

Mit Poster
» Bastelboard-Übersicht
» Smart-Home-Projekte aus Make

IKEA-Hacking

- ▶ Smart-Home-Produkte intelligenter machen
- ▶ Feinstaubsensor ins WLAN bringen
- ▶ Luftreiniger mit ESP8266 steuern



Projekte

- ▶ Pixel-Display mit Retro-Charme
- ▶ Einbruchalarm mit ESP32CAM
- ▶ Smart Home: Jalousien steuern

Know-how

- ▶ So gehts: 3D-Drucke veredeln
- ▶ I²C-Adresse mit Picaxe finden
- ▶ Aber sicher: Arduino für die Industrie

Gehäuse aus Multiplex

- ▶ Schick und individuell
- ▶ Schnell und einfach
- ▶ Frontplatte aus dem Drucker



Wallbox hacken

- ▶ E-Auto per Solar laden



Bild: metamorworks/Shutterstock.com

5/21
30.9.2021
CH CHF 25,80
AT 14,20
Benelux 15,20
€ 12,90





UNSER SORTIMENT VON TECHNIKERN FÜR TECHNIKER

The best part of your project:
www.reichelt.de

Nur das Beste für Sie – von über 900 Markenherstellern.

Unsere Produktmanager sind seit vielen Jahren bei reichelt tätig und kennen die Anforderungen unserer Kunden. Sie stellen ein breites Spektrum an Qualitätsprodukten zusammen, optimal auf den Bedarf in Forschung & Entwicklung, Instandhaltung, IT-Infrastruktur und Kleinserienproduktion sowie auf Maker zugeschnitten.

Holybro X500 V2 Kit, Drohnenbausatz

Mit dem X500KIT bauen Sie eine Drohne, die sich durch individuelle Erweiterbarkeit und durchgängige OpenSource-Technologie auszeichnet. Der komplette Rahmen besteht aus Carbon wodurch die Drohne sowohl eine hohe Stabilität, als auch ein geringes Gewicht aufweist. Das Holybro X500 Kit verfügt außerdem über eine Befestigungsplatte, welche diverse Modifikationen sowie Zubehör ermöglicht.

- Pixhawk 4 Autopilot
- Power Management PM07
- 4 x 2216 KV880 (V2) Motor
- 4 x 1045 Propeller (V2)
- Pixhawk 4 GPS
- 433MHz Telemetrie Radio



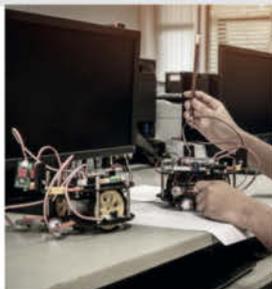
Bestell-Nr.:
X500 KIT

539,⁹⁵



Immer die aktuellste Auswahl
an Robotik, Bausätzen,
und Modulen
finden Sie online!

Gleich entdecken
► www.reichelt.de/robotik



EU-Drohnenverordnung – was zu beachten ist

Die neuen Vorschriften der EASA (Europäischen Agentur für Flugsicherheit) gelten seit dem 31.12.2020 und sollen bezüglich des Einsatzes von Drohnentechnologie Klarheit schaffen. Wir geben einen Überblick über die neuen Regelungen.

Mehr Infos ►



■ Top Preis-Leistungs-Verhältnis

■ über 120.000 ausgesuchte Produkte

■ zuverlässige Lieferung – aus Deutschland in alle Welt

www.reichelt.de

Bestellhotline: +49 (0)4422 955-333

reichelt
elektronik – The best part of your project

Es gelten die gesetzlichen Widerrufsregelungen. Alle angegebenen Preise in € inklusive der gesetzlichen MwSt., zzgl. Versandkosten für den gesamten Warenkorb. Es gelten ausschließlich unsere AGB (unter www.reichelt.de/agb, im Katalog oder auf Anforderung). Abbildungen ähnlich. Druckfehler, Irrtümer und Preisänderungen vorbehalten. reichelt elektronik GmbH & Co. KG, Elektronikring 1, 26452 Sande, Tel.: +49 (0)4422 955-333

TAGESPREISE! Preisstand: 6. 9. 2021

Nostalgie ohne Reue

Normalerweise verbindet man mit Nostalgie schöne Gefühle. Ich hingegen bekomme in letzter Zeit in dem Zusammenhang regelmäßig schlechte Laune. Weil ich nämlich feststelle, dass ich häufig Sachen wegwerfe, die ich doch noch hätte gebrauchen können. Und schlimmer noch: Weil ich bei den letzten Umzügen Dinge entsorgt habe, die mir vor langer Zeit Freude bereitet haben und die nun in der über mich hinwegschwappenden 80'er-Retrowelle wieder eine wichtige Rolle spielen.

Jüngere Familienmitglieder hören plötzlich Langspielplatten und fragen mich nach alten Musikkassetten: „Wie, einfach weggeschmissen?“ Der Datasetten-Artikel auf S. 62 warf beim Gegenlesen bei mir die Frage auf, in welch' geistiger Umnachtung ich einst meinen C64 mit originale Monitor, Floppy, Drucker und Akustikkoppler an den Straßenrand zur Mitnahme stellte (1996, Bremen, Admiralstraße, vielleicht erinnert sich jemand und gibt ihn mir zurück)? Und bei der Recherche zu TTL-Chips in LED-Panels (Artikel S. 56) vermisste ich mein altes (gedrucktes) Handbuch von Texas Instruments – ich hab's wohl zum Papierrecycling gebracht.

Nun stehe ich vor meinen immer voller werdenden Bastelkisten, Sammlungen an technischen Devotionalien und Büchern und

merke, der freie Platz wird immer weniger. Trotzdem traue ich mich nicht mehr, irgend etwas wegzuwerfen. Könnte der erste Raspberry Pi B irgendwann noch eine Rolle spielen? Brauche ich diese Z80-Platine mit CP/M noch (N8VEM)? Sollte ich noch alte SRAMs und EPROMs aufheben? Und wann haben wir eigentlich zuletzt mit der Wii und der PSP gespielt?

Denn wenn ich mich heute davon trenne, kommt in zwei Wochen garantiert der Moment, wo ich es bereue. Droht mir nun ein Leben als Maker-Messi? Muss ich Lagerräume anmieten? Wie gehen Sie, liebe Leserinnen und Leser damit um? Ich hoffe, Sie können leichter loslassen als ich! Damit Sie ohne Reue zurückblicken können.

► make-magazin.de/xv9j

Daniel Bachfeld
Daniel Bachfeld

Sagen Sie uns Ihre Meinung!

mail@make-magazin.de



Inhalt

Wallbox hacken

Triggerwarnung: Es geht um Elektroautos. Und Solarstrom! Unser Projekt lädt das Fortbewegungsmittel der Stunde nachhaltig stets mit dem gerade aus der heimischen Photovoltaik-Anlage verfügbaren Strom. Ganz kostenlos! Die Steuerung versucht dabei, dem Energieversorger möglichst wenig zurückzuschicken.

20 E-Auto per Solar laden



IKEA-Hacking

Wer sich trotz Duftkerzen- und Stehrüchchen-Abteilung in das unmögliche Möbelhaus wagt, findet bisweilen hackenswerte Kleinode mit erstaunlichem Gebrauchswert: Einen Luftfilter etwa, der darauf wartet, mit unserer CO₂-Ampel gesteuert zu werden, oder den Feinstaubsensor Vindriktning, der mit einem ESP8266 ganz einfach an das heimische IoT angebunden wird.

- 8 Luftreiniger mit ESP8266 steuern
- 14 Feinstaubsensor ins WLAN bringen

- 3 Editorial
- 6 Leserforum
- 8 **Projekt: CO₂-Ampel steuert IKEA-Luftfilter**
- 13 **Poster-Beilage: Bastelboard-Übersicht und Smart-Home-Projekte aus Make**
- 14 **Projekt: IKEA-Staubsensor mit WLAN aufbohren**
- 20 **Projekt: Photovoltaik an der E-Auto-Wallbox**
- 28 Werkstattberichte: Neues aus der Szene, Comic
- 30 Projekt: Die Welt als LED-Kugel
- 38 **Projekt: Smartes Alarmsystem mit ESP32-CAM**
- 44 Was uns inspiriert: Klang-Iglu, Stick-Software, Flechtwerk aus dem 3D-Drucker
- 48 Projekt: Gervifonn – CD-Cover erkennen mit KI
- 56 **Projekt: Pixel-Display mit Retro-Charme**
- 62 Projekt: Raspberry Pi als SD-Karten-Laufwerk für C64 & Co.
- 66 Projekt: Booster für E-Gitarren
- 72 Community-Projekt: KRITIS Powerbox
- 74 Community-Projekt: Cyborg Control

Elektronik-Projekte

Sie haben gar keine E-Gitarre für unsere Booster-Tretmine? Macht nix, wir haben noch mehr Lötkolbenfutter im Heft: Für die eigenen vier Wände etwa die Home-Assistent-Jalousiesteuerung oder das smarte ESP32-Alarmsystem, und als Hingucker ein vollfarbiges POV-Rotationsdisplay, das mit fertigen LED-Streifen arbeitet.

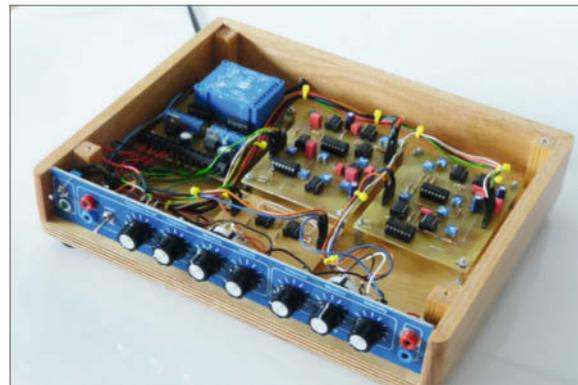
- 30 Die Welt als LED-Kugel
- 38 Einbruchalarm mit ESP32CAM
- 66 Booster für E-Gitarren



Schicke Gehäuse

Mit einer ansprechenden Behausung für das eigene Elektronik-Projekt hadern noch viele Maker, und nur selten genügt eine Zigarrenkiste oder eine Keksdose höheren ästhetischen Ansprüchen. Dabei ist es eigentlich ganz einfach, mit Multiplex-Sperrholz und Alu-Leisten ein schickes Gehäuse zu bauen.

- 112 Gehäuse aus Multiplex
- 116 Gehäusebau mit FreeCAD



- 76 Community-Projekt: Ein Tablet für's Auto
- 78 Projekt: Zink-Sauerstoff-Akku selber herstellen
- 86 **Workshop: Arduino für die Industrie**
- 92 **TV und Heimkino mit Home Assistant, Teil 2: Jalousien steuern**
- 102 Reingeschaut: Lichtpistole „Super Marksman“
- 104 **Projekt: I²C-Adressen mit Picaxe finden**
- 108 Holz-Challenge: Die Gewinner
- 112 **Workshop: Gehäuse aus Multiplex**
- 116 **Workshop: Gehäusebau mit FreeCAD, Teil 1**
- 120 **Workshop: 3D-Druck-Veredlung**
- 124 Kurzvorstellungen: Mikrocontroller, ESP-Klaviatur, Lernpakete, CAD-Software, Entlötkolben, Punktschweißgerät
- 128 Medien: Leiterplatten-Design, Raspi-Pico-Einstieg, Corona-Experimente, Kinderfahrzeuge hacken, Maker verbessern die Welt
- 130 Impressum/Nachgefragt

Themen von der Titelseite sind rot gesetzt.

CD-Player mit Cover-Erkennung

Zum CD-Regal gehen und mit verträumtem Blick ein Album auswählen - dieses in Streaming-Zeiten aussterbende Kulturgut belebt unser CD-Player aufs Neue. Das Magische daran: Man muss die Silberscheibe zum Anhören gar nicht entnehmen, es reicht, dem künstlich intelligenten Player das Cover zu zeigen. Wie geil ist das denn!

- 48 Gervifonn: CD-Cover erkennen mit KI



Leserforum

Fehler im Schaltplan

WLAN-Alarmkontakt mit Batteriebetrieb,
Make 4/21, S. 68

Im Artikel zum WLAN-Alarmkontakt in der vorigen Make-Ausgabe ist uns leider ein Fehler im Schaltplan und den Platinendateien unterlaufen: GPIO 16 sollte **nicht** angeschlossen sein. Wir haben auf GitHub (siehe Link zum Artikel) den Schaltplan und die Platinendateien aktualisiert.

► make-magazin.de/xpf2

Vertauschte MOSFETs

Ein- und Ausschalten per Softswitch,
Make 4/21, S. 76

In der Kurzinfo/Bauteileliste zum Artikel werden für T2 die MOSFETs IRF7415 bzw. IRML2502 genannt und von Typ *P-Channel* gekennzeichnet. Laut Datenblatt handelt es sich in beiden Fällen aber um *N-Channel*-Typen. Wenn man der Schaltsymbolik in der Abbildung 1 vertraut, wäre das auch so. Dagegen kämen wohl in den Schaltungen 4 und 7 für T2 offenbar tatsächlich *P-Channel*-MOSFETs (ohne nähere Bezeichnung) zum Einsatz. Sehe ich das richtig?

zaboraner (übers Online-Forum)

Sie haben vollkommen recht, da haben wir leider genau alle MOSFETs in der Kurzinfo vertauscht.

Richtig wären, wie auch im Online-Forum beschrieben, der IRF7410 oder der IRLML 6402. Danke für den Hinweis!

Mit dem Zero ins All?

Astrofotografie mit Pi und HQ-Kamera,
Make 4/21, S. 30

Danke für Ihren gelungenen Artikel zur Astrofotografie. Als ich meinen Kindern von diesem Projekt berichtete, waren sie begeistert, da sie auch schon in den letzten Tage sehr viele der Perseiden-Sternschnuppen beobachten konnten.

Zu dem Aufbau ergeben sich für mich die folgenden Fragen:

1. Welches sind die Mindestvoraussetzungen für den Raspberry Pi (Modell, Speicher)?
2. Bei der Recherche bin ich auf Aufbauten mit dem Raspberry Pi Zero W mit 512MB gestoßen. Was sind die Nachteile dieser Konfiguration?

Alexander Abraham

Wir würden immer den Raspi 4 empfehlen, das ist einfach aktuell der preiswerteste. Pi1 oder 2 und damit auch der Zero sind wohl eher zu langsam, wenn man, wie im Artikel empfohlen, per VNC und GUI arbeiten möchte. Für eine All-sky-Kamera, die nur Bilder per Netzwerk oder Website liefert, wird es aber wohl auch reichen. Ansonsten eben per Kommandozeile anspre-

chen und die Bilder auf einem Netzwerk-Laufwerk speichern. Dann ist es aber recht schwierig mit dem Fokussieren ohne Livebild. Falls ein Raspi 3 vorhanden ist, lohnte sich der Versuch per VNC. Wenn man irgendwie einen Monitor per HDMI anschließt, kann man natürlich auch raspistill/raspivid nutzen und die Echtzeitvorschau ohne GUI bekommen.

raspistill-Vorschau per VNC

Wenn Sie im VNC-Server (Raspberry mit verbundener Kamera) in der Taskleiste auf das VNC-Icon klicken, in dem sich öffnenden Fenster oben rechts auf das „Burger-Menü“ gehen, dort *Optionen/Problembehandlung* öffnen, bei *Bildschirmaufnahme optimieren* den *Direkterfassungsmode* aktivieren, wird auch die Vorschau von raspistill auf den VNC-Client über das Netzwerk übertragen.

Jörg Henkel

Danke für den Tipp! Da haben wir wohl nicht tief genug gebuddelt bei der Recherche oder zu sehr auf veraltetes Wissen verlassen ...

Klimaschädling Beton

Betonfiguren aus der 3D-Druckform,
Make 4/21, S. 88

„Es gibt tausende Ideen, was man mit Beton machen kann. Beton ist billig und leicht verfügbar.“ Ich würde mir wünschen, die Autor*innen würden solche Sätze im Zeitalter des menschengemachten Klimawandels einfach mal sein lassen. Die weltweite Betonproduktion ist für etwa 9% aller menschengemachten CO₂-Emissionen verantwortlich, und man sollte mit solchen Materialien sparsam umgehen, und sie nur da einsetzen, wo man sie wirklich braucht.

Jörg Zschimmer

Im Prinzip haben Sie natürlich recht. Allerdings wurden allein im vergangenen Jahr weltweit über 4 Milliarden Tonnen Beton produziert; gemessen daran werden die geringen Mengen des Materials, die durch unseren Artikel angeregt vielleicht zusätzlich verbraucht werden, kaum ins Gewicht fallen. Aber generell sind auch Maker immer in der Verantwortung, für sich und die

Kontakt zur Redaktion

Leserbriefe bitte an:

heise.de/make/kontakt/

Wir behalten uns vor, Zuschriften unter Umständen ohne weitere Nachfrage zu veröffentlichen; wenn Sie das nicht möchten, weisen Sie uns bitte in Ihrer Mail darauf hin.

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

www.make-magazin/forum

 www.facebook.com/MakeMagazinDE

 www.twitter.com/MakeMagazinDE

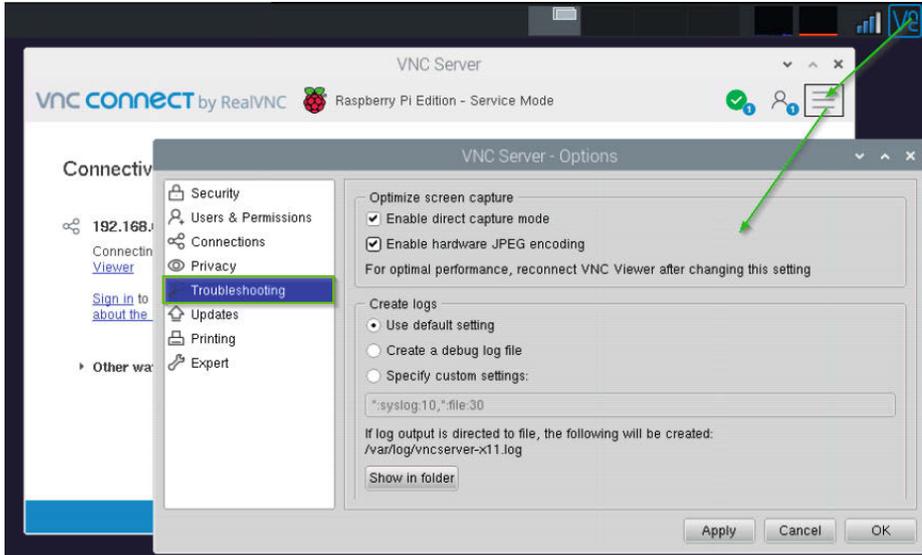
 [instagram.com/MakeMagazinDE](https://www.instagram.com/MakeMagazinDE)

 [pinterest.com/MakeMagazinDE](https://www.pinterest.com/MakeMagazinDE)

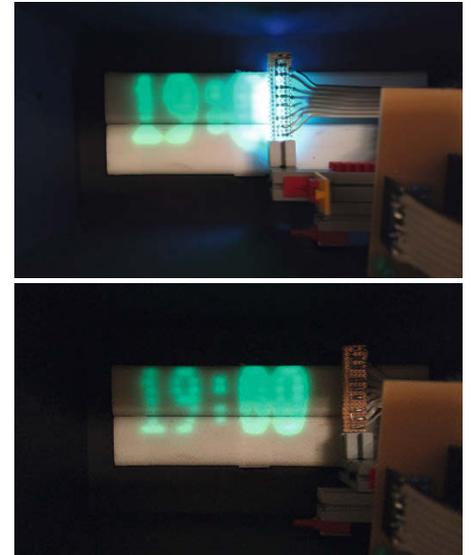
 [youtube.com/MakeMagazinDE](https://www.youtube.com/MakeMagazinDE)

Korrekturen

Manchmal unterläuft uns ein Fehler, der dringend korrigiert gehört. Solche Informationen drucken wir weiterhin auf den Leserbriefseiten im Heft, aber seit Ausgabe 1/17 finden Sie alle Ergänzungen und Berichtigungen zu einzelnen Heft-Artikeln auch zusätzlich über den Link in der Kurzinfo am Anfang des jeweiligen Artikels.



So geht's: Vorschau von raspistill übers Netzwerk auf den VNC-Client übertragen



Die Leuchtziffer-Uhr von Andreas Döring

Welt zu entscheiden: Muss das jetzt sein? Denn auch andere Materialien, Komponenten und Dienste, die Maker üblicherweise verwenden, haben ihren eigenen CO₂-Fußabdruck.

Stromsparen mit dem Nano

Strom sparen bei ESP-Mikrocontrollern, Make 3/21, S. 86

Vielen Dank für Euren Artikel zum Stromsparen mit den ESPs. Kürzlich habe ich für eine Vitrine eine Uhr gebaut, die mittels UV-LEDs

nach guter alter Dot-matrix-printer-Manier eine Uhrzeit auf einen Phosphoreszenz-leuchtstreifen „brennt“. Die Ziffern verblassen dann nach einiger Zeit und können nach zwei Minuten aktualisiert werden. Da in der Vitrine kein Strom liegt, muss die Uhr mit einem Akku betrieben werden. Der Nano zieht etwa 25mA, das sind in 1:50 Minuten 2750mAs, während des 10-sekündigen Schreibvorgangs benötigt er etwa 120mA, also insgesamt nur 1200mAs. Mehr als das Doppelte des Energieverbrauchs geht also für's Nichtstun drauf.

Also habe ich die Power-LED des Nanos mechanisch entfernt und außerdem in den Wartezeiten die Stromsparfunktionen akti-

viert; er geht jetzt für jeweils 1000ms in den Energiesparmodus und schaut dann an Hand der RTC nach, ob es schon wieder Zeit ist, die neue Uhrzeit anzuzeigen. Jetzt benötigt die Uhr während der Wartezeit nur noch 10mA, das ist angesichts der Tatsache, dass er weiterhin mit dem bordeigenen Spannungswandler versorgt wird, ganz akzeptabel. In 1:50 Minuten werden nur noch 1100mAs benötigt, in Summe entspricht dies einer mittleren Stromstärke von etwas mehr als 20 mA. Der 3200 mAh-Akku hält damit locker eine 5-Tage-Woche: Ziel erreicht!

Andreas Döring

iENA
4.-7. Nov. 2021
Internationale Fachmesse
• Ideen
• Erfindungen
• Neuheiten
www.iena.de

6.+7. Nov. 2021
Hack & Make
Das Kreativ- und Technikfestival

4.+5.11.2021
iINNOVATIONS KONGRESS
Messe Nürnberg · NCC West

2-Tages Kongressticket
€ 80,- statt € 120,-
Promocode: MAKE21INKO
iena.de/ticket

Partner:

CO₂-Ampel steuert IKEA-Luftfilter

Als Abhilfe bei schlechter Luft bietet Ikea jetzt den Luftreiniger FÖRNUFTIG. Mit der CO₂-Ampel als Steuerung läuft er nur dann, wenn es nötig ist.

von Guido Burger



Die Pandemie hat uns gelehrt, dass eine gute Luftqualität gerade in viel genutzten Innenräumen sehr wichtig ist. Neben der Vermeidung von Infektionen über den Luftweg, wie bei COVID-19, steigert frische Luft auch das allgemeine Wohlbefinden und die Konzentration. Genug Gründe, einen Automaten zu bauen, der die Luftreinigung für mich übernimmt.

Statt einen Luftfilter komplett selber zu konstruieren, fiel meine Wahl auf den brandneuen Filter *Förnuftig* vom schwedischen Möbelhaus Ikea. Dieser ist mit einem Preis von 49,99 Euro günstig und es gibt ihn in zwei Ausbaustufen: mit dem Partikelfilter HEPA oder zusätzlich mit einem Aktivkohlefilter. Außerdem lässt er sich einfach öffnen und ist somit schnell gehackt. Die automatische Steuerung übernimmt ein ESP-basierter Mikrocontroller mit Umweltsensoren und einem Spannungswandler. Mehr Hardware ist nicht nötig.

Das Projekt bietet sich auch zum Nachbau im Rahmen des MINT-Unterrichts an Schulen an. Hier können von Grundlagen zum Internet der Dinge über die Regelungstechnik bis zur Datenauswertung zahlreiche Aspekte anschaulich vermittelt werden.

Die Platine

In der Make 5/20 hatte ich eine CO₂-Ampel auf Basis des IoT Octopus vorgestellt. Inzwischen habe ich dafür eine spezielle CO₂-Ampel-Platine entworfen, die es als DIY-Bausatz mit einem NodeMCU und dem CO auf *Tindie* zu kaufen gibt und die ich hier genutzt habe. Zusammen mit einem Spannungswandler reicht diese CO₂-Ampel bereits aus, um den Förnuftig automatisch zu steuern. Ich habe die Platine aus Komfortgründen noch um den Umweltsensor BME680, eine 7-Segment-Anzeige und zwei RGB-LEDs für die optische Darstellung der Messergebnisse erweitert. Der BME680 kann entweder auf der Bausatz-Platine verlötet oder über einen Grove-Anschluss angesteckt werden.

Ebenso einfach ist der Nachbau mit dem originalen IoT Octopus. Auf diesem Board sind der BME680 und zwei RGB-LEDs bereits integriert. Der CO₂-Sensor SCD30 kann bei Bedarf als Aufsteckplatine nachgerüstet bzw. aufgesteckt werden. Schließlich können auch zahlreiche weitere ESP8266-basierte Platinen wie der Wemos D1, NodeMCU V2 oder Adafruit Feather Huzzah 8266 genutzt werden.

Der BME680 misst die organischen Komponenten (VOC, volatile organic components), die etwa durch Stoffwechselprozesse in unserem Körper entstehen und sich mit der Zeit in geschlossenen Räumen ansammeln. Weitere Schadstoffquellen sind Rauchen, Ausdünstungen von Putzmitteln, Möbeln und Teppichen, oder die Schwebeteilchen entstehen beim Kochen. Ist die Konzentration dieser Partikel im Raum zu hoch, leidet das Wohlbefinden und es können sich Kopfschmerzen einstellen.

Kurzinfo

- » Ikea Förnuftig hacken
- » Luftqualität mit Gassensor BME680 bestimmen

Checkliste



Zeitaufwand:
ein Nachmittag



Kosten:
ab 75 Euro

Mehr zum Thema

- » Guido Burger, Richard Fix und Klaus-Uwe Gollmer, Der CO₂-Warner für die Schule, Make 5/20, S. 10
- » Heinz Behling, Antiviren-Programme, Make 6/20, S. 28
- » Guido Burger, IoT Octopus, Make-Sonderheft IoT, S. 16

Material

- » IKEA Förnuftig Luftreiniger
- » Förnuftig Gasfilter
- » ESP-basierter Mikrocontroller z.B. IoT Octopus
- » Umweltsensor BME680
- » Spannungswandler 24V nach 5V
- » Jumperkabel
- » USB-Kabel

Alles zum Artikel im Web unter make-magazin.de/x7ra

Auch der CO₂-Anteil der Luft, den ich mit dem SCD30 messe, ist ein guter Indikator für die Luftqualität. In der Make 5/20 sind die Hintergründe dazu und Messprinzipien der beiden Sensoren genauer erklärt.

Um die gemessenen Werte direkt im Blick zu haben, kann die Ampel mit verschiedenen Anzeigemöglichkeiten ausgestattet werden. Neben der 7-Segment-Anzeige sind das zum Beispiel LED-Matrizen von Adafruit im Feather-Wing-Format zum Aufstecken oder eine rot-gelb-grüne LED-Ampel.

Ikea-Luftreiniger

Anfang des Jahres hat Ikea mit dem Förnuftig einen eigenen Luftreiniger ins Programm genommen. Er misst 31 × 45 Zentimeter und wiegt rund 3,6 Kilogramm, sodass man ihn an der Wand aufhängen oder mit dem mitgelieferten Ständer hinstellen kann. Über ein beigelegtes 24-Volt-Netzteil wird er mit Spannung versorgt. Mit nur drei Betriebsstufen für den Ventilator im Inneren ist seine Bedienung äußerst übersichtlich. Diese werden außerdem einfach über drei unterschiedliche Frequenzen gesteuert. Mittels Oszilloskop konnte ich die Ansteuerfrequenzen identifizieren.

Der Luftreiniger kann einfach geöffnet werden, um die Filter auszutauschen. Standardmäßig ist nur der HEPA-Filter installiert. Der Gasfilter aus Aktivkohle wird für dieses Projekt ebenfalls benötigt. Beide Filter sollten im Betrieb dann regelmäßig erneuert werden.

Hochleistungs-Partikelfilter (HEPA, High Efficiency-Particulate Airfilter) fangen Schwebeteilchen aller Art auf. Dies ist zunächst grober Hausstaub, den man als graue Schicht

nach einigen Wochen Betrieb auf dem Filter sehen kann. Aber auch kleine Partikel, wie zum Beispiel Aerosole, verfangen sich im Glasfasergewebe des Filtermaterials. Je nach Filterleistung werden die Geräte in unterschiedliche Klassen eingeteilt: Ein Filter der Klasse 12, wie von Ikea ursprünglich für den Förnuftig angegeben, besitzt einen Abscheidungsgrad von 99,5 Prozent. Damit kommen von 100.000 Teilchen nur 500 durch das Filtergewebe. Professionelle Luftfilter der HEPA Klasse 14 besitzen allerdings einen Abscheidungsgrad von 99,995 Prozent und können also deutlich mehr Partikel zurückhalten. Inzwischen verzichtet Ikea auf die Angabe einer HEPA-Klasse.



CO₂-Ampel (DIY Octopus) mit 7-Segment Anzeige, zwei Status-LEDs und bestücktem BME680

Die Lüfterstufen im Vergleich

Lüfterstufe	Lüfterdurchsatz m³/h	Lautstärke dB(A)	Ansteuerfrequenz Hz
I	25-30	28	100
II	80-90	49	220
III	120-140	60	300



Bei einer Ansteuerfrequenz von 220 Hertz läuft der Föhn auf der mittleren Stufe.



Um einen durchschnittlichen Klassenraum möglichst geräuschlos zu reinigen, wären zehn Geräte auf Stufe II nötig.

Solche Luftfilter fangen Feinstaub-Partikel auf, gasförmige Stoffe können das Filtermaterial jedoch problemlos passieren. So sammeln sich CO₂ und organische Verbindungen aus

der Ausatemluft in der Raumluft an. Da wir neben den Aerosolen auch die VOCs aus der Luft entfernen möchten, ist der Aktivkohlefilter unumgänglich. Aufgrund der großen Oberfläche der porösen Aktivkohle kann er weitere Verbindungen binden. Das CO₂ in der „verbrauchten Luft“ passiert aber auch die Filterstufe aus Aktivkohle. Deshalb kann ein Luftfilter die regelmäßige Fensterlüftung nicht ersetzen.

Steuerung

Das Board soll den Luftreiniger nach dem folgenden Prinzip automatisch steuern: Sobald ein Grenzwert für VOCs überschritten wird, wird die Drehzahl des Ventilators erhöht, damit die VOCs aktiv aus der Luft gefiltert werden. Wenn der Grenzwert wieder unterschritten wurde, kann die Drehzahl und damit

die Lärmbelastung wieder reduziert werden. Bei den Grenzwerten habe ich mich an der internationalen Richtlinie für die „Innenluftqualität“ (iAQ) orientiert, die auch in der Sensorbibliothek genutzt wird. So liefert der BME680 iAQ-Werte zwischen 0 und 500. Von 0 bis 100 ist die Luft sauber. Bei Werten zwischen 101 und 150 wird Ventilation empfohlen und darüber hinaus sollte zusätzlich gelüftet, also frische Luft von außen zugeführt werden.

Da wir nun unseren Regelkreis „Input“ haben, können wir uns um den „Output“ der Steuerung kümmern. Dazu nutze ich einfach die bereits ermittelten Ansteuerfrequenzen: 100, 220 und 300 Hertz.

Jetzt wird gebastelt!

Um die Steuerung zu ersetzen, habe ich den Reiniger geöffnet, die Steuerungselektronik ausgebaut und durch den Spannungswandler ersetzt. Die Platine mit Sensoren wird nicht eingebaut, damit sie den Raum überwachen kann. Die Stoffverkleidung und die Filter des Luftreinigers sind nur aufgesteckt und können einfach abgezogen werden. Um die obere innere Abdeckung zu öffnen, müssen dann zwei Schrauben gelöst werden. Die beiden Drehregler können ebenfalls abgezogen und ihre Sicherungsmutter entfernen werden.

Die Steuerplatine ist mit zwei Schrauben gesichert und ansonsten nur eingesteckt. Nach dem Ausbau sind noch die beiden Kabel zu ziehen, an denen die Spannungsversorgung (P1) und die Ansteuerung des Ventilators (P2) hängen.

Nun habe ich den Spannungswandler (24 zu 5 Volt) eingebaut und zwischen P1 und P2 geschaltet. Hier kann man einfach Jumperkabel nutzen. P1 liefert die Spannung direkt vom externen Steckernetzteil (gelbes Kabel). Über die Terminalklemme (blau) liefert der Spannungswandler die 24 Volt an P2 und damit den Ventilator. Außerdem versorgt er die CO₂-Ampel mit 5 Volt über einen USB-Stecker. Die Steuerleitung CLK des Ventilators wird gleich mit Pin 15 der Ampel verbunden (rotes Kabel).

Dazu müssen das USB-Kabel und die Steuerleitung nach außen geführt werden. Dafür



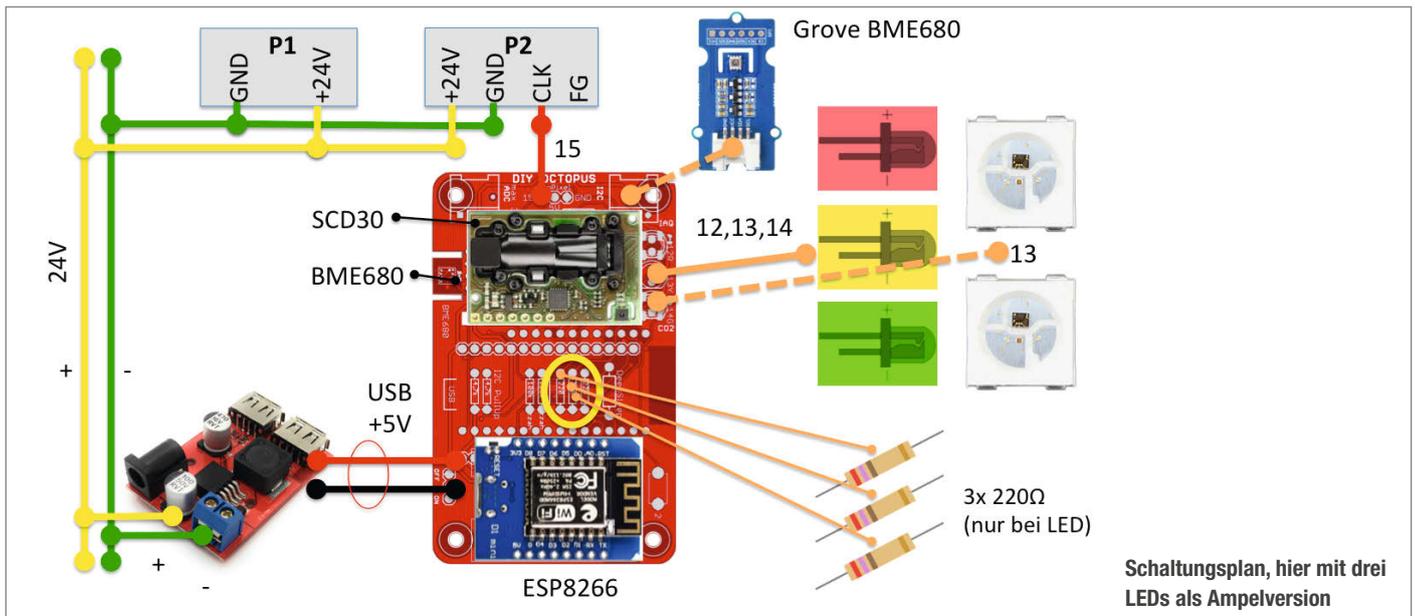
Abschrauben der Abdeckung der Elektronik



Darunter kommt die Steuerplatine zum Vorschein.



Rückseite der Platine mit den beiden Steckern



habe ich die Öffnung des nun überflüssigen Drehreglers genutzt. Je nach verwendetem Kabel muss das Loch eventuell noch etwas aufgebohrt werden.

Anschließend habe ich die Abdeckung wieder aufgesetzt und mit den beiden Schrauben fixiert. Dabei dürfen die innenliegenden Kabel nicht gequetscht werden. Mit dem Einsetzen der Filter und der Stoffverkleidung ist der Luftreiniger wieder komplett. Nun muss nur noch die CO₂-Ampel an das USB-Kabel angesteckt und das Steuerkabel mit dem Pin 15 verbunden werden. Da die Ampel außen liegt, kann ich jederzeit neue Programme aufspielen.

Beim Aufstellen des Luftreinigers darf sie später nicht im Luftzug stehen, da sie sonst keine genauen Messwerte ermitteln kann.

Das Programm

Wie bei der ersten CO₂-Ampel habe ich zum Programmieren die IoT-Werkstatt genutzt – eine Variante der kostenlosen Arduino-Pro-

grammierumgebung. Im Download (siehe Kurzinfo) sind alle notwendigen Bibliotheken enthalten sowie die ArduinoBlock-Erweiterung, die das block-basierte Programmieren in der IDE ermöglicht. Der Fokus der IoT-Werkstatt liegt, wie der Name bereits sagt, auf dem Internet der Dinge und dem WLAN-fähigen ESP8266. Für die hier verwendeten und weitere Sensoren sind passende Programmierblöcke enthalten.

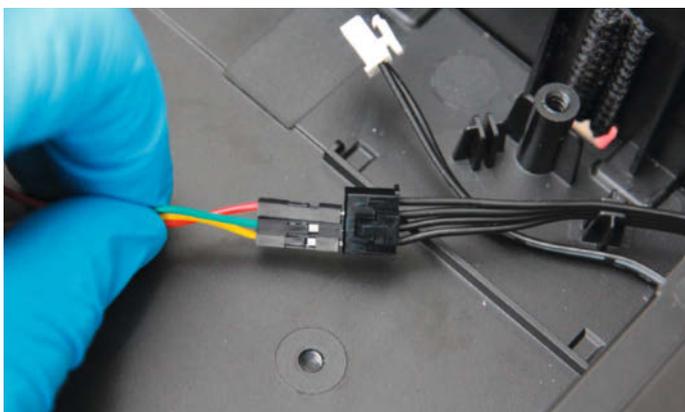
Der Code besteht aus einem Haupt- und zwei Unterprogrammen, in denen die Auswertung der Messwerte jeweils getrennt erfolgt (*TestCO2* und *TestIAQ*). Die Ansteuerung des Ventilators erfolgt dabei über die ESP8266-Befehle `analogWriteFreq` und `analogWrite`, die über den Baustein *C Code* (unter *Kontrollstrukturen*) eingebunden werden können. Über *Kontrollstrukturen* können auch die Unterprogramme eingebunden werden.

Mit `analogWriteFreq(0)` wird die Frequenz beim Start des Programms auf 0 Hertz gesetzt und mit `analogWrite(15, 100)` auf Pin 15

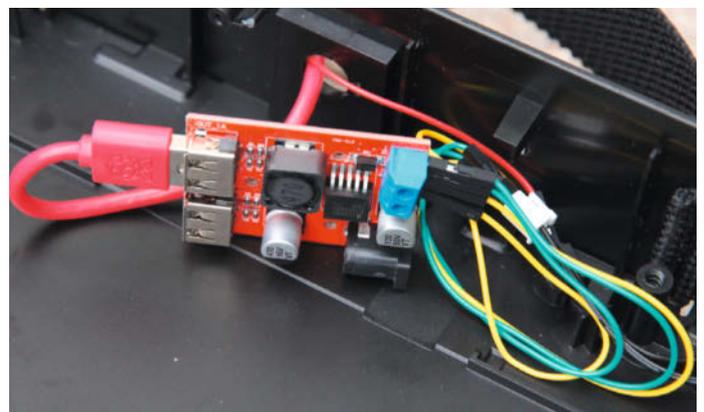
ausgegeben, an dem der Ventilator hängt. Anschließend setze ich die Variablen `CO2` und `IAQ`, welchen der jeweilige Sensorwert zugewiesen wird. Der CO₂-Wert wird dann auf der 7-Segment-Anzeige angezeigt, bevor die eigentlichen Abfragen zu den Grenzwerten ablaufen.

Nach dem Einschalten des BME680 kann es eine Weile dauern, bis die Luftqualität richtig berechnet ist. Der Sensor muss dafür unterschiedlich „schlechte“ Luft messen. Erst wenn die *IAQ Accuracy* über 1 beträgt, liegen die Werte korrekt vor. Für die Überprüfung in einer Größer-Gleich-Abfrage kann die *IAQ Accuracy* einfach im Drop-Down-Menü des Sensors ausgewählt werden. Solange die Werte noch ungenau sind, leuchtet eine RGB-LED blau auf und der serielle Monitor der Arduino IDE zeigt über *Serial print* den Hinweis *IAQ not ready!* Wer keine LEDs verbaut hat, kann den Baustein einfach aus dem Programm nehmen.

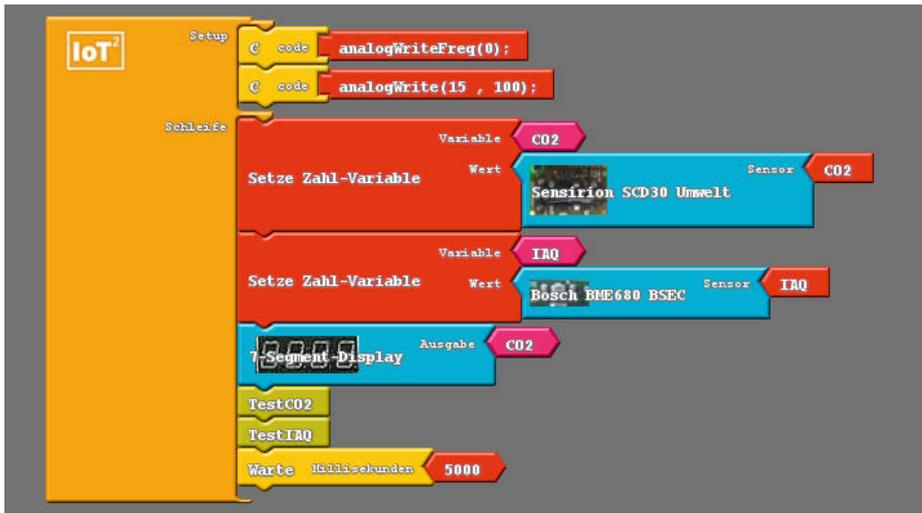
Danach überprüfe ich die vorher festgelegten IAQ-Grenzwerte von 100 und 150 in zwei verschachtelten *Falls-Sonst*-Abfragen. Unter 100



P2-Stecker, die Verbindung zum Ventilator – rot ist CLK, grün GND und gelb die Spannungsversorgung.



Der Spannungswandler und seine Verkabelung: das USB-Kabel und das Steuerkabel werden nach außen geführt.



Das Hauptprogramm

soll der Ventilator auf Stufe I laufen, unter 150 auf Stufe II und darüber hinaus auf Stufe III. Die Ansteuerung erfolgt wieder über analogWriteFreq() und analogWrite(15, 100), wobei nur noch die Ansteuerfrequenz eingetragen werden muss. Dazu habe ich jeweils noch eine Farbausgabe für die RGB-LEDs vorgesehen und lasse die Stufe bzw. einen Aufruf zum Querlüften auf dem seriellen Monitor anzeigen.

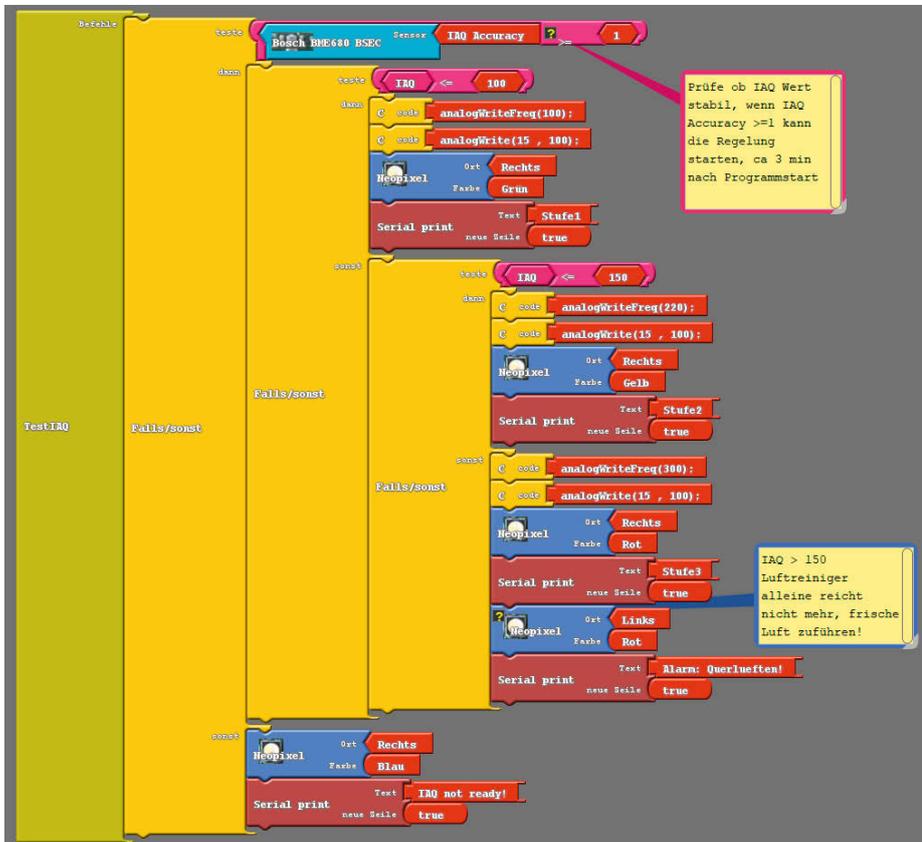
Das Unterprogramm *TestCO2* ist weitestgehend mit dem Code aus der Make 5/20 identisch – erweitert um die Anzeige auf dem seriellen Monitor. Wer keinen BME680 hat, könnte es entsprechend umbauen und zum Steuern des Ventilators nutzen. Das komplette Programm kann aus meinem Github-Repository runtergeladen werden (siehe Link in der Kurzinfor).

Der finale Test

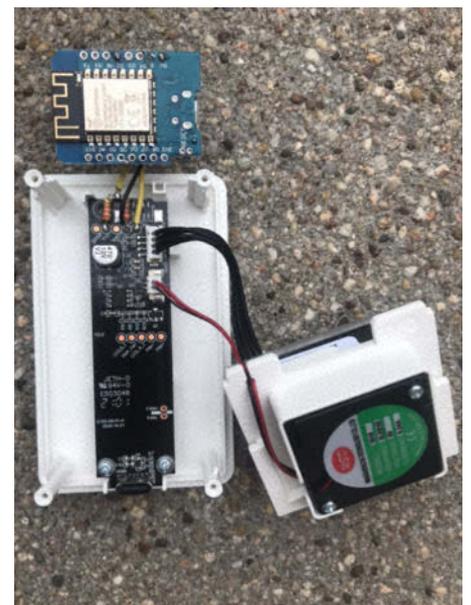
Um das Programm und die Steuerung zu testen, kann man auf den CO₂- oder VOC-Sensor hauchen. Dann sollte sich der Messwert und damit die Drehzahl erhöhen. Dann heißt es warten, bis der Filter die gewünschte Reinigungsleistung erbracht hat und die Drehzahl sich wieder reduziert. Nun haben wir einen kleinen Automaten im Internet der Dinge, der sich immer um saubere Luft kümmert. Wer möchte, kann ihn um einige Komfortfunktionen erweitern.

Da man den Filterwechsel nicht vergessen darf, könnte man noch einen Betriebsstundenzähler in das Programm einbauen. Zusätzlich könnte die Ventilator Drehzahl über das Tachosignal (FG) kontrolliert werden, das ebenfalls über den Stecker P2 läuft, aber in diesem Projekt nicht verwendet wurde. Die Kabelverbindung zwischen der Ampel und dem Luftreiniger könnte durch ein Mesh-WLAN ersetzt werden. Auch könnten so mehrere Lüfter von einer Ampel gesteuert werden. Zusätzlich bietet es sich an, die Sensordaten zum Beispiel in einem Grafana-Dashboard zu visualisieren und für Auswertungen zu nutzen.

Insbesondere in Bastelkellern, wo es gelegentlich mal staubig wird, kann ein Feinstaubsensor zum Einsatz kommen – die IoT-Werkstatt unterstützt dafür den Grove PM2.5 mit einem Ardublock. Wer bei Ikea bleiben möchte, kann mit Vindriktning deren Luftqualitätssensor nutzen. Auch hier gibt es einen passenden Hack, der nur einen weiteren Wemos D1 erfordert, wie ab Seite 14 beschrieben. Meine Variante bzw. der Quellcode dafür ist in meinem Github zu finden. —hch



Überprüfung des IAQ-Wertes



Kann ebenfalls gehackt werden: der Luftqualitätssensor Vindriktning von Ikea.

Poster-Beilage: Bastelboard-Übersicht und Smart-Home-Projekte aus Make

Allen gedruckten Heften liegt in dieser Make-Ausgabe ein Poster bei, als Geschenk an Sie, liebe Leserinnen und Leser! Für die eine Seite haben wir das Universum gängiger Bastelboards wie Arduino, Raspberry Pi, ESP & Co. in eine grafische Übersicht gebracht, inklusive Pinouts wichtiger Vertreter der einzelnen Familien. Die Idee dabei: So ein Poster kann nicht nur eine Wand schmücken,



sondern sich auch als schnelle Referenz nützlich machen, sodass man vom Lötplatz nur einmal zur Wand aufschauen muss, wenn man wieder vergessen hat, auf welchem Pin beim Teensy4.1 nochmal RX5 liegt ...

Die andere Seite des Posters zeigt eine wirklich smarte Wohnung, denn sie ist gespickt mit einem Großteil der Projekte fürs Smart Home und rund ums eigene Heim, die wir in den letzten Jahren in der Make beschrieben haben. Bei der Zusammenstellung der Projekte waren wir selbst erstaunt, wie viel sich da angesammelt hat. Auf dem Poster finden Sie zu jedem Projekt nicht nur die Angabe, in welchem Heft wir die genaue Beschreibung veröffentlicht haben, wir haben uns außerdem die Mühe gemacht, über Symbole und Kürzel die wichtigsten technischen Eckpunkte wie Funktechnik (WLAN, Bluetooth, LoRaWAN ...) und Plattform (Raspi, ESP, Arduino ...) aufzuführen, damit Sie schnell sehen können, welche Projekte sich mit wenig Aufwand in Ihre vielleicht schon laufende Smart-Home-Infrastruktur einfügen ließen.



Beide Seiten des Posters wurden von Rebecca Husemann (rehu) aus der Make-Redaktion gestaltet. Wer die Make rein digital als App oder im Browser liest, kann sich die PDFs der beiden Posterseiten herunterladen.

Jetzt interessiert uns natürlich, wie das Poster bei Ihnen ankommt: Gefällt es Ihnen? Haben Sie einen Platz gefunden, es aufzuhängen? Für welche Seite haben Sie sich entschieden? War das Poster eine gute Idee, würden Sie sich für die Zukunft weitere Poster wünschen, falls ja, zu welchen Themen? Vielleicht wollen Sie uns aber auch einfach nur ein Foto schicken, das zeigt, wie unsere Beilage jetzt Ihre Bastelecke, Werkstatt oder das FabLab schmückt und sich dort hoffentlich auch so nützlich macht, wie gedacht – darüber freuen wir uns natürlich auch! Schicken Sie Ihr Feedback einfach an mail@make-magazin.de. —pek

HANNOVER

ROBOTIK | Künstliche Intelligenz | Virtual Reality

ROBOTIK HAT EIN ZUHAUSE: DIE REGION HANNOVER.

SEI DABEI UND MACH DEN NÄCHSTEN ZUG!

www.wirtschaftsfoerderung-hannover.de/robotics-city

WIRTSCHAFTSFÖRDERUNG

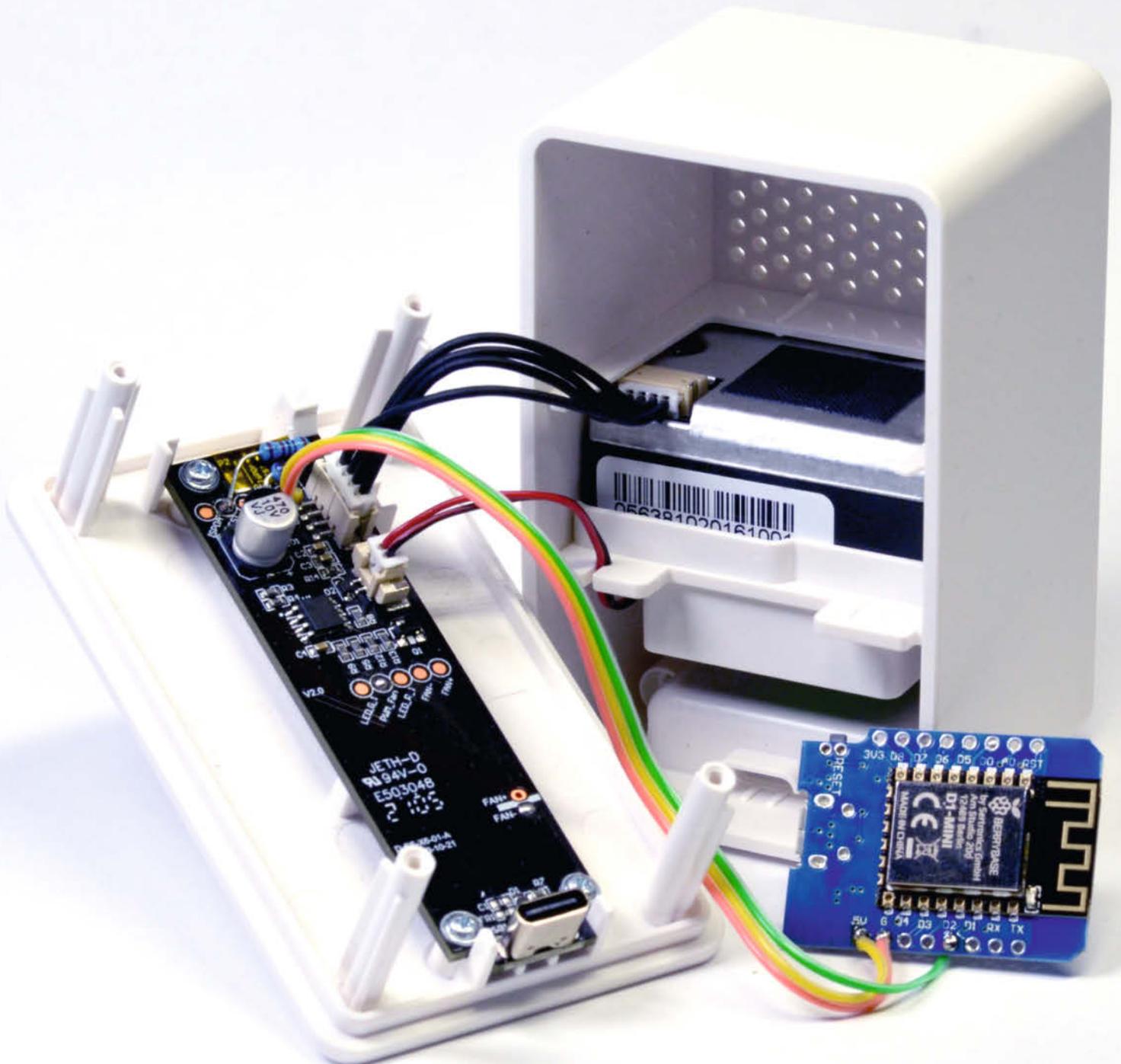
20 Jahre

Region Hannover

Ikea-Staubsensor mit WLAN aufbohren

Der günstige Feinstaubsensor *Vindriktning* von Ikea lädt zum Hacken ein: Mit einem ESP8266-Board und ein paar Lötverbindungen wird er zum echten IoT-Sensor, mit dem sich Staubbelastungen aufzeichnen und überwachen lassen.

von Carsten Wartmann



Der Einstieg in die Welt des IoT (Internet of Things) kann steinig sein. Um Ihnen die ersten Schritte zu erleichtern, benutzen wir gut zu beschaffende, günstige Hardware und ein einfaches, aber wohldurchdachtes System, um die Daten aufzubereiten. Die Lötarbeiten sind einfach und auch für Einsteiger durchführbar. Die Software ist für die Arduino IDE geschrieben, die sich für den Einstieg millionenfach bewährt hat. Heraus kommt ein Gerät, das aufgrund seiner Herkunft in jedem Wohnzimmer eine gute Figur macht, unter der Haube aber viel mehr kann, als der Hersteller vorgesehen hat.

Grundlage ist der mit unter zehn Euro erstaunlich günstige Staubsensor *Vindriktning* von Ikea. Hier bekommt man einen funktionalen Feinstaubsensor mit einer als Ampel gestalteten Anzeige der Staubbelastung in einem kompakten Gehäuse. Der Preis der einzelnen Komponenten auf dem freien Markt liegt hier schon über dem Ikea-Verkaufspreis. Zum Ausschlichten ist er aber zu schade, deshalb erweitern wir seine Möglichkeiten.

Bestandsaufnahme

Der Sensor von Ikea kommt ohne Netzadapter, hierfür ist eine USB-C-Buchse vorgesehen. Steckt man eine Stromversorgung ein, so blinken die LEDs erst einmal etwas, dann beginnt der Sensor zu messen. Probieren Sie doch einmal ein (altes) Kissen oder ein Stofftier vor dem Sensor auszuschütteln! Der Effekt sollte direkt sichtbar sein.

Geht man mit dem Ohr nahe an die Hinterseite des Sensors, so hört man den Lüfter. Hier scheint es auch eine Serienstreuung bei den Lautstärken zu geben. In einem ruhigen Raum hört man den Lüfter auch von weiter weg. Der Lüfter macht alle 20 Sekunden für 10 Sekunden eine Pause, was das Lüftergeräusch noch auffälliger macht. Warum diese Pause erfolgt, ist unklar. Das Datenblatt (siehe unten) erfordert diese jedenfalls nicht, eventuell soll dies Anlagerungen von Staub verhindern.

Ikea hat den *Vindriktning*-Sensor ¹ sehr hackfreundlich aufgebaut, er ist durch das Entfernen von vier Schrauben zu öffnen. Im Gehäuse befindet sich die Hauptplatine mit der USB-C-Buchse, dem Prozessor und den LEDs. Interessanterweise befindet sich auf der Platine auch eine IR-LED, die durch die LED-Abdeckung hinauschaud. Möglicherweise soll hiermit zukünftig, kabellos ein Luftfilter gesteuert werden.

Das Herz der Platine bildet ein Mikroprozessor von *Eastsoft*. Bisher sind mir aber noch keine Firmwareanpassungen oder Hacks bekannt, geschweige denn die Information, ob sich der *Eastsoft*-Mikroprozessor überhaupt mit einem neuen Programm flashen lässt. Die dafür erforderlichen Pins sind jedenfalls auf dem Board herausgeführt. Daneben existieren

Kurzinfo

- » Aufbau und Hardware des Sensors
- » Lauschangriff auf die Sensorkommunikation
- » Daten auf IoT-Dienst Adafruit IO speichern und visualisieren

Checkliste



Zeitaufwand:
eine Stunde



Kosten:
15 Euro

Material

- » Ikea *VINDRIKTNING* Luftqualitätssensor, PM2,5
- » Wemos D1 Mini ESP8266 Board
- » Widerstände je einen 1kΩ und 3kΩ
- » 30cm dünne Litze farbig isoliert

Mehr zum Thema

- » Uwe Rohne, WLAN-Alarmkontakt mit Batteriebetrieb, *Make* 4/21, S. 68
- » Carsten Wartmann, Internet-of-Things-Dienste für Maker, *Make* 3/21, S. 32
- » Daniel Bachfeld, Strom sparen bei ESP-Mikrocontrollern, *Make* 3/21, S. 84
- » Guido Burger, Dr. Richard Fix, Klaus-Uwe Gollmer, Der CO2-Warner für die Schule, *Make* 5/20, S. 10

Werkzeug

- » Schraubendreher, PH0
- » Lötkolben, Lötzinn, Schwamm
- » Seitenschneider
- » Abisolierwerkzeug

Alles zum Artikel im Web unter make-magazin.de/xbgw

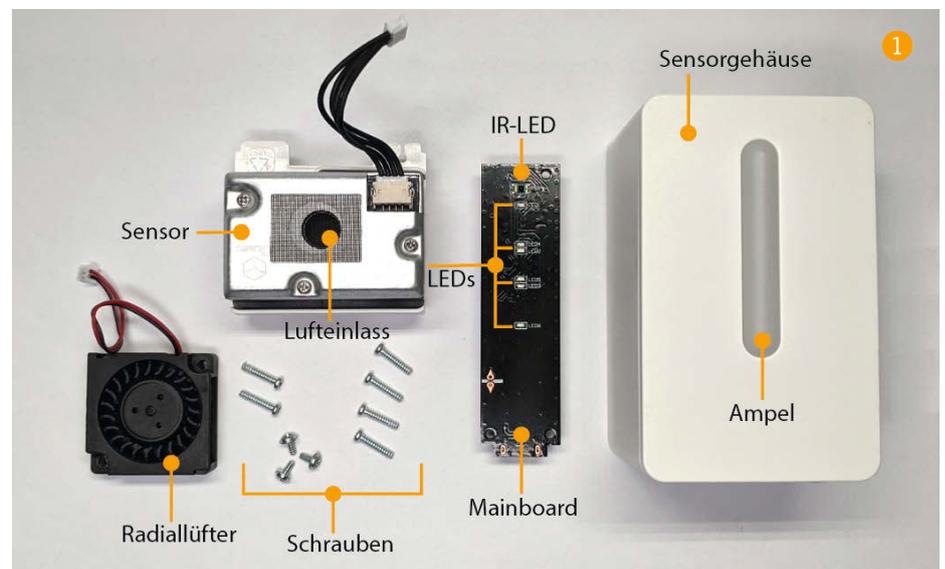
auch Pins, die dem neugierigen Maker den Status der LEDs und des internen Ventilators verraten. Die Hoffnung, dass die Fan- und Fan-PWM-Pins für einen externen Lüfter oder Filter gedacht sind, wird enttäuscht: Ich konnte bei meinen Experimenten weder ein PWM-Signal noch eine Schaltungsspannung finden, die nicht synchron mit dem pro 20 Sekunden einmal stoppenden internen Lüfter ist.

Das eigentliche Herz des Sensors ist ein Partikelsensor *Cubic PM1006* mit einem angeflanschten Radiallüfter. Dieser Lüfter saugt die Luft durch

die oberen Luftlöcher in den Sensor und bläst sie durch die unteren Lüftungslöcher wieder aus. Der Sensor (Datenblatt siehe Kurzinfo-Link) misst 2,5µm-Partikel von 0–1000µg/m³ mit einer Genauigkeit von ±20%. Der Sensor ist selbst schon mit einem Mikroprozessor ausgestattet und überträgt seine Messwerte seriell.

Lauschangriff

Am Stecker zum Main-Board findet man 5V, GND, und die seriellen Sende- (TX) und Empfangs-





schlüsse (RX). Praktischerweise werden diese Leitungen zu großen Löt- und Testpunkten auf dem Board geleitet. Dies ist ja schon mal eine Steilvorlage für uns.

Mit einem Oszilloskop an GND und REST (TX vom Sensor), kann man die serielle Kommunikation (9600 Baud, 8N1, 4,5V Pegel) des Partikelsensors ansehen ² und auf digitalen Oszilloskopen auch mitschneiden und dekodieren. Für den Screenshot habe ich den Sensor mit Löt-Rauch gequält und entsprechend hoch sind die PM2.5-Werte, die in den Bytes 5 und 6 (von Null gezählt) geliefert werden. Die

Ausgabe erfolgt hier in Hexadezimal-Ziffern. Byte 5 wird mit 255 (FF in Hexadezimal) multipliziert und dann Byte 6 addiert, um den Wert zu erhalten – hier 970µg/m³, knapp an der PM2.5-Messgrenze des Sensors.

Die ersten beiden Bytes geben dann auch genaue Auskunft, welcher Sensor hier aktiv ist: Laut Datenblatt ist es mit dieser Bytekombination 0x16 und 0x11 der Cubic PM1006. Die Modellvariante Cubic PM1006k liefert laut Datenblatt auch PM1.0- und PM10- Werte. Interessanterweise werden beim PM1006-Sensor aus der Ikea-Box diese Werte ebenfalls an entsprechender Stelle geliefert. Ob diese sich sinnvoll nutzen lassen, konnte ich aber mangels „Kalibrierstaub“ mit definierter Partikelgröße nicht herausbekommen.

ESP-Board anschließen

Würden wir nun das Mainboard des Vindriktning-Sensors durch ein ESP-Board ersetzen, könnten wir über die serielle Schnittstelle den Sensor mit ein paar Bytes dazu auffordern, uns die Messwerte zurückzusenden. Dabei würden wir aber auch die durchaus brauchbare „Ampelschaltung“ verlieren und wahrscheinlich auch das Gehäuse für einen neuen Stromanschluss modifizieren müssen.

Was aber auch funktioniert, ist, das ESP-Board einfach an der seriellen Schnittstelle lauschen zu lassen. Der User *Hyper* auf Github (Links in Kurzinfo) hat diesen Prozess ausführlich dokumentiert. Dies funktioniert auch gut, allerdings ist das hier verwendete ESP-Board *Wemos Mini D1* mit dem ESP8222 nicht für Signalpegel von über 3,6V (Signalpegel des Sensors sind 4,5V) spezifiziert, hier droht die Gefahr einer Beschädigung. Zum Glück lässt sich dieses Problem leicht mit einem Spannungswandler aus zwei Widerständen beheben. Die ganze Schaltung zwischen Mainboard und D1 Mini ist in Abbildung ³ zu sehen. Ich verwende je einen 3kΩ- und 1kΩ-

Widerstand. Rein rechnerisch wären andere Werte ideal, allerdings beeinflussen weitere Bauteile auf dem Ikea-Board den Spannungsteiler und sorgen damit für zu niedrige Pegel.

Beim Löten entfernt man am besten die Stecker von Sensor und Lüfter und nimmt die Platine nach Lösen der drei Schrauben heraus. So läuft man nicht Gefahr, mit dem LötKolben das Gehäuse oder Kabel zu schmelzen.

Um die beiden Platinen zu verbinden, kann man jede nicht zu dicke Litze benutzen. Am besten mit verschiedenfarbigen Isolierungen, dann ist die Gefahr, dass man eine Ader verwechselt, nicht so groß. 7–8cm reichen aus. Den Spannungsteiler kann man auf der Ikea-Platine oder auf dem D1-Mini installieren. Wichtig ist nur, dass der größere Widerstandswert an GND (Masse) angeschlossen wird, der kleinere an das TX-Signal (am REST-Lötpin). Der Pin D2 (GPIO 4) wird an die direkte Verbindung der beiden Widerstände angeschlossen. VCC auf dem Ikea-Board wird mit dem 5V-Pin auf dem ESP-Board verbunden, GND mit GND.

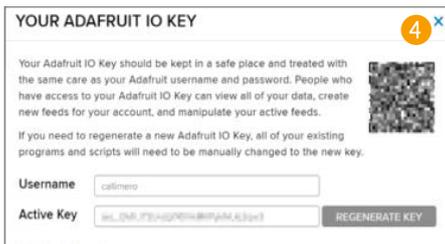
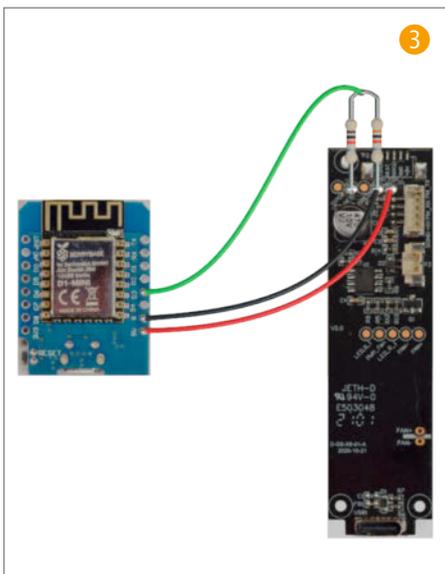
Meine anfängliche Idee, die USB-C-Buchse auf der Platine mit dem USB-Anschluss auf dem D1-Mini zu verbinden, musste leider begraben werden. Diese Buchse des Sensors hat keine für die Datenübertragung nötigen Pins und liefert nur den Strom.

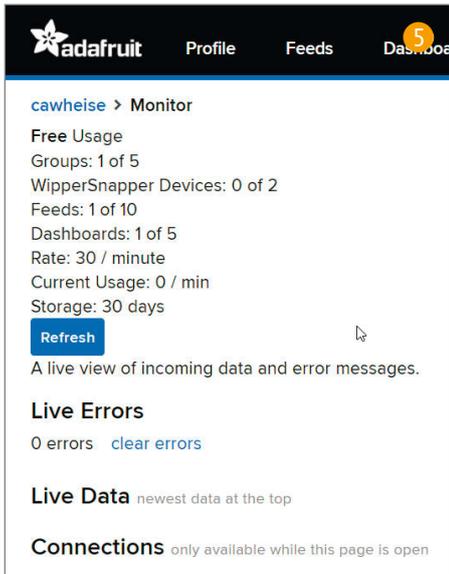
Adafruit IO

Um schnell zum Ziel zu gelangen, wird in diesem Artikel *Adafruit IO* (Links in Kurzinfo) verwendet. Adafruit IO stellt eine einfache Web-Umgebung zur Verfügung, um auf IoT-Geräte zu reagieren, Daten zu speichern und zu visualisieren. Adafruit IO ist als Lernsystem gedacht und beschränkt sich auf das Wesentliche, aber genau diese Beschränkung und die sehr gute Dokumentation machen es für erste Experimente so interessant. Benötigt man mehr, kann man auf das kostenpflichtige *IO+* wechseln, sucht sich einen anderen IoT-Provider (siehe Kurzinfo: Mehr zum Thema) oder richtet sich einen IoT-Server im heimischen Netz ein.

Die kostenlose Registrierung für Adafruit IO erfolgt über den Menüpunkt *Get Started for Free* und verläuft genau wie so viele Registrierungen im Web, aber sehr niederschwellig. Name, E-Mail und Passwort werden abgefragt und schon ist man registriert und kann loslegen. Man bekommt eine Willkommensmail, aber Newsletter & Co. sind nicht automatisch aktiviert.

Der gelbe Menüpunkt *Mykey*, rechts auf der IO-Webseite, zeigt Ihnen auf einen Blick und für Copy & Paste geeignet die Zugangsdaten zu Adafruit IO ⁴ an. Diese Werte müssen später in das Programm auf dem ESP-Board eingegeben werden, damit Adafruit IO dieses zuordnen kann und als berechtigt akzeptiert. Falls Sie einmal diesem Key nicht mehr trauen oder ihn versehentlich veröffentlicht haben, so





können Sie hier auch einen neuen generieren, müssen diesen dann allerdings wieder neu in die Konfiguration aller Boards eintragen. Die *Profile/Monitor*-Seite **5** zeigt Ihnen, ob und welche Geräte aktiv sind. Im Moment werden hier noch keine Daten angezeigt.

Arduino-Software

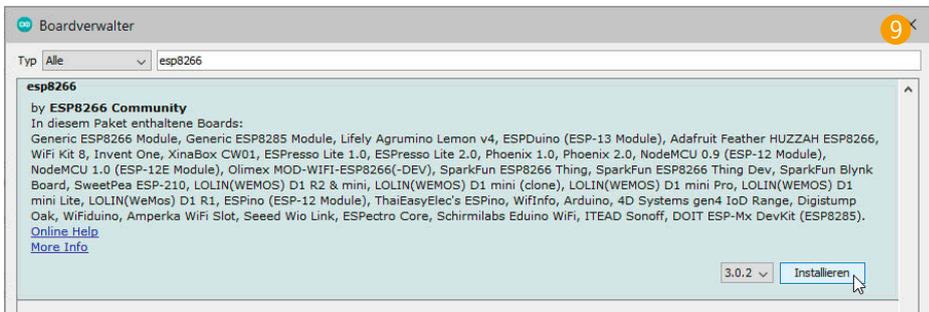
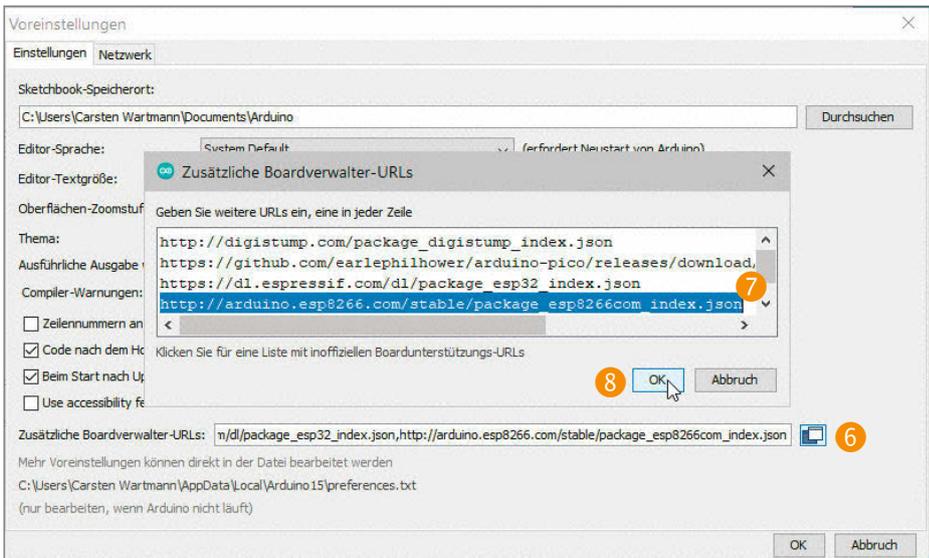
Verbinden Sie nun das D1-Mini-Board per Micro-USB-Kabel mit Ihrem Rechner, die Hardware-Erkennung sollte die Schnittstelle automatisch erkennen. Windows spielt dabei einen Ton ab, bei Linux können Sie per *dmesg*-Befehl nachschauen und beim Mac vertrauen sie einfach darauf, dass es erkannt wird.

Wenn Sie nun die Arduino IDE starten, sollte unter *Werkzeuge/Port/Serielle Ports* ein neuer Eintrag auftauchen. Wenn Sie noch nie ein ESP8266-Board programmiert haben, muss nun noch die Unterstützung dafür installiert werden.

Öffnen Sie in der Arduino IDE *Datei/Voreinstellungen*. Klicken Sie auf **6**. In dem neuen Fenster schreiben Sie in eine neue Zeile **7** die URL `http://arduino.esp8266.com/stable/package_esp8266com_index.json`, dann mit **OK** **8** bestätigen und auch die Voreinstellungen mit **OK** schließen.

Jetzt ist die IDE informiert, wo im Internet sie suchen soll, aber das Board muss trotzdem noch im Boardverwalter **9** installiert werden, den Sie über das Menü *Arduino IDE/Werkzeuge/Board/Boardverwalter* öffnen. Geben Sie dann in die Suche *ESP8266* ein. Ein Klick auf *Installieren* installiert das Paket.

Jetzt können Sie den D1 Mini im Board-Menü **10** auswählen. Kontrollieren Sie noch einmal, ob der richtige Port ausgewählt ist. Zur Sicherheit können Sie dann mit dem klassischen Blink-Programm aus Listing *Blink.ino* testen (zu finden auch unter *Datei/Beispiele/01. Basics/Blink*). Zum Übersetzen und Upload drü-



cken Sie *Strg-U* oder benutzen das Pfeil-Icon aus der Werkzeugeiste.

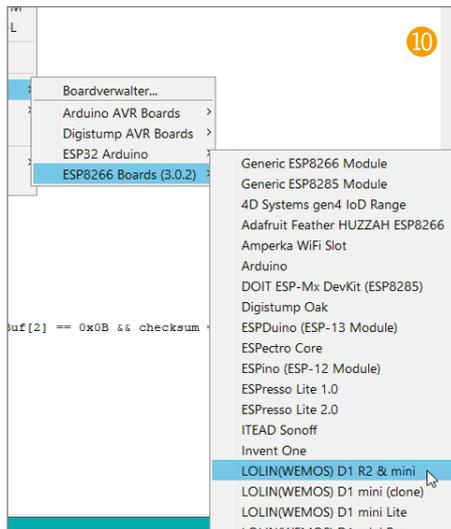
Das Programm wird nun kompiliert, also in für den Mikrocontroller lesbare Anweisungen übersetzt. Nach dem Upload, hierbei blitzt die LED immer wieder kurz auf, wird die LED am ESP-Board langsam blinken. Sollte es eine Fehlermeldung geben und die LED nicht blinken, so prüfen Sie, ob alle Schritte aus den vorherigen Abschnitten durchgeführt wurden. Andere ESP8266-Boards oder Klone des D1 Mini können eine andere Board-Einstellung erfordern.

