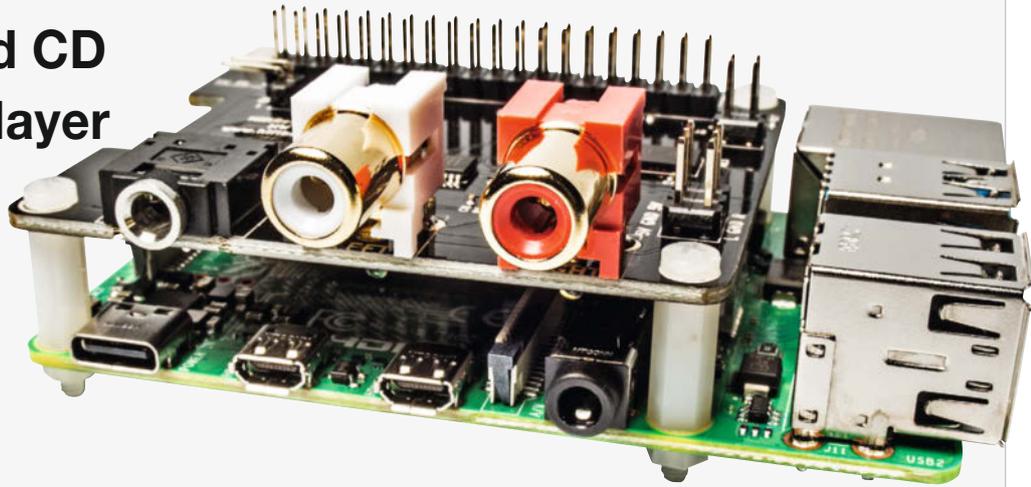




# HiRes-Audio mit dem Raspberry Pi

- ▶ Besser als MP3 und CD
- ▶ Günstiger Media-Player für Ihre Anlage
- ▶ Selber bauen und kostenlos Dienste testen



## Workshop

- ▶ Platinen layouts und fräsen
- ▶ Smart Home: Raus aus der Cloud
- ▶ Pappmaché aus 3D-Druck-Formen

## Projekte

- ▶ Smart: Pi steuert Luftentfeuchter
- ▶ Füllstandsmessung für Zisternen
- ▶ Alexa dimmt IKEA-Todesstern

## Doppel-Lötstation

- ▶ Lötkolben mit 40W und 150W
- ▶ Steuerung über Touchscreen
- ▶ Platinendaten zum Download



## KI für Maker

- ▶ Chatbot programmieren in AIML
- ▶ Einsteigertutorial: Bilderkennung
- ▶ Videodrohne verfolgt Objekte



6/21  
9.12.2021  
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](http://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.

### Der am einfachsten zu bedienende 3D-Drucker von Monoprice

Der MP Cadet von Monoprice ist der perfekte 3D-Drucker für Anfänger, die ein anwenderfreundliches Gerät suchen.

- druckt Objekte bis zu 100 x 105 x 100 mm
- Technologie:  
Schmelzschichtungsverfahren (FDM/FFF)
- Filament: PLA, PLA Pro
- Schichtdicke: 0,10 - 0,40 mm
- fertig montiert geliefert



**37% SPAREN** ~~189,00~~

Bestell-Nr.:  
**MONOPRICE 140959**

**119,00**

Nur solange der Vorrat reicht!



### Unser YouTube Kanal

Auf unserem YouTube Kanal finden Sie über 80 spannende Elektronik How-To's & Unboxings

Jetzt entdecken ►  
[www.youtube.com/reichelt](http://www.youtube.com/reichelt)



Ein Fest der Technik  
Risikolos und bequem einkaufen.

Erfahren Sie mehr ►

[www.reichelt.de/fest-der-technik](http://www.reichelt.de/fest-der-technik)



■ Top Preis-Leistungs-Verhältnis

■ über 120.000 ausgesuchte Produkte

■ zuverlässige Lieferung – aus Deutschland in alle Welt

[www.reichelt.de](http://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](http://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: 25. 11. 2021



Toby Giessen

## „Und was für ein Maker bist du?“

Meine Kollegin sitzt vor mir und schaut mich erwartungsvoll an. Ich bin der neue Volontär, gerade in der zweiten Woche und habe diese Frage schon lange nicht mehr gestellt bekommen. Früher im Studium lautete sie: „Und was für ein Designer bist du?“ – da ist sie wieder, diese Anspannung. Ich weiß, dass sie aus Interesse fragt, aber mir fällt die Antwort dennoch jedes Mal schwer. Natürlich könnte ich sagen, dass ich Spiele programmiere, 3D-Drucker baue, meine Pflanzen dank Automatisierung selbst im Weltall überleben könnten und ich am liebsten Stan Winston bei *Jurassic Park* oder Disney als *Imageneer* unterstützen würde. Die Frage geht aber tiefer.

Erst gestern hat jemand zu mir gesagt: „Ich würde mich nicht als Maker bezeichnen, aber ich schraube gern an meinen Fahrrädern.“ So oder so ähnlich haben das vermutlich schon viele von euch gehört. Häufig begegnen mir auch Menschen, die überzeugt behaupten, gänzlich unkreativ zu sein. Woran liegt das und wer darf das überhaupt beurteilen? Es mag sein, dass Kreativität schwer zu greifen oder in einer Leistungsgesellschaft vermeintlich nur denjenigen vorbehalten ist, die „richtig Ahnung“ davon haben. Wenn ihr mich fragt, sollten gesellschaftliche Grenzen die eigene kreative Entfaltung nicht einschränken – außer, jemand könnte zu Schaden kommen. Deswegen ist es so schön, dass es die Maker-Szene gibt, den

Ort, an dem alle willkommen sind, um Kreativität zu feiern, miteinander und voneinander zu lernen, mit dem gemeinsamen Ziel, etwas Neues zu schaffen, zu reparieren oder zu optimieren – aus reinem Interesse und Spaß an der Sache. Das heißt nicht, dass das Verwenden eines Werkzeugs dazu verpflichtet, sich Maker zu nennen. Ich ermutige nur alle, an ihre eigene Kreativität zu glauben und stolz auf das zu sein, was sie erschaffen. Und wenn es nur eine Kleinigkeit ist, die den Unterschied macht – jede Schraube zählt.

„Ich bin kreativ, neugierig und lerne gern,“ entgegne ich der Kollegin nach einer kurzen Pause. „Daraus entsteht dann etwas Neues. Das fasziniert mich. Und ich freue mich, durch meine Arbeit in der Make-Redaktion bald noch viel mehr Maker kennenzulernen, denen es genauso geht.“

Ákos Fodor

► [make-magazin.de/xrqu](https://make-magazin.de/xrqu)

**Sagen Sie uns Ihre Meinung!**

[mail@make-magazin.de](mailto:mail@make-magazin.de)

# Inhalt

## Workshops

Wer heute „mal eben“ eine Platine für sein kleines Projekt braucht, muss nicht mehr mit Fotolacken und Eisen-III-Chlorid herumschmaddern – es genügt eine kleine CNC-Fräse und der richtige Workflow. Der kommt auch bei unserem Pappmaché-Workshop zum Tragen, wo erstaunlich stabile Objekte in 3D-gedruckten Formen entstehen.

- 60** Erstaunlich stabil: Pappmaché
- 104** Der Weg zur Platine, Teil 1



## HiRes-Audio mit dem Raspberry Pi

Keine Frage: Für echte High-End-Freaks ist unser Projekt viel zu günstig und zu ehrlich. Erschwingliche, aber trotzdem hervorragende D/A-Wandler und ein simpler Raspberry Pi liefern ein unkomprimiertes Streaming-Vergnügen, das jeden Toningenieur begeistert – wenn auch vielleicht nicht den solventen HiFi-Freund mit Vorliebe für esoterisches Marketing-Geschwätz.

- 10** HiRes-Audio mit dem Raspberry Pi

- 3** Editorial
- 6** Leserforum
- 8** Test: Raspberry Pi Zero 2
- 10** Projekt: HiRes-Audio mit dem Raspberry Pi
- 18** Projekt: Luftentfeuchter mit Raspi steuern
- 24** Projekt: ESP32-Orgel mit Piano-Tastatur
- 30** Projekt: IKEA-Todesstern dimmen mit Alexa
- 36** Werkstattberichte: Neues aus der Szene, Comic
- 38** In eigener Sache: Willkommen im Make: Universum
- 42** Projekt: KI steuert Follow-Me-Drohne
- 48** Workshop: KI für den ESP32
- 54** Know-how: Chatbot programmieren mit AIML
- 60** Projekt: Erstaunlich stabile Objekte aus Pappmaché
- 66** Was uns inspiriert: Mechatronik-Street-Art, Gehörschutz-Kopfhörer, Portrait-Plotter
- 70** Make Projects: Gewinnerprojekte des Smart-Home-Wettbewerbs

## KI für Maker

Neuronale Netze auf Mikrocontrollern und Kleinstrechnern – das ist keine Utopie mehr. Um Ihre Projekte mit künstlicher Intelligenz aufzuwerten, müssen Sie nicht unter Tage in einem geheimen Labor frönen: Es reicht, sich ein wenig mit frei verfügbaren Libraries und aktueller Hardware auseinanderzusetzen.

- 42 KI steuert Follow-Me-Drohne
- 48 KI für den ESP32
- 54 Chatbot programmieren in AIML



## Projekte

Wussten Sie, dass man ein Blutdruck-Messgerät dazu verwenden kann, den Inhalt eines großen Tanks fernüberwacht zu messen? Dass sich der legendäre IKEA-Todesstern mit Alexa steuern lässt oder ein Luftentfeuchter mit dem Raspi? Manchmal sind wir selbst erstaunt, was so alles geht – und oft auch, wie leicht.

- 18 Luftentfeuchter mit Raspi steuern
- 30 IKEA-Hack: Todesstern dimmen mit Alexa
- 74 Füllstandsmessung einer Regenwasserzisterne



- 74 Projekt: Füllstandsmessung einer Regenwasser-Zisterne
- 80 Reingeschaut: Smart Home vom Discounter
- 82 **Know-how: Tuya ohne Cloud**
- 90 Community-Projekte: Winterlandschaft im Glas, Kunst-Shredder, insektoider Roboter
- 96 **Projekt: HotPens – die duale DIY-Lötstation**
- 104 **Workshop: Der Weg zur Platine, Teil 1**
- 116 Workshop: Gehäusebau mit FreeCAD, Teil 2
- 120 Tipps & Tricks: Mehrweg-Filament-Spule
- 124 Kurzvorstellungen: Mikrocontroller-Boards, Smart Home, LoRaWAN, 3D-Drucker
- 128 Medien: Lego-Roboter, Heimautomation, 3D-Druck im Flugmodellbau, Animations-Werkstatt für Kinder
- 130 Impressum/Nachgefragt

## Doppel-Lötstation

Die Weller-Lötstationen gehören zu den besten ihrer Zunft – aber auch zu den teuersten. Auf die Qualität der RT-Lötspitzen muss man nicht verzichten, wenn man sie mit einer selbstgebauten Versorgungseinheit ergänzt. Unser HotPens-Projekt befeuert gleich zwei Kolben mit 40 und 150 Watt und ist damit für alle Lötarbeiten gewappnet.



- 96 HotPens: Duale DIY-Lötstation

Themen von der Titelseite sind rot gesetzt.

# Leserforum

## Tipfehler im Artikel

Carsten Wartmann, Ikea-Staubsensor mit WLAN aufbohren, Make 5/21, S. 14

Im Heft steht auf Seite 16 im Fließtext sowie im Kasten auf Seite 18 fälschlicherweise, dass Byte 5 mit 255 (FF in Hexadezimal) multipliziert würde, um dann Byte 6 zu addieren, damit der PM2.5-Wert herauskommt. Korrekt ist die Multiplikation mit 256. Unser Code von GitHub funktioniert aber, weil dort die Multiplikation durch eine Verschiebung um acht Bits realisiert ist (Zeile 71):

```
return (serialRxBuf[5] << 8 |
        serialRxBuf[6]);
```

## Use it or loose it

Editorial: Nostalgie ohne Reue, Make 5/21, S. 3

Die meisten Jäger und Sammler werden die Krankheit lebenslang nicht mehr los, zumal das Jagen und Sammeln ja auch in unseren Genen angelegt ist. Das Einzige, was nach meiner Erfahrung hilft, ist, wählerisch zu werden, und alles Zeug, was ins Haus kommt, nach dem „Use It Or Loose It“-Prinzip zu bewerten. Das sage ich aus der Perspektive von jemandem, der schon mehrmals in seinem Leben in die „Sammlerfalle“ getappt ist und auch weiß, was für eine Mühe es macht, sich davon zu

befreien, wenn der Berg an unerledigten Ideen eine gewisse Größe erreicht hat. Das Ganze soll doch Spaß machen, und nicht in einen unbezahlten Zweitjob ausarten, oder?

Frank Nerstheimer

## Lieblingsrechner weg

Meinen Vater, sonst bei jeder Schraube mich fragend, ob ich sie noch bräuchte, packte wohl der Rappel, als er von allen im Keller des Elternhauses geparkten Altgeräten ausgerechnet meinen Atari 600XL Homecomputer zum Recycling gab. Meinen Lieblingsrechner, der meinen späteren Berufsweg entscheidend steuerte und in den ich liebevoll neben einer Parallelschnittstelle und LEDs auch eine DRAM-Erweiterung von 16 auf sagenhafte 64kB per Einzelchips einlötete (ein Kaufmodul war viel zu teuer), den ich in- und auswendig kannte (damals noch möglich), verharnte nun in einem Elektroschrottcontainer, den mein Vater reuevoll mit mir auch prompt aufsuchte. Jedoch hatte wohl auch ein anderer den Wert dieses Schmuckstücks erkannt: Außer ein paar Anschlusskabeln war auch nach einer Stunde Wühlen nichts mehr auffindbar. Weg. Genau so wie die Nerven des Personals vor Ort ob unserer „Wühlblockade“. Ich hoffe, „mein“ Atari ist heute ein Ausstellungsstück eines Computermuseums!

Jürgen Striegel

## Werde schon Platz finden

Ich stehe vor genau dem gleichen Problem. In meinem Bastelkeller werden die von mir aufgestellten Schwerlastregale immer voller, auch deshalb, weil ich vor einiger Zeit begonnen habe, vormals aussortiertes Equipment (Stichwort C64, Amiga, ZX81 etc.) gebraucht wieder zu erwerben. Meine Raspberry-Sammlung umfasst mittlerweile auch 24 Stück, nur ein kleiner Teil davon ist in produktiver Verwendung. Aber eine Trennung kommt nicht in Frage, für 2-3 weitere Regale werde ich schon noch Platz finden.

Johannes Puchinger

## Blaue „Stromwolken“ drohen

Uwe Rohne, Photovoltaik an der E-Auto-Wallbox, Make 5/21, S. 20

Da hoffe ich mal, das niemand auf die Idee kommt, Euch die Kosten für das kaputte Bordladegerät in Rechnung zu stellen. Wer eine ZOE mit dem Bauvorschlag lädt, wird bei der Umschaltung von ein- nach dreiphasiger Ladung sein blaues Wunder erleben, wenn die blauen „Stromwolken“ aus dem Ladegerät im Fahrzeug kommen. Ein veritabler Kurzschluß wäre das kleinste Übel. Je nach Ladevorgang (also ein- oder dreiphasig) werden in der ZOE die Motorspulen (die ZOE nutzt den Motor zur Ladung) anders verschaltet. Wenn man während der Ladung einfach die Phasen neu schaltet, knallt es im Fahrzeug. Das Problem gab's auch schon mit klebenden Schützen im Fahrzeug, wenn man vorher jeweils im anderen Modus geladen hat.

Roland Klose

*Die Probleme des ZOE sind im Grundsatz bekannt. Im Artikel werden nicht direkt Phasen ab- oder dazu geschaltet, sondern es wird bei einer Umschaltung erst über alle drei Phasen stromlos geschaltet und dann mit ein oder drei Phasen neu „gestartet“. Auch für das manuelle Umschalten, was ja im Artikel empfohlen wird, empfiehlt sich ein Schalter, der eine Neutralposition hat, also: dreiphasig – null – einphasig. Zudem verweist der Artikel am Ende darauf, im Zweifelsfall die Bedienungsanleitung zu Rate zu ziehen. (Uwe Rohne)*

## Kontakt zur Redaktion

Leserbriefe bitte an:

[heise.de/make/kontakt/](https://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](https://www.make-magazin/forum)



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



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



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



[www.twitter.com/MakeMagazinDE](https://www.twitter.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.

## Überschussladen ohne Löten

Ich habe gerade mit Interesse Euren Artikel zur Überschussladung mit der ESP8266-Ladeautomatik gelesen. Da ich selber für mein neues E-Auto eine bestehende Open-Source-Lösung rein auf Software-Basis für eine funktionierende Überschussladung angepasst habe, wollte ich Euch kurz die Links zu zwei bereits verfügbaren Software-Lösungen mitteilen:

1. <https://openems.io/>
2. <https://evcc.io/>

Beide Software-Projekte bieten bereits eine Unterstützung von bestehenden Wallboxes, Energiezählern, Elektroautos und weiteren Komponenten, sodass Nutzer nicht zwingend Kommunikationsprotokolle „reverse engineer“ müssten. Bei evcc existiert derzeit sogar schon eine erste Implementierung des EEBUS-Standards. Auch kann das Starten des Ladevorgangs und die genutzte Ladeleistung bei manchen E-Autos über Online-APIs gesteuert werden, wodurch man nicht steuerbare Wallboxen ohne Hardware-Anpassung umgehen kann.

Markus Wagner

## E-Auto als mobiler Speicher?

Ich hätte eine Frage dazu, evtl. wäre das sogar einen Folgeartikel wert: Falls man die Kombination PV-Anlage und E-Auto besitzt bzw. plant anzuschaffen – gibt es die Möglichkeit, den Autoakku auch als Batteriespeicher (mit) zu nutzen? Soviel ich weiß, haben die meisten E-Autos größere Akkus als die Batteriespeicher für mittelgroße PV-Anlagen (für den Privatbereich). Gibt es Wallboxen, die auch so etwas unterstützen? Evtl. etwas „getweakt“ mit zusätzlicher Elektronik?

Vielleicht müsste man die Ladezyklen der Autobatterie etwas begrenzen, um die Lebensdauer nicht zu sehr zu verkürzen. Auf der anderen Seite könnte man eine interessante Rechnung aufmachen: Für den doppelten Preis eines mittleren Batteriespeichers bekommt man schon ein gebrauchtes E-Auto und hat dabei noch den Vorteil eines Autos :-). Nachdem das CO<sub>2</sub>-Problem immer dringender wird und ich noch keine große Aufbruchsstimmung wahrnehme, wäre das ein

Projekt, was zumindest interessante Anstöße (und einen praktischen Mehrwert) geben kann!

Clemens Kraus

*Sie sind der Zeit einige Jahre voraus. Sicherlich wird das in einigen Jahren auch möglich sein, dann aber mit einer anderen Wallbox-Technologie. In den aktuellen E-Autos ist sozusagen ein Mini-DC-Lader verbaut, der aus dem Wechsel- bzw. Drehstrom der Wallbox einen Gleichstrom von ca. 400 Volt erzeugt und in die Batterien lädt. Da fehlt also noch einiges, diesen Prozess umzukehren. Der Hyundai IONIQ 5 allerdings zeigt meines Wissens bereits erste Ansätze dafür, allerdings nur bis ca. 1,5 kW Leistung. Also in der MAKE Nr. 1/2024 kann die Idee vielleicht als DIY konkret werden. (Uwe Rohne)*

## (Leider) Smart Home

**Poster-Beilage: Bastelboard-Übersicht und Smart-Home-Projekte aus Make, Make 5/21, S. 13**

Als heute meine Zeitung bei mir angekommen ist, habe ich mich sehr über die angekündigte Übersicht über die Mikrocontroller auf dem Poster gefreut. Schließlich erstelle ich selbst zur Zeit eine ähnliche Übersicht wegen meiner Abschlussarbeit. Leider kam dann doch ein wenig die Ernüchterung, als ich das Poster geöffnet habe: Das Layout ist so zwar schon hübsch geworden, aber insgesamt leider nicht sehr informativ. Eine tabellarische Übersicht, anhand der man seinen µC passend zum nächsten Projekt auswählen kann, wäre sicherlich besser, als Mikrocontroller im Weltall. So ist es (leider) die Smart-Home-Übersicht geworden, die bei mir als Ideengeber meine Wand zieren soll. Trotzdem ein schönes Konzept, ich freue mich auf mehr davon. Als Anregung für eine nützlichere Übersicht: Stromaufnahme bzw. Spannung, verwendbare Sprachen, Netzwerkadapter und ähnliches. Die Anschlussübersicht war ein guter Anfang.

Thomas

## Viel zu bunt

Danke für das Poster, das ist erstmal eine super Idee und gute Überraschung. Aufhängen werde ich das Poster dennoch nicht, da es mir

viel zu bunt ist. Für's Kinderzimmer sicher geeignet, aber für meine Bastelecke weniger. Ich werde das Poster jedoch als Anregung nehmen und mir ein eigenes Übersichtsposter erstellen, das weniger bunt und aufgeräumter ist. Vielleicht machen das andere Maker auch und man könnte die Posterdesigns auf Make tauschen. Dennoch danke.

Martin Dürrschnabel

*Danke für die Anregung! Wir freuen uns natürlich weiterhin über Ideen und Anregungen für weitere Poster und natürlich auch über Entwürfe dafür! Einfach mailen an [mail@make-magazin.de](mailto:mail@make-magazin.de)*

## Gervifonn mit NAS?

**Fabian Jansen, Gervifonn: CD-Cover erkennen mit KI, Make 5/21, S. 48**

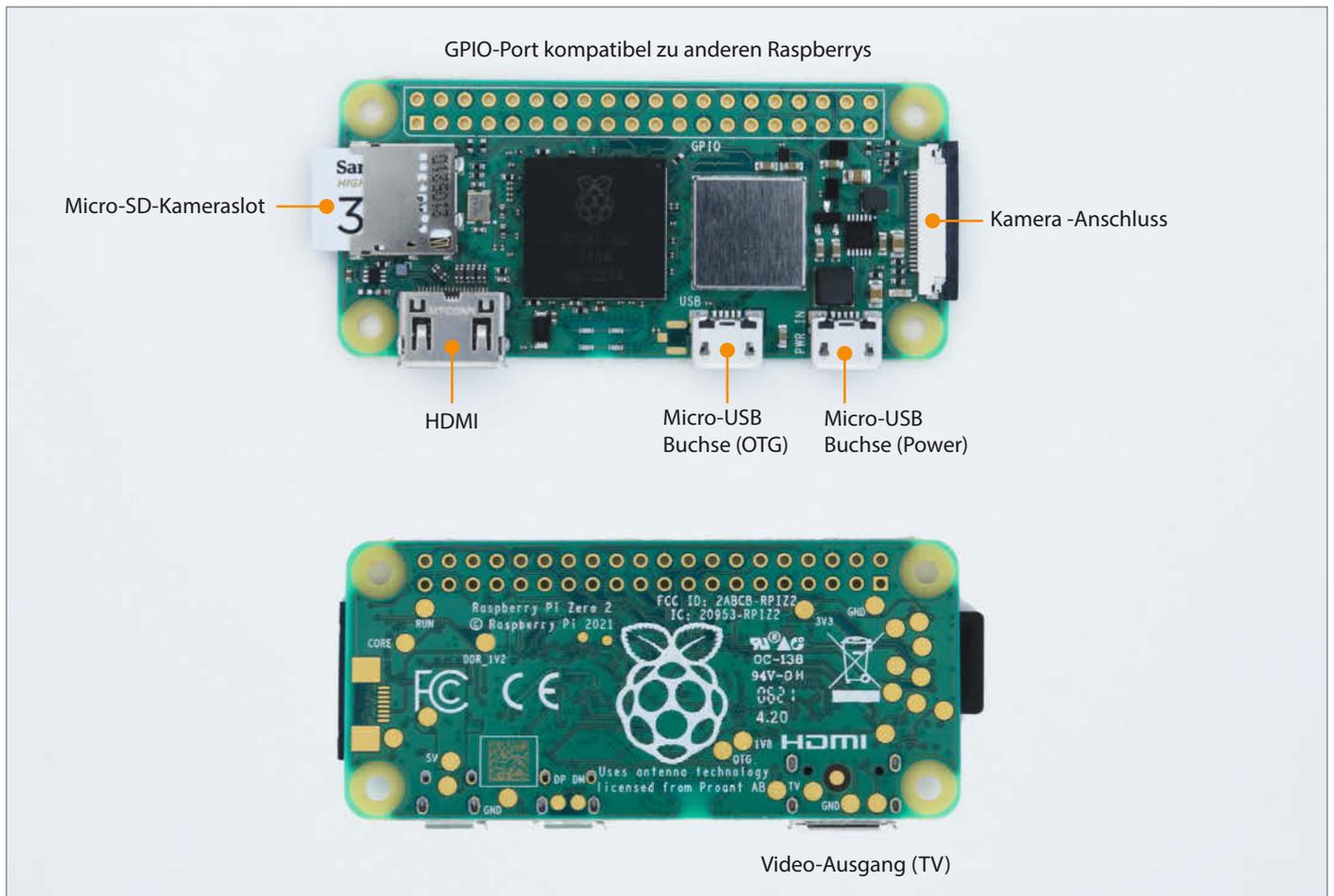
Der 3D-Drucker ist bereits am Drucken ... Die Hardware liegt bereit ... Ihr tolles Projekt ist als Geburtstagsgeschenk anvisiert. Eine Frage: wäre auch eine Integration an ein NAS (bsp. Festplatte an der Fritz!Box) denkbar?

Oliver Voß

*Ein NAS als Speicher für die Musiksammlung zu verwenden, sollte kein Problem sein. Auf dem Gervifonn selbst müssen sowieso keine Musikdateien abliegen, nur die Ordnerstruktur und die Coverfotos müssen im Dateisystem verfügbar sein. Der Music Player Daemon, der tatsächlich auf die Musik zugreifen muss, unterstützt von Haus aus mehrere Möglichkeiten direkt auf Netzwerkspeicher zuzugreifen (siehe Link). Wenn das Gervifonn nicht auf eine bestehende Multiroom-Infrastruktur aufsetzt, sondern als Standalone-Gerät mit eigener Soundkarte betrieben werden soll, würde es sich anbieten, auf Snapcast zu verzichten und mit dem MPD direkt die Soundkarte anzusteuern. Für dieses Szenario habe ich mittlerweile den Lautstärkereglern des Gervifonn im Code auf GitHub angepasst. (Fabian Jansen)*

► [make-magazin.de/xx67](https://make-magazin.de/xx67)

Zuschriften zum Sonderheft *Loslegen mit Metall* veröffentlichen wir in der kommenden Make-Ausgabe 1/22. Wenn Sie dazu noch Feedback haben, mailen Sie uns!



# Raspberry Pi Zero 2

**Nachwuchs bei Raspberrys: Mit der zweiten Version des Raspberry Pi Zero gibt es jetzt ein Board mit vier Prozessorkernen. Wir haben ausprobiert, was das bringt und welche Unterschiede es noch zum Vorgänger gibt.**

von Heinz Behling

**B**ereits 2015 erschien der erste *Raspberry Pi Zero*, da wurde es Zeit, den kleinen Raspi zu überarbeiten. Herausgekommen ist dann im Oktober dieses Jahres ein Board, an dem sich äußerlich kaum etwas verändert hat (siehe Titelbild des Artikels). So ist es immer noch halb so groß wie seine Brüder Raspberry Pi 3 und 4. Auch die USB- und die Mini-HDMI-Buchsen sitzen wie der Speicherkartenslot und die 40polige GPIO-Leiste an der alten Stelle. Probleme mit Einbauten in Gehäuse oder schon vorhandene Geräte dürfte es nicht geben, solange rings um die HDMI-Buchse etwa 1,5mm Platz ist. Diese Buchse hat nun nämlich einen leicht nach

außen gebogenen Rand und steht etwas aus der Platine heraus.

Der erste Blick auf die Platine zeigt drei Unterschiede: Da ist der größere Prozessor, der jetzt auf den Namen *RP3A0* hört. Er besitzt nun vier Kerne (A53, 1GHz) und soll dem Chip des Raspberry Pi 3 entsprechen. Die 512 MB RAM sitzen im selben Gehäuse, quasi huckepack auf dem Prozessor. Die Kühlung scheint durch das ein wenig größere Gehäuse etwas besser zu sein. In unseren Test lag die Prozessortemperatur unter Last mit 51°C etwa 2° niedriger als bei der alten Zero-Version.

Der zweite auffällige Unterschied ist die Blechabschirmung über dem WLAN-/Blue-

tooth-Baustein. Dies hatte aber keine Auswirkungen in der Reichweite. Vermutlich dient das eher der Einhaltung von EMV-Vorschriften.

Der dritte Unterschied ist schon etwas schwieriger zu finden: Das vierpolige Anschlussfeld zwischen GPIO-Leiste und Kameraanschluss fehlt. Dort lag beim alten Zero unter anderem ein Video-Anschluss, der ein Signal nach PAL-Norm zum Anschluss an analoge Eingänge älterer TV-Geräte oder Monitore zuließ. Dieser Anschluss liegt jetzt nur noch als kreisrunder goldener Testpunkt an der Unterseite der Platine. Dort finden sich auch Lötunkte der Aktivitäts-LEDs für den WLAN- und Bluetooth-Transfer.

## Leistung

Die vier Kerne des Prozessors lassen auf einen kräftigen Leistungsschub hoffen. Wir ermittelten Werte für den Speicherzugriff, den Datentransfer über die USB-Schnittstellen sowie die Geschwindigkeit beim Verschlüsseln und beim Komprimieren und Dekomprimieren. Aber auch die aufgenommene Leistung interessierte uns. Im Leerlauf zieht der Raspberry Pi Zero 2 etwa 110mA und liegt damit in etwa gleichauf mit seinem Vorgänger. Unter Vollast braucht der neue etwa 100mA weniger als der alte Zero. Das kann bei Akku- oder Batteriebetrieb schon einen deutlichen Unterschied machen. Aber nur, wenn man den Zero dann auch längere Zeit in Vollast laufen lassen würde.

Die Prozessorleistung (siehe Tabelle) liegt hingegen deutlich über der des Vorgängers: Der Test mit dem Komprimierprogramm 7ZIP brachte etwa die doppelte Leistung, selbst wenn man nur einen Prozessorkern nutzte. Bei Einsatz aller vier Kerne war die Leistung beim Komprimieren etwa dreimal, beim Dekomprimieren sogar viermal so groß und erreichte etwa 55 Prozent der RasPi4-Leistungsdaten in diesem Punkt. Das kann sich sehen lassen.

Beim Verschlüsseln mit *openssl* ergab sich eine etwa 40-prozentige Steigerung gegenüber der alten Zero-Ausgabe und ebenfalls 55 Prozent der 4er-Leistung. Auch bei den Speicherwerten für den Datentransport sieht es ähnlich aus. Die Latenzzeit beim Speicherzugriff ist hier bei 32KB-Paketgröße deutlich besser: Da arbeitet vermutlich ein mindestens 32MB großer Cache im Chip. Insgesamt ist offenbar schnelleres RAM verbaut, da auch die anderen beiden Latenzzeiten geringer sind.

Der Zugriff auf Massenspeicher (im Test auf einen sehr schnellen USB-Stick) ist erwartungsgemäß deutlich schwächer als beim RasPi4 und bringt auch keine Verbesserungen gegenüber dem alten Zero. Schließlich ist hier der Flaschenhals USB 2 im Spiel, der ohne Cache im Zero nicht wesentlich mehr als 33MB/s zulässt.

## Fazit

Der neue Raspberry Pi Zero 2 verfügt über eine deutlich höhere Rechenleistung als sein Vorgänger, der Betrieb mit einem Desktop bei hoher Auflösung war kein Problem. Somit könnte er für viele Anwendungen interessant werden, die den alten Zero überforderten. Wäre da nicht doch noch ein Nadelöhr namens Arbeitsspeichergröße. Nur 512 Megabyte sind für anspruchsvolle Tätigkeiten, etwa beim KI-Training, zu wenig. Bei den herkömmlichen Zero-Anwendungen hingegen sollte man nicht mehr zur alten Version greifen, zumal die höhere Leistung auch noch etwas stromsparender ist. —hgb

## Kurzinfo

- » Äußere Unterschiede zum alten Raspberry Pi Zero
- » Der neue Prozessor RP3A0
- » Benchmarks im Vergleich zum Raspberry Pi Zero und 4

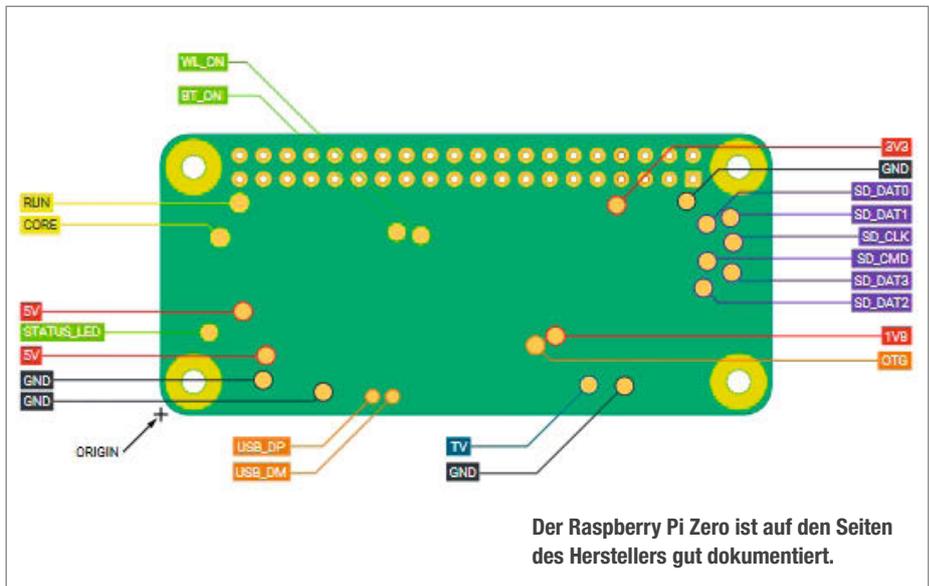
## Mehr zum Thema

» Heinz Behling, Raspberry Pi 4B+, Make 4/19, S. 30

Alles zum Artikel im Web unter [make-magazin.de/x4xk](https://www.make-magazin.de/x4xk)

## Messwerte Benchmarks

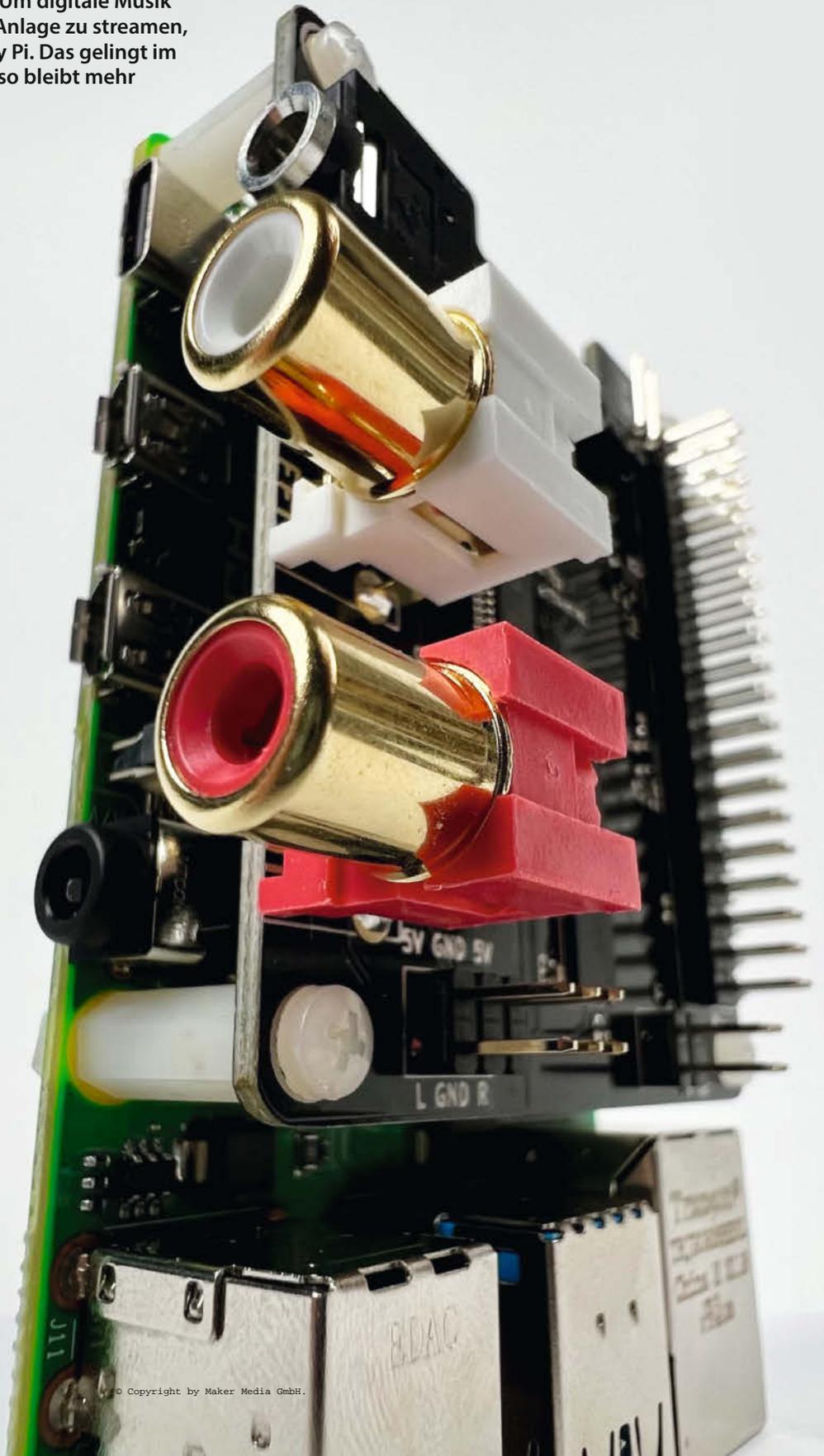
Boards	Raspberry Pi 4B	Raspberry Pi Zero	Raspberry Pi Zero 2
<b>7ZIP</b>			
<b>mit einem Prozessor-Kern</b>			
Compress	1821 MIPS	314 MIPS	656 MIPS
Decompress	1991 MIPS	624 MIPS	1075 MIPS
<b>mit allen Prozessor-Kernen</b>			
Compress	3307 MIPS	314 MIPS	1711 MIPS
Decompress	7521 MIPS	624 MIPS	4168 MIPS
<b>openssl (aes-128-gcm)</b>	36145 KB/s	11678 KB/s	19928 KB/s
<b>tinymembench</b>			
memcpy	2401,6 MB/s	417,1 MB/s	1302 MB/s
Memory latency time (32kb/1MB/16MB)	0ns/40ns/192ns	20,1ns/205ns/243ns	0ns/80ns/161ns
<b>hdparm</b>			
cached	749 MB/s	492 MB/s	645 MB/s
unbuffered	336 MB/s	33 MB/s	34 MB/s
<b>Stromaufnahme</b>			
Leerlauf	0,7A	0,10A	0,11A
unter Last	>2A	0,54A	0,45A
<b>Temperatur</b>			
unter Vollast	>80°C	53°C	51°C



# HiRes-Audio mit dem Raspberry Pi

Für HiFi können Sie viel Geld ausgeben. Um digitale Musik in höchster Qualität an die bestehende Anlage zu streamen, reicht allerdings ein günstiger Raspberry Pi. Das gelingt im Handumdrehen und ganz ohne Löten – so bleibt mehr Zeit, Ihre Musik neu kennenzulernen.

von Roman Radtke, Ákos Fodor



**E**ines Tages, wenn Sie in gemütlicher Runde mit Freunden in alten Zeiten schwelgen, werden Sie vielleicht fragen: „Erinnert ihr euch noch an MP3? War das nicht eine tolle Erfindung, als plötzlich all unsere Lieblingsongs in die Hosentasche passten?“ Währenddessen wählen Sie über Ihr Smartphone aus Millionen Musiktiteln den nächsten aus, um ihn in HiRes (High Resolution) auf die HiFi-Anlage zu streamen. „Ja, das war wie beim Walkman.“ Und jemand ruft wohlwollend aus der Ecke: „Aber die Audioqualität auch!“ Alle lachen.

Mit wenigen Handgriffen könnte dieser Tag schon morgen Abend sein. Wie das geht, zeigen wir Ihnen mithilfe eines Raspberry Pi und dem HiFiBerry-Erweiterungsboard und schicken ihre Ohren auf akustische Höhenflüge – ganz gemütlich vom Sofa aus. Die Einzigen, die sich darüber beschweren könnten, sind Ihre Nachbarn.

Aber nochmal einen Gang zurückgeschaltet. Was ist überhaupt HiRes-Audio und wieso sollten Sie das unbedingt ausprobieren? Kurz gesagt ist es der heilige Gral der Audiodigitalisierung, denn Audiosignale aufzuzeichnen und wiederzugeben, geht immer mit Verlust einher. Deshalb setzen viele Musikstudios auf HiRes, also eine höhere Auflösung bei der Aufzeichnung als sonst üblich, damit Sie beim Hören so nah wie möglich am Original sein können.

Ob Sie den am Computer sichtbaren klanglichen Unterschied später auch hören und genießen können, hängt von der Qualität der Aufnahme, der Signalkette auf dem Weg zum Ohr einschließlich äußerer Störfaktoren und natürlich dem Ohr selbst ab. Im Folgenden umreißen wir einmal, was das konkret bedeutet.

### Digitalisierung von Audio

Bei einer analogen Audioaufnahme wird ein akustisches Eingangssignal mithilfe eines Mikrofons in ein analoges elektrisches Signal umgewandelt, verstärkt und dann auf ein analoges Wiedergabemedium übertragen, wie eine Schallplatte oder einen Lautsprecher bei einem Live-Konzert. Die Membran des Lautsprechers gibt idealerweise dasselbe Signal wieder, das das Mikrophon empfangen hat und ein dazwischen geschalteter Verstärker sorgt dafür, dass ein ganzer Saal oder Ihr Wohnzimmer damit beschallt werden kann.

Für die Verarbeitung an einem Computer müssen analoge Signale zuerst in digitale umgewandelt werden. Hierzu wird ein Analog-Digital-Wandler (englisch: *Analog-Digital-Converter* oder kurz *ADC*) verwendet. Dieser tastet die Audiosignale ab und je genauer das passiert, desto stärker entspricht das digitale Endprodukt der analogen Eingabe. Die Genauigkeit wird dabei durch die verwendete *Samplingrate* und *Samplingtiefe* bestimmt ❶.

## Kurzinfo

- » Einen HiRes-Player mit Raspberry Pi und einem HiFiBerry DAC zusammenstecken
- » Player-Software installieren und einrichten
- » Passende Streamingdienste verbinden

### Checkliste



**Zeitaufwand:**  
etwa eine Stunde



**Kosten:**  
ca. 130€ für den Player  
zzgl. mögliche Kosten für die Musik

### Material

- » Raspberry Pi 4 Model B 4GB RAM
- » microSD-Karte mit 8GB (empfohlen)
- » HiFiBerry DAC2 Pro
- » Raspberry-Pi-5V-Netzteil

### Werkzeug

- » Kleiner Kreuz-Schraubendreher

### Mehr zum Thema

- » Carsten Meyer, Lautsprecher richtig einsetzen, Make 2/2020, S. 104
- » Heine Nielsen, Designer-Lautsprecher, Make 4/2019, S. 54
- » Ulrich Schmerold, Wertewandel, Make 5/2017, S. 40

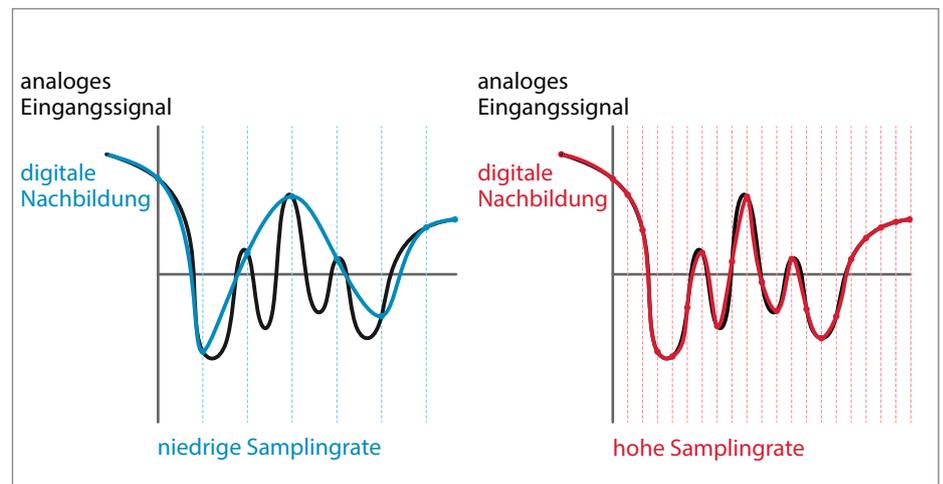
Alles zum Artikel im Web unter [make-magazin.de/xwsg](https://make-magazin.de/xwsg)

**Samplingrate (Hz):** Akustische Signale bestehen aus einem Mix an spezifischen Frequenzen, gemessen in Hertz (Hz). Für die Erfassung des Signales wird eine mehr als doppelt so hohe Abtastrate verwendet, als die höchstmöglich auftretende Frequenz des aufzuzeichnenden Signals – also z.B. eine Samplingrate von 44,1kHz, um ungefähr jede Frequenz abzubilden, die unser menschliches Gehör wahrnehmen kann, von 20Hz (tiefster Ton) bis 20kHz (höchster Ton).

**Samplingtiefe (Bit):** Ähnlich verhält es sich mit der Samplingtiefe, die in Bits angegeben wird. Sie ist dafür verantwortlich, in

welchem Dynamikumfang ein Audiosignal aufgezeichnet werden kann, also wieviele Abstufungen zwischen lautlos und voller Lautstärke zur Verfügung stehen. Würden wir beispielsweise mit 1 Bit aufzeichnen, hätten wir nur laut oder lautlos. Bei 2 Bits wären es dann 4 Stufen und so weiter.

Um aus den digitalisierten Aufnahmen etwas Hörbares zu machen, benötigen wir wiederum einen Wandler, der das genaue Gegenteil macht – einen Digital-Analog-Wandler (englisch: *Digital-Analog-Converter* oder kurz *DAC*). Wie beim ADC hängt die Qualität der Ausgabe auch hier von den Eigenschaften des jeweiligen DAC ab.



❶ Je höher die Samplingrate ist, desto genauer können die Signale abgebildet werden.

## HiRes-Audio-Anbieter

	Streaming	Kauf	Testphase	Kosten pro Monat
Amazon Unlimited	Ja	Nein	3 Monate	ab 7,99€ (Familie ab 14,99€)
Apple Music	Ja	Nein	3 Monate	9,99€ (Familie ab 14,99€)
Highresaudio	Ja	Ja	7 Tage	ab 89,99€ für 6 Monate
Linn Records	Nein	Ja	keine	nur für den Kauf
Qobuz	Ja	Ja	1 Monat	ab 12,50€ (Familie ab 20,83€)
Tidal HiFi Plus	Ja	Nein	30 Tage	19,99€ (Familie 29,99€)

### HiRes, Lossless und Lossy

Der Prozess von analog zu digital und wieder zurück ist immer verlustbehaftet und nur so gut wie das schwächste Glied in der Kette. Fakt ist, dass sich selbst mit der höchsten Datenrate und der besten Musikanlage nichts aus einem Stück herausholen lässt, welches schlecht gemastert wurde. Um bei der Audiodigitalisierung möglichst präzise zu sein, verwenden Studios Samplingraten ab mindestens 44,1kHz und Samplingtiefen ab 16 Bit (beides entspricht CD-Audio-Qualität), gehen aber noch höher, um das beste Ausgangsmaterial für die Bearbeitung zu haben. Hier kommen Samplingraten mit 96kHz oder 192kHz und Samplingtiefen von 24 Bit oder 32 Bit zum Einsatz – ab dieser Stufe spricht man von HiRes-Audio.

Diese Daten werden dann in einem verlustfreien Format (*Lossless*) gesichert, z.B. WAV, AIFF oder FLAC, von denen letzteres zwar denselben unverfälschten Inhalt enthält, jedoch komprimiert um ein vielfaches kleiner ist als eine rohe WAV-Datei.

Warum sollte man aber nun von MP3 abkehren? In den Anfängen digitaler Musik war der Speicherplatz knapp und die Qualität dementsprechend gering – nicht zu sprechen von der damaligen Internetgeschwindigkeit. Verlustbehaftete (*Lossy*) Formate, zu denen MP3, AAC und OGG zählen, haben die digitale Musik für unterwegs kompakter gemacht,

verlieren jedoch langsam ihre Daseinsberechtigung. Die Kodierung spart zwar Speicherplatz ein, indem die „unhörbaren“ Anteile eines Audiosignals weggelassen werden, je nach Komprimierungsfaktor kann das aber zu deutlich hörbaren Artefakten führen – ähnlich wie bei einem Foto als JPEG. Heutzutage, mit vielfach größeren Speicherkapazitäten und leistungsfähiger mobiler Hardware, ist es kaum noch notwendig, die Klangqualität zu beschneiden.

### Woher nehmen und nicht stehlen?

Wenn Sie neugierig geworden sind und Musik in HiRes erleben möchten, gibt es zwei Optionen: kaufen oder kostenpflichtig streamen. Sollten Sie bisher noch nicht von der SACD (*Super Audio CD*), der *Audio DVD* oder *Blu-ray-Audio-Discs* gehört haben, könnte es daran liegen, dass diese Formate bei weitem nicht so populär sind wie Audio-CDs, mehr kosten und bis auf das letzte Format spezielle Abspielgeräte erfordern. Einfacher geht es direkt über das Internet. Zu diesem Zweck können Sie aus einer Vielzahl von Dienstleistern wählen, deren Angebote eine Zeit lang kostenfrei getestet werden können. Der Vorteil solcher Plattformen liegt auf der Hand: Alles ist sofort verfügbar. An dieser Stelle sei vermerkt, dass unter den Millionen von angebotenen Titeln in verlustfreier Qualität, HiRes noch einen kleinen, aber

stetig wachsenden Prozentsatz ausmacht und die Samplingqualität variieren kann.

Die bekanntesten Streamingdienste mit HiRes-Unterstützung haben wir in einer Tabelle gegenübergestellt. Drei von ihnen kommen für unser Projekt in Frage, da sie nativ in die Software integriert sind, die unseren HiRes-Player steuern wird: *TIDAL*, *Qobuz* und *Highresaudio*.

### Der HiRes-Player

Davon ausgehend, dass Sie eine gute Anlage daheim haben und jetzt wissen, woher Sie HiRes-Audio beziehen können, fehlt nur noch der HiRes-Player, der die Audiodaten aus dem Netz an die HiFi-Anlage übergeben kann.

Zum Einsatz kommt eine steckbare Lösung, bestehend aus einem Raspberry Pi 4 (Pi 3 geht auch) und einem HiFiBerry DAC <sup>2</sup> des Herstellers *Modul 9*. Anschließend bespielen wir das System mit der Distribution namens *Volumio*.

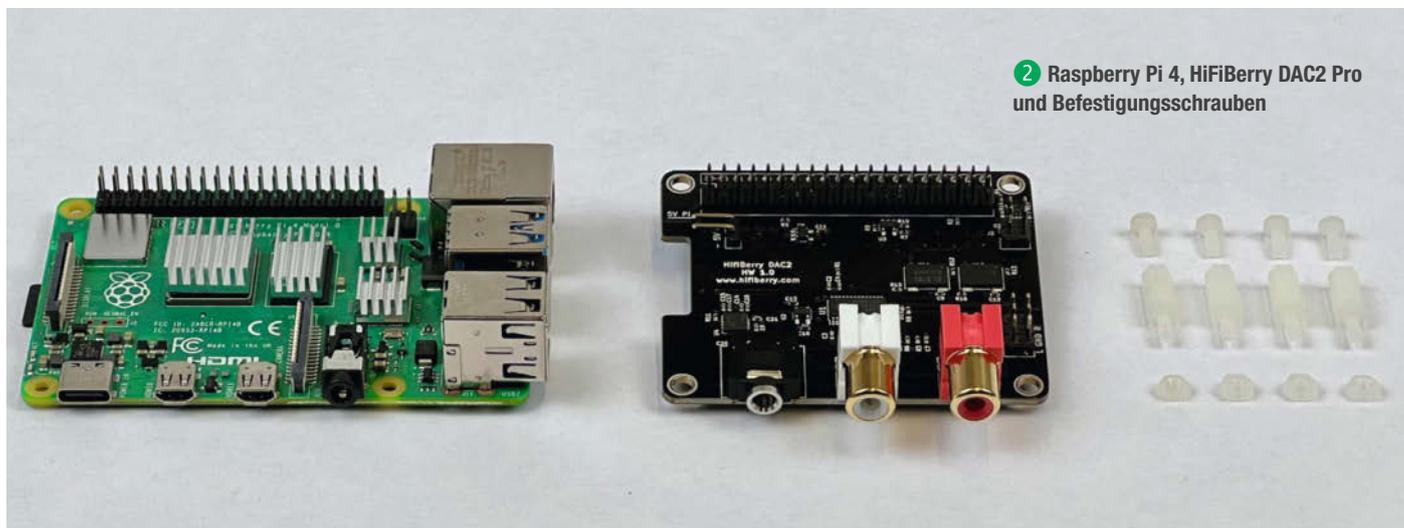
### HiFiBerry DAC

Der für unser Projekt verwendete *HiFiBerry DAC2 Pro* ermöglicht Audiokonvertierung mit bis zu 192kHz bei 24 Bit. Er besitzt zusätzlich zum Stereo-Output eine 3,5mm-Klinkenbuchse sowie einen integrierten Kopfhörerverstärker.

Angeschlossen wird der HiFiBerry als HAT (*Hardware-Attached-on-Top*) über die 40-Pin-GPIO-Schnittstelle des Raspberry Pi <sup>3</sup> und bezieht den Strom direkt aus dem Raspberry-Pi-Netzteil. Mithilfe eines Cinch-Kabels stellen wir zum Schluss noch die Verbindung zur Stereoanlage her.

### Die Software

Jetzt, da alles steckt, benötigen wir einen Software-Player, der die Audio-Daten auch über den DAC ausgibt. Dank der einfachen Einrichtung und Bedienung verwenden wir



<sup>2</sup> Raspberry Pi 4, HiFiBerry DAC2 Pro und Befestigungsschrauben

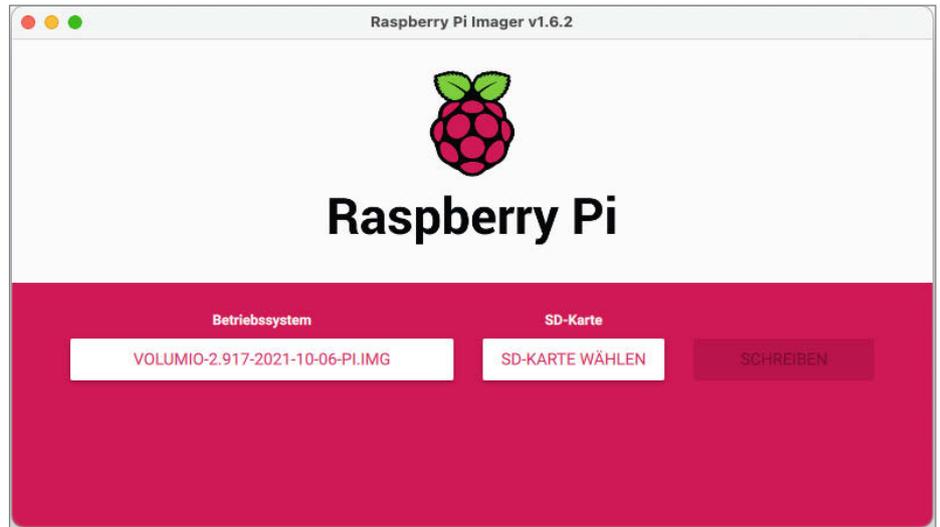


3 Der HiFiBerry DAC2 Pro wird nach dem Zusammenstecken zusätzlich festgeschraubt.

die Distribution *Volumio*, die auch für Linux-Neulinge geeignet und in wenigen Minuten einsatzbereit ist. Dafür benötigen wir lediglich einen Computer und müssen keine weitere Peripherie an den Raspberry Pi anschließen.

### Volumio im Detail

Der in Florenz ansässige Hersteller bietet seine gleichnamige Streamingsoftware in drei verschiedenen Ausführungen an: *Volumio free*,



4 Im Bereich Betriebssystem fügen wir manuell das Volumio-Image über den letzten Menüpunkt hinzu.

*MyVolumio Virtuoso* (28,99 Euro/Jahr) und *MyVolumio Superstar* (69,99 Euro/Jahr). Alle Versionen sind in der Lage, hochauflösende Musik aus unterschiedlichen Quellen abzuspielen. Der wesentliche Vorteil der Premiumvarianten besteht darin, ein oder mehrere

Volumios über das Internet zu steuern, persönliche Einstellungen zu synchronisieren, die HiRes-Audio-Streamingdienste *TIDAL*, *Qobuz* oder *Highresaudio* zu verwenden sowie verlustfrei Audio-CDs zu rippen, wenn man ein externes Laufwerk anschließt.

# Vielfalt MACHT BEI DER REGION HANNOVER Karriere

Hier arbeiten 3.300 Beschäftigte in ganz unterschiedlichen Berufen

**HOCHWERTIGE AUSBILDUNG**

**SICHERER JOB**

auch in MINT-Bereichen

**JETZT ONLINE BEWERBEN:**  
[www.DeineAusbildungMitZukunft.de](http://www.DeineAusbildungMitZukunft.de)  
[www.DaWillIchArbeiten.de](http://www.DaWillIchArbeiten.de)  
 Telefonnummer 0511/616 11011

## Ausbildung

Verwaltungsinformatik – Dualer Studiengang (m/w/d)  
 Fachinformatikerin/ Fachinformatiker - Systemintegration (m/w/d)  
 Malerin/Lackiererin/ Maler/Lackierer – Gestaltung und Instandhaltung (m/w/d)  
 und vieles mehr...

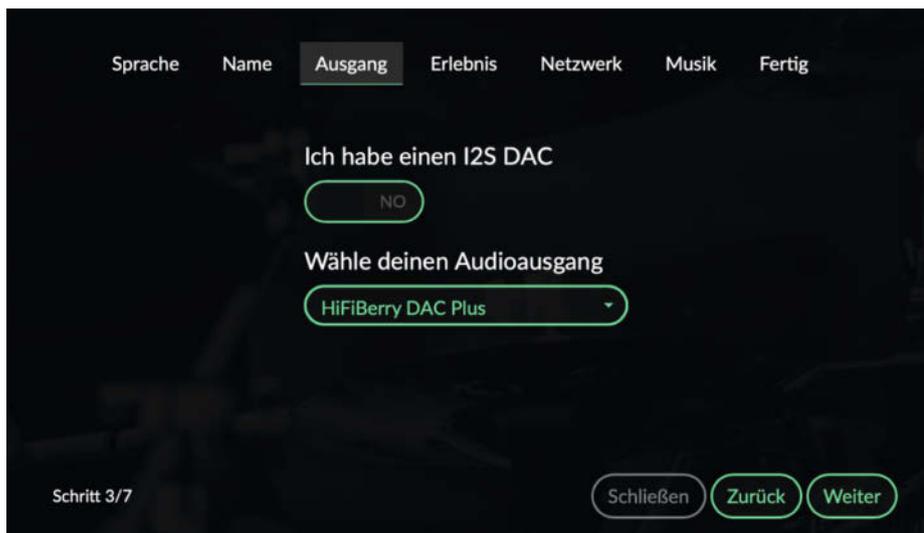
## Jobs

Bauingenieurin/ Bauingenieur (m/w/d)  
 Architektin/ Architekt (m/w/d)  
 und vieles mehr...

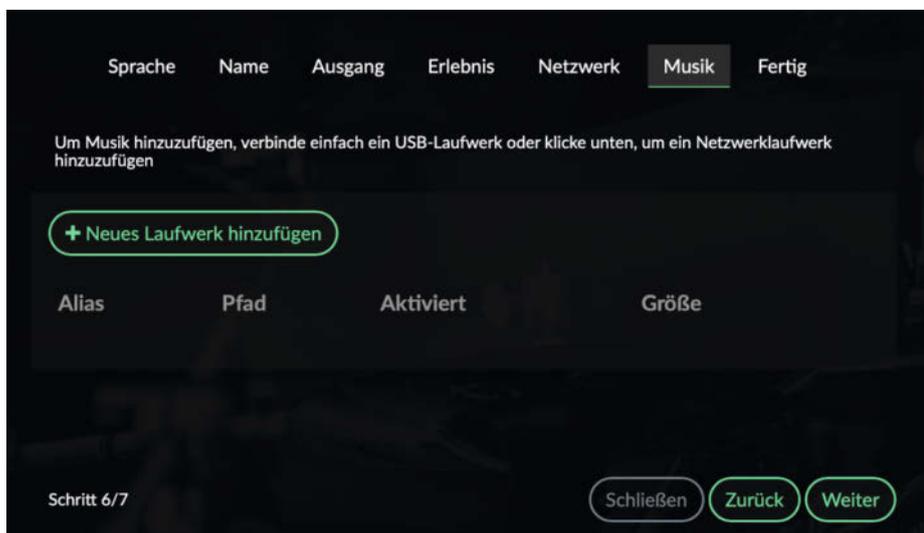
Eigens-stock.adobe.com

## JOB ODER AUSBILDUNG BEI DER REGION HANNOVER – ÖFFENTLICHER DIENST

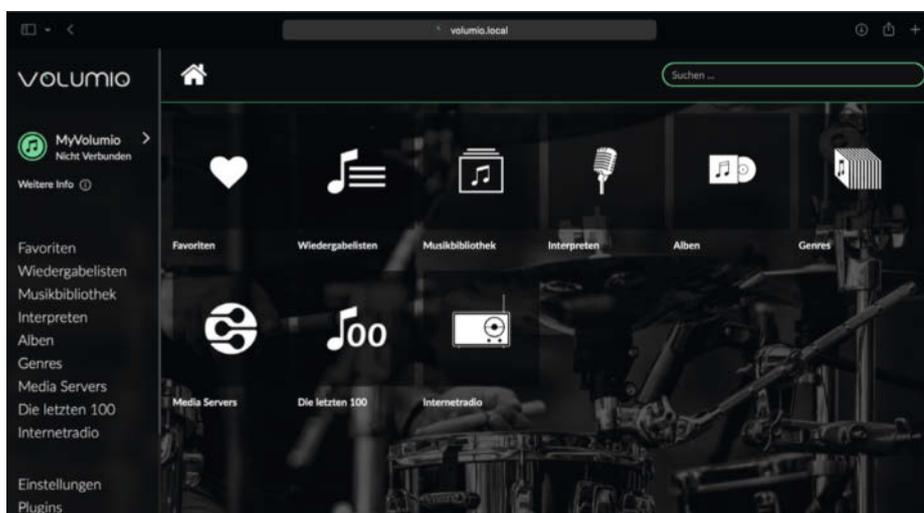
© Copyright by Maker Media GmbH.



5 Damit Musik herauskommt, müssen wir den angeschlossenen DAC auswählen.



(Netz-)Laufwerke lassen sich direkt während der Installation hinzufügen.



6 So sieht die Mediathek aus. Jetzt benötigen wir nur noch Musik.

Praktisch ist, dass sich zahlreiche weitere Plug-ins, wie *Spotify* mit *Spotify Connect* (7) (bislang allerdings ohne Lossless/HiRes-Angebot) bequem über die Benutzeroberfläche (WebUI) installieren und verwalten lassen. Für speziellere Lösungen, wie eine *Youtube*-Anbindung findet sich im Netz eine große Community, die hilfreich mit Installationsanleitungen zur Seite steht. Falls Sie beabsichtigen, *TIDAL* zu nutzen, werden Sie sich über Volumios *TIDAL*-Connect-Integration freuen, die es Ihnen ermöglicht, vergleichbar mit *Spotify Connect*, die Musikwiedergabe direkt über die *TIDAL*-App am Computer oder Smartphone zu steuern – das ist sogar noch komfortabler. Suchen Sie dazu später im Wiedergabefenster der *TIDAL*-App nach dem Lautsprecherbox-Symbol und wählen Sie anschließend Ihren HiRes-Player als Ausgabequelle.

### Installation

Als Erstes laden wir das kostenfreie Volumio (V.2.1917 zum Zeitpunkt des Tests) direkt von der Herstellerseite herunter und flashen es anschließend z.B. mit dem *Raspberry Pi Imager* auf eine MicroSD-Karte (4). Sobald diese im Raspberry Pi sitzt und dieser eingeschaltet wird, erstellt Volumio automatisch einen WLAN-Hotspot namens Volumio, mit dem wir uns über einen Computer oder ein Smartphone verbinden können.

Falls sich nicht automatisch ein Willkommensfenster öffnet, geben wir in dem Browser unserer Wahl die Adresse <http://volumio.local> ein. Das Passwort lautet *volumio2*, sollte aber anschließend in den Einstellungen geändert werden. Folgen wir den Anweisungen des Einrichtungsassistenten, werden wir nach unserem Audio-Ausgabegerät gefragt. Hier sollte der HiFiBerry ausgewählt werden (5). Zum Schluss verbinden wir noch Volumio mit unserem WLAN, da ein Ethernetkabel Störungen im Audiosignal verursachen kann, und schließen das Fenster.

### Bedienung

Auch zukünftig erreichen wir das WebUI über den Browser mit der von uns eingestellten Adresse (falls nicht geändert, weiterhin mit <http://volumio.local>), allerdings ab jetzt über unser WLAN, denn die Hotspot Funktion ist nach der Einrichtung deaktiviert und schaltet sich erst wieder ein, falls unser WLAN nicht erreichbar sein sollte.

Die Benutzeroberfläche von Volumio präsentiert sich übersichtlich und folgt mit einer Kategorisierung nach Alben, Interpreten, Genres und Playlists den gängigen Standards (6). Zudem bietet sich die Steuerung mit dem Smartphone über die hauseigene App aus dem Google Playstore oder Apple AppStore an – ganz bequem vom Sofa aus, einfacher kann eine Fernbedienung kaum sein. Jetzt fehlen nur noch die Inhalte.

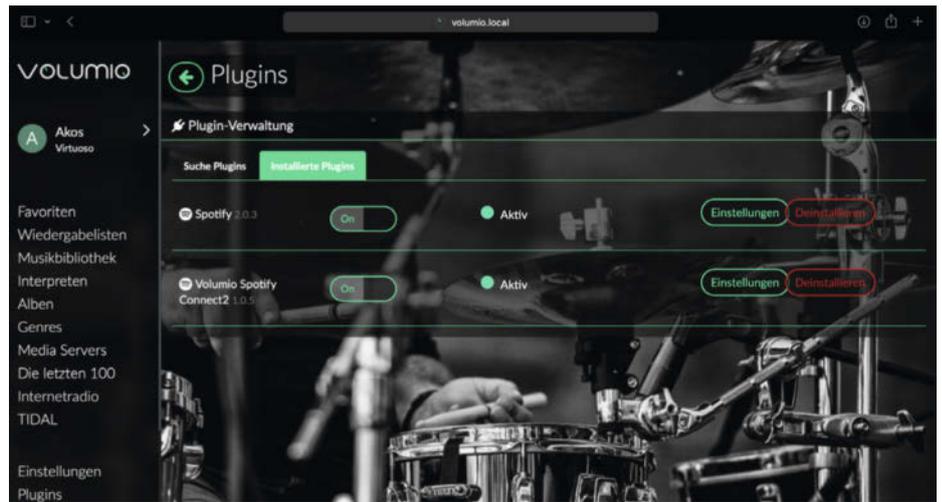
## Es werde Musik

Wer die klangliche Qualität erst einmal testen möchte, ohne in *MyVolumio* und einen Streamingdienst zu investieren, findet auf <http://2l.no> unter *HiRes Test Bench* eine Auswahl hochauflösender Audioaufnahmen, die mithilfe eines Datenträgers direkt an Volumio übergeben werden können.

Um Volumios native Anbindung an *TIDAL* oder *Qobuz* nutzen zu können, schalten wir direkt über das WebUI *MyVolumio* frei, das 15 Tage getestet werden kann. Da wir vorerst „nur“ einen HiRes-Player mit Volumio verwenden, genügt die Option *MyVolumio Virtuoso*. Eine Kreditkarte oder Paypal-Zahlungsinformationen sind für diesen Schritt notwendig. Nach erfolgter Aktivierung müssen wir nur noch über *Einstellungen/Quellen* die Login-Daten für den jeweiligen Dienst eingeben, der daraufhin als Menüpunkt am linken Bildschirmrand erscheint **8** und wir sind fertig. Der HiRes-Player ist jetzt einsatzbereit.

## Alternativer DAC

TIDAL klingt ausgezeichnet mit dem HiFiBerry. Um jedoch dessen volles HiRes-Potenzial auszuschöpfen, benötigen Sie einen externen



**7** Das Spotify-Plugin ist leicht zu installieren.

DAC, der das von TIDAL genutzte MQA-Format (*Master Quality Authenticated*) über die Hardware dekodieren kann **9**. Anstelle des HiFiBerry können Sie z.B. den *iFi ZEN DAC* **10** über USB an den Raspberry Pi anschließen. Er kostet zwar das drei- bis vierfache, ist jedoch universeller einsetzbar – beispielsweise am heimischen PC, um diesen klanglich

„aufzufrischen“ – und konvertiert mit bis zu 384kHz bei 32 Bit.

Weitere sinnvolle Features sind der analoge Lautstärkeregel sowie ein ebenfalls hervorragender Kopfhörerverstärker. Für die garantiert störungsfreie Wiedergabequalität empfiehlt *iFi* das eigens für Audio entwickelte Netzteil *iPower X* zu verwenden, das zudem

# Make:markt

## BÜCHER / ZEITSCHRIFTEN



**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](http://www.dpunkt.de)

## METALLBAU



**MakerBeam: Mini T-Nut Alu-Profil**  
Unbegrenzte Möglichkeiten in Modell- und Prototypenbau  
Das MakerBeam Sortiment:  
- 10mm & 15mm Profile  
- Linearlager, Scharniere, Eckwürfel  
- Halterungen für Servo & NEMA17  
- M3 Schrauben, Nutensteine, Abstandshalter  
[www.makerbeam.de](http://www.makerbeam.de) | [www.chartup.com](http://www.chartup.com)

## MIKROCONTROLLER



**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](http://www.emedia.de)



[christiani.de/arduino-education](http://christiani.de/arduino-education)

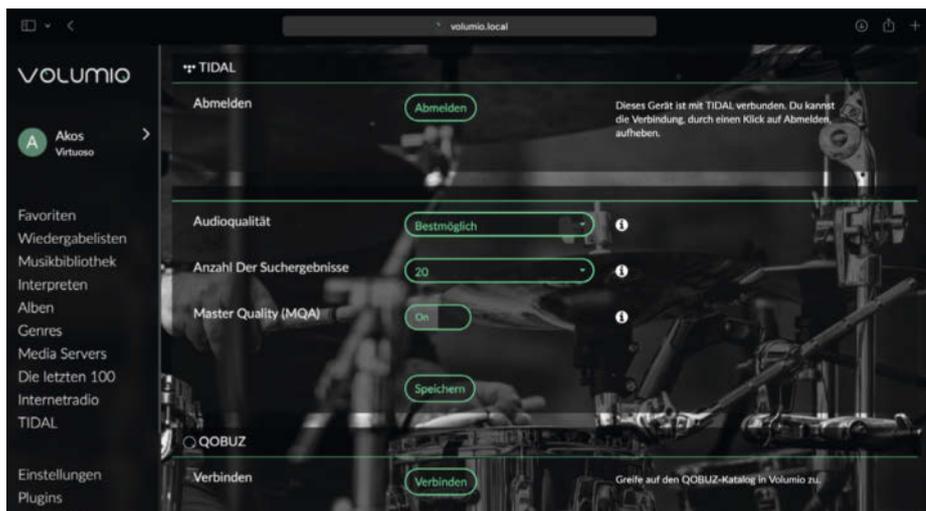
**Arduino® Education – bietet ein durchgängiges Lernkonzept mit Soft-, Hardware und Support**

Leichter Einstieg in die Elektronik und Programmierung mit spannenden Projekten

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



seit 1931



8 TIDAL und Qobuz können mit abonniertem myVolumio Virtuoso über Einstellungen/Quellen konfiguriert werden.

mit ANC (Adaptive Noise Cancellation) Störgeräusche aus der Stromleitung filtert. Die technischen Daten sprechen für sich: ein sehr geringer Rauschabstand, geringste Verzerrungen und ein so linearer Frequenzgang sind in dieser Preisklasse sonst kaum zu finden.

### Klangoptimierung

Auch die Aufstellung unserer Boxen sowie der Raum selbst haben einen Einfluss auf das Klangerlebnis. Wenn wir das meiste aus unserem HiRes-Player herausholen möchten,

sollten wir nicht vergessen, auch das bestehende System zu überprüfen.

Wird ein Lautsprecher zu dicht an einer Wand oder gar in der Ecke eines Raumes aufgestellt, so erfährt der Bass eine Anhebung – die Wiedergabe klingt dumpf und matschig. Ebenfalls führt ein mit Teppichen, Vorhängen und Polstermöbeln gedämpfter Raum durch die starke Absorption der hohen Frequenzen zu einem dumpfen Klangbild. Hat der Raum hingegen viele „akustisch harte Flächen“ wie Fliesenböden, große Fenster oder nackte Betonwände, so kommt es durch die starken Reflexionen im Höhenbereich zu einem hellen, oft als unangenehm wahrgenommenen Klangbild.

Generell sollten die Lautsprecher relativ zum Lieblingshörplatz ein gleichseitiges Dreieck bilden, wobei der Abstand der Lautsprecher zur Wand – falls möglich – mindestens 50cm betragen sollte. Da die höheren Töne gerichtet abgestrahlt werden, stellen Sie am besten die Hochtöner auf Ohrenhöhe auf. Regalboxen gehören übrigens weder ins Regal, noch auf den Boden, sondern sollten mit genügend Wandabstand auf Boxenständern montiert sein. Von dieser Basisausrichtung aus kann nun das Feintuning erfolgen.

Die Devise lautet also: Nicht zufrieden mit dem Klang? Dann zuerst Umstellen ausprobieren!

### Fazit

Für wenig Geld lässt sich eine leicht zu bedienende, universelle und vor allem klanglich überzeugende Wiedergabelösung für digitale Audiodaten bauen. Die Möglichkeit, ohne akustische Einschränkungen bequem vom Sofa aus auf die gesamte Mediathek zuzugreifen, ist zudem ein immenser Komfortgewinn. So bildet der HiRes-Player eine vielversprechende Grundlage zur Erweiterung der bestehenden HiFi-Anlage und zeigt: HiRes-Streaming macht richtig Spaß.

Noch so leise Klänge erscheinen spielerisch und losgelöst aus dem akustischen „Nichts“ – kein Rauschen, kein Brummen, keine Verzerrungen und ein vorbildlich linearer Frequenzgang, bei noch so anspruchsvollem Material. Die Instrumente verteilen sich im Raum, Vocals klingen dynamisch und kristallklar – Details, die bei unseren Versuchen mit MP3s verborgen geblieben sind.

Es ist ein anderes Hören, konzentrierter, genussvoller, bewusster, neugieriger, vielseitiger und manchmal auch anstrengender, wie ein Tag in einer neuen Stadt, in der es viel zu erleben gibt – auf jeden Fall eine Erfahrung, die Sie dazu inspirieren könnte, Musik neu und auch neue Musik zu entdecken.

