bei mehr Peripherie braucht man USB-Hubs, die aber unter Linux gerne mal den Dienst versagen.

INTEL EDISON

Das briefmarkengroße Board ist das einzige mit einem x86-kompatiblen Atom-Prozessor – trotzdem läuft im Moment nur Linux statt Windows drauf. Seine zwei Kerne arbeiten mit 500 MHz. Da der Atom-Prozessor keine I/O-Ports hat, hat Intel ihm einen Quark-Prozessor mit 100 MHz zur Seite gestellt, der diese Aufgabe übernimmt und mit über 45 GPIOs aufwartet. Mit an Bord sind 1 GByte RAM, 4 GByte Flash-Speicher, WLAN mit 2,4 und 5 GHz sowie Bluetooth 4.0.

Alles drin, alles dran, allein man kommt nicht ran – das Board lässt sich eigentlich nur mit zusätzlichen Erweiterungsboards wie dem Edison Kit for Arduino oder dem Breakout-Board benutzen. Letzteres führt allerdings die Pins nur physisch nach außen, ohne die Spannungslevel von 1,8 V auf eine im Bastlerbereich normale benutzbare Spannung jenseits von 3,3 V anzuheben.

Klein und praktisch sind die von Sparkfun angebotenen Edison-Blocks, die nicht viel größer als der Minicomputer selbst sind und einfach aufgesteckt werden. Neben Modulen zur Motorensteuerung gibt es A/D-Wandler, Gyro-Sensoren und Platinchen mit Pegelwandlern. Über USB-Port-Module schließt man weitere Peripherie an. Um den Umgang mit den Ports zu vereinfachen, hat Intel die Bibliothek MRAA veröffentlicht. Die Lib lässt sich mit C/C++, Python und JavaScript nutzen, mappt die teilweise dreistelligen internen Portnummern auf einprägsamere Nummern und stellt Funktionen für deren Konfiguration bereit. Ein Python-Skript zum Steuern zweier Motoren und für Hinderniserkennung reduziert sich so auf 20 Zeilen.

Der Edison hat keinen Videoausgang und ist nur als Embedded-PC mit Shell-Zugriff gedacht. Trotzdem haben wir versucht, mit unserem Facetracker die Frames zu messen und kamen auf 5 fps.

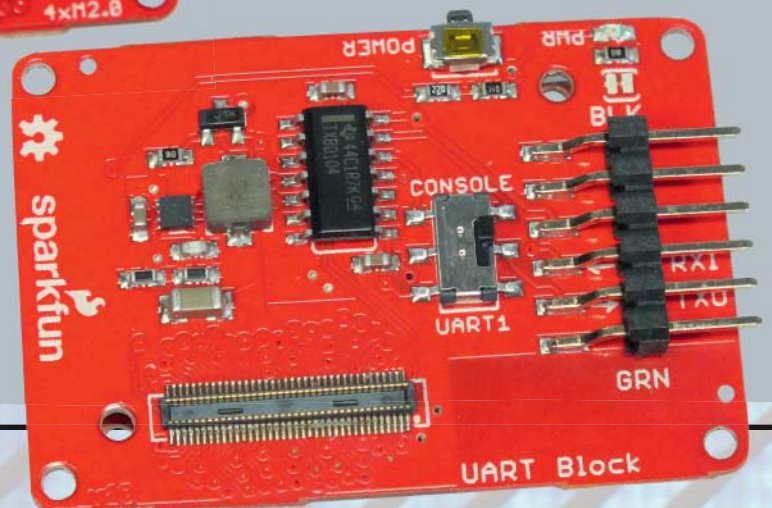
Intel Edison

Prozessor	Atom, 64 Bit, 500 MHz
Grafik	-
RAM	1 GByte
OS-Speicher	4 GByte Flash
Video	-
Audio	-
Netzwerk	WLAN
SATA	-
USB-Host	1
USB-OTG	1
I/O	46
I/O-Spannung	1,8 V
5V-tolerant	-
PWM	4
I2C	2
UART	2
SPI	1
IR	-
ADC	-
CAN	-
Versorgung	abhängig vom Board
Webseite	www.intel.com
Preis	50 €



Der Edison ist nur die kleine grüne Platine. Er steckt auf einem Block-Modul von Sparkfun.

Die Sparkfun-Platine hat Pegelwandler integriert, um die 1,8 V auf verarbeitbare Spannungen zu heben.



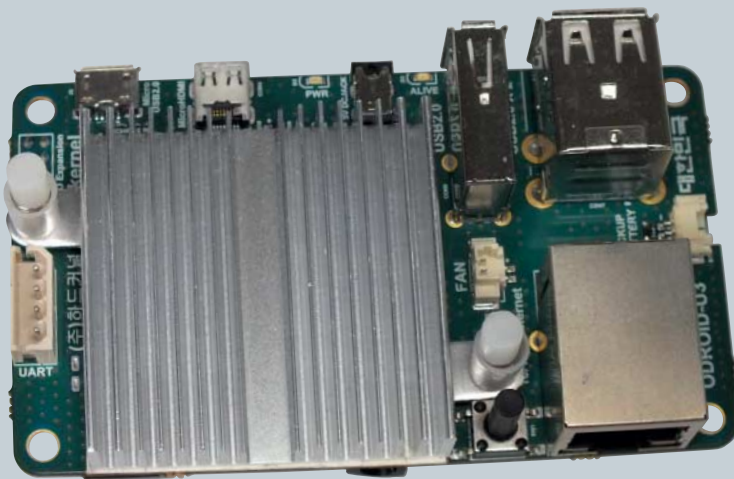
ODROID U3

Das Odroid-U3-Board hat dank 4 Kernen und 2 GByte RAM ordentlich Dampf im Kessel, um Robotik-Anwendungen ruckellos antreiben zu können. Der Prozessor kam ursprünglich im Samsung Galaxy S III zum Einsatz. Das U3 hat selbst nur wenige I/O-Ports. Auf einem 8-poligem Header stehen UART, I2C und GPIO bereit, ein zweiter 4-poliger-Header bietet Anschluss an den SPI-Bus. Über die Header lassen sich Erweiterungsboards anschließen. So gesehen ist das Board nicht die erste Wahl, wenn es darum geht, Servos und DC-Motoren anzusteuern oder analoge Messwerte von Sensoren einzulesen. Andererseits kann der U3 durch seine Rechenleistung Robotik-Aufgaben wie Pfadplanung, Lokalisierung, Objekterkennung und Spracherkennung erledigen, die mit anderen Mini-Computern oder reinen Mikrocontrollern nur schwer umzusetzen sind. Unser Odroid U3 kam mit unserem Facetracker auf 16 bis 18 fps und war damit das schnellste System im Feld.

Für den Mini-Rechner stehen diverse Linux-Images zum freien Download bereit, darunter Ubuntu 14.04 mit dem schlanken LXDE-Desktop sowie OpenSuse, Fedora, Debian und Android. Der Odroid hat eine ziemlich aktive Community, die die Images regelmäßig aktualisiert und anpasst sowie das monatlich erscheinende, kostenlose „Odroid Magazine“ herausgibt. Der Roboterforscher Christopher D. McMurrough hat beispielsweise die „Ubuntu 14.04 Robotics Edition for ODROID“ zusammengestellt, in der bereits das Robotik-Framework ROS (Indigo), OpenCV und die Point Cloud Library PCL für 3D-Anwendungen installiert sind. Daneben sind auch bereits Treiber für die 3D-Tiefenkamera Asus Xtion Pro (Live) enthalten.

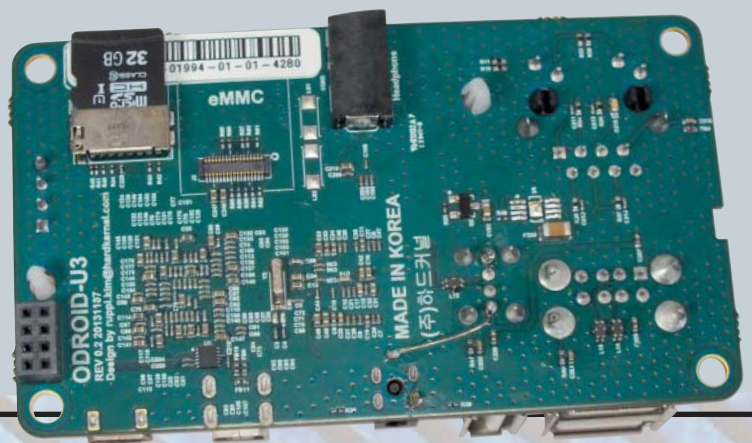
Odroid U3

Prozessor	ARM Cortex-A9 Quad-Core 1,7 GHz mit 1 MB L2 Cache
Grafik	ARM Mali-400M P2 (OpenGL ES 2.0/1.1)
RAM	2 GByte DDR2
OS-Speicher	MicroSD (UHS-1), eMMC 4.5 nachrüstbar
Video	Micro-HDMI
Audio	HDMI, Klinke
Netzwerk	10/100
SATA	–
USB-Host	3
USB-OTG	1
I/O	9
I/O-Spannung	3,3 V
5V-tolerant	–
PWM	–
I2C	1
UART	1
SPI	1
IR	–
ADC	–
CAN	–
Versorgung	Hohlbuchse, 5 V / 2 A
Webseite	www.hardkernel.com
Preis	70 €



Viel Rechenleistung, aber wenig I/O-Ports: Nur ein Header für UART und 8 I/O-Leitungen sind vorhanden.

Das Odroid-Board lässt sich auf der Unterseite mit schnellem eMMC-Speicher ausstatten.



ODROID C1

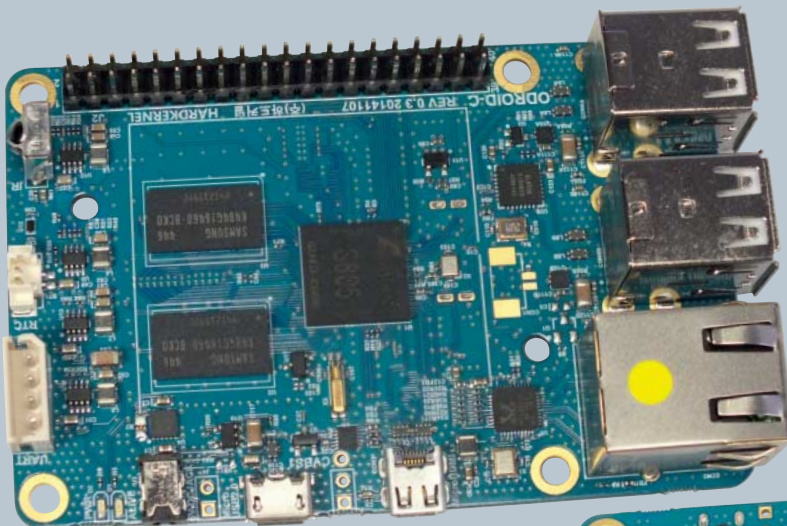
Das Odroid C1 ist klar gegen den Raspberry Pi 2 B+ positioniert, sowohl was den Preis, Formfaktor als auch die Leistungsfähigkeit angeht. Anders als das Odroid U3 hat das C1-Modell eine 40-polige Stiftleiste mit zahlreichen I/O-Ports. Die Belegung ist weitgehend identisch mit dem Pi. Wer nicht genau hinschaut, könnte den C1 glatt mit einem Pi verwechseln: die USB-Schnittstellen, der RJ45-Port und sogar die Platinenbohrungen sind an den gleichen Stellen.

Für das Odroid C1 gibt es praktischerweise eine Portierung der WiringPI-Bibliothek. Damit wird das Steuern der I/O-Ports aus eigenen Programmen heraus einfach. Die Portierung unterstützt bereits die zwei analogen Eingänge des Hardkernel-Boards. Während die anderen GPIOs alle mit maximal 3,3 Volt Eingangs- beziehungsweise Ausgangsspannung arbeiten, vertragen die zwei ADC jedoch nur 1,8 Volt. Zwar passen Erweiterungsplatinen für den Pi auf das Odroid C1 sowohl mechanisch als auch signaltechnisch, allerdings können genau diese beiden Ports den fehlerfreien Betrieb verhindern.

Leistungsmäßig liegt das Vierkern-Board fast gleichauf mit dem U3: 17 fps erreichte unsere Facetracker-App mit einem LXDE-Ubuntu 14.04. Auch als Mediacenter eignet sich das C1. Zwar muss man auf SATA verzichten, dafür kann die Mali-GPU H.265-kodierte Filme in Echtzeit dekodieren. Etwas wackelig wirkt der billige MicroSD-Kartenslot, der seltsam weit von der Platine absteht und die Karte nur zu einem Drittel aufnimmt. Im Betrieb kam es aber nie zu Kontaktproblemen mit der Karte.

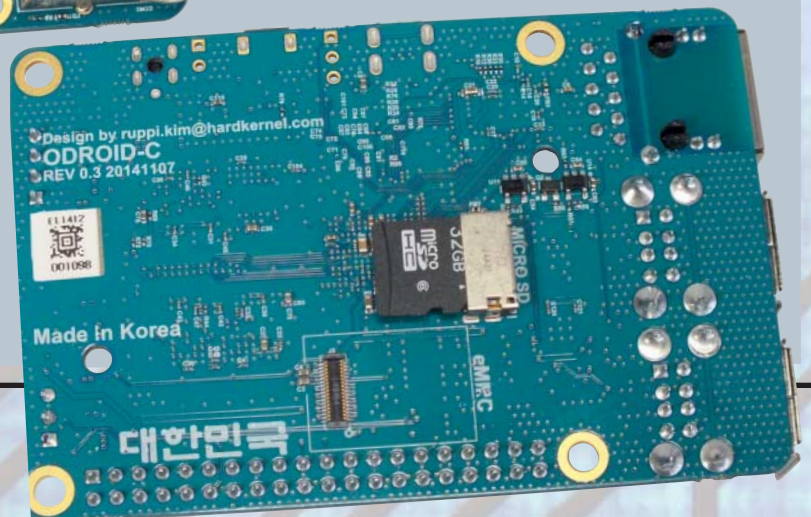
Odroid C1

Prozessor	Amlogic S805 Cortex A5 (ARM v7), 1,5 GHz, QuadCore
Grafik	Grafik: Mali-450 MP2 GPU (OpenGL ES 2.0/1.1)
RAM	1 GByte DDR3
OS-Speicher	MicroSD (UHS-1), eMMC 4.5 nachrüstbar
Video	Micro-HDMI
Audio	HDMI
Netzwerk	Gigabit Ethernet
SATA	-
USB-Host	4
USB-OTG	1
I/O	27
I/O-Spannung	3,3 V
5V-tolerant	-
PWM	2
I2C	2
UART	2
SPI	1
IR	1
ADC	2
CAN	0
Versorgung	Hohlbuchse, 5 V / 2 A
Webseite	www.hardkernel.com
Preis	44 €



Das C1-Board ähnelt stark einem Raspberry Pi 2, ist aber schneller und besser ausgestattet.

Der SD-Kartenslot ist unpraktischerweise auf der Mitte der Unterseite statt am Rand platziert.



RASPBERRY PI 2 B

Der Raspberry Pi 2 hat mit seinen drei zusätzlichen Kernen und 512 MByte mehr Speicher zwar ordentlich aufgeholt. Es reicht leider nicht, um an den anderen Boards vorbeizuziehen. Dennoch spielt er weiterhin eine wichtige Rolle – einfach, weil er bei 5 Millionen verkauften Exemplaren in drei Jahren weiterhin das Maß aller Dinge ist. Alle Hersteller von Zusatzplatinen, Software und Gadgets orientieren sich an seiner Form, seinen I/O-Pins und den verfügbaren Linux-Distributionen.

Mit 9 bis 11 fps lief auch die Fracetracker-Anwendung ausreichend flüssig, um den Pi 2 in spannenden Robotik-Anwendungen einsetzen zu können. Auch die bislang nur im Raspbian-Image enthaltene Wolfram-Engine profitiert vom Leistungszuwachs. Die Wolfram Language ist eine interessante Alternative bei der Verarbeitung von Sensordaten und Bilddaten. Unter anderem unterstützt Wolfram Funktionen für „Maschinelles Lernen“, mit denen man einem Computer beibringen kann, Fotos von männlichen und weiblichen Personen zu unterscheiden.

Ferner hat Microsoft angekündigt, eine Embedded-Version von Windows 10 für den Pi 2 zu veröffentlichen – das gibt es nicht mal für den Intel Edison. Durch den Umstieg von ARMv6 auf dem Pi 1 zu ARMv7 auf dem Pi2 läuft auf dem Bastelboard auch Ubuntu – mit aktuelleren Paketen als bei den üblichen Debian-Distributionen. Allerdings ist etwa zur Installation der Version 14.04 noch ein wenig Fickelarbeit nötig.

Der Pi hat einen speziellen CSI-Port zum Anschluss der PiCam. Für den DSI-Port zur Ansteuerung von Displays gibt es bislang kein maßgeschneidertes Produkt.

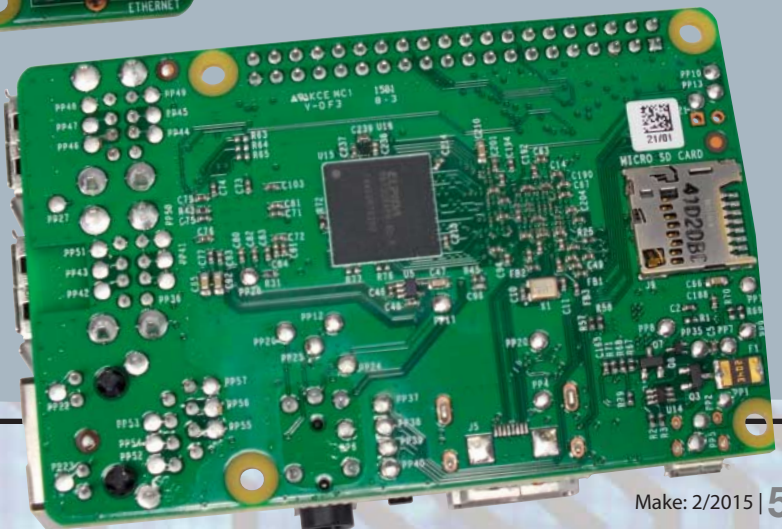
Raspberry Pi 2 B

Prozessor	ARM Cortex-A7 Quad-Core, 900 MHz
Grafik	VideoCore IV
RAM	1 GByte
OS-Speicher	MicroSD (UHS-1)
Video	HDMI
Audio	HDMI, Klinke
Netzwerk	10/100
SATA	–
USB-Host	4
USB-OTG	–
I/O	26
I/O-Spannung	3,3 V
5V-tolerant	–
PWM	1
I2C	2
UART	1
SPI	1
IR	–
ADC	–
CAN	–
Versorgung	USB, 5 V / 1 A
Webseite	www.raspberrypi.org
Preis	39 €



Der Raspberry Pi hat neben den I/O-Pins noch einen Anschluss für Kameras (CSI) und Displays (DSI).

Anders als beim Raspberry Pi 1 hat der Hersteller den Speicher nicht unter dem SoC, sondern auf der Unterseite der Platine verlötet.



TINKERFORGE

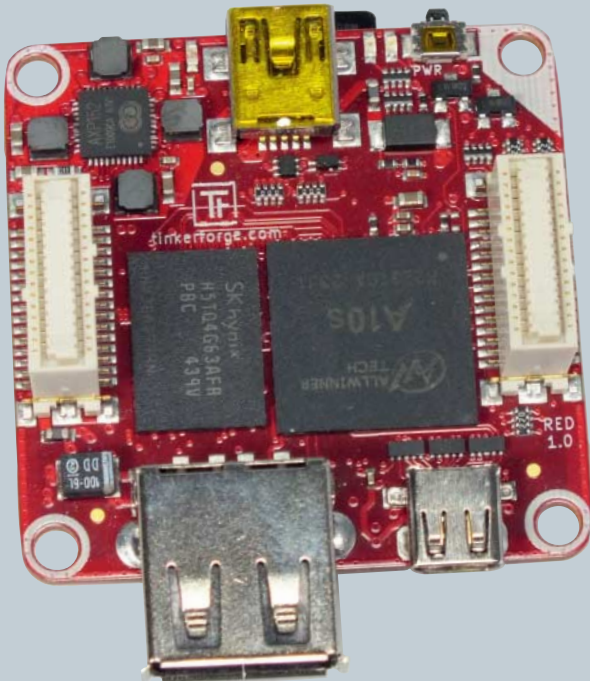
Das Baukastensystem lässt sich seit Dezember vergangenen Jahres abgekoppelt vom PC betreiben. Der RED Brick (RED steht für Rapid Embedded Development) arbeitet autark auf Grundlage eines speziellen Debian-Linux. Das nur 4 × 4 Quadratzentimeter große Platinchen weist keine eigenen I/O-Ports im eigentlichen Sinne auf. Vielmehr lassen sich auf zwei speziellen Anschlüssen Erweiterungsplatinen aufstecken. Diese sogenannten Bricklets lassen keine Robotik-Wünsche offen: IR- und Sonar-Abstandssensoren, GPS, Umweltsensoren, Analog-in- und Analog-out-Boards, Treiber für Schrittmotoren, DC-Motoren und Servos, Gyro- und Beschleunigungssensor, LC-Display, Knöpfe, Schieber und vieles mehr.

So kann man sich seinen eigenen Roboter maßgeschneidert zusammenstellen und jederzeit um weitere Funktionen ergänzen. Für die jeweiligen Boards und ihre Funktionen liefert der Hersteller Tinkerforge APIs für zahlreiche gängige Programmier- und Skript-Sprachen, sogar Delphi/Lazarus ist dabei. Obwohl ein Linux auf der Brick läuft, können auch reine Windows-Anwender damit umgehen. Mit der komfortablen Oberfläche des Brick-Viewer erledigt man die Konfiguration und Programmierung der RED Brick.

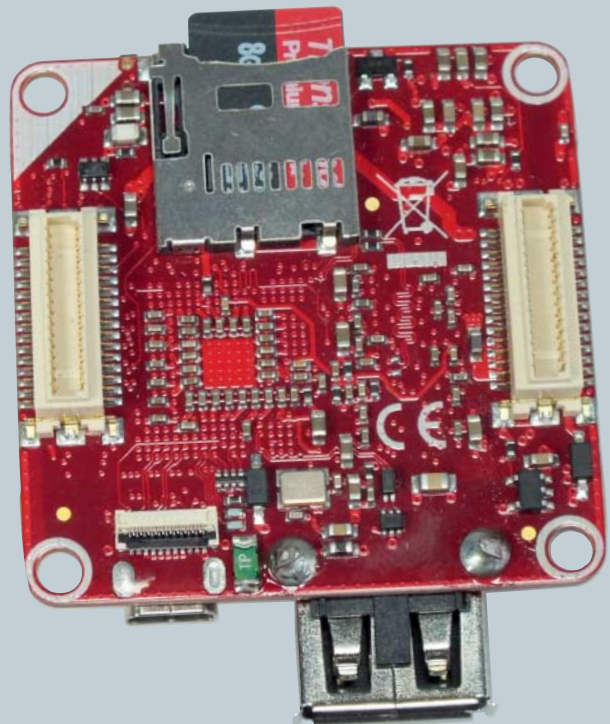
Linux-Anwender können einen HDMI-Monitor an die Brick anschließen und auf dem LXDE-Desktop arbeiten oder sich per SSH einloggen. Auf dem System sind diverse Software-Pakete wie Python, Qt, OpenCV und Mono vorinstalliert. Mit der etwas betagten OpenCV-Version 2.3.1. erreichte die RED Brick immerhin 5 bis 7 fps. —dab

Tinkerforge RED Brick

Prozessor	Allwinner A10 Cortex A8 1 GHz
Grafik	ARM Mali-400M P2 (OpenGL ES 2.0/1.1)
RAM	512 GByte DDR2
OS-Speicher	MicroSD (UHS-1)
Video	Micro-HDMI
Audio	HDMI
Netzwerk	-
SATA	-
USB-Host	1
USB-OTG	-
I/O	-
I/O-Spannung	3,3 V
5V-tolerant	-
PWM	-
I2C	-
UART	-
SPI	-
IR	-
ADC	-
CAN	-
Versorgung	USB, 5 V / 1 A
Webseite	www.tinkerforge.de
Preis	70 €



Ein vollständiger PC mit Micro-HDMI und USB



Die weißen Pfosten nehmen die Bricklets auf.

Anzeige



Bewegende Momente

Kurzinfo

Zeitaufwand:
2 Stunden

Kosten:
50 Euro

Programmieren:
Arduino

Schwierigkeitsgrad



Mit Bewegungs- und Abstands-Sensoren kann man sich nähernde Objekte erkennen. Wie man sie an einen Arduino anschließt und per Software auswertet, zeigt der folgende Artikel.

von Friederike Maier

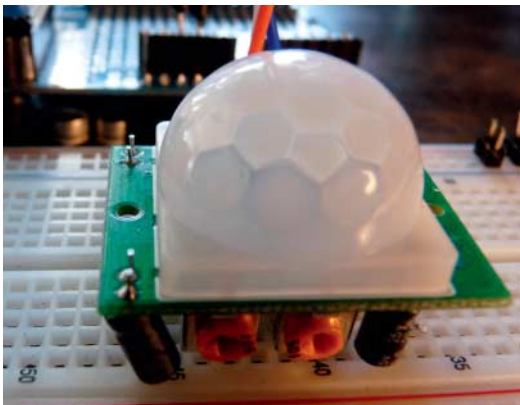
Einkaufsliste

- » Arduino Uno
- » PIR-Sensor HC-SR501
- » Ultraschallsensor HC-SR04
- » IR-Sensor Sharp GP2Y0A02YK0F
- » Breadboard + Jumper optional Servos
- » Schlaugaugen (eBay)

Jeder bessere Roboter sollte auf seine Umgebung reagieren können. Sei es, um nirgendwo dagegen zu stoßen oder um auf jemanden oder etwas zu reagieren, das vorbeikommt. Schon kleine Reaktionen auf die Umgebung erzielen eine große Wirkung. Der Roboter im Bild links blinzelt mit den Augen, wenn sich jemand in der Umgebung bewegt, und fängt an, den Kopf zu drehen. Wenn sich jemand nähert, dreht er den Kopf in Richtung der sich nähernden Person. Später soll er noch sprechen und drucken lernen.

Bewegt sich was?

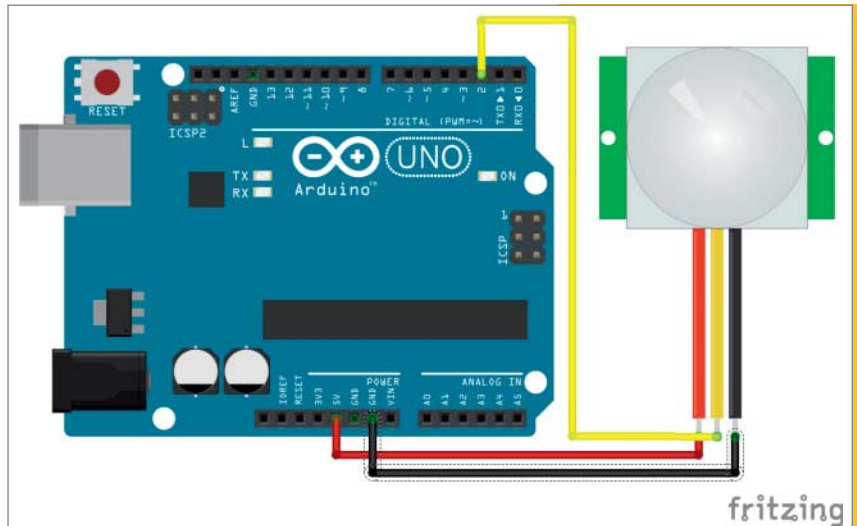
Bewegungs-Sensoren sind prima, wenn nur detektiert werden soll, ob sich etwas bewegt. Den genauen Abstand können sie nicht messen. Dafür sind sie schon für wenige Euro zu haben und einfach anzusteuern. Sie sind zum Beispiel massenhaft in automatischen Lichtschaltern verbaut. Die am häufigsten verwendeten PIR-Sensoren (englisch: passive infra-red) arbeiten rein passiv.



Die wabenähnliche Struktur des PIR-Bewegungsmelders teilt die Umgebung in verschiedene Abschnitte auf. Dadurch lassen sich Bewegungen besonders einfach detektieren.

Wenn eine Person an solch einem Bewegungssensor vorbeiläuft, strahlt sie Wärme im Infrarotbereich ab. Im eigentlichen Sensor sind piezoelektrische Elemente verbaut, die auf eine zeitliche Temperaturänderung mit Ladungstrennung reagieren. Das kann als Spannung an zwei Elektroden abgenommen werden. Vor dem Sensor ist eine milchig-weiße Sammellinse verbaut, die die Umgebung in verschiedene Abschnitte aufteilt. Dabei wechseln sich solche, aus denen die IR-Strahlung gebündelt empfangen wird, und solche, aus denen nicht empfangen wird, ab. Bewegt sich jemand durch diese Abschnitte, ändert sich die Temperatur am Sensor ständig, die nachgeschaltete Elektronik wertet dies aus und setzt den Ausgangspin auf „High“, was mit dem Arduino sehr einfach ausgewertet werden kann.

Da der Sensor rein passiv funktioniert, muss das, was detektiert werden soll, eine gewisse Wärme ausstrahlen. Ein interessantes Experiment ist es, sich in



Die Verkabelung des PIR-Sensors

Rettenfolie zu wickeln und dann an einem PIR-Sensor vorbeizugehen.

Über den linken der zwei orangenen Potis im Bild links lässt sich die Dauer, die der Ausgangspin auf „High“ gesetzt wird, im Uhrzeigersinn von 5 Sekunden bis 300 Sekunden einstellen. Mit einem Jumper auf der Rückseite kann noch eingestellt werden, ob während dieser Zeit weiter detektiert wird, und wenn sich noch was bewegt, der Pin auf „High“ bleibt, oder ob er nach dieser Zeit erst mal wieder auf „Low“ geschaltet wird (das ist die Standardeinstellung). Das rechte Poti verändert im Uhrzeigersinn die Distanz, bis zu der Bewegungen detektiert werden, von 3 bis 7 Metern.

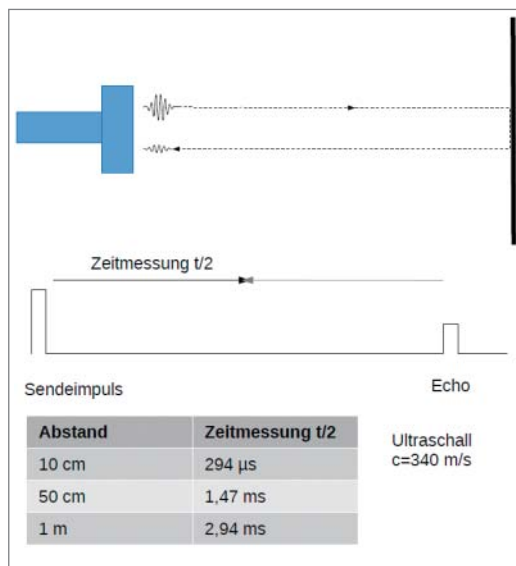
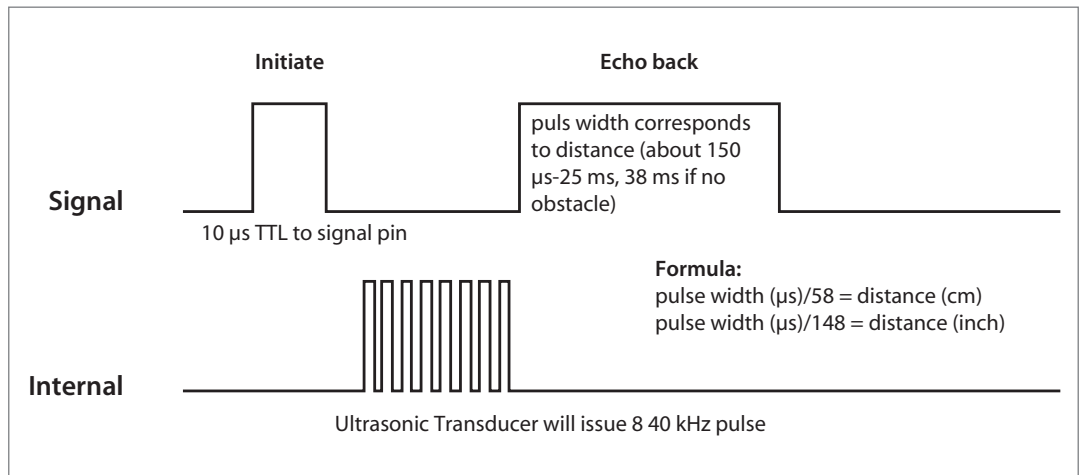
In Listing 1 ist ein Sketch zu finden, mit dem ein Arduino die Zustände eines HC-SR501-PIR-Sensors auswertet.

Der Sketch zeigt mit der auf dem Arduino verbauten LED den Zustand des Detektors an. Zudem kann man auf der seriellen Schnittstelle den Zustand sehen.

Listing 1

```
const int ledPin = 13;
const int inputPin = 2; // Anschluss für PIR
int ledStatus = LOW;
int val;
void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(inputPin, INPUT);
  Serial.begin(9600);
}
void loop(){
  val = digitalRead(inputPin);
  if (val == HIGH) {
    digitalWrite(ledPin, HIGH);
    Serial.println("XXXXX");
  }
  else {
    digitalWrite(ledPin, LOW);
    Serial.println("00000");
  }
  delay(200) ;
}
```

Das Timing-Diagramm des Herstellers zur Ansteuerung des Sensors



Das Echo-Laufzeitenverfahren zur Abstandsmessung

Listing 2

```

const int triggerPin = 8;
const int echoPin = 9;
long zeit, abstand;
int maxAbstand = 400;
void setup() {
  Serial.begin(9600);
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
void loop() {
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  zeit = pulseIn(echoPin, HIGH);
  abstand = zeit/58.3;
  if (abstand <= maxAbstand){
    Serial.println(abstand);
  }
  delay(100);
}
    
```

Abstand halten

Wenn der Abstand eines Objekts gemessen werden soll, werden meist entweder Infrarot- oder Ultraschall-Sensoren verwendet. Daneben gibt es noch Laser-Sensoren, die sind allerdings nicht mehr für ein paar Euro erhältlich und werden hier nicht weiter behandelt.

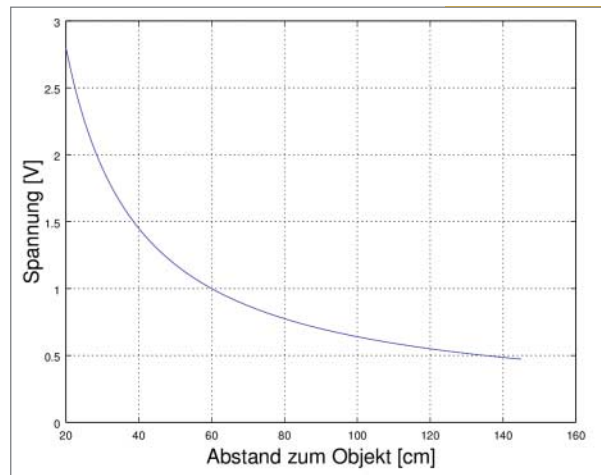
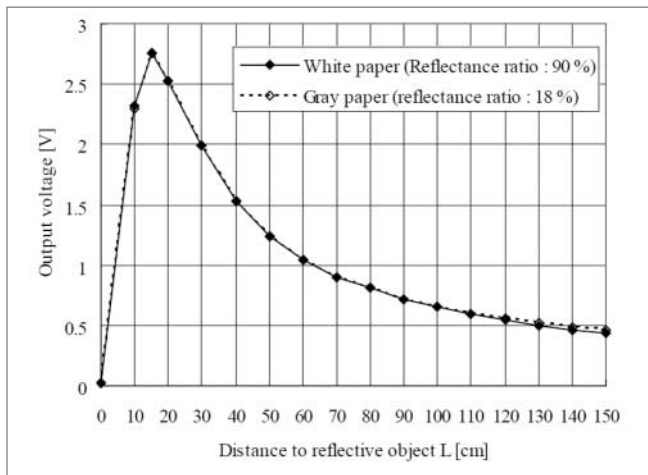
Als Ultraschall werden Schallwellen oberhalb von 20 kHz also oberhalb des hörbaren Bereichs bezeichnet. Die meisten Ultraschall-Sensoren arbeiten bei einer Frequenz von 40 kHz. Die Ultraschall-Abstands-Sensoren senden aktiv ein Signal aus und messen die Laufzeit t , mit der das an einem Objekt reflektierte Signal zurückkommt. Aus dieser Zeit, der Echo-Laufzeit, lässt sich der Abstand d berechnen zu:

$$d = t \cdot c / 2$$

Der Wert c ist hier die Ausbreitungsgeschwindigkeit der Schallwelle. Die Schallgeschwindigkeit in trockener Luft bei 20 °C beträgt rund 343 Meter pro Sekunde. Die Abhängigkeit der Schallgeschwindigkeit von der Temperatur wird hier vernachlässigt und ein Näherungswert von 340 m/s verwendet.

Den Ultraschallsensor HC-SR04 gibt es für wenige Euro zu kaufen und er soll laut Datenblatt Entfernungen zwischen 2 und 400 cm messen. Es gibt jede Menge ähnliche Sensoren, die nach dem gleichen Prinzip funktionieren, beispielsweise von Parallax. Manche haben nur 3 Pins, dann liegen Trigger- und Echosignal auf demselben Pin. Der HC-SR04 hat vier Pins; zwei für die Stromversorgung, einen für Trigger und einen für den Echo-Puls. Da sich die Laufzeit des Ultraschalls in einem Bereich von einigen Millisekunden bewegt, ist sie für den Arduino noch direkt messbar. Das Anschlusskabel kann man leicht mit einer Präzisions-Buchsenleiste und etwas Litze zusammenschweißen oder man packt ihn auf eine Steckplatine. Im Datenblatt des Herstellers ist die Ansteuerung direkt beschrieben und kann in einen Arduino-Sketch einfach umgesetzt werden. Listing 2 zeigt ein Beispiel.

Der Trigger an Pin 8 wird von Low auf High und wieder auf Low gesetzt, um acht Ultraschallimpulse aussenden zu lassen. Mit dem Befehl `pulseIn(echoPin,`



HIGH) misst der Arduino ab Start des Befehls die Zeit, bis am echoPin ein High-Impuls ankommt. Der Abstand wird aus der Zeit berechnet und über die serielle Schnittstelle ausgegeben.

Der Winkel des Lichts

Infrarot-Sensoren (IR) arbeiten mit elektromagnetischen Wellen, die sich mit Lichtgeschwindigkeit (circa 300 000 km/s) ausbreiten. Die Frequenzen

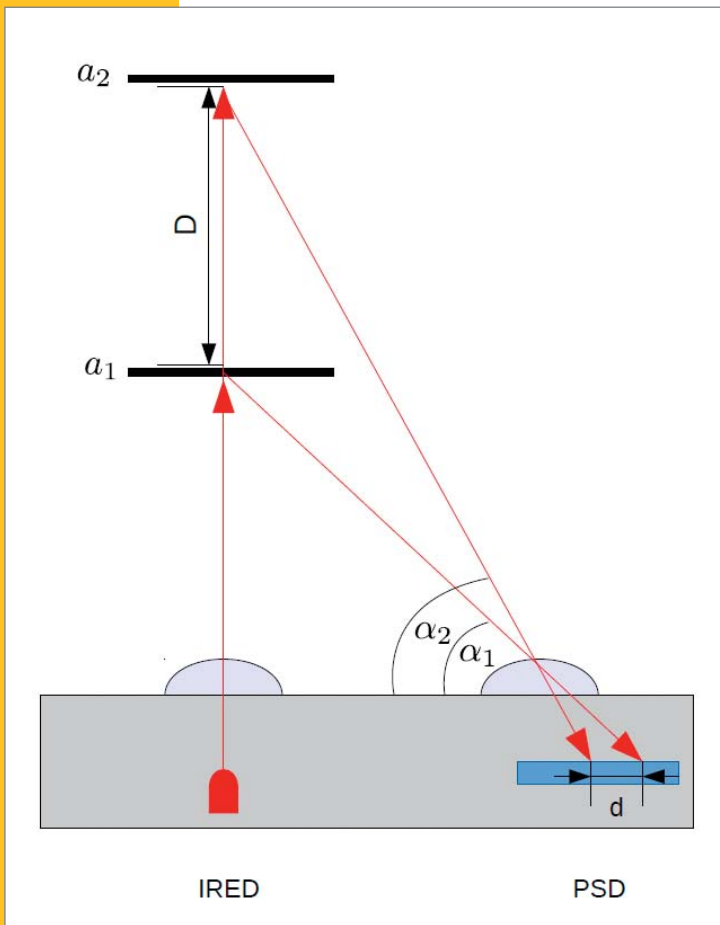
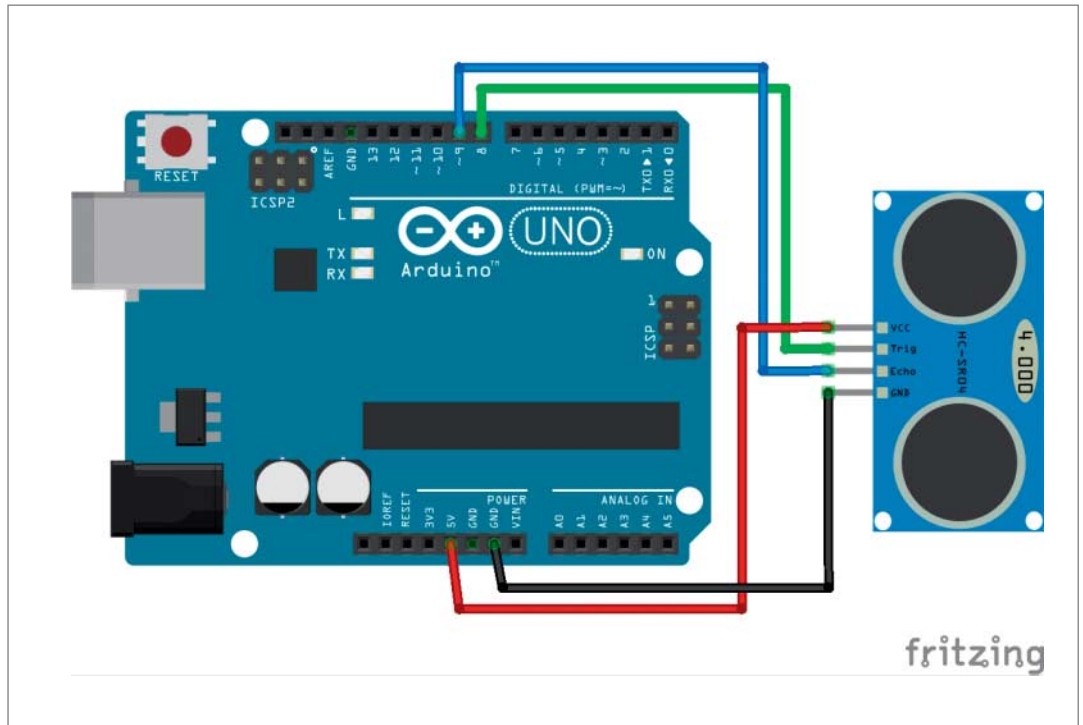
des Infrarot-Bereichs liegen unterhalb des sichtbaren Lichts. Da die Echozeiten bei Infrarot im Bereich von Nanosekunden liegen, wird bei vielen IR-Sensoren ein anderes Verfahren zum Messen des Abstands verwendet als bei den Ultraschall-Sensoren.

Bei der Triangulationsmethode wird der Zusammenhang zwischen den Winkeln in einem Dreieck und den Kantenlängen ausgenutzt. Das reflektierte Signal einer Infrarotdiode (IRED) trifft im Detektor je

Links das im Datenblatt angegebene Verhältnis zwischen Spannung und Abstand. Rechts die angenäherte Kurve: $d[\text{cm}] = 54 / (\text{Spannung}[\text{V}] - 0,1)$ ergibt eine gute Näherung an die Originalkurve.

Anzeige

Für die Verkabelung des Ultraschall-Sensors genügen vier Jumper mit Male-Female-Anschlüssen.

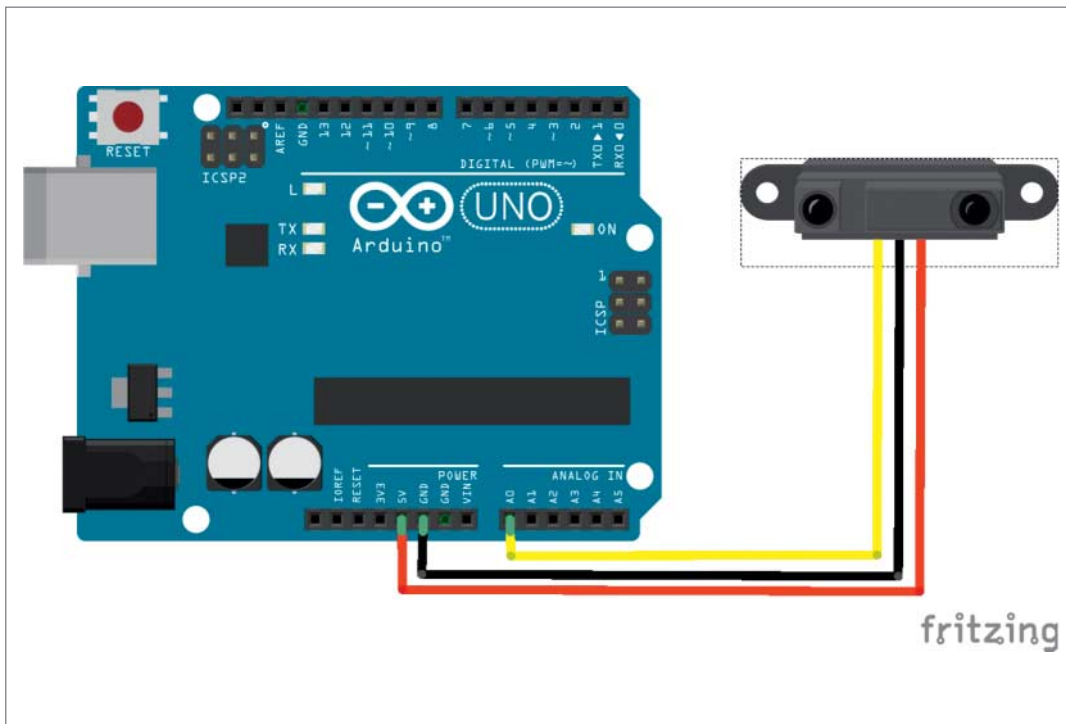


Der IR-Sensor nutzt die Triangulationsmethode, um den Abstand zu Objekten zu messen.

nach Entfernung des Objekts mit einem anderen Winkel auf (siehe Bild rechts unten). Und je nach Winkel trifft das Signal an einer anderen Stelle auf den PSD (position sensitive detector). Der PSD wandelt die Position in eine Spannung um.

Die Klassiker unter dieser Art von Sensoren ist die GP2Y-Serie von Sharp. Der GP2Y0A02YK0F-Infrarotsensor kostet schon ein paar Euro mehr als der Ultraschallsensor, aus Asien gibt es ihn inzwischen auch schon für um die 8 Euro. Bei der Bestellung sollte man darauf achten, dass das Anschlusskabel mit dreipoligem Stecker bereits dabei ist (falls nicht, das ist ein „3 Pin JST“ [Japan Solderless Terminal]-Stecker). Der GP2Y sendet im Infrarotbereich mit einer Wellenlänge von ungefähr 850 nm. Laut Datenblatt soll er Entfernung in einem Bereich von 20 bis 150 cm messen können. Die Spannung, die der Sensor ausgibt, hängt nicht linear vom Abstand ab. Die im Datenblatt angegebene Kurve kann aber im relevanten Bereich über eine einfache Formel angenähert werden (siehe Bild oben).

Die Spannung lässt sich leicht über einen analogen Eingang des Arduino einlesen. Der A/D-Wandler hat eine Auflösung von 10 Bit, was Werten zwischen 0 und 1023 entspricht, also 0 bis 5 Volt. Bei allen Objekten unter einem Abstand von 20 cm wird ein falsches Messergebnis ausgegeben, dort sind die Spannungswerte nicht mehr eindeutig. Gerade bei sich bewegenden Objekten, bei denen der Sensor dazu verwendet wird, Kollisionen zu vermeiden, muss man daher darauf achten, dass dieser so montiert ist, dass die Minimalentfernung gar nicht auftreten kann. Die Sensordaten suggerieren nämlich, dass sich das Objekt wieder entfernt, wenn es zu nahe kommt. Kollisionen sind so vorprogrammiert. Andere



Die Verkabelung des Sharp IR-Sensors

Modelle von Sharp gibt es für Entfernungen von 10 bis 80 cm und von 2 bis 10 cm. Man kann die Sensoren auch miteinander kombinieren.

Listing 3 zeigt, wie die Abstandsmessung mit dem Sharp GP2Y0A02YK0F funktioniert.

Die Schleife liest regelmäßig den Wert vom A/D-Wandler ein. Ist der Abstandswert unter 100 (für weniger als 135 cm), so berechnet das Programm zunächst die zum Digitalwert passende Spannung und daraus wiederum mit der Umrechnungsformel den Abstand.

Bei horizontaler Bewegung von Objekten und Objekten, die sich eher in vertikaler Richtung verändern, also zum Beispiel stehenden Menschen, sollte der IR-Sensor vertikal verbaut werden.

Infrarot oder Schall?

Ein Vergleich der beiden Sensoren zeigt, dass der IR-Sensor die Abstände deutlich genauer misst. Auch direkte Sonneneinstrahlung hat den IR-Sensor in meinen Tests ziemlich unbeeindruckt gelassen, ebenso heiße Gegenstände in der Umgebung, die ja ebenfalls infrarote Strahlung abgeben. Bei dem getesteten Sensor ließ die Genauigkeit allerdings ab einer Entfernung von 120 cm stark nach und bei Abständen unter 20 cm sind die Ergebnisse, wie oben beschrieben, nicht mehr eindeutig.

Der Ultraschallsensor wird bei grobem oder flauschigem Gewebe schnell ungenau. Entfernungen bis 0,5 m klappen noch ganz gut, weiter entfernte Fliesen und insbesondere grobe Wollpullover waren praktisch unsichtbar.

Der Winkel, bis zu dem Objekte wahrgenommen werden, ist beim IR-Sensor sehr eng. Schon ab circa

5° Abweichung wird nichts mehr detektiert. Der Detektionsbereich des US-Sensors ist mit 10° etwas breiter. Getestet wurde beide Male bei einem Abstand von 0,5 m. Mit dünnen Gegenständen, wie die etwa 1 cm dünne Breitseite eines Metermaßes hatten beide Sensoren keine Probleme.

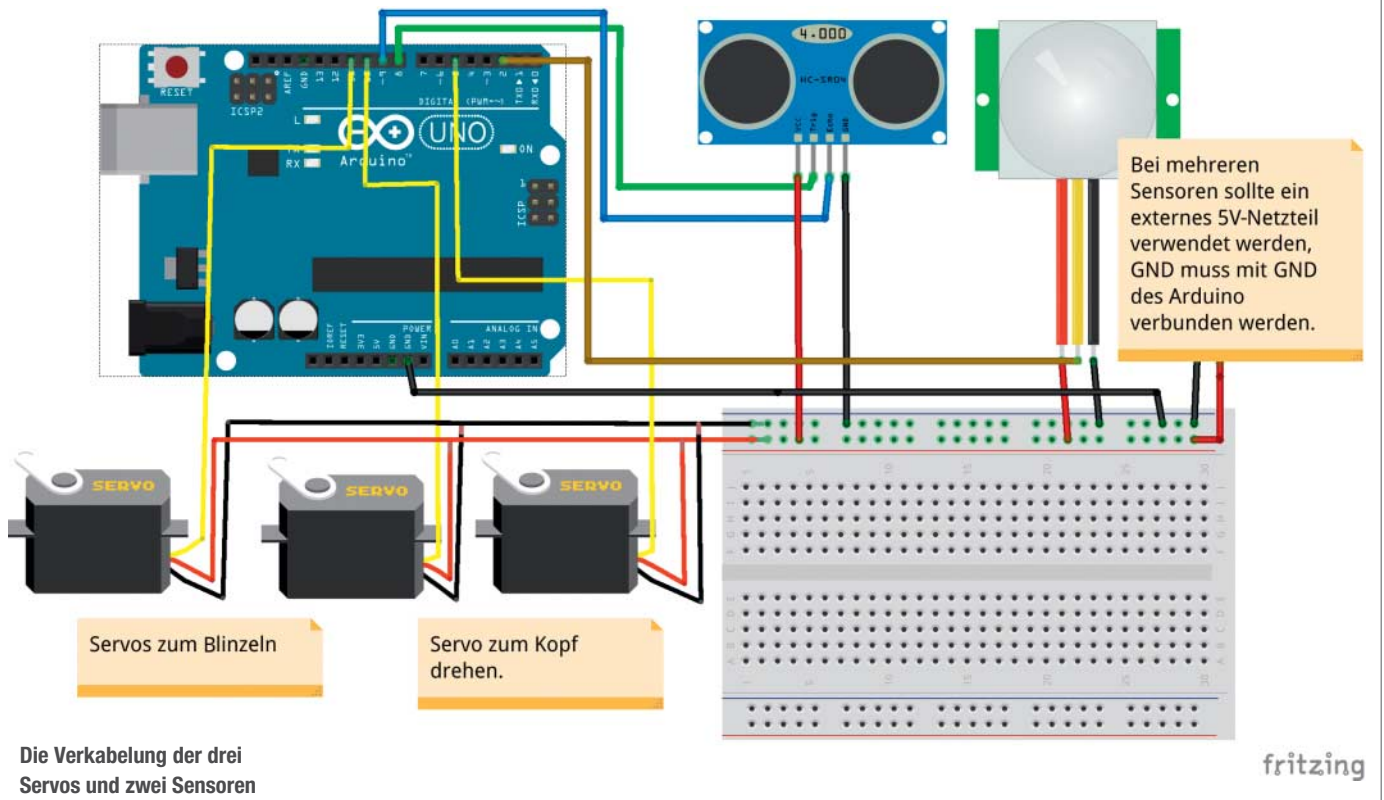
Der Augenklimper-Roboter

Die Sensor-Wahl für den Augenklimperer fiel auf die Ultraschall-Sensoren, da diese günstig im 5er-Pack ersteigert werden konnten, und der Roboter mehrere Sensoren bekommen soll, um auch mal auf mehrere sich nähernde Personen reagieren zu können

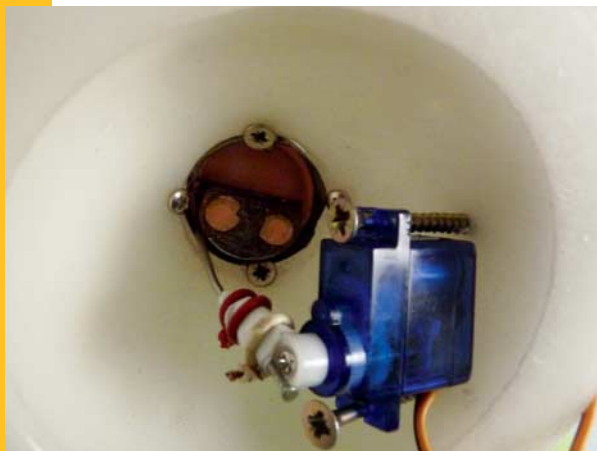
Listing 3

```
#define sensorPin A0
int threshold = 100;
int sensorOutput = 0;
double sensorSpannung = 0;
int abstand = 0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  sensorOutput = analogRead(sensorPin);
  if (sensorOutput >= threshold) {
    sensorSpannung = (double)sensorOutput * 0.0049;
    abstand = 54/(sensorSpannung-0.1);
    Serial.println(abstand);
  }
  delay(200);
}
```

SENSOREN



Das Gewicht in den Schlaftaugen ist so verteilt, dass sich die Augen in waagerechter Position (beim Schlafen) schließen. Die Schwerkraft wird hier durch einen von einem Servo angetriebenen Nagel ersetzt. Das Ganze ist in eine weiße Plastikhalbkugel eingebaut.



