

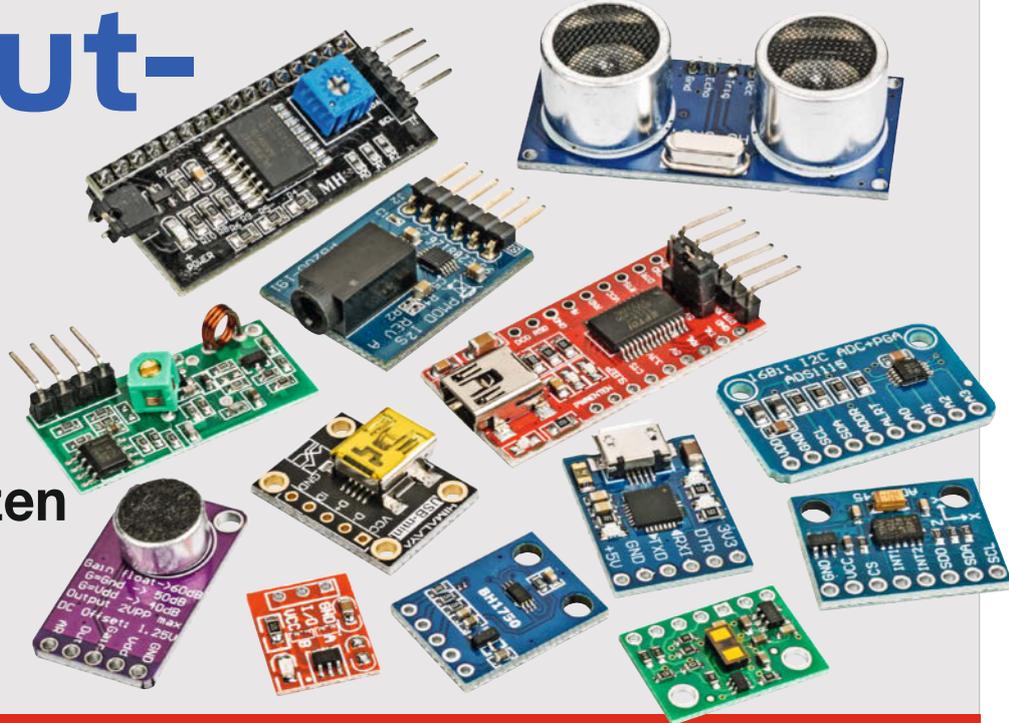
# Make:



**Pico**  
im Test:  
Das bringt  
der neue  
4-Euro-Raspi

## Breakout-Boards

- ▶ Fertige Module für jeden Zweck
- ▶ Anschließen und fertigen Code nutzen
- ▶ Für Pi, ESP, Arduino & Co.



## Werkstatt

- ▶ Komfort-Controller für Lasercutter K40
- ▶ 3D-Konstruktion gratis mit FreeCAD
- ▶ So geht's: Holztüren recyceln

## Raspi-Projekte

- ▶ Alles vernetzen mit Home Assistant
- ▶ Routenplaner für Roboter
- ▶ Frickeln ohne Reue mit Docker

## Hydroponik

- ▶ Salat ziehen ohne Gießen
- ▶ Für Balkon, Terrasse, Garten
- ▶ Mit ESP32-Steuerung



## Nistkasten 2.0

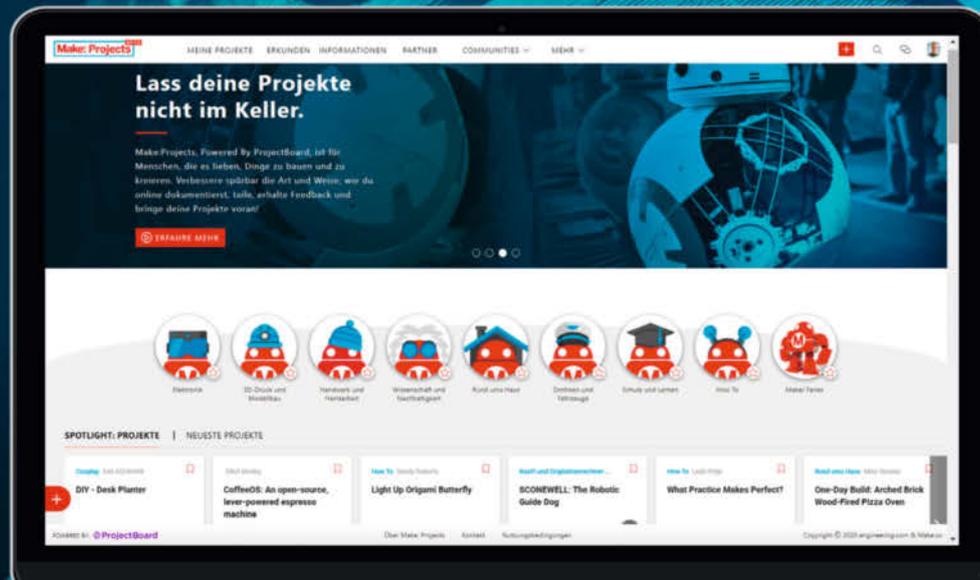
- ▶ Live-Bilder aus dem Nest
- ▶ Mit Klimasensor und Infrarot-Kamera
- ▶ Raspberry Pi Zero mit Bildspeicher



1/21  
11.2.2021  
CH CHF 25.80  
AT 14,20  
Benelux 15,20  
€ 12,90



# Make: Projects

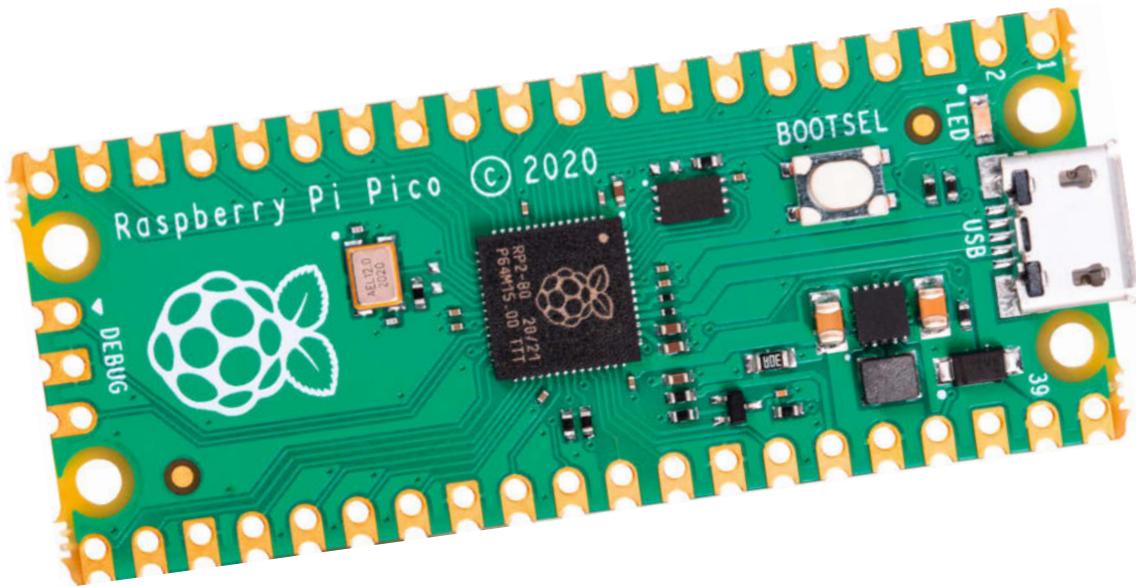


**Zusammen mehr machen –  
Teile deine Ideen, hol dir Inspiration,  
revolutioniere deine Projekte**

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

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

© copyright by Make: Media GmbH



## Ende einer Ära

Für mich gehört *Arduino* zur Maker-Welt wie für andere das Oktoberfest zu München – es geht auch ohne, aber ohne fehlt gewaltig etwas. Ich schreibe allerdings auch schon mehr als zehn Jahre über Maker-Themen und seinerzeit hatte der Name *Arduino* noch eine ganz andere Aura als heute – wer erst seit ein paar Jahren dabei ist, denkt dabei möglicherweise nur an die kostenlose (und ein wenig schlichte) Software, mit der man seine ESP-Boards programmiert.

Dabei war 2005 mit dem ersten Arduino-Board eine echte Revolution ins Rollen gekommen: Es machte die bisher Spezialisten vorbehaltene Programmierung von Mikrocontrollern zur Spielwiese für Quereinsteiger wie Designer, Künstler, Bastler und schließlich Maker, die von den neuen Möglichkeiten begeistert Gebrauch machten. Davon befeuert erlebte Arduino in den folgenden Jahren einen raketenhaften Aufstieg (der in einen waschechten Rosenkrieg unter den Initiatoren mündete), befand sich aber schon vor gut drei Jahren im amtlichen Sinkflug und reif für den Abgesang: „Die Revolution frisst ihre Kinder“ lautete folgerichtig der Titel unseres Editorials in *Make 5/17* (online nachzulesen unter [make-magazin.de/xc8u](http://make-magazin.de/xc8u)), das dem einstigen Maker-Vorzeigeprojekt mittlerweile mangelnde Innovation, zu teure oder an der Praxis vorbei entwickelte Arduino-Boards und Orientierung am Industriebedarf bescheinigte.

Gut, letzteres kennt man auch von anderen Projekten aus den Kindertagen der Maker-Szene: Auch *Ultimaker*, *MakerBot* und *German RepRap* (um nur mal ein paar Beispiele aus der 3D-Druck-Welt zu nennen), sind längst keine Open-Source-Bastelbuden mehr, sondern entwickeln Maschinen für die Industrie. Kann man auch verstehen, von Ruhm und Ehre lebt es sich nun mal deutlich schlechter als von professionellen Margen. Aber immerhin hält ja noch die *Arduino IDE*, die kostenlose Entwicklungsumgebung, die Fahne hoch – was sich

damit nicht programmieren lässt, davon lassen schließlich die meisten Maker die Finger.

Oder?

Der neue Hit heißt *MicroPython* (siehe auch Seite 72) – für immer mehr Boards schreibt man seinen Code in Python. Und diese Programmiersprache ist relativ einfach zu lernen, man kann damit erstaunlich vielen Anwendungsprogramme skripten und nicht zuletzt ist sie das Idiom der Wahl bei vielen KI- und Machine-Learning-Anwendungen. Kurz: Wer Python lernt, kann nicht viel falsch machen.

Die *Raspberry Pi Foundation* hat das schon lange erkannt und setzt bei ihren Einplatinenrechnern von Anfang an auf Python. Mit dem neuen Mikrocontroller-Board *Pico* (siehe auch Seite 8) steckt sie aber zum ersten Mal ihren Claim im angestammten Revier von *Arduino*, *ESP & Co.* ab und lässt – dank *MicroPython* – auch noch die *Arduino IDE* links liegen, denn die wollte von Python noch nie was wissen. Womit wir wieder beim Thema „mangelnde Innovation“ wären, da hat sich in den letzten drei Jahren offenbar bei *Arduino* jenseits des Profi-Bereichs nicht viel getan.

Das Oktoberfest wird es 2021 bestimmt wieder geben. Eine Maker-Welt ohne *Arduino* fühlt sich hingegen schon fast normal an.

Peter König

Peter König

**Sagen Sie uns Ihre Meinung!**

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

# Inhalt

## Holz und Elektronik

Alte Zimmertüren mit Wabenfüllung wandern gemeinsam auf den Wertstoffhof, wenn sich nicht ein geneigter Maker findet, der daraus tolle Gitarren zaubert. Aus ein paar übriggebliebenen Multiplex-Platten und einem Raspi entstand dagegen die Neuauflage unseres Birdcam-Nistkastens.

- 28 Nistkasten 2.0
- 120 Wabentüren-Upcycling



## Breakout-Boards

Für ganz kleines Geld und fast ohne Lötten bieten die kleinen Breakout-Boards jedem Arduino oder Raspi den raschen Zugang zu hochinteressanten Peripherie-Chips und Sensoren, die als SMD-Winzlinge sonst der Industrie vorbehalten sind. Wir stellen Ihnen unsere Favoriten vor – vom Spannungswandler bis zum LiDAR-Entfernungsmesser.

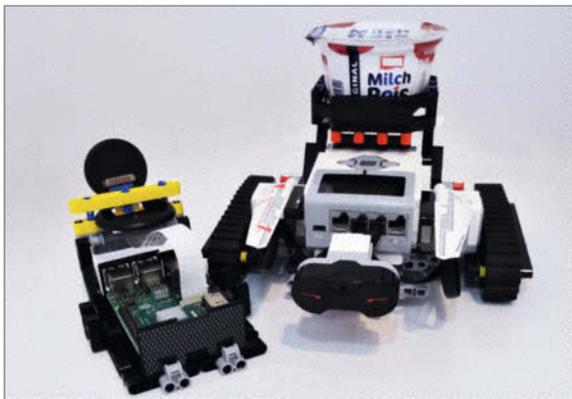
- 10 Breakout-Boards

- 3 Editorial
- 6 Leserforum
- 8 **Erster Test: Raspberry Pico**
- 10 **Breakout-Boards**
- 20 Für zwei Euro: Windmesser fürs Handy
- 26 Werkstattberichte: Termine und Fablab-News, Comic
- 28 **Nistkasten 2.0**
- 36 **Hydroponikanlage mit ESP32-Steuerung**
- 46 Modellbahn-Komponenten mit dem ESP32
- 52 Der Photostick – Malen mit Licht
- 62 Was uns inspiriert: Unendliche Bibliothek, Retro-Konsolen, festliche Leuchtkunst
- 66 So entsteht eine interaktive Riesenmaschine
- 72 MicroPython beschleunigen
- 78 **Pfadplanung für Roboter**

## Raspi-Projekte

Roboter als Handelsreisende: Wie kommt man am schnellsten von A nach B, wenn einige Hindernisse im Weg liegen? Ein Raspi zeigt unserem Lego-Roboter, wo es lang geht – und wir Ihnen, wie man mit Docker-Containern Versions- und Installationsprobleme auf dem Raspi umschifft.

- 78 Pfadplanung für Roboter
- 92 Docker für Raspberry Pi
- 100 Intelligentes Heim mit Home Assistant



## Werkstatthilfen

Mit einem Steuerungs-Upgrade wird aus dem störrischen Laser-Klepper K40 zwar kein Vollblut, aber ein gut dressiertes Ross für die Werkstatt, das ganz neue Kunststücke beherrscht. Da wird Ihnen auch unser Entwurfs-Workshop mit dem kostenlosen FreeCAD gelegen kommen.

- 112 Modernisierung für den China-Laser
- 128 3D-Entwurf mit FreeCAD



- 86 Community-Projekte: Modellbau-Synthesizer, OSSSO bringt Ordnung, Fablab-Verwaltungssystem
- 92 **Docker für Raspberry Pi**
- 100 **Intelligentes Heim mit Home Assistant**
- 110 Reingeschaut: Schwangerschaftstester
- 112 **Modernisierung für den China-Laser**
- 120 **Wabentüren-Upcycling**
- 126 Tipps & Tricks: Flüssiges Gummi
- 128 **Workshop: 3D-Entwurf mit FreeCAD**
- 138 CAD-Designtipps für den 3D-Druck
- 140 Kurzvorstellungen: Abo-Bastelbox, Experimentier-Board, 3D-Drucker und -Material, CNC, Plasma-Aktivierer, Lern-Roboter
- 144 Bücher & Co.: Raspberry Pi, Pico, Python, ESP32, Fotobasteln, Hackerinnen
- 146 Nachgefragt, Impressum

Themen von der Titelseite sind rot gesetzt.

## Hydroponik

Wasser, Wetter und Elektronik – verträgt sich das? Und ob: Unsere Hydroponik-Anlage für den Balkon versorgt Zier- und Nutzpflanzen stets mit der richtigen Menge an Feuchtigkeit und Nährstoffen. Dank ESP32-Steuerung verlangt das Hochbeet aus HT-Rohren nur ein Minimum an Pflege.

- 36 Hydroponikanlage mit ESP32-Steuerung



# Leserforum

## Alternativer Op-Amp

Fledermaus-Scanner, Make 6/20, S. 72

Der im Artikel empfohlene Operationsverstärker ist derzeit nicht im Chipgehäuse erhältlich. Der Operationsverstärker MCP 602-1/P ist pin-kompatibel, sodass das Layout der Lochrasterplatine unverändert bleibt, und benötigt außerdem nur 2,7 Volt. Einen Bezugslink finden Sie unter der Kurz-URL zum Artikel:

► [make-magazin.de/x8ra](https://make-magazin.de/x8ra)

## Mehr zu Oszilloskopen

Messgeräte für Maker, Make 6/20, S. 10

Erstmal vielen Dank für diesen interessanten Artikel in der Make. Ein bisschen kurz war mir die Sektion zu den Logik-Analysatoren/Oszilloskopen. Ich bin interessiert, I<sup>2</sup>C, SPI, USB und eventuell andere(?) Bussignale zu messen (vor allem im Arduino/ESP32-Bereich). Da ist mir nicht klar geworden, auf was ich achten muss (außer der Bandbreite und mindestens 2 Kanälen), um diese Signale praxisnah messen zu können, was vermutlich ja auf eine geeignete Triggerung hinausläuft. Haben alle diese Oszis sowieso diese Technik eingebaut oder woran in der Beschreibung eines Oszis kann ich erkennen, dass sie das können? Im Artikel stellen Sie ein Foto vom Rigol MSO4012 dar (was meinen Wünschen wohl gewachsen wäre),

welches aber nicht mehr erhältlich ist. Welche Funktionen würden Sie empfehlen (4 Kanäle oder höhere Bandbreite, ...)?

Auch ist mir nicht klar, inwieweit bei den Oszis/Logik-Analysatoren (in der Regel) die Messspitzen mit dabei sind (wenn man sie in einem Shop kauft) oder ob man sich diese separat besorgen muss. Die Beschreibungen geben das oft nicht her.

Jörn Becker

*Sicher sind Oszilloskope im Artikel gemessen an der Komplexität der Thematik etwas kurz gekommen; eigentlich wäre dies einen eigenen Artikel wert, womöglich sogar im Rahmen eines Tests. Wenn in der Beschreibung eines Oszilloskops nichts über die Dekodierung von Bus- und seriellen Signalen gesagt wird, kann das Gerät dies auch in der Regel nicht; wie im Artikel beschrieben ist dies meist eine kostenpflichtige Option oder wird werbewirksam herausgestellt. Natürlich können Sie mit geeigneter Trigger-Einstellung I<sup>2</sup>C- oder SPI-Signale auch mit einem Zweikanal-Gerät stabil darstellen, aber um diese zu analysieren (d.h. die übertragenen Impulsfolgen als Bytes und Zahlen anzuzeigen), benötigen Sie bei SPI mindestens drei Kanäle (oder ein Oszilloskop mit eingebautem Logikanalysator) und die Analysefunktion. Übrigens werden selbst bei Billig-Angeboten in der Regel Messspitzen mitgeliefert. Und das Rigol MSO4012 war damals relativ teuer (Basispreis ca. 1500 Euro, mit allen gezeigten Optionen wären über 3000 Euro fällig gewesen).*

## Genauso schlau wie vorher

Mit Spannung habe ich mir den Artikel „Messgeräte für Maker“ durchgelesen. Die Denkanstöße sind bestimmt hilfreich für Leute, die sich bereits mit Messgeräten auskennen und verschiedene Modelle der jeweiligen Kategorie kennen und ggf. schon im Einsatz hatten und jetzt nach Alternativ-Geräten suchen. Als Einsteiger/Anfänger konnte ich aber leider mit dem Artikel nicht viel anfangen. Ich bin genauso schlau wie vorher. Bei der Vielzahl von Geräten ist man komplett erschlagen und hätte gerne konkrete Tipps von den „Profis“. Hier hätte ich mir lieber drei konkrete Beispiele für sinnvolle Neu-Geräte in den Preisstufen klein/mittel/teuer pro Kategorie (Multimeter, Labornetz-Teil, Oszilloskop usw.) gewünscht, die man sich dann konkret anschauen und dann eine Auswahl treffen kann.

Silvio Seifert

*Wir halten es hier wie die Kollegen von der c't: Siegel mit „Kaufempfehlungen“ oder gar „Test-sieger“ finden Sie bei uns nicht. Vielmehr haben wir hier die relevanten Punkte aufgezählt, auf die ein Käufer achten sollte; einen Test-Charakter haben wir in diesem Beitrag bewusst vermieden. Die Geräte sind nach Wichtigkeit geordnet, Sie dürfen gern alles ignorieren, was nach den ersten vier Messgeräteklassen folgt. Wir nehmen Ihre Anregung für Tests einzelner Gerätefamilien aber gern auf.*

## 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 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)



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



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



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



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



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

## Ponal für PLA

3D-Drucke clever kleben, Make 6/20, S. 114

Hier noch eine Anmerkung aus meiner Praxis: Ich habe PLA-Drucke mit Holzleim (Ponal) geklebt, das hält bei mir bombenfest. Einzig der weiße Kleber auf schwarzen Material braucht ab und zu etwas Nachbehandlung.

Uwe Seidler

*Danke für die Anregung, das probieren wir auf jeden Fall aus.*

## Bekanntere Tricks vergessen

Vielen Dank für Ihren Artikel. Aber leider haben Sie dort ein paar bekannte Tricks vergessen, welche ich Ihnen auch zum Test empfehle:

- Dichlormethan bei PLA und ABS (sehr vorsichtig auftragen, am Anfang „trinkt“ der Kunststoff förmlich, auch eine Spritze mit Kanüle kann hilfreich sein, auch das Andicken mit Filamentresten)
- Bei ABS: *Tamiya ABS-Cement* (Plastikkleber, der aber bei größeren Flächen zu dickflüssig ist)
- Bei Nylon: hochkonzentrierte Ameisensäure

Torsten Thurow

*Ui, mit Ausnahme des ABS-Cement sind uns, dem gefährlichsten DIY-Magazin Deutschlands, Ihre Tipps offen gesagt zu heikel: Dichlormethan ist ein starkes Atemgift und konzentrierte Ameisensäure ist nicht nur ziemlich giftig, sondern auch noch stark ätzend. Von Experimenten mit beiden Stoffen raten wir **dringend** ab.*

## Zu pauschal

Ihren Bericht zum Thema „3D Drucke clever kleben“ habe ich mit Interesse gelesen. Der Bericht ist soweit gut, bis auf das Fazit, da wird pauschal gesagt, dass Kunststoffe schwierig zu kleben seien. Das kann man so nicht stehen lassen. Der Autor hat leider nur billige Bastelkleber getestet. Ich arbeite bei einem Zulieferer für Flugzeug-Innenausstattung. Da wird seit vielen Jahrzehnten fast alles geklebt. Für nahezu jedes Material gibt es Spezialkleber. Aluminium, Kunststoffe und Faserverbundstoffe untereinander sind kaum ein Problem, wenn man den richtigen Kleber hat.

Es gibt sehr gute 2-Komponenten-Kleber von z.B. 3M, die verschiedene Kunststoffe sehr gut kleben. Im Bastelbereich habe ich recht gute Erfahrungen mit einem UV-härtenden Kleber für Plexiglas bzw. Acryl gemacht, der auch ohne UV-Licht gut klebt, und auch bei anderen Kunststoffen nicht kleinlich ist. Und in der Not hilft mir oft ein Kleber mit dem Namen *Shoe Goo*, den gibt es beim bekannten Internet-Auktionshaus. Ohne Werbung machen zu wollen ist dieser Kleber für Kunststoffe, Leder, Stoffe usw. bestens geeignet. Eins haben die Kleber jedoch gemeinsam: Sie kosten deutlich mehr und kein Kleber ist für alles geeignet.

Hans-Peter Dietrich

*Vielen Dank für die Tipps. Dass es für nahezu jede Materialverbindung einen passenden Spezialklebstoff gibt, bezweifeln wir nicht und es steht*

*auch – zugegebenermaßen etwas indirekt – im Artikel drin: „... zumal Spezial-Klebstoffe bereits in geringen Mengen unverschämt teuer sind, man sie in der Regel selten braucht und sie nach kurzer Zeit eintrocknen oder unbrauchbar werden.“ Heißt: Es gibt sie zwar und sie wären wohl auch zu bekommen, aber aus den genannten Gründen liegen sie in der typischen Maker-Werkstatt eben nicht parat, weshalb sie im Artikel auch nicht besprochen wurden.*

## Teures Holz-Heft

Make-Sonderheft 2020 „Loslegen mit Holz“

Ich möchte mich für das klasse Holz-Make-Heft bedanken, welches für mich mit einer hohen Geldausgabe endete. Nämlich dem Kauf einer CNC-Fräse *Stepcraft M700* ... Damit habe ich dann auch den Holzlautsprecher aus dem Heft gefertigt.

Jan Karl

## Parallel statt Reihe

Romantischer Kerzenschein im Smart-Home, Make 6/20 S. 52

Das, was ich auf dem Bild  sehe, ist doch eine Parallelschaltung und keine Reihenschaltung!

Robert Stefanz

*Sie haben recht, das ist ein Fehler, tut uns leid. Im Absatz vorher ist korrekt von „Parallelschaltung“ die Rede, im Absatz mit der Hinweisziffer und in der Bildunterschrift fälschlicherweise nicht.*

## Grüße aus der Realität

Editorial: Glaubensfragen, Make 6/20, S. 3

Eigentlich trug ich den Gedanken, mein Abo wieder zu kündigen. Das Heft entwickelt sich immer weiter hin zu „bau' schnell nette Sachen, die dann im Eck stehen“. Mir fehlen mehr Konzepte wie z.B. Hausbuslösungen, Energiegewinnung ... 3D-Druck und der K40-Laser waren richtig gut.

Aber Eure aktuelle Seite 3 sichert Euch meine weitere Unterstützung. Leider werdet Ihr auch viel andere Post bekommen, aber bleibt stark.

Martin Vogl

*Vielen Dank! In der Tat hat das Editorial einigen gar nicht gefallen, aber das lässt uns nicht wirklich zweifeln. Was die gewünschten Themen angeht: Vielleicht freuen Sie sich über den neuen K40-Artikel auf Seite 112 und kommen auch bei der in diesem Heft startenden Smart-Home-Serie (S. 100) auf Ihre Kosten.*



**Im Make-Sonderheft war der Bau des passiven Holzlautsprechers mit konventionellem Werkzeug beschrieben, aber wer sich eine CNC-Fräse zulegt wie unser Leser Jan Karl, kann den nachhaltigen Klangverstärker natürlich auch damit bauen.**

# Raspberry Pi Pico

Wir konnten kurz vor Druck des Heftes einen der ersten neuen Picos von der Raspberry Pi Foundation ergattern. Kern ist ein neuer leistungsstarker Prozessor, den auch andere Hersteller von Boards einsetzen werden. Wir haben ihn einem Praxistest unterzogen.

Alles zum Artikel  
im Web unter  
[make-magazin.de/x5hc](https://make-magazin.de/x5hc)

von Andreas Perband

Der Raspberry Pi Pico - kurz *Pico* - ist deutlich kleiner als ein Raspberry Pi Zero, bisher der kompakteste der Einplatinencomputer-Familie von Raspberry Pi. Er hat eine Menge nützliche Funktionen für Steuerungen an Bord – alle zugänglich über die externen Pins.

Der Pico reitet auf der Welle des aktuellen *MicroPython*-Trends: Dank eines eigenen Interpreters können Programmieranfänger den Mikrocontroller in der leicht erlernbaren Sprache Python programmieren. Fortgeschrittene können ihren Pico-Code in C/C++ schreiben.

Der Pico ist so groß wie ein Arduino Nano. Der auf der Platine verbaute SoC RP2040 hat zwei ARM-Kerne vom Typ Cortex-M0+, die mit bis zu 133MHz takten. Eingebaut sind auch 256KByte SRAM, unterteilt in mehrere Bänke, damit die beiden Kerne ihn parallel nutzen können. Zudem bringt der Pico 2MByte Flash mit. Ihm fehlt allerdings WLAN und Bluetooth.

## Programmable IOs (PIO)

Der Pico bringt zudem eine besondere Art von programmierbaren I/Os mit, die ihn von den meisten anderen Mikrocontroller-Boards unterscheiden.

Er bietet zwei PIO-Blöcke mit jeweils vier sogenannten Zustandsautomaten. Vereinfacht gesagt wechseln solche Automaten zwischen verschiedenen Zuständen, wenn ein bestimmtes Ereignis stattfindet. Sie reagieren selbstständig etwa auf Signalwechsel an digitalen Eingängen und schalten passend dazu digitale Ausgänge.

Diese beiden PIO-Blöcke können quasi Mini-Programme ausführen, um GPIOs zu manipulieren und Rohdaten zu übertragen. Sie laufen aber unabhängig von den ARM-Kernen. Um sie zu programmieren, überträgt man aus normalen Programmen zur Laufzeit Code in Register der PIOs. Auf diese Weise kann man verschiedene Peripherieprotokolle wie SPI, UART und I<sup>2</sup>C ab-

wickeln, ohne die Ressourcen der ARM-Kerne zu belasten. Für einige Protokolle gibt es bereits fertige Bibliotheken. Laut Hersteller lassen sich auf diese Weise auch ein VGA-Ausgang und beliebige andere Protokolle implementieren.

## Varianten von anderen Firmen

Anders als bei den bisherigen Chips für die Pi hält die Foundation diesmal nicht den Daumen drauf. Vielmehr stellt sie den RP2040 auch anderen Herstellern wie Sparkfun, Adafruit und Arduino zur Verfügung. Arduino hat in seinem Blog den Arduino Nano RP2040 Connect angekündigt, auf dem das WLAN- und Bluetooth-Modul *u-blox Nina* sitzen soll (in dem wiederum ein ESP32 steckt). Statt nur 2MByte Flash wird der Arduino Nano RP2040 Connect einen 16-MByte-Chip besitzen und zusätzlich einen 9-Achsen-MEMS-Bewegungssensor sowie ein Mikrofon mitbringen.

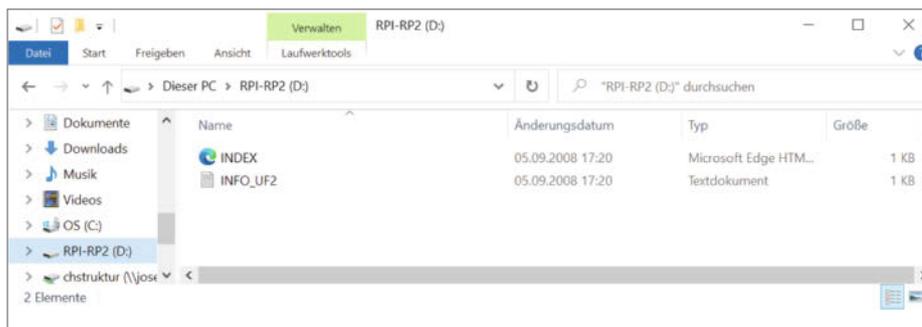
Adafruit und Sparkfun haben ebenfalls eigene Boards angekündigt.

Pimoroni schließlich hat sein *PicoSystem* mit kleinem Display, Steuerkreuz und Feuer-tasten vor allem für Spiele auf dem RP2040 entwickelt.

## Benchmark

Wir wollten herausfinden, wie der Pico im Leistungsvergleich zu anderen Mikrocontrollern abschneidet. Es gibt leider keine anerkannten Benchmarks für Mikrocontroller. Um wenigstens ansatzweise die Geschwindigkeit zu testen, haben wir den Pico mit einem kleinen Python-Skript getestet, das eine einfache Multiplikation durchführt. Dasselbe Skript haben wir anschließend auf einem ESP32 laufen lassen.

Der Pico war nur rund 5 Prozent schneller als der ESP32. Zugegeben, das war nur ein simples Skript, das auch nur einen der Kerne beschäftigt. Und es kommt auch auf die jeweiligen MicroPython-Implementierungen an - wir haben jeweils die aktuellsten verwendet.



Der Pico wird als Laufwerk eingebunden. Darauf kommt MicroPython und die Software per Drag & Drop.

Hier das Skript:

```
import time
now1 = time.time()
def calc(x):
    x*x
    return x
z = range(1000000)
for y in z:
    calc(y)
now2 = time.time()
print(now2 - now1)
```

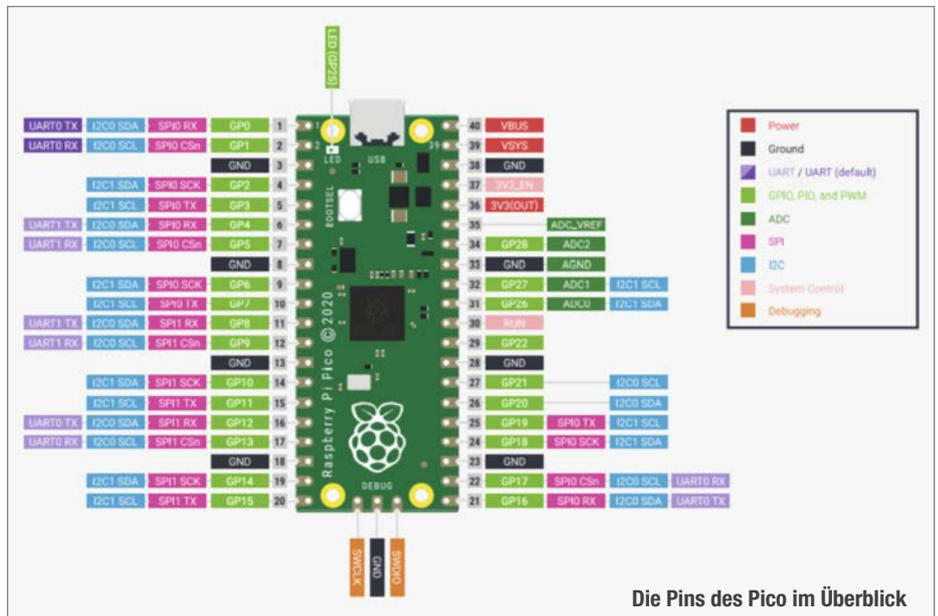
Danach maßen wir außerdem mit dem Pystone-Benchmark für MicroPython (Download siehe Link zum Artikel). Dieser ergab 8 Prozent Geschwindigkeitsvorteil für den Pico.

### Fazit

Der Raspberry Pi Pico ist für 4 Euro ein extrem günstiger Mikrocontroller mit jeder Menge I/O-Ports und vielen Schnittstellen für SPI, UART und I<sup>2</sup>C sowie drei A/D-Eingängen. Leider fehlt ihm ein D/A-Ausgang. Seine programmierbaren PIO-Module versprechen spannende Anwendungen für High-Speed-Peripherie. Da er außerdem mehr Speicher und eine höhere Taktrate hat als typische Mikrocontroller der Arduino-Klasse, bietet er die Möglichkeit, anspruchsvollere Steueraufgaben zu übernehmen. Allerdings sollte man sich klar machen, dass das Mehr an Speicher und Takt dem Einsatz von MicroPython geschuldet ist. Mit weniger Speicher und geringerem Takt würde die Skriptsprache nicht zufriedenstellend bis gar nicht funktionieren. Ein MicroPython-Programm auf einem Pico ist daher auch nicht unbedingt viel schneller als ein in C geschriebenes Programm auf einem Arduino Nano. Wer den Pico hingegen in C/C++ programmiert, wird voraussichtlich von der höheren Leistung profitieren. —anp

## Raspberry Pi Pico: Technische Daten

|                     |  |
|---------------------|--|
| Prozessor           | RP2040, Dual-Core ARM Cortex-M0+   |
| Takt                | 133MHz   |
| Speicher            | 256KB on-chip RAM (SRAM), 2048 KB Flash-RAM, Support für bis zu 16MB Flashspeicher via dedicated QSPI Bus                        |
| Pins                | 30 GPIO Pins, darunter 3, die für analogen Input benutzt werden können   |
| Pinabstand:         | 2,54mm   |
| Controller          | 2 × UARTs, 2 × SPI Controllers und 2 × I <sup>2</sup> C Controller, 1 × USB 1.1 Controller und PHY, mit Host- und Device-Support |
| Anzahl PWM Channel  | 16   |
| Besonderheiten      | 8 × Raspberry Pi Programmable I/O (PIO) state machines   |
| WLAN/Bluetooth      | Nein   |
| Spannungsversorgung | USB, 5V DC   |
| Maße (LxBxH)        | 51,3mm x 21mm x 3,9mm  |
| Programmiersprachen | C/C++, MicroPython   |
| Preis               | ca. 4 Euro   |



Werde Profi-Maker und studiere

## Smart Technology (B.Sc.)

an der Fachhochschule Wedel bei Hamburg

» Studienstart im April und im Oktober möglich

[www.fh-wedel.de](http://www.fh-wedel.de)



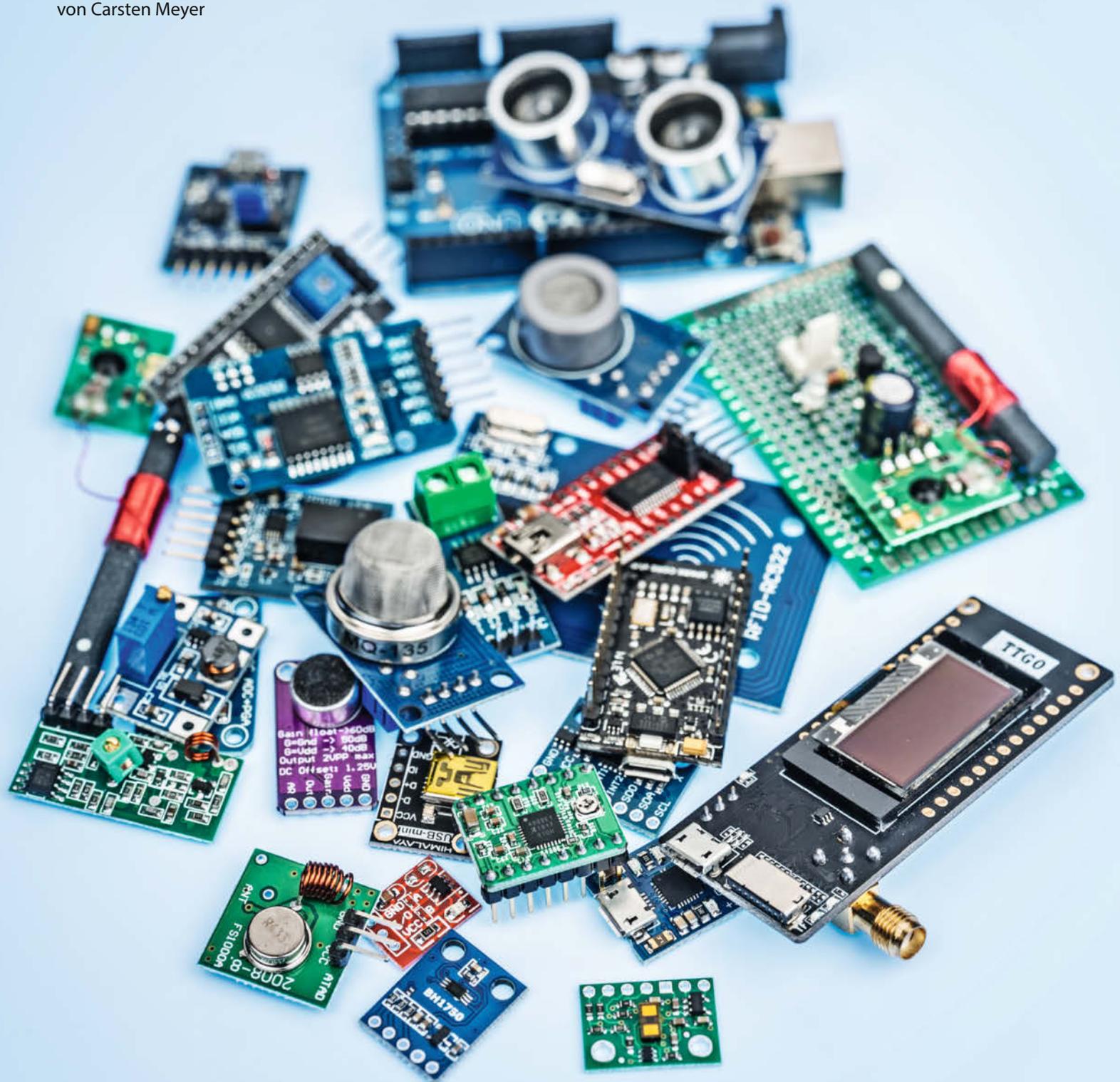
**TOP**  
HOCHSCHULE  
in Deutschland  
**Award 2020**  
StudyCheck.de

© Copyright by Maker Media GmbH.

# Breakout-Boards

Kleine Adapter-Platinen, die die Anschlüsse unhandlich winziger oder Breadboard-untauglicher Bauteile auf das Standard-Rastermaß bringen, nennt man Breakout-Boards. Die werden nicht nur bei lötnwilligen Steckbrett-Bastlern immer beliebter. Wir haben uns auf dem Markt umgesehen und präsentieren eine Auswahl unserer Favoriten.

von Carsten Meyer



**A**dapter-Platinen für SMD-Bauteile gibt es schon lange: Selbst professionelle Elektronik-Entwickler greifen gern auf sie zurück, um eine Schaltung vor dem Platinenlayout-Prozess auf einer Löttrasterplatine aufzubauen und zu optimieren. Das spart Zeit und Geld, denn Musterplatinen im Express-Service kosten immense Aufschläge und sind trotzdem erst nach einigen Tagen da.

Das ging so lange gut, wie die Bauteile von ruhigen, erfahrenen Löttern noch selbst auf der Adapter-Platine montiert werden konnten. Doch selbst Hardcore-Löt-künstler geraten mit den Pin-losen QFN-Gehäusen und Pins im 0,4mm-Abstand an ihre Grenzen. BGA-Bauteile (das sind die mit den Löt-kügelchen an der Unterseite) lassen sich per Handlötung überhaupt nicht mehr montieren.

### Schrittliche Zugriff auf die Kleinode

Leider sind es gerade die interessantesten Bauteile, die es nicht mehr im bedrahteten Gehäuse gibt – etwa die bei Makern so beliebten Touch-, Beschleunigungs- und Umweltsensoren. Parallel mit der Arduino-Erfolgsgeschichte entwickelte sich deshalb ein Markt für vorbestückte Kleinstplatinen, die nicht nur das winzige Bauteil, sondern oft auch gleich seine Mindestbeschaltung (z.B. Pullup-Widerstände, Abblock-Kondensatoren, Spannungs- und Pegelwandler) enthalten.

Genau das macht ein typisches Breakout-Board aus: Ein solches Modul kann dann direkt mit dem Arduino oder dem RasPi verbunden werden, notfalls sogar über Steckbrücken. In unserer Übersicht fehlen deshalb Breakout-Boards, die nur ein simples bedrahtetes Bauteil enthalten, etwa einen Taster oder einen lichtempfindlichen Widerstand – so etwas gilt nicht, da es keine unpraktischen Anschlüsse zum „breakouten“ gibt. Sinnvoll mögen allerdings jene Boards sein, die bedrahtete Bauteile mit ungewöhnlichen Rastermaßen adaptieren; darunter fallen zum Beispiel Sub-D- und USB-Buchsen, die den Lochraster-Löter regelmäßig zur Verzweiflung bringen.

### Schriftliches

Gewisse Amazon- und eBay-Rubriken werden von Adapter- und Breakout-Boards chinesischer Provenienz geradezu überschwemmt. So scheint (dort) die Produktion trotz Preisen im Cent- und Ein-Euro-Bereich ein einträgliches Geschäft zu sein: Tatsächlich können die Hersteller mit den kleinen Platinen prima die immer entstehenden Lücken auf den Produktions-Panels für größere Boards füllen, die sonst in den Abfall wandern würden. So erhält man oft komplette Breakout-Boards zu Preisen, für die es hierzulande nicht einmal das nackte Bauteil gibt.

Problematisch an den Angeboten aus Fernost ist in der Regel die Dokumentation; ein

### Kurzinfo

- » Was sind Breakout-Boards?
- » Typische Anwendungen
- » Übersicht: Unsere Favoriten

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

Schaltbild gibt es nur in Ausnahmefällen, und um Treiber für verbaute Peripherie-Chips muss man sich auch selbst kümmern. Ist zum Breakout-Schnäppchen partout keine Dokumentation aufzutreiben, hilft nur noch das Datenblatt des Halbleiter-Herstellers. Breakout-Boards übernehmen gern die Beispiel-Applikation aus dem Datenblatt oder sind schlicht Plagiate der Original-Schaltungen von Sparkfun, Adafruit oder Arduino; oft fließen kleine Änderungen mit ein, um einen überschüssigen Lagerbestand loszuwerden. Wundern Sie sich also nicht, wenn ein Pullup-Widerstand mit ungewöhnlichen 4,3 statt 4,7kOhm bemessen ist – dann hat der Hersteller noch ein paar Rollen 4,3kOhm-Widerstände übrig gehabt.

Bei namhaften Lieferanten hat man mehr Glück: Die betreiben eigene Webshops und bieten dort die Produkte samt Dokumentation und gegebenenfalls passenden Treibern an. Wer nicht Stunden auf Google verbringen will, nimmt den kleinen Aufpreis für diesen Service gern in Kauf. Positiv überrascht waren wir zum Beispiel vom (durchaus preiswerten) Anbieter AZ Delivery: Den in niedlichen Kartons verpackten Modulen war ein Kärtchen beigelegt, das einen Link und einen QR-Code zu einem E-Book mit der passenden Dokumentation preisgibt. Händler wie Sparkfun, Adafruit, Watterott und Eckstein geben ebenfalls vorbildlich Auskunft, oft mit einer eigenen Git-

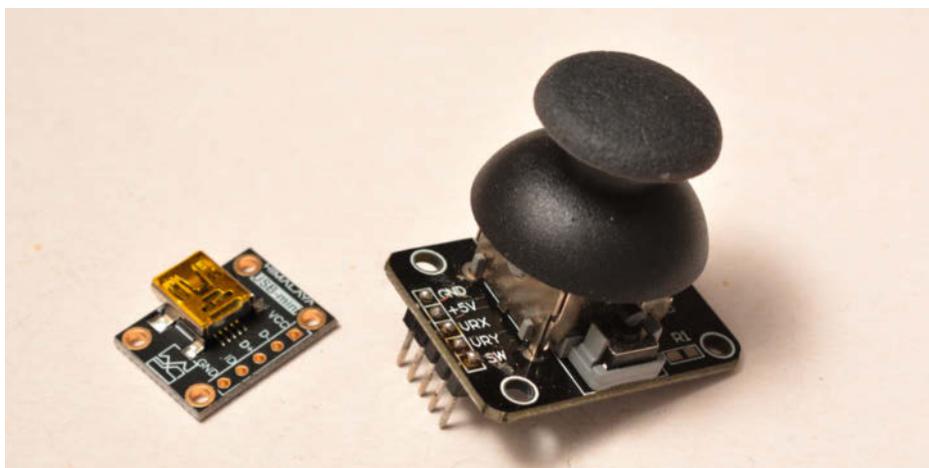


So sollte es sein: Den Boards von AZ Delivery liegt ein Kärtchen mit dem Link zur Dokumentation bei.

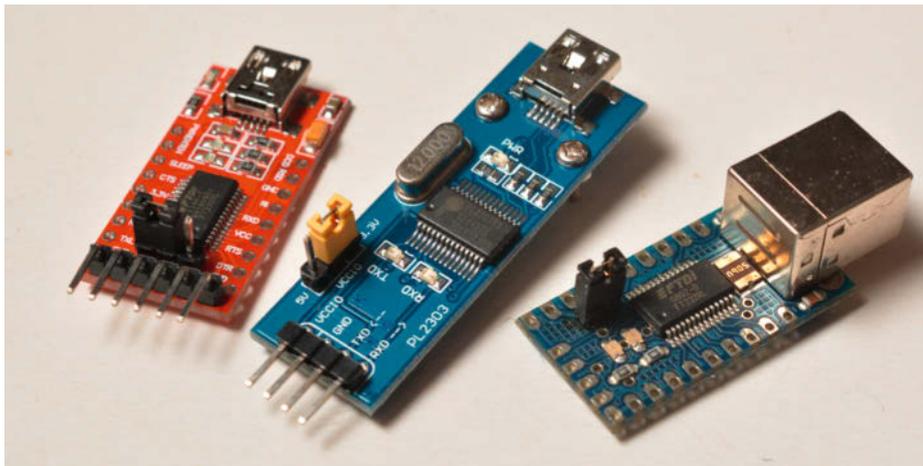
hub-Projektseite zum jeweiligen Produkt. Unter dem Link im Info-Kasten finden Sie Verweise zu Beispielanwendungen, Herstellern und Datenblättern der vorgestellten Produkte.

### Nicht ganz lötfrei

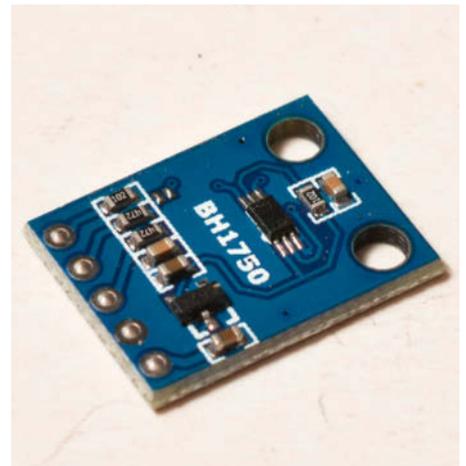
So ganz ohne Löten kommt man auch beim Einsatz von Breakout-Boards nicht aus: Unsere Musterbestellungen erreichten uns zwar mit beigelegten, aber eben nicht eingelöteten Stiftleisten. Immerhin hat man damit noch die Option, stattdessen Kabel anzulöten oder die Anschlussstifte wahlweise von oben (Bauteil-seite) oder unten einzusetzen, gerade so, wie es die Einbausituation erfordert. Die Anschluss-Pads sind indes immer einreihig im



Breakout-Boards für bedrahtete Bauelemente lohnen eigentlich nur, wenn das Bauteil nicht ins Lochraster-Maß passt - wie bei dem Joystick-Potentiometer rechts und der Mini-USB-Buchse. Ansonsten tun es auch angelötete Stiftleisten.



Bei USB-Seriell-Adaptoren (1 bis 5 €) sind die Interface-Chips PL2303, FT232 und CP2102 gängig.



Mehr als eine Fotodiode: Lichtsensor BH1750

Breadboard- und Lochraster-kompatiblen 2,54mm Abstand, somit lassen sich auch die beliebten „Dupont“-Jumperkabel direkt anstecken.

Vorsicht ist allerdings angebracht, wenn sich zwei Welten begegnen: Während Arduino & Co. mit 5V-Logikpegel arbeiten, darf die Eingangsspannung an vielen RasPi-Pins 3,3V nicht überschreiten. Viele 5V-Breakouts arbeiten auch an 3,3V noch einwandfrei, dagegen kann ein 3,3V-Modul an 5V Schaden nehmen, ebenso wie der RasPi vom Ausgangspegel eines 5V-Moduls zerstört werden kann. Dann sind weitere Bauteile zur Pegelanpassung nötig, was den Sinn eines Breakout-Boards etwas untergräbt.

Immerhin: Der Logikpegel einiger Breakout-Boards lässt sich mit einem Jumper oder einer Lötbrücke umstellen und einige der verbauten Chips sind trotz 3,3V-Betrieb 5V-tolerant. Im Zweifel hilft hier nur das Datenblatt des IC-Herstellers. Kleiner Tipp: Wenn es sich nicht gerade um bidirektionale Leitungen

handelt (z.B. I<sup>2</sup>C-Bus), genügt oft ein Widerstand von einigen hundert Ohm in der Signalleitung zwischen 3,3V- und 5V-Schaltung – der sorgt für eine Begrenzung des Stroms, der in einen Eingangspin hineinfließt, wenn der treibende Pegel oberhalb der Betriebsspannung liegt. Demgegenüber erkennen 5V-Eingänge in der Regel auch 3,3V-Logikpegel zuverlässig, hier kann man auf eine Pegelanpassung verzichten.

### Kleine Helfer

Breakouts mit Steckanschluss für Potis und Taster lassen sich übrigens leicht selbst fertigen, indem man einfach eine Stiftleiste an das Bauteil lötet, die auch kleine Differenzen im Rastermaß überbrücken kann. Das wäre bei dem abgebildeten Joystick-Poti allerdings schon recht umständlich, hier ist eine Breakout-Platine durchaus sinnvoll. Die nimmt bereits eine Verdrahtung der Poti-Enden mit Masse und Betriebsspannung vor, so dass man

(einschließlich der zentralen Taster-Funktion) nur 5 Leitungen anschließen muss.

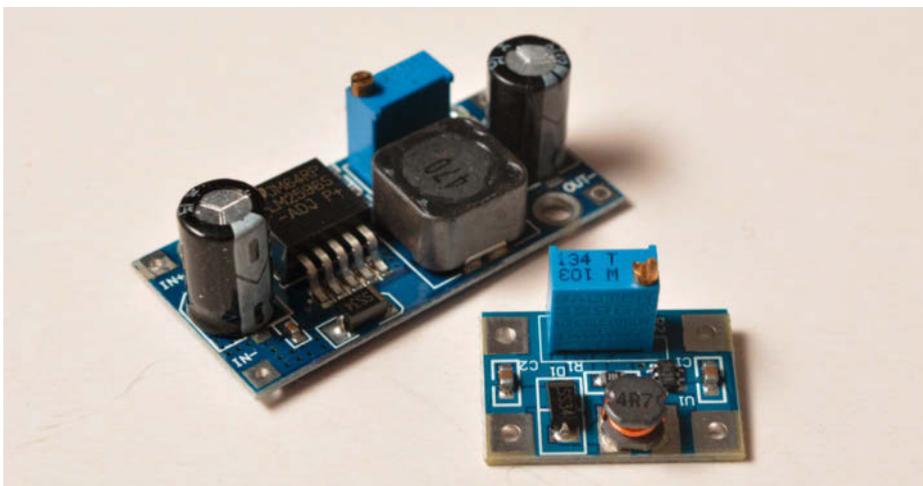
Bedrahtete Sub-D- und USB-Buchsen passen auf kein Steckbrett, weil sie ein „unrundes“ Rastermaß aufweisen und ihre Anschlüsse eng beieinander oder in zwei Reihen angeordnet sind. Eine Adapterplatine ist hier die einzig sinnvolle Möglichkeit, sie experimentell einzusetzen. Ein großer Vorteil bei den USB-Adapterplatinen ist die aufgedruckte Anschlussbelegung – die Pinbelegung von USB-Buchsen führt immer wieder zur Verwirrung. Die USB-Adapter verwenden wir auch dann gern, wenn es nur um die Stromversorgung einer Schaltung geht: USB-Steckernetzteile sind allgegenwärtig und sehr preiswert.

### Lichtsensor mit BH1750

So geht Lichtmessung heute: Das IC BH1750 von Rohm Semiconductor ermittelt die Beleuchtungsstärke direkt in Lux, seine spektrale Empfindlichkeit entspricht etwa der des menschlichen Auges. Gegen Infrarotstrahlung, die das Messergebnis verfälschen könnte, ist er unempfindlich. Sogar ein Flacker-Filter ist eingebaut, er lässt sich also auch von Leuchtstoffröhren und getakteten LEDs nicht verwirren. Der Messbereich geht von 1 bis 65535 Lux. Zum Auslesen der Messwerte dient der I<sup>2</sup>C-Bus, zwei Adressen (0x23 oder 0x5C) sind wählbar. Die Platine enthält einen Spannungsregler, damit sich der 3,3V-Chip auch mit 5V-Logik verträgt.

### USB-Seriell-Adapter

Kaum ein neuzeitlicher PC ist noch mit einer echten seriellen Schnittstelle gesegnet. Um Daten mit dem Mikrocontroller-Projekt auszutauschen, braucht man einen USB-Seriell-Wandler; aus unserem Labor sind die entsprechenden Breakout-Boards nicht mehr wegzudenken. Am bekanntesten sind die mit



Schaltspannungswandler-Platinen ( 1,50 bis 5 €) sind ein sinnvoller Ersatz für die wenig effizienten Linearregler.

dem FTDI-Chip FT232, der wegen seiner Beliebtheit auch bald Opfer von Plagiaten wurde: Die spielten eine Zeit lang, bis FTDI seine Treiber so änderte, dass sie mit den gefälschten Chips nicht mehr funktionierten. Breakout-Boards mit dem FT232 sind deshalb nur zu empfehlen, wenn sie aus zuverlässiger Quelle stammen.

Mit den ICs PL2303, CP2102 und CH340G stehen inzwischen preiswerte Alternativen zur Verfügung, die ebenso gut arbeiten. Allerdings ist der Interface-Pegel beim CP2102 und CH340G nicht umschaltbar (3,3V beim CP2102, 5V beim CH340G); FT232- und PL2303-Boards sind dagegen oft mit einem Jumper versehen, über den man den Pegel wählen kann. Die beste Treiber-Unterstützung bietet nach wie vor FTDI, neben virtuellen COM-Port-Treibern für sämtliche Plattformen stehen auch hardwarenahe Programmierschnittstellen zur Verfügung.

### Spannungswandler

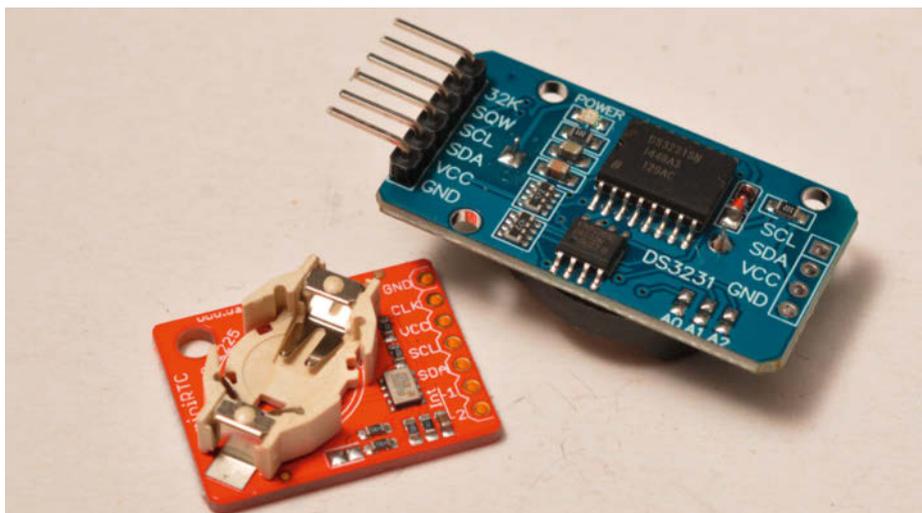
Die verfügbare (Akku-)Spannung ist zu klein oder zu hoch für die Versorgung Ihrer mobilen Schaltung? Zum Heruntersetzen auf die üblichen 3,3 oder 5V genügt bei geringen Ansprüchen an die Energieeffizienz ein dreibeiniger Linearregler, der bei höherem Strombedarf aber gekühlt werden muss - wozu oft der Platz nicht reicht. Die energiesparende Alternative sind kleine Schaltregler-Platinen (Step-Down- oder Buck-Konverter), die ohne Kühlkörper auskommen.

Nicht nur das: Je nach Ausführung können sie eine geringe Batteriespannung auch erhöhen – die heißen dann Step-up- oder Boost-Wandler. Beliebt sind Wandler-Platinen, die aus den 3,7V eines Lilon-Akkus 5V zaubern. Bei den am Markt zu findenden Schaltwandlern ist die Ausgangsspannung über ein kleines Trimpotentiometer einstellbar. Die Einstellung sollte tunlichst vor dem Anschluss Ihrer Schaltung geschehen, sonst ist die Ausgangsspannung womöglich zu hoch.

### Echtzeituhr (RTC)

Auch mit einem zusätzlichen DCF-Empfänger (siehe rechts) ist ein RTC-Modul bei Uhren- und Timer-Projekten Pflicht: Niemand will nach einer Stromunterbrechung jedesmal die Uhrzeit manuell neu stellen wollen, und der DCF-Empfänger benötigt oft Minuten, bis ein vollständiges DCF-Telegramm störungsfrei empfangen wurde. RTC-Module sind deshalb mit einer Pufferbatterie ausgestattet, die den Uhrenschaltkreis auch ohne Versorgungsspannung weiterlaufen lässt. Dessen Stromverbrauch ist so gering, dass eine größere Knopfzelle den Betrieb über Jahre sicherstellt.

Die üblichen RTC-Breakouts arbeiten mit den Chips DS1307, DS3231 oder RV8523, die auch parallel mit anderen Sensoren am I<sup>2</sup>C-Bus des



Mit einem RTC-Modul (3 bis 6 €) kennt Ihr Arduino stets die aktuelle Uhrzeit – bei Uhren-Projekten und Langzeit-Timern essentiell.

Mikrocontrollers betrieben werden können. Der moderne RV8523 (zum Beispiel auf dem miniRTC-Modul von Watterott) hat nicht nur eine geradezu verschwindend geringe Stromaufnahme von 180nA, er besitzt auch ein internes Trimming-Register zur Gangkorrektur. RTCs weichen ansonsten im Monat um ein bis zwei Minuten von der korrekten Zeit ab.

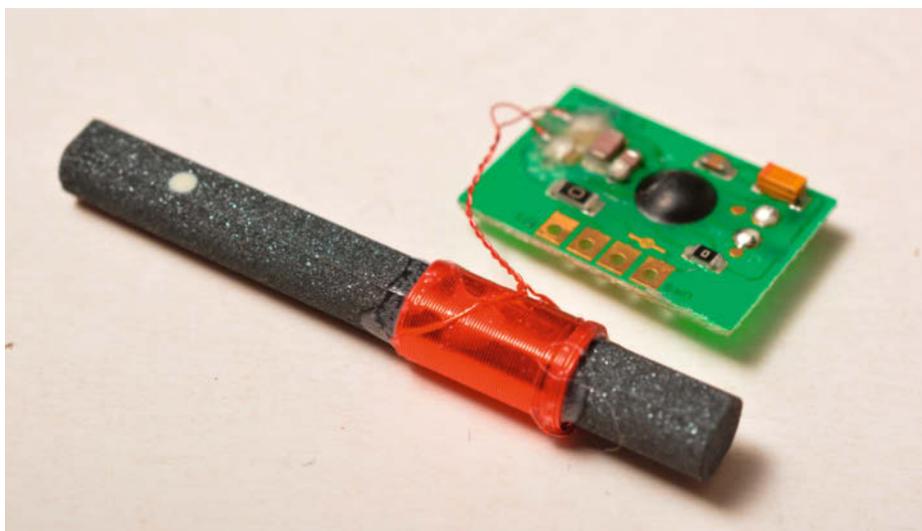
### DCF77-Empfänger

Ihr autonomes Projekt benötigt stets eine hochgenaue Uhrzeit? Wer einen ESP8266 oder ESP32 einsetzt, kann die zumindest über das Internet beziehen. Am Arduino ist zwar oft ein RTC-Modul ausreichend, aber auch das will regelmäßig gestellt werden – zumindest, so lange es noch Sommer- und Winterzeit gibt. Voll autonom arbeitet dagegen ein DCF77-Zeit-

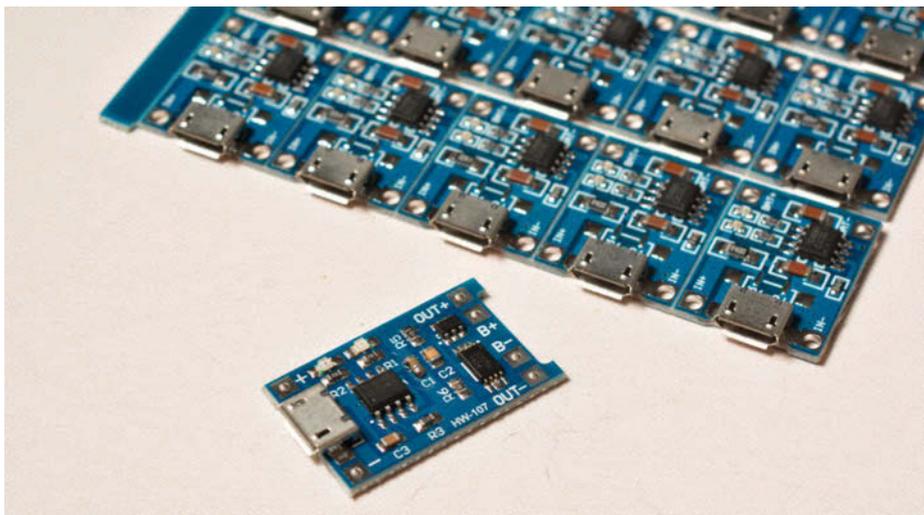
zeichenempfänger mit eigener Antenne. Er liefert Sekunden-Impulse gemäß der gut dokumentierten DCF77-Kodierung, die Umsetzung in die richtige Uhrzeit ist dann Aufgabe der Firmware.

Die Empfangsleistung der mit einer Ferritantenne ausgestatteten Module ist nur ausreichend, so lange keine Störquellen in der Nähe sind und der Ferritstab quer zum Sender ausgerichtet ist. Schon ein Schaltwandler oder eine gemultiplexte LED-Anzeige stören den Empfang nachhaltig, so dass man das Empfänger-Modul dann auslagern muss.

Ganz einfache Module liefern nur einen extrem geringen Ausgangsstrom von wenigen Mikroampere, womit sich schon der interne Pullup-Widerstand eines Arduino-Ports verbietet; andere besitzen eine invertierende Pufferstufe, die genau diesen Pullup-Wider-



Jederzeit die richtige Uhrzeit - mit einem DCF-Empfänger (5 bis 15 €) kein Problem, wenn man einige Fallstricke beachtet.



Lädt gefahrlos einzelne Lilon-Zellen über USB: Ladeschaltung mit dem TC4056A (0,50 bis 1,50 €).

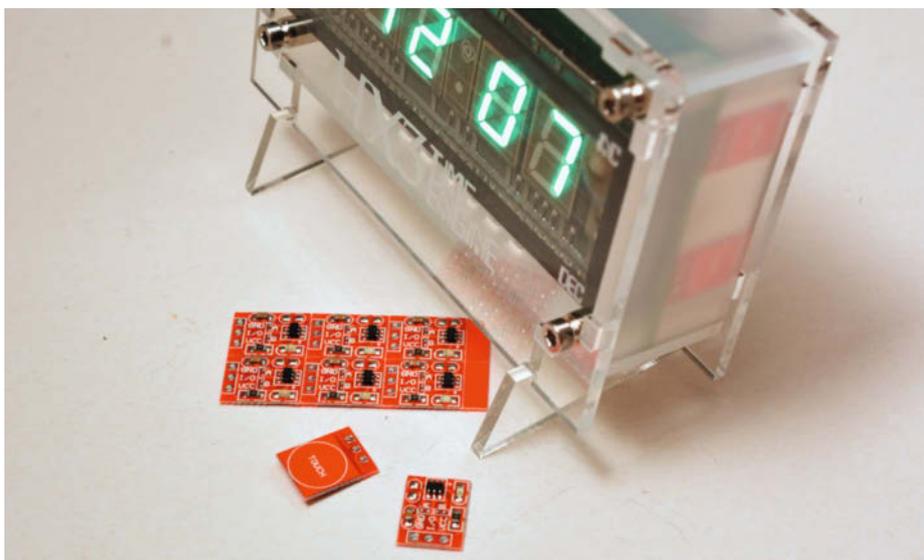


Der Distanzsensor HC-SR04 (3 bis 4 €) arbeitet nach dem Ultraschall-Prinzip, er ist einfach aufgebaut und anzusteuern.

stand benötigt. Eine saubere Betriebsspannung ist Pflicht, hier hilft ein 220Ohm-Vorwiderstand nebst 470µF-Elko von Vcc nach Masse in der Betriebsspannungsleitung zum Modul. Weil sie oft in batteriebetriebenen Uhren eingesetzt werden, ist die Stromaufnahme von DCF77-Empfängern sehr klein.

### Lilon-Ladeschaltung

Lilon-Akkus werden auch im Selbstbau-Bereich immer beliebter – sie haben ein hervorragendes Platz/Leistungsverhältnis und liefern auch als Einzelzelle eine recht hohe Spannung von 3,7 bis 4,2V. Die abgebildete kleine Ladeschaltung mit dem TC4056A kann einen Lilon-Akku mit bis zu 1A über den Micro-USB-Anschluss laden, die Spannungsüberwachung schaltet automatisch in den Konstantspannungsbetrieb



Kapazitive Touch-Sensoren (0,20 bis 1,50 €) funktionieren auch durch Plexiglas und Plastik hindurch. Man kann Bedienelemente deshalb gut „verstecken“, wie bei diesem VFD-Wecker.

und verhindert damit ein gefährliches Überladen der Zelle. Eine rote LED zeigt den Ladevorgang an, eine grüne die Betriebsbereitschaft. Das IC enthält einen Temperatursensor, der die Ladung bei Überlastung deaktiviert.

Der Ladestrom wird über einen aufgelöteten SMD-Widerstand eingestellt; bei der uns gelieferten Platine war er mit 1,2kOhm bestückt, womit sich ein Ladestrom von recht genau 1A ergibt. Durch Erhöhen des Widerstandswerts lässt sich der Ladestrom reduzieren, was für kleine Zellen (oder weniger leistungsfähige USB-Anschlüsse) sinnvoll sein kann. Wenn Sie hier stattdessen 2,2kOhm einlöten, liefert die Schaltung nur noch rund 500mA. Es ist auch noch eine erweiterte, kaum teurere Version der Platine im Handel, die ein zusätzliches IC zum Überstrom- und Tiefentladungsschutz enthält. Die ist dann sinnvoll, wenn die zu versorgende Schaltung keine eigene Unterspannungs-Abschaltung besitzt – also eigentlich immer.

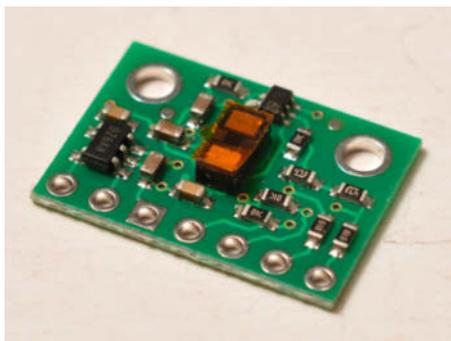
### Touch-Sensoren

Vor Monaten bestellten wir von einem chinesischen Ebay-Händler ein Päckchen mit 50 kapazitiven Touch-Sensoren für gerade einmal 6 Euro - der Preis war verlockend, und vielleicht könnte man sie ja in den nächsten Jahren mal irgendwann einsetzen. Inzwischen sind sie fast aufgebraucht: Wo immer sonst ein Knackfrosch-Taster eingesetzt wird, kommen bei uns nun die winzigen Platinen zum Einsatz. Sie sind mit dem Touch-Sensor-IC TP223 bestückt, das die komplette Elektronik für einen kapazitiven Näherungssensor enthält. Es arbeitet nach dem zuverlässigen Ladungspumpen-Prinzip und kalibriert sich nach dem Einschalten selbst, so dass man die Sensorfläche auch vergrößern kann.

Der Anschluss ist äußerst einfach: Betriebsspannung (3,3 bis 5V) anschließen, der Ausgang wird dann aktiv (High-Pegel), wenn man sich mit dem Finger der Sensor-Fläche nähert. In der Praxis triggerten schon drei Millimeter Abstand der Fingerspitze den Sensor. Er funktioniert auch durch alle dielektrischen Materialien hindurch, selbst 5mm dickes Plexiglas sind kein Problem. Auf der Platine sind Pads für einen SMD-Kondensator der Baugröße 0603 frei, mit dem man die Empfindlichkeit verringern kann (einige zehn Picofarad reichen). Es gibt die Sensor-Platinen auch mit (leider etwas sparsam) beleuchtetem Touch-Feld. Hier kann man über Lötbrücken einstellen, ob die „Taste“ nur als Momentkontakt arbeitet oder abwechselnd ein- und ausschaltet (siehe Datenblatt des TP223).

### Distanzsensoren

In der Robotik immer wieder benötigt: Sensoren, die den Abstand zu umgebenden Objekten



**Time-of-Flight-Distanzsensoren mit dem VL53L1X arbeiten nach dem LiDAR-Prinzip und sind noch relativ teuer (7 bis 14 €) .**

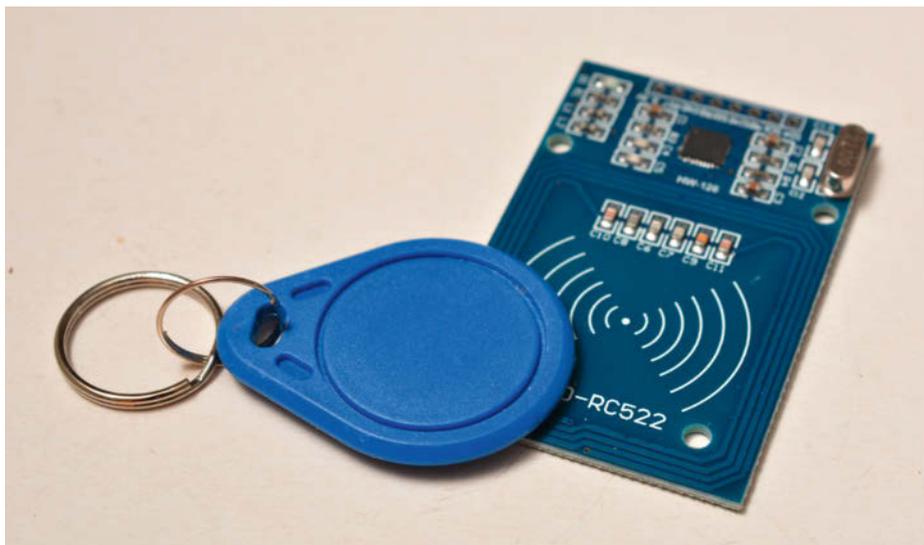
ten messen können. Bei geringen Entfernungen sind Infrarot-Sensoren eine geeignete Wahl: Sie messen die Lichtmenge, die von einem Objekt reflektiert wird. Da diese nicht nur von der Entfernung, sondern auch von der Größe und vom Material des angestrahlten Objekts abhängt, sind die Ergebnisse eher Schätzungen. Der Messausgang von einfachen Modulen liefert üblicherweise eine analoge Spannung - je näher das Objekt ist, desto höher ist sie. Andere detektieren einfach einen fest vorgegebenen Mindestabstand und geben dann einen digitalen High-Pegel aus.

Auch bei den Ultraschallsensoren für den mittleren Entfernungsbereich wie dem abgebildeten HC-SR04 ist die Zuverlässigkeit von der Größe und Beschaffenheit der reflektierenden Fläche abhängig. Die Auswertung erfolgt hier durch Zeitmessung vom ausgehenden zum eintreffenden Impuls; man gibt auf einem Pin den Startimpuls und misst dann die Zeit, bis das Echo (auf einem weiteren Pin) eintrifft. Die Zeitmessung muss dabei der steuernde Mikrocontroller (bzw. eine Library) erledigen.

### Time-of-Flight-Sensor

Im Prinzip sind Time-of-Flight-Sensoren natürlich auch Distanzsensoren, sie arbeiten aber deutlich genauer: Sie messen die (äußerst kurze) Zeit, die ein ausgesandter Laserimpuls unterwegs ist, bis er vom reflektierenden Objekt zurückkommt und von einem Photodetektor registriert wird. Diese Zeitmessung im Sub-Nanosekunden-Bereich kann natürlich nicht mehr der steuernde Mikrocontroller übernehmen, dafür sind hochspezialisierte Bauteile nötig.

Das abgebildete Modul von Pololu verwendet den Sensor VL53L1X von ST Microelectronics, der ein komplettes LiDAR-System (Light Detection and Ranging) auf einem Chip integriert. Bei nicht allzu hellem Umgebungslicht kann er Entfernungen von einigen Zentimetern bis 3m mit einer Auflösung von 1mm messen – so etwas leisten weder Infrarot- noch Ultraschall-Sensoren.



**Für Alarmanlagen, Türöffner oder Zeiterfassung: RFID-Chipkartenleser RC522 (5 €) mit Transponder.**

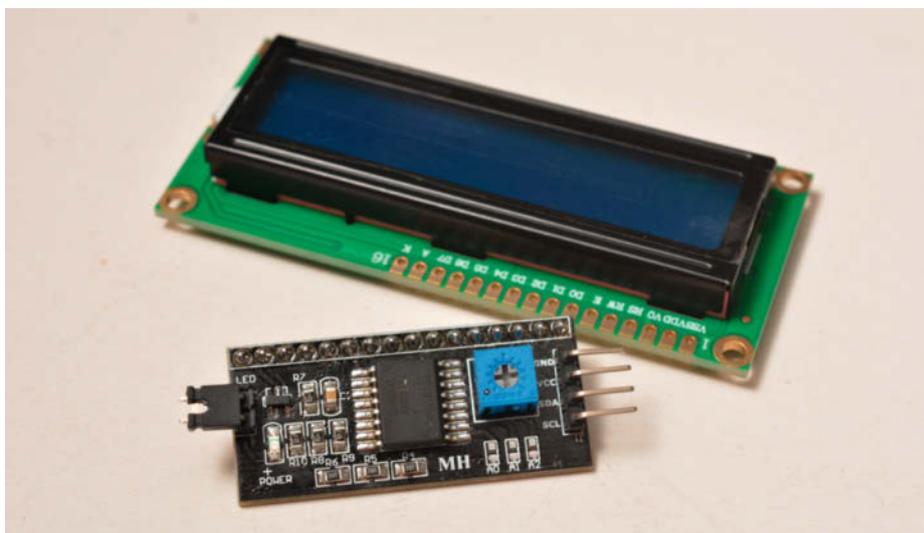
Der Sensor liefert 30 bis 50 Messungen pro Sekunde, die Auswertung erfolgt über den I<sup>2</sup>C-Bus – wobei die Komplexität der Register und Befehle unbedingt den Einsatz der Pololu-Library anraten lässt. Das ausgesandte Licht ist unsichtbar (naher Infrarot-Bereich), im Bereich unter 130cm ist die Detektion vom Umgebungslicht unabhängig. Der Öffnungswinkel des Sensor-Arrays (Field of View, FoV) überstreicht einen Bereich von 27°; dieser lässt sich per Software einschränken, damit der Sensor Objekte außerhalb der Haupt-„Zielrichtung“ (zum Beispiel nahe Anbauteile an einem Roboter) ignoriert.

### RFID-Leser

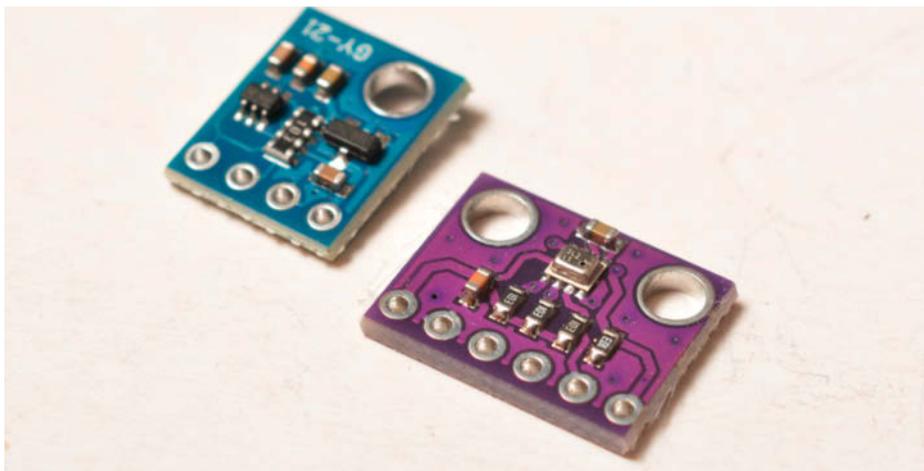
Dieses Set mit dem NXP-Chip MFRC522, einer S50 RFID/NFC-Karte, einem blauen Transpon-

der als Schlüsselanhänger und Steckleisten erhalten Sie für gerade einmal 5 Euro. Als Controller-Interfaces stehen wahlweise SPI, I<sup>2</sup>C und UART (seriell) zur Verfügung, wobei die umfangreiche Arduino-Bibliothek von Miguel Balboa eine SPI-Anbindung voraussetzt. Der MFRC522 detektiert ansonsten anhand der Logikpegel an den Interface-Pins, welche Schnittstelle gerade verwendet wird; fortgeschrittene Anwender können die quelloffene Bibliothek entsprechend modifizieren, um etwa die I<sup>2</sup>C-Schnittstelle (benötigt weniger Leitungen) zu verwenden.

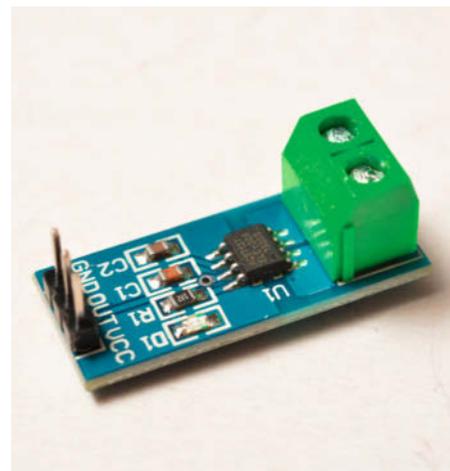
Der MFRC522 kann Karten und Transponder nach den Standards ISO/IEC 14443 A/MIFARE und NTAG mit einer Übertragungsfrequenz von 13,56MHz lesen und beschreiben. In der Praxis reicht eine Annäherung des Transponders auf rund 5cm an die Sensorfläche für eine



**Mit dem I<sup>2</sup>C-Adapter FC-113 (4 €) belegt ein alphanumerisches Display nur noch zwei statt (mindestens) sieben Mikrocontroller-I/Os.**



Die Platine GY-BMP280 (1,30 bis 4 €) ist ein Luftdruck-Sensor mit enormer Auflösung, während das Breakout GY-21 mit dem Si7021 (links, 1,50 bis 5 €) die Luftfeuchtigkeit und Temperatur bestimmt.



Potentialgetrennte Strom-Messungen bis 30A ermöglicht das ACS712-Modul (1 bis 3 €).

Detektion; die Platine kann deshalb problemlos in einem Kunststoffgehäuse „versteckt“ werden. Das etwa 6cm x 4cm große Modul benötigt 3,3V als Versorgungsspannung, verbraucht Betriebsströme zwischen 10 und 30mA und hat einen Ruhestrom von 80µA.

### I<sup>2</sup>C für LC-Displays

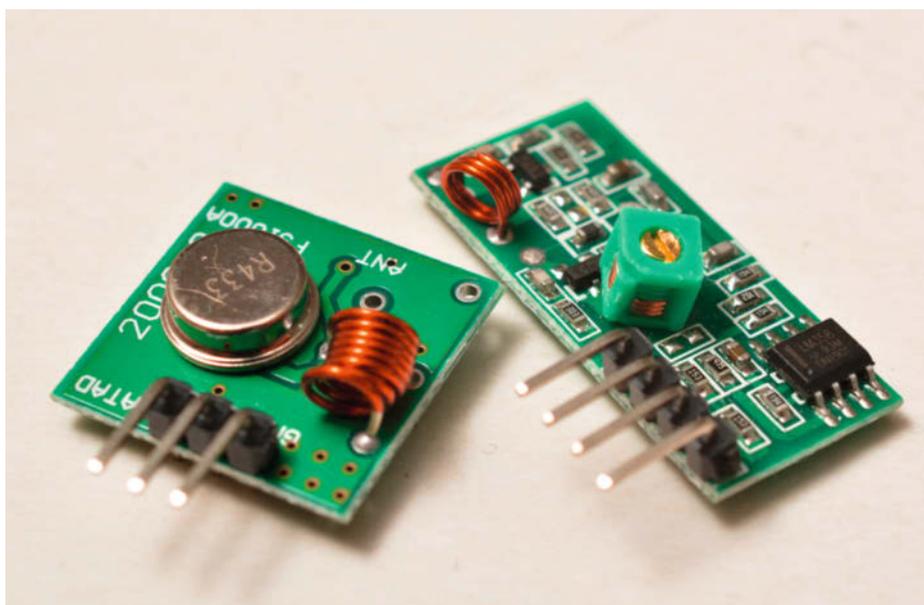
Wenn die I/O-Pins am Arduino knapp werden, aber trotzdem ein LCD für die Benutzerschnittstelle erforderlich ist, hilft diese schmale Platine: Sie ermöglicht den LCD-Anschluss über I<sup>2</sup>C statt über (mindestens) sieben I/O-Pins und kann direkt an handelsübliche LCD-Module mit 14 (ohne Beleuchtung) oder 16 (mit Beleuchtung) Inline-Pins angeflanscht werden. Die Hintergrund-Beleuchtung lässt sich über einen Jumper abschalten, ansonsten wird sie über einen Port-Pin (P3) gesteuert. Das Trimpoti dient zur Einstellung des Display-Kontrastes.

Die Schaltung arbeitet ähnlich wie einige Port-Expander mit dem älteren Interface-Chip PCF8574, der zwei fortlaufende I<sup>2</sup>C-Adressen im Bereich 0x40 bis 0x4F belegt – je nachdem, welche Basisadresse mit den drei Eingängen A0 bis A2 (auf Löt-Jumper geführt) eingestellt ist. Im Lieferzustand waren die Jumper offen, die Adresseingänge liegen dann über Pull-Ups auf „1“, womit sich die Adressen 0x4E und 0x4F ergeben - dies ist beim Einbinden der I<sup>2</sup>C-LCD-Library zu berücksichtigen. Leider unterstützt der PCF8574 nur einen I<sup>2</sup>C-Takt von 100kHz; SMD-Lötprofis könnten ihn durch den schnelleren, Pin- und adresskompatiblen PCA9554 ersetzen, der 400kHz zulässt.

### Luftdruck- und Temperatursensor

Der BMP280 ist ein winziger Luftdruck- und Temperatursensor der Bosch-Sparte Sensortec, der über I<sup>2</sup>C angeschlossen wird und mit einer Versorgungsspannung von 3,3V arbeitet – für den Anschluss an einen 5V-Arduino benötigt man zwingend eine Pegelwandler-Schaltung, denn seine I<sup>2</sup>C-Pins vertragen nicht mehr als 3,6V. Die Pegelwandler sind beispielsweise auf dem BMP280-Board von Adafruit bereits enthalten, nicht aber bei dem abgebildeten: Das ist ohne weitere Maßnahmen nur für den RasPi geeignet, aber deutlich billiger (2 statt 10 Euro).