Wir wünschen Ihnen viel Spaß! Hören Sie Musik!  
—akf

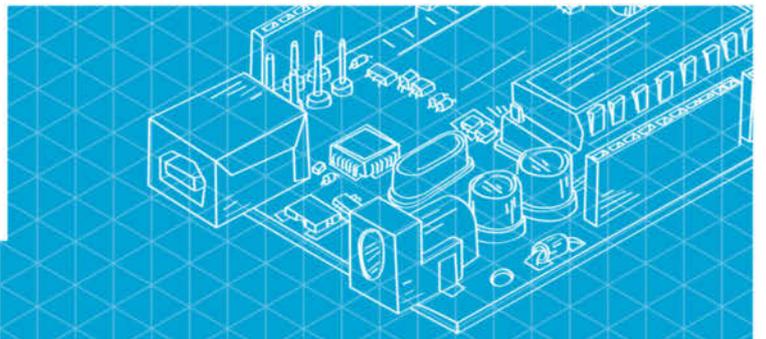


9 Qobuz (oben) spielt in Volumio höhere Samplingraten ab als TIDAL (unten).

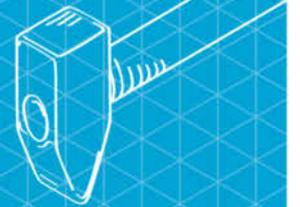


10 Der iFi ZEN DAC kann MQA dekodieren und ist universell einsetzbar.

# Make:



## DAS KANNST DU AUCH!



**GRATIS!**



## 2x Make testen und über 9 € sparen!

### Ihre Vorteile:

- ✓ **GRATIS dazu:** Make: Tasse
- ✓ Jetzt auch im Browser lesen!
- ✓ Zugriff auf Online-Artikel-Archiv\*
- ✓ Zusätzlich digital über iOS oder Android lesen

Für nur **16,10 € statt 25,80 €**

\* Für die Laufzeit des Angebotes.

Jetzt bestellen: [make-magazin.de/miniabo](https://make-magazin.de/miniabo)

# Luftentfeuchter mit Raspi steuern

Mithilfe eines Raspberry Pi und eines Feuchtigkeitssensors läuft  
dieser Luftentfeuchter unter dem Dach ganz automatisch.

von Vladimir Poliakov



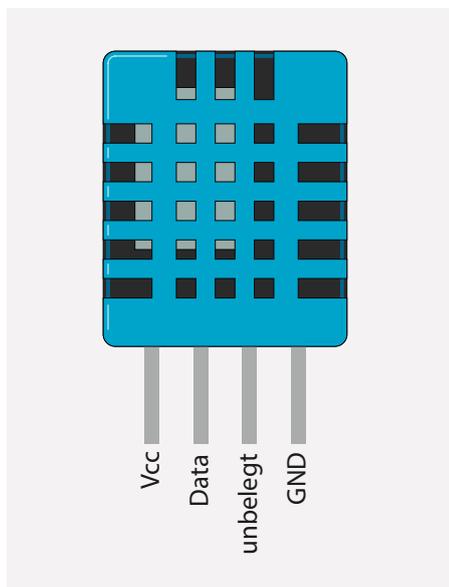
**F**ür unseren Abstellraum unter dem Dach habe ich einen einfachen Luftentfeuchter bei Conrad gekauft. Nach einigen Läufen hoch und runter zum Ein- und Ausschalten des Geräts habe ich beschlossen, seinen Betrieb zu automatisieren.

Von meinen ersten Experimenten mit dem Einplatinenrechner Raspberry Pi vor Jahren hatte ich noch einen Luftfeuchtigkeitssensor DHT11 und das Raspi-Modell 1 in der Zubehörkiste. Damit kann ich bereits die Luftfeuchtigkeit messen und auswerten. Die Idee war dabei, den Luftentfeuchter per Raspberry Pi einzuschalten, wenn die Luftfeuchtigkeit einen kritischen Grenzwert überschritten hat. Nach weiteren Recherchen bin ich auf Raspberry-Pi-kompatible Relais gestoßen. Diese Relais (z. B. das SRD-05VDC-SL-C) ermöglichen die Steuerung von Modulen mit höherer Spannung und können per GPIO von Raspberry Pi bedient werden.

Somit hatte ich alle Puzzleteile beisammen und musste „nur“ alle Komponenten zusammenbauen und ein wenig Code in Python schreiben.

### Bau der Anlage

Der Sensor DHT11 misst auch die Temperatur. Er ist sehr günstig, hat dafür aber nur einen sehr begrenzten Einsatzrahmen: zwischen 20 und 90 Prozent Luftfeuchte, bei Temperaturen zwischen 0 und 50 Grad Celsius. Das reicht für den Abstellraum aber völlig aus. Seine Auflösung beträgt auch nur 1 Grad bzw. 1 Prozent Luftfeuchte. Für genaue Messergebnisse habe ich daher noch eine Kalibrierung im Python-Code eingebaut. Sein Nachfolgemodell, der DHT22, arbeitet da präziser – zwischen 0 und 100 Prozent Luftfeuchte und -40 bis 80 Grad Celsius.



Die Anschlüsse des DHT11

## Kurzinfo

- » Luftentfeuchter automatisieren
- » Steuerung über Raspberry Pi
- » Kontrolle mit Feuchtigkeitssensor

### Checkliste



**Zeitaufwand:**  
2 Stunden



**Kosten:**  
90 Euro

### Werkzeug

- » Seitenschneider
- » Feiner Kreuzschlitzschraubendreher
- » Heißklebepistole oder anderer Klebstoff

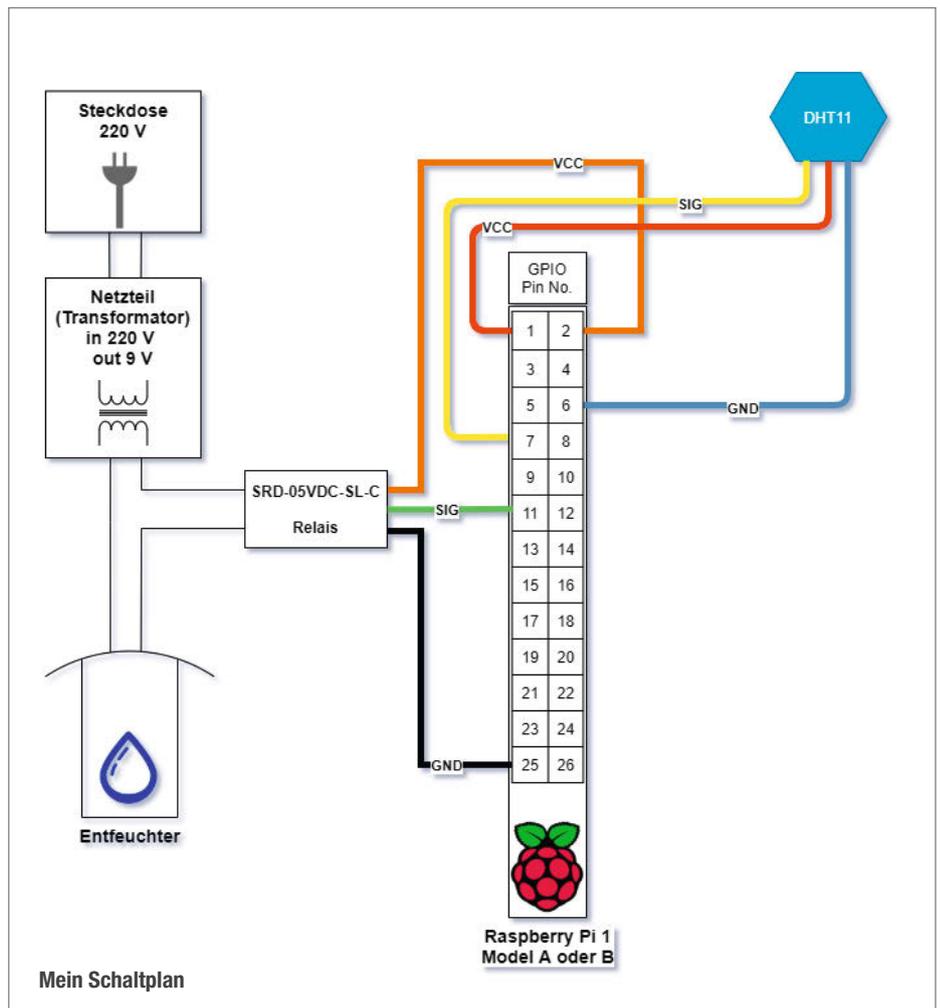
### Mehr zum Thema

- » Dirk Knop, Internet of Things: Sensoren zur Umweltanalyse, online
- » Florian Schäffer, Umweltsensoren für Mikrocontroller, Make 1/19, S. 8
- » Make Elektronik Special inkl. Experimentierset

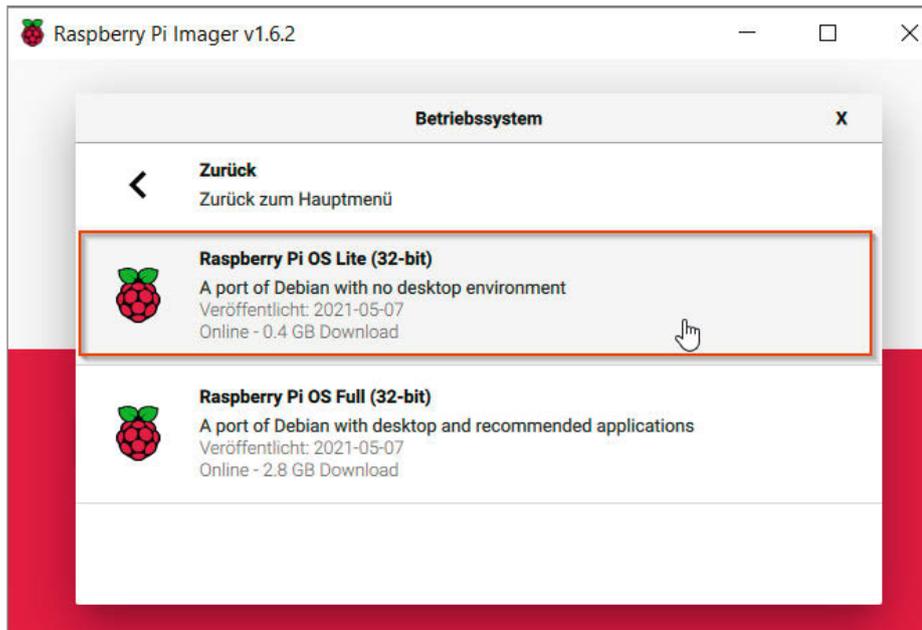
↓ Alles zum Artikel im Web unter [make-magazin.de/xd76](http://make-magazin.de/xd76)

### Material

- » Entfeuchter
- » Relais SRD-05VDC-SL-C
- » Luftfeuchtigkeitssensor DHT11 oder DHT22
- » Raspberry Pi 1 oder andere Version mit SD-Karte und Netzteil
- » 6 Jumperkabel female-female, 20 cm



Mein Schaltplan



Konfiguration mit dem Imager



Raspi und Relais auf der Rückseite des Luftentfeuchters

## Luftentfeuchter-Steuerung

```
import sys
import time
import datetime
import Adafruit_DHT as ADHT
import RPi.GPIO as GPIO

humidity_alert = 65
time_to_sleep = 600

sensor = ADHT.DHT11
sensor_GPIO_Nr = 4
relay_GPIO_Pin = 11
GPIO.setmode(GPIO.BOARD)
GPIO.setup(relay_GPIO_Pin, GPIO.OUT)

if __name__ == '__main__':
    try:
        while True:
            humidity, temperature = ADHT.read_retry(sensor,
            sensor_GPIO_Nr)

            datetime = datetime.datetime.now().strftime(
            '%d.%m.%Y %H:%M:%S')

            if humidity is not None and temperature is not None:
                print('Time={0} Temp={1:0.1f}* Humidity={2:0.1f}%'
                .format(datetime, temperature, humidity))
            else:
                print('Failed to get reading humidity and temperature.
                Try again!')
                sys.exit(1)
            if humidity > humidity_alert:
                GPIO.output(relay_GPIO_Pin, GPIO.HIGH)
            else:
                GPIO.output(relay_GPIO_Pin, GPIO.LOW)

            time.sleep(time_to_sleep)

    except KeyboardInterrupt:
        GPIO.cleanup()
        print('End of GPIO-Script')
    except Exception as error:
        raise error
```

Da sich mein Sensor auf einer 3-Pin-Modulplatine befindet, konnte ich ihn direkt an die Pinleiste des Raspberry anschließen. Eigentlich hat der Sensor vier Pins und erfordert einen 4,7k- oder 10k-Widerstand zwischen der Datenleitung und VCC (siehe Bild). Die Pinbelegung der Module ist nicht einheitlich, aber meist auf der Platine abgedruckt. Sonst hilft ein Blick ins Datenblatt. Weitere Anleitungen finden Sie online, siehe Link in der Kurzinfo.

Relais sind Schalter, die mit einem kleinen Steuerstrom (wie von einem Raspberry Pi) einen großen Verbraucher wie den Luftentfeuchter schalten. Als Raspi-kompatible Boards gibt es sie in verschiedenen Ausführungen, bis hin zu 16-Kanal-Modulen, um entsprechend viele Lasten zu steuern. Für meinen Zweck genügte ein Relais, das direkt an die Pins des Raspberry Pi angeschlossen wird. Mittels eines Impulses niedriger Spannung vom Raspi wird der Relais-Schalter umgelegt und der Luftentfeuchter, obwohl er mit deutlich höherer Spannung läuft, dadurch eingeschaltet. Die Stromkreise bleiben dabei voneinander getrennt und es besteht kein Risiko, dass der Raspberry Pi durchbrennt.

Der Schaltplan zeigt, welche Pins des Raspberry Pi mit welchen Pins des Luftfeuchtigkeitssensors und des Relais verbunden werden. Die Verkabelung dauert nur einige Minuten. Ein Kabel vom Luftentfeuchter habe ich durchtrennt und das Relais zwischengeschaltet. Der Raspi steckt in einem handelsüblichen Gehäuse, das ich mit der Heißklebepistole an der Rückseite vom Luftentfeuchter befestigt habe. Die Relaisplatine habe ich ebenfalls auf das Raspi-Gehäuse geklebt. Am

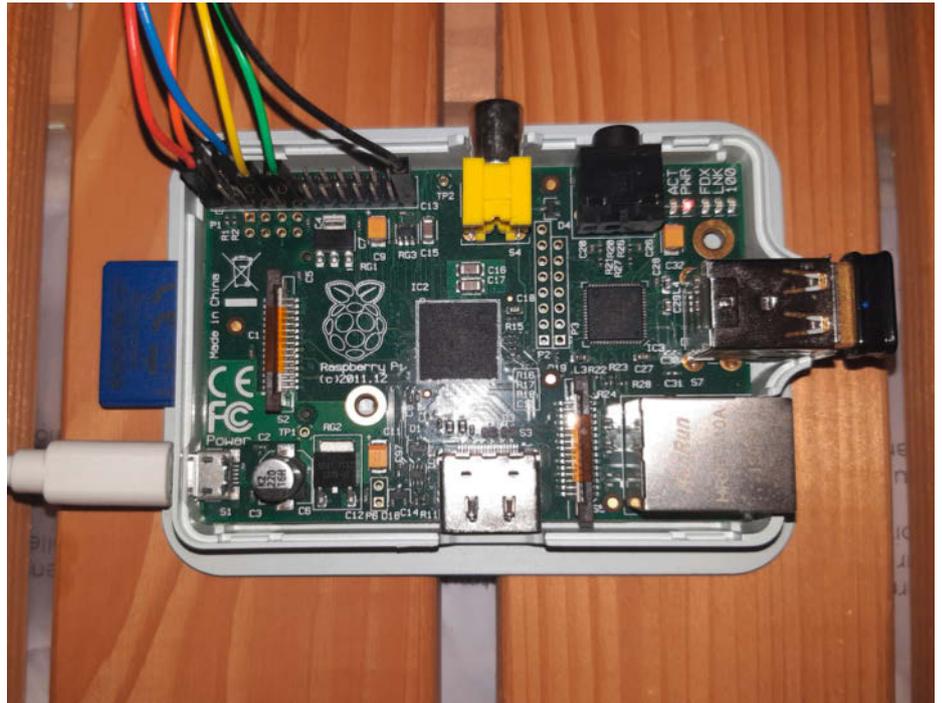
Luftentfeuchter ist oben eine Öffnung zum Einsaugen der feuchten Luft vorhanden, deren Strömung die Messergebnisse des Sensors beeinflussen kann. Deswegen habe ich den Sensor oben an einem Holzstab, genauer gesagt einem Esstäbchen, platziert. Für die Befestigung habe ich auch Heißkleber verwendet.

Nachdem der Luftentfeuchter und der Sensor am Raspberry Pi angeschlossen waren, habe ich die Steuerungssoftware installiert und getestet.

### Installieren der Software

Falls der Raspberry Pi neu aufgesetzt wird, sollte zunächst das Betriebssystem auf einer SD-Karte installiert und konfiguriert werden. Standardmäßig geht das am schnellsten mit dem *Raspberry Pi Imager*. Er hilft nicht nur, die SD-Karte zu formatieren, sondern auch den SSH-Zugriff via WLAN zu konfigurieren. Nach dem Einstecken der SD-Karte und dem Start des Programms muss das Betriebssystem ausgewählt werden. Für dieses Projekt reicht die *OS-Lite*-Variante ohne Desktop aus. Anschließend für die Konfiguration `Ctrl+Shift` und `X` drücken, in den Optionen *Enable SSH* und *Configure wifi* aktivieren und die eigenen Zugangsdaten setzen. Außerdem kann man eine Zeitzone auswählen – sofern sie nicht passend voreingestellt ist – und das Tastaturlayout auf *de* (für Deutsch) ändern. Zum Schluss nur noch die SD-Karte auswählen und bespielen. Nun ist der Raspberry Pi einsatzbereit und via PuTTY bzw. WinSCP erreichbar.

Der nächste Schritt ist das Aktualisieren des Betriebssystems und der Pakete, damit alles auf dem aktuellen Stand ist:



Am Raspberry Pi sind nur wenige Anschlüsse an der Pinleiste nötig.

```
sudo apt-get update
sudo apt full-upgrade
```

Beim ausgelieferten Betriebssystem (in meinem Fall *Raspbian GNU/Linux Version 10*) war *Python 2.7.x* bereits mit dabei, was für den Zweck des Projektes völlig ausreichend ist. Allerdings musste der *pip*-Installer nachinstalliert werden. Das Tool wird zum Aktualisieren und Installieren der Python-Pakete benötigt. Das geht mit einer weiteren Zeile im PuTTY-Terminal:

```
sudo apt-get install python-pip
```

Ist das *pip*-Tool erfolgreich installiert, kann auch die *Adafruit-DHT*-Bibliothek leicht aus dem Repository via *pip* installiert werden:

```
sudo pip install Adafruit_DHT
```

Somit sind alle notwendigen Updates und Vorbereitungen durch und man kann mit dem Testen des Sensors und des Relais anfangen.

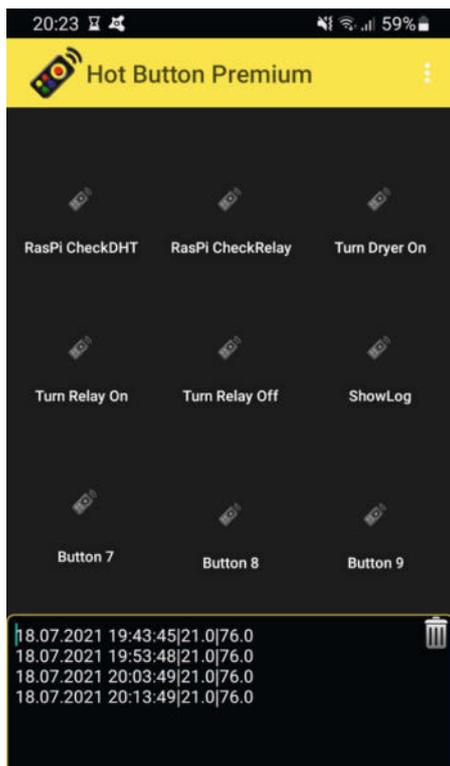
Es gibt **10** Arten von Menschen. iX-Leser und die anderen.



3x als Heft



Jetzt Mini-Abo testen:  
3 Hefte + Bluetooth-Tastatur nur 16,50 €  
[www.ix.de/testen](http://www.ix.de/testen)



Mobil kann ich die Skripte auch per Knopfdruck starten.

## Programmierung

Sinnvoll ist, als Erstes auf dem Raspberry Pi einen Projekt-Ordner im Home-Ordner zu erstellen, in dem alle Python-Skripte sowie die Logs gespeichert werden:

```
mkdir GPIO_Dryer
```

Das bringt Ordnung in die Verzeichnisstruktur des Projektes. Die Shell-Steuerungsskripte können direkt im Home-Verzeichnis bleiben. Alle Skripte zum Prüfen der Luftfeuchtigkeit via GPIO des Raspberry Pi und zum Steuern des Luftentfeuchters habe ich in meinem GitHub-Repository bereitgestellt (siehe Link in der Kurzinfo). Ich empfehle aber, zuerst die Beispiele der Adafruit-DHT-Bibliothek herunterzuladen und anzuschauen. Das geht mit einem `git`-Befehl im PuTTY-Terminal, da das Tool beim Raspbian Linux bereits installiert ist:

```
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
```

Den kompletten Quellcode zu zeigen würde den Rahmen des Artikels sprengen, deswegen beschreibe ich hier nur das Grundgerüst.

Nach dem Import der Bibliotheken setze ich einen Schwellenwert für die Luftfeuchtigkeit (`humidity_alert`) und eine Pause in Sekunden (`time_to_sleep`). Anschließend wird das DHT11-Signal auf dem GPIO 4 (Pin 7) ausgelesen. Ich nutze außerdem das GPIO-Layout, um direkt die Pin-Nummern verwenden zu können. Das Relais ist am GPIO 17 (Pin 11) angeschlossen und wird als Output gesetzt.

In der `if`-Schleife werden nun dauerhaft Luftfeuchtigkeit und Temperatur via DHT11-Sensor gemessen und die Zeit ermittelt. Falls die Luftfeuchtigkeit höher als der Schwellenwert von 65 Prozent ist, wird das Relais und somit der Luftentfeuchter eingeschaltet. Nach 10 Minuten wird die Luftfeuchtigkeit wieder gemessen. Ist der Wert unter den Schwellen-

wert gefallen, wird das Relais wieder ausgeschaltet. Den Code zum Ermitteln der Luftfeuchtigkeit habe ich aus dem Beispiel der Adafruit-DHT-Bibliothek genommen.

## Inbetriebnahme

Damit ich mich nicht jedes Mal via PuTTY zum Starten, Stoppen und Auswerten der Python-Skripte anmelden muss, habe ich ein paar Shell-Steuerungsskripte im Home-Verzeichnis meines Raspberry Pi erstellt. Sie sind denkbar einfach gehalten, wie dieses Skript zum Start des Kontrollprogramms:

```
cd /home/pi/GPIO_Dryer
sudo ./controlDryer.py
exit
```

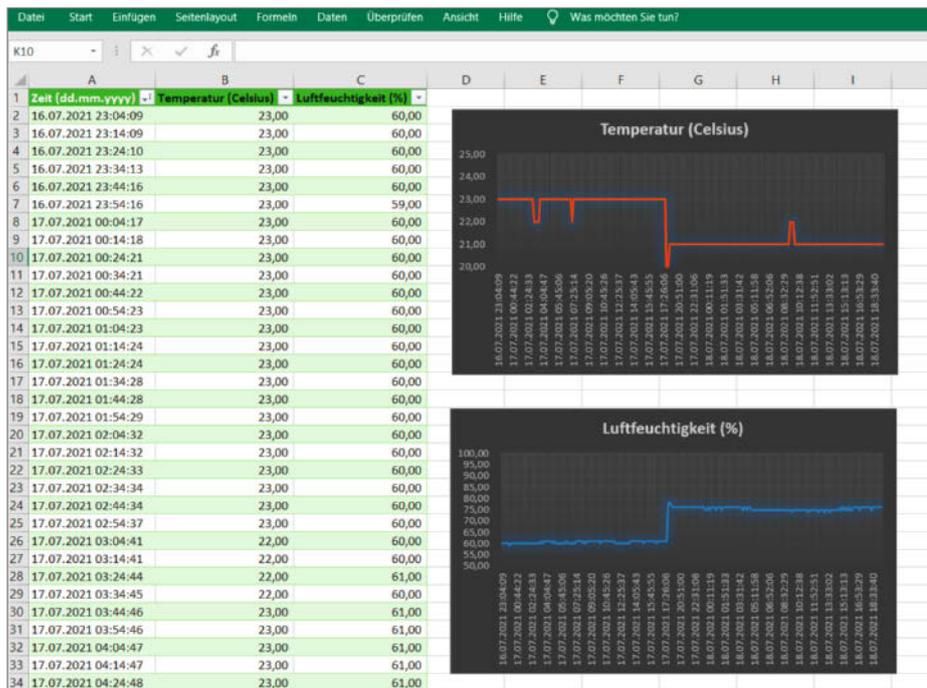
Mit der App *Hot Button Premium SSH Widget* habe ich schließlich Verknüpfungen zu den Skripten auf meinem Handy angelegt. Leider ist die App inzwischen nicht mehr über den *Google Play Store* erhältlich. Eine einfache Alternative zur mobilen Steuerung ist die Kommandozeilen-App *ConnectBot*. Sicherlich gibt es andere, elegantere Lösungen, aber für die ganz einfachen Auswertungen und für das Prüfen der Anwendung reicht es vollkommen aus.

In meiner vollständigen Version des Programms `controlDryer.py` (siehe GitHub) werden noch die Temperatur und die Luftfeuchtigkeit protokolliert. Logischerweise können die Daten nach der Erhebung ausgewertet und analysiert werden. Dazu gibt es verschiedene Lösungen von *Excel* (siehe Bild) oder anderen Tabellenkalkulationsprogrammen wie *LibreOffice* auf dem lokalen Rechner bis zu Cloud-Lösungen, wie etwa Microsofts Cloud-Plattform *Azure*.

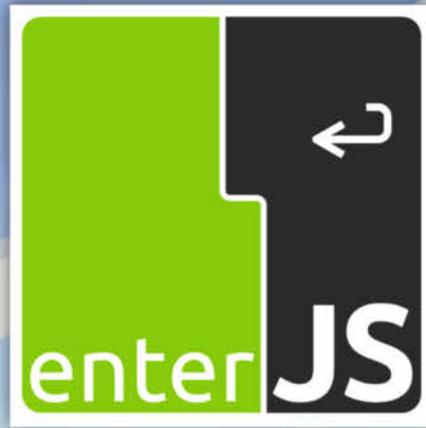
Um alles ganz genau zu wissen, habe ich für die Berechnung der Sensorwerte noch Korrekturfaktoren eingefügt. Dazu vergleiche ich regelmäßig die Messwerte mit einem analogen Feuchtigkeitsmessgerät, das im gleichen Raum steht, und passe, wenn nötig, den Korrekturfaktor an – etwa, wenn ich den Wasserbehälter des Entfeuchters leere. Das ließe sich sicher besser lösen, aber irgendwie bin ich noch nicht dazu gekommen. Eine Alternative ist der Einsatz des DHT22, der genauer misst.

## Was kommt als Nächstes?

Bevor der alte Raspberry Pi sinnlos rumliegt, sollte er lieber zum Einsatz kommen. Und das tut er jetzt auch. Für meinen Zweck ist diese Steuerung des Luftentfeuchters komplett ausreichend. Ich kann vom Sofa aus die Werte anschauen und das Skript starten. Das Einzige, was noch schön wäre, wäre zu ermitteln, ob der Wasserbehälter voll ist und geleert werden soll. In diesem Fall wäre eine Benachrichtigung hilfreich. Vielleicht werde ich diese Herausforderung demnächst angehen. —hch



Auswertung mit Excel



# Die Konferenz für Enterprise-JavaScript

22. und 23. Juni 2022 - Darmstadt

Call for Proposals  
noch bis  
**17.1.2022**  
Jetzt  
einreichen!

[www.enterjs.de](http://www.enterjs.de)

Veranstalter



 heise Developer

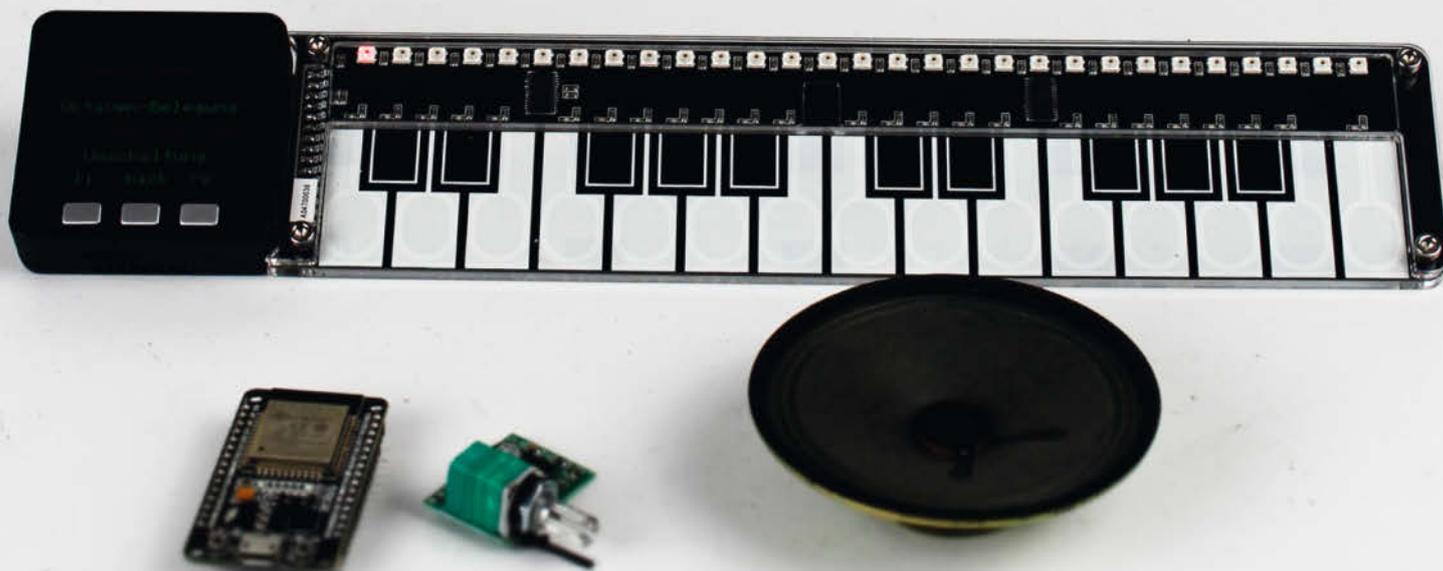
© Copyright by Maker Media GmbH.

 dpunkt.verlag

# ESP32-Orgel mit Piano-Tastatur

Die Piano-Tastatur von M5Stack mit ihren schwarzen und weißen Tasten provoziert geradezu den Bau eines Musikinstruments. Das geht ganz einfach mit dem Basis-Modul von M5Stack, aber auch mit jedem anderen ESP32-Board. Mit der etwas aufgebohrten Beispiel-Software des Herstellers beherrscht die kleine Orgel sogar vier umschaltbare Oktaven und Halbtöne.

von Heinz Behling



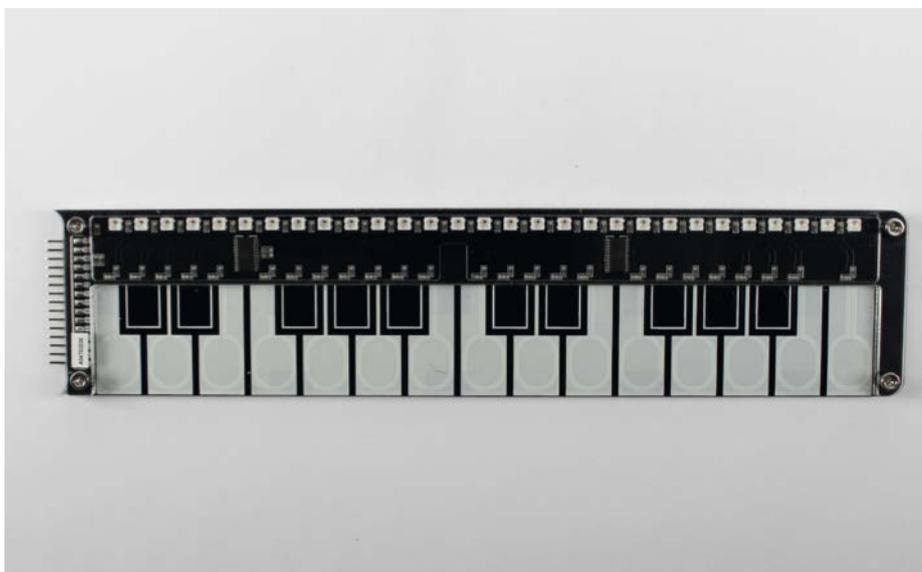
Um es gleich vorwegzunehmen: Super-Soundqualität kann man hier nicht erwarten, es ist eher eine Konkurrenz zur Reißzwecken-Orgel aus der Make 1/16. Wer besseren Klang möchte, muss zur Tonerzeugung hochwertigere Hardware benutzen. Für diesen Fall erfährt man hier aber, wie man die Piano-Tastatur anschließt und per Software ausliest.

## Tastatur-Ausstattung

Die kleine Klaviatur **1** hat 25 Tasten, aufgeteilt in 15 weiße und 10 schwarze (für die Halbtöne) wie bei einem Klavier. Die Anordnung entspricht zwei Oktaven mit C beginnend. Allerdings sind die Tasten (genauer gesagt sind es Touch-Sensoren) schmaler als bei einem normalen Tasteninstrument. Schlanke Finger sind daher von Vorteil.

Die 25 Taster sitzen an zwei Controller-ICs, die per I<sup>2</sup>C-Bus mit ESP32- oder anderen Mikrocontrollerboards kommunizieren (Adressen: 0x6A und 0x7A). Zusätzlich sitzen auf dem Board noch 29 RGB-LEDs (Typ WS2812), die frei adressierbar einzeln angesteuert werden können. Das Modul braucht 3,3V (für die Tasten) und 5V (für die LEDs) Versorgungsspannung. Leiterbahnen und Elektronik-Bauteile auf der Ober- und Unterseite liegen unter Plexiglas geschützt.

Alle Anschlüsse liegen an einer 15poligen Stiftleiste an der linken Seite der Tastatur an (siehe Kasten *Kontaktbelegung*). Diese Leiste passt direkt an das Core-Modul von M5Stack **2**. Das Zusammenbauen beschränkt sich also auf das Einstecken. Deshalb bringe ich hier keine Bauanleitung für die Modul-Version. Wer dennoch das M5-Core-Modul verwenden möchte, muss nach dem Zusammenstecken nur die Software installieren.



**1** Die Klaviatur hat leider nur 13mm-breite Tasten.

## Kurzinfo

- » Piano-Tastatur mit I<sup>2</sup>C-Anschluss und Touch-Sensoren
- » Bauanleitung mit M5-Core-Module und ESP 32
- » Software mit Halbtönen und Registerumschaltung

## Checkliste



**Zeitaufwand:**  
1 bis 2 Stunden



**Kosten:**  
ab 30 Euro

## Mehr zum Thema

- » Heiko Dobslaw, Richard Kurz, Rainer Specker, Reißzwecken-Orgel, kostenloser Online-Artikel aus Make 6/16, S. 12

Alles zum Artikel  
im Web unter  
[make-magazin.de/x6kb](https://make-magazin.de/x6kb)

## Material

- » M5Stack Piano-Tastatur
- » M5-Core-Modul (Basic-Version) mit passendem Netzteil oder
- » ESP-32-Board (zum Beispiel ESP32-Node-MCU)
- » Verstärkermodul (5V Betriebsspannung)
- » Kleinlautsprecher
- » 3 Taster
- » Steckleiste 15polig, Rastermaß 2,54mm
- » Schalllitze 0,14mm<sup>2</sup>

## Werkzeug

- » Lötkolben

Aber mit etwas Schalllitze lassen sich auch andere ESP32- oder ähnliche Boards anschließen. Der Bau solch einer Mini-Orgel ist etwas aufwändiger und wird daher hier genau beschrieben. In dieser Anleitung wird es eine ESP32-NodeMCU sein. Auch andere ESP-32-Boards sind möglich, dann müssen aber eventuell die benutzten Port-Nummern für die Taster angepasst werden.

## Mangelhafte Beispielsoftware

Die ersten Experimente wurden mit der an einem M5-Core-Modul angesteckten Piano-Tastatur unternommen. Das Modul wird per mitgeliefertem USB-C-Kabel mit dem USB-Anschluss des PCs verbunden. Darüber wird zum einen der Akku des Moduls geladen, zum



**2** Das M5-Core-Modul in der Basic-Version mit 16MB Flash-Speicher, 320 x 240 Pixel-Farbdisplay, WLAN, Bluetooth BLE, Akku mit Ladefunktion über USB-C-Buchse, drei programmierbare Taster, Mikro-SD-Kartenslot, 10 ADC- und 2 DAC-Wandler und Lautsprecher

```

184
185 if ((data0 & 0x10) == 0x10) {
186     Set_Bit(key[0], 0);
187     Leds[0] = CRGB::Red;
188     // pixels.setPixelColor(0, pRed);
189     M5.Speaker.tone(NOTE_DL1, 20);
190 } else {
191     Leds[0] = CRGB::Black;
192     // pixels.setPixelColor(0, pBlack);
193     Clr_Bit(key[0], 0);
194 }
195 if ((data0 & 0x20) == 0x20) {
196     Set_Bit(key[0], 1);
197     Leds[1] = CRGB::Red;
198     // pixels.setPixelColor(1, pRed);
199     M5.Speaker.tone(NOTE_DL1, 20);
200 } else {
201     Leds[1] = CRGB::Black;
202     // pixels.setPixelColor(1, pBlack);
203     Clr_Bit(key[0], 1);
204 }
    
```

anderen erfolgt darüber auch die Programmierung.

Vom Hersteller gibt es auf Github eine Beispielsoftware für dieses Hardware-Duo, die zwei Oktaven (von c<sup>1</sup> bis c<sup>2</sup>) spielen kann. Bei einem Tastendruck leuchtet jeweils die darüberliegende LED auf. Sogar Polyphonie (also das gleichzeitige Spielen mehrerer Töne) ist möglich. Diese Software wurde mit der Arduino-IDE auf das Core-Modul übertragen. Dazu musste dort unter *Datei* und *Voreinstellungen* als *zusätzliche Boardverwalter-URL* diese Zeile hinzugefügt werden:

[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)  
[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)

Außerdem musste die Bibliothek *M5Stack* mit dem *Bibliotheksverwalter* hinzugefügt werden. In aktuellen Versionen der Arduino IDE wird die Bibliothek bereits unter *Bibliothek verwalten/Bibliothek einbinden* zur Installation angeboten. In älteren IDE-Versionen ist zuvor noch der Download der ZIP-Datei von Github und die Einbindung über den Punkt *ZIP-Bibliothek hinzufügen* nötig. Die Download-Adresse finden Sie über den Kurzinfo-Link. Auf jeden Fall ist der Download der Dateien auf den Github-Seiten von *M5-ProductCodes* notwendig (Adresse ebenfalls über Kurzinfo-Link erhältlich). Darin findet man unter *App* und *Piano* die Datei *M5Piano.ino*, das schon genannte Beispielprogramm.

In der Arduino-IDE stellt man unter *Werkzeuge* und *Board* dann *M5Stack-Core-ESP32* sowie die richtige Portnummer ein (für gewöhnlich die höchste angezeigte) und lässt das Programm kompilieren und übertragen.

Schnell stellte sich aber heraus, dass die Beispielsoftware eine große Schwäche hatte: Sie spielte keine Halbtöne. Wenn man eine der schwarzen Tasten drückte, ertönte stattdessen der Ton, der zur jeweiligen weißen Taste links daneben gehört, statt Cis ertönte so ein C. Da musste also nachgearbeitet werden.

### Software-Weiterentwicklung mit dem Core-Modul

Da das Halbton-Problem ohnehin gelöst werden sollte, wurde auch gleich, ähnlich einer Orgel, eine Registerschaltung in den Software-Plan aufgenommen. Damit sollte man per Tastendruck die Belegung der beiden Tastatur-Hälften jeweils um eine Oktave tiefer beziehungsweise höher legen können.

Ein Blick in die Software (zugegeben, mit etwas Suchen) fand schnell die Ursache des Halbton-Mangels, hier am Beispiel der beiden linken Tasten c und cis gezeigt: Beide werden zwar mit einer eigenen Routine (Zeilen 185 bis 193 für die erste, Zeilen 195 bis 203 für die zweite Taste usw. ③) erkannt, aber es wird von beiden Routinen derselbe in der Konstanten *Note\_DL1* gespeicherte Ton gespielt.

#### 3 Zwei Tastenabfragen, derselbe Ton

```

130 #define NOTE_D0 -1
131 #define NOTE_D1 524
132 #define NOTE_D2 588
133 #define NOTE_D3 660
134 #define NOTE_D4 698
135 #define NOTE_D5 784
136 #define NOTE_D6 880
137 #define NOTE_D7 988
138
139 #define NOTE_DL1 261
140 #define NOTE_DL2 293
141 #define NOTE_DL3 329
142 #define NOTE_DL4 349
143 #define NOTE_DL5 392
144 #define NOTE_DL6 440
145 #define NOTE_DL7 494
146
147 #define NOTE_DH1 1048
148 #define NOTE_DH2 1176
149 #define NOTE_DH3 1320
150 #define NOTE_DH4 1396
    
```

4 In den Notendefinitionen fehlen die Halbtonfrequenzen.

```

130 #define NOTE_DL1 261
131 #define NOTE_DL15 277
132 #define NOTE_DL2 293
133 #define NOTE_DL25 311
134 #define NOTE_DL3 329
135 #define NOTE_DL4 349
136 #define NOTE_DL45 370
137 #define NOTE_DL5 392
138 #define NOTE_DL55 415
139 #define NOTE_DL6 440
140 #define NOTE_DL65 466
141 #define NOTE_DL7 494
142
143 #define NOTE_D1 524
144 #define NOTE_D15 554
145 #define NOTE_D2 588
146 #define NOTE_D25 622
147 #define NOTE_D3 660
148 #define NOTE_D4 698
149 #define NOTE_D45 740
150 #define NOTE_D5 784
151 #define NOTE_D55 831
152 #define NOTE_D6 880
153 #define NOTE_D65 932
154 #define NOTE_D7 988
155 #define NOTE_DH1 1048
    
```

5 Der Halbton zwischen *NOTE\_DL1* und *NOTE\_DL2* wurde *NOTE\_DL15* genannt. So ging es weiter für zwei Oktaven.

Das liegt daran, dass in den Zeilen 189 für die weiße und in Zeile 199 für die schwarze Taste derselbe Befehl steht:

```
M5.Speaker.tone(NOTE_DL1, 20);
```

Also suchte ich die Notendefinitionen und fand sie ab Zeile 130 **4**.

Da hatte es sich der Autor des Programms etwas einfach gemacht, denn dort waren keine Definitionen für die Halbtöne enthalten. Ich suchte sie schnell in Wikipedia (siehe Kurzinfo-Link) und ergänzte die Definitionen für beide Oktaven. Die Halbtöne erhielten entsprechende Namen wie *NOTE\_DL15* **5**.

Entsprechend änderte ich auch die Abspielbefehle für die Halbton-Tasten.

```
M5.Speaker.tone(NOTE_DL15, 20);
```

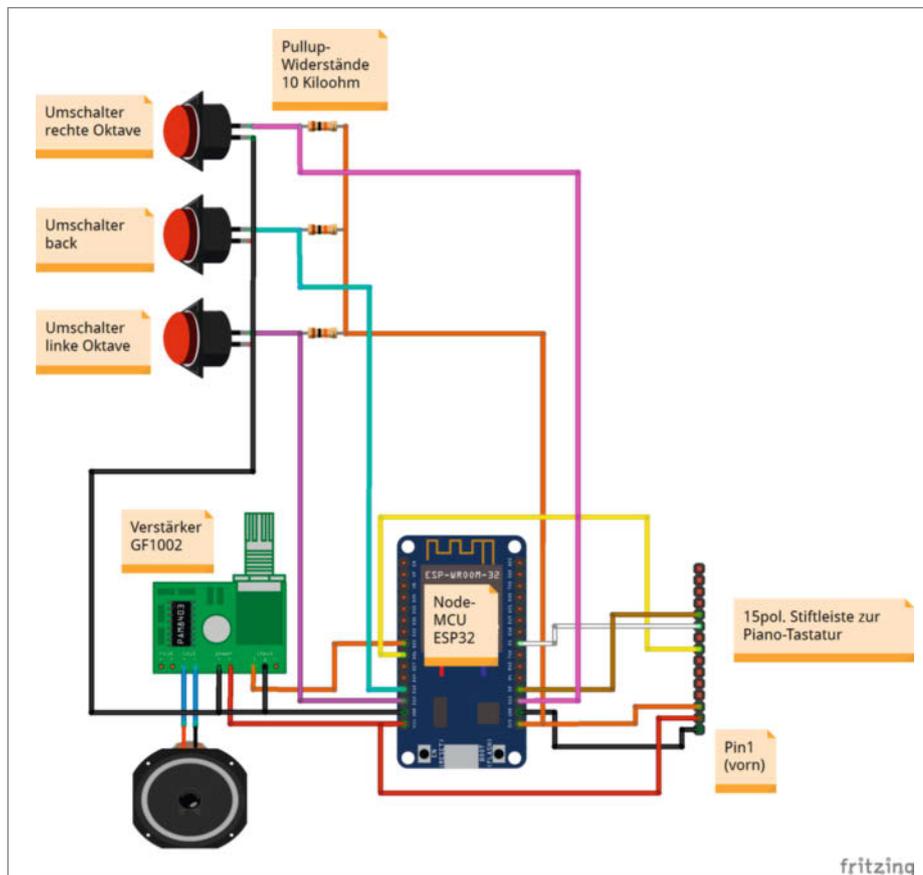
Nach Kompilieren und Hochladen war die Orgel halbtönföähig.

### Registerschaltung

Bei der Registerauswahl, also dem Umschalten der jeweils zu spielenden Oktave, machte ich mir die Tatsache zunutze, dass das Verhöltnis der Tonfrequenzen zweier aufeinander folgenden Oktaven 1:2 betröagt. Die Frequenz von  $c^0$  ist die Hölftle von  $c^1$ . Und  $c^2$  hat die doppelte Frequenz von  $c^1$ . Ich musste also nur bei den Abspielbefehlen die jeweilige Notenkonstante

```
392 //Registerwahl
393 if (M5.BtnA.pressedFor(10, 200)) {
394     FaktorLow = 0.5;
395     ColorLow = (CRGB::Green);
396 } else if (M5.BtnB.pressedFor(10, 200)) {
397     FaktorLow = 1;
398     FaktorHigh = 1;
399     ColorLow = (CRGB::Red);
400     ColorHigh = (CRGB::Red);
401 } else if (M5.BtnC.pressedFor(10, 200)) {
402     FaktorHigh = 2;
403     ColorHigh = (CRGB::Blue);
404 } else if (M5.BtnB.wasReleasedFor(700)) {
405     M5.Lcd.setCursor(0, 0);
406 }
407 if (FaktorLow == 1) {
408     M5.Lcd.setCursor(10, 128);
409     M5.Lcd.setTextColor(TFT_RED, TFT_BLACK);
410     M5.Lcd.printf("c1 - h1");
411 }
412 if (FaktorLow == 0.5) {
413     M5.Lcd.setCursor(10, 128);
414     M5.Lcd.setTextColor(GREEN, TFT_BLACK);
415     M5.Lcd.printf("c0 - h0");
416 }
417 if (FaktorHigh == 1) {
418     M5.Lcd.setCursor(180, 128);
419     M5.Lcd.setTextColor(TFT_RED, TFT_BLACK);
420     M5.Lcd.printf("c2 - c3");
421 }
422 if (FaktorHigh == 2) {
423     M5.Lcd.setCursor(180, 128);
424     M5.Lcd.setTextColor(BLUE, TFT_BLACK);
425     M5.Lcd.printf("c3 - c4");
426 }
```

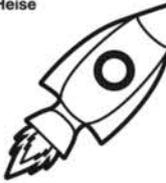
**6** Die Registerumschaltung ändert den Tonhöhenfaktor *FaktorLow* für die untere (linke) und *FaktorHigh* für die obere (rechte) Tastaturhälfte. Außerdem wird auf dem Display des Core-Moduls die jeweilige Oktave angezeigt.



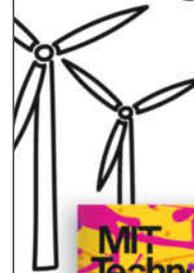
**7** Schaltplan des Pianos mit ESP32-Board

# MIT Technology Review

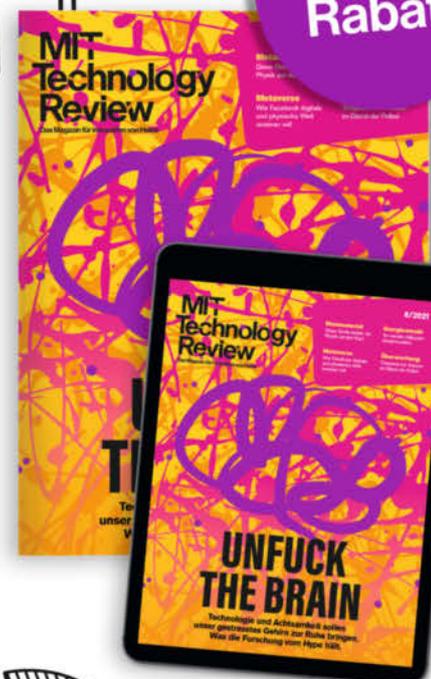
Das Magazin für Innovation von Heise



Die Welt verstehen, wie sie morgen sein wird.

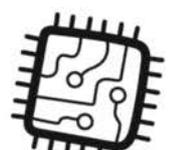


35% Rabatt



2 Ausgaben inklusive Prämie nach Wahl:

[mit-tr.de/testen](https://mit-tr.de/testen)



✉ [leserservice@heise.de](mailto:leserservice@heise.de)

☎ +49 541/80 009 120

🌐 [mit-tr.de/testen](https://mit-tr.de/testen)



8 Das Core-Modul zeigt auf dem Display die aktuelle Tastaturbelegung an.

noch mit einem Faktor von 0,5, 1 oder 2 multiplizieren. Die Registerumschaltung stellt dann mithilfe der drei Taster des Core-Moduls (an den GPIO-Pins 37, 38 und 39) lediglich die Werte dieser Faktoren entsprechend ein. Das geschieht in meinem Programm in den Zeilen 392 bis 426 6. Um die Pinnummern muss man sich hier nicht kümmern, da die durch die Bibliothek M5Stack schon definiert wurden.

Daneben wird in den Variablen *ColorLow* und *ColorHigh* die Farbe festgelegt, in der die LEDs über den Tasten jeweils aufleuchten, sobald man einen Ton spielt: Grün steht für die Oktave ab  $c^0$ , rot für den ursprünglichen Zustand und blau für die Oktave ab  $c^3$ . Wenn Sie die Umschaltung sowie die Anzeige einmal genau untersuchen möchten, schauen Sie sich die Kommentarzeilen im Programm an.

Die Bedienung der Register-Umschaltung ist einfach: Mit dem linken Taster des Core-Moduls können Sie die linke Hälfte der Klaviatur jeweils eine Oktave tiefer (c0 bis h0, LEDs leuchten grün) und wieder zurück (c1 bis h1, LEDs leuchten rot) einstellen. Die rechte Taste kann entsprechend die rechte Tastaturhälfte

um einer Oktave höher (c3 bis h4, LEDs leuchten blau) und wieder zurück (c2 bis c3, LED stimmen. Mit der mittleren Taste (back) gelangen Sie in die Ausgangs-Stimmelage zurück, also links bei c1 beginnend bis c3 ganz rechts.

Auf dem Display wird das alles auch im Klartext angezeigt.

### Orgel mit ESP32-NodeMCU

Wie schon erwähnt, muss man sich nicht unbedingt das Core-Modul anschaffen, um die Tastatur zur Orgel zu machen, ein herkömmliches ESP32-Board reicht völlig. Der Schaltplan 7 zeigt Ihnen, wie Sie die Tastatur an eine ESP32-NodeMCU anschließen.

Zusätzlich werden noch ein Verstärkermodul mit Lautsprecher sowie drei Taster benötigt. Auf ein Display wie beim Original 8 habe ich verzichtet, die jeweils gewählte Oktave wird ja durch die Farbe der LEDs auf der Tastatur angezeigt.

Nachdem Sie die Schaltung zusammengebaut haben, schließen Sie sie an den PC an. Die Software ist dieselbe wie für das M5-Core-

## Kontaktbelegung

Nur sechs der 15 Anschlüsse der Pinleiste sind für die Piano-Tastatur erforderlich (von vorn nach hinten):

Pin	Belegung
1	GND
2	5V
3	3,3V
8	I <sup>2</sup> C SDA
10	I <sup>2</sup> C SCL
11	LED DATA

Modul. Auch die Einstellung bezüglich des Boards ist gleich. Da aber die Pins, an denen die Taster des Core-Moduls hängen (37, 38 und 39), auf Standard-ESP32-Boards nicht zur Verfügung stehen, musste ich andere verwenden (12, 13 und 15). Software-mäßig ist dazu lediglich eine Bibliothek anzupassen. Sie finden sie im Ordner *Eigene Dateien/Arduino/libraries/M5Stack-master/src/utility*. Sie heißt *Config.h* und kann mit jedem Texteditor bearbeitet werden. Die im Bild 9 in grün gezeigten Zeilen müssen Sie durch die braun geschriebenen ersetzen. Vergessen Sie nicht, die Datei wieder zu speichern.

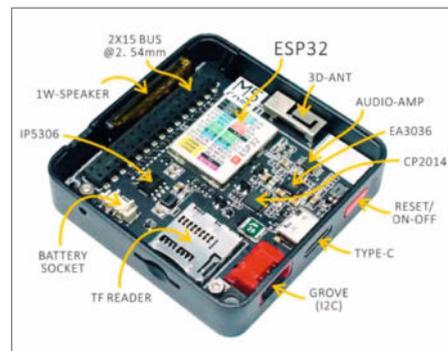
Falls Sie jedoch später einmal das M5-Core-Modul verwenden möchten, denken Sie daran, diese Änderung rückgängig zu machen. Sonst bleiben die Taster des Moduls wirkungslos. Technische Unterlagen zum Modul und der Tastatur sind per Kurzinfo-Link verfügbar 10.

Wenn Sie danach die Software kompiliert und übertragen haben, schalten Sie das Verstärkermodul ein und schon können Sie oder Ihre Kinder die fetzigsten Weihnachtslieder spielen. —hgb

```

/* alte Button-Definitionen
#define BUTTON_A_PIN 39
#define BUTTON_B_PIN 38
#define BUTTON_C_PIN 37
*/
//Button-Definitionen für ESP32-Boards
#define BUTTON_A_PIN 13
#define BUTTON_B_PIN 12
#define BUTTON_C_PIN 15
    
```

9 In der Config-Datei müssen Die Pin-Nummern für die drei Taster angepasst werden.



10 Die Dokumentation zu den Bestandteilen des M5-Core-Moduls finden Sie über den Kurzinfo-Link.

# secIT by Heise

Der Treffpunkt für Security-Anwender und -Anbieter

**29. – 31.03.2022,**  
Hannover

**SAVE THE DATE**

Seien Sie dabei und profitieren Sie als Besucher von neuesten Trends, Produkten und Software-Lösungen in der IT-Security.

**Das erwartet Sie:**

- Große Ausstellung in zwei Hallen
- Spannende Fachvorträge auf drei Bühnen
- Praxisnahe Workshops mit Security-Experten
- Informative Expert-Talks der wichtigsten Security-Anbieter
- Austausch mit der Community vor Ort

Am **13. und 14. September 2022**  
findet zusätzlich eine **digitale secIT** statt.

Veranstalter

 **Heise Medien**

© Copyright by Maker Media GmbH.

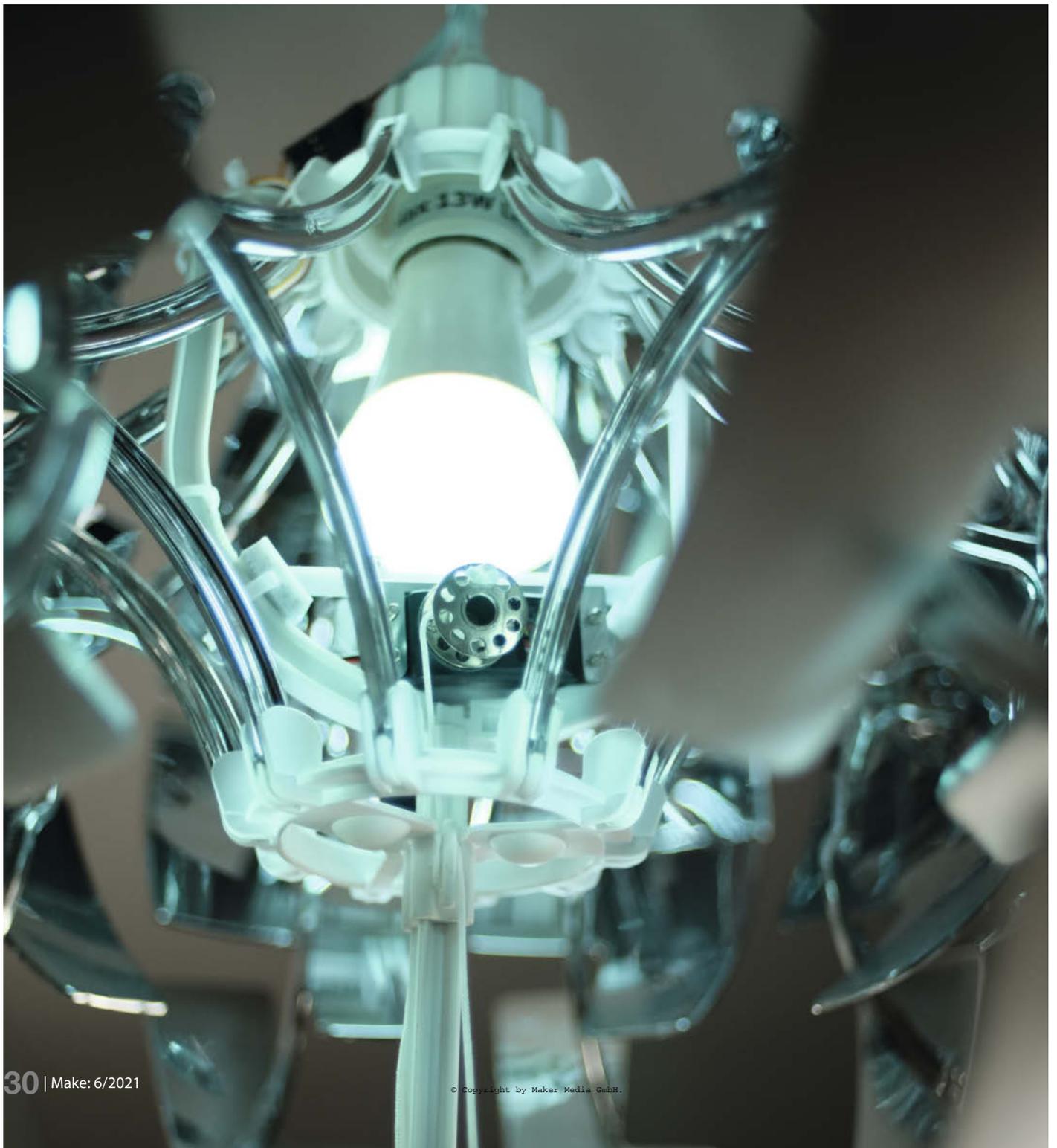
Mehr zu den  
Highlights finden Sie hier:

**sec-it.heise.de**

# Ikea-Hack: Todesstern dimmen mit Alexa

Die von Science-Fiction-Filmen inspirierte Hängeleuchte IKEA PS 2014 lässt sich mit einem Zug an der Schnur dimmen, da sich die Plastikschalen der Lampe beim Ziehen Stück für Stück öffnen. Doch das lästige Aufstehen und An-der-Schnur-Ziehen wird schnell Vergangenheit, wenn man eine Sprachassistentin wie Alexa hat, die die müßige Arbeit mit Unterstützung eines WeMos-D1-Mini-Boards für einen erledigen kann. Wir zeigen die nötige Technik, damit das Ganze wie am Schnürchen läuft.

von Lisa Ihde



**F**ür die an einen *Todesstern* aus *Star Wars* erinnernde Ikea-Lampe findet man (neben dem Artikel aus der Make 2/2017, siehe Kurzinformatio) einige Hacks im Internet, doch keiner davon konnte mich wirklich überzeugen. Viele von ihnen sehen sehr abenteuerlich aus und sind nicht wirklich darauf ausgelegt, leicht nachmachbar zu sein.

Was mir außerdem auffiel: Viele Maker fokussieren sich darauf, die Lampe entweder ganz auf (100 Prozent) oder ganz geschlossen (0 Prozent) zu halten, aber überspringen die Schritte dazwischen. Einige nutzen zwar eine Fernbedienung und stoppen mit ihr den Motor, doch rein technisch kann man die Lampe exakter dimmen.

Damit kommen wir zu meinem ersten Ziel für das Projekt: Die Lampe soll sich per Sprachsteuerung nahtlos auf Werte wie 20 Prozent oder 80 Prozent öffnen lassen und nicht nur binär zwischen 0 Prozent und 100 Prozent wechseln. Und die Umsetzung soll sich möglichst elegant in der Lampe verbergen, denn dieser Aspekt wird auch oft nicht bedacht.

Beispielsweise hat User JasonC189 auf Instructables (Link in der Kurzinformatio) eine Anleitung hochgeladen, bei der er eine lange Gewindespindel in der Lampe verbaut hat, die unten aus der Lampe herauschaut. Dazu hat er zwei Schrauben so platziert, dass sie gegen Buttons stoßen sollen, um Signale für *offen* und *geschlossen* zu vermitteln. Leider sind bei dieser Konstruktion sehr viele Kabel im Spiel, die das Design der Lampe stören.

Ein weiterer Maker namens Emanuel stellt auf seinem Blog Coldcorner einen Hack vor, bei dem er einen sehr großen Motor verbaut hat, woraufhin dann kein Platz mehr für das normale Leuchtmittel ist. Er greift daher zu vier *G9-LED*-Leuchtmitteln, die mit der Befestigungsart nicht wirklich zur Ästhetik der Lampe passen.

Durch diese Recherche entstanden konkretere Ansprüche an meinen Hack: Er sollte sehr einfach nachzumachen und dabei so unauffällig wie möglich sein. Zusätzlich fiel mir auf, dass einige Hacks nicht mit defekten Komponenten rechnen und somit der Austausch von Teilen nur schwer umzusetzen wäre. Daher ist es mir wichtig, dass die Komponenten leicht austauschbar sind. Doch bevor es ans Hacken geht, schauen wir uns erstmal an, wie der Mechanismus in der Lampe genau funktioniert.

## Mechanik der Lampe

Im Innern der Ikea-Lampe sitzt ein weißes Gestell, das der Form eines Pokals ähnelt. Im Bogen befindet sich eine weiße Spule, über die eine Schnur führt. Wenn man nun an der Schnur zieht, wird die Kraft umgelenkt. Dadurch öffnet sich die Lampe genau um die Strecke, die man an der Schnur gezogen hat. An der Schnur ist ein Ring befestigt, der sich entlang einer Stange nach oben bewegt. An

## Kurzinformatio

- » Drahtlose Steuerung über ESP8266-Modul und Alexa
- » Lampe punktgenau dimmen dank Servo mit Encoder
- » 3D-Druck einer Servohalterung

## Checkliste



### Zeitaufwand:

etwa 2 bis 3 Stunden  
(plus 40 Minuten Druckzeit)



### Kosten:

60 Euro für Ikea-Lampe;  
50 Euro für restliche Materialien



### Löten:

einfache Lötarbeiten



### 3D-Druck:

Servo-Halterung drucken



### Programmieren:

Code mit der Arduino IDE aufs Board laden

## Material

- » IKEA PS 2014
- » WeMos D1 Mini-Board
- » Mikro-USB-Kabel
- » Feetech FB5317M 360° Servomotor
- » Servohalterung aus Aluminiumlegierung passend für S3003 MG995 Standard-Servos
- » 2 x 10cm-Kabel mit 0,2mm<sup>2</sup> für 5V (etwa Jumper-Kabel)
- » 2 x 10cm-Kabel mit 1,5mm<sup>2</sup> für 230V
- » 1 x 30cm-Kabel mit 4 Adern je 0,2mm<sup>2</sup> als Kabelverlängerung vom Motor zur Decke (etwa ein flaches Telefonkabel)
- » 7 weiße Kabelbinder
- » Metallspule für Nähmaschinen
- » Netzteilbaustein mit Schraubklemmen und Ausgangsspannung von 5V
- » 2 Wago-Hebelverbindungsklemmen mit 3 x 0,2-4mm<sup>2</sup>

## Mehr zum Thema

- » Florian Schäffer, Der Todesstern fürs Wohnzimmer, Make 2/17, S. 18
- » Daniel Bachfeld, Her Master's Voice, Make 1/18, S. 16
- » Daniel Bachfeld, Die IoT-Alleskönner: ESP32 und ESP8266, Make 6/19, S. 8

Alles zum Artikel  
im Web unter  
[make-magazin.de/xfqx](http://make-magazin.de/xfqx)

## Werkzeug

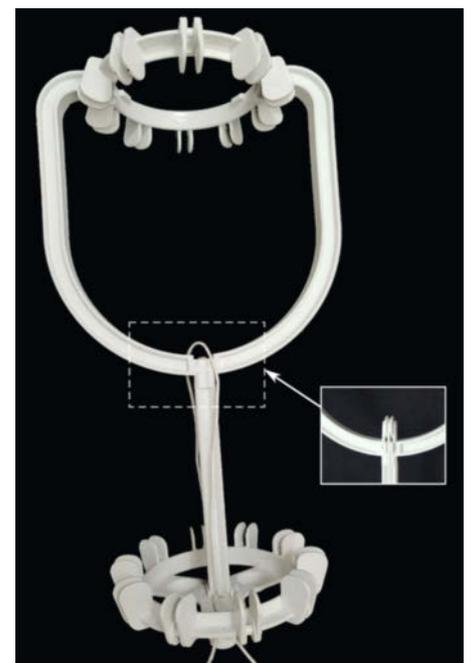
- » Schere
- » LötKolben
- » Lötzinn
- » Abisolierzange
- » Crimpzange für Kabelschuhe
- » Crimpzange für Dupontstecker
- » Lineal

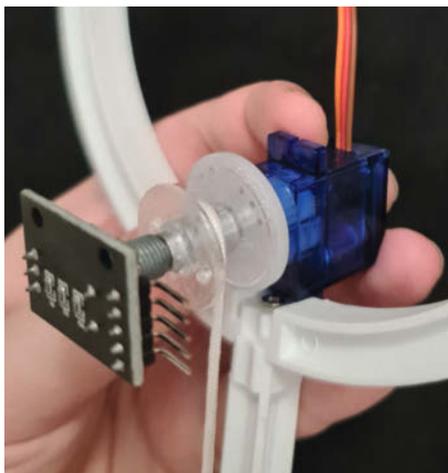
diesem Ring sind die Scharniere der Plastikschalen befestigt, mit denen diese geöffnet und geschlossen werden können. Diese Mechanik können wir später weiterverwenden, indem wir einen Motor an der Schnur ziehen lassen. Damit genug Platz dafür ist, können wir die weiße Spule einfach entfernen.

## Der richtige Dreh

Wir benötigen also einen Servomotor, der an der Schnur ziehen und sie dabei aufwickeln kann. Das klappt, wenn man eine Spule für Nähmaschinen an einem Servohorn befestigt und dieses dann am Motor anbringt. Doch der Motor würde sich auf diese Weise auch dann weiter drehen, wenn die Lampe schon maximal geöffnet ist. Wir müssen also initial herausfinden, wie viele Umdrehungen der

Die weiße Plastikspule, auf der normalerweise die weiße Schnur liegt, kann ganz leicht entfernt werden.





**Der gescheiterte Versuch, mit einem Feetech FS90R Servomotor und einem Dreh-Encoder die Ikea-Lampe zu öffnen**

Motor braucht, um die Schnur so weit aufzuwickeln, dass die Lampe maximal geöffnet ist. Das ist dann unsere maximale Anzahl an Umdrehungen, die möglich sein wird – also gut aufschreiben. Die Umdrehungen müssen später auch von der Lampe mitgezählt werden.

Zum Mitzählen benutzen wir einen Encoder, also einen Sensor für den Drehwinkel. Es gibt praktischere Motoren, in denen bereits ein Encoder verbaut ist. Bei meinem ersten Versuch habe ich noch einen Feetech FS90R

Servomotor verwendet, der keinen Encoder besaß. In dem Fall habe ich einen Dreh-Encoder zur Hilfe genommen. Der Drehknopf kann dafür auch an der Spule befestigt werden, sodass von einer Seite das Servohorn und von der anderen der Encoder angebracht sind.

Der Feetech FS90R Servomotor hat anfangs gut funktioniert, allerdings liegt die maximale Belastung bei 1,5kg. Die Ikea-Lampe benötigt zum Öffnen und Schließen aber ein größeres Stellmoment, das hat der Motor dann langfristig nicht ausgehalten. Zusätzlich gefiel mir auch nicht, dass die Komponenten verklebt wurden, da sie auf diese Weise schwer austauschbar sind. Als ich dann nach einem stärkeren Motor geschaut habe, bin ich auf den Feetech FB5317M Servomotor gestoßen, der einen magnetischen Encoder verbaut hat und somit den Zusammenbau wesentlich einfacher macht.

## Mit dem Motor kommunizieren

Doch wie funktioniert nun die Kommunikation mit dem Motor? Dafür nutzen wir den Wemos d1mini mit dem ESP8266-Modul. Dieser Mikrocontroller kann mit dem WLAN verbunden werden und für andere Geräte wie Alexa sichtbar sein. Zur Nutzung als Gerät für Alexa bindet man die Espalexa-Bibliothek (Link in der Kurzinfor) ein. So kann die Ikea-Lampe als Gerät in der Alexa-App erscheinen – und

die Steuerung des Mikrocontrollers über den Alexa-Sprachassistenten funktioniert reibungslos, wenn die entsprechende Callback-Funktion implementiert wird.

Den kompletten Code zum Hack findest Du online auf meinem Github (Link in der Kurzinfor) in der Datei *IKEAPS2014.ino*. Im Code heißt die Callback-Funktion `deviceChanged`, wie in Listing 1 zu sehen ist. Wenn Du Alexa „Öffne Star zu 20 Prozent“ befehlst, dann wird die Funktion `deviceChanged(brightness)` ausgeführt, wobei die 20 Prozent dort als `brightness` von 51 übergeben wird. Der Wert 51 ergibt sich daraus, dass `brightness` zwischen 0 und 255 liegt und 20 Prozent von 255 demnach 51 sind.

Nun berechnet der Algorithmus, wie oft sich die Spule drehen muss. Dafür muss man die maximale Anzahl an Umdrehungen (die wir ja bereits festgestellt haben) als Wert bei `MAX_ROTATION` festlegen. Dieser Wert bestimmt, wie oft die Schnur auf der Nähgarnspule aufgewickelt werden muss, bis die Lampe vollständig geöffnet ist. `MAX_ROTATION` hat im Code beispielsweise den Wert von 4500, das wären ungefähr 4,4 Umdrehungen. Für die Umrechnung muss der Wert durch 1024 geteilt werden, denn der Encoder des Motors liefert Werte von 0 bis 1024 für den Drehbereich von 0°-360°. Schlussfolgernd berechnet der Code nun 20 Prozent von 4500, was 900 ergibt. Da 900 kleiner als 1024 ist, ist somit das Ziel für den Motor keine ganze Umdrehung (tar-

### Listing 1

```
01 void startRotate() {
02   if (currentTurn * MAX_FEEDBACK + pos < targetTurn * MAX_FEEDBACK + targetPos) {
03     servoMotor.writeMicroseconds(2000); // when high level is 1.5-2ms, servo rotates counter clockwise
04     onRotate(Direction::forward); // open lamp
05   } else {
06     servoMotor.writeMicroseconds(1000); // when high level is 1-1.5ms, servo rotates clockwise
07     onRotate(Direction::backward); // close lamp
08   }
09 }
10
11 void deviceChanged(uint8_t brightness) { // brightness value between 0-255
12   if (oldBrightness == brightness) {
13     return;
14   }
15   oldBrightness = brightness;
16   int targetRotation = static_cast<int>(MAX_ROTATION * (static_cast<float>(brightness) / 255.f));
17   targetTurn = targetRotation / MAX_FEEDBACK;
18   targetPos = targetRotation % MAX_FEEDBACK;
19   pos = analogRead(FEEDBACK_PIN);
20   oldPos = pos;
21
22   servoMotor.attach(SERVO_PIN);
23   startRotate();
24   servoMotor.writeMicroseconds(1500); // servo will stop rotating when reaching 1.5ms
25   servoMotor.detach();
26 }
27
28 void setup() {
29   pinMode(FEEDBACK_PIN, INPUT);
30   wifiSetup();
31   espalexa.addDevice("star", deviceChanged);
32   espalexa.begin();
33 }
```

## Vorbereitungen

Zur Programmierung wird die *Arduino IDE 1.8.9* (Link in der Kurzinfor) verwendet. Um den *WeMos D1 Mini* anzusteuern, muss in der Arduino IDE unter *Datei > Voreinstellungen > Zusätzliche Boardverwalter-URLs* folgende Adresse hinzugefügt werden: [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json). Dann kann man „esp8266“ in das Suchfeld unter *Werkzeuge > Board > Boardverwalter* eingeben und auf „Installieren“ klicken. Nun erscheint das *LOLIN(WEMOS) D1 R2 & mini* unter den Boards und kann ausgewählt werden.

Zur Kommunikation mit Alexa nutzen wir die Bibliothek *espalexa* mit der Version 2.4.7. Dort ist ein kleiner Fehler, den man fixen kann, wenn man in Zeile 197 in *Espalexa.h* `arg(0)` zu `argName(0)` ändert.

In Zeile 5 und 6 von *IKEAPS2014.ino* musst Du Deine Zugangsdaten für Dein WLAN eintragen. Danach kannst Du den Code auf Deinen Mikrocontroller hochladen. Anschließend kannst Du Alexa bitten, nach neuen Geräten zu suchen und müsstest das Gerät *Star* finden.

der Spule stoppt, sobald die korrekte Rundenanzahl und Grad erreicht wurde.

Das war es schon! So lässt sich die Lampe mit gut 100 Zeilen Code dimmbar bewegen.

## Bauanleitung

Endlich hacken wir die Ikea-Lampe! Schritt für Schritt statten wir sie mit Technik aus, sodass wir die Lampe am Ende mit Alexa ansteuern können. **Vorsicht:** Bitte prüfe, dass Du die Sicherungen Deiner Stromzufuhr ausgemacht hast, denn 230V sind gefährlich – das weiß ich, auch ohne Elektrikerin zu sein.

Wie schon zuvor erwähnt, muss zur Vorbereitung die weiße Plastikspule vom Gestell der Lampe entfernt werden. Schneide die Schnur so durch, dass eine Seite der Schnur deutlich länger als die andere ist. Die längere Seite wird später an die Nähgarnspule geknotet. Die kürzere Seite kann noch kürzer gemacht werden, da die Schnur mit Kugel nur noch Deko sein wird.

Wie in der Grafik zu sehen, ist die Verkabelung überschaubar. Bis auf den Servomotor werden alle Komponenten später in der Lampenschale Platz finden. Meistens wird der Wemos ohne angelötete Beinchen geliefert. Da wir zwei Pins verwenden werden, musst Du hier mit dem Lötcolben männliche Pins bei D0 und A0 anbringen.

