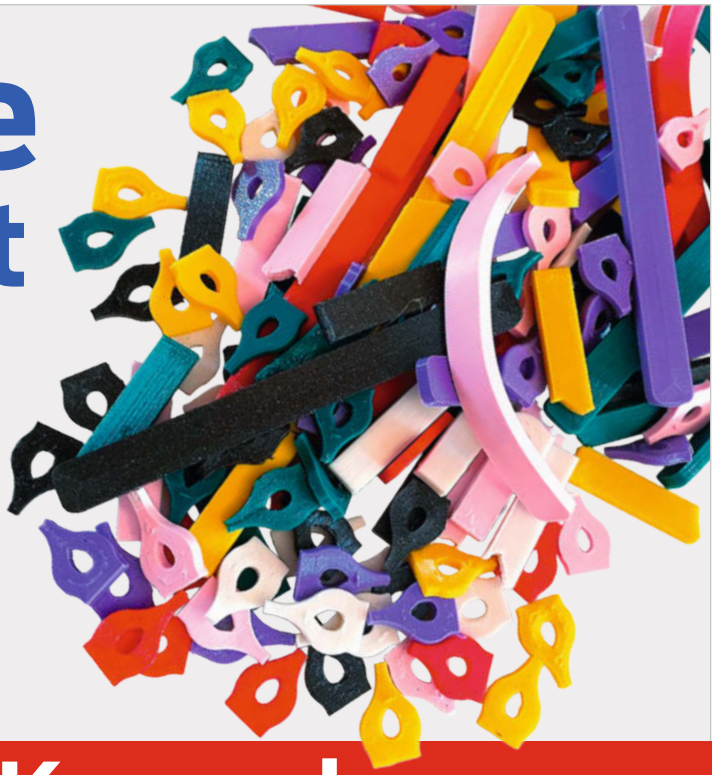
DEUTSCHLANDS GEFÄHRLICHSTES DIY-MAGAZIN

c't Make:

Make:

Exklusiv: Interview mit RasPi-Erfinder Eben Upton



Filamente im Stresstest

- 3D-Materialien verschiedener Hersteller im Vergleich
- Zug- und Bruchfestigkeit, Temperaturbeständigkeit
- Welches Filament für welchen Zweck

Projekte

- IoT-Gaszähler upgraden
- Steckerleiste per WLAN steuern
- LED-Couchkissen

Know-how

- Fernzugriff auf den Pi
- FreeCAD mit Add-ons erweitern
- ESP32-Coprozessor programmieren

Teilefinder

- Schubladen mit Signal-LEDs
- Finden per Handy-App

Sprachsteuerung fürs Smart Home

- Mit Home Assistant, ESP32-Pico und Mikrofon
- Programmieren und einrichten

5/23
29.9.2023
CH CHF 26.50
AT 14,90
Benelux 15,90
€ 13,50

© Copyright by Maker Media GmbH.

Elektronikgehäuse | u-maker Box

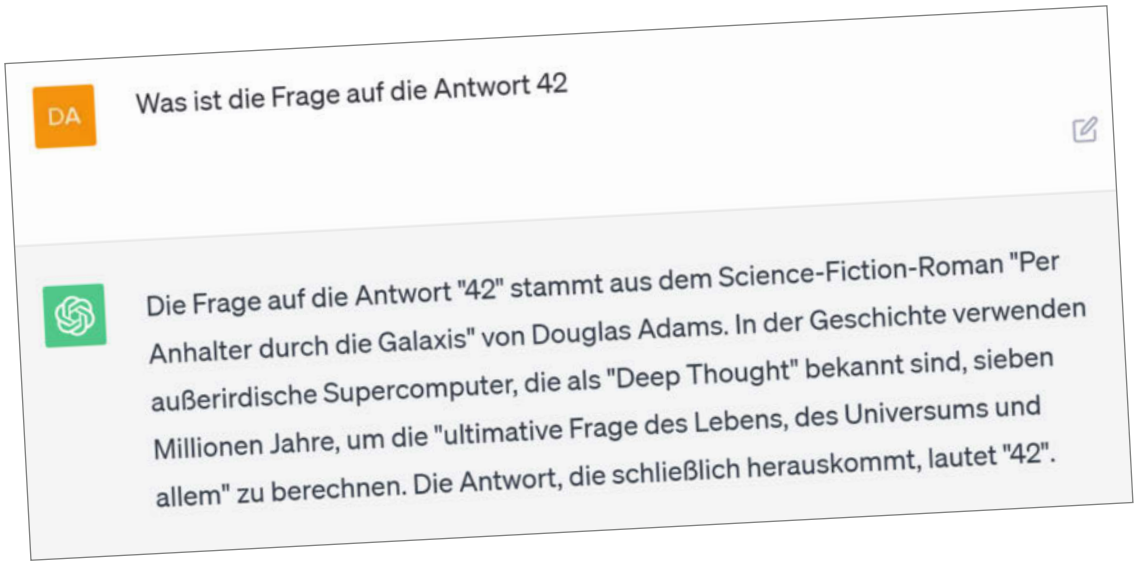
Das individuelle Raspberry Pi Gehäuse für Maker

- Voll flexibel, modular, erweiterbar und smartes Design
- 3D Druckvorlagen für Zubehör
- Automation- und IIoT-Software vom Industrieprofi

Mehr Informationen:
www.weidmueller.de/u-maker

© Copyright by Maker Media GmbH.

Weidmüller



ChatGPT, das bessere Google

Kein Tag vergeht, an dem keine Hype-Meldung oder Warnmeldung über ChatGPT erscheint, das für die meisten Menschen mittlerweile als Synonym für das gesamte Feld der Künstlichen Intelligenz steht. Mal rettet ChatGPT angeblich ein Kind mit einer richtigen Diagnose gerade noch rechtzeitig, mal wälzt es künftig den Unterricht in Schulen um, ein andermal macht es Übersetzer und Journalisten bald überflüssig. Im Gegenzug würden Propagandisten damit mächtiger und Kriminelle raffinierter. Und zwischendrin fordern angebliche Experten ein Moratorium, um nicht die Kontrolle über die Entwicklung der KI zu verlieren. KI sei wie eine Atombombe.

Dass die ChatGPT zugrunde liegenden Large Language Models mit ihrer statistischen Wahrscheinlichkeit beeindruckende Texte schreiben können, ist unstrittig. Dass man daraus die Rettung oder den Untergang der Menschheit ableitet, ist die übliche Reaktion, wie sie bei jedweder neuen, massentauglichen Technik zu registrieren ist. Früher waren das Kino und Fernsehen, dann Personal Computer und Roboter, später Internet und Smartphone. Und jedes Mal versichern Unkenrufer aufs Neue, jetzt sei alles ganz anders und wirklich sehr viel dramatischer, ich schwör.

Alles was ChatGPT generiert, ist längst in irgendeiner Form im Internet präsent. Was es liefert, sind im Grunde nur Echos dessen, was Menschen schon geschrieben, gezeichnet oder aufgenommen haben. ChatGPT schreibt einfach nur gute Zusammenfassungen. Quasi ein Super-Wikipedia. Wo es nicht weiter weiß, beginnt es zu halluzinieren vulgo zu raten, Statistik eben.

Make-Leser wissen, dass KI viel mehr als nur ChatGPT ist und wie man damit spannende und nützliche Projekte umsetzt. Wer das System hinter aktuellen KIs verstanden hat, erkennt, dass ChatGPT weder interpretieren noch kreativ sein kann, aber prima als Recherche-Assistent, Brainstorming-Partner und Programmier-Copilot fungieren kann. Wer ein gutes Prompt formuliert, bekommt auch verwertbare Antworten. Das ist bei Google ja genauso.

Welches ChatGPT-Prompt hat Ihnen bei der Lösung eines Problems oder Umsetzung eines Projekts geholfen? Oder wo waren Sie enttäuscht? Schreiben Sie uns!

Happy Hacking

Daniel Bachfeld

▶ make-magazin.de/x3w2

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

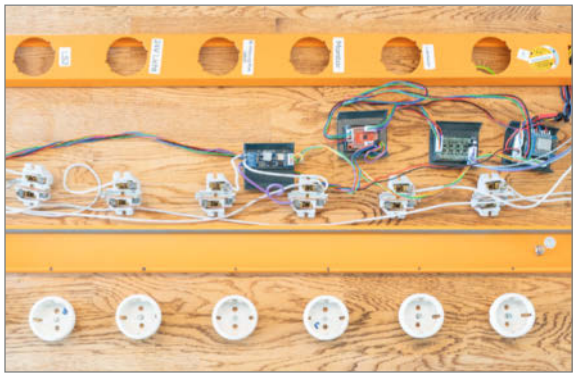
© Copyright by Maker Media GmbH.

Inhalt

Projekte

Per Handy schaltbare Steckdosen waren unserem Autor nicht genug. Seine Mehrfach-Steckdose schaltet Geräte abhängig von seiner Anwesenheit oder der Temperatur. Apropos Temperatur: Mit zusätzlichen Sensoren machen wir den ioBroker-Gaszähler aus Ausgabe 3/23 noch intelligenter. Und wenn Sie es dann warm in der guten Stube haben, sorgen die LED-Kissen für ein farbenfrohes Ambiente.

18 LED-Kissen mit dem Pixelblade Controller
48 Temperatursensoren für den ioBroker-Gaszähler
54 Schlaue Steckdose



Filamente im Stresstest

3D-Druck-Filamente gibt es fast so viele wie Sand am Meer und in allen Preislagen. Doch welches Material ist stabil genug für ein Projekt, PLA, PETG oder ASA? Welche Biege- und Zugkräfte kann man Filamenten zumuten? Wie schlagfest sind die Kunststoffe? Dieser Artikel zeigt nicht nur, wie man das und vieles mehr herausfindet, sondern stellt auch Selbstbau-Testgeräte vor, die den Kunststoffwürmern so richtig Stress machen, um die Spreu vom Weizen zu trennen.

8 Filamente im Stresstest

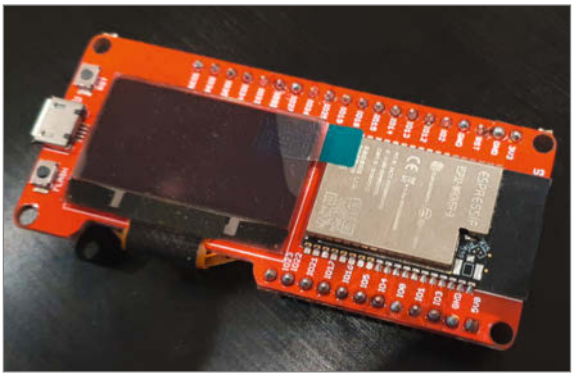
3 Editorial
6 Leserforum, Comic
8 Test: Filamente im Stresstest
18 Projekt: LED-Kissen mit dem Pixelblade Controller
28 Bauen, tüfteln, basteln: Die Maker Faire Hannover
32 Report: Interview mit Pi-Erfinder Eben Upton
38 Projekt: Smarte Werkstattboxen
47 In eigener Sache
48 Projekt: Temperatursensoren für den ioBroker Gaszähler
54 Projekt: Schlaue Steckdose
64 Community-Projekt: POV-Display aus CD-Spieler
66 Community-Projekt: Das Balkonkraftwerk im Blick
68 Community-Projekt: DIY-Holoprojektor

© Copyright by Maker Media GmbH.

Know-how

Ein kleiner Begleiter im ESP32 namens ULP-Prozessor kann dessen Stromhunger um 99 Prozent senken. Wie das geht, erfahren Sie hier. Außerdem geht es um den Zugriff auf Raspberry-Pi-Server, die ohne Monitor und Tastatur im Stillen werkeln, ab und zu aber doch Zuwendung brauchen. Schließlich zeigen wir Ihnen noch, wie Sie FreeCAD aufmöbeln, beschleunigen, verbessern oder einfach nur chic machen können.

72 Verbindungen zum Raspberry Pi Server
104 PowerToys für FreeCAD
112 ESP32: Wie ULP den Stromverbrauch senkt



Smarte Werkstattboxen

Kleinteilemagazine für all die winzigen Platinen, Sensoren und sonstigen Bauteile sind im Maker-Alltag sehr praktisch, aber etwas darin wiederzufinden kann selbst bei vollständiger Beschriftung aller Fächer zeitraubend sein. Diesem ESP32-gesteuerten Werkstatt-Organizer sagen Sie per Smartphone-App, was Sie suchen, und er weist Ihnen per LED den Weg zum richtigen Schublädchen. Da geht Ihnen ein Licht auf!

38 Smarte Werkstattboxen

72 Workshop: Verbindungen zum Raspberry Pi Server
78 Test: His Master's Voice: Arduino hört mit
84 Projekt: Communicator für Home Assistant
90 Reingeschaut: Bildschirmspiel 1
92 Tipps & Tricks
96 Know-how: 3D-Drucker mit Klipper optimieren
104 Report: PowerToys für FreeCAD
110 Bücher zu FreeCAD
112 ESP32: Wie ULP den Stromverbrauch reduziert
118 Kurzvorstellungen: Orange Pi Compute Module, 3D-Drucker AnkerMake M5C, Flexkabel-Display-Verbinder, Displays mit Touchscreen, Arduino-Board, IoT-Starterkit, Taschen-Oszilloskop, ESP32-S3-Boards
122 Impressum/Nachgefragt

Themen von der Titelseite sind rot gesetzt.

Communicator für Home Assistant

Wenn Captain Jean-Luc Picard auf der Enterprise sich an die Brust fasst, ist das kein Zeichen eines Herzinfarkts, sondern er aktiviert seinen Communicator, um Befehle zu erteilen. Unser kleiner Sprachassistent fürs Smarthome wird ähnlich bedient: Erst ein Druck auf seine Oberfläche lässt ihn zuhören, Sprachbefehle empfangen und veranlasst den Home-Assistant-Server, sie auszuführen.

84 Communicator für Home Assistant

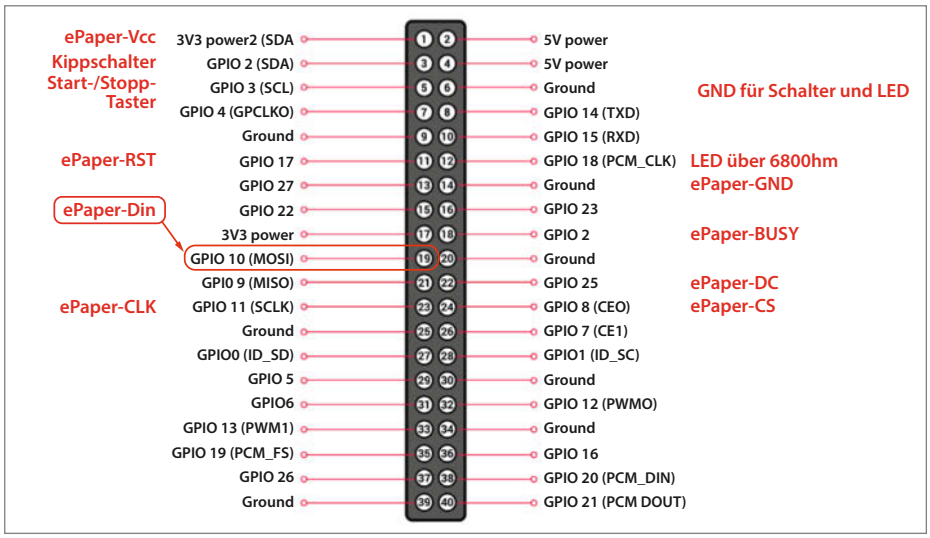
© Copyright by Maker Media GmbH.

Leserforum

Pin vertauscht

Die Backup-Station, Make 4/23, S. 22

Bei diesem Artikel ist uns ein klassischer Druckfehler unterlaufen, sprich: Im Manuskript des Autors war alles noch korrekt, aber bei der Umsetzung fürs Heft ist bei der Abbildung der Pinbelegung die Beschriftung für „ePaper-Din" verrutscht. Din gehört an MOSI, GPIO10, Pin 19. Tut uns leid!



Der Din-Anschluss des ePaper-Displays für die Backup-Station gehört an Pin 19.

Alternative Gehäuse

Ich wollte mich gerade an den Bau der Backup-Station aus der Make 4/23 machen. Eine gute Idee und ein ebenso guter Artikel! Allerdings bin ich bei der Beschaffung der Teile bereits auf große Probleme gestoßen. Sie beschreiben den Einbau in ein altes 5,25"-Floppy-Gehäuse, das aber sicher die Wenigsten noch zuhause haben dürften. Mein letztes habe ich z.B. im letzten Jahr entsorgt .

Bodo Eich

Unser Autor Ralf Stoffels hatte das Gehäuse noch im Fundus. Die Konstruktion passt auch in handelsübliche Universal-Gehäuse wie unter dem Link.

▶ make-magazin.de/xwwh

Händlerliste

Editorial „Nachfrage erhöhen", Make 4/23, S. 3

Auch dieses Editorial hat mich angesprochen. Wir Maker sollten eine Vorreiter-Rolle spielen, was Nachhaltigkeit und Umweltprobleme angeht. Also: am liebsten keine Billig-Produkte aus China, sondern europäische Fertigung. Dass dies schwierig ist, ist mir klar, und ich gestehe, dass ich auch nicht immer in diesem Sinne kaufe. Ein erster Schritt ist auf jeden Fall, europäische Händler zu bevorzugen gegenüber Amazon und Alibaba. Vielleicht könntet ihr mal eine Liste solcher Alternativen publizieren?

Jean-Claude Feltes

Nutriscore

Technikbegeisterte Menschen sehen gerne über die gesellschaftlichen Bedingungen ihres Hobbys hinweg. Deshalb begrüße ich es sehr, dass Sie auf das Dilemma aufmerksam machen, dass es nicht ganz einfach ist auf Produkte zu verzichten, die in China produziert werden. Auch wenn ein völliger Verzicht auf chinesische Produkte schwierig sein dürfte, kann ja ein Umstieg auf verfügbare regional produzierte Alternativen schon ein erster Schritt sein und durch die Nachfrage einen Markt für Anbieter aus der näheren Umgebung schaffen. Sie könnten als Fachzeitschrift einen Beitrag hierfür leisten. Für den einzelnen Bastler ist es sicher mühsam für die benötigten Bauteile eine regional produzierte Alternative zu finden. Amazon hilft da jedenfalls nicht. Mein Vorschlag: Richten Sie auf Ihrer Internetseite einen Bereich ein, in dem Sie solche Alternativen auflisten. So hätten Ihre Leser einen Anlaufpunkt für eine Suche.

Kontakt zur Redaktion

Leserbriefe bitte an:

heise.de/make/kontakt/

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

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

www.make-magazin/forum

instagram.com/MakeMagazinDE

www.facebook.com/MakeMagazinDE

pinterest.com/MakeMagazinDE

www.twitter.com/MakeMagazinDE

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.

© Copyright by Maker Media GmbH.

Wahrscheinlich werden Leser Ihnen bald auch weitere Produkte melden, die sie ausfindig gemacht haben. Wie wäre es mit einem Nachhaltigkeits-Score in Anlehnung an den Nutri-Score im Tiefkühlregal?

Günther Fischer

Die Idee einer Online-Übersicht von Anbietern auf unseren Seiten ist klasse. Gerne nehmen wir Hinweise auf allen unseren Kanälen entgegen.

Entsetzen

Entsetzt musste ich in der Ausgabe 4/23 feststellen, dass die Autoren und auch Redakteure ihrer Zeitschrift anscheinend nur ihre eigenen Artikel im Blick haben. Im Editorial wird über die Nachhaltigkeit bzgl. der Maker-Community philosophiert, um dann auf Seite 108 für die Fertigung von Prototypen aus China zu werben. Das erinnert mich ein wenig an die Klimakleber, die sich nach ihrer Aktion in den Ferienflieger nach Thailand setzen. Das war dann wohl meine letzte Ausgabe der Make.

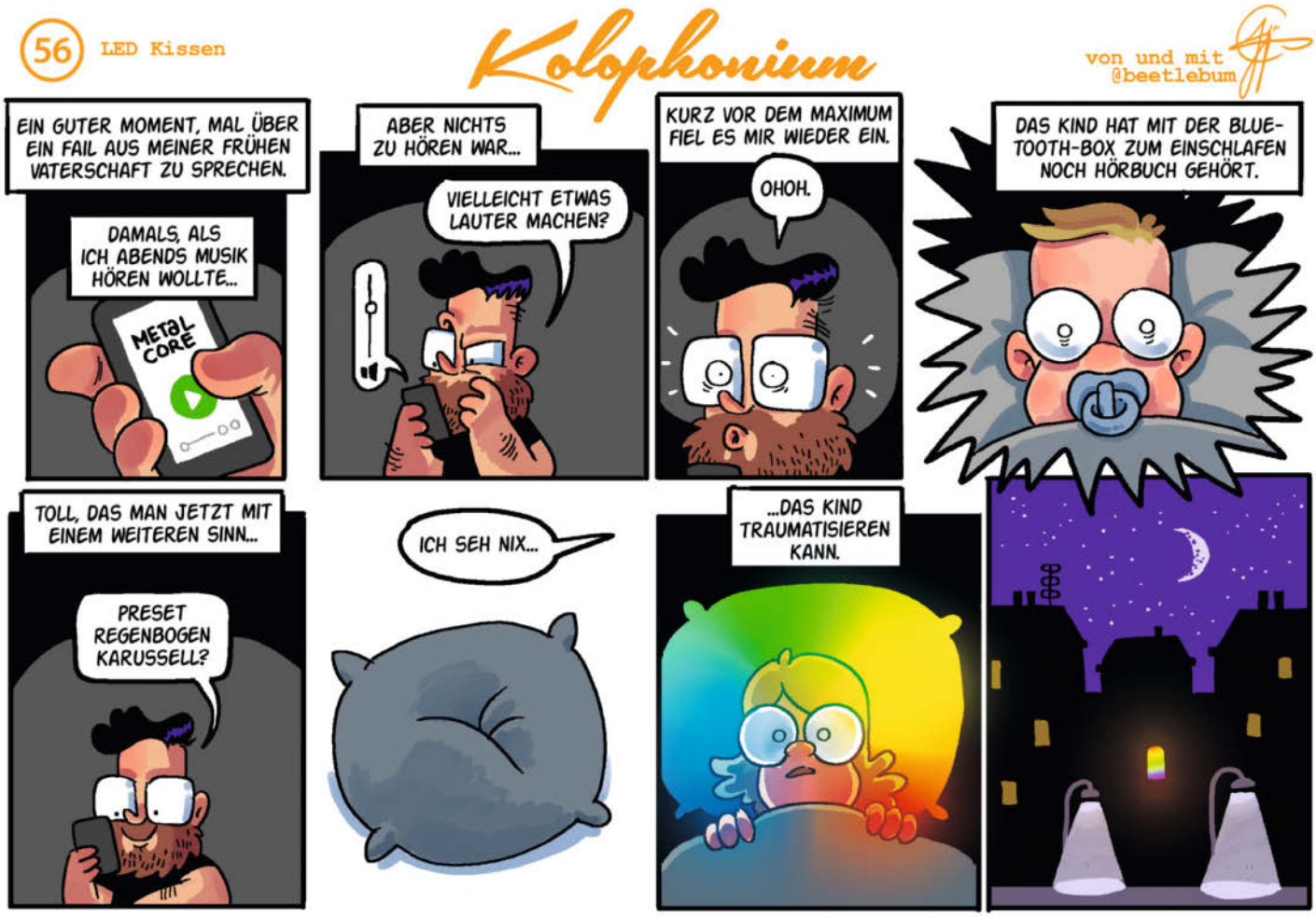
Detlef Woge

Hingucker

Radio mit ESP32, Make 2/23, S. 16

Vielen Dank für dieses tolle Projekt. Aufgrund der guten Beschreibung konnte ich das Projekt auch selber sehr leicht umsetzen, wobei ich auf den UKW Empfang verzichtet habe. Hier – wie gewünscht – zwei Fotos. Ich finde Holz ist immer noch ein Hingucker. Außerdem habe ich (noch) keinen 3D-Drucker. Die Holzkiste habe ich in einem schwedischen Möbelhaus in der Badeabteilung gefunden. Als Bastler sieht man/Frau die Sachen halt anders. Die Frontseite habe ich mit selbstklebender Folie versehen. Das Radio ist immer dort, wo ich im Hintergrund Musik höre - beispielsweise beim nächsten Projekt.

Peter



© Copyright by Maker Media GmbH.

Filamente im Stresstest

Welches 3D-Druck-Filament eignet sich für meine Anwendung am besten? Wie stark kann das fertig gedruckte Bauteil im Anschluss belastet werden? Dieser Artikel stellt DIY-Testgeräte vor und wir untersuchen und vergleichen damit beliebte Filamente unterschiedlicher Hersteller auf ihre mechanischen Eigenschaften.

von Jan Wegener

© Copyright by Maker Media GmbH.

Mit 3D-Druckern lassen sich unter anderem schöne Dinge wie Figuren und Dekorationsgegenstände herstellen. Für Maker sind hingegen maßgeschneiderte Gehäuse, mechanische Bauteile oder Ersatzteile die häufigsten Anwendungen – die Möglichkeiten sind nur durch das Druckverfahren begrenzt.

Auch ich setze bei fast allen Projekten auf den 3D-Druck und verwende dabei viel recyceltes PETG. Da in den einschlägigen Foren wenig darüber diskutiert wird und einer der größten 3D-Druck-Shops auf Anfrage mitteilte, dass nur knapp 7–8% der verkauften Filamente recycelt sind (Stand Januar 2022), wollte ich untersuchen, ob es signifikante Unterschiede zwischen Recycling-PETG und konventionell hergestelltem PETG-Filament gibt.

Deshalb habe ich im Rahmen eines Jugendforscht-Projektes begonnen, eine Reihe von Testverfahren zu entwickeln, um die wichtigsten mechanischen Eigenschaften der Druckmaterialien zu ermitteln. Das Projekt überzeugte im Wettbewerb auch auf Landesebene und motivierte mich, die Prüfvorrichtungen zu verbessern und weitere Filamente zu testen.

Seitdem lade ich regelmäßig Testgeräte zum Nachbau auf Printables hoch (siehe Links in der Kurzinfo) und zeige Materialtests und -vergleiche auf meinem YouTube-Kanal @JanTec3D (Links in der Kurzinfo).

Auch auf der Maker Faire Ruhr 2023 stieß das Projekt auf reges Interesse und ich wurde oft nach dem besten Filament gefragt. Da sich diese Frage, wie so viele andere, nicht pauschal beantworten lässt, werde ich in diesem Artikel einen Überblick über die wichtigsten mechanischen Eigenschaften von 3D-Druck-Kunststoffen geben und einige der beliebtesten Materialien testen und vergleichen.

Dazu gibt es Einblicke in den Aufbau von Testgeräten, die wichtigsten Maßeinheiten in der Materialprüfung, Tipps für die richtige Auswahl von Filamenten und den richtigen Umgang mit den Datenblättern der Filamenthersteller.

Im Folgenden werden die für den Vergleich verwendeten Prüfgeräte kurz vorgestellt. Allen Verfahren gemeinsam ist, dass die 3D-gedruckten Proben im Namen der Wissenschaft bis zur Zerstörung belastet werden.

Kurzinfo

» Beliebte 3D-Druck-Filamente und Materialien im Vergleich
» Praktische Testtheorie und Eigenbau-Testgeräte
» Diskussion der Ergebnisse in Bezug auf den Maker-Alltag

Checkliste

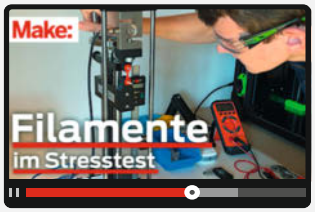
Zeitaufwand:
ab 2 Stunden für einfache Versuche

Kosten:
ab 20 Euro

Alles zum Artikel im Web unter *make-magazin.de/x5ru*

Mehr zum Thema

» Gerd Michaelis und Stella Maria Risch, 3D-Drucke clever kleben, Make 6/20, S. 114
» Robert Kränzlein, Filamentwaage für 3D-Drucker, Make 4/19, S. 100
» Heinz Behling, 30 Filamente für jeden Zweck, Make 2/19, S. 8
» Filementtest-Video des Autors



Der Zugversuch

Mit dem Zugversuch kann die Festigkeit (siehe auch Kasten Kleine Werkstoffprüfungskunde) von Werkstoffen untersucht werden. Eine hohe Festigkeit ist vor allem bei Bauteilen wichtig, auf die nach außen gerichtete Kräfte wirken, die also an ihnen ziehen. Beispiele dafür sind Befestigungen wie Haken oder Aufhängungen.

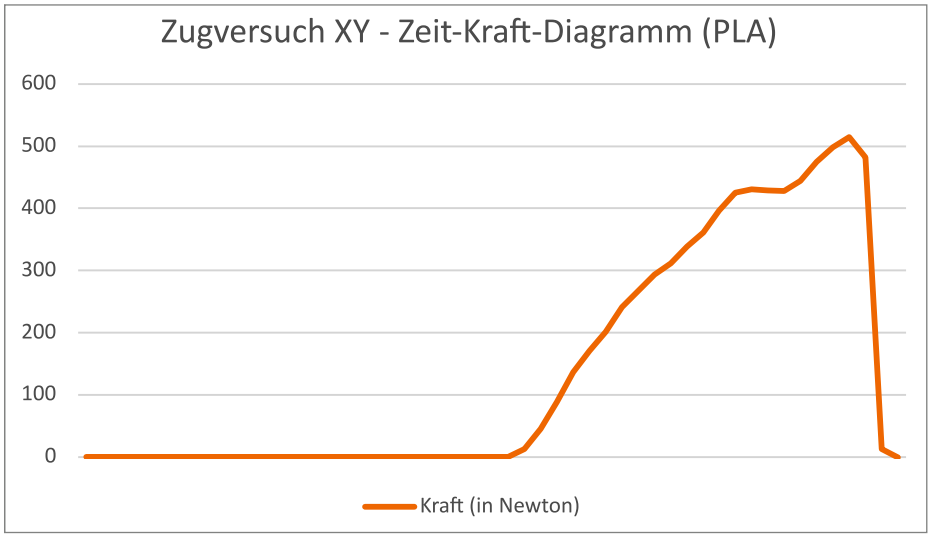
Für meinen Zugversuch habe ich einen manuellen Kurbelprüfstand mit einer Wägezelle und 3D-gedruckten Halterungen ausgestattet. Für diesen Apparat können Proben sowohl liegend als auch stehend gedruckt werden, um sowohl die reine Zugfestigkeit in XY-Druckrichtung als auch die Festigkeit in Z-Richtung, die maßgeblich durch die Schichthaftung bestimmt wird, zu untersuchen.

Während des Versuchs liest ein Arduino über ein Sparkfun HX711 Signalverstärkerboard (siehe auch Mehr zum Thema) die Spannung an der Wägezelle aus und rechnet diese in die Kraft (in Newton) um, die dann über die Zeit aufgezeichnet wird.

Am Scheitelpunkt dieser Kurve kann die maximale Zugkraft vor dem Bruch der Probe abgelesen werden. Anschließend wird die Bruchfläche der Probe gemessen. Die maximale Zugkraft dividiert durch die Bruchfläche ergibt die Zugfestigkeit in MPa (siehe Kasten).

Zugversuch einer Probe

© Copyright by Maker Media GmbH.



Ein typisches Zeit-Kraft-Diagramm im Zugversuch

3-Punkt-Biegeversuch

Mit einem 3-Punkt-Biegeversuch kann der Biegemodul (siehe Kasten) bestimmt werden, der als Maß für die Steifigkeit angesehen werden kann. Vereinfacht ausgedrückt handelt es sich um den Widerstand gegen elastische Verformung.

Bei Befestigungen von Druckköpfen oder Spindeln auf Linearschlitten ist meist eine besonders hohe Steifigkeit erwünscht, damit das Werkzeug möglichst präzise bewegt werden kann und nicht schwingt oder nachgibt. Bei Smartphone-Hüllen beispielsweise kann eine eher geringe Steifigkeit von Vorteil sein, die eine gewisse Durchbiegung zulässt, um das Gerät einlegen zu können und Stöße abzufangen.

Bei der Prüfung wird eine 10cm lange Probe an zwei Punkten abgestützt und in der Mitte mit einem Ring ein Behälter befestigt, um mittels eingelegten Gewichten eine Kraft auszuüben. Mit einer Messuhr wird dann die elastische Verformung der Probe bei verschiedenen Gewichten (entsprechend der Kraft) sehr genau bestimmt – im Rahmen der technischen und finanziellen Möglichkeiten. Es wird eine Messung für jedes nachgelegte Gewicht (hier sechs) durchgeführt und separat das Biegemodul errechnet.

Konkret lässt sich aus den Abmessungen des Probekörpers, der aufgebrachten (durch Gewichte bestimmten und bekannten) Kraft und der gemessenen Durchbiegung der Elastizitätsmodul E mit Hilfe der Formel aus der Abbildung berechnen.

$$E = \frac{P \cdot L^3}{4 \cdot w_0 \cdot a^3 \cdot b}$$

Dabei ist P die Kraft in Newton, die durch die Gewichte auf die Probe wirkt, L die Länge der Probe (hier 100mm), w_0 die gemessene Durchbiegung in Millimeter, a die Dicke der Probe (hier 4 mm) und b die Breite der Probe (12 mm). Alle Berechnungen werden in einer großen Excel-Tabelle durchgeführt, die als Referenz in den Downloads zum Artikel zu finden ist.

Diese Werte können dann auch mit Werten in der Literatur und Datenblättern anderer Werkstoffe oder Materialien verglichen werden.

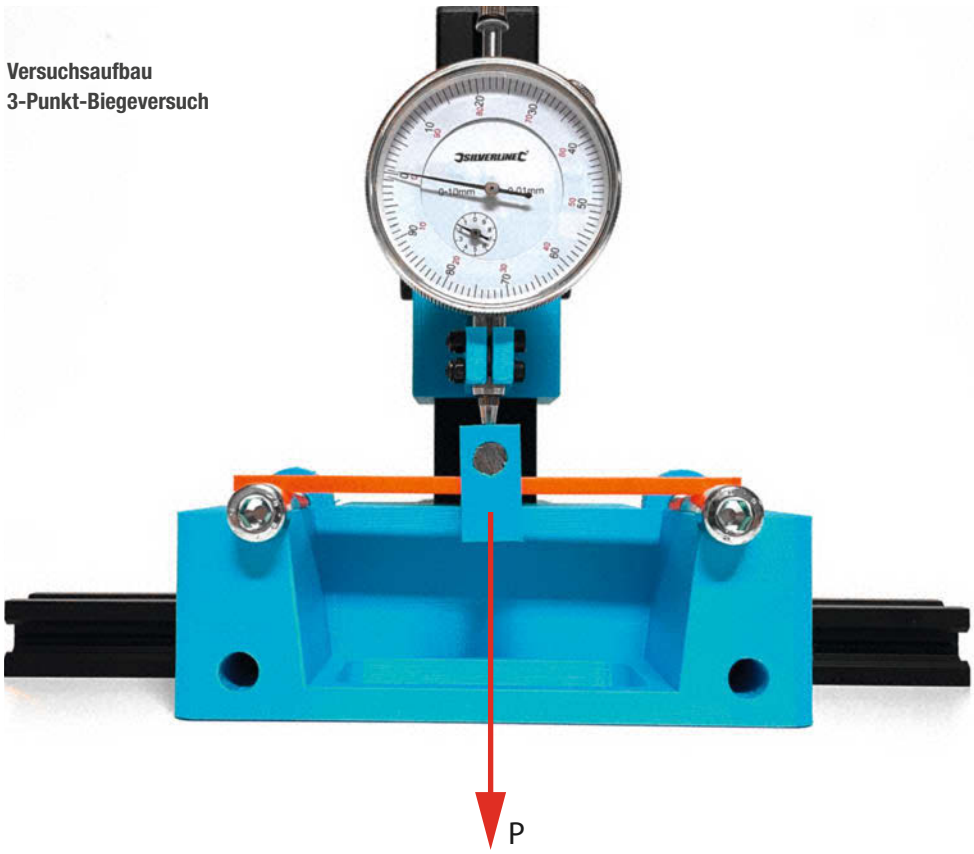
Temperaturtest

Die beim FFF(FDM)-3D-Druck verwendeten Materialien sind Thermoplaste, die – wie der Name schon sagt – die charakteristische Eigenschaft besitzen, sich bei bestimmten Temperaturen zunächst thermoelastisch und plastisch verformen zu lassen, bis sie schließlich in den fließfähigen Zustand übergehen. Auf diese Weise verarbeitet auch der Drucker das Filament, indem er es erhitzt und durch eine Düse presst.

Für Anwendungen in wärmeren Umgebungen, z. B. für Zubehörteile im Auto oder Gehäuse für elektronische Bauteile, ist es interessant, die Heat Deflection Temperature (kurz HDT, deutsch Wärmeformbeständigkeitstemperatur) zu kennen. Das ist die Temperatur, bei der sich das Material unter Belastung plastisch verformt, was in diesem Fall unerwünscht ist.

Mit meinem Temperaturtest lässt sich diese HDT vereinfacht und näherungsweise bestimmen. Dazu werden längliche Proben an zwei Punkten gehalten und in der Mitte mit einem Gewicht (9 Gramm in Form einer Flügelmutter M6 auf einer Schraube) beschwert, um eine konstante Kraft auf sie auszuüben.

Nun legt man diese Anordnung in einen Ofen und misst die Temperatur während des Aufheizens (hier 130 °C) regelmäßig. So kann man (am besten per Videoaufzeichnung) beobachten, bei welcher Temperatur die Probe weich wird und dann dem Gewicht der Mutter nicht mehr standhalten kann. Die Messwerte sind jedoch nicht 1:1 mit den gängigen Normen vergleichbar, da die Norm für die HDT eine konstante Aufheizrate von zwei Grad pro Minute und eine genaue Messung der Durchbiegung vorsieht, was mit einem Haushaltsbackofen und meinem Aufbau nicht erreichbar ist. Die Proben werden mit drei Perimetern



Versuchsaufbau 3-Punkt-Biegeversuch

© Copyright by Maker Media GmbH.

und 25% Gyroid Infill gedruckt, um realistische Bauteile abzubilden.

Kerbschlagbiegeversuch

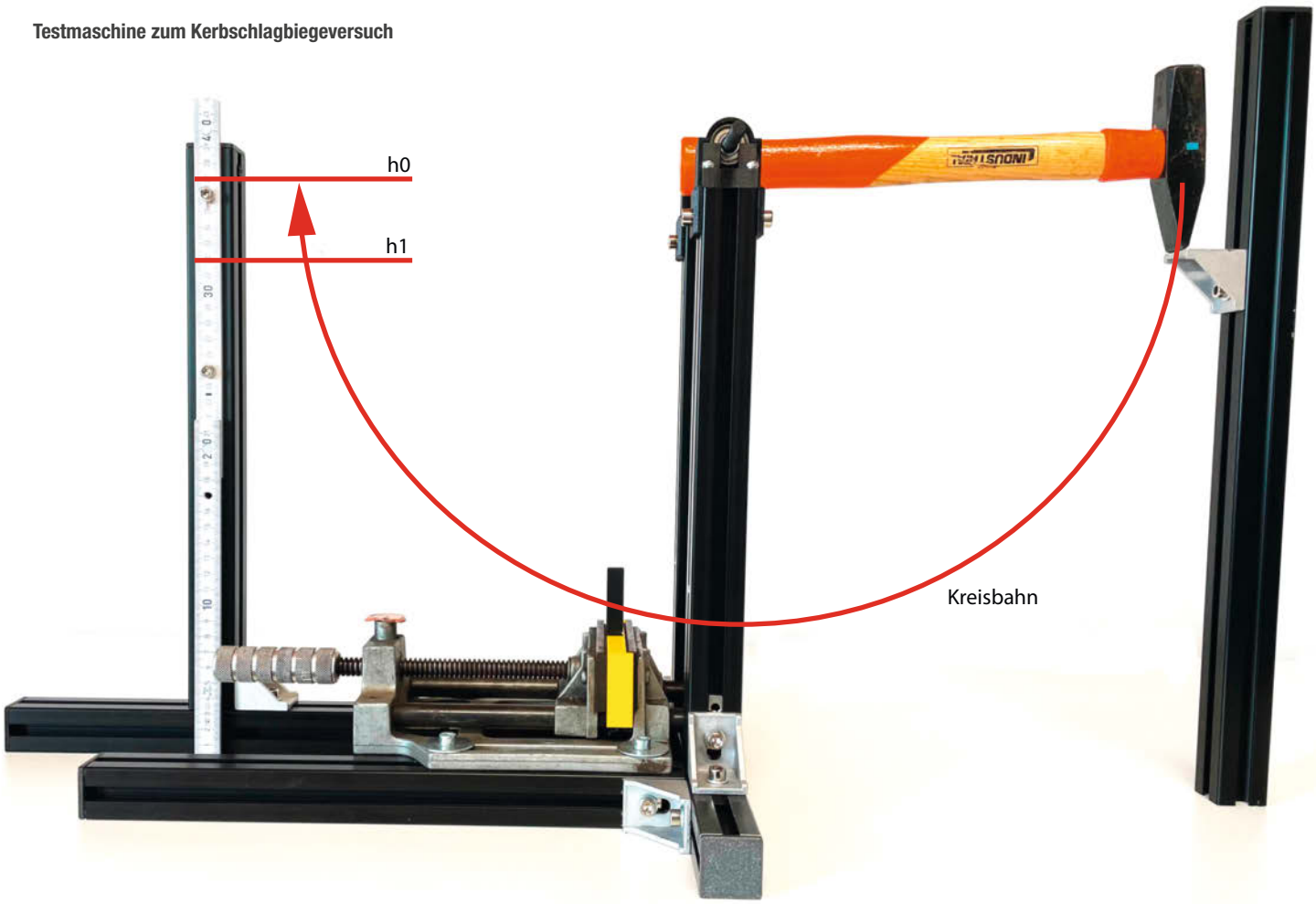
Mit Hilfe des Kerbschlagbiegeversuchs kann die Widerstandsfähigkeit des Werkstoffs gegen eine plötzliche, stoßartige Belastung untersucht werden. Konkret kann damit die Kerbschlagzähigkeit berechnet und mit Literaturdaten verglichen werden.

Eine hohe Kerbschlagzähigkeit kann z. B. bei der Konstruktion eines Quadcopter-Rahmens, von Schutzabdeckungen oder gedruckten Werkzeugen von Bedeutung sein.

Bei der Prüfung wird der obere Teil einer gekerbten Probe mit einem schwingenden Pendelhammer abgeschlagen. Gemessen wird die Nullhöhe h_0 (wie hoch schwingt der Hammer, wenn keine Probe im Weg ist) und die Höhe h_1 nach dem Abschlagen der Probe. Für die Höhenmessung verwende ich ein fest montiertes Maßband und verfolge den markierten Schwerpunkt des Hammers in einer Videoaufzeichnung.

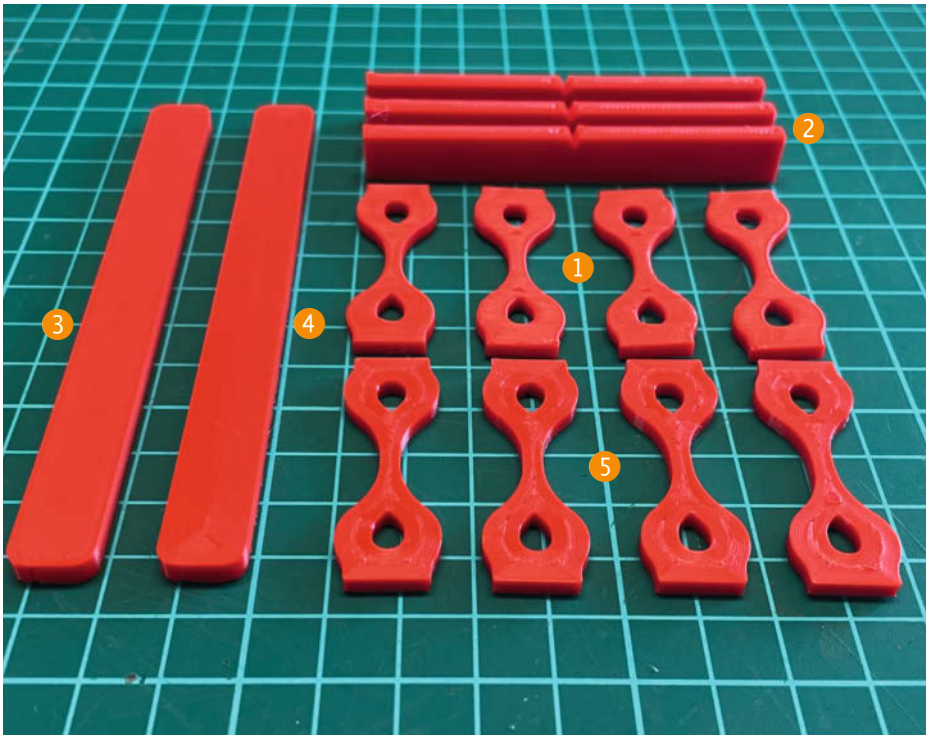
Versuchsaufbau für den Temperaturtest

Testmaschine zum Kerbschlagbiegeversuch



© Copyright by Maker Media GmbH.

Die getesteten Filamente



Ein Satz Proben, hier aus Prusament PLA Lipstick Red

Da die Masse des Hammers und die Erdbeschleunigung (g) bekannt und konstant sind und die Versuche bei möglichst konstanten Probentemperaturen durchgeführt werden, lässt sich die potenzielle Energie des Hammers leicht berechnen. Die Probe absorbiert einen Teil dieser Energie, die mit der Formel

$$E = m_{Hammer} \cdot g \cdot (h_0 - h_1)$$

berechnet werden kann.

Dividiert man das Ergebnis durch die Bruchfläche der Probe, so erhält man die Kerbschlagzähigkeit in kJ/m^2.

Vergleichbarkeit mit den Profis

Obwohl ich mich bei der Konstruktion der Prüfgeräte an Normen und professionellen Geräten orientiert habe, können Ungenauigkeiten und Messfehler auftreten. Die Tests dienen in erster Linie dem eigenen Vergleich verschiedener Kunststoffe. Beim Vergleich der Daten zu denen aus Datenblättern sollte man vorsichtig sein, zumal hier oft herstellerinterne Standards auftauchen.

© Copyright by Maker Media GmbH.

Dafür lassen sich die Messgeräte aber einfach und kostengünstig zu Hause oder im Makerspace nachbauen! Wie bereits erwähnt, sind die Baubeschreibungen einiger meiner Testgeräte bereits auf Printables zum kostenlosen Download verfügbar.

Der Vergleich

Wer sich schon einmal mit 3D-FFF-Druckern beschäftigt hat, kennt die Materialien PLA, PETG und ABS sicher. Dies sind die beliebtesten Kunststoffe in diesem Bereich. Sie werden in den verschiedensten Farben und Ausführungen als Filament auf Spulen verkauft.

Für den Test haben wir Filamente von etablierten Herstellern ausgewählt. Dies bietet eine gewisse Sicherheit, dass die Materialien auch nach Erscheinen dieses Artikels in gleicher Qualität und ausreichender Menge verfügbar sind.

Mit dabei ist „Das Filament", ein deutscher Hersteller mit preiswerten Filamenten und einer großen Farbauswahl. Für den Test verwende ich PLA Sonnengelb und PETG Opalgrün.

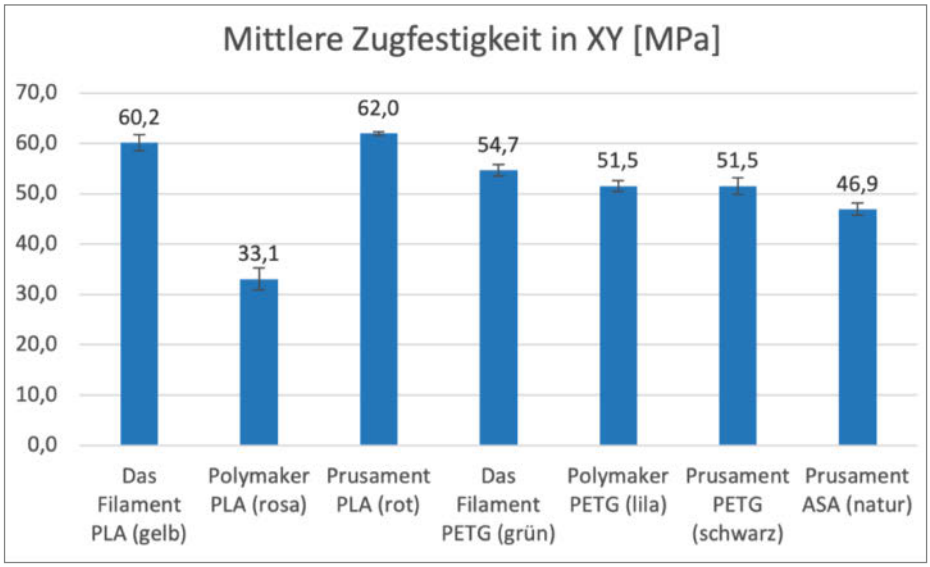
Der Hersteller Polymaker aus den USA hat sich in den letzten Jahren unter anderem auf preiswerte und zudem umweltfreundliche Filamente spezialisiert. Produziert werden die Filamente aber noch in China. In diesem Vergleich teste ich neben dem normalen PolyLite PETG Violett auch das PolyTerra PLA Sakura Pink, welches laut Hersteller mit organischen Materialien kombiniert wurde, um den Kunststoffanteil zu reduzieren und somit umweltfreundlicher zu sein.

Prusament von Prusa gilt als hochwertiges, aber auch relativ teures Filament. Für den Vergleich habe ich neben PLA Lipstick Red und PETG Galaxy Black auch ASA Natural ausgewählt. ASA ist in den letzten Jahren als Alternative zu ABS bekannt geworden und ersetzt dieses mittlerweile in sehr vielen Bereichen. Gründe dafür sind die einfache Verarbeitung im Vergleich zu ABS, die geringere Geruchsbildung beim Drucken und die gute UV-Beständigkeit.