(die Augen lassen sich unabhängig voneinander drehen!). Die Genauigkeit der Abstandsmessung ist für diese Anwendung nicht so wichtig. Ein etwas weiterer Winkel, aus dem detektiert wird, ist dagegen gut, damit der Roboter weniger tote Winkel hat. Wollpullover-Träger werden zumindest durch den PIR-Sensor wahrgenommen.

Das Beispielprogramm zeigt (siehe Link unten) wie man zwei Sensoren und drei Servos einigermaßen parallel laufen lassen kann. Der Grundtakt von 15 ms wurde gewählt, da die Servos damit gut angesteuert werden können. Mithilfe der Funktion `millis()` wird die Programmlaufzeit ständig abgefragt. Alle Funktionen, die nicht alle 15 ms ausgeführt werden sollen, bekommen einen Zähler, der so weit inkrementiert wird, bis die Funktion wieder ausgeführt werden soll. Zu Beginn der Funktion werden dann die Programmlaufzeit und der Zähler verglichen und die Funktion nur ausgeführt, wenn die Laufzeit den Zähler schon erreicht hat.

Der PIR-Bewegungssensor weckt den Roboter auf, solange sich in einer Umgebung von ~7 m etwas bewegt. Dann fängt er an, mit den Augen zu klimpern und den Kopf zu drehen. Währenddessen analysiert der Ultraschallsensor, ob jemand näher herankommt. Wenn jemand auf weniger als 1 Meter herankommt, wendet er den Kopf in diese Richtung. —dab

↓ Links und Foren
make-magazin.de/xvz6

Anzeige

Im Angesicht des Roboters

Mit wenigen Befehlen bringt man einen PC mit einer Webcam dazu, menschliche Gesichter in Bildern zu finden. Ein Roboter kann auf diese Weise erkennen, dass ein Mensch vor ihm steht und beispielsweise eine Interaktion anfangen.

von Daniel Bachfeld

Kurzinfo



Zeitaufwand:
eine Stunde

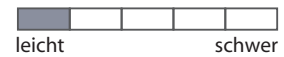


Kosten:
20 Euro



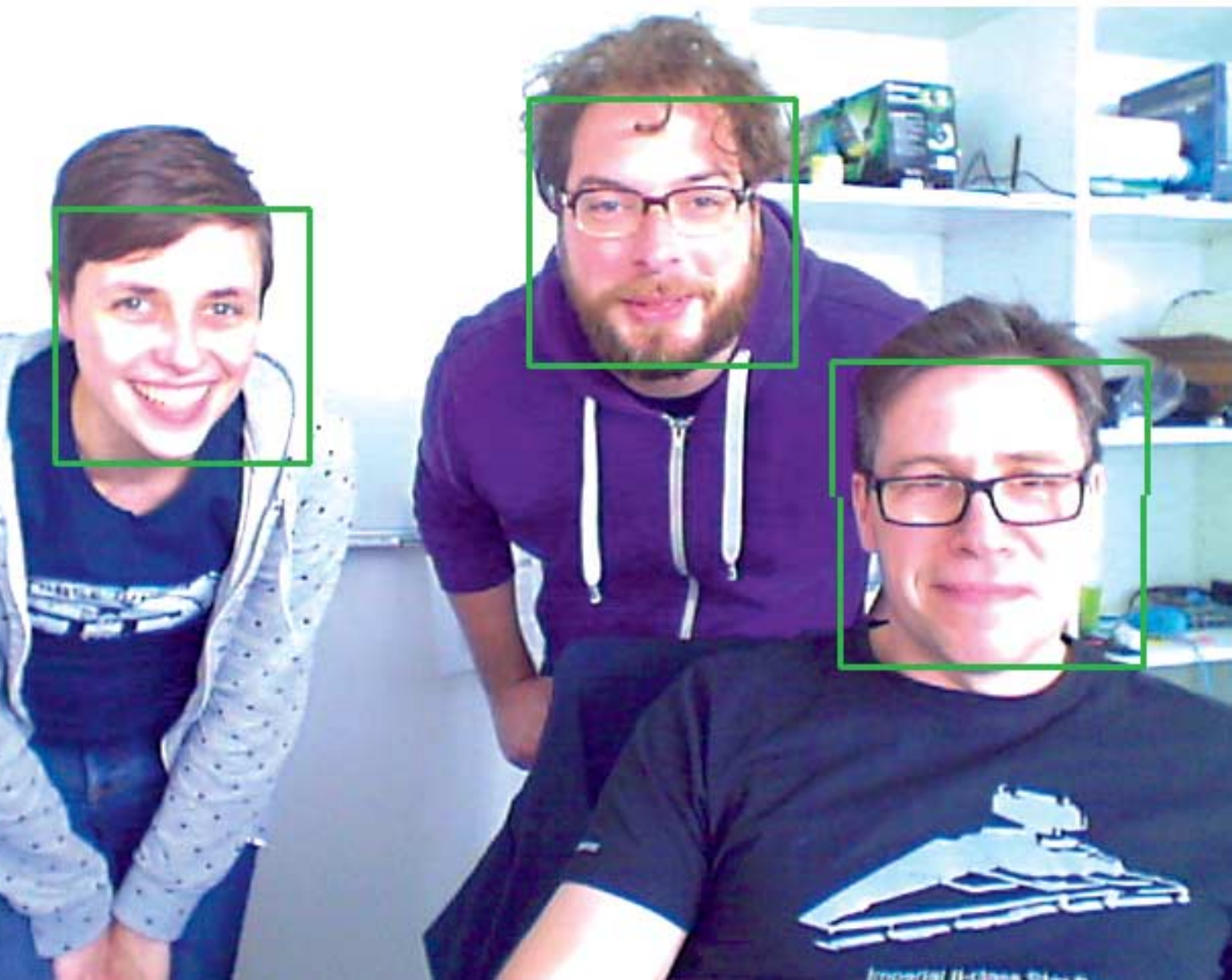
Programmieren:
Python

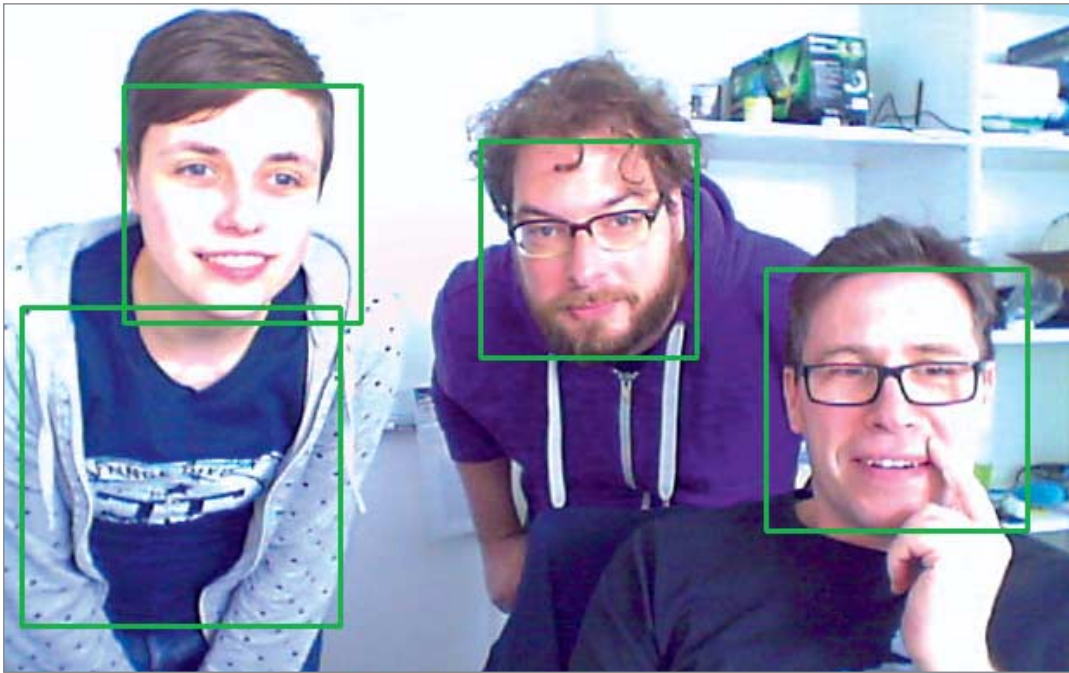
Schwierigkeitsgrad



Einkaufsliste

- » UVC-kompatible Webcam
- » Linux-PC oder -Mikrocontroller-Board





OpenCV hat beim Autor trotz abdeckender Hand das Gesicht erkannt, links allerdings ein Gesicht zu viel gesehen.

Auf Seite 38 haben wir bereits ein kleines Projekt mit Python und OpenCV vorgestellt, bei dem ein farbiger Ball im Mittelpunkt des Interesses steht. Doch OpenCV kann noch viel mehr als Objekte entdecken. In einem weiteren Schritt kann man mit der Bibliothek Gesichter erkennen. Gefundene Gesichter lassen sich in einem weiteren Schritt sogar Personen zuordnen. So kann der PC oder der Roboter die Person etwa mit ihrem Vornamen ansprechen. Allerdings ist dafür noch etwas Training mit Bildmaterial nötig, was den Rahmen dieses Artikels sprengen würde.

Für die Gesichtserkennung stellt OpenCV mächtige Bibliotheken bereit, die dem Entwickler praktischerweise fast die ganze Arbeit abnehmen. Dahinter steckt reichlich akademische Forschung und eine ziemlich große Portion höherer Mathematik. Die muss man nicht unbedingt kennen; es reicht, sich einige Punkte zu merken. Die Gesichtserkennung beruht auf der Merkmalsextraktion, hier den Helligkeitsunterschieden im Gesicht. Der Augenbereich ist in einem Gesicht üblicherweise dunkler als der Wangenbereich und die Augen sind meist dunkler als die Nase.

In Graubildern teilt man die Bereiche eines Bildes in Rechtecke auf und summiert darin die einzelnen Pixel der Region auf. So entstehen bestimmte Hell-Dunkel-Muster. Vergleicht man die erzeugten Muster mit den Mustern einer Gesichtsdatenbank (Haar-like-Features, Merkmale), kann man feststellen, ob sich im Bild ein Gesicht befindet. Der Begriff Haar-like bezieht sich auf die Funktion „Haar Wavelet“ (von Alfréd Haar), die in der Signalverarbeitung zur Kompression verwendet wird.

Neben der Suche nach „Haar-like Features“ in Bildern unterstützt OpenCV die Suche nach „Local Binary Patterns“ (LBP). Dabei wird ein Bild ebenfalls in Regionen eingeteilt. Statt aber alle Pixel für eine Region aufzusummieren, erzeugt man durch mathematische Funktionen eines Pixels mit seinen Pixelnachbarn leicht zu verarbeitende binäre Muster. Da LBPs nur auf

Integer-Operationen beruhen, findet man damit schneller ein Gesicht als mit Haar-like Features.

In diesem Zusammenhang kommt das Konzept der „Cascade of Classifiers“ ins Spiel. Anstatt jedes Feature einer Datenbank (mit beispielsweise 6000 Einträgen) mit jeder Region im Bild zu vergleichen, werden aus Geschwindigkeitsgründen nur solche Regionen näher untersucht, bei denen bereits das erste Feature detektiert wurde. Sucht man also vereinfacht gesagt im ersten Durchlauf nach Augen-Nasen-Features, werden alle weiteren Features auch nur in diesen erkannten Regionen durchprobiert. Jede Region, die nicht zum Feature im ersten Durchlauf passt, wird in allen weiteren Schritten ignoriert.

Die Cascading Classifier funktionieren nur für eine bestimmte Blickrichtung auf ein Gesicht, beispielsweise die Frontalansicht. Für seitlich gedrehte Köpfe benötigt man eine neue antrainierte Datenbank. Die muss ein Skript gegebenenfalls in einem weiteren Lauf durchprobieren.

Praxis

Unser Python-Skript (siehe folgende Seite) nutzt ausgiebig OpenCV-Funktionen. Praktischerweise ist unter Raspbian OpenCV bereits vorinstalliert. Um die Funktionen von Python aus nutzen zu können, benötigt man einen Wrapper, den man mit `sudo apt-get install python-opencv` nachinstalliert – je nach Distribution wird zur Auflösung der Abhängigkeiten ein eventuell fehlendes OpenCV-Paket gleich mit installiert.

Zeile 1 bis 3 des Skripts importieren verschiedene Bibliotheken. Zeile 5 liest den Pfad zur Datenbank, in der die Features abgelegt sind. Das kann einmal `lbpcascade_frontalface.xml` oder `haarcascade_frontalface_default.xml` sein. Mit Zeile 7 legt man diese Datenbank als den Cascading classifier fest.

Die Webcam wird mit Zeile 9 als Bildgerät initialisiert und die Bildgröße auf `800 × 600` gesetzt. Die

GESICHTSERKENNUNG

Der Gesichtserkennung ist es egal, ob sein Träger lebt oder nur das Cover der Make zielt.



```
1 import cv2
2 import sys
3 import time
4
5 cascPath = sys.argv[1]
6
7 faceCascade = cv2.CascadeClassifier(cascPath)
8
9 video_capture = cv2.VideoCapture(0)
10 video_capture.set(3,800)
11 video_capture.set(4,600)
12 while True:
13     start=time.time()
14
15     ret, frame = video_capture.read()
16
17     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
18
19     faces = faceCascade.detectMultiScale(
20         gray,
21         scaleFactor=1.1,
22         minNeighbors=5,
23         minSize=(120, 120),
24         flags=cv2.cv.CV_HAAR_SCALE_IMAGE
25     )
26
27
28     for (x, y, w, h) in faces:
29         cv2.rectangle(frame, (x, y), (x+w, y+h), (
30             0, 255, 0), 2)
31
32
33     cv2.imshow('Video', frame)
34     end=time.time()
35     fps= end -start
36     print 1/fps
37     if cv2.waitKey(1) & 0xFF == ord('q'):
38         break
39
40
41 video_capture.release()
42 cv2.destroyAllWindows()
```

While-Schleife läuft unendlich, bis man die q-Taste drückt. Die Schleife speichert zunächst die Zeit und liest einen Frame aus der Kamera. Zeile 18 konvertiert das Farbbild in Grauwerte.

Der Aufruf von Zeile 20 bis 26 sucht im Bild nach Gesichtern. Der Parameter `scaleFactor` legt fest, um welchen Faktor die Funktion das Bild in seiner Größe verändern darf. Mit `minNeighbors` definiert man, wie viele detektierte „Treffer“-Kandidaten ein Gesicht haben muss, damit es als gesichert erkannt gilt. Die Angabe `minSize` legt die Mindestgröße eines Gesichts fest, ab der es erkannt wird. Kleinere Werte erlauben größere Entfernungen zur Kamera, bedeuten allerdings einen erheblich längeren Suchdurchlauf pro Bild. Bei größeren Werten muss man näher an der Kamera sein, dafür ist die Funktion schneller fertig.

Das Flag `CV_HAAR_SCALE_IMAGE` teilt der Funktion mit, dass bei Durchläufen das Bild und nicht die Features skaliert werden sollen. Laut Dokumentation ist das speicherschonender. Die Funktion liefert die Positionen und Dimensionen eines oder mehrerer Rechtecke zurück. Die For-Schleife in Zeile 29 durchläuft diese Liste und zeichnet die passenden Rechtecke in den Frame ein. Zeile 33 bringt das fertige Bild auf den Desktop. Die Zeilen 34 bis 36 berechnen, wie oft das pro Sekunde geschieht und geben das Ergebnis als Frames per Second aus. Bei Abbruch der Schleife schließt Zeile 41 das Video-Gerät und Zeile 42 alle Fenster.

Das ganze Skript startet man mit `python face.py lbpcascade_frontalface.xml` respektive `python face.py haar-cascade_frontalface_default.xml`. —dab



Links und Foren
make-magazin.de/x6md

Anzeige

Auch Roboter können schwärmen

Mit wenigen Zeilen Code kann man selbst leistungsschwachen Robotern wie Lego Spybotics kooperatives Verhalten beibringen. Damit bewegen sie sich auf faszinierende Weise als Kolonne oder als Schwarm durch Räume.

von Sonja und Niklas Braband





Sonja (14) und Niklas (13) Braband haben mit ihren Experimenten zu Roboterschwärmen 2014 den 2. Platz beim Landeswettbewerb „Schüler experimentieren“ belegt.



Viele Tierarten, insbesondere Ameisen, Bienen, Fische oder Vögel zeigen sogenanntes Schwarmverhalten. Gerade kleine Tiere koordinieren etwa die Futtersuche oder Fortbewegung so, als ob sie ein einziges großes Lebewesen wären. Das wird häufig auch Schwarmintelligenz genannt. Dabei ist von außen schwer wahrzunehmen, wie sich die Tiere oder Insekten abstimmen. Forscher versuchen seit längerem, winzige Roboter zu bauen und diesen Schwarmverhalten einzuprogrammieren. Wir wollten das ebenfalls probieren. Zuhause hatten wir schon mit Lego Roboter gebaut und einfache Programme dafür geschrieben, beispielsweise für den RCX. Daher haben wir uns gefragt, ob wir auch unseren Robotern einfaches Schwarmverhalten beibringen können.

Zunächst haben wir uns überlegt, welche Arten von Schwarmverhalten es gibt, und wie wir es auf Roboter übertragen können. Dabei muss man berücksichtigen, dass unsere Roboter ihre Umgebung nur grob wahrnehmen können, zum Beispiel über Infrarot-Sender und -Empfänger. Wir wollten versuchen, Robotern folgende Verhalten beizubringen: 1. Suchen einer Futterquelle oder eines Artgenossen, 2. Führen von Artgenossen zu einer Futterquelle (Ameisen-Kolonie), 3. spontanes Bilden eines Schwarmes und 4. ein Ausweichmanöver, wenn ein Feind kommt.

Tierisch

Wir haben zur Vorbereitung einen Tag lang Fischschulen in einem Aquarium aus der Nähe beobachtet. Dabei haben wir auch Videos aufgenommen und gemerkt, dass Fische nicht exakt und perfekt nebeneinander schwimmen, sondern sich immer wieder neu anpassen und ausrichten. „Schwarmverhalten wird zu Schwarmintelligenz, wenn eine Gruppe eine Aufgabe im Kollektiv löst, die kein Angehöriger der Gruppe alleine lösen könnte“, schreibt Len Fisher in seinem Buch „Schwarmintelligenz“. Vor etwa 30 Jahren haben Forscher in aufwendigen Experimenten an Fischschwärmen herausgefunden, dass Schwärme nach einfachen Regeln funktionieren, die jedes Mitglied befolgt: a) Folge dem Fisch vor dir (wenn es einen gibt) und b) halte die Geschwindigkeit des Fisches neben dir.

Später wurden diese Regeln noch etwas verallgemeinert: 1. Halte genügend Abstand von anderen, 2. Bewege dich in die durchschnittliche Richtung deiner nächsten Nachbarn und 3. Bewege dich auf die durchschnittliche Position deiner nächsten Nachbarn zu. Dies bedeutet, dass alle Mitglieder sich gleich verhalten – und die Roboter alle gleich programmiert sind.

Eine konkrete Umsetzung dieser Regeln sieht man manchmal bei Bildschirmschonern für PCs oder in Zeichentrickfilmen, in denen Tierschwärme realistisch nachgeahmt werden sollen. Es funktioniert deshalb gut, weil alles auf einem schnellen Rechner simuliert wird. Der weiß immer exakt, wo sich jedes Mitglied befindet und mit welcher Geschwindigkeit es sich in welche Richtung bewegt.

BEZUGS-QUELLE

Auf eBay findet man zwar gebrauchte Spybotics-Module, allerdings schwankt die Zahl der Angebote sehr stark. Auf der Händlerseite www.bricklink.com findet man erheblich mehr angebotene Module für teilweise unter 10 Euro (Suchbegriff „Spybotics Module“). Dort bekommt man auch das zum Programmieren notwendige Kabel (Suchbegriff „Electric, Serial to Light Sensor Cable“).

ONLINE

Weitere Infos zu den Experimenten, Vorüberlegungen, Beobachtungen von Sonja und Niklas Braband, Videos und Listings finden Sie unter dem Link am Ende des Artikels.

Kurzinfo

Zeitaufwand:
ein Wochenende

Kosten:
100 Euro

Programmieren:
Brick, Grundkenntnisse C

Schwierigkeitsgrad

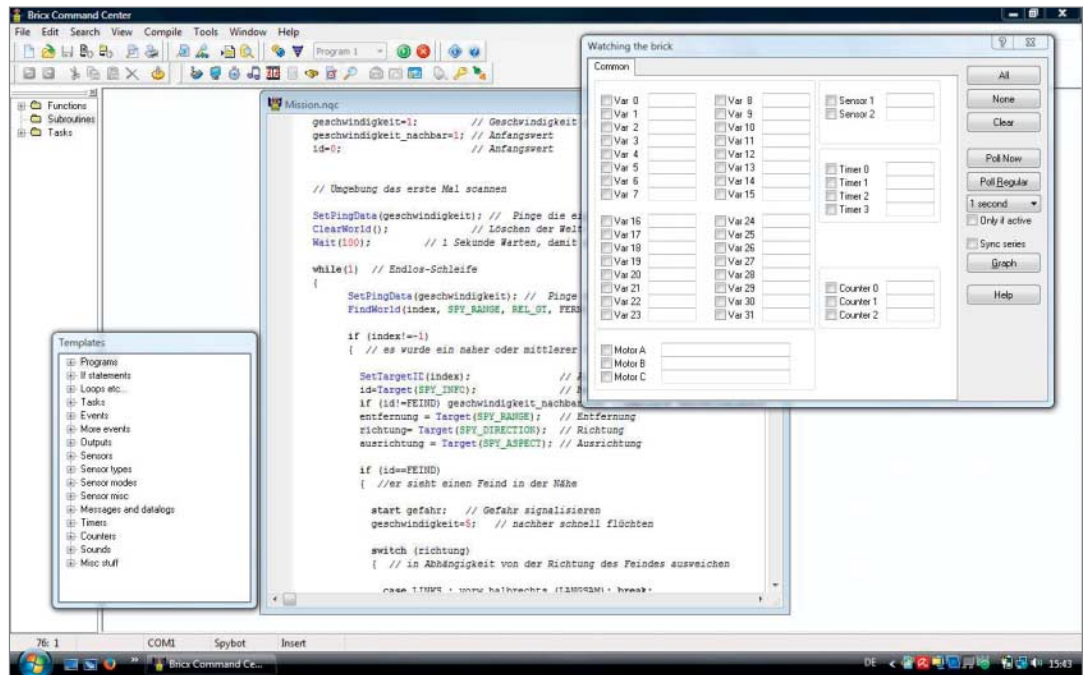


Einkaufsliste

- » 3–5 Lego-Spybotics-Roboter
- » Lego-Technik-Bausteine
- » Räder
- » Achsen
- » Programmierkabel für Lego Spybotics

ROBOTER PROGRAMMIEREN

Die grafische Oberfläche BricxCC erleichtert das Programmieren und die Fehlersuche.



Praxis

Für unsere Schwarm-Umsetzung mit echten Robotern haben wir die uns zur Verfügung stehenden Möglichkeiten überdacht. Der Lego EV3 war noch nicht rausgekommen, als wir angefangen haben. Er ist wie unser normaler Lego Mindstorms RCX zu teuer – insbesondere wenn man mehrere davon benötigt. Die günstigen Lego Scouts sind zu einfach, aber die Lego Spybotics kamen in Frage: Sie sind gebraucht günstig zu bekommen und sie haben noch ausreichend Rechenleistung.

Sie ähneln kleinen Autos und haben drei Infrarot-sender und zwei Infrarotempfänger. Über die Sender schicken sie mehrmals pro Sekunde kurze Infrarot-nachrichten (Pings) in alle Richtungen. Die Empfänger sitzen vorne am Sphbot, etwa dort, wo beim Auto die Frontscheinwerfer sind. Jeder Spybotic in der Nähe empfängt die Pings von anderen Spybotics in seinem Gesichtsfeld und berechnet daraus grob die Entfernung und die Richtung jedes Senders. Dies macht jeder Spybotic praktischerweise automatisch und legt die Ergebnisse in einer so genannten „Welt-Tabelle“ im Speicher ab. Eigene Programme können

EINBLICKE

Am geöffneten Lego Spybotic kann man klar die Infrarot-Sender und -Empfänger erkennen. Außerdem haben die Module zwei Motoren, einen Tastsensor vorne, einen Lichtsensor hinten sowie mehrere LEDs und einen Tongenerator. Sie können mit normalen Lego-Technik-Bauteilen erweitert werden und es gibt Infrarot-Fernsteuerungen. Spybotics wurden 2002 von Lego vorgestellt, aber nur kurze Zeit verkauft. Lego hat 2008 jegliche Unterstützung eingestellt. Aber man findet noch viele Informationen im Internet. Heute kann man sie nur noch gebraucht kaufen. Bei eBay kosten sie ohne Zubehör je nach Zustand zwischen 10 und 20 €. Damit waren sie die einzigen Roboter, die für uns in größerer Zahl erschwinglich und für unser Projekt geeignet waren.



auf die Daten zugreifen. Zwar muss man sich selbst nicht um die Erstellung der Welt-Tabelle kümmern, allerdings hat es den Nachteil, dass man die internen Funktionen nicht ändern oder die Genauigkeit verbessern kann.

Spybotic-Programmierung

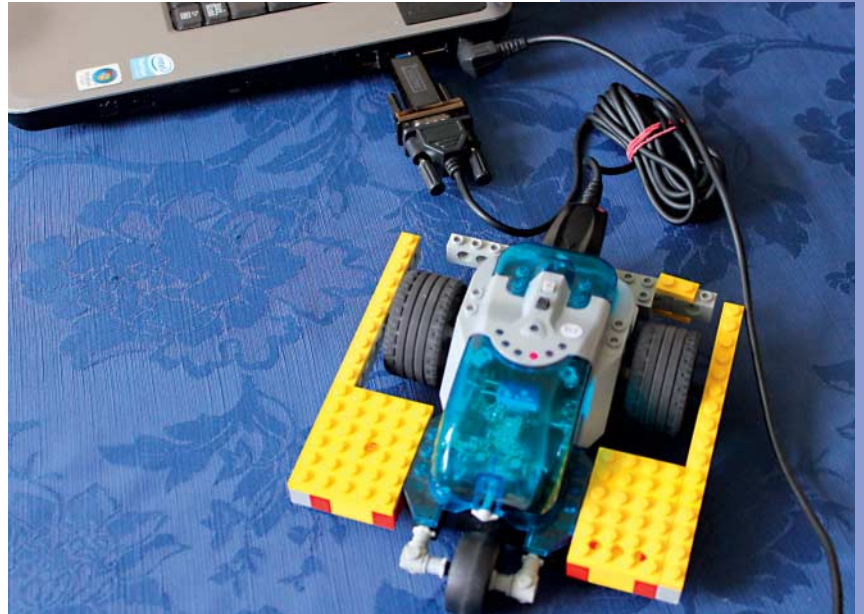
Lego Spybotics können mit vielen Programmiersprachen und Werkzeugen programmiert werden. Wir hatten bisher Lego Mindstorms in der grafischen Programmiersprache „RCX Code“ programmiert. Aber die gab und gibt es für Spybotics leider nicht. Wir haben uns deshalb für „Not Quite C“ (nicht ganz C, NQC) als Programmiersprache entschieden. Die IDE Bricx Command Center (BCC) vereint einen NQC-Compiler und eine grafische Oberfläche zum Programmieren und Testen. Für NQC gibt es diverse Anleitungen und eBooks auf deutsch im Internet. NQC ist eine einfache Variante der Programmiersprache C, die viele Sonderbefehle für die Programmierung von Lego-Robotern enthält, etwa zum Ansteuern der Motoren oder das Auswerten von Sensoren. Ein einfaches Programm, das den Spybotic eine Sekunde mit Höchstgeschwindigkeit geradeaus fahren lässt, sieht in NQC so aus:

```
task main()
{
  SetPower(OUT_A+OUT_B,7);
  OnFwd(OUT_A+OUT_B);
  Wait(100);
  Off(OUT_A+OUT_B);
}
```

Das Hauptprogramm ist immer task main, die beiden Motoren sind als OUT_A und OUT_B definiert. SetPower() setzt für beide die Geschwindigkeit 7. Die Funktion OnFwd() schaltet für beide den „Vorwärtsgang“ ein, Off schaltet sie nach 1 Sekunde wieder aus. Setzt man die Motoren auf unterschiedliche Geschwindigkeiten, ist es möglich, Kurven zu fahren.

Das BCC bietet für die Programmierung einen komfortablen Editor und mehrere Möglichkeiten, den Ablauf des Programmes auf dem Spybotic zu beobachten. So kann man sich im Debugger die Werte der Variablen während des laufenden Programms anschauen. Das ist besonders vorteilhaft bei der Fehlersuche, da die Spybotics außer den LEDs keine Anzeige besitzen.

BCC überträgt die (in Byte-Code) übersetzten Programme über Infrarot-Programmierkabel vom PC auf den Spybot. Für das Programmierkabel braucht man einen USB-Seriell-Adapter, da die meisten PCs mittlerweile keine serielle Schnittstelle mehr besitzen. Wir haben unsere Programme auf PCs unter Windows Vista und 7 entwickelt. Die Programme können Sie unter dem Link am Ende des Artikel downloaden.



Das Programm muss seriell per Infrarotkabel und Adapter auf das Modul geladen werden.

Die Lego Spybotics sind sehr einfach aufgebaut und haben zahlreiche Einschränkungen. So kann man nur mit ganzen Zahlen rechnen und nur maximal 32 Variablen und 7 Unterprogramme benutzen. Der Programmspeicher ist nur 4 KByte groß. Der Prozessor ist mit 3 MHz auch ziemlich langsam. Wir haben mit einem Testprogramm gemessen, dass ein Zugriff auf die Welt-Tabelle im Mittel 13 ms dauert, das heißt der Rechner kann höchstens 75 mal pro Sekunde einen Eintrag abfragen. Das liegt daran, dass die Programme vom Compiler nicht in Maschinsprache übersetzt werden, sondern in Byte-Code. Der Bytecode wird auf dem Spybotic Schritt für Schritt interpretiert und ausgeführt, was relativ langsam geht.

Die Welt-Tabelle

Name	Beschreibung	Werte
SPY_NOTE	eigene Notiz über diesen Sender	0–255
SPY_LINKID	Fernbedienung, mit dem dieser Spybot verbunden ist	0–7 (0 bedeutet: nicht mit einer Fernbedienung verbunden)
SPY_RANGE	Entfernung des Senders	Nowhere(0), anywhere(1), there(2), here(3)
SPY_DIRECTION	Richtung des Senders	Left(0), Center left(1), Center(2), Center right(3), Right(4)
SPY_ASPECT	Ausrichtung des Senders zum Empfänger	Front Left(0), Front(1), Front right(2), Back Right(3), Back(4), Back Left(5)
SPY_INFO	Nachricht des Senders	0–255
SPY_SHORTID	Identifikation des Senders	0–255

Dies hieß für uns, dass unsere Programme einfach sein mussten, damit die Spybotics schnell genug auf die Umgebung reagieren können. Die Roboter fahren maximal mit einer Geschwindigkeit von 2 Meter pro Sekunde. Daher muss man 5 bis 10 Mal pro Sekunde die Umgebung abfragen, um auf Signale reagieren zu können. Das Programm ist für alle Module identisch, es gibt keinen Anführer oder dergleichen.

Umweltdaten

Die „Welt-Tabelle“ (siehe Seite vorher) gibt Auskunft über andere Infrarotsender (Spybotics oder Fernbedienungen) in der Umgebung. Die Roboter aktualisieren sie automatisch und unabhängig von den Anwenderprogrammen. Theoretisch könnte jeder Spybot bis zu 16 Infrarotsender in der Welt-Tabelle verwalten. Aus der Welt-Tabelle kann man über jeden Sender immer die aktuellen Informationen abrufen.

Die Welt-Tabelle enthält nur grobe Informationen über die Umgebung. Insbesondere kann ein Spybot nur andere Infrarot-Sender erkennen, aber keine Hindernisse. Auch gibt es keine Möglichkeit, die Ge-

schwindigkeit anderer Spybotics selbst zu ermitteln. Aber jeder Roboter könnte seine eigene Geschwindigkeit in `SPY_INFO` eintragen und so den anderen mitteilen. Die Werte in der Welt-Tabelle erscheinen allerdings nicht immer logisch, zum Beispiel haben größere Entfernungen kleinere Zahlenwerte.

Die Einträge der Welt-Tabelle kann man über einen Index abrufen. Jeder gültige Eintrag hat eine „Hausnummer“ von 0 bis 15. Es gibt NQC-Befehle, mit denen man nach Einträgen suchen kann, die bestimmte Eigenschaften haben. Der Befehl `FindWorld(index,SPY_RANGE,REL_GT,RANGE_NOWHERE)` sucht den nächsten Eintrag, von dem überhaupt ein Ping geortet wurde (Entfernung größer als 0), und liefert dessen Hausnummer in der Variablen `index` zurück. Wenn kein passender Eintrag gefunden wurde, wird -1 als Hausnummer gesetzt. `RANGE_NOWHERE` und andere häufig vorkommenden Werte sind als Konstanten vordefiniert. Wir haben sie für unsere Programme mit deutschen Namen ersetzt, z. B. `NIRGENDWO`. Zur Vereinfachung kann man mit `SetTargetID(index)` ein Ziel für alle weiteren Abfragen setzen und mit Befehlen wie `Target(SPY_RANGE)` die Entfernung des Ziels abfragen. Dies funktioniert genauso für die anderen Werte in der Welt-Tabelle.

VERMESSUNG DER WELT

Um besser zu verstehen, was die Einträge in der Welt-Tabelle in Wirklichkeit bedeuten, haben wir Tests gemacht. Wir änderten die Entfernung eines IR-Senders und beobachteten die Variable `SPY_RANGE` im BCC, während ein einfaches Programm läuft, das kontinuierlich die Welt-Tabelle liest. Wir veränderten den Winkel und die Entfernung der beiden Spybotics zueinander und trugen die abgelesenen Entfernungen in ein Diagramm ein. Die Ergebnisse variierten mit dem Umgebungslicht. Wir haben unsere Messungen bei normalem Tageslicht im Schatten gemacht. Im Sonnenlicht werden die Ergebnisse schlechter, in der Dunkelheit besser. Der Grund ist der Infrarotanteil des Lichts. Er ist für Menschen nicht sichtbar. Aber im Dunkeln kann man Infrarotlicht mit einer einfachen Digitalkamera sichtbar machen. Die Pings der Roboter waren allerdings zu kurz, als dass wir sie fotografieren konnten.

Bei den Ergebnissen überraschte uns, dass sich die Spybotics auf große Entfernungen erkennen können (gelb=FERN), erst ab 6 bis 8 m klappt es nicht mehr. Die Entfernungsbestimmung ist wegen der etwas schrägen Ausrichtung der Empfänger vom Winkel abhängig. Im vorderen Gesichtsfeld beginnt zwischen 50–150 cm der mittlere Bereich (orange=MITTEL) und zwischen 15–40 cm der Nahbereich (rot=NAH). Die Erkennung der Richtung (`SPY_DIRECTION`) ist bei größerer Entfernung sehr unzuverlässig, im Fernbereich (gelb=FERN) wird meistens MITTE gemeldet.



Das Gesichtsfeld des Spybotics zeigt klar die Ausrichtung der Infrarot-Empfänger sowie die Ungenauigkeit der Ortung.

Sucher

Um uns in die Programmierung der Spybotics einzuarbeiten, haben wir mit einer einfachen Aufgabe angefangen: Ein Roboter sollte möglichst nahe zu einem Infrarot-Sender hinfahren. Dabei sollte es nur einen Sender geben, der sich im Gesichtsfeld befindet. Wir nahmen auch an, dass sich keine Hindernisse im Weg befinden. Die Grundidee unseres Programms ist einfach. Wir lesen in einer Schleife immer wieder die Welt-Tabelle und schalten die Motoren so, dass wir immer etwa in Richtung des Ziels fahren. Wenn wir NAH dran sind, bleiben wir stehen. Der Ablauf sieht wie folgt aus:

1. Finde den Eintrag in der Welt-Tabelle und lies ihn
2. Wiederhole solange bis der Abstand NAH ist (SPY_RANGE=NAH)
 - 2a. Lies den Eintrag
 - 2b. Fahre in die Richtung des Senders (SPY_DIRECTION)
3. Fahre noch etwas näher an den Sender und halte an

Schritt 2b kann man noch derart abändern, dass man in Abhängigkeit von der Entfernung unterschiedlich schnell fährt (die Motoren haben sieben Geschwindigkeitsstufen). Für die verschiedenen Richtungen haben wir jeweils ein kleines Unterprogramm geschrieben (in der Header-Datei `spy_deutsch.nqh`, beginnend mit dem Prefix `vorw_`), das die Motoren entsprechend steuert. Der wesentliche Teil zum Suchen des Infrarotsenders sieht in NQC wie in Listing 1 aus: Die Konstanten haben wir durch deutsche Namen ersetzt (ebenfalls in `spy_deutsch.nqh` definiert), um das Programm lesbarer zu machen. `FindWorld()` findet den Sender in Welt-Tabelle, den `SetTarget(ID)` als Ziel setzt. Die Variable `entfernung` erhält die Distanz zum Sender. Solange der Spybot dem Ziel nicht nah genug ist, wird die While-Schleife ausgeführt. Über das Switch-Case-Konstrukt lenkt der Spybotics in die Richtung des Senders. Sobald er nah genug dran ist, verlässt das Programm die While-Schleife und er fährt langsam noch ein kleines Stück weiter vorwärts.

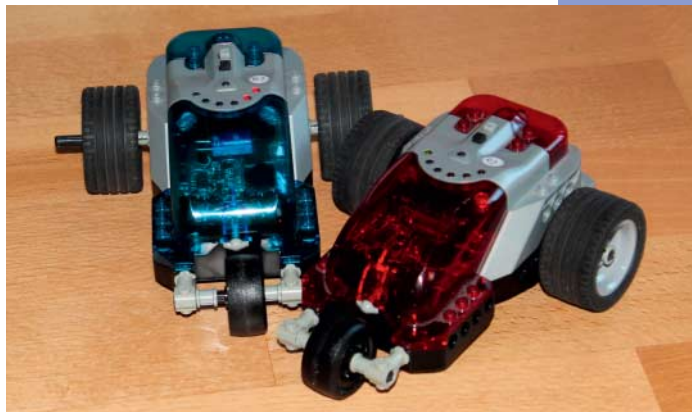
So klappt das Suchen von Infrarot-Quellen recht gut. Das Verhalten der Spybotics variiert aber von Versuch zu Versuch. Es hängt stark vom Untergrund, der Batterieladung, der Raumbeleuchtung sowie Verschleiß des Motors ab. Jeder Roboter verhält sich deshalb anders. Darum muss man im Programm möglichst schnell immer wieder auf die Veränderungen der Umgebung, also Veränderungen in der Welt-Tabelle reagieren.

Kolonne

Als Erweiterung der ersten Aufgabe haben wir uns eine ähnliche Aufgabe ausgedacht: das Kolonnenfahren. Dabei sollte ein Spybot dem anderen hinterherfahren. Da die Roboter sich bei großer Entfernung nur schwer finden können, mussten sich die Spybotics in naher oder mittlerer Entfernung zueinander befinden. Außerdem sollten sie die einzigen Roboter in der Umgebung sein. Der voranfahrende darf nicht zu schnell fahren, da er den hinteren Spybot nicht abhängen soll. Die Schwierigkeit besteht

Listing 1

```
FindWorld (index,SPY_ RANGE,REL_ GT,NIRGENDWO);
SetTargetID(index);
entfernung = Target(SPY_RANGE);
while (entfernung != NAH)
{
    entfernung = Target(SPY_RANGE);
    richtung = Target(SPY_DIRECTION);
    switch (richtung)
    {
        vorw_links (LANGSAM);
        break;
        case HALBLINKS :
        vorw_halblinks (ZUEGIG);
        break;
        case MITTE :
        vorw (SCHNELL);
        break;
        case HALBRECHTS :
        vorw_halbrechts (ZUEGIG);
        break;
        case RECHTS :
        vorw_rechts (LANGSAM);
        break;
    }
}
vorw (LANGSAM);
Wait(50);
```



Die Spybotics verhakten sich oft bei der Schwarmbildung und blockierten sich dann gegenseitig.



Am besten funktionierten unsere Experimente mit großen Hinterrädern und nur einem, möglichst glatten Vorderrad, damit sich die Roboter gut drehen können.

ROBOTER PROGRAMMIEREN

Listing 2

```

switch (richtung){
  case LINKS :
    switch (ausrichtung){
      case ZIEL_HINTEN_RECHTS:
        anpassen(geschwindigkeit, geschwindigkeit_nachbar);
        vorw_halblinks (geschwindigkeit);
        break;
      case ZIEL_HINTEN:
        anpassen(geschwindigkeit, geschwindigkeit_nachbar);
        vorw_links (geschwindigkeit);
        break;
      default:
        vorw(geschwindigkeit);
        break;
    }
    break;
  case HALBLINKS :
    anpassen(geschwindigkeit, geschwindigkeit_nachbar);
    vorw_halblinks(geschwindigkeit);
    break;
  ...
}

```

darin, dass man die genaue Entfernung nicht ermitteln kann. Man kann nur auswerten, ob die Entfernung NAH oder MITTEL ist und dann die Geschwindigkeit verringern oder erhöhen. Da der Spybotic nur Geschwindigkeiten von 0 bis 7 kennt, änderten wir die Geschwindigkeit jeweils um eine Stufe. Wir wollten damit erreichen, dass sich die Geschwindigkeiten der Spybotics annähern und sich der Abstand auf circa 30 cm einpendelt.

Das Kolonnenfahren mit Geschwindigkeitsregelung funktioniert folgendermaßen:

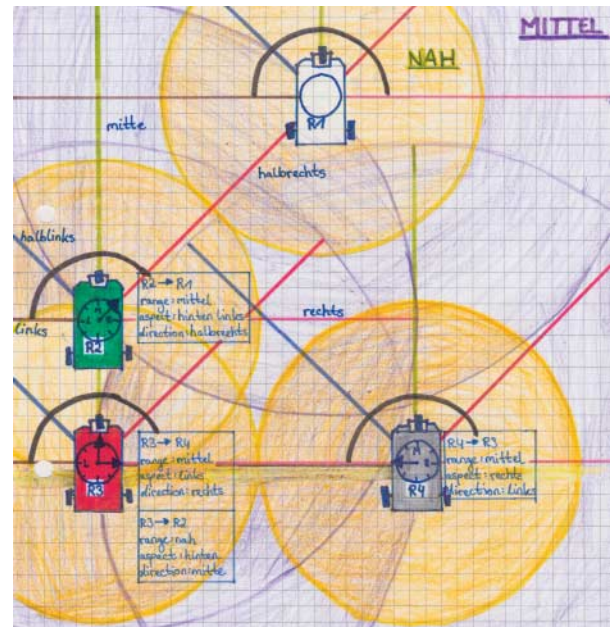
1. Finde den (einzig) Eintrag in der Welt-Tabelle und lies ihn
2. Wiederhole solange der Abstand nicht FERN ist (SPY_RANGE!=FERN) (oder ein anderer Fehler auftritt)
 - 2a. Lies den Eintrag
 - 2b. Fahre in die Richtung des Senders (SPY_DIRECTION)
 - 2c. Wenn die Entfernung zum anderen Spybotic NAH ist, dann verringere die Geschwindigkeit.
 - 2d. Wenn die Entfernung zum anderen Spybotic MITTEL ist, dann erhöhe die Geschwindigkeit.

Im Prinzip klappt das Kolonnenfahren ganz gut (siehe Video), allerdings gibt es auch einige Probleme: Die Roboter fahren ziemlich dicht hintereinander und teilweise auch aufeinander. Wenn der vordere Roboter eine enge Kurve fährt, fährt der Nachfolger manchmal an ihm vorbei und verliert ihn. Diese Probleme gibt es aber teilweise auch im normalen Autoverkehr: wenn zum Beispiel fünf Autos in Kolonne zu einer Sportveranstaltung fahren wollen, darf das erste Auto auch nicht zu schnell fahren oder spontan abbiegen, sonst wird die Kolonne abgehängt.

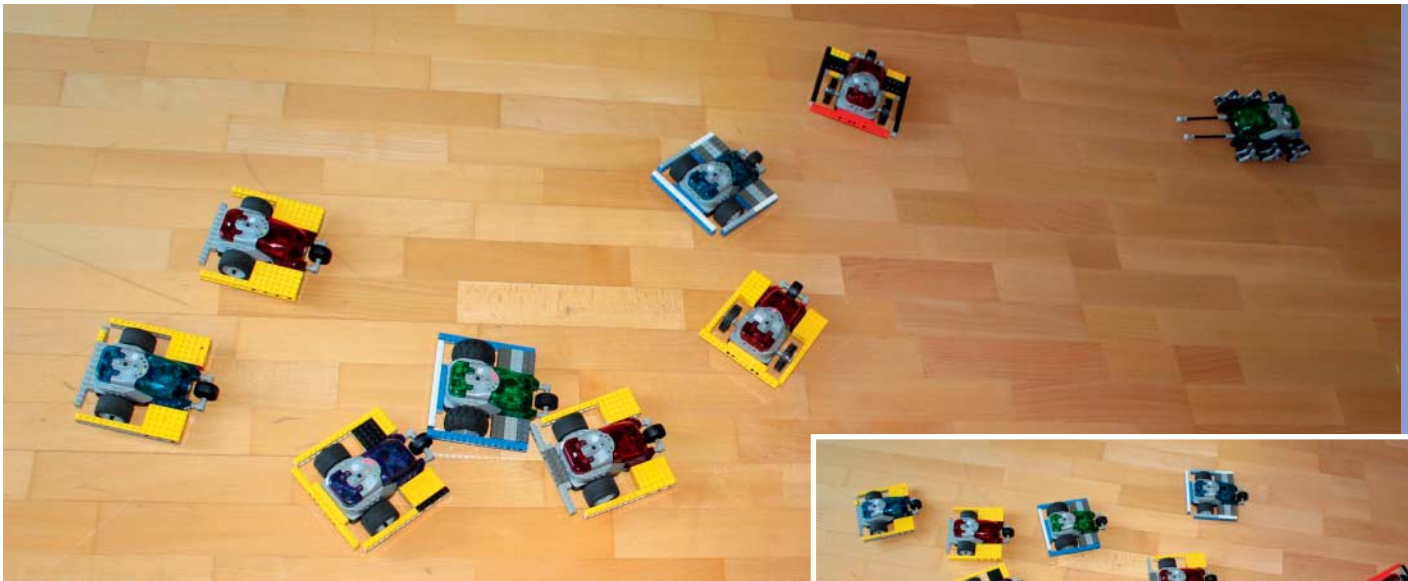
Schwarm

Wie anfangs geplant, wollten wir die Roboter im Schwarm fahren lassen. Da die Positionen von weit entfernten Spybotics ungenau sind, ist es leichter, den Roboter nur nahe oder mittelweit entfernte Spybotics suchen zu lassen. Das bedeutet aber auch, dass sich die Module beim Start in naher oder mittlerer Entfernung zueinander befinden müssen.

Wir versuchten, die einfachsten Schwarmregeln wie bei Fischen zu programmieren. Dazu muss man

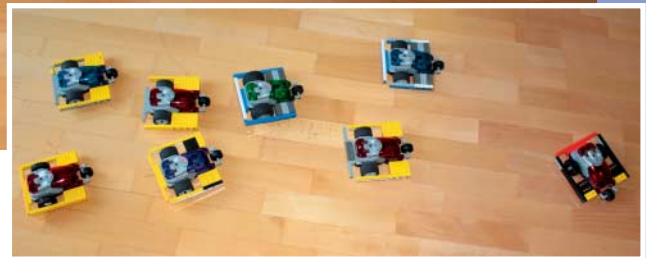


Bei der Fehlersuche haben wir viele verschiedene Szenarien und Konstellationen durchgespielt. Stift und Papier waren dabei unsere Hilfsmittel.



Der Schwarm weicht einem Feind aus.

Mit Blockierschutz gelingt die Schwarmbildung in den meisten Fällen.



die Geschwindigkeiten von den anderen Spybotics kennen, aber es gibt keine Möglichkeit, sie selbst zu messen. Deshalb soll jeder Roboter seine eigene Geschwindigkeit als Information an alle anderen versenden. Das lässt sich mit dem Befehl `SetPingData()` erledigen, mit der jeder Spybot einen Wert an andere übertragen kann. Konkret überträgt man den Geschwindigkeitswert zwischen 1 und 7, den die Nachbarn dann aus ihrer Welt-Tabelle aus der Variablen `SPY_INFO` auslesen können.

Der Programmablauf ist im Prinzip einfach:

1. Wiederhole die folgenden Schritte (ohne Ende)
2. Pinge die eigene Geschwindigkeit (`SetPingData(geschwindigkeit)`)
3. Finde den nächsten Eintrag in der Welt-Tabelle mit Abstand NAH oder MITTEL
4. Lies den Eintrag
5. Passe Fahrtrichtung in Abhängigkeit von Richtung und Ausrichtung des anderen Spybot an
6. Passe dich dessen Geschwindigkeit an (`SPY_INFO`)

Obwohl die Regeln einfach sind, wird das Programm trotzdem relativ aufwendig. Damit das Fahren im Schwarm gut funktioniert, mussten wir nämlich für jede Kombination von Positionen der Spybots zueinander eine Reaktion programmieren. Wenn ein Roboter einen anderen Roboter in Richtung halb links von hinten rechts sieht, soll er nach halblinks fahren und ihm folgen. Der Quellcode steht unter dem Link am Ende zur Verfügung. Ein exemplarischer Ausschnitt sieht wie in Listing 2 aus.

Anfangs hatten wir noch Programmierfehler gemacht und die Spybotics sind in alle möglichen Richtungen auseinander gefahren. Um die Fehler besser finden zu können, haben wir uns Variablen wie `SPY_DIRECTION` oder `SPY_ASPECT` mit den LEDs

anzeigen lassen. So konnten wir besser erkennen, was die einzelnen Roboter gerade gesehen haben.

Aber auch nach Beseitigung der Programmierfehler gab es noch Probleme. Die Roboter erkennen baubedingt nicht gut, wenn sie parallel zueinander herfahren. Mitunter meinen sie zu früh, dass sie nebeneinander fahren, obwohl sie noch schräg hintereinander fahren. Die Robotermodule können zwar die Geschwindigkeiten pingen, allerdings hängt die tatsächliche Geschwindigkeit von der Batteriestärke und der Größe der Räder ab. In der Praxis sind die Spybotics deshalb trotz gleicher Geschwindigkeitseinstellung unterschiedlich schnell. Dennoch waren unsere Experimente erfolgreich. Auch mit den vergleichsweise leistungsschwachen, aber günstigen Robotern konnten wir Schwarmverhalten simulieren.

Das zum Download angebotene Programm `mission.nqc` enthält sowohl die Kolonnenfahrt, die Schwarmbildung als auch das Ausweichen vor Feinden. Der Clou: Stellt man die Roboter als Kolonne hintereinander auf, dann fahren sie auch Kolonne (da sie in der Regel nur einen Spybot vor sich sehen). Wenn sie mehrere sehen, halten sie sich an die Schwarmregeln.