Beachtlich sind die Leistungen des kleinen Chips: Seine Temperatur-Auflösung beträgt im Bereich -25 bis 85°C nur 0,01°C, und den Luftdruck kann er innerhalb 300 bis 1100hPa auf 1hPa absolut und 0,12hPa relativ bestimmen – das entspricht einem Höhenunterschied von gerade mal einem Meter. Ein aktivierbarer Digitalfilter lässt sehr kurzzeitige Luftdruckschwankungen (etwa beim Zuschlagen einer Tür) unberücksichtigt. Wer statt Luftdruck lieber die Feuchte misst, greift zum GY-21-Board mit dem Si7021, ebenfalls ein I<sup>2</sup>C-Sensor.



Mit dem Sende-/Empfängerpärchen (2,25 bis 3 €) kann man einige Meter drahtlos überbrücken.

### 433-MHz-Übertragungsstrecke

Gleich zwei Sende-/Empfängerpärchen erhielten wir von Makershop.de für 4,50 Euro - die ideale Lösung, wenn ein harmloses Datenvolumen (etwa die Daten eines Außentemperaturfühlers) über ein paar Meter drahtlos übermittelt werden soll. Die Datenrate der im 433MHz-ISM-Bandes mit Amplitudenmodulation arbeitenden Übertragungsstrecke ist auf rund 4kBit/s begrenzt. Der einfache Empfänger arbeitet nach dem bauteilesparenden

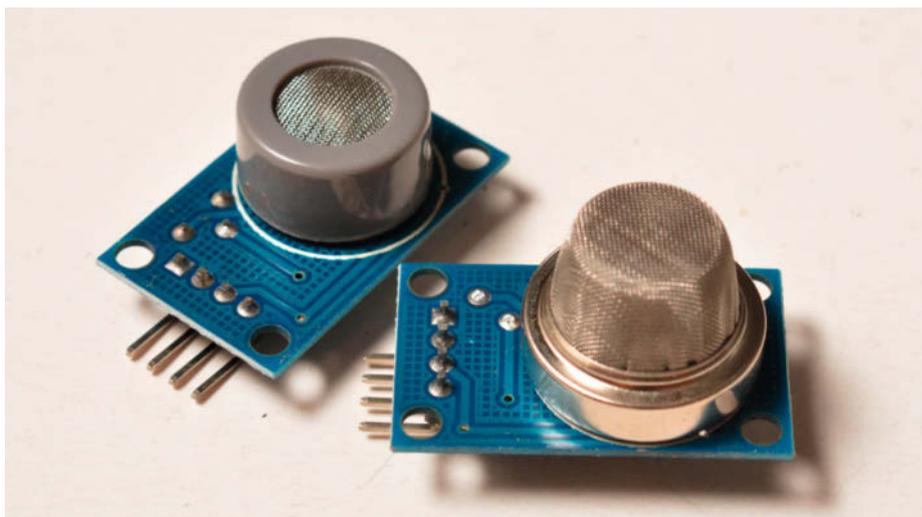
Pendler-Prinzip und ist deshalb kein Ausbund an Empfindlichkeit und Trennschärfe; für viele Anwendungen ist er trotzdem ausreichend.

Mit der Dokumentation des Sets sah es noch etwas mager aus, erst im Makershop-Forum fanden wir die Anschlussbelegung. Die Anwendung ist ansonsten recht trivial: Aus dem Empfänger tröpfeln die Bits so, wie man sie beim Sender hineingeschickt hat. Trotzdem empfiehlt sich zur Absicherung der Übertragungsstrecke gegen Datenfehler die Radio-Head-Library von Mike McCauley, genauer gesagt die RH\_ASK.h für amplitudenmodulierte Übertragungen (Amplitude Shift Keying).

### 30A-Stromsensor

Der ACS712T von Allegro Microsystems ist ein Stromsensor nach dem Halleffekt-Prinzip: Er misst das Magnetfeld, das einen stromdurchflossenen Leiter umgibt. Die Messung erfolgt sozusagen berührungslos – die durch das IC führende Stromschleife ist vom Rest der Schaltung völlig isoliert, laut Datenblatt hält die Isolationsstrecke im Chip über 2000V aus und weist dabei einen Innenwiderstand von nur 0,0012 Ohm auf. Die Halleffekt-Messbrücke ist temperaturkompensiert und liefert bei der auf unserer Platine verbauten 30A-Version des Chips etwa 66mV/A.

Ohne fließenden Strom stellt sich am Ausgang eine Spannung von genau 2,5V ein (genau die Hälfte der 5V-Versorgungsspannung), und je nachdem, ob der gemessene Strom positiv oder negativ ist, weicht die Spannung proportional zum Strom nach oben oder unten ab. Der Chip kann deshalb durchaus auch Wechselströme messen, wenn die auswertende Schaltung (zum Beispiel der Analog-Eingang eines Arduino) schnell genug



Die FC-22-Platine wird mit verschiedenen Gas-Sensoren geliefert (3 bis 30 €, je nach Sensor).

folgen kann und die Messwerte integriert. Die Bandbreite wird durch einen Kondensator an Pin 6 bestimmt, mit  $C2=1nF$  wie auf unserem Muster ergeben sich rund 27kHz. Zum Ausfiltern von Störspitzen kann man  $C2$  vergrößern, mit 100nF liegt die Bandbreite bei nur noch 500Hz.

### Gas- und Luftqualitätssensoren

Die chinesischen Hersteller Henen Hanwei und Zhengzhou Winsen fertigen eine ganze Palette von äußerlich ähnlichen Gas-Sensoren, etwa den MQ-3 für Alkoholdämpfe, den MQ-9 für entflammbare Gase und Kohlenmonoxid, den MQ-135 als allgemeinen Luftqualitäts-Sensor oder den MG-811 für Kohlendioxid, wobei

letzterer angesichts des steigenden Bedarfs während der Corona-Pandemie schon sehr teuer geworden ist (oft über 30 Euro, während man Breakout-Boards mit den anderen Typen oft für 2 bis 3 Euro erhält). Die übliche Darreichungsform ist die FC-22-Platine, die neben dem eigentlichen Sensor (der immer eine analoge Ausgangsspannung liefert) auch einen einstellbaren Schwellwertschalter (LM393 als Komparator) mit einstellbarer Ansprechschwelle enthält.

Die analoge Spannung liegt am A0-Pin an, der Ausgang des Schwellwertschalters an D0; der Zustand wird auch über eine aufgelötete LED angezeigt. Zur Auswertung (Ausgangsspannung im Verhältnis zur Gaskonzentration) muss man unbedingt die Datenblätter der Sensoren heranziehen. Die Stromaufnahme

## SMART LOSLEGEN

## mit dem Node-RED Special von Make!

Jetzt auch komplett digital erhältlich!

**inkl. D1 Mini**

### Make Special: Node-RED

Für Einsteiger ohne tiefere Programmierkenntnisse bietet **Node-RED** die ideale Lösung. Man verknüpft fertige grafische Bausteine zu einem „Flow“ und kann so komplexe Anwendungen für **IoT und Smart Home** entwerfen. Im neuen **Make Node-RED Special** bietet die deutsche Make-Redaktion Anfängern und Fortgeschrittenen einen leichten Einstieg in die Programmierung von Smart Homes. Das Heft wird **portofrei inklusive eines ESP8266 D1 Mini mit WLAN** geliefert!

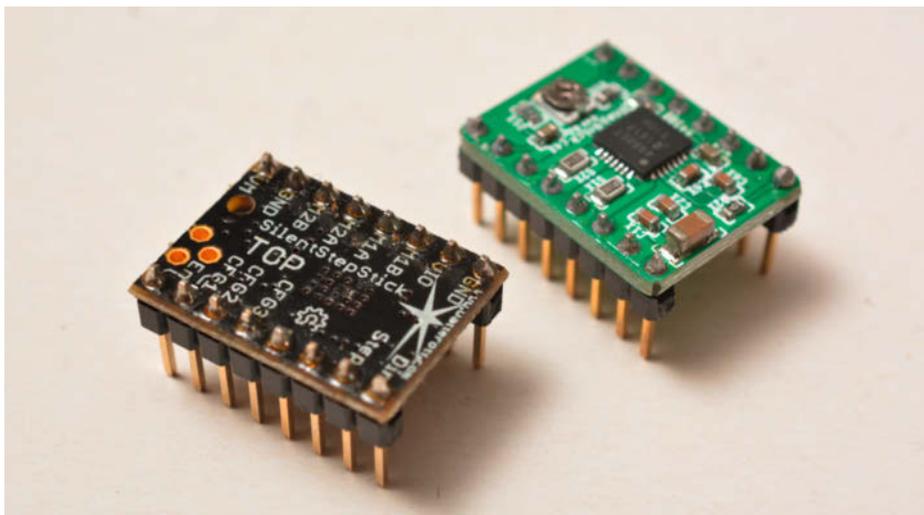
[shop.heise.de/ma-nodered](http://shop.heise.de/ma-nodered)

29,95 € >



[shop.heise.de/ma-nodered](http://shop.heise.de/ma-nodered) >

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



Für kleine NEMA-17-Motoren sind die „Stepsticks“ ideal (2 bis 10 €, je nach Treiber-Chip).

der beheizten Sensoren ist mit rund 1W relativ hoch, außerdem benötigen sie eine gewisse Aufheizzeit von etwa einer Minute und sollten vor Verwendung 48 Stunden unter Betriebsbedingungen „eingebraunt“ werden.

### StepStick-Schrittmotor-Treiber

War früher der L297 das IC der Wahl, wenn es galt, einen Schrittmotor in Bewegung zu setzen, hat der historische Schaltkreis längst würdige Nachfolger gefunden: Die sind kleiner, kommen mit weniger Kühlung aus und können den Motor genauer und geräuscharmer positionieren. Mikroschritt heißt hier das Buzzword: Während der L297 und vergleichbare Treiber aus der gleichen Ära die Motoren

im Voll- und Halbschrittbetrieb voranpoltern ließen, können neuzeitliche Vertreter aus der Stepper-Zunft die Schritte so fein aufteilen, dass die Stepper nur noch ein leises Summen verlauten lassen.

Dazu gehören die in 3D-Druckern allgegenwärtigen Treiber-Chips Allegro A4983 und A4988 (2 oder 2,5A, 1/16-Schritte), der DRV8825 von Texas Instruments (2,5A, 1/32-Schritte) sowie die Trinamic-Familien TMC2100 und TMC2200 (1,2 bis 1,7A, bis zu 256 Mikroschritte). Die besonders leise und laufruhig arbeitenden Trinamic-Module bietet Watterott als „Silent-StepStick“ an.

Einer der neuesten Treiber ist dabei der (noch recht teure) Trinamic TMC5160, der direkt über SPI gesteuert und konfiguriert werden kann; dank besonders niederohmiger MOSFET-Stufen kann er bis zu 3A Ausgangsstrom liefern. Die StepSticks sind zwar grundsätzlich untereinander pinkompatibel, ihre Konfigurationsspins haben aber unterschiedliche Bedeutung. Das ist zu beachten, wenn sie auf einer vorgefertigten Trägerplatine (z.B. einem Arduino-Shield) zum Einsatz kommen sollen.

### Electret-Mikrofon

Selbst ein einfacher Arduino ist in der Lage, Klänge über einen seiner Analogeingänge aufzunehmen und zu bearbeiten, etwa mit einer schnellen FFT-Library – so lange das Audio-Signal mit ausreichendem Pegel vorliegt. Dazu benötigt man einen Vorverstärker, denn übliche Mikrofonkapseln liefern nur einige zehn Millivolt.

Eine besonders elegante Lösung ist dieses Adafruit-Modul mit dem Vorverstärker-Chip MAX9814: Der besitzt eine integrierte Gain-Automatik (AGC) mit einem Regelbereich von 20dB, so dass eine Einstellung der Verstärkung an die gerade auftretende Lautstärke entfällt. Die maximale Gesamtverstärkung kann eben-



Das MAX9814-Breakout-Modul AGC Preamp 1713 (5 bis 15 €) von Adafruit macht eine Anpassung der Verstärkung an unterschiedliche Lautstärken überflüssig.

so wie der AGC-Regelbeich über die Beschaltung zweier Pins eingestellt werden. Der Ausgang ist gleichspannungsgekoppelt und mit 1,25V „vorgespannt“, so dass man ihn direkt mit einem ADC-Eingang eines Mikrocontrollers verbinden kann.

### Gyroskop/ Beschleunigungsaufnehmer

Der auf dem Breakout-Board GY-521 verbaute MPU6050 von TDK InvenSense, ein typischer Vertreter der „Microelectromechanical Systems“ oder kurz MEMS, findet heute ein breites Anwendungsspektrum: Zur Lage- und Gestenerkennung in Game-Controllern, in Mobiltelefonen oder Technikspielzeug (Drohnen, Roboter), zur Ergänzung von Navigationssystemen, als Bewegungssensor im Fitnessbereich oder bei Bildstabilisatoren in Digitalkameras.

Er vereint ein 3-Achsen Gyroskop und einen 3-Achsen-Beschleunigungsaufnehmer, deren Daten ein digitaler Bewegungs-Prozessor (Digital Motion Processing, „DMP“) zusammenführt und wahlweise als Eulersche Winkel oder Quaternion ausgibt, sowie einem Temperatursensor. Die 16 Bit breiten Sensordaten werden über den I<sup>2</sup>C-Bus ausgelesen, daher benötigt man neben der Stromversorgung (3,3V, können aus dem gleichnamigen Arduino-Pin bezogen werden) nur noch 2 Leitungen (SCL und SDA). Die Platine enthält eine Pegelwandler-Elektronik, so dass sie auch am 5V-Arduino betrieben werden darf.

Sehr ähnlich sieht ein Board mit dem ADXL345 von Analog Devices aus, ebenfalls ein MEMS-Sensorchip. Im Unterschied zum MPU6050 kann er aber nur Beschleunigungen aufnehmen, nicht aber „Verkippungen“ um die Längs- Hoch- und Querachse. Preislich gibt es kaum einen Unterschied zum GY-521-Breakout. —cm



Bei Drohnen-Selbstbauern wohlbekannt: Der Gyro-Sensor MPU6050 auf dem Breakout-Board GY-521 (ab 2,60 €). Hinten eine Ausführung mit dem ADXL345, der nur Beschleunigungen aufnimmt (ab 2,50 €).

## LED-LÖSUNGEN



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

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

[www.led-studien.de](http://www.led-studien.de) | [shop.led-studien.de](http://shop.led-studien.de)

## ELEKTRONIK / ZUBEHÖR



USB-Controller für: Joystick, Tastatur, Maus, Drehgeber, universeller I/O

Module, Beschleunigungs-/Drehratensensoren, Interface für I2C, SPI im Dongle-Format, I/O und 12 Bit ADC am USB

Komponenten für intelligentes Licht: DMX, IEC62386  
Ständig Angebote im Webstore:  
[www.codemercs.com/shop](http://www.codemercs.com/shop)

## 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)



**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)

## STELLENANGEBOT



Die **experimenta** ist Deutschlands größtes Science Center mit einer einzigartigen Wissens- und Erlebniswelt. Sie bietet eine außergewöhnliche interaktive und multimediale Vermittlung von Naturwissenschaft und Technik.

Zur Verstärkung unseres Teams suchen wir ab sofort in Vollzeit:

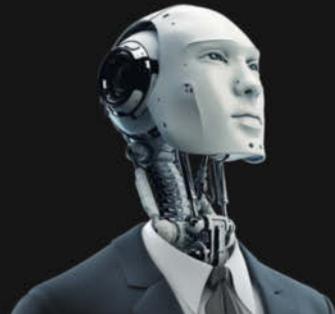
**PROJEKTBETREUER IM MAKER SPACE (m/w/d)**

Weitere Informationen zum Maker Space finden Sie hier:

<https://makerspace.experimenta.science/>

Weitere Informationen zur Bewerbung unter:  
<https://www.experimenta.science/de/jobs>

\*Ausgabe 11/2013: Computer machen die Arbeit.



## Alle reden heute über die Zukunft der Arbeit – wir seit 2013.\*



**Technology Review**  
Das Magazin für Innovation

### Testen Sie 3 Ausgaben Technology Review mit 35% Rabatt.

Jetzt bestellen: [trvorteil.de/testen](http://trvorteil.de/testen)

✉ [leserservice@heise.de](mailto:leserservice@heise.de) ☎ +49 541/80 009 120

+ Ihr Geschenk:



Smartwatch



Lesen, was wirklich zählt in Energie, Digitalisierung, Mobilität, Biotech.

# Windmesser fürs Handy

Egal ob es ums Segeln, Windenergie, Flugzeug-Modellbau oder Drohnen geht: Alle diese Dinge werden vom Wind beeinflusst und wenn man eines dieser Hobbys ausübt, interessiert auch die Windgeschwindigkeit. Warum also nicht mal ein Taschen-Anemometer selber bauen? Es kann mithilfe von 3D-Druck und einer Android-App für gerade mal zwei Euro Materialkosten realisiert werden.

von Michael Jentsch



**D**ie Idee: Drei halbkugelförmige Schalen sind an einer vertikalen Rotorachse montiert und lassen ein Magnetfeld in der Achse rotieren. Die Rotation des Magnetfelds wird vom Android-Handy mit dem eingebauten Magnetfeldsensor erfasst und so die Drehfrequenz der Achse ermittelt. Aus der Rotation kann im Handy dann die Windgeschwindigkeit errechnet und auf dem Display dargestellt werden.

Diese Bauform nennt sich Kugelschalenanemometer. Sie hat etwa gegenüber dem Flügelrad-Anemometer den Vorteil, dass keine Windrichtungsnachführung nötig ist, da dank der vertikalen Achse die Windrichtung egal ist.

Neben dem Kugelschalen- und dem Flügelrad-Anemometern gibt es noch eine Reihe weiterer Bauarten und Prinzipien, von denen die meisten aber wesentlich komplizierter umzusetzen sind.

Ein großer Vorteil der Nutzung des Android-Handys und der Übertragung der Rotation per Magnetfeld ist, dass kein zusätzlicher Mikrocontroller und keine zusätzliche Batterie benötigt wird.

## Die Umsetzung

Bei der Realisierung der Idee bin ich schnell auf die offensichtlichen Probleme gestoßen.

So muss etwa die Achse so gut gelagert werden, dass praktisch keine Reibung vorhanden ist und das Windrad muss extrem leicht sein, um auch bei sehr geringem Wind die Windgeschwindigkeit messen zu können.

An dieser Stelle habe ich auch nach vielen Versuchen keine Lösung gefunden, die ausschließlich mit 3D-gedruckten Teilen zufriedenstellend funktioniert hat. Daher habe ich mich für eine Lösung entschieden, die aus zwei Pogo-Pins als Achsen-Spitzen und den Köpfen von M3-Senkkopfschrauben als Lager-schalen zusammengesetzt ist.

Die Pogo-Pins sind hart, spitz, rostfrei, leicht und gefedert. Sie sind sehr gut geeignet, um damit die Spitzen der Achse zu realisieren. Die Köpfe der Edelstahl-M3-Schrauben sind auch hart und rostfrei. Zudem eignen sie sich erstaunlich gut als Lagerschalen für die Pogo-Pins, da die Flanken der Kreuzschlitze mit zunehmender Tiefe zu einer Spitze zusammengeführt werden.

Das vom in der Achse steckenden Magneten erzeugte Feld muss zwar messbar sein, darf aber nicht so stark sein, dass der Magnetfeldsensor in Mitleidenschaft gezogen wird und der Kompass des Handys nicht mehr funktioniert. Hinzu kommt, dass ein Magnet auch vom Erdmagnetfeld beeinflusst wird (so wie eine Kompassnadel).

Um diese Probleme zu lösen, habe ich zwei Magneten verwendet, die mit entgegengesetzter Nord-Süd Ausrichtung eingeklebt werden. So hat das Erdmagnetfeld nur noch einen

## Kurzinfo

- » Nutzen Sie die Sensoren eines Android-Smartphones als Windmesser.
- » Schritt für Schritt inklusive Kalibrierung – ohne Smartphone-Zerstörung
- » 3D-Druck der notwendigen Teile

## Checkliste



**Zeitaufwand:**  
2 Stunden



**Kosten:**  
2 Euro

## Material

- » 2 M3 Senkkopf-Kreuzschrauben Länge egal, es werden nur die Köpfe benötigt
- » 2 Pogo-Pins
- » 2 runde Scheibenmagnete Ø 8mm, Höhe 3mm
- » Gummiband Größe je nach Smartphone
- » 2 Neodymmagnete
- » Kleber zum Verbinden von Magneten und 3D-Form
- » 3D-Druck Filament

## Werkzeug

- » 3D-Drucker
- » Android-Smartphone
- » Stahlsäge
- » Akkuschauber (Optional, hilft beim Sägen der M3-Schrauben)

## Mehr zum Thema

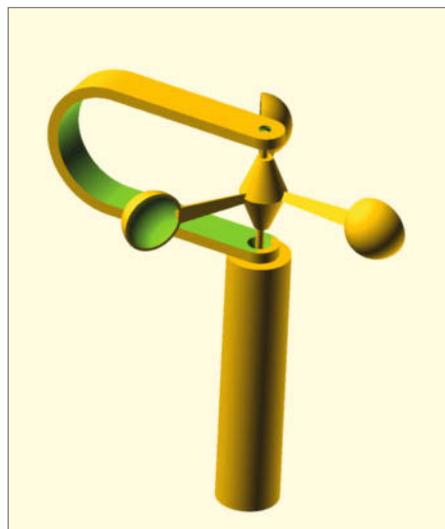
- » Burkhard Fleischer, Anemometer mit Windfahne
- » Pit Noack, Smartphone als Fernsteuerung, Make 4/2018 S. 68
- » Florian Schäffer, Maker-Apps selber machen, Make 3/2018 S. 34

Alles zum Artikel im Web unter [make-magazin.de/xn3r](https://make-magazin.de/xn3r)

sehr geringen Einfluss auf die Bewegung, da die magnetischen Kräfte durch die entgegengesetzte Polung nahezu neutralisiert werden. Um den Magnetfeldsensor im Handy einem nicht zu starken Feld auszusetzen, habe ich die verwendeten Magnete vorsichtshalber kurzerhand entmagnetisiert, sodass sie

gerade noch so eben vom Sensor erfasst werden können.

Das Kalibrieren des Anemometers erfolgt mithilfe des GPS-Empfängers an einem windstillen Tag. So wie man durch schnelles Laufen einen Drachen an einem windstillen Tag in der Luft halten kann, so kann man auch ein Ane-



Das Kugelschalenanemometer – die gewählte Form für dieses Projekt



Das Flügelradanemometer – eine weitere mögliche Form

## Beaufort-Tabelle

| Windstärke in Bft | Geschwindigkeit in m/s | Bedeutung          |
|-------------------|------------------------|--------------------|
| 0                 | 0 - 0,2                | Windstill          |
| 1                 | 0,3 - 1,5              | Leiser Zug         |
| 2                 | 1,6 - 3,3              | Leichte Brise      |
| 3                 | 3,4 - 5,4              | Schwache Brise     |
| 4                 | 5,5 - 7,9              | Mäßige Brise       |
| 5                 | 8,0 - 10,7             | Frische Brise      |
| 6                 | 10,8 - 13,8            | Starker Wind       |
| 7                 | 13,9 - 17,1            | Steifer Wind       |
| 8                 | 17,2 - 20,7            | Stürmischer Wind   |
| 9                 | 20,8 - 24,4            | Sturm              |
| 10                | 24,5 - 28,4            | Schwerer Sturm     |
| 11                | 28,5 - 32,6            | Orkanartiger Sturm |
| 12                | ab 32,7                | Orkan              |

mometer mit Wind versorgen. Der Vorteil ist, dass man dank des GPS-Empfängers die Geschwindigkeit der Bewegung ermitteln und damit die Rotation ins Verhältnis zur (Wind-) Geschwindigkeit bringen kann.

## Der 3D-Druck

Für den Windmesser werden die Flügel, die Achse und der Rahmen im 3D-Drucker gedruckt. Die STL-Dateien können unter dem Link in der Kurzinfo heruntergeladen werden. Alle Teile zusammen benötigen weniger als 10g Filament.

Bis auf die Achse können alle Teile ohne Support oder besondere Einstellungen gedruckt werden. Einzige Ausnahme ist die Achse. Da sie am besten aufrecht gedruckt wird, habe ich im Slicer den Druck mit einem Rand versehen, um die Haftung der Achse auf dem Druckbett zu verbessern. In Cura kann

man hier einfach unter *Build Plate Adhesion* den Typ *Brim* wählen. Dadurch wird sichergestellt, dass die Achse auf dem Druckbett stehen bleibt und nicht irgendwann kippt. Nach dem Druck wird der Brim (Rand) einfach mit einem Cutter entfernt.

Der Druck aller Teile dauert weniger als eine Stunde. Am besten fängt man mit der Achse an, dann können hier schon die Magnete und die Pogo-Pins eingeklebt werden, während der Rest noch gedruckt wird.

## Zusammenbau

Die beiden Pogo-Pins werden einfach von oben und unten in die Öffnungen der Achse geklebt. Sollte der Pogo-Pin auf der Unterseite der Achse nicht richtig passen, liegt das eventuell daran, dass der 3D-Drucker nicht korrekt gelevelt ist oder dass der 3D-Drucker mehr Filament liefert, als notwendig ist (Over-

Extrusion). Passen beide Pogo-Pins nicht richtig in die dafür vorgesehenen Löcher, hat dein 3D-Drucker vermutlich ein etwas anderes Druckbild als meiner. In dem Fall können Sie einfach die OpenSCAD Datei (Link unter der Short-URL) etwas anpassen und die Achse mit leicht verändertem Radius für die Pogo-Pins noch einmal drucken.

Vor dem Einkleben der Magnete müssen diese noch entmagnetisiert werden. Das funktioniert am einfachsten mit einem kleinen Gasbrenner, wie man ihn etwa zum Flammbieren verwendet. Es geht aber auch im Ofen oder im Kochtopf.

Zum Entmagnetisieren muss man den Magnet über seine maximale Einsatztemperatur hinweg erhitzen. Dadurch verliert er einen großen Teil seiner Magnetisierung; wie viel, hängt von der Höhe der Temperatur ab. Bei vielen Neodym-Magneten beginnt der Verlust der Magnetisierung schon bei 80 °C. Es gibt aber auch Neodym-Magneten, die erst bei über 200 °C ihre Magnetisierung verlieren. Deshalb entmagnetisieren Sie die Magnete bitte nicht zu lange.

Ein kleiner Gasbrenner kann problemlos Temperaturen von mehr als 1000 °C erreichen, was bei allen Neodym-Magneten dazu führt, dass sie nahezu vollständig entmagnetisiert werden. Es reicht also, die beiden Magnete für wenige Sekunden mit dem Gasbrenner zu erhitzen, um sie für das Anemometer vorzubereiten.

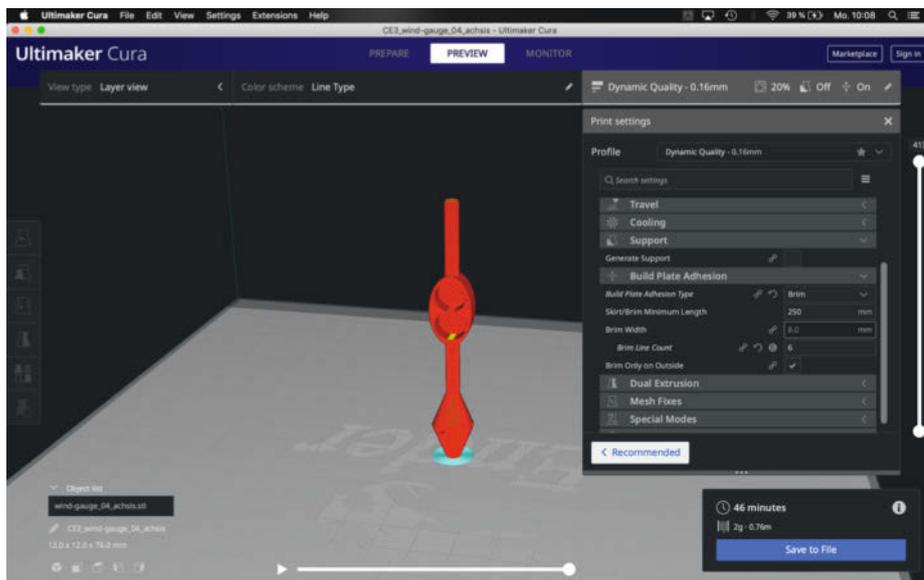
Nach dem Entmagnetisieren haben meine Magnete so gerade eben noch die Kraft, sich gegenseitig anzuheben. Das ist vollkommen ausreichend für den sehr empfindlichen Magnetfeldsensor im Handy.

Nach dem Abkühlen werden die beiden Magnete in die vorgesehenen Vertiefungen eingeklebt. Dabei müssen Sie darauf achten, dass sich die Ausrichtung der Pole beider Magneten unterscheidet, um eine Störung der Rotation durch das Erdmagnetfeld zu verhindern.

Als Nächstes werden die Flügel in die Achse eingesetzt. Die Arme an den Kugelschalen verjüngen sich zu den Spitzen hin, sodass sie auch ohne Kleber sehr gut in der Achse halten. Es reicht also, die Flügel mit sanfter Gewalt in die Öffnungen zu drücken. Auf diese Weise kann das Anemometer auch leicht wieder demontiert und für den Transport verpackt werden.

Die Köpfe der Kreuzschrauben dienen als Lager für die Achse des Windmessers. Bevor Sie die Köpfe der Kreuzschrauben oben und unten in den Rahmen einkleben kannst, müssen sie noch vom unteren Teil der Schraube getrennt werden.

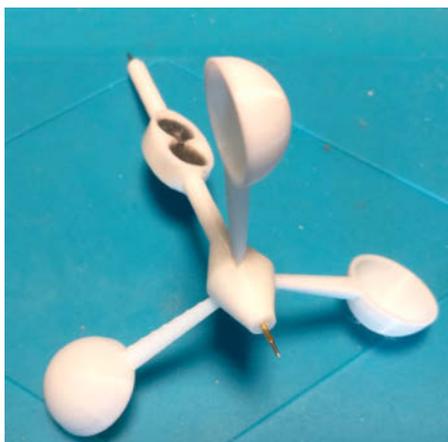
Am einfachsten kann man Schrauben mit einem Akkuschauber und eine Eisensäge durchsägen. Als erstes spannen Sie die Schraube in das Bohrfutter des Akkuschaubers und sägen dann langsam und gleich-



In Cura unter *Build Plate Adhesion* den Typ *Brim* wählen. Dann bleibt die Achse stehen.



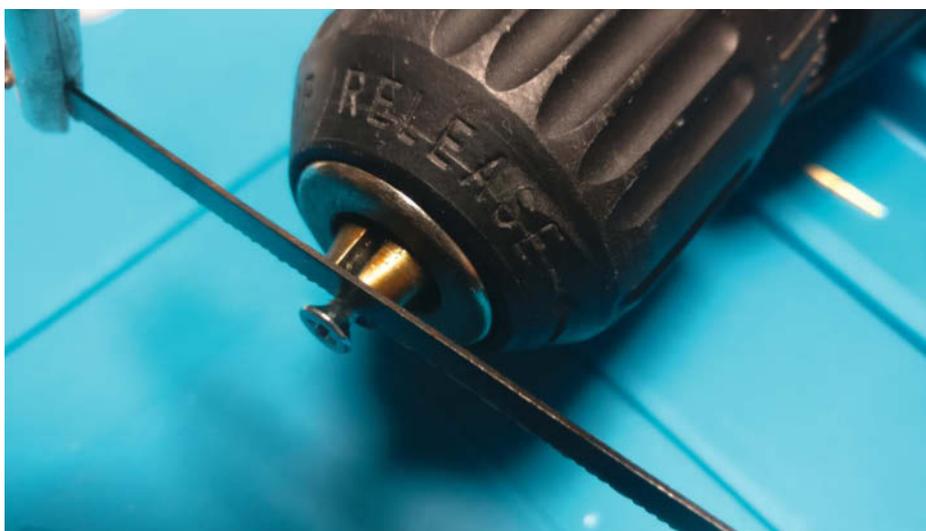
Die Neodyme durch kurzes Erhitzen etwas entmagnetisieren.



Das Flügelschalenanemometer fertig montiert.



Notwendig sind nur die Schraubenköpfe als Lager.



Die Schraube im Akkuschauber festklemmen und den Schraubenkopf abtrennen.

# Ihr Erste-Hilfe-Set:

## Das Notfall-System für den Ernstfall



**Auch komplett auf USB-Stick oder als Heft inkl. PDF mit 29% Rabatt erhältlich.**

### JETZT NEU! c't wissen Desinfec't 2020/21

Ist Ihr Windows erst verseucht, sind persönliche Daten in Gefahr. Hier greift Desinfec't 2020/21 ein, denn das Sicherheitstool bringt sein eigenes Betriebssystem mit und startet direkt von einem USB-Stick. So ist weiterer Schaden gebannt und mit den 5 Viren-Scannern geht's dann auf die Jagd nach dem Übeltäter.

[shop.heise.de/desinfec2020](http://shop.heise.de/desinfec2020)

Einzelheft für nur **14,90 €**

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



[shop.heise.de/desinfec2020](http://shop.heise.de/desinfec2020)



Die Pogo-Pins für die Achse



Das fertig zusammengebaute Anemometer

mäßig, während sich die Schraube im Akkuschauber dreht. Auf diese Weise bekommen Sie einen schönen, sauberen und gleichmäßigen Schnitt.

Die beiden Schraubenköpfe werden dann oben und unten in die Vertiefungen des Rahmens eingeklebt.

Ist der Kleber trocken, kann die Achse eingesetzt werden und der untere Teil des Rahmens mit dem Gummiband verbunden werden. Hier eignet sich besonders gut der „Ankerstich“ (siehe Bild des fertig zusammengebauten Anemometers). Der Knoten ist sehr einfach und lässt

sich sehr leicht wieder lösen. Ich habe hier das Gummiband aufgrund der Länge sogar doppelt gelegt. So hält es noch besser.

### Die Android-App

Damit die Rotation des Magnetfelds im Handy gemessen werden kann, muss der Windmesser nach Möglichkeit so befestigt werden, dass sich der Magnetfeldsensor in der Nähe des Magneten befindet. Bei allen von mir getesteten Handys von Samsung und OnePlus war das oben links der Fall. Allerdings kann ich das nicht für alle Geräte garantieren. Am besten finden Sie die richtige Position mit der *phyphox*-App vom RWTH Aachen (Link unter der Short-URL). Mit der App lassen sich die Messwerte aller möglichen Sensoren im Handy auslesen und auf dem Display darstellen. Sie können also einfach die Daten des Magnetfeldsensors anzeigen lassen und dabei einen Magneten über das Handy bewegen. An der Stelle, an der die Kurve am stärksten ausschlägt, befindet sich der Sensor.

Hat man das Anemometer am Handy befestigt, fehlt nur noch die App, um die Windgeschwindigkeit messen zu können. Die Android-App kann man sich entweder aus dem Google Play Store herunterladen oder selber kompilieren. Das Android Studio Projekt hierfür finden Sie unter der Short-URL zum Artikel.

### Die App kalibrieren

Die App ist schon vorkonfiguriert und sollte direkt nach der Installation funktionieren. Es ist aber möglich, dass sich die Eigenschaften durch die verwendeten Schrauben, Pogo-Pins oder das Druckbild unterscheiden. Für den Fall habe ich auch eine Möglichkeit vorgesehen, die App selber zu kalibrieren.

Das Kalibrieren funktioniert, indem man das Handy mit dem Anemometer etwa auf

einem Fahrradlenker montiert und an einem windstillen Tag mehrmals in unterschiedlichen Geschwindigkeiten eine lange gerade Straße entlang fährt. Nach etwa 15 Minuten hat man ausreichend Messwerte, um daraus einen validen Umrechnungsfaktor von Rotation zu Windgeschwindigkeit zu erhalten.

Nach dem Kalibrieren wird der neue Umrechnungsfaktor gespeichert.

### Weitere Einstellungen

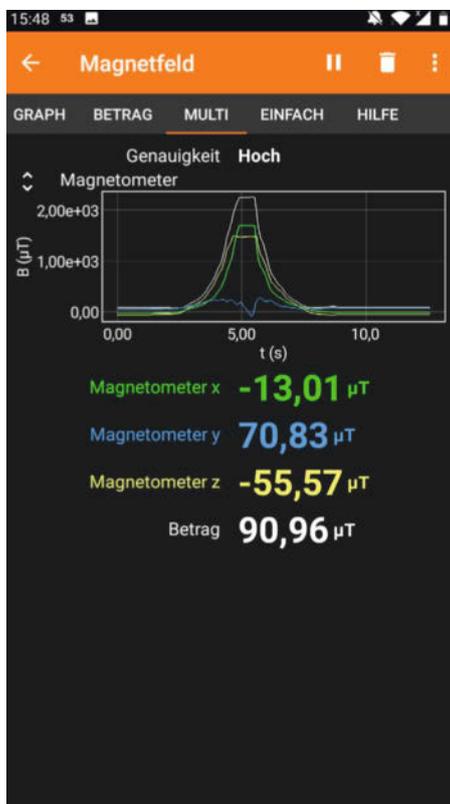
In der App kann man folgende Einstellungen vornehmen:

1. Tiefpass-Filter: Für eine stabilere Anzeige bei stark böigem Wind kann die Anzeige mit einem Tiefpass gefiltert werden, sodass die Anzeige bei Windböen nicht so stark schwankt.
2. Kalibrierungsgeschwindigkeit: Mit der Mindestgeschwindigkeit für die Kalibrierung wird angegeben, ab welcher Geschwindigkeit Messungen für die Kalibrierung vorgenommen werden sollen.
3. GPS-Genauigkeit: Mit der GPS-Genauigkeit wird angegeben, ab welcher GPS-Genauigkeit die Kalibrierung beginnen soll.
4. Drehfrequenz (Kalibrierung): Mit der Min. Drehfrequenz wird angegeben, ab welcher Drehfrequenz des Sensors gemessen werden soll.

### Fazit

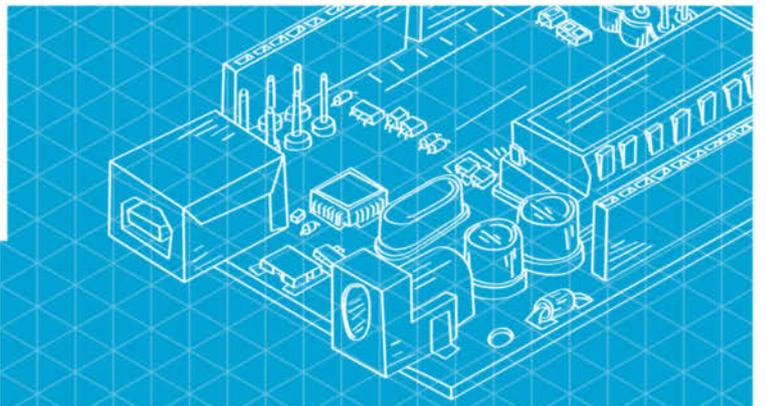
Das Android-Anemometer eignet sich gut zur Messung der Windgeschwindigkeit unabhängig von der Windrichtung, ist modular aufgebaut und einfach nachzubauen. Die kostenlose Open-Source Android-App kann schnell und einfach an die persönlichen Bedürfnisse angepasst und beliebig erweitert werden.

Leider ist der Windmesser nicht ganz so präzise wie professionelle Produkte, aber dafür sehr flexibel und extrem preiswert. —*anp*



An der Stelle, an der die Meßwerte ansteigen, liegt der Magnetfeldsensor des Smartphones.

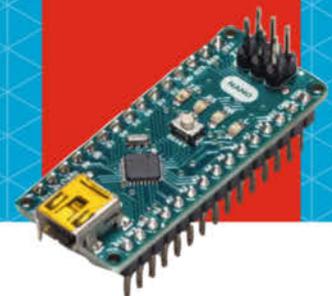
# Make:



## DAS KANNST DU AUCH!



**GRATIS!**



## 2x Make testen und 6 € sparen!

**Ihre Vorteile:**

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

**Für nur 15,60 Euro statt 21,80 Euro.**

\* Für die Laufzeit des Angebotes.

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

© Copyright by Maker Media GmbH.

## Umzug in Steyr

Der Makerspace Steyr-Werke ist im Herbst aus dem Museum Arbeitswelt ausgezogen und nun am Wieserfeldplatz 11 in Steyr untergekommen.

[steyr-werke.at](http://steyr-werke.at)

## Virtuelle Utopiastadt

Der Hackerspace /dev/ tal aus Wuppertal hat dem Kreativprojekt Utopiastadt eine virtuelle 2D-Abbildung zum Geburtstag geschenkt, die online dauerhaft zu sehen ist.

[utopiastadt.online](http://utopiastadt.online)

## Maker-Termine

**FabLabKids@home:**  
Im Coding-Camp zum Coding-Champ ab 2. März dienstags online

[fablabkids.de](http://fablabkids.de)

**Gründungstreffen Makerspace Erlangen**  
4. März online

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

**CNC14-Workshop 29**  
16.–18. April  
Tischlerei Holst, Hamburg  
[cnc14.de](http://cnc14.de)

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

## Veranstalten Sie selbst?

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

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



HappyLab-Geschäftsführer Karim Jafarmadar und Roland Stelzer in den neuen Räumen.

Bild: HappyLab Wien

## Baustart in Wien

### Das HappyLab zieht in das Wiener Stuwerviertel

Das HappyLab Wien vergrößert sich: In diesem Frühling geht es in den neuen Standort im Stuwerviertel, der mit 900 Quadratmetern gleich mehr als dreimal soviel Fläche wie bisher bietet. Dort sollen mehr Maschinen aus den Bereichen Metall- und Holzbearbeitung und Finishing einziehen, darunter ein großer Metall-Laser. Neu ist auch ein Co-Working-Space. „Im neuen HappyLab im Stuwerviertel können Prototyping, Produktentwicklung und Kleinserienfertigung an einem Ort umgesetzt werden“ so die Geschäftsführer Karim Jafarmadar und Roland Stelzer. Schließlich sollen mehr Schulungen und Coachings durchgeführt werden, darunter Maßnahmen, um Makerinnen zu erreichen.

Die Umzugskosten in Höhe von 1,5 Millionen Euro werden zur Hälfte über das Programm „Innovations-

werkstätten“ der Österreichischen Forschungsförderungsgesellschaft (FFG) finanziert: Das österreichische Bundesministerium für Digitalisierung und Wirtschaftsstandort (BMDW) unterstützt das HappyLab über vier Jahre mit 500.000 Euro. Eine weitere Viertelmillion kommt von der Wirtschaftsagentur Wien – das HappyLab ist das Leitprojekt der Wirtschaftsstrategie Wien 2030 zum Thema „Smarte Produktion in der Großstadt“. Die andere Hälfte der Kosten wird aus Eigenmitteln gestemmt. Wer lieber praktisch arbeiten möchte, kann beim Umzug auch durch Inbetriebnehmen und Testen der Maschinen, Malern, der Elektrik oder Bau neuer Möbel helfen.

—hch

► [happylab.at](http://happylab.at)

## Neuer Makerspace in Wuppertal

### Im DevLab-TSA liegt der Fokus auf Nachhaltigkeit und digitaler Kunst

Mit dem Verein DevLab-TSA hat im vergangenen Sommer der zweite Makerspace in Wuppertal eröffnet. Dort beschäftigt man sich mit Open Source, Recycling, Retro-Computing und digitaler Kunst. Die Geschichte des Spaces reicht mit der digitalen Kunstgruppe *Tristar* bis ins Jahr 1987 zurück. Daraus gliederte sich 2012 *The Solaris Agency* (T.S.A.) aus, um sich auf sogenannte Demos zu konzentrieren – Echtzeit-Animationen, die meist mit begrenzter Rechenleistung programmiert werden. Mit der Werkstatt im Mählersbeck 1 hat sich die TSA nun mit Makern zum DevLab-TSA zusammengeschlossen. Das Lab bietet nicht nur den Mitgliedern die Möglichkeit, die 60 Quadratmeter große Räumlichkeit und Werkzeuge zu nutzen, auch mit anderen interessierten Personen arbeitet man gerne zusammen und ist dabei sehr hilfsbereit, besonders in der aktuellen Corona-Situation.

Eine Besonderheit sind die Geräte zum Recyceln von Kunststoff. Dafür wurden Maschinen aus der *Precious-Plastic-Community* gebaut und abgeändert. Einer Spritzgussmaschine mit Handhebel wurde zum Beispiel ihr platzraubender Standfuß entfernt.



Bild: DevLab

Mit einer CNC-Fräse entstehen die Alu-Formen für den Spritzguss. Zum Zerkleinern und Schmelzen des Kunststoffs stehen Schredder und Ofen bereit. Diese Maschinen teilen sich den Space mit 3D-Drucker, Kappsäge und Drehbank. Der Verein legt Wert auf Dokumentation. Dafür pflegen sie Auftritte auf mehreren Plattformen, wie Hackaday.io. Außerhalb des Lockdowns ist der Space meist Samstag und Sonntag Nachmittag geöffnet. Die aktuellen Termine können auf der Webseite nachgelesen werden.

—stri

► [devlab.the-solaris-agency.net](http://devlab.the-solaris-agency.net)

# Millionenförderung für Fab City Hamburg

In den nächsten Jahren sollen sechs neue OpenLabs entstehen

In den kommenden vier Jahren sollen in Hamburg voraussichtlich sechs neue OpenLabs entstehen, die in unterschiedlichen Themenfeldern neue Produktionsmethoden erproben. Das Projekt der Helmut-Schmidt-Universität (HSU) wird vom Bund mit insgesamt neun Millionen Euro gefördert. Der neu gegründete Verein Fab City Hamburg vernetzt die weiteren Projektpartner mit dem langfristigen Ziel, „ein OpenLab in jedem Stadtteil“ zu etablieren. Zu den geplanten Pilot-Labs gehört etwa ein *OpenLab port*, das am Hamburger Hafen helfen könnte, vor Ort neue Ersatzteile herzustellen. Angedacht sind außerdem eine Textilwerkstatt, ein *Med-Tec-Lab* und eine mobile Werkstatt sowie das *OpenLab circular* zur Erprobung von Kreislaufwirtschaften und eine Kooperation mit lokalem Handwerk. Ein zentraler Ansatz ist dabei der Einsatz und die Weiterentwicklung von Open-Hardware-Maschinen.

Die Maschinen in den jeweiligen Werkstätten sollen möglichst direkt vor Ort von Interessierten gebaut werden und das Wissen über *Train-the-Trainer-Workshops* und eine *OpenLab Academy* weitergegeben werden. Außerdem soll mit dem *FabCity-Haus* ein zentraler Hub für das lokale Netzwerk entstehen. Neben der HSU sind an dem Ende 2020 gegründeten Verein



Bild: Fab City Hamburg

die Werkstätten *Fabulous St. Pauli*, das *HoFaLab* und die *Welcome Werkstatt* sowie etwa *Code für Hamburg*, der *Impact Hub* und die *Hamburg Kreativ Gesellschaft* beteiligt. Die Förderung in Höhe von neun Millionen Euro erfolgt über das *Zentrum für Digitalisierungs- und Technologieforschung der Bundeswehr (DTEC.Bw)*, das durch das Konjunkturprogramm der Bundesregierung zur Überwindung der Covid19-Krise finanziert wird. Die Stadt Hamburg unterstützt das Projekt im Rahmen ihrer Innovationsstrategie und ist dem Fab City Network im Sommer 2019 beigetreten, als erste Stadt aus dem deutschsprachigen Raum.

—hch

► [fabcity.hamburg](http://fabcity.hamburg)

## Neue Räume in Bielefeld

Der Makerspace Bielefeld hat ein neues Zuhause gefunden. Weitere Infos gibt es, sobald die Räume wieder öffentlich besucht werden können.

[hackerspace-bielefeld.de](http://hackerspace-bielefeld.de)

## Sanierung in Lünen abgeschlossen

Die Räume des Fablab Lünen im Bürgerhaus sind endlich fertig saniert. Los gehen wird es ebenfalls, wenn die Corona-Regeln dies erlauben.

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

41

Don't stick with the stick.

# Kolophonium

von und mit @beetlebun



# Nistkasten 2.0

Der Frühling naht, die Vögel beginnen bald, sich einen Nistplatz zu suchen. Bauen Sie doch einen Nistkasten, in dem die Piepmätze ihren Nachwuchs aufziehen und Sie sie dabei beobachten können. Wir haben unsere Bauanleitung von 2018 überarbeitet und erweitert.

von Heinz Behling



Im März vorigen Jahres hatte ich den in der Make 2/18 vorgestellten Nistkasten mit Infrarotkamera und Beleuchtung auf meinem Balkon installiert. Kurze Zeit später nahm bereits ein Blaumeisen-Männchen (die sind für die Nistplatzsuche zuständig) ihn als Eigenheim in Besitz und präsentierte ihn etwas später seiner Gattin. Familie Meisenkaiser war gegründet. Und ich konnte von außen jederzeit zuschauen (ach, wie indiskret!).

Die Frau des Hauses hatte natürlich ein paar Kleinigkeiten an der Ausstattung auszusetzen: Insbesondere die Löcher im Boden, die ich nicht nur zur Luftzufuhr, sondern auch zum Entweichen eventuell eindringender Feuchtigkeit vorgesehen hatte, erregten ihr Missfallen ①. Erst als ich die von außen zuklebte, war sie zufrieden, und das Paar begann mit dem Nestbau.

Einige hundert Starts und Landungen später war das Nest gebaut. Zwischendurch gab es aber einen schweren Fall von Raub: Das fast fertige Nest diente einem anderen Meisenpaar als einfach zu erreichende Materialquelle für ihren Eigenheimbau. Leider konnte ich den Dieb nicht genau beobachten, denn die Raubzüge waren eine Sache von Sekunden. Da keimte in mir der Wunsch nach einer zusätzlichen Außenkamera, um zu erkennen, aus welcher Richtung der Räuber anflieg.

Meisenkaisers jedoch gaben nicht auf und bauten das Nest aus kuscheligem Moos und auch eigenen Federn erneut. Zwei Wochen später erblickten zehn Eier das Infrarotlicht ihrer kleinen Welt ②.

Das Brüten, das ausschließlich von Frau Meisenkaiser durchgeführt wird, begann. Ihr Gatte kam nur ab und zu zum Füttern vorbei und servierte meist frische Raupen als Rohkost. Zunächst hatte ich Angst, die Meisen hätten das Nest verlassen, denn am Morgen des zweiten Bruttages verließ Madame das Nest.

## Problem Sonne

Vielleicht lag der Grund in einer zu hohen Innentemperatur, denn der Kasten hing auf der Südseite in praller Sonne. Das dunkle Holz wurde von außen sehr warm. Meine Befürchtung war daher, dass die Eier im Inneren eher gekocht als bebrütet würden und Frau Meisenkaiser dies auch so sah. Hätte ich doch ein Thermometer für innen vorgesehen (siehe Kasten auf Seite 33). Außerdem hatte die Ausrichtung nach Süden einen weiteren Nachteil: Die Sonne schien durch das Einflugloch in den Nistkasten und blendete die Infrarot-Kamera. Die hatte zwar einen zuschaltbaren IR-Filter. Jedoch musste zum Einschalten dieses Filters der Raspberry jedes mal neu gebootet werden. Das hatte ich zwar etwas später mittels eines zeitgesteuerten Skripts gelöst, hatte dann aber das Problem, dass an regnerischen

## Kurzinfo

- » Automatische Infrarot-Kamera in den Nistkasten einbauen
- » Klimasensor für den Innenraum
- » Anschlussmöglichkeit für zusätzliche Außenkamera
- » automatisches Auslagern der Fotos und Videos

## Checkliste



**Zeitaufwand:**  
8 Stunden



**Kosten:**  
circa 80 Euro

## Werkzeug

- » Schraubendreher
- » Heißklebepistole
- » Lötkolben
- » Laubsäge

## Mehr zum Thema

- » Die Bauanleitung zur ersten Version des Nistkastens und viele Infos zur Vogelwelt finden Sie in Make 2/18 ab Seite 66, der online kostenlos zur Verfügung steht.
- » Beispiele für Nachbauten finden Sie im Online-Artikel „Die Nachbauten“

## Material

- » Raspberry Pi Zero W V1.1
- » MakerHawk IR Fisheye Kamera mit automatischem Infrarot-Filter und IR-LEDs
- » Kamerakabel für Raspberry Pi Zero
- » Mikro-SD-Speicherkarte 32GB
- » USB-Hub für Raspberry Zero
- » Temperatur-/Feuchtigkeitssensor DHT22
- » Kohleschicht-Widerstand 10 Kiloohm
- » USB-Kamera V4L-tauglich, optional
- » Steckernetzteil 5V, 1000mA
- » Siebdruckplatte 12mm stark, Maße siehe Download-Zeichnung
- » Sperrholz 3mm, Birke o. ä., Maße siehe Download-Zeichnung
- » Rundholz 10mm Durchmesser, Hartholz
- » Universalschrauben 3mm x 25mm
- » Kabel 2adrig, 0,75mm<sup>2</sup>, für Verbindung zum Netzteil Länge nach Bedarf
- » Reststücke Schaumstoff-Fensterdichtung
- » 2 Scharniere rostfrei, 35mm breit
- » Schraube M3, 12mm mit Mutter

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



- ① Gut gemeint, aber die Löcher im Boden, durch die das Licht hineinscheint, waren keine gute Idee.



2 Zehn Eier waren innerhalb von acht Tagen gelegt.

Tagen mit Infrarotfilter und ohne Sonne keine guten Bilder möglich waren. Außerdem war der Blickwinkel der Kamera etwas eng, sodass ich nicht das komplette Nest sehen konnte. Da musste ich mir etwas einfallen lassen und fand im Netz ein Raspberry-Kameramodul mit Superweitwinkel, das die Beleuchtung messen und den Filter automatisch ohne Bootvorgang

ein- und ausschalten kann. Die wurde gleich für die nun fest geplante Sanierung des Meisenheims bestellt.

Doch die Meise kehrte zurück und nach knapp drei Wochen schlüpfte die erste neue Blaumeise. Insgesamt kamen fünf Küken hervor. Das war der Mutter dann genug und sie entsorgte die restlichen Eier. Was zunächst



3 Die fünf Halbstarke einen Tag vor dem Verlassen des Nestes

grausam klingt, ist aber sinnvoll: Die Vögel ziehen nur so viele Küken auf, wie sie auch ernähren können. Besser fünf voll lebensfähige als zehn unterernährte schwache Vögel, die dann bald anderen als Futter dienen würden.

Eines Tages war jedoch der Aufruf der Webseite, die der im Inneren des Nistkastens arbeitende Raspberry Pi Zero erzeugte, nicht mehr möglich. Offenbar war irgendetwas kaputt. Schweren Herzens öffnete ich den Deckel des Kastens: Frau Meisenkaiser verließ fluchtartig ihr Eigenheim, setzte sich in einen Baum und schimpfte erst einmal. Ich deckte den Nistkasten provisorisch ab und widmete mich der Elektronik. Der Schaden: Die Speicherkarte war voll mit Bildern und Videos. Die als Betriebssystem benutzte Distribution *MotioneyeOS* reagiert dann mit endlos wiederholten Bootvorgängen. Also befreite ich die Karte vom Ballast, indem ich die Bilder auf meinen PC überspielte und setzte alles wieder zusammen. Frau Meisenkaiser kehrte glücklicherweise kurz danach wieder zurück. Ich musste mir also etwas einfallen lassen, um notfalls ohne Öffnen an die Speicherkarte heranzukommen.

Mitte Mai waren die fünf Küken dann ausgewachsen 3, flogen aus und kehrten nie mehr ins Nest zurück. Die Familie siedelte sich auf der Dachterrasse des Hauses an und befreite die Bäume in der Nachbarschaft von zahlreichen Raupen. Ende des Jahres nahm ich den Kasten daher ab, säuberte ihn und begann mit der Modernisierung. Als Ergebnis entstand die folgende Bauanleitung, mit der Sie Ihren bereits vorhandenen oder auch einen neuen Nistkasten nach letztem Stand der Meisentechnik (um-)bauen können.

## Holzarbeiten

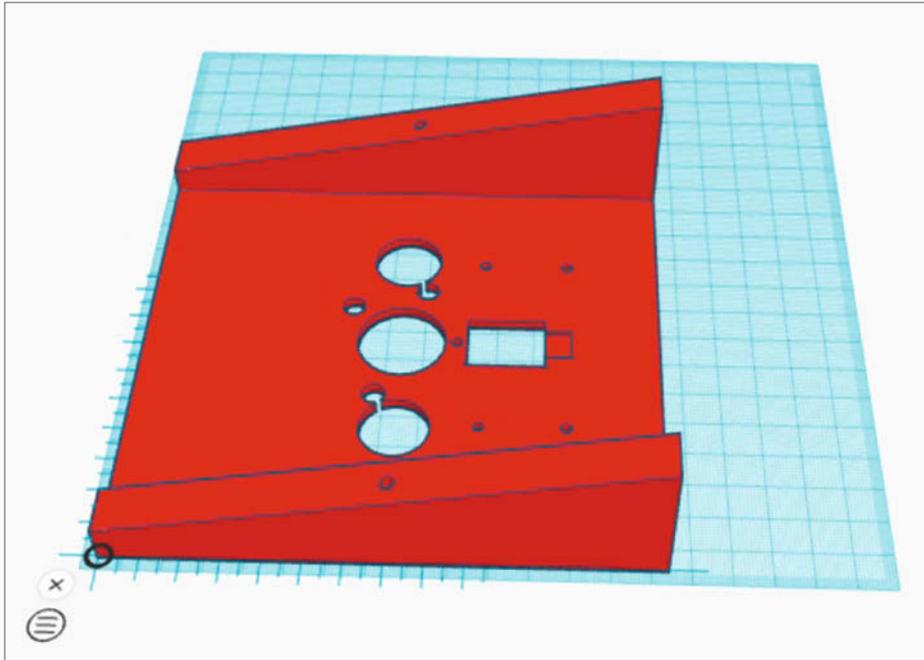
Das Äußere des Nistkastens ist gegenüber der ersten Version unverändert, bis auf die Löcher im Boden. Im Inneren musste der Halter für die Kamera und den Raspberry angepasst werden. Dafür finden Sie über den Kurzinfolink Zeichnungen, die Sie am besten im Maßstab 1:1 drucken, per Kohlepapier auf eine Sperrholzplatte übertragen und dann etwa mit einer Laubsäge aussägen sollten. Für Besitzer eines 3D-Druckers stehen auch entsprechende Druckdateien zur Verfügung 4.

Die Zeichnungen für den Kasten selbst mit allen Maßangaben finden Sie ebenfalls über den Link 5.

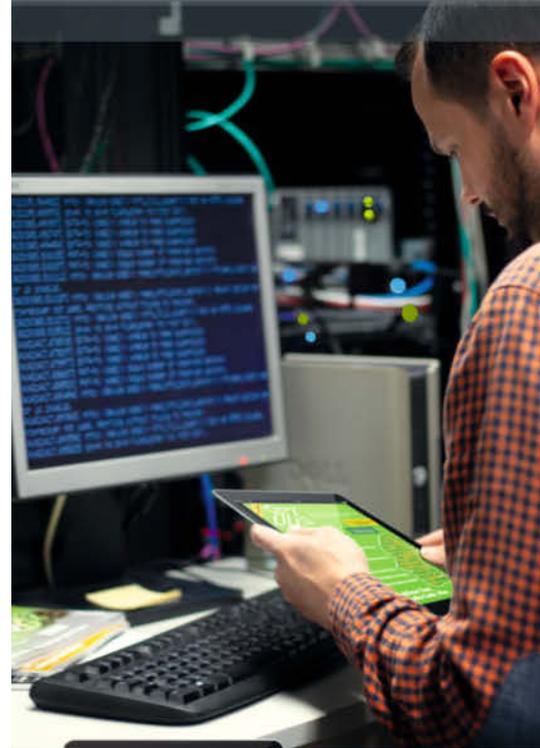
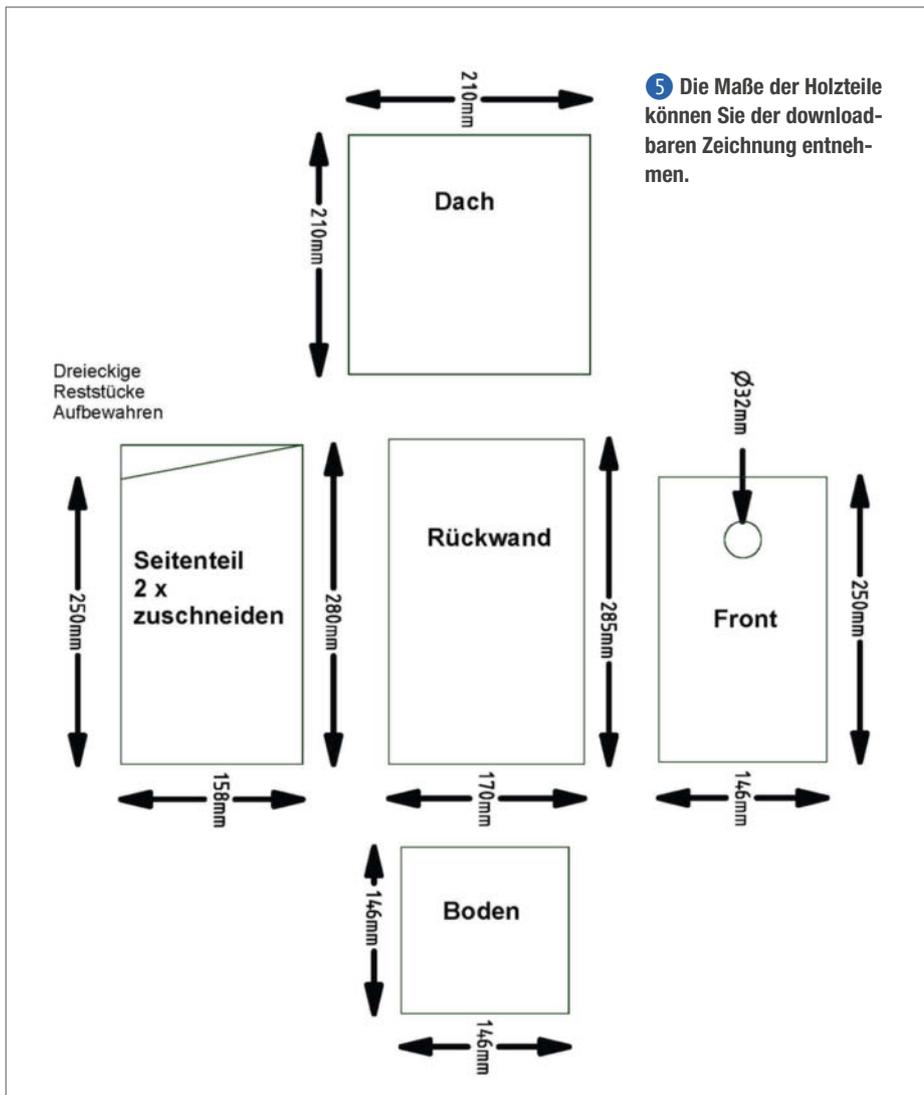
Zuerst schrauben Sie die beiden LED-Leuchten an das Kameramodul, stecken das Kabel für den Raspberry ein und bauen das Modul in den Halter ein. Ich habe es mit Heißkleber befestigt. Wichtig: Achten Sie darauf, dass es gerade sitzt, sonst schielt die Kamera später zur Seite. Der Temperaturfühler DHT22 (notfalls geht auch ein DHT11, der ist aber etwas ungenauer) wird mit einer M3-Schraube nebst

Es gibt **10** Arten von Menschen.

iX-Leser und die anderen.



4 Der Kamera-/Elektronikhalter kann auch als 3D-Druck gefertigt werden.



Jetzt Mini-Abo testen:

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

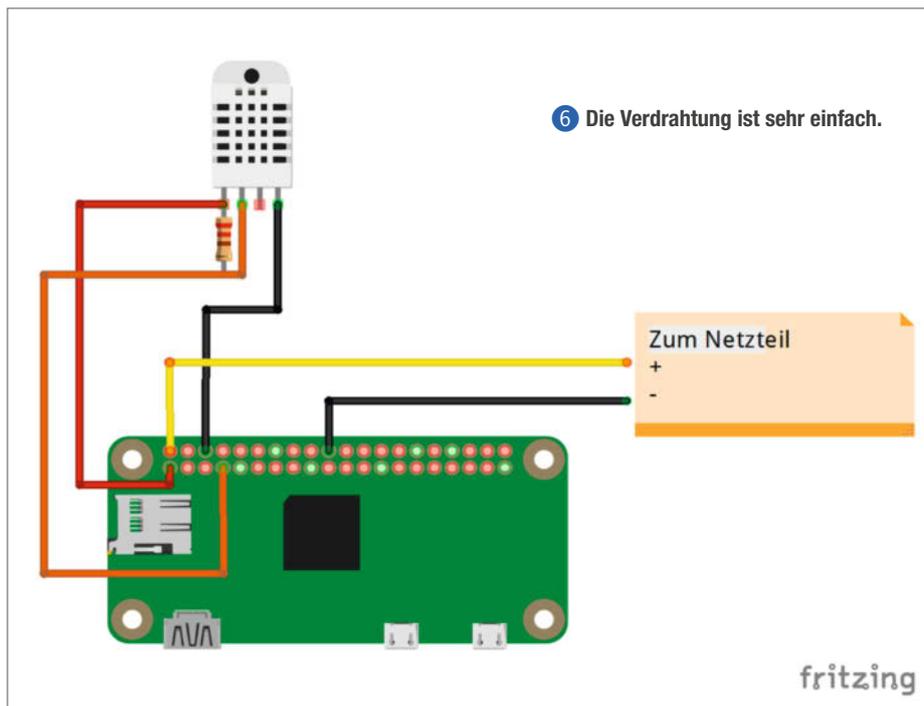
[www.ix.de/digital-testen](http://www.ix.de/digital-testen)



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

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

49 (0)541 800 09 120



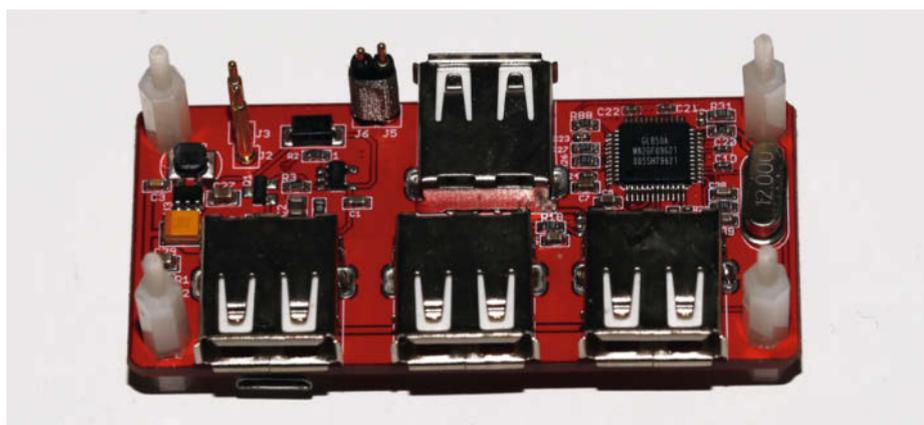
Mutter eingesetzt. Die Seite mit den Luftschlitzen muss auf der selben Seite wie das Kameraobjektiv liegen.

Danach löten Sie den 10kΩ-Widerstand an die Sensoranschlüsse und die Anschlussdrähte (je etwa 15cm lang). Verbinden Sie dann alles mit dem Raspberry entsprechend dem Schaltplan 6.

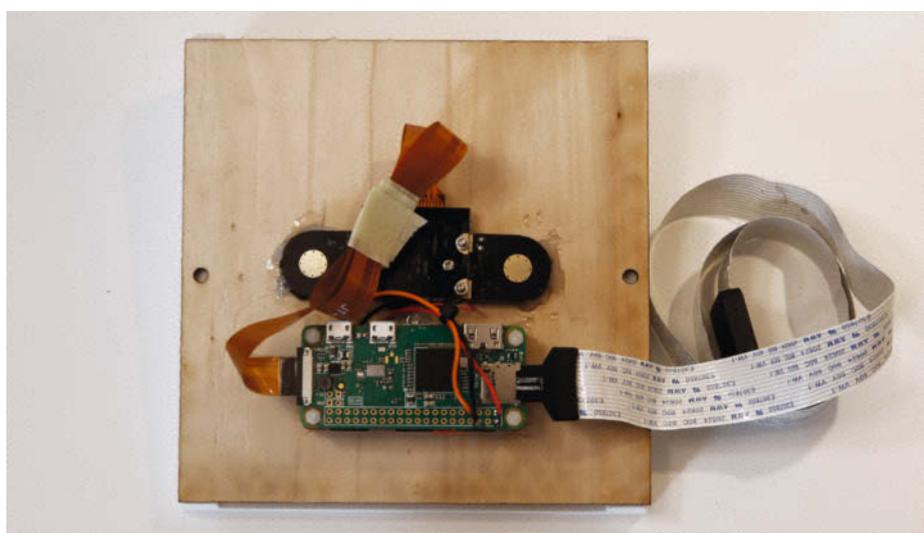
Falls Sie eine zusätzliche externe Kamera anschließen möchten, geschieht dies am besten mit einem USB-Hub 7, der auf den Raspberry Zero aufgeschraubt und dabei über Federkontakte (auch Pogo-Kontakte genannt) alle erforderlichen Verbindungen herstellt. Man könnte stattdessen zwar auch einen OTG-Adapter für die Mikro-USB-Buchse verwenden. Doch zeigte sich, dass deren Stecker zu groß sind und nicht ohne Entfernen ihres Gehäuses in den Nistkasten passen.

Bei Verwendung des Hubs müssen Sie eventuell kleine Leisten und die dreieckigen Halter im Deckel unterlegen, weil die Elektronik sonst zu dicht unter dem Deckel säße.

Danach kann der Raspberry mit M2,5-Schrauben auf die Trägerplatte montiert werden. Bleibt noch das Problem mit der Zugänglichkeit der Speicherkarte ohne Öffnen des Deckels. Hier hatte ich zunächst angedacht, dem Raspberry Zugriff auf einen Samba-Server zu ermöglichen, auf den er dann mithilfe eines Cron-Skripts täglich die aufgenommenen Videos und Fotos überträgt. Das ist problemlos möglich, erfordert aber, dass der Server dann auch läuft. Da aber sicher nicht jeder solch einen Server (oder einen PC mit entsprechender Windows-Freigabe) täglich zur selben Zeit laufen lassen möchte, habe ich eine andere Lösung gewählt: eine simple Verlängerung des SD-Kartenslots. Damit kann ich die Speicherkarte nach außen legen und komme so im Problemfall heran. Wer möchte, kann sich auch zwei Speicherkarten für den RasPi einrichten und dann bei Bedarf die Karten wechseln und die Bilder von der entnommenen Karte ganz in Ruhe am PC auf Festplatte speichern (Übrigens: Bei meinen Meisen ergaben sich während der Brutsaison etwa 1,2 Terabyte an Bildern und Videos!). Diese Methode erscheint mir zuverlässiger. Also stecken wir nun noch den Stecker der Verlängerung in den RasPi-Kartenslot und sind so mit der Nistkasten-Elektronik fertig 8.



7 Mit Hilfe dieses USB-Hubs zum Aufschrauben haben Sie die Möglichkeit, bis zu vier USB-Geräte (zum Beispiel Kameras) an den Raspberry Zero anzuschließen.



8 Die gesamte Elektronik des Nistkastens

### Software-Einrichtung

Beim Betriebssystem ging ich diesmal einen anderen Weg: Statt wie in der ersten Version die Distribution *MotioneyeOS* zu verwenden, habe ich diesmal zunächst *Raspberry Pi OS Lite (32bit)* installiert und *Motioneye* nachträglich als Softwarepaket hinzugefügt. Vorteil dieses Vorgehens: Die üblichen Paketquellen für Raspbian stehen zur Verfügung und man kann einfacher Software hinzufügen, beispielsweise

für die Temperaturmessung. Die Betriebssysteminstallation auf der SD-Karte habe ich mit dem *Pi Imager* erledigt. Sie finden die richtige BS-Version dort unter *Raspberry Pi OS (other)*.

Nach dem Beschreiben der Karte lassen Sie sie noch im PC. Fügen Sie ins Hauptverzeichnis der Karte bitte eine leere Datei namens *ssh* (ohne Dateierweiterung) hinzu. Das ermöglicht später den Zugriff auf den RasPi per Terminalprogramm wie *Putty*. Außerdem legen Sie die Datei *wpa\_supplicant.conf* [9](#) an, die die Zugangsdaten Ihres WLANs enthalten muss.

Für SSID setzen Sie den Namen Ihres WLANs ein, für *Passwort* gilt das entsprechend. Damit ist die Speicherkarte nun bereit für den ersten Start im Raspberry. Stecken Sie sie in den Kartenslot der Verlängerung, schließen Sie, falls gewünscht, auch die externe USB-Kamera an und stecken Sie das Nistkasten-Netzteil in eine Steckdose. Geben Sie dem Raspberry nun ein paar Minuten Zeit und schauen Sie dann auf Ihrem Router nach, welche IP-Adresse der Raspberry (er meldet sich mit dem Namen *raspberrypi*) bekommen hat. Bei der Gelegenheit sollten Sie den Router auch gleich so einstellen, dass er dem RasPi stets dieselbe IP-Adresse zuordnet.

Die Installation der weiteren Software-Pakete geschieht nun über eine ssh-Verbindung, zu der ich das Programm *Putty* benutze (Download siehe Kurzinfor-Link). Starten Sie es, geben Sie als Adresse die IP des RasPis ein und starten Sie die Verbindung. Die Sicherheitsfrage beantworten Sie per Klick auf *Ja*. Als Benutzername geben Sie *pi* und als Passwort *raspberrypi* ein (daran denken, dass das amerikanische Tastaturlayout benutzt wird, x und y also vertauscht sind). Danach können Sie die Befehle aus dem Listing [10](#) einzeln nacheinander eingeben und so die fehlenden Programme installieren. Fragen wie *Do you want to continue?* beantworten Sie mit *y*. Die Befehle brauchen einiges an Zeit, da reichlich Daten aus dem Internet geholt werden müssen. Sollte beim Befehl

```
sudo pip install motioneeye
```

die Meldung *Killed* auftauchen, dann müssen Sie anschließend erst

```
sudo apt-get install python-pillow
```

eingeben, danach den Befehl

```
sudo pip install motioneeye
```

wiederholen und mit den restlichen Anweisungen fortfahren.

Als Nächstes müssen Sie den Onboard-Kameraanschluss des Rasperrys aktivieren. Mit dem Befehl

```
sudo raspi-config
```

gelangen Sie ins Setup-Programm. Wählen Sie darin mit den Cursor- und der Eingabetaste

## 9 wpa\_supplicant.conf

```
01 country=DE
02     update_config=1
03     ctrl_interface=/var/run/wpa_supplicant
04
05     network={
06         scan_ssid=1
07         ssid="SSID"
08         psk="Passwort"
09     }
```

## 10 Installationsbefehle

```
01 sudo apt-get update
02 sudo apt-get install motion ffmpeg v4l-utils
03 sudo apt-get install python-pip python-dev curl libssl-dev libcurl4-
04     openssl-dev libjpeg-dev
05 sudo pip install motioneeye
06 sudo mkdir -p /etc/motioneye
07 sudo cp /usr/local/share/motioneye/extra/motioneye.conf.sample /etc/
08     motioneye/motioneye.conf
09 sudo mkdir -p /var/lib/motioneye
10 sudo cp /usr/local/share/motioneye.systemd-unit-local /etc/systemd/
11     system/motioneye.service
12 sudo systemctl daemon-reload
13 sudo systemctl enable motioneeye
14 sudo systemctl start motioneeye
```

# Temperaturanzeige

Der Sensor DHT22 kann die Temperatur und die Luftfeuchtigkeit messen und über ein 1-Wire-Interface übertragen. 1-Wire heißt, dass außer der Masseleitung nur ein Anschluss für Stromversorgung und Datenübertragung gebraucht wird. Dies wird hier aber nicht benutzt, sondern Daten und Strom werden getrennt übertragen.

Haben Sie den Sensor wie im Schaltbild [6](#) verkabelt, brauchen Sie noch etwas Software, um die gemessenen Werte auszulesen. Die installieren Sie im *Putty*-Konsolefenster mit den Befehlen

```
sudo apt-get install build-essential python-openssl
```

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

```
adafruit/Adafruit_Python_DHT.git
```

```
cd Adafruit_Python_DHT
```

```
sudo python setup.py install
```

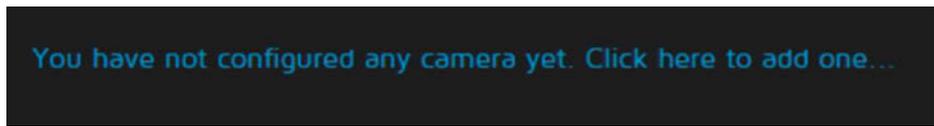
Zum Auslesen der Werte genügt dann

```
python ~/Adafruit_Python_DHT/
examples/AdafruitDHT.py 22 4
```

Das wird dann recht unspektakulär wie im Bild unten angezeigt.

Da ich die Werte nur ab und zu bei hohen Außentemperaturen auslesen möchte, genügt mir dieser Befehl. Wer möchte, kann ihn aber auch in ein Skript mit einfachem Namen packen. Beachten Sie dabei aber, dass man den Sensor nicht in kürzeren Abständen als 4 Sekunden auslesen darf.

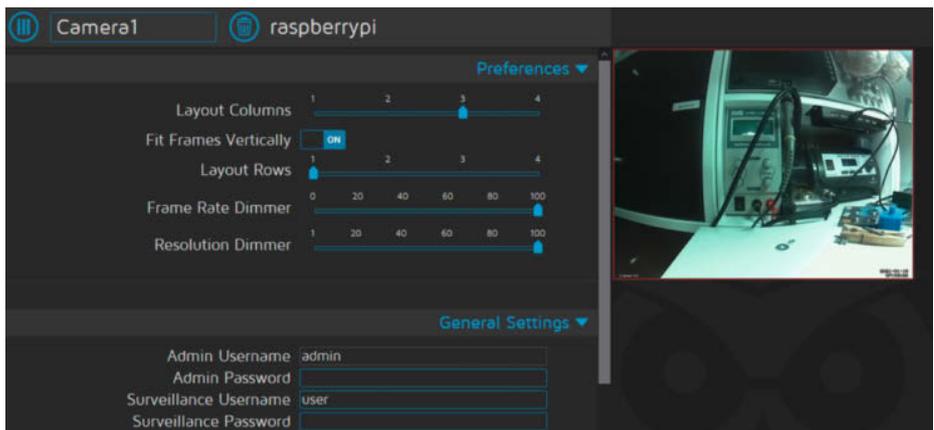
```
pi@raspberrypi:~/Adafruit_Python_DHT/examples $ python AdafruitDHT.py 22 4
Temp=35.7* Humidity=26.9%
pi@raspberrypi:~/Adafruit_Python_DHT/examples $ █
```



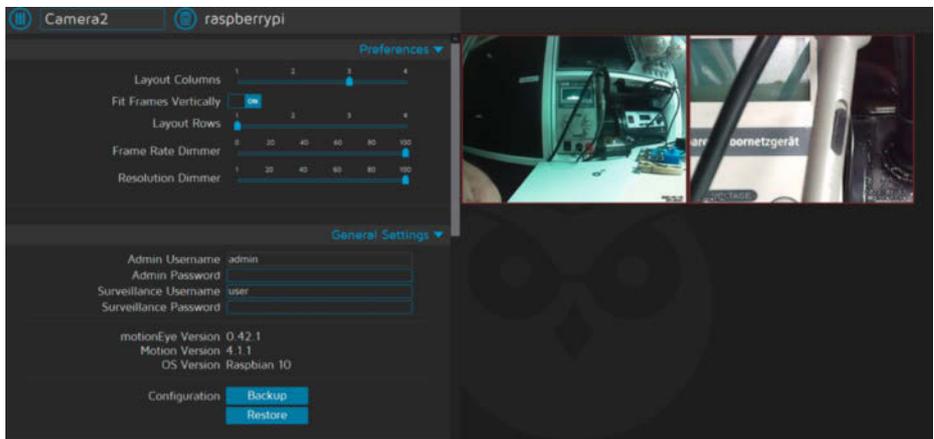
11 Per Klick auf diese Meldung gelangen Sie zur Kamera-Konfiguration.



12 Mit diesen Einstellungen erreichen Sie die Raspi-Kamera.



13 Das erste scharfe Bild: Per Klick auf den Kamera-Namen können Sie weitere Kameras hinzufügen.



14 Die Bilder beider Kameras: Das Superweitwinkel-Objektiv der internen Kamera verzerrt das Bild ein wenig, dafür nimmt es später aber das komplette Nest auf.

den Punkt 3 und dann P1. Die dann folgende Frage beantworten Sie mit Yes, bestätigen mit OK und Finish. Dann lassen Sie den Raspberry neu booten.

Von nun an können Sie in Ihrem Browser den Raspberry über

ip-Adresse: 8765

erreichen. Für IP-Adresse setzen Sie die Adresse Ihres RasPis ein. Sie sehen aber lediglich das Login-Fenster. Geben Sie als Username Admin ein, ein Passwort ist nicht erforderlich. Als Nächstes müssen Sie die Kamera(s) konfigurieren. Klicken Sie dazu auf die Meldung 11 im Browser-Fenster.

Im Konfigurationsfenster aktivieren Sie zunächst das IR-Kameramodul. Achten Sie darauf, dass die Einstellungen wie in Bild 12 gewählt sind.

Halten Sie die Kamera nun mit etwa 15cm Abstand vor einen Gegenstand und stellen Sie durch Drehen des Objektivs ein scharfes Bild ein. Danach klicken Sie auf das Feld mit Camera1 13.

Wählen Sie add Camera und als Camera Type dann Local V4L2 Camera. Im Feld Camera wählen Sie dann den Eintrag USB2.0 PC Camera. Der letzte Eintrag kann mehrfach vorkommen oder auch, je nach Kamerafabrikat, etwas anders aussehen. Da müssen Sie notfalls mehrere Testdurchgänge machen, um den richtigen Eintrag zu finden. Mit dem Papierkorb-Symbol neben dem Kamera-Namen können Sie unnötige Kameraeinträge wieder entfernen.

Stellen Sie schließlich auch das Bild der zweiten Kamera scharf 14.

Nun können Sie die Elektronikplatte an den Deckel des Nistkastens schrauben und den dann mit den Scharnieren am Vogel-Eigenheim befestigen. Die Oberkante des Nistkastens habe ich mit Resten von Schaumstoff-Fensterdichtungen beklebt. Die schützen nicht nur vor Feuchtigkeit, sondern erlauben es auch, das Speicherkarten- und das Stromkabel sowie gegebenenfalls das USB-Kabel der zweiten Kamera nach außen zu führen.

Der Nistkasten ist damit nahezu bezugsfertig. Sie müssen ihn nur noch an geeigneter Stelle aufhängen und die zweite Kamera im Abstand von etwa 30 bis 40cm seitlich des Einfluglochs anbringen. Schlitz am Gehäuse der USB-Kamera, durch die Regenwasser eindringen könnte, habe ich übrigens mit Silikon aus dem Baumarkt abgedichtet. Das Gehäuse für die Speicherkarte brachte ich einfach in einem Kunststoffbeutel unter, der mit Tesafilm wasserdicht verschlossen wurde. Dieser Verschluss lässt sich zum Kartenwechsel aber wieder leicht öffnen.

Dann bleibt nur das Warten auf interessierte Mieter, die sie dann demnächst online beobachten können. —hgb

# DEVELOPER-KONFERENZEN + WORKSHOPS 2021



## betterCode()

**Domain-Driven Design:** 25.02.2021  
Online



16. bis 18.03.2021  
Online

## » Continuous Lifecycle »

**Dev(Sec)Ops Day:** 03.03.2021  
Online

## // heise devSec()

18.03.2021 (Thema folgt)  
Online

## betterCode()

**APIs:** 11.03.2021  
Online



24. – 25.3.2021  
Online

Veranstalter:



@ heise Developer



dpunkt.verlag

© Copyright by Maker Media GmbH.

Weitere Informationen unter:

[www.heise.de/developer](http://www.heise.de/developer)

# Hydroponikanlage mit ESP32-Steuerung

Auch mit wenig Platz lassen sich Salat und Gemüse selbst anziehen und automatisch im geschlossenen Kreislauf bewässern. Mit Teilen aus dem Baumarkt und einem ESP32 als Steuerung ist diese Anlage schnell gebaut und individuell anpassbar.

von Florian Grimm



**M**aker, die trotz wenig Platz von selbst angebautem Gemüse und eigenen Kräutern träumen, sollten sich einen vertikalen Aufbau an der Wand ansehen. Mit einer Photovoltaikzelle lässt sich das System autark mit Energie versorgen, während ein Anschluss ans Haus-WLAN Kontrollmöglichkeiten bietet. Zum Bau so einer Hydrokulturanlage reichen einfache Baumarktmaterialien.

## Grundprinzip Hydrokultur

Bei der Gestaltung des Systems orientiert man sich zunächst an der Natur: Pflanzen brauchen nur Licht, Luft, Wasser und einige Nährstoffe, damit sie wachsen und Biomasse aufbauen. Als Nebeneffekt sprießen manchmal Früchte, Blätter oder Samen, die wir essen können. Klassischerweise geschieht das Wachstum in der Erde. Sie gibt den Wurzeln eine Stützstruktur, speichert Feuchtigkeit und versorgt die Pflanze mithilfe von Mikroorganismen mit Nährstoffen. Da dieses System nicht geschlossen ist, kann allerdings Wasser in den Boden versickern oder Nährstoffe auswaschen, außerdem ist der Anbau in Erde sehr platzintensiv.

Hydrokultur versucht daher, einen geschlossenen Wasser- und Nährstoffkreislauf herzustellen. Hier wachsen die Pflanzen in einem anorganischen Substrat wie Kies oder Blähton, das vor allem als Stützstruktur dient. Die Versorgung mit Nährstoffen und Wasser erfolgt über eine Nährlösung, die die Wurzeln umspült. Die Nährstoffe werden lediglich in der benötigten Menge eingebracht und gelangen nicht in die Umwelt. Da das Wasser nicht versickert, ist auch der Wasserbedarf erheblich niedriger. Lediglich der Teil, der durch Verdunstung und Pflanzenwachstum entzogen wird, muss ersetzt werden. Mithilfe künstlicher Beleuchtung und Beheizung kann der Pflanzenanbau so unter Extrembedingungen erfolgen, wie auf der Internationalen Raumstation oder auf der Neumayer-Station in der Antarktis.