Wer einen ausführlichen Test von ABS vermisst, findet auf meinem YouTube Kanal @JanTec3D einen ausführlichen Vergleich von ABS, mattem ABS und glasfaserverstärktem ABS.

Druck der Proben

Zu Vergleichszwecken habe ich für die oben vorgestellten Testgeräte mehrere entsprechende Muster aus allen Materialien gedruckt. Um vergleichbare Ergebnisse zu erhalten, habe ich immer den gleichen 3D-Drucker (Caribou MK3S mit 0,4 mm Revo Messingdüse) verwendet und die Drucktemperaturen und Lüftergeschwindigkeiten im Prusa Slicer jeweils nach Herstellerangaben eingestellt.



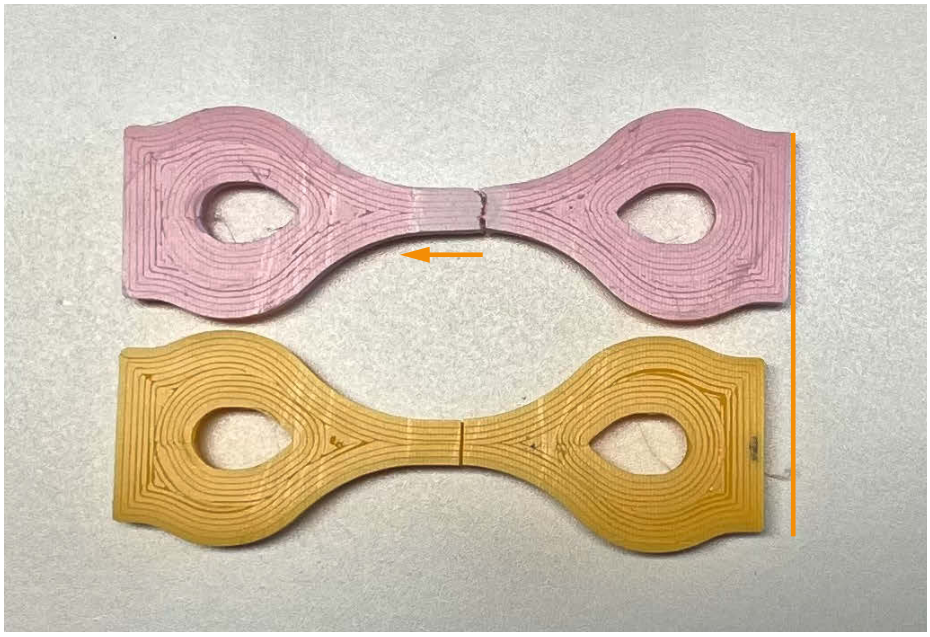
Balkendiagramm Zugversuch in XY-Richtung

Die Proben werden mit unterschiedlichen Slicer-Einstellungen gedruckt:

- Die Proben für den Y-Zugversuch 1 und Kerbschlagbiegeversuch 2 werden komplett gefüllt gedruckt (100% Infill), da hier die Bruchfläche später genau vermessen werden muss, um die Messungen zu normieren und vergleichbar zu machen. Wäre diese Fläche mit einem komplexen Infill-Muster gefüllt, wäre eine exakte Messung praktisch unmöglich.
- Für die Temperaturtestprobe 3 in Form eines einfachen Balkens habe ich jedoch 25% Gyroid-Füllung gewählt, um realistischere Bauteile abzubilden.
- Die Balken für den Biegeversuch 4 und die liegenden Proben für den XY-Zugversuch 5 werden mit der maximalen Anzahl an Konturen (Perimeter) gedruckt, um den Kraftfluss senkrecht bzw. entlang der Drucklinien zu führen und die Ergebnisse nicht durch zusätzliche Geometrien wie Infill oder schräg gestapelte Linien an Ober- und Unterseite zu verfälschen.

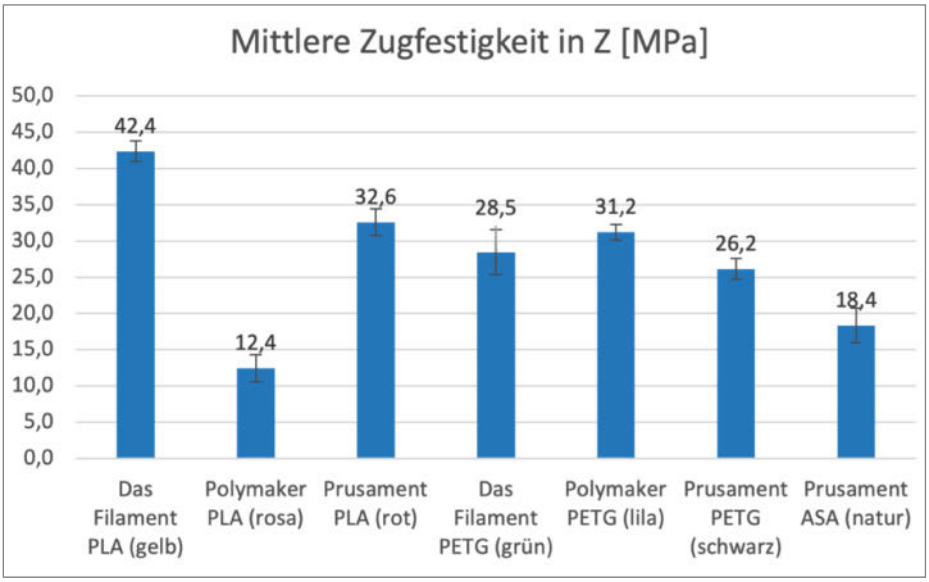
Ergebnisse

Schon bei der Betrachtung der gedruckten Proben fallen erste Unterschiede auf. Während die PLA-Filamente von Prusament und Das Filament leicht glänzen, ist das PLA von Polymaker matt. Außerdem fühlt sich die Ober-

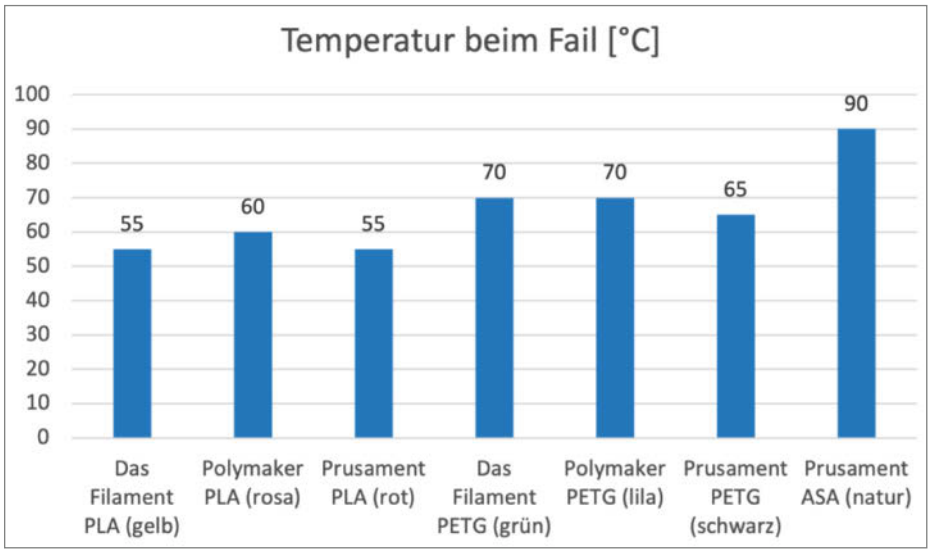


Die hohe Bruchdehnung bei Polymaker PLA

© Copyright by Maker Media GmbH.



Balkendiagramm Zugversuch in Z-Richtung



Balkendiagramm Temperaturtest

fläche bei der Fingernagelprobe eher weich und nicht hart an, wie es bei PLA üblich ist.

Im Folgenden werden die Ergebnisse der einzelnen Tests kurz dargestellt. Die Interpretation und die Antwort auf die Frage, welches Filament für welche Anwendung am besten geeignet ist, folgt im Anschluss.

Zugversuch

Man erkennt, dass zwei PLA-Proben zugfester sind als PETG und ASA. Das PolyTerra PLA von Polymaker fällt jedoch deutlich ab und erreicht nur etwa die Hälfte der Zugfestigkeit.

Außerdem erkennt man an den zerstörten Proben (siehe Bild), dass die Bruchdehnung, also die Dehnung bis zum Bruch, beim Polymaker PLA besonders hoch ist. Die Probe dehnt sich gut 1 bis 2 mm, bevor sie reißt. Bei den beiden anderen PLA-Filamenten kann keine wesentliche Bruchdehnung festgestellt werden. Das Bild zeigt eine gerissene Polymaker-PLA-Probe (oben) und eine gerissene Probe, die mit „Das Filament" PLA gedruckt wurde (unten).

Bei der Zugfestigkeit in Z-Richtung als Maß für die Schichthaftung gewinnt das PLA von „Das Filament" recht deutlich, Polymaker PLA schneidet wiederum sehr schlecht ab. Bei den PETG-Filamenten sind anhand der Fehlerbalken keine signifikanten Unterschiede zu erkennen.

ASA hat jedoch eine leicht schlechtere Schichthaftung als PETG. An dieser Stelle ist es wichtig zu erwähnen, dass die Schichthaftung stark von der verwendeten Drucktemperatur und der Bauteilkühlung abhängt: Je heißer und flüssiger die Schichten aufeinander gedruckt werden, desto besser verbinden sie sich. Gleichzeitig nimmt aber auch das Stringing (Fadenziehen) zu, die Oberflächenqualität kann leiden und schwach (selbst-)gestützte Bauteile können bei höheren Temperaturen kollabieren.

Temperaturtest

In den Ergebnissen des Temperaturtests ist gut zu erkennen, dass ASA mit Abstand am temperaturbeständigsten ist. Erstaunlich ist auch, dass der Unterschied zwischen Polymaker PLA und Prusament PETG mit nur 5 °C nicht besonders groß ist. Datenblätter und allgemeine Empfehlungen in Foren sehen hier oft einen größeren Unterschied. Hier wäre eine Quantifizierung durch Vergleich mit ungefärbten Filamenten des gleichen Herstellers interessant, vielleicht verändern die farbgebenden Komponenten die Wärmeleitung im Bauteil oder ähnliches. Dies muss aber in der Zukunft geklärt werden.

3-Punkt-Biegeversuch

Je höher der Biegemodul, desto steifer (d. h. weniger elastisch) ist der Kunststoff. PLA ist bei weitem der steifste der gängigen Filamenttypen. Dies ist besonders bei Befestigungen für Extruder oder Spindeln auf Linearschlitten erwünscht, damit das Werkzeug möglichst präzise bewegt werden kann und nicht vibriert. Bei Smartphone-Hüllen etwa kann eine eher geringe Steifigkeit von Vorteil sein, um eine gewisse Biegung für das Einlegen des Gerätes zuzulassen und Stöße besser abzufangen.

Kerbschlagbiegeversuch

Die Kerbschlagzähigkeit beschreibt, wie viel Energie die Probe bei einer stoßartigen Beanspruchung aufnehmen kann. Aus Gründen der Vergleichbarkeit wird sie in Energie (SI-Einheit Joule) pro Fläche (m^2) angegeben. Daraus ergibt sich die Einheit kJ/m^2.

PolyTerra PLA fällt hier positiv auf, da es mit 11,2 kJ/m^2 eine vergleichsweise hohe Schlagzähigkeit aufweist. PLA gilt normalerweise als spröde, was auch zu einer niedrigen Schlagzähigkeit führt. PolyTerra PLA wirkte jedoch von Anfang an weich und zäh, was erklärt, warum es mehr Energie aufnehmen kann, bevor es bricht.

Dies kann auch ohne Prüfgerät untersucht werden, indem man einfach ein Stück Filament von der Spule abreißt oder es zumindest versucht: Einige Filamente (wie PETG) brechen kaum.

Auswertung

Dieser Test zeigt deutlich: PLA ist nicht gleich PLA. Bei PETG ist die Streuung deutlich geringer. Generell kann man sagen, dass die me-

© Copyright by Maker Media GmbH.

chanischen Eigenschaften der Materialien stark vom Hersteller beeinflusst werden. Er entscheidet, welche Rohstoffe, Farbpigmente und sonstigen Zusätze er verwendet und wie er das Filament herstellt. Damit bringt er ein Filament mit ganz bestimmten Eigenschaften auf den Markt, das sich sehr von dem Grundmaterial unterscheiden kann.

Dem Polymaker PolyTerra PLA werden beispielsweise organische Stoffe beigemischt, um es umweltfreundlicher zu machen. Das erklärt die matte Optik und die zumeist schlechteren mechanischen Eigenschaften. So bleibt für ein bestimmtes mechanisches Bauteil immer die Frage offen, ob es den Belastungen standhält, wenn es mit einem neuen Filament gedruckt wird.

Auf der anderen Seite steht der Anwender, der durch Optimierung der Druckparameter wie Temperaturen, Druckgeschwindigkeit, Kühlung und Extrusionsfaktor sowohl optisch als auch technisch bessere Ergebnisse erzielen kann. Fehler in diesem Bereich (z. B. zu starke Bauteilkühlung bei PETG oder ABS) können jedoch auch beim theoretisch besten Material zum Bauteilversagen führen.

Doch was können wir als Maker konkret aus diesen Ergebnissen mitnehmen?

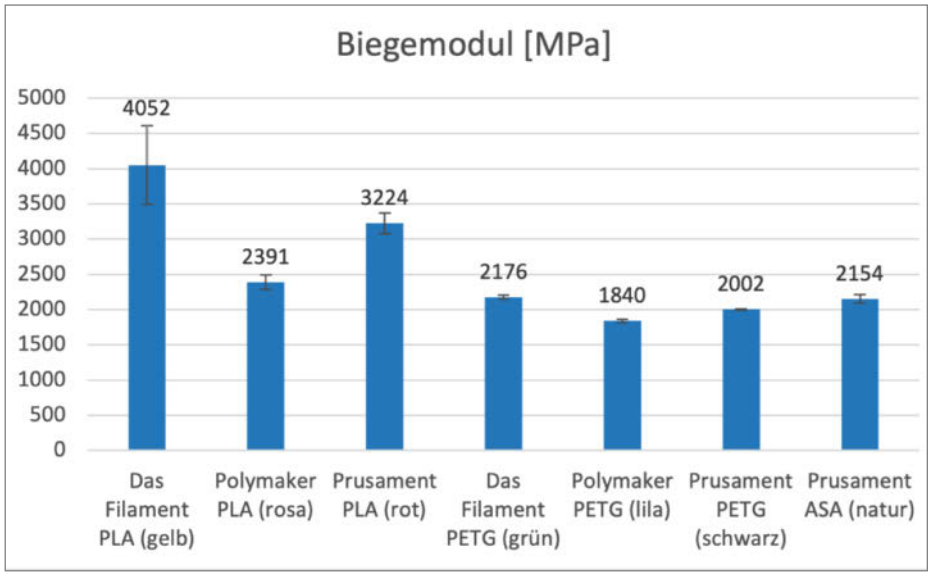
Preis-Leistung

Gerade wenn größere Modelle gedruckt werden sollen, kann der Preis des Filaments eine wichtige Rolle spielen und sich stark auf die Endkosten des Projekts auswirken. Vor allem bei Figuren oder Modellen ohne mechanische Anforderungen kann es sich lohnen, auf preisgünstiges PLA wie PolyTerra zurückzugreifen, das zudem in vielen Farben erhältlich und gut druckbar ist.

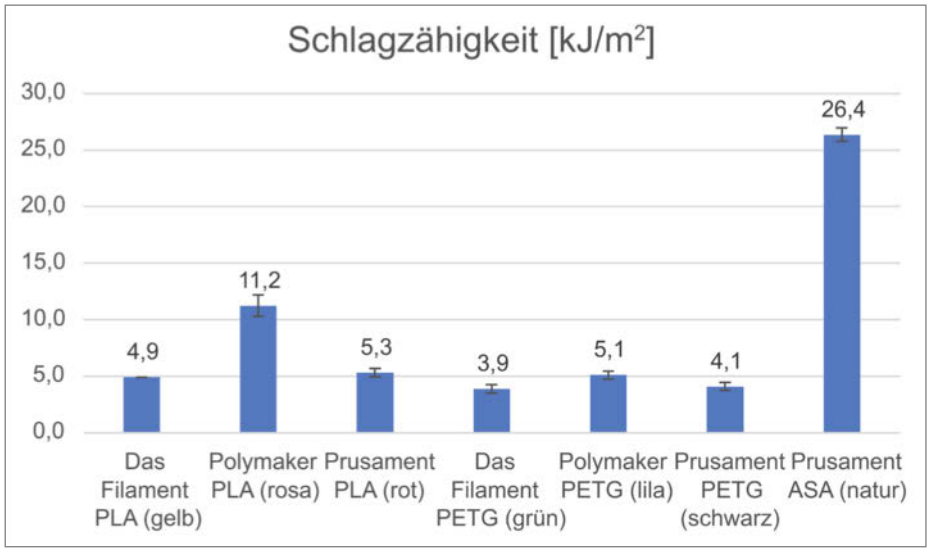
Nachbearbeitung

Die HDT des Filaments sagt auch einiges über die Nachbearbeitbarkeit der gedruckten Teile aus. So lassen sich Bauteile aus PLA beispielsweise mit warmem Wasser oder einem Föhn verformen. Dies kann in einigen Fällen eine erneute Konstruktion und Druck ersparen. Bei ASA (und ABS) wäre dies ohne Verbrennungsgefahr nur umständlich möglich. Dafür lassen sich Teile aus diesem Material wesentlich besser schleifen. PLA und auch PETG dürfen beim Schleifen und Bohren nicht zu heiß werden, sonst verschmiert der Abtrag und man bekommt keine guten Oberflächen und das Schleifmittel bzw. der Bohrer setzen sich mit geschmolzenem Material zu. Nassschleifen kann hier Abhilfe schaffen. Das Bohren, Gewindeschneiden und Fräsen von gedruckten Teilen ist bei einem Material mit hoher Kerbschlagzähigkeit einfacher, da es eher eine Art Späne produziert und nicht so leicht schmilzt und dann schmiert.

Die Proben nach dem Temperaturtest



Die Biegemodule aus 3-Punkt-Biegeversuch



Balkendiagramm Kerbschlagzähigkeit

© Copyright by Maker Media GmbH.

Ausrichtung auf dem Druckbett

Wenn bereits bekannt ist, in welche Richtung(en) das gedruckte Bauteil später am stärksten belastet wird, kann der vergleichsweise schlechten Schichthaftung z. B. bei ASA u. a. durch Drehen des Bauteils auf dem Druckbett entgegengewirkt werden. Ist das Bauteil dann ohne Stützmaterial nicht mehr druckbar, bietet sich die Verwendung von organischen Stützen im PrusaSlicer an (Treesupports in Cura).

Druckeinstellungen

Wie bereits erwähnt, kann die Schichthaftung durch höhere Drucktemperaturen und geringere Bauteilkühlung verbessert werden. Auch eine zu kühle Umgebung verschlechtert die Schichthaftung. Bei ABS und ASA ist ein geschlossener, evtl. sogar beheizter Bauraum sehr förderlich.

Anwendungen für PLA

Vor allem für Figuren, Modelle und Prototypen, die nur zum Ausstellen und Anschauen gedacht sind, eignet sich PLA sehr gut, da es am billigsten und sehr leicht zu drucken ist und eine Warping-Tendenz (Verziehen durch unterschiedlich schnelles Abkühlen der Schichten) praktisch nicht vorhanden ist.

Trotzdem sollte PLA nicht unterschätzt werden. Gerade die hohe Zugfestigkeit und Steifigkeit machen es für technische Anwendungen bei Raumtemperatur sehr interessant, da es zudem sehr einfach zu lagern und zu drucken ist. Beispiele sind Wandhalterungen, Kleiderhaken, Lagerschalen für Antriebe, Halterungen für mechanische Komponenten in der Robotik und für 3D-Drucker.

Anwendungen für PETG

PETG kann für fast alles verwendet werden, wenn die Festigkeits- und Steifigkeitsparameter den Anforderungen entsprechen. Leichte Nachteile sind, dass das Material je nach Drucker etwas schwieriger zu verarbeiten ist, vor allem Stringing ist manchmal schwer zu vermeiden. PETG kann bei offener Lagerung mit der Zeit Feuchtigkeit aufnehmen, was die Druckbarkeit verschlechtert, und es ist etwas teurer als PLA.

Anwendungen für ASA

ASA hat zwei klare Vorteile: die hohe Temperaturbeständigkeit und die Schlagzähigkeit. Hinzu kommt die gute UV-Beständigkeit, die für Bauteile im Außenbereich wichtig ist. Konkret denke ich hier an Zubehör oder Ersatzteile für Fahrräder oder den Garten.

Ein Nachteil ist allerdings, dass ASA beim Drucken zum Warping neigt, was bei größeren Bauteilen auf offenen Druckern zu Problemen führen kann. Ein geschlossener Bauraum, eventuell sogar mit aktiver Temperierung, kann auch den eher unangenehmen Geruch beim Drucken verhindern. Ein weiterer Nachteil ist der relativ hohe Preis.

Wenn die Temperaturbeständigkeit zusammen mit den mechanischen Eigenschaften benötigt wird, wie z. B. bei Gehäusen für elektronische Bauteile, lohnt sich der Einsatz von ASA. Als alltägliches Material würde ich es jedoch aufgrund der oben genannten Nachteile nicht verwenden.

Datenblätter richtig lesen

Grundsätzlich ist es auch möglich, Filamente zu vergleichen, ohne eigene Prüfgeräte zu bauen. Die meisten Hersteller stellen Datenblätter ihrer Filamente zum Download zur Verfügung, „Das Filament" hat uns die Datenblätter auf Anfrage zugeschickt. Dabei ist Folgendes zu beachten:

Verschiedene Hersteller verwenden meist unterschiedliche Testmethoden oder stellen die jeweiligen Proben unterschiedlich her. So haben z. B. spritzgegossene Muster andere Eigenschaften als tatsächlich gedruckte und schon ist die Vergleichbarkeit dahin.

Zudem gibt es oft unterschiedliche Testarten und auch je nach Materialklasse unterscheiden sich die Verfahren und sollten daher nur mit Bedacht direkt miteinander verglichen werden.

Außerdem ist zu beachten, dass auch die Farbe (und andere Zusatzstoffe) die mechanischen Eigenschaften beeinflussen können. Allerdings habe ich noch keinen Hersteller gefunden, der für jede Farbe ein eigenes Da-

Testergebnisse

Material	Das Filament PLA	Polymaker PolyTerra PLA	Prusament PLA	Das Filament PETG	Polymaker PETG	Prusament PETG	Prusament ASA
Farbe im Test	gelb	rosa	rot	grün	violet	schwarz	natur
Herkunftsland	DE	CN	CZ	DE	CN	CZ	CZ
Preis [€/kg]	20,56	19,99	29,99	27,44	29,99	29,99	35,28
Material der Spule	Kunststoff	Pappe	Kunststoff/ Pappe	Kunststoff	Pappe	Kunststoff/ Pappe	Kunststoff/ Pappe
Refill Option?	Ja	Nein	Bedingt (einige Farben)	Ja	Nein	Nein	Nein
Zugfestigkeit XY [MPa]	59	33	61	55	51	47	46
Zugfestigkeit Z [MPa]	42	12	33	29	31	26	18
Erweichungs-temperatur [°C]	55	60	55	70	70	65	90
Biegemodul [MPa]	4052	2391	3224	2176	1840	2002	2154
Kerbschlagzähig-keit [kJ/m2]	4,9	11,2	5,3	3,9	5,1	4,1	26,4
Eignung							
Druckbarkeit	⊕⊕	⊕⊕	⊕⊕	⊕	⊕	⊕	⊖
dekorative Objekte	⊕⊕	⊕⊕	⊕⊕	⊕	⊕	⊕	○
technische Objekte	⊕⊕	⊖	⊕⊕	⊕⊕	⊕⊕	⊕⊕	⊕⊕
Hitzefestigkeit	⊖	⊖	⊖	○	○	○	○
UV-Beständigkeit	○	○	○	⊕	⊕	⊕	⊕⊕
Nachbearbeitung	⊕	⊕	⊕	⊕⊕	⊕⊕	⊕⊕	⊕⊕

⊕⊕ sehr gut ⊕ gut ○ zufriedenstellend ⊖ schlecht ⊖⊖ sehr schlecht

© Copyright by Maker Media GmbH.

Kleine Werkstoffprüfungskunde

Megapascal (MPa): Die SI-Einheit Pascal ist die in der Werkstoffprüfung sehr wichtige Einheit der mechanischen Spannung, welche ein Maß für die innere Beanspruchung eines Körpers ist. Diese Beanspruchung ist auf äußere Belastungen, wie Zug oder Druck zurückzuführen. Die Einheit Pascal ist definiert als 1 Pa = 1 N/m², also die auf eine Fläche ausgeübte Kraft.

Aufgrund der kleinen Proben (bzw. der kleinen Bruchflächen der Proben, im mm²-Bereich), die für diese Art von Versuchen verwendet werden, wird hier die Einheit 1 MPa = 1 Megapascal = 1 N/mm² verwendet.

Zugfestigkeit: Die Zugfestigkeit beschreibt die Widerstandsfähigkeit gegenüber zugartiger (äußerer) Belastungen. Diese Eigenschaft eines Werkstoffes beschreibt die Beanspruchbarkeit durch mechanische Belastungen, bevor es zu einem Versagen kommt. Die Werte werden angegeben als mechanische Spannung, also Kraft pro Querschnittsfläche, ebenfalls üblicherweise in MPa (siehe dort).

Biegemodul: Der Biegemodul (auch Elastizitätsmodul genannt) gibt das Verhältnis der aufgebrachten Spannung zur Verbiegung an und gilt damit als Maß für die Steifigkeit des Materials. Auch hier ergeben sich Werte in der Einheit Megapascal (MPa).

Kerbschlagzähigkeit: Wird berechnet als Verhältnis aus Schlagarbeit und Querschnitt der verwendeten Probe. Daraus ergibt sich die Einheit J/m². Die Schlagarbeit lässt sich in unserem Fall (Hammer-Probe) durch den Unterschied an potenzieller Energie des Hammers vor und nach dem Schlag berechnen, indem man des Ausgangs- und Endhöhe des Hammers misst.

Fehlerbalken: Die Fehlerbalken in den Balkendiagrammen geben grafisch die Streuung der dargestellten Messdaten an. Sie zeigen an, in welchem Bereich die gemessenen Werte der Einzeltests liegen und ob es signifikante Unterschiede zwischen den verschiedenen Datenpunkten gibt (in diesem Fall zwischen den getesteten Filamenten). Liegen die gemittelten Ergebnisse innerhalb des Fehlerbalkens eines anderen Tests ist messtechnisch kein Unterschied anzugeben.

tenblatt angibt; in der Regel gibt es ein allgemeines Datenblatt, in dem die getestete Farbe nicht einmal angegeben ist. Wer sich für dieses Thema interessiert, dem empfehle ich das deutsche Video von CNC Kitchen zu diesem Thema (Link in der Kurzinfo).

Am besten kann man allein anhand der Datenblätter die verschiedenen Typen Filament eines Herstellers vergleichen, da davon auszugehen ist, dass er sein Material auf stets dieselbe Weise testet.

Ausblick

Neben weiteren Herstellern und Materialien können auch unterschiedliche Druckeinstellungen getestet und somit verglichen werden. Konkret können Drucktemperaturen, Schichthöhen, Füllmuster und andere Einstellungen des Slicers optimiert werden. Allein aus diesem Grund kann es sinnvoll sein, ein paar Testmaschinen zu bauen, wenn man hochoptimierte Mechaniken mittels 3D-Druck herstellen möchte. So helfen diese Tests, Druckeinstellungen, Druckgeschwindigkeit und Materialverbrauch mit reproduzierbaren Daten zu optimieren.

Auch der Vergleich von Klebstoffen, z. B. für PLA oder PETG, kann interessant sein, da man durch Zug- und Biegeversuche vergleichbare Werte der Klebkraft ermitteln und vergleichen kann.

Daher kann ich jedem, der sich für dieses Thema interessiert, nur raten, ebenfalls mit dem Testen zu beginnen. Traut euch! Die Bauanleitungen und Stücklisten einiger meiner Testgeräte sind auf Printables abgelegt, weitere werden folgen. Eine Kofferwaage und zwei Haken, zwischen denen eine Probe zerrissen wird, reichen für den Anfang aber auch.

Je nach Anreise bringe ich alle oder eine Auswahl meiner Testgeräte auch mit auf Maker Faires. Wer also Interesse hat, kann dort gerne vorbeikommen und ein paar eigene Proben mit den Testgeräten zerstören. Die passenden Druckdateien finden Sie in den Downloads zum Artikel. Ich freue mich! —*caw*

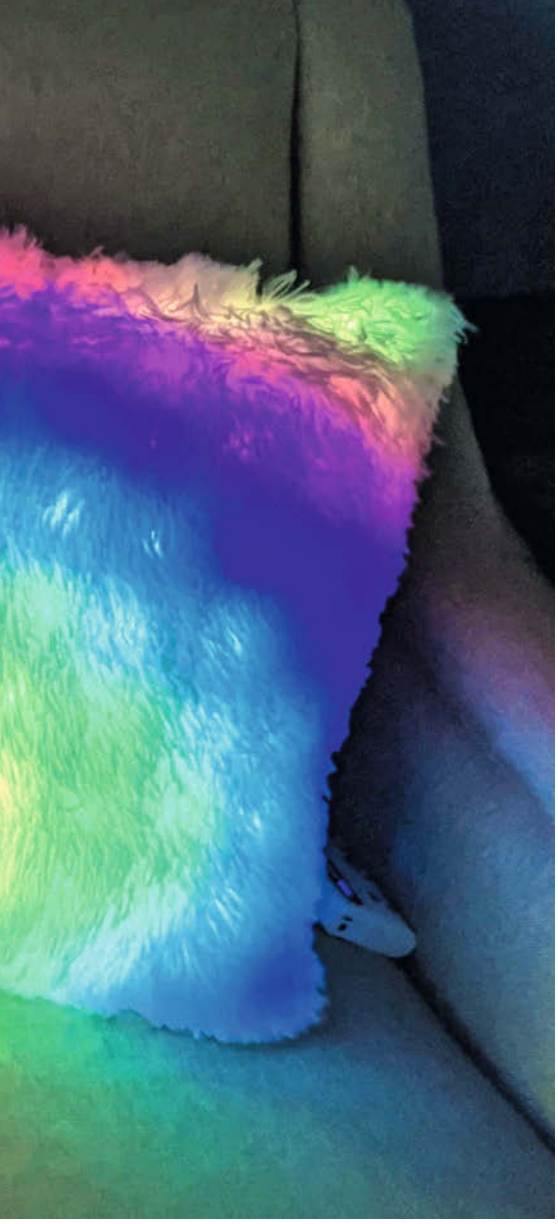
© Copyright by Maker Media GmbH.

LED-Kissen
mit dem Pixelblaze Controller

Mithilfe einer LED-Kette und dem Pixelblaze Controller lassen sich über das Internet steuerbare LED-Kissen basteln, die in jeder Wohnung ein Blickfang sind.

von Debra Ansell

© Copyright by Maker Media GmbH.



Kurzinfo

» Muster und Effekte frei programmierbar und übers Web steuerbar
» Umsetzung mit dem Pixelblaze Controller

Checkliste

Zeitaufwand:
10 bis 15 Stunden

Kosten:
160 Euro

Material

» **Pixelblaze V3 Standard - WiFi LED Controller**
» **2 Kissen** 50 cm × 50 cm
» **2 Kissenbezüge aus durchscheinendem Stoff** weißer Plüschstoff oder Samt
» **2 Stepp-Einlagen** ca. 1,3 cm dick in Quadraten von 50 cm × 50 cm
» **2 RGB-LED-Ketten für den Außenbereich** 10 Meter mit adressierbaren LEDs mit 10 cm Abstand
» **2 dreipolige JST-SM-Stecker**
» **Dreipoliges JST-SM-Verlängerungskabel**
» **Netzstoff** 2 Quadrate von 53 cm × 53 cm und 4 mal 53 cm × 30 cm
» **200 Stück doppeladrigen Bindedraht mit Ummantelung** Weiß 10 cm
» **Netzteil** 5V/3A mit Universaladapter

Werkzeug

» Drahtschneider
» Abisolierzange
» Stoffschere
» Nähmaschine oder Nadel und Faden
» Lineal
» Dunkler Filzstift
» Nähklammern
» Zange
» Kleiner Schraubenzieher
» Lötkolben

Mehr zum Thema

» Carsten Wartmann, Disco is back: Lichtshow mit WLED auf ESP-Basis basteln und per App steuern, Make 2/23, S. 8
» Daniel Springwald, Elektronische Lavalampe basteln: Mit ESP32 stimmungsvolle Lichteffekte erzeugen, Make 3/23, S. 36

Alles zum Artikel im Web unter *make-magazin.de/x4bd*

Es ist ein zutiefst menschlicher Zwang, unsere Umgebung so zu verändern, dass sie zu uns passt. Die Viertelmilliarden-Dollar-Industrie für Heimdekoration in den USA ist ein spielerisches (und profitables) Beispiel dafür: Wir schmücken unsere Wohnräume, um uns auszudrücken und unsere Individualität zu zeigen. Farbenfrohe Lichtspiele schmücken belebte Veranstaltungsorte für Konzerte und Sportarenen, aber sie können auch als bezaubernde Dekoration für das eigene Zuhause dienen.

Beleuchtete Innenarchitektur

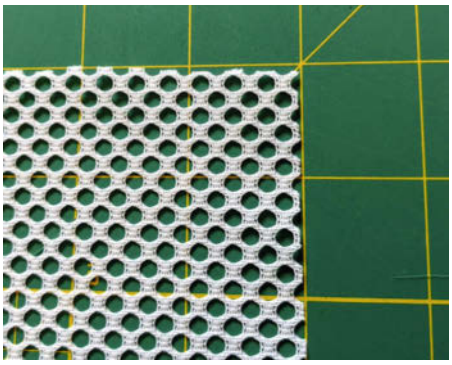
Dieses LED-Matrix-Kissenprojekt ist nicht nur hübsch, sondern auch praktisch. Die Elektronik ist an einem abnehmbaren DIY-Kissenbezug befestigt, sodass der äußere dekorative Bezug und die gefüllte Kisseneinlage zur Reinigung entfernt werden können. Sie können die Farben und Muster der Anzeigen ändern, die von der programmierbaren Pixelblaze-Platine gesteuert werden, um sowohl die Optik zu verschönern als auch für Unterhaltung zu sorgen. Es hat etwas unerwartet Reizvolles, Elektronik in weiche, kuschelige Objekte einzubetten, und diese glitzernden Kissen sorgen bei mir zu Hause für viel Freude.

Stoff zuschneiden und ein Raster markieren

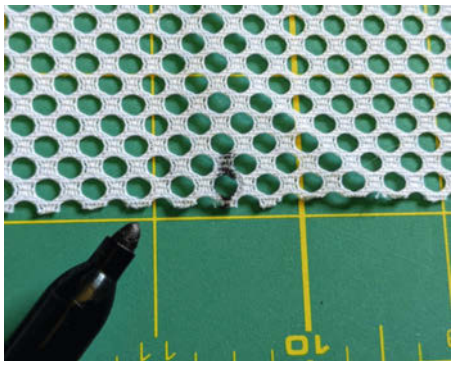
Für zwei Kissen mit den Masen 50 cm × 50 cm schneiden Sie sechs Rechtecke aus dem Netzstoff: zwei mit 53 cm × 53 cm Maßen und vier mit 30 cm × 53 cm. Schneiden Sie die Kanten sauber entlang der Löcher des Netzes zu, um später das Ausrichten der Matrix zu erleichtern.

Falten Sie ein 53-cm-Quadrat in der Hälfte und machen Sie mit einem Filzstift einen Punkt an einem Ende der Mittellinie.

Falten Sie das Quadrat auf und legen Sie es auf eine ebene Fläche mit einem Stück Schmierpapier darunter. Zeichnen Sie nun



Eine Unterlage mit Hilfslinien vereinfacht das Projekt.

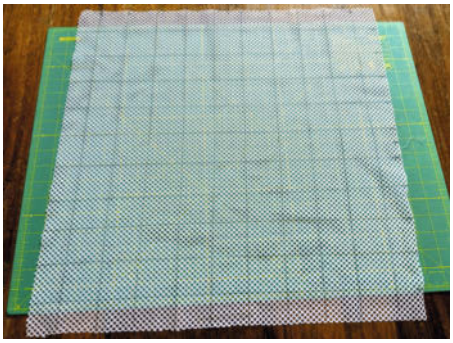


Markierungen müssen mit einem gut sichtbaren Stift gemacht werden.

© Copyright by Maker Media GmbH.

Nachdem die erste Seite markiert wurde, müssen neun gleich große Segmente ersichtlich sein.

mit dem Filzstift zwei vertikale Markierungen, die sich 2,5 cm auf jeder Seite der Mittellinie befinden. Fahren Sie damit fort, in Abständen von 5 cm auf beiden Seiten vertikale Markierungen zu zeichnen, bis Sie 10 symmetrisch angeordnete Markierungen haben. Beachten Sie hierbei, dass die Lochgröße des Stoffes auf den hier gezeigten Bildern anders sein kann als bei dem von Ihnen verwendeten Stoff. Setzen Sie die Markierungen nach Längenabständen, nicht nach Lochzahl.



Der Stoff muss am Ende in neun mal neun gleich große Felder eingeteilt sein.

Benutzen Sie als nächstes ein Lineal, das an den Maschenlöchern ausgerichtet ist, und ziehen Sie Linien, um die Markierungen auf die gegenüberliegende Seite des Maschenquadrats zu verlängern. Drehen Sie dann das Maschenstoffquadrat um 90 Grad und ziehen Sie auf die gleiche Weise identische Linien, die senkrecht zur ersten Reihe verlaufen. Jetzt haben Sie ein 9 × 9-Gitter aus 5-cm-Quadraten. Wiederholen Sie diesen Vorgang mit dem zweiten 53-cm-Quadrat.

Nähen Sie die Kissenbezüge

Legen Sie ein 53-cm-Maschenquadrat mit der markierten Seite nach oben auf eine ebene Fläche. Legen Sie zwei Rechtecke mit 30 cm × 53 cm so darüber, dass die langen Kanten mit den gegenüberliegenden Seiten des Quadrats übereinstimmen. Die Rechtecke werden sich in der Mitte überlappen. Dadurch entsteht eine Öffnung, durch die später das Kissen eingelegt werden kann. Befestigen Sie die Kanten aller drei Teile mit Heftklammern oder Büroklammern.

Nähen Sie mit Ihrer Nähmaschine oder mit Nadel und Faden eine Naht mit einer Nahtzugabe von 1,5 cm um den gesamten Umfang der überlappenden Stoffteile. Verstärken Sie die Nähte, indem Sie in der Nähe der Ecken und an den Stellen, an denen sich die rechteckigen Lagen überlappen, rückwärts nähen. Wenn Sie mit der Hand nähen, verwenden Sie kleine, feste Stiche, um den Stoff gut zu fixieren und verknoten Sie die Fadenenden fest.

Sobald der Netzkissenbezug fertig genäht ist, schneiden Sie die vier Ecken mit einer scharfen Schere diagonal und knapp außerhalb der Naht ab.

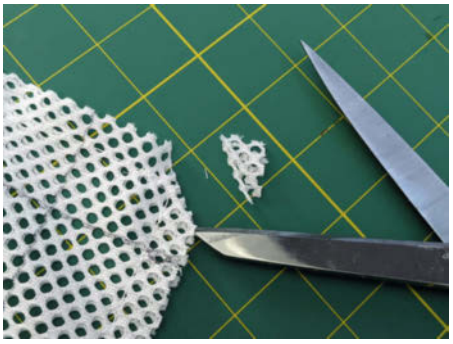
Drehen Sie den Bezug um und drücken Sie die Ecken mit Ihren Fingern von innen nach außen. Legen Sie ein Kissen in den Netzbezug ein, sodass der Bezug ohne Spiel ausgefüllt wird.

Erstellen Sie die LED-Matrix

Das auf den Bezügen eingezeichnete Raster hat einen Abstand von 5 cm, aber die benachbarten LEDs auf den Ketten haben einen Abstand von 10 cm. Wir werden diesen Unterschied ausgleichen, indem wir die LED-Ketten in ein Muster aus zwei senkrechten Zickzack-Gittern weben, die sich perfekt überlappen, um eine LED an jedem Gitterschnittpunkt zu platzieren.

Jede der mit Harz ummantelten LEDs hat eine unterschiedliche Vorder- und Rückseite. Die Rückseite ist etwas flacher und deutlich kleiner als die Vorderseite. Achten Sie darauf, die LEDs mit der Vorderseite nach oben zu platzieren. Zum Befestigen der Matrix an Ihrem Netz-Kissenbezug verwenden Sie doppeladrigen Bindedraht mit Ummantelung. Verwenden Sie keinen normalen Draht – dieser würde sich verdrehen und Ihren Kissenbezug durchstechen. Schneiden Sie den Bin-

Durch die Überlappung in der Mitte entsteht eine Öffnung, durch die später das Kissen eingelegt werden kann.



Die Ecken müssen leider weichen.

© Copyright by Maker Media GmbH.

dedraht mit einer Drahtschere oder einem Seitenschneider in gleich lange Stücke von etwa 2,5 cm.

Dann nehmen Sie das Eingangsende der LED-Kette und legen es entlang einer der Randgitterlinien, wobei Sie die LEDs an den Schnittpunkten ausrichten.

Befestigen Sie die LED-Kette in regelmäßigen Abständen am Netzbezug, indem Sie diese neben jeder LED mit einem abgeschnittenen Bindedraht umwickeln. Binden Sie nicht beide Seiten der LEDs fest, denn es muss genug Spielraum bleiben, um später einen anderen Teil der LED-Schnur darunter durchlaufen zu lassen. Schieben Sie ein Bindedrahtstück durch zwei nebeneinander liegende Löcher, die die Filzstift-Linie begrenzen, sodass der Bindedraht senkrecht zur Lichterkette steht.

Das Bindedrahtstück ist etwas breiter als die Löcher, daher müssen Sie das Netz eventuell straff ziehen und ein wenig wackeln, um es durchzuschieben.

Knicken Sie mit einer Zange den Bindedraht beidseitig fest um die LED-Kette. Biegen Sie den verbleibenden überhängenden Abschnitt mit der Zange zurück unter die LED-Schnur und drücken Sie ihn fest. Stecken Sie das letzte kurze Segment unter der Schnur in die Maschen.

Fahren Sie auf diese Weise mit der gesamten ersten Reihe fort, indem Sie kurz vor jeder LED einen Bindedraht anbringen. Am Ende der Reihe biegen Sie die Lichterkette U-förmig um und platzieren die nächste LED in der nächstgelegenen Reihe, sodass sie um eine Spalte von der vorherigen LED versetzt ist. Fahren Sie mit dieser Technik fort, um alle Reihen zu füllen, bis die Hälfte der LED-Kette in einem Zickzack-Muster am Netzbezug befestigt ist. Die LEDs sollten abwechselnd an den Kreuzungspunkten des Gitters sitzen. Vergewissern Sie sich, dass alle LEDs mit der Vorderseite nach oben ausgerichtet sind.

Weben Sie als Nächstes die zweite Hälfte der Lichterkette ein. Beginnen Sie damit, das Kissen um 90 Grad zu drehen, und legen Sie dann den nächsten losen Teil der LED-Schnur über die nächstgelegene Reihe am Rand.

So sieht der Bezug am Ende ohne Kissen aus.

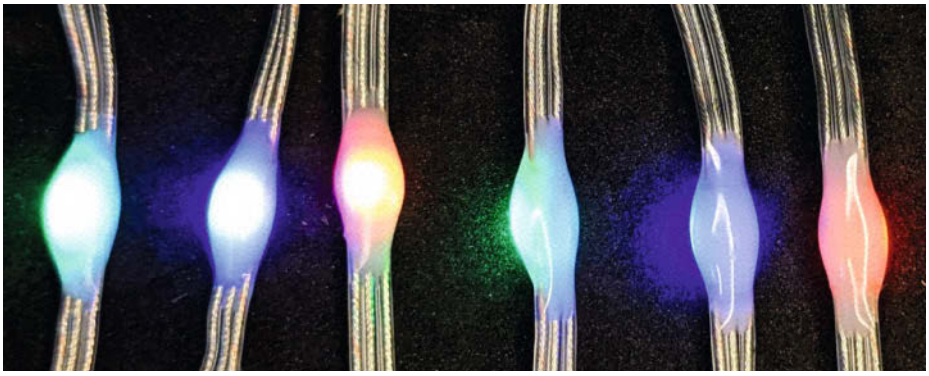
Befestigen Sie die erste LED an der Kreuzung, die der Ecke des Kissens am nächsten liegt. Fahren Sie damit fort, die LED-Kette entlang der Reihen zu verbinden, indem Sie den losen Teil der Kette unter jede der LEDs in senkrechten Spalten schieben.

Wenn Sie fertig sind, sollten die Eingangs- und Ausgangsanschlüsse der LED-Ketten von den gegenüberliegenden Kissenecken ausgehen, und die Kette wird zu einem Muster verwoben.

Sie können weitere Bindedrähte verwenden, um Teile der LED-Schnur zu befestigen, wenn sie lose erscheinen. Nachdem die erste LED-Matrix befestigt ist, wiederholen Sie den Vorgang mit dem zweiten Kissenbezug. Dieser Schritt kann bei zwei Kissen Stunden dauern; vielleicht möchten Sie sich mit einem Podcast oder einer Fernsehsendung ablenken.

Pixelblaze einrichten und einschalten

Als nächstes schließen Sie den Pixelblaze Controller und die Stromversorgung an eine der LED-Matrizen an. Sowohl das Pixelblaze als



Die LEDs der Lichterkette sollten alle so versiegelt sein.

Mit einem Seitenschneider oder einer Drahtschere kann man den Bindedraht leicht zuschneiden.

© Copyright by Maker Media GmbH.

Der doppeladrige Bindedraht mit Ummantelung soll die LED-Kette befestigen.

auch die LED-Kette arbeiten mit 5 V und können mit derselben Stromquelle betrieben werden. Führen Sie den Strom für die LED-Kette aber nicht über den USB-Micro-B-Anschluss des Pixelblaze, da dieser nicht dafür ausgelegt ist, so viel Strom zu führen. Die spezifische Verdrahtung hängt davon ab, ob Ihr LED-Ketten-Stecker zusätzliche Stromkabel hat, die herausgeführt sind. Ist dies der Fall, ist es einfach, diese zusätzlichen Stromkabel an den Schraubanschluss der Buchse anzuschließen. Dann bringen sie ein zusätzliches 3-poliges JST-SM-Kabel an den LED-String-Eingang und verbinden die Strom-, Daten- und Erdungsleitungen dieses Kabels mit den entsprechenden Anschlüssen des Pixelblaze.

Wenn Ihre Steckverbinder keine zusätzlichen Stromkabel haben, müssen Sie die 5V- und GND-Drähte des JST-Steckverbinders mit einem Spleiß verbinden. Dafür verbinden Sie die Stromkabel sowohl mit dem Pixelblaze-Controller als auch mit der LED-Lichterkette. Danach diese Drähte an die Schraubklemmen der DC-Buchse anschließen, um die Stromversorgung zu gewährleisten.

Sobald das Pixelblaze und der Stromadapter mit dem Eingang einer Matrix verbunden sind, verwenden Sie ein JST-SM-Verlängerungskabel, um den Ausgang des ersten Kissens mit dem Eingang des zweiten Kissens zu verbinden. Wenn Sie die Kissen weiter voneinander entfernt aufstellen möchten, können Sie ein zweites (oder drittes) Verlängerungskabel zwischen ihnen einfügen.

Wenn die Verkabelung abgeschlossen ist, stecken Sie das Netzteil in eine Steckdose, um alles einzuschalten. Nach dem Einschalten startet ein neues Pixelblaze-Board automatisch im Access-Point-Modus mit seinem eigenen ungesicherten WLAN-Netzwerk, das mit dem Präfix Pixelblaze beginnt. Verbinden Sie ein WLAN-fähiges Gerät mit dem Netzwerk und besuchen Sie dann http://192.168.4.1 im Browser des Geräts, um die Pixelblaze-WLAN-Einstellungen mit Ihrem lokalen Netzwerknamen und Passwort zu konfigurieren. Achten Sie darauf, das Kontrollkästchen „Enable discovery service“ zu aktivieren.

Nach der Konfiguration wird das Pixelblaze neu starten und sich mit Ihrem Netzwerk verbinden. Um es zu finden, besuchen Sie den Cloud Discovery Service unter discover.electromage.com, der alle Pixelblaze-Boards in Ihrem Netzwerk anzeigt.

Wenn Sie das Board auswählen, wird die Pixelblaze-Web-App in Ihrem Browser geöffnet, wo Sie die LED-Einstellungen und Anzeigemuster konfigurieren können. Wählen Sie zunächst die Registerkarte Einstellungen, um die Eigenschaften Ihrer LED-Ketten zu konfigurieren.

- Geben Sie Ihrer Platine einen aussagekräftigen Namen.
- Setzen Sie den LED-Typ auf WS2812 und die Datengeschwindigkeit auf 800 Kbps. Wenn Sie sich bei der Farbreihenfolge nicht sicher sind, wählen Sie GRB – wenn die Farben in einigen Mustern falsch aussehen, müssen Sie möglicherweise mit dieser Einstellung experimentieren, um die richtige Option zu finden.
- Setzen Sie die Anzahl der Pixel für die beiden LED-Matrizen auf 200. Es ist eine gute Idee, die LED-Helligkeit in den Einstellungen zu begrenzen, um den Stromverbrauch und die Wärmeabgabe zu reduzieren. Die Angabe von maximal 25 % Leistung ist in der Regel mehr als ausreichend, um ein helles Display zu erzeugen.