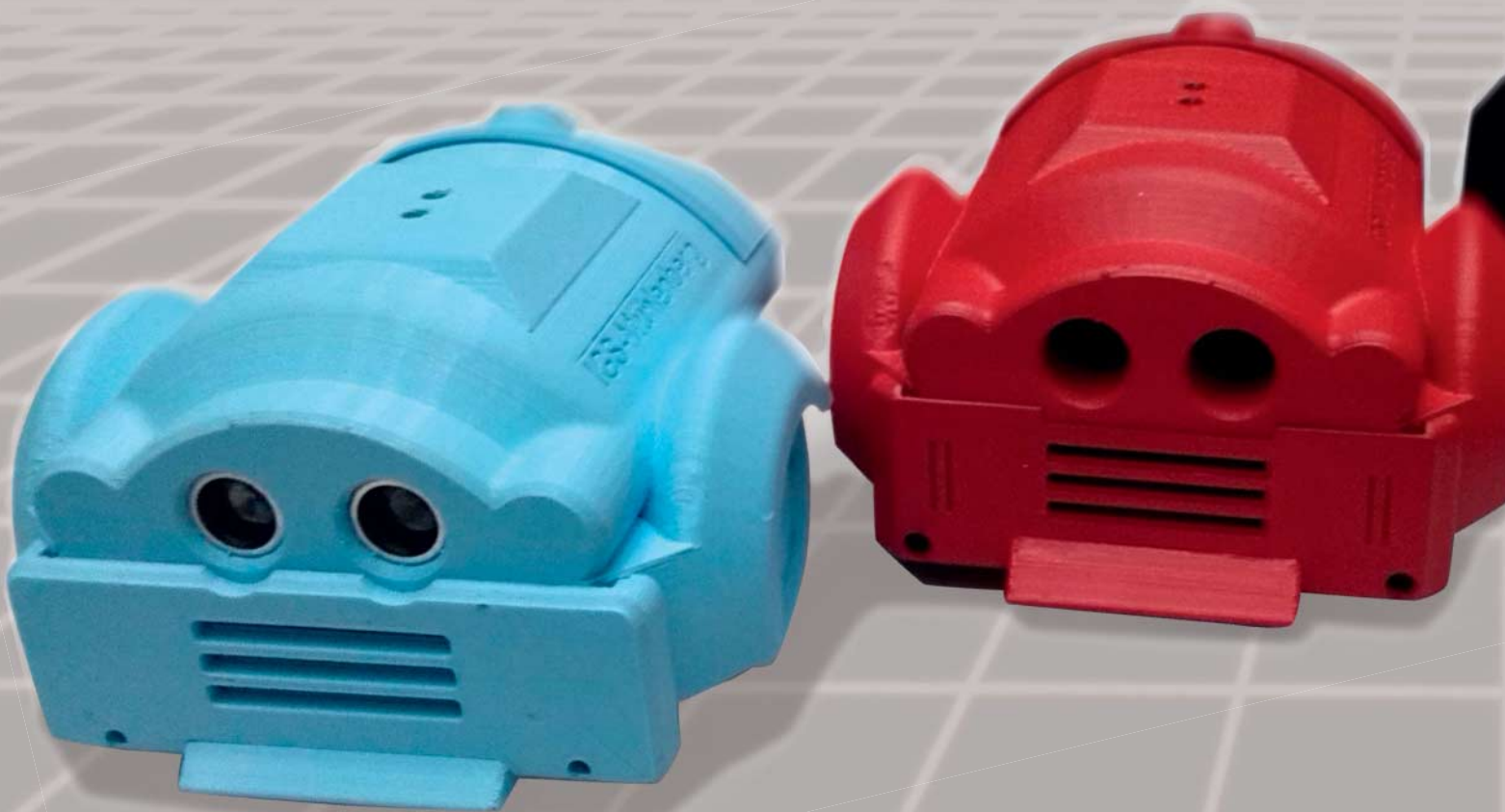
Wenn kein Feind da ist, kommt das Ausweichprogramm nicht zum Tragen. Nur der Feind folgt anderen Regeln, das wir als `Feind2.nqc` hinterlegt haben. Trifft der Schwarm auf den Feind, weichen die Spybotics aus. Und nun viel Spaß beim eigenen Roboterbasteln. —dab

↓ Links und Foren
make-magazin.de/xcjt

R4e1 Robot 4 each 1

Die IGS Mühlenberg in Hannover hat ein Konzept für einen günstigen Arduino-kompatiblen Roboter entwickelt, den andere Schulen und Hobby-Robotiker leicht nachbauen können. Das Chassis steht als 3D-Druckdateien kostenlos zum Download zur Verfügung.

von Ersan Beksan, Ralf Lürig und Marco Nicolai



Kurzinfo



Zeitaufwand:
2 Stunden (ohne 3D-Druck)



Kosten:
60 Euro



Programmieren:
ArduBlock

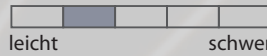


Löten:
leichte Lötarbeiten



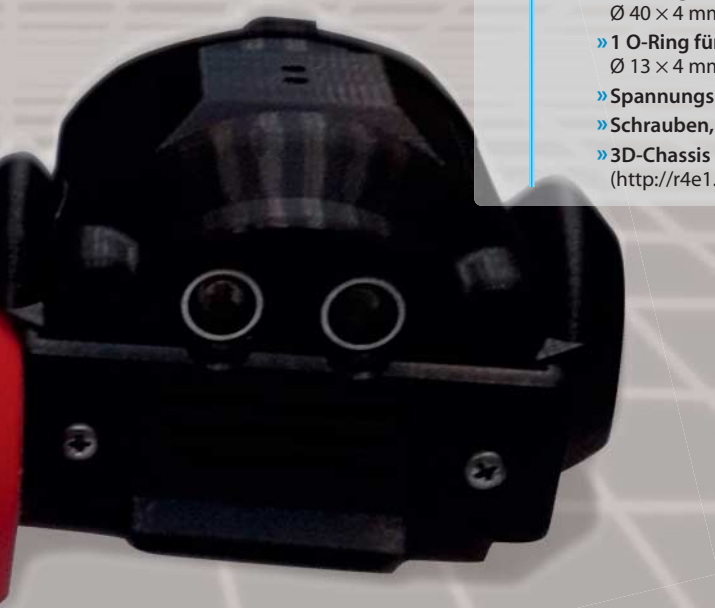
Feinmechanik:
Servos hacken

Schwierigkeitsgrad



Einkaufsliste

- » Sainsmart Uno R3
- » 1 Ultraschallsensor HC-SR04
- » 2 Liniensensoren RedBot Sensor
- » 2 Servos Towerpro MG995
- » 4 Lithium-Ionen-Akku
3,6 V 900 mAh
- » 2 O-Ringe als Reifen
Ø 40 × 4 mm
- » 1 O-Ring für ein kleines Rad
Ø 13 × 4 mm
- » Spannungsregler 78S05
- » Schrauben, Kleinteile
- » 3D-Chassis
(<http://r4e1.igsmuehlenberg.de/>)



Eigentlich sollte es heutzutage eine Selbstverständlichkeit sein, dass Schülerinnen und Schüler am PC und weiterführend auch in der Programmierung geschult werden. Dass dafür aber so gut wie keinerlei Ressourcen zur Verfügung gestellt werden, ist der breiten Öffentlichkeit kaum bekannt. Die vom Land Niedersachsen vorgegebenen Stundenpläne, die so genannten Stundentafeln, weisen beispielsweise keine einzige Schulstunde PC-Unterricht während der gesamten Schullaufbahn auf. Schulen, die dennoch PC-Unterricht geben, schieben in ihren begrenzten Möglichkeiten Stunden hin und her, um ihren Schülern den „Luxus“ einer Einführung in die Informatik geben zu können.

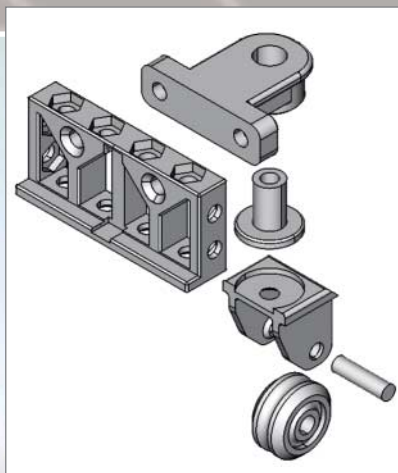
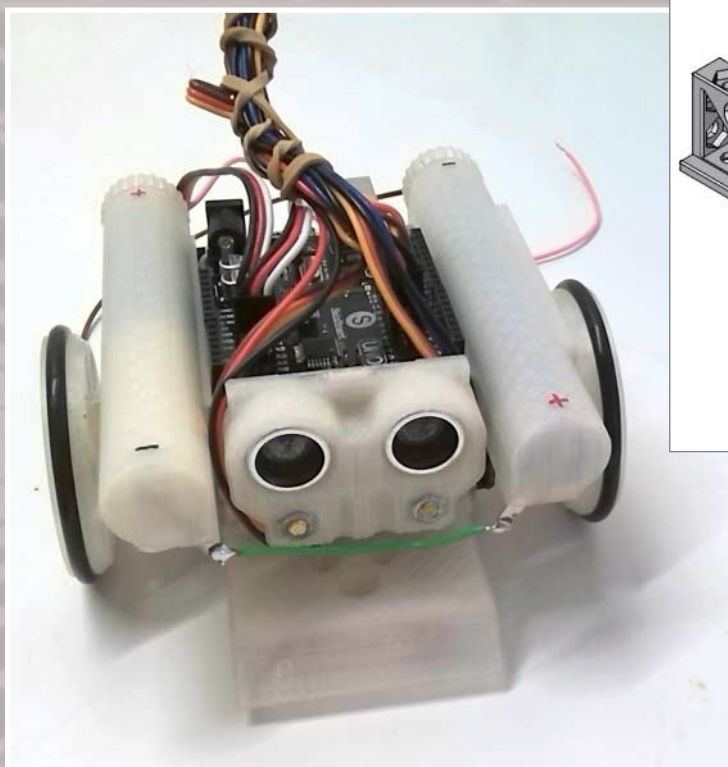
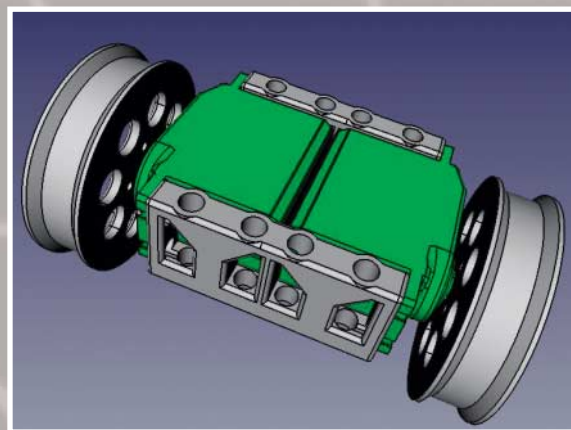
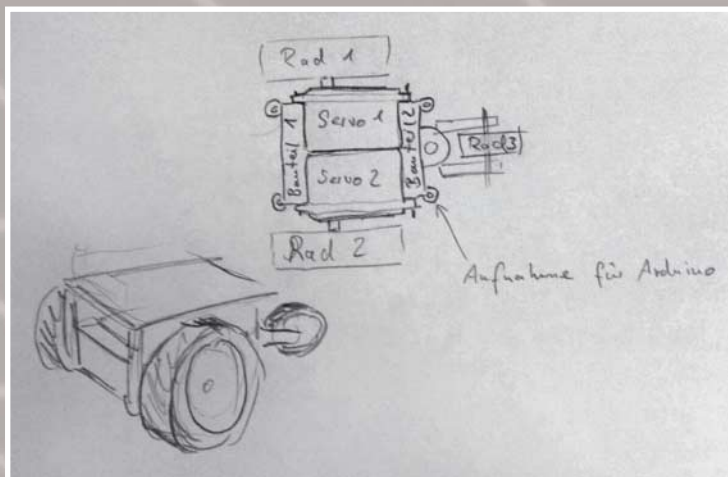
Ein weiteres Problem ist das notwendige technische Zubehör: Die Kommunen oder Städte stellen dafür keine ausreichenden finanziellen Mittel zur Verfügung. Ergibt sich dann hier und da doch die Möglichkeit, etwa mit Robotern zu arbeiten und sie zu programmieren, ist mit hundertprozentiger Sicherheit die Spende einer Firma oder Stiftung vorausgegangen. So etwas von Steuergeldern zu zahlen, wäre eigentlich doch ein Bekenntnis zu guter Bildung.

Selbst Angebote an Bildungseinrichtungen von europäischen Firmen, die mittlerweile nicht mehr in Deutschland produzieren, liegen für den Bausatz eines Roboters bei nahezu 400 Euro. Diese schlechten Rahmenbedingungen haben uns aber nicht davon abgehalten, ein kleines „Lichtlein“ anzuzünden, obwohl wir auch mal gerne über die Finsternis schimpfen, wie man oben lesen kann. Die Idee war, eine Möglichkeit für Schüler zu schaffen, die finanziell interessant, schultauglich ist und Interesse am Robotik- und Informatikunterricht fördert. Eine günstige Konstellation von Kollegen an der IGS Mühlenberg ergab eine fruchtbare Kooperation, so dass die Bereiche Konstruktion, Elektronik und Organisation sich ergänzen konnten.

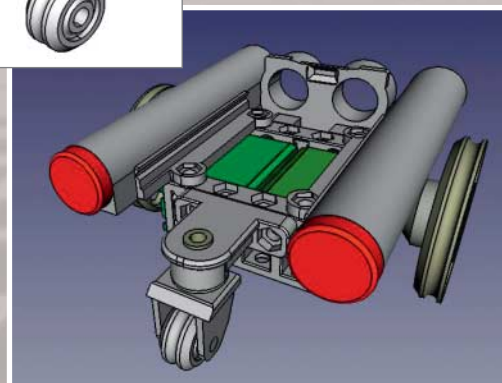
Die Spende einer Firma ermöglichte außerdem die Anschaffung von Versuchs- und Unterrichtsmaterial, sowie eines 3D-Druckers, der die nötige Genauigkeit hatte. Außerdem stellte sich der Förderverein der Schule hilfreich an unsere Seite. So waren Grundlagen gelegt, einen Roboter für jedermann, einen „Robot for each one“ oder kurz R4e1 zu entwickeln und in naher Zukunft zu bauen.

Gehäuse

Anfangs arbeiteten wir mit dem Lego Mindstorms NXT, mit dem Schüler erste Programmierübungen durchführen können. Doch die Kästen sind zu teuer, um sie in größeren Stückzahlen anzuschaffen. Die Idee lag nahe, selber Roboter zu bauen. Dies ge-



Im ersten Schritt des Projekts entwickelten wir ein Chassis, um die einzelnen Komponenten zusammenzuhalten. Später entwarfen wir ein vollständiges Gehäuse.



schah zunächst mit Fertigbausätzen und dem Arduino. Mit der Anschaffung eines 3D-Druckers eröffneten sich neue Möglichkeiten. Als erstes entwarfen wir Bauteile, die die Servos und Sensoren zusammenhielten. Danach entwickelten wir Räder, Verbindungen für den Arduino und das Batteriefach. Zur Bereifung wurden handelsübliche O-Ringe verwendet.

Um den Roboter zu Programmierübungen auch Schülern zugänglich zu machen, dachten wir darüber nach, eine komplette Karosserie zu konstruieren, um die Kabel und die Elektronik zu schützen. Außerdem sollten Schüler in die Konstruktion einbezogen werden. Mit dem

Programm FreeCAD haben wir basierend auf dem bisherigen Chassis ein vollständiges Gehäuse entworfen. Räder und Kühlergrill wurden von mehreren Schülern konstruiert. Somit besitzen wir ein recht großes Sortiment an variablen Bauteilen, die miteinander kombiniert Design-Varianten zulassen.

Bei der Konstruktion waren einige Dinge bezüglich des 3D-Drucks zu beachten. Der wichtigste Punkt ist, dass ein 3D-Drucker nicht in die Luft drucken kann. Somit sind überhängende Bauteile nicht möglich. Unser 3D-Drucker schafft gerade noch einen Winkel von 30 Grad zur Horizontalen, besser sind jedoch 45 Grad. Durch ent-

sprechende Ausrichtung auf der Druckplatte erweitern sich die Möglichkeiten.

Antrieb und Gehirn

Als Antrieb für den R4e1 dienen zwei Servos vom Typ Towerpro MG995. Vorteilhaft bei diesem digitalen Servotyp sind vor allem der günstige Preis, der relativ einfache und gut dokumentierte Umbau zu einem kontinuierlich rotierenden Servo, das robuste Metallgetriebe und seine Leistungsstärke. Zudem erleichtert seine eckige Bauform, im Vergleich zu einem herkömmlichen DC-Getriebemotor, wesentlich den Einbau in unser 3D-Chassis. Der

Umbau von herkömmlichen 180-Grad-Servos lässt sich generell in zwei Varianten durchführen: in der PWM-Variante wird der interne Potentiometer durch zwei Festwiderstände (5 kOhm) ersetzt und damit permanent in Mittelstellung gebracht. Diese Variante hat den Vorteil, dass man keinen Motortreiber mehr benötigt und nur ein PWM-Pin benötigt wird. Bei der PWM-Variante wird der Servo mit den gleichen PWM-Signalen angesteuert wie ein nicht gehackter Servo. Das Eingangssignal wird vom Arduino-Mikrocontroller generiert. Die Pulsweite beeinflusst Drehrichtung und -geschwindigkeit des Motors.

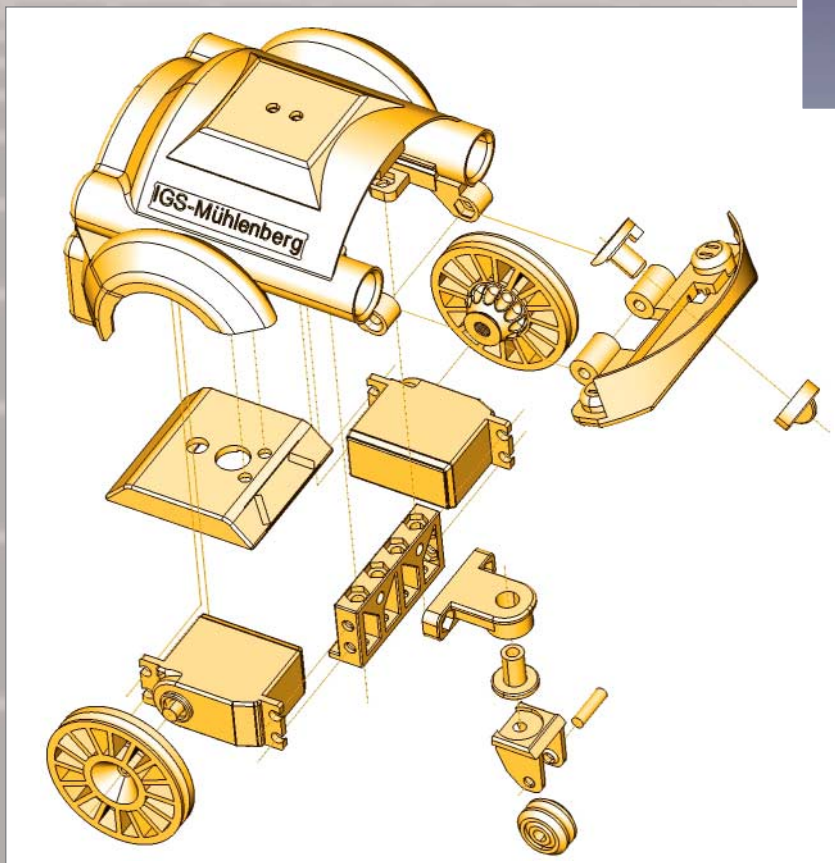
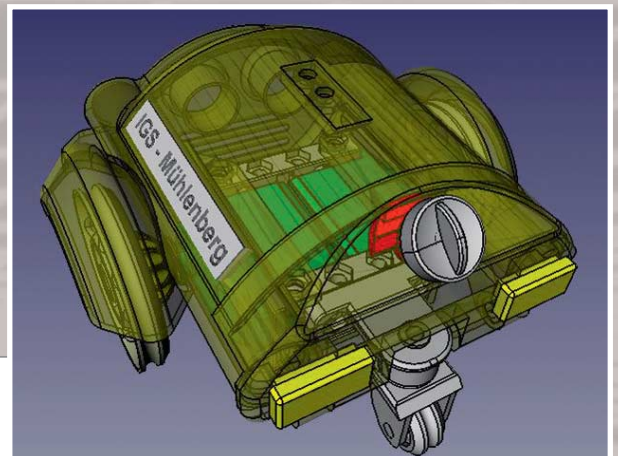
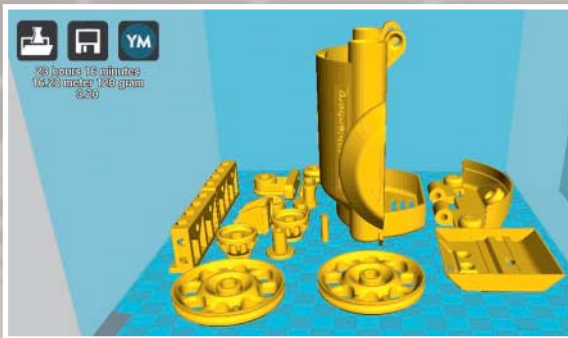
In der zweiten Variante, dem DC-Hacking, wird die Servoelektronik vollständig entfernt und der Motor direkt mit zwei Pins

verbunden. Bei dieser Variante wird der Servo mit Gleichstrom gesteuert. Es besteht hierbei kein Unterschied zwischen einem solchen Servo und einem handelsüblichen Getriebemotor. Die Drehrichtung und -geschwindigkeit kann mit der Polarität und Stärke des Gleichstromes gesteuert werden. Allerdings benötigt man hier einen zusätzlichen Motortreiber und man belegt zwei PWM-Pins des Arduinos pro Servo.

Bei beiden Varianten müssen die mechanischen Beschränkungen des 180-Grad-Servos entfernt werden. Beim MG995 erreicht man dies durch Entfernen eines kleinen Metallstifts und der kleinen Plastikhülse (Potentiometersteuerung) im Inneren des Enddachsenszahnrad. Wie man das genau macht, ist in diversen Tutorials im Internet

beschrieben oder wird auf YouTube vorgeführt (Link siehe unten). Entscheidet man sich für die PWM-Variante, muss man zum ordnungsgemäßen Betrieb des Servos eine zusätzliche Spannungsregelung einbauen.

Servos arbeiten mit einer Betriebsspannung von 4,8 bis 6 Volt. Eine höhere Spannung würde dauerhaft die Servoelektronik zerstören. Im Modellbau wird dies durch vorgeschaltete sogenannte BEC-Regler (Batterie Eliminator Circuit) verhindert. Eine einfache Spannungsregelung lässt sich durch einen Spannungsregler vom Typ 78S05 (5 V, max. 2 A), zwei Keramikkondensatoren und einer Freilaufdiode auf einer Platine verwirklichen (siehe Fritzing-Bild). Als Spannungsversorgung dienen zwei Ultrafire-AA-Lithiumthionylchlorid-Akkus mit 3,7 V Nenn-

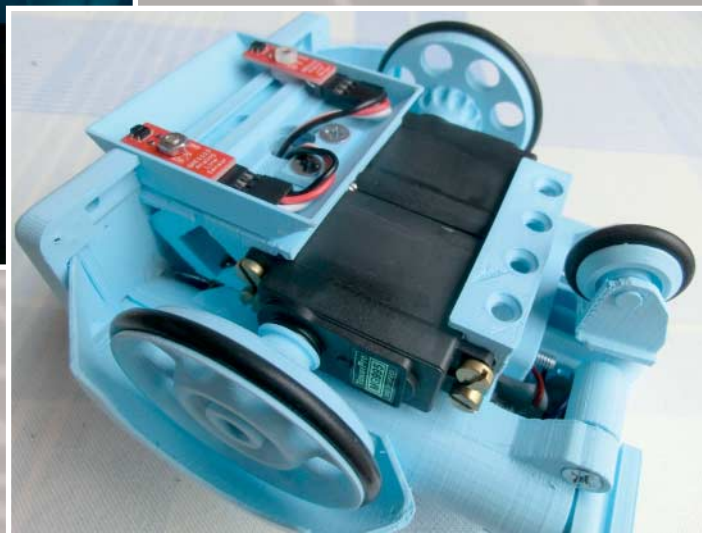


Explosionszeichnung aller Bauteile und die Positionierung für den 3D-Drucker (Download der stl-Datei siehe unten). Die gesamte Druckzeit betrug mit unserem Drucker 23 Stunden und 16 Minuten.





Für die Liniensensoren ist eine eigene Haltevorrichtung vorgesehen.



spannung, die in Reihe in das linke Batteriefach passen. Zwei weitere Akkus im rechten Batteriefach versorgen den Arduino.

Bei der DC-Variante entschieden wir uns für den Motortreiber L9110S. Jeder Kanal kann hierbei bis zu 800 mA liefern. Eine Spannungsregelung ist hierbei nicht notwendig, die Motoren werden mit den 7,4 Volt der beiden Li-Akkus versorgt. Zur Steuerung benötigt man insgesamt vier PWM-Pins des Arduinos. Diese sind bei uns die Pins 11, 10, 9 und 6.

Zur Hinderniserkennung dient ein günstiger Ultraschallsensor HC-SR04. Dieser passt exakt in die vordere Halterung und wird an die Pins 5 (Trigger) und 4 (Echo) angeschlossen. Aus der Laufzeit des reflektierten Echos ermitteln solche Sensoren den Abstand zu einem Hindernis. Zur Linienerkennung dient ein analoger Liniensensor von Sparkfun, den wir beim Internetversender Exp-tech günstig erwerben konnten. In die Halterung passen zwei oder mehr von diesen Sensoren und die Spurbreite lässt sich variabel einstellen. Um äußere störende Lichtfaktoren zu minimieren, haben wir mit dem 3D-Drucker eine Abdeckung konstruiert.

Als Mikrocontroller haben wir uns für den Arduino-kompatiblen Sainsmart Uno R3 mit ATmega328-AU entschieden. Dies lag zum einen am günstigen Preis, aber auch an den sehr praktischen 3er Pins (Gnd, 5V, Signal), an die man die Sensoren und Motortreiber mit Female-Female Steckverbindern anschließen kann. Der ATmega328-AU unterstützt sowohl 5 V und 3,3 V und besitzt dazu einen I/O-Spannungsumschalter. Es besitzt zwei weitere analoge Ausgänge (A6, A7) für zusätzliche Hardware. Dieses Arduino-kompatible Board lässt sich in unser Robo-Chassis reinschieben. Ein aufwändiges Verschrauben entfällt dabei.

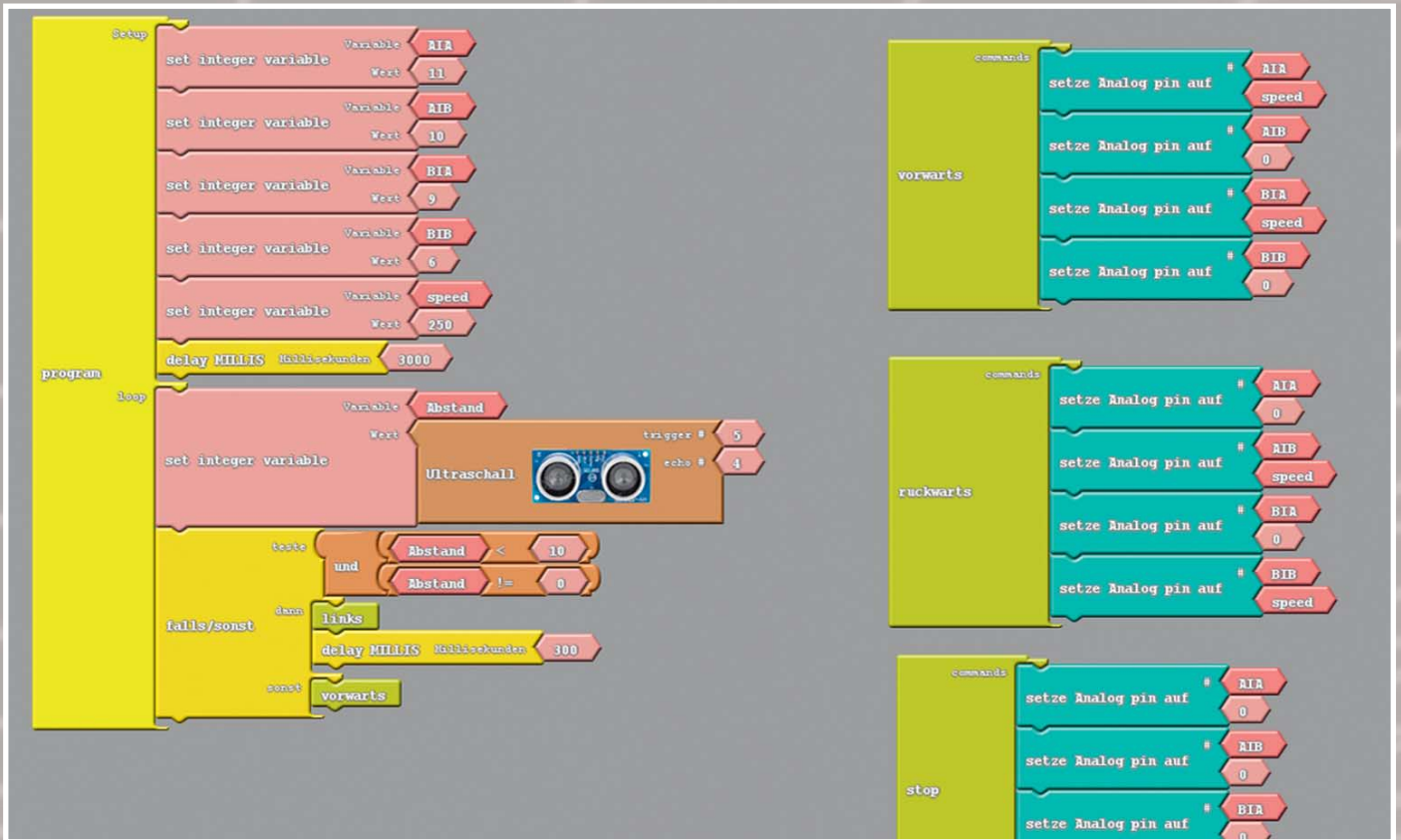
Programmiert wird das Ganze mit der grafischen Programmieroberfläche ArduBlock. Wir haben uns für diese Scratch-ähnliche Variante entschieden, um auch jüngere Schüler anzusprechen. Dabei handelt es sich um ein Java-Applet, welches Programmierblöcke bereitstellt. Diese müssen miteinander kombiniert werden. Für einzelne Bauteile wie den HC-SR04 existiert ein eigener Block, in den man nur noch die Pin-Nummern der Arduino-Anschlüsse eintragen muss. Eine

umständliche Programmierung entfällt somit.

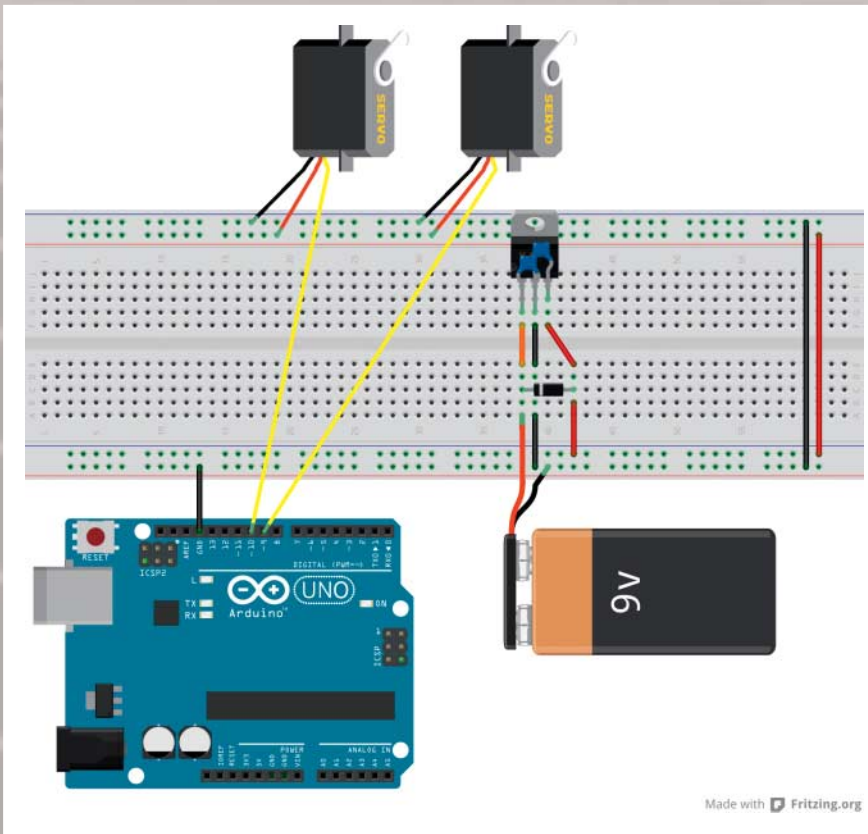
Aus den ArduBlock-Blöcken macht die IDE Programmcode, den man in die Arduino-IDE lädt, kompiliert und auf den Arduino überträgt. Ein weiterer Pluspunkt von ArduBlock ist die Kommentarfunktion, die sich ein und ausblenden lässt. Diese Dokumentationsfunktion erleichtert besonders jüngeren Schülern die Einführung und Einarbeitung in die Struktur dieser Programmiersprache. Trotz all dieser Vereinfachungen ist ArduBlock aber immer noch viel näher an der eigentlichen Arduino-Programmiersprache dran als die Programmierumgebung Scratch.

Ausblick

In einem kreativen Prozess entwickeln wir zusammen mit unseren Schülern unseren Roboter laufend weiter. Neuartige, innovative Ideen fließen ständig ein, so wollen wir den R4e1 zukünftig mit einem Bluetooth-Modul erweitern und mit einer selbst entwickelten Smartphone-App steuern können. Diese befindet sich bereits in der Entwicklungsphase.



Mit ArduBlock kann man den Arduino und seine Sensoren grafisch programmieren.



Da der Arduino mit der Versorgung der Servos überfordert ist, benötigt man einen separaten 5V-Regler.

Leider machen es einem die Rahmenbedingungen dabei nicht einfach. So sind etwa die Vorgaben der Stadt Hannover für Anschaffungen sehr restriktiv, wenn nicht gar weltfremd: Online-Einkäufe sind beispielsweise vollständig verboten, ein Kauf im Ausland sowieso undenkbar. Die Gründung einer Schülerfirma zur Produktion unseres Roboters für interessierte Schulen wäre mit dieser Voraussetzung unmöglich. Dies ist sehr traurig, denn die verantwortlichen Kommunen und Städte sollten eigentlich praxisnahen Unterricht fördern und nicht nur Steine in den Weg legen. Glücklicherweise können wir alle Einkäufe über das Konto des Fördervereins abwickeln und im Sommer 2015 wird die Schülerfirma mit der Produktion des R4e1 beginnen.

Ein Download der Druckdatei ist aber schon jetzt unter <http://r4e1.igsmuehlenberg.de/> möglich. Später werden auf dieser Seite Arbeitsblätter für den Unterricht folgen, eventuell auch Bestellmöglichkeiten für einzelne Bauteile oder ganze Bausätze des R4e1. — *dab*

↓ Links und Foren
make-magazin.de/xtf9

Gameboy Pocket

Dieser Minihandheld ist eine ganze Spielekonsole.
Wir zeigen, was in ihm steckt und welches Potenzial
für Spielereien er noch bietet.

von Rebecca Husemann



Der Gameboy ist der bekannteste batteriebetriebener Handheld der Neunziger. Die kompakte Variante des Klassikers ist der Gameboy Pocket: 20 Prozent kleiner als der Vorgänger, dafür ein größerer Bildschirm mit 65 Millimetern Bildhöhe. Und trotzdem passt alles rein. Wir haben uns das mal angeschaut. Das Geheimnis hinter der Platzersparnis: Im Gameboy Classic ist Leerraum, der beim Pocket optimal genutzt wurde. Die ganze Elektronik passt jetzt auf eine Platine. Viel Platz wurde außerdem bei den Batterien gespart. Statt zwei AA-Batterien benötigt er nur zwei AAA-Batterien für zehn Stunden Spielzeit. Außerdem stecken in ihm eine Menge Möglichkeiten, sich kreativ auszutoben. Eine Anleitung finden Sie online (siehe Downloadlink).

Der Bildschirm des Pocket bildet Graustufen kontrastreicher ab als der Vorgänger. Besonders gut erinnere ich mich an die Nächte, in denen ich heimlich unter der Bettdecke

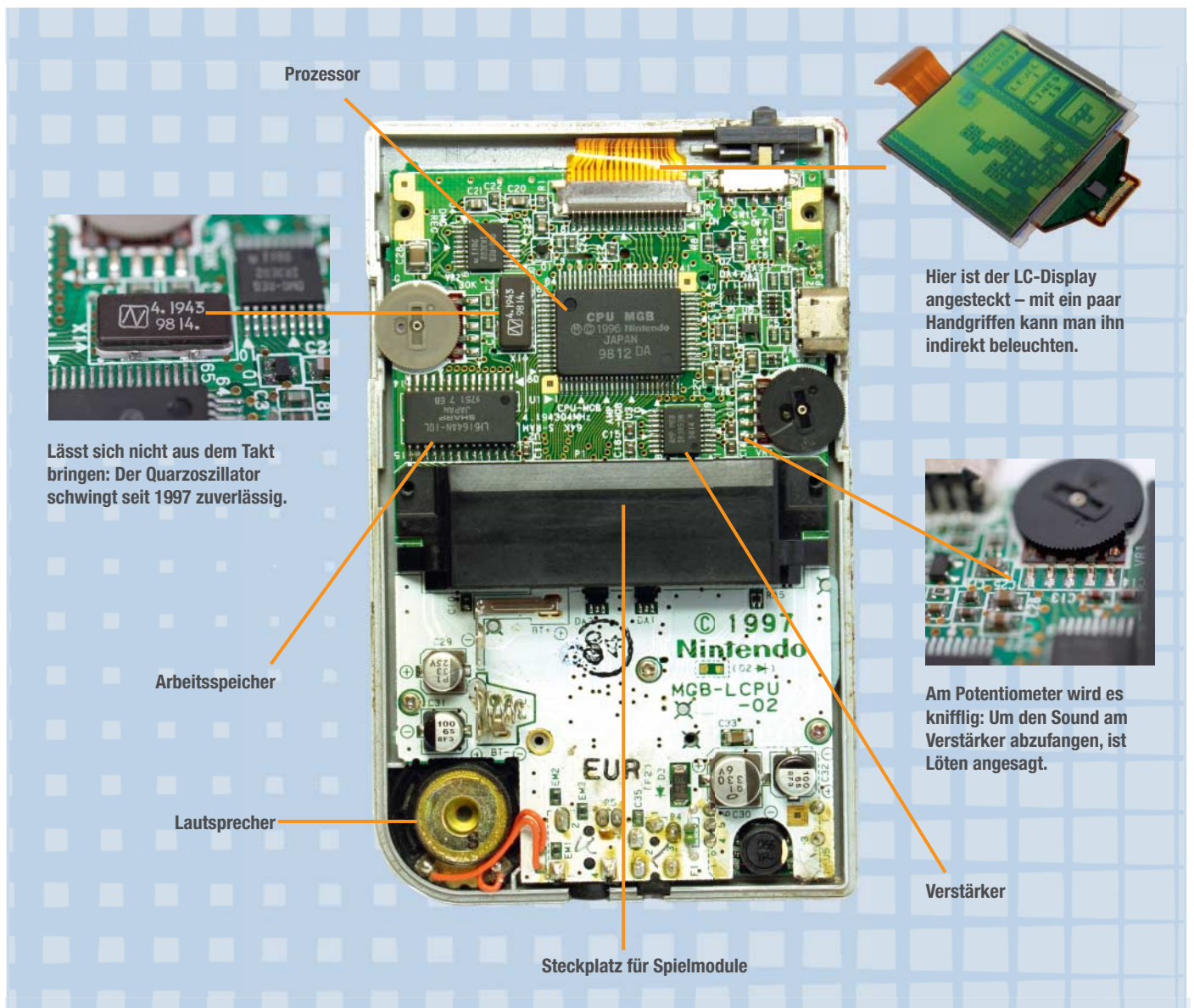
Gameboy gespielt habe – mit der Taschenlampe. Abhilfe schaffen LED-beleuchtete Displays in verschiedenen Farben. Diese lassen sich leicht einbauen: Der LC-Display des Pockets besteht aus mehreren Schichten. Die oberste löst man vorsichtig vom Glas. Diese Folie ist reflektierend und würde das Licht der LEDs nicht durchlassen. Die unterste Schicht ersetzt der neue Leuchteinsatz. Diesen lötet man in den Stromkreislauf direkt unter dem Bildschirm ein. Jetzt können wir auch im Bett Tetris spielen.

Auf der Vorderseite des Motherboards sitzt ein Quarzoszillator. Er erzeugt Schwingungen und deren Frequenz bestimmt die Spielgeschwindigkeit des Gameboys. Der Gameboyoszillator läuft mit 4,19 Megahertz. Wenn man stattdessen einen Taktgeber mit einer höheren Frequenz einlötet, wird der Gameboy schneller. So kann man zum Beispiel Spiele mit höherer Geschwindigkeit spielen. Dabei verändert sich auch die Ton-

höhe – spannend für alle, die mit dem Gameboy auch Musik machen möchten.

Der unverwechselbare Klang eines Gameboys entsteht durch Prozessor, Verstärker und Potentiometer. Um ihn wie ein Musikinstrument zu benutzen, braucht man ein spezielles Steckmodul. Das kann man sich im Internet bestellen: Die (meist freie) Software zum Musizieren ist schon vorinstalliert. Damit das gut klingt, lohnt es sich, einen Klinkenanschluss einzulöten. Bei dieser Methode fängt man die Musik am Potentiometer ab und erhält den rohen, unverstärkten Sound. Dafür lötet man die Klinkenbuchse an den Audioausgang auf der Hauptplatine. Schließt man den Gameboy mit dem Klinkenstecker an einen Verstärker an, hat die Musik den optimalen Klang. —*rehu*

Links und Foren
make-magazin.de/xqu6



„Selber machen kann jeder!“

Warum ist es gut, an mehreren Projekten gleichzeitig zu arbeiten? Und warum steht die deutsche Maker-Szene erst am Anfang? Ein Interview mit Mario Lukas.

Interview: Daniela Metz

Bild: Mario Lukas



Bild: VDI Technologiezentrum, Frank Nürnberg

Mario mit Prof. Dr. Johanna Wanka (Bundesministerin für Bildung und Forschung) auf dem Photonik-Kongress in Berlin im Februar 2014.



Bild: Daniela Metz

Den LED-Arcade-Tisch stellt Mario Lukas auch im Interview mit Martin Holland von c't uplink vor (siehe Link am Ende des Artikels).

Bild links: Zur Inspiration und zum Austausch über seine Projekte reist Mario Lukas im Mai zum zweiten Mal zur Maker Faire in die Bay Area.

Mario Lukas ist immer beschäftigt. Der 34-jährige Informatiker aus Aachen bastelt, schraubt und programmiert – oft an mehreren Projekten im Wechsel. Ob intelligenter Getränkeuntersetter, Klopapier-Printer oder der selbst konstruierte Stereolithografie-3D-Drucker Open Exposer, er baut es einfach und schreibt im Internet darüber. Für eine Veranstaltung setzte er spontan einen Raspberry Pi, eine Kinect-Kamera und einen 3D-gedruckten Griff zu einem mobilen 3D-Scanner zusammen.

Mit dem holografischen Display namens Lumicator hat Mario Lukas gerade zum zweiten Mal in Folge den Wettbewerb „Make Light“ gewonnen, den das Bundesministerium für Bildung und Forschung (BMBF) ausgeschrieben hat. Im Rahmen der gleichnamigen Initiative konnten dort Maker aus ganz Deutschland ihre Eigenbauprojekte rund um das Thema Licht einreichen.

Make: Warum und wie bist du Maker geworden? Was war dein erstes Projekt?

Mario Lukas: Bastler war ich eigentlich schon immer. Der Begriff Maker wurde ja erst in den letzten Jahren, hauptsächlich in den USA, geprägt und findet mittlerweile auch in Europa Beachtung. Lediglich die Art meiner Basteltätigkeit hat sich verändert. Wo ich früher noch mit Feile und Säge meine Prototypen aus Holz und andere Materialien gebaut habe, nutze ich heute den 3D-Drucker, Lasercutter oder eine CNC-Fräse, um mir Teile zu fertigen. Das ist toll, denn es geht schneller und präziser. Hinzu kommt, dass von der Idee bis zum ersten Prototyp nicht mehr so viel Zeit vergeht.

Meiner Meinung nach trägt das Internet auch zur Prägung des Maker-Begriffs bei. Ich glaube, ein Grundgedanke ist es, die eigenen Projekte mit anderen zu „teilen“ und so zum Nachbau anzuregen.

Es ist deshalb schwierig, mein erstes Projekt zu benennen. Im Jahr 2011 gab es in der c't den Wettbewerb „Mach flott den Schrott“. Es ging darum, etwas Kreatives aus Teilen vom Schrott oder Sperrmüll zu basteln. Dafür habe ich damals einen Drucker gebastelt, der auf Toilettenpapier Nachrichten-Feeds aus dem Internet druckt. Der Drucker war auch eins der ersten Projekte, das ich detailliert im Netz dokumentiert habe.

Make: Woran arbeitest du gerade?

Lukas: Der Open Exposer, mein Laser-3D-Drucker, ist momentan mein Hauptprojekt, mit dem ich beim Hackaday Prize übrigens in die Top 50 gekommen bin. Der Drucker arbeitet nach dem Stereolithografie-Prinzip. Das ist ja das älteste Verfahren des 3D-Drucks, bei dem mit einem UV-Laser flüssiges Harz ausgehärtet wird. Der Laser arbeitet punktuell und härtet

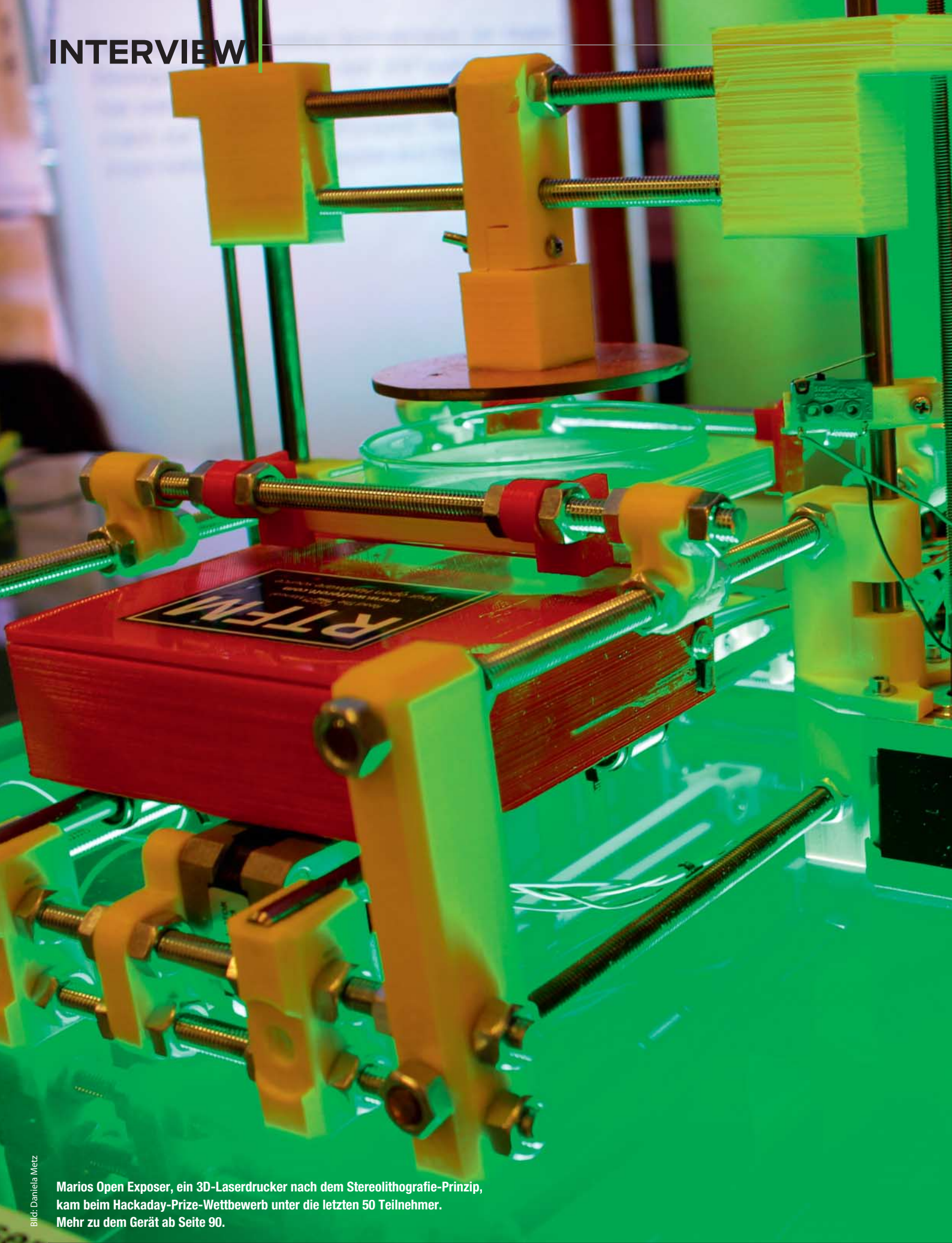


Bild: Daniela Metz

Marios Open Exposer, ein 3D-Laserdrucker nach dem Stereolithografie-Prinzip, kam beim Hackaday-Prize-Wettbewerb unter die letzten 50 Teilnehmer. Mehr zu dem Gerät ab Seite 90.

Schicht für Schicht, sodass das Objekt langsam aufgebaut wird. Die meisten Patente sind jetzt irgendwann gefallen, daher ist die Technik relativ einfach zugänglich. Aber die meisten 3D-Laserdrucker, die in dem Bereich gebaut werden, arbeiten mit Laser-Galvanometern, kleinen Motoren, die die Spiegel stellen. Wenn man da einen ordentlichen haben möchte, muss man schon 100 bis 200 Euro pro Stück bezahlen.

„Mehr Unterstützung aus der Industrie wäre eine gute Idee.“

Ich setze da ganz bewusst auf was anderes. Ich baue nur Teile ein, die man entweder günstig kaufen kann oder die man in einem alten Laserdrucker findet und ausbauen kann. Das ist der Trick an der Sache. Und ich wollte ein Tischgerät bauen, also keine „Umrüstung“ eines großen DLP-Projektors. Das funktioniert ganz gut.

Noch habe ich das kleine Problem mit dem Gerät, dass der Druck nicht so richtig an der Bauplattform haftet. Aber ich habe mich auf dem letzten Dorkbot-Treffen mit einigen Leuten ausgetauscht, die mir verschiedene Materialien geliefert haben, sodass ich die Sache jetzt umkonstruieren konnte. Gerade teste ich die Haftungseigenschaften. Den Drucker gab es letztes Jahr auch schon auf mehreren Maker Faires zu sehen, unter anderem in Hannover. In diesem Jahr möchte ich das neue Modell vorstellen.

Und dann arbeite ich noch am 3D-Scanner FabScan. Das ist so eine Herzensangelegenheit von mir. Zum einen schreibe ich die Software neu und erweitere sie, zum anderen möchte ich den FabScan von der Scanqualität her verbessern.

Der Open Exposer und der FabScan sind momentan die beiden Hauptprojekte, es kommen natürlich immer wieder andere Sachen hinzu – wie beispielsweise für den „Make Light“-Wettbewerb. Den Lumicator habe ich ja eigens dafür gebaut.

Make: Und warum so viele Projekte gleichzeitig?

Lukas: Die Leute fragen mich immer, ob ich nicht einen Schaden habe, weil ich so viele Projekte gleichzeitig mache (lacht). Aber ich meine, dass das so sein muss,

denn wenn man permanent nur an einer Sache arbeitet, verliert man ja irgendwann die Lust. Ich lege dann einfach ein Projekt zur Seite und nehme das nächste und arbeite eine Woche später dann wieder an dem anderen weiter. Das ist aber scheinbar gang und gäbe, das habe ich in Hannover letztes Jahr festgestellt, als ich mich mit anderen Leuten auf der Messe unterhalten habe.

Aber ich habe auch schon viele Leute begeistern können, wie zum Beispiel meinen Kollegen Christoph, der mit mir auf der Maker Faire 2014 in Hannover war. Am Anfang hat er mit Elektronik gar nichts zu tun gehabt, er kommt aus der Software-Branche. Als ich ihm dann gezeigt habe, wie einfach es mit einem Arduino geht, sind gleich schon als erstes größeres Projekt die LED-Arcade-Tische entstanden. Wir haben dann auch noch den Highstriker zusammen gebaut, einen modernen „Hau den Lukas“, der über das Smartphone der Besucher gespielt wird und die Ergebnisse virtuell anzeigt. Die Software dafür hat Christoph zum größten Teil geschrieben. Und dann haben wir für die Mini Maker Faire in Kerkrade eine Cocktailmaschine gebaut.

Make: Und zuletzt hast du mit dem Lumicator das zweite Mal in Folge den „Make Light“-Wettbewerb gewonnen.

Lukas: Der Lumicator ist ein Gerät, mit dem man über Video kommunizieren kann. Aber ich wollte die Sache auf die nächste Ebene heben und ein bisschen den Sci-Fi-Faktor mit reinbringen – wie bei Star Wars. Dann dachte ich mir, dass es noch cool ist, wenn man weitere Features einbaut. Zum Beispiel die Steuerung von Google Maps über Gesten in Form eines virtuellen Globus. Oder die virtuelle Feuerstelle, die man mit dem Smartphone ein- und ausstellen kann.

Make: Und wie funktioniert der Lumicator?

Lukas: Es sind zwei Sachen, die ich da miteinander vereint habe. Einmal das Fogscreen-Display, das es im Netz schon ganz häufig gibt. Das zweite ist „IKEA-Hacking“, das ich übrigens total klasse finde. Das musste ich irgendwie miteinander verbinden.

Ich gebe zu, ein echtes 3D-Hologramm ist es nicht, es ist eigentlich gar kein Hologramm. Der Effekt wird aber gigantisch, indem ich den Hintergrund „wegschneide“:



Bild: Daniela Metz

Für den Open-Source-3D-Scanner FabScan schreibt Mario Lukas die Software. Wie man das Gerät selber aufbaut, steht in c't Hacks 2/13.

Make Light

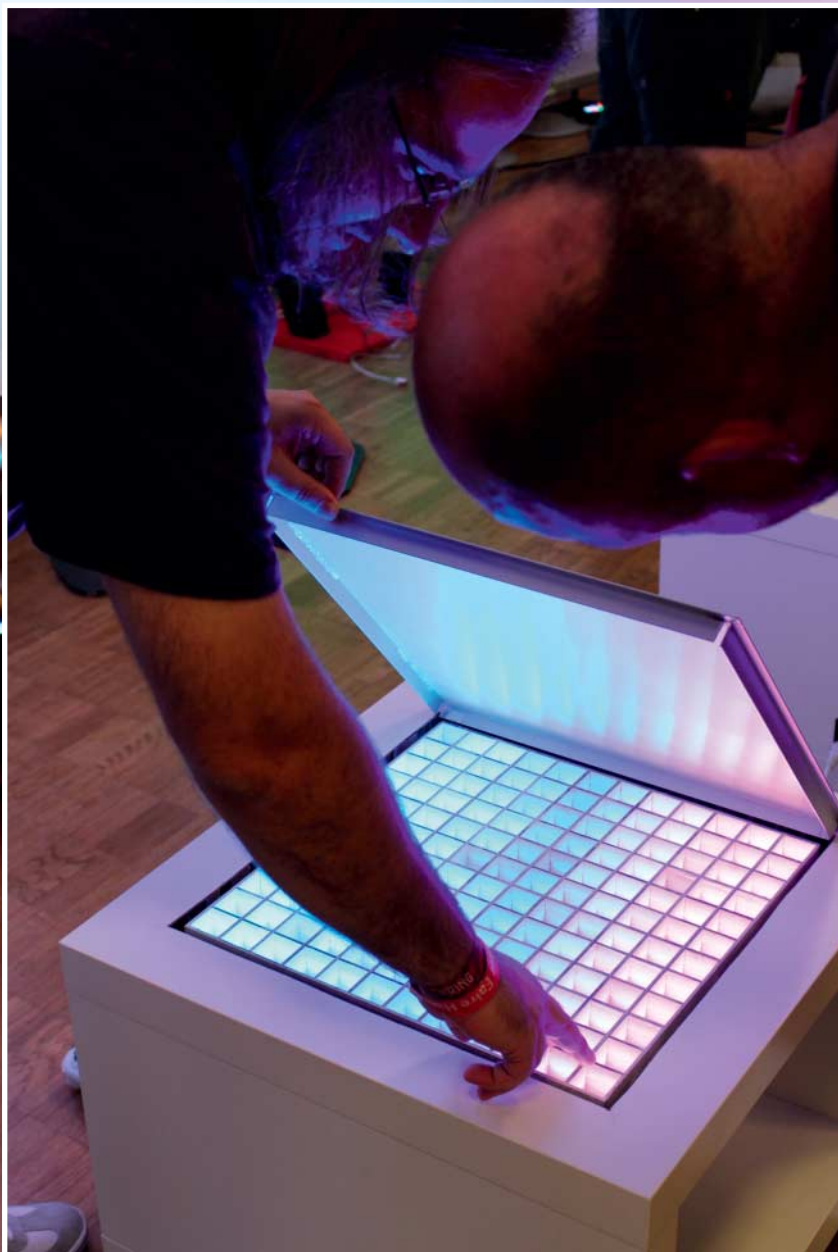


Bild: Daniela Metz

Der LED-Arcade-Tisch von Mario und Christoph ist ein IKEA-Hack. Auf ihm lässt sich beispielsweise Tetris spielen.