Der Servomotor bringt vier Kabel mit. Das orange Kabel (vom Encoder) steckst Du in A0

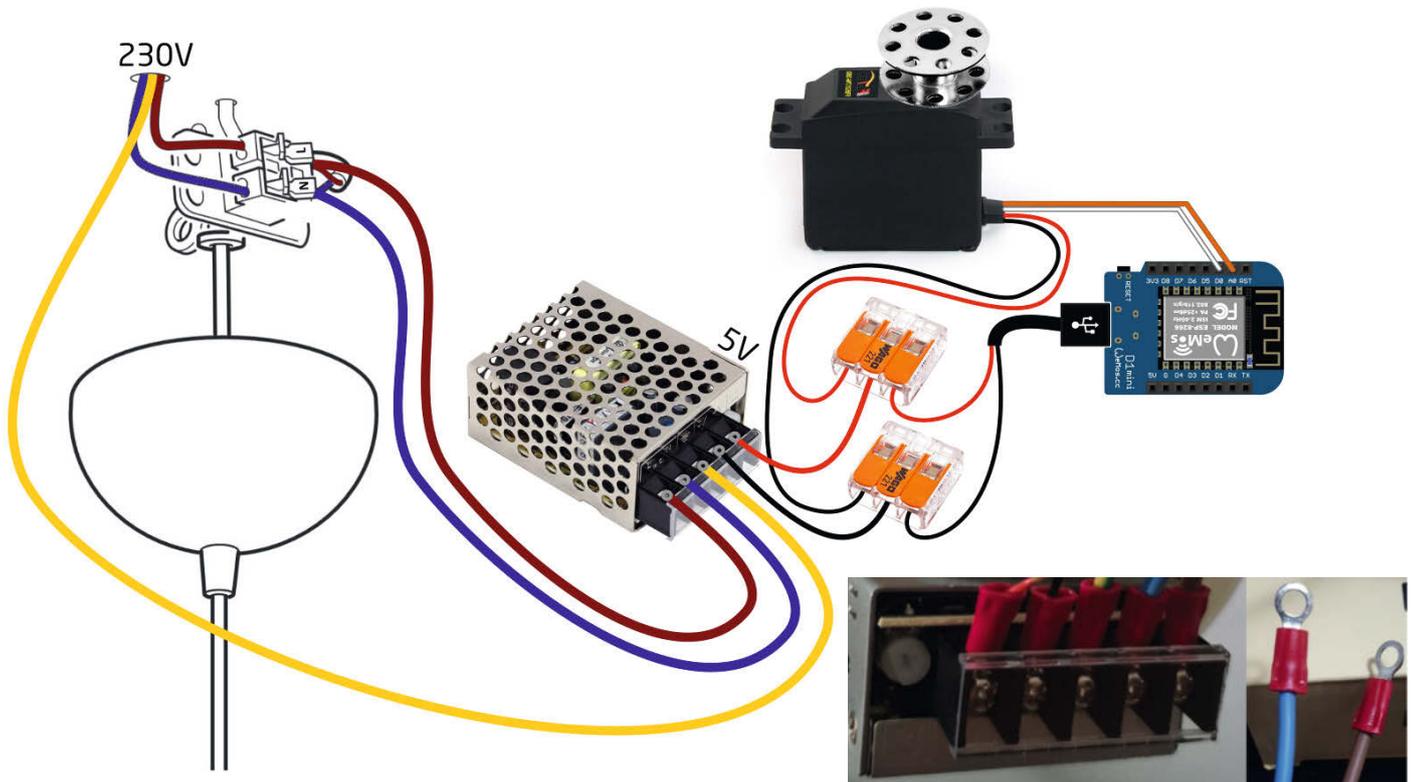
`getTurn` ist also 0) und 900 als Drehwinkel (`targetPos` ist somit 900) zu schaffen, damit die Lampe zu 20 Prozent geöffnet wird.

Nun wird im Code vom Encoder, angeschlossen am `FEEDBACK_PIN`, die aktuelle Position des Motors abgelesen (Zeile 19), um zu entscheiden, ob die Lampe sich entweder öffnen (Zeile 4) oder schließen (Zeile 7) soll, bis die 20 Prozent erreicht sind.

Hier kommt dann die Funktion `onRotate(val)` zum Einsatz. Diese zählt mit, wie viele Runden die Spule schon gedreht wurde. Immer, wenn die Werte vom `FEEDBACK_PIN` von

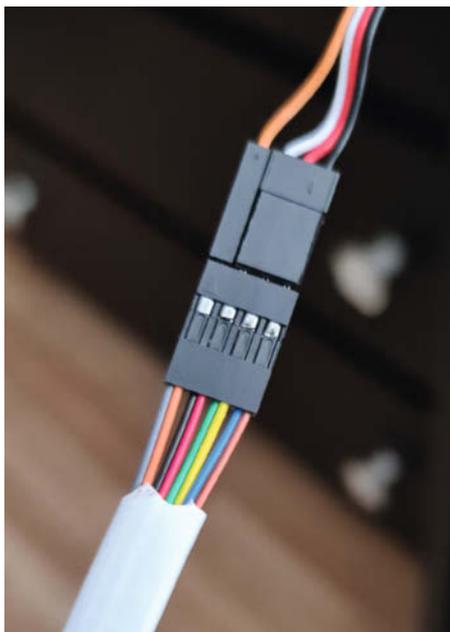
1024 auf 0 wechseln, ist eine Runde geschafft. Daher wird die Differenz zwischen dem letzten Positionswert und dem aktuellen berechnet.

Da der Encoder etwas ungenau ist, wird nicht auf exakt 1024 geprüft, sondern nur, ob der Betrag größer als 500 ist. Sobald die richtige Anzahl an Umdrehungen geschafft ist – in unserem Fall 0 Umdrehungen – wird der Drehwinkel geprüft. Dieser soll für 20 Prozent bei 900 als `targetPos` sein. Da der Encoder auch hier manchmal etwas ungenau ist, wird zur aktuellen Position ein `OFFSET` gerechnet, der beliebig angepasst werden kann. Das Drehen



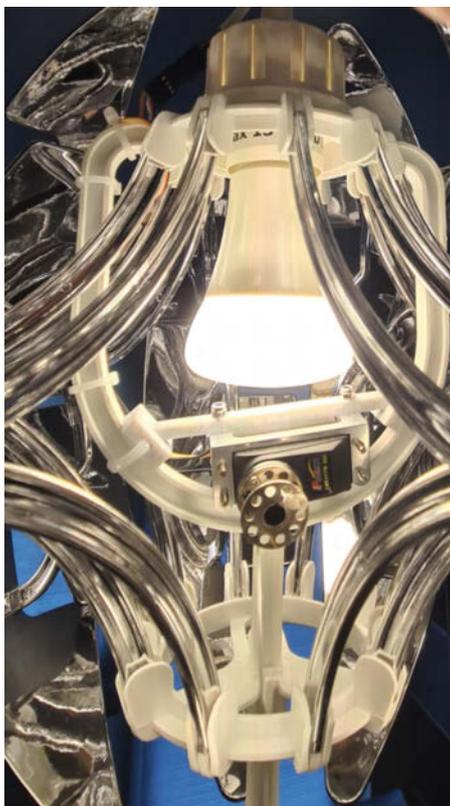
So schließen wir den Servo an.

Die Verkabelung mit dem Netzteil



Das Servokabel muss man verlängern.

vom Wemos. Das weiße Datenkabel steckst Du in D0. Die Stromversorgung für den Motor läuft nicht über den Mikrocontroller, denn dieser gibt über die Pins nur bis zu 500mA ab. Der Motor könnte den Mikrocontroller überlasten, da dieser bis zu 1800mA zieht, wenn er mit Widerstand dreht.



Den Motor mit Kabelbindern am Gestell befestigen.



Die von mir modellierte Halterung kann 3D-gedruckt werden und hilft, den Servo in der Ikea-Lampe an der richtigen Position zu halten.



Motorhalterung anbringen

Daher kommt die Stromzufuhr für den Motor und Mikrocontroller vom Netzteil, das 5V ausgibt. Damit die vier Kabel (jeweils Plus und Minus vom Motor- und Mikro-USB-Kabel) an die zwei Anschlüsse vom Netzteil passen, wird jeweils das rote und schwarze Kabel durch die Wago-Klemmen zusammengeführt. Richtig gelesen, dafür musst Du vom Mikro-USB-Kabel ein Ende abschneiden und die beiden Kabel abisolieren. Man kann alternativ auch fertige Mikro-USB-Kabel kaufen, die an einem Kabelende schon offen, abisoliert und versiegelt sind. In die Wago-Klemmen steckst Du dann jeweils die 10cm-Kabel

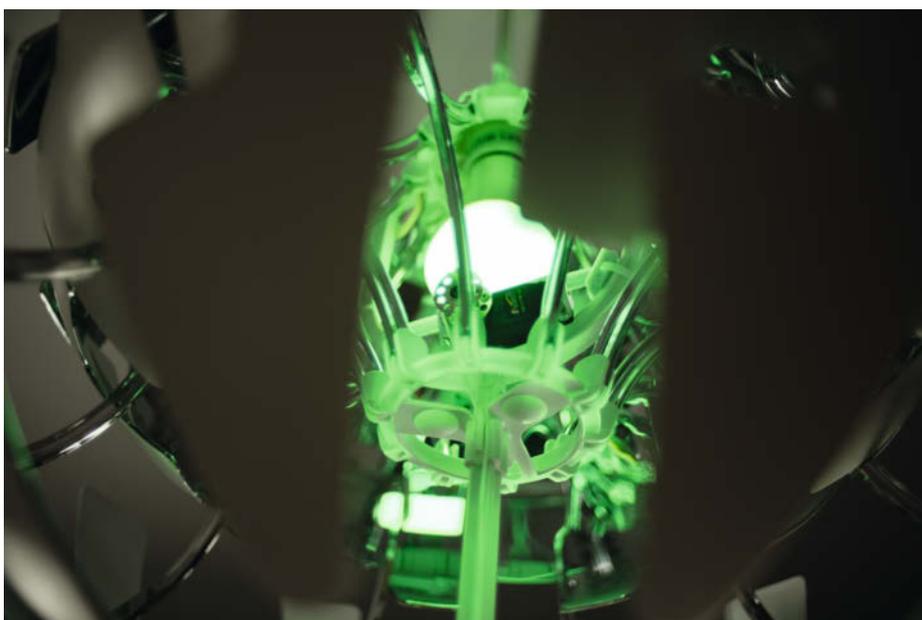
mit 0,2mm<sup>2</sup>, die im nächsten Schritt ans Netzteil kommen. Hier kannst Du alternativ auch Jumperkabel nutzen, wenn Du die Enden abschneidest.

Nun werden die zwei Kabel jeweils am Netzteil befestigt, indem das rote Kabel an +V und das schwarze Kabel an -V geschraubt wird. Damit man die Kabel besser anschrauben kann, sollten man die Enden mit Kabelschuhen ausstatten. Dafür kann man eine Crimpzange verwenden. Diese Kabelschuhe lassen sich dann in die Schraubklemmen setzen: Man dreht die Schrauben raus und schraubt diese dann mit dem Kabelschuh dazwischen wieder fest.

Das gelbe Erdungskabel aus der Wohnungsdecke wird dann an die Erdung des Netzteils angeschlossen. Danach befestigt man Kabelschuhe an zwei 10cm-Kabeln (mit 1,5mm<sup>2</sup> Dicke wegen der 230V), um diese dann bei N (blau) und L (braun) ans Netzteil zu schrauben. Das andere Ende wird dann an der Lampe in die Lüsterklemme gesteckt, wie in der Grafik mit dem braunen und blauen Kabel zu sehen ist.

## Stromversorgung optimieren

Der Servomotor wird später in der Lampe befestigt, alles andere bringen wir oben in der Schale an der Decke unter. Ich fand den vorhandenen Platz in der Lampe etwas eng für das Netzteil, denn schnell wird die Mechanik behindert und die Optik gestört. Somit müssen die nötigen vier Kabel für den Motor von der Stromversorgung nach unten in die Lampe geleitet werden. Da die vier Kabel vom Motor zu kurz sind, sollten wir sie verlängern. Dazu benutzen wir ein 30cm langes Kabel mit vier Adern (je 0,2mm<sup>2</sup>) und



Nutzt man eine Philips Hue, kann die Lampe in verschiedenen Farben leuchten.

befestigen mit einer Crimpzange an einem Ende einen männlichen Dupont-Stecker mit vier Steckplätzen, sodass dort die vier Kabel vom Servomotor ranpassen. Am anderen Ende des Kabels befestigen wir einen Dupont-Stecker mit zwei Steckplätzen für A0 und D0, die nebeneinander am Wemos liegen. Die Verlängerung vom roten und schwarzen Kabel braucht keinen Dupont-Stecker, da wir diese jeweils in die Wago-Klemmen stecken.

In meinen Fall habe ich ein flaches Telefonkabel genutzt. Meins hatte allerdings acht Adern. Aus diesem Grund habe ich die Dupont-Stecker so angebracht, dass immer zwei Adern zusammensitzen. Alternativ kannst Du, wie erwähnt, gleich ein 4-adriges Kabel nutzen.

### Motor installieren

Das war nun alles zur Verkabelung. Als Nächstes befestigen wir die Nähgarnspule am Motor. Dafür wird die Nähgarnspule mit zwei kleinen Schrauben an das schwarze, kreisförmige Servohorn geschraubt. Anschließend kann man das Servohorn an den Motor schrauben.

Damit wir den Motor in der Lampe am Gestell befestigen können, verwenden wir eine Kombination aus einem Servohalter und einer von mir erstellten, 3D-gedruckten Halterung. Der Motor wird zuerst an den Servohalter aus Aluminium geschraubt. Anschließend wird das 3D-Modell gedruckt und der Servo mit Aluminiumhalter angeschraubt.

Das 3D-Modell besteht aus einer Stange mit zwei Schlitz an den Enden, an denen man jeweils einen Kabelbinder befestigt, der dann ans Gestell der Lampe gehört. Somit kann man den Motor ganz leicht jederzeit ausbauen.

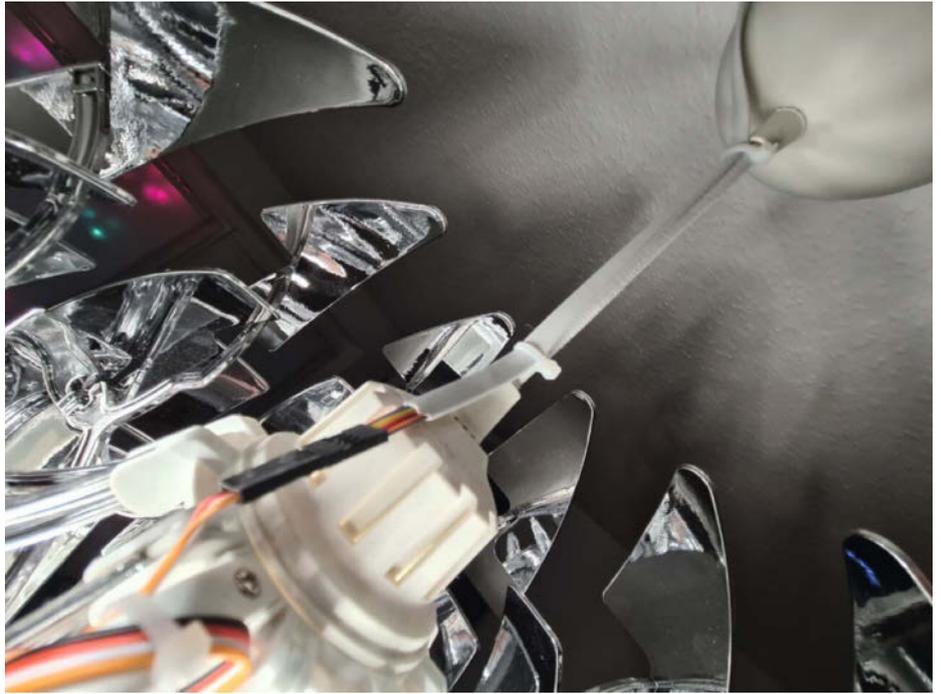
Nun befestigen wir mit zwei Kabelbindern die Stange vom 3D-Modell am Gestell, wie auf dem Foto zu sehen ist. Anschließend wird die weiße Schnur der Lampe an der Metallspule vom Servomotor mit einem Knoten so angebracht, dass die Schnur einigermaßen auf Spannung ist, wenn der untere Ring ganz unten ist. Schneide die überschüssige Länge von der Schnur ab. Mit weiteren Kabelbindern kannst Du dann Dein Kabel vom Motor entlang des Gestells hoch zur Decke führen.

Fertig ist der Zusammenbau!

### Fazit & Ausblick

Ich habe bei meinem IKEA-Hack eine *Philips Hue Lampe* verbaut, die Farbwechsel unterstützt. So kann ich meine Lampe in jeder Farbe leuchten lassen, die ich möchte, und bestimmte Effekte – wie *Farbschleifen* – abspielen.

Wer möchte und sich an das Steckernetzteil der Lampe traut, könnte das mitgelieferte Kabel zwischen Decke und Lampe aus-



Hochführung des Kabels vom Servomotor zur Decke

tauschen. Wenn dort weitere vier Adern für den Motor vorhanden wären, müsste kein weißes Kabel mehr zusätzlich von der Lampe zur Decke führen. Dafür müssten dann allerdings auch die Öffnungen an der Lampenschale und bei der Lampe vergrößert werden.

Ein weiterer Vorschlag, der mir zugetragen wurde, ist das Designen einer eigenen Lam-

penschale, auch bekannt als *Lampenbalda chin*, für die Decke: Dort könnte man einen RGB-Leuchtbstreifen einbauen und ansteuern. Mir persönlich reicht der jetzige Effekt schon komplett aus und erfüllt alle meine Wünsche. Natürlich könnte man die Lampe auch von außen künstlerisch wie einen Todesstern bemalen. Viele Inspirationen dazu findest Du im Internet. —rehu



Der Lichteffect an der Decke, der durch die Designerlampe entsteht.

## Umzug in St. Pölten

In St. Pölten ist der Hackerspace Segmentation Vault in die Ludwig-Stöhr-Str. 8 gezogen und hat nun dienstags wieder für alle geöffnet.

[segvault.space](http://segvault.space)

## Haus der Innovationen

Das Fablab Siegen ist in das Haus der Innovationen in der Sandstraße 26 gezogen.

[fablab-siegen.de](http://fablab-siegen.de)

## Maker-Termine

**Jugend hackt Dresden**  
10.–12. Dezember 2021  
Technische Sammlungen  
Dresden

[jugendhackt.org](http://jugendhackt.org)

**Remote Chaos Experience**  
27.–30. Dezember 2021  
online  
[events.ccc.de](http://events.ccc.de)

**TIPE 3D Printing**  
18.–20. Januar 2022  
online  
[tipe3dprinting.com](http://tipe3dprinting.com)

**Coding Da Vinci Ost<sup>3</sup>**  
12. Februar – 2. April 2022  
SLUB Dresden, online  
[codingdavinci.de](http://codingdavinci.de)

Diese und weitere Termine stehen auch laufend aktualisiert in unserem Kalender auf der Webseite unter: [heise.de/make/kalender/](http://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](mailto:mail@make-magazin.de)



Bild: Open Knowledge Foundation/Deutschland

## Reparieren, erhalten, erfinden

### In der Lausitz bringen Mobile Fablabs offene Hardware voran

Wie können wir liebgewonnene Gegenstände länger erhalten? Was kann jeder und jede dafür tun, um die wachsende Trockenheit in den Griff zu bekommen? Und welche Ideen gibt es für mehr Mobilität? Mit diesen und weiteren Fragen wird sich in den kommenden zwei Jahren das Projekt *Mobile Fablabs in der Lausitz* (MoFab) beschäftigen. Dafür stellt das *Netzwerk Offener Werkstätten Brandenburg* seine Werkstätten zur Verfügung – oder kommt mit einem mobilen Fablab vorbei.

Zum Auftakt des Projekts trafen Ende Oktober mehrere mobile Makerspaces auf dem Cottbusser Altmarkt zusammen: die Makerkutsche aus Südbrandenburg, das Fabmobil aus Ostsachsen und das mobile Fablab aus Potsdam. Zudem konnten auf einem Lastenrad der TH Brandenburg Drehbänke und eine Vibrationsstichsäge ausprobiert werden. Ergänzt wurde die tem-

poräre „Wagenburg“ mit einer Ausstellung zum Thema Open Hardware, mit Exponaten wie dem Laptop *MNT Reform*, der selbstreinigenden Dusche *Showerloop* und dem Selbstbau-Windrad *Windkit*. Unter dem Motto „Reparieren, erhalten, erfinden“ wird das Projekt nun in den nächsten Monaten in der Lausitz unterwegs sein. Neben der Ausstellung werden dabei Repair Cafés und Workshops für offene Hardware und das Reparieren werden.

Anfang 2022 wird schließlich ein Förderprogramm für Open Hardware starten, mit dem sechs Projektgruppen unterstützt werden. Im Fokus soll dabei Hardware stehen, die lokale Probleme im ländlichen und kleinstädtischen Raum löst oder die Reparaturkultur verbessert.

—hch

► [now-bb.de/projects/mofab](http://now-bb.de/projects/mofab)

## Maker Faire Deutschland – Digital Edition

### Der Jahresauftakt der deutschsprachigen Maker Community

Aufgrund der großen Resonanz der Maker Faire–Digital Edition im vergangenen Juni wird die Veranstaltung künftig regelmäßig stattfinden und damit das bisherige Event-Programm der Make ergänzen. Den Jahresauftakt macht dabei künftig die Maker Faire Deutschland – Digital Edition 2022, bei der Trends und Themen für das neue Jahr präsentiert und diskutiert werden. Am 12. Februar stellen dort 20 ausgesuchte Maker und Partner an virtuellen Ständen neue Projekte vor.

Für alle Teilnehmer besteht die Möglichkeit sich mit ihnen auszutauschen, zu vernetzen und Fragen zu stellen. Neben den Ausstellerständen wird es wieder eine Entertainmentshow mit abwechslungsreichem und mitreißendem Live-Programm geben. Auch die Make-Redaktion ist dabei: Freuen Sie sich auf Vorträge, Tipps und Tricks sowie Fragerunden im engen Kreis. Mit welchen weiteren Ausstellern Sie rechnen

können und wie das Programm im Detail aussieht, wird im Januar bekannt gegeben.

—anlu

► [maker-faire.de/deutschland-digital-edition](http://maker-faire.de/deutschland-digital-edition)



# Schlüsselübergabe in Erlangen

## Nach einigen Zwischenstationen gibt es feste Räume für den Makerspace

In Erlangen hat der *Betreiberverein* die Schlüssel für das *Zentrum für Austausch und Machen (ZAM)* von der Stadt bekommen, um dort künftig seinen Makerspace einrichten zu können. Das ZAM wird in den Räumen des ehemaligen Ladengeschäfts Greiner unterkommen, die von der Stadt aufgekauft wurden. Nach einer Renovierung sollen hier verschiedene Werkstätten entstehen, etwa für Prototyping, Elektronik-, Metall- und Textilarbeiten. Kreativer und künstlerischer Austausch soll ebenso möglich sein, wie Workshops und Bildungsangebote weiterer lokaler Initiativen und städtischer Einrichtungen. Mit über 3.000 Quadratmetern Fläche wird dafür im ZAM auf jeden Fall reichlich Platz sein.

Im Sommer 2020 entstand mit dem Pop-up Space *ex-Teppich* bereits für eine kurze Weile ein neuer Experimentierraum, der 2021 mit dem „Show-Room“ im ehemaligen Lederwarengeschäft Pfeiffer weitergeführt wurde. Seit März 2021 gibt es mit dem Betreiberverein nun einen Träger für das Projekt. Der weitere Aufbau wird

vom Bundesinnenministerium über das Programm „Post-Corona-Stadt“ gefördert. Die nötigen Arbeiten in den Bereichen Brandschutz und Sicherheit werden voraussichtlich noch bis 2023 dauern – im Schaufenster des ZAM sollen bis dahin aber immer wieder Makerprojekte zu bewundern sein. —hch

► [betreiberverein.de](http://betreiberverein.de)



Bild: Jochen Hunger

## Förderung für Technikbildung

Schulen und Bildungsinitiativen, die Technikprojekte vom 3D-Druck bis zum Roboterbau umsetzen, können ab sofort beim VDI-Joachim-Herz-Technikfonds bis zu 2.500 Euro Förderung beantragen.

[vdi.de/netzwerke-aktivitaeten](http://vdi.de/netzwerke-aktivitaeten)

## Umzug in Neuenstadt

In Neuenstadt hat das Fablab seit Oktober wieder geöffnet – und ist vorher auch noch schnell umgezogen, in die Öhringer Str. 24.

[fablab-neuenstadt.de](http://fablab-neuenstadt.de)

### 46 DIY Erzgebirge

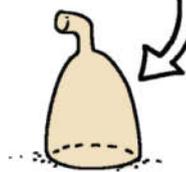
# Kolophonium

von und mit @beetlebum

MAN STELLE SICH EIN BELIEBIG KOMPLEXES RÄUCHERMÄNNCHEN VOR.



WENN MAN ES VON AUßEN NACH INNEN DREHT, SIEHT ES GAR NICHT MEHR SO KOMPLEX AUS.



DAS BEKOMMT JA WOHL JEDER IN DEM 3D-PROGRAMM SEINER WAHL KONSTRUIERT!

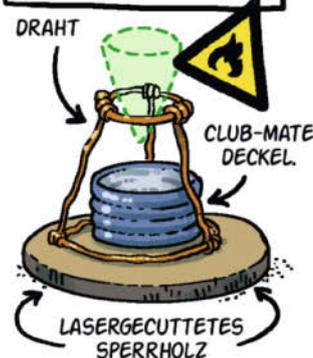
JETZT LÄDT MAN SICH EINFACH EINE BELIEBIGE FIGUR ALS STL AUS DEM INTERNET...



SCHIEBT BEIDE FIGUREN INEINANDER UND FÜHRT DIE BOOLSCHHE OPERATION „ABZIEHEN“ DURCH. NUN...



... NUR BEACHTEN: EINE ORDENTLICHE AUFHÄNGUNG FÜR RÄUCHERKERZEN!



DIE RÄUCHERKERZE NICHT AUF PLA STELLEN!



AM ENDE: FROHES FEST!



ICH SPRECHE DA AUS ERFAHRUNG!

# Willkommen im Make: Universum

von Rebecca Husemann

Die Make – das ist nicht nur eine Zeitschrift, sondern eine ganze Community. Und wir freuen uns, dass wir so eine engagierte Gemeinschaft von Lesern an unserer Seite wissen. Manchmal sind sich Leser allerdings nicht ganz sicher, wie sie selbst mitmachen können, darum gibt's hier nochmal einen knackigen Überblick über unsere Community-Angebote.

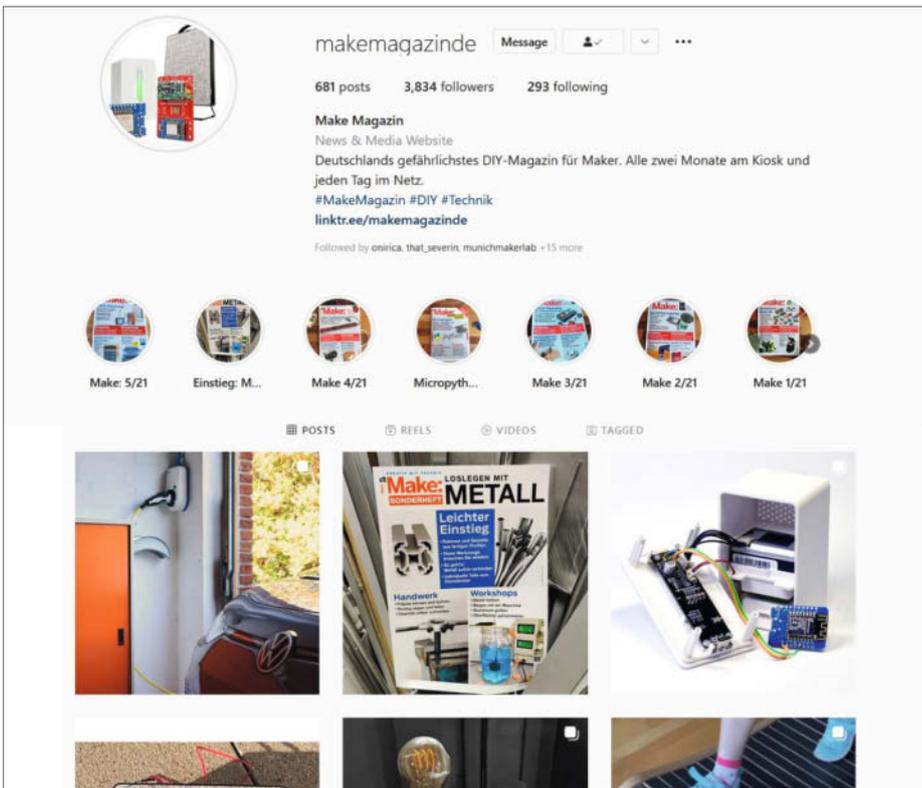
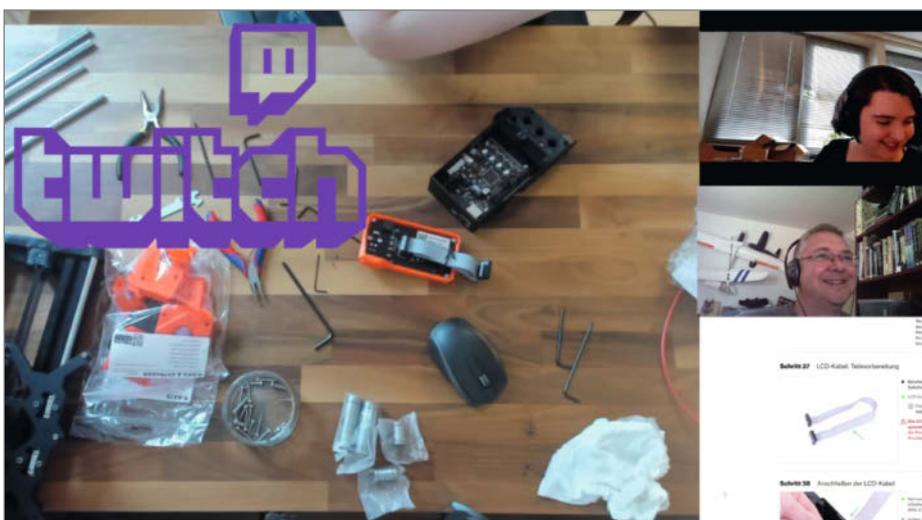
## Lernen Sie die Redaktion kennen – auf YouTube und Twitch ...

Die *Make* besteht nicht nur aus ein paar Projekten auf Papier, sondern auch aus einer Redaktion voll enthusiastischer Hardware-Fans und Elektroschrott-Horter. Wenn Sie nie die Chance hatten, uns mal live auf einer *Maker Faire* kennen zu lernen, dann schauen Sie uns doch digital über die Schulter.

Auf unseren Kanälen auf **Youtube** und **Twitch** basteln, frickeln, schwatzen und lachen wir, geben Touren durch unsere Werkstätten, testen coole Geräte, machen vieles heile und auch manches kaputt. Sie finden uns dort unter *MakeMagazinDE*. Und weil die Make nicht von gestern ist, dürfen Sie sich in Zukunft sogar auf noch mehr Video-Content freuen. Haben Sie spezielle Wünsche? Immer her damit!

## ... und auf unseren Social Media-Kanälen

Unter dem Namen *MakeMagazinDE* finden Sie die Make auch auf **Twitter**, **Facebook**, **Pinterest** und **Instagram**. Besonders Instagram ist immer einen Besuch wert, denn dort gewähren wir in unseren Stories regelmäßig kleine Einblicke in die Redaktion und die Projekte, an denen wir gerade arbeiten. Auf Twitter halten wir Sie auf dem Laufenden, welche Meldungen online erschienen sind und retweeten auch gerne spannende Posts aus der Community – taggen Sie uns ruhig.



## Machen Sie mit!

Aller Anfang ist schwer, aber wir freuen uns, wenn auch Sie aktiver Teil der Community werden wollen. Von „Nur mal reinschnuppern“ bis „Den anderen zeigen, wie fies die Tesla-spule zappen kann“ gibt es bei uns ganz viele Möglichkeiten:

### Maker Faires – mal hautnah, mal online

Die **Maker Faires** sind eines unserer Herzensprojekte und die beste Gelegenheit, uns kennenzulernen und sich mit uns auszutauschen. Sie sind ein Festival für Inspiration, Kreativität und Innovation, für alle Generationen. Auf jeder Maker Faire gibt es viele interessante Mitmachstationen, ergänzt um spannende Vorträge und Workshops. Kommen Sie als Besucher – oder noch besser: als Aussteller! Große und kleine Maker Faires gibt es in ganz Deutschland, Österreich und der Schweiz und sie geben uns die Gelegenheit, Maker aus allen Ecken zu treffen. Die nächste Maker Faire findet am 25. und 26. Juni 2022 in Reutlingen statt und in Hannover sehen wir uns (hoffentlich) vom 10. bis 11. September 2022. Weitere Termine gibt's unter [maker-faire.de](http://maker-faire.de).

In Pandemie-Zeiten können wir uns leider alle nicht live treffen, um uns gegenseitig unsere Lieblingsprojekte unter die Nase zu halten. Dieser Austausch fehlt uns, daher haben wir verschiedene neue Formate ins Leben gerufen: Zum Beispiel unsere große **Online-Maker Faire** und die **Make-Entertainmentshows**, kostenlos und frei zugänglich für alle, live gestreamt auf *Youtube* und *Twitch*.

### Make Projects

Wenn Ihre eingestaubten Projekte Sie vorwurfsvoll durch die transparenten Wände ihrer Aufbewahrungsboxen anstarren, dann ist es an der Zeit, sie mit anderen zu teilen. Auf unserer Projektplattform **Make Projects** können Sie sich registrieren und dort Ihre Projekte dokumentieren und Schritt-für-Schritt-Anlei-

tungen schreiben. Wenn Ihnen das etwas zu schnell geht, bauen Sie doch erstmal eins der Projekte nach, die Maker aus der Community schon hochgeladen haben. Wie wäre es zum Beispiel mit selbst gedrehten Gewürzmöhlen, die aussehen, wie Schachfiguren? Los geht's! Unsere Projektplattform finden Sie unter [makeprojects.com/de](http://makeprojects.com/de).

### Forum

Das Heise-Forum hat zwar nicht den besten Ruf, aber das **Make-Forum** ist eine kleine heile Parallelwelt, in der wir Fragen zu Artikeln beantworten, Gleichgesinnte vernetzen und auf Lob und Tadel reagieren. Das Forum lebt von Ihrem Input und regen Diskussionen, also machen Sie gerne mit: Schreiben Sie uns Ihre Meinung, beantworten Sie Fragen von anderen Makern und finden Sie vielleicht sogar Menschen, mit denen Sie gemeinsam Projekte umsetzen können. Zum Forum geht es hier lang: [heise.de/s/Xojo](http://heise.de/s/Xojo)

### Kalender

Japp, sogar unser **Kalender** ist *community-driven*. Haben Sie ein spannendes Event gesehen oder sind Sie vielleicht sogar der Veranstalter? Tragen Sie es in den Make-Veranstaltungskalender ein. Ganz gleich, ob Löt-Workshop, Community-Treffen oder Seifenkistenrennen, in unserem Kalender ist es richtig und findet sich vielleicht sogar in unseren *Werkstattberichten* wieder: [heise.de/make/kalender/](http://heise.de/make/kalender/)

### Werden Sie Teil des Magazins!

Wir suchen stets neue **Autorinnen und Autoren**: Rubriken wie *Tipps* und *Tricks* leben von Zuschriften aus der Community. Kein Tipp ist zu doof – und wenn doch, werden wir Ihnen das ganz schonend beibringen.

Sie haben ein spannendes Projekt gebaut? Dann schicken Sie uns eine E-Mail mit einer kurzen Beschreibung und Fotos, Videos oder



weiterführenden Links an [mail@make-magazin.de](mailto:mail@make-magazin.de). Wir freuen uns auch über Hinweise auf kuriose Basteleien, nützliche Werkzeuge oder spannende Materialien, gerne auch über unsere Social-Media-Profilen.

### Leserbriefe

Sie wollen uns einen **Leserbrief** zu unserem Heft schreiben? Dann am besten schnell zum Kontaktformular! Das Kontaktformular finden Sie unter [heise.de/make/kontakt](http://heise.de/make/kontakt). Es ist auch für verschiedene andere Anliegen geeignet, etwa wenn Sie Fragen zum Abo oder zum Heftkauf haben.

## Unsere Specials und Sonderhefte

Im Make-Magazin veröffentlichen wir alle zwei Monate Anleitungen und Einstiegsartikel zu Elektronik- und Mikrocontroller-Projekten. Für alle, die sich in ein Thema so richtig reinernen wollen, gibt es dann noch unsere Sonderhefte und Specials: Mit einem Schwerpunkt, den man von vorne nach hinten durcharbeiten kann – von Metallbearbeitung über ESP32 und Arduino bis zu Elektronik-Grundlagen. Alle hier aufgeführten Hefte

finden Sie unter [shop.heise.de/magazine/make-magazin/sonderhefte](http://shop.heise.de/magazine/make-magazin/sonderhefte).

### Sonderhefte aus der Werkstatt: „Loslegen mit Metall“ und „Loslegen mit Holz“

Unser neuester Zugang ist das Make-Sonderheft 2021 *Loslegen mit Metall*. Auf 162 Seiten geben wir eine komplette Übersicht über die

wichtigsten Metallbearbeitungstechniken für Maker – und ein paar experimentelle Techniken, die Sie ausprobieren können, wenn Schrauben und Schleifen zu langweilig geworden ist. Wir demonstrieren, wie man richtig körnt, bohrt, sägt und feilt, wie man Metalle verbindet und wagen uns in die Welt des Schweißens. (12,90 Euro)

Im Vorgänger *Loslegen mit Holz*, dem Make-Sonderheft von 2020, dreht sich natürlich



Yuliya D'yakova / Shutterstock.com

Einstieg in die Elektronik für die Schule, die Lehre, aber auch zu Hause. (44,95 Euro)

### Programmieren mit MicroPython

Keine Lust mehr auf C++ auf Mikrocontrollern? Dann probieren Sie doch mal MicroPython. Das *Make MicroPython Special* bietet auf 86 Seiten Stoff für Einsteiger und Umsteiger und zeigt, wie man leicht und schnell eigene Projekte mit dem ESP32 umsetzt. Wir erklären, wie man GPIOs steuert, Sensoren mit I<sup>2</sup>C abfragt, MQTT-Pakete per WLAN verschickt, Multithreading nutzt, Fehler mit try-catch abfängt und Strom mit *DeepSleep* spart. Da das MicroPython Special ohne Elektronikbeilage kommt, gibt es dieses ausnahmsweise auch als PDF zu kaufen. (19,90 Euro/12,90 Euro)



Dmytro Vietrov / Shutterstock.com

### ESP32-Special

Der ESP32 ist einer der beliebtesten Mikrocontroller – unter anderem, weil der Einstieg in die Hardware-Programmierung nicht nur einfach, sondern auch günstig ist. Im *Make ESP32 Special* erklären wir ESP-Grundlagen, zeigen, wie man Software schnell einrichten kann und vermitteln Hardware-Funktionen leicht verständlich. Wenn Sie dann Profi sind, können Sie in unseren Projektideen für Bastler schmökern: Wie wäre es mit einem Türsensor mit Cloud-Anbindung zur Überwachung Ihrer Haustür? Im ESP32-Special ist ein *ESP32 NodeMCU Entwicklerboard* enthalten, damit Sie sofort mit dem Ausprobieren beginnen können. (24,95 Euro)

### Node-RED mit dem ESP8266

IoT für Anfänger gibt es in unserem *Node-RED-Special*: Mit Node-RED kann man auch ohne tiefere Programmierkenntnisse komplexe Anwendungen für IoT und Smart Home entwerfen: Statt mit einem Editor Programmcode zu schreiben, verknüpft man dabei fertige grafische Bausteine zu einem „Flow“. Das macht nicht nur Spaß, sondern führt im Handumdrehen zu eindrucksvollen Anwendungen – auf Wunsch sogar mit komfortabler Bedienoberfläche. Durch den mitgelieferten *WLAN-Mikrocontroller ESP8266* können Sie Ihre Sensoren und Aktoren sogar drahtlos per MQTT und Node-RED kontrollieren. In drei Projekten zeigen wir exemplarisch, wie man Sensordaten abfragt, weiterverarbeitet und einen Motor steuert. (24,90 Euro)

alles nur um Holz. Für Maker mit zwei linken Händen starten wir ganz sanft mit Werkzeugkunde und arbeiten uns dann zur manuellen Arbeit mit Holz vor. Danach geht's zur maschinellen Arbeit mit der Säge und für Perfektionisten stellen wir uns auch nochmal an die CNC-Fräse. (9,99 Euro, leider nur noch als PDF verfügbar)

### Elektronik-Grundlagen lernen – mit Experimentierset

Unser *Make Elektronik Special* bietet einen einfachen und praxisorientierten Einstieg in Transistorschaltungen, die Maker in eigenen Projekten einsetzen können. Die Grundlagen der Elektronik erklären wir dazu leicht verständlich und holen Sie auf Ihrem Wissensstand ab. Das beiliegende Experimentierset beinhaltet insgesamt 44 Prototypenadapter mit allen wichtigen Bauteilen für das Lernen und für die Realisierung von Experimentierschaltungen. Es bildet, zusammen mit dem mitgelieferten Breadboard und dem Kabelset, eine einfach handhabbare Grundlage für den

### Programmieren in BASIC: PICAXE-Special

Das geht wirklich einfach: Programmieren mit dem Mikrocontroller PICAXE und der Programmiersprache BASIC. Das *Make Special PICAXE* ist ideal für alle, die mal näher an der Hardware programmieren wollen und noch einen ersten



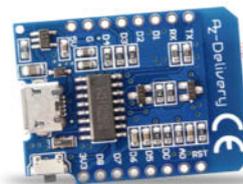


Anknüpfungspunkt suchen. Unsere BASIC-Programmier-Projekte sind so konzipiert, dass Sie das Heft in Ruhe von vorne bis hinten durcharbeiten und nachprogrammieren können. Damit Sie in die Arbeit mit dem PICAXE-Chip auch direkt einsteigen können, haben wir die Nano-Axe-Platine entwickelt. Und damit die Platine nicht nach dem Durcharbeiten des Specials in einer Schublade verstaubt, veröffentlichen wir regelmäßig neue Projekte online, die sie mit Ihrem PICAXE Nano-Axe-Board umsetzen können. (24,95 Euro)

### Make Arduino Special 1 und 2

Unsere beiden *Make Arduino Specials* lüften die Geheimnisse der Arduino-IDE und der

Hardware-nahen Programmierung. *Timer, Interrupts, Ports, Bit-Operationen*: Verstehen Sie nur Bahnhof? All das wird verständlich anhand praktischer Beispiele erklärt. Wir starten mit den Grundlagen der Arduino-IDE. Darauf aufbauend zeigt das Heft, wie man SD-Karten an den Arduino anschließt und für Projekte wie Datenlogger mit Echtzeituhr und fantastisch klingende Sound-Player nutzt. Ein großes Kapitel widmet sich der Ansteuerung von VGA-Monitoren und wie der Arduino dort mehrfarbige Grafiken anzeigt. Lernen Sie zudem, wie man effizient und übersichtlich programmiert und eigene Bibliotheken für die Arduino-IDE baut. In beiden Heften ist ein Arduino UNO gleich mit dabei. (je 7,99 Euro) —rehu



# KI steuert Follow-Me-Drohne

Um eine normale Drohne in eine Follow-Me-Drohne für elegante Videoaufnahmen zu verwandeln und sie ihrem Besitzer automatisch folgen zu lassen, muss man nur ihre Videoausgabe mit einem Algorithmus zur Erkennung von Personen verbinden. So braucht es nicht mal eine teure Fernbedienung, um die Drohne zu kontrollieren.

von Michael Jentsch



Unter den vielen verfügbaren Drohnen ist die Mini-Drohne *DJI Ryze Tello* etwas Besonderes: Mit ihr können wir auch ohne Vorkenntnisse eine autonome Drohne realisieren. Dank des *Tello SDKs* (das *Software Development Kit* des Herstellers) kann man mit ganz einfachen Befehlen eigene Anwendungen für sie entwickeln. Das macht die *Tello* zu einer idealen Einsteiger-Drohne für alle Maker, die mehr wollen, als ihre Drohne nur über die Wiese fliegen zu lassen.

Darüber hinaus besticht die *Tello* mit hervorragenden Flugeigenschaften und ist dank diverser Sensoren gut darin, eine stabile Position aufrechtzuerhalten. Die *DJI Ryze Tello* eignet sich auch deshalb gut als Einsteigerprojekt, da für die Programmierung viele inoffizielle *Libraries* zur Verfügung stehen. Sie erleichtern die Programmierung mit *Python*, *Java*, *C#*, *JavaScript*, *PHP*, *Go*, *Visual Basic* und vielen anderen Programmiersprachen enorm. Wir arbeiten für dieses Projekt mit *Python*.

### Zahlen, Daten, Fakten

Die *DJI Ryze Tello* hat trotz des geringen Gewichts von gerade mal 80g einiges zu bieten: Sie verfügt über eine 5-Megapixel-Kamera mit elektronischer Bildstabilisierung, die 720p-Videos mit 30fps streamen kann. Der 1,1Ah-Akku reicht für eine Flugzeit von gut 13 Minuten. Zur Stabilisierung ist ein Barometer und ein optisches Positionsbestimmungssystem verbaut, das die Drohne erstaunlich stabil in der Luft hält. Die maximale Geschwindigkeit wird mit 28,8km/h angegeben, die im *Schnell*-Modus erreicht werden kann. Im *Langsam*-Modus erreicht die Drohne immer noch 14,4km/h.

Leider kann die *Tello*-Drohne Videos und Bilder nicht lokal auf einer SD-Karte speichern, sodass bei einer schlechten WiFi-Verbindung schon mal Frames verloren gehen können. Eine Fernbedienung ist im Lieferumfang nicht enthalten. Die Steuerung erfolgt mithilfe einer Handy-App, die für *Android* und *iOS* über den entsprechenden Store kostenlos installiert werden kann. Es ist auch möglich, eine Fernbedienung zu verwenden, die als zusätzliche

Die Steuerung überlassen wir dem Jetson Nano.



### Kurzinfo

- » Günstige Einsteiger-Drohne programmieren
- » Personen tracken mit einfachem Python-Script
- » Objekterkennung auf dem Jetson Nano einrichten

### Checkliste



**Zeitaufwand:**  
8 Stunden



**Kosten:**  
200 Euro



**Programmieren:**  
bei Bedarf Python-Script anpassen

### Mehr zum Thema

- » Daniel Bachfeld, Intelligente Webcam für 5 Euro, *Make* 1/20, S. 28
- » Daniel Bachfeld, Gesichtsteuerung, *Make* 2/20, S. 16
- » Daniel Springwald, iGoBot, *Make Sonderheft* 2019, S. 70
- » Heinz Behling, JetBot, *Make Sonderheft* 2019, S. 40

### Material

- » DJI Ryze Tello
- » Jetson Nano
- » WiFi USB-Stick
- » SD-Karte
- » Gehäuse für den Jetson Nano (optional)
- » Powerbank 5 Volt

Alles zum Artikel im Web unter [make-magazin.de/x1ma](https://make-magazin.de/x1ma)

Option einzeln angeboten wird. Für die Steuerung der autonomen *Follow-Me-Drohne* ist bei diesem Projekt aber weder eine App noch eine teure Fernbedienung nötig. Stattdessen wird sie von einem Einplatinencomputer gesteuert, der den Videostream des Geräts auswertet, um einer Person zu folgen.

Es gibt die *Tello* in der einfachen Ausprägung und in der *EDU*-(Education)-Version. Ich verwende die *DJI Tello EDU*, die für einen Preis von 149 Euro angeboten wird. Die normale *Tello* ist schon für 99 Euro zu haben. Beide Versionen sind mit der hier verwendeten *DJI-TelloPy Library* kompatibel. Technisch sind beide Drohnen im Prinzip identisch, die *EDU*-

Version bringt allerdings das neuere SDK 2.0 mit. Der wichtigste Unterschied zwischen den beiden ist, dass die *Schwarm*-Funktion nur von der *DJI Tello EDU* unterstützt wird. Die brauchen wir hier aber auch nicht.

### Jetson Nano

Der *Jetson Nano* ist ein Einplatinencomputer mit vier CPU-Kernen und einer GPU mit 128 CUDA-Kernen. Im *Developer Kit* ist Gigabit-Ethernet, ein HDMI-Anschluss, eine CSI-Kamera Schnittstelle und vieles mehr enthalten. Er eignet sich sehr gut für den Einstieg in *Deep-Learning*. In der 2GB-RAM-Variante wird



Die nur 80g schwere DJI Ryze Tello

## Rechtliches

Seit 2021 gilt die neue *EU-Drohnenverordnung*. Seither benötigt man für fast alle Drohnen einen Drohnenführerschein, den man mit der Beantwortung einiger Fragen erlangen kann. Die *DJI Ryze Tello* unterliegt mit einem Aufstiegsge­wicht von unter 250g dieser Regelung allerdings nicht. Sie brauchen für den Flug mit der Follow-Me-Drohne also **keinen Drohnenführerschein** (schaden tut er natürlich trotzdem nicht). Der Drohnenführerschein für den Privatgebrauch kostet übrigens einmalig etwa 25 Euro.

Was Sie allerdings trotzdem benötigen, ist eine Registrierung als Drohnen-Betreiber beim Luftfahrt-Bundesamt und eine Drohnen-Haftpflichtversicherung – außer, Sie wollen ausschließlich in den eigenen vier Wänden fliegen.

Für den Flug mit der Drohne gelten ansonsten die üblichen Regeln: Fliegen Sie nur auf Sichtweite, nicht in Flugverbotszonen, nicht über fremden Wohngrundstücken und nicht über Personen, die Ihnen keine Erlaubnis dazu gegeben haben.

der *Jetson Nano* für einen Preis von gut 60 Euro angeboten.

Das Betriebssystem basiert auf Ubuntu, Linux-Vorkenntnisse sind für dieses Projekt also von Vorteil. Die von NVIDIA kostenlos angebotene Software enthält Bibliotheken und APIs zur Bildklassifizierung, Objekterkennung, Sprachverarbeitung, Videoverbesserung und weitere Features. Das SD-Karten-Image kann man auf der NVIDIA-Webseite

herunterladen und auf der SD-Karte installieren. Für den einfachen Einstieg in *Deep-Learning* und KI stellt NVIDIA außerdem das *JetPack-SDK* bereit.

## Installation des Jetson Nano

Bevor es mit der Verfolgungsjagd losgehen kann, müssen wir erst den *Jetson Nano* aufsetzen. Neben dem Einplatinencomputer ist ein

WiFi-USB-Sticker erforderlich, um über WLAN eine Verbindung zur Drohne aufbauen zu können. Für eine größere Reichweite empfehle ich einen Stick mit einer oder zwei externen Antennen. So erhält man bessere Videos ohne lästige Framedrops. Ich verwende zum Beispiel den *Maxesla USB WiFi Adapter* mit zwei 5dBi-Antennen. Die beiden Antennen steigern die Übertragungsrate und die Reichweite merklich.

Beginnen wir mit dem ersten Schritt: dem Download und der Installation des Images auf der SD-Karte. Das Image kann man auf der NVIDIA-Webseite (Link in der Kurzin­fo) herunterladen. Das *Flashen* der SD-Karte ist zum Beispiel mit dem kostenlosen Programm *balenaEtcher* möglich. Dazu muss man einfach nur das Programm herunterladen (Link in der Kurzin­fo) und es ausführen. Nach dem Start wählt man das vorher heruntergeladene Image und die SD-Karte aus, die sich im Schreib/Lesegerät befinden muss. Dann klickt man nur noch auf *Flash!* und das Image wird auf die SD-Karte geschrieben.

Ist das Image auf die SD-Karte kopiert, kann man sie in den *Jetson Nano* stecken und das erste Mal booten. Der *Jetson Nano* muss dazu an Bildschirm, Maus, Tastatur und Internet angeschlossen sein. Nach dem ersten Booten wird man aufgefordert die „End User License Agreements“ zu akzeptieren und die Systemkonfiguration (Sprache, Tastaturlayout, Land, Benutzername, etc.) vorzunehmen. Nach wenigen Minuten ist der *Jetson Nano* betriebsbereit und kann verwendet werden.

Dazu ein Hinweis: Wenn die Menü- und Desktop-Schriftart zu klein zum Lesen ist, hilft es in der Datei `/usr/share/lightdm/lightdm.conf.d/50-xserver-command.conf` die DPI-Auflösung zu ergänzen (`xserver-command=X-core -dpi 96`) und den *Jetson* neu zu booten.

## KI installieren

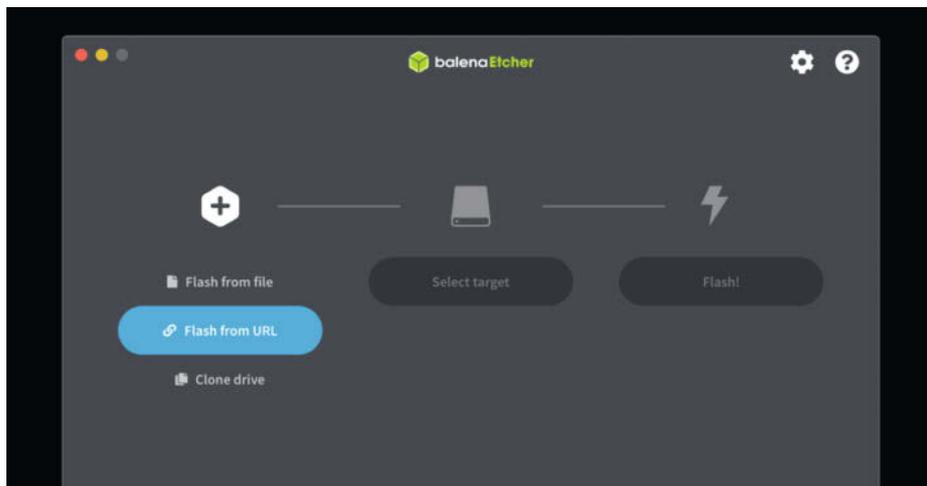
Als Nächstes müssen wir die nötigen Bibliotheken, KI-Modelle und Tools installieren. Für eine einfache Installation habe ich alles Nötige in einem *GitHub-Repository* zusammengetragen, das mit dem folgenden Git-Befehl einfach auf den *Jetson Nano* heruntergeladen werden kann:

```
git clone https://github.com/msoftware/tello-tracking.git
```

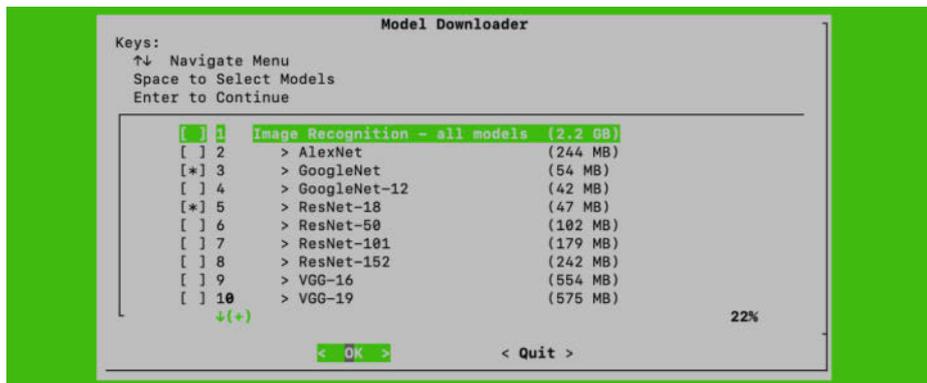
Wenn Sie noch keine Erfahrung mit Git und GitHub haben, finden Sie eine Anleitung in der Kurzin­fo. Die Installation und Einrichtung von Git sollte nur ein paar Minuten in Anspruch nehmen.

Da die Installation der einzelnen Pakete sehr aufwändig und fehleranfällig ist, habe ich mehrere kleine Skripte im *Setup*-Ordner des *Git-Repositories* abgelegt, die alle nacheinander im Terminal aufgerufen werden müssen.

– *01\_opencv\_part1.sh* – Vorbereitung des *Jetson Nano* auf die Installation von *OpenCV*



Mit balenaEtcher schreiben wir das Image auf die SD-Karte.



Die Auswahl an KI-Modellen ist groß.

```

NVIDIA Jetson Nano (Developer Kit Version) - Jetpack 4.5.1 [L4T 32.5.1]
CPU1 [|||||] Schedu11 - 69% 1.5GHz
CPU2 [|||||] Schedu11 - 62% 1.5GHz
CPU3 [|||||] Schedu11 - 63% 1.5GHz
CPU4 [|||||] Schedu11 - 78% 1.5GHz

Mem [|||||] 1.70/2.000 (1fb 2x1MB)
1mb [|||||] 0.0v/252.0kb (1fb 252kb)
Swp [|||||] 0.07108/7.108 (cached 4MB)
EMC [|||||] 25% 1.6GHz

GPU [|||||] [99%] 921MHz
Dsk [|||||] 28.708/29.108

[Info]
UpT: 1 days 23:47:48
FAN [|||||] % Tam 0%
Jetson Clocks: inactive
NV Power[0]: MAXN
[HW engines]
APE: 25MHz
NVENC: [OFF] NVDEC: [OFF]
NVJPG: [OFF]

[Sensor] [Temp]
AO 46.00C
CPU 48.00C
GPU 39.00C
PLL 36.00C
thermal 39.00C

[ALL] 2GPU 3CPU 4MEM 5CTRL 6INFO Quit
Raffaello Bonghi
    
```

jtop überwacht den Zustand des Jetson Nano.

- 02\_opencv\_part2.sh - Installation der OpenCV-Dependencies
- 03\_opencv\_part3.sh - Download und Installation von OpenCV 4.5
- 04\_djitellopy2.sh - Download und Installation der djitellopy2 Python-Bibliothek und aller Abhängigkeiten
- 05\_jetson-inference.sh - Installation der Jetson-Inference-Modelle für die Objekterkennung

Vor allem das Skript 03\_opencv\_part3.sh ist sehr aufwändig. Es übersetzt OpenCV aus dem Quellcode und nimmt für die Ausführung mehrere Stunden in Anspruch. Da hier wegen des vergleichsweise kleinen RAM des Nano intensiver Gebrauch vom Swap-Speicher gemacht wird, ist es unbedingt erforderlich, eine sehr schnelle und qualitativ hochwertige SD-Karte zu verwenden. Ich verwende eine microSDXC-Speicherkarte mit 64 GB und einer maximalen Transferrgeschwindigkeit von 120MB/s und dem A1-Standard, der vor allem bei Schreib- und Leseoperationen von kleinen Datenblöcken sehr effizient ist. Trotzdem dauert das Übersetzen von OpenCV 4.5 inklusive der benötigten CUDA- und GTK-Unterstützung mehr als fünf Stunden.

Gelegentlich wird man im Installationsprozess aufgefordert, sein Passwort einzugeben oder die KI-Modelle auszuwählen, die installiert werden sollen. Hier muss man nichts Besonderes beachten: Das benötigte Jetson-Inference Model (SSD-MobileNet-v2) ist schon vorausgewählt. Wer eine langsame Internetleitung hat, sollte möglichst wenig Modelle auswählen. Für dieses Projekt wird nur das Deep-Learning Modell SSD-MobileNet-v2 verwendet. Alle anderen Modelle sind nicht nötig, können aber natürlich für Erweiterungen oder andere Projekte eingesetzt werden.

Nach der Installation sind folgende Komponenten auf der SD-Karte zu finden: Die openCV-Bibliothek für Bildverarbeitung und Computer Vision, die djitellopy2-Bibliothek für die Kommunikation mit der Tello-Drohne sowie die jetson-inference-Deep-Learning-Inferenznetzwerke und Deep-Vision-Grundele-

mente. Zusätzlich wird mit dem Paket jetson-stats noch das kleine Programm jtop installiert, mit dem man sehr gut die Auslastung der CPU-Kerne, der GPU und weitere Daten überprüfen kann.

### Programm der Follow-Me-Drohne

Das Programm zum Tracking von Personen mit der Tello-Drohne befindet sich in der Datei tello-tracking.py. Neben dem Python-Script sind noch eine Reihe anderer Dateien im Github-Repository: Die Datei pytrt ist eine Library, die ich dem Github-Repository tensorsrt\_demos von JK Jung entnommen habe, der es unter der MIT Lizenz veröffentlicht hat. Das Skript wifi-connect.sh verbindet den Jetson Nano mit dem WLAN der Tello-Drohne. Und wifi-list.sh, wifi-on.sh, wifi-off.sh, wifi-scan.sh und wifi-state.sh sind eine Reihe von kleineren Shell-Skripten zum einfachen WiFi-Management des Jetson Nano auf der Konsole.

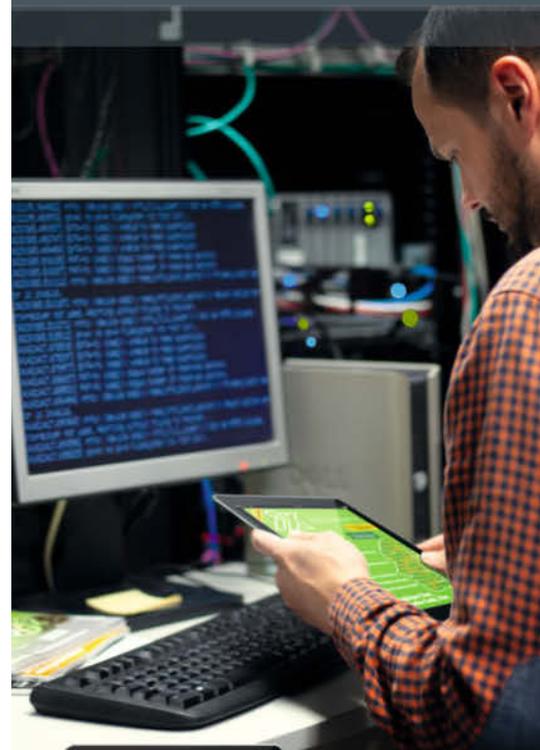
Die Tello initialisiert nach dem Start ein eigenes offenes Netzwerk, das man unter dem Namen TELLO-[XXXX] findet, wenn man den Namen nicht in der Smartphone-App geändert hat. Diese SSID muss in die Datei config.sh eingetragen werden, damit das wifi-connect.sh-Shell-Skript korrekt funktioniert. Ist die SSID konfiguriert und die Drohne eingeschaltet, kann man mit den beiden Kommandos wifi-scan.sh und wifi-connect.sh eine Verbindung aufbauen und das Python3-Programm tello-tracking.py aufrufen. Der erste Start von tello-tracking.py dauert einige Zeit, da das verwendete Netzwerk bei der ersten Nutzung für den Jetson Nano optimiert wird.

### Die Bilderkennung

Und woher weiß die Drohne, dass sie uns folgen soll? Darum kümmert sich das vormodellierte und vortrainierte neuronale Netzwerk namens MobileNet – ein Objektdetektor, der speziell für die Bedürfnisse vom mobilen und eingebetteten Bildverarbeitungsanwendun-

Es gibt 10 Arten von Menschen.

iX-Leser und die anderen.



Jetzt Mini-Abo testen:

3 digitale Ausgaben + Bluetooth-Tastatur nur 16,50 €

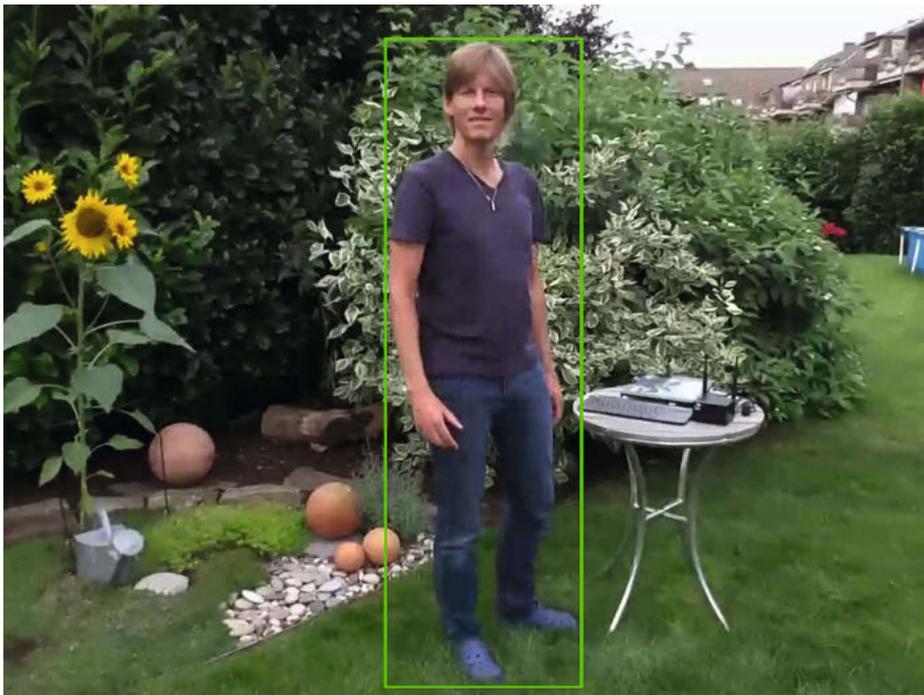
[www.ix.de/digital-testen](http://www.ix.de/digital-testen)



[www.ix.de/digital-testen](http://www.ix.de/digital-testen)

leserservice@heise.de

49 (0)541 800 09 120



Die Drohne erkennt Menschen automatisch und folgt ihnen.

gen entwickelt wurde. Das hier verwendete Netzwerk wurde auf dem Coco-Datensatz trainiert, mit dem 91 unterschiedliche Objekte identifiziert werden können. Darunter sind neben Personen auch Katzen, Hunde, Äpfel, Bücher und viele andere. Die Liste aller Coco-Label kann man auf GitHub (Link in der Kurzinformatio) finden. Coco steht für „Common Objects in Context“, es gibt aber natürlich auch noch weitaus spezialisiertere Datensätze – ein paar stellen wir später vor.

## Den Flug starten

Mit *tello-tracking.py* wird die Drohne gestartet und der Videostream der Drohne abgerufen. Sobald die Drohne in der Luft ist, sucht sie aktiv nach einer Person im Bild und dreht sich langsam, ohne dass sie sich vom Fleck bewegt. Erst wenn eine Person im Bild erscheint, bewegt sich die Drohne – und versucht, der Person zu folgen und sie im Bild zu halten.

## Mehr zu KI auf Mikrocontrollern

Dieses Projekt verbindet diverse Libraries, um die Verfolgungs-Automatik umzusetzen, wie die *OpenCV*-Bibliothek für die Bildverarbeitung, *MobileNet* als Objektdetektor sowie die *Deep-Learning*-Bibliothek *jetson-inference*. Wir erklären hier nicht die Funktionsweise jeder dieser Quellen im Detail, doch viele von ihnen haben schon in anderen unserer Projekte Anwendung gefunden.

Wie Objekterkennung, Trainings-Algorithmen und KI auf Mikrocontrollern im Allgemeinen funktionieren, erklären wir in diesem Heft im Artikel *KI für den ESP32* auf Seite 48 ganz ausführlich.

Auf den Datensatz von *MobileNet* greift zum Beispiel auch unsere *ESP32-CAM* mit

Gesichtserkennung aus der Make 1/20 zu. Im Nachfolge-Artikel *Gesichtssteuerung* aus der Make 2/20 erklären wir auch, wie Sie die Objekterkennung speziell auf ihr Gesicht trainieren können.

Wie man eigene KI-Projekte auf dem *Jetson Nano* umsetzen kann, erkunden wir am Beispiel des *JetBots* im Robotik-Sonderheft von 2019. Hier trainieren wir den Bot mit selbst geschossenen Fotos auf seine Umgebung.

Eine Bilderkennung mit *OpenCV* besitzt zum Beispiel auch der *iGoBot* aus unserem Robotik-Sonderheft, der dank ihr ein Go-Brett auslesen und das Spiel spielen kann. Die Links zu diesen Artikeln finden Sie in der Kurzinformatio.

## Das Programm individualisieren

Das Python-Programm kann man ganz beliebig ergänzen und an die eigenen Bedürfnisse anpassen. Folgende Konfigurationsparameter kann man zum Beispiel (fast) ohne Programmierkenntnisse ändern:

- `low_bat = 15` – Liegt das Batteriewert unter 15 Prozent, landet die Drohne automatisch.
- `enableVideoRecorder = True` – Sollen die Videos aufgezeichnet werden, dann `True`, ansonsten `False` auswählen.
- `video_fps = 15` – Die Framerate, mit der die Videos aufgezeichnet werden. Auf dem *Jetson Nano* ist 15 schon die maximale Framerate. Eine höhere Framerate ist zum Beispiel auf dem *Jetson Xavier NX Developer Kit* oder dem *Jetson AGX Xavier* möglich.
- `max_yaw_velocity = 50` – Die maximale Rotationsgeschwindigkeit der Drohne.
- `max_up_down_velocity = 40` – Die maximale Geschwindigkeit für Auf- und Ab-Bewegungen.
- `max_forward_backward_velocity = 40` – Die maximale Geschwindigkeit für Vor- und Zurück-Bewegungen.

## Ideen für weitere Projekte

Eine nette Übung könnte es zum Beispiel sein, die Distanz der Kamera zum aufgenommenen Objekt von der Halbtotale auf Nah oder Totale zu ändern oder das *SSD-MobileNet-v2*-Netzwerk durch das *SSD-Inception-v2*-Netzwerk zu ersetzen, das auf die Erkennung von Fahrzeugen spezialisiert ist. Das *SSD-Inception*-Netzwerk kann man auch mit dem Shell-Script *download-models.sh* herunterladen. Alternativ könnte man auch mit anderen Modellen wie beispielsweise *DenseDepth* (Link in der Kurzinformatio) eine Drohne programmieren, die Hindernisse erkennen kann oder eine grobe Karte der Umgebung erstellt.

## Fazit

Wie man sieht, ist es dank der vielen frei verfügbaren Software-Komponenten und der vortrainierten *Deep-Learning*-Modelle kein Problem, eine Drohne dazu zu bringen, einer Person auf Schritt und Tritt zu folgen. Dabei kann sie sogar filmen, ohne dass ein Pilot die Drohne steuern muss. Mit dem *Jetson Nano* hat man einen günstigen Einplatinencomputer, der kleinere *Deep-Learning*-Modelle in beeindruckender Geschwindigkeit und äußerst energieeffizient durchführen kann. Die *DJI Ryze Tello* steht dank der verschiedenen Sensoren sehr gut in der Luft und kann auch bei leichtem Wind problemlos die Position halten, ohne dass man manuell eingreifen muss. So steht dem Flugvergnügen nichts mehr im Weg. —*rehu*

# Klein – aber oho!

Mit MicroPython spielend leicht  
Mikrocontroller programmieren

und exklusiv im heise shop  
**NEU**



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 CO<sub>2</sub>-Ampel und mehr.

Auch als einzelnes Heft erhältlich.

[shop.heise.de/make-micropython](http://shop.heise.de/make-micropython)

Heft + PDF  
für nur

19,90 € >

heise Shop

[shop.heise.de/make-micropython](http://shop.heise.de/make-micropython) >

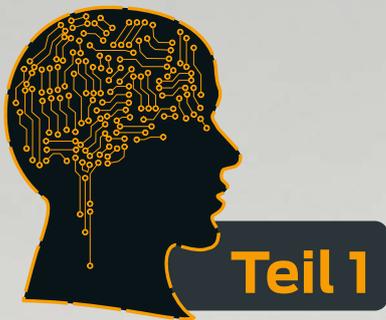
> Generell portofreie Lieferung für Heise Medien- oder Maker Media Zeitschriften-Abonnenten oder ab einem Einkaufswert von 20 €. Nur solange der Vorrat reicht. Preisänderungen vorbehalten.

© Copyright by Maker Media GmbH

# KI für den ESP32

Auch auf Mikrocontrollern mit vergleichsweise wenig RAM und geringem CPU-Takt können neuronale Netze anspruchsvolle Aufgabe übernehmen. Wir zeigen in einer dreiteiligen Artikelstrecke Schritt für Schritt wie das funktioniert.

von Josef Müller



Wenn man an praktische Anwendungen für neuronale Netze denkt, kommt einem schnell der Gedanke an Sprachverarbeitung in großen Rechenzentren wie bei *Amazon Alexa* oder *Google Echo* oder auch Bildverarbeitung für autonomes Fahren. Beides verbindet man mit komplexer Software und teurer Hardware, die nur mit viel Aufwand programmiert und betrieben werden können.

Das muss aber dank neuer und moderner Bibliotheken mittlerweile nicht immer so sein. In der *Make 2/21* haben wir ein Beispiel vorgestellt, welches eine neuronale Anwendung auf einem günstigen Mikrocomputer realisiert, ganz ohne Cloud-Anbindung. Es wurde eine neuronale Bilderkennung auf einem ESP32 mit integrierter Kamera kombiniert. Damit haben wir einen analogen Wasserzähler quasi digitalisiert und sehr günstig an die Datenerfassung und Haussteuerung angebunden <sup>1</sup>. Das Ganze ist für weniger als 30 Euro sehr kostengünstig realisierbar. Die Idee dahinter lässt sich mit ein wenig Fantasie auch auf andere Bereiche anwenden. Das Projekt fand sehr viel Resonanz, darunter auch viele Nachfragen, wie es im Detail funktioniert und wie man es auf eigene Ziffern und Zähler anpassen könnte.

Wir wollen deshalb in einer dreiteiligen Artikelreihe die wesentlichen Grundlagen hinter dem Projekt erläutern und Schritt für Schritt erklären, wie man selbst Bilderkennung mit neuronalen Netzen entwickelt und auf dem ESP32 zum Laufen bekommt. Am Ende der Artikelserie solltet ihr idealerweise ein selbst trainiertes neuronales Netz zur Bilderkennung mit wenigen Zeilen Code auf eurem ESP32 zum Laufen bringen. Darum herum könnt ihr dann hoffentlich die eigenen Ideen und Anwendung verwirklichen.

### Alt und neu

Der große Vorteil von neuronalen Netzen in der Bilderkennung liegt darin, dass man recht einfach Objekte erkennen kann, ohne die Regeln dafür genau vorgeben zu müssen. Will man einem Menschen das Aussehen eines Dreiecks erklären, kann man ihm Dinge über Ecken, Seiten und Winkel und mögliche Konstellationen erzählen. Oder man zeigt ihm mehrere Bilder von Dreiecken und erklärt ihm, dies seien Dreiecke. In letzterem Fall werden die Regeln implizit im neuronalen Netz gelernt und hinterlegt.

Um dies zu verdeutlichen, nehmen wir ein komplexeres Beispiel: ein 20 x 30 Pixel großes Schwarzweißbild, welches eine Ziffer zwischen 0 und 9 zeigt. Dafür ein Regelwerk der Art „wenn dieses Pixel weiß und/oder diese schwarz ist, dann ...“ aufzustellen, um die richtige Ziffer zu erkennen, ist extrem aufwändig.

Selbst wenn die Zifferngröße und Position fix ist, sind dies doch recht viele und verschachtelte Regeln. Wenn dann die Ziffern auch noch an unterschiedlichen Positionen

## Kurzinfo

- » Bilder für Neuronale Netze vorbereiten
- » Python-Skripte mit Jupyter-Notebooks nachvollziehen
- » Mehr Trainingsdaten dank Data Augmentation

### Checkliste



**Kosten:**  
0 Euro



**Zeitaufwand:**  
2 Stunden



**Software:**  
Python und Anaconda Navigator

### Mehr zum Thema

- » Josef Müller, ESP32CAM liest Wasseruhr, *Make 2/21*, S. 14
- » Pina Merkert, Bilderstürmer, c't 16/18, S. 174
- » Daniel Bachfeld, Einstieg in KI, *Make 6/18*, S. 36

Alles zum Artikel im Web unter [make-magazin.de/x665](http://make-magazin.de/x665)

liegen und zu guter Letzt auch noch als Farbbild vorliegen, dann kann man sich leicht vorstellen, dass klassische Regelwerke mit „wenn ... dann ... sonst“-Konstruktionen sehr schnell überlastet sind.