Zuordnung der LEDs

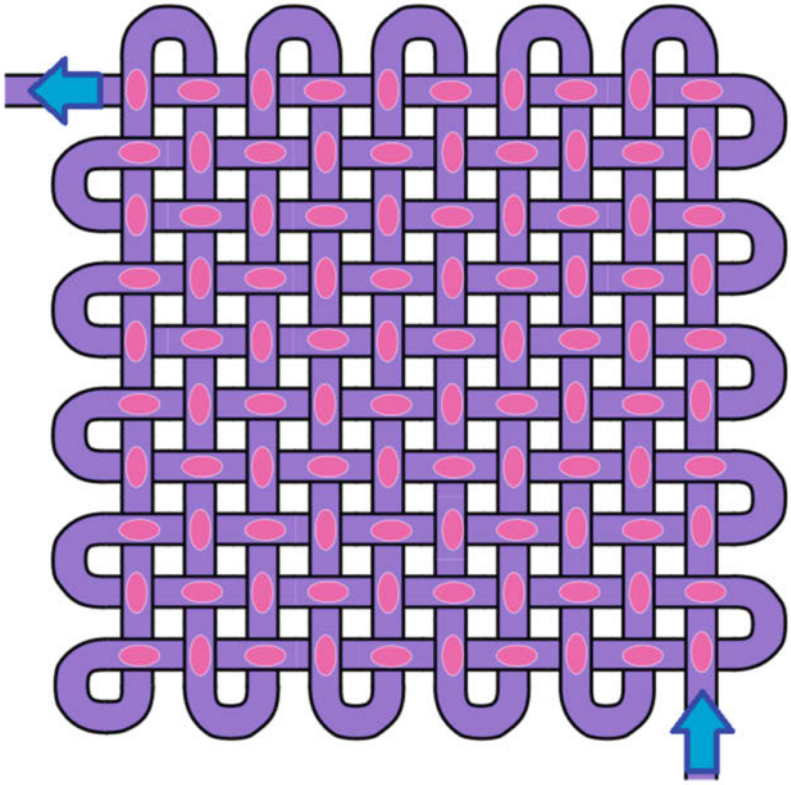
Die Pixelblaze-App speichert die LED-Konfigurationen unabhängig vom Animationscode,

Damit alles hält, muss der Bindedraht richtig verdreht werden.

© Copyright by Maker Media GmbH.

sodass das gleiche Muster ohne Änderung auf verschiedenen Layouts läuft. Um das Muster einer bestimmten Pixelanordnung zuzuordnen, müssen Sie der App die relative Position aller LEDs mitteilen. Geben Sie die LED-Positionen auf der Registerkarte Mapper an. Alle dort eingegebenen Koordinaten werden von der App automatisch auf einen Bereich von [0,1] skaliert, um eine standardisierte Verwendung im Code für verschiedene LED-Muster zu ermöglichen. Das Textfeld auf der Registerkarte Mapper akzeptiert eine formatierte Liste kartesischer Koordinaten, kann aber auch Pixelpositionen aus einem JavaScript-Programm generieren.

Laden Sie die Codedatei PixelPillowMapperCode.txt von der Projektseite herunter (siehe Link in der Kurzinfo) und fügen Sie ihren Inhalt in das Mapper-Textfeld ein. Dieser Code legt die Pixelpositionen für die beiden Kissen fest und verwendet variable Parameter, um ihre Ausrichtung und ihren Abstand zu definieren. Klicken Sie auf die Schaltfläche Speichern, um die neue Zuordnung beizubehalten. Sie wissen, dass die Zuordnung korrekt ist, wenn unter dem Mapper-Feld eine bestätigende Meldung in grünem Text erscheint und das Bild, das die LED-Positionen zeigt, dem phy-



Die LED-Kette auf dem Stoff wird in diesem Muster angeordnet.

2023 · 282 Seiten · 29,90 €
ISBN 978-3-86490-913-9

2023 · 494 Seiten · 34,90 €
ISBN 978-3-86490-936-8

Mit frischen Projekten durch den Herbst

2023 · 332 Seiten · 29,90 €
ISBN 978-3-86490-952-8

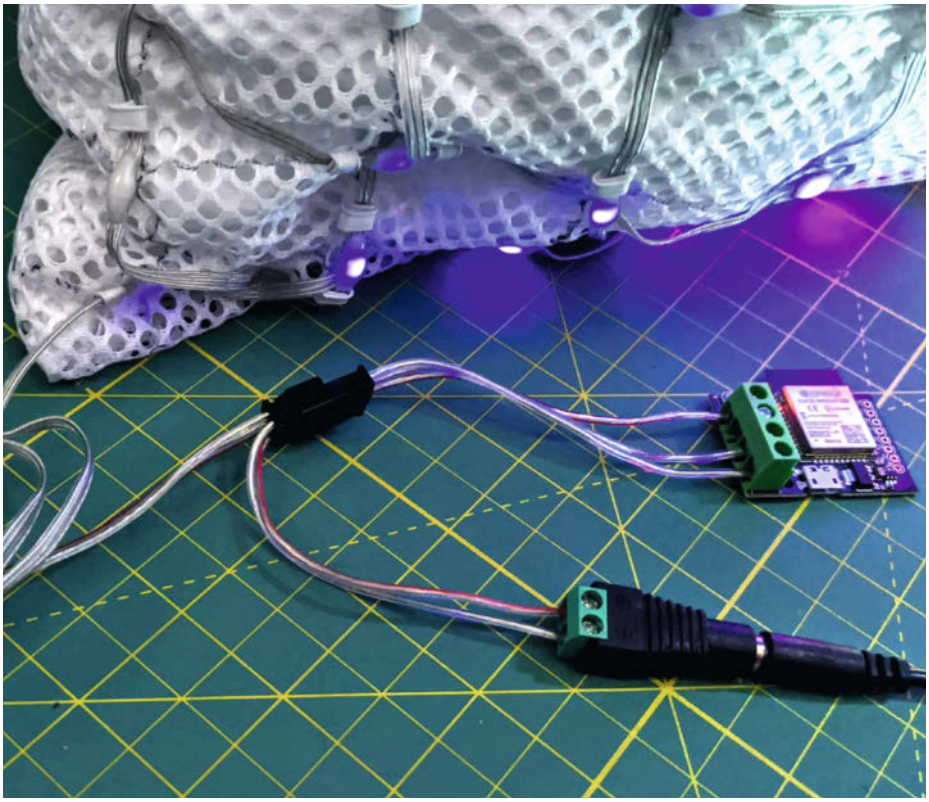
2023 · 346 Seiten · 39,90 €
ISBN 978-3-86490-937-5

2023 · 208 Seiten · 26,90 €
ISBN 978-3-86490-970-2

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

dpunkt.verlag

© Copyright by Maker Media GmbH.



Die LED-Kissen brauchen eine externe Stromversorgung für die Lichter und den Controller.

sischen Layout der LED-Matrizen Ihrer Kissen entspricht.

Sie können auch mit den Optionen zum Füllen und Begrenzen experimentieren. Diese dehnen das Muster, um den leeren Raum am Rand zu eliminieren oder beizubehalten, um das Seitenverhältnis des Musters zu erhalten, falls es nicht zu dem der Matrix passt. Diese Auswahl ist rein ästhetisch.

Anbringen von Diffusor und Abdeckung

Ein 47 cm × 47 cm großes Quadrat aus Steppwatte über den LEDs hilft, das ausgestrahlte Licht zu streuen und verhindert, dass die Bindedrahtstücke herausragen.

Legen Sie ein 47-cm-Quadrat aus Watte über jede Matrix und legen Sie dann das Kissen und die Watte in einen Samtbezug. Ziehen Sie an den Ecken, um sicherzustellen, dass es den Bezug gleichmäßig ausfüllt. Glätten Sie die Watte, falls sie sich verknäult hat, und schließen Sie dann den Verschluss am Samtbezug, sodass nur ein kleiner Spalt für die Verkabelung frei bleibt. Ordnen Sie die Pixelkissen auf einem Sofa oder Bett an und verbergen Sie dabei die Kabel, damit sie nicht sichtbar sind. Wenn Sie die Ausrichtung und Position des Kissens ändern, müssen Sie den Pixelblaze Mapper aktualisieren.

Die gesamte Länge der Lichterkette wird Stück für Stück auf dem Bezug befestigt.

TIPP: Der flauschige weiße Zottelstoff ist auch ein guter Lichtdiffusor, also nicht nur eine lustige Textur, sondern auch ein funktionales Element.

Wählen Sie Ihre Muster

Dieser letzte Schritt ist bei weitem der spaßigste in der Ausführung. Auf dem Pixelblaze Controller sind mehrere Dutzend Animationsmuster vorinstalliert. Um zwischen ihnen zu wechseln, klicken Sie einfach auf den Namen des Musters in der Registerkarte Muster der App. Die Anzeige des Kissens wird sofort aktualisiert. Sie können den Code der Muster im Code-Editor der App anzeigen und ändern. Wählen Sie die Schaltfläche Bearbeiten neben einem bestimmten Muster auf der Registerkarte Muster, um den Code in das Editorfenster zu bringen. Sie können auch direkt über die Registerkarte Bearbeiten auf den Musteditor zugreifen und dort neue Muster erstellen oder sie aus Musterdateien im .epe-Format hochladen. Es sind nur wenige Codezeilen erforderlich, um ein eigenes originelles Muster zu erstellen wie z. B. das Bullseye-Muster aus dem Screenshot.

Beim Erstellen oder Ändern von Mustern können Sie überprüfen, ob Ihre Farbeinstellungen und die Pixelanordnung korrekt sind, indem Sie das Vorschaufenster des Editors mit der tatsächlichen Kissenanzeige vergleichen.

Wenn Sie sich nicht trauen, Ihre eigenen Muster zu programmiern, ist es eine gute Mög-

© Copyright by Maker Media GmbH.

Wenn der Controller das erste Mal eingerichtet wird, erscheint diese Meldung.

lichkeit, kleine Änderungen am bestehenden Code vorzunehmen und die Auswirkungen in der Echtzeitanzeige zu beobachten.

Der Pixelblaze-Pattern-Editor hat zu viele nützliche Funktionen, um sie hier alle zu beschreiben, aber Sie können sie im Netz näher kennenlernen (zu finden über den Link in der Kurzinfo). Dort gibt es eine Open-Source-Bibliothek mit Pixelblaze-Mustern von verschiedenen Mitwirkenden. Diejenigen mit einer 2D-Bezeichnung im Namen sehen im Allgemeinen auf dem Kissendisplay am besten aus. Und der Programmierer Zranger1 hat in seinem Pixelblaze-Repository eine wunderbare Sammlung komplizierter 2D-Muster erstellt, die empfehlenswert sind.

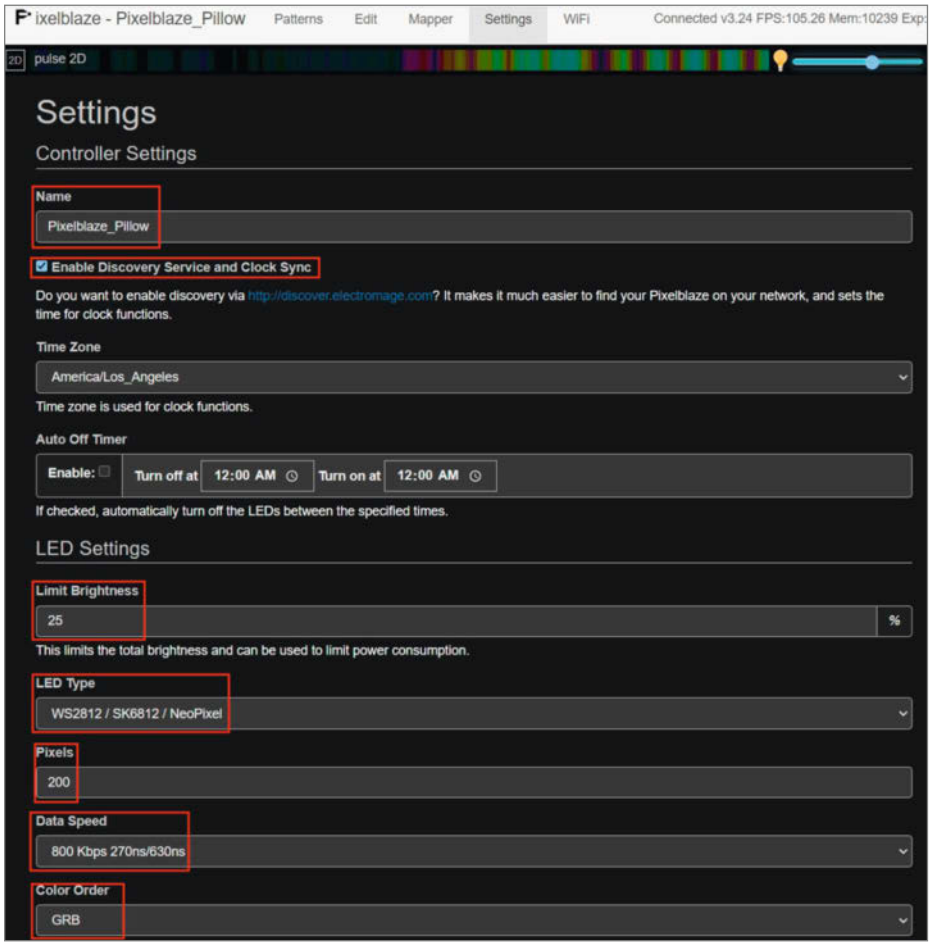
Funkelnde Freude

Diese Kissen sind ein DIY-Projekt, das sich sehen lassen kann. Sie verwandeln sich von harmlosen Wohnaccessoires in etwas viel Schöneres und Interessanteres, wenn man nur einen Schalter betätigt. Fotos können die dynamische Schönheit der ständig wechselnden LED-Animationen nur schwer wiedergeben, aber die Bilder in diesem Artikel sind ein kleines Beispiel dafür, was möglich ist. Sie halten auch einiges aus, denn die wetterfesten LED-Strings für den Außenbereich lassen sich leicht biegen. Ich würde sie nicht unbedingt für eine Kissenschlacht empfehlen, aber sie überstehen problemlos den Gebrauch normaler Kissen (solange niemand an der Verkabelung reißt). Und die äußeren Bezüge lassen sich zum Waschen abnehmen.

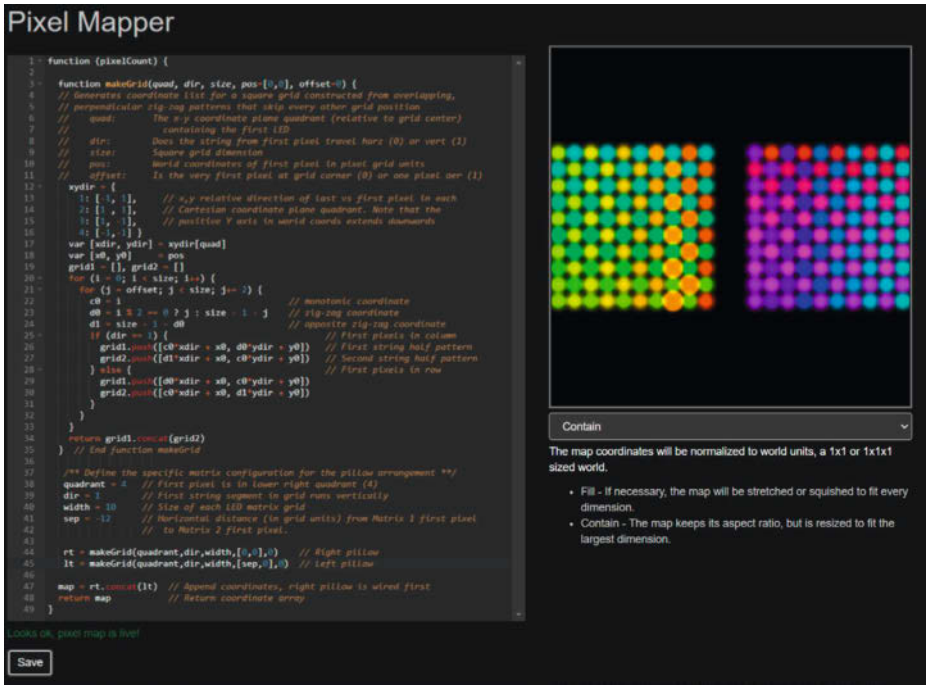
Nächste Schritte

Wenn Ihnen Ihre LED-Matrix-Kissen gefallen, finden Sie hier ein paar Ideen, wie Sie dieses Projekt weiterführen können:

- Verwenden Sie die Playlist-Funktion der Pixelblaze-App, um Ihre Lieblingsmuster automatisch abzuspielen.
- Fügen Sie eine Pixelblaze-Sensorplatine für Klang- oder Bewegungsreaktivität hinzu.
- Erstellen Sie zusätzliche Kissen, die Sie in Reihe schalten können, um ein größeres dekoratives Display zu erhalten.
- Nähen Sie eine passende LED-Decke.



Über das Web-UI der Kissen können viele Einstellungen vorgenommen werden.

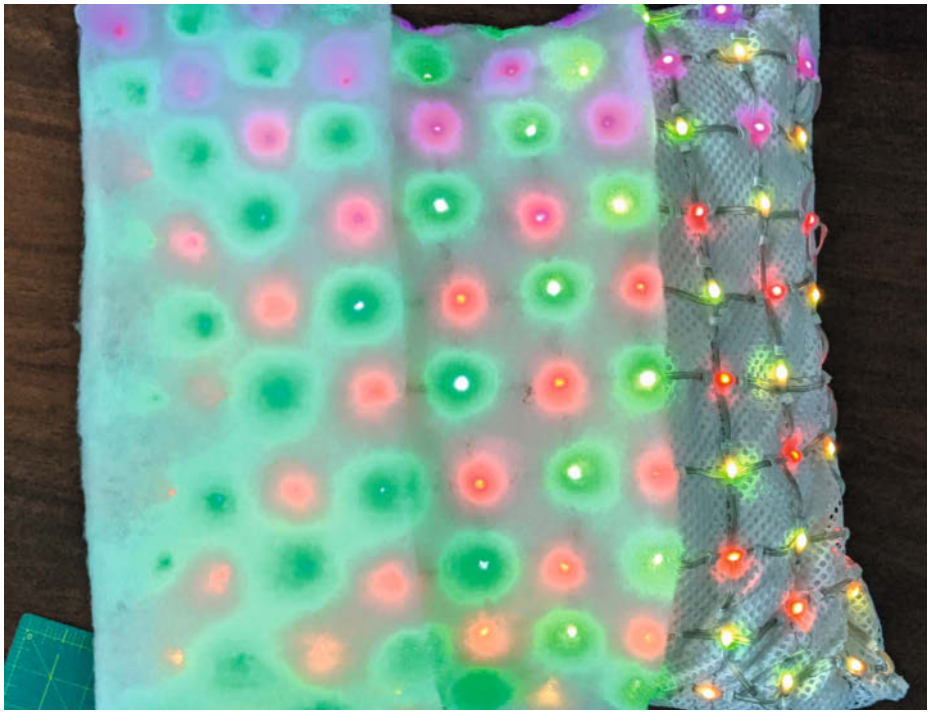


Neue Muster lassen sich direkt in der Web-App auf die Kissen aufspielen.

© Copyright by Maker Media GmbH.

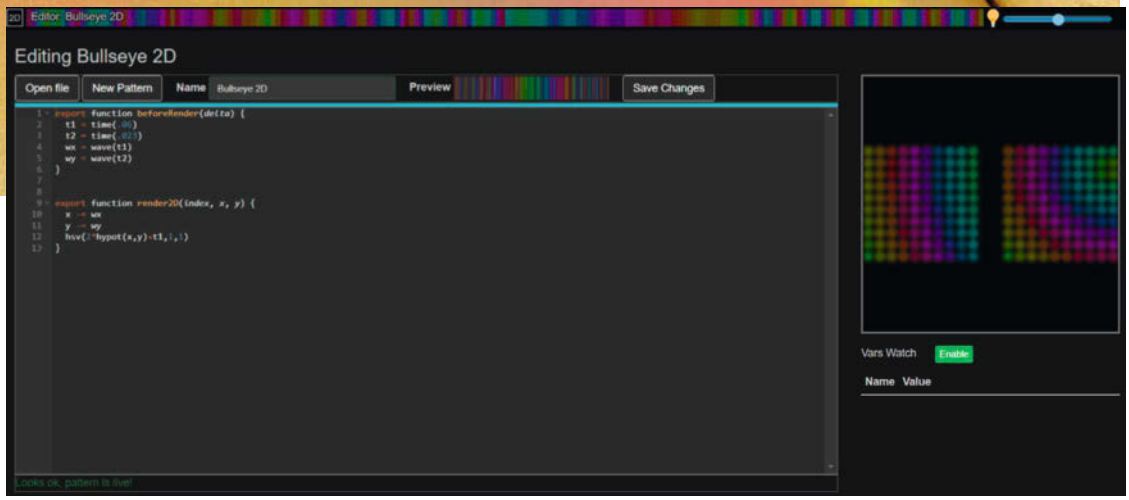
– Schalten Sie auf Batteriebetrieb um. Ich habe die Kissen kontinuierlich mit einer Helligkeit von 25 % betrieben. Die Stromaufnahme blieb die ganze Zeit unter 2 A, sodass jede tragbare Powerbank, die 2 A leisten kann (viele der Geräte mit größerer Kapazität können das), für den mobilen Einsatz gut geeignet ist.

Es hat mir viel Spaß gemacht, an diesem LED-Bauwerk zu basteln, und ich hoffe, dass auch andere sich an der einfachen Konstruktion und dem vielseitigen Display erfreuen werden. Es ist eine buchstäblich brillante Heimdekoration. *—das*



Der Unterschied im Erscheinungsbild von LED-Licht, das durch (von rechts nach links) kein Streumaterial, eine einzelne Lage Watte und eine doppelte Lage scheint.

So kuschelig und einladend sehen die selbstgemachten LED-Kissen aus.



Eigene Muster können direkt im Browser programmiert werden.

© Copyright by Maker Media GmbH.

Alles im Blick

mit dem Sonderheft zur ESP32-Kamera

Inklusive ESP32-CAM Development Board + OV2640 Kameramodul

Das 80-seitige Make-Special zeigt Ihnen, wie Sie aus dem mitgelieferten ESP32-CAM-Board samt 2-MP-Kameramodul und Programmier-Adapter das Meiste rausholen. Damit können Sie sofort Ihr erstes Funk-Kamera-Projekt starten!

- Kamera-API im Griff
- Projekte: Nistkasten mit WLAN, Zeitrafferkamera, Objekterkennung mit KI und mehr…
- Bilder per Mail verschicken
- Tipps & Hacks: Externe Antenne anschließen, Kamera auf Infrarot umbauen, Reset-Pin nachrüsten

Heft inklusive ESP32-CAM Development Board + OV2640 Kameramodul 29,90 €

shop.heise.de/make-esp32cam

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

© Copyright by Maker Media GmbH.

heise Shop

Bauen, tüfteln, basteln
Die Maker Faire Hannover

Hämmern, Sägen, Löten, Schweißen, Programmieren – das alles und vieles mehr gab es am 19. und 20. August 2023 auf der DIY-Messe Maker Faire in Hannover zu sehen, zu hören und zu riechen. An über 150 Ständen zeigten Maker aus aller Welt ihre Projekte und tauschten sich mit den Besuchern und untereinander aus.

von Daniel Schwabe

HANNOVER
Maker Faire®
19.-20. August
2023

© Copyright by Maker Media GmbH.

Vom 19. bis 20. August 2023 fand im Hannover Congress Centrum die 9. Maker Faire Hannover statt. Mitten drin ich, als noch ziemlich neuer Technical Writer für das Make Magazin. Das erste Mal mit neugierigem Blick auf der gut besuchten, wuseligen und hochinteressanten Messe. Als Frischling in Deutschlands gefährlichstem DIY-Magazin war ich gemeinsam mit Maskottchen Makey begeistert von den vielen Projekten, den begabten Tech-Enthusiasten und dem Einblick in eine kreative und höchst einfallsreiche Community.

Wer noch nie auf der Messe war, sollte wissen: Generell ist die Maker Faire erst einmal überwältigend. Zum einen sind da die vielen kreativen Menschen vor Ort. Um genau zu sein, waren es 14.200 Besucher, die sich von den Projekten beeindrucken und inspirieren ließen. Es war viel über neue Produkte und Neuerungen zu erfahren und an den Ständen gab es jede Menge Mitmach-Aktionen. Zum anderen ist es vorrangig eine Technikmesse. An allen Ständen – und auch dazwischen – blinkte, piepte und surrte es um die Wette.

In den Hallen erreichen einen unglaublich viele Eindrücke auf einmal. Man weiß gar nicht, welchem riesigen 3D-gedruckten Blickfang man sich zuerst zuwenden soll; welcher Bildschirm gerade am interessantesten ist, und welcher Maker gerade Zeit für einen kurzen Schwatz hat. Das Gute daran ist: Man kann nichts falsch machen. Egal wo man hingeht, man stößt auf etwas Interessantes. Bei etlichen anwesenden Industrie-Firmen konnte man außerdem gleich ein cooles neues Gadget, eine Plattform oder Schnickschnack wie farbwechselndes Filament bekommen.

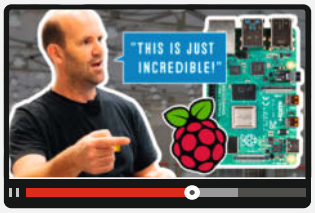
Große Augen, große Ohren, offene Münder

Nicht nur das Bauen von Geräten ist eine DIY-Disziplin, sondern auch Cosplayen bedarf viel handwerklichen Geschicks. Und – ohne zu übertreiben – auf der Maker Faire waren ALLE da: Jedi-Ritter, Aliens, der Joker höchstpersönlich und in Felle eingewickelte „Krieger". Die detaillierten Kostüme zogen so manchen Blick auf sich und die Fotoapparate klickten fleißig. Neben dedizierten Cosplayern waren auch die Steampunker vertreten, die Outfits im Stile der viktorianischen Epoche trugen, welche mit mechanischer Technik und viel Metall angereichert wurde.

Neben sämtlichen Robotern des Star-Wars-Universums, die munter zwischen den Gästen wuselten und Zeppelinen, die durch die Lüfte pflügten, konnte man sich auch akustisch von der Kreativität der DIY-Community überzeugen. „PonyTrap" verwöhnte die Maker-Ohren in regelmäßigen Abständen mit Trommeleinlagen. Aber natürlich nicht mit IRGENDWELCHEN Trommeleinlagen, sondern mit Musik von ihrem 3 Meter hohen Schlagzeugroboter.

Auch Interessant

» Eben Upton, Raspi-Mitbegründer im Interview mit Ákos Fodor. Auf Seite 32 und auf Youtube
» Johannes Börnsen, Retro-Leuchtreklame im Eigenbau: Ein Briefkasten für die Maker Faire



Alles zum Artikel im Web unter *make-magazin.de/x81a*

Mit dieser passenden Untermalung konnte man sich dann zum Beispiel das weltgrößte Streichholzschiff „Meteor" anschauen oder diverse Lego-Apparaturen dabei beobachten, wie sie entweder gehackt wurden, um als CNC-Fräse genutzt zu werden oder wie sie fantasievoll eine Kugel im Kreis um einen Stand herum beförderten.

Wer aber glaubt, ein drei Meter großer Trommelautomat war das Größte an Opulenz und Blickfang, der irrt gewaltig. Denn selbstverständlich machte es die Maker Faire nicht unter einem riesigen Metallpferd, das im Außenbereich durch die Menschen galoppierte.

Ein Foto aus Area 51 oder von der Maker Faire Hannover?

Das Lebensgefühl auf der Maker Faire

Man merkt schon, auf einer Maker Faire gibt es wirklich alles! Von Kunst über selbstgebaute Musikinstrumente bis hin zu funktionalen Projekten, wie spezielle Adapterstücke, um Staubsauger an Standbohrmaschinen anzuschließen.

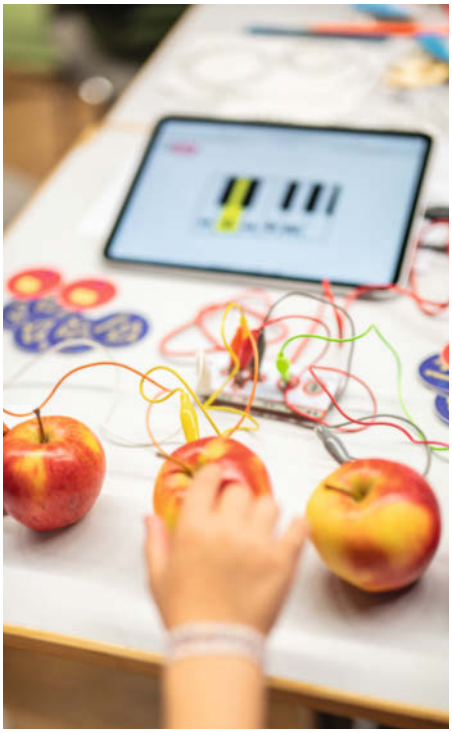
Auch bei entspannten 30 Grad im Schatten haben es sich die echten Fans nicht nehmen lassen, ihre besten Kostüme aufzutragen.

© Copyright by Maker Media GmbH.

Auch die Trommel wurde auf der Maker Faire nur im DIY-Style geschlagen.

Hüh-Hott – auf der Maker Faire schalten auch die Pferde in den höchsten Gang.

Und es ist offensichtlich, Maker-Themen sind sehr vielfältig. Alle Teilnehmenden der Messe haben etwas Spezielles, mit dem sie sich beschäftigen. Sie bewegen sich in Nischen, von denen viele nicht einmal wissen, dass es sie gibt. Es gibt Leute, die zeigen, was man aus ausrangierter 90er-Jahre Fernsehtechnik noch rausholen kann. Oder Menschen, die Fernschreiber mit dem Internet verbinden und so Nachrichten austauschen. Egal, wie obskur einem selbst ein Projekt von der Maker Faire erscheint, wenn man sich vor Ort mit den Enthusiasten unterhält, bemerkt man sofort die Leidenschaft, mit der getüftelt und gebaut wird und das Funkeln in den Augen bei Gesprächen darüber.



Eine Apfel-Orgel. Spielerischer Zugang zu Technik und anderen Maker-Themen wird auf der Messe groß geschrieben.

Die Aussteller und Ausstellerinnen auf der Maker Faire sind wirklich kreativ und fantasievoll. Als Besucher fühlt man sich da gleich ein bisschen überwältigt. Aber Selbstzweifel sind nicht angebracht, denn niemand auf der Messe ist dort, um jemand anderen zu bewerten. Alle zeigen stolz ihre Projekte und freuen sich über Interesse aus der Community. Fachsimpeln muss sein, auch wenn nicht alles immer perfekt aussehen oder funktionieren muss. Es können Kabel aus den Gehäusen hängen oder unperfekte Lötstellen zu sehen sein. Wichtig ist nur, dass es selbst gemacht ist und man es der Welt präsentieren will.

Auch Networking ist hier wichtig. Von einem Autor des Magazins habe ich gehört, dass er sich direkt mit drei anderen Ausstellern kurzgeschlossen hat, um im nächsten Jahr als Dreiergespann einen Stand aufzubauen und dort die Symbiose der verschiedenen Projekte zu zeigen.

Denn neben Einzelpersonen haben sich auch Teams als Aussteller angemeldet. Die Gruppen treten mal als allgemeiner Makerspace auf oder als Club, der sich einem ganz bestimmten Aspekt oder Projekt verschrieben hat.

Aber es gab dieses Jahr nicht nur alles, sondern es gab auch FÜR alle etwas. An vielen Ständen waren Mitmachaktionen für die kleinsten Makerinnen und Maker aufgebaut. Kinder konnten mit Lego ihrer Kreativität freien Lauf lassen, die simple Faszination von selbstgemachtem Holzspielzeug entdecken oder Roboter bauen, die später vielleicht einmal die Menschheit auf dem Mars unterstützen werden.

Und das ist auch ein ganz wichtiger Punkt auf der Maker Faire Hannover: Jeder kann Maker werden. Ob man sich als Kind mit Legorobotern in das Thema einarbeitet und später mit Arduinos und Co die Posteulen fliegen lässt, oder als Erwachsener in einen 3D-Drucker investiert, um sich kleine Ersatzteile zu drucken. Making ist eine Lebenseinstellung. Der Wunsch, ein konkretes Projekt mit den eigenen Fähigkeiten umzusetzen und sich dann an dem Ergebnis zu erfreuen.

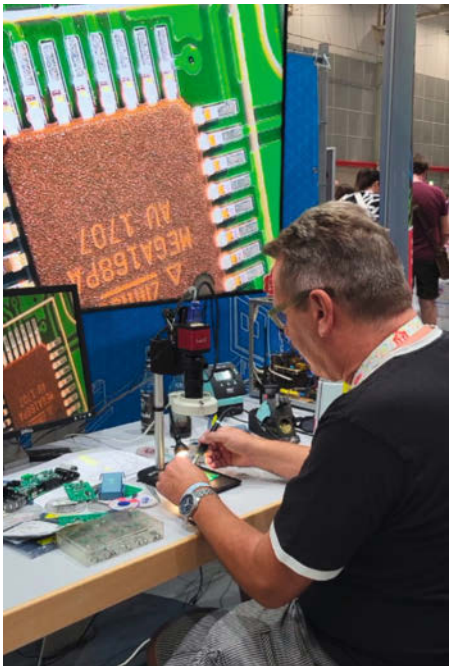
Prominenz und die Zukunft des Landes

Aber die Maker Faire ist natürlich nicht nur ein Event für Endnutzer, sondern auch für Industrieveteranen. So hat dieses Jahr Eben Upton, der Gründer der Raspberry Pi Foundation, Hannoveraner Messeluft geschnuppert und der Make nicht nur ein ausführliches Interview gegeben, sondern auch den Tetris Highscore am Make-Stand aufgestellt. Das ganze Interview gibt es unter dem YouTube-Link in der Onlineinfo und im Heft ab Seite 32.

Im Rahmen der Messe gab es noch einige besondere Events. Zum zweiten Mal wurden Lehrerinnen und Lehrer vom Make-Magazin, der Region Hannover und dem Niedersächsischen Landesinstitut für Schulische Qualitätsentwicklung zu Fortbildungen zum Thema MINT in der Schule eingeladen.

Mit Workshops und Vorträgen zum Thema Programmierung und der Relevanz von MINT im Klassenzimmer wurden ihnen viele Infor-

© Copyright by Maker Media GmbH.



Mit ruhiger Hand lötete Carsten Meyer Schaulustigen auf der Messe etwas vor.

Prominenz am Make-Stand: Eben Upton ledert die Redaktion in Tetris ab.

mationen an die Hand gegeben, mit denen sie den eigenen Unterricht mit pfiffigen Komponenten anreichern können. Mit dabei: Julia Willie Hamburg, Niedersachsens Kultusministerin. Sie machte auch den großen Presse-Rundgang mit und ließ sich von Make-Redakteur Peter König über die Messe führen und von den Projekten beeindrucken.

Der Letzte macht die Lichter aus – bis nächstes Jahr

Heimlich, still und leise hatte sich auch der langjährige, ehemalige Make-Redakteur Carsten Meyer auf die Messe geschlichen und gab am Make-Stand eine Löt-Show. So hat er es auch dieses Jahr wieder (trotz Renteneintritt) auf das Team-Foto der Redaktion geschafft, die nach zwei Tagen begeistert und mit vielen neuen Eindrücken, Inspirationen und Ideen ihre Zelte im Hannover Congress Centrum abbrach – aber natürlich schon wieder überlegt, was man bis zum nächsten Jahr entwickeln und bauen könnte.

Am 24. und 25. August 2024 wird in Hannover wieder zur Maker Faire geladen. Wer nicht bis dahin warten will, kann am 21. Oktober 2023 die Maker Faire Sindelfingen besuchen. —*das*

„Hands On“ heißt es auch für Niedersachsens Kultusministerin Julia Willie Hamburg.

Die Make-Redaktion – zufrieden am Ende einer erfolgreichen Maker Faire. Das Bild zeigt unten von links Carsten Wartmann, Ákos Fodor, Dunia Selman, Louis Behrens. Oben von links Johannes Börnsen, Peter König, Heinz Behling, Daniel Schwabe und Carsten Meyer (nicht im Bild der Corona-erkrankte Chefredakteur Daniel Bachfeld).

© Copyright by Maker Media GmbH.

Interview mit Eben Upton

Auf der Maker Faire Hannover gab es nicht nur sehenswerte Projekte, sondern ebenso viele interessante Menschen – darunter auch den Mitbegründer des Raspberry Pi. Wir haben die Chance genutzt und ihm ein paar Fragen gestellt.

von Ákos Fodor

© Copyright by Maker Media GmbH.

Wenn ihr auf der Maker Faire wart, seid Ihr mit hoher Wahrscheinlichkeit an Eben Upton vorbeigelaufen und habt es gar nicht gemerkt. Kein Wunder, denn der sympathische Brite, dem wir den Raspberry Pi verdanken, fügte sich mit seinem lässigen Look und der aufgeschlossenen Art nahtlos in das Gewusel ein. Richtige Maker glänzen nunmal mit ihren Projekten und so unterstützte er am Berrybase-Stand neugierige Besucher mit Rat und Tat. Zu unserem Glück durften wir ihn zwischendurch kurz zur Seite nehmen und interviewen. Das ursprünglich in Englisch geführte Gespräch haben wir für das Heft mithilfe von DeepL ins Deutsche übersetzt und redaktionell bearbeitet. Die Original-Fassung findet ihr auf unserem YouTube-Kanal (siehe Link).

Make: Hallo Eben Upton, vielen Dank, dass Sie hier sind. Wie ist Ihr bisheriger Eindruck?

Upton: Danke, dass ich hier sein darf. Es ist unglaublich (zeigt auf die Maker Faire im Hintergrund). Das ist meine erste Maker Faire seit Corona und ich glaube, sogar meine erste Maker Faire seit etwa 2017, denn wir bekamen ein Baby und dann kam Corona. Ich war also schon lange nicht mehr auf einer Maker Faire und finde sie einfach unglaublich.

Make: Waren Sie schon einmal hier?

Upton: Bisher noch nicht. Ich war früher ziemlich regelmäßig auf den amerikanischen Maker Faires, der World Maker Faire in New York, auf der Bay Area Maker Faire in San Mateo und auf einer oder zwei in Großbritannien, aber noch nie in Deutschland.

Make: Dann haben sie ja vielleicht einen neuen Anlaufpunkt.

Upton: Auf jeden Fall. Das ist mein neues Zuhause.

Make: Sehen Sie sich selbst auch als Maker?

Upton: Ja, absolut. Das ist die Kultur, aus der wir kommen, auch wenn sie damals in Großbritannien vielleicht ein bisschen weniger entwickelt war, als wir mit den Raspberry Pis anfingen. Wir waren schon Maker, aber der Idee des Machens (Making) als Kultur nicht so weit verschrieben, wie es wahrscheinlich in den USA oder auch hier in Deutschland der Fall ist.

Aber als der Raspberry Pi sich auch außerhalb Großbritanniens verbreitete, kamen wir in Kontakt mit der Maker-Community und der Maker-Kultur. Und wir haben irgendwie festgestellt, dass wir in ihr sozusagen unser Zuhause gefunden haben.

„Ich mache zwar eines der besten Spielzeuge der Welt mit dem Raspberry Pi, aber ich habe vergleichsweise nur wenig Zeit, um mit ihm zu experimentieren."

Make: Arbeiten Sie denn auch privat an Maker-Projekten?

Upton: Nun, ich sollte es, aber ich schaffe es nicht. Eine der Tragödien des Raspberry Pi ist für mich persönlich, dass er als Projekt meine ganze Zeit in Anspruch nimmt – das und natürlich meine Familie. Deshalb verbringe ich heutzutage verhältnismäßig wenig Zeit mit Spielen. Ich mache zwar eines der besten Spielzeuge der Welt mit dem Raspberry Pi, aber ich habe vergleichsweise nur wenig Zeit, um mit ihm zu experimentieren. Es geht um die Work-Life-Balance. Aber geben Sie mir fünf Jahre und vielleicht sieht es dann anders aus.

Make: Wann können wir denn mit dem nächsten Spielzeug in Form eines Raspberry Pi rechnen?

Upton: Nun, es wird offensichtlich irgendwann einen Raspberry Pi 5 geben. Das Seltsame am Raspberry Pi 4 ist natürlich, dass er jetzt schon vier Jahre alt ist. Wir haben ihn im Juni 2019 auf den Markt gebracht. Aber weil wir zwischendurch eine Pandemie und eine Lieferkettenkrise hatten, fühlt es sich gleichzeitig wie ein ziemlich altes Produkt an, aber auch wie ein Produkt, das gerade erst anfängt, auf die Beine zu kommen. Selbstverständlich müssen wir noch etwas Neues machen. Unser Entwicklungsteam hat in den letzten Jahren viel Zeit damit verbracht, die Kernprodukte so umzugestalten, dass sie trotz der Lieferengpässe weiterhin hergestellt werden können.

Make: Und wie sieht es mit Neuentwicklungen aus?

Upton: Die Leute waren vielleicht ein wenig davon abgelenkt, darüber nachzudenken, was wir als Nächstes tun werden. Aber jetzt, da die Krise überwunden ist, ist es natürlich großartig, dass wir hier auf der Maker Faire die Raspberry Pis in unbegrenzten Mengen zum angegebenen Preis verkaufen können. Das ist ein echtes Zeichen dafür, dass wir die Krise hinter uns gelassen haben. Und das Ingenieurteam hat jetzt ein bisschen mehr Zeit, um darüber nachzudenken, wie ein Raspberry Pi 5 oder ein Raspberry Pi Pico 2 oder so etwas aussehen könnten.

Make: Mit dem Raspberry Pi 4 lassen sich ja auch schon allerhand anspruchsvolle Projekte umsetzen.

Upton: Ja, und er hat ja auch ein Midlife-Upgrade bekommen. Wir starteten damals mit 1,5GHz. Moderne Raspberry Pi 4 laufen jetzt mit 1,8GHz. Dafür waren auch Änderungen an der Stromversorgung notwendig, die den 1,8GHz-Betrieb erst ermöglichen. In einer früheren Generation hätten wir das Upgrade vielleicht als Plus-Produkt bezeichnet oder 4 Plus genannt. Stattdessen haben wir die Neuerungen aber in die laufende Kette eingegliedert.

© Copyright by Maker Media GmbH.

Make: Wie kam es eigentlich zu der Entscheidung, mit dem Raspberry Pi Pico einen eigenen Mikrocontroller auf den Markt zu bringen?

Upton: Nun, das ist interessant. Ich hatte heute schon auf der Messe die Gelegenheit, darüber zu sprechen. Als Ingenieure nutzen wir individuell alle schon seit 10, 20, 30 Jahren Mikrocontroller. Der Raspberry Pi selbst, unser Kernprodukt, besitzt kein Mikrocontroller-Element. Aber fast jedes Zubehör, das wir herstellen, verfügt über eine Art Mikrocontroller, der das Verhalten des Zubehörs steuert. Wir haben also eine Menge Erfahrung mit Mikrocontrollern und etwa um das Jahr 2017 herum hatten wir schon eine ziemlich gute Vorstellung davon, was uns an den Mikrocontrollern anderer Leute gefiel und was nicht. Wenn man etwas in Millionenstückzahlen pro Jahr verwendet, fragt man sich natürlich irgendwann, ob man es vielleicht selber besser machen könnte.

Make: Und daraus entstand dann der RP2040?

Upton: Er ist sozusagen die Summe aller Erfahrungen des Teams mit den guten und den schlechten Aspekten anderer Plattformen und hat sich für uns wirklich gut bewährt. Wir hatten natürlich auch Glück, dass er im Januar 2021 auf den Markt kam, als sich die Lieferketten für das Kernprodukt gerade verengten. Zugang zu Chips zu haben, die uns fast unbegrenzt zur Verfügung standen war wirklich hilfreich, da wir einen Teil der Produktion des Raspberry Pi auf industrielle Abnehmer umleiten mussten. Eine weitere Produktlinie zu haben, die nicht knapp war und die wir nutzen konnten, um mit der Maker-Community in Kontakt zu bleiben, war wirklich wertvoll für uns und ich denke, auch für die Community.

„Bei Python ist die unterste Sprosse der Leiter, wie bei BASIC, sehr nah am Boden. Aber anders als bei BASIC ist die oberste Sprosse in den Wolken. Man kann auch in Python riesige Stücke professioneller Software schreiben."

Make: Wenn man auf die Geschichte des Raspberry Pi zurückschaut: Erfüllt der Pico das ursprünglich gesetzte Ziel vielleicht sogar noch besser?

Upton: Das tut er. Und das Seltsame daran ist, dass die Programmiererfahrung in gewisser Weise sogar eher dem entspricht, was ich mir ursprünglich vorgestellt habe. Es ging mir um eine sehr reduzierte Programmiererfahrung. Und das ist die Art von Kommandozeilenerfahrung, die man bekommt, wenn man sich mit einem seriellen Terminal verbindet. Wir haben eine nette kleine Demo am Stand von Berrybase laufen, wo man mit NeoPixeln und einem Raspberry Pi Pico experimentieren kann. Ich war in der Lage, sehr schnell etwas mit einem Pico W zu bauen, das die Informationen darüber abruft, welche Astronauten sich gerade im Weltraum befinden.

Make: Wieviele sind es denn?

Upton: Im Moment sind zehn Menschen im All: Sieben auf der ISS und drei auf der chinesischen Station. Und ich konnte meine NeoPixel mit sieben blauen und drei grünen Lichtern zum Leuchten bringen. Es gibt eine Web-API, die Sie aufrufen können, um das hinzubekommen. Das ist toll und der Code besteht nur aus wenigen Zeilen. Und es ist eine unmittelbare Programmiererfahrung. Man bootet nicht einen Linux-Computer und lässt dann Python darauf laufen, sondern man hat einen Rechner, der in Python bootet, so wie ein Commodore 64 in BASIC gebootet hat. Vielleicht sind meine ursprünglichen Ansichten falsch, aber ich sehe einen Wert in dieser unmittelbaren Programmiererfahrung.

Make: Für den RP2040 haben Sie sich für MicroPython entschieden. Ist das ein besserer Einstieg als BASIC?

Upton: Wir glauben, dass Python der natürliche Nachfolger von BASIC ist. Es besitzt diese Einfachheit in der Syntax, wo „print hello world" buchstäblich nur „print hello world" ist. Und es gibt kein umständliches Gerüst. Sie wissen schon, wenn Sie sich „Hallo Welt" in Java ansehen, ist der Code ungefähr 10 Zeilen lang und es gibt eine Menge drumherum. Man muss eine Klasse erstellen und eine statische Methode und so weiter. Das ist ziemlich viel am Anfang und wenn man versucht, das jemandem beizubringen, muss man sagen: „Oh, mach dir keine Sorgen um dies. Oh, kümmere dich nicht um das." Das ist eine wirklich schlechte Art, ins Programmieren einzusteigen und eine wirklich schlechte Ausgangsbasis, wenn man versucht, jemandem etwas über Computer beizubringen. „Oh, dieser Teil ist schwierig, der dort auch und jener erst." Stattdessen hat man „print hello world".

Make: Und wenn man Erfahrungen gesammelt hat, kann man ja auch komplexere Programme schreiben.

Upton: Bei Python ist die unterste Sprosse der Leiter, wie bei BASIC, sehr nah am Boden. Aber anders als bei BASIC ist die oberste Sprosse in den Wolken. Man kann auch in

© Copyright by Maker Media GmbH.

Python riesige Stücke professioneller Software schreiben. Das ist und war schon immer unsere Auffassung von Python. Und die Entwicklung von MicroPython durch Damien George in den letzten fünf Jahren hat es uns wirklich ermöglicht, all diese Werte in den Mikrocontroller-Bereich zu bringen. Das ist großartig, wissen Sie!

Wenn Sie wollen, können Sie Ihren RP2040 auch gern in C programmieren. Das ist eine der von uns unterstützten Optionen, Programmierung in C, C++ oder Assembler. Aber wenn es nur darum geht, etwas in einer eingebetteten Umgebung zu machen – und in 90% der Fälle wollen Leute genau das – solange Sie genug Speicher und CPU-Leistung haben, können Sie Ihren Code in Python schreiben und eine viel höhere Produktivität erreichen.

Und MicroPython unterstützt zudem Inline-ARM-Assembler und Inline-PIO-Assembler. Man kann also tatsächlich den PIO (Programmed Input/Output) ansteuern, den interessantesten Aspekt der RP2-Architektur. Man kann ihn tatsächlich steuern, ohne dafür C zu verwenden.

„Massimo und Fabio (von Arduino) sind gute Freunde. Wir kennen diese Jungs. Und ich hoffe, dass wir noch mehr Produkte zusammen machen können."

Make: Es scheint auf dem Mikrocontroller-Markt einige Kooperationen zu geben. Wie ist Ihr Verhältnis zu anderen Herstellern?

Upton: Zweifellos sind wir sehr gute Freunde von Arduino. Sie haben den Weg beschritten, Kunden aus dem Bildungsbereich zu unterstützen, Maker zu befähigen und ihnen beim Übergang in professionelle Rollen zu helfen. Damit waren sie uns wahrscheinlich fünf Jahre voraus. Das Interessante an Arduino und Raspberry Pi ist, dass es selbst jetzt, wo wir den Pico haben, nur sehr selten signifikante Überschneidungen zwischen Dingen gibt, die man eher auf einem Raspberry Pi oder auf einem Arduino macht. So haben wir es geschafft, mit Arduino befreundet zu bleiben. Jetzt stellen sie den RP2040 Nano Connect her, ein Arduino-Produkt, das auf unseren Chips aufbaut. Und sie waren einer unserer frühen Partner. Sie hatten drei Monate vor der Markteinführung Zugang zu unserem Chip-Design. Massimo und Fabio sind gute Freunde. Wir kennen diese Jungs. Und ich hoffe, dass wir noch mehr Produkte zusammen machen können.

Make: Viele Maker sind daran interessiert, mit Elektronik zu spielen und zu lernen, möchten aber gleichzeitig keine unethischen Arbeitsbedingungen unterstützen. Lässt sich das vermeiden?