Nun ist es an der Zeit das eigentliche Programm auf das Board zu bringen. Laden Sie *Particles2IO_2* von Github (Links in Kurzinfo). Eine Kurzerklärung zu Github gibt es dort auch. Über den grünen Download-Knopf können Sie ein ZIP-Archiv herunterladen und entpacken. In dem entstehenden Ordner befindet sich dann ein Unterordner mit dem Namen *Particles_2IO_2*, in dem sich das Hauptprogramm *Particles_2IO_2.ino* befindet. Dies können Sie in der Arduino IDE öffnen.

Für die Adafruit IO-Funktion benötigen Sie nun noch ein paar Libraries: Rufen Sie den Bibliotheksverwalter mit *Strg-Shift-L* auf und geben Sie in die Suche **11** *Adafruit IO Arduino* ein. Eventuell müssen Sie dann etwas scrollen, um den *Adafruit-IO-Arduino*-Eintrag zu finden. Klicken Sie dann dort den Button *Installieren* **12**

an. Es erscheint die Abfrage, ob Sie die Abhängigkeiten mitinstallieren möchten. Ich empfehle dies auf jeden Fall mit *Install All* zu erlauben, auch wenn ein paar Bibliotheken mit installiert werden, die sie eventuell nie brauchen werden. Dies ist doch sehr viel einfacher, als alle Abhängigkeiten per Hand zu installieren.

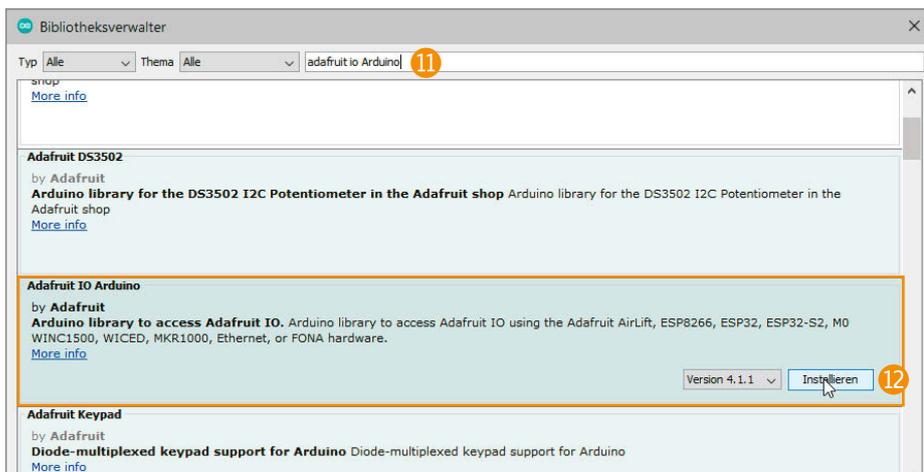
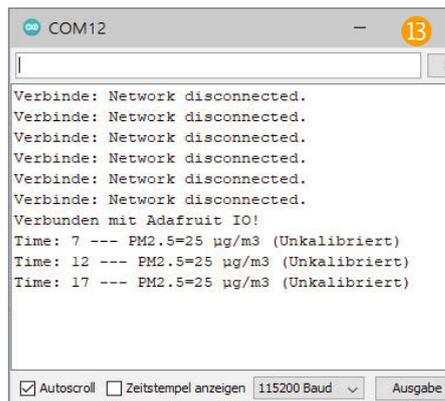
Jetzt müssen Sie unter dem Tab *config.h* die Zugangsdaten (SSID und Passwort) zu ihrem WLAN (2.4GHz) sowie den Usernamen und den Zugangskey von Adafruit IO eintragen.



Blink.ino

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the
  voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the
  voltage LOW
  delay(1000); // wait for a second
}
```



```
// Keys und User
#define IO_USERNAME "IOUsername"
#define IO_KEY "aio-key"
#define WIFI_SSID "WLANSSID"
#define WIFI_PASS "WLANPASS"
```

Damit Sie bei Adafruit IO einen eindeutigen Namen für den Daten-Feed bekommen, sollten Sie in der Zeile `AdafruitIO_Feed *pm25 = io.feed("TEST_PM25");` das `TEST_PM25` durch einen eigenen Namen ersetzen. Besonders wichtig ist dies, wenn Sie zwei oder mehr Sensoren verwenden.

Code Erläuterung

In der Hauptschleife `loop()` muss regelmäßig `io.run()` aufgerufen werden. Die Sensorabfrage wird aufgerufen, und wenn dort ein Wert vorhanden und eine Wartezeit von mindestens fünf Sekunden verstrichen ist, wird der Messwert an Adafruit IO gesendet. Wenn Sie mehrere Sensoren haben, sollten Sie im Adafruit-IO-Monitor (s. o.) prüfen, ob Adafruit IO sie ausbremst (Throttle) und dann die Wartezeit `send-interval` im Quellcode erhöhen.

Die eigentlichen Daten des Sensors werden in der `getSensorData()`-Funktion ausgelesen. Innerhalb der `while`-Schleife wird so lange von der seriellen Schnittstelle gelesen, bis 20 Bytes da sind.

Nun wird eine Prüfsumme über die empfangenen Bytes berechnet. Diese muss immer Null ergeben, ansonsten ist bei der Übertragung etwas schiefgegangen. Der Sensor hat dazu die 19 Daten-Bytes addiert und ein Byte hinzugefügt, sodass, wenn alles in einem Byte (0 bis 255) ohne Überlauf addiert wird, Null herauskommt.

Abschließend wird geprüft, ob die Kennung des Sensors in den ersten drei Bytes korrekt und die Checksumme null ist. Ist dies der Fall, wird aus den Bytes 5 und 6 der Messwert errechnet. Dazu wird Byte 5 mit 255 multipliziert und dann der Wert von Byte 6 dazu addiert. So bekommt man dann den Messwert als Ganzzahl im Bereich von 0-1000µg/m³ Partikeln.

Jetzt übersetzen (*Strg-U*) Sie den Quellcode und öffnen (Lupe-Icon in Werkzeugleiste) nach dem Upload einen seriellen Monitor. Stellen Sie sicher, dass die Geschwindigkeit auf *115200 Baud* eingestellt ist. Es sollte eine Ausgabe wie in 13 erscheinen. Das ESP-Board verbindet sich mit dem WLAN und dann mit Adafruit IO. Daraufhin startet die Messung und die Daten gehen hinaus.

In Adafruit IO gehen Sie nun zum Menüpunkt *Feeds*, hier sollte jetzt Ihr Daten-Strom aufgelistet sein. Durch einen Klick auf den Namen bekommen Sie schon mal eine Grafik 14 der Messwerte und eine tabellarische Auflistung der Daten. Schütteln Sie nochmal ein altes Kissen vor dem Sensor aus oder experimentieren Sie anderweitig mit dem Sensor.

Im Menü *Dashboards* und dann im Untermenü *view all* können Sie nun durch Klick auf *+New Dashboard* ein neues Dashboard erstellen. Vergeben Sie einen Namen und eine Beschreibung, danach wird das neue Dashboard aufgelistet. Durch einen Klick auf den Namen in der Liste können Sie es aufrufen. Im Moment ist es noch leer. Über den Knopf mit dem Zahnrad erhalten Sie ein Menü, in dem sie das Dashboard-Layout verändern, neue Blöcke hinzufügen und sonstige Einstellungen wie *Dark Mode* oder die Freigabe der Daten regeln können.

Zuerst sollten Sie *Create New Block* anklicken. Sie werden jetzt mittels eines Assistenten durch den Prozess geführt. In Schritt eins wählen Sie die Art des Blocks aus, es gibt Schalter, Regler, Test und Tabellen, Landkarten und für uns im Moment am interessantesten das *Line Chart*, ein Liniendiagramm. Nach Auswahl von *Line Chart* können Sie wählen, welche Feeds in diesem Diagramm erscheinen sollen. Wählen Sie ein oder mehrere Feeds aus der Liste und dann *Next step*. Im nächsten Schritt können Sie das Aussehen und die Beschriftung des Diagramms anpassen. Dies kann aber auch alles später noch jederzeit geändert werden. Mit *Create Block* erstellen Sie dann diesen Diagramm-Block und er erscheint auf dem Dashboard. Eventuell müssen Sie etwas warten, bis die ersten

Daten angezeigt werden oder benutzen Sie das Zahnrad-Menü im Dashboard Editor ¹⁵ um das Layout zu ändern und zu speichern. Die Blöcke editieren Sie ebenfalls mit dem kleinen Zahnrad-Icon im Block, hier lässt sich auch schnell die Zeitachse in Stunden und Tagen einstellen.

Adafruit hat wie gewohnt auch für IO gute Dokumentationen, aber mit etwas Experimentieren gelangen Sie dank der einfach gehaltenen Bedienung auch schnell zum Ziel und können die Möglichkeiten von IoT ausprobieren.

Erfahrungen

Bei uns Zuhause sind jetzt seit ein paar Tagen zwei Vindriktning-Sensoren mit ESP8266 in Betrieb. Neben den offensichtlichen Warnungen, wenn in der Wohnküche gekocht und gebraten wird, oder wenn ich im Arbeitszimmer löte, ergeben sich neue Erkenntnisse.

Ist es Nachts ruhig, so sinken die Staubbelastungen, weil sich der Staub setzen kann. Ich bin gespannt auf die Heizperiode und ob die Luftkonvektion die Staubbelastung erhöht. An den Werten erkennt man deutlich, wann wir als vierköpfige Familie aufstehen und durch hektisches Herumgerenne Staub aufwirbeln. Auch das morgendliche Lüften erhöht die Werte – sicher nicht, weil die Luft draußen schlechter ist, sondern weil Staub in der Wohnung hochwirbelt. An einem nebligen Tag aber warnte der Ikea Sensor neben der Terrassentür sogar in Ampel-Rot. Die Nebeltropfchen und Luftfeuchtigkeit werden vom Sensor als Staub erkannt.

Bei ¹⁶ sieht man um 20:00 (8:00PM) einen Peak in der Abendbrotzeit. Im Laufe der Nacht ¹⁷ sinkt dann der Staubgehalt, bis um etwa 7:00 ¹⁸ wieder Staub aufgewirbelt wird. Homeoffice ¹⁹ im Haus lässt einen recht konstanten Level entstehen. Nachmittags kurz vor vier Uhr ²⁰ habe ich den Ikea-Luftfilter (siehe auch Seite 8) im Arbeitszimmer einmal testweise auf die höchste Stufe angeschaltet, deutlich an der blauen Kurve zu erkennen.

Auch der Rauch von schwelendem Papier oder sonstigen Bränden wird erkannt und könnte zusammen mit einem Adafruit-IO-Trigger eine Meldung per Messenger auf das Handy senden, sozusagen die IoT-Vorstufe eines Rauchmelders.

Die absoluten Messwerte der Sensoren sind eher unzuverlässig. Ein weiterer unserer Sensoren zeigte in unteren Bereichen immer etwa 20µg/m³ weniger an als die anderen. Auch Vergleiche mit kalibrierten Sensoren in Internetforen bestätigen hier eine etwas ungenaue Messung. Da die Tendenzen in den Kurven gleich sind, wenn beide Sensoren nebeneinander stehen, kann man das ignorieren und die im Datenblatt angegebene Messgenauigkeit wird ja auch erreicht. Nicht zuletzt ist der Sensor nur mit einer Infrarot-

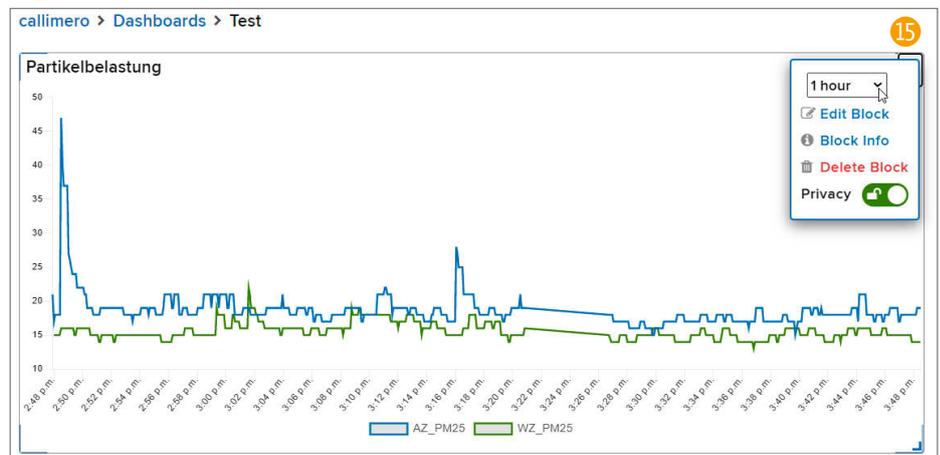


Mess-LED ausgestattet. Diese Messmethode hat, gegenüber einem Laser-bestückten Sensor, einen weniger guten Ruf, was Genauigkeit angeht.

Generell finde ich persönlich die Warnstufen (Ampel) des Sensors etwas zu hoch gewählt – wenn er anschlägt, ist die mangelnde Luftqualität schon recht offensichtlich. Was bleibt, ist ein interessanter, preiswerter, hackbarer Sensor, egal, ob man ihn modifiziert oder ausschaltet. Auch in Home-Automations-

Foren wird bereits heftig gebastelt und die erweiterten Ikea-Sensoren werden schon in den gängigen Systemen unterstützt.

Angesichts der Dinge, die man allein und eher unerwartet aus so einem Sensor auslesen kann, möchte ich Sie allerdings noch bitten zu überdenken, was Sie an Daten per IoT oder auf anderen Kanälen in das öffentliche Internet leiten. Verschlüsselung und Sicherheit hin oder her – im Zweifelsfall sind nur keine Daten sichere Daten im Internet. —caw



Photovoltaik an der E-Auto-Wallbox

Die KfW-Förderung für Wallboxen macht hauseigene Ladeterminals attraktiv wie nie. Wer zudem schon eine Photovoltaik-Anlage installiert hat, kann sein Auto kostenfrei mit Solarstrom laden. Voraussetzung ist allerdings eine intelligente Steuerung – wenn die fehlt, rüstet man einfach unsere ESP8266-Ladeautomatik nach.

von Uwe Rohne



Eine Photovoltaik-Anlage oder kurz PVA hat sich so mancher Eigenheim-Besitzer dank recht großzügiger Förderung und Vergütung bereits installiert, und viele besitzen auch schon ein E-Auto – das man praktischerweise an einer sogenannten *Wallbox* in der heimischen Garage lädt. Leider kümmern sich viele (vor allem ältere) Wallboxen nicht um die momentane Leistung einer schon vorhandenen PVA und laden das Auto stattdessen munter mit dem teuren Strom vom Energieversorger, wenn sich mal dicke Wolken vor die Solarzellen schieben.

Dabei ist das Prinzip des Überschusladens eigentlich ganz einfach zu implementieren: Wenn die Sonne scheint und die PVA kräftig Strom produziert, der ansonsten an den Energieversorger zurückgeliefert würde, wird das Laden eingeschaltet. Der Ladestrom für das E-Auto wird reduziert, wenn andere Verbraucher im Haus Strom abzapfen oder das Wetter etwas trüber wird. Er wird auf Wunsch ganz abgeschaltet, wenn die Solarzellen nichts Nennenswertes mehr liefern.

Das kann man natürlich auch manuell erledigen – doch wer will schon bei jedem Sonnenstrahl zum Auto laufen, um die Ladung zu starten? Eine automatische Steuerung muss den Ladestrom so regeln, dass bei noch nicht vollständig geladenem Akku möglichst alle überschüssige Energie der PVA als Ladestrom genutzt wird. Einziges Kriterium dafür ist der Stromfluss am Einspeisepunkt des Hauses gleich hinter der Hauptsicherung: Fließt Strom zum Energieversorger zurück, steht Energie für das Laden des Autos zur Verfügung.

Umschalten muss man dabei nichts; es genügt, den Ladestrom so zu regeln, dass der Energiefluss zum Versorger möglichst klein wird. In unserem Bauvorschlag ermittelt ein Energiezähler mit WLAN-Interface Strom, Spannung und Phasenlage am Übergabepunkt; ein schlichtes ESP8266-Modul liest den aktuellen Energiefluss aus und steuert anhand der Messwerte den Ladestrom der Wallbox, ebenfalls über WLAN. So einfach ist das!

Wallbox-Kriterien

Die aktuelle E-Autogeneration der Kompakt- und Mittelklasse, die bei den Händlern in den Showrooms steht, hat je nach Modell eine Akkuleistung von ca. 30 bis 70kWh. Dabei verbraucht ein E-Auto ca. 12-25kWh auf 100km Fahrstrecke. Wenn wir nun als typischen Vertreter der aktuellen Autogeneration den VW ID.3 mit Typ-2-Ladebuchse, mittelgroßer Batterie von 58kWh und einem angenommenen Verbrauch von 20kWh/100km nehmen, ergibt sich folgende Tabelle:

Die Aufladungen nach Zeilen 1 und 2 der Tabelle können noch mit einem Schukosteckerladegerät durchgeführt werden. Diese sind im Handel ab ca. 160 Euro erhältlich. Der

Kurzinfo

- » Aufgaben der Wallbox
- » Solarüberschuss mit Stromsensor ermitteln
- » Wallbox-Ladestrom per WLAN oder RS485 steuern

Checkliste



Zeitaufwand:
2 Wochenenden



Kosten:
130 Euro
(zuzüglich Wallbox-Installation)

Material

- » Drehstrom-Energiezähler Shelly 3EM (z.B. bei Reichelt)
- » WEMOS-D1-Modul mit ESP8266
- » SD- oder Micro-SD-Adapter Breakout-Board
- » Neopixel-LED 5mm bedrahtet
- » Kleine Lochrasterplatine
- » USB-Steckernetzteil

Mehr zum Thema

- » Filament-Kabeltrommel mit Leistungsmesser, Make 3/21, S. 46

Alles zum Artikel im Web unter make-magazin.de/xngu

zeitliche Aufwand für das Laden ist jedoch immens. Eine richtige Wallbox deckt dagegen alle Anwendungsfälle der Tabelle ab. Aber was sind nun die Aufgaben der Wallbox?

– Elektrische Sicherheit: Zusätzlich zum bereits in der Hausinstallation vorhandenen Fehlerstromschutzschalter (FI) vom Typ A, der nur Wechselstromfehler erkennt, ist in



Bei Sonnenschein kostenlos laden: Wallbox in der heimischen Garage.

Wallbox-Ladezeiten

Ladeleistung AC (Wechsel- / Drehstrom)	Ladezeit für 100km	Ladezeit von 20% auf 80%
Kleinste Ladung mit 6A bei 230V (1,4kW)	14,5h	25h
Ladung mit 10A bei 230V (2,3kW)	8,7h	15h
Ladung mit 16A bei 230V (3,7kW)	5,4h	9,4h
Kleinste Ladung Drehstrom 3x6A (4,1kW)	4,9h	8,5h
Größte Ladung Drehstrom 3x16A (11,0kW)	1,8h	3,2h

KfW-Förderung

Das Bundesverkehrsministerium hat ein Programm zur Förderung der Installation privater Ladepunkte initiiert. Die Kreditanstalt für Wiederaufbau (kfw.de) setzt dieses mit ihrem Programm 440 konkret mit nachfolgenden (vereinfachten) Eckpunkten um:

- » Ein privater Ladepunkt (Wallbox, Zubehör und Installation) wird mit 900 Euro gefördert. Es müssen Kosten von mindestens 900 Euro eingereicht werden. Wer also zu sparsam ist und seine Wallbox für z.B. 895Euro installieren lässt, erhält keine Förderung!
- » Ein Antrag muss online vor der Beschaffung gestellt werden (Dauer ca. 10 Minuten). Es wird dann eine E-Mail mit der För-

derzusage generiert. Die Förderzusage gilt für maximal 9 Monate.

- » Die Wallbox muss auf der Liste der förderfähigen Wallboxen stehen (Wallboxen ohne Kommunikationsschnittstelle und einer nicht auf 11kW reduzierbaren Leistung werden grundsätzlich nicht gefördert).
- » Man muss eine PV-Anlage und/oder einen Ökostromvertrag besitzen, planen oder beantragt haben.
- » Die Wallbox muss durch einen Elektrofachbetrieb installiert werden.
- » Man benötigt kein E-Auto! Sollte die Anschaffung erst in einigen Jahren geplant sein, wird die Installation trotzdem gefördert.

- der Wallbox ein FI-Schalter für Gleichstromfehler (Typ B) verbaut. Dieser schützt bei Versagen der Leistungselektronik im E-Auto vor der Spannung der Hochvoltbatterie.
- Aufladen, Schalten des Lastrelais: Das Auto signalisiert durch eine Reduzierung des Widerstandes auf der CP-Leitung **1** seine Ladebereitschaft. Die Wallbox schaltet daraufhin das Lastrelais ein. Bei Erreichen der gewünschten Batterieladung (SoC, State of Charge) erhöht das E-Auto den Widerstand auf der CP-Leitung wieder und das Lastrelais fällt ab. Wenn das Lastrelais angezogen hat, blockiert das E-Auto über eine Kerbe mechanisch den Ladestecker, so dass dieser nicht abgezogen werden kann.

- Regulierung Ladestrom: Durch das Verhältnis des durch die Wallbox generierten PWM-Signals (Pulsweitenmodulation) signalisiert die Wallbox dem E-Auto, mit welcher Stromstärke maximal geladen werden kann. Die Regelung, mit wieviel Strom des maximal möglichen tatsächlich geladen wird, passiert im E-Auto.
- Praktischer Ablauf des Ladens: Im einfachsten Fall wird die Ladung durch das Anstecken des Ladekabels gestartet. Es muss kein Knopf oder ähnliches gedrückt werden. Das Kabel wird während des Ladens gegen das Abziehen im E-Auto mechanisch verriegelt. Bei einfachen Wallboxen wird das Tastverhältnis der PWM und damit der maximale

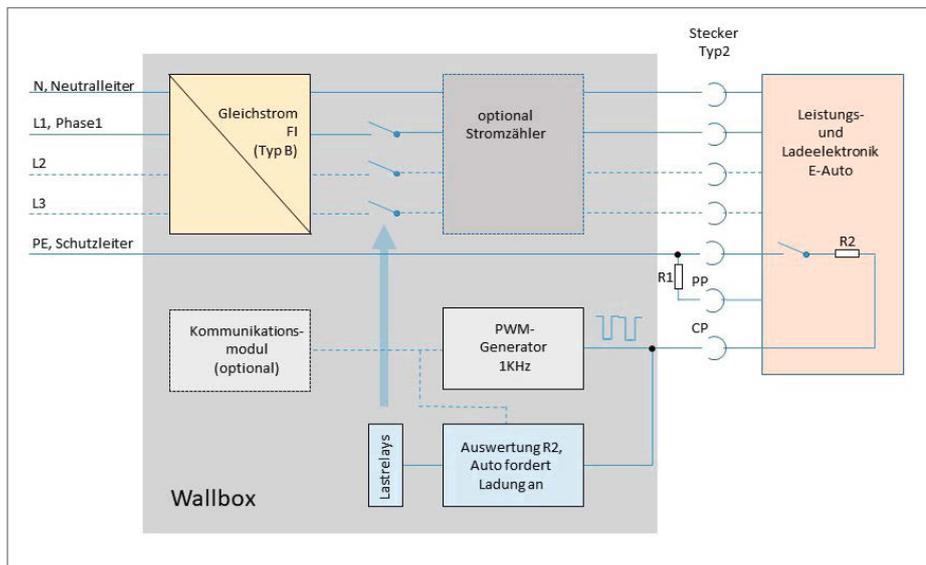
Ladestrom bei der Installation eingestellt und kann während des Betriebs nicht mehr verändert werden. Wallboxen mit einer Kommunikationsschnittstelle können das PWM-Signal während des Ladevorgangs verändern und so den Ladestrom regeln **1**. Außerdem kann über die Kommunikationsschnittstelle die aufgeladenen kWh und vieles mehr angezeigt werden. Der mögliche Ladestrom hat als Minimalwert 6 Ampere. Das bedeutet, dass die minimale Ladeleistung $6A \times 230V = 1380W$ beträgt. Die Größe des Widerstandes auf dem Kontakt „PP“ signalisiert dem E-Auto, bis zu welchem Ladestrom das Ladekabel verwendet werden darf **2**. So gibt es 16A- und 32A-Ladekabel zu kaufen.

Überschussladen

Kommt jetzt zu dem Projekt Wallbox eine bereits vorhandene oder geplante Photovoltaik-Anlage (PVA) hinzu, ist schnell der Wunsch nach einem PV-Überschussladen da: Statt Strom ins Netz einzuspeisen, soll doch lieber das Auto geladen werden. Fertige Lösungen sind unseres Wissens nicht unter 1500 Euro zu haben und setzen meist auch noch bestimmte Wechselrichter-Modelle der PVA voraus.

Unsere konkrete Lösung ist hier unabhängig: Wir setzen für die Steuerung den WLAN-fähigen Prozessor ESP8266 ein. Die jeweils aktuelle Einspeisung der PVA liefert das Energiemessgerät *Shelly 3EM*. Dieses ebenfalls WLAN-fähige Gerät arbeitet mit drei Stromzangen, über die es den Stromfluss der aktuellen Netzeinspeisung im Verteilerkasten erfasst.

Mit Netzeinspeisung ist hier die Leistung der PV abzüglich der aktuell laufenden Verbraucher im Haushalt gemeint. Wenn also z.B. die PVA 4kW liefert und im Haushalt 1kW verbraucht wird, werden 3kW Netzeinspeisung gemessen. Dieser Wert wird alle fünf Sekunden abgefragt und aus den jeweils letzten 60 Werten ein Mittelwert gebildet. Die Steuerung kennt drei verschiedene Betriebsmodi, die mit einem Taster eingestellt und über eine LED angezeigt werden. Die Steuerung arbeitet gemäß der Tabelle *Betriebsmodi Wallbox-Steuerung*.



1 Prinzipschaltung einer Wallbox

2 Ladestecker Typ2

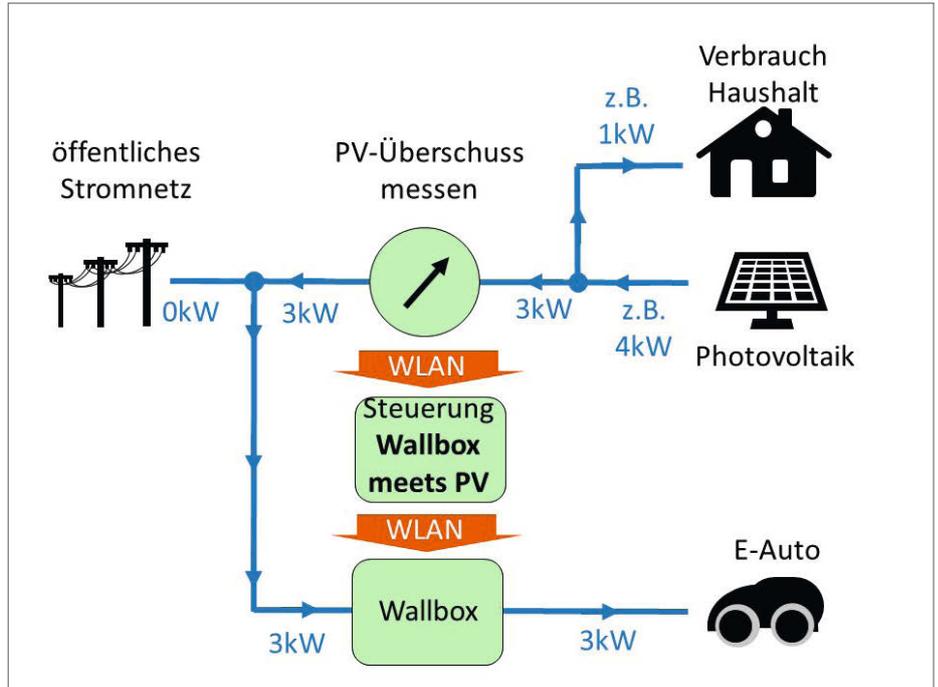


Derzeit unterstützt unsere Steuerung zwei namhafte Hersteller von Wallboxen: Den *go-eCharger* in der 11- und 22kW-Version sowie die *Heidelberg energy control*. Obwohl die Heidelberg-Box mit ihrem Edelstahlgehäuse robuster und besser verarbeitet ist, würden wir doch eher den *go-eCharger* empfehlen, weil er bereits mit einem WLAN-Modul und einer App ausgestattet ist.

Die Heidelberg-Wallbox ist nur leitungsgebunden über eine RS485-Modbus-Schnittstelle steuerbar, eine externe Steuerung muss über innenliegende DIP-Schalter konfiguriert werden. Das hat zudem den Nachteil, das bei Ausfall unserer Steuerung das Gerät geöffnet werden muss, um es wieder auf Stand-alone-Betrieb umzustellen.

Steuerungsautomatik

Bei Verwendung des *go-eCharger* sind dank vorgefertigter Module nur sehr wenige Bauteile notwendig, die auf einer Lochrasterplatine Platz finden und von jedem begabten Lötter verdrahtet werden können. Für die Micro-SD Karte kann, falls kein fertiges Breakout-Board zur Verfügung steht, auch gemäß Bild 5 ein Micro-SD-Adapter verwendet werden.



3 Übersicht Wallbox, Photovoltaik und die hier vorgestellte Steuerung: Stellt diese am Einspeisepunkt einen Energieüberschuss fest, wird das Laden mit dem verfügbaren Überschuss eingeschaltet. Leider ist der Ladestrom nur grob gestuft einstellbar; die Steuerung versucht aber, eine möglichst günstige Stufe zu wählen.

Heise Gruppe

JOB GESUCHT?

Ein gutes Team braucht viele verschiedene kluge und kreative Köpfe – und gleichzeitig den Freiraum, diese Potenziale zu entfalten und einzusetzen.

Gestalten Sie mit uns die Zukunft der Medienwelt!

Jetzt informieren und bewerben unter www.heise-gruppe.de/karriere.

Oelmann Elektronik GmbH

Die Oelmann Elektronik GmbH mit Sitz in Springe bietet zum nächst möglichen Termin ambitionierten Softwareentwicklern (m/w/d) ein anspruchsvolles Tätigkeitsfeld als

JUNIOR-ENTWICKLER (Software)

und

ENTWICKLER (Software) mit erster Führungserfahrung

Sie sind engagiert, experimentierfreudig und entwickeln effiziente und funktions sichere Software für den Einsatz in sicherheitskritischen Kommunikationssystemen? Entwicklungsumgebungen und Programmiersprachen wie C und C++ gehören zu Ihren täglich genutzten Entwicklertools? Dann senden Sie Ihre aussagefähigen Bewerbungsunterlagen mit Gehaltsvorstellung an Andreas.Scheller@Oelmann-Elektronik.de

www.Oelmann-Elektronik.de



Typische Größe einer PV-Anlage mit 9kW Spitzenleistung: Was nicht im Haus verbraucht wird, lädt nun das Auto.

Betriebsmodi Wallbox-Steuerung

Modus-LED	Ladung >10min. aus	Mittelwert 300s	Auto startet Ladung mit
Rot	egal	egal	16A
Gelb	nein	egal	0A
Gelb	ja	<3A	0A
Gelb	ja	>3A	PV-Überschuss 6 bis 16A
Grün	nein	egal	0A
Grün	ja	<6A	0A
Grün	ja	>6A	PV-Überschuss 6 bis 16A

Die SD-Karte enthält eine Konfigurationsdatei namens `config.txt` mit den Daten des WLANs und den IP-Adressen der Geräte (Energiezähler, eCharger) sowie einige „Schalter“ (Wallbox-Typ, einphasige/mehrphasige Umschaltung). Außerdem speichert sie Statistiken der Energiebilanz für jeden Tag. Neben den kompletten Projekt-Source

finden Sie eine vorbereitete `config.txt`-Datei wie üblich auf unserer Projektseite unter dem Link im Info-Kasten. Bei entsprechenden Programmierkenntnissen lässt sich das Projekt den eigenen Bedürfnissen noch weiter anpassen. Wer dieses nicht kann oder will, kann auch ein fertig übersetztes Image herunterladen (bin-Datei).

Firmware flashen

Um die (fertig kompiliert vorliegende) Firmware-Datei auf den Wemos zu flashen, sind ein Windows PC, ein USB-Kabel und eine Flash-Software notwendig. Sollte der PC noch keinen Treiber für den USB-Seriell-Wandler CH340G (dieser ist auf der Wemos D1 Mini Platine verbaut) haben, ist dieser zuerst zu installieren. Den Treiber findet man u.a. auf der Seite des Chipherstellers (siehe Link im Kurzinfo-Kasten). Als Software zum Flashen empfehlen wir den *ESPEasyflasher*, ebenfalls bei den Links zu finden.

Die `bin`-Datei von unserer Projektseite ebenfalls downloaden und im Verzeichnis des *ESPEasyflasher* speichern. Dann das Programm *FlashESP8266.exe* starten, den COM-Port des Wemos D1 mini auswählen (lässt sich im Zweifel über den Windows-Gerätemanager herausfinden) und die zu flashende `bin`-Datei ebenfalls auswählen (sie *muss* im selben Verzeichnis sein) und die Programmierung mit *Flash* starten. Es öffnet sich ein Fenster, in dem die Aktivitäten des Flashers gezeigt werden. Nach der Meldung *flash complete* ist die Steuerung für die Inbetriebnahme fertig.

Unsere Steuerung muss jetzt noch wissen, wie das WLAN heißt und mit welchem Passwort es gesichert ist. Außerdem muss die IP-Adresse des Energiemeters Shelly 3EM und die des go-eChargers eingetragen werden; dies geschieht über die bereits erwähnte `config.txt`-Datei auf der SD-Karte. Bei Verwendung der Heidelberg-Box darf die Zeile mit der (hier ignorierten) eCharger-Adresse nicht gelöscht werden. Die Software des ESP8266 liest nur die Werte, die zwischen den spitzen Klammern stehen, alles außerhalb der Klammern wird als Kommentar interpretiert.

Elektrik-Trick

Jetzt kommt der Elektriker zum Zug: Im Zähler-schrank haben Laien keine Änderungen vorzunehmen. Da der Elektroinstallateur ohnehin



4 Das sind die Zutaten für unseren Bauvorschlag: WEMOS-1-Modul, Micro-SD-Kartenadapter, Neopixel-LED und ein USB-Netzteil

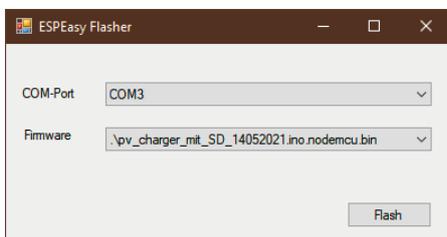


5 Micro-SD-Adapter als Breakout-Alternative: Bitte beim Anlöten der flexiblen Drähte aufpassen, dass es nicht zu heiß wird, sonst schmilzt das Plastik.

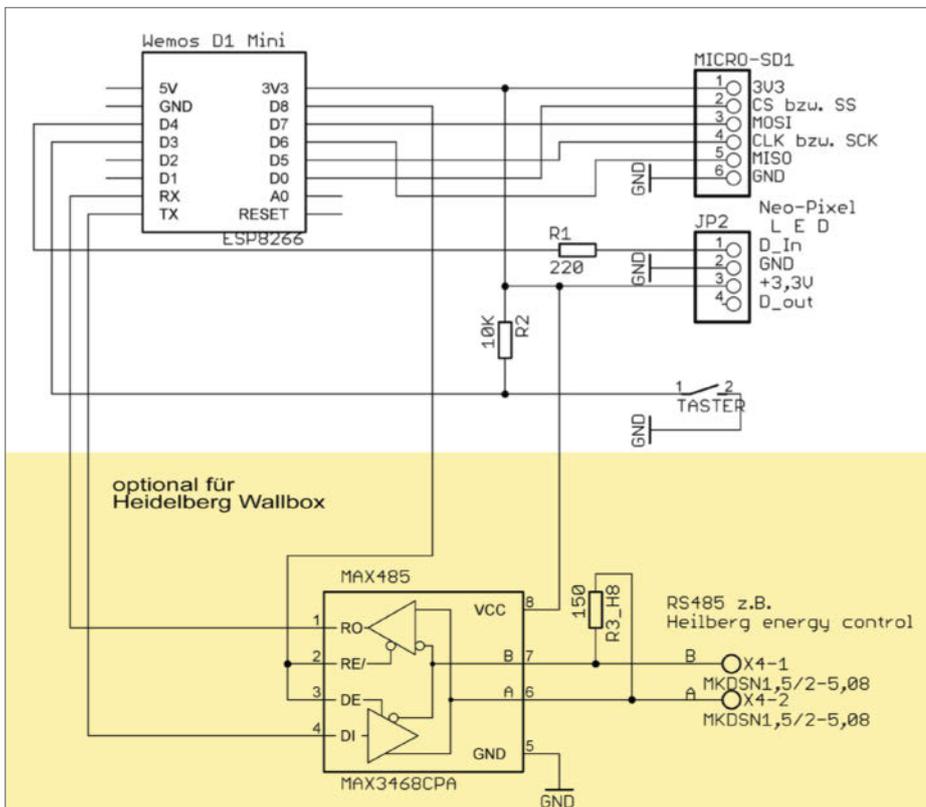
kommen muss, um die Wallbox anzuschließen (seine Rechnung wird für die KfW-Förderung benötigt), kann er den Energiemesser Shelly 3EM gleich mit installieren. Das Ganze sollte dann so wie auf Bild 8 aussehen.

Beim Einschalten leuchten bei der LED nacheinander mehrere Farben auf. Wenn alles durchläuft, zeigt die LED nach einigen Sekunden den Lademodus „rot“ an. Dieser kann nun bei Bedarf mit dem Taster weitergeschaltet werden (1 × drücken: gelb, 2 × drücken: grün, 3 × drücken: rot usw.) Die Farbe violett signalisiert einen Fehler: Entweder kann die SD-Karte nicht gelesen werden oder das Shelly-System und die Wallbox sind nicht erreichbar. In diesem Fall empfehlen wir folgende Vorgehensweise:

- Den go-eCharger-Status im Browser über seine IP-Adresse aufrufen (z.B. <http://192.168.178.123/status>).
- Den Status des Smartmeters Shelly 3EM (z.B. <http://192.168.178.124/status>) aufrufen.
- Die Steuerung *Wallbox_meets_PV* mit der IP-Adresse des ESP8266 aufrufen (11).
- Die WLAN-Signalstärken an den Einbau-Örtlichkeiten kontrollieren.
- Auch das Debuggen via Telnet auf Port 23 der ESP8266-Steuerung, etwa mit dem Tool PuTTY (9), kann aufschlussreich sein. Die Statistikdaten der Steuerung werden alle fünf Sekunden aktualisiert (10). Sollten die



7 ESPEasy Flasher in Aktion: Zur Auswahl stehen nur Dateien, die das Tool im eigenen Verzeichnis findet.



Schaltplan auf Basis des Wemos D1 Mini. Die Schaltung kann problemlos auf einer Lochrasterplatte aufgebaut werden. Der farbige hinterlegte Teil wird nur bei der Heidelberg-Wallbox benötigt.

Router sicher im Griff!



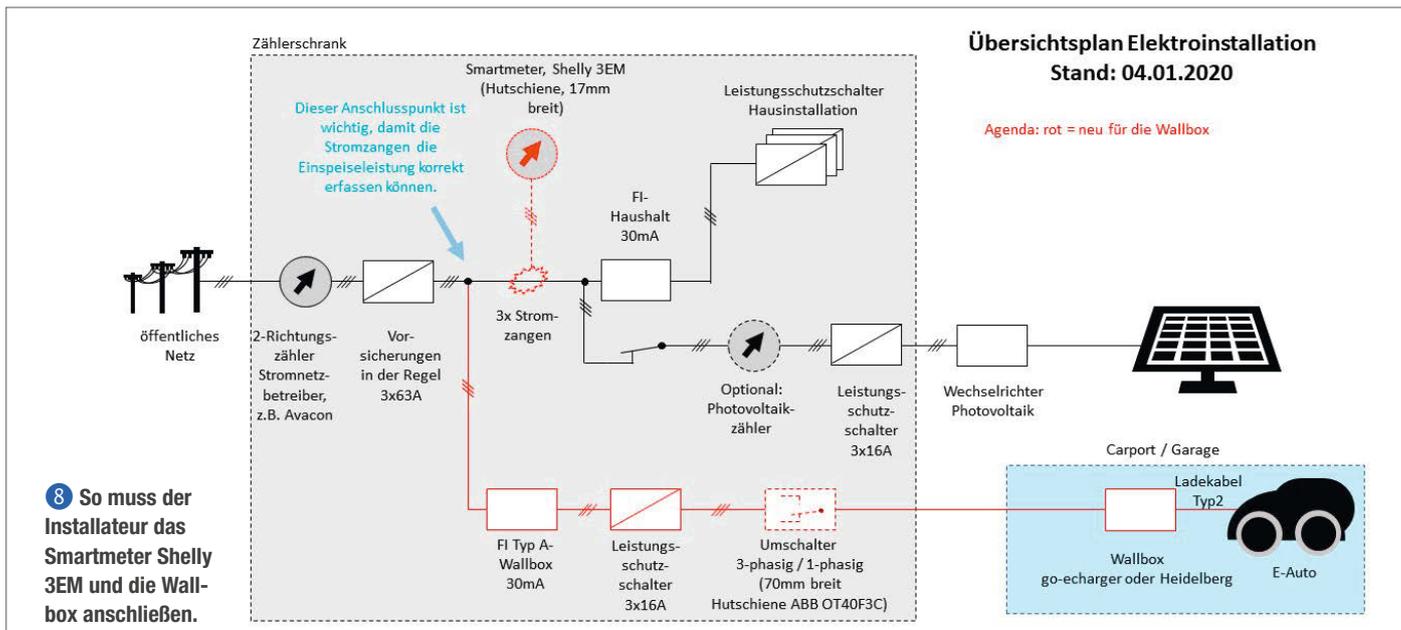
Heft + PDF mit 29% Rabatt

- ▶ Ihre Fritzbox absichern, erweitern, tunen
- ▶ Auch im Set mit Fachbuch erhältlich
- ▶ Über 170 Seiten konzentrierte c't-Expertise
- ▶ Für Abonnenten portofrei

Heft für 14,90 € • PDF für 12,99 € • im Set 29% günstiger!

shop.heise.de/ct-fritzbox21

Generell portofreie Lieferung für Heise Medien- oder Maker Media Zeitschriften-Abonnenten oder ab einem Einkaufswert von 20 €. Nur solange der Vorrat reicht. Preisänderungen vorbehalten.



IP-Adressen nicht bekannt sein (weil automatisch vom Router vergeben), muss man sie auf der Router-Konfigurationseite nachschlagen.

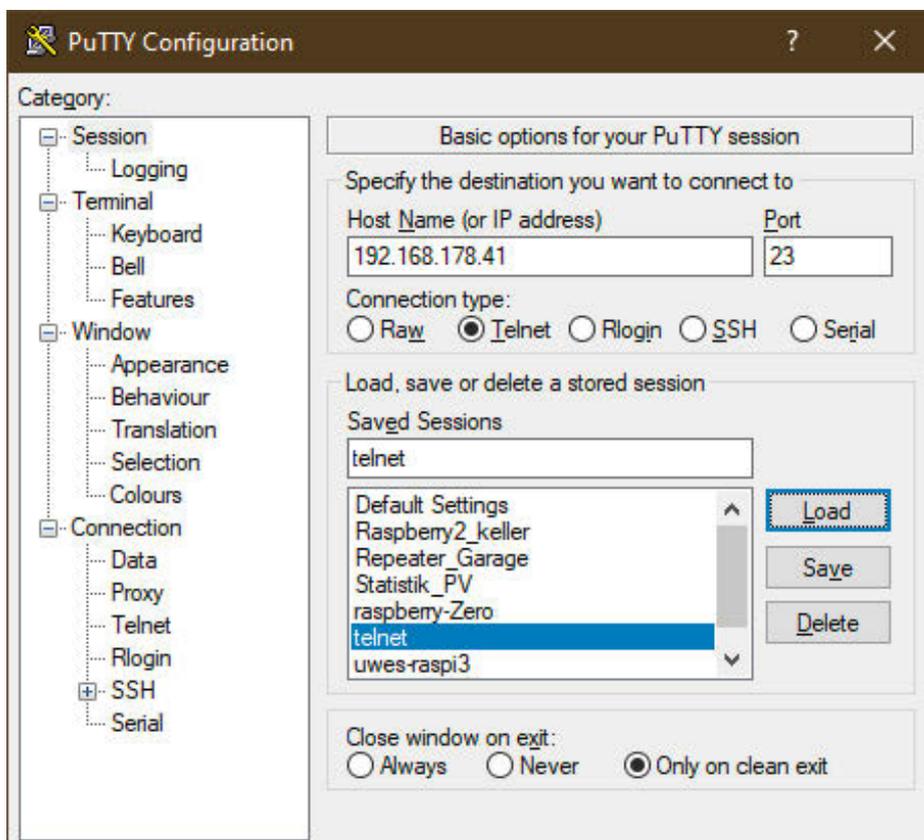
Web-Interface

Durch die Eingabe der vom Router vergebenen IP-Adresse unserer Steuerung in einem Browser wird ihre Web-Oberfläche **11** angezeigt.

Der OK-Button dient zum Umschalten der Betriebsmodi, der Refresh-Button zeigt den aktuellen PV-Überschuss und die Ladeleistung der Wallbox an. Die Funktion „Garagentor“ haben wir bisher noch nicht erwähnt: Durch Verwendung eines Optokopplers oder eines kleinen Relais parallel zum Garagentor-Taster kann das Tor ebenfalls über die Browseroberfläche gesteuert werden; dafür ist eine I/O-Lei-

tung des ESP8266 reserviert. Nähere Hinweise dazu finden Sie auf unserer Projektseite.

Wie schon der config.txt-Beschreibung zu entnehmen war, kann die Steuerung Statistiken generieren und, falls aktiviert, auf der SD-Karte ablegen. Es wird für jeden Tag eine eigene CSV-Datei erzeugt, die dann zum Beispiel mit Excel beliebig aufbereitet werden kann. Nach sieben Tagen wird eine Datei wieder gelöscht, so dass die SD-Karte nicht „überläuft“.



9 Starten von Telnet (hier mit der beispielhaften IP-Adresse 192.168.178.41) mit dem Tool PuTTY

Ein- oder Dreiphasig?

Aus gutem Grund haben wir diese Thematik bisher ausgespart: Wie wir anfangs gesehen haben, hat der Ladestrom einer Wallbox mindestens 6A. Diese können nun über eine Phase fließen, was dann ca. 1,4 kW entspricht, oder aber auch über drei Phasen, also $3 \times 1,4\text{kW}$ (4,2kW). Die vorhandene eCharger-Box kann mit sehr kleiner Leistung aber nur im einphasigen Betrieb laden, dreiphasig versorgt liefert sie immer mindestens 4,2kW.

Nun ist es leider so, dass sich eine Photovoltaikanlage darum nicht schert und die abgegebene Leistung je nach Wetter und Haushaltsverbrauch zwischen Null und Peak (z.B. 8kW) liegt. Mit dem Peak-Wert wird die Maximalleistung der PVA bei optimaler Sonneneinstrahlung bezeichnet. Soll jetzt das Optimum an Leistung für die Wallbox erreicht werden, kommt man um eine Umschaltung zwischen ein- und dreiphasigem Betrieb nicht herum, damit auch geringe Leistungen der PVA optimal genutzt werden können.

Unsere Steuerung sieht eine Option mit entsprechend gesteuerten Schützen (Leistungsrelais) in einer Platinen-Version vor, die wir online bereithalten. Steht eine Umschaltung an und sind mindestens 30 Minuten seit der letzten Umschaltung vergangen, wird die

Garagentor / Wallbox

GARAGENTOR

Ladeautomatik

PV-Ueberschuss: 7271W, WallBox: 3530W

Betriebsmodus

- 100% Netz (rt)
- min. 50%PV (ge)
- 100% PV (gn)

OK

Refresh

11 Der ESP8266 auf dem WEMOS-Board erzeugt ein einfaches Web-Interface mit Einstellung des Betriebsmodus und einer optionalen Garagentor-öffner-Funktion.

Wallbox komplett mit Hilfe eines Lastrelais mit Ruhekontakten stromlos geschaltet, dann werden über ein weiteres Lastrelais mit zwei Arbeitskontakten zu der einen ständig vorhanden Phase zwei Phasen dazu- oder abgeschaltet. Dieses Steuerungsverhalten kann ebenfalls in config.txt konfiguriert werden.

Ausblicke

Es ist anzunehmen, dass das Re-Design des go-eChargers ab Sommer 2021 die Umschaltung ein-/dreiphasig von sich aus können wird, so dass die Relaissteuerung für neu beschaffte go-eCharger nicht nötig ist. Die ESP8266-Firmware beherrscht diese Funktion in der Installation des Autors bereits.