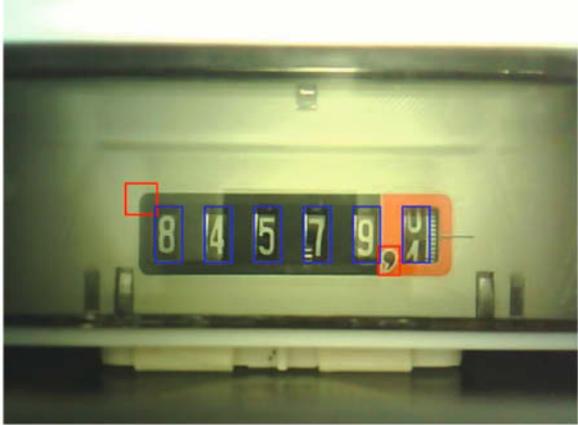
In <sup>2</sup> sieht man unterschiedliche Ziffern 3, die alle durch ein neuronales Netz, welches

im Wasserzähler verwendet wird, zuverlässig korrekt zugeordnet werden. Auch ein neuronales Netz kann man sich als sehr komplexes Regelwerk vorstellen. Auch hier gibt es Strukturen, die mit einem „wenn ... dann ...“-Konstrukt und einer Art Regelwerk vergleichbar sind.

## Digitizer - AI on the edge

An ESP32 all inclusive neural network recognition system for meter digitalization

Overview	Configuration	Recognition	File Server	System
----------	---------------	-------------	-------------	--------



<b>Value:</b>
84579.8
<b>Previous Value:</b>
84579.8
<b>Raw Value:</b>
84579.N
<b>Error:</b>
no error
<small>Last Page Refresh:07:38:29</small>



<sup>1</sup> Der Zählerstand wird per neuronalem Netz digitalisiert.

## Artikelserie

Um sich dem Thema „KI auf ESP32“ schrittweise zu nähern, haben wir das Projekt zur Verwendung eines neuronalen Netzes auf drei Artikel aufgeteilt:

In **Teil 1** in diesem Heft geht es zunächst um die Beschreibung des Anschauungsbeispiels und hier im Schwerpunkt um die Trainingsdaten. Das Design und Training finden in einer Python-Umgebung auf einem normalen PC statt. Ziel ist es, Rohdaten in das notwendige Format zu konvertieren, die Eingangsbilder und Daten für das Training zu handhaben und das Prinzip der *Data Augmentation* zu verstehen. Dazu wird auf einem PC eine Python-Umgebung installiert und die grundlegenden Bibliotheken für die Bildvorbereitung und -konditionierung werden vorbereitet.

In **Teil 2** geht es um den Background, das Design und das Training des neuronalen Netzes. Auch dies findet aus Performance-Gründen noch in der Python-Umgebung des PCs statt. Das trainierte Netz wird für den Transfer auf den ESP32 vorbereitet. Am Ende dieses Teils ist man in der Lage, sein erstes selbst trainiertes neuronales Netz auf dem PC zu verwenden und eigene Bilder zu klassifizieren.

In **Teil 3** geht es dann konkret um ein neuronales Netz auf dem ESP32. Dort zeigen wir, wie man das mit Python erzeugte Modell eines neuronalen Netzes mit wenigen Zeilen C-Code auf den ESP32 portiert. Daraus lassen sich viele Ideen für anderen Netze ableiten, womit der Startschuss für eigene Projekte fällt.



2 Unterschiedliche Ziffernarten bei Wasser- und Stromzählern



3 Trainingsdaten für die Ziffern 0 bis 9 sowie nicht eindeutige Zwischenwerte (Not-a-Number)

Der wesentliche Unterschied zwischen klassischen Regelwerken und neuronalen Netzen liegt zum einen in der Struktur, wie die Informationen verarbeitet werden, und zum zweiten, wie die Regeln trainiert werden. Es werden nur Eingangsbilder und Angaben zu den erwarteten Ergebnissen zu diesen Bildern benötigt. Über einen Trainingsalgorithmus werden die Regeln sequenziell so angepasst, dass mit immer größerer Wahrscheinlichkeit das gewünschte Ergebnis herauskommt. Die Regeln entsprechen dann den Parametern des Netzes, etwa den sogenannten *Gewichten*. Weitere Details dazu erläutern wir dann im zweiten Teil.

Die Formulierung „mit immer größerer Wahrscheinlichkeit“ zeigt einen wesentlichen Knackpunkt dieser Art der Bildverarbeitung auf: Das Training der neuronalen Netze enthält einen zufälligen Anteil, sodass quasi kein trainiertes Netz identisch mit einem anderen ist, auch wenn dieselben Daten zum Training genommen werden. Zum zweiten kann man die internen Zustände im neuronalen Netz nicht klassischen Eigenschaften zuordnen, sodass man eigentlich nur anhand der Statistik der Endergebnisse eine Aussage über die Qualität machen kann.

### ESP32 und TensorFlow

In den letzten Jahren gab es aber sowohl auf der Hardware- wie auch auf der Softwareseite wesentliche Weiterentwicklungen, womit es jetzt möglich ist, künstliche Intelligenz auch in DIY-Projekten mit günstiger Hardware und verhältnismäßig einfach anzuwendender Soft-

ware zu verwenden. Auf der Hardwareseite gibt es den ESP32, der mit bis zu 240MHz Takt-rate und 4MB PSRAM genügend Performance hat, um mittels neuronaler Netze Bilder zu erkennen.

Auf der Softwareseite ist mit der sehr mächtigen und frei verfügbaren Bibliothek *TensorFlow* von Google eine gut dokumentierte Implementierung für verschiedene Plattformen verfügbar. Seit einiger Zeit gibt es mit *TFlite* bzw. *tfmicro* eine dedizierte Ableitung für C-basierte Mikrocontroller-Entwicklungsumgebungen.

Der typische Wasser- oder Stromzähler besteht aus mehreren einzelnen Durchlauf-rädern, die jeweils eine Ziffer des Zählerstandes darstellen. Das neuronale Netz soll den Zählerstand in digitaler Form zur Verfügung stellen. Um das neuronale Netz so einfach wie möglich und zugleich auch so flexibel wie möglich zu halten, soll es nur das Bild einer einzelnen Ziffer digitalisieren.

Das Thema Bildaufnahme und Zerlegung in die einzelnen Ziffern klammern wir hier mal aus. Wir setzen fertige Bilder der einzelnen Ziffern 0 bis 9 voraus. Da die Ziffern auf umlaufenden Ringen gedruckt sind, gibt es auch immer wieder unklare Zifferndarstellungen, wenn der Zähler zwischen zwei Zahlen steht. Dieser Status soll auch erkannt werden und mit einer speziellen 11. Klassifizierung, der sogenannten *Not-A-Number* (NaN) erkannt werden. 3 zeigt einen Satz typischer Beispielm-bilder und die Klasse NaN.

### Vorbereitung

Das Ergebnis eines neuronalen Netzes ist immer nur so gut, wie die Eingangsdaten, die man zum Training zur Verfügung hat. Daher sollte man hier durchaus etwas Aufwand investieren. Idealerweise verwendet man zur Aufnahme der Bilder schon die gleichen Beleuchtungs- und Kameraeinstellungen, die später im finalen System Verwendung finden. Beim Wasser- bzw. Stromzähler wird mittels der OV2640-Kamera in der ESP32CAM die Oberfläche des Wasserzählers abfotografiert und dann in die einzelnen Ziffern zerlegt. Wichtig ist hier, darauf zu achten, dass die Orientierung, Position und Größe der Ziffer im einzelnen Bild vergleichbar sind. Als grobe Regel habe ich die Ziffern immer zentral mittig platziert und jeweils an der Seite einen Rand von ca. 15% gelassen (siehe auch 3).

Um später das neuronale Netz zu trainieren, muss man natürlich auch wissen, was auf dem einzelnen Bild zu sehen ist. Dieser Vorgang wird als *Labeling* (= Beschriftung) bezeichnet. Dabei wird jedem Bild der Inhalt zugeordnet, den man als Ausgabe des neuronalen Netzes erwartet. Dieser kann bei komplexeren Aufgaben auch aus vielen Informationen bestehen (Objekt, Farbe, Position, Objektdetails, ...).

In umfangreichen Aufgaben kann das Labeling eine sehr aufwendige Aufgabe sein (z. B. Sprachdateien für Texterkennung). Hier ist es recht einfach: Man benötigt zu jedem Bild nur die Zahl, welche darauf zu sehen ist.

Die Verknüpfung von Bild und Inhalt kann auf verschiedene Arten gespeichert werden. Hier habe ich mich der Einfachheit halber entschieden, sie direkt an erster Position im Dateinamen zu speichern. So kann man später ohne Hilfsdatei etc. einfach den Zielwert direkt aus dem Dateinamen ablesen.

An dieser Stelle ein Vorgriff auf das Training: Wenn man feststellt, dass eine Ziffer immer hartnäckig falsch erkannt wird, lohnt es sich durchaus mal zu prüfen, ob sich beim Labeling des entsprechenden Bildes ein Fehler eingeschlichen hat. Wie schnell hat man sich dort vertippt und schon trainiert man das falsch beschriftete Bild einer 8 mit dem Zielwert einer 9.

### Variationen

Idealerweise decken die Trainingsbilder des neuronalen Netzes alle möglichen vorkommenden Variationen der Eingangsbilder umfangreich ab. Das ist für die regulären Ziffern (0, 1, ... 9) noch einfach machbar. Schwierig wird es schon bei der NaN-Klasse, denn man kann natürlich nicht von jedem möglichen Zwischenstand ein Bild erzeugen. Das ist sowohl was die Menge, als auch die Erzeugung der Bilder angeht, nicht möglich. Hier begnügt man sich mit typischen Beispielbildern. Daraus lernt das neuronale Netz dann die Abstrahierung auf die allgemeine Regel. Gleiches gilt auch für die anderen Eigenschaften eines Bildes, wie Helligkeit, Position der Ziffern, unterschiedliche Umgebungen etc.

Je mehr Trainingsdaten vorhanden sind, desto zuverlässiger erkennt das Netz später Ziffern. Nehmen wir mal an, es würde pro Ziffer nur ein einziges Bild zum Training verwendet. Und zufälligerweise wurde das Bild der Ziffer 0 bei etwas helleren Belichtungsverhältnissen aufgenommen (es war gerade das Kellerlicht an). Jetzt trainieren wir das Netz und

alle Trainingsziffern werden hervorragend erkannt.

Voller Zuversicht verwenden wir das trainierte Netz in der Praxis. Plötzlich stellen wir aber fest, dass auch immer wieder andere Ziffern als 0 erkannt werden. Bei genauerer Analyse merken wir, dass dies immer dann der Fall ist, wenn das Kellerlicht gerade an ist. Unser neuronales Netz hat also gar nicht die Ziffer 0 gelernt, sondern erkennt eigentlich den Zustand „helleres Bild“.

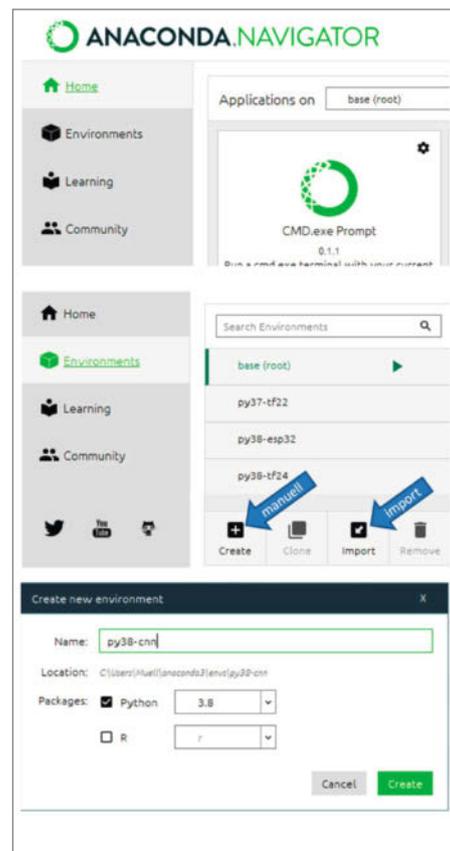
Da man nicht in das Innere des neuronalen Netzes reinschauen kann (zumindest nicht mit einfachen Mitteln), weiß man nicht, welche Eigenschaften der Bilder tatsächlich zur Klassifizierung führen. Daher ist es wichtig, möglichst viele dieser Variationen auch in den Trainingsdaten abzubilden. Je größer dabei das neuronale Netz ist, desto mehr Trainingsdaten werden auch benötigt.

Um nun nicht selbst hunderte Bilder für alle möglichen Varianten manuell aufnehmen zu müssen, gibt es die sehr mächtige Möglichkeit der *Data Augmentation*. Man vervielfältigt die Trainingsdaten, indem man typische Variationen künstlich erzeugt und in den Bildern z. B. Helligkeit und Rotation leicht verändert. Dies wird im letzten Abschnitt noch genauer erklärt, vorher richten wir aber noch die Trainingsumgebung ein, um die Trainingsbilder aufzubereiten.

### Umgebung

Damit ihr euch nicht mit der teilweise doch recht aufwendigen Datenaufnahme beschäftigen müsst, haben wir für euch im GitHub-Repository einen ersten Datensatz an Rohdaten zur Verfügung gestellt. Dieser ist absichtlich nicht besonders groß und umfangreich, damit das Training auch im zweiten Teil nicht zu lange dauert und man verschiedene Parameter spielerisch variieren kann.

Für die Data Augmentation und zum Trainieren unseres Netzes nutzen wir eine Python-Umgebung mit *TensorFlow*. Unter Windows kann man Python sehr einfach mittels der



4 Mit dem Anaconda Navigator kann man verschiedene Entwicklungsumgebungen installieren und verwalten.

Plattform *Anaconda* installieren, verwenden und verwalten 4. Sie lässt sich selbst recht einfach installieren, man kann unterschiedliche Testumgebungen definieren und auch eine sogenannte *Jupyter*-Umgebung ist verfügbar. Mit Jupyter kann man im Browser quasi kommentierte Python-Skripte (*Notebooks* genannt) Schritt für Schritt ablaufen und sich Ergebnisse anzeigen lassen. Die genaue Installations- und Bedienungsanleitung und ein Video dazu findet ihr unter dem Link in der Kurzinfor.

```

from PIL import Image
image = Image.open('Size_Example_1.jpg')

size_x = 40
size_y = 64

for i in range(4):
    resize = image.resize((size_x, size_y), Image.NEAREST)
    print("Größe: " + str(size_x) + " x " + str(size_y) + " Pixel")
    display(resize)
    size_x = int(size_x / 2)
    size_y = int(size_y / 2)
        
```

Größe: 40 x 64 Pixel



Größe: 20 x 32 Pixel



Größe: 10 x 16 Pixel



Größe: 5 x 8 Pixel



5 Skalierung des Testbildes

```

import glob
import os
from PIL import Image

Input_dir = 'bilder_original'
Output_dir = 'bilder_resize'
target_size_x = 20
target_size_y = 32

#LÖSCHEN hier nicht abgedruckt

files = glob.glob(Input_dir + '/*.jpg')
for aktfile in files:
    print(aktfile)
    image = Image.open(aktfile)
    image = image.resize((target_size_x, target_size_y), Image.NEAREST)
    base=os.path.basename(aktfile)
    save_name = Output_dir + '/' + base
    image.save(save_name, "JPEG")

```

6 Skalierung aller Trainingsbilder auf eine einheitliche Größe

## Trainingsbilder

Oft haben die Originaldaten unterschiedliche Größen, etwa weil der Ausschnitt aus den Rohdaten variiert oder einfach eine viel größere Auflösung bietet, als für die neuronale Berechnung notwendig ist. Für das später verwendete neuronale Netz muss eine einheitliche Eingangsgröße verwendet werden. Generell gilt, je kleiner die Bildgröße, desto schneller ist auch das spätere Training und die Erkennung. Andererseits gehen Informationen verloren, wenn die Bilder zu klein werden. Für meine neuronalen Netze hat sich folgende Daumenregel bewährt: Wenn *ich* auf dem verkleinerten Bild das Merkmal noch erkennen kann, dann ist es auch ausreichend für die automatische Erkennung durch ein neuronales Netz.

Das erste Python-Skript in Form eines Jupyter-Notebookes (*1\_Test\_Groesse\_Trainingsbilder.ipynb*) lädt mittels der *Pillow*-Bildverarbeitungsbibliothek das Beispielbild und gibt immer kleiner skalierte Versionen aus 5. Zu beachten ist hier, dass als Skalierungsmethode keine Interpolation, sondern die Methode *Nearest* angewendet wird. Dies entspricht am ehesten der später auf dem ESP32 verwendeten Algorithmen. Man kann erkennen, dass in der kleinsten Version (5 × 8 Pixel) die Ziffer nicht mehr lesbar ist. Die Version 10 × 16 Pixel lässt sich noch sehr gut lesen und wäre somit gut geeignet. Wenn man an spätere Variationen mit deutlich dünneren Schriftarten denkt, so ist es sinnvoll, nicht die kleinste mögliche Größe zu nehmen, sondern mit der Größe 20 × 32 zu

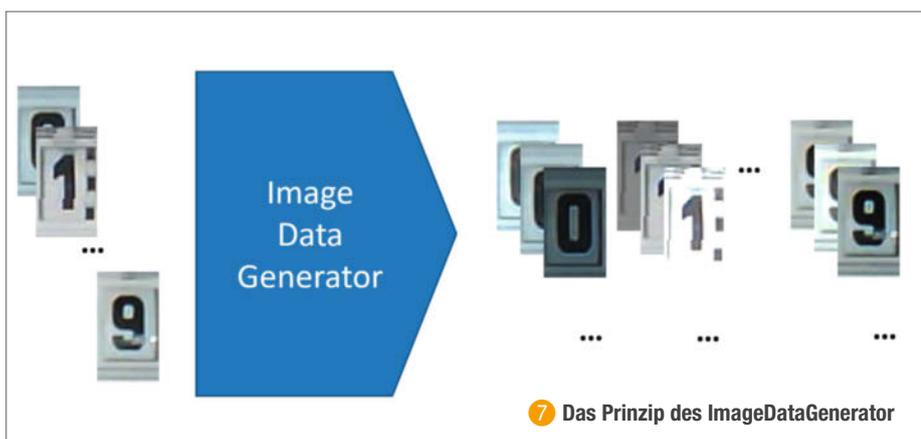
starten. Dies kann man mit dem Beispielbild 2 auch mal testen.

Nachdem die Größe nun auf 20 × 32 Pixel festgelegt ist, werden mit einem zweiten Skript alle Trainingsbilder auf die einheitliche Größe skaliert. Das ganze findet sich im Skript (*2\_Skalierung\_Trainingsdaten.ipynb*) 6. Zunächst werden neben der Bildverarbeitungsbibliothek *Pillow* auch noch zwei Bibliotheken für die Dateibearbeitung geladen (*glob*, *os*). Im ersten Schritt wird das Zielverzeichnis (*Output\_dir*) gelöscht und dann werden in einer zweiten Schleife alle Originaldateien geladen, skaliert und abgespeichert.

## Data Augmentation

Praktischerweise bringt die Bibliothek von TensorFlow bereits Funktionen für Data Augmentation mit. Allgemein stellt man unter TensorFlow die Trainingsdaten als großes mehrdimensionales Array (Bilddaten und Zielwerten) bereit. Will man die Data Augmentation anwenden, muss man noch einen sogenannten *ImageDataGenerator* vorschalten. Diesen kann man sich wie einen Modulator vorstellen, der die Bilddaten als Eingangswerte bekommt, mit verschiedenen einstellbaren Eigenschaften modifiziert und dann wieder als Bild ausgibt 7.

Ausprobieren kann man das Beispiel der Helligkeitsvariation mit dem Skript *3\_Augmentation\_Helligkeit.ipynb* (siehe Link). Zunächst muss neben der Bildverarbeitungsbibliothek noch die Funktion *DataIma-*



7 Das Prinzip des ImageDataGenerator

geGenerator aus der TensorFlow-Bibliothek geladen werden. Es wird wieder das bisherige Beispielbild verwendet. Das neuronale Netz und auch der Datengenerator erwarten ein Array von Trainingsdaten, also genau genommen ein 4-dimensionales Array, wobei der erste Index über die Anzahl der Trainingsbilder iteriert und die anderen die Bildinformationen enthalten (x, y, RGB). Zu Demonstrationszwecken zeigen wir hier nur ein einzelnes Trainingsbild, weshalb einige Operationen mit der Python-Bibliothek *numpy* notwendig sind, um es (mit den Funktionen *array* und *expand\_dims*) in ein Format umzuwandeln, mit dem der *ImageDataGenerator* umgehen kann.

Der Generator lässt sich recht einfach nutzen, wie in den folgenden Zeilen zu sehen ist:

```
Brightness_Range = 0.5
datagen = ImageDataGenerator(
    brightness_range = [1-Brightness_Range,1+Brightness_Range])
iterator = datagen.flow(sample, batch_size=1)
```

Zunächst muss man definieren, welcher Parameter in welchen Grenzen variiert werden soll. Der Parameter für die Helligkeit heißt *brightness\_range* (die klein geschriebene Version im Listing) und erwartet ein Array mit der minimalen und maximalen Helligkeitsveränderung (1 = unverändert, 0 = dunkel, 2 = weiß). Hier wird eine Variation mit 50% (=0.5) verwendet. Jetzt muss man noch definieren, auf welche Bilddaten die Variation angewendet werden soll. Dies ist hier als Iterator definiert und besteht nur aus dem einzelnen Testbild *sample* (mit der *batch\_size=1*) – zu letzterem im nächsten Teil dann mehr.

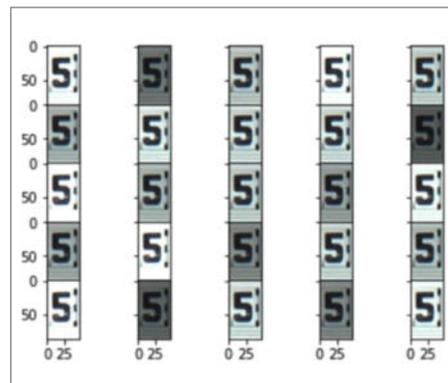
Jetzt bekommt man über die Funktion *iterator.next()* immer ein neues Bild, auf welches zufällig die Helligkeitsvariation angewendet wurde. Da es nur ein einziges Eingangsbild gibt, sieht man sehr schön, wie die Funktion arbeitet und aus quasi einem einzigen Bild eine beliebige Anzahl an Trainingsbildern erzeugt. Und damit ist auch klar, warum ein so trainiertes neuronales Netz jetzt eben nicht mehr den Zustand der Kellerlampe lernt, sondern unabhängig von der Bildhelligkeit die darauf abgebildete Ziffer 8.

Neben der Bildhelligkeit gibt es noch eine ganze Menge anderer Parameter, die über diese Funktion variiert werden können (siehe Tabelle *Variationen für Bilder*).

Der Einfluss dieser Parameter ist im letzten Python-Skript *4\_Augumentation\_All.ipynb* anhand zunächst der Variation der einzelnen Parameter und dann auch anhand einer Kombination aller zu sehen. In diesem Beispiel kann man auch verschiedene Parameter durchtesten und auch eigene Bilder prüfen. Das letzte Beispiel mit der Variation aller Parameter zeigt auch, dass man sich genau überlegen muss, wie weit die Parameter gestreut werden sollen 9. Nicht bei allen Bildern kann man noch erkennen, dass es sich ursprünglich um die Ziffer 5 gehandelt hat. Daraus kann man ableiten, welche Variationen noch sinnvoll sind. Das ist eine gute Übung zur Vorbereitung für den nächsten Teil.

### Ausblick

In diesem Teil wurde die Trainingsumgebung Anaconda eingeführt und auf die Vorbereitung der Trainingsdaten eingegangen. Ein



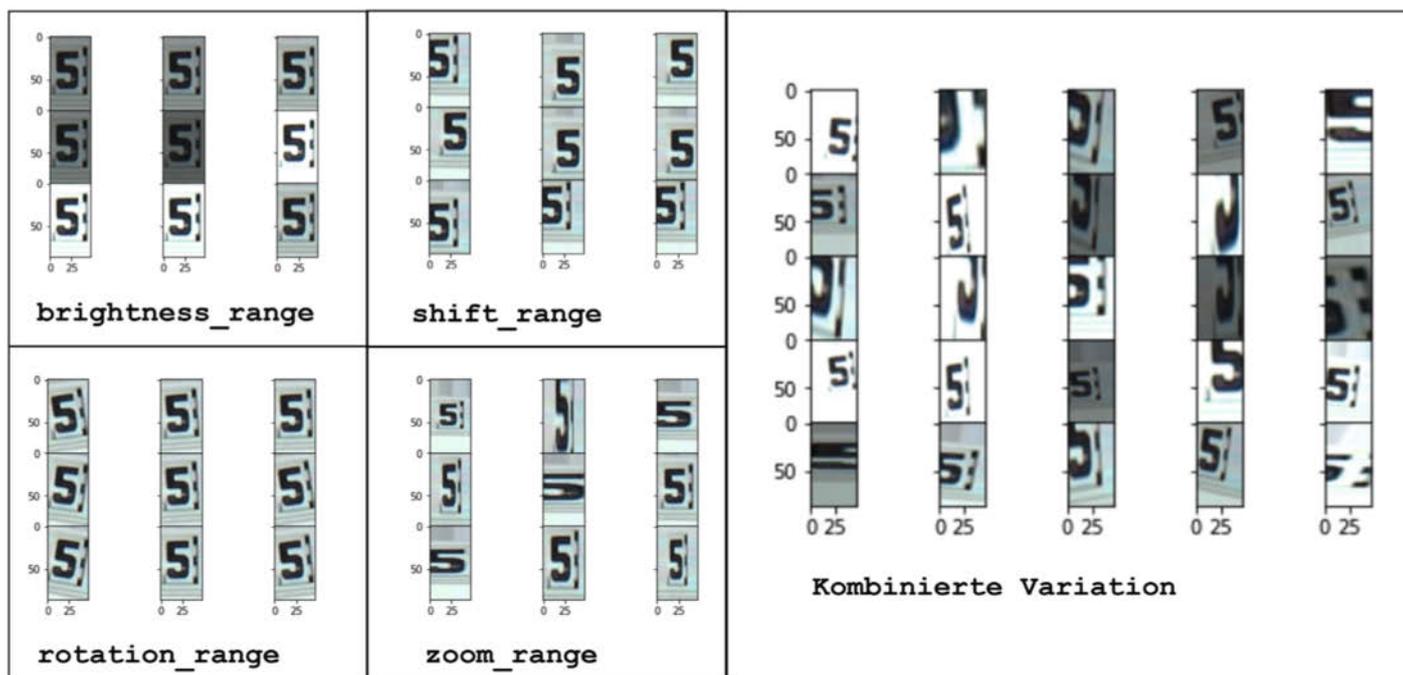
8 Durch Data Augmentation werden aus einem Ursprungsbild viele weitere Bilder mit unterschiedlicher Helligkeiten generiert.

### Variationen für Bilder

Art	Parameter
Helligkeit	<i>brightness_range</i>
Verschiebung	<i>width_shift_range</i> <i>height_shift_range</i>
Verdrehung	<i>rotation_range</i>
Verzerrung	<i>zoom_range</i>

wichtiger Aspekt ist die künstliche Variation der Trainingsdaten, um später ein möglichst robustes neuronales Netz zu erhalten.

Viel Spaß mit den ersten Experimenten zu den Trainingsdaten und eventuell dem Aufnehmen und Erzeugen von eigenen Beispielen. Vielleicht kann das neuronale Netz im nächsten Teil dann schon mit den eigenen Problemen trainiert und damit geübt werden! —dab



9 Weitere durch Data Augmentation generierte Bilder der Ziffer 5

# Dialog mit der Maschine

Mit der Chatbot-Technik AIML kann man mit dem eigenen Projekt über Text-Nachrichten in Dialog treten oder sogar per gesprochener Sprache kommunizieren.

von Daniel Springwald


Chat with Nikola Tesla

● Nikola Tesla 20:14

Guten Tag.

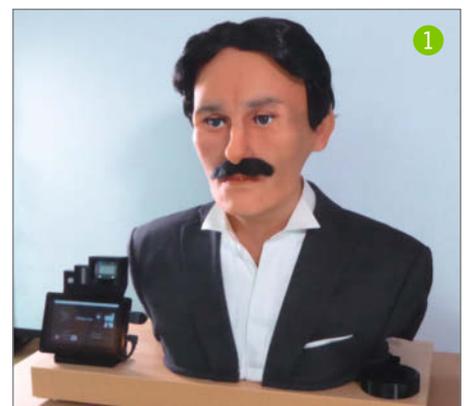
20:14 You ●

Kennen Sie Edison?

● Nikola Tesla 20:14

Ja, der gute Thomas Alva Edison. Wenn es nach ihm gegangen wäre, würde heute Gleichstrom statt Wechselstrom verwendet. Wegen des enormen Energieverlusts bei längeren Leitungen würden dann in jeder Stadt im Abstand von 1 bis 2 Kilometern Kraftwerke stehen.

Send



In der Science Fiction ist es selbstverständlich, mit Maschinen, Robotern oder dem Bordcomputer aus Star Trek per natürlicher Sprache zu kommunizieren. Dank Sprachassistenten wie *Alexa*, *Siri* oder *Google Assistant* ist diese Art der Mensch-Maschine-Interaktion Realität geworden und hat im privaten Alltag vieler Menschen einen festen Platz eingenommen. Für eigene Projekte lassen sich diese Systeme mittels selbsterstellter Apps oder Skills verwenden. Für den Betrieb wird dazu aber eine permanente Internet-Verbindung benötigt und man wird auch die fest eingebauten, vorgegebenen Phrasen wie „Alexa...“ nur schwer oder gar nicht los.

Will man beispielsweise einem eigenentwickelten Gerät oder einer Figur das Sprechen beibringen, möchte man die Dialoge, die Stimme und den Charakter wahrscheinlich gerne selbst prägen. So ging es zumindest mir beim Bau der interaktiven Büste von Nikola Tesla ① (siehe Make 4/20, S.70) und dem Desktop-Assistenten *HoloGaito* ② (Make 3/18, S.24). Bei beiden Projekten ist die natürlichsprachliche Interaktion mithilfe von sogenannten Chatbots auf Grundlage des offenen Standards AIML umgesetzt.

AIML steht für *Artificial Intelligence Markup Language* und wurde 1995 von Dr. Richard Wallace für seinen Chatbot ALICE entwickelt. Obwohl dies bereits einige Jahre zurückliegt, wird AIML kontinuierlich weiterentwickelt: Die letzte Überarbeitung der Spezifikation auf Version 2.1 stammt aus dem Jahr 2018.

### AIML-Interpreter

Um AIML einzusetzen, benötigt man einen Interpreter. Dieses Programm lädt das Wissen des Chatbots, nimmt Eingaben in Textform entgegen und liefert die passende Antwort als Textausgabe. Über diese Text-Ein- und Ausgaben könnte man beispielsweise per Facebook-Messenger oder Telegram-Chat einem selbstgebautes Roboterfahrzeug Nachrichten wie „Fahre 1 Meter vorwärts“ oder „Wie ist dein Status?“ senden.

Rund um den offenen Standard AIML hat sich im Lauf der Zeit eine lebhaft Community gebildet, die für nahezu alle Programmiersprachen und Betriebssysteme freie AIML-Interpreter, Dialog-Beispiele und andere Werkzeuge entwickelt hat. Die Interpreter sind in der Regel offline-fähig, d. h. man kann damit autonome Geräte ohne WLAN-Möglichkeiten um ein Dialogsystem erweitern.

Die Instanziierung eines AIML-Interpreters ist mit wenigen Zeilen Programmcode erledigt – wie das Beispiel des Interpreters von Andrio Celos (Link siehe Kurzinfor) in C# zeigt:

```
using Aimpl;
var bot = new Bot(aimlFilePath);
bot.LoadConfig();
bot.LoadAIML();
```

### Kurzinfor

- » **Stichwörter, Fragen und Antworten in natürlicher Sprache einfach festlegen, erweitern und pflegen**
- » **Externe Daten einbinden und Events auslösen**
- » **Anbindung von Erkennung und Erzeugung gesprochener Sprache möglich**

### Checkliste



**Zeitaufwand:**  
je nach gewünschtem Repertoire des Dialogsystems



**Kosten:**  
keine für die Software

### Mehr zum Thema

- » Daniel Springwald, Animatronischer Tesla, Make 4/20, S. 70
- » Was uns inspiriert: HoloGaito, Make 3/18, S. 24
- » Heinz Behling, Sprachausgabe für Raspberry-, Arduino- und ESP-Projekte, Make 2/21, S. 8

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

Welcher Interpreter am besten zum eigenen Projekt passt, richtet sich nach der verwendeten Plattform bzw. welche der dafür verfügbaren Programmiersprachen einem sympathisch ist. Eine Übersicht an verfügbaren AIML-Interpretern findet sich ebenfalls unter der URL in der Kurzinfor.

### Einsatz in Hardware-Projekten

Will man einen AIML-Chatbot zu Marketing- oder Supportzwecken entwickeln, benötigt man für die ersten Schritte neben einem Interpreter keine weiteren Zutaten. Der Interpreter

muss lediglich in den entsprechenden Kommunikationskanal eingebunden werden. AIML-Hosting-Provider wie *Pandorabots* (siehe Link) bieten dies direkt out-of-the-box an.

Maker-Projekte sind hingegen meist physikalischer Natur, wodurch der Interpreter dort nur einen Teil der benötigten Software darstellt und entsprechend ergänzt werden muss. Das gilt für die Ein- und Ausgabe von Sprache, die Steuerung von Aktoren, Auslesen von Sensoren oder für das Abfragen externer Informationsquellen lassen sich zum Beispiel über Webservices anzapfen: Die Frage „Wie ist das Wetter in Bochum?“ könnte über einen Wetter-



service abgerufen werden, welcher die Daten-grundlage für die Antwort „In Bochum ist es 18 Grad und wolkig“ zurückliefert. Ein solcher Abruf muss Interpreter-spezifisch eingerichtet und könnte zum Beispiel mithilfe des `get-Elementes` umgesetzt werden (dazu später mehr).

Um einem kleinen Roboter per Sprache oder Telegram-Chat Kommandos zu geben und dabei die eine oder andere witzige Antwort einzubauen, ist AIML ebenfalls gut geeignet. Wer eine (historische) Persönlichkeit für eine Museumsinstallation per Chat oder Sprachdialog zum Leben erwecken möchte, kann mit AIML ebenfalls schnell erste Erfolge erzielen. So kann man zum Beispiel mit meiner animatronischen Büste von Nikola Tesla über dessen Lebenswerk und die eine oder andere Anekdote sprechen.

Wer stattdessen ein möglichst universelles Antwortgerät mit Live-Informationen ähnlich Alexa oder Google Assistent erstellen möchte, müsste den Interpreter um derartig viele Abrufe unterschiedlichster Datenquellen erweitern, so dass AIML wahrscheinlich nicht die Technik der Wahl wäre. Zudem müssten viele Eingaben dabei noch in passende Datentypen umgewandelt werden, was von AIML nicht aktiv unterstützt wird. Dessen Vorzüge liegen umgekehrt in der Offline-Fähigkeit, der Verwendung bei eher geschlossenen Inhaltsdomänen oder wenn man dem Dialog eine persönliche Note geben möchte.

### Gesprochene Sprache

Für Projekte wie eine animatronische Büste ist es attraktiv, mit dieser per gesprochener Sprache statt über Texteingaben kommunizieren zu können. Für solche Fälle werden, neben dem AIML-Interpreter, Routinen zur Spracherkennung und Sprachausgabe benötigt. Möglichkeiten zur Sprachausgabe sind auf nahezu allen Plattformen und Betriebssystemen verfügbar. Bei den Standard-Stimmen sollte man allerdings nicht unbedingt erwarten, dass sie besonders natürlich klingen. Wer sein Projekt auf Basis eines Raspberry Pi aufbaut, findet zum Beispiel in der Make-Ausgabe 2/21 passende Tipps dazu. Investiert man etwas

mehr Aufwand in die Recherche und Einrichtung der Sprachausgabe, wird man mit einer Stimme belohnt, die kaum noch von einem echten Menschen zu unterscheiden ist.

Routinen zur Erkennung von Spracheingabe sind dagegen rarer. Systeme, die eine feste Liste zuvor definierter Befehle erkennen können, lassen sich noch verhältnismäßig einfach finden. Für die Eingabe in den AIML-Interpreter wird jedoch die Erkennung von freiem Text benötigt – quasi eine Diktierfunktion. Wenn die Verwendung von Webservices für das eigene Projekt infrage kommt, wird man bei den großen Anbietern wie Microsoft, Google und Amazon schnell fündig. Eine lokale Offline-Erkennung hingegen kann je nach Betriebssystem Aufwand bedeuten – insbesondere, wenn deutsche Sprache erkannt werden soll.

### Syntax und Elemente

Ein zunehmender Teil der heute verfügbaren Dialogsysteme wird anhand von FAQ-Listen, Gesprächs-Beispielen oder anderen Texten automatisch trainiert; der Chatbot lernt sein Wissen zum Teil selbständig. Bei einem AIML-Chatbot wird die Logik hingegen von Hand implementiert. Das erfordert mehr Hingabe, wird aber dafür mit echter Kontrolle über die gegebenen Antworten und das Verhalten der selbst geschaffenen, künstlichen Persönlichkeit belohnt.

Das entsprechende Wissen wird einem Chatbot mithilfe von AIML-Dateien beigebracht. Technisch handelt es sich dabei um Text-Dateien im XML-Format, die je nach verwendetem Interpreter die Dateiendung `.aiml`, `.txt` oder `.xml` haben können. AIML ist keine Programmiersprache, man muss für das Formulieren von Dialogen folglich nicht das Programmieren erlernen. Dennoch können mit den Möglichkeiten von AIML komplexe Diskussionen umgesetzt werden.

Innerhalb einer AIML-Datei heißt eine einzelne Interaktion `category` und wird mit einem entsprechenden XML-Element dargestellt. Eine solche `category` enthält ein Muster zur Erkennung der Benutzereingabe (`pattern-`

Element) und eine Vorlage für die darauf folgende Reaktion (`template-Element`).

Für eine einfache `category`, welche auf die Eingabe „Hallo“ mit der Antwort „Herzlich willkommen!“ reagieren soll, könnte der Quellcode der AIML-Datei wie folgt aussehen:

```
<aiml version="2.0.1"
encoding="UTF-8">
  <category>
    <pattern>Hallo</pattern>
    <template>Herzlich
      willkommen!
    </template>
  </category>
</aiml>
```

Weitere `category`-Elemente werden einfach unten angefügt. Angenommen, der Chatbot soll auf die Eingabe „Guten Tag“ ebenfalls mit „Herzlich willkommen!“ antworten, sähe die Datei so aus:

```
<aiml version="2.0.1"
encoding="UTF-8">
  <category>
    <pattern>Hallo</pattern>
    <template>Herzlich
      willkommen!
    </template>
  </category>
  <category>
    <pattern>Guten Tag</pattern>
    <template>Herzlich
      willkommen!
    </template>
  </category>
</aiml>
```

### Normierung und Vereinfachung

Im vorherigen Beispiel fällt auf, dass die Antwort „Herzlich willkommen!“ zweimal in identischer Form hinterlegt ist. Auf den ersten Blick scheint dies eine übersichtliche Struktur zu sein, denn man kann an jeder Stelle die jeweilige Aktion und Reaktion 1:1 nachvollziehen.

Es ist aber anzunehmen, dass im Laufe der Zeit weitere, gleichbedeutende Benutzereingaben wie „Huhu“ oder „Guten Abend“ hinzukommen werden. Würden diese Eingaben alle nach dem bisherigen Prinzip eingetragen, hätte dies zahlreiche `category`-Elemente mit identischem `template` für die Ausgabe „Herzlich willkommen!“ zur Folge. Soll die Antwort später verändert werden, muss man alle passenden `category`-Elemente finden und diese einzeln anpassen. Das kann dann unübersichtlich werden und ist sowohl aufwändig als auch fehleranfällig.

Glücklicherweise gibt es für diese Fälle das AIML-Element `srai`. Zur wörtlichen Bedeutung dieses Akronymes gibt es keine offizielle

**3 srai in der Praxis**

```
<category>
  <pattern>Hallo</pattern>
  <template>Herzlich willkommen!</template>
</category>
<category>
  <pattern>Guten Tag</pattern>
  <template><srai>Hallo</srai></template>
</category>
<category>
  <pattern>Huhu</pattern>
  <template><srai>Hallo</srai></template>
</category>
```

Definition – stattdessen kursieren diverse Erklärungen. „SR“ könnte demnach *stimulus response, symbolic reduction, syntactic rewrite* oder *synonym resolution* bedeuten. Bei „AI“ ist man sich aber einig, dass es für *artificial intelligence* steht.

srai wird innerhalb eines `template`-Elementes verwendet. Der Inhalt des `srai`-Elementes wird dabei nicht als Antwort ausgegeben, sondern als neue Eingabe in den AIML-Interpreter eingesetzt. Es wird so getan, als hätte der Benutzer den Inhalt des `srai`-Elementes eingegeben. Damit können unterschiedliche Eingaben auf eine einzige `category` zurückgeführt werden.

Verwendet man im vorherigen Beispiel das `srai`-Element für die drei Eingaben, sieht das Ergebnis trotz der zusätzlichen `category` aufgeräumter aus **3**.

Nur die erste `category` für die Eingabe „Hallo“ liefert hier eine echte Antwort. Die beiden anderen Eingaben werden mit `srai` auf die Eingabe „Hallo“ umgeleitet. Bei weiteren Eingaben dieser Art muss man sich somit nur noch merken, dass „Hallo“ die normierte Form für alle Arten der Begrüßung sein soll.

### Star

In AIML kann man somit `pattern` unterschiedlich formulierter Eingaben zusammenfassen. Wenn sich hingegen die Eingaben inhaltlich ähnlich sind, kann das `star`-Element hilfreich sein. `star` ist eine der *Wildcards* in AIML, die als Platzhalter für beliebige Eingaben verwendet werden können.

Bei einem Smart-Speaker möchte man vielleicht Befehle wie „Schalte Licht ein“, „Schalte Fernseher ein“ und die Namen weiterer Smart-Home-Geräte eintragen. Ein universelles `pattern` für diesen Zweck könnte man so formulieren, wie im Codebeispiel **4** gezeigt.

Der `*` kann in der Eingabe des Benutzers für beliebig viele Worte stehen. Im `template` werden diese erkannten Worte dann in das Element `<star />` eingesetzt. Die Antwort auf „Schalte Licht ein“ lautet dann: „Ich habe *Licht* für Dich eingeschaltet.“

Neben `star` gibt es noch andere *Wildcards* in AIML, die leicht unterschiedliches Verhalten aufweisen. Gemeinsam haben sie aber alle, dass sie ausschließlich für ganze Wörter gelten und somit nicht auf einzelne Zeichen innerhalb von Worten angewendet werden können. Komplexe Muster, wie man sie zum Beispiel bei *Regulären Ausdrücken* verwendet, sind in AIML an dieser Stelle nicht vorgesehen. Einen Überblick der AIML *Wildcards* findet sich in den Links zum Artikel in der Kurzinfor.

`star` wird ebenfalls für die sogenannte *Ultimate Default Category* (UDC) verwendet. Sie liefert eine Antwort, wenn keine passende `category` gefunden werden kann. Damit stellt sie eine Art Notfall-Antwort dar, denn

## 4 Pattern mit \*

```
<category>
  <pattern>Schalte * ein</pattern>
  <template>Ich habe '<star />' für Dich eingeschaltet.</template>
</category>
```

mit ihr kann auf unerwartete Eingaben reagiert werden.

Eine Beispiel-Implementierung einer UDC könnte so aussehen:

```
<category>
  <pattern>*</pattern>
  <template>Tut mir Leid, das habe ich nicht verstanden.
</template>
</category>
```

### Set

Es ist wenig hilfreich, wenn ein Gerät nur *sagt*, es habe das Licht eingeschaltet. Sinnvoll wäre es, wenn die entsprechende Aktion zudem noch ausgeführt wird. Dafür kann man das Element `set` verwenden, mit welchem man Zustände an den Interpreter senden kann.

Im Quellcode könnte dies zum Beispiel so aussehen:

```
<category>
  <pattern>Schalte * ein</pattern>
  <template>
    Ich habe '<star />' für Dich eingeschaltet.
  <think>
    <set name="TurnOn">
      <star />
    </set>
  </think>
</template>
</category>
```

Das erste neue Element in diesem Beispiel ist `think`. Darin enthaltene Elemente oder Texte werden nicht als Antwort ausgegeben – der Interpreter *denkt* sich dies nur. Im `think`-Element ist ein weiteres, neues Element `set` enthalten. Dieses setzt den Zustand mit dem Namen *TurnOn* auf den vom `*`-Wildcard gefangenen Inhalt – in unserem Beispiel *Licht*. Wie dann im AIML-Interpreter Änderungen am Zustand *TurnOn* zu einer entsprechenden Schaltaktion beim angegebenen Gerät führen können, wird gleich noch gezeigt.

Da der Inhalt der `star`-Wildcards in diesem Beispiel auch noch ausgegeben werden soll, könnte man auf das `think`-Element verzichten und alles elegant zusammenfassen:

```
<category>
  <pattern>Schalte * ein</pattern>
  <template>
```

Ich habe '`<set name="TurnOn"><star /></set>`' für Dich eingeschaltet.

```
</template>
</category>
```

### GET / SET

Wie oben erwähnt kann es je nach Anwendungszweck notwendig sein, während des Dialogs (externe) Informationen zu lesen oder zu schreiben. In den bisherigen Beispielen wurde dafür das Element `set` verwendet, das passende Gegenstück dazu ist `get`. Diese beiden Elemente sind dazu vorgesehen, Details über den aktuellen Benutzer bzw. Erkenntnisse aus dem Gespräch zu speichern.

Im folgenden AIML-Beispiel schreibt der Interpreter für die Eingabe „Hallo, ich heiße Peter“ den Wert *Peter* in den Speicherplatz *Name*. Anschließend kann der gespeicherte Namen als Antwort auf die Eingabe „Wie heiße ich?“ ausgegeben werden.

```
<category>
  <pattern>Ich heiße *</pattern>
  <template>
    <think>
      <set name="name">
        <formal><star />
      </formal>
    </set>
  </think>
  Schön Sie kennenzulernen,
  <get name="name" />.
</template>
</category>
<category>
  <pattern>Wie heiße ich</pattern>
  <template>
    Soweit ich mich erinnere, heißen Sie
    <get name="name" />.
  </template>
</category>
```

Das Element `formal` in diesem Quellcode schreibt alle Anfangsbuchstaben groß. Gibt man „ich heiße max mustermann“ ein, wird als Name *Max Mustermann* gespeichert.

Gespeicherte Informationen sollen nach dem Gespräch nicht verloren gehen, sondern beim nächsten Kontakt mit dem Chatbot weiterhin abgerufen werden können. Wo diese Daten dabei gespeichert werden, kann

## 5 User anlegen

```
using Aiml;

// Bot laden
var bot = new Bot(aimlFilePath);
bot.LoadConfig();
bot.LoadAIML();

// User mit ID "user" erzeugen
var userId = "user";
var user = new User(userId, bot);

// Eingabe "Hallo" verarbeiten und Ausgabe auf der Konsole ausgeben
var response = bot.Chat(new Request("Hallo", user, bot), trace: false);
Console.WriteLine("Bot output: " + response.Output);
```

## 6 Edison-Antwort

```
<category>
  <pattern>^ Edison ^</pattern>
  <template>
    <think>
      <set name="action">speak</set>
      <set name="mood">amused</set>
    </think>
    Ja, der gute Thomas Alva Edison. Wenn es nach ihm gegangen wäre,
    würde heute Gleichstrom statt Wechselstrom verwendet. Wegen des enormen
    Energieverlusts bei längeren Leitungen würden dann in jeder Stadt im
    Abstand von 1 bis 2 Kilometern Kraftwerke stehen.
  </template>
</category>
```

je nach Anwendungsfall unterschiedlich sein: Auf einem Webserver könnte dies in einer SQL-Datenbank anhand einer Session-ID des Benutzers sein. Bei einem lokal verwendeten Raspberry-Pi könnte eine einfache JSON-Datei auf der SD-Karte genügen.

Die Aufgabe der Speicherung dieser Daten wird dazu bewusst vom AIML-Interpreter ent-

koppelt und ermöglicht es hier, mit der Welt außerhalb des Interpreters Daten auszutauschen. Dabei kommt man natürlich um Programmierung in der Sprache des jeweiligen Frameworks nicht herum.

Durch diese Entkopplung muss man dazu am Interpreter selbst auch keinen Quellcode anpassen. Das ist insbesondere dann praktisch, wenn man später ein Update der Interpreter-Software durchführen möchte. Man verliert dann keine hinzugefügten Änderungen an der Software. Beim erwähnten Interpreter von Andrio Celos wird dafür ein User-Objekt erzeugt und dem Chatbot zu jeder Anfrage als Kontext mitgegeben 5.

AIML ermöglicht es hier, mit der Welt außerhalb des Interpreters Daten auszutauschen. Dies kann man – neben der gezeigten Speicherung von Benutzer-Informationen – auch dynamisch verwenden. Im vorhin gezeigten Beispiel „Schalte Licht ein“ wurde Mittels get-Element der Wert *Licht* in den Speicher mit dem Namen *TurnOn* geschrieben. Ein angepasstes User-Objekt könnte den Namen *TurnOn* erkennen und daraus einen Smarthome-Befehl erzeugen, der das Gerät *Licht* einschaltet. Bei der animatronischen Büste von Nikola Tesla werden auf diesem Weg die Gestik und Bewegungen übergeben, die dann mit Motoren auf das Gesicht übertragen werden. Bei der Frage nach Thomas Edison sieht das Setzen der

Aktivität *speak* und der Gestik *amused* in AIML aus wie in 6.

Die ^-Zeichen im pattern-Element sind (wie der Stern) ebenfalls Wildcards. An Stelle des Sterns können *ein oder mehrere Worte* eingegeben werden, beim ^-Zeichen können es *kein Wort oder eine beliebige Anzahl Worte* sein.

Die hier gezeigten Beispiele enthalten die wichtigsten Elemente für den Einstieg. AIML bietet aber noch viel mehr Möglichkeiten – ein umfassender Überblick über die gesamte AIML-Syntax würde den Umfang eines Artikels übersteigen. Hierfür können die Website der AIML-Foundation (siehe Link) oder der Quellcode anderer AIML-Autoren gute Anlaufstellen sein.

## Fazit und Ausblick

Will man ein selbst gebautes Gerät mit den üblichen Standardfunktionen eines Sprachassistenten ausstatten, ist es am einfachsten, ein Projekt wie *MyCroft* oder eine Implementierung von *Alexa* und *Co.* zu verwenden. Die dazu notwendigen Funktionen sind dort enthalten und mit geringem Aufwand verwendbar.

Wünscht man sich hingegen volle Kontrolle über die gesprochenen Dialoge und möchte dem Gespräch einen eigenen Charakter verleihen, erhält man mit AIML ein leistungsfähiges Werkzeug mit dennoch geringer Einstiegshürde. Anwendungsfälle für eher individuelle Dialogsysteme trifft man in jüngster Vergangenheit im kommerziellen Umfeld häufig an. So hat die Deutsche Bahn mit dem Roboterkopf *SEMMI* (siehe Link) ein KI-basiertes Sprachdialogsystem präsentiert, das testweise im Kundenservice von zwei Reisezentren eingesetzt wird.

Ein ähnliches Produkt kann man in Japan in der Praxis finden: Dort sind in Tokyo an Bahnhöfen *AI Sakura-san*-Bildschirm-Terminals aufgestellt, in denen eine Anime-Figur gezeigt wird 7. An der Seite des Gerätes ist ein Telefonhörer angebracht, über den man mit diesem virtuellen Ratgeber mit einer Art Videoanruf über Fahrplan- und Fahrkarten-Fragen sprechen kann. Die Idee, dazu einen Telefonhörer zu verwenden, ist eine elegante Lösung für das Problem der lauten Hallenakustik im Bahnhof. Vermutlich der aktuellen Pandemie geschuldet scheint der Hersteller im Augenblick allerdings nur kontaktlose Geräte anzubieten.

In anderen Einsatzgebieten wie der Entwicklung von Videospielen bietet AIML ebenfalls spannende Möglichkeiten. So könnte es bei der Interaktion mit NPCs freiere Dialoge ermöglichen, als sie in Videospielen heute angeboten werden.

Ich persönlich habe jedenfalls große Lust darauf, als eines meiner nächsten Projekte etwas wie die japanischen Sprachterminals zu bauen – dabei wird AIML sicherlich meine erste Wahl sein. —pek



# 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!

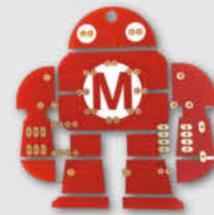
### ArduiTouch-Set

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

36,90 €

~~69,90 €~~



### 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.

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

99,90 € >

~~117,60 €~~



### 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 designt 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](http://shop.heise.de) oder per E-Mail: [service@shop.heise.de](mailto: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 >



# Erstaunlich stabil: Pappmaché

Pappe zu recyceln und daraus haltbare Gegenstände herzustellen, ist einfach und bringt Spaß. Wir zeigen, wie man 3D-Druck mit Pappmaché kombiniert, um nützliche Helfer und kleine Geschenke zu machen.

von Carsten Wartmann



**P**appe ist ein toller Werkstoff und lässt sich gut recyceln, sei es als Verpackung, für Basteleien oder in Kistenform als Stauraum für selten gebrauchte Sachen. Wer viel bestellt, hat aber irgendwann genug Kisten. Aus übrig gebliebener Pappe und Papier kann man dann Pappmaché machen. Denkt man an seine Kindheit zurück, dann entstanden aus Pappmaché eher lustige Figuren oder Masken mit recht rauer Oberfläche und man hat nur bedingt reproduzierbare Gegenstände erhalten. Agustín Arroyo (@flowalistik auf Twitter) hat zusammen mit BuildBee das Projekt *Pulp It!* auf den Weg gebracht, um das Recycling zu thematisieren und Objekte mit einer eigenen Ästhetik, ohne den Look von Eierkartons zu generieren. Einige der Formen sind frei herunterladbar (siehe Links in Kurzinformato), sehr professionell und haltbar entworfen. Das ist prima, da Formenbau immer ein recht kompliziertes Thema ist.

Wir haben es zu Hause ausprobiert und geben unsere Erfahrungen hier weiter. Ein paar neue Ideen, wie man den Prozess verbessern kann oder eigene Formen erstellt, stellen wir ebenfalls vor. Eines vorweg: Mit einem Eierkarton haben die entstehenden Objekte nicht viel zu tun, das gepresste Pappmaché ist sehr viel härter und fast wie Holz zu verarbeiten. In den Links finden Sie ein Video dazu.

## 3D-Druck

Wir haben bei uns zuerst einmal die herunterladbare Form für die Schale (Titelbild dieses Artikels und **1**) ausprobiert. Der 3D-Druck ist durchaus zeitintensiv (13 Stunden auf einem Prusa i3 Mk2) und es geht auch einiges an Material (ca. 240g) durch die Düse. Ich habe mein Lieblings-Druckmaterial PETG verwendet, dies ist aber etwas zu flexibel, sodass sich die Form bei hohen Pressdrücken leicht ausbeult und man an den Kanten etwas mehr überstehendes Material bekommt. PLA ist stabiler und daher vermutlich besser geeignet. Eine PLA-Form schmolz mir aber beim Trocknen auf einem zu heißen Druckbett. Am Ende des Artikels zeigen wir noch einfachere Formen für Dekorationsgegenstände, die wesentlich weniger Material und Druckzeit benötigen. Auch die Untersetzer-Formen von *Pulp It!* sind gut geeignet zum Einstieg und praktisch im alltäglichen Gebrauch. Durch Einlagen, die man leicht selbst gestalten kann, sind sie auch einfach personalisierbar und ideal als Last-Minute-Geschenk.

Der Druck ist, von der langen Druckzeit mal abgesehen, nicht besonders fordernd und kann bei Bedarf in zwei oder drei kleinere Portionen aufgeteilt werden, wenn man die großen Teile einzeln druckt. Ich habe in 0,2mm-Schichten gedruckt, mit einer 0,4mm-Düse. Wegen der hohen Presskräfte wurden vier Schichten für die oberen und unteren Flächen verwendet sowie ein hexagonales Infill mit 20% Füllichte.

## Kurzinfo

- » Pappmaché-Rezepte und -Herstellung
- » Arbeitsweise für haltbare Objekte
- » Tipps und Tricks für eigene Formen

### Checkliste



#### Zeitaufwand:

1 Stunde (ohne Druck und Trocknung)



#### Kosten:

5 Euro für Druckmaterial

### Werkzeug

- » 3D-Drucker
- » Mixer Hand- oder Stabmixer
- » Stoff- oder Netzbeutel
- » Schraubzwingen oder großer Schraubstock
- » Cutter und Schere
- » Schleifwerkzeuge

### Material

- » 3D-Druckfilament PLA, PETG
- » Kartonage oder Papier
- » Kleber Tapetenkleister, PVA-Leim, Mehl oder Reisstärke
- » Zuschlagstoffe wie Wollfäden, Glitter, farbige Papierschnipsel

### Mehr zum Thema

- » Carsten Wartmann, Betonfiguren aus der 3D-Druckform, Make 4/21, S. 88
- » Stella Maria Risch, Kunststoffabfälle selbst recyceln, Make 4/21, S. 94
- » Grace Dobush, Vom Abfall zum Einfall, Make 1/20, S. 64

Alles zum Artikel  
im Web unter  
[make-magazin.de/x9w9](https://make-magazin.de/x9w9)

## Der Pulp!

*Pulp* ist das Fachwort für die feuchte Pappmaché-Rohmasse. Das Grundrezept ist eventuell schon aus der eigenen Kindheit bekannt. Im einfachsten Fall zerkleinert man Papier oder Pappe fein und mischt es mit Wasser an. Dann wird ein Großteil des Wassers entfernt, man erstellt seine Objekte und trocknet sie. Möchte man besonders haltbare Objekte erhalten, so wird noch etwas Leim oder sonstiger Kleber hinzugefügt. Wir haben mit Tapetenkleister und Mehl gearbeitet und in beiden Fällen sehr feste Gegenstände erhalten.

Verwenden Sie die typischen Versandhandels-Kartonagen, so müssen Klebeband, vor allem solches aus Kunststoff und mit Verstär-

kungsfasern, sorgfältig entfernt werden. Auch Metallheftklammern können die Messer des Mixers beschädigen. Generell sollte die verwendete Pappe oder das Papier möglichst langfaserig sein, um haltbares Pappmaché herzustellen.

## Zerkleinern

Zuerst zerlegt man die Pappe mit dem Cutter auf einer Schneidmatte in Streifen. Danach schneidet man mit einer Schere **2** kleine Stücke von diesen Streifen ab, am besten gleich in das Mixgefäß hinein. Solange die Pappe noch trocken ist, fällt es schwer zu schätzen, wie viel man bereits hat, das Volumen reduziert sich um etwa 80%, sobald die Pappe ein-





weicht. Für die Box sollte man etwa 140g Pappe bereitstellen.

Jetzt wird etwas heißes Wasser **3** über die Pappe gegossen. Mit einem Löffel oder ähnlichem leicht zusammenstampfen, danach ist wieder Platz für den Rest der Pappschnipself.

Das Zerkleinern geht am besten mit einem Standmixer, aber auch ein Handmixer **4** funktioniert. Bei einem Handmixer ist, besonders wenn Kinder mitbasteln, auf die mehr oder weniger offenen Messer und die dadurch hohe Verletzungsgefahr zu achten!

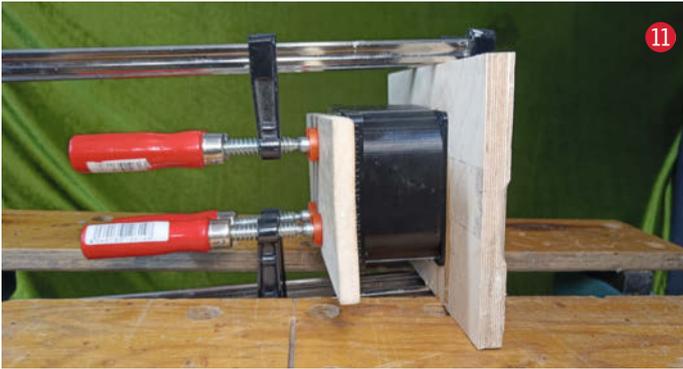
Nun geht es ans Mixen. Je mehr Wasser man einsetzt, umso mehr muss später der Pulp wieder entwässert werden. Allerdings belastet zu trockener Pulp den Mixer. Wenn man einen Stabmixer verwendet, wird das Mixen zu einer wirklich anstrengenden Sache: Sobald die Masse anfängt zäh zu werden, saugt sich der Mixer regelrecht in dem Pulp fest.

Nach den ersten Runden mit dem Mixer kann auch der Kleber hinzugegeben werden. In der Pappe von Kartonagen ist oft schon genug Kleber bzw. Binder enthalten, der die

Gegenstände beim Trocknen wieder verfestigt. Wir haben für die ersten Versuche 2–3 Esslöffel Mehl in die Masse gemixt. Das funktioniert prima, verströmt aber beim Trocknen einen Geruch, der nicht jedem gefällt, aber im trockenem Zustand verfliegen ist. 2–3 Esslöffel Tapetenkleister funktionieren ebenso und der Leim ist geruchslos. Holzleim und anderer PVA-Kleber, traditioneller Reis-Kleber oder Stärke sollen ebenso funktionieren. Der Pulp **5** sollte am Ende kaum noch fließen. Etwas zu viel Wasser schadet nicht und macht das Mixen leichter.

Alles wird jetzt so lange gemixt, bis keine Pappstücke mehr sichtbar sind, eventuell findet man nun auch noch ein paar farbige Teile von Klebeband oder ähnlichem und kann diese noch herausfischen. Wir haben allerdings auch mit Zuschlagstoffen wie Glitter, Stücken farbigen Papiers und Wollresten experimentiert, die dem fertigen Gegenstand eine interessante Oberfläche geben. Bei den Wollresten sollte man nicht zu lange Fäden verwenden, damit sich diese nicht um die Achse des Mixermessers wickeln.





## Entwässern

Ein wichtiger Schritt ist nun das Entwässern des Pulp **6**. Dessen Wassergehalt bestimmt später in der Form, wie solide und glatt das Objekt wird. Zu viel Wasser im Pulp sorgt für ein eher schaumiges (Eierkarton-artiges) Endergebnis, da das Wasser beim Verdunsten Hohlräume hinterlässt. Auch die Oberfläche wird so viel unebener. Zu trockenes Pulp wiederum kann schwer in die Form zu bringen sein, man erhält aber sehr glatte Oberflächen und extrem feste Objekte.

Das Auspressen des Wassers erledigt am besten das Familienmitglied mit den stärksten Händen. Wir haben dazu etwa faustgroße Mengen Pulp in einen Kleider-Netz-Beutel (vermutlich von Knieschonern) gegeben. Wäschesäcke oder selbstgenähte Beutel aus stabilem Stoff sollten auch funktionieren. Jetzt den Beutel oben zuhalten, verdrillen und schon läuft das erste Wasser unten heraus. Mit beiden Händen so lange arbeiten, bis man erschöpft ist oder kein Wasser mehr kommt. Die Konsistenz des Pulp **7** sollte etwa wie die von nassem Ton oder fester Knete sein.

## Das Formen

Fügen Sie nun die Form zusammen. Die Klammer, die die Formwand zusammen hält, muss dabei schon aufgeschoben werden. Bei der Schale und dem Stifthalter ist es besser, wenn Sie die Form umdrehen, sodass die Ausstülpung unten auf dem Tisch steht, so lässt sich die Form besser füllen und es muss beim Pressen weniger Material verdrängt werden. Eine wasserfeste Unterlage oder eine flache Schüssel verhindert eine zu große Sauerei. Flecken sind aber leicht feucht abwischbar, auch aus Kleidung können sie problemlos ausgewaschen werden. Benutzt man farbige Pappe oder Papier, so können die Farbstoffe darin allerdings Hände und Kleidung nachhaltig verfärben!

Wurde viel Kleber verwendet, kann man zur Sicherheit die Form leicht mit Öl oder Vaseline benetzen, damit das trocknende Pappmaché nicht festklebt. Nur eine dünne Schicht auftragen, sonst kann durch das Öl das spätere Anmalen mit Wasserfarben erschwert sein.

Je nach Wassergehalt kann man den Pulp zwischen den Händen in kleine Brösel zerreiben oder man zerrupft ihn mit den Fingern. Die kleinen Stücke werden jetzt in die Form eingebracht **8** und festgedrückt. Währenddessen die Form mit der anderen Hand festhalten, damit man diese nicht versehentlich auseinanderzieht. Dabei rundherum in Schichten arbeiten. Ein Rundholz oder das glatte Ende eines Fasermalers **9** kann benutzt werden, um die Schichten gut in die für Erwachsenenhände zu engen Lücken zu drücken.

Dort wo die Wände in den Boden übergehen, muss sorgfältig gearbeitet werden, um eine glatte und gleichmäßig dicke Schicht zu modellieren. Die Form sollte etwa bis 2–3mm unter den Rand gefüllt werden, damit der Boden noch gut eingesetzt werden kann. Sobald das geschehen ist, mit den Händen und dem ganzen Körpergewicht die Form zusammenpressen. Bei recht trockenem Pulp wird nicht viel Wasser austreten, aber wenn man den Boden **10** nochmals entfernt, erkennt man bereits, ob die Oberfläche gleichmäßig ist. Sind noch Lücken oder Löcher zu sehen, kann man dort noch mit etwas Pulp ausbessern.

## Pressen der Form

Durch das Pressen der Form werden Wasseransammlungen herausgepresst, die sonst



später Hohlräume bilden würden. So erhalten wir ein homogenes und stabiles Pappmaché. Wir haben noch zwei kleine Brettchen verwendet, um die Form gegen die doch recht hohen Pressdrücke der Schraubzwingen zu schützen **11**. Erfolgreich waren wir auch, indem wir die Form zum Pressen in die Backen einer Werkbank oder einen großen Schraubstock eingespannt haben.

Die Form sollte langsam und gleichmäßig gepresst werden. Dabei wird schon das erste Wasser austreten. Am besten eine Schüssel unterstellen, denn insgesamt tritt soviel Wasser in den ersten Stunden aus, dass es ohne eine ziemliche Sauerei wäre. Wenn die Form nicht zu voll gepackt wurde, sollte sie vorsich-

## Überschuss

Das heraus gepresste Wasser ist mit Kleber gesättigt und kann für eine neue Portion Pulp benutzt werden. Sie sollten lieber zu viel Pulp herstellen. Überschüssiges Material kann für die klassische Pappmaché-Modellage von kleinen Figuren benutzt werden – oder Sie können es einfrieren. Das aufgetaute Pulp funktioniert problemlos.





tig bis zum kompletten Schließen gepresst werden. Noch mehr könnte jetzt die Form beschädigen!

### Trocknung und Entformen

Die Zwingen können nach 6–8 Stunden entfernt werden. Die originalen *Pulp It!*-Formen erwiesen sich in unseren Versuchen als sehr wasser- und luftdicht, aber wo kein Wasser austritt und keine Luft hineinkommt, kann das Pulp nicht richtig austrocknen. Daher hilft das Öffnen wenigstens einer Seite der Form sehr.

Je nachdem, wie trocken der Pulp bei der Verarbeitung war, kann jetzt auch schon vorsichtig der Boden der Form entfernt werden. Dazu hat es sich als praktikabel erwiesen, ein Plektrum <sup>12</sup> oder einen Plastikspatel vorsichtig in die Lücke zwischen Form und Boden einzuführen und den Boden (das Formteil ohne Logo) rundherum gleichmäßig aufzuhebeln. Bei uns hörte man deutlich ein Zischen, wenn der Unterdruck in der Form sich abbaut. Ist der Boden offen, kann der Inhalt besser trocken, am besten in einem geheizten Raum. Man kann die Form auch auf eine Heizung oder das heizbare Druckbett eines 3D-Druckers legen. Aber Vorsicht mit den Temperaturen, damit die Form (besonders bei PLA) nicht weich wird.

Nach weiteren 6–8 Stunden kann man bei den einfachen Formen wie der Schale auch schon den konvexen Teil <sup>13</sup> entfernen, bei dem Stifthalter sollten Sie lieber noch länger warten. Auch hier vorsichtig, rund herum mit einem Plektrum o.ä. arbeiten: Wenn sich der Boden nicht bewegt, auf keinen Fall mit Gewalt probieren, lieber noch etwas länger trocknen lassen. Sollte ein Stück des Objekts an der Form kleben bleiben, so kann man dieses vorsichtig wieder in die Fehlstelle einsetzen und festdrücken.

Ist man ungeduldig, kann man bei einfachen Objekten auch schon im feuchten Zustand vorsichtig die Klammer und die Formwand entfernen. Das nicht getrocknete Objekt ist noch sehr weich und empfindlich! Fasst man es zu hart an, geht es kaputt oder man sieht seine Fingerabdrücke als kleine Beulen.

### Nachbearbeitung

Nachdem das Objekt komplett getrocknet ist, ist es leicht und stabil. Es kann auf viele Arten nachbearbeitet werden. Das Pappmaché ist in etwa mit MDF-Material zu vergleichen. Schleifen, Sägen, Schneiden und Bohren sind einfach möglich. Ränder, die durch zwischen die Formteile gepressten Pulp entstanden sind, lassen sich mit einem scharfen Seitenschneider <sup>14</sup> leichter und sauberer als mit einem Messer entfernen. Dann noch einmal mit einem Stück Schleifpapier nacharbeiten.

Anschließend kann man die Objekte bemalen, lackieren, mit Strass-Steinen bekleben, sonstwie verzieren oder man belässt sie im Rohzustand. Vielleicht haben Sie ja auch schon Zuschlagstoffe, wie oben erwähnt, in den Pulp gegeben, dann würde sich vielleicht noch eine Versiegelung mit klarem Lack anbieten.

Die fast weißen Schalen im Aufmacherbild entstanden aus Instant-Pappmaché, das nur mit Wasser angerührt werden muss und

fast schon langweilig glatte Oberflächen produziert. Uns gefällt der raue, immer leicht wechselnde Look der Recyclingpappe am besten, damit ist jedes Stück auch ein Unikat.

### Eigene Formen

Als zugegebenermaßen 3D-Nerd wollte ich natürlich auch das Erstellen eigener Formen ausprobieren. Ich mache alle meine 3D-Objekte in *Blender*, einer Open-Source-3D-Anwendung. Die Vorlagen von *Pulp It!* werden als *Fusion360*-Dateien zur Verfügung gestellt, ein guter Startpunkt für Anwender dieser Software. Angesichts der nahenden Weihnachtszeit und der fehlenden Freizeit sollten aber einfachere und in die Saison passende Formen entstehen. Die Entscheidung fiel auf weihnachtlichen Baumschmuck <sup>15</sup>.

Wenn Sie planen, selbst eine Form zu entwickeln, dann muss diese so konstruiert sein, dass sich das fertige Objekt gut entformen lässt. Dazu muss die Form mindestens zweiteilig sein, dies ermöglicht schon einfache flache Objekte wie Sterne und Ähnliches. Ist das Objekt dicker, wird es trotz abgeschrägter Seiten schwierig, das Objekt zerstörungsfrei aus der Form herauszubekommen. Dann muss man eventuell eine dreiteilige Form konstruieren. Bei sehr hohen Objekten muss dann auch die Wandung, wie in den *Pulp It!*-Formen, teilbar sein.

Für den Stern habe ich erst ein Experiment mit einer zweiteiligen Form gemacht. Dies funktioniert prima, nur muss man durch die flache Form den Pulp sehr sorgfältig in die kleinen Vertiefungen einmassieren, da bei dem Boden kaum ein Überstand zum Pressen existiert. Die dreiteilige Form behebt dieses Problem und macht das Entformen leichter. Die Formen wurden mittels *Booleans* (3D-Objekte voneinander abziehen und zusammenfügen) und weiteren *Modifiern* konstruiert, sicher nicht so elegant wie ein komplett parametrisches Objekt, aber schnell zu konstruieren. Die 3D-Dateien finden Sie auf unserem GitHub (siehe Link in der Kurzinfor).

Wenn man auf gute Entformbarkeit achtet, kann Pappmaché auch sehr kleine Details abbilden: Die Drucklayer von 0,2mm, mit denen meine Formen gedruckt wurden, sieht man zumindest problemlos im Objekt. Der Twitter-User *Smatkala (@mxbasse)* hat eine Form konstruiert, mit der man eine Pappmutter erstellen kann, die mit Silberfarbe angesprüht in jedem Film als leichtgewichtige Requisite durchgehen würde. Auch das Formen-Design ist genial und benutzt das Positiv einer Schraube als Form für das Gewinde und gleichzeitig als Klemmvorrichtung.

Also viel Spaß beim Entwerfen, matschen und basteln! Gerne können Sie uns eine E-Mail mit Ihren Ergebnissen und Erfahrungen schreiben: [mail@make-magazin.de](mailto:mail@make-magazin.de) —caw





# building **IoT**

**Die Konferenz zu IoT und Industrie 4.0**

**10. und 11. Mai 2022 - Karlsruhe**

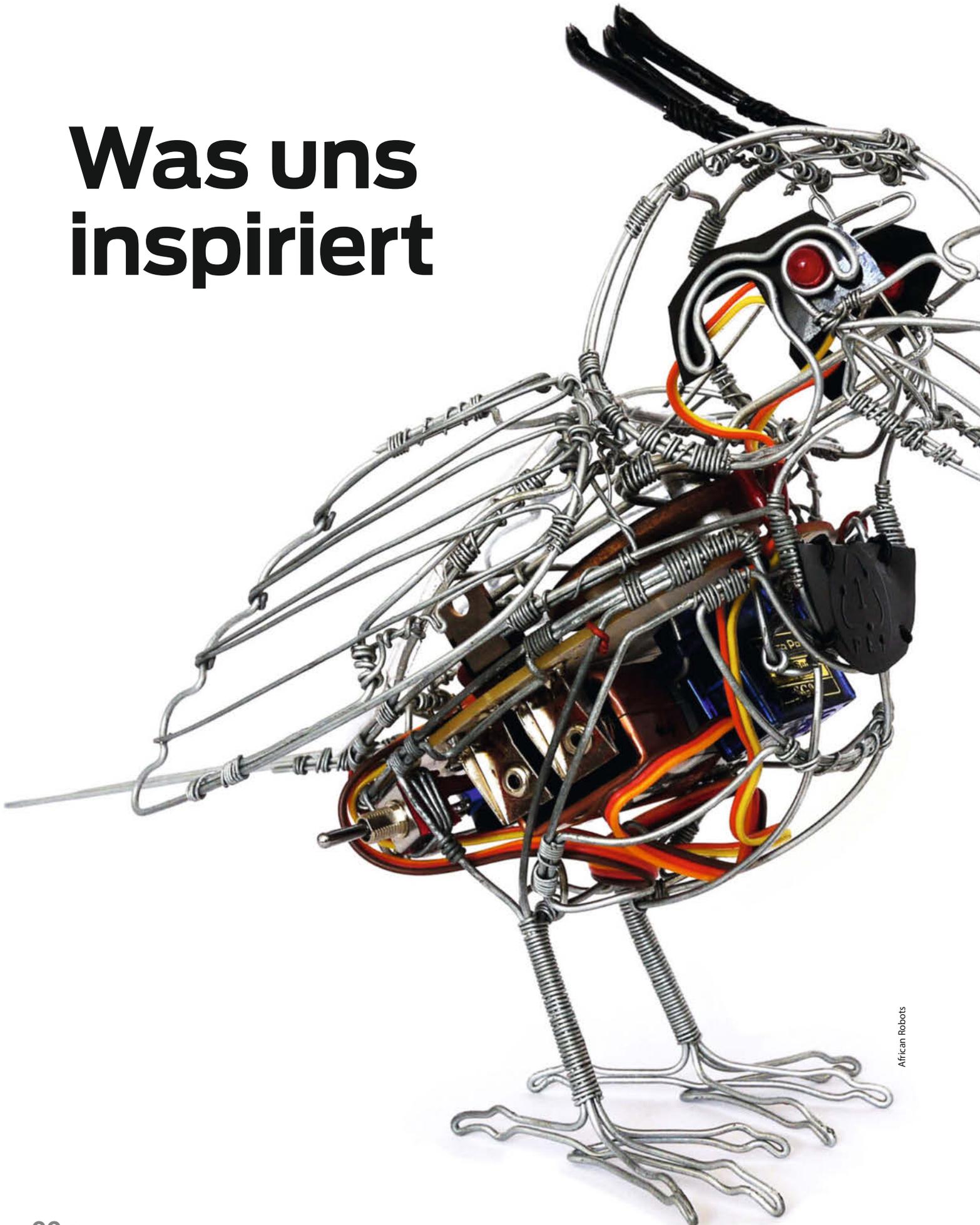
**Mögliche Vortragsthemen:**

- Protokolle und Standards
- Industrial IoT und Automatisierungstechnik
- Security und Safety
- Edge- und Fog-Computing
- Machine Learning und Datenanalyse
- Testing von IoT-Plattformen
- Datenschutz und rechtliche Aspekte

Call for Proposals  
noch bis  
**10.1.2022**  
Jetzt  
einreichen!