Eine Software macht ihn schwarz, man sieht nur die freigestellte Person. Das wirkt durch das Flackern des Bildes, das auf feine Tröpfchen im Luftstrom projiziert wird, schon toll. In Kerkrade auf der Mini Maker Faire habe ich das Ding das erste Mal ausgestellt, um zu schauen, wie die Leute darauf reagieren. Es war toll, wie sie dann da standen, rein griffen und suchend drum herum gingen. Es ist nicht gleich offensichtlich, dass da irgendwo ein Projektor ist.

Make: Wie geht es mit dem Lumicator weiter?

Lukas: Ich möchte ihn natürlich gerne in diesem Jahr auf der Maker Faire in Hannover vorstellen – und dann auch mit zwei Geräten. Weil ich für den Wettbewerb nur ein begrenztes Budget hatte und der Lumicator ein größeres Projekt ist, konnte ich nur ein Gerät bauen. Die Kommunikation von Gerät zu Gerät konnte ich so leider nur simulieren. Inzwischen haben mich aber schon ganz viele Leute nach dem Lumicator gefragt und wollen das Ding sehen.

Neben der Maker Faire in Hannover ist in diesem Jahr ja dann auch noch das Chaos Communication Camp bei Berlin im August. Das ist ähnlich wie ein Festival, nur fünf Tage Camping mit Hackern, eine gigantische Spielwiese. Ich möchte neben dem Lumicator einige andere Projekte, den Highstriker und die LED-Arcade-Tische mitnehmen.

Make: Um im Juni in Hannover dabei sein zu können, musst du allerdings mit einer langjährigen „Rock am Ring“-Tradition brechen. Wie hart ist das?

Lukas: (grinst) Das war schon ein Schlag ins Gesicht, als ich gesehen habe, dass die Maker Faire zum gleichen Zeitpunkt stattfindet. Ich bin ja ein leidenschaftlicher Festivalgänger, von Wacken bis Rock am Ring. Wenn irgendwo ein Festival ist, das mich

anspricht, dann bin ich auch da – und das schon seit Jahren. Rock am Ring habe ich bis jetzt seit 17 oder 18 Jahren jedes Mal mitgenommen. Aber momentan bin ich so viel am Basteln und die Maker Faire kann ich nicht auslassen. Das wäre schon ein Frevel. Und mittlerweile ist so ein Hotelzimmer auch schon angenehmer als ein Zelt ...

Make: Dein Preis beim „Make Light“-Wettbewerb ist der mehrtägige Besuch einer internationalen Maker Faire. Du hast direkt gesagt, dass du wieder zur Bay Area Maker Faire (16. und 17. Mai bei San Francisco) möchtest. Warum?

„Ein Grundgedanke ist es, die eigenen Projekte mit anderen zu ‚teilen‘ und so zum Nachbau anzuregen.“

Lukas: Der erste Grund ist, dass es mir letztes Jahr unglaublich gut gefallen hat. Meine großen Projekte kann ich ja leider schlecht mitnehmen, aber wenn man dann mit den Leuten dort ins Gespräch kommt, kann ich ihnen ein paar Bilder und Videos auf meinem Tablet zeigen.

Aber es ist nicht nur die Maker Faire, die mich da interessiert. Die Bay Area Maker Faire liegt ja auch direkt im Silicon Valley und da habe ich natürlich ganz viel Sachen vor, ich will zu verschiedenen Firmen einen längeren Ausflug machen. Zudem würde ich auch gerne in den Hackerspace Noisebridge in San Francisco, um einen Workshop mit FabScans machen. Ich habe bei Mitch Altman, einem Mitbegründer dieses Hackerspaces angefragt, ob ich das machen könnte. Er findet das gut – und für mich wäre das eine ziemlich coole Sache,

weil der FabScan noch nicht so richtig über den großen Teich hinaus ist.

Ich schreibe auch gerade eine neue Software für den FabScan, die das Augenmerk auf die Community legt, damit man die Scans auch untereinander austauschen kann. Ich möchte, dass man daraus ein „Internet of Things“-Gerät macht, das dann direkt an die Cloud oder ans Netz angebunden ist.

Make: Das BMBF unterstützt die Maker-Szene in Deutschland im Rahmen der „Make Light“-Initiative. Wie wichtig findest du diese Unterstützung?

Lukas: Das ist absolut wichtig! Die Unterstützung, die vom BMBF kommt, finde ich total Wahnsinn. Das ermöglicht mir Sachen zu machen, die ich sonst aus finanziellen Gründen nicht – oder nicht in so kurzer Zeit – machen könnte. Ich habe ja schon andere Projekte mit Hilfe des BMBF umgesetzt, zum Beispiel den 3D-Laserdrucker Open Exposer, an dem ich immer noch arbeite. Das war ein größerer Kostenfaktor, den ich nicht so ohne weiteres zahlen konnte. Ein gutes Oszilloskop schafft man sich halt nicht an, wenn man es nicht auch wirklich braucht. Ich finde es super, dass es die Make-Light-Initiative gibt. Und ich freue mich, wenn ich Leute mit meinen Projekten begeistern kann.

Make: Muss noch mehr getan werden im Bereich Förderung und Unterstützung?

Lukas: Wenn man eine Idee hat und etwas bauen möchte, dann ist es schon sehr wichtig, dass man irgendwo Unterstützung bekommt. Ich glaube, in den USA ist das schon etwas leichter. Kommerzielle Firmen unterstützen die Szene mehr, sie machen beispielsweise Veranstaltungen oder Workshops. Das ist in Deutschland noch zu wenig. Mehr Unterstützung aus der Industrie wäre eine gute Idee, wenn die sagen „Hey, wir haben hier ein paar Komponen-



Dale Dougherty besuchte den Make-Light-Stand auf der Maker Faire Hannover 2014 und ließ sich von Mario und Christoph ihre Projekte zeigen. Er probierte den Highstriker mit seinem Smartphone auch direkt aus.

ten, ein paar Samples. Die schicken wir euch zu. Macht mal was Tolles draus!“.

Make: Wo steht die Maker-Bewegung in Deutschland gerade?

Lukas: Darüber habe ich schon mit ganz vielen Leuten auf verschiedenen Messen gesprochen. Ich glaube, dass die Maker-Bewegung hier noch nicht so richtig angekommen ist. Die Leute wissen oft gar nicht richtig, was das ist – vielleicht ist das für einige aber auch noch zu neu. Das merkt man ja schon am Wachstum der Maker Faire Hannover. Im ersten Jahr waren etwa 4300 Leute da, ein Jahr später mehr als doppelt so viele. Und wenn man sich dann auch mal die Entwicklung in

Rom anschaut, wo 2014 über 70 000 Besucher zur Maker Faire gepilgert sind, dann glaube ich, dass das in Deutschland noch in den Kinderschuhen steckt und in den nächsten Jahren definitiv noch größer wird.

Ich glaube aber auch, dass es schon immer Maker gab. Der Maker ist ja eigentlich der Bastler. Mein Paradebeispiel dafür ist folgendes: Man muss sich nur anschauen, wie die Menschen vor der Wiedervereinigung in der DDR gelebt haben. Not macht erfinderisch. Die haben sich aus allem etwas gebaut und haben sich aus den Sachen, die sie bereits hatten, etwas Neues gemacht, das dann den anderen Leuten zur Verfügung gestellt, die wieder-

um etwas Neues daraus gemacht haben. Das ist ein super Beispiel für die Maker-Bewegung, das ist „Makertum“.

Make: Wie siehst du die Entwicklung der deutschen Maker-Szene im Vergleich mit der internationalen Maker-Szene?

Lukas: Die „internationale Maker-Szene“ ist hauptsächlich die USA. Die haben sogar schon eine Maker Faire im Weißen Haus gehabt! Das zeigt, dass die Idee dort in der Mitte der Gesellschaft angekommen ist. Die Regierung und die Industrie sind darauf aufmerksam geworden, sie haben gemerkt, dass sie Vorteile von den Ideen haben, die im Keller entstehen – und davon auch profitieren können.



Bild: Daniela Metz

Seine zahlreichen Bastelprojekte machen Mario Lukas zum gefragten Interviewpartner.

Ansonsten sind die Niederlande in Europa ein echter Hotspot, die sind uns ein wenig voraus. Da gibt es ganz viele Maker Faires und Mini Maker Faires. Ich selber fahre allerdings nur nach Kerkrade, da es direkt vor meiner Haustür liegt. Wenn es die nicht gäbe, gäbe es eine in Aachen, da bin ich mir ganz sicher (lacht).

Rom, London, Hannover und demnächst Berlin – in Europa geht es jedenfalls los. Übrigens: Ob Maker, Hacker oder sonst was, in den USA machen die da weniger Unterschiede. Hier in Deutschland hat man den Chaos Computer Club, seinen Hacker-Verein oder sein FabLab. In den USA wird nicht so stark separiert, das ist oft alles eins.

Make: Was rätst du allen, die sich für das ganze „Maker-Ding“ interessieren, sich aber nicht zutrauen, selber was zu machen?

Lukas: Grundsätzlich empfehle ich allen, die sich dafür interessieren, einen Besuch auf einer Maker Faire. Am besten auf einer größeren wie beispielsweise in Hannover. Danach sollte eigentlich jeder fasziniert und inspiriert sein! Es gibt mittlerweile auch sehr viele Bücher und Informationen in diversen Blogs. Selber machen kann eigentlich jeder, es geht ja nicht immer gleich um komplexe Themen wie Elektronik, 3D-Drucker und Fräsen. Wer sich dennoch mit Elektronik versuchen will, dem ist ein Arduino-Starter-Kit zu empfehlen. So habe ich im Prinzip meinen Weg auch gefunden.

Make: Vielen Dank Mario für das Gespräch und weiterhin viel Erfolg und Spaß mit deinen Projekten!
—phs

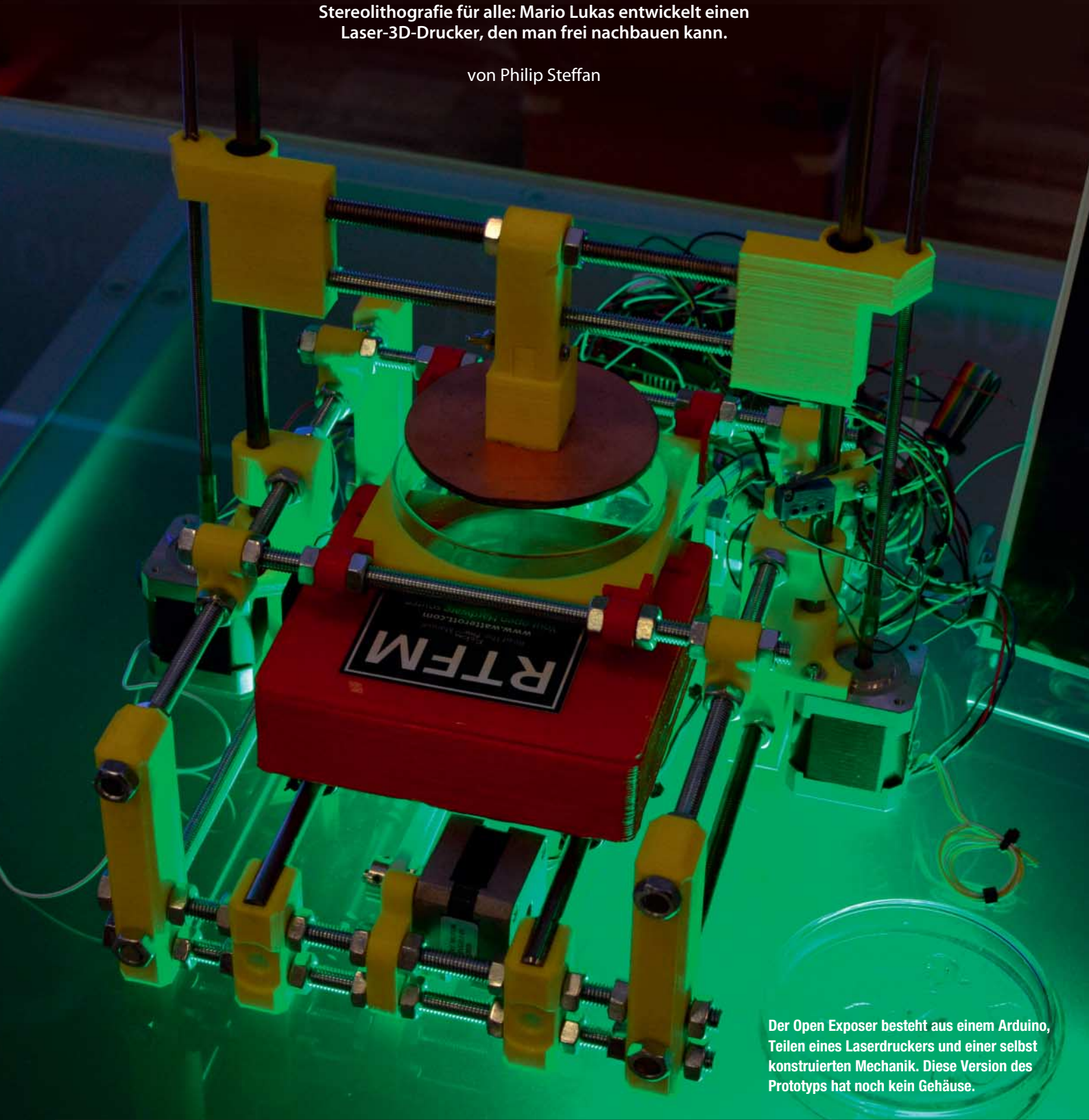


Links und Foren
make-magazin.de/xx61

Open Exposer

Stereolithografie für alle: Mario Lukas entwickelt einen Laser-3D-Drucker, den man frei nachbauen kann.

von Philip Steffan



Der Open Exposer besteht aus einem Arduino, Teilen eines Laserdruckers und einer selbst konstruierten Mechanik. Diese Version des Prototyps hat noch kein Gehäuse.

Eine der Eigenkonstruktionen von Mario Lukas, die man im vergangenen Jahr auf der Maker Faire in Hannover bewundern konnte, war sein 3D-Drucker Open Exposer. Die unscheinbare Maschine aus Gewindestangen und roten und gelben Kunststoffteilen macht mit einem Laser aus einem flüssigen Harz feste Kunststoffobjekte – bald jedenfalls, denn noch ist sie ein Prototyp. Das Verfahren nennt sich Stereolithografie und wurde in den 1980er Jahren von der Firma 3D Systems entwickelt.

Der Laser im Open Exposer funktioniert wie in einem herkömmlichen Laserdrucker für Papier: Ein rotierender Polygonspiegel erzeugt aus dem Lichtpunkt einer UV-Laserdiode eine Laserlinie. Motor- und Lasertreiber steuern die Drehzahl des Spiegels und die Einschaltzeit des Lasers, wodurch auf der Laserlinie beliebige Muster erzeugt werden können. Über einen Umlenkspiegel und einen Schlitz im Gehäuse belichtet diese Laserlinie ein UV-empfindliches Harz in einer Petrischale. Dort, wo das Laserlicht auftrifft, wird das Harz fest. Zwei Schrittmotoren bewegen die Lasereinheit in Y- und die Druckplattform in Z-Richtung, sodass die Laserzeile schichtweise ein Objekt in das Harz hinein belichten kann.

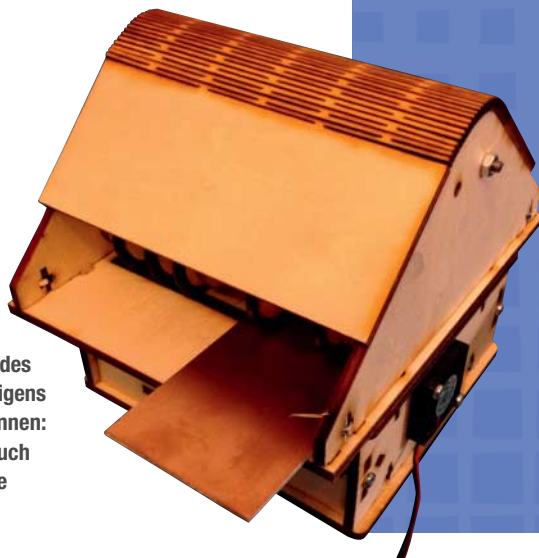
Nach diesem Prinzip arbeiten auch andere Stereolithografie-Drucker wie zum Beispiel der Form 1 (siehe Make: 1/2015, Seite 122). Im Unterschied dazu ist der Open Exposer allerdings, wie der Name schon andeutet, Open Hardware: Man darf ihn frei nachbauen und weiterentwickeln, genau wie beispielsweise die RepRap-3D-Drucker. Die Elektronik ist Arduino-kompatibel, die Firmware hat Mario Lukas, genau wie den Präprozessor für die Druckdaten, selbst geschrieben, das Gehäuse und die mechanische Konstruktion selbst entworfen.

Schon jetzt kann man sich am Nachbau eines Open Exposer versuchen: Die Software und die Gehäusepläne gibt es im Internet. Mario Lukas hofft auch auf Erfahrungswerte aus der Community: Noch haftet das auszu-druckende 3D-Teil nicht so gut am Drucktisch, außerdem gibt es die Polygonspiegel, die man sich am besten aus einem ausrangierten Laserdrucker ausbaut, mit verschiedenen elektronischen Ansteuerungen. Ein zukünftiger Arduino-Shield für den einfacheren Aufbau eines Open Exposer sollte natürlich so viele davon unterstützen wie möglich. —*phs*

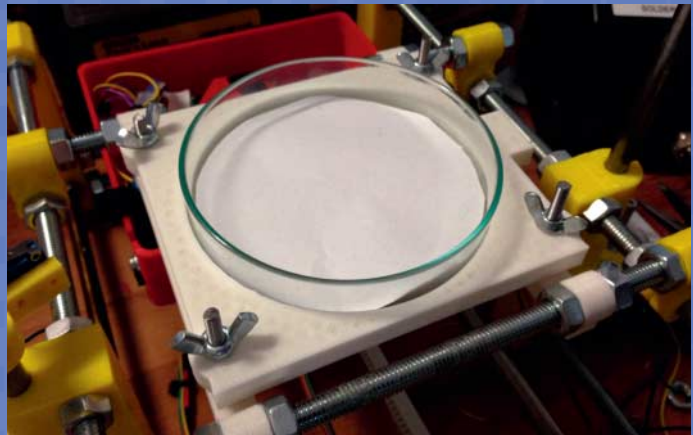
Mehr Informationen gibt es auf Mario Lukas' Webseite:

► www.openexposer.com

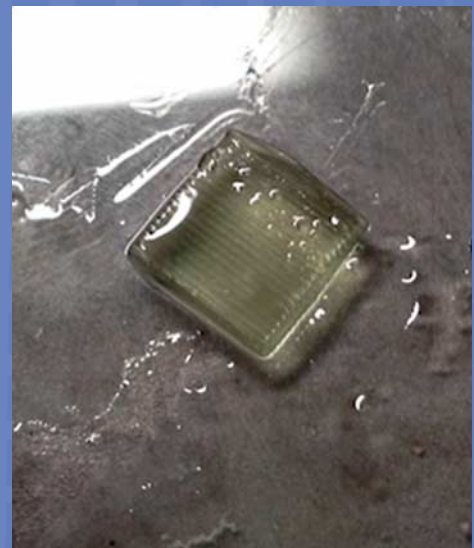
Mit der Belichtungseinheit des Open Exposer soll man übrigens nicht nur in 3D drucken können: Mit dem Laser kann man auch auch Platinen Zeile für Zeile belichten.



Der Polygonspiegel rotiert und erzeugt dadurch eine Laserlinie. Ein solches Modul lässt sich samt Steuerelektronik aus einem herkömmlichen Laserdrucker gewinnen.



In dieser Petrischale wachsen keine Bakterien, sondern 3D-Teile, die maximal neun Zentimeter Durchmesser haben können.



Die ersten Druckversuche mit dem Open Exposer: Da, wo der Laser auf das Harz trifft, wird es fest. Ein Liter der Flüssigkeit kostet rund 60 Euro.

Captain Herrmano's Mystery Box

Dieses Geburtstagsgeschenk gibt sein Geheimnis nicht so schnell preis: Die Arduino-getriebene Schatzkiste stellt Rätselfragen und öffnet ihren Tresor nur am richtigen Ort.

von Philip Steffan



Die Schatzkiste von Raffael Herrmann enthält kein Gold, sondern den digitalen Geist von Captain Herrmano, der über ein Display immer neue Rätsel stellt. Unter der Haube stecken ein Arduino und zahlreiche Sensoren.

Die elektronische Schatzkiste des Piraten Captain Herrmano zeigt, wie viel Mühe man sich mit selbst gebauten Geburtstagsgeschenken geben kann. Der Tüftler Raffael Herrmann hat sich dafür vom „Reverse Geocache Puzzle“ inspirieren lassen, das Mikael Hart 2009 auf seinem Blog beschrieben hat: Im Gegensatz zum üblichen Geocaching muss man kein Objekt an einem bestimmten Ort finden, sondern ein Objekt zu gewissen Koordinaten bringen, um bei der Schnitzeljagd weiterzukommen.

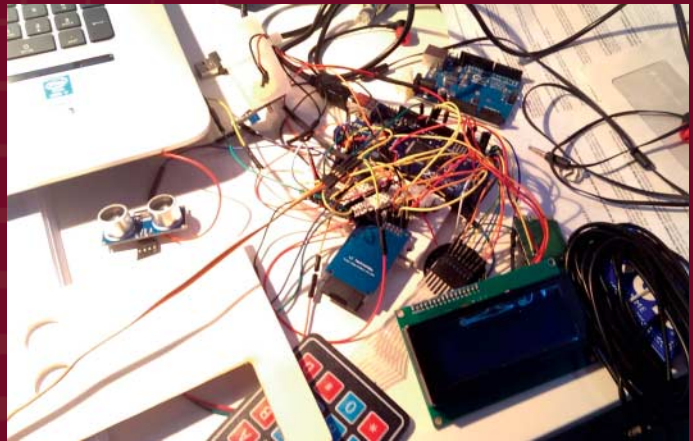
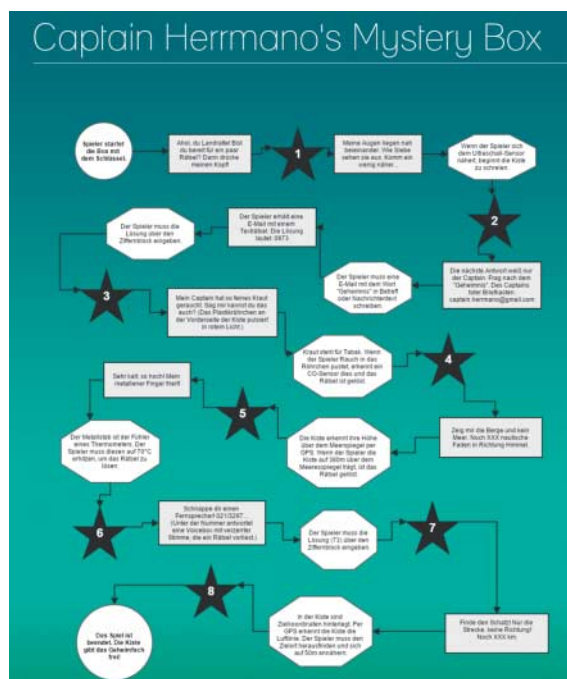
In der „Mystery Box“, ursprünglich eine rustikale Dekorationskiste, sorgt ein batteriebetriebener Arduino Mega mit GPS-Modul für diese Funktion. Der Beschenkte muss mit der Kiste tatsächlich durch die Gegend fahren, um ihr Geheimnis zu knacken. Raffael Herrmann hat sich allerdings noch diverse weitere Rätsel ausgedacht und elektronisch implementiert. Über das vierzeilige Display formuliert „Captain Herrmano“ immer neue Puzzlefragen und Wünsche, die man verstehen und umsetzen muss, um zum nächsten Schritt zu kommen.

Da der Captain seinen Tabak vermisst, muss man an einer Stelle Zigarettenrauch in eine Öffnung blasen – ein CO₂-Sensor am Arduino schlägt darauf an. Friert der Pirat, geht es erst weiter, wenn man seinen „Finger“ auf über 70 Grad erhitzt. Will er auf einen Berg, muss man die Kiste tatsächlich auf 380 Meter über den Meeresspiegel tragen. Auch diese Information liefert das GPS-Modul. Nicht alle Puzzles stecken in der Kiste: Will man das Rätsel der Schatztruhe lösen, muss man auch eine E-Mail schicken und eine angezeigte Telefonnummer anrufen. Eine automatische Antwort und ein Anrufbeantworter liefern dann den jeweils nächsten Schritt.

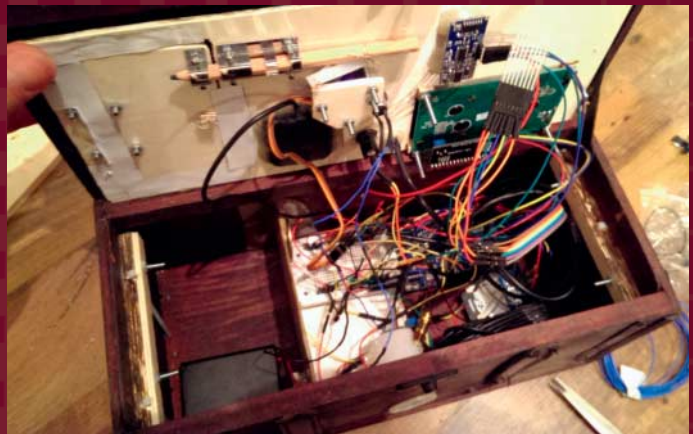
Nähert man sich beim letzten Rätsel den richtigen Koordinaten, öffnet sich in der Kiste eine kleine Klappe, hinter der sich das echte Geschenk verbirgt. Das Schloss besteht aus einem Servomotor, Teilen eines Scharniers und – Bassteil macht erfinderisch – einem halben Bleistift.

Mehr Informationen und Videos gibt es in Raffael Herrmanns Blog:

► www.code-bude.net/category/arduino



Am Arduino Mega hängen ein vierzeiliges Display, ein Tastenfeld, ein Servomotor und GPS-, Temperatur- und Kohlendioxid-Sensoren.



Bei einem Einzelstück wie diesem ist die fliegende Verkabelung kein Problem. Links das Geheimfach, das sich am Ende der Reise öffnet.



So einfach konstruiert man ein elektronisches Schloss: Servomotor, Metallteile von einem Türscharnier und ein zufällig herumliegender Bleistift als Riegel.

Acht Rätsel musste der Beschenkte in der richtigen Reihenfolge lösen, bis er am Ziel war.

GPS-Navigation mit Arduino

Vor wenigen Jahrzehnten noch undenkbar, trägt es heute fast jeder bei sich: Ein Gerät, das den Standort auf 10 Meter genau mitteilt. Mit einem preiswerten GPS-Modul und einem Arduino-Mikrocomputer lassen sich die Grundlagen der satellitengestützten Navigation nachvollziehen.

von Roland Pleger

Kurzinfo



Zeitaufwand:
1 Stunde



Kosten:
ab 45 Euro



Programmieren:
Arduino SDK



Löten:
ggf. Stiftleisten anlöten



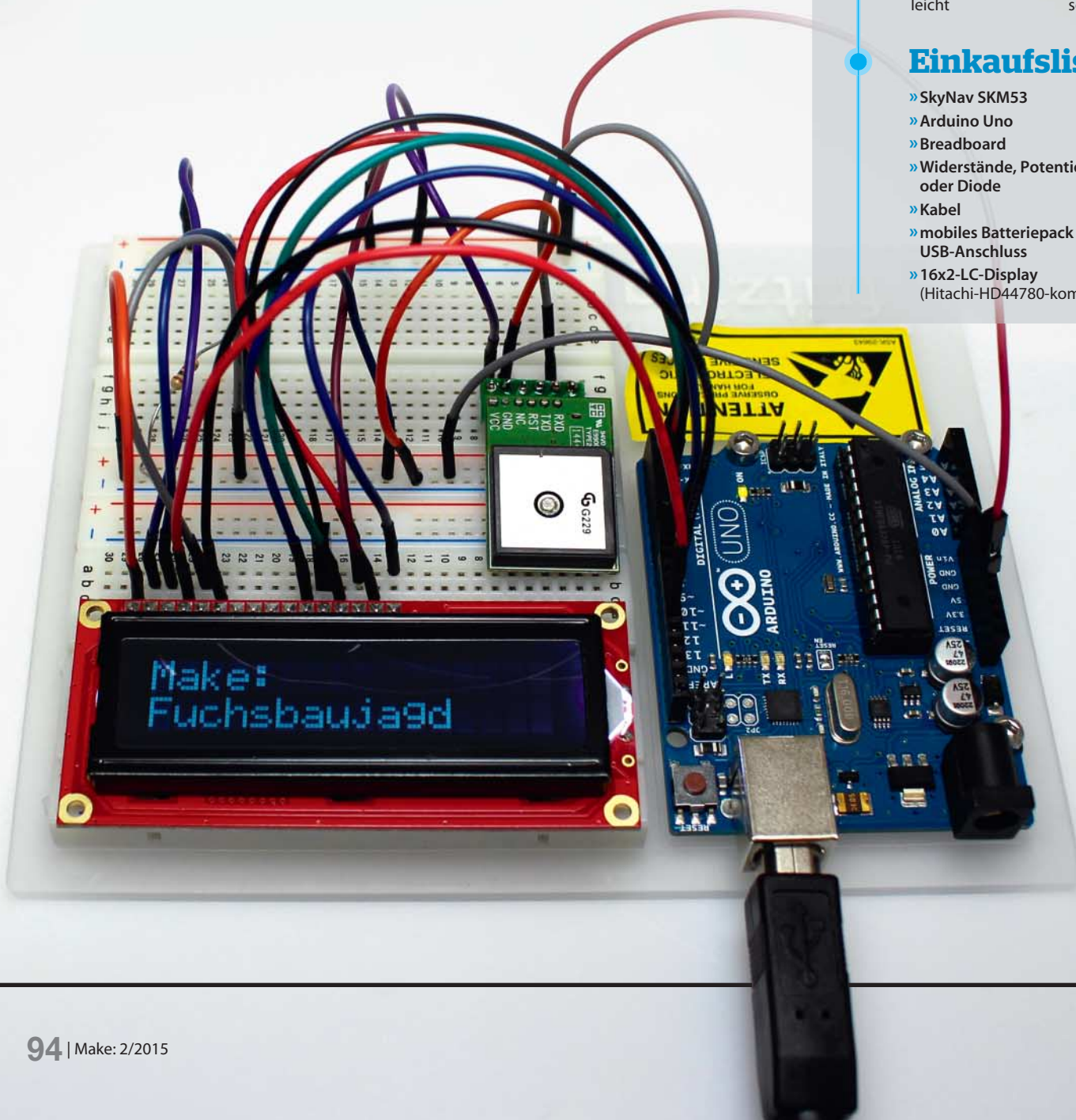
Elektronik:
Einsteigerprojekt

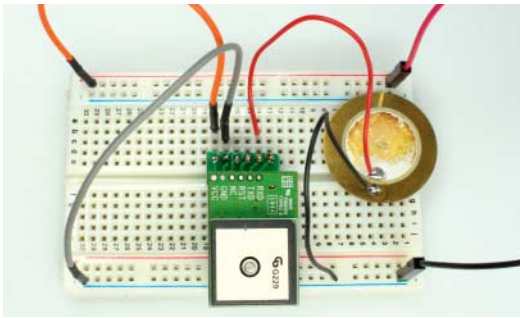
Schwierigkeitsgrad

leicht schwer

Einkaufsliste

- » SkyNav SKM53
- » Arduino Uno
- » Breadboard
- » Widerstände, Potentiometer oder Diode
- » Kabel
- » mobiles Batteriepack mit USB-Anschluss
- » 16x2-LC-Display (Hitachi-HD44780-kompatibel)





Der Chip lässt sich akustisch überprüfen, hier mit einem Piezoschallwandler.

Die folgenden Versuche basieren auf dem Modul SkyNav SKM53 der Firma Skylab mit dem GPS-Empfängerchip MediaTek 3329. Die Patch-Antenne dominiert das Design des Moduls und ist ein kleines technisches Meisterwerk bezüglich Baugröße und Antennengewinn. Eine Adapterplatine passt das Rastermaß der Kontaktstifte von 2 Millimeter auf das übliche Rastermaß von 2,54 Millimeter an. Je nach Ausführung muss die Platine noch angelötet werden. Grundsätzlich eignen sich auch andere GPS-Module mit serieller Schnittstelle. Einen weiteren Artikel über u-blox' Neo-Module finden Sie online, siehe Downloadlink.

Kommunikation mit der Außenwelt

Für die einfachen Versuche mit dem Chip reichen drei Leitungen: zwei für die Spannungsversorgung von 5 Volt (GND und VCC) und eine weitere für das Ausgangssignal (TX). Der Chip überträgt die Daten seriell mit einer Baudrate von 9600. Bereits ein hochohmiger Kopfhörer oder Verstärker genügt, um die Funktion zu überprüfen (siehe Bild oben). Alle Sekunde hört man ein Summen von 200 bis 500 Millisekunden Dauer.

Der Universal Asynchronous Receiver Transmitter (UART) ist für die Datenumwandlung in serielle Daten verantwortlich. Ein Datenwort beginnt mit



Der Chip in Originalgröße mit der Antenne rechts und dem Adapter links

einem Startbit, gefolgt von 8 Datenbits und einem Stopbit. Das Wort „asynchron“ in der Abkürzung UART ist wörtlich zu nehmen. Auch wenn die Daten meist als zusammenhängender Datenstrom übertragen werden, kann jederzeit nach dem Stopbit eine Pause beliebiger Länge auftreten.

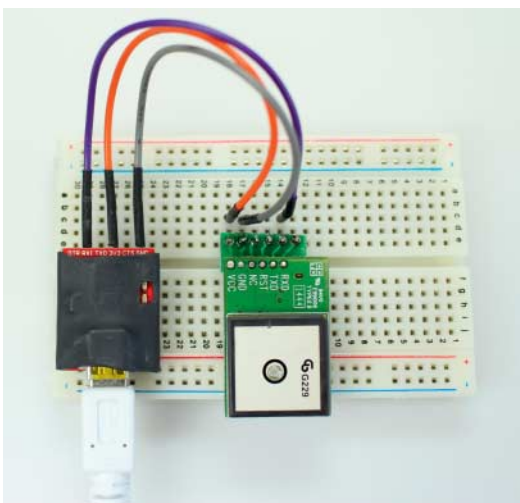
Um die Störsicherheit bei der Übertragung zu erhöhen, wird der UART eines Rechners üblicherweise über eine RS232-Schnittstelle angesteuert. Moderne Rechner halten nur noch USB-Schnittstellen vor. Hier hilft ein RS232/USB-Wandlerkabel. Das RS232-Protokoll legt unter anderem die Spannungspegel fest. Eine Spannung zwischen -15 Volt und -3 Volt wird als „logisch 1“ interpretiert, von +3 bis +15 als „logisch 0“. Der Spannungsbereich zwischen -3 Volt und +3 Volt bleibt undefiniert.

USB-Anschlüsse liefern nur eine Versorgungsspannung bis zu 5 Volt, es gibt aber zahlreiche Tricks, um daraus Hilfsspannungen von ± 12 Volt zu generieren. Beispiele sind Ladungspumpen, wie sie auch der Treiberchip MAX232 einsetzt, oder ein Zweckentfremden statischer RS232-Leitungen wie Signale für DTR oder CTS. Die meisten RS232/USB-Umsetzer legen den verbotenen Bereich großzügig aus: 3 Volt bis 15 Volt steht wie gehabt für eine logische 0. Aber die logische 1 beginnt bereits ab 0 Volt, nicht erst bei -3 Volt. In diesem Fall genügt ein Spannungsintervall von 0 bis 5 Volt, um die Signalpegel richtig weiterzugeben.

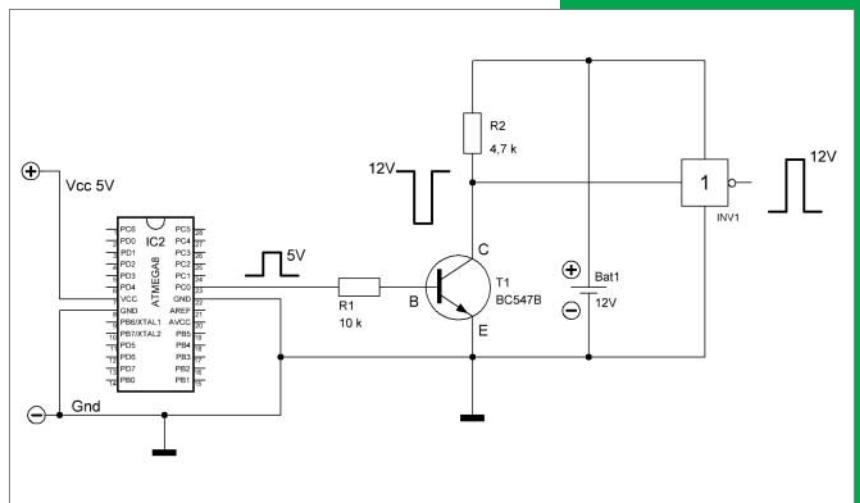
Da RS232 mit negativer Logik arbeitet, müssen die Pegel invertiert werden: kein Signal (0 Volt) wird zu einer Spannung größer 3 Volt, ein anliegendes Signal



Informationen werden digital durch 0 und 1 übertragen. Solch binär kodierte Daten lassen sich durch den Unterschied elektrischer Spannungen darstellen.



Einfach auslesen lässt sich das GPS-Modul mit einem FTDI-Adapter.



Mit einem Transistor wird die positive Kodierung von High = 1 und Low = 0 zur negativen vertauscht.

Python-Listing

```
import sys
import serial
import time
serial_port = "/dev/ttyUSB0" # Linux
rs232obj = serial.Serial(serial_port, 9600, 7
serial.EIGHTBITS, serial.PARITY_NONE, serial.STOPBITS_ONE)
time.sleep(2)
print "start"
while 1:
    onechar=rs232obj.read()
    if onechar == "$":
        sys.stdout.write('')
        sys.stdout.write(onechar)
```

(5 Volt) wird zu einer Spannung nahe 0 Volt. Um den Ausgang vor Leitungskurzschlüssen zu schützen, sollte eine einfache Verstärkerschaltung zwischengeschaltet werden. Beides erledigt eine Transistor-Emitterschaltung, wie sie in der c't Hacks 2/14 vorgestellt wurde. Leitung 5 des Sub-D Steckers wird mit der Masse verbunden, Leitung 2 mit dem Ausgang der Emitterschaltung. Und fertig ist die GPS-Maus.

Bei Linux ist die Treibersoftware für einen RS232/USB-Wandler bereits installiert. Windows-Nutzer erhalten die notwendigen Treiber meist mit dem Kabel. Ein Terminalprogramm wie Putty macht

NMEA-STANDARD

Die National Marine Electronics Association (NMEA) legt Standards für den Austausch von Positions- und Navigationsdaten fest. Wenn wir hier von NMEA-Daten sprechen, meinen wir das Ausgabeformat für die GPS-Daten. Das Format wurde lange vor GPS für Funkdienste in der Schifffahrt definiert und nimmt Rücksicht auf Ausgabegeräte mit geringer Rechenleistung.

Ein Dollar-Zeichen \$ am Anfang und ein Stern *, gefolgt von zwei Prüfzeichen, umschließen eine Datenzeile. Die Buch-

stabenfolge wie PMTK steht für Empfänger-Einstellungen. Beispielsweise bestätigt der Mediatec Chip mit \$PMTK010 und dem Code 001 das erfolgreiche Einschalten. Zeilen mit Navigationsdaten beginnen mit GP. Wir beschränken uns in den Beispielen auf die Codes GGA (Global positioning system fixed data), GSA (GNSS DOP and active satellites), GSV (GNSS satellites in view) und RMC (Recommended minimum specific GNSS data). Weitere Codes finden Sie im Internet (siehe Download-Link).

NMEA-GPS-Codes

\$GPGGA,093004.000,5052.0000,N,00710.0000,E,2,7,1.09,93.4,M,47.1,M,0000,0000*65

093004.000	9h30:04.000 UTC (Uhrzeit)
5052.0000,N,00710.0000,E	50° 52.0000' nördl. Breite und 7° 10.0000' westl. Länge
2	EGNOS-Korrektursignal wird empfangen. (1: ohne DGPS-Korrektur; 0: kein Empfang)
7	Anzahl empfangener Satellitensignale
1.09	Geometriefehler (HDOP): 1 ist ideal, 5 ausreichend, 10 schlecht
93.4,M	normale Höhe in Meter
47.1,M	Geoundulation, die zusammen mit Ellipsoiden zur normalen Höhe führt
0000,000	DGPS-Korrekturen
*65	Checksumme

\$GPGSA,A,3,24,12,15,14,25,06,04,,,,,1.83,1.09,1.47*03

A,3	automatische Auswahl 3D-Daten (Positionsdaten und Höhe)
24,12,15,14,25,06,04	PRN-Nummern von 7 Satelliten mit Signal
1.83,1.09,1.47	Geometriefehler (DOP), ausreichend, wenn kleiner als 5

\$GPGSV,3,2,12,33,27,208,20,17,22,043,,15,17,178,28,04,14,063,15*73

3,2	zweite von 3 Zeilen
12	Nummer der erfassen Satelliten (teilweise ohne Signal)
33,27,208,20	PRN-Satellitennummer, Elevation, Azimuth, S/N
17,22,043,,	PRN-Satellitennummer, Elevation, Azimuth, S/N = 0 (kein Empfang)
15,17,178,28	PRN-Satellitennummer, Elevation, Azimuth, S/N
04,14,063,15	PRN-Satellitennummer, Elevation, Azimuth, S/N

\$GPRMC,093004.000,A,5052.0000,N,00710.0000,E,0.00,54.71,180714,,,D*59

093004.000	9h30:04.000 UTS (angepasste UTC-Zeit)
A	gültige Daten (sonst V)
5052.0000,N,00710.0000,E	Breite und Länge
0.00	Geschwindigkeit des Empfängers
54.71	Richtung
180714	Datum 18.7.2014
,,,	magnetische Abweichungen
D	EGNOS-Korrektur (sonst A)

den Datenstrom sichtbar. Im Terminal von Mac oder Linux kann auch `screen` genutzt werden (etwa `screen /dev/ttyUSB0 9600`). Das Python-Programm (Python 2, siehe Listing) veranschaulicht, wie eine Hochsprache auf die Daten zugreift. Die Bibliothek `pySerial` (siehe Download-Link) muss installiert sein.

Während ein schnelles Terminalprogramm die Datenblöcke im Sekundentakt ausgibt, erscheinen die Daten bei einem langsamen Rechner mit langsamer Software kontinuierlich auf dem Bildschirm. Auffällig ist das Verhalten, wenn man den Datenstrom am GPS-Modul unterbricht. Es werden weiterhin Daten ausgegeben. Wenn das Leseprogramm nicht schnell genug ist, speichert es die Daten in einem Puffer zwischen. Die meiste Zeit geht es gut. Aber irgendwann läuft der Puffer über und es gehen unkontrolliert Daten verloren.

Wo bin ich: Koordinaten auf der Erde

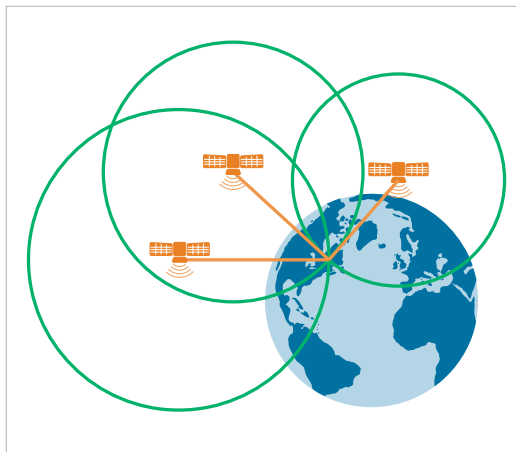
GPS ist die Abkürzung für NAVSTAR Global Positioning System, dem US-amerikanischen Satellitensystem zur Positionsbestimmung. Man sieht ihm an, dass es aus einer anderen Zeit stammt: In den siebziger Jahren des letzten Jahrhunderts dachte noch niemand daran, personenbezogene Daten in großem Umfang zu sammeln. Denn bei einem GPS-System weiß nur der Empfänger, wo er sich befindet. Heutzutage würde man einen Server zwischenschalten, der alles notiert. Beispielsweise gibt es kein Handy, das ausschließlich GPS-Empfang erlaubt. Immer ist die Positionsbestimmung an eine WLAN- oder Funkzellenortung geknüpft, die auf einen Datenbankserver angewiesen ist. Die Nutzung wird nur freigeschaltet, wenn man der Übertragung seiner Standortdaten zustimmt.

GPS arbeitet nach dem Verfahren der Multilateration (siehe Grafik rechts). Während bei einer Triangulierung Winkel zu Referenzpunkten bestimmt werden, misst eine Lateration Entfernungen. Um jeden Referenzpunkt denkt man sich eine Kugel mit dem Radius der gemessenen Entfernung. Der Beobachtungsort liegt im Schnittpunkt der Kugeloberflächen. Die Referenzpunkte sind Satelliten. Ihr Abstand ergibt sich aus der Signallaufzeit zum Beobachter. Bei einer Lichtgeschwindigkeit von 300 000 Metern pro Sekunde bedeutet eine Mikrosekunde Zeitfehler eine Entfernungsunsicherheit von 300 Metern. Satelliten messen ihre Zeit mit Atomuhren, deren Zeitfehler kleiner ist als die Messgenauigkeit.

GNSS-SYSTEME

GNSS ist der Oberbegriff für satellitengestützte Positionsbestimmung und steht für Global Navigation Satellite System. GPS bezeichnet das US-amerikanische GNSS-System. Das russische heißt Glonass. Viele Geräte empfangen Daten von beiden Systemen. Die Genauigkeit verbessert sich dadurch nicht, aber Glonass-fähige Geräte dürfen in Russland importiert werden. WAAS (Wide Area Augmentation System) übermittelt dank geostationärer Satelliten GPS-Korrektursignale, die die Ortsauflösung in den USA verbessern. Die gleiche Aufgabe, aber für Europa, übernimmt EGNOS (Euro-

pean Geostationary Navigation Overlay Service). Europa arbeitet ferner an einem eigenen GNSS-System mit dem Namen Galileo. Acht Satelliten funktionieren bereits, die letzten beiden befinden sich allerdings in niedrigeren Orbits als geplant. Weitere sollen in diesem Jahr in den Orbit geschossen werden. 28 Satelliten sind für einen Vollausbau erforderlich. Das chinesische System Beidou (Großer Wagen) befindet sich ebenfalls im Aufbau. Es umfasst mehrere geostationäre Satelliten, die über China stehen, weshalb es vor allem für den asiatischen Raum interessant ist.



Bei der Multilateration wird die Zeit gemessen, die jedes Signal vom Satelliten zum GPS-Empfänger braucht.

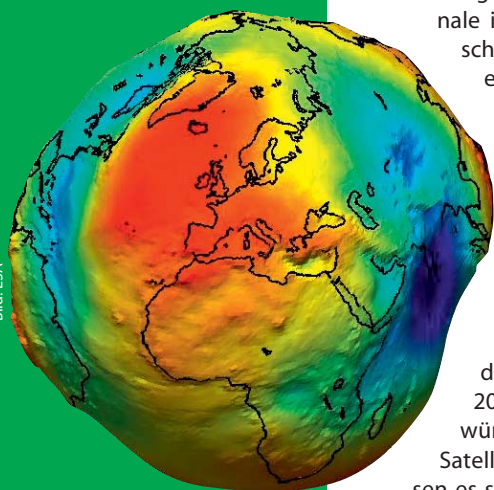
Ihre Positionsdaten funken die Satelliten zum GPS-Empfänger. Verglichen mit Satelliten-TV ist die Übertragungsrate von GPS-Satelliten mit 50 Bit pro Sekunde ausgesprochen niedrig. Es dauert 30 Sekunden, bis ein Satellit seine Bahndaten, die Ephemeriden, mitgeteilt hat, und 12 Minuten, bis die vollständigen Daten aller Satelliten vorliegen (Cold Start). Ist der komplette Katalog, der Almanach, einmal gespeichert, kann der Empfänger die Position der Satelliten vorhersagen und bereits wenige Minuten nach dem Start erste Positionswerte anzeigen (Hot Start).

GPS UND DIE RELATIVITÄTSTHEORIE

Jeder Satellit bestätigt sowohl die spezielle als auch die allgemeine Relativitätstheorie. Gemäß der speziellen Relativitätstheorie laufen die Uhren, bezogen auf den ruhenden Beobachter auf der Erde, langsamer. Da sich die Schwerkraft mit zunehmender Höhe verringert, laufen die Uhren gemäß der

allgemeinen Relativitätstheorie schneller als am Boden. In einer Höhe von circa 3000 Kilometern heben sich beide Effekte auf, in größeren Höhen überwiegt der Schwere-Effekt. Unkorrigiert würde sich der Zeitfehler über einen Tag auf circa 10 Mikrosekunden summieren.

Bild: ESA



Die Erde als Geoid mit eingefärbter und überhöht dargestellter Geoundulation. Rote Flächen ragen über das Ellipsoidmodell hinaus, blaue bleiben darunter.

GPS-NACHTEILE

In Gebäuden funktioniert GPS oft nicht, dafür sind die Satellitensignale tatsächlich zu schwach und anfällig gegenüber Störungen.

Die Signalstärke der gesendeten Positionssignale ist niedriger als das thermische Rauschen. Dass es einfache Patch-Antennen empfangen können, liegt zum einen an der Spreizung des Signals. Es belegt eine Bandbreite von 20 Megahertz. Zum Vergleich: Rundfunk-Mittelwellensender begnügen sich mit einer Bandbreite von 0,01 Megahertz. Zum anderen kennt der Empfänger die erwarteten Signale, die sogenannten PRN-Codes der Satelliten.

Zurzeit sind 32 GPS-Satelliten aktiv, die die Erde in einer Höhe von circa 20 000 Kilometern umkreisen. Genügen würden 24, um jederzeit mindestens vier Satelliten in Sichtweite zu haben. Vier müssen es sein, da nicht nur drei Raumkoordinaten bestimmt werden, sondern als vierter Parameter auch die Zeit. So spart sich der Empfänger eine eigene Atomuhr.

Hinzu kommen EGNOS-Transponder auf geostationären Satelliten (siehe Kasten GNSS-Systeme). Während die GPS-Satelliten lediglich Abschätzungen mitteilen, inwieweit ihr Signal durch die Ionosphäre verzögert wird, übermittelt das EGNOS aktuelle Korrekturwerte. Die Lateralaufklärung verbessert sich unter günstigen Bedingungen auf weniger als 5 Meter.

Im Alltag rechnen wir mit Ortskoordinaten wie Länge, Breite und Höhe, die stillschweigend die Erdoberfläche voraussetzen. Satelliten kennen als Bezugspunkt nur den Schwerpunkt der Erde, denn die Erdachse ist nicht stabil und die Kontinente driften, sodass sich Positionsangaben nicht ohne weiteres ineinander umrechnen lassen. GPS-Koordina-

ten beziehen sich auf das System WGS84 (World Geodetic System 1984).

Die Eurasischen Kontinentalplatte ist in sich recht stabil, driftet aber um circa 2,5 Zentimeter pro Jahr in nordöstlicher Richtung relativ zum Rest der Welt. Europa rechnet deshalb mit dem European Terrestrial Reference System (ETRS89). Je nach System und Epoche unterscheiden sich WGS84 und ETRS89 um circa 50 Zentimeter. Die gute Nachricht: Genügt eine Genauigkeit von wenigen Metern, unterscheiden sich die unterschiedlichen Koordinatensysteme nur wenig voneinander.

Für die Höhenbestimmung ist der Koordinatenursprung im Erdmittelpunkt unpraktisch, da er zu Angaben wie 6 371 621 Meter führt. Stattdessen interessiert die Höhenangabe bezogen auf die Oberfläche der Erde. Ein weiteres Problem: die Gestalt der Erde ist nur in nullter Näherung eine Kugel. In erster Näherung ist sie ein Ellipsoid, da die Erdrotation den Körper am Äquator um circa 20 Kilometer abplattet. Insgesamt ähnelt die Erdfigur einer Kartoffel, mit Abweichungen von bis zu 100 Metern verglichen mit einem optimalen Ellipsoiden. Die Differenz zwischen Ellipsoiden und Schwerefeld-Referenzfläche (Geoid) ist die Geoundulation. Im WGS84 werden basierend auf dem Earth Gravitational Model 1996 (EGM96) die Bezugsgrößen Ellipsoid, Geoid und Nullpunkt (Nullmeridian) festgelegt.