```
192.168.178.32 - PuTTY
2021.08.22;10:04:11;12Asoll; 2840PV;2car; 12LS;13=; 7<;22>;2b;2770Wb;117Aist;0R;4288717s;391x
2021.08.22;10:04:16;11Asoll; 2720PV;2car; 11LS;13=; 7<;22>;2b;2770Wb;117Aist;0R;4288712s;386x
2021.08.22;10:04:21;11Asoll; 2618PV;2car; 11LS;13=; 7<;22>;2b;2580Wb;109Aist;0R;4288707s;381x
2021.08.22;10:04:26;11Asoll; 2540PV;2car; 11LS;13=; 7<;22>;2b;2560Wb;108Aist;0R;4288702s;376x
2021.08.22;10:04:31;10Asoll; 2489PV;2car; 10LS;13=; 7<;22>;2b;2560Wb;109Aist;0R;4288697s;371x
2021.08.22;10:04:36;10Asoll; 2440PV;2car; 10LS;13=; 7<;22>;2b;2330Wb;99Aist;0R;4288692s;366x
2021.08.22;10:04:41;10Asoll; 2384PV;2car; 10LS;13=; 7<;22>;2b;2330Wb;98Aist;0R;4288687s;361x
2021.08.22;10:04:46;10Asoll; 2361PV;2car; 10LS;13=; 7<;22>;2b;2350Wb;99Aist;0R;4288680s;354x
2021.08.22;10:04:51;10Asoll; 2339PV;2car; 10LS;13=; 7<;22>;2b;2350Wb;99Aist;0R;4288677s;351x
```

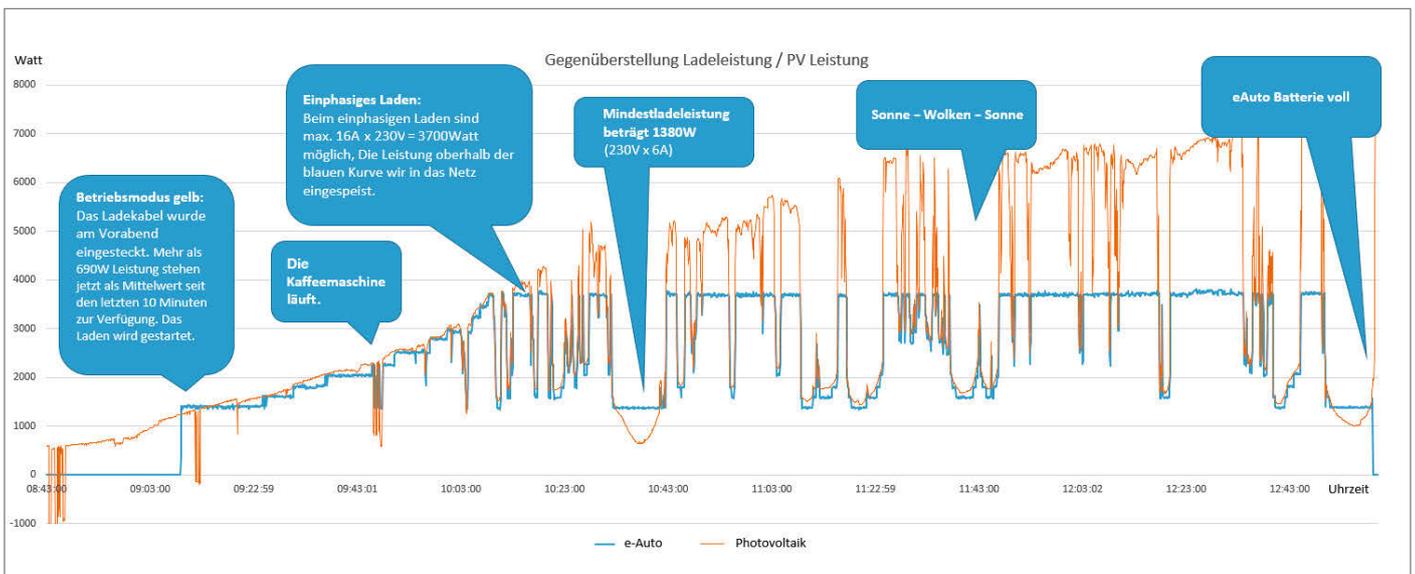
10 PuTTY-Telnet-Ausgabe von Zeit, Sollstrom, Leistung PV, Miw = Mittelwert, b = Betriebsmodus (0 = rot, 1 = gelb, 2 = grün), Ladestrom = Aist in 1/10A, x = 10-Minutentimer in Sekunden



Anschluss im Verteilerkasten: rechts der Smartmeter-Baustein Shelly 3EM, links oben am herausgezogenen Kabel einer der drei Stromsensoren des Smartmeters, ein Klappferrit mit Übertragerspule

Bei anderen als den genannten Wallboxen dürfte man allerdings um eine Anpassung der Firmware auf das dort verwendete Protokoll (API, MQTT/REST, JSON-Objekte) nicht herumkommen. Dafür wären dann schon fortgeschrittene Programmierkenntnisse erforderlich, vor allem, wenn das Protokoll nicht offenliegt und aus dem Web-Interface

der jeweiligen Box „re-engineered“ werden muss. Die API des go-eChargers dagegen liegt offen und ist gut dokumentiert. Und noch ein Hinweis: Möglicherweise widerspricht die hier skizzierte Lösung Vorgaben des Herstellers Ihres E-Autos. Ziehen Sie also unbedingt auch dessen Bedienungsanleitung zu Rate. —cm



Eine optimale Leistungsbilanz erreicht man mit einer Umschaltung von drei- auf einphasigen Betrieb, so dass auch geringere Leistungen gut ausgenutzt werden. Erst wenn die Leistung der PVA auf unter 1,4kW fällt, wird Strom vom Energieversorger bezogen.

Neuer Maker-space in Mannheim

Im Juli hat der Tech Hub des Mafnex-Technologie-zentrums im Bereich Ecke Windeck- und Julius-Hatry-Straße eröffnet. Der Makerspace des Hubs richtet sich an Studierende und Startups.

next-mannheim.de

Maker-Termine

KiCon Germany

8. Oktober

online

germany.kicon.dev

Jugend hackt Berlin

8. – 10. Oktober

Tüchtig – Raum für Inklusion

jugendhackt.org

Abschluss Open Health HACKademy #5

9. Oktober

online

matchymaker.de/open-health-hackademy

International Repair Day

16. Oktober

weltweite Events

openrepair.org/international-repair-day

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

Veranstalten Sie selbst?

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

mail@make-magazin.de



Good Lab / Andi Weiland

Ein Ort zum Tüfteln mitten in Berlin

Im Good Lab sollen nachhaltige Projekte entstehen

„Welchen ökologischen Fußabdruck hinterlasse ich?“, „Wie sieht ein nachhaltiger Stadtteil aus?“ oder „Wie wollen wir in Zukunft leben?“ Wie Technik bei den Antworten auf diese Fragen helfen kann, soll künftig im *Good Lab* erforscht werden. In dem neuen Fablab der *Junge Tüftler gGmbH* wird das Tüfteln an nachhaltigen Projekten im Mittelpunkt stehen. Am 9. Oktober ist es soweit: Das *Good Lab* öffnet im Herzen Berlins, im *Aufbauhaus* am Moritzplatz seine Türen.

„Wir glauben an die Kraft, Technologie für Gutes einzusetzen, um einen nachhaltigen Unterschied zu machen und gemeinsam mit allen eine Zukunft für alle zu gestalten. Dafür bringen wir Menschen zusammen und schaffen Formate des Austausches und des

Lernens“, so Dr. Julia Kleeberger, Co-Gründerin und Geschäftsführerin.

Geplant sind daher Schulprojekttage und Lehrkräftefortbildungen, eine wöchentliche offene Werkstatt, Workshops am Nachmittag und Ferienangebote. Derzeit ist das *Good Lab* am Dienstag und Mittwoch von 10 bis 18 Uhr und Donnerstag und Freitag von 10 bis 14 Uhr geöffnet. Für Projekttage kommt das *Good Lab* mobil auch in die Schulen. Neben Veranstaltungen in Berlin gibt es schließlich über die Lernplattform *Tüftel-Akademie* die Möglichkeit, Online-Workshops zu buchen.

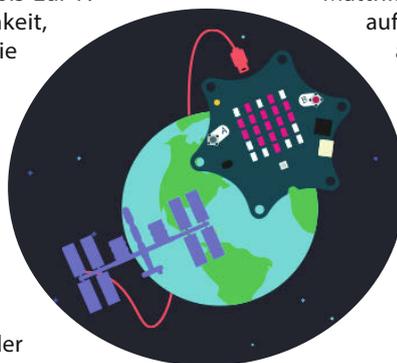
–hch

► good-lab.org

Programmier-Aktion für Schulen

Matthias Maurer nimmt den Mikrocontroller Calliope mit auf die ISS

Rund um den Mikrocontroller *Calliope mini* läuft mit *CalliopeO* eine Aktion zum Einstieg ins Programmieren. Schülerinnen und Schüler bis zur 7. Klasse haben dabei die Möglichkeit, selbstgeschriebenen Code auf die internationale Raumstation ISS zu schicken. Ziel ist es, kreative Programme zu entwickeln, die genauere Einblicke in die Umweltbedingungen auf der ISS ermöglichen: Wie warm ist es auf der Internationalen Raumstation? Haben die Astronautinnen und Astronauten dort oben Licht? In welche Richtung zeigt der *Calliope mini* gerade? Beantwortet werden diese Fragen direkt aus dem All.



Denn Ende Oktober soll eine SpaceX-Raumkapsel zur ISS fliegen. An Bord ist der deutsche Astronaut Matthias Maurer, der die Programme auf dem *Calliope mini* an Bord der ISS ausführen wird. Neben *CalliopeO* stehen bereits 35 Experimente auf seinem Missionsplan. Interessierte Schulklassen können sich bis zum 21. Dezember für die Teilnahme an der Aktion bewerben. Weitere Informationen gibt es online und in einer Online-Session für Lehrkräfte am 26. Oktober.

—Anna Eichler/hch

► calliope.cc/calliopeo

Bild: Calliope gGmbH

Repair Cafés leisten Fluthilfe

Ehrenamtliche reparieren Geräte in der Ahrregion

Küchenmaschinen, Akkuschauber, Stichsägen – die Flut im Ahrtal hat auch vor den Elektrogeräten nicht halt gemacht. Doch trotz Wasser und Schlamm sind viele Teile noch zu retten. In Walporzheim hat sich dafür die Gruppe *Elektroseelsorge* zusammengefunden. Am alten Sportplatz in der Ahruferstraße bieten die Freiwilligen täglich von 9 bis 18 Uhr kostenlose Reparaturen an. Mit Trockeneis befreien sie die Geräte vom Schlamm, um dann zu schauen, ob und wie sie wieder funktionstüchtig werden können. Auch jetzt ist der Bedarf an Unterstützung immer noch sehr groß. Die Gruppe freut sich daher über Interessierte, die mitarbeiten möchten. Wer Kenntnisse in Elektrik oder Elektronik hat und freie Zeit, kann gern vorbeikommen. Eigenes Werkzeug ist hilfreich, da es derzeit wenig Lagerraum gibt. Für die Unterkunft bei einem längeren Aufenthalt ist mit einem Schlafcontainer allerdings gesorgt.

Ebenfalls im Ahrtal unterwegs ist das *Repair Café* der Hochschule Koblenz, das von den VDE- und Amateurfunk-Hochschulgruppen organisiert wird. Zusammen mit zwei lokalen Repair Cafés wurden im August in Antweiler und Bad Neuenahr Elektrogeräte aller Art wieder in Schuss gebracht. Der Fokus lag dabei vor



Hochschule Koblenz/ Theisen

allem auf Werkzeugen – um damit wiederum die Reparatur weiterer Geräte zu ermöglichen. „Man kann zehnmal reparieren, aber nur einmal wegwerfen“, betont Prof. Johannes Stolz, Initiator der Aktion. Die direkte Erfolgsquote liege bei über 90 Prozent, nur wenige Geräte seien wirklich unreparierbar. So vermeide man weiteren Müll und könne mehr Geld in den Wiederaufbau – statt in Neukäufe – stecken. —hch

► hs-koblenz.de

► facebook.com/groups/867192947560286

Förderung für Open Hardware

Die Open Knowledge Foundation wird ab dem kommenden Jahr bis zu sechs Open-Hardware-Projekte fördern, die lokale Probleme im kleinstädtischen und ländlichen Raum lösen. okfn.de/blog

Umbau in München

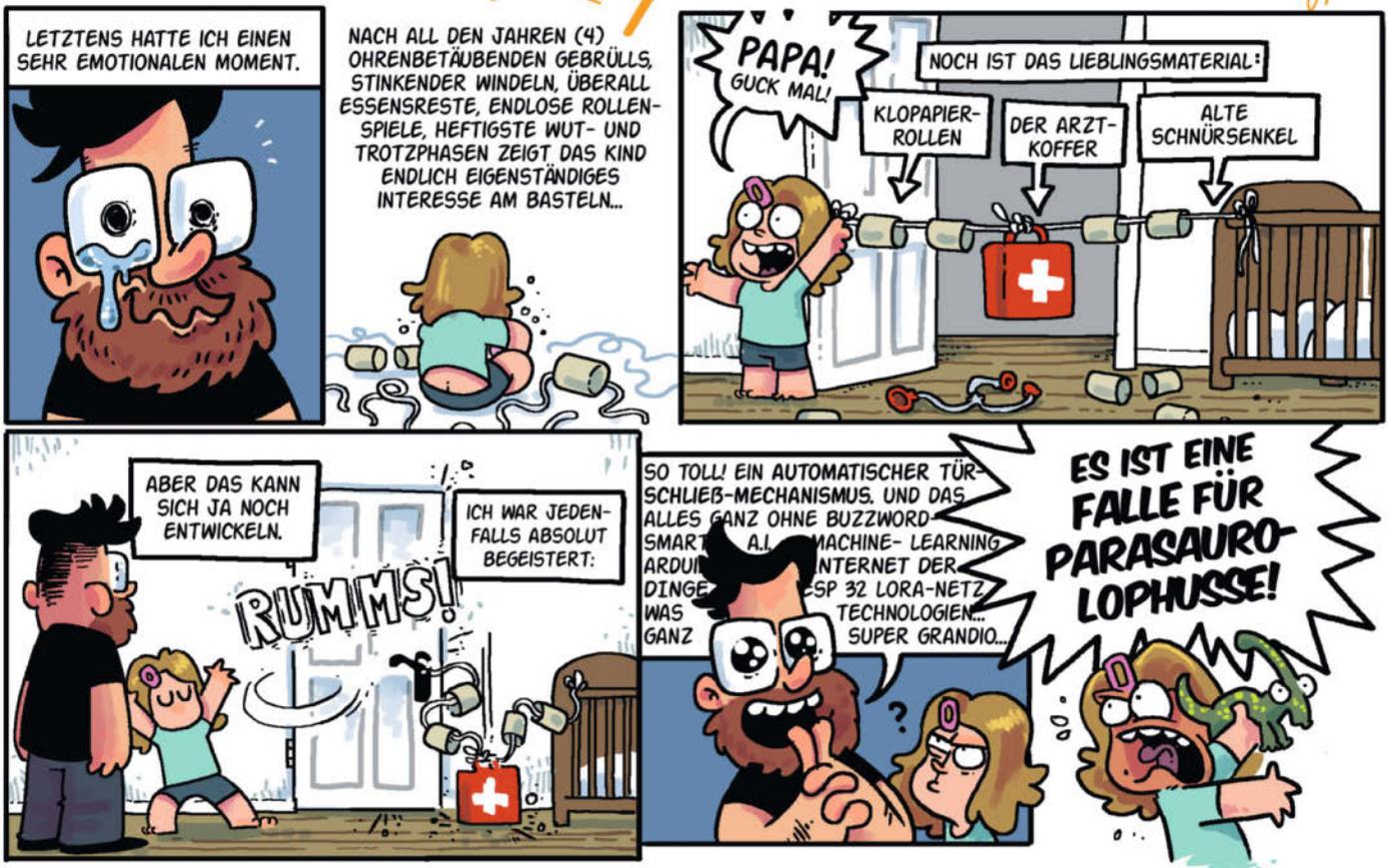
Im Haus der Eigenarbeit hat man in der Corona-Zeit die Werkstätten umgebaut sowie einen Repair-Café-Arbeitsplatz eingerichtet.

hei-muenchen.de

45 Vatergefühle

Kolophonium

von und mit @beetlebum



Die Welt als LED-Kugel

Im Rahmen der Abschlussarbeit des Autors entstand der Prototyp eines vollfarbigen sphärischen Displays. Gesteuert von einem Raspberry Pi trickst es das Auge mit dem POV-Effekt aus.

von Tim Kamenik



Mit LEDs lassen sich viele eindrucksvolle Projekte umsetzen. Mehrere hundert mehrfarbige Lichtpunkte lassen sich mit einem einfachen Mikrocontroller ansteuern und erlauben es, fantastische Skulpturen und Installationen zu erschaffen. Durch die einfache Verfügbarkeit von professionell gefertigten Elektronik-Modulen ist dies auch ohne großen Löt Aufwand für Maker möglich.

In meiner Abschlussarbeit an der Hochschule der Medien in Stuttgart habe ich mich mit einer solchen LED-Installation beschäftigt. Da ich im Laufe meines Studiums der audiovisuellen Medien immer wieder mit LEDs in unterschiedlichen Kontexten und Mengen in Berührung gekommen bin, habe ich eine Faszination für die Lichtpunkte entwickelt. Grundlegend wollte ich einen Prototypen für ein kugelförmiges Display entwickeln, das durch Rotation und eine schnelle Ansteuerung ein Bild anzeigt. Als Eingrenzung sollten dazu nur die typischen Werkzeuge eines Makers und leicht zu beschaffende Komponenten zum Einsatz kommen.

Das Prinzip des Displays ähnelt dem von USB-Ventilatoren, die mithilfe von LEDs die Uhrzeit anzeigen und ist unter dem Begriff *Persistence of Vision* (POV) bekannt. Für die betrachtende Person soll ein stehendes Bild erzeugt werden: Dazu werden auf den Rotoren (meist einfarbige) LED-Reihen angebracht. Die Ansteuerung geschieht abhängig von der aktuellen Drehgeschwindigkeit, aus der sich ergibt, welche LEDs wann aufleuchten müssen. Auf dieser Grundlage stellte ich mir die Frage: Was braucht es, um dieses Prinzip auf eine Kugel zu übertragen?

Der Teil, der als Erstes ins Auge fällt, ist der Rotor. Für die Kugelform ist der Hauptbestandteil ein Ring, der um eine vertikale Achse rotiert. Auf seiner Außenseite sind die bild erzeugenden LEDs angebracht. Um nicht auf einfarbige Bilder beschränkt zu sein, kommen RGB-LEDs zum Einsatz, also LEDs, die in einem Gehäuse drei unterschiedlich farbige LEDs kombinieren und durch Farbmischung eine große Auswahl an Farben erzeugen. Der ganze Aufbau ist auf einem Sockel montiert und wird über einen Riemen von einem Motor angetrieben.

Nachdem der erste grobe Plan stand, waren die entscheidenden Fragen klar: Wie müssen Ansteuerung und LEDs dimensioniert sein? Wie werden die LEDs auf dem Ring mit Strom und Daten versorgt? Und grundsätzlich: Wie kommt das Bild auf die Kugel?

Vorüberlegungen

Die erste Entscheidung besteht in der Größe des rotierenden Rings. Aufgrund des Bau raums meines 3D-Druckers und der Abstände der einzelnen LEDs auf Streifen mit 144 LEDs pro Meter beträgt der Außendurchmesser des

Kurzinfo

- » Kugelförmiges Rotationsdisplay
- » Prototyping mit 3D-Druck, LEDs und Raspberry Pi
- » Zeitkritische Ansteuerungen realisieren

Checkliste



Zeitaufwand:
10 Stunden, ohne 3D-Druck



Kosten:
ab 150 Euro

Werkzeug

- » FFF-3D-Drucker Bauraum min 210mm × 210mm × 200mm
- » Lötwerkzeug
- » übliches Maker-Werkzeug wie Zangen, Schraubendreher, ...
- » 2-K-Epoxy-Kleber
- » Sandpapier zur Nachbearbeitung der Drucke
- » Holzwerkzeug für den Sockel
- » Crimpwerkzeug

Mehr zum Thema

- » Lichtwischer-Kugel, Ulrich Schmerold, Make 3/16, S. 12
- » Der Zeitlupenrahmen, Ulrich Schmerold, Make 1/18, S. 58

Material

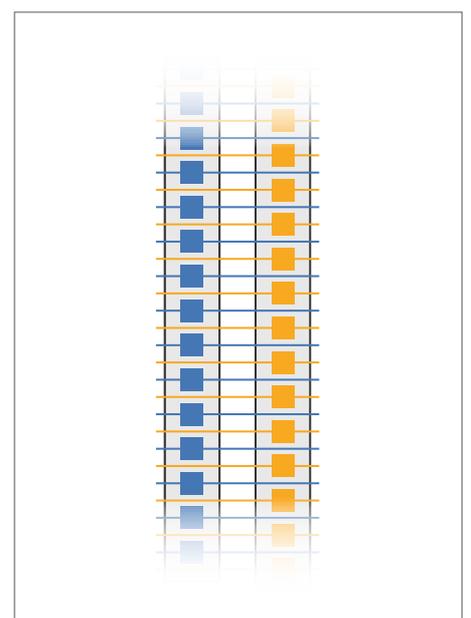
- » Raspberry Pi Zero W
- » Motor für Antrieb z. B. aus altem Tintenstrahldrucker, 4000 U/min
- » APA102C LED-Streifen 144 LEDs/m
- » Schleifkontakte Messingringe, (Feder-)Stahlblech, Federn
- » Zahnriemen GT2 260mm
- » Pulley GT2 20T
- » 3 Step-Down-Wandler MP1584EN oder LM2596
- » H-Brückenmodul LN298N
- » ATiny85 und entsprechender Programmierer
- » Poti und Schalter für die Motorsteuerung
- » Achse 8mm × 450mm
- » 3D-Druckteile Achsaufnahme, Schleifringhalterung, ...
- » Lager 8mm Kugel/Nadellager
- » Gehäusematerial MDF o. ä.
- » Netzteil 24V bzw. zum Motor passend
- » Pufferkondensator 1mF
- » div. Schrauben, Muttern und Gewindestange M3
- » div. Kabel
- » Wagoklemmen für einfachere Verkabelung
- » Dupont-Buchsen zum Aufcrimpen

Alles zum Artikel im Web unter make-magazin.de/x45h

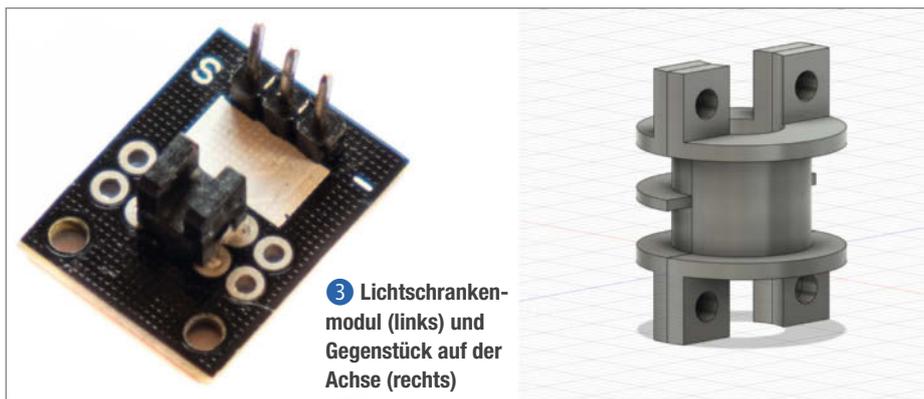
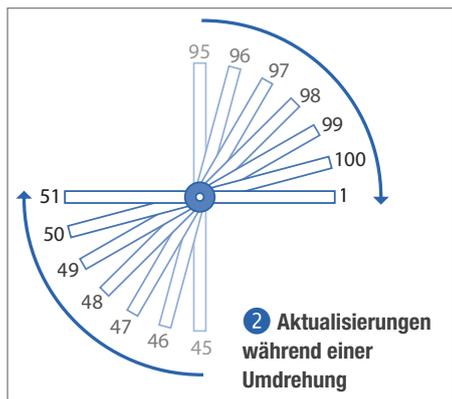
Rings etwa 27 cm. Daraus ergibt sich eine vertikale Auflösung von 56 LEDs pro Ringseite. Um diese Auflösung zu verdoppeln, kommt ein Halbbildverfahren (siehe Kasten) zum Einsatz. Letztendlich liegt die vertikale Auflösung dadurch bei 112 Pixeln.

Hinsichtlich der horizontalen Auflösung des Kugelbildes sind nicht die festgelegten mechanischen Rahmenbedingungen ausschlaggebend, sondern Software und Leistungsfähigkeit der Ansteuerung. Sie wird festgelegt durch die Anzahl an Aktualisierungen der LEDs innerhalb einer Umdrehung und bestimmt damit die Zahl der Spalten des Bildes. Die Wahl der horizontalen Auflösung in Verbindung mit der geplanten Rotationsgeschwindigkeit stellt wiederum Anforderungen an die steuernde Hardware.

Als Ziel habe ich mir eine horizontale Auflösung von 100 Spalten gesetzt. Die Drehfrequenz von 15 Umdrehungen pro Sekunde ist ein Kompromiss aus nötiger Drehzahl für eine halbwegs flackerfreie Darstellung, der Mecha-



1 Versatz der beiden Ringseiten



nik und dem verfügbaren Motor. Aus diesen Randbedingungen ergibt sich, dass alle LEDs auf dem Streifen 100 Mal pro Umdrehung aktualisiert werden müssen. Im Betrieb sind das 1500 Aktualisierungen aller LEDs pro Sekunde, was einer Frequenz von 1,5kHz entspricht.

Als letzte Vorüberlegung ist noch zu klären, wo die Ansteuerung und damit das „Gehirn“ des Displays platziert sein soll. Eine Möglichkeit wäre, die Ansteuerung im Sockel unterzubringen und die Daten für die LEDs zusätzlich zur Stromversorgung auf den Rotor zu

bringen. Das Problem dabei ist, dass Lösungen, die sowohl eine angemessen hohe Datenrate, als auch die benötigte Drehfrequenzen stabil abdecken, im industriellen Bereich liegen und keine einfach zu beschaffenden Komponenten sind.

Daher bietet sich eine zweite Lösung an, bei der die Ansteuerung im Rotor verbaut ist und sich mit ihm bewegt. Dadurch muss nur Strom auf den Rotor gebracht werden. Die Bilddaten sollen entweder im Stillstand auf die Ansteuerung aufgespielt und dort gespeichert werden oder aber über eine drahtlose Verbindung übertragen werden.

Halbbildverfahren

Das Halbbildverfahren dient der Verdopplung der vertikalen Auflösung, also der Anzahl der Zeilen. Dabei werden die LED-Streifen auf den beiden Ringseiten nicht exakt gleich angebracht, sondern um den halben Abstand der LEDs auf den Streifen versetzt 1. Betrachtet man während der Drehung 2 die Zeilen, so liegen die LEDs der beiden Ringseiten nicht auf gleicher Höhe, sondern die Zeilen der zweiten Ringseite liegen genau zwischen denen der ersten. Ringseite A zeichnet also die ungeraden Zeilen, bzw. das obere Halbbild, während Ringseite B die geraden Zeilen bzw. das untere Halbbild darstellt. Dieses Halbbild-Verfahren (*interlace*)

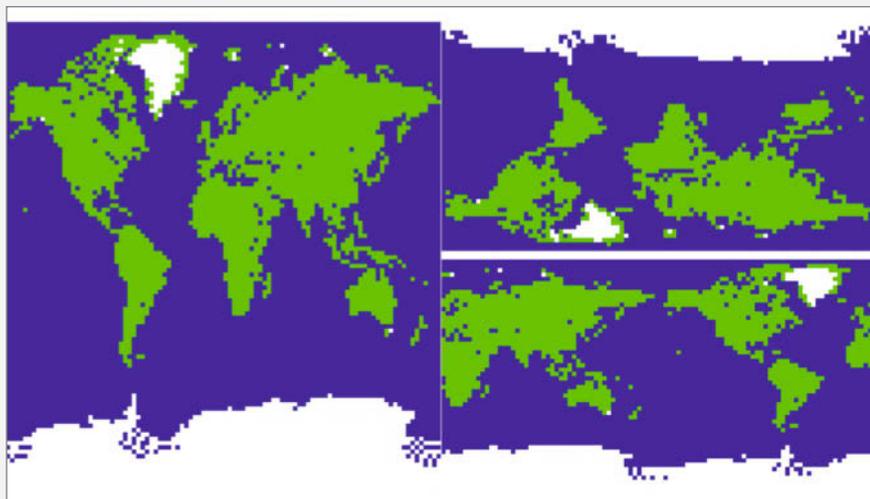
wurde auch in alten TV-Standards eingesetzt, um Übertragungsbandbreite zu sparen.

Das Verfahren führt dazu, dass eine komplette Umdrehung nötig ist, bevor das ganze Bild dargestellt ist, statt nur einer halben, wenn die LEDs gleich angeordnet wären. Auf Kosten der Bildwiederholfrequenz wird so eine höhere Auflösung erreicht. Da die Daten unten an Ringseite A eingespeist werden, läuft die Nummerierung der LEDs dort den Ring entlang nach oben und auf Ringseite B von oben nach unten. Daher steht das eine Halbbild in den vorbereiteten Bildern auf dem Kopf.

Auswahl der elektronischen Komponenten

Wie bereits in den Vorüberlegungen festgestellt, müssen die LED-Streifen mit einer Frequenz von 1,5kHz aktualisiert werden. Bei einer Anzahl von 112 LEDs mit je einem Byte für den Rot-, Grün- und Blaukanal ergibt sich eine Datenrate von ca. 504kByte/s. Dies entspricht etwa einer Bit-Frequenz von knapp 4Mbit/s. Die weitverbreiteten WS2812-LED-Streifen mit einer Datenleitung takten laut Datenblatt allerdings nur mit einer Frequenz von 800kHz. Damit sind sie deutlich zu langsam. Stattdessen sind APA102C-LED-Streifen verbaut, die im Gegensatz zu den WS2812 neben einer Datenleitung auch eine Taktleitung besitzen. Dadurch können diese LED-Streifen deutlich schneller angesteuert werden. Die maximale Taktfrequenz der APA102C-LED-Streifen liegt laut Datenblatt bei 24MHz und bietet damit genug Datendurchsatz für das Rotationsdisplay. Das gilt auch noch unter dem Aspekt, dass das Protokoll der APA102C neben den drei Farb-Bytes ein weiteres Byte erwartet, das die Helligkeit bestimmt und in diesem Fall immer auf das Maximum gesetzt wird.

Als Steuerrechner wird ein Raspberry Pi Zero W eingesetzt. Durch seine geringe Größe und Gewicht ist er ideal geeignet, um in dem Rotor eingebaut zu werden. Außerdem ist die WLAN-Fähigkeit ein weiterer Vorteil, sowohl für die Übertragung der Software als auch der Bild-



Ausgangsbild (links) und Bild mit für die Halbbilder angepasster Pixelordnung (rechts)

inhalte. Der *Raspberry Pi Zero W* besitzt zudem ein *Serial Peripheral Interface (SPI)*, das die Kommunikation mit den LEDs über Daten- und Taktleitung übernimmt.

Für die Drehzahlmessung ist ein Lichtschrankenmodul **3** verbaut, das über einen einfachen digitalen Eingang am Raspberry Pi mit der Software kommuniziert. Das Lichtschrankenmodul bewegt sich mit dem Rotor, wobei das Gegenstück auf der Achse montiert ist und die Lichtschranke jeweils eine halbe Umdrehung verdunkelt.

Spannungsversorgung

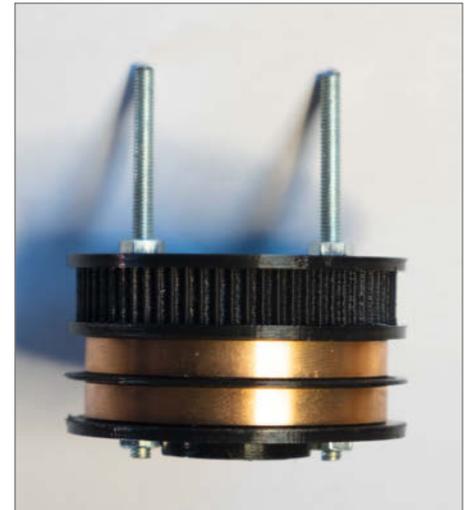
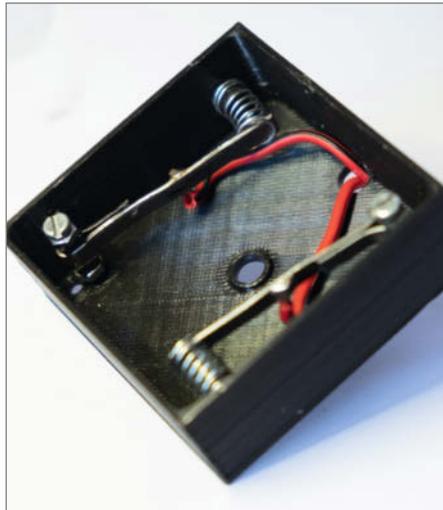
Um die elektronischen Komponenten der Ansteuerung und auch die LED-Streifen im Ring mit Strom zu versorgen, ist eine Lösung notwendig, die sowohl mit den zu erwartenden Strömen als auch mit der Drehgeschwindigkeit funktioniert. Eine Versorgung per Akku, wie beispielsweise einer Powerbank, ist schwierig, da das Gewicht schnell zu einer Unwucht führen kann. Deswegen fiel die Wahl auf eine Schleifkontaktierung **4** mit zwei Kontakten. Der statische Teil sitzt auf der Oberseite des Sockels. Jeder Kontakt besteht aus einem schmalen Stahlblech als leitendes Element, sowie einer Feder, die das Blech gegen den rotierenden Teil des Schleiferkontakts presst. Als Gegenstück im Rotor dient ein Messingring pro Kontakt, der unterhalb des Rings montiert ist. Das Stahlblech der statischen Hälfte läuft während der Drehung auf der Außenseite des Messingringes. Über die Zeit hat sich gezeigt, dass diese Kontaktierung bei 15 Umdrehungen pro Sekunde sehr zuverlässig funktioniert.

Im Rotationsdisplay werden zwei unterschiedliche Spannungen benötigt. Die Komponenten auf dem Ring, also der Raspberry Pi, die LED-Streifen und der Drehzahlsensor, benötigen 5V als Betriebsspannung. Der Gleichstrommotor im Sockel läuft mit 24V. Daher wird für die Stromversorgung ein 24V Netzteil verwendet. Die Umwandlung von 24V auf 5V für den Ring findet im Rotor statt. In Verbindung mit einem Pufferkondensator können die kurzen Spannungsaussetzer kompensiert werden, die durch die Schleifkontaktierung entstehen.

Sogenannte Step-Down-Module **5** regulieren die ankommenden 24V auf 5V (einstellbar) und stellen jeweils einen Strom von maximal 3A bereit. Insgesamt sind drei Module verbaut, die die beiden LED-Stränge und den Raspberry Zero versorgen.

Antrieb

Um den Ring anzutreiben, wird ein Motor aus einem alten Tintenstrahldrucker mit einer maximalen Drehzahl von 4000 U/min im unbelasteten Betrieb verwendet. Die Übertragung auf den Ring findet mittels eines GT2-Zahn-



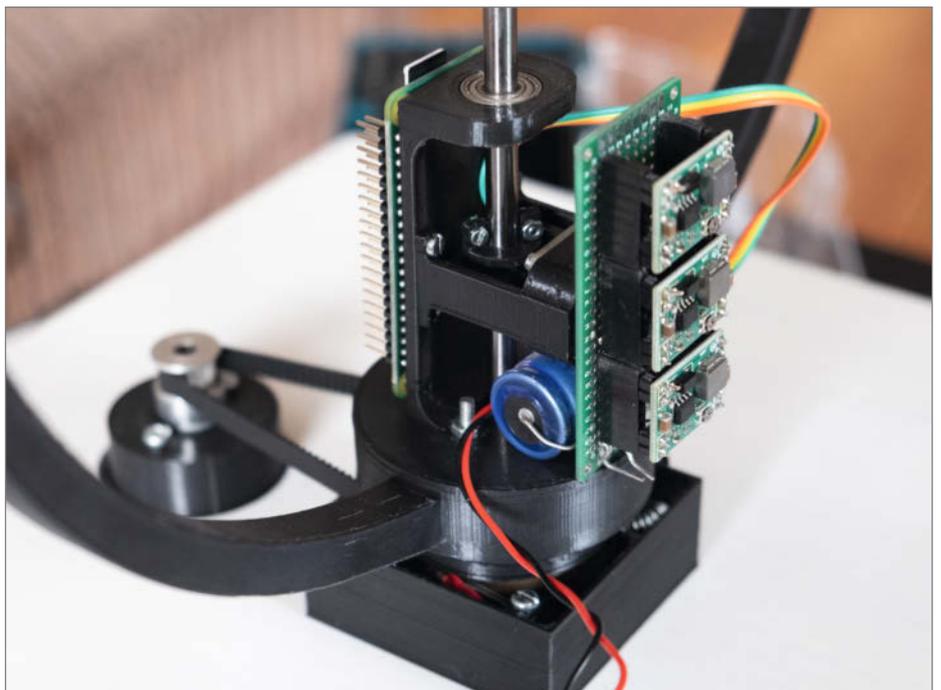
4 Schleifkontaktierung. Links: Sockelmodul, rechts: Rotormodul mit zwei Messingringen und Zahnscheibe

riemens statt **6**, der über eine Zahnscheibe mit 20 Zähnen am Motor und eine Scheibe mit 68 Zähnen am Rotor läuft. Aus dem Quotienten der Anzahl der Zähne ergibt sich eine Untersetzung mit einem Faktor von etwa 0,29. Die maximale Drehzahl am Rotor beträgt also theoretisch ca. 19 Umdrehungen pro Sekunde (U/s). Durch Verluste (Reibung, Luftwiderstand) sind jedoch nur ca. 15 U/s stabil möglich.

Für den Motor kommt ein kleines Fertigmodul zum Einsatz. Dies besteht aus einem H-Brückenmodul, das die Leistungsstufe bildet und die Drehzahl über einen 0V bis 5V Analogeingang steuerbar macht. Zusätzlich

stellt es auch 5V für einen *ATtiny85*-Mikrocontroller bereit, der die Logik für die Motorsteuerung implementiert. Daran angeschlossen ist ein Potentiometer zur Regelung der Drehzahl sowie ein Schalter zum Ein- und Ausschalten des Antriebs. Ist der Schalter eingeschaltet, liest der ATtiny den Wert des Potentiometers aus und gibt die entsprechende Spannung an das H-Brückenmodul aus.

Um zu verhindern, dass der Motor direkt mit der am Potentiometer eingestellten Drehzahl versucht, den stehenden Ring in Rotation zu versetzen, wird die Spannung beim Einschalten nicht direkt gesetzt, sondern



5 Step-Down Module und Pufferkondensator, Pin-Header und SD-Karte des Raspberry Pi Zero W

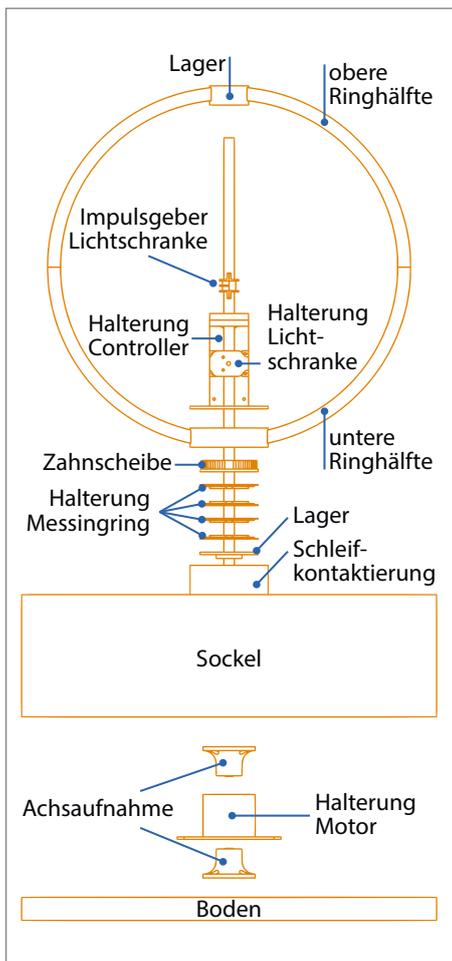


6 Motor in der Fassung mit montierter Zahnscheibe und Zahnriemen



7 Gesamter Aufbau des Displays mit Sockel, Rotor-Ring und den LEDs

kontinuierlich erhöht (Sanftanlauf). Beim Ausschalten wird analog dazu die Spannung langsam heruntergefahren, um den Rotor nicht schlagartig abzubremsen.



8 Explosionszeichnung des Systems

Aufbau

Als Basis für den Aufbau 7 dient ein Holzsockel. Die Stärke der verwendeten MDF-Bretter ist etwas großzügiger gewählt, um dem Sockel und damit dem gesamten Aufbau durch das höhere Gewicht einen stabilen Stand zu verschaffen. Dies ist wichtig, um den Rotor gleichmäßig und resonanzfrei laufen zu lassen. Auf der Rückseite des Sockels hält ein Aluminiumblech die Netzteil-Buchse sowie den Schalter und Potentiometer zur Steuerung des Motors. Der Motor wird auf der Oberseite des Sockels mithilfe einer gedruckten Fassung verschraubt (STL-Druckvorlagen siehe Kurzinfor).

Die Drehachse 8 für den Ring bildet ein Stahl-Rundstab mit 8mm Durchmesser. Er wird durch eine Bohrung in der Mitte der Sockeloberseite geführt und im Sockel mit gedruckten Aufnahmen fixiert. Um die Achse herum ist auf der Oberseite des Sockels der statische Teil der Schleifkontaktierung verschraubt.

Der Rotor besteht aus mehreren Teilen, die mit zwei Gewindestiften verschraubt sind. Der Ring mit den LEDs bildet den größten Teil und besteht aufgrund der Druckraumgröße meines *Anycubic i3 Mega* wiederum aus zwei Hälften. Ober- und Unterhälfte des Rings werden mit Zweikomponentenkleber verklebt. In der oberen Hälfte findet ein Lager Platz, durch das der Rotor um die Drehachse läuft. An der unteren Hälfte ist in den Ring hineinragend eine gedruckte Halterung für die elektronischen Komponenten der Ansteuerung und die Step-Down-Module verschraubt. Außerdem ist hier das Lichtschrankenmodul für die Drehzahlmessung montiert. Unterhalb des Rings folgen dann mehrere gestapelte Druckteile. Zuerst liegt eine gedruckte Zahnscheibe,

über die der Rotor mit einem GT2-Riemen angetrieben wird. Darunter folgen die beiden Messingringe der Schleifkontaktierung. In das unterste Teil ist ein weiteres Lager eingepresst.

Softwarestruktur

Um ein für das Auge stehendes Bild zu erzeugen, ist die exakte Ansteuerung der LEDs zum richtigen Zeitpunkt zu gewährleisten. Die Software 9 (Link in Kurzinfor) besteht deshalb zum einen aus einem zeitkritischen Teil, der die Ansteuerung der LEDs und die Auswertung des Drehzahlsensors übernimmt. Zum anderen aus einem zeitunkritischen Teil, der die Bildinhalte vorverarbeitet und zur Verfügung stellt. Als Schnittstelle zwischen beiden Programmteilen dient ein gemeinsamer Speicherbereich (Shared Memory), auf den beide Softwareteile zugreifen können. Der zeitkritische Treiber ist in C geschrieben, um Geschwindigkeit und Zuverlässigkeit zu maximieren, der vorverarbeitende, zeitunkritische Zuspield Teil aus Gründen der Flexibilität und Wartbarkeit in Python.

Zeitgenaue Ansteuerung

Die Ansteuerung der LEDs wird durch zwei Parameter bedingt: zum einen durch die Rotationsgeschwindigkeit, zum anderen durch den Startzeitpunkt einer Umdrehung. Beide Werte liefert das Lichtschrankenmodul. Zur Anbindung kommt die *WiringPi*-Bibliothek zum Einsatz, die Zugriff auf die GPIO-Pins des Raspberry Pi bietet. Über sie wird eine Funktion im Code mit einem definierten Ereignis an einem GPIO-Pin des Raspberry Pi verknüpft. Die Lichtschranke wird für die Dauer von je einer halben Umdrehung blockiert bzw. freigegeben. Daraus ergeben sich pro Umdrehung eine steigende und eine fallende Flanke am verbundenen GPIO-Pin. In diesem Fall wird mit der fallenden Flanke gearbeitet: Zunächst wird der aktuelle Zeitstempel gespeichert. Aus der Differenz zwischen neuem und vorherigem Zeitstempel wird dann die Umdrehungsdauer berechnet. Sie wird gewichtet mit der Dauer der letzten Umdrehung verrechnet, um Ungenauigkeiten in den Messungen auszugleichen.

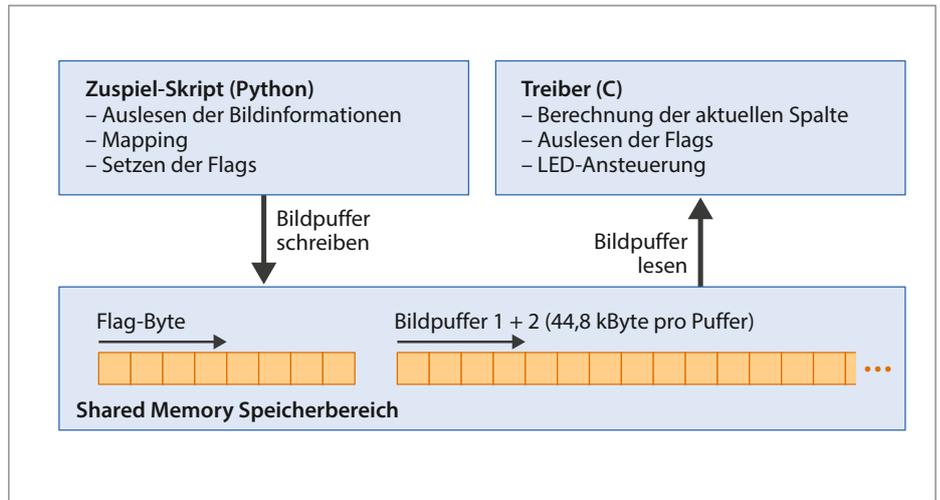
Nun ist also bekannt, wie lang die Umdrehung dauert und wann sie begonnen hat. Damit kann berechnet werden, wo sich der Ring momentan innerhalb der Umdrehung befindet, welcher der nächste Spaltenindex ist und wann die Aktualisierung dafür stattfinden muss.

Der verbleibende Teil des Programms ist nicht an den Status der Lichtschranke geknüpft. Es prüft kontinuierlich, ob sich der aktuelle Zeitstempel zuzüglich einer Toleranz in der Nähe des nächsten Ansteuer-Zeitpunktes befindet. Liegt der aktuelle Zeitstempel außerhalb der Toleranz und vor dem nächsten Ansteuer-Zeitpunkt, wird gewartet. Wenn er dahinter liegt, weil beispielsweise die An-

steuerung zu spät reagiert, wird die aktuelle Spalte übersprungen und direkt auf die nächste Spalte gewartet um das „Anstauen“ von Spalten zu verhindern. Liegt der Zeitpunkt innerhalb der Toleranz, wird die Aktualisierung der LEDs gestartet. Dazu werden die zu schreibenden Daten an das SPI übergeben. Der momentan benötigte Datensatz wird über einen Versatz innerhalb des Bildpuffers adressiert. Dieser Versatz ergibt sich aus dem aktuellen Spaltenindex.

Die Daten des Bildpuffers wiederum liegen in einem gemeinsamen Speicherbereich. Dieser wird beim Start durch das Python-Skript reserviert und ist danach aus dem C-Code über einen definierten Schlüssel erreichbar. Die Shared-Memory-Funktionalität wird vom Betriebssystem zur Verfügung gestellt und wird in C und in Python von `SysV`-Bibliotheken nutzbar gemacht. Der reservierte Bereich ist so groß, dass der gesamte Bildpuffer dort zweimal abgelegt werden kann. Außerdem kommt ein Status-Byte hinzu. Die doppelte Auslegung ermöglicht es, den aktiven Bildpuffer anzuzeigen, während der inaktive Bereich mit neuen Daten gefüllt wird (*Double Buffering*). Welcher Bildpuffer der aktive ist, zeigt ein Bit im Status-Byte an. Es wird zu Beginn jeder Umdrehung ausgelesen.

Ein weiteres Bit im Status-Byte aktiviert eine Rotation des aktuellen Bildes. Damit sind Pseudoanimationen, beispielsweise eine drehende Erdkugel, möglich. Die Rotation wird erreicht, indem zum aktuellen Spaltenindex ein Versatz addiert wird, der alle 200ms erhöht wird. Dadurch wandert das Bild in 20s einmal um die Drehachse. Die Daten im Bildpuffer bleiben dadurch unverändert. Das zugehörige Status-Bit wird ebenfalls zu Beginn einer Umdrehung ausgelesen.



9 Die Softwarestruktur in der Übersicht

Der C-Treiber 10 enthält einige Optimierungen für größtmögliche Performanz. Zum einen wird mit O3-Optimierung kompiliert. Dadurch werden beispielsweise die Anweisungen aufgerufener Funktionen während des Kompiliervorgangs an die aufrufende Stelle kopiert, sodass im Programmablauf kein Funktionsaufruf und somit auch keine Sprünge innerhalb des Programms mehr stattfinden.

Zum anderen wird die Priorität des Prozesses beim Scheduler des Betriebssystems erhöht. Er ist dafür verantwortlich, die verfügbare Rechenzeit unter den laufenden Prozessen aufzuteilen. Dabei werden Prozesse mit höherer Priorität bevorzugt behandelt und erhalten dadurch häufiger Rechenzeit. Das ist in diesem Fall besonders wichtig, da die Ansteuerung zwar nicht viel, dafür aber in möglichst kurzen Intervallen Rechenzeit benötigt.

Vorverarbeitung der Inhalte

Um dem C-Treiber die Farbdaten in einem Format zu übergeben, das nach Möglichkeit direkt an die LEDs weitergegeben werden kann, findet eine Konvertierung der Bilddaten statt. Der Ausgangspunkt sind Bilder im PNG- oder JPG-Format, die in einem bestimmten Ordner im Dateisystem abgelegt sind. Diese werden für die Konvertierung vom Python-Zuspielskript eingelesen und auf ihre Auflösung geprüft.

Bilddateien, die den Anforderungen entsprechen, werden dann in drei Schritten in ein Format gebracht, das sich schnell verarbeiten lässt. Das Ziel ist, die Farbdaten aller Pixel aneinandergereiht zu speichern. Dabei gehören immer vier Byte-Werte zu einem Pixel, 112 Pixel zu einer Spalte und 100 Spalten zu einem Bild.

Heft + PDF mit 29 % Rabatt



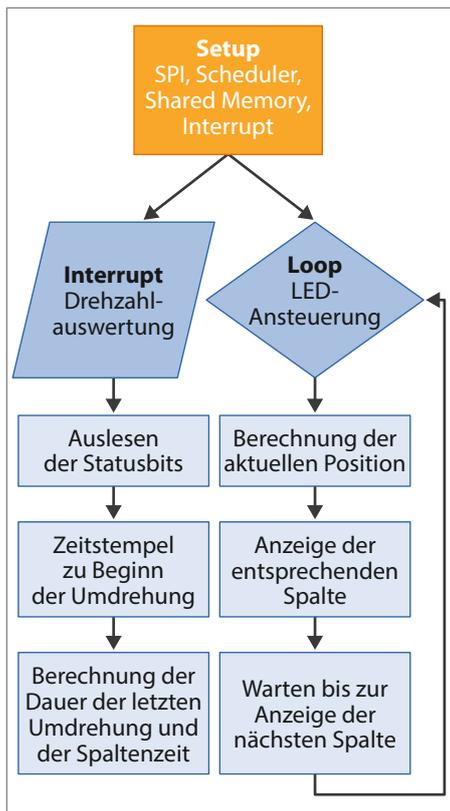
shop.heise.de/ct-datenschuetzen

Verteidigen Sie Ihre Privatsphäre!

- ▶ E-Mails und Telefone absichern
- ▶ Welche Messenger verschlüsseln wirksam
- ▶ Welche Angriffe bedrohen Ihre Privatsphäre on- und offline
- ▶ Passwörter für alle Systeme sicher verwalten
- ▶ Für Abonnenten portofrei
- ▶ Auch im Set mit Reiner SCT Authenticator zum Sonderpreis

Heft für 14,90 € • PDF für 12,99 € • Bundle Heft + PDF 19,90 €

shop.heise.de/ct-datenschuetzen



10 Ablaufplan des Treiberprogramms



Dieses bekannte Motiv bot sich einfach an.

Nach diesem Schema wird also jedes Bild als lange Kette von Byte-Werten gespeichert.

Als erstes werden die Pixelpositionen für das Halbbildverfahren entsprechend den beiden Ringseiten berechnet und zugeordnet. Dann werden das rote und das blaue Farbbyte getauscht, um der Farbreihenfolge der APA102C-LEDs zu entsprechen. Im dritten Schritt erfolgt eine Gammakorrektur, um die Farbdarstellung zu verbessern. Im Anschluss werden die Pixeldaten aneinandergereiht und

jeweils das noch fehlende vierte Byte für die Helligkeit der LED hinzugefügt. Am Ende der Konvertierung liegt eine Liste mit allen Bildern als Kette von Byte-Werten vor.

Soll ein Bild angezeigt werden, liest das Zuspieldskript zunächst das Status-Byte im gemeinsamen Speicherbereich aus, um zu bestimmen, welcher Bildpuffer aktiv ist. Danach werden die Byte-Werte im inaktiven Bildpuffer abgelegt. Zuletzt wird das Bit für den aktiven Bildpuffer invertiert, sodass in der nächsten Umdrehung der aktive Puffer wechselt und die neuen Bilddaten angezeigt werden.

Über das Zuspieldskript kann auch das Bit im Status-Byte gesetzt werden, dass die Rotations-Animation des Bildes aktiviert oder deaktiviert.

Die notwendige Interaktion mit dem Python-Zuspieldskript, um ein neues Bild anzuzeigen oder die Rotation ein- und auszuschalten, kann entweder über Eingaben in die Konsole per SSH-Verbindung oder per UDP-Nachricht erfolgen. Durch die Flexibilität von Python können so auch schnell neue Wege der Interaktion oder Bereitstellung von Inhalten implementiert werden.

Animierte Inhalte

Auf diesem Stand war es nun möglich, statische Bilder anzuzeigen und durch die Rotationsfunktion einfache Animationen darzustellen. Da Funktion und Umfang die Anforderungen an meine Abschlussarbeit erfüllt hatten, landete das Rotationsdisplay nun für einige Zeit im Regal.

In letzter Zeit hat es mich aber dann in den Fingern gejuckt, eine länger angedachte Funktion umzusetzen, nämlich die Wiedergabe echter Animationen. Bisher wurde lediglich das Bild verschoben, ohne dass sich die zugrundeliegenden Daten veränderten. Die Erweiterung auf echte Animationen sollte ermöglichen, Einzelbilder eines Videos zu konvertieren und dann auf dem Display wiederzugeben.

Im Grunde waren dazu nur wenige Anpassungen nötig, da die beiden bisherigen Programmteile die Aufgaben sehr strikt trennen. Im C-Treiber ist die einzige Neuerung, zu Beginn jeder Umdrehung ein weiteres Bit im Status-Byte zu setzen, das den Takt für den Bildwechsel angibt.

Auf Seiten der Zuspieldung waren die Änderungen etwas komplexer. Da die Konvertierung von Bildern bereits bei einer kleinen Anzahl zu einer relativ langen Ladezeit führt, ist es nicht sinnvoll, Sequenzen mit mehreren hundert Einzelbildern bei jedem Programmstart zu konvertieren. Der Raspberry Pi ist für diese Aufgabe nicht optimal geeignet. Daher kommt ein weiteres Python-Skript dazu, das nicht auf dem Raspberry Pi läuft, sondern auf einem Laptop oder Desktop-Computer um-

fangreiche Einzelbildsequenzen konvertiert. Das Zwischenergebnis ist das gleiche wie bei der bisherigen Konvertierung, es wird jedoch deutlich schneller erreicht. Die Byte-Strings der einzelnen Bilder der Sequenz werden wieder in einer Liste gesammelt. Diese wird dann mittels der Python-JSON-Bibliothek serialisiert und in einer Datei gespeichert. Diese wird dann vom Zuspieldskript auf Raspberry Pi wieder eingelesen.

Soll nun eine Animation angezeigt werden, wartet das Zuspieldskript auf den Impuls des C-Treibers, der anzeigt, dass die nächste Umdrehung begonnen hat. Sobald der Impuls kommt, wird das Bit im Status-Byte wieder gelöscht. Während die Umdrehung läuft, schreibt das Zuspieldskript das erste Bild der Sequenz in den inaktiven Bildpuffer. Sobald das abgeschlossen ist, wird das Bit für den aktiven Bildpuffer invertiert, sodass das erste Bild während der nächsten Umdrehung angezeigt wird. Kommt der Impuls für diese nächste Umdrehung, wechselt der C-Code auf den neuen Bildpuffer und das Zuspieldskript kann nun das zweite Bild der Sequenz in den anderen, jetzt inaktiven Bildpuffer schreiben und daraufhin wieder einen Wechsel des Bildpuffers über das Status-Byte anzeigen.

Dieser Mechanismus läuft, bis alle Bilder der Sequenz angezeigt wurden und beginnt dann wieder von vorn. Er garantiert, dass jedes Bild der Sequenz für eine Umdrehung angezeigt wird, was wiederum bedeutet, dass die Wiedergabegeschwindigkeit der Animation direkt von der Drehzahl abhängig ist.

Und jetzt?

Auch wenn das Rotationsdisplay letztlich als hübscher Effekt und weniger als praktisches Display zu verstehen ist, zeigt der Aufbau des Prototypen, dass es mit den Mitteln eines Makers, wie 3D-Drucker, LötKolben, Raspberry Pi und LED-Streifen, durchaus möglich ist, auch für komplexere Systeme den Zustand eines *Proof of Concept* zu erreichen. Außerdem hatte ich nicht nur viel Spaß bei der Konzeption und Umsetzung des Projekts, sondern konnte durch die vielen kleinen und größeren Herausforderungen jede Menge lernen.

Beispielsweise stellte sich die strikte Trennung der beiden Softwareteile als optimale Grundlage für neue Funktionen heraus. Der nächste Schritt ist nun herauszufinden, wo die softwareseitigen Grenzen der horizontalen Auflösung, also der Anzahl an Aktualisierungen pro Umdrehungen liegt. Und ich denke, darüber hinaus werden sich noch einige Erweiterungen finden, wie der Upload von Bildern per Website. Aber auch abgesehen von neuen Funktionen und Entwicklungsschritten gibt es durch die ungewöhnliche Form des Displays viele interessante und schöne Motive, die ausprobiert werden wollen. —caw

NEU



KNIPEX Quality – Made in Germany

Zangen- schlüssel XS

Nur 100 Millimeter klein:
bis Schlüsselweite 21 mm
schonend schrauben, Werk-
stücke sicher greifen, halten,
pressen und biegen

- ▶ Optimale Zugänglichkeit bei engsten Platz-
verhältnissen: besonders kompakte Bauform,
sehr schlanker Kopf
- ▶ Einhändige Feinverstellung für die einfachste
Anpassung an verschiedene Werkstückgrößen
durch Anschieben
- ▶ Robustes Gleitgelenk bietet hohe Stabilität
und Belastbarkeit
- ▶ Glatte Backen für die schonende Montage
von empfindlichen und oberflächenveredel-
ten Armaturen: Arbeiten direkt auf Chrom
durch spielfreie, vollflächige Anlage!
- ▶ Vielseitige Einsatzgebiete: Feinmechanik,
Sanitär, Bordwerkzeug für Fahrrad oder Mo-
torrad, EDC (Every Day Carry), Modellbau
- ▶ Klemmschutz verhindert Quetschverletzungen
- ▶ Feinverstellung mit 10 Einstellpositionen



Trotz geringer Größe greift der
Zangenschlüssel XS Muttern und
parallele Werkstücke bis 21 mm
sicher und schonend.

Smartes Alarmsystem mit ESP32-CAM

Passend zum Beginn der dunklen Jahreszeit baue ich ein Einbruchmeldesystem mit optionaler Alexa-Integration. Dazu verwende ich eine ESP32-CAM und ergänze sie um ein Infrarot-Bewegungserkennungsmodul.

von Andreas Koritnik



Mit nur wenigen Bauteilen ist dieses Alarmsystem schnell gebaut. Neben der Alexa-Integration unterstützt es die Alarmierung per Telefonanruf und das Versenden von E-Mails mit angehängtem Bild. Wichtig sind dabei sowohl die gute Bedienbarkeit als auch die Vermeidung von Fehlalarmen. In diesem Projekt nutze ich ein Infrarot-Bewegungserkennungsmodul, um mögliche Einbrecher zu erkennen. Es ist zudem möglich, das System einfach um Tür- und Fenstersensoren zu erweitern – zum Beispiel mit batteriebetriebenen ESP12-Modulen im *Deep-Sleep*-Modus, die Alarme per WLAN an die ESP32-CAM melden. Der Schaltplan und das Programm sind in meinem Github zu finden (siehe Downloadlink in der Kurzinfor).

Das Alarmsystem basiert auf der *Espalexa*-Bibliothek, die ein Philips-Hue-Gerät emuliert, sodass es im Netzwerk von Amazons Sprachassistentin erkannt wird. Dadurch kann das Alarmsystem über Alexa-Sprachbefehle ein- und ausgeschaltet werden. Das hat sich als sehr bequem und praktisch erwiesen. Alternativ kann das System über die Alexa-App oder ein Web-Interface gesteuert werden. Wie schon in der Make 3/21 im Projekt zur Gesichtserkennung (ab S. 62) kommt der URL-Trigger-Service von www.virtualsmarthome.xyz zur Anwendung. Hier nutze ich ihn jedoch, um Alarm-Routinen zu starten, wie einen Alarm-sound abzuspielen oder das Licht zu schalten.

Um Fehlalarme zu vermeiden, wartet der Melder bei einer erkannten Bewegung 30 Sekunden, ob eine weitere Bewegung erfolgt. Ist dies der Fall, ruft der ESP außerdem auf einem Mobiltelefon an (dies erfordert eine Fritzbox) und schickt per E-Mail ein Bild, das von der ESP-CAM aufgenommen wurde. Speziell das angehängte Bild erlaubt schon eine erste Einschätzung, ob es sich um einen echten oder um einen Fehlalarm handelt. Ist kein Eindringling auf dem Bild zu sehen, handelt es sich wahrscheinlich um einen Fehlalarm. Sofern ein VPN-Zugang zum heimischen Netzwerk existiert, kann aus der Ferne per Web-Interface auf das Live-Streaming-Bild der ESP32-CAM zugegriffen werden. Spätestens jetzt sollte erkennbar sein, ob es sich um einen Fehlalarm oder einen echten Einbruch handelt.

Software

Die Software kann mit der kostenlosen Programmierumgebung von Arduino angepasst und auf das Board übertragen werden. Dazu muss die ESP32-Board-Unterstützung in der IDE installiert sein. Außerdem benötigt das Projekt fünf externe Bibliotheken, die eingebunden werden müssen: *ArduinoWebsockets*, *EMailSender*, *Espalexa*, *ArduinoJson* und *TR-064*. Ausführliche Anleitungen für beides finden Sie online, siehe Download-Link in der

Kurzinfor

- » ESP32-Cam mit Infrarotsensor verbinden
- » Alexa-Anbindung über espalexa-Bibliothek
- » Alarm per Telefon, E-Mail oder Alexa schicken

Checkliste



Zeitaufwand:
2 Stunden



Kosten:
20 Euro



Programmieren:
Grundkenntnisse Arduino

Material

- » ESP32-CAM AI Thinker
- » HC-SR501 PIR-Sensor
- » 5-Volt-Netzteil
- » USB-zu-Seriell-Konverter
- » Kabel
- » Ferritring optional

Werkzeug

- » Lötkolben
- » 3D-Drucker optional

Mehr zum Thema

- » Andreas Koritnik, Gesichtserkennung steuert Alexa, Make 3/21, S. 62
- » Michael Jentsch, Smarter Bewegungsmelder, Make 4/19, S. 66
- » Daniel Bachfeld, Intelligente Webcam für 5 Euro, Make 1/20, S. 28

Alles zum Artikel
im Web unter
make-magazin.de/xter

Kurzinfor. Im dort angegebenen Github-Repository stelle ich die komplette Software, inklusive weitergehender Erklärungen, zum Download bereit. In der Arduino-IDE wählt man nun unter *Werkzeuge/Board* das *ESP32 Wrover Module* aus und unter *Partition Scheme* dann *Default 4 MB mit Spiiffs...*

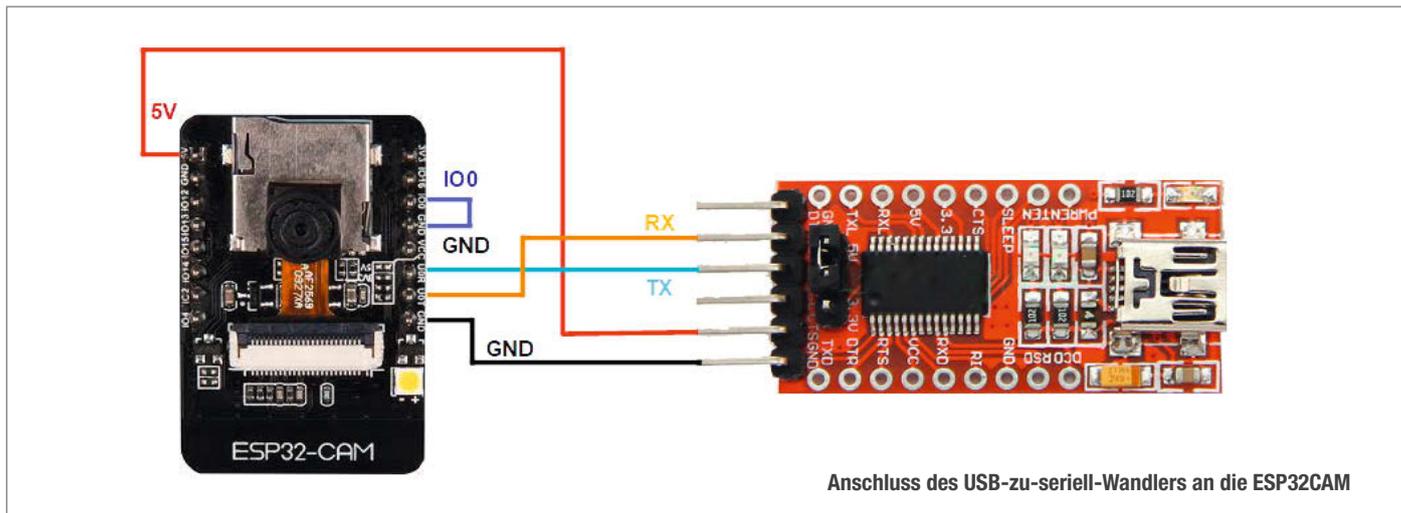
Vor dem ersten Programmieren des Boards sollten die Einstellungen im Programmcode angepasst werden. Dies gilt insbesondere für die WLAN-Einstellungen, damit das System sich mit dem lokalen Netzwerk verbinden kann. Danach sind Uploads *Over the Air* (OTA) über das WLAN möglich.

Flashen

Um das initiale Programm auf dem ESP32-CAM-Modul zu installieren, braucht man einen externen USB-zu-Seriell-Adapter, der mit dem Jumper auf 5 Volt eingestellt wird. Anschließend müssen die Pins IO0 und GND, RX und TX über Kreuz sowie 5V und GND verbunden werden. Falls der Adapter mit Netzteil und PIR-Sensor genutzt wird, entfällt die 5V-Leitung. Dann die *RST*-Taste auf der Cam drücken und den Upload-Prozess in der Arduino-IDE starten – dort muss noch der USB-Serial-Port des Adapters ausgewählt sein. Schließlich IO0 von



Die Kamera im eigens 3D-gedruckten Gehäuse



GND lösen und erneut *RST* drücken, um die Cam neu zu starten.

Um ab jetzt Updates Over the Air einzuspielen, muss man in der Arduino-IDE den Netzwerk-Gerätenamen (*AlarmSystem*) statt des seriellen Ports einstellen. Der USB-Adapter wird nicht mehr benötigt. Manchmal ist nach dem OTA-Upload die *RST*-Taste zu drücken, um das aktualisierte Programm zu starten. Alternativ kann man kurz die Stromversorgung durch Ziehen des Netzteils unterbrechen.

Einstellungen im Code

Die Benachrichtigungsmethoden können im Code einzeln ein- oder ausschaltet werden. Anrufe als Alarmbenachrichtigung sind nur mit bestimmten WLAN-Routern möglich, da ich das Protokoll TR-064 nutze. Es funktioniert nur mit Modellen mit Telefonsupport, hauptsächlich von AVM und bei einigen Zyxel-Modellen. Mit *SILENT_ALARM On* kann schließlich ein stiller Alarm einschaltet werden. Das be-

deutet, dass keine interne LED der ESP32-CAM blinkt und kein Alexa-Benachrichtigungston abgespielt wird.

Unter *WLAN credentials* werden Name und Passwort des eigenen WLANs eingetragen. Der GeräteName *AlexaName* sollte von *Alarm* auf etwas Eigenes geändert werden, um sicherzustellen, dass kein Einbrecher den Gerätenamen errät und den Alarm ausschaltet.

Um TR-064 zu nutzen, muss auf der Fritzbox ein Benutzer angelegt und die Funktion *Wählhilfe* aktiviert werden. Details sind in der Bibliotheksdokumentation zu finden. Neben den Anmeldeinformationen werden bei den entsprechenden Variablen (*FB_****) noch die anzurufenden Telefonnummer und die IP-Adresse der Fritzbox eingetragen.

E-Mail-Benachrichtigungen versende ich mit der Bibliothek *E-MailSender*. Die Voreinstellungen beziehen sich auf Google-Mail-Konten, es ist jedoch möglich, andere Provider einzutragen. Ich habe einfach auf *mail.google.com* ein neues Gmail-Konto erstellt und die Anmeldeinformationen im Programm eingesetzt (*M_****). Wichtig: Es müssen *weniger sichere Verbindungen* in den Google-Mail-Sicherheitseinstellungen zugelassen werden. Bei *destination@domain.de* wird die eigene E-Mail-Adresse angegeben. Im Alarmfall trifft hier eine Benachrichtigung mit dem Betreff „Intruder Alert!“ und dem Text „Zeitstempel: Movement detected!“ ein. Außerdem wird ein Bild angehängt, das bei der zweiten erkannten Bewegung aufgenommen wurde.

Für die Alexa-Routinen habe ich mich bei *www.virtualsmarthome.xyz* mit meinem Amazon-Konto angemeldet und den Skill *URL Routine Trigger* in Alexa aktiviert. Bei *virtualsmarthome* habe ich dann zwei Trigger (*Alarm* und *Ping*) angelegt, deren URLs in das Programm kopiert werden (**URL[]*). Der Dienst stellt drei Varianten der URLs bereit, die alle funktionieren sollten: HTML, XML und JSON. Ich nutze letzteres.

Einstellungen

```
// Change credentials, Alexa device name and Trigger URLs!
//*****
#define CALL_PHONE On
#define SEND_MAIL On
#define USE_ALEXA On
#define SILENT_ALARM Off

// WLAN credentials
char* ssid = "ssid";
char* password = "password";

// Alexa device name
char* AlexaName = "Alarm";

// Fritzbox credentials for TR064
char* FB_NUMBER = "0123456789";
char* FB_USER = "user";
char* FB_PASSWORD = "password";
char* FB_IP = "192.168.0.1";
int FB_PORT = 49000;

// Mail credentials for gmail account
char* M_USER = "myaccount@gmail.com";
char* M_PASSWORD = "password";
char* M_DEST = "destination@domain.de";

// Trigger URLs: 1. Alexa alarm routine. 2. Ping notification sound
const char *URL[] PROGMEM={
    "https://www.virtualsmarthome.xyz/url_routine_trigger/activate.
    php?trigger=",
    "https://www.virtualsmarthome.xyz/url_routine_trigger/activate.
    php?trigger="
};
```

Um verschlüsselte Kommunikation zu ermöglichen, muss im Code ein Root-Zertifikat hinterlegt sein. In der Datei *certificate.h* habe ich dafür *ISRG Root X1* der Zertifizierungsstelle *Let's Encrypt* eingebaut. Damit sollte das Programm auch ab Oktober funktionieren, wenn das von *virtualsemarthome.xyz* genutzte Zertifikat Ende September abläuft. Ich werde die Situation beobachten und gegebenenfalls nötige Änderungen auf Github dokumentieren. Weitere Infos zu den Hintergründen finden Sie auch in Make 3/21.

Hardware

Das Alarmsystem besteht nur aus drei Komponenten: einem 5-Volt-Netzteil, einer ESP32-CAM und dem HC-SR501-PIR-Sensor. Zusätzlich ist für die erste Programmierung der USB-zu-Seriell-Adapter notwendig. Bezugsquellen für alle verwendeten Komponenten sind auf Github zu finden. Dort stehen auch Dateien für ein Gehäuse zum 3D-Drucken bereit.

Das HC-SR501-Modul ist ein passiver Infrarot-Sensor (PIR, Passive InfraRed) mit integrierter Auswerteelektronik. Der Sensor reagiert auf Änderungen der Wärmeverteilung im Raum, die beispielsweise eine sich bewegende Person auslöst. Genauer vorgestellt haben wir ihn in der Make 4/19 ab Seite 66. Das Sensormodul wird mit drei Kabeln mit der ESP32-CAM verbunden. Die ESP32-CAM wird noch über eine 5,1mm-Steckbuchse oder direkt an das Kabel des 5-Volt-Netzteils angeschlossen – einfach die Kabel anlöteten oder auf einem Steckbrett mit Jumperkabeln zusammenstecken.

Das PIR-Modul ist sehr empfindlich. Zur Vermeidung von häufigen Fehlalarmen ist daher die richtige Einstellung der Empfindlichkeit sehr wichtig. Am Modul selbst kann man die Empfindlichkeit über das linke Potentiometer ändern. Drehen gegen den Uhrzeigersinn verringert die Empfindlichkeit. Sie sollte so weit reduziert sein, dass nicht zu viele falsche Auslöser auftreten, zum Beispiel durch normale Temperaturänderungen. Das rechte Potentiometer definiert die Haltezeit bzw. die Länge des Signals bei einem Alarm. Bitte ganz nach links auf die kürzeste Zeit einstellen (ca. drei Sekunden).

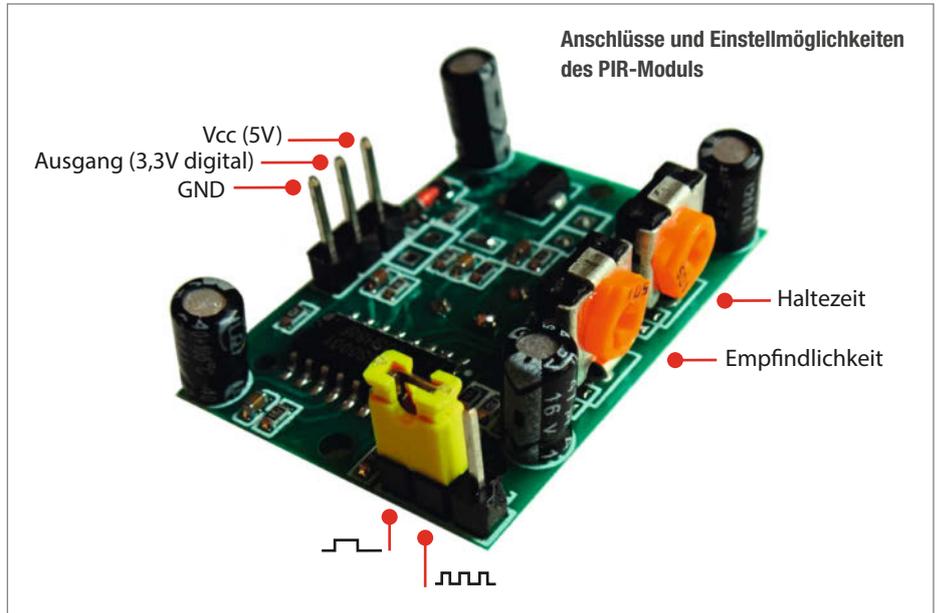
Um Interferenzen des WLANs zu minimieren, habe ich außerdem die Kabel zum PIR-Modul durch einen Ferritring geführt. Er schützt es vor Störungen durch das ESP-WLAN, die von den Kabeln übertragen werden. Ohne diese Filterung waren die Fehlalarme zumindest bei mir zu häufig. Zusätzlich kann es nötig sein, die WLAN-Sendeleistung zu verringern:

```
esp_wifi_set_max_tx_power(60);
```

Mit diesem Befehl in der Funktion *setup()* setze ich die Sendeleistung vom Maximum

Anschlüsse

ESP32-CAM	HC-SR501	Spannungsversorgung
5V	VCC	5V
GND	GND	GND
IO2	Ausgang	



20dBm auf 15dBm herunter. Das hat ebenfalls geholfen, Fehlalarme zu verhindern.

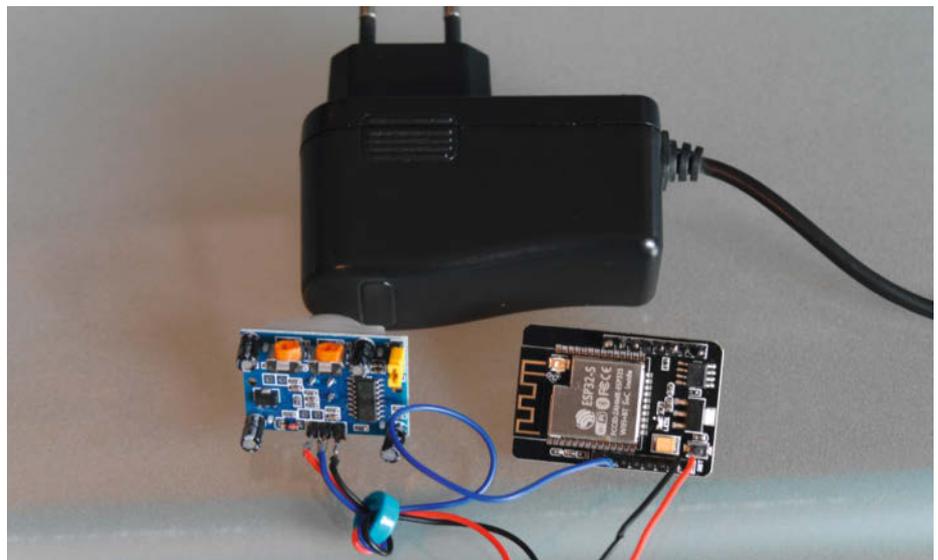
Nach dem Hochladen des Sketches und der Empfindlichkeitseinstellung kann man das Alarmmodul zusammenbauen und zu Hause so platzieren, dass es mit dem Infrarot-Sensor verdächtige Bewegungen erkennen kann.

Anbindung an Alexa

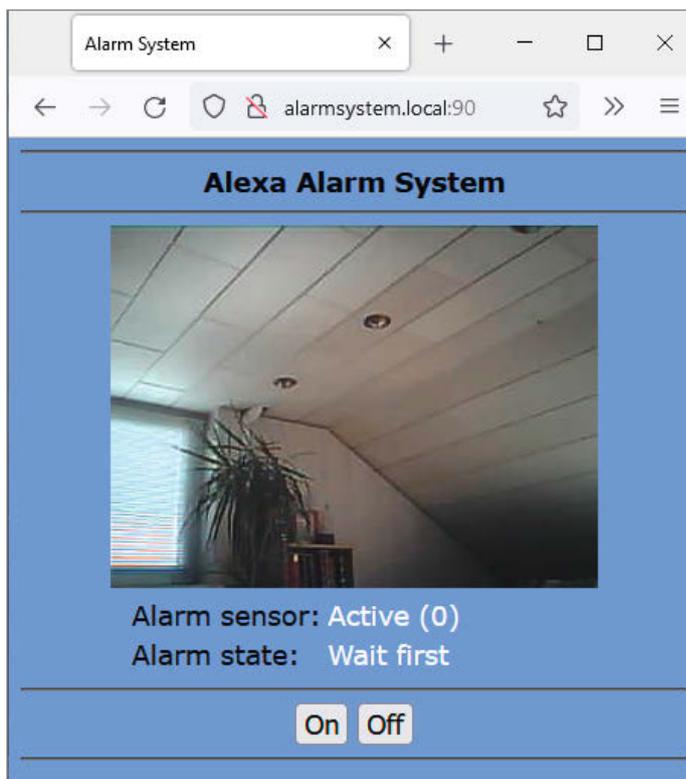
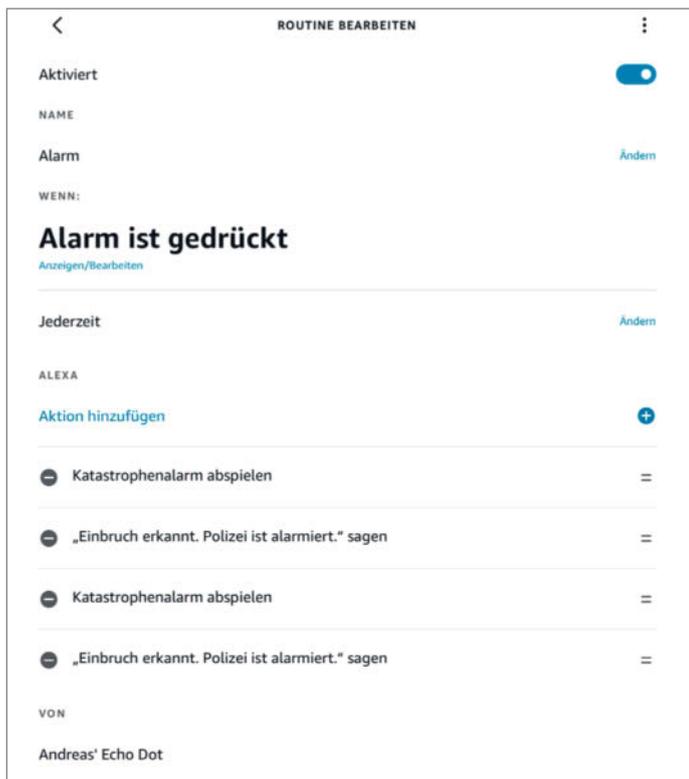
Nun kann die Gerätereckennung mit der Alexa-App gestartet werden. Obwohl die *Espalexa-*

Bibliothek ein Hue-Gerät emuliert, muss hier *Sonstige Geräte* gewählt werden. Alexa sollte dann ein neues Gerät mit dem im Programm angegebenen Namen anzeigen.

In der Alexa-App habe ich zwei Routinen erstellt: *Alarm* ist die Routine für Aktivitäten wie das Abspielen von Tönen in Alarmsituationen. *Ping* ist ein Benachrichtigungston, der mich beim Heimkommen erinnert, die Alarmanlage wieder zu deaktivieren. Die Nutzung der *Alexa-Drop-In-Funktion* wäre ebenfalls möglich. Damit kann man in den Raum hinein-



Der Hardware-Aufbau



Im Einbruchfall warnt meine Alexa mit einem Katastrophenalarm und droht mit der Polizei.

Das Web-Interface mit dem Live-Kamerabild habe ich sehr übersichtlich gehalten.

hören oder sogar mit einem eventuellen Ein-dringling kommunizieren. Wer mag, kann weitere, eigene Routinen umsetzen und etwa Lichter ein- und ausschalten.

Beim Verlassen der Wohnung oder des Hauses wird die Alarmanlage über „Alexa, Alarm an“ oder über die Alexa-App auf dem Smartphone eingeschaltet. Dann dauert es allerdings noch 60 Sekunden, bis der Melder aktiviert wird. Dies ermöglicht mir das Verlassen des Hauses, ohne den Bewegungsmelder zu aktivieren. Für die Rückkehr habe ich eine Verzögerungszeit von 15 Sekunden vorgesehen, in der das Modul mit „Alexa, Alarm aus“ ausgeschaltet wird, ohne einen Alarm auszulösen. Nach einem Alarm ist eine Wartezeit von 5 Minuten definiert, bevor ein neuer Alarm ausgelöst werden kann. Die Wartezeit kann mit jedem Ein/Aus-Befehl gestoppt werden.

Web-Interface

Um das System über ein Web-Interface zu steuern, gibt man einfach `IP-Adresse:90` im Browser ein. Der Port 90 ist notwendig, da der Port 80 bereits für den `Espalexa`-Dienst verwendet wird. Die IP-Adresse wird beim Start des Sketches im seriellen Monitor der IDE angezeigt. Falls der Client das Bonjour-Protokoll (mDNS) unterstützt, kann man auch `alarmsystem.local:90` im Browser ange-

ben. Der Videostream wird per WebSocket-Verbindung zum Client gesendet und nur ein Stream unterstützt. Versuchen weitere Clients eine Verbindung aufzubauen, wird eine Fehlermeldung angezeigt.

Im Browser wird nun die aktuelle Kameraansicht angezeigt. Im Alarmfall kann man so prüfen, wer sich im Raum bewegt. Mit den Buttons `On` und `Off` kann ich das System an- oder ausstellen. Beim Scharfschalten über das Web-Interface wird das System direkt, ohne Verzögerungszeit, aktiviert. Der Zustand `Active (0)` zeigt die Zähler für einzelne PIR-Trigger. Über die URL `IP-Adresse:90/uptime` kann man außerdem die aktuelle Uhrzeit, die Uptime (in Stunden), den verbleibenden Speicher des Systems sowie die Einzel-/Doppel-PIR-Zähler und die maximale Bildrate ansehen.

```
Time: 18:17:44
Uptime: 0 (hours)
Free Heap: 175984
Single=0 Double=0
Max FPS=5
```

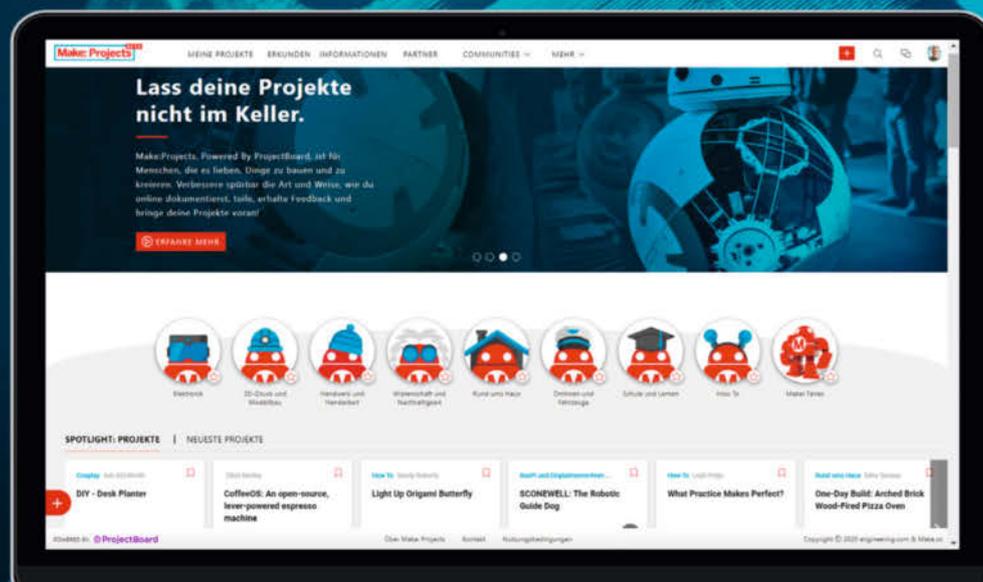
Die Zähler helfen beim Überprüfen und Einstellen der Empfindlichkeit des PIR-Moduls. Wenn die Empfindlichkeit zu hoch ist, sind falsche Trigger zu sehen. Ein Aus-An-Befehl setzt die Werte wieder auf null. Die Streaming-Bildfrequenz (FPS) habe ich im Programm auf 5 Bilder pro Sekunde eingestellt. Das ist aus-

reichend zur Beurteilung der Situation. Im WLAN sind zwar bis zu 30 Bilder/Sekunde möglich. Zu hohe Bildraten können aber die Steuerung des Geräts durch Alexa stören, da das Senden die zeitnahe Behandlung der Hue-Alexa-Kommunikation bei verbundenem Web-Client behindern kann.

Mit der URL `IP-Adresse:90/alarm_trigger?s=Sensorname` können externe (ESP12-)Sensoren einen Alarm triggern, wobei der Sensorname als Parameter `s` angegeben werden kann. Der Sensorname wird in der Alarm-E-Mail angezeigt. Wird die URL `IP-Adresse:90/test` im Browser eingegeben, wird die Funktion des Virtual-smarthome-URL-Trigger-servises getestet. Dazu wird die URL für die Benachrichtigungsroutine aufgerufen. Alexa sollte daraufhin den definierten Ton abspielen.

Diese Steuerung des Systems funktioniert allerdings nur, wenn man im heimischen WLAN eingeloggt ist. Um auch von unterwegs auf das Web-Interface zugreifen zu können, muss ein VPN-Zugang zum heimischen Netzwerk eingerichtet sein. Ich nutze dazu eine Fritzbox mit einer IPv4-Adresse. Das geht mit Apple- oder Android-Geräten recht einfach (Details siehe Github). Mit einem IPv6-Anschluss wird es komplizierter. Sie können dann einen Open-VPN-Server auf einem Raspberry Pi mit einem kommerziellen Portmapper-Dienst (z.B. `feste-ip.net`) verwenden. —hch

Make: Projects



**Zusammen mehr machen –
Teile deine Ideen, hol dir Inspiration,
revolutioniere deine Projekte**

... all das gemeinsam auf Make Projects!

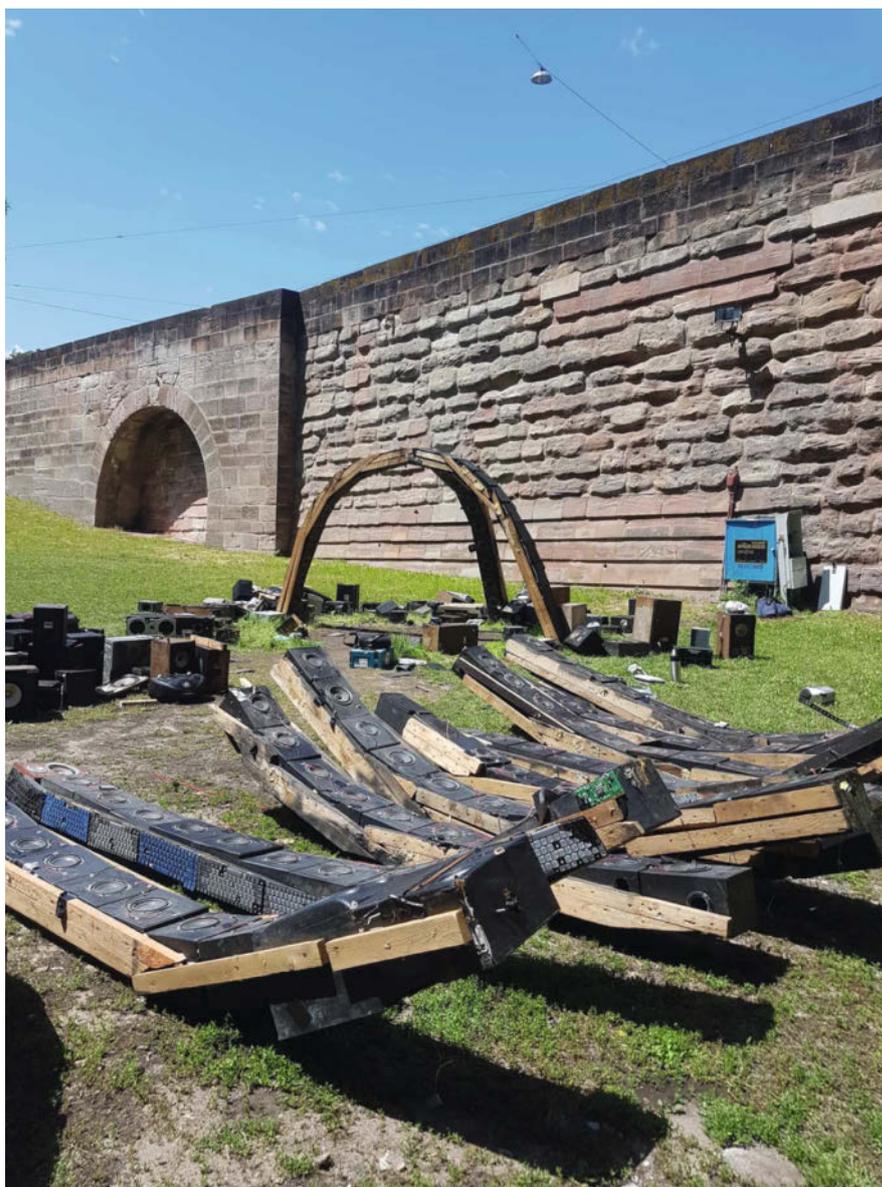
www.makeprojects.com/de

© copyright by Make: Media GmbH



Was uns inspiriert



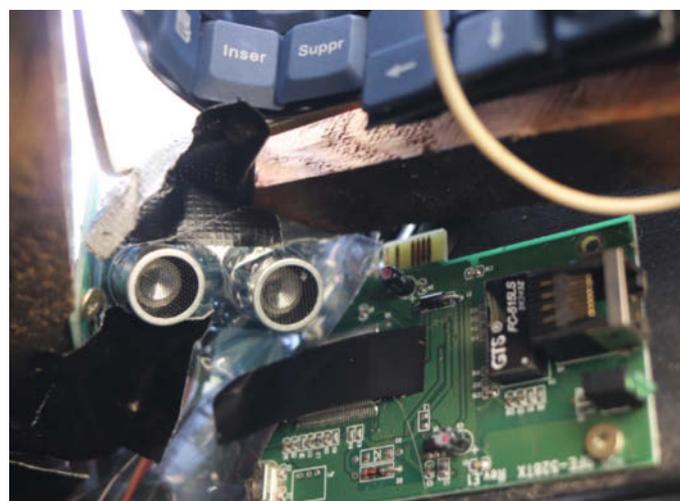


Heimatlose Lautsprecher

Der *Audio Igloo*, ein begehbare Klang-Kunstwerk aus 300 Lautsprecherboxen vom Sperrmüll, reagiert auf Besucher mit interaktiv gesteuerten Melodien und Klängen. Paul Biessmann ist Musiker und Klangkünstler, Benoît Maubrey Spezialist für Installationen und interaktive Skulpturen. Zusammen haben sie das halbkugelförmige Gebilde aus Lautsprechern konstruiert: Das begehbare Werk geht interaktiv auf den Besucher ein, indem es seine Bewegungen in sphärische Klänge und Tonfolgen umsetzt.

Sechs versteckte Ultraschall-Sensoren ermitteln im *Audio Igloo* dabei Position und Bewegungen des Besuchers, um über eine von Paul Biessmann erdachte Synthesizer-Matrix die Abfolge von Akkorden, Basslinien und einzelnen Noten oder auch die „Dichte“ von Geräuschen zu steuern. Verwendet haben Paul Biessmann und Benoît Maubrey dabei die visuelle Open-Source-Programmierungsumgebung *Pure Data*. —cm

► <https://heise.de/-6160783>



Biessmann/Maubrey



Margaret Low

Bestechend einfach

Programmieren lernen ist viel einfacher, wenn man ein konkretes Projekt hat – das man in diesem Fall am Ende sogar anfassen kann. Mit *TurtleStitch* können handarbeitsbegeisterte Programmier-Anfänger über eine *Scratch*-basierte Oberfläche ihre Stickmaschine direkt im Browser programmieren. Dort arbeitet man mit den klassischen vorgefertigten Code-Blöcken, die auf Stickmaschinen-Anweisungen angepasst wurden. In der Vorschau kann man genau erkennen, welche Stickbahnen die Maschine in welcher Reihenfolge abfahren wird – so entsteht tolle abstrakte Programmierkunst.

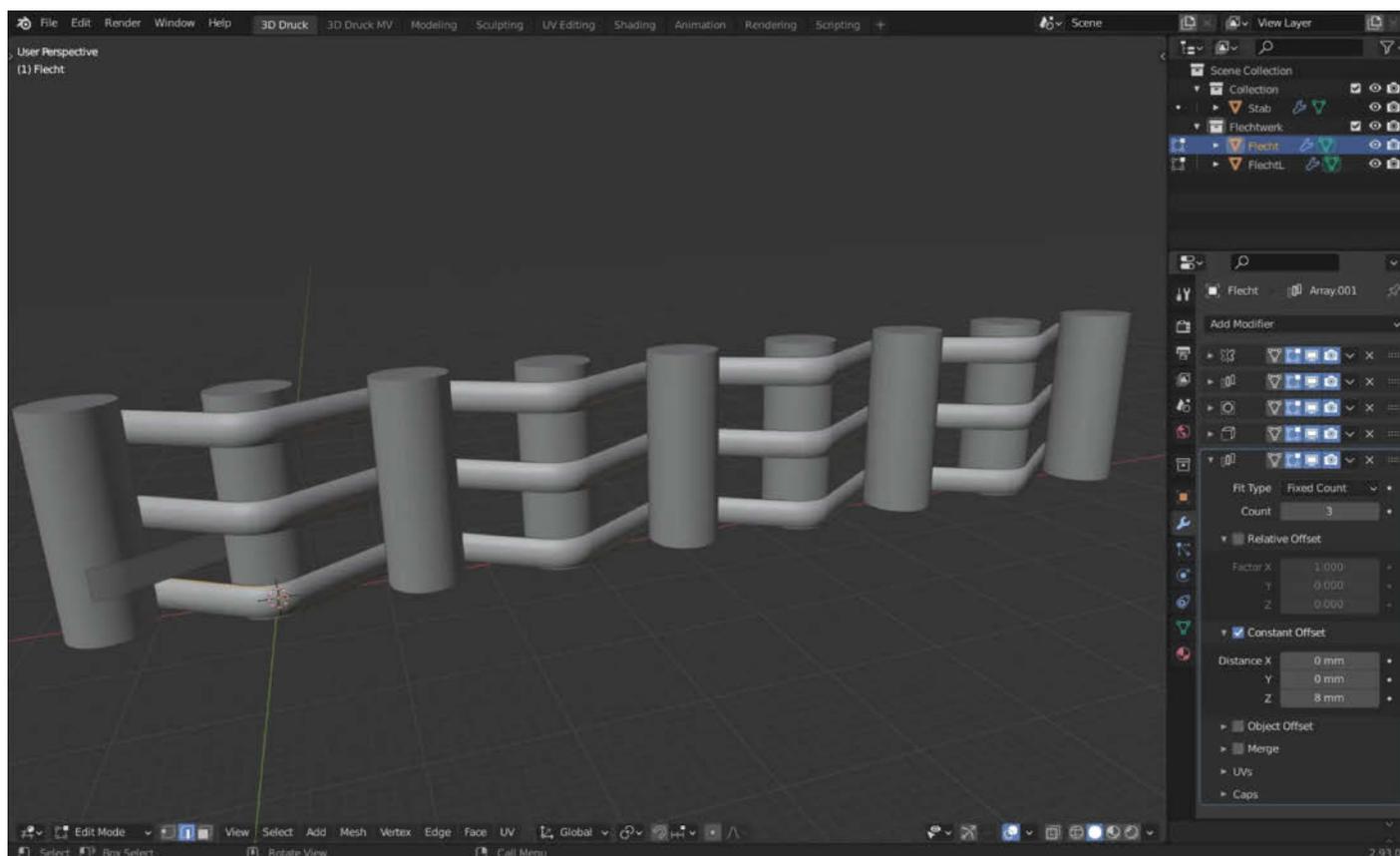
Wir haben die Kickstarter-Kampagne zum *TurtleStitch*-Projekt von Andrea Mayr-Stalder und Michael Aschauer bereits im Jahr 2018 online vorgestellt, seitdem hat sich aber viel getan. *TurtleStitch* hat eine äußerst aktive Community, die sich begeistert gegenseitige Ressourcen zur Verfügung stellt und immer wieder die Limits der Sticksoftware testet. Wer stick- und nähbegeisterten Nachwuchs hat, findet hier einen faszinierenden und niedrigschwelligen Einstieg in die Hardware-Programmierung. —rehu

► turtlestitch.org



Zack Boston





Flechtende 3D-Drucker

Nachdem Make-Redakteur Carsten Wartmann die Flechtstrukturen aus dem 3D-Drucker von Maker *BreaksnMakes* auf Twitter entdeckt hat, musste er diese Technik auch gleich testen – ganz im Sinne unserer Rubrik „Was uns inspiriert“. *BreaksnMakes* hat durch Experimente eine Struktur geschaffen, die an geflochtenes Rattan oder Bambus erinnert. Dieser Effekt ist nicht nur eine Textur auf einem geometrischen Objekt, sondern komplett ausmodelliert.

Hier wird ausgenutzt, dass Filament-Drucker durch *Bridging* Flecht-Strukturen ohne Stützstrukturen aufbauen können. Beim Bridging zieht die Druckdüse das Extrudat zwischen zwei „Brückenpfeilern“ hin und her. Der Faden erkaltet so schnell, dass er kaum durchhängt. Das Objekt, das dabei entsteht, hat interessante mechanische Eigenschaften: Entlang der Fasern ist es sehr flexibel und erlaubt recht kleine Biegeradien, ohne die Struktur zu beschädigen. Bei zu engen Biegeradien knicken die Fasern, brechen aber nicht. Erstaunlich ist auch, dass sich das Teststück kaum in sich verdrehen lässt (Torsion). Dies liegt an den sich kreuzenden Fasern, die wie ein Fachwerk in der Trägerkonstruktion wirken. —caw

► heise.de/-6187361

Was inspiriert Dich?

Wir freuen uns über Vorschläge an:

mail@make-magazin.de



BreaksnMakes

Gervifonn: CD-Cover erkennen mit KI

Futter für das Multiroom-Soundsystem. Ein Raspberry Pi erkennt per Kamera den Musikwunsch und streamt die Musik vom Server auf Musikgeräte.

von Fabian Jansen



Wie viele andere auch, habe ich ein Regal voller CDs. Um den Komfort eines Multiroom-Audiosystems nutzen zu können, habe ich die Musik auf Festplatte kopiert. Sie wird jetzt per *Music Player Daemon* (mpd) auf einem Pi und *Snapcast* in der ganzen Wohnung abgespielt. Während die Bedienung per Smartphone viele Vorteile hat, fehlte mir aber die physische Interaktion mit meinen CDs – per App durch die Sammlung zu navigieren ist anders, als mit den Fingern über die Cover zu streichen, eines herauszunehmen und dann die Musik abzuspielen. Ein Gerät in der physischen Welt musste her!

Konzept

Da ich nicht alle CDs mit einem RFID-Chip versehen wollte, habe ich mich entschieden, das Problem mit einer Kamera und Bilderkennung anzugehen - und fertig war die Idee für das Gervifonn (Isländisch gervigreind - KI und fönn - Plattenspieler). Auf die erste Idee folgte eine Analyse der Anforderungen. Das Gerät muss im Wohnzimmer gut aussehen, es sollte intuitiv bedienbar sein, man sollte einfach erkennen können, was für Musik gerade läuft, es muss sich mit der bestehenden Musiksammlung und -Abspielumgebung verbinden.

Meine Lösung sieht folgendermaßen aus

1. Die abzuspielende CD-Hülle lehnt aufrecht an einer leicht geneigten Fläche. Davor ist eine Kamera positioniert, um ein Bild des Covers aufnehmen zu können. Da zwischen CD-Hülle und Kamera etwas Abstand nötig ist, um das Cover formatfüllend aufzunehmen, ist dazwischen Platz, um liegend ein Display mit Touchscreen anzubringen. Auf diesem wird das aktuell abgespielte Lied angezeigt, auch die restliche Bedienung kann hiermit erfolgen. Alternativ sind auch drei physikalische Knöpfe und ein Drehregler für die Steuerung verbaut.

Die aktuelle Bedeutung der Tasten wird im Display angezeigt. Die Musiksammlung selbst liegt auf einer Festplatte, die mit einem Raspberry Pi verbunden ist. Auf diesem läuft der *Music Player Daemon*, der sich um die Musik- und Playlistverwaltung kümmert und die Audiodaten an einen Snapserver weiterleitet. Der Snapcast-Server streamt die Musik an compatible Clients. In jedem Raum gibt es dann einen Raspberry Pi Zero mit Snapclient und Aktivboxen.

Meine Musiksammlung habe ich mit dem Python-Tool *Jack* (siehe Link) von allen CDs eingelesen, wobei ihm früher die freie Datenbank *FreeDB* (Compact Disc Database, CDDb) die Album- und Tracknamen lieferte. Mittlerweile wurde der Betrieb von *FreeDB* leider eingestellt. Bei meinen neu hinzukommenden CDs erstelle ich die *jack.freedb*-Datei mit den nötigen Daten mittlerweile manuell. Die Musiksammlung ist folgendermaßen organisiert: Es gibt für jeden Künstler einen Ordner mit

Kurzinfo

- » Tensorflow Lite auf dem Raspberry Pi
- » Training mit Data Augmentation
- » Touchscreen und Hardwarebedienelemente verbinden

Checkliste



Zeitaufwand:
1 Wochenende plus Druckzeit



Kosten:
100 Euro



Löten:
Einfache Lötarbeiten



Programmieren:
Python

Material

- » Raspberry Pi 3 oder 4
- » Raspberry Pi Camera
- » 3,5-Zoll Touchscreen
- » 12mm Taster
- » Drehgeber KY-040
- » Drehknopf für Drehgeber
- » Flachbandkabel mit 10-Pin-Stecker
- » MicroSD-Karte
- » Netzteil für Pi
- » Kabel oder Schalllitze
- » M3-Schrauben und -Muttern

Mehr zum Thema

- » Daniel Bachfeld, Einstieg in KI, Make 6/18, S. 36
- » Detlef Heinze, KI für Lego-Roboter, Make 6/18, S. 48

Werkzeug

- » 3D-Drucker
- » Lötkolben, Seitenschneider

Alles zum Artikel im Web unter make-magazin.de/xkf3

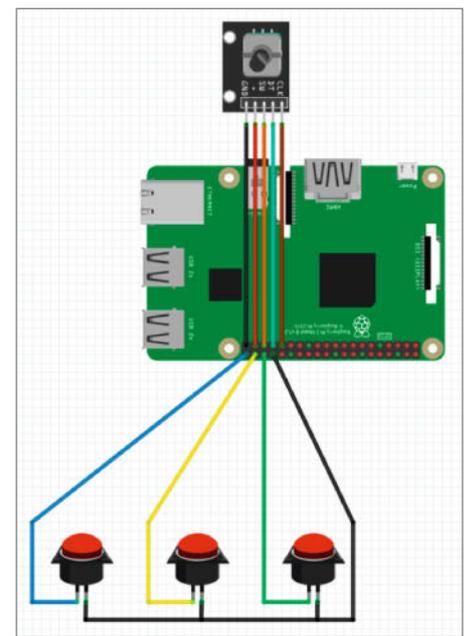
dem Namen des Künstlers oder der Künstlerin. Dieser enthält für jede CD einen Ordner mit dem Namen der CD. Diese Ordner enthalten dann die Musikdateien in einem von MPD unterstützten Format: mp3, ogg usw. Ein Beispiel zur Ordnerstruktur finden Sie auch auf dem Github-Projekt zum Gervifonn.

Aufbau

Im Inneren des Gervifonn werkelt ein Raspberry Pi 3. Als Kamera dient eine Raspberry Pi Kamera. Das Display ist ein 3,5 Zoll großer Touchscreen, welcher direkt auf die GPIO-Leiste aufgesteckt wird. Der Anschluss des



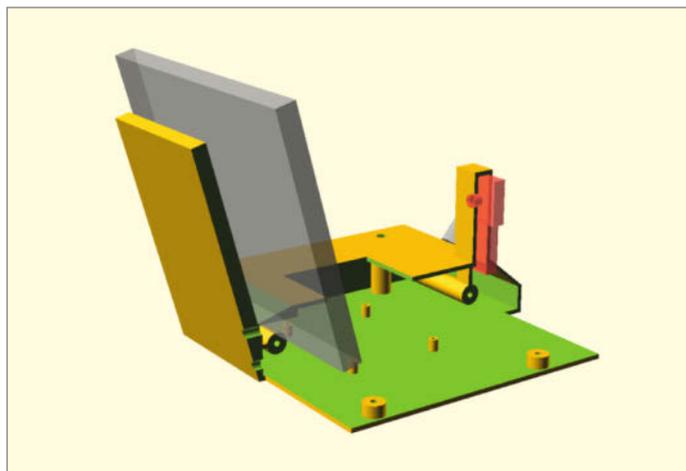
1 Das Gehäusedesign des Gervifonn



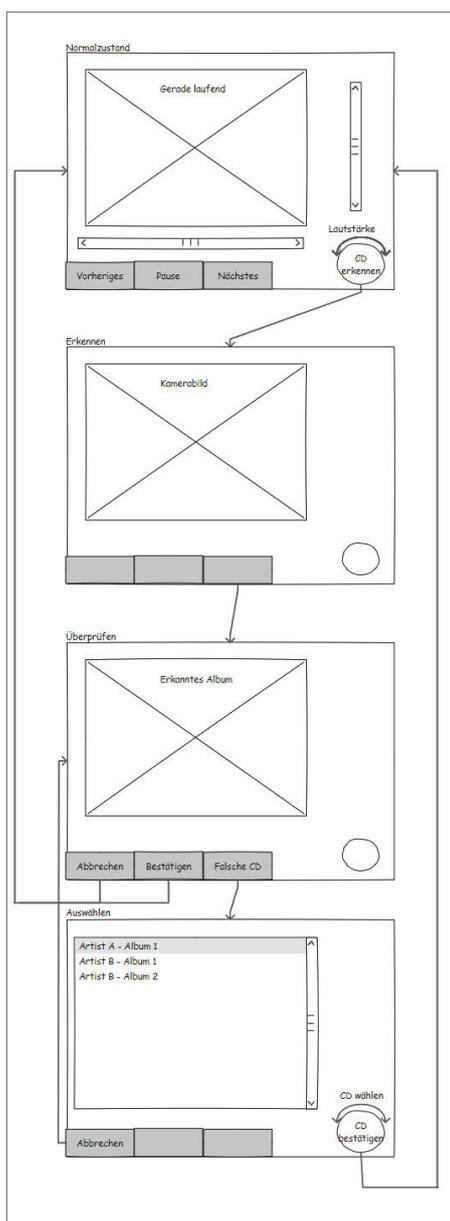
2 Verkabelung der Taster und des Drehgebers



3 Gehäuse des Gervifonn in OpenSCAD mit CD in grau



4 Gehäuse ohne linke Hälfte, in rot die Position des Kameramoduls



5 Bedienkonzept des Gervifonn

Displays lässt die letzten 14 Pins frei, an die letzten 10 Pins sind 3 Buttons sowie ein KY-040-Drehencoder angeschlossen. Die Stromversorgung erfolgt über ein zum Raspberry Pi 3 passendes Steckernetzteil (Micro-USB).

Die Verdrahtung der Knöpfe sowie des Drehencoders ist sehr einfach - ich habe ihn mit einem 10-adrigen Flachbandkabel angeschlossen (Pins 29 und 30 musste ich entfernen und den Stecker oben etwas abflachen, damit er unter das Display passt). Für die Knöpfe habe ich vier der unbenutzten Adern aufgetrennt und an die Knöpfe angelötet (siehe Fritzing-Schaubild 2). Der KY-040 benötigt eine Spannungsversorgung, aber da der Touchscreen keine der spannungsführenden Pins frei lässt, muss ein auf HIGH geschalteter GPIO Ausgang (GPIO 26) aushelfen und die notwendige Spannung liefern. Da der Encoder wenig Strom zieht, ist das kein Problem.

Nach einem Probeaufbau mit einem Gehäuse aus Pappe habe ich das Gehäuse mit OpenSCAD entworfen 3. Die OpenSCAD-Datei sowie die daraus erzeugten STL-Dateien stehen im Projekt-Github zum Download bereit. Das Gehäuse wird in drei Teilen gedruckt und von einigen M3-Schrauben und -Muttern zusammengehalten. Es ist so konstruiert, dass das Kameramodul ohne weitere Schrauben eingeklemmt wird 4. Der Raspberry Pi ist auf der abnehmbaren Bodenplatte verschraubt. Im Gehäuse wäre auch noch genug Platz für eine kleine USB-Soundkarte, um ein Standalone-Gerät zu bauen; eine Kabeldurchführung für passende Aktivboxen ist vorbereitet.

Software

Nachdem die Hardware soweit fertig war, ging es an die Software. Als ersten Schritt habe ich ein Bedienkonzept erstellt (siehe Skizze 5). Das Gervifonn kann sich in vier Zuständen befinden. Im Normalzustand läuft Musik, deren Album-Cover im Display angezeigt wird.

Mit den Knöpfen kann man pausieren, das Lied wechseln und mit dem Drehregler die Lautstärke ändern.

Durch einen Druck auf den Drehencoder startet man den Erkennungsmodus 6. Das Gervifonn macht ein Foto der auf ihm stehenden CD-Hülle und zeigt dieses auf dem Display an, während es versucht das Cover zu erkennen. Sobald die Erkennung abgeschlossen ist, wechselt es in den Überprüfungsmodus. Hier wird das erkannte Cover angezeigt 7. Mit einem Druck auf den mittleren Knopf wird dem Gervifonn mitgeteilt, dass es das richtige Album erkannt hat. Es spielt dieses ab und wechselt zurück in den Normalzustand 8.

Hat das Gervifonn die falsche CD erkannt, kann man mit einem Druck auf den rechten Knopf in den vierten Modus wechseln, den Auswahlmodus. Hier wird eine alphabetisch sortierte Liste aller CDs in der Musiksammlung angezeigt. Mit dem Drehrad kann man auswählen, welche die richtige wäre und diese dann durch einen Druck auswählen und abspielen. Ich habe absichtlich auf einen vollautomatischen Modus verzichtet, bei dem das Gervifonn permanent prüfen würde, ob eine andere als die aktuell laufende CD vor der Kamera steht und diese dann sofort abspielt.

Umgesetzt ist die Software in Python (siehe Github). Die Oberfläche ist mit Hilfe des Moduls *tkinter* programmiert. Beim Start des Programms werden alle Widgets (unsichtbar)

Vorarbeiten

Damit die Software wie beschrieben auf dem Pi funktioniert, sind ein paar Vorinstallationen notwendig. Wie man Snapcast, mpd und CUDA installiert, finden Sie in unseren weiterführenden Hinweisen unter dem Link in der Kurzinfo.



6 GUI: Kamerabild, wird angezeigt während die Erkennung läuft



7 Die CD wurde erkannt und ist zum Abspielen bereit.



8 So sieht die Bedienoberfläche aus, wenn die CD abgespielt wird.

angelegt und die Buttons mit passenden Icons versehen. Für jeden der Hauptbetriebszustände (der Erkennungsmodus verwendet das selbe Layout wie der Überprüfungsmodus) gibt es je eine Funktion, um die für diesen Zustand benötigten Elemente anzuzeigen und wieder zu verstecken.

Bedienung

Für jeden der drei Buttons und für *Drehen* und *Drücken* des Drehencoders gibt es eine Funktion. Diese sind sowohl als Callback an die GPIO-Pins (mittels `GPIO.add_event_detect()`) als auch als `command` an die tkinter-Buttons auf dem Touchdisplay gekoppelt. In jeder dieser Funktionen wird als erstes in einer Fallunterscheidung geprüft, in welchem Zustand sich das Gervifonn befindet. Dann werden die passenden Aktionen durchgeführt. Wenn diese einen Zustandswechsel beinhaltet, wird vorher die Funktion zum Verstecken der GUI-Elemente aufgerufen und abschließend die Funktion, welche die passenden Elemente für den neuen Zustand anzeigt.

Die Steuerung des Music Player Daemons ist mit der Bibliothek `python-mpd2` realisiert, die `snappy`-Bibliothek übernimmt die Lautstärkeinstellung. Im Normalzustand wird der MPD-Server jede Sekunde nach der aktuell laufenden Musik gefragt und das entsprechende Cover sowie der Name des Interpreten, des Albums und des Lieds angezeigt. Mittels zweier Schieberegler wird auch die Lautstärke und Position im Lied angezeigt. Diese können auch per Touchscreen verändert werden.

Hier ergibt sich ein kleines Problem mit der Synchronisation. Das Programm passt die Position der Schieberegler an die aktuelle Abspielposition und Lautstärke an. Ich habe jedoch keinen Weg gefunden, um in tkinter zu unterscheiden, ob der Callback für *Position des Schiebereglers geändert* durch eine Aktion des Nutzers auf dem Touchpad ausgelöst wurde oder vom Programm selbst. Als Workaround wird bei programmatischen Änderungen ein Flag gesetzt. Dieses verhindert im Callback, dass die neue Lautstärke/Abspielposition an

den Server gesendet wird, und wird dann zurückgesetzt.

Ein weiteres Problem ist der Umgang mit Alben, die aus mehreren CDs bestehen. Diese werden von Jack wie mehrere, nicht zusammenhängende CDs behandelt. Hier habe ich immer, wenn nach einer CD gleich die nächste abgespielt werden soll, eine symbolische Verknüpfung mit dem Namen „next“ und dem Ordner der nächsten CD als Ziel angelegt. Wird solch ein Album erkannt, werden immer gleich alle Folge-CDs zur MPD-Playlist hinzugefügt.

Bilderkennung

Um meine komplette CD-Sammlung schnell fotografieren zu können, habe ich ein kleines

Skript (verfügbar auf GitHub unter `src/gervifonn_capture_covers.py`) geschrieben. Es macht bei jedem Druck auf die linke Taste des Gervifonn ein Foto in der für das spätere Training passenden Auflösung, zeigt es zur Kontrolle an, und speichert es im aktuellen Verzeichnis. Man kann es auch mittels