**[www.buildingiot.de](http://www.buildingiot.de)**

# Was uns inspiriert



African Robots

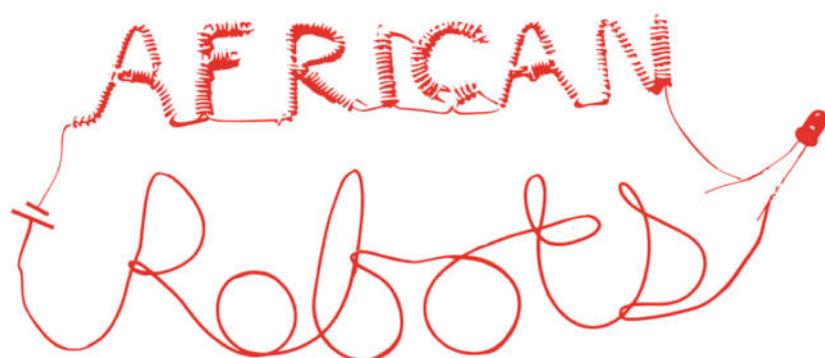
# Street Art trifft Mechatronik

*African Robots* ist das Projekt eines südafrikanischen Künstlerkollektivs rund um Skulpturenkünstler *Ralph Borland*. Sein Ziel: Kunst und Elektronik im urbanen Raum zusammenzuführen. In diesem Kollektiv kommen *Wire Artists*, die traditionelle, fragil wirkende Skulpturen aus Draht herstellen, und Mechatroniker zusammen, die den animatronischen Figuren Leben einhauchen.

Das Projekt startete 2013 mit dem ersten Roboter, dem *Starling 1.0*: Die Inneren des Vogels bestehen aus Nokia-Handyteilen und anderen billigen elektronischen Komponenten. Dank dieser ist der Vogel mit beweglichen Flügeln, leuchtenden Augen und einem auf Licht reagierenden Vogelruf ausgestattet.

*Wire Art* kann in Südafrika auf eine lange Tradition als *Street Art* zurückblicken. Dabei teilen sich die Drahtkünstler die gleichen urbanen Räume wie informelle Handy-Reparaturstände – die Wiederverwendung von ausrangierten Handyteilen macht die Symbiose also perfekt. Mittlerweile ist aus diesem Kollektiv ein weiteres Projekt namens *SPACECRAFT* entstanden, das Raumschiffe aus Draht entwickelt. —*rehu*

► [africanrobots.net/](http://africanrobots.net/)





Mathias Franz

# Stiller als still

Wenn ein Kopfhörer zerbricht, muss er noch lang nicht in die Tonne – im Gegenteil! Bastler *Mathias Franz* hat sein Headset mit dicken Ohrenschützern kombiniert und so einen *Noise-Cancelling-Kopfhörer* mit ganz besonders viel Stille gebastelt. Da sich die großen Plastikflächen der *Micky Mäuse* gut desinfizieren lassen, sind sie sogar im Reinraum nutzbar.

Der Umbau wurde notwendig, nachdem das Kabel des Kopfhörers zerknickt war. Dafür hat Franz erst sowohl den Kopfhörer als auch die Ohrenschützer auseinandergenommen. Zur Bedienung kamen dann vier Knöpfe für *an/aus*, *play/pause*, *vor* und *zurück* an die Schalen. Später folgte noch ein Knopf zur Bedienung der *Active Noise Cancellation (ANC)*. Anschließend ging es an den Kabelersatz: Um die zehn Adern der Platinen zu verbinden, bot sich ein Netzwerkkabel aus dem Wegwerfstapel an – quasi das I-Tüpfelchen. Ein Pluspunkt: So hat jeder Knopf ein eigenes Kabelpaar mit eigener Farbe. Dann fehlte nur noch ein USB-Breakoutboard für die Spannungsversorgung. Den gesamten Bastelprozess hat Franz auf *Mastodon* mit vielen Bildern dokumentiert.

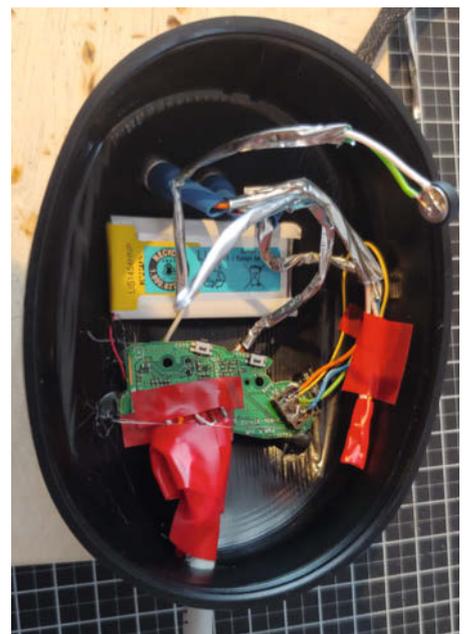
—hch/rehu

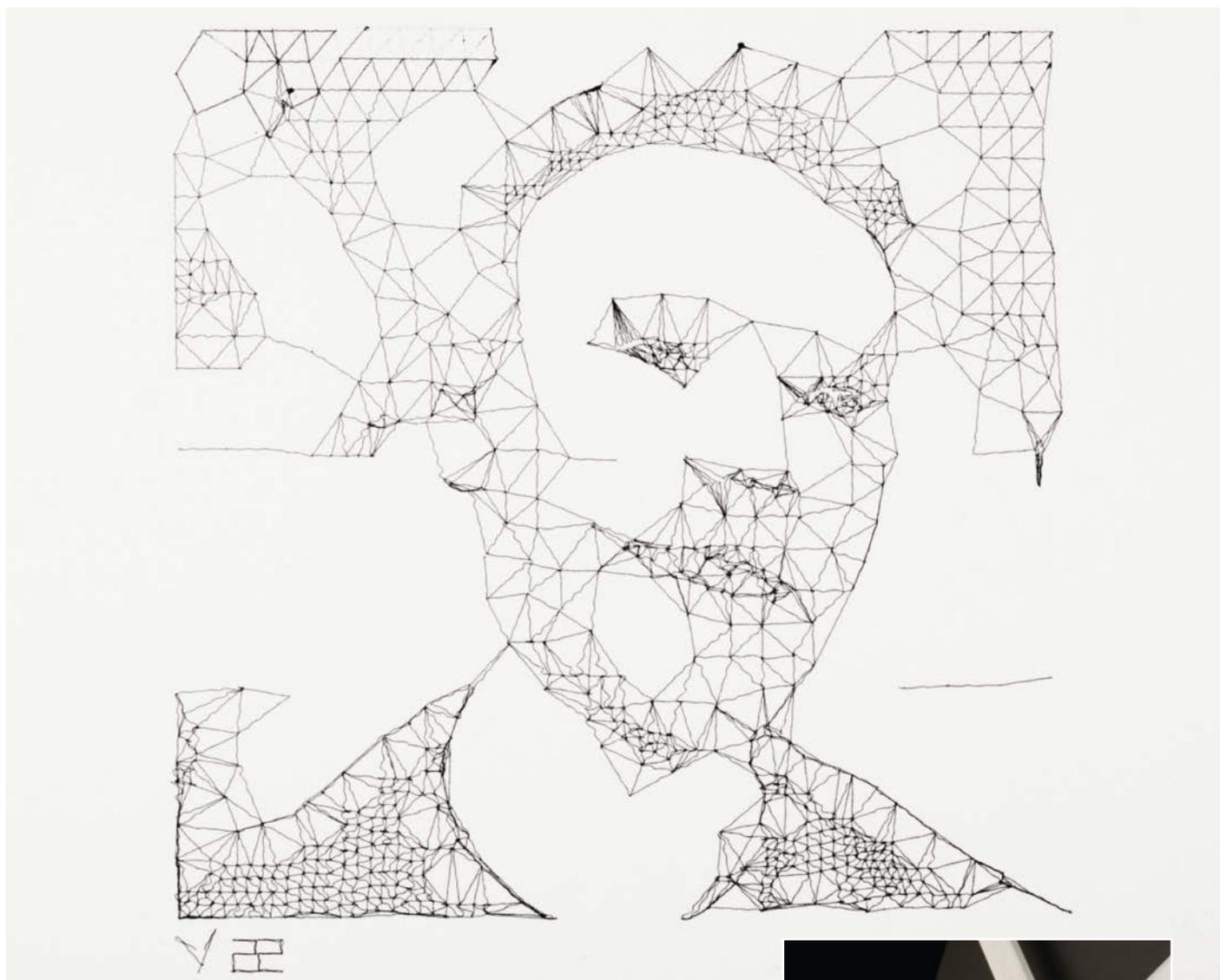
► [nerdculture.de/@M/106325061889938604](https://nerdculture.de/@M/106325061889938604)

## Was inspiriert Dich?

Wir freuen uns über Vorschläge an:

[mail@make-magazin.de](mailto:mail@make-magazin.de)





Felix Fiskus

## Next Level Selfies

Wer lieber programmiert, als selbst zu zeichnen, kommt schnell zum Stiftplotter. Die Mediendesigner Felix Fiskus und Joris Wegner haben den *Pankraz Piktograph* als Selfiemaschine entwickelt. Ihr Plotter besteht aus der Zeichenfläche, zwei Armen mit Stifthalterung, einem kleinen Display mit Anweisungen und einer Kamera. Der Auslöser ist über ein Kabel mit dem Aufbau verbunden.

Für das Display-Interface und die Bildbearbeitung nutzen Fiskus und Wegner *openFrameworks* bzw. die *Programm-bibliothek OpenCV*. Während des *Preprocessing* werden die Farben auf Grautöne umgerechnet und der Kontrast erhöht, um die Konturen zu verstärken. Anschließend geht es zur Vektorisierung und dem *Postprocessing*. Die Arbeit übernimmt im Plotter ein *Raspberry Pi 3*, der auch das Display und die Kamera steuert. Für die Steuerung des Stifts ist ein *Teensy*-Mikrocontroller zuständig. Wer sich den Piktographen genauer ansehen möchte, kann auf der Projektseite die Masterarbeit von Fiskus und Wegner lesen.

—hch

► [felixfigus.de/work/017\\_pankraz\\_piktograph](https://felixfigus.de/work/017_pankraz_piktograph)





# Gewinnerprojekte des Smart-Home-Wettbewerb

Die 10 Gewinner des von AVM und Make gemeinsam veranstalteten Wettbewerbs „Mach dein Smart Home wirklich Smart“ wurden gekürt. Das Spektrum reicht von schicken Einzelstücken bis zum Smart Home am Rande des Over-Engineering.

von Daniel Bachfeld

Alles zum Artikel  
im Web unter  
[make-magazin.de/x5e1](https://make-magazin.de/x5e1)

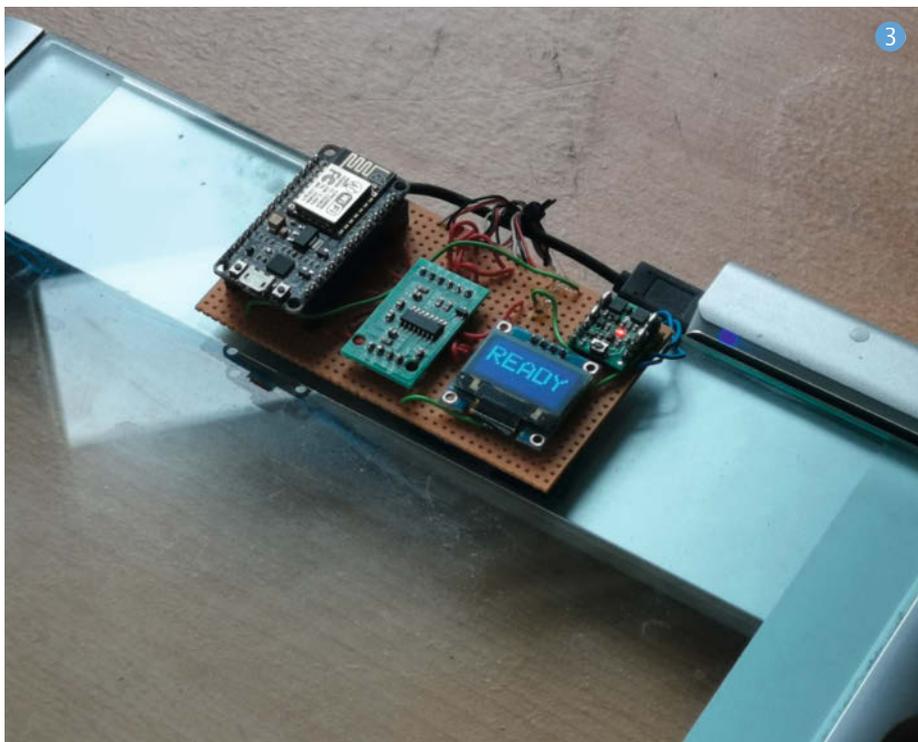
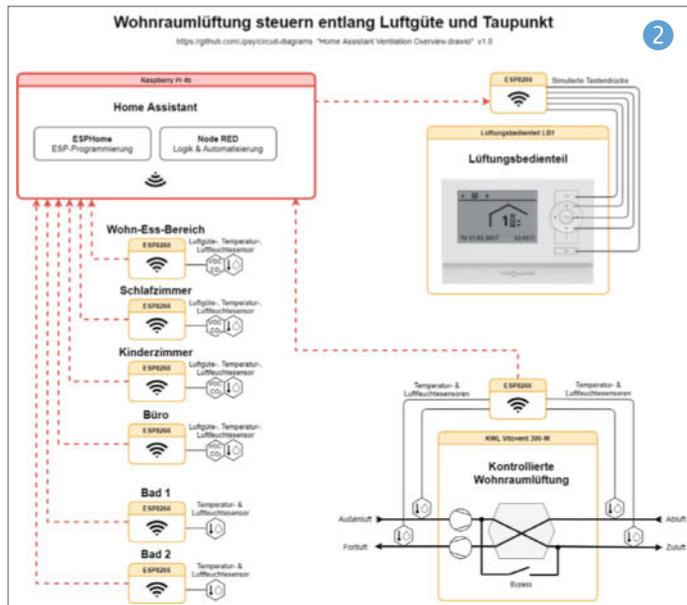
Vom smarten Wecker im Steampunk-Design bis zur Steuerung der Hausinstallation reichte das Spektrum der insgesamt 30 eingereichten Projekte des Wettbewerbs *Mach dein Smart Home wirklich Smart*. Gesucht waren Projekte, die übers WLAN kommunizieren und verschiedene Sensoren und Aktoren miteinander verknüpfen. Bei der Auswahl der Produkte war es den Teilnehmerinnen und Teilnehmern freigestellt, welche zum Einsatz kamen, es mussten keine von AVM sein.

Von Mitte Juni bis Mitte September hatten die Teilnehmer Zeit, auf unserer Plattform *Make: Projects* ihre Umsetzungen mit Texten, Bildern und Videos vorzustellen und zu beschreiben. Als Gewinn winkten Einkaufsgutscheine im Wert von je 500 Euro. Anders als bei anderen Wettbewerben sollte es keinen 1., 2. oder 3. Platz geben. Stattdessen sollten die besten zehn Projekte gekürt werden.

Die Jury, bestehend aus Gordon Priebe von AVM, Mario Lukas, Maker und Entwickler des *FabScan*, Guido Burger, Maker und Entwickler

des *IoT Octopus* sowie Daniel Bachfeld vom *Make-Magazin*, hatte die Aufgabe, aus 30 Projekten die besten heraus zu sieben. Hier stellen wir die Gewinnerprojekte kurz vor. Gratulation an alle Gewinner und ein großes Dankeschön an alle Teilnehmer.

Die ausführlicheren Beschreibungen mit Materiallisten, Anleitungen, Listings sowie die nicht gekürten Projekte finden Sie auf *Make: Projects*. Einige der Projekte bieten 3D-Druckdaten und Listings zum Download an, andere sind zu individuell, um sie nachzubauen. Los geht's:



Das Projekt *Automated Oxygen* 1 – *Luft per Smartphone* von Heiko Eckle (@devBioS) löst ein Problem bei Patienten, die nachts auf eine Sauerstoffversorgung angewiesen sind. Ein mobiler Tank ist nach sechs Stunden bereits leer, ein großer in einem Lagerraum untergebrachter Tank kann mit einem langen Schlauch zwar Abhilfe leisten, erfordert aber längere Fußwege. Dank eines Laser-geschnittenem Getriebes, eines 3D-gedruckten Gehäuses, eines ESP32 und passender Smartphone-App lässt sich das Ventil des

Sauerstoffbehälters ferngesteuert öffnen und schließen.  
*Die klassische KWL wird smart!* 2 heißt der Beitrag von Jörg Wagner (@Jpsy). Das Projekt kombiniert mehrere ESP8266 mit *Home Assistant*, *NodeRed* und *ESPHome*, um in einem Niedrigenergiehaus die „kontrollierte Wohnraumlüftung“ (KWL) zu steuern. Sensoren in den Räumen melden die Luftgüte, Temperatur und die Luftfeuchte an die Zentrale, die über ausgefuchste Algorithmen die Be- und Entlüftung steuert. Aus der zeitlichen Darstellung der

Messwerte aller Sensoren lassen sich sogar überraschende Details herauslesen: nächtliche Flatulenzen und der Verlauf des Atemalkohols.  
*Luna – Large UNnecessary Automation* 3 von Daniel Jäger (@nerdworld) kann unter anderem das gerade mit der Badezimmerwaage (mittels ESP8266 erweitert) ermittelte Gewicht per Sprachausgabe laut im Haus verkünden. Insgesamt 32 Module können auf Ereignisse reagieren oder Ereignisse auslösen. So gibt es noch eine Türschlosssteuerung, Haustürklingel- sowie eine Anrufererkennung,

## Make Projects

um beispielsweise *Spotify* leiser zu stellen und den Namen des Anrufers anzusagen.

Die *Regenwarnlampe* von Jonas Niesner (@jonas44) schützt vor überraschender Durchnässung beim Verlassen des Hauses. Ein *Home-Assistent*-Skript fragt die regionalen Regen-Radar-Datenbanken des Deutschen Wetterdienstes per JSON ab und signalisiert per steuerbarer RGB-Lampe, ob in der Vorhersage Regen droht, wie in 4 zu sehen. Auch ein Mähroboter lässt sich so rechtzeitig zurückbeordern. Über eine Erweiterung kann man auch eine Blitzortung integrieren.

Ein Stück Holz, eine *Trådfri*-Birne, ein *Google Nest Mini* und ein Stück Leder: Fertig ist der *Rustikale SmartHome Wecker* 5 von Michael Ruhland (@michaelruhland). Damit kann man vom Schlafzimmer aus andere *Zigbee*-Geräte per *IOBroker* steuern, ohne dass man das Smarthome neben das Bett legen muss. Die Programmierung erfolgt per *Blockly*.

Smart Home ist auch unterwegs: Wenn man mehrere ESP32-Controller und einen Raspberry Pi mit der Zentraleinheit eines *Westfalia-Campers* verbindet, dann kommt *Smart Home meets Campervan* => *Smart Camper* 6

von Matthias Harzheim (@MatthiasHarzheim) heraus. Damit kann man die Funktionen seines Wohnmobils per Smartphone-App genauer regeln, steuern und überwachen.

*Smarter Bilderrahmen* 7 von Martin Kienast und Sohn Kilian Kienast (14 Jahre) (@Watzmann): Aus einem alten Display, einem aus-rangierten Bilderrahmen und einem Raspberry Pi wird eine Anzeige für Bilder und Statusmeldungen des von *Home Assistant* gesteuerten Smart Homes.

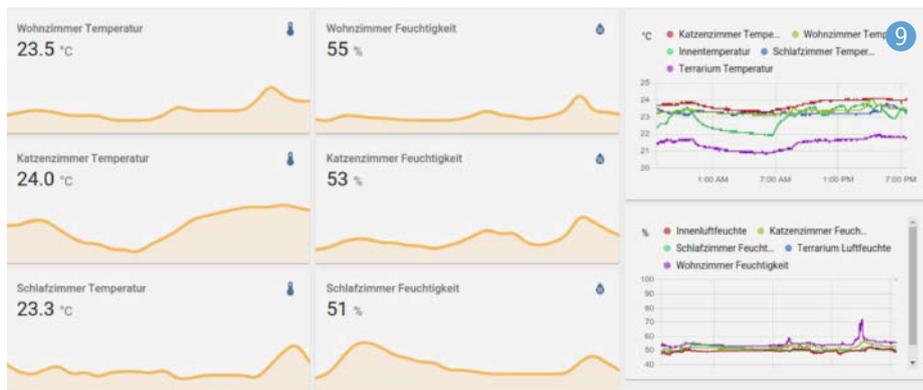
Fünf Schwimmer im Tank signalisieren einem Arduino den Wasserstand. *Der Smarte*



**Wassertank** 8 von Christian Schneider (@christian\_s) schickt per LoRaWAN Daten ins Things Network, die man dort per MQTT zurück ins heimische LAN schicken und mit FHEM auswerten kann. Dank der minutiösen Anleitung ein Projekt zum Nachbauen, auch für Anfänger.

Zwar weniger zur Anleitung geeignet, aber fesselnder Lesestoff: *Trägheit, Tiere, Traumfrau: Die Stakeholder des Smart Homes* 9 von Florian Klug (@flurl) erzählt, wie man Umweltdaten in allen Räumen überwacht, weshalb auf Lichtschalter geklebte Trådfri-Fernbedienungen Ungemach nach sich ziehen und wie und warum man Katzenkot mit Hilfe einer ESP32-CAM und eines Roboterarms sortiert.

Das Projekt *Vorausschauendes Laden des PV-Speichers* 10 von Daniel Kleine-Albers (@dka23) beschreibt einen einfachen Algorithmus, der den Speicher einer Photovoltaik-Anlage so lädt, dass am Abend der Speicher voll ist, tagsüber die Anlage aber möglichst nie abregelt, um das 70%-Einspeiselimiteinzuhalten. Dabei helfen der Vorhersagedienst Solcast, OpenHAB und Grafana zur Überwachung. —dab



## 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](http://www.dpunkt.de)



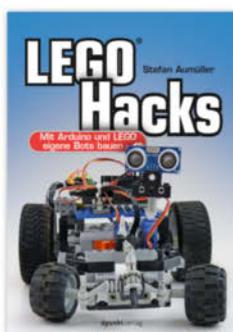
212 Seiten · 26,90 €  
ISBN 978-3-86490-799-9



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



# Füllstandmessung einer Regenwasserzisterne

Für den selbstgebauten Füllstandsensor meiner Regenwasserzisterne kommt ein ausrangierter Blutdruckmesser zum Einsatz, der sich per ESP-Board in mein SmartHome einklinkt.

von Wolfgang Neußer



In meinem Garten befindet sich eine Regenwasserzisterne, vier Meter tief, zwei Meter Durchmesser, Fassungsvermögen ca. 9550 Liter. Die Zisterne (siehe Aufmacherbild) ist mit einem schweren Gullideckel verschlossen, die Kontrolle des Füllstandes somit optisch nicht möglich. Trotzdem hätte ich gerne eine Information über die Höhe meiner Wasserreserven, und diese sollte eventuell auch in meinem Heimnetz verfügbar sein. Erste Versuche, mit Ultraschallsensoren die Füllhöhe zu ermitteln, habe ich schnell wieder aufgegeben; zu viele Reflexionen ließen kein eindeutiges Ergebnis zu. Außerdem konnte ich keine WLAN-Verbindung aus der Zisterne herstellen, zudem hat das feuchte Klima in der Zisterne meinen Basteleien schnell zugesetzt.

Bei der Recherche nach industriellen Messsystemen, die mir letztlich alle zu teuer waren, ist mir ein System besonders aufgefallen: Messung des hydrostatischen Drucks am Zisternenboden per Gegendruck. Dazu muss nur ein nach unten offener Schlauch in die Zisterne herabgelassen werden, die Messvorrichtung **1** selbst kann an einem beliebigen Ort außerhalb platziert werden.

Das Messprinzip: Mittels einer Handpumpe wird Luft in den Schlauch gepumpt. Der ansteigende Druck verdrängt das Wasser im Schlauch. Wenn sämtliches Wasser verdrängt ist und die Luft am Schlauch-Ende austritt, steigt der Druck nicht mehr weiter an. Der jetzt anstehende Druck entspricht der verdrängten Wassersäule, woraus man deren Höhe und damit den Wasserstand errechnen kann. Aus der Formel **2** zur Berechnung des hydrostatischen Druckes ist ersichtlich, dass nur die Höhe des Wasserstandes maßgebend ist, Form und Innendurchmesser des Schlauches spielen keine Rolle.

Dies erschien mir interessant für ein Selbstbauprojekt, doch wie umsetzen? Schließlich braucht man eine Druckpumpe und einen Drucksensor. Dabei kam mir folgende Idee: Der maximale Füllstand in meiner Zisterne beträgt 3,04m, das entspricht 3040mm Wasser-



## Kurzinfo

- » Recycling alter Blutdruckmesser
- » Mikrocontroller misst hydrostatischen Druck und errechnet Füllstand
- » WLAN-Anbindung mit ESP8266

### Checkliste



#### Zeitaufwand:

3 Stunden



#### Kosten:

35 Euro ohne Blutdruckmesser

### Werkzeug

- » **Maker-Werkzeuge** Schraubendreher, Zangen, Cutter ...
- » **LötKolben und Zubehör**
- » **Bohrmaschine und Bohrer**

### Material

- » **Blutdruckmesser** vom Schrott oder billig neu
- » **SparkFun Qwiic MicroPressure Sensor**
- » **ESP 8266 ESP-12E AI-Thinker** oder Dolla-Tech
- » **Erweiterungskarte Doit oder Dolla Tech L293D WiFi Motor Drive**
- » **Pneumatik-Polyurethan-Schlauch (PUN)**
- » **USB-Netzteil**

### Mehr zum Thema

- » Carsten Meyer, Breakout-Boards, Make 1/21, S. 10
- » Robert Kränzlein, LoRaWAN-Sensor für den Heizöltank, Make 6/20, S. 66
- » Simon Ilgenfritz und Tobias Lohf, Datenlogger für den Stratosphärenflug, Make 6/19, S. 70

↓ Alles zum Artikel im Web unter [make-magazin.de/x6mc](http://make-magazin.de/x6mc)

## Pascal'sches Gesetz [ Bearbeiten | Quelltext bearbeiten ]

2

Der hydrostatische **Druck** für **Fluide** mit konstanter **Dichte** im **homogenen Schwerefeld** (= **Inkompressible Fluide**, insbesondere **Flüssigkeiten**) berechnet sich nach dem **Pascal'schen** (oder **pascalschen**) **Gesetz** (benannt nach **Blaise Pascal**):

$$p(h) = \rho gh + p_0$$

Formelzeichen:

$\rho$  = **Dichte** [für Wasser:  $\rho \approx 1.000 \text{ kg/m}^3$ ]

$g$  = **Schwerebeschleunigung** [für Deutschland:  $g \approx 9,81 \text{ m/s}^2$ ]

$h$  = **Höhe des Flüssigkeitsspiegels** über dem betrachteten Punkt

$p_0$  = **Luftdruck** auf Flüssigkeitsoberfläche

$p(h)$  = **hydrostatischer Druck** in Abhängigkeit von der Höhe des Flüssigkeitsspiegels.<sup>[1]</sup>

**Einheiten** [ Bearbeiten | Quelltext bearbeiten ]

Die **Physikalischen Einheiten** für den hydrostatischen Druck sind:

- international die **SI-Einheit**

**Pascal** (Pa):  $1 \text{ Pa} = 1 \text{ N/m}^2$ ;

- zudem in Deutschland und Österreich die „gesetzliche Einheit“

**Bar** (bar):  $1 \text{ bar} = 100.000 \text{ Pa}$  bzw.  $\text{N/m}^2$  (=  $100 \text{ kPa}$ )

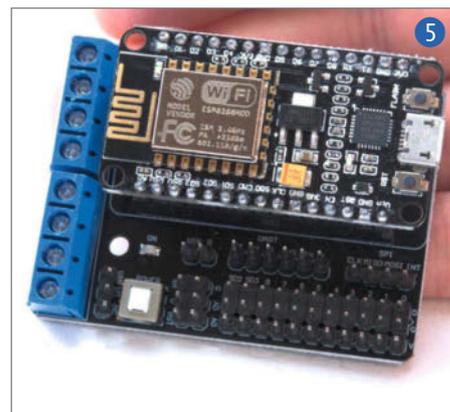
Zur Beschreibung des hydrostatischen Drucks wird zum Teil auch noch die nicht-SI-konforme veraltete Maßeinheit **Meter Wassersäule** (mWS) verwendet.



3



4



5

Sparkfun

Dolla Tech

säule bzw. 29810 Pascal (Pa) Druck. Ein elektronischer Blutdruckmesser misst Drücke im Bereich bis ca. 300mmHg, das entspricht etwa 40000Pa. Außerdem beinhaltet ein solches Gerät alle Elemente, die für die Füllstandmessung notwendig sind.

Zufällig hatte ich gerade meinen alten Blutdruckmesser **3** wegen defekter Oberarmmanschette ausrangiert, also ging es gleich ans Umsetzen. Im Prinzip wäre das Gerät direkt verwendbar, wenn man die Firmware für eigene Zwecke umschreiben könnte. Doch das ist mir leider nicht möglich. Auch den integrierten Drucksensor konnte ich leider nicht verwenden, da ich keinerlei Informationen darüber hatte. Dafür habe ich einen passenden Ersatz **4** im Internet gefunden, den *Sparkfun Micropressure Sensor*.

Als Controller hatte ich mir den ESP8266 herausgesucht, vor allem wegen der einfachen WLAN-Anbindung. Jetzt waren noch

Treiber zur Steuerung von Pumpe und Ventil erforderlich. Auch dafür habe ich eine praktikable Lösung gefunden, die alles Nötige mitbringt **5**. Das *Doit L293D WiFi Motor Drive Board* trägt einen ESP8266, stellt einen Motortreiberbaustein bereit und kann mit 5V betrieben werden.

### Die Umsetzung

Jedes handelsübliche elektronische Blutdruckmessgerät ist geeignet. Von Vorteil sind Oberarmmessgeräte. Diese haben einen zentralen Schlauchanschluss und bieten genug Platz für eigene Einbauten. Neue Geräte sind schon ab ca. 25 Euro erhältlich, manchmal sogar noch günstiger. Das von mir verwendete Gerät ist ein *Sanitas SBM 30*. Unter der Anzeige und Hauptplatine **6** liegen die Druckpumpe und darunter das Entlüftungsventil **7**. Das graue Kunststoffteil oben links an einem Schlauchende ist die Zwangsentlüftung. Damit wird zum Messen des Blutdruckes, nachdem die Pumpe abgeschaltet wurde, die Manschette langsam entlüftet und damit der Druck verringert.

Achtung! Diese Zwangslüftung muss unbedingt entfernt werden, wir wollen ja ein dichtes System haben, in dem sich nur durch

den Füllstand der Druck ändert. Ich hatte leider keinen passenden Blindstopfen, deshalb habe ich das Bauteil abgezogen, mit ein paar Tropfen Kleber verschlossen und anschließend wieder aufgesteckt.

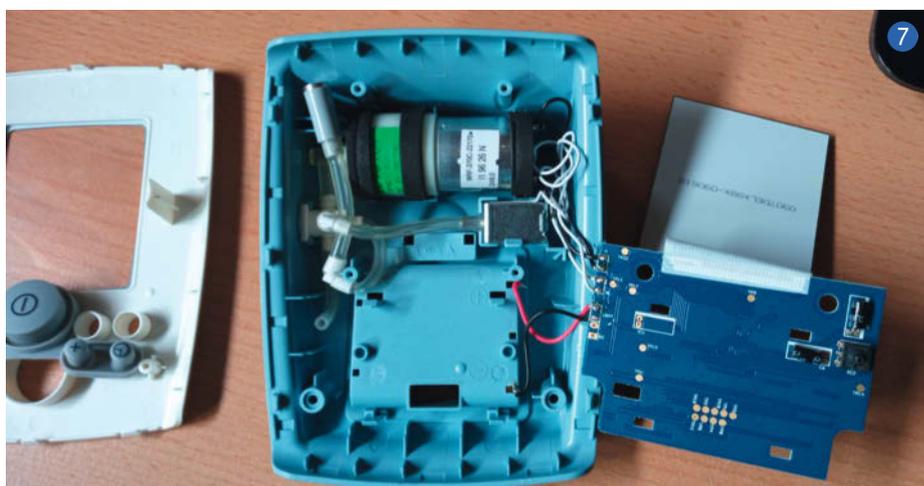
Mit dem Entlüftungsventil (Pneumatikschaltplan **8**) wird normalerweise nach der Blutdruckmessung die Oberarmmanschette komplett entlüftet. Das Ventil ist im Ruhezustand offen, muss also bei der Messung elektrisch geschlossen werden. Kein Problem; es stehen genügend IO-Ports am ESP zur Verfügung und ein zweiter Motortreiber ist auch vorhanden. Der offene Schlauch links unten war auf dem Drucksensor aufgesteckt, dort wird auch der neue Sensor angeschlossen.

Die Platine wird nicht mehr benötigt und muss aus Platzgründen entfernt werden. Damit sind leider auch die Bedientasten nicht mehr verwendbar. Die vier Drähte zum Entlüftungsventil und zur Pumpe werden an der Platine abgelötet, diese werden später am ESP-Entwicklungsboard wieder angeschlossen. Sollte man keinen ausrangierten Blutdruckmesser bekommen können, so gibt es Pumpen, Ventile und Schlauchverbinder natürlich auch im Versandhandel.

Das Batteriefach wird ebenfalls nicht mehr benötigt, ich habe es vorsichtig herausge-



6



7

schnitten. Hier passt jetzt das Entwicklungs-board gut hinein.

### Vorbereitung des Boards

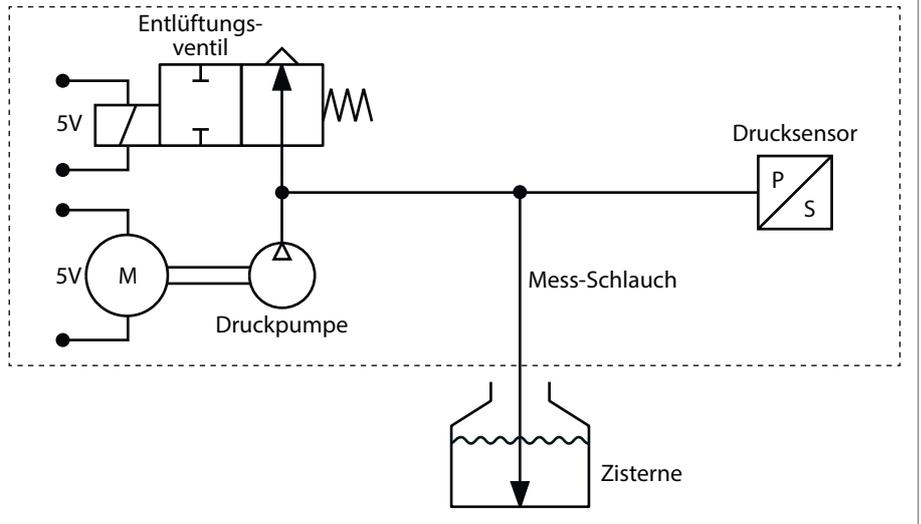
Das Board (Schaltplan 9 Mitte) wird mit 5V über die ESP-Power-Pins versorgt, für das Betreiben von Motoren ist ein zweiter Anschluss (Motor Power) für Spannungen von 4,5V–36V vorhanden. Da die Pumpe und das Ventil ebenfalls mit 5V betrieben werden, können die Motortreiber per Steckbrücke an die 5V angekoppelt werden. Somit benötigen wir nur eine 5V-Stromquelle. Ein passendes USB-Netzteil ist schnell in der Bastelkiste gefunden. Allerdings sollte es genügend Strom liefern können, etwa 1A sind ausreichend. Für die Stromzufuhr habe ich das USB-Kabel aufgetrennt und durch eine 5mm Bohrung in das Gehäuse geführt. Ein Kabelbinder dient als Zugentlastung. Auf dem Board stehen zum Anschluss Schraubklemmen zur Verfügung, ebenso für Ventil und Pumpe.

Die digitalen Ports (GPIOs) des ESP sind auf einer Stiftleiste am Board verfügbar, jeweils in Reihe mit einem GND- und einem 3V3-Pin. Achtung, bei Varianten des Boards (Doit oder Dolla Tech z.B.) sind die GND- und VCC-Pins und weitere Pins vertauscht, bitte prüfen Sie dies vor dem Anschließen. Mangels einer Dokumentation musste ich die Zuordnung der Pins zu den tatsächlichen GPIOs durch Messen herausfinden, siehe Tabelle 10.

Der Drucksensor verfügt über eine I<sup>2</sup>C-Schnittstelle und muss mit 3,3 Volt versorgt werden. Die vier Leitungen SDA, SCL, 3V3 und GND 11 habe ich an die passenden Stifte am Board direkt angelötet. Damit ist die Verdrahtung 11 abgeschlossen.

### Pneumatik-Schaltplan

8



Als Messleitung habe ich einen Pneumatik-Polyurethan-Schlauch (PUN) mit 4mm Außendurchmesser verwendet. Am Schlauchende wurde ein Verbinder mit 4mm Durchmesser aufgesteckt und daran eine Schnur mit einem kleinen Gewicht befestigt 12.

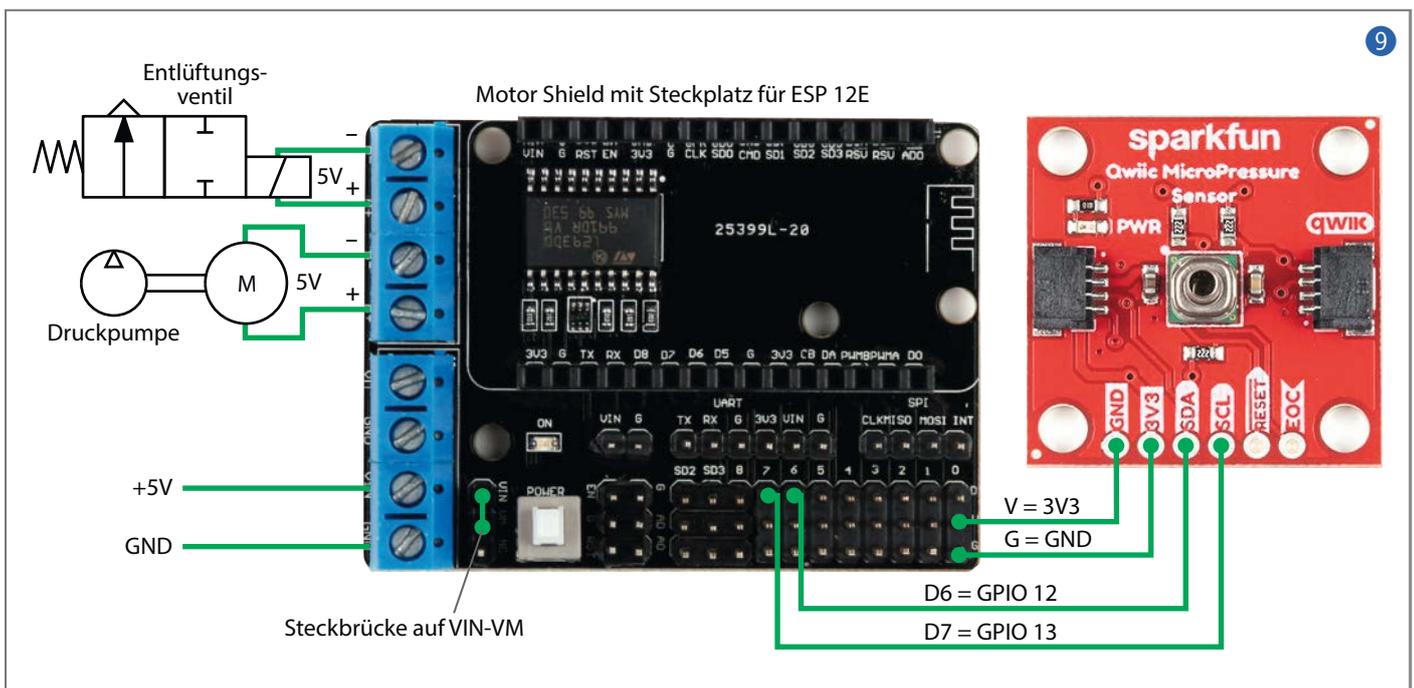
Diesen Schlauch habe ich durch das Versorgungsrohr zu meiner Zisterne geschoben und dann in die Zisterne herabgelassen, bis das Gewicht den Boden berührt. Der Schlauchverbinder sollte nicht den Boden berühren, damit kein Schmutz in die Leitung gelangt. Jetzt entspricht der Abstand zwischen Schlauchende und meinem maximalen Wasserspiegel genau 3,04 m, also 3040 mmH<sub>2</sub>O (Millimeter Wassersäule), die mein System bei maximalem Füllstand messen soll.

Das andere Ende des Schlauches kommt auf meiner Terrasse an und führt in einen kleinen wetterfesten Elektrokasten, in den auch das Messgerät mit dem USB-Netzteil platziert wird 13. Den Originalschlauch am Blutdruckmesser habe ich gekürzt und den Messschlauch hineingeschoben, das passt zufällig genau. Mit einem Kabelbinder kann das Ganze noch fixiert werden.

### Software

Für den Messablauf habe ich einen Arduino-Sketch erstellt, den Sie über die Links in der Kurzinfo von GitHub beziehen können. Er beinhaltet den Messvorgang mit Bedienung und Rückmeldung über die serielle Schnittstelle

9



Dolla-Tech/Sparkfun

### 10 Zuordnung der Motorshield-Pins zu den ESP-GPIOs

Pin	GPIO	interne Verbindung	externe Verbindung
D0	16		
D1	5	Motor A PWM	
D2	4	Motor B PWM	
D3	0	Motor A Richtung	
D4	2	Motor B Richtung	
D5	14		
D6	12		SDA Sensor
D7	13		SCL Sensor
D8	15		
SD3	10		
SD2	9		

bzw. WLAN. Natürlich könnten Sie die Messung auch über einen digitalen Eingang mit einem Taster starten oder automatisch in einem beliebigen Zeittakt. Für die Ausgabe des Füllstandes könnte ich mir auch ein Display oder eine Siebensegmentanzeige gut vorstellen.

Die hier angegebenen Zeilennummern beziehen sich auf den Code vom 5. November 2021 aus dem GitHub.

Neben der üblichen Bibliothek für WLAN wird auch die Bibliothek *ArduinoOTA* eingebunden. Damit kann nach dem ersten seriellen Flashen künftig über WLAN geflasht werden (OTA: **O**ver **T**he **A**ir). Das geht schneller und benötigt keine kabelgebundene serielle Verbindung mehr.

Der Drucksensor wird über I<sup>2</sup>C ausgelesen. Deshalb muss die Standardbibliothek *Wire.h* eingebunden werden. Für die Kommunikation mit dem Sensor ist die Bibliothek *SparkFun\_MicroPressure* zuständig. Diese kann im Bibliotheksverwalter unter *MicroPressure* gesucht oder aus dem GitHub-Repository abgerufen werden.

Für die Verbindung zum Heimnetz müssen die Parameter zur SSID und dem Passwort des Netzes in den Zeilen 59/60 **14** eingetragen sein. In die Konstante A trägt man die Grundfläche der Zisterne in cm<sup>2</sup> ein. Zusammen mit der gemessenen Füllhöhe kann so das aktuelle Volumen errechnet werden.

In der Routine *setup()* werden die digitalen IO's initialisiert, wie in den Zeilen ab 46 definiert. Der Motortreiber auf dem Board hat für die Motoren A und B jeweils einen Eingang für PWM (Pulsweitenmodulation) und Richtung. Ich habe jedem Eingang einen GPIO des Controllers zugeordnet, obwohl für unsere Anwendung der PWM-Eingang allein genügen würde. Nach der Initialisierung der seriellen Schnittstelle werden die I<sup>2</sup>C-Schnittstelle und der Sensor initialisiert. Sollte keine Kommunikation mit dem Sensor zustande kommen, bleibt die Routine hier in einer Endlosschleife hängen. Möglicherweise liegt dann ein Fehler in der Verdrahtung vor.

Anschließend wird das WLAN initialisiert und die Verbindung zum Heimnetz hergestellt. Nachdem der Server aktiviert ist, wird die aktuelle IP-Adresse auf der seriellen Schnittstelle ausgegeben und kann so mit dem seriellen Monitor der Arduino-IDE angezeigt werden. Mit dieser Adresse kann der Server im Heimnetz von einem Browser aufgerufen werden. In den Zeilen 180–184 wird festgelegt, dass beim Start des Flash-Vorganges über WLAN (was ja zu jedem beliebigen Zeitpunkt vorkommen kann) Pumpe und Ventil ausgeschaltet werden. Danach wird ein Passwort (*esp8266*) für das Flashen per WLAN festgelegt und die OTA-Funktion aktiviert.

Als Letztes wird die Variable *atmDruck* mit einem Startwert initialisiert.

Im Hauptprogramm *loop()* wird zyklisch die OTA-Routine aufgerufen, um auf eine Flashanforderung reagieren zu können. Danach wird die Steuerung über die serielle Schnittstelle bearbeitet (ab Zeile 202). Über den seriellen Monitor kann mit dem Kommando *start* der Messablauf gestartet werden. Mit den Kurzkommandos *v0,v1* und *p0/p1* können Ventil und Pumpe zu Testzwecken direkt geschaltet werden. Mit der Eingabe von ? (Fragezeichen) werden alle gültigen Kommandos angezeigt.

Ab Zeile 241 wird alle 10ms der Sensor ausgelesen. Durch den Aufruf der Sensorroutine *readPressure()* mit dem Parameter PA wird der gemessene Wert in Pascal zurückgegeben (weitere mögliche Einheiten sind *KPA, TORR, INHG, ATM* und *BAR*). Der Aufruf findet innerhalb einer Filterroutine statt, die den Mittelwert aus 50 Messungen bildet. Anschließend wird der Druck in Millimeter Wassersäule (mmH<sub>2</sub>O) umgerechnet. Dazu muss erst der atmosphärische Druck *atmDruck* vom aktuellen, hydrostatischen Druck abgezogen werden, danach folgt die Umrechnung (1Pa = 0,101972 mmH<sub>2</sub>O). Zur Sicherheit wird bei einem Messwert von mehr als 300mm über der maximalen Füllhöhe die Pumpe ausgeschaltet.

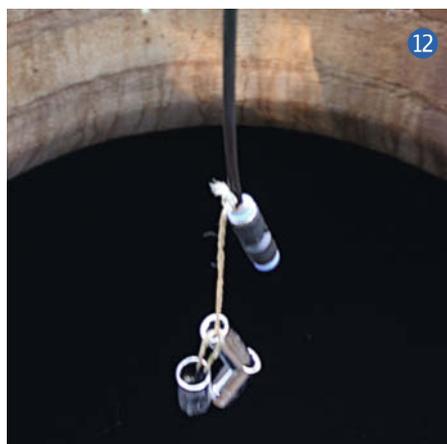
Als letztes wird der Server abgefragt. Steht eine Client-Anfrage an, werden die letzten Ergebnisse als eine einfache Website übertragen und eine Messung gestartet. Dazu wird mit der Funktion *client.print()* direkt der HTML-Code mit eingebetteten Messwerten **15** ausgegeben.

Durch die Angabe *http-equiv='refresh' content='10'* in Zeile 264 startet der Client automatisch alle 10s eine neue Anfrage. In Zeile 265 wird die Größe der Ausgabe an das jeweilige Display angepasst.

Für den schrittweisen Ablauf des Messvorganges ist eine State-Machine in Routine *messablauf()* (Zeilen 73–132) zuständig. Solange die Variable *messSchritt* auf 0 ist, wird in jedem Programmzyklus der aktuelle Druck in die Variable *atmDruck* übertragen. Das ist der aktuelle atmosphärische Luftdruck, der für die Ermittlung des Wasserstandes benötigt wird. Dieser Wert ist abhängig von der geografischen Lage und kann sich je nach Witterung ändern. Der Sensor erfasst diesen Druck während das Entlüftungsventil offen ist. Sobald die Variable *messSchritt* auf 1 gesetzt wurde, beginnt der Ablauf.

**Schritt 1:** Der aktuelle Druck wird als Vergleichswert festgehalten, das Entlüftungsventil wird geschlossen und die Druckpumpe eingeschaltet. Start der Mindestmesszeit von zwei Sekunden.

**Schritt 2:** Hier beginnt die Überwachung des Druckes durch Vergleich mit dem Ver-



gleichswert. Ist der aktuelle Druck um 10Pa höher als der Vergleichswert, wird der aktuelle Wert zum neuen Vergleichswert mit Verlängerung der Messzeit um eine Sekunde. Steigt die ermittelte Wassersäule auf 200mm über dem Maximalfüllstand, so wird die Messung abgebrochen und direkt mit Schritt 4 weitergearbeitet. In diesem Fall muss eine Verstopfung des Messschlauches vorliegen. Ist die Messzeit abgelaufen, ohne dass der Druck weiter gestiegen ist, wird die Pumpe ausgeschaltet und die Beruhigungszeit gestartet.

**Schritt 3:** Nach Ablauf der Beruhigungszeit wird der aktuelle Wasserstand in 0 bis 100% Füllstand umgerechnet und zusammen mit der Füllhöhe in Zentimetern und dem errechneten Volumen in Liter ausgegeben. Die gleiche Ausgabe erfolgt über WLAN, wenn sich ein Client angemeldet hat.

**Schritt 4:** Hier wird das Ende des Ablaufs durch das Öffnen des Entlüftungsventils eingeleitet. Die Zustandsvariable `messSchritt` wird wieder auf 0 gesetzt.

### Erfahrungen

Die Länge und der Durchmesser des Messschlauches sind unerheblich für den Messwert. Aber je länger und je größer im Durchmesser der Schlauch ist, umso mehr Luft muss gepumpt werden, was je nach Kapazität der Pumpe erheblich mehr Zeit benötigt.

Wenn die komplette Wassersäule aus dem Schlauch gedrückt ist, bilden sich am Schlauchende Luftblasen, die zu kleinen Druckschwankungen führen. Deshalb ist die Mittelwertbildung und die Beruhigungszeit nach dem Abschalten der Pumpe nötig.

Wenn das komplette Pneumatiksystem, so wie bei mir, nicht zu 100% luftdicht ist, darf die Beruhigungszeit nicht zu lang sein, sonst wird das Messergebnis zu sehr verfälscht. Bei mir ist möglicherweise die Pumpe nicht absolut dicht. Die Leckage ist allerdings gering, die Genauigkeit des Ergebnisses reicht mir als Aussage über den Füllstand völlig aus.

Seit dem Frühjahr dieses Jahres ist der Füllstandsmesser bei mir im Einsatz. Ich habe ihn zusätzlich in mein persönliches SmartHome-System integriert und bin so immer über meinen Wasservorrat informiert. —caw



13

```

40 // Ohne Parameter werden Default Werte verwendet
41 SparkFun_MicroPressure mpr;
42
43 // Bibliothek für das Flashen über WLAN
44 #include <ArduinoOTA.h>
45
46 // Zuordnung der Ein- Ausgänge
47 #define VENTIL      5           // GPIO5 (PWM MotorA)
48 #define DA         0           // GPIO8 (Richtung MotorA)
49 #define PUMPE      4           // GPIO4 (PWM MotorB)
50 #define DB         2           // GPIO2 (Richtung MotorB)
51 #define SDA        12          // GPIO12 I2C
52 #define SCL        13          // GPIO13 I2C
53 #define AUF        LOW        // Ventil öffnen
54 #define AUS        LOW        // Pumpe ausschalten
55 #define ZU         HIGH       // Ventil schliessen
56 #define EIN        HIGH       // Pumpe einschalten
57
58 // Heimnetz Parameter (an eigenes Netz anpassen)
59 const char* ssid = "Heimnetz-Name";
60 const char* pass = "Heimnetz-Passwort";
61
62 // An eigene Zisterne anpassen (zur Berechnung der Füllmenge)
63 const int A = 3140;           // Grundfläche der Zisterne in cm^2 (d * d * 3,14 / 4)
64 const int maxFuellhoehe = 3040; // Füllhöhe der Zisterne in mm
    
```

14

### State-Machine, ein endlicher Zustandsautomat

Bei einem sogenannten *Endlichen Zustandsautomaten* (engl. *Finite State Machine*, kurz *FSM*) handelt es sich um ein Steuerungskonzept, welches eine abstrahierte Maschine zum Vorbild hat, die eine Reihe von Zuständen besitzt. Diese Zustände sind durch die benötigten Funktionen definiert, hier also Funktionen zum Messen, Beruhigungszeit oder Abbruch der Messung bei Fehlern.

Während der Automat arbeitet, gehen die Zustände ineinander über oder verharren, bis ein Ereignis eintritt, z. B. die Beruhigungszeit abgelaufen ist. Damit

wird dann zum nächsten definierten und sinnvollen Zustand geschaltet. Während der Zustände reagiert der Automat nur auf definierte Ereignisse, die wiederum bestimmen, welcher Zustand als Nächstes aufgerufen wird.

Der Zustandsautomat wird immer über einen Takt aufgerufen. In der Hauptschleife des vorliegenden Arduino-Sketchs etwa alle 10ms. Damit kann der Automat nicht sofort auf Ereignisse reagieren und hat eine gewisse Latenz, was aber nur bei sehr schnellen Echtzeitprozessen stört, bei der Füllstandsmessung aber nicht.

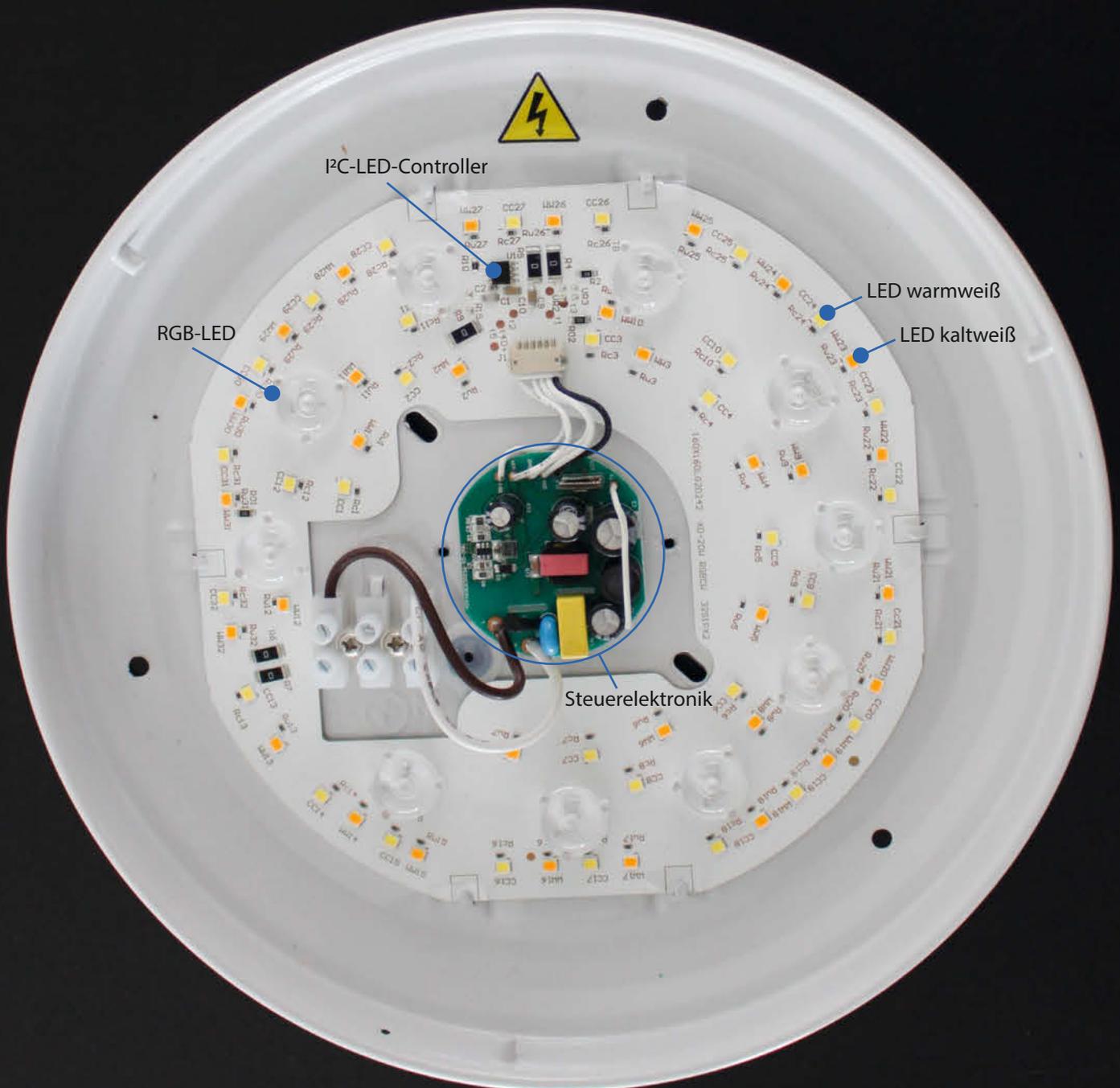


15

# Smarthome vom Discounter

Was steckt in den billigen smarten Lampen, Schaltsteckdosen, Bewegungsmeldern und ähnlichem vom Non-Food-Discounter Action? Wir haben das Innenleben untersucht.

von Heinz Behling



Per WLAN steuerbare LED-Birnen für 3,95€ ①, Außenleuchten mit Bewegungsmelder oder Deckenleuchten für 10,95€, Schaltsteckdosen für 7,49€? Da kann doch nicht viel an Technik drinstecken. Oder wird es bei der Integration ins Smarthome schwierig? Das waren die Gedanken, als ich auf den Internetseiten von Action diese sehr preiswerten Smarthome-Komponenten entdeckte, die unter dem Label *Smart Connect* verkauft werden.

Eine erste Recherche im Internet ergab, dass die vom Hersteller LSC stammenden Teile allesamt mit der Tuya-Firmware arbeiten. Die Steuerung erfolgt dabei per Smartphone-App über die Cloud des chinesischen Unternehmens. Möchte man solche Komponenten ins eigene Smarthome-System integrieren, muss man den Smarthome-Server von außen per Internet zugänglich machen. Da kann man schon mal Unbehagen empfinden angesichts der immer häufiger werdenden Hackerangriffe auf Netze und Server von Cloudbetreibern. Wer das nicht möchte, muss solche Komponenten irgendwie von der Cloud abnabeln.

Im Netz fanden sich glücklicherweise zahlreiche Hinweise, dass im Inneren der LSC-Geräte Controller vom Typ ESP8266 werkeln würden und sich die Tuya-Firmware problemlos gegen das cloudlos-arbeitende ESPHome austauschen ließe. Wie das geht, hat die c't bereits Anfang 2019 in einem Online-Artikel gezeigt (siehe Link am Ende des Artikels). Noch ein Grund, mal in die Teile hineinzuschauen und dies abzuchecken.

Erstes Problem: Nahezu alle Geräte waren gegen zerstörungsfreies Öffnen durch Verkleben sehr wirksam geschützt. Einzig die Deckenleuchte ② offenbarte ohne weiteres nach Abnehmen der sehr wabbeligen Haube ihr Innenleben.

Aber die LED-Ausstattung konnte sich sehen lassen: 64 weiße LEDs (je zur Hälfte warm- und kaltweiß) sowie 10 RGB-LEDs mit aufgesetzten Lupen zur Vergrößerung des Abstrahlwinkels sitzen auf der ringförmigen Platine (siehe Titelbild links).

In der Mitte thront in einem Kunststoffgehäuse die Steuerelektronik ③, die von außen über eine Lüsterklemme die Netzspannung erhält und an der anderen Seite einen I<sup>2</sup>C-Bus zum entsprechenden IC auf der LED-Platine weitergibt.

Die Steuerelektronik enthält neben einem kleinen Schaltnetzteil, das 3,3V für den Controller und die Spannung für die LEDs bereitstellt, auch das Controller-Board vom Typ WBLCS. Dies ist eine Tuya-Eigenentwicklung mit einem 32-Bit-Controller, WLAN und Bluetooth BLE (Datenblatt siehe Link zum Artikel), aber leider kein ESP-kompatibles Board. Damit war die Chance des Firmware-Flashens vorbei, wie auch ein Versuch mit der Software *Tuya-Convert* bestätigte.



① Schaltsteckdosen zum sehr günstigen Preis



② Die Deckenleuchte zeigte, was in ihr steckt.

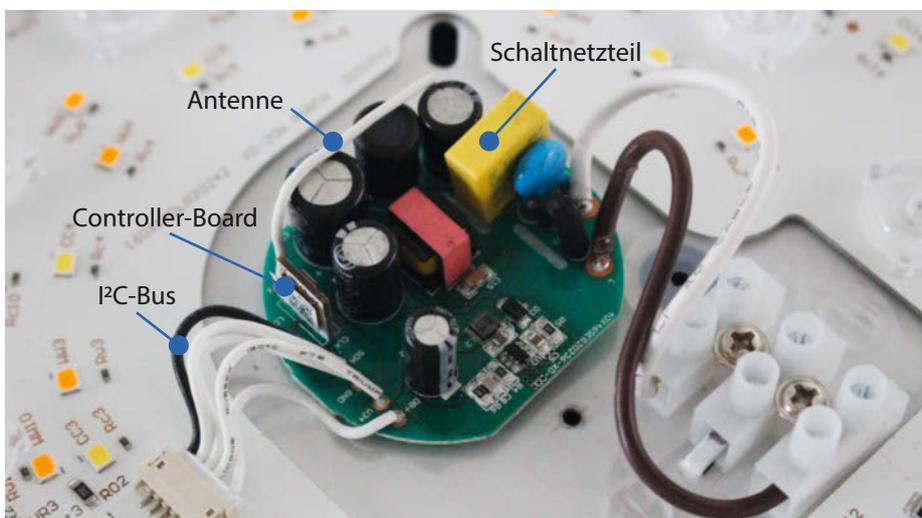
Auch die mehr oder weniger gewaltsam geöffneten anderen Smart-Collect-Komponenten wie Birnen, Steckdosen und so weiter enthielten dieses oder ähnliche Boards. Offenbar hat der Gerätehersteller LSC seine Produkte flash-sicher umgearbeitet.

### Alternativen zum Flashen

So schnell gab ich jedoch nicht auf und machte mich auf die Suche nach anderen Lösung als dem Firmware-Austausch. Bei der Deckenleuchte mit ihrem I<sup>2</sup>C-Bus könnte man beispielsweise den Original-Controller durch ein ESP-01-Modul ersetzen, auf dem dann ESPHome als Firmware sitzt. Das ließe sich problemlos durch Smarthome-Server wie Home Assistant steuern. Da das kleine Controller-Board mit nur vier Kontakten an die LED-Platine gelötet ist, von denen zwei GND und 3,3V sind, wäre die Zuordnung der I<sup>2</sup>C-Pins kein Problem.

Doch das löst nicht das Problem der verklebten anderen Geräte, deren Gehäuse man zerstören musste, um den Controller zu tauschen. Außerdem ist der Platz in einer LED-Birne so eng, dass man da kaum etwas nachträglich ändern kann. Zumindest für Home Assistant fand ich da auch eine Lösung: Local Tuya. Dies ist eine HACS-Integration (also eine von Anwendern programmierte Erweiterung, die im **Home Assistant Community Store** erhältlich ist) für Home Assistant, die im Grunde nichts weiter als eine auf dem Home-Assistant laufende Tuya-Cloud darstellt. Wie man das einrichtet, lesen Sie im Artikel *Tuya ohne Cloud* ab der nächsten Seite. —hgb

Alles zum Artikel im Web unter [make-magazin.de/x7u9](https://make-magazin.de/x7u9)



③ Die sauber aufgebaute Steuerelektronik

# Tuya ohne Cloud

Die cloudbasierte Smarthome-Firmware Tuya kann auch ohne Kontakt zum Server in Fernost rein lokal gesteuert werden. So haben Hacker, Server-Kaperer oder Cloud-Ausfälle keine Chance mehr und Sie behalten die Kontrolle über Ihr Heim.

von Heinz Behling



Das smarte Heim, in dem man sich um (fast) nichts mehr selbst kümmern muss! Das muss doch zu schaffen sein. Ist es auch. Die notwendigen Geräte gibt es inzwischen preiswert nicht nur direkt in China, sondern sogar beim Discounter. Noch ein paar Euros günstiger als die meist ZigBee-kompatiblen Teile von Lidl, Aldi und Co. sind Komponenten mit der Tuya-Firmware (siehe *Reingeschaut* auf Seite 80).

Während die in der *Make-Home-Assistent*-Reihe (Make 1/21 bis 5/21) benutzten Geräte lokal vom Smarthome-Server gesteuert werden, ist das bei der Firmware Tuya anders: Die wird Geräteherstellern zu einem relativ geringen Preis angeboten, vorausgesetzt, sie lassen die damit ausgestatteten Geräte nur über die Tuya-Cloud steuern. Da laufen also alle Befehle von der Smarthome-App über den Tuya-Cloudserver im Internet, der dann wiederum die Befehle an die Geräte im Smarthome sendet. Auch für Home Assistant gibt es eine Erweiterung (Integration), die das so macht.

Dagegen ist im Grunde nichts einzuwenden, wenn es da nicht kriminelle Leute im Netz

## Kurzinfo

- » Zugriffsschlüssel für Tuya-Geräte herausfinden
- » Lokale Tuya-Cloud in Home Assistant installieren
- » Tuya-Geräte vom Internet trennen

### Checkliste



**Zeitaufwand:**  
1 Stunde



**Kosten:**  
keine

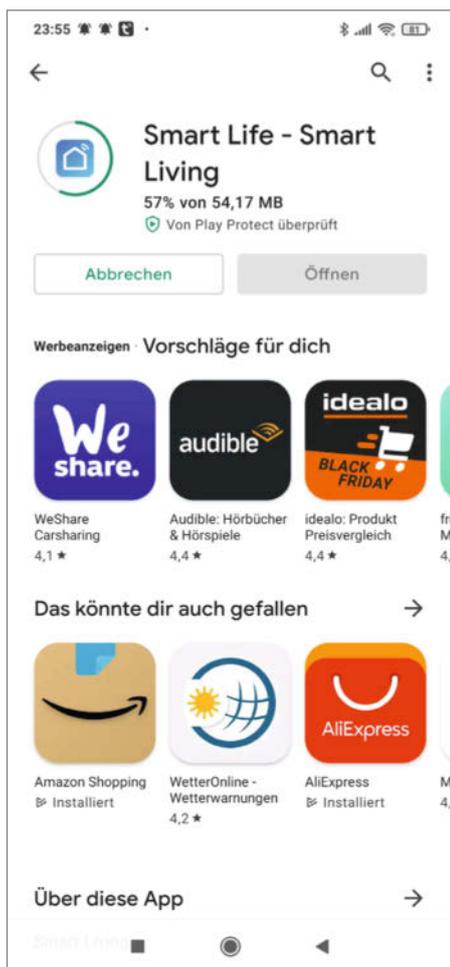
### Werkzeug

- » Smartphone, auf dem die Smartlife-App läuft
- » Computer (Windows oder Mac)
- » Internet-Verbindung nur zur Einrichtung

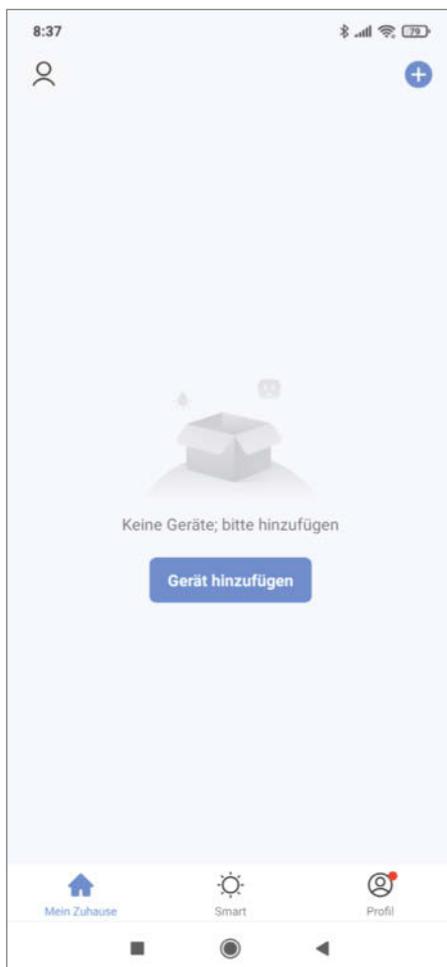
### Mehr zum Thema

- » Heinz Behling, Intelligentes Heim mit Home Assistant, kostenpflichtige Online-Version des Artikels aus Make 1/21, S. 100
- » Heinz Behling, Heizung unter Kontrolle, kostenpflichtige Online-Version des Artikels aus Make 2/21, S. 22

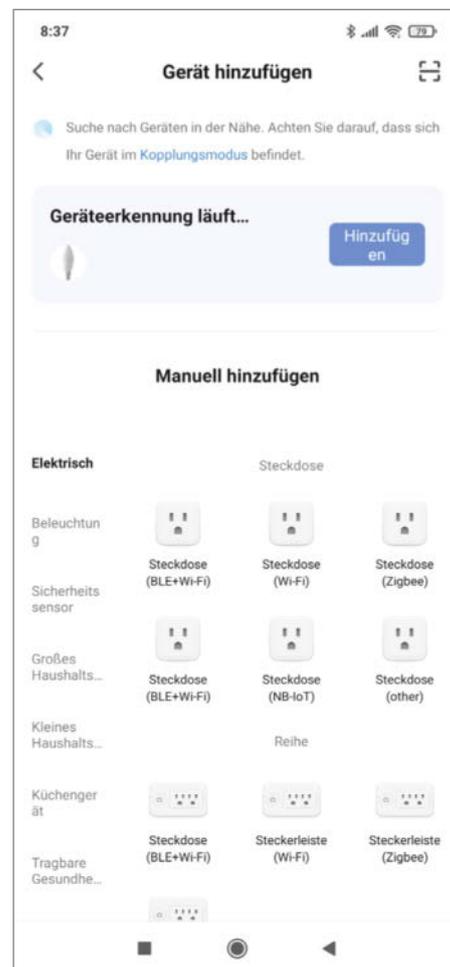
Alles zum Artikel im Web unter [make-magazin.de/xs37](https://make-magazin.de/xs37)



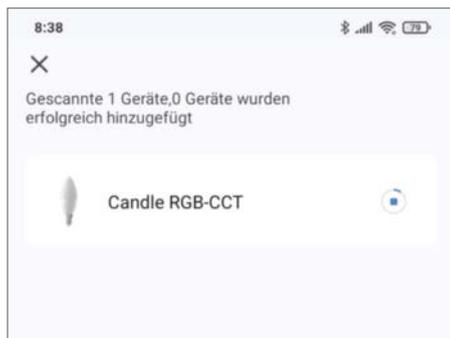
1 Achten Sie auf das Logo, es gibt mehrere Apps, die sich Smartlife nennen.



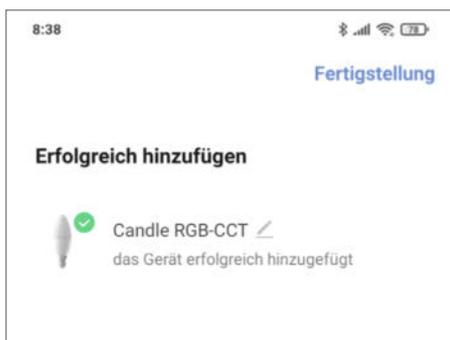
2 Die Gerätesuche startet nach Antippen des Pluszeichens.



3 Das erste Gerät wurde gefunden, der Scan läuft noch.



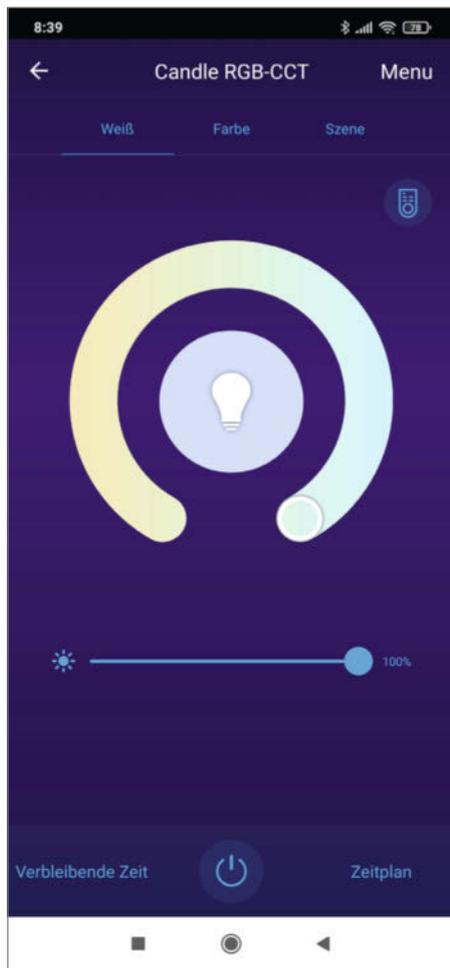
4 Birne gefunden, Lichterkette fehlt.



5 Sobald das grüne Häkchen erscheint, ist dieses Gerät mit der App gepaart.



7 Unbekannte Geräte können unter „Weitere“ gefunden werden.



6 Probieren Sie jeden Schalter und Schieber aus.



8 Die Konfiguration dauert etwa eine Minute.



9 Die Lichterkette wurde gefunden.

gäbe, die gerne einmal Server kapern und damit unter ihre Kontrolle bringen. Würde das bei einem Tuya-Server passieren, könnten Fremde in Ihrem Smarthome schalten und walten ... Anderer Fall: Was passiert, wenn Tuya diesen Server irgendwann abschalten würde? Dann wären Tuya-Komponenten wertlos, weil nicht mehr zu steuern.

Beides mag Ihnen im Moment vielleicht als geringes Risiko erscheinen, aber auf jeden Fall ist es ein unnötiges, denn man kann Tuya-Geräte von der Cloud lösen und mit Home Assistant voll lokal steuern. Das gilt übrigens nicht nur für die auf Seite 80 genannten Geräte von Action, sondern für nahezu alle mit der Tuya-Firmware arbeitenden Geräte.

### Letzte Internet-Verbindung

Einmal jedoch brauchen wir noch das Internet und die Smarthome-App. Um auf Tuya-Geräte zugreifen zu können, benötigen wir nämlich deren ID – sowie die dazugehörigen Schlüssel-Codes. Beides wird vom Cloudserver vergeben, sobald man mit der App ein Gerät dort anmeldet. Tuya macht es uns allerdings nicht ganz einfach, an diese Daten zu kommen: Aktuelle Apps lassen keinen Zugriff auf die im Smartphone gespeicherten Daten zu. Mit ein paar Tipps und Tricks, verfeinert mit der richtigen Software und ein wenig Zeit kommt man aber doch dran.

Doch zunächst zur App und gleichzeitig zum ersten Tipp: Auch wenn die Hersteller der Geräte gerne ihr eigenes Süppchen kochen und eine eigene App vorschreiben: Tuya-Geräte lassen sich auch mit der Original-App namens *Tuya* oder ihrem Fast-Zwilling *Smartlife* steuern. Fast-Zwilling, weil Smartlife den Zugriff auf die Codes ermöglicht. Deshalb installieren wir die Smartlife-App auf dem Smartphone. Die folgende Anleitung wurde für ein Android-Smartphone gemacht.