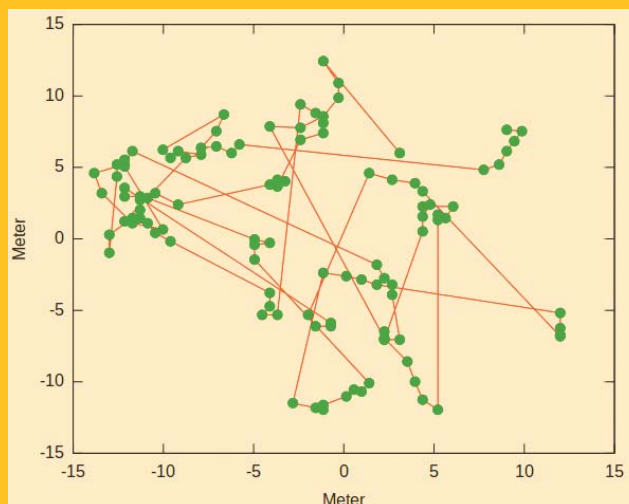
Ist der Standort auf der Erde bekannt, lässt sich die Entfernung vom Erdmittelpunkt zur Ellipsoid-Oberfläche angeben. Eine Tabelle liefert die Geoundulation und so die Bezugsfläche für die Normalhöhe. Damit wird aus der Angabe von 6 371 621 Metern für die Entfernung zum Erdmittelpunkt die Höhenangabe von 100 Metern Normalhöhenull (NHN).

WIE GENAU IST GPS?

Die Meinungen, wie genau die Ortsauflösung von GPS ist, reichen von einigen Millimetern bis zu 100 Metern. Die populäre Vorstellung von 100 Metern Genauigkeit resultiert aus der Zeit vor dem Jahr 2000. Bis dahin wurden zivile GPS-Signale verschlechtert und waren für die Straßennavigation nicht zu gebrauchen.

In der Praxis rechnet man mit einem Wert von 5 bis 15 Metern in der Fläche und 10 bis 20 Metern Fehler in der Höhe. Die Auflösung reicht aus, um Straßen zu finden. Sie ist zu klein, um auch die Fahrspur eines Autos sicher genug zu erkennen. Mit der Verrechnung von Korrekturdaten aus bodengestützten, differentiellen GPS-Systemen verschiebt sich die Auflösungsgrenze bis in den Millimeterbereich. Ein Anbieter für Korrekturdaten in Deutschland ist beispielsweise der Satellitenpositionierungsdienst der deutschen Landesvermessung (SAPOS).

Konventionelle GPS-Empfängerchips versagen ihren Dienst bei schneller Bewegung, typischerweise ab 1500 Stundenkilometern Geschwindigkeit, der vierfachen Erdbeschleunigung oder in 18 000 Metern Höhe. Damit taugen sie nicht für den Einsatz in militärischen Raketen.



Lokalisierungen während einer 30-minütigen Aufzeichnung eines ruhenden Empfängers

GPS-Daten einlesen

```

void setup() {
  Serial.begin(9600);
  Serial.println("Start");
}
void loop() {
  if (Serial.available())
    Serial.write(Serial.read());
}

```

Angewandte Satellitenpositionsbestimmung

Für die Praxis der GPS-basierten Positionsbestimmung nutzen wir nun einen Empfänger mit einem Arduino-Mikrocontroller. Zum Einlesen der Daten genügt ein Programm, das kompiliert im Arduino nur 2 kB Speicher belegt (siehe Listing).

Der Arduino kommuniziert mit einem Rechner über eine serielle Schnittstelle und einen eingebauten USB-Umsetzer. Der Pin 0 (Rx) empfängt die Daten vom GPS-Tx-Ausgang, der Pin 1 (Tx) kommuniziert wiederum mit dem Rechner. Wenn dieser keine Daten schickt, ist die Arduino-Leseleitung frei. Wir nutzen sie, um Daten aus dem GPS-Modul entgegenzunehmen. Zwei Einschränkungen sind zu beachten: 1. Schreib- und Leserate lassen sich nicht unabhängig voneinander einstellen. Da das GPS-Modul mit 9600 Baud sendet, ist dies auch die Ausgabe-geschwindigkeit des Rechner-Monitorprogramms. 2. Die GPS-Sendeleitung Tx darf erst nach dem Upload der Software an den Arduino Tx Pin 0 angeschlossen werden. Anderenfalls funkt der Chip dazwischen und verhindert die Übertragung des Programms.

So könnte ein empfangener Datensatz zu Beginn aussehen:

```

$PMTK011,MTKGPS*08
$PMTK010,001*2E
$GPGGA,074958.539,8960.0000,N,00000.0000,E,0,0,,137.0,M,1
3.0,M,,*4B
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPGSV,1,1,00*79
$GPRMC,074958.539,V,8960.0000,N,00000.0000,E,0.00,0.00,12
0714,,,N*78

```

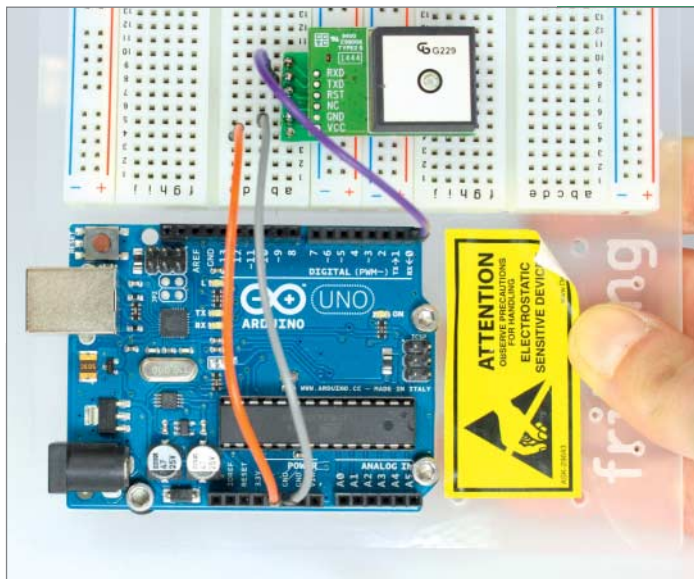
Nach circa 30 Minuten sieht er so aus:

```

$GPGGA,093004.000,5052.0000,N,00710.0000,E,2,7,1.09,83.4,
M,47.7,M,0000,0000*65
$GPGSA,A,3,24,12,15,14,25,06,04,,,,,,,,,1.83,1.09,1.47*03
$GPGSV,3,1,12,24,73,121,37,12,72,253,34,14,34,307,28,25,34,
250,27*77
$GPGSV,3,2,12,33,27,208,20,17,22,043,,15,17,178,28,04,14,06
3,15*73
$GPGSV,3,3,12,06,13,094,22,22,08,274,,02,04,125,,32,02,349,*
77
$GPRMC,093004.000,A,5052.0000,N,00710.0000,E,0.00,54.71,1
80714,,,D*59

```

Der Empfänger meldet sich mit der herstellerspezifischen PMTK-Sequenz. In der GPGGA-Sequenz sind alle Zeit- und Positionsdaten ungültig, da das Valid

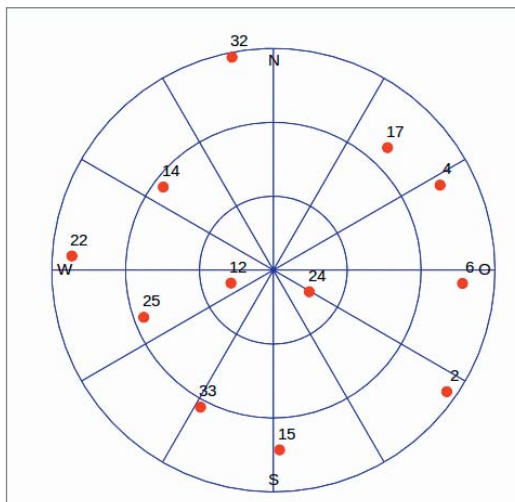


Mit 3 Kabeln und 6 Zeilen Code wird ein Arduino zu einem GPS-Lesegerät.

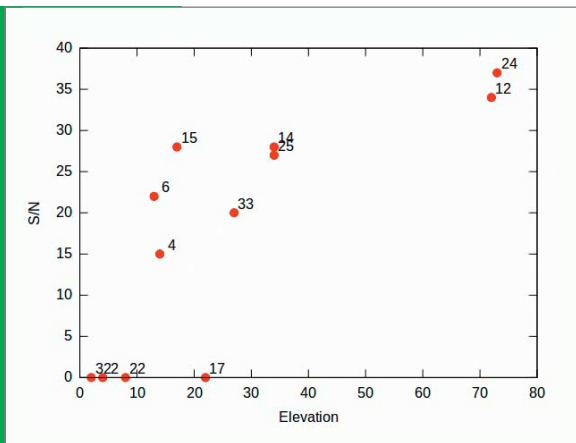
Flag (nach dem „E“) auf Null steht. Das gleiche gilt für die folgenden Blöcke. Bald darauf erkennt der Receiver mindestens einen Satelliten und stellt die Uhr. Nach 10 Minuten bestimmt er erstmals die Position. Das Valid-Flag wechselt auf 1. Nach weiteren 5 Minuten gelingt der Empfang des EGNOS-Korrektursignals, das Valid-Flag springt auf 2. Das Polardiagramm (s. Abb. unten) verzeichnet alle Satelliten mit Sichtverbindung zum Empfänger.

Das Signal-Rausch-Verhältnis S/N (siehe Bild nächste Seite) der horizontnahen Satelliten ist zu gering für eine Auswertung der Signale. Zum einen wird es aufgrund der langen Laufzeit durch die Atmosphäre geschwächt. Zum anderen bevorzugt die Richtcharakteristik der Patch-Antennen Signale senkrecht zur Antennenfläche. Aus den 12 erkannten Satelliten wählt die GPGSA-Datenzeile sieben aus, die für die Positionsbestimmung zur Verfügung stehen. Bei vier Satelliten ist das Signal zu schwach oder wird gar nicht ausgestrahlt (möglicherweise bei PRN17).

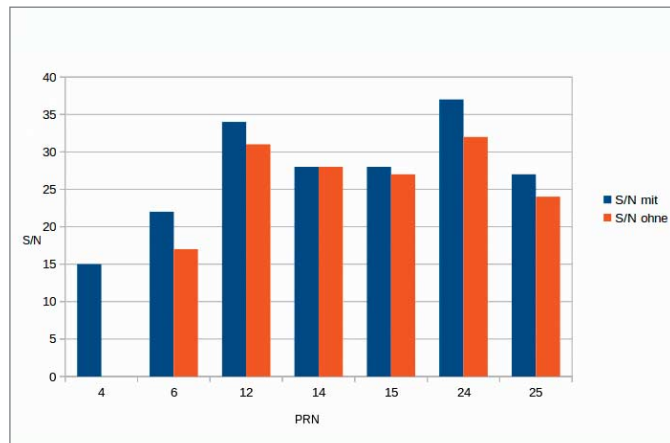
Dem Sender mit der PRN-Nummer 33 kommt eine Sonderrolle zu. Während die anderen Satelliten zügig ihre Bahnen am Himmel ziehen, steht er orts-



Satelliten in Sicht in Köln am 18. Juli 2014 um 10:30 Uhr: Die Satelliten PRN 12 und 24 stehen fast im Zenit, PRN 32, 22 und 2 nahe am Horizont jeweils in nördlicher, westlicher und südöstlicher Richtung. Die Empfangsstärke korreliert mit der Höhe über dem Horizont.



Die Signalstärke S/N in Abhängigkeit von der Satelliten-Elevation



Vergleich der Signalstärke S/N mit und ohne Reflektorfläche

fest am Himmel, von Köln aus gesehen in südsüdwestlicher Richtung. Es handelt sich dabei um den geostationären Satelliten Inmarsat 3-F2 mit einem EGNOS-Transponder. Er strahlt für Europa GPS-Korrektursignale aus, überträgt aber keine Signale für die Positionsbestimmung. Er ist der fünfte Satellit von 12 erkannten Satelliten, die nicht zur Standortberechnung herangezogen werden.

Bei schlechten Empfangsbedingungen geht das Signal von PRN 33 verloren, was auch in der GPRMC Signatur notiert wird: D wechselt auf A.

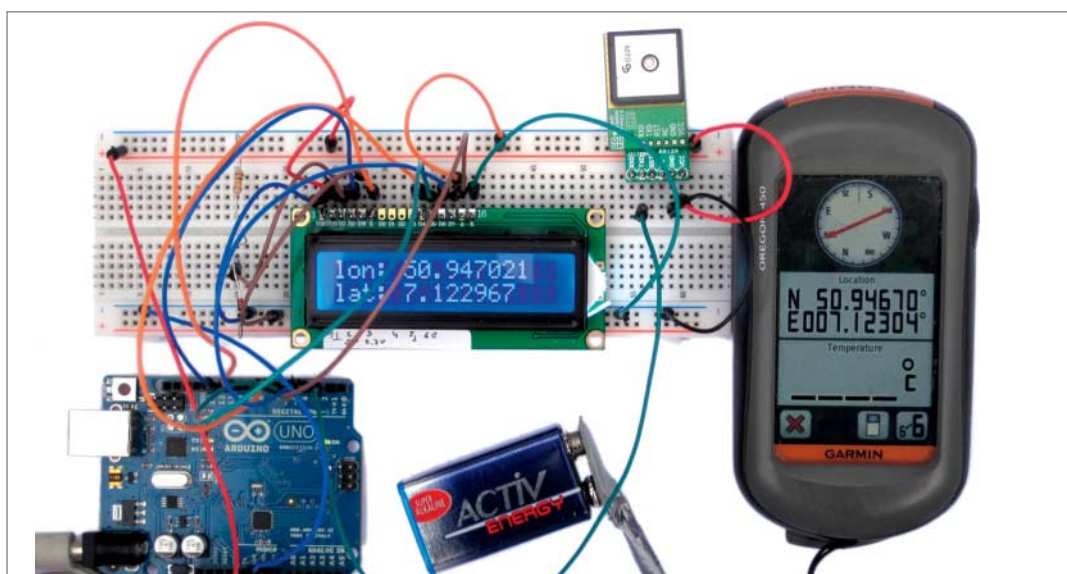
Um die Baugröße einzuhalten, verzichtet die Antenne des Skylab-GPS-Moduls auf eine Reflektorfläche. Rüstet man sie nach, steigert sie den Antennengewinn um 2 bis 3 Dezibel. Als Experiment wird ein Metallstreifen unter das Modul geschoben und mit Masse verbunden. Es genügt, wenn er auf drei Seiten über die Patch-Antenne hinausragt. So vermeidet man Kurzschlüsse der Stiftleiste an der Vorderseite. Die Grafik vergleicht einige experimentelle Werte. Unterhalb von circa 15 Dezibeln bricht der Empfang ein. Satellit PRN 4 ist ohne das Hilfsmittel nicht mehr zu empfangen.

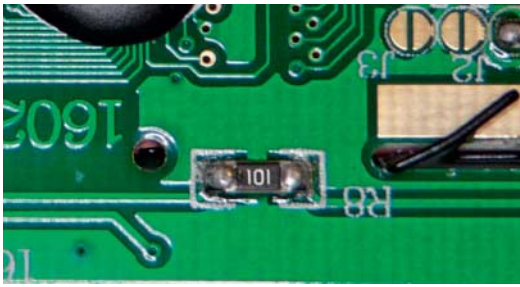
Mobile GPS-Fuchsbaujagd

Eine Anwendung für GPS-Empfänger ist das Spiel Fuchsjagd. Funkamateure sprechen von einer Fuchsjagd, wenn sie versteckte Sender durch Kreuzpeilung ausfindig machen. In unserem Spiel ist das Suchobjekt passiv, es sendet nicht. Stattdessen gilt es einen Ort zu finden, den Fuchsbau. Dafür lassen wir uns vom selbst gebauten GPS-Gerät die Entfernung zu unserem Ziel ausgeben. Dort müssen wir noch eine Aufgabe lösen.

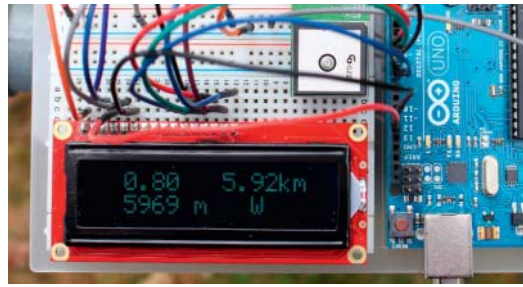
Um die Daten unabhängig vom USB-Anschluss des Arduino auszutauschen, brauchen wir eine eigene serielle Schnittstelle. Die SoftwareSerial Library (in neueren Arduino-Versionen enthalten, ansonsten siehe Download-Link) stellt sie uns zur Verfügung. Sie wird fast so bedient wie die eingebaute, mit dem Unterschied, dass wir die Ein/Ausgabe-Pins frei wählen können. Sie ist auch fast so schnell und überträgt 9600 Baud ohne Probleme. Die Auswertung der NMEA-Strings vereinfacht die Bibliothek TinyGPS+ (siehe Download-Link).

Der selbst gebaute mobile GPS-Empfänger im Vergleich mit einem kommerziellen Modell





Ob die Hintergrund-Beleuchtung einen Schutzwiderstand braucht, beantwortet ein Blick auf den Anschluss auf der Unterseite der Platine. Hier begrenzt ein 100-Ohm-SMD-Widerstand ($101 = 10 * 1e1$) den Strom.



Am Anfang der Reise: Es sind noch 5969 Meter in westlicher Richtung bis zum Ziel, gehe dazu 0,80 Kilometer nach Süden und 5,92 Kilometer nach Westen.

Zu guter Letzt benötigen wir ein Ausgabe-medium. Schließlich soll der Empfänger ohne Rechnermonitor arbeiten. Wir wählen eine 16x2 LCD-Anzeige. Auch für sie gibt es eine Bibliothek (siehe Download-Link), wenn das Display Hitachi HD44780 kompatibel ist. Da wir keine Daten vom Display empfangen möchten, liegt die Leitung R/W auf Masse. Neben vier Datenleitungen zur Übertragung von Halb-Bytes und zwei Steuerleitungen ist das Display mit zwei weiteren Leitungen für die Spannungsversorgung mit dem Arduino verbunden (siehe Tabelle). Eine Spannung von 0,6 bis 0,7 Volt am LCD-Pin V0 stellt den Kontrast ein. Verwendet werden können ein Potentiometer, Spannungsteiler oder eine Schaltodiode. Dabei liegt die Diode in Durchlassrichtung über einen Schutzwiderstand auf 5 Volt.

Ein GPS-Gerät weiß nicht, in welche Richtung es schaut. Aus der Bewegung heraus könnte die Richtung zum Ziel abgeleitet werden. Wir machen uns es einfacher und geben die Himmelsrichtung an.

Die Box auf dem Foto ist ihrem Ziel schon recht nahe. Noch 5969 Meter fehlen bis zum Ziel in westlicher Richtung. Haben sich die Schatzsucher auf

weniger als 300 Meter genähert, wechselt die Anzeige. Sie gibt die aktuelle Höhe aus und verrät den Namen des Zielobjekts. Im Beispielprogramm ist es der Booser Eifelturm in der Eifel. Bei 50 Meter Abstand stellt das Gerät die Rätselfrage: Wie hoch ist der Turm? Aus der Differenz der Messwerte am Boden und auf dem Turm folgt eine Höhe von circa 25 Metern.

Wenn sich die richtigen Schatzsucher zusammenfinden, könnten sie das Programm erweitern. Wenn zum Beispiel das erste Ziel erreicht ist, legt das Programm intern einen Schalter um und führt zur nächsten Aufgabe. Die Zeitmessung des Arduino könnte die Schnelligkeit bewerten, mit der eine vorgegebene Strecke zurückgelegt wird. Abfragen, die eine Beobachtung vor Ort erfordern, könnten Alternativrouten vorschlagen. Im Speicher des Arduino Uno ist dafür noch ausreichend Platz. —hch

↓ Links und Foren
make-magazin.de/xd2r

Anschlüsse

Display

1	VSS	-
2	VDD	+
3	V0	Kontrast
4	RS	Digital 12
5	R/W	-
6	E	Digital 11
11	D4	Digital 7
12	D5	Digital 6
13	D6	Digital 5
14	D7	Digital 4
15	A	+
16	K	-

GPS-Modul

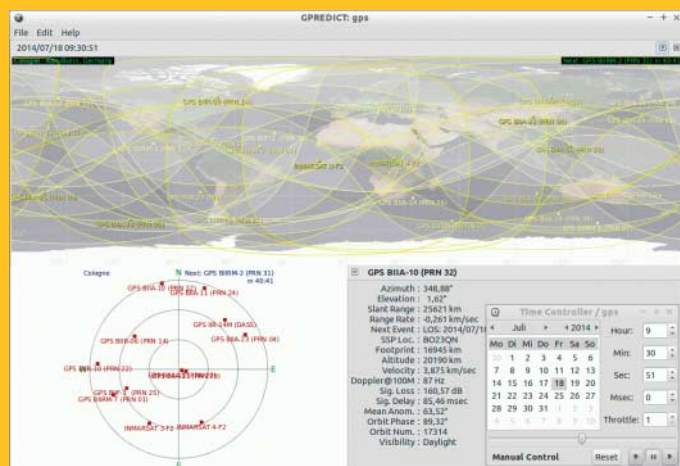
VCC	+
GND	-
TXD	Digital 2

Bei einigen Displays ist die Pinbelegung nicht auf der Platine vermerkt. Dann hilft ein Blick ins Datenblatt.

GPREDICT

Das freie Programm Gpredict zeigt alle Satelliten an, deren Bahndaten frei zugänglich sind, im Augenblick mehr als 1300 Satelliten. Dabei unterscheidet es nicht zwischen Satelliten, die außer Dienst gestellt wurden und solchen, die noch funken. Nach Angabe des Beobachtungsstandorts erscheinen die Satelliten so, wie sie gerade am Himmel stehen. Oder man verstellt die Uhr und simuliert die Konstellation zu einem beliebigen Zeitpunkt. Auf diese Weise entstand der Bildschirmausdruck. Er lässt sich unmittelbar mit den Daten vergleichen, die das GPS-Modul im Bild ganz oben empfängt.

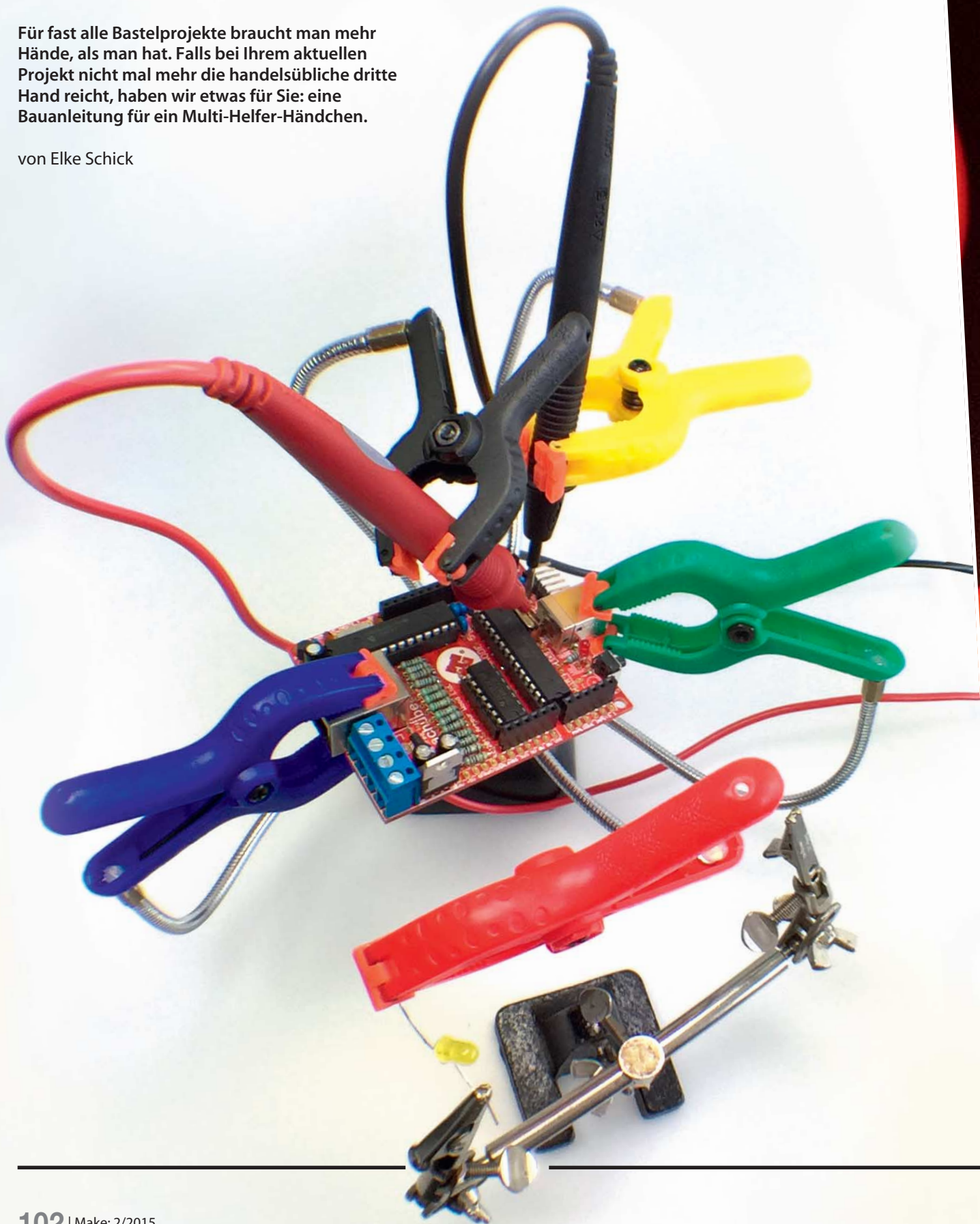
GPS-Satelliten auf einen Blick



Händchen hilf!

Für fast alle Bastelprojekte braucht man mehr Hände, als man hat. Falls bei Ihrem aktuellen Projekt nicht mal mehr die handelsübliche dritte Hand reicht, haben wir etwas für Sie: eine Bauanleitung für ein Multi-Helfer-Händchen.

von Elke Schick



Wer lötet, kennt und braucht sie: die dritte Hand. Ein kleines, äußerst sinnvolles Werkzeug mit zwei biege- und verschiebbaren Armen, an deren Enden zwei Klemmen alles halten können, was beim Löten nicht mehr in die eigenen Hände passt. Häufig ist dieses Werkzeug noch mit einer Lupe versehen, die man ebenso verschieben kann – auch das eine sehr große Hilfe bei der Arbeit mit kleinen Bauteilen. Die dritte Hand kann nicht nur beim Löten zweckmäßig sein – auch bei allen Projekten, bei denen zum Beispiel Kleber trocknen muss, ist sie ein geeignetes Werkzeug.

Aber die handelsübliche dritte Hand ist meist nur 8 cm hoch und hat eine Spannweite von 11 cm. So kommt sie schnell an ihre Grenzen, wenn größere Projekte fixiert werden sollen. Und zwei Klemmen können auch

gerne einmal zu wenig sein. Wir haben daher eine Hand gebaut, die mehr als nur die dritte Hand ist. Sie hat fünf biegbare Arme und ist mit einer Höhe von 23 cm ein Stück größer als die durchschnittliche dritte Hand. Die Grundlage für unsere Bastelei bildet eine günstige LED-Effekt-Lampe, die für unter zehn Euro in einem Discounter erhältlich war. Die LEDs und Lampenschirme an den Enden haben wir durch leichte Plastikklemmen ersetzt. Damit die Hand stabil ist, haben wir den Fuß mit Füllmaterial aus einem alten Lampenfuß beschwert. Wer das – im Gegensatz zu uns – nicht einfach vorrätig hat, kann auf Bleikugeln zurückgreifen. Bezugsquellen für alle Materialien finden Sie in den Links. Mit ein wenig Schrauben, Bohren, Kleben und Schneiden war unsere helfende Hand schnell fertig.

Kurzinfo



Zeitaufwand:
30 bis 60 Minuten



Kosten:
circa 15 Euro



Feinwerkzeug:
Bohren, Schrauben, Trennen, Heißkleben

Schwierigkeitsgrad



Einkaufsliste

- » Lampe mit vielen Metallschlauch-Armen
- » genügend Klemmen für die Arme
- » Füllmaterial für den Lampenfuß
- » Schrauben
- » Distanzbolzen mit Innengewinde und Zapfen



Unser Ausgangsmaterial war durchweg sehr günstig oder gratis.



1 Lampe entkern

Wir fangen unten an, weil das im Falle dieser Lampe einfacher ist. Zuerst entfernen wir die Batterien der Lampe und schrauben den Fuß auf. Den darin befindlichen Mikrocontroller können wir einfach ausbauen und die Kabel zu den LEDs durchtrennen.



2 LEDs ausbauen

Die Gehäuse um die LEDs bestehen aus zwei zusammengeklebten Teilen. Mit dem Hammer lässt sich der obere Teil leicht mit einem Schlag ablösen. Danach braucht man für die Entfernung der LEDs nur ein Schraubchen zu lösen.



3 Restgehäuse entfernen

Den unteren Teil des Gehäuses müssen wir aufschneiden. Das Plastik ist sehr glatt und rutschig – es lässt sich am einfachsten mit einem Multifunktionswerkzeug mit Trennscheibe bearbeiten. Schneiden Sie das Gehäuse an zwei Stellen von oben bis unten auf. Danach lassen sich die Gehäuse mit etwas Kraft entfernen.



4 Gewinde anbringen

Um die Klemmen auf die Arme der Lampe schrauben zu können, drehen wir in die Metallschläuche M3-Distanzbolzen mit Innengewinde und Zapfen. Das erfordert von allen Arbeitsschritten die meiste Geduld und Kraft. Manche Schläuche müssen wir mit einem Bohrer vorsichtig etwas weiten, um die Zapfen eindrehen zu können.



5 Klemmen montieren

Unsere Klemmen haben an den Griffen bereits Löcher – diese müssen wir noch ein wenig aufbohren. Dann schrauben wir die Klemmen mit M3-Zylinderskopfschrauben in den Bolzen fest. Beim Gebrauch der Multi-Hand stellte sich heraus, dass die Bohrlöcher an den Klemmen sich relativ schnell weiten. Wir behelfen uns da mit einem Tropfen Heißkleber.



6 Hand stabilisieren

Unser Händchen ist jetzt zwar fertig, hat aber einen zu leichten Fuß und kippt daher schnell um. Um es zu beschweren, plündern wir einen Lampenfuß aus unserem Büro. Er ist gefüllt mit runden Metallplättchen. Diese füllen wir in den Fuß unseres Händchens ein. Durch die Aussparungen für die Batterien können wir den Fuß nicht bis zum Rand befüllen, das Gewicht reicht so aber auch aus. Zum Schluss schrauben wir die Bodenplatte wieder auf. Die Öffnungen in der Platte schließen wir mit Heißkleber. —esk

↓ Links und Foren
make-magazin.de/x8gt

WAS UNS INSPIRIERT



KLEIDUNG, DIE LEUCHTET

Mit einem Arduino und dem passenden AYAB-Shield kann man alte Strickmaschinen aus den 1970er Jahren mit modernen Computern ansteuern. Dass man damit mehr als nerdige Kissenbezüge machen kann, zeigen die Arbeiten von Lisa Lang: Ihre Schals aus Merinowolle leuchten dank eingewebter EL-Kabel, beim Design hat sie sich vom Filmklassiker „Tron“ inspirieren lassen. Hergestellt werden sie im Berliner Textil-Hackerspace ETIB.

Der Mantel (rechts) ist eine Zusammenarbeit von Langs Agentur Electrocouture mit dem polnischen Label Mia Manu. Er ist nicht selbstgestrickt, sondern besteht aus

einem dünnen, waschbaren Neoprenstoff. Über Reißverschlüsse ist er horizontal teilbar, wird also je nach Wetterlage schnell zur Jacke.

Die Akkus sind sowohl im Schal als auch im Mantel eingearbeitet und können einfach per USB-Buchse aufgeladen werden, wenn dem Kleidungsstück der Strom ausgeht. In Zukunft soll es noch einfacher werden: Lisa Lang und ihr Team arbeiten an einer Methode, die Leuchtkleidung drahtlos aufzuladen. —*phs*

► www.electrocouture.com



Bilder: Cristopher Santos

Außer Kleidung stellt Electrocouture auch Schmuck her. Der Anhänger aus Acryl und Leuchtdioden ist vom Art déco inspiriert.

VOGELFLAUSCHROBOTER

Die Einwohner Zimpopodus heißen zwar Robobrds, können aber gar nicht fliegen. Dafür sind sie viel zu flauschig und quadratisch. Entwickelt wurden sie von der Kanadierin Erin Kennedy, die ihre Kreationen bereits auf vielen nordamerikanischen Maker Faires vorgestellt hat.

Kennedys Traum ist, mehr Menschen für Wissenschaft und Technik zu begeistern. Die Robovögel sind daher Open-Source-Hardware und können selbst gebaut werden – die Anleitungen vermitteln mechanische, elektrische und Programmiergrundlagen. Besonders Fokus legt die Programmiererin auf die möglichen Interaktionen mit der Umwelt. So erkennen die Robobrds dank Nahfeldkommunikation, wenn sie einen bestimmten Hut tragen und

reagieren darauf: mit dem Zylinder wird gewunken, das lila Baseballcap sorgt für Freudentänze.

Dank einer erfolgreichen Crowdfunding-Kampagne vertreibt Erin seit einiger Zeit Kits zum Zusammenbau und handgefertigte Roboter. Mit dem klassischeren Buddy 4000 und Botbait, einem Tentakel, sind inzwischen weitere Modelle entstanden. Ihre Arbeit an Neuentwicklungen dokumentiert Kennedy auf ihrem Blog. Für Robotikfans veranstaltet sie außerdem regelmäßige Google-Hangouts namens RobotParty. Dort können eigene Roboterbauten vorgestellt werden.

—hch

► www.robobrrd.com



Bild: Erin Kennedy



Bild: Daniel Tännler

STEAMPUNK-R2-D2

Der Schweizer Steampunk-Künstler und Maker Daniel Tännler hat nicht nur viele Namen (in der Szene kennt man ihn als The Chocolatist und als Dan Aetherman), sondern auch eine Idee nach der anderen. Sein aktuelles Projekt ist eine viktorianische Version des Star-Wars-Roboters R2-D2. Der Astromechdroide besteht aus Holz, Kupfer und Messing, wie es sich für eine Steampunk-Kreation gehört (siehe auch *c't Hacks* 1/2014).

Die 6,5 Kilo schwere und 58 Zentimeter hohe R2-Einheit ist allerdings kein stilles Dekorationsobjekt für die Steampunk-Wohnung: Im Inne-

ren steckt moderne Robotertechnik, sodass Dan Aetherman seinen Droiden mit einem Android-Smartphone per WLAN steuern kann. Natürlich ist auch die zugehörige App im Steampunk-Look gestaltet.

Ein Soundmodul sorgt dafür, dass R2-D2 sich der Umwelt mitteilen kann. Anstelle der Pfeiftöne des berühmten Vorbilds erklingen dabei allerdings passende Maschinengeräusche und natürlich ein Zischen wie von einer Dampfmaschine.

—phs

► www.thechocolatist.com/steampunk-r2-d2-astromech

Röhrentechnik für Einsteiger

Für die Generation Smartphone/Spielekonsole ist die Welt der mit Röhren bestückten Unterhaltungselektronik unbekanntes Terrain. Dieser Artikel soll – nicht nur ihr – leicht verständlich aufzeigen, zu was diese Art der Elektronik zu leisten imstande war und wie man auch heute noch damit Spaß haben kann – vorausgesetzt, man fasst den Lötkolben an der kalten Seite an und weiß, wie man mit verhältnismäßig hohen Spannungen umgehen muss.

von Hans Borngräber



Die Geschichte der Elektronenröhre beginnt mit der Glühlampe, erfunden von Thomas Edison. Dieser hatte den Verdacht, dass Elektronen sich auch im Vakuum fortbewegen können. Um dies zu beweisen, erstellte er 1870 einen Versuchsaufbau mit einer Glühlampe, einer Spannungsquelle und einem empfindlichen Messgerät.

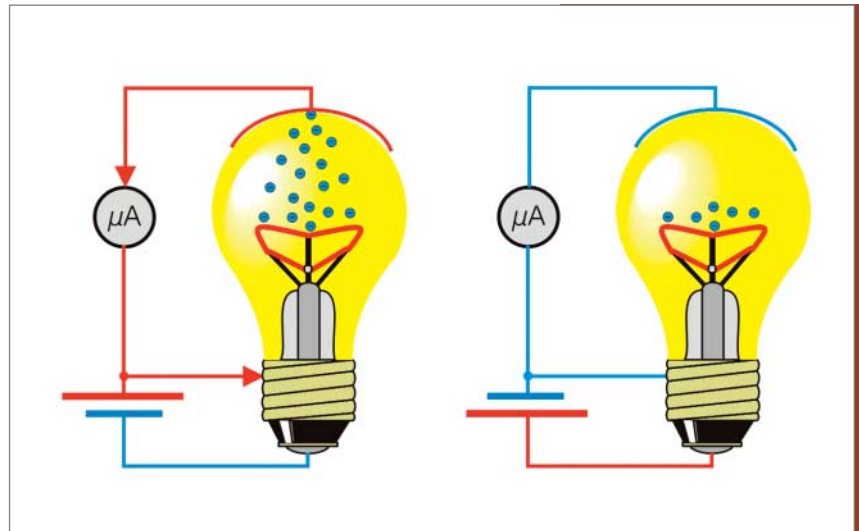
Der durch das Messinstrument angezeigte Strom bewies, dass von einem glühenden Metallkörper (der Glühwendel) Elektronen ausgesandt werden. Der Stromfluss funktionierte allerdings nur vom negativen Pol (Kathode) zum positiven Pol (Anode) hin – polt man die Spannung um, fließt kein Strom. Diese Messanordnung war im Grunde die erste Elektronenröhre, eine Diode. Edison ließ sich zwar diese Erkenntnis patentieren, aber er erkannte ihre Möglichkeiten nicht. In einigen Sprachen hat die Elektronenröhre bereits in dieser Zeit ihren Namen bekommen, so heißt sie auf französisch heute noch „la lampe“, auf Deutsch: die Lampe.

Der amerikanische Wissenschaftler Richardson erarbeitete in den folgenden Jahren die mathematischen Grundlagen zur Elektronen-Emission von glühenden Metallkörpern. Er erhielt dafür 1928 den Physik-Nobelpreis. Aber auf die Idee, diesen Strom steuerbar zu machen, kam auch er nicht.

Die erste richtige Elektronenröhre erfand dann 1906 der österreichische Telefonanlagenhersteller Thomas von Lieben. Er war auf der Suche nach einem Verstärker für seine Anlagen auf das ausgelaufene Patent von Edison gestoßen. Und er überlegte sich, wie er eine verhältnismäßig geringe Spannung wieder verstärken konnte, um die Signalübertragung über längere Telefonstrecken zu verbessern.

Die Elektronenröhre ist ein Kind der Telefontechnik. Das erklärt auch den Begriff „Kathodenstrahlrelais“, den von Lieben in seiner Patentschrift von 1906 verwendete. 1910 folgte das zweite Patent, das er zusammen mit Eugen Reiß und Sigmund Strauß beim kaiserlichen Patentamt in Berlin einreichte. Ende 1910 reichte von Lieben das dritte Patent ein, das man als Geburtsstunde der modernen Elektronenröhre bezeichnen kann. Zwar hatte der Amerikaner Lee de Forest 1906 eine ähnliche Konstruktion zum Patent angemeldet, aber seine „Audion-Röhre“ funktionierte nicht. Erst von Lieben gelang es, mit einer verhältnismäßig schwachen Steuerspannung eine große Spannungsänderung an der Anode der Röhre zu erzielen. Dazu baute er eine Drahtgitterkonstruktion in die Röhre ein, die er mit einer Spannung ansteuerte. Diese angelegte Steuerspannung wiederum bestimmte die über einem Lastwiderstand abfallende Ausgangsspannung. Von Lieben hatte damit die erste Röhren-Triode (Drei-Pol-Röhre) erfunden.

Die beiden Weltkriege führten dazu, dass die Elektronenröhre eine rasante Entwicklung nahm. Funk, Radar und Weitverkehrs-Telefonie waren die Nutznießer dieser Entwicklungen. Zunächst gefördert vom deutschen Kaiserhaus, durch Gründung der Firmen Telefunken und AEG, wurde die Röhrentechnik zu einem Selbstläufer und zur Keimzelle heute selbstverständlicher Technik – die aber inzwischen meist ohne sie auskommt.



Edisons Messaufbau: Beschaltet man die Glühwendel einer Lampe als Kathode (negativ), wandern Elektronen durchs Vakuum zur Anode hin (links). Bei Umpolung fließt allerdings kein Strom (rechts).

Elektronenröhren findet man heute zum Beispiel noch in Gitarrenverstärkern: Ein Gitarrist, der etwas auf sich hält, würde nie mit „Sand“ (Siliziumtransistoren) spielen. Rundfunksendeanlagen mit großer Leistung für Lang-, Kurz- und Mittelwelle werden mit Elektronenröhren betrieben. Auch in Hi-Fi-Verstärker werden gerne Röhren eingebaut – allerdings nicht in Geräten für die breite Masse: Ihr Preis ist zu hoch.

Es gibt also immer noch einen Markt für Elektronenröhren, und deshalb gibt es auch noch Hersteller von Röhren. Neue Röhren kommen heute aus Shunguang in China oder dem Reflektor-Werk in Saratov (Russland). Handelsnamen dieser Röhren sind etwa Sovtek, Electro Harmonix, Tung-Sol, Mullard, Golden Lion und viele andere – teilweise auch alte Markennamen, die aber mit den ursprünglichen Firmen nichts mehr zu tun haben. Die heutigen Hersteller arbeiten noch auf Fertigungsanlagen, die in den 60er und 70er Jahren des vergangenen Jahrhunderts in den damaligen Ostblock verlagert wurden, um Produktionskosten zu sparen. Es gibt von den genannten Herstellern aber auch wieder neu entwickelte Röhren, etwa die ECC99.

Funktion einer modernen Röhre

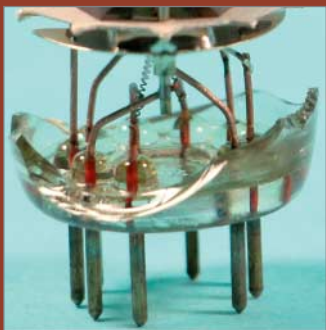
Die Von-Lieben-Röhre war eine Konstruktion von der Größe einer Ein-Liter-Bierflasche. Für viele Anwendungen war das zu groß. Die Röhre zu verkleinern und für die Massenproduktion tauglich zu machen war vor allem Verdienst von AEG und Telefunken.

Eine moderne Röhre (Triode) besteht aus folgenden Komponenten:

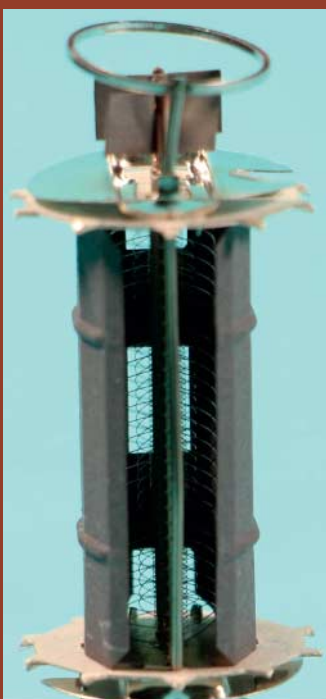
- Glaskolben, luftleer gepumpt
- Glas-Sockel mit Anschlussstiften
- Heizwendel
- Kathodenblech (bei indirekt geheizten Röhren)
- Steuergitter

VON MINUS NACH PLUS?

Es gibt in der Elektrotechnik zwei Stromrichtungen: die **physikalische** von Minus nach Plus, um (wie hier) physikalische Vorgänge zu beschreiben, und die **technische Stromrichtung** von Plus nach Minus. Diese wurde eingeführt, um damit einfacher rechnen zu können.



Röhrensockel einer EL83 mit Glasboden und Heizungsanschlüssen



Die drei Gitter (Drahtspiralen), die Anodenkühlbleche, der ringförmige Getter-Träger

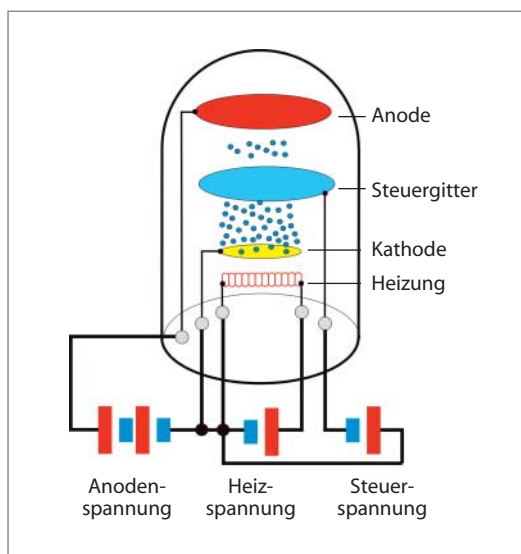
- Anodenblech
- Getter (metallischer Niederschlag, um Sauerstoffatome zu binden).

Diese sieben Bestandteile bilden heute eine moderne Elektronenröhre. Der evakuierte Glaskolben liefert das Gehäuse für das Innenleben der Röhre und den Träger für den Stecksockel. Die Heizung ist die Basis und das zuunterst eingebaute Element. Sie besteht aus einer als Doppelhelix ausgeführten Glühwendel. Aber warum doppelt?

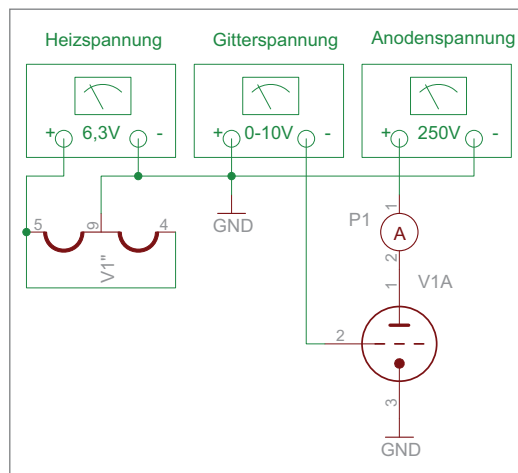
Am Anfang ihrer Entwicklung wurden Elektronenröhren mit Gleichspannung geheizt, was teure Akkus erforderte. Bald kam man auf die Idee, stattdessen die Netz-Wechselspannung zu nutzen. Doch damit brockte man sich ein neues Problem ein: den Heizungsbrumm. Diese 50 Hertz sind in so manchem alten Radio gut zu hören. AEG-Telefunken entwickelte daraufhin eine Heizwendel mit Doppelhelix – diese neutralisiert die für den Brumm verantwortlichen Magnetfelder.

Eine Anmerkung: Ich beschreibe hier absichtlich nur indirekt geheizte Röhren. Direkt geheizte Röhren sind heutzutage kaum noch zu finden, es gibt höchstens noch Exemplare im Kilowattbereich bei Rundfunksendern. Oh weh, jetzt hätte ich fast die Hi-Fi-Fans vergessen, die auf direkt geheizte Trioden wie die Western Electric 300B schwören. Diese Röhre wurde für Weitverkehr-Telefonverstärker entwickelt und soll für manche Audiophile magische Eigenschaften haben. Originalröhren dieses Typs erzielen heute vier- bis fünfstelligen Preise bei Auktionen. Nachbauten dieser Röhre bekommt man preiswerter, aber ein dreistelliger Betrag ist da immer noch fällig.

Nächstes Element ist das Kathodenblech – dies ist die eigentliche Elektronenquelle. Wird das Blech beheizt, fängt es an, Elektronen auszusenden, die sich in Richtung des nächstgelegenen positiven Poles bewegen. Elektronen sind faul, sie nehmen



Die schematische Darstellung einer Triode zeigt, warum man für ihren Betrieb drei verschiedene Spannungen braucht.



Messanordnung zur Kontrolle des Anodenstroms einer ECC83

Bauarten von Röhren

Diode	kein Gitter	z. B. Gleichrichter EZ81
Triode	1 Gitter	z. B. Vorverstärkerröhre EC83
Pentode	3 Gitter	z. B. Endverstärkerröhre EL84
Heptode	5 Gitter	z. B. Mischverstärkerröhre EK90
Oktoide	6 Gitter	z. B. Batterieröhre für mobile Empfänger DK21

den Weg des geringsten Widerstandes, und zwar unabhängig von der Lage. Werden sie dabei gestört, verringert sich der Stromfluss durch die Röhre. Und schon hat man einen Weg gefunden, den Stromfluss zu steuern: Man muss nur eine Störquelle einbauen, die den Elektronenfluss behindert. Dies geschieht in Form eines geflochtenen Gitters oder über Drahtspiralen. An diese wird eine negative Spannung gegenüber der des Kathodenblechs angelegt. Da sich gleiche Ladungen abstoßen, gelangen nicht mehr so viele Elektronen zur Anode und der Stromfluss verringert sich. Kleine Spannungsänderungen am Steuergitter führen so zu einer großen Spannungsänderung an der Anode. Somit hat man den gewünschten steuerbaren Verstärkungseffekt erreicht.

Diese Art, mit einem Gitter den Elektronenfluss zu beeinflussen, wird bei einer Triode angewandt. Aber es gibt noch andere Bauarten von Röhren, die mit mehr Gittern ausgestattet sind oder ganz ohne auskommen. Die Namensgebung orientiert sich an der Anzahl der Systeme. Es gibt immer Anode und Kathode plus eine Anzahl von Gittern (siehe Tabelle).

Ein großes Problem bei der Herstellung der Röhren stellte der in den Bauteilen vorhandene Sauerstoff dar. Dieser verschlechtert während des Betriebes das Vakuum im Inneren der Röhre. Seit der Erfindung der Quecksilber-Vakuumpumpen war es zwar möglich, die Glaskolben effektiv zu evakuieren (luftleer zu pumpen), es blieb aber immer noch ein Rest Sauerstoffatome im Glaskolben zurück.

Denen rückte man im Produktionsprozess mit einer sogenannten Getter-Pille zu Leibe. Auf einem Metallring in der Spitze der Röhre wurde eine Tablette aus einer Barium-, Magnesium- oder Aluminium-Verbindung platziert und nach Verschluss des Glaskolbens induktiv erhitzt. Dabei verdampfte die Metalllegierung und band die restlichen Sauerstoffatome im Glaskolben. Das überschüssige Getter-Material schlug sich als metallisch glänzender Belag in der Spitze des Glaskolbens nieder. Dieser Belag diente dazu, die während des Betriebes der Röhre anfallenden Sauerstoffatome zu binden. Findet man eine Röhre, bei der sich der metallische Belag weiß verfärbt hat, so ist das Getter komplett aufgebraucht oder die Röhre hat Luft gezogen – in beiden Fällen ist sie unbrauchbar.

Probe aufs Exempel

Wo wir gerade bei defekten Exemplaren sind: Röhren sind einem Verschleiß unterworfen und müssen irgendwann ausgetauscht werden. Bei korrekter Beschaltung sind jedoch mehrere Tausend Betriebsstunden normal. Um nicht zu warten, bis ein elektronisches Gerät wegen einer defekten Röhre seinen Dienst versagte, gab es eigene Röhrentestgeräte. Die bekanntesten Tester stammten von der Firma Funke aus Adenau in der Eifel, etwa der Funke W19, der Standard-Röhrentester der Bundeswehr.

90 Prozent der Tester waren sogenannte Emissionstestgeräte, das heißt, es wurde die Röhre beheizt und der Strom gemessen, der zwischen Kathode und Anode floss. Das war genau der Versuch, den Edison schon 1870 durchführte, aber nur eine sehr rudimentäre Testmethode, die wenig über die Steuerbarkeit und die Kenndaten einer Röhre aussagte. Doch solche Testgeräte waren kostengünstig herzustellen und daher weit verbreitet. Böse Zungen nennen diese Geräte noch heute „Schätzseisen“.

Will man eine Röhre wirklich auf Funktionstüchtigkeit testen, muss man etwas mehr Aufwand treiben. Tester, die das leisteten, waren teuer und man fand sie nur in Laboren von Elektronikentwicklern und in der Ausrüstung des Militärs. Prominente Vertreter dieser Gattung waren die Geräte von Neuberger (zum Beispiel RPM70), das russische L3-3 oder das amerikanische TV-7. Sowohl das russische wie auch das amerikanische Gerät wurden zusammen mit dem entsprechenden Waffensystem ausgeliefert, auf Panzern und Schiffen mitgeführt und waren dementsprechend robust aufgebaut. Sollte man heute noch so ein Gerät irgendwo finden, braucht man sich über dessen Funktionsfähigkeit keine Gedanken zu machen: Das Gerät funktioniert. Es sei denn, es ist jemand mit dem Panzer drübergefahren oder es ist zusammen mit dem Schiff untergegangen.

Will man selbst eine Röhre messen, um herauszubekommen, was sie noch leistet, kann man nach dem Schema in der Grafik auf der vorigen Seite unten vorgehen. Dabei sind drei unabhängige Spannungsquellen nötig:

- feste Heizspannung (6,3 V)
- feste Anodenspannung (250 V)
- regelbare negative Gitterspannung (0–10 V)

Mit diesen drei Spannungen kann das Steuerverhalten der Röhre nun genau wie in einer Anwendungsschaltung getestet werden. Man verändert die negative Gitterspannung und misst den dann durch die Röhre fließenden Strom; hierzu dient Messgerät P1. Ob die gemessenen Werte einer guten oder schlechten Röhre entsprechen, kann



Ein Selbstbau-Vorröhren-Tester für Gitarrenverstärker

BAUTEIL-QUELLEN

Wer glaubt, Bauteile für Röhrenschaltungen wären nur noch auf Flohmärkten zu finden, der irrt sich. Das Internet bietet eine große Anzahl von Bezugsquellen für Röhrenprojekte. Hier eine nicht vollständige Liste von mir bekannten Lieferanten, die auch an Privatpersonen liefern:

www.tube-town.de: alles, was man zum Bauen von neuen Röhrengeräten benötigt

www.askjanfirst.de: historische Röhrenbauteile

www.btb-elektronik.de: Röhrenspezialist

www.oppermann-electronic.de: Restpostenhändler

www.welter-electronic.de: Netztrafos und Ausgangsübertrager

www.rainer-foertig.de: gebrauchte Messtechnik

www.pollin.de: Röhren und Fassungen

www.wilmsmetall.de: Bleche für den Chassis-Bau

www.indutecshop.com: Spannungsfeste Schaltlitzen

DIE WERKZEUGKISTE

Wer ohnehin elektronische Schaltungen aufbaut, hat bereits 90 Prozent der Werkzeuge bereitliegen, die für den Aufbau von Röhrenschaltungen notwendig sind. Im Einzelnen braucht man:

- Lötkolben und bleihaltiges Lötzinn
- Seitenschneider
- Biegezange
- Akkuschauber
- Bohrer 1 mm – 10 mm, 0,5 mm steigend
- Stufenbohrer 6 mm – 32 mm
- Schraubenschlüssel 5 mm – 17 mm

Ein weiteres unverzichtbares Hilfsmittel ist ein Vielmessinstrument. Aber machen Sie einen Bogen um die billigen Digitalgeräte, denn die überleben den Messbetrieb in einer Röhrenschaltung keine fünf Minuten lang. Entweder be-

schafft man sich ein funktionsfähiges Gebrauchtgerät der Firmen HP, Metrawatt, Hartmann & Braun oder Philips, um nur einige zu nennen. Oder man greift tiefer in den Geldbeutel und beschafft sich ein aktuelles neues Profigerät.