```
DISPLAY=:0 python gervifonn_capture_covers.py
```

per SSH auf dem Gervifonn starten.

Das Herzstück des Ganzen ist natürlich die Bilderkennung. Wahrscheinlich ließe sich diese Aufgabe auch mit der populären Computer-Vision-Software OpenCV und sogenannten Haar Cascades lösen, aber ich wollte ausprobieren, ob das auch mit neuronalen

recognize.py

```
01 from picamera.array import PiRGBArray
02 from picamera import PiCamera
03 import tfLite_runtime.interpreter as tflite
04 import numpy as np
05
06 #TensorFlow Lite initialisieren
07 interpreter = tflite.Interpreter(model_path="gervifonn.tflite")
08 interpreter.allocate_tensors()
09 input_details = interpreter.get_input_details()
10 output_details = interpreter.get_output_details()
11
12 #Kamera initialisieren
13 imgsize = 224
14 camera = PiCamera()
15 camera.resolution = (imgsize, imgsize)
16
17 #Bild aufnehmen und in passendes Format bringen
18 rawCapture = PiRGBArray(self.camera)
19 videoframe = np.asarray(rawCapture.array[:imgsize, :imgsize, :],
20 dtype=np.float32)
21 videoframe = np.expand_dims(videoframe, axis=0)
22
23 #TensorFlow Lite aufrufen
24 interpreter.set_tensor(input_details[0]['index'], videoframe)
25 interpreter.invoke()
26 preds = interpreter.get_tensor(output_details[0]['index'])
27
28 #Album mit der besten Übereinstimmung finden
29 scores = []
30 for (i, pred) in enumerate(preds[0, :]):
31     scores.append([music[i], pred])
32 scores.sort(key=lambda tup: -tup[1])
33 recognized = scores[0][0]
```

generator.py

```

01 from keras.preprocessing.image import ImageDataGenerator
02 from keras.utils import Sequence
03 from keras.utils import to_categorical
04 from keras.preprocessing.image import load_img
05 from imutils import paths
06 import numpy as np
07 import random
08 import os
09
10 batchsize = 2
11 imgsize = 224
12
13 class gervifontrainingdatagenerator(Sequence):
14     def __init__(self, musicfolder):
15         imagePath = list(paths.list_images(musicfolder))
16         self.data = []
17         self.labels = []
18         self.classes = dict()
19         for imagePath in imagePath:
20             if imagePath.endswith('cover.png'):
21                 label = os.path.sep.join(imagePath.split(os.path.sep)[-3:-1])
22                 image = load_img(imagePath, target_size=(imgsize, imgsize))
23                 image = np.asarray(image)
24                 self.data.append(image)
25                 self.classes[label] = len(self.classes)
26                 self.labels.append(self.classes[label])
27         self.data = np.array(self.data, dtype="uint8")
28         self.labels = to_categorical(self.labels)
29         self.datagen = ImageDataGenerator()
30
31     def __len__(self):
32         return int(np.floor(len(self.data)/ batchsize))
33
34     def __getitem__(self, index):
35         indexes = range(index * batchsize:(index + 1) * batchsize)
36         batch_x = []
37         batch_y = []
38         for ind in indexes:
39             batch_y.append(self.labels[ind])
40             image = np.copy(self.data[ind])
41             image[:, :, 0] = (0.7+(0.3*random.random()))*image[:, :, 0]
42             image[:, :, 1] = (0.7+(0.3*random.random()))*image[:, :, 1]
43             image[:, :, 2] = (0.7+(0.3*random.random()))*image[:, :, 2]
44             image = self.datagen.apply_transform(x=image,
45                 transform_parameters={"theta": -20+(40*random.random()),
46                 "tx": (imgsize*((random.random()-0.5)*0.1)),
47                 "ty": (imgsize*((random.random()-0.5)*0.1)),
48                 "shar": -20+(40*random.random()),
49                 "zx": min(1, 0.5+random.random()),
50                 "zy": min(1, 0.5+random.random())})
51             batch_x.append(image)
52         return np.array(batch_x), np.array(batch_y)

```

Netzwerken geht. Das neuronale Netzwerk ist mit *TensorFlow Lite* realisiert. Ich habe für eine schnellere Erkennung auch mit einem *Coral USB Accelerator* mit Edge TPU experimentiert. Letztendlich war der Raspberry Pi 3B+ auch so schnell genug, nach etwa einer Sekunde ist eine CD identifiziert. Mit dem passenden Modell benötigt man nur wenige Zeilen Code, um die Bilderkennung durchzuführen (siehe *recognize.py*).

Nach dem obligatorischen Import der nötigen Bibliotheken (Zeilen 1–4) kann man das *tf-lite*-Modell laden (Zeile 7), Speicher zuweisen (Zeile 8) und den Ein- und Ausgangstensor abfragen (Zeilen 9 + 10). Nach Initialisierung der Kamera (Zeilen 13–15) kann ein Bild aufgenommen werden. Während die reguläre *capture*-Funktion das aufgenommene Bild direkt

in eine Datei schreibt, kann *picamera.array.PiRGBArray* direkt ein *numpy*-Array zurückgeben (Zeile 18). Dieses muss dann noch in die passende Form gebracht werden (Zeilen 19–20), um dann als Eingang für das *TensorFlow Lite*-Modell gesetzt zu werden (Zeile 23).

Nun kann das Modell berechnet (Zeile 24) und das Ergebnis ausgelesen werden (Zeile 25). Das Ergebnis ist ein *numpy*-Array mit den vom Modell ermittelten Wahrscheinlichkeiten, inwiefern das aufgenommene Bild mit den verschiedenen Covers aus der Musikbibliothek übereinstimmt (d.h. der erste Eintrag ist die Übereinstimmung mit dem ersten Cover, der zweite mit dem zweiten Cover, usw.).

Um das wahrscheinlichste Album auszuwählen, wird in Zeilen 28–30 eine Liste mit Titel-Wahrscheinlichkeit-Pärchen erstellt (die

music-Liste enthält alle Alben in der selben Reihenfolge wie die Ausgänge des neuronalen Netzwerks), diese dann nach der Wahrscheinlichkeit sortiert (Zeile 31) und schließlich der erste Eintrag ausgewählt (Zeile 32). Aber woher bekommt man das passende neuronale Netzwerk?

KI-Training

Die Herausforderung bei der Erstellung des Modells ist der Mangel an Trainingsdaten. Normalerweise benötigt man große Mengen an Fotos, um neuronale Netze zur Bilderkennung zu trainieren, ich wollte aber meine CDs nicht in zig Positionen bei verschiedenen Lichtbedingungen fotografieren. Hier kam mir ein Artikel aus der *c't* zum Thema *Data Augmen-*

tation zu Hilfe. Die Grundidee ist, sich selbst weitere, künstliche Trainingsdaten zu generieren. Als Ausgangsmaterial habe ich lediglich ein einziges Foto pro CD mit dem Gervifonn aufgenommen und zusätzlich das Coverbild der jeweiligen CD aus dem Internet verwendet. Diese Bilder werden dann gedreht, vergrößert oder verkleinert, mit einem Farbstich versehen und vor verschiedenen Hintergründen platziert.

Ursprünglich wollte ich komplett auf selbst erstellte Fotos verzichten, aber damit war die Erkennungsgenauigkeit nicht hoch genug – jede zweite CD wurde falsch zugeordnet. Bei über 200 CDs in der Sammlung ist das zwar gar keine schlechte Trefferquote, aber eben überhaupt nicht alltagstauglich.

Zur Erstellung des Modells habe ich das Framework *Keras* verwendet. Der Einfachheit halber habe ich nicht selbst eine Architektur für das Neuronale Netz entworfen, sondern einfach ausprobiert, welches der im Modul *keras.applications* mitgelieferten Modelle die besten Ergebnisse erzielt. Das einzige weitere Auswahlkriterium war, ob das Netz sich auf meiner Grafikkarte trainieren ließ. Ein Training auf der CPU allein dauert zu lange, aber da ich nur eine günstige GeForce GTX 1650 Super mit lediglich 4 GByte Speicher besitze, haben ei-

nige der Modelle nicht in den Grafikspeicher gepasst.

Man kann hierbei auch die Größe der Eingangsbilder variieren, diese hat natürlich einen großen Einfluss auf den Speicherverbrauch, aber eben auch auf die Qualität der Erkennung. Als Alternative zu einem kompletten Training wäre es auch möglich, ein vor-trainiertes Modell als Basis zu verwenden und lediglich die letzte Schicht auszutauschen. Man nennt dies Transfer Learning. Auch das habe ich ausprobiert, aber letztlich waren die Ergebnisse etwas schlechter, und das Training von Anfang an auch innerhalb einer Nacht erledigt. Die besten Ergebnisse habe ich mit dem Modell *DenseNet121* bei einer Bildgröße von 224x224 Pixeln erzielt.

Listings erklärt

Jetzt aber zum Code: Zur Verfremdung der Bilder habe ich einen Generator geschrieben (eine vereinfachte Version ist im Listing *generator.py* abgedruckt, der vollständige Code ist auf github verfügbar). Dieser erbt von der *Sequence* Klasse aus *keras.utils* und muss die folgenden Funktionen bereitstellen: `__len__` liefert die Zahl der Batches und `__getitem__` (N) die Daten von der n-ten Batch. Optional

kann die Funktion `on_epoch_end` implementiert werden, welche eine Veränderung der Trainingsdaten nach jeder Trainingsepoche erlaubt.

Zuerst aber zur `__init__` Funktion. Ich habe die Coverbilder der Alben mit dem Namen *cover.png* in den jeweiligen Ordnern abgelegt. Als erstes sucht der Generator also nach allen Bilddateien (Zeile 15) mit diesem Namen (Zeilen 19+20), und leitet aus dem Pfad den Albumtitel ab (Zeile 21). Dann wird das Bild geladen (Zeile 22) und als *numpy*-Array in `self.data` abgelegt (Zeile 23-24).

Da das neuronale Netz nichts von Albennamen weiß, sondern nur in durchnummerierten Kategorien denkt, wird dann in den Zeilen 25 und 26 dem Album eine Nummer gegeben, und diese Nummer als zu den Daten passend in `self.labels` gespeichert. Diese werden dann mit Hilfe der Funktion `keras.utils.to_categorical` (Zeile 28) zu richtigen Ausgangstensenoren umgewandelt. Nun zum Kern des Generators, der `__getitem__` Funktion: In dieser wird das Bild auf zufällig Art und Weise verfremdet.

Auch hierfür gibt es bereits eine fertige Funktion, nämlich `ImageDataGenerator.apply_transform` aus `keras.preprocessing.image` (Zeile 44). Sie bekommt als

Make:markt

BÜCHER / ZEITSCHRIFTEN



Was Maker schon alles geschaffen haben!

Die Antwort und viele Beispiele finden Leser in unseren Zeitschriften „Space – das Weltraum Magazin“, „Wissen 2021“ und dem „Urknall“ vieler Computer- und Make-Enthusiasten – dem „Retro Gamer“.

www.emedia.de

LED-LÖSUNGEN



LED-Studien.de ist der Spezialist für hochwertige LED-Streifen und Controller.

NEONFLEX, COB-LED, TUNABLE WHITE, RGBW sowie DMX und PIXEL-LÖSUNGEN – auch für große Projekte / komplette Raumbeleuchtungen für Privat und Gewerbe. Wir finden die beste Lösung für Sie!

www.led-studien.de | shop.led-studien.de

MIKROCONTROLLER



Der Verlag für kreative Köpfe!

Informatik und Elektronik können komplex, theoretisch und anstrengend sein. Es geht aber auch einfach, anschaulich und leicht nachvollziehbar – wenn man die Dinge in die eigenen Hände nimmt und zum »Maker« wird: Mit Büchern vom dpunkt.verlag.

www.dpunkt.de



christiani.de/arduino-education

Arduino® Education – bietet ein durchgängiges Lernkonzept mit Soft-, Hardware und Support

Leichter Einstieg in die Elektronik und Programmierung mit spannenden Projekten

- Inkl. Online Lernplattform mit verschiedenen didaktischen Unterrichtseinheiten
- Vermittlung der 21st Century Skills



seit 1931

training.py

```

01 from keras.applications import DenseNet121
02 from keras.optimizers import Adam
03 import tensorflow as tf
04 numepochs = 48
05 imgsize = 224
06
07 generator = gervifonntestingdatagenerator(musicfolder)
08
09 model = DenseNet121(input_shape=(imgsize, imgsize, 3), weights=None, classes=len(generator.classes))
10 model.compile(loss="categorical_crossentropy", optimizer=Adam(), metrics=["accuracy"])
11 model.fit_generator(generator, steps_per_epoch=len(generator), epochs=numepochs)
12
13 converter=tf.lite.TFLiteConverter.from_keras_model(model)
14 tflite_model = converter.convert()
15 with open('gervifonn.tflite', 'wb') as f:
16     f.write(tflite_model)
17
18 with open("labels.txt", "w") as f:
19     for key in generator.classes.keys():
20         f.write("{} {}\n".format(generator.classes[key], key))

```

Parameter die zu transformierenden Bilddaten, und verschiebt, dreht, schert und zoomt diese dann in Abhängigkeit der weiteren Parameter. Diese Parameter werden zufällig gewählt, so dass bei jedem Trainingsdurchlauf ein anderes Bild entsteht und das neuronale Netzwerk lernt, nicht stur auf bestimmte Pixel zu schauen, sondern auf die Struktur der Bilder zu achten.

Wenn man nur diese Art der Transformation anwendet, bekommt man zwar eine super Genauigkeit, aber in der Realität erkennt das Netz alles falsch. Alle geometrischen Transformationen verändern nämlich nicht die Farbe der Bilder. Und gerade bei Albumcovern, die häufig sehr bunt sind, lernt das Netzwerk dann vor allem, auf die Farben zu achten. Bei der späteren Nutzung mit wechselndem Umgebungslicht unterscheiden sich die von der Kamera erfassten Farben allerdings deutlich von den digital originalen der Cover. Deshalb habe ich in den Zeilen 41-43 auch die Farbkanäle der Bilder mit verschiedenen Zufallswerten skaliert, um verschiedene Farbstiche zu erzeugen.

Augmentation

Im vollständigen Code des Generators werden zusätzlich noch zufällig ausgewählte Bilder aus meiner Fotosammlung überlagert und als Hintergrund verwendet, um das Netzwerk noch robuster gegen Bildstörungen zu machen. Außerdem werden nicht nur die Cover, sondern auch (wie bereits oben beschrieben) auf dem Gervifonn aufgenommene Fotos der CD-Hüllen für das Training verwendet. Zur weiteren Verbesserung der Erkennungsqualität wird jedesmal, wenn das Gervifonn eine CD abspielt das zur Erkennung verwendete Foto abgespeichert. Man kann also von Zeit zu Zeit das Modell neu trainieren, und das Gervifonn wird immer besser.

Nun kann man das eigentliche Training durchführen, indem man das Python-Skript mit folgenden Parametern für Ordner, Batchsize, Anzahl der Epochen und Speichernutzung startet:

```
python train_gervifonn.py -m ~/music -b
~/pictures -s 4 -e 64 -d
```

-m gibt den Musikordner an, -b einen Ordner mit Hintergrundbildern, -s die Batchsize, -e die Zahl der Epochen und -d aktiviert die dynamische Speicherallokation.

Auch hier reicht der Platz zur Erklärung nur für den Abdruck einer vereinfachten Version (*training.py*). Zuerst wird der gerade beschriebene Generator initialisiert (Zeile 7). Dann das DenseNet121-Modell geladen (Zeile 9). Es bekommt als Parameter die Größe der Eingangsbilder (*input_shape*) und die Zahl der zu erkennenden CDs (*classes*). Mittels *weights=None* gibt man an, dass man das Netz komplett neu trainieren möchte.

In Zeile 10 wird das Modell für die Optimierung konfiguriert, *loss="categorical_crossentropy"* gibt an, dass der Optimierer (*optimizer=Adam()*) versuchen soll, möglichst eindeutige Zuordnungen zu finden. Nun wird in Zeile 11 die eigentliche Optimierung gestartet, als erstes Argument wird der Generator für die Trainingsdaten übergeben, mit *epochs* wird die Zahl der Trainingsdurchläufe angegeben.

Zu guter Letzt wird das Trainingsergebnis in den Zeilen 13-16 in ein TensorFlow-Lite-Modell umgewandelt und abgespeichert und in Zeilen 18-20 die Zuordnung der Alben zu den Ausgängen des Netzwerks. Im vollständigen Code gibt es noch einen weiteren Generator für Validierungsbilder. Dieser verwendet nur die echten Fotos, nicht aber die Covergrafiken, und verfremdet sie auch nicht. Damit lässt sich dann der Trainingsfortschritt beobachten und bewerten. Bei meiner CD-Sammlung erreicht

das Modell eine Genauigkeit von über 98% – man muss also nur bei jeder 50. CD korrigierend eingreifen!

Um nun das Gervifonn in Betrieb zu nehmen, laden Sie den Code von GitHub auf den Pi herunter. Im folgenden gehe ich davon aus, dass das Projekt in *~/gervifonn* auf dem Raspberry Pi abgespeichert wurde. Nun legen sie das auf dem PC erstellte Modell *gervifonn.tflite* zusammen mit der Datei *labels.txt* in in den Ordner *~/gervifonn/src/ab*.

Mit dem Befehl

```
python gervifonn.py -m ~/music -n
server.local -s server.local -c
Wohnzimmer
```

starten Sie das Skript auf dem Raspi. Die Option -m gibt nen Musikordner an, -n den Hostnamen des MPD-Servers, -s den Hostnamen des Snapcastservers und -c den Namen des Snapcastclients.

Wenn alles funktioniert, können Sie den Raspberry Pi so konfigurieren, dass nach dem Start automatisch ein Nutzer angemeldet wird. Wenn sie die Datei *gervifonn.desktop* nach *~/config/autostart/* kopieren, wird die Software bei jedem Start automatisch geladen. Und fertig ist das Gervifonn!

Ausblick

Ich würde mich freuen, wenn sich jemand zu einem Nachbau oder einer Variante (z.B. für Schallplatten) inspirieren lässt. Auch den Code des Gervifonn kann man sicher noch verbessern. Ich denke auch darüber nach, die Trainingsdaten mit Hilfe einer KI zu erzeugen, so dass nicht mehr jede CD fotografiert werden muss. Jede Weiterentwicklung wird auf meinem Github-Projekt dokumentiert, dort gibt es auch ein Video des Gervifonn in Aktion! —*dab*

// heise devSec()

Die Konferenz für sichere Software- und Webentwicklung

Online am 5. und 6. Oktober 2021

Jetzt
Tickets
sichern!

Sichere Software beginnt vor der ersten Zeile Code ...

Das Programm bietet Vorträge zu folgenden Themen:

- ⊙ Container hinter Schloss und Riegel
- ⊙ Probleme durch Lösungen – Kryptografie in der Praxis
- ⊙ Agile Sicherheitsmodelle und -prozesse in der Zukunft
- ⊙ Authentifizierung einfach und sicher gemacht mit Keycloak
- ⊙ (Irreparable) Datenschutz-Fehler vermeiden

In der Keynote beschäftigt sich Bianca Kastl ausführlich mit den Schwachstellen der Luca-App.

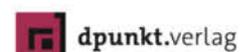
Außerdem: Workshops zu Web Application Security, Threat Modeling, Java Security und Authentifizierung mit Keycloak

Jetzt anmelden: www.heise-devsec.de

Goldsponsor



Veranstalter



Pixelart mit Pi

„Kind, Du verdirbst dir die Augen“, hörte man früher, wenn man auf flimmernde, pixelige Anzeigen starrte. Heutzutage sind flimmernde, pixelige Displays en vogue und stehlen Monitoren und TVs mit 4K-Auflösung locker die Show. Wir zeigen, wie man Bilder und animierte GIFs auf LED-Panels wiedergibt.

von Daniel Bachfeld



Man kennt sie vom Times Square, von Konzerten und aus Fußballstadien: Großdisplays oder Videowände mit scrollenden Texten, bunten Bildern und Videos. Sie bestehen aus vielen zusammenschalteten LED-Panels, die zu einer großen Anzeigetafel zusammenschaltet werden. Die Abstände der Pixel zueinander liegen je nach Modell zwischen 2mm und 6mm. Ab einer gewissen Entfernung nimmt das menschliche Auge die einzelnen Pixel kaum noch wahr.

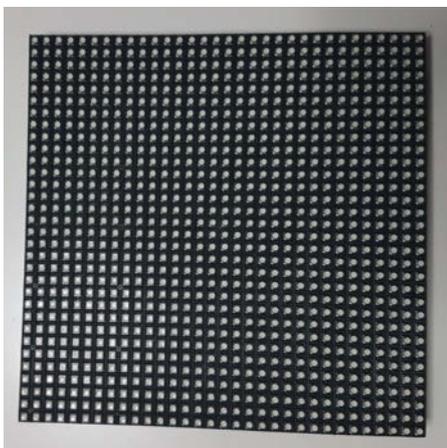
In diesem Artikel geht es allerdings nicht darum, Großdisplays für die Gartenparty zu bauen, sondern einzelne Panels für schicke Leucht-Projekte einzusetzen **1**. Die Panels eignen sich nämlich prima, um kleine, grob-pixelige Anzeigen zu bauen, die im Wohnzimmer, Hobbykeller, Büro, Arcade Hall oder Geschäft den Retrocharme der 8-Bit-Heimcomputer-Ära versprühen. Damit lassen sich Grafiken und animierte GIFs darstellen. Und um es mal mit Lorient's Worten zu sagen: „Da ist dann immer ein großes Hallo...“

Die gängigsten Panels werden mit 32 × 32 RGB-Pixel hergestellt, wobei die Abstände (*Pitch*) zwischen den Pixeln zwischen 2mm und 6mm liegen. Es gibt sie aber auch mit 64 × 32 Pixeln. Die Panels bringen keinerlei Eigenintelligenz mit und müssen von einem externen Controller gesteuert werden **2**. Wie die Panels arbeiten, lesen Sie im Kasten *Ansteuerung*.

Während Großdisplays mit einem PC-Controller angesteuert werden, genügt für einzelne Panels ein Mikrocontroller. In diesem Projekt haben wir uns für den Raspberry Pi entschieden, der mit einem speziellen *RGB Matrix HAT* von Adafruit und freier Software die Ansteuerung übernimmt. Es gibt aber auch Lösungen für Teensy-, ESP32- und Arduino-Boards.

Aufbau

Das *RGB Matrix HAT* wird als vorbestückte Platine geliefert, auf der noch Stift- und Steckerleisten sowie der Stromanschluss nachzulöten



1 Ein LED-Panel mit 32 × 32 Pixel und 5mm Pitch

Kurzinfo

- » So funktionieren LED-Panel
- » Ansteuerung mit Pi und RGB Matrix HAT
- » Bilder und GIFs aufbereiten

Checkliste



Zeitaufwand:
2 Stunden



Kosten:
120 Euro

Material

- » Raspberry Pi 2, 3 oder 4 plus Netzteil, SD-Karte etc.
- » Adafruit 32 × 32 RGB-Panel, 5mm
- » Netzteil 5V, 4A, Hohlbuchse
- » Adafruit RGB Matrix Panel

Werkzeug

- » Lötkolben
- » Schraubenzieher

Mehr zum Thema

- » Peter Tschulik, LED-Laufschrift mit ESP32, Make 4/21, S. 8

Alles zum Artikel im Web unter make-magazin.de/x3dc

sind **3**. Sofern man mit Panels der Größe 64 × 64 arbeiten will, muss man noch eine Brücke auf der Unterseite der Platine löten. Genaueres dazu findet sich in der Aufbauanleitung von Adafruit (siehe Link in der Kurzinfo). Zusätzlich kann man optional die Pins 4 und 18 mit einer Drahtbrücke verbinden, das reduziert später in Zusammenhang mit der Software das Flimmern auf dem Panel. Das HAT enthält noch einen IC mit einer Echtzeituhr (RTC) sowie einen Batteriehalter. In dieser Anleitung nutzen wir diese Funktionen jedoch nicht.

Ist das HAT fertig gelötet, steckt man es auf den Raspberry. Wir nutzen hier einen Pi 3. Ältere Pi-Versionen lassen sich ebenfalls einsetzen, sofern sie eine 40polige Leiste aufweisen. Zwischen dem HAT und dem HDMI-Anschluss haben wir zur Sicherheit noch ein Stück Isolierband geklebt **4**. Das Panel wird mit dem mitgelieferten Flachbandkabel verbunden **5**. Zusätzlich schließt man das Stromversorgungskabel an die Klemmleisten am HAT und am anderen Ende am Stecker am Panel an **6**. Achten Sie auf die richtige Polarität **3**.

Das HAT wird über eine Hohlbuchse mit Strom versorgt. Zum Betrieb eines Panels reicht ein Netzteil mit 5V und 2A, 5V mit 4A bieten jedoch mehr Reserve, insbesondere wenn ein zweites Panel dazu kommt. Das HAT versorgt den Pi über die Steckerleisten mit Strom, wenn kein Netzteil am MicroUSB angeschlossen ist. Im Test mit einem Netzteil mit 5V und 4A beklagte ein Pi 3 des öfteren eine zu niedrige Spannung, wenn nur das Netzteil am HAT angeschlossen war. Durch den Anschluss eines Pi-Netzteil verschwanden die Meldungen.

Software

Für die Auskennung: Das HAT enthält neben der RTC noch zwei Treiber-Chips (74AHCT245, 8-fach Transceiver), die als Pegelwandler von 3,3V auf 5V fungieren. Man kann sich solch ein HAT selbst auf einem Breadboard oder einer Lochrasterplatine zusammenbauen, sollten die rund 27 Euro für das HAT und Kabel zu viel erscheinen. Der Autor der im Folgenden erklärten Ansteuerungssoftware, Henner Zeller,



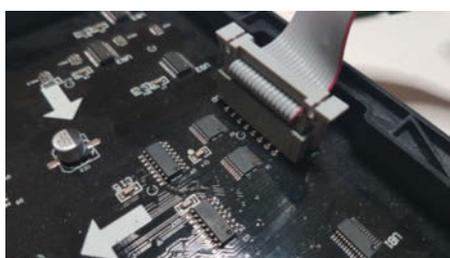
2 Ein Panel von hinten: Die jeweils drei links oben, links unten, rechts oben und rechts unten angeordneten ICs sind Schieberegister mit 16 Ausgängen. Im unteren und oberen Segment jeweils 2 × 16 Ausgänge für je R, G und B in den 32 Spalten. An der Eingangsbuchse ist ein Pegelwandler und rechts daneben sind zwei Multiplexer zum Steuern der jeweils 16 Zeilen im oberen und unteren Segment.



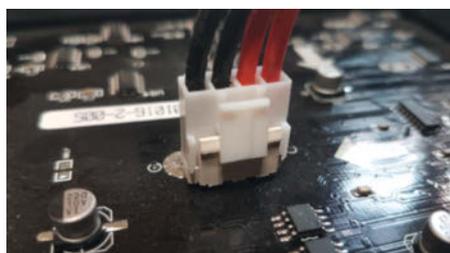
3 Auf der Platine muss man die Steckerleisten und die Klemmleiste selbst anlöten.



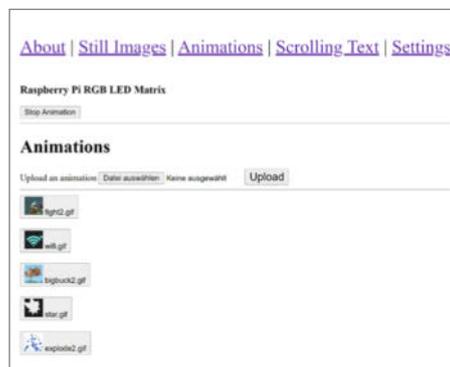
4 Bevor man das HAT auf den Pi steckt, isoliert man den HDMI-Anschluss gegen Kontakte von oben.



5 Das Flachbandkabel steckt man in die mit Input bezeichnete Steckerwanne. Eine Kerbe im Stecker verhindert das Verpolen.



6 Das Stromkabel ist zumindest am Stecker verpolungssicher. Am HAT müssen Sie die Kabel wie in Bild 3 anschließen.



7 Nicht schön, aber es funktioniert: Bilder lassen sich mit dem Browser hochladen und durch Klick auf dem Panel anzeigen.

hat im dazugehörigen Github (siehe Link) einige Vorschläge für den Selbstbau nebst Platinedateien veröffentlicht.

Adafruit stellt für sein HAT ein Skript zum Download bereit, das seinerseits die Software aus dem Github *hzeller/rpi-rgb-led-matrix* lädt, sie entpackt, Abhängigkeiten installiert und den C-Quellcode übersetzt. Man lädt und startet das Skript mit:

```
curl https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/master/rgb-matrix.sh > rgb-matrix.sh
```

```
sudo bash rgb-matrix.sh
```

Nach dem Start stellt es dem Nutzer ein paar Fragen, etwa zum Modell des HATs und ob man die Pins 4 und 18 kurzgeschlossen hat oder erstmal ohne dieses Feature arbeiten will. Im Anschluss stehen auf dem Pi im Ordner */home/pi/rpi-rgb-led-matrix* diverse Tools und Demos bereit. Für einen ersten Test wechselt man in den genannten Ordner und von dort in den Unterordner

```
cd example-api-use
```

Hier befindet sich ein Demo-Tool, mit dem sich das Panel ansteuern lässt.

```
sudo ./demo -D 9
```

startet Demo Nummer 9 von insgesamt 11 verfügbaren: *Volumebars*. Das sudo ist notwendig, um das Programm mit Root-Rechten auszuführen und so Zugriff auf bestimmte Hardwareregister zu erhalten. Um sich sudo künftig zu sparen, macht man Root zum Besitzer des Programms und setzt das Setuid-Bit, mit dem es mit den Rechten des Besitzers statt des Ausführenden gestartet wird:

```
sudo chown root ./demo
sudo chmod +s demo
```

Spannender als die eingebauten Demos zu starten, ist die Anzeige eigener Bilder und insbesondere animierter GIFs. Das Tool *led-image-viewer* macht dies möglich und findet sich unter */home/pi/rpi-rgb-led-matrix/utils*. Leider hat das obige Skript von Adafruit diesen Ordner beim Bau der Tools nicht beachtet, sodass man die Software nachträglich manuell übersetzen muss. Das ist aber in zwei Schritten erledigt. Zuerst installiert man noch zwei nötige Pakete:

```
sudo apt-get install libgraphicsmagick++-dev libwebp-dev -y
und übersetzt den Quellcode im Ordner mit
make led-image-viewer
```

Im Ordner liegt nun die ausführbare Datei *led-image-viewer*, die sich mit

```
sudo ./led-image-viewer
```

starten lässt. Will man sich auch hier sudo sparen, muss man analog zum Programm *demo* den Besitzer und Setuid ändern.

Das Programm erwartet aber noch ein paar Optionen, insbesondere welche Art von Adapter es ansteuert, weil diese unterschiedliche Belegungen der GPIOs nutzen. Mit

```
sudo ./led-image-viewer --led-gpio-mapping=adafruit-hat bild.jpg
```

konfiguriert man die Software für das Adafruit-HAT. Es zeigt auf dem Panel das Bild *bild.jpg* an, welches im gleichen Ordner wie das Programm *led-image-viewer* liegt. Sind die Bilder unter einem anderen Ordner gespeichert, muss man den Pfad dorthin entsprechend angeben. Das Bild muss für erste Versuche noch nicht herunterskaliert sein, das übernimmt die Software automatisch. Besser ist jedoch ein Bild selbst mit 32 x 32 Pixel oder vielfachen davon zu erzeugen.

Ansteuerung

Die LED-Panels bringen keinerlei Controller mit und müssen durch eine externe Logik gesteuert werden. Bei dem in diesem Artikel verwendeten Panels von 32×32 Pixel sind insgesamt 1024 RGB-LEDs anzusteuern. Bei drei Farben also insgesamt 3072 LEDs, die man aus ökonomischen Gründen natürlich nicht einzeln mit einem Controller ansteuert. Stattdessen setzt man auf eine Multiplex-Technik, bei der die einzelnen LEDs aus einer Matrix von mehreren Zeilen und Spalten angesteuert werden: Die Anode einer LED hängt beispielsweise an einer Spaltenleitung und die Kathode an einer Zeilenleitung. Indem man nun beispielsweise Spalte 10 auf HIGH legt und Zeile 8 auf LOW, leuchtet die entsprechende LED. Allerdings hätte man immer noch $32 + 32$ Leitungen, die ein Controller steuern musste.

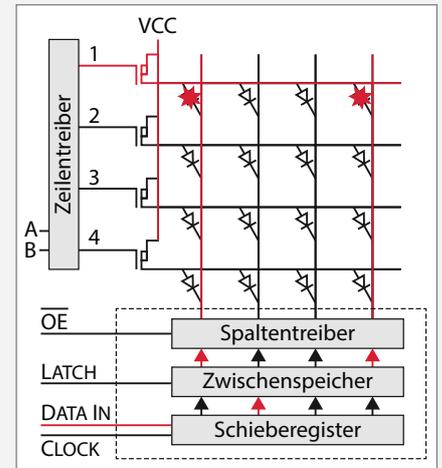
Um weitere Leitungen eines externen Controllers zu sparen, nutzt man sogenannte Schieberegister. In den Panels sind in der Regel 16-Bit-Register verbaut. Über deren seriellen Eingang kann man taktgesteuert einzelne Bits übertragen, die das Schieberegister an seinen parallelen Ausgängen wieder ausgibt **8**. Taktet man 16 Bits nacheinander hinein, stehen anschließend am Parallelport alle 16 Bits parallel zur Verfügung. Neben den seriellen Eingängen haben Schieberegister-ICs auch noch serielle Ausgänge. Auf diese Weise kann man mehrere Register hintereinander schalten, um mehr als 16 Bit parallel auszugeben, etwa 32, 48, 64 usw. Bei einer Spaltenbreite von 32 Pixeln benötigt man pro Farbe je zwei Schieberegister in Reihe, also 2×16 und das für jeweils R, G und B.

Für die Zeilensteuerung nimmt man sogenannte Multiplexer-ICs: Sie haben mehrere Adresspins, deren Logikpegel bestimmen, welcher ihrer Ausgabepins auf HIGH oder LOW geht. Auch sie kann man hintereinander schalten, sodass man die Zeilen des Panels einzeln ansteuern kann. Durch

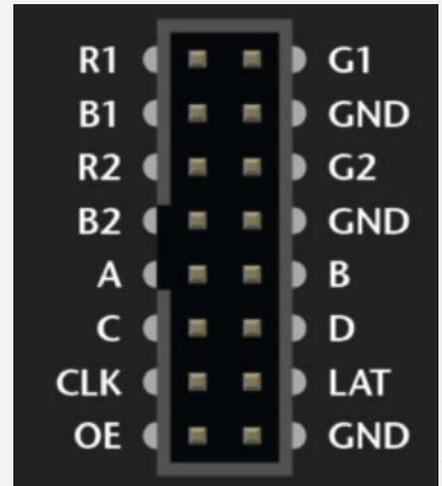
geschickte Ansteuerung aller Spalten und Zeilen nacheinander und in schneller Folge kann man ein vollständiges Bild auf das Panel zaubern. Das menschliche Auge nimmt aufgrund seiner Trägheit die sequentielle Ansteuerung wie beim analogen TV kaum war. Im vorliegenden 32×32 Panel sind die LEDs als Segmente zu zweimal 32×16 realisiert, ein Segment ist oben, eines unten. Während jedes Segment für jede Farbe R, G und B einen eigenen Eingang hat, teilen sie sich die Ansteuerung der Zeile, das heißt in jedem Segment ist immer die gleiche Zeile aktiv. Bei den Zeilen spricht man auch von *Scan-Lines*. Mit zwei aktiven Scan Lines und je 32 RGB-LEDs leuchten 64 Pixel. Bei 1024 Pixel entspricht das einem Verhältnis von 1 : 16, weshalb man auch von einer *Scan Rate* von 1 : 16 spricht.

Man könnte auch die LEDs als vier Segmente zu 32×8 organisieren oder ansteuern, bei denen dann jeweils 4 Zeilen gleichzeitig aktiv sind. Dann läge die Scan Rate bei 1 : 8. Anders als bei RGB-LEDs vom Typ WS2812x respektive *NeoPixel* kann man bei LED-Panels die Helligkeit der einzelnen RGB-Farben nicht angeben. Eine Farbe ist entweder an oder aus. Durch gezielte Aus- oder Anzeiten einzelner Farben kann man aber dennoch die Helligkeit variieren und damit auch die Farben der Pixel variieren.

Obwohl es keinen offiziellen Standard für LED-Panels aus China gibt, hat sich hinsichtlich der physischen Schnittstelle als auch des Protokolls ein Konsens namens *HUB75* etabliert. Gängige Panels haben eine 16-polige Pfostenleiste als Eingang **9**. 32×32 -Panels haben je einen seriellen Eingang für jede Farbe und jedes Segment, hier R1 und R2 für Rot, G1 und G2 für Grün und B1 und B2 für Blau. Die Eingänge A, B, C und D adressieren die Zeilen. Die restlichen Pins (CLK, LAT und OE) sind Takt- und Steuerleitungen.



8 Über die serielle Schnittstelle taktet man die Bits für die jeweiligen Spalten in das Schieberegister. Der Zeilentreiber ist ein Multiplexer, an dessen Eingang eine Adresse im Binärformat ansteht.



9 Ein HUB75-Anschluss. Die Bezeichnungen variieren, manchmal heißen die Farbeingänge auch R0 und R1 usw. Zudem sind je nach Panelgröße mal mehr oder mal weniger Adresspins (A, B, C, D, E) belegt.

Für erste Versuche haben wir in unserem Github für Sie einige animierte GIFs in 32×32 und 64×32 (falls Sie gleich zwei Panels haben) sowie PNGs hinterlegt. Weitere Bilder, die sich automatisch herunterskalieren lassen und sich gut auf dem Panel machen, finden Sie im Internet zuhause, etwa auf *DevianArt*, *pixilart.com*, *pixelartmaker.com* und vielen Plattformen

mehr. Erste Anlaufstelle für weitere animierte GIFs zu jedem Thema dürfte die Online-Sammlung *Giphy* sein.

Bilder selbst zurechtschneiden und skalieren kann man mit Bildbearbeitungsprogrammen oder online beispielsweise mit der Seite *ezgif.com*. Sie bietet zahlreiche Funktionen wie Crop, Resize, Video to GIF und unterstützt

verschiedene Formate sowie animierte GIFs und das moderne Format WEBP von Google.

Optionen

Sollte die Software beim Start wegen Timing-Problemen meckern, fügen Sie dem obigen Befehle die Option `--led-no-hardware-pulse`



hinzu. Strahlt das Bild zu grell, dimmt man mit der Option `--led-brightness=50` auf 50 Prozent Helligkeit oder beliebige andere Werte herunter. Eine vollständige Liste aller Optionen finden Sie im Github-Repo des Tools. Apropos dimmen: Anders als bei NeoPixels oder WS2812B-LEDs können bei LED-Paneln die LEDs nicht einzeln per PWM in der Helligkeit gesteuert werden.

Standardmäßig ist die Auflösung auf 32×32 eingestellt. Haben Sie ein Panel mit 64×32 Pixel, starten Sie die Software mit `--led-cols=64`. Haben Sie 2 Panels á 32×32 hintereinander kaskadiert, so übergibt man mit `--led-chain=2` die Anzahl der Panels. Im Endeffekt kommt aber ebenfalls eine Auflösung von 64×32 heraus. Der gesamte Befehl sieht dann so aus:

```
sudo ./led-image-viewer --led-chain=2
--led-gpio-mapping=adafruit-hat
--led-brightness=50 --led-no-
hardware-pulse giphy.gif
```



wobei *giphy.gif* in diesem Beispiel ein 64×32 kleines animiertes GIF ist.

Fernbedienung

Damit Sie zur Anzeige eines Bildes nicht jedes Mal eines der Softwaretools direkt auf dem Pi bedienen müssen, gibt es auch eine praktische Bedienung per Webserver. Mit einem Browser können Sie neue Bilder hochladen und auswählen, welche angezeigt werden sollen, oder die Anzeige stoppen.

Der Server und die App funktionieren aktuell jedoch nur für einzelne Panels. Kaskadiert man beispielsweise mehrere 32×32 -Panel, so wird auf allen Paneln jeweils die gleich 32×32 Grafik dargestellt. Grafiken mit 64×32 werden so skaliert, dass sie auf ein 32×32 Panel passen.

Alternativ kann man sich bei Headless-Konfigurationen ohne angeschlossenen Monitor auch per SSH auf dem Pi einloggen und etwa in einer *tmux*-Shell die Tools mit den erforderlichen Parametern starten. Das funktioniert auch per VNC, sofern aber kein Monitor angeschlossen ist, muss man noch per *raspi-config* eine Grafikauflösung einstellen. Andernfalls zeigt der VNC-Client später keinen Desktop an.

Zunächst stellen Sie sicher, dass Java auf dem Pi installiert ist. In einer Shell sollte der Befehl `java --version` eine Versionsnummer ausspucken. Kommt eine Fehlermeldung, muss man mit `sudo apt install default-jdk` Java nachinstallieren.

Anschließend besorgt man sich in der Shell mit

```
wget https://s3-eu-west-1.amazonaws.com/payara.fish/
Payara+Downloads/5.2021.6/payara-
micro-5.2021.6.jar
```

den Java-Server *Paraya* sowie die Webapp

```
wget https://cdn-learn.adafruit.com/
```

```
assets/assets/000/040/940/original/
rpi-rgb-led-matrix-webapp-0.0.1-
SNAPSHOT.war
```

Beides muss im gleichen Ordner liegen. Der Befehl

```
java -jar payara-micro-5.2021.6.jar
--deploy rpi-rgb-led-matrix-webapp-
0.0.1-SNAPSHOT.war
```

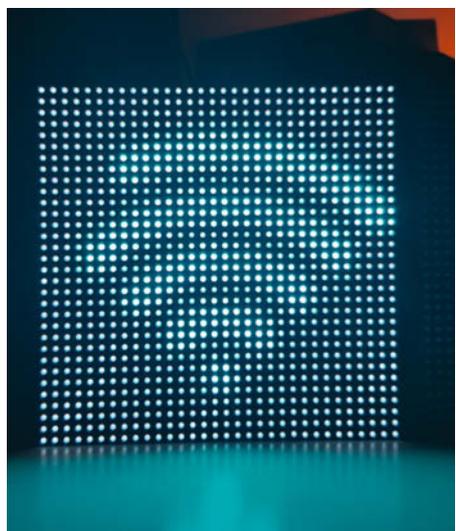
startet den Server und die WebApp, was je nach Pi-Modell bis zu zwei Minuten dauern kann. Anschließend ruft man in einem Browser auf einem anderen PC folgende Zeile mit der für Sie richtigen IP auf:

```
http://192.168.2.144:8080/rpi-rgb-
led-matrix-webapp-0.0.1-SNAPSHOT/
```

Die sich öffnende Webseite wirkt wie aus der GUI-Hölle Ende der 90er des letzten Jahrtausends, ist aber dennoch übersichtlich und intuitiv **7**. Unter *Settings* müssen die Pfade zu den Tools und den Ordnern der Bilder eingetragen werden. Der Punkt *Still Images* dient dem Hochladen und der Auswahl von Standbildern (bei uns funktionierten nur Bilder im PNG-Format), analog dazu *Animations* für animierte GIFs, die in der Übersicht auch animiert angezeigt werden. Ein Klick, aber bitte nur einen Klick, auf das gewünschte Bild und schon wird es angezeigt. Mehrere Klicks schnell hintereinander bringen unter Umständen den Server durcheinander, der mehrere Prozesse des *led-image-viewer* spawnt, was zu Fehler führt.

Ausblick

Um das Projekt wirklich wohnzimmertauglich zu machen, sollte man dem Panel noch einen Rahmen und eine Halterung für den Pi spendieren. Sie können es aber auch in bestehende Möbel einbauen. Auch als animiertes Marquee oder Side Art in Arcade-Gehäusen sehen die Panels fantastisch aus. —*dab*





Die Heise-Konferenz für Speichernetze und Datenmanagement

3 x online im Herbst 2021

Auch in diesem Herbst bieten wir Ihnen wieder aktuelles Storage-Wissen. Besuchen Sie unsere Online-Events zu praxisrelevanten Themenschwerpunkten:

- ☑ **27. Oktober:**
Der Storage Strategy Day
- ☑ **30. November:**
Der Backup & Storage Security Day
- ☑ **07. Dezember:**
Der Storage Technology Day

**Neu: Zu jeder Veranstaltung stehen
Freitickets zur Verfügung.**

Jetzt
**Freitickets
sichern!**

www.storage2day.de

Veranstalter



dpunkt.verlag

Goldsponsoren

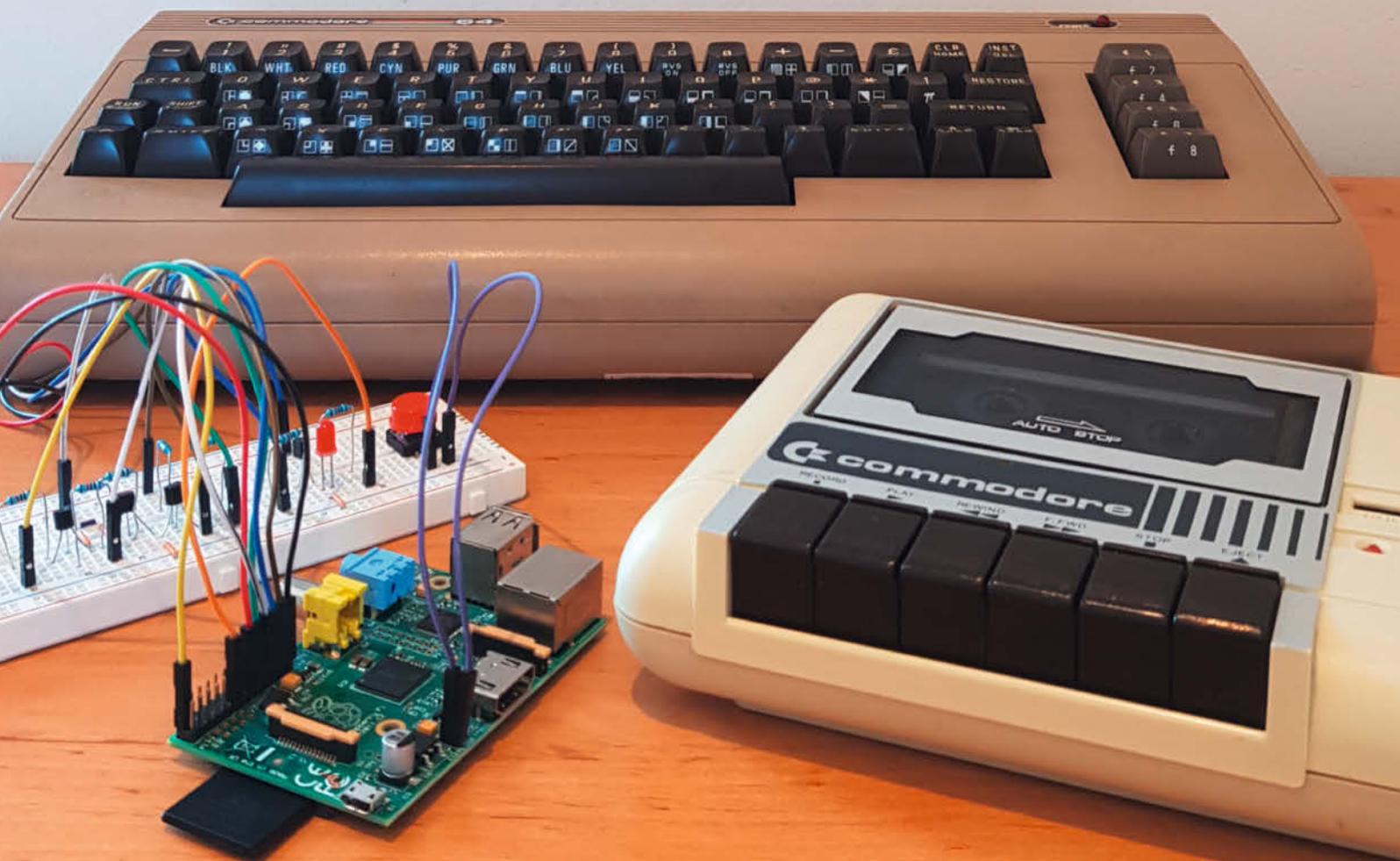


© Copyright by Maker Media GmbH.

Raspberry Pi als SD-Karten-Laufwerk für C64 & Co.

Die 8-Bit-Homecomputer der 80er Jahre wie C64 oder VC20 von Commodore sind voll retro und immer noch bei vielen Fans in. Vielen gut erhaltenen Geräten fehlt jedoch der Massenspeicher, die Datasette oder das Floppy-Laufwerk, um die Programm- und Datenflut zu archivieren. Passende Speichermedien sind auch rar geworden. Ein Raspberry Pi kann diesen Engpass beseitigen.

von Marcel Timm



Die Beliebtheit der 8-Bit-Home-Computer der 1980er Jahre ist in letzter Zeit immer weiter angestiegen. Auch für die Commodore Computer - insbesondere den C64 - ist das Interesse groß. Sei es, um die Spiele von damals wiederzuentdecken (nicht nur am Emulator, sondern wie früher am Fernseher mit Joysticks usw.) oder weil man die Systeme als mittlerweile Erwachsener nun einmal besser verstehen, eventuell sogar neue Spiele oder Ähnliches entwickeln möchte.

Kommunikation mit einem Home Computer

Ob es nun Programme aus dem vorigen Jahrhundert oder neue Retro-Entwicklungen sein sollen, die Daten müssen irgendwie auf den alten Rechner gebracht werden. Dazu gibt es verschiedene Möglichkeiten. Wenn einem Datasette und Diskettenlaufwerk zu langsam sind oder einfach nicht (mehr) zur Verfügung stehen, kann man auf diverse Speziallösungen zurückgreifen.

Massenspeicher-Alternativen für Commodore-Computer

Für den C64 gibt es die größte Auswahl an Speichermöglichkeiten. Von der einfachen Lösung mit SD-Karten *SD2IEC* bis zum *Ultimate-II+*, welches das Floppy-Laufwerk *VC1541* **1** komplett simuliert. Das Projekt *Pi1541* verwendet sogar ebenfalls einen Raspberry Pi, hier für die Simulation des Diskettenlaufwerks.

Die bereits genannten Lösungen verwenden alle den seriellen Port (für Diskettenlaufwerke) und belegen meist noch weitere Anschlüsse, am Expansion Port und/oder Datasettenanschluss, die dann nicht mehr für weitere Geräte zur Verfügung stehen. *TapeCart* ist ein Projekt, das nur den Kassettenanschluss belegt **2**. Für die Vorgänger von *VC20* und *C64*, die *CBM/PETs*, gibt es beispielsweise *petSD*.

CBM Tape Pi

Die in diesem Artikel besprochene Lösung *CBM Tape Pi* hat die Vorteile des Open-Source-Quell-Codes, die notwendige Hardware ist minimal (keine ICs, nur diskrete Bauteile) und dementsprechend sind die Kosten sehr gering. Außer am Stecker muss auch nichts gelötet werden. Außerdem wird nur der Datasettenanschluss des Home-Computers belegt.

Für den Raspberry Pi reicht ein alter Pi 1, 2, 3, Zero, o.ä., wie sie in vielen Bastler-Schubladen herumliegen und im Internet günstig gebraucht zu erstehen sind. Ein weiterer Pluspunkt von *CBM Tape Pi* ist der gleichzeitige Support von C64, VC 20 und CBM/PETs. Natürlich kann *CBM Tape Pi* nicht nur Daten zum Commodore senden, sondern auch Programme auf der SD-Karte speichern.

Kurzinfo

- » SD-Karte als Speichermedium für Programme und Daten
- » geeignet für Commodore VC 20, C64 und CBM/PET
- » Befehlsweiterung für schnelleren Daten-/Programmtransfer

Checkliste



Zeitaufwand:
2 Stunden



Kosten:
30 Euro (zzgl. Raspberry Pi)

Mehr zum Thema

- » Winnie Forster, Die vergessenen Heimcomputer der 80er Jahre, heise+
- » Michael Steil, Brotkasten für die Welt, c't retro 2018, S. 80
- » Mirko Dölle, Anschlusshilfen, c't retro 2018, S. 166

↓ Alles zum Artikel im Web unter [make-magazin.de/x9fn](https://www.make-magazin.de/x9fn)

Material

- » Raspberry Pi (1, 2, 3 oder Zero - andere funktionieren evtl. auch)
- » Netzteil (passend für den RasPi)
- » LED (5mm, rot)
- » 3 NPN-Transistoren BC547 oder PN2222A (T2, T3 und T4 im Schaltplan)
- » PNP-Transistor BC557 (T1 im Schaltplan)
- » 10 Widerstände 10 Kiloohm
- » Stecker für Commodore-Kassettenport (12polig)
- » SD-Karte (passend zum Raspberry Pi)
- » Breadboard
- » Jumperkabel
- » optional: Taster als Reset-Button für den RasPi

Werkzeug

- » Lötkolben

Zusammenbau

Die Schnittstellen-Schaltung kann einfach auf einem Breadboard/Steckbrett aufgebaut werden **3**.

Der Schaltplan **4** zeigt, wie die Teile miteinander verbunden werden müssen. Alternativ kann man eine Platine verwenden, wie sie von *Sven Petersen* entwickelt wurde.

Erläuterungen und Grafiken zu den Positionen der verwendeten GPIO-Anschlüsse sind online beispielsweise unter

<https://pinout.xyz>

die Positionen der Datasetten-Port-Anschlüsse unter

- 2** Die Schnittstellen an der Rückseite des C64 (von links): Expansionport, Antennenausgang, Video-/Tonausgang, serieller Port für Floppy/Drucker, Kassettenport, Userport

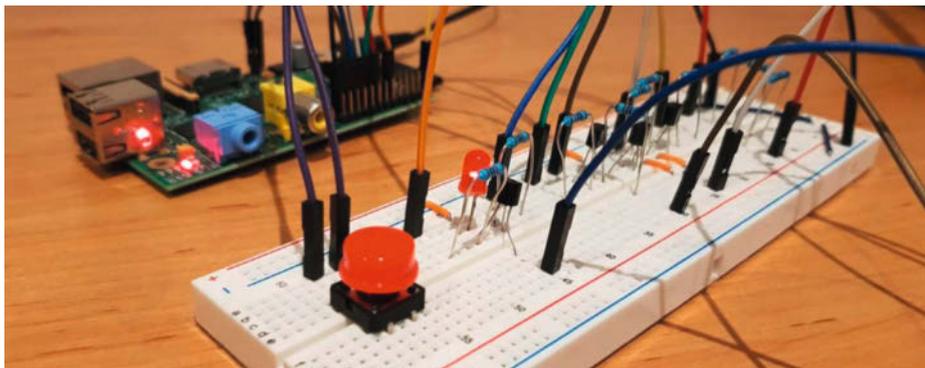


Wikipedia-User Ratopi

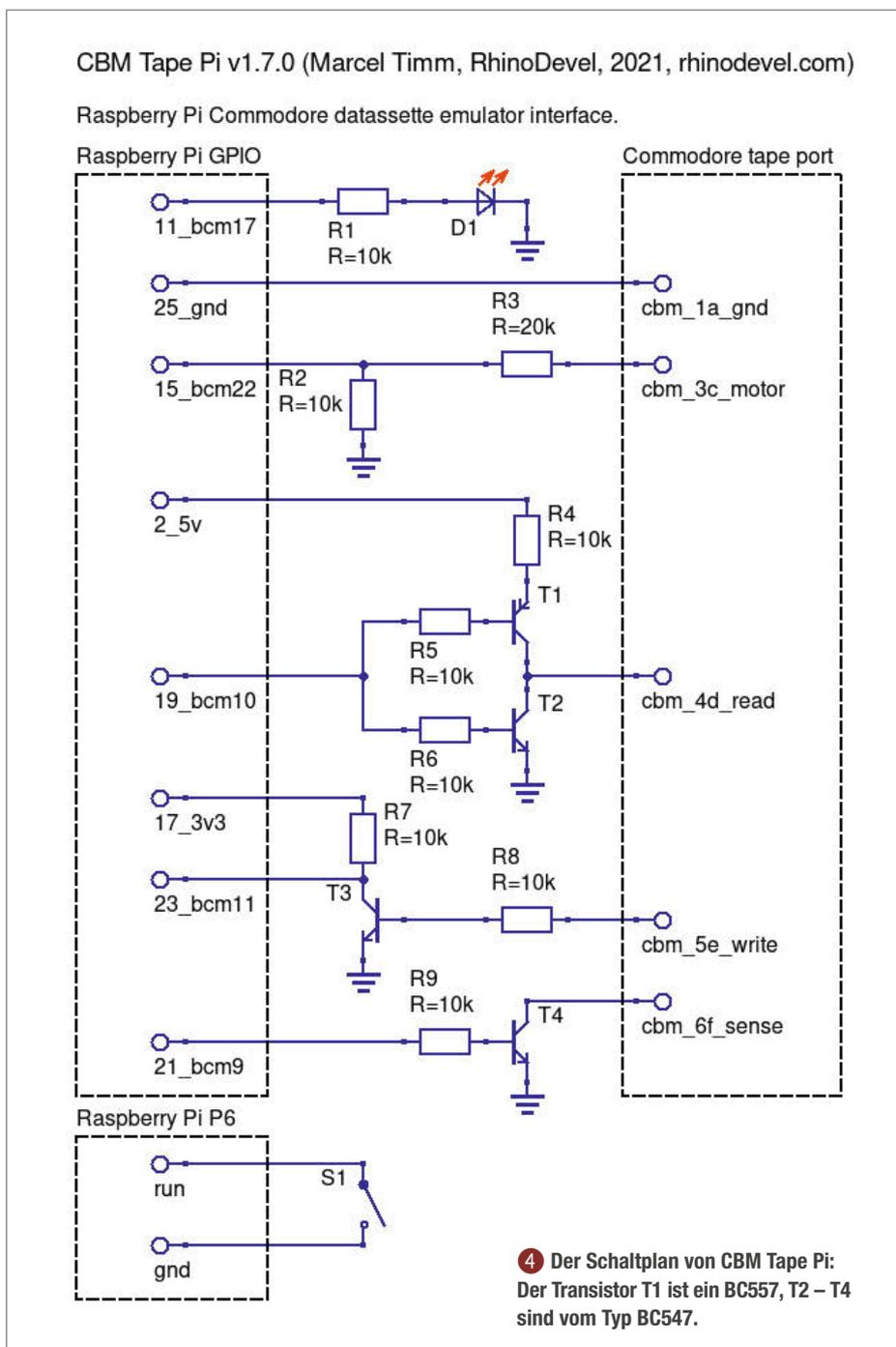
- 1** Das VC1541-Laufwerk konnte serienmäßig stolze 165 Kilobyte an Daten und Programmen auf Floppy-Disks aufnehmen und hatte die Dimensionen eines Schuhkartons der Größe 54!



Public Domain



3 Die Schaltung kann auf einem Breadboard aufgebaut werden.



www.c64-wiki.de/wiki/Kassettenport

verfügbar, aber auch beides auf der Projekt-Website.

SD-Karte vorbereiten

CBM Tape Pi ist eine sogenannte Bare-Metal-Anwendung, kommt also ohne Betriebssystem aus. Die Boot-Partition der SD-Karte muss im FAT32-Format formatiert sein. Darauf legt man die Dateien *fixup.dat*, *start.elf* und *bootcode.bin*, die man unter folgender URL herunterlädt 5.

Jetzt fehlt nur noch die CBM-Tape-Pi-Datei, zu finden in dem Archiv unter 6.

Diese ZIP-Datei beinhaltet je eine Datei für die unterschiedlichen Raspberry-Pi-Modelle, die unterstützt werden (1, 2, 3 und Zero). Bitte die zum verwendeten Pi passende Datei wählen und ebenfalls auf die SD-Karte kopieren.

Ansonsten können jetzt PRG-Dateien auf die SD-Karte kopiert werden, die später auf den Commodore geladen werden sollen.

Inbetriebnahme

Bevor Pi oder Commodore eingeschaltet werden, müssen alle Kabel verbunden sein. Auf der Commodore-Seite ist dies nur der Kassettenanschluss. Es wird empfohlen, immer zunächst den Raspberry Pi einzuschalten. Wenn die auf dem Pi verbaute ACT-LED im Sekundentakt blinkt (quasi als Puls), ist CBM Tape Pi bereit und der Commodore-Computer kann gestartet werden.

Da CBM Tape Pi verschiedene Rechner unterstützt, muss nun eingestellt werden, welcher Schnelllade-/Speicher-Modus aktiviert werden soll. Es geht auch ohne, dann aber im von der Datassette gewohnten Schnecken-tempo (67,5 Byte/s). Auf dem C64 muss beispielsweise dies eingegeben werden:

SAVE"MODE C64"

Weitere Modi können der Projekt-Website entnommen werden (für andere Rechner und zur Installation der Befehls-erweiterung in andere Speicherbereiche). Ab dem nächsten Start befindet sich der Raspberry Pi immer automatisch in dem gewählten Modus.

Falls man einmal einen anderen Computer benutzen möchten, kann man mit der entsprechenden SAVE-Zeile jederzeit den Modus wechseln. Nach einem Neustart versteht sich CBM Tape Pi dann mit dem neuen Computer.

Nun schaltet man erst den Commodore aus, und drückt, falls vorhanden, den Reset-Button der Schaltung. Ohne Reset-Knopf schaltet man den Raspberry Pi aus und wieder ein, und schaltet den Commodore dann wieder ein.

Befehlserweiterung installieren

Der Raspberry Pi sendet nun ein kleines Programm in einer Endlosschleife an den Commodore-Computer, bis der dieses Programm lädt und es dort installiert wurde (als ob im Kassettenlaufwerk ein Endlosband mit dieser PRG-Datei abgespielt wird). Am C64 lädt und installiert man diese Befehlserweiterung beispielsweise mittels der Tastenkombination *SHIFT + RUN/STOP*. Dies ist bei jedem Start nötig, da die Befehlserweiterung im RAM des C64 installiert und beim Ausschalten/Neustart gelöscht wird.

Danach ist der *Fast Mode* erreicht und PRG-Dateien können sehr schnell von der SD-Karte geladen sowie darauf gespeichert werden.

CBM Tape Pi verwenden

Die Befehle der Erweiterung beginnen immer mit einem Ausrufezeichen *!*. Das sagt dem Heimcomputer, dass es sich um einen CBM-Tape-Pi-Befehl handelt. Beispiel: Der Befehl, um eine PRG-Datei namens *MYGAME.PRG* von der SD-Karte zu laden, lautet:

5 Adresse für die Startdateien

<https://github.com/raspberrypi/firmware/tree/1.20171029/boot>

6 Adresse für die CBM-Tape-Pi-Software

https://github.com/RhinoDevel/cbmtapepi/releases/download/v1.7.0/CBM.Tape.Pi.-.v1_7_0.zip

!MYGAME.PRG

Das Speichern, beispielsweise des eigenen BASIC-Programms im C64-Speicher, erfolgt durch:

!+MYPROG

CBM Tape Pi bietet weitere Befehle, etwa durch die Verzeichnisse auf der SD-Karte wandern, den Verzechnisinhalt auflisten, Dateien löschen usw. Sie sind auf der Projekt-Website beschrieben. Hier ist noch anzumerken, dass Version 1.7.0 nur Dateinamen mit maximal zwölf Zeichen unterstützt, wobei keine Sonderzeichen sondern

nur *ASCII* beziehungsweise *PETSCII* unterstützt werden.

Ausblick

CBM Tape Pi wird aktiv weiterentwickelt und in Zukunft um neue Funktionen erweitert. Beispielsweise sind der Support für TAP-Dateien, die Nutzung von CBM Tape Pi als Linux-Anwendung direkt unter Raspbian oder der Support von weiteren Commodore Computern wie C16 mögliche nächste Verbesserungen. Ein Port zur Verwendung eines Arduino anstatt des Raspberry Pi ist in Arbeit. —hgb



Mit LEGO® programmieren lernen

Programmieren lernen wird zur spielerischen Erfahrung – mit den neuen LEGO-Robotik-Sets und mit diesem Buch gelingt das besonders gut. Autor Alexander Schulze bietet einen methodisch sinnvollen Weg, die zwei Sprachen Scratch und Python Schritt für Schritt zu erlernen.

316 Seiten · 29,90 €
ISBN 978-3-86490-856-9

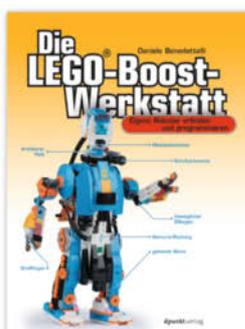
Bundle up!
Print & E-Book
www.dpunkt.de



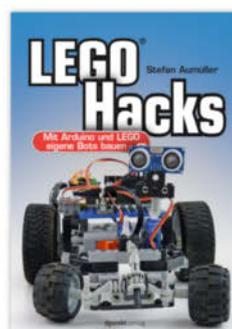
212 Seiten · 26,90 €
ISBN 978-3-86490-799-9



356 Seiten · 32,90 €
ISBN 978-3-86490-687-9



272 Seiten · 26,90 €
ISBN 978-3-86490-644-2



320 Seiten · 29,90 €
ISBN 978-3-86490-643-5

© Copyright by Maker Media GmbH.

Booster für E-Gitarren

Der Gitarrist mag Booster. Das sind Vorverstärker, mit denen er seinem Verstärker Dampf machen kann, dass die Röhren röhren! Aber ein Booster vermag auch einfach nur unterschiedliche Gitarren- oder Effektgerätepegel auszugleichen. Unser Bauvorschlag kann über ein einziges Poti als Bedienelement sowohl verstärken als auch abschwächen.

von Detlef Grell



Die meisten Gitarristen besitzen mehr als eine Gitarre und oft auch mehr als einen Verstärker. Zwischen Amp und Gitarre sind meistens noch eine Handvoll „Tretminen“ geschaltet, also per Fußschalter zu aktivierende Kistchen mit Funktionen wie Hall, Chorus, Verzerrer, Kompressoren und etliches mehr. Unser Booster gleicht nun nicht nur den Pegel unterschiedlicher Gitarren aus, er erspart beim Umstöpseln auch einen „Neuabgleich“ der Effektgeräte-Kette und sorgt mit seiner Verstärkung auch für genügend Vorstufen-Crunch des Gitarren-Amps, gerade bei Instrumenten mit kleinerem Ausgangspegel (oft bei *Single-Coil*-Pickups).

Job-Beschreibung

Ein solches *Rig* ist pegelmäßig oft sehr fein austariert. Was mit der Stratocaster gerade noch *clean* (unverzerrt) oder *crunchy* (leicht angezerrt) klingt, klingt dann mit der Gibson Les Paul nur noch verzerrt. Oder es wird richtig eklig, weil eine der Stompboxes übersteuert wird. Speziell die heute üblichen Digital-Kistchen können beim Übersteuern gruselige Sounds erzeugen – ganz anders als die geile Zerre einer übersteuerten Verstärkervorstufe.

Zur Pegelanpassung mit den Volume-Knöpfen an der Gitarre selbst mögen nur wenige Gitarristen (ich eingeschlossen) hantieren, denn der jeweilige Widerstandsanteil des Volume-Potis, der dann in Serie zum Gitarrenkabel liegt, erzeugt mit der Kapazität des geschirmten Kabels und dem kapazitiven Teil der Eingangsimpedanz des Verstärkers einen Tiefpass. Sprich: Es gehen Höhen verloren.

Dem kann man zwar mit einem 1nF-Kondensator parallel zum Poti-Widerstand abhelfen, aber das stößt auf Probleme: An wertvollen Gitarren, womöglich Originalen aus den 50ern, wird man nicht herumfummeln. Hier herrscht das Gesetz des „Du sollst Vintage-Gitarren nicht verbasteln, falls du sie jemals zu verkaufen gedenkst.“

Und natürlich ist das für die Fraktion der Goldhörchen, die es nicht nur in der HiFi-Welt gibt, ebenso ein nicht verhandelbares No-go: Selbstverständlich hört so jemand den Unterschied von Gitarren mit und ohne einen solchen Treble-Bleed-Kondensator. Wohlgermerkt auch dann, wenn das Poti voll aufgedreht ist und den Kondensator vollkommen kurzschließt – das hat dann aber wohl quantenmechanische Ursachen im Orgonfeld der Lötstellen. Unser Booster löst das Problem elegant: Solange Sie die Gitarre über dasselbe Kabel anschließen, erhalten Sie stets dieselbe Eingangsimpedanz und damit dasselbe Frequenzverhalten.

Einknopf-Lösung

Selbst einige der besseren Booster kosten gerade mal 50 Euro. Aber nur bei ganz wenigen lässt sich mit nur einem Drehknopf der

Kurzinfo

- » Wozu sind Booster gut?
- » Einknopf-Lösung mit Operationsverstärker-IC
- » Stromversorgung: Netzteil oder Batterie?

Checkliste



Zeitaufwand:
3 Stunden



Kosten:
25 Euro



Materialbearbeitung:
Lötstation, Bohrmaschine oder Akkuschrauber

Alles zum Artikel im Web unter make-magazin.de/x3br

Material

- » Platine nach Gerber-Daten (siehe Link)
- » Bauteile nach Stückliste/Schaltplan
- » Thomann-Fußschalter FS-2 als Gehäuse

Mehr zum Thema

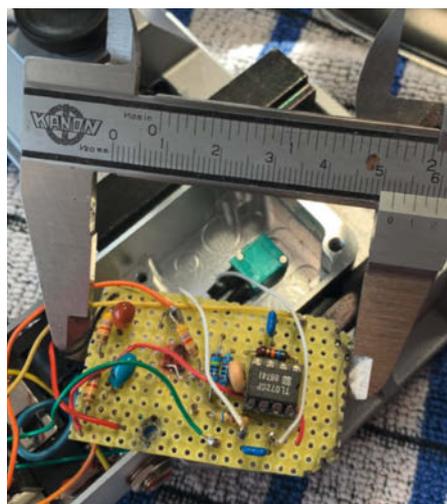
- » Volkmar Spinnler und Tino Werner, Bot-Kontrolle mit Operationsverstärker, Make 4/16, S. 60
- » Hans-Joachim Heckert, Gitarrenverstärker für unterwegs, c't Hacks 2/12, S. 118

Pegel sowohl anheben als auch abschwächen. Die meisten haben eine Zwei-Knopf-Bedienung über *Gain* und *Volume/Level*. Die muss man immer erst nach Gehör einpegeln, wenn es ganz clean und möglichst rauschfrei zugehen soll.

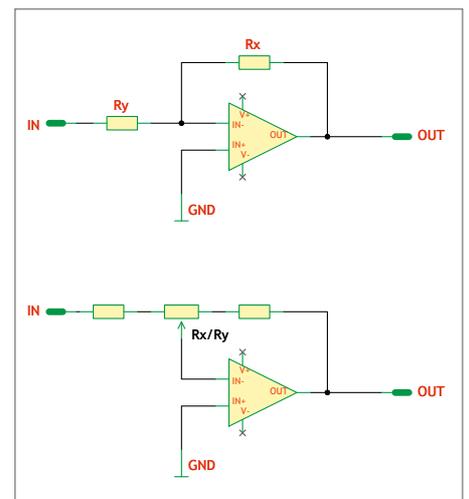
Kernstück des Boosters ist das IC TL072, das zwei separate, rauscharme Operationsverstärker (im Folgenden kurz „OpAmp“) in einem 8-poligen DIP-Gehäuse enthält. Der in JFET-Technik ausgeführte TL072 ist für ein paar Cent zu haben. Neben seinen günstigen Rauschwerten ist er auch intern

gut kompensiert (keine Schwingneigung) – ein wichtiger Punkt, denn längst nicht alle OpAmps arbeiten bei kleinen (!) Verstärkungen stabil.

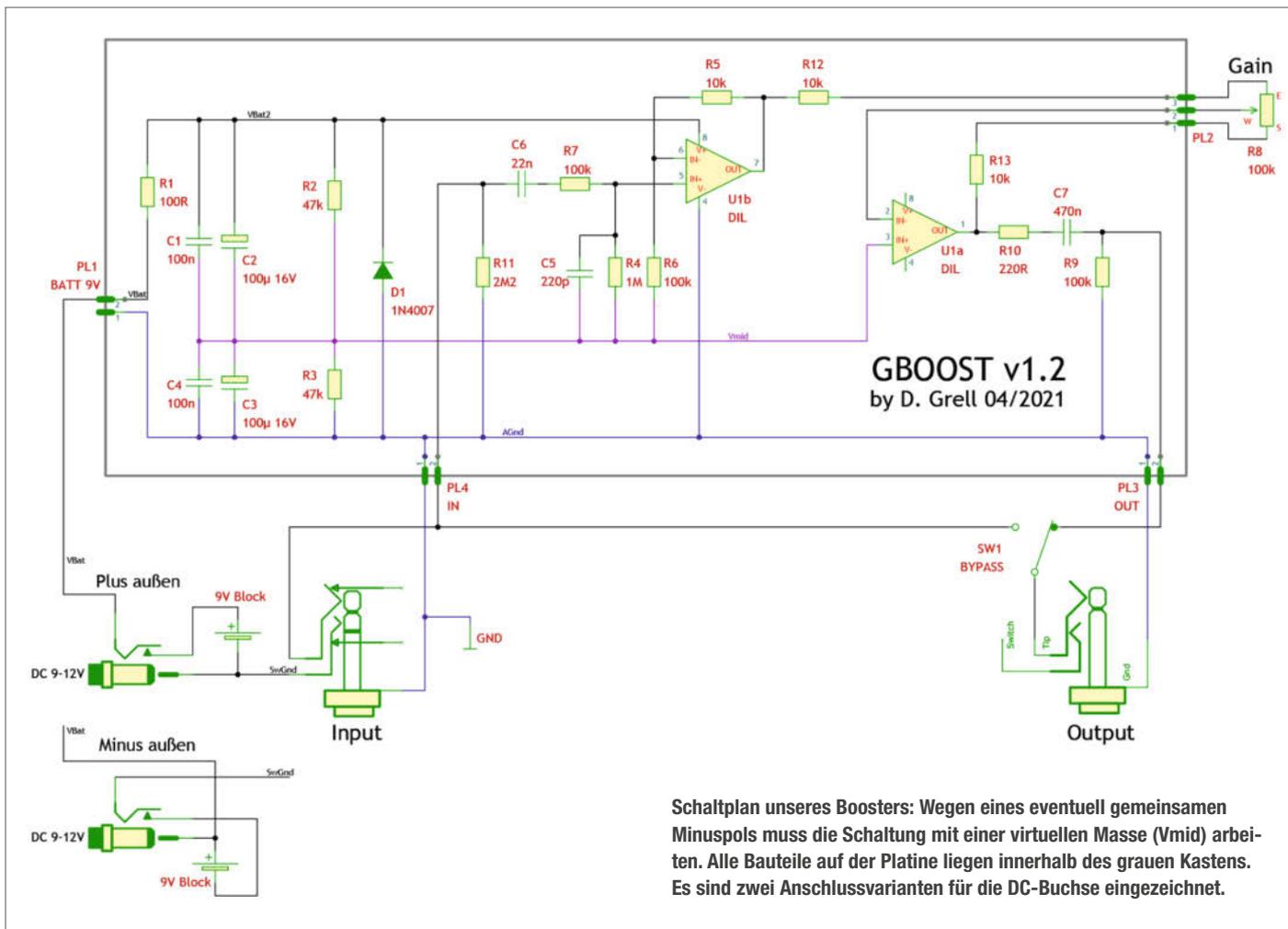
20dB Spannungsverstärkung respektive -abschwächung entsprechen jeweils Faktor 10. Will man das mit nur einem Potentiometer variieren, lässt sich das am einfachsten mit einem OpAmp in invertierender Schaltung realisieren. Dessen Verstärkung wird durch das Verhältnis zweier Widerstände R_x/R_y im Gegenkopplungszweig definiert. Ist das Verhältnis glatt 1, ist auch die Verstärkung 1.



Erster Lochraster-Prototyp, hier noch in etwas anderer Beschaltung. Unsere aktuelle Platine ist mit 48mm x 30mm sogar noch kleiner geworden und passt problemlos in das abgebildete kleine Gehäuse.



Die Verstärkung des OpAmps im oberen Bild entspricht genau dem Verhältnis R_x/R_y . Mit einem Potentiometer (unteres Bild) wird das Verhältnis von R_x zu R_y einstellbar. In der Mittelstellung ergibt sich ein Verstärkungsfaktor von genau 1.



Ist R_x größer als R_y , gibt das Verhältnis die Verstärkung an, ist R_x kleiner, findet eine Abschwächung statt. In unserem Design ist das Verhältnis von R_x/R_y über das Poti R8 stufenlos von 10:1 bis 1:10 einstellbar.

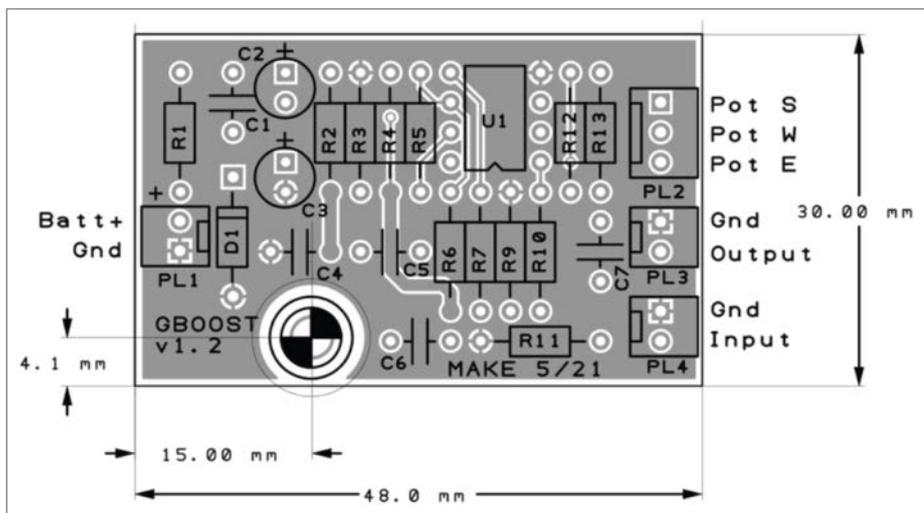
Da bei der invertierenden Schaltung Änderungen der Verstärkung durch R8 auch den Eingangswiderstand beeinflussen, setzt man üblicherweise einen Pufferverstärker davor. Die Schaltung ist auf etwa 700 kOhm Eingangs-

widerstand ausgelegt. Der ergibt sich aus der Parallelschaltung von R11 und der Summe von R7 plus R4. R7 bildet mit C5 zusammen einen Tiefpass gegen Störeinstrahlungen. Mit 220pF für C6 liegt die Grenzfrequenz bei etwa 7kHz. Wer meint, seine Gitarre liefere höhere Frequenzen, kann die Grenze mit 150pF bei 10kHz ziehen.

Die etwa 10 Prozent Signalverlust am Tiefpassfilter aus C5, R7 und R4 kompensiert der nichtinvertierende erste Teil des TL072. R11 sorgt als Potenzialausgleich dafür, dass beim Umschalten zwischen Booster und Bypass-Betrieb keine Knackgeräusche auftreten, ebenso wie R9 am Ausgang. R4 stellt den Arbeitspunkt der Schaltung ein.

Über das 100kOhm-Poti wählt man die gewünschte Verstärkung oder Abschwächung. Ist das Poti R8 am Anschlag S, gilt volle Abschwächung, steht es auf E, gilt volle Verstärkung. Setzt man ein lineares Poti ein, ergibt sich in Mittelstellung eine Verstärkung von 1. Die Widerstände R12 und R13 begrenzen die Abschwächung bzw. Verstärkung in den Poti-Endstellungen auf den Faktor 10.

Das Gebinde aus R1 bis R3 und C1 bis C4 glättet die Betriebsspannung und stellt für den OpAmp eine virtuelle Masse auf der halben



Bei der Bestückung kommen nur lötfreundlich bedrahtete Bauteile zum Einsatz. An die Anschlüsse „Pot“ wird das Gain-Poti angeschlossen: Schleifer an Pot W, Anfang an Pot S und Ende an Pot E. Das Layout finden Sie unter dem Link im Info-Kasten.

Betriebsspannung bereit. Oder anders formuliert: OpAmps werden üblicherweise symmetrisch versorgt, was hier mittels der zwei Hälften der Batteriespannung simuliert wird.

Max Headroom

Wenn man eine E-Gitarre ordentlich „schrubbt“, liefert sie deutlich höhere Ausgangsspannung als beim zärtlichen Zupfen. Als passabler Richtwert hat sich ein Wert von 200mV effektiv (280mV Spitzenwert einer Halbwelle) erwiesen, wenn es um die Dimensionierung im Hinblick auf Aussteuerungsgrenzen geht. Wie viele Standard-OpAmps lässt sich der TL072 zuverlässig bis 1,5V unterhalb der Versorgungsspannungen aussteuern: bei der üblichen Stompbox-Betriebsspannung von 9V ($\pm 4,5V$) also bis $\pm 3V$. Wenn man den Spitzenwert 280mV mit 10 multipliziert, liegt die Aussteuerung bei 2,8V. Das kann knapp werden, aber wenn's zerrt, dreht man halt die Verstärkung etwas herunter.

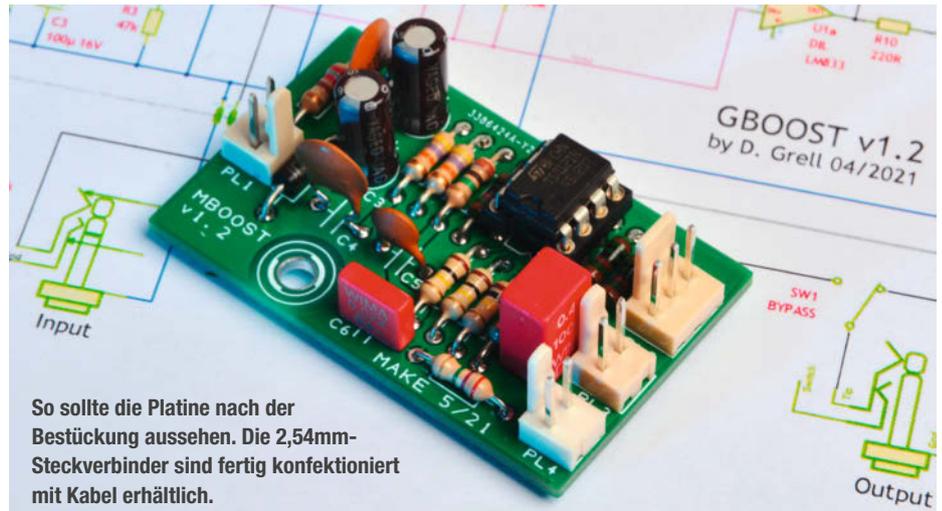
Für meine Gitarren mit moderater Ausgangsspannung passt das prima. Es hindert Sie aber auch niemand daran, die Betriebsspannung auf 12 oder 18V zu erhöhen. Bei 12V erhöht sich der Headroom bereits auf $\pm 4,5V$, bei 18V auf $\pm 7,5V$. Mehr Headroom kann man auch für eine höhere Verstärkung nutzen, etwa indem man R12 verkleinert. Bedenken Sie aber, dass das mitverstärkte Rauschen irgendwann stört. 20dB überschreiten daher auch die meisten kommerziellen Booster nicht.

Der Hohlstecker für den Netzteilanschluss des Boosters führt, wie in der Stompbox-Welt üblich, Plus am äußeren Kontakt, es gibt allerdings auch Ausnahmen (*Eventide* etwa). Diode D1 schließt allerdings die Betriebsspannung bei falscher Polung zum Schutz der Schaltung einfach kurz. Schonender für Netzteil und Diode wäre, wenn man sie in Serie zur Booster-Schaltung legt, aber dann verliert man knapp 1 Volt kostbaren Headroom.

Obwohl sich die Schaltung problemlos auf einer Lochraster-Platine aufbauen lässt, haben wir eine kleine Platine entworfen, deren Layout Sie online unter dem Link im Info-Kasten finden. Sie ermöglicht trotz ausschließlicher Verwendung bedrahteter Bauteile einen besonders kompakten und noch dazu ästhetischen Aufbau. Für eine besonders universelle Verwendbarkeit – ohne an ein bestimmtes Gehäuse gebunden zu sein – haben wir Klinkenbuchsen und Poti nicht darauf untergebracht, diese müssen anhand des Schaltbilds frei verdrahtet werden.

Günstige Behausung

Das für den Booster gewählte Gehäuse *Lead Foot FS-2* von Thomann ist eigentlich ein Doppel-Fußschalter. Für 12,90 Euro bekommt man hier Gehäuse und Tretschalter im Paket, in das



So sollte die Platine nach der Bestückung aussehen. Die 2,54mm-Steckverbinder sind fertig konfektioniert mit Kabel erhältlich.

neben der Platine auch die 9V-Batterie passt. Vielleicht liegt bei Ihnen aber auch ein „schlachtrettes“ (defektes, rauschiges) Kistchen rum, das auf seine Chance als Organspender (Gehäuse, Tret-Schalter, Buchsen) wartet. Die Booster-Platine dürfte sogar in viele Gehäuse von Mini-Stompboxen passen.

Beim FS-2 sind zwei Tretschalter eingebaut, einen davon nimmt man raus und setzt dort das Poti R8 ein. Schon mal ein Bohrloch im Aluminium gespart! Das geschirmte Stereoklinkenkabel an der Rückseite brauchen Sie nicht. Die Seitenwände sind aus Plastik und damit besonders leicht zu bohren – hier sollten die 6,3mm-Klinkenbuchsen für Ein- und Ausgang eingesetzt werden. Wenn Sie eine 9V-Batterie einbauen wollen, dann benutzen Sie eine Stereobuchse am Eingang. An den Kontakt für den Ring legen Sie dann die Minus-Versor-

gung. Der Monostecker des Gitarrenkabels schließt dann den Kontakt zum Gehäuse. Zieht man den Stecker ab, fließt kein Strom und die Batterie wird geschont.

Hohlsteckerbuchsen mit Zentralverschraubung für die Stromversorgung können unter Umständen den Außenkontakt zwangsweise auf Masse legen. Da möglicherweise außen an der Buchse aber der Pluspol liegt, am Gehäuse-Äußeren (bzw. an den Klinkenbuchsen) jedoch Minus, lässt sich die Buchse am einfachsten in den Plastikseitenwänden isoliert einbauen, falls nötig. Vergessen Sie nicht, das Gehäusemetall mit Masse zu verbinden. Sollten die unabgeschirmten Seitenwände zu viele Störungen durchlassen, dann können Sie die mit Alu- oder Kupferfolie auskleiden.

Aus dem Plus-Minus-Dilemma (siehe Kasten *Netzteil-Frage*) erklärt sich auch ein eher

Netzteil-Frage

Im Musikalienhandel finden Sie im Bereich um 10 Euro 9V-Schaltnetzeile und ebenso welche, die auch auf 12V umschaltbar sind. Schaltnetzeile sind leicht und preiswert, viele arbeiten allerdings mit Schaltfrequenzen im hörbaren Bereich. Mit etwas Pech kann man Einstreuungen davon im Nutzsignal hören. In meiner Praxis hat es sich bewährt, in solchen Fällen Ferrit-Schalenkerne auf das Kabel der 9V-Zuleitung zu klipsen. Man erzeugt so ein Tiefpassfilter für die Schaltfrequenzen. Die Ferrit-Ringe gibt es im Elektronikfachhandel etwa bei Reichelt oder Conrad für verschiedene Kabeldurchmesser.

Achten Sie auf jeden Fall auf die Polung des Steckers: Bei Netzteilen für Effektgeräte ist aus historischen Gründen oft

der Pluspol an der Stecker-Hülse, während bei Consumer-Elektronik überwiegend „Minus außen“ gilt. Geradezu verwirrend ist zudem die Auswahl an Innen- und Außendurchmessern der koaxialen Stecker, auch diese müssen natürlich zu Ihrer Buchse passen.

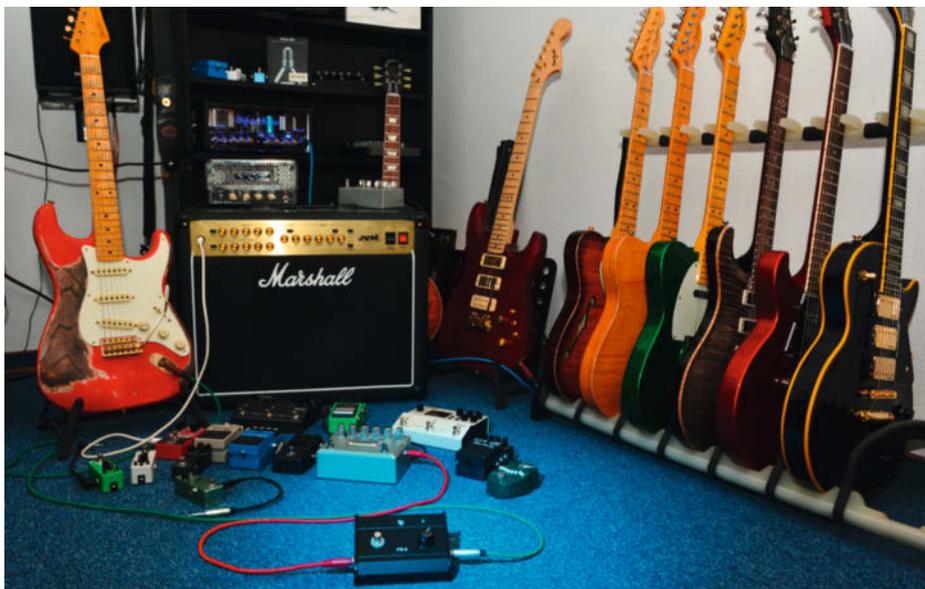
Die meisten Hohlstecker-Buchsen haben drei Anschlüsse: Einen für den Mittelpin, einen für die Außenhülse und einen Schaltkontakt, der nur bei gezogenem Stecker mit dem Anschluss für die Außenhülse verbunden ist. Der dient zum Abschalten der Batterie (siehe Schaltplan): Ohne Stecker kommt die Batterie zum Zuge, bei gestecktem Netzteil wird sie abgeklemmt und nicht versehentlich geladen – was ihr nicht bekommen würde.



Fertiger Aufbau des Boosters: Sie sollten jetzt vom FS-2 ein zweipoliges Klinkenkabel und einen Tretschalter übrig haben. Die Stromversorgung wird durch Einstecken des Gitarrenkabels eingeschaltet, einen separaten Ein-Ausschalter gibt es nicht.



Beispiel-Aufbau mit unserer Platine im Gehäuse des FS-2. Die abnehmbaren Plastik-Seitenteile sind besonders einfach zu bearbeiten, deshalb haben wir Stromversorgungs- und Klinkenbuchsen hier montiert. Durch die kurzen Leitungswege sind abgeschirmte Kabel hier überflüssig.



Viele Gitarren, viele Effekte, viele Pegel: Unser Booster gleicht Lautstärkeunterschiede verlustfrei aus.

ungewöhnliches Schaltungsdetail: Der „natürliche“ Massepunkt wäre die virtuelle Masse am Mittelpunkt des Spannungsteilers, die Ein-/Ausgangsbuchsen haben wir jedoch auf den Minus-Teil der Spannungsversorgung gelegt. Das geht, weil schaltungstechnisch die Netzteilkondensatoren C1 bis C4 für die Signalspannungen einen Kurzschluss darstellen.

Zum Schluss noch ein kleiner Hinweis: Üblicherweise haben Stompboxes ihren Eingang rechts und den Ausgang auf der linken Seite. Das hatten wir bei unserem ersten Prototypen (Detailbilder) nicht bedacht, die Verdrahtung ist also „spiegelverkehrt“. Das endgültige Gerät (siehe Bild auf der ersten Seite des Artikels) erhielt dann eine Stompbox-konforme Buchsenanordnung. In den Plänen zum Projekt (siehe Link im Info-Kasten) haben wir noch weitere Informationen zusammengetragen, etwa zum Einsatz anderer Operationsverstärker und die dazugehörigen Messergebnisse. —cm

True Bypass

True Bypass bedeutet, dass beim Abschalten der Stompbox-Funktion eine vollkommen vom Innenleben getrennte Verbindung zwischen Ein- und Ausgangsbuchse geschaltet wird; dafür wird ein zweipoliger Umschalter benötigt (eigentlich sind sogar drei Schaltkontakte sinnvoll, um im Bypass-Betrieb auch eine Active-LED abzuschalten).

Uralte Geräte wie *MXR Phase 90* oder *Distortion+* hatten kein True Bypass und verunstalteten das Signal beim Ab-

schalten gnadenlos, weil die Eingangsstufe das Gitarrensignal weiterhin kapazitiv und mit ihrem recht niedrigen Eingangswiderstand belasteten.

Ist der Eingangswiderstand jedoch so hoch wie in unserer Booster-Schaltung, reicht es völlig, nur den Ausgang mit einem einpoligen Umschalter wie im FS-2 mitgeliefert umzuschalten. Die Eingangsbeschaltung des Boosters kann das Signal nicht wirklich verfälschen. Eine solche Umschaltung ist im Schaltbild eingezeichnet.

Wenn man Wert auf eine LED als Aktivitätsanzeige legt, sollte man dafür einen eigenen Schalter-Pfad reservieren. Für die konsequente Lösung True Bypass plus Betriebs-LED benötigen Sie dann einen (Tret-)Schalter mit 3-fach-Umschalt-Funktion. Den finden Sie am einfachsten und wohl auch am günstigsten in einer Billig-Stompbox aus China. Wenn diese als True Bypass deklariert mit Status-LED kommt, dann erhält man ein Paket aus 3-fach-Umschalter, Gehäuse und Buchsen schon um die 20 Euro.

Ein Breadboard ist für Sie kein Schneidebrett?

Dann sind Sie bereit für den Einstieg in die Elektronik!

Exklusiv
im heise shop!



Inklusive
Experimentier-
set

NEU

Make Elektronik Special

Make Elektronik Special bietet einen einfachen und praxisorientierten Einstieg in Transistorschaltungen, die Maker in eigenen Projekten einsetzen können. Das mitgelieferte Experimentierset inkl. Breadboard, Kabeln und 45 Elektronikbauteilen enthält alles, um die gezeigten Schaltungen sofort nachbauen und testen zu können. Dabei machen die speziellen Adapterplatinen den Aufbau zum Kinderspiel. Jede Menge Projekte zum Nachbauen sind ebenfalls dabei. Also – sofort loslegen!

shop.heise.de/make-elektronik21

Heft + Experimentierset für nur
44,95 €

 heise Shop

shop.heise.de/make-elektronik21

Generell portofreie Lieferung für Heise Medien- oder Maker Media Zeitschriften-Abonnenten oder ab einem Einkaufswert von 20 €. Nur solange der Vorrat reicht. Preisänderungen vorbehalten.

© Copyright by Maker Media GmbH

KRITIS Powerbox

Die Unwetterkatastrophe in NRW und Rheinland-Pfalz hat gezeigt, dass die Befürchtungen vieler Wissenschaftler schnell Realität werden können: Innerhalb von nur wenigen Stunden waren ganze Landstriche von der Außenwelt abgeschnitten. Die Powerbox stellt Strom für die wichtigsten Verbraucher wie Kommunikation und medizinische Geräte bereit.

von Daniel Jedecke



Wir als Gesellschaft sind gewohnt, dass alles funktioniert. Diese Tatsache ist als Verletzlichkeitsparadoxon bekannt und gilt gerade für kritische Infrastrukturen als Hauptargument, um diese zu schützen. Es gibt mehrere Gesetze, um die Sicherheit und Verfügbarkeit dieser Infrastrukturen zu gewährleisten. Naturkatastrophen können jedoch unberechenbar und verheerend sein.

Es gibt im Katastrophenfall vor Ort oft keinen Strom und daher z. B. kein sauberes Wasser und keine Kommunikation. In dem Community-Projekt der AG KRITIS soll daher gezeigt werden, wie man sich auf solche Situationen vorbereiten und mit vorhandenen Mitteln vor Ort eine kleine Ersatzversorgung aufbauen kann. Je nach Bedarf gibt es sehr viele am Markt verfügbare Optionen. So kann eine USB-Powerbank für die Kommunikation reichen oder eine große unterbrechungsfreie Stromversorgung (USV) im Keller helfen. Diese kostet aber oft über 1.000 Euro. Es entstand die Idee einer mobilen Notstromversorgung auf Photovoltaik-Basis mit Batteriepuffer. Das System besitzt einen Batteriespeicher, unterstützt das Laden von USB-Geräten und hat ein 230V-Wechselstrom-System für Dinge des alltäglichen Bedarfs (Flaschenwärmer, Licht, etc.) sowie einen 12V Ausgang für Ladegeräte. Viele Materialien fanden sich auch in den Krisengebieten und hätten kurzfristig umgebaut werden können (Kabel, Photovoltaik-Panels, Batterien). Kosten sollte das System maximal 300 Euro.

Es wurden normale KFZ-Batterien verwendet, da diese überall erhältlich sind und im *worst case* aus einem defektem Fahrzeug vor Ort ausgebaut werden können. Diese KFZ-Batterien sind nicht für den Dauereinsatz geeignet (Zyklusfestigkeit), aber im günstigen Notsystem nutzbar. Um mobil zu bleiben, sollte nur ein kleines Photovoltaik-Panel genutzt werden, um die Batterie unterwegs wieder aufzuladen.

Da man bei Bleibatterien nie die volle Leistung entnehmen sollte (maximal 50 %), können mit der ausgewählten Batterie 27Ah Energie gespeichert werden, womit der 300W-Wechselrichter für eine Stunde bei maximaler Last laufen kann. Je nach Budget und Größe der Box können natürlich größere Batterien benutzt werden. Mit Batterie-Schnellklemmen kann die Batterie einfach gegen eine andere Batterie getauscht werden. Es wurde darauf geachtet, dass mittels MC4-Adapter auch andere PV-Panels (wie von einem Dach) genutzt werden können. Hierbei ist darauf zu achten, dass die PV-Wechselrichter die Leistung der Module unterstützen.

Unter dem folgenden Link finden Sie eine detaillierte Bauanleitung, Erfahrungsberichte und Teilelisten.

—caw

► <https://heise.de/s/wm2j>



Das Herzstück, der Solar-Wechselrichter. Dieser übernimmt das Laden der Batterie und bietet direkt für zwei USB-Geräte Spannung.



Die Box beim Aufwärmen von Babybrei. Hierbei ist der Kocher am 230V-Netz angeschlossen



Die fertige Box beim Nachladen an der Sonne

Cyborg Control

Mit einem ESP-Mikrocontroller kommt altes Elektronikspielzeug wieder auf den Stand der Technik – und lässt sich prima in interaktive Projekte integrieren.

von Helga Hansen



Bilder: Niklas Roy & Kati Hyypää (CC BY 3.0)

Was passiert, wenn die Pandemie auf die Singularität und erwachte Roboter trifft? Im Schaufenster ihrer Berliner Werkstatt hat sich das Bastlerduo Kati Hyypä und Niklas Roy dieser Frage angenommen. Im Mittelpunkt ihres interaktiven Projekts „Cyborg Control“ steht ein gehackter Spielzeugroboter, den man im Vorbeigehen steuern kann. Über mehrere Wochen wuchs außerdem seine (oder ihre?) Welt um immer neue „Gebäude“ an.

Bei dem Roboter handelt es sich um einen *Robosapien* ihrer Nachbarin Heike, den die beiden im Keller fanden und umbauen durften. Der schwarz-weiße Gefährte ist ein bewegliches Spielzeug, das der NASA-Robotikspezialist Mark Tilden bereits 2004 entwickelte und das einfach zu modifizieren ist. Es wird mit einer Infrarotfernbedienung gesteuert, die das Mainboard im Rücken anspricht. Die Infrarotsignale müssen aber nicht von der Fernbedienung kommen, sondern können von einem Mikrocontroller simuliert werden.

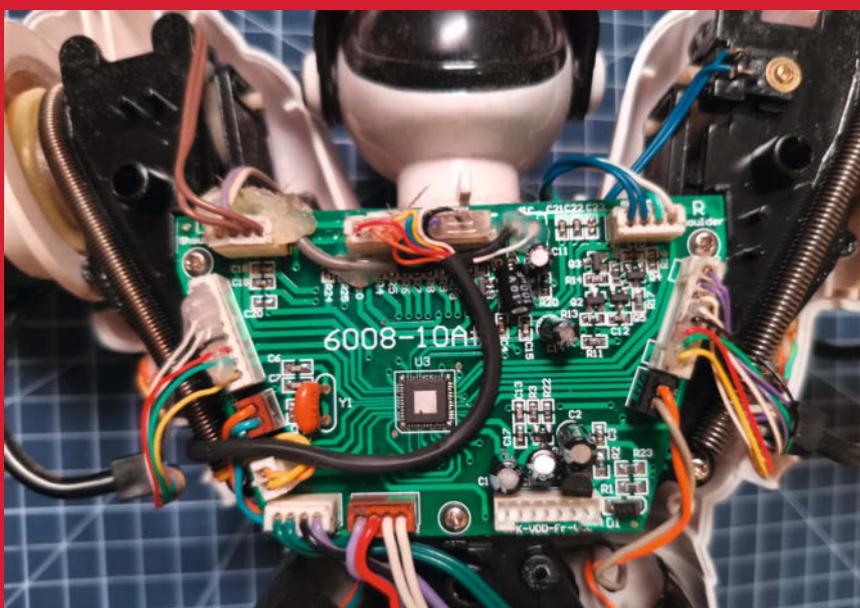
Dafür baute das Berliner Duo den WLAN-fähigen Mikrocontroller NodeMCU V1.0 ein, der wie der Robosapien bei 3,3 Volt betrieben wird und daher direkt angeschlossen werden kann. Neben der Ansteuerung des Mainboards muss der NodeMCU noch weitere Aufgaben bewältigen: Darauf laufen der Webserver und darauf wiederum die Webseite zum Kontrollieren der Roboterbewegungen. Aus Sicherheitsgründen ist das WLAN nur lokal – direkt vor dem Schaufenster – erreichbar. Ein Tipp der Beiden ist dabei die Arduino SPIFFS bzw. LittleFS-Bibliothek. Damit können alle Dateien für das Webinterface getrennt im Flashspeicher abgelegt werden, statt sie in ein langes und unübersichtliches Arduino-Programm zu quetschen. Auch äußerlich wurde der Robosapien überarbeitet. Mit einer roten LED im Auge eines Puppenkopfes erinnert er nun an einen Androiden, der eng mit einem Maschinenaufstand verbunden ist: den Terminator.

Um den Roboter und sein Publikum stetig zu unterhalten, erweiterten Hyypä und Roy schließlich Schritt für Schritt die Kulisse im Schaufenster: Nach einem Besuch im Testcenter steht dort inzwischen der Besuch im Club *Borghain* offen, bevor es auf dem Nachhauseweg noch schnell in den Alien-geführten Späti geht. Die animierten und beleuchteten Paneele reagieren dabei auf Knopfdruck des Roboters oder erkennen seine Präsenz durch Sensoren. Dafür steckt jeweils ein Arduino in den Modulen. Welche Tricks er nutzt, um interaktive Installationen besonders einladend zu gestalten, hat uns Niklas Roy übrigens in der Make 1/21 verraten. —hch

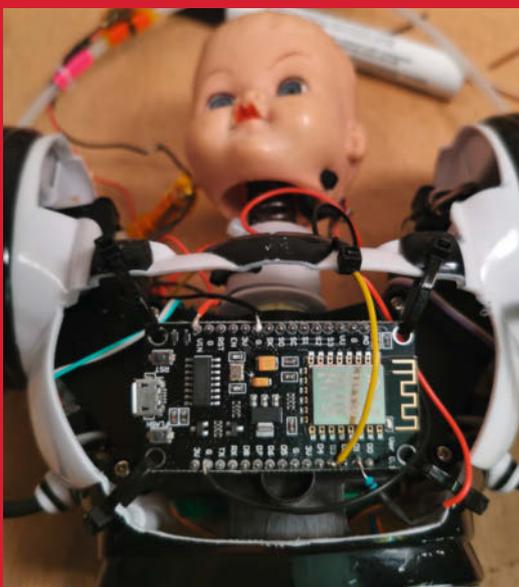
► niklasroy.com/cyborg_control



Der Robosapien im Originalzustand



Die Steuerung des Mainboards erfolgt über Infrarotsignale, ...



... statt einer Fernbedienung schickt die jetzt aber ein NodeMCU direkt auf den Eingangspin.

Ein Tablet für's Auto

Wir hatten schon eins, zwar zu klein, zu dunkel, zu lahm, keine Updates, aber im Wohnmobil angenehm. Zum Schluss war der Touchscreen defekt. Also musste ein neues Gerät her, mit Android und Touchscreen, aber mit einem Raspberry Pi als Herz und ansonsten auch alles viel besser.

von Hartmut Krummrei



Am Anfang stand die Internetrecherche. Auf der Suche nach passender Software stieß ich auf Foreneinträge eines finnischen Entwicklers, der sich *Konsta* nennt und der *LineageOS* für den Raspberry Pi 4 angepasst hat. LineageOS betreibt schon unsere diversen Smartphones und Tablets, also war klar: damit werde ich das realisieren.

Die Suche nach einem passenden Monitor endete bei einem 9-Zoll-Bildschirm, mit einer Auflösung von 1280 × 720 Pixeln. Wichtig an diesem Gerät: Die Anschlüsse befinden sich alle auf der Rückseite und nicht an den Seiten des Bildschirms, er ist für den Einbau in einem Rahmen vorgesehen.

Ich habe dann auf einer Grundplatte eine Form gebaut, auf der ich mithilfe von Hitze einen Rahmen aus Kunststoffprofil biegen konnte. Wie bei allen Selbstbauprojekten konnte ich dabei viel lernen: Die Holzform sollte ebenfalls angewärmt werden und das fertige Teil muss Zeit zum Auskühlen haben.

Aus Kunststoff-Quadratprofilen entstand dann das Einbaugerüst, von der Größe her genau abgemessen für die Öffnung im Armaturenbrett. Den Bildschirmrahmen konnte ich dann an das Gerüst anpassen und befestigen. Das Innenleben ist auf Flachprofilen befestigt, dies ist einfach zu montieren und gleichzeitig stabil. Das Ganze wird mit einem KFZ-DIN-Stecker für Autoradios komplettiert und man kann es genau wie jene einschieben, verbinden und ... läuft.

Ein Problem ist das HDMI-Kabel vom Rechner zum Monitor: Das Kabel ist steif, die Stecker brauchen viel Platz. Eine Lösung für dieses Problem fand ich bei einer Firma für Drohnenzubehör: Hier gibt es Stecker in jeder Größe und für jeden Winkel einzeln zu kaufen. Verbunden werden die Stecker mit kurzen, sehr flexiblen Flachbandkabeln.

Außer dem Raspi und Monitor fanden noch etliche weitere Komponenten den Weg in das Radio: Zwei Endverstärker, ein Spannungswandler von 12V auf 5V, ein GPS-Empfänger, eine Webcam, ein Beschleunigungssensor und ein USB-Hub. Eigentlich sollte am Monitor das per HDMI gelieferte Audiosignal abgegriffen werden, aber da liegt zu viel Störsignal drauf, deshalb wurde noch eine USB-Soundkarte verwendet.

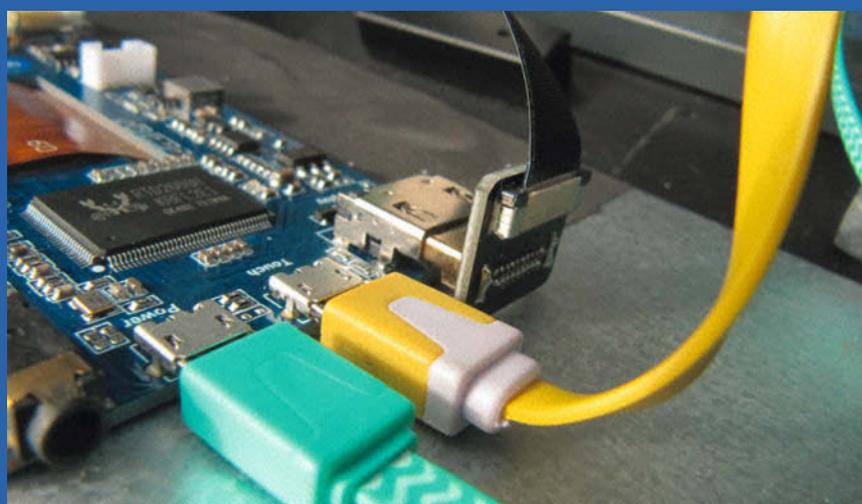
Dem ganzen fehlen die praktischen Hardwaretasten von Tablet und Co, daher wurden Tasten oben am Rahmen eingebaut, sie werden per GPIO-Pins abgefragt. Die Tasten links steuern den Monitor.

Oft nennt man diese Geräte *Moniceiver* (aus *Monitor und Receiver*), *Naviceiver* oder *Infotainmentsystem*. Für mich ist es schlicht ein Tablet für's Auto und das sorgt für Navigation (*Osmand+*), Entertainment in Bild (*Youtube, etc.*) und Ton (*Foldplay und Transistor*) und die Dashcam (*Daily Roads Voyager*).
—caw

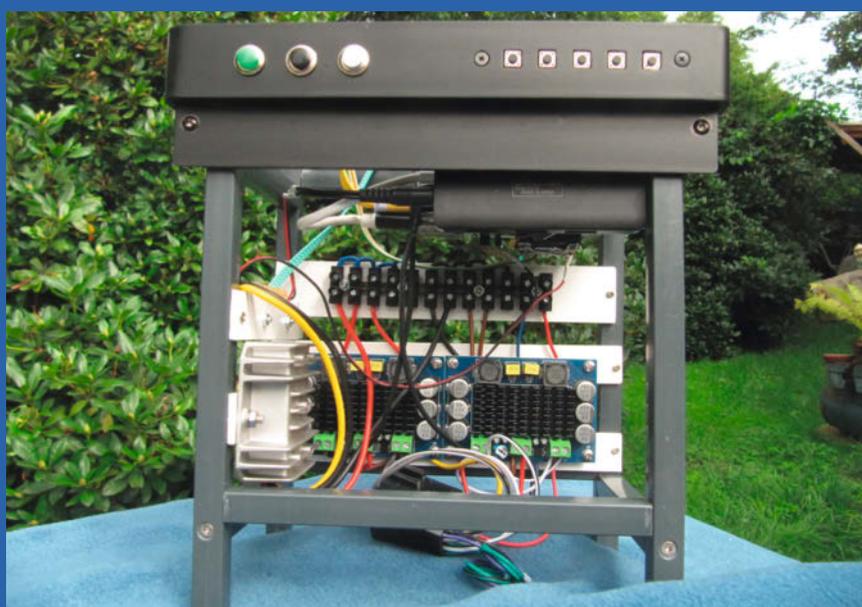
► <https://heise.de/s/2gvZ>



Monitorrahmen und Gerüst bei der Anprobe



Spezial-HDMI-Stecker und Kabel für beengte Platzverhältnisse

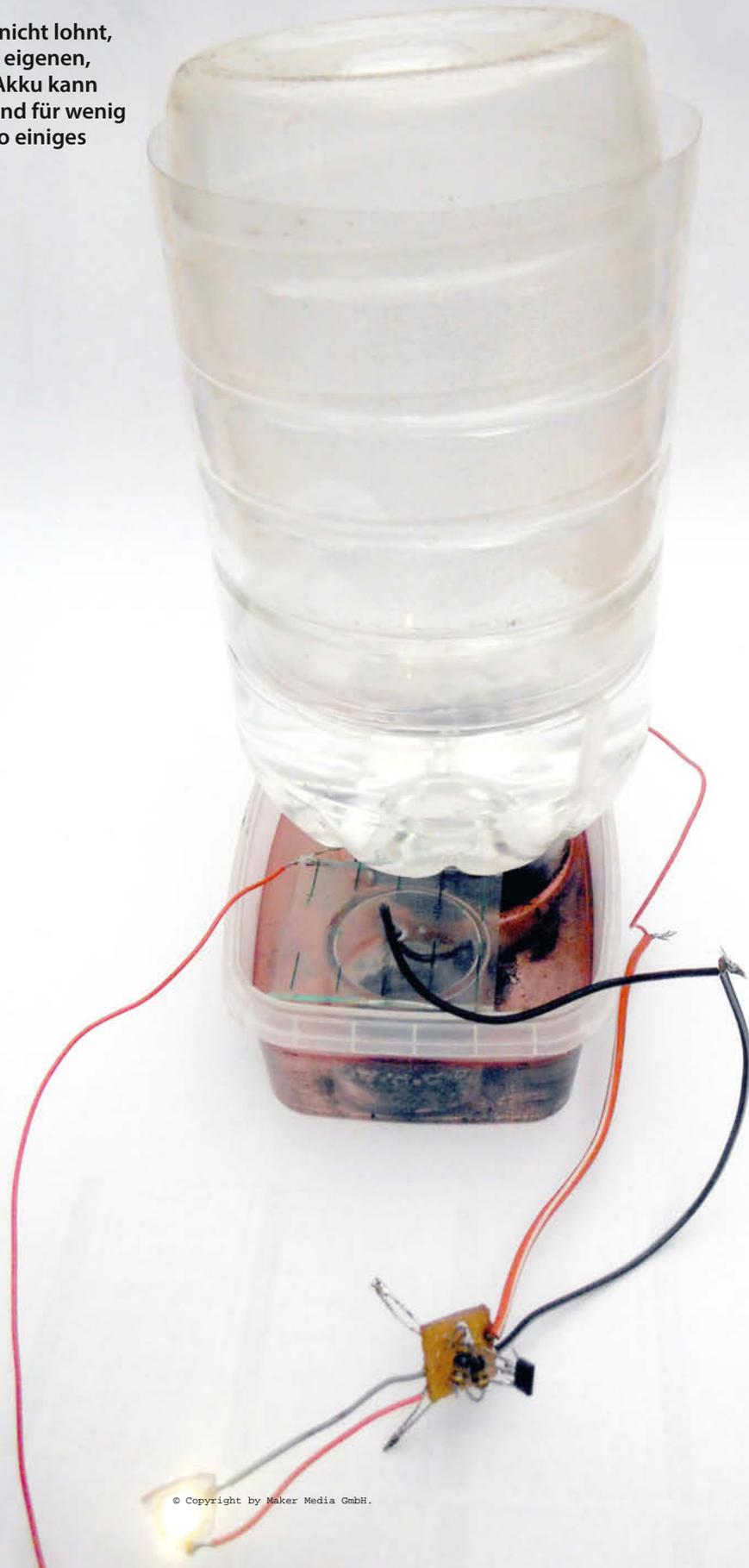


Das Gerät noch ohne Monitor

Zink-Sauerstoff-Akku selber herstellen

Maker denken oft, dass es sich gar nicht lohnt, selbst Akkus zu bauen. Doch einen eigenen, funktionsfähigen Zink-Sauerstoff-Akku kann man schon mit einfachen Mitteln und für wenig Geld herstellen – und dabei noch so einiges über Chemie lernen.

von Harald Winkler



Die Herstellung von Akkus ist eine der Schlüsseltechnologien der Gegenwart. Unzählige Forscher arbeiten in hervorragend ausgestatteten Laboratorien von Universitäten und Konzernen an immer ausgefeilteren Konzepten zur Stromspeicherung. Scheint es da nicht völlig aussichtslos und geradezu vermessen, sich als Hobbybastler an dieselbe Aufgabe zu wagen?

Gute Frage!

Tatsächlich werden wir es kaum schaffen, mit Heimwerkermitteln Akkus zu bauen, die bei gegebener Größe und gegebenem Gewicht eine ähnliche Leistung zur Verfügung stellen können wie moderne Lithium-Akkus. Diese haben sich aufgrund ihrer enormen Leistungsdichte als Traktionsbatterien für Elektrofahrzeuge und für andere mobile Anwendungen weitgehend durchgesetzt. Doch für die stationäre Stromspeicherung, wie sie die Umstellung auf erneuerbare Energien erforderlich macht, sind sie weitaus weniger geeignet.

Diese Anwendung stellt völlig andere Anforderungen an die Stromspeicher: Da sie nicht bewegt werden müssen, ist das Gewicht meist egal. Auch eine hohe elektrische Leistung, also die Fähigkeit, die gesamte gespeicherte Energie in kurzer Zeit freisetzen zu können, ist hier nicht gefragt. Im Gegenteil, das wäre sogar im wahrsten Sinne des Wortes brandgefährlich!

Dagegen rücken andere Kriterien, wie leichte Verfügbarkeit und Umweltfreundlichkeit der Rohstoffe, sowie die Möglichkeit eines vollständigen Recyclings in den Vordergrund. Hier geht es – anders als bei den Batterien für mobile Anwendungen – nicht darum, um jeden Preis in möglichst kurzer Zeit möglichst viel elektrische Energie auf möglichst kleinem Raum bei minimalem Gewicht abrufbereit zu halten, sondern darum, so sicher und billig wie möglich zu speichern. Simpel und einfach ist hier gefragt. Und immer wenn das der Fall ist, hat der Amateurbastler gute Karten.

Einfach selber machen

Die hier vorgestellte Zink-Sauerstoffzelle besteht aus leicht erhältlichen und vergleichsweise harmlosen Materialien: Zink, Holzkohle, Kartoffelstärke, Carbon-Roving (ein Strang aus Kohlefaser-Filament, aus dem Modellbauzubehör), Nickeldraht, Blech, Kalilauge und leeren Plastikverpackungen. Die Kalilauge ist hier der einzige Stoff, der mit einer gewissen Vorsicht gehandhabt werden muss, da er in der erforderlichen Konzentration ätzend wirkt.

Die Funktion der Batterie beruht auf der Reaktion von Zink und Sauerstoff, die den gewünschten Elektronenfluss hervorruft. Diese Reaktion ist umkehrbar: Der in umgekehrter Richtung fließende Ladestrom trennt den

Kurzinfo

- » Minimal-Aufbau für Prototypen
- » Hier entsteht Energie: Kohle reagiert mit Zink in Kalilauge
- » Netter Nebeneffekt: Grillfete veranstalten

Checkliste