Upton: Ich kann das durchaus verstehen und bin mir nicht sicher. Ich meine, offensichtlich sind wir jetzt ein verhältnismäßig großes Unternehmen. Es gibt eine ganze Reihe von Maßnahmen rund um die Lieferkette und die Ethik in unserer Lieferkette. Wir arbeiten sehr hart daran, denn während wir unsere Produkte zwar nicht direkt in China herstellen, beziehen wir viele unserer Komponenten und Produkte von dort. Wir haben einige fantastische chinesische Partner, vor allem im Bereich der Elektromechanik, z.B. für Leiterplatten, Steckverbinder und Zubehör. Und wir tun unser Bestes, um sicherzustellen, dass all diese Leute sich so verhalten, dass wir stolz auf sie sein können.

Das ist der Test, wissen Sie? Wäre man glücklich, ein Problem zuerst auf der Titelseite einer Zeitung vorzufinden? Stellen Sie sich vor, es gäbe keine Geheimnisse. Würden Sie sich immer noch wohlfühlen? Das ist immer die Messlatte, die wir an die Ethik der Lieferketten angelegt haben. Ich kann diese Bedenken allerdings verstehen. Wenn man bei AliExpress kauft, ist es unwahrscheinlich, dass das gekaufte Teil unter Bedingungen hergestellt wurde, auf die man stolz sein kann.

Make: Planen Sie weitere Produktionen in Europa?

Upton: Wir produzieren alle unsere Endprodukte in Europa, im Vereinigten Königreich. Und das erspart uns natürlich viele Probleme. Tatsächlich baut Sony die Produkte für uns. Sie werden also in Großbritannien gebaut und zusätzlich noch von einem internationalen Blue-Chip-Elektronikunternehmen. Das hilft uns ungemein. Wenn wir beispielsweise viele Geräte bauen, wie es im Moment der Fall ist, wir uns also in einer Art Stoßbetrieb befinden, lagern wir einen Teil der Produktion aus, aber innerhalb der Sony-Organisation. Dieser Teil der Produktion erfolgt bei Spitzenauslastung dann in Japan, aber, wie gesagt, von Sony produziert. Wir können uns auf ihre Integrität und Fähigkeit stützen, in der Lieferkette vorgelagerte ethische Standards zu gewährleisten.

Make: Ließen sich manche Komponenten auch in Europa herstellen?

„Wir haben sehr deutlich gemacht, dass es unser Ziel ist, die beste Hardware für unsere Kunden zu bauen. Wir sind nicht ideologisch, sondern wollen ein nachhaltiges Unternehmen aufbauen, das es sich leisten kann, die wirklich immensen Investitionen zu tätigen, die wir tätigen müssen."

Upton: Leiterplatten sind überraschend schwer herzustellen und eine Produktion im Vereinigten Königreich in großem Maßstab aus dem Boden zu stampfen, wäre sehr schwierig. So etwas lohnt sich vor allem bei großen Mengen. Unser Lieferant, ein Unternehmen namens Olympic in China, ist absolut riesig und ein fantastisches und sehr ethisches Unternehmen.

Könnte man Steckverbinder oder USB-Stecker im Vereinigten Königreich herstellen? Vielleicht, und wir haben das auch in Betracht gezogen. Aber ich denke, dass es besser ist, einen Partner aus dem Fernen Osten zu finden, den man überprüfen kann und mit dem man sich wohl fühlt – nicht, weil sie niedrige Arbeitskosten oder irgendeinen besonderen Vorteil haben, sondern weil sie es einfach die ganze Zeit und seit Jahren tun und die Schwachstellen aus dem Herstellungsprozess beseitigt haben.

Make: Zurzeit wird über Open-Source-Projekte diskutiert, die zu Closed-Source gewechselt sind. Auch Prusa steht mit dem MK4 in der Kritik. Wie ist Ihre Perspektive auf das Thema Open-Source?

Upton: Wir haben nie alles als Open Source angeboten. Der Raspberry Pi war nie ein offenes Hardwareprodukt. Es gibt auch keine Bestrebungen, ein offenes Hardware-Produkt zu machen. Wir haben sehr deutlich gemacht, dass es unser Ziel ist, die beste Hardware für unsere Kunden zu bauen. Wir sind nicht ideologisch, sondern wollen ein nachhaltiges Unternehmen aufbauen, das es sich leisten kann, die wirklich immensen Investitionen zu tätigen, die wir tätigen müssen. Der RP2040-Mikrocontroller ist ein 5-Millionen-Dollar-Programm. Wenn man Projekte in dieser Größenordnung durchführt, muss man sicher sein, dass sich das auch rentiert, und dass man ein nachhaltiges Geschäft hat. Und dafür ist es nicht unbedingt die beste Wahl, alle Hardware-

© Copyright by Maker Media GmbH.

„Wenn man ehrlich sagt, was man getan hat und warum man es getan hat, dann können die Leute einem widersprechen. Und wenn sie es tun, das Produkt einfach nicht kaufen. Dann hat man auch keinen Grund, sich zu beschweren."

Designs offen zu legen. Wir versuchen, so viel von unserer Software offen zu gestalten, wie wir können.

Make: Hat das auch schon einmal zu Problemen geführt?

Upton: Etwa in 2012 bestand ein erheblicher Teil unserer Softwareunterstützung für dieses Ding, das die Leute manchmal negativ als „Blob" bezeichnen, start.elf, eine Closed-Source-Komponente, die den größten Teil der Multimedia-Funktionen des Systems ausführte (und nur als Binary veröffentlicht wurde, Anm. d. Red.). Wir haben damals versprochen, uns zu kümmern. Natürlich gab es auch Skeptiker, aber wir haben über die Jahrzehnte hinweg geschafft, nach und nach Hardware-Komponenten zu nehmen, sie von der Blob-Komponente zu befreien und in offenen Code zu überführen, der auf dem ARM-Chip läuft. Die erste dieser Komponenten war natürlich die 3D-Hardware, aber jetzt sind die Scan-Out-Hardware und Elemente der Videocodec-Hardware offen.

Make: Also öffnet sich der Raspberry Pi immer mehr?

Upton: Ich denke, wir werden sehr bald an einen Punkt gelangen, an dem es keinen wirklich geschlossenen Quellcode mehr gibt. Vielleicht bleibt ein wenig Code, der die Takte und die Leistung steuert, denn dieser Code ist sehr anfällig und heikel, und wir würden ihn am liebsten einbrennen, sodass niemand sich daran zu schaffen macht. Abgesehen davon vermute ich aber, dass wir am Ende eine Plattform haben werden, die eine geschlossene Hardware, aber 100% offene Software hat.

Make: Und manchmal muss man ein Projekt auch ein wenig schützen.

Upton: Ja, wir sind pragmatisch, nicht ideologisch. Es ist immer wichtig, dass man ehrlich damit umgeht. Wenn man herumgeht und sagt, das Projekt wäre... Und ich glaube ehrlich gesagt nicht, dass Prusa das getan hat. Die Leute sind vielleicht unnötigerwei-

„Wenn man sich die Leute anschaut, die in der Technologiebranche arbeiten, dann sehen alle in jeder Hinsicht so aus wie ich. Und warum sehen sie so aus? Weil Leute wie ich es waren, die in den 1980er Jahren mit Computern herumgespielt haben."

© Copyright by Maker Media GmbH.

se ein bisschen böse auf sie. Wenn man herumgeht und sagt, ein Projekt sei Open Source, und dabei ist es das nicht, dann hat man es verdient, Vorwürfe zu erhalten. Wenn man ehrlich sagt, was man getan hat und warum man es getan hat, dann können die Leute einem widersprechen. Und wenn sie es tun, das Produkt einfach nicht kaufen. Dann hat man auch keinen Grund, sich zu beschweren.

Make: Was wünschen Sie der Maker-Community und was würden Sie gern mehr sehen?

Upton: Wenn ich hier rüberschaue, sehe ich viele junge Leute in diesem Raum, die die demografische Verteilung in der Gesellschaft besser widerspiegeln als es die erwachsene Ingenieurgemeinschaft tut. Es sind vor allem mehr Mädchen hier. Das ist ein großes Thema. Ich bin sicher, in Deutschland ist es ähnlich wie im Vereinigten Königreich: Wenn man sich die Leute anschaut, die in der Technologiebranche arbeiten, dann sehen alle in jeder Hinsicht so aus wie ich. Und warum sehen sie so aus? Weil Leute wie ich es waren, die in den 1980er Jahren mit Computern herumgespielt haben. Die demografischen Probleme von heute sind die Nachkommen anderer demografischer Probleme in den 1980er Jahren.

Make: Und das scheint sich zu ändern?

Upton: Wenn man sich die Leute hier unten anschaut, dann sieht man, dass es in 20 oder 30 Jahren sehr wahrscheinlich ist, dass unsere Ingenieursindustrie der Gesellschaft viel ähnlicher sein wird als es heute der Fall ist. Nichts gegen Leute, die aussehen wie ich, aber es wäre schön, wenn unsere Ingenieurgemeinschaft die Gesellschaft widerspiegeln würde. Man darf nicht vergessen, dass es sich hierbei um einen ständigen Prozess handelt. Es ist sehr leicht, zurückzufallen und wir müssen diesen Druck ständig aufrechterhalten, damit sich niemand ausgeschlossen fühlt. Es gibt eine Menge positiver Dinge, die man tun kann.

„Eines der Dinge, auf die wir bei unserem Code-Club-Netzwerk besonders stolz sind, ist die Tatsache, dass fast 50 Prozent der Teilnehmer, d. h. der 9- bis 13-Jährigen, Mädchen sind."

In unserem Club-Netzwerk bemühen wir uns zum Beispiel sehr darum, dass es Mentorinnen gibt, und wir arbeiten verstärkt daran, dass Schulen in wirtschaftlich benachteiligten Gegenden Clubs haben. Eines der Dinge, auf die wir bei unserem Code-Club-Netzwerk besonders stolz sind, ist die Tatsache, dass fast 50 % der Teilnehmer, d. h. der 9- bis 13-Jährigen, Mädchen sind. Ob sie eine benachteiligte Schule sind, lässt sich im Vereinigten Königreich gut daran ablesen, wie hoch der Anteil der Kinder ist, die subventionierte Schulspeisen erhalten. Betrachtet man auf dieser Grundlage eine benachteiligte Schule und eine begünstigte Schule, so hat die benachteiligte Schule aber mit größerer Wahrscheinlichkeit einen Club als die begünstigte Schule. Das sind alles großartige Trends. Ich glaube auch nicht, dass es nur ein Thema im Vereinigten Königreich ist. Und wenn man sich hier unten umschaut, dann ist das, was man sieht, so etwas wie das erste Ergebnis vieler vereinzelter Bemühungen, die Gesellschaft ein wenig stärker in dieser Gemeinschaft abzubilden.

Make: Vielen Dank für das interessante Gespräch und dass Sie sich die Zeit genommen haben.

Upton: Nun, danke, dass wir hierher kommen und Sie treffen durften. Es ist, wie gesagt, meine erste Maker Faire seit vielleicht sechs oder sieben Jahren. Und ich werde mich auf jeden Fall an sie erinnern. *—akf*

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

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

© Copyright by Maker Media GmbH.

heise Shop

Smarte Werkstattboxen

Bauteile schnell wiederfinden? Unser ESP32-gesteuerter Werkstatt-Organizer zeigt per LED an, in welcher Box ein Teil liegt. Die Abfrage erledigt eine Smartphone-App, die man mit der freien No-Code-IDE Thunkable bequem zusammenklickt.

von Bernd Heisterkamp

© Copyright by Maker Media GmbH.

Wer kennt das nicht: Man weiß genau, dass man noch diese oder jene Komponente in seiner Werkstatt herumliegen hat, aber nicht, in welcher der vielen Kästchen oder Boxen sie abgelegt ist. Natürlich kann man alles fein säuberlich beschriften, aber wer hat schon Lust, die Etiketten aktuell zu halten. Cooler ist es natürlich, die Bauteile über eine simple Smartphone-App zu verwalten und die Box, in der sich das gesuchte Teil befindet, mit einer LED aufleuchten zu lassen. Für meine Werkstatt habe ich solch ein System entwickelt und beschreibe in diesem Artikel, wie man es nachbauen kann. Links zum Projekt und dem GitHub-Repository befinden sich in der Kurzinfo.

Kurzinfo

» Einzelteile in der Werkstatt mit leuchtenden Boxen leicht wiederfinden
» In Thunkable eine Smartphone-App mit Suchfunktion programmieren
» Mit der App Befehle an einen ESP32 und NeoPixel-LEDs senden

Checkliste

Zeitaufwand:
6 bis 8 Stunden

Kosten:
ca. 50 Euro (ohne Werkzeug)

Material

» **ESP32- oder ESP8266 Development Board**
» **NeoPixel-LED-Streifen**
» **Sortierkisten**
» **Holzbrett** für die Rückseite der Sortierkiste
» **Silberdraht oder Kabel**

Dienste und Software

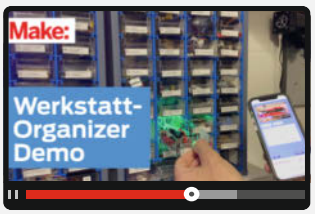
» **Google Sheets**
» **Thunkable**
» **Cloudinary**
» **Thonny**

Werkzeuge

» **Heißklebe-Pistole**
» **Seitenschneider**
» **Lötkolben**
» **Isolierband**

Mehr zum Thema

» Carsten Wartmann, Disco is back – Lichtshow mit WLED, Make 2/23, S. 8
» Alexander Alber und Thomas Schmid, Festivaltotem mit RGB-LEDs, Make 4/23, S. 58
» Florian Schäffer, Maker-Apps selber machen, Make 3/18, S. 34
» Daniel Springwald, Der Lava Frame, Make 3/23, S. 36



Alles zum Artikel im Web unter *make-magazin.de/xab7*

Geschickt verknüpft

Der Werkstatt-Organizer basiert auf einer Bauteilverwaltung, für die ich eine Google-Sheets-Tabelle verwende. Auf dieses Warenlager greift eine App zu, die in Thunkable programmiert ist, einem Ableger des MIT App Inventor mit dem man sich leicht Apps für Smartphones zusammenklicken kann. Mit unserer programmierten App kann man Boxen samt ihrer Inhalte schnell finden und verwalten (optional auch über QR-Codes). Klickt man beim Suchen in der App auf das Bild einer Box, sendet sie einen HTTP-GET-Request mit der zugehörigen Boxnummer über das lokale Netzwerk an einen ESP32. Dieser bringt daraufhin eine NeoPixel-LED hinter der Box zum Leuchten (Bild 1).

Da die einzelnen Funktionen aufeinander aufbauen und die LEDs im Projektaufbau erst zum Schluss mit dem System verknüpft werden, kann man bereits mit dem Nachbau des Werkstatt-Organizers anfangen, auch wenn die zugehörige Hardware noch fehlt.

Das virtuelle Warenlager

Los gehts mit Google Sheets. Ich empfehle, für das Projekt einen neuen Google-Account anzulegen und diesen später auch für Thunkable und den Bilderdienst Cloudinary zu verwenden, damit sich die Daten untereinander synchronisieren lassen. Die Google-Tabelle sollte folgende Spalten enthalten:

– **BoxNr**: Eine eindeutige, fortlaufende Nummer (z.B. 1, 2, 3)
– **Inhalt**: Stichwörter, die den Inhalt der Box beschreiben
– **Typ**: Hier kann man neben Boxen auch Regalbretter oder Werkzeuge klassifizieren. So findet man Teile über die App auch dann, wenn noch keine LEDs installiert sind.
– **Bereich**: Wenn man nicht nur ein paar Sortierkisten, sondern gleich die ganze Werkstatt organisieren möchte, empfiehlt sich eine Bereichseinteilung mit Buchstaben oder ganz militärisch (z.B. Bohrmaschine auf 9 Uhr).
– **Foto**: Speichert die URL zu einem Foto

In der Tabelle sollten mindestens 2 Zeilen mit Beispieldaten gefüllt werden, damit man die App vernünftig gestalten kann (Bild 2). Im Foto-Feld reicht erstmal ein Link zu einer beliebigen JPG-Datei im Internet.

Zur App ohne Code

Jetzt geht es bereits an die Programmierung der Smartphone-App. Diese erstelle ich in Thunkable, einer visuellen Programmierumgebung, die Scratch sehr ähnlich ist und sich leicht erlernen lässt. Dafür lege ich mir unter x.thunkable.com einen Account an und wähle „Anmeldung über Google" (Sign in with Google) aus. Dadurch kann ich in Thunkable auf die zuvor erstellte Google-Tabelle zugreifen.

Die Benutzeroberfläche von Thunkable ist übersichtlich aufgebaut und unterteilt sich in zwei Modi: Design und Blocks. Wenn man ein neues Projekt startet, beginnt man standardmäßig damit, die Benutzeroberfläche mithilfe von vorgefertigten Komponenten zu erstellen, die am linken Bildschirmrand aufgelistet sind. Von dort zieht man die benötigten Elemente auf das leere Layout (Screen) in der Mitte des Bildschirms und definiert ihre Eigenschaften im Menü auf der rechten Seite.

Im Modus Blocks lassen sich die eingefügten Komponenten dann mithilfe von unterschiedlich gefärbten Bausteinen programmieren – ohne auch nur eine Zeile Code zu schreiben.

Die Projekte werden alle auf dem Thunkable-Server gespeichert und sind mit einem kostenlosen Account für jeden sichtbar (public). Je nachdem, was man programmiert, sollte man also möglichst keine sensiblen Daten in seinen Projekten hinterlegen, auch wenn Verknüpfungen zu anderen Accounts (z.B. Google Sheets) nicht für andere einsehbar sind und gekappt werden, sobald jemand ein Projekt dupliziert. Das ist praktisch, denn sonst könnte jeder mein Warenlager durcheinanderbringen.

© Copyright by Maker Media GmbH.

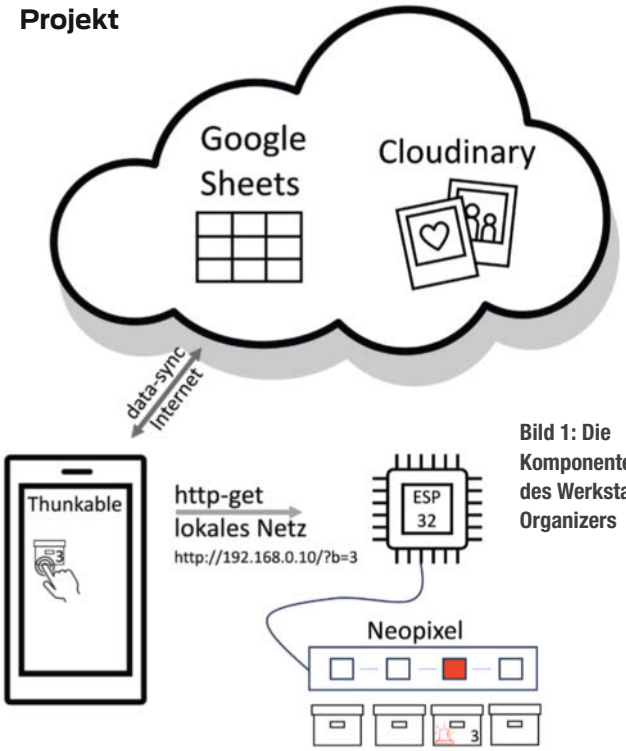


Bild 1: Die Komponenten des Werkstatt-Organizers

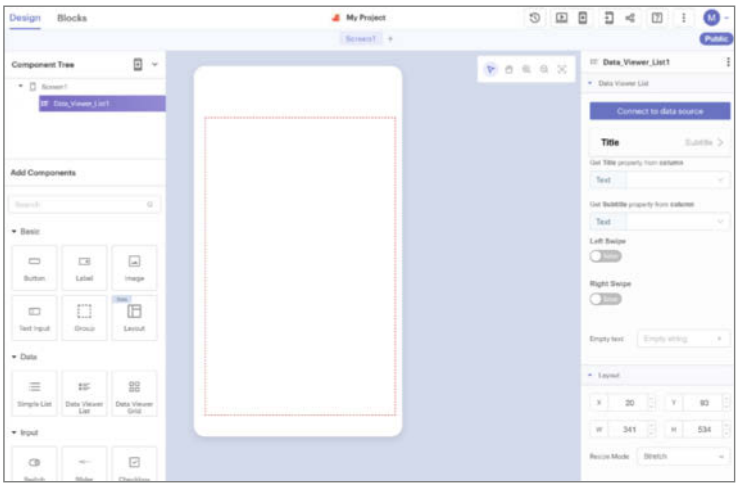


Bild 3: Nachdem man die „Data Viewer List“ auf Screen 1 gezogen hat, benötigt sie noch Inhalte.

Wer mehr Features oder Privatsphäre benötigt, muss dafür mindestens 15 Euro im Monat bezahlen.

Da Thunkable im Browser läuft, kann man seine Apps auch in diesem testen. Dazu klickt man auf das Web-Preview-Icon, das sich rechts oben im Menü befindet. Um die App gleich auf dem Smartphone-Bildschirm auszuprobieren, kann man sich die „Thunkable Live“-App aus dem jeweiligen Store seines Smartphones (iOS/Android) installieren. Diese benötigt man auch für die spätere Kommunikation mit dem ESP, da man über die Web-Preview keine lokalen HTTP-Requests senden kann.

Benutzerführung schnell erstellt

Nachdem der Thunkable-Account eingerichtet ist, öffnet sich automatisch ein neues Projekt, ansonsten klickt man auf „Create New App“. Im Design-Modus gestalte ich als Erstes den Hauptbildschirm (Screen 1) meiner App, über den man die Boxen auf unterschiedliche Weise auswählt oder neue hinzufügt. Um die Google-Tabelle bzw. die Boxen als Liste in das Layout einzubinden, ziehe ich die Komponente „Data Viewer List“ vom linken Bildschirmrand (Add Component) auf mein leeres Layout in der Mitte (Bild 3). Auf der rechten Seite klicke ich anschließend auf die blaue Schaltfläche „Connect to data source“, wähle über „Create New/Google Sheets/Select a Sheet“ die vorbereitete Tabelle aus und füge sie mit einem Klick auf Create hinzu.

Danach ändere ich die Darstellungsart der Listenelemente und ordne Bild, Titel, Untertitel und Text wie in Bild 4 den Tabellenspalten zu. Jede der aufgelisteten Boxen ist anwählbar und führt später zu einer detaillierteren Ansicht (Screen 2), in der man ihren Inhalt bearbeiten kann. Um auch neue Boxen hinzufügen zu können, platziere ich am linken oberen Bildschirmrand von Screen 1 noch eine Button-Komponente und ändere ihr Label in ein Pluszeichen (Bild 5).

Weiter geht es mit Screen 2, auf dem man die Daten der Boxen bearbeiten kann. Um ihn zu erstellen, klicke ich oben in der Mitte des Bildschirms neben dem Titel Screen 1 auf das Pluszeichen. In dem neuen Layout platziere ich anschließend ein paar Komponenten, die ich mit den Tabellendaten verknüpfen werde. Bild 6 zeigt die Felder, die der neue Screen enthalten sollte.

Ganz oben befindet sich eine Image-Komponente, die das Foto des Boxinhalts anzeigt. Darunter sitzen ein paar Label, die mit BoxNr, Typ, Bereich und Inhalt beschriftet sind. Ihr zugehöriger Tabellenwert wird je nach Box in den übrigen Feldern angezeigt. Die grün markierten findet man unter den Komponenten als „Text Input“. In diesen kann man später die Attribute der ausgewählten Box ändern. Für das Eingabefeld unterhalb des Inhalt-Labels bietet es sich an, die Option „Multiline“ für einen Zeilenumbruch bei längeren Beschreibungen auszuwählen. Unterhalb der Textfelder platziere ich die Buttons Cancel, OK und „New Pic“, auf deren Funktion ich im nächsten Abschnitt näher eingehe.

Wenn man die Komponenten im Design-Modus vernünftig benennt (z.B. bttNew oder In4_Inhalt), findet man sie später leichter beim Programmieren wieder. Dazu klickt man rechts neben dem Komponenten-Namen auf die drei Punkte.

	A	B	C	D	E
1	BoxNr	Inhalt	Typ	Bereich	Foto
2	1	Nägel Schrauben	Box	9 Uhr	http://res.cloudinary.com/di3l2aoy7/ima
3	2	Kleber	Box	12 Uhr	http://res.cloudinary.com/di3l2aoy7/ima
4	3	A4 Papier	Regal	15 Uhr	http://res.cloudinary.com/di3l2aoy7/ima

Bild 2: Die Tabelle in Google Sheets verwaltet das Inventar.

Eigenständige Thunkable-Apps

Android-Nutzer können sich mit dem kostenfreien Thunkable-Account eine Installationdatei ihrer App herunterladen, iOS-Nutzer müssen sich entweder mit Thunkable Live zufriedengeben (was gut funktioniert) oder auf einen Thunkable-Pro-Account upgraden, über den man die App z.B. auch im Apple App Store veröffentlichen kann. Dieser Account kostet allerdings mindestens 38 US-Dollar pro Monat und man benötigt eine zusätzliche Apple-Entwicklerlizenz.

© Copyright by Maker Media GmbH.

Visuell programmieren

Bisher habe ich nur in Thunkables Design-Modus gearbeitet und den Komponenten noch keine Funktion zugewiesen. Diese ergänze ich nun mit Code, indem ich Screen 1 und dann den Modus Blocks anklicke.

Daraufhin ändert sich die Anzeige und der Bildschirm teilt sich in zwei Hälften, mit den Programmierbefehlen auf der linken und der Programmierfläche auf der rechten Seite. Unter „UI-Components" findet man spezifische Befehle für die Komponenten, die man zuvor im Design-Modus hinzugefügt hat. Zusätzlich gibt es eine Reihe von Grundfunktionen (Core) und „App Features", unter denen man etwa „Data Sources" findet, um mit der erstellten Tabelle zu interagieren, sowie die Kategorie „Advanced". Wer mein Programm nachbaut, kann sich an den Farben der Blöcke orientieren. Hat man noch nie mit Thunkable gearbeitet, empfehle ich, sich im Vorfeld ein paar Einführungsvideos anzuschauen (siehe Link in der Kurzinfo).

Bild 5: Über den Plus-Button kann man direkt in der App neue Boxen zum Organizer hinzufügen.

Der in Bild 7 abgebildete Code für Screen 1 bewirkt, dass die Zeilennummer einer Box, die man anwählt, in der Variablen `int_SelectedRow` gespeichert wird. Zudem setzt das Programm den Wert der Variablen `bol_EditBox` auf `true` und merkt sich, dass der Eintrag editiert werden soll. Anschließend navigiert das Pro-

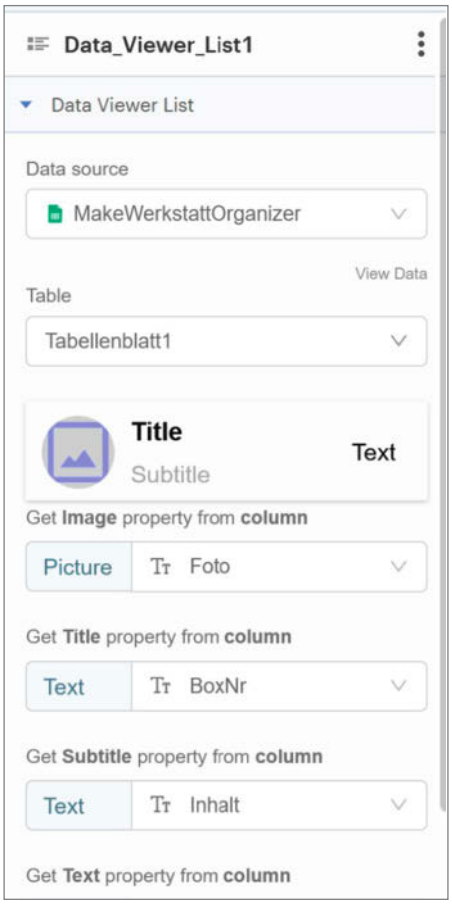


Bild 4: Ist die Google-Tabelle importiert, verknüpft man sie mit den Platzhaltern der „Data Viewer List".

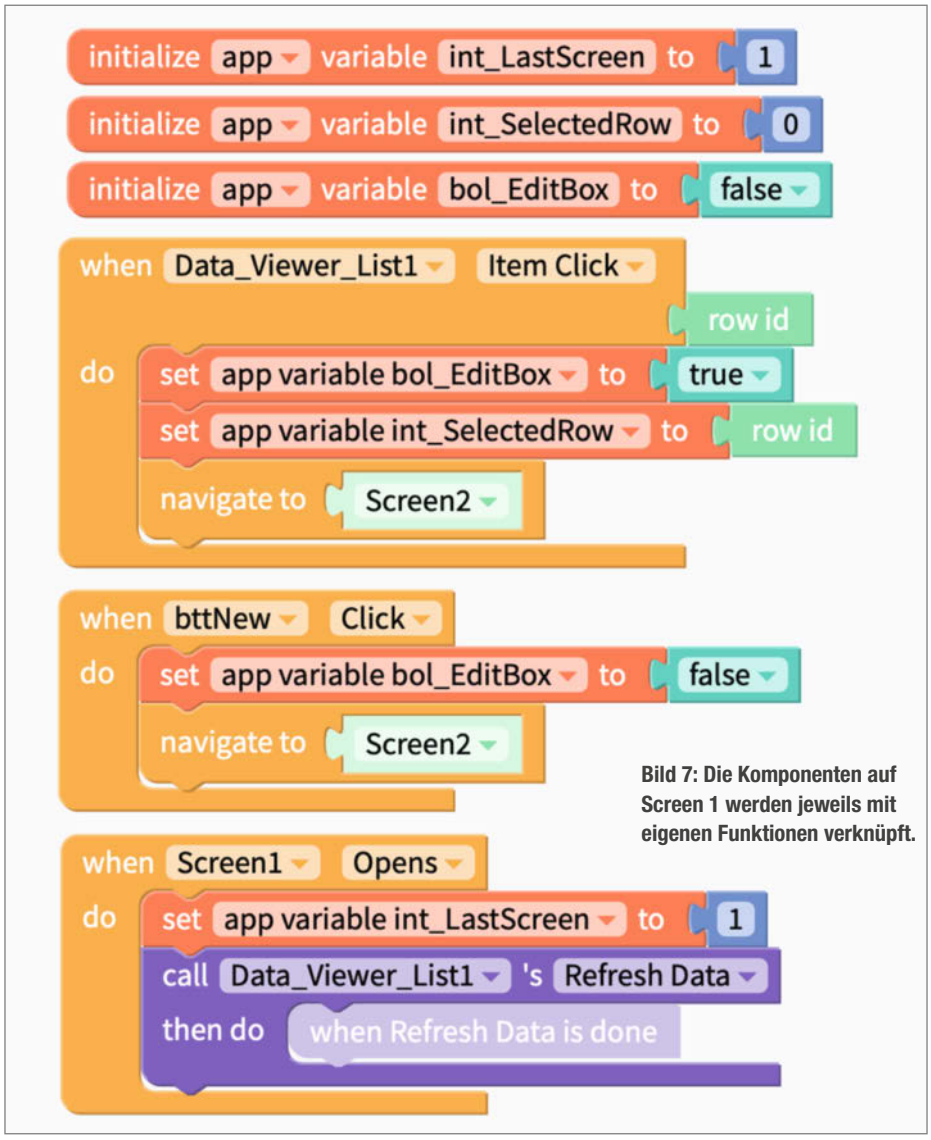


Bild 7: Die Komponenten auf Screen 1 werden jeweils mit eigenen Funktionen verknüpft.

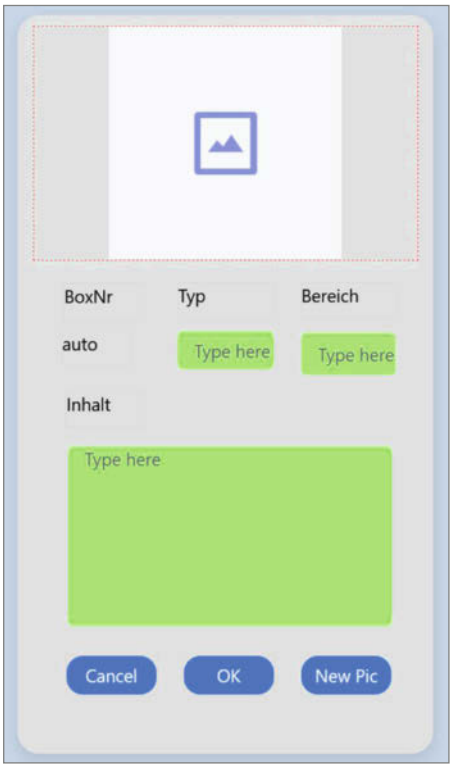


Bild 6: Das Layout von Screen 2: Hier kann man später den Inhalt der Boxen anpassen.

© Copyright by Maker Media GmbH.

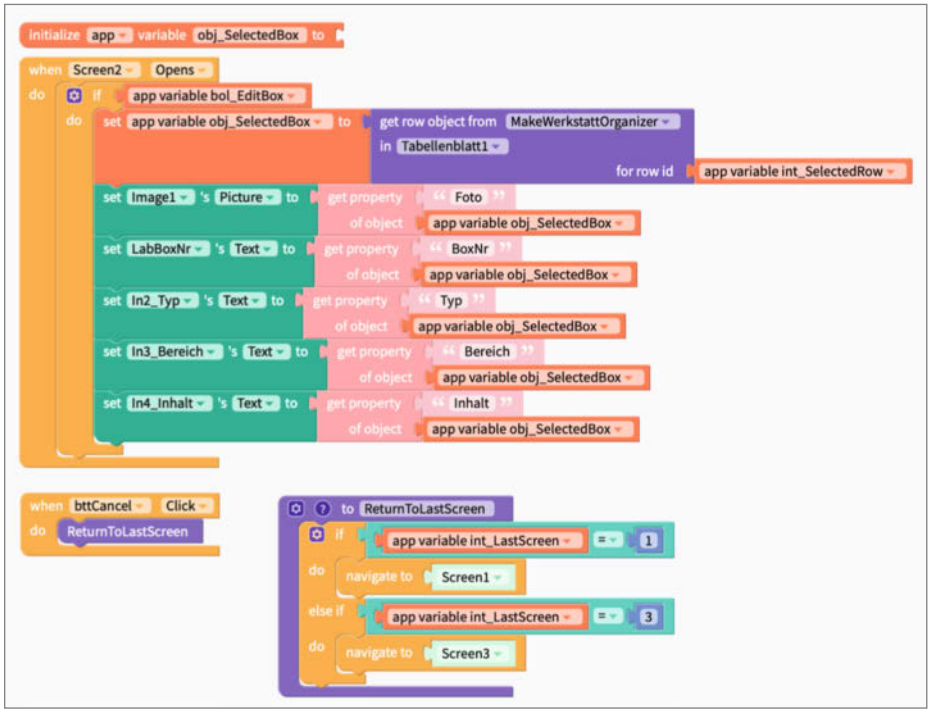


Bild 8: Der obere Block weist die Tabellendaten den Textfeldern auf Screen 2 zu.

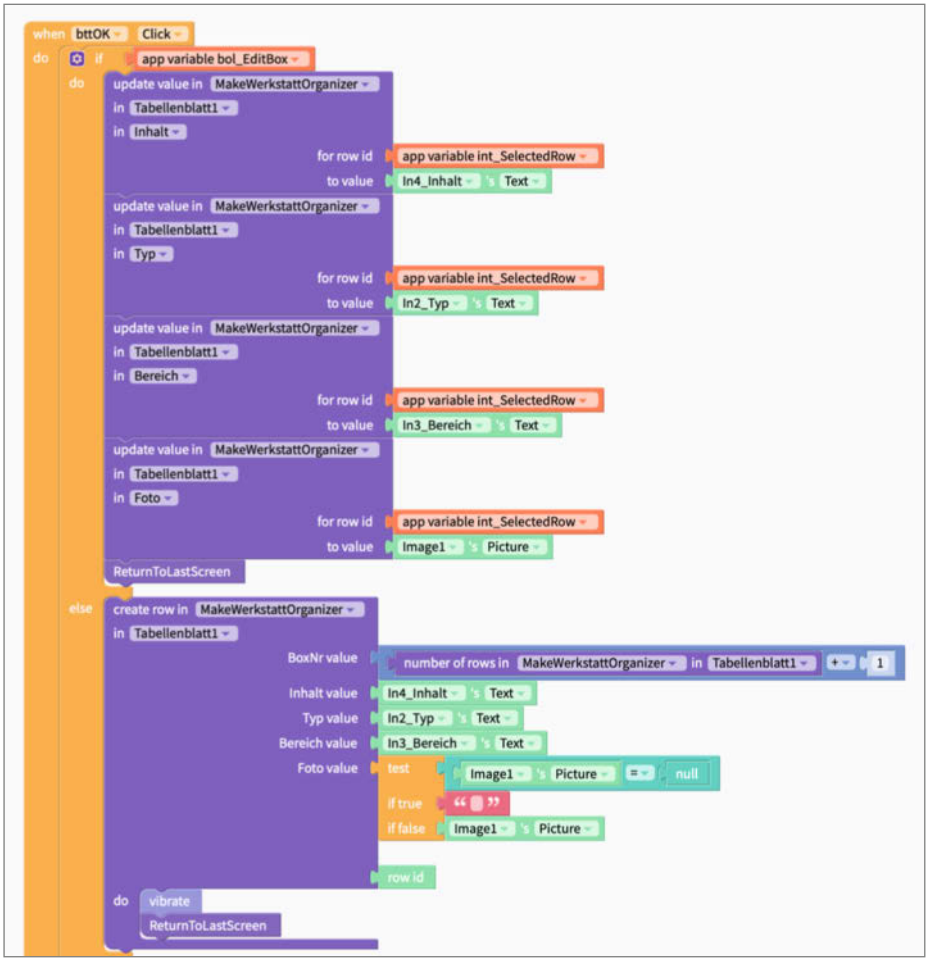


Bild 9: Mit dem OK-Button auf Screen 2 speichert man seine Änderungen oder erstellt eine neue Box.

gramm zu Screen 2. Hat der Nutzer stattdessen den Button mit dem Pluszeichen gedrückt, ist der Wert der Variablen `bol_EditBox` gleich `false` und die Eingabefelder bleiben leer. Damit die Boxen-Liste auf Screen 1 aktuell bleibt, wird sie jedes Mal neu geladen, sobald man ihn öffnet.

Im Screen 2 fügt man nun im Blocks-Modus den Code aus Bild 8 ein. Dieser bewirkt, dass beim Öffnen von Screen 2 die einzelnen Labels und Eingabefelder mit den Tabellendaten der ausgewählten Box gefüllt werden. Dazu liest das Programm die Werte aus der Google-Tabelle in das Objekt `obj_SelectedBox` und weist sie danach den Textfeldern zu.

Klickt der Nutzer auf den Cancel-Button, gelangt er über die Funktion `ReturnToLastScreen`, die man selbst erstellen muss, direkt wieder zum Screen 1 zurück. Mit OK speichert man die bearbeiteten Eingabefelder (Bild 9). Der Befehl `update value in …` aktualisiert dabei die Zeile der Google-Tabelle, die in der Variablen `int_SelectedRow` gespeichert ist, mit dem geänderten Inhalt der Textfelder.

Wenn `bol_EditBox` gleich `false` ist, es sich also um einen neuen Eintrag handelt, erstellt die Funktion `create row in …` eine neue Zeile in der Google-Tabelle und füllt sie mit den eingetragenen Inhalten. Damit dies funktioniert, müssen alle leeren Zeilen gelöscht werden.

Eigene Fotos verwenden

Um Platzhalterbilder in der Tabelle durch eigene zu ersetzen, kann man diese bei dem Web-Dienst Cloudinary speichern, dessen API bereits Teil von Thunkable ist. Für den kostenlosen Cloudinary-Account verwende ich erneut den zuvor erstellten Google-Account.

Um Thunkable mit Cloudinary zu verbinden, wählt man im Design-Modus auf der linken Seite das Settings-Icon aus und gibt beim Eintrag „Cloudinary Settings“ die per-

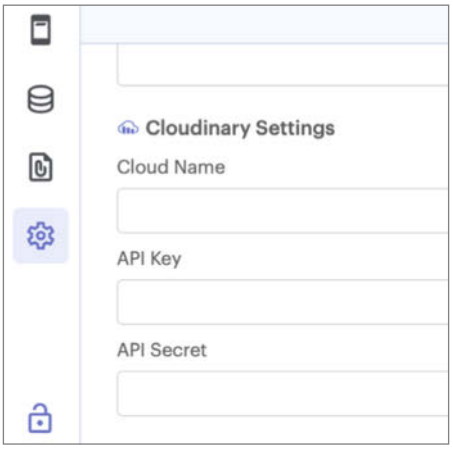


Bild 10: Über das Zahnradsymbol gelangt man zum Eingabefeld für die Cloudinary API-Schlüssel.

© Copyright by Maker Media GmbH.

Bild 11: Bilder kann man auf Screen 2 direkt von seinem Smartphone einbinden.

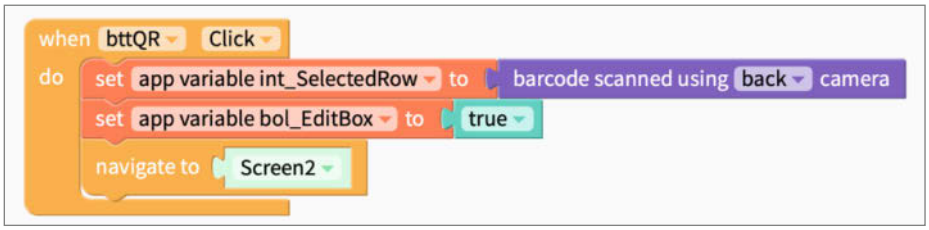


Bild 12: Der hinzugefügte Scan-Button auf Screen 1 ist leicht programmiert.

sönlichen Daten für „Cloud Name", „API Key", und „API Secret" ein (Bild 10). Diese Informationen findet man in Cloudinary, wenn man dort nach dem Login auf das Menü „Dashboard" wechselt.

Der Code für Screen 2, mit dem man ein Bild aus der Smartphone-Bibliothek auswählen, skalieren, in die Cloud hochladen und in der App darstellen kann, ist in Bild 11 gezeigt. Als Ergebnis hinterlegt das Programm die von Cloudinary erhaltene URL des hochgeladenen Bildes im Feld `Image1`. Sobald man mit OK bestätigt, speichert die App diese URL genau wie die Daten der anderen Felder.

Alternativ zum Befehl `Photo from photo library` könnte man auch `Photo from camera` verwenden. Dann hat man jedoch keine Möglichkeit, das Bild vor dem Abspeichern manuell zuzuschneiden und ggf. zu rotieren.

Boxen mit QR-Codes verwalten

Jede Box erhält einen eigenen QR-Code, hinter dem sich ihre Boxnummer verbirgt. Damit man diesen mit der Thunkable-App einscannen kann, um zu Screen 2 der jeweiligen Box zu gelangen, platziere ich einen Button mit der Beschriftung „Scan" auf Screen 1. Die Scan-Funktion zu programmieren ist im Blocks-Modus denkbar einfach, wie Bild 12 zeigt. Drückt man den Scan-Button auf Screen 1, liest

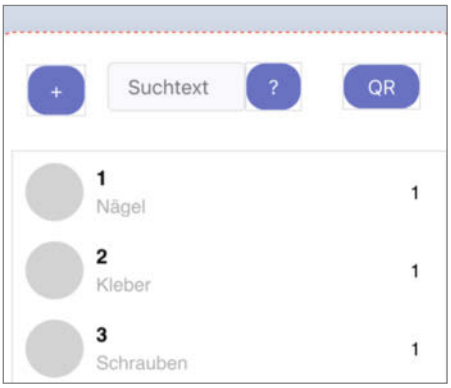


Bild 13: Das Suchfeld samt Button hat noch genug Platz zwischen den anderen beiden.

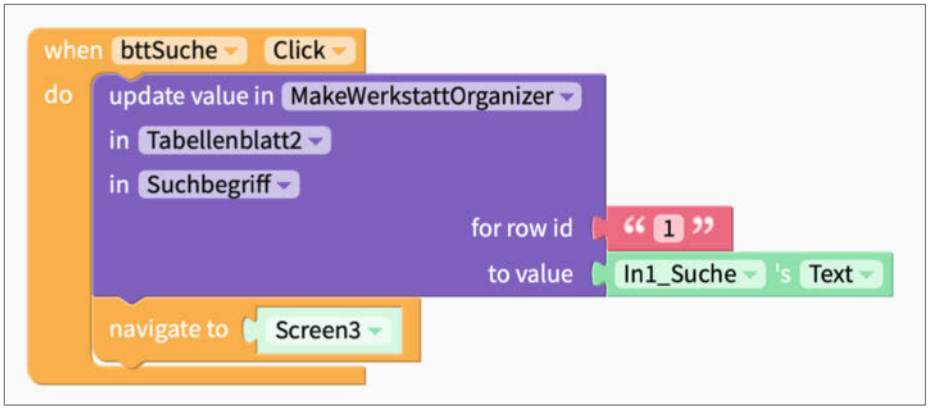


Bild 14: Die Such-Funktion auf Screen 1 wechselt zu Screen 3.

OXOCARD CONNECT
PLUG-AND-PLAY ELEKTRONIK
Einstecken und sofort ausprobieren
Steck eine Cartridge in die OXOCARD CONNECT und sofort startet dein Code
Leistungsfähige browserbasierte IDE mit über 100 Beispielen und komplettem Source-Code
NEU
OXOCARD INNOVATOR KIT
Kostenloser Elektronik- und Programmierkurs mit 15 Experimenten (14–99J)
Jetzt im heise Shop bestellen
In der Schweiz bei Galaxus oder Brack
www.oxocard.ch

© Copyright by Maker Media GmbH.

B2

=FILTER(Tabellenblatt1!A:E, (ISTZAHL(Tabellenblatt1!A:A)) * (ISTZAHL(SUCHEN(KLEIN(Tabellenblatt2!A2), KLEIN(Tabellenblatt1!B:B)))))

	A	B	C	D	E	F
1	Suchbegriff	BoxNr	Inhalt	Typ	Bereich	Foto
2	kle	2	Kleber	Box	12Uhr	http://res.cloudinary.com/di3l2aoy7/imag
3		5	Kleister	Box	12Uhr	http://res.cloudinary.com/di3l2aoy7/imag
4						
5						

Bild 15: Diese Suchformel in Tabellenblatt2 nutzt den in der App eingegebenen Suchbegriff.

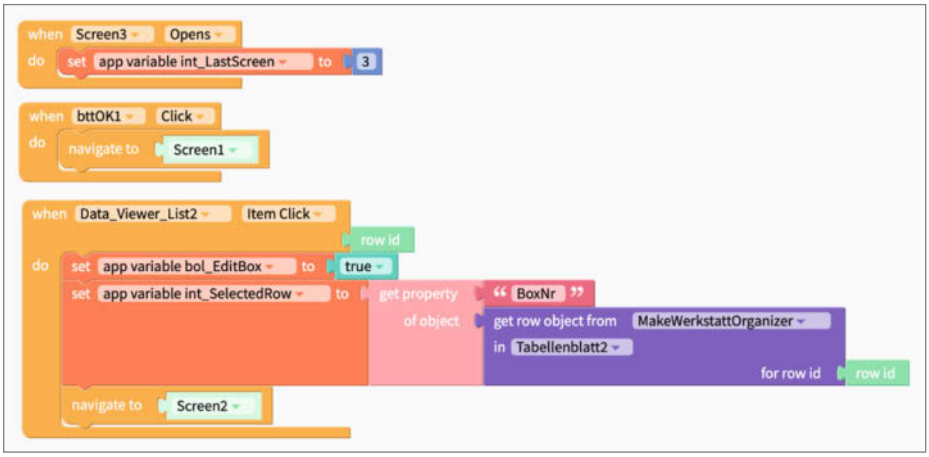


Bild 16: Auf Screen 3 wird die gefilterte Liste von Tabellenblatt2 angezeigt.

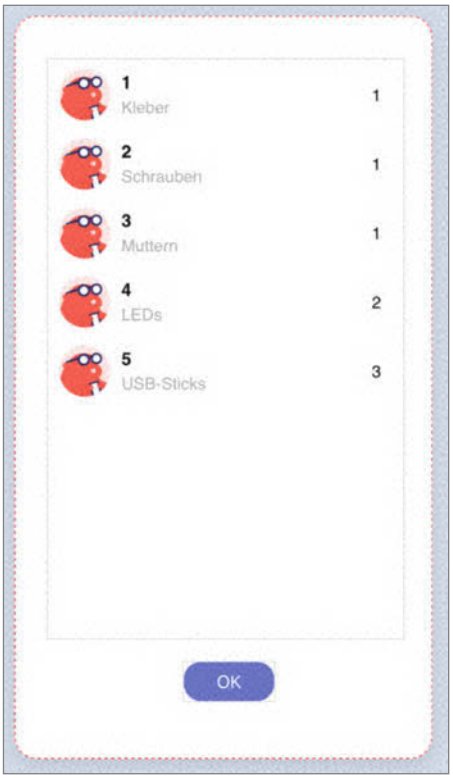


Bild 17: Mit dem OK-Button kommt man von Screen 3 zu Screen 1 zurück.

die rückseitige Kamera des Smartphones den anvisierten QR-Code und speichert die Zahl, die der Tabellenzeile bzw. Boxnummer entspricht, in der Variable `int_SelectedRow`. Danach setzt das Programm `bol_EditBox` auf `true` und wechselt zu Screen 2, in dem man die Werte der Box bearbeiten kann.

Effizienter suchen

Als Nächstes benötigt die App eine Suchfunktion, für die ich ein Eingabefeld und einen Suchen-Button mit einem Fragezeichensymbol auf Screen 1 hinzufüge (Bild 13).