Digitale Geräte zeigen bei den hohen Spannungen schon mal Mist an. Das hängt mit ihren hochohmigen Eingängen und der fast lastfreien Messung zusammen. Da werden schon mal Störspannungen von der Nachbarleitung ungewollt mitgemessen und verfälschen das Messergebnis. Deshalb schwöre ich auf alte Analogmessgeräte vom Typ Unigor oder Philips. Sie haben einen geringen Innenwiderstand und messen nur Spannungen, die auch wirklich vorhanden sind.

SOCKEL ODER FASSUNG?

Der **Socket** ist der Glasboden der Röhre mit seinen Kontaktstiften. Die **Fassung** ist das Bauteil, in das die Röhre hineingesteckt wird. Beide Begriffe werden gerne verwechselt, was in Internetforen regelmäßig zu wilden Diskussionen führt. Ursache für diese Begriffsvertauschung ist das englische Wort **Socket**, das sowohl für Fassung wie auch für Socket verwendet wird.



Links eine Chassis-Oktalfassung, in der Mitte eine Oktalfassung für die Platinenmontage, rechts eine EL34 mit dem passenden Oktalsocket

man mit dem Datenblatt überprüfen, der zur Röhre gehört.

Mit System

Schon früh stellte sich für die Industrie als Problem heraus, dass genormte Röhrenbezeichnungen und Fassungen fehlten. Deshalb einigte man sich auf ein einheitliches Bezeichnungssystem für Röhren und normierte die Fassungen. Die Tabelle unten listet die am häufigsten vorkommenden europäischen Bezeichnungen auf.

Damit kann man nun leicht eine europäische Röhre identifizieren, etwa eine ECC83:

E = Heizung 6,3 V parallel

C = Triode 1

C = Triode 2

83 = laufende Nummer



Zwei Novalfassungen unterschiedlicher Bauart ECC803 mit Novalsocket, daneben die passende Röhre. Oktal- und Novalfassungen sind heute gebräuchlich.



Eine Außenkontaktfassung, wie sie bis zum Ende des zweiten Weltkriegs üblich war, daneben eine EL8 mit dem passenden Außenkontaktsocket

Es handelt sich also um eine Röhre mit 6,3 V Heizspannung, die zwei Triodensysteme enthält. Die laufende Nummer wurde einmalig vergeben und von allen Herstellern dieser Röhre einheitlich verwendet. Steht die 83 zwischen dem E und vor dem CC, lautet die Bezeichnung also E83CC, so handelt es sich um eine besonders selektierte Röhre mit erhöhter Lebensdauer, meist in militärischen Geräten zu finden.

Ein weiterer wichtiger Punkt, der bei Entwicklung und Verbreitung der Röhrentechnik normiert werden musste, waren die Röhrensocket und deren Fassungen. Am Anfang setzte man gerne die noch heute gebräuchlichen 4-mm-Laborstecker ein. Doch die waren einfach zu groß und man entwickelte Außenkontaktfassungen. Diese wurden bis Ende des Zweiten Weltkrieges für Endverbraucherröhren verwendet. Die heute gebräuchlichsten Fassungen sind die Oktal- und Novalfassungen.

Es gibt noch mehr Bauformen für Röhrenfassungen. Sie sind jedoch so zahlreich, dass ich da auf die im Internet verfügbaren Informationen verweisen muss.

Datenblätter

Historisch bedingt gibt es die Daten der Röhren original nur in Papierform. Im Internetzeitalter hat sie Frank Philipse online verfügbar gemacht. Er hat Tausende von europäischen, amerikanischen und russischen Röhrendatenblättern gescannt und in einer Online-Datenbank bereitgestellt. Wer Daten einer Röhre benötigt, wird unter www.tubedata.org fündig.

Europäische Röhrenbezeichnungen

1. Buchstabe			2. Buchstabe		
Heizung	Spannung/ Strom	Heizungsart	Röhrensystem	System	Verwendung
A	4 V	Parallel	A	Diode	HF-Anwendung
B	180 mA	Seriell	B	Doppel-Diode	HF-Anwendung
C	200 mA	Seriell	C	Triode	Vorverstärker
D	1,2–1,4 V	Parallel	D	Triode	Leistungsverstärker
E	6,3 V	Parallel	E	Tetrode	Leistungsverstärker
F	12,6 V	Parallel	F	Pentode	Vorverstärker
G	5 V	Parallel	H	Hexode, Heptode	Mischanwendung
H	150 mA	Seriell	K	Oktode, Heptode	Mischanwendung
K	2 V	Parallel	L	Pentode, Tetrode	Leistungsverstärker
M	2,4 V–2,8 V	Parallel	M	Anzeigeröhre	Magische Augen
P	300 mA	Seriell	Q	Ennode	Modulationsverstärker
U	100 mA	Seriell	X	Gasdiode	Leistungsgleichrichter
V	50 mA	Seriell	Y	Diode	Einweggleichrichter
X	600 mA	Seriell	Z	Doppel-Diode	Zweiweggleichrichter
Y	450 mA	Seriell			
Z	0 V	ohne Heizung			

VORSICHT BEI DER ARBEIT MIT RÖHREN!

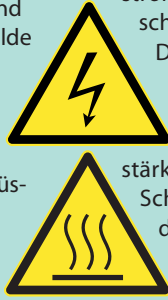
Bei der Arbeit mit Röhren lauern im Wesentlichen zwei Gefahren. Da ist zunächst einmal die hohe Temperatur, die so ein Glaskolben im Betrieb erreichen kann – bis zu 230 Grad Celsius sind bei einigen Röhren möglich und normal. Anfassen ist damit während des Betriebs strikt untersagt. Auch wärmeliebende Stubentiger sind von Röhren unbedingt fernzuhalten. Ansonsten gibt es ein verbranntes Fell.

Richtig lebensgefährlich sind jedoch die hohen Anodenspannungen. Hier ist äußerste Vorsicht und Sorgfalt beim Aufbau und der Inbetriebnahme von Schaltungen geboten: Niemals wilde Drahtverhaue hochziehen, sondern immer alles übersichtlich und sicher befestigt auf einem geerdeten Metall-Chassis aufbauen. Hilfreich sind dabei entweder Lötleistenaufbauten, wie sie gerne in Gitarrenverstärkern verwendet werden, oder fertige Platinen. Alle Spannungen müssen über Schmelzsicherungen abgesichert sein.

Bei Messungen an unter Spannung stehenden Aufbauten gilt die Regel: „Eine Hand in der Hosentasche, die andere

Hand für das Messgerät.“ Sprich: Niemals mit beiden Händen einen unter Spannung stehenden Aufbau anfassen, das kann tödlich enden. Beachtet man diese Grundregeln, dann kann man gefahrlos an Röhrenaufbauten arbeiten.

Noch ein Hinweis für die Leute, die sich mit alten Röhrengeräten beschäftigen möchten, und sie vielleicht restaurieren wollen: Bis zum Ende der sechziger Jahre wurden sogenannte Allstromgeräte gebaut. Solche Apparate sind der elektrotechnische Super-GAU: nicht geerdete Metall-Chassis, die je nach Drehrichtung des Netzstromsteckers unter voller Spannung stehen, dazu keine galvanische Trennung der Schaltung vom Lichtnetz. Von dieser Machart waren seinerzeit vor allem Fernseher, Radios und Bühnenverstärker; viele elektrische Unfälle auf Bühnen sind auf diese Schaltungstechnik zurückzuführen. Ich lasse grundsätzlich die Finger von solchen Geräten. Ihre Instandsetzung ist sinnlos, da sie ohnehin nicht mehr sicher betrieben werden können.



dig. Die Datenbank ist in Zwischenzeit so wichtig geworden, dass sie insgesamt achtmal weltweit gespiegelt wurde. Es ist die größte Röhren-Datenbank der Welt und meiner Meinung nach ein technisches Weltkulturerbe.

Einem solchen Datenblatt kann man als Erstes die Belegung der Sockelanschlüsse und ihre Zählrichtung entnehmen. Des Weiteren kennzeichnen folgende Begriffe die Betriebsparameter einer Röhre:

- R_a = zulässiger Lastwiderstand bei Eintakt-Schaltung
- R_{aa} = zulässiger Lastwiderstand bei Gegentakt-Schaltung
- U_a = Anodenspannung, bei der die angegebenen Werte erreicht werden
- I_a = Anodenstrom
- $-U_{g1}$ = Gittervorspannung

U_{g2-x} = Gitterspannungen bei Röhren mit mehreren Gittern

P_{max} = maximale zulässige Anodenverlustleistung
 U_f = Heizspannung

Viele Datenblätter enthalten darüber hinaus Kennlinien, die die Betriebsparameter einer Röhre bei unterschiedlichen Spannungen zeigen. Hier sind vor allem die deutschen Philips-Röhrendatenblätter hervorzuheben. Beispiel-Beschaltungen mit den passenden Bauteilwerten helfen, eine Röhre optimal einzusetzen.

An dieser Stelle verzichte ich auf weitere Erklärungen – ab der nächsten Seite geht es in die Praxis. Wer an mehr Informationen interessiert ist, findet im Kasten unten Literaturempfehlungen und über den Link eine Zusammenstellung von Internetquellen.

BEZEICHNUNGEN

Die Bezeichnung der Röhren ist im gesamten europäischen Raum einheitlich. In Russland und den USA gibt es andere Arten der Bezeichnung. Vor allem das russische System mit seinen kyrillischen Zeichen führt da sehr häufig zu Verwirrung.

VERTIEFENDE LITERATUR

Die folgenden Bücher sind meiner Ansicht nach die wichtigsten deutschen Quellen zu Elektronenröhren, auf die auch die meisten Neuerscheinungen zum Thema basieren. Die Originale sind allerdings zu 99 Prozent nur noch antiquarisch zu bekommen. Bei ihrer Beschaffung der Bücher ist das ZVAB (Zentrales Verzeichnis Antiquarischer Bücher) eine gute Anlaufstelle. Unter www.zvab.com wurde ich bisher fast immer fündig. Geduld und in gewissen Zeitabständen wiederholtes Suchen lohnen sich.

- H. Schröder: Nachrichtentechnik Band 1–3
- O. Diciol: Röhren-NF-Verstärker Praktikum

- H. Barkhausen: Lehrbuch der Elektronen-Röhren Band 1–4
- Telefunken: Telefunken Laborbuch Band 1–6
- K.H. Schubert: Das große Radiobastelbuch
- L. Ratheiser: Rundfunkröhren. Eigenschaften und Anwendung
- J. Schwandt: Röhren-Taschen-Tabelle



Netzgerät für Röhrenschaltungen

Kurzinfo

Zeitaufwand:
1,5 Stunden

Kosten:
40 Euro (ohne Gehäuse)

Elektronik:
solide Grundkenntnisse,
sicherer Umgang mit
Netzspannung

Messen:
robustes und leistungs-
fähiges Multimeter

Löten:
etwas Routine erforderlich

Schwierigkeitsgrad



Nach so viel Theorie folgt nun das erste praktische Projekt: ein kleines Netzteil, mit der die nachfolgend vorgestellten Röhrenschaltungen versorgt werden können.

Zentrales Bauteil ist ein Ringkerntrafo. Er generiert aus der Netzspannung drei Wechselspannungen:

~ 250 V / 100 mA

~ 6,3 V / 1,5 A

~ 12,6 V / 1 A

Der Ringkerntrafo ist der ideale Trafo für solche Anwendungen, da er ein sehr geringes Streufeld hat und von daher wenige Brummprobleme verursacht. Sollte ein anderer Trafo zum Einsatz kommen, so ist darauf zu achten, dass die Spannungen stimmen. Ansonsten gibt es Probleme mit den Ausgangsspannungen und der Spannungsfestigkeit der Bauteile.

Die drei Wechselspannungen werden generell über separate Sicherungen geführt. Diese sind geringfügig größer dimensioniert als die maximale Belastbarkeit der einzelnen Wicklungen. Der Grund hierfür sind die Anlaufströme der Röhren. Deren Heizungen sind Kaltleiter und ziehen im kalten Zustand bedeutend größere Ströme als später im Betrieb, wenn sie warm sind. Zu knapp ausgelegte Sicherungen müssten ständig getauscht werden.

Der Schalter S2 ermöglicht es, sowohl P-Röhren als auch E-Röhren mit der passenden Heizspannung zu versorgen. Das erleichtert die Beschaffung der Röhren, denn viele E-Röhren gibt es auch als P-Röhren. Sie unterscheiden sich nur in der Heizspannung. Mit R2 lassen sich Asymmetrien der Heizwendel kompensieren, die Heizungsbrumm verursachen würden. Das war schon der Heizspannungsteil des Netzteiles. Simpel, oder?

Etwas komplizierter wird es bei der Erzeugung der Anodenspannung. Nach der obligatorischen Sicherung erfolgt eine Gleichrichtung mit einem Brückengleichrichter und dem nachfolgenden Siebglied. Hierbei handelt es sich um eine sogenannte C-R-C-Siebung: C2-R1-C3 bilden einen Filter für die mit 100 Hz pulsierende Gleichspannung und machen

daraus eine saubere Gleichspannung mit nur wenigen mV Wechselspannungsanteil.

Der nachfolgende Spannungsteiler aus R3 und R4/C4 bildet das Massepotenzial für die Heizung. Normalerweise würde man einen Anschluss der Heizung direkt an Masse legen. Doch dieser Spannungsteiler hat den Vorteil, dass das Massepotenzial höher als der Wechselspannungsanteil der Röhrenheizung liegt. Somit kann kein Wechselstrom zur Kathode gelangen. Damit wird effektiv Heizungsbrummen verhindert.

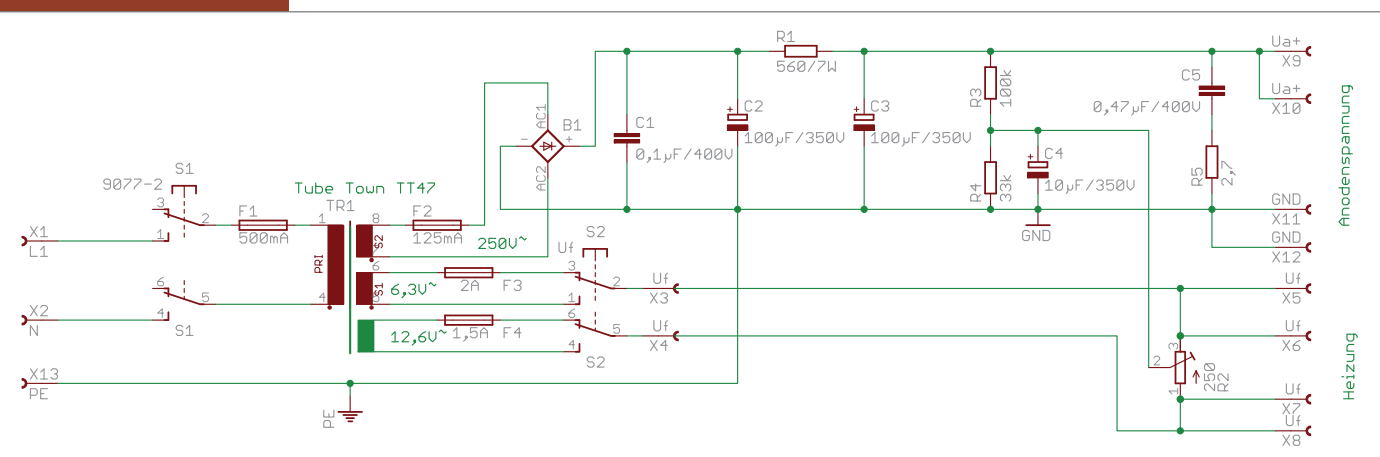
C1 und R5/C5 dienen dazu, Impulsspitzen zu filtern, die von den Halbleiterdioden des Gleichrichters verursacht werden. Diese Spitzen würde man ebenfalls als Brumm in der nachfolgenden Schaltung hören.

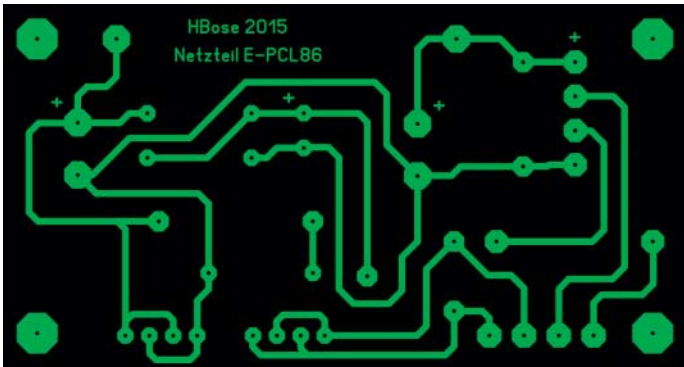
Was die Spannungsfestigkeit der Bauteile anbetrifft, muss weit über die eigentliche Betriebsspannung hinausgegangen werden. Wird die Anodenspannung mit etwa 80 mA belastet, so stellt sich eine Ausgangsspannung von rund 250 V ein (auf ein paar Volt mehr oder weniger kommt es bei Röhrenschaltungen nicht an). Ist das Netzteil unbelastet, steigt die Spannung auf über 300 V an, weil an R1 keine Spannung mehr abfällt. Das bedeutet, die Bauteile müssen die wesentlich höhere Leerlaufspannung aushalten können. Kritisch sind die Kondensatoren – je höher deren Spannungsfestigkeit, desto besser.

Eine Vergrößerung der Kapazität der Sieb-Kondensatoren ist meist ungünstig, denn je größer die Elkos sind, umso größer werden die Impulsspitzen der Halbleiter-Dioden und desto schlechter bekommt man sie herausgefiltert. Durch größere Kondensatoren wird auch der Trafo unnötig stark belastet, denn die Elkos wollen geladen werden. Je größer sie sind, desto mehr Leistung muss der Trafo dafür liefern.

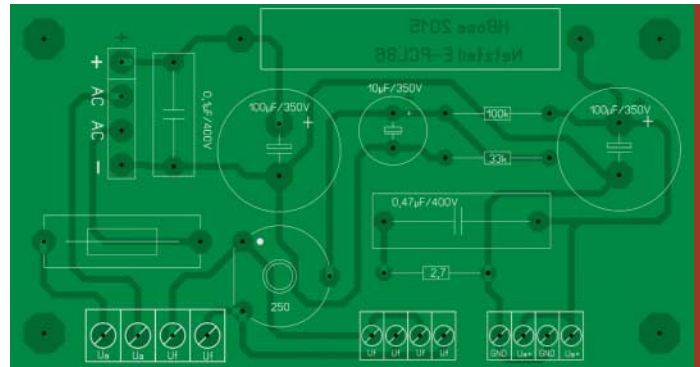
Die Anodenspannung ist – wie zuvor beschrieben – vom entnommenen Strom abhängig. Sollte diese zu groß sein, kann man die Spannung reduzieren, indem man R1 vergrößert. Hier hilft etwas Mathematik:

Schaltbild des Netzgerätes für Röhrenschaltungen





Die Netzteilplatine der Größe 100 mm × 53 mm von der Lötseite aus gesehen



Bestückungsplan des Netzteiles auf der anderen Seite der Platine

$$\text{Anodenspannung} = \text{Leerlaufspannung} - \text{Spannungsabfall}_{R1}$$

$$\text{Spannungsabfall}_{R1} = \text{Laststrom} \times R1$$

Mit diesen beiden einfachen Formeln lässt sich der Wert von R1 ausrechnen, um die Anodenspannung anzupassen.

Um einen anfängertauglichen und sicheren Aufbau zu gewährleisten, habe ich eine Platine entworfen (Download siehe Link im Kasten auf Seite 116). Auf ihr lassen sich fast alle Bauteile montieren. Die Ausnahmen sind die Sicherungen F1, F3, F4, der Trafo und die beiden Schalter – diese Bauteile gehören direkt in das Gehäuse. Die Platine und die restlichen Bauteile passen prima in ein Hammond-Alu-Gehäuse (siehe Stückliste, ebenfalls online). Beim Aufbau und beim Messen bitte unbedingt die Sicherheitshinweise beachten.

Wichtig ist vor allem die korrekte Erdung: Der Schutzleiter (PE) gehört unbedingt mit der Signalmasse (GND) verbunden. Schummeleien mit Widerständen oder Kondensatoren zwischen PE und GND sind lebensgefährlich! Dies wird meist gemacht, um Brummschleifen zu unterbrechen. Es ist aber bedeutend sicherer, die Ursache für die Brummschleife zu



suchen und zu beseitigen. Brummschleifen entstehen, wenn über die Masseverbindung ein Strom fließt. Ursache hierfür ist ein unsauberer Aufbau und eine schlechte Masseverdrahtung.

Das Netzteil ist in der Lage, zwei Röhrensaltungen mit allen Spannungen zu versorgen – zum Beispiel in einem Stereoverstärker. Damit sind wir beim nächsten Projekt.

Dieser Ringkerntrafo kommt im Netzgerät für Röhrensaltungen zum Einsatz.



Die fertig bestückte Platine des Netzteiles für Röhrensaltungen

Audioverstärker mit Verbundröhre PCL86

Kurzinfo

Zeitaufwand:
1,5 Stunden pro Kanal

Kosten:
40 Euro pro Kanal
(ohne Gehäuse)

Elektronik:
solide Grundkenntnisse,
sicherer Umgang mit
Netzspannung

Messen:
robustes und leistungs-
fähiges Multimeter

Löten:
etwas Routine erforderlich

Schwierigkeitsgrad



Der Ärger über die miesen PC-Lautsprecher an meinem Büroarbeitsplatz war der Auslöser, diesen Verstärker zu entwickeln. Das muss doch besser gehen, sagte ich mir, kramte ein altes Projekt aus und designte es für den Aktivboxenbetrieb um. Herausgekommen sind zwei kleine Aktivboxen mit Röhrenendstufe und Breitbandlautsprecher. Die Endstufe beschreibe ich hier, wer Informationen zu den Boxen braucht, kann sich gerne über die Make-Redaktion an mich wenden.

Als Röhre für den Verstärker kommt eine PCL86 zum Einsatz, eine Verbundröhre, die zwei Röhrensysteme beinhaltet und für Fernsehgeräte und Musiktruhen entwickelt wurde. Im Glaskolben befinden sich sowohl eine Triode als Vorverstärker und eine Pentode als Endverstärkerröhre. Das Ganze gibt es in zwei Varianten als ECL86 und als PCL86. Beide Röhren sind gleich bis auf ihre Heizspannung – und den Preis: Eine ECL86 kostet das Zehnfache einer PCL86; das verstehe wer will. Da das zuvor beschriebene Netzgerät beide Typen mit Heizspannung versorgen kann, kann uns der Preis egal sein.

Es handelt sich um einen zweistufigen Verstärker in Eintakt-A-Schaltung. Das Trioden-System der PCL86 verstärkt das von einem PC oder einem CD-Spieler kommende Musiksinal und bringt es auf einen Pegel, mit dem die Pentode der PCL86 dann gesteuert wird. Da die Röhrenausgangsspannung eine sehr hochohmige Spannungsquelle darstellt, würde nun ein direkt angeschlossener Lautsprecher zu einem Kurzschluss führen. Deshalb haben die Röhrentechniker den Ausgangsübertrager erfunden. Dieser Übertrager hat in unserem Fall auf der der Röhre zugewandten Seite einen Wechselspannungswiderstand von 5,2k Ω ; damit ist alles wieder gut. Die dem Lautsprecher zugewandte Seite hat hingegen nur 8 Ω , optimal für einen Lautsprecher.

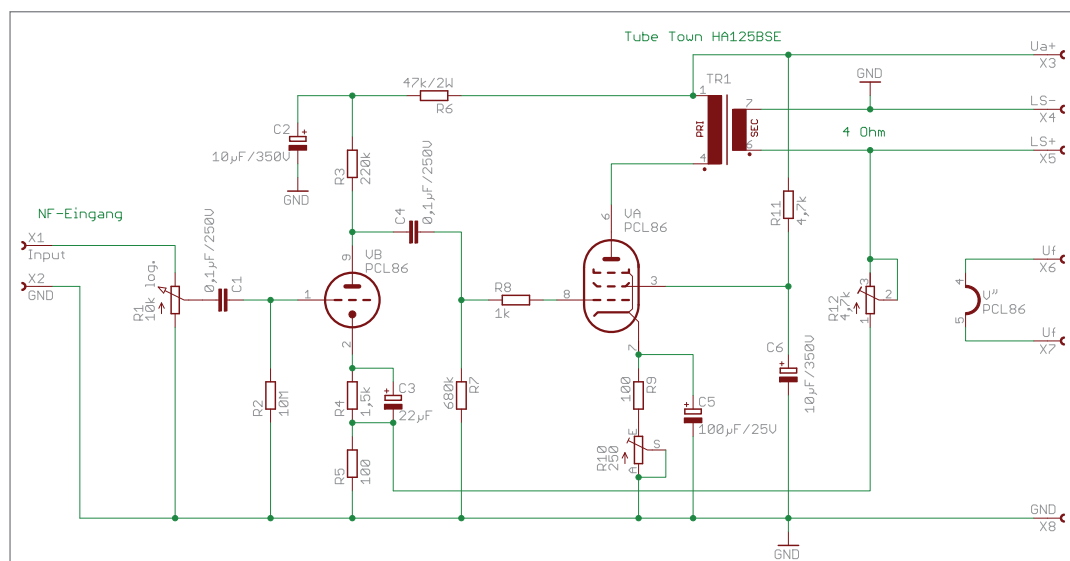
Der Übertrager ist ein Impedanzwandler, der eine hochohmige Spannungsquelle in eine niederohmige übersetzt. Anders als bei einem Netztrafo werden die Wicklungen der Primär- und Sekundärseite ineinander verschachtelt. So etwas ist nur mit viel Aufwand zu fertigen, was den sehr hohen Preis dieses Bauteiles begründet. Ein weiteres Qualitätskriterium ist der Eisenkern (Blechpaket) des Übertragers. Hier gilt: je mehr Eisen, desto besser.

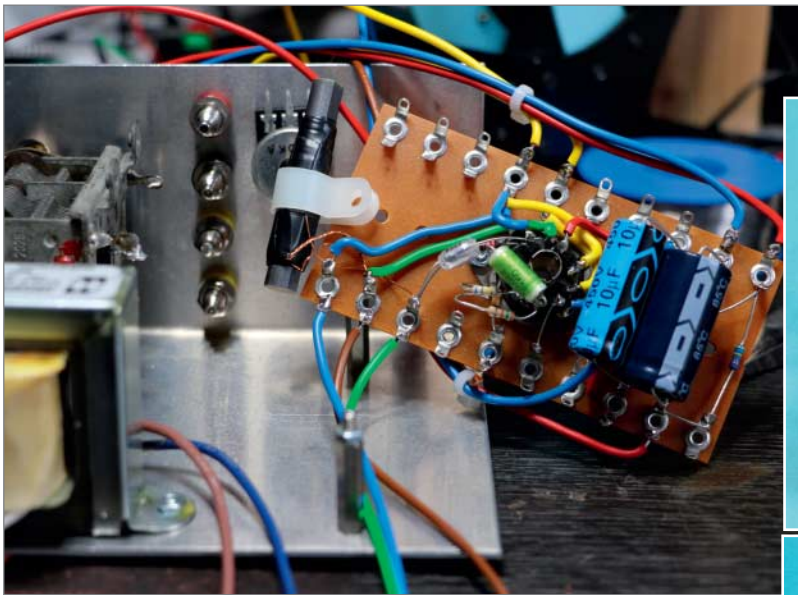
Bei diesem Projekt kommt ein Ausgangsübertrager der Firma Hammond zum Einsatz, der für unterschiedliche Primärwiderstände konfiguriert werden kann. Hier verweise ich auf das Datenblatt bei der Firma Tube-Town, die den Übertrager in Deutschland liefert.

Unser Verstärker soll sich gut anhören, deshalb wird er nicht in die magnetische Sättigung gesteuert. Hierbei hilft uns das Poti R10: Es begrenzt den Strom, der durch den Übertrager fließen kann. Optimaler Wert bei unserer Röhren-Übertrager-Kombination sind 40 mA bei $U_a = 250$ V. Dieser Wert kann über R9 gemessen werden.

Um Verzerrungen weiter zu unterdrücken, die dem Musiksinal auf seinem Weg durch den Verstärker beigemischt werden, verfügt die Schaltung über eine Gegenkopplung: Teile des Ausgangssignales werden phasenverschoben dem Eingangssignal beigemischt. Dies führt dann zu einer Auslöschung der unerwünschten Verzerrungen. Mit dem Poti R12 lässt sich der Grad der Beeinflussung einstellen. Dies sollte mit Fingerspitzengefühl geschehen: Zu viel Gegenkopplung lässt den Verstärker dumpf klingen. Zu wenig kann zu einer Mitkopplung führen und damit zum Schwingen des Verstärkers. Diese Schwingungen können bis hoch in den UKW-Bereich reichen. Ich habe meine Verstärker bei Gleichtarbeiten schon des Öfteren un-

Schaltbild des Röhrenverstärkers für einen Kanal





Der Schaltungsaufbau des Prototypen auf der Löt­leiste, von unten gesehen



Das Audion als Aufbau auf einem Testchassis, einmal von vorne und einmal von der Rückseite gesehen

der Empfänger mit HF-Müll zugestopft wird. Ich habe drei Tage nach einem Schaltungsfehler gesucht, bis ich in der Schreibtischschublade eine alte Funkmaus fand, die munter vor sich hin sendete. Shit Happens.

Für die Fotos habe ich die wenigen Bauteile der Röhrenscha­lung auf eine Löt­leiste gesetzt, um auch das mal zu zeigen. Wer das so nachbauen will, den bitte ich um äußerste Vorsicht beim Hantieren: Alle Drähte so kurz wie möglich halten und die Leiste auf einem geerdeten Metallchassis befestigen. Eine gute Erdung ist ohnehin für den Radioempfang notwendig.

Soweit der erste Teil des Ausflugs in die faszinierende Welt der Elektronenröhren. Der nächste

Teil folgt in einer der kommenden Make-Ausgaben, dann geht es um Kaltkathoden-Anzeigeröhren, die auch Nixies genannt werden. Und natürlich gibt es dazu wieder eine Schaltung zum Nachbauen.

RADIOEMPFANG MIT RÖHREN

Der Rundfunk ist das wichtigste zivile Betätigungsfeld für die Elektronenröhren, neben ihrem Einsatz in der Telefontechnik und beim Militär. Schon 1906 ließ die kaiserliche Regierung von Telefunken eine Großsendeanlage in Nauen bei Berlin errichten. Sie diente zunächst dazu, mit einem mechanischen Funkensender Telegrafiesignale in die Überseekolonien zu senden.

Die hierbei gewonnenen Erfahrungen wurden dann für die Radiotechnik eingesetzt, als die erste Rundfunksendeanlage auf dem Funkenberg in Königs Wusterhausen errichtet wurde. Der Probebetrieb startete am 22. Dezember 1920 und wurde 1923 zum Regelbetrieb mit festem Programm. Gesendet wurde im Langwellenfrequenzbereich, was zu einer enormen Reichweite führte.

Die ersten Probesendungen konnten in Deutschland allerdings nicht gehört werden, da das Radiohören bei Strafe verboten war.

Aber in Luxemburg, England und Skandinavien waren die Sendungen des deutschen Rundfunksenders gut empfangbar, was die vielen Zuschriften aus diesen Ländern belegten. Wegen der besseren Verständlichkeit wurde bald auf den Mittelwellenbereich umgerüstet, was zwar die Reichweite verringerte, aber die Tonqualität verbesserte.

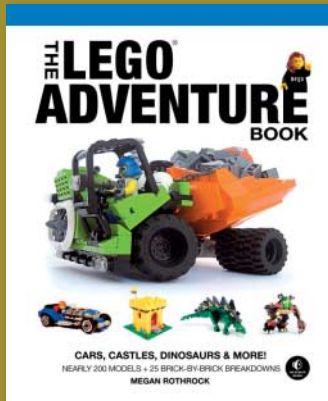
Die ersten Radioempfänger benutzten noch keine Röhren und bestanden aus einer Spule, einem Drehkondensator, einer Halbleiterdiode und einem hochohmigen Kopfhörer. Sie wurden Detektor-Empfänger genannt und es war eine große Kunst, sie für den Empfang der abendlichen Nachrichten einzustellen. Aber schon bald benutzte man Röhren für den Radioempfang – etwa in Form eines Audions, wie es hier zum Nachbau vorgestellt wird.

—pek

THE LEGO ADVENTURE BOOK

Cars, Castles, Dinosaurs & More

Wollten wir nicht alle schon mal die Welt-herrschaft an uns reißen? Eine Roboter-armee in den Kampf gegen Dinosaurier schicken? Mit dem LEGO Adventure Book kommen wir dem Traum ein Stück näher.



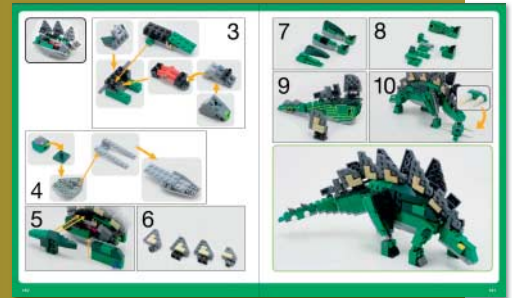
Die Autorin nimmt uns als Legofigur Megs mit auf ihre Abenteuer. Dabei zeigt sie uns professionelle Legobauer und ihre Techniken. Zu Beginn bauen wir ein „Ideenlabor“, dann einen Roboterassistenten und schließlich vielseitige Welten.

Dabei zeigt das Buch anschaulich, wie man ein Basisdesign variieren kann: Mit anderen Farben wird aus dem Café eine Bäckerei und aus dem Polizeiwagen ein Stadtauto.

Das Adventure Book ist eine Sammlung von Bauanleitungen. Diese bestehen, wie man es aus Legoanleitungen kennt, rein aus Bildern. Nur kurze comichaft Sequenzen und Megs Notizen sind schriftlich. So ist das Buch leicht zu verstehen, auch wenn man (noch) nicht lesen kann.

Wer die einfallsreichen Techniken käuflicher Legosets bewundert, kommt hier auf seine Kosten. Lego-Designer verraten

ihre Tipps und Tricks kinderfreundlich aufbereitet – und gleichzeitig spannend für Erwachsene. —rehu



Autor	Megan Rothrock
Verlag	No Starch Press
Umfang	200 Seiten
ISBN	978-1-59327-442-9
Preis	16,95 €

ROBOT BUILDER

The Beginner's Guide to Building Robots

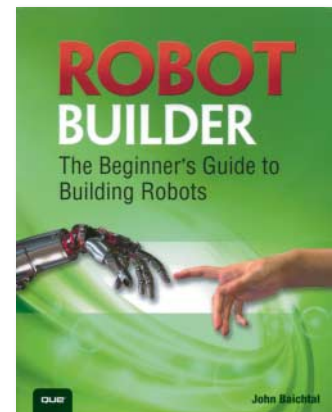
Dem Thema Roboter haben wir in diesem Heft mehrere Artikel in unterschiedlichen Schwierigkeitsgraden gewidmet. Anfänger, denen dieser Einstieg noch nicht reicht, sollten sich das englischsprachige Buch von John Baichtal zu Gemüte führen. Das Buch ist allerdings etwas wirr strukturiert und sollte nicht als Leitfaden zur Einarbeitung in das Thema gesehen werden. In den zwölf Kapiteln springt der Autor inhaltlich hin und her. Trotzdem enthält das Buch viele wert-

volle Informationen, die man als Einsteiger über Roboter wissen sollte.

Nach einer kurzen Einführung in die Komponenten eines Roboters, lernt man am Ende des ersten Kapitels beispielsweise, wie man einen per Vibrationsmotor angetriebenen Roboter baut. Das nächste Kapitel gibt eine Übersicht über diverse Modelle aus der Maker-Szene mit kurzen Beschreibungen. Das inspiriert, bietet aber kaum Infos zur eigenen Umsetzung. Die nächsten

Kapitel widmen sich Antrieben, Solarenergie, einer Einführung ins Löten sowie der Steuerung von Robotern mit Arduino. Der Einstieg in die Arduino-Programmierung folgt jedoch erst ein Kapitel später. Es folgt ein Ausflug in die Datenübertragung per Infrarot sowie eine Übersicht über Robotik-Bausets und wie man einige davon zusammenbaut. Kapitel 10 führt wichtige Werkzeuge für den Selbstbau auf.

Besonders gelungen fand der Rezensent die Übersicht über



Manipulatoren sprich Greifer und wie man sie sich aus Lego oder Holz selbst baut. Das Buch schließt mit dem Selbstbau eines Schwimmroboters und der Vorstellung verschiedener Kunstroboter ab, die beispielsweise die Straße bemalen oder besprühen. Die Texte sind locker geschrieben, die Erklärungen mitunter etwas oberflächlich. Das Buch liefert jedoch eine interessante Mischung aus Theorie und Praxis. —dab

Autor	John Baichtal
Verlag	Que Publishing
Umfang	412 Seiten
ISBN	978-0-7897-5149-2
Preis	34,99 US-\$



GETTING STARTED WITH ADAFRUIT FLORA

Making Wearables with an Arduino-Compatible Electronics Platform

Endlich Kräfte wie die Superhelden dank tragbarer Gadgets – so beginnt das neue Buch von Adafruit's Becky Stern über ihr Flora-Board. Speziell für Wearable-Projekte entwickelt, erklärt das Buch den Aufbau, die

passenden Sensoren und Leuchtdioden sowie das nötige Werkzeug von der Nadel bis zum Lötkolben. Nach einem Einstieg in die Programmierung gibt es drei Projekte: Shorts mit Blinkfisch, einen leuchtenden Rock und eine GPS-Uhr.

Neben dem kompakten Überblick sind vor allem die Hinweise zum Umgang mit leitfähigem Garn hilfreich, der sowohl Näh- als auch Elektronikfans vor Herausforderungen stellt. Damit erschöpft sich allerdings der Nutzen des Buches, das ein wenig lieblos aufbereitet wirkt. Die Bilder sind alle ähnlich eingefügt, im ersten Kapitel falsch nummeriert und zwei kurze Zeilen Code verirren sich auch mal auf die nächste Seite. Hier hätte noch jemand ein prüfendes Auge auf das Buch werfen sollen. —hch

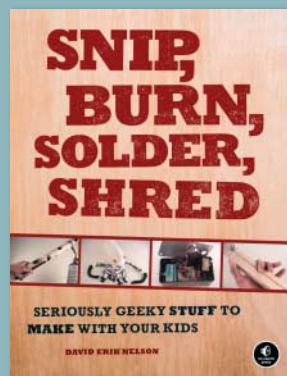


Autor	Becky Stern und Tyler Cooper
Verlag	O'Reilly
Umfang	111 Seiten
ISBN	978-1-4571-8322-5
Preis	13,00 €

SNIP, BURN, SOLDER, SHRED

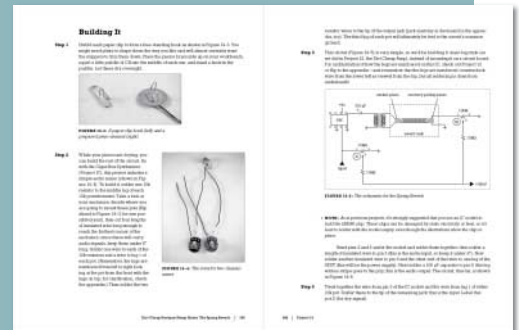
Seriously Geeky Stuff to make with your Kids

In diesem Buch sind 24 Bastelprojekte versammelt, die der Autor als Lehrer an einer freien Schule im Unterricht an unmotivierten Teenagern getestet hat. Vom PVC-Tipi, über Gesellschaftsspiele und elektrische Instrumente bis zu Boomerangs, Wasserraketen und einfachen Robotern ist hier alles vertreten. Alle Projekte entsprechen, laut dem Autoren, drei Qualitätskriterien. Erstens: Man baut immer ein großartiges Ding. Zweitens: Das Ding kostet kaum etwas oder es ist gratis. Drittens: Beim Bau des Dings erlernt man Techniken und Kenntnisse, die es erlauben, weitere großartige Dinge zu bauen.



oder nichts kosten, ist eine gut ausgestattete Werkstatt. Die Projekte sind in drei Themenbereiche unterteilt: verschiedene Spielzeuge, Musikprojekte und Flieger beziehungsweise Wurfgeschosse. Innerhalb der Bereiche sind die Basteleien nach dem Schwierigkeitsgrad geordnet, für keines braucht man Vorkenntnisse. Die meisten Basteleien sind für Kinder ab 12 Jahren geeignet – sie können mit diesem Buch Grundlagen im Nähen, Tischlern und der Elektronik erlernen. In den übersichtlichen und ausführlichen Anleitungen findet man wertvolle zusätzliche Informationen zu den verwendeten Werkzeugen und Techniken. Die Gestaltung des gesamten Buchs in Schwarz-Weiß tut der Übersichtlichkeit gut, macht es hier und da aber

eben etwas farblos. Für alle, die mit Gruppen von Kindern werken, ist dieses Buch eine absolute Empfehlung. Bisher ist es nur auf Englisch erhältlich, die Mühe des Umrechnens von Maßen und Nachschlagen von Fachvokabeln ist es aber sicherlich wert. —esk



Autor	David Erik Nelson
Verlag	no starch press
Umfang	359 Seiten
ISBN	978-1-59327-259-3
Preis	18,50 bis 20,00 € (je nach Anbieter)

Transistortester

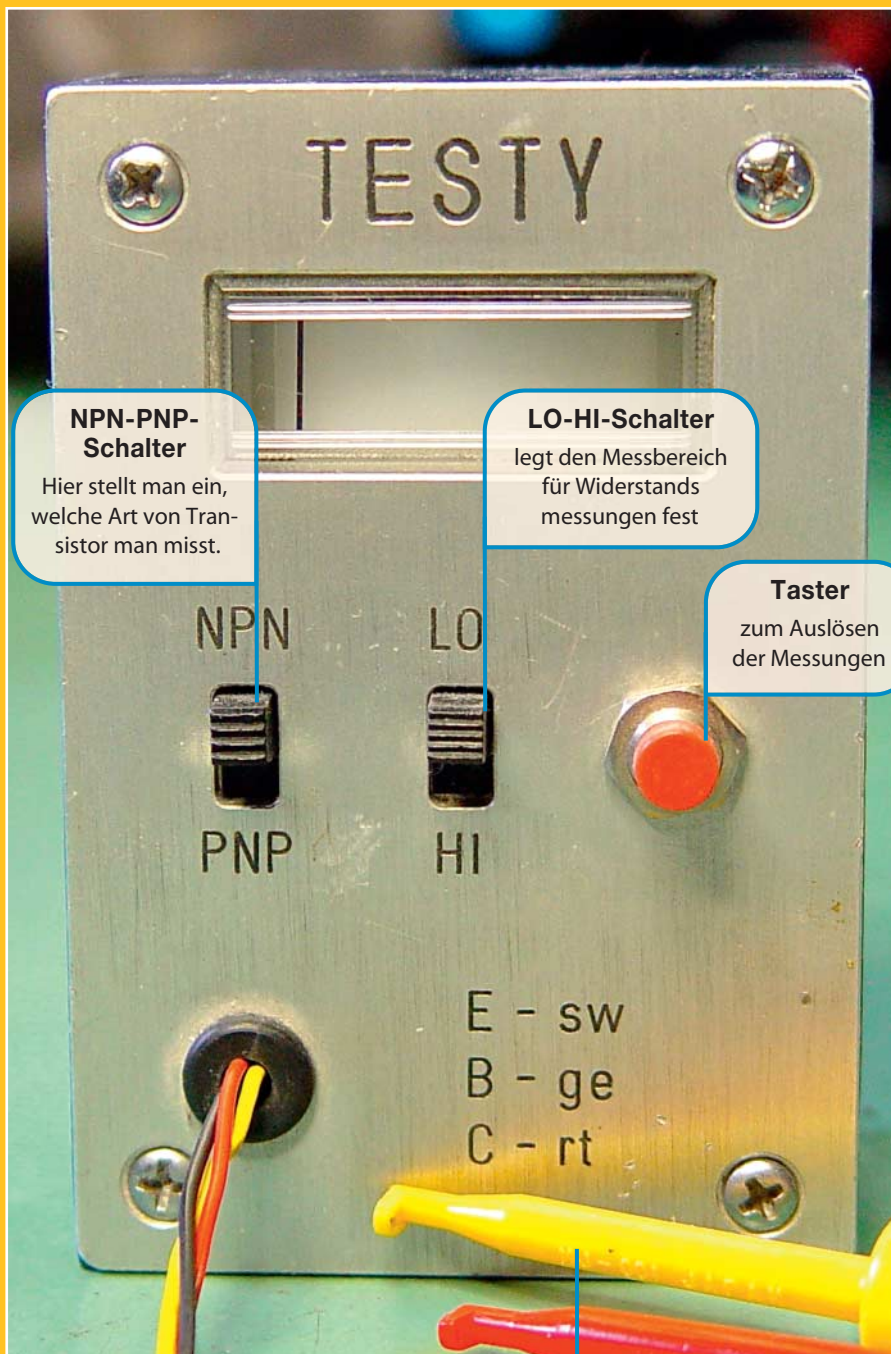
von Manfred Kienzle

In meiner Ausbildung zum Radio- und Fernsehtechniker hatte ich ein Ohm-Meter – Zapfelmax genannt. Ein äußerst praktisches Werkzeug, um Durchgang und Schlüsse zu messen. Mein Eigenbau-Transistortester „Testy“ kann das auch: Im LO- und HI-Bereich liegt der Endausschlag bei 0Ω , die Mitte liegt bei LO bei circa $3 \text{ k}\Omega$, bei HI sind es ungefähr 15Ω zwischen der roten und der schwarzen Klemme. Natürlich kann mein Transistortester aber noch mehr. Mit ihm kann man Dioden und LEDs testen – letztere leuchten auf, wenn man am NPN-PNP-Schalter die richtige Einstellung wählt. Muss man bei einem Transistor ermitteln, welches Beinchen welcher Anschluss ist, kann man das mit Testy tun. Man misst die 6 möglichen Durchlassstrecken zwischen jeweils zwei Beinchen (siehe Messschema in der c't Hacks 4/2014 auf S. 109).

Zwischen zwei der Beinchen kann man in keiner Richtung eine Dioden-Durchlassspannung messen – das sind Kollektor und Emitter. Das übrige Beinchen muss also die Basis sein. Um jetzt Emitter und Kollektor definitiv festzulegen, misst man an der Basis und einem der anderen Beinchen. Über den Taster gibt man Basisstrom auf die Basis und misst an den beiden anderen Anschlüssen – der Kollektorstrom ist um die Stromverstärkung B höher.

Bei manchen altherwürdigen Germanium-Transistoren sieht man ohne gedrückten Taster bereits „Restströme“ – mit gedrücktem Taster zeigt sich dann häufig doch eine Stromverstärkung. Auch kleine Thyristoren können mit Testy getestet werden, nach dem Tastendruck bleibt der Ausschlag stehen. Für meine Messungen steht Testy meist auf NPN und LO (für kleine Transistoren). Mit HI misst man die dicken Brummer.

Wer sich selbst einen Transistortester bauen möchte, findet in den Links einen Schaltplan von Testy und eine Liste der notwendigen Materialien. —esk



NPN-PNP-Schalter
Hier stellt man ein, welche Art von Transistor man misst.

LO-HI-Schalter
legt den Messbereich für Widerstandsmessungen fest

Taster
zum Auslösen der Messungen

Drei Prüfspitzen
zum Ermitteln, welcher Anschluss welcher ist

↓ **Links und Foren**
make-magazin.de/x3d5

Dein Lieblingswerkzeug
Welches Werkzeug macht Dir das Leben leichter? Schick uns Deine Idee an:
mail@make-magazin.de

Anzeige

Anemometer mit Windfahne

Ob mobil in der Modellfliegerei oder als dauerhafte DIY-Ergänzung für den Balkon – dieser Windmesser lässt sich flexibel einsetzen. Günstig ist er auch noch. Benötigt werden nur Holzreste, Haushaltsmaterialien und ein Fahrradcomputer.

von Burkhard Fleischer



Kurzinfo



Zeitaufwand:
5 Stunden



Kosten:
ab 2 Euro



Holzbearbeitung:
Sägen



Metalbearbeitung:
Anschleifen und Bohren

Schwierigkeitsgrad

leicht schwer

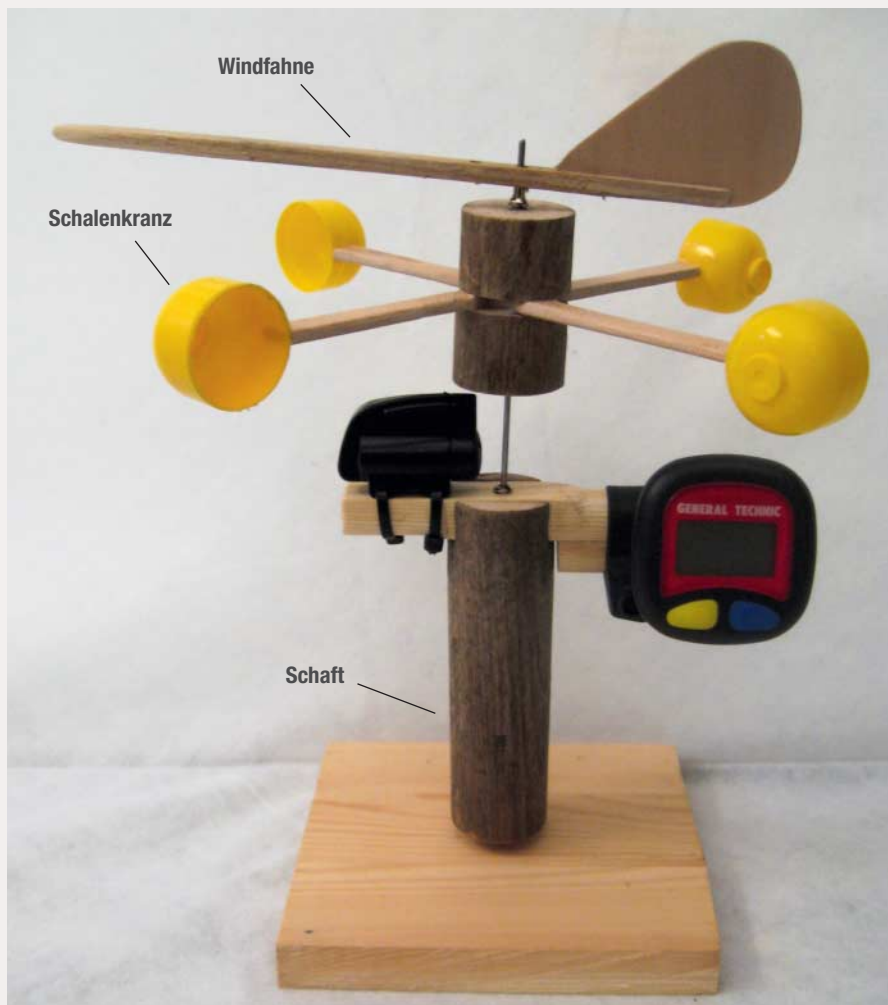
Einkaufsliste

- » Fahrradcomputer
- » Fahrradspeiche mit 2 Nippeln
- » 4 halbe Überraschungseier-Schalen
- » Rundmagnet (5 × 25 mm)
- » Messingrohr
- » Holzreste
- » Kreuzschlitzschraube
- » Klebstoff

Windmesser gibt es in vielen Ausführungen zu kaufen, dabei lassen sie sich auch einfach selber herstellen, angepasst an die eigenen Bedürfnisse. Ich habe spezifische Anforderungen an einen Windmesser: Er soll stationär, aber auch mobil eingesetzt werden. Es sollen Windgeschwindigkeiten in Höhen bis zu 2,50 Metern gemessen werden können. Ferner möchte ich wissen, woher der Wind weht. Und natürlich: Das Gerät darf kein Vermögen verschlingen. Für den Eigenbau eignen sich Konstruktionen basierend auf drei mechanischen Prinzipien: Das Schalenkranzanemometer, das Flügelradanemometer und die Wildsche Windstärketafel.

Das letzte Modell ist am einfachsten zu realisieren, aber auch das ungenaueste. Eine Tafel, die auch eine Spielkarte sein kann, wird an einer Kante drehbar gelagert. Bläst der Wind die Tafel an, so wird sie um einen bestimmten Winkel ausgelenkt. Der Auslenkwinkel korreliert mit der Windgeschwindigkeit und kann auf einer Skala abgelesen werden. Zum Ablesen muss sich das Gerät stets in Augenhöhe befinden – für eine Windmessung in 2,50 Meter Höhe benötigt man hierzu eine Leiter. Einen Ausweg böte die Verwendung eines Feinschleifwiderstandes, mit dem der Auslenkwinkel der Windstärketafel analog erfasst werden könnte. Aber dieser Widerstandstyp ist für den Bastler nur schwer zu beschaffen.

Mir fiel ein alter Netzteil Lüfter in die Hand. Mit ihm hätte ich rasch ein Flügelradanemometer basteln können. Denn der Lüftermotor kann nicht nur eine Last antreiben, sondern wird zum Generator, wenn er von außen bewegt wird. Das funktionierte auch. Fuhr ich mit dem Lüfter in der Hand Fahrrad, so erzeugte der Generator linear zur Geschwindigkeit eine Spannung. Bei 10 Stundenkilometern maß ich



Das Anemometer und seine Bestandteile

1 Volt, bei 25 Stundenkilometern 2,5 Volt. Unter 10 Stundenkilometern blieb er allerdings stehen, der Anlaufwiderstand war zu groß. Diese zunächst so einfach erscheinende Lösung eignete sich für meine Zwecke somit nicht.