Als erstes suchen Sie im App-Shop nach der Smartlife-App und wählen *Installieren* 1.

Sorgen Sie dafür, dass Ihre Tuya-Geräte eingeschaltet sind. Falls Sie die Geräte schon einmal mit einer App benutzt haben, müssen Sie sie aus dieser früheren App entfernen (geht nur in der jeweiligen App) und in den Kopplungsmodus bringen. Wie das geht, steht in der Anleitung des Geräts. Meist muss dazu eine Taste länger gedrückt werden. Falls keine Taste vorhanden (LED-Birne), hilft oft dreimaliges Aus- und Einschalten. Die Birne beziehungsweise die Kontrollleuchte am Gerät muss dann blinken. Im Folgenden werden wir eine LED-Birne sowie eine Weihnachts-Lichterkette mit der App paaren.

Starten Sie die App und legen Sie ein neues Benutzerkonto an. Schließlich landen Sie beim App-Screen für die Gerätesuche 2.

Klicken Sie auf das blaue Pluszeichen. Der Geräte-Scan startet und die meisten Geräte

werden daraufhin von der neuen App automatisch gefunden und angezeigt 3.

Nach etwa ein bis zwei Minuten ist der Scan-Vorgang beendet und die erkannten Geräte werden untereinander angezeigt 4. Einige Geräte erkennt die App nicht automatisch, beispielsweise unsere Lichterkette. Daher erscheint sie nicht im Scan-Ergebnis.

Doch zunächst zur Birne: Kurze Zeit später erscheint ein grünes Häkchen neben dem Birnen-Symbol. Tippen Sie auf Fertigstellung 5.

Sie gelangen dann zum Bedienfenster für dieses Gerät und sollten testen, ob alles funktioniert 6.

Nun zur nicht gefundenen Lichterkette: In diesem Fall muss manuell gesucht werden. Dazu gehen Sie zurück zur Gerätesuche. Wählen Sie in der Liste auf der linken Seite die passende Kategorie (Beleuchtung oder Entertainment) aus. Daraufhin werden die der App bekannten Geräte als kleine Bilder angezeigt. Falls Sie das passende finden, tippen Sie darauf. Tipp: Gibt es in kein passendes Gerät, wählen Sie die Kategorie Weitere. Darin tippen Sie auf Sonstiges (Wi-Fi) 7.

Danach werden Ihnen zur Kontrolle die WLAN-Daten (SSID und Passwort) angezeigt und Sie müssen das Zurücksetzen des Geräts bestätigen. Bitte beachten: Tuya arbeitet nur mit 2,4GHz-Netzen, nicht mit 5GHz! Die App sucht nun nach unbekanntem Geräten und konfiguriert diese 8.

Kurze Zeit später bestätigt das grüne Häkchen die erfolgreiche Suche 9.

Tippen Sie auf Fertigstellung und das Bedienfeld erscheint zum Test.

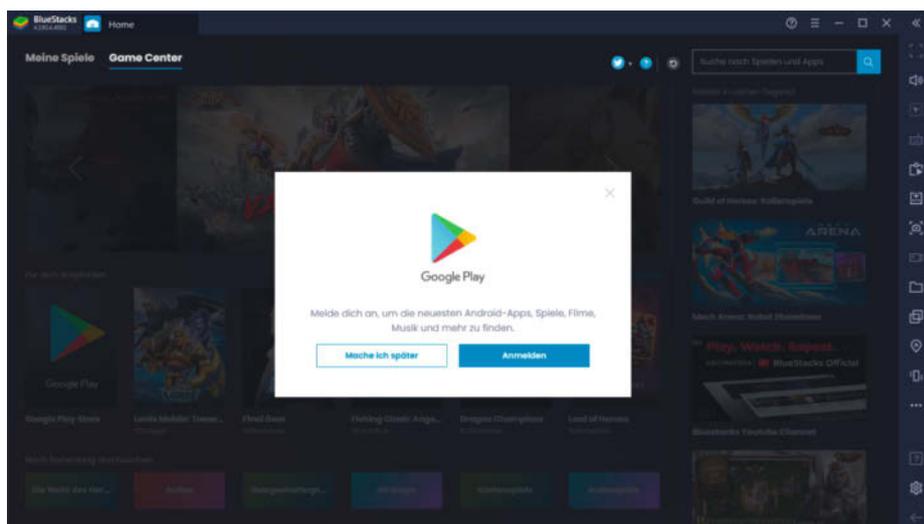
Wiederholen Sie diese Schritte für alle noch nicht mit der App gepaarten Tuya-Geräte. Sie können die App nun beenden. Weiter geht es am Computer.

### Trick mit Android-Emulator

Jetzt ist jedes Ihrer Tuya-Geräte bereits mit einem ID- sowie Zugriffs-Code ausgestattet. Nur kommen Sie auf dem Smartphone nicht so ohne weiteres daran. Da hilft der Android-Emulator BlueStack in der Version 4 (Achtung: Version 5 funktioniert nicht) für den PC (Windows oder Mac) weiter. Die Download-Adresse dafür und auch alle anderen erhalten Sie über den Kurzinfo-Link. Im Folgenden benutzen wir einen Windows-PC.

Überspielen Sie die passende BlueStack-Version und starten Sie die Installation. Das dauert einige Minuten. Danach erscheint der Emulator auf dem Computer-Bildschirm 10.

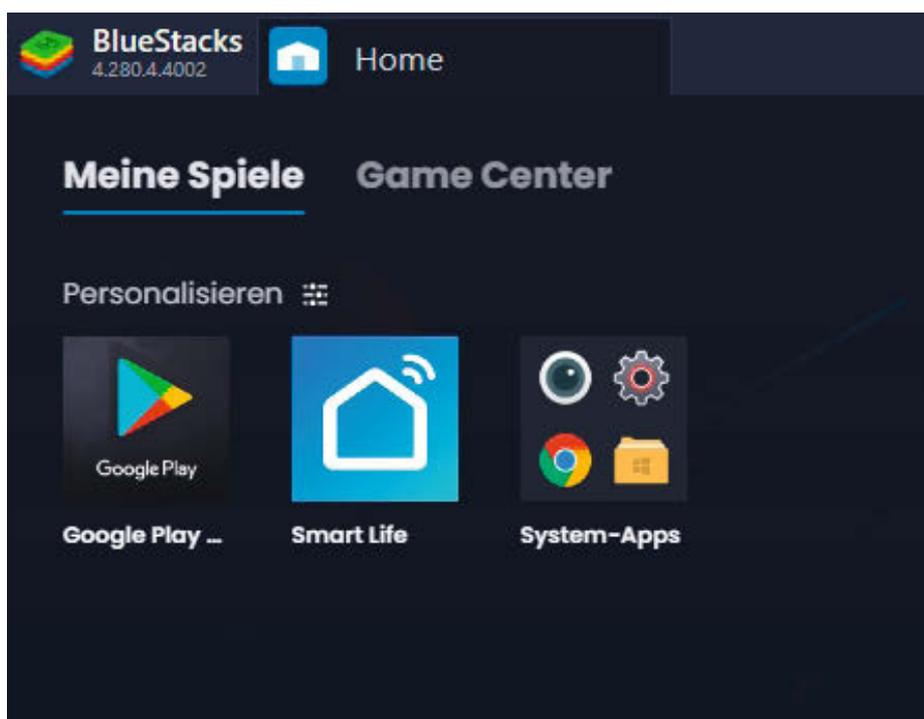
Sollten Sie im GameCenter landen, klicken Sie auf Mache ich später und Meine Spiele. In BlueStack können Sie nun Android-Apps installieren, also auch Smartlife. Der Clou dabei: Wir verwenden hier eine ältere Version (3.6.1) der App, die ihre Daten (also auch die Codes) im Klartext in einem Ordner auf der SD-Card



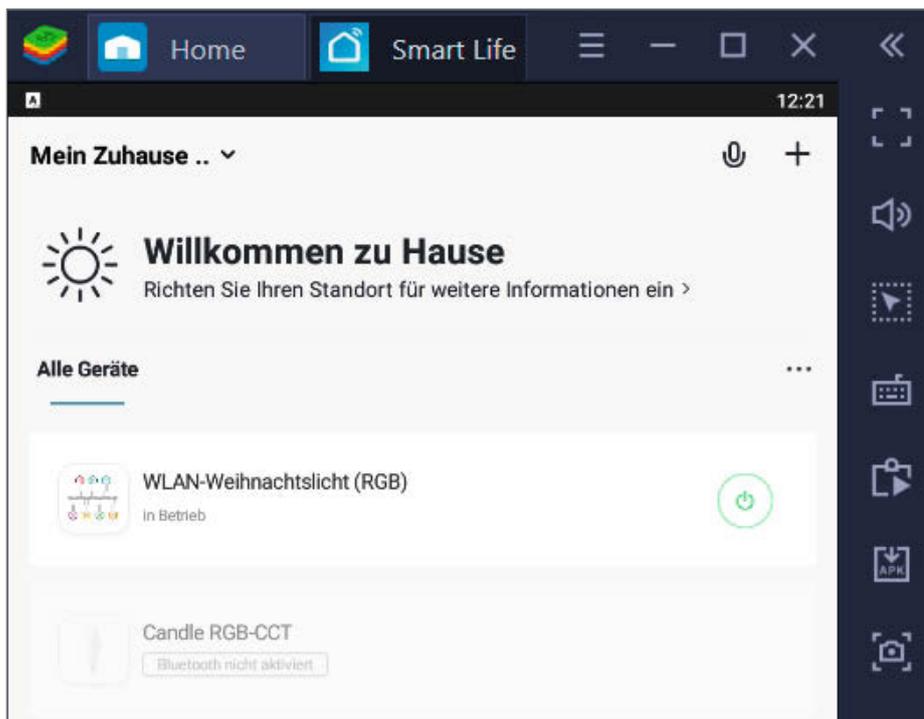
10 BlueStack startet.

Download			
This release may come in several variants. Consult our handy FAQ to see which download is right for you.			
Variante	Architektur	Minimum Version	Screen DPI
3.6.1 APK			
70	armeabi-v7a	Android 4.1+	nodpi
November 23, 2018 GMT+0100			

11 Achten Sie auf die Versionsnummer der Smartlife-App.



12 Die App ist einsatzbereit.



13 Einmal starten reicht, und schon sind die Codes auf der Festplatte.

des Smartphones speichern würde. Diese Speicherkarte wird auf der Festplatte des Computers als virtuelle Speicherkarte angelegt und kann mit geeigneter Software ausgelesen werden. Also kommen wir dort an die benötigten Codes. Allerdings ist diese App-Version nicht mehr im Google-Shop erhältlich, sondern an anderer Stelle als Download (siehe Kurzinfor-Link). Downloaden Sie die Datei auf Ihren Computer 11.

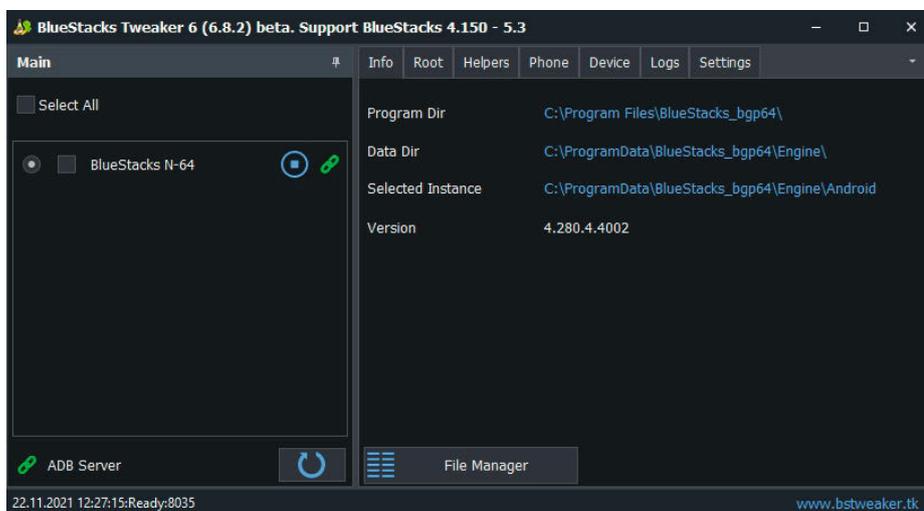
Die Installation in BlueStack ist einfach: Ziehen Sie mit der Maus die heruntergeladene Datei ins BlueStack-Fenster. Nach ein paar Minuten erscheint dann die App dort auf dem Desktop 12.

Starten Sie die App und melden Sie sich mit denselben Kontodaten an, die Sie auch in der Smartphone-App benutzt haben. Daraufhin erscheint die Liste mit den gepaarten Geräten 13.

Sie können die App nun beenden (kleines X oben im Smartlife-Reiter). BlueStack allerdings sollte noch weiter laufen.

### BlueStack Tweaker

Nun brauchen wir ein Programm, dass die virtuelle Speicherkarte des Emulators auslesen kann: *BlueStack Tweater*. Laden Sie die Zip-Datei auf Ihren Computer und entpacken Sie diese. Klicken Sie sich im entpackten Ordner



14 Der Tweaker ist einsatzbereit und hat Zugang zu den Emulator-Daten.

durch bis zur Datei *BlueStacksTweaker.exe*. Starten Sie die Datei per Doppelklick. Der Start dauert ein wenig. Sobald im Tweaker-Fenster unten links ein grünes Symbol vor *ADB Server* auftaucht, ist er einsatzbereit 14.

Klicken Sie auf *File Manager*, dann im nächsten Fenster auf */data* und in der Ordnerliste schließlich auf den Ordner *data*. In diesem Ordner suchen Sie den Folder *com.tuya.smartlife*. Öffnen Sie diesen Ordner und dann den Folder *shared\_prefs*. Jetzt erscheint die Datei, auf die es ankommt: *preferences\_global\_key-euxxxxxxxxxxxx.xml*. *xxxxxxxxxxxx* steht hier für einen längeren, individuellen Zahlencode. Wählen Sie im linken Teil des *File Managers* den *Desktop* aus, klicken Sie einmal auf die xml-Datei und kopieren Sie diese mit *F5 Copy* auf den Desktop.

Sie können sich die gerade kopierte Datei mal mit einem Editor ansehen. Die Suche und insbesondere das Finden der Codes ist aber etwas schwierig, da sie in sehr langen Textzeilen irgendwo versteckt sind. Aber auch dafür gibt es ein Programm: Auf der Github-Projektseite *TuyaKeyExtractor* von *MarkWatt-Tech* findet sich alles Notwendige, nämlich die Datei *MWT TuyaKeyExtractor.zip*. Laden Sie sie auf Ihren Computer und entpacken Sie diese.

Im entpackten Ordner starten Sie die Datei *TuyaKeyExtractor.exe*. (Falls sie nicht startet, müssen Sie zuvor noch *.NET Core 3.1 Runtime* installieren, siehe Kurzinfor-Link) 15.

Tippen Sie die *1* und die *Enter*-Taste, geben Sie dann den Pfad zur zuvor kopierten Code-Datei (*c:\Users\Username\Desktop*), für *Username* bitte Ihren Windows-Benutzernamen einsetzen) und direkt dahinter den Namen der Code-Datei ein. Danach zweimal *Enter* drücken und dann die *2* und nochmals *Enter* eingeben. Und schon sehen Sie die Daten, die Sie brauchen 16.

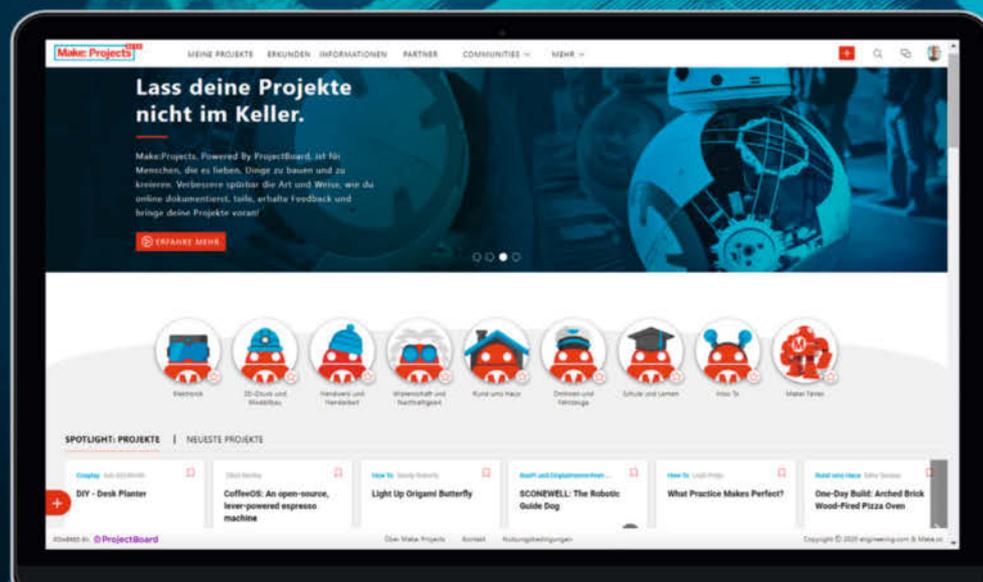
### Home Assistant und Local Tuya

Haben Sie sich die Code-Zeilen (drei pro Gerät) notiert oder in eine Datei kopiert? Dann können Sie Bluestack, den Tweaker und den KeyExtractor beenden. Jetzt geht es auf die Home-Assistant-Oberfläche. In Home Assistant muss *HACS* installiert sein (eine Anleitung dazu finden Sie im Artikel *Heizung unter Kontrolle* in *Make 2/21*). In *HACS* wählen Sie *Integrationen* und *Durchsuchen und Hinzufügen von Repositories*. Suchen Sie dann *Local Tuya* und installieren Sie die Version 3.2.3. Anschließend ist ein Home-Assistant-Neustart notwendig.

Dann geht es weiter mit *Einstellungen, Integrationen* und *Integration hinzufügen*. In der Integrations-Liste suchen Sie die *LocalTuya Integration* und installieren sie 17.

Anschließend zeigt Ihnen Home Assistant die ID des ersten Tuya-Geräts an 18.

# Make: Projects



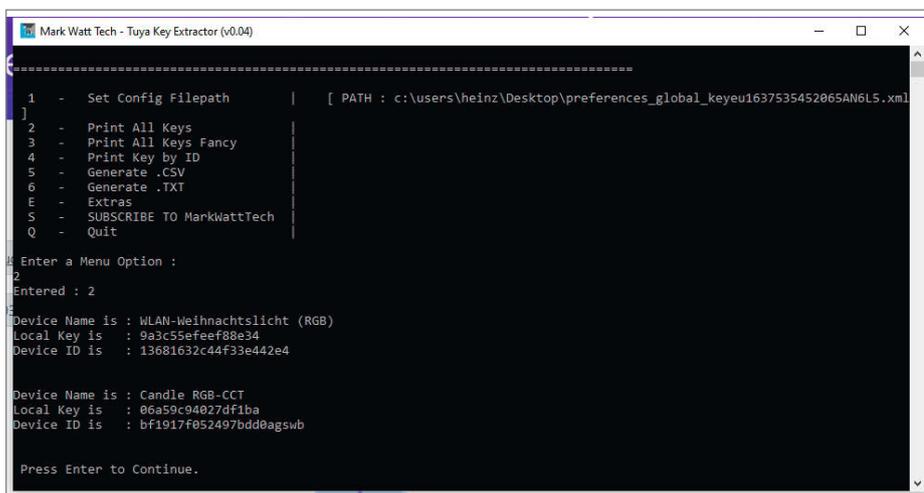
**Zusammen mehr machen –  
Teile deine Ideen, hol dir Inspiration,  
revolutioniere deine Projekte**

**... all das gemeinsam auf Make Projects!**

[www.makeprojects.com/de](http://www.makeprojects.com/de)



15 Im TuyaKeyExtractor müssen Sie zuerst den Pfad zur Code-Datei eingeben.



16 Die Codezeilen sollten Sie sich merken oder in eine Textdatei kopieren.



17 Ein Klick auf LocalTuya genügt zur Installation.

Notieren Sie sich die in Klammern stehende IP-Adresse. Klicken Sie auf *Absenden*. Im nächsten Fenster geben Sie den zur ID gehörenden *Local Key* ein, den Sie zuvor mit dem *KeyExtractor* ausgelesen haben. Auch den Geräte-Namen übernehmen Sie aus der *Key-Extractor*-Liste. Dann klicken Sie auf *Absenden*.

Wählen Sie dann den richtigen *Entity Type*, also zum Beispiel *light* für Lampen, gefolgt von erneutem *Absenden*.

Jetzt müssen Sie das *Settings*-Fenster ausfüllen. Zu den gängigen Tuya-Geräten gibt es dafür eine Liste auf Github (siehe Kurzinfolink). In diesem Beispiel für eine LED-RGB-Birne sind diese Werte richtig:

- ID: 20
- Friendly name: *Candle RGB-CCT*
- Brightness: 22
- Color Temperature: 23
- Color Mode: 21
- Color: 24

Dann klicken Sie in diesem und im nächsten Fenster auf *Absenden*. Anschließend können Sie noch den Bereich eingeben, in dem sich das Gerät befindet, zum Beispiel *Wohnzimmer*.

Jetzt erscheint die neue Integration in Home Assistant 19.

Das wiederholen Sie nun für alle weiteren Tuya-Geräte. Für jedes wird in Home Assistant eine eigene Integration angelegt. Danach können Sie die Bedienfelder in Home Assistant einsetzen. Wechseln Sie zum Beispiel zu Übersicht und Wohnzimmer. Klicken Sie auf *Karte hinzufügen* und wählen Sie für die LED-Birne die *Licht*-Karte. Als Entität geben Sie *light.candle.rgb.cct* ein. Nach dem Speichern sehen Sie die Karte 20.

Wenn Sie auf die drei Punkte oben rechts in der Karte klicken, gelangen Sie auch zu weiteren Einstellmöglichkeiten der Lampe. So können Sie beispielsweise die Farbtemperatur im Weiss-Betrieb zwischen kalt und warm stufenlos ändern. Klicken Sie auf *Colour*, dann gelangen Sie zum Auswahlfeld für die Leuchtfarbe 21, in dem Sie die Leuchtfarbe der RGB-LEDs ändern können.

### Abnabeln vom Internet

Zum Schluss müssen Sie noch Ihren Router bemühen: Die Tuya-Geräte haben im Moment noch Kontakt zur Cloud im Internet. Den müssen Sie im Router kappen. Bei einer Fritzbox beispielsweise müssen Sie für das Gerät, dass Sie anhand der zuvor notierten IP-Adresse erkennen, die *Kindersicherung* einschalten und dort die Internet-Verbindung sperren. Schauen Sie dazu im Handbuch Ihres Routers nach. Sie können die Tuya-Geräte danach übrigens immer noch per Smartphone steuern, allerdings nicht mehr in der ursprünglichen, sondern nur noch in der Home-Assistant-App.

Und nun viel Spaß mit den neuen cloud-freien Tuya-Geräten. —hgb



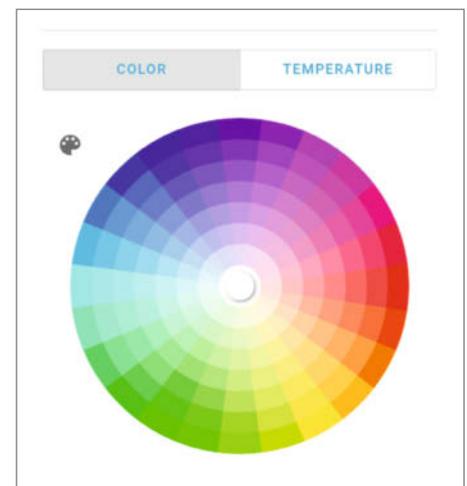
18 Merken Sie sich die in Klammern stehende IP-Adresse.



19 Es ist geschafft: Das erste Local-Tuya-Gerät in Home Assistant.



20 Das Bedienfeld der Lampe: Nach einem Klick auf die drei Punkte oben rechts können Sie auch die Farbe wählen.



21 Die Farbeinstellungen der Tuya-Lampe

Heft + PDF  
mit 29 % Rabatt



# Hoch hinaus...

Wie Cloud-Dienste die Arbeit erleichtern

Das c't-Sonderheft zeigt Ihnen Wege, wie Sie sich die Arbeit mit Servern aus der Cloud erleichtern und Neues ausprobieren können.

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

 [shop.heise.de/ct-cloud21](https://shop.heise.de/ct-cloud21)

# Winterlandschaft im Glas

Eine Schneekugel mit LED-Beleuchtung und vielen Details sorgt für winterliche und weihnachtliche Stimmung.

von Julia Pfundt



Die Motivation für dieses Projekt kam mir an einem Sommerabend. Kurz zuvor hatte ich die Glaskugel geschenkt bekommen, aus der ich ursprünglich ein schlichtes, dekoratives Glas mit Pflanzen oder Glassteinen gestalten wollte.

In der Hitze des Sommers sehnte ich mir die schönen Schneelandschaften und die märchenhaften Wälder aus den Bergen herbei. Eine schnelle Skizze reichte aus, um alle benötigten Materialien aufzulisten. Im Modellbauladen fand ich den Großteil von ihnen; Schneepulver, Bäume, den Bausatz für das Häuschen und so weiter. Viele Objekte habe ich auch selbst gebastelt, zum Beispiel den Zaun aus Zahnstochern und den Holzhaufen aus kleinen Ästen aus dem Garten. Die Eiszapfen wurden aus Heißkleber geformt, der während des Trocknens langgezogen und strukturiert wurde. Den Schneemann hat mein Bruder mit dem Computerprogramm *Blender* nach meinen Vorstellungen gestaltet; er wurde dann auf dem 3D-Drucker meines Onkels ausgedruckt. Mit verdünntem Fliesenkleber wurde die Schneelandschaft gegossen und die Tannen in einen dicken Schneemantel gehüllt.

Die Technik der LEDs war eine völlig neue Herausforderung für mich; ich wollte das Haus sowie die Tanne im Vordergrund beleuchten können, wenn möglich in mehreren Farben. Auch das Wählen der passenden Widerstände, damit die LEDs nicht zu grell wurden, war Neuland für mich. Daher holte ich mir fachmännische Hilfe und lernte, mit einem Lötkolben umzugehen. Das Ergebnis entsprach meinen Vorstellungen und durch einen Schalter im Inneren des Hauses (erreichbar durch das abnehmbare Dach) kann man die Weihnachtsbeleuchtung nach Belieben an- und ausschalten. Sie verleiht der Landschaft einen noch realistischer wirkenden Touch, und man könnte meinen, in eine echte Szene hineinzublicken.

Natürlich lief nicht immer alles reibungslos. Durch die langen Trocknungszeiten des Klebers und kurzen Perioden, in denen ich nicht an der Kugel gearbeitet habe, dauerte es eine lange Zeit, das Projekt fertigzustellen. Es war sehr schwierig, den Schnee reinweiß zu halten, da jede Verunreinigung darauf zu sehen war. Da es sich um ein Goldfischglas mit Boden handelt, musste ich stets vorsichtig und von oben arbeiten. Kleine Teile durften nicht verloren gehen und der Kunstschnee musste immer in zwei Schichten aufgetragen werden, um ein realistisches Ergebnis zu erzielen.

Alles in allem schätze ich die Ausgaben für Materialien auf 60 Euro und die Zeit auf etliche Stunden entspannten Bastelns. Wegen der Trocknungszeiten hat es mich manchmal gestört, nicht durchgehend arbeiten zu können und immer nur schrittweise voranzukommen. Aber die Geduld hat sich gelohnt und ich freue mich, meine Schneekugel zu Weihnachten wieder aufzustellen. Ein Projekt der außergewöhnlichen Art, auf das ich sehr stolz bin. —caw



Die Basis der Landschaft bilden Hartschaumplatten.



Die Landschaft ist fertig und mit Kunstschnee versehen.



Effektvolle Beleuchtung durch auf dünnen, isolierten Kupferdraht gelötete SMD-LEDs

# Maker baut Banksy-Schredder, nur mit Geld

Das Projekt des Makers Luis Marx visualisiert den Stand des eigenen Portfolios. Bei Gewinn fallen Scheine in den Kasten, bei Verlust werden sie geschreddert.

von Carsten Wartmann



**W**ie kann der Stand des eigenen Portfolios eindrücklich visualisiert werden, fragte sich der Maker und YouTuber *Luis Marx*. Kurven und Grafiken sind toll, aber auch abstrakt. Daher entwickelte Luis einen vom Künstler Banksy und seiner Aktion des selbstzerstörenden Bildes inspirierten Rahmen, der sich dem Stand des Portfolios folgend mit Geld füllt und bei Verlusten die Scheine auch gleich schreddert.

Der Rahmen wurde wie in seinem YouTube-Video zu sehen, komplett auf dem Balkon selbst geschreinert, da der Maker, wie so viele von uns, keine Werkstatt hat. Und man kann hier noch etwas lernen – etwa, dass für so ein Projekt auch ein Balkon und ein Küchentisch als Arbeitsplatz ausreichen. Holzarbeiten, Lackieren, Technik samt Schredder, Arduino und Raspi; der „Geldspender“ entsteht im 3D-Druck: klassisches Making eben.

In der Konstruktion ist zum einen ein Raspberry Pi Zero W verbaut, der über eine Börsen-API die Kurse des Portfolios oder einer Krypto-Währung abfragt. Die Kurse werden dann zu prozentualen Schwankungen umgerechnet und je nach Gewinn oder Verlust, wird Geld nachgefüllt oder zerschreddert. Im Code kann eingestellt werden, in welchen Zeitintervallen die Abfrage stattfinden soll.

Genau, der Code! Luis Marx ist angehender Wirtschaftsingenieur mit dem Schwerpunkt Maschinenbau und gibt freimütig zu, nicht besonders gut programmieren zu können. Auch sonst werden ein paar kreative Methoden benutzt, um zum Ziel (und das ist vor allem das Video) zu gelangen: Die Steuerung von Schredder und Geld-Nachfüllmechanismus übernimmt ein Arduino, weil der, mit 5V Spannung an den Pins, direkt Servos steuern kann. Die Kommunikation zwischen den beiden Mikroprozessoren findet einfach per GPIOs statt.

Als Schredder hat Luis einen kleinen Aktenvernichter benutzt. Der Schalter wurde mit einem 3D-Druck-Teil und einem Servo modifiziert, damit er vom Arduino steuerbar ist. Eine elegantere Lösung wäre gewesen, einfach ein Relais in den Stromkreis des Aktenvernichters einzulöten, gibt Luis zu: Aber er kenne sich mit dem Schaltkreis nicht genau aus und bei 230V Netzspannung löste er das lieber mechanisch.

Einzig offen bleibt, ob es sich um echtes Geld handelt oder ob das geschredderte Papier ebenso substanzlos ist wie der *fairy dust*, der durch die Börsen weht.

Wer Luis Marx noch nicht kennt, sollte einmal auf seinen Social-Media-Kanälen vorbeischauen. Luis ist immer witzig, hat die besten Mikrofonhalter des Webs und zwischendurch blitzt der eine oder andere Tipp auf, vor allem, wenn er seine CAD-Konstruktionen erklärt. Dies passiert aber eher in seinen YouTube-Videos, da hier mehr Zeit als etwa bei *TikTok* zur Verfügung steht, wo er ebenfalls aktiv ist. —caw

► [make-magazin.de/xajm](https://make-magazin.de/xajm)



Der Schredder, per Servo vom Arduino geschaltet



Für den Nachschub an Scheinen sorgt ein in CAD konstruierter, motorisierter Spender.



An der Börse Verlust gemacht und schon wird geschreddert.

# Insektoider Roboter spricht Basic

Laufroboter sind komplex. Hier stelle ich eine Lösung für einen 6-beinigen Roboter vor, der nur drei Servomotoren benötigt und dessen einfache Mechanik sich fast komplett mit einem 3D-Drucker herstellen lässt. Das Gehirn basiert auf dem Mikrocontroller Raspberry Pi Pico.

von Thomas Euler



Ein Lauroboter komplett selbst entwerfen und bauen – freie CAD-Programme, erschwingliche 3D-Drucker, billige Mikrocontroller und Servomotoren machen es möglich. Entstanden ist so der handteller-große *Robotling2*.

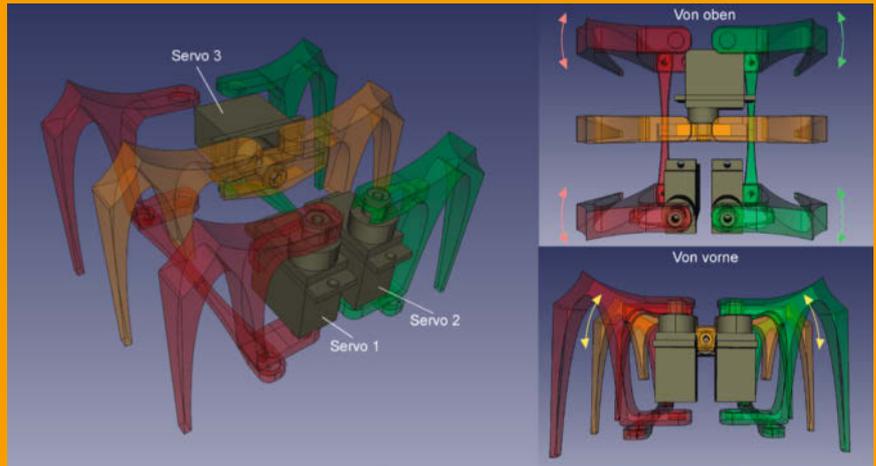
Inspiziert vom 20 Jahre alten Buch *Insectronics* von Karl Williams nutzt der Roboter ein sehr einfaches Prinzip zur Fortbewegung: Der erste Servo bewegt das rechte Vorder- und Hinterbein vor oder zurück; der zweite Servo tut dies auf der linken Seite. Der dritte Servo bewegt die mittleren Beine auf und ab. Um vorwärtszulaufen, kippt der dritte Servo den Roboter zunächst auf eine Seite, sodass nur das Vorder- und Hinterbein der anderen Seite den Boden berühren. Schwingen diese Beine nun nach hinten, bewegt sich der Roboter schräg nach vorne. Gleichzeitig drehen sich die in der Luft hängenden Vorder- und Hinterbeine nach vorne. Dann kippt der Roboter auf die andere Seite, und die nun auf dem Boden stehenden Beine bewegen sich zurück – es entsteht eine etwas watschelnde Vorwärtsbewegung. Um auf der Stelle zu drehen, muss man diese Sequenz nur leicht abändern; zum rückwärts Krabbeln kehrt man die Bewegungsrichtung der Beine einfach um.

Wegen dieser einfachen Mechanik benötigt das Design bis auf die 3D-gedruckten Teile nur die Servos, einen Akku und eine einfache Platine mit dem Pico-Controller. Kugellager an den Stellen, an denen sich Plastikteile gegeneinander bewegen, sind empfohlen, aber nicht notwendig. Die hier gezeigte Version besitzt außerdem ein kleines Display, auf dem der Zustand des Roboters angezeigt wird, sowie drei vorne montierte Abstandssensoren. Letztere sind schräg nach unten ausgerichtet und messen den Abstand zum Boden vor der Maschine. Durch diesen Trick können nicht nur Hindernisse erkannt werden (Distanz wird kleiner), sondern auch Tischkanten (Distanz wird sprunghaft größer).

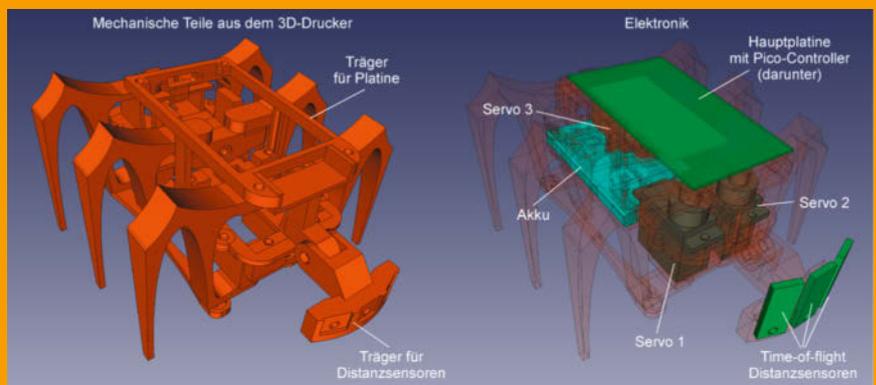
Gesteuert wird der Robotling2 von einem in *MMBasic* geschriebenen Programm. *MMBasic* wurde von Geoff Graham für Mikrocontroller entwickelt, ist modern und leistungsfähig. *MMBasic* erlaubt, direkt auf die Hardware des Pico zuzugreifen und unterstützt alle Pico-Anschlüsse, I<sup>2</sup>C, SPI, SD-Karten, viele TFT-Displays und Sensoren. Als Besonderheit bietet dieses BASIC einen eingebauten Editor mit Syntax-Highlighting – programmieren kann man also in einem Terminalprogramm direkt auf dem Pico. Obwohl der BASIC-Code interpretiert wird, läuft er sehr schnell. *MMBasic* bietet sich damit neben MicroPython und C(++) als weitere Programmiersprache für Mikrocontroller an – und bringt etwas vom unmittelbaren Programmiergefühl der 80er Jahre zurück.

—caw

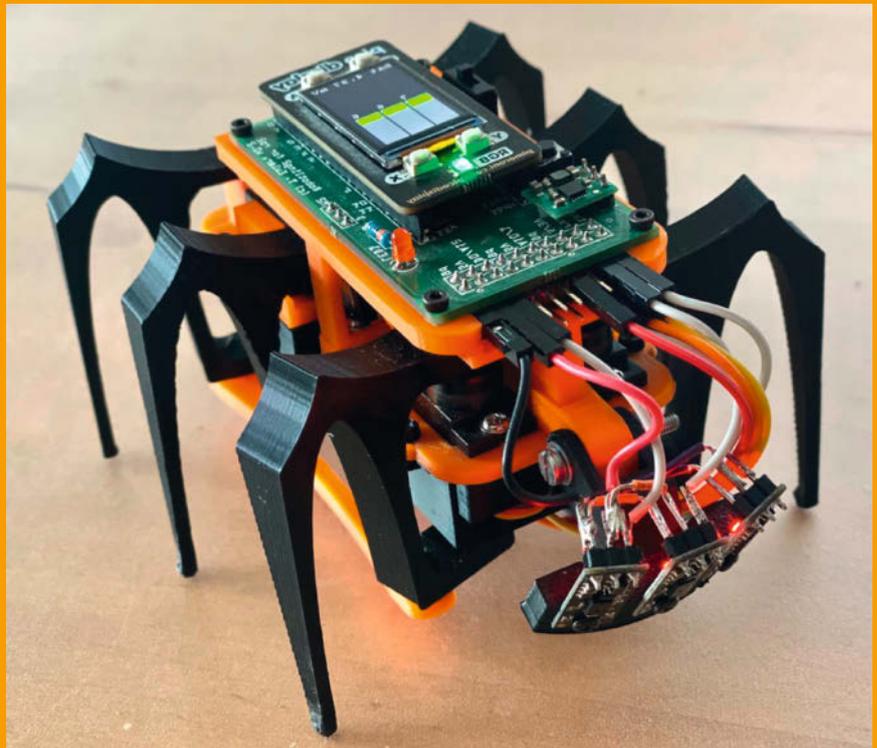
► [github.com/teuler/robotling2](https://github.com/teuler/robotling2)



Servo-Konfiguration für die Beine



Fast komplett 3D-gedruckt

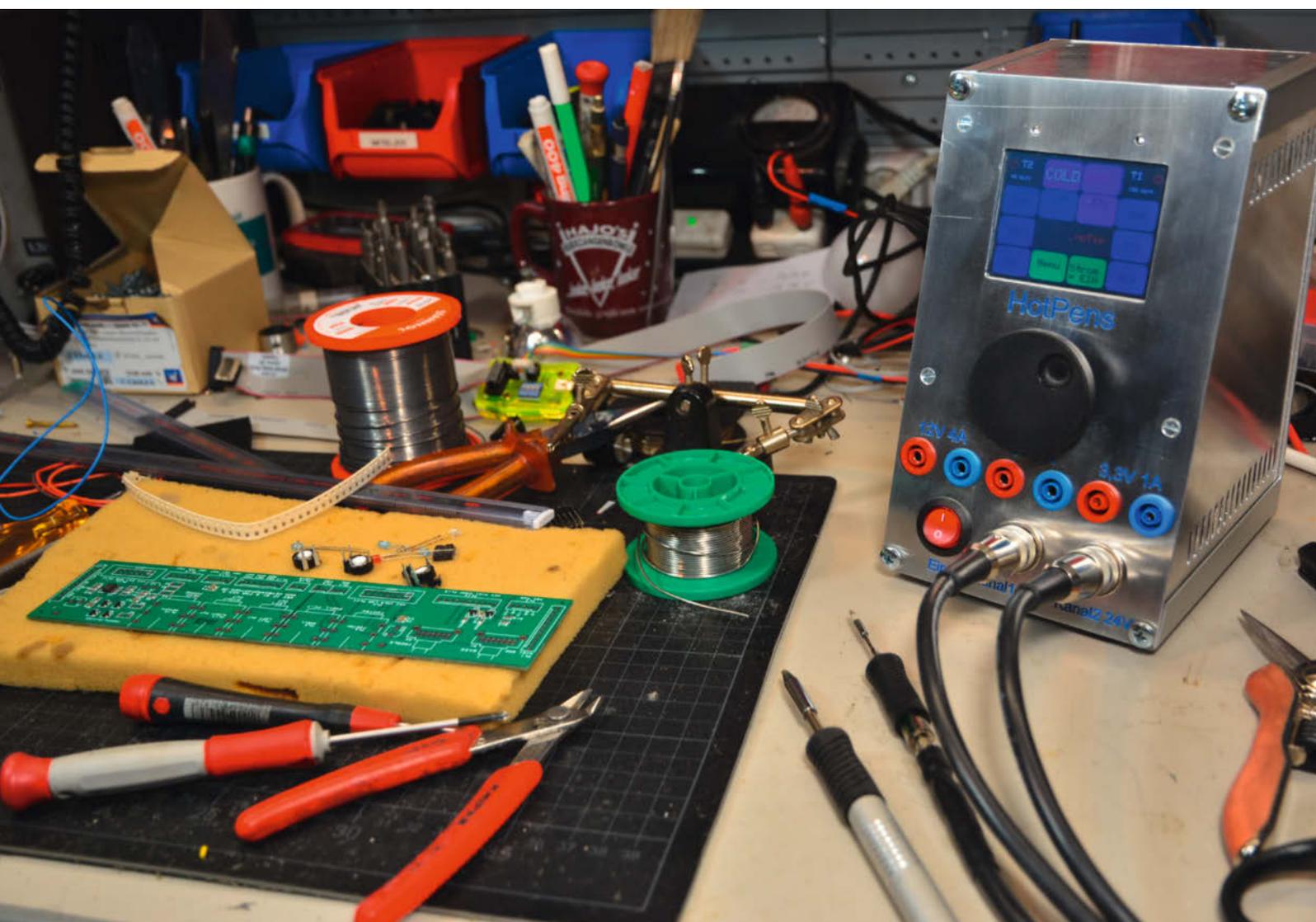


Robotling2 mit Display und ToF-Abstandssensoren (Time of Flight)

# HotPens: Duale DIY-Lötstation

Angeregt durch den Artikel „Lötstation – selbstgebaut“ in Make 5/2016 kam bald der Wunsch nach einer stärkeren LötKolbenspitze auf – 40W sind oft zu wenig. Unser neuer Entwurf befeuert gleichzeitig eine 40W- und eine 150W-Lötspitze von Weller und wird komfortabel über einen Touchscreen bedient.

von Michael Boretius und Hubert Schmucker



Schon vor fünf Jahren beschäftigten wir uns mit den Weller-RTM-Lötspitzen – die ja eigentlich schon halbe LötKolben sind, denn Heizelement und Temperatursensor sind bereits eingebaut. Wegen ihrer charakteristischen Form hat der Autor sie damals „Maiskolben“ getauft, und so hieß dann auch unsere Selbstbau-Lötstation. Allerdings war sie wegen der nur mäßigen Heizleistung eher für das SMD-Löten geeignet.

Neben den vornehmlich für SMD-Bestückung gedachten 40W-RTM-Spitzen gibt es von Weller aber auch die RTU-Serie mit 150W und dem gleichen Klinkenstecker-Anschluss, die mit 24V statt 12V betrieben werden. Weller bietet eine Dual-Lötstation für die beiden Lötspitzentypen an, diese kostet als Set über 1000€ und liegt damit außerhalb üblicher Maker-Budgets. Basierend auf diesen Eckdaten machten wir uns an die Arbeit, eine Version der Lötstation mit Touch-Bedienung und ebenfalls zwei Kanälen zu entwerfen. Das Projekt setzt einige Kenntnisse in Elektronik und Programmierung voraus – und nebenbei wollten wir noch erwähnen, dass man zum Bau einer Lötstation eine Lötstation braucht!

## Entwurfsfragen

Wir haben uns zur Versorgung für das 150W-Netzteilmodul Meanwell LRS-150-24 entschieden, da es recht preisgünstig ist, recht kleine Abmessungen hat und lüfterlos arbeitet. Heizen beide Lötspitzen zugleich auf, sollte das Netzteil jedoch mindestens 200W liefern, um Überlastreaktionen zu vermeiden; üblicherweise arbeitet man aber nur mit einem Kolben gleichzeitig. Für die 40W aus 12V setzen wir einen aufgelöteten DC/DC-Wandler ein, der die 24V verlustarm auf 12V herunterregelt. Dann benötigt die Schaltung noch 5V für den Arduino sowie 3,3V für den Touchscreen. Die 5V generieren wir mit einem DC/DC-Wandler, den es mit der vom 7805 bekannten Pinbelegung gibt. Dieser ist zwar etwas teurer als ein 7805, dafür verbrauchen wir aber keine Verluste beim Herunterregeln von 24V.

Der Bildschirm kann für die Signalsteuerung wahlweise mit 5V oder 3,3V arbeiten, jedoch will er für das Background-Licht unbedingt 3,3V. Weil man bei manchen Komponenten wie auch beim Bildschirm nicht sicher sein kann, dass es dieses Modell weiterhin gibt, nutzen wir die 3,3V vom Arduino zusammen mit einem Konverter-Baustein, um die Arduino-Signale auf 3,3V-Pegel zu übersetzen. Aber Achtung: Ein normaler Arduino Nano kann hierfür auch die erforderliche Leistung bereitstellen. Um den unterschiedlichen Nutzungswünschen Rechnung tragen zu können, bietet die Platine mittels Lötbrücken Konfigurationsmöglichkeiten, von denen wir drei herausstellen wollen:

Die RT-Lötspitzen von Weller bestehen aus der eigentlichen Spitze, einem Heizkörper mit Temperatursensor und einer Grifffläche mit 3,5mm-Klinkenstecker.

## Kurzinfo

- » Lötstation für zwei leistungsfähige LötKolben mit Weller-Spitzen
- » Temperaturmessung und Regelung
- » Mikrocontroller-Steuerung mit Touchscreen

### Checkliste

-  **Zeitaufwand:**  
10 Stunden
-  **Kosten:**  
80 bis 100 Euro (ohne Lötspitzen)
-  **Programmieren:**  
Arduino-IDE
-  **Löten:**  
Bedrahtete Bauteile bestücken
-  **Metallbearbeitung:**  
Gehäuse und Griffstücke anfertigen
-  **Bohren:**  
Standbohrmaschine

↓ Alles zum Artikel  
im Web unter  
[make-magazin.de/xz8w](http://make-magazin.de/xz8w)

### Material

- » Platine Hot-Pens (gefertigt nach Gerber-Daten, siehe Link)
- » Weller RTM- und RTU-Lötspitzen
- » Schaltnetzteil 24V/150W
- » Arduino Nano Every
- » 2,8"-Display ILI9341-kompatibel, z.B. TJCTM24028-SPI
- » Step-Down-Schaltregler 12V/4A Ausgang
- » Bauteile laut Stückliste (siehe Link)

### Werkzeug

- » Lötstation (!) oder FeinlötKolben
- » Bohrmaschine und Metallbohrer für die Gehäusebearbeitung
- » Lötspitzen-Thermometer zur Kalibrierung (optional, z.B. RST FG-100)

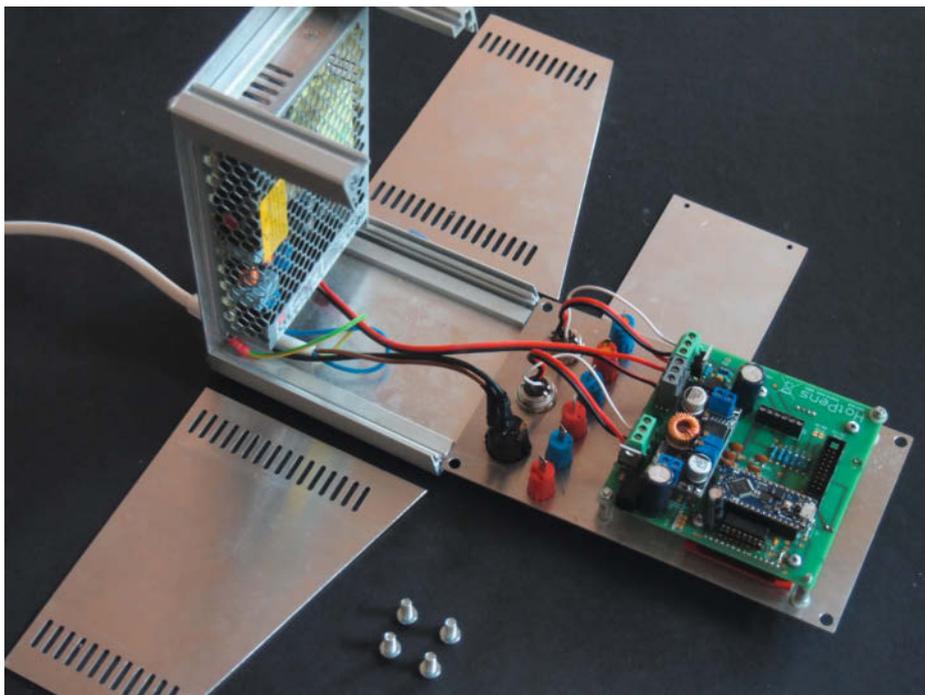
### Mehr zum Thema

- » Luca Zimmermann, Lötstation – selbstgebaut, Make 5/16, S. 12

1. Nur ein DC/DC-Wandler 12V: Der Leistungsteil erhält die 24 Volt und die daraus gewonnenen 12V vom DC/DC-Wandler. Der Arduino Nano Every (mit einem ATmega-4809) wird mit 12V gespeist. Die 5V für die Regelschaltung und den Bildschirm entnehmen wir, genauso wie die 3,3 Volt für den Bildschirm, dem Arduino Nano Every. Diese Konstellation hat den Nachteil, dass die Arbeitsspannung für die 40Watt Lötspitze gleichzeitig die Versorgung für den Arduino ist. Daher für die sichere Seite:
2. 12V nur für die Heizung: Zusätzlich löten wir den schon besagten 5V-DC/DC-Wandler ein. Dieser versorgt die Regelschaltung, den Arduino und den Bildschirm. Die 3,3 Volt kommen wie bei der Variante 1 vom

- Arduino Nano Every. Beide Varianten 1 und 2 haben den Nachteil, dass sie nur mit dem Arduino Nano Every funktionieren, denn für die 3,3V, die der Bildschirm mit der Hintergrund-LED benötigt, liefert der Standard-Nano nicht genügend Strom.
3. Einsatz des *Arduino Nano* (Standard). Die Schaltung nutzt die 12V-Schiene für 5V- und 3,3V-Linearregler. Somit ist der Arduino von der Versorgung mit diesen Span-





Die beschriftete Frontplatte und die Seitenteile mit den Lüftungsschlitzen haben wir bei einem Dienstleister fertigen lassen. Auf der Rückwand erkennt man das 150W-Netzteil.



Das Bestücken der Platine mit den Modulen und Steckverbindern sollte auch noch mit Ihrer alten Lötstation klappen.



In der Seitenansicht erkennt man die Sandwich-Bauweise aus Frontplatte, Display und Steuerungsplatine.

nungen entlastet. Nachteil: Die HotPens-Software läuft unverändert nicht auf dem Arduino Nano in der Standardausführung.

### Optionales

Das HotPens-Konzept mit vier verschiedenen Spannungen und einem Touch-Bildschirm bietet außer der Lötspitzenregelung noch Potential beispielsweise für eine Labornetzteil-Erweiterung. Dazu liegen die freien Arduino-Ports A0, A1, A4, A5, und D9, sowie D0 und D1 zusammen mit den drei Spannungen 12V, 5V und 3,3V auf einem Erweiterungsstecker. Eine aufsteckbare Netzteilplatine müsste die Komponenten enthalten, um den entnommenen Strom und zwei Spannungen am Display anzeigen zu können. Angedacht ist auch noch eine Spannungseinstellung mittels des Drehreglers – dies jedoch unter der Voraussetzung, dass im Arduino Nano Every noch genügend Programmspeicher zur Verfügung steht.

Die Schaltung besteht neben dem Arduino Nano im Wesentlichen aus Messverstärker, Leistungsschalter und Touchbildschirm mit Interface. Hinzu kommen neben den besprochenen Spannungsreglern die Anschaltung des Drehreglers sowie zwei an Jumpfern JP10 und JP 11 herausgeführte Signalleitungen D0 und D1, die im LötKolbenhalter verwendet werden könnten, um einen Ruhezustand zu detektieren, sowie JP1, der mit D4 verbunden ist und die Drucktaster-Funktion des Dreh-Encoders an das Programm weitergibt.

Die originale Weller-Lötstation erkennt einen Lötspitzenwechsel. Das ist vor allen Din-

gen wichtig, um die Regelparameter auf jede Lötspitze individuell abstimmen zu können. Weil dieses Zusammenspiel der Komponenten im Handstück nicht bekannt ist, muss HotPens sich auf die einfache Aufschaltung der Lötspitze mittels 12V/24V und Masse sowie der Temperaturmessleitung beschränken. Daher muss notgedrungen eine Ersatzlösung her, um eine korrekte Temperatureinstellung für verschiedene Lötspitzen zu erreichen; dazu später mehr.

### Regelschleife

In der Lötspitze befindet sich ein Thermoelement. Es erzeugt aus Wärme elektrische Energie, deren Spannung man messen kann. Eine Änderung der Temperatur ändert auch die Spannung. Da die Spannung zu klein ist (etwa 16µV pro °C Temperaturänderung), um sie direkt an einem analogen Eingang des Arduinos zu messen, benötigt man für jeden Kanal einen Messverstärker in Form eines OpAmps (Operationsverstärker).

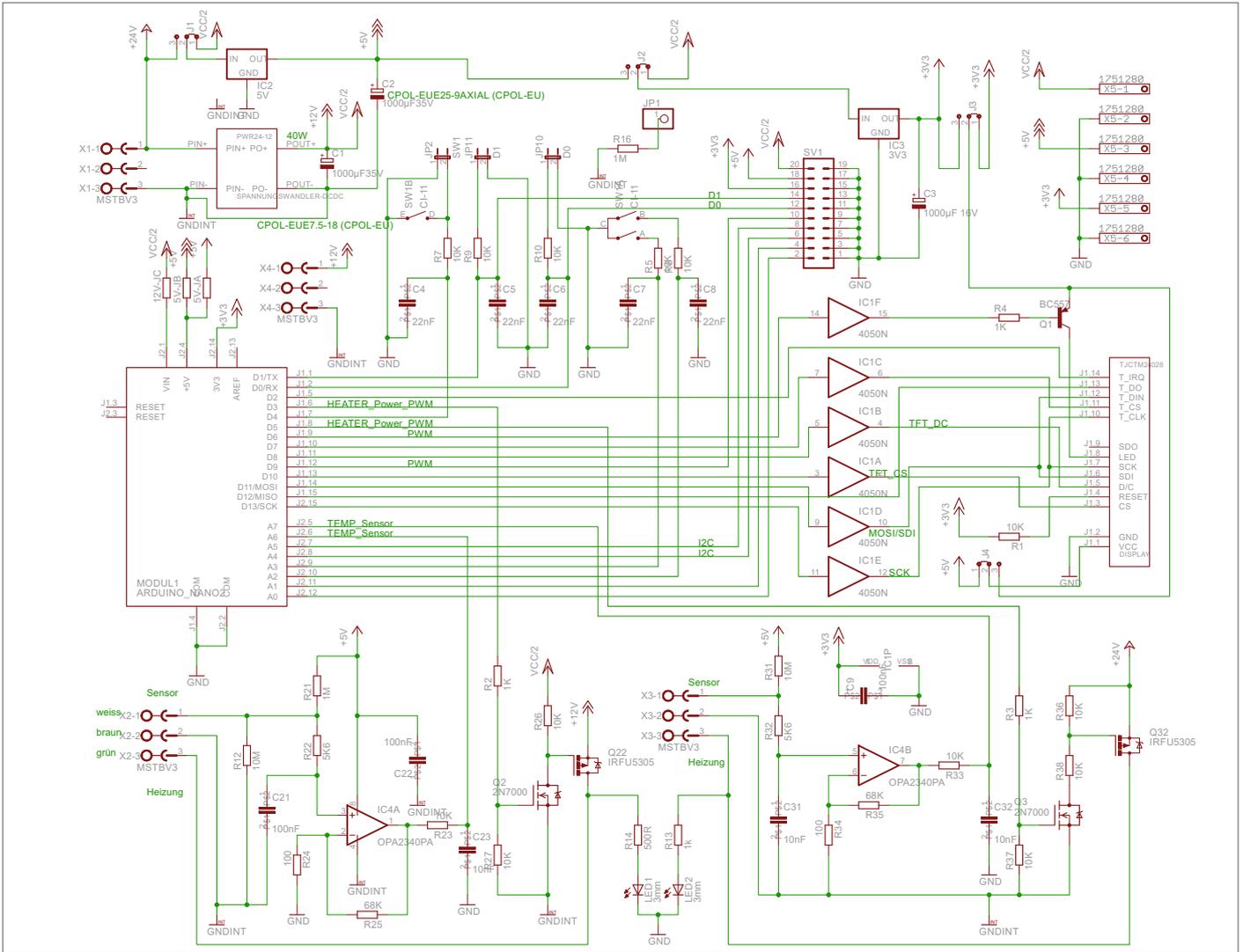
Auf der Suche nach einem geeigneten Zweifach-Operationsverstärker fiel die Wahl auf den noch erschwinglichen OPA2340, ein Rail-to-Rail-Typ mit geringem Offset und einer sehr guten Gleichspannungsunterdrückung bei unsymmetrischer Versorgung. Theoretisch geeignet wären auch die Typen NJM2732D oder MCP602-E/P. In Tests mit der HotPens-Schaltung lieferte der OPA2340 jedoch die besseren Ergebnisse im Hinblick auf das Signal-Rauschverhalten, und je geringer das Rauschen, desto weniger schwanken die Messwerte.

Bei Höchsttemperatur (ca. 450°C) liefert das Thermoelement 7mV. Über die Widerstände R4 und R7 wird ein Verstärkungsfaktor von  $68k\Omega/100\Omega = 680$  eingestellt. Damit ergibt sich eine Höchstspannung von  $680 \times 0,007V = 4,76V$  am analogen Anschluss des Arduino. Die Auflösung am Analog/Digital-Wandler beträgt 10 Bit, d.h. jeder Spannung von 0V bis 5V wird ein Integer-Wert zwischen 0 und 1023 zugeordnet. Mit den Formeln  $5V/1023 \times (0,00016V/680) = 0,45$  erhält man dann die Temperatur der Lötspitze:  $Temperatur = Messwert \times 0,45$  (plus ein etwaiger Offset durch den Gleichspannungsverstärker).

Die Leistungstreiber-Stufe mit dem P-Kanal-MOSFET IRFU5305 haben wir der Einfachheit halber vom „Maiskolben“ übernommen, jedenfalls für den 40W-Kanal. Beim 150W-Kanal sind 24 Volt zu schalten; hier wird die Schaltspannung am Gate über R36 auf 12V reduziert, ansonsten gibt es hier keine Änderungen.

### Heizungssteuerung

Vor der Einbindung eines zweiten Heizkanals in die Steuerung war es hilfreich und notwen-

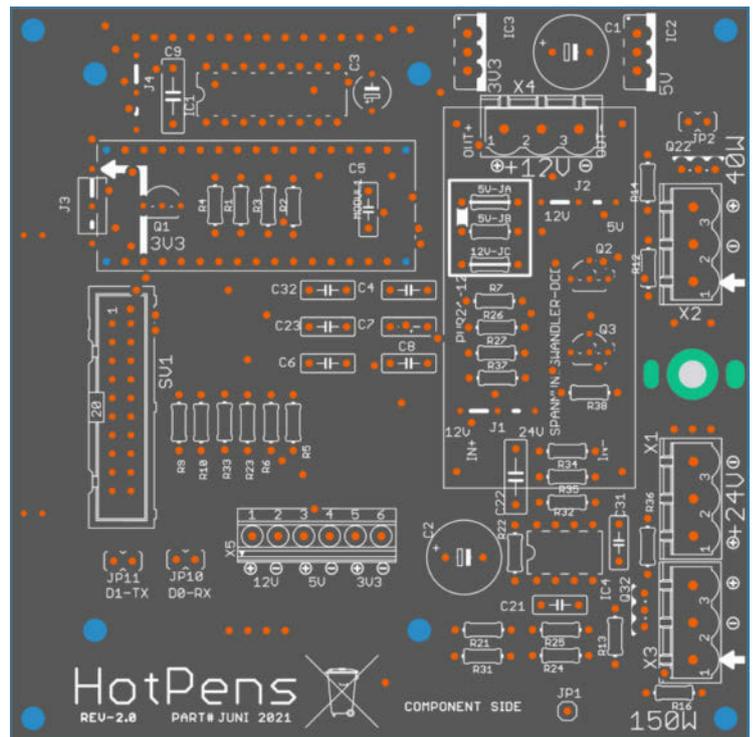


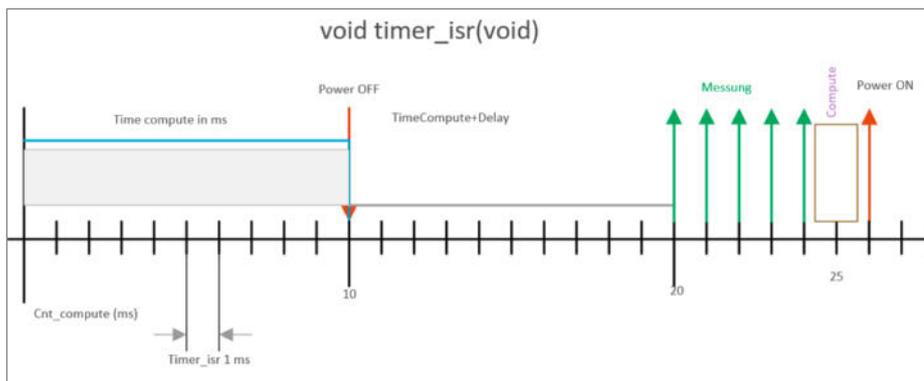
Sieht gefährlicher aus, als es ist: Der Schaltplan besteht größtenteils nur aus Verbindungen zwischen den Modulen, Steckverbindern und dem Arduino Nano Every.

dig, den Ablauf eines Zyklus' zu rekonstruieren. Der „Maiskolben“-Beschreibung konnten wir entnehmen, dass gleichzeitiges Heizen und Messen nicht möglich ist: Der durch den Heizstrom verursachte Spannungsabfall an den Leitungen würde die Messung extrem verfälschen, weil Heiz- und Messspannung die gleiche Masseleitung teilen.

Daher wird vor Messungsbeginn der Heizstrom abgeschaltet und kurze Zeit gewartet, bis auch die durch das Abschalten entstehenden induktiven Spannungsspitzen abgebaut sind, um dann mit einer Messreihe von fünf Messungen zu beginnen. Aus der Messreihe wird abschließend ein Mittelwert gebildet, um das Rauschen zu unterdrücken. Zum Verständnis der im Folgenden beschriebenen Routinen ist es erforderlich, dass Sie sich die (hier nicht abgedruckten) Listings aus unserem Github (siehe Link im Info-Kasten) bereithalten.

**Bestückungsplan für die Platine: Beachten Sie, dass die Steckverbinder für das Display und der Drehencoder auf der Rückseite montiert werden.**





Zum Messen der Spitzentemperatur über fünf gemittelte Messwerte wird die Heizung für einige Millisekunden abgeschaltet. Der Spannungsabfall an der gemeinsamen Masseleitung ist deshalb ohne Belang.

Nachdem die Messungen für jeden Kanal aufgenommen sind, berechnet der PID-Regler mit `compute` daraus den erforderlichen Heizwert für die Ausgabe als PWM (Pulsweitenmodulation). Dieser Wert geht dann für die nächste Heizphase an den PWM-Ausgang. Im oben gezeichneten Verlauf erkennen wir, dass `compute` zuerst den PWM-Wert für die 12V bildet und danach den PWM-Wert für die 24V-Schiene, bevor für beide Kanäle die Leistung aufgeschaltet wird.

Der für die Heizungsregelung interessante Teil ist `compute...()`. Der darin enthaltene Aufruf `heaterPID.compute()` bekommt seine Variablen als Array, wobei `set_td` der Sollwert ist, d.h. die gewünschte Spitzentemperatur, dann `cur_td` als aktuell gemessener Temperaturwert. `pid_val` ist der erzeugte

Korrekturwert, der als `pwm` an den Port zu geben ist. Zusätzlich enthält das Array noch die Regelkonstanten: P (proportional), I (integral), und D (differential). Wem das Regelverhalten mit den fest eingestellten Werten nicht gefällt, bekommt hier eine sehr schöne Spielwiese. Vorsicht jedoch bei unbedarften Änderungen: Die Heizung der Lötspitze ist schneller durchgebrannt als man glaubt. Dabei ist dem Verfasser schon die eine oder andere teure Spitze unwiederbringlich zum Opfer gefallen.

### Programmatisch

In der Hauptschleife beschäftigt sich unser Programm regelmäßig mit einigen wichtigen Dingen:

- Prüfen, ob ein Touch-Interrupt eingetrudelt ist: `if (ts.tirqTouched())...`
- Prüfen, ob der Drehregler bewegt worden ist: `if (newValue != lastLoopDisplayedRotaryCount)...`
- Prüfen, ob der Kalibriermode aktiviert wurde: `if (Kal12V || Kal24V)`
- Aktualisierung der Bildschirmausgabe: `if (!UMenu && !Kal12V && !Kal24V) { display() ...};`
- Regelung des Stand-By-Modus: `AutoPower();...`

Hier liegen übrigens auch die Ausgaben für verschiedene Werte, wenn man per `ifdef` in den Testmodus geschaltet hat. Diese `serial.print`-Ausgaben kann das Entwicklungssystem VisualMicro (ein Plug-in für den sehr empfehlenswerten Editor Microsoft Visual Studio Code) in einem separaten Fenster darstellen oder in ein Logfile schreiben.

Die Touch-Service-Routine 1 setzt die Variablen `Touch_Zeile` und `Touch_Spalte` anhand der Funktionen `Zeile` und `Spalte`. Hier standen wir vor der Überlegung, wie exakt die Auflösung des Berührungsrasters auf andere Bildschirme reproduzierbar ist. Um den Umstand einer Bildschirmpkalibrierung zu umgehen, teilen wir den Bildschirm in ein Berührungsraster von 4 Zeilen und 4 Spalten ein. Mit den so gebildeten 16 Feldern kann man eigentlich ganz gut arbeiten. Sollten sich bei Verwendung eines neuen Bildschirms Abweichungen ergeben, sind an dieser Stelle die Korrekturen zu hinterlegen. Auch das zuletzt berührte Feld und die Aufrufe der entsprechenden Bildschirmseiten je nach gesetzten Flags sind hier enthalten. Die `while`-Schleife verhindert das „Kontaktprellen“, das auch Touchscreens aufweisen.

Die Bibliotheksroutine für den Drehregler heißt `qdec.h`, sie wird wie die anderen auch im `setup`-Teil mit `qdec.begin()` und in `definitions.h` mit `::SimpleHacks::initialisiert`. Hier übergeben wir auch die Portnummern der angeschlossenen Schalter. Die zugehörige Interrupt-Service-Routine erkennt, ob links- oder rechtsherum gedreht worden ist und setzt alle im Programm notwendigen Variablen. So erfolgt die Begrenzung der angezeigten Werte beim Hoch- und Herunterzählen während der Drehbewegungen.

Der Drehregler arbeitet mit zwei versetzt betätigten Schaltern. Das bewirkt, dass die durch Drehbewegung ausgelösten Impulse eine Phasenverschiebung erzeugen und somit die Drehrichtung erkennbar ist. Der zugehörige Interrupt `ISR_FOR_QDEC` liefert zur Weiterverwendung die Variable `rotary-Count`.

Die Bildschirmausgabe ist sozusagen ein stetiger Refresh der einzelnen adressierbaren Pixel. Nützlicherweise sprechen wir hierfür mit der Bibliotheksfunktion `Adafruit_ILI9341`, die wir im `definitions.h`-Bereich auf mit `tft` initialisiert haben.

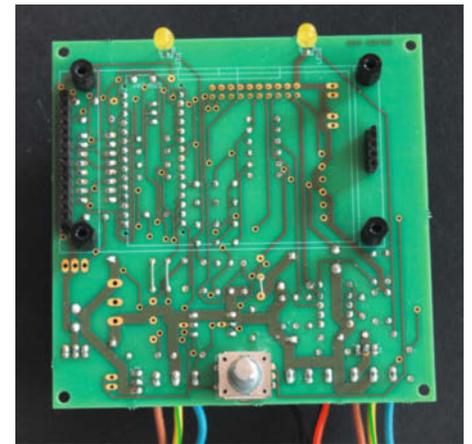
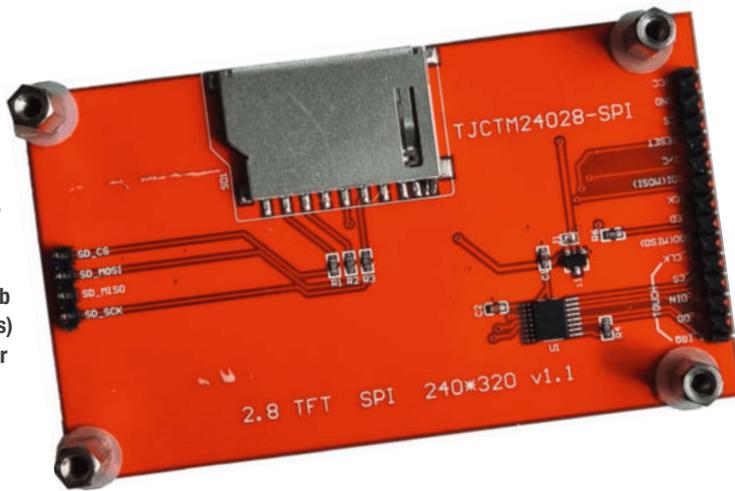


Der Hauptbildschirm bietet für beide Spitzen vorgegebene Temperaturen auf „Knopdruck“, weitere Einstellungen erledigt man unter „Menü“. Der „Strom“-Button ist für die Netzteil-Erweiterung vorgesehen.

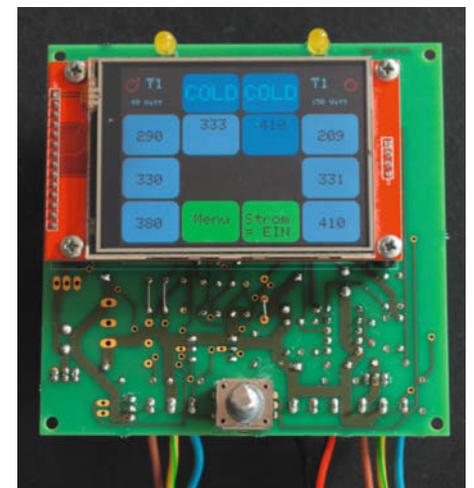
Zur genauen Kalibrierung benötigt man ein Lötspitzen-Temperaturmessgerät wie dieses. Überschlüssig geht es auch, wenn man versucht, mit der heißen Spitze Lötzinn mit bekanntem Schmelzpunkt zu erweichen.



**Wichtig:** Für die verwendeten 5V-Pegel muss die Brücke J1 am Display (oberhalb des SMD-Transistors) geöffnet werden. Der SD-Kartenslot wird nicht verwendet.



Die Rückseite der Platine trägt die Anschlüsse für das LCD und den Drehgeber ...



... und ist damit eigentlich die Oberseite.

Den Standby-Modus konnten wir so vom „Maiskolben“ übernehmen, er bietet aber noch Potential für Verbesserungen. Das macht sich spätestens nach dem Rücksprung aus dem Untermenü bemerkbar – oder wenn der Standby manuell geschaltet wird, in dem der Count-Down-Zähler nicht mehr läuft. Ebenso bietet der Wake-Up aus dem Standby noch genügend Raum für Nachjustierungen, denn nicht immer springt die Heizung wieder an, bzw. resettet den Count-Down-Zähler.

Zur Aktivierung des Kalibriermodus arbeitet die Hauptschleife zunächst mit den gelöschten Menüflags UMenu, Ka12V und Ka124V und steuert die Heizung, die Bildschirmanzeige und den Stand-By-Mode. Der Druck auf den grünen Button „Menu“ bewirkt das Setzen des Flags UMenu=true, somit ruft die Hauptschleife die Unterfunktion() auf.

## Setup-Prozess

Der Programmstart zeigt zuerst den Meldebildschirm mit der Programmversion. Die Abfrage auf `FFFFh = true` zeigt dem Programm, dass die default Parameter zu laden sind. Wenn false, dann wird dieser Init-Teil übersprungen. Bei Bedarf lassen sich die Init-Parameter erneut laden, indem während des Einschaltens der Drehregler-Knopf eingedrückt bleibt. Nach Durchlauf der Initialisierungsroutine bzw. Laden der gespeicherten Arbeitswerte:

- drei voreingestellte Temperaturwerte
- die aktuell gewählte Solltemperatur,
- Kalttemperatur aus der Kalibrierung für je vier Lötspitzen,
- Warmtemperatur aus der Kalibrierung für je vier Lötspitzen

erfolgt der Aufbau des Hauptbildschirms. HotPens in der Version 2.1 (der Melde-String liegt in `definitions.h`) ist erstellt für den Arduino-Nano Every; der Programm-Speicher des Arduino Nano Standard ist leider zu knapp. In der aktuellen Fassung sind 87% des Speichers des Every verwendet. Leider kann man daraus nicht schließen, dass noch 13% zur freien Verfügung steht. Beim Hochladen meldet sich das Entwicklungssystem (MS Visual-Micro) auch nicht, um zu sagen „halt- bis hier und nicht weiter“, sondern es schießt den Code ein,

der dann schlicht und ergreifend nicht läuft. Folgende Bibliotheken sind einzubinden:

- `EEPROM.h` hiermit spricht das Programm den internen Arduino-Nano EEPROM-Bereich an. Es speichert dort die oben beschriebenen Arbeitsparameter
- `Qdec.h` Die Serviceroutine für den Drehregler
- `PID_v1.h` Die Serviceroutine für den PID-Regler
- `EveryTimerB.h` Die Serviceroutine für die Timerfunktion
- `Adafruit_ILI9341.h` hierüber laufen die Bildschirmausgaben
- `XPT2046_Touchscreen.h` Die Serviceroutine für den Touchscreen

## Platine und Display

Unsere Platine hat die Abmessungen 100 x 100mm, sie wird beidseitig bestückt, aber nur mit bedrahteten Bauelementen. Aus Platzgründen sind unterhalb des Arduino Nano noch einige Bauteile platziert. Es ist sicher kein Fehler, den Arduino zu sockeln. Ebenso erhöht steht der 24V-12V DC/DC-Wandler auf verlängerten Lötdrähten, um den untenliegenden Bauteilen Platz zu schaffen. Eine Kühlung der Komponenten ist nicht nötig, trotzdem sollte das Gehäuse eine Luftzirkulation ermöglichen.