## Aufbau des Projektes

Als Grundstruktur für mein Projekt auf dem heimischen Balkon nutze ich die bewährten HT-Rohre aus dem Baumarkt. Diese sind günstig, überall verfügbar, individuell konfigurierbar, gut zu bearbeiten und wasserdicht. Viele Zubehörteile wie Deckel, Übergänge, Winkel und Wandhalterungen gibt es für wenig Geld von der Stange. Die Struktur habe ich mit verschiedenen Teilen aus dem 3D-Drucker ergänzt.

Die Rohre bilden ein wasserleitendes Grundgerüst. Aus einem Vorratstank (Eurobox aus dem Baumarkt) pumpt eine kleine Pumpe die Nährlösung in das oberste Rohr, von dort aus fließt sie an den Pflanzen vorbei in einen Sandfilter, der ebenfalls in einem HT-

## Kurzinfo

- » Grundgerüst aus HT-Rohren bauen
- » Steuerung mit ESP32
- » Google Charts in Homepage einbinden

### Checkliste

-  **Zeitaufwand:**  
ca. drei Tage (ohne Druckzeit)
-  **Kosten:**  
300 Euro (ohne autarke Stromversorgung nur 220 Euro)
-  **Maschinen:**  
3D-Drucker, Lochsäge, Forstnerbohrer
-  **Löten:**  
einfache Lötarbeiten
-  **Programmieren:**  
Arduino IDE

### Material

- » Die Materialliste finden Sie online unter dem Kurzlink.

### Mehr zum Thema

- » Florian Schäffer, Umweltsensoren für Mikrocontroller, Make 1/19, S. 8
- » Elke Schick, Alleskönner Abflussrohr, Make 3/18, S. 76
- » Make: ESP32 Special, 2019

↓ Alles zum Artikel im Web unter [make-magazin.de/xj99](https://make-magazin.de/xj99)

Rohr untergebracht ist. Vom Filter aus wird die Flüssigkeit schließlich zurück in den Tank geleitet. Die Energieversorgung erfolgt autark über eine Solarzelle und eine Autobatterie. Die Steuerung lasse ich zusammen mit einem Webserver über einen ESP32 laufen, der auch Temperatur- und Feuchtigkeitssensoren auswertet.

Prinzipiell nimmt die Anlage durch die wandhängende Montage wenig Platz in Anspruch. Der Standort sollte jedoch genau be-

dacht und gegebenenfalls optimiert werden. Denn auch wenn Pflanzen Licht brauchen, sollte pralle Sonne vermieden werden. Sonst verdunstet über die Pflanzen zuviel Wasser und in heißen Sommern verbrennen die Blätter. Gerade junge Pflanzen sind direkt nach dem Ausbringen in die Hydrokultur sehr sensibel. Ein weiterer Nachteil ist, dass bei zu hohen Temperaturen die Nährlösung überhitzt und ihre Sauerstoffsättigung sinkt. Ebenso sollte der Standort nach Möglichkeit wind-



Bild: NASA

Hydrokulturrexperimente in der Schwerelosigkeit auf der Internationalen Raumstation



Bild:DLR (CC-BY 3.0)

Forscher Paul Zabel mit der ersten Salaternte der Antarktis

und wettergeschützt sein. In der Übergangszeit im Frühjahr muss man ansonsten bei Starkregen oder Hagel mit Beschädigungen der Pflanzen rechnen. Wer nur beschränkte Aufstellmöglichkeiten hat, sollte beispielsweise mit einem Sonnensegel den Schutz gegen die Elemente optimieren.

### Hardware

Zu den 3D-gedruckten Teilen gehören Wasserstandsbegrenzer und Pflanztöpfe. Optional sind ein Gehäuse für die Unterbringung der Hauptplatine und die Verschlüsse für die Griffe der Eurobox, in denen ich eine Konnektor-

platine für die Pumpe untergebracht habe. Schließlich empfehle ich ein paar Deckel zu drucken, mit denen nicht benötigte Pflanzöffnungen verschlossen werden. Es gibt sie in zwei Varianten: Eine passt auf die Pflanztröge, die andere kann auf das HT-Rohr gesteckt werden. Die 3D-Druckteile finden Sie über den Kurzlink zum Download, ebenso wie eine Materialliste mit den jeweiligen Typenbezeichnungen der Baumarktteile. Ich habe alles mit PLA gedruckt und auch nach zwei Jahren Außeneinsatz keine Probleme.

Eine Anpassung des Aufbaus an den eigenen Balkon ist dank der breiten Auswahl an HT-Rohren flexibel möglich. Für den Anfang empfehle ich eine überschaubare Anzahl an Windungen und Pflanzplätzen. Die Rohre habe ich einfach zusammengesteckt – sie haben integrierte Dichtungen und können mit einer Handsäge passend gekürzt werden. Übereinanderliegende Segmente werden durch vier 45°-Winkel-Rohrstücke verbunden. Zur Planung der Bohrungen für die Pflanztröge eignet sich die aufgedruckte Markierung. Ich habe im Abstand von 15 cm erst vorgebohrt und die Löcher dann mit einer 45mm Lochsäge langsam und vorsichtig eingebracht. Die Markierung kann dann für eine schönere Optik mit verdünntem Aceton entfernt werden.

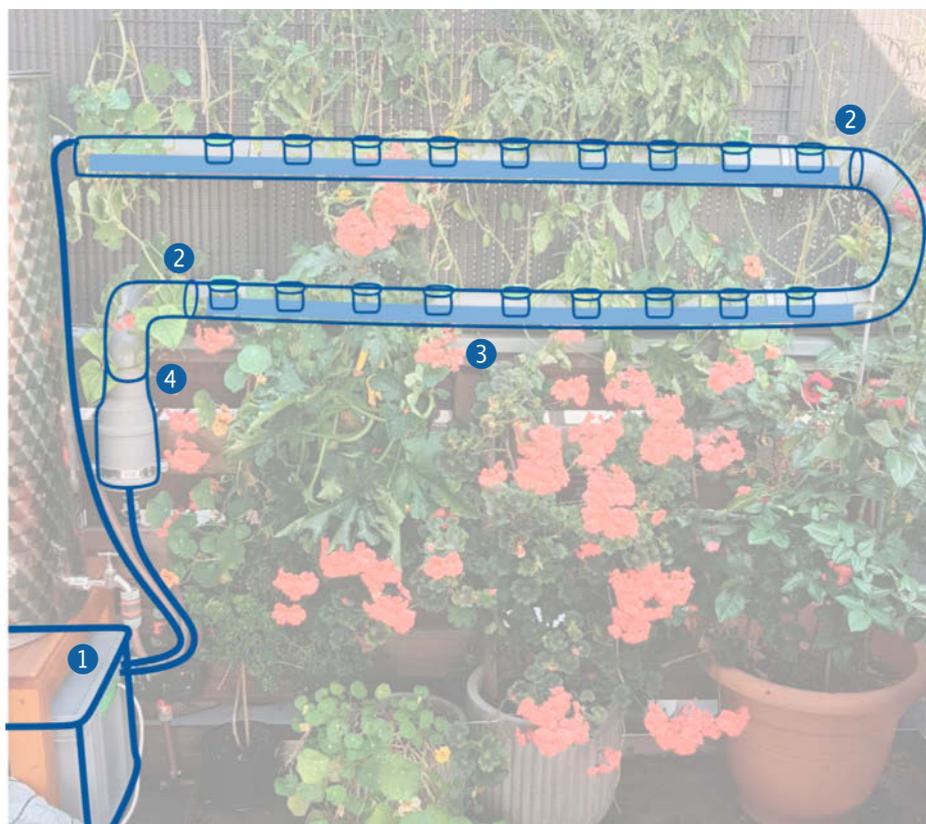
### Wasserstand

Da die Nährstoffliquidität nur in Intervallen ersetzt werden soll, muss beim Aufbau stets auf wasserdichte Verbindungen geachtet werden. Innerhalb der Rohre soll außerdem ein kontinuierlicher Flüssigkeitsstand gewährleistet sein. Dafür habe ich Wasserstandsbegrenzer mit 100 Prozent Füllung 3D-gedruckt. Mit einem O-Ring versehen werden sie in die HT-Rohre gesteckt, wobei HT-Gleitmittel aus dem Baumarkt hilft. Um den Wasserstand zu regulieren, kann der Begrenzer später einfach im HT-Rohr gedreht werden.

Das oberste Rohr habe ich mit einem modifizierten Muffenstopfen verschlossen, durch den die Wasserzuführung erfolgt. Dazu habe ich in die Oberseite ein 20mm großes Loch gebohrt und einen Hahnanschluss (3/4" IG) mit einem innenliegenden Regentonnenanschluss (3/4" IG, 1" AG) verbunden. Wichtig ist die sichere Verschraubung mit einem Dichtungsring, die den Hebelkräften beim Ein- und Ausstecken des Schlauches standhalten muss.

### Wandmontage

Mit HT-Schellen kann das System nun an einer Terrassenwand oder am Mauerwerk montiert werden. Die Schellen müssen dabei möglichst waagrecht positioniert sein, da später keine größere Nivellierung mehr möglich ist. Damit sich abgestorbene Pflanzenteile nicht absetzen und das geschlossene System verstopfen, ist eine kontinuierliche Filterung unabding-



Die Nährliquidität wird mit einer 12-Volt-Pumpe im Tank 1 über das Schlauchsystem in das obere HT-Rohr gepumpt. Über einen Wasserstandsbegrenzer 2 wird ein einheitlicher Füllstand gewährleistet. Durch die Schwerkraft fließt die Nährliquidität vorbei an den Pflanztrögen 3 bis zum Sandfilter 4 und von dort in einem geschlossenen Kreislauf wieder zurück in den Vorrattank.

bar. Den Filteranschluss habe ich mit einem 87°-Bogen und einem 45°-Bogen vorbereitet. Um einen höheren Querschnitt für den Filter zu ermöglichen, nutze ich schließlich eine HT-Reduzierung DN 110/75.

Der Sandfilter ist platzsparend in einer DN-110-Muffe untergebracht. Zuerst habe ich den HT-Muffenstopfen (DN 110) mittig mit einem 25mm großen Forstnerbohrer durchbohrt. Hier wird ein weiterer Regentonnenanschluss ( $\frac{3}{4}$ " IG, 1" AG) mit Dichtung fest eingeschraubt. Auf die Innenseite kommt ein Saugkorb ( $\frac{1}{2}$ " AG), sodass kein größeres Material aus dem Sandfilter in das Schlauchsystem gelangen kann. Außen habe ich einen Hahnanschluss (1" IG) aufgeschraubt, auf den der Ablaufschlauch gesteckt wird. Die Befüllung des Sandfilters erfolgt in drei Schichten mit grobem und mittlerem Aquarienkies und einer Lage Sand. Nun kann der Filter an das Reduzierstück gesteckt werden. Aufgrund seines Gewichts sollte er mit einer weiteren Schelle an der Rückwand fixiert werden.

Als Tank für die Nährstofflösung dient eine Stapelbox mit Deckel, die 64 Liter fasst. Die Griffstücke habe ich mit 3D-Druckteilen verschlossen. In einem Griff ist zusätzlich eine kleine Platine für die Spannungsversorgung der Pumpe untergebracht. Direkt daneben sind zwei Anschlüsse für die Wasserdurchführung. So kann die Box bis knapp unterhalb der Griffe befüllt werden.

Da der Platz durch die Stege sehr begrenzt ist, habe ich die Bohrlöcher (15 mm) genau ausgemessen und aufgezeichnet. Bei den Schlauchanschlüssen ist Kreativität gefragt: Die Hahnanschlüsse ( $\frac{3}{4}$ " IG) habe ich mit Schlauchverschraubungen ( $\frac{3}{4}$ " AG) kombiniert. Diese werden von außen durch die Bohrung gesteckt und innen mit den Schläuchen und einer Rohrschelle verbunden. Den ablaufenden Schlauch habe ich auf Bodenni-



Vorbohren der HT-Rohre (oben), dann langsames Aussägen der Pflanzhalterungen mit einer Lochsäge (unten)

veau gekürzt und am Zulauf die Wasserpumpe mit weiteren Rohrschellen gesichert. Für die Ansaugung habe ich noch ein weiteres kurzes Schlauchstück mit einer Verschraubung ( $\frac{3}{4}$ " IG) und einem Saugkorb angefügt. Nun ist die Hardware vollständig aufgebaut.

## Elektronik

Im Griffstück des Tanks ist eine 2cm x 8cm Lochrasterplatine untergebracht, die Platz für zukünftige Erweiterungen wie Sensoren oder Füllstandsschalter bietet. Aktuell wird damit nur die Spannungsversorgung der Pumpe auf



Eingesetzter Wasserstandsbegrenzer im HT-Rohr: Der Flüssigkeitsspiegel wird durch Drehen des Begrenzers eingestellt.



Montageteile der Wasserzuführung



Fertig montierte Grundstruktur der Hydrokultur



Innenliegender Saugkorb im Filter



Befüllung des Sandfilters mit Kies und Sand

eine 2,54mm Buchsenleiste gelegt und in den Tank geführt. Eine zusätzliche Freilaufdiode fängt die Induktionsströme beim Abschalten der Pumpe ab.

Die Stromversorgung des Projektes habe ich über eine kleine Autobatterie mit 50Ah und einem 10-Watt-Solarpanel gesichert. Dadurch läuft es autark, kostet aber um die 100 Euro mehr. Alternativ kann man den Aufbau über ein günstigeres 12-Volt-Netzteil realisieren. Der Basisstromverbrauch bei eingeschaltetem WLAN und laufendem Webserver liegt bei 0,3 bis 0,4 Watt. Der größte Anteil des Gesamtverbrauchs entfällt auf die Wasserpumpe mit 12 Watt im Betrieb. Allerdings wird die Pumpe je nach Größe des Aufbaus und je nach Außentemperatur nur alle 30 bis 180 Minuten etwa 20 bis 30 Sekunden betätigt. Die Verbindung von Solarpanel, Batterie und Elektronik übernimmt ein herkömmlicher Solarladeregler. Auf dem eigentlichen Mainboard werden dann 5 Volt über einen Step-Down-Regler und die 12-Volt-Batteriespannung bereitgestellt. Sämtliche Komponenten finden in einem 3D-gedruckten Gehäuse Platz.

### Hauptplatine

Das Herzstück der Elektronik ist das ESP32-Board auf der Hauptplatine, das vom Step-Down-Regler mit 5 Volt versorgt wird. Die Pumpe wird mit 12 Volt betrieben, während die eigentliche Steuerung über Pin D19 erfolgt, mit einem 470-Ohm-Vorwiderstand (R14) und einem TIP120-Transistor. Dieser Aufbau ist nicht zum Dauerbetrieb geeignet, bei nur bis zu 30 Sekunden Schaltleistung ist jedoch keine Kühlung des Transistors notwendig. Die drei LEDs geben optische Statusmeldungen nach außen und sind an den Pins D12, D14 und D27 angeschlossen. Zur Temperatursteuerung habe ich einen DHT22-Sensor an Pin D13 verbaut. Die Messung der Batteriespannung er-

folgt über den Spannungsteiler mit R10 (47 kOhm) und R11 (10 kOhm) am Pin D32, einem Eingang des Analog-Digital-Wandlers.

Für zukünftige Erweiterungen gibt es einen 5-Kanal-Navigationsschalter an den Pins D25, 26, 34, 36 und 39. Dafür sind derzeit noch Pull-Down-Widerstände (R4 bis R8) vorgesehen, statt denen beim ESP32 aber auch die internen genutzt werden können. In der aktuellen Programmierung nutze ich nur einen Taster. Der weitere Sensoreingang wird ebenfalls noch nicht verwendet, wobei ich eine 5-Volt-Spannungsversorgung und einen Spannungsteiler (R12 und R13 an D33) vorgesehen habe. Schließlich sind am oberen Ende des Boards weitere Leiterbahnstreifen für individuelle Erweiterungen übrig.

### Software

Ein Arduino-Sketch erledigt die Pumpensteuerung, die Abfrage der Messwerte und die Steuerung eines Webservers mit Status-Homepage, über den die Anlage im heimischen Intranet überwacht werden kann. Im Folgenden werde ich nur exemplarisch auf wichtige Funktionsabschnitte eingehen. Das komplette Skript ist online (siehe Downloadlink) abrufbar, wie auch Anleitungen zum Einbinden des ESP in die Arduino IDE. Beim Programmieren habe ich mich am *Make ESP32 Special* orientiert.

Das Öffnen einer WLAN-Verbindung ist mit den Bibliotheken *WiFi* und *WiFiClient* schnell und zuverlässig erledigt, wenn die Angaben SSID und PASSWORD mit den heimischen Einstellungen gefüllt sind. Der ESP32 wird dann im *Station-Mode* initialisiert (`WiFi.mode(WIFI_STA);`) und die Verbindung zum WLAN mit `WiFi.begin(SSID, PASSWORD);` hergestellt. Setzt man `DEBUG` auf `true`, erfolgt jeweils eine Statusausgabe über den seriellen Monitor über die Helferfunktion `debug(String Msg)`. Das ist vor allem am Anfang hilfreich, um alle Funktionen zu testen.

Zur Einstellung der Systemzeit nutze ich den *Network Time Protocol Server* `de.pool.ntp.org`. In `void setup` werden mit `configTzTime(TZ_INFO, NTP_SERVER);` die Zeitzone und der Servername übergeben. Danach erfolgt die Synchronisation über `getLocalTime(&local, 10000);`. In `void loop` wird die Zeit dann über `time(&now)` abgespeichert und mit `localtime_r(&now, &local);` zerlegt. Über die Struktur `local` lässt sich auf die Unterelemente der Zeitangaben zugreifen (`local.tm_mday, local.tm_hour, etc...`). Alle 24 Stunden erfolgt ein Update der Systemzeit, wenn die Bedingung `local.tm_mday != last-day` nicht mehr zutrifft.

### Messen und Pumpen

In der Hauptroutine (siehe Listing 1) werden außerdem regelmäßig die Temperatur-, Luft-



Griffstück mit Elektronik und Wasserein- und -ausgang. Das Kabel zur Steuerplatine wird durch den Deckel des Griffstücks nach außen geführt.



Anschlüsse zum ablaufenden Schlauch (oben) und 12-Volt-Pumpe (unten) im Innenraum der Tankbox

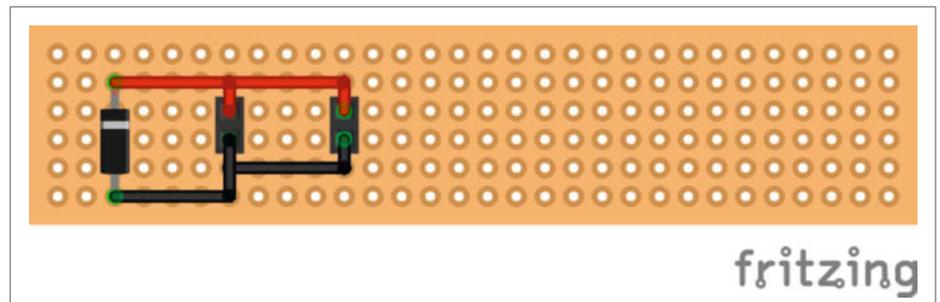
feuchtigkeits- und Spannungswerte ermittelt und nach einem Pumpvorgang in einen Ringpuffer geschrieben sowie das Pumpintervall festgelegt. In Zeile 2 bis 4 wird zunächst überprüft, ob seit der letzten Messung (`time_m`) mehr als 300 Sekunden (`tm`) vergangen sind. Ist dies der Fall, werden die Messergebnisse über die Funktion `messen()` aktualisiert und der WLAN-Status wird über die Funktion `wlanstatus()` geprüft. Ab Zeile 6 überprüft eine ähnliche Abfrage das Ablaufen des Gießintervalls (`interval1`). Soweit kein Fehler vorliegt (`error==0`), wird die Pumpe angeschaltet und das Gießintervall auf die Gießzeit `tp` gesetzt (in meinem Fall 25 Sekunden). Nach dem Gießen wird die Pumpe über die `else-if`-Abfrage abgeschaltet (ab Zeile 17) und die Messergebnisse in den Ringpuffer mit der Größe `int Punkte` geschrieben. Diese können später auf der Webseite ausgegeben werden. Je nachdem, wie heiß es gerade ist, wird außerdem das Pumpintervall neu festgelegt (Zeile 21 bis 27).

In `void messen()` liest der ESP über `dht.readHumidity()` die Luftfeuchte und über `dht.readTemperature()` die Temperaturwerte des DHT und speichert die Ergebnisse in `luftfeuchte` bzw. `temp`. Die am Spannungsteiler anliegende Spannung des Akkus wird über `analogRead(adcAkku)` gemessen. Mit 3,3 Volt multipliziert und durch die Auflösung 4095 geteilt sollte sie eigentlich genau berechenbar sein. In der Praxis arbeiten die Analog-Digital-Wandler des ESP32 weder sehr genau noch linear, weshalb ich eine anschließende Korrekturformel ( $V_{akku} = 1.11 + 5.56 * V_{ADC}$ ) ermittelt habe. Dafür habe ich bei unterschiedlich anliegenden Spannungen zwischen 10 und 15 Volt eine Eichkurve erstellt und daraus Steigung und y-Achsenabschnitt der Korrekturgleichung bestimmt. `VADC` gebe ich daher ebenfalls auf dem Webserver aus. Sollte die Batteriespannung auf unter 10 Volt sinken, wird `error=1`; gesetzt und die Pumpe nicht mehr betätigt, bis die Spannung wieder ansteigt.

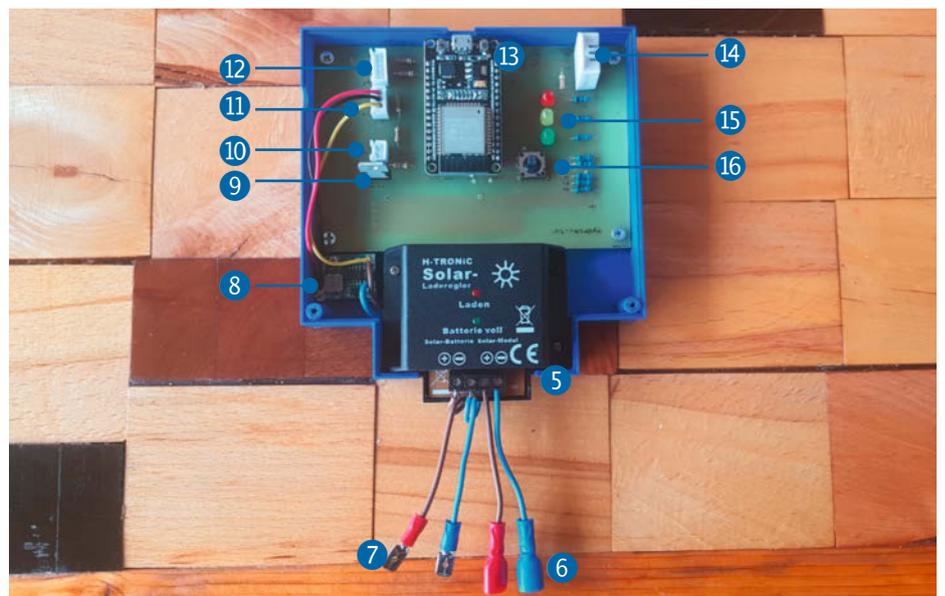
Mit der Funktion `void wlanstatus()` prüft der ESP schließlich, ob eine WLAN-Verbindung besteht (`wifi.status() != WL_CONNECTED`). Ist dies nicht der Fall, versucht er alle 30 Sekunden lang, eine neue Verbindung herzustellen.

## Server und Homepage

Für den Webserver kommen noch die Bibliotheken `Webserver` und `ESPmDNS` hinzu. Zunächst wird das `WebServer`-Objekt `server` auf Port 80 erzeugt. In Form von Eventhandlern



Schaltbild der Lochrasterplatine zur Verbindung der Hauptplatine mit der Pumpe im Tank



Aufbau der Elektronik: Solar-Laderegler 5, Anschlüsse des Solarpanels 6, Anschlüsse zur Batterie 7 – verbunden zum Step-Downregler 8, TIP120-Transistor 9, Pumpenausgang 10, 12- und 5-Volt-Eingänge 11, Sensoreingang 12, ESP32-Board 13, DHT22-Sensor 14, LEDs 15 und Navigationsschalter 16.

## 1. Hauptroutine

```

01 void loop() {
02   if (difftime(time_now, time_m)>tm){
03     messen();
04     wlanstatus();
05   }
06   if (difftime(time_now, time_p)>intervall){
07     time_p=time_now;
08     if (!pON){
09       intervall=92*60;
10       if (error==0){
11         pON=true;
12         intervall=tp;
13         digitalWrite(led, HIGH);
14         digitalWrite(pumpe, HIGH);
15       }
16     }
17     else if (pON) {
18       pON=false;
19       digitalWrite(led, LOW);
20       digitalWrite(pumpe, LOW);
21       if (temp<=10) {intervall=360;}
22       else if (temp>10 && temp<=15) {intervall=180*60;}
23       else if (temp>15 && temp<=20) {intervall=120*60;}
24       else if (temp>20 && temp<=25) {intervall=90*60;}
25       else if (temp>25 && temp<=30) {intervall=60*60;}
26       else if (temp>30) {intervall=30*60;}
27       else {intervall=91*60;}
28       // Ringpuffer
29       azeit[indx]=zeitstr2();
30       atemp[indx]=temp;
31       aVakku[indx]=Vakku;
32       aLF[indx]=luftfeuchte;
33       indx=indx+1;
34       if (indx==punkte) {indx=0;}
35     }
36   }
37 }

```

## 2. Homepage

```

01 void handleRoot() {
02   String message="<html><head>"
03   //...
04   message += "data.addColumn('number', 'Volt');data.addColumn('number',
05     'Luftf');data.addRow(['";
06   int j= indx-1;
07   if (j<0) {j=punkte-1;}
08
09   for (int i = 1; i <= rows; i++) {
10     message += "[new Date(" +String(azeit[j]) +"),"+ String(atemp[j])
11     +","+ String(aVakku[j]) +","+ String(aLF[j]) + "],";
12     j=j-1;
13     if (j<0) {j=punkte-1;}
14   }
15   message += "]);"
16   //...
17   message += "</body></html>";
18   server.send(200,"text/html", message);
19 }

```

ist in void setup das weitere Vorgehen bei Serveranfragen definiert. Durch server.on("/", handleRoot); wird bei einer Abfrage ohne Pfadangabe void handleRoot() aufgerufen. Wird ein unbekannter Pfad angefragt, wird über void handleNotFound() eine Fehlermeldung an den Browser ausgegeben. Mit server.begin(); wird der Server gestartet. Die eigentlichen Anfragen werden in der

Hauptroutine mit server.handleClient(); abgehandelt.

Nun geht es an die Homepage (siehe Listing 2), genauer gesagt void handleRoot(), das bei einer Browseranfrage im Intranet über die IP-Adresse des ESP32 (z.B. http://192.168.1.200/) aufgerufen wird. Zunächst wird der String message schrittweise aufgebaut und über die Funktion server.

send(200,"text/html", message); an den Browser geschickt. Dies ist zugegebenermaßen nicht sehr elegant, da aber etliche Variablen übernommen werden müssen und Veränderungen leichter durchführbar sind, ist es ein pragmatischer Ansatz. Außerdem lässt der ESP-Arbeitsspeicher derart groß angelegte Stringoperationen zu – auf einem einfachen Arduino Mega wären sie speichertechnisch nicht möglich.

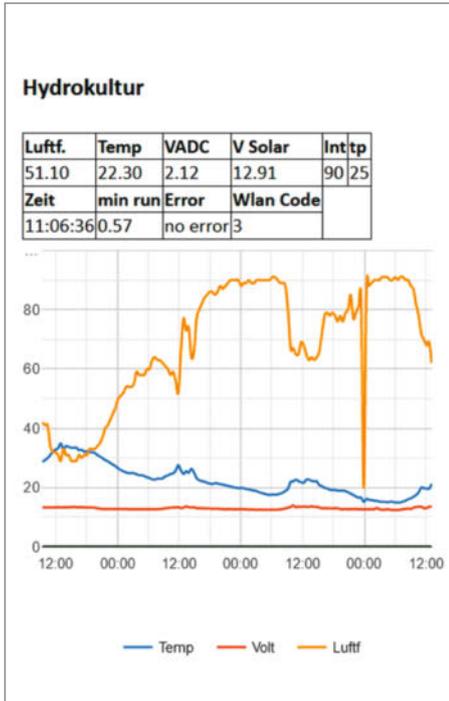
Auf der Homepage zeige ich die aktuellen Sensorwerte derzeit in einer einfachen Tabelle, die aus Standard-HTML aufgebaut ist. Die einzelnen Werte werden über Befehle wie +String(luftfeuchte)+ in den String message geschrieben. Zusätzlich sind die Sensorwerte des beschriebenen Ringpuffers in einem Diagramm zu sehen. Hierzu nutze ich die *Google charts library*. Mit ein paar Zeilen Code entstehen so sehr elegante und interaktive Diagramme.

Der Grafikaufbau beginnt im HTML-Header (siehe Listing 3) mit dem Aufrufen des Loaders (Zeile 3). Danach werden die zu ladenden Pakete (Zeile 5) und eine Callbackfunktion (Zeile 6) festgelegt. Innerhalb der Callbackfunktion wird ein Tabellenobjekt aufgebaut und die einzelnen Spalten angehängt (Zeile 8 bis 12). In Zeile 12 werden alle Tabellenreihen, die unsere Sensorwerte enthalten, auf einmal hinzugefügt. Hierbei wird in einer for-Schleife der Ringpuffer durchlaufen (siehe Listing 2, Zeile 10). Die Variable Punkte bezieht sich auf die Größe des Ringpuffers, die Variable rows auf die bereits gespeicherte Anzahl von Datenpunkten. So werden nur die schon beschriebenen Plätze ausgegeben und leere Zeilen ignoriert.

Zeile 14 legt die Datumsformatierung fest, die in Zeile 15 auf die unterschiedlichen Spalten angewendet wird. Mit options können noch weitere Formatierungen angegeben werden, etwa zur grafischen Ausgabe der Messwertverläufe. Mit den Zeilen 16 und 17 wird die Grafik dann aufgebaut und gezeichnet. Im Body des HTML-Dokuments wird das Diagramm in einem festgelegten <div>-Bereich mit 520 Pixeln Weite und 400 Pixeln Höhe eingebunden.

## Im Einsatz

Für die Ermittlung der Korrekturformel der Akkuspannung kann bereits das Webinterface genutzt werden. Im Programmcode muss außerdem die Gießzeit int tp eingetragen werden, die man am Besten individuell durch Ausprobieren ermittelt: In dieser Zeit soll die komplette Flüssigkeit im Röhrensystem einmal ausgetauscht werden. Sie variiert je nach Volumen, Pumpleistung und der Höhe des Systems. Auch der Sandfilter sorgt für einen gewissen Abflusswiderstand. Wählt man tp zu groß, riskiert man ein Überlaufen des Systems. Bei dem Test kann man außerdem die



Webinterface mit den Messwerten in einer Tabelle und einer Grafik

### 3. HTML-Header

```

01 <html>
02 <head>
03   <script type="text/javascript" src="https://www.gstatic.com/charts/
    loader.js"></script>
04   <script type="text/javascript">
05     google.charts.load('current', {'packages':['corechart']});
06     google.charts.setOnLoadCallback(drawChart);
07     function drawChart() {
08       var data = new google.visualization.DataTable();
09       data.addColumn('date', 'Zeit');
10       data.addColumn('number', 'Temp');
11       data.addColumn('number', 'Volt');
12       data.addColumn('number', 'Luftf');
13       data.addRows([[new Date(2020,11,2,13,24,59),20.50,12.02,56],
14         [new Date(2020,11,2,13,24,31),20.60,11.86,55],]);
15       var date_formatter = new google.visualization.DateFormat({
16         pattern: "dd.MM HH:mm"});
17       date_formatter.format(data, 0);var options = {curveType:
18         'function',legend: { position: 'bottom' },hAxis: {format:
19         'HH:mm'},pointSize: 0, chartArea:{left:20,top:10}};
20       var chart = new google.visualization.LineChart(document.
21         getElementById('curve_chart'));
22       chart.draw(data, options);
23     }
24   </script>
25   <style>table {border-collapse: collapse;} table, td, th {border:
26     1px solid black;text-align: left;}</style>
27 </head>
  
```

ZUKUNFT. DIGITAL. NACHHALTIG.

# DIGITAL TECHNOLOGIES

STUDIERN.

**JETZT BEWERBEN!**



Alle Infos zum Bachelor- und Masterstudiengang:  
[www.digitecstudieren.de](http://www.digitecstudieren.de)

Ein gemeinsames Studienprogramm der



## Projekt

Anlage auf Dichtigkeit prüfen und vor allem den Flüssigkeitsstand in den Röhren im Auge behalten. Sollte bei den Füllstandsbegrenzern Flüssigkeit austreten, kann man sich mit einer Lage Teflonband zwischen Gummiring und Plastikteil behelfen. Etwas warmes Wachs versiegelt undichte Oberflächen 3D-gedruckter Teile.

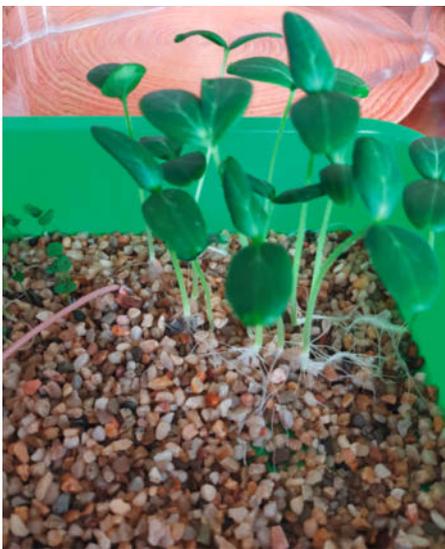
Dann wird die Anlage mit Nährflüssigkeit befüllt, die die Pflanzen vor allem mit Stickstoff, Kalium, Phosphor und Calcium versorgt. Auch Spurenelemente wie Magnesium, Mangan, Kupfer, Zink, Molybdän und Bor müssen vorliegen. Zudem sind Basenpuffer erforderlich, um den pH-Wert der Lösung konstant zu halten. Für den Anfang sollte man eine fertige Flüssig- oder Feststoffmischung im spezialisierten Versandhandel erwerben und nach

Anleitung anmischen. Verbrauchte Menge wird einfach mit neuer Nährflüssigkeit aufgefüllt.

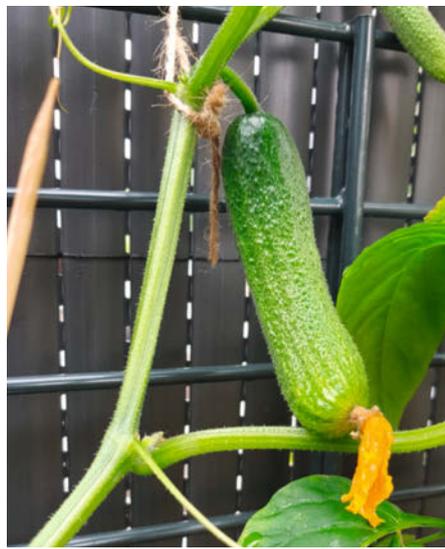
Die Anzucht von Samen im Hydrokulturssystem ist meiner Erfahrung nach nicht erfolgreich, stattdessen sollten sie in einem Zimmergewächshaus angezogen werden. Als Substrat eignet sich feiner Aquarienkies und zum Gießen kann die Nährflüssigkeit verwendet werden. Beachtet werden muss die Lichtvorliebe der Pflanzen beim Keimen. Erst wenn die jungen Pflanzen eine entsprechende Größe erreicht haben und den Umweltbedingungen draußen trotzen können, bringe ich sie vorsichtig in die ausgedruckten Pflanztröge in Kies ein. Dabei werden einige Wurzeln durch die Öffnungen im Boden geführt, so dass sie die Nährflüssigkeit erreichen. Anschließend

wird der Trog vollständig mit Kies befüllt und in die Hydrokultur gestellt. Alternativ können fertige Setzlinge aus der Gärtnerei verwendet werden, was sich besonders bei Salat lohnt. Damit keine zusätzlichen Substrate in die Nährflüssigkeit kommen, muss die Erde vollständig abgewaschen werden. Das gelingt, wenn man die Pflanzen einige Zeit in Wasser einweicht.

Gerade am Anfang bedarf das System noch einiger Aufmerksamkeit: Je nach Pflanze müssen eventuell geeignete Stützstrukturen aufgebaut und der Sonnenschutz optimiert werden. Im Feldversuch habe ich gute Erfahrungen mit Salat, Chillipflanzen, Snackgurken, Zucchini und Kapuzinerkresse gesammelt. Wurzelgemüse ist natürlich ungeeignet, ansonsten steht der Kreativität nichts im Wege. —hch



Anzucht der Keimlinge im Zimmergewächshaus



Die Früchte des Erfolgs: Snackgurken



Zucchini



Chillis

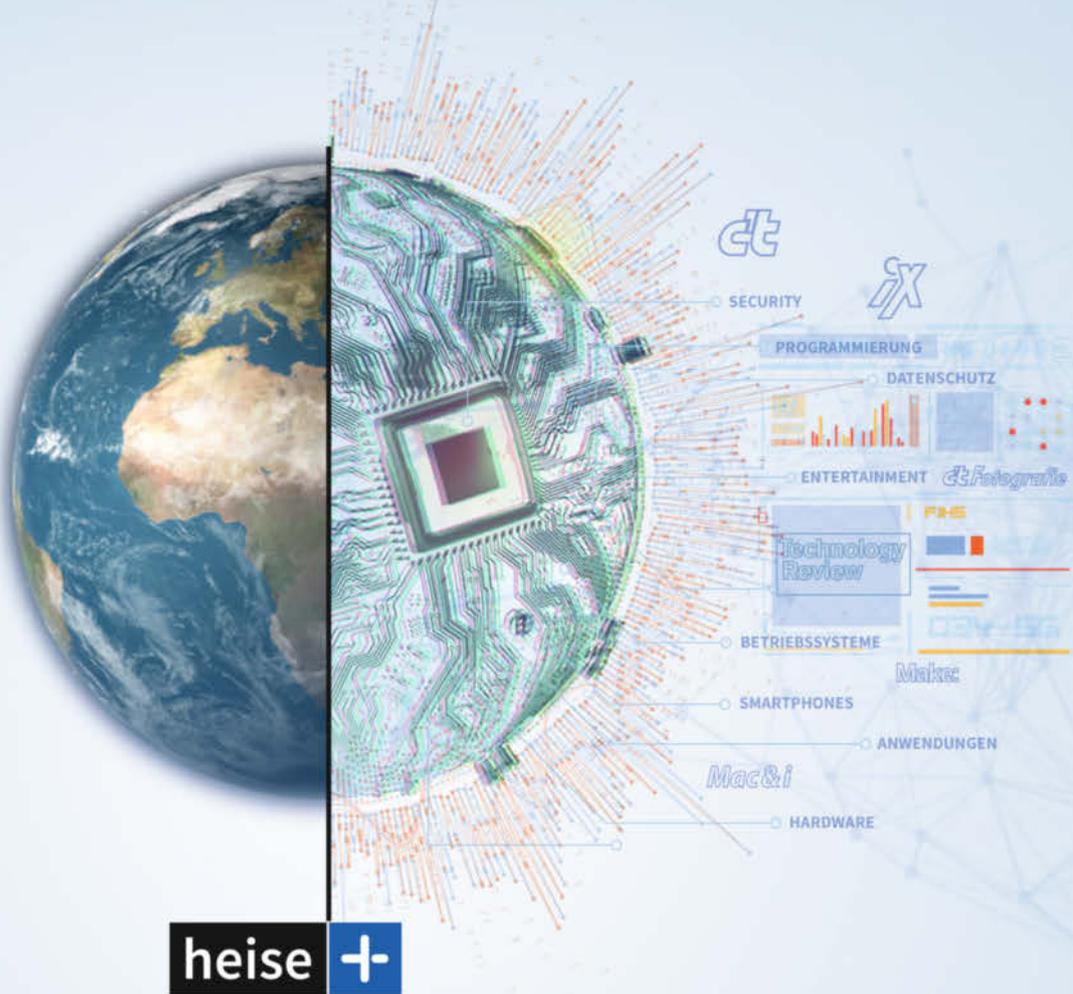


Tomaten mit Basilikum



Kapuzinerkresse

Make-  
Abonnenten  
lesen bis zu  
**60%**  
günstiger



heise +

## Das digitale Abo für IT und Technik.

**Exklusives Angebot für Make-Abonnenten:** Lesen Sie zusätzlich zum Make-Magazin unsere Magazine bequem online auf [heise.de/magazine](http://heise.de/magazine) und erhalten Sie Zugang zu allen heise+ Artikeln.

- ✓ Für Make-Abonnenten 5 €/Monat
- ✓ Jeden Freitag Leseempfehlungen der Chefredaktion im Newsletter-Format
- ✓ 1. Monat gratis lesen – danach jederzeit kündbar
- ✓ c't, iX, Technology Review, Mac & i, Make, c't Fotografie direkt im Browser lesen

**Sie möchten dieses Exklusiv-Angebot nutzen?  
Unser Leserservice hilft Ihnen gern beim Einrichten.**

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

☎ 0541 80009 120



Weitere Informationen zum  
Abo-Upgrade finden Sie unter:

[heise.de/plus-info](http://heise.de/plus-info)

Ein Angebot von: Heise Medien GmbH & Co. KG • Karl-Wiechert-Allee 10 • 30625 Hannover

# Modellbahn-Komponenten mit dem ESP32

Eine digitale Modellbahn besteht neben Loks und Gleisen auch aus digitalen Komponenten, mit denen man die Züge steuert. Dazu kann man in den einschlägigen Geschäften alle notwendigen Teile kaufen. Oder man baut sie selbst, was mit meinem Projekt CANguru einfach ist.

von Gustav Wostrack



**D**amit meine Loks zwischen kleinen Häusern, Seen und Tunneln durch Berglandschaften immer wissen, wo sie gerade hinfahren sollen, habe ich mit dem von mir entwickelten CANguru-System eigene Steuerkomponenten geschaffen. So bringe ich meine beiden Hobbies – das Tüfteln mit dem ESP32 und Modellbahnen – unter einen Hut. Der Name entstand, da das System auf dem Märklin-CAN-Bus aufsetzt. Es besteht aus der zentralen CANguru-Bridge, Decodern und dem CANguru-Server. Die CANguru-Bridge basiert auf einem ESP-Modul mit integrierter CAN-Schnittstelle und die Decoder auf einer von mir entwickelten Platine, die jeweils mit dem günstigen ESP32-WROOM läuft. Diese Platinen können für unterschiedliche Zwecke wie Gleisbesetzmelder oder Weichensteuerung eingesetzt werden. Je nach Einsatzzweck werden sie mit anderen Bauteilen bestückt und programmiert. Neben den ESP-Sketches habe ich mit dem CANguru-Server noch eine Erweiterung für PC-Steuerprogramme geschrieben, um die CANguru-Komponenten zu verwalten.

Alle Entwicklungsergebnisse (Platinenlayout, die Software sowie weitere Hinweise wie 3D-Druckdateien) stelle ich auf [github.com/CANguru-System](https://github.com/CANguru-System) zum Herunterladen zur Verfügung. Auch Weiterentwicklungen lege ich dort stetig ab. Zwei Anlagen, eine kleine und eine mittelgroße, sind bei mir seit einiger Zeit mit den CANgurus zuverlässig in Betrieb. Und das Beste daran ist: Wenn mir irgendetwas nicht gefällt, kann ich es einfach ändern.

### Kurzinfo

- » CANguru-System zur Steuerung von Modellbahnen
- » Kommunikation über ESP-NOW
- » Beispielbauteil Weichendecoder

### Mehr zum Thema

- » Jörg Pleumann, Arduino steuert Märklin-Modellbahn, c't Hacks 1/13, S. 54
- » Martin Berke, Wunderland en miniature, Make 1/17, S. 30
- » Florian Schäffer, Servos für jeden Zweck, Make 3/18, S. 46

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



» Gustav Wostrack, Digitale Modellbahn selbstgebaut, 2019, dpunkt.verlag

### Die CANguru-Bridge

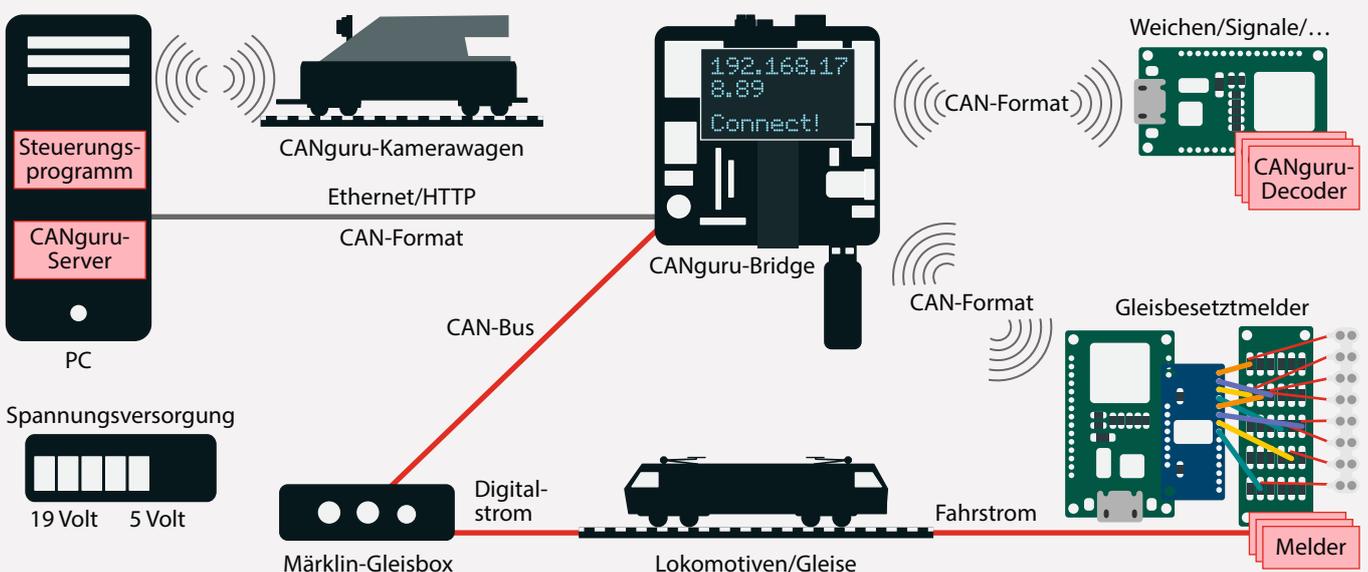
Der Dreh- und Angelpunkt des Systems ist die CANguru-Bridge. Sie stellt die Verbindung zwischen der Steuersoftware der Modellbahn auf einem PC und den Bahnkomponenten her. Alle Befehle, die der Nutzer über das Steuerprogramm an die Loks, Weichen oder Signale gibt, werden im Märklin-CAN-Format über ein Ethernetkabel an die CANguru-Bridge und dann auf

den physikalischen CAN-Bus gelegt und an die Gleisbox geleitet. Sie erzeugt das entsprechende Signal und führt es über das Gleis an die Lokomotiven. Mit den Decodern kommuniziert die Bridge drahtlos über ESP-NOW, einem speziellen Funkprotokoll von Espressif, das nicht auf dem WLAN-Standard beruht.

Für die Bridge nutze ich das ESP32-Modul Olimex ESP32EVb mit CAN-Schnittstelle. Zusammen mit der Steuerprogrammerweite-

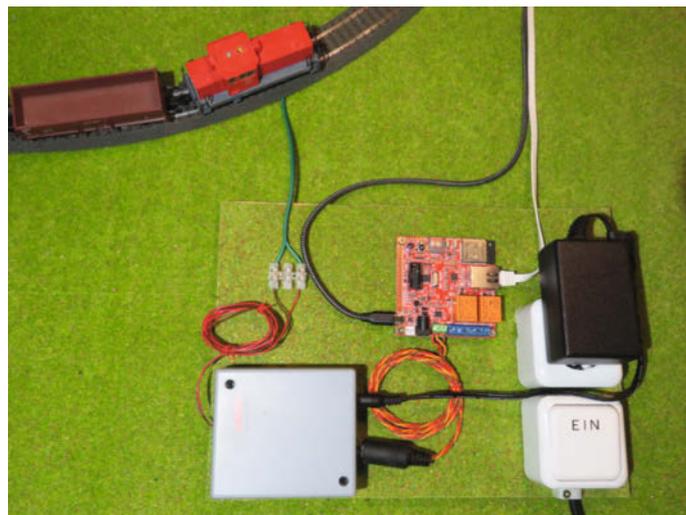
### Die Kommunikationsverbindungen im CANguru-System

Die CANguru-Bridge steuert die unterschiedlichen CANguru-Decoder und wird bei deren Verwaltung von dem CANguru-Server unterstützt.

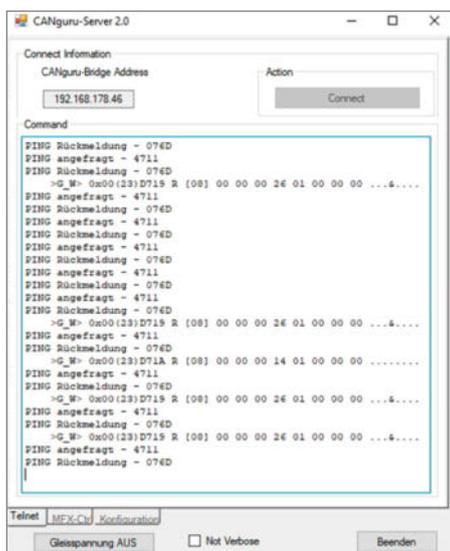




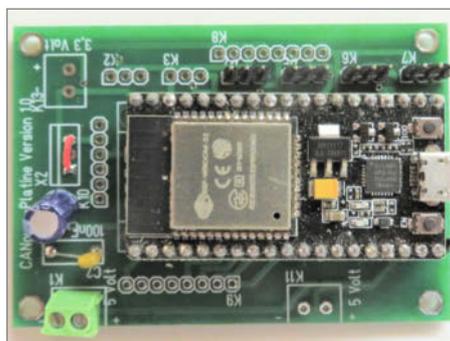
Die CANguru-Bridge verteilt alle Kommandos an die Decoder – hier mit großem LC-Display aufgebaut. Alternativ ginge aber auch ein kleines Display.



Die Minimalausstattung besteht neben den PC-Komponenten nur aus der Gleisbox und der CANguru-Bridge mit den Netzteilen. Damit können bereits Züge gesteuert werden.



Der CANguru-Server ist das Interface der Decoder, mit dem deren Parameter eingestellt werden. Auch die Kommandos vom Steuerungsprogramm werden angezeigt.



Um nur wenige Bauelemente ergänzt, wird aus dem ESP32-WROOM ein Weichendecoder, an den vier Servos angeschlossen werden können.

Die CANguru-Server reicht dies bereits aus, um Loks zu steuern. Die Bridge kann außerdem Informationen, wo sich die Züge befinden, aus dem Bahnsystem aufnehmen und an das Steuerungsprogramm weitergeben.

### CANguru-Server

Quasi die Mensch-Maschine-Schnittstelle der CANguru-Bridge ist der CANguru-Server, der die Steuersoftware ergänzt. Das Programm ist in C# geschrieben und läuft derzeit nur auf Windows. Ich nutze als Steuerprogramm das kostenpflichtige WinDigiPet, wofür ich den CANguru-Server ursprünglich entwickelt habe. Inzwischen funktioniert es aber auch mit anderen Lösungen, wie etwa dem kostenlosen Railcontrol. Mit dem CANguru-Server „verwalte“ ich meine ESP-Module, denn alle Decoder benötigen bei ihrem ersten Einsatz Angaben, wie sie einzusetzen sind. Das ist beispielsweise eine Adresse oder mit welcher Geschwindigkeit die Weichen umgeschaltet werden sollen. Über den CANguru-Server kann ich die Module inzwischen auch drahtlos (Over The Air, OTA) neu programmieren.

Mit Klick auf den Button *Connect* wird zunächst eine Verbindung zwischen dem Server und der Bridge und dann zwischen der Bridge und den Decodern hergestellt. Ist dies erfolgt, kann das Steuerungsprogramm gestartet werden. Nun werden alle Befehle wie Lok-Geschwindigkeit, Weichenstellen etc. vom Steuerungsprogramm direkt an die Bridge und von dort an die zuständigen Decoder oder an die Gleisbox gesendet. Der CANguru-Server spielt jetzt nur noch eine dokumentierende Rolle, in dem die auszuführenden Befehle angezeigt werden.

### ESP-NOW als Alternative zum WLAN

ESP-NOW ist ein drahtloses Kommunikationsprotokoll von Espressif, mit dem mehrere ESP32-Module Daten ohne ein WLAN austauschen können. Um es einzusetzen, müssen die beteiligten Geräte zunächst einen Pairing-Prozess abschließen. Anschließend besteht eine sichere Verbindung und es sind keine weiteren Schritte wie etwa Handshaking nötig. ESP-NOW funktioniert als Primary/Secondary-System, mit der CANguru-Bridge als Primary-Controller und den CANguru-Decodern als Secondaries.

Der Pairing-Prozess läuft in etwa folgendermaßen ab: Alle Decoder senden für bestimmte Zeit ihre eindeutige MAC-Adresse. Die CANguru-Bridge empfängt die Adressen und sendet jeweils eine Quittung. Leider ist die Anzahl der Netzteilnehmer auf 20 begrenzt. Mit 19 Decodern kann man aber noch eine mittelgroße Modellbahn ausstatten. ESP-NOW bietet auch die Möglichkeit, die Kommunikation zu verschlüsseln – dabei wird aber die Anzahl der Module halbiert, die mit der Zentrale kommunizieren können. Da ich die Wahrscheinlichkeit als gering ansehe, dass es Hacker gibt, die meine Modellbahn unbedingt entgleisen lassen wollen, hab ich mich für die 20 Module entschieden.

### Die Decoder

Alle Decoder habe ich über die CANguru-Platine mit dem ESP32-WROOM als zentralem Baustein aufgebaut. Je nach Einsatzgebiet wird sie entsprechend um wenige Bauteile, wie etwa Spannungsregler, Kondensatoren oder den PCA9685-Controller zur Ansteuerung

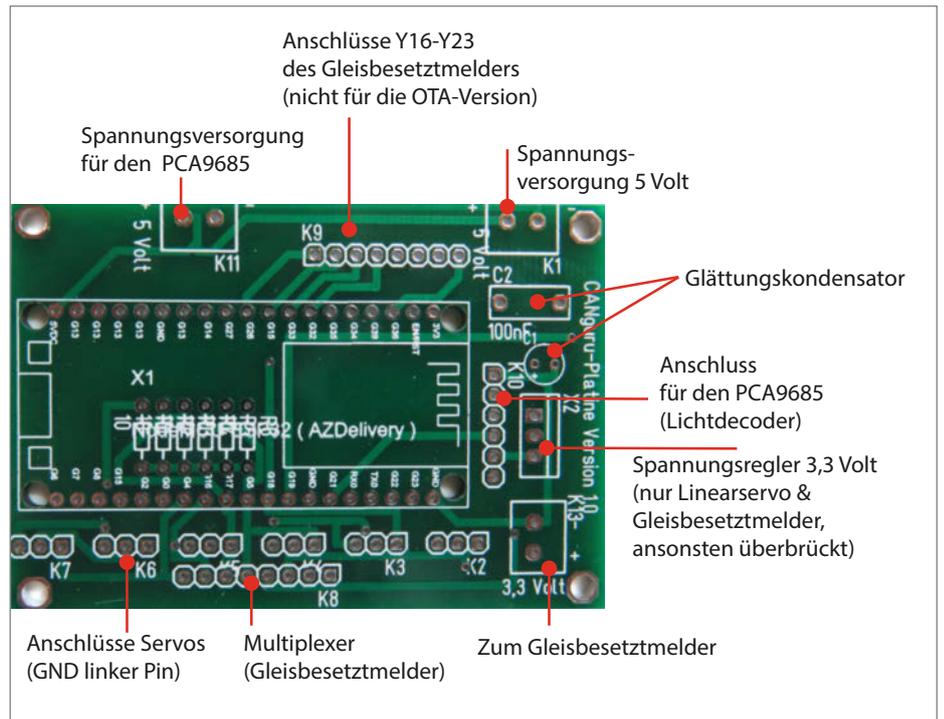
rung von LEDs, ergänzt. Für die Spannungsversorgung gibt es einen 5-Volt-Anschluss. Dies ist die einzige eingehende Leitung, so dass beispielsweise ein Servo-Decoder zusätzlich nur Kabel für die einzelnen Servos braucht. Die notwendigen Infos kommen drahtlos über ESP-NOW.

Was die Software anbelangt, so haben die unterschiedlichen Decoder auch unterschiedliche Software, allerdings mit einem gemeinsamen Kern, einer Art Betriebssystem, der sie befähigt, mit den anderen Komponenten zu kommunizieren und die benötigten Daten bereitzustellen. Die Sketches habe ich in der Programmiersprache C++ in der kostenlosen Programmierumgebung Visual Studio Code mit PlatformIO erstellt. Um das Rad nicht jedes Mal neu erfinden zu müssen, nutze ich, wenn möglich, Bibliotheken aus der Arduino-Welt.

### Weichendecoder

Als Beispiel soll hier eine Weiche dienen, denn das Stellen der Weichenzungen ist eine der wichtigsten Aufgaben bei einer Modellbahn. An den Decoder können dafür bis zu vier Servos angeschlossen werden.

Deren Ansteuerung ist für einen ESP32 keine Herausforderung, da er lediglich die Steuerleitungen der Servos bedienen muss. Deshalb sieht der Schaltplan sehr übersichtlich aus. Die beiden Kondensatoren am Eingang sorgen für stabile Verhältnisse bei der Versorgungsspannung und die Widerstände an den Steuerleitungen gewährleisten dort steile Flanken.



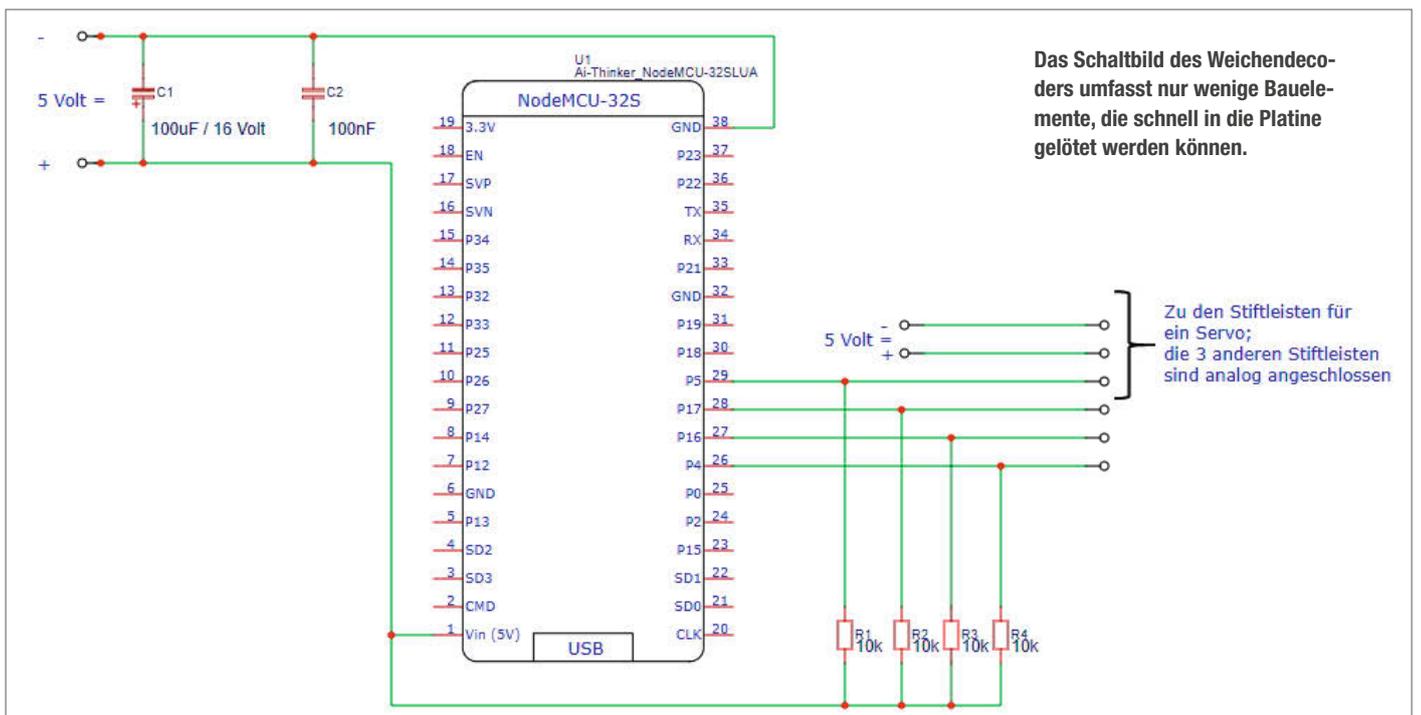
Auf dieser Standardplatine werden alle Decoder mit wenigen Komponenten aufgebaut.

### Servos und Halterung

Zwei Servo-Varianten sind im Modellbahnbau gebräuchlich. Einmal gibt es den Servo, der auch häufig im Flugmodellbau verwendet wird. Er bewegt einen kleinen Arm, der sich auf einer Achse um einen zentralen Punkt dreht und damit die Weichenzungen bewegt. Daneben werden Linear-Servos angeboten,

die eine geradlinige Bewegung ausführen. Sie sind in kleinen und flachen Ausführungen erhältlich, was für den Einbau unter Weichen besonders praktisch ist. So muss man keine großen Löcher in die Weichenaufgabe bohren.

Die Halterung für den Linear-Servo habe ich selbst entworfen und 3D-gedruckt, da es sie so nicht zu kaufen gibt. Als Entwurfssoftware habe ich TinkerCAD von Autodesk ge-



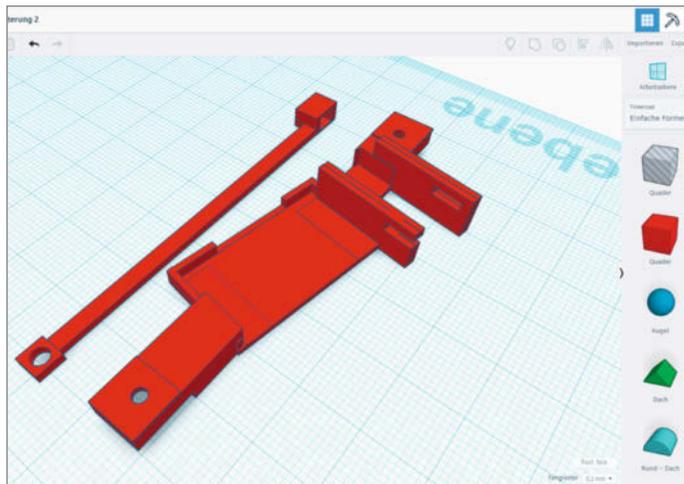
Das Schaltbild des Weichendecoders umfasst nur wenige Bauelemente, die schnell in die Platine gelötet werden können.

### Listing 1

```

01 if (pos != (destpos + endpos))
02 {
03   if ((micros() - lastUpdate) > updateInterval)
04   {
05     pos += increment;
06     ledcWrite(channel, pos);
07     lastUpdate = micros();
08   }
09 }
10 else
11 {
12   if (way == longway)
13   {
14     increment *= -1;
15     endpos = 0;
16     way = noway;
17   }
18 }

```



Die Halterung für den Linear-Servo habe ich mit TinkerCAD entworfen.

nutzt. Dank des schlichten Klötzchenkonzepts gelingt ein Design auch Ungeübten recht schnell. Die Software ist leicht bedienbar, im Browser sowie als App auf dem Tablet verfügbar und zudem kostenlos.

### Software

Die Software sorgt für die richtige Position des Servoarmes sowie die passende Geschwindig-

keit. Darüber hinaus muss sie damit umgehen, dass eventuell mehrere Servos ihre Position gleichzeitig verändern. Daher ist unter anderem zu klären, welche Weiche betroffen ist, welcher Servo also aktiv werden soll.

Damit die Weiche zuverlässig schaltet, wird die Zunge jeweils ein wenig weiter als die eigentliche Zielposition endpos bewegt, zu destpos. Erst danach fährt sie zur endgültigen Stellung zurück, was mit einer if-Abfra-

ge überprüft wird (siehe Listing 1). Die Geschwindigkeit des Servoarms wird mit Hilfe eines internen Timers umgesetzt, der seine Zeit mit der Funktion micros() bereitstellt. Nur wenn nach der letzten Zeitnahme (lastUpdate = micros();) die Zeit updateInterval verstrichen ist, wird die Position des Servoarmes überprüft und bei Bedarf korrigiert. Das tatsächliche Fahren des Armes wird mit der Arduino-Funktion ledcWrite(channel, pos); ausgelöst. Die Richtung der Bewegung bestimmt das Vorzeichen der Variablen increment. Damit wird genau ein Servo bedient. Um alle vier Servos anzusteuern gibt es noch eine übergeordnete loop-Schleife.

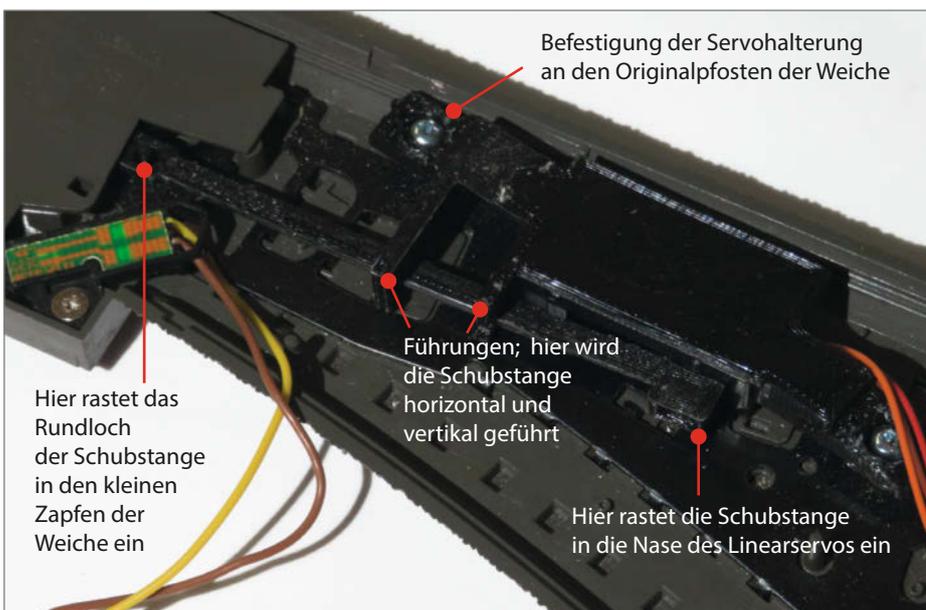
Die zyklisch aufgerufene Prozedur Servos[servo].Update(); in Listing 2 arbeitet die eben besprochenen Zeilen ab und stellt sicher, dass alle Servos immer die korrekte Position aufweisen. Abgesehen von unterschiedlichen Parametern für die angesteuerten Winkel und die Verstellgeschwindigkeit ist die Software prinzipiell für beide Servotypen nutzbar.

### Listing 2

```

01 for (uint8_t servo = 0; servo < num_servos; servo++)
02 {
03   Servos[servo].Update();
04 }

```



Das Bild zeigt eine umgebaute Weiche von unten. Der Linear-Servo passt in die Halterung aus dem 3D-Drucker. Mit der kleinen Schubstange wird auf der Oberseite die Weichenzunge umgestellt.

### Weiterbasteln

Basierend auf diesem System habe ich inzwischen auch komplexere Projekte wie einen Kamerawagen umgesetzt. Darauf ist ein ESP32-Modul gemeinsam mit einer Kamera montiert. Der Wagen wird von einer Lok durch die Modellanlage geschoben und sendet seine Bilder per WLAN an den PC, wo sie direkt angesehen oder für später abgespeichert werden können.

Für alle Interessierten, die dem Modellbahn hobby erlegen sind, erkläre ich das System ausführlich in einem Buch, das im dpunkt.verlag unter dem Titel *Digitale Modellbahn selbstgebaut: CANguru-Steuerung mit ESP32 in Arduino-Umgebung* erschienen ist. —hch

# So spannend kann Wissen sein!

Das Magazin, das Wissen schafft.



**TESTEN SIE WISSEN  
MIT 30% RABATT!**

**2 Ausgaben für nur 11,20 €\*  
statt 15,80 €\* im Handel**

**Hier anfordern:**

**[www.emedia.de/wissen-mini](http://www.emedia.de/wissen-mini)**

\*Preis in Deutschland.



(0541) 80009 126  
(werktags von 8 – 20 Uhr, samstags von 10 – 16 Uhr)



[wissen-abo@emedia.de](mailto:wissen-abo@emedia.de)



Leserservice eMedia Wissen,  
Postfach 24 69, 49014 Osnabrück



© Copyright by Maker Media GmbH.  
[emedia.de](http://emedia.de)

 eMedia GmbH

# Der Photostick – Malen mit Licht

Moderne RGB-LEDs ermöglichen brillante Lichteffekte für Langzeitbelichtungen bei Dunkelheit. Mit etwas Nachhilfe durch einen Arduino verwandelt dieses Projekt einen herkömmlichen LED-Streifen in ein virtuelles Display für mannshohe Computerbilder und Grafiken.

von Manuel Mohr und Hermann Mohr



Unter Fotografen ist *Light Painting* schon seit längerem eine beliebte Technik. Bewegt man während einer Langzeitaufnahme eine Lichtquelle vor der Kamera, lassen sich faszinierende Muster in real existierende Hintergründe „malen“. Die so entstehenden Bilder sind auch in Zeiten von Photoshop & Co. noch echte Hingucker. Das hier beschriebene *Photostick*-Projekt hebt das Light Painting auf die nächste Stufe, indem es erlaubt, nicht nur abstrakte Muster, sondern komplexe Bilder in die Luft zu malen. Eine ähnliche Konstruktion wie unseren Photostick kann man auch als fertiges Produkt kaufen, als *Pixelstick* für 350 US-Dollar.

Die Grundidee des Photosticks ist einfach: Ein Bild wird Spalte für Spalte auf einem senkrechten LED-Streifen angezeigt, wobei jede LED genau ein Pixel in der gerade angezeigten Bildspalte bildet. Die Höhe des Bildes in Pixeln muss also genau der Anzahl der verfügbaren LEDs entsprechen. Bewegt man nun den Photostick in passender Geschwindigkeit an einer Kamera vorbei und macht währenddessen damit eine Langzeitbelichtung, kombiniert die Kamera die einzeln angezeigten Spalten wieder zum ursprünglichen Bild. Das Bild scheint auf dem entstandenen Foto in der Luft zu schweben.

Damit lassen sich verschiedenste Effekte erzielen – wolltet Ihr beispielsweise nicht schon immer mal Euer Treppenhaus in Brand setzen? Mit dem Photostick benötigt Ihr dafür lediglich ein feuriges Foto sowie ein bisschen Zeit.

Diese Kurzbeschreibung gibt uns schon die Minimal-Zutaten vor, um unser Photostick-Projekt umsetzen zu können:

- Einen RGB-LED-Streifen, möglichst dicht besetzt mit LEDs, um eine gute Auflösung zu erreichen.
- Einen Akku zur Stromversorgung, damit der Photostick auch mobil ist.
- Ein Speichermedium, um vorbereitete Bilder mit passender Größe auf dem Photostick hinterlegen zu können.
- Einen Mikrocontroller, um die Bilder laden und den LED-Streifen geeignet ansteuern zu können.
- Und schließlich ein Gestell mit Griff, um die Komponenten montieren zu können und die ganze Apparatur bedienbar zu machen.

## LEDs und Akku

Bei den LED-Streifen fiel unsere Wahl auf WS2812B-Streifen, die günstig und mit einer relativen hohen Dichte von 144 LEDs pro Meter verfügbar sind. Zwei solcher Streifen ergeben mit 288 LEDs eine gute Auflösung für jede Bildspalte (der kommerzielle *Pixelstick* bietet lediglich 200 LEDs). Mit einer Höhe von zwei Metern sollte unser Eigenbau also gute Bilderergebnisse liefern und trotzdem einigermaßen handhabbar bleiben.

## Kurzinfo

- » LED-Streifen als Display verwenden
- » Komplexe Bilder in Langzeitbelichtungen malen
- » Akkubetrieb trotz hohem Energiebedarf

### Checkliste



**Zeitaufwand:**  
ein Tag



**Kosten:**  
150 Euro



**Löten:**  
einfache Lötarbeiten



**Programmieren:**  
mit der Arduino-IDE

### Hilfsmittel

- » LiPo-Ladegerät
- » microSD-Karte
- » Panzertape
- » Tennisschläger-Griffband
- » Kabelbinder oder Klettbander

### Mehr zum Thema

- » Ulrich Schmerold, Lichtwischer-Kugel, Make 3/16, S. 12
- » Christiane Rütten, Arduino-Lichtwischer, c't Hacks 1/12, S. 56



Für die Auswahl unseres Energielieferanten werfen wir einen kurzen Blick in das Datenblatt unserer LED-Streifen. Mit den dort angegebenen 0,3 Watt pro LED bei 288 LEDs ergibt sich eine relativ hohe (theoretische) maximale Leistungsaufnahme von 86,4 Watt. Auf Grund der guten Verfügbarkeit wählen wir daher einen 2S-Lithium-Polymer-Modellbauakku (LiPo). Dieser liefert 7,4V, woraus ein genügend dimensionierter DC/DC-Wandler die benötigten 5V für die LEDs macht.

Bezüglich der benötigten Akku-Kapazität und der Entladegeschwindigkeit gibt uns eine kurze Überschlagsrechnung Aufschluss. Bei 7,4V Ausgangsspannung und (aufgerundeten) 100W Last muss unser Akku 13,5A liefern. Um unseren Photostick bei Volllast zehn Minuten lang zu versorgen, benötigen wir also eine Kapazität von  $13,5A \cdot 1/6h = 2250mAh$ . Wenn wir annehmen, dass eine durchschnittliche

### Material

- » **Arduino Mega** (oder kompatibles Board); Arduino Uno nur ohne Touchscreen-GUI
- » **Adafruit 2.8" TFT Touch Shield** mit microSD-Kartenslot
- » **2 Meter LED-Streifen** mit hoher Pixeldichte, z.B. WS2812B-Streifen von BFT-Lighting mit 144 LEDs pro Meter
- » **DC/DC-Wandler** ausgelegt für mindestens 100 Watt (Bezugsquelle siehe Link)
- » **1000µF-Kondensator** 6,3 V oder mehr
- » **330Ω- oder 470Ω-Widerstand**
- » **10kΩ-Widerstand**
- » **15kΩ-Widerstand**
- » **Schrumpfschlauch**
- » **LiPo-Akku** z.B. 20C-2S-LiPo-Akku mit mindestens 2000mAh
- » **XT60-Stecker** bzw. anderer, passend zum Akku
- » **Kippschalter** ohne Beleuchtung
- » **Kabel** ca. 2m, ausreichend für 20A
- » **Stück Kabelkanal 40mm x 60mm** Länge 30 bis 40cm
- » **Stück Holzlatte** Länge 10cm; Profil 30mm x 50mm
- » **2 Aluprofilschienen** 2 Meter lang, 25mm breit, 10mm hoch; idealerweise mit milchiger Plastikabdeckung
- » **Alublech** ca. 20cm x 10cm
- » **Alu-Vierkantrohr** ca. 30cm lang, Durchmesser so wählen, dass es in die Profilschiene passt
- » **M4-Senkkopfschrauben** passend zu Bohrungen in Aluprofilschienen
- » **M4-Muttern** passend zu Schrauben
- » **Montagematerial** Winkel, Muttern, Schrauben, je nach eigener Konstruktion

Langzeitbelichtung 10 Sekunden dauert, können wir also 60 Bilder mit einer Akkuladung machen.

In der Realität können wir aber deutlich mehr Bilder erwarten, da die wenigsten Bilder bei voller LED-Helligkeit gemacht werden und während einer Aufnahme auch selten alle 288 LEDs permanent mit voller Helligkeit aktiv sind. Daher sind einfache Akkumodelle, etwa ein 2000mAh-20C-2S-LiPo-Akku, der für unter 20 Euro erhältlich ist, für unser Projekt völlig ausreichend. Für solch einen Akku stellt auch unsere benötigte Entladegeschwindigkeit kein Problem dar.

## Controller und Display

Beim Mikrocontroller entschieden wir uns für einen *Arduino Mega*, nicht zuletzt wegen der Unmenge an verfügbaren Hardware-Erweite-



rungen und Software-Bibliotheken. Der geradezu üppige Arbeitsspeicher gegenüber einem *Arduino Uno* (8KiB statt 2KiB) macht es uns später einfacher, die Software verständlich zu halten und ein paar Komfortfunktionen einzubauen. Arbeitsspeicher ist nämlich durchaus eine knappe Ressource in unserem Projekt: Bei 288 LEDs belegt die Farbinformation einer einzigen Bildspalte im üblichen Format mit 3 Bytes pro Pixel (jeweils 1 Byte für die Farbkanäle Rot, Grün und Blau) bereits 864 Byte RAM – die 2KiB RAM eines *Arduino Uno* schmelzen schnell dahin. Praktischerweise läuft ein *Arduino* mit Versorgungsspannungen zwischen 6,5V und 12V, weshalb wir den *Arduino* einfach direkt an den Ausgang unseres Akkus hängen können.

Für die Datenablage wollen wir natürlich eine *microSD*-Karte verwenden, was ein bequemes Kopieren neuer Bilder auf den Photostick ermöglicht. Statt eines der vielen *Arduino*-Shields, das nur einen *SD*-Karten-Slot bietet, haben wir uns für ein kombiniertes *LCD*/*SD*-Shield mit Touchscreen entschieden. Der Touchscreen ermöglicht es uns, eine sehr flexible und bequem zu verwendende Oberfläche zu implementieren, ohne weitere Taster oder Knöpfe verbauen zu müssen.

### Gestell

Unsere Komponenten montieren wir auf ein Gestell, das wir aus zwei *Alu*-Profilschienen mit

passendem Innendurchmesser basteln. Den *LED*-Streifen verlegen wir im Inneren der Schiene und schließen diese mit einem matten, aber lichtdurchlässigen Kunststoffprofil ab, was für eine gleichmäßigere Lichtverteilung sorgt. Auf Bild 1 sieht man die Größenverhältnisse (*LED*-Streifen senkrecht).

### Der Bau – Schritt für Schritt

Zunächst kümmern wir uns um die Verbindung unserer *LED*-Streifen. Jeder *WS2812B*-*LED*-Streifen besitzt drei Eingangspins (*GND*/*+5V* für die Versorgungsspannung und *DIN* als Dateneingang) sowie drei Ausgangspins (*GND*/*+5V* sowie *DOUT* als Datenausgang). Um unsere zwei je 1m langen Streifen miteinander zu verbinden, verlöten wir also die Ausgänge des ersten Streifens direkt passend mit den Eingängen des zweiten Streifens. Die direkte Verbindung ist nötig, um eine Lücke beim Übergang zwischen den beiden Streifen zu vermeiden, die im Bild später als störender dunkler Streifen erkennbar wäre 2.

Am *Arduino* verwenden wir *Digital*-Pin 2, um das Datensignal an die *LED*-Streifen zu senden. Für einen ersten Test können wir ein Jumperkabel verwenden, später verlöten wir das Kabel direkt auf der Unterseite des *Arduino*. Um den Dateneingang zu schützen, setzen wir einen kleinen Widerstand (330Ω oder 470Ω) in die Datenleitung direkt vor *DIN* des *LED*-Streifens 3.

### Die Spannungsversorgung

Wer ein passendes regelbares Netzteil zur Verfügung hat, kann fürs Testen natürlich auch zunächst dieses zur Spannungsversorgung verwenden. Ansonsten verlöten wir als erstes unseren *XT60*-Stecker mit zwei kurzen Kabelstücken. Den Pluspol verbinden wir mit einem Kontakt unseres Schalters, den anderen Schalterkontakt verbinden wir mit dem Eingang *IN+* des *DC*/*DC*-Wandlers. Den Minuspol des *XT60*-Steckers verbinden wir mit dem Eingang *IN-* des Wandlers. Die Kontakte des Wandlers mit Schraubklemmen sollten es leicht machen, von *IN+* und *IN-* zusätzlich die unregelmäßig 7,4V des Akkus an die Pins *VIN* und *GND* des *Arduino* abzuzweigen.

Nun müssen wir den *DC*/*DC*-Wandler noch auf die Ausgabe von 5V einstellen. Wir schließen also ein Multimeter an die Ausgänge des Wand-

lers an, verbinden unsere Spannungsquelle und justieren dann die Stellschrauben auf dem Wandler so, dass am Ausgang 5V anliegen. Die Ausgänge des Wandlers verbinden wir nun mit den Pins *GND* und *+5V* unseres ersten *LED*-Streifens. Um Spannungsspitzen abzufedern, setzen wir zwischen die Ausgänge des Wandlers noch einen 1000µF-Kondensator.

### Erster Test

An dieser Stelle können wir unser Werk nun das erste Mal testen. Wir installieren in der *Arduino*-IDE die *FastLED*-Bibliothek (Download und Installationsanleitung siehe Link in der Kurzinfor), wählen das Beispielprogramm *FirstLight* aus und konfigurieren folgende Einstellungen:

```
#define NUM_LEDS 288
#define DATA_PIN 2
```

In der *setup()*-Funktion noch folgende Zeile einfügen:

```
FastLED.addLeds<WS2812B, DATA_PIN, GRB>(leds, NUM_LEDS);
```

Achtung: Die nötige Farbreihenfolge ist *GRB*! Sobald wir das Programm auf den *Arduino* laden und die *LED*-Streifen mit Spannung versorgen, sollten wir ein weißes Lauflicht bewundern können.

### Das Display

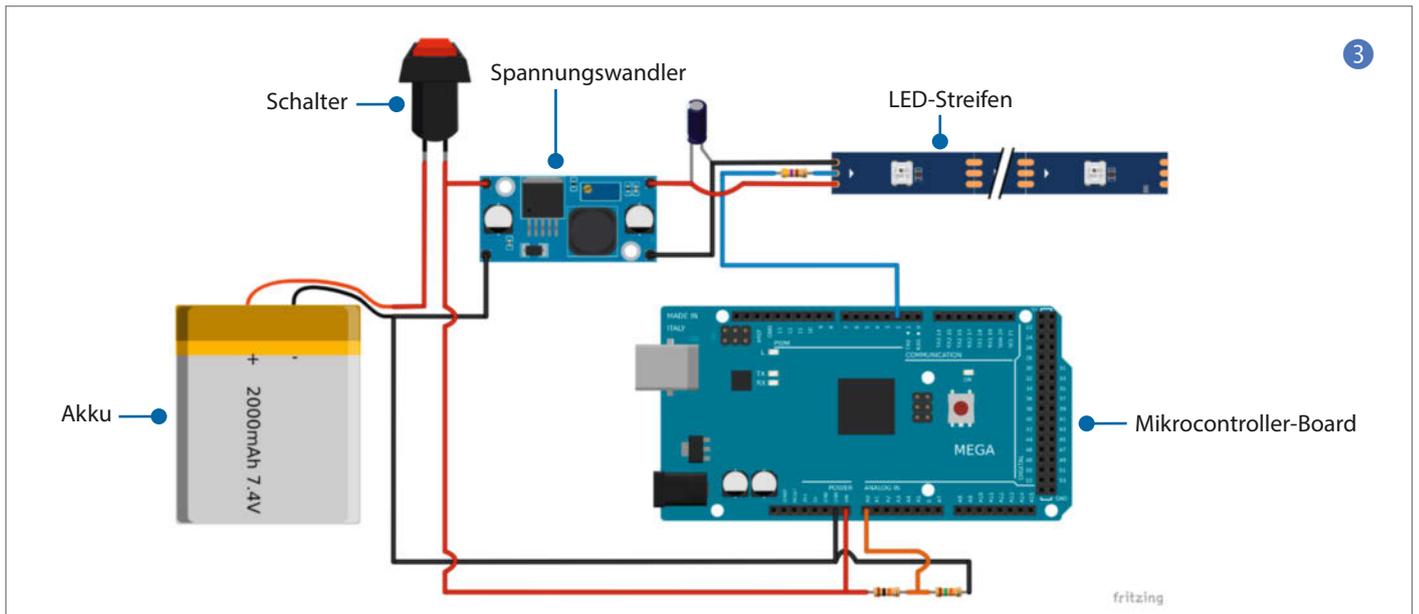
Das *Adafruit*-Display ist eigentlich schon komplett gebrauchsfertig und muss nur noch auf den *Arduino* aufgesteckt werden. Damit wir später während der Langzeitbelichtung das Display ausschalten können, müssen wir allerdings noch wie in der Dokumentation beschrieben (siehe Kurzinfor-Link) die Lötbrücke 3 mit der Beschriftung *backlite* schließen. Danach kann das Display auf den *Arduino* gesteckt werden und ist einsatzbereit.

Nach der Installation der Bibliotheken *Adafruit\_GFX\_Library*, *Adafruit\_ILI9341*, *Adafruit\_STMPE610*, *Adafruit\_TouchScreen* über den Bibliotheksmanager kann man den Touchscreen mit den mitgelieferten Beispielprogrammen testen.

### Das Gehäuse

Das Stück Kabelkanal dient uns als Gehäuse für *Arduino*, *DC*/*DC*-Wandler, den Ein-/Aus-





schalter, sowie die zugehörige Verkabelung. Den Kabelkanaldeckel kürzen wir auf 13cm. Anschließend schrauben oder kleben wir den Arduino auf das Stück Holzlatte, sodass das LCD bündig mit dem oberen Ende des Kabelkanals abschließt. Es kann dabei nötig sein, ein bisschen Holz abzunehmen, sodass die Kabel genug Platz finden **4**.

Nun das Holzstück inklusive Arduino so im Kabelkanal befestigen, dass der SD-Karten-Slot nach außen zeigt **5** **6**. Ein kleines Loch in der Seite für den Schalter **7** sowie die Befestigung des DC/DC-Wandlers auf der anderen Seite des Kabelkanals komplettieren das Gehäuse **8**.

### Das Gestell

Für das Gestell schrauben wir zunächst mit den Senkkopfschrauben und Muttern die bei-

den Aluprofilschienen Rücken an Rücken aneinander. In die eine Aluschiene verlegen wir vorsichtig den LED-Streifen und schließen die Schiene mit der Plastikabdeckung ab. Die Kabel (Spannungsversorgung und Daten) des LED-Streifens führen wir in der zweiten Aluschiene zurück zur Mitte der Schiene **9**.