Zeitaufwand:

2 bis 3 Stunden plus einige Tage Trockenzeit



Kosten:

30 Euro

Mehr zum Thema

- » Elke Schick, Es funkt zwischen Früchtete und Sonnenlicht, Make 3/15, S. 144
- » Jan Wegener, Lithium-Ionen-Akkus testen und wiederverwenden, Make 2/21, S. 70
- » Florian Schäffer, Akkutechnik, Make Sonderheft 2018, S. 68



Vorsicht

Kalilauge ist stark ätzend, daher gilt es bei der Handhabung, Haut- und Augenkontakt unbedingt zu vermeiden! Schutzbrille und Schutzhandschuhe tragen und Gefahrenhinweise des Herstellers beachten!

Material

- » Bitumendichtmasse z.B. SOS Reparaturband Memeasy
- » Zink etwa alte Auswuchtgewichte von PKW-Felgen, kaputte Teile aus Zinkdruckguss etc.
- » Grillkohle
- » Dünnes Edelstahlblech und kleine Blechreste vom Schrott
- » Plastikflaschen und Schalen
- » Strohhalme
- » Kartoffelmehl oder Kartoffelstärke
- » Nickeldraht
- » Carbon-Roving oder Kohle-Roving
- » Kaliumhydroxid
- » Destilliertes Wasser
- » Holzbrettchen
- » Regelbares Labornetzteil oder Solarpanel und LM2596 Step-Down DC-DC Spannungsregler
- » Sand im Gurkenglas
- » Neodymmagnete
- » LED zum Testen

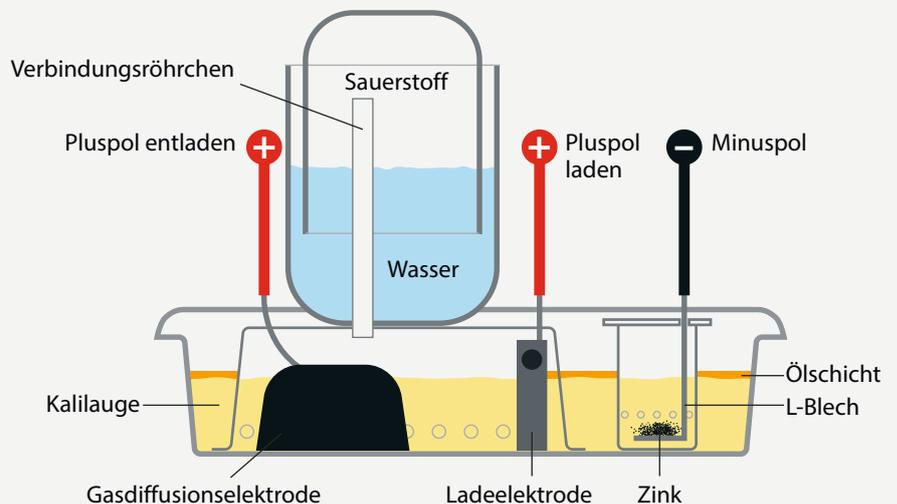
Werkzeug

- » Schere
- » Grill oder Holzofen
- » Bohrmaschine oder Lochzange
- » Lötkolben
- » Heißklebepistole
- » Multimeter

Alles zum Artikel im Web unter make-magazin.de/xjg9

Zink-Sauerstoff-Akku

Schematischer Aufbau in der Plastikschale

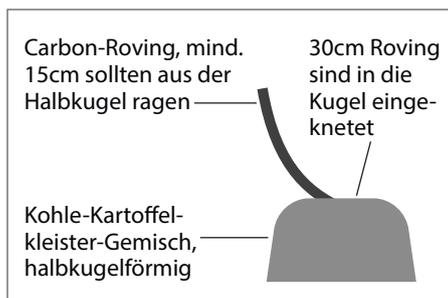




Hübsch anzusehen, aber unpraktisch: Die blattartigen Wucherungen der Dendriten.

Sauerstoff wieder vom Zink und regeneriert so die Zelle für die nächste Entladung.

Da Sauerstoff bekannterweise ein Gas ist, an das man nicht einfach ein Stromkabel anschließen kann, benötigen wir, ähnlich wie in einer Brennstoffzelle, eine sogenannte Gasdiffusionselektrode. Sie vermittelt den elektrischen Kontakt zwischen der Kalilauge – dem flüssigen Elektrolyten unserer Zelle – und dem Stromkabel. Im einfachsten Fall kann das ein simples Stück Holzkohle sein, das allerdings schon einmal im Feuer bis zur Rotglut



Zusammensetzung der Gasdiffusionselektrode



Die gebrannte Elektrode, das Roving schaut auf der Oberseite heraus.

erhitzt worden sein muss. Das geht zum Beispiel auf dem Gartengrill und nebenbei der nächsten Grillparty.

Grillkohle vorbereiten

Unter den brutzelnden Bratwürsten kann man mit der Grillzange vorsichtig ein glühendes Stückchen Holzkohle herausheben, das dann sofort in trockenen Sand oder trockene Asche gesteckt werden sollte. Asche oder Sand sollten sich zweckmäßigerweise in einem luftdicht verschließbaren Gefäß (zum Beispiel ein leeres Gurkenglas mit Schraubverschluss) befinden. Die glühende Holzkohle wird dann so in die Asche oder den Sand gedrückt, dass sie überall bedeckt ist und nirgends das Glas berührt. Das ist wichtig, weil das Glas sonst zerspringen könnte!

Sobald uns diese diffizile Operation geglückt ist, schrauben wir den Blechdeckel fest auf das Gurkenglas. So verhindern wir zuverlässig den weiteren Zutritt von Luftsauerstoff und damit die vollständige Verbrennung des Holzkohlestückchens. Jetzt können wir in aller

Ruhe Bratwurst mit Essiggurken essen, da wir das Glas mit unserer Gasdiffusionselektrode in spe erst am nächsten Tag wieder öffnen sollten – wenn die darin befindliche Glut ganz sicher erloschen und der Inhalt vollständig abgekühlt ist. Wir finden darin dann ein Stückchen Holzkohle, das rein äußerlich auf den ersten Blick genauso aussieht wie vor dieser Behandlung. Der wesentliche Unterschied offenbart sich aber, sobald wir ein Multimeter zur Hand nehmen und den elektrischen Widerstand messen: Während der Widerstand von neu im Supermarkt gekaufter Holzkohle irgendwo im Megaohm-Bereich liegt, messen wir bei der geblühten Kohle einen Widerstand von nur wenigen Ohm. Je weniger, desto besser, denn schließlich soll diese Kohle ja einen Stromfluss vermitteln!

Die Minimalversion

Mit dieser geblühten, leitfähig gemachten Kohle können wir nun, bevor wir uns an einen komplexeren Aufbau wagen, die primitivste Minimalversion einer Zink-Sauerstoff-Batterie herstellen: Wir füllen ein kleines Schälchen mit 20-prozentiger Kalilauge. Diese erhalten wir, wenn wir 20g des handelsüblichen Kaliumhydroxids mit 100ml destilliertem Wasser vermischen. Das Schälchen muss natürlich beständig gegen Lauge sein. PE-Kunststoff ist gut geeignet, PP in den meisten Fällen ebenfalls, Polycarbonat ist ungeeignet. Um welchen Kunststoff es sich handelt, erkennt man am Recyclingcode auf der Schale. Das Symbol ist ein Dreieck aus drei Pfeilen, darunter steht das Material.

Wenn wir nun ein Stück blankgeschmirgeltes Zink und die geblühte Kohle in diese Lauge tauchen, ohne dass sie sich berühren, so können wir zwischen Zink und Kohle eine Spannung von etwa 1,3 bis 1,5 Volt messen. Es lohnt sich, diesen simplen Vorversuch tatsächlich durchzuführen, da er uns mit einigen Tücken der Materie vertraut macht, die es im folgenden zu meistern gilt. Dazu zählt vor allem die Eigenschaft des Zinks, mit dem in der Luft allgegenwärtigen Kohlendioxid zu reagieren und sich mit einer Schicht aus Zinkcarbonat zu überziehen. Diese verhindert eine weitere chemische Reaktion des Zinks und sabotiert so die Funktion unserer Stromquelle. Daher ist es wichtig, sie durch Schmirgeln, Kratzen oder Feilen aufzubrechen, damit die Kalilauge überhaupt an das Zink herantreten kann.

Auch die Kalilauge selbst reagiert mit dem Kohlendioxid. Das führt im Verlauf mehrerer Wochen schließlich dazu, dass sie für unsere Zwecke unbrauchbar wird. Wenn wir also einen langfristig funktionsfähigen Stromspeicher bauen wollen, müssen wir ihn als ein geschlossenes, von unserer Kohlendioxid-haltigen Atmosphäre abgeschirmtes System konstruieren. Was allerdings bedeutet, dass unsere

Gasdiffusionselektrode auch den notwendigen Sauerstoff nicht mehr einfach aus der Luft bekommen kann. Sie benötigt stattdessen einen kleinen Sauerstofftank. In diesem wird der Sauerstoff, der beim Laden entsteht, für den Entladevorgang aufgespeichert.

Dendriten sind ein Problem

Und es gibt noch ein anderes Problem: Wenn wir versuchen, unsere primitive „Minimalzinkluftbatterie“ mehrmals wieder aufzuladen, werden wir feststellen, dass das nicht allzu oft funktioniert. Lassen wir einen Ladestrom fließen, so können wir zwar sehen, dass das bei der Entladung im Elektrolyten gelöste Zink tatsächlich auf die Zinkelektrode zurückwandert. Je länger aber der Ladevorgang andauert, umso deutlicher wird, dass es das in einer sehr eigentümlichen Form tut: Es setzt sich nicht einfach als gleichmäßige, kompakte Zinkschicht ab, sondern wächst in Form verästelter, kleiner Bäumchen auf den nach der Entladung verbliebenen Rest der Zinkelektrode auf.

Diese sogenannten Dendriten sind zwar sehr schön anzuschauen, neigen aber dazu, beim Entladen von der Elektrode abzubrechen. So wird diese schnell kleiner und liegt



Die gereinigten und zugeschnittenen Gefäße

9 JAHRE BASTLER-KNOW-HOW KOMPAKT

NEU
+ portofrei



Das komplette Make-Archiv auf 32 GByte USB-Stick

Auf dem USB-Stick finden Sie 54 Ausgaben c't Hacks (2011 bis 2014) und Make (2015 bis 2020) mit Evergreens wie den Teehasen, die Fräse MaXYposi, die Low-Cost-Wärmebildkamera, Tetris, den DIY-Lötkolben, die Peltiereselampe, die Plotclock, Lixie-Anzeigen, die Reißzweckenorgel und viele weitere zeitlose Projekte zum Nachbauen.

shop.heise.de/make-archiv20

99,90 € >

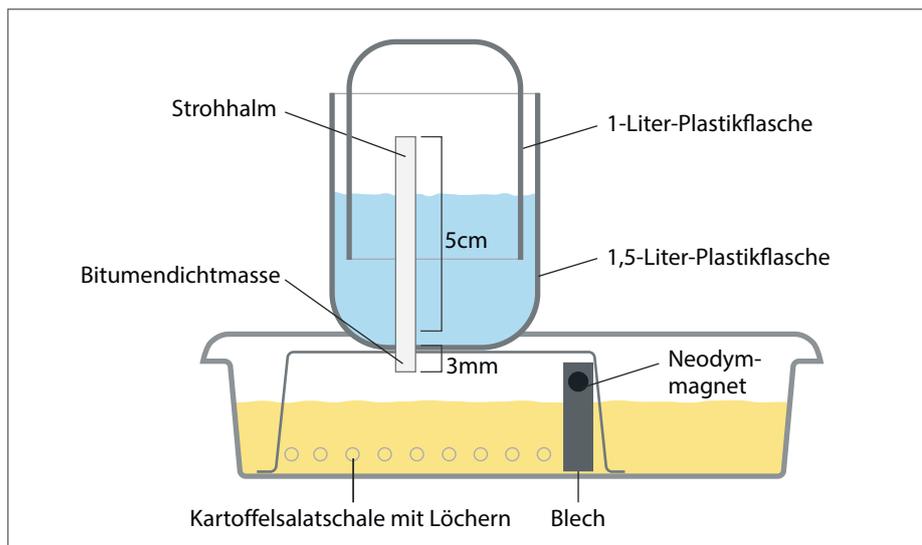
 heise Shop

shop.heise.de/make-archiv20 >

> Generell portofreie Lieferung für Heise Medien- oder Maker Media Zeitschriften-Abonnenten oder ab einem Einkaufswert von 20 €. Nur solange der Vorrat reicht. Preisänderungen vorbehalten.

© Copyright by Maker Media GmbH.





Konstruktion des Sauerstoffspeichers

schließlich als trauriges Häuflein nutzloser Zinkbrösel ohne Kontakt zum Stromanschluss am Boden unseres Versuchsaufbaus. Dieses Problem lässt sich lösen, indem wir die Zinkelektrode in ein kleines Plastikbecherglas setzen, in das ein L-förmiges Kontaktblech gestellt ist. Und zwar so, dass der untere Schenkel des „L“ den größten Teil des Bodens des Becherglases bedeckt und der obere bis kurz unter den Rand reicht. Hier wird der Stromanschluss angelötet. Das Blech kann aus Messing, vernickeltem oder verzinktem Stahl bestehen. Messing blank putzen!

Etwa einen halben Zentimeter über dem Boden des Becherglases bohren wir rundherum Löcher. So können die Kalilauge hineinlaufen und die Zinkionen ungehindert zur Gasdiffusionselektrode wandern. Wenn jetzt die abgebrochenen Dendriten nach unten fallen, liegen sie nicht mehr isoliert auf dem Plastikboden, sondern treffen auf das mit dem Stromanschluss verbundene Kontaktblech und können weiter Strom produzieren. Außerdem verhindert das Becherglas unkontrollierte „Wucherungen“ der Dendriten. Ohne diese Begrenzung könnten sie am Boden der



Seitenansicht: Links befindet sich der Sauerstoffbehälter und die Gasdiffusionselektrode, rechts erahnt man den Behälter mit der Zinkelektrode.

Zelle so lange weiter wachsen, bis sie irgendwann einen Kurzschluss zwischen Zink und Gasdiffusionselektrode herbeiführen.

Kohle-Tücken

Auch die Gasdiffusionselektrode in Form eines simplen, porösen Holzkohlestückchens ist noch verbesserungsbedürftig. Wenn wir unser Multimeter in den Strommessbereich schalten und den Kurzschlussstrom zwischen Kohle und Zink messen, werden wir feststellen, dass er umso größer wird, je fester wir die Messspitzen auf die Kohle drücken. Das liegt daran, dass wir damit den Übergangswiderstand senken. Wenn wir aber zu fest drücken, zerbröseln die Kohle. Oft können wir auch feststellen, dass es große Unterschiede macht, an welcher Stelle wir die Kohle kontaktieren. Das lässt darauf schließen, dass die Kohle nicht homogen ist, sondern aus verschiedenen Zonen unterschiedlicher Innendurchdringung besteht. Diese Zonen sind durch erhöhte Innenwiderstände voneinander isoliert. Es ist auch leicht einzusehen, dass eine größere Kohle größere Ströme produzieren kann. Wünschenswert ist also ein großer Holzkohlebrocken, der von einer möglichst niederohmigen Stromableitung durchzogen wird.

Kohle-Massenproduktion

An dieser Stelle kommen die Kartoffelstärke, der Nickeldraht und der Carbon-Roving ins Spiel. Als erstes schmelzen wir noch mal den Grill an und produzieren diesmal nicht nur ein geglühtes Stück Gasdiffusionskohle, sondern mehrere. Je mehr, desto besser! Wenn wir viele haben, dann können wir mithilfe unseres Multimeters die besten Stücke mit dem geringsten ohmschen Widerstand herausuchen.

Als nächstes legen wir ein sauberes Blatt Papier auf eine feste Unterlage, legen etwas selektierte Kohle darauf und zerklappen sie mit einem Hammer zu Holzkohlepulver. Das Pulver schütten wir dann von dem Papier in einen Becher und nehmen die nächste Kohle, bis wir die gewünschte Menge Pulver zusammen haben. Hier gilt der Grundsatz: Viel Pulver, viel Strom. Aber leider auch viel Arbeit. Daher sollte man an dieser Stelle auch schon die Größe der später für den Akku verwendeten Plastikgefäße im Auge haben: Die Gasdiffusionselektrode muss noch hineinpassen und es muss Platz für eine Ladeelektrode bleiben, da die Gasdiffusionselektrode selbst nicht zum Aufladen verwendet werden sollte.

Kartoffelkleister

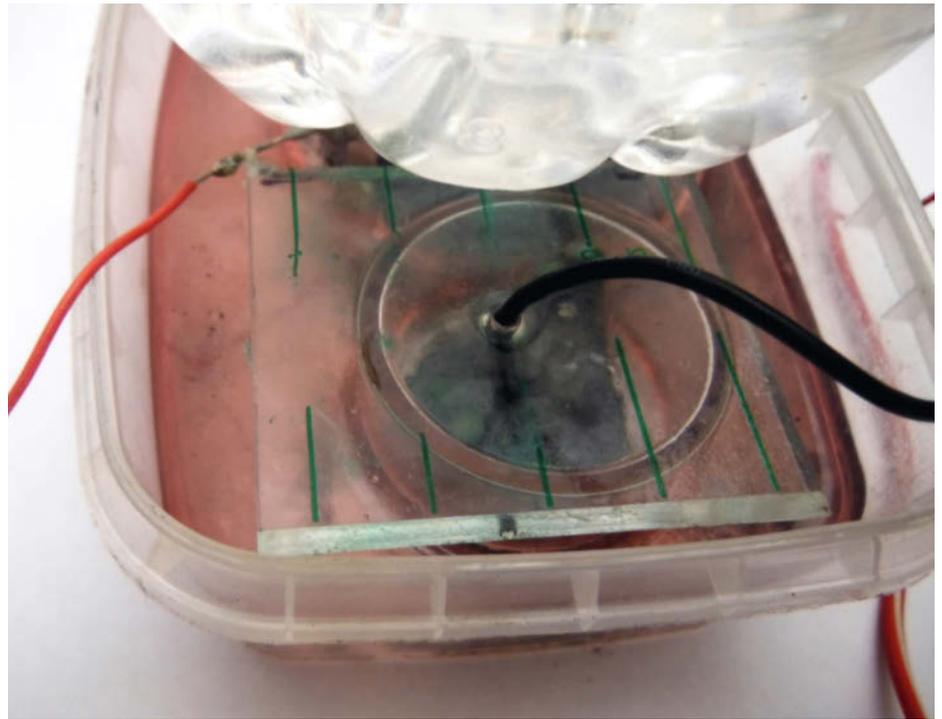
Als nächstes geht es an den Küchenherd, wo wir uns aus Wasser und Kartoffelstärke unter stetem Umrühren einen klebrigen Kartoffelkleister kochen. Sobald dieser gequollen und abgekühlt ist, mischen wir so viel davon unter das Holzkohlepulver, bis eine teigige, gut knet-

bare Masse entsteht. Dann schneiden wir etwa einen halben Meter Carbon-Roving ab, den wir mit einem Feuerzeug oder einem Lötbrenner abflämmen. Dies dient dazu, die Appretur zu entfernen, mit der der Roving beschichtet ist. Sie könnte sonst den Stromfluss behindern. Dann kneten wir etwa 30cm des Rovings sehr gründlich in die Holzkohlemasse. Die einzelnen Fasern sollten möglichst gleichmäßig in der Masse verteilt sein. Es sollte ein teigiger Klumpen entstehen, aus dem noch mindestens 15cm Roving heraussehen. Wenn der Klumpen nicht mehr zusammenhält, ist der Faseranteil zu hoch und es muss etwas weniger Roving eingeknetet werden.

Schließlich den Klumpen auf einer ebenen Unterlage und einem Blatt Papier so plattdrücken, dass eine Halbkugel entsteht. Aus ihrem Scheitel sollte der Carbon-Roving heraussehen. Er bildet später den Stromanschluss. Das ganze etliche Tage gut durchtrocknen lassen – die letzten Tage auf der Heizung oder in der Sonne.

Elektrode brennen

In dieser Zeit kann man aus zwei leeren Konservendosen einen Blechbehälter bauen, in



Der Minuspol (schwarz) ist mit dem Zink verbunden. Der Zinkbecher ist mit einem Plastikdeckel abgedeckt.



Java lernen – von Anfang an wie ein Experte

Sie wollen endlich Programmieren lernen und Ihre ersten Projekte umsetzen? Dann sind Sie hier genau richtig: Java-Experte Michael Inden erklärt die Grundlagen der Java-Programmierung leicht und verständlich.

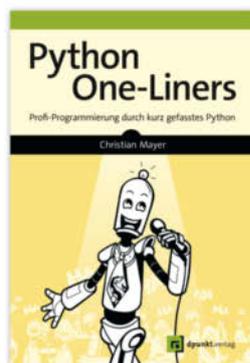
424 Seiten · 22,90 €
ISBN 978-3-86490-852-1

Bundle up!
Print & E-Book
www.dpunkt.de

Oder mit Python durchstarten



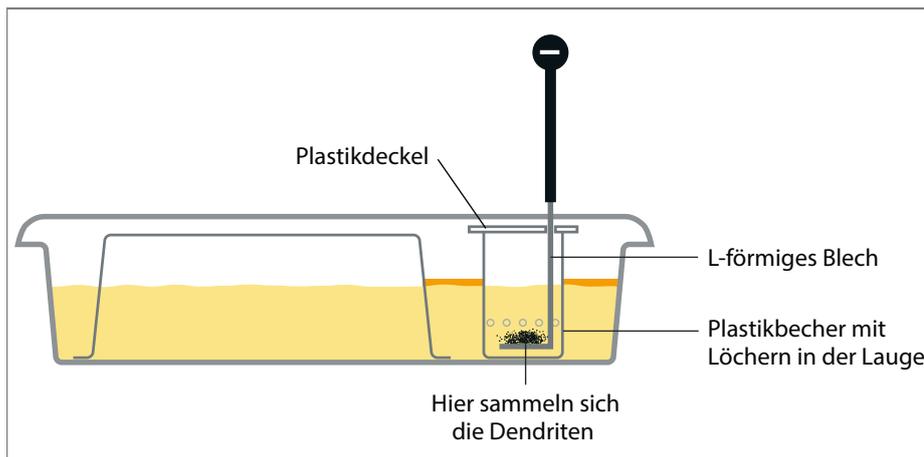
696 Seiten · 34,90 €
ISBN 978-3-86490-753-1



248 Seiten · 29,90 €
ISBN 978-3-86490-805-7



526 Seiten · 34,90 €
ISBN 978-3-86490-809-5



Aufbau der Zinkelektrode

den die getrocknete Halbkugel hineinpasst. Wenn man den Rand der einen Konservendose etwas weitet (Falz bei beiden Dosen abschneiden), dann lässt sie sich auf die andere als Deckel aufsetzen. In dieser Dose wird die Gasdiffusionselektrode schließlich abschließend gebacken, wodurch sie fest und laugenbeständig wird.

Die Dose sollte dabei im Feuer oder in der Glut eines kräftig angefachten Holzkohlegrills bis zu leichter Rotglut erhitzt werden. Diese Temperatur etwa eine knappe halbe Stunde lang aufrechterhalten. In den Deckel sollte dabei ein kleines Loch von etwa einem Millimeter Durchmesser gestochen sein, andernfalls wird er durch die ausgasenden Dämpfe abgesprengt – das kann angesichts der Rotglut gefährlich werden! Sicherheitsabstand einhalten! Falls vorhanden, kann die Dose

auch ganz ungefährlich in einem geschlossenen Holzofen oder in einem Brennofen für Töpferwaren erhitzt werden.

Hat die Elektrode das Brennen heil überstanden, so folgt noch ein letzter wichtiger Schritt zur Fertigstellung: Sie wird auf eine ebene Fläche gestellt und von oben (das ist die Seite, auf der der Carbon-Roving rauskommt) mit einem kurzen Stoß Imprägnierspray eingesprüht. Das ist das sogenannte *Hydrophobieren* oder auf deutsch *wasserabstoßend machen*. Dadurch wird verhindert, dass sich die Elektrode bei längerer Betriebsdauer vollständig mit Kalilauge vollsaugt. Das hätte zur Folge, dass kein Sauerstoff mehr eindringen kann. Es darf aber auch nicht so viel imprägniert werden, dass gar keine Lauge mehr an die Elektrode herankommt. Deshalb muss die Unterseite frei von Imprägnierspray bleiben.



Wenn alles passt, leuchtet die LED

Jetzt kann der Carbon-Roving vorsichtig an ein Stück Nickeldraht angeschlossen werden. Dazu werden die letzten 10cm des Drahtes umgebogen, der Roving durch die so entstandene Schlinge geführt und etwa 3cm lang um die beiden parallel liegenden Drahtabschnitte gewickelt. Dann bleiben noch 7cm Draht übrig. Dieser Rest biegt man noch einmal um und wickelt ihn in entgegengesetzter Richtung über diese Carbonwicklung, die dadurch fixiert wird.

Plastikgefäße vorbereiten

Als nächstes widmen wir uns den Plastikgefäßen. Eine Plastikschaale, wie sie zur Verpackung von 500g Hackfleisch verwendet wird, hat etwa die Abmessungen 20cm x 14cm x 5cm. Darin findet bequem ein auf den Kopf gestelltes Schälchen von 11cm x 8cm x 4cm (Kartoffelsalatverpackung) Platz. Und es bleibt noch genug Raum für die Abdeckkappe eines Deosprays, welche das Becherrchen für die Zinkelektrode bildet.

Über dem Kartoffelsalatschälchen thront stolz die untere Hälfte einer 1,5l-Getränkeflasche, in die die umgedrehte untere Hälfte einer 1l-Salatölflasche geschmeidig hineingleiten kann. Diese Flaschen bilden den Sauerstoffspeicher. Durch passende Bohrungen in der 1,5l-Flasche und dem Kartoffelsalatschälchen steckt man einen Strohhalm, der etwa 3mm in das Schälchen und etwa 5cm in die Flasche hineinragt. Gegen beide Bohrungen wird er mit Bitumendichtmasse abgedichtet.

Dann stanzt man mit einer Lochzange in etwa 5mm Abstand zum Rand umlaufend Löcher mit etwa 4mm Durchmesser in das Kartoffelsalatschälchen. Der Lochabstand beträgt etwa 1cm. In den Boden des Schälchens werden zwei kleine Löcher für den 0,5mm Nickeldraht gestochen. Durch eins der Löcher führt man den Anschlussdraht der Gasdiffusionselektrode und dichtet ihn mit Bitumendichtmasse ab. Dann führt man durch das zweite – genauso abgedichtete – den Anschlussdraht für die Ladeelektrode, die aus einem einfachen Streifen Edelstahlblech besteht und mit diesem durch ein M3 Edelstahlschraubchen verbunden werden kann.

Blech präparieren

Das Blech sollte so platziert werden, dass es mit der einen Seite fast bis zum Boden reicht und später in die Kalilauge eintauchen kann, mit der anderen Seite aber oberhalb der Lauge im Trockenen steht. Mit Hilfe zweier kleiner Neodymmagnete kann es am Plastik verschiebbar fixiert werden. Die Magnete sollten im Trockenen bleiben, um Korrosion zu vermeiden. Auch der Nickeldraht sollte im trockenen Bereich angeschlossen werden.

Das Edelstahlblech sollte möglichst eng an der Wandung des Kartoffelsalatschälchens an-

liegen und nicht zu weit in dieses hineinragen, da sich später auch die Gasdiffusionselektrode unter diesem Schälchen befindet und diese das Edelstahlblech nicht berühren darf! Der Grund dafür ist die Sauerstoffentwicklung beim Laden: Frisch elektrolysiertes, sogenannter *naszierender* Sauerstoff ist nämlich hochreaktiv und würde mit dem Kohlenstoff der Gasdiffusionselektrode zu Kohlendioxid reagieren. Die Zerstörung der Elektrode wäre die Folge. Deshalb ist das sauerstoffbeständige Edelstahlblech als Anschluss für den Ladestrom erforderlich, der nicht über die Kohleelektrode fließen darf!

Nun wird diese Baugruppe so in die größere Schale gestellt, dass das Kartoffelsalatschälchen wie eine Kuppel die Gasdiffusionselektrode und die Ladeelektrode abdeckt. Über der Kuppel, fixiert durch den durchgesteckten Strohhalm und die Dichtungsmasse, steht die halbe 1,5l-Flasche, die nun einige Zentimeter hoch mit Wasser gefüllt werden kann. Der Strohhalm muss aber noch aus dem Wasser heraus schauen. Nun steckt man kopfüber – wie eine kleine Taucherglocke – den abgeschnittenen Unterteil der 1l-Flasche in dieses Wasser. Der Innenraum dieser Glocke ist jetzt über den Strohhalm mit dem Innenraum des Kartoffelsalatschälchens verbunden. Die größere Schale wird zweckmäßigerweise auf einem Holzbrettchen mit etwas Heißkleber angeklebt. An diesem Brettchen können später auch die Kabel der drei Stromanschlüsse fixiert werden.

Kalilauge los

Als nächstes wird die Kalilauge in die größere Schale eingefüllt, etwa 2cm hoch. Die 4mm-Löcher müssen vollständig unterhalb der Lauge liegen. Dann ist der Innenraum vollständig von der äußeren Atmosphäre isoliert. Wenn sich nun später beim Laden an der Ladeelektrode in Form kleiner Bläschen Sauerstoff entwickelt, gelangt dieser durch den Strohhalm unter die Glocke und hebt diese langsam aus dem Wasser. Wird dieser Sauerstoff beim Entladen verbraucht, sinkt sie wieder weiter ein. Somit ist sie gleichzeitig Gasspeicher und Ladestandsanzeige.

Die Gasdiffusionselektrode sollte etwa zur Hälfte in die Lauge eintauchen. Sollte sie zu klein geraten sein, und „ertrinken“, kann einfach etwas Plastik untergelegt werden, bis sie wieder auftaucht.

Jetzt wird das Becherglas für die Zinkelektrode in die Lauge gesetzt. Durch die Bohrungen kann sie hineinlaufen. Alle Bohrungen müssen komplett unter der Lauge sein. Aus einem ebenen Stück transparentem Plastik (z.B. Deckel des Kartoffelsalatschälchens) schneidet man nun noch einen Deckel für das Becherglas. Er bekommt an passender Stelle ein Loch, durch das das Anschlusskabel

geführt wird. Es wird am Holzbrettchen fixiert und stellt den Minuspol unseres Stromspeichers dar. Die Anschlusskabel von der Ladeelektrode (Pluspol Ladung) und von der Gasdiffusionselektrode (Pluspol Entladung) ebenfalls am Holzbrettchen fixieren.

Es werde Ladung

Jetzt kann es losgehen: Für den ersten Zyklus müssen wir Zink auf das L-Blech legen. Blech und Zink sollten blank sein, um guten elektrischen Kontakt zu ermöglichen. Um Kontaktprobleme zu umgehen, kann man auch für den ersten Zyklus eine Kupferlitzte an das Zink löten und diese provisorisch mit dem Minuspol verbinden. Nach dem ersten Aufladen ist dieses Kabel dann überflüssig, weil sich während des Ladevorgangs frisches Zink direkt auf dem Blech abscheidet und so einen guten Kontakt herstellt.

Jetzt sollte zwischen Minuspol und Entladepol eine Leerlaufspannung von etwa 1,3V zu messen sein. Der dauerhafte Entladestrom beträgt etwa ein Ampere pro Liter Gasdiffusionselektrode. Das heißt, eine Elektrode der beschriebenen Größe kann etwa eine weiße LED eines typischen Solarlämpchens betreiben, wenn der entsprechende Step-Up-Wandler (zum Beispiel QX 5252f) verwendet wird.

Die Ladung erfolgt mit etwa 2,5 Volt ausschließlich über den Ladepluspol. Eine Spannung von 2,5 Volt kann man entweder über ein einstellbares Netzgerät erreichen oder von einem Solarpanel bekommen. Dabei kann man entweder ein kleines Panel mit 5 bis 6 Zellen verwenden – wie es zahlreich in solare Gartenlämpchen verbaut wird und das direkt die passende Spannung liefert – oder man kann die Spannung eines großen Solarpanels mit 36 Zellen mithilfe eines Step-Down-Reglers auf 2,5 Volt begrenzen. Auf dem Step-Down-Regler sind vier Anschlüsse, *In+*, *In-*, *Out+* und *Out-*. *In+* wird mit dem Pluspol eines beliebigen Solarmoduls mit einer Eingangsspannung zwischen 5 Volt und 36 Volt verbunden, *In-* mit dem entsprechenden Minuspol. An *Out+* und *Out-* lässt sich dann mithilfe des Potis auf der Platine eine Spannung von 2,5 Volt einstellen.

Bei der Ladung kann die Bildung der Zinkendriten und das Füllen des Sauerstoffspeichers beobachtet werden. Wenn alles funktioniert, sollte man abschließend noch etwas Mineralöl als Schutzschicht gegen Kohlendioxid auf die Kalilauge gießen.

Die Leistung der Gasdiffusionselektrode lässt sich übrigens noch deutlich steigern, wenn sie galvanisch mit einer dünnen Silberschicht überzogen wird, die als Katalysator wirkt. Dazu wird sie für einige Minuten in Silbernitrat getaucht, und mit dem Minuspol einer 5V Gleichstromquelle verbunden. Ein Silberstift, der ebenfalls ins Silbernitrat taucht, kommt an den Pluspol. —rehu

NEU
und exklusiv
im heise shop

Klein – aber oho!

Mit MicroPython spielend
leicht Mikrocontroller
programmieren



Heft + PDF
mit 29%
Rabatt

Make MicroPython Special

Diese Make-Sonderausgabe zeigt Ein- und Umsteigern, wie man mit MicroPython leicht und schnell eigene Projekte mit dem ESP32 umsetzt. Wie immer in Make Specials geht's sofort in die Praxis mit Audio-Projekten, einer CO2-Ampel und mehr.

Auch als einzelnes Heft erhältlich.

shop.heise.de/make-micropython

Heft + PDF
für nur

19,90 € >

Generell portofreie Lieferung für Heise Medien- oder Maker Media Zeitschriften-Abonnenten oder ab einem Einkaufswert von 20 €. Nur solange der Vorrat reicht. Preisänderungen vorbehalten.

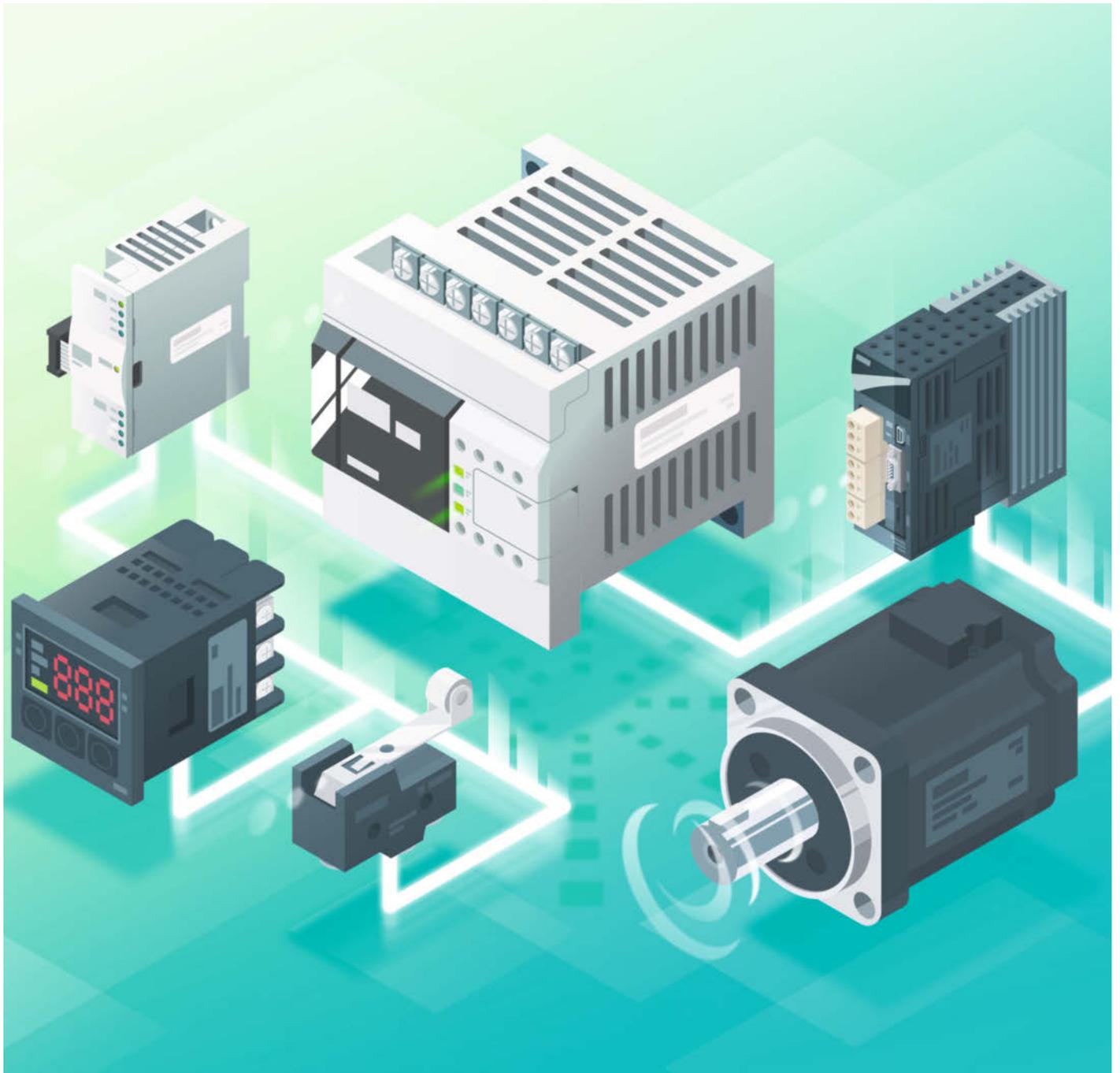
heise Shop

shop.heise.de/make-micropython

Maker-Hardware in der Industrie

Der Weg von Arduino, Raspberry Pi & Co. in die Industrie
und wie Maker davon profitieren könnten.

von Stephan Feldker



Arduino, Raspberry Pi und Co. erfreuen sich seit ihrer Entstehung großer Beliebtheit. In Fachzeitschriften und auf einschlägigen Internet-Portalen wird eine schier endlose Zahl erfolgreich umgesetzter Projekte präsentiert. Egal ob Hausautomation, mobile Roboter, Gaming oder medizinische Anwendungen wie gesteuerte Prothesen – der Kreativität von Makern sind (fast) keine Grenzen gesetzt.

In der Industrie-Automation konnten sich diese Boards aber (noch) nicht durchsetzen und waren bislang keine ernstzunehmende Konkurrenz für etablierte Steuerungssysteme.

Woran das liegt und ob sich dies in Zukunft ändern wird – damit beschäftigen wir uns in diesem Artikel. Wir werfen einen Blick auf sogenannte *industrietaugliche Varianten* von Arduino und Raspberry Pi und wollen klären, was dieses *industrietauglich* eigentlich bedeutet. Am Beispiel des *Controllino* wird gezeigt, wie man einen „Industrie-Arduino“ programmiert und wie Projekte mit dieser Low-Cost-SPS umgesetzt werden können.

Industrie vs. Arduino

Was unterscheidet Steuerungen für die Industrie von Maker-Hardware? Nur der höhere Preis? Die Industrie setzt bei Steuerungen vor allem auf Geräte namhafter Hersteller. Eine Umfrage aus 2018 unter 300 Maschinenbauunternehmen in Deutschland (alle Links in der Kurzinfor) zeigt, dass die bekanntesten Hersteller solcher Geräte Siemens, Beckhoff, Rockwell und B&R sind. Siemens **1** dominiert den deutschen Markt.

Seit sie vor über 50 Jahre das Licht der Welt erblickt hat, hat sich die *Speicherprogrammierbare Steuerung* (kurz SPS) in einem breiten Anwendungsfeld durchgesetzt und fest verdrahtete „Klappertechnik“ (Relais-basiert) sowie Lochkartengeräte vollständig abgelöst. Teilweise findet sich im deutschen Sprachgebrauch auch die Bezeichnung PLC, also die Abkürzung für die englische Bezeichnung *Programmable Logic Controller*.

Kurzinfor

- » **Arduino und Raspberry Pi in der Industrie**
- » **Wie man Maker-Hardware standardisiert**
- » **Controllino in PlatformIO programmieren**

Checkliste



Zeitaufwand:
1–2 Stunden



Kosten:
ab 115 Euro

Material

- » **Controllino** z. B. *mini* oder *mini pur*
- » **Taster, USB-Kabel, ggf. LED mit Vorwiderstand**

Werkzeug

- » **Schraubendreher** für die Kontaktklemmen
- » **Visual Studio Code** mit PlatformIO-Erweiterung

Mehr zum Thema

- » Peter König, Kurzvorstellung Controllino Make 4/19, S. 122
- » Florian Schäffer, Die richtige IDE für Sie, Make 2/19, S. 66

Alles zum Artikel im Web unter make-magazin.de/x7qj

Die Geräte der genannten und weiterer Hersteller haben sich seit Jahrzehnten bewährt und der Begriff *SPS* ist zum Synonym für robuste zuverlässige Steuerungstechnik geworden. Seit Jahrzehnten ist auch die Art, wie eine SPS programmiert wird, mit der einschlägigen Norm (IEC 61131-3) quasi in Stein gemeißelt.

Innerhalb dieser Rahmenbedingungen gab es bislang wenig Raum für Plattformen wie Arduino & Co, die bei Makern beliebt sind. Allerdings deuten aktuelle Trends darauf hin, dass sich der Markt zunehmend und aus zwei Richtungen verändert. Zum einen bieten Hersteller ihre Maker-Hardware als Industrie-Variante an und zum anderen integrieren Hersteller von Industrie-Systemen Open Source und Open Hardware oder stellen gänzlich auf

offene IT-Ökosysteme um. Das Totschlagargument Arduino & Co. seien nicht *Industrie-Standard* lässt sich zwar bei Anwendern und Entscheidern in der Industrie durchaus noch wahrnehmen, langsam aber sicher scheint diese Wahrnehmung aber zu bröckeln und es findet ein Umdenken statt. Von dieser Entwicklung können auch Maker, Arduino- und Raspberry Pi-Enthusiasten profitieren. Siemens hat bspw. vor einiger Zeit sein Portfolio um IoT-Geräte mit Linux Betriebssystem erweitert, die einen Steckplatz für Arduino-Shields beinhalten und dessen Programmierung geübten Raspbian-Entwicklern leichtfallen sollte.

Dies kann auch damit begründet sein, dass die Welt der Steuerungssysteme bunter wird und Automatisierer heute vor der Qual der

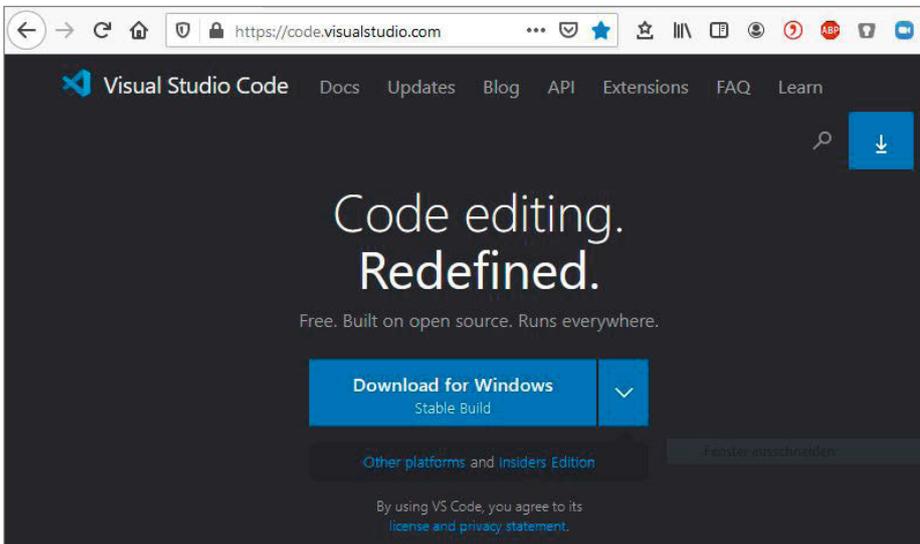


1 Die Produktfamilie Siemens S7 1500

Siemens AG



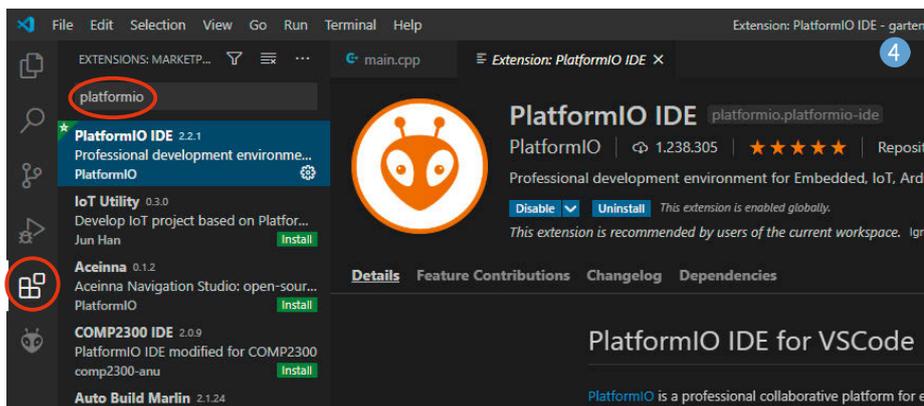
2 Controllino Maxi (links) und Revolution Pi (rechts)



3 Download von Visual Studio Code für Windows

Wahl stehen zwischen SPS, Industrie-PC, Edge-Computer oder IoT-Device. Industrie-Systeme werden offener und die Anschaffung teurer Entwicklungsumgebungen kann teilweise entfallen. Die unter 1 gezeigten Steuerungen werden mit dem Programm

TIA-Portal von Siemens programmiert. TIA steht für *Totally Integrated Automation*, quasi Nachfolger von STEP7. Die Kosten für eine Entwicklungsumgebung liegen schnell bei mehreren hundert oder tausend Euro. Alternative zu diesen Systemen sind klassische



DIY-Plattformen wie Arduino und Raspberry Pi, die als industrietaugliche Version herausgebracht werden. Als IDE stehen den Makern dann die bekannten Tools zur Verfügung.

Arduino in die Industrie!

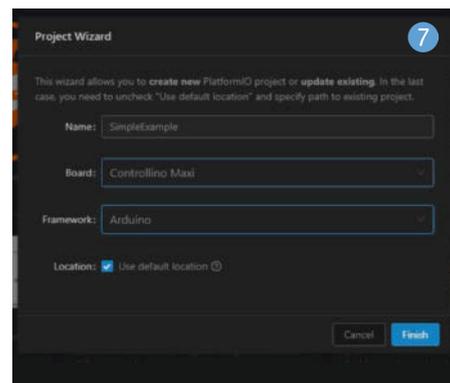
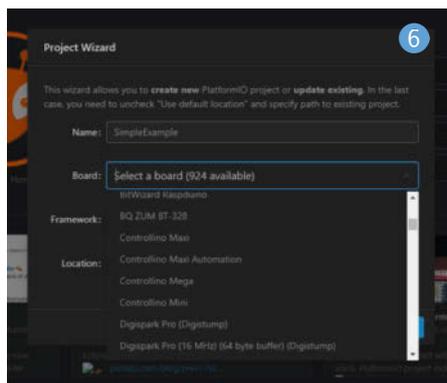
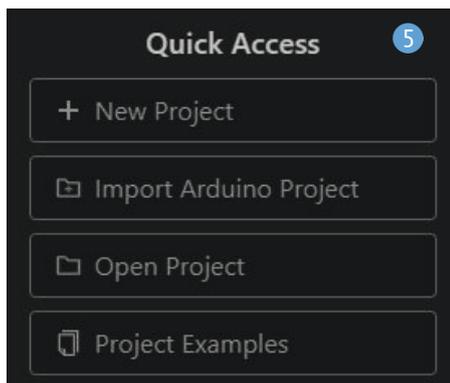
Beispiele für industrietaugliche Versionen der bekannten Maker-Plattformen sind *Controllino*, *Revolution Pi*, *Industruino* oder *Pixtend* (Links in Kurzinfo).

Dabei haben die Hersteller Arduino und Raspberry Pi nicht einfach in schicke Industrie-Gehäuse verfrachtet, sondern modulare Plattformen mit geeigneten Schnittstellen entwickelt. Üblicherweise sind dies digitale Signale mit 24V DC und nach DIN IEC 60381-1/-2 genormte analoge Strom- und Spannungssignale (z. B. 4 – 20mA, 0 – 10V, +/- 10V). Darüber hinaus werden zur Kommunikation in der Industrie-Automation Feldbusse, Industrial-Ethernet und Funk-Technologien eingesetzt. Die in Deutschland am häufigsten eingesetzten Standards sind *Profibus DP* (Feldbus) und *Profinet* (Industrial Ethernet).

Entstehung und Funktionen des Controllino

Am Beispiel des Controllino soll im folgenden gezeigt werden, wie einfach die Arbeit mit diesem Controller ist. Der *Controllino* baut auf der Plattform Arduino auf und verwendet dieselben Prozessoren wie der *Arduino Uno* und der *Arduino Mega*. Der Entwickler selbst bezeichnet den Controllino als „eine industrialisierte Variante des Arduino“ (siehe Links). Die Entstehung geht auf das Jahr 2014 zurück, als der Controllino-Urheber Marco Riedesser eine defekte Kaffeemaschine mittels Arduino reparieren und optimieren wollte. Er stand vor mehreren Herausforderungen: Zusätzliche Peripherie zum Schalten größerer Spannungen und Ströme sowie eine fachmännische Verdrahtung und Einhausung waren notwendig. Dies war die Geburtsstunde des Controllino.

Der Controllino ist aktuell in den Versionen *Mini*, *Mini pure*, *Maxi* und *Mega* verfügbar. Bezugsquelle ist unter anderem *Conrad Electronic*. Die Entwicklung erfolgt durch das österreichische Unternehmen *CONELCOM GmbH*. Der Controllino macht aus dem Arduino eine SPS – mit allem, was dazugehört. So sind alle Kontakte, die nach außen geführt werden, kurzschlussicher. Der Controllino ist gegen elektrostatische Auf- und Entladungen (ESD) geschützt und verfügt über ein Gehäuse, welches für die Montage auf einer Hutschiene geeignet ist. Eine Spannungsversorgung ist über 12V oder 24V möglich. Die Spannung der Eingänge passt sich automatisch an die Versorgungsspannung an (kein Jumpers notwendig). Damit ist der Controllino von Hause aus für 24V-Anwendungen (z. B. Automation



von Maschinen und Anlagen) sowie für 12V-Anwendungen (z. B. Automation mobiler Systeme) geeignet. Über potenzialfreie Kontakte (Relais, Schütz) können höhere Lasten geschaltet werden. Die Verdrahtung erfolgt über Schraubkontakte. Alle Versionen verfügen über eine USB-Schnittstelle. Die Versionen *Maxi* und *Mega* verfügen zusätzlich über eine Ethernet-Schnittstelle.

Hochsprache oder grafische Programmierung?

Bevor näher auf die Programmierung des Controllino eingegangen wird, noch ein Blick auf die eingangs bereits genannte Programmier-Norm IEC 61131-3. Die hier spezifizierten Programmiersprachen sind *Anweisungsliste (AWL)*, *Kontaktplan (KOP)*, *Funktionsplan (FUP)*, *Ablaufsprache (AS)* und *Strukturierter Text (ST)*. Die Programmierung erfolgt dabei grafisch und ist angelehnt an den Stromfluss wie in einem *Stromlaufplan*, an logische Gatterschaltungen wie aus der Digitaltechnik bekannt (FUP), oder an die Beschreibung eines Ablaufes ähnlich denen eines Flussdiagramms (AS).

Diese Art der Programmierung steht auch für den Controllino zur Verfügung. Die Arduino-Plattform wird dadurch auch für Personen interessant, die in der Welt der Hochsprachen nicht zuhause sind, aber Erfahrungen in der SPS-Programmierung nach IEC-Norm mitbringen.

Übrigens spezifiziert die IEC mit *Strukturierter Text* auch einen an Hochsprachen angelehnten Standard. Die verbreitete Vorgehensweise bei der Programmierung einer SPS entspricht allerdings der grafischen Programmierung, bei der ggf. einzelne Funktionen (z. B. mathematische Gleichungen) durch *Strukturierten Text* programmiert werden. Die *Anweisungsliste (AWL)* entspricht einem Befehlssatz und findet in Einzelfällen noch Verwendung, kann (und sollte) aber vollständig durch *Strukturierten Text* ersetzt werden.

Für den Controllino steht „klassischen“ SPS-Programmierern das Tool *logi.CAD 3* zur

Seite (siehe Links). *logi.CAD 3* bietet die Möglichkeit einer grafischen Programmierung mit dem *Funktionsplan (FUP)* sowie mit *Strukturierter Text (ST)*. Das Tool basiert auf der bekannten Entwicklungsumgebung *Eclipse* und ist laut Website „das erste Engineering-Toolkit nach IEC 61131-3 für die Programmierung von Controllino-Steuerungen“.

Programmierung in C/C++ mit PlatformIO

Entwickler, die sich für den Einsatz des Controllino in ihrem Projekt entscheiden, stehen also vor der Wahl: grafische Programmierung oder Hochsprache?

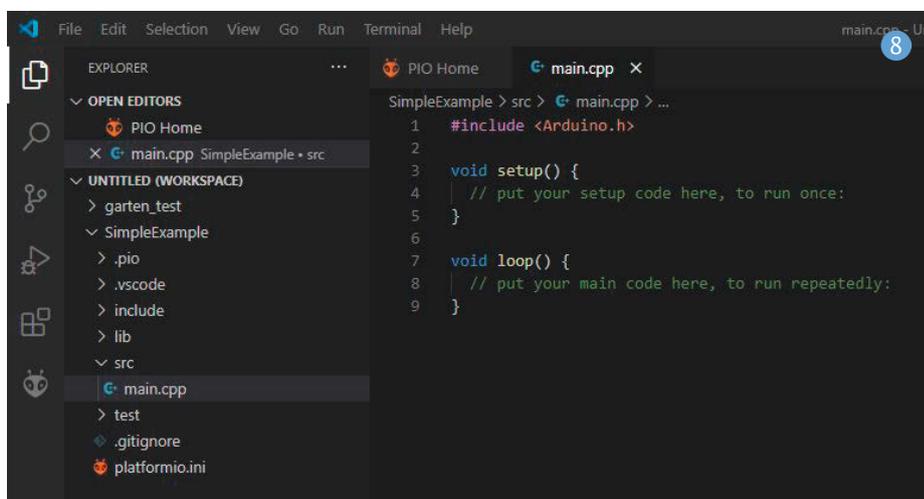
Da es sich beim Controllino um eine Arduino-Plattform handelt, stehen Arduino IDE und Arduino Bibliotheken zur Verfügung. Folgerichtig ist auch eine Programmierung mit anderen für Atmel-Mikrocontrollern geeigneten Tools möglich (z. B. Atmel Studio). Der Controllino ließe sich also theoretisch auch in Assembler oder Basic programmieren.

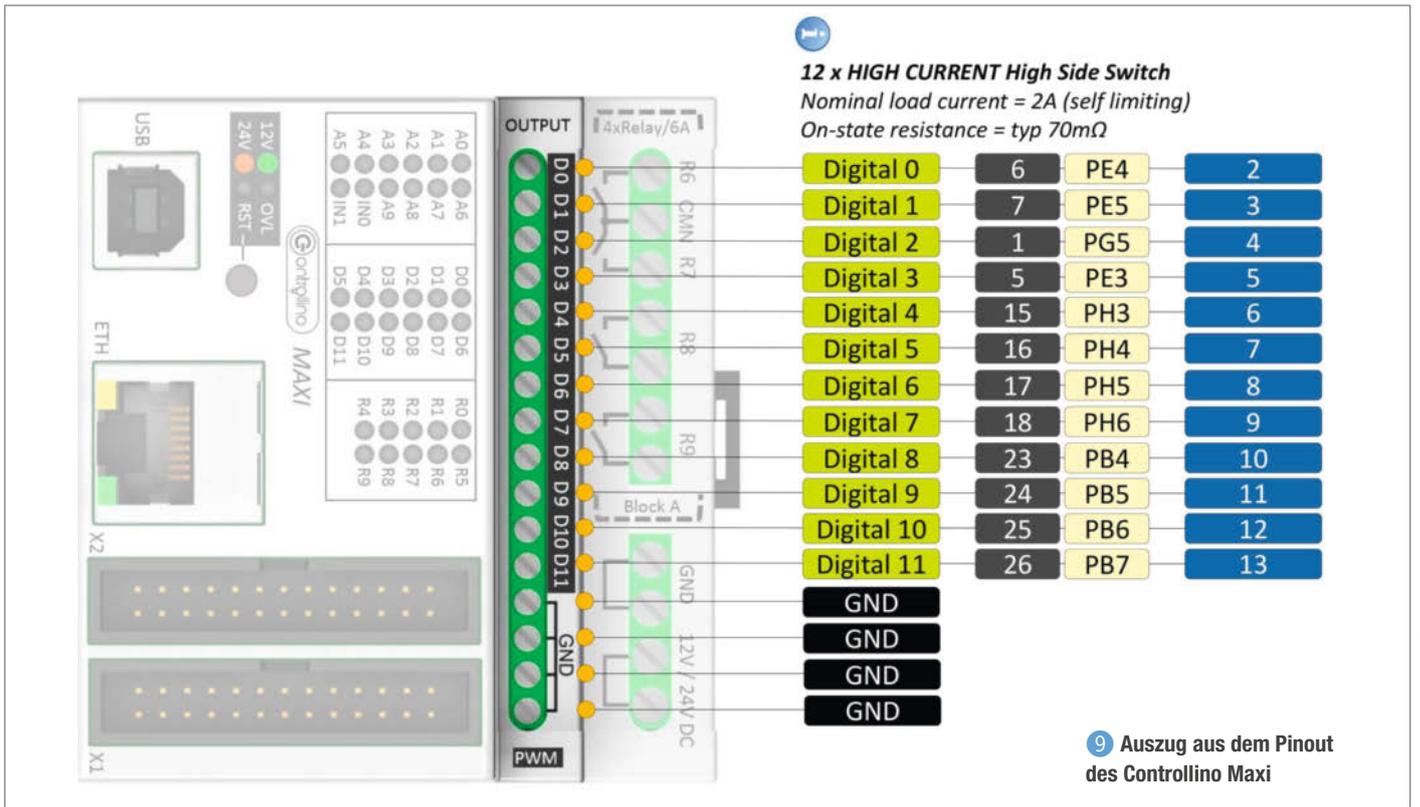
Ein Entwicklungswerkzeug, auf das sich ein näherer Blick lohnt, ist *PlatformIO*. Es handelt sich um ein professionelles Cross-Plattform-

Tool für Embedded-Systems-Entwickler (siehe Links), dass die Erstellung von Quellcode für verschiedene Mikrocontroller erleichtert, u. a. auch für den Arduino (und damit auch den Controllino). Die größte Verbreitung hat PlatformIO als Plug-in für *Visual Studio Code (VSC)*. VSC bietet gegenüber der rudimentären *Arduino IDE* mehrere Vorteile (bessere Organisation des Codes, gutes Syntax-Highlighting, Auto-Vervollständigung) und lohnt sich besonders für umfangreichere Projekte. Eine parallele Nutzung der Arduino IDE bietet sich unter Umständen an, um zügig kleinere Funktionen oder ein paar schlanke Programmzeilen ohne Ballast zu testen.

Im Folgenden wird an einem einfachen Beispiel die Programmierung des Controllino mit Visual Studio Code und PlatformIO in C/C++ erläutert. VSC kann kostenlos heruntergeladen ³ werden. Visual Studio Code (nicht zu verwechseln mit Visual Studio) ist ähnlich wie Eclipse eine offene Plattform, die für die Entwicklung in verschiedenen Sprachen und für mehrere Betriebssysteme geeignet ist.

Nach dem Einbinden von PlatformIO lässt sich das Plugin über das kleine Icon ⁴ im linken Menüband öffnen.





Controllino Maxi Pinout V1.1

Ein neues Projekt kann z.B. über den Button *New Project* im *Quick Access* 5 auf dieser Seite angelegt werden.

Es öffnet sich der *Project Wizard* 6. Über das Dropdown-Menü lassen sich eine Vielzahl an unterschiedlichen Plattformen für die Programmierung und PlatformIO auswählen.

In unserem Beispiel wird der Controllino Maxi 7 ausgewählt.

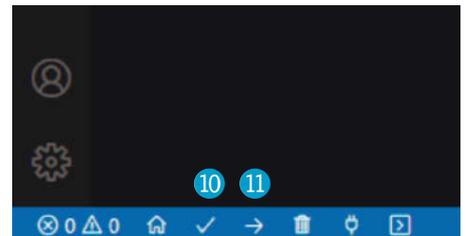
Damit steht das Framework 8 und wir können mit der Programmierung loslegen. Die Programmstruktur mit `setup()` und `loop()` ist identisch mit der Programmierung des Arduino in der Arduino IDE.

Im konkreten Beispiel dient der Controllino als Bewässerungscomputer für den heimischen Garten. Es können Ventile direkt über die digitalen Ausgänge mit der Versorgungsspannung (12VDC oder 24VDC) geschaltet werden oder mit potentialfreien Kontakten flexibel über die Fremdspannung passend zum gewünschten Ventil. Entsprechend den Möglichkeiten eines Arduino kann der Controllino um Hardware erweitert werden, die in vielen Arduino-Kits mitgeliefert werden oder über die einschlägigen Shops bezogen werden können. Im Beispiel wurde der Controllino um eine *Real Time Clock (RTC)*, Inkrementalgeber und LED-Matrix ergänzt. Die RTC ermöglicht zuverlässig das zeitgesteuerte Ein- und Ausschalten der Ventile, die Inkrementalgeber dienen als Drehschalter zur Auswahl von Ventil und Zeitraum und die LED-Matrix zeigt das Menü und den aktuellen Zustand. Alle diese Funktionen können über

die bekannten Arduino-Beispiele und -Bibliotheken eingebunden werden. Den Unterschied zur Programmierung des Arduino stellt das Lesen und Schreiben der digitalen und analogen Ein- und Ausgänge dar. Der Controllino hat eigene Bezeichnungen der Ein- und Ausgänge. Ein Auszug aus der Controllino-Pinbelegung ist in Abbildung 9 dargestellt.

Das Pinout zeigt die verschiedenen Bezeichnungen für digitale Ausgänge. Das Arduino-Pinout ist blau, das Controllino-Pinout grün dargestellt. Der Ausgang *Digital 0* am Controllino entspricht dem *Pin 2* am Arduino (sowie dem *Pin 6* am Atmel-IC, bzw. dessen Port *PE4*), der Ausgang *Digital 1* am Controllino entspricht dem *Pin 3* am Arduino, usw. Die Nutzung der Controllino-Bibliothek *Controllino.h* hat den Vorteil, dass direkt die Bezeichnungen der Controllino-Ein- und Ausgänge im Quellcode adressiert werden können. Ein einfaches Quellcode-Beispiel zum Umschalten des Ausgangs *Digital 0* am Controllino durch einen Taster am Eingang *Analog 0* ist in Listing *Tastenabfrage* dargestellt.

Durch das Einbinden von *Controllino.h* können direkt die Ein- und Ausgänge des Controllino adressiert werden (z. B. `digitalRead(CONTROLLINO_A0)`). Die Ein- und Ausgänge des Controllino können auch mit der Arduino-Pin-Belegung angesprochen werden, allerdings ist in diesem Fall ein Blick in das Pinout notwendig, um einen Bezug zwischen dem Aufdruck auf dem Controllino-Gehäuse und dem Quellcode herzustellen.



Das Programm kann mit einem Klick auf den kleinen Haken-Button 10 am unteren linken Bildschirmrand kompiliert werden. Wenn das Programm fehlerfrei kompiliert wurde, kann es durch den nebenstehenden Pfeil-Button 11 auf den Controllino geladen werden.

Fazit

Wer Erfahrung mit der Arduino-Programmierung hat, dem gelingt der Einstieg in die Controllino-Programmierung spielend. Visual Studio Code ist ein sehr nützliches Tool für schnelles und sauberes Coden und punktet gegenüber der Arduino IDE vor allem bei umfangreicheren Projekten. PlatformIO erleichtert die Mikrocontroller-Programmierung – nicht nur für die Arduino- bzw. Controllino-Plattform.

Der Controllino bietet eine gute Möglichkeit für alle, die für ihr Projekt eine günstige SPS suchen. Mit *logi.CAD 3* steht auch SPS-Programmierern, die sich in den klassischen SPS-Sprachen heimisch fühlen, ein Tool für die Programmierung nach IEC-Norm zur Verfügung. —caw

Für Maker!

Zubehör und Gadgets

shop.heise.de/gadgets



Waveshare Game HAT für Raspberry Pi

Ein Muss für jeden Retro Gamer! Verwandeln Sie Ihren Raspberry Pi in kürzester Zeit in eine Handheld-Konsole. Mit Onboard-Speakern, 60 Frames/s, Auflösung von 480x320 und kompatibel mit allen gängigen Raspberrys.

shop.heise.de/game-hat

41,90 € >

BEST-SELLER



ODROID-GO

Mit diesem Bausatz emulieren Sie nicht nur Spiele-Klassiker, sondern programmieren auch in der Arduino-Entwicklungsumgebung.

shop.heise.de/odroid

49,90 € >



NVIDIA Jetson nano

Das Kraftpaket bietet mit 4 A57-Kernen und einem Grafikprozessor mit 128 Kernen ideale Voraussetzungen für die Programmierung neuronaler Netze, die ähnlich wie Gehirnzellen arbeiten. **Inklusive Netzteil!**

shop.heise.de/jetson

134,90 € >



Raspberry Pi-Kameras

Aufsteckbare Kameras, optimiert für verschiedene Raspberry Pi-Modelle mit 5 Megapixel und verschiedenen Aufsätzen wie z. B. Weitwinkel für scharfe Bilder und Videoaufnahmen.

shop.heise.de/raspi-kameras

ab 18,50 € >



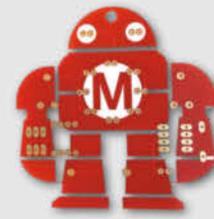
NEUER PREIS!

ArduiTouch-Set

Setzen Sie den ESP8266 oder ESP32 jetzt ganz einfach im Bereich der Hausautomation, Metering, Überwachung, Steuerung und anderen typischen IoT-Applikationen ein!

shop.heise.de/arduitouch

~~69,90 €~~
36,90 €



Makey Lötbausatz

Hingucker und idealer Löt-Einstieg: das Maskottchen der Maker Faire kommt als konturgräste Platine mitsamt Leuchtdioden, die den Eindruck eines pulsierenden Herzens erwecken.

Jetzt neu mit Schalter!

shop.heise.de/makey-bausatz

ab 4,90 € >



NEUER PREIS!

Komplettset Argon ONE Case mit Raspberry Pi 4