Damit man in der App effizienter Teile finden kann, ohne jedes Mal ihren vollständigen Namen eingeben zu müssen, erweitere ich die Google-Tabelle um ein weiteres Blatt (Tabellenblatt2). Den Suchbegriff sende ich aus der Thunkable-App an dessen Feld A2 und nutze ihn als Filter. Dafür ergänze ich zunächst den Code für

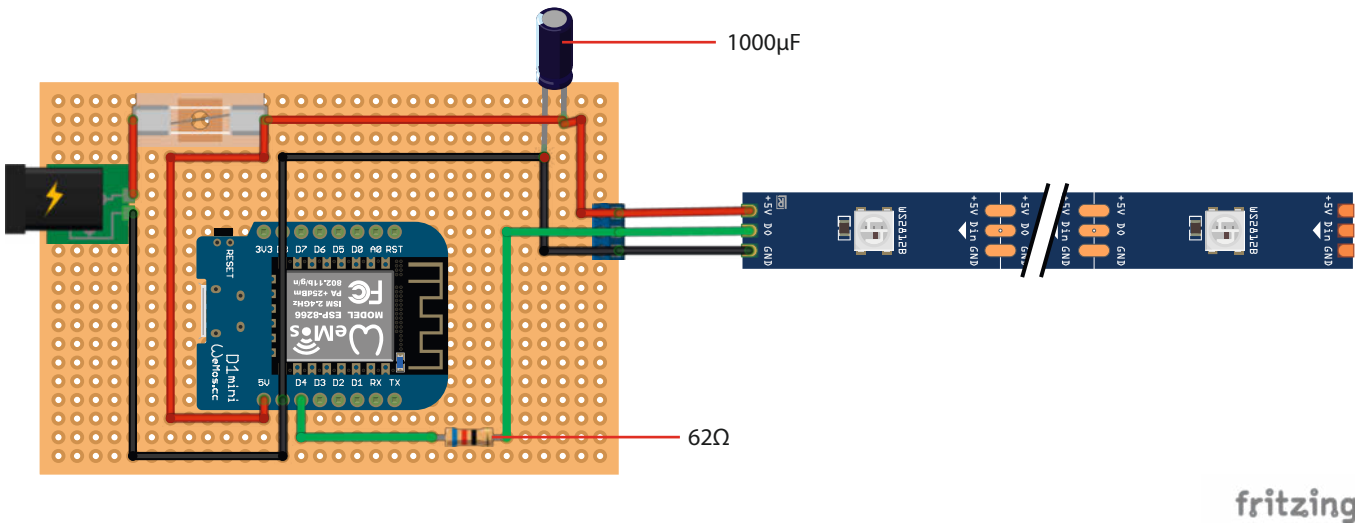


Bild 18: Die NeoPixel-Schaltung mit einem ESP8266

© Copyright by Maker Media GmbH.

Screen 1 um den Suchbefehl in Bild 14. In das Feld B2 des Tabellenblatt2 füge ich die in Bild 15 gezeigte Funktion ein. Sie listet alle Zeilen der „Inhalt"-Spalte von Tabellenblatt1 auf, in denen der Suchbegriff enthalten ist und deren „BoxNr"-Spalte eine Zahl enthält (ansonsten könnte auch die Titelzeile gefunden werden).

Um die Suchergebnisse darzustellen, füge ich in Thunkable einen neuen Screen (Screen 3) mit einer Data-Viewer-List hinzu. Als Data-Source wähle ich diesmal das gefilterte Tabellenblatt2 aus. Erscheint dieses noch nicht im Auswahlmenü, muss man die Google-Tabelle über das Data-Icon auswählen, das im Design-Modus oben links zu finden ist, und synchronisieren.

Unten platziere ich einen OK-Button, mit dem man wieder auf Screen 1 zurückkommt. Das fertige Programm für Screen 3 zeigt Bild 16, das zugehörige Layout Bild 17.

```
main.py (Ausschnitt)

import machine
import neopixel
import ure
import urequests
import socket
import time

# Konfiguration der den Boxen zugehörigen Neopixel
NEOPIXEL_CONFIG = {
    1: [0, 1],
    2: [2, 3],
    3: [4, 5],
    4: [6, 7],
    5: [8, 9],
    # Add more groups as needed
}

PIXEL_PIN = machine.Pin(2, machine.Pin.OUT)
NUM_PIXELS = 10
```

Es werde Licht

Nachdem das Grundgerüst der App steht und man die Boxen verwalten kann, baue ich als Nächstes die LED-Schaltung auf. Diese entspricht der in der Make 02/23 vorgestellten NeoPixel-Steuerung (Link siehe Kurzinfo). Das Schema ist in Bild 18 dargestellt. Da immer nur wenige Pixel auf einmal leuchten, ist der Strombedarf sehr gering. In meiner ersten Version habe ich die NeoPixel daher ohne eine externe Stromversorgung direkt an einen Ausgang des ESP32 angeschlossen und die Stromversorgung auf den USB-Pin gelegt. Das funktioniert, sofern man auch bei den Testaufbauten darauf achtet, nicht mehr als zwei bis drei NeoPixel gleichzeitig zu betreiben. Bei einer größeren Belastung kann der ESP32 nämlich schnell abrauchen.

Um die LEDs auf die Sortierboxen zu verteilen, habe ich die NeoPixel in Reihen auf ein Brett aufgeklebt und miteinander verlötet. Bild 19 zeigt eine Version mit einem ESP8266: Die NeoPixel-Streifen sind mit einem unisolierten Silberdraht verbunden und anschließend mit Isolierband abgedeckt. Hierbei muss man unbedingt darauf achten, dass man immer einen Dateneingang an einen Datenausgang anschließt, die Pfeile also durchgängig sind. Nachdem man die Funktion ausgiebig getestet hat, kann man das Brett einfach auf die Rückseite der Sortierkisten schrauben. Es ist nicht weiter tragisch, wenn einige LEDs genau zwischen zwei Boxen positioniert sind. Diese lässt man später in der Programmkonfiguration einfach außen vor.

Für den Anschluss weiterer Boxen empfiehlt es sich, einen entsprechenden Stecker am Ende der NeoPixel-Reihen vorzusehen. Wer anstelle kleiner Sortierboxen große Schachteln und Kisten auf einem Regal finden möchte, klebt die LEDs einfach vorne an den Regalboden. Mit unterschiedlichen Farben findet man so auch übereinander gestapelte Kisten.

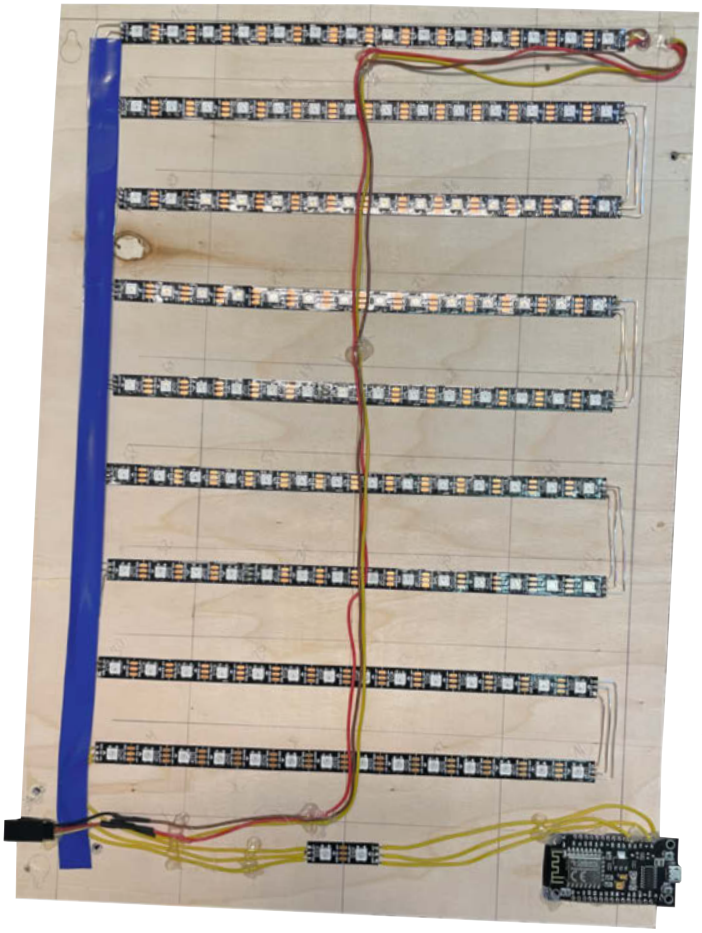


Bild 19: Dieses Brett lässt sich von hinten an die Sortierkiste montieren.

Wenn man mehrere Boxen und Regale beleuchten möchte, sollte man allerdings eine externe Stromversorgung nutzen. Alternativ sind auch mehrere kleine Systeme vorstellbar. Dazu muss man den Code der Thunkable-App nur so anpassen, dass je nach Boxnummer unterschiedliche ESPs angesprochen werden.

ESP32 einrichten

Der ESP32 steuert die NeoPixel-LEDs mit einem MicroPython-Programm (Link siehe Kurzinfo). Ich habe für dieses Projekt MicroPython gewählt, weil sich das Programm damit besonders leicht anpassen lässt, wenn der Werkstatt-Organizer geändert oder erweitert wird. Der im Listing gezeigte Ausschnitt des Codes zeigt die verwendeten Module sowie die Konfigurationszeilen, die individuell angepasst werden sollten. Am wichtigsten ist die Matrix `NEOPIXEL_CONFIG`. Hier wird je Box konfiguriert, welche Neo-Pixel leuchten sollen. In meinem Beispiel sind dies für Box 4 die NeoPixel 6 und 7.

Eine ausführliche Anleitung, wie man MicroPython mit dem ESP32 verwendet, würde den Rahmen dieses Artikels sprengen. Im Folgenden beschreibe ich aber die wichtigsten Schritte, mit denen man den Mikrocontroller für dieses Projekt einrichtet. Zudem sind weitere Anleitungen über einen Link in der Kurzinfo zu finden.

1. MicroPython-Firmware mit dem Python-Programm esptool.py auf einen ESP32 flashen (siehe Link in der Kurzinfo).
2. Eine Micropython-IDE auf dem Computer installieren. Für dieses Projekt habe ich Thonny verwendet.

© Copyright by Maker Media GmbH.

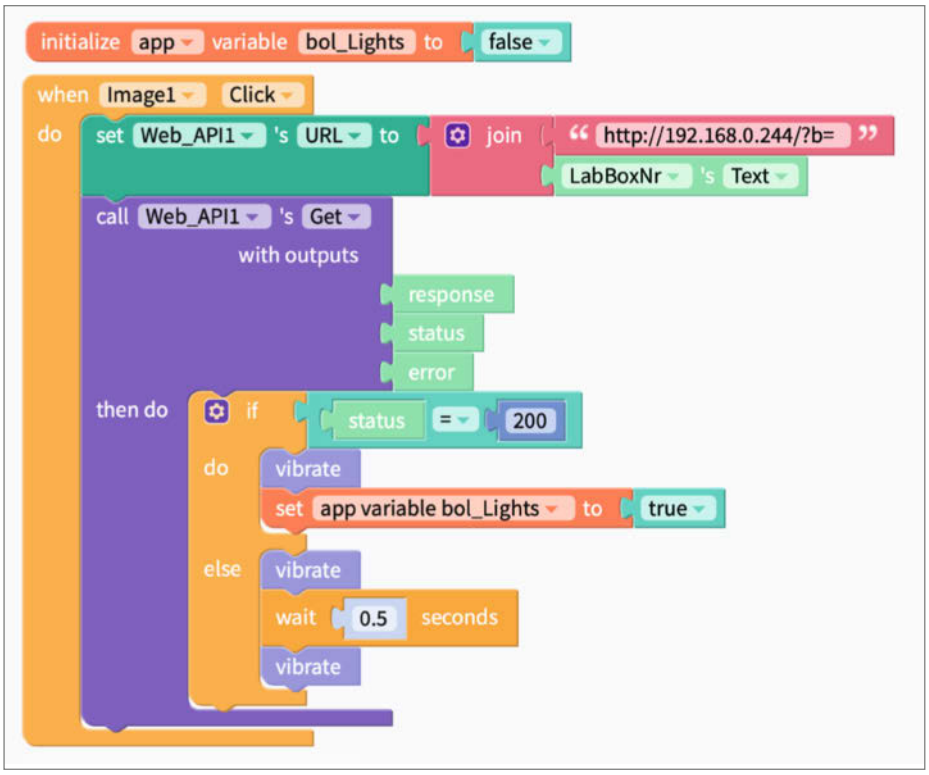


Bild 21: Der HTTP-GET-Befehl auf Screen 2 lässt die LEDs auf dem ESP32 leuchten.

3. Benötigte Module (siehe Listing) über das Thonny-Menü „Tools/Manage Packages" installieren.
4. Die 3 Python-Programme boot.py, secrets.py und main.py vom GitHub-Repository auf den ESP32 kopieren (Bild 20).
5. Netzwerk-SSID(s) und Passwörter in secrets.py anpassen.
6. Die im Listing gezeigte LED-Konfiguration für das eigene Projekt anpassen.

Nach jedem Neustart führt der ESP32 das Python-Programm boot.py einmal aus. Es verwendet die Zugangsdaten aus der Datei secrets.py, um sich mit dem drahtlosen Netzwerk zu verbinden. Anschließend lädt automatisch das Hauptprogramm main.py und startet einen einfachen Webserver auf dem Mikrocontroller. Mit dessen Hilfe kann der ESP auf eingehende Befehle der Thunkable-App reagieren.

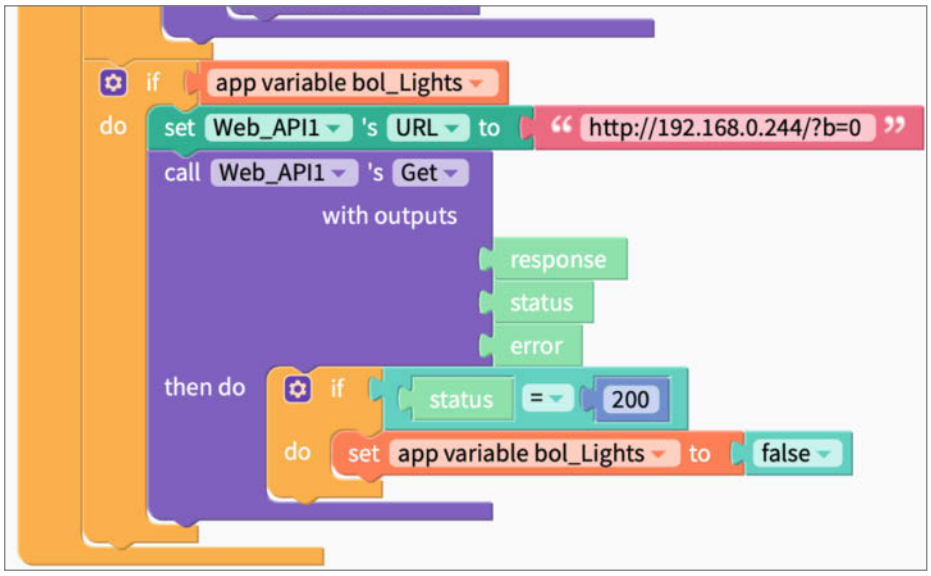


Bild 22: Dieser Code muss beim OK-Button in Screen 2 hinten angefügt werden.

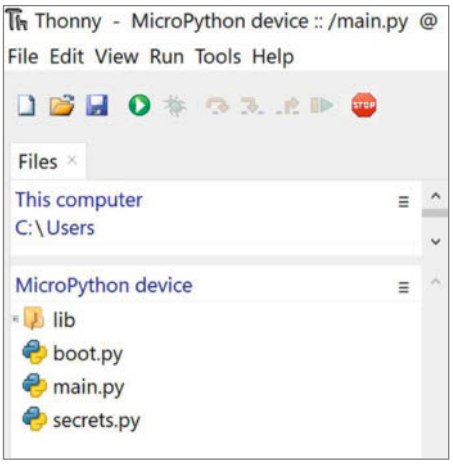


Bild 20: Nach dem Kopieren sollte die Dateistruktur auf dem ESP32 aussehen wie im GitHub-Repository des Projekts.

Befehle an den ESP32

Jetzt fehlt nur noch das Interface der Thunkable-App zum ESP32 und den NeoPixeln. Die ausgewählte Box soll schließlich leuchten, wenn man in der App auf das zugehörige Foto tippt. Hierzu füge ich in Thunkable im Blocks-Modus eine Web API hinzu, indem ich links in der Kategorie Advanced auf das Plussymbol neben „Web APIs" klicke. Mit ihrer Hilfe kann die App mit dem ESP32 kommunizieren und abgleichen, welche LEDs aufleuchten sollen. Dazu ergänze ich im Screen 2 den Code aus Bild 21. Dieser sendet mithilfe eines HTTP-GET-Befehls die Boxnummer, die in `LabBoxNr` gespeichert ist, als Parameter `b` (für Box) an die IP-Adresse des ESP32. Der darauf laufende Webserver extrahiert `b` aus der Anfrage, reagiert mit den entsprechenden LEDs und sendet eine Rückmeldung an die Thunkable-App. Konnte das Smartphone diese ordentlich empfangen (`Status = 200`), dann vibriert es einmal kurz. Bei jeder anderen Antwort vibriert es zweimal.

Drückt der Nutzer danach im Screen 2 den OK-Button, um zurück zum vorhergehenden Bildschirm zu gelangen, schalten sich die NeoPixel wieder aus. Dafür sendet die App den Parameter `b` mit dem Wert `0` an den ESP32. Den dafür zuständigen Befehl muss man im Code aus Bild 9 ergänzen. Hierzu fügt man die Zeilen aus Bild 22 hinter dem ersten `if`-Befehl ein.

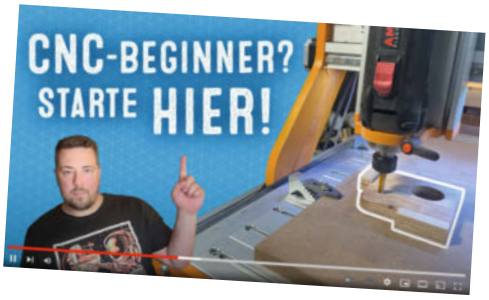
Geschafft

Damit ist der Werkstatt-Organizer fertiggestellt. Mithilfe des Smartphones lassen sich Einzelteile in der Werkstatt jetzt nicht nur schneller finden, sondern auch komfortabel verwalten. Für Erweiterungen, insbesondere bei der Thunkable-App gibt es noch viele Möglichkeiten, aber hier sollte jeder Maker selbst aktiv werden. Ich wünsche viel Spaß beim Nachbauen und erweitern. *—akf*

© Copyright by Maker Media GmbH.

Make-Tutorials auf Youtube: CNC-Fräsen für Einsteiger, Docker installieren, klebrigen Kunststoff reinigen

Wie der Einstieg ins CNC-Fräsen gelingt, zeigt unser Videotutorial auf dem Youtube-Kanal des Make Magazins anhand eines einfachen Beispiels. In der dreiteiligen Serie erstellen wir eine Radiusschablone für die Oberfräse, die gezeigten Schritte sind aber auf komplexere Fräsaufgaben übertragbar. Am Anfang steht eine digitale Zeichnung, für die Radiusschablone reicht eine 2D-Skizze der gewünschten Konturen. Im ersten Teil erstellen wir diese in Inkscape, das durch ein paar Voreinstellungen zum maßstabsgerechten Zeichenwerkzeug wird. Im zweiten Teil konvertieren wir die 2D-Zeichnung in ein Fräsprogramm. Dabei kommt das kostengünstige Estlcam zum Einsatz. Hier wird für jede Kontur festgelegt, mit welchem Fräser wie schnell und tief gefräst wird - und ob der Fräser nur außer- oder innerhalb der Kontur entlangfährt oder eine Tasche entstehen soll. In Teil drei der Serie wird die Schablone aus Multiplex an einer Mittelklasse-CNC-Fräse ausgefräst. Wer sich bisher nicht ans CNC-Fräsen getraut hat, findet hier einen einfachen Einstieg, der auch auf andere Programme und Fräsen übertragbar ist und einen Einblick in den Arbeitsablauf gibt.



CNC-Berührungsängste? Nicht an die Spindel fassen! Ansonsten ist der Einstieg weniger wild, als oft befürchtet.

Preissturz beim Raspberry Pi

Die Preise des Raspberry Pi haben über die Sommermonate eine rasante Talfahrt hingelegt und sind inzwischen wieder bei unter 100 € für das 8 GB-Modell angekommen - ein idealer Zeitpunkt also, um endlich die ganzen auf Eis liegenden Raspi-Projekte nachzuholen. Da der Kleincomputer aber mehr Rechenleistung mitbringt, als die meisten Projekte benötigen, lohnt ein Blick auf Docker. Damit lassen sich mehrere Anwendungen auf einem Raspi kombinieren und einfach warten. Wie die Installation von Docker und mehreren Projekten innerhalb einzelner Container funktioniert, zeigt unser Tutorial-Video. Neben einem PC oder Mac ist dafür nur ein ans Netzwerk angeschlossener Raspberry Pi nötig, einen Bildschirm, Maus oder Tastatur benötigt man für den Raspi nicht. Stattdessen wird dieser übers Netzwerk eingerichtet und via SSH-Zugriff Docker sowie die Container-Verwaltung Portainer installiert. Eine Anleitung auch für Einsteiger und Netzwerk- sowie Raspi-Unerfahrene.

Und wer nach CNC-Fräsen und Raspi-Projekten Lust auf dreckige Hände hat, kann die klebrig gewordenen Elektrogeräte aus der Schublade holen und mit unserer Anleitung zum Entfernen der sogenannten Softtouch-Oberfläche das Wochenende abrunden. Nachdem unser Videoredakteur Johannes Börnsen im Netz auf den Tipp gestoßen ist, diese nach einiger Zeit anhänglich werdenden Beschichtungen mit Orangenöl-Reiniger zu entfernen, haben uns über 500 Kommentare und Hinweise erreicht, welche Mittel noch geeignet wären. Also haben wir uns mit allerlei mehr oder weniger naheliegenden Mitteln bewaffnet und die Tipps an alten Geräten ausprobiert. Von Babypuder über Aceton und Isopropanol zu Backofenreiniger und Geschirrspülmaschine waren diverse Hausmittel im Einsatz - mit teils überraschenden Ergebnissen. Auch dieses Video finden Sie auf dem Youtube-Kanal vom Make Magazin und über den Link am Ende dieses Artikels. *—jom*

Bitte nicht anfassen – besser erst reinigen! Aber womit verschwindet die Klebeschicht am schnellsten?

▶ Die Links zu den genannten Videos gibt es unter: make-magazin.de/xbvf

Immer freitags: Das Make Magazin auf Youtube

Wenn in unserer Make-Videowerkstatt das Licht angeht, wird gelötet, gesägt, gehämmert und geschweißt: Jede Woche zeigt unser Videoredakteur Johannes Börnsen, wie es geht. Egal, ob eine Maker-Werkbank für 50 €, ein selbst gebauter Halter fürs Balkonkraftwerk, WLED für Einsteiger oder eine Retro-Lampe aus Pappe und Sperrholz im Stil von Marquee-Buchstaben: Unsere Youtube-Videos sind kostenlos und ohne Abo abrufbar. Unterstützt wird Johannes dabei von den anderen Redaktionsmitgliedern, von denen Johannes sich gerne erklären lässt, was er selbst nicht weiß. Übrigens: Unsere bisher auf Youtube veröffentlichen Artikelergänzungsvideos laden wir auch weiterhin auf Youtube hoch. Den Link dazu finden sie hinter der Short-URL des jeweiligen Artikels im Heft. *—jom*

▶ Das Make Magazin gibt es unter: youtube.com/MakeMagazinDE

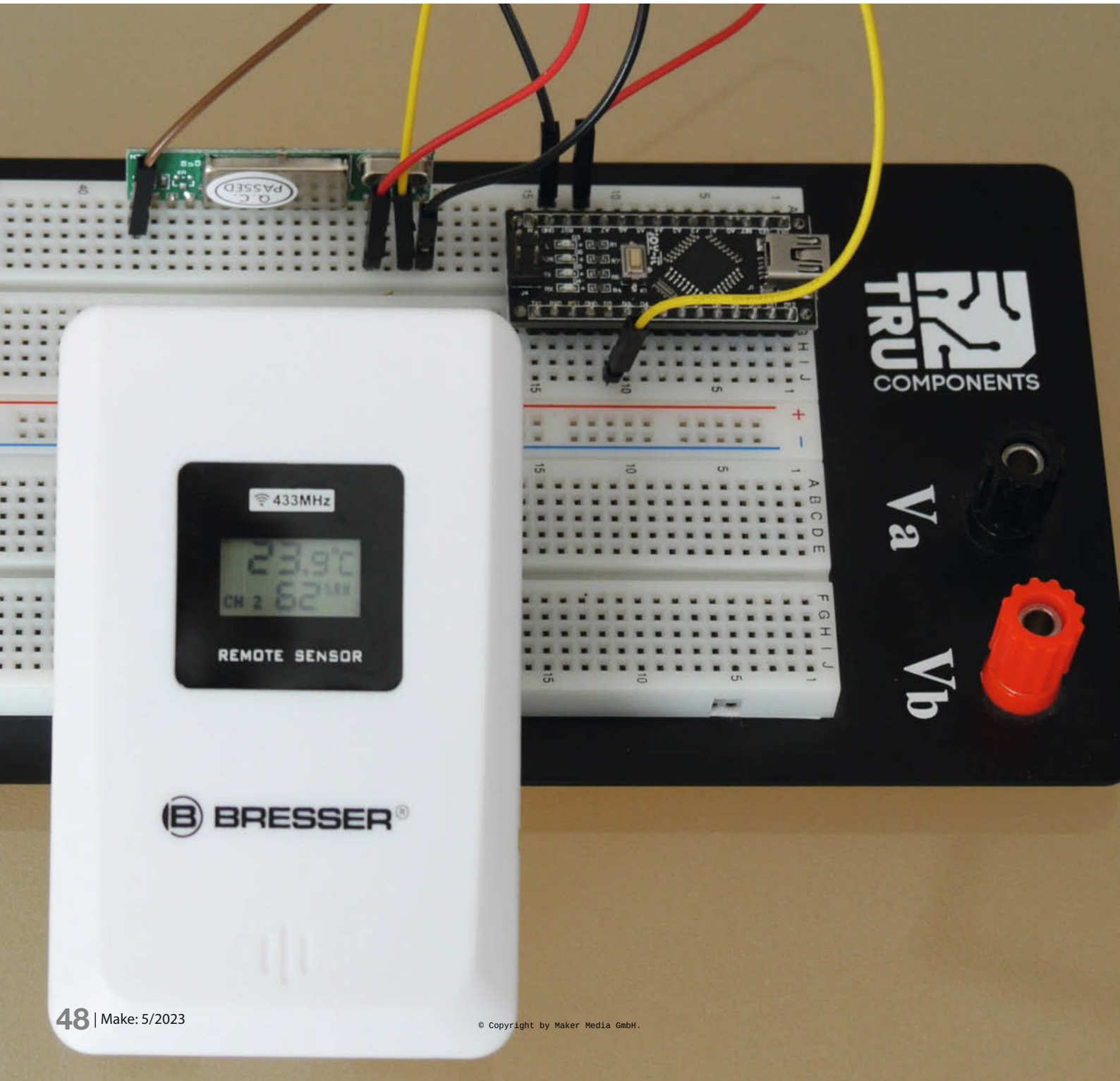
Bitte nicht aufs Bild touchen! Funktionierende Videolinks gibts in der Short-URL und auf unserem Youtube-Kanal.

© Copyright by Maker Media GmbH.

Temperatursensoren für den ioBroker Gaszähler

In Make 3/23 wurde das Projekt zur Integration eines Gaszählers in das Smarthome vorgestellt. In diesem Artikel geht es darum, über die Temperaturmessung weitere Daten zur Verbrauchsoptimierung zu erhalten. Dazu können die üblichen Maker-Sensoren oder sehr praktische Funksensoren auf 433MHz-Basis verwendet werden.

von Andreas Koritnik



© Copyright by Maker Media GmbH.

Das Messen und Aufzeichnen von Umweltdaten wie Temperatur und Luftfeuchtigkeit ist für viele Smart-Home-Projekte unerlässlich. Insbesondere wenn es um die Einsparung von Heizenergie geht, ist es sinnvoll, neben den Verbrauchsdaten auch die entsprechenden Umweltdaten zu messen und aufzuzeichnen. Nur so können die Auswirkungen von Änderungen beim eigenen Heizverhalten richtig eingeschätzt werden.

Zwei beliebte Sensoren sind der DHT11 und der BME280, die preiswert und leicht zu beschaffen sind. Bibliotheken für die Arduino-Umgebung sind ebenfalls verfügbar. Etwas komplizierter wird es, wenn man mehrere Sensoren im Haus verteilt, um Temperatur und Luftfeuchtigkeit in verschiedenen Räumen zu messen. Eine Kabelverbindung zu einem zentralen Arduino oder Raspberry ist sicherlich keine elegante oder praktikable Lösung. Deshalb zeige ich in diesem Artikel, wie es besser geht: Nämlich mit separaten ESP8266-Modulen (D1-Mini), die die Daten per WLAN und MQTT-Protokoll an einen zentralen Server senden. MQTT (Message Queuing Telemetry Transport) ist ein Protokoll, das eine einfache Übertragung von Werten im Netzwerk ermöglicht. Als zentralen Server verwende ich ioBroker auf einemRaspberry Pi. Es funktionieren aber auch andere Smarthome Systeme (z. B. Home Assistant), die MQTT unterstützen.

Speziell für Szenarien, in denen keine kabelgebundene Stromversorgung möglich ist, eignen sich auch externe Sensoren von Wetterstationen. Diese batteriebetriebenen Sensoren senden die gemessenen Daten regelmäßig über die Frequenz 433 MHz. Zu Hause hatte ich bereits einen solchen Sensor der Marke Bresser zur Anzeige von Außentemperatur und Luftfeuchtigkeit auf meiner Wetterstation im Wohnzimmer.

Was lag also näher, als den vorhandenen Sensor zu nutzen und die Daten im ioBroker zu verwenden? Eigentlich findet man für fast alle Probleme fertige Lösungen im Internet – für meinen Bresser-Sensor leider nicht. Allerdings habe ich eine Beschreibung für das verwendete Übertragungsprotokoll gefunden. Damit konnte ich schon etwas anfangen.

Ein Arduino Nano (Bild 1) mit einem 433-MHz-Empfänger und einer kurzen Antenne reicht aus, um die vom Sensor gesendeten Daten zu empfangen und zu dekodieren. Die Daten werden dann JSON-kodiert und über die serielle USB-Schnittstelle an den Raspberry Pi ausgegeben. Mit einem kleinen Javascript-Programm können die Daten dann in ioBroker eingelesen und als Objektinformationen zur Verfügung gestellt werden.

Die Programme auf GitHub sind so geschrieben, dass die ioBroker-Visualisierung des smarten Gaszählers aus Make 3/23 mit den vergebenen Objektnamen direkt funktioniert. Für die Innentemperaturen verwendet man einen DHT11 oder einen BME280. Für die Außenwerte den Bresser-Sensor mit Arduino Nano.

Kurzinfo

» Der smarte Gaszähler wird noch smarter mit Temperatursensoren
» Bresser-433MHz-Funksensor mit Arduino dekodieren
» Integration in ioBroker per JavaScript

Checkliste

Zeitaufwand:
1 Stunde pro Sensor

Kosten:
10 bis 30 Euro je nach Sensor

Löten:
Grundkenntnisse

Programmieren:
Anpassen von Arduino-Skripten und Javascript

Werkzeug

» **Übliches Maker-Werkzeug** Zangen, Seitenschneider, Schraubendreher,
» **Lötwerkzeug** Lötkolben, Lötzinn, Lötschwamm, …

Material

» **ESP8266** D1-Mini
» **Umweltsensor** BME280 oder DHT11
» **USB-Netzteil** für D1-Mini
» **Bresser-3CH-Funksensor**
» **Arduino Nano**
» **433MHz-Superhet-Empfänger** RXB6
» **Litze** zur Verkabelung
» **passendes Gehäuse**

Mehr zum Thema

» Andreas Koritnik, Smarter Gaszähler mit Smarthome-Integration, Make 3/32, S. 20
» Uwe Rohne, Smarter Dimmer fürs DIY-IoT, Make 1/20, S. 56

Alles zum Artikel im Web unter *make-magazin.de/xhe1*

Messung mit DHT11

Der DHT11 hat in der Regel drei Anschlüsse. Neuerdings werden DHT11 mit vier Anschlüssen geliefert, wobei einer der Anschlüsse keine Funktion hat (Bild 2).

Die drei Pins sind: Versorgungsspannung (3,3 bis 5 Volt), GND und Daten. Für die korrekte Funktion ist ein Pull-up-Widerstand (4,7 kOhm zwischen Plus und Daten) erforderlich. Dieser Widerstand ist oft bereits auf den

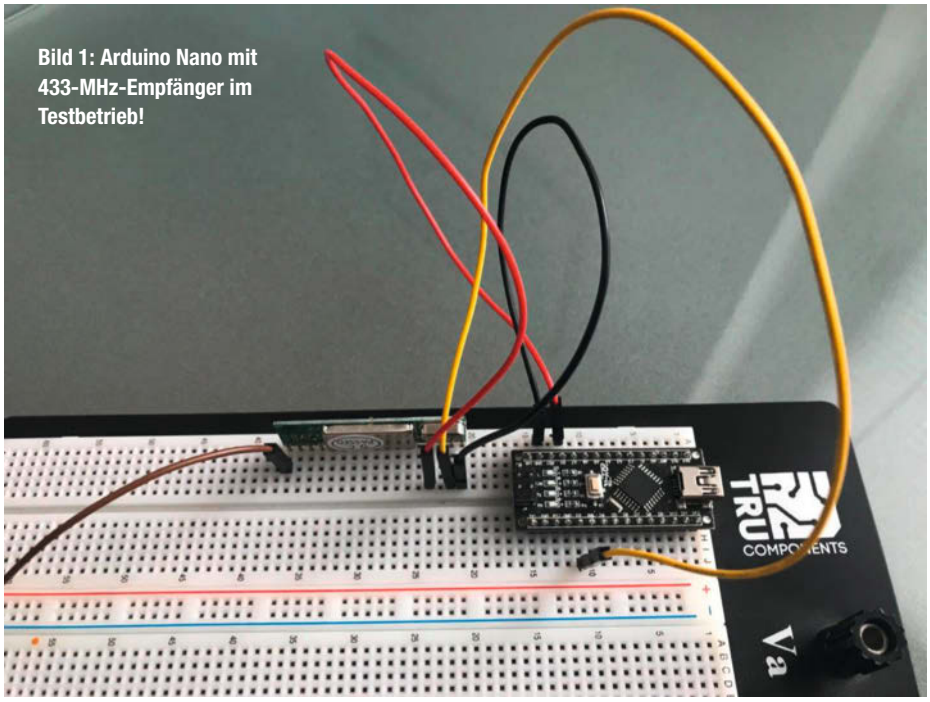


Bild 1: Arduino Nano mit 433-MHz-Empfänger im Testbetrieb!

© Copyright by Maker Media GmbH.

433-MHz-Funk

Der Frequenzbereich 433 MHz (433,05 bis 434,79 MHz, 10 Kanäle) liegt in einem sogenannten ISM-Band, das sind Frequenzbereiche, die ursprünglich für **I**ndustrial, **S**cientific und **M**edical Use (Industrie, Wissenschaft und Medizin) zum Betrieb von Hochfrequenzgeräten reserviert waren. Viele dieser Bänder sind lizenz- und genehmigungsfrei zu nutzen. Später wurden diese Bänder auch für kurze Audio- und Videoübertragungen genutzt, diese waren aber häufig von Störungen betroffen. Erst neue digitale und störungsunempfindlichere Übertragungsverfahren ermöglichten die Übertragung mit einer gewissen Zuverlässigkeit. Damit kamen immer mehr Sensoren (Temperatur, Bewegung etc.), Funkdienste (Babyphone, Funkgeräte für kurze Distanzen, Funkkopfhörer, Funkschalter, etc.) und Datenübertragungen (z. B. für elektronische Preisschilder) in das Band.

Je nach Dienst kommen unterschiedliche Modulationsarten wie AM (Amplitudenmodulation) oder FM (Frequenzmodulation) mit typischerweise 10 Milliwatt Leistung zum Einsatz, auf denen dann die Protokolle und Datenformate aufbauen. Die Reichweiten liegen je nach Situation zwischen wenigen und einigen hundert Metern.

Aufgrund der hohen Auslastung im Funkverkehr und der Interferenzen durch die eigentlichen ISM-Geräte ist mit hohen Datenverlusten zu rechnen, die durch Prüfsummen und getaktete Übertragungen kompensiert werden. Als Alternative bietet sich das 868-MHz-Band an, das in vielen Ländern ebenfalls lizenzfrei genutzt werden kann. Allerdings sind hier strengere Vorschriften zu beachten, die sicherstellen sollen, dass sich viele Nutzer das Band teilen können.

gelieferten Platinen vorhanden. Wenn das nicht dokumentiert ist, probiert es erst einmal ohne zusätzlichen Widerstand. Wenn es mit dem DHT11 nicht funktioniert, dann den Widerstand hinzufügen. Der Anschluss an den D1-Mini (Bild 2) ist schnell gemacht. Einfach 3,3 Volt, GND und Daten mit D6 (GPIO 12) verbinden. Der Datenpin D6 ist im Arduino-Sketch entsprechend definiert. Das Programm selbst ist auf GitHub im bereits vorhandenen Repository zum Gaszähler zu finden, Link siehe Kurzinfo.

In der Arduino-IDE muss als Board der „ESP8266 Lolin(WEMOS) D1 R2 &mini“ eingestellt werden. Außerdem müssen zwei Bibliotheken installiert werden: einmal PubSubClient für MQTT und dann noch die Adafruit DHT Sensor Unterstützung aus der Adafruit Unified Sensor Library. Die Installation kann bequem mit dem Arduino Library Manager durch Eingabe und Suche der Bibliotheksnamen erfolgen.

Als Nächstes müssen noch die WLAN-Zugangsdaten, die Adresse des MQTT-Servers und der Name des Sensors und eventuell Zugangsdaten für MQTT angegeben werden (Kasten Anpassung des Arduino-Sketch). Das war's.

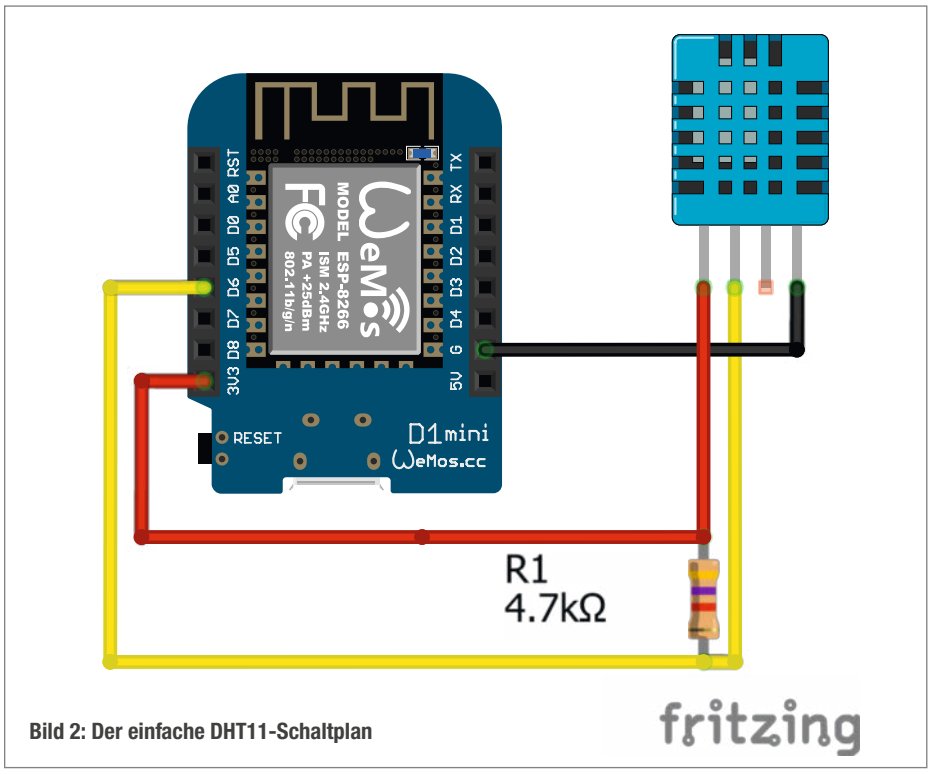


Bild 2: Der einfache DHT11-Schaltplan

Unweltdaten mit BME280

Ein weiterer beliebter Sensor ist der BME280 des Herstellers Bosch, der neben Temperatur und Feuchte auch den Luftdruck messen kann. Zur Datenübertragung verwendet der BME280 das I²C-Protokoll. Für den Anschluss an den D1-Mini werden vier Anschlüsse benötigt: 3,3 Volt Versorgungsspannung, GND und die beiden Datenleitungen SDA und SCL.

Die Anschlüsse für ein Sparkfun BME280 Breakout zum D1-Mini sind in Bild 3 dargestellt. Dieses Board benötigt für unsere Zwecke keine weiteren Anschlüsse. Die Adresse kann auf der Rückseite des Boards geändert werden, standardmäßig ist sie auf die I²C-Adresse `0x77` gesetzt und muss im Sketch entsprechend eingetragen werden (`bme.begin(0x77)`, Zeile 53 im GitHub). Leider gibt es bei den Boards etwas Wildwuchs, daher immer die Dokumentation und Beispiele des Herstellers beachten.

Der BME280-Sketch steht auch auf der GitHub-Seite zum Download bereit. Für WLAN und MQTT bitte die gleichen Änderungen vornehmen, wie für den DHT11 beschrieben. Zusätzlich zur PubSubClient-Bibliothek muss noch die Adafruit-BME280-Bibliothek mit dem Bibliotheksverwalter der Arduino IDE installiert werden.

Bresser-Funksensor dekodieren

Die Messung mit den beiden bisher beschriebenen Sensoren ist sicher schon einigen Lesern bekannt. Nun aber zu etwas Neuem: Die Verwendung eines Bresser-3-Kanal-Sensors. Dieser Sensor kann z. B. im Bresser-Onlineshop als Ersatz-Funksensor für Bresser-Wetterstationen bestellt werden. Eine Bresser-Wetterstation wird nicht benötigt. Hier nimmt uns der Sensor einiges an Arbeit ab, wo bei den Maker-Sensoren noch für jeden Sensor im Arduino-Sketch gebastelt und eine Firmware übertragen werden musste.

Für den Datenempfang verwende ich einen Arduino Nano und einen 433-MHz-Empfänger, bei mir ein RXB6-Superhet-Empfänger (Bild 4). Damit funktioniert alles sehr gut. Es gibt auch

© Copyright by Maker Media GmbH.

einfachere und vielleicht billigere Empfänger. Diese haben aber in der Regel eine deutlich schlechtere Empfangsqualität als der RXB6.

Die wiederum sehr einfache Verkabelung erfolgt wie in Bild 4 gezeigt. Als Antenne wird ein Draht mit einer Länge von 17,3 cm verwendet, das ist ein Viertel der Wellenlänge (λ). Damit hat man schon einen guten Empfang. Noch bessere Reichweiten erreicht man mit Wendelantennen für 433 MHz, die man im Internet bei vielen Anbietern findet, oder auch mit einer Selbstbauantenne (siehe Link in der Kurzinfo).

Software für den Nano

Den Arduino-Sketch findet ihr ebenfalls auf meiner GitHub-Seite (Link siehe Kurzinfo). Als Board müsst ihr in der Arduino IDE den Arduino Nano auswählen. Je nach Bezugsquelle oder Alter des Nanos kann es sein, dass ihr den Prozessor noch auf „ATmega328P (Old Bootloader)“ umstellen müsst, damit der Upload funktioniert.

Nachdem das Programm kompiliert und hochgeladen wurde, wartet der Nano auf Datagramme von den Bresser-Sensoren. Die Sensoren senden etwa einmal pro Minute Daten. Sobald ein gültiges Datagramm empfangen wurde, wird es über die serielle USB-Schnittstelle ausgegeben. Die Kodierung erfolgt im JSON-Format (JSON = JavaScript Object Notation). Die Ausgabe kann mit dem seriellen Monitor der Arduino-IDE betrachtet werden.

Das sieht dann beispielsweise so aus:
`{"id":63, "ch":2, "temp":18.8, "hum":62, "lowbatt":0}`

- `id` ist eine zufällige Byte-Nummer (0-255), die sich nach jedem Spannungsverlust (z. B. Batteriewechsel, alle 1-2 Jahre nötig) ändert.
- `ch` ist die Kanalnummer (1-3), die mit einem kleinen Schalter im Gehäuse des Sensors eingestellt werden kann. Es können also drei Bresser-Sensoren parallel verwendet werden.
- `temp` ist die Temperatur in °C .
- `hum` ist die relative Luftfeuchte in % .
- `lowbatt` ist der Batteriezustands-Indikator (0=OK, 1=niedrige Batteriespannung).

Die ID jedes Sensors ändert sich nach dem Zufallsprinzip bei jedem Batteriewechsel. Bitte notiert Euch euch die angezeigte ID und die Kanalnummer des Sensors. Sie werden später bei der Konfiguration des ioBroker-Skripts zur Identifizierung der einzelnen Sensoren benötigt. Durch die Kombination von Kanalnummer und zufälliger ID können sehr viele Sensoren eingebunden werden. Prinzipiell ist es möglich, dass zwei Sensoren zufällig die gleiche ID erhalten. In diesem Fall muss nur die Batterie eines Sensors kurz entfernt und wieder eingesetzt werden. In diesem Fall und nach einem Batteriewechsel muss auch das

Anpassen des Arduino Sketches

```
// Update these with values suitable for your network.
const char* ssid = "ssid";
const char* password = "password";
// Set IP of MQTT server and MQTT Client ID
const char* mqtt_server = "192.168.0.71";
String clientId = "DHT11-Sender";
```

Der frei wählbare Name (`clientId`) sollte eindeutig sein, damit die Sensoren unterschieden werden können. Nach dem Upload werden Temperatur und Luftfeuchtigkeit alle fünf Sekunden gemessen und per MQTT an den Server (bei mir ioBroker) gesendet. Standardmäßig verwendet der Sketch MQTT ohne Benutzername/Passwort. Falls ihr MQTT mit Zugangsdaten konfiguriert habt, müsst ihr die Zeile

```
if (MQTT_client.connect(clientId.c_str()))
```

zu

```
if (MQTT_client.connect(clientId.c_str(), "Benutzername", "Passwort"))
```

ändern.

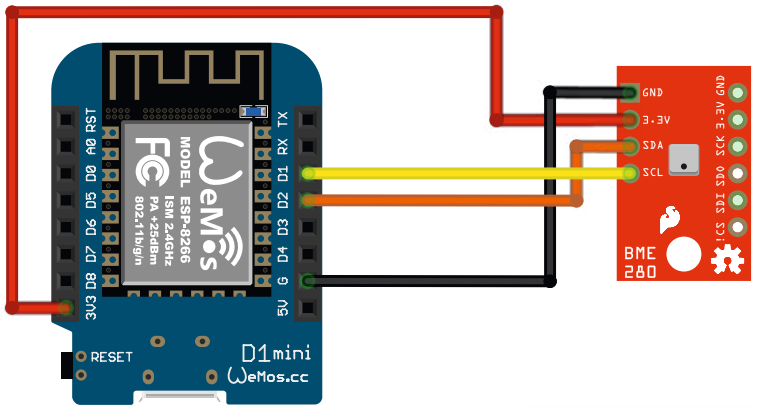


Bild 3: Der BME280-Schaltplan

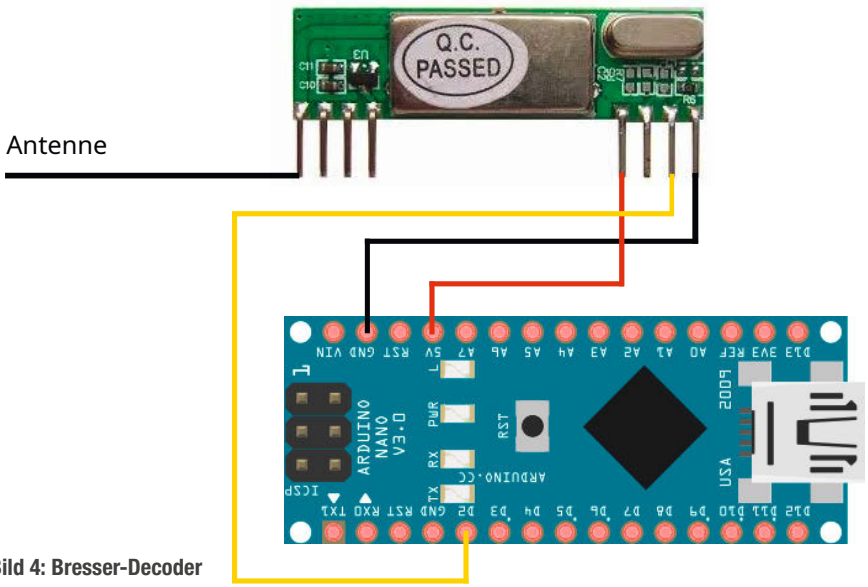


Bild 4: Bresser-Decoder

© Copyright by Maker Media GmbH.

ioBroker-Skript gestoppt, die ID eingegeben und das Skript neu gestartet werden. Mehr dazu im letzten Abschnitt.

Funktionsweise des Programms

Die Kernfunktion des Decoderprogramms ist die Interruptfunktion `rx433Handler()`. Diese Funktion wird bei jeder Zustandsänderung der Datenleitung des RXB6-Empfängers aufgerufen. Die Übertragung der Daten erfolgt im sogenannten On-Off-Keying-Verfahren, bei dem die Sendeleistung des Senders jeweils kurz ein- und wieder ausgeschaltet wird. Der Empfänger erkennt das Vorhandensein bzw. Nichtvorhandensein der gesendeten Frequenz. Ist die Datenleitung des Empfängers High, bedeutet dies, die Sendefrequenz wurde erkannt. Dementsprechend bei Low: keine Frequenz erkannt. In der Funktion wird die Dauer jedes High-Pulses gemessen und mit dem definierten Timing des Bresser-Sensors verglichen. Die Datagramme werden ca. alle 60 Sekunden gesendet und 15 Mal wiederholt.

Genau diese Signale (siehe Infografik Format und Timing des Bresser-Datagrams) werden von der Funktion gesucht. Die Bits werden in einem Puffer gesammelt und gezählt. Jede gültige Synchronisationssequenz (750 µs) oder jedes ungültige Timing setzt den Pufferzähler auf null zurück.