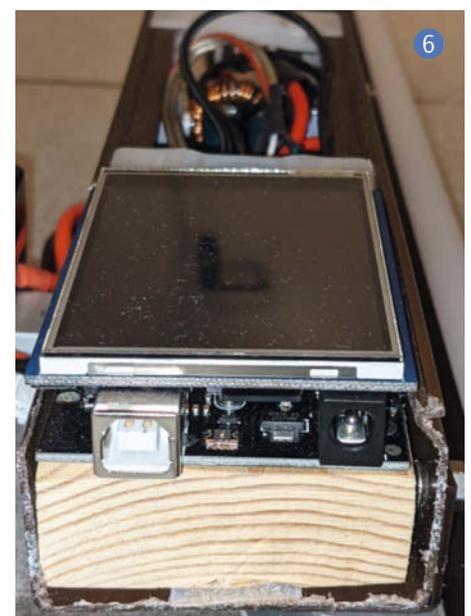
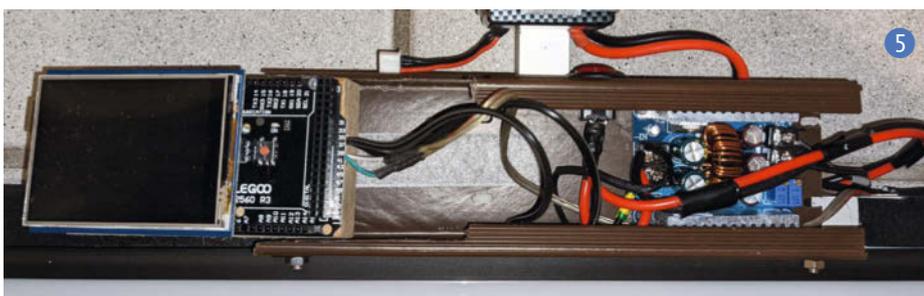
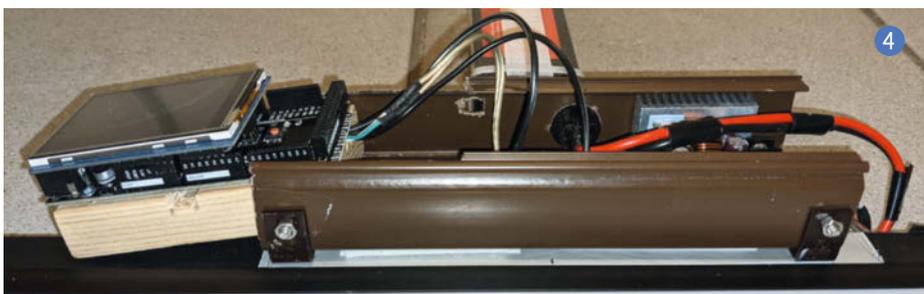
Für den Griff stecken wir das Vierkantrohr in die LED-abgewandte Profilschiene, sodass es im 90-Grad-Winkel absteht. Mit dem Alublech (bei Bedarf geeignetes zusätzlich) verbinden wir Vierkantrohr und Profilschiene **10**.

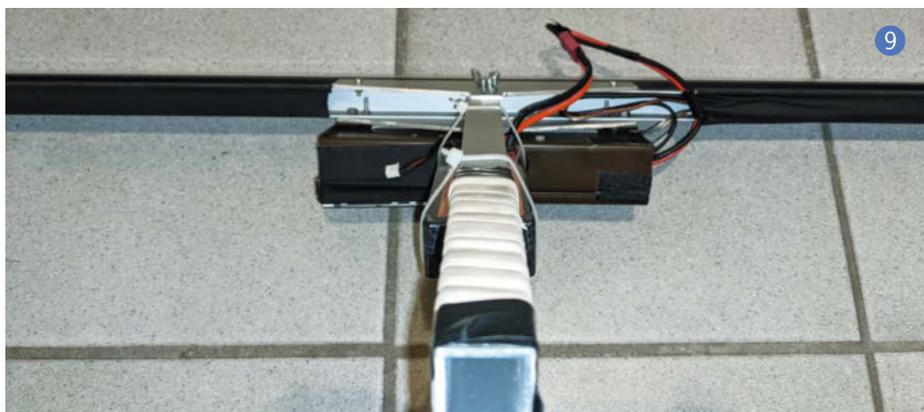
Das Gehäuse und den Akku befestigen wir ebenfalls auf diesem Vierkantrohr, was einen guten Schwerpunkt der Gesamtkonstruktion garantiert. Für etwas bequemere Handhabung umwickeln wir schließlich das Vierkantrohr mit dem Tennisschläger-Griffband **11**.

### Die Software

Die Aufgabe unserer Arduino-Software ist konzeptuell einfach: In jedem Zeitschritt laden wir eine Spalte des anzuzeigenden Bildes von der SD-Karte und geben die Farbinformationen der Pixel auf dem LED-Streifen aus.

Zum Laden von Daten von der SD-Karte ist eine Vielzahl von Bibliotheken verfügbar. Wir verwenden hier die *SdFat*-Bibliothek, da sie in unseren Tests die höchste Ladegeschwindigkeit lieferte. Da wir permanent Daten nachladen, ist die Ladegeschwindigkeit wichtig, da sie direkt über die Bildwiederholfrequenz entscheidet. Die Bibliothek kann wie gewohnt über die Bibliotheksverwaltung der Arduino-IDE installiert werden.





Unser kombiniertes LCD-/SD-Shield legt den SD-Chip-Select auf Pin 4 des Arduino. Mehr brauchen wir auch gar nicht zu wissen, um die bei SdFat mitgelieferten Beispielprogramme ausprobieren zu können.

Als nächstes müssen wir uns Gedanken dazu machen, welches Dateiformat wir für unsere Bilder verwenden möchten. Komplizierte Formate wie JPEG oder PNG fallen aufgrund der aufwändigen Dekomprimierung weg. Da es aber ein weit verbreitetes Format sein sollte, bietet sich das BMP-Format an.

BMP weist eine sehr einfache Dateistruktur auf, bei der die Farbwerte der einzelnen Pixel unkomprimiert abgelegt sind, was uns die Verarbeitung erleichtert. Ein kleines Problem gilt es noch zu umschiffen: Im BMP-Format werden die Pixeldaten zeilenweise abgelegt. Da wir spaltenweise auslesen, würde somit im schlimmsten Fall für jeden Pixel ein neuer Sektor (512 Byte) von der SD-Karte geladen, was wir natürlich unbedingt vermeiden müssen.

Die Lösung ist aber einfach: Wir speichern das Bild einfach um 90° rotiert ab. Somit entspricht nun jede Zeile im rotierten Bild genau einer Spalte im Originalbild und wir können sehr effizient unsere Farbdaten einlesen.

Anschließend müssen wir nur noch die eingelesenen Farbdaten an unsere LED-Streifen weitergeben. Hierzu dient die *FastLED*-Bibliothek. Wir initialisieren diese mit Pin 2 als Daten-Pin und setzen die Anzahl der LEDs auf 288.

Nun müssen wir im `loop()`-Teil unseres Programms nur noch alle Teile zusammensetzen. Wir laden eine Zeile aus dem rotierten Bild, wandeln die Pixeldaten in ein Array aus *FastLED-CRGB*-Objekten um, übergeben dieses an *FastLED* und rufen schließlich `FastLED.show()` auf. Fertig ist die Kernschleife 12 unserer Photostick-Software!

Der gezeigte Beispielcode ist eine absolute Minimalvariante zur Verdeutlichung des Prinzips, bei der sämtliche Prüfungen fehlen, die der Code zum Download bei Github natürlich enthält. Das Programm lädt das Bild *Test.bmp* von der SD-Karte und zeigt dieses Bild auf dem LED-Streifen an. Das Bild muss um 90° rotiert (sodass es 288 Pixel breit ist) und mit 24 Bit Farbtiefe abgelegt worden sein.

## Die Oberfläche

Natürlich ist die Software jetzt noch sehr umständlich zu benutzen, da es keinerlei Interaktionsmöglichkeiten gibt. Doch genau hierfür haben wir das Touchscreen-LCD verbaut. Für die Erstellung einer ressourcenschonenden grafischen Oberfläche eignet sich die *GUISlice*-Bibliothek. Hiermit lässt sich die funktionale Kernschleife bedienbar machen.

Wir stellen die erweiterte Photostick-Software auf Github zur Verfügung (Download über den Link in der Kurzinfo). Unsere Software erlaubt über den Touchscreen gesteuert



**WIR MACHEN  
KEINE WERBUNG.  
WIR MACHEN EUCH  
EIN ANGEBOT.**

**ct**

[ct.de/angebot](https://ct.de/angebot)

Jetzt gleich bestellen:

 [ct.de/angebot](https://ct.de/angebot)

 +49 541/80 009 120

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

**ICH KAUF MIR DIE c't NICHT. ICH ABONNIER SIE.**

Ich möchte c't 3 Monate lang mit 35 % Neukunden-Rabatt testen.  
Ich lese 6 Ausgaben als Heft oder digital in der App, als PDF oder direkt im Browser.

**Als Willkommensgeschenk erhalte ich eine Prämie nach Wahl,  
z. B. einen RC-Quadrocopter.**

© Copyright by Maker Media GmbH.



NEU  
im heise shop

# Unterwegs in Deutschlands schönsten Ecken!



## c't Fotografie FOTOTOUREN

Für fantastische Fotoausflüge muss man nicht in die Ferne schweifen. c't Fotografie führt Sie auf Exkursionen durch kleine und große Städte sowie malerische, heimische Landschaften quer durch die drei Republiken.

[shop.heise.de/fototouren21](http://shop.heise.de/fototouren21)

12,90 € >

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

## Projekt



13 die Auswahl des zu ladenden Bildes 14 sowie die Konfiguration wichtiger Parameter wie Helligkeit und Geschwindigkeit 15 und bietet auch ein Kreativ-Menü für abstrakte Lichtmuster 16 17. Nach Installation der benötigten Bibliotheken lässt sich die Software einfach über die Arduino-IDE laden.

Der Photostick kann in einigen Punkten natürlich noch verbessert werden. So wäre eine Steckmechanik praktisch, mit der sich der zwei Meter lange Stab in zwei Hälften zerlegen ließe. Dabei ist eine sehr kompakte Verbindung zwischen den beiden LED-Streifen wichtig, da sonst störende dunkle Streifen in den Bildern entstehen könnten. —pek

## LiPo-Wächter

Wer möchte, kann bei der Spannungsversorgung eine kleine Erweiterung vornehmen: Mit einem 10kΩ-Widerstand und einem 15kΩ-Widerstand können wir einen Spannungsteiler realisieren, der es erlaubt, den Arduino gleichzeitig noch als einfachen LiPo-Wächter zu verwenden, damit wir in der Hitze des Fotogefechts keinen Akkuschaden riskieren.

Unter Annahme von 4,2V für eine vollgeladene LiPo-Zelle und etwa 3,4V für eine leere Zelle liefert unser 2S-Akku also zwischen 8,4V (voll) und 6,8V (leer). Der Spannungsteiler gibt uns davon  $(15k\Omega / (10k\Omega + 15k\Omega)) = 60\%$ , was also mit etwa 5V (voll) und etwa 4V (leer) Spannungen genau im erlaubten Bereich der Arduino-Analogeingänge ergibt. Die Software kann somit bei Spannungen  $\leq 4V$  am Analogeingang A0 den Nutzer warnen.

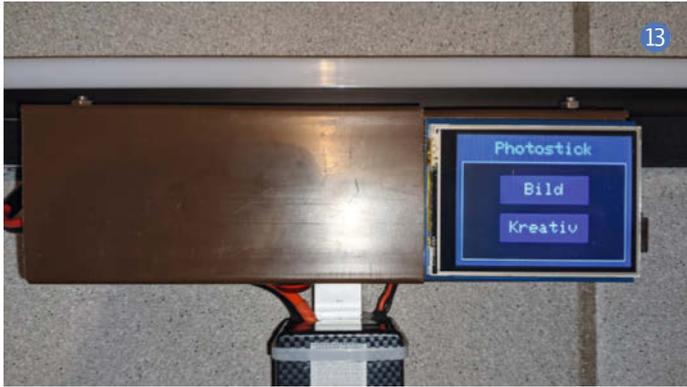
## 12 Kernschleife Photostick (gekürzt)

```
01 #include "FastLED.h"
02 #include "SdFat.h"
03
04 #define DATA_PIN 2
05 #define NUM_LEDS 288
06 #define SD_CS 4
07 #define IMAGE_FILENAME "Test.bmp"
08
09 CRGB leds[NUM_LEDS];
10 SdFat sd;
11 SdFile file;
12 uint32_t imageOffset; uint32_t imageHeight;
13
14 void setup() {
15     FastLED.addLeds<WS2812B, DATA_PIN, GRB>(&leds[0], NUM_LEDS);
16
17     sd.begin(SD_CS, SD_SCK_MHZ(50));
18     file.open(IMAGE_FILENAME, O_RDONLY);
19     file.seekSet(10);
20     file.read(&imageOffset, sizeof(imageOffset));
21     file.seekSet(22);
22     file.read(&imageHeight, sizeof(imageHeight));
23 }
24
25 int32_t row = 0;
26
27 void loop() {
28     if (row >= imageHeight) return;
29
30     static uint8_t buf[NUM_LEDS * 3];
31
32     file.seekSet(imageOffset + (imageHeight - row - 1) * sizeof(buf));
33     file.read(&buf[0], sizeof(buf));
34
35     for (int i = 0; i < NUM_LEDS; ++i) {
36         leds[i] = CRGB(buf[3 * i + 2], buf[3 * i + 1], buf[3 * i]);
37     }
38
39     ++row;
40
41     FastLED.show();
42     delay(10);
43 }
```

 heise shop

[shop.heise.de/fototouren21](http://shop.heise.de/fototouren21)





23. - 25.  
FEBRUAR 2021

Interessante redaktionelle Keynotes  
und Partner-Vorträge

Virtuelle Fachausstellung

Interaktive Breakout-Sessions

Diverse Networking-Optionen

Viele kostenlose, digitale und  
reale Gadgets

sec-it.heise.de

Heise Medien

Unsere Partner



# Der digitale Treffpunkt für Security-Experten

## KEYNOTE:

Avant-Garde InfoSec // Dr. Melanie Rieback

## INTERAKTIVE VORTRÄGE:

Einstiegsmöglichkeiten in die Cybersicherheit für KMUs //

Viktor Rechel

Schatzsuche im Notfallmanagement – die neue Voranalyse  
im BSI-Standard 200-4 // Lukas Reike-Kunze

Anleitung zum Emotet-Selbsttest // Martin Junghans

Cyber-Versicherung: Risikogruppe IT-Betriebe? // Tobias Wenhart

„Behind the Hype: Buzzwords kritisch hinterfragt“ // Stefan Strobel

„Experten-FAQ“ mit Sidekick // Olaf Pursche und Tobias Schrödel

## PODIUMSDISKUSSION:

zum Thema Hackback // Andreas Könen, Manuel Atug, Tobias Haar

## HEISE SHOW SPEZIAL:

Computer-Forensik – Fakten und Fiktion //

Martin Wundram und Krimiautor Constantin Gillies

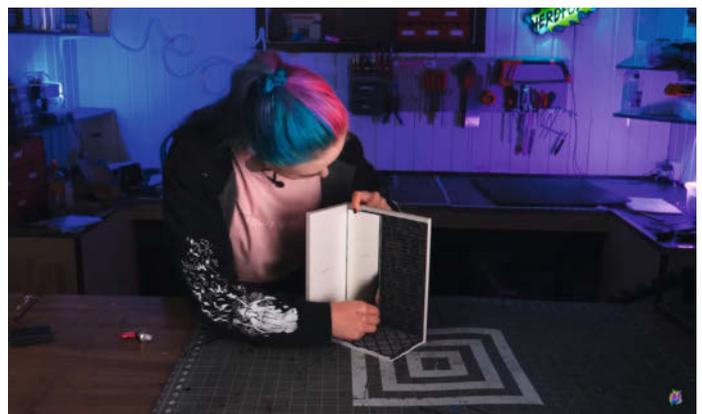
Preis: 249,00 € (inkl. MwSt.)

© Gorodenkoff





# Was uns inspiriert



Bilder: Nerdforge

# Die unendliche Bibliothek

Diese verspielte Diorama-Buchstütze ist eine sogenannte *Book Nook*. Wie man eine einfache Book Nook baut, haben wir in unserem Sonderheft „Loslegen mit Holz“ gezeigt, doch die norwegischen YouTuber Martina und Hansi von *Nerdforge* haben mit ihrer Buchstütze die Grenzen der Maker-Kreativität gesprengt. Martinas Bibliotheks-Diorama ist mit einem unserer Lieblings-Make-Projekte ausgestattet: dem Infinity Mirror. Man blickt direkt auf einen halbdurchlässigen Spiegel, der die Buchreihen und Fackeln scheinbar unendlich reflektiert.

Ihrer winzigen Bibliothek hat Martina außerdem mit einem Trick aus der Theaterkiste Leben eingehaucht: Der *Pepper's Ghost*-Effekt ist eine Bühnenillusion, die Ende des 19. Jahrhunderts für das Theater entwickelt wurde. Mithilfe eines Spiegels und einer Lichtquelle erweckt man den Eindruck, dass ein Geist auf der Bühne erschienen ist. Martina verwendet diese Technik, um die Fackeln in der Bibliothek flackern zu lassen. Dazu setzt sie ihr Handy mit einer Flammenanimation in eine Aussparung an der linken Wand – sehen kann man am Ende nur die perfekt positionierte Spiegelung im unendlichen Gang.

—rehu

► [thenerdforge.com](https://thenerdforge.com)

Bilder: Nerdforge





## Spaß am Retro-Gerät

Maker Oriol Ferrer Mesjà hat die trübe Zeit der Pandemie genutzt, um sich in den Umgang mit parametrischer 3D-Software einzuarbeiten und die Grenzen des 3D-Drucks zu erkunden. Entstanden sind dabei eine Reihe von experimentellen, aber trotzdem funktionstüchtigen Retro-Computer-Terminals. Manche sind von alten Telefonzellen inspiriert, andere möglichst minimal gehalten – Retro-Nostalgie erzeugen sie alle.

Seine 3D-Modelle hat Oriol mit *Autodesk Fusion360* entwickelt und mit PLA auf einem *Creativity Ender 3* gedruckt. Die meisten Terminals laufen mit einem Raspberry Pi 4 und einer Raspberry-Pi-Tastatur. Das kupferne, ultraweite 8,8-Zoll-LCD-Terminal hat ein ziemlich exotisches Seitenverhältnis von 4:1. Da dieses Format die Größe des Druckbetts überschreitet, war die größte Herausforderung am Ende für ihn, das Modell „unsichtbar“ in zwei Teile zu teilen. Die Naht kann man nur noch erahnen.

—rehu

► [uri.cat/projects/modern-retro-terminal/](https://uri.cat/projects/modern-retro-terminal/)



Bilder: Oriol Ferrer Mesjà





## Festliche Leuchtkunst

Als ST und Robert von *Project-Insanity* (eine Gruppe von Makern aus Stuttgart und Umgebung) auf einer kleinen Party waren, nutzten sie die Gelegenheit, STs neu gebauten und mit LEDs beleuchteten Quadcopter zu testen. Während des Flugs hat Robert mit seiner Sony ILCE-6300 einige fantastische Langzeitbelichtungsbilder gemacht. Die malerischen Leuchteffekte entstanden bei Sonnenuntergang und in der frühen Nacht. Robert hat die Fotos in Flugrichtung der Drohne geschossen.

Um die LED-Streifen an den Armen mit dem Copter zu verbinden, hat ST je vier Drähte verdreht, verlötet und an den Flugregler angeschlossen. Dank der Flugsteuerungs-Software *Betaflight* war es für ihn einfach, spezielle Funktionen wie die Ansteuerung von WS2812-LEDs zu konfigurieren. Im Betaflight-Konfigurator kann

man jede einzelne LED-Farbe und -Funktion individuell ändern und sogar an den „Heckleuchten“ Bremsen und Blinker simulieren. Wer braucht da noch Feuerwerk?

—rehu

► [blog.project-insanity.org](http://blog.project-insanity.org)

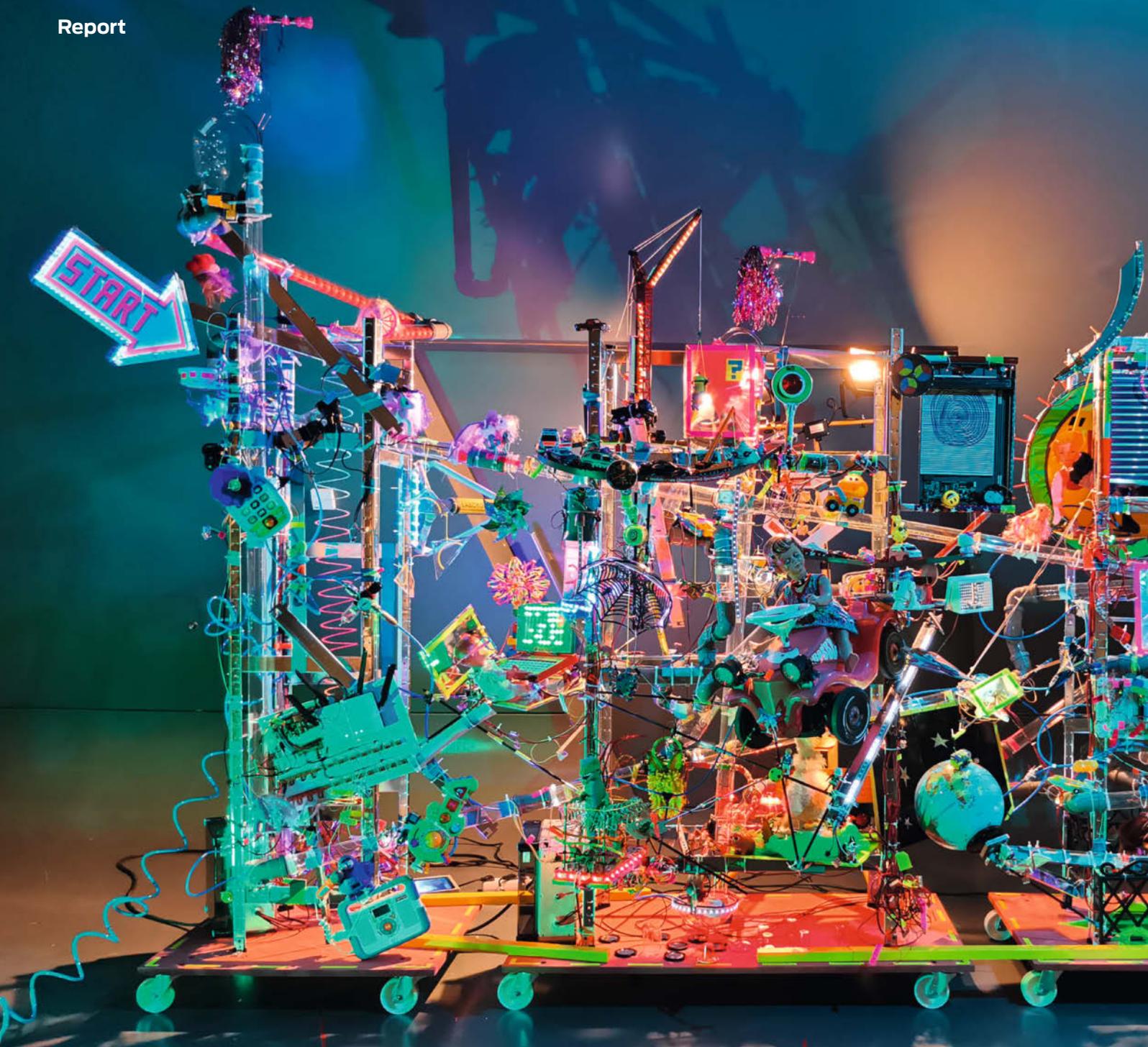
### Was inspiriert Dich?

Wir freuen uns über Vorschläge an:

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



Bilder: Robert/Project-Insanity



Bilder: Niklas Roy und Felix Figus

# So entsteht eine interaktive Riesenmaschine

Ein halbes Jahr lang unterhielt die *Smart Fairy Tale* alle, die am Schaufenster des Wolfsburger Science Center *Phaeno* vorbei liefen. Wir haben die beiden Erbauer gefragt, worauf man beim Konstruieren eines so komplexen Projekts achten muss.

von Helga Hansen

Alles zum Artikel  
im Web unter  
[make-magazin.de/xq9p](https://make-magazin.de/xq9p)



**B**linkende Lichter, winkende Barbie-Arme und piepsende Plastikgitarren – wer in den letzten Monaten in Wolfsburg aus dem Zug stieg, wurde vom angrenzenden Phaeno mit einer glitzernden Maschine im Schaufenster empfangen. Ein ebenso bunt beleuchtetes Plakat lud ein, über eine Webseite die Maschinenkontrolle zu übernehmen. Auf Knopfdruck schoß dann eine Kugel durch das Labyrinth transparenter Plastikrohre und setzte allerlei Animationen in Gang, die dem Spielzeug Bewegungen und Töne entlocken. Wo genau es lang ging, war ebenfalls über die Webseite zu steuern.

So faszinierend wie das grelle Äußere ist auch der Blick ins Innenleben: Insgesamt wer-

kelten in der Maschine neben einem Raspberry Pi 4 als Server 25 Arduino Nanos, die fast ebenso viele animatronische Module und 720 LEDs steuerten. Drei Druckluftanschlüsse bewegten die Kugeln durch den Rohrkreislauf, der rund 30 Meter lang ist. Schließlich sorgten 600 Kabelbinder und 140 Meter Tape für Sicherheit und Stabilität, damit weder Arduinos noch Barbies verloren gehen. Hinter der interaktiven Installation stecken die zwei Bastler und Künstler Nikolas Roy und Felix Figus, die uns coronakonform über einen Videochat ver-raten haben, wie man in zwei Wochen eine elektronische Riesenmaschine baut, die viele kleine Geschichten erzählt

## Start im Sommer

Die Idee zum Projekt kam vom Phaeno selbst, das als Science Center viel Erfahrung mit naturwissenschaftlich-technischen Ausstellungen und Demonstrationsobjekten zum An-fassen hat. Für die jährliche *Phaenomenale*, ein Kulturfestival rund um Kunst und Digitales, sollte eine neue interaktive Installation her. Durch eine vorherige Kooperation kannte das Phaeno-Team bereits den Berliner Roy, der seinen Freund Figus mit an Bord holte. Die beiden sollten dabei nicht alleine im Kämmerchen basteln – mit einer Spendenaktion für gebrauchtes Spielzeug und die Animatronik-Workshops wurden insbesondere die Schülerinnen und Schüler der Umgebung gleich mit eingebunden.

Relativ schnell verwerfen mussten Roy und Figus allerdings die erste Idee, eine Rube-Goldberg-Maschine zu bauen, bei der eine Aufgabe dank zahlreicher Unterschritte so kompliziert wie möglich ausgeführt wird. Denn alles, was dabei umgestoßen oder heruntergeworfen wird, müsste über einen zweiten Mechanismus wieder aufgestellt werden. Statt doppelten Entwicklungsaufwand und mehr Fehlerquellen zu riskieren, entschied sich das Team bald für eine Murmelbahn mit Lichtschranken und Weichen. Läuft eine Murmel an einer Schranke vorbei, startet eines der Animatronikmodule und bewegt sich. Anschließend wird alles einfach wieder ruhig. Je nachdem, auf welchem Weg eine Kugel durch die Maschine rollt, ergibt sich so eine andere Geschichte.

Nach einigen Basteleien mit Lichterschranken und Prototypen für die Smartphone-Steuerung ging es dann im Spätsommer zum ersten Mal ins Phaeno, um in einer Woche das Grundgerüst aus Metallstreben und Rohren zu bauen. Klar waren den beiden nun schon die weiteren Anforderungen: Der Aufbau sollte möglichst modular und wieder auseinandernehmbar erfolgen, um durch Türen zu passen oder später sogar im Auto transportiert werden zu können. Für die Künstler eher ungewohnt waren dazu die strikten Sicherheits-

vorschriften des Phaeno, das die Spannungsversorgung durch seine Elektriker gleich in Schaltkästen verstauen ließ. Die Maschinenteile wurden nur noch mit ungefährlichen 12, 5 und 3 Volt betrieben.

Äußerst praktisch war wiederum, dass das Phaeno für seine Ausstellungen überall Pressluft-Anschlüsse bereit stellt und mit einem 3D-Drucker jederzeit neue Anschlussstücke oder andere Teile entstehen konnten. Einige Komponenten dürften Fans von Roy bekannt vorkommen, der die transparenten Rohre von einem vorigen Projekt wiederverwendet hat. Aus einer groben Idee, wo die Bahn anfangen und enden sollte, entstand so langsam das Grundgerüst: drei Türme, durch die die Kugeln fallen, bis sie jeweils von einem Pressluft-Aufzug wieder nach oben vor eine Abzweigung befördert werden. „Eine Murmelbahn mit drei Weichen.“ – „Das klingt so unspektakulär.“ heißt es von den beiden etwas ernüchtert. Spektakulär viele Features bedeuteten aber oft Frust, stellen sie schnell klar, denn dann sei man nur noch mit dem Verstehen der Spielmechanik beschäftigt, statt einfach zu spielen.

Anschließend blieb noch ein Monat Zeit bis zur Workshop-Woche, was Roy und Figus Zeit zum Shoppen gab. Da die Stadt Wolfsburg die Phaenomenale finanziell unterstützt, stand sogar ein ordentliches Budget zur Verfügung – auch das eine eher seltene Erfahrung für die Künstler. Arduinos, Magnetventile und natürlich Kabelbinder standen auf der Einkaufsliste, wobei letztere später nachgekauft werden mussten. Bei nur einer Woche Zeit für den Bau der Animationen war klar, dass alles möglichst auf Anhieb funktionieren und befestigt werden müsste, gemäß dem Motto: „Schnell bauen, aber stabil.“ Besonders bewährt hat sich dabei die Methode, auf glatten Oberflächen wie den Rohren erst mit doppellagigem Klebeband und obendrauf Kabelbindern zu arbeiten. Alternativ helfen schnell gebohrte Löcher und Kabelbinder über Kreuz gelegt, so ziemlich alles schnell festzuzurren.

## Workshop

Durch die Pandemie gestalteten sich die Workshops eher ungewohnt. Obwohl das Ideenforum im Phaeno viel Platz bietet und die Tische mit Abstand aufgestellt wurden, konnten doch nur zehn Jugendliche dabei sein. Desinfektionsmittel und Masken sorgten für weitere Sicherheit. Das einzige „Super-spreader-Event“, so Roy, sei die gute Laune gewesen. Das Phaeno-Team startete kurz vorher noch eine Aktion, bei der sich Grundschulkinder als Inspiration Szenen mit dem gespendeten Spielzeug überlegen und zeichnen sollten. Umgesetzt wurden die Ideen nicht unbedingt, aber auf einem Whiteboard neben der Maschine präsentiert. Das Board rotierte während der Ausstellung fleißig vor sich hin



Phaeno-Mitarbeiterin Mareike Wiese sortiert das gesammelte Spielzeug.



In der Maschine stecken insgesamt drei solcher Module aus Metallstreben und Rohren.

und zeigte abwechselnd die Kinderzeichnungen und Schaltbilder der Maschine.

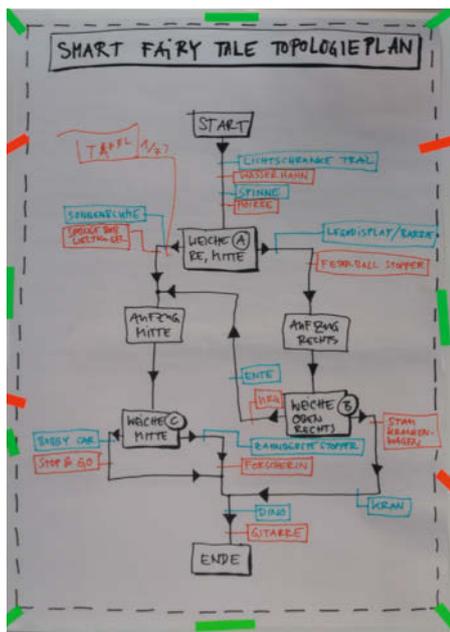
Nun ging es an die letzte Herausforderung: aus einem großen Haufen alter Puppen, Dinofiguren und einem Bobbycar interaktive Szenen mit Arduinosteuerung zu bauen und zu programmieren und auf der Murmelbahn zu einem Gesamtkunstwerk zusammenzustellen. Die beteiligten Jugendlichen kamen aus Technik-AGs und brachten daher bereits einige Erfahrung mit, aber nicht unbedingt direkt mit Mikrocontrollern. Daher hieß es für die beiden Bastler zunächst, die Ideen auf ihre realistische Umsetzbarkeit zu prüfen. Innerhalb von einer Woche sollten schließlich alle Module fertig sein. So wurde der Bastelprozess noch einmal gestrafft. Während die ersten Experimente noch gelötet wurden, mussten später Breadboards und viel Heißkleber reichen.

Auf der Programmseite steckt in den vielen Arduinos ganz einfach Beispielcode. Das Meiste „hat Lady Ada für uns programmiert“ verweist Roy auf Limor Fried, die Gründerin der Hardwarefirma *Adafruit* und ihre zahlreichen Tutorials zur Elektroniksteuerung. Damit beim Ausfall eines Mikrocontrollers die Maschine weiterläuft, ist jedes Modul eine unabhängige Einheit, die mit den anderen nicht weiter kommuniziert, was den Programmieraufwand noch weiter begrenzte.

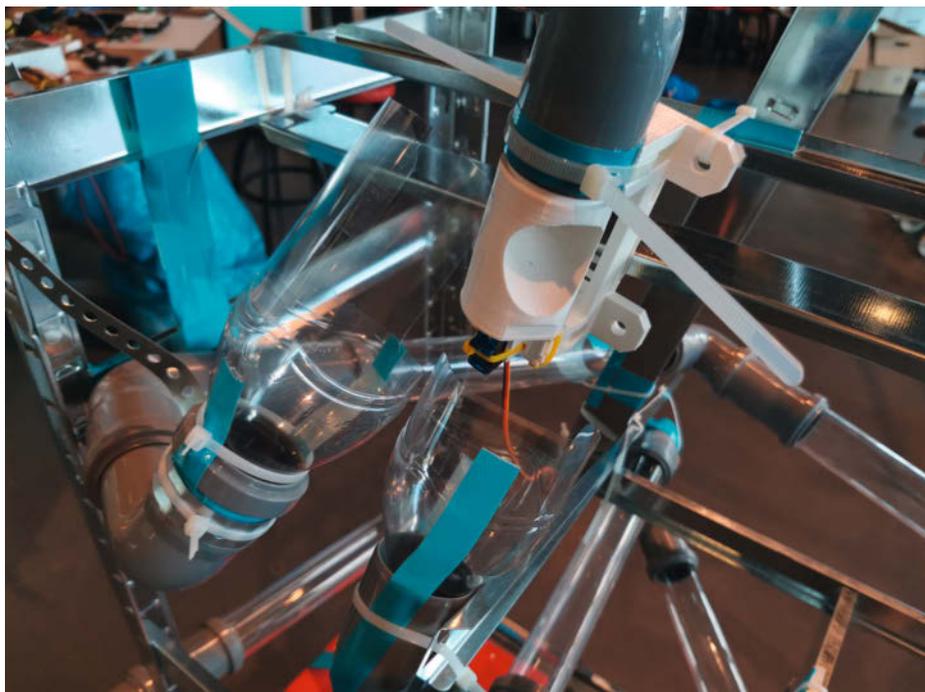
### Elektronik anbringen

Auch wenn die Maschine am Ende gewollt chaotisch aussieht und die Bastler sonst eher freihändig arbeiten – ein paar Tests und Überprüfungen mussten sein, damit die Smart Fairy Tale am Ende möglichst problemlos und wartungsfrei läuft. Den ersten Test galt es für die Animatronikszenen vor dem Einbau zu bestehen. Nur Projekte, die mindestens drei Mal hintereinander fehlerfrei funktionierten, durften eingesetzt werden. Wo genau auf dem Gerüst das sein könnte, war wiederum „Work in Progress“ und ergab sich im Lauf des Zusammenbaus. Um als „fertig“ eingebaut zu gelten, musste für jedes Projekt eine Checkliste mit drei Punkten erfüllt sein: Die zugehörige Lichtschranke brauchte eine Beschriftung, das Projekt einen eigenen Netzteilanschluss und alle Kabel und Schläuche mussten mit Zugentlastungen (meist Tape und Kabelbindern) gesichert sein.

Nach der Workshop-Woche waren so nicht nur 18 Szenen vorbereitet, sondern die meisten auch tatsächlich im Gerüst verbaut. Auch hier steckt unter dem wilden Äußeren ein strukturiertes Vorgehen, bei dem nicht einfach Arduinos zwischen den Acrylrohren baumeln. Die Elektronik ist jeweils mit Kabelbindern auf stabilen Grundplatten und diese wiederum am Gerüst befestigt. Je Turm liefert ein PC-Netzteil die nötigen Spannungen, die mit zwischengeschalteten Volt-Meter-Modu-



Welchen Weg die Kugeln durch die Maschine nehmen, entscheiden später die Besucherinnen und Besucher über ihr Smartphone.

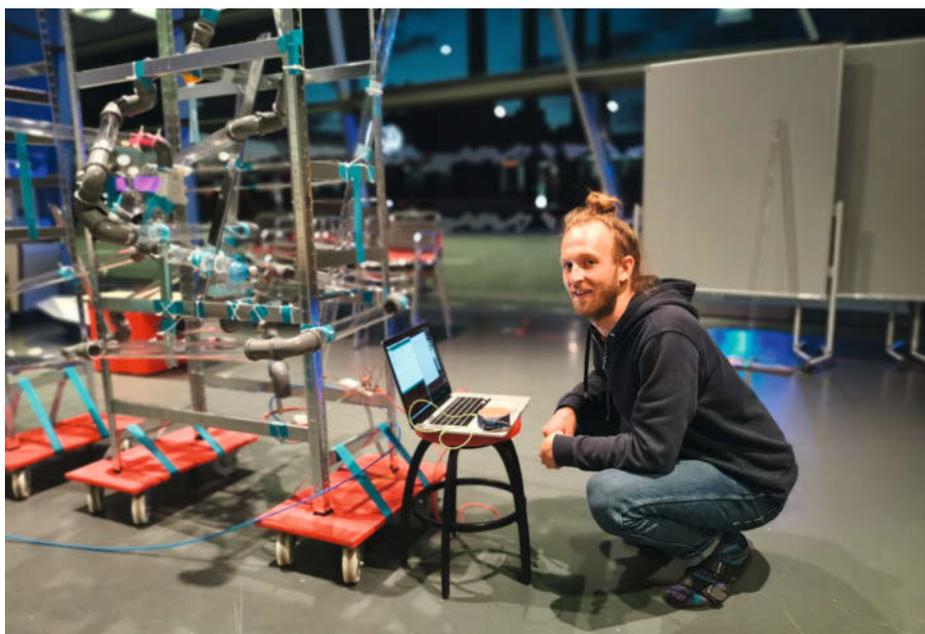


3D-gedruckte Teile verbinden das Rohrsystem ebenso wie zurechtgeschnittene Plastikverpackungen.

len einfach abzulesen und zu kontrollieren sind. Außerdem schwören die beiden bei der Elektroinstallation auf *Wago*-Klemmen, in denen Kabel ohne Schrauben eingeklemmt werden. Für den wilden Anblick sorgen zwei andere Tricks. LED-Scheinwerfer leuchten die Maschine grundsätzlich aus. Zusätzliche, bunt blinkende Neopixel lassen sie außerdem immer in Aktion erscheinen. Um zur Interaktion einzuladen, „muss etwas passieren, selbst wenn gerade niemand dran sitzt“ erklärt Fisgus.

## Highlights

Eine besondere Animation stammt von einem lokalen Künstler. Der Fotograf und Künstler Wolfgang Kowar ist mit seinen 70 Jahren kein Schüler mehr, tauchte aber trotzdem zum Workshop auf und baute fleißig mit. Er wollte einmal mit Pneumatik arbeiten und konnte sich mit einem aufblasbaren Paar an Luftpuppen seinen Traum erfüllen. Die Puppe auf dem motorisierten Bobbycar ist Roys Lieblingsprojekt, während es Fisgus der Kran angetan hat, der einen gelben Minion durch die Luft schwenkt. Mit Soundeffekten von den Spielzeugen, dem Zischen von Hydraulikzylindern und Klackern der Bälle durch die Rohre bietet die fertige Maschine auch einiges für die Ohren. Damit die Besucherinnen und Besucher die Maschine trotz der Scheibe nicht nur sehen, sondern sogar hören konnten, wurde der Sound vom Phaeno kurzerhand mit Mikrofonen aufgenommen und auf die Außenlautsprecher übertragen.



Felix Fisgus beim Einspielen neuer Software im Phaeno

Nur dort, vor dem Gebäude, war auch das WLAN aufgespannt, über das die Smart Fairy Tale gesteuert werden konnte. Das Interface hatte Fisgus so einfach wie möglich gestaltet, mit einem Foto der Maschine, über das lediglich Buttons für den Start und drei Weichen gelegt waren. Aufgrund der Größe der Maschine war aber immer nur ein Ausschnitt zu sehen. Nur wer sich einen Augenblick Zeit nahm und durch das Bild scrollte, konnte alle Varianten der Geschichte durchspielen. Neben der Sicherheit ein weiterer Vorteil der

lokalen Serverinstallation auf dem Raspi: Über das verwendete WebSocket-Protokoll werden alle verbundenen Geräte in Echtzeit aktualisiert. Drückt eine Person einen Button, ist dies nicht nur in der Maschine, sondern auch auf allen eingeloggten Smartphones zu beobachten.

Jetzt galt es nur noch einen letzten Test vor der Eröffnung zu absolvieren. Mit einem extra Knopf lassen sich die Bälle direkt durch die Maschine jagen und eine Dauerschleife simulieren, die in der Praxis selbst die ausgelassenen



Niklas Roy befestigt Breadboard und Elektronik für eine Animation auf einer stabilen Grundplatte.



Mit dem gesammelten Spielzeug durften Jugendliche aus der Region eigene Animatronik-Installationen bauen.

sten Jugendlichen nicht erreichten. Die zwei Wochen Einsatz auf der Phaenomenale Ende September überstand die Maschine dann nicht nur ohne Probleme – sie begeisterte so viele Menschen, dass sie sogar bis Mitte Januar weiterbetrieben wurde und ab März wieder aufgestellt werden wird.

### Noch einmal?

Für eine Neuauflage der *Smart Fairy Tale* – oder einer ähnlichen Maschine – haben die beiden

trotzdem ein paar Ideen. Im Belastungstest habe sich das Ballsystem als Schwachstelle erwiesen, für das es Verbesserungspotential gibt. Sinnvoller könnte ein geschlossenes und komplett 3D-gedrucktes System aus Weichen und Aufzügen sein. So sprangen die Kugeln über offene Stellen von einem Rohr ins nächste, etwa in Trichter aus abgeschnittenen Flaschen – was manchmal schief ging. Auch die Schalen zum Sammeln der Kugeln sind eine potentielle Schwachstelle, denn sie könnten überlaufen. Um in die Bahn eingreifen zu kön-

nen, sollten auch in einem geschlossenen System schließlich Wartungsöffnungen vorgesehen werden.

Bleibt am Ende noch die Frage: Was muss ich machen, damit das Phaeno nächstes Jahr *bei mir* anfragt? Auch dafür haben Roy und Fiskus eine Antwort: Coole Sachen basteln und im Internet zeigen. Irgendwann kommt jemand mit einem Auftrag. Auf den Webseiten der beiden sind daher auch viele weitere Bilder und Videos der Maschine zu sehen (siehe Kurzlink). —hch



Mittendrin, aber auf den ersten Blick kaum zu erkennen: Das Steuerboard mit Raspberry Pi, WLAN-Router und vielen Wago-Klemmen.



Unter dem Bobby Car ist ein LED-Spot versteckt, der die darunterstehende Animation ausleuchtet.

# Make: Projects

Gewinnt  
tolle  
Preise

## MAKERSPACE-CHALLENGE:

Verpasst eurem Space  
ein Upgrade

### Make:

Zeigt uns eure besten Makerspace-Hacks und Helferlein, vom misanthropischen Saugroboter über elaborierte Schlüsselsysteme bis zur praktischen Aufbewahrung für Kleinstelektronik.

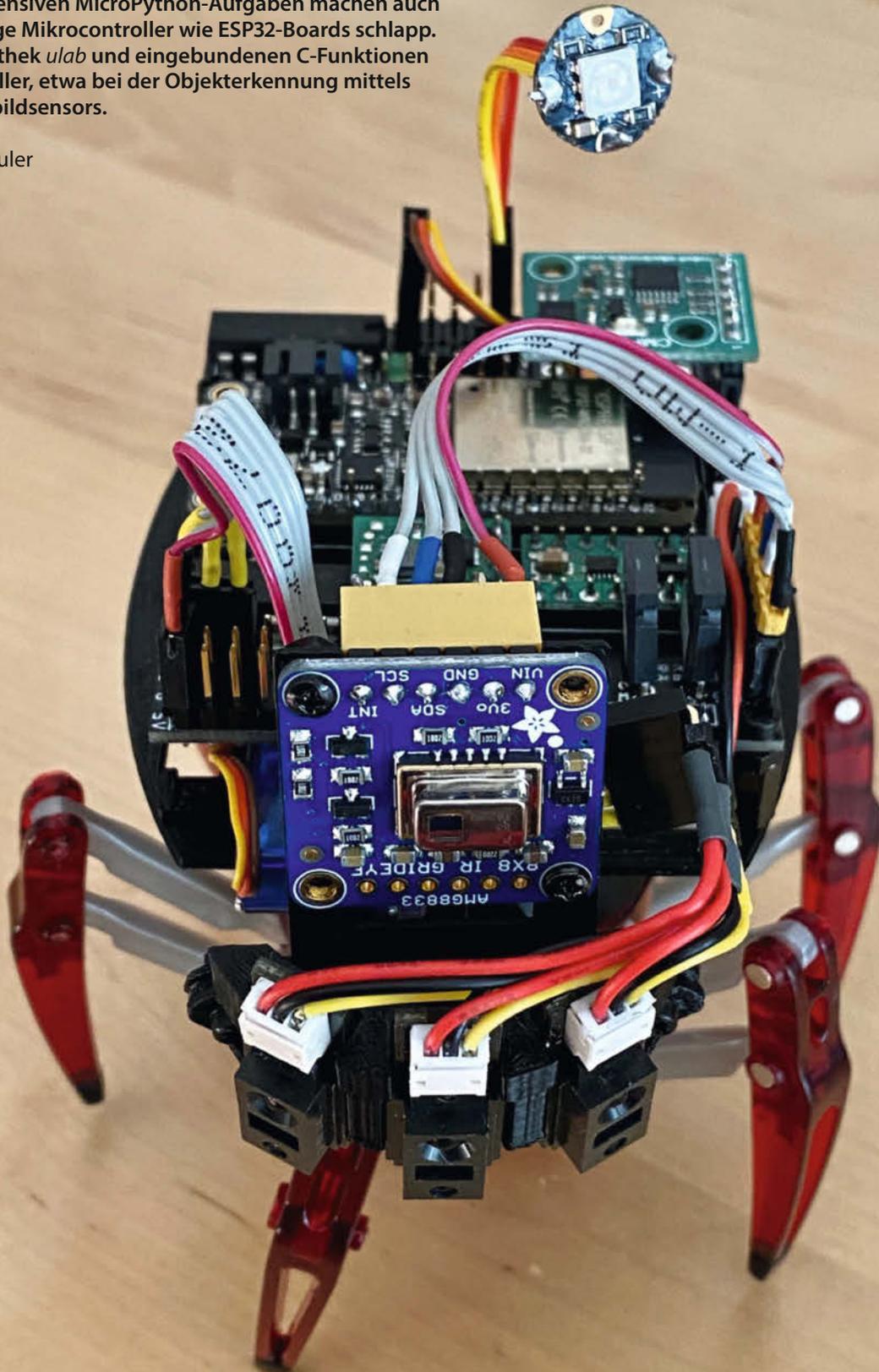
Meldet euch gemeinsam als Team für euren Makerspace, offene Werkstatt, Hackerspace und Co. auf Make Projects an. Erstellt ein Projekt und dokumentiert dort in Bild und Text, wie ihr alles gebaut habt.

[makeprojects.com/de/makerspacechallenge](https://makeprojects.com/de/makerspacechallenge)

# MicroPython beschleunigen

Bei rechenintensiven MicroPython-Aufgaben machen auch leistungsfähige Mikrocontroller wie ESP32-Boards schlapp. Mit der Bibliothek *ulab* und eingebundenen C-Funktionen geht es schneller, etwa bei der Objekterkennung mittels eines Wärmebildsensors.

von Thomas Euler



Trotz schneller Mikrocontroller-Boards stößt man bei komplizierteren Berechnungen in MicroPython schnell an Grenzen bei der Performance. Wie beim „echten“ Python wird ein Python-Skript vor dem eigentlichen Start zwar in einen optimierten Bytecode umgewandelt, die Programmausführung übernimmt aber ein Interpreter, was den Prozess verlangsamt. Zum Glück gibt es Möglichkeiten, Berechnungen in MicroPython gezielt zu beschleunigen. Als Beispiel dient die Erkennung von Objekten in Bildern einer einfachen Wärmebildkamera, direkt auf einem Mikrocontroller. Dies ermöglicht dem *Robotling*, dem MicroPython-betriebenen Laufroboter aus Make 3/19, S. 88, einer Hand autonom zu folgen.

In diesem Artikel stellen wir zwei Optimierungswege vor. Die erste Möglichkeit besteht darin, die Bibliothek *ulab* in die MicroPython-Firmware einzubinden, die für effiziente numerische Berechnungen optimiert wurde. Die zweite Möglichkeit ergänzt die Firmware um neue Befehle, die als C-Funktionen implementiert werden, da diese schneller ausgeführt werden. Beide Wege erfordern derzeit, die Mikrocontroller-Firmware neu zu kompilieren, was aber dank der hervorragenden MicroPython-Dokumentation relativ einfach ist (siehe Kasten auf der nächsten Seite). Eine aktuelle Firmware mit *ulab* inklusive habe ich daher auch zum Download vorbereitet (siehe Link in der Kurzinfor). Außerdem kann man *ulab* unter *CircuitPython* verwenden, der besonders einsteigerfreundlichen MicroPython-Variante von Hardware-Hersteller Adafruit. Dort gehört es seit Version 6 zur Grundausstattung. Dann muss man allerdings einen anderen Mikrocontroller nutzen, da *CircuitPython* den klassischen ESP32 derzeit nicht unterstützt.

### Objekte im Bild finden

Als Beispiel dient eine Aufgabenstellung aus der Bildverarbeitung: Eine Funktion soll ein

## Kurzinfor

- » *ulab* in MicroPython einbinden
- » Wärmebildsensor auslesen
- » Objekterkennung mit C-Funktionen
- » Firmware unter Windows 10 mit Linux-Subsystem kompilieren

### Checkliste

-  **Zeitaufwand:**  
ein Nachmittag
-  **Kosten:**  
80 Euro
-  **Programmieren:**  
Python-Kenntnisse

### Material

- » HUZZAH32-Board
- » Wärmebildsensor AMG833 Grid-EYE
- » Jumperkabel
- » Breadboard
- » USB-Kabel
- » PC

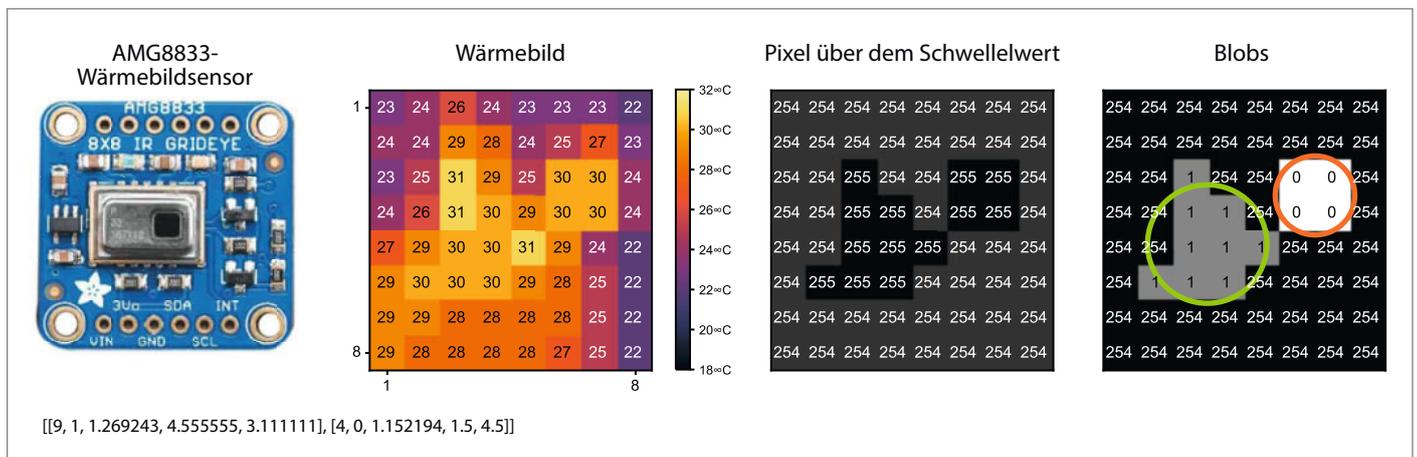
### Mehr zum Thema

- » Thomas Euler, Einstieg in MicroPython, Make 2/19, S. 104
- » Thomas Euler, Krabbeltier mit Python-Antrieb, Make 3/19, S. 88
- » Thomas Euler, Krabbelroboter sendet Telemetrie, Make Sonderheft Roboter 2019, S. 112

↓ Alles zum Artikel im Web unter [make-magazin.de/x8cb](https://make-magazin.de/x8cb)

Bild nehmen, darin Objekte erkennen und als Liste von Objektkoordinaten und -größen zurückerliefern. Als Kamera nutzen wir den Wärmebildsensor AMG833 Grid-EYE von Panasonic. Er ist gerade sehr populär, da er Temperaturbilder direkt in Grad Celsius ausgibt und so in der Covid19-Pandemie für ein erstes Fieber-Screening verwendet werden kann. Er erfasst ein Blickfeld von etwa 60 Grad und liefert alle 100 Millisekunden ein 8x8 Pixel großes Bild. Der Sensor ist zum Beispiel als Breakout-Modul von Adafruit erhältlich (Bezugsquellen siehe Kurzlink). Er besitzt eine I<sup>2</sup>C-Schnittstelle und kann leicht von einem Mikrocontroller ausgelesen werden.

Wirkliche Objekte wie eine Petunie oder einen Wal können wir damit natürlich nicht erkennen. Stattdessen geht es hier um zusammenhängende Bildbereiche (*Blobs*), die sich vom Hintergrund absetzen, weil sie wärmer als dieser sind. Dies können etwa eine Hand oder eine heiße Teetasse sein. Der Algorithmus zur Blob-Erkennung geht folgendermaßen vor: Zuerst berechnet er den Mittelwert und die Standardabweichung der Temperatur über alle 64 Bildpixel. Die Standardabweichung ist ein Maß dafür, wie stark die Temperatur um diesen Mittelwert schwankt, also wie variiert das Bild ist. Als nächstes sucht der Algorithmus alle Pixel, die mindestens um



Von links nach rechts: AMG833 Wärmebildsensor auf Breakoutboard, ausgelesenes 8x8-Pixel-Bild mit Temperaturwerten in Grad Celsius, Pixel über dem Schwellenwert und erkannte Blobs

# MicroPython kompilieren

Um die MicroPython-Firmware für ESP32-Mikrocontroller zu kompilieren, muss man die entsprechenden Linux-Werkzeuge („*tool chain*“) installieren. Das erledigt man unter Windows 10, indem man zunächst dessen Linux-Subsystem aktiviert (WSL) und dort Ubuntu 18.04 LTS aus dem Microsoft-Store installiert. Eine ausführliche Anleitung der weiteren Schritte finden Sie in meinem Github-Repository (siehe Link in der Kurzinfo). Der Zeitaufwand beträgt insgesamt eine knappe Stunde, wobei man die meiste Zeit mit Warten verbringt. Anschließend versichert man sich, dass der „Build“-Prozess erfolgreich durchläuft und die Firmware-Datei

(*firmware.bin*) erzeugt. Jetzt kann man weitere Funktionen zu MicroPython hinzufügen. Um zum Beispiel *ulab* einzubinden, klonet man den Code von Github und baut die Firmware erneut:

```
mkdir ~/modules
cd ~/modules
git clone https://github.com/v923z/micropython-ulab.git ulab
cd ~/micropython/ports/esp32
make -j 2 USER_C_MODULES=../../modules/ulab
```

Der Parameter `-j` legt fest, wie viele Kerne der Compiler verwenden darf. Wer einen PC mit vielen Kernen sein eigen nennt, kann den Wert hinter `-j` schrittweise erhö-

hen und versuchen, den Übersetzungsvorgang zu beschleunigen. Falls hierbei Fehler auftreten, kann ein `make clean` helfen, um aufzuräumen. Anschließend überträgt man die neue Firmware auf einen über USB angeschlossenen HUZZAH32-Mikrocontroller, wobei `make purge` nur notwendig ist, wenn man den kompletten Flash-Speicher löschen möchte.

```
make purge
make deploy
```

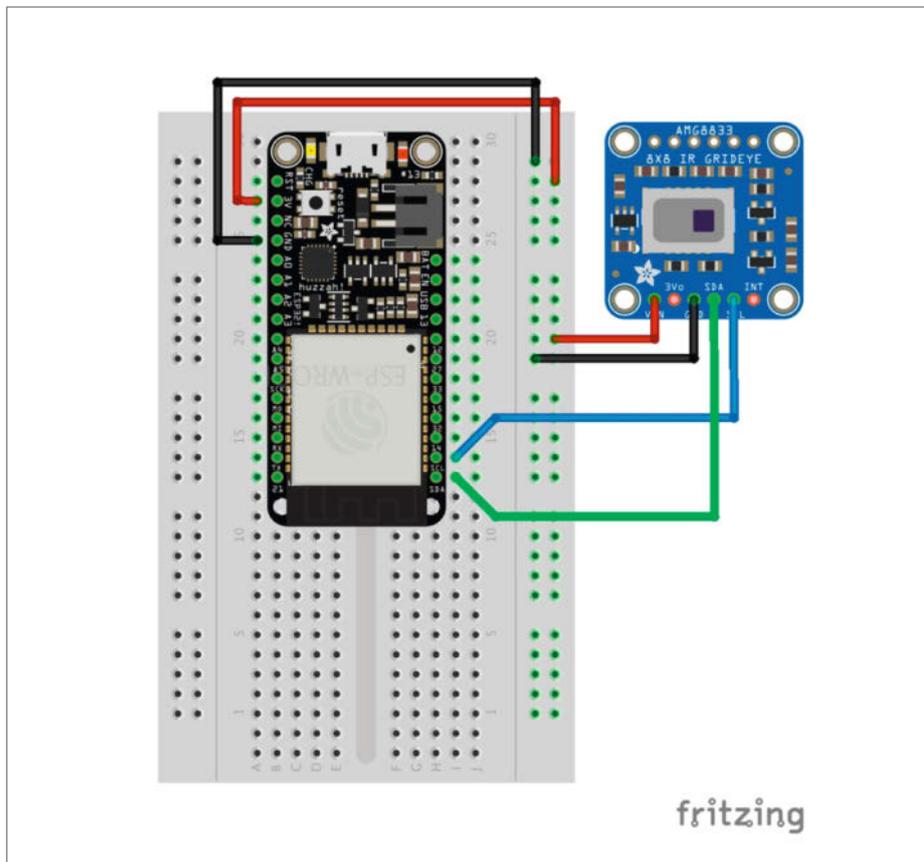
Nun sollte man in der MicroPython-Kommandozeile REPL mit `import ulab` die neue Bibliothek importieren können.

einen bestimmten Betrag, den Schwellenwert, wärmer als der Mittelwert sind und markiert sie.

Anschließend sucht der Algorithmus Gruppen von nebeneinander liegenden, markierten Pixeln, den Blobs. Hierzu nimmt er einen markierten Pixel als Startpunkt und sucht so-

lange dessen Umgebung ab, bis keine direkt benachbarten Pixel mit Markierung mehr gefunden werden. Die Pixel, die zu diesem Blob gehören, werden mit einer eindeutigen Nummer versehen und aus dem Pool der markierten Pixel genommen. Nun nimmt sich der

Algorithmus den nächsten markierten Pixel vor, bis alle einem Blob zugeordnet sind. Schließlich werden die Blobs nach der Größe ihrer Fläche in absteigender Reihenfolge sortiert, sodass sich immer die größten vorne in der Liste befinden, die die Funktion zurück gibt. Den Code zur reinen Python-Version dieses Algorithmus findet man in *blob.py* ebenfalls in meinem Github-Repository. Er ist sicherlich nicht der eleganteste, aber es geht ja darum zu demonstrieren, wie man On-board-Berechnungen beschleunigt.



Der AMG8833-Sensor hängt am I<sup>2</sup>C-Bus des HUZZAH32-Boards und wird von diesem mit 3,3 Volt Spannung versorgt.

## AMG8833-Wärmebildsensor auslesen

Die Beispiele in diesem Artikel funktionieren in der Regel mit jedem Mikrocontroller, auf dem MicroPython läuft und bei dem man auf die I<sup>2</sup>C-Schnittstelle zugreifen kann. Auch Boards, die Adafruits MicroPython-Ableger CircuitPython verwenden, sollten mit kleinen Anpassungen funktionieren. Wir verwenden hier das HUZZAH32-Board von Adafruit, das mit einem WLAN-fähigen ESP32-Chip ausgestattet ist und für das es eine sehr ausgereifte MicroPython-Portierung gibt. Die notwendige Verkabelung zwischen AMG8833-Sensor und dem Board ist übersichtlich. Auf dem Board muss die aktuelle MicroPython-Firmware ( $\geq 1.13$ ) und auf dem PC eine entsprechende Test- und Entwicklungsumgebung installiert sein, etwa *Atom* mit dem *pymakr*-Plugin. Die Einrichtung von Controller und Umgebung haben wir im Artikel „Einstieg in MicroPython“ in der Make 2/19 ab Seite 104 beschrieben.

Um das erste Beispielprogramm *test\_amg8833.py* (Listing 1) in der REPL auszuführen, müssen der Sensortreiber (*amg8xx.py*) und der Ordner mit Hilfsfunktionen (*helper*) in den

Flash-Speicher geladen werden. Alle Dateien finden Sie über den Kurzlink auf Github. Das Programm wird auf dem HUZAH32 ausgeführt: Es liest die Bilder vom AMG8833-Sensor und misst die Zeit, die die Blob-Erkennung benötigt.

Nach den notwendigen Imports (Zeilen 1 bis 3) wird ein I<sup>2</sup>C-Bus-Objekt erzeugt, das Pin 22 für die Takt- (SCL) und Pin 23 für die Datenleitung (SDA) verwendet und den Bus mit 400 kHz betreibt (Zeile 6). Zeile 7 erzeugt das Sensorobjekt und übergibt den I<sup>2</sup>C-Bus. Nun könnten Wärmebilder ausgelesen werden (Zeile 10). Diese bestehen hier aus einer Liste von 8x8, also 64 Temperaturwerten in Grad Celsius.

Die Zeilen 13-27 implementieren eine Funktion, die *n* Bilder einliest, darin nach Blobs sucht (Zeile 18) und, falls welche gefunden wurden, diese ausgibt (Zeile 20-25). Die Funktion `find_blobs` liefert eine Liste, in der jeder Eintrag eine Blob-Beschreibung (auch als Liste) enthält. Diese Beschreibung gibt etwa an, wo im Bild der Mittelpunkt des Blobs liegt und wie groß er ist. Gleichzeitig wird die Zeit gemessen, die der Aufruf von `find_blobs` dauert (Zeilen 17 und 19), um am Ende die gemittelte Dauer ausgegeben zu können (Zeile 27). So können wir messen, welche Implementierung des Algorithmus am schnellsten ist. Schließlich importieren die restlichen Zeilen (29 bis 39) im Listing 1 verschiedene Implementierungen von `find_blobs` und messen deren Performance. Zeile 42 schließlich gibt den I<sup>2</sup>C-Bus frei.

## Schnelle Berechnungen mit ulab

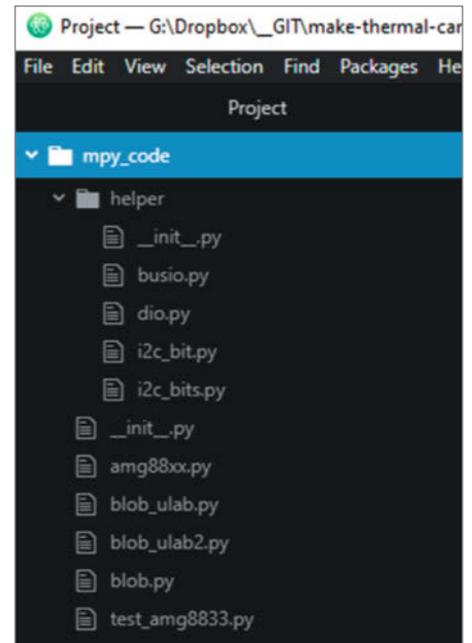
Die reine MicroPython-Implementierung von `find_blobs` in `blob.py` benötigt im Schnitt etwa 20 Millisekunden pro Aufruf, wobei die Dauer vom Bildinhalt abhängt: in einem Bild mit einem deutlich begrenzten Objekt (z.B. einer heißen Teetasse) funktioniert die Blob-Erkennung üblicherweise schneller als in einem verrauschten Bild ohne eindeutige Objekte. Zwar erscheinen 20 Millisekunden nicht wirklich lang, aber man muss bedenken, dass die Bildabfrage ebenfalls etwas Zeit benötigt (ca. 8 Millisekunden). Wenn man mit der maximalen Sensorbildrate von 10Hz arbeiten möchte, bleiben nur etwas mehr als 70 Prozent der Rechenzeit pro Zyklus übrig.

Die Berechnung beschleunigen wir nun mit der Bibliothek `ulab`. Sie ist auf die Manipulation und Berechnung von Vektoren und Matrizen ausgelegt und orientiert sich stark an `numpy`, einer der Python-Standardbibliotheken für wissenschaftliche Berechnung – `numpy`-Kenner finden sich daher sofort zu recht. Allen anderen sei die hervorragende Dokumentation empfohlen (siehe Link). Aus Speicherplatzgründen beschränkt sich `ulab` auf eine Untermenge der `numpy`-Funktionali-

tät. Die Bibliothek wird von einer sehr aktiven Gruppe um den Programmierer Zoltán Vörös weiterentwickelt. Sie ist in C geschrieben und daher sehr schnell. In unserem Beispiel wird `ulab` vom Modul `blob_ulab.py` benutzt, um das Finden von Blobs zu beschleunigen (Zeilen 34 und 35 in Listing 1). Obwohl sich in diesem Fall nur manche Teile des Codes mit `ulab` optimieren lassen (siehe Beispiel im Kasten), halbiert sich die Dauer pro Aufruf auf ca. 9 Millisekunden.

## MicroPython um eigene Funktionen erweitern

Die zweite, aufwendigere Möglichkeit ist, einen Blob-Algorithmus in C zu implementieren und als Modul in die MicroPython-Firmware einzubinden. Hier bietet es sich an, `ulab` zu erweitern. Die Bibliothek enthält ein Modul namens `user`, das für solche Erweiterungen gedacht ist. Voraussetzung ist, dass die `tool chain` zum Bauen der Firmware wie bereits beschrieben für den HUZAH32 ein-



Die Pythonprogramme im Flashspeicher

### 1. test\_amg8833.py

```
01 from time import ticks_us, ticks_diff, sleep_ms
02 from amg88xx import AMG88XX
03 from helper import busio
04
05 try:
06     i2c = busio.I2CBus(400000, 22, 23)
07     amg = AMG88XX(i2c)
08     sleep_ms(500)
09
10     img = list(amg.pixels_64x1)
11     print("Image:", img)
12
13     def find_some_blobs(n):
14         delta = 0
15         for i in range(n):
16             img = list(amg.pixels_64x1)
17             t = ticks_us()
18             res = blob.find_blobs(img, (8, 8))
19             delta += ticks_diff(ticks_us(), t)
20             s = ""
21             for b in res:
22                 if b[ > 1:
23                     s += "{0:.0f}-pixel blob at {1:.1f},{2:.1f} ".format(b[1], b[2],
24                                     b[3])
25                     if len(s) > 0:
26                         print("{0:2.0d}: {1}".format(i, s))
27                         sleep_ms(200)
28                         print('{:6.3f}ms per call (average)\n'.format(delta/(n*100)))
29
30     print("Find blobs (just MicroPython):")
31     import blob
32     find_some_blobs(20)
33
34     print("Find blobs (MicroPython +ulab):")
35     import blob_ulab as blob
36     find_some_blobs(20)
37
38     print("Find blobs (ulab +C code):")
39     import blob_ulab2 as blob
40     find_some_blobs(20)
41 finally:
42     i2c.deinit()
```

## Schneller Pixel markieren

Als Beispiel dient der Code, der in der reinen MicroPython-Version alle Pixel oberhalb des Temperaturschwellenwerts markiert:

```
pMsk = array.array("B", [0]*nPix)
nThres = 0
for i in range(nPix):
    if pImg[i] >= thres:
        pMsk[i] = 255
        nThres += 1
```

Durch die Verwendung von *ulab*-Arrays für *pImg* und den darauf optimierten Funktionen (z.B. *np.sum*, mit *from ulab import numpy as np*), wird der Code kompakter und schneller:

```
pMsk = (pImg >= thres) *255
nThres = int(np.sum(pMsk) /255)
```

gerichtet ist und getestet wurde. Man fügt dann die eigenen Funktionen der Datei *user.c* im Verzeichnis */ulab/code/user* hinzu und teilt *ulab* mit, dass das *user*-Verzeichnis berücksichtigt werden soll, indem man die Definition *ULAB\_HAS\_USER\_MODULE* gegen Ende von *ulab.h* auf (1) ändert. Nun baut man die Firmware neu.

Um unsere C-Version des *find\_blobs*-Algorithmus einzubinden, kopiert man den Inhalt des Ordners *blob\_ulab2* aus dem Projekt-Repository nach */ulab/code/user*; existierende

## 2. user.c

```
01 static mp_obj_t user_dummy(mp_obj_t arg_obj) {
02     if(!mp_obj_is_type(arg_obj, &ulab_ndarray_type)) {
03         mp_raise_TypeError(translate("`dummy` requires an ndarray"));
04     }
05     ndarray_obj_t *arg_arr = MP_OBJ_TO_PTR(arg_obj);
06     mp_float_t *arg_items = (mp_float_t *)arg_arr->array->items;
07     mp_float_t sum, mean, sd;
08     sum = 0;
09     for(size_t i=0; i<arg_arr->len; i++) {
10         sum += arg_items[i];
11     }
12     mean = (mp_float_t)sum /arg_arr->len;
13     sum = 0;
14     for(size_t i=0; i<arg_arr->len; i++) {
15         sum += (mp_float_t)pow(arg_items[i] -mean, 2);
16     }
17     sd = (mp_float_t)sqrt(sum /arg_arr->len);
18     mp_obj_t templ = mp_obj_new_list(0, NULL);
19     mp_obj_list_append(templ, mp_obj_new_float(mean));
20     mp_obj_list_append(templ, mp_obj_new_float(sd));
21     return templ;
22 }
```

Dateien überschreibt man. In unserem Beispiel werden weitere C-Dateien benötigt: *stack.h* und *stack.c*. Dies teilt man dem Compiler mit, indem man die Datei */ulab/code/micropython.mk* um folgende Zeile ergänzt:

```
SRC_USERMOD += $(USERMODULES_DIR)/
user/stack.c
```

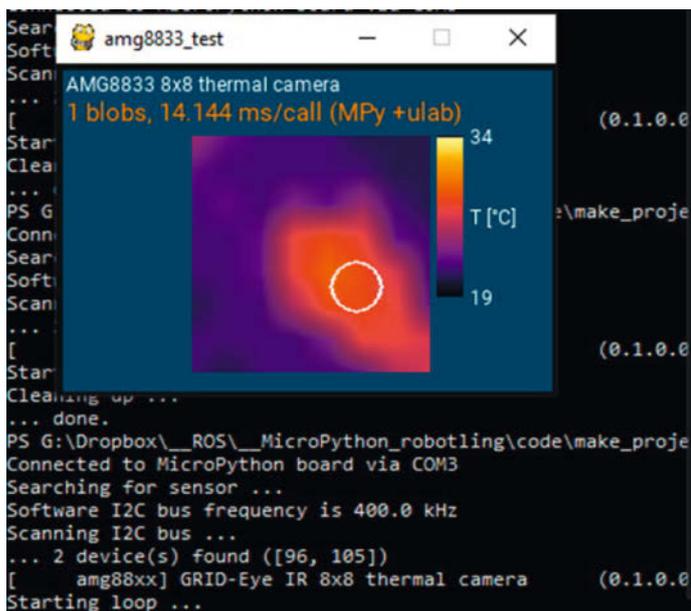
Sobald die Firmware neu übersetzt und auf den Mikrocontroller übertragen wurde, kann die in C geschriebene Version von *find\_blobs* importiert werden:

```
from blob_ulab2 import find_blobs
```

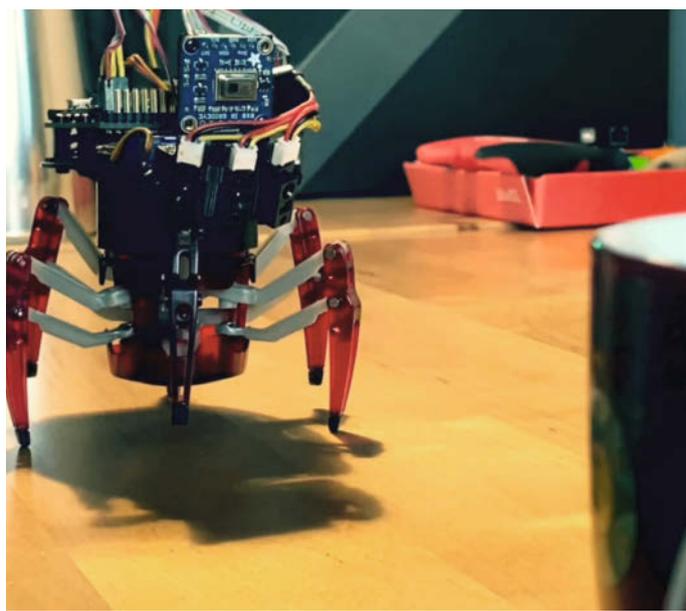
Im Listing 1 passiert das in den Zeilen 38 und 39. Die C-Version benötigt inklusive des Extraaufwands, Objekte von Python nach C und

wieder zurück zu konvertieren, im Schnitt nur eine Millisekunde pro Aufruf und ist damit 20 mal schneller als die MicroPython-Version. Mit einem optimierten Blob-Algorithmus ließe sich das wahrscheinlich noch verbessern.

Wie wird eine C-Funktion nun in *ulab* und damit in MicroPython eingebunden? Die Datei *user.c* (Listing 2) demonstriert dies anhand eines Beispiels: Die Funktion *dummy* nimmt ein MicroPython-Array entgegen, berechnet Mittelwert und Standardabweichung und gibt diese beiden Werte als Liste zurück. Zunächst wird in den Zeilen 2 bis 4 überprüft, ob das Funktionsargument *arg\_obj* tatsächlich vom Typ *ndarray* ist, sonst wird ein Fehler ausgegeben. Die nächsten Zeilen vereinfachen den Zugriff auf die Daten im MicroPython-Objekt



Das zweite Beispielprogramm holt sich Wärmebilder vom Mikrocontroller und zeigt sie zusammen mit den gefundenen Blobs auf dem Bildschirm an.



Der Robotling mit AMG8833-Wärmebildsensor auf dem Weg zum heißen Teebecher.

arg\_obj: Mit `arg_item[0]` kann zum Beispiel auf den ersten Wert des Arrays zugegriffen werden. Dann werden Mittelwert (Zeilen 8 bis 12) und Standardabweichung (Zeilen 13 bis 17) der Werte im Array berechnet. Schließlich wird eine leere MicroPython-Liste erzeugt (Zeile 18), zu der die beiden Werte hinzugefügt werden (Zeilen 19 und 20). In Zeile 21 wird diese Liste als MicroPython-Objekt zurückgegeben.

Zusätzlich müssen wir `ulab` mit der neuen Funktion bekannt machen, damit wir sie später in MicroPython benutzen können. Wie das geschieht, kann man sich im hier nicht gezeigten Rest von `user.c` anschauen. Die `ulab`-Dokumentation enthält unter *Programming ulab* eine detaillierte Beschreibung mit weiteren Beispielen. Ist die Beispielfunktion eingebunden und die neue Firmware auf den HUZ-ZAH32 übertragen, so kann man sie verwenden:

```
>> from ulab import numpy np
>> from ulab import user
>> a = np.array([17, 41, 94, 66, 35,
20, 58, 59, 29, 71])
>> user.dummy(a)
[49.0, 23.46061]
```

Um zu überprüfen, ob wir auch richtig gerechnet haben, wiederholen wir die Berechnung mit den `ulab`-Standardfunktionen:

```
>> print(np.mean(a), np.std(a))
49.0 23.46061
```

### Wärmebilder auf den PC streamen

Das zweite Beispielprogramm (`test_amg8833_gui.py`) ist ein Python-Skript für den PC. Es wurde unter Windows 10 getestet, sollte aber nach Anpassung auch unter Linux laufen. Der Mikrocontroller mit dem Wärmesensor muss über USB mit dem PC verbunden sein und dieselben Dateien im Flash-Speicher haben, wie bereits für Beispiel 1 benötigt. Das Programm holt sich kontinuierlich Wärmebilder

## Pyboard

Unter den Möglichkeiten, vom PC aus mit MicroPython auf einem Mikrocontroller zu interagieren, bietet das Python-Modul `pyboard` eine der interessantesten. Es gehört zum Lieferumfang von MicroPython und erlaubt zum Beispiel, über eine USB-Verbindung, MicroPython-Befehle auf dem Mikrocontroller abzusetzen. Es ist quasi eine Fernsteuerung für die MicroPython-Kommandozeile REPL (Read Evaluate Print Loop). Daher muss hierfür kein Programm auf den Mikrocontroller geladen werden – außer etwaigen Treibern für externe Hardware, wie hier dem AMG8833-Sensor. Der folgende Code kann auf dem PC in einem Python-Skript ausgeführt werden. Voraussetzung ist, dass sich `pyboard.py` im selben Verzeichnis wie das Skript befindet und

der Mikrocontroller über USB (hier COM3) mit dem PC verbunden ist.

```
>> import pyboard
>> pb = pyboard.Pyboard(COM3)
>> pb.enter_raw_repl()
>> result = pb.exec("print(1+2)")
>> print(result[:-2].decode())
3
```

Das Ergebnis (`result`) ist ein Byte-Array (hier `'b'3\r\n'`), das die Ausgabe enthält, also das, was MicroPython in die REPL druckt. Im Beispiel oben wandeln wir das Byte-Array in einen String um und drucken es aus (3). Das Beispielprogramm `test_amg8833_gui.py` benutzt `pyboard`, um die Daten von der Wärmebildkamera am Mikrocontroller auf den PC zu streamen.

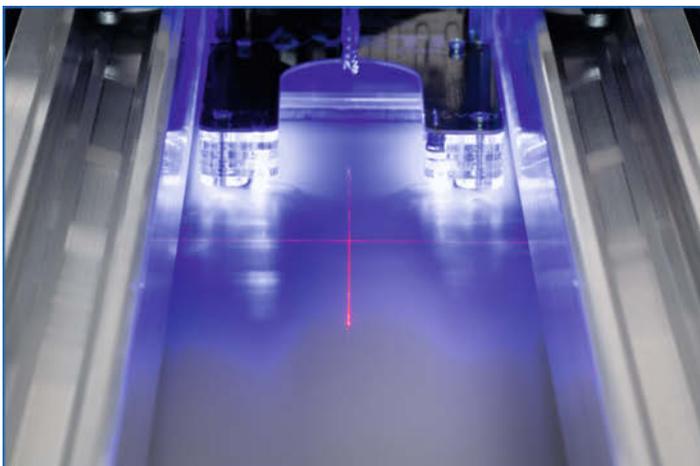
und detektierte Blobs vom Mikrocontroller und zeigt sie auf dem Bildschirm an. Mit den Tasten 1, 2 und 3 kann man zwischen den verschiedenen Versionen des Blob-Algorithmus wechseln; die Taste S schaltet einen Glättungsfilter hinzu. Das Programm nutzt die `pyboard`-Schnittstelle (siehe Kasten) und demonstriert damit eine interessante und mächtige Möglichkeit, von einem Python-Programm aus mit MicroPython auf dem Mikrocontroller zu kommunizieren.

### Ausblick

Ausgerüstet mit dem Wärmebildsensor kann der Robotling autonom einem warmen Objekt wie einer Hand folgen, indem er sich immer an der Position des größten Blobs im Bild ausrichtet. Über den Kurzlink finden Sie ein Video, in dem das Prinzip zu sehen ist: Die Kamera und

damit der Robotling folgt der Bewegung des heißen Teebechers. Das Wärmebild, das der Roboter „sieht“, wird auf einem PC angezeigt, an den es zeitgleich über MQTT-Telemetrie übermittelt wird. Das Verhalten des Roboters ist simpel: Er hält in regelmäßigen Abständen an, dreht sich in Richtung der größten Wärmequelle und läuft dann darauf zu.

Die Anwendung solcher Wärmebildsensoren in der Covid19-Pandemie wurde bereits erwähnt; der hier skizzierte Blob-Algorithmus wäre zum Beispiel eine Möglichkeit zu erkennen, ob eine erhöhte Temperatur lokal im Bild vorkommt, was auf eine Person mit Fieber hinweisen könnte. Die Lokalisierung von Blobs ist aber nicht nur auf Wärmebilder beschränkt – so gehören „intelligente“ Kameras, die mittels ausgefeilter Algorithmen farbige Objekte lokalisieren, schon lange zum Standardrepertoire der Robotik. —hch





neu. innovativ. professionell.

### Mobile CNC-Fräse SMARTBENCH

- ✓ Großformatige Bearbeitung bis 2.500 x 1.250 mm
- ✓ Intuitive Bedienung über Touch-Pad
- ✓ Doppelte Z-Achsen Spindelführung
- ✓ Kreuzlinienlaser für X/Y-Nullpunkt
- ✓ Echtzeit-Anzeige der aktuellen Motor-Last

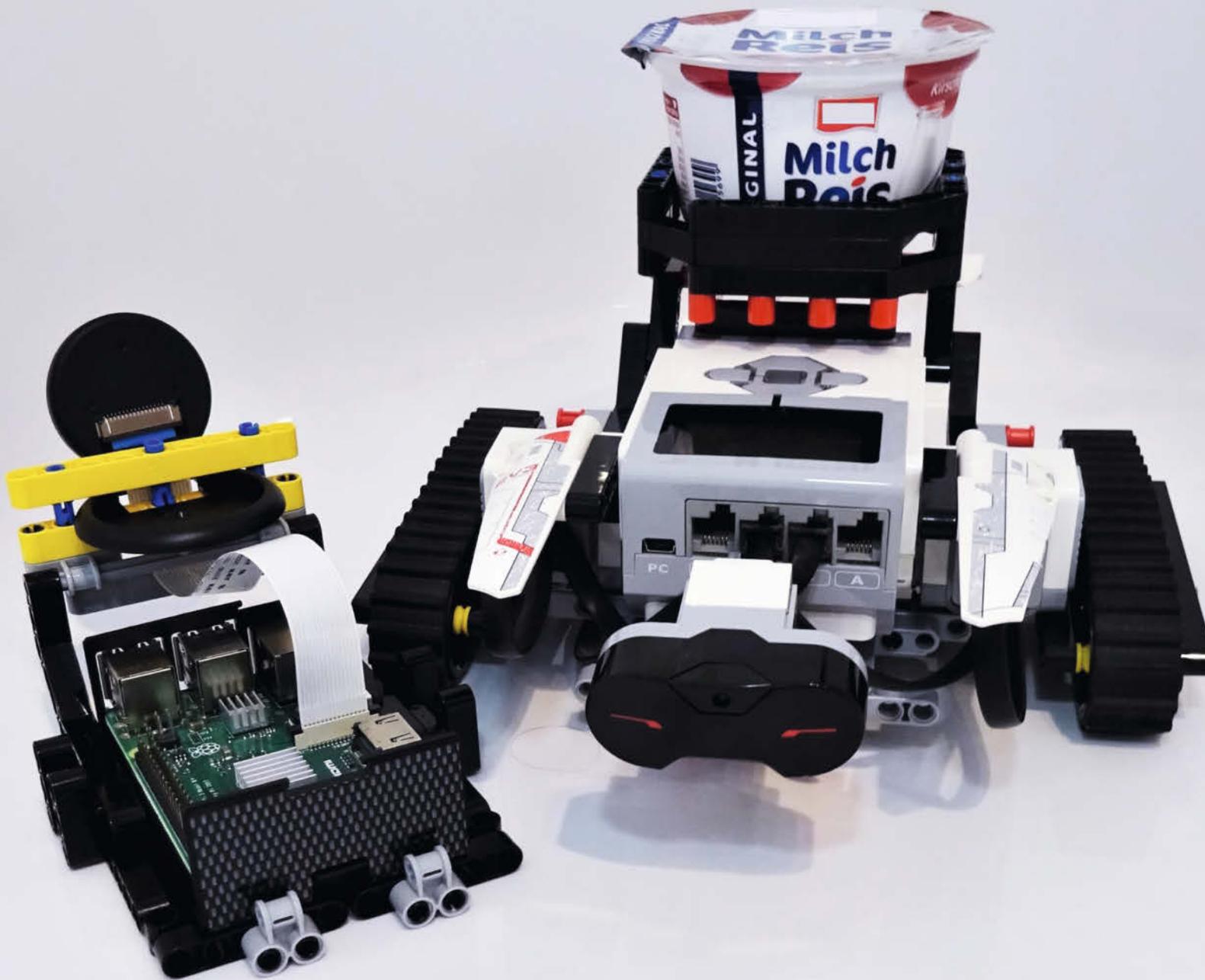
+49 (0)8143 /99129-0  
[www.sauter-shop.de/cnc-fraese/](http://www.sauter-shop.de/cnc-fraese/)



# Pfadplanung für Roboter

Autonome Roboter bewegen sich ohne menschliche Eingriffe durch Industriehallen auf der Erde und auf der Oberfläche entfernter Planeten. Ein Pfadplanungs-Algorithmus berechnet dabei den kürzesten Fahrweg zum gewünschten Ziel. Wir zeigen als Beispiel, wie ein Lego-EV3-Roboter so programmiert wird, dass er Milchreis auf einer Terrasse serviert. Wer keinen Roboter hat, kann mit unserer Simulationssoftware trotzdem eigene Experimente durchführen.

von Detlef Heinze



Im Jahr 2004 landeten die zwei NASA-Mars-rover *Spirit* und *Opportunity* auf dem Mars. Die Signalübertragung zum Mars dauert bis zu 24 Minuten – in eine Richtung. Eine Fernsteuerung durch Menschen ist daher nicht möglich. Die beiden Roboter nutzten daher das Ergebnis einer Pfadplanung und fuhren dann autonom ihre Wege zu ihren Missionszielen. Dieses Prinzip setzen NASA und ESA auch bei den nachfolgenden Marsmissionen ein.