Das Argon One Case ist eines der ergonomischsten und ästhetischsten Gehäuse aus Aluminiumlegierung für den Raspberry Pi. Es lässt den Pi nicht nur cool aussehen, sondern kühlt auch perfekt und ist leicht zu montieren. Praktisch: alle Kabel werden auf der Rückseite gebündelt ausgeführt – kein Kabelsalat!

shop.heise.de/argon-set

~~117,60 €~~

99,90 € >



Stockschirm protec'ted

Innen ist Außen und umgekehrt. Dieser etwas andere Regenschirm sorgt für interessierte Blicke auch bei grauem und nassem Wetter. Als Highlight kommt noch das stilvolle und dezente Design in Schwarz und Blau mit der mehr als passenden Aufschrift "Always protec'ted" daher.

shop.heise.de/ct-schirm

22,90 € >



c't Tassen

c't-Leser und -Fans trinken nicht einfach nur Kaffee, sie setzen Statements. Und zwar mit drei hochwertigen Blickfängern, individuell design für Ihr Lieblings-Heißgetränk: „Kein Backup, kein Mitleid“, „Deine Mudda programmiert in Basic“ oder „Admin wider Willen“. Perfekt für Büro und Frühstückstisch!

shop.heise.de/ct-tassen

ab 12,90 € >

NEU



„No Signal“ Smartphone-Hülle

Passend für Smartphones aller Größen bis 23cm Länge blockt diese zusammenrollbare Hülle alle Signale von GPS, WLAN, 3G, LTE, 5G und Bluetooth, sowie jegliche Handy-Strahlung. Versilbertes Gewebe im Inneren der Tasche aus recycelter Fallschirmseide bildet nach dem Schließen einen faradayschen Käfig und blockiert so alles Signale.

shop.heise.de/no-signal-sleeve

29,90 € >

> Bestellen Sie ganz einfach online unter shop.heise.de oder per E-Mail: service@shop.heise.de

> Generell portofreie Lieferung für Heise Medien- oder Maker Media Zeitschriften-Abonnenten oder ab einem Einkaufswert von 20 €.
> Nur solange der Vorrat reicht. Preisänderungen vorbehalten.

© Copyright by Maker Media GmbH.

heise Shop

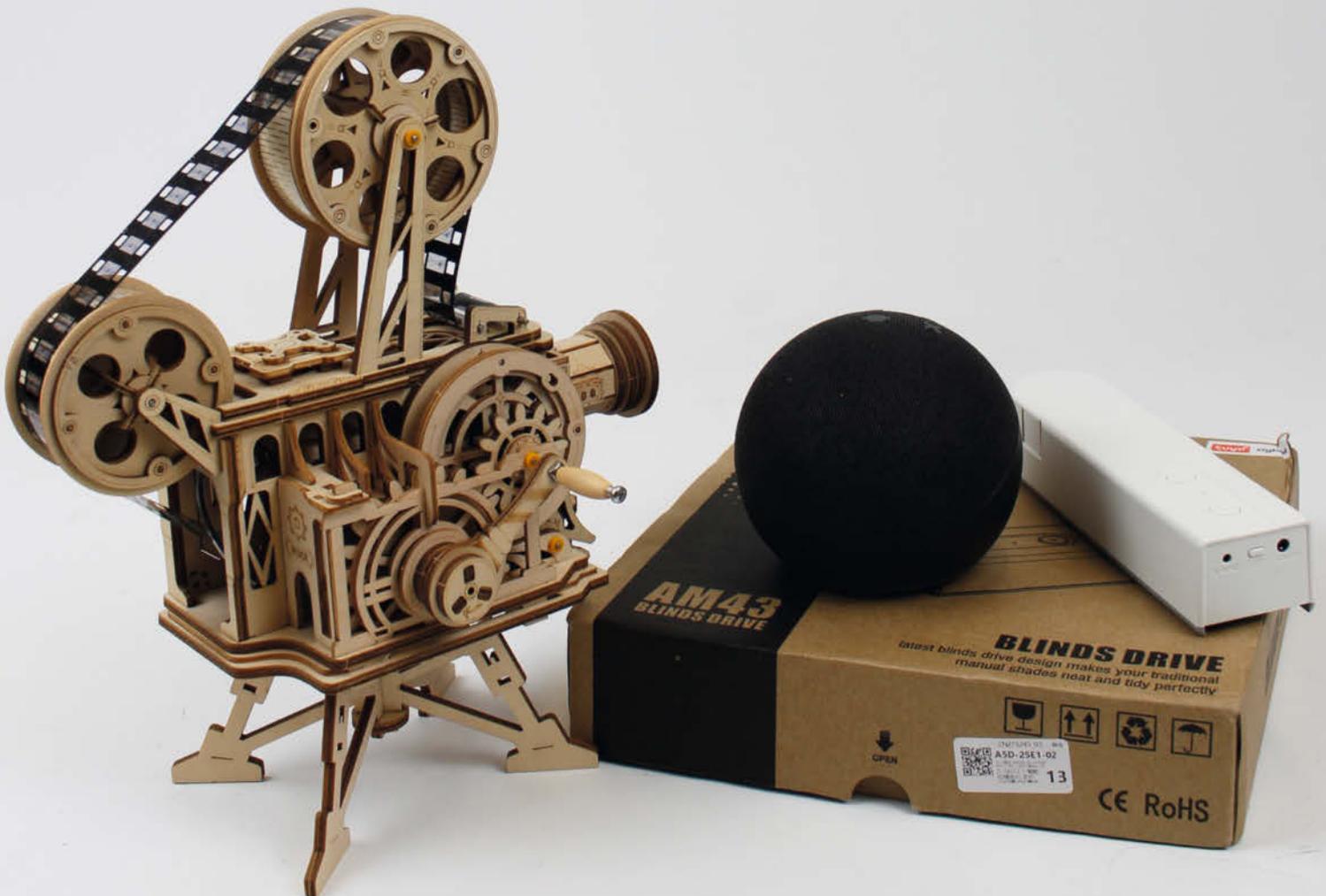
shop.heise.de >



TV und Heimkino mit Home Assistant, Teil 2

Im zweiten Heimkino-Teil bringen wir mit Rollo-Motoren Bewegung in die Verdunkelung des Homecinemas. Außerdem lernt Ihr Home Assistant, aufs Wort zu gehorchen per Alexa-Sprachassistent.

von Heinz Behling



Nach der ungewollten, aber notwendigen Pause der Smarthome-Serie in der vorigen Ausgabe, bedingt durch ein Update (siehe *Updates* auf Seite 100), das meine Rolllosteuerung lahmlegte, erfahren Sie nun, wie Sie Ihr Heimkino verdunkeln können. Das funktioniert dann sogar per Sprachbefehl, weil ich Ihnen auch zeige, wie Sie Amazons Sprachassistentin Alexa **1** mit ins Smarthome holen. Ein paar nützliche Home-Assistant-Automatiken legen wir dann auch noch an, die Ihr gesamtes Heimkino-Equipment mit nur einem Befehl starten oder nach dem Filmgenuss auch wieder beenden und die Verdunkelung aufheben.

Rollo-Antrieb

Bei der Wahl des Antriebs für die Verdunkelungsrollos spielten diese Überlegungen eine Rolle: Die Motoren sollten preiswert sein, denn oft wird mehr als ein Antrieb benötigt, beispielsweise bei mehreren Fenstern im Raum. Die Lösung sollte mit Rücksicht auf Mietwohnungen spurlos rückbaubar sein, damit bei einem Auszug keine teuren Reparaturen beschädigter Fenster fällig werden. Außerdem sollte sich die Verdunkelung auch bei Stromausfall wieder öffnen lassen, denn wenn keine Lampe mehr im Raum funktioniert, ist jedes Licht von außen willkommen.

Das führte dazu, dass eine Rollladen-Steuerung schnell vom Tisch war: Die hierzu benötigten fernsteuerbaren elektrischen Antriebe fallen schnell mit deutlich über 100 Euro pro Fenster ins Gewicht. Die meisten arbeiten außerdem mit Netzstrom, sodass sie bei Stromausfall nicht funktionieren. Da außerdem Bekannte von mir sich mit solch einem Rollladenantrieb, der auch an der Terrassentür installiert war, einmal komplett aus ihrem Eigenheim ausgesperrt hatten, kam es zu der Entscheidung für innen anzubringende, preiswerte Rollos mit Kettenantrieb. Achten Sie beim Kauf darauf, dass es wirklich Verdunkelungsrollos sind. Diese sind komplett lichtundurchlässig.



1 Dies sind Alexa-Echo-Lautsprecher. Es gibt aber Alexa auch mit Bildschirm.

Kurzinfo

- » Rollo-Motoren mit Home Assistant steuern
- » Automatik-Skript für den Heimkino-Start
- » Sprachassistent Alexa in Home Assistant integrieren
- » Per Home Assistant Cloud das Smarthome von überall erreichen

Checkliste

- Zeitaufwand:**
2 bis 3 Stunden
- Kosten:**
ab 80 Euro

Material

- » Rollo-Motor AM43 ZigBee-Version
- » Smarter Lautsprecher Alexa Echo alle Versionen möglich

Werkzeug

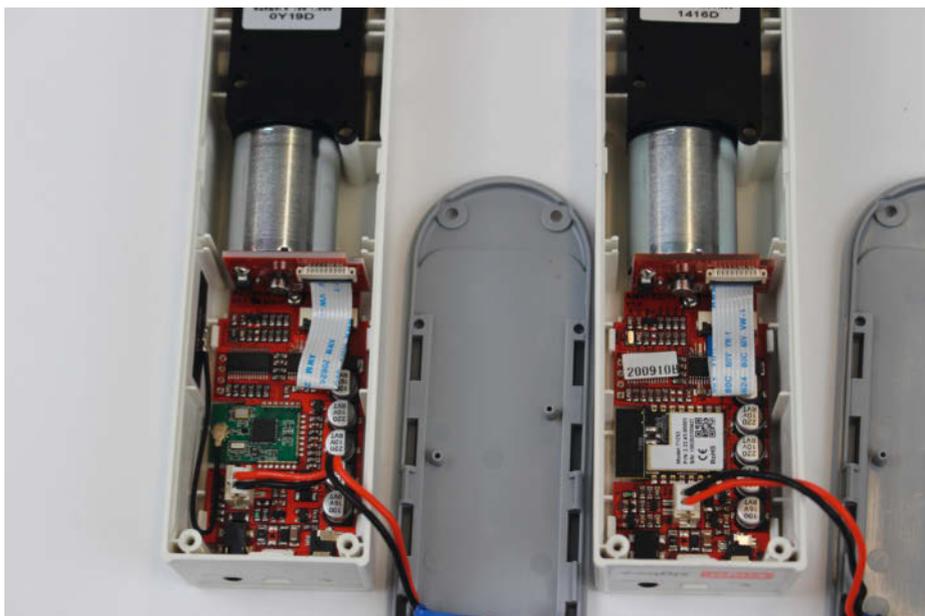
- » evtl. Bohrmaschine und Schraubendreher zur Rollo-Montage

Mehr zum Thema

- » Heinz Behling, Intelligentes Heim mit Home Assistant, Make 1/21, S. 100
- » Heinz Behling, Heizung unter Kontrolle, Make 2/21, S. 22
- » Heinz Behling, TV und Heimkino mit Home Assistant Teil 1, Make 3/21, S. 90

Alles zum Artikel im Web unter make-magazin.de/xhy9

The screenshot shows the Home Assistant web interface. At the top, there's a navigation bar with 'Getting started', 'Documentation', 'Integrations', 'Examples', 'Blog', and a help icon. Below this is a search bar for integrations. A sidebar on the left lists categories like 'All (1853)', 'Featured', 'Added in:', 'Alarm (37)', 'Automation (23)', 'Binary Sensor (155)', 'Calendar (7)', 'Camera (41)', 'Car (15)', 'Climate (79)', 'Cover (58)', 'DIY (54)', 'Device Automation (1)', 'Doorbell (5)', 'Downloading (11)', 'Energy (47)', 'Environment (17)', 'Fan (28)', 'Finance (14)', 'Front End (4)', 'Geolocation (8)', 'Health (26)', 'History (18)', 'Hub (104)', 'Humidifier (5)', and 'Image Processing'. The main area displays a grid of integration cards for various brands and protocols, including Abode, Beckhoff TwinCAT ADS, Aladdin Connect, BleBox devices, Bond, Brunt Blind Engine, Command Line Cover, Cover, Phoscon deCONZ, devolo Home Control, Fibaro, Freedompro, Garadget, Gogogate 2, HomeKit Controller, HomeMatic, Homematic IP Cloud, Hunter Douglas PowerView, IKEA TRÅDFRI, Insteon, KNX, LCN, and Lutron.



3 Links ist die Bluetooth-, rechts die ZigBee-Version des Rollmotors.



4 Die ZigBee-Integration erhält immer den Namen des verwendeten ZigBee-Adapters.



5 Der Set-Taster ist an der Motor-Unterseite.

Wer dennoch lieber seine Rollläden verwenden möchte: Für viele Rollläden-Motoren gibt es passende Integrationen in Home Assistant (Stichwort *cover*). Auf der Seite

<https://www.home-assistant.io/integrations/#cover>

können Sie einsehen, ob etwas Passendes für Ihre per WLAN fernsteuerbaren Motoren existiert 2. Dann muss lediglich die zutreffende Integration installiert und entsprechend der dazugehörigen Dokumentation auf Ihre Rollläden-Motoren konfiguriert werden.

Bei Rollos gibt es zwei verschiedene Arten: Solche mit eingebautem Antrieb (in der Regel Netzstrom-betrieben) und die Erweiterung herkömmlicher Rollos mit externen Motoren (meist mit Akku-Stromversorgung, teils auch mit Fotozellen ergänzt). Auf solche externen Antriebe fiel schließlich meine Wahl, denn diese Motoren erfüllen alle Anforderung: Netzstrom-unabhängig, preiswert (ab ca. 35Euro/Fenster) und leicht rückbaubar (siehe Abschnitt *Rollo-Montage*). Daneben gibt es zum Beispiel auf Thingiverse noch zahlreiche Selbstbauprojekte zum Rollo-Antrieb. Allerdings verwenden die meisten davon leistungsschwache Motoren. Ausführungen mit stärkeren Aggregaten hingegen erreichen schnell Bauteilkosten, die den Selbstbau sinnlos machen. So fiel dann die Wahl auf Motoren vom Typ AM43, die von verschiedenen Herstellern in Fernost angeboten werden. Wenn Sie Zeit haben (Lieferzeit nach eigenen Erfahrungen bis zu drei Monaten) bestellen Sie sie direkt in China. Händler aus dem europäischen Raum verlangen zum Teil mehr als das doppelte und mogeln gerne mal bei der Formulierung der Lieferzeit: *Lieferung*

bis zum 1.5. bedeutet etwas anderes als *Versand* bis zum 1.5.!

Die gewählten Motoren gibt es dann noch mit unterschiedlichen Netzwerkverbindungen: Meist werden Bluetooth-BLE-fähige Versionen angeboten. Die hatte ich zuerst auch verwendet, bekam dann aber Probleme nach einem Update des MQTT-Brokers, die sich aber durch ein erneutes Update einige Wochen später wieder lösten. Inzwischen probierte ich auch AM43-Motoren mit ZigBee aus. Schließlich habe ich ja bereits eine ganze Menge an solchen Geräten in Betrieb genommen, unter anderem diverse Lampen und Schaltsteckdosen 3. Da diese Motoren problemlos funktionierten, habe ich mich schließlich für ZigBee entschieden. Nach meiner Erfahrung müssen die auch nicht so oft nachgeladen werden wie ihre Bluetooth-Kollegen. Offenbar ist ZigBee stromsparender.

Rollo-Karte in Home Assistant

Als Erstes sollten Sie den Akku des neuen Rollo-Motors aufladen. Etwa sechs Stunden braucht das mitgelieferte Ladegerät dafür. Sobald die LED am Lader von rot auf grün wechselt, ist dieser Job erledigt.

Danach folgt das Pairing mit dem ja bereits in den vorigen Folgen dieser Smarthome-Reihe ZigBee-fähig gemachten Home Assistant-Server. Gehen Sie im Browser Ihres Computers auf die Web-Oberfläche des Servers mit der Adresse:

homeassistant:8123

Nach Klicks auf *Einstellungen* und *Integrationen* klicken Sie im Feld der ZigBee-Integration auf *Konfigurieren* 4.

Mit einem Klick auf *Gerät hinzufügen* starten Sie die Suche nach neuen Zigbee-Komponenten.

Nun müssen Sie den Rollo-Motor in den Pairing-Modus schalten. Dazu ist er zunächst zu starten, indem Sie den Set-Schalter an seiner Unterseite etwa eine Sekunde lang drücken. Anschließend blinkt die LED des Motors zwei Sekunden lang. Dann drücken Sie den Set-Taster zusammen mit der Abwärts-Taste für drei Sekunden. Erneut blinkt die LED 5.

Kurze Zeit später zeigt Ihnen Home Assistant an, dass ein neues Gerät gefunden wurde 6.

Machen Sie sich die weitere Arbeit einfacher, indem Sie den Gerätenamen und den Bereich auf etwas leicht zu merkendes ändern, zum Beispiel *Rollo Grosses Fenster* und *Wohnzimmer* 7.

Danach können Sie in der Home-Assistant-Weboberfläche zur Übersichtsseite zurückkehren. Jetzt brauchen wir noch einen Eintrag im Bereich *Wohnzimmer*, mit dem wir das Rollo auch steuern, also öffnen und

schließen können. Wechseln Sie also auf den Reiter fürs *Wohnzimmer*, klicken oben rechts auf die *drei Punkte* und wählen *Benutzeroberfläche konfigurieren*. Mit + *Karte hinzufügen* gelangt man ins Auswahlfeld für die neue Karte (das sind die einzelnen Kontroll- und Betätigungsfelder in der Weboberfläche). Für Rollos (oder auch Rollläden) gibt es keine spezielle Karte, daher ist die schlichte *Entität* das richtige **8**.

Danach klicken Sie auf die Zeile *Entität* und geben dort den zuvor geänderten Gerätenamen *Rollo Grosses Fenster* ein. Daraufhin erscheint ein Auswahlfeld mit dem genauen Namen des Motors. Klicken Sie dort hinein **9**.

Nun können Sie den Bearbeitungsmodus wieder mit einem Klick auf das große X oben links beenden. Im Home-Assistant-Wohnzimmer sehen Sie nun die neue Rollo-Karte **10**.

Falls Sie nun die Schalter suchen, mit dem Sie den Motor in Bewegung setzen können, klicken Sie einmal auf die neue Karte. Daraufhin erscheint ein Fenster, in dem Sie per Klicks auf einen der nach oben und unten gerichteten Pfeile den Motor zur entsprechenden Bewegung veranlassen können **11**.

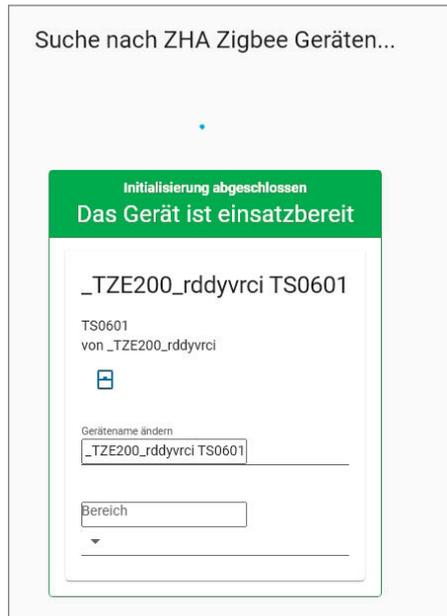
Falls Sie weitere Motoren verwenden möchten, wiederholen Sie dieses Kapitel für jeden davon, verwenden aber jeweils unterschiedliche Gerätenamen.

Motor montieren

Ich gehe hier davon aus, dass Sie das beziehungsweise die Rollo(s) bereits an den Fenstern montiert haben. Wie das geht, lesen Sie bitte in der zum Rollo gehörenden Montageanleitung.

Dem Motor sind unterschiedliche Zahnräder beigelegt **12**, die zu den verschiedenen bei Rollos benutzten Ketten gehören. Suchen Sie das passende aus und stecken Sie es nach Entfernen der weißen Schutzkappe auf die Achse des Motors. Führen Sie die Kette des Rollos um das Zahnrad. Der Motor sollte nun an der tiefsten Stelle der Kette hängen. Bevor Sie die Schutzkappe wieder aufsetzen, müssen Sie zwei Kunststoffflaschen herausbrechen. Damit legen Sie die Öffnungen frei, durch die die Kette führt.

Zum Motor gehört eine Halterung, die auf zweierlei Art an der Wand befestigt werden kann: Kleben oder Schrauben. Das Kleben mithilfe des mitgelieferten beidseitigen Klebebands funktioniert nur auf völlig glatten Flächen (zum Beispiel glänzende Kacheln). Auf allen anderen Wandoberflächen sollte der Motor angeschraubt werden. Bei der Befestigung ist zu beachten, dass der Motor in der Halterung eingerastet ist und die Kette leicht(!) gespannt ist. Besonders beim Ankleben ist es sehr empfehlenswert, die Position zunächst anzuzeichnen, bevor man das Klebeband anbringt. Dieses Band klebt auf glatten, sauberen



6 Der Motor wird unter einer recht kryptischen Bezeichnung gefunden.

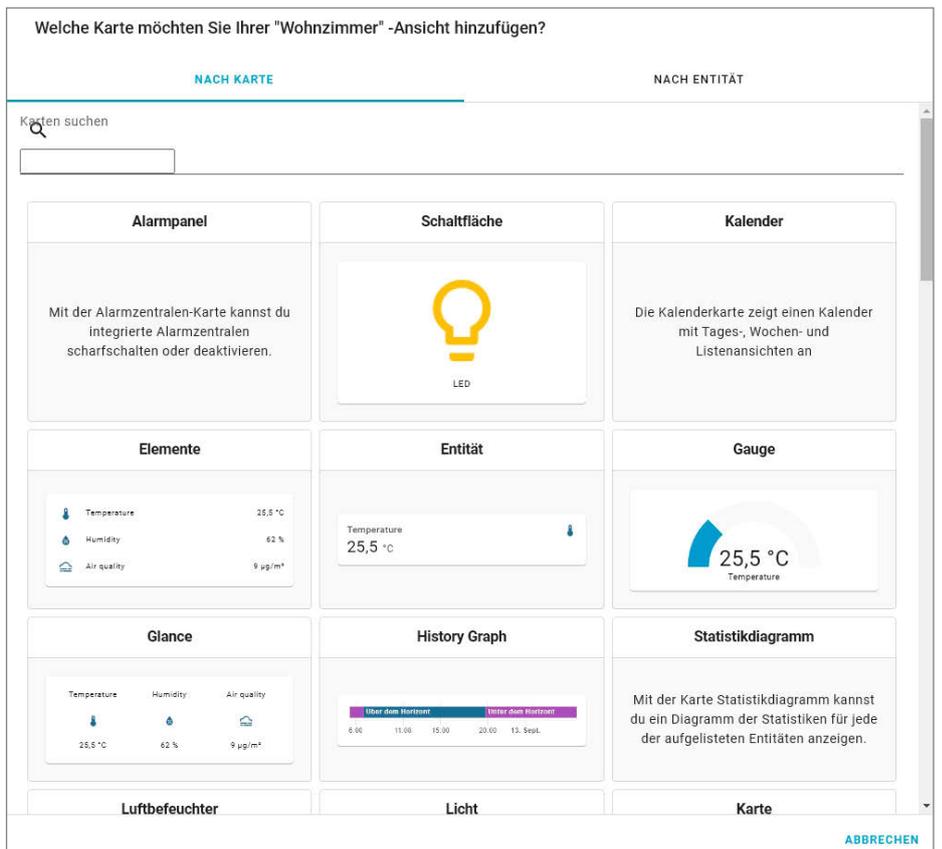


7 Diesen Gerätenamen kann man sich bestimmt erheblich besser merken.

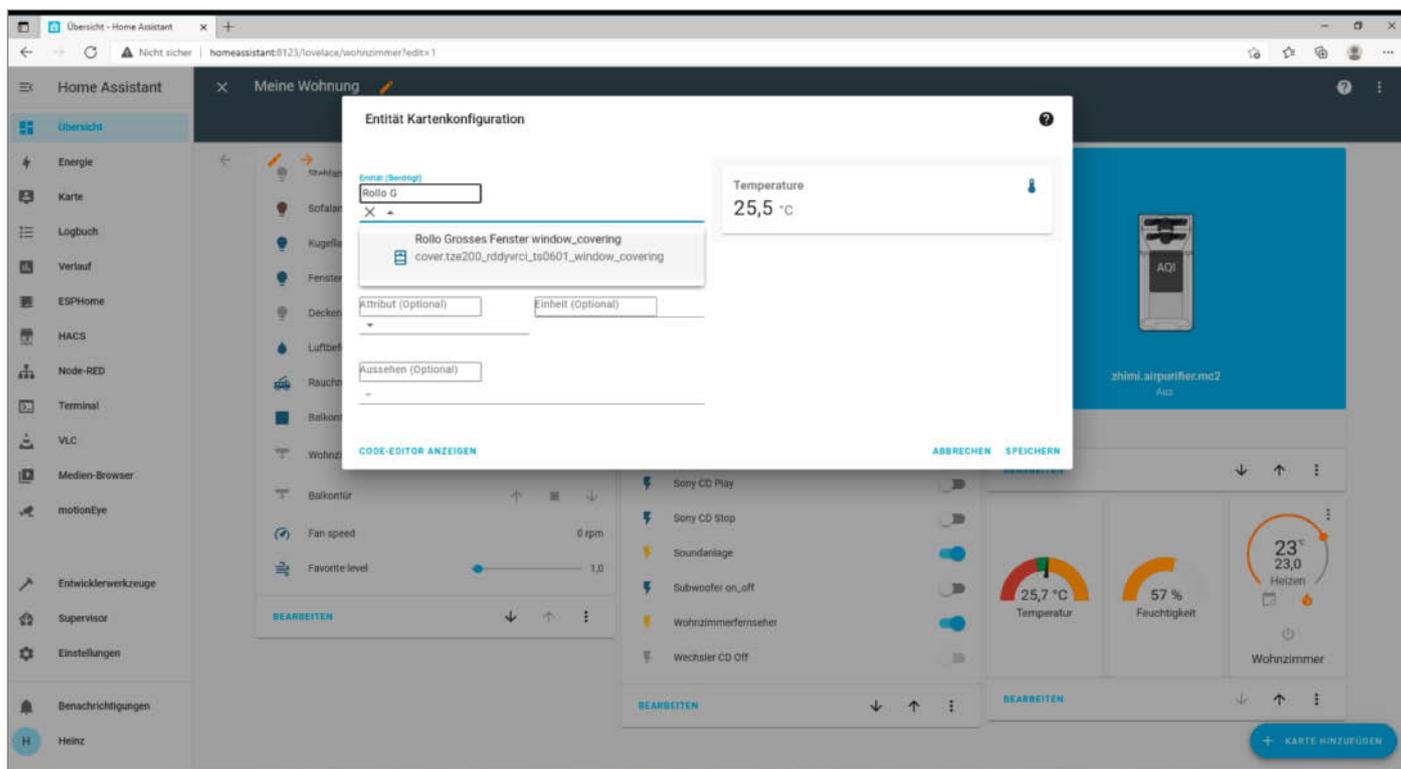
Oberflächen nämlich wirklich bombenfest und kann nachträglich nicht mehr korrigiert werden **13**.

Wichtig: Bevor Sie das Rollo zum ersten Mal per Home Assistant betätigen, müssen Sie die

obere und untere Endposition des Rollos am Motor einstellen. Dazu die Set-Taste drei Sekunden drücken. Die LED blinkt rot. Nun wählen Sie mit der Aufwärts- beziehungsweise Abwärtstaste, ob Sie die obere oder untere



8 Das Rollo ist für Home Assistant eine schlichte Entität.

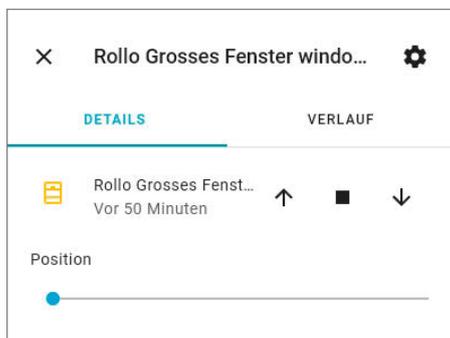


9 Erst mit einem Klick auf den angezeigten Rahmen wird die richtige Entitätsbezeichnung übernommen.

Endposition festlegen möchten. Dann fahren Sie mit den Tasten das Rollo in die entsprechende Position. Ist die erreicht, drücken Sie die Set-Taste erneut für drei Sekunden. Wiederholen Sie das mit der zweiten Endposition. Jetzt folgt der spannende Moment: Probieren Sie auf der Weboberfläche von Home



10 Ganz schlicht: Die neue Karte zeigt nur den aktuellen Zustand des Rollos an.



11 Ein Klick auf einen Pfeil fährt den Motor ganz in die entsprechende Richtung. Wenn Sie den blauen Punkt verschieben, können Sie auch Zwischenpositionen einstellen.

Assistant die Rollobetätigung aus. Das Rollo sollte leicht surrend seine Positionen erreichen ohne zu haken (Kette klemmt?) oder zu quietschen (Kette zu stark gespannt). Bei den Endpositionen stoppt die Bewegung automatisch.

Bei Stromausfall fällt natürlich die Steuerung per Home Assistant ebenfalls aus. Mit den beiden Tasten am Motor selbst lässt sich das Rollo aber auch dann betätigen, da der Motor seinen Strom ja aus dem eingebauten Akku bezieht. Das setzt natürlich voraus, dass Sie das Nachladen nicht vergessen. Es sollte nach Herstellerempfehlung etwa alle sechs Monate erfolgen. Sie können die Zeitspanne verlängern, wenn das Fenster einige Stunden täglich direkte Sonne erhält. Dem Motor ist nämlich eine kleine Solarzelle beigelegt, die mit Saugnäpfen an der Fensterscheibe befestigt und mit dem Motor per steckbarem Kabel verbunden werden kann 14. Sie lädt dann den Akku etwas nach. Ich habe aber die Erfahrung gemacht, dass das Kabel die Rollobewegung behindern kann und der Nachladeeffekt recht gering ist. Daher benutze ich das Solarpanel nicht.

Wer es ganz besonders komfortabel mag (oder schlicht vergesslich ist), kann das Nachladen auch automatisieren: Das Ladegerät wird einfach permanent mit dem Motor verbunden und über eine ZigBee-Schaltsteckdose jedes halbe Jahr einmal von Home Assistant gesteuert für einige Stunden eingeschaltet. Man kann auch das Ladegerät direkt in eine normale Steckdose stecken. Der Stromver-

brauch dürfte in beiden Fällen ähnlich sein, denn auch eine Schaltsteckdose verbraucht ständig Energie.

Alexa installieren

Nun geht es darum, dem Home Assistant das Hören beizubringen. Ich habe mich hier für die Amazon-Sprachassistentin Alexa entschieden. Mit dem entsprechenden Google-Assistenten geht es auch. Da ich aber auch Amazon Prime benutze, habe ich mich für Alexa entschieden. Achtung: Mit Alexa verlassen Sie die räumliche Abgeschlossenheit Ihres Heimes, denn die Spracherkennung erfolgt via Internet auf einem Amazon-Server. Dabei kann Amazon auch Audio-Aufnahmen anfertigen, die ausgewertet werden können (zur Verbesserung der Erkennungsleistung). Alexa sollte daher nicht gerade in Räumen verwendet werden, in dem vertrauliche Besprechungen stattfinden.

Da die Spracherkennung im Internet erfolgt, muss Ihr Home Assistant auch von außen, also vom Internet erreichbar sein. Und zwar unter einer Adresse, die sich nicht ständig durch neues Verbinden seitens Ihres Internetproviders ändert. Das erreicht man durch einen kostenpflichtigen Zugang zur Home Assistant-Cloud bei Nabu Casa (das ist der Home Assistant-Gründer). Das kostet 5 US-Dollar pro Monat.

Die Software ist bereits in Home Assistant enthalten. Sie müssen lediglich einen Ac-



12 Das Zahnrad muss zur Rollokette passen.



13 Zur Halterung wird ein doppelseitiges Klebeband und ein Dübel-/Schraubenset mitgeliefert.

count anlegen. Mit Klicks auf *Einstellungen*, *Home Assistant Cloud* und *Konto verwalten* gelangen Sie zu Nabu Casa. Unter *Sign Up* können Sie nun den Account anlegen 15. Die Abrechnung erfolgt per Paypal oder Kredit-

karte. Es gibt allerdings einen kostenlosen Testmonat. Vor Ablauf dieses Monats müssen Sie keinerlei Daten Ihrer Kreditkarte oder Paypal-Accounts angeben. Sie laufen also keine Gefahr, bei Ablauf des Testmonats auto-

matisch in einen kostenpflichtigen Dienst zu laufen.

Legen Sie den Account also an. Falls noch nicht geschehen, installieren Sie jetzt als Nächstes Alexa. Dazu ist ein Smartphone er-

TECHNIKUNTERRICHT MACHT ENDLICH SPÄß!



Make:Education

Mit **Make Education** erhalten Sie jeden Monat kostenlose Bauberichte und Schritt-für-Schritt-Anleitungen für einen praxisorientierten Unterricht:



Für alle weiterführenden Schulen



Fächerübergreifend



Digital zum Downloaden



Monatlicher Newsletter

Jetzt kostenlos downloaden:

make-magazin.de/education



14 Das Kabel des Solarpanels stört meist den Rollo-Lauf.

Alexa

Mit der Alexa-Integration für Home Assistant Cloud können Sie alle Ihre Home Assistant-Geräte über jedes Alexa-fähige Gerät steuern.

- [Aktiviere den Home Assistant-Skill für Alexa](#)
- [Konfigurationsdokumentation](#)

Statusberichterstattung aktivieren

Wenn die Statusberichterstattung aktiviert wird, sendet Home Assistant alle Statusänderungen exponierter Entitäten an Amazon. So wird in der Alexa-App immer der neueste Status angezeigt.

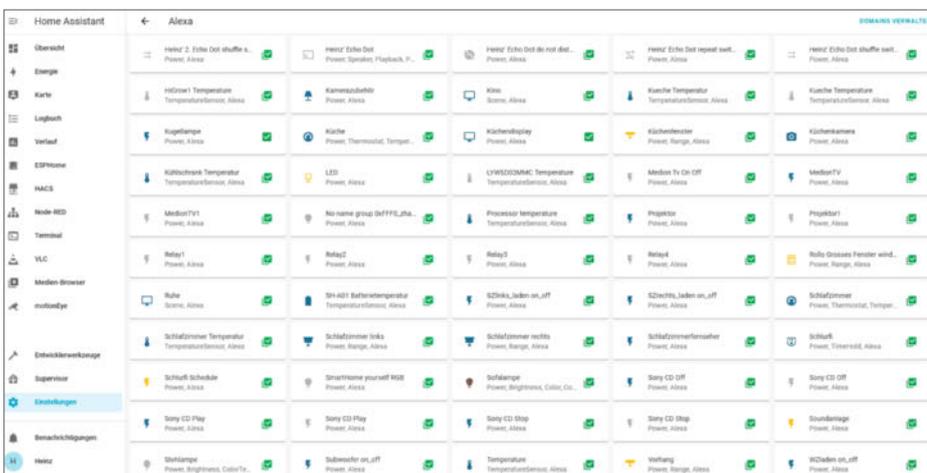
ENTITÄTEN ZU AMAZON SYNCHRONISIEREN
ENTITÄTEN VERWALTEN

16 Mit dem kleinen Schalter neben Alexa aktivieren Sie die Sprachassistentin in Home Assistant.

forderlich. Die Anleitung dazu erhalten Sie von Amazon automatisch, wenn Sie dort einen Alexa-fähigen Lautsprecher (etwa einen Echo) oder Medienplayer gekauft haben. Auf dem Smartphone muss die Alexa-App installiert werden. Die führt Sie dann automatisch Schritt

für Schritt durch die Installation. In der App müssen Sie anschließend unter *Skills* noch *Home Assistant Smart home* aktivieren.

Jetzt muss Ihrem Home Assistant noch beigebracht werden, dass er mit Alexa zusammenarbeiten soll. Auch das ist schnell erledigt,



17 Das alles kann Alexa steuern.

Home Assistant Cloud

powered by NABU C.A.S.A.

Sign in with your email and password

Email

Password

[Forgot your password?](#)

Sign in

Need an account? [Sign up](#)

15 Das Anmeldefenster der Home Assistant Cloud

denn es genügt unter Einstellungen und Alexa ein Klick auf den Schalter 16.

Wundern Sie sich nicht, falls Alexa jetzt bereits losplappert. Home Assistant offenbart ihr nämlich, welche Geräte in Ihrem Smarthome vorhanden sind. Alexa meldet das brav, indem es so etwas sagt wie: „Ich habe das Gerät so-und-so gefunden und kann damit folgendes machen“

Wenn Alexa sich wieder beruhigt (oder gar nicht erst anfängt zu sprechen), klicken Sie auf *Entitäten verwalten*. Daraufhin erscheint eine Liste, die je nach Ausbaustufe Ihres Smarthomes mehr oder weniger umfangreich ist 17.

Darin sollte unter *Exponierte Entitäten* auch das *Rollo Grosses Fenster* auftauchen. Falls nicht, steht es unter *Nicht angezeigte Entitäten*. In diesem Fall klicken Sie darauf und wählen *Entität anzeigen*.

Jetzt hört Alexa aufs Wort: Sprechen Sie „Rollo Grosses Fenster schließen“ und das Rollo fährt nach unten. „Rollo Grosses Fenster öffnen“ bringt es wieder nach oben. Die Spracherkennung ist übrigens unabhängig davon, wer spricht. Lustig wird's, wenn im Fernsehen eine Alexa-Werbung läuft und dort jemand sagt „Alexa, mach das und das ...“

Kino-Automatik

Jetzt kommen wir zu unserem eigentlichen Ziel: Auf einen Sprachbefehl hin soll das gesamte Heimkino-Equipment starten, also der Fernseher/Beamer und die Soundanlage eingeschaltet und der Raum verdunkelt werden. Dazu brauchen wir ein Skript, das dies erledigt, und dann durch Alexa gestartet wird. Um es anzulegen, klicken Sie auf *Einstellungen* und *Skripte* sowie *+ Neues Skript erstellen*.

Zunächst geben wir dem Skript einen Namen, zum Beispiel *Heimkino*. Sie können auch ein Symbol verwenden, etwa ein Moni-

tor-Zeichen. Dann setzen Sie in das Symbolfeld *mdi:monitor* ein.

Unter *Sequenz* geben wir nun als Gerät *Dienst ausführen* an. Dann wählen Sie die den *Dienst Schalter: Turn on* und schließlich darunter die Entitäten, die eingeschaltet werden sollen, also *Soundanlage* und *Projektor* (wie Sie die ins Smarthome einbinden, haben Sie bereits in der *Make 3/21* erfahren) **18**.

Dann klicken Sie auf *Aktion hinzufügen* und wählen als Gerät *Rollo Grosses Fenster*. Die Position des Rollos setzen Sie auf 0.

Jetzt unbedingt auf *Skript speichern* klicken, sonst geht alles verloren.

Nun folgt noch das Skript zum Beenden der Vorführung. Es erhält den Namen *Filmende*. Als Gerät wählen wir wieder *Dienst ausführen* und als *Dienst Schalter: Turn off*. Damit können wir aber nur die *Soundanlage* ausschalten. Der *Projektor* (oder die meisten anderen Infrarot-gesteuerten Geräte) haben ja zum Ein- und Ausschalten denselben Befehl. Also brauchen wir eine weitere Aktion, dort wieder *Dienst ausführen*. Diesmal verwenden wir aber den *Dienst Schalter: Turn on* und als Entität den *Projektor*.

Nun muss das Skript noch das Rollo hochfahren. Also noch eine Aktion hinzufügen, als

Name

Symbol

Entitäts-ID

Der Modus steuert, was passiert, wenn ein Skript aufgerufen wird, während es noch von einem oder mehreren vorherigen Aufrufen ausgeführt wird. Lies [Skript-Dokumentation](#) für weitere Informationen.

Modus

Sequenz

Die Abfolge der Aktionen dieses Skripts. [Erfahre mehr über verfügbare Aktionen.](#)

Aktionstyp

Dienst

Turn a switch on

Ziele
 Was dieser Dienst als Bereiche, Geräte oder Entitäten verwenden soll.

Soundanlage
 Projektor

18 Das Heimkino-Skript

shop.heise.de/ho-beleuchtung

**Heft + PDF
mit 29% Rabatt**

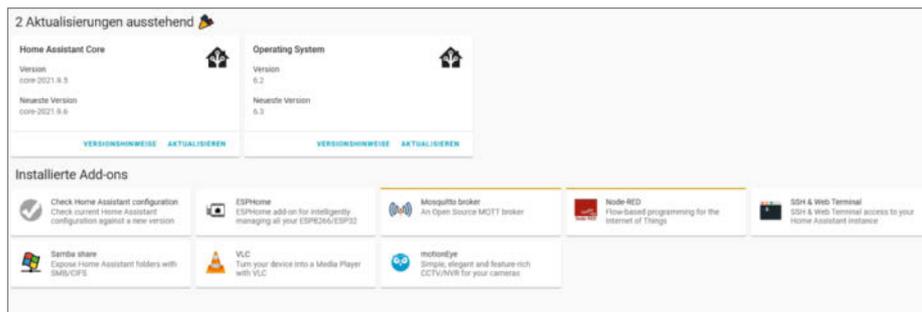
Das bringt Licht ins Dunkel!

Starten Sie mit dem Thema *Smarte Beleuchtung* in die **neue Heftreihe von heise online:**

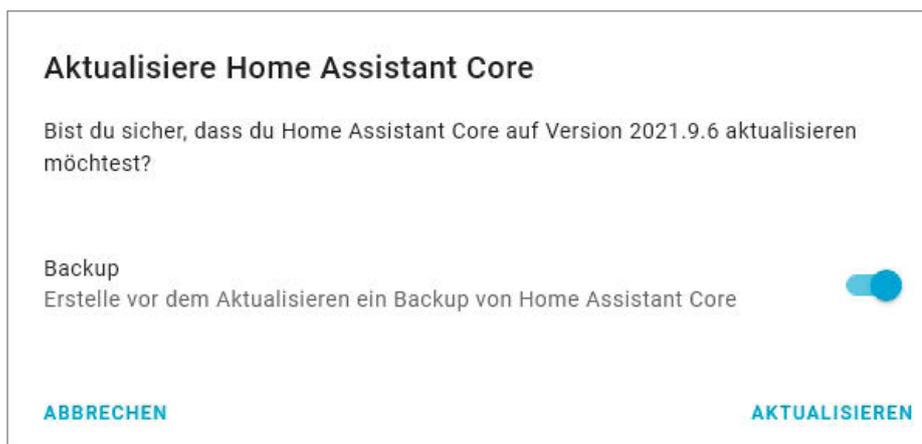
- ▶ Smartes Licht in Haus und Garten
- ▶ Smarte LED-Lampen für E14 und E27 im Test
- ▶ Auch im Set mit Smart Steckdose zum Sonderpreis
- ▶ Für Abonnenten portofrei

Heft für 14,90 € • PDF für 12,99 € • Bundle Heft + PDF 19,90 €

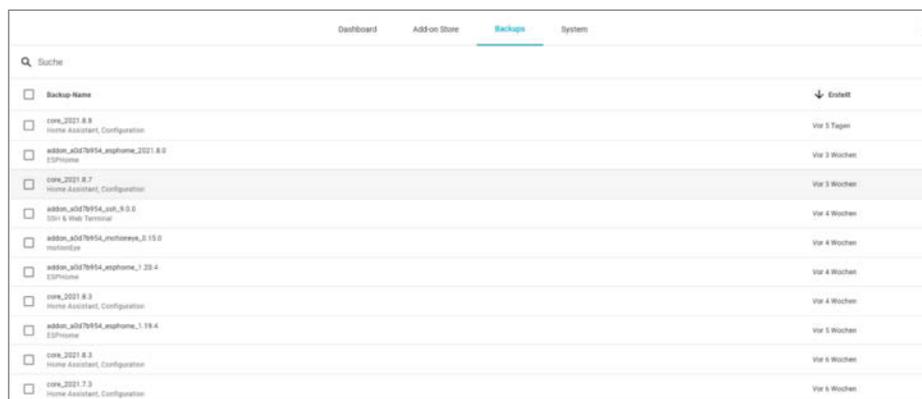
shop.heise.de/ho-beleuchtung



19 Oben werden Updates für Home Assistant und das Betriebssystem gemeldet. Darunter zeigen farbige Rahmen an, für welche Integrationen es Aktualisierungen gibt.



20 Wichtig: Der blaue Schalter muss auf ein (rechte Position) stehen!



21 Hier können Sie das rückgängig zu machende Update auswählen.

Gerät das *Rollo Grosses Fenster* wählen und die Position auf 100 setzen. Speichern nicht vergessen. Jetzt müssen wir Alexa noch dazu bringen, die Skripte zu erkennen. Dazu klicken Sie auf *Einstellungen, Home Assistant Cloud* und *Entitäten zu Amazon synchronisieren*.

Von nun an können Sie mit „Alexa, Heimkino einschalten“ und „Alexa, Filmende einschalten“ Ihr Homecinema starten und beenden.

Die Sprache ist noch etwas holprig. Das liegt daran, dass Alexa standardmäßig nur

die Befehle ein- und ausschalten bei Skripten versteht. In einer der nächsten Ausgaben werde ich Ihnen zeigen, wie man den Befehls-Wortschatz der Sprachassistentin erweitert, sodass dann auch elegantere Formulierungen möglich sind.

Hitzeschutz

Übrigens können Sie die Rollos auch noch zu anderen Zwecken benutzen: So liegt mein

etwa 10 Quadratmeter grosses Wohnzimmerfenster genau in Richtung Süden. Im Sommer hab ich so einen solarbetriebenen Backofen. Da mein Home Assistant aber ohnehin in jedem Raum die Temperatur misst, kann man mit einer Automatik bei Überschreiten einer Temperatur die Rollos herunterfahren und nach Unterschreiten des Temperaturgrenzwerts wieder öffnen lassen. In dieser Automatik verwendet man als Auslöser den *numerischen Zustand* der Entität *Wohnzimmerthermometer* (oder eines anderen sonnenbelasteten Zimmers). Den Temperaturwert setzt man bei der Schließautomatik ins Feld *Über*, bei der Öffnungsautomatik ins Feld *Unter* ein. Als Aktion dient wie beim Heimkino *Dienst ausführen* und *cover.set_cover_position*. Dann gibt man noch die jeweilige *Position* ein (0 = komplett geschlossen, 100 = komplett offen).

Wie geht es weiter?

Mit diesem Teil ist mein Smarthome-Projekt zunächst relativ fertig, obwohl, fertig wird es wahrscheinlich nie ganz. In den folgenden Make-Ausgaben werde ich daher immer wieder einmal das Thema aufgreifen und neue Ausbaustufen zeigen. Vielleicht haben Sie ja auch eigene Projekte zum Home Assistant verwirklicht oder Tipps zu seiner Anwendung. Schicken Sie mir eine Mail. Mit etwas Glück können Sie dann demnächst in Ihrer Make davon lesen.

Bis dahin viel Spaß bei der Filmvorführung im dunklen Kino.

Updates

Home Assistant und seine Erweiterungen werden ständig weiterentwickelt und dementsprechende Updates zur Verfügung gestellt. Schauen Sie ab und zu einmal unter *Supervisor* nach. Dort erfahren Sie, ob Ihr Home Assistant up to date ist 19.

Wenn Sie solch ein Update durchführen, achten Sie darauf, dass eine Sicherheitskopie angelegt wird. Danach werden Sie vor dem Update gefragt 20. So können Sie nämlich wieder zurück zum ursprünglichen funktionierenden System, falls ein Update irgend etwas so ändert, dass es in Ihrem Smarthome nicht mehr arbeitet.

Die Restore-Funktion finden Sie unter *Supervisor* und *Backups*. Dort erscheint eine Liste mit allen angelegten Backups 21.

Klicken Sie auf das Backup, dass Sie wiederherstellen möchten. Im nächsten Fenster klicken Sie dann auf das Kästchen vor der wiederherzustellenden Komponente und wählen dann *Restore*. Ein paar Minuten später ist dann Ihr Home Assistant wieder in der Vergangenheit. Aber ich hoffe, dass Sie diese Restore-Funktion nie brauchen werden. —hgb

MINT



JOBTAG 2021

Mathematik – Informatik – Naturwissenschaften und Technik

Heise Medien und das Innovationsmagazin **MIT Technology Review** begrüßen Sie beim **MINT-Jobtag** in Stuttgart und München. Zusammen mit unserem Partner, dem Online-Stellenmarkt-**Jobware**, bieten wir Ihnen die ideale Plattform für die Jobsuche. Zahlreiche attraktive Arbeitgeber präsentieren sich Ihnen mit ihren aktuellen Stellenangeboten und Aus- und Weiterbildungsplätzen.

TEILNEHMENDE STÄDTE

STUTTGART

05.10.2021 | 12-17 Uhr
Haus der Wirtschaft,
Willi-Bleicher-Straße 19

MÜNCHEN

19.10.2021 | 12-17 Uhr
IHK Campus, IHK für
München und Oberbayern
Orleansstraße 10-12

Eine Auswahl der teilnehmenden Unternehmen:



Kostenfrei registrieren unter

www.mint-jobtag.de

Lichtpistole „Super Marksman“

Kellerfund: Ein elektronisches Spielzeug aus den späten 1970er Jahren lehrt binäre Zahlen.

von Carsten Wartmann



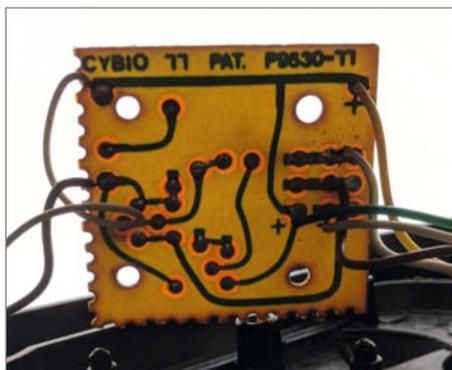
Der Kellerfund war etwas unheimlich, eine Pistole? In der dunklen Schublade sieht die Pistole doch recht echt aus. Beim ersten Anfassen wird aber schnell klar, dass es keine echte Waffe ist. Trotzdem viel zu realistisch, um meine Kinder damit spielen zu lassen. Meine Neugier war aber trotzdem geweckt: Da ist eine Linse im Lauf und ein Batteriefach im Griff. Der Abzugsmechanismus und der Hahn (*Double-Action*) war aber erst mal nicht mehr funktional. Ist das nur ein Lichteffekt oder ein elektronisches Schießspiel?

Der Make-Magazin-Twitch-Stream (siehe Link) kam da genau zur rechten Zeit und die Pistole wurde dort live aufgeschraubt, analysiert und repariert. Die Schaltung aus einem Transistor, einem Widerstand und einem Kondensator, zusammen mit dem vom Abzug betätigten Umschalter, war schnell entschlüsselt. Nur die Lampe im Lauf war defekt, mit einer hellen LED konnte aber schnell die grundsätzliche Funktion getestet werden.

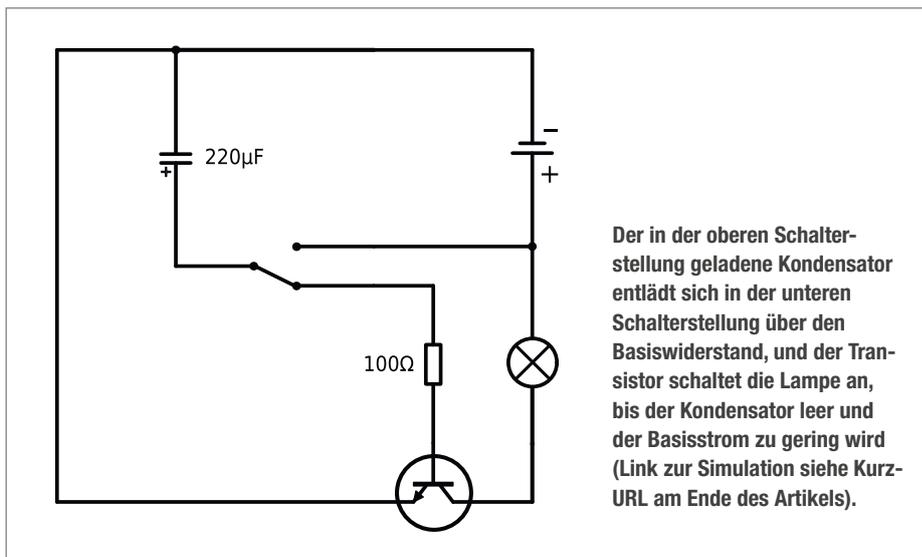
Nur war mit der LED jeder „Schuss“ ein sekundenlanges Leuchten. Um dem abzuweichen, habe ich einen Widerstand parallel zur LED angeschlossen, um den viel höheren Stromverbrauch der Lampe zu imitieren. Die LED wurde zusammen mit dem Widerstand in eine alte Lampenfassung eingebaut und so musste keine Modifikation an der Pistole vorgenommen werden, falls sie sich doch noch als rares Sammlerstück erweisen sollte.

Um mehr über dieses Gerät herauszufinden war es hilfreich, dass auf der Platine ein Firmenname (*Cybio*) aufgedruckt ist. So konnte ich im Internet die spärliche Information ausgraben, dass es sich um das 1970er-Jahre-Spiel *Super Marksman CY778* handelt. Auf der Platine gibt es einen Aufdruck, der eine Patentnummer sein soll, ich konnte aber kein Patent dazu finden. Dies scheint damals üblich gewesen zu sein, egal ob das Patent in Prüfung (Pat. pend.) oder nur als Drohkulisse gedacht war.

Das Ziel war eine orange Eule mit einer Zielscheibe auf dem Bauch. Sie zählte die Punkte und kreischte, wenn sie getroffen wurde. Arme Eule. Das Ziel haben wir leider nicht mehr in der Schublade.



Per Hand gezeichnete Platine



Der in der oberen Schalterstellung geladene Kondensator entlädt sich in der unteren Schalterstellung über den Basiswiderstand, und der Transistor schaltet die Lampe an, bis der Kondensator leer und der Basisstrom zu gering wird (Link zur Simulation siehe Kurz-URL am Ende des Artikels).

Je nachdem, wie gut getroffen wird, gibt es 1, 2, 4 oder 8 Punkte. Soll man hier binär rechnen lernen? Der Aufdruck $1 + 2 + 4 + 8 = ?$ scheint dies zu fordern, mithin können maximal 15 Punkte erzielt werden. Wie genau dies elektronisch gelöst war, bleibt mangels Eule leider ein Geheimnis. Im Internet fand ich nur die Angabe, dass man 60 Sekunden Zeit hatte für seinen Versuch. Wie viel Schüsse dabei abgegeben werden sollten, ist nicht bekannt.

Um die Spielerfahrung zu testen, könnte man doch einen Mikrocontroller mit einem lichtempfindlichen Widerstand nehmen? Das funktioniert erstaunlich gut und trotz meiner primitiven Programmierung kann man, per Integration des Lichtimpulses, einen guten von einem schlechten Treffer, unterscheiden. Aber so richtig Spaß macht das alles nicht. Es ist und bleibt ein Schießspielchen und die Pistole wird wohl wieder in einer Schublade verschwinden. —caw

Alles zum Artikel im Web unter make-magazin.de/xysh

Das Zielobjekt, eine arme Eule

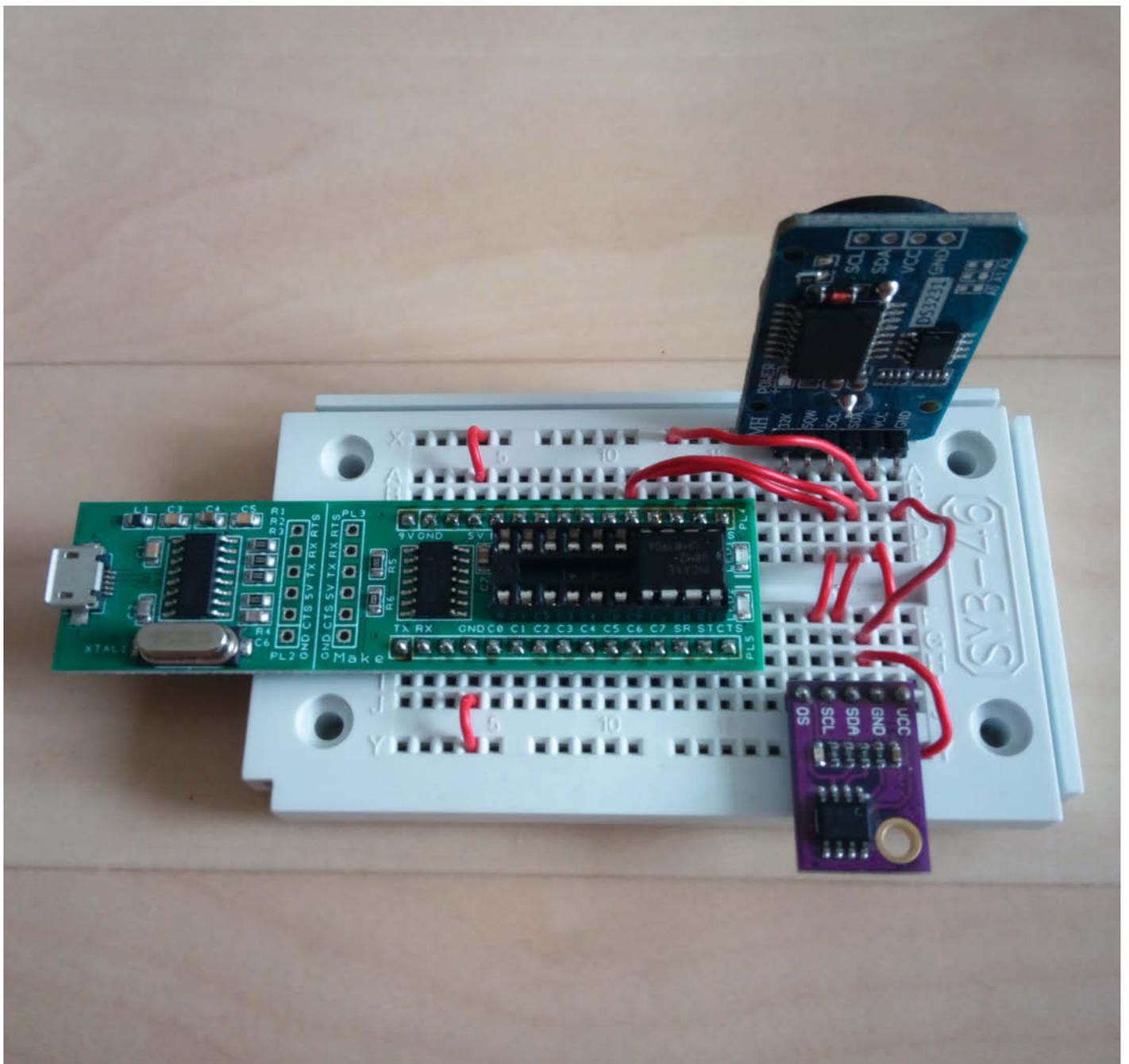


ebay.uk

I²C-Bus nach Teilnehmern abscannen

Beim Experimentieren mit dem I²C-Bus ist es hilfreich, wenn man prüfen kann, ob und welche Bausteine angeschlossen sind. Mit dem Nano-Axe-Board und einem Picaxe-Chip bauen wir dafür einen I²C-Scanner.

von Michael Gaus und Miguel Köhnlein



Verschiedene Sensoren, eine Uhr und externer Speicher – bei der Verwendung des seriellen Datenbus I²C können wir mehrere Bausteine hintereinander an einen Mikrocontroller anschließen und mit jedem einzeln kommunizieren. Unser I²C-Scanner hilft dabei den Überblick zu behalten, ob die unterschiedlichen Teilnehmer tatsächlich richtig angeschlossen sind. Dafür nutzen wir das *Nano-Axe-Board* und den *Picaxe-08M2*. Sie reichen bereits aus, um mit einem kleinen Software-Trick herauszufinden, auf welchen I²C-Adressen ein Teilnehmer mit dem Bestätigungsbit ACK reagiert. Das Nano-Axe-Board gibt es exklusiv mit dem Make Special PICAXE (siehe Kasten), weitere Hardware ist nicht nötig.

Mit diesem Scanner können unbekannte Adressen ausgelesen und bei Bausteinen mit externen Adresspins – wie dem Temperatursensor LM75A – die richtige Adresskonfiguration geprüft werden. Ebenso können wir eingrenzen, ob bei einer nicht-funktionierenden I²C-Kommunikation statt eines Softwareproblems möglicherweise ein Hardwarefehler vorliegt. Oft ist auch unklar, ob sich die im Datenblatt angegebene I²C-Adresse nun auf die 7-Bit-Interpretation ohne Schreib-/Lese-Bit (R/W-Bit) oder auf die 8-Bit-Angabe inklusive R/W-Bit bezieht. Mit dem I²C-Bus-Scanner lässt sich dies leicht bestimmen.

Beim Picaxe kann mit den I²C-Befehlen `hi2cout` und `hi2cin` eigentlich nicht herausgefunden werden, ob ein Teilnehmer reagiert hat. Bei `hi2cin` wird zwar bei nicht vorhandenem Teilnehmer standardmäßig der Wert `0xFF` zurückgeliefert – allerdings könnte dies auch ein gültiges Datenbyte sein, beispielsweise von einem EEPROM. Als Trick nutzen wir daher die folgende Methode: Unser Scanner versucht, nacheinander über alle Adressen (jeweils eingerahmt mit Start- und Stop-Bedingung) mit möglichen Teilnehmern zu kommunizieren und wertet aus, ob als Quittierung ein ACK gemeldet wird. Falls ja, ist an dieser Adresse ein I²C-Client vorhanden.

Kurzinfo

- » I²C-Scanner mit Nano-Axe-Board bauen
- » Ermitteln der Adressen von Teilnehmern eines I²C-Bus
- » Fehlersuche bei Projekten mit I²C-Bausteinen

Checkliste



Zeitaufwand:
eine Stunde



Kosten:
30 Euro



Programmieren:
Picaxe-BASIC

Material

- » Nano-Axe-Board mit Picaxe-08M2
- » Temperatursensor LM75A
- » DS3231-RTC-Modul

Mehr zum Thema

- » Michael Gaus und Miguel Köhnlein, Temperatursensor mit I²C auslesen, *Picaxe Special*, S. 34
- » Michael Gaus und Miguel Köhnlein, Für Halloween: Beleuchtetes Kürbis-Gesicht mit Nano-Axe-Board, online
- » Michael Gaus und Miguel Köhnlein, *PicAsSo* - Picaxe Adverts-Sonntagskalender, online

Alles zum Artikel
im Web unter
make-magazin.de/xw1a

In einer Schleife wird per `hi2csetup`-Befehl die anzusprechende Adresse nacheinander von `0x02` bis `0x7E` gesetzt und per `hi2cout`-Befehl ein Nullbyte als Datenbyte gesendet. Intern schreibt der Picaxe zunächst die I²C-Adresse in das Senderegister `SSP1BUF`. Wird die Adresse durch ein ACK quittiert, wird anschließend das Nullbyte ins Register geschrieben. Kommt jedoch aufgrund eines fehlenden Teilnehmers an dieser Adresse kein ACK, dann bricht `hi2cout` ab, sodass im Register die I²C-Adresse stehen bleibt. Im nächsten Befehl wird durch die Anweisung `peek sfr` das Register `SSP1BUF` gelesen. Falls Null drin steht, ist ein Teilnehmer an der Adresse vorhanden. Dieses Nullbyte sollte den Teilnehmer normalerweise nicht stören, da erst eine Registeradresse erwartet wird, bevor nachfolgende Daten eine Aktion auslösen. Die Idee stammt

aus dem Picaxe-Forum. Die dort beschriebene Methode haben wir so angepasst, dass die erkannten Adressen im 8-Bit-Format angezeigt werden – wie sie für den `hi2csetup`-Befehl beim Picaxe benötigt werden.

Durch die Verwendung des Nano-Axe-Boards ist für den Busscanner keine zusätzliche externe Hardware nötig. Die I²C-Teilnehmer werden direkt mit den Signalen SDA und SCL sowie mit GND verbunden. Optional kann die 5-Volt-Spannung des Nano-Axe zur Versorgung der Teilnehmer verwendet werden. Schließlich ist es sogar möglich, einen I²C-Bus mit 3,3-Volt-Logikpegeln zu scannen, da wir im Picaxe den System-Management-Bus (SM-Bus) aktivieren, in dem bereits ab 2,1 Volt ein High-Pegel erkannt wird.

Make Special PICAXE

Statt `void()` und Pointerarithmetik bietet das Picaxe-BASIC auch blutigen Anfängern verständliche Befehle mit vielen Funktionen zum Steuern von Hardware. Unser Sonderheft führt in die Grundlagen der BASIC-Programmierung und die interne Hardware des Chips ein. Anhand verständlicher und praktischer Projekte lernt man, die Befehle konkret einzusetzen. So gelingt die Programmierung eigener Projekte noch leichter als beim Arduino-C.

Das Special wird im Bundle mit einem Picaxe-08M2 und unserem eigens dafür entwickelten Programmier- und Test-Board *Nano Axe* geliefert. Damit lässt sich der Picaxe leicht per USB programmieren und debuggen. Das Board kann außerdem mit den Chips 14M2, 20M2 und 20X2 genutzt werden. Es ist Breadboard-kompatibel und verfügt über eine ansteuerbare LED und eine Status-LED. Auch die Schaltung aus Widerständen, die Picaxe-Chips am seriellen Eingang benötigen, ist bereits an Bord.



Das Heft kostet 24,95 Euro und ist versandkostenfrei über den heise Shop zu beziehen.

Beispielanwendung

Als Beispielanwendung schließen wir mit dem Temperatursensor LM75A und der Echtzeituhr DS3231 zwei I²C-Module am Nano-Axe-Board an. Die Uhr verbinden wir direkt mit dem Board, während der Sensor als Teilnehmer im Bus dahinter geschaltet wird. Im Schaltplan sind die Module im gestrichelt eingetrahmten Bereich dargestellt. Noch anschaulicher ist der Aufbau auf dem Steckbrett. Beim Einsetzen des Picaxe-Chips muss auf die richtige Orientierung geachtet werden – die Einkerbung muss mit der Markierung auf dem Board übereinstimmen.

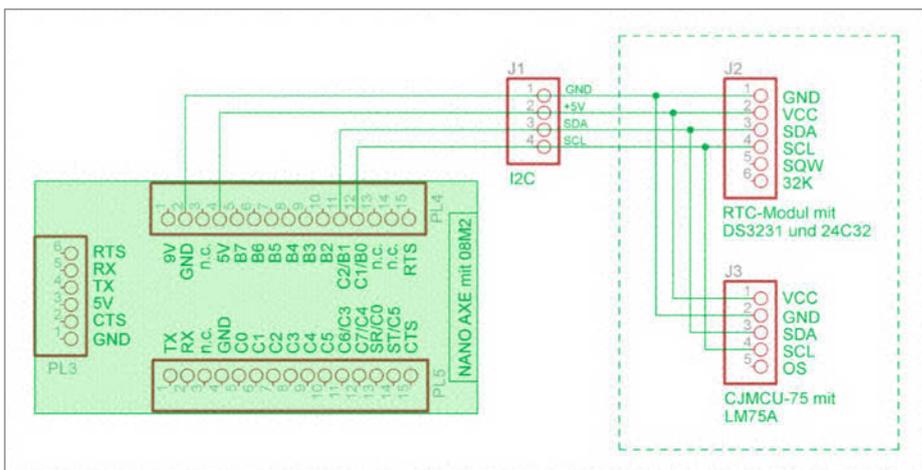
Für die Programmierung nutzen wir den kostenlosen Picaxe-Editor. Dort wird das Basic-Programm `i2c_bus_scanner.bas` geladen und auf das Nano-Axe-Board übertragen. Im seriellen Terminal sind die Textausgaben des I²C-Bus-Scanners sichtbar mit einer Baudrate von 4800 Bd.

Im Beispiel wurden drei I²C-Adressen gefunden, da außer dem LM75A (Adresse 90) und dem DS3231 (Adresse D0) noch ein I²C-EEPROM 24C32 (Adresse AE) auf dem RTC-Modul vorhanden ist. Die EEPROM-Adresse ist:

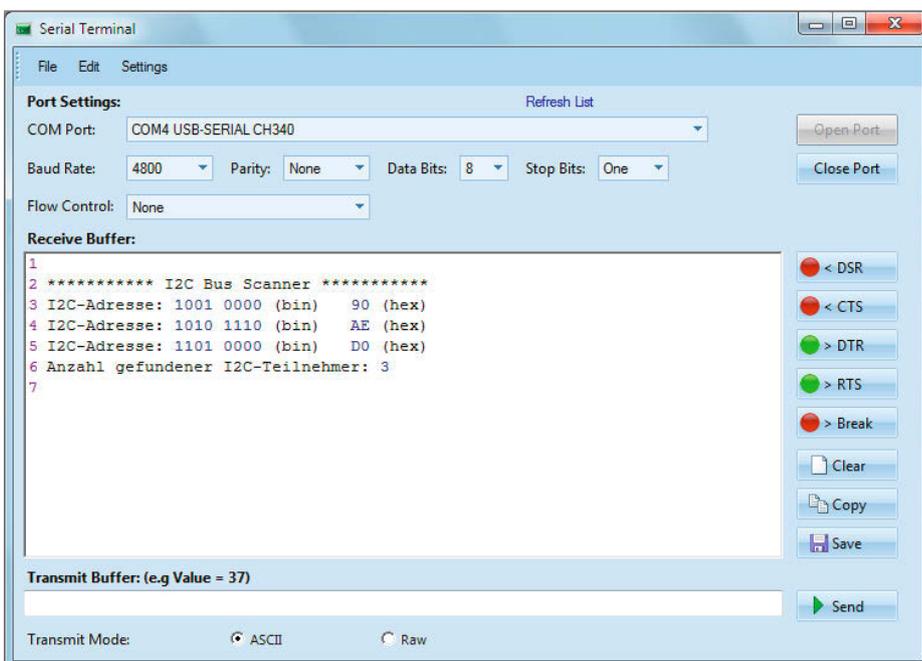
10 10 A2 A1 A0 0

Da die drei Adresspins A0-A2 auf dem Modul über Pullup-Widerstände mit 5 Volt verbunden sind, ergibt sich als binäre Adresse 10101110 oder AE in hexadezimaler Schreibweise.

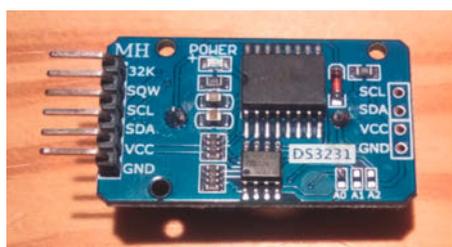
Das Programm haben wir mit dem Picaxe-Chip 08M2 getestet, es sollte jedoch auch mit den 14M2- und 20M2-Chips funktionieren. Dann ist zu beachten, dass die I²C-Signale auf anderen Pins des Nano-Axe-Boards liegen.



Beispiel-Schaltplan für Anschluss von I²C-Teilnehmern am Nano-Axe



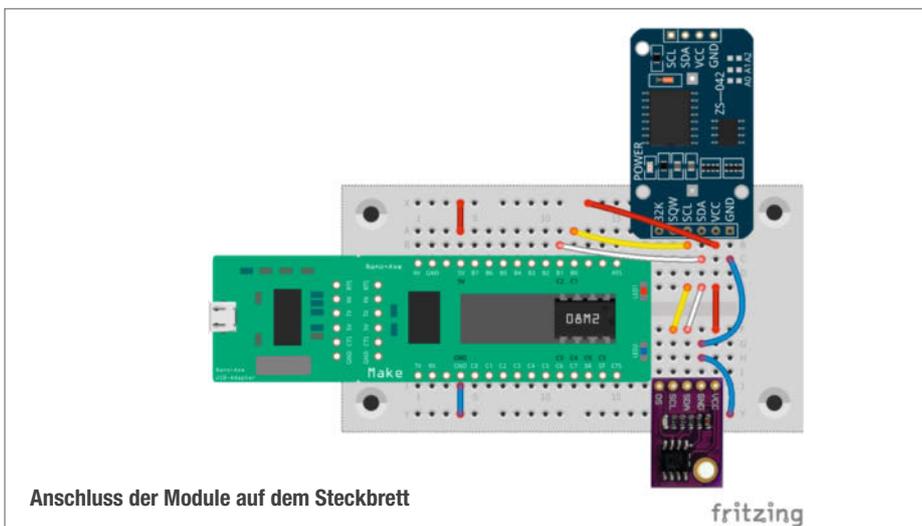
Erkannte I²C-Adressen in der Terminalausgabe



Vorderseite des Echtzeituhr-Moduls



Rückseite des LM75A-Moduls mit einstellbarer Adresse



Anschluss der Module auf dem Steckbrett

Quellcode

Zunächst sorgen wir mit `#terminal 4800` dafür, dass sich das serielle Terminal beim Start des Programms öffnet, auf dem die Adressen angezeigt werden sollen. Außerdem brauchen wir die Adressen für das I²C-Sende- und Statusregister: `0x91` bzw. `0x94`. Mit `sertxd` beginnt dann die Textausgabe auf dem Terminal.

Die I²C-Adressen werden nun in einer `for-next`-Schleife, beginnend mit Adresse 2, über den Befehl `hi2csetup` abgefragt. Damit das niedrigstwertigste Bit (LSB) bzw. das Schreib/Lese-Bit immer 0 beträgt und uns den schreibenden Zugriff erlaubt, erfolgt der Durchlauf in Zweierschritten.

Durch den Befehl `pokesfr` aktivieren wir das SM-Bus-Bit (`0x40`) im Statusregister, wodurch der Picaxe bereits ab 2,1 Volt einen HIGH-Pegel an den I²C-Pins erkennt. So ist es möglich auch Module zu nutzen, bei denen der I²C-Bus an 3,3 Volt angeschlossen ist. Es gibt beispielsweise OLED-Module, die zwar mit 5 Volt versorgt werden, jedoch nur 3,3 Volt als I²C-Pegel vertragen, weshalb die integrierten Pullup-Widerstände für SDA und SCL über einen Spannungsregler an 3,3 Volt angeschlossen sind.

Mit dem Befehl `hi2cout` versuchen wir anschließend, das Nullbyte zu senden. Wenn

auf die aktuelle I²C-Adresse kein ACK erfolgt ist, bricht `hi2cout` aber ab und sendet das Nullbyte nicht. Jetzt lesen wir mit `peeksfr` das Statusregister `SSP1BUF`, in dem das zuletzt gesendete Datenbyte steht. Solange kein Teilnehmer gefunden wurde, ist das die letzte angesprochene I²C-Adresse. Ist der Wert aber Null, wurde `hi2cout` komplett ausgeführt und das ACK eines I²C-Teilnehmers empfangen. In diesem Fall wird die Zählvariable `cnt` um 1 erhöht und die aktuelle Adresse auf dem Terminal angezeigt. Nach der Ausgabe im Binärformat ruft der Picaxe das Unterprogramm `printHex` auf, um den Wert in das hexadezimale Format umzurechnen und anzuzeigen. Falls im Register weder eine Null noch die letzte Adresse gespeichert sein sollte, kommt über die `else`-Abfrage eine Fehlermeldung – dies sollte allerdings nie passieren. Schließlich überprüft `next`, ob bereits alle Adressen abgefragt wurden.

Nachdem die `for-next`-Schleife komplett durchlaufen wurde, wird abschließend die Anzahl der gefundenen I²C-Teilnehmer auf dem Terminal ausgegeben. Das Abscannen wiederholen wir alle drei Sekunden, um neue Teilnehmer zeitnah zu finden. —hch

Hauptprogramm