Wenn alle 40 Bits ohne Timing-Fehler empfangen wurden (32 Datenbits und 8 Bits Prüfsumme), wird die empfangene Prüfsumme berechnet und mit der für die 32 Datenbits verglichen. Ist die Prüfsumme korrekt, werden die Daten in einen Ringspeicher geschrieben.

Die Kodierung der 40 Bits ist in Bild 5 zu sehen. Dabei bedeuten die Bits:

- `i`: 8 Bit für die ID des Sensors (s. o.)
- `b`: Batteriespannung (0: OK, 1: zu niedrig)
- `s`: Test/Sync (0: Normalbetrieb, 1: Test-Button gedrückt / Synchronisation mit Wetterstation)
- `c`: Kanal (MSB zuerst), 2 Bit für die Kanalnummer (gültige Kanäle sind 1, 2, 3)
- `t`: Temperatur (MSB zuerst, Big-Endian) 12 Bit vorzeichenlose Grad Fahrenheit mit Offset 90 und multipliziert mit 10
- `h`: Luftfeuchte (MSB zuerst) 8 Bit relative Feuchtigkeit in Prozent
- `x`: Prüfsumme = (Byte1 + Byte2 + Byte3 + Byte4) % 256

MSB beschreibt die Bitwertigkeit, wobei MSB für Most Significant Bit steht; das Gegenteil ist LSB (Least Significant Bit). Zusammen mit Big- Endian (Gegenstück Little-Endian) beschreibt dies die Reihenfolge, in der die Werte in Bits bzw. Bytes kodiert werden. Das %-Zeichen steht für eine Modulo-Rechenoperation, den Rest bei der Ganzzahldivision. Aber ich möchte hier nicht ins Detail gehen. Bei Interesse einfach im Internet nach den Begriffen suchen und sich nicht wundern, dass Big-Endian etwas mit Gullivers Reisen und Eiern zu tun hat.

In der Funktion `loop()` wird ständig geprüft, ob ein neuer Wert im Ringpuffer verfügbar ist. Ist das der Fall, werden die Daten mit der Funktion `getResults()` nach dem beschriebenen Schema dekodiert und Fahrenheit in Celsius umgerechnet. Die dekodierten Daten werden dann im JSON-Format über die serielle USB-Schnittstelle ausgegeben.

ioBroker Integration per Skript

Die JSON-formatierte Ausgabe kann nun einfach mit Smarthome-Programmen verwendet werden. Hier zeige ich die Integration in die ioBroker-Plattform mittels JS (JavaScript). Um das Skript in ioBroker verwenden zu können, muss zuerst der JavaScript/Blockly Adapter in ioBroker installiert werden. Falls der Adapter noch nicht installiert ist, einfach im ioBroker-Admin-Fenster auf der linken Seite „Adapter" auswählen. Dann nach dem Adapter „Skriptausführung" suchen und „JavaScript/Blockly" mit dem +-Knopf installieren (Bild 6).

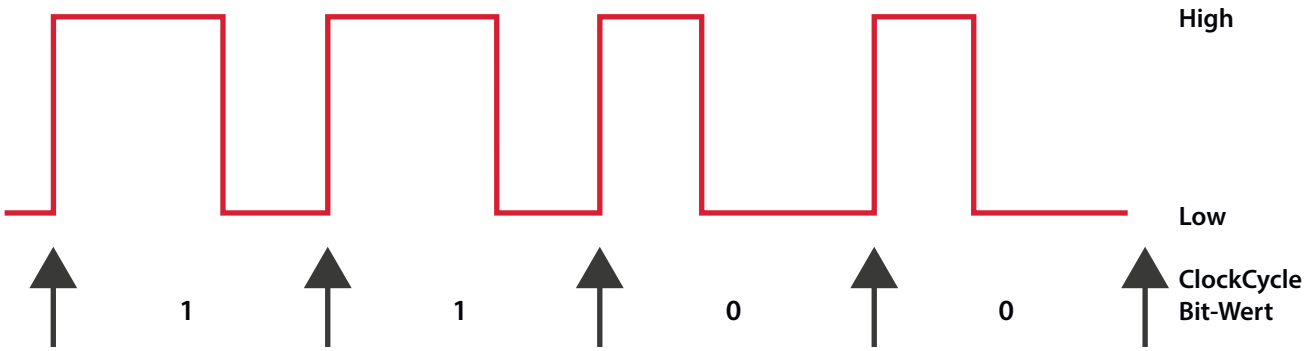
Mit dem Skript aus dem Listing „JSON Parser for ioBroker" werden die JSON-Daten vom Nano gelesen und entsprechende Statusinformationen für Objekte gesetzt.

Listing: JSON Parser für ioBroker

```
// Create a serial port
const { SerialPort } = require('serialport');
const port = new SerialPort({
  path: '/dev/ttyACM0',   // Set the device name
  baudRate: 115200,
});

// Create line parser
const { ReadlineParser } = require('@serialport/parser-readline');
const parser = port.pipe(new ReadlineParser({ delimiter: '\r\n' }));

// Create states if they do not exist yet
createState("javascript.0.Temperature", 0,
  {name: "Temperature",  role: 'variable', type: 'number'});
createState("javascript.0.Humidity", 0,
  {name: "Humidity",  role: 'variable', type: 'number'});
createState("javascript.0.BatteryState", 0,
  {name: "Battery State",  role: 'variable', type: 'number'});
createState("javascript.0.C2_Temperature", 0,
  {name: "C2_Temperature",  role: 'variable', type: 'number'});
createState("javascript.0.C2_Humidity", 0,
  {name: "C2_Humidity",  role: 'variable', type: 'number'});
createState("javascript.0.C2_BatteryState", 0,
  {name: "C2 Battery State",  role: 'variable', type: 'number'});

// Define parser function

parser.on('data', function(data){
  // console.log(data);
  try{
    var messageObject = JSON.parse(data);
  }
  catch (e) {
    return console.error(e);
  }

  const temp =  messageObject.temp;
  const hum =  messageObject.hum;
  const ch = messageObject.ch;
  const id = messageObject.id;
  const lowbatt = messageObject.lowbatt;

  // Adjust channel and id to your needs
  if(ch == 1 && id == 4) {
    setState("javascript.0.Temperature", temp, true);
    setState("javascript.0.Humidity", hum, true);
    setState("javascript.0.BatteryState", lowbatt, true);
  }

  if(ch == 2 && id == 63) {
    setState("javascript.0.C2_Temperature", temp, true);
    setState("javascript.0.C2_Humidity", hum, true);
    setState("javascript.0.C2_BatteryState", lowbatt, true);
  }
});
```


© Copyright by Maker Media GmbH.

Format und Timing des Bresser-Datagrams



- Ein langer High-Impuls von 500µs gefolgt von einem Low-Impuls von 250µs ist eine logische 1.
- Ein kurzer High-Impuls von 250µs gefolgt von einem langen Low-Impuls von 500µs ist eine logische 0.
- Vor jeder Übertragung sendet der Sensor eine Synchronisationsfolge von jeweils 750µs abwechselnden High/Low-Impulsen, die viermal wiederholt wird.

Das Skript, das natürlich ebenfalls auf GitHub heruntergeladen werden kann, muss in den ioBroker-Javascript-Editor eingegeben und gespeichert werden. Der Skript-Editor ist über den Menüpunkt Scripts auf der linken Seite im Menü zu erreichen.

Für das Skript gehe ich davon aus, dass ioBroker auf einem Raspberry Pi läuft. Falls nicht, muss der Gerätename für die serielle USB-Schnittstelle im Skript entsprechend geändert werden. Wenn der Nano die einzige USB-Seriell-Schnittstelle am Raspberry ist, sollte `/dev/ttyACM0` der richtige Gerätename sein. Ansonsten kann der Gerätename mit dem Kommando `dmesg` auf der Kommandozeile des Raspberrys herausgefunden werden. Die Stromversorgung des Nano erfolgt direkt über die USB-Buchse des Raspberry.

```
1111 1100 | 0001 0110 | 0001 0000 | 0011 0111 | 0101 1001 0 65.1 F 55 %
iiii iiii | bscc tttt | tttt tttt | hhhh hhhh | xxxx xxxx
```

Bild 5: Kodierung der Bits in einem Datagramm

Das Skript legt die benötigten Objekte in ioBroker selbst an, wenn sie noch nicht vorhanden sind. Das bereitgestellte Skript unterstützt zwei Sensoren. Wenn weniger oder mehr Sensoren verwendet werden sollen, können die Zeilen einfach auskommentiert oder dupliziert werden. Für jeden Sensor müssen die Kanalnummer `ch` und die zufällige ID (`id`, auch nach Batteriewechsel alle 1-2 Jahre) geändert und die Statusnamen in `setState()` angepasst werden. Die Werte der aktuellen Sensoren können mit dem seriellen Monitor der Arduino IDE angezeigt werden. Die Gas-Meter-Visualisierung in ioBroker unterstützt auch den Batteriestatus. Eine niedrige Sensor-Batteriespannung wird mit einem blinkenden Batteriesymbol angezeigt.

Allerdings muss bei jeder Änderung des Skripts die Javascript-Instanz im ioBroker gestoppt und neu gestartet werden. Ansonsten kommt es zu Fehlermeldungen über ein blockiertes serielles Gerät und das Skript funktioniert nicht.

Mit den zusätzlichen Informationen zur Sensorintegration ist der smarte Gaszähler nun vollständig und bereit für die nächste Heizperiode und die Optimierung des Verbrauchs anhand von Umweltdaten. Viel Spaß beim Nachbauen. —*caw*

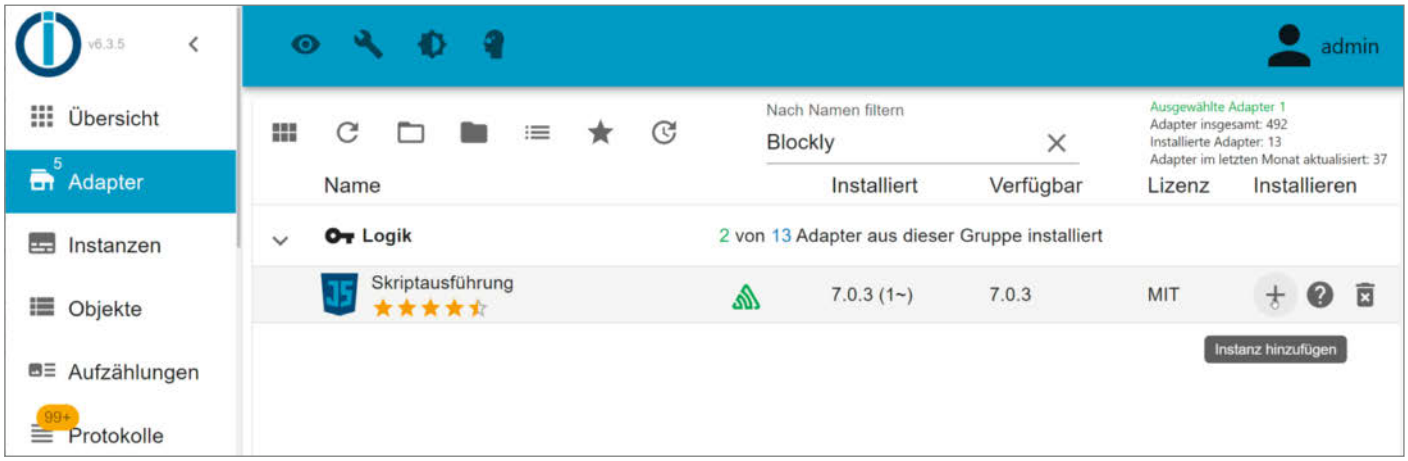
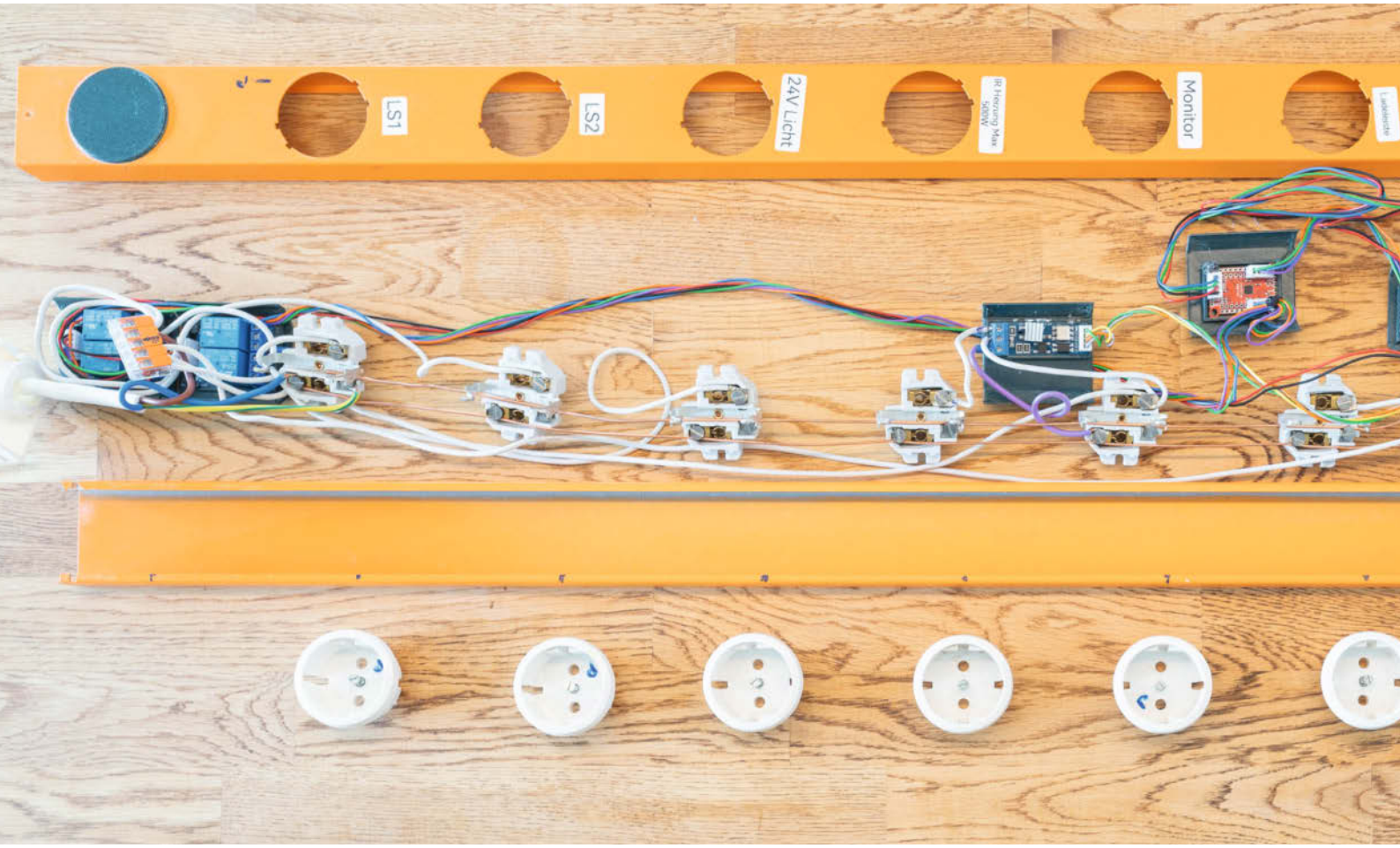


Bild 6: Skriptausführung in der Listenansicht installieren.

© Copyright by Maker Media GmbH.

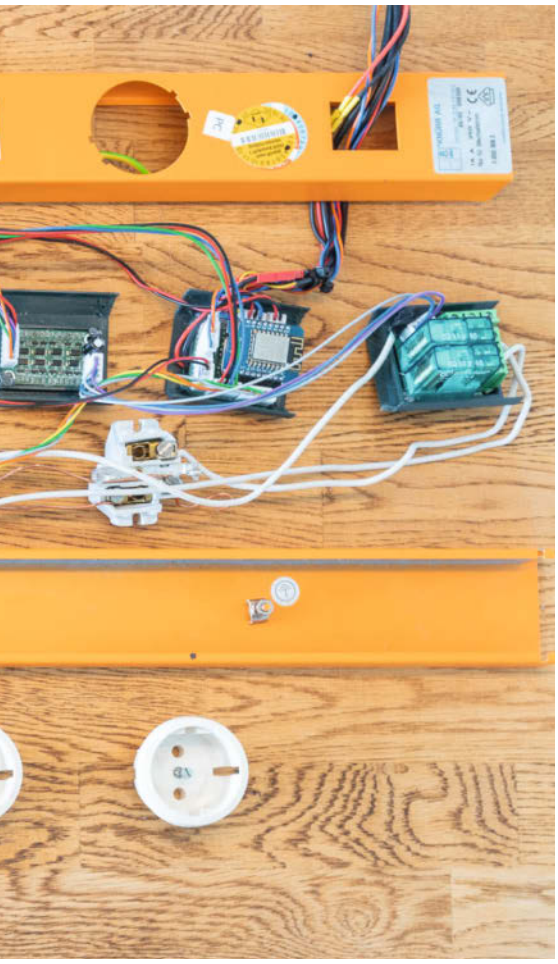


Schlaue Steckdose

Steckdosen, die man über das Handy schalten kann, sind nichts Neues. Auch gibt es Mehrfachsteckdosen, die nur Strom verteilen, wenn ein bestimmtes Gerät an ist. Das alles ist unserem Autor nicht genug und deshalb hat er sich eine Steckdosenleiste gebaut, die Geräte abhängig von seiner Anwesenheit mit Strom versorgt oder bei bestimmten Temperaturen die Heizung regelt.

von Jan Kipping

© Copyright by Maker Media GmbH.



Kurzinfo

» Strommanagement über Home Assistant
» Umsetzung mit Wemos D1 mini
» Komplette Technik in einer Steckdosenleiste

Checkliste

Kosten:
120 Euro

Werkzeug

» **Abisolierzange**
» **Schraubenzieherset**

Material

» **Wemos D1 Mini Controller**
» **2 Finder-Relais** Typ 40.31
» **Doppel-Relais** Typ Tru Components TC-9072472
» **Port Expander für D1 Mini Controller** Typ SX1509
» **Relais-Ansteuerplatine**
» **Temperatursensor** Typ Dallas DS18B20
» **DC/DC-Wandler** Typ MP1584EN
» **Signalkabel** 0,2 mm²
» **Stromkabel** NYM-J zertifiziert

Mehr zum Thema

» Heinz Behling, Smart Home: Intelligentes Heim mit Home Assistant, Make 1/21, S. 100
» Heinz Behling, Stromverbrauch mit Messgeräten ermitteln und senken, Make 1/23, S. 10
» Jan Kipping , Gut verbunden?, Make Sonderheft 2022 „Elektrotechnik", S. 38

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

Meine Steckdose soll zumindest die Peripherie für den Rechner sinnvoll ein- und ausschalten. Das klingt nach einer üblichen Steckdose mit Follow-Funktion. Ein Verbraucher wird überwacht, ab einer bestimmten Schwelle geht der Rest an. Also prinzipiell geht das. Bei mir leider nicht und vermutlich bei vielen anderen auch nicht. Dazu später mehr – siehe Kapitel Relaiskiller.

Damit eine Steckdose die Bezeichnung „intelligent" verdient, muss sie etwas mehr können:

– einen Teil der Peripherie abschalten, wenn ich nicht am Computer bin,
– den vorzeitigen Relaistod verhindern,
– den Arbeitsplatz mit minimalem Strom schön warm machen,
– die Ladegeräte am Schreibtisch managen,
– selbst wenig Strom verbrauchen,
– den Verbrauch der angeschlossenen Geräte schätzen,
– eine Relaisputzfunktion haben,
– AC-Dimmer mit Regelung für Infrarotheizung,
– bistabile Relais für minimalen Stromverbrauch,
– Port Expander, wenn mal wieder die I/O-Pins ausgehen,
– alles in einem stabilen und sicheren Gehäuse.

Die Software und die Integration basieren auf ESPhome und Home Assistant. Hier werden nur die speziellen Konfigurationen beschrieben.

Basisaufbau

Vor jedem Projekt stelle ich mir die Frage: Wo muss es hin?

Dieses Projekt hat sieben Steckdosen, die müssen geschaltet oder geregelt werden. Da muss ein robustes Gehäuse her. Ich habe schon viele schlechte Gehäuse gebaut. Das rächt sich, sobald ein Umbau ansteht, oder allein für die Renovierung das Projekt ab- und aufgebaut werden soll.

Bewährt haben sich bei mir massive Steckdosenleisten aus dem Serverbereich. Die sind sehr stabil und bieten viel Platz. Auf dem Bild ist eine Knürr-Steckdosenleiste mit weitem Abstand der Dosen zu sehen. Das Gehäuse ist aus Metall und hält wirklich was aus. Gebraucht auf eBay findet man immer mal wieder welche für einen anständigen Preis.

Für die Steckdose habe ich kleine Schiffchen gedruckt, die die eigene Elektronik aufnehmen. Sie lassen auf der Unterseite Kabel durch, um Strom und Signal zu verteilen. Die Elektronik selbst fixiere ich mit einem 3D-Pen, also einem kleinen 3D-Drucker für den Handdruck. Das ist meine erste nützliche Anwendung dafür, denn ich kann mir nicht vorstellen, mit den Geräten wirklich etwas händisch zu drucken.

Vorsicht, Netzspannung!

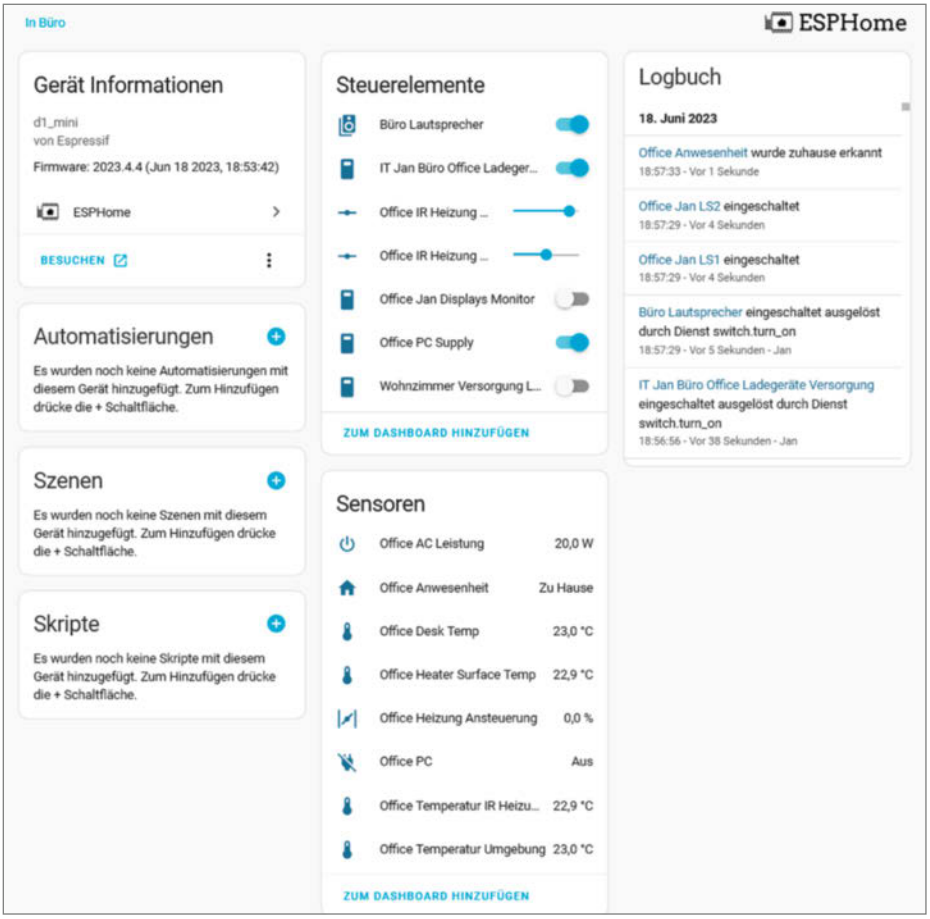
Der Umbau dieser Steckdosenleiste erfordert Erfahrungen bei der Verdrahtung netzspannungsführender Teile und die Beachtung einschlägiger VDE-Vorschriften. Das Hantieren mit Netzspannung ist lebensgefährlich!

Hier sind sieben Schiffchen verbaut für:

– den Controller selbst, hier ein Wemos D1 mini mit eigener Basisplatine,
– den AC-Dimmer,
– 2 × 2 Relais,
– 1 × 2 bistabile Relais,
– die Ansteuerplatine für die bistabilen Relais,
– einen Port Expander, da dem D1 mini für meine Projekte immer ein paar Pins fehlen.

Wie immer versuche ich zu verhindern, dass vermeidbare Fehler mein Projekt zerstören. Im Artikel „Gut verbunden" (siehe „Mehr zum Thema" in der Kurzinfo) ist das genauer erklärt. Hier nur so viel: Alle Signalkabel sind steckbar, die Stecker-Codierung ist möglichst eindeutig. Masse liegt immer auf Pin 2 der Stecker. Damit lebt die Elektronik auch bei Ungeschick oder Unwissen deutlich länger.

© Copyright by Maker Media GmbH.



So soll es am Ende in ESPHome aussehen.

Der Aufbau

Ich habe für den ESP8266 von Wemos (Mini D1) eine Adapterplatine entwickelt, damit die Schnittstellen einfach verbunden werden und Pull-up- und Sicherheitswiderstände einfach ergänzt werden können. Platz für einen FET (Feldeffekttransistor) und Spannungsteiler war auch noch, bleibt hier aber ungenutzt.

Die Stromversorgung kommt von einer DC/DC-Huckepack-Platine, die unter den Wemos passt. Sie sollte vor dem ersten Zusammenbau auf die richtige Spannung – hier 5 Volt – gesetzt werden. Alternativ kann auch direkt mit 5 Volt versorgt werden. Eine andere tolle Erfindung sind USB-C Power Delivery Adapter. Die kann man auf alle Spannungen setzen, die das Netzteil liefert – z. B. auch 5 Volt.

Die I/O Schnittstellen bestehen aus:

1. einem Bewegungsmelder, der sicherstellt, dass ein abwesender Benutzer nicht unnötig bespaßt oder beheizt wird,
2. dem analogen Eingang mit Optokoppler, der das 12-Volt-Signal von einem SATA-Stecker ausliest und erkennt, ob der PC an ist. Der Optokoppler mag redundant sein, aber bevor Masse oder gar Stromschleifen entstehen, lieber die Stromkreise trennen,
3. AC-Dimmer-Schnittstelle: Diese ist direkt in ESPHome integriert. Ein Pin liest den Nulldurchgang des AC-Signals, ein anderer steuert einen Thyristor,
4. zwei Temperatursensoren für die Heizungsregelung,
5. ein Port Expander mit I²C. An diesem befinden wiederum vier normale Relais: Monitor, Lautsprecher 1, Lautsprecher 2, Aux und zwei bistabile Relais (mehr dazu weiter unten).

Es kommen also etliche Bauteile zusammen. Daher ist am Ende alles recht dicht gepackt in der großen Steckdosenleiste.

Bevor alles eingebaut werden kann, versuche ich für alle Teile eine Position zu finden. Wenn das geschafft ist, konfektioniere ich die Kabel. Am Ende liegt der Zusammenbau neben der Leiste und wird, wenn alles passt, eingeschoben. Allerdings rate ich dringend dazu, vor dem Einschieben die Elektronik zu testen, am besten ohne 230 Volt. Und nach dem Einschieben nochmal! Zumindest bei mir lösen sich beim Zusammenbau häufiger wieder Verbindungen.

Die Konfiguration

Damit das gut funktioniert, nehme ich Stück für Stück die Konfiguration für die Schnittstellen in Betrieb. Wie das geht, wurde schon mehrfach in der Make beschrieben – siehe Kurzinfo. Gerade bei diesem Projekt ist das sehr individuell. Eine Kopie der Gesamtkonfiguration ist ebenfalls über den Link in der Kurzinfo zu finden, aber bestimmt nicht für jeden passend und eher als Ausgangsbasis gedacht. Ich glaube, es ist besser, wenn ihr jeden Teil selbst einrichtet und testet.

Diese Konfigurationen müssen in der config.yaml von Home Assistant eingestellt werden.

1. Anwesenheitserkennung

```
substitutions:
  presence_debounce_time: 10min

binary_sensor:
  - platform: gpio
```

Die Basis bildet eine metallverkleidete Knürr-Steckdosenleiste.

© Copyright by Maker Media GmbH.

```
    id: gpio_presence
    name: "Office Anwesenheit"
    device_class: presence
    pin:
      number: D0
      mode: INPUT
    filters:
      - delayed_off:
          ${presence_debounce_time}
```

Ich definiere die Anwesenheit auf mindestens 10 Minuten, der Wert ist Geschmackssache. Variablen habe ich gerne am Anfang der YAML-Datei als Ersetzung `substitution`, das hilft Code und Daten zu trennen und beim Recyceln des Codes.

Achtung bei diesen Sensoren: Die PIR-Bewegungssensoren, die billig zu haben sind, weisen unterschiedlichste Eigenschaften auf. Manche lösen bei WLAN in der Nähe dauerhaft aus. Die robusteren haben eine kleine Reichweite. Manche mögen einen Pull-Up-Widerstand am Eingang des Prozessors nicht. Hier hilft leider nur ausprobieren.

2. I/O-Mangel oder der Port Expander

Nie hat ein Prozessor genug Ein- und Ausgänge. Ich helfe mir, indem ich einen Port Expander per I²C anschließe. In ESPHome können damit nahezu beliebig viele Ein- und Ausgänge hinzugefügt und auch direkt verwendet werden. Das ist sehr praktisch.

Ihr könnt auch einfach einen größeren Prozessor nehmen wie den ESP32, aber der ist halt größer. Kostenmäßig ist es fast egal, die besseren Port Expander sind auch nicht billig. Denn: Billige Port Expander sind schlimm. Das hängt auch ein bisschen damit zusammen, wie ESPHome Ausgänge initialisiert. Ihr seid gewarnt! Ich nehme den SX1509.

```
i2c:
  sda: D2
  scl: D1
  scan: true
  id: i2c_a

sx1509:
  - id: sx1509_hub1
    address: 0x3E
```

Erst wird der I²C angelegt. Falls ihr alles richtig angeschlossen habt (4,7 Kiloohm an Pull-Up-Widerständen an den Signalleitungen) werden beim Einschalten die angeschlossenen Geräte im Log angezeigt. Der SX1509 wird als Hub definiert mit einer ID, die später in den Ein-/Ausgängen benötigt wird.

3. Die Schalter

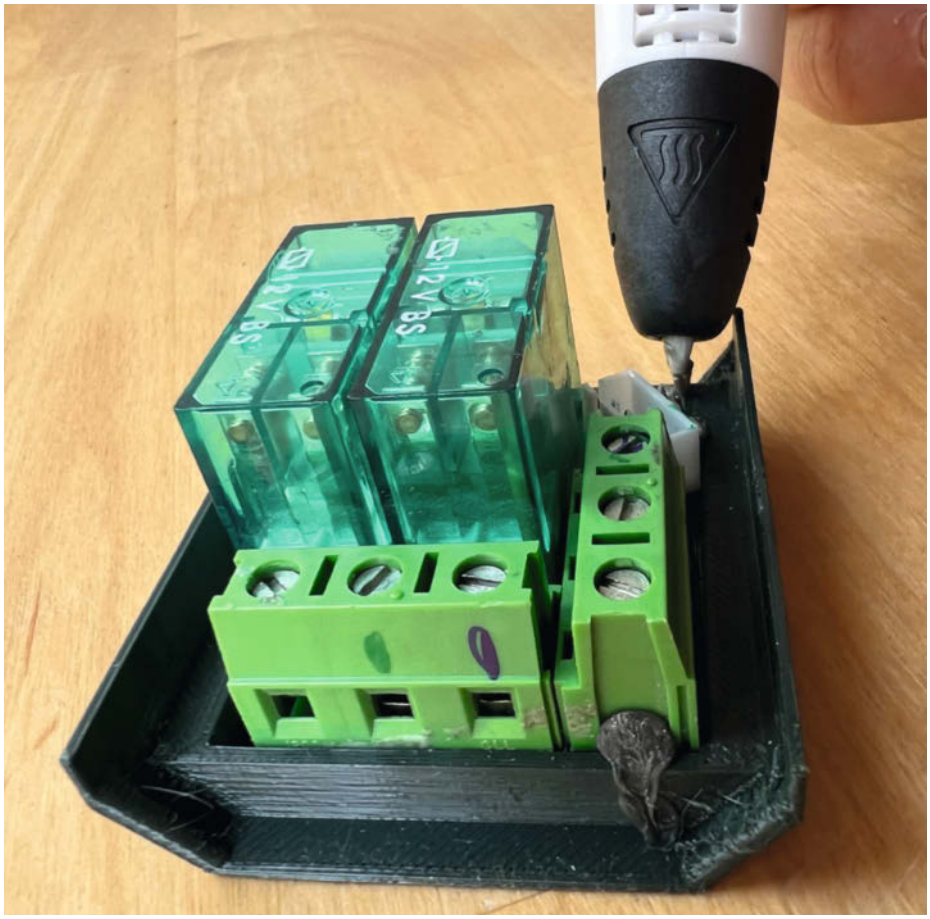
Ich definiere sie zuerst, da sie später bei den Sensoren referenziert werden.

```
switch:
  - platform: gpio
    name: "Office Jan LS1"
    id: "sw_speaker1"
    entity_category: "diagnostic"
    pin:
      sx1509: sx1509_hub1
      number: 4
      mode:
        output: true

  - platform: gpio
    name: "Office Jan LS2"
    id: "sw_speaker2"
    entity_category: "diagnostic"
    pin:
      sx1509: sx1509_hub1
      number: 7
      mode:
        output: true

  - platform: gpio
    name: "Office Jan Displays
Monitor"
    id: "sw_display"
    pin:
      sx1509: sx1509_hub1
      number: 5
      mode:
        output: true
```

In diese Halterungen (Schiffchen) werden die Elektronikteile eingesetzt.

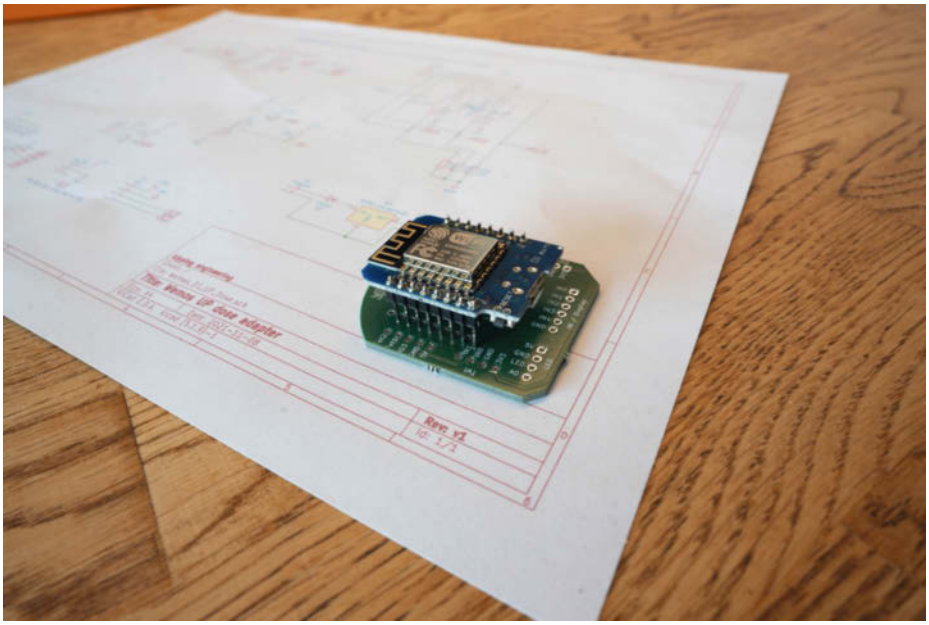
Damit habe ich die Ausgänge selbst definiert. Die `entity_category` stellt in Home Assistant sicher, dass die Schalter nicht prominent angezeigt werden. Sie werden nämlich als logischer Schalter zusammengefasst:

```
  - platform: template
    name: "Büro Lautsprecher"
    id: "office_speaker"
    icon: "mdi:speaker-multiple"
    optimistic: true
    turn_on_action:
      - delay: 200ms
```



Mit einem 3D-Druckstift werden die Bauteile in den Halterungen befestigt.

© Copyright by Maker Media GmbH.



Über diesen Controller wird später die Steckdosenleiste gesteuert.

```
    - switch.turn_on: sw_speaker1
    - delay: 200ms
    - switch.turn_on: sw_speaker2
  turn_off_action:
    - delay: 1sec
    - switch.turn_off: sw_speaker1
    - delay: 20ms
    - switch.turn_off: sw_speaker2
    - delay: 20ms
    - switch.turn_on: sw_speaker1
    - delay: 20ms
    - switch.turn_on: sw_speaker2
    - delay: 20ms
    - switch.turn_off: sw_speaker1
    - delay: 20ms
    - switch.turn_off: sw_speaker2
```

Das ist hier eine Besonderheit für Aktivlautsprecher, die gerne die Haussicherung auslösen. Der `template` Schalter sieht im Frontend wie ein Schalter aus, schaltet aber zwei Relais. Das `delay` stellt sicher, dass nacheinander und leicht verzögert geschaltet wird, damit die Hauptsicherung nicht überlastet wird. Die vielen Abschaltbefehle erkläre ich im Abschnitt Relaiskiller.

Ich konfiguriere den `template` Schalter als `optimistic`. Das stellt sicher, dass in Home Assistant der Schaltzustand sofort ankommt, denn die Programmlogik könnte auch viel komplizierter sein. Beispielsweise wird der Ein-Zustand erst nach Ablauf der Verzögerungen übertragen. Das brauche ich nicht. Sobald das `template` den Einschaltbefehl bekommt, wird es auch als eingeschaltet angezeigt. Hier wähle ich auch gleich ein hübsches Icon, das geht aber auch später in Home Assistant.

4. Bistabile Relais

Details erkläre ich weiter unten. Hier reicht es zu wissen, es gibt einen Pin zum Einschalten und einen zum Ausschalten:

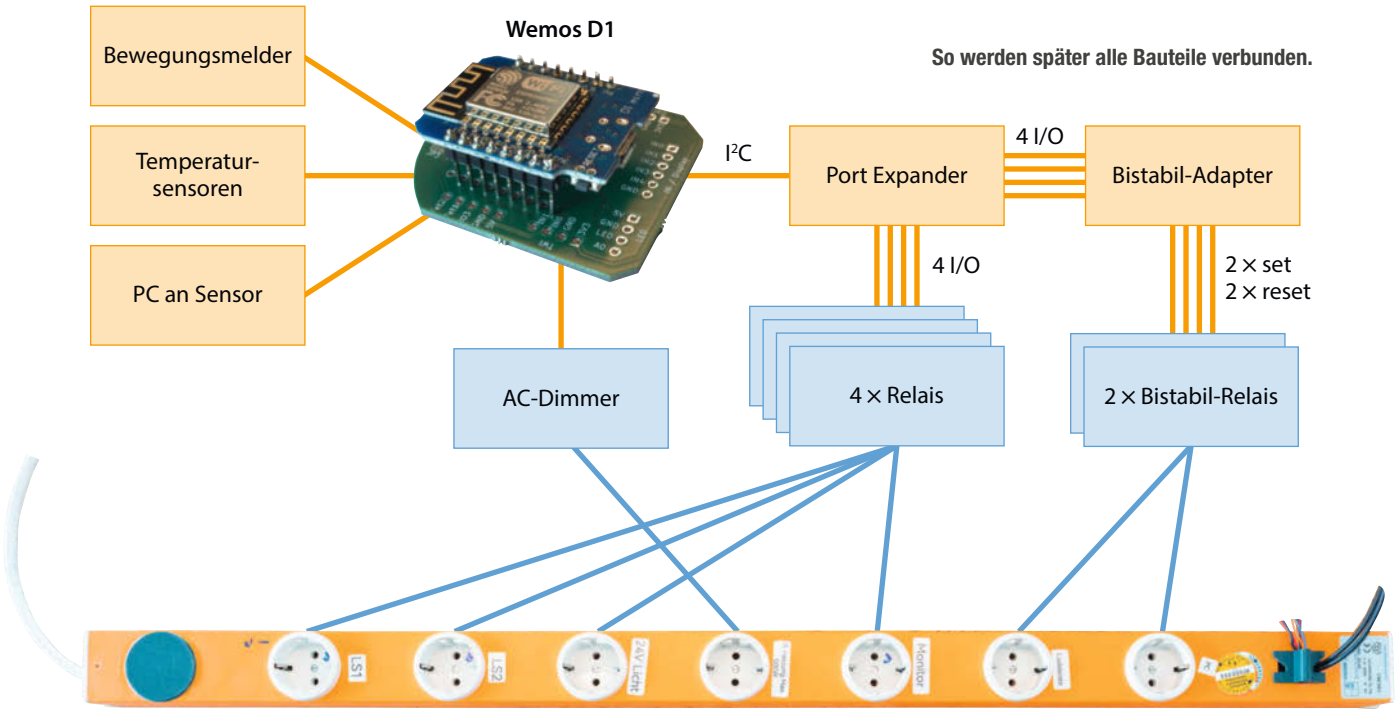
```
- platform: gpio
  id: "gpio_rset_bs_rel1"
  pin:
    sx1509: sx1509_hub1
    number: 8
    mode:
      output: true
    inverted: false

- platform: gpio
  id: "gpio_set_bs_rel1"
  pin:
    sx1509: sx1509_hub1
    number: 9
    mode:
      output: true
    inverted: false
```

Und ein Template (siehe Kasten Template) macht aus den zwei Schaltern wieder einen logischen:

Damit ESPHome den Rechner nach einem Reset nicht standardmäßig ausschaltet, ist ein Restore Mode Pflicht.

5. Rechner-eingeschaltet-Erkennung

Ich habe ein SATA-Kabel modifiziert, einen Optokoppler angeschlossen und als Eingang verwendet. Hier muss vielleicht jeder seine



So werden später alle Bauteile verbunden.

© Copyright by Maker Media GmbH.

Lösung selbst finden. Ein Optokoppler ist nicht immer sinnvoll oder nötig (erst recht nicht auf jedem Relaismodul).

Alternativ kann auch ein simpler Spannungsteiler den Zustand erfassen.

```
binary_sensor:
  - platform: gpio
    id: pc_an
    name: "Office PC"
    pin:
      number: D5
      mode: INPUT
      inverted: true
    device_class: power
```

Falls eine Hardwareschaltung nicht möglich/gewollt ist: In Home Assistant kann auch die FritzBox oder die meisten anderen Router eingebunden werden. Sie liefern dann für jedes Gerät einen Zustand, ob das WLAN/LAN gerade aktiv ist, welches dann den Zustand `home` annimmt. Für einen Desktop-PC bevorzuge ich ein echtes Signal.

6. AC-Dimmer

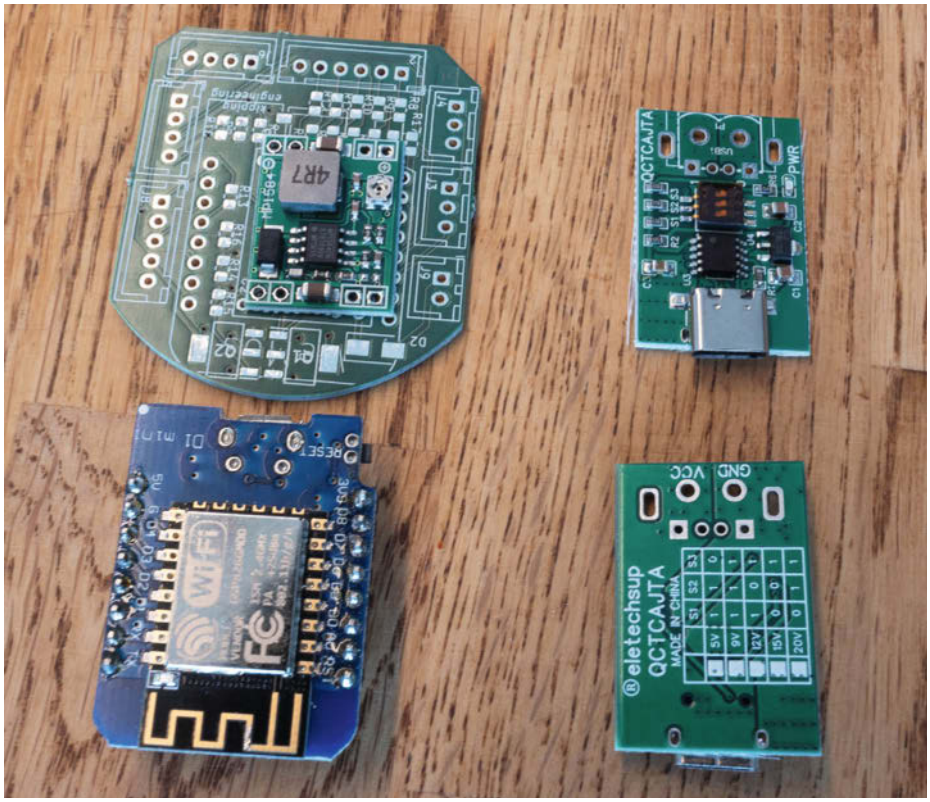
Ich verwende die Module von Robotdyn, es sollte aber jedes Modul funktionieren, das den Nulldurchgang des AC-Signals bestimmen kann und eine Ansteuerleitung besitzt.

```
output:
  - platform: ac_dimmer
    id: dimmer1
    gate_pin: D7
    zero_cross_pin:
      number: D6
      mode:
        input: true
    inverted: yes
```

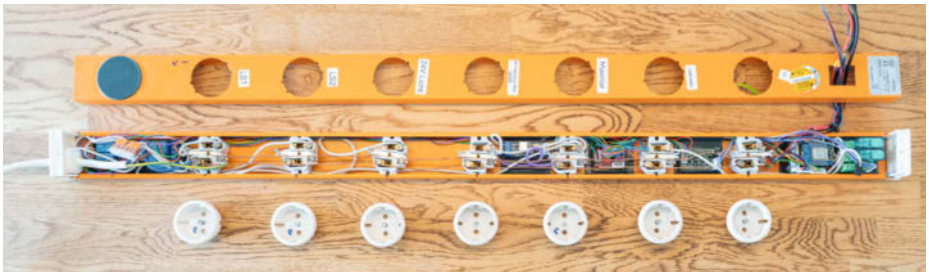
Achtung: das Dimmen erfolgt Interrupt-gesteuert, das ist auch gut so. Es bedeutet: zu jedem Nulldurchgang wird der normale Prozessorablauf angehalten und das Ansteuersignal für den SC-Dimmer-Ausgang eingestellt (Phasenabschnitts-Steuerung). Allerdings steigt die Interrupt-Last deutlich. Signalschnittstellen, die mit Software emuliert werden, wie z.B. die Dallas Temperatursensoren, leiden deutlich in der Übertragungsrate. Die Ansteuerung wird weiter unten bei der Temperaturreglung beschrieben.

7. Temperatursensoren

Ich verwende für das Projekt zwei Dallas DS18B20-Sensoren. In der Vergangenheit war das eine robuste Lösung mit wenigen Kabeln, da nahezu beliebig viele Sensoren an einer Leitung hängen können. Leider sind die meisten „Zehn-Stück-Ein-Euro"-Sensoren nachgemacht. Auch die funktionieren meistens, aber nicht immer!



Der Controller mit dem selbst entwickelte Adapterboard, DC/DC-Platine und USB-C Spannungswandler.

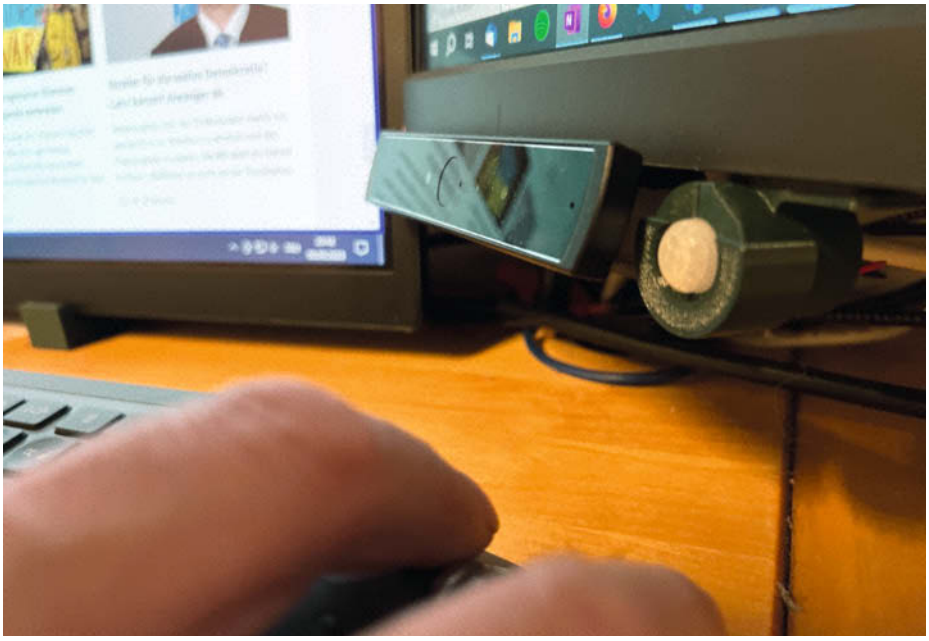


All diese Bauteile passen in das Originalgehäuse der Steckdosenleiste.

Template

```
- platform: template
  id: bs_rel_pc_supply
  name: "Office PC Supply"
  optimistic: true
  restore_mode: RESTORE_DEFAULT_ON
  turn_on_action:
    then:
      - switch.turn_off: gpio_rset_bs_rel2
      - delay: 20ms
      - switch.turn_on: gpio_set_bs_rel2
      - delay: 100ms
      - switch.turn_off: gpio_set_bs_rel2

  turn_off_action:
    then:
      - switch.turn_off: gpio_set_bs_rel2
      - delay: 20ms
      - switch.turn_on: gpio_rset_bs_rel2
      - delay: 100ms
      - switch.turn_off: gpio_rset_bs_rel2
```

© Copyright by Maker Media GmbH.