*Spirit* und *Opportunity* verwendeten ab 2006 einen Algorithmus aus der sogenannten D\*-Familie. Aus dieser Familie von Algorithmen setzen wir hier für unseren *PathRunner* genannten EV3-Roboter den Algorithmus D\*-Lite ein (gelesen *D-Star-Lite*). Er ist in Python implementiert und mit einer grafischen Oberfläche für den menschlichen Operator versehen. Das Programm läuft auf dem Raspberry Pi und heißt *Interactive D\*Lite*.

### Pfadplanung

Pfadplanungsalgorithmen gehören mit zu den ältesten Algorithmen der künstlichen Intelligenz. Die meisten kennen sie von der Navigation im Auto oder vom Routenplaner auf Google Maps. Schon 1959 hat der niederländische Mathematiker Edgar W. Dijkstra einen Algorithmus zur Pfadplanung beschrieben, mit dem sich der kürzeste Weg zwischen zwei Punkten vorab berechnen lässt. Im Prinzip hat man dabei Orte (Knoten) durch die man fahren kann, die durch Kanten (Straßen) miteinander verbunden sind. Den Kanten ordnet man Kosten zu (Länge der Straße). Das Geflecht aus den per Kanten miteinander verbundenen Kanten nennt man auch Graph.

Anhand der Zahl der Knoten, deren Verbindungen über Kanten und den Kantenge-

## Kurzinfo

- » Interaktive Pfadplanung verstehen und anwenden
- » Raspberry Pi steuert Lego-EV3-Roboter über Bluetooth
- » Programmierung mit Python und Lego EV3 Mindstorms
- » Bedienen des Pfadplanungssimulators

### Checkliste

- Zeitaufwand:**  
4 Stunden
- Kosten:**  
60 Euro (ohne Lego-Baukasten)
- Programmieren:**  
Python unter Linux oder Windows

### Material

- » Raspberry Pi ab Modell 3B+ oder höher (ohne Kamera)
- » MicroSD-Speicherkarte mit installiertem Raspbian Buster oder Raspberry Pi OS Buster (beide Python 3.7.3)
- » Lego-EV3-Baukasten 31313

---

### Mehr zum Thema

- » Detlef Heinze, Pi steuert Lego EV3, Make Sonderheft Robotik 2019

↓ Alles zum Artikel im Web unter [make-magazin.de/xb4a](http://make-magazin.de/xb4a)

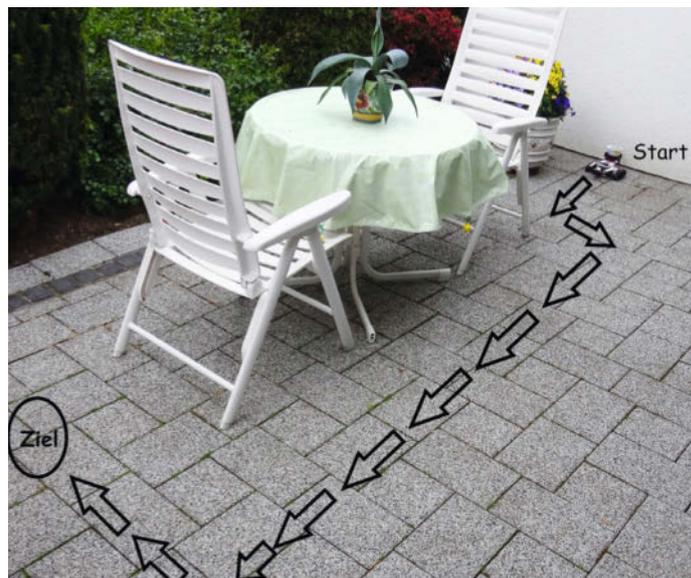
wichtungen kann der Dijkstra-Algorithmus den kürzesten Weg per Programm von einem vorgegebenen Startknoten zu einem Zielknoten iterativ ermitteln. Allerdings geht er dafür relativ stur alle Knoten und Kanten durch.

Weiterentwicklungen wie A\* (gelesen *A-Star*) verwenden für den Start zusätzlich eine erste Abschätzung welche Kanten von Knoten schneller zum Ziel führen könnten. Solche Abschätzungen nennt man auch Heuristiken und man findet sie in vielen Bereichen der Computertechnik. Bei Pfadplanungsalgorithmen ist eine einfache Heuristik beispielsweise die Annahme einer Luftlinie als erste

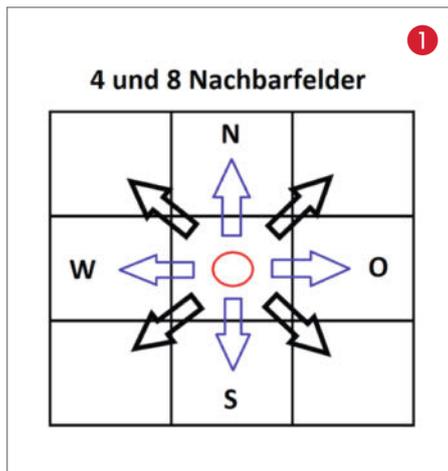
Schätzung für die Entfernung. A\* plant daher in vielen Fällen schneller als der Algorithmus von Dijkstra.

A\* hat jedoch eine Schwäche: Trifft der Roboter auf ein zuvor unbekanntes Hindernis, also eine Unterbrechung einer Kante, wird eine komplette Neuplanung um aktuellen Standort bis zum Ziel durchgeführt. Der D\*-Algorithmus (gelesen *D-Star*) verwendet hingegen Teile der vorherigen Planung weiter und ist somit schneller. Solche Algorithmen nennt man *inkrementelle* Planungsalgorithmen.

Eine oft eingesetzte Weiterentwicklung von D\* ist unser hier eingesetzter Algorithmus



**Pfadplanung:** Die Terasse (rechts) wird in Kacheln aufgeteilt (links). Tisch und Stühle sind die zu umfahrenden Hindernisse.



D\*Lite. Er ist ebenfalls ein inkrementeller, heuristischer Suchalgorithmus, der aber noch effizienter arbeitet und trotzdem leichter zu implementieren ist.

D\*Lite gehört, wie A\* und D\* auch, zu den sogenannten Grid-basierten Verfahren. Dabei wird das Missionsgebiet des Roboters in eine Matrix aus quadratischen Feldern gleicher Größe aufgeteilt. Auf der vorigen Seite rechts unten (Screenshot) sehen Sie unser Missionsgebiet *Terrasse* mit einem fertigen Plan. Jede Kachel stellt quasi einen Knoten dar, der über Kanten mit anderen Knoten verbunden ist. Die Kanten zu den Knoten oben, unten, links und rechts haben alle die gleichen Kosten, was die Implementierung vereinfacht.

Eine genauere Erklärung der Arbeitsweise von Pfadplanungsalgorithmen würde den Rahmen dieses Praxisartikels sprengen. Für diejenigen Leser, die sich den D\*Lite-Algorithmus jedoch im Detail ansehen möchten, haben wir unter dem Link in der Kurzinfo Verweise auf die Arbeiten ihrer Erfinder von Koenig und Likhachev (2002) sowie auf eine Foliensammlung hinzugefügt. Letztere beschreibt D\*Lite an einem Beispiel mit einem Schritt für Schritt-Ablauf, den Sie mit unserem Programm auch ohne Roboter nachstellen können.

### Mission

Unser Missionsgebiet (die Terrasse) ist 10 Felder breit und 7 Felder hoch. Jedes einzelne Feld ist 40cm x 40cm groß. Die Feldgröße orientiert sich an der Größe des Roboters plus einem Spielraum um den Roboter herum. Die Größe des Missionsgebiets richtet sich nach der konkreten Aufgabe. Der Operator, also Sie, bestimmt die Größe jeweils selbst.

Einige der Felder spielen eine spezielle Rolle: Genau ein Feld ist das Startfeld (grün), und genau ein weiteres Feld ist das Zielfeld (rot). Der Operator legt im Programm fest, welche das jeweils sind. Darüber hinaus kann der Operator Felder als bekannte Hindernisse (braun) kennzeichnen. Diese darf der Roboter nicht befahren, da dort eine Kollision droht. Er muss sie umfahren. Im Beispiel ist dies der Bereich, wo die Stühle und der Tisch auf der Terrasse stehen.

Die Pfadplanung hat nun die Aufgabe, einen möglichst kurzen Pfad zwischen Start- und Zielfeld unter Umgehung der bekannten Hindernisse zu planen. Auf der vorherigen Seite ist der geplante Pfad in blauer Farbe zwischen Start- und Zielfeld gekennzeichnet. Die gelben Felder hat sich der Algorithmus für eine mögliche Umplanung vorgemerkt, aber nicht endgültig behandelt, da der kürzeste Pfad in eine andere Richtung verläuft, wie man später sehen wird.

### Nachbarfelder

Ein wichtiger Parameter für die Planung ist die maximale Anzahl der Nachbarfelder, die ein Feld besitzt 1. Es gibt zwei Möglichkeiten: 4 oder 8 Nachbarfelder. Bei 4 Nachbarfeldern kann der Roboter vom aktuellen Feld aus nach Norden (oben), nach Osten (nach rechts), nach Süden (nach unten) oder nach Westen (nach links) fahren. Bei 8 Nachbarfeldern kommen noch die 4 diagonalen Richtungen Nordwest, Nordost, Südost und Südwest dazu. Am Rand der Feldmatrix ist die Anzahl der Nachbarfelder entsprechend kleiner. Das muss die Pfadplanung berücksichtigen.

Unser Programm kann mit beiden Einstellungen (4/8) planen. Der jeweilige Pfad kann dann bei gleichem Start und Ziel sowie gleichen bekannten Hindernissen unterschiedlich sein. Da der *PathRunner* mit Raupenantrieb fährt, ist die Genauigkeit im Zusammenspiel mit dem Untergrund bei Schwenks um 45 Grad begrenzt. Deshalb steuert das Programm den *PathRunner* nur mit Plänen, die auf vier Nachbarfeldern basieren.

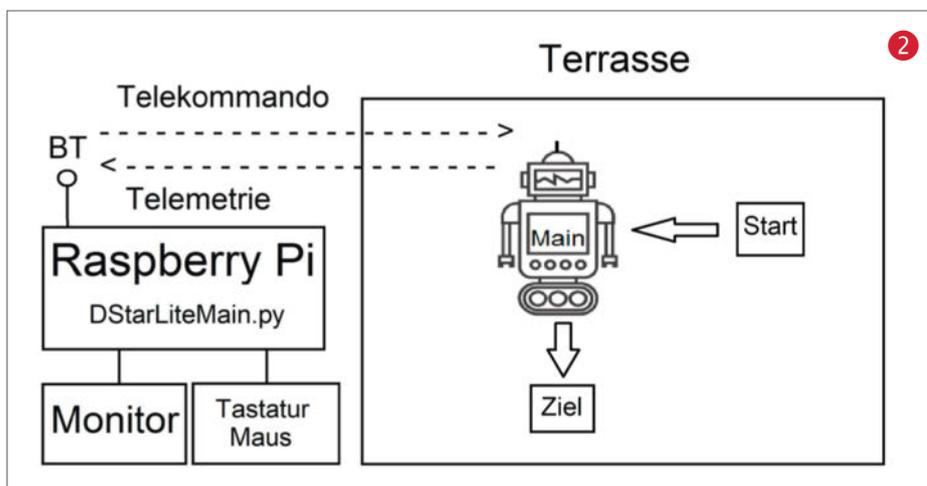
Während der Roboter den vorgeplanten Pfad entlang fährt, können zuvor unbekannte Hindernisse im Weg stehen, etwa ein Haustier. Der Roboter muss diese Situation selbstständig erkennen und eine Umplanung des Pfades zum Ziel einleiten. Unser *PathRunner* nutzt dafür den IR-Sensor, mit dem er größere Hindernisse im nächsten Feld vor ihm erkennt. Der *PathRunner* informiert dann den Raspberry Pi per Bluetooth über das aktuelle Geschehen. Der D\*Lite-Algorithmus erstellt sofort für den restlichen Weg zum Ziel einen neuen Plan.

Wenn Sie sich das dritte Video ansehen, auf dem der *PathRunner* auf der Terrasse fährt, können Sie beobachten, wie der Roboter vor dem ungeplanten Hindernis anhält und kurz darauf wieder losfährt. Der kurze Moment dazwischen ist die Umplanungszeit auf dem Raspberry Pi inklusive Bluetooth-Kommunikation. Sie liegt im Beispiel bei ca. 0,1 – 0,15 Sekunden.

### Systemaufbau

Unser kleines System besteht aus dem Raspberry Pi und dem Roboter *PathRunner* 2. Ein Monitor, eine Tastatur und eine Maus sind am Raspberry Pi angeschlossen.

Auf dem Raspberry Pi läuft das Hauptprogramm *DStarLiteMain.py* für die Pfadplanung. Der *PathRunner* befindet sich in seinem Missionsgebiet und erwartet mit seinem Hauptprogramm *Main* über Bluetooth Telekommandos des Raspberry Pi. Grundsätzlich ist der Auslöser für einen



### Videos

Im GitHub-Repository (siehe Link in der Kurzinfo) sind neben allen Programmen und Anleitungen auch drei Videos enthalten. Sie zeigen, wie ein Operator eine Pfadplanung mit der Software durchführt. Der verwendete Roboter *PathRunner* ist ein leicht modifizierter *Track3r* aus dem Lego-Mindstorms-Baukasten 31313. Er unterscheidet sich nur durch eine Ladeplattform für den Milchreis und durch seine Software vom *Track3r*.

Kommunikationsablauf über die Bluetooth-Schnittstelle immer der Raspberry Pi. Der Roboter reagiert darauf, kann aber selbst keine Telekommandos starten.

### Interactive D\*Lite

Nun schauen wir uns das Programm zur Pfadplanung an, mit dem Sie eigene Pfade planen, simulieren und ausführen können. Nach dem Start des Programms öffnet sich das Hauptfenster mit den drei Reitern *Design*, *Planning* und *Execution* 3.

Der erste Schritt bei der Pfadplanung ist das Design des Missionsgebietes. Die Breite (*Grid width*) und Höhe (*Grid height*) aus 40cm x 40cm großen Feldern stellen Sie über die GUI ein. Im Beispiel sind es 10 mal 7 Felder. Die Einstellung kann zwischen 4 und 11 Feldern je Dimension erfolgen. Sie ist begrenzt, damit das Missionsgebiet nicht zu groß für die Bluetooth-Reichweite wird.

Die Schaltfläche *Recreate grid* generiert dann eine neue leere Feldmatrix und einen leeren Pfadplan. Mit der Maus und dem ausgewählten *Click Mode* (siehe 3) positionieren Sie Start- und Zielfeld (*Goal*) sowie die bekannten Hindernisse (*Obstacle*). Sie können im entsprechenden Modus Start- und Zielfeld auch auf ein freies Feld umsetzen. Hindernisse löscht man, indem man im Modus *Obstacle* erneut anklickt.

Der zweite Schritt ist die Planung auf dem Reiter *Planning* 4. Die Planung stellt drei Modi zur Auswahl: *Fast*, *Slow step* und *Manual step*. Der Modus *Fast* plant ohne Verzögerung. Mit den beiden anderen Modi können Sie D\*Lite bei der Arbeit beobachten. Sie sehen, wie sich die gelben Felder verändern und der Algorithmus sich vorarbeitet. D\*Lite plant übrigens rückwärts vom Ziel zum Start. Der Modus *Slow step* fügt nach jedem Planungsschritt zwei Sekunden Pause ein. *Manual step* hält die Planung nach jedem Schritt an und fragt, wann es weitergehen soll.

Die Option *h=0* legt fest, ob D\*Lite mit oder ohne Heuristik planen soll. Voreingestellt ist die Planung mit Heuristik. Soll D\*Lite ohne Heuristik planen, verlängert sich die Planungszeit deutlich. In unserem Beispiel 4 benötigt D\*Lite mit Heuristik 20 und ohne Heuristik 40 Planungsschritte.

Bei ausgewählter Option *Only direct neighbors* plant D\*Lite mit 4 Nachbarn, ansonsten mit 8. Die Planung starten Sie mit der Schaltfläche *Start planning*. Versperren Hindernisse den Weg zwischen Start und Ziel komplett, kann D\*Lite keinen kürzesten Pfad finden 5.

Nach erfolgreicher Planung wechseln Sie zum Reiter *Execution*, auf dem Sie den Pfadplan ausführen können 6. Er bietet zwei Optionen: *Screen Simulation* und *Lego EV3 Control*. Bei der Simulation fährt ein simulierter Robo-



ter als blauer Pfeil über den geplanten Pfad und führt dabei auch erforderliche Schwenks aus. Die Startausrichtung des Roboters legen Sie vor der Ausführung des Plans mit der Auswahl *Robot start orientation* fest.

Während der simulierte Roboter fährt, können Sie ein ungeplantes Hindernis mit der Maus auf dem Pfad (blau) vor dem Roboter einfügen. Es verursacht eine Umplanung, sobald der Ro-

boter das neue Hindernis erreicht hat. Die Länge des Pfades können Sie im grünen Startfeld ablesen: In unserem Fall ist er 13 Felder lang.

Wählen Sie die Option *Lego EV3 control*, dann steuert das Programm den echten *PathRunner*, sobald die Schaltfläche *Execute plan* geklickt wurde. Achten Sie vorher auf die korrekte Startausrichtung zum nächsten Feld. Der Roboter muss außerdem auf dem Startfeld zu



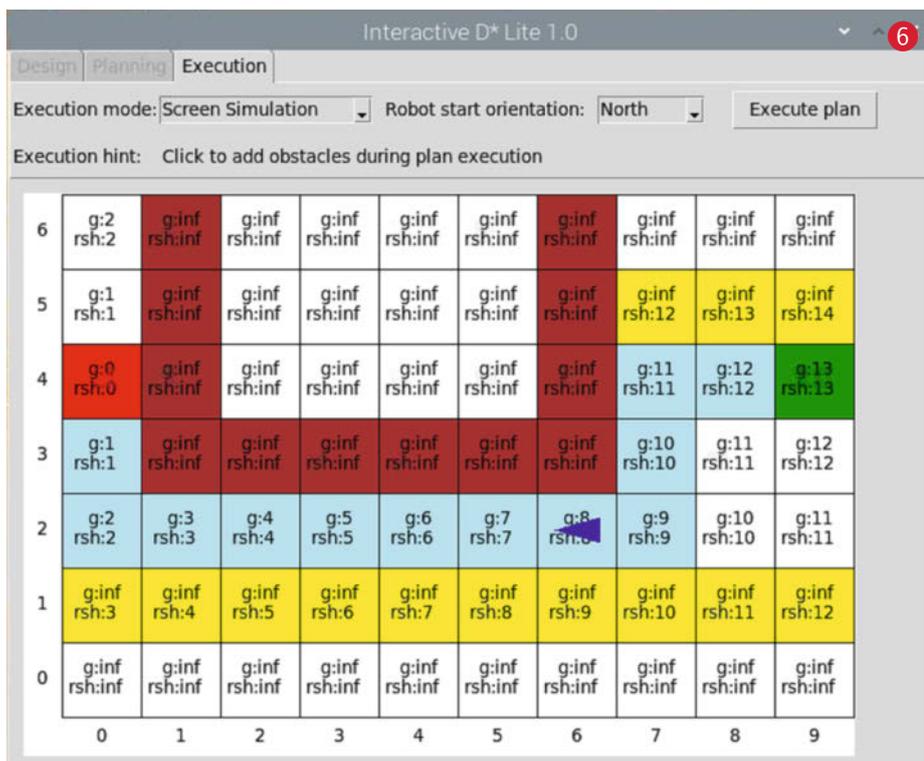


allen Seiten mittig positioniert werden. Möchten Sie am Ende wieder von vorne mit einem neuen Plan starten, so wechseln Sie zum Reiter *Design* und klicken *Recreate grid*.

### Bluetooth-Schnittstelle

Der Raspberry Pi sendet die Telekommandos über die Bluetooth-Schnittstelle ②. Telekom-

mandos (TC) sind kurze Textnachrichten, die dem *PathRunner* Fahrbefehle und weitere Befehle auftragen. Er muss diese in einem vorgegebenen Zeitraum abarbeiten. Der *PathRunner* quittiert eingehende Telekommandos sofort mit einer kurzen Telemetrie-Nachricht (*True* für „verstanden“) und führt sie dann aus. Der Raspberry Pi gibt für die Ausführung einen Zeitraum vor und wartet solange.



Ist das Telekommando abgearbeitet, sendet der *PathRunner* Telemetrie (TM) zur Bestätigung. Beim Servieren des Milchreises im Video sendet der Raspberry Pi insgesamt 24 Telekommandos und empfängt 48 Telemetrie-Nachrichten ⑦.

Nun zu den einzelnen Telekommandos: Das Telekommando *Heartbeat* dient zu Beginn der Ausführung eines Pfadplans dazu, einen erfolgreichen Verbindungsaufbau zu prüfen. Die weiteren Telekommandos unterstützen alle Pfadpläne, die D\*Lite mit 4 Nachbarfeldern pro Feld geplant hat. Wir benötigen 90-Grad-Schwenks nach rechts (*TurnR90*), nach links (*TurnL90*) sowie eine Kehrtwende um 180 Grad (*Turn180*). Weiterhin muss der Roboter ein Feld geradeaus fahren (*Drive*) und anhalten können (*Stop*).

Der Befehl *CheckDistance* prüft einmalig, wenn der Roboter auf dem Startfeld steht, ob dort schon ein Hindernis vor dem Losfahren in Sensorrichtung steht. Je nachdem, wie der Pfad verläuft, kann dann schon eine Umplanung möglich sein.

Hängt der *PathRunner* an seine Telemetrie-Nachricht ein „!“, so ist dies ein Zeichen dafür, dass er ein Hindernis in Blickrichtung des IR-Sensors erkannt hat, beispielsweise *Driveok!*. War das Hindernis vorher noch unbekannt, ist eine Umplanung notwendig.

Wenn Zeitüberschreitungen oder Verbindungsstörungen der Bluetooth-Verbindung auftreten, bricht der Raspberry Pi die Abarbeitung des Pfadplanes ab und zeigt eine Meldung an. Die Bluetooth-Verbindung ist jedoch in der Regel sehr stabil. Der Raspberry Pi sollte aber nicht mehr als 2,5m vom Missionsgebiet entfernt aufgestellt sein.

Technisch basiert die Kommunikation auf einer seriellen Bluetooth-Verbindung, die Lego für die Kommunikation zwischen zwei intelligenten EV3-Steinen implementiert hat. Der Raspberry Pi muss sich daher ebenfalls wie ein EV3-Stein verhalten. Wir nutzen die Mailbox-Funktionalität dieser Schnittstelle, um Nachrichten austauschen. Weitere Details beschreibt der Artikel *Pi steuert EV3 Roboter* im Make-Sonderheft Robotik 2019.

### Programmierung

Die Programmierung auf dem Raspberry Pi erfolgt mit Python 3.7.x und Thonny. Da das Programm etwas größer ist und Erweiterungen später möglich sein sollen, verteilt sich der gesamte Code auf neun Python-Dateien ⑧. Die Implementierung ist objektorientiert und verwendet Python-Klassen (siehe GitHub-Repository).

Eine Übersicht über die Implementierung mit ihren Klassen, Funktionen und Beziehungen zeigt das UML-Klassendiagramm *DStarLite\_Class\_Diagram.pdf*, das auch im GitHub-Repository liegt.

## Roboter-Programmierung

Die Roboter-Programmierung mit der Lego-EV3-Programmierungsumgebung besteht aus der Implementierung der Telekommandos und der Telemetrie 9. Die Projektdatei heißt *PathRunner\_V1.ev3* und liegt im Git-Hub-Repository.

Das Programm beginnt mit der Initialisierung von Variablen im eigenen Block *Init*. Hier müssen Sie die Anzahl der Motordrehungen für die Fahrbefehle anpassen, wenn der Roboter bei den Schwenks von 90 oder 180 Grad merklich abweicht. Im Projekt gibt es das Testprogramm *testTurn*, mit dem Sie prüfen können, ob der Roboter auf Ihrem Untergrund die Schwenks korrekt fährt. Dort können Sie die Einstellung verändern, prüfen und anschließend die Variablen im *init*-Block verändern.

Interessant zu erwähnen ist, dass *D\*Lite* völlig unabhängig von der physischen Größe der Felder plant. Diese legen wir erst hier im *init*-Block in der Variablen *drive* fest. Dazu passend ermitteln wir die Anzahl der Motordrehungen, die nötig ist, um 40cm geradeaus zu fahren. Dabei hilft das Programm *testForward*.

Nachdem alle Variablen initialisiert sind, beginnt die Schleife und wartet auf Telekommandos des Raspberry Pi. Ist ein Telekommando eingetroffen, wird es sofort quittiert und dann im Schalterblock ausgeführt. Der Befehl *Abort* bricht das Programm ab, wird aber zur Zeit nicht verwendet.

In 9 sehen wir den Befehl *TurnR90*. Nach dessen Ausführung sendet das Programm mit dem eigenen Block *SendResult* die Telemetrie-Nachricht. Dort prüft das Programm auch, ob ein Hindernis in Sichtweite ist. Die Sichtweite ist auch im *init*-Block in der Variablen *obstacleDistance* festgelegt. Der Roboter prüft, wenn er sich dem Ende eines Feldes nähert, ob im Folgefeld in Blickrichtung ein Hindernis mit maximal diesem Abstand (IR-Sensor) ist.

## 7 Telekommandos (TC) und Telemetrie (TM)

| Telekommando  | Quittierung | Telemetrie | Fertigmeldung   | max. Zeit (s) |
|---------------|-------------|------------|-----------------|---------------|
| Heartbeat     | ja          | nein       | entfällt        | entfällt      |
| TurnR90       | ja          | ja         | TurnR90ok       | 12            |
| TurnL90       | ja          | ja         | TurnL90ok       | 12            |
| Turn180       | ja          | ja         | Turn180ok       | 12            |
| Drive         | Ja          | ja         | Driveok         | 12            |
| Stop          | ja          | ja         | Stopok          | 12            |
| CheckDistance | ja          | ja         | CheckDistanceok | 12            |

## 8 Python-Quellen und ihre Aufgabe

| Klassen-/Dateiname | Aufgabe  |
|--------------------|--|
| DStarLiteMain.py   | Hauptprogramm zum Starten der Anwendung                                      |
| DStarLiteView      | Anwendungsfenster mit drei Reitern für den Operator                          |
| DStarLitePlanner   | Implementierung des D*Lite Algorithmus in der Funktion <i>mainPlanning()</i> |
| ScreenExecuter     | Planausführung mit simuliertem Roboter                                       |
| EV3_Executer       | Planausführung mit Lego-EV3-Roboter  |
| Vertex             | Feld in der Feldmatrix. Hier ist die Heuristik implementiert.                |
| PriorityQueue      | Sortierte Prioritätsliste mit Feldern für D*Lite                             |
| TMTCpi2EV3         | Senden von Telekommandos und Empfangen von Telemetrie                        |
| EV3mailbox.py      | Kodierung/Dekodierung der Telekommandos und Telemetrie                       |

Da die Messung von den Lichtverhältnissen abhängt, muss sie eventuell angepasst werden. Sie ist im Projekt auf bedeckten Himmel eingestellt. Eine gleichmäßige Ausleuchtung verbessert die Messung.

Die Implementierung der Fahrbefehle hat noch eine Besonderheit. Es kommt oft vor, dass der Pfadplan eine Folge von Feldern enthält, die hintereinander liegen. Würde der Roboter bei jedem Feld anfahren und am Ende eines Feldes wieder anhalten, käme ein ruckeliges Fahrbild zustande.

Um das zu vermeiden, sendet der Roboter bereits nach 80 Prozent der Strecke eines Feldes die Fertigmeldung. Der Raspberry Pi

schickt dann den Folgebefehl. Dieser muss dann aber den Roboter noch die restliche Strecke zu Ende fahren lassen, bevor der eigentliche Folgebefehl bearbeitet wird. In 9 sehen wir bei der Implementierung des *TurnR90*-Befehles den Aufruf des *wait*-Blocks, der dieses Verhalten implementiert.

## Installation und Test

Die Installation des Pfadplanungsprogramms sowie die Einrichtung auf dem Raspberry Pi bestehen im Überblick aus folgenden Schritten. Die einzelnen Details mit Hinweisen beschreibt eine *readme*-Datei im

# 8 JAHRE BASTLER-KNOW-HOW KOMPAKT



### Make Know-how XXL auf 32 GByte USB-Stick

Das komplette Make-Archiv auf einem USB-Stick: in den 47 Ausgaben c't Hacks (2011 bis 2014) und Make (2015 bis 2019) finden Sie Evergreens wie den Teehasen, die Fräse MaXYposi, den DIY-Lötkolben, die Plotclock, Lixie-Anzeigen, die Reißzweckenorgel und viele weitere zeitlose Projekte zum Nachbauen.

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

99,- €

NEU + portofrei



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

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

GitHub-Repository im Verzeichnis *20\_Raspberry\_Pi\_project*.

Nachdem Sie sich mit dem Benutzer *pi* angemeldet haben, installieren Sie den Bluetooth-Manager. Er wird benötigt, um eine serielle Bluetooth-Schnittstelle zwischen dem Raspberry Pi und dem Lego-EV3-Roboter aufzubauen. Der EV3-Roboter sieht den Raspberry Pi unter dem Namen *pi4robo*, den Sie auf dem Raspberry Pi anlegen. Danach booten Sie den Raspberry Pi und es erscheint ein weiteres Bluetooth-Icon in der Startleiste für den Bluetooth-Manager, das Sie aber ignorieren können.

Nun erfolgt das einmalige Bluetooth-Pairing von Raspberry Pi und Lego EV3 Roboter. Achten Sie hierbei besonders auf die Hinweise in unserer *readme*-Datei. Nach erfolgreichem Pairing muss jetzt nur noch eine serielle Bluetooth-Verbindung eingerichtet werden. Diese müssen Sie nach jedem Booten des Raspberry Pi bzw. des Lego-EV3-Roboters neu einrichten, was aber sehr schnell geht. Die Bluetooth-Verbindung ist erfolgreich eingerichtet, wenn auf dem Display des EV3-Roboters oben links ein BT-Icon und daneben *<>* erscheint.

Zum Abschluss der Installation müssen Sie das GitHub-Repository auf den Raspberry Pi und auf Ihren PC herunterladen und entpacken. Laden Sie das Projekt *PathRunner\_V1.ev3* auf den Roboter und starten Sie das Programm *main*. Auf dem Raspberry Pi starten Sie nun *DStarLiteMain.py*. Zum Test der Installation können Sie mit unserem Programm einen kurzen Pfad planen und ausführen, der über 3-4 Felder geht.

Die neun Python-Quellen des Pfadplanungsprogramms laufen auch unter Windows 10 mit Python. Sie können Pfade planen und simulieren, jedoch keinen Lego-EV3-Roboter steuern. Versuchen Sie dies, erscheint ein entsprechender Hinweis.

### Erweiterungsmöglichkeiten

Um Schwenks genauer und unabhängig vom Untergrund zu fahren, macht der Einsatz eines Gyro-Sensors Sinn. Er misst, um wieviel Grad der Roboter sich gedreht hat.

Dazu müssen die Schwenks *TurnR90*, *TurnL90* und *Turn180* entsprechend im Lego-EV3-Programm angepasst werden.

Mit unserem Pfadplanungsprogramm können Sie beliebige andere Lego EV3-Roboter mit Raupenantrieb steuern. Bedenken Sie, dass die Motoren des *PathRunner* rückwärts drehen, wenn er vorwärts fährt. Ist dies bei Ihrem Roboter nicht so, verwenden Sie für beide Motoren den *Invert Motor Block*. Tipps zum Steuern von Nicht-Lego-Robotern finden Sie im Kasten „Hinweis für Fortgeschrittene“.

—*dab*

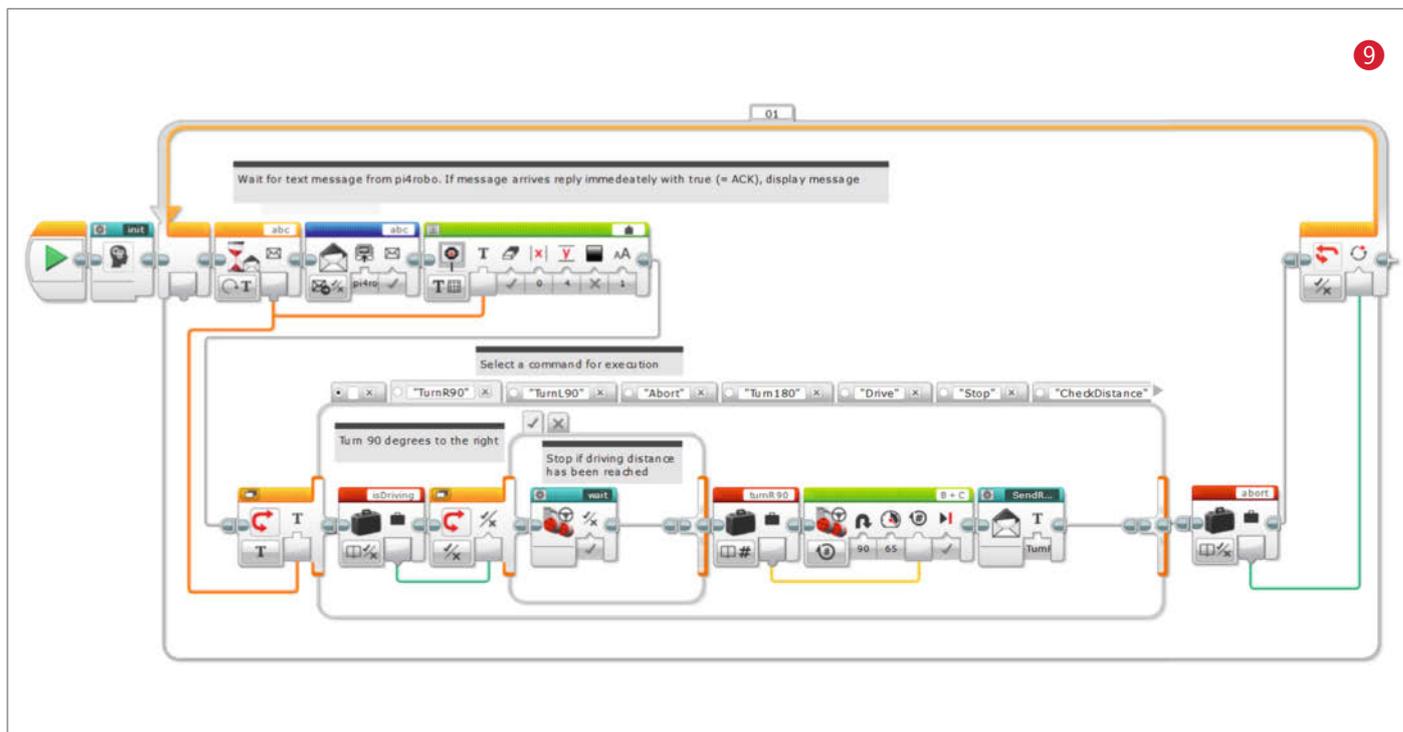
### Hinweis für Fortgeschrittene

Das Pfadplanungsprogramm kann in Projekten mit anderer Roboter-Hardware verwendet werden. Die Implementierung ist dafür bereits vorbereitet. Dazu müssen Sie eine neue auf Ihre Roboter-Anbindung angepasste *Executer*-Klasse implementieren. Sie muss von der Klasse *ScreenExecuter* erben, genauso wie die Klasse *EV3\_Executer*. Letztere kann Ihnen als Muster dienen.

Implementieren Sie in der neuen Klasse neben der *\_\_init\_\_*-Funktion die Funktionen für Ihren speziellen Roboter, die in der Klasse *EV3\_Executer* mit *Over-*

written im Kommentar gekennzeichnet sind. Diese werden vom *ScreenExecuter* aus seiner Funktion *executePlan()* gerufen. Ihr Roboter ist dann in die Steuerung eingebettet.

In der grafischen Oberfläche ergänzen Sie einen Eintrag für Ihren Roboter auf dem Reiter *Execution* in der Combobox *cbExecMode*. Sie werten die Auswahl des Benutzers in der Funktion *DStarLitePlanner.executePlan()* aus und erzeugen Ihren *Executer*. Nun ist Ihr spezieller Roboter bereit, sein Missionsgebiet zu erkunden.



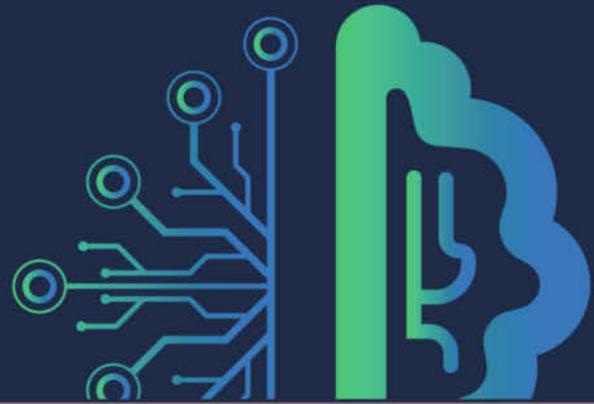


# Workshops 2021

## KÜNSTLICHE INTELLIGENZ

Deep Learning mit Tensorflow

23. – 26. Februar 2021, online



## Parallele Programmierung in Java:

Effiziente Software für Multi-Core Systeme

01. – 02. März 2021, online



## Cybersicherheit:

Aktuelle Angriffstechniken und ihre Abwehr

25. – 26. Februar 2021, online



## Datenanalyse mit Python – Einstiegskurs:

Arbeiten mit NumPy und Pandas, Visualisierung, Data Literacy

02. – 03. März 2021, online



## Continuous Integration mit Jenkins

02. – 03. März 2021, online



Weitere Infos unter:  
<http://www.heise-events.de/workshops>



# Modellbau-Synthesizer

Ein altes Tastenpult für eine Modellbahn – samt seiner Originalverpackung – ist die Basis für diesen schicken, lichtsensitiven Retro-Synthesizer.

von Helga Hansen



Bilder: Marc

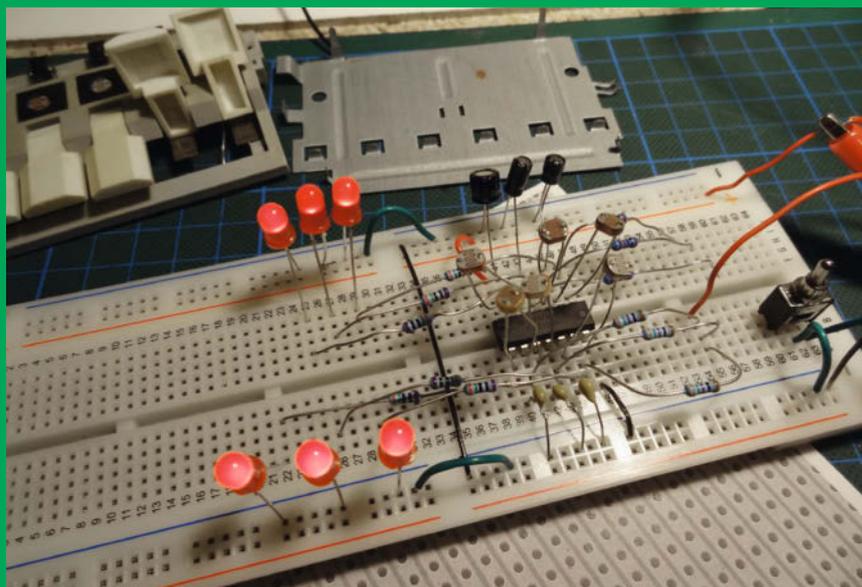
Zugegeben – die Töne, die aus diesem Kästchen kommen, sind eher schrammelig. Dafür überzeugt der Modellbau-Synthesizer mit seinem originellen Aussehen und der cleveren Verwendung alter Technik. Denn seine sechs Tasten gehören zu einem Controller, mit dem ursprünglich Modellbahnen des ostdeutschen Herstellers *Berliner TT-Bahnen* gesteuert wurden. Mit Fotowiderständen, LEDs und einer Oszillatorschaltung hat der Berliner Bastler Marc daraus ein interaktives Instrument gebaut, das gerade so in den Originalkarton passt.

Grundlage des Synthesizers ist ein kleiner Schaltkreis mit einem Hex-Schmitt-Trigger, den wir in Make 1/18, ab S. 48 ausführlich vorgestellt haben. Der Chip enthält sechs identische Schaltungen, mit denen sechs unterschiedliche Töne erzeugt werden können, und passt damit perfekt zum Tastenpult. Er verträgt praktischerweise zwischen 3 und 18 Volt und liefert immer definierte Ausgangssignale: Rechteckschwingungen, eine der Grundformen der Klangerzeugung. Die Frequenz der Schwingungen wird von den verbauten Kondensatoren und Widerständen bestimmt. Mit dem Umlegen der Tasten des Pults lässt sich einstellen, ob Strom fließt oder nicht. Die Tonhöhe (respektive Frequenz) wird anschließend über die Fotowiderstände geändert, die auf Helligkeitsunterschiede reagieren. Sechs LEDs zeigen zusätzlich visuell an, welche Töne gerade erklingen.

Die Schaltung ist „normal“ auf einer kleinen Lochrasterplatine verlötet, aber um das komplette Projekt in das kleine Kistchen zu stopfen, musste Marc beim Einbau kreativ werden. Die Kabel hat er so weit gekürzt, dass sie nur noch verbunden werden konnten, als der Deckel bereits halb geschlossen war. Damit die Kontaktstellen nicht zu Kurzschlüssen führen, hat er sie mit Flüssiggummi (siehe Seite 126) isoliert. Schrumpfschläuche zu erhitzen war in einer leicht entzündlichen Pappkiste keine Option. Schließlich passt auch der 9-Volt-Block als Spannungsversorgung haargenau in die Kiste. Das Tastenpult ist mit vier Schrauben fest auf dem Deckel angebracht.

Besonders effektvolle Musik entlockt man dem Modellbau-Synthi, wenn man zum Spielen schwarze Handschuhe trägt und eine Taschenlampe oder ein blinkendes Fahrradlicht bereit hält. Nötig ist noch ein Mini-Verstärker, trotzdem bleibt das Projekt leicht transportierbar. Viele Bilder, weitere Erläuterungen und natürlich eine akustische Vorführung hat Marc auf seine Webseite gestellt. Da Deutsch nicht seine Muttersprache ist, freut er sich außerdem über Hilfe, um die Dokumentation weiter zu verbessern. —hch

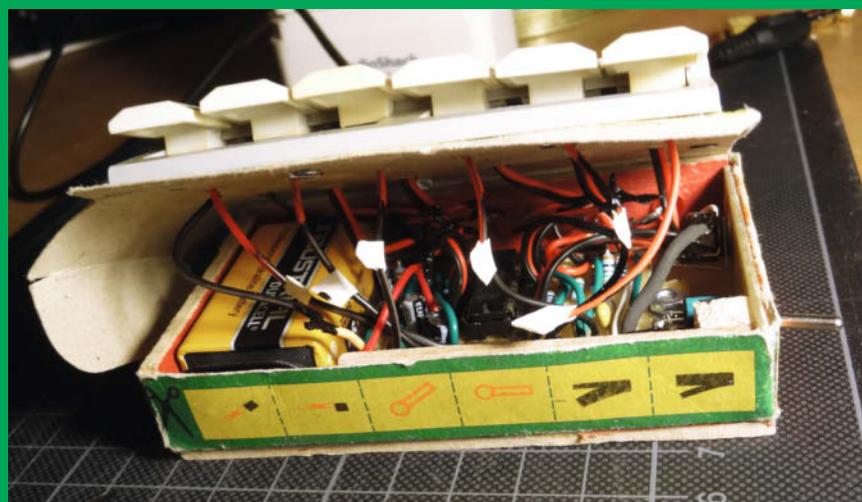
► [genusitineris.net/musik/modellbahn-synthesizer](https://genusitineris.net/musik/modellbahn-synthesizer)



Testaufbau der Schaltung auf einem Breadboard



Originalkarton mit Lochrasterplatine und Batterie



Das weiße Tape kennzeichnet die sechs Tastenanschlüsse.

# OSSSO bringt Ordnung in Werkstätten

Mehr Zeit zum Basteln und Bauen, statt ständig verlorene Dinge im Makerspace zu suchen und zu putzen – das Open-Source-System OSSSO kann helfen.

von Helga Hansen



Bilder: OSSSO / Hobbyhimmel

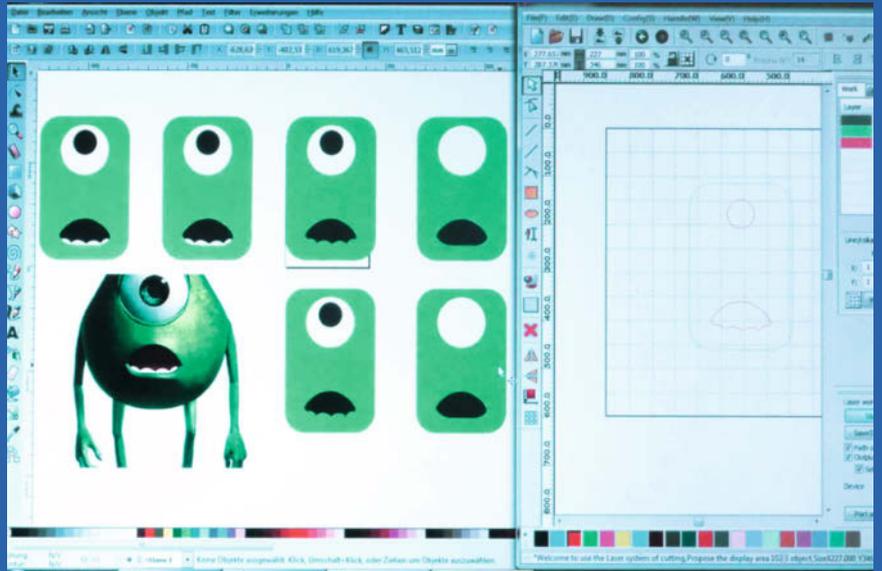
**S**chnell was sägen wollen, aber die Säge nicht finden und auf der Suche noch mal eben die seit Wochen herumfliegenden Holzspäne aufsaugen – wer sich regelmäßig in einer offenen Werkstatt oder Hackspace aufhält, kennt vermutlich das Problem. Statt immer nur die gleiche Arbeit zu erledigen, müsste es doch möglich sein, die Systeme dahinter besser zu gestalten. Das dachte man sich zumindest im *Hobbyhimmel Stuttgart* und hat OSSSO entwickelt. Die Abkürzung steht recht selbsterklärend für **O**pen **S**ource **S**ystem für **S**auberkeit und **O**rdnung für **O**ffene Werkstätten.

Dahinter verstecken sich sechs Teilprojekte: osSHELF, osVAC, osCLEAN, osBOX, osTRAY und osWALL, die jeweils ein Werkstattproblem angehen und unterschiedlich weit umgesetzt sind. So wurde unter osVAC das Staubsaugerkonzept des Hobbyhimmels überdacht, das zuvor dank vieler verschiedener Staubsauger mit inkompatiblen Stutzen und Schlauchanschlüssen und hohem Beutelverbrauch eher für Frust statt saubere Arbeitsplätze sorgte. Nun gibt es ein Kupplungssystem mit Fokus auf den D32-Standardschlauch und einen Werkstattsauger mit Zyklonabscheider. Für die zahlreichen Adapter, die die vorhandenen Staubsaugerteile passend kombinieren, gibt es die Druckvorlagen auf der 3D-Druckplattform Thingiverse zum Download. Weiterentwicklungen sind dabei ausdrücklich erwünscht.

Bereits praxiserprobt ist auch osCLEAN, das die Putz- und Reinigungsmittel ordentlich zusammenführt. In den Werkstatträumen hängen nun Putzwände, an denen die jeweils benötigten Geräte wie Handfeger und Stofflumpen platzsparend aufbewahrt werden. Nach einem ähnlichen Prinzip soll auch osWALL funktionieren. Da die Corona-Pandemie derzeit das gemeinsame Arbeiten erschwert, ist das Projekt aber noch am Anfang. Werkstattwände mit vorbereiteten Aufhängungen gibt es schon einige, allerdings sieht man in Stuttgart noch Verbesserungspotential bei den Möglichkeiten zum Um- und Nachbau. Manche Werkzeuge seien schwer ihrem Platz zuzuordnen und beim Einzug von neuen Geräten sei die Umorganisation oft langwierig.

Da das Projekt Open Source ist, freut man sich im Hobbyhimmel über alle Interessierten, die die Vorlagen weiterentwickeln oder eigene Ideen umsetzen wollen und ihre Varianten auf Thingiverse dazu laden. Um den Einstieg in die Werkstatt möglichst niedrigschwellig zu gestalten, ergänzen seit kurzem Kurzeinführungen auf YouTube das Angebot. Dort werden Gerätschaften wie Sägen, Schleif- und Bohrmaschinen vorgestellt und erklärt, worauf es bei der Benutzung zu achten gilt. Die Videos dürfen explizit von anderen Werkstätten genutzt werden, ersetzen aber nicht die Einführung am jeweiligen Gerät. —hch

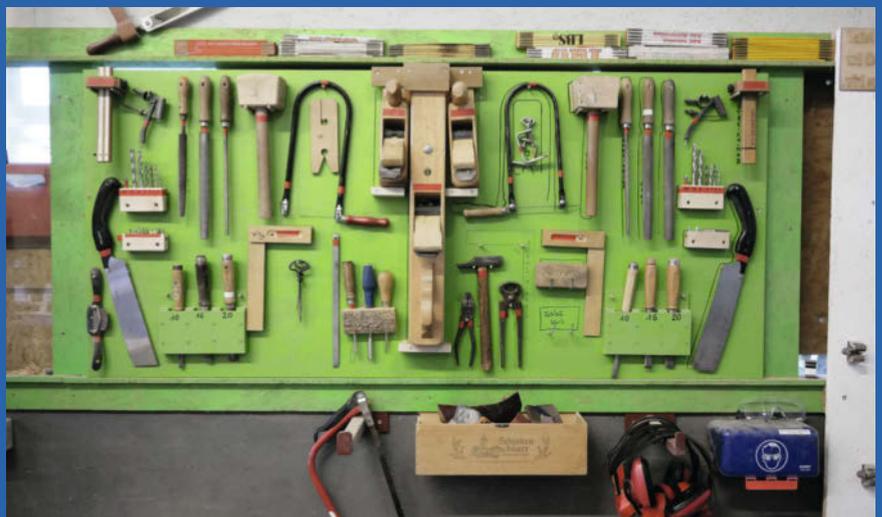
► [osso.de](http://osso.de)



Der einäugige Mike aus dem Film *Monster AG* diente als Vorbild für die Tücherbox.



Die unterschiedlichen Staubsaugeranschlüsse werden mit den osVAC-Adaptern durch ein einheitliches System rund um den DN32-Standardschlauch ersetzt.



Mit osWALL soll später noch ein offenes System für Werkstattwände entstehen.

# FabAccess

Mit FabAccess sollen Makerspaces bald ihre Maschinen und Nutzer verwalten können – als föderiertes System sogar über einzelne Werkstätten hinaus.

von Helga Hansen



Prototyp des An-/Ausschalters für Maschinen

Als Mitglied vom Makerspace A einfach Zugang zu den Maschinen von Fablab B und Werkstatt C haben und am Ende nur eine Rechnung zahlen: Mit dem Open-Source-Verwaltungssystem FabAccess könnte dieses Szenario in Zukunft Wirklichkeit werden. Die Automatisierungsinfrastruktur soll über einzelne Werkstätten hinaus die Nutzung von Räumen und Maschinen einfacher machen. Einen Zwischenstand ihrer Arbeit stellte das Projekt FabAccess auf dem virtuellen Congress des Chaos Communication Clubs (rc3) im Dezember vor.

Die Idee zu FabAccess entstand aus dem in Fablabs notwendigen Kompromiss zwischen Sicherheit beim Arbeiten mit gefährlichen Maschinen und der Vorgabe, dort möglichst eigenständig lernen und tüfteln zu können. Mit dem System sollen außerdem Angestellte entlastet werden, um mehr Zeit für ihre eigentlichen Aufgaben zu haben, statt ständig in den Werkstätten Aufsicht führen zu müssen. Konkret geplant sind dafür vier Komponenten: eine Maschinenverwaltung, ein Berechtigungssystem und die Nutzerverwaltung sowie eine Möglichkeit zum werkstattübergreifenden Betrieb als Föderation.

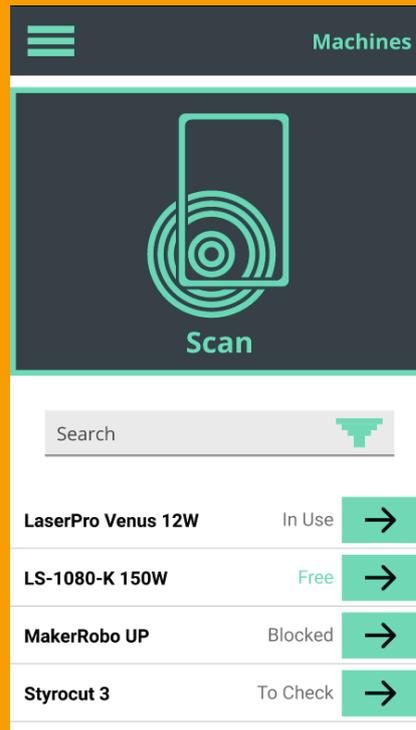
Dabei werden die Maschinen zunächst über ihren Stromanschluss an- und ausgestellt – insbesondere, wenn zum sicheren Betrieb eine Einweisung nötig ist. Für die Stromtrennung können Zwischenstecker wie der Shelly Plug genutzt werden und Systeme, die mit der ESP8266-Firmware Tasmota geflasht werden. Umgesetzt werden sollen auch eine Reservierungsfunktion und eine Abnahme nach der Benutzung. Letzteres führe in der Praxis zu deutlich saubereren Maschinen, so das Projektteam. Gespeichert wird daher, wer zuletzt ein Gerät benutzt hat, um bei Problemen oder Ausfällen Rücksprache halten zu können. Ein rollenbasiertes Zugriffssystem soll bei der Kontrolle helfen, wer an einer Maschine bereits eingewiesen wurde. Die Einbindung von einfachen Maschinen ohne zwingende Einweisung ist aber ebenfalls vorgesehen. Ein SmartCard-System aus FabCard und einem FabReader hilft schließlich bei der Verwaltung der Nutzenden und der Anbindung an die Maschinen.

Am Ende soll die Software jeweils von den Spaces selbst gehostet werden, aber über die einzelnen Instanzen hinweg als föderiertes System miteinander verbunden sein und Daten austauschen. Nutzerinnen und Nutzer könnten dann mit wenig Aufwand verschiedene Werkstätten nutzen. In der letzten Entwicklungsstufe sollen auch übergreifende Abrechnungsmöglichkeiten hinzukommen. Gleichzeitig kann das FabAccess-System mit eigenen Skripten erweitert und angepasst werden. Die bereits erstellten Anwendungen liegen auf Gitlab bereit und neue Interessierte, die sich an der Umsetzung beteiligen wollen, sind gerne gesehen. —hch

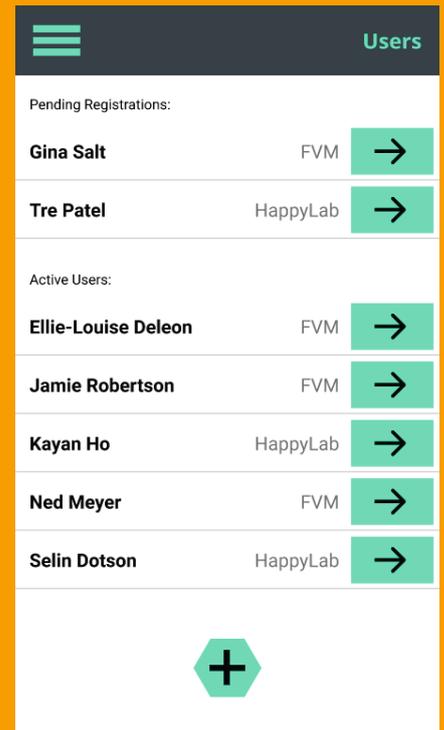
► [fab-access.org](https://fab-access.org)



Aktuelle Prototypen des FabReaders



In der FabAccess-Software soll der Status der Maschinen abrufbar sein...



... und auch die (hier noch fiktiven) Mitglieder verwaltet werden.



Ein erster Prototyp des Lesegeräts

# Docker für Raspberry Pi

Mit Docker können Sie verschiedene Anwendungen auf dem Pi (und Ihrem PC) in Containern isoliert nebeneinander laufen lassen, ohne dass sich deren Softwareabhängigkeiten, Versionsstände und Dienste ins Gehege kommen. Geht mal bei der Installation oder Konfiguration was schief, setzt man einen Container in Minuten neu auf, ohne die Stabilität des Gesamtsystems zu gefährden. Unsere Einführung führt Sie schnell und einfach zum ersten Erfolgserlebnis.

von Daniel Bachfeld

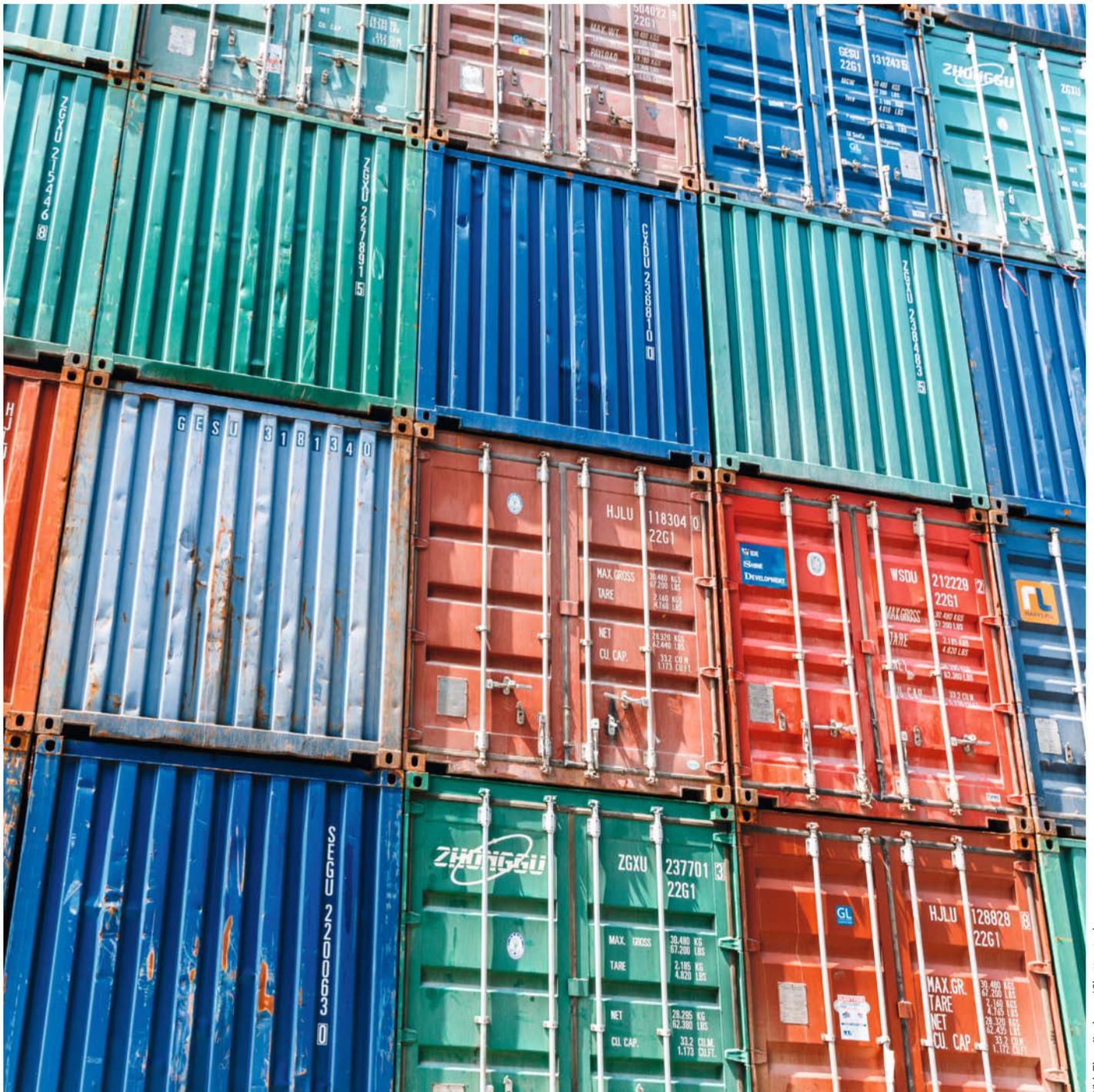


Bild: Zhao Jiankang / Shutterstock.com

Solange man Anwendungen für den Raspberry Pi nur aus dem offiziellen Repository der Pi-Foundation installiert, treten selten Probleme auf. Benötigt man aber mal eine aktuellere Version einer Software oder ist die Software gar nicht im Repo für den Pi vorhanden, treten mitunter Konflikte beim Installieren auf: Statt der bereits vorhandenen Version einer Bibliothek erwartet die neue Software beispielsweise eine höhere Versionsnummer. Die könnte man zwar auch selber übersetzen und anschließend beten, dass das keine weiteren Konflikte nach sich zieht. In der Praxis passiert aber oft genau das. Und manchmal ist das System dann so zerrödelt, dass es sich nur noch schwer restaurieren lässt. Mit der freien und quelloffenen Lösung *Docker* gehören solche Desaster der Vergangenheit an.

Mit Docker isolieren Sie Anwendungen, Server und, wenn Sie wollen, ganze Entwicklungsumgebungen in sogenannten Containern. Alle für eine Anwendung notwendigen Bibliotheken, Programme, Tools und Dienste laufen ebenfalls in diesem Container und sind perfekt darauf abgestimmt. Docker ist zudem die perfekte Lösung, wenn man mehrere Server parallel auf einem Pi laufen lassen will, etwa Nginx, Node-RED, MQTT und weitere.

Im Unterschied zur Virtualisierung werden einem Container weder dedizierter Speicher noch einzelne Kerne auf einer CPU zugewiesen. Vielmehr nutzen Container über den Docker-Dienst die Ressourcen des sie umgebenden Hosts direkt und effizient. In einem Container arbeiten aber dennoch sämtliche Anwendungen, sprich Prozesse, isoliert von anderen Containern und vom Gesamtsystem.

Die Container nutzen über den Docker-Dienst den Kernel des Hostsystems (weshalb sie auch in dessen Prozessliste auftauchen). Für Kernelexperimente ist Docker deshalb nicht geeignet, hier muss man Lösungen wie *VirtualBox* einsetzen – was mittlerweile auf dem Pi sogar funktioniert. Anwendungen mit einer grafischen Oberfläche lassen sich unter Docker mit einfachen Tricks auf einem Desktop darstellen, etwa über VNC oder X11-Unix-Sockets, dazu später mehr.

Im einfachsten Fall verfrachtet man eine einzige Anwendung mit ihren erforderlichen Bibliotheken in einen Container. Häufig betreibt man auch Webserver mit einer Datenbank in Containern. Und sogar ein Ubuntu-Desktop lässt sich mit Docker betreiben – auf einem Raspi.

### Isoliertes Innenleben

Ein Container stellt eine Linux-Betriebsumgebung für eine Anwendung bereit. Das sind neben Sammlungen von Bibliotheken, Dienstprogrammen und Daten auch ein eigenes, dediziertes Dateisystem. Die Geschmacksrichtung dieser Umgebung kann man durch die

## Kurzinfo

- » Docker verstehen und installieren
- » Eigene Container erstellen und starten
- » Images bauen

### Checkliste



**Zeitaufwand:**  
1 Stunde



**Kosten:**  
keine (bei vorhandenem Raspi)

### Material

» Raspberry Pi Version 3 oder 4

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

Auswahl eines sogenannten *Images* selbst wählen. Das Image kann man sich wie eine Küche aus dem Katalog vorstellen: Man kann zwischen verschiedenen Herstellern (Linux-Distribution), Designs (Version der Distribution) und Ausbaustufen wählen. Die gelieferte Küche ist bei jedem Kunden die gleiche, aber in seinen Räumen montiert und mit Geschirr in den Schränken wird daraus ein individueller Container.

Zur Verinnerlichung: Das Image ist die Grundlage eines jeden Containers, quasi die Blaupause. Das im Container laufende Linux kann eine völlig andere Distribution sein, als auf dem Host, auf dem Docker läuft. Anders als bei echten Küchen (es sei denn, Sie haben viel Geld) können Sie aus dem digitalen Image beliebig viele Container erzeugen, ausstatten, starten, stoppen und bei Nichtgefallen einfach wieder löschen.

Es gibt fertige Basis-Images und Parent-Images, die beispielsweise ein Minimal-Linux mit den allerwichtigsten Tools und Libs bereitstellen und die man selbst beliebig erweitern kann. Diese Images stellt beispielsweise die Seite Docker Hub zum Download bereit (siehe Link in der Kurzinfo). Auf dieser Grundlage hat die Community viele weitere Images entwickelt, die speziellere Aufgaben erledigen und Dienste anbieten, beispielsweise OwnCloud, Pi-Hole, GitLab, Node-RED, MQTT, Nginx, MySQL, Home Assistant und viele mehr. Praktisch daran ist, dass viele Punkte der offiziellen Installationsanleitungen der Softwarelösungen oftmals schon erledigt sind, sodass man schnell mit ersten Versuchen für das eigene Projekt anfangen kann.

Grundsätzlich läuft Docker auf den PC-Architekturen x86, x64, arm32v6, armv32v7 sowie arm64v8 und mit der kostenpflichtigen Enterprise-Version sogar auf IBMs S/390. Aber nicht jedes auf Docker Hub verfügbare Image ist auch für jedes System geeignet. Gerade für die

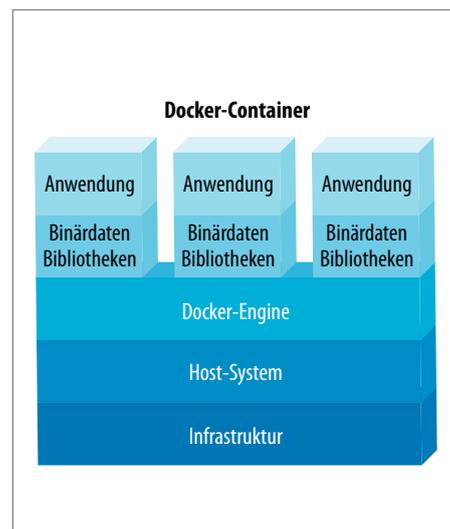
ARM-Plattform des Raspberry Pi sind viele fertige Images nicht verfügbar, sodass man sich passende Image oft selber bauen muss, auch dazu später mehr.

### Installiert

Docker ist mit wenigen Schritten in der Shell auf dem Pi installiert und gestartet:

```
sudo apt update
sudo apt upgrade
sudo curl -sSL https://get.docker.com | sh
```

Der Befehl lädt ein Skript und startet die Installation. Das Skript lädt die zum System passenden Docker-Dateien herunter, je nachdem, ob auf dem Pi die 64-Bit- oder die 32-Bit-Version von Raspbian läuft. Ist das erledigt, muss man den User *pi* in die Gruppe *docker* aufnehmen:



So sind Docker-Container aufgebaut.

## Die wichtigsten Befehle

| Befehl  | Funktion                              |
|---|---------------------------------------|
| <code>docker run &lt;Imagename&gt;</code>               | Image laden und Container starten     |
| <code>docker pull &lt;Imagename&gt;</code>              | Image laden                           |
| <code>docker ps -a</code>                               | Alle Container zeigen                 |
| <code>docker image ls</code>                            | Alle Images zeigen                    |
| <code>docker rm &lt;Containername&gt;</code>            | Container löschen                     |
| <code>docker exec -it &lt;Containername&gt; bash</code> | Interaktive Bash im Container starten |
| <code>docker attach &lt;Containername&gt;</code>        | Mit Prozess im Container verbinden    |
| <code>docker start &lt;Containername&gt;</code>         | Container starten                     |
| <code>docker stop &lt;Containername&gt;</code>          | Container stoppen                     |

```
sudo usermod -aG docker pi
```

Anschließend meldet man sich einmal vom Desktop ab und wieder an oder man bootet den Pi neu. Fortan wird der Docker-Dienst automatisch im Hintergrund gestartet und der Nutzer *pi* kann ihm über den mitinstallierten Docker-Client Anweisungen geben. Der Docker-Client ist ein textbasiertes Tool für die Kommandozeile respektive Shell. Im Folgenden zeigen wir die wichtigsten Befehle für die Praxis.

## Ausprobiert

Als ersten Test gibt man in der Shell den Befehl

```
docker run hello-world
```

ein. Er lädt ein Test-Image von Docker Hub herunter, erzeugt (instanziert) daraus einen Container und startet ihn. Das Minimalst-Image macht nichts anderes, als einen Text in der Shell auszugeben, dass alles funktioniert.

Für den nächsten Schritt nehmen wir mal an, Sie wollten schon immer rein interessehalber in der eher selten benutzten Sprache Erlang (im Tiobe-Index irgendwo hinter Platz 50) auf dem Pi ihre ersten Gehversuche machen – und dann ausgerechnet in der aktu-

ellsten Version, die es gibt, nämlich 23.2. Im offiziellen Raspbian-Repo liegt allerdings nur 22.2. Statt nun die Sources runterzuladen, die Anleitung zum Übersetzen und Installieren zu studieren, weitere Tools und Libs zu installieren und dann manuell zu übersetzen, laden Sie einfach das Docker-Image auf den Pi. Denn glücklicherweise gibt es Docker-Fans in der Erlang-Community, die die aktuellste Version schon für Sie zusammengebaut und auf Docker Hub zur Verfügung gestellt haben. Der Befehl

```
docker run -it erlang
```

lädt das aktuellste Image herunter und startet anschließend einen Container interaktiv (-i) in einer Shell respektive Terminal (-t). Dort erscheint das Prompt des Erlang-Interpreters (>) – probieren Sie mal *20+20*. einzugeben. Es sollte 40 rauskommen. Mit *STRG+c* und *a + Enter* kommen Sie wieder aus dem Interpreter heraus. Gleichzeitig wird der Container gestoppt.

## Start und Stop

Wollen Sie den Container wieder starten, geben Sie `docker start <name>` ein, wobei Sie leider den richtigen Namen des Containers

erst noch herausfinden müssen. Denn *erlang* ist ja nur der Name des Images! Woher hat der Container also einen Namen? Docker selbst vergibt zufällig gewählte Namen, die man durch

```
docker container ls -a
```

oder

```
docker ps -a
```

heraus bekommt. Sie bestehen immer aus einem Adjektiv und dem Nachnamen einer mehr oder minder bekannten Person.

Die Befehle zeigen alle angelegten Container, auch wenn diese gerade nicht in Betrieb sind. Ohne die Option *-a* (all) würden in der Liste nur laufende Container zu sehen sein. Neben dem Container-Namen (letzte Spalte) zeigt die Liste auch die jeweils zugehörigen Images, aus denen die Container erzeugt wurden. In unserem Fall hat der Container den Namen *loving\_curie* erhalten, bei Ihnen heißt er sicherlich anders, aber ähnlich schräg.

Hätte man beim ersten Start statt

```
docker run -it erlang
```

besser

```
docker run --name erlangtest -it erlang
```

angegeben, hieße der Container nun *erlangtest*.

Sie können diesen erweiterten Befehl ruhig nachholen. Docker instanziiert dann einen zweiten Container mit dem gleichen Inhalt, nur mit anderem Namen. Das geht diesmal erheblich fixer als beim ersten Mal, weil das Image zum Erstellen des Containers ja bereits auf der Festplatte liegt. Den Container mit dem seltsamen Namen können Sie einfach mit

```
docker container rm seltsamer_name
```

löschen.

Zum Starten des neuen (oder alten) Containers geben wir diesmal den selbst vergebenen Namen an:

```
docker start erlangtest
```

Der Befehl führt diesmal allerdings nicht zur Anzeige der Erlang-Shell, sondern endet in der normalen Pi-Shell. Der Docker-Container läuft nun im Hintergrund und wir müssen uns erst mit dem Interpreter verbinden:

```
docker attach erlangtest
```

Und schon ist das Prompt wieder da. Alternativ hätte man sich beim Start mit der zusätzlichen Option *-i* gleich verbinden können:

```
docker start -i erlangtest
```

Beide genannten Befehle verbinden das Terminal mit dem zuletzt gestarteten Prozess innerhalb des Containers.

```
pi@raspberrypi:~$ docker run -it erlang
Unable to find image 'erlang:latest' locally
latest: Pulling from library/erlang
6e5587ff5efa: Pull complete
439dbbb05ea0: Pull complete
3b89c8b4e5b2: Pull complete
4a53f70a43c3: Pull complete
0cd9fb1f233c: Pull complete
3f107a2c58e1: Pull complete
42c7d0781074: Pull complete
2ad6c11d1628: Pull complete
Digest: sha256:4afe82096a3121185201e363184dcbe01e51efa62992fea7e25a4f6bd2cf67c4
Status: Downloaded newer image for erlang:latest
Erlang/OTP 23 [erts-11.1.5] [source] [64-bit] [smp:4:4] [ds:4:4:10] [async-threads:1]

Eshell V11.1.5 (abort with ^G)
1> date()
1> .
{2021,1,24}
2> 1+2.
3
3>
BREAK: (a)abort (A)bort with dump (c)ontinue (p)roc info (i)nfo
(1)oaded (v)ersion (k)ill (D)b-tables (d)istribution
a
pi@raspberrypi:~$
```

Erlang in a box: Docker lädt das Image, entpackt es und startet einen interaktiven Container.

```

pi@raspberrypi:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS          PORTS                    NAMES
a583501f8b0e   erlang        "erl"                   About an hour ago   Exited (0)    About an hour ago      lucid_pare
ff9c54d39637   md2web       "./startserver.sh"     23 hours ago     Exited (137)  7 hours ago           webserver
0ec3d170162a   portainer/portainer-ce "/portainer"         45 hours ago     Up 7 hours    8000/tcp, 0.0.0.0:9000->9000/tcp  portainer
f24edf8fc9ce   ubuntu-vnc   "/entrypoint.sh"      2 days ago       Exited (137)  7 hours ago           ubuntu-lked
5eacae970a64   ros-melodic-vnc "/entrypoint.sh"      3 days ago       Up 6 hours
f7f13a1052c1   homeassistant/raspberrypi4 "/init"              3 days ago       Exited (129)  7 hours ago           ros-master
06302b4feae0   ubuntu      "/bin/bash"           4 days ago       Exited (0)    47 hours ago           home-assistant
bold_brown
    
```

Die Ausgabe von `docker ps` zeigt alle Container und ihren Status.

```

pi@raspberrypi:~$ docker image ls
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
md2web              latest      3f357b16f27b    25 hours ago    466MB
ubuntu-vnc         latest      566ac05f4887    2 days ago      1.13GB
ros-melodic-vnc    latest      02dbcfd36503    3 days ago      3.4GB
homeassistant/raspberrypi4-homeassistant stable      686aa4a58c69    7 days ago      1.09GB
erlang              latest      14bff0204d2a    12 days ago     1.15GB
portainer/portainer-ce latest      feee005770f9    2 weeks ago     127MB
ubuntu              latest      1c28a1589115    8 weeks ago     65.7MB
ubuntu              18.04      84282c80cdda    8 weeks ago     56.7MB
    
```

Einige Images enthalten nur das Nötigste und sind sehr klein, andere belegen nach dem Download mehr als ein GByte.

### Einblick

Man kann in laufenden Containern von außen aber auch weitere Programme starten. Beispielsweise will man sich in manchen Fällen in den Container selbst einloggen und mit einer Shell umsehen, ob alles an seinem rechten Platz ist.

Die Zeile

```
docker exec -it erlangtest bash
```

startet im laufenden Container eine Shell (bash) und verbindet sie quasi mit dem äußeren Terminal (-it) des Hosts. Gibt man den Befehl `ls` ein, sieht man, dass im Container ein vollständiges Dateisystem vorhanden ist. Aus

Platzspargründen sind in den Containern (und den zugrunde liegenden Images) jedoch nur die für den Betrieb notwendigsten Tools respektive Befehle vorhanden. Nicht mal der Schmalspur-Editor `vi` ist im Erlang-Image installiert. Im Prinzip lässt sich das schnell ändern, denn jeder Container hat über den Host automatisch Zugriff aufs Netzwerk.

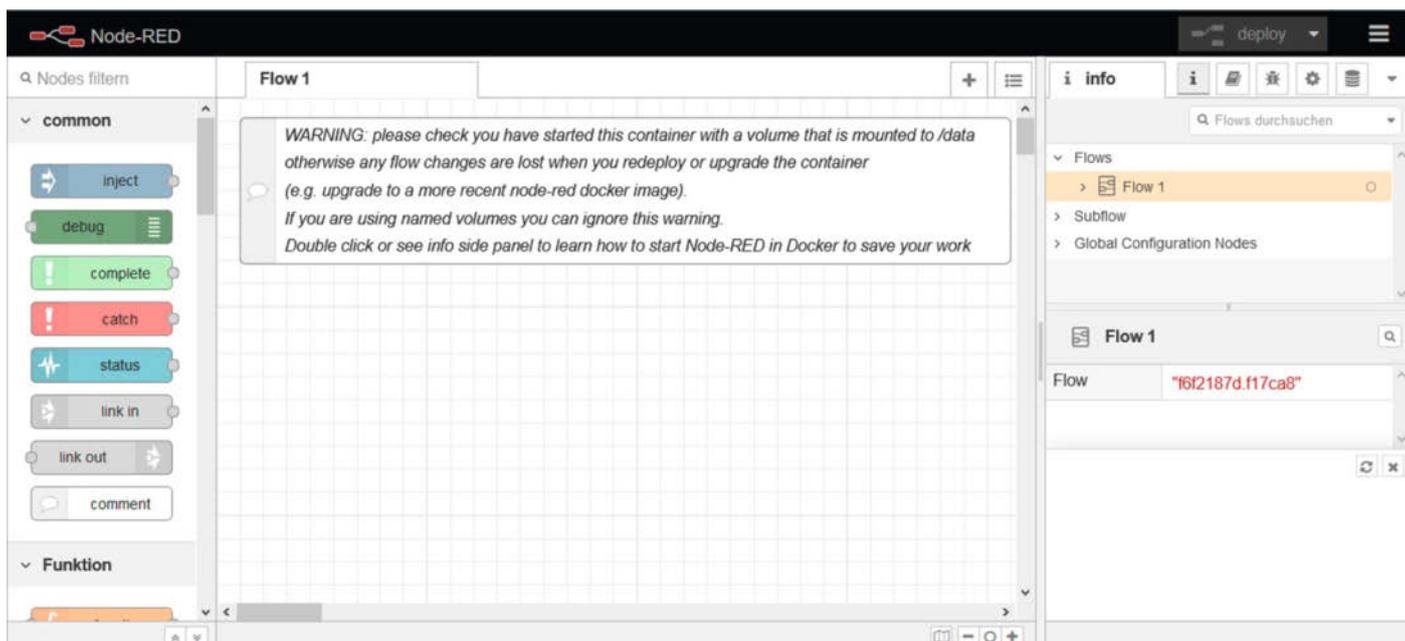
Mit dem Kommando

```
apt-get install vim
```

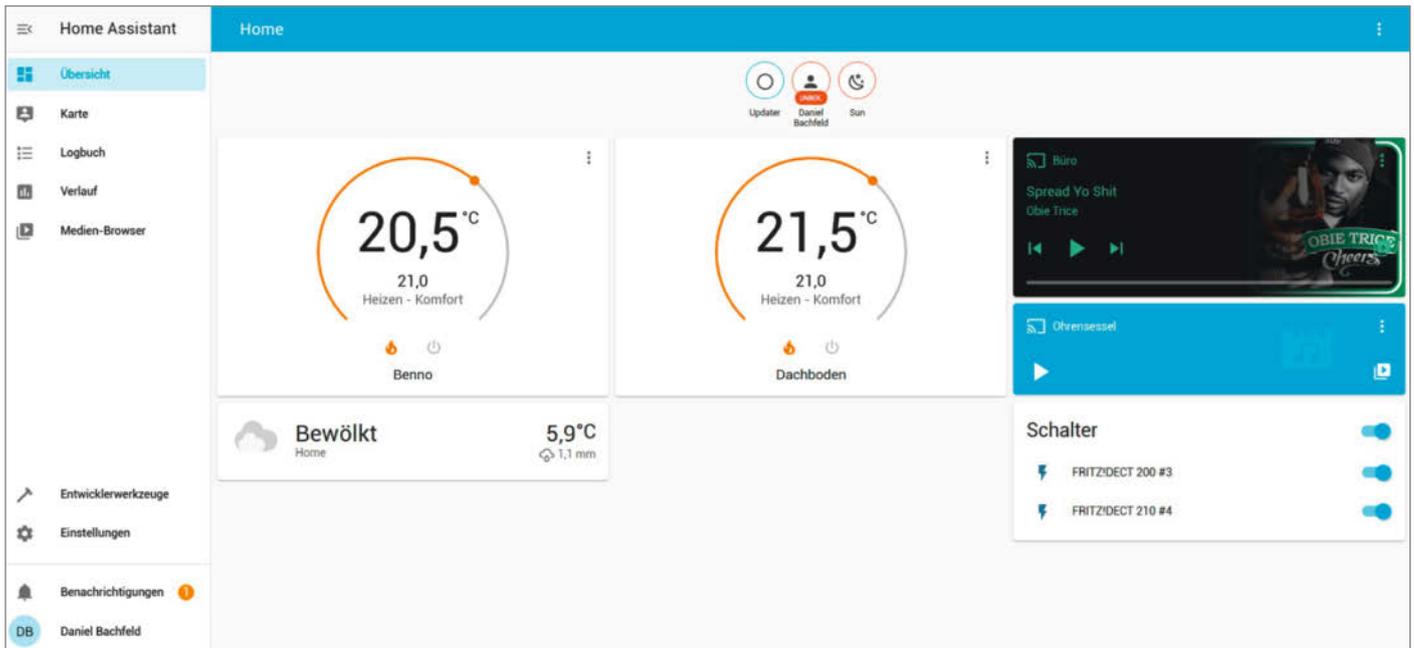
(in der Container-Shell) ist der Editor schnell installiert und verfügbar. Auf diese Weise könnten Sie ihren Container mit weiteren Tools ausstatten und speziell an ihre Wünsche an-

### Sicherheit

Docker-Images von Docker Hub und anderen Seiten herunterzuladen ist wie Programme aus dem Internet zu installieren - man kann sich Schadprogramme einfangen. Auch wenn die Software später im Container läuft, kann man den Host mit gezielten Manipulationen schaden oder das Netzwerk ausspähen. Achten Sie darauf, aus welchen Quellen das Image stammt und wer der Ersteller ist.



Die Oberfläche von Node-RED warnt vor Datenverlust, falls der Container ohne Volume erstellt wurde.



Der Home-Assistant-Container ist schnell installiert und mit wenigen Klicks sind die Fritzbox nebst Aktoren und die Sonos-Boxen integriert.

passen. Sie können später den Container stoppen (`docker stop erlangtest`), bei Bedarf wieder starten und mit weiteren Paketen per `apt-get` einrichten. Zu diesem Vorgehen gibt es aber sehr kontroverse Meinungen in der Community.

Professionelle Admins raten von dieser Art des Container-Baus ab und nutzen diesen Weg nur für Testzwecke. Denn nach der „reinen Lehre“ sollten eigentlich alle erforderlichen Pakete bereits im Image zu finden sein und der Container sollte nur beim Start konfiguriert werden müssen. Was im professionellen Betrieb mit vielen Containern sicherlich die Wartung und Nachvollziehbarkeit erheblich

erleichtert, ist für den Hobbyisten mit seinen Bastelcontainern nicht unbedingt die oberste Prämisse.

### Nützlich

Zugegeben, das Erlang-Beispiel war etwas konstruiert, um zu zeigen, dass man mal eben schnell etwas außergewöhnliches ausprobieren kann, ohne sich stundenlang mit Neben-Quests der Installation herumzuschlagen und ohne das Gesamtsystem mit unnötigen Bibliotheken und speziellen Tools zu überfrachten. Wechseln wir nun zu etwas nützlichem, beispielsweise *Node-RED*.