```
#terminal 4800

symbol value = b0
symbol nibble = b1
symbol addr = b2
symbol cnt = b3

symbol SSP1BUF = 0x91
symbol SSP1STAT = 0x94

main:
  pause 3000
  sertxd (cr, lf, "***** I2C Bus Scanner *****", cr, lf)
  cnt = 0
  for addr = %00000010 to %11111110 step 2
    hi2csetup i2cmaster,addr,i2cslow,i2cbyte
    pokesfr SSP1STAT, 0x40
    hi2cout (0)
    peeksfr SSP1BUF, value
    if value = 0 then
      value = addr
      inc cnt
      sertxd ("I2C-Adresse: ")
      sertxd (#bit7,#bit6,#bit5,#bit4," ",#bit3,#bit2,#bit1,#bit0, "
(bin)  ")
      gosub printHex
      sertxd (" (hex)", cr, lf)
    else
      if value <> addr then
        sertxd ("Fehler: ",#value, cr, lf)
      endif
    endif
  next addr
  sertxd ("Anzahl gefundener I2C-Teilnehmer: ", #cnt, cr,lf)
  goto main
```

NEU

Selbermachen leicht gemacht!



c't PROJEKTE 2021

shop.heise.de/ct-projekte21
Heft + PDF
für nur

19,90 € >

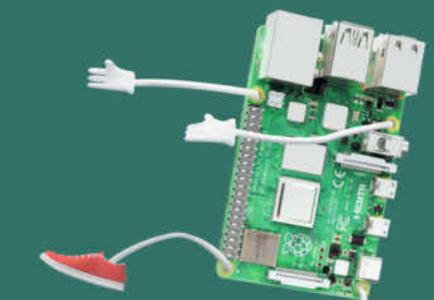


Bild: Andreas Martini

Generell portofreie Lieferung für Heise Medien- oder Maker Media Zeitschriften- Abonnenten oder ab einem Einkaufswert von 20 €. Nur solange der Vorrat reicht. Preisänderungen vorbehalten.

shop.heise.de/ct-projekte21

Holz-Challenge: Die Gewinner



Egal ob geschnitzt, geleimt, geschliffen oder CNC-gefräst – Hauptsache, aus Holz! Das war die einzige Vorgabe für Projekte, die bei unserem Sommer-Wettbewerb auf der Projektplattform Make Projects eingereicht werden durften.



von Peter König

Bei so viel Freiraum verwundert es eigentlich nicht, dass die Bandbreite unter den eingereichten 17 Projekten beeindruckend ist – sie reicht von selbst geschnitzten Gewürzlöffeln über E-Gitarre und Kajak bis zu raumfüllenden Möbelkonstruktionen wie Regalwand und Hochbett (samt Treppenschrank). Neben – klar, Holz – in verschiedenen Zustän-

den der Vorverarbeitung von der Baumscheibe bis zum ausgedienten Skateboard-Deck kamen aber auch Kunstharz, Metall, mechanische Bauteile und nicht zuletzt bei manchem Projekt auch eine gehörige Portion Elektronik zum Einsatz. Kein Wunder, dass es am Ende der Jury mal wieder schwerfiel, unter so vielen tollen Projekten drei heraus-

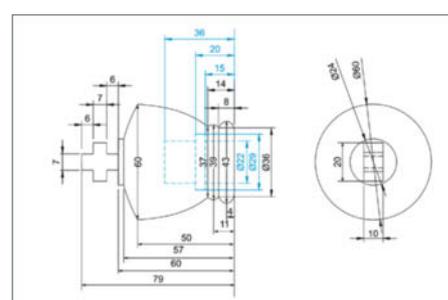
zusuchen, die mit Preisen belohnt wurden. Deshalb stellen wir hier nicht nur die drei Gewinnerprojekte vor, sondern auch noch einige andere bemerkenswerte. Auf *Make Projects* – am bequemsten zu erreichen über die Kurz-URL – gibt es aber noch viel mehr: mehr Projekte, mehr Bilder und mehr lesenswerte Beschreibungen.

1. Platz: Gedrechselte Schach-Mühlen

Britt (@brittliv) hat – anders als viele Maker – eine Drechselbank zur Verfügung und wollte sich nach ein paar gedrechselten Schalen zum ersten Mal an Längsholz heranwagen. „In einer grenzenlosen Überschätzung meiner Fähigkeiten habe ich mich dafür entschieden, Salz- und Pfeffermühlen zu drechseln, die wie Schachfiguren aussehen“, schreibt sie auf *Make Projects* – für ihren Vater, der ein großer

Schach-Fan ist. Auch wenn ihr Vorhaben ganz sicher eine echte Herausforderung war, überzeugt das Ergebnis doch absolut, sodass wir es mit dem ersten Preis (einem Baumarkt-gutschein über 300 Euro) ausgezeichnet haben. Das lag nicht nur am fertigen Projekt selbst, sondern ebenso an Britts liebevoller und detaillierter Darstellung jedes einzelnen Arbeitsschrittes in vielen schönen Fotos und

ausführlichen Texten, wobei sie auch zeigt, was zwischendrin schiefgelaufen ist. Am Ende passt aber alles, die gekauften Keramik-Mahlwerke sitzen prima in den Figuren aus Birken- und Walnussholz. Dazu gibt es auf *Make Projects* auch alle Planzeichnungen und Maße für die Figuren und sogar 3D-Dateien im STL-Format zum Download, falls jemand solche Mühlen lieber drucken als drechseln möchte ...



2. Platz: Puppenhaus im Maßstab 1:12

Das Puppenhaus, das **Stefan (@StefanB)** für seine vierjährige Tochter gebaut hat, wirkt mit seinen Sprossenfenstern und den sichtbaren Mauersteinen sehr nostalgisch – „viktorianischer Stil“ sei das Ziel gewesen, schreibt er auf *Make Projects*. Beim Bau kam allerdings viel moderne Technik zum Einsatz: So entstand der Entwurf in der 3D-Software *Sketchup*, in der Werkstatt machten Tischkreissäge und Kantenfräse Krach und die filigranen Sprossenfensterrahmen und viele andere Teile entstanden auf einer Eigenbau-CNC-Fräse. Der „Keller“ des Hauses, der als Schublade ausgeführt ist, nimmt gerade nicht genutzte Spielutensilien auf und er enthält auch ein Fach für die Batterie, die für die elektrische Beleuchtung der einzelnen Räume nötig ist. Allen benutzten Maschinen zum Trotz steckt in dem Puppenhaus allerdings auch jede Menge liebevoller Handarbeit – so hat Stefan beispielsweise alle Stufen der beiden Treppengiebel von Hand mit der Japansäge herausgearbeitet und die Naturstein-Optik der Fassade mit einem Dremel samt Kugelfräser herausgearbeitet,



indem der diesen bei fester Frästiefe an einem angelegten Metalllineal entlang führte. In diesem Puppenhaus steckt mehr, als man auf den ersten Blick meinen mag, weshalb wir seinem Architekten und Erbauer einen der beiden zweiten Preise zusprachen (ein Make-Miniabo Plus).

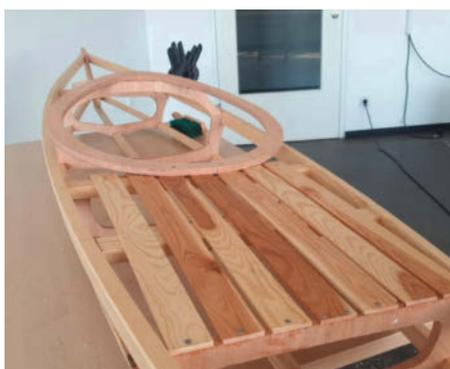


2. Platz: Ein Kajak und ein Paddel für die Tochter

„Die Baupläne sind gekauft, alles andere ist selbst gemacht.“ So lautet lapidar die Kurzbeschreibung des für die Holz-Challenge eingereichten Projekts von **Max (@maximi-maxima)**. Der fährt „für sein Leben gern“ Kajak, wie er schreibt, und wollte seiner achtjährigen Tochter die Möglichkeit geben, es ebenfalls auszuprobieren. Kinderkajaks sind eher selten im Verleih, warum also nicht gleich eines selber bauen, statt es zu kaufen? Der Entwurf eines solchen Boots ist natürlich ein ganz eigenes Projekt und erfordert Erfahrung, deshalb kaufte sich Max lieber einen Plan für eine Konstruktion, bei der über ein Gerippe aus Kiefernleisten und Spanten aus Bootsbau-sperrholz eine Bespannung aus sogenanntem *ballistischem Nylon* gezogen wird. Eine spe-

zielle Beschichtung sperrt anschließend das Wasser aus. Während die Beschichtung trocknete, hat Max noch kurzerhand aus einem Kiefernkantholz das passende Paddel ausgesägt und vor allem gehobelt – oder genauer gesagt, zwei Paddel, weil er den ersten Versuch „nicht so doll“ fand. Apropos zwei:

Seinem Fazit auf *Make Projects* nach zu schließen, wird das wohl nicht das letzte Kajak aus der heimischen Werft gewesen sein ... Wir waren jedenfalls so überzeugt von diesem ersten Boot, dass wir Max den zweiten der beiden zweiten Preise zusprachen (ein Make-Miniabo Plus).

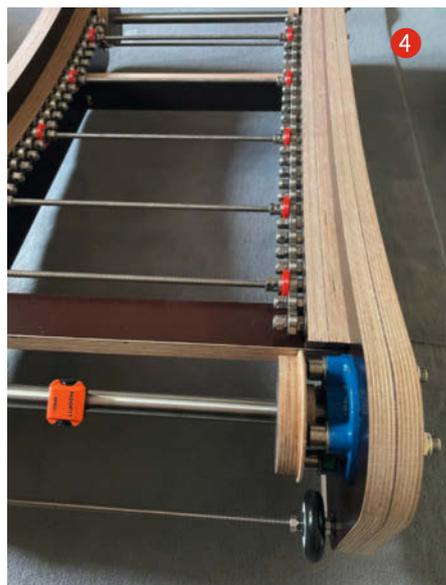


Mit High-Tech: E-Gitarre, Tretmühle und LED-Regal

Natürlich hatten wir als Jury noch eine Menge anderer Kandidaten auf der Shortlist, weshalb wir hier noch unbedingt ein paar weitere Projekte aus der Challenge vorstellen müssen. Eines davon ist die **Skateboard-Gitarre** von @54m, der schreibt: „Knapp 400 Tage nach dem allerersten Konzept, rund 60 Stunden in CAD und 24 Stunden in der Werkstatt später, ist mein Gitarrenbau nun abgeschlossen. Entstanden ist ein bunter Kanarienvogel und eine der am angenehmsten bespielbaren Gitarren, die ich je in der Hand hatte.“ ① Die Kanarienvogel-Anmutung rührt vom Korpus-Material her, denn dessen Deckschicht besteht aus zerrockten Skateboard-Decks, die mit Hilfe der Furnierpresse im *Holzkombinat* Chemnitz zu einem mehr oder weniger homogenen Schichtholzblock verleimt und später in Form gefräst wurden ②. Ansonsten ist die Gitarre mit ihrem *Headless-Design*, den sieben Saiten und den aufgefächerten Bundstäben auf dem Griffbrett alles andere als ein Instrument von der Stange – und ihre ganze Entstehungsgeschichte gibt es ausführlich auf *Make Projects* nachzulesen.

Die **Tretmühle** von **Claudius Albat** (@ClaudiusA) hingegen wirkt in ihrer gedeckten Siebdruckplatten-Optik auf den ersten Blick eher unspektakulär ③ – aber im Inneren steckt eine ausgeklügelte Mechanik ④. Zweck des ganzen ist nämlich, als praktisches kleines Endlos-Laufband für Bewegung der Beine bei der Büroarbeit am Stehtisch zu sorgen. Die Siebdruckplatten sind dabei nicht nur ziemlich robust, sie bieten gerade durch ihre strukturierte Oberfläche auch den passenden *Grip* fürs Laufen auf der Stelle. Die Seitenteile wurden mit einer selbstgebaute CNC-Fräse angefertigt, die Latten des Bandes mit der Tischkreissäge um 15 Grad abgeschrägt, damit Zehen keinesfalls Gefahr laufen, eingeklemmt zu werden. Insgesamt 54 Kugellager sorgen im Inneren für einen reibungslosen Lauf und ein Bluetooth-Sensor misst die Gehgeschwindigkeit. Wahrlich innere Werte!

Das **LED-Regal** von **Martin H.** (@maatin) erweitert das bereits vorhandene Spirituosen-Regal und sorgt gleichzeitig über LED-Streifen für stimmungsvolle Beleuchtung ⑤. Als Vorbild diente ihm eine Konstruktion, die eigentlich für den 3D-Druck vorgesehen war, aber diese Fertigungstechnik schied aus, weil seine Erweiterung aus dem gleichen Holz sein sollte wie das vorhandene Regal. Mit der Kreissäge schnitt er Nuten in die Stirnseite der Bretter, in die dann RGB-LED-Streifen gelegt wurden, die er anschließend mit Kunstharz vergossen hat. Weiße Acrylfarbe dient als Diffusor, um das Licht zu streuen. Neben diversen Lichteffekten kann das Regal auch als Reihe großer Sieben-Segment-Anzeigen dienen und die Uhrzeit anzeigen ⑥.



Back to the Woods: Holz (fast) pur

Elnige Projekte, die für unseren Wettbewerb eingereicht wurden, stellten auch gewachsenes Massivholz in seiner noch erkennbar natürlichen Form in den Mittelpunkt. Oft geht es dabei um Holz, das man selbst noch als lebendigen Baum gekannt hat, wie im Fall von **Mario Redel (@M.Redel)**, der zusammen mit seiner Freundin Bretter aus dem Kirschbaum aus dem Garten seiner Eltern zu einer **Esstischplatte** verarbeitet hat **7**. Der Baum spaltete sich auf und musste gefällt werden – die daraus geschnittenen Bretter wurden getrocknet, gehobelt und in eine selbstgebaute flache Gießform gelegt, in die anschließend über 20 Liter gefärbtes Epoxidharz eingegossen wurden. Nach zwei Wochen Trocknungszeit konnte dann aus dem entstandenen flachen Block die fertige Tischplatte gefräst und geschliffen und mit Haarnadelbeinen versehen werden. Solche filigranen Beine aus Metall nutzte auch **@Taam** für ihren **Wohnzimmertisch**, doch kombinierte sie diese mit einer massiven Baumscheibe, per Kettensäge aus dem vollen Stamm geschnitten **8** und dann per Hand entrindet, geglättet und geölt.

Viel handlicher sind hingegen die Holzprojekte von **Jan Kutzt (@Jan2021)**, denn er schnitzt **Gewürzlöffel** aus den unterschiedlichsten Holzarten **9**. Solche Löffel sind ein ideales Einsteigerprojekt, denn der Werkzeugaufwand ist ebenso gering wie der Materialverbrauch. Zudem liest sich seine wunderbare Projektvorstellung eher wie eine Schritt-für-Schritt-Anleitung zum Selbermachen (inklusive der Beschreibung einer selbstgebauter Schleifmaschine für Spezialmesser aus



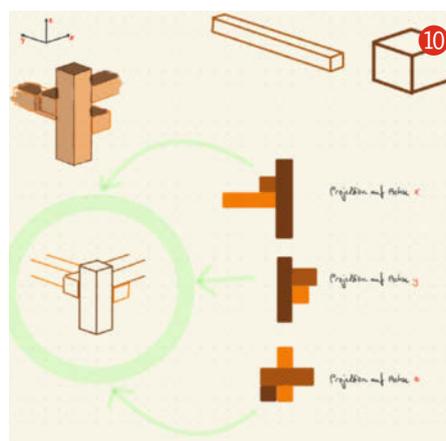
einer alten Festplatte). Viel schiefgehen kann außerdem auch nicht, sagt Jan: „Das Schöne an der Arbeit mit Holz ist, dass es fast keine Situation gibt, die so katastrophal ist, dass es der Löffel nicht überlebt. Risse im Holz werden

mit Sekundenkleber oder mit Zinn gefüllt. Sollte der Löffel am Schaft hinter der Kelle brechen, kann ich einen Stahlstift mit 2K-Klebstoff einsetzen und bekomme immer noch ein brauchbares Ergebnis hin.“

Noch mehr (Lese)stoff

Klar, das war noch nicht alles, deshalb lohnt sich ein Besuch auf *Make Projects* unbedingt: So kann man **Sebastian Löwe (@Semok)** samt Familie bei der kollektiven Planung und Umsetzung des **Hochbetts mit Treppenschrank** für den unweigerlich älter werdenden Nachwuchs quasi über die Schulter ins Skizzenbuch schauen – vor dem Baubeginn stand hier nämlich viel Zeichenarbeit auf der Suche nach der perfekten Lösung **10**. **Christian Knappe (@Christian)** hingegen tritt bei seinem **Karl X**, dem Regal unter der Dachschräge, den Beweis an, dass die 3D-Programmierungsumgebung *OpenSCAD* durchaus auch zur Planung von Möbeln taugt. Wunderschön anzuschauen sind **Nachtlicht und Wecker**, die **@fürlefranz** für seine Tochter gebaut hat **11**. Alles in allem waren wir begeistert von den Einreichungen

zu unserem Holz-Wettbewerb und freuen uns schon sehr auf die kommenden Challenges bei *Make Projects*! —pek



DIY-Gehäuse schnell gebaut

Durch die geschickte Kombination von Baumarktmaterial wie Holzplatten für den Korpus und Aluprofilen für die Frontplatte entstehen mit minimalem Werkzeugaufwand wertige, professionell anmutende Gehäuse für selbst gebaute Geräte. Hier ist das Rezept dafür.

von Kurt Diedrich



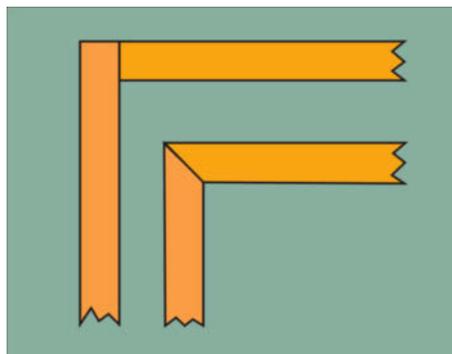
Erst durch Knöpfe, Schalter und Buchsen wird eine nackte Platine zu einem fertigen Gerät, das man zum Beispiel im Labor einsetzen kann. Während heute fast alle benötigten Bauteile für die Elektronik im Internet zu finden sind, ist es um die Vielfalt der dazu notwendigen Gehäuse oft weniger gut bestellt. Viele der im Versandhandel angebotenen Exemplare aus Kunststoff erinnern eher an Dosen zur Aufbewahrung von Pausenbroten als an die schicken Gehäuse renommierter Gerätehersteller, sodass oft nur der Selbstbau in Frage kommt. Der scheitert jedoch häufig am mangelnden handwerklichen Know-how. Dieser Beitrag zeigt, wie sich mit gut erhältlichen Materialien auf einfache Weise ein ansprechendes Gehäuse herstellen lässt. Die sich ergebende Frontplattenhöhe von 4 beziehungsweise 6 Zentimetern ist für viele Laborgeräte ein geeignetes Maß, das zudem noch das Übereinanderstapeln mehrerer solcher Gehäuse erlaubt.

Zeitbedarf

Die hier gezeigte Anleitung ist allerdings nichts für Ungeduldige: Obwohl die Bearbeitung und die Montage der Teile in vier bis fünf Stunden zu schaffen ist, sollte man für die komplette Bauzeit dennoch etwa vier bis fünf Tage veranschlagen, weil das Trocknen des Klebstoffs mehrere Stunden und das Trocknen der idealerweise zweimal aufgetragenen Lackschicht sogar jeweils zwei Tage benötigt. Die Wartezeit lässt sich durch das Anfertigen der Frontplattenfolie überbrücken, ein Vorgang, der je nach Zeichentalent und Übung im Umgang mit Grafik-Software und tückischen Klebefolien schon mal einen Tag oder länger dauern kann.

Kosten

Da die Preise für Holz zurzeit stark variieren, lassen sich keine genauen Angaben zu den Gesamt-Materialkosten machen, die sich durchaus auf 30 bis 40 Euro belaufen können. Hinzu kommen noch einmal ca. 10 bis 20 Euro



1 Montage der Seitenteile: rechtwinklig oder auf Gehrung

Kurzinfo

- » Gehäuse auf Maß mit Baumarktmaterial
- » Stapelbar, robust und optisch ansprechend
- » Frontplattenbeschriftung aus dem Fotodrucker

Checkliste



Zeitaufwand:

4 bis 5 Stunden reine Arbeitszeit, wegen Trockenzeiten verteilt auf 4 bis 5 Tage



Kosten:

40 bis 60 Euro, je nach Größe



Computer:

mit CAD- oder Vektorgrafiksoftware und Drucker für Fotopapier

Material

- » 2 Multiplexplatten 8 bis 12mm stark, Maße nach eigenem Bedarf, für Deck- und Bodenplatte (identische Abmessungen)
- » Buchenleisten 4 bis 6cm breit, Länge ausreichend für zwei Seitenwände und eine Rückwand
- » Alu-Winkelschiene ebenfalls 4 bis 6cm breit, wie die Buchenleiste, für die Frontplatte
- » 4 Holzleisten Länge wie Höhe der Frontplatte, Querschnitt 1,5 bis 2cm im Quadrat
- » 4 Gehäusefüße
- » Fotopapier für den Drucker
- » transparente Klebefolie
- » Holzschrauben zur Deckelbefestigung, Torx empfohlen
- » farbloser Lack etwa Parkett- oder Bootslack

Werkzeug

- » Säge für Holz mit Gehrungslade oder Gehrungssäge
- » Metallsäge
- » Feile für Metall
- » Bohrmaschine möglichst mit Bohrständler, Bohrer verschiedener Durchmesser, eventuell Bohrfräser oder Stufenbohrer
- » Oberfräse oder Schwingschleifer zum Abrunden der Kanten, optional
- » Hammer und Körner
- » Holzleim und Zwingen zum Pressen
- » Alleskleber
- » Cutter oder Lochstanz-Werkzeug
- » Schleifpapier
- » Lackroller

Mehr zum Thema

- » Dominik Laa, Gehäuse aus dem 3D-Drucker, Make 2/19, S. 18
- » Carsten Meyer, Maßgeschneiderte Gehäuse, c't Hacks 3/14, S. 58

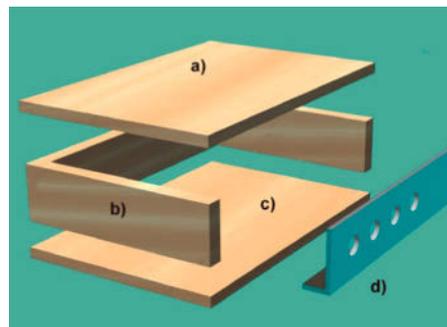
Alles zum Artikel im Web unter make-magazin.de/xp5j

für Holzleim und weitere Kleinteile, wobei Sie die Leisten, den Leim und die Kleinteile auch zum Bau eines weiteren Gehäuses verwenden können. Die Kosten für individuelle Frontplat-

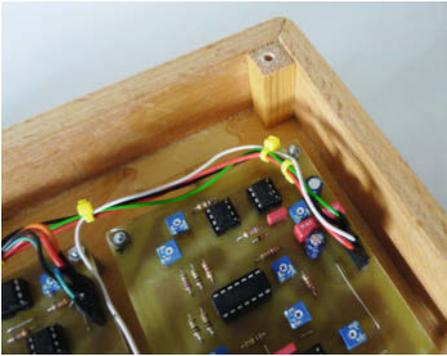


Sicherheitshinweise

Dass man im Umgang mit Elektrowerkzeugen besonders vorsichtig sein muss, versteht sich von selbst. Besonders beim Bohren von Löchern in Metall können uns Schutzhandschuhe vor Schäden durch rotierende Bohrspäne bewahren. Vor umherfliegenden Metallspänen (und Holzsplittern) schützt man sich durch spezielle Arbeitsbrillen, die auch mit optischen Gläsern erhältlich sind. Beim Schleifen, Schmirgeln und Feilen ist eine Staubmaske empfehlenswert, da das Einatmen von Holzstaub gesundheitsschädlich sein kann.



2 Anordnung der Einzelteile: a) Deckplatte (nicht verleimen), b) Seitenteile, Rückwand, c) Bodenplatte, d) Frontplatte (Alu-Winkelpprofil)



3 Befestigungsleisten für den Deckel



4 Profilansicht: Abgerundete Kanten von Deckel und Boden sowie Position der Frontplatte

ten-Bedienelemente sind in diese Kalkulation nicht mit einbezogen.

Drei Materialien

Abgesehen von einigen weiteren Kleinteilen benötigen Sie hauptsächlich drei unterschiedliche Materialien aus dem Baumarkt: *Deck- und Bodenplatte* aus Multiplex-Platten, *Seitenwände und Rückwand* aus Buchenleisten sowie eine *Frontplatte* aus einer Alu-Winkelschiene.

Deck- und Bodenplatte

Die beiden gleich großen Platten aus (empfohlen) ca. 8 bis 12mm dickem Multiplex-Material erhalten Sie am Holz-Zuschnitt Ihres Baumarktes. Prüfen Sie, ob beide Platten auch deckungsgleich sind. Die Abmessungen richten sich natürlich nach dem Innenleben Ihres Gehäuses. Eine ansprechende Form ergibt sich, wenn Länge und Breite des Gehäuses

einem DIN-A4-Blatt bzw. dessen Seitenverhältnis entsprechen, aber natürlich sind auch abweichende Maße und Verhältnisse möglich.

Achtung: Gehäuse für Schaltungen, die Wärme erzeugen, müssen mit ausreichend Lüftungslöchern versehen werden. Für eine gute Luftzirkulation empfehlen sich Löcher im Boden und im Deckel. Als Bohrschablone kann ein Lochblech aus Metall verwendet werden.

Seitenwände und Rückwand

Viele Baumärkte bieten gehobelte Buchenholzleisten von 1cm Dicke, 2m Länge und 4 bzw. 6cm Breite an. Buchenholz ist zwar sehr hart, lässt sich jedoch recht gut bearbeiten. Achten Sie darauf, dass Sie keine krummen Leisten erwischen. Schneiden Sie die Leisten mit einer guten Gehrungssäge entweder auf Gehrung oder rechtwinklig 1. Vorsicht: Minderwertige Gehrungssägen oder Gehrungssleihen erzeugen oft schiefe Schnitte, so dass beide Teile nicht sauber aneinander grenzen. Kleinere Fehler lassen sich jedoch mit farblich passendem Holzkitt ausbügeln.

Frontplatte

Baumärkte bieten eine breite Vielfalt günstiger Alu-Schienen mit rechtwinkligem Profil an. Wählen Sie eine Schiene, bei der eine Seite die gleiche Höhe wie Ihre Buchenholzleiste aufweist (also 4 oder 6cm). Die Breite der anderen Seite sollte ca. 1 bis 2cm betragen. Schneiden Sie die Schiene auf passende Länge und bohren Sie die Löcher für die Potis, Schalter und Buchsen (sowie die Löcher zur Befestigung am Boden), eventuell nach einem per CAD- oder Vektorzeichenprogramm gefertigten Bohrplan – dazu gleich mehr.

Bohren Sie die Löcher, nachdem sie mit einem Körner markiert wurden, zuerst mit einem dünnen, scharfen Bohrer von ca. 3 bis 4mm vor. Für größere Löcher (über 10mm) empfiehlt sich die Verwendung einer Metallfräse zum Einspannen in die Bohrmaschine (Ständerbohrmaschine empfohlen). Entfernen Sie überstehende Säge- und Bohrgrate mit einer Metallfeile.

Zusammenbau

Leimen Sie Boden und Seitenteile (Teile **b**) und **c**) auf Bild 2) mit Holzleim und Zwingen zusammen. Der Klebstoff trocknet bereits nach ein paar Stunden extrem fest. Verleimen Sie kleine Holzstäbe mit 4 oder 6cm Länge und ca. 1,5 bis 2cm Kantenbreite an den Innenseiten, damit die Schrauben zur Befestigung des Deckels guten Halt finden und nicht zu dicht am Rand sitzen 3.

Bohren Sie nun ein Loch an jeder Ecke des Deckels, das dem Durchmesser der Befestigungsschrauben entspricht. Ich empfehle Torx-Holzschrauben, da Kreuzschlitzschrauben schnell ausleiern. Runden Sie die oberen Kanten des (festgeschraubten) Deckels und die unteren Kanten des Bodens mit einer Oberfräse oder mit einem Schwingschleifer ab. Dies erfordert allerdings etwas Übung 4.

Lackierung

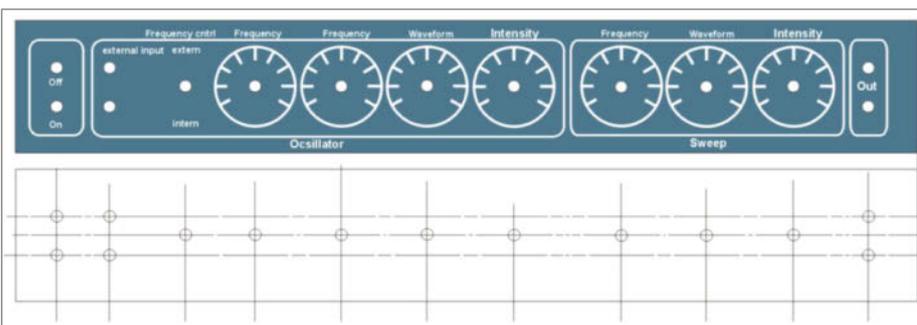
Mein Vorschlag: Lackieren Sie das fertige Gehäuse (ohne Frontplatte) innen und außen mit farblosem Parkett- oder Bootsack. Dadurch wird die Maserung des Holzes deutlich hervorgehoben. Nach der ersten Lackierung könnten sich die Fasern des Holzes von Deckel und Boden aufrichten, sodass die Oberfläche rau wird. Bearbeiten Sie das Gehäuse in diesem Fall mit ganz feinem Schmirgelpapier und tragen Sie eine zweite Lackschicht auf. Lassen Sie den Lack zwischen den einzelnen Arbeitsschritten mindestens zwei Tage lang trocknen. Niemals schmirgeln, wenn sich der Lack noch klebrig anfühlt.

Tragen Sie den Lack mit einer Schaumstoff-Lackrolle auf und vermeiden Sie unschöne, herabrinnende Lacktropfen. Man erkennt sie am besten, wenn sich das Gegenlicht in der Oberfläche spiegelt.

Frontplatten-Beschriftung und -Montage

Zeichnen Sie mit einem CAD- oder Vektorzeichenprogramm wie *CorelDraw* oder *Inkscape* die zum Gehäuse passende Frontplatte mit Beschriftung, den Skalen für die Potis und eventuellen Symbolen. 5 zeigt die Vorlage für ein Beispiel, 6 die fertige Frontplatte. Drucken Sie die farbige Frontplatte mit einem Tintenstrahldrucker auf Fotopapier. Ein zusätzlicher schwarzweiß-Ausdruck kann als Bohrplan verwendet werden.

Bekleben Sie Vorder- und Rückseite des Ausdrucks mit hochwertiger und möglichst dicker, durchsichtiger Klebefolie. Hierbei ist größte Sorgfalt und Sauberkeit erforderlich, damit kein Staub und Schmutz unter die Folie kommen und sich beim Aufkleben keine Luftblasen bilden. Das geht am besten zu zweit und mit einem Tuch zum Andrücken. Wer ein Laminiergerät in passender Größe hat, kann auch das



5 Frontplattengrafik und Bohrplan



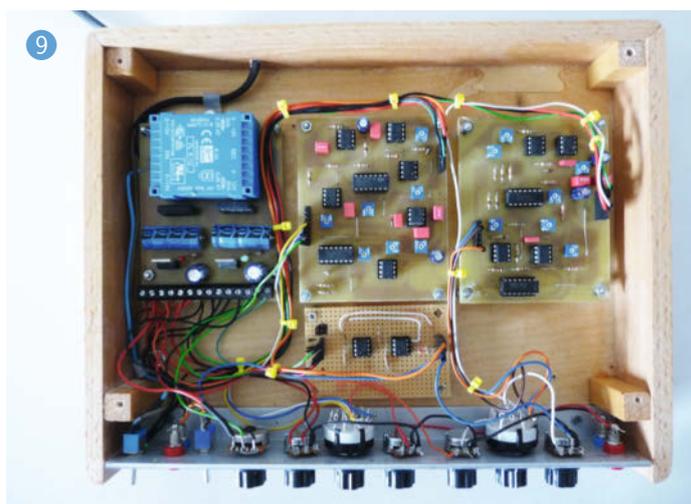
6 Die fertige Frontplatte meines selbstgebaute Funktionsgenerators



7



8



9

für diesen Zweck benutzen. Schneiden Sie anschließend die gedruckte Frontplatte so aus dem Fotopapier, das sie etwas größer als die Alu-Schiene ist, damit genügend Spielraum zum Verschieben und Anpassen existiert.

Schneiden Sie die Löcher für die Potis, Buchsen und Schalter mit einem Cuttermesser (oder besser noch einem Lochstanz-Werkzeug) aus und kleben Sie die danach mit einem gut verteilten Alleskleber-Film versehene Folie auf die Alu-Schiene, und zwar so, dass die Löcher in der Folie und in der Alu-Schiene zur Deckung kommen.

Wenn der Kleber getrocknet ist, können Sie die Ränder der überstehenden Folie mit einem Cuttermesser entlang den Kanten der Alu-Schiene abschneiden. Eventuell in ihrer Position geringfügig abweichende Löcher in der Alu-Schiene können Sie noch mit einer Metall-Rundfeile an den Ausdruck anpassen. Damit sich der Fotopapier-Ausdruck an den Rändern nicht ablöst, sollte er mit einer weiteren Klar-sichtfolie überklebt werden, deren Ränder um die Ränder der Alu-Schiene herumgebogen werden. Damit keine Luftblasen an den Kanten entstehen, empfiehlt sich ein festes Anreiben über die Kanten mit einem Tuch.

Befestigen Sie nun die Frontplatte wie in 4 gezeigt am Gehäuseboden.

Der letzte Schliff

Nun fehlen nur noch das Anbringen passender Gehäusefüße und die Montage der Frontplatten-Elemente, um Ihrem selbstgebaute Gehäuse den letzten Schliff zu verleihen. Die

Platinen im Inneren werden über Distanzhülsen mit dem Gehäuseboden verschraubt. Die Bilder 7 bis 9 zeigen meinen nach diesem Muster selbst gebaute Funktionsgenerator, Bild 10 einen ELF-Receiver mit Arduino-Recorder. —pek

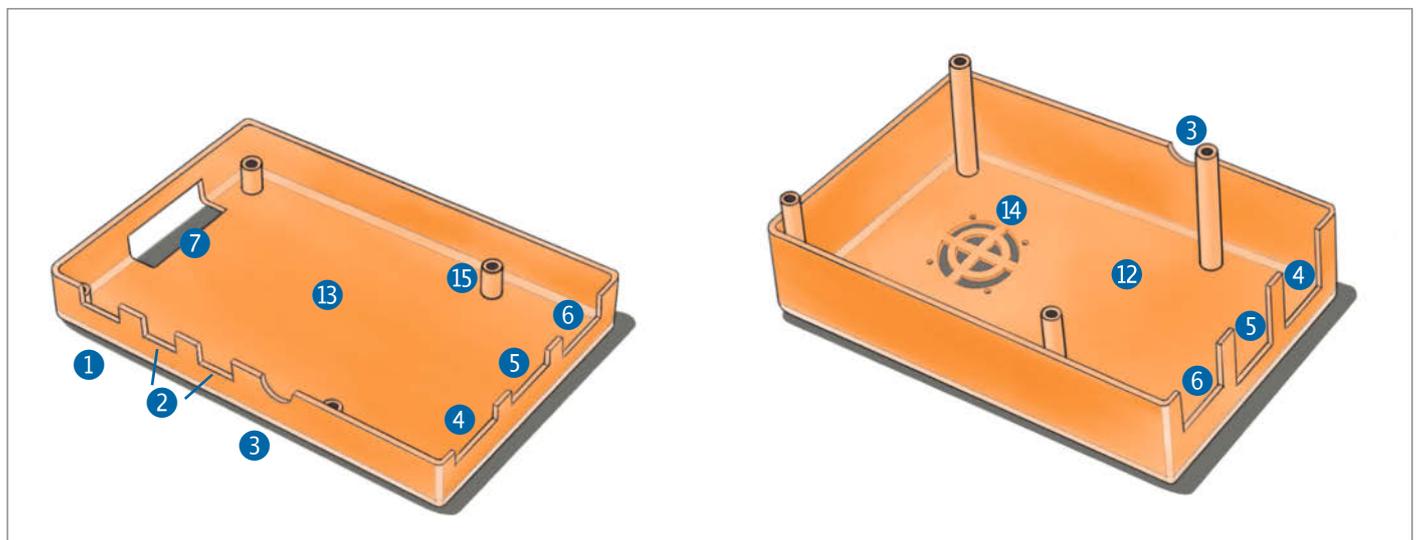
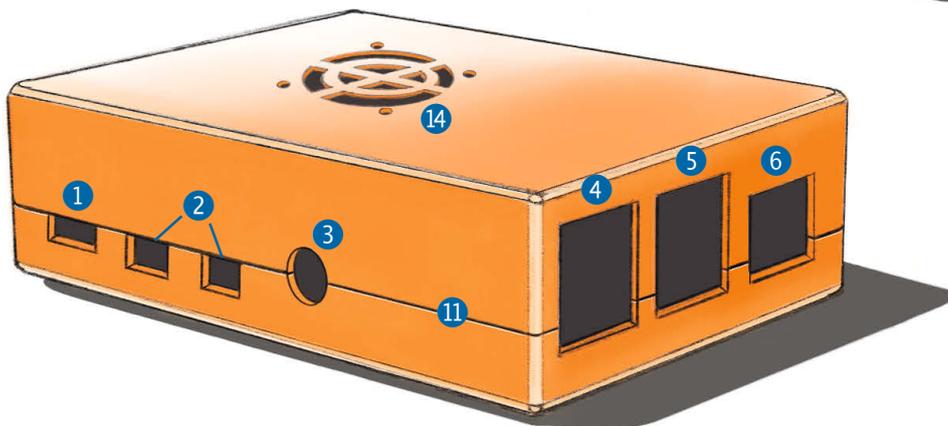
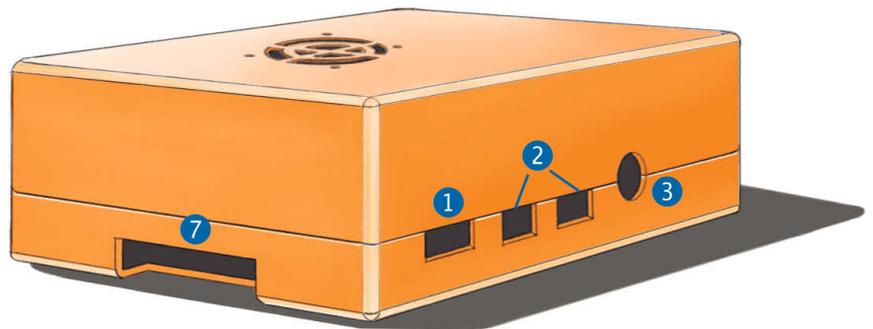


10

Gehäusebau mit FreeCAD, Teil 1

Kleinstrechner wie der Raspberry Pi kommen ohne Gehäuse daher. Diese kann man zwar dazukaufen, das ist aber nicht ganz Maker-like. In unserer neuen Artikelserie zeigen wir deshalb am Beispiel des Raspi, wie Sie mit der kostenlosen Software FreeCAD Ihr eigenes Wunschgehäuse für einen Einplatinenrechner konstruieren, um es auf einem 3D-Drucker zu fertigen.

von Matthias Mett



In vergangenen Make-Ausgaben haben wir mit mehreren Beispielprojekten wie einem Zirkel, einem Stempel und einem Ring die ersten Schritte der Arbeit mit der Open-Source-Konstruktionssoftware *FreeCAD* gezeigt. Jetzt steht ein größeres Vorhaben auf dem Programm: Wir zeigen Ihnen in mehreren Artikeln in den nächsten Heften Schritt für Schritt, wie Sie für den Raspberry Pi 4 Model B oder einen ähnlichen Einplatinenrechner ein Gehäuse konstruieren können, das genau die von Ihnen benötigten Maße und Ausschnitte etwa für Lüfter aufweist. Zum Start geht es in diesem Artikel um die grundsätzlichen Designentscheidungen. Zur Veranschaulichung haben wir vom geplanten Gehäuse Designzeichnungen erstellt (siehe Titelseite des Artikels), um die Ideen im Vorhinein besser darzustellen. Ab der nächsten Folge der Serie geht es dann konkret mit der Konstruktion los, wobei Sie im Heft nur eine kurze Beschreibung der Vorgehensweise finden, die Sie dann jeweils mit Hilfe eines ausführlichen Videos Schritt für Schritt in *FreeCAD* nachvollziehen können.

Die Füllung ...

Der Raspberry Pi 4 Model B ist ein Einplatinencomputer, der je nach Größe des Arbeitsspeichers ab ca. 40 Euro zu haben ist. Er besteht aus einer Platine mit Anschlüssen, über die sich Peripheriegeräte wie Tastatur und Maus verbinden lassen. Dazu gekauft haben wir noch ein passendes Netzteil, eine microSD-Speicherkarte für das Betriebssystem und einen HDMI-Adapter für den Anschluss an einen Monitor oder Fernseher. Außerdem wollen wir noch Kühlkörper und einen Lüfter einbauen, um den Raspi später ohne Hitzeprobleme gegebenenfalls übertakten zu können.

Beim Raspberry Pi kamen seit seiner ersten Version im Laufe der Zeit immer mehr Anschlüsse hinzu. Bei unserem aktuellen Modell gibt es einen USB-C-Anschluss für die Stromversorgung ①, zwei Micro-HDMI-Anschlüsse ②, einen 3,5mm Klinkenanschluss ③, je zwei USB-2.0- und USB-3.0-Anschlüsse ④ ⑤, einen Gigabit-Netzwerkanschluss ⑥ und – nicht zu vergessen – den Einschub für die microSD-Karte ⑦. Alle Anschlüsse sollen im Gehäuse entsprechend ausgespart sein, um sie direkt von außen erreichen zu können, ohne das Gehäuse zu öffnen. Die Steckerleiste mit den GPIO-Pins ⑧ könnte man im Gehäusedeckel ebenfalls aussparen. Davon sehen wir aber in diesem Beispiel der Einfachheit halber erst mal ab; am Ende unseres Workshops sollte es für Sie allerdings kein Problem sein, solche Aussparungen selbst vorzusehen. Die Platine besitzt zur Befestigung in der Nähe der Ecken jeweils eine Bohrung ⑨. Damit lässt sie sich zuverlässig in einem Gehäuse positionieren und fixieren, wovon wir im Folgenden Gebrauch machen werden.

Kurzinfo

- » Konstruieren eines einfachen Raspberry-Pi-Gehäuses (4, Model B) mit der freien Software *FreeCAD* in 3D
- » Ermitteln der richtigen Maße der Platine
- » Zeichnen eines Gehäuses und Platzieren der richtigen Aussparungen für die Anschlüsse

Checkliste

-  **Zeitaufwand:**
etwa drei Stunden für die Konstruktion
-  **Kosten:**
keine für die Konstruktion, je nach Material für den Druck ab 1 Euro mit dem eigenen Drucker und ab 25 Euro beim Dienstleister
-  **Konstruieren:**
Arbeit mit CAD-Software, aber keine Vorkenntnisse nötig
-  **Computer:**
Desktop oder Notebook mit Windows, macOS oder Linux
-  **3D-Druck:**
mit eigenem Drucker, im Fablab oder beim Dienstleister

Mehr zum Thema

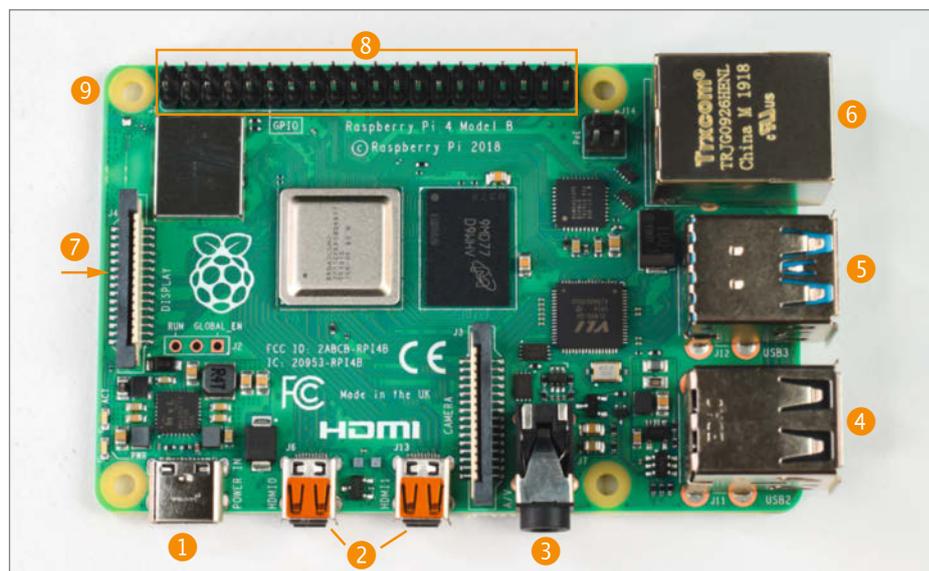
- » Matthias Mett, 3D-Entwurf mit *FreeCAD*, Make 1/21, S. 128
- » Matthias Mett, Stempeln mit *FreeCAD*, Make 4/20, S. 106
- » Matthias Mett, Ring frei für *FreeCAD*, Make 2/20, S. 114
- » Billie Ruben, CAD-Designtipps für den 3D-Druck, Make 1/21, S. 138
- » Peter König, Gratis-CAD für Maker, Make 4/18, S. 19

Alles zum Artikel im Web unter make-magazin.de/xjk7

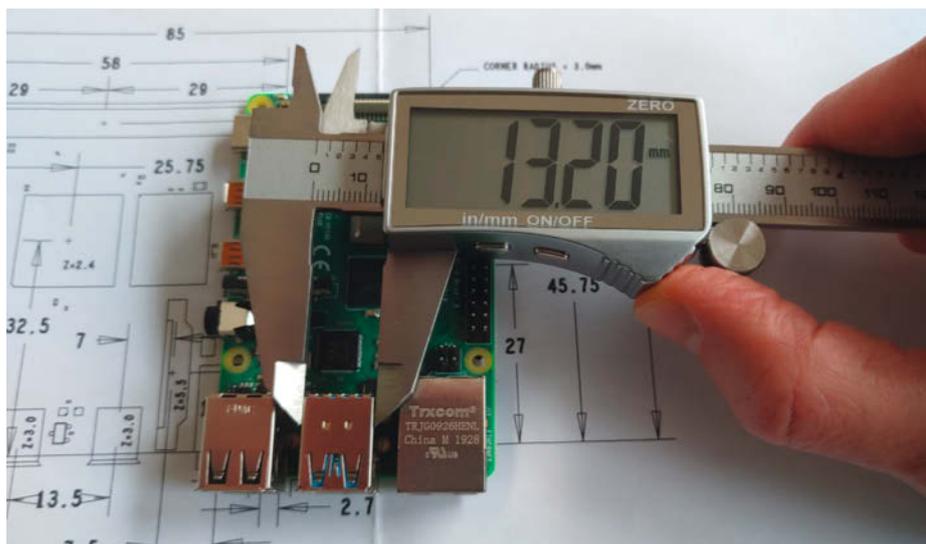
... und die Hülle

Um die Größe des Gehäuses zu bestimmen und die Öffnungen für die Anschlüsse positionieren zu können, benötigen wir die entsprechenden Maße der Platine. Die Raspberry Pi Foundation macht es einem einfach, indem sie auf ihrer Webseite eine technische Zeichnung mit den Maßangaben zum Herunterladen zur Verfügung stellt. Diese können Sie im Bereich *Raspberry Pi 4 Model B specifications*

unter *Raspberry Pi 4 Model B mechanical drawings* finden (siehe Link in der Kurzinfo). Die technische Zeichnung gibt die Größe der Platine, die Positionen der Anschlüsse sowie dessen Bauhöhe an, jedoch nicht deren Breite, die für die Größe der vorzusehenden Ausschnitte allerdings entscheidend ist. Die Breite der einzelnen Bauteile lässt sich aber gut mit einem Messschieber messen ⑩. Die USB-Anschlüsse haben eine Besonderheit, und zwar einen Falz an der Vorderseite. Daher



trickyamir/Shutterstock.com



10 Die Breite der Bauteile lässt sich mit einem Messschieber messen. Im Hintergrund die ausgedruckte technische Zeichnung der Raspberry Pi Foundation.

sollte das Gehäuse relativ eng an der Platine anliegen, sodass die USB-Anschlüsse mit dem Falz aus dem Gehäuse herausragen und die Wände dennoch eine gewisse Stärke aufweisen. Details zur Ermittlung der einzelnen Maße sind Thema des folgenden Artikels der Serie.

Design-Varianten

Mit der Konstruktion des Gehäuses wollen wir es uns möglichst einfach machen: Der Boden soll flach sein und die Seitenwände möglichst gerade. Unnötige Rundungen an den Gehäuse-Ecken und -Kanten wollen wir vermeiden, um die Zeichnung im CAD-Programm nicht unnötig zu verkomplizieren. Außerdem soll das Gehäuse auf dem 3D-Drucker einfach zu drucken sein, ohne zu viel Stützstruktur zu benötigen. Auch aus diesem Grund soll die Konstruktion möglichst viele gerade Flächen haben; speziell die Ober- und Unterseite sind als plane Flächen angelegt, die direkt auf das Bett des 3D-Druckers gedruckt werden können. Wir zeichnen also einen einfachen Kasten, an dem lediglich die Ecken und Kanten eine kleine Rundung haben, damit wir sie nicht mit Feile oder Schleifpapier entgraten müssen.

Aufgrund der Anforderungen der einfachen Konstruktion und der Fertigung im 3D-Drucker haben wir gleich zu Beginn ein paar mögliche Design-Varianten verworfen. So hat das offizielle Gehäuse der Raspberry Pi Foundation (siehe Link) beispielsweise einige Rundungen im Deckel und an den Seiten, die im 3D-Druck viel Stützmaterial (und damit später Entstützaufwand) erfordern. Im Spritzgussverfahren, in dem dieses Gehäuse hergestellt wird, sind solche Designelemente hingegen kein Problem.

Eine weitere Idee war, das Gehäuse aus mehreren dünnen Schichten zusammenzu-

bauen. Neunlagige Acrylgehäuse, im Lasercutter erzeugt, finden sich von verschiedenen Herstellern im Handel, etwa bei *Berrybase* (siehe Link). Bei diesem Design-Ansatz wären jedoch bei neun Lagen genauso viele Zeichnungen in FreeCAD notwendig, bei denen die Öffnungsgrößen in jeder Schicht variieren. Um die einzelnen Schichten bei den Öffnungen zusammenzuhalten, wären wiederum Stützstrukturen um die Stecker herum notwendig, die man durch den inneren Teil führen müsste. Das würde jede einzelne Zeichnung noch einmal sehr verkomplizieren, daher haben wir von dieser Variante ebenfalls abgesehen.

Der Entwurf

Nachdem wir die Maße herausgefunden haben und wir die Position der Anschlüsse kennen, haben wir zur besseren Veranschaulichung eine Designzeichnung erstellt (siehe Titelseite des Artikels). Diesen Aufwand müssen Sie für einen eigenen Entwurf nicht unbedingt betreiben, für den Artikel war dieser Schritt allerdings nötig, damit auch Sie vorab eine bildliche Vorstellung des geplanten Gehäuses bekommen.

Soviel sei schon verraten: Das fertige Gehäuse am Ende entspricht nicht hundertprozentig dem ersten Entwurf, das ist ganz normal. So zeigte sich im Druck, dass die Rundungen an der Unterseite und der Oberseite des Gehäuses Nachteile für die Haftung der Werkstücke am Druckbett hatten, weshalb wir am Ende dann doch auf sie verzichtet haben, und die Lüfteröffnung wurde am Ende auch noch anders gestaltet – dazu aber mehr im weiteren Verlauf der Artikelserie.

In den beiden Außenansichten der Designzeichnung sind sämtliche Aussparungen für die Anschlüsse sichtbar, wobei auf der nicht abgebildeten Längsseite des Gehäuses auch

keine weiteren Aussparungen vorhanden sind. Wir teilen das Gehäuse für den Druck mit einem geraden Schnitt, der auf dem Entwurf als horizontale Linie zu erkennen ist 11. Einerseits lässt sich so später die Platine leichter einbauen, andererseits das Gehäuse leichter drucken: Bei beiden Teilen beginnt der Druck auf dem Druckbett direkt mit der jeweiligen horizontalen Platte für die Ober- 12 und Unterseite 13 des Gehäuses, auf die der Drucker dann jeweils die senkrechten Seitenwände aufbaut. Falls die Teile im Druck keinen hundertprozentig sauberen Abschluss an den oberen schmalen Kanten der Seitenwände bekommen und dadurch Unebenheiten oder ein Spalt beim Zusammenbau der Gehäusehälften entstehen, kann man dies relativ einfach mit Schleifpapier begradigen.

Zusätzlich wollen wir einen kleinen Lüfter im Gehäuse unterbringen. Auf der Oberseite des Gehäuses planen wir daher Lüftungsschlitze ein 14. Zur Befestigung des Lüfters sind Bohrungen neben den Schlitzen vorgesehen. Den Lüfter wollen wir von innen anbringen und die Schrauben durchstecken, um sie mit Muttern auf der Innenseite zu befestigen. So halten wir die Befestigung einfach und benötigen keine speziellen Konstruktionen, um den Lüfter direkt im Gehäuse befestigen zu können. Das erspart noch einmal jede Menge Konstruktionsaufwand.

Platinenmontage

Die Platine soll mit Hilfe von Stützhülsen 15 positioniert werden, was man gut in der Designzeichnung der aufgeklappten Gehäusehälften erkennen kann. Dazu müssen die Hülsen genau an den Positionen der Befestigungslöcher in der Platine sitzen. Zudem muss die Höhe der Hülsen in der unteren Gehäusehälfte um die Platinendicke kleiner sein, damit bei eingesetzter Platine kein Spalt zwischen den Gehäusehälften entsteht.

Zur gleichzeitigen Befestigung der Platine und der Verbindung der Gehäusehälften dienen vier M3-Schrauben. Sie werden später von der Unterseite des Gehäuses eingeschraubt, deshalb sind im Gehäuseboden Senkungen für die Schraubenköpfe und entsprechend große Bohrungen im Unterteil des Gehäuses vorgesehen. In die Hülsen der oberen Gehäusehälfte schneiden wir später mittels Gewindebohrer Muttergewinde, damit die Schrauben gut sitzen, aber beim Einschrauben nicht die Hülsen zerbrechen.

Jetzt sind Sie dran: Bis zur nächsten Folge, in der es um das genaue Ausmessen und das Übertragen der Maße in eine *FreeCAD*-Zeichnung geht, können Sie sich schon mal überlegen, welchem Ihrer Einplatinenrechner Sie ein schickes Gehäuse spendieren wollen, welche Anschlüsse zu berücksichtigen sind und was Sie eventuell sonst noch im Gehäuse unterbringen wollen.

—pek

KI – vom Modell zur Produktion

Exklusiver
Frühbucher-
Rabatt!

 heise Academy

Ob Deep, Supervised, Unsupervised oder Reinforcement Learning, ob NLP oder MLOps – dank unseres Angebot setzen Sie die rasanten Entwicklungen beim Machine Learning in die technische Realität um.

Unser Angebot

**Künstliche
Intelligenz – MLOps:
Machine Learning**

25. – 27. Oktober

WORKSHOP

**Big Data-Analyse
mit PySpark**

4. – 5. November

WORKSHOP

**Automatisierte
Textanalyse mit
Machine Learning**

29. November – 2. Dezember

WORKSHOP

Mehr Informationen:

heise-academy.de/ki_datascience

3D-Druck-Veredlung

Durch das Verzieren von 3D-Drucken können Sie ein Objekt wie aus Sand oder Beton hergestellt erscheinen lassen. So geht's.

von Carsten Wartmann



Auch wenn 3D-Drucke eine ganz eigene Ästhetik haben, möchte man mitunter die Oberfläche bearbeiten, etwa um die Plastikoptik zu kaschieren. Schleifen ist natürlich immer möglich, je nach Objekt aber sehr aufwändig. Alternativ kommt auch ein Farbauftrag infrage. Im Bastel- und Künstlerbedarf gibt es auch Effekt-Farben und Sprühlacke, die eine metallische oder gar eine raue, wie Beton wirkende Oberfläche versprechen. Oft sind diese Farben aber teuer bei nur mäßigem Effekt.

Für die Maker Faire Berlin habe ich eine Möglichkeit gesucht, meine astronomischen Anschauungsmodelle etwas zu verschönern. Konkret ging es hier um die Drucke von Mond- und Marskratern. Nach ein paar Experimenten erzielte ich ansprechende Ergebnisse. Eines der Objekte war der Marskrater *Gale* ¹ im Maßstab 1:1.925.000. Zum Vergleich: Der Rover *Curiosity* wäre in diesem Maßstab etwa 2µm groß, was der Größe einer *E. Coli*-Darmbakterie entspricht.

Material und Verfahren

Gedruckt wurde in PLA, darauf haften die, hier als Kleber verwendeten, Spraylacke gut und das Filament ist einfach zu drucken. PETG, welches sonst mein Lieblingsfilament ist, lässt sich leider nur schlecht ohne Grundierung lackieren und ABS drucke ich wegen des *Warplings* und der Geruchsbelastung nur ungern. Je nach Art der Beschichtung sollten die Drucke entweder in neutralen oder zur Beschichtung passenden Farben gedruckt werden, damit nicht komplett deckende Stellen nicht so auffallen. Ein Druck aus quietschgrünem Filament wird nur sehr schwer mit hellem Beschichtungsmaterial zu verbergen sein.

Als Material zum Beschichten bzw. Bestäuben bieten sich nicht zu grobe, rieselfähige und trockene Materialien an:

Kurzinfo

- » Oberflächen von 3D-Drucken verschönern
- » mögliche Beschichtungstoffe
- » Tipps und Tricks zu Materialien

Checkliste



Zeitaufwand:
1 Stunde



Kosten:
ab 5 Euro (je nach Lack)

Material

- » Lack aus der Dose oder Spraydose
- » Kreppband zum Abkleben
- » staubförmiges Beschichtungsmaterial
Sand, Mörtel, Staub ...

Werkzeug

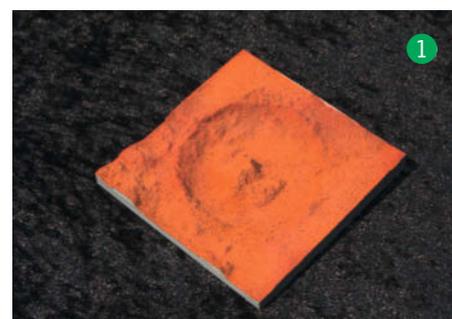
- » Schlagbohrmaschine
- » kleines Küchensieb
- » Pinsel/Bürste oder Druckluft (Blasebalg oder Kompressor)
- » ein paar Gefäße und Behälter

Mehr zum Thema

- » Carsten Wartmann, Betonfiguren aus der 3D-Druckform, *Make* 4/21, S. 88
- » Gerd Michaelis und Stella Maria Risch, 3D-Drucke clever kleben, *Make* 6/20, S. 114
- » Sebastian Müller, Schöner leben – mit 3D-Druck, *Make* 1/20, S. 108
- » Peter König, 3D-Drucker für Maker, *Make* 5/19, S. 18
- » Heinz Behling, 30 Filamente für jeden Zweck, *Make* 2/19, S. 8

Alles zum Artikel im Web unter make-magazin.de/xyv9

- Sand (Quarzsand, Spielsand, Vogelsand)
- Gips, Zement
- Salz, Zucker, Pfeffer (sollte mit Klarlack versiegelt werden)
- Steinstäube, mittels Schlagbohrmaschine aus Backsteinen, Beton oder Natursteinen gewonnen
- Metallspäne oder -Pulver
- Pigmente aus dem Künstler- oder Bastelbedarf, Glitter
- ... alles andere, was Ihnen einfällt





Es kommt natürlich auf den gewünschten Look und Detailerhaltungsgrad des 3D-Drucks an, als Faustregel sollten aber möglichst feine Materialien benutzt werden. Die besagte Schlagbohrmaschine produziert wirklich feinen Staub. Wer Zugang zu einer Kugelmühle oder einem ähnlichem Gerät hat, kann natürlich auch aus vielen anderen Materialien Staub machen oder vorhandenes grobkörniges Material fein mahlen. Mit Glas-, Diamanten- oder Goldstaub habe ich noch nicht gearbeitet, aber Bilder von so behandelten Objekten würde ich gern sehen.

Generell sollte beim Sprayen und beim Arbeiten mit Stäuben eine Staubmaske getragen werden, auch eine Schutzbrille und gute Belüftung sind wichtig. Bei Zement ist darauf zu achten, dass der eine ätzende Wirkung zusammen mit Feuchtigkeit entwickelt. Dies ist bei der Verarbeitung (Handschuhe tragen!), aber auch bei der Präsentation der fertigen Objekte zu beachten — auf einer Maker Faire sollten so bestäubte Objekte nur hinter Glas gezeigt werden.

Besonders bei reinem Zement, der eine sehr feine, betonartige Oberfläche ergibt, kann es passieren, dass er partout nicht auf dem Lack haften will. Dies hat wohl etwas mit

der Polarität von Zement gegenüber dem Lösungsmittel im Lack zu tun. In diesem Fall kann man eventuell mit wasserlöslichen Lacken oder anderen Klebern experimentieren.

Lack auftragen

Im Falle der Krater ② wurden die Seiten mit Kreppband abgeklebt, damit dort weiterhin der Druck erkennbar bleibt und sich ein schöner Kontrast zur beschichteten Oberfläche ergibt. Bei Objekten, die eher organische Formen haben, ist es hilfreich, einen Griff ③, z. B. einen Holzstab, am Objekt zu befestigen (mit Kitt oder Heißkleber), damit das Objekt gut deckend eingesprüht werden kann. Zur Not kann man das Objekt auch hinstellen und in mehreren Durchgängen besprühen und bestäuben, dies ist aber schwieriger und wird auch nicht so gleichmäßig.

Der Farbauftrag sollte bei nicht zu hohen Temperaturen erfolgen, da ansonsten der Lack zu schnell trocknet und nicht mehr gut klebt. Versuche bei 30°C ließen nur wenige Sekunden zwischen Farbauftrag und erfolgreichem Bestäuben der Drucke. Im Gegensatz zum normalen Lackieren, bei dem man lieber mehrere dünne Schichten aufbringt, darf ruhig etwas mehr Farbe benutzt werden. Je dicker die Farbschicht, desto mehr verschwinden sowohl die Drucklayer als auch die Details des Objekts. Hier gilt es, durch Übung und Versuche das richtige Maß zu finden. Wenn in mehreren dünnen Schichten lackiert und bestäubt werden muss, dann ist nach der ersten Trocknung möglichst viel von dem nicht haftenden Staub zu entfernen (abklopfen, Druckluft, Blasebalg, ggfs. vorsichtig pinseln), damit beim weiteren Farbauftrag keine Klumpen entstehen. Einfacher ist es auf jeden Fall, alles in einem Durchgang zu beschichten, was aber bei großen Drucken schwierig sein kann.

Bestäuben

Nach dem Auftragen der Farbe muss es schnell gehen: Kleine Objekte kann man direkt in einen

Behälter mit dem für die Bestäubung vorgesehene Material werfen und dann schütteln. Bei den Kratern wurde einfach der Staub in die durch das Klebeband entstandene Mulde geschüttet und dann durch Schütteln gut verteilt. Für organische Objekte hat sich ein Teesieb ④ als gut geeignet erwiesen, um die Beschichtung gleichmäßig und flächig aufzubringen.

Vor dem weiteren Behandeln oder Säubern muss das Objekt gut trocknen. Dann kann mittels Pinsel, Druckluft oder Blasebalg überschüssiges Material entfernt werden, so dass hoffentlich eine gut deckende Schicht aus Material übrig bleibt, ansonsten versucht man noch eine Schicht aufzubringen. Für eine höhere Abriebfestigkeit kann man dann noch Klarlack auftragen, am besten in einer matten Ausführung. Diese Versiegelung ist besonders bei Ausstellungsstücken, die auch angefasst werden, zu empfehlen und verhindert auch einen Abrieb der Beschichtung.

Ideenreich

Experimentiert habe ich auch mit Sprühkleber, der für das Problem mit dem Zement eine Lösung war. Vieles andere habe ich hingegen noch nicht ausprobiert, womit sich ebenfalls Experimente lohnen würden, etwa Lacke und Farben aus der Dose, die mit dem Pinsel aufzutragen sind. Auch wasserlösliche Lacke oder Holzleim sind interessant. Bindet Zement oder Gips ab, wenn man ihn nach dem Auftragen vorsichtig nass macht? Kann man wie ein Sandkünstler verschiedene Sand- oder Staubarten kombinieren, um Effekte zu erreichen? Sind andere Klebstoffe wie Tapetenkleister oder gar Cyanacrylat (Superkleber) brauchbar? Was passiert mit Metallpulver, wenn man es mit gut haftendem Kleber aufträgt und dann poliert oder mit einer Patina versieht? Rosten Eisenspäne, wenn sie so auf ein Objekt aufgebracht wurden? Sie haben sicher noch Hunderte Ideen und Vorschläge – gerne würden wir davon hören und vor allem freuen wir uns über Fotos Ihrer durch Bestäubung oder anderswie veredelter 3D-Drucke. —caw

Spezialfilamente

Spezielle Druckmaterialien mit Zuschlagstoffen wie Metall-, Holz- oder Gipspartikeln versprechen Drucke, die dann wie Metall, Holz oder Gips aussehen. Je nach Material muss man hier aber mit erheblichen Kosten und einem schwierigen Handling der Filamente rechnen. Zudem wirken etwa die metallhaltigen Filamente erst dann wirklich gut, wenn man sie intensiv nachbearbeitet (schleifen, polieren). Bronzeffill etwa sieht nach dem Druck erst mal wie stumpfer Ton aus.



**WIR MACHEN
KEINE WERBUNG.
WIR MACHEN EUCH
EIN ANGEBOT.**

ct

ct.de/angebot

Jetzt gleich bestellen:

 ct.de/angebot

 +49 541/80 009 120

 leserservice@heise.de

ICH KAUF MIR DIE c't NICHT. ICH ABONNIER SIE.

Ich möchte c't 3 Monate lang mit 35 % Neukunden-Rabatt testen.
Ich lese 6 Ausgaben als Heft oder digital in der App, als PDF oder direkt im Browser.

**Als Willkommensgeschenk erhalte ich eine Prämie nach Wahl,
z. B. einen RC-Quadrocopter.**

© Copyright by Maker Media GmbH.



Klaviertastatur für ESP32



Der ESP32 ist durchaus geeignet, Töne und somit auch Musik zu produzieren. Eine Klaviatur direkt daran anzuschließen erweist sich aber als aufwändig, denn die Anzahl der Eingangspins ist gering. Statt Direktanschluss wäre da die Verwendung eines Datenbusses wie I²C besser geeignet. Das offizielle Klavier-Board von M5Stack bietet genau das: 25 Tasten in klavierähnlicher Form. Das reicht mit der geeigneten (selbst zu schreibenden) Software für immerhin zwei Oktaven Tonumfang. Und das bei nur vier benötigten Anschlusspins, wovon zwei für die Versorgungsspannung und Masse verwendet werden.

Zusätzlich bietet das Tastatur-Board noch je eine RGB-LED pro Taste sowie zwei Touch-Sensoren, die frei programmiert werden können. Ursprünglich ist die Tastatur für den direkten Anschluss an das M5Stack-Core-Modul gebaut. Dafür besitzt es eine Steckleiste, auf die das Core-Modul paßt. Mit einigen Jumperkabeln ist aber auch der Anschluss an jedes andere ESP32-Board möglich. Dann aber muss man sich zusätzlich um die Ansteuerung eines Verstärkers nebst Lautsprecher kümmern, die das Core-Modul praktischerweise gleich mitbringt.

Die Tastatur hat die Maße 240mm × 53mm × 8mm. Die Breite der Tasten liegt damit deutlich unter der bei einem echten Klavier üblichen, zumal bei der Länge des Keyboards noch der Platz wegfällt, der für den M5Stack-Core vorgesehen ist. Die I²C-Adressen sind 0x6a und 0x7a. Direkt in China bestellt kostet die Tastatur etwa 20 Euro (inkl. Versand), bei tinytronics.nl bekommt man sie für 15 Euro (zzgl. Versand, siehe Link). —hgb

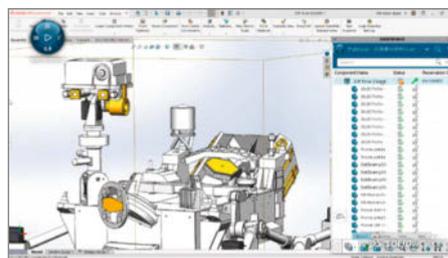
Hersteller	M5Stack
URL	make-magazin.de/xdnk
Preis	ca. 15 € (inkl. Versand)

Solidworks for Makers

Profi-CAD im Abo für nicht kommerziellen Einsatz

Dassault Systemes bietet Anwendern, die private Projekte konstruieren wollen und damit maximal 2000 US-Dollar pro Jahr verdienen, ein spezielles Jahresabo für den CAD-Werkzeugkasten *3DEXperience Solidworks* an. Für 99 US-Dollar plus Steuern bekommt man dabei Zugriff auf das parametrische Konstruktionswerkzeug *3D Creator* und den Freiformmodellierer *3D Sculptor*, wobei man während der Konstruktion zwischen beiden Werkzeugen mit einem einzigen Tastendruck wechseln können soll. Beides läuft dabei online im Browser; gespeichert werden die Daten in der Cloud des Anbieters. Das ähnelt sehr dem Konkurrenzangebot *Autodesk Fusion 360*, das für nicht kommerziellen Einsatz zwar noch kostenlos, aber in der Funktion mittlerweile eingeschränkt ist. Wie bei *Fusion 360* installiert man sich auch bei *Solidworks for Makers* zusätzlich einen lokalen Client auf dem eigenen Rechner (Windows 10 vorausgesetzt).

Der Funktionsumfang des Angebots soll laut Hersteller dem einer klassischen Profi-Version von Solidworks in nichts nachstehen, allerdings lässt sich die Maker-Ausgabe weder durch Add-ins erweitern noch auf demselben Rechner nutzen wie eine lokale Solidworks-Installation. Und auch die Datei-



en in den hauseigenen Formaten enthalten Wasserzeichen, sodass sie nicht zwischen Makern und Profis ausgetauscht werden können – es sei denn, man wählt den Umweg über neutrale Austauschformate wie STP oder IGES. Dem Cloud-Charakter des Angebots entsprechend verspricht *Dassault Systemes* den Abonnenten auch gleich eine eigene Online-Community zum Austausch mit anderen nicht kommerziell interessierten Solidworks-Nutzern. —pek

Hersteller	Dassault Systemes
URL	www.3ds.com/products-services/solidworks/disciplines/makers
Preis	99 US-\$/Jahr (zzg. Steuer)

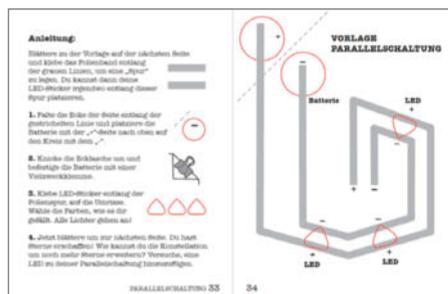
Chibitronics

Deutsche Anleitung für die LED-Sticker

Auch auf Papier lässt sich mit Elektronik basteln – mit LED-Stickern und Kupferklebeband bietet die US-Firma Chibitronics dafür die passenden Komponenten. Seit kurzem gibt es nun endlich die Anleitung auf Deutsch übersetzt und kostenlos herunterzuladen, die wir bei unserem ersten Test vor ein paar Jahren noch vermisst haben.

Das Handbuch steht auch im kleinen Booklet-Format zur Verfügung, mit dem es papiersparend auf 12 statt 90 Seiten ausgedruckt werden kann. In fünf Kapiteln werden in dem Buch Grundlagen der Elektronik und erste Schaltungen vorgestellt. Neben der Theorie gibt es immer direkt Platz, um das Gelernte in vorgedruckten Schaltkreisen auszuprobieren und eigene Ideen zu entwickeln.

Für Lehrkräfte gibt es die Schaltkreis-Seiten außerdem im Klassensatz mit vier Vorlagen pro Seite, die einfach vervielfältigt werden können. Da das Handbuch unter der *Creative-Commons-Lizenz BY-SA-4.0* steht, darf es bei Namensnennung und Weitergabe unter den gleichen Bedingungen verändert und verbreitet werden.



Die Anleitungen für den Mikrocontroller *Love to Code* sind aber leider noch nicht übersetzt. Dieses Open-Source-Board wurde von der Firma speziell für den Einsatz mit Kupferklebeband und Papier entwickelt. Es ermöglicht die gezielte Steuerung der LEDs über einfache Schalter hinaus und richtet sich ebenfalls an Schulen: Zur Programmierung reicht bereits ein Smartphone mit Audioanschluss. —hch

Hersteller	Chibitronics
URL	https://chibitronics.com/templates/
Preis	kostenlos

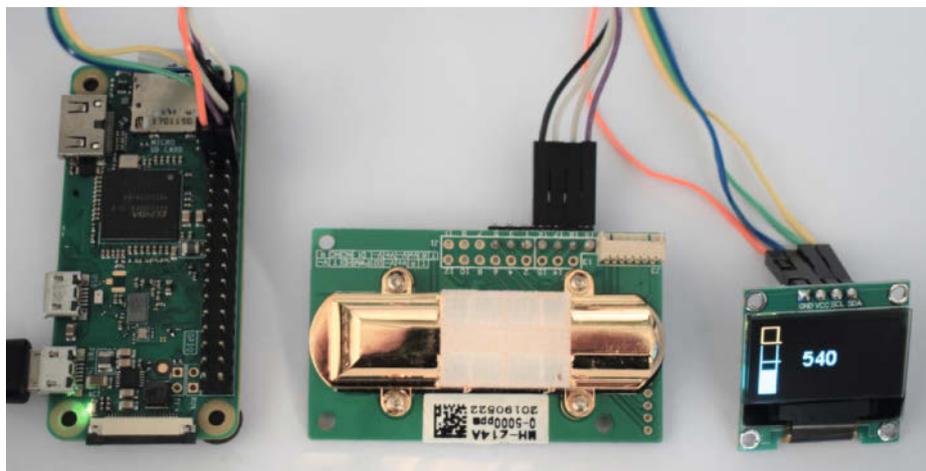
LabPi

Raspberry-Pi-basierte Messstation für den MINT-Unterricht

Hinter der Messstation *LabPi* verbirgt sich ein modulares System aus Hard- und Software der Uni Jena, mit dem der Einsatz von verschiedenen Umweltsensoren im Unterricht einfach möglich sein soll. Über eine Adapterplatine können die Sensoren mit dem Einplatinenrechner Raspberry Pi verbunden werden. Die LabPi-Software ermöglicht die Datenauswertung und Vernetzung der Messstationen.

Bereits unterstützt werden Experimente mit Sensoren zur Messung von Temperatur, CO₂-Anteil der Luft, pH-Wert und Leitfähigkeit sowie Photometer. Später sollen noch Module rund um GPS-Positionsbestimmung, Feuchtigkeitsmessung, Spektrometer, Colorimeter und weitere Sensoren hinzukommen. Die LabPi-Software bringt die jeweils nötigen Treiber mit und ermöglicht, die Messwerte direkt auf einer grafischen Oberfläche anzusehen und auszuwerten. Neben der lokalen Installation gibt es außerdem die Cloud-Plattform *COMPare*, über die Ergebnisse mehrerer Stationen gespeichert und verglichen werden können.

Bastelfreudige Lehrkräfte können die Hardware einzeln besorgen und selbst zusammenstecken. Für den Eigenbau sind ein



Raspberry Pi 3B+ oder anderer Einplatinenrechner, ein Display und Raspi-kompatible Sensoren nötig – die Software kann kostenlos heruntergeladen werden. Um Kabelchaos beim Anstecken mehrerer Sensoren an die Pin-Leiste des Raspi zu verhindern, hat das LabPi-Team eine Adapterplatine entwickelt. Für alle, die keine Zeit zum Lötten haben, gibt es fertige Messstationen und die passenden Sensoren im LabPi-Shop. Seit kurzem ist auch

ein Bausatz mit Raspi Zero W, OLED-Display und CO₂-Sensor als CO₂-Ampel erhältlich (siehe Bild).
—hch

Hersteller	Friedrich-Schiller-Universität Jena
URL	labpi.de/shop.php
Preis	Software kostenlos, CO ₂ -Ampel-Kit 137 €, Messstation 395 €

EPH-40

Entlötpumpe mit integriertem LötKolben

Dass beim Basteln etwas falsch herum angelötet wird oder defekte Teile ausgetauscht werden müssen, kommt häufig vor. Das Entlöten stellt sich dann oft als Geduldsprobe heraus. Die gängigen Methoden haben ihre Haken: Während bei einer einfachen mechanischen Entlötpumpe zur optimalen Nutzung mehr als zwei Hände benötigt werden, bekommt man mit Entlötlitze nicht immer alles sauber aus den Lötlöchern heraus. Eine automatisierte Maschine kommt aufgrund des hohen Preises für viele Hobbybastler nicht in Frage. Der Hersteller der Entlötpumpe *EPH-40* verspricht hingegen sauberes

und leichtes Entfernen von Lötzinn und somit eine Lösung für die genannten Probleme: Dieses Entlötgerät besteht aus einem LötKolben

mit hohler Spitze und einer im Griff integrierten Entlötpumpe, die durch die Bohrung in der Spitze das geschmolzene Lötzinn ansaugen soll.

Da kein Schalter vorhanden ist, muss zum Starten das 230V Netzkabel eingesteckt werden, nach circa 10 Minuten ist der Kolben aufgeheizt. Durch einfaches Runterziehen des Kolbens entsteht ein Vakuum, welches durch einen Knopfdruck am Kolben gelöst wird und das Lötzinn in die Pumpe zieht, wie von den einfachen Entlötpumpen gewohnt.

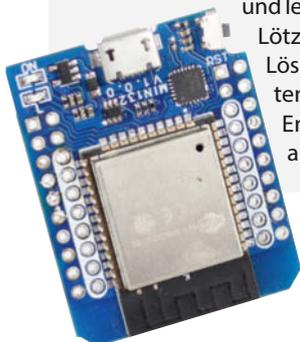
Wir haben das Entlötgerät an einem ESP32-Board mit falsch herum angelöteten Buchsenleisten getestet, das Ergebnis sieht man im Bild. Nach wenigen Versuchen war die Bedienung der Entlötpumpe intuitiv und leicht. Es hat anschließend nur ein paar



Minuten gedauert, die falschen Teile zu entlöten, die längste Zeit haben wir mit Warten auf das Aufheizen des Kolbens verbracht. Aufgrund der integrierten Pumpe wird nur eine Hand benötigt, was Verrenkungen erspart. Zudem saugt die Pumpe das Lötzinn sauber und schnell auf, so dass die Lötlöcher komplett frei werden. Damit haben wir Bauteile im Wert von 10 Euro gerettet, womit sich das kleine Gerät, das selbst deutlich weniger kostet, bereits beim ersten Einsatz bezahlt gemacht hat.
—AnnaEichler/hgb

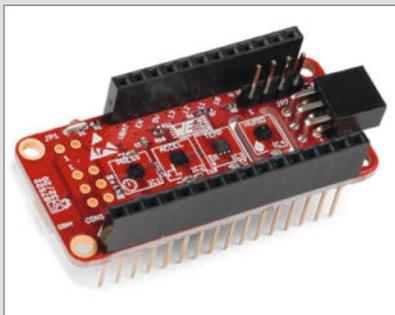
Hersteller	McPower
Vertrieb	etwa berrybase.de
Preis	5,90 €

Ausprobiert
— von Make! —



FeatherWings

Zu Adafruits Feather-Board compatible Sensor- und WiFi-Boards



Der Hersteller Würth ist vielen vielleicht nur durch sein Schraubensortiment bekannt, bietet aber seit vielen Jahren als *Würth Elektronik (WE)* auch hochwertige elektronische Bauteile und eine Platinenfertigung an; vielen Makern dürften zum Beispiel die Induktivitäten von WE ein Begriff sein. Seit einiger Zeit sind auch Halbleiterbauteile im Programm, darunter vor allem Sensoren sowie Funkmodule.

Passend zum kompakten *Feather*-Formfaktor von Adafruit, für das Adafruit eine ganze Reihe von Basis-Boards mit verschiedenen CPUs anbietet, liefert WE nun compatible Sensor- und Funkmodule mit Bauteilen aus dem eigenen Sortiment, so genannte *FeatherWings*. Ganz neu ist zum Beispiel ein Sensor-Feather-Wing mit gleich vier WE-Sensorfunktionen für Temperatur, Absolutdruck, Feuchtigkeit und 3-Achsen-Beschleunigung. Alle vier Sensoren sind über den gemeinsamen I²C-Bus verbunden, das Board ist außerdem mit dem QWIIC-Verbindungssystem von Sparkfun kompatibel.

Für seine FeatherWings hat Würth ein eigenes Github-Repo eingerichtet. Das Angebot richtet sich mit seinen Preisen allerdings eher an Entwickler aus der Industrie als an sparsame Maker. —cm

Hersteller	Würth Elektronik
URL	www.we-online.com
Preis	ab 60 € (Sensor-Board)

XIAO RP2040

Pico-Prozessor-Board aus China

Der Seeeduino XIAO RP2040 ist ein Mikrocontroller-Board, das den Raspberry-Pico-SoC RP2040 verwendet. Er läuft mit bis zu 133MHz, verfügt über vielfältige Schnittstellen in einem winzigen Format und unterstützt neben *MicroPython* auch *CircuitPython*. Die Onboard-Schnittstellen reichen für die Entwicklung zahlreicher Anwendungen aus, für erste Schritte gibt es auch schon eine Trägerplatine mit Display und auf Steckleisten herausgeführten Anschlüssen. Ansonsten ist das nur 20mm × 17,5mm kleine Board aber auch Steckbrett-kompatibel.

Elf I/O-Pins stehen zur Verfügung, davon lassen sich vier als Analogeingänge schalten. Alle I/Os beherrschen PWM. Für weitere Peripherie können die Anschlüsse auch als I²C, UART und SPI konfiguriert werden. Der Zweikerner ARM Cortex M0+ hat Zugriff auf 264KByte internes RAM und 2MByte Flash-Speicher. Für die Programmentwicklung



stehen *MicroPython*, die Arduino-IDE und *CircuitPython* bereit, zum Flashen dient die USB-3-C-Buchse. —cm

Hersteller	Seeedstudio
URL	www.seeedstudio.com/XIAO-RP2040-v1-0-p-5026.html
Preis	5,40 US-\$

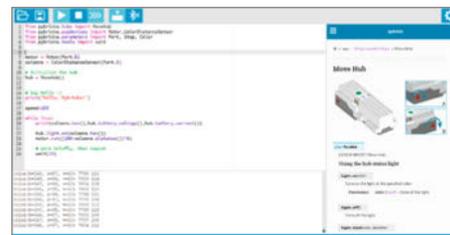
Pybricks

MicroPython für Lego-Robotik-Sets

Die Open-Source-Software *Pybricks*, basierend auf *MicroPython*, wurde bisher für die meisten Bluetooth-fähigen Controller („Powered Up“) wie *Lego Boost*, *City*, *Technic* sowie *Mindstorms* und *Spike* portiert. Dabei werden nicht nur die Controller, sondern auch die vielfältigen Sensoren und Aktuatoren unterstützt. Besonders profitieren hier die kostengünstigen *Boost*-, *City*- und *Technic*-Bricks, die bisher nur über eine App auf dem Smartphone oder Tablet mittels grafischer Programmierung gesteuert werden konnten. *Pybricks* läuft dagegen autonom auf dem Brick und das bis zu hundertmal schneller.

Bei *Mindstorms*- und *Spike*-Bricks war es zwar schon immer möglich, diese autonom agieren zu lassen, durch die gemeinsame API wird nun aber eine Portierung der Steuerungen von einem zum anderen Brick viel leichter. Die Web-IDE läuft unter Chrome (Windows, Mac, Linux, Chromebook und Android) und greift per Bluetooth auf die Controller zu. Aus der Webumgebung ist es auch möglich, die *Pybricks*-Firmware aufzuspielen. Die originale Lego-Firmware kann wieder per Lego-App aufgespielt werden, falls man selbst (oder die Kids) doch dahin zurückmöchten.

Auf den kleineren Controllern, wie dem *Boost-Hub*, ist *Pybricks*-*MicroPython* etwas



eingeschränkt. So werden z.B. keine Fließkommazahlen unterstützt. Auf dem Hub ist auch kein Platz, um ein Programm zu speichern, daher muss das Programm zusammen mit der Firmware auf den Controller geladen werden, was ein paar Minuten dauert. Solange man aber per Bluetooth mit dem Controller verbunden ist, kann man sein Programm in Sekundenschnelle testen und die Ausgaben auf der REPL-Schnittstelle lesen. In der IDE hat man direkten Zugriff zu der gut gemachten Dokumentation, die auch viele Beispiele liefert. Alles in allem eine sehr willkommene Entwicklung, die sich besonders für *Lego Boost* lohnt. —caw

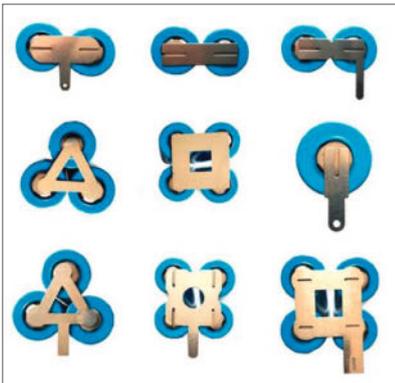
URL	code.pybricks.com
Preis	kostenlos

Ausprobiert
— von Make: —

Schweißgerät für Akkuzellen

Notebook-Akkus oder Akkupacks im Modellbau und viele anderen Geräten sind zwar meist aus Standardzellen etwa vom Typ 18650 aufgebaut. Trotzdem wehren sie sich meist gegen eine Reparatur, weil die Kontaktfahnen zwischen den Einzelzellen verschweißt werden müssen. Das macht den Austausch schwierig.

Diese kleine Punkt-Schweißanlage macht das einfacher: Die beiden aus dickem Kupfer gefertigten Elektroden müssen nur in geringem Abstand auf die Nickel-Kontaktstreifen der Akkuzellen aufgesetzt werden. Eine kurze Betätigung des Fußschalters lässt dann einen Strom von bis zu gut 600 Ampere bei nur wenigen Volt Spannung fließen. Das reicht, um die Kontaktstreifen so stark



zu erhitzen, dass sie an den Berührungsstellen der Elektroden miteinander verschweißen. Geliefert wird dieser in neun Stufen einstellbare Strom aus Kondensatoren im Inneren der kleinen Maschine. Das Gerät selbst wird mit einem Steckernetzteil versorgt. Alternativ gibt es auch eine netzteillose Ausführung, die mittels USB gespeist werden kann. Übrigens ist die „5000 Watt“-Angabe der Produktbeschreibung Quatsch, die sich nicht mit den sonstigen angegebenen Spezifikationen deckt: Wenn man die einmal durchliert, findet man als (Impuls-)Leistung an den Schweißelektroden 4,2V und 650A, macht 2730Watt. Das geht zwar nur im Millisekunden-Bereich, sollte aber zum Schweißen reichen.

Das Gerät gibt es bei Aliexpress. Dort erhält man auch für Selbstbau-Projek-



te vorgefertigte Kontaktstreifen für die Akkupack-Fertigung. Die verschiedenen Versionen erlauben sowohl die Serien- als auch die Parallelschaltung der Akkuzellen. —hgb

Hersteller	Shenzhen Welding tools Store
URL	de.aliexpress.com/item/1005001971701444.html
Preis	72 €

EDU:BIT

Lernkit, um spielerisch programmieren zu lernen

Für den britischen MikroController *micro:bit* hat der malaysische Hersteller *Cytron* als Erweiterung eine Übungs- und Projektplatine namens *EDU:BIT* entwickelt. In der getesteten Variante ist die Platine Teil eines Lern-Kits: Neben dem Board werden zusätzliche Elektronikteile, Spielzubehör und ein Erklärbuch mitgeliefert; auch ein *micro:bit* selbst ist enthalten.

Zu dessen bereits bekannten Optionen kommen mit der Erweiterungsplatine unter anderem ein Geräuschsensor, ein Infrarot-Sensor, Anschlüsse für einen Servomotor, Drucktasten und Anschlüsse für eine I²C-Schnittstelle hinzu. Schülerinnen und Schülern stehen damit mehr Möglichkeiten als beim *micro:bit* zur Verfügung, ohne dass extra Hardware gekauft werden muss.

Besonders für Anfänger, die noch keine eigenen Ideen haben, eignet sich das bei-

gelegte Booklet. In elf Kapiteln liefert es nach dem Motto „Code, Play, Learn“ Schritt-für-Schritt-Anleitungen, um Kindern auf spielerische Weise einen Einstieg in die Welt des Programmierens zu geben. Jedes Kapitel beinhaltet ein klassisches Spiel zum Nachprogrammieren, etwa *Twister*, *Schere-Stein-Papier* oder das *Leiterspiel*. Die dafür benötigten Materialien sind im Lern-Kit enthalten. Die Anleitungen sind sehr detailliert und kindergerecht geschrieben. Neben verspielten Grafiken werden zahlreiche Screenshots gezeigt, die das Nachprogrammieren vereinfachen.

Software für den *EDU:BIT* programmiert man wie den *micro:bit* wahlweise grafisch, in *Python* oder *JavaScript* im Online-Editor *makecode.org*. Für Anfänger eignet sich der Blockprogrammiermodus. Man zieht die Blöcke einfach per Drag&Drop zum gewünschten Programm zusammen. Fortgeschrittene Benutzer können im *MakeCode*-Editor ganz einfach in den *JavaScript*- oder *Python*-Modus für textbasiertes Programmieren wechseln. Da es vollständig auf einer Web-Umgebung basiert, ist keine Software-Installation auf dem Entwickler-Rechner erforderlich. Das im Browser zusammen-



Ausprobiert
— von Make: —

geklückte oder geschriebene Programm schickt man wie gewohnt über das USB-Kabel auf den *micro:bit* auf dem *EDU:BIT*.

Für Kinder mit eigenen Projektideen besteht die Möglichkeit, einzelne Module auf der *EDU:BIT*-Platine abzutrennen, um das Projekt noch flexibler zu gestalten. Für den Anschluss der abgetrennten Teile an die Hauptplatine werden Grove-Kabel mitgeliefert.

Das Kit ist zwar nur in der englischen Variante erhältlich, das Booklet steht aber auch als Download in deutscher Sprache zur Verfügung. —Anna Eichler/pek

Hersteller	Cytron
Vertrieb	beim Hersteller oder etwa Robotshop.com
Preis	ca. 50 €



Cadus Debate

Humanitäre Hilfe mit Open Tech



Cadus ist eine gemeinnützige Organisation, die medizinische Hilfe in Krisengebiete bringt – zum Beispiel nach Nordost-Syrien und in den Irak. Um für die unterschiedlichen Probleme vor Ort auch technische Lösungen selbst zu entwickeln und umzusetzen, betreibt die Organisation einen Makerspace. Dort entstand der *Life Sensor*, ein Open-Source-Vitalparameter-Monitor, der als EKG fungiert und Sauerstoffsättigung, Puls und Blutdruck misst.

Mit *Cadus Debate* gibt es außerdem eine (meist) monatliche Veranstaltungsreihe, bei der die Arbeit mit Gästen reflektiert und vorgestellt wird. Seit dem Beginn der Pandemie werden die Veranstaltungen gestreamt und sind daher online nachschaubar. Inhaltlich wechseln sich politische und technische Themen ab. So ging es etwa schon einmal um die Frage, wie und ob Sanktionen funktionieren. Deutlich technischer sind dagegen Sessions zu Making in Zeiten von Corona, Open-Source-Software für NGOs und Überlegungen, wie offene Hardware in Post-Konfliktgebieten eingesetzt werden kann. Als Gäste sind zahlreiche Engagierte aus unterschiedlichen Organisationen dabei: von den *Ärzten ohne Grenzen* über den *Chaos Computer Club* bis hin zu *Open Source Design*. —hch

Termin erster Donnerstag im Monat, 20 Uhr
URL cadus.org/de/debate

Tuning für Kinderautos

Eigenbau, Umbau und Motorisierung von Kinderfahrzeugen

Wer ein (kleines) Kind hat, kommt um ein Bobby-Car oder eine der zahllosen Rutschauto-Varianten kaum herum. Wie man aus ihnen noch mehr herausholt oder ein eigenes Kinderauto baut, erklärt der Bastler Tom Masselter in diesem Buch. Garniert mit vielen bunten Fotos zeigt er Projekte aus seiner langjährigen Bastelkarriere, vom 10-Euro-Auto bis zur motorisierten und fernsteuerbaren Planierraupe. Doch zunächst geht es im Buch um die grundlegenden Kenntnisse: Welches Werkzeug ist nötig, welche Bauteile lassen sich einsetzen und wie wird daraus ein Mitmachprojekt – damit die zukünftigen Fahrerinnen und Fahrer mitbasteln können?

Die Projekte sind dann nach ihrem steigendem Schwierigkeitsgrad, aber auch steigendem Spaßfaktor angeordnet. Los geht es mit

einfachen Tuning-Ideen, um geschmeidiger laufende Räder einzubauen oder die Achsen zu verstärken. Am Beispiel eines Akkuschauber-getriebenen Bobby-Cars wird die Motorisierung genauer erklärt. Weitere Modelle mit Elektro- oder sogar Benzinmotor werden aus Platzgründen nur angerissen, um erfahrenen Bastlern weitere Anregung zu geben. Einzig der Einsatz von Beleuchtung ist uns etwas zu kurz geraten, dabei begeistern Lichter ebenso wie Geschwindigkeit und leisten sogar noch einen Beitrag zur Sicherheit. —hch



Autor Tom Masselter
Verlag VTH
Umfang 168 Seiten
ISBN 978-3-88180-511-7
Preis 19,90 €

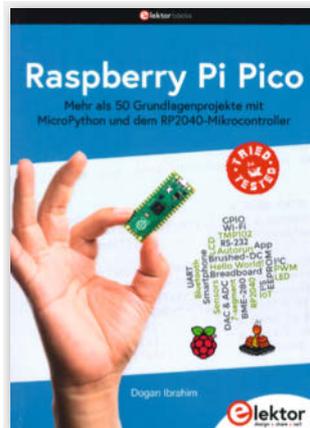
Raspberry Pi Pico

Mehr als 50 Grundlagenprojekte mit MicroPython und dem RP2040-Mikrocontroller

Dieses Buch will kein Lehrbuch für Einsteiger in die Programmierung oder die Hardware von Mikrocontrollern sein, so sagt es der Autor im Text. Die genaue Beschreibung der Projekte macht es aber dennoch auch dazu: Von einfachen, nur rechnenden Programmen bis hin zur Verwendung von Sensoren und LC-Displays am I²C-Bus erklärt es langsam im Schwierigkeitsgrad steigend Hardware und Software in nicht allzu technischer Sprache.

Lediglich zu Beginn, bei der Vorstellung des Raspberry Pi Pico und anderer Boards mit demselben Chip, wird zumindest die Kenntnis einiger Fachbegriffe vorausgesetzt. Das regt entweder zum Nachschlagen an oder man lässt dieses Kapitel aus, denn für die folgenden Projektbeschreibungen braucht man es nicht unbedingt.

Die Projekte sind gut bebildert – nicht nur mit Schaltplänen, sondern auch Realbildern der Bauteile und den Anschlussbelegungen. Alle Listings werden erklärt, wodurch man die Programmiersprache *MicroPython* ganz nebenbei lernt. Insgesamt ist dieses Buch ein guter Einstieg in die Anwendungsmöglichkeiten des relativ neuen, kleinen Rasperrys – ob es nun Ihr erster Mikrocontroller ist oder Sie schon Erfahrungen mit anderen haben, ist dabei nicht von Belang. —hgb



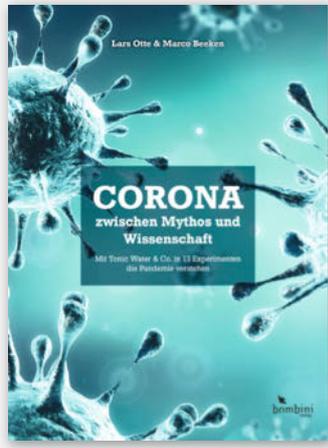
Autor Dogan Ibrahim
Verlag Elektor
Umfang 256 Seiten
ISBN 978-3-89576-456-1
Preis 34,80 € (Print), 29,99 € (PDF)

Corona zwischen Mythos und Wissenschaft

Mit Tonic Water & Co. in 13 Experimenten die Pandemie verstehen

Kann man sich über Geldscheine mit Corona infizieren? Wie wirkt sich die Lufttemperatur und -feuchtigkeit auf die Aerosol-Ausbreitung aus? Was schützt besser: Seife oder Desinfektionsmittel? Bei der Antwort auf solche Fragen (die sich oft aus verbreiteten Mythen rund um SARS-CoV-2 ergeben) helfen anschaulich die im Untertitel erwähnten 13 Experimente. Da es sich dabei durchweg um Versuche mit Modellsubstanzen handelt (und nicht solche am „lebenden“ Virus), braucht man dazu vor allem Dinge, die es im Haushalt gibt: etwa Marmeladengläser, Eier, eine Sprühflasche und zuckerfreies Tonic Water.

Gerade der Beschreibung der Experimente samt Interpretation der Ergebnisse merkt man



an, dass das Autorenteam dieses Buches aus der (Chemie-)Didaktik kommt. Infoboxen, Experimente, ausführliche, aber lebendig geschriebene erklärende Texte und ein eigenes Quellenverzeichnis für jedes Kapitel machen die stellenweise komplizierte Materie gut zugänglich und vertiefbar. Dass es zwischen den Kapiteln einige Redundanzen gibt, fällt beim kompletten Durchlesen in einem Rutsch etwas auf, erleichtert aber, sich einzelne Abschnitte herauszupicken oder die Experimente in anderer Reihenfolge zu absolvieren, zum Beispiel im Unterricht.

Das Buch hat zwar schon einige Monate auf dem Buckel – die Arbeit am Manuskript haben die Autoren im November 2020 abgeschlossen

– das fällt dennoch nur an wenigen Stellen ins Gewicht, obwohl es seinerzeit noch keine verbreiteten Schnelltests, keinen fertigen Impfstoff und noch nicht so viel Kenntnisse über Mutanten gab. Das allermeiste, was es rund um Corona in diesem Buch zu lesen gibt, ist nach wie vor aktuell. Und mit Themen wie Handhygiene, Schutz durch Mund-Nase-Bedeckungen, Aerosolverbreitung in der Atmosphäre und Umgang mit Zahlen wie R-Wert und *False Positives* werden wir uns wohl auch nach dieser Pandemie sicher irgendwann wieder beschäftigen müssen ...

—pek

Autor	Lars Otte, Marco Beeken
Verlag	Bombini
Umfang	196 Seiten
ISBN	978-3-946496-21-2
Preis	16,95 € (Buch), 12,95 € (E-Book)

Leiterplattendesign mit Eagle

Ein Einstieg mit fünf Praxisprojekten

Eigene Platinen anzufertigen ist mit vielen Schritten verbunden. Am Anfang steht immer ein Schaltplan, aus dem ein Platinenlayout wird. Diverse Hersteller bieten dafür Software an und die Anwenderlager von Eagle, KiCad oder irgendeiner anderen Software verhalten sich ähnlich apologetisch wie die Anhänger von Windows, macOS und Linux – seien Sie also vorbereitet.

Eagle erfreut sich immerhin seit über 30 Jahren einer großen Fangemeinde, auch weil es in einer kostenlosen Version verfügbar ist. Der Autor, Professor an der TH Aschaffenburg, führt seine Studenten an die Software in Vorlesungen heran und schult regelmäßig Anwender in der Industrie. Seine Erfahrungen sind in fünf Projekte im Buch eingeflossen: Binärzähler, Netzteil, USB-Sound-Stick, Class-D-Verstärker und ein Mikrocontroller-Board.

Von der Bauteilerauswahl, dem Schaltplan, der Platzierung der Elemente auf der Platine bis zum Verlegen, Entflechten und Korrigieren der Leiterbahnen wird alles minutiös und mit vielen farbigen Bildern und leicht verständlichen Texten erklärt. Dabei nimmt die Komplexität von einseitigen und doppelseitigen Leiterplatten bis zu vierlagigen lang-

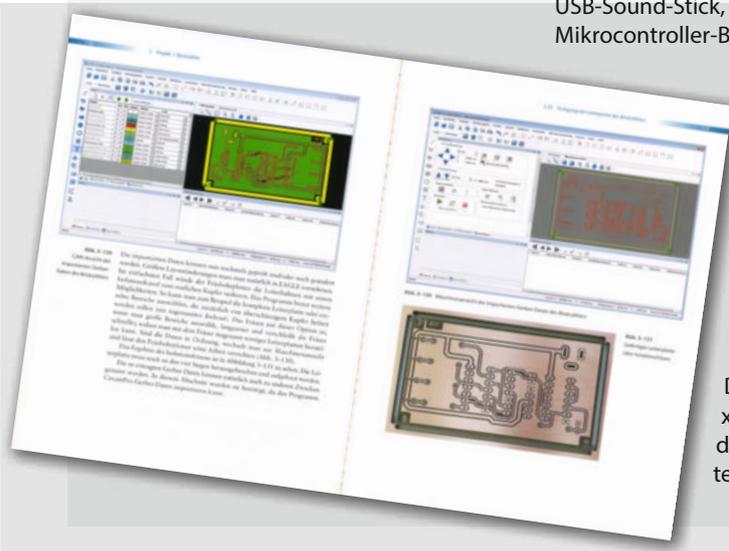


sam zu. Ein Abschnitt des Buches widmet sich sogar dem Isolationsfräsen von Leiterplatten.

Das Buch bietet Makern einen fundierten und sicheren Einstieg in die Kunst des Platinenlayouts.

—dab

Autor	Francesco Volpe
Verlag	dpunkt
Umfang	284 Seiten
ISBN	978-3-86490-545-2
Preis	36,90 € (Buch, 29,99 € (E-Book), 41,90 € (Bundle)



IMPRESSUM

Make: Nächste Ausgabe erscheint am 4. November 2021

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: Heinz Behling (hgb), Helga Hansen (hch),
Rebecca Husemann (rehu), Carsten Meyer (cm),
Elke Schick (esk), Carsten Wartmann (caw)

Mitarbeiter dieser Ausgabe: Beetlebum (Comic), Guido Burger, Kurt Diedrich, Anna Eichler, Stephan Feldker, Michael Gaus, Detlef Grell, Fabian Jansen, Daniel Jedecke, Tim Kamenik, Andreas Koritnik, Miguel Köhnlein, Hartmut Krummrei, Matthias Mett, Uwe Rohne, Marcel Timm, Harald Winkler

Assistenz: Susanne Cölle (suc), Christopher Tränkmann (cht),
Martin Triadan (mat)

Leiterin Produktion: Tine Kreye

DTP-Produktion: Martina Bruns, Martin Kreft (Korrektorat)

Art Direction: Martina Bruns (Junior Art Director)

Layout-Konzept: Martina Bruns

Layout: Nicole Wesche

Fotografie und Titelbild: Andreas Wodrich

Digitale Produktion: Kevin Harte, Pascal Wissner

Hergestellt und produziert mit Xpublisher:
www.xpublisher.com

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

Verlagsleiter: Dr. Alfons Schröder

Stellv. Verlagsleiter: Daniel Bachfeld

Anzeigenleitung: Michael Hanke (-167)
(verantwortlich für den Anzeigenteil),
mediadaten.heise.de/produkte/print/das-magazin-fuer-innovation

Leiter Vertrieb und Marketing: André Lux (-299)

Service Sonderdrucke: Julia Conrades (-156)

Druck: Dierichs Druck + Media GmbH & Co.KG,
Frankfurter Str. 168, 34121 Kassel

Vertrieb Einzelverkauf:

VU Verlagsunion KG
Meißberg 1
20086 Hamburg
Tel.: 040/3019 1800, Fax.: 040/3019 145 1800
E-Mail: info@verlagsunion.de
Internet: www.verlagsunion.de

Einzelpreis: 12,90 €; Österreich 14,20 €; Schweiz 25.80 CHF;
Benelux 15,20 €

Abonnement-Preise: Das Jahresabo (7 Ausgaben) kostet
inkl. Versandkosten: Inland 77,00 €; Österreich 84,70 €;
Schweiz/Europa: 90,65 €; restl. Ausland 95,20 €

Das Make-Plus-Abonnement (inkl. Zugriff auf die App, Heise Magazine sowie das Make-Artikel-Archiv) kostet pro Jahr
6,30 € Aufpreis.

Abo-Service:

Bestellungen, Adressänderungen, Lieferprobleme usw.:

Maker Media GmbH
Leserservice
Postfach 24 69
49014 Osnabrück
E-Mail: leserservice@make-magazin.de
Telefon: 0541/80009-125
Telefax: 0541/80009-122

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. Alle beschriebenen Projekte sind ausschließlich für den privaten, nicht kommerziellen Gebrauch. Maker Media GmbH behält sich alle Nutzungsrechte vor, sofern keine andere Lizenz für Software und Hardware explizit genannt ist.

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.

Published and distributed by Maker Media GmbH under license from Make Community LLC, United States of America. The 'Make' trademark is owned by Make Community LLC Content originally partly published in Make: Magazine and/or on www.makezine.com, ©Make Community LLC 2020 and published under license from Make Community LLC. All rights reserved.

Printed in Germany. Alle Rechte vorbehalten.
Gedruckt auf Recyclingpapier.

© Copyright 2021 by Maker Media GmbH

ISSN 2364-2548

Nachgefragt

Wenn du ein Smart Home mit KI hättest – wie würdest du sie nennen und was zeichnet sie aus?



Fabian Jansen
Hemmingen, gibt uns auf Seite 48 einen Einstieg in die KI-Programmierung mit dem Gervifonn
Ich würde meine KI wie in Star Trek mit „Computer“ ansprechen. Sie wäre schlau genug zu erkennen, wann sie gemeint ist. Ansonsten würde sie unauffällig im Hintergrund arbeiten.



Paola Pollicino
Hannover, schnuppert als Auszubildende in die Arbeit unserer Maker Fairies rein
Mein Smart Home mit KI würde ich „Suri“ nennen. Sie wäre im Alltag eine kleine Helferlin, würde bei schlechtem Wetter Stimmung machende Musik spielen und auch mal eine Gutenachtgeschichte erzählen.



Tim Kamenik
Herrenberg, trickst auf Seite 30 unsere Augen mit dem POV-Effekt auf einem sphärischen Display mit Raspberry Pi aus
Da gibt keine andere Möglichkeit als den Namen „Smartin“. Und ich denke, er ist zwar immer gerne behilflich, aber auch immer etwas genervt.

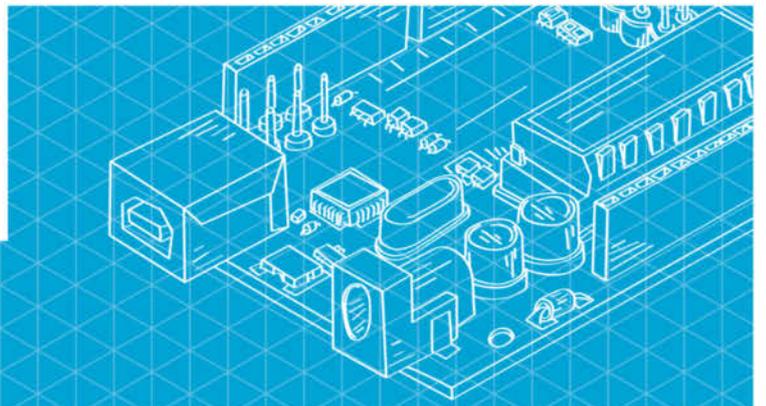


Heinz Behling
Hannover, als Autor unserer Smart-Home-Reihe für diese Frage prädestiniert
Da ich dazu neige, Politiker-Äußerungen wörtlich zu nehmen, würde ich mich auf den Amtseid berufen und den Namen „Merkel“ wählen. Was sie auszeichnet? Reichlich Geduld, um auf die Ereignisse zu warten.

Inserentenverzeichnis

AFAG Messen und Ausstellungen GmbH, Nürnberg	7	Region Hannover, Hannover	13
dpunkt.verlag GmbH, Heidelberg	65, 83	Reichelt Elektronik GmbH & Co., Sande	2
KNIPLEX-Werk – C. Gustav Putsch KG, Wuppertal	37	TUXEDO Computers GmbH, Augsburg	132
Oelmann Elektronik GmbH, Springe	23	Make:markt	53

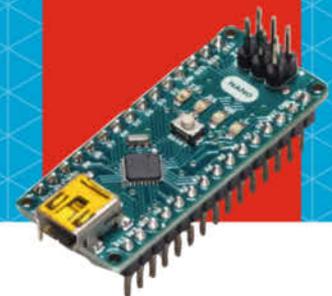
Make:



DAS KANNST DU AUCH!



GRATIS!



2x Make testen und 6 € sparen!

Ihre Vorteile:

- ✓ **GRATIS dazu:** Arduino Nano
- ✓ **NEU:** Jetzt auch im Browser lesen!
- ✓ Zugriff auf Online-Artikel-Archiv*
- ✓ Zusätzlich digital über iOS oder Android lesen

Für nur 15,60 Euro statt 21,80 Euro.

* Für die Laufzeit des Angebotes.

Jetzt bestellen: make-magazin.de/miniabo

© Copyright by Maker Media GmbH.



Maßgeschneiderter Anzug

Wählen Sie die entscheidenden Komponenten selbst



Prozessor
Intel oder AMD CPU



Grafikkarte
Integriert oder dediziert



Arbeitsspeicher
Jederzeit erweiterbar



Speicherplatz
Schnell, groß und erweiterbar



Netzwerk
Drahtlos, Mobilfunk
oder Gigabit LAN



100%
Linux

5

Jahre
Garantie



Lifetime
Support



Gefertigt in
Deutschland



Deutscher
Datenschutz



Support
vor Ort

TUXEDO COMPUTERS

tuxedocomputers.com

© Copyright by Maker Media GmbH.