## Kabel und Griffstück

Ein Original-Kabel für die RT-Spitzen kostet bei Weller sportliche 168€ - zugegeben mit Stecker und einem schicken Griff, der prima in der Hand liegt. Es galt also, nach preiswerten Alternativen Ausschau zu halten. Nach längerem Suchen haben wir ein geeignetes hochflexibles Kabel gefunden, dessen Ummantelung bis 180°C hitzebeständig ist und vom Außendurchmesser zu den verwendeten Gehäusesteckern passt, das Ölflex HEAT 180 EWKF 3G0,75 (Leiter mit 0,75mm<sup>2</sup>) für rund 2€ pro Meter. In der gleichen Preislage liegt das etwas dünnere HELU SiHF-3x05-qmm mit kleinerem Leiterquerschnitt (0,5mm<sup>2</sup>), das für den 40W-Kolben ausreichend wäre.

Um Verwechslungen bzw. Fehlkopplungen zu vermeiden, haben wir für den 40W-Kanal

einen dreipoligen und für den 150W-Kanal einen vierpoligen Gehäusesteckverbinder gewählt – den klassischen „Kleintuchel“-DIN-Stecker mit Schraubverriegelung. Abschließend empfiehlt es sich, die Lötstellen an der Handgriffseite mit Schrumpfschlauch zu stabilisieren, wobei die Verwendung unterschiedlicher Farben empfehlenswert ist.

Unschönes Detail am „Maiskolben“ war das fehlende Handstück – man hielt praktisch nur die Lötspitze mit ihrer Grifffläche in der Hand. Wer eine Standbohrmaschine mit Maschinenschraubstock oder noch besser eine Drehmaschine besitzt, kann sich aus Alu-Rohr passende Griffstücke anfertigen. Das abgebildete für die 150W-Spitze besteht aus einem 75mm langen 12mm-Alurohr aus dem Baumarkt, für die 40W-Spitze sind 10mm ausreichend. Der Innendurchmesser muss (am besten mit einer Reibahle) in der Tiefe der Spitzen-Aufnahme auf exakt 9 bzw. 10mm aufgeweitet werden. Mit etwas Geschick passt dann auch die Klinkenbuchse in die Hülse. Zur Zugentlastung haben wir sie mit Epoxidharz-Kleber eingeklebt. Das



Für den Selbstbau der Griffhülse benötigt man eine Ständerbohrmaschine oder besser noch eine Drehmaschine zum Aufweiten des Alu-Rohrs.



Verschiedene Stecker (DIN-Kleintuchel) beugen einer Verwechslung vor: Der vierpolige gehört zum 150W-Kolben.



Die Kalibrierung führt über mehrere Bildschirmseiten und ist für beide Lötkolben auszuführen.

Ende des Röhrchens zierte eine Kabeltülle (mit Sekundenkleber am Röhrchen befestigen).

Auf die Gestaltung und den Bau des Gehäuses möchten wir nicht näher eingehen. Die Abbildung unten zeigt die nach dem Bohrplan im Github-Repo (siehe Link im Info-Kasten) gefertigte Frontplatte. Das Gehäuse beherbergt außer der frontseitig angebrachten Platine mit den Bedien- und Kontaktelementen nur noch das Netzteil. Aus Platzgründen haben wir das vorliegende Baumuster hochkant, mit einer leicht nach hinten geneigten Frontplatte gestaltet. Den Bohrplan finden Sie (wie auch die anderen Daten

zum Projekt) online unter dem Link im Info-Kasten.

### Kalibrierung

Wie bereits oben beschrieben, durchläuft die Setup-Routine nach den Interrupt-Initialisierungen eine Abfrage, ob sie auf einem frisch programmierten Arduino läuft, bevor die Hauptschleife zur Ausführung gelangt. Bereits jetzt lassen sich die je drei Temperaturwahl-Buttons mittels Drehregler einstellen. Das Aktivieren der gewünschten Schaltfläche erfolgt mit kurzer Berührung. Die Schaltfläche wechselt daraufhin die Farbe auf dunkelblau

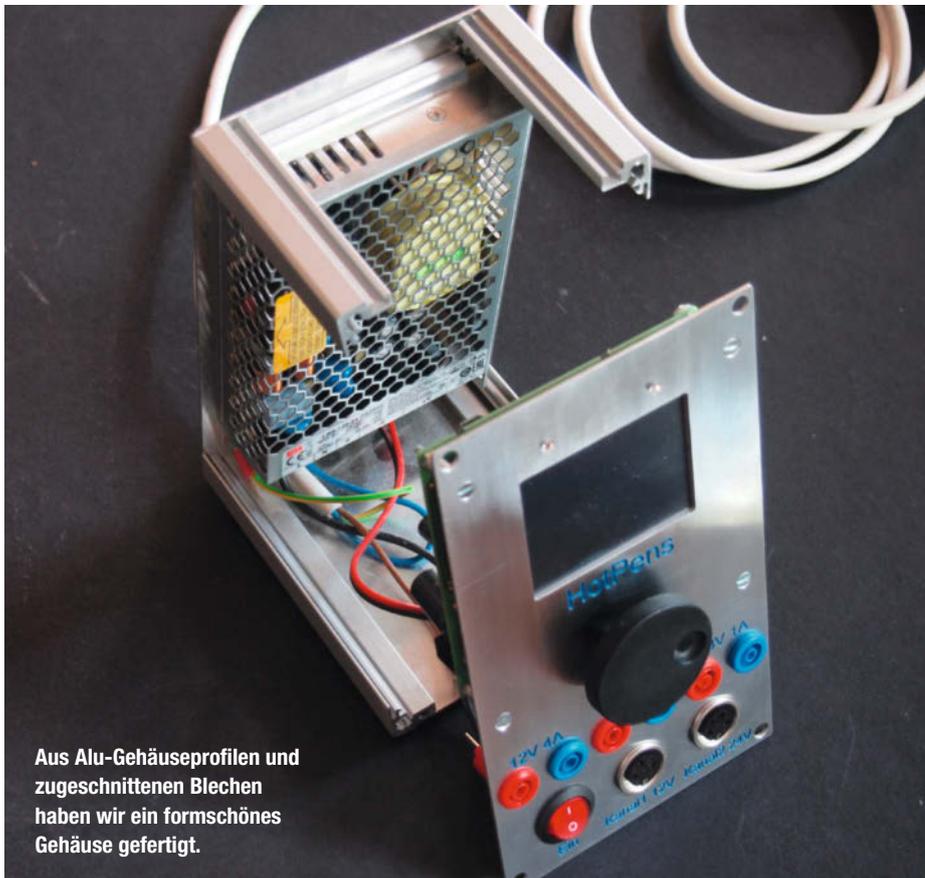
und zeigt somit an, wie sich der dargestellte Temperaturwert beim Drehen des Reglerades verändert. Das Abspeichern der Werte erledigt das Programm ohne gesonderte Aufforderung. Diese vorgewählten Temperaturwerte gelten unabhängig von der eingesetzten Lötspitze, im Unterschied zu den Kalibrierwerten.

Die „Kalt“-Einstellung bildet den `adc_offset`-Wert (geringste durch das Thermoelement messbare Temperatur). Hierzu misst man die Lötspitzen-Temperatur mit einem geeigneten Thermometer (siehe Link im Info-Kasten), und zwar nachdem das Programm durch Heizen einen ersten Temperaturanstieg festgestellt hat.

Der Vorgang beginnt damit, dass das Programm den Kaltwert bestimmt („Ist“) und den Sollwert schrittweise inkrementiert. Hat der Sollwert den gemessenen „Ist“-Wert erreicht, endet der Inkrement und die Regelung pendelt sich entsprechend ein. Messen Sie nun die Spitzentemperatur mit einem Thermometer und stellen den ermittelten Wert mittels Drehregler in der grünen Anzeige ein.

In der „Warm“-Einstellung drücken Sie den Temperatur-Button, dessen Wert sich aus dem im Hauptbildschirm als Arbeitswert hinterlegten Wert ergibt. Nach der Aufheizphase versucht die Regelung den Soll-Ist-Wert-Abgleich herzustellen. Nun messen Sie die tatsächliche Temperatur an der Lötspitze. Um den Ausgleich herbeizuführen, lässt sich am Drehregler die Heizleistung verstellen. Mangels Pfeil im Zeichenvorrat gibt es das Zeichen „Underline“ für weniger Heizleistung und „Dreieckspitze nach oben“ für mehr Heizleistung. Der angezeigte Kontrollwert entspricht dem Umrechnungsfaktor `adc_gain`.

Durch dieses Verfahren ermitteln wir relativ einfach die für jede Lötspitze charakteristischen Kennwerte, um eine möglichst genaue Lötspitzen-Temperatur zu gewährleisten. Mit Druck auf den „Sich“-Button speichern Sie die so erhaltenen Werte `adc_offset` und `adc_gain` für die in der ersten Zeile angezeigte Spitzennummer. Sind die Kalibrierungen abgeschlossen, kann das Arbeiten mit der neuen Lötstation beginnen. —cm



Aus Alu-Gehäuseprofilen und zugeschnittenen Blechen haben wir ein formschönes Gehäuse gefertigt.

# Von Hackern lernen!

Heft + PDF  
mit 29% Rabatt



Schützen Sie sich vor Angriffen! Das c't-Sonderheft zeigt Ihnen, wie Sie Netzwerk, PC, Smartphone und Online-Accounts absichern:

- ▶ Hacking-Tools als Problemlöser einsetzen
- ▶ Daten in Servern und Webanwendungen absichern
- ▶ Seriösen und sicheren VPN-Anbieter finden
- ▶ Aktuelle Hardware für 2-Faktor-Authentifizierung

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

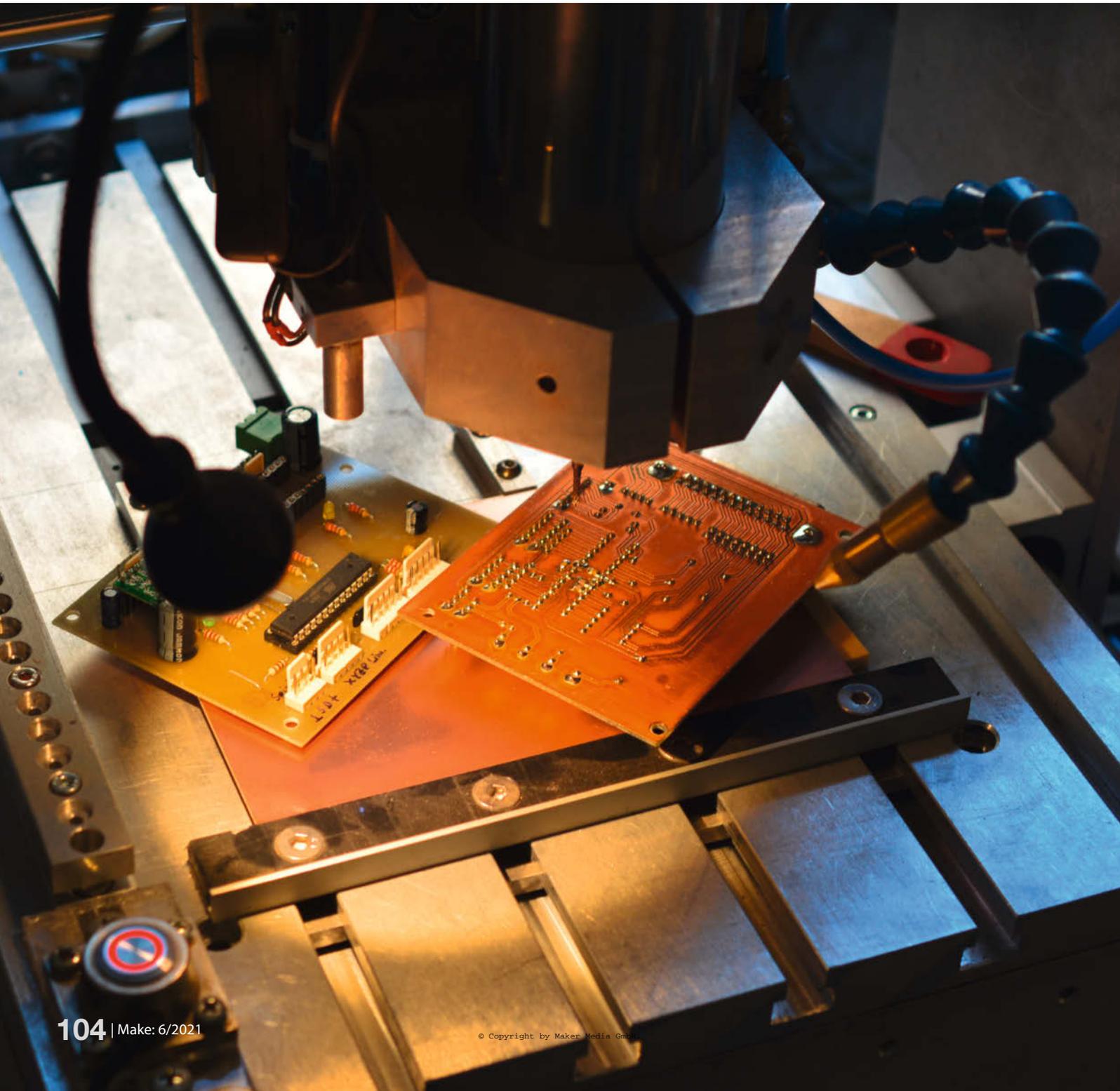


[shop.heise.de/ct-securitytipps21](https://shop.heise.de/ct-securitytipps21)

# Der Weg zur Platine, Teil 1

Was auch immer man mit einer Schaltung realisieren will, irgendwie muss man die elektronischen Bauelemente wie vorgesehen verbinden. Ist die Schaltung übersichtlich klein und auch nicht für die Ewigkeit gedacht, reicht noch ein Breadboard oder eine Lochrasterplatine aus, doch umfangreichere Projekte verlangen eine richtige Platine. Wir zeigen einen einfachen Weg vom Layout bis zur gefrästen Musterplatine.

von Gustav Wostrack



Die Eingabe des Schaltplans und die Layouterstellung sind natürlich immer mit etwas Mühe verbunden, und um die Einarbeitung in ein Layoutprogramm kommt man nicht herum. Etwas anders liegt der Fall, wenn für eine Schaltung bereits ein Platinenentwurf vorliegt. Dann hat ein freundlicher Maker bereits den Entwurfsprozess durchlaufen und eine Datei hinterlegt, mit der die Platine gefertigt werden kann – oder er bietet gleich fertige Platinen zum Kauf an.

Wir gehen hier aber zunächst vom ungünstigsten Fall aus. Der liegt dann vor, wenn wir als Ausgangspunkt eine Schaltung nur als Skizze auf dem Papier oder eben auch als Hardware auf einem Breadboard haben. Dann sind alle Schritte zu durchlaufen, die wir im folgenden detailliert durchgehen.

## Werdegang

Wir haben natürlich den Ehrgeiz, die Platine mit möglichst großer Werkzeugunterstützung zu fertigen. Leider ist mir kein Werkzeug bekannt, bei dem man vorne eine Schaltung eingibt und hinten kommt eine fertige Platine heraus. Vielmehr ist eine nicht ganz unbedeutende Menge an Tools notwendig, die dann jeweils einen Teil des Gesamtprozesses unterstützen. Sie bilden dann die *Toolchain* (Werkzeugkette), bei der das Resultat von Werkzeug n auch von Werkzeug n+1 weiterverarbeitet werden kann.

Der Prozess beginnt mit der Darstellung der Schaltung, also welche Bauelemente benötigt und wie sie untereinander verbunden werden. Daraus entsteht der Platinenentwurf, über den dann später die Platine gefertigt wird. Da ich vermeiden möchte, mit ätzenden Flüssigkeiten oder Laserlicht hantieren zu müssen, habe ich mich für den Einsatz einer CNC-Fräse entschieden. Mit einem VHM-Frässtichel und einem geeigneten Programm wird das Layout (bzw. die Isolationsbahnen) aus der Kupferschicht herausgefräst. Die vorherigen Schritte dürften indes auch für jene interessant sein, die ihr Platinenlayout zum Dienstleister geben.

## Layout mit EasyEDA

Als Entwurfswerkzeug habe ich mir das kostenlose Programm *EasyEDA* ausgesucht. Diverse Gründe sprechen für EasyEDA: Neben problemloser Bedienbarkeit war es vor allem die Vielzahl an Bauelementen, die das Werkzeug direkt oder indirekt über andere Quellen zur Verfügung stellt. Dieser Aspekt ist ganz wichtig, denn was nutzt ein großartiges Werkzeug, das die für die Schaltung notwendigen Bauelemente nicht zur Verfügung stellt?

Sicherlich gibt es Alternativen zu EasyEDA; meist sind das aber in der Funktion oder im

## Kurzinfo

- » Vom Schaltplan zum Layout
- » Aufbereitung der Leiterplattendaten
- » Selbst gefräste Musterplatinen

### Checkliste



**Zeitaufwand:**  
4 Stunden



**Kosten:**  
4 Euro für Platinenmaterial

### Mehr zum Thema

- » Carsten Meyer, MaXYposi-Kontrollpult im Einsatz, Make 5/2017, S. 132
- » Carsten Meyer, Erste Schritte mit der CNC-Fräse, Make 1/2016, S. 92
- » Carsten Meyer, Isolationsfräsen, Make 2/2016, S. 48

### Material

- » FR4-Basismaterial, einseitig kupferkaschiert, Eurokarten-Größe

### Werkzeug

- » Gravierstichel 30° Spitzenwinkel, 0,1mm verrundet, 3,175mm Schaft
- » Hartmetall-Fräser 1mm Durchmesser, 3,175mm Schaft
- » Kleine CNC-Fräse z.B. Genmitsu 3018-PROVer

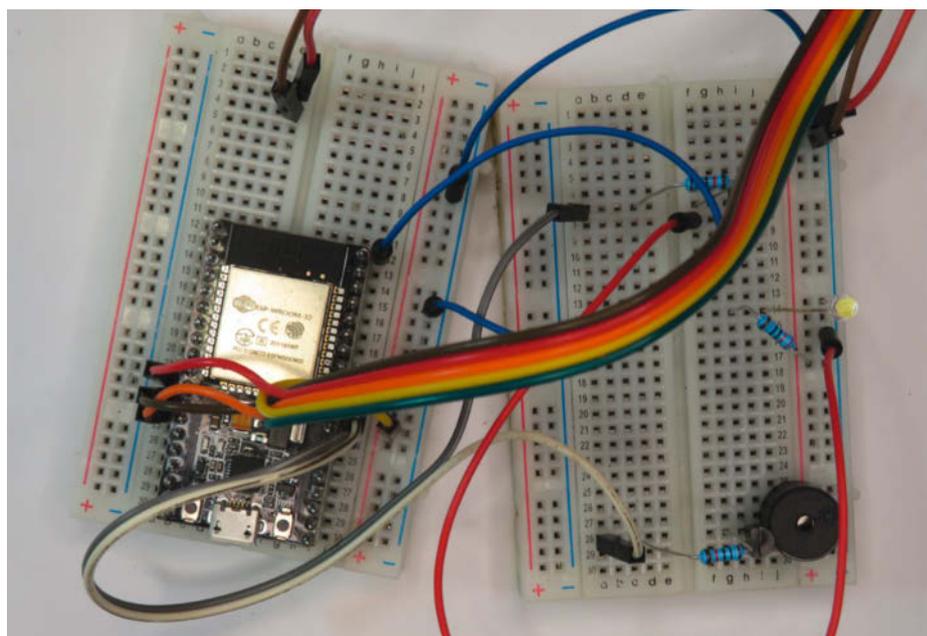
Alles zum Artikel  
im Web unter  
[make-magazin.de/xkch](https://make-magazin.de/xkch)

Umfang abgemagerte Versionen kommerzieller Werkzeuge. EasyEDA stellt zur Fertigung unserer Platine das Gerber-Format zur Verfügung. Mit diesem Format kann man bei einem professionellen Platinenproduzenten seine Platine fertigen lassen. Damit ist meist das sogenannte *Extended Gerber* (RS274X), oder kurz XGerber gemeint. Es gilt als das allgemein gültige Austauschformat und wird von nahe-

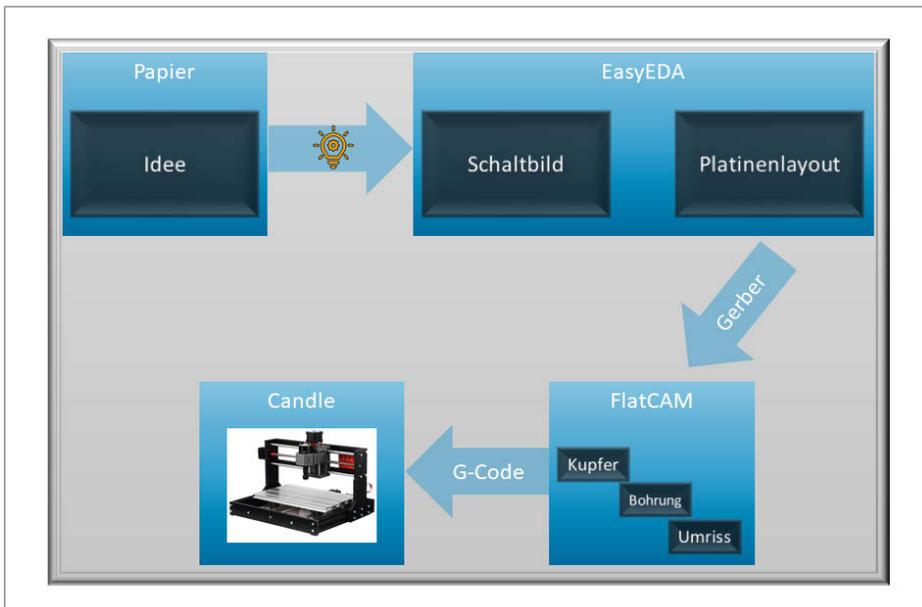
zu allen Layoutprogrammen beherrscht. Damit sind Sie also auf der sicheren Seite.

## Gefräst statt geätzt

Wenn man diesen einfachen Weg gehen will, ist der Prozess hier beendet. Ansonsten benötigen wir nun ein Programm, das das Gerber-Format nun in G-Code wandelt, das



Zum Ausprobieren okay, aber zum dauerhaften Einsatz ungeeignet: Breadboard-Versuchsaufbau.



Die Toolchain von der Idee zur fertigen Platine

die CNC-Fräse für ihren Verfahrensweg benötigt. Hierfür setze ich das kostenlose Werkzeug FlatCAM von Juan Pablo Caram ein. Es macht im ersten Moment einen etwas sperrigen Eindruck, leistet aber alles, was für unsere Zwecke benötigt wird und hat deshalb eine breite Fangemeinde gefunden. Das Ergebnis sind dann eine oder mehrere Dateien im G-Code-Format. Damit füttern wir nun ein Programm, mit dem die CNC-Fräse gesteuert wird.

Ein Frässtichel fährt mit hoher Genauigkeit das Layout des Platinenentwurfs nach. Durch den Vorgang dieses Einritzens werden die einzelnen Leiterbahnen in der Kupferschicht voneinander isoliert, sodass ein Ebenbild des Platinenlayouts auf der Platine entsteht. Dieses Verfahren nennt man daher auch *Isolationsfräsen*. Es folgen noch zwei weitere ähnliche Arbeitsgänge, in denen Löcher für

die Bauelemente gebohrt werden sowie das Heraustrennen der Platine aus einem größeren Werkstück durch einen weiteren Fräsvorgang. Beim ersten Lesen erscheint der Gesamtvorgang langwierig und unübersichtlich. Stimmt! Aber nachdem man den Prozess zwei- oder dreimal durchlaufen hat, sind die einzelnen Arbeitsschritte verinnerlicht und laufen wie von selbst ab.

### Die ersten Schritte

Wir gehen einmal davon aus, dass wir einen Schaltungsentwurf in einer Zeitschrift gesehen haben, den wir gerne umsetzen wollen und steigen damit in das Verfahren ein. Weiterhin setzen wir voraus, dass die oben erwähnten Programme bereits auf unserem Windows-10-Rechner installiert sind.

Für den ersten Schritt starten wir nun EasyEDA. Falls dies der erste Start ist, müssen wir uns zunächst registrieren, ansonsten melden wir uns an und starten ein neues Projekt (*Datei /Neu/Projekt*). Wir nennen es *REFLOW*. Unter *Description* (Beschreibung) tragen wir etwa „Interface-Schaltung für den Reflow-Ofen“ ein. Nachdem wir diese Angaben gespeichert haben, beginnt der Entwurfsprozess mit einem zunächst leeren Layoutplan. Wir haben für unser Projekt keinen Speicherort auf unserer Festplatte angeben müssen. Das liegt daran, dass EasyEDA alle Projekte grundsätzlich in seiner EasyEDA-Cloud speichert. Wollen wir das Projekt weitergeben, müssen wir es explizit exportieren.

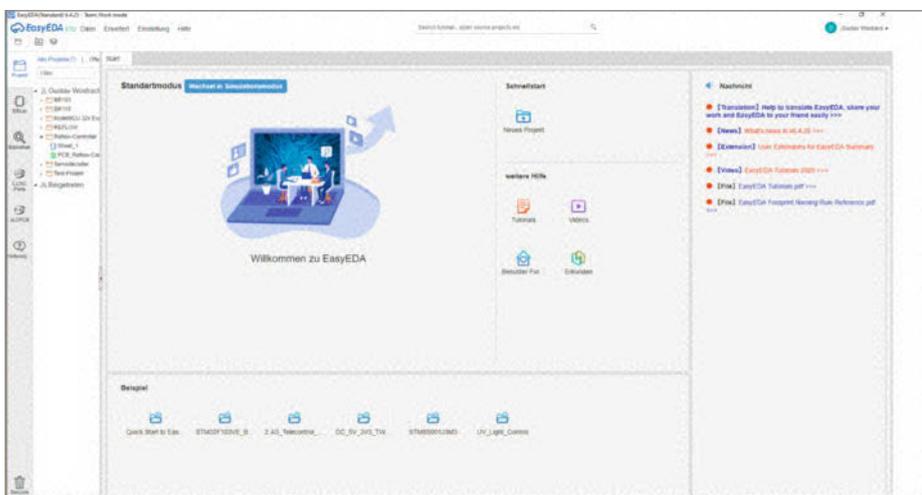
### Schaltungsentwurf

Aus dem Schaltplan entnehmen wir zunächst die benötigten Bauelemente. Als Beispiel suchen wir die beiden Transistoren BC517. Um sie in den Plan einzusetzen, gibt es mehrere Möglichkeiten. Die einfachste ist wohl, die Bibliothek in der linken Spalte des Programmes anzuklicken und in dem nun erschienenen Formular die Bezeichnung der Transistoren *BC517* einzugeben und mit *Enter* zu bestätigen. Es erscheinen nun diverse Fundstellen. Wir können direkt den ersten Eintrag auswählen, da dort die Variante mit dem Gehäuse *TO-92* angegeben ist.

Wenn wir nun den Button *Platzieren* drücken, sind wir wieder zurück in unserem Plan und haben ein Transistorsymbol am Mauszeiger hängen. Drücken wir jetzt die linke Maustaste, bleibt das Transistorsymbol am Mauszeiger, aber es wird auch ein Transistorsymbol auf den Plan gedruckt. Bei jeder weiteren Betätigung der linken Maustaste kommt ein Transistorsymbol auf dem Plan hinzu. Dieses Vorgehen ist sehr praktisch, wenn mehrere identische Symbole benötigt werden.

Wir brauchen aber nur zwei Transistoren, sodass wir nach dem zweiten Klick die *ES-CAPE*-Taste drücken und den Mauszeiger vom Transistorsymbol befreien. Wir können nun die Symbole auf dem Plan anklicken und mit gedrückter Maustaste verschieben bis sie an der gewünschten Stelle im Plan zu liegen kommen. So verfahren wir mit den restlichen Bauelementen aus dem Plan.

Im nächsten Schritt verbinden wir nun die Bauteile, so wie es der Plan vorgibt. Im schwebenden Fenster *Verdrahtung Werkzeuge* klicken wir das Drahtsymbol *Wire* an oder wir geben einfach den Buchstaben *W* ein. Der Mauszeiger ändert sich in ein Fadenkreuz. Gehen wir an ein Ende eines Bauteiles, wird dort ein ausgefüllter Kreis angezeigt. Jetzt betätigen wir die Maus und haben damit ein Ende einer Verbindung festgelegt. Genauso verfahren wir mit dem anderen Ende. Im Schaltbild sind „schräge“ Verbindungen verpönt; deshalb fügen wir an

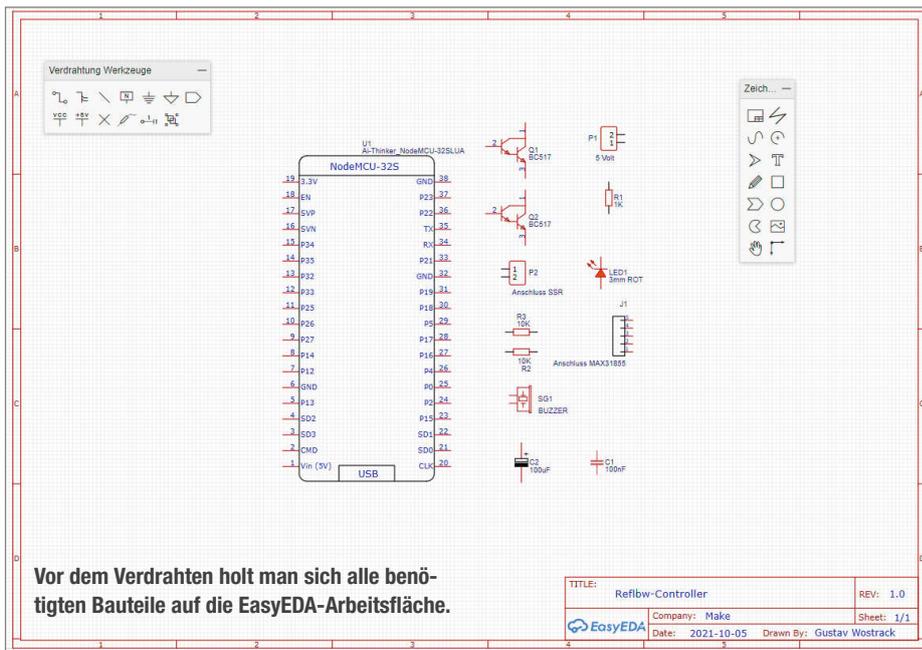


Aufgeräumt: EasyEDA organisiert Schaltpläne und zugehörige Layouts in Projekten.

geeigneter Stelle 90°-Knicke ein: Verbindung am gewünschten Knickpunkt anklicken und dann zurechtrücken.

Auf diese Weise haben wir alle Verbindungen schnell gezogen. Dieser Arbeitsschritt erfüllt somit zwei Zwecke. Einmal haben wir ein vorzeigbares Schaltbild und zum Zweiten – und das ist der Hauptzweck – haben wir die Grundlage für den Entwurf der Leiterplatte gelegt (oft als *Netzliste* bezeichnet). Es gibt auch Design-Programme, die sich diesen Schritt schenken. Im Zusammenhang mit der Anfertigung einer SMD-Platine im zweiten Teil lernen wir ein solches Programm kennen.

Da dem Programm nun bekannt ist, welche Schaltung wir auf eine Platine bringen wollen, können wir nun zum nächsten Schritt übergehen und das Platinenlayout entwerfen. Die Übertragung des Schaltplans in den Layouter geschieht durch einen Klick auf *Entwurf/Konvertiere Schema in PCB* oder einfach Alt+P. Bevor wir aber zur Platine geführt werden, bietet uns das Programm eine Überprüfung an. Die sollte man, insbesondere wenn man Neuling ist, auf jeden Fall wahrnehmen. Es werden jetzt diverse Warnungen angezeigt. Man muss dabei allerdings wissen, dass in unserem Falle viele Warnungen auf dem Umstand beruhen, dass die meisten Pins des



Vor dem Verdrahten holt man sich alle benötigten Bauteile auf die EasyEDA-Arbeitsfläche.

ESP32 nicht verbunden sind; das ist hier aber gewollt. Wenn man sich sicher ist, dass keine Fehler die Ursache für die Warnungen sind, kann man die erneut gestellte Frage nach der Überprüfung getrost mit *NEIN* beantworten.

Aber wir sind noch nicht bei der Leiterplatte, denn vorher müssen wir noch deren Abmessungen angeben. In unserem Falle geben wir ein *Start (X): 0; Start (Y): 70.0; Breite: 55.0; Höhe: 70.0*. Die Maßeinheit lassen wir



## Let's code!

Scratch, die farbenfrohe Drag-and-drop-Programmiersprache, die fast ohne geschriebenen Code auskommt, wird auf der ganzen Welt von Millionen von Coding-Einsteigern verwendet. Al Sweigart macht es mit seinem Buch einfacher denn je, deine Programmierfähigkeiten Block für Block aufzubauen.

2. Auflage  
198 Seiten · 19,95 €  
ISBN 978-3-86490-859-0



## Tüftelspaß für Kids und Family



208 Seiten · 24,90 €  
ISBN 978-3-86490-806-4



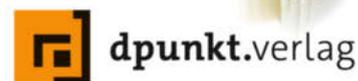
160 Seiten · 19,95 €  
ISBN 978-3-86490-537-7

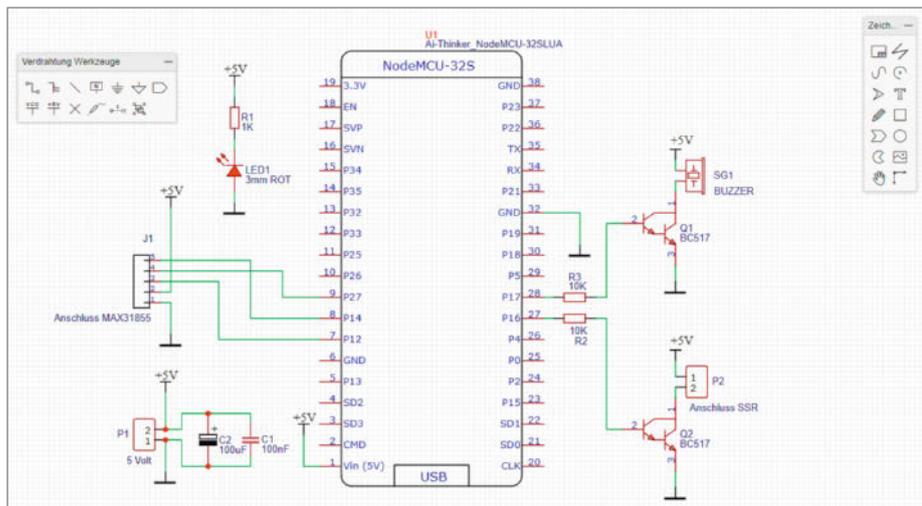


364 Seiten · 26,90 €  
ISBN 978-3-86490-296-3



428 Seiten · 29,90 €  
ISBN 978-3-86490-433-2





Auch beim Schaltplanentwurf sollte man die Bauteile so anordnen, dass wenig Kreuzungen und möglichst kurze Verbindungen entstehen.

natürlich bei mm. Der eventuelle Versuch, den Eintrag *Copper Layer* von 2 auf 1 zu ändern, wird mit dem Hinweis verhindert, dass einlagige Platinen momentan von EasyEDA nicht unterstützt werden. Egal, wenn man auf dem zweiten Layer nichts anlegt, wird er auch nicht ausgegeben. Nicht nur die Eintragung von Breite und Höhe sind wichtig, sondern auch die für die beiden Startpunkte. Denn

die bestimmen später, wo der Nullpunkt für die CNC-Fräse zu liegen kommt.

### Luftlinien im Rattennest

Wenn wir nun auf den Knopf *An Fenster anpassen* drücken, sehen wir die Umriss unserer Platine sowie daneben alle Bauelemente, die durch sogenannte *Luftlinien* miteinander

verbunden sind. Diese Luftlinien sind die Verbindungen, die wir im vorangegangenen Arbeitsschritt in das Schema eingezeichnet haben. Somit sind sie die Vorläufer der Leiterbahnen auf unserer Platine. Wenn wir die Bauelemente mit der Maus anfassen und verschieben, dann bleiben diese Verbindungen erhalten. Jetzt kommt der kreativste Teil der ganzen Prozedur.

Bei umfangreicheren Projekten werden die Luftlinien-Verbindungen zunehmend unübersichtlich; es entsteht ein undurchschaubares „Rattennest“. Erster Schritt muss dann sein, die Bauelemente so auf der Platine zu verteilen, dass möglichst wenig Überschneidungen der Luftlinien entstehen (die später zu Kreuzungen von Leiterbahnen werden). Das ist bei einer einseitigen Platine schon eine Herausforderung.

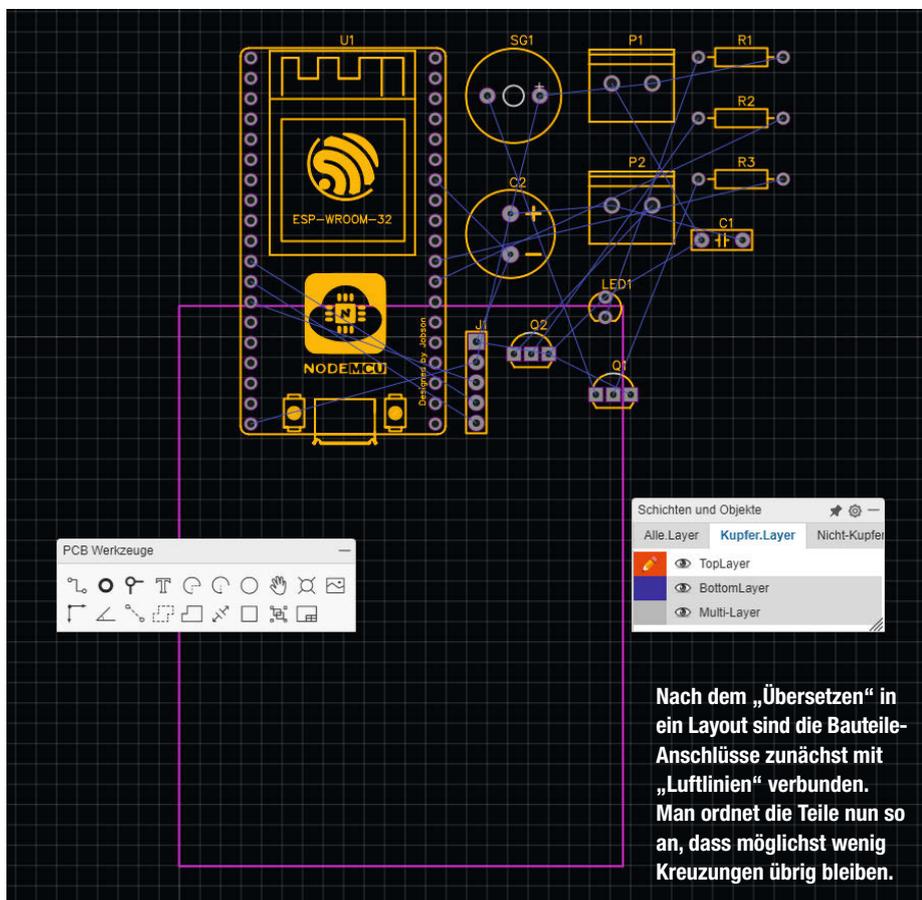
Man kann dazu die Bauelemente so lange verschieben und rotieren, bis das gewünschte Resultat erzielt ist. Widerstände, auch Kondensatoren laden gerade dazu ein, „untertunnelt“ zu werden, d.h. eine Leiterbahn läuft unter dem Element hindurch, wodurch eine Kreuzung realisiert werden kann. Lässt sich das nicht realisieren, müssen Drahtbrücken her. Sollten davon zu viele gebraucht werden, bleibt nichts anderes übrig, als die einseitige Platine aufzugeben und auf eine zweite Seite auszuweichen.

### Aufgeräumt

Hat man es dann endlich geschafft, eine halbwegs kreuzungsfreie Verteilung abzuliefern, müssen nun die Luftlinien durch Leiterbahnen ersetzt werden. Dazu aktivieren wir in dem schwebenden Fenster *Schichten und Objekte* den Reiter *Kupfer.Layer* und klicken dann auf *Bottom Layer*. Denn auf der Oberseite sind die Bauelemente und die Kupferschicht mit den Leiterbahnen ist auf der Unterseite.

Zum Verlegen der Leiterbahnen gehen wir analog zur Verdrahtung unseres Schaltplanes vor. Aus dem Fenster *PCB Werkzeuge* wählen wir *Track* bzw. *W* auf der Tastatur aus. Es ist das gleiche Symbol wie beim Schaltplan, allerdings mit dem Unterschied, dass die Verbindungen zwischen den Bauelementen nicht mit einer Linie, sondern mit einer Leiterbahn gezogen werden. Auch hier heißt es wieder, ein Bauelement an einem Beinchen anzuklicken und die Kupferbahn so zu verlegen, dass keine andere Verbindung verhindert wird.

An dieser Stelle soll noch darauf hingewiesen werden, dass auch jetzt noch Änderungen am Schaltplan möglich sind. Dazu klickt man auf den entsprechenden Reiter am oberen Rand des aktuellen Fensters und kann dann die Änderungen im Schaltplan vornehmen. Anschließend kommt man mit *Entwurf/PCB Aktualisieren* wieder zurück zur aktualisierten Platine. Hat man das alles hinter sich, sollte man



jetzt über die Breite der Leiterbahnen nachdenken. Denn die sollten möglichst breit sein, dann wird das spätere Fräsen unkritisch; ohnehin sollte man Masse- und Versorgungsleitungen breitere Leiterbahnen gönnen. Um die Breite zu ändern, klickt man die betroffene Leiterbahn und kann dann im Fenster rechts (Leiterbahneigenschaft) den Eintrag bei Breite entsprechend ändern. Eventuell muss man die Leiterbahn etwas verschieben, damit ausreichend Raum für die neue Breite entsteht.

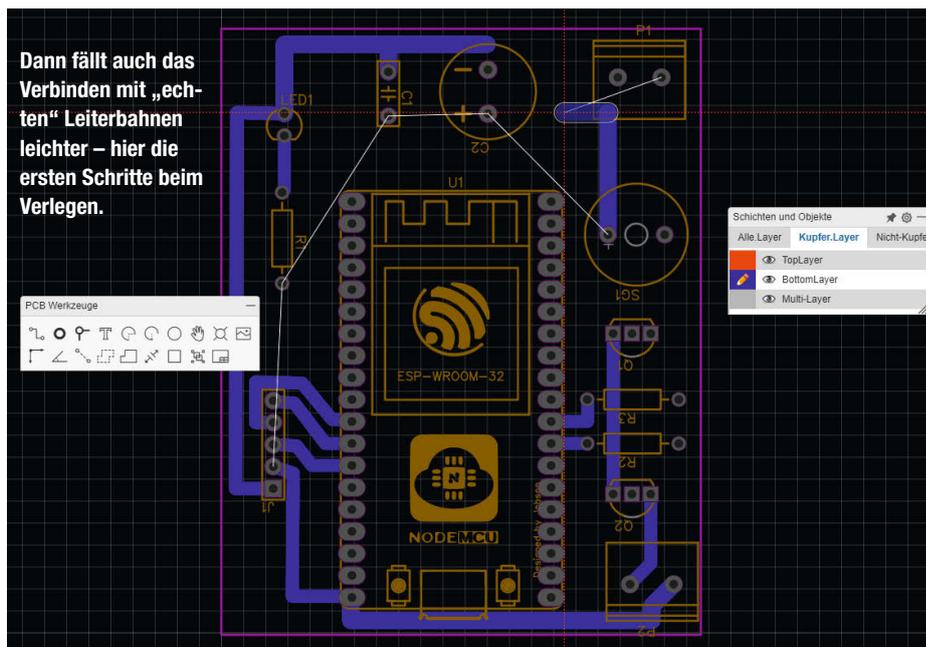
Sollte das angesprochene Fenster rechts nicht sichtbar sei, so hilft es, auf den schmalen Anfasser ganz rechts am Fensterrand zu klicken. Als letzten Schritt müssen wir leider noch etwas Fleißarbeit erbringen. Es geht um die Durchmesser der Bohrlöcher, die bei den verwendeten Bauelementen natürlich sehr unterschiedlich sind. Wir müssen sie daher markieren und in Fenster auf der rechten Seite unter *Lochdurchmesser* für alle Löcher, die ein Bauelement aufnehmen, einen möglichst einheitlichen Wert eingeben. Dies erspart uns später Schwierigkeiten beim Erstellen der entsprechenden Bohr-G-Code-Datei. Ich habe hier 1,0mm eingegeben, der ist für Steckverbinder nicht zu klein und für normale Bauteile-Anschlüsse noch nicht zu groß.

## Exportschlager

Ist man dann irgendwann mit dem Ergebnis zufrieden, so muss die Platine im richtigen Format gespeichert werden, um sie später der CNC-Fräse zur Verfügung stellen zu können. Das Format unserer Wahl ist wie bereits weiter oben dargestellt das Gerber-Format. Dieses Format hat sich industrieweit als das Austauschformat für Platinenlayouts etabliert. Den Export stoßen wir an mit *Fabrikierung/ Gerber-Datei* oder mit *Datei/Generate PCB Fabrication File(Gerber)*. Bevor Sie nun endlich zum Speicherdialog gelangen, haben Sie noch einmal die Chance, Ihr Ergebnis hinsichtlich der Design-Regeln (Mindest-Abstände usw.) überprüfen zu lassen. Wenn Sie unter *Entwurf/Designregel* keine eigenen aufgestellt haben, werden die Standardregeln bei der Überprüfung angewendet. Im folgenden Dialogfenster können wir nun die Gerber-Datei erzeugen lassen – und uns gegebenenfalls für die Abkürzung entscheiden und die Leiterplatte bei einem Dienstleister in Auftrag geben.

Ich denke, Sie haben sich für die Gerber-Datei entschieden, die Sie dann komprimiert im Verzeichnis Ihrer Wahl vorfinden. Der Name lautet etwa *Gerber\_PCP\_REFLOW\_DATUM.ZIP*. Wie vermutet enthält diese ZIP-Datei mehrere Einzeldateien:

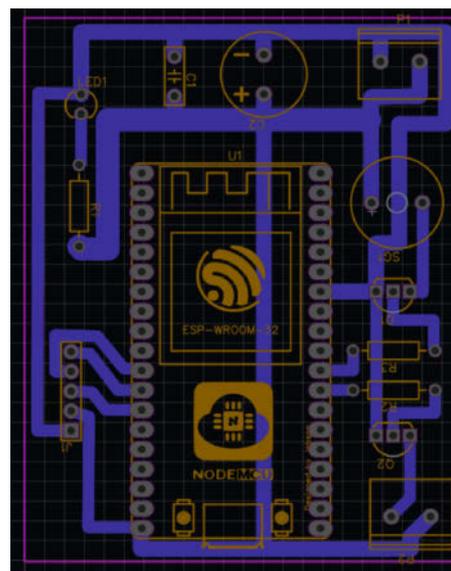
1. *Gerber\_BoardOutline.GKO*: Hier sind die Umriss der Platine aufgezeichnet. Wir verwenden später diese Datei, um unsere Platine aus einem größeren Stück herauszufräsen.



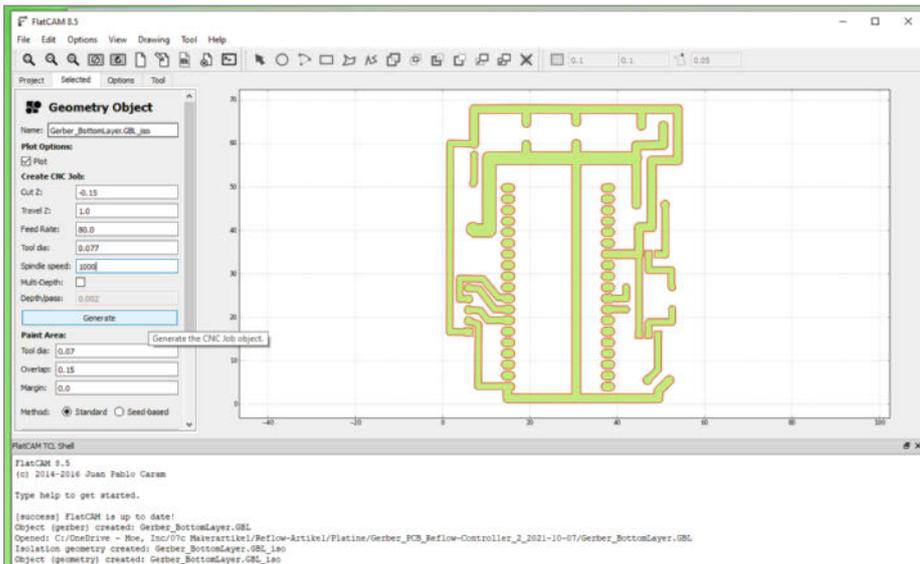
2. *Gerber\_TopLayer.GTL*: Wenn wir eine zweiseitige Leiterplatte angefertigt hätten, dann wären hier die Leiterbahnen der Oberseite enthalten.
3. *Gerber\_BottomLayer.GBL*: Das ist wohl unsere wichtigste Datei, denn hier sind die Leiterbahnen der unteren Kupferseite aufgezeichnet.
4. *Gerber\_Inner1.G1, Gerber\_Inner2.G1 ...*: Wenn wir eine mehrlagige Leiterplatte entworfen hätten, dann wären hier die Leiterbahnen der inneren Kupferschichten enthalten.
5. *Gerber\_TopSilkLayer.GTO*: Das sind die Symbole der Bauelemente auf der Oberseite. Für unsere Anwendung relevant, wird aber nicht weiter verwendet, da wir keinen Druck auf der Oberseite aufbringen.
6. *Gerber\_BottomSilkLayer.GBO*: Wie oben, nur Bauelemente auf der Unterseite.
7. *Gerber\_TopSolderMaskLayer.GTS*: Lötmaske der Oberseite. Für unsere Anwendung nicht relevant.
8. *Gerber\_BottomSolderMaskLayer.GBS*: Lötmaske der Unterseite. Für uns relevant, wird aber nicht angewendet.
9. *Gerber\_Drill\_PTH.DRL*: Bohrschema, gibt an, an welchen Stellen Bauelemente durch die Platine gesteckt werden. Bei einer industriellen Fertigung wird die Innenseite des Bohrloches metallisiert, sodass bei einer zweiseitigen Leiterplatte eine Verbindung zwischen Ober- und Unterseite als Durchkontaktierung entsteht. Diese Datei wird von uns benötigt, der Aspekt der Durchkontaktierung wird von uns aber nicht umgesetzt.
10. *Gerber\_Drill\_NPTH.DRL*: Ebenfalls ein Bohrschema, aber ohne Durchkontaktierung. Hier werden nur die elektrisch irrelevanten

Bohrungen eingebracht, beispielsweise für Halterungen.

11. *Gerber\_TopPasteMaskLayer.GTP*: Mit diesem Layer wird die Lötmaske für SMD-Lötungen auf der Platinenoberseite erzeugt, dem sogenannten *Stencil*. Für diese Anwendung nicht nötig, da wir keine SMD-Bauteile einsetzen. Im zweiten Teil unserer Artikelserie werden wir allerdings ein *Stencil* einsetzen, um die winzigen SMD-Teile verlöten zu können.
12. *Gerber\_BottomPasteMaskLayer.GBP*: Dito für die Unterseite.



Bei zu fräsenden Platinen wählt man eine möglichst große Leiterbahnbreite. Masseleitungen sollten ohnehin besonders dick sein, das vermindert Störimpulse durch die Stromaufnahme der Bauteile.



FlatCAM ist auf das Isolationsfräsen spezialisiert: Es erzeugt einen Werkzeugweg um die Leiterbahnen herum.

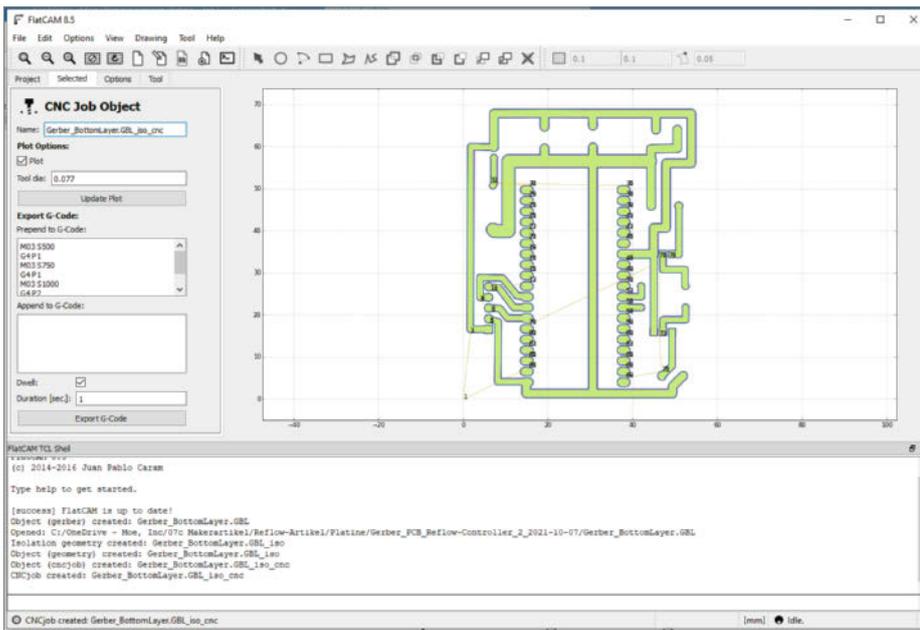
Das waren jetzt die wichtigsten Dateien, die zum Gerber-Format gehören. Wir benötigen zunächst, wie bereits oben angemerkt, nur die Dateien mit dem Bohrschema (9), den Umrissen (1) und den Lötbahnen (3) für die Unterseite.

### Vom Gerber zum G-CODE

G-Code und Gerber sind sich zwar ähnlich, sie stehen aber für unterschiedliche Dinge. Eine Gerber-Datei beschreibt, wie ein Layer einer Leiterplatte aussieht – genau genommen den Weg eines Fotoplotters, was historische Gründe hat. Dagegen wird G-Code beim Isolations-

fräsen zur Ansteuerung von CNC-Fräsen bzw. dessen Fräs Werkzeug verwendet. Dabei muss die Bewegung des Stichels außen um die Kupferelemente herumlaufen, sodass Isolationskanäle aus dem Kupferbelag herausgefräst werden. Bei der Konvertierung vom Gerber-Format zum G-Code müssen Werkzeugdurchmesser und Leiterbahnbreiten und weitere Aspekte berücksichtigt werden.

Für diesen Schritt werden wir das Programm FlatCAM einsetzen. Wir starten FlatCAM und öffnen mit *File/Open Gerber* die Gerber-Datei, die wir fräsen wollen. Das ist zunächst die Datei *Gerber\_BottomLayer.GBL*.



Nach dem „Generate“ zeigt uns FlatCAM auch, in welcher Reihenfolge die Isolationsbahnen gefräst werden.

Sobald der Dateiname eingegeben wurde, erscheinen im rechten Fenster die Leiterbahnen der Platine. Den Dateinamen markieren wir durch Anklicken. Wir wählen dann den Reiter *Selected* aus. In dem nun erschienenen Fenster müssen die Angaben eingetragen werden, wie wir anschließend unsere Platine bearbeiten wollen. Unter der Überschrift *Isolation Routing* geben wir den Durchmesser des Fräswerkzeugs ein.

### Gefräst, nicht geätzt

Bevor wir die Werte einsetzen, ein paar Worte zu der eingesetzten Hardware. Die meisten Anwender setzen für das Isolationsfräsen Gravierstichel ein. Um Geld zu sparen, hatte ich anfangs günstige Werkzeuge aus Fernost benutzt. Da ich aber damit zu oft enttäuscht wurde, beziehe ich nun mein Zubehör aus heimischer Herstellung; der Aufpreis zahlt sich aus. Das gilt auch für das Platinenmaterial.

*Tool dia:* Hier ist prinzipiell der Durchmesser des verwendeten Werkzeuges einzutragen. Der Frässtichel, den ich nutze, ist aber konusförmig, sodass der Spitzenwinkel und die Eindringtiefe den effektiven Werkzeugdurchmesser ergeben. Mit der Formel

$$b = \text{Spitzendurchmesser} + 2 \times \text{Eintauchtiefe} \times \tan(\text{Spitzenwinkel} / 2)$$

lässt sich der effektive Werkzeugdurchmesser berechnen. Realistisch sollte man nicht unter 0,2mm ansetzen.

*Width (#passes):* Hier ist nicht die Weite, sondern die Anzahl der Umläufe gemeint. Beim Fräsen der Leiterbahnen hier eine 1 eintragen.

*Pass overlap:* Bei nur einem einmaligen Umlauf spielt dieser Wert eine untergeordnete Rolle. Wir tragen hier auch eine 1 für 100 Prozent ein. *Combine Passes* haken wir auch ab.

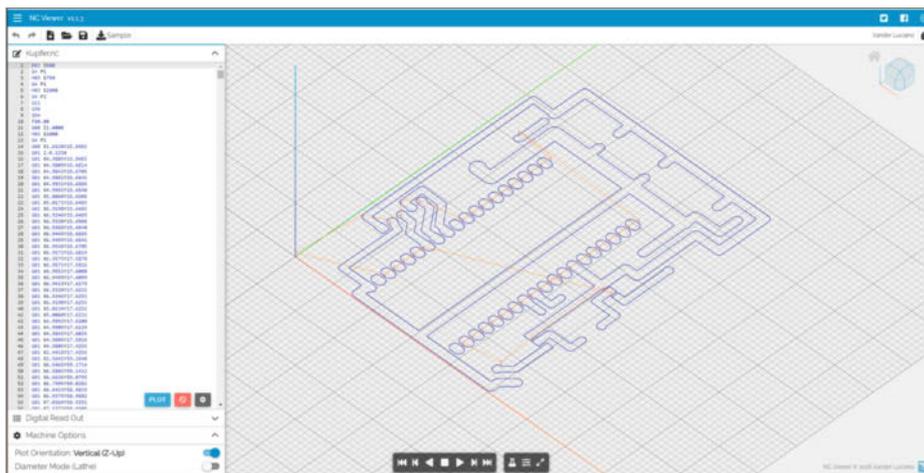
Sind alle Werte eingetragen, klicken wir *Generate Geometry* an. Im rechten Fenster wurde daraufhin neben den Leiterbahnen eine rote Linie gezeichnet. Das ist der Pfad, an dem später unser Fräser Material herausfräsen wird. Es lohnt sich, diese Linie genau anzuschauen, insbesondere da, wo Leiterbahnen eng beieinander verlaufen. Anschließend wechseln wir wieder auf den Reiter *Project* und markieren das neu generierte File (*\*\_iso*). Jetzt wieder auf den Reiter *Selected* wechseln.

*Cut Z* ist die gleiche Frästiefe wie oben bereits verwendet von -0,125mm (diesmal mit einem Minuszeichen, denn wir wollen unter die Oberfläche). So tief dringt der Fräser später in das Material ein. Der Wert sollte natürlich größer sein als die Kupferdicke von ca. 0,035mm. *Travel Z* setzen wir auf 1mm. Um diesen Wert hebt sich der Fräser an, wenn er eine neue Position erreichen muss.

Die *Feedrate* ist die Geschwindigkeit, mit der sich der Fräser durch das Material bewegt. Da mein Fräsmotor keine allzu hohen Drehzahlen (und damit keine hohen Schnittgeschwindigkeiten) erlaubt, setzte ich hier nur 60–80mm/min ein. Bei der Umdrehungsgeschwindigkeit des Fräsmotors *Spindle Speed* tragen wir einfach 1000 (maximale Drehzahl) ein. Die Eintragung macht nur Sinn, wenn die Steuerung über die Möglichkeit verfügt die Geschwindigkeit des Fräsmotors zu kontrollieren.

Dann wieder auf *Generate* klicken. Wir können sehen, dass sich die roten Linien blau verfärbt haben und mit Zahlen versehen sind. Während die roten Linien den mittleren Verfahrensweg, d.h. den Mittelpunkt des Fräsers darstellen, geben die nun erschienen blauen Linien die tatsächliche Breite des ausgefrästen Kanals an. Die Zahlen zeigen die Reihenfolge an, wie die Leiterbahnen bearbeitet werden.

Nun wieder auf den Reiter *Project* und das neue File markieren (\*\_iso\_cn). In den Reiter *Selected* wechseln. Hier haben wir die Möglichkeit, etwas eigenen G-Code vor und hinter den im Anschluss von FlatCAM generierten Code einzufügen. Mit den folgenden Zeilen, die ich in das Feld *Prepend to G-Code* einlege, erreiche ich, dass die Drehzahl des Fräsers in Stufen auf die Enddrehzahl von



Die endgültige Kontrolle des Werkzeugwegs anhand des erzeugten G-Codes erfolgt mit dem Programm NC Viewer.

1000 hochläuft. Damit werden Stromspitzen am Controller vermieden.

```
M03 S500
G4 P1
M03 S750
G4 P1
M03 S1000
G4 P2
```

Anschließend auf *Export G-Code* klicken. Bei der folgenden Möglichkeit, den Dateinamen des generierten G-Codes anzugeben, sollten Sie am besten die Endung *.nc* verwenden, da das folgende Programm dies erwartet. Da hier die Kupferseite der Platine bearbeitet wurde, nenne ich diese Datei immer *Kupfer.nc* und lege sie in das gleiche Verzeichnis wie die Gerber-Dateien.

# ct Fotografie

## Das Magazin von Fotografen für Fotografen



**Porträtfotograf und c't Fotografie-Autor**

Tilo Gockel, Fotografiert u. a. für Spiegel-Online, Vogue.it, Pablo-Magazin, DIGIT

*„Porträts? Sind hart und stressig... und damit die interessanteste Art von Aufnahmen! :-)”*

Tilo Gockel, Fotopraxis.net



### 2 x c't Fotografie testen

- 2 Ausgaben kompaktes Profiwissen für 14,30 €
- 35 % Rabatt gegenüber Einzelheftkauf
- Inkl. Geschenk nach Wahl



Jetzt bestellen:

[www.ct-foto.de/miniabo](http://www.ct-foto.de/miniabo)



Für Platinen bis zur Eurokarten-Größe ist die Genmitsu 3018-PROVer eine gute Wahl. Im Unterschied zu anderen Portalfräsen verfährt hier die Traverse nur in X-Richtung, für die Y-Achse wird die Arbeitsfläche bewegt.

Nachdem nun ein vollständiger Durchlauf für einen Layer abgeschlossen ist, sollten wir vielleicht ein kleines Zwischen-Fazit ziehen. Wir haben ein häufiges Hin- und Her zwischen den Reitern *Project* und *Selected* erlebt. Vielleicht haben Sie auch die Logik dahinter erkannt, nämlich „Werte eintragen“ und „Werte anwenden“. Dieses Schema sollten wir im Hinterkopf behalten, denn wir müssen noch zweimal durch diese Prozedur, nämlich für die Bohrungen und dann für das Herausfräsen der Platine. Wir könnten nun zunächst den Kupfer-G-Code anwenden, die Platine fräsen und anschließend wieder zu FlatCAM zurück und so weiter. Wir ersparen uns dieses Durcheinander und bereiten die drei benötigten G-Code-Dateien gemeinsam vor.

### Die Bohrdatei

Zunächst laden wir die Datei *Gerber\_Drill\_PTH.DRL*. Aber diesmal mit *File / Open Excellon*, ansonsten handeln wir uns eine Fehlermeldung

ein. Im rechten Fenster sehen wir nun neben den Leiterbahnen auch die Bohrungen für die Bauelemente. Nun das bekannte Spiel, Dateiname markieren und *Selected* anklicken. Bei *Tools* sehen wir jetzt nur einen Eintrag, nämlich 1.0. Hätten wir beim Platinenlayout alle ursprünglichen Bohrdurchmesser zugelassen, hätten wir hier eine Fülle von zu beachtenden Durchmessern. So klicken wir in das Feld mit der Nummer 1 unterhalb des Zeichens # und markieren damit, dass dieser Durchmesser zu bearbeiten ist.

Bei *Cut Z* tragen wir dieses Mal -2 ein, damit der Bohrer auch sicher durch die (1,6mm dicke) Platine kommt. Die restlichen Werte sind bekannt. Wir klicken auf *Generate* und anschließend wieder auf *Project*. Dort markieren wir den neuen Eintrag *\*\_cnc* und gehen wieder auf *Selected*. Hier tragen wir nur den Durchmesser mit 1,0 und wieder die wenigen Zeilen G-Code ein, die wir bereits oben verwendet haben. Der Button *Export to G-Code* erzeugt schließlich den Bohr-G-Code, wir nennen ihn *Bohru.nc*.

### Platinenumriss

Nun fehlt noch die Umrissdatei. Wir laden dazu die Datei *Gerber\_BoardOutline.GKO* mit *File / Open Gerber*, markieren den Eintrag und gehen zu *Selected*. Wir gehen zum Ausschnitt *Board Cutout*. Dort finden wir nun folgende Einträge:

*Tool dia*: Für das Herausfräsen der Platine nutze ich das gleiche Werkzeug wie für das Bohren, nämlich einen 1mm Fräser. Also trage ich hier 1.0 ein. Schneller ginge es allerdings mit einem dickeren und damit robusteren Fräser mit 2 oder 2,5mm Durchmesser. FlatCAM korrigiert den Fräsweg um den jeweiligen Werkzeugdurchmesser.

*Margin*: Hier trägt man einen Wert > 0 ein, wenn die Platine größer als der ursprünglich veranschlagte Wert werden soll. Ich trage eine 0 ein.

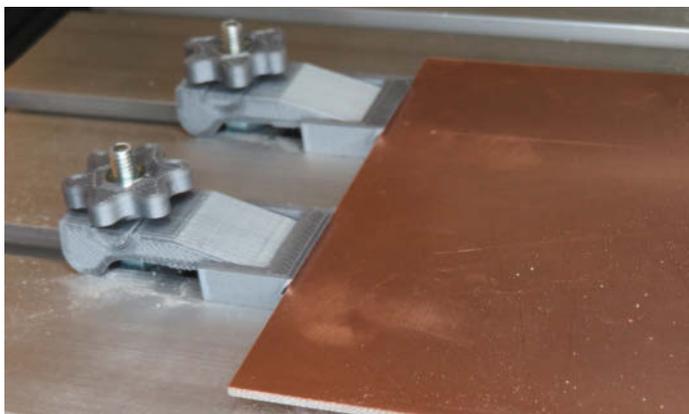
*Gap Size*: Damit die Platine nicht beim Ende des Fräsens unkontrolliert herausfällt, können hier kleine Stege definiert werden. Ein kleiner Wert wie 0.1 ist empfehlenswert. *Gaps* gibt die Anzahl der Stege an, der Wert sollte mindestens bei 2 liegen.

Nun wieder bei *Project* den neuen Eintrag markieren, bei *Selected* die Werte für *Cut Z* mit -2 und *Tool dia* mit 1.0 eintragen. Um unseren Fräser nicht abzuberechnen, erledigen wir diesen Gang nicht in einem Durchgang, sondern arbeiten uns langsam vor, indem wir bei *Multi-Depth* ein Häkchen setzen und im Feld *Depth/pass* beispielsweise 0.2 eingeben. Dann läuft der Fräser bei 1,6mm Platinendicke achtmal herum. Das dauert zwar etwas länger, aber der dünne Fräser überlebt es. Dann quittieren wir mit *Generate*. Erwartungsgemäß wählen wir den neuen Eintrag aus und erzeugen unter *Selected* mit den bekannten Zeilen G-Code die entsprechende Datei *Umriss.nc* mit *Export G-Code*. Nun liegen alle Dateien vor, die wir zum Fräsen der Leiterplatte benötigen.

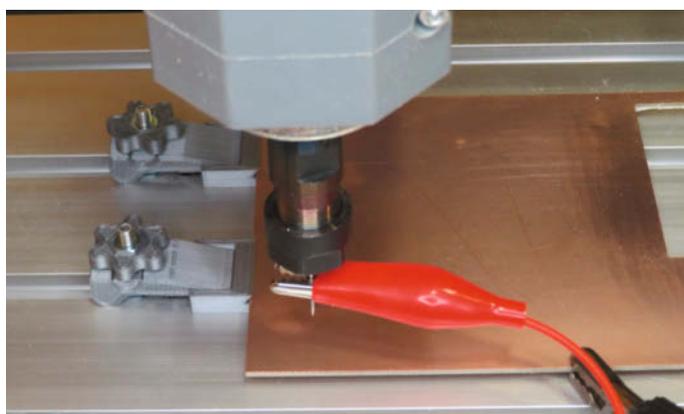
Zum Abschluss noch ein Hinweis zu FlatCAM. Wenn Sie häufiger mit diesem Werkzeug arbeiten, empfiehlt es sich, den Reiter *Options* auszufüllen. Damit setzen Sie alle Ihre spezifischen Standardwerte und ersparen sich eine häufige Neueingabe. Es ist übrigens zu diesem Zeitpunkt sehr empfehlenswert, den Inhalt der G-Code-Dateien auf Fehlerfreiheit zu überprüfen. Das machen wir mit einem G-Code-Viewer, beispielsweise *NC Viewer*, der auf *NCViewer.com* als Web-Anwendung zur Verfügung steht. Sie können dann hier schon sehen, welchen Weg Fräser und Bohrer nehmen. Fehler, die man hier entdeckt, kosten kein Geld.

### Kleinfräsen

Wikipedia definiert CNC-Fräsen als „Werkzeugmaschinen, die durch den Einsatz von Steuerungstechnik in der Lage sind, Werkstücke mit hoher Präzision auch für komplexe Formen automatisch herzustellen“. Ein solches Gerät



Die selbstgedruckten Platinenhalter lassen auf der Unterseite genügend „Luft“ zum Bohren.



Kupferoberfläche und Gravierstichel bilden gleichzeitig auch den Sensor zum Abtasten der Oberfläche und zum Einstellen des Z-Nullpunkts.

besteht prinzipiell aus drei Komponenten: dem mechanischen Aufbau der Fräse mit Nutenplatte und Fräskopf, einem Controller, der die Stepermotoren für das Verfahren der Nutenplatte in X- und Y-Richtung sowie des Fräskopfes in Z-Richtung steuert, und letztlich einer Steuersoftware, die auch als das Bediener-Interface fungiert.

Wenn Sie im Internet das Schlagwort *Isolationsfräsen* eingeben, bekommen Sie häufig den Hinweis, dass der Vorteil der Methode die schnelle und chemiefreie Herstellung von Leiterplatten ist, dafür aber sehr hochwertige und damit teure CNC-Maschinen notwendig sind. Das hält viele davon ab, sich an dieses Projekt heranzuwagen.

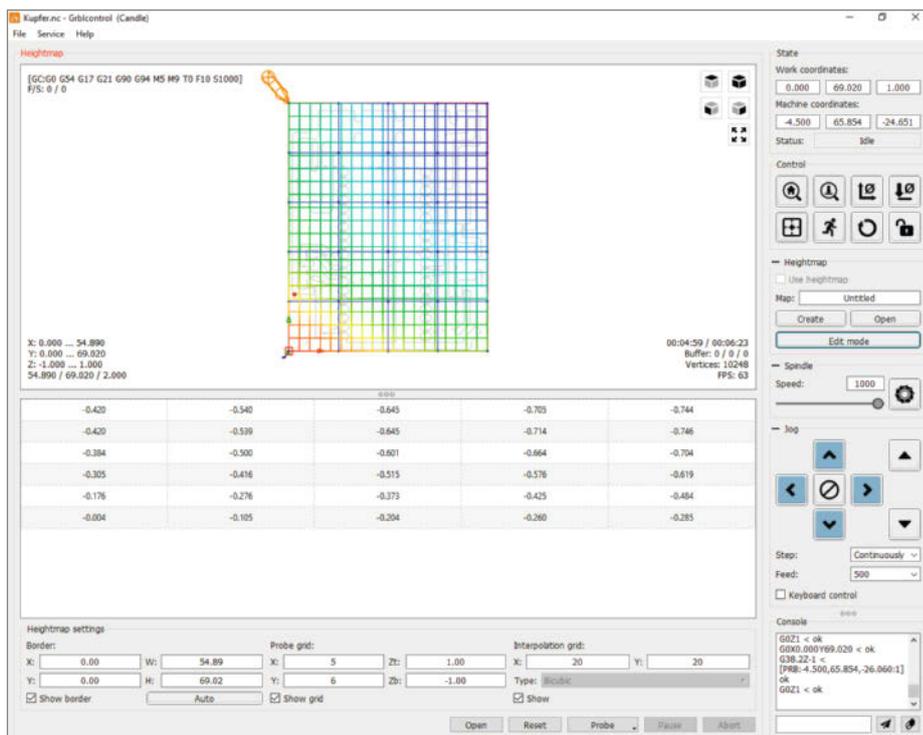
Aber: Wir haben nicht den Anspruch, Bauteile für die nächste Mondlandung zu liefern. Ich hatte seinerzeit die *Genmitsu 3018-PROVer* angeschafft, eine preiswerte und noch recht kompakte Maschine aus China. Aber auch in Deutschland werden herausragende Produkte für Maker und Modellbauer gefertigt.

Bevor ich meine erste Platine gefräst habe, waren doch noch diverse Einstellungen und Anbauten an der Maschine notwendig. So sind Endschalter zur Begrenzung der Achsen unbedingt zu empfehlen. Ansonsten können Sie den Nullpunkt der Maschine nicht einstellen und die Maschine bleibt während eines Fräsvorganges unvermittelt stehen, weil sie annimmt, sich außerhalb ihres möglichen Verfahrensweges zu befinden. Ebenso ist der Tastmechanismus zur Einstellung des Nullpunktes der Z-Achse unverzichtbar. Man muss sich vergegenwärtigen, dass beim Isolationsfräsen von Leiterbahnen auf einer Leiterplatte mit einer 35µm dicken Kupferschicht nur ca. 0,1mm tiefe und 0,1-0,2mm breite Isolationsbahnen gefräst werden.

## Schnellläufer

Unter dem Strich lautet meine Empfehlung: Kaufen Sie eine vollständige Maschine mit Endschaltern, einem Motor mit ausreichend Kraft und Drehzahl (für das Isolationsfräsen min. 12.000 Upm, besser 20.000 Upm) sowie einer Einrichtung zur problemlosen Nullstellung der Z-Achse. Ein solches Gerät ist beispielsweise die *Genmitsu 3018-PROVer*. Mit 260mm × 155mm × 35mm ist der Verfahrensweg für unsere Zwecke ausreichend groß. Viele weitere Informationen über diese Maschine finden Sie im Wiki des Herstellers unter dem Link im Info-Kasten.

Mit der Genmitsu-Maschine haben Sie noch einen wesentlichen Vorteil: Sie wird mit der Software *Candle* ausgeliefert, die über eine *Heightmapping*-Funktion verfügt. Damit kann man die Oberfläche von Leiterplatten abtasten und Unebenheiten in die Z-Achse des Verfahrensweges einrechnen lassen. Ohne Heightmapping ist ein Isolationsfräsen nicht



Die mitgelieferte Steuerungssoftware für die Genmitsu 3018-PROVer beherrscht das zum Isolationsfräsen immer notwendige Heightmapping.

möglich, zumindest nicht auf unserer einfachen Maschine. Bei anderen Geräten probieren Sie das kostenlose Programm *Autoleveller* (siehe Link im Info-Kasten).

Genauso wichtig wie das Heightmapping, das Sie natürlich bei jeder Leiterplatte neu durchführen müssen, ist das einmalige Vermessen der X-, Y-, und Z-Achse. Es gibt Fertigungstoleranzen, die es notwendig machen, der Maschine mitzuteilen, welche Strecke der Fräskopf bei einem Vortrieb tatsächlich zurückgelegt hat. Wenn die Annahme der Maschine nicht mit der Wirklichkeit übereinstimmt, dann werden Sie kaum ein DIL-Gehäuse genau ausfräsen können. Für diese Prozedur möchte ich auf zwei Videos des Herstellers verweisen (siehe Link im Info-Kasten).