Mit der populären grafischen Programmierumgebung erstellt man im Handumdrehen Flows für Steuerungen und anderes. Node-RED ist ein webbasiertes Tool, dessen Webserver standardmäßig auf Port 1880 seine Oberfläche anbietet. Folgender Befehl

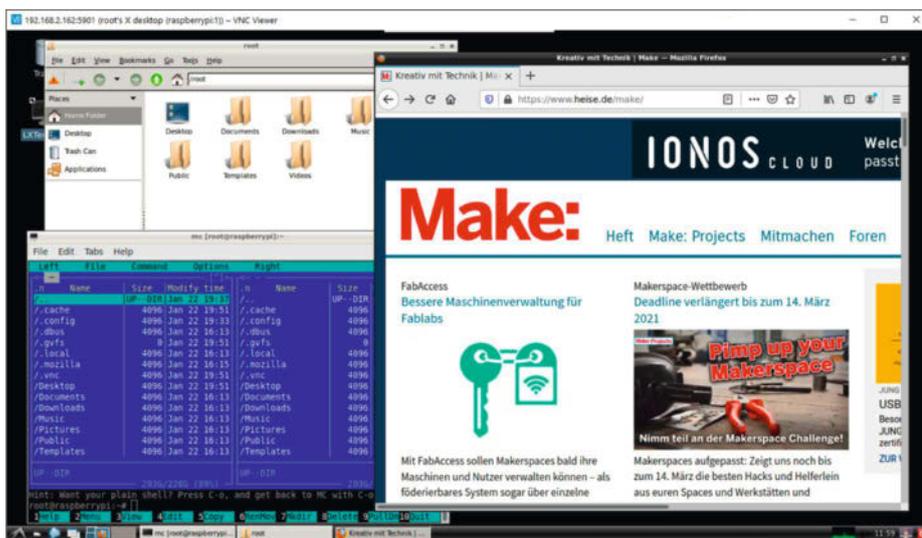
```
docker run -d
-p 1880:1880
-v node_red_data:/data
--restart unless-stopped
--name mynodered
nodered/node-red
```

lädt das Image herunter und erstellt daraus einen Container mit einer speziellen Konfiguration: Die Option `-d` (detach) verhindert Logging-Meldungen des Containers im Terminal. Sie sehen nur eine kryptische Zahlenfolge – die ID des erfolgreich gestarteten Containers.

Die Option `-p 1880:1880` bildet den im Container genutzten (TCP-)Port 1880 des Node-RED-Servers auf den Port 1880 des Hosts ab. Dank dieser Angabe ist der Server im Container von außen über das Netzwerk mit der IP-Adresse des Hosts erreichbar. Der Host reicht alle an ihn gerichteten Pakete mit dem Port 1880 an den Container weiter.

Zudem kann man auf diese Weise den inneren Port elegant auf einen anderen äußeren Port umbiegen, falls der Port auf dem Host schon von einem anderen Dienst belegt ist.

Die Angabe `-v node_red_data:/data` erzeugt ein sogenanntes *Volume* mit dem Namen `node_red_data` und mounted es innerhalb des Containers unter dem Wurzelverzeichnis an den Ordner `data`. Ist das Verzeich-



Ubuntu im Container mit LXDE-Desktop und VNC-Zugriff

nis im Container nicht vorhanden, wird es erzeugt. Das Volume ist quasi ein Speicherort außerhalb des Containers, auf das der Container aber völlig transparent zugreifen kann.

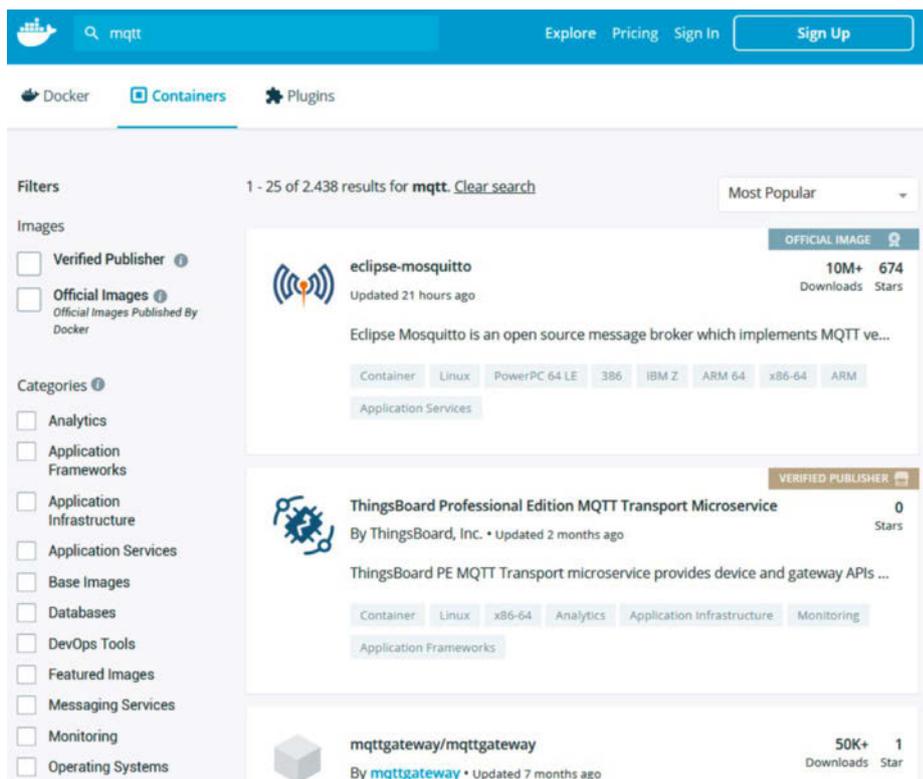
### Volumes

Aus Sicht von Node-RED speichert es im Dateisystem alle installierten Module, erstellten Flows und Einstellungen ab. In Wirklichkeit landen sie aber alle im Volume. Der Vorteil von Volumes ist, dass man auf diese Weise Anwendungen und Daten voneinander trennen kann. Muss der Container aus irgendwelchen Gründen neu aufgesetzt oder aktualisiert werden, löscht man einfach den alten, erzeugt einen neuen und bindet das vorhandene Volume wieder ein. Hätte man die Daten nur im Container abgelegt, müsste man diese zuvor aus dem Container extrahieren und später wieder einspielen. Das zuvor gezeigte Volume ist eine spezifische Funktion von Docker, es ist nicht auf dem Host sichtbar. Man kann sich alle Volumes aber mit

```
docker volume ls
```

anzeigen lassen.

Man kann auch bereits existierende Ordner des Hosts in das Dateisystem des Containers



Docker Hub ist eine der größten Community-Bibliotheken für Docker-Images.



Gute Aussichten für Fotobegeisterte.

Sparen Sie 35% im Abo und sammeln wertvolles Know-how:

- 2 Ausgaben kompaktes Profiwissen für 14,60 € (Preis in DE)
- Workshops und Tutorials
- Tests und Vergleiche aktueller Geräte



Inklusive Geschenk nach Wahl

z. B. Kamera-Reinigungsset

Jetzt bestellen:

[ct-foto.de/miniabo](http://ct-foto.de/miniabo)

## Dockerfile

```
FROM ubuntu:18.04
RUN apt-get update
RUN DEBIAN_FRONTEND=noninteractive apt-get install -y lxde-core lxterminal
RUN apt-get -y install tightvncserver firefox nano
# files for VNC
RUN touch /root/.Xresources
RUN touch /root/.Xauthority
WORKDIR /root
RUN mkdir .vnc
# COPY xstartup with start for lxde
COPY xstartup /root/.vnc/
RUN echo "export USER=root" >> /root/.bashrc
ENV USER root
# COPY script. removes Lock files and start tightvncserver
COPY entrypoint.sh /entrypoint.sh
# set password
RUN printf "maketest\nmaketest\n\n" | vncpasswd
ENTRYPOINT ["/entrypoint.sh" ]
```

einbinden. Dazu muss die Angabe des Volumens immer mit einem Slash (/) beginnen:

```
-v /home/pi/node_red_data:/data
```

Nach dem Start des Containers landen alle in den internen Ordner *data* geschriebenen Dateien im Host-Ordner */home/pi/node\_red\_data*.

Kommen wir zu den restlichen Optionen des obigen Befehls: Durch die Option `--restart unless-stopped` startet der Container automatisch wieder – und bei Reboots des Pi ebenfalls. Die Angabe `--name mynodered` gibt dem Container einen Namen und die Angabe zum Image `nodered/node-red` legt fest, das Image *node-red* aus dem Repo des Anbieters *nodered* zu laden. Die Angabe des Anbieters ist deshalb wichtig, weil es diverse Node-RED-Images auf Docker Hub gibt.

## Optionen

Mit allen erklärten Angaben ist dieser Container nun ein für alle Mal konfiguriert. Beim späteren Aufrufen (entweder automatisch oder mit `docker start mynodered`) müssen Sie diese Optionen nicht mehr angeben. Nachträgliche Änderungen an einem bestimmten Container sind allerdings auch nicht mehr möglich, man kann aber jederzeit einen neuen Container mit anderen Optionen erzeugen und den alten löschen.

Im Hintergrund läuft auf dem Pi nun der Node-RED-Container. Im Browser des Pi gibt man `localhost:1880` ein und schon landet man in der Bedienoberfläche und kann loslegen. Wie man Flows in Node-RED baut, zeigt

Ihnen unser Make Special Node-RED, das sie im heise Shop bekommen. Über das LAN ist der Server ebenfalls erreichbar, dazu ersetzen Sie aber `localhost` durch die IP-Adresse des Pi, die Sie im Terminal mit dem Befehl `ifconfig` herausbekommen.

Sollte Sie der Artikel über Home Assistant auf Seite 100 inspiriert haben, die Software mal auszuprobieren: Auch dafür gibt es fertige Docker-Images, die sogar noch schneller starten, als das native Image für die SD-Karte. Geben Sie

```
docker run --init -d
--name="home-assistant"
-e "TZ=Europe/Berlin"
-v /home/pi/homeassistant:/config
--net=host homeassistant/
raspberrypi4-homeassistant:stable
```

ein, wenn Sie einen Pi4 benutzen, andernfalls tauschen Sie die 4 gegen eine 3 im Namen des Images aus.

Die meisten Optionen sind bereits bekannt, neu ist `--init`, mit dem verwaiste Prozesse im Container beendet werden, deren Parent-Prozess abgestürzt ist. Dies sorgt für weniger Ressourcenverbrauch. Die Angabe `-d` für *detach* schickt den Container gleich in den Hintergrund, ohne dass man im Terminal irgendeine Ausgaben sieht. Mit `-e` übergibt man an den Container eine Umgebungsvariable (environment). Statt eines Volumens nutzt Home Assistant einen Ordner im Home-Verzeichnis des Hosts. Die Option `--net=host` erlaubt dem Container, den gesamten Netzwerkstack des Hosts zu benutzen. Man spart sich damit unter anderem das

Mapping einzelner Ports. Sie sollten diese Option aber sparsam einsetzen und immer schauen, ob sich Dienste in die Quere kommen.

Sobald der Container *up* ist (`docker ps -a`), können Sie sich mit dem Browser über die Adresse `localhost:8123` verbinden. Alles weitere dazu finden Sie dann im Artikel auf Seite 100. Sollte Ihnen Home-Assistant doch nicht zusagen, löschen Sie einfach den Container und das Image.

## Image bauen

Statt nur auf fertige Images zurückzugreifen, kann man auch eigene Images bauen. Doch nicht nur der Wunsch nach einer individuellen Konfiguration kann die Triebfeder dafür sein. Während die auf Docker Hub angebotenen Images eigentlich immer in einer Version für x86/x64-Plattformen angeboten werden, sind die Images für ARM-Plattformen wie Pi und Odroid derzeit noch nicht ganz so verbreitet.

Im folgenden zeigen wir ein Beispiel, wie man ein eigenes Ubuntu-Image mit LXDE-Desktop und VNC-Server baut. Damit kann man einen Container erstellen, mit dessen Desktop man sich vom Pi oder aus der Ferne per VNC verbinden kann. Man hat dann quasi ein Raspbian und ein Ubuntu für Testzwecke parallel am Start. Wir haben diesen Ansatz für unsere Robotik-Experimente gewählt, um das Robot Operating System (ROS) mit GUI auf einem Pi laufen zu lassen.

Listing Dockerfile zeigt den Inhalt eines sogenannten Dockerfiles. Die Idee beim Bau eines Images ist, ein Parent-Image zu laden und darin nach und nach weitere Pakete zu installieren, quasi in einem temporär erzeugten Container. `FROM` lädt eine bestimmte Minimal-Ubuntu-Version und startet diese. `RUN` sendet Befehle an die Shell im laufenden Container, hier etwa Installationsbefehle (`apt-get install`) für diverse Pakete. Nachdem die Pakete installiert sind, legen wir mit `touch` ein paar Dateien und Ordner an, die der VNC-Server benötigt. Mit `WORKDIR` kann man ein Arbeitsverzeichnis festlegen, auf das sich nachfolgende Befehle beziehen.

Der Befehl `COPY xstartup /root/.vnc/` kopiert die Datei *xstartup* aus dem Host-Ordner, in dem sich das Dockerfile befindet, in den Ordner */root/.vnc* des Container-Dateisystems. Die Datei enthält Shell-Befehle zum Start des LXDE-Desktops. Das im folgenden kopierte Skript *entrypoint.sh* ist der Einstiegspunkt, der beim Start eines Containers aufgerufen wird. In ihm kann man zusätzliche Befehle unterbringen. Konkret löscht es in diesem Beispiel eventuell übriggebliebene „Reste“ vorhergehender VNC-Verbindungen und startet den VNC-Server auf dem nächsten freien Port (5900 oder 5901).

Der vorletzte Befehl konfiguriert das VNC-Passwort, indem er die Zeichenfolge *make-test* (inklusive Enter) an das Tool *vncpasswd* schickt. Zum Schluss wird der sogenannte Entry-Point festgelegt, hier das bereits erwähnte Skript. Alle Dateien und das Dockerfile finden Sie in unserem Github-Repository (siehe Link in der Kurzinfo).

Um das Image zu bauen, wechselt man im Terminal in den Ordner, in dem das Dockerfile liegt und gibt folgendes ein:

```
docker build -t ubuntu-vnc .
```

Die Option *-t* gibt dem Image einen Namen, hier *ubuntu-vnc*, der Punkt bedeutet „in diesem Verzeichnis liegt das Dockerfile“ (in situ). Docker benötigt einige Zeit zum Bau des Images und lädt viele Dateien nach. Am Ende steht etwas wie *Successfully built 566ac05f4887* und *Successfully tagged ubuntu-vnc:latest*. Nun erstellen und starten Sie einen eigenen Container:

```
docker run -v /dev:/dev
--privileged -d
--name ubuntu-lxde
--net=host ubuntu-vnc
```

Der Container wird beim Start durch die Kombination *-v /dev:/dev --privileged* mit privilegierten Zugriffsrechten auf Ressourcen des Hosts ausgestattet. Das ist notwendig, um etwa aus dem Container heraus Zugriff auf USB-Geräte wie FTDI-Adapter zu ermöglichen, die im Dateiverzeichnis des Hosts immer unter */dev* registriert werden. Im Prinzip würde das auch ohne die Option *--privileged* funktionieren, allerdings müssen dann die Geräte beim Start des Containers bereits angeschlossen sein. Mit der zusätzlichen Angabe registriert der Container auch nach dem Start neue, im Device-Tree des Hosts hinzugekommene

Geräte. Im übrigen ist mit der Angabe einer der beiden Optionen auch der Zugriff auf die GPIOs des Pi freigeschaltet. Welche man nimmt, ist dann egal.

Läuft der Container *ubuntu-lxde*, können Sie sich mit dem lokal installierten VNC-Viewer des Pi mit dem VNC-Server im Container auf Port 5900 verbinden. Auch aus der Ferne funktioniert die Verbindung – und der Desktop im Container arbeitet überraschend fix. Apropos fix: In der Praxis laufen Anwendungen in Containern etwa 15 Prozent langsamer als auf dem Host.

### Grafisch

Arbeiten Sie mit mehreren Images, Containern und Volumes auf einem Host, wird die Sache bald unübersichtlich und den Befehlszeilen-basierten Docker-Client finden eher Desktop-verwöhnte Anwender sowieso unpraktisch. Abhilfe schaffen freie Adminstrationsstools wie *Portainer*. Man installiert es mit

```
docker run -d
-p 9000:9000
--restart always
--name portainer
-v portainer_data:/data
-v /var/run/docker.sock:
/var/run/docker.sock
portainer/portainer-ce
```

Im Webbrowser ruft man über *localhost:9000* (oder die IP-Adresse) die Oberfläche auf und legt zunächst ein Konto an. Das Tool läuft ebenfalls als Container und bietet über seine Weboberfläche eine Übersicht aller lokal vorhandenen Images, laufende und ruhende Container sowie angelegte Volumes. Daneben kann man komfortabel nach Images auf Docker Hub suchen und herunterladen. Eigene Images kann man erstellen, indem man in der

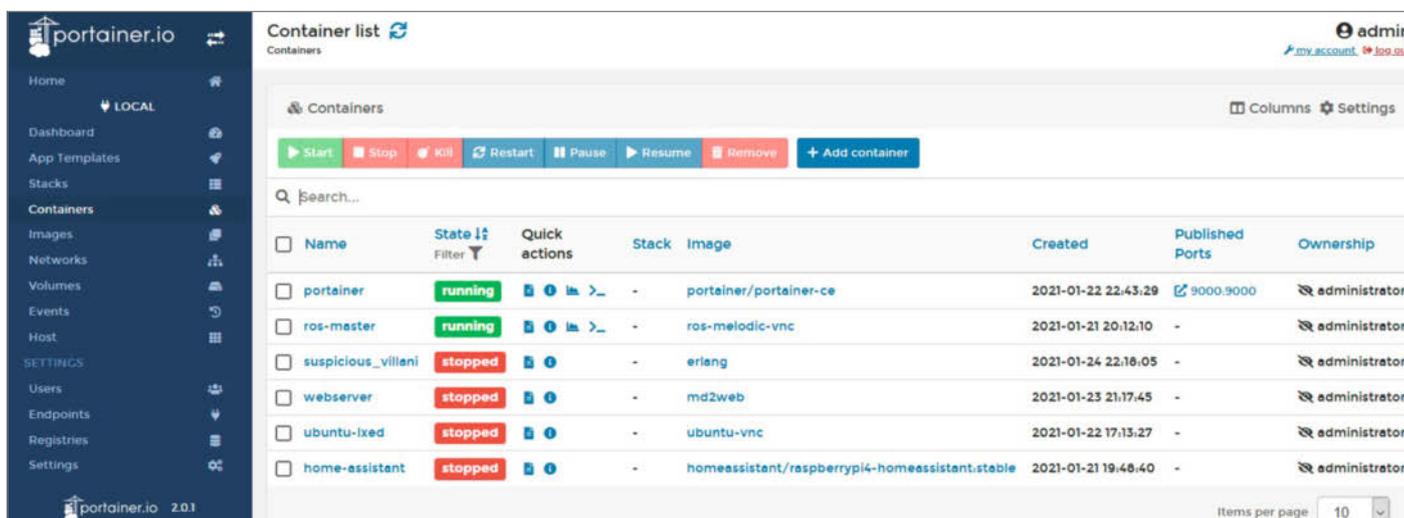
### WLAN-Probleme

In unseren Tests mit Docker auf dem Raspberry Pi gab es immer wieder Probleme im Zusammenhang mit dem WLAN. Mitunter war nach dem Start eines Containers oder während des Builds eines Images das gesamte Netzwerk abgestürzt. Im Betrieb über das Ethernet traten diese Probleme nicht auf. Abhilfe brachte es, für den Buildvorgang zusätzlich die Option *--network host* anzugeben. Beim Start eines Containers (run) half die Angabe *--net=host*. Allerdings führt dies dazu, dass der Container seine Netzwerports quasi komplett auf den Host abbildet, was in manchen Situationen zu Sicherheitsproblemen führen kann.

Oberfläche ein Dockerfile anlegt. Container lassen sich ebenfalls komfortabel erzeugen und insbesondere die möglichen Optionen sind übersichtlich und erklärt dargestellt.

Mit gestarteten Containern kann man über die Option *Quick Actions* in einem virtuellen Terminal sogar über eine Shell zugreifen. Das ist dann nützlich, wenn ein Container wie beim Erlang-Beispiel auf einer Interaktion im Terminal beruht. Anders als im eingangs beschriebenen Beispiel müssen Sie nach dem Start des Terminals den Interpreter mit *er1* manuell aufrufen.

Mit Portainer wird der Umgang mit Docker lokal und aus der Ferne zum Kinderspiel und man kann „mal eben schnell“ Container bauen, um neue Dinge und Ideen auszuprobieren. Endlich Frickeln ohne Reue! —*dab*



Portainer bietet mehr Übersicht und Komfort als der textbasierte Docker-Client.

# Intelligentes Heim mit Home Assistant

Ein Smarthome ist mehr als nur die Fernsteuerung diverser Geräte per Smartphone oder Tablet. Erst die Zusammenarbeit der kleinen elektrischen Helfer, am besten automatisch, macht Ihr Heim wirklich smart. Hier und in den folgenden Make-Ausgaben erfahren Sie, wie das geht.

von Heinz Behling



Schon vor 99 Jahren gab es den Traum vom Smarthome. Glauben Sie nicht? Dann schauen Sie sich mal den Film „Das voll-elektrische Haus“ (orig.: The electric House) von Buster Keaton aus dem Jahr 1922 auf YouTube an. Tonfilm gab es damals noch nicht, wohl aber den Wunsch, das Eigenheim durch zahlreiche dienstbare Elektrogeister komfortabler zu machen. In der Schwarzweiß-Komödie geht das so richtig schief, aber auch heutzutage wird in diesem Bereich so einiges, sagen wir mal, nicht gerade optimal gemacht.

Das liegt nicht zuletzt an dem, was die Verkäufer der durchaus zahlreichen Smart-home-Produkte so alles versprechen: Kinderleicht soll es sein, Energie sparen oder zumindest kostengünstig sollen die verschiedenen Systeme sein. Da fängt es schon an: Verschiedene Systeme. Jeder Hersteller kocht sein eigenes Süppchen, eine Zusammenarbeit mit den Systemen anderer Hersteller ist nicht vorgesehen und meist auch gar nicht erwünscht. Mit Startpaketen versucht man, Kunden ein für allemal an sich zu binden (So ging es auch mir: Mit einer Schaltsteckdose passend zur Fritzbox fing es an).

Wenn man dann merkt, dass es Smart-home-Produkte woanders preiswerter als beim zuerst gewählten Hersteller gibt, ist kaum einer bereit, das wertvolle, mit diesen Alternativprodukten jedoch nutzlose Starterpaket auf den Müll zu werfen (in meinem Fall waren es Heizkörperthermostate, die gegenüber den Fritzbox-Produkten nur halb so teuer waren, aber ich hatte ja schon die Steckdose...). Folglich erweitern die meisten dann ihr Smarthome mit Produkten desselben Herstellers oder lassen es mit dem Smarthome einfach sein, weil es ihnen zu kostspielig wird.

Erreicht wird diese Kundenbindung dadurch, dass zum Steuern der Heizkörperthermostate, Schaltsteckdosen, Lampen usw. immer eine Zentralstation, meist *Gateway* genannt, und oft auch eine App für Smartphone, Tablet oder PC erforderlich sind. Und die verstehen in der Regel nur die Steuersignale des eigenen Systems, selbst wenn sie auf einem eigentlich allgemeinverständlichen Standard wie *Zigbee* beruhen. Philips etwa geht mit seinem *Hue*-System diesen Weg: Es beruht auf *Zigbee*, kann aber mit *Zigbee*-Geräten anderer Marken nichts anfangen.

### Open Source als Alternative

Engagierte Leute entwickelten aber Open-Source-Smarthome-Software, mit deren Hilfe zum Beispiel der Raspberry Pi zu einem universellen Smarthome-Server wird. Da gibt es unter anderem *FHEM*, *openHAB* oder *Home Assistant*. Auf der Suche nach dem – meiner Meinung nach – besten habe ich da einiges ausprobiert. Jedes davon funktioniert, aber dennoch hatte ich schnell meinen Favoriten: Home Assistant.

## Kurzinfo

- » Smarthome-Server für Raspberry Pi
- » Sinnvolle Aufgaben für ein Smarthome
- » Einrichtung von Home Assistant
- » Integration kommerzieller Produkte am Beispiel FritzDECT
- » Automatisierungen selbst programmieren

### Checkliste



#### Zeitaufwand:

1 bis 2 Stunden für die Server-Installation



#### Kosten:

70 Euro (für den Server, zzgl. Kosten für die Geräte)

Alles zum Artikel im Web unter [make-magazin.de/x73r](http://make-magazin.de/x73r)

### Material

- » Raspberry Pi ab Version 3, empfohlen Version 4 mit 2 GB RAM
- » Mikro-SD-Speicherkarte 32 Gigabyte
- » Netzteil passend zum Raspberry Pi
- » ESP-Board Wemos D1 Mini mit USB-Netzteil
- » DHT-11- oder DHT-22-Shield für Wemos D1 Mini

### Mehr zum Thema

- » Heinz Behling, Smarthome-Firmware für ESP-Module, Make 4/20, S. 34
- » Wie man sich ein Smarthome vor 99 Jahren vorstellte, zeigt der Film „Das vollelektrische Haus“ auf YouTube.

Dieses System kommt nicht nur mit den meisten kommerziellen Smarthome-Produkten zurecht, sondern erlaubt durch die Erweiterung *ESPHome* auch Eigenbauten auf Basis von ESP-Boards. Und das alles ist auch noch sehr ausführlich dokumentiert inklusive der erforderlichen Programmcode-Zeilen. Dies habe ich bei den anderen Systemen in diesem Umfang ver-

misst. Daher habe ich mich für Home Assistant entschieden.

### Zusammenarbeit der Systeme

Außerdem ist es in Home Assistant möglich, kommerzielle und DIY-Komponenten nicht

#### o As an image for your device:

- [Raspberry Pi 3 Model B and B+ 32-bit](#) (32-bit is required for GPIO support)
- [Raspberry Pi 3 Model B and B+ 64-bit](#)
- [Raspberry Pi 4 Model B \(1 GB, 2 GB and 4 GB model\) 32-bit](#) (32-bit is required for GPIO support)
- [Raspberry Pi 4 Model B \(1 GB, 2 GB, 4 GB and 8 GB model\) 64-bit](#) (64-bit is required for 8 GB model)
- [Tinkerboard](#)
- [Odroid-C2, Odroid-C4, Odroid-N2, Odroid-XU4](#)
  - [Guide: Flashing Odroid-N2 using OTG-USB](#)
- [Intel NUC](#)

1 Außer auf Raspberrys kann Home Assistant auch auf Odroid-Modellen und dem Intel NUC installiert werden.

## 2 Inhalt der Datei my-network

```
[connection]
id=my-network
uuid=72111c67-4a5d-4d5c-925e-f8ee26efb3c3
type=802-11-wireless

[802-11-wireless]
mode=infrastructure
ssid=MY_SSID
# Uncomment below if your SSID is not broadcasted
#hidden=true

[802-11-wireless-security]
auth-alg=open
key-mgmt=wpa-psk
psk=MY_WLAN_SECRET_KEY

[ipv4]
method=auto

[ipv6]
addr-gen-mode=stable-privacy
method=auto
```

nur zu fernzusteuern, sondern auch problemlos zusammenarbeiten zu lassen und zu automatisieren. Erst dadurch wird ein Haus wirklich smart, wenn es wichtige Dinge einfach selbst erledigt. Falls es irgendeinen dafür benötigten Sensor nicht gibt, baut man ihn selbst. Kleines Beispiel: Mein Discounter-Luftbefeuchter, der

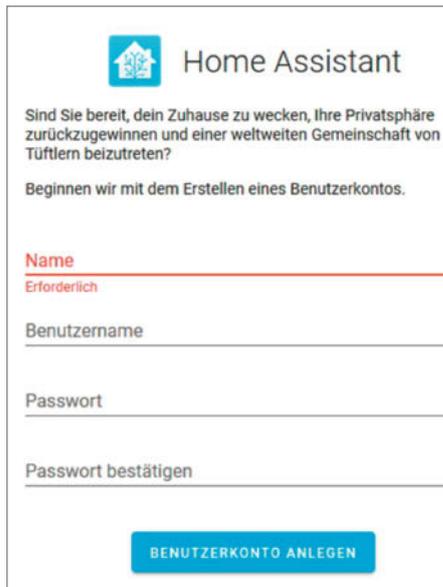
eigentlich gar kein smartes Gerät ist, wird über eine FritzDECT-Schaltsteckdose von einem DIY-ESP-Feuchtesensor gesteuert. Oder meine Rollo-Automatik: Wird es im Wohnzimmer (Südseite) durch Sonneneinstrahlung zu warm, dann werden die Rollos heruntergelassen. Die Information über die von mir gewünschte Raumtemperatur stammt von den Heizkörper-Thermostaten, die Ist-Temperatur misst wiederum ein ESP32 mit DHT22-Sensor.

Daraus entstand diese Artikelserie: In dieser und den folgenden Make-Ausgaben zeige ich Ihnen, wie man Home Assistant installiert, kommerzielle und Selbstbau-Komponenten einbindet und zusammenarbeiten lässt. Den Schwerpunkt habe ich dabei auf Anwendungen gelegt, die mir echt Arbeit abnehmen und somit mein Heim wirklich komfortabler machen.

### Mythos Energiesparen

Bei der Gelegenheit möchte ich Ihnen auch gleich einen Zahn ziehen: Energie sparen Sie mit einem Smarthome in der Regel nicht. Oft hört man Argumente wie: Heizen Sie nur, wenn Sie auch zu Hause sind. Schalten Sie von unterwegs auf dem Heimweg die Heizung ein und sparen Sie so eine Menge Geld. Das hört sich sehr verlockend an, in den meisten Fällen ist der Effekt aber gering.

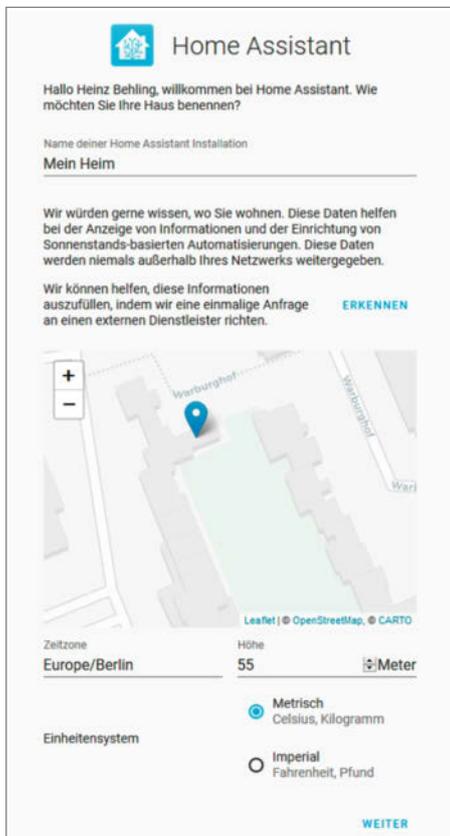
Moderne, gut gedämmte Häuser brauchen ohnehin wenig Heizenergie, sodass eine Absenkung bei Tage nicht viel bringt, denn dann bleiben die meisten Heizkörper ohnehin nahezu kalt, weil die Sonneneinstrahlung oft schon zur Temperaturhaltung genügt. In schlecht isolierten Gebäuden hingegen sinkt während der Abschaltung die Temperatur schnell ab und daher muss beim Wiedereinschalten der Heizung kräftig nachgeheizt werden, was die



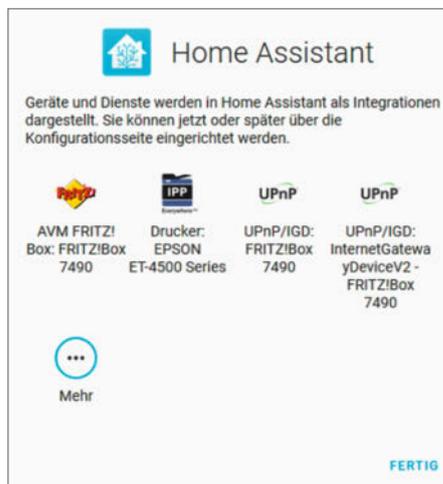
3 Diese Angaben sind zwingend erforderlich.

Energieeinsparung meist wieder aufhebt. In solchen Fällen sollten Sie besser zunächst in die Isolation und die Fenster investieren beziehungsweise mit Ihrem Vermieter darüber reden, bevor Sie versuchen, mit einem Smarthome-System Heizenergie zu sparen. Demnächst können Sie das selbst überprüfen, denn Home Assistant kann Ihnen nicht nur die jeweils aktuelle Temperatur anzeigen, sondern auch Temperaturverläufe in Form von Diagrammen. Dann sehen Sie, wie sich das Ab- und wieder Einschalten der Heizkörper auswirkt und können die Energieeinsparung besser einschätzen.

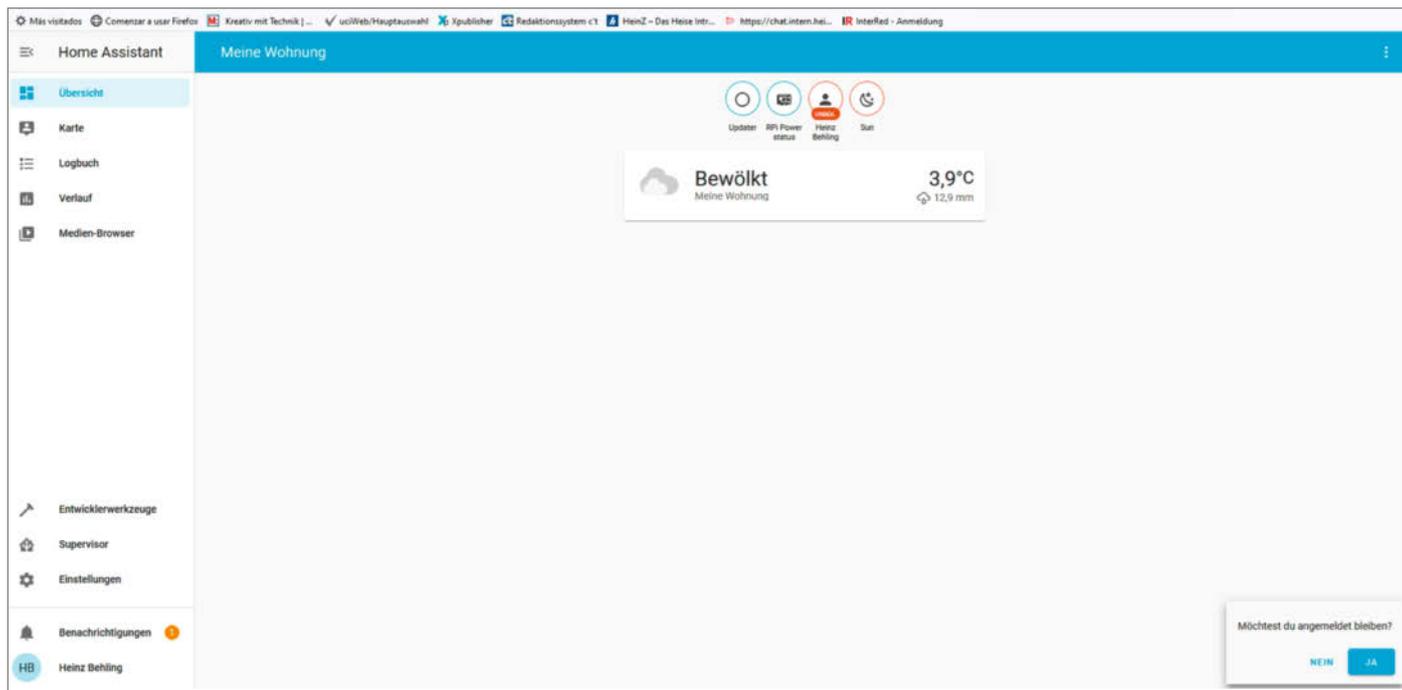
Aber dennoch ist es sinnvoll, auch die Heizung mithilfe von vernetzungsfähigen Heizkörper-Thermostaten ins Smarthome-System mit einzubinden. Falls Sie mal früher von einer längeren Reise zurückkehren, können Sie so



4 Ort und Zeitzone sind wichtig für zahlreiche Funktionen des Home Assistant.



5 Diese Geräte hat Home Assistant erkannt.



6 Noch sehr aufgeräumt: der Hauptbildschirm

für ein mollig warmes Heim bei der Ankunft sorgen. Die Abschaltung beim Öffnen eines Fensters ist so ebenfalls möglich und funktioniert meiner Erfahrung nach auch besser als bei den einfachen elektronischen Thermostaten. Doch dazu später in der Folge zu Heizung und Klima mehr.

In jeder Folge unserer Artikelserie finden Sie eine Anleitung für eine komplette Anwendung. In dieser Ausgabe gibt es neben der Installation des Serversystems und der wichtigsten Erweiterungen auch gleich die schon erwähnte Luftbefeuchter-Steuerung und den Selbstbau-Sensor. Ich erkläre Ihnen an diesem Beispiel, wie man Automationen in Home Assistant verwirklicht. Für die nächste Ausgabe plane ich unter anderem die Rollo-Steuerung, Sprachsteuerung und Multimedia werden ebenfalls Themen dieser Reihe sein. Außerdem zeige ich Ihnen nach und nach, wie Sie die gängigen kommerziellen Smarthome-Systeme mit ins Home-Assistant-System einbauen. Ergänzt wird es durch zu den DIY-Komponenten passenden Gehäusen (als 3D-Druck oder Lasercut), die ich dann zeitnah auf unserer neuen Plattform *Make Projects* veröffentliche.

Falls Sie ebenfalls Ideen fürs Smarthome haben, dann schreiben Sie mir bitte. Der Plan für diese Artikelstrecke ist nicht in Stein gemeißelt, sondern soll auch zum Mitmachen einladen.

### Server-Installation

Auf der Homepage von Home Assistant gibt es Image-Dateien der Software passend für

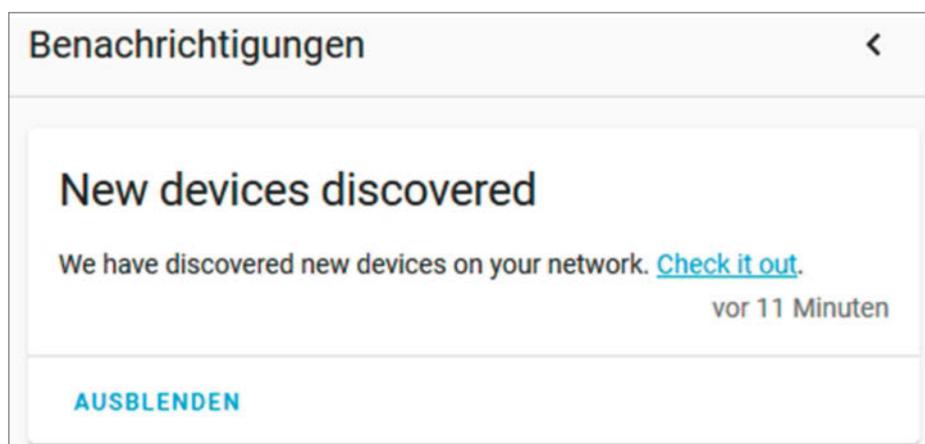
Raspberry 3 und 4 sowie einige Odroid- und andere Boardsmodelle 1.

Ich verwende die 32-Bit-Version für den Raspberry Pi 4 und schreibe das Image mit Hilfe des *Pi Imagers* auf die Speicherkarte. Anschließend lege ich auf der Karte eine Datei mit den Zugangsdaten fürs WLAN an. Auf Windows-PCs sollte man dazu als Editor *Notepad* verwenden, damit die Zeilenumbrüche linux-konform werden. Der Inhalt der Datei sieht so aus: 2.

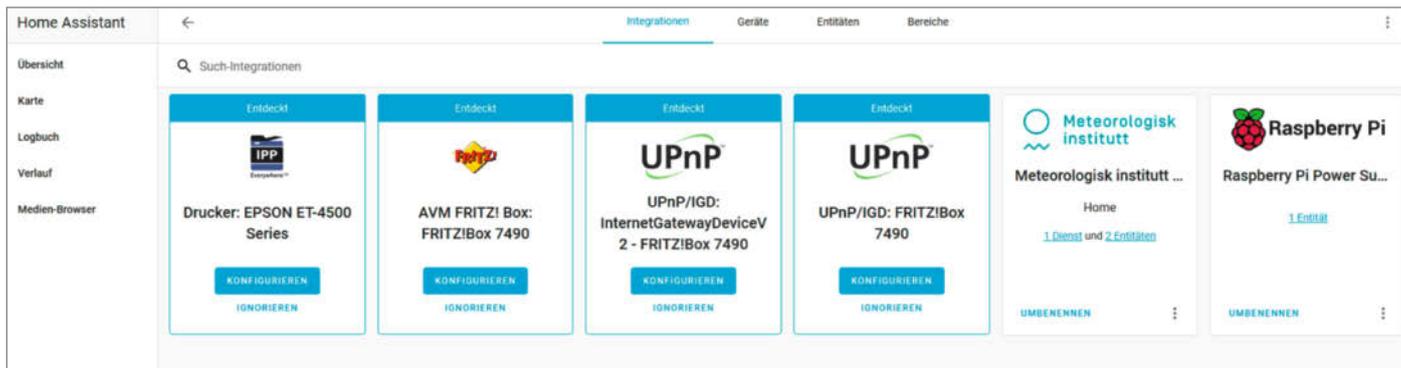
Für MY\_SSID setzen Sie den Namen Ihres WLANs ein, entsprechend für MY\_WLAN\_SECURITY\_KEY das dazugehörige Passwort. Die Datei speichern Sie dann unter dem Namen *my-network* auf der SD-Karten-Partition *hassos-boot* im Ordner */CONFIG/network*. Sollten diese Ordner dort nicht existieren, legen Sie sie an.

Danach ist die Speicherkarte einsatzbereit. Stecken Sie sie in den Raspberry Pi, verbinden Sie ihn mit Strom und warten Sie den Boot-Vorgang ab. Beim ersten Start macht Ihr neuer Home Assistant-Server nun eine Art Selbstfindung durch, die bis zu 20 Minuten dauern kann. Hier werden nämlich unter anderem auch alle im Netzwerk bereits vorhandenen Geräte auf ihre Smarthome-Tauglichkeit geprüft. Nach Ablauf dieses Vorgangs ist Ihre Smarthome-Zentrale im Browser unter der Adresse <http://homeassistant.local:8123> zu erreichen.

Bei der Ersteinrichtung müssen Sie ein paar Daten über Ihr Heim und über Sie selbst eingeben. Zunächst werden Sie nach Ihrem Namen, den Benutzernamen und einem Passwort gefragt 3.



7 Es wurden neue Geräte im Netzwerk gefunden. Beim ersten Start ist das nahezu selbstverständlich.



**8** Fritzbox, der Drucker, ein Wetterdienst und der Raspberry selbst wurden beim Start entdeckt.

Dann sind Angaben zum Standort Ihres Heimes fällig. Geben Sie alles möglichst genau an, auch den Standort **4**. Die Landkarte hilft, Ihr Haus ganz genau zu finden. Diese Angaben brauchen wir hinterher noch, um die automatische Ortung von Haushaltsmitgliedern zu ermöglichen. Diese Daten werden nur lokal auf den Raspberry gespeichert.

Anschließend zeigt Ihnen Home Assistant an, welche Smarthome-kompatiblen Geräte im Netzwerk gefunden wurden **5**.

Schließlich ist es soweit: Sie landen im Hauptbildschirm von Home Assistant **6**.

Unten links finden Sie eine farbige Markierung neben *Benachrichtigungen*. Das passiert immer, wenn Home Assistant Ihnen etwas mitteilen möchte, beispielsweise, weil beim Start etwas fehl schlug. Um die Nachricht lesen zu können, klicken Sie auf *Benachrichtigungen*, und Sie erfahren dann Näheres **7**.

Neue Geräte (devices) wurden beim Start entdeckt. Ein Klick auf *Check it out*, und Sie erfahren, welche **8**.

Das kann natürlich bei Ihnen daheim entsprechend der Geräteausstattung anders aussehen. Wenn Sie jetzt genau auf den Bildschirm schauen, stehen die neuen Komponenten alle unter dem Registerreiter *Integrations*. Das sind in Home Assistant Erweiterungen beispielsweise für kommerzielle Smarthome-Systeme (die Fritzbox ist so eines), Online-Dienste (wie der schwedische Wetterdienst) und auch für den Raspberry gibt es solche eine Erweiterung.

Geräte in Ihrem Netzwerk gesucht und für gefundene gleich die entsprechende Erweiterung mit geladen. Jede dieser Integrationen bietet nun die Möglichkeit, sie zu konfigurieren, das heißt, die jeweils dazugehörigen Geräte einzustellen und auszuwählen, welche der von ihnen zur Verfügung gestellten Informationen angezeigt werden sollen.

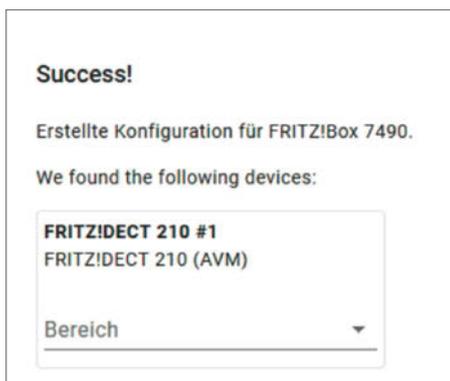
Beginnen wir mit der Fritzbox: Klicken Sie dort auf *Konfigurieren*. Sie werden nach dem Benutzernamen und dem Passwort für den Fritzbox-Zugang gefragt. Anschließend sieht man dann auch endlich wirklich die Geräte, die in diesem Fall an der Fritzbox angeschlossen sind. Siehe da, die Schaltsteckdose **9**!

Konfigurieren Sie auf diese Weise alle anderen Integrationen und Geräte durch. Sie werden staunen, was da alles auftauchen kann. Übrigens: Bei vielen Geräten können Sie angeben, in welchem Teil Ihrer Wohnung es steht, also beispielsweise im Wohnzimmer. Falls in der Auswahlliste ein Raum fehlt, können Sie weitere Räume hinzufügen. Vielleicht halten Sie das im Moment für unnötig, schließlich wissen Sie ja, wo Ihr Drucker oder Ihr Router steht. Wir werden im Verlauf dieser Serie aber noch einige Komponenten so bauen, dass sie möglichst nicht auffallen, ja sogar regelrecht versteckt werden. Dann ist es gut zu wissen, in welchem Raum der jeweilige elektronische Helfer arbeitet. Wenn Sie die Integrationen schließlich alle konfiguriert haben, wird ihnen zu jeder angezeigt, wie viele Geräte und/oder Dienste dazugehören **10**.

Ich weiß, den ersten Lesern schwirrt es jetzt langsam im Kopf: Integrationen, Geräte, Dienste... Diese Begriffe begegnen Ihnen in Home Assistant immer wieder. Keine Angst, Sie gewöhnen sich rasch daran, das lernen Sie ganz nebenbei.

Weiter geht es: Die Integration *Fritz!SmartHome* umfaßt ein Gerät. Klickt man darauf, wird wieder die Bezeichnung der Schaltsteckdose angezeigt **11**.

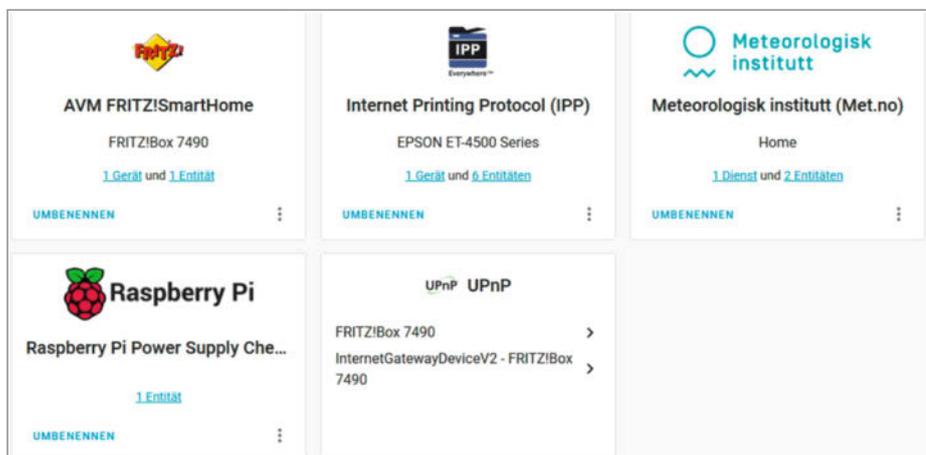
Außerdem besitzt die Fritzbox aber auch noch eine *Entität*. Was ist das denn nun schon



**9** Das erste kommerzielle Gerät ist eingebunden: die Fritzbox-Steckdose.

**Erweiterungsmöglichkeiten durch Integrationen**

Bei der Selbstfindung, von der ich beim ersten Booten sprach, hat Home Assistant die



**10** Zu den einzelnen Integrationen gehören sowohl Geräte als auch Dienste.

| ↑ Gerät           | Hersteller | Modell         | Bereich | Integration         | Batterie |
|-------------------|------------|----------------|---------|---------------------|----------|
| FRITZ!DECT 210 #1 | AVM        | FRITZ!DECT 210 |         | AVM FRITZ!SmarHo... | -        |

11 Das bislang einzige Gerät an meiner Fritzbox: die Steckdose *Fritz!Dect 210*

| <input type="checkbox"/> | ↑ Name  | Entitäts-ID             | Integration        | Bereich | Status  |
|--------------------------|---|-------------------------|--------------------|---------|---|
| <input type="checkbox"/> |  FRITZ!DECT 210 #1 | switch.fritz_dect_210_1 | AVM FRITZ!SmarHome |         |  |

12 Die Schaltsteckdose ist aber auch eine Entität.

wieder? Klicken Sie darauf, und Sie sehen einen etwas anderen Eintrag 12.

Laut Wikipedia ist eine Entität so etwas wie das Seiende, Existierende. In Home Assistant übersetzt man es vielleicht am besten mit *Ding*. Alles, was Home Assistant in irgendeiner Weise benutzen kann, ob es ein Gerät, ein Internet-Dienst, ein Programm, eine Funktion eines Gerätes oder anderes ist, wird als Entität bezeichnet. Übrigens ist auch jeder Benutzer eine Entität. Insofern ist der Begriff Ding nicht ganz korrekt. Wichtig: Jede Entität hat ihren eigenen Namen (*Entitäts-ID*). Bei der Steckdose lautet er `switch.fritz_dect_210_1`. Diese Namen werden nach einem System vergeben: Zu Beginn steht in der Regel die Funktion, die zur Verfügung steht. Hier also eine Schaltfunktion (*switch*). Dann folgt meist die Typenbezeichnung (*fritz\_dect\_210*) gefolgt von einer laufenden Nummerierung (*1*). Es könnte ja sein, dass man mehrere gleichartige Geräte betreibt. Die ließen sich dann anhand der

laufenden Nummer unterscheiden. Diese Entitäts-IDs brauchen wir immer dann, wenn wir mit den dahintersteckenden Dingen irgendwas machen möchten.

Übrigens: Ist Ihnen das Ausrufezeichen am Ende der Entitäts-Zeile aufgefallen? Das besagt, dass diese Entität im Moment nicht erreichbar ist. Hintergrund: Ich hatte die Steckdose ja schonmal mit der Fritzbox benutzt und daher dort angemeldet. Im Moment liegt sie jedoch untätig im Schrank. Sobald man solche Geräte aber wieder in Betrieb nimmt, verschwindet das Ausrufezeichen.

Jetzt können Sie sich einmal durch alle von Ihrem Home Assistant angezeigten Geräte und Entitäten durchklicken. Bei einigen findet man überraschende Funktionen. Mein Drucker beispielsweise zeigt Sensoren für die Füllstände der vier Tintentanks, die Betriebsstundenzeit und noch einen weiteren Sensor an, von dem ich später erfuhr, dass er aktuelle Druckvorgänge anzeigt.

### Zimmer einrichten

Klicken Sie einmal auf den Registerreiter *Bereiche*. Dort sollten Sie für jeden Raum Ihres Hauses/Ihrer Wohnung einen Eintrag einsetzen. Küche, Schlaf- und Wohnzimmer sind schon vorhanden. Unten rechts steht das Feld *Bereich hinzufügen*. Klicken Sie darauf und geben Sie den Namen des neuen Raumes ein. Ich gehe im folgenden von diesen Räumen aus: 13

### Die erste Anzeige

Mit einem Klick auf *Übersicht* (links im Menü) kehren wir jetzt zum Hauptbildschirm zurück. Der ist nun schon deutlich voller, denn alle zuvor konfigurierten Geräte und Dienste erscheinen nun dort 14.

Später werden wir Ordnung in dieses Gewusel bringen und für jedes Zimmer jeweils einen eigenen Registerreiter mit den entspre-



**+ Nano-Axe-Board mit PICAXE-08M2**

**Make Picaxe Special**  
shop.heise.de/make-picaxe

24,95 € >

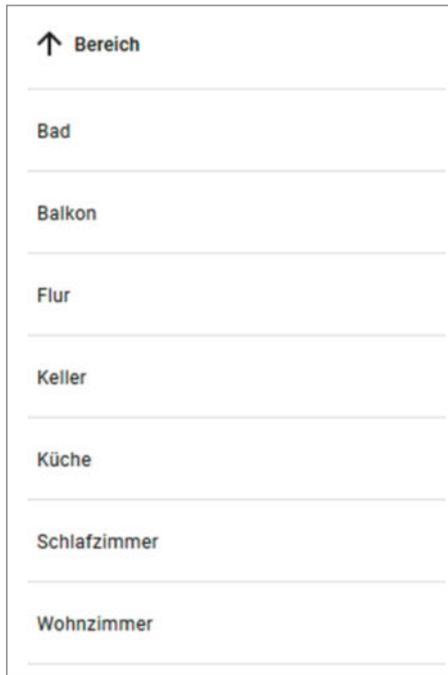
**Exklusiv im heise shop!**

# Back to BASIC

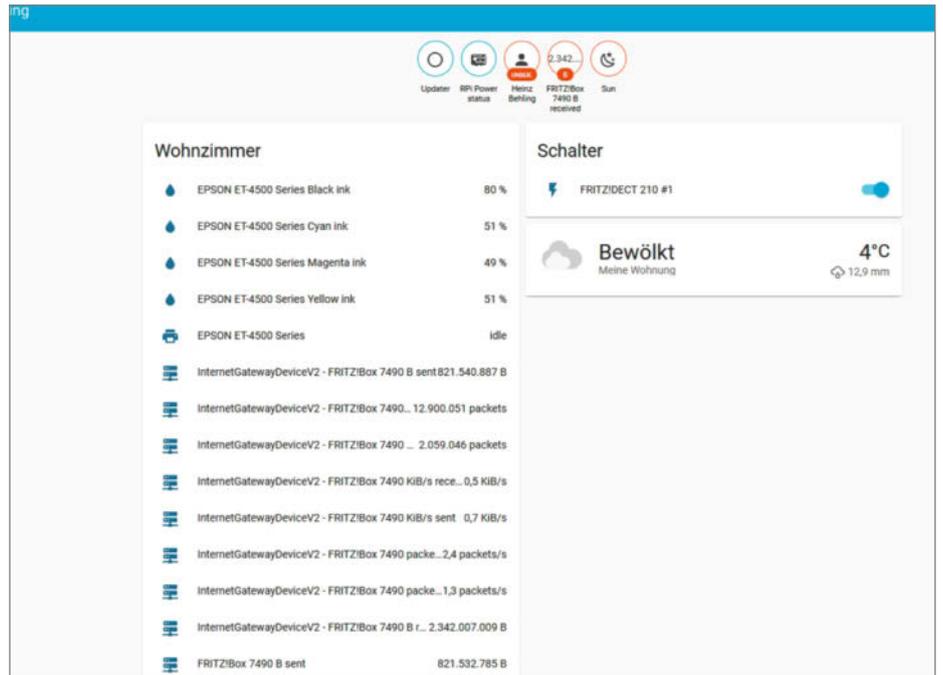


shop.heise.de/make-picaxe >

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



13 Die Raumliste unserer Musterwohnung



14 Alle Geräte und Dienste erscheinen nun auf dem Hauptbildschirm.

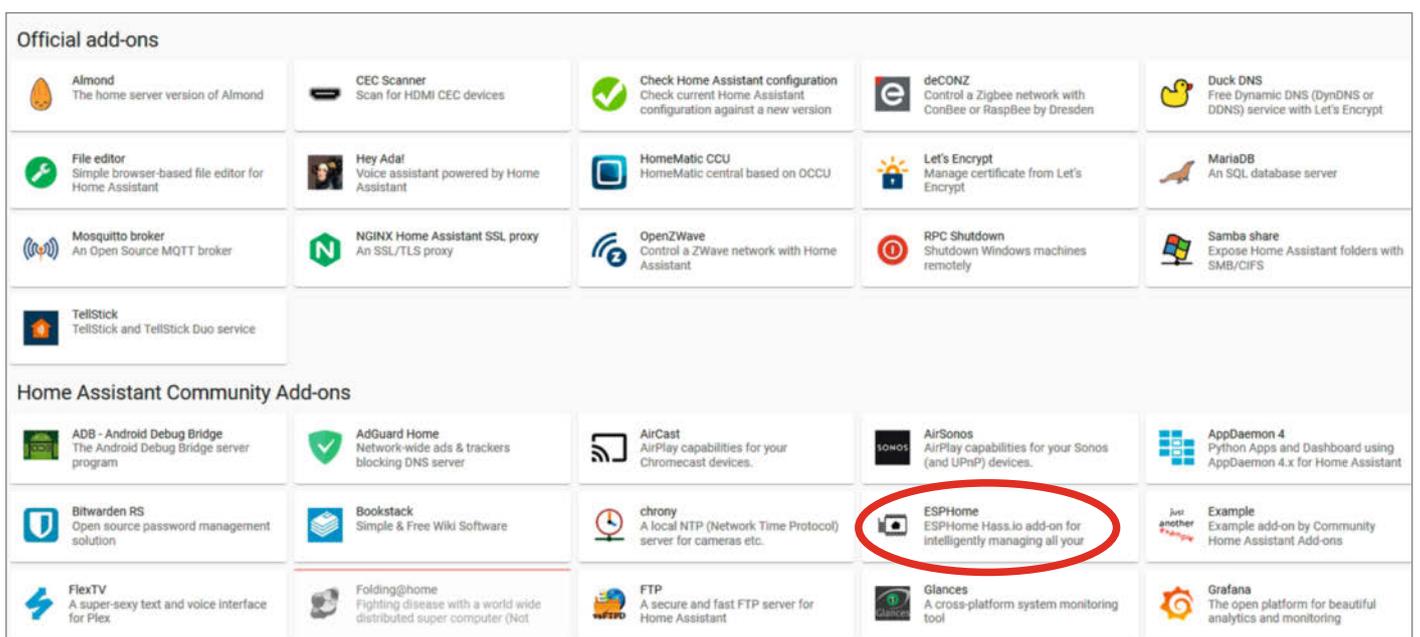
chenden Anzeigen einrichten. Im Moment ist das aber noch nicht so wichtig. Sie sehen hier jetzt auch schon das Schaltersymbol für die Fritz-Steckdose, die ich in der Zwischenzeit wieder in Betrieb genommen habe. Mit der Maus können Sie den Schalter auch betätigen und hören an der Steckdose, wie sie schaltet. Das erste Erfolgserlebnis. Ein Teil unseres Projektvorhabens namens *Feuchte Luft* ist also schon betriebsbereit.

### System erweitern

Jetzt geht es an den Sensor. Als Board verwende ich ein *Wemos D1 mini* und ein dazu passendes Shield mit dem Temperatur-/Feuchtigkeitssensor *DHT22*. Die Bezugsquellen dafür erhalten Sie über den Kurzinfo-Link. Bevor das Board jedoch programmiert werden kann, braucht unser Home Assistant erst noch eine Erweiterung, nämlich die Integration *ESP*

*Home*. Sie finden sie im *Add-on Store*, den Sie nach einem Klick auf *Supervisor* in der Menüleiste finden 15.

Klicken Sie auf *ESPHome* und dann auf *Install*. Die Erweiterung wird aus dem Netz gesaugt und nach ein bis zwei Minuten steht sie zur Verfügung. Bevor Sie auf *Start* klicken, kontrollieren Sie, ob die Schalter *Start on boot* und *Show in sidebar* eingeschaltet sind. Die Erweiterung erscheint dann in der Menüleiste 16.



15 ESPHome finden Sie in den *Community Add-ons*.

## Selbstbau-Sensor

Mit Hilfe von ESPHome schreiben wir nun das Programm zur Lufttemperatur- und Feuchtemessung in das Wemos-Board. Immer, wenn man ein ESP-Board zum ersten Mal mit Home Assistant benutzt, muss diese Programmübertragung über ein USB-Kabel direkt vom Raspberry zum Board erfolgen. Spätere Änderungen hingegen können drahtlos via WLAN erfolgen (OTA: **over the air**). Leider erkennt Home Assistant Boards an den USB-Buchsen nicht automatisch. Sie müssen daher den Raspberry jetzt neu booten. Dazu klicken Sie auf *Supervisor*, dann auf den Registerreiter *System* und auf *Reboot Host*. Bestätigen Sie die Sicherheitsfrage.

Klicken Sie dann in der Menüleiste auf *ESP Home*. Um eine neue ESP-Komponente (in Home Assistant *Node* genannt) hinzuzufügen, klicken Sie auf das blinkende Pluszeichen unten rechts. Nach einem Klick auf *Begin* geben Sie den Namen des Nodes ein. Für den Namen dürfen Sie nur Kleinbuchstaben verwenden. Er wird später als Adresse für diesen Node zum Beispiel beim Update übers WLAN benutzt. Daher lässt er sich später nur ändern, wenn Sie den Node erneut über ein Kabel mit dem Raspberry verbinden. Tipp: Der Name sollte den Raum und die Funktion beinhalten, dann wird es einfacher, später die Übersicht zu behalten. Nennen wir ihn also *temp\_feucht\_wohnzimmer*. Nach einem Klick auf *Next* suchen Sie die korrekte Board-Bezeichnung aus, also *Wemos D1 mini*. Wieder ein *Next* weiter werden dann noch WLAN-Name (*WiFi SSID*) und Passwort (*WiFi Password*) abgefragt. Noch ein Klick auf *Next* und Sie haben das Grundgerüst für die ESP-Firmware angelegt. Klicken Sie auf *Submit*.

Dann erhalten Sie eine Liste mit allen ESP Home-Nodes. Im Moment ist es aber nur einer **17**.

Der enthält jedoch nur die Codezeilen für den Netzwerkzugang, aber keinerlei Programmcode für die eigentliche Funktion, nämlich die Messung von Temperatur und Feuchte. Da helfen aber die Internet-Seiten von ESPHome weiter. Dort finden Sie für eine schier endlose Zahl von Komponenten den jeweils passenden Programmcode. Öffnen Sie in Ihrem Browser einfach einen zweiten Tab und gehen Sie auf <https://esphome.io/>



**17** Der erste ESPHome-Node

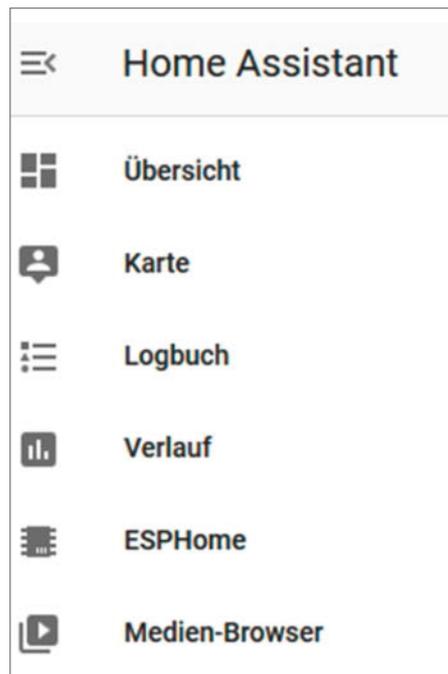
components/. In die Suchzeile dort geben Sie *dht22+configuration* ein. In der Regel finden Sie bereits im ersten gefundenen Eintrag das Programmbeispiel **18**.

Kopieren Sie den grau unterlegten Text in die Zwischenablage und wechseln Sie wieder in den *Home Assistant*-Tab. Dort klicken Sie auf *Edit* unter der Bezeichnung unseres gerade angelegten Nodes. Im dann angezeigten Textfeld kopieren Sie den Text aus der Zwischenablage unter die letzte vorhandene Zeile **19**.

Eine Änderung ist noch notwendig. Bei dem hier benutzten Sensorshield wird der Anschluss *D4* für die Datenübertragung verwendet, nicht *D2*. Ändern Sie also die Zeile entsprechend. Außerdem müssen Sie die genaue Modellbezeichnung eintragen. Das sollte am Ende dann so aussehen: **20**

## Eingebaute Programmprüfung

Mit einem Klick auf *Save* wird alles gespeichert. Schließen Sie das Fenster mit *Close*. Sie kommen wieder in die Node-Anzeige. Dort können Sie den Programmcode überprüfen lassen. Dazu genügt ein Klick auf *Validate*. Ist alles ok, erscheint eine entsprechende Meldung. Andernfalls haben Sie beim Kopieren über die Zwischenablage meist ein Zeichen vergessen. Schließen Sie dann das Validate-



**16** Glückwunsch: Sie haben gerade Ihre erste Integration von Hand installiert!

Fenster. Klicken Sie auf *OTA* und wählen Sie die USB-Schnittstelle aus. Ein Klick auf *Upload* kompiliert dann die Firmware (dauert ein paar Minuten) und überträgt sie ins ESP-Board.

The screenshot shows the ESPHome website search results for 'DHT22 Temperature & Humidity Sensor'. It includes a table of contents, a note about pull-up resistors, and a table of sensor data. A red circle highlights the example configuration code for the sensor.

|   |                         |         |
|---|-------------------------|---------|
| 🔥 | Living Room Temperature | 15.6 °C |
| 💧 | Living Room Humidity    | 63 %    |

```
# Example configuration entry
sensor:
  - platform: dht
    pin: D2
    temperature:
      name: "Living Room Temperature"
    humidity:
      name: "Living Room Humidity"
    update_interval: 60s
```

**18** Der Programmcode für den Sensor steht ganz unten.

Ist die Übertragung komplett, werden die vom neuen Node übertragenen Daten nach etwa einer Minute angezeigt. Sie werden bemerken, dass Home Assistant Ihnen wieder etwas mitteilen möchte, da neben Benachrichtigungen erneut eine farbige Markierung steht. Schauen Sie nach: Es wurde wieder ein neues Gerät gefunden. Mit einem Klick auf *Check it out* sehen Sie, dass der gerade eingerichtete Node entdeckt wurde <sup>21</sup>.

Konfigurieren Sie ihn wie am Anfang beschrieben.

### Automatik-Betrieb

Somit haben wir auch die zweite Komponente unseres Feuchte-Luft-Projekts, den Sensor. Jetzt müssen wir beides zusammenbringen. Wir wollen ja, dass der Luftbefeuchter bei einer Luftfeuchtigkeit unter 50% ein-, bei Erreichen von 50% hingegen wieder ausgeschaltet wird. Für beide Vorgänge brauchen wir je eine Regel, die wir unter *Einstellungen* und *Automatisierungen* anlegen. Nach einem Klick auf *Automatisierung hinzufügen* wählen wir *Mit einer leeren Automatisierung beginnen*.

Geben Sie einen Namen für die Automatik an, zum Beispiel *Luftbefeuchter an (Feuchte < 50%)*. Es ist sehr hilfreich, bei solchen einfachen Schaltautomatiken mit festen Schwellpunkten den Schwellwert jeweils im Namen zu erwähnen. Nach einigen Wochen erinnert man sich nämlich bestimmt nicht mehr daran und muss dann jeweils nachschauen, falls etwas nicht so läuft, wie es soll. Als Auslösertyp wählen Sie *Numerischer Zustand* und zwar für die Entität *sensor.living\_room\_humidity*. Das ist die Entität-ID unseres Feuchtesensors. Bei einer Feuchte unter 50% soll eingeschaltet werden, also schreiben Sie hinter *unter* eine *50*. Damit ist der Auslöser fertig.

Scrollen Sie nun etwas nach unten bis zu den *Aktionen*. Der Aktionstyp ist *Gerät*, denn das ist unsere Steckdose ja. Darunter müssen Sie auswählen, um welches Gerät es sich handelt. In diesem Beispiel die *Fritz!DECT 210 #1*. Als *Aktion* schließlich kommt nur *Schalte Fritz!DECT 210 #1 ein* in Frage. Klicken Sie auf *Speichern*, und diese Automatik ist fertig. Ein Klick auf den Reiter *Automatisierungen* bringt Sie zurück. In der Liste steht nun unsere Automatik.

Jetzt wiederholen wir das Ganze für das automatische Abschalten, wenn die Feuchte über 49% steigt (der Sensor meldet nur ganze Zahlenwerte). Diese Automatik nennen wir *Luftbefeuchter aus (Feuchte > 49%)*. Statt einen Wert bei *unter* tragen wir diesmal einen bei *über* ein, und zwar *49*. Als *Aktion* wählen wir außerdem *Schalte Fritz!DECT 210 #1 aus*. Speichern nicht vergessen, Luftbefeuchter in die Schaltsteckdose einstecken

```
Editing: temp_feucht_wohnzimmer.yaml
1 esphome:
2   name: temp_feucht_wohnzimmer
3   platform: ESP8266
4   board: d1_mini
5
6 wifi:
7   ssid: "hbvdr"
8   password: "babybaer"
9
10 # Enable fallback hotspot (captive portal) in case wifi connection fails
11 ap:
12   ssid: "Temp Feucht Wohnzimmer"
13   password: "CZhxT7XfVmZ4"
14
15 captive_portal:
16
17 # Enable logging
18 logger:
19
20 # Enable Home Assistant API
21 api:
22
23 ota:
24 # Example configuration entry
25 sensor:
26 - platform: dht
27   pin: D2
28   temperature:
29     name: "Living Room Temperature"
30   humidity:
31     name: "Living Room Humidity"
32   update_interval: 60s
```

<sup>19</sup> Der Programmcode muss immer hinter der letzten Zeile eingefügt werden.

```
# Example configuration entry
sensor:
- platform: dht
  model: dht22
  pin: D4
  temperature:
    name: "Living Room Temperature"
  humidity:
    name: "Living Room Humidity"
  update_interval: 60s
```

<sup>20</sup> Der Programmcode mit Modellbezeichnung

und die Luftfeuchtigkeit wird nicht mehr unter 50% sinken und uns einen trockenen Hals oder eine ausgedörrte Nase verursachen.

Damit endet nun die erste Folge der Smarthome-Serie. Sie haben nun bereits eine Menge gelernt. Schauen Sie in den nächsten Wochen einmal auf *Make Projects* nach. Dort werde ich ein 3D-Druck-Gehäuse für den Sensor zeigen. Übrigens: Sie können ihn nun vom RasPi trennen und mit einem eigenen USB-Netzteil versorgen.

In der nächsten Folge befaße ich mich dann mit der Temperatur und Heizung. Vielleicht machen Sie ja inzwischen Pläne, was Sie alles in Ihrem Heim automatisieren möchten. Schreiben Sie mir ruhig eine Mail mit Anregungen und Wünschen. Falls möglich, werde ich das berücksichtigen. Bis dahin: Bleiben Sie gesund! —hgb



<sup>21</sup> Der neue Sensor wurde entdeckt.

Der digitale Treffpunkt für Security-Experten

## ONLINE-WORKSHOPS

### 3. MÄRZ 2021

// **Schneller als der eigene Schatten –  
Entwicklung einer schnellen Reaktionsfähigkeit im Notfall**

Lukas Reike-Kunze

// **Stolpersteine in der Wolke –  
Sicherer Einsatz von Microsoft Office 365**

Kevin Kirchner

// **„Ist das sicher oder in JavaScript?“ –  
Webanwendungen in den Augen eines Angreifers**

Christian Biehler

### 4. MÄRZ 2021

// **Active Directory in Gefahr: Was Fehlkonfigurationen bewirken und wie man  
Angriffe entdeckt und verhindert**

Frank Ullly

// **Ohne Bullshit-Bingo: Windows-Sicherheit mit Bordmitteln**

Christian Biehler

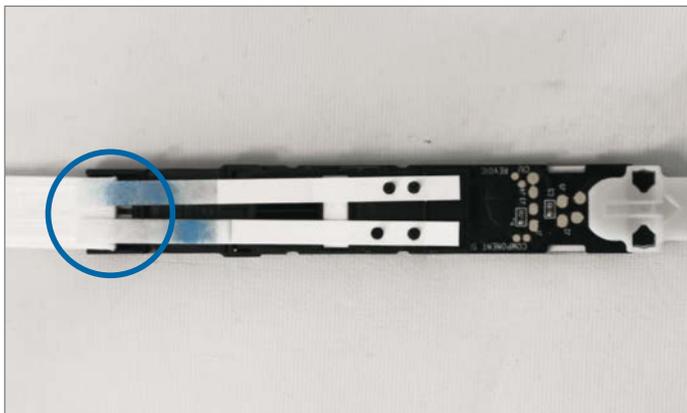
// **Panik und Schockstarre vermeiden: Richtig reagieren bei IT-Sicherheitsvorfällen**

Marco Lorenz

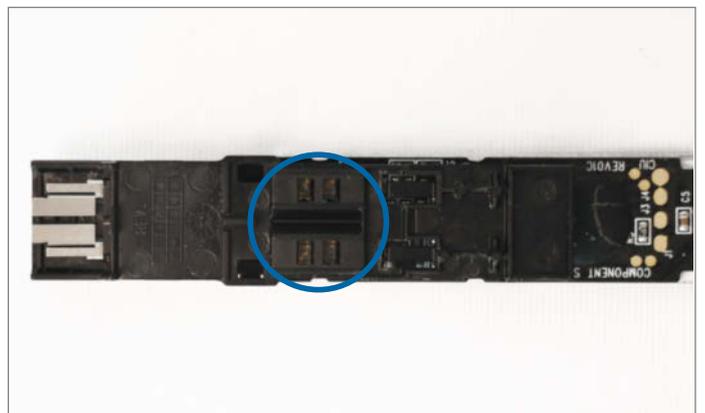
# Digitaler Schwangerschaftstest Clearblue

Herzlichen Glückwunsch, Sie sind schwanger! Was früher nur ein Arzt Frauen mitteilte, erledigt heute meist ein Schwangerschaftstest aus der Drogerie oder Apotheke. Inzwischen gibt es sogar elektronische Tester, die das Ergebnis im Klartext anzeigen. Was steckt drin in diesen Einweg-Geräten?

von Heinz Behling



1 Das Innere des digitalen Tests: Am Übergang zwischen dem Saugvlies (links) und den schmalen Teststreifen kann man die Elektroden zum Einschalten der Elektronik erkennen.



2 Hinter diesen vier Öffnungen liegen LEDs der Lichtschranken. Der Sensor für das reflektierte Licht befindet sich in der Mitte zwischen den Öffnungen unter der Kunststoffabdeckung.

Der Testablauf ist einfach: Das saugfähige Vlies am Ende des Kunststoff-Teststifts in den Urin halten, kurze Zeit warten und schon steht das Ergebnis fest (siehe Textkasten). Gegen mögliche Ablesefehler sollen elektronische moderne Ausführungen mit LC-Display helfen, auf dem dann *schwanger* oder *nicht schwanger* erscheint. Um herauszufinden, wie sie das machen, haben wir mal reingeschaut.

Auch in diesen Tests findet man die übliche Kombination aus Saugvlies und Teststreifen. Der elektrisch gut leitfähige Urin passiert auf dem Weg zu den Teststreifen zunächst zwei Elektroden und überbrückt diese. Das schaltet die von zwei Knopfzellen versorgte Elektronik ein 1.

Dann gelangt die Flüssigkeit in die Teststreifen mit den Antikörpern (siehe Kasten). Unter den Zonen mit den farbigen Anzeigestreifen liegen jedoch Reflexlichtschranken, die man von außen nicht sieht. 2.

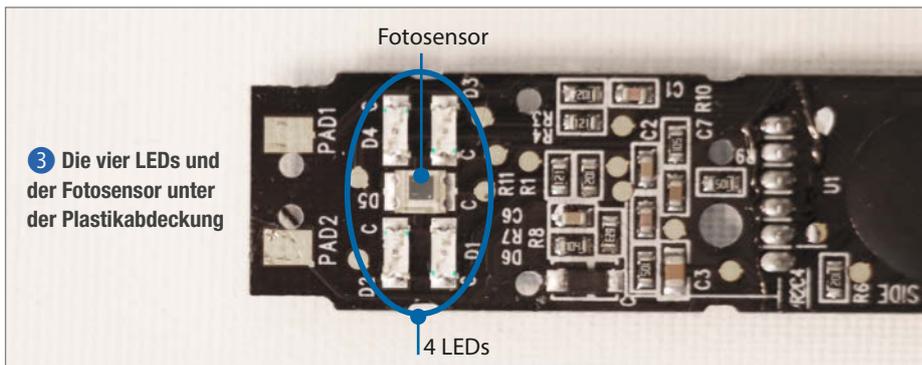
Die bestehen aus mehreren LEDs und einem gemeinsamen Fotosensor 3.

Die LEDs werden vom Controller nacheinander eingeschaltet und die je nach Färbung unterschiedlich große Lichtmenge wird vom Sensor registriert und vom Controller ausgewertet. Der zeigt das Ergebnis dann auf dem Display an. Auch die Kontrollzone B wird hierbei ausgewertet. Falls dort kein Farbwechsel registriert wird, zeigt dies das Display mit einer entsprechenden Fehlermeldung an. Daher ist eine Fehlablesung ausgeschlossen.

Dieser digitale Schwangerschaftstest enthält zwei Teststreifen. Deren Anzeigezonen brauchen unterschiedlich hohe hCG-Konzentrationen zum Farbwechsel. Da die Hormonkonzentration vor allem in den ersten drei Schwangerschaftswochen stark ansteigt, kann aus den Farbtensitäten der Testzonen berechnet werden, wie lange die Schwangerschaft bereits besteht, also ein bis zwei, zwei bis drei oder mehr als drei Wochen 4.

Bleibt noch die Frage, ob man mit dem Test anschließend noch etwas anfangen kann. Zunächst: Auch ein negativ verlaufender Test ermöglicht nicht, den Teststift nochmals als Schwangerschaftstest einzusetzen. Ansonsten sind höchstens die Knopfzellen noch wiederverwendbar. Der Rest ist reif für den Müll.

Aber wie wäre es, den Egoshooter *Doom* darauf zu spielen? Das geht in der Tat, ein Hacker namens *Foone* hat es gemacht. Allerdings hat er lediglich das Gehäuse wiederverwendet, Controller und Display mussten durch andere Typen ersetzt werden. Wer mehr über *Pregnancy Test Doom* wissen möchte, findet den Link am Ende des Artikels. —hgb



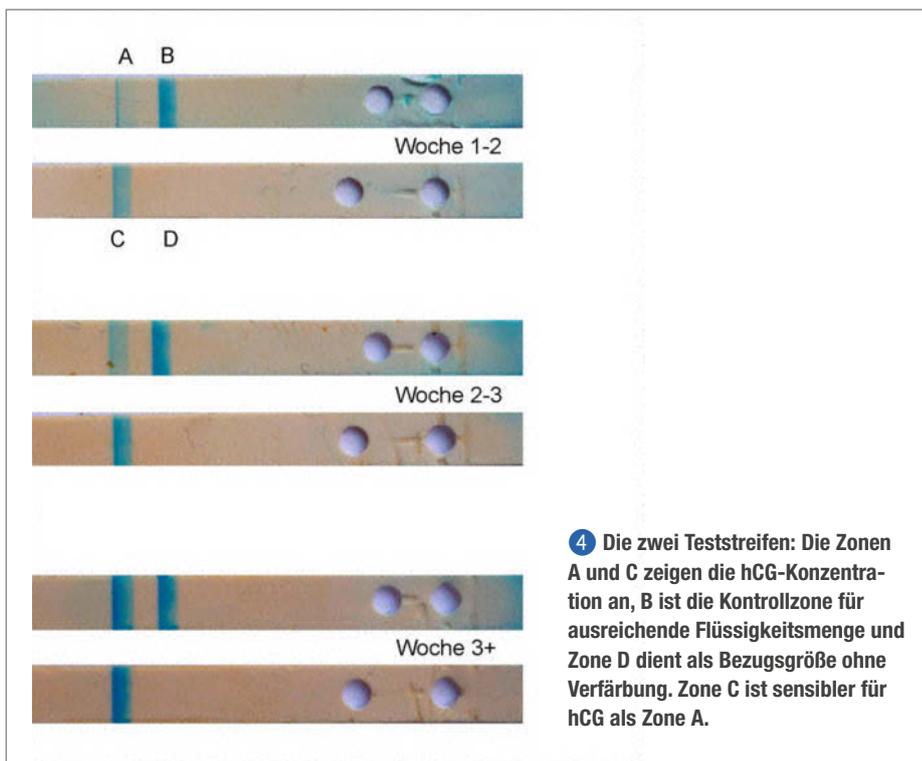
3 Die vier LEDs und der Fotosensor unter der Plastikabdeckung

### Wie funktioniert ein Schwangerschaftstest?

Im Urin einer Schwangeren ist die Konzentration des Hormons hCG (humanes Choriongonadotropin) bis zu 50.000 mal erhöht. Die Teststifte bestehen aus einem Saugvlies und einem daran anliegenden Teststreifen. Dort, wo der Streifen am Saugvlies anliegt, befinden sich Antikörper für hCG. Ist es vorhanden, binden die Antikörper daran. Sowohl der Hormon-Antikörper-Komplex als auch der nicht gebundene Rest der Antikörper wandern mit der Flüssigkeit in etwa 30 Sekunden durch den Teststreifen. In der Anzeigzone liegt auf deren Weg zunächst ein Streifen mit einem weiteren Antikörper, der nur auf den Hormon-Antikörper-Farbstoff-Komplex mit einer

Farbänderung reagiert und damit eine Schwangerschaft anzeigt. Noch etwas weiter folgt ein weiterer Antikörper, der ausschließlich auf ungebundene hCG-Antikörper mit Farbwechsel reagiert. Diese Kontrollzone zeigt an, dass eine ausreichend große Urinmenge aufgebracht wurde, und ist der einzige Farbstreifen, der bei einer Nichtschwangeren erscheint.

Wenn eine Schwangere eine zu geringe Urinmenge aufbringt, erreicht die Flüssigkeit nicht die Kontrollzone. Dann erscheint ebenfalls nur ein Farbstreifen, was oft fälschlicherweise als *nicht schwanger* gedeutet wird.



4 Die zwei Teststreifen: Die Zonen A und C zeigen die hCG-Konzentration an, B ist die Kontrollzone für ausreichende Flüssigkeitsmenge und Zone D dient als Bezugsgröße ohne Verfärbung. Zone C ist sensibler für hCG als Zone A.

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

# Modernisierung für den China-Laser

Der billige kleine Lasercutter aus China, meist K40 genannt, ist nicht sehr komfortabel: Die Software arbeitet nur mit Dongle-Unterstützung, die Intensitätseinstellung des Laserstrahls, Schneiden und Gravieren in einem Arbeitsgang und eine Netzwerkverbindung sind nicht möglich. Mit einer moderneren Steuerelektronik kann man das ändern.

von Heinz Behling



Für den Preis von etwa 320 Euro kann man bei einem Lasercutter mit 40W Strahlleistung zusätzlich nicht auch noch Bedienungs-komfort erwarten. Er funktioniert ja auch so. Doch wenn man ihn sehr gerne und häufig benutzt, wie ich das mache, wird das eine oder andere Manko doch lästig.

Da ist zum einen der Transport der Steuerdateien zum Cutter. Der läuft ausschließlich über eine USB-Leitung von einer speziell gepatchten Corel-Draw-Version (erfordert einen USB-Dongle) oder dem Open-Source-Programm K40-Whisperer (erfordert speziellen USB-Treiber) aus. Per Netzwerk oder USB-Stick ist das nicht möglich. Die USB-Verbindung des K40-Mainboards ist zudem recht störanfällig und wird oft während eines Schneidevorgangs unterbrochen. Das Material ist dann meist verdorben, weil der Cutter unterbrochene Arbeiten nicht fortsetzen kann. Speichern lassen sich die Dateien für eine spätere Wiederverwendung leider auch nicht auf dem Cutter.

Dazu kommt die umständliche Justierung der Laserleistung: Sie erfolgt durch Einstellen des Stromes durch die Laserröhre, entweder analog per Poti und Milliampere-Meter oder digital in Prozentwerten, was eine gehörige Portion Erfahrung braucht. Mit dem neuen Controller kann man einen einmal gefundenen Wert für ein Material in einer Datenbank speichern und sogar in der jeweiligen Schnittdatei eintragen, falls man dasselbe Objekt nochmal herstellen möchte. Beim Original-K40 geht das nicht. Eine Speicherung eines einmal gefundenen Optimalwertes in den Steuerdateien ist nicht möglich. Darin kann nur durch unterschiedliche Schnittgeschwindigkeit eine gewisse Leistungsanpassung vorgenommen werden.

Als letzter Punkt sei hier noch erwähnt, dass Schnitt- und Gravurkombinationen (zum Beispiel beim Ausschneiden und Beschriften einer Frontplatte) nicht in einem einzigen Arbeitsgang möglich sind. K40-Whisperer kann zwar anhand der Farben in den Steuergrafiken unterscheiden, ob etwas geschnitten oder graviert werden soll. Trotzdem müssen die Vorgänge jeweils separat durchgeführt werden.

Die Gründe für diese und weitere Einschränkungen (etwa die fehlende Möglichkeit, direkt am Cutter den Schneidekopf zu positionieren oder den Schnittpunkt festzulegen) liegen in der einfachen und damit billigen Steuerelektronik, die kaum mehr Leistung und Speicher als ein Arduino Uno bietet.

Tauscht man das elektronische Gehirn des kleinen Lasercutters jedoch gegen ein leistungsfähigeres Organ aus, dann blüht der K40 regelrecht auf. In China ist jetzt bereits ein entsprechend aufgerüstetes Nachfolgemodell des K40 erhältlich. Der Preis ist aber mit knapp 1000 Euro (zzgl. Fracht- und Zollkosten) recht happig. Das Controllerboard namens Ruida

## Kurzinfo

- » Komfortablere Steuerelektronik für China-Lasercutter K40
- » Bau eines Kabel-Adapters
- » Konfiguration des Controllers
- » Bedienung der Software

## Checkliste



**Zeitaufwand:**  
6 Stunden



**Kosten:**  
250 Euro bei vorhandenem Lasercutter

## Material

- » **Lasercontroller-Set Ruida RDC 6442S** (inkl. Display-/Tastaturmodul, Kabeln und Steckern)
- » **2 Schrittmotor-Treiber DM542**
- » **3m Litze 0,5mm<sup>2</sup>**
- » **0,5m<sup>2</sup> Plexiglas (3mm)**
- » **Schrauben und Muttern (M3)**
- » **Folienkabel-Buchse (I2polig, RM 1,25mm)**
- » **optional: Adapterplatine K40 MiddlemanV2A**

## Werkzeug

- » **Bohrmaschine mit Metall-Bohrer 3,5mm und 4mm**
- » **LötKolben**

## Mehr zum Thema

- » Mehr zum Lasercutter K40 lesen Sie im Artikel „Laser für Leser“ in Make 3/18 ab Seite 16.
- » Eine Bauanleitung für eine Abgasanlage gibt es online

Alles zum Artikel im Web unter [make-magazin.de/x9ma](http://make-magazin.de/x9ma)

RDS6442S gibt es jedoch auch einzeln zu kaufen. Der Einbau ist, nicht zuletzt wegen eines recht ausführlichen englischsprachigen Manuals, recht einfach.