Ein Sensor schaut einem auf die Finger.

```
dallas:
  - pin: D4
    update_interval: 2sec

sensor:
  - platform: dallas
    address: 0xed3c02f649eb1828
    name: "Office Temperatur Umgebung
    Rohwert"
    id: office_desk_temp
    filters:
      - sliding_window_moving_average:
          window_size: 4
          send_every: 2
      - or:
          - delta: 0.2
          - throttle: 10min

  - platform: dallas
    address: 0x203c2af648f5eb28
    name: "Office Temperatur IR
    Heizung Rohwert"
    id: ir_surface_temp
    filters:
      - sliding_window_moving_average:
          window_size: 2
          send_every: 2
      - or:
          - delta: 0.2
          - throttle: 10min
```

Ein Sensor misst die Oberflächentemperatur der IR-Heizung, der andere die Umgebungstemperatur. Wegen des Interrupt-Problems wird sehr häufig ausgelesen, damit häufiger Signale ankommen.

Anschließend filtere ich die Signale mit Mittelwerten. Damit wird der Home Assistant nicht überflutet. Signale werden erst an den Server gesendet, wenn 10 Minuten abgelaufen sind oder sich die Temperatur um mehr als 0,2 °C geändert hat. Die Adresse jedes einzelnen Dallas-Sensors ist anders. Die eigenen werden im Logging von ESPHome beim Starten angezeigt.

Alle Schnittstellen sind jetzt beisammen. Nun kommt der interessantere Teil – die Automatisierung.

Abwesenheit

```
binary_sensor:
  - platform: gpio
    name: "Office Anwesenheit"
    …
    on_press:
      then:
        - lambda: if(id(pc_an).state) {id(office_speaker).turn_on()};
    on_release:
      then:
        - switch.turn_off : office_speaker
```

Ich bevorzuge es, wenn Dienste und Daten möglichst auf dem lokalen Gerät liegen. Der Monitor wird mit dem PC eingeschaltet und nicht während des PC-Betriebs abgeschaltet. Ein Standby des Betriebssystems reicht mir und Windows 10 mag es nicht, wenn Monitore verschwinden. Dazu wird der PC_an-Sensor erweitert.

```
binary_sensor:
  - platform: gpio
    id: pc_an
 …
    on_press:
      then:
        - logger.log:
            format: "PC an!"
            level: INFO
        - switch.turn_on : sw_display
    on_release:
      then:
        - switch.turn_off : office_
          speaker
        - switch.turn_off : sw_display
        - delay: 50ms
        - switch.turn_on : sw_display
        - delay: 50ms
        - switch.turn_off : sw_display
        - delay: 50ms
        - switch.turn_on : sw_display
        - delay: 50ms
        - switch.turn_off : sw_display
```

Die `on_press/on_release` Einträge starten Aktionen, wenn der PC an oder aus geht. Die komischen `on_release` Einträge werden bei Relaiskiller erklärt.

Lautsprecher bei Abwesenheit ausschalten

Ich platziere den Anwesenheitssensor so, dass er meine Hände beobachtet. Wer nicht an der Tastatur ist, braucht auch keinen Lautsprecher. Die Parameter sind wie immer ein Kompromiss: Wenn maximal Energie gespart werden soll, reichen sicher 5 Minuten. Allerdings schaue ich auch mal einigermaßen bewegungslos einen Film. Dann sind eher 20 Minuten angesagt.

Das Ein- und Ausschalten der Lautsprecher bei Abwesenheit erfolgt im Sensor der Anwesenheit selbst, die Konfiguration wird wie im Kasten Abwesenheit erweitert.

Beim Einschalten (`on_press`) wird der Lautsprecher eingeschaltet, jedoch nur, wenn der PC auch an ist. Das ist ein Stück C-Code, der mit dem Key `lambda` eingeleitet wird. So kann ich beliebig komplexe Programme in C schreiben. Das ist oft viel einfacher als die YAML-Syntax. Es wäre aber genauso möglich, in YAML eine condition hinzuzufügen. Wenn die Anwesenheit nach Entprellung ausgeschaltet wird, gehen sofort die Lautsprecher aus.

© Copyright by Maker Media GmbH.

Der Relaistod

Sie sterben. Und natürlich die billigen zuerst. Aber ich habe auch wirklich viele recht teure Markenrelais auf dem Gewissen.

Das Problem sind die modernen Schaltnetzteile. Ich habe drei der fiesen Sorte an meinem Arbeitsplatz. Wird ein Schaltnetzteil nach längerer Zeit mit Strom versorgt, entstehen Stromspitzen bis 40 Ampere. Das ist alles im erlaubten Bereich, wenn man halt nur eins gleichzeitig einschaltet.

Bei mir sind zwei Nubert-Aktivlautsprecher und ein OLED-Monitor-Netzteil verdammt aggressiv. Das führt beim Einschalten mittels Schaltsteckdose zuverlässig dazu, dass die Hauptsicherung fliegt. Daher werden die Relais immer zeitversetzt aktiviert. Das verhindert das Dunkelzimmer, bedarf aber auch vieler Relais.

Soweit so gut, die Sicherung bleibt drin, aber nach dem zweiten, zehnten oder hundertsten Einschalten verschweißt das Relais. Der hohe Strom beim Einschalten verklebt hier die Kontakte. Meine Abhilfe dafür ist eine eigene Putzfunktion. Relais sind in der Regel so aufgebaut, dass sie beim Schalten leicht an den Kontakten reiben, um Oxidationen zu entfernen. Wenn es verschweißt ist, hilft der Hammer noch ein paar Mal als Trennelement, aber gut ist das nicht. Also die Putzfunktion: Sie schaltet das Relais mehrfach hin und her, allerdings erst beim Abschalten. In dem Fall sind die Schaltnetzteile und deren Kondensatoren aufgeladen, Stromspitzen entstehen in der Regel bei kurzzeitigem Abschalten nicht mehr. Jedes Abschalten putzt die Kontakte, seitdem habe ich kein Verschweißen mehr beobachtet.

Das bistabile Relais

Das bistabile Relais hat die schöne Eigenschaft, dass es keinen Strom braucht, um den Zustand zu halten. Ein Relais braucht geschaltet ca. 0,5 Watt und sollte so verschaltet sein, dass es möglichst wenig Zeit bestromt ist. Für Stromkreise, die fast die Hälfte der Zeit an sind, kann ein bistabiles Relais helfen. Ich gebe zu, das ist vermutlich Liebhaberei, denn der Aufwand eines zu benutzten, ist nicht gerade klein. Bistabile Relais werden mit großem Strom eingeschaltet und mit kleinem Strom abgeschaltet. Ich habe mir dafür eine eigene Schaltung gebaut, samt PCB, die diese Aufgabe übernimmt. Es wird also ein paar Jahre dauern, bis die Energieeinsparung die Hardware amortisiert. Falls überhaupt. Ist aber trotzdem cool. Schaltplan und PCB-Layout sind für Nachbauwillige über den Link in der Kurzinfo zu finden.

Temperaturregelung

Um Heizkosten zu sparen, muss die Durchschnittstemperatur in der Wohnung runter.

Infrarot-Paneele

```
number:
  - platform: template
    name: "Office IR Heizung Obere Grenze"
    id: heater_max_temp
    unit_of_measurement: "°C"
    optimistic: true
    restore_value: true
    min_value: 10
    max_value: 30
    step: 1

  - platform: template
    name: "Office IR Heizung Untere Grenze"
    id: heater_min_temp
    unit_of_measurement: "°C"
    optimistic: true
    restore_value: true
    min_value: 10
    max_value: 30
    step: 1


sensor:
  - platform: template
    name: "Office Heizung Ansteuerung"
    id: heater_value
    icon: "mdi:valve"
    lambda: |-
      if (isnan(id(ir_surface_temp).state) ||
        ((id(heater_max_temp).state - id(heater_min_temp).state) == 0))
      {
        ESP_LOGI("main", "Invalid input %f", id(ir_surface_temp).state);
        return id(heater_value).state;
      } else {
        float new_val;
        new_val = (id(heater_max_temp).state - id( office_desk_temp).state
        ) * 100 /
                  (id(heater_max_temp).state  - id(heater_min_temp).state
                  );
        if (new_val > 100) new_val = 100;

        // nur heizen, wenn PC an und jemand da:
        if (new_val < 0 || !id(pc_an).state || !id(gpio_presence).state )
        new_val = 0;

        id(dimmer1).set_level(new_val / 100);
        return new_val;
      }
    unit_of_measurement: "%"
    update_interval: 6s
```

Mit einem Temperatursensor auf dem Heizmodul wird auch das Infrarotpanel smart.

© Copyright by Maker Media GmbH.

Template-Sensor

```
- platform: template
  name: "Office AC Leistung"
  id: power_ac_sum
  icon: "mdi:power"
  lambda: |-
    float power_sum = 0;
    if (id(sw_display).state) {
      power_sum += ${display_on_power};
    }
    if (id(pc_an).state) {
      power_sum += ${pc_on_power};
    }
    if (id(office_speaker).state) {
      power_sum += ${speaker_on_power};
    }
    power_sum += id(heater_value).state / 100 * ${irheat_max_power} ;
    return power_sum;
  unit_of_measurement: "W"
  update_interval: 20s
```

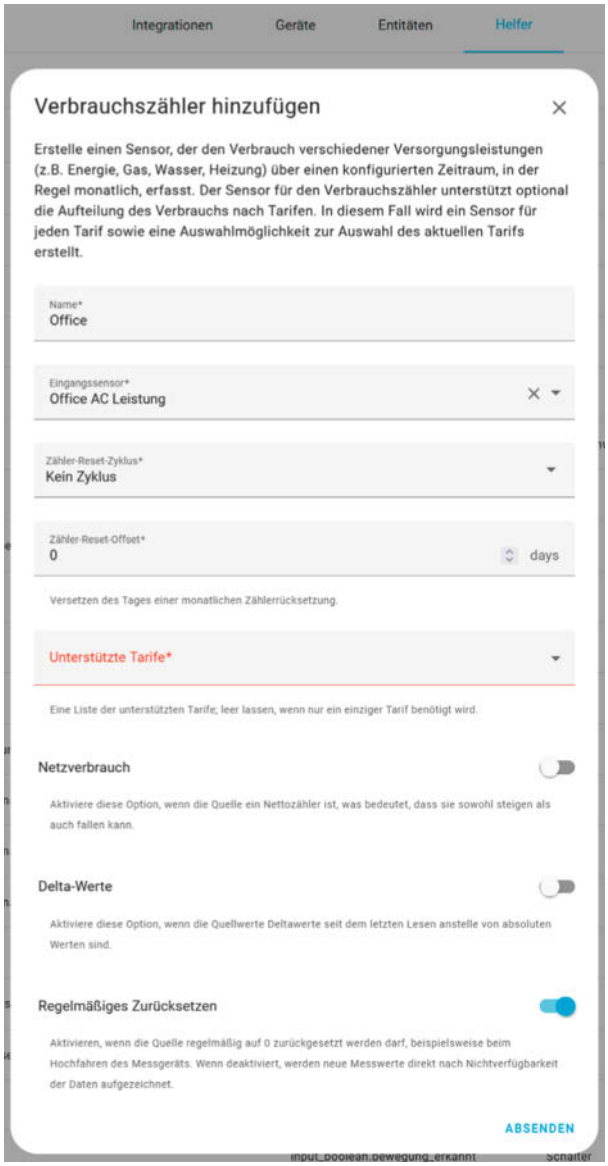
Das geht am einfachsten, indem man die Heizung abschaltet. Frieren ist total cool, aber sehr unangenehm, daher helfen schnelle Heizungen, die nur mich, aber nicht die Wohnung aufwärmen.

Infrarot-Paneele unter dem Schreibtisch bewirken Wunder. Nur wird es ohne Regelung manchmal ein bisschen zu warm. Ich habe dafür eine Overkill-Regelung gebaut. So fancy muss sie gar nicht sein, es hilft auf jeden Fall, wenn man die Leistung regeln kann. Der Code dafür ist im Kasten Infrarot-Paneele. Was bei mir nie richtig geklappt hat, ist eine schlichte Temperaturregelung mit einem Sensor. Denn wo soll der hin? Unter den Schreibtisch, auf den Schreibtisch, es passt einfach nie.

Die einfache Lösung: Sobald jemand am Platz ist, wird die Heizung eingeschaltet. Die Leistung hängt von der Umgebungstemperatur ab, damit es in der Übergangszeit nicht zu warm wird. Anstelle eines AC-Dimmers bietet ESPHome eine SlowPWM mittels Relais, also Takten im Minutentakt. Das funktioniert bei mir in der Küche prima, klackt halt im Minutentakt.

Code: Wir bauen einen Sensor, der die Ansteuerung in Prozent ausgibt in Abhängigkeit von der Umgebungstemperatur. Die Funktion habe ich wieder direkt als Code implementiert:

Die `Number`-Einträge tauchen in Home Assistant auf. Damit kann ich den Regelbereich einstellen. Bei der maximalen Temperatur ist die Heizung aus, bei der minimalen Temperatur voll an.



Der Verbrauchshelfer überwacht in Home Assistant den Stromverbrauch.

Das Template prüft auf korrekte Eingangssignale (Teilung durch Null muss selbst abgefangen werden) und berechnet dann einen Dreisatz. Das `set_level` setzt den AC-Dimmer entsprechend.

Geheizt werden soll natürlich nur, wenn jemand am Rechner sitzt, also der PC an ist und auch jemand davor sitzt. Dafür wird der Dimmerwert auf 0 gesetzt, wenn die Zustände nicht passen.

Oder die Overengineering-Lösung: In Abhängigkeit der Umgebungs- oder mittleren Wohnungstemperatur will ich die Strahlungsleistung einregeln. Denn auch die Platte braucht etwas Zeit, bis sie Wärme abgibt, das ist der Teil mit Overengineering. Ein Kapton-Klebeband hält den Sensor auf der Platte.

Mit dem Temperaturwert des Sensors kann die Heizung geregelt werden. Der Link zum Code-Download findet sich in der Kurzinfo, hier würde es den Rahmen sprengen.

Stromverbrauch ermitteln

Hier wird nicht gemessen, sondern nur gut geschätzt. Eine Messsteckdose ist natürlich genauer, aber auch sie braucht ja Strom. Mir reicht es schon, einen groben Wert zu wissen, um die Statistik in Home Assistant zu befüllen.

Dazu lege ich wieder Template-Sensoren an, die anhand der Zustände die aktuelle Leistung ausgeben. Der Code ist im Kasten Template-Sensor.

Den Stromverbrauch ermittele ich mit einer Messsteckdose und hoffe einfach, dass die Leistung meiner Geräte nicht zu stark schwankt. Ich hinterlege die Daten am Anfang der YAML als `substitution`, wegen der Trennung von Code und Daten:

```
substitutions:
  presence_debounce_time: 30min
  irheat_max_power: "260"
  pc_on_power: "250"
  speaker_on_power: "20"
  display_on_power: "80"
```

Jetzt muss nur noch integriert werden, damit aus der Leistung dann kWh werden. Home Assistant erwartet im Energiemanagement Signale in kWh, um daraus Statistiken zu erstellen.

Zum aktuellen Zeitpunkt ist das Arduino-Framework zu mächtig geworden. Die benötigten Komponenten passen nicht mehr auf den ESP8266. Das ist kein Problem, ich weiche auf Home Assistant aus. In Home Assistant wird ein Helfer Verbrauchszähler hinzugefügt, der die Integration übernimmt:

Der Winter kann jetzt wieder kommen. Arbeiten am Rechner ist nun komfortabel möglich – bei minimalem Energieverbrauch. Gutes Gelingen! *—das*

© Copyright by Maker Media GmbH.

Make:

JETZT IM ABO GÜNSTIGER LESEN

2× Make testen mit über 30 % Rabatt

Ihre Vorteile im Plus-Paket:

- ✓ Als **Heft** und
- ✓ **Digital** im Browser, als PDF oder in der App
- ✓ Zugriff auf **Online-Artikel-Archiv**
- ✓ **Geschenk,** z. B. Make: Tasse

Für nur 19,40 € statt ~~27 €~~

Jetzt bestellen:
make-magazin.de/miniabo

infoservice@maker-media.de +49 541/80 009 125 make-magazin.de/miniabo

© Copyright by Maker Media GmbH

POV-Display aus CD-Spieler

Mit einem Arduino nano, ESP-01s und speziellem PCB wird es möglich: eine Anzeige für Uhrzeit oder Wetter. Das alles im Eigenbau.

von Daniel Schwabe

https://github.com/lhm0/rotating_display

© Copyright by Maker Media GmbH.

POV steht für persistence of vision (auf deutsch in etwa „Beständigkeit der Sicht“). POV-Displays sind Displays, die die Trägheit des Auges ausnutzen und mit wenigen LEDs durch Bewegung ein Bild erzeugen. Bekannt ist der Effekt besonders durch Ventilatoren, auf deren drehenden Rotoren Text durchläuft.

So ein Display hat der User des YT-Channels „Maker Electronics Projects“ mithilfe eines CD-Players realisiert. Der alte Discman-Motor treibt in diesem Fall einen Rotor an, auf dem LEDs angebracht sind. Diese zeigen bei schnellem Drehen z.B. die Uhrzeit an.

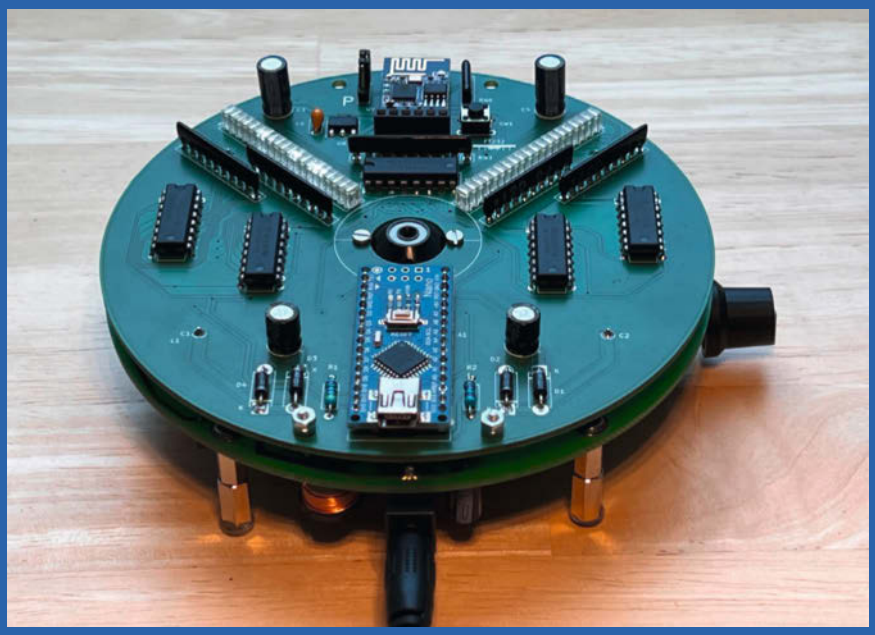
Auf der GitHub-Page des Projektes ist alles zu finden, was man braucht, um dieses Projekt nachzubauen. Neben einer ausführlichen Anleitung, Stückliste und erforderlichen Daten sind auch detaillierte Schaltpläne für das Display einsehbar. Außerdem werden sämtliche Funktionen des Displays und die Bedienung erklärt.

Auf einem selbstgefertigten runden PCB sind alle Display-Bauteile aufgebracht. Ein Arduino nano bedient die insgesamt 20 LEDs und ein ESP-01s erstellt die angezeigten Inhalte und verbindet das POV-Display mit dem Internet. Über eine Netzwerkverbindung lässt sich das Display einstellen. Neben der Uhrzeit und Wetterdaten, die live aus dem Internet abgerufen werden, kann man auch Bilder über ein Webinterface auf das Gerät hochladen, die dann angezeigt werden. Das geschieht über eine benutzerfreundliche Bedienoberfläche, die mit der ESP-Library ESPAsyncWebServer umgesetzt wurde. Für Funktionen bezüglich der Wetteranzeige ist eine Verbindung mit einem Wetterdienst vorausgesetzt. Insgesamt hat das Display etwa eine Auflösung von 110 auf 110 Pixeln. Neben der Display-Scheibe gibt es noch das Netzteil-PCB, das die gleichen Dimensionen wie das Display hat und unter diesem liegt.

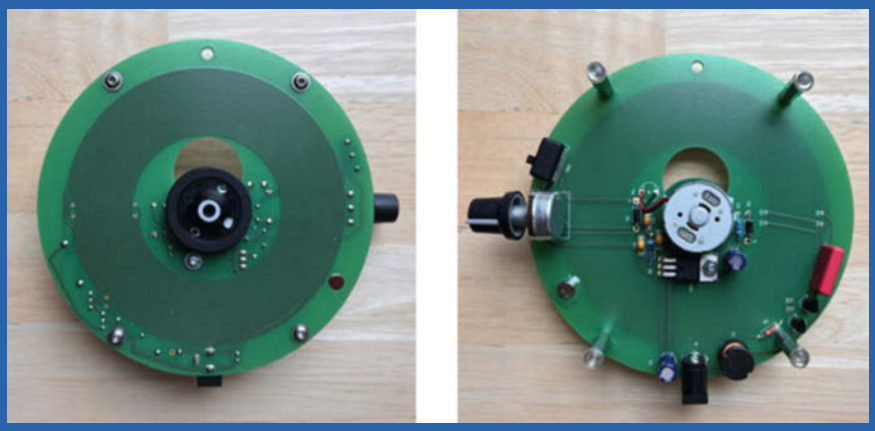
Mithilfe eines Potentiometers wird der Motor gesteuert. Alle Bauteile des POV-Displays sind so angeordnet, dass der Rotor ausbalanciert ist. Das ist wichtig, weil die sich drehende Scheibe sonst nicht rund läuft und der Effekt, dass es für das menschliche Auge wie ein durchgehendes Display aussieht, nicht funktioniert. Deshalb sind auch einige Schrauben ausschließlich als Gewichte zum Ausgleich verbaut. Des Weiteren müssen die LEDs genau im richtigen Moment angesteuert werden, um den richtigen Display-Inhalt darzustellen. Dazu hat der Entwickler des Projektes auf der Github-Seite eine ausführliche Erklärung des technischen Vorganges veröffentlicht. Das Projekt findet man unter folgendem Link: *—das*

▶ **make-magazin.de/xwk7**

Das POV-Display gibt Informationen über die Uhrzeit und das Wetter.



Oben befindet sich das Display, darunter die Stromversorgung.



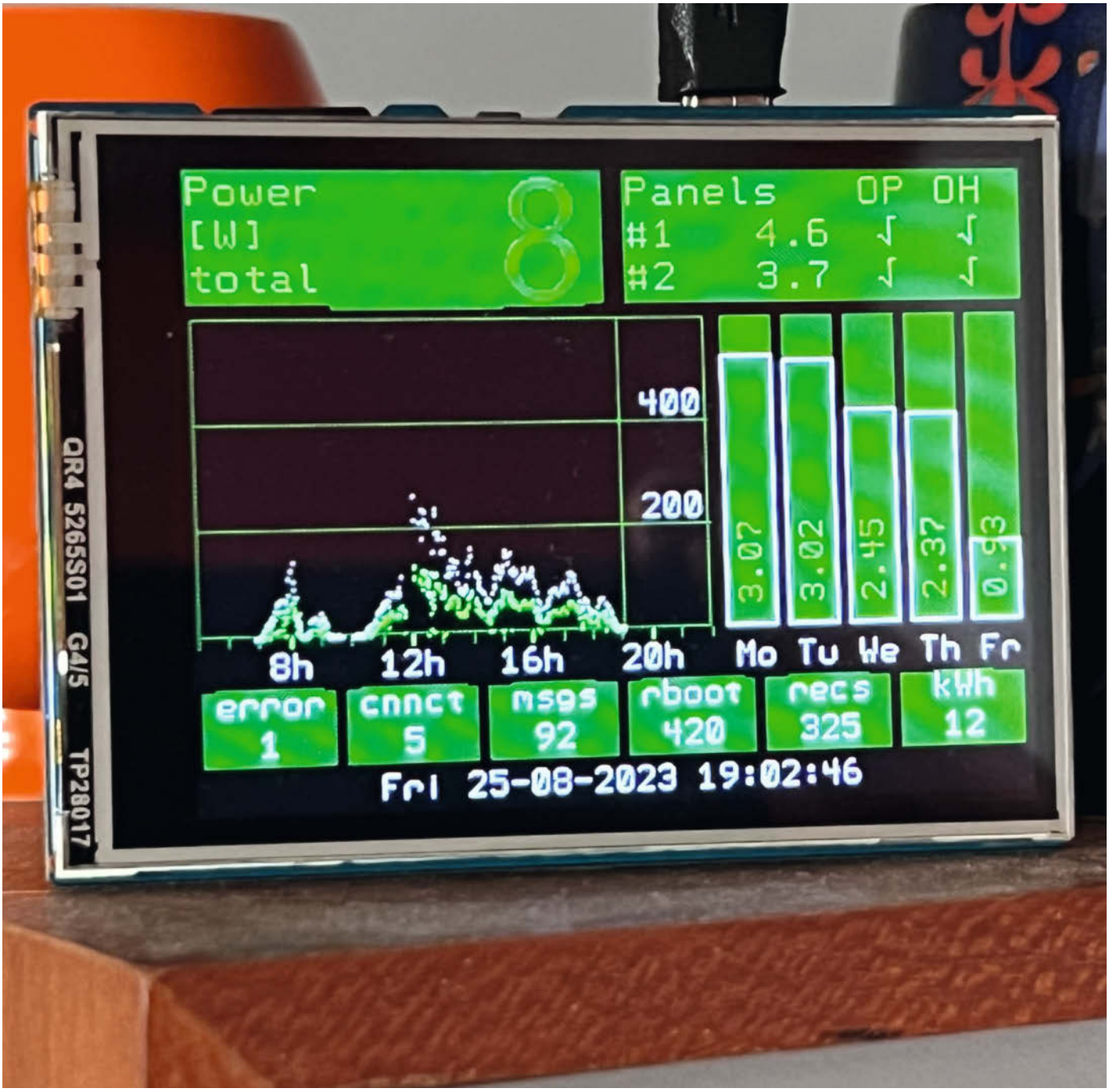
Der untere PCB für die Stromversorgung beherbergt auch den CD-Motor.

© Copyright by Maker Media GmbH.

Das Balkonkraftwerk im Blick

Das Balkonkraftwerk ist montiert und nun stellt sich die Frage: Wie viel Energie wird wann erzeugt? Wann lohnt es sich, den Geschirrspüler anzustellen? Wir stellen ein praktisches Display für das Smarthome vor, welches diese und ähnliche Probleme löst.

von Thomas Euler



© Copyright by Maker Media GmbH.

Fast alle Wechselrichter haben einen WLAN-Zugang, über den sie ein paar Daten preisgeben. Also Smartphone zücken, einloggen, kleine Grafiken anschauen … das wird schnell umständlich. Und Geräte mit proprietärer Firmware möchte man vielleicht auch gar nicht ins heimische WLAN lassen. So entstand der Wunsch nach einem einfachen Display, das immer die wichtigsten Daten auf einen Blick anzeigt.

In dem hier vorgestellten Projekt holt sich ein Raspberry Pi Pico W die entsprechenden Solardaten vom Homeserver und zeigt sie auf einem aufgesteckten TFT-Display an. Auf dem Pico läuft die WebMite-Firmware, die MMBasic mit einer Reihe von Netzwerkfunktionen ausstattet. Das Programm demonstriert, wie einfach sich drahtlose Überwachungsaufgaben mit einem modernen BASIC realisieren lassen.

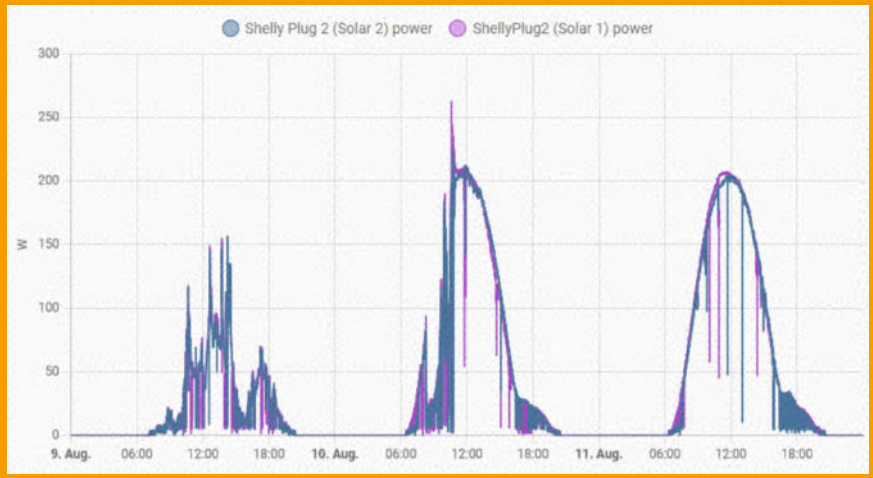
Das Balkonkraftwerk besteht aus zwei Panels, die über zwei 300W-Wechselrichter die erlaubten 600 Watt liefern. Da die Module senkrecht am Balkongeländer montiert sind, wurden wegen der besseren Ausbeute 400Wp-(Watt peak)-Panels gewählt. Die Module sind auf verschiedenen Stockwerken montiert und speisen in zwei separate Außensteckdosen ein. Um den erzeugten Strom zu messen, befindet sich zwischen Wechselrichter und Hausnetz – wetterfest verpackt – jeweils ein Zwischenstecker vom Typ Shelly Plug S. Diese melden die aktuell erzeugte Leistung über WLAN an einen Raspberry Pi, auf dem Home Assistant läuft. Der Home Assistant kann standardmäßig viele Geräte einbinden, darunter auch die Shelly Plugs. Die Software enthält auch einen integrierten MQTT-Broker. Ein kurzes Home Assistant-Skript (siehe Link) fragt regelmäßig die Solarleistung ab und bietet diese Daten als MQTT-Dienst an. Das MMBasic-Programm auf dem Pico abonniert diesen Dienst und wird so ständig mit frischen Daten vom Balkonkraftwerk versorgt.

Das Display am Gerät zeigt die aktuelle Leistung an und stellt den Leistungsverlauf über den Tag und als Summe über die letzten fünf Tage grafisch dar. Da das Programm alle Daten auch in dem Flashspeicher des Pico sichert, schadet es nicht, wenn der Pico einmal vom Strom getrennt wird. Ein Watchdog sorgt dafür, dass das Programm normal läuft und die Netzwerkverbindung nicht verloren geht. Im Fehlerfall startet der Watchdog den Pico automatisch neu. Das Display hat einen Steckplatz für eine SD-Speicherkarte; ist eine solche eingesteckt, speichert der Pico alle paar Minuten einen Screenshot.

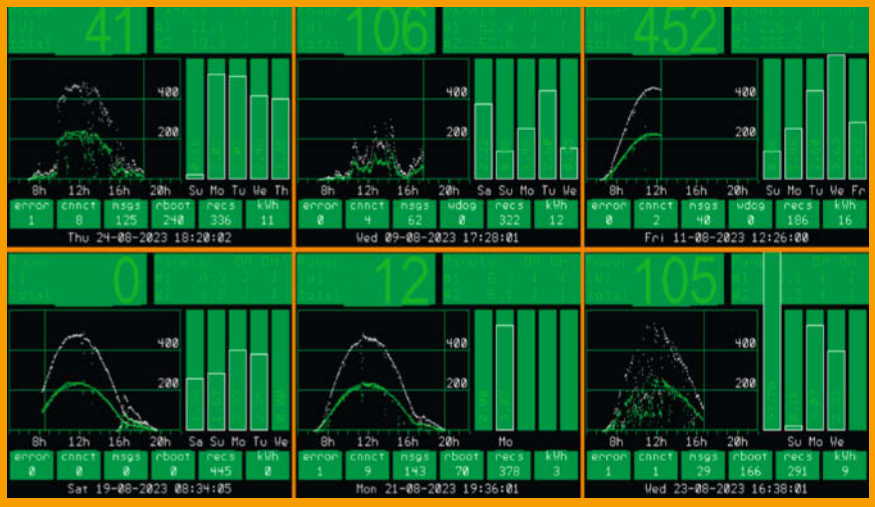
Eine wünschenswerte Erweiterung wäre die Anzeige des aktuellen Strombedarfs im Haushalt, um zu sehen, wie viel von der eigenen Stromernte tatsächlich verbraucht wird. Mit einem entsprechenden Sensor im Sicherungskasten wäre dies leicht zu realisieren. —*caw*

▶ **github.com/teuler/solarmqtt**

Die beiden linken Panels gehören zur überwachten Solaranlage.



Die Solarerträge in Homeassistant



Beispiele der Anzeige an verschieden ertragreichen Tagen

© Copyright by Maker Media GmbH.

DIY-Holoprojektor

Objekte wie in Star Wars räumlich zu projizieren, ist faszinierend. Mit sogenannten Volumendisplays ist das möglich, die sind jedoch teuer. Wie man sich mit einem alten Beamer und etwas Elektronik selbst solch ein Gerät baut, zeigt dieses Projekt.

von Markus Mierse

© Copyright by Maker Media GmbH.

Es gibt wohl kaum einen Star-Wars-Fan, der nicht davon träumt, über einen Holoprojektor quer durch die Galaxis zu kommunizieren oder geheime 3D-Baupläne aus allen erdenklichen Winkeln betrachten zu können – und das ganz ohne Spezialbrille. Volumendisplays kommen optisch ziemlich nah an diesen Wunsch heran und können virtuelle Objekte innerhalb gewisser Grenzen dreidimensional im Raum schweben lassen. Da ich mich generell für ungewöhnliche Displays interessiere, konnte ich nicht widerstehen und habe selbst eine erschwingliche DIY-Variante entwickelt, die Raumschiffe und andere 3D-Modelle auf bis zu 160 × 100 × 30 mm monochrom darstellen kann – sogar animiert.

Volumendisplays können sehr unterschiedlich aufgebaut sein. Meine Version verwendet einen Projektor, der zeitlich abgestimmt Schichten eines 3D-Objekts auf einer oszillierenden Projektionsfläche (Swept-Volume) abbildet. Da sich diese schnell auf und ab bewegt, verschmelzen die einzelnen Ebenen zu einer räumlichen Darstellung. Um einen möglichst flimmerfreien Eindruck zu erzeugen, benötigt man eine Bewegungsfrequenz von etwa 30Hz (Volume-Refresh).

Modifizierter Projektor

Die Auflösung des Modells bzw. die Anzahl der einzelnen Schichten ergibt sich aus der Bildwiederholfrequenz des verwendeten Beamers. Um nicht in einen teuren High-Speed-Projektor investieren zu müssen, suchte ich nach einem erschwinglichen DLP-Projektor (LCD-Projektoren sind zu langsam). Meine Wahl fiel schließlich auf das BenQ-Modell W1070, das Eingangssignale mit bis zu 120Hz verarbeitet und bei einem Volume-Refresh von 30Hz 4 Schichten auf der Projektionsfläche darstellen könnte. Um ihre Anzahl zu verdreifachen, habe ich das RGB-Farbrad aus dem optischen Strahlengang des Beamers entfernt. Dadurch verliert man zwar die Farbdarstellung, gewinnt aber eine höhere Bildwiederholfrequenz – in meinem Fall 360Hz, mit denen sich 3D-Modelle in 12 monochromen Ebenen darstellen lassen.

Bewegung ohne Motor

Als Nächstes musste ich eine Lösung finden die Projektionsfläche mit 30Hz schwingen zu lassen. Die entscheidende Idee kam mir buchstäblich in den Sinn, als ich beim Musikhören auf meine Lautsprecher schaute: Schwingspulen! Denn diese können schnelle und sehr präzise lineare Bewegungen ausführen.

Da es sich bei 30Hz eindeutig um den Bass- bzw. Subwoofer-Bereich handelt, habe ich daraufhin zwei alte Tieftöner geschlachtet und alles bis auf die Spule, die Spinne (Zentriermembran) und die Trägerbauteile entfernt. Übrig geblieben ist ein einfacher Linearhub-Aktor, der sich mithilfe einer Sinusschwingung in Bewegung versetzen lässt.

Flexible Halterung

Eine besondere Herausforderung stellte die Befestigung der oszillierenden Projektionsfläche dar, die sich mit 30Hz möglichst linear und frei im Raum auf und ab bewegen können sollte. Für diese Geschwindigkeit benötigte ich eine Lösung, die wenig wiegt und belastbar ist. Als Aufhängung habe ich daher nachgiebige Mechanismen (engl. compliant mechanism) konstruiert – Strukturen, die aus einem Stück gefertigt sind, sich aber punktuell elastisch bewegen können. Sie sind leicht, kommen ohne weitere mechanische Teile aus und lassen sich gut im 3D-Druckverfahren herstellen. Als Neuling auf dem Gebiet brauchte ich viele Versuche, bis ich eine Form gefunden hatte, die so gut funktionierte, dass sie die Eingangsschwingung effektiv in eine verstärkte Vertikalbewegung umwandelte. Auch musste ich erst lernen, wie entscheidend das Druckmaterial für sich ständig bewegende Strukturen ist. Die ersten Versionen mit PLA und PETG brachen regelmäßig auseinander, da beide Kunststoffe einfach zu spröde sind.

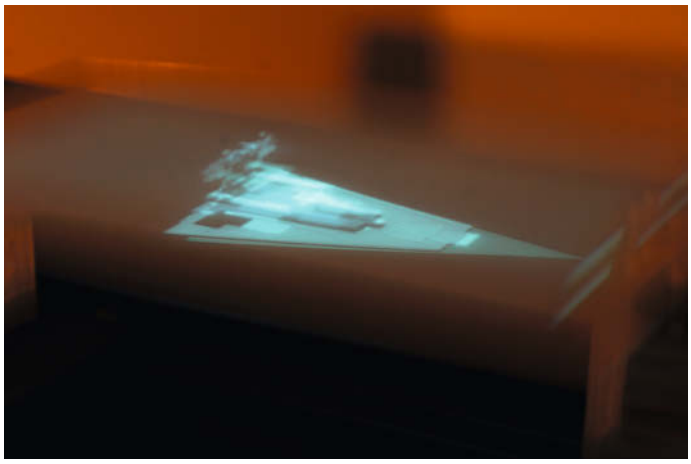
Also musste ich mich nach anderen druckbaren Materialien umsehen und bin nach einiger Recherche auf Polypropylen (PP) gestoßen, das leicht ist, aber auch eine hohe Ermüdungsfestigkeit und eine sehr gute dynamische Belastbarkeit aufweist. Man kennt PP von vielen Lebensmittelverpackungen, bei denen Teile des Deckels beweglich sind (z.B. TicTac-Verschluss). Aus diesem Kunststoff konnte ich Halterungen drucken, die eine dauerhaft bruchsichere Bewegung der Projektionsfläche gewährleisten.

Geschickt ausgeblendet

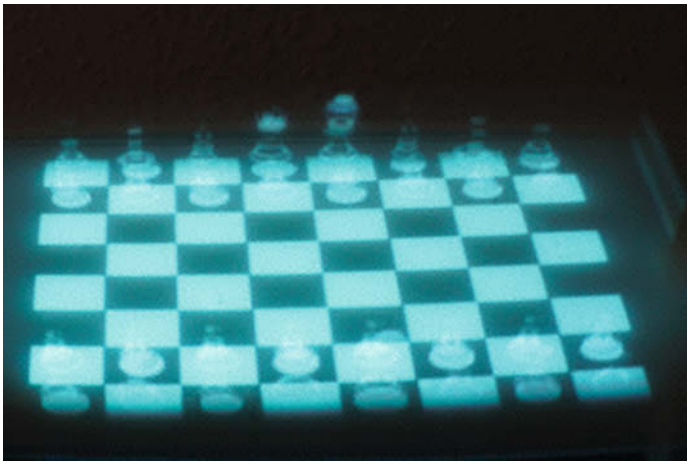
Um Störelemente wie Doppel-Frames aus dem projizierten Bild zu entfernen, die normalerweise für die verbesserte Farbwiedergabe genutzt werden, setzte ich einen LCD-Shutter ein. Solche Shutter kennt man z.B. von den aktiven 3D-Brillen, die bis vor einigen Jahren für Fernsehgeräte angesagt waren. Sie sind mit einem Ein-Pixel-LC-Display vergleichbar und können Inhalte je nach Zustand blockieren oder durchlassen. Die im ms-Bereich schal-

Mehr zum Thema

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

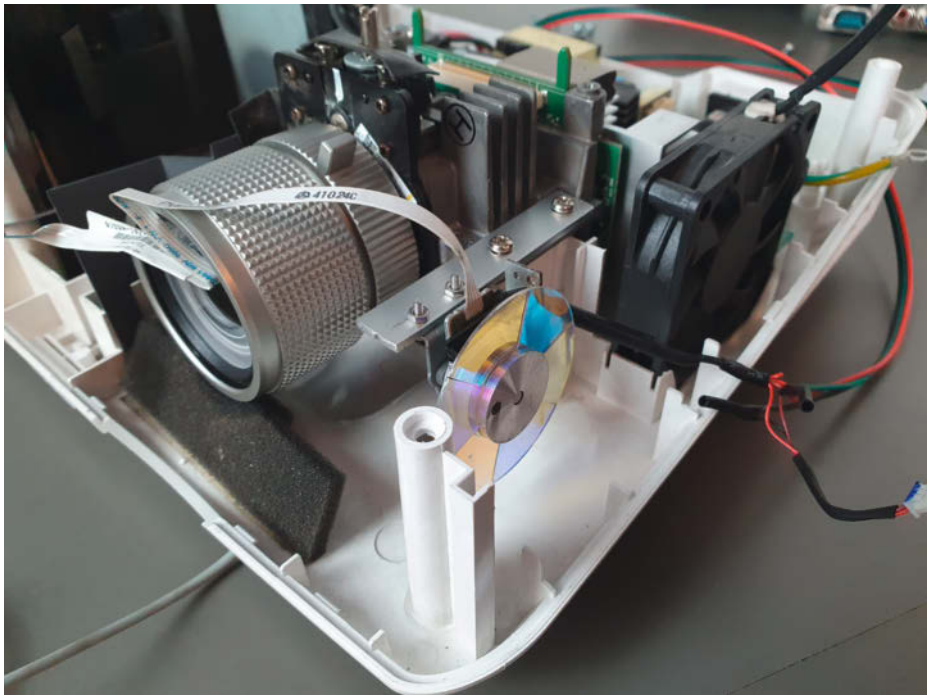


Das Volumendisplay ist wie für Raumschiffe gemacht.



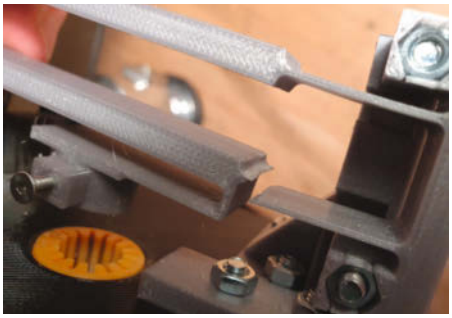
Aber auch eine Partie Holo-Schach sieht verlockend aus.

© Copyright by Maker Media GmbH.



Das Farbrad sitzt nicht mehr vor der Linse, wird aber noch benötigt.

Zwei Tieftöner bringen die Projektionsfläche zum Schwingen.



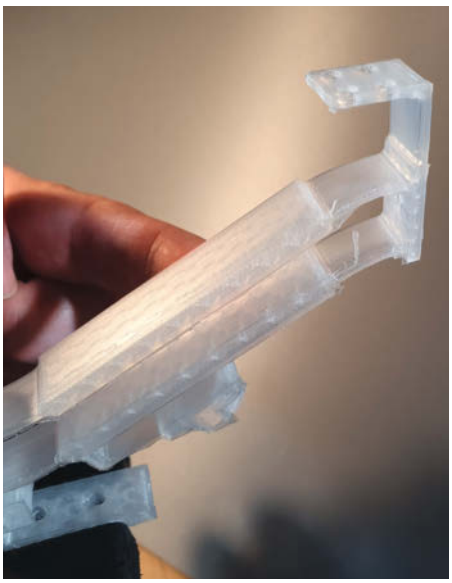
3D-gedruckte Teile aus PLA und PETG haben nicht lange gehalten.

Für den LCD-Shutter habe ich eine aktive 3D-Brille auseinandergenommen.

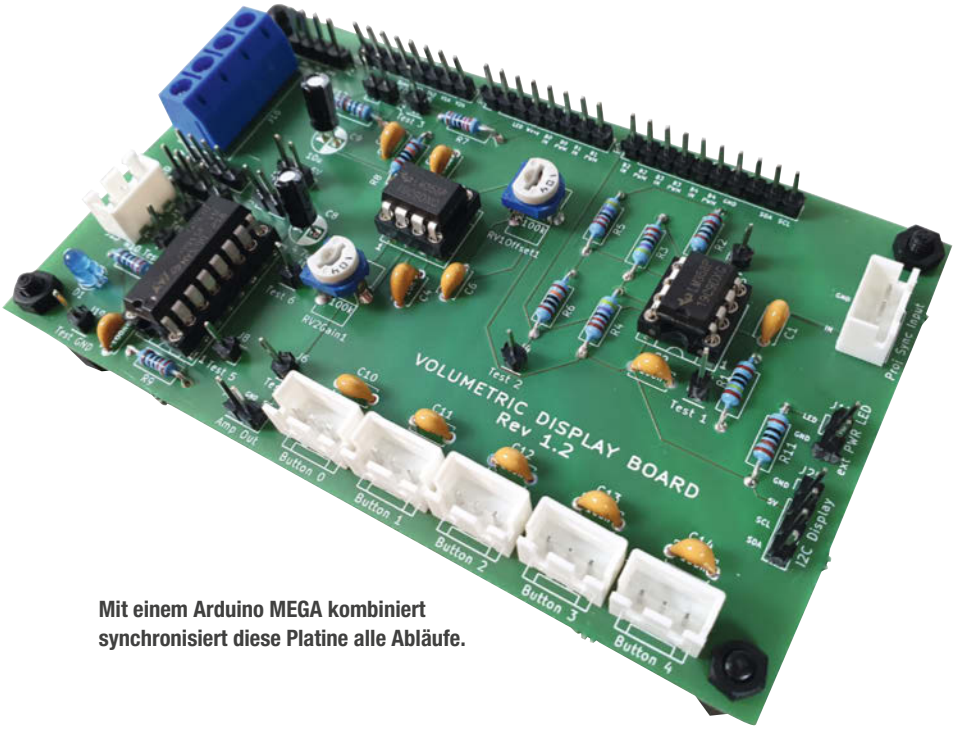
tenden Shutter benötigen eine alternierende Spannung, die ich mithilfe einer H-Brücke (L293) angesteuert habe.

Arduino als Steuereinheit

Die Schaltung, die den Projektor und die Schwingspulen synchronisiert, verwendet einen Arduino MEGA. Als Trigger/Timer habe ich den Sensor des versetzten Farbrads angezapft und werte das Signal über eine Triggerstufe an einem Interrupt-Eingang des Mikrocontrollers aus. Dadurch ist der MEGA in der Lage, eine Sinusschwingung exakt



PP ist für dieses Projekt besser geeignet.

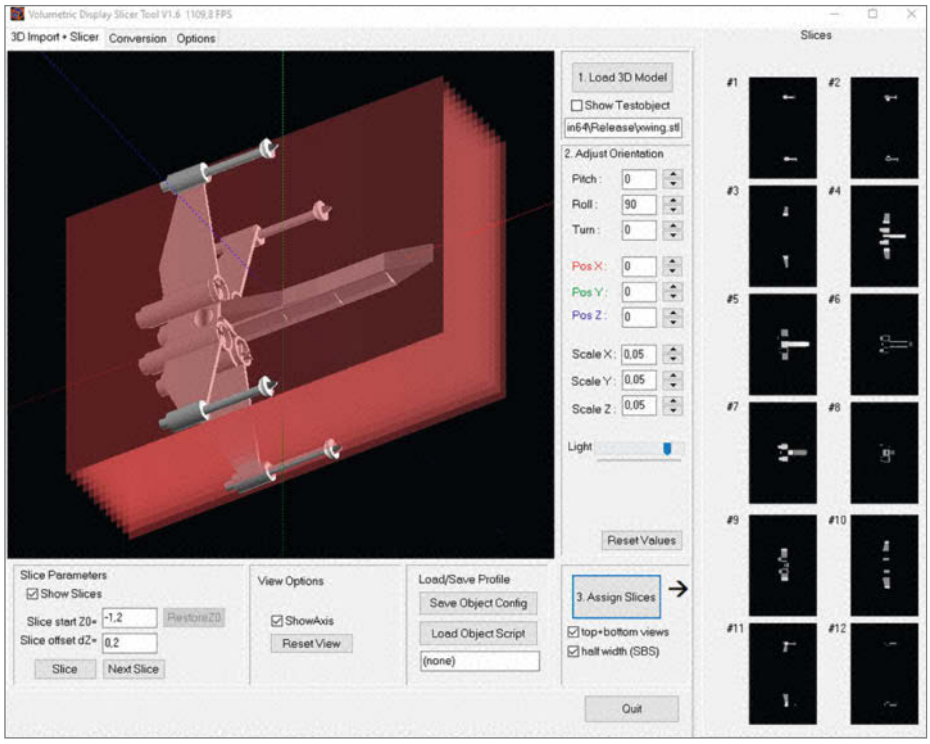


Mit einem Arduino MEGA kombiniert synchronisiert diese Platine alle Abläufe.

© Copyright by Maker Media GmbH.

synchron zu den Projektionsbildern auszugeben und über einen Tiefpass sowie eine Offset-Kompensation an einen einfachen Stereo-Verstärker weiterzuleiten. Die gleiche Interrupt-Routine steuert auch den Shutter. Hierzu kontrolliert der Mikrocontroller den L293-Baustein über zwei I/O-Ports, der die notwendige Spannung am Shutter generiert. Die Elektronik mit den notwendigen Ein- und Ausgangsschaltungen des MEGA habe ich zunächst auf einem Breadboard aufgebaut, getestet und anschließend auf dieser Basis in KiCAD eine eigene Platine entwickelt.