Bei dieser Maschine arbeitet im Controller ein ATmega 328 mit der bekannten Schrittmotor-Steuerungssoftware *GRBL*. Für die Anpassung dieser Software an die tatsächlichen Verhältnisse Ihrer Fräse gibt es diverse Parameter, die man auch über ein Terminal-Programm ändern kann. Auf Dauer werden Sie als Nutzer einer solchen Maschine nicht umhinkommen, sich damit zu beschäftigen.

## Eingespannt

Die Platine muss natürlich unverrückbar, möglichst eben und so eingespannt sein, dass die Bohrungen vorgenommen werden können, ohne dass der Bohrer die darunterliegende Nutenplatte beschädigt. Mit den mitgelieferten Spannpratzen sind diese Bedingungen

nur schwerlich zu erfüllen. Häufig wird die Aufgabe gelöst, indem eine MDF-Platte glatt gefräst wird und die Leiterplatte darauf mit doppelseitigem Klebeband oder etwas Klebstoff aus der Heißklebepistole an den vier Ecken befestigt wird. Ich habe mich für vier Halter aus dem 3D-Drucker entschieden (Pläne unter dem Link im Info-Kasten).

Bevor wir die Maschine starten, spannen wir eine Platine mit Hilfe der Halter ein. Überall verfügbar sind einseitig kupferkaschierte Platinen aus Epoxyd-Glasfaser (FR4) mit den Maßen 160mm × 100 mm, dem sogenannten Europaformat. Wie bereits angedeutet, empfiehlt es sich, ein paar Cent mehr für Qualitätsprodukte auszugeben. Dort ist die Wahrscheinlichkeit für windschiefe Produkte mit schwankender Dicke der Kupfereauflage deutlich geringer.

## Nullen zuerst

Als erste Aktion wird die Fräse in die Nullposition gefahren. Das ist dann erfolgreich, wenn der Fräskopf an das obere Ende der Z-Achse gelaufen ist und die Nutenplatte in die vordere rechte Ecke. Dann stehen die Maschinenkoordinaten auf 0/0/0. Der nächste Schritt ist, den Frässtichel einzusetzen. Achten Sie bei dessen Beschaffung auf den korrekten Schaftdurchmesser von 3,175 mm. Wenn Sie den Fräser einsetzen, schieben Sie ihn möglichst tief ein, lassen ihn aber noch so weit herausstehen, dass er die Platine gut erreicht. Mit dem Jog-Kreuz fahren Sie nun den Fräskopf

in die Nähe der linken unteren Ecke der Platine. Das ist der X/Y-Nullpunkt für diese Fräsung. Teilen Sie dies der Maschine durch einen Klick auf das entsprechende Symbol (ZERO XY) mit.

Nun suchen wir die Nullstellung der Z-Achse. Den Fräser fahren Sie dazu bis auf etwa 1cm an die Platine heran. Bei den folgenden Schritten bleibt der Fräsmotor ausgeschaltet. Nun befestigen Sie eine der beiden Krokodilklemmen des Tastwerkzeuges so an der Platine, dass elektrischer Kontakt besteht. Die andere Klemme kommt an den Fräser. Wenn Sie nun auf den Knopf Z Probe klicken, fährt der Fräskopf zweimal langsam herunter, bis Kontakt zur Platine besteht. Der Stromkreis ist geschlossen und der Fräser bleibt stehen. Klicken Sie nun auf das Symbol ZERO Z. Nun stehen auch die Arbeitskoordinaten auf 0/0/0. Die beiden Krokodilklemmen belassen wir an der Platine und dem Fräser.

## Landkarte

Für das notwendige Heightmapping müssen wir als ersten Schritt das Layout der Platine (Kupfer.nc) laden. Das Programm misst praktischerweise nicht die gesamte eingespannte Platine aus, sondern nur den für unser Layout relevanten Teil. In der rechten Spalte von Candle finden Sie einen Abschnitt Heightmap. Eventuell müssen Sie unter Service/Settings/Panels das Häkchen bei Heightmapping setzen. Klicken Sie nun auf den Button Create. In den darauf erscheinenden Heightmap settings haken Sie Show border ab und betätigen den Knopf Auto. Das Programm hat um das angezeigte Layout ein Rechteck gezogen.

Heightmap wird anschließend die Z-Achse innerhalb der Layoutfläche in gleichbleibenden Abständen abtasten und sich die Höhenunterschiede zur ersten Messung, dem Z-Achsen-Nullpunkt, merken. Diese Differenzen werden dann beim Fräsen berücksichtigt,

sodass Schwankungen in der Kupferdicke, Verwindung der Platine oder sonstige Unregelmäßigkeiten berücksichtigt werden.

Damit ist eine relativ gleichmäßige Eindringtiefe sichergestellt. Diese Abstände leiten sich aus den Ausmaßen der Layoutfläche ab. Es hat sich als guter Kompromiss zwischen Zeitaufwand und Qualität herausgestellt, diese Abstände als jeweils ein Zehntel der Höhe und Breite zu nehmen. Die jeweils nächste ganze Zahl tragen wir unter Probe Grid in die Felder X bzw. Y ein. In unserem Falle sind das 5 für X und für Y die Zahl 7. Jetzt klicken wir auf Probe. Behutsam wiederholt die Fräse den Tastvorgang und notiert die Abweichung vom Nullpunkt in den Feldern unterhalb des Layouts. Gleichzeitig spannt sich ein farbiges Netz über das Layout, das das abgetastete Gebirge auch bildlich darstellt.

Wenn nach einigen Minuten die 35 Tests abgearbeitet sind, müssen wir die Ergebnisse speichern. Dazu klicken wir zunächst auf Edit, die farbliche Überdeckung unseres Layouts verschwindet und wir drücken auf den Knopf Open (!). Im folgenden Dialog können wir unsere Daten zunächst abspeichern und anschließend sofort wieder öffnen. Sinnigerweise nenne ich diese Datei immer height.map, sie wird ja nur einmal benötigt. Dieser Name erscheint nun auch in dem bisher leeren Feld Map in der rechten Spalte und zeigt uns, dass die eben erhobenen Daten nun zur Verfügung stehen.

Damit sie beim Fräsvorgang auch tatsächlich berücksichtigt werden, dürfen wir nicht vergessen, das Häkchen bei Use Heightmap zu setzen. Der Fräser steht noch an der letzten Teststelle. Also müssen wir ihn an den Nullpunkt zurückfahren. Das geht am einfachsten in dem wir in das Feld Console ganz unten in der rechten Spalte den Befehl G0X0Y0 eingeben und mit Enter abschicken. Dieser G-Code veranlasst den Fräser, sich auf die Position X=0 und Y=0 zu begeben. Jetzt ist alles vorbereitet.

## Mutiger Knopfdruck

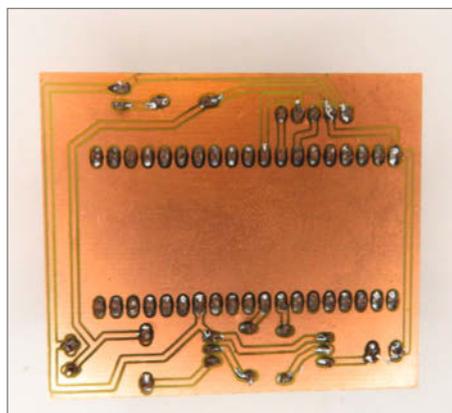
Den Fräsvorgang starten wir mit einem beherrzten Klick auf den Knopf Send am unteren Rand des Programmes. Der Motor läuft langsam hoch und schiebt sich an die erste Position, senkt sich und beginnt, die erste Leiterbahn zu fräsen. So wird das mit allen notwendigen Bahnen fortgesetzt, bis nach circa 10 Minuten eine sauber gefräste Platine vor uns liegt. Leider dürfen wir die Platine nicht aus der Halterung nehmen, um sie besser betrachten zu können. Denn sobald sich die Lage der Leiterplatte verändert, ist der Nullpunkt und das Ergebnis unseres Heightmapping-Vorganges ungültig. Also lassen wir die Leiterplatte eingespannt und entfernen den Staub mit Pinsel (oder einer alten Zahnbürste) und Staubsauger.

Nach der letzten Leiterbahn ist der Fräser auf den Nullpunkt gefahren und hat sich einige Zentimeter angehoben. Mit dem Jog-Werkzeug fahren wir den Fräser noch etwas höher, und zwar so, dass wir gefahrlos den Gravierstichel gegen den 1mm-Fräser austauschen können. Diesen verwenden wir auch zum Bohren.

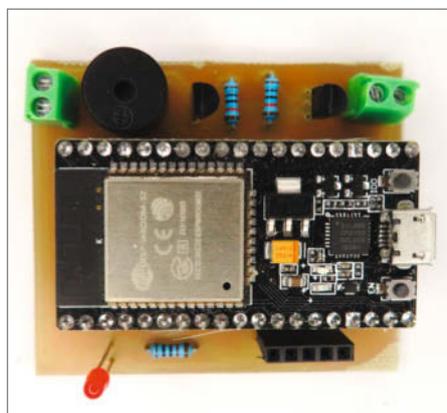
Um nun die „Bohr-Session“ starten zu können, laden wir die Datei Bohrung.nc. Da sich die Werkzeughöhe und damit der Z-Nullpunkt durch das Einsetzen des Bohrers verändert hat, müssen wir den eingangs vorgenommenen Tastvorgang wiederholen. Eine Krokodilklemme befindet sich noch an der Platine, die andere klemmen wir wieder an den Fräser, fahren bis rund 10mm an die Platine heran und drücken auf das Z Probe-Symbol. Nach dem Abtastvorgang stellen wir den Nullpunkt mit dem Symbol ZERO Z zurück. Die Arbeitskoordinaten stehen nun wieder auf 0/0/0. Damit wir an der Krokodilklemme noch viel Spaß haben, entfernen wir sie nun vom Fräser. Beim Bohren ist es nicht notwendig, aber – vielleicht auch, um uns daran zu gewöhnen – drücken wir beim Heightmap den Open-Knopf, laden wieder die Datei height.map und setzen bei Use Heightmap das Häkchen.

Das Bohren ist relativ schnell erledigt. Ein wenig saubermachen, und wir können den letzten Durchgang starten, das Herausfräsen der Platine. Hier läuft alles analog wie eben beim Bohren beschrieben, nur eben mit der Datei Umriss.nc. Der Rest ist dann schnell erledigt. Um das spätere Löten zu erleichtern und das leicht „anlaufende“ Kupfer zu schützen, sprühe ich Lötlack auf die Platine, lasse ihn trocknen und beginne dann mit dem Bestücken.

Die abgebildete Platine gehört übrigens zu einem Projekt, das wir als zweiten Teil des Artikels voraussichtlich in der nächsten Make-Ausgabe veröffentlichen: Einem WLAN-gesteuerten Lötöfen-Thermostaten zum Umbau eines Pizzaofens, mit dem sich SMD-Platinen auf einfache Weise im Reflow-Verfahren löten lassen. —cm



Gefräste Platine nach dem Bestücken: Achten Sie darauf, dass Kupferspäne nicht ungewollte Kurzschlüsse verursachen.



Mit einer richtigen Platine sieht unser Projekt viel professioneller aus als der Breadboard-Aufbau.

# Kanban Day

Kanban im Unternehmen –  
mehr als nur Projektmanagement

Jetzt  
anmelden!

22. Februar 2022 ONLINE

Der Kanban Day bietet einen fundierten, praxisorientierten Überblick, wie sich Kanban bei der Implementierung in Organisationen gewinnbringend einsetzen lässt, welche Metriken sinnvoll sind, um Veränderung und kontinuierliche Verbesserung zu verfolgen, und legt einen Schwerpunkt auf die Anwendung von Kanban als Managementmethode.

- ☑ Kanban – das Schweizer Messer zur Unternehmensverbesserung
- ☑ Warum Kanban gerade für Teams wichtig bleibt
- ☑ Flow in der Strategiefindung
- ☑ Kanban und Verantwortung: Leadership auf allen Ebenen
- ☑ Navi für Ihren Change – das Kanban Maturity Model
- ☑ Kanban-Metriken in der Praxis

Veranstalter

@ heise Developer

 dpunkt.verlag

in Kooperation mit

it-agile 

leanovate  
beyond product

kanban.inside-agile.de

# Gehäusebau mit FreeCAD, Teil 2

In unserer Artikelserie zeigen wir, wie Sie mit der kostenlosen Software FreeCAD Ihr eigenes Gehäuse für Einplatinenrechner wie den Raspberry Pi entwerfen. Diesmal geht es darum, die äußeren Maße des Gehäuses zu ermitteln. Für die konkrete Umsetzung gibt es ein Online-Video, das Sie Schritt für Schritt durch den Konstruktionsprozess führt.

von Matthias Mett



In diesem Artikel werden wir alle Maße ermitteln, die man zum Zeichnen des Grundkörpers des Gehäuses für den Raspberry Pi 4 in der kostenlosen 3D-Konstruktionssoftware FreeCAD benötigt. Im Online-Video (zu erreichen über den Link in der Kurzinfo) konstruieren wir dann Schritt für Schritt das Gehäuse anhand dieser Maßangaben in der Software. Die Aussparungen für die Anschlüsse, die Stege zum Befestigen der Platine und die Aussparungen für den Lüfter behandeln wir in den kommenden

Teilen unserer Artikelserie und weiteren Videos.

Die *Raspberry Pi Foundation* macht es uns einfach, die Maße für die Abstände der Anschlüsse auf der Platine zu ermitteln: Sie stellt auf ihrer Webseite eine technische Zeichnung mit den Maßangaben zum Herunterladen zur Verfügung. Diese können Sie im Bereich *Raspberry Pi 4 Model B specifications* unter *Raspberry Pi 4 Model B mechanical drawings* finden (siehe Link). Die technische Zeichnung gibt die Größe der Platine und

die Positionen der Anschlüsse an ①. Die Breite der einzelnen Bauteile messen wir hingegen selbst mit Hilfe eines Messschiebers.

Das Raspberry Pi Board hat laut Zeichnung eine Länge von 85mm und eine Breite von 56mm. Die USB-3.0-Anschlüsse haben auf der Außenseite einen Falz, der aus dem Gehäuse- rand herausragen soll. Dies ist ein guter Startpunkt für die erste Messung. Der Abstand zwischen der Platine und dem Falz beträgt laut Messschieber 2,75mm ②.

## Luft lassen

Um Schwankungen des Materials beim 3D-Druck auszugleichen und die Platine im Gehäuse ohne Druck platzieren zu können, lassen wir ihr innen 0,5mm auf jeder Seite Platz. Kleinere Materialverwerfungen könnte man auch noch mit Schleifpapier ausgleichen, falls die Platine sich dennoch nicht leicht montieren lassen sollte. Wir wollen zwecks Stabilität auf eine Wandstärke des Gehäuses von 2,25 bis 2,5mm kommen. Die Micro-HDMI-Anschlüsse ragen nicht ganz so weit aus der Platine und sind nach außen hin gespreizt ③. Probeweise stecken wir einen Stecker hinein, um zu sehen, wie groß der Abstand zwischen dessen Umhüllung und der Buchse ist. Beim HDMI-Adapter sind da noch gut 1,5mm Platz, genügend Luft, um dem Gehäuse die geplante Wandstärke von 2,5mm zu geben.

Der 3,5mm-Klinkenanschluss hingegen überragt die anderen Anschlüsse nach außen hin noch um ein Stück, würde aber auch nicht ganz exakt mit dem geplanten Gehäuse abschließen – um einen Stecker in die Buchse zu stecken, reicht es aber locker. So kommen wir jetzt auf die Innenmaße des Gehäuses von 86mm Länge und 57mm Breite und Außenmaße von 90,5mm Länge und 61,5mm Breite. Als Dicke der Bodenplatte wählen wir ebenfalls 2,5mm.

## Kurzinfo

- » Konstruieren eines einfachen Raspberry-Pi-Gehäuses (4, Model B) mit kostenloser Software 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 komplette 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



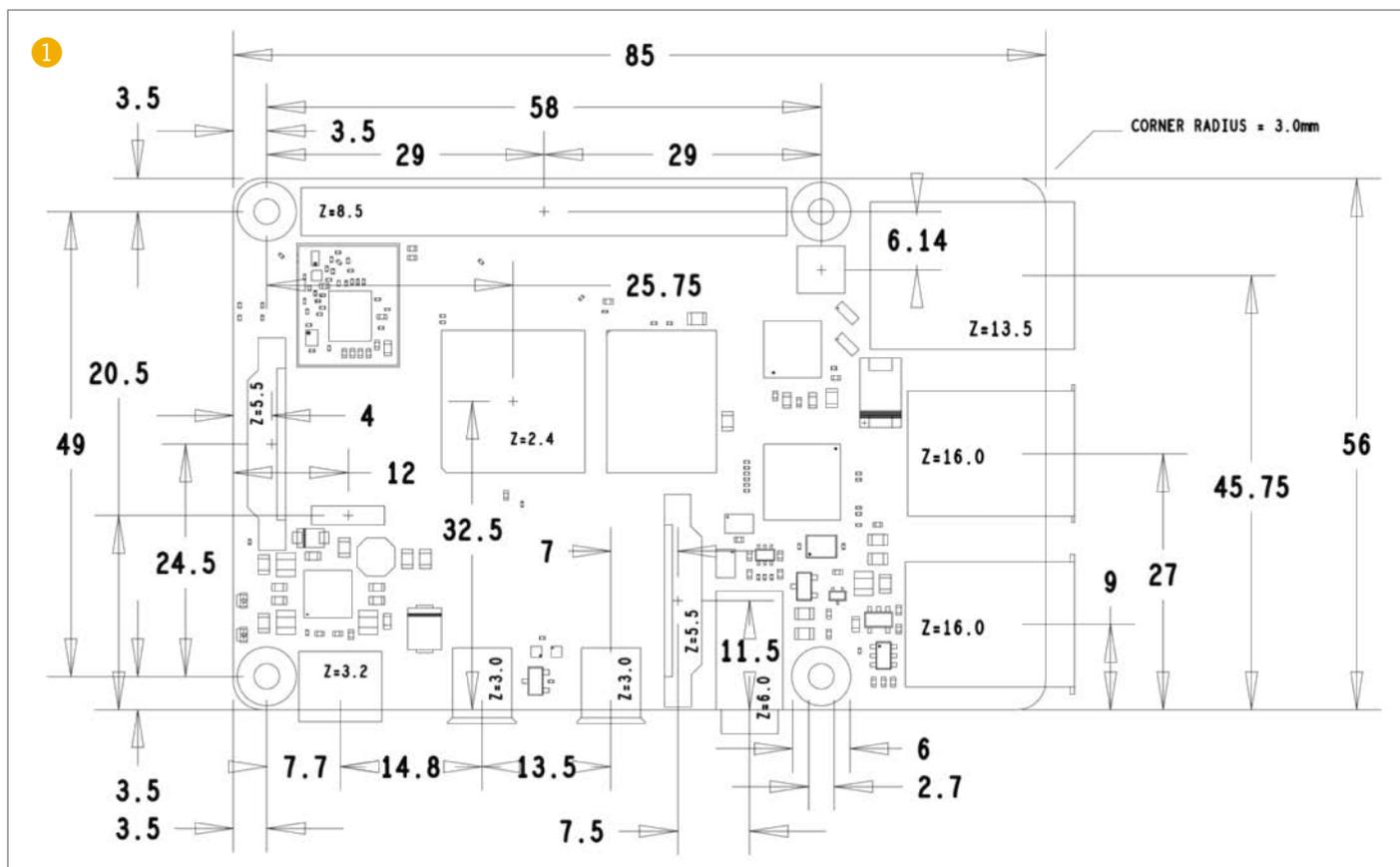
### 3D-Druck:

mit eigenem Drucker, im Fablab oder beim Dienstleister

## Mehr zum Thema

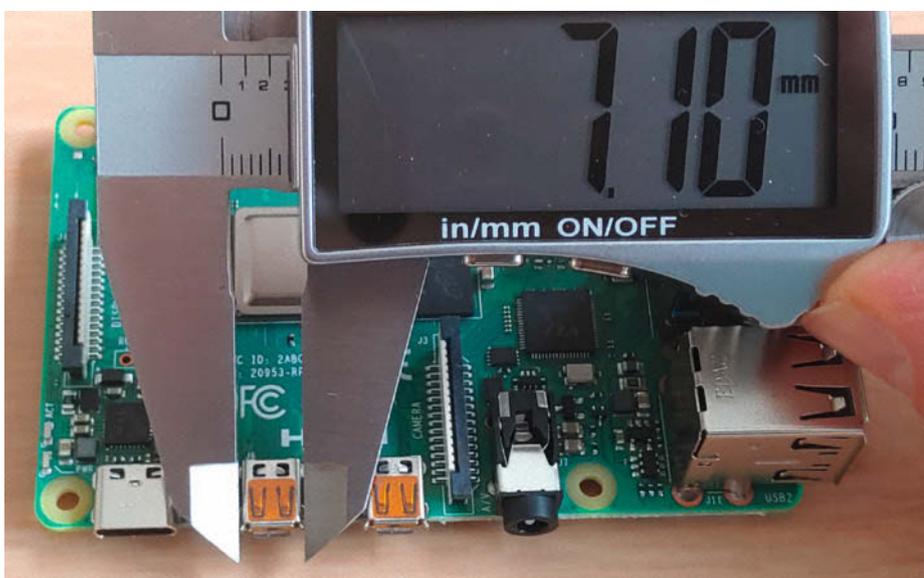
- » Matthias Mett, Gehäusebau mit FreeCAD, Teil 1, Make 5/21, S.116
- » Matthias Mett, 3D-Entwurf mit FreeCAD, Make 1/21, S. 128
- » Matthias Mett, Stempeln mit FreeCAD, Make 4/20, S. 106
- » Matthias Mett, Ring frei für FreeCAD, Make 2/20, S. 114
- » Billie Ruben, CAD-Designtipps für den 3D-Druck, Make 1/21, S. 138
- » Peter König, Gratis-CAD für Maker, Make 4/18, S. 19

Alles zum Artikel  
im Web unter  
[make-magazin.de/xad2](https://make-magazin.de/xad2)

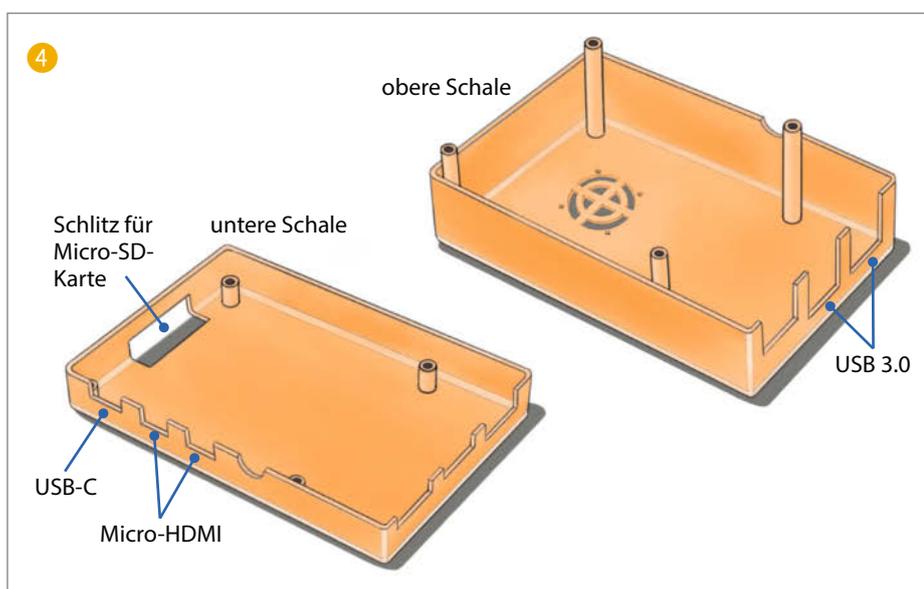




2 Messung der Abstände der Platine zum Falz des USB-3.0-Anschlusses.



3 Messung der Breite der Micro-HDMI-Anschlüsse, die nach außen hin gespreizt sind



## Höhenpeilung

Jetzt liegt der Grundriss fest, aber es sind noch die erforderlichen Höhen der unteren und der oberen Schale des Gehäuses zu ermitteln.

Die Platine soll so in der unteren Schale sitzen, dass die Micro-HDMI-Anschlüssen nur dort eine Aussparung brauchen, nicht oberhalb des Schnitts zwischen der oberen und der unteren Schale des Gehäuses. Diesen Schnitt kann man gut in der Designzeichnung des Gehäuses erkennen, die das Titelbild dieses Artikels bildet.

Die untere Schale muss keine großen nach unten aus der Platine ragenden Bauteile aufnehmen – wir messen auf deren Unterseite für Montagepins und den Micro-SD-Kartenhalter einen Raumbedarf von etwa 2mm und geben noch 3mm für die Luftzirkulation hinzu. Um davon ausgehend auf die gesamte Höhe der unteren Schale zu kommen, rechnen wir die Platinendicke samt der vollen Steckerhöhe der Micro-HDMI-Anschlüsse von aufgerundet 5mm hinzu, so muss weder für diese noch für die USB-C-Buchse zur Stromversorgung in der oberen Schale eine Aussparung vorgesehen werden, was die Konstruktion vereinfacht 4. Damit kämen wir auf eine gesamte Höhe der Unterschale von 10mm. Berücksichtigen wir noch die Dicke des Bodens von 2,5mm, kämen wir auf 12,5mm. Hier runden wir aber auf 12mm ab, damit wir einen geraden Wert bekommen, dann sind es halt nur noch 2,5mm Luft unter der Platine, reicht auch.

Für die Höhe der oberen Schale messen wir das größte Bauteil: die USB-Anschlüsse mit 15mm an den vorderen Falzen. Diese Bauteile sind so hoch, dass auch zusätzliche Kühlkörper auf dem Prozessor und den anderen Chips von der Höhe über die USB-Anschlüsse nicht herausragen. Der vorgesehene zusätzliche Lüfter in der oberen Gehäusehälfte soll zu diesen Kühlkörpern genügend Platz haben, daher lassen wir ihn nicht weiter als die Oberkante der USB-Ports in Richtung der Platine hineinragen. Aus diesem Grund rechnen wir einfach die Höhe des Lüfters von 7,8mm (aufgerundet 8mm) zu den 15mm der USB-Anschlüsse hinzu. Damit hätten wir eine Höhe von 23mm über Platinenoberfläche, wobei wir noch 2,5mm Deckendicke hinzunehmen und auf einen geraden Wert von 26mm aufrunden.

Somit hätten wir alle Maße zusammen, um die beiden Gehäusehälften in FreeCAD zeichnen zu können. Hierzu finden Sie eine Schritt-für-Schritt-Anleitung als Online-Video, Link siehe Kurzinfo. Dort zeigen wir Ihnen etwa, wie Sie mit dem FreeCAD-Werkzeug *Vertiefen* die Gehäusehälften erzeugen, ohne die Seitenwände einzeln zeichnen zu müssen. Die Aussparungen für die Anschlüsse sind dann Thema der nächsten Folge und vor allem des nächsten Videos.

—pek

# Hybrides Arbeiten mit Microsoft Teams

## In 4 Online-Trainings zum Teams-Experten

Unsere Experten erklären Ihnen Schritt für Schritt den Einsatz von Microsoft Teams in einer hybriden Arbeitswelt. Angefangen mit den wichtigsten Funktionen und Neuerungen der Plattform lernen Sie außerdem die Möglichkeiten bezüglich Telefonie, Meeting-Räume und Co. kennen.

### DIE TERMINE:

**19. Januar 2022**

Microsoft Teams in der Praxis

**26. Januar 2022**

Hybrides Arbeiten mit Microsoft Teams: Telefonie, Meeting-Räume und Co.

**9. Februar 2022**

Microsoft Teams und Apps im praktischen Einsatz

**16. Februar 2022**

Bessere Zusammenarbeit mit Microsoft Viva und Microsoft Teams

**Exklusiver Kombi-Preis: 499,-**

Einzelpreis: 169,-

**Jetzt Kombi-Rabatt sichern und über 150,- sparen!**

© Copyright 2021 heise.de/hybrides-arbeiten-mit-ms-teams/



# Tipps & Tricks

Schon gewusst? Manche Händler bieten 3D-Druck-Material als Rolle ohne Spule an. Mit der hier beschriebenen Eigenbau-spule zum Wechseln wird solches Filament im Handumdrehen gebrauchsfertig.

von Erhard Raub

Wieder vier leere Filamentspulen im Regal hinter dem 3-D Drucker. Nutzloser Plastikmüll. Kann man das nicht vermeiden? Der Händler meines Vertrauens (siehe Link am Ende des Artikels) bietet neben herkömmlichem Filament auf Spule auch *Refill-Filament* an – gerollte Filamentstränge ohne Spule ❶. Doch wie bekommt man den Faden ohne Spule auf den Drucker? Nachdem ich testweise ein paar Möglichkeiten durchgespielt habe, war mir klar, dass es mit „Umwickeln“ – also neues Filament auf alte Spule wickeln – nicht funktioniert. Das dauert ewig, das Filament kann dabei geknickt oder beschädigt werden und die Gefahr, ein nicht entwirrbares Filamentknäuel zu produzieren, ist einfach zu groß.

Eine andere, eine zerlegbare Spule muss her. Bei zerlegbaren Spulen fällt das Wickeln weg und die frische Filamentrolle kann direkt eingebaut werden – und immer wieder aufs Neue. Diese neue Spule wollte ich bewusst aus Holzresten bauen. Davon hat man

normalerweise genug im Bastelkeller rumliegen.

## Spule bauen

Als Werkzeug braucht man Holzleim, eine Laubsäge, Klebeband und drei Holzschrauben, etwa 15mm lang.

Das verwendete Material sind Sperrholzreste in Dicken zwischen 3mm und 9mm. Die Bauzeit beträgt etwa zwei Stunden – ohne Trocknungszeit – je nach persönlicher Geschicklichkeit mit der Laubsäge. Die Bilder hier im Heft zeigen die wichtigsten Arbeitsschritte, noch mehr Fotos der einzelnen Bau-stufen gibt es online (siehe Link am Ende des Artikels).

Als Erstes werden die Ringe für den Spulenkern aufgezeichnet und ausgesägt – Innendurchmesser 4cm und Außendurchmesser 7cm. Es sollten zwei möglichst gleich große Ringe aus stabilem Sperrholz (mindestens 5mm dick) entstehen ❷. An diesen Holz-

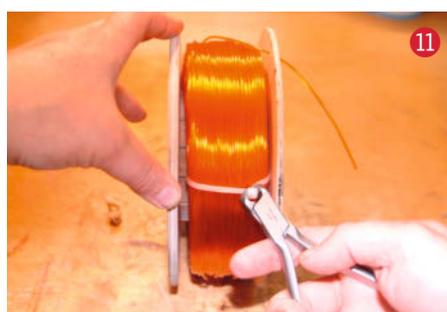
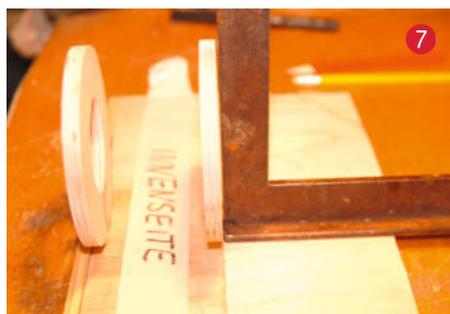
ringen werden später mal die Außenseiten angeschraubt.

Um die Ringe herum wird der Kernmantel aufgebaut. Das geht am besten mit einer Schablone. Zwei Leisten werden dazu auf einer Grundplatte genau im rechten Winkel zueinander aufgeklebt ❸. Innerhalb dieses Winkels können die fertig vorbereiteten Einzelteile des Kernmantels arrangiert und eingepasst werden.

## Rechnen oder messen

Ehe es an die Herstellung der Einzelteile geht, ist allerdings etwas Überlegung angesagt: Aus wie vielen Teilen soll der Kernmantel später einmal bestehen und wie groß sollen die jeweils sein? Um diese Fragen zu beantworten, braucht man entweder die alten Griechen und etwas Mathematik:  $\text{Kreisumfang} = 2 \times \pi \times r$  ( $r$ =Radius) – oder einen alten Handwerkertrick: Man nimmt den Ring und markiert eine Stelle an seiner Außenseite und eine Startstelle auf dem Tisch ❹. Jetzt rollt man selbigen Ring





vorsichtig zwischen zwei Fingern von der Startstelle weg in eine Richtung, bis die Ringmarkierung wieder auf den Tisch zeigt – das ist der Endpunkt. Die Entfernung zwischen Start- und Endpunkt entspricht dem abgerollten Kreisumfang – wir haben eine einfache „Abwicklung“ gemacht, ganz ohne Mathe.

Bei mir ergab das einen Umfang von 18,5 cm, und da ich 14 Segmente verbauen wollte, brauchte ich 14 Sperrholzplättchen, die jeweils 1,3 cm breit und 6,0 cm lang waren. Jetzt kommt die Laubsäge zum Einsatz. 14 Sperrholzplättchen werden ausgesägt, an den Kanten mit Schleifpapier abgerundet und in die vorbereitete Schablone eingesetzt **5**. Je genauer diese Plättchen in die rechteckige Schablone eingepasst werden, desto einfacher sind die folgenden Arbeitsschritte.

### Kern-Montage

Sitzen alle Teile präzise am vorgesehenen Ort, werden sie mit einem Streifen Malerkrepp miteinander verklebt **6**. Achtung: Es gibt große Qualitätsunterschiede bei Klebbändern, bitte ein Band mit ausreichender Klebekraft verwenden! Sonst lösen sich später die Teile und es wird nervig.

Nun werden die am Anfang vorbereiteten Ringe verbaut. Beide Ringe werden einander gegenüber auf die zwei Enden desselben Sperrholzplättchens geklebt **7**. Sie werden mit Hilfe von Winkeln genau ausgerichtet und trocken gelassen.

Sind diese Verklebungen durchgetrocknet, können alle restlichen Teile des Kernmantels verklebt werden. Dafür streicht man an der entsprechenden Stelle Klebstoff auf, fasst das überstehende Kreppband mit zwei Fingern

und wickelt alles zusammen in einem Zug um die Ringe **8**. Das Klebeband verbleibt im Inneren des Kerns und kann später, nach dem Aushärten, abgezogen werden **9**. Alles mit Klebeband von außen fixieren und zum Trocknen stehen lassen.

Nachdem auch diese Verklebungen ausgehärtet sind, wird der jetzt fertige Kern glatt geschliffen. Falls noch eine kleine Lücke im Mantel klafft, wie auf Bild **9** zu sehen, kann diese mit einem passend zugeschnittenen Sperrholzteilchen verschlossen werden.

Für die Seitenteile habe ich 3mm starkes Sperrholz verwendet. Das ist leicht und trotzdem ausreichend fest, weil in diesem Bereich nicht so stabil gebaut werden muss wie beim Kern. Man sägt zwei identische Holzscheiben mit Loch im Zentrum, Maße diesmal: Kernlochdurchmesser 4 cm, Außendurchmesser 20cm. Die Umriss werden mit einem Zirkel auf ein Holzbrett aufgezeichnet und dann ausgesägt. Danach werden die Sägekanten mit Schleifpapier abgeschmirligt und geglättet, damit

später das Filament nicht an rauen Stellen hängen bleibt.

Als Nächstes wird eines der Seitenteile fest mit dem Kern verklebt **10**. Danach wird das gelieferte Refill-Filament über den einen Spulenteil geschoben und das zweite Seitenteil mit drei Schrauben auf den Kern geschraubt. Erst jetzt werden die Kabelbinder, mit denen die Filamentrolle in Form gehalten wird, durchtrennt und seitlich herausgezogen **11**. Damit ist die Spule fertig zum Einsatz am 3-D-Drucker.

Ein kleiner Tipp zum Schluss: Bohrt man in eines der Seitenteile ein Loch mit 4mm Durchmesser, dann kann man das lose Ende des Filaments hier durchstecken **12**. So aufgeräumt findet man es beim nächsten Materialwechsel sofort. —pek

↓ Alles zum Artikel im Web unter [make-magazin.de/x9a2](http://make-magazin.de/x9a2)

### Machen Sie mit!

Kennen Sie auch einen raffinierten Trick? Wissen Sie, wie man etwas besonders einfach macht? Wie man ein bekanntes Werkzeug oder Material auf verblüffende Weise noch nutzen kann? Dann schicken Sie uns Ihren Tipp – gleichgültig aus welchem Bereich (zum Beispiel Raspberry, Arduino, 3D-Druck, Elektronik, Platinenherstellung, Lasercutting, Upcycling ...).

Wenn wir Ihren Tipp veröffentlichen, bekommen Sie das bei Make übliche Autohonorar. Schreiben Sie uns dazu einen Text, der ungefähr eine Heftseite füllt und legen Sie selbst angefertigte Bilder bei. Senden Sie Ihren Tipp mit der Betreffzeile *Lesertipp* an:

[mail@make-magazin.de](mailto:mail@make-magazin.de)

# Von den alten Römern bis zur Gegenwart – All about History!



## Tauchen Sie ein in vergangene Zeiten:

All About History beleuchtet alle zwei Monate die großen Epochen und Ereignisse der Weltgeschichte und gräbt alte Schätze aus. Erfahren Sie alles über die verschiedenen Zeitalter von der Antike bis zur Gegenwart, die unsere Welt geprägt haben.



**35%  
RABATT**

### Kennenlern-Angebot: 2 Ausgaben testen

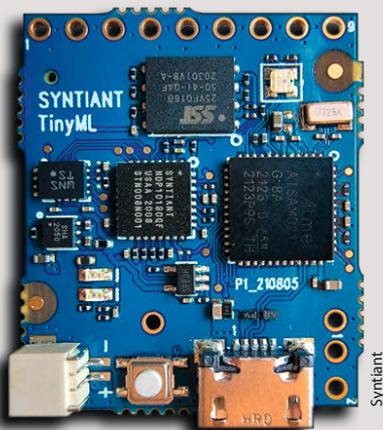
- 2 aktuelle Ausgaben als Heft
- 35% Kennenlern-Rabatt
- 3 Tage vor dem Handel erhältlich
- erscheint 6 x im Jahr

Jetzt bestellen:

[www.emedia.de/history-mini](http://www.emedia.de/history-mini)

## Syntiant TinyML

### Arduino-kompatibler neuronaler Entscheidungsprozessor



Viele Entscheidungsprozesse lassen sich mit einem neuronalen Netz leichter und schneller erledigen, sei es das Erkennen von Sprache oder das Reagieren auf typische Sensorwerte. Beim *TinyML*-Board von *Syntiant* erledigt dies der neuronale Prozessor NDP101, der einem SAMD21 von *Microchip* (Low-Power ARM Cortex M0, 48MHz) zur Seite steht. Das Gespann soll im Betrieb nur durchschnittlich 100µW verbrauchen, selbst bei Spracherkennung soll es mit 140µW auskommen.

Der SAMD21 hat Zugriff auf 256 KByte Flash-ROM und 32 KByte SRAM, zusätzlich sind noch reichlich Speicher, Sensoren und Peripherie auf dem Board verbaut: 2 MByte Serial Flash, eine RGB-LED, ein microSD-Kartenslot, ein Sechsen-Achsen-Beschleunigungssensor BMI160 und ein MEMS-Mikrofon SPH0651LM4H. Der NDP101 ist über SPI an die CPU angebunden, die Schnittstelle steht auch auf den fünf I/O-Pins zur Verfügung.

Das nur 24mm × 28mm große Board ist mit der *Arduino MKR Zero*-Spezifikation kompatibel und kann mit der *Arduino-IDE* programmiert werden. Die Unterstützung für den NDP101 steuert der *AI-Spezialist Edge Impulse* bei; die mitgelieferte Beispiel-Applikation steuert zum Beispiel ein RC-Modellauto über Sprachbefehle. —cm

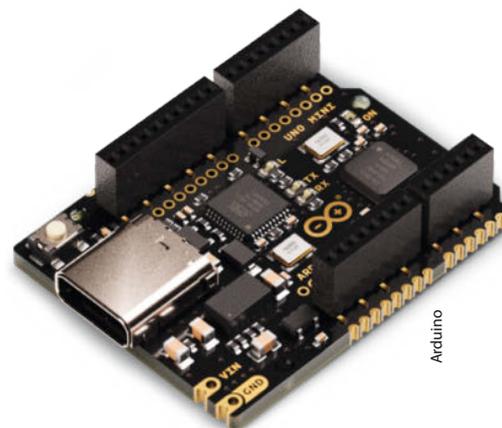
Hersteller	Syntiant
Vertrieb	u.a. DigiKey
Preis	35 US-\$

## Arduino UNO mini

### Mikrocontroller-Board für Sammler und Fans

Das klassische Maker-Board *Arduino UNO* ist seit seinem ersten Erscheinen im Jahr 2010 laut Hersteller in 10 Millionen Exemplaren verkauft worden – und das feiert man mit einer Sonderausgabe des Bastel-Boards namens *UNO Mini*. Die Hardware soll dabei komplett kompatibel zum regulären *UNO* sein (den man inzwischen in *Rev 3* für 17 Euro bekommt), allerdings hat die Platine des Mini mit 34,2mm × 26,7mm nur ein Viertel der Fläche des großen *UNO*. Die Hohlbuchse für die Stromversorgung fehlt, dafür ist statt der Mikro-USB-Buchse für Datenübertragung und Energieversorgung eine zeitgemäße USB-C-Buchse an Bord. Auch wenn das Board voll funktionsfähig ist, ist es mit seiner edlen Aufmachung in Schwarz und Gold und mit der aufwändigen Verpackung samt fortlaufender Nummerierung und signiertem Zertifikat der *Arduino*-Prominenz wohl vor allem als Sammlerstück gedacht.

Der *Arduino UNO Mini* war bei Redaktionsschluss ausschließlich in dieser limitierten Edition vorbestellbar, die in der normalen Produktliste auf der *Arduino*-Webseite gar nicht auftauchte, sondern nur über einen Blog-Bei-



trag zu erreichen war (siehe Link in der Tabelle). Wie groß die limitierte Auflage ist und ob es den *UNO Mini* vielleicht später auch mal regulär, unlimitiert und günstiger gibt, war bei Drucklegung dieses Hefts noch offen. —pek

Hersteller	Arduino
URL	make-magazin.de/xmvj
Preis	40 €

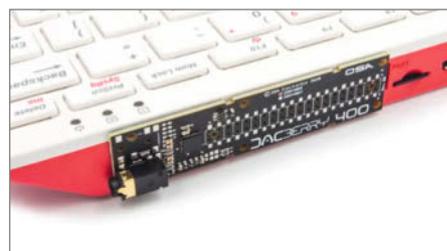
## DACberry 400

### Aufsteck-Soundkarte für den Raspberry Pi 400

Obwohl der *Raspberry Pi 400* gerne als *Komplett-Computer* bezeichnet wird, fehlt ihm doch der Sound-Anschluss. Die Tonausgabe musste bislang entweder über einen HDMI-Monitor mit eingebautem Lautsprecher oder über eine USB-Soundkarte erfolgen. Mit dem *DACberry 400* gibt es jetzt eine Soundkarte zum Aufstecken auf den GPIO-Port – und zwar in zwei Ausführungen: Der *DACberry 400S* besitzt lediglich einen Sound-Ausgang in Form einer 3,5mm-Stereo-Klinkenbuchse, die demnächst lieferbare Variante *DACberry 400M* soll zusätzlich auch einen Stereo-Eingang mitbringen.

Auf dem Board, das mit dem Raspi per I<sup>2</sup>S-Port kommuniziert, arbeitet ein DAC-Wandler mit einer Samplingrate von 96KHz bei 32 Bit Auflösung. Der Rauschabstand soll dadurch 102dB betragen. Der ADC-Wandler der M-Variante soll 92dB Rauschabstand bieten.

Ein eigener DSP (Digitaler Signal-Prozessor) zusammen mit dem I<sup>2</sup>S-Port soll den Prozessor des Raspi entlasten, sodass bei Sound-Aus- und -Eingabe mehr Rechenleistung übrig bleibt. Daneben bringt der DSP auch Zusatz-



funktionen wie einen Equalizer mit. Der *DACberry 400S*, der auch mit den anderen *Raspberry*-Versionen mit 40poligem Port sowie *Odroid*- und *BananaPi*-Boards zusammenarbeiten soll, kostet mit derzeit etwa 20 Euro allerdings deutlich mehr als eine USB-Soundkarte. Zudem behindert die Platine die Nutzung des GPIO-Ports. —hgb

Hersteller	OSA Electronics
Vertrieb	u.a. RS Components
Preis	rund 20 €

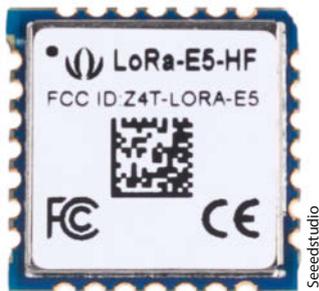
# LoRa-E5

## Funkmodul mit Stromspar-STM32

Wenn es um zuverlässige Funk-Kommunikation unabhängig von Wireless-Netzen geht, ist der LoRa-Standard eine gute Wahl: Er ermöglicht eine zuverlässige Kommunikation auch über einige Kilometer hinweg. Das nur etwa 12mm x 12mm große Modul LoRa-E5 hat Seeedstudio nach Industriestandards gefertigt, es basiert auf einem 48MHz schnellen ARM-Prozessor STM32WLE5JC von STMicroelectronics, der neben einem extrem stromsparenden ARM-Cortex-M4-Kern zusätzlich die IP des SX126x-Funkchips von Semtech enthält. Im Empfängerbetrieb nimmt das Modul weniger als 7mA auf, in Pausen sind es nur 2,1µA. Beim Senden können bei höchster Leistung kurze Stromspitzen bis 110mA auftreten.

Der STM32WLE5 beherrscht neben LoRa die Modulationsarten GFSK, GMSK und BPSK. Als I/O herausgeführt sind 3 UARTs, I<sup>2</sup>C, ein

ADC-Kanal mit 12 Bit, SPI und sechs universelle Port-Pins. Der chinesische Hersteller betont, dass das Modul nach FCC und CE (EU868/US915) zertifiziert ist. Es wird mit einer AT-Command-Firmware ausgeliefert; wer einen STM32cube-Programmer besitzt, kann über die STM32-Toolchain auch eigene Programme entwickeln. Wie bei Seeedstudio üblich gibt es eine gute Dokumentation und Code-Beispiele, hier in Form eines eigenen LoRa-E5-Wikis (siehe Link). —cm



► [make-magazin.de/xmvj](http://make-magazin.de/xmvj)

Hersteller	Seeedstudio
URL	seeedstudio.com/LoRa-E5-Wireless-Module-p-4745.html
Preis	9,90 US-\$

# Loris-Base

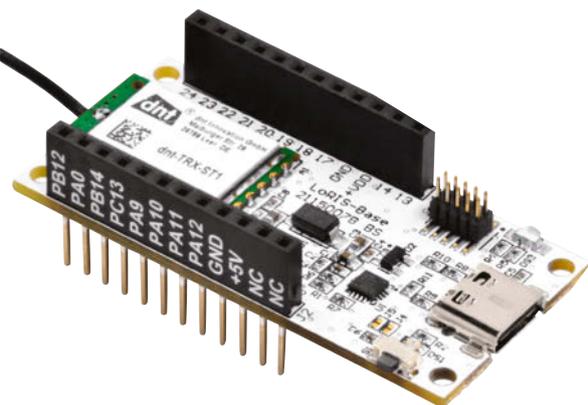
## Experimentierplattform für LoRaWAN

Einen günstigen Einstieg in erste Experimente mit dem sparsamen Weitfunkstandard LoRaWAN bietet das Loris-Base-Board von ELV. Das Board kommt im Feather-Formfaktor mit Stiftleisten fürs Breadboard. Kern der Platine ist der auf dieser Seite ebenfalls beschriebene STM32WL. 16 GPIOs sind auf die Stiftleisten gelegt, über die zusätzliche Module gesteuert werden können.

In der Standardausführung kann das Basis-Modul zwei festgelegte Eingänge kontrollieren und deren Zustände an das The Things Network (TTN) weiterleiten. Von dort kann

man Daten dann per Internet abholen und auswerten. ELV stellt für TTN sogenannte Payload-Parser zur Verfügung, die das vereinfachen sollen. Anders herum lässt sich per TTN ein Ausgang auf dem Modul aus der Ferne schalten.

Das Board lässt sich im Prinzip zwar frei programmieren, allerdings nur mit Entwicklungsumgebungen für STM32-Controller wie die STM32CubeIDE und einem passenden Programmieradapter. Arduino-Anhänger bleiben bei der Entwicklung von eigenen Projekten leider (noch) außen vor. Um das Board andere Funktionen übernehmen zu lassen, bietet ELV derzeit Aufsteckmodule für Temperatur- und Feuchtigkeitsmessung sowie ein Kontakt-Interface für Schalter und ähnliches an. Zum Betrieb muss man verschiedene, von ELV vorgefertigte Firmwares per USB auf dem Base-Board installieren, die zum kostenlosen Download bereitliegen. —dab



Hersteller	ELV
URL	de.elv.com
Preis	19,95 €

# Valve-Actuator

## DIY-Fußbodenheizungscontroller fürs Smart Home



Will man eine Fußbodenheizung per Smart Home steuern, muss man deren meist mit 230V betriebenen Ventile übers Netzwerk ansteuern, wofür in der Regel ein zusätzlicher und meist mehrere hundert Euro teurer Controller nötig ist. Allerdings hat der rumänische Konstrukteur Voltlog auf Github die Pläne für einen solchen Controller vorgestellt, der bis zu zehn Ventile einer Fußbodenheizung ansteuern kann. Das Gerät bietet er fertig für 99 US-Dollar auf Tindie an, wo es derzeit aber ausverkauft ist. Kein Problem aber für Maker, denn es kann durch die auf Github veröffentlichten Platinenlayout-Dateien (im KiCAD-Format) auch selbst gebaut werden. Drinnen arbeitet ein ESP32; der kann mittels der Firmware Tasmota und MQTT Befehle übers WLAN entgegennehmen beziehungsweise Daten an Home Assistant oder andere Smart-Home-Server liefern.

Auf dem Board sitzt außerdem ein 1-Wire-Port, an den zum Beispiel Temperatur-Sensoren angeschlossen werden können. Versorgt wird der Ventil-Controller mittels 230V. Batterien sind also nicht notwendig. Die Ausgänge für die Ventile werden mittels Triacs geschaltet. Die Programmierung erfolgt nur beim ersten Mal per Kabel. Danach ist die Firmware OTA-fähig: Updates werden also drahtlos (over the air) via WLAN überspielt. —hgb

Konstrukteur	Voltlog
URL	github.com/voltlog/Valve-Actuator
Preis	99 US-\$ auf Tindie

**Ausprobiert**  
— von Make: —

**Ausprobiert**  
— von Make: —

## PicoMite

### Basic-System für den Raspberry Pico



Für den *Raspberry Pico* gibt es mit *PicoMite* einen kompletten BASIC-Interpreter für *MMBasic* mit breiter Hardwareunterstützung, Grafikbefehlen und integriertem PIO-Assembler. Der *MMBasic*-Editor läuft dabei (inklusive Editor) direkt auf dem *Pico*. Weiterhin bietet er Fließkommazahlen, 64-Bit-Integerzahlen, Strings, lange Variablenamen, Arrays, Subroutinen und Funktionen. Der Interpreter ist weitgehend mit *Microsoft Basic* kompatibel, allerdings um moderne strukturierte Programmelemente erweitert und folgt dem ANSI-Standard für BASIC (X3.113-1987) bzw. ISO/IEC 10279:1991.

*MMBasic* wurde ursprünglich für den *Maxmite*-Retro-Computer entwickelt (siehe *Make* 5/20, S.24). Im Gegensatz zu allen anderen *MMBasic*-Versionen steht das *PicoMite*-System unter einer Open-Source-Lizenz. *PicoMite* unterstützt von Haus aus schon eine beachtliche Anzahl von anschließbarer oder auf dem *Pico* vorhandener Hardware. Einen ausführlicheren Artikel dazu und zur Installation lesen Sie online (siehe Link).

Im Gegensatz zu Lösungen wie ESP32-Boards mit VGA- und PS/2-Anschlüssen fehlt beim *PicoMite* das (Home-)Computer-Feeling zwar etwas, allerdings ist das System auch eher dazu gedacht, später als Mikrocontroller zu arbeiten. —caw

► [make-magazin.de/xmvj](http://make-magazin.de/xmvj)

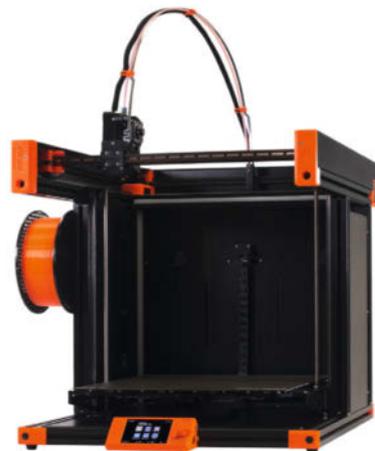
**Programmierer** Geoff Graham, Peter Mather  
**URL** [github.com/UKTailwind/PicoMite](https://github.com/UKTailwind/PicoMite)  
**Preis** kostenlos

## Prusa XL

### Großer FDM-Drucker mit bis zu fünf Extrudern

Bisher waren die Modelle der Reihe *Original Prusa i3 MK* die größten FDM-Drucker von *Prusa Research*. Mit dem jetzt vorgestellten *Prusa XL* wächst der Bauraum allerdings auf satte 36cm in allen drei Dimensionen. Damit dabei eine möglichst hohe Druckqualität herauskommt, hat der Hersteller seine bisherige Standardkonfiguration mit horizontal bewegtem Druckbett gegen eine klassische *XY-Core*-Konstruktion eingetauscht, bei der sich das Bett nur entlang der Z-Achse bewegt.

Auch sonst bringt der *Prusa XL* viele technische Neuerungen mit, etwa einen komplett neu entwickelten Extruder, eine Druckdüse, die gleichzeitig als Z-Achsen-Sensor arbeitet, ein segmentiertes Heizbett zur Einsparung von Heizenergie bei kleinen Objekten und nicht zuletzt einen optionalen Werkzeugwechsler, der bis zu fünf Extruder für den Multi-Material-Druck unterstützt. Diese und noch mehr Details stellt der Hersteller ausführlich in einem Video und einem Blogpost vor (siehe Link in der Tabelle). Die Maschine soll in der zweiten Jahreshälfte 2022 ausgeliefert werden, gegen eine Anzahlung kann



Prusa Research

man sich aber jetzt schon auf die Bestellliste setzen lassen. Entscheidet man sich später doch gegen den Kauf, soll es das Geld zurückgeben. —pek

<b>Hersteller</b>	Prusa Research
<b>URL</b>	<a href="http://make-magazin.de/xmvj">make-magazin.de/xmvj</a>
<b>Preis</b>	ab 2099 € (1 Extruder, teilmontiert) bis 4199 € (5 Extruder, fertig montiert)

## Creality Ender Pro 5

### Günstiger XY-Core-3D-Drucker

Der *Ender 5 Pro* ist ein *XY-Core*-3D-Drucker, das bedeutet, dass die Bauplattform entlang der Z-Achse bewegt wird, während sich der Kopf in einer festen horizontalen Ebene in X- und Y-Richtung bewegt. Die magnetische Dauerdruckplatte ist selbsthaftend und so schnell bequem abnehmbar; auf ihr finden Objekte mit bis zu 220mm x 220mm Grundfläche Platz. Die maximale Bauhöhe beträgt 300mm.

Die *Pro*-Version des *Ender 5* kostet im Vergleich zur Standard-Version einen Zuschlag von 60 bis 80 Euro. Die Upgrades bestehen aus *Silent*-Mainboard mit thermischem Durchgangsschutz, dazu passende leise Schrittmotoren-Treiber, Vollmetall-Extruder und einem hochwertigeren Schlauch für die Filamentzuführung des Bowden-Extruders. Zur Bedienung gibt es ein Monochrom-LCD und Wahlrad. Die maximale Bett-Temperatur wird mit 135°C angegeben, der Extruder soll 260°C erreichen.

Das Gerät kommt als Bausatz, der aber einsteigerfreundlich aus nur wenigen vormontierten Komponenten besteht, die verschraubt und zusammengesteckt werden – damit waren die Kollegen von *TechStage* in ihrem Test in weniger als 40 Minuten fertig. Ihr Fazit:

**Ausprobiert**  
— von Tech-  
Stage



Mit dem *Ender 5 Pro* bekommt man zwar einen guten FDM-Drucker zum günstigen Preis, aber das nur einseitig befestigte Druckbett führt ihrer Erfahrung nach zu Vibrationen, was wiederum die Vorteile der *XY-Core*-Bauweise zunichte macht. Den ausführlichen Testbericht lesen Sie online. —pek

► [make-magazin.de/xmvj](http://make-magazin.de/xmvj)

<b>Hersteller</b>	Creality
<b>URL</b>	<a href="http://creality.com">creality.com</a>
<b>Preis</b>	ab ca. 250 €

# Halot One und Halot Lite

## Schnelle Resin-3D-Drucker von Creality

Der *Halot One* wird fast betriebsfertig geliefert: Nach dem Auspacken und Entfernen der Transportsicherungen wird lediglich die Druckplattform aufgeschraubt und gelevelt, die Resin-Wanne aufgesetzt und fixiert. Nach Füllen des Tanks mit Resin kann der erste Test starten – sofern man das Material zur Hand hat, denn anders als andere Firmen legt der Hersteller weder eine erste Portion Harz noch die übliche Schutzausrüstung wie Einweghandschuhe bei.

Dafür ist der Preis mit rund 200 Euro recht moderat und im Test von *TechStage* überzeugte der Drucker durch hohe Geschwindigkeit und sehr gute Druckqualität. Die Kollegen führen dies auf die Tatsache zurück, dass im Gerät ein monochromes 2K-Display (5,96 Zoll) mit einer Auflösung von 1620 × 2560 für die

Belichtung sorgt. Dieses Schwarz-Weiß-Display schluckt weniger UV-Licht als RGB-Displays und erlaubt dadurch erheblich kürzere Belichtungszeiten und ein höheres Drucktempo. Der Bauraum misst 127mm × 80mm × 160mm. Einen ausführlichen Test lesen Sie online (siehe Link).

Der *Halot One* ist dabei nicht mit dem größeren *Halot Lite* zu verwechseln, den der Hersteller Creality kürzlich neu vorgestellt hat. Bei dem dürfen die Objekte bis zu 192mm × 120mm × 200mm groß werden; das 8,9 Zoll

große und 3840 × 2400 Rasterpunkte auflösende Belichtungs-LCD arbeitet mit einer vom Hersteller selbst entwickelten UV-Lichtquelle, die eine sehr gleichmäßige Ausleuchtung gewährleisten soll. Die Druckgeschwindigkeit soll 1 bis 4 Sekunden pro Schicht bei einer Schichtdicke von 0,01 bis 0,2mm betragen – das liegt in einem ähnlichen Bereich wie beim *Halot One*, bei dem die Kollegen von *TechStage* im Test mit einer Belichtungszeit von 2,1 Sekunden für 0,05mm dünne Schichten erfolgreich waren. —pek



► [make-magazin.de/xmvj](https://make-magazin.de/xmvj)

Hersteller	Creality
URL	creality.com
Preis	rund 200 € (One), ab 600 € (Lite)

# Prusa SL1S Speed

## Schneller Resin-Drucker

Mit dem *SL1S Speed* hat *Prusa Research* die überarbeitete Version seines ersten Resin-3D-Druckers *SL1* (siehe *Make* 1/20, S. 120) auf den Markt gebracht und den Vorgänger gleich aus dem Programm genommen. Der Hersteller hat uns ein Testgerät geschickt. Auspacken und Inbetriebnahme haben wir in einem Video ausführlich dokumentiert (siehe Link). Etwas stutzig machte uns der direkte Vergleich der neuen Maschine mit dem Vorgänger, da sich beide äußerlich kaum unterscheiden. Die neue aber angeblich einen um 25% größeren Bauraum aufweist. Der Zuwachs versteckt sich allerdings komplett in der leicht gewachsenen Grundfläche von Display und Drucktisch (jetzt

127mm × 80mm statt bisher 120mm × 68mm), die maximale Modellhöhe von 150mm ist geblieben. Leider ebenfalls geblieben ist der Umstand, dass weiterhin maximal hohe Drucke nicht in die (an sich sehr praktische) Nachhärtungs- und Waschmaschine *CW1S* passen, die es zusätzlich zu kaufen gibt – bei der ist nämlich bei rund 130mm Modellhöhe Schluss.

Überzeugend ist hingegen das Tempo, das der neue Drucker schon im Namen führt. Dank monochromen LCD für die Belichtung (auf das der Hersteller 2000 Stunden Garantie gibt), dauert das Aushärten pro Schicht deutlich kürzer als bisher, sodass sich die Druckzeit vergleichbarer Objekte gegenüber dem Vorgänger mehr als halbiert, bei gleichbleibender Qualität. Mit einem 15cm hohen Turm aus 0,5mm dünnen Schichten ist der *SL1S Speed* nach 3 Stunden und 48 Minuten fertig, der *SL1* brauchte dafür noch 10 Stunden und 28 Minuten. Einen ausführlichen Bericht mit Videos und Testdruck-Fotos lesen Sie online. —pek



► [make-magazin.de/xmvj](https://make-magazin.de/xmvj)



Hersteller	Prusa Research
URL	prusa3d.com
Preis	1979 €, 2549 € mit <i>CW1S</i>

Ausprobiert  
— von Make: —

## Tinkering Belles

Tamara Robertson schnackt mit Makerinnen



Ob sie schweißen, schnitzen oder schneiden – im Podcast *Tinkering Belles* wird über so ziemlich alles gefachsimpelt, was mit Making zu tun hat. Allerdings lädt Macherin Tamara Robertson dazu nur Makerinnen ein, denn sie möchte die Frauen in Werkstätten sicht- bzw. hörbar machen. So kommen in den bisher neun Folgen sehr unterschiedliche Makerinnen zusammen: Amie Danielle Dansby ist bekannt für ihre Cosplay- und Lego-kreationen, während Anne Briggs auf ihrem Bauernhof in einer Holzwerkstatt werkelt und die Studentin Danielle Boyer mit ihrer Organisation *Steam Connection* Kindern Roboter näher bringt. Zu Beginn jeder Episode gibt es im Segment *Tech Talk* noch thematisch passende Hintergrundinfos, wie etwa zur Entwicklung der Legosteine.

Robertson ist selbst Ingenieurin und in den USA in Fernsehshows wie *SciJinks* und den *Mythbusters* zu sehen. Diese Verbindung von Technik und Unterhaltung wird auch im Podcast deutlich, wenn sich Diskussionen über CNC-Fräsen und andere Werkzeuge mit Tipps zu *Content Creation* und YouTube als Einnahmequelle abwechseln. Ein *Maker Shop Talk* solle der Podcast sein, so Robertson. Auf Deutsch übersetzt also eine Fachsimpelerei in einer Makerwerkstatt und genau so klingen die Gespräche am Ende auch. Der einzige, wortwörtliche Knackpunkt: Manchmal nehmen Gäste ihre Projekte in die Hand, wenn sie darüber sprechen. Je nach eigenem Anspruch an die Audioqualität stört dies beim Hören oder bringt erst so recht das Werkstattgefühl auf. —hch

URL tinkeringbelles.buzzsprout.com

## LEGO-Roboter

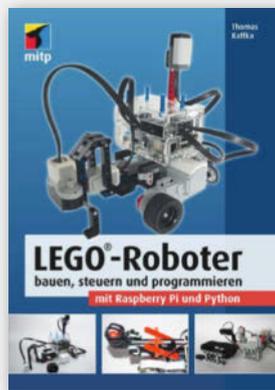
Bauen, steuern und programmieren mit Raspberry Pi und Python

Maker lieben *LEGO*, *Fischertechnik* und sonstige Baukastensysteme. So lassen sich schnell Ideen ausprobieren, Prototypen bauen oder einfach Spaß haben. Auch Roboterbausätze hat *LEGO* in vielen Varianten im Programm, diese gehen aber ziemlich auf die Maker-Kasse. So liegt es nahe, *LEGO* mit Maker-Boards wie *Raspberry* und *Arduino* und schon vorhandener Hardware wie Sensoren und Servos zu kombinieren.

Thomas Kaffka geht in seinem Buch genau diesen Weg: Der Steuerrechner ist ein *Raspberry Pi3*, der durch eine Erweiterung zum *BrickPi3* wird und somit auch *LEGO*-Motoren und Sensoren direkt anschließbar macht. Programmiert wird in der Programmiersprache *Python*, zu der es auch eine über 60-seitige Einführung im Buch gibt.

Dann werden verschiedene Modelle aus *LEGO*-Bauteilen, kombiniert mit *LEGO*-Robotik-

Teilen oder Fremd-Hardware, gebaut und erklärt. Natürlich gibt es auch fahrende Roboter, die Tischkanten vermeiden, Linien folgen oder Objekte sammeln. Viele dieser Roboter werden allerdings per Expertensystem, neuronalem Netz oder Bilderkennung gesteuert. Das Thema künstliche Intelligenz (KI) kann in diesem Rahmen sicher nicht komplett diskutiert werden, aber für erste Experimente sind die Erklärungen ausreichend und motivieren zu weiterem Lernen mittels anderer Quellen. —caw



<b>Autor</b>	Thomas Kaffka
<b>Verlag</b>	mitp
<b>Umfang</b>	344 Seiten
<b>ISBN</b>	978-3-74750-310-2
<b>Preis</b>	27 €

## Heimautomation mit IP-Symcon

Einrichten, steuern, automatisieren

Von den Grundlagen bis zum eigenen, mit der kommerziellen Server-Software *IP-Symcon* arbeitenden Smarthome soll dieses Buch Sie laut Autor führen. Es wird diesem Anspruch gerecht: Beginnend mit den grundlegenden Begriffen der Elektrotechnik, den einzuhaltenden Sicherheitsregeln und den wichtigsten Technologien, widmet es sich dann anhand eines Musterhauses der Planung eines Smarthomes und lässt auch das Thema Notbetrieb, falls die Zentrale mal ausfällt, nicht aus.

Dann wird es konkret: Welche Version von *IP-Symcon* ist die Richtige? Wie installiert man sie und welche Programmierkenntnisse brauche ich, bevor es dann daran geht, das eigen Heim Stück für Stück intelligent zu machen? Nach der Beantwortung dieser Fragen folgt die detaillierte Installationsanleitung.

Schließlich werden zahlreiche Sensoren, Aktoren und andere zur Server-Software passende Komponenten vorgestellt, ohne sich auf einzelne Hersteller festzulegen. Sie erfahren aber auch, wo es sinnvoller ist, auf eine zentrale Steuerung zu verzichten: Ein preiswerter Bewegungsmel-

der kann eine Lampe im Schrank nämlich wesentlich kostengünstiger schalten. Energiesparen ist ein weiteres großes Thema, zum Beispiel bei der Auswertung der Wetterdaten zur Steuerung des Klimas im Haus oder das Herunterfahren diverser Systeme, falls niemand daheim ist. Auch das Thema Photovoltaik wird angesprochen.

Insgesamt erhält man mit diesem Buch eine gute Anleitung, die schnell zu einem funktionierenden smarten Heim führt. Obwohl speziell für *IP-Symcon* gedacht, kann man viele der enthaltenen Informationen auch auf andere Server-Software anwenden. —hgb

*Hinweis: Dieses Buch wird von einem Verlag herausgegeben, der wie die Maker Media GmbH zur Heise Gruppe gehört.*



<b>Autor</b>	Harry Kellner
<b>Verlag</b>	Rheinwerk
<b>Umfang</b>	707 Seiten
<b>ISBN</b>	978-3-83627-583-5
<b>Preis</b>	44,90 € (Buch), 39,90 € (PDF/ebook)

# 3D-Druck im Flugmodellbau

## Vom ersten Entwurf zum finalen Druck

Seit aus 3D-Druckern für die heimische Werkstatt zuverlässig brauchbare Teile herauskommen, erfreuen sich die Maschinen großer Beliebtheit in der Modellbauszene. Die Scale-Modellpiloten stellten schon bald Details wie Zylinder-Attrappen und Radabdeckungen damit her – aber für den 3D-Druck von tragenden Teilen bis hin zur kompletten Zelle eines flugfähigen Modells braucht es bis heute eine gute Portion Pioniergeist.

Thomas Fischers Buch richtet sich an Flugmodellbauer, für die der 3D-Druck noch Neuland ist und deshalb widmen sich die ersten Kapitel ausführlich einer Einführung in diese Technik, vor

allem in Abgrenzung zum Modellbauern oft geläufigeren CNC-Fräsen. Dann wird es speziell, denn der Autor diskutiert ausführlich (und anhand einer Fülle eigener Erfahrungen) alle wichtigen Aspekte des Flugmodellbaus mit 3D-Drucker-Hilfe vom CAD-Entwurf über den Druck der einzelnen Teile

bis zu deren sicherer Verbindung. Letzteres ist ein Kernthema, denn kaum ein Modell passt komplett in den Bauraum eines üblichen 3D-Druckers. Rumpf und Tragflächen bestehen stets aus vielen einzelnen Segmenten.

Statt dabei die Geburt eines konkreten Modells mit einer speziellen CAD-Software, einem Material und einem Slicer kleinschrit-

tig zu beschreiben (wovon man wenig hätte, falls man einen ganz anderen Flieger mit anderer Software konstruieren wollte), erörtert der Autor klugerweise lieber übertragbare Aspekte wie den Druck von leichten, aber stabilen Rümpfen, die Wahl des richtigen Materials und dessen Verklebung sowie die Frage, inwieweit der 3D-Druck etwa in Bezug auf Gewicht, Zeitaufwand oder Reparaturfreundlichkeit gegenüber konventionellen Modellbautechniken punkten kann. Die konkrete Beschreibung von drei seiner umgesetzten Modelle und Tipps fürs 3D-Drucker-Tuning runden das Buch ab. —pek



<b>Autor</b>	Thomas Fischer
<b>Verlag</b>	vth
<b>Umfang</b>	192 Seiten
<b>ISBN</b>	978-3-88180-512-4
<b>Preis</b>	32,90 €

# Kinder-Werkstatt Animation

## Vom Daumenkino zum Stop-Motion-Film

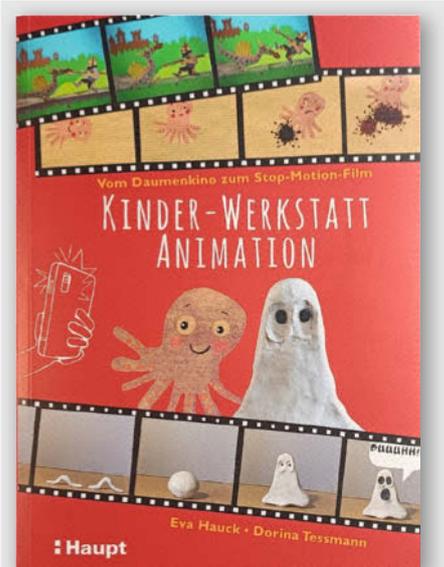
Wer beim Titel *Kinder-Werkstatt Animation* nur Stop-Motion-Filme erwartet, wird eventuell enttäuscht – oder aber, wie wir, positiv überrascht sein. Hier ist Animation in der Ursprungsbedeutung aus dem Lateinischen *animare*, also „zum Leben Erwecken“ gemeint. Dabei sollen statische Bilder zum bewegten Bild werden, sei es per physikalischer Kräfte, optischer Täuschung, Kinetographie oder eben doch Stop-Motion. Es wird ein vielfältiger Mix an Materialien eingesetzt: Papier, Karton, Knete, Washi-Tape, Stoff, Kreide, Magnete, Holz, Moos-

gummi, Draht und Schnur lassen die Verschiedenartigkeit der gezeigten Projekte erahnen. Bei dem einen oder anderen Bild spielt die Makerin in uns mit dem Gedanken, Servos und LEDs einzusetzen, die kommen im Buch allerdings nicht vor. Im Stop-Motion-Kapitel dürfen dann wieder Smartphone und Computer mitspielen.

Alle Beispiele sind großzügig bebildert und man bekommt sofort Lust, mit dem Basteln loszulegen. Die benötigten Materialien sind meist im Haushalt vorhanden, so steht einem spontanen Kreativanfall nichts im Wege.

Kinder im lesefähigen Alter finden sich gleich zurecht, zumal sich das Buch mit vertrautem *Du* direkt an sie richtet. Es gibt zu jedem Projekt eine Altersempfehlung und gleich dazu die Ansage, diese zu ignorieren, wenn ein Projekt besonders interessiert.

Physikalische Grundlagen und historische Hintergründe werten das Buch, sicher ganz im Sinne der Eltern, zu einem Wissen vermittelnden Nachschlagewerk auf. —caw



<b>Autor</b>	Eva Hauck, Dorina Tessmann
<b>Verlag</b>	Haupt
<b>Umfang</b>	144 Seiten
<b>ISBN</b>	978-3-25860-241-7
<b>Preis</b>	24 €



# IMPRESSUM

**Make:** Nächste Ausgabe erscheint am 10.2.2022

## 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](http://www.make-magazin.de)

**Leserbriefe und Fragen zum Heft:** [info@make-magazin.de](mailto: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), Ákos Fodor (akf), Helga Hansen (hch), Rebecca Husemann (rehu), Carsten Meyer (cm), Elke Schick (esk), Carsten Wartmann (caw)

**Mitarbeiter dieser Ausgabe:** Beetlebum, Michael Boretius, Thomas Euler, Lisa Ihde, Michael Jentsch, Matthias Mett, Josef Müller, Wolfgang Neußer, Julia Pfundt, Vladimir Poliakov, Roman Radtke, Erhard Raub, Hubert Schmucker, Daniel Springwald, Gustav Wostrack

**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](http://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](http://www.make-magazin.de)

**Herausgeber:** Christian Heise, Ansgar Heise

**Geschäftsführung:** Ansgar Heise, Beate Gerold

**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:**  
DMV DER MEDIENVERTRIEB GmbH & Co. KG  
Meßberg 1  
20086 Hamburg  
Telefon: +49 (0)40 3019 1800  
Telefax: +49 (0)40 3019 1815  
E-Mail: [info@dermedienvertrieb.de](mailto:info@dermedienvertrieb.de)  
Internet: [dermedienvertrieb.de](http://dermedienvertrieb.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](mailto: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](http://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

Welches Deiner Haushaltsgeräte sollte man auf keinen Fall mit KI ausstatten?



**Josef Müller**  
Walddorfhäslach, erklärt auf Seite 48, wie KI auf Mikrocontrollern wie dem ESP32 funktioniert

Ich würde so ziemlich alles mit KI ausstatten – wenn auch nicht alles mit dem Internet verbinden. Alles, bis auf meine Musikinstrumente: So herrlich schief wie ich bekomme das auch keine KI hin.



**Michael Jentsch**  
Herne, stattet auf Seite 42 zwar kein Haushaltsgerät, dafür eine Drohne mit Objekterkennung, KI und Verfolgungsdrang aus

Keins. Denn selbst ein einfacher Löffel kann durch eine KI verbessert werden, wenn er mir sagen kann, dass meine Suppe zu heiß oder mein Joghurt nicht mehr gut ist.



**Ákos Fodor**  
Hannover, verstärkt das Make-Redaktionsteam als Volontär. Herzlich willkommen!

„Und ich schlafe in der Dusche, weil die Dusche zu mir hält“ ... Ja, auf jeden Fall die Dusche. Bei allem, was sich optimieren lässt, möchte ich zumindest unter der Dusche ein fehlbarer Mensch sein.



**Julia Pfundt**  
Berlin, bringt uns auf Seite 90 in zauberhafte Weih-nachtsstimmung

Den Rasensprenger. Selbst wenn er eine starke Trockenheit oder Temperatur in der Luft misst, möchte ich nicht gerne beim Sonnenbad kalt erwischt werden. Da drehe ich doch lieber selbst den Hahn auf.

## Inserentenverzeichnis

dpunkt.verlag GmbH, Heidelberg ..... 73, 107  
Region Hannover, Hannover ..... 15  
Reichelt Elektronik GmbH & Co., Sande ..... 2

TUXEDO Computers GmbH, Augsburg ..... 132  
Make:markt ..... 15

# Maker Faire®

Die nächsten Events





# 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](http://tuxedocomputers.com)

© Copyright by Maker Media GmbH.