Ich habe mich für diese Elektronik entschieden, nachdem ich damit Erfahrungen am großen Lasercutter in unserem Redaktionslabor

machen konnte. Der Controller wird als Set für etwa 200 Euro angeboten <sup>1</sup>. Es besteht aus dem in einem stabilen Kunststoffgehäuse eingebauten Mainboard, einem Bedienpanel mit Folientastatur und LC-Display, USB- und Netzwerkkabeln sowie den zum Anschluss erforderlichen Steckverbindern <sup>2</sup>. Bei der Bestell-

<sup>1</sup> **Achtung: Obwohl auf dem Bild Bedienpanel und Controller zu sehen sind, bestellen Sie für diesen Preis nur das Panel.**



2 Das Montagezubehör des Sets ist allein schon etwa 50 Euro wert.

lung (meist über AliExpress oder eBay) sollte man unbedingt darauf achten, das komplette Set zu bestellen. Die Komponenten werden nämlich auch einzeln angeboten und locken so mit einem vermeintlich sehr günstigen Preis.

### Elektronik mit Ausbau-Möglichkeiten

Das Mainbord hat Steckanschlüsse für bis zu vier Schrittmotoren. Das reicht nicht nur für die Ansteuerung der X- und Y-Achsen des Cutters, sondern bietet auch Reserven für die spätere Nachrüstung mit einem motorisch höhenverstellbaren Wabentisch inklusive Autofokus-Funktion und zur Ansteuerung einer Rotationseinheit zum Gravieren runder Objekte. Allerdings sind für die Schrittmotoren jeweils noch Stepperdriver (Typ DM542 oder ähnlich) notwendig, die den Signalen des Con-

trollers entsprechend die Spulen der Motoren ansteuern. Solche Bausteine kosten in Fernost weniger als 10 Euro.

Der Controller stellt außerdem Anschlüsse zum Schalten eines Airassist-Systems (zum Beispiel das in meinem Artikel in Make 3/18 selbstgebaute Minisystem) sowie zur Überwachung des Kühlwasserkreislaufs, des Deckel-Sicherheitsschalters sowie zur Ansteuerung von Warnlampen bei aktivem Laser bereit.

Neben einer USB-Buchse, die wie bisher zum Einspeisen der Steuerdaten benutzt werden kann, stehen auch eine Ethernet- sowie eine zweite USB-Buchse zum Anschluss eines USB-Speichermediums (Stick oder stromsparende SSD-Festplatten) zur Verfügung. Ein WLAN-Adapter ist als Zubehör lieferbar, jedoch genügt auch ein preiswerter WLAN-Repeater mit Ethernet-Buchse zur drahtlosen Verbindung.

Mein Plan war daher, den Ethernet- und den USB-Anschluss an die Stelle der alten USB-Buchse einzubauen, das einfache Original-Bedienfeld durch das Panel zu ersetzen und die restliche Elektronik unterhalb des Panels im Elektronikfach des Cutters unterzubringen. Dort war inzwischen auch wieder Platz, denn die dort früher einmal residierende Wasserkühlung habe ich inzwischen durch ein externes Kühlaggregat ersetzt.

Zum Einbau habe ich mir drei Platten aus 3mm-Plexiglas geschnitten: Die erste ist ein Rahmen für das Bedienpanel, die außerdem vier Schaltern (einer davon für den Cutter selbst) und der USB-Buchse für das Speichermedium Platz bietet. Die drei zusätzlichen Schalter können nach Belieben verwendet werden. Ich schalte damit die beiden Steckdosen an der Rückseite des Cutters, an der das Abluftgebläse sowie das Kühlaggregat angeschlossen sind. Der dritte kann in Serie mit dem Deckelschalter geschaltet werden und so den Laser bei Bedarf blockieren. Mehr Bedienelemente sind nicht notwendig.

Die zweite Platte nimmt das Controllergehäuse sowie die Stepperdriver auf. Außerdem hat sie Bohrungen zum Befestigen des Folienkabel-Adapters (siehe Kasten auf Seite 119). Sie wird später an der Trennwand zwischen Elektronik- und Schneideabteil des Cutters befestigt. Die dritte Platte nebst einem Abstandsrahmen hält die Ethernet- und die zweite USB-Buchse anstelle des Original-USB-Anschlusses. Die Schnittdateien für alle Platten stehen unter dem Kurzinfo-Link zum Download bereit. Achtung: Falls Sie die Platten benutzen möchten, schneiden Sie sie vor dem Zerlegen des Lasercutters!

### Ausschlachten

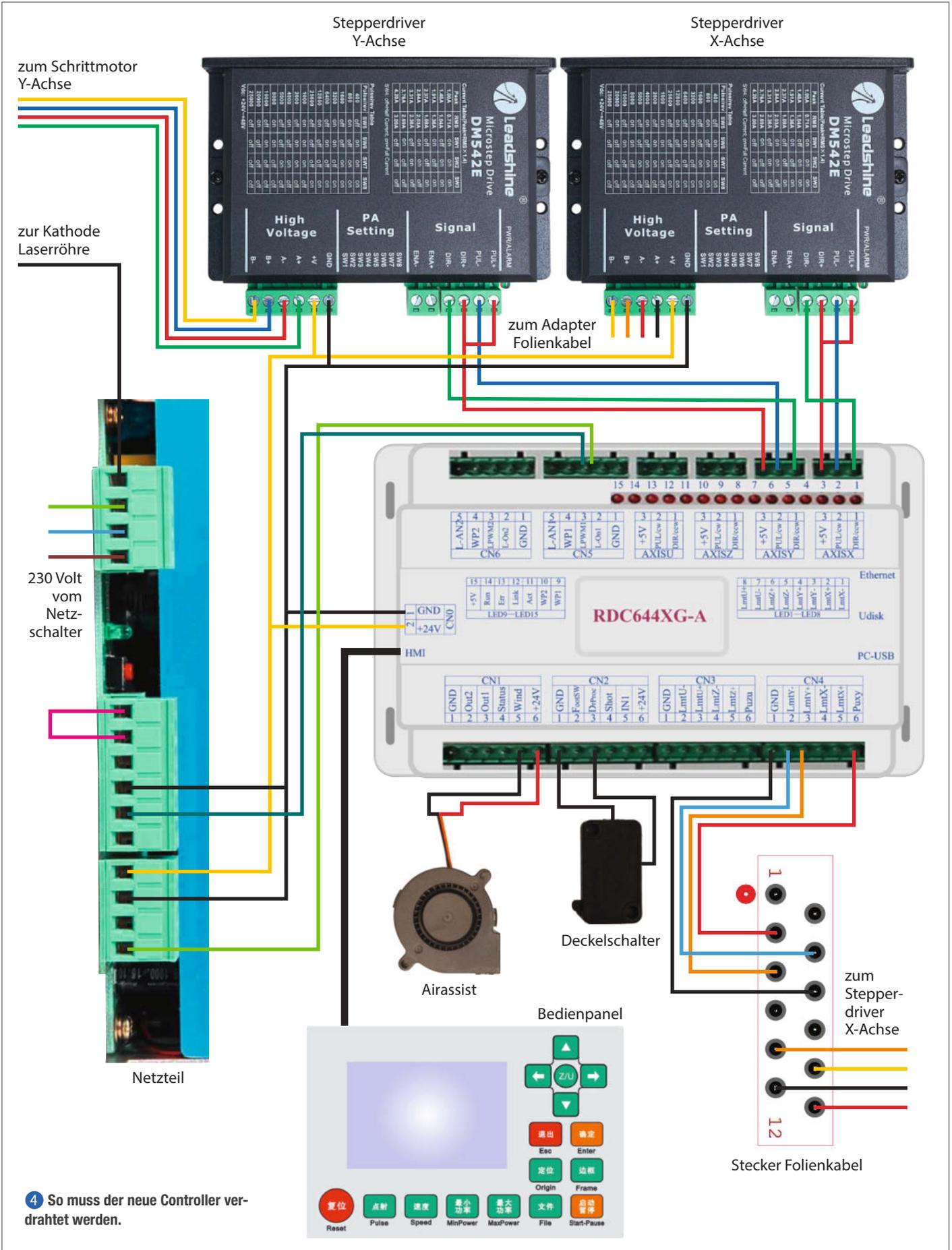
Nach dem Schneiden der Platten können Sie die nicht mehr benötigten Eingeweide des K40 ausbauen: Zunächst ziehen Sie den Netzstecker. Dann bauen Sie das komplette Bedienfeld und die Steuerplatine inklusive ihres Halblechs aus. Die schwarze Leitung, die bislang am Amperemeter hing (bzw. an der Platine der Digitalanzeige) verbinden Sie entsprechend dem Schaltplan mit dem Netzteil. Das ist der Kathodenanschluss der Laserröhre.

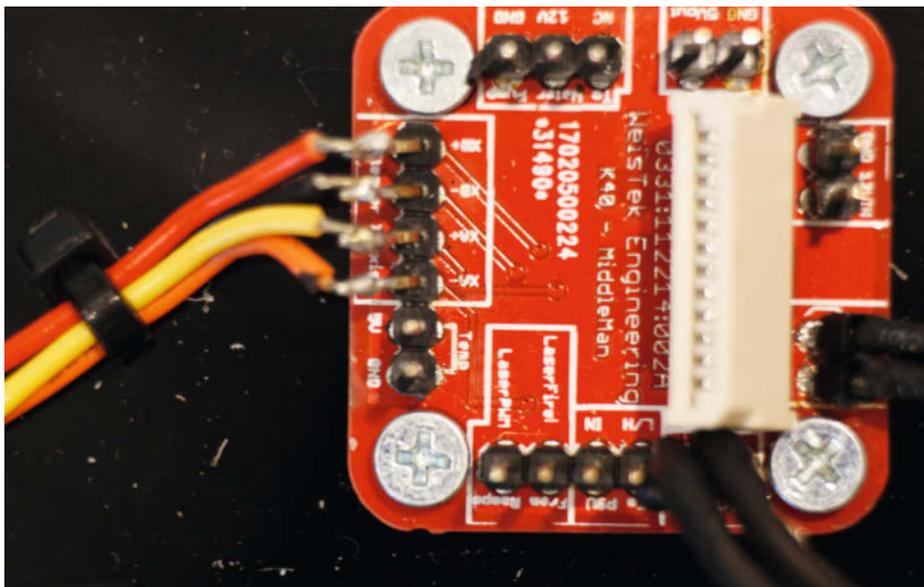
Dann wird das Bedienpanel mit vier M4-Schrauben in seine Platte eingebaut 3. Die Inbusköpfe sehen meiner Meinung nach übrigens besser aus als Kreuzschlitzschrauben, aber das ist Geschmacksache. Die vier Schalter werden nur eingeklipst. Für die USB-Buchse sind Schraubenlöcher (M3) vorhanden.

Das Controllergehäuse und die beiden Stepperdriver werden auf die zweite Platte geschraubt. Die Verkabelung entsprechend dem Schaltplan 4 sollte man sehr sorgfältig ausführen und mit Kabelbindern sichern,



3 Das Bedienpanel in seiner Plexiglasplatte





5 Die Middleman-Platine teilt die Leitungen des Folienkabels auf.

damit sich nichts durch die Vibrationen beim Arbeiten der Maschine lockern kann.

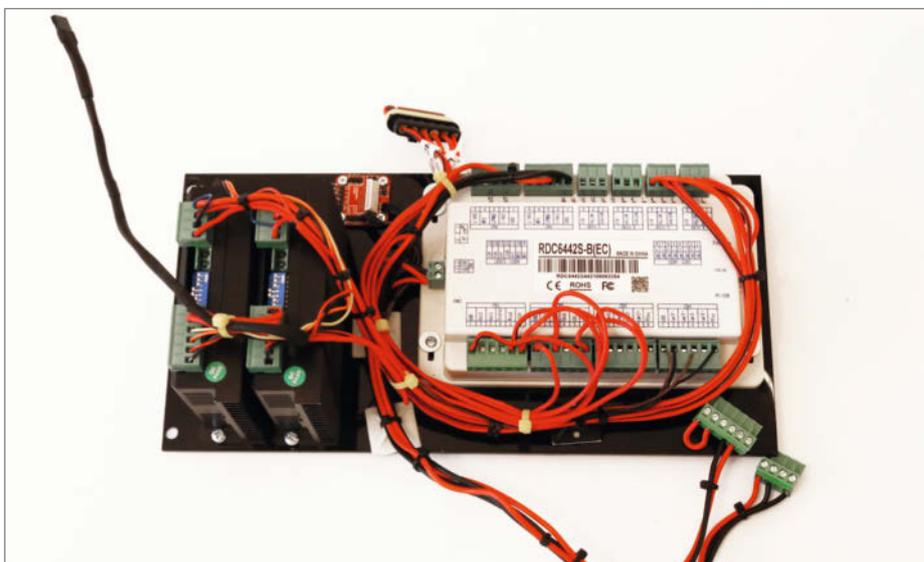
Zwei etwa 20cm lange Kabelbäumchen führen zum Netzteil und enden in den entsprechenden Steckern. Die Leitungen zum Deckelschalter und zum Airassist-Lüfter liegen an einem 6poligen KFZ-Steckverbinder an. Ein weiterer Kabelbaum mit einem 4poligen Dupont-Stecker dient zum Anschluss an den Schrittmotor der Y-Achse.

### Adapter-Selbstbau

Die Leitungen zum X-Motor sowie zu den beiden Endschaltern liegen an einem 12poligen Folienkabel an. Um das mit dem Controller zu verbinden, braucht man auf jeden Fall eine

entsprechende Buchse (Bezugsquelle siehe Kurzinfo-Link). Daran kann man entweder entsprechend dem Schaltplan direkt Litzen anlöten und zum Controller führen. Oder man benutzt eine kleine Adapterplatine (*Middleman* genannt) 5, in die man die Buchse einlötet und die die Leitungen dann an einzelnen beschrifteten Lötanschlüssen bereitstellt. Die Plexiglasplatte enthält bereits Löcher zur Schraubbefestigung dieser Platine.

Bei der Verkabelung der Schrittmotoren beachten Sie bitte die farbige Kennzeichnung im Schaltplan: Falls Sie dort Anschlüsse verwechseln, läuft der entsprechende Schrittmotor später verkehrt herum oder gar nicht. Zum Schluss erhält man eine leicht einbaubare Elektronik 6.



6 Die sauber verkabelte Elektronik: Der Stecker oben dient zum Anschluss des Deckelschalters und des Airassist-Lüfters.

### Einbau

Die Ethernet- und die zweite USB-Buchse werden mit den kleinen Plexiglasplatten an die Stelle der alten USB-Buchse eingebaut. Dazu sind vier 3,5mm-Löcher zu bohren 7. Die Elektronik-Platte schrauben Sie mit einer der ursprünglichen M5-Schrauben an die Seite des Elektronik-Abteils. Ich habe zur Befestigung des Controllergehäuses eine der vier Schrauben als 50mm lange M3-Schraube gewählt. In die Zwischenwand zwischen Elektronik- und Schnittabteil habe ich ein entsprechendes Loch gebohrt und die Platte dann mit einem Abstandhalter (30mm) zusätzlich mit dieser Schraube befestigt, da eine Schraube allein keine stabile Befestigung ergibt.

Für das Bedienpanel müssen vier 4,5mm Löcher in den Deckel des Elektronikteils gebohrt werden. Zum Anzeichnen kann man einfach die Plexiglasplatte auflegen. Anschließend die Löcher entgraten und die Platte gefühlvoll festschrauben (Vorsicht, Plexiglas bricht schnell bei Überlastung). Schließlich muss man nur noch die Stecker verbinden (zum Folienkabel, dem Y-Motor, Airassist, Bedienpanel, Netzwerk, USB). Vergessen Sie nicht die Leitungen am Netzschalter 8. Die Kontaktbelegung ist dieselbe wie beim Original-Schalter (der übrigens auch in die Öffnung passt). Das ist der einzige Punkt, wo Sie es direkt mit Netzspannung zu tun bekommen. In der restlichen Elektronik sind 24 Volt die höchste auftretende Spannung. Falls Sie sich Sorgen wegen der Hochspannung für die Laserröhre machen: Mit der kommen Sie bei diesem Umbau nicht in Berührung.

Der Hardware-Umbau ist damit fertig. Ich weiß, es kribbelt jetzt in den Fingerspitzen und Sie möchten den Cutter einschalten. Gedulden Sie sich aber noch etwas, denn zuerst sollten Sie die Software installieren. Mit der müssen Sie dann noch den Controller konfigurieren, damit er weiß, wie groß die Arbeitsfläche ist, wie die Motoren angesteuert werden und einiges mehr. Die Download-Adresse gibt es ebenfalls über den Kurzinfo-Link. Installieren Sie aus dem entpackten Archiv zunächst die USB-Treiber und danach die Software *RDWorks*. Als Sprache steht auch Deutsch zur Verfügung, allerdings ist die Übersetzung etwas holprig, jedoch durchaus brauchbar.

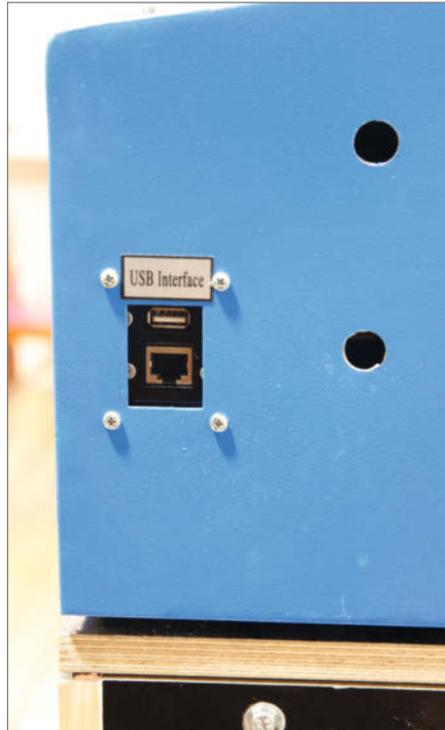
### Erster Kontakt

Verbinden Sie nun den Lasercutter per USB-Kabel mit dem PC oder schließen Sie ihn ans Stromnetz an und schalten Sie ihn ein. Wichtig: Tippen Sie am Lasercutter auf die ESC-Taste. Dadurch bricht der Cutter das Anfahren der Nullposition ab. Da er ja noch nichts über seine Maße und die Ansteuerung der Motoren und Schalter weiß, würde er andernfalls hemmungslos an die Anschläge rattern! Die Kon-

figurationsdateien stehen ebenfalls zum Download bereit. Sie übertragen diese mit Hilfe von *RDWorks* zum Lasercutter. Starten Sie daher das Programm.

Bevor Sie die Dateien zum Cutter senden können, müssen Sie zunächst einmal Kontakt mit ihm aufnehmen. Ist er per USB mit dem PC verbunden, geschieht dies automatisch (wenn Sie die Treiber installiert haben). Bei einer Netzwerk-Verbindung müssen Sie zunächst die IP-Adresse des K40 herausbekommen. Das geschieht am einfachsten am Cutter selbst: Drücken Sie die *Z/U*-Taste und wählen Sie mit den Pfeiltasten (nur *auf* und *ab*) den Punkt *IP-Konfig+*. Auf dem Display **9** sehen Sie nun die IP-Adresse, die der Cutter vom Netzwerk-Router erhalten hat.

In *RDWorks* klicken Sie dann unten rechts im Programmfenster auf *Settings* beziehungsweise *Rahmen* (erinnern Sie sich an meine Bemerkung über die holprige Übersetzung), falls Sie Deutsch als Sprache eingestellt haben. Klicken Sie anschließend auf *Add (Hinzugefügt)*, *Web* und geben Sie die IP-Adresse an. Nach einem hoffentlich erfolgreichen *Test* bestätigen Sie mit *OK*, setzen dann einen Haken vor die gerade hinzugefügte Adresse und beenden die Prozedur mit *Exit (Beenden)*.



**7** Netzwerk- und USB-Buchse sitzen gut zugänglich an der Seite.



**8** So aufgeräumt sieht zum Schluss das Elektronik-Abteil aus.



# Einrichtung eines modernen (Homeoffice-) Arbeitsplatzes



**06. – 07.05.2021, online**

Wie führe ich einen modernen Arbeitsplatz unter ganzheitlicher Betrachtung ein? Profitieren wir von einem modernen Arbeitsplatz automatisch, weil wir im Home-Office arbeiten?

In dieser Schulung zeigen wir Ihnen auf, was sich hinter dem Ansatz *New Work* verbirgt und welche Faktoren hierbei berücksichtigt werden sollten (Mensch, Ort und Technologie). Dies reicht von der Technologie und der Sicherheit der eingesetzten Systeme bis hin zum Ort und der Veränderung der Arbeitsweise der Mitarbeiter.

#### THEMENSCHWERPUNKTE:

- Was ist *New Work*?
- Arbeitsplatz gestern vs. Morgen
- Hardware / Software Assessment
- Setup einer Lösungsumgebung und erste Schritte

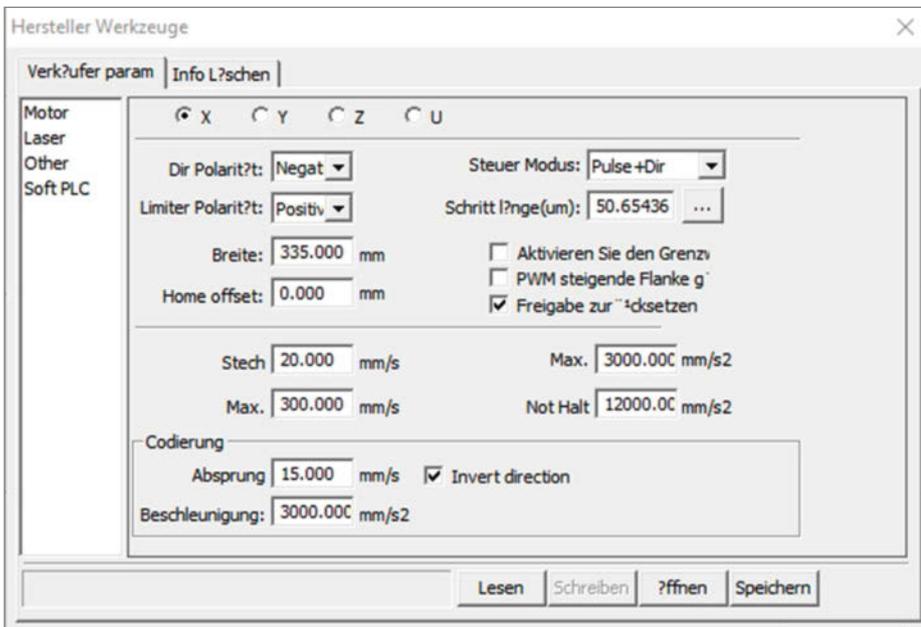


Trainer: Nico Ziegler

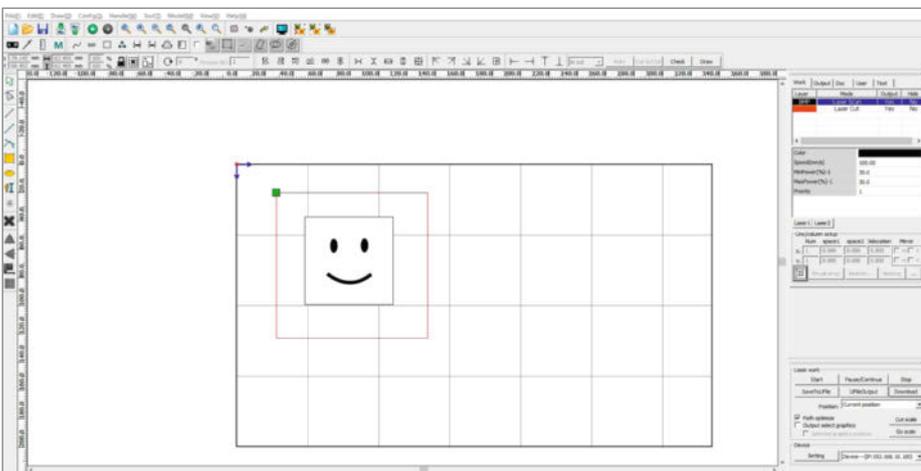
**Frühbucher-Preis  
bis  
8. April 2021**



9 Auf dem Display verrät der K40 seine Netzwerk-Adresse.



10 Bevor Sie Konfigurationseinstellungen ändern, müssen Sie zuerst auf Lesen klicken.



11 Augen und Mund des Smileys sollen graviert und dann das Rechteck um den Smiley herum ausgeschnitten werden.

## Konfiguration

Klicks auf *File (Datei)* und *Vendor Settings (Werkzeugeinstellungen)* bringen Sie ins Konfigurationsfenster. Vorher allerdings wird ein Passwort abgefragt, es lautet *rd8888*. Wichtig: Damit Sie hier wirksam Änderungen in der Maschine vornehmen können, müssen Sie zunächst die in der Maschine gespeicherten Einstellungen auslesen (10 (Klick auf *Read (Lesen)*)). Anschließend klicken Sie auf *Open (?ffnen)*, wie gesagt: holprig, wählen die zuvor überspielte Datei *K40\_Vendorsettings.RDVSet* aus und klicken auf *Öffnen*.

Die folgende Bestätigung nehmen Sie per Klick auf *OK* zur Kenntnis. Das Wichtigste: Erst ein Klick auf *Write (Schreiben)* schickt die Einstellungen zum K40. Mit *Exit (Beenden)* schließen Sie das Einstellungsfenster.

Das war aber nur die halbe Arbeit: Unter *Config (Konfig)* gehen Sie dann in die *System Settings (Systemeinstellungen)*. Hier geht es um die Werte, die *RDWorks* künftig berücksichtigen soll, also beispielsweise die Größe des angezeigten Schnittbereichs. Auch hierfür habe ich eine Datei vorbereitet. Klicken Sie auf *Import*, wählen Sie die Datei *K40\_Settings.cfg* und bestätigen Sie mit *Öffnen*. Achten Sie darauf, dass *Auto fresh page setting (Automatische Aktualisierung)* aktiv ist und schließen Sie das *Setting*-Fenster.

## Erster Arbeitseinsatz

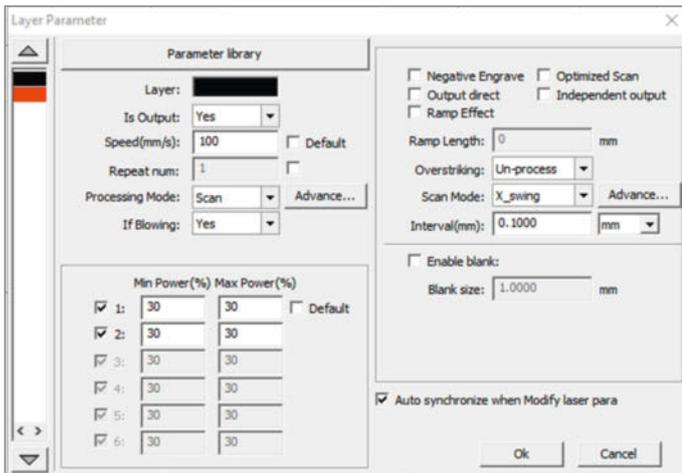
Nun ist Ihr Lasercutter einsatzbereit. Um zu zeigen, wie das funktioniert, habe ich in den Download-Dateien auch eine Schnittdatei hineingepackt (*Smiley.rld*). Sie besteht aus einem kleinen Bild, das graviert werden soll, und einem rechteckigen Schnitt, der das Bild aus dem Material ausschneidet. Laden Sie sie per *File (Datei)* und *Open (?ffnen)*. Die Datei ist einfach: Ein Smiley in der Mitte und ein Rechteck drumherum (11).

| Layer | Mode       | Output | Hide |
|-------|------------|--------|------|
| BMP   | Laser Scan | Yes    | No   |
|       | Laser Cut  | Yes    | No   |

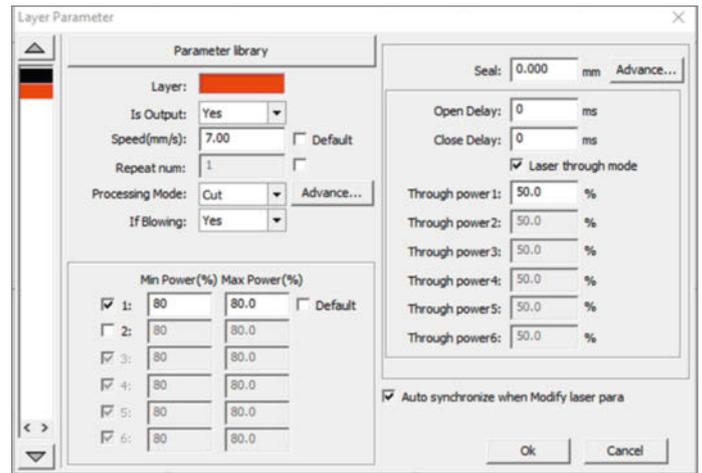
  

|                |  |        |
|----------------|--|--------|
| Color          |  |        |
| Speed(mm/s)    |  | 100.00 |
| MinPower(%) -1 |  | 30.0   |
| MaxPower(%) -1 |  | 30.0   |
| Priority       |  | 1      |

12 Die Work-Liste zeigt alle Gravier- und Schnittaufträge in der gerade aktuellen Schnittdatei.



13 Die Schnitt-/Gravierparameter stellen Sie in diesem Fenster ein.



14 Fürs Schneiden geben Sie eigene Parameter ein.

Der Smiley soll zuerst graviert und danach das Rechteck ausgeschnitten werden. Nun müssen Sie dem K40 mitteilen, mit welcher Geschwindigkeit und welcher Strahlintensität das geschehen soll. Rechts oben im *RDWorks*-Fenster sehen Sie eine kleine Liste 12.

Wenn Sie doppelt auf einer der Zeilen darin klicken (zum Beispiel auf den schwarzen Eintrag), öffnet sich das Fenster, in dem Sie die Einstellungen vornehmen können 13.

Ganz oben steht die Farbe des Layers: Das heißt, alle Einstellungen, die sie hier vornehmen, gelten für die Objekte in der Schnittdatei, die die gleiche Farbe (*schwarz*) haben. Achten Sie darauf, dass als *Processing mode* auch *Scan* gewählt ist, also Gravieren. Als Geschwindigkeit empfehle ich hier Werte von 100 bis 150mm/s. Die Laserleistung sollte zwischen 15 und 40 Prozent liegen, ist aber stark vom Material abhängig. Üben Sie hier mit Reststücken. Das *Yes* hinter *If Blowing* schaltet das Airassist ein. Mit einem Klick auf *OK* schließen Sie das Fenster. Wiederholen Sie die Prozedur dann mit dem zweiten Tabelleneintrag (rot).

Dort muss die Geschwindigkeit deutlich geringer sein: Für 3mm Sperrholz etwa 7mm/s bei einer Laserleistung von 80 bis 90 Prozent 14. Das ist aber sehr von der Holzart abhängig. Auch dieses Fenster schließen Sie per *OK*. Für den herkömmlichen K40 gibt es die Empfehlung, nicht mehr als 80 Prozent Leistung zu verwenden, um die Lebensdauer der Laseröhre zu verlängern. Ich habe in den Konfigurationsdateien die maximale Leistung bereits auf 80 Prozent begrenzt. Wählt man hier 100 Prozent, so entspricht das effektiv nur 80 Prozent Laserleistung.

Nun wird es spannend: Legen Sie das entsprechende Material in den K40 und klicken Sie auf *Start*. (Luftabzug und Kühlung nicht vergessen.) Der kleine Chinakracher legt los. Nach kurzer Zeit ist er fertig 15.

Übrigens: Falls Sie statt *Start* auf *Herunterladen* klicken und danach einen Dateinamen



15 Alles in einem Arbeitsgang

eingeben, wird die Datei im Lasercutter gespeichert. Voraussetzung dafür ist aber, das in der USB-Buchse im Bedienfeld ein Speichersstick steckt. Der dient dann als Festplattenersatz im K40. Später kann man dann mit der *File*-Taste am Bedienpanel eine Liste aller dort gespeicherten Dateien abrufen und mit den

Pfeiltasten und *Enter* eine zur Produktion auswählen.

Das war jetzt nur ein erster Anreißer über die neuen Fähigkeiten. Mehr gibt es demnächst online bei Ihrer Make. Viel Spaß beim Umbau und: Entdecke die Möglichkeiten! —hgb

## Adapterplatine Middleman

Diese kleinen Platine 5 ist bei keinem Händler mehr auf Lager. Es gibt jedoch einen Service in den USA, der sie auf Bestellung anfertigt; Mindestabnahme drei Stück für 6,95 Dollar. Hinzu kommen allerdings recht deftige Frachtkosten von 27 Dollar. So kommt man auf einen Stückpreis von etwa 11,30 Dollar. Der Zoll kann da in Deutschland dann nochmal etwas draufschlagen. Die Lieferung erfolgte etwa 14 Tage nach der Bestellung per Übernachts-Lieferung, daher der Preis. Außerdem brauchen Sie zur Platine noch zusätz-

lich den Stecker für das Folienkabel (Bezugsadresse siehe Kurzinfo-Link).

Es geht aber auch ohne die Platine, nur mit dem Stecker und etwas Lötarbeit (siehe Schaltplan). In der Redaktion haben wir etwa 100 dieser Stecker vorrätig. Falls Sie einen haben möchten, schicken Sie eine kurze E-Mail an [hgb@make-magazin.de](mailto:hgb@make-magazin.de). Sie erhalten dann Nachricht, ob Sie einen Stecker erhalten können und senden uns dann bitte einen frankierten und adressierten Rückumschlag.

# Wabentüren-Upcycling

Sie sind groß, erstaunlich leicht und landen bei Renovierungsarbeiten meist im Müllcontainer. Aber aus alten Wabentüren kannst du dein eigenes Sperrholz und noch einiges mehr herstellen – mit weniger Aufwand, als du denkst.

von Tim Sway (Übersetzung: Niq Oltman)



Neues aus Altem machen ist keine neue Idee, aber die gezielte Fertigung von Gegenständen aus wiedergewonnenen Materialien ist inzwischen schon so etwas wie ein Statement: Die Upcycling-Bewegung und das steigende Interesse an umweltverträglichen Gütern verschafft ehemals weggeworfenen Dingen einen neuen Wert. Heruntergekommene und baufällige Scheunen, die man früher einfach verfallen ließ, sind neuerdings gesuchte Fundgruben für Altes, aus dem Neues entstehen soll. Sogar Versandpaletten sind heute als Rohmaterial für Holzkonstruktionen en vogue. Und jetzt kommt noch meine neueste Leidenschaft dazu: Wabentüren **1**.

Jeder weiß, wovon ich rede: diese flachen, dünnen Türen, die vor allem Anfang bis Mitte der 70er richtig in Mode waren. Zu kleinem Preis zu haben, damals noch stylish, leichtgewichtig, überraschend langlebig – an sich konnte man sich gar nicht beschweren. Tja, die Zeiten ändern sich, und wenn heutzutage jemand die eigene Behausung renoviert, gehören Wabentüren häufig zu den ersten Dingen, die auf dem Müll landen (geschätzt zehntausende pro Jahr). Aber warte mal – daraus kann man doch noch so einiges machen! Ich habe mir vorgenommen, so viele dieser Türen wie möglich zu retten und ihnen einen neuen Nutzen zu geben. Hier also – in Kurzfassung – alles, was du über das Upcycling von Wabentüren wissen musst.

### Eine Anatomie der gemeinen Wabentür

Wabentüren bestehen üblicherweise aus zwei ca. 3mm starken Sperrholz-Furnierplatten (die beiden großen Außenflächen), montiert auf einem etwa 2,5cm dicken, massiven Holzrahmen, meistens aus Kiefer. Es gibt auch deutlich dickere Exemplare und je nach Jahrgang der Tür findet man auch andere Holzarten. Der Hohlraum zwischen den Platten enthält ein wabenartig strukturiertes Füllmaterial aus Karton oder Wellpappe.

Insgesamt ist die Tür meist ab 3cm dick. Der Holzrahmen ist an der Ober- und Unterkante je etwa 5cm tief (daher sägt man in echtes Holz, wenn man so eine Tür an eine Laibung anpassen muss) und an den Seiten etwa 4cm – zur Befestigung der Scharniere. Zusätzlich befindet sich an der Innenkante beider Rahmenseiten jeweils ein etwa 5cm breiter und ca. 30–45cm langer Balken, der zur Anbringung der Türgriffe dient (wahlweise auf der „linken“ oder „rechten“ Seite der Tür).

Einige Türen haben innen eine extra Stützstruktur aus dünnen Holzrippen oder Spanplattenstreifen mit Löchern zur Gewichtsparnis und und letztens habe ich eine Charge Türen entdeckt, deren Hohlräume mit Zugschnitten aus dickem Papprohr verstärkt waren.

## Kurzinfo

- » Wie Wabentüren aufgebaut sind
- » Wabentüren gekonnt zerlegen
- » Selbst Sperrholz verleimen

## Mehr zum Thema

- » Grace Dobush, Vom Abfall zum Einfall, Make 1/20, S. 64

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





### Achtung!

Beim Zersägen des Holzrahmens auf kleine Nägel und Heftklammern achten. Deine Tischkreissäge sollte damit kein großes Problem haben, aber vielleicht nimmst du dazu lieber ein schon etwas älteres oder weniger teures Sägeblatt. Unbedingt ordentliche Schutzausrüstung benutzen, etwa Handschuhe und Schutzbrille tragen!

### Wabentüren finden

Wenn man auf dem Sperrmüll nicht fündig wird, hält man am besten Ausschau nach Häusern aus den 1950ern bis -90ern, die gerade renoviert werden. Sofern da noch die Originaltüren verbaut sind, sind es mit hoher Wahrscheinlichkeit Wabentüren. In so einem Fall: einfach hingehen und fragen! Die meisten Handwerker überlassen einem die Türen gerne, da sie sie dann nicht selbst entsorgen müssen. Am besten einem/r Handwerker/in aus der Umgebung mitteilen, dass man solche Türen sucht – mitunter bekommt man dann immer mal wieder einen Anruf, wenn es neue (alte) „Ware“ abzuholen gibt.

Ich poste auch entsprechende Gesuche auf örtlichen Facebookgruppen und anderen sozialen Netzwerken. Auch meine Freunde und Familie habe ich gebeten, meine Anfragen weiterzureichen. Bisher hatte ich nie Schwierigkeiten, jemanden mit einem Keller voller Türen oder einem aufgeschobenen Renovierungsvorhaben zu finden. Einmal kam mein Freund Peter Kelly (von 542customwoodworks.com) von einem großen Renovierungsauftrag und ich konnte mir über 40 dieser Türen, eigentlich für die Müllkippe bestimmt, auf einen Schlag schnappen 2!

### Wabentüren zerlegen

Aus einer 75cm x 200cm großen Wabentür lassen sich zwei Sperrholzplatten in ca. 60cm x 190cm Größe gewinnen; dazu noch ein paar brauchbare Stücke Kiefernholz.

Die Türen sind mit so etwas Ähnlichem wie Heißkleber zusammengefügt und die Pappteile bekommt man auch leicht wieder ab, aber wenn man ungeduldig ist und versucht, auch das dünne Sperrholz direkt vom Holzrahmen abzureißen, geht das so gut wie immer daneben. Besser ist, einen Teil des Holzrahmens zu opfern, indem man diesen entlang der Außenkanten zersägt (aber keine Sorge; es bleibt noch genug davon übrig).

Zuerst die langen Seiten absägen (ca. 4cm vom Rand sollten ausreichen) 3 und beiseite legen. Ich nehme dazu meistens eine Tischkreissäge und zersäge die Tür in einem Durchgang, es geht aber auch mit einer Handkreissäge. Jetzt kann man innen die oberen und unteren Querbalken sowie die Balken für die Türgriffe sehen. Als Nächstes die oberen und unteren Querbalken absägen (am besten mit einer Handkreissäge und etwas Abdeckband, damit die Schnittkanten sauber bleiben) 4. Jetzt sind die Türgriff-Balken dran 5 – dazu am besten noch einmal ein je ca. 5cm breites

Stück von den langen Seiten der Tür absägen. An diesen Balken klebt dann jeweils vorne und hinten ein schmaler Streifen Sperrholz, den man abreißen und für kleinere Bauvorhaben aufbewahren kann 6.

Manchmal bekommt man beim Ablösen der Sperrholzschiene von den Türgriff-Balken gleich ein größeres Stück Sperrholz ab (ca. 70cm x 190 cm), aber dabei kann es auch zerreißen. Ich mache es meistens so, dass ich den einen Türgriff-Balken sauber absäge und es bei dem anderen darauf ankommen lasse, das Sperrholz großflächig abziehen.

Wenn man alle massiven Holzteile aus der Tür entfernt hat, kann man die beiden Sperrholzsichten auseinander ziehen 7. Dabei sollte man besser Handschuhe tragen, da die verklebten Kanten der Wellpappe ziemlich scharf sind. Anschließend nehme ich meistens einen Holzhammer, um die noch festhängenden Pappreste abzuklopfen 8 – mit einem Schaber geht es auch, aber das dauert mir zu lange.

Alternativ kann man die Tür auch aufrecht hinstellen und dann mit einem länglichen Stück Holz oder Metall, das genau in den Spalt zwischen den beiden Sperrholzsichten passt, die darin festgeklebte Wellpappe in einem Durchgang von beiden Seiten ablösen.



Das ist ein ganz gutes Fitnessstraining, aber dafür kann man beide Seiten in einem Schwung erledigen **9**.

Die Tür ist jetzt fertig zerlegt. Eine 75cm x 200cm große Tür sollte nun in folgenden Einzelteilen vor dir liegen: Kiefernholz in den ungefähren Größen 2,5cm x 4cm x 200cm (2x), 2,5cm x 5cm x 70cm (2x) und 2,5cm x 5cm x 45cm (2x, davon eines mit einem Loch für den Türgriff); zwei ca. 3mm dicke Sperrholzplatten in etwa 60cm x 190cm Größe, vier Sperrholzstreifen in ca. 5cm x 75cm und sehr wenig Abfall – mit Ausnahme der Wellpappe, die zum Recycling in die Papiermülltonne kann. Gute Arbeit!

### Direkte Weiternutzung

Das 3mm-Sperrholz macht sich super in meiner Werkstatt: Es eignet sich perfekt zum Herstellen von Schablonen, Abstandshaltern, Schubladen-Böden und mehr. Für fast alle meine Ideen baue ich die ersten Prototypen aus dem Tür-Sperrholz **10**. Das Material lässt sich auch prima im Lasercutter bearbeiten (und ist kostenlos) – auf diese Weise habe ich mir unter anderem eine komplette Akustikgitarre gebaut **11**!

### Selbstgemachtes Sperrholz

Durch schlichtes Aufeinanderleimen mehrerer Sperrholzplatten aus alten Türen kannst du dir eigenes Sperrholz in jeder gewünschten Stärke herstellen. Der einzige Haken ist, dass du für alles über circa 6mm Endstärke beide Seiten der einzelnen Sperrholzschnitten abschleifen musst, um alle Papierkleber-Reste von den Innenflächen zu entfernen. Das ist der lästigste Teil der Arbeit, aber wenn du mit 60er Schleifpapier anfängst, geht es eigentlich recht fix.

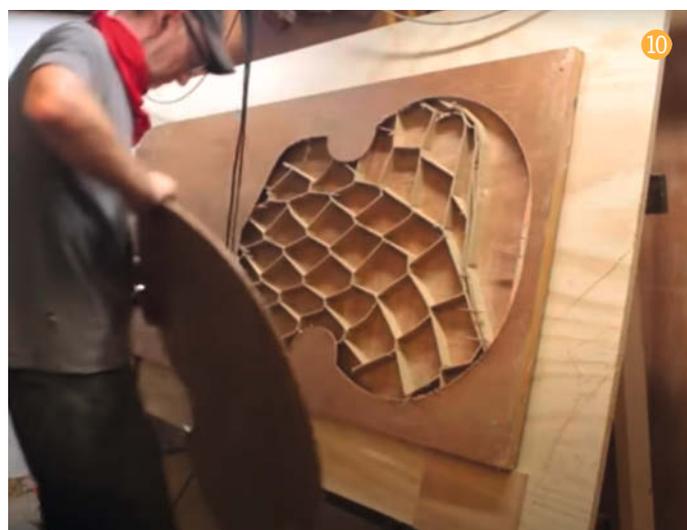
Den Leim großzügig auftragen und darauf achten, dass die Fläche vollständig mit Leim

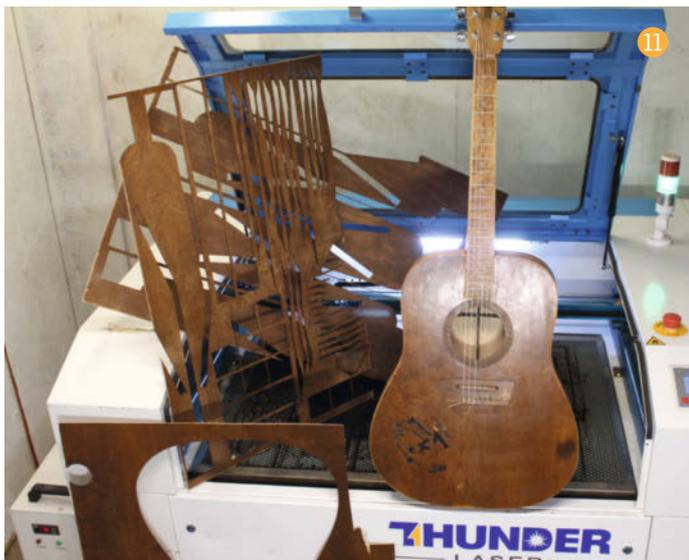


bedeckt ist **12**. Die Außenkanten halte ich mit jeder Menge Federklemmen zusammen; die Mitte beschwere ich mit so vielen Gewichten, wie ich finden kann **13**. Das Ergebnis war bisher immer zufriedenstellend und ohne „Blasen“ **14**.

### Laminieren

Warum sollte hier Schluss sein mit Leimen? Da ein Teil deiner „Beute“ ohnehin in kleineren Stücken vorliegt, spricht überhaupt nichts dagegen, aus dem Sperrholz gleich ganze laminierte Blöcke herzustellen **15**. Durch Zusam-





menfügen von Schichten in abwechselnden Richtungen zum Beispiel entstehen vielfältige Muster – hier kann man recht kreativ werden. Ich leime mir aus etlichen Schichten dünnem Sperrholz diverse Rohlinge zusammen, die ich dann in verschiedenen Winkeln zersäge und abschleife – so ergeben sich alle möglichen interessanten Wellenmuster. Aus dem resultierenden Material habe ich schon mehrere Gitarren und andere Gegenstände gebaut **16**.

### Noch mehr Nutzungsideen

Zwar geht es hier vor allem um das Sperrholz, aber aus dem restlichen Holz kann man ebenfalls noch einiges machen. Der Rahmen – einschließlich der Balken für die Türgriffe – besteht üblicherweise aus Douglasie oder einer anderen Kiefernart, zumeist von ziemlich

hoher Qualität, und ist obendrein – da es jahrzehntelang in einer Tür abgegangen wurde – ausgesprochen gerade und trocken. Im Baumarkt findest du wahrscheinlich kein Kiefernholz dieser Güte.

Mit der Tischsäge entferne ich manchmal das noch festklebende Sperrholz und bekomme ein paar hübsche Stücke rohes Massivholz heraus; ab und zu lasse ich das Sperrholz aber auch einfach dran, woraus sich ganz neue und einmalige Strukturen ergeben **17**.

Hoffentlich siehst du eine alte, ausranigierte Schrank- oder Zimmertür jetzt in einem neuen Licht und vielleicht fällt dir plötzlich ganz viel ein, das du daraus bauen könntest. Wabentüren stehen also jetzt auf unserer Upcycling-Materialliste; was ist als nächstes dran?  
—pek

## Make-Sonderheft: Loslegen mit Holz

Für viele Maker ist die Arbeit mit Holz ein ungewohntes Metier, weshalb wir uns im Make-Sonderheft „Loslegen mit Holz“ diesem Thema einmal ganz intensiv widmen. In dieser Sonderausgabe von Make erfahren Sie, welche Werkzeuge Ihnen den Einstieg bei minimalem Aufwand maximal erleichtern – und zwar unabhängig davon, ob Sie mit handlichen Maschinen schnell zum Ziel kommen wollen oder ob der Umgang mit Holz für Sie eine Rückkehr zum echten Handwerk sein soll. Von Lasercuttern und CNC-Fräsen ist natürlich ebenfalls die Rede, für alle, die lieber digital planen und die Maschine den Rest machen lassen wollen. Darüber hinaus gibt es ausführliche Schritt-für-Schritt-Anleitungen für Holz-

projekte, die von Werkstatthelfern wie einer einfachen Handkreissägeführung über eine klassische Sägebank samt Sägebock bis hin zu Hinguckern wie dem raffinierten Vollholz-Lautsprecher ohne Membran und Elektronik reichen, der nur durch den ausgesägten Klangkanal im Inneren die Musik vom hineingesteckten Smartphone verstärkt.

Das Make-Sonderheft „Loslegen mit Holz“ ist seit dem 5. November für 10,90 Euro im Zeitschriftenhandel oder im heise shop erhältlich (siehe Link in der Kurzinfo) – dort wahlweise in gedruckter Form oder als PDF zum Download. Make-Abonnenten haben das Heft bereits zugeschickt bekommen, denn es ist in ihrem Abo enthalten.



## Mitmachen und posten

Unter dem Hashtag #HollowCoreDoors-AreTheNewPallet findest du weitere Ideen; markiere auch deine eigenen Projekte damit! Auch mein Kanal unter youtube.com/TimSway ist voller Infos, darunter mehr als ein Dutzend Waben-türen-Projekte sowie ein umfangreiches Video über meine Türen-Zerle-gungs-Methode.



14



15



16



17

Es gibt **10** Arten von Menschen.

iX-Leser und die anderen.



Jetzt Mini-Abo testen:  
3 Hefte + Bluetooth-Tastatur  
nur 16,50 €

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



- [www.ix.de/testen](http://www.ix.de/testen)
- [leserservice@heise.de](mailto:leserservice@heise.de)
- 49 (0)541 800 09 120

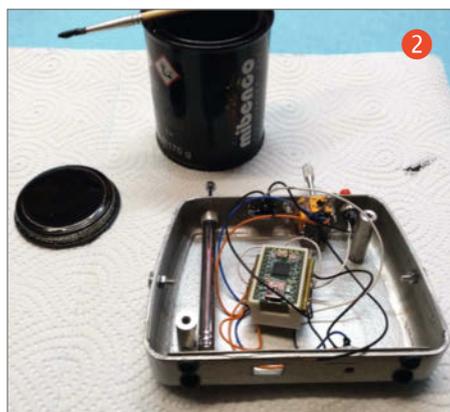
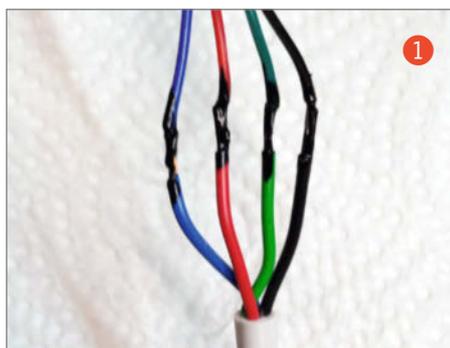
# Tipps & Tricks

Es klebt, isoliert, markiert und macht rutschfest: Flüssiges Gummi aus dem Farbtropf oder der Sprühdose – einfach zu verarbeiten und rückstandsfrei wieder zu entfernen. Nur die Trockenzeit braucht etwas Geduld.

von Robert Kränzlein

**F**lüssiggummi oder *Liquid Rubber* stammt eigentlich aus der Tuning-Szene und wird benutzt, um Felgen oder ganze Fahrzeuge statt mit einer Folie mit eben diesem Gummi zu beschichten. Ändert sich der Trend oder wechselt das Fahrzeug den Besitzer, lässt sich die Beschichtung rückstandslos entfernen. Erhältlich ist Flüssiggummi als Spraydose, in Großgebinden oder auch in kleinen Blechdosen mit rund 200g Füllmenge.

Für den Bastler bieten sich unzählige Möglichkeiten. Hauptanwendung ist die Isolierung von Lötstellen und Steckern. Wer hat nicht schon einmal vergessen, den Schrumpfschlauch über die Litze zu ziehen, bevor beide Enden verlötet wurden? Nachträglich geht da bei Schrumpfschlauch nichts mehr, mit Flüssiggummi immer **1**. In der Mini-Fernsteuerung auf Bild **2** sind etwa die Beinchen des Joysticks mit Flüssiggummi isoliert.



Bei Steckern entsteht durch Isolierung mit dem Material zudem eine Zugentlastung und es ist eine farbliche Markierung möglich, denn über ein Dutzend Farben in matt und glänzend sind im Handel **3**.

Flüssiggummi eignet sich auch für die luftdichte Verbindung von Bauteilen, so habe ich zum Beispiel den Trichter und den Deckel meines LoRa-Heizöltank-Sensors aus Make 6/20 (S. 66) damit verkittet.

Gegenstände können auch getaucht werden, um eine gummierte Oberfläche zu erzeugen – im Bild **4** sieht man etwa einen *Safegrabber* zum berührungslosen Öffnen von Türklinken mit Überzug aus einer Schicht Flüssiggummi. Dem Gehäuse eines China-Signalgenerators auf Bild **5** wurden hingegen in drei Durchgängen rutschhemmende Füßchen verpasst.

Generell eignet sich das Material zudem hervorragend zur Markierung des eigenen Werkzeugs – es gibt eigentlich keine Oberfläche, auf der es nicht haftet. Zum Auftragen ist mir ein Pinsel meist zu schade – Rührstäbchen eignen sich ebenso gut zum Auftupfen. Große Flächen taucht man am besten kurzerhand ein. Einer der diversen Hersteller des Materials bietet sein Gummi als *Liquid Tape* in einer Dose mit eingebautem Pinsel im Deckel an.

Ein wenig Zeit braucht die Trocknung: Je nach Schichtdicke sollte man dem Gummi schon zwischen zwei und acht Stunden Zeit lassen.

—pek



## 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 Autorenhonorar. 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)

# TECHNIKUNTERRICHT MACHT ENDLICH SPASS!



Spannende  
Unterrichts-  
materialien  
GRATIS

## Make: *Education*

Mit **Make Education** erhalten Sie jeden Monat kostenlose Bauberichte und Schritt-für-Schritt-Anleitungen für einen praxisorientierten Unterricht:



Für alle weiterführenden  
Schulen



Digital zum Downloaden



Fächerübergreifend



Monatlicher Newsletter

Jetzt kostenlos downloaden: [make-magazin.de/education](https://make-magazin.de/education)

© Copyright by Maker Media GmbH.

# 3D-Entwurf mit FreeCAD

Diesmal bauen wir mit FreeCAD einen Einhandzirkel, wie er früher in der Seefahrt unentbehrlich war und sich heute auch in der Werkstatt nützlich macht. Darüber hinaus lernt man bei seiner Konstruktion Schritt für Schritt, wie sich mit der kostenlosen Software komplexe mechanische Bauteile präzise in 3D entwerfen lassen.

von Matthias Mett

Unser Einhand-Stechzirkel aus dem 3D-Drucker in traditioneller Umgebung – auf einer Seekarte. Um auf der Entfernungen zu messen, stellt man den Zirkel etwa auf den Abstand zwischen zwei Schiffsorten ein, die man per Kreuzpeilung von Landmarken ermittelt hat und legt anschließend den Zirkel auf Höhe der Messung an die Skala links oder rechts der Karte an, um die Entfernung abzulesen (eine Bogenminute der Breitengrade entspricht einer Seemeile). Das liefert präzise Ergebnisse, obwohl die winkeltreue Mercator-Projektion von Seekarten dazu führt, dass deren Maßstab in Nord-Süd-Richtung variiert.

In vergangenen Make-Ausgaben haben wir mit den relativ einfachen Beispielprojekten eines Rings und eines Stempels erste Einblicke in die Arbeit mit der Konstruktionssoftware FreeCAD gegeben. Diesmal steht eine komplexere 3D-Aufgabe auf dem Programm, denn wir konstruieren einen speziellen Zirkel, der sich mit einer Hand mithilfe der oben gebogenen Schenkel bedienen lässt. Gewöhnliche Zirkel mit geradem Schenkel lassen sich mit einer Hand zwar schließen, aber nicht mehr einfach öffnen. Solche Einhandzirkel benutzt man traditionell in der Seefahrt, um Strecken auf der Seekarte abzumessen. Daher nennt man diese Art von Zirkel auch Marine- oder Navigationszirkel. So ein Zirkel ist aber auch in der Werkstatt praktisch, etwa zum Anreißen oder um direkt Maße von einem Werkstück abzunehmen und auf ein anderes zu übertragen.

Unser Zirkel besteht aus zwei identischen Metallschenkeln, die mittels einer Rändelschraube zu einem Stechzirkel mit zwei Spitzen verbunden werden. Eine Zirkelseite haben wir in einem zweiten Schritt noch so modifiziert, dass man dort einen Bleistift einspannen kann, mit dem sich dann auch Kreise zeichnen lassen. Diese Variante beschreibt ein eigener Online-Artikel, zu erreichen über den Link in der Kurzinfor.

Falls das Ihr erstes Projekt mit dieser Software ist: Bei FreeCAD handelt es sich um eine kostenlose Open-Source-Software zum Erstellen von dreidimensionalen Modellen. Die Software lässt sich über die Webseite [freecadweb.org](http://freecadweb.org) für die Betriebssysteme Windows, macOS und Linux herunterladen. Nachdem Sie das Programm installiert haben, beginnen Sie über das Dashboard ein neues Dokument. Speichern Sie es zuerst unter *Datei* und *Speichern unter* mit dem Namen „Zirkel“ ab. Auf der linken Seite des Programms befindet sich eine Baumansicht, in der die Dokumentenstruktur dargestellt ist. Sobald man ein neues Objekt erstellt, findet es sich dort wieder. Klickt man es an, öffnet sich unterhalb der Baumansicht das Eigenschaftenfenster, in dem sich die Parameter des Objektes anpassen lassen.

## Querschnitte markieren

Jeder der beiden identischen Schenkel unseres Zirkels weist verschiedene Querschnitte auf: Er fängt mit der Spitze an, die in einen dreieckigen Abschnitt übergeht, der wiederum zum Ende hin zum viereckigen Querschnitt wechselt. Diese Querschnitte zeichnen Sie im Folgenden auf mehrere Bezugsebenen, woraus sich anschließend 3D-Objekte erstellen lassen, die automatisch für die passenden Übergänge zwischen den Querschnitten sorgen.

Für die richtige Positionierung der Bezugsebenen benötigt man vorab mehrere Hilfslinien. Wechseln Sie dazu in der obersten Sym-

## Kurzinfor

- » Konstruieren mechanischer Teile mit der freien Software FreeCAD in 3D
- » Komplexe Formen über 2D-Querschnitte definieren
- » Symmetrien geschickt nutzen, um den Konstruktionsaufwand zu minimieren

### Checkliste



**Zeitaufwand:**  
etwa zwei Stunden



**Kosten:**  
je nach Material und Drucker zwischen 5 und 120 Euro



**Lieferdauer:**  
bei Bestellung bis zu 3–4 Wochen



**Konstruieren:**  
CAD, aber keine Vorkenntnisse nötig



**Computer:**  
Desktop oder Notebook mit Windows, macOS oder Linux

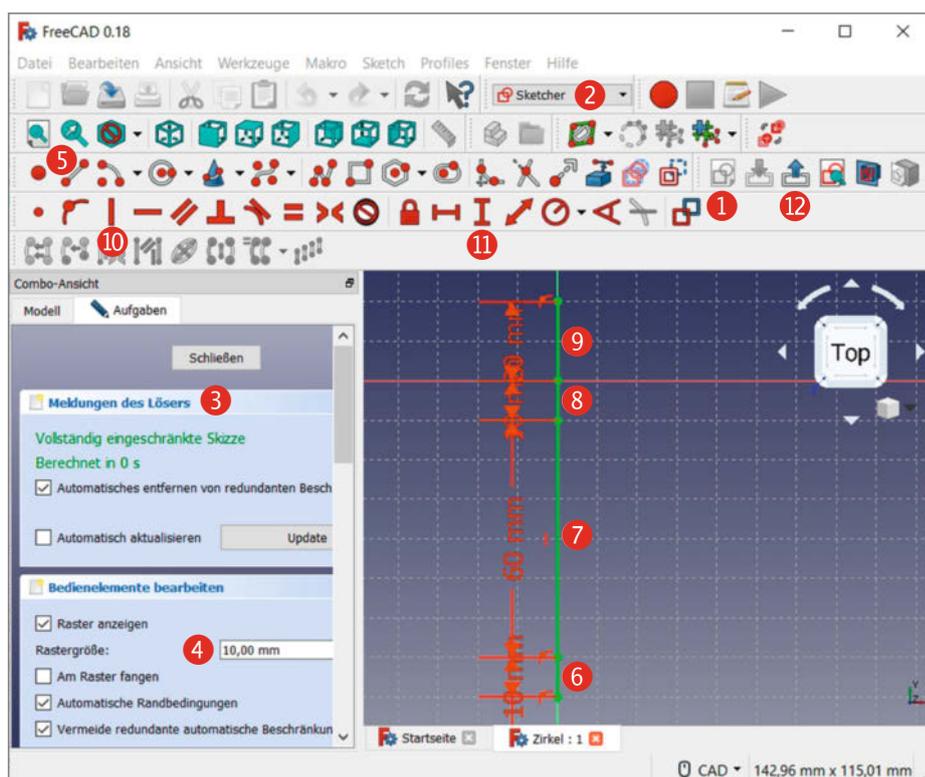


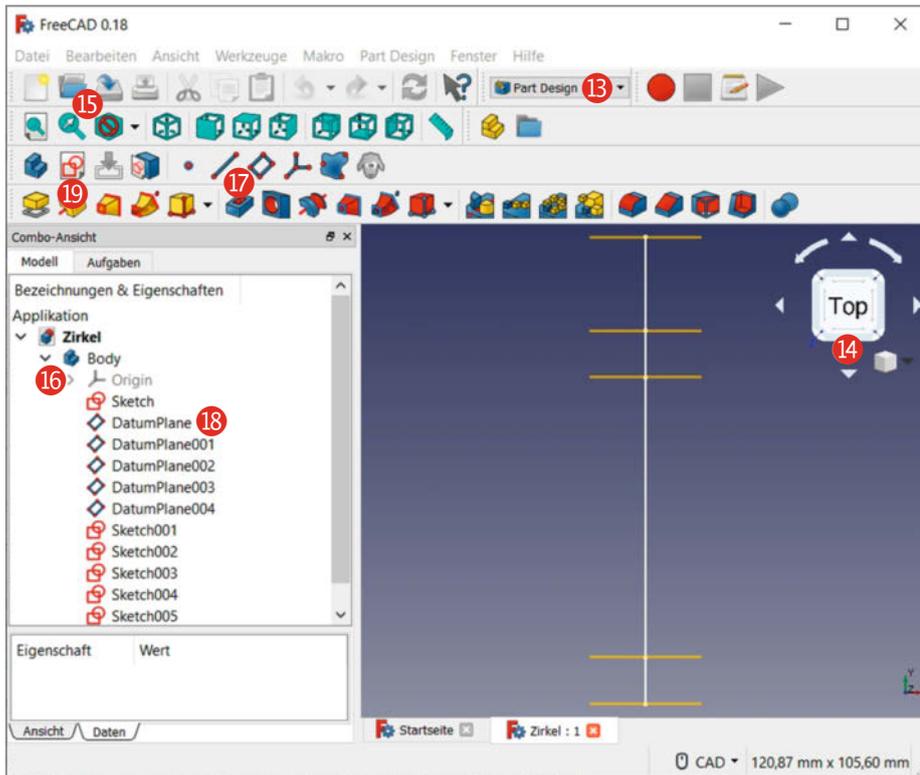
**3D-Druck:**  
mit eigenem Drucker, im Fablab oder beim Dienstleister

### Mehr zum Thema

- » Matthias Mett, Stempeln mit FreeCAD, Make 4/20, S.106
- » Matthias Mett, Ring frei für FreeCAD, Make 2/20, S. 114
- » Peter König, Gratis-CAD für Maker, Make 4/18, S. 19

Alles zum Artikel im Web unter [make-magazin.de/xh8v](http://make-magazin.de/xh8v)





Wählen Sie den Button **Neue Skizze erstellen** (1) und markieren Sie im Aufgabenbereich die Ebene *XY\_Plane (Basis-Ebene)*, die Sie mit **OK** bestätigen, woraufhin FreeCAD in den Arbeits-

bereich *Sketcher* (2) wechselt. Auf der linken Seite in den *Aufgaben* haken Sie unter *Meldungen des Löses* (3) die beiden Punkte *Automatisches Entfernen von redundanten Beschränkungen* und *Automatisch aktualisieren* an. Unter dem Punkt *Bedienelemente bearbeiten*

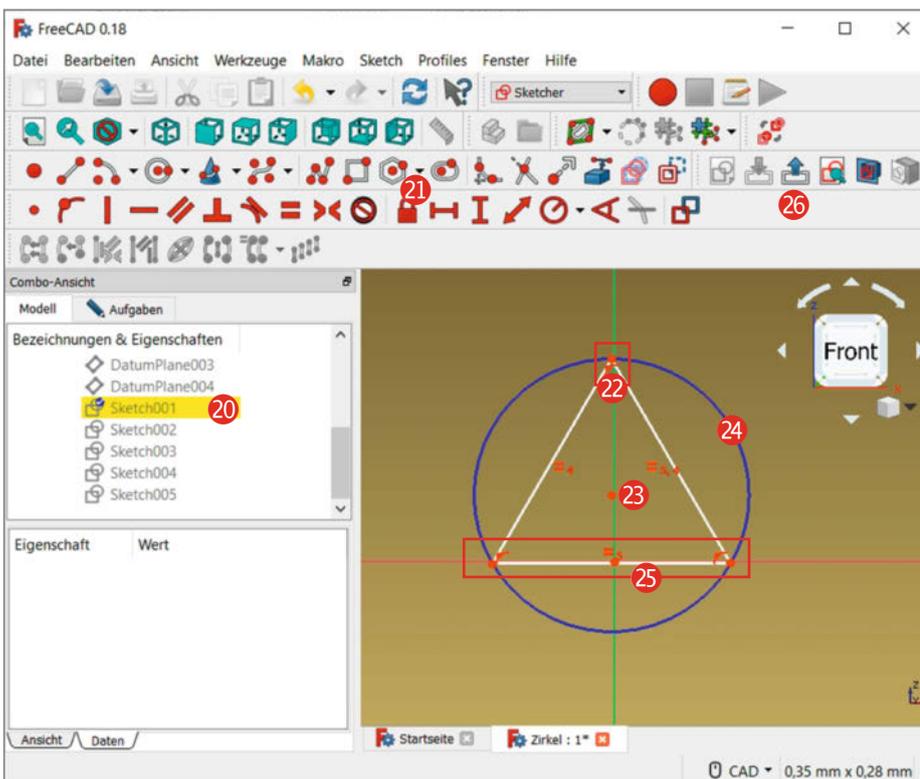
lassen Sie sich das *Raster anzeigen* und stellen die Rastergröße auf 10mm (4).

Zählen Sie im Rasterbereich vom rot markierten Mittelpunkt aus 8 Kästchen – also 80mm – auf der vertikalen Linie nach unten (unter Umständen müssen Sie dazu über das Mausrad noch etwas herauszoomen; die Ansicht verschieben Sie mit der Maus, indem Sie gleichzeitig **STRG** und die rechte Maustaste gedrückt halten). Zeichnen Sie von gefundenen Punkt aus mit dem *Linienwerkzeug* (5) eine Linie etwa ein Kästchen weit (also ein Zentimeter lang) nach oben (6). Die Linie müssen Sie nicht allzu genau zeichnen. Die endgültigen Maße legen *Einschränkungen*, auch *Constraints* genannt fest, die wir gleich hinzufügen.

Zeichnen Sie eine weitere Linie vom oberen Endpunkt der vorherigen Linie ausgehend 6 Kästchen (6cm) nach oben in Richtung des Mittelpunktes des Fadenkreuzes (7) und achten Sie dabei (und bei allen folgenden Linien) darauf, dass Sie genau vom Endpunkt der vorigen Linie los zeichnen (daran zu erkennen, dass der vorhandene Punkt sich grün färbt). Vom letzten Endpunkt ausgehend zeichnen Sie eine weitere Linie 1cm lang genau zum Mittelpunkt des Achsenkreuzes (8). Fügen Sie eine weitere Linie vom Mittelpunkt aus 2cm nach oben hinzu (9).

Mit *Einschränkungen* oder *Constraints* legen Sie jetzt die Lage und die Länge der Linien fest. Mit dem Werkzeug *Eine vertikale Beschränkung für das gewählte Element setzen* (10) richten Sie die Linien vertikal aus, indem Sie nach einem Klick auf den Button alle Linien nacheinander anklicken. Eventuell hat FreeCAD beim Zeichnen schon automatisch diese *Einschränkung* hinzugefügt. Erkennen können Sie es daran, dass ein kleiner vertikaler roter Strich links neben der Linie sitzt.

Mit dem Button *Den vertikalen Abstand zwischen zwei Punkten oder Streckenenden festlegen* (11) bestimmen Sie die Länge der einzelnen Linien. Beginnen Sie mit der untersten Linie, die Sie auf 10mm festlegen. Legen Sie die nächste obere Linie mit 60mm fest sowie die darüber liegende mit 10mm und die oberste mit 20mm. Verlassen Sie den *Sketcher*-Modus mit dem Button *Bearbeitung der Skizze beenden* (12).



### Bezugsebenen und Skizzen anlegen

Wieder im Arbeitsbereich *Part Design* (13) angeht, wechseln Sie mit dem *Navigation-Cube* (14) in die Draufsicht durch einen Klick auf *Top*. Passen Sie die Linie in das Fenster ein, indem Sie mit der Tastenkombination **STRG+A** alles markieren und die Auswahl mit dem Button *Selektierte Objekte auf dem Bildschirm einpassen* (15) an den Bildschirmausschnitt anpassen.

Nun erstellen Sie die *Bezugsebenen* mit den richtigen Abständen anhand der End-

punkte der Hilfslinien. Beginnen Sie mit der ersten Bezugsebene, indem Sie unterhalb des Objektes *Body* bei *Origin* auf den Pfeil 16 klicken und dann die Ebene *XZ\_Plane* markieren. Betätigen Sie nun den Button *Erstellung einer neuen Bezugsebene* 17 und wählen Sie den untersten Punkt der ersten Linie aus. Bestätigen Sie dann im Aufgabenbereich die Ebene mit dem Button *OK*.

Im Aufgabenbereich ist ein Objekt *DatumPlane* 18 hinzugekommen, wobei eine kurze horizontale Linie am Ende der untersten vertikalen Linien erscheint. Erstellen Sie auf gleiche Weise für alle Schnittpunkte und Endpunkte solch eine Bezugsebene, bis fünf Objekte mit dem Namen *Datum Plane* vorhanden sind. Achtung: Sie müssen für jede Ebene einzeln wieder *XZ\_Plane* auswählen!

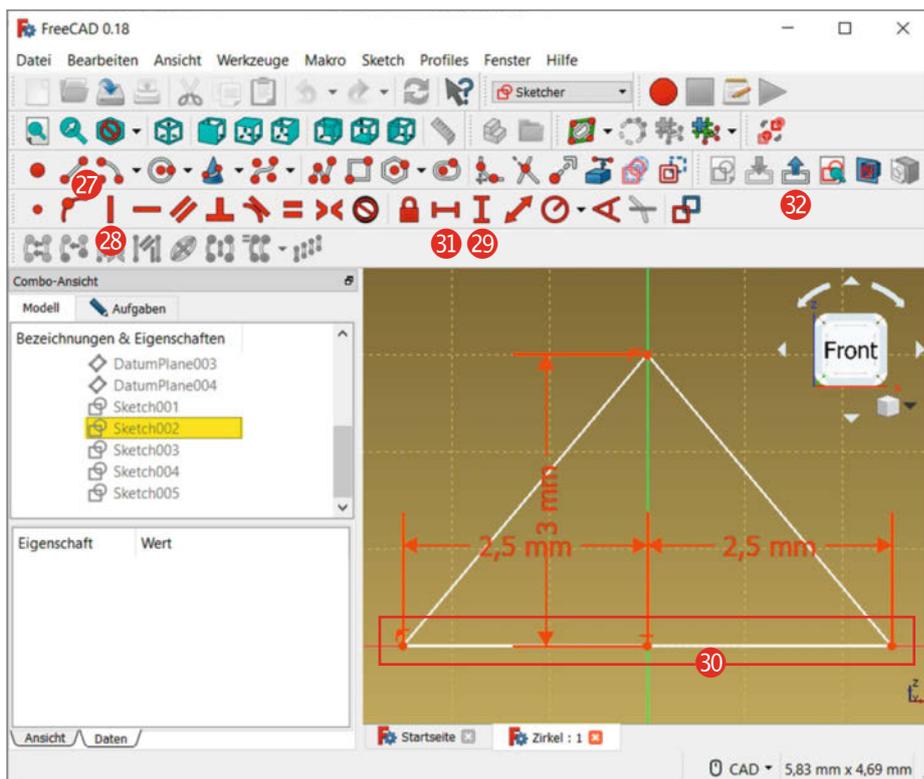
Für die einzelnen Bezugsebenen erstellen Sie nun Skizzen, auf denen Sie zeichnen können. Markieren Sie dazu entweder in der Zeichnung oder in der Objektansicht die unterste Bezugsebene und erstellen Sie eine neue Skizze mit dem Button *Neue Skizze erstellen* 19. Schließen Sie die Eigenschaften der Skizze in der Aufgabenansicht direkt wieder. Gehen Sie von unten nach oben weiter vor und erstellen Sie für jede Bezugsebene eine neue Skizze.

### Dreieck als Spitze

Erstellen Sie nun die Spitze des Zirkels. Wechseln Sie dazu in den untersten Zeichenbereich, indem Sie auf *Sketch001* 20 doppelklicken. Zoomen Sie mit der mittleren Maustaste oder im Menü unter *Ansicht* und *Zoom* weiter heran, sodass der Zeichenbereich den Bildschirm ausfüllt. Falls die Skizze aus dem Bildschirmbereich herausgeschoben ist, können Sie ihn mit der Steuerungstaste und der rechten Maustaste verschieben.

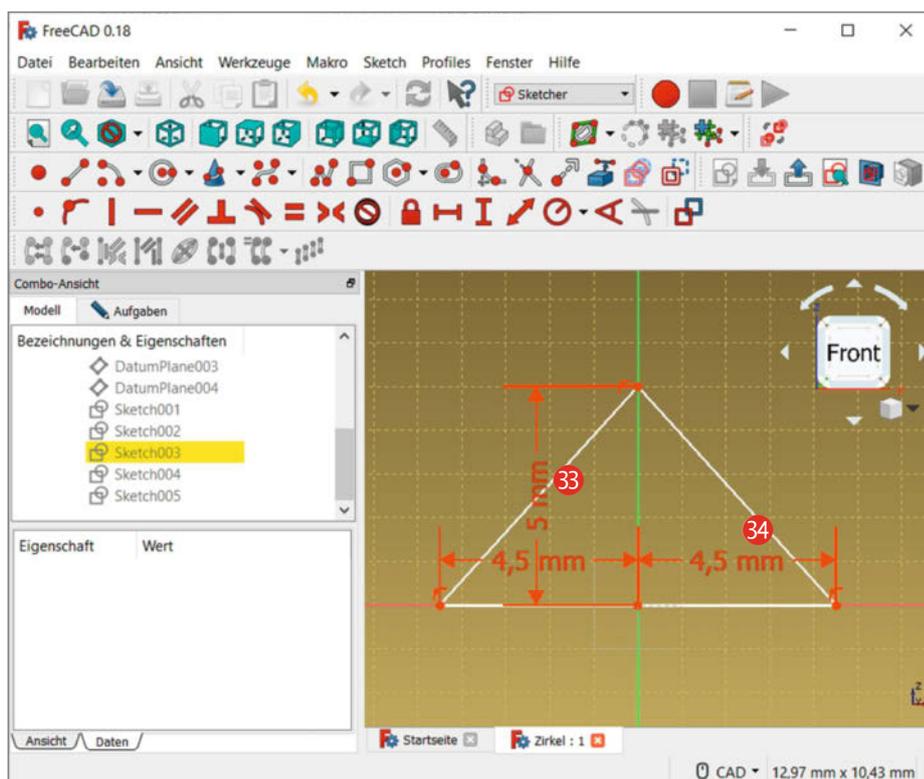
Die Zirkelspitze lässt sich nicht aus einem einzelnen Punkt als 3D-Objekt extrahieren, daher zeichnen Sie ein möglichst kleines Dreieck. Dazu verwenden Sie das Werkzeug *Dreieck*, das Sie durch Klick auf den Pfeil nach unten neben dem Button für das Polygon 21 zum Vorschein bringen.

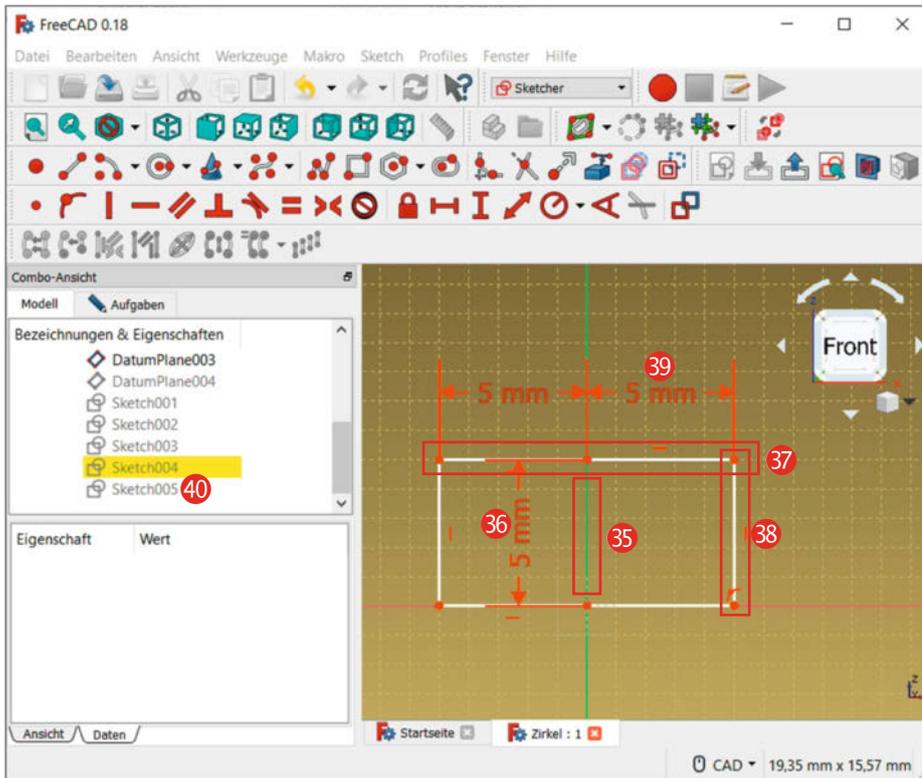
Setzen Sie mit dem ersten Klick den Mittelpunkt des Dreiecks auf die senkrechte Mittelachse, aber etwas oberhalb des Kreuzungspunkts mit der horizontalen Achse. Deaktivieren Sie das Dreieckswerkzeug, indem Sie rechts auf den Hintergrund der Skizze klicken. Richten Sie dann den obersten Punkt des Dreiecks an der vertikalen Linie aus, indem Sie diesen anklicken und mit gedrückter Maustaste entsprechend verschieben 22. Falls sich der Mittelpunkt des Dreiecks 23 dabei verschoben haben sollte, schieben Sie diesen wieder in die Mitte der Hilfslinien. Fassen Sie das Dreieck am blau angezeigten Umkreis 24 und skalieren Sie



es kleiner, bis bei den eingeblendeten Maßen (0.1, -0.0) angezeigt wird. Fassen Sie das Dreieck an der unteren horizontalen Linie und schieben diese auf die horizontale Hilfslinie 25. Der oberste Punkt des Dreiecks sollte weiter auf der vertikalen Hilfslinie liegen.

Verlassen Sie die Skizze nun mit dem Button *Bearbeitung der Skizze beenden* 26. Blenden Sie die Skizzenobjekte *Sketch* und *Sketch001* und die dazugehörige Bezugsebene mit dem Namen *Datum Plane* in der Baumansicht aus, indem Sie diese markieren und die Leertaste auf der Tastatur drücken oder mit





der rechten Maustaste im Kontextmenü *Ein/Ausblenden* wählen. So sind die einzelnen Objekte nicht durch die anderen Ebenen hindurch sichtbar, was sehr stören würde. Im Folgenden werden auf die weiteren fünf Bezugsebenen die Querschnitte des Zirkelschenkels skizziert.

### Querschnitte zeichnen

Zeichnen Sie nun das erste kleinere Dreieck hinter der Zirkelspitze. Da es sich nicht um ein gleichseitiges Dreieck handelt, nimmt man dazu nicht das Dreieckswerkzeug, sondern konstruiert die Form von Hand. Wechseln Sie

dazu in die zweite Skizze von unten, indem Sie auf *Sketch002* doppelklicken. Zeichnen Sie eine kurze Linie (27) vom Mittelpunkt der Hilfslinien aus nach oben. Falls nicht schon automatisch geschehen, schränken Sie sie vertikal ein (28). Fügen Sie noch eine Einschränkung mit dem Button *Den vertikalen Abstand zwischen zwei Punkten oder Streckenenden festlegen* (29) mit 3mm ein.

Zeichnen Sie zwei horizontale Linien auf die horizontale Hilfslinie (30). Schränken Sie die beiden Endpunkte zum Mittelpunkt hin mit dem Button *Den horizontalen Abstand zwischen zwei Punkten oder Streckenenden festlegen* (31) jeweils mit 2,5mm ein. Zeichnen Sie nun vom linken und rechten Endpunkt nach oben jeweils eine Linie, sodass ein Dreieck entsteht.

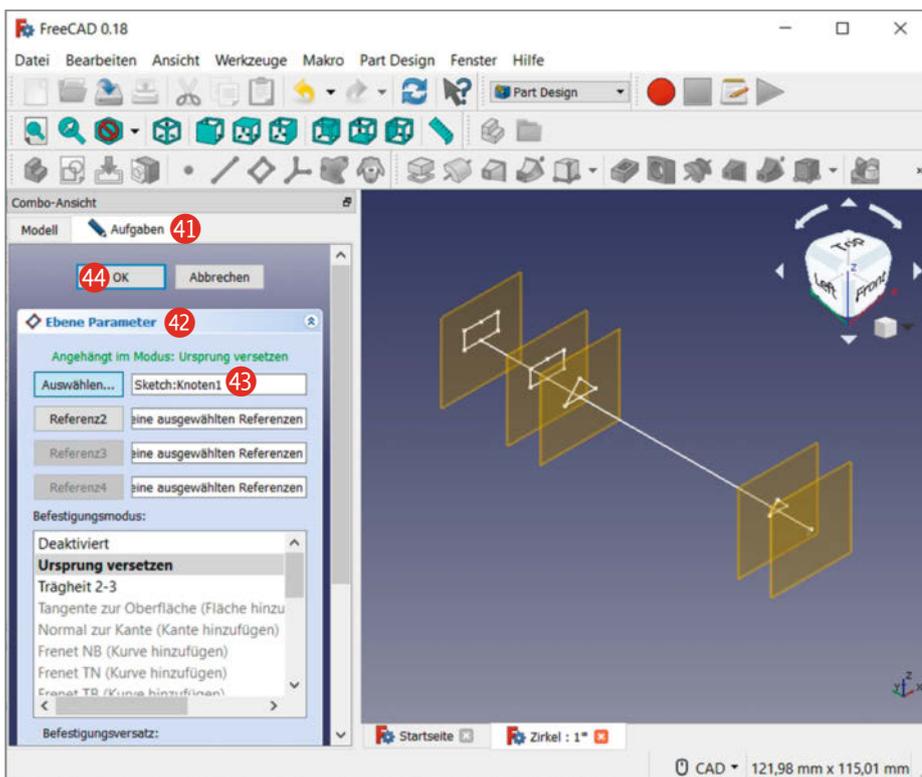
Achten Sie bei dieser und den folgenden Skizzen unbedingt darauf, dass die Endpunkte der Linien aufeinander liegen und die Formen dadurch geschlossen sind. Dabei hilft ein (zugegeben wenig auffälliges) Punktsymbol, das beim Linienzeichnen neben dem Mauszeiger erscheint, wenn zwei Endpunkte aufeinanderliegen.

Markieren Sie nun die vertikale Hilfslinie und entfernen Sie sie mit der Taste *Entf*. Die vertikale Einschränkung von 3mm bleibt dabei bestehen. Verlassen Sie die Skizze wieder mit dem Button *Bearbeitung der Skizze beenden* (32) und blenden Sie auch dieses Skizzenobjekt und die dazugehörige Bezugsebene aus.

Nun erstellen Sie auf gleiche Weise das größere Dreieck auf der nächsten Bezugsebene. Wechseln Sie dazu in die dritte Skizze von unten mit dem Namen *Sketch003*. Zeichnen Sie wieder eine kurze Linie vom Mittelpunkt der Hilfslinien aus nach oben und schränken Sie sie vertikal ein, falls dies nicht schon automatisch geschehen ist. Legen Sie wieder per Einschränkung den vertikalen Abstand fest, diesmal auf 5mm (33).

Zeichnen Sie wieder zwei horizontale Linien auf die Hilfslinie und schränken Sie jeweils den Abstand der beiden Endpunkte zum Mittelpunkt hin auf jeweils mit 4,5 mm ein (34). Schließen Sie das Dreieck nach oben wie zuvor auch und entfernen dann die vertikale Hilfslinie. Verlassen Sie auch diese Skizze wieder mit dem Button *Bearbeitung der Skizze beenden* und blenden Sie auch dieses Skizzenobjekt mit der dazugehörigen Bezugsebene aus.