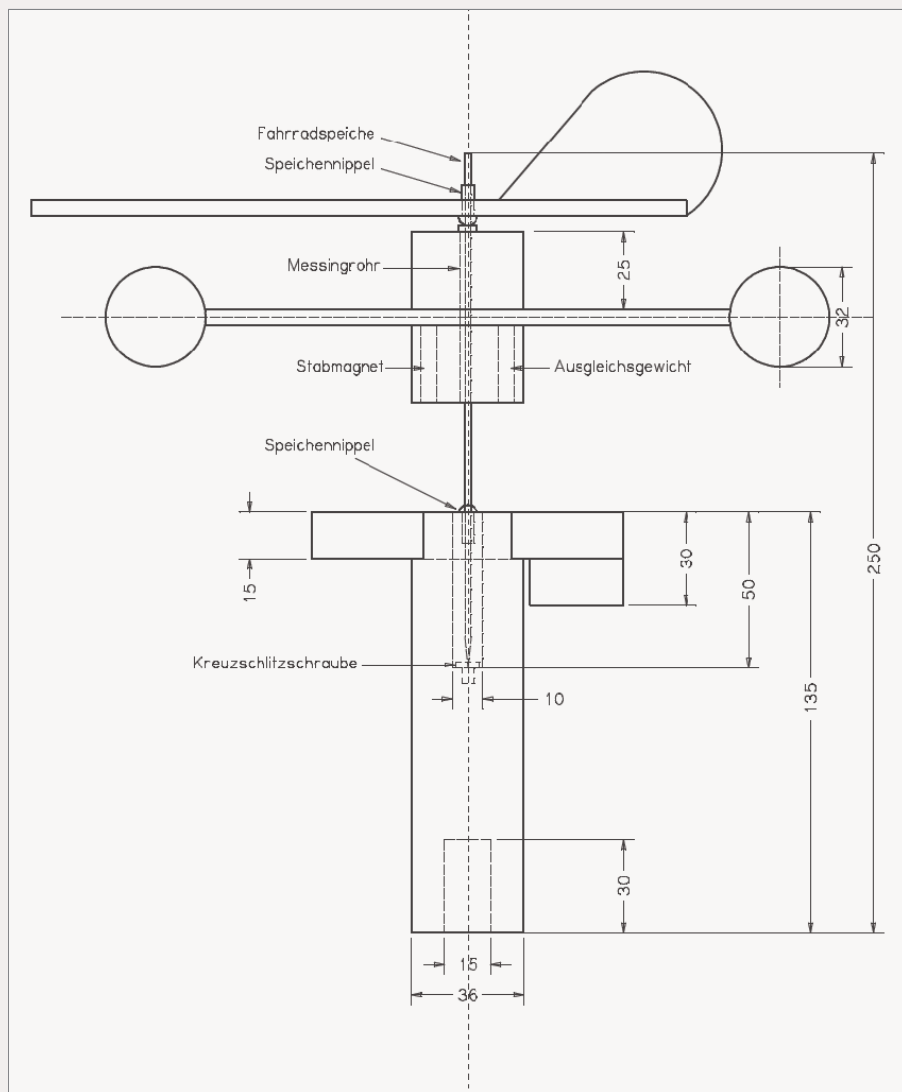
Bleibt das Schalenkranzanemometer übrig. Wie auch beim Lüftermodell können die Daten analog erfasst werden: Der Schalenkranz ist auf der Achse eines Generators montiert und die erzeugte Spannung korreliert mit der Drehgeschwindigkeit.

TIPP

Für die Windschalen reichen zwei Überraschungseier aus. Um Unwuchten zu vermeiden, sollten die oberen und unteren Hälften jeweils gegenüber angebracht werden. Auch halbe Tischtennisbälle eignen sich.

Wir haben das Anemometer nachgebaut und dabei ein Esstäbchen und ein Pappstück in die Windfahne verwandelt. Etwas Klarlack schützt gegen Feuchtigkeit.





Die Konstruktion hat ein handliches Format. Beim Nachbau können die Bauteile je nach vorhandenem Material aber von den Maßen abweichen.

keit. Die Erfahrung mit der Lüfterlösung hätte es jedoch nötig gemacht, einen leicht laufenden Generator anzuschaffen. Die digitale Lösung, die Impulshäufigkeit des sich drehenden Schalenkranzes zu messen, erschien mir einfacher. Es handelt sich hierbei um eine massenhaft produzierte und damit preiswerte Technik: Jeder Fahrrad-Computer arbeitet nach diesem Prinzip.

Mein Windmesser hat 1,95 € gekostet. Das ist der Preis für einen Fahrrad-Computer, den ich bei Pollin erstanden habe. Alle anderen Materialien fand ich in meinem Materialarchiv: Griff und Schalenkranznabe

waren mal ein Tischbein, die Windschalen waren Verpackungen für Spiele in Überraschungseiern, Holzleisten waren zu Silvester mit Raketen in den Himmel geschossen worden, alte abgerissene Fahrradspeichen samt Nippeln waren bereits als Schrott aussortiert worden. Diese Materialien sind nicht obligatorisch. Die Phantasie ist grenzenlos.

Konstruktion

Die Einzelheiten der Konstruktion sind in der Zeichnung aufgezeichnet. Beim Nachbau ist zu beachten, dass die Bohrungen

möglichst zentrisch erfolgen und Schnitte möglichst rechtwinklig. Tischbohrmaschine, Maschinenschraubstock und Gehrungssäge sind hierbei große Hilfen. Das Schalenkreuz ist aus einem 4 Millimeter starken Sperrholzrest ausgesägt. Die Schalen sind stark angeraut und mit Pattex stumpf an das Schalenkreuz geklebt.

Die Lagerung des Schalenkranzes ist als Spitzenlager ausgebildet. Die Achse aus einer alten Fahrradspeiche wird hierzu in die Tischbohrmaschine eingespannt und mit einer Eisenfeile angespitzt. Als unteres Lager dient eine Kreuzschlitzschraube, die ursprünglich mal in einem PC-Gehäuse saß. Der Schraubenkopf ist nach unten hin kegelförmig geformt. Für das obere Lager ist ein Speichennippel so weit aufgebohrt, dass die Achse sich leicht dreht, ohne dabei allzu sehr zu klappern.

Die Windfahne ist aus einem Silvester-Raketenstab und einem 1 Millimeter starken Sperrholzrest gebaut. Das Achslager ist auch hier wieder mit einem aufgebohrten Speichennippel ausgeführt worden. Es befindet sich im Schwerpunkt der Windfahne. Im Betrieb dreht sich der Schalenkranz ständig mit der Speiche, während die Windfahne stur stehend die Windrichtung anzeigt. Da beide Bauteile sich die Drehachse teilen, könnte die stehende Windfahne den drehenden Schalenkranz bremsen. Dies mag bei sehr niedrigen Windgeschwindigkeiten auch der Fall sein, bei höheren ist die Reibung, die durch eine Unterlegscheibe etwas verringert wird, eher unbedeutend, wie Versuche ergaben. Zwischen Messungen mit aufgesetzter und entfernter Windfahne gab es keine signifikanten Unterschiede.

POLYPROPYLEN

Die gelben Schalen bestehen aus dem Kunststoff Polypropylen (PP). Er lässt sich schwer kleben und ist beständig gegen Fette, viele Lösungsmittel, Säuren und Laugen. Deshalb wird er so oft als Verpackung für Lebensmittel eingesetzt. Oder als Verpackung im Überraschungsei.

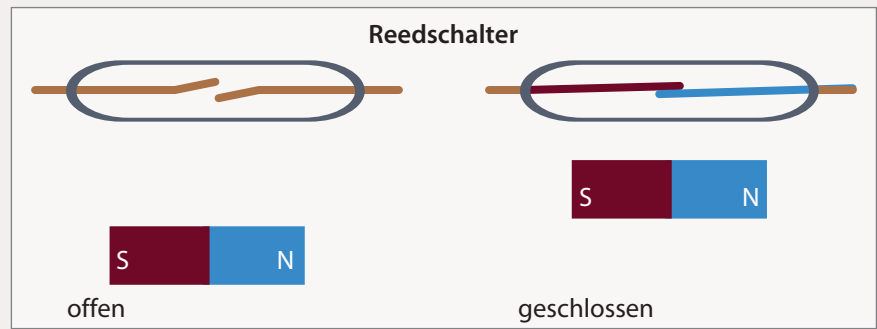
Die 15-mm-Bohrung im Schaft ermöglicht es, den Windmesser entweder bequem auf einer Platte mit eingelassenem 15-Millimeter starkem Holzdübel abzustellen, oder ihn auf einen 15-Millimeter starken längeren Holzstab aufzustecken. Ist der Stab lang genug, können auch Messungen in luftiger Höhe bequem ausgeführt werden.

Messtechnik

Geschwindigkeiten (v) werden berechnet, indem eine zurückgelegte Strecke (s) durch die vergangene Zeit geteilt wird:

$$v = \frac{s}{t}$$

Bei Fahrrädern wird als Strecke der Reifenumfang genutzt und durch die Zeit einer Radumdrehung geteilt. Um die Geschwindigkeit mittels eines Fahrrad-Computers zu messen, muss der Radumfang des eigenen Rades manuell eingegeben werden. Die Zeit einer Radumdrehung wird elektronisch gestoppt. Hierzu ist an einer Speiche ein kleiner Magnet montiert und an der Fahrradgabel ein Reedschalter. Bei jeder Umdrehung schließt der Magnet den Reedschalter und es wird ein kurzer Impuls an den Fahrradcomputer gesandt, den dieser zu unterschiedlichen Messungen auswerten kann. Die Verbindung zwischen Reedschalter und Fahrrad-Computer kann per Kabel oder auch kabellos erfolgen. Bei der kabellosen Version ist der Reedschalter zusammen mit



Links ist der Reedschalter offen. Kommt ein Magnet in seine Nähe, schließt das Magnetfeld die Kontaktzungen.

einem Sender gefertigt und im Fahrradcomputer ist ein kleiner Empfänger als Gegenstück untergebracht.

Der Pollin-Fahradcomputer ist so ein kabelloses Exemplar. Sowohl Sender als auch Empfänger sind auf kurzer Distanz auf einer Querleiste am Griff montiert. Als Signalmagnet schob ich einen 25-Millimeter langen Rundmagnet (5-Millimeter Durchmesser) in eine Bohrung der Schalenkranznabe. Ich verzichtete auf den Originalmagneten, weil der Rundmagnet einschließlich Ausgleichgewicht sich besser montieren ließ. Die beiden Teile presste ich jeweils in eine Bohrung. Bis hierher griff ich auf Fahrrad-Technik zurück. Es wäre zu einfach, könnte diese Technik auch auf die

Messungen übertragen werden. Während ein Fahrradreifen per Haftreibung laufend einseitig mit dem Asphalt verbunden ist, greifen auf das Windrad unterschiedliche Kräfte gleichzeitig an verschiedenen Punkten an (vgl. Kasten unten).

Die Windgeschwindigkeit (v) kann also aus der Drehgeschwindigkeit (u) und einer Gerätekonstante (k) des Schalenkranzes berechnet werden:

$$v = u \cdot k$$

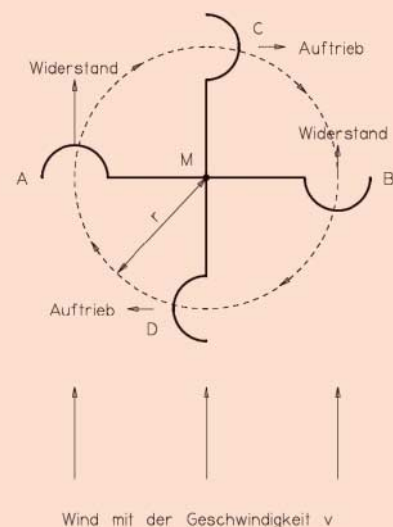
Kalibrieren und Einsatz

Im Netz werden viele Bemühungen diskutiert, die Windgeschwindigkeit theoretisch

SO DREHT SICH DER SCHALENKRANZ

Vereinfacht nehmen wir an, dass der Wind überall mit der gleichen Geschwindigkeit bläst. Er trifft zum einen auf die beiden Halbschalen A und B – dabei hat die offene Halbschale einen größeren Widerstandsbeiwert (c_w -Wert) als die geschlossene. Es gilt somit, dass der Widerstand der Halbschale A größer ist als der von B ($W_A > W_B$). Dadurch sind die Drehmomente unterschiedlich. Entsprechend dreht sich das Schalenkreuz um den Mittelpunkt M ($W_A \cdot r > W_B \cdot r$).

Verstärkt wird die Drehbewegung zum Anderen durch im rechten Winkel zur Windrichtung wirkende Kräfte der Halbschalen C und D. Von vorne angeblasene gebogene Flächen erzeugen eine quer zur Windrichtung wirkende Kraft in Wölbungsrichtung, den Auftrieb. Dies ist auch das Prinzip, nach dem Flugzeuge fliegen.





Ob mit oder ohne Kabel – bei stationärem Einsatz lässt sich der Fahrradcomputer auch abseits des Anemometers befestigen.

zu ermitteln. Experimentell wird vorgeschlagen, den Selbstbauwindmesser mittels eines Referenzgerätes (Fahrrad-Computer, Autotacho) zu kalibrieren.

Ich wählte eine andere Möglichkeit. Der von mir benutzte Fahrrad-Computer hat eine Anzeige für die Tagesfahrstrecke. Diese errechnet das Gerät aus dem Reifenumfang multipliziert mit der Anzahl der Impulse. Der Reifenumfang wird manuell in das Gerät eingegeben. In einer ersten Versuchsreihe wurde der Kreisumfang des Windmessers eingegeben. An einem windstillen Tag fuhr ich, den Windmesser an einem Stab in der Hand, auf einem ebenen und geraden Fahrradweg neben einer Bundesstraße 500 Meter ab. Die Distanz lässt sich leicht ermitteln, denn an einer Bundesstraße stehen ziemlich genau alle 50 Meter Baken am Straßenrand. Den Versuch wiederholte ich mehrmals in beiden

Richtungen. Ich erhielt auf dem Fahrrad-Computer einen Durchschnittswert von circa 125 Meter für die abgefahrenen 500 Meter. Dieser angezeigte Wert machte also ein Viertel der tatsächlich zurückgelegten Strecke aus. Ich vervierfachte die Eingabe des Reifenumfangs am Fahrrad-Computer. Und siehe da, in der nächsten Versuchsreihe lag die Abweichung zwischen gemessener und gefahrener Distanz nur noch bei circa 10 Prozent. Nach der nächsten Einstellung reduzierte sich die Abweichung auf einen Betrag, der mit meiner Methode nicht weiter zu präzisieren ist.

Das vorgestellte Anemometer ist preiswert, lässt sich einfach in kurzer Zeit herstellen, funktioniert zuverlässig und liefert Daten mit ausreichender Genauigkeit. Für den Dauereinsatz im Freien ist das Gerät allerdings nicht geeignet. Hierzu reicht der

Wetterschutz nicht aus. Ich setze das Gerät temporär dann ein, wenn ich mit meinem Segelflugmodell unterwegs bin. Das Gerät liefert mir für diesen Zweck zuverlässige Daten über Windrichtung und Windstärke. Allerdings zeigt das Gerät die Geschwindigkeit ausschließlich in Kilometern pro Stunde an, für die Modellfliegerei sind Angaben in Metern pro Sekunde üblich. Eine kleine Umrechnungstabelle kompensiert diese Einschränkung wieder. Besonders schätze ich die Mobilität, die mir dieses Gerät für die Messungen bietet. —hch

↓ Links und Foren
make-magazin.de/xqyw

Anzeige

Neues Leben für alte Tasten

Wenn die Tasten klemmen, macht das Schreiben keinen Spaß mehr – das Keyboard gehört aber noch lange nicht in den Müll. Aus den Einzelteilen wird nerdiger Schmuck, sie verschönern Teelichter oder machen sich als Magnet nützlich.

von Helga Hansen

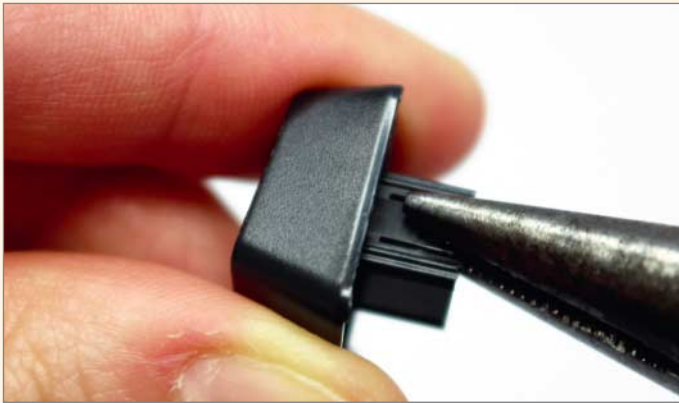


Schnellhack: Magnete



1 Tasten lösen

Als Erstes lösen wir die Tasten von der Tastatur. Dazu reicht ein Schraubendreher, mit dem wir die Tasten abhebeln. Anschließend sollte man die Tasten in warmem Seifenwasser reinigen.



2 Halterung entfernen

Die Plastikhalterung der Tasten lässt sich mit einer Spitzzange vorsichtig abbrechen oder rausdrehen.



3 Kanten glätten

Mit einem Seitenschneider entfernen wir alle Plastikreste, die über die Kante der Taste abstehen. Hilfreich, aber nicht notwendig ist es, wenn die inneren Kanten gleich hoch sind.



4 Magnetisieren

Mit einem zugeschnittenen Stück selbstklebender Magnetfolie ist der erste Magnet schon fertig. Alternativ können auch kleine Magnete in Sugru oder Knete in die Taste gedrückt werden – vor dem Benutzen sollten die allerdings trocknen und fest werden.



Kurzinfo

Zeitaufwand:
ab 5 Minuten

Kosten:
ab 2 Euro

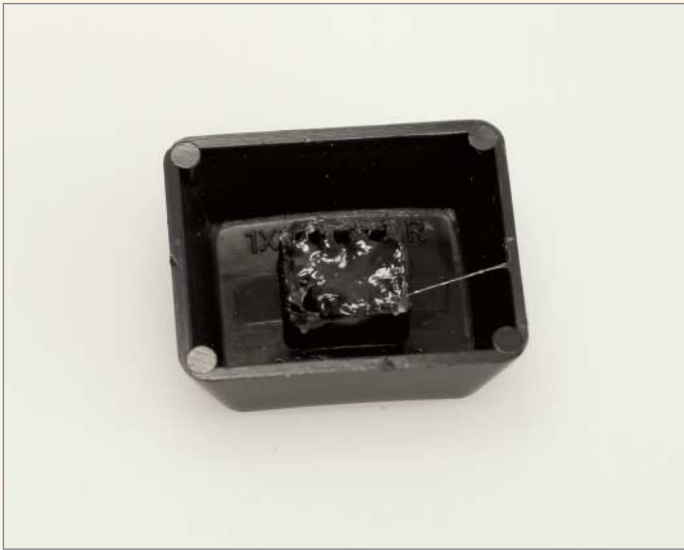
Schwierigkeitsgrad

leicht schwer

Einkaufsliste

- » alte Tastatur
- » selbstklebende Magnetfolie
- » Knete oder Sugru
- » Magnet

Schmuck selber machen



1 Tasten füllen

Um die Tasten auf den Klebeflächen der Schmuckteile zu befestigen, müssen wir das Loch in der Mitte füllen. Dazu haben wir Heißkleber genommen, der dann aushärtet. Der Kleber darf nicht über die Kanten der Tasten hinaus ragen und sollte möglichst eben abschließen. Wer Sugru benutzt, kann die Bauteile direkt aufdrücken, mit Heißkleber funktioniert das meistens nicht.



Kurzinfo

Zeitaufwand:
ab 5 Minuten

Kosten:
ab 1 Euro

Schwierigkeitsgrad

leicht schwer

Einkaufsliste

- » alte Tastatur
- » Schmuckbauteile zum Bekleben
- » Heißkleber und Sekundenkleber
- » Sugru



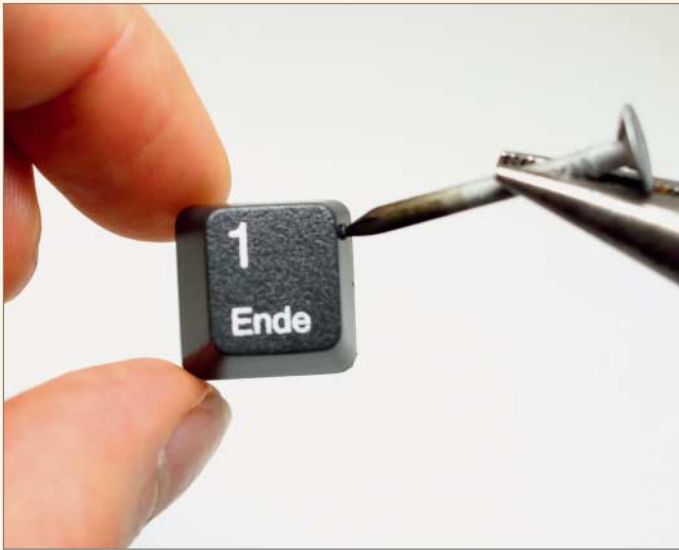
2 Kleben und festhalten

Die Schmuckteile mit einer glatten Klebefläche befestigen wir mit einem Klecks Sekundenkleber. Kurz andrücken, aber bitte nicht den Finger mit festkleben.



3 Auf die Form kommt es an

Einige Schmuckteile, wie etwa Haarspangen und Anstecker, haben löchrige Klebeflächen. Hier lohnt sich Heißkleber oder Sugru, der etwas großzügiger aufgetragen wird und dann durch die Poren quillt.



4 Löcher bohren

Mit Löchern lassen sich die Tasten auch auffädeln. Dazu haben wir mit der Spitzzange einen Nagel eingeklemmt und über einem Feuerzeug erhitzt. Mit dem heißen Nagel bohren wir links und rechts zwei Löcher. Diese sollten möglichst oben sitzen. Zum einen sitzen die Befestigungsreste in der Mitte, zum anderen dreht sich die Taste sonst entweder um sich selbst oder hängt verkehrt herum.

ACHTUNG

Der Nagel wird sehr heiß und darf erst wieder nach dem Abkühlen angefasst werden!
Alternativ können Sie auch einen kleinen Dremel nutzen.



5 Kette schließen

Die Kette muss mindestens an einer Seite offen sein, um die Taste aufzufädeln. Dann erst kann der Verschluss befestigt werden. In der einfachen Variante bilden wir am Ende eine Öse und drücken mit der Spitzzange einen kleinen Ring platt.



6 Weiter basteln

Die gezeigten Stücke sind nur ein Ausschnitt der Möglichkeiten. Auch andere Accessoires, zum Beispiel Manschettenknöpfe, lassen sich mit Tasten individualisieren. Mit Löchern können sie auf Stoffen aufgenäht werden.

Tasten als Deko



1 Gegenstände bekleben

Wer mit filigranem Schmuck nicht viel anfangen kann, klebt die Tasten auf alles andere im Büro oder Wohnzimmer. Dabei müssen Sie den Kleber auf die Oberfläche abstimmen. Auf dem Bilderrahmen halten die Tasten mit Sekundenkleber – auf Pappoberflächen verursacht der aber unschöne Flecken.

2 Jetzt in großflächig

Wer ganze Flächen beklebt, muss Geduld und Aufmerksamkeit mitbringen; besonders, wenn nur die Kanten geklebt werden. Alternativ können Sie die Tasten auch ganz füllen und so die Klebefläche vergrößern.

3 Für Faule

Hier haben wir die Magnetfolie auf Bildergröße geschnitten und in den Rahmen eingesetzt. Und dann die Magnete von der ersten Seite draufgesetzt.

Kurzinfo

Zeitaufwand:
ab 1 Minute

Kosten:
ab 2 Euro

Schwierigkeitsgrad

leicht schwer

Einkaufsliste

- » alte Tastatur
- » Magnetfolie
- » Heißkleber oder Sugru
- » Schachtel
- » Bilderrahmen
- » Pappkiste
- » Sekundenkleber

TIPP

Noch zu langweilig?
Mehr Abwechslung gibt es mit buntem Tesafilm oder speziellen Tastaturstickern. Sie haften umso besser, je flacher die Tasten sind.

Teelichter verhüllen



1 Es geht ans Eingemachte

Zum Schluss schrauben wir die Tastatur auf, um an das Innenleben zu kommen. Sind die obere und untere Hälfte ineinander eingerastet, hilft wieder der Schraubendreher. Innen befinden sich dann mehrere Plastikfolien. Wir nutzen die beiden Schichten, auf denen sich die Leiterbahnen und Kontakte befinden.

Kurzinfo

Zeitaufwand:
ab 10 Minuten

Kosten:
ab 1 Euro

Schwierigkeitsgrad

leicht schwer

Einkaufsliste

- » alte Tastatur
- » Glas
- » Tesafilm



2 Teelicht einkleiden

Mit der Schere schneiden wir die Folien auf die gewünschte Größe zu und kleben sie mit etwas Tesafilm um das Glas. Dieses sollte möglichst ein gerader Kreiszylinder sein. Hier haben wir das Glas von einem Brotaufstrich verwendet. Wer nur Tesafilm hat, kann die Folien auch direkt als Windschutz verwenden.



3 Dekorieren

Den Druck auf die Taste überträgt eine Schicht mit Gummipads auf die Kontaktfolien. Ausgeschnitten lassen sich die runden Pads als Deko für die Teelichter verwenden. Dabei lassen wir einen kleinen Rand für den Kleber und kleben auf die ausgerollte, glatte Folie. Dabei sparsam mit dem Kleber umgehen, da er schnell weiße Spuren hinterlässt. —hch



Links und Foren
make-magazin.de/x1pn

Batterieüberwachung mit MSP430

Elektrofahrzeuge erfreuen sich wachsender Beliebtheit – sie produzieren zumindest keine direkten Abgase, sind leise und im Betrieb günstig: Eine Kilowattstunde Strom kostet deutlich weniger als ein Liter Benzin, bei etwa vergleichbarer Reichweite. Nur mit der Tankanzeige solcher Fahrzeuge hapert es in der Praxis sehr oft. Gegen die Frage, wie lange man noch fahren kann, bis man das Fahrzeug stehen lassen muss, hilft der hier beschriebene Batteriemonitor.

von Karl-Heinz Dahlmann



Kurzinfo



Zeitaufwand:
ein Tag



Kosten:
etwa 25 Euro



Löten:
einfache Lötarbeiten



Kunststoffbearbeitung:
ggf. Gehäuse anpassen



Programmieren:
C (optional)

Schwierigkeitsgrad



Häufig haben Elektrofahrzeuge und andere batteriebetriebene Großverbraucher keine oder nur eine sehr einfache Ladezustandsanzeige mit wenig Aussagekraft. Was bei Handys oder Smartphones selbst bei den billigsten Geräten den Standard darstellt, ist bei Herstellern batteriebetriebener Fahrzeuge offenbar noch nicht in den Köpfen vorgedrungen oder wird aus anderen Gründen stiefmütterlich behandelt. Die meisten Bordcomputer, sofern überhaupt vorhanden, überwachen nur die Batteriespannung, aber die allein hat wenig Aussagekraft für den tatsächlichen Ladungszustand.

Unangenehm, wenn einen das Gefühl beschleicht, der Akku könnte eventuell nicht mehr für die geplante Strecke inklusive Rückfahrt reichen; ganz davon zu schweigen, wenn man wirklich mal in der Pampa stehen bleibt. Aber auch, wenn es nur darum geht, eine Tiefentladung der Akkus zu vermeiden, ist der hier vorgestellte Batteriemonitor sehr nützlich. Er überwacht den Ladezustand von klassischen Blei-Säure-Akkus, wie sie im PKW vorkommen. Man findet sie aber auch in größeren Elektrofahrzeugen wie Golfcarts, Krankenhaufahrstühlen oder elektrischen Rollstühlen, bei Elektrobooten, in Wohnmobilen, zur Energieversorgung der batteriebetrie-

nen Soundanlage für die Gartenparty oder als Energiespeicher für Solaranlagen, etwa auf dem Gartenhäuschen am Stadtrand.

Entwickelt wurde der Batteriemonitor für ein Golfcart, das auf dem Münchner Messegelände häufig benutzt wird, um zwischen Messehallen und Lager zu pendeln. Dabei fällt es schwer abzuschätzen, wie weit man mit der Akkuladung noch kommt: Zum einen ist die Entfernung zwischen den einzelnen Hallen recht unterschiedlich, zum anderen ist es schwierig, die gefahrenen Kilometer im Kopf zu addieren. Erschwerend kommt hinzu, dass das Fahrzeug von verschiedenen Personen benutzt wird. Die werksseitig eingebaute Ladekontrolllampe ist nicht besonders aussagekräftig und leuchtet irgendwann mal auf, geht aber dann auch wieder aus. Das ist zwar ein Hinweis, dass man das Golfcart „bald“ wieder aufladen soll, aber wann ist es schon „wirklich bald“?

Prinzipiell kann die Schaltung zwar auch mit anderen Akkutypen eingesetzt werden, die hier vorgestellte Messvariante mit einem Shunt (einem niederohmigen elektrischen Widerstand, der zur Messung des elektrischen Stromes verwendet wird), die Programmierung der Entladekennlinien und nicht zuletzt die eigene Stromaufnahme der Messelektronik ist aber eher auf

größere Blei-Akkus mit mehreren Amperestunden zugeschnitten. Auch wenn der durchschnittliche Eigenverbrauch mit etwa 4 Milliampere nach nicht viel klingt, kommt am Tag eben doch 0,1 Ah zusammen oder in 10 Tagen etwa 1 Ah. Bei klassischen Autobatterien mit 36–44 Ah oder mehr ist das jedoch verschmerzbar. Die nachfolgende Betrachtungen gelten für alle klassischen Blei-Säure-Akkus, egal ob Nassakku, AGM- oder GEL-Variante, und sollen die grundsätzliche Problematik beim Überwachen erläutern.

Mysterium Batterie

Der Batteriemonitor ist nicht nur als informativer Bordcomputer interessant, sondern kann insbesondere auch die Lebensdauer der eingesetzten Akkus erhöhen, indem er vor Tiefentladung und deren Folgen wie deutlichem Kapazitätsverlust warnt. Durch die recht genaue Einschätzung der zur Verfügung stehenden Energie kann die Batterie optimal ausgenutzt werden, ohne dass man sich Sorgen um eventuelle Schäden machen muss. Eine Tiefentladung kann nicht wirklich anhand einer Spannung festgemacht werden, weder im Leerlauf noch unter einer meist unbekannteren Last – zu vielschichtig sind die Fak-



toren, die die Klemmenspannung beeinflussen, etwa die Erholungszeit nach einer Belastung oder Ladung oder die interne Säureschichtung.

Typische Faustformeln im Internet wie „nie unter 11,5 Volt bei einer 12-Volt-Autobatterie“ (manchmal ist auch von 10,8 Volt die Rede) kratzen nur an der Oberfläche und gelten nur unter bestimmten Bedingungen wie Leerlauf oder mittlere Belastung. Ansonsten würde die Batterie allein schon durch die normalen Startvorgänge bei Betätigung des Anlassers Schaden nehmen. Die Tabelle rechts gibt die Kenndaten eines als Beispiel ausgewählten Blei-Säure-Akkus höherer Qualität wieder.

Die Kapazität wird bei Autobatterien als K20-Wert bei einer Entladezeit von 20 Stunden angegeben, im vorliegenden Fall also mit 2,75 Ampere. Wird sie stärker belastet, sinkt auch die entnehmbare Kapazität – und zwar mitunter deutlich – durch den sogenannten Peukert-Effekt; dazu gleich mehr. Manche Hersteller geben die Kapazität, je nach Anwendungszweck, auch als K10- oder K5-Wert an oder auch die Betriebszeit in Minuten bei einer bestimmten Stromentnahme.

Der sogenannte Kaltstartstrom (englisch CCA) wird meist nach Europäischen Normen (EN) spezifiziert und sagt aus, wie viel Strom die Batterie für 30 Sekunden bei einer Temperatur von minus 18 Grad (0 Fahren-

heit) noch liefern kann, ohne eine Spannung von 1,2 Volt pro Zelle beziehungsweise 7,2 Volt bei einer 12-V-Batterie zu unterschreiten. Zwar ist die Batterie im Fahrzeug so dimensioniert, dass in der Regel deutlich weniger Strom für den Anlasser benötigt wird, aber mitunter braucht man auch mehrere Startvorgänge, bis der Motor anspringt, je nach Zustand des Fahrzeugs und Außentemperatur. Allein hieran zeigt sich schon, dass eine Regel wie „nie unter 11,5 Volt entladen“ im Grunde wenig relevant sein kann.

Der Innenwiderstand wird eher selten angegeben, sagt aber viel über die Güte der Batterie aus. Er ergibt sich aus dem internen Aufbau des Akkus, der Leitfähigkeit der Batteriesäure und der verfügbaren Bleioberfläche. Die Leitfähigkeit der verwendeten Schwefelsäure ist stark abhängig vom Ladezustand und bei 38 Prozent Anteil am höchsten. Dieser Zustand entspricht einem vollen Akku. Im Laufe des Entladens sinkt der Säureanteil auf etwa 10 Prozent ab; spätestens dann sollte sie wieder geladen werden. Man sollte in diesem Fall mit dem Aufladen nicht zu lange warten, da ein vollständig entladener Akku anfällig für Sulfatierung ist: die Bildung größerer Sulfatkristalle, die die Oberfläche der Elektroden verringern und teilweise zu Boden sinken. Dieser Zustand ist nicht reversibel und mindert dauerhaft die Kapazität, weil auch die aktive Masse im Akku verringert wird. Vollständig-

Optima YellowTop S 4.2

technische Daten

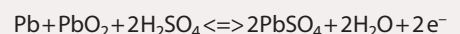
Nennspannung	12 V
Kapazität (K20/CA)	55 Ah
Kaltstartstrom (CCA)	765 A
Innenwiderstand	2,8 mΩ
Leistung	660 Wh

ges Entladen ist aber generell kein Problem, solange man den Blei-Akku in diesem Zustand nicht tage- oder gar wochenlang in der Garage abstellt.

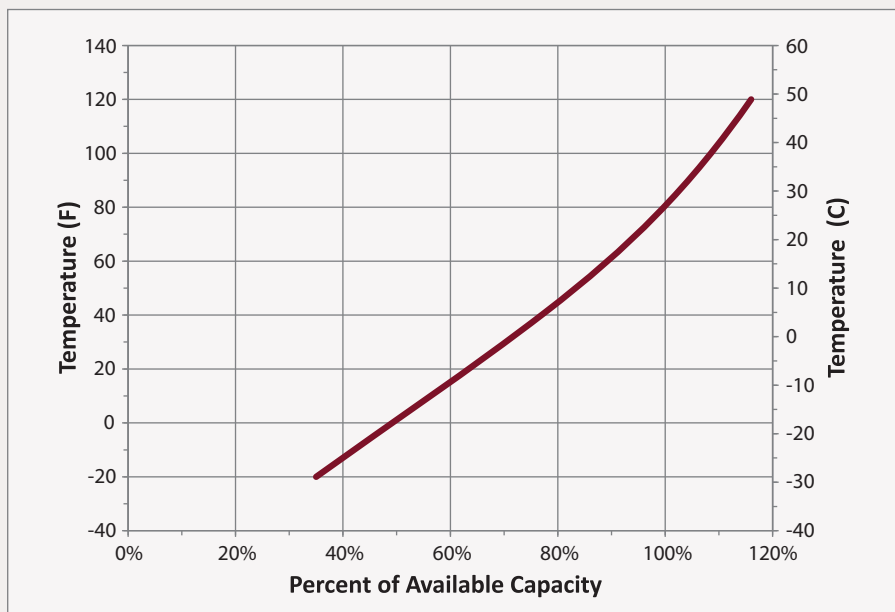
Kälteschock

Die Kapazitätsangabe K20 seitens des Herstellers gilt definitionsgemäß bei 27 Grad Celsius (80 Fahrenheit) und sinkt darunter stark ab, wie das Schaubild unten zeigt. Das ist der Hauptgrund, warum Autobatterien grundsätzlich im Winter ausfallen und deren Altersschwächen schonungslos offenbart werden. Die Kapazität erhöht sich schlagartig wieder, wenn die Batterie erwärmt wird. Wer also im Winter Probleme mit der Batterie hat und kein Ladegerät verfügbar ist, kann sich wirksam behelfen, indem er die Batterie abends ausbaut, in der Wohnung abstellt und morgens mit Raumtemperatur wieder einbaut. Das ist natürlich keine Dauerlösung.

Die folgende Redoxformel beschreibt die chemische Reaktion beim Entladen eines Blei-Akkus:



Dabei wird Blei, Bleidioxid und Schwefelsäure zu Bleisulfat und Wasser verwandelt und pro Säuremolekül ein Elektron abgegeben. Die Konzentration der Säure nimmt von 38 Prozent im vollgeladenen Zustand ab bis etwa 10 Prozent bei vollständiger Entladung. Man kann die Kapazität des Akkus daher auch mit einem Säureheber messen. Die Dichte der Säure schwankt zwi-



Die Leistung der Batterie fällt bei sinkender Temperatur rapide ab – bei minus 18 Grad (0 Fahrenheit) bringt die Batterie nur noch etwa 50 Prozent Leistung.

Peukert-Wert

Peukert-Wert	0,05 CA	0,4 CA	1,2 CA	2,4 CA
1,1	100 %	81 %	73 %	68 %
1,2	100 %	66 %	54 %	47 %
1,3	100 %	54 %	39 %	32 %

schen etwa 1,28 (voll) und 1,05 (leer). Beim Laden läuft die Reaktion genau umgekehrt: Man muss Elektronen zuführen und der Säureanteil und die Dichte steigen wieder.

Die sich ergebende Spannung lässt sich nach der Nernstschen Formel berechnen und schwankt zwischen 2,16 Volt und 2,04 Volt pro Zelle beziehungsweise zwischen 12,96 Volt und 12,24 Volt für eine komplette Autobatterie. Das ist weit entfernt von dem Daumenwert von 11,5 Volt. Allerdings kann die Spannung aufgrund von Säureschichtung direkt nach dem Entladen zu niedrig beziehungsweise direkt nach dem Laden zu hoch angezeigt werden.

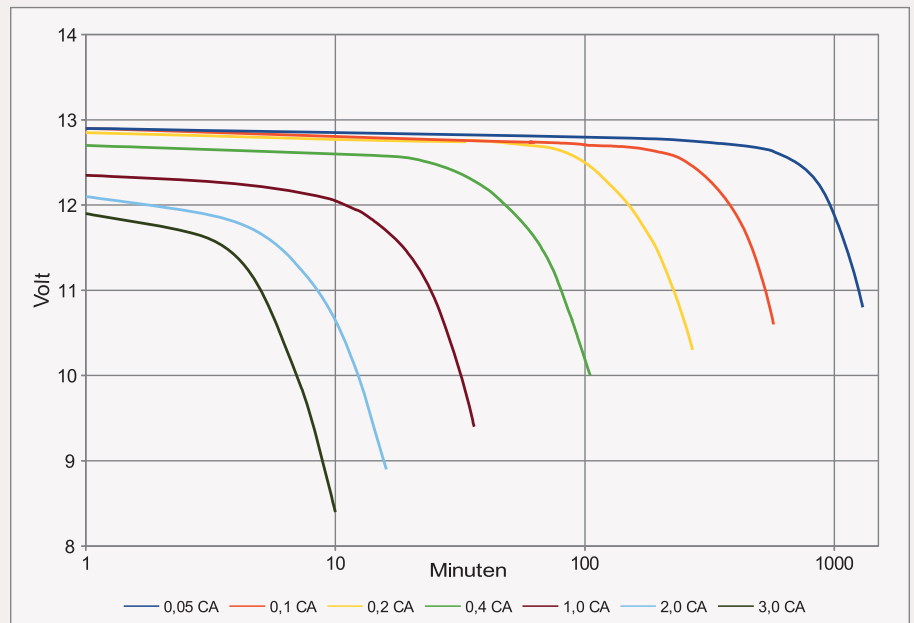
Während bei Fahrzeugen dieser Effekt durch Erschütterung schneller verschwindet, kann er bei stationären Batterien das Messergebnis stark verfälschen – unter Umständen muss man vor einer sinnvollen Messung mehrere Stunden warten. Am Absinken der Spannung an den Batterieklemmen bei starker Belastung ist hauptsächlich der Innenwiderstand schuld, der mit sinkendem Ladestand größer wird. Auch wenn der Innenwiderstand eines Blei-Akkus sehr niedrig ist, macht er sich bei großen Strömen durchaus bemerkbar.

Peukert-Effekt

Den Effekt der sinkenden Restkapazität von Akkus bei hoher Stromentnahme hatte Wilhelm Peukert schon 1897 festgestellt. Die entnehmbare Kapazität ist nämlich nicht immer gleich, sondern hängt von der Dauer der Entladung einer Batterie ab. Bei hohen Strömen sinkt die verfügbare Kapazität, weil der Ionen-Austausch in einem Blei-Akku nicht beliebig schnell vonstatten geht, sondern wie jede chemische Reaktion in ihrer Geschwindigkeit begrenzt ist, die zudem von der Temperatur abhängt.

Die Kapazität eines Akkus wird daher bei einer definierten Stromentnahme über eine bestimmte Zeitspanne definiert, zum Beispiel als K20-Wert für die Entladung in 20 Stunden. Je nach Hersteller und Verwendungszweck werden auch Werte für kürzere Zeiträume angegeben, üblich sind etwa 10 oder 5 Stunden. Den Peukert-Wert kann in diesen Fällen anhand zweier definierter Kapazitätsangaben bei unterschiedlicher Last berechnet werden. Er sagt direkt etwas über die Qualität des Akkus aus.

Der ideale Wert ohne Kapazitätsverlust wäre 1,0, ist aber bei Blei-Akkus nicht ansatzweise zu realisieren. Übliche Werte



Bei großen Strömen ergeben sich deutliche Abweichungen in der entnehmbaren Kapazität.

liegen zwischen 1,1 und 1,3. Wenn der Hersteller keine besonderen Angaben macht, kann man hier im Mittel mit 1,2 rechnen. Wer einen hochwertigen oder nagelneuen Akku benutzt, kann auch 1,1 ansetzen, wer noch eine alte Gurke nutzen oder seinen Akku besonders schonen will, rechnet mit 1,3. Wie die Tabelle zeigt, muss man dann aber deutliche Abschläge bei der zur Verfügung stehenden Leistung hinnehmen.

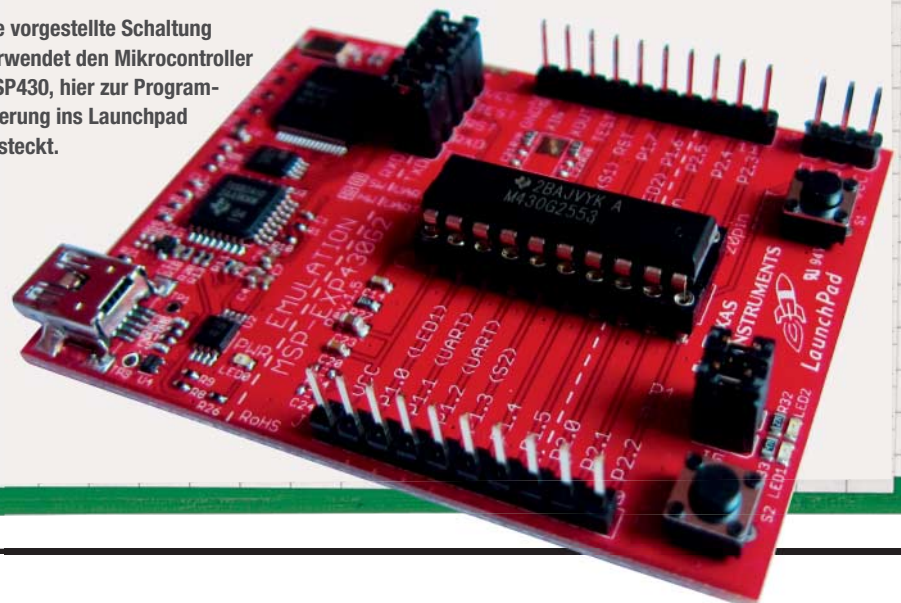
Die Formeln für die Berechnung der entnehmbaren Kapazität bei einer definierten Stromstärke I beziehungsweise für den Peukert-Wert über zwei Kapazitätsangaben lauten allgemein:

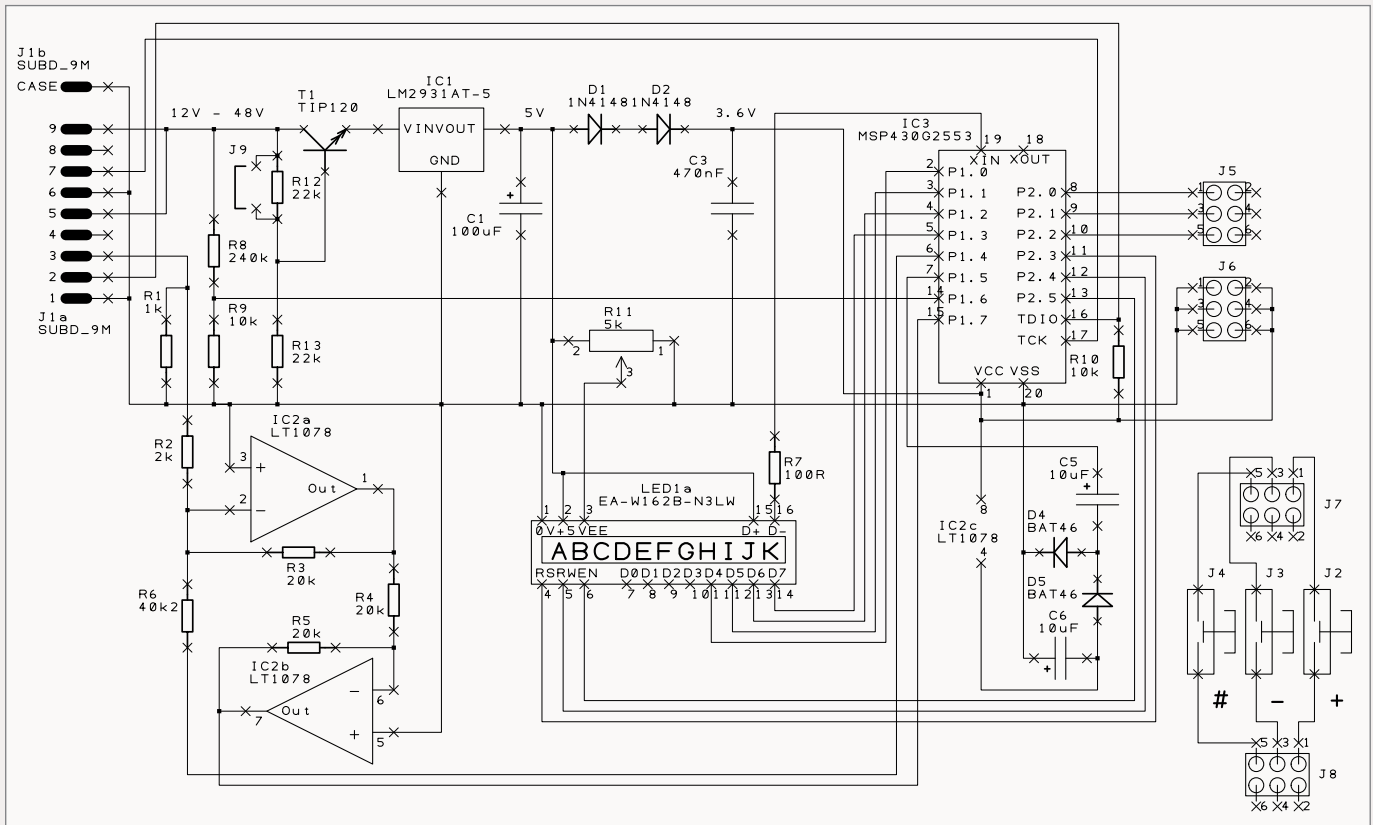
$$C_{\text{eff}} = C_{20} \times (C_{20} / I_{\text{Last}} \times 20)^{k-1}$$

$$k = (\log(C_{\text{eff}} \times I_{\text{Last}} / 20)) / (\log(C_{20} \times I_{\text{Last}} / 20))$$

Dabei ist C_{20} die Kapazität über den Zeitraum von 20 Stunden, I_{Last} der benötigte Strom und k der Peukert-Wert. Man berechnet also in der ersten Formel den Klammerausdruck hoch 0,1 bis 0,3 (statt 1,1 bis 1,3) und erhält dann einen Faktor, aus dem sich die entnehmbare Kapazität beim benötigten Strom ergibt. Das ergibt bei großen Strömen deutliche Abweichungen, wie das logarithmische Diagramm oben verdeutlicht. Bei Angabe von anderen Werten wie K10 oder K5 kann man einfach C_{20}

Die vorgestellte Schaltung verwendet den Mikrocontroller MSP430, hier zur Programmierung ins Launchpad gesteckt.





Die Schaltung benötigt nur wenig aktive Komponenten, da der verwendete Mikrocontroller MSP430 fast alle nötigen Anschlüsse mitbringt, inklusive A/D-Wandler, Eingängen sowie einer internen Referenzspannungsquelle.

durch C_{10} oder C_5 ersetzen und den Multiplikator von I_{Last} entsprechend auf 10 oder 5 anpassen.

Da die Kennlinien für Akkus identischer Bauart, aber mit unterschiedlicher Kapazität letztlich gleich sind, hat sich die Nutzung eines CA-Wertes durchgesetzt, der mit der Kapazität multipliziert den Strom ergibt. 1 CA entspricht dem Strom bei Entladung in einer Stunde ohne Kapazitätsverlust, also entspricht bei einer 44-Ah-Batterie 1 CA = 44 A.

An den Kennlinien kann man auch gut die Entladeschlussspannung ablesen, die für den jeweils entnommenen Stromwert gilt und den Spannungsabfall am steigenden Innenwiderstand berücksichtigt. So ergibt sich eine Entladeschlussspannung von 10,8 Volt für eine Belastung von 0,05 CA oder auch 8,5 Volt bei 3 CA. Aus diesem Grund ist eine Spannungsüberwachung des Akkus für die Ermittlung der Restkapazität denkbar ungeeignet.

Winterbetrieb

Zwar sind die temperaturabhängigen Effekte eines Blei-Akkus nicht zu vernachlässigen, auf den Anschluss eines Temperatursensors wurde beim hier vorgestellten Batteriemonitor aber der Einfachheit halber verzichtet. Eigentlich sollte man auf die Nutzung von Elektrofahrzeugen an sehr kalten Tagen eher verzichten oder sie zumindest nicht in der Kälte stehen lassen. Sofern man aus der warmen Garage fährt, dürfte die Temperatur des Blei-Akkus bei stärkerer Belastung kaum absinken, da der Innenwiderstand hier wie eine eingebaute Heizung wirkt – zumindest solange man fährt.

Problematisch ist allerdings, Fahrzeuge bei deutlichen Minusgraden längere Zeit im Freien abzustellen. Man braucht nämlich bei kalten Akkus eine höhere Ladespannung, der Ladevorgang dauert also in jedem Fall deutlich länger als bei gut temperierten Akkus. Nebenbei steigt der Ge-

frierpunkt mit der Entladung an und liegt bei entladenerm Akku bei nur wenigen Grad minus (10 Prozent Schwefelsäure oder auch 1 mol/l). Es ist daher sinnvoll, im Winter den Akku sofort nach der Rückkehr von einer Fahrt zu laden, solange er sozusagen noch auf Betriebstemperatur ist.

Überwachungsprogramm

Damit liegen die Strategien für die Überwachung der Batterie durch den Batteriemonitor fest. Basis bildet die Grundkapazität (K20) der Batterie. Der entnommene Strom wird über einen Shunt-Widerstand gemessen, in Amperesekunden umgerechnet und mit dem Peukert-Wert gewichtet. Bei großen Stromstärken wird so die entnommene Energiemenge rechnerisch erhöht. Die auf diese Weise ermittelte Ladung wird aufsummiert, die verbleibende Kapazität berechnet und als Prozentwert angezeigt. Außerdem wird die Restlaufzeit auf Basis der



Je nach Geschmack kann man Displays mit weißer Schrift und blauer Hintergrundbeleuchtung nutzen; alternativ schwarze Beschriftung mit gelbgrünem Hintergrund, die man bei Tageslicht auch ohne Beleuchtung ablesen kann.

durchschnittlichen Belastung der letzten Minuten ermittelt.

Beim Laden misst der Batteriemonitor die zugeführte Ladung in Amperesekunden (ebenfalls mit dem Peukert-Wert gewichtet) und kann so den Ladezustand bei Zwischenladungen korrigieren. Zusätzlich gibt es eine Spannungsüberwachung, die in Abhängigkeit von der Realbelastung (CA) die zulässige Ladeschlussspannung errechnet und bei Unterschreiten Alarm gibt. Diese wird dynamisch aus der hinterlegten Peukert-Tabelle im Flash des Mikrocontrollers ermittelt.

Letztlich ist die Schaltung aber nur ein Messgerät und greift nicht aktiv in die Steuerung ein. Sie kann insbesondere keine Tiefenentladung verhindern, aber zumindest durch entsprechende Hinweise aktiv davor warnen: Ähnlich wie eine Tankanzeige blinkt der Batteriemonitor, wenn die Energie knapp wird, um so auf die Situation aufmerksam zu machen.

Spannungsversorgung

Die zu überwachende Batterie dient gleichzeitig als Spannungsversorgung für den Batteriemonitor und liefert damit bereits die erste direkte Messgröße: die aktuelle Spannung. Der Strom wird indirekt über einen Shunt gemessen, einen sehr kleinen Widerstand, an dem eine proportionale Spannung abfällt, aus der sich der fließenden Strom berechnen lässt.

Das Modell auf dem Foto ist für 75 Ampere Dauerbelastung ausgelegt und hat dabei einen Spannungsabfall von 75 Millivolt. Solche Shunts findet man je nach Ge-

naugigkeit bereits für wenige Euro im Elektronikfachhandel oder auch bei eBay. Shunts gibt es für verschiedene Nennstrombereiche. Zu groß dimensionierte Exemplare verursachen Messfehler, zu klein dimensionierte erzeugen zu viel Spannungsabfall. Man sollte den Shunt deshalb individuell für den jeweiligen Anwendungszweck auswählen.

Es ist auf ausreichend große Querschnitte der Kabel zu achten, um die Ausgangsspannung durch zusätzliche Kabelwiderstände nicht unnötig herabzusetzen. Jedes Milliohm macht sich bei größeren Strömen störend bemerkbar. Beim Einbau sollte man sorgfältig vorgehen und den Shunt auf jeden Fall gut isolieren – insbesondere bei Fahrzeugen! Im Idealfall überzieht man ihn mit einem entsprechend großen Schrumpfschlauch, ansonsten kann man ihn mit Klebeband isolieren.