Erstellen Sie nun als Nächstes ein Viereck in der Skizze *Sketch004*. Zeichnen Sie eine kurze Linie vom Mittelpunkt der Hilfslinien aus nach oben (35) und schränken Sie diese gegebenenfalls vertikal ein. Fügen Sie noch eine Einschränkung mit dem Button *Den vertikalen Abstand zwischen zwei Punkten oder Streckenenden festlegen* mit 5mm ein (36). Zeichnen Sie wie bei den Dreiecken zuvor je eine Linie auf die horizontale Hilfslinie vom Mittelpunkt aus nach links und nach rechts. Zeichnen Sie am



oberen Ende der vertikalen Linie ebenfalls weitere Linien nach links und nach rechts 37. Schließen Sie die Enden mit je einer vertikalen Linie, sodass ein Viereck entsteht 38. Achten Sie darauf, dass alle Punkte verbunden sind und die Linien geschlossen sind, und schränken Sie sie gegebenenfalls noch vertikal ein.

Schränken Sie den Abstand der oberen Linie nach links und rechts mit jeweils 5mm ein 39. Markieren Sie nun die zuerst gezeichnete vertikale Hilfslinie und löschen Sie diese mit der Taste *Entf*. Verlassen Sie die Skizze wieder mit dem Button *Bearbeitung der Skizze beenden* und blenden Sie auch dieses Skizzenobjekt mit der dazugehörigen Bezugsebene aus. Wiederholen Sie den vorherigen Schritt, nur dass Sie das Viereck mit identischen Maßen diesmal auf der Skizze *Sketch005* zeichnen 40.

### Zwischendrin aufräumen

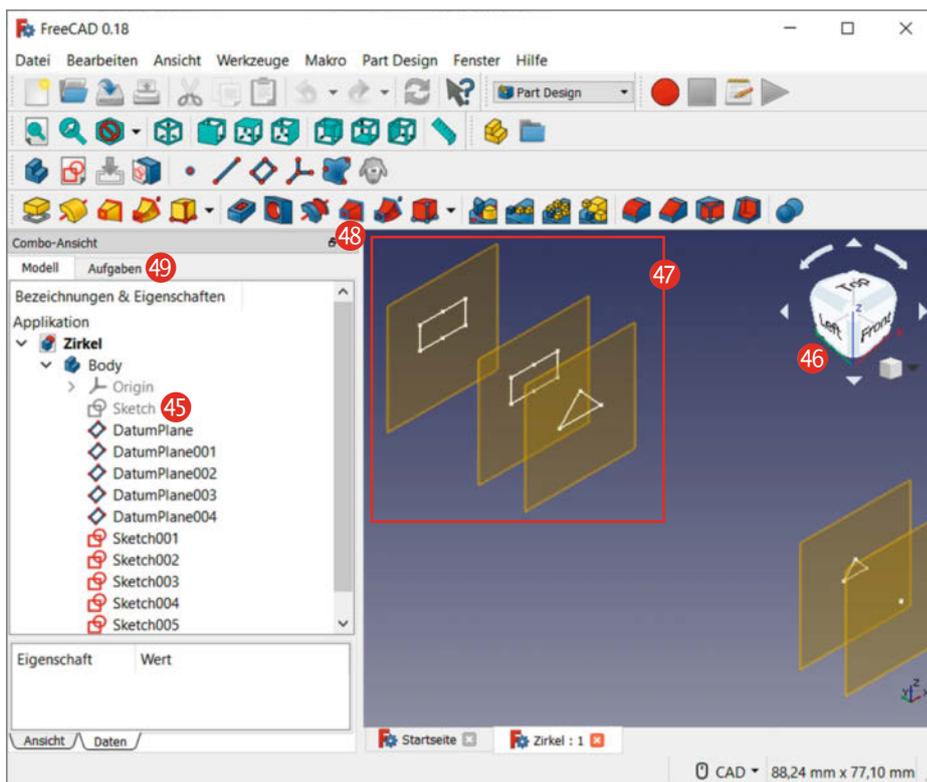
Nachdem Sie die letzte Skizze verlassen haben und sich wieder im Arbeitsbereich *Part Design* befinden, blenden Sie anschließend alle Skizzenobjekte und Bezugsebenen in der Baumansicht links wieder ein – entweder über das Kontextmenü oder durch Drücken der Leertaste. Die zuallererst gezeichneten Hilfslinien zur Positionierung der Bezugsebenen benötigen wir für den nächsten Schritt nicht mehr. Damit diese bei der 3D-Objekterstellung nicht stören, löschen wir sie wie folgt:

Entfernen Sie zuerst die Objektabhängigkeiten der Bezugsebenen zu den Hilfslinien. Doppelklicken Sie dazu in die erste Bezugsebene mit dem Namen *Datum Plane* und löschen Sie in den *Aufgaben* 41 unter *Ebene Parameter* 42 den ersten Punkt neben dem Button *Auswählen* mit der Bezeichnung *Sketch:Knoten1* 43 heraus, indem Sie ihn markieren und die Taste *Entf* drücken. Bestätigen Sie dann die Änderung mit *OK* 44. Löschen Sie auf diese Weise die Referenzen aus allen Bezugsebenen heraus.

Wechseln Sie anschließend in die erste Skizze mit dem schlichten Namen *Sketch* ohne folgende Nummer 45, markieren dort alle Linien und löschen Sie diese. Kippen Sie mithilfe des Navigation Cube 46 die Ansicht auf die linke untere Ecke der Fläche *Top*. Jetzt dürften nur noch die einzelnen Bezugsebenen mit den Skizzenzeichnungen sichtbar sein 47. Klicken Sie dann in den *Aufgaben* auf *Schließen*, um wieder in den Arbeitsbereich *Part Design* zu gelangen.

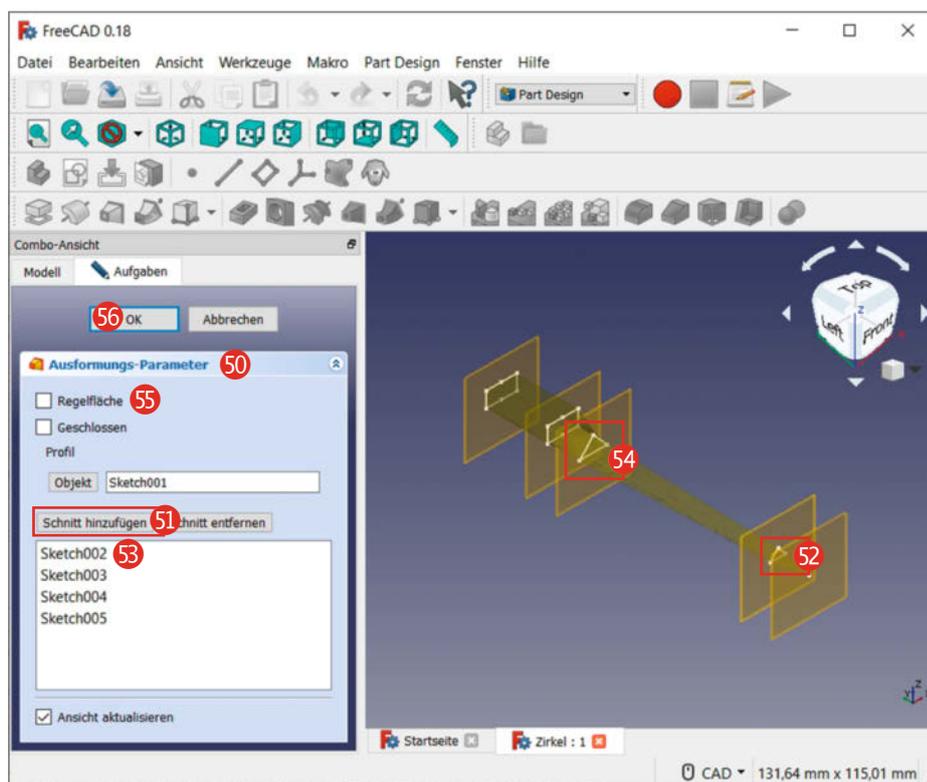
### 3D erzeugen

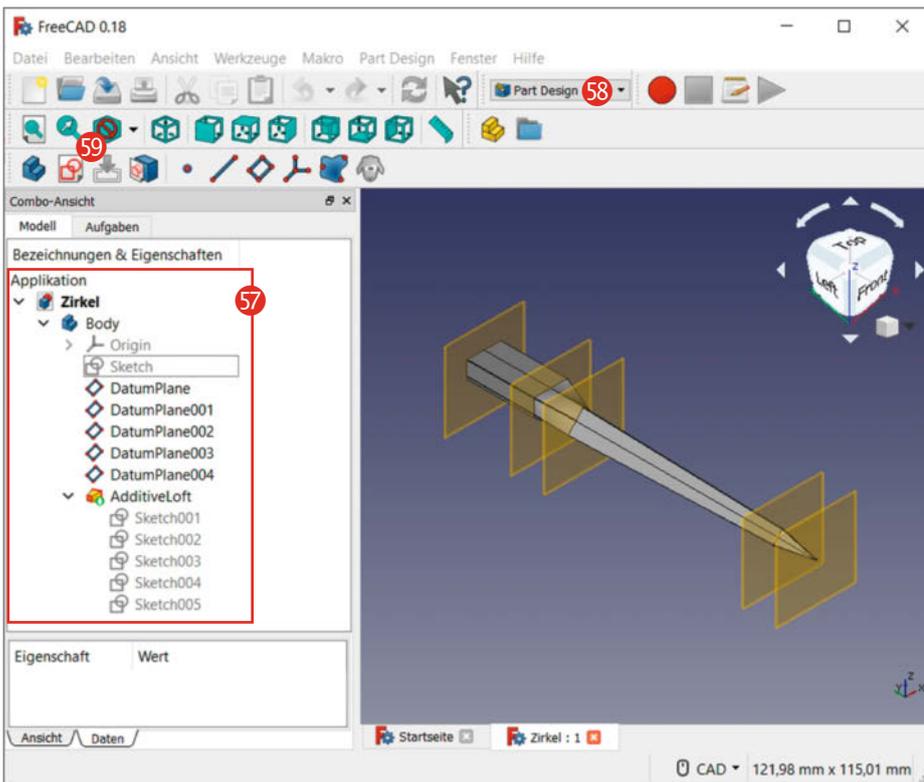
Aus den Querschnitten lässt sich jetzt ein 3D-Objekt erzeugen. Dazu drücken Sie in der Symbolleiste den Button *Loft eines ausgewählten Profils durch andere Profilabschnitte* 48. Markieren Sie in den *Aufgaben* 49 das Element *Sketch001* (*Gültig*) und bestätigen Sie mit



OK. In den jetzt erschienenen Ausformungs-Parametern 50 drücken Sie den Button *Schnitt hinzufügen* 51 und markieren in der zweiten Skizze eine Linie des Dreiecks 52 (Achtung: Nicht stattdessen die *DatumPlane* anklicken!). FreeCAD erstellt zwischen den beiden Zeichenebenen ein 3D-Objekt der Zirkelspitze.

Außerdem ist unterhalb des Buttons das Objekt *Sketch002* 53 hinzugekommen. Falls kein 3D-Objekt erscheint, ist das ein Hinweis darauf, dass die Form auf der zuletzt gewählten Skizze nicht geschlossen ist. Brechen Sie in diesem Fall den Vorgang ab und nehmen Sie nochmal die betreffende Skizze unter die Lupe, ob die





Bestätigen Sie nun die Fertigstellung mit OK **56**. Blenden Sie noch alle Zeichenebenen-Objekte in der Baumsicht **57** mit der Leertaste oder der rechten Maustaste unter *Ein/Ausblenden* aus, die mit dem Namen *DatumPlane* beginnen.

### Zirkelschlag

Zeichnen Sie nun die Zirkelbögen, mit denen sich der Zirkel mit einer Hand auseinanderdrücken lässt. Wechseln Sie dazu im Navigation Cube in die Ansicht von rechts, indem Sie auf dem Würfel auf *Right* klicken.

Der Arbeitsbereich *Part Design* **58** lässt pro Körper nur ein 3D-Objekt zu. Mit dem Befehl *Erzeugen und Aktivieren eines neuen Körpers* **59** erstellen Sie einen neuen, der in der Baumsicht mit der Bezeichnung *Body001* erscheint **60**. Erstellen Sie darin eine neue Skizze und wählen Sie die Ebene *YZ\_Plane (Basis-Ebene)*. Zur Vereinfachung orientieren Sie sich an der rechten Seite des Zirkelkörpers **61** und zoomen Sie dort etwas hinein.

Zeichnen Sie von der rechten unteren Ecke des Zirkelkörpers drei waagerechte Linien **62** aneinanderhängend nach rechts. Schränken Sie sie von links nach rechts mit 6mm **63**, 22mm **64** und 28mm **65** ein. Falls sich die Linien verschoben haben sollten, schieben Sie sie wieder an die rechte untere Ecke des Zirkels.

Zeichnen Sie nun eine Linie vom Startpunkt der ersten Linie, also der rechten unteren Zirkelcke, nach links, so dass diese in den Zirkel hineinragt. Schränken Sie die Linie mit 2mm ein **66**. Der Startpunkt der ersten Linie soll an der rechten unteren Zirkelcke bleiben, falls nicht, verschieben Sie sie wieder dorthin. Zeichnen Sie vom linken Punkt der 2mm-Linie aus eine weitere Linie senkrecht nach oben, in der Höhe des Schenkels **67**. Von dem oberen Punkt der Linie zeichnen Sie eine weitere Linie nach rechts von etwa 20mm Länge **68**.

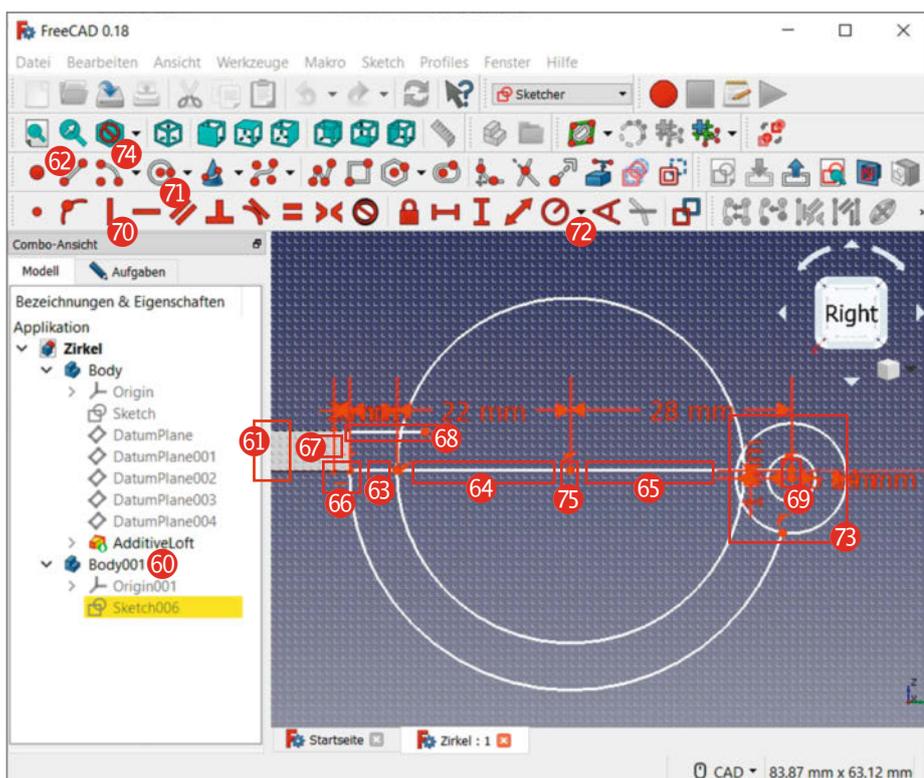
Damit der Zirkel sich am Ende schließen lässt, ist ein kleiner Abstand zum Gelenk hin zwischen den Schenkeln nötig, was man erreicht, indem man das Gelenk etwas außer der Mitte setzt. Dazu zeichnen Sie nun vom äußeren rechten Punkt der 28mm langen Linie eine kurze Linie nach unten **69**, schränken sie auf die Länge von 1mm ein und fügen noch eine vertikale Einschränkung hinzu **70**. Von diesem unteren Punkt zeichnen Sie einen kleineren Kreis **71** und anschließend einen größeren Kreis, wieder mit dem gleichen Mittelpunkt. Schränken Sie den kleinen Kreis mit einem Durchmesser von 6,1mm ein und den größeren mit 14mm – das Werkzeug zur Einschränkung des Durchmessers bringt ein Klick auf den kleinen Abwärtspfeil in der Werkzeugleiste zum Vorschein **72**. Beide Kreise zusammen bilden das Zirkelgelenk **73**.

Um das Gelenk mit den Schenkeln zu verbinden, zeichnen Sie einen 3-Punkt-Bogen **74**.

Endpunkte der einzelnen Linien wirklich alle aufeinander liegen.

Erscheint hingegen wie erwartet die 3D-Form, drücken Sie wieder den Button *Schnitt hinzufügen* und markieren eine Linie des Dreiecks der dritten Zeichenebene **54**. Dadurch verlängert sich das vorhandene

3D-Objekt in Richtung der Zeichenebene und das Objekt *Sketch003* erscheint im Aufgabenbereich. Fahren Sie so weiter fort, bis das 3D-Objekt durch alle Zeichenebenen verläuft. Das 3D-Objekt erscheint sehr verdreht, mit Anhängen des Kästchens *Regelfläche* **55** in den Aufgaben nimmt es die richtige Form an.



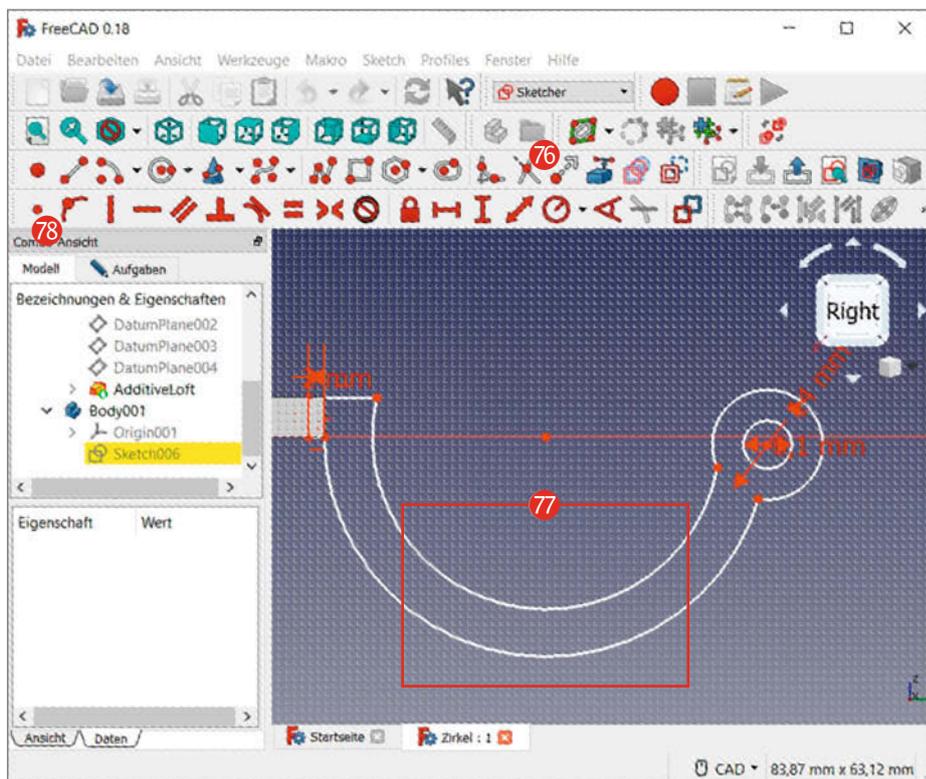
Für den Bogen verwenden Sie als Mittelpunkt den Punkt zwischen der 22mm- und der 28mm-Linie **75**. Als Startpunkt nutzen Sie den Punkt zwischen der 2mm- und der 6mm-Linie an der Schenkelecke **66**. Als Endpunkt nehmen Sie den größeren Kreis auf der rechten Seite **73**, wobei sich dieser bei Annäherung mit der Maus gelb verfärbt. Zeichnen Sie nun um demselben Mittelpunkt einen Kreis, den Sie am Verbindungspunkt zwischen der 6mm- und der 22mm-Linie schließen.

Die Konstruktion enthält jetzt noch zu viele Linienstücke, die später bei der Aufdickung stören würden. Beschneiden Sie deshalb nun die überstehenden Linien mit dem Werkzeug *Trimmen einer Kante bezüglich deren ausgewählten Position* **76**. Dabei ist die Reihenfolge wichtig: Trimmen Sie zuerst den Überstand des zuletzt gezeichneten großen Kreises innerhalb des Umkreises des Gelenks auf der rechten Seite **73**. Daraufhin den kompletten oberen Teil des zuletzt gezeichneten Kreises. Dann den Überhang der oberen Linie auf der linken Seite **68** und anschließend den Abschnitt des Umkreises des Gelenks auf der rechten Seite, der zwischen den beiden Bögen liegt. Löschen Sie die Hilfslinien aus den Schritten **63** bis **65** sowie **69** komplett.

Der Zirkelbogen **77** sollte nun aus einem Teil bestehen und keine Überstände mehr haben. Durch Ziehen einer Linie und anschließendes Zurücknehmen dieses Schrittes (STRG+Z) können Sie feststellen, ob alle Linien für die 3D-Objekterstellung geschlossen sind. Falls nicht, finden Sie die Endpunkte der jeweiligen Abschnitte und verbinden Sie diese mit dem Werkzeug *Eine Koinzidenzbeschränkung für das gewählte Element setzen* **78**.

### Aufpolstern

Schließen Sie die Skizze und erstellen Sie einen weiteren Körper **79** im Arbeitsbereich *Part Design*. Kopieren Sie die Skizze mit dem Namen *Sketch006* mithilfe der Menüleiste



unter *Bearbeiten/Ausgewähltes Objekt duplizieren*. Um die Abhängigkeiten nicht mit zu kopieren, quittieren Sie die angezeigte Warnmeldung mit *Nein*. Daraufhin entsteht ein Objekt mit der Bezeichnung *Sketch007* im *Body002*.

Wählen Sie mittels Klick der rechten Maustaste auf das Objekt *Body001* im Kontextmenü den Punkt *Aktiven Körper umschalten*. Markieren Sie *Sketch006* und erstellen Sie mit dem Button *Selektierte Skizze aufpolstern* **80** ein 3D-Objekt. Ändern Sie beim Aufpolstern im Aufgabenfenster unter *Parameter der Aufpolsterung* die Länge in 2,4mm und bestätigen Sie mit OK. Wiederholen Sie die Erstellung eines 3D-Objektes im Körper *Body002* mit dem *Sketch007*.

Beide Zirkelbögen sitzen noch auf der gleichen Position. Um zwischen diesen für den nötigen Abstand zu sorgen, klicken Sie zur besseren Übersicht im Navigation Cube auf die obere rechte Ecke der Seite *Right*, sodass die Ansicht auf die Seite kippt **81**. Für die Positionierung des ersten Zirkelbogens markieren Sie in der Baumansicht bei *Body001* unter *Pad* den *Sketch006* **82** und klicken in den Eigenschaften in *Attachment*. Drücken Sie dort den Button mit den drei Punkten und geben Sie im Aufgabenbereich unter *Verschiebung* bei *Z* den Wert 2,6mm an. Damit verschiebt sich ein Zirkelbogen an den vorderen Rand des Schenkels und der zweite Zirkelbogen wird sichtbar. Damit zwischen

# Datenschutzrecht leicht gemacht!

**c't DSGVO 2021 – Neuauflage!**

Auch als einzelnes Heft erhältlich.

[shop.heise.de/ct-dsgvo21](http://shop.heise.de/ct-dsgvo21)

Heft + PDF für nur **29,90 €**

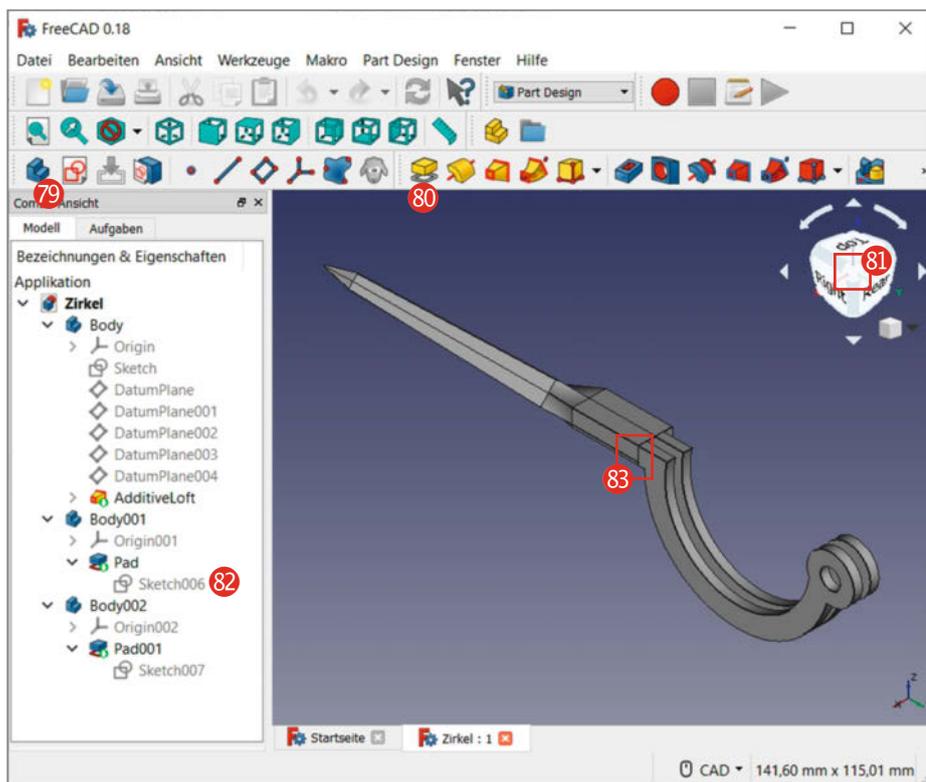


Heft + PDF mit über **20% Rabatt**

**NEUE**  
3. Auflage

 **heise shop**

[shop.heise.de/ct-dsgvo21](http://shop.heise.de/ct-dsgvo21)



dem Zirkelbogen und dem Schenkel ein wenig Spiel ist, verschieben Sie den Zirkelbogen unter X um 2mm aus dem Schenkel heraus **83**.

Positionieren Sie nun den zweiten Zirkelbogen mit dem Namen *Sketch007*, indem Sie

unter *Verschiebung* bei Z den Wert von -2,4mm und bei X wieder 2mm angeben.

Vereinigen Sie schließlich alle Körper zu einem Objekt, indem Sie zuerst das Schenkelobjekt mit dem Namen *Body* **84** mit der rechten Maustaste zum aktiven Körper machen

und dann mit gedrückter Steuerungstaste *Body001* **85** und *Body002* **86** markieren, bis alle Objekte im Zeichenbereich grün sind. Betätigen Sie den Button *Boolesche Operationen mit zwei oder mehr Körpern* **87** und überprüfen Sie, ob im Aufgabenfenster **88** im unteren Drop-Down-Fenster *Vereinigung* steht. Wenn die Objekte *Body001* und *Body002* im Auswahlfenster verfügbar sind, drücken Sie OK.

Falls sich die Objekte nicht vereinigen lassen, besteht eventuell eine Lücke zwischen den Bögen und dem Schenkel. Positionieren Sie in diesem Fall die Bögen wieder ein Stück weit näher in Richtung des Schenkels.

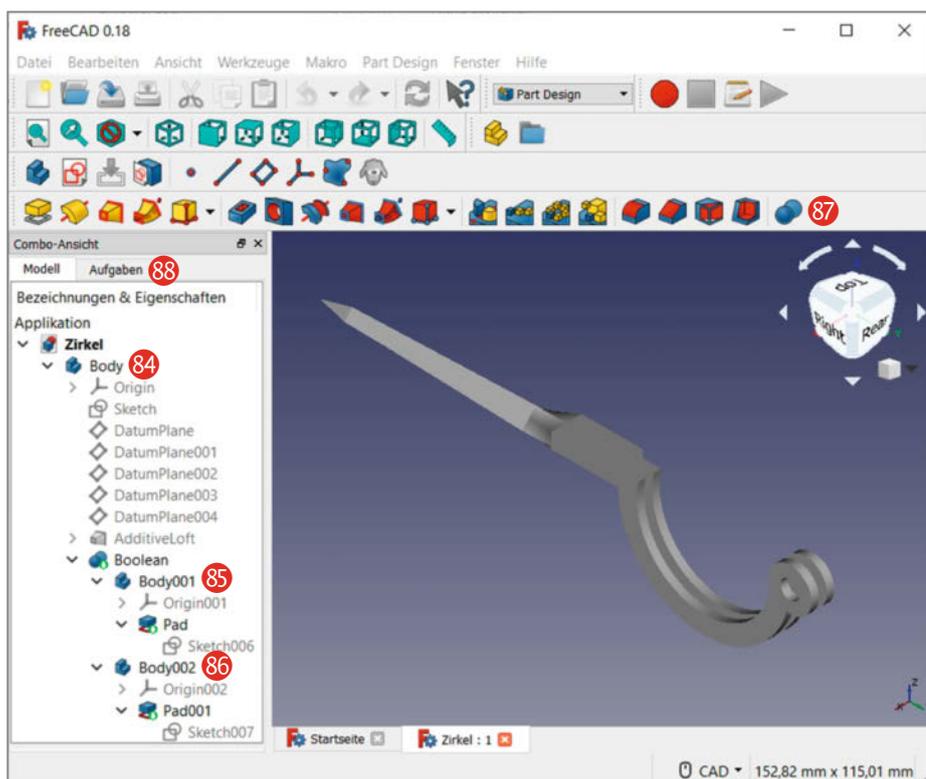
Passt alles, exportieren Sie den Zirkel jetzt, indem Sie in der Baumsicht den vereinigten Körper *Body* mit gedrückter Steuerungstaste markieren und in der Menüleiste auf *Datei/Export* klicken. Verwenden Sie das Exportformat *STL-Mesh (\*stl,\*ast)* und geben Sie als Dateinamen „Zirkel“ an.

### Drucken oder drucken lassen

Das Schöne an unserer Konstruktion ist, dass der Zirkel aus zwei exakt gleichen Teilen besteht, die nur einmal modelliert werden müssen. Wir haben den Zirkelbogen als ersten Prototypen aus PLA auf einem Ultimaker 2+ auf der Seite liegend ausgedruckt. Dabei fügt die Drucksoftware Stützstrukturen zwischen den Zirkelbögen und teilweise unterhalb des Schenkels ein, die später mit entsprechendem Aufwand entzückt werden müssen. Daraufhin haben wir den Zirkel zudem bei *fabberhouse.de* aus ABS mit herauswaschbaren Stützen drucken lassen. Eine Bilderstrecke mit den verschiedenen Druckversionen finden Sie online über den Link in der Kurzinfor.

Als erste Anschauungsobjekte taugen die Kunststoffdrucke zwar, aber sie sind bei den gewählten schlanken Maßen für die Zirkelschenkel jedoch nicht formstabil genug, um präzise damit arbeiten zu können. Deshalb haben wir es uns zudem gegönnt, den Zirkel bei *i.materialize* aus Stahl zu bestellen, aus Metallpulver lasergesintert (zu sehen auf dem Aufmacherfoto zu diesem Artikel). Die Kosten kamen auf 28,21 Euro pro Zirkelseite und beim online beschriebenen Bleistifthalter auf 38,37 Euro plus Umsatzsteuer und Versandkosten. Der Ausdruck aus Metall ist wesentlich fester als die Kunststoffversionen, jedoch sind die Zirkelspitzen erst einmal nicht spitz genug. Diese muss man entsprechend nachschleifen.

Als Gelenk haben wir eine Rändelschraube *hohe Form* DIN 464 M3 x 6mm verwendet sowie eine Rändelmutter *hohe Form* DIN 466 M3 (Bezugsquellen siehe Link in der Kurzinfor). Falls die Schraube im Gelenk nicht genügend Spiel hat, hilft ein Stück Schmirgelpapier – in der Konstruktion ist ein zehntel Millimeter Luft vorgesehen. —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 € >



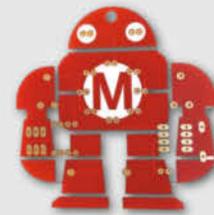
NEUER PREIS!

### 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

~~69,90 €~~  
36,90 €



### Makey Lötbausatz

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

Jetzt neu mit Schalter!

shop.heise.de/makey-bausatz

ab 4,90 € >



NEUER PREIS!

### Komplettset Argon ONE Case mit Raspberry Pi 4

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

shop.heise.de/argon-set

~~117,60 €~~

99,90 € >



### Stockschirm protec'ted

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

shop.heise.de/ct-schirm

22,90 € >



### c't Tassen

c't-Leser und -Fans trinken nicht einfach nur Kaffee, sie setzen Statements. Und zwar mit drei hochwertigen Blickfängern, individuell 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 >



# CAD-Designtipps für den 3D-Druck

von Billie Ruben

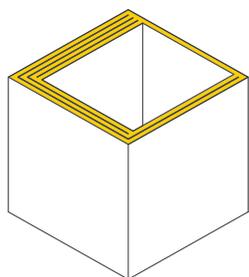
Wenn man hochwertige, funktionale und schöne Objekte für den 3D-Druck mit CAD-Software entwickeln möchte, gibt es einige Einschränkungen, die man beachten muss – wie bei eigentlich allen Produktionsprozessen. Um anderen den Einstieg in das Thema zu erleichtern, habe ich meine Erfahrungen in dieser Infografik gebündelt. Ich liebe es, zu sehen, wie Menschen ihre eigenen De-

signs entwerfen und mit ihnen ihre ganz individuellen Bedürfnisse und Wünsche erfüllen können.

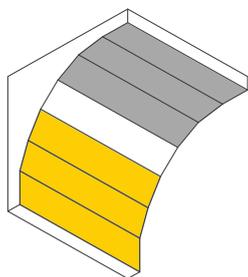
Diese Infografik ist Teil einer Reihe von Postern von mir zum Thema 3D-Druck. Alle meine Poster stehen kostenlos zum Download zur Verfügung, ich verkaufe aber auch Drucke über die Webseite RedBubble (siehe Link).

Außerdem gibt es eine sehr hilfsbereite 3D-Druck-Community auf Discord (siehe Link), die bei Problemen gerne mit Ratschlägen und Hilfestellung dient. —rehu

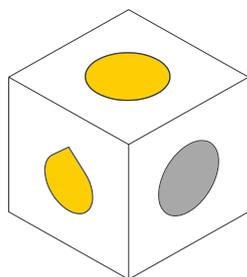
Alles zum Artikel im Web unter [make-magazin.de/x2qh](https://make-magazin.de/x2qh)



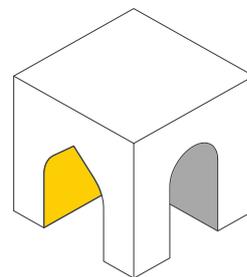
Für sauberes Slicing sollte die Wanddicke möglichst ein ganzzahliges Vielfaches der eingestellten Filamentbahnbreite (extrusion width) sein.



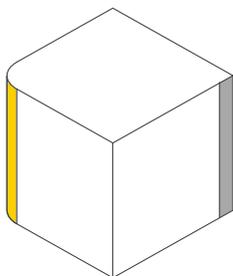
Meidet man steile Überhänge, spart man Stützmaterial (Support). Jede neue Schicht sollte beim Druck bequem auf der vorherigen liegen können.



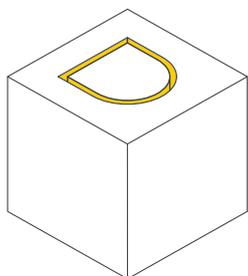
Löcher in horizontalen Flächen sind kein Problem, doch für Löcher in vertikalen Flächen ist eine Tropfenform besser – sie verringert die Überhänge.



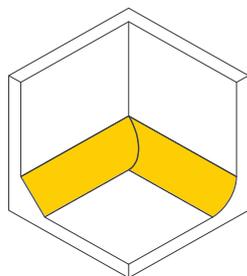
Analog zu den tropfenförmigen Löchern in horizontalen Flächen sind auch Spitzbögen besser zu drucken als Rundbögen.



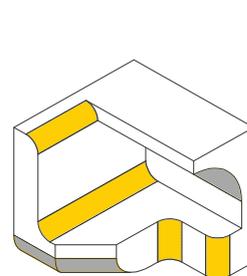
Wenn man vertikale Kanten leicht abrundet, erhöht sich die Druckqualität, da der Drucker nicht mit ruckartigen Richtungsänderungen zu kämpfen hat.



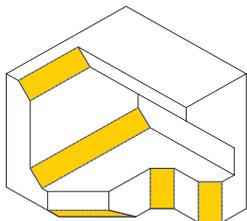
Damit man zwei Teile optimal ineinander stecken kann, sollte man am Ende des Modellierprozesses eine Lücke von 0,3mm zwischen ihnen hinzufügen.



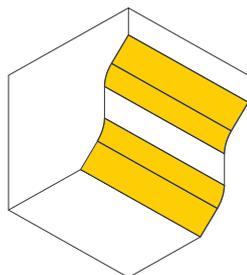
Wenn man eine innere Kante abrundet oder eine Fase hinzufügt, erhöht sich die Stabilität dieser Kante deutlich.



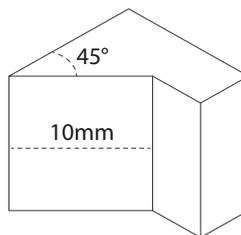
Abrundungen an der Unterseite von Flächen kommen meist nicht gut raus, da ihr Überhang zu steil ist. An allen anderen Stellen sehen sie schick aus.



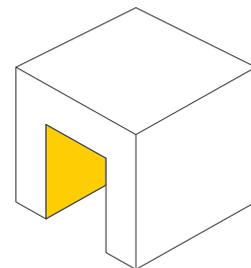
Gleichmäßige Fasen funktionieren dagegen immer (auch auf der Unterseite einer Fläche), da der Überhang konstant bei gut druckbaren 45° bleibt.



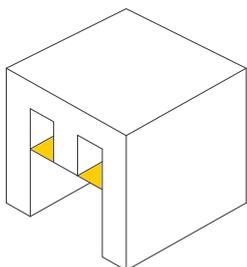
Die Mischung macht's: Die Überhänge von Rundungen sind mit einer benachbarten Fase geringer, die Fase selbst wirkt gleichzeitig sanfter.



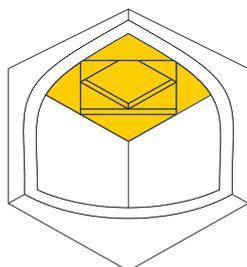
Nutzt man eine Software mit parametrischer Konstruktion und Einschränkungen wie FreeCAD (S. 128), kann man Entwürfe nachträglich gut modifizieren.



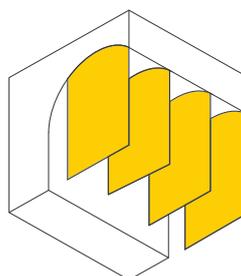
Die meisten 3D-Drucker können kleine Lücken im Modell problemlos überbrücken. Die meisten von ihnen bewältigen mindestens 2cm Distanz.



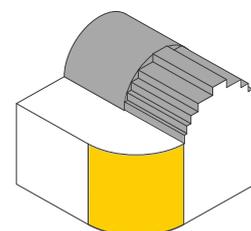
Eine schmale zusätzliche Brücke im Modell verringert den Stützmaterialbedarf – und kann nach dem Druck leicht entfernt werden.



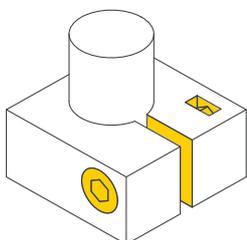
Unter einem größeren freitragenden Dach helfen Schichten gestaffelter Dreiecke, größere Distanzen zu überbrücken.



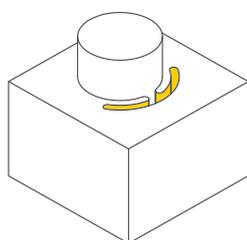
Mit provisorischen senkrechten Rippen kann man dem 3D-Drucker den Druck von großen Überhängen mit wenig Aufwand erleichtern.



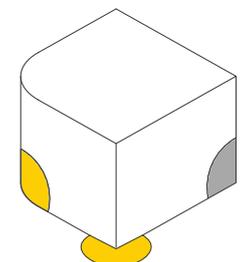
Kurven um eine senkrechte Achse sehen gut aus – durch die sichtbaren Druckschichten können sie in anderen Richtungen stufig wirken.



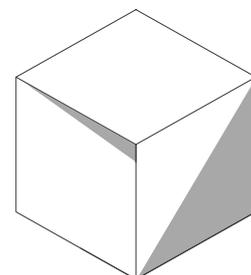
Ergänzt man ein Loch im Modell mit einer Aussparung, einer Schraube und einer versenkten Mutter, kann man Objekte in dieses Loch einspannen.



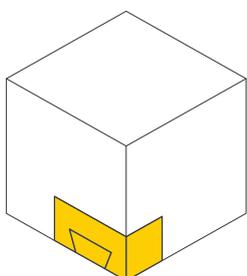
Kleine Aussparungen im Modell machen das Material an dieser Stelle flexibler – so greifen zum Beispiel Steckverbindungen besser.



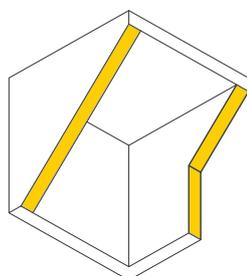
Drucke lösen sich seltener vom Druckbett, wenn man die Ecken abrundet oder kleine Mäusehörchen hinzufügt, die man später abschneidet.



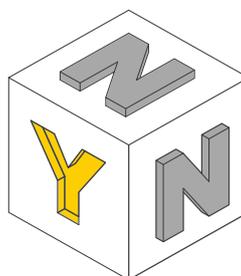
Software wie Tinkercad erzeugt zuverlässig Objekte, die „manifolds“ sind, was die meisten Slicer-Fehler vermeidet, etwa Lücken oder Dopplungen.



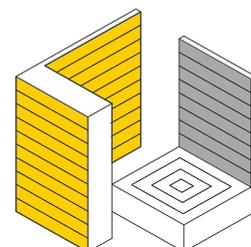
Für komplexe Elemente eines größeren Objekts oder solche, die genau zu anderen passen müssen, lohnt sich meist ein separater Probedruck.



Man kann diagonale Rippen hinzufügen, um ein Dach abzustützen. Das kann auch innerhalb von großen Modellen von Vorteil sein.



3D-gedruckter Text sieht am besten aus, wenn er in eine vertikale Fläche geprägt ist. Hier gibt es die wenigsten Überhänge und die beste Auflösung.

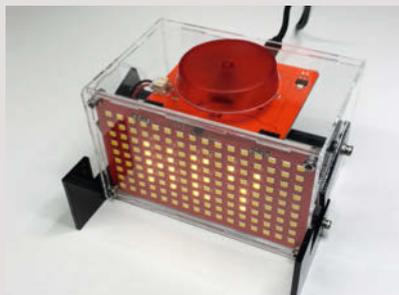


Da 3D-Drucker ihre Schichten in einer Ebene aufbringen, hat die Druckausrichtung einen erheblichen Einfluss auf die Stabilität.

**Ausprobiert**  
— von Make: —

## STEM Box

### Abo-Bastelbox für Kinder ab 11 Jahren



Die neue Abo-Box von *CircuitMess* liefert alle drei Monate neue DIY-Elektronikprojekte ins Haus. Der Zusammenbau soll Kindern ab elf Jahren mit etwas Hilfe möglich sein, richtet sich aber auch an Erwachsene. Später sollen die Projekte jeweils einzeln erhältlich sein. Über die Crowdfunding-Plattform *Indiegogo* können Einzel-Abos mit bis zu neun Boxen und Klassensätze bestellt werden. Wir haben die erste Box mit dem Sprachassistenten *Spencer* angeschaut, die mit einem praktischen Werkzeugset inklusive Lötkolben geliefert wird. Ähnlich wie Siri oder Alexa ermöglicht *Spencer* die sprachgesteuerte Abfrage von Informationen, momentan allerdings nur auf Englisch. Die englischsprachige Online-Bauanleitung ist gut gelungen, sodass der Bau auch Anfängern in der angegebenen Zeit möglich ist. Bei Problemen helfen die Online-Community und der Hersteller auf der Plattform *Discord* schnell weiter. Insgesamt ein hochwertiger, wenn auch nicht ganz preisgünstiger Bausatz. Das größte Manko: Um eigene Programme zu ermöglichen, sollte *Spencer* in der grafischen Programmierumgebung *Circuit-Blocks* der Firma unterstützt werden. Bis zum Redaktionsschluss war diese jedoch nicht freigeschaltet. Unser *Spencer* bespaßt trotzdem die ganze Familie.

—Jens Hackel/hch

Einen ausführlicheren Testbericht finden Sie online unter [make-magazin.de/xjnp](http://make-magazin.de/xjnp)

|            |  |
|------------|--|
| Hersteller | CircuitMess  |
| URL        | <a href="http://circuitmess.com">circuitmess.com</a> |
| Preis      | ab 116 US-\$/Box                                     |

## Piezobrush PZ3

### Plasma-Handgerät zur Oberflächenaktivierung

Gute Haftung ist wichtig beim Kleben, Lackieren und 3D-Druck – und die lässt sich in vielen Fällen verbessern, indem man die Oberflächen durch Kaltplasma (ionisiertes Gas) aktiviert, dass durch Hochspannung erzeugt wird.

Solch eine Behandlung ist mit dem Plasma-Handgerät PZ3 von Relyon Plasma überall möglich, denn das kleine Gerät arbeitet mit Luft als Arbeitsgas. Ein Piezo-Transformator erzeugt die erforderliche Hochspannung, mit deren Hilfe die Luft ionisiert und somit zum Plasma wird. Der PZ3 muss lediglich über die entsprechende Oberfläche geführt werden und aktiviert sie dabei. Die Leistung des Geräts lässt sich für empfindliche Oberflächen auf bis zu 40 Prozent reduzieren.

Dem Geräte-Set im stabilen Kunststoffkoffer zum Preis von 2400 Euro liegen außer einem Steckernetzteil noch zwei wechselbare Module bei, je eines für elektrisch leitende und nichtleitende Oberflächen. Auf Anfrage stellt der Hersteller auch Leihgeräte zur Verfügung.

Die Bedienung erfolgt über Drucktasten, eine Farbdisplay zeigt die Einstellungen an.



Bei Versuchen in der Redaktion konnte das Warming von 3-Drucken aus ABS deutlich verringert werden. Allerdings zeigte sich dabei auch, dass für die Arbeit mit dem PZ3 eine Abzugsvorrichtung dringend zu empfehlen ist, die das in großer Menge entstehende Ozon noch in Düsennähe des Geräts absaugt.

—hgb

Das Gerät wurde uns vom Hersteller für den Test zur Verfügung gestellt.

|            |  |
|------------|--|
| Hersteller | Relyon Plasma GmbH   |
| URL        | <a href="http://make-magazin.de/xjnp">make-magazin.de/xjnp</a> |
| Preis      | 2400 €   |

## OpenHAB 3.0

### Open-Source-Heimautomation

Mit Version 3 des Open-Source-Smart-Home-Systems lassen sich die Oberflächen für die Interaktion der Nutzer mit dem Smart Home individuell gestalten – je nach Zweck und Zielgruppe etwa als Landkarten, Grundrisse oder Diagramme. Wer gerne visuell programmiert, kann seine Skripte für die Hausautomatisierung jetzt auch mit *Blockly* bauen.

Eine sogenannte semantische Modellierung abstrahiert von den technischen Details der eingebundenen Geräte. Das ist Teil der Vereinheitlichung, die mit *openHAB 3* unter der Haube vollzogen wurde. So ist die in Version 2 noch vorhandene Kompatibilitätsschicht zu 1.x über Bord geworfen worden – zum Ausgleich wurden viele *Bindings* aus dieser Ära aber neu implementiert, weil die

alten nicht mehr funktionieren. Im Zuge der Angleichung existiert auch nur noch eine einheitliche *Rule Engine*, die die in der eigenen Domain Specific Language (DSL) von *openHAB* verfassten Regeln im User Interface integriert. Daneben lassen sich Regeln jetzt auch in JavaScript, Groovy und der ehemals als *JPython* bekannten Java-Implementierung von Python namens *Jython* erstellen.

Mehr zu *OpenHAB 3.0* lesen Sie unter [make-magazin.de/xjnp](http://make-magazin.de/xjnp)

|            |  |
|------------|--|
| Hersteller | OpenHAB  |
| URL        | <a href="http://openhbab.org">openhbab.org</a> |
| Preis      | kostenlos                                      |

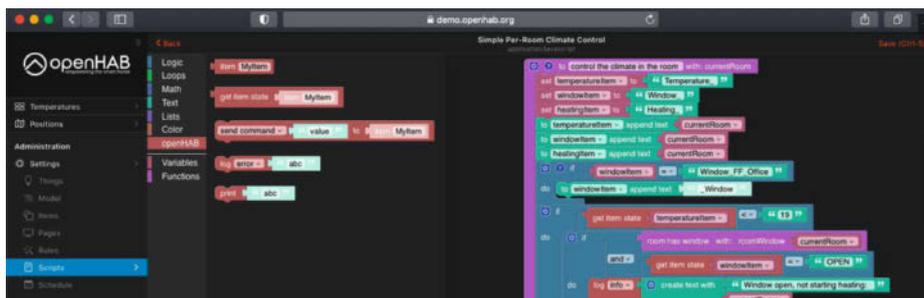


Bild: openhab.org

## Joy-Car

### Lernroboter auf Micro:Bit-Basis

Autonomes Fahren, Linien folgen, Hindernissen ausweichen, fernsteuern per Bluetooth - das verspricht das Joy-Car des Herstellers Joy-IT. Der zweirädrige Fahrroboter kommt als Bausatz und muss erstmal zusammengeschaubt und -gesteckt werden. Die bunte Anleitung dazu ist inhaltlich und grafisch hervorragend. Leider trübten den Aufbauspaß fehlende Teile, ein unpassiges Design und schief sitzende Platinen. Konkret fehlten eine lasergeschnittene Batteriehalterung aus Acryl und die Encoderscheiben für die Motoren schleiften an der Motorhalterung. Die Line-track- und Abstandssensoren waren in THT-Bestückung gefertigt, das heißt, auf einer Seite schauen die Beinchen der Bauelemente heraus. Das führte dazu, dass sich die Sensorplatinen nicht plan auf die Chassis-Platten montieren ließen, sondern halb schief halb schwebend verschraubt werden mussten. Das sieht nicht nur unschön aus, es zerkratzt auch das Acryl. Mit SMD-bestückten Platinen wäre das nicht passiert.

Die Verkabelung aller Sensoren und Motoren mit der Adapterplatine (Mainboard genannt) des Micro:Bit ist fummelig, war aber dank farbiger Kabel übersichtlich und schnell erledigt. Zum Betrieb reichen vier AA-Batterien. Die Funktion der verschiedenen Sensoren und wie man sie einstellt, wird auf der Homepage des Joy-Car erklärt. Zur Programmierung



**Ausprobiert**  
— von Make: —

des Micro:Bit bietet Joy-IT die grafische Oberfläche *Makecode* und *MicroPython* in Kombination mit dem Editor *Mu* an. Für beide Varianten gibt es Tutorials zum Steuern der Roboter-Funktionen, wobei die für *Makecode* detailliert für Einsteiger erklärt sind. Die Beispiele für *MicroPython* hingegen sind weder erklärt noch enthalten die Listings Kommentare und sollen damit laut Hersteller eher Experten ansprechen. Alle Beispiele stehen als fertige Programme zum Download bereit. Der

Bausatz kostet rund 55 Euro, der Micro:Bit ist aber nicht enthalten und muss für knapp 16 Euro separat gekauft werden. —dab

*Der Bausatz wurde uns vom Hersteller für den Test zur Verfügung gestellt.*

|            |                      |
|------------|----------------------|
| Hersteller | Joy-IT               |
| URL        | joycar.joy-it.net/de |
| Preis      | ab 55 €              |

## Seeeduno Xiao Expansion Board

### Experimentier-Board

Mit dem briefmarkenkleinen Xiao hat der Hersteller Seeed vor einiger Zeit einen günstigen, knapp 6 Euro billigen Mikrocontroller mit ARM-Cortex-M0+ auf den Markt gebracht. Passend dazu gibt es für den 14-poligen Winzling nun ein Expansion-Board für knapp 17 Euro mit Tasten, Summer, microSD-Kartensteckplatz, OLED-Anzeige und Echtzeituhr nebst Batteriehalter zu deren Versorgung mit Strom. Daneben lässt sich an das Board ein LiPo-Akku anschließen, sollte der Xiao nicht per USB mit Strom versorgt werden. Hängt das Board dennoch am USB-Port, wird der Akku über eine Ladeelektronik aufgeladen. Mit einem Schiebeschalter kann man das Board im Betrieb vom Akku trennen. Daneben bringt das Board vier der Seeed-typischen Groove-Steckplätze mit, an die man Sensoren

und Aktoren mit passenden Steckern anschließen kann.

Seeed hat das Board umfangreich dokumentiert und Tutorials zu allen Funktionen erstellt, die mit der Arduino-IDE funktionieren. So lässt sich mit der Nachinstallation einiger Bibliotheken schnell ein interaktives Projekt mit Grafik, Sound und Eingabe erstellen. Der Xiao unterstützt auch *MicroPython*; dafür zeigt die Dokumentation einige Beispiele zur Ansteuerung von LEDs, des Buzzers und wie man Daten von der SD-Karte liest. Leider fehlt ein Beispiel zur Ansteuerung des Displays. Da es sich um einen SSD1306-Typen mit I<sup>2</sup>C-Ansteuerung handelt, sind aber *MicroPython*-Module prinzipiell verfügbar und müssen nur leicht angepasst werden.

Mit dem Board spart man sich das Zusammenstecken verschiedener Komponenten



**Ausprobiert**  
— von Make: —

und Breakout-Boards und kann kompakte Prototypen bauen. Und für den Preis macht man nichts verkehrt. —dab

*Das Board wurde uns vom Hersteller für den Test zur Verfügung gestellt.*

|            |                      |
|------------|----------------------|
| Hersteller | Seeed                |
| URL        | make-magazin.de/xjnp |
| Preis      | 17 €                 |

## Prusament PVB

### Mit Isopropyl polierbares Filament für 3D-Druck



Die Firma *Prusa Research*, Hersteller von 3D-Druckern mit eigener Filamentproduktion, bringt mit seinem *Prusament PVB* (für Polyvinylbutyral) ein Material für den FDM-Druck auf den Markt, das sich mithilfe des relativ harmlosen Isopropylalkohols zu nahezu glasartigen Oberflächen glätten lässt. Mit Lösemitteln lassen sich zwar auch andere Materialien auf ähnliche Weise glätten, doch die dafür verwendeten Substanzen sind deutlich aggressiver oder hoch giftig.

PVB soll mit einer Düsen- und Druckbetttemperatur von 215°C und 75°C verarbeitet werden und ähnliche mechanische Eigenschaften wie PETG besitzen. Größter Nachteil laut Hersteller ist die geringe Wärmebeständigkeit der fertigen Werkstücke von 60°C, ferner eine etwas geringere Haftung zwischen den Schichten als bei PLA. Prusa empfiehlt speziell beim Druck im sogenannten Vasenmodus, bei dem Formen ausschließlich als eine Materialbahn breite Hülle gedruckt werden, den Einsatz von Druckdüsen mit einer unüblich großen Bohrung von 0,8mm zur Verringerung der Druckzeit. Die dadurch entstehende rauere Oberfläche kann anschließend leicht geglättet werden. Durch die größere Schichtdicke soll sich sogar die Transparenz erhöhen.

Das Filament gibt es in sechs Farbtonen, die aber alle transparent sind, sodass geglättete Drucke wie aus gefärbtem Glas aussehen sollen. Neben den üblichen 500g-Spulen gibt es auch eine Musterbox mit je 25g Filament in allen sechs Farben, die mit ebenfalls 25 € im Vergleich aber überraschend teuer ist. —hgb/pek

|            |                  |
|------------|------------------|
| Hersteller | Prusa Research   |
| URL        | shop.prusa3d.com |
| Preis      | 25 € / 500g      |

## Anet A8 Plus

### Günstiger FDM-3D-Drucker

Hand aufs Herz: Blicken Sie bei den verschiedenen Drucker-Modellen aus China noch durch, die alle irgendwie *Anet A8* heißen? Auf den originalen *Anet A8* folgte eine überarbeitete Version unter identischem Namen, in Make 5/20 haben wir auf Seite 134 den *Anet A8E* kurz vorgestellt, jetzt haben die Kollegen von *Techstage* das Modell *Anet A8 Plus* einem eingehenden Test unterzogen – und dabei kräftig geflucht: Denn für (je nach Händler) weniger als 200 Euro bekommt man damit zwar einen 3D-Drucker mit satten 30cm × 30cm × 35cm Bauraum, Metallrahmen, Display, beheiztem Glasdruckbett und Direkt-Extruder (maximale Düsentemperatur: 250°C). Unter Umständen wird die Maschine aber nur als echter Bausatz geliefert, in einer Kleinteiligkeit, wie sie vor zehn Jahren mal Standard war, etwa beim allerersten *MakerBot* – noch dazu mit stark verbesserungsbedürftiger Bauanleitung.

Da der fertige Drucker durchaus gute Ergebnisse liefert und die Maschine mit ihrem erstaunlichen Preis/Bauraum-Verhältnis interessant ist, empfiehlt *Techstage* dringend, beim Kauf genau hinzuschauen: Denn den Drucker gibt es als vormontiertes *Kit*, aus dem sich die Maschine in zwei bis drei Stunden aufbauen lassen sollte, und eben als *Bausatz*, mit

**Ausprobiert**  
– von *techstage*



dem die Kollegen trotz einschlägiger Erfahrung anderthalb Tage zu kämpfen hatten. Der ist höchstens einschlägig erfahrenen und geduldigen Bastlern zu empfehlen. —pek

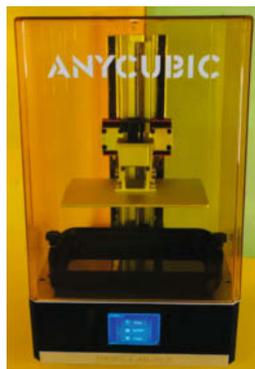
Einen ausführlicheren Testbericht mit vielen Bildern und Preisvergleich finden Sie online unter [make-magazin.de/xjnp](http://make-magazin.de/xjnp)

|            |                          |
|------------|--------------------------|
| Hersteller | Anet                     |
| URL        | anet3d.com/pages/a8-plus |
| Preis      | ab etwa 180 €            |

## Photon Mono X

### Resin-3D-Drucker mit großem Bauraum

Viele sehr günstige Resin-3D-Drucker, die lichtempfindliches Flüssigharz zu feinsten Werkstücken aushärten, arbeiten mit handelsüblichen Smartphone-Displays und bringen daher nur recht kleine Objekte zustande. Für die Belichtung werden diese Displays aus der Telefonproduktion zudem auf UV-Backlight umgerüstet. Der Bauraum des *Anycubic Photon Mono X* hingegen misst satte 19,2cm × 12cm × 24,5cm – das gibt es selten bei einem Resindrucker, noch seltener bei einem unter 1000 Euro. Im Drucker steckt ein monochromes 4K-Display, was nicht nur eine deutlich höhere Lebensdauer verspricht (bei solchen Druckern sind die Displays tatsächlich Verschleißteile!), sondern auch mehr UV-Licht durchlässt, sodass im Test des *Mono X* durch *Techstage* mit Belichtungszeiten von gerade mal anderthalb Sekunden pro



Schicht gute Ergebnisse zu erzielen waren.

Auch sonst waren die Kollegen von diesem Gerät angetan: Die Bedienung ist einfach, das Gerät solide gebaut und verarbeitet und die Qualität der Drucke ließ nichts zu wünschen übrig. Ob man so große Resin-3D-Drucke tatsächlich braucht, muss man sich allerdings gut überlegen – da das Material an sich recht teuer ist, geht auch jeder einzelne große Druck natürlich ins Geld. —pek

Einen ausführlicheren Testbericht mit vielen Bildern und Preisvergleich finden Sie online unter [make-magazin.de/xjnp](http://make-magazin.de/xjnp)

|            |               |
|------------|---------------|
| Hersteller | Anycubic      |
| URL        | anycubic.com  |
| Preis      | ab etwa 800 € |

**Ausprobiert**  
– von *techstage*

# 32-Bit Mainboard und Direktantrieb

## Upgrades für den 3D-Drucker Creality Ender 3

Der *Creality Ender 3* gehört in seinen verschiedenen Varianten zu den beliebtesten günstigen FDM-3D-Druckern aus dem Bausatz. Alle haben von Haus aus einen Bowden-Antrieb für die Filamentzufuhr. Der funktioniert gut bei harten Filamenten, bei elastischen Materialien gibt es Probleme, da sich diese im langen Schlauch zwischen Vorschubmotor am Rahmen und beweglichen Druckkopf leicht verkleben können. Als Abhilfe bietet Creality jetzt einen Umbausatz auf Direktantrieb an: Das Set besteht aus Druckkopf (inklusive Lüftern und 0,4mm-Düse), Filament-Antriebsmotor und der Trägerplatte, auf der beides zusammen übereinander im Abstand von wenigen Zentimetern montiert wird. Zur Montage muss lediglich der Druckkopfschlitten auf der x-Achse ausgetauscht und das Kabel vom alten auf den neuen

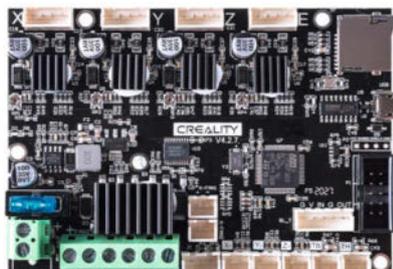


Bild: Creality

Schrittmotor umgesteckt werden. Eine Firmware-Änderung ist nicht notwendig.

Ältere Ender-3-Modelle mit dem Mainboard V1.1.4 drucken nicht gerade leise. Abhilfe schafft zwar das *Silent-Mainboard V1.1.5* (Make 6/19, S.125), zum ebenfalls wünschenswerten Anschluss eines genaueren Sensors fürs Bed-Leveln braucht man damit aber noch einen zusätzlichen Adapter und muss den Beeper des Druckers lahmlegen. Nimmt man stattdessen

das ursprünglich für den *Ender 5* gedachte Board V4.2.7, geht beides: Neben den leisen TMC2208-Treiber-ICs besitzt es einen eigenen Anschluss für den Touchsensor. Geliefert wird das Board mit installierter Marlin-2-Firmware, allerdings in einer Version für den Ender 5. Tipps zum Flashen mit der richtigen Firmware lesen Sie online unter [make-magazin.de/xjnp](http://make-magazin.de/xjnp) —hgb



Bild: Creality

|            |  |
|------------|--|
| Hersteller | Creality   |
| URL        | <a href="http://creality3dshop.eu">creality3dshop.eu</a> |
| Preis      | Mainboard: 34 €<br>Direktantrieb: 46 €                   |

# CNC3-3018 Pro

## CNC-Gravierer aus dem Bausatz

Beim 3018 Pro handelt es sich um einen von verschiedenen Herstellern aus China angebotenen und sehr günstigen Bausatz, mit dem man eine handliche CNC-gesteuerte Gravurmaschine aufbauen kann. Der Name verrät schon die wichtigsten Informationen: Der befahrbare Raum beläuft sich auf 30cm in der Breite und 18cm in der Tiefe. Die Spindel kann von der Maschine um 8cm in der Höhe bewegt werden. Zur Feinjustierung kann man den Motor um etwa 1cm von Hand in der Klemmfassung verschieben.

Im Bausatz sind alle Teile enthalten, um nach dem Aufbau sofort losfräsen zu können: Rahmenteile wie Aluminiumprofile, Bakelit-Formstücke, Gewindestangen und Führungsrohre, Schrittmotoren und GRBL-kompatibles Controlboard, sowie eine vorgefertigte Spindelaufnahme inklusive Bürstenmotor. Dazu gibt es die üblichen Montageteile wie Schrauben, Kugellager,

3D-gedruckte Führungen und einen Blumenstrauß Kabelbinder. Schlussendlich wird das Paket mittels Netzteil, einem Satz Gravurstichel und vier Klemmvorrichtungen zum Fixieren des Arbeitsmaterials abgerundet. Der Aufbau stellte sich im Test als etwas fummelig, aber innerhalb von zwei Stunden problemlos machbar heraus.

Fehlt nur noch, die mitgelieferte – oder wahlweise auch eigene – Software zu installieren und schon kann es losgehen. Ein Opferbrett liegt bei und kann mithilfe der mitgelieferten Fräsdateien und einer kabelgebundenen Fernsteuerung schnell befräst werden. Eigene Vorlagen lassen sich relativ simpel etwa mit *Inkscape* herstellen.

Die 3018 Pro stellt ein schönes und überschaubares Projekt dar, um sich im Bereich CNC und Holzbearbeitung die Hände schmutzig zu machen und Lust auf mehr zu kriegen. —Joshua von Gizycki/rehu



|            |  |
|------------|--|
| Hersteller | diverse, etwa SainSmart  |
| URL        | <a href="http://make-magazin.de/xjnp">make-magazin.de/xjnp</a> |
| Preis      | 150 bis 200 €  |

**Ausprobiert**  
— von Make: —

## Feminismus und Computer Kram

Podcast des Berliner Hackerinnenpace „Heart of Code“



Beim Podcast des Hackerinnenvereins „Heart of Code“ geben die Hackerinnen einen Einblick in das Miteinander und die Projekte in ihrem Space. Das Podcast-Projekt ist noch ganz frisch, bisher gibt es vier – sehr informative – Folgen. Darin erzählen sie unter anderem in der Folge „Wir haben uns nen Hackspace gemakert“, wie sie ihren Hackspace gegründet und sich ihre ungewöhnliche Räumlichkeit eingerichtet haben. In der zweiten Folge erzählt Leli von ihren Erfahrungen im *Recurse Center*, einem dreimonatigen Programmier-Stipendium in New York. Die weiteren Folgen geben einen Einblick in das Leben als Penetration-Testerin und den Bau einer geodätischen Kuppel.

Man bekommt im Laufe des Podcasts einen schönen Einblick, was für Frauen sich in so einem Hackerinnenspace tummeln, wie sie sich fortbilden und welche Projekte sie in Angriff nehmen. Jenny Genzmer moderiert den Podcast routiniert, lässt viele verschiedene Stimmen zu Wort kommen und auch an der Soundqualität gibt es nichts zu meckern. Dieser Podcast ist genau das richtige für alle, die ihren Hackspace gerade sehnsüchtig vermissen oder planen ihren eigenen, ganz besonderen Space aufzubauen. —rehu

URL [feminismusundcomputerkram.org](http://feminismusundcomputerkram.org)

## Heimwerken in der Fotografie

Kameras selber bauen – Objektive adaptieren

Schon beim ersten Durchblättern fallen die großzügigen, oft ganzseitigen Beispielbilder auf – und der fotoaffine Leser möchte sofort wissen, mit welcher Technik die zum Teil sehr ästhetischen Bilder entstanden sind. Dabei vermittelt das Wort „Heimwerken“ im Titel einen arg verharmlosenden Eindruck: Es geht um durchaus anspruchsvolle, präzise Feinarbeit – etwa beim Falten und Kleben eines Balgens für die selbstgebaute Laufbodenkamera mit Tilt-/Shift-Fähigkeiten, mit der man die Schärfenebene beliebig durchs Bild ziehen kann, oder bei der Konstruktion einer Schlitzblende aus Rasierklingen, mit der sich spektakuläre Aufnahmen von sich bewegenden Objekten machen lassen. Besonders umfangreich ist der Teil des Buches, der sich dem Adaptieren alter bis historischer Objektive an moderne Systemkameras widmet. Die Auswahl der Objek-



tiv-Empfehlungen beweist, dass der Autor etwas vom Fach versteht – und gibt dem Leser Ansporn, selbst auf die Suche nach solchen Schätzen zu gehen. Experimente mit Lupen und Linsen, Loch-, Stereo- und Mehrfach-Objektiven kommen ebenfalls nicht zu kurz. Obwohl der Autor eine ausführliche Anleitung zum Thema Objektive, Blenden und Linsen liefert, ist das Buch für Foto-

Anfänger eher weniger geeignet. —cm

*Hinweis: Dieses Buch wird von einem Verlag herausgegeben, der wie die Maker Media GmbH zur Heise Gruppe gehört.*

|        |                                  |
|--------|----------------------------------|
| Autor  | Cyrill Harnischmacher            |
| Verlag | dpunkt                           |
| Umfang | 252                              |
| ISBN   | 978-3-86490-800-2                |
| Preis  | 34,90 € (Buch), 27,99 € (E-Book) |

## Mikrocontroller ESP32

Das umfassende Handbuch

Das Prädikat „umfassend“ in der Unterzeile beschreibt den Inhalt des Buches sehr gut, denn es werden ziemlich alle Aspekte rund um den populären ESP32 detailliert behandelt, mit denen man in Berührung kommen könnte und darüber hinaus. Von der Vorstellung der Modulvarianten über den Umgang mit Breadboards, dem Bedienen von *Fritzing*, einem Lötkurs, der Übersicht von Entwicklungsumgebungen, einem Einstieg in C/C++, Vorschlägen für eine Werkstattausrüstung, der Erklärung der Funktionen des ESP32 bis zu IoT-Diensten und Projektvorschlägen ist alles vertreten.

Sogar eine Anleitung zum Auflöten der ESP-Module auf Adapterplatinen fehlt nicht. Hier zeigt sich aber auch ein Problem des durchgehend farbig bebilderten Buches: Es ist einfach zu viel des Guten. Für fortgeschrittene Maker fängt der neue und interessante Teil erst nach rund 250 Seiten bei den Pro-



grammbeispielen an. Einsteiger auf der Suche nach einem schnellen Erfolgserlebnis (LED-Blinken lassen wir hier mal nicht gelten) werden mit hoher Wahrscheinlichkeit davor schon aufgegeben haben.

Trotzdem werde ich mir das Buch in mein Regal stellen. Warum? Denkt man sich einige Kapitel wie Stromversorgung, Werkstatt und andere weg, ist das Buch nämlich ein sehr gutes Nachschlagewerk für immer wiederkehrende Fragen und —dab

Probleme.

*Hinweis: Dieses Buch wird von einem Verlag herausgegeben, der wie die Maker Media GmbH zur Heise Gruppe gehört.*

|        |  |
|--------|--|
| Autor  | Udo Brandes  |
| Verlag | Rheinwerk  |
| Umfang | 616 Seiten   |
| ISBN   | 978-3-8362-7445-6                                  |
| Preis  | 44,90 € (Buch), 39,90 € (E-Book), 49,90 € (Bundle) |

# Mach was mit Python & Raspberry Pi!

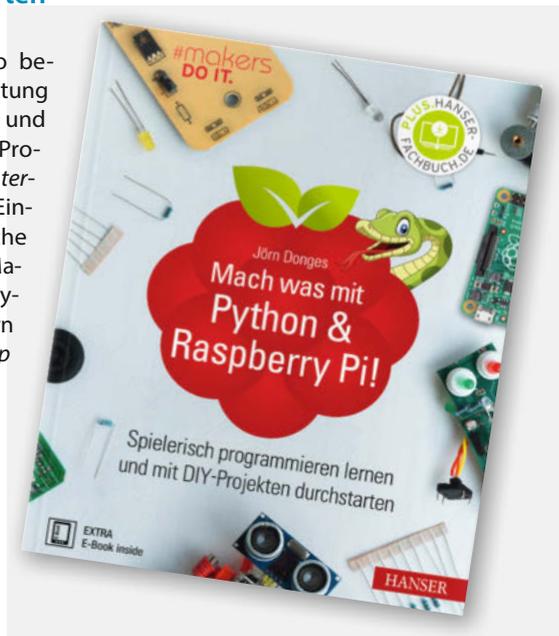
Spielerisch programmieren lernen und mit DIY-Projekten durchstarten

Raspberry Pi und Python – das passt. Zu Beginn war das Raspberry-Pi-Projekt auf die Förderung des Informatikunterrichts an Schulen und in Entwicklungsländern ausgerichtet – als Programmiersprache präferierte man Python. Später wurde das ursprüngliche Modell auch außerhalb seines Zielmarktes verkauft – speziell in der Maker-Szene.

Das Cover-Bild des Buches täuscht aber über den Inhalt. Auf dem Cover sieht man Bauteile, etwa Ultraschallsensor, LED und Platinen. Davon ist allerdings im Buch dann nichts mehr zu sehen. Der Schwerpunkt ist das Erlernen von Python und dem Programmieren. Nur ein Projekt – ein Webradio mit dem Raspberry Pi – geht ein bisschen in die Maker-Richtung - aber nur ein bisschen, denn selbst der Anschluss eines Lautsprechers ist über den eingebauten Soundchip und Klinkestecker des

Raspberry Pi denkbar simpel. Also beschränkt sich der Autor auf die Einrichtung der Software. Das ist jedoch sauber und Schritt für Schritt erklärt; plus der Programmierung eines Web- oder Tkinter-Interfaces. Fazit: ein gutes Python-Einsteiger-Buch – auch für Jugendliche geeignet, allerdings ohne richtige Make-Projekte. Dafür hat man beim Python-Lernen nicht nur Theorie, sondern praktische Ergebnisse. —anp

|               |                               |
|---------------|-------------------------------|
| <b>Autor</b>  | Jörn Donges                   |
| <b>Verlag</b> | Hanser                        |
| <b>Umfang</b> | 254 Seiten                    |
| <b>ISBN</b>   | 978-3-446-46150-5             |
| <b>Preis</b>  | 29,99 € (Buch), 23,99 € (PDF) |



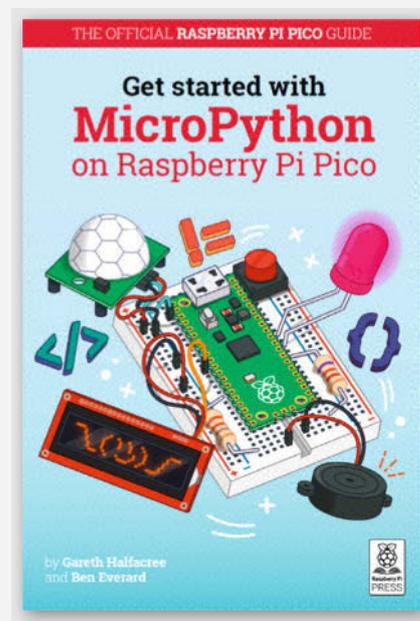
# Get started with MicroPython

on Raspberry Pi Pico

Es gibt einen neuen Player in der Mikrocontroller-Szene: den *Raspberry Pi Pico* (siehe auch Seite 8 und unser Editorial). Programmiert wird mit der GCC-Toolchain oder MicroPython und anders als bei beiden bisherigen Raspi-Einplatinenrechnern gibt es kein Betriebssystem. Da die bisher vorliegenden Anleitungen für Raspis beim Pico nicht helfen, besteht Bedarf an Anleitungen für den Einstieg und für Beispielprojekte drumherum. Mit „The official Raspberry Pi Pico Guide“, so der Obertitel, liegt bereits

das erste Buch zum Pico vor. Als PDF steht es zum kostenlosen Download bereit. Darin finden des Englischen mächtige Einsteiger – vom Jugendlichen bis zum Erwachsenen – neben Grundsätzlichem zu MicroPython vor allem praktische Anleitungen zu Maker-Themen.

Los geht es mit einer Vorstellung des Pico, dann schließen sich Tipps zum Löten an – schließlich wird die Platine ohne Pinleiste geliefert. Es folgen Installation und eine Kurzeinführung in MicroPython, bei der es allerdings hilfreich ist, wenn man schon etwas Erfahrung in anderen Programmiersprachen hat. Ab Kapitel 3 (“Physical Computing“) werden bereits externe Komponenten wie Taster, LEDs, Sensoren und Summer angeschlossen. Es folgen mehrere Projekte, deren Auswahl aber eher auf Jugendliche ausgerichtet ist, etwa eine Spielzeugampel oder ein Reaktionspiel. Ein Anhang liefert technische Daten und Pinout-Schema, zeigt aber auch, wie man einen Neopixelstreifen anschließt. —anp



|                |  |
|----------------|--|
| <b>Autoren</b> | Gareth Halfacree und Ben Everard                 |
| <b>Verlag</b>  | Raspberry Pi Press                               |
| <b>Umfang</b>  | 139 Seiten                                       |
| <b>ISBN</b>    | 978-1-912047-86-4                                |
| <b>Preis</b>   | 10 £ (Buch, rund 11,30 €), PDF gratis            |
| <b>URL</b>     | hackspace.raspberrypi.org/books/micropython-pico |



# IMPRESSUM

**Make:** Nächste Ausgabe erscheint am 8. April 2021

## Redaktion

**Make: Magazin**  
Postfach 61 04 07, 30604 Hannover  
Karl-Wiechert-Allee 10, 30625 Hannover  
Telefon: 05 11/53 52-300  
Telefax: 05 11/53 52-417  
Internet: [www.make-magazin.de](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 xxx@make-magazin.de oder xxx@make-magazin.de. Setzen Sie statt „xx“ oder „xxx“ bitte das Redakteurs-Kürzel ein. Die Kürzel finden Sie am Ende der Artikel und hier im Impressum.

**Chefredakteur:** Daniel Bachfeld (dab) (verantwortlich für den Textteil)

**Stellv. Chefredakteur:** Peter König (pek)

**Redaktion:** Heinz Behling (hgb), Helga Hansen (hch), Carsten Meyer (cm), Rebecca Husemann (rehu), Andreas Perband (anp), Elke Schick (esk)

**Mitarbeiter dieser Ausgabe:** Beetlebum (Comic), Thomas Euler, Josha von Gizycki, Florian Grimm, Jens Hackel, Detlef Heinze, Michael Jentsch, Robert Kränzlein, Matthias Mett, Hermann Mohr, Manuel Mohr, Niq Oltman (Übersetzung), Stella Risch (stri), Billie Ruben, Tim Sway, 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:** Anna Hager, 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

**Geschäftsführer:** Ansgar Heise, Dr. Alfons Schröder

**Verlagsleiter:** Dr. Alfons Schröder

**Stellv. Verlagsleiter:** Daniel Bachfeld

**Anzeigenleitung:** Michael Hanke (-167) (verantwortlich für den Anzeigenteil), [mediadaten.heise.de/produkte/print/das-magazin-fuer-innovation](mailto:mediadaten.heise.de/produkte/print/das-magazin-fuer-innovation)

**Leiter Vertrieb und Marketing:** André Lux (-299)

**Service Sonderdrucke:** Julia Conrades (-156)

**Druck:** Dierichs Druck + Media GmbH & Co.KG, Frankfurter Str. 168, 34121 Kassel

**Vertrieb Einzelverkauf:**

VU Verlagsunion KG  
Meßberg 1  
20086 Hamburg  
Tel.: 040/3019 1800, Fax.: 040/3019 145 1800  
E-Mail: [info@verlagsunion.de](mailto:info@verlagsunion.de)  
Internet: [www.verlagsunion.de](http://www.verlagsunion.de)

**Einzelpreis:** 12,90 €; Österreich 14,20 €; Schweiz 25.80 CHF; Benelux 15,20 €

**Abonnement-Preise:** Das Jahresabo (7 Ausgaben) kostet inkl. Versandkosten: Inland 77,00 €; Österreich 84,70 €; Schweiz/Europa: 90,65 €; restl. Ausland 95,20 €

Das Make-Plus-Abonnement (inkl. Zugriff auf die App, Heise Magazine sowie das Make-Artikel-Archiv) kostet pro Jahr 6,30 € Aufpreis.

## Abo-Service:

Bestellungen, Adressänderungen, Lieferprobleme usw.:

**Maker Media GmbH**  
Leserservice  
Postfach 24 69  
49014 Osnabrück  
E-Mail: [leserservice@make-magazin.de](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 Helferlein würdest Du Dir bauen oder mitnehmen, wenn Du außerhalb der Zivilisation leben würdest?



### Florian Grimm

Kirchentellinsfurt, hat die Hydroponik-Anlage auf Seite 36 gebaut

Unbedingt eine Erweiterung meiner Hydrokultur zum Kaffeeanbau – am besten mit Brau- und Transferfunktion zur Hängematte!



### Andreas Perband

Buch am Erlbach, seit 2021 frischgebackenes Mitglied der Make-Redaktion

Wenn das Lebensnotwendige wie Unterkunft, Wärme und Verpflegung da ist, würde ich eine Wärmebildkamera basteln - um nachts mitzukriegen, was um mich los ist. Auf Steinzeitlevel zurückgeworfen wäre es wohl eher der Faustkeil und ein Tipi.



### Hermann Mohr

Heilbronn, beschreibt zusammen mit Manuel Mohr den Photostick auf Seite 52

Da ich vermutlich keinen Strom habe, würde ich mich mit genügend Filmen eindecken und meine alten Analogkameras aus der Vitrine reaktivieren. Ohne Foto außerhalb der Zivilisation - das geht gar nicht.



### Manuel Mohr

Karlsruhe, beschreibt zusammen mit Hermann Mohr den Photostick auf Seite 52

Nach anfänglicher Begeisterung über den Abenteuerurlaub würden sich meine Anstrengungen schnell darauf konzentrieren, zurück zur Zivilisation zu finden – ein Kompass wäre da bestimmt hilfreich.

## Inserentenverzeichnis

|   |     |  |    |
|---|-----|--|----|
| Fachhochschule Wedel gGmbH, Wedel .....     | 9   | Technische Universität Clausthal, Goslar ..... | 43 |
| Reichelt Elektronik GmbH & Co., Sande ..... | 148 | <b>Make:markt</b> .....                        | 19 |
| Sauter GmbH, Inning .....                   | 77  |  |    |

# AUS GUTEM HOLZ GESCHNITZT!

Jetzt  
portofrei bestellen!



Auch als  
PDF zum  
Download!

## Make „Loslegen mit Holz“

Das Make-Sonderheft „Loslegen mit Holz“ hilft Ihnen speziell bei den ersten Schritten im Umgang mit Holzwerkstoffen, Voll- und Leimholz und zeigt etwa, wie man Holz mit Schrauben, Leim, Dübeln oder Nut und Feder einfach, aber solide verbindet. Sie erfahren, welches das richtige Holz für Ihr Projekt ist (und warum) und welche Werkzeuge Sie für den Einstieg wirklich brauchen. Außerdem sieben Holzprojekte zum Nachbauen und noch vieles mehr.

Make „Loslegen mit Holz“ auch digital als PDF-Download für 9,99 € verfügbar.

Heft hier portofrei für 10,90 € bestellen:  
[shop.heise.de/make-holz2020](http://shop.heise.de/make-holz2020)

Noch mehr Lesestoff rund ums Make-Magazin finden Sie hier:  
[shop.heise.de/make-hefte](http://shop.heise.de/make-hefte)

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

Heise Medien GmbH & Co. KG | heise shop | Karl-Wiechert-Allee 10 | 30625 Hannover | Telefon: +49[0] 02152 / 915 229 | E-Mail: [service@shop.heise.de](mailto:service@shop.heise.de)

 **heise shop**

[shop.heise.de/make-holz2020](http://shop.heise.de/make-holz2020)





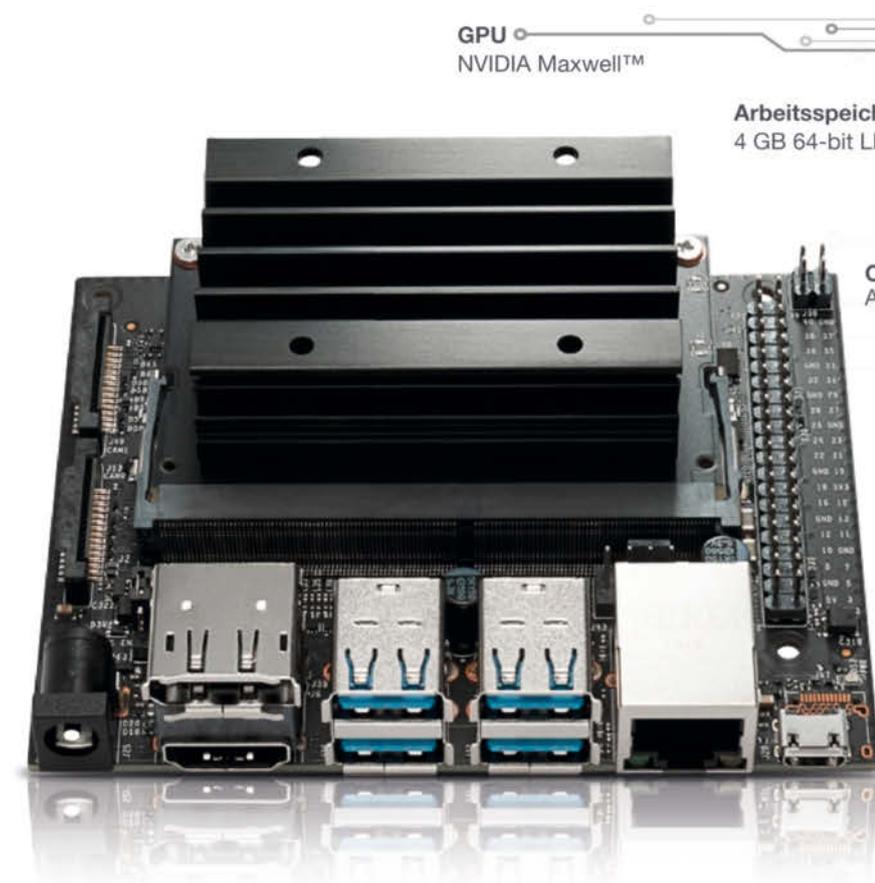
# UNSER SORTIMENT VON MAKERN FÜR MAKER

The best part of your project: [www.reichelt.de](http://www.reichelt.de)

MODERNE KI FÜR OBJEKTERKENNUNG & SPRACHVERARBEITUNG

## JETSON NANO™ – ENTWICKLERKIT

High Performance Computing, mit 472 GFLOPs bei nur 5 bis 10 Watt.



GPU  
NVIDIA Maxwell™

Arbeitsspeicher  
4 GB 64-bit LPDDR4

CPU  
ARM Cortex-A57

Kamera  
2x MIPI CSI-2 DPHY

Bestell-Nr.: JETSON NANO KIT

**109,95**

Mehr unter [reichelt.de/ki](http://reichelt.de/ki)

■ Top Preis-Leistungs-Verhältnis

■ über 110.000 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: 14. 1. 2021