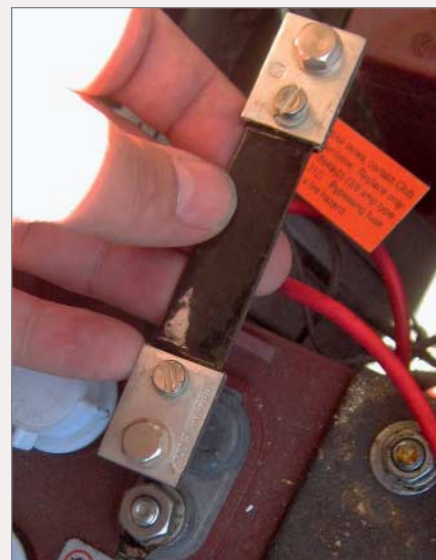
Versehentliche Kurzschlüsse während einer Fahrt oder beim Einbau können nicht nur irreparablen Schaden am Akku anrichten und Blitze oder Funkenregen erzeugen, sondern auch gravierendere Folgen haben – etwa Risse, durch die Säure austritt, oder sogar eine Explosion bei länger andauerndem Kurzschluss. Daher sollte man beim Einbau größte Vorsicht walten lassen, den Pluspol vorübergehend abklemmen oder zumindest mit einer Folie oder Plastiktüte abkleben oder abdecken. Große Blei-Akkus können bei einem Kurzschluss mehrere hundert Ampere liefern und eignen sich dadurch auch zum Schweißen, selbst wenn man das gar nicht möchte. Aus Sicherheitsgründen und um hohe Spannungspotenziale für die Mess-

schaltung zu vermeiden, wird der Shunt in die Masseleitung (Minuspol) geschaltet.

Die Spannungsversorgung erfolgt durch einen Spannungsregler vom Typ LM2931AT-5, der auch hohe Eingangsspannungen bis zu 30 Volt verträgt und trotzdem als Low-Drop-Version recht sparsam ist. Der Regler erzeugt eine stabilisierte 5-V-Spannung für das Display und dessen Beleuchtung, sofern vorhanden. Bei größeren Spannungen wie 36 oder 48 Volt für Golfcarts hilft T1 und halbiert in etwa die Eingangsspannung. Die Schaltung kann in dieser Form bis maximal 60 Volt benutzt werden.

Mikrocontroller

Der verwendete Mikrocontroller vom Typ MSP430G2553 verträgt maximal 3,6 V, die hier durch zwei einfache Siliziumdioden D1 und D2 aus 5 V erzeugt werden. Der Controller ist bereits im Lieferumfang des MSP430-Launchpads für die Programmierung enthalten (Einführung siehe c't Hacks 1/2014, S. 80). Die meisten Anschlüsse sind als digitale Ausgänge konfiguriert, lediglich die drei Taster zur Steuerung der Funktionen und Eingaben sind digitale Eingänge mit Controller-internen Pull-up-Widerständen. Port 1.6 und 1.7 sind die A/D-Wandler-Eingänge für die Messung von Spannung und Strom. Port 1.4 stellt die interne Referenzspannung des Con-



Durch die großen Schraubklemmen am Shunt fließt der Strom, an den kleineren wird die Messspannung abgegriffen.

trollers von 2,5 V zur Verfügung, die vom Messverstärker genutzt wird.

Port 2.6 steuert die Beleuchtung des Displays. Port 1.5 wird als Ladungspumpe betrieben und erzeugt aus den 3,6 V des Controllers eine negative Spannung von -3,0 V für die Operationsverstärker. Zur Anzeige der Messwerte und Bedienung des Batteriemonitors wird ein zweizeiliges Matrix-LCD mit 2×16 Zeichen genutzt. Alle Displays mit den bewährten HD44780- oder ST7036-Controllern sind verwendbar.

Wichtig ist das Poti R11 für die Kontrasteinstellung des Displays, sonst sieht man entweder nichts oder nur eine schwarze Fläche. R7 ist ein grober Richtwert und kann gegebenenfalls individuell angepasst werden, falls man ein beleuchtetes Display verwendet. Hier empfiehlt sich 75 Ohm bei den klassischen grügelben Displays oder 150 Ohm für Displays mit blauer Hintergrundbeleuchtung und weißer Schrift. Gegebenenfalls muss man das Datenblatt zu Rate ziehen oder messen. 10 mA sollten ausreichen, mehr als 20 mA sollte man für eine lange Lebensdauer des Displays vermeiden.

Messen am Shunt

Die Batteriespannung wird über den Spannungsteiler aus R8 und R9 direkt an den Mikrocontroller geführt und ist für bis zu 60 V Eingangsspannung ausgelegt, die auf $1/25$ reduziert wird (maximal 2,4 V) und

beim Controller eine Messung in 60-mV-Schritten (10 Bit) ermöglicht. Für die Strommessung als Spannungsabfall am Shunt benötigt man einen Operationsverstärker, um die nominelle Messspannung von 75 mV bei Volllast möglichst gleichmäßig über den Messbereich des Controllers von 2,5 V zu verteilen.

Der Shunt wird in die Masseleitung der Batterie gesetzt. Diese Variante hat den Vorteil, dass man nur mit kleinen Spannungspotenzialen von 75 Millivolt in Bezug auf GND oder Masse zu tun hat und nicht mit der gesamten Batteriespannung von bis zu 48 Volt. Der Nachteil dieser Variante ist, dass die Messspannung auch negativ zum Bezugspotenzial Masse werden kann. Die A/D-Wandler-Eingänge des Mikrocontrollers sind aber nur für positive Spannungen ausgelegt und vertragen laut Datenblatt nicht mehr als -0,3 Volt. Doch selbst wenn man diesen Bereich nicht verlässt, spuckt der A/D-Wandler für Spannungen unter 0 Volt keine sinnvollen Werte aus.

Dieses Manko behebt der Doppel-Operationsverstärker IC2. Der erste Verstärker ist dabei als klassischer Summierer geschaltet und kann die Eingänge mit wenig Beschaltung auch ungleich gewichten, also verschieden verstärken. Das wird hier genutzt: Er verstärkt hier die Shunt-Spannung von ± 75 mV mit Faktor 10 auf ± 750 mV und halbiert gleichzeitig die Referenzspannung des

Mikrocontrollers von 2,5 V. Die Summe beider Spannungen findet sich am Ausgang des ersten Operationsverstärkers, also $1,25 \text{ V} \pm 0,75 \text{ V}$. Damit schwankt die Messspannung für den Strom im Bereich von 0,5 V bis 2,0 V bei Volllast, was noch Reserven für Spitzen von ± 20 Prozent bietet.

Der Nachteil des klassischen Summierers ist jedoch, dass am Ausgang die Spannung negativ ist, also statt $+0,5 \text{ V}$ bis $+2,0 \text{ V}$ stellt sich hier $-0,5 \text{ V}$ bis $-2,0 \text{ V}$ ein. Der zweite Operationsverstärker behebt dieses Problem und ist als klassischer Inverter geschaltet (oder auch als Summierer mit Verstärkung 1) und korrigiert das Vorzeichen der Spannung, sodass der A/D-Wandler des Controllers problemlos mit ihr umgehen kann.

Praktischer Einsatz

Die Bedienung des Batteriemonitors erfolgt über drei Tasten. Die Grundfunktionen sind dabei PLUS und MINUS für die Änderung von Werten; die SELECT-Taste übernimmt den eingestellten Wert und springt automatisch auf das nächste Eingabefeld. Außerdem kann man im Betrieb mit PLUS und MINUS zwischen den Anzeigemodi wechseln, etwa zwischen Restlaufzeit und Entfernung (wahlweise in Kilometern, Meilen oder Seemeilen).

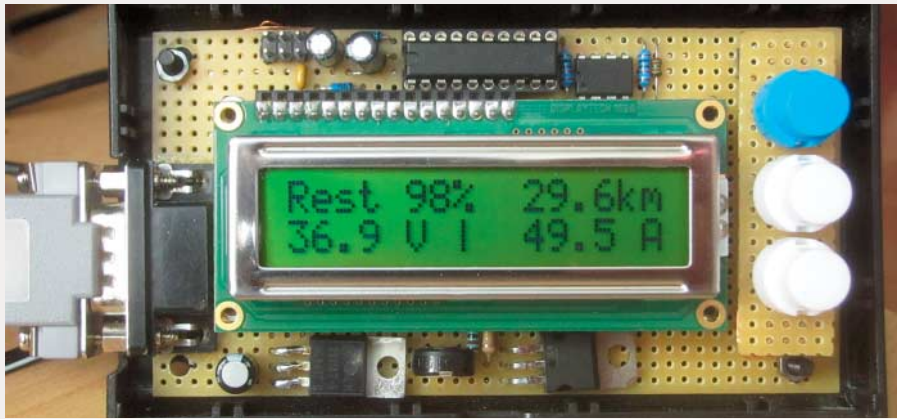
Bei dem ersten Anschluss des Batteriemonitors an das zu überwachende System lotst er automatisch durch die Konfiguration, in der der aktuelle Ladezustand (Voll, Leer oder Teilladung), Nominalspannung und Kapazität des Blei-Akkus, der Peukert-Wert von 1.1 bis 1.3) sowie die Shunt-Daten (Spannung in mV und zugehöriger Strom in A) abgefragt werden.

Mit einer Offset-Korrektur kann man im unbelasteten Zustand eine Nullwertkalibrierung vornehmen, falls der Batteriemonitor einen anderen Wert als 0 anzeigt. Sofern man ein Fahrzeug benutzt, lässt sich auch ein Umrechnungsfaktor der Kapazität in Entfernungen in der gewünschten Einheit auswählen.

Hier sind die Werte jedoch mit Vorsicht zu genießen, da die Stromaufnahme pro Kilometer durchaus unterschiedlich sein kann, je nachdem, ob man den Berg hoch oder herunter oder gegen oder mit der Strömung fährt. Das Konfigurationsmenü lässt sich jederzeit aufrufen, indem man die Tasten PLUS und MINUS für mindestens drei Sekunden gleichzeitig gedrückt hält. Dann kann man den aktuellen Ladezustand



Oben der geöffnete Batteriemonitor mit eingesetztem Mikrocontroller, unten das per USB angeschlossene Launchpad mit der leeren Fassung in der Mitte – für die Inboard-Programmierung sind die Signale des Controllers mit den zugehörigen Launchpad-Pins verbunden.



Für Regler und Transistor sollten Exemplare im TO-220-Gehäuse eingebaut werden, da im Fall der Display-Beleuchtung etwa 20 bis 25 mA fließen und die Bauteile warm werden.

anpassen oder weitere Konfigurationsdaten ändern. Mit einem Passwort lässt sich die Konfiguration auch schützen.

Programmierung

MSP430-Controller verfügen über eine empfindliche Brownout-Erkennung (= kurzfristige Schwankung der Spannungsversorgung) und lösen bei entsprechenden Impulsen daher schnell einen Reset aus. Um zu verhindern, dass dabei die Konfiguration und vor allem der aktuelle Ladezustand gelöscht werden, ist die Funktion `_system_pre_init` wichtig, die die Plausibilität der Daten im RAM prüft und eine vollständige Initialisierung bei kurzen Brownouts verhindert. Dadurch ist auch ein kurzzeitiges Abstecken des Batteriemonitors möglich, ohne dass die Konfiguration verloren geht.

Die Firmware des Controllers wurde in C programmiert und kann bei Bedarf angepasst werden (Download des Codes und Stückliste der Schaltung siehe Link am Ende des Artikels). Der Beispielcode belegt 8 kByte Flash und 256 Byte RAM, damit wird der verfügbare Speicher nur zur Hälfte ausgenutzt. Intern arbeitet der Batteriemonitor nur mit Integer-Arithmetik; 27,5 Ampere zum Beispiel werden intern durch die Ganzzahl 275 repräsentiert. Der Grund: Die Gleitkommabibliotheken von C rechnen nicht nur sehr aufwendig, sondern fressen auch viel Speicher.

Wann immer es möglich ist, nutzt der Code Schiebeoperationen für Multiplikation oder Division, weil das deutlich schneller abläuft. Sofern man aus Gründen der Genauigkeit zum Beispiel einen Zwischenwert mit 100 oder 1000 multipliziert und später

wieder dividieren will, bietet sich etwa eine Multiplikation oder Division mit 128 (2^7) oder 1024 (2^{10}) an, die effizient durch Schiebeoperationen erfolgen. Auch printf ist mit Vorsicht zu genießen: Bei diesem Controller wird standardmäßig nur eine Minimalversion gelinkt, die nur wenige Datentypen und Optionen für Formatierung kennt. Die volle printf-Unterstützung belegt nämlich zusätzlich 10 kByte Flash.

Ebenfalls vorsichtig muss man bei Rechnungen mit unterschiedlichen Datentypen sein, wie int oder long beziehungsweise signed oder unsigned. Im Zweifel sollte man die Datentypen explizit casten, um Überraschungen zu vermeiden. Die Peukert-Werte, für deren Berechnung Logarithmus und Exponentialfunktionen nötig sind, wurden vorab als Tabelle berechnet und auf den Initialwert von 2^{15} (32768) und 1/320 CA skaliert, um die Berechnungen zu vereinfachen.

Die A/D-Wandler messen mit einer Frequenz von 1 kHz, dabei werden jeweils 500 Werte gemittelt, also zwei echte Messwerte pro Sekunde verarbeitet. Die Aktualisierung der Anzeige erfolgt nur einmal pro Sekunde, um das Display (außer bei der Konfiguration) weitestgehend ruhig zu halten. Bei einem Ladezustand von 15 Prozent und darunter blinkt das Display für jeweils 10 Sekunden, wenn der Prozentwert um 1 sinkt. Der Batteriemonitor misst bis -20 Prozent, was man möglichst nicht ausreizen sollte. Wenn die Mindestspannung unterschritten wird (abhängig vom Entladestrom), blinkt das Display ebenfalls für 10 Sekunden.

Wichtig für die Genauigkeit sind möglichst präzise Widerstände für R2 bis R6 sowie R8 und R9 (1 Prozent Toleranz oder

weniger) sowie die technischen Daten des verwendeten Operationsverstärkers IC2. Hier sollte auf jeden Fall ein Low-Offset- und möglichst auch ein Low-Power-Typ eingesetzt werden, da der Stromverbrauch eines einzigen Standard-OPs durchaus so hoch sein kann wie jener der gesamten restlichen Schaltung. Insbesondere der Offset bringt hier aufgrund der geringen Messspannung von nominal 75 mV größere Fehler mit sich. Die Software erlaubt zwar auch eine Nullkalibrierung für eine automatische Korrektur des Offsetfehlers. Ein kleinerer Offset bei einem OP bedeutet aber auch weniger Temperatureffekte. Der vorgeschlagene Typ LT1078 hat eine vernachlässigbare Offset-Spannung von maximal 70 Mikrovolt gegenüber sonst 5 mV bei billigen Standard-Operationsverstärkern.

Für die Programmierung des MSP430-G2553 benötigt man das Code Composer Studio von Texas Instruments, das man bis 16 kByte Code gratis nutzen kann. Der Controller wird einfach auf das Launchpad gesetzt, das über das mitgelieferte USB-Kabel mit dem Computer verbunden wird und ein neues Projekt mit dem passenden Controller-Typ angelegt. Dann kopiert man das Sourcefile des Batteriemonitors in das Verzeichnis, kompiliert durch Klick auf das Hammersymbol und startet anschließend den Programmier- beziehungsweise Debuggingvorgang über das grüne Käfersymbol. Danach kann man den so programmierten Controller in den Batteriemonitor einsetzen.

Wer größere Änderungen an der Programmierung des Controllers vornehmen möchte, wird sicherlich die Inboard-Programmierung bevorzugen, die auch gleichzeitig das Debuggen ermöglicht, ohne den Mikrocontroller jedes Mal umzustecken. Dabei werden die Signale TDIO, TCK und GND einfach über kurze Litzen mit den entsprechenden Pins auf dem Launchpad (16, 17 und 20) verbunden. Zu beachten ist bei Erweiterungen, dass insgesamt nur 512 Bytes RAM und 16 kByte ROM (256 Bytes beziehungsweise 8 kByte frei) zur Verfügung stehen und manche eingebundene C-Funktionen sowohl RAM als auch ROM sprengen können.

—pek

↓ Links und Foren
make-magazin.de/x17h

Superzeitlupe mit GoPro

Erinnern Sie sich an die Zeitlupeneffekte aus Filmen wie Matrix oder Blade? Wenn Sie eine Action-Kamera wie die GoPro Hero und die dazugehörige kostenlose Software GoPro Studio besitzen, können Sie auch ohne teures Spezialequipment ähnliche Effekte erzielen – egal ob für den Urlaubsfilm oder das nächste YouTube-Tutorial.

von Alexander Moser



Man kennt es aus vielen Actionfilmen: Der Held greift zunächst in normaler Geschwindigkeit zur Waffe und feuert dann nahtlos in Superzeitlupe seine Kugeln ab und bringt den Gegenspieler zur Strecke. Im Kino erreicht man diesen weichen Zeitlupeneffekt mit sogenannten Hochgeschwindigkeitskameras, die mehr als die im Kino üblichen 24 Bilder pro Sekunde aufzeichnen. Ein Film mit beispielsweise 96 Bildern pro Sekunde wird später einfach mit normaler Geschwindigkeit abgespielt und man erhält eine viermal langsamere, flüssige Zeitlupe.

Zum Glück ist die technische Entwicklung inzwischen so weit, dass man in jedem Elektronikmarkt kleine Kameras bekommt, die ebenfalls mit hohen Bildraten aufzeichnen können. Damit kann man nicht nur den Actionfilm aus dem eigenen Garten oder den Quadropterflug über den Acker aufpeppen: Auch die Zuschauer des Tutorials Ihres nächsten Eigenbau-Projekts können davon profitieren, wenn Sie die entscheidenden Bewegungsabläufe in Zeitlupe präsentieren.

Bevor wir jetzt unseren neuen Blockbuster bei YouTube oder Vimeo ins Netz stellen können, benötigen wir natürlich zunächst eine zündende Idee für unser Video. Darin sollte natürlich eine interessante schnelle Bewegung zu sehen sein. Diese Anleitung können Sie aber auch erst einmal ohne eigenes Material nachvollziehen, indem Sie die Datei pingpong.mp4 vom Make-Sever herunterladen (Link am Ende des Artikels).

Hierzu habe ich mir in einem Supermarkt ein Ballspiel mit Fangkörben besorgt. Sie werden nicht glauben, wie oft ich das mit meiner Freundin üben musste, bis wir uns die Bälle zuspielden konnten.

Beachten Sie, dass die Datei im Format 1280 x 720 Pixel bei knapp 120 fps (Frames per second, Bilder pro Sekunde) im NTSC-Modus der Kamera aufgenommen wurde. Das ist wichtig, um später für die viermal so langsame Zeitlupe sauber von 120 auf 30 Bilder pro Sekunde heruntergehen zu können. Solange Sie Ihre Videos am Computer oder auf YouTube nutzen wollen, können Sie auch getrost den NTSC-Modus der Kamera mit den höheren Bildraten nutzen. Nur wenn Sie Ihr Werk noch auf eine Video-DVD brennen wollen, sollten Sie den PAL-Modus mit 25 Bildern pro Sekunden benutzen, um dort Bildruckeln zu vermeiden.

Viele Wege führen zum Ziel

Um den Zeitlupeneffekt zu erreichen, gibt es zwei mögliche Wege. Bei der ersten Variante ist das Vorgehen sehr einfach, aber leider geht dabei der Ton verloren. Bei der zweiten Variante ist der Vorgang mit etwas Arbeit verbunden, aber dafür bekommen Sie dann auch den Ton von der Software in Zeitlupe gerechnet.

Fangen wir mit der einfachen Variante an: Nachdem Sie die Software installiert haben, gibt Ihnen GoPro Studio noch eine kleine Einführung, wie man Videos in die Software importiert. Diesen Part können Sie überspringen. Anders als in Programmen wie Adobe Premiere müssen Sie die Videos für die Verarbeitung in GoPro Studio zunächst in ein ande-

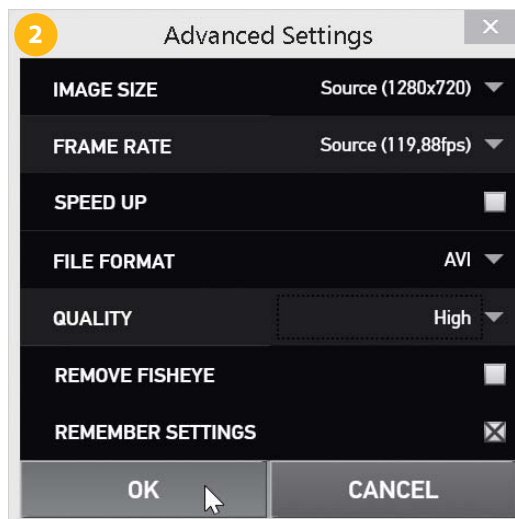


So präsentiert sich GoPro Studio nach dem Öffnen. Die Beispieldatei pingpong.mp4 finden Sie unter dem Link auf Seite 150.

res Format konvertieren: Die MPEG4-Dateien von der GoPro-Kamera sind nämlich so komprimiert, dass ein Hin- und Herspulen oder Schneiden nicht ohne Weiteres möglich ist.

1 Für die Konvertierung klicken Sie oben links auf „Import New Files“ und wählen zunächst unsere Video-Datei pingpong.mp4. Nach dem Import erscheint die Datei und einige Informationen dazu auf der linken Seite. Als Nächstes müssen Sie die Konvertierungseinstellungen überprüfen. Klicken Sie dazu auf das Video, sodass es im großen Hauptbereich erscheint und danach darunter auf „Advanced Settings“. Die richtigen Werte zeigt das untenstehende Bild. 2

Mit OK bestätigen Sie Ihre Eingaben. Rechts neben „Advanced Settings“ befindet sich ein Feld mit dem Dateinamen für die konvertierte Datei. Durch die Konvertierung erhält die neue Datei unter Windows die Endung .avi, auf dem Mac .mov. Die Originaldatei pingpong.mp4 bleibt dabei bestehen. Sie können auch angeben, wo die neue Datei gespeichert werden soll.



KRUMME BILD RATEN

Genau genommen zeichnet die Kamera die Videos nicht mit 30 oder 120 Bildern pro Sekunde auf, sondern mit 29,97 und 119,88. Das geht zum Teil auf die US-amerikanische Farbfernsehnorm NTSC und das dort herrschende Stromnetz mit 60 Hz zurück. Dies im Detail zu erläutern, würde jedoch hier den Rahmen sprengen. Für dieses Tutorial reicht es zu wissen, dass wir aus circa 119,88 Bildern pro Sekunde gute Videos mit 29,97 Bildern bekommen.

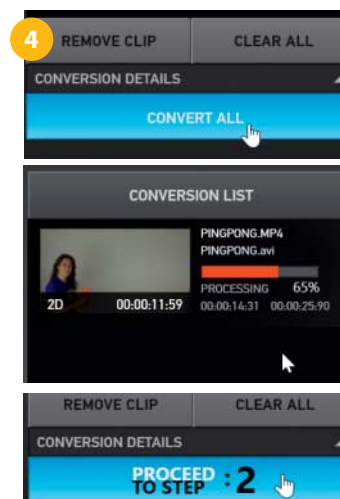
Die Import-Einstellungen für neue Videos: Auflösung und Bildrate beibehalten, hohe Qualität auswählen



Bereits vor dem Konvertieren wählen Sie den gewünschten Ausschnitt aus dem Rohmaterial aus.

Damit GoPro Studio nicht die gesamte Datei mit allen Fehlversuchen des Ballspiels konvertiert, kann man mit dem rautenförmigen Schieberegler unter dem Videofenster den Abschnitt einstellen, der umgewandelt werden soll. Unten links über Rotate/Flip sind zur Markierung der Ein- und Ausstiegspunkte zwei Knöpfe, deren Symbole an geschweifte Klammern erinnern. 3 Als Startzeit (In) habe ich hier mit dem Schieberegler 00:00:14:30 eingestellt und mit dem linken Knopf markiert. Alternativ können Sie auch die Taste I drücken. Als Ausstiegspunkt (Out) habe ich 00:00:25:90 gewählt und mit dem rechten Knopf bestätigt, alternativ Taste O auf der Tastatur.

Die Knöpfe unter dem Videofenster für Play, Step Forward und Step Back können Ihnen bei der genauen Einstellen der Zeitpunkte sehr behilflich sein. Ein Klick auf den blauen Knopf „Add Clip to Conversion List“ stellt den markierten Abschnitt in die Warteschlange. 4 Da wir nur einen Clip haben, können wir die Konvertierung mit „Convert All“ starten. Nach



Mit einem Klick auf „Convert All“ bereitet GoPro Studio die gewünschten Clips für den Schnitt auf.

etwas Rechenzeit verändert sich der Text auf dem blauen „Convert All“-Knopf zu „Proceed to Step 2“.

Clips zerschneiden

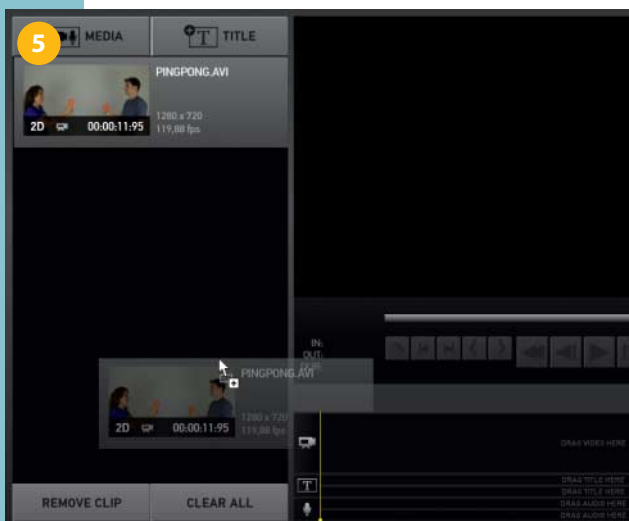
Klicken Sie hier oder oberhalb des Videofensters auf „Step 2 – Edit“. So gelangen Sie zur nächsten Ebene von GoPro Studio, dem Schnittplatz, an dem man das Material schneidet und bearbeitet. Dort angekommen begrüßt Sie zunächst ein Vorlagenassistent, den Sie mit einem Klick auf Cancel gerne links liegen lassen können.

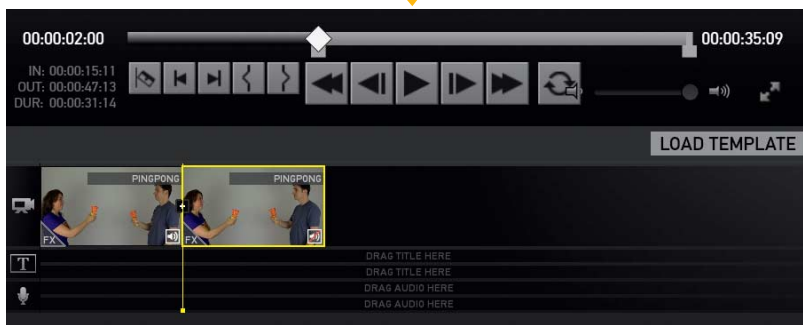
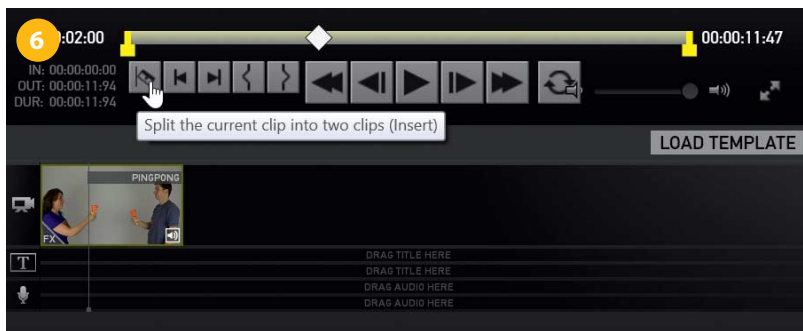
Oben links unter dem Punkt „Media“ finden Sie unseren konvertierten Abschnitt des Videos, das nun den Namen pingpong.avi (oder .mov) trägt und nur noch knapp 12 Sekunden lang ist. 5 Diese Datei ziehen Sie nun auf die mit „Drag Video Here“ beschriftete Zeitleiste (Timeline) unterhalb des noch schwarzen Videofensters in der Mitte. Nachdem Sie die Maustaste losgelassen haben, erscheint ein Vorschaubild sowohl in der Timeline als auch im Videofenster darüber.

Unter dem großen Videofenster schieben wir nun den rautenförmigen Schieberegler an die Zeitposition 00:00:02:000. Genau an dieser Stelle ist nämlich der Ball noch im rechten Korb, bevor er wieder losgeschossen wird. Zur genaueren Positionierung im Video stehen auch hier wieder die Tasten Step Forward und Step Back in der Mitte unterhalb des Videofensters zur Verfügung. Unter dem Videofenster befinden sich links mehrere Knöpfe. Der erste davon sieht aus wie eine Rasierklinge. 6 Mit einem Mausklick darauf teilen Sie das Video an dieser Stelle in zwei Teile.

Unten in der Zeitleiste Ihres Projekts hat sich nun etwas getan: Dort sind nun zwei Vorschaubilder zu sehen. Eines mit dem Startbild des Videos und rechts davon ein Bild, das genau der Stelle entspricht, an der wir den Schnitt platziert haben. Da wir jetzt zwei Videoschnipsel haben, erlaubt uns GoPro Studio, für beide unterschiedliche Einstellungen vorzunehmen.

Ziehen Sie das Video mit der Maus in die Zeitleiste unten im Schnittprogramm.





Ein Klick auf die Rasierklinge zerteilt einen Videoclip.

Die Einstellmöglichkeiten hierzu befinden sich rechts neben dem Videofenster. Neben Menüs für Audio und Farben ist dort unter Video Controls auch eines für die Geschwindigkeit.

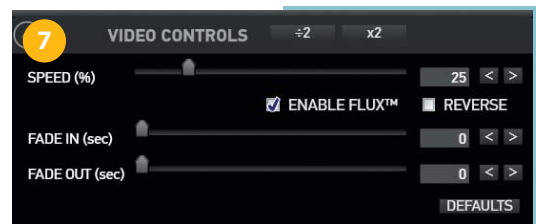
Wählen Sie mit einem Mausklick zunächst das rechte Vorschaubild in der Zeitleiste und klicken Sie dann auf Video Controls. **7** Das Menü klappt auf und gibt unter anderem einen Regler für Speed frei. Vermutlich denken Sie es sich schon, hier können Sie die Geschwindigkeit für unseren gewählten Filmstreifen einstellen. Da unser Video ursprünglich 120 Bilder pro Sekunde hatte und unsere fertige Videodatei später nur noch 30 Bilder pro Sekunde haben soll, können wir nun die Geschwindigkeit auf ein Viertel (25 Prozent) reduzieren und haben dennoch für jedes der 30 Bilder später genau ein Bild, ohne dass GoPro Studio Bilder doppelt zeigen müsste. Würden Sie den Regler zum Beispiel auf 15 Prozent stellen, müsste die Software einige Bilder wiederholen.

Schieben Sie den Regler auf 25 Prozent, um ein Viertel der normalen Geschwindigkeit einzustellen. Jetzt macht sich leider der Nachteil dieser einfachen Variante bemerkbar: Unten in der Zeitleiste erscheint im rechten Vorschaubild ein durchgestrichenes

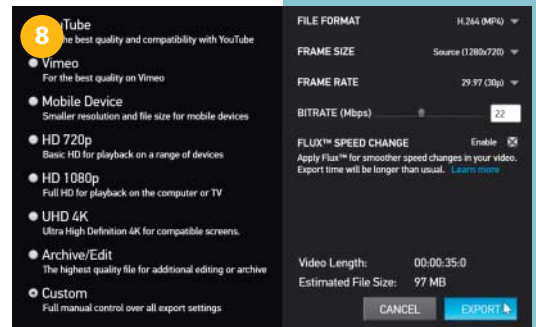
Lautsprechersymbol. GoPro Studio rechnet beim Reduzieren der Geschwindigkeit leider nicht den Ton um, das bedeutet, wir werden nach dem Schnitt in unserer superweichen Zeitlupe keinen Ton haben. Wenn Sie an dieser Stelle aus Gründen der Dramatik sowieso eine Musik einspielen wollen und den Originalton nicht benötigen, ist diese Variante für Sie der einfache Weg zu Ihrem Effekt.

Wenn Sie den Ton aber benötigen, zeige ich Ihnen gleich die andere Variante. Doch zunächst sollten wir erst einmal unser bisheriges Werk bewundern und zur Ausgabe auf Festplatte fertigstellen. Per Klick auf „Step 3 – Export“ können wir unser fertiges Werk als MPEG4-Datei abspeichern. **8** Bei den nun angezeigten fertigen Export-Vorgaben für YouTube und Co. kann es gerade bei Geschwindigkeitsänderungen im Schnitt in der Zeitleiste zu fehlerhaften Ergebnissen kommen. Es ist daher besser, die fertigen Einstellungen zu meiden und „Custom“ zu verwenden, um die Kontrolle über die Ausgabe zu behalten. Da unser Quellmaterial im Format 1280 x 720 Pixel vorliegt, wählen wir für die Ausgabe die Werte, die der Screenshot oben zeigt.

Hier eine höhere Auflösung zu wählen, würde das Video nicht verbessern und zu längeren Re-



Die zweite Hälfte des Videos soll in Zeitlupe laufen. Wir wählen ein Viertel der Originalgeschwindigkeit.



Die empfohlenen Exporteinstellungen. Ein Kreuzchen bei FLUX SPEED CHANGE sorgt für schönere Zeitlupen, treibt aber die Rechenzeiten beim Export in die Höhe.

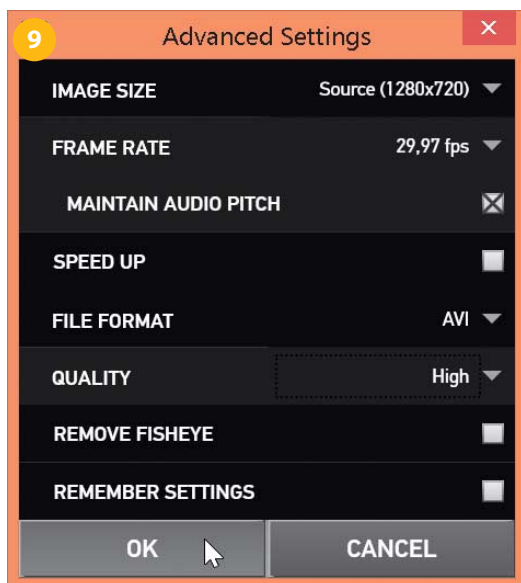
GOPRO STUDIO HERUNTERLADEN

Die Schnittsoftware finden Sie auf der GoPro-Website unter Software & App:

<http://de.gopro.com/software-app/gopro-studio-edit-software>

Beim Start fragt das Programm unter Windows nach der Installation von Apple Quicktime. Das wird für diese Anleitung zwar nicht gebraucht, aber wenn Sie Quicktime installieren, hören diese Erinnerungen auf.

Bei dieser Methode ändern wir die Bildrate des Quellvideos bereits vor der Konvertierung. Alle anderen Werte bleiben bestehen.



chenzeiten führen. Nach dem Klick auf EXPORT geben Sie einen Dateinamen für die Ausgabedatei ein. Nach etwas Rechenzeit ist Ihr fertig geschnittenes Video zum Abspielen bereit.

Die andere Variante mit Ton

Um GoPro Studio auch Ton in der Zeitlupe abzurufen, muss man ein wenig tricksen. Klicken Sie zunächst oben links wieder auf „Step 1“. Hier finden wir wieder unsere Originaldatei vom Anfang. Ganz links sehen wir die Originaldatei pingpong.mp4, ganz rechts in der Conversion List ist die umgerechnete Datei pingpong.avi (oder .mov) für den Schnitt. Im Videofenster in der Mitte sind immer noch unsere Ein- und Ausstiegspunkte markiert. Klicken Sie jetzt noch einmal auf „Advanced Settings“ unten links unter dem Videofenster.

Wieder erscheinen die Optionen für die Konvertierung. 9 Bei Frame Rate wählen Sie nun 29,97 aus und bestätigen mit OK die Eingabe. Klicken Sie nun unten rechts auf „Add Clip to Conversion List“. Oben rechts erscheint eine neue Datei pingpong-001.avi mit dem Status „Waiting“. Klicken Sie auf Convert all, um die Datei für den Schnitt umzurechnen. Nachdem der Computer die Datei fertiggestellt hat, wechseln Sie wieder in den Schnittbereich (Step 2). Dort finden Sie jetzt auf der linken Seite zwei Dateien,

eine mit 119,98 und eine andere mit 29,97 fps. Bei der Konvertierung hat GoPro Studio den Ton umgerechnet und auch die 119,98 Bilder pro Sekunde sind jetzt auf 29,97 Bilder pro Sekunde verteilt, sodass die Datei nun 47 Sekunden Länge hat.

Unter dem Videofenster sehen Sie noch unseren vorherigen Schnitt mit den zwei Vorschaubildern in der Zeitleiste. Wählen Sie den rechten Videoschnipsel, dessen Geschwindigkeit wir auf 25 Prozent gesetzt hatten, und löschen Sie diesen mit der Entfernen-Taste. Sie haben jetzt nur noch ein Vorschaubild in Ihrer Zeitleiste, das bei unserem Schnitt aufhört.

Ziehen Sie jetzt das neu konvertierte Video mit 29,97 Bildern pro Sekunde hinter das Bild in der Zeitleiste. Nun haben Sie wieder zwei Vorschaubilder in der Zeitleiste. Das erste Bild repräsentiert den Anfang des Videos mit normaler Geschwindigkeit, das zweite steht für den Abschnitt, der bereits durch die Konvertierung in Zeitlupe ist. Wenn Sie sich jetzt mit der Abspiel-taste das ganze Video ansehen, werden Sie feststellen, dass das neukonvertierte Zeitlupenvideo den Anfang der Szene noch einmal zeigt. Wir müssen also an diesem Clip vorne noch etwas wegschneiden.

Da wir für unseren Schnitt den Augenblick genau vor dem Losschießen gewählt haben, können wir ihn mit dem rautenförmigen Schieberegler und den Knöpfen „FRAME FORWARD“ und „FRAME BACK“ im neuen Video gut wiederfinden, hier an der Stelle 00:00:05:000. Mit einem Klick auf den Knopf mit der Rasierklinge setzen wir an dieser Stelle wieder einen Schnitt.

Jetzt erscheinen drei Vorschaubilder in der Zeitleiste. Das erste steht für den Anfangsteil mit normaler Geschwindigkeit, das zweite für denselben Teil in Zeitlupe und das dritte für den Teil nach dem Schnitt in Zeitlupe. Das Bild in der Mitte ist also der Teil, den wir wegnehmen wollen. Dazu wählen wir es einfach per Mausklick aus und löschen es mit der Entfernen-Taste. 10 Die Schnipsel links und rechts davon schnappen dann zusammen und wir haben durch etwas Handarbeit unseren Effekt auch mit Ton erreicht. Mit einem Klick auf „Step 3 – Export“ können wir das fertige Video mit den vorhin gemachten Einstellungen auf Festplatte ausgeben. —phs

Links und Foren
make-magazin.de/xuj9

Der mittlere Teil taucht doppelt im Video auf. Ein Klick zum Markieren und die Taste „Entfernen“ sorgen für einen nahtlosen Übergang in die Zeitlupe.

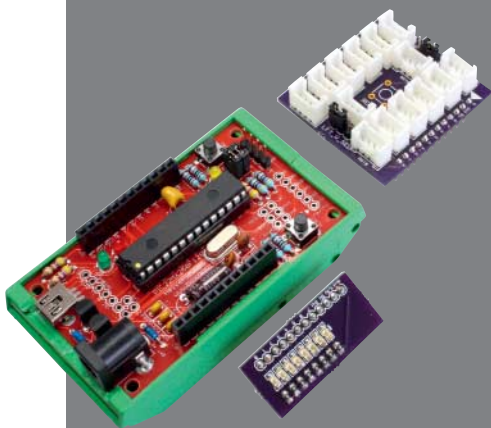


Anzeige

SCHWEIZER BOARD

Mikrocontroller-Board HelvePic32

Schon vor einiger Zeit ist der Hersteller Diligent auf den Arduino-Zug aufgesprungen und bietet mit seiner chipKit-Reihe kompatible Entwicklungs-Boards an, die aber um einiges leistungsfähiger sind. Von dieser auf dem 32-Bit-PIC-Mikrocontroller beruhenden Reihe gibt es nun einen ersten offiziellen Klon aus der Schweizer Community. Unser Autor Mathias Wilhelm hat den HelvePic32 entwickelt, der mit einem



PIC32MX250F128B (40 MHz) arbeitet. Darin stecken 128 KByte Flash und 32 KByte RAM. Da der Controller von Hause aus USB 2.0 unterstützt, kann man sich die Umsetzung von Seriell-auf-USB wie beim Arduino Uno sparen.

Das Board ist für Einsteiger konzipiert und kommt als Bausatz für 17,50 Schweizer Franken. Die Platine ist in Durchsteckmontage gefertigt und wartet darauf, dass man alle Bauteile selbst verlötet. Eine genaue Anleitung steht online zur Verfügung. Die chipKit-Plattform bietet eine mit dem Arduino sehr verwandte IDE, die UECIDE vereint die beiden Plattformen sogar. Die Bibliotheken aus der Arduino-Welt sind meist direkt oder mit kleinen Anpassungen verwendbar. Neben der Software-Kompatibilität sind auch viele Arduino-Shields kompatibel. —*dab*

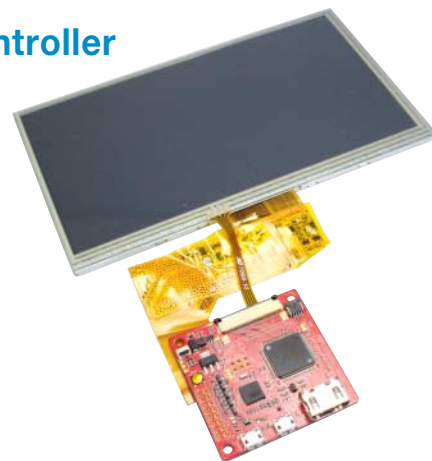
Hersteller Mathias Wilhelm
Distributor <http://shop.boxtec.ch/helvepic32-kit-p-42323.html>
Preis 17,50 CHF

MAUS-DISPLAY

HDMI-Displays mit Touch-Controller

Zwar gibt es für eigene Projekte günstige ARM-Boards in Hülle und Fülle, will man aber etwa sein Mini-Media-Center oder Heimautomationszentrale mit einem größeren Touch-Display ausstatten, hatte man bislang wenig Auswahl. Watterott hat nun zwei Displays in sein Sortiment aufgenommen. Ein 5-Zoll- und ein 7-Zoll-Display mit jeweils 800 × 480 Pixeln und resistivem Touchcontroller für 60 Euro beziehungsweise 66 Euro. Beide kommen mit einem separaten HDMI/DVI-RGB-Adapter mit Touchcontroller. Daneben bietet Watterott ein 7-Zoll-Display mit kapazitiver Touchfunktion an, das ebenfalls mit dem Adapter funktioniert.

Der Controller kommt allerdings mit nichts als einem Bootloader, die Firmware mussten wir erst selbst auf dem Arduino-Klone installieren. Das ist zwar normalerweise im Handumdrehen erledigt, allerdings soll man vorher zwei Dateien in der Arduino-IDE patchen, eine Hardware-Extensions installieren, den Sketch richtig konfigurieren und dann kompilieren. Das ist recht fehlerträchtig; Watterott überlegt bereits, fertige Firmware und einen Uploader zur Verfügung zu stellen.



Ansonsten ist der Touch-Display-Controller, der auch Full-HD ansteuern kann, schön klein und ziemlich durchdacht: Die Touchfunktion wurde so umgesetzt, dass der Controller wie eine Maus arbeitet, ein spezieller Treiber ist damit überflüssig. Der Micro-USB-Anschluss dient zur Stromversorgung und zur Datenübertragung. Das Backlight des Displays lässt sich über den Controller dimmen. —*dab*

Hersteller Watterott
Distributor www.watterott.com
Preis ab 60 €

SCHALTKREIS-STICKER

Chibitronics

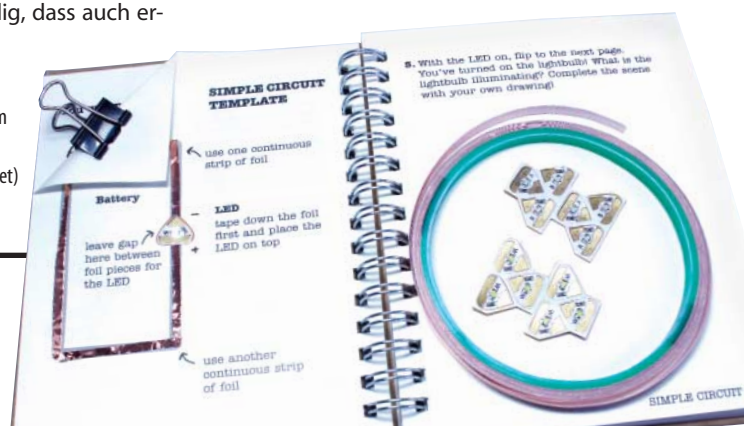
Schaltkreise kann man inzwischen ja nähen, häkeln, stricken und malen. Mit den Chibitronics der Künstlerin Jie Qi kann man sie jetzt auch kleben. Die Sticker sind für Menschen gedacht, die sich in Elektronik einarbeiten wollen. Im Starter-Set erhält man 18 weiße und 18 bunte LED-Sticker, zwei Knopfzellen, zwei Vielzweck-Klemmen und Kupfer-Klebeband. Zusätzlich gibt es das „Circuit Sticker Scetchbook“, das sowohl erklärt, wie die einzelnen Komponenten zu verwenden sind, als auch die Möglichkeit bietet, eigene Entwürfe zu kleben.

Die verschiedenen Sticker sind ein Wunderwerk an Miniatur-Technik – die Arbeit mit ihnen erfordert Fingerspitzengefühl und Geduld. Sehr hilfreich dabei sind die Hinweise im Scetchbook. So ist das Herstellen eines Schaltkreises mit LED und Batterie mit dem Kupferklebeband zuerst so fummelig, dass auch er-

fahrenere Bastler das Aufleuchten der LED bejubeln. Schön sind die kreativen Ideen, wie die, aus einer umgeknickten Ecke einer Heftseite einen Schalter zu machen. Auf der Webseite des Unternehmens findet man Anleitungen für weitere Elektronik-Papier-Basteleien, Video-Tutorials und Projekte von anderen Bastelnden.

Für Elektronik-Frickler sind die klebbaren Schaltkreise, auch wenn man es um zusätzliche Sets mit Sensoren und einem Mikrocontroller-Sticker oder Lichteffect-Stickern ergänzt, fachlich bestimmt keine Herausforderung. Es ist aber ein wirklich schöner Einstieg für Kreative, die sich in Schaltkreis-Design und elektronische Grundlagen einarbeiten wollen. Und für die Elektronik-Spezis, die alles einfach mal anders machen wollen, lohnt es sich ganz sicher auch. —*esk*

Hersteller Chibitronics
Distributor www.chibitronics.com
Preis 29 US-\$ (Starter-Set)
 bis 99 US-\$ (Deluxe-Set)



Anzeige

IMPRESSUM

Make: Nächste Ausgabe erscheint am 25. Juni 2015

Redaktion

Make: Magazin
Postfach 61 04 07, 30604 Hannover
Karl-Wiechert-Allee 10, 30625 Hannover
Telefon: 05 11/53 52-300
Telefax: 05 11/53 52-417
Internet: www.make-magazin.de

Leserbriefe und Fragen zum Heft: info@make-magazin.de
Die E-Mail-Adressen der Redakteure haben die Form xx@make-magazin.de oder xxx@make-magazin.de. Setzen Sie statt „xx“ oder „xxx“ bitte das Redakteurs-Kürzel ein. Die Kürzel finden Sie am Ende der Artikel und hier im Impressum.

Chefredakteur: Daniel Bachfeld (dab)
(verantwortlich für den Textteil)

Stellv. Chefredakteur: Peter König (pek)

Redaktion: Helga Hansen (hch), Rebecca Husemann (rehu), Carsten Meyer (cm), Elke Schick (esk), Philip Steffan (phs)

Mitarbeiter dieser Ausgabe: Beetlebun, Ersan Beksan, Hans Borngräber, Niklas Braband, Sonja Braband, Burkhard Fleischer, Karl Heinz Dahlmann, Florian Fusco, Anika Kehrer, Manfred Kienzle, Ralf Lürig, Friederike Maier, Daniela Metz, Alexander Moser, Marco Nicolai, Roland Pleger, Ulrich Schmerold

Assistenz: Hans-Jürgen Berndt (hjb), Susanne Cölle (suc), Tim Rittmeier (tir), Sebastian Seck (sbs), Christopher Tränkmann (cht), Martin Triadan (mat)

DTP-Produktion: Wolfgang Otto (Ltg.), Martina Bruns, Martina Fredrich, Ines Gehre, Jürgen Gonnermann, Jörg Gottschalk, Birgit Graff, Angela Hilberg, Anja Kref, Martin Kref, Astrid Seifert, Edith Tötsches, Dieter Wahner, Brigitta Zurheiden

Art Direction: Martina Bruns (Junior Art Director)

Layout-Konzept: Martina Bruns

Layout: Stefanie Rosemeyer

Fotografie: Andreas Wodrich, Melissa Ramson

Titelbild: Thomas Saur, www.tsamedien.de

Verlag

Maker Media GmbH
Postfach 61 04 07, 30604 Hannover
Karl-Wiechert-Allee 10, 30625 Hannover
Telefon: 05 11/53 52-0
Telefax: 05 11/53 52-129
Internet: www.make-magazin.de

Herausgeber: Christian Heise, Ansgar Heise

Geschäftsführer: Ansgar Heise, Dr. Alfons Schröder

Mitglied der Geschäftsleitung: Beate Gerold

Verlagsleiter: Dr. Alfons Schröder

Stellv. Verlagsleiter: Daniel Bachfeld

Anzeigenleitung: Michael Hanke (-167)

(verantwortlich für den Anzeigenteil),
www.heise.de/mediadaten/make

Leiter Vertrieb und Marketing: André Lux (-299)

Service Sonderdrucke: Julia Conrades (-156)

Druck: Firmengruppe APPL echter druck GmbH,
Delpstraße 15, 97084 Würzburg

Vertrieb Einzelverkauf:

VU Verlagsunion KG
Am Klingenberg 10
65396 Walluf
Tel.: 0 61 23/62 01 32, Fax: 0 61 23/62 01 332
E-Mail: info@verlagsunion.de

Einzelpreis: 9,90 €; Österreich 10,90 €; Schweiz 17,50 CHF;
Benelux, Italien, Spanien 10,90 €

Abonnement-Preise: Das Jahresabo (6 Ausgaben) kostet inkl. Versandkosten: Inland 52,80 €, Österreich 54,00 €, Schweiz 78,00 CHF; Europa 58,80 €; restl. Ausland 70,20 €. Schüler-/Studentenabo: Inland 32,40 €; Österreich 33,60 €; Schweiz CHF 56,40; Europa 38,40 €; restl. Ausland 49,80 €

Abo-Service:

Bestellungen, Adressänderungen, Lieferprobleme usw.:

Maker Media GmbH

Leserservice

Postfach 24 69

49014 Osnabrück

E-Mail: leserservice@heise.de

Telefon: 05 41/8 00 09-120

Telefax: 05 41/8 00 09-122

Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden. Kein Teil dieser Publikation darf ohne ausdrückliche schriftliche Genehmigung des Verlags in irgendeiner Form reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Die Nutzung der Programme, Schaltpläne und gedruckten Schaltungen ist nur zum Zweck der Fortbildung und zum persönlichen Gebrauch des Lesers gestattet.

Für unverlangt eingesandte Manuskripte kann keine Haftung übernommen werden. Mit Übergabe der Manuskripte und Bilder an die Redaktion erteilt der Verfasser dem Verlag das Exklusivrecht zur Veröffentlichung. Honorierte Arbeiten gehen in das Verfügungsrecht des Verlages über. Sämtliche Veröffentlichungen in Make erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes.

Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.

Printed in Germany. Alle Rechte vorbehalten.
Gedruckt auf Recyclingpapier.

© Copyright 2015 by Maker Media GmbH

ISSN 2364-2548

NACHGEFRAGT



Wozu würdest du einen Roboter benötigen?

Friederike Maier

(Hannover, weiß ab S. 54 alles über Abstandssensoren)

Ein Roboter, der bei Bedarf als Doppelgängerin einspringt, wäre toll. Unangenehme Termine und nervige Arbeit könnte sie dann übernehmen. Dafür würde ich so manches Ölkännchen lockermachen.

Artur Kriening

(Hannover, Praktikant, schreibt für Ausgabe 3 über CubeSats)

Eine vollautomatisierte Küchenhilfe! Ein Rezept oder das vordefinierte Menü auswählen, damit das frisch zubereitete Essen nach der Arbeit auf dem Tisch steht. Nicht nur praktisch, sondern auch gesund.

Rebecca Husemann

(Braunschweig, Praktikantin, Gameboy-Reinschauerin, S. 80)

Mein Roboter sollte meine Schwächen ausgleichen können – ich telefoniere überhaupt nicht gerne und bräuchte vermutlich einen Roboter, der unangenehme Anrufe für mich erledigt. Und zwar souveräner als ich.

Burkhard Fleischer

(Celle, hat das Anemometer auf S.126 gebaut)

Der Roboter hat mein Arbeitszimmer so aufzuräumen, dass alles seinen festen Platz hat. Ferner müsste er mir das gewünschte Buch, das erforderliche Werkzeug, das benötigte Material in die Hand drücken.

Stefanie Rosemeyer

(Hannover, Layouterin bei der Make)

Meinen Roboter würde ich zum Beispiel dafür benötigen, für mich bei Bedarf zur Arbeit zu gehen, zu kochen, Wäsche zu waschen und einkaufen zu gehen. So hätte ich genügend Zeit für das Bereisen fremder Länder.

INSERENTENVERZEICHNIS

Beta LAYOUT GmbH, Aarbergen	21
CadSoft Computer GmbH, Pleiskirchen	17
dpunkt.verlag GmbH, Heidelberg	125
Eurocircuits GmbH, Kettenhausen	156
iFixit GmbH, Stuttgart	23
Messe Erfurt GmbH, Erfurt	41

Messe Friedrichshafen, Friedrichshafen	57
Preisvergleich Internet Services AG, A-Wien	37
Reichelt Elektronik GmbH & Co., Sande	7
Rheinwerk Verlag GmbH, Bonn	33
segor electronics, Berlin	57

Anzeige

Anzeige