Der MEGA bietet neben den vielen I/O-Ports den Vorteil mehrerer Hardware-UARTs, die ich nutze, um den Projektor über dessen serielle Schnittstelle RS232 (Recomended Standard 232) fernzusteuern. So kann ich den Beamer bequem über den Microcontroller ein- und ausschalten, den Status abfragen oder wichtige Einstellungen vornehmen. Mein Design enthält zudem noch diverse Frontpanel-Elemente (OLED-Display, Taster etc.) und natürlich zahlreiche LEDs.



So lässt sich ein 3D-Objekt leicht in 12 Schichten zerteilen.

Eigene Toolchain

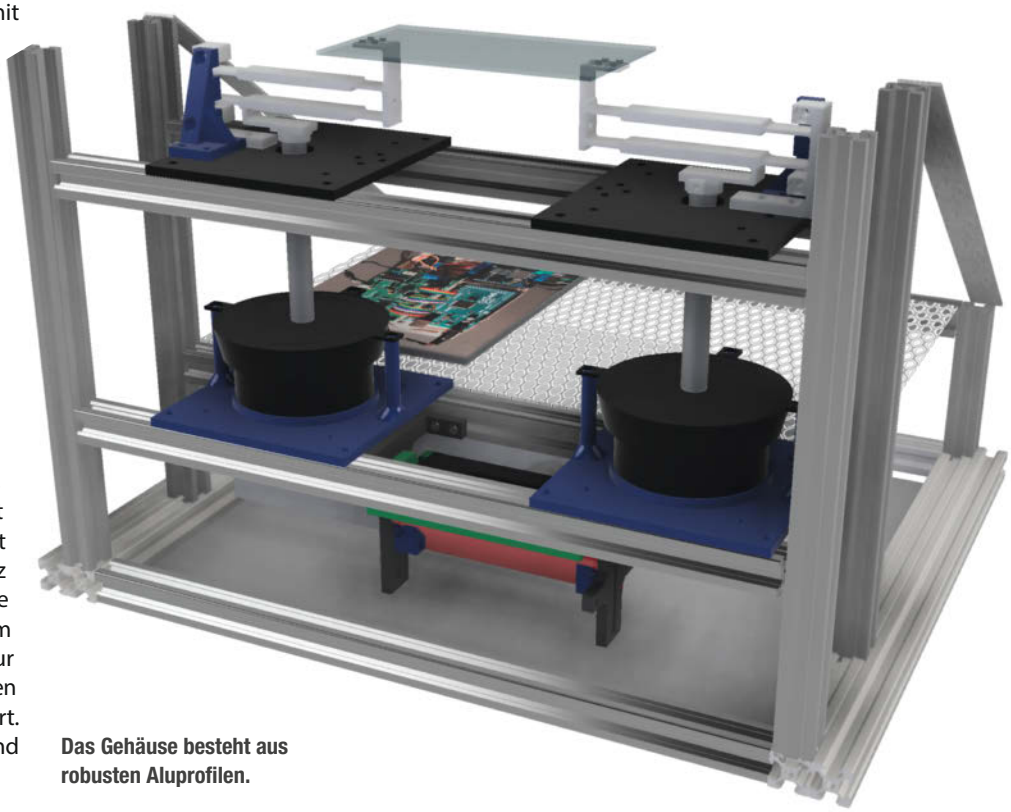
Damit das Volumendisplay auch etwas darstellen kann, habe ich drei Windows-Tools programmiert: Einen Kalibrierungshelfer, um die Einzelbilder der richtigen vertikalen Position zuzuordnen, einen Player und einen Slicer. Mit ihm können beliebige Objekte im STL- oder OBJ-Format eingelesen, ausgerichtet und in 12 Schichtbilder zerlegt werden. Diese werden dann in der richtigen Reihenfolge den ehemaligen RGB-Frames zugeordnet, damit der Player die Schichten auch korrekt auf der Projektionsfläche darstellt. Animationen kann man durch das Anzeigen mehrerer 3D-Ansichten erzeugen. Der Player läuft entweder auf einem angeschlossenen PC oder auf dem integrierten Lattepanda Delta SBC, der somit einen vollständigen Standalone-Betrieb des Volumendisplays ermöglicht.

Galaktisches Gehäuse

Nachdem ich die Technik ausgiebig getestet hatte, wollte ich für das Volumendisplay ein passendes Gehäuse bauen und als Star Wars-Fan musste ich ihm einfach das Aussehen und das Gefühl einer weit, weit entfernten Galaxie geben. Das Grundgerüst besteht aus verschraubten Aluprofilen. Ganz unten ist der Beamer an einer Bodenplatte befestigt, dessen Projektion über einen um 45 Grad geneigten Spiegel nach oben zur Projektionsfläche gelangt. Die Schwingspulen habe ich oberhalb des Projektors montiert. Daneben befindet sich die Elektronik und ganz oben sind die Halter angebracht.

Für die Front- und Seitenabdeckungen habe ich mich für geschichtete Panels mit typischen Star-Wars-Elementen entschieden, bestehend aus quadratisch gekachelten roten und weißen Knöpfen, vielen LEDs, weißen Stiftstreifen und weiteren Verzierungen. Die Paneele habe ich dafür 3D-gedruckt. Als besonderes Detail habe ich den USB-Port des MEGA zentral in einen Droid-Port integriert.

Wer das Volumendisplay nachbauen oder in Aktion sehen will, findet die vollständige Bauanleitung, Videos und weitere Details unter dem Link in der Kurzinfo. *—akf*



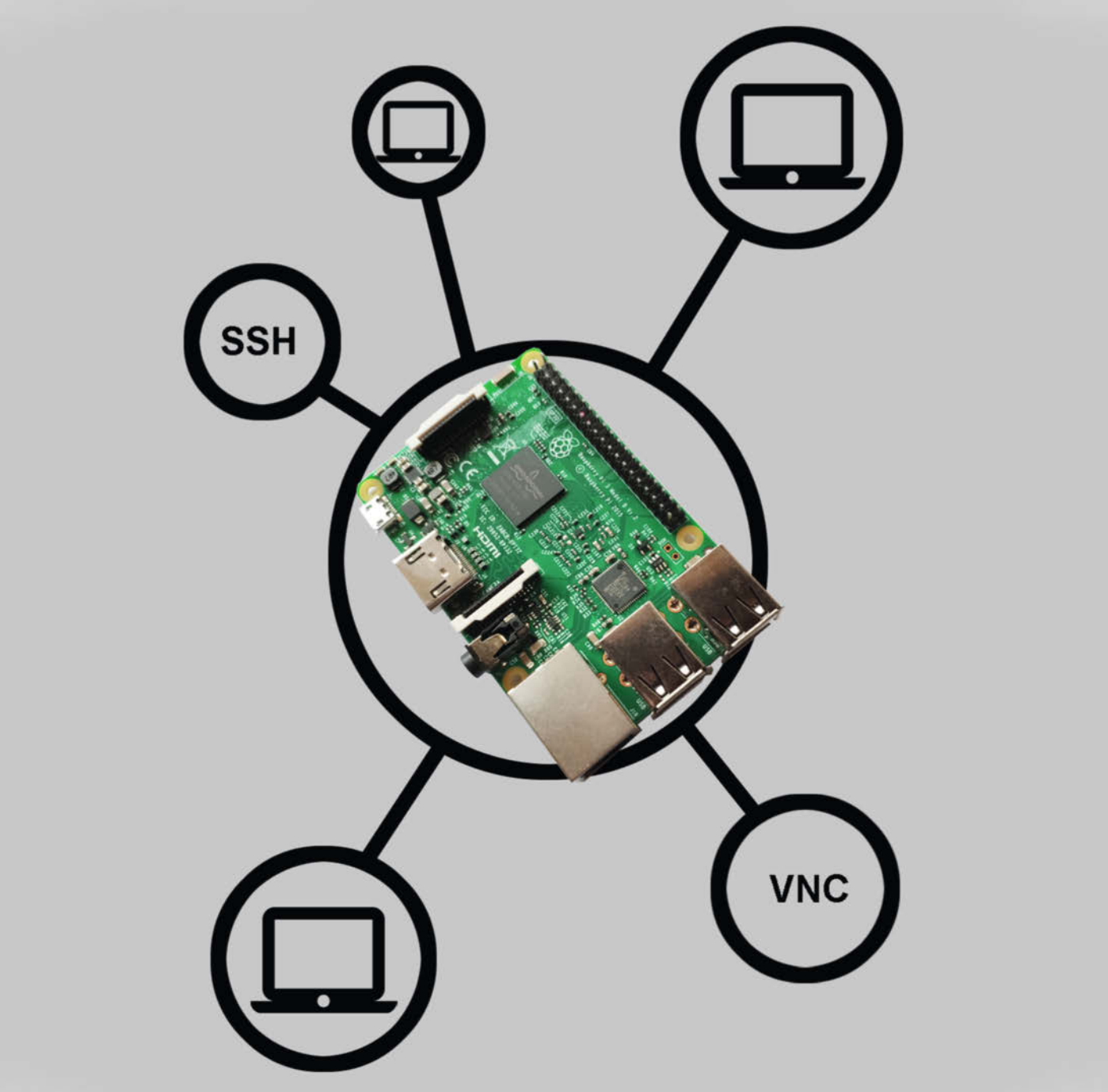
Das Gehäuse besteht aus robusten Aluprofilen.

© Copyright by Maker Media GmbH.

Verbindungen zum Raspberry Pi Server

Betreibt man einen Raspberry Pi daheim als sparsamen Server, fehlen ihm meist Monitor und Tastatur. Wie man trotzdem auf den Server zugreifen, Befehle aus der Ferne starten und Daten ausgeben kann, zeigt unser Artikel.

von Daniel Schwabe



© Copyright by Maker Media GmbH.

Als Steuerzentrale für das Smarthome oder als Netzwerkspeicher (NAS), für viele dieser Anwendungen kann man einen Raspberry Pi verwenden. Egal wofür man den Einplatinen-Computer benutzt, man benötigt ab und an Zugriff auf die Daten oder die Kommandozeile des Raspberry Pis – beispielsweise um Sicherheitsupdates zu installieren. Neben einem physischen Zugriff über einen angeschlossenen Monitor und eine Tastatur gibt es auch Zugriffsmöglichkeiten über das Netzwerk. Vier davon stellen wir hier vor.

Kurzinfo

» Fernzugriff über SSH
» Aufrufen von grafischen Programmen von einem anderen PC
» Fernsteuern des gesamten Desktops

Material

» Ein Raspberry Pi mit Netzteil
» Ein Windows PC

Mehr zum Thema

» Heinz Behling, Smart Home: Intelligentes Heim mit Home Assistant, Make 1/21, S. 100
» Ákos Fodor, Mediacenter mit dem Raspberry Pi, Make 1/23, S. 32

Alles zum Artikel im Web unter *make-magazin.de/x5nx*

Kommandozeile über SSH

SSH steht für „Secure Shell". Über das Netzwerk kann damit eine Verbindung zur Eingabezeile des Raspberry Pi aufgebaut werden. Diese Verbindungsmöglichkeit lässt sich bereits beim Kopieren des Betriebssystems auf die SD-Karte einrichten. Mit dem Tool „Raspberry Pi Imager" kann man während des Kopiervorgangs mit folgenden Einstellungen – nach einem Klick auf das Zahnrad unten rechts – direkt die SSH-Funktion aktivieren.

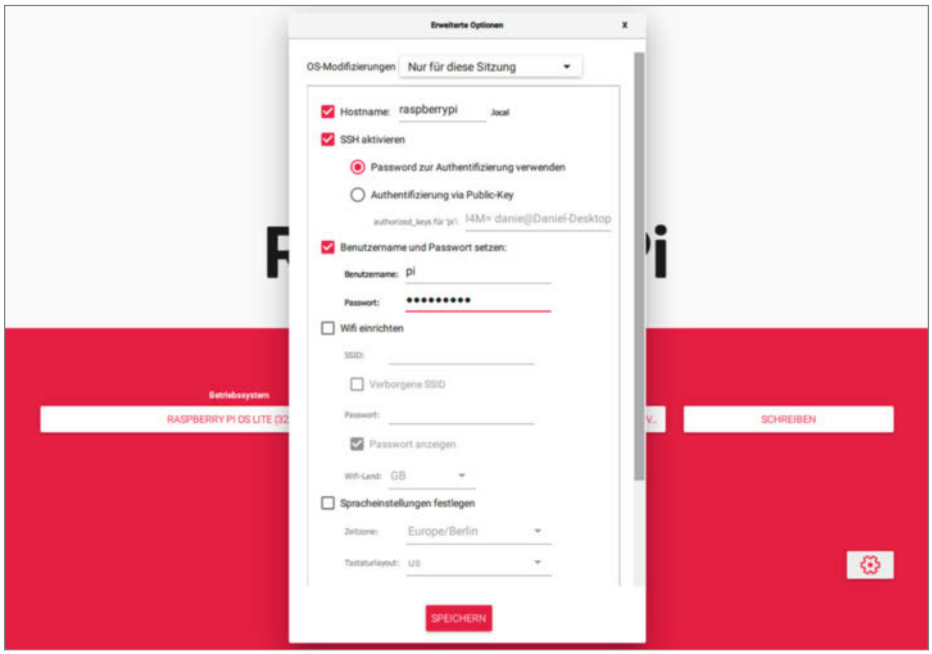
Mit dem Einstellen eines Hostnamens (im Beispiel raspberrypi.local) erübrigt sich später das Herausfinden der IP-Adresse und der Raspberry Pi ist direkt mit diesem Hostnamen ansprechbar. Der Haken bei „SSH aktivieren" macht genau das, was drauf steht: direkt nach dem ersten Hochfahren den SSH-Server aktivieren. Um später auf den Pi über diesen Weg zuzugreifen, richtet man die Authentifizierung über ein Passwort ein und trägt es in der Option „Benutzername und Passwort setzen" ein. Diese Daten werden immer für den Zugriff benötigt.

Nachdem das alles eingestellt ist, kann man auf „Speichern" klicken und das Image auf eine SD-Karte speichern. Soll auf einem bereits bespielten Raspberry Pi der SSH-Zugriff aktiviert werden, muss über einen anderen Weg Zugriff auf die Kommandozeile des Computers hergestellt und folgender Befehl eingegeben werden:

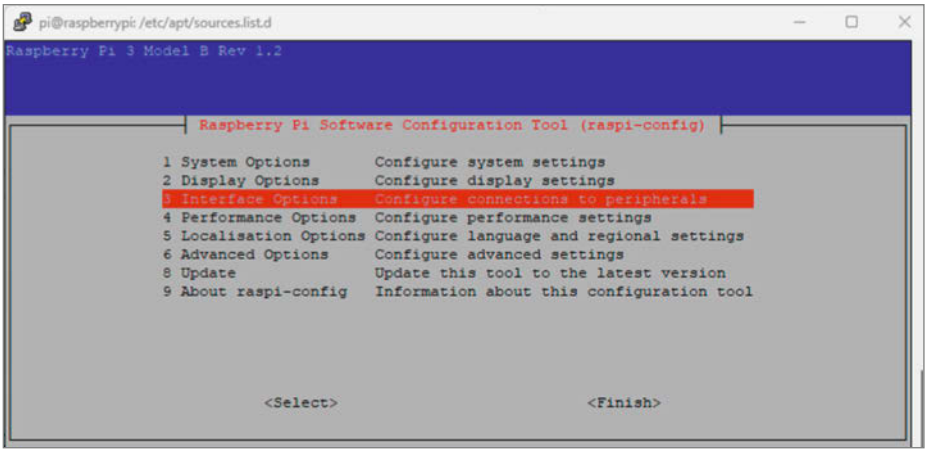
```
sudo raspi-config
```

Im Auswahlfenster navigiert man „Interface Options/SSH" und bestätigt mit Enter. Nach einem Neustart des Raspberry Pi ist der SSH-Zugriff über die IP-Adresse und die Zugangsdaten möglich.

Wenn für diesen Server kein Hostname eingestellt wurde, muss man die IP-Adresse über den Router herausfinden. Wenn ein Monitor und eine Tastatur an den Raspberry Pi angeschlossen sind, kann man die IP-Adresse auch über den Befehl `ifconfig` herausfinden. Je nachdem, wie der Raspberry Pi mit dem Internet verbunden ist, lässt sich die IP im Eintrag „eth0" bei einer LAN-Verbindung oder unter „wlan0" bei einer WLAN-Verbindung ablesen. Um jetzt von einem Windows-PC vor

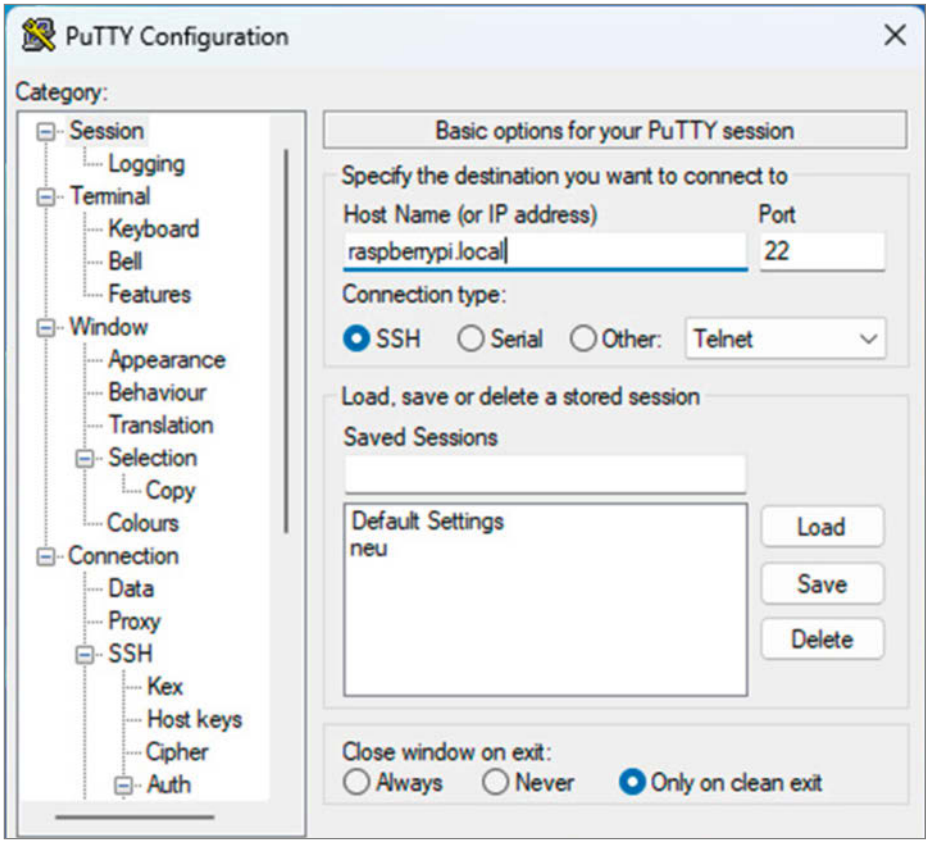


Der Raspberry Pi Imager erleichtert die Installation von Raspberry Pi OS.



Über `sudo raspi-config` lassen sich viele Parameter des Einplatinen-Computers einstellen.

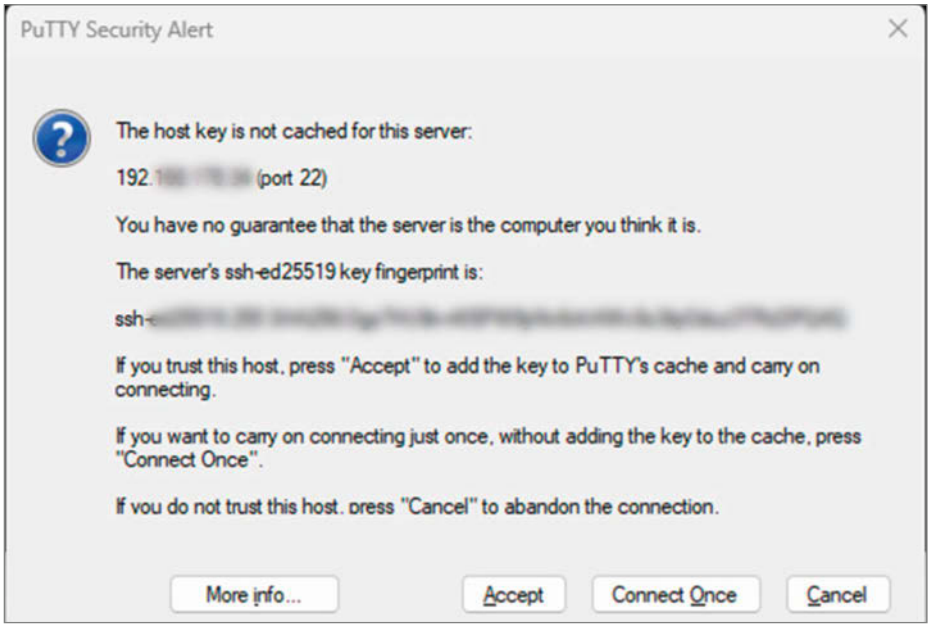
© Copyright by Maker Media GmbH.



Putty ist ein Tool mit großem Funktionsumfang, um Verbindungen zu anderen Computern herzustellen.

Windows 10 per SSH auf den Rapsberry Pi zuzugreifen, benötigt man das Programm Putty. Ab Windows 10 ist bereits ein SSH-Client in das Betriebssystem integriert. Die Anleitung dafür folgt weiter unten. Nach der Installation von Putty und dem Start des Programms kann man unter „Host Name (or IP address)" jetzt „raspberrypi.local" eintragen und mit einem Klick auf „Open" die Verbindung herstellen.

Bei der ersten Verbindung über diesen Weg öffnet sich ein Fenster, welches darüber informiert, dass ein Zugriff auf einen unbekannten Server erfolgt. Hier kann man diesen Server dann durch einen Klick auf „Accept" zu den bekannten Servern hinzufügen. Wenn man auf „Connect Once" klickt, wird diese Nachricht für diese Sitzung ignoriert und sie erscheint bei der nächsten Verbindung wieder.



Bei der ersten Verbindung mit dem Raspberry Pi erscheint so eine Fehlermeldung.

Es öffnet sich ein Eingabefenster, in dem man zuerst nach dem Benutzernamen gefragt wird. In diesem Beispiel ist das „pi". Nachdem man mit Enter bestätigt hat, wird man zur Eingabe des Passworts aufgefordert. In diesem Fall ist das „raspberry". Bei der Eingabe des Passworts werden keine Zeichen, weder die eingegebenen Buchstaben noch Sterne, im Eingabefenster angezeigt. Nachdem man wieder mit Enter bestätigt hat, ist man auf dem Rapsberry Pi eingeloggt und kann in dessen Eingabeaufforderung Befehle ausführen.

Wer einen Computer mit mindestens Windows 10 nutzt, kann auch mit Boardmitteln eine SSH-Verbindung aufbauen. Das kann man über die PowerShell oder cmd machen. Dort kann dann über den Befehl `ssh pi@raspberrypi.local` auf die Kommandozeile des Raspberry Pi Servers zugegriffen werden. Bei der ersten Verbindung wird man auch hier darauf hingewiesen, dass der lokale Computer den entfernten Server noch nicht kennt. Diese Meldung muss mit „yes" bestätigt werden.

Visuelle Übertragungen über SSH X

Über SSH können auch grafische Programme auf einem entfernten Raspberry Pi ausgeführt und auf dem lokalen Rechner angezeigt und genutzt werden. Dafür benötigt man auf dem lokalen Windows-PC das Programm Xming. Nach der Installation von Xming muss das Programm gestartet werden. Es erzeugt allerdings kein eigenes Fenster, sondern das Xming-Symbol wird nur in der Statusleiste angezeigt. Jetzt ist noch eine Einstellung in Putty notwendig. Unter „Connection/SSH/X11" klickt man die Option „Enable X11 forwarding" an und fügt im Feld „X Display location" `localhost:0.0` ein. Man kann Xming auch mit dem von Windows integrierten SSH-Client nutzen. Die Anleitung dafür steht weiter unten.

Jetzt wird eine Verbindung mit dem Raspberry Pi wie bei einer normalen SSH-Verbindung hergestellt. Um zu testen, ob alles funktioniert, ist mit dem Kommando `sudo apt install x11-apps` eine Sammlung kleiner Programme installierbar. Wenn die Installation erfolgt ist, startet der Befehl `xcalc` einen Taschenrechner.

Wer auch hier Boardmittel verwenden will, muss ebenfalls zuerst das Programm Xming installieren und starten. Danach in der PowerShell oder cmd den Befehl `setx DISPLAY` „`127.0.0.1:0.0`" ausführen. Nach einem Neustart der Eingabeaufforderung kann mit `ssh -Y pi@raspberrypi.local` auch so eine grafische Verbindung aufgebaut werden.

Wichtig hierbei: Programme, die über SSH -X bzw. SSH -Y übertragen werden, laufen weiter-

© Copyright by Maker Media GmbH.

X Window System

Mit X Window wird ein System bezeichnet, das die Eingabe an Programme und die grafische Ausgabe von Programmen verwaltet.

Auf dem Computer, an dem gearbeitet wird, läuft der sogenannte X-Server. Er verwaltet die Maus- und Tastatureingaben und schickt sie an die mit ihm verbundenen X-Clients. Die Clients sind in diesem Fall Programme. Ein Client sendet seine Informationen zurück an den X-Server, der diese dann in Form einer grafischen Darstellung ausgibt.

Das kann sowohl lokal (X-Server und X-Client laufen auf dem selben Computer) als auch über das Netzwerk (X-Server und X-Client laufen auf unterschiedlichen Geräten) genutzt werden.

Im Beispiel in diesem Artikel läuft auf dem lokalen Rechner mit Xming ein X-Server, der sich mit dem X-Client xcalc auf dem Raspberry Pi über das Netzwerk verbindet.

hin auf dem entfernten Raspberry Pi. Dokumente, die man in einer Textverarbeitung bearbeitet, können beispielsweise nicht auf dem lokalen Rechner gespeichert werden.

Zugriff per Browser mit Cockpit

Die Software Cockpit erlaubt den Zugriff auf einen Server per Browser. Dabei kann man nicht nur Befehle in der Kommandozeile ausführen (wie bei einer SSH-Verbindung), sondern erhält einen Überblick über die Hardware, laufende Prozesse und weitere Funktionen. Die Installation von Cockpit auf dem Raspberry Pi erfolgt mit dem Befehl `sudo apt install cockpit`. Nach der Installation hört Cockpit auf den Port 9090. Die Internetadresse https://raspberrypi.local:9090 im Browser erlaubt das Einloggen.

Für Cockpit gibt es verschiedene Add-Ons. Beispielsweise einen Dateiexplorer, den man einfach mit folgenden Befehlen installiert:

```
wget -qO - https://repo.45drives.com/key/gpg.asc | sudo gpg --dearmor -o /usr/share/keyrings/45drives-archive-keyring.gpg
cd /etc/apt/sources.list.d
sudo curl -sSL https://repo.45drives.com/lists/45drives.sources -o /etc/apt/sources.list.d/45drives.sources
sudo apt update

sudo apt install cockpit-navigator
```

Weitere Add Ons kann man unter https://cockpit-project.org/applications finden.

VNC – Desktop fernsteuern

Mittels VNC (Virtual Network Computing) kann die grafische Oberfläche eines Computers

Hier kommt Verstärkung

Inklusive Experimentierset

Das **Make-Sonderheft** bietet einen praxisorientierten Einstieg in Schaltungen mit Operationsverstärkern inkl. Experimentierset. Will man Sensorsignale verarbeiten oder verstärken, Spannungen überwachen oder Audiosignale filtern: Mit geringem Aufwand und ohne komplizierte Berechnungen setzt man Operationsverstärker ein. Das Heft erklärt, wie alle Schaltungen funktionieren.

- Operationsverstärker verstehen
- Komparatoren und Schmitt-Trigger erklärt
- Spannungsversorgungen und virtuelle Masse
- Schaltungen selbst entwerfen und berechnen
- Viele praktische Anwendungen
- **Inkl. Experimentierset Operationsverstärker**

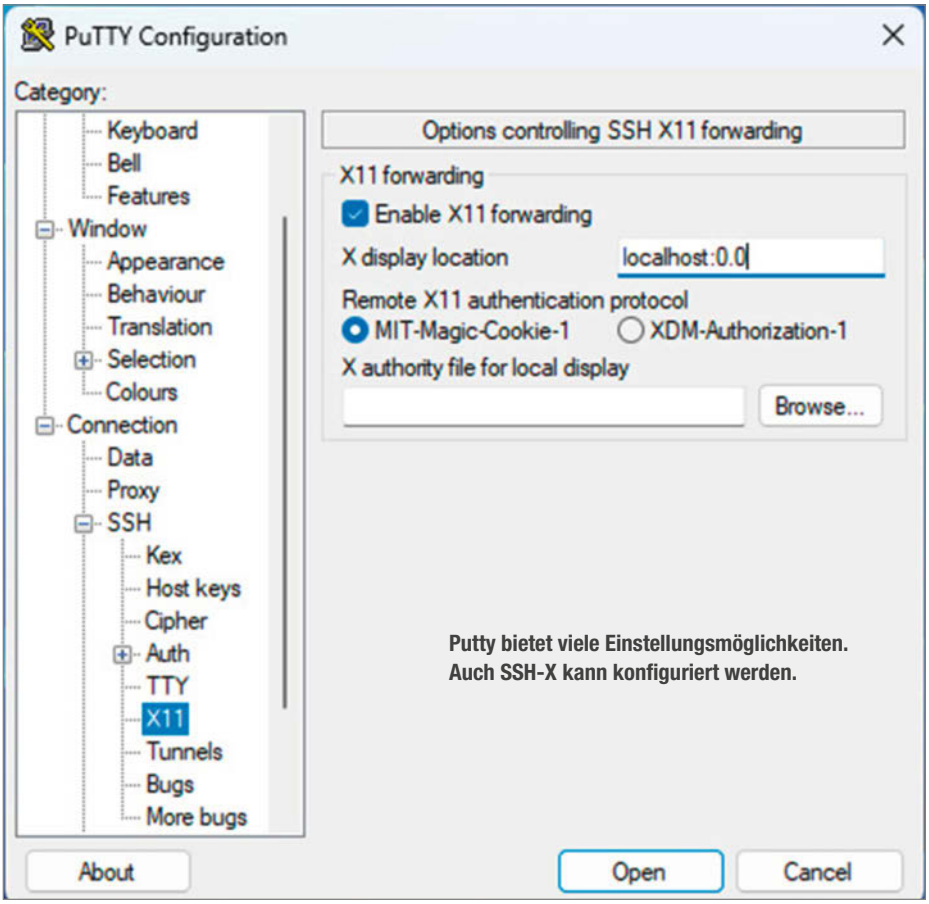
Heft + Experimentierset für nur 49,95 €

shop.heise.de/make-opv

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

© Copyright by Maker Media GmbH.

heise Shop



Putty bietet viele Einstellungsmöglichkeiten. Auch SSH-X kann konfiguriert werden.

Mit dem Navigator Add On hat man Zugriff auf alle Dateien und Ordner auf dem Raspberry Pi.

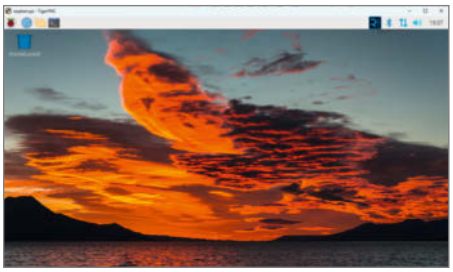
Für eine VNC-Verbindung ist eine grafische Umgebung notwendig.

ferngesteuert werden. Um mit einem Raspberry Pi eine VNC-Verbindung herzustellen, muss beim Flashen des Betriebssystems eine Version MIT grafischer Oberfläche ausgewählt werden. Betriebssysteme, die mit „Server" oder „Lite" bezeichnet werden haben in der Regel KEINE Desktopumgebung.

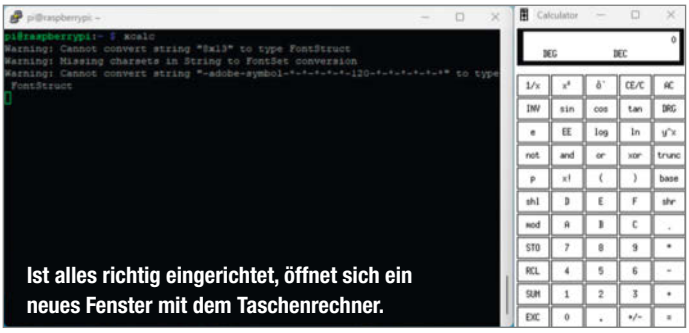
Um den VNC-Dienst einzurichten, braucht man anfangs auch eine SSH-Verbindung. Dafür die Einstellungen aus dem Anfang dieses Artikels übernehmen. Nach der Anmeldung per SSH mit den Befehlen `sudo apt update` und `sudo apt upgrade` die Software auf den neusten Stand bringen. Danach mit `sudo raspi-config` das Konfigurationspanel öffnen. Dort die Option „Interface Options/VNC" aktivieren und bestätigen. Nachdem alles installiert ist, nochmals mit `sudo raspi-config` die Konfiguration öffnen. Jetzt unter „Display Options/VNC Resolution" eine Auflösung wählen. Danach den Raspberry Pi neu starten. Jetzt braucht man auf dem PC, der sich verbinden soll, das Programm TigerVNC. Es funktionieren auch andere Programme wie RealVNC. Bei TigerVNC muss unter Options/Security die Option RSA-AES aktiviert und mit OK bestätigt werden. Jetzt im Verbindungsfeld raspberrypi.local eintragen und verbinden. Bestätigen, dass eine Verbindung hergestellt werden soll und mit den Zugangsdaten einloggen. Jetzt erscheint ein Fenster mit dem Desktop des Raspberry Pi.

Connection Closed

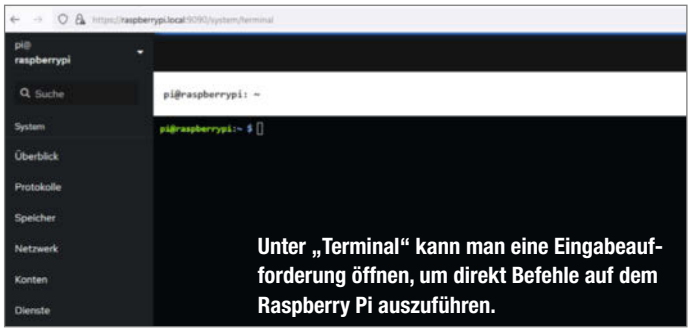
Dies waren einige Möglichkeiten, um auf einen Raspberry Pi im Netzwerk zuzugreifen. Dadurch wird das Verwalten von Servern sehr viel einfacher. *—das*



Über VNC kann auf dem Raspberry Pi Desktop von einem anderen Rechner aus gearbeitet werden.



Ist alles richtig eingerichtet, öffnet sich ein neues Fenster mit dem Taschenrechner.



Unter „Terminal" kann man eine Eingabeaufforderung öffnen, um direkt Befehle auf dem Raspberry Pi auszuführen.

© Copyright by Maker Media GmbH.

betterCode() – Konferenzen

MAGAZIN FÜR PROFESSIONELLE IT

dpunkt.verlag

JAVA 21

Die Heise-Konferenz zur neuen LTS-Version

4. Oktober 2023 – Online

Jetzt Tickets sichern!

- Bessere Nebenläufigkeit mit Virtual Threads, Structured Concurrency und Scoped Values
- Pattern Matching for Switch ist finalisiert
- Foreign Functions & Memory API
- Wie finden neue Features ihren Weg ins OpenJDK?
- Was bedeutet das LTS-Release für Projekte?

Workshops zu den neuen Features in Java 21
und zu sicheren Java-Anwendungen mit Spring Security

java.bettercode.eu

.NET 8.0

Das Online-Event von Heise und www.IT-Visions.de zum neuen .NET-LTS-Release

21. November 2023 – Online

Jetzt Tickets mit Frühbucher-rabatt sichern!

- Die Neuerungen von .NET 8.0: SDK, Runtime und Basisklassen
- Einfacher lesbarer, stabilerer Code mit C# 12.0
- Alle Neuerungen von ASP.NET Core 8.0 und Blazor 8.0
- Neues beim OR-Mapping mit Entity Framework Core 8.0
- Das hat sich mit Windows Forms 8.0, WPF 8.0 und WinUI 3 verändert
- Cross-Plattform-Entwicklung mit .NET MAUI
- Ausblick auf .NET 9.0

Kooperationspartner
www.IT-Visions.de
Dr. Holger Schwichtenberg

Workshops zu C# 12.0, Entity Framework Core 8.0, Blazor 8.0 und .NET MAUI 8.0

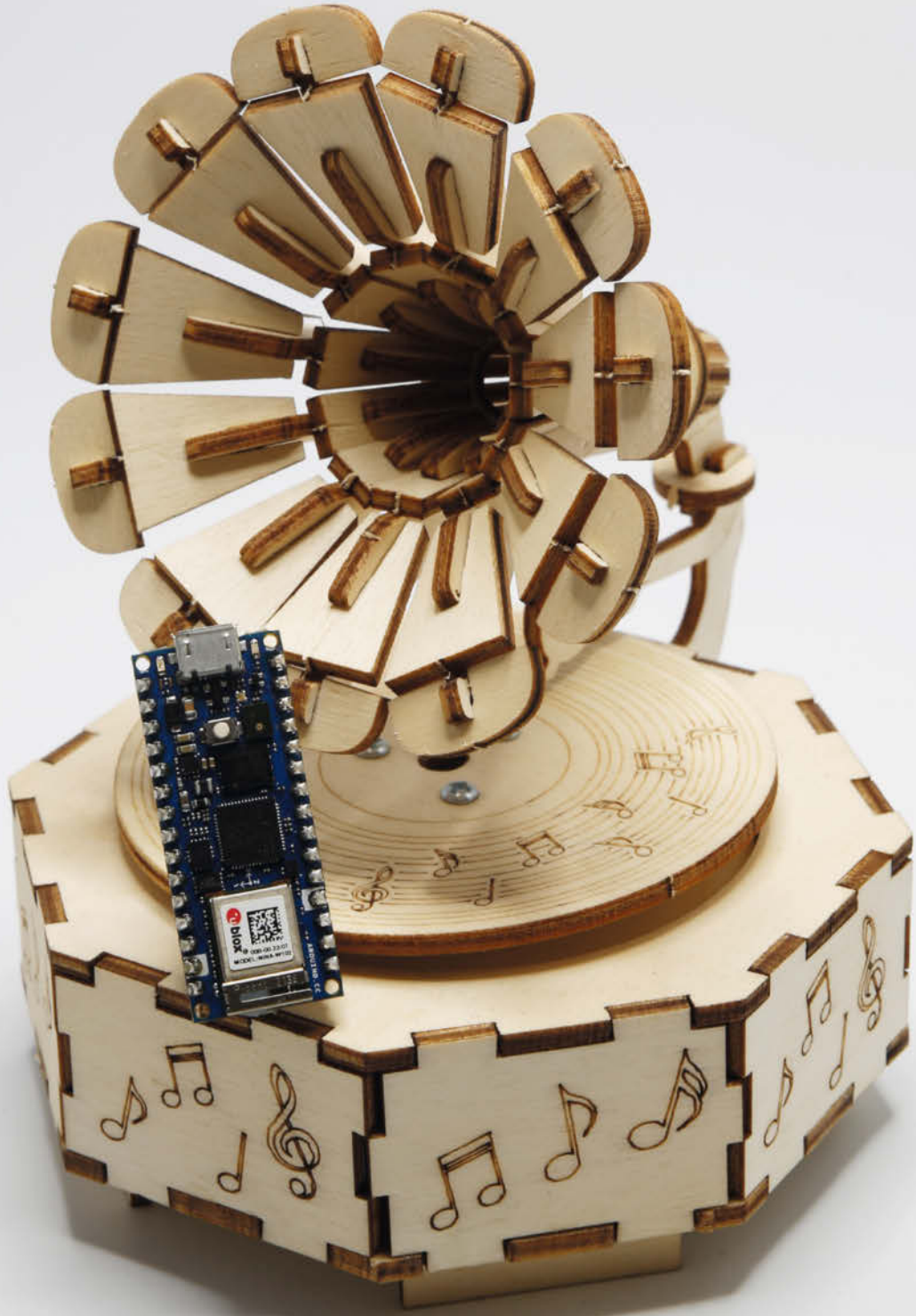
net.bettercode.eu

© Copyright by Maker Media GmbH.

His Masters Voice:
Arduino hört zu!

Spracherkennung mit einem Arduino, das war bislang mangels Rechenpower unvereinbar. Der Arduino Nano RP2040 connect hat aber einen leistungsfähigen Prozessor aus der Raspberry-Familie an Bord und soll zusammen mit der richtigen Software diese Aufgabe bewältigen können. Wir haben das zum einen getestet, und liefern Ihnen zum anderen anhand des Testprojekts gleich eine Anleitung.

von Heinz Behling



© Copyright by Maker Media GmbH.

Eines der neuesten Controllerboards aus dem Hause Arduino ist der Arduino Nano RP2040 connect. Wie der Name schon vermuten lässt, hat das Board das Format eines herkömmlichen Nanos, allerdings werkelt darauf ein von Raspberry stammender 32-Bit-Prozessor, der RP2040 mit einem zweikernigen ARM Cortex M0+ (Bild 1). Der mit 133 MHz laufende Chip hat genügend Dampf unter der Haube, um auch rechenintensive Aufgaben zu bewältigen, beispielsweise KI-Engines zur Spracherkennung. Das muss doch mal getestet werden. Dabei interessierte insbesondere auch, ob die Spracherkennung lokal oder online erfolgt.

Passend dazu gibt es eine Software-Bibliothek von der taiwanesischen Firma Cyberon, einem Spezialisten für Spracherkennung. Die kostenlose Version dieser Bibliothek reicht für einfache Aufgaben und erste Tests. Für 9 Euro kann man im Arduino-Shop aber auch einen Voucher für die Vollversion der Bibliothek erwerben, die dann mehr Funktionen bietet. Entsprechende Bibliotheken gibt es auch für Arduino Portenta H7 und den Arduino Nano 33 BLE Sense Rev1 und 2. Das war schon mal ein erster Hinweis darauf, dass die Spracherkennung ohne Netzwerkverbindung auskommen soll, denn der Nano 33 BLE hat nur Bluetooth, aber kein WLAN.

Am Ende dieses Tests sollte der Arduino seine Onboard-RGB-LED (Bild2) mit verschiedenen Lichteffekten sprachgesteuert erstrahlen lassen. Wir wollen schließlich auch wissen, wie einfach (oder auch nicht) die Programmierung ist. Da im Folgenden die Programmierung des kleinen Testprojekts beschrieben wird, können Sie diesen Test auch gleichzeitig als Projekt-Anleitung benutzen.

Kurzinfo

» Installation der Spracherkennungsbibliothek
» Anlernen neuer Sprachbefehle in Deutsch
» Sprachsteuerung der Onboard-RGB-LED des Arduino Nano RP2040 connect

Checkliste

Zeitaufwand:
2 Stunden

Kosten:
ab 35 Euro

Material

» Arduino Nano RP2040 connect

Mehr zum Thema

» Heinz Behling, Voice Recognition, Make 1/20, S. 38
» Heinz Behling, Sprecherunabhängige Spracherkennung, Make 6/20, S. 120

Alles zum Artikel im Web unter *make-magazin.de/x78x*

Arduino IDE nachrüsten

Um den Nano RP2040 connect dazu zu bringen, aufs Wort zu gehorchen und entsprechende Aktionen auszulösen, muss er mithilfe der Arduino IDE programmiert werden. Im Folgenden wurde das mit der aktuellen Version 2.2.1 erledigt. Die musste zunächst mit der Bibliothek „Cyberon_DSpotterSDK_Maker_RP2040" aufgerüstet werden, die nicht nur zur Spracherkennung, sondern auch zum Auslesen der Seriennummer des Arduinos benötigt wird. Ohne diese Seriennummer erhält man nämlich keine Lizenz für die Bibliothek und die Spracherkennung funktioniert dann nicht. Die IDE macht die Installation aber einfach: Unter „Sketch/Bibliothek einbinden/Bibliotheken verwalten" war ins Suchfeld nur der Name der Bibliothek einzugeben und dann auf „installieren" zu klicken (Bild 3). Wunderschön einfach, ohne sich mit Github oder sonstigen Softwarequellen zu befassen.

Die Bibliothek hat einige Abhängigkeiten, ohne die sie nicht funktioniert. also wurden die auch alle installiert (Bild 4).

Nach Verbinden des Arduinos per USB-Kabel mit dem Computer und einem Klick auf den Pfeil im Feld „Board wählen" erschien darunter bereits der Eintrag für den Nano. Ein Klick darauf übernahm die Board- und Portauswahl. Die Arduino IDE forderte nun, die für das Board notwendige Core-Datei zu installieren (Bild 5). Erlaubnis wurde mit „Ja" erteilt. Die Installation dauerte knapp zwei Minuten, unterbrochen von Sicherheitsfragen, die mit „Ja" beziehungsweise „Installieren" beantwortet wurden.

Lizenz-Seriennummer

Damit war alles für die Ermittlung der Seriennummer erforderliche an Board. Unter „Datei/Beispiele/DSpotter_SDK_Maker_2040" fand sich der Sketch „GetSerialNumber". Den muss man laden, kompilieren und auf den Arduino hochladen. Anschließend zeigt der serielle Monitor unterhalb des Programmcodes die Seriennummer an (Bild 6).

Damit ließ sich nun die Lizenz der Spracherkennungs-Bibliothek beschaffen. Auf der Internet-Seite https://tool.cyberon.com.tw/ArduinoDSpotterAuth/FDMain.php war dazu der Boardtyp „Arduino Nano RP2040 connect" auszuwählen und die gerade ermittelte Seriennummer einzugeben. Sicherheitshalber wurde dieses 30-stellige Monster per Kopieren und Einfügen aus dem seriellen Monitor übertragen. Anschließend ging es mit „Submit" weiter. Als Resultat erschien ein Zahlenfeld, der Lizenzschlüssel (Bild 7).

Ausprobieren

Jetzt konnte die Spracherkennung zum ersten Mal zeigen, ob sie funktioniert: Unter „Datei/Beispiele/DSpotter_SDK_Maker_2040" gibt es dazu den Sketch „VoiceRecognition". Der Sketch besteht aus vier Dateien:

- VoiceRecognition.ino
- CybLicense.h
- Model_L0.h
- Model_L1.h

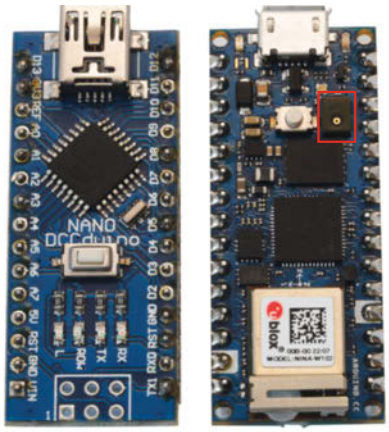


Bild 1: Links der herkömmliche Arduino Nano, rechts der Arduino Nano RP2040 connect: Rechts neben seinem Reset-Taster befindet sich das Mikrofon.

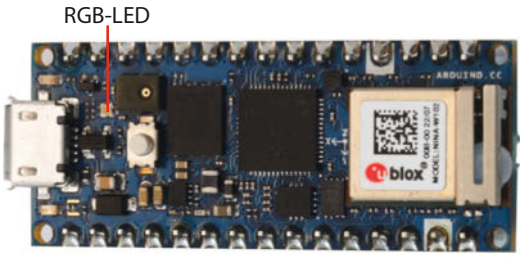


Bild 2: Die kleine RGB-LED auf dem Board fällt kaum auf, wenn sie nicht leuchtet. Das soll per Sprache geändert werden.

© Copyright by Maker Media GmbH.

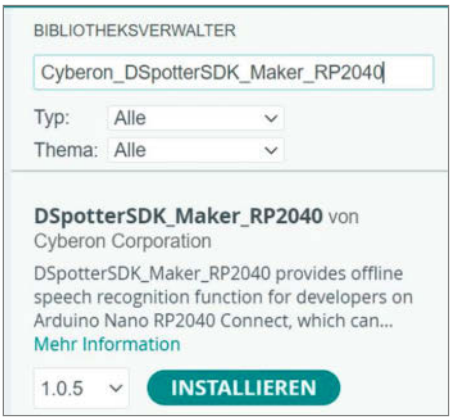


Bild 3: Die Spracherkennungs-Bibliothek brauchen Sie zunächst, um die Seriennummer des Arduinos auslesen zu können.

Als Erstes mussten die Lizenzdaten (alles, was in Bild 7 hinter „Result" steht) in die Datei „CybLicense.h" eingegeben werden, und zwar in Zeile 4 zwischen die geschweiften Klammern (Bild 8).

Zurück ging es zu VoiceRecognition.ino. Ein Durchsuchen des Sketches ergab: Nirgendwo wurde eine WLAN-Verbindung aufgebaut. Also sollte die Spracherkennung tatsächlich völlig autonom auf dem Arduino arbeiten. Das sollte nun ausprobiert werden: Nach dem Kompilieren des Sketches und Überspielen auf den Arduino zeigte er die Liste der momentan möglichen Sprachkommandos (Bild 9).

Der Arduino kannte nun zwei sogenannte Trigger-Befehle: „Hey Arduino" und „Hey Cyberon". Das sind die Befehle, auf die er im Trigger-Modus ständig hört (ähnlich wie bei Amazons Echo-Lautsprecher das Wort „Alexa"). Nur wenn er einen davon hört, schaltet er in den Kommando-Modus und wartet darauf, dass Sie einen der Befehle aus der command list sprechen. Die jeweils erkannten Befehle zeigte der serielle Monitor ebenfalls an (Bild 10). Dabei nahm der Arduino übrigens nur 40mA an Strom auf.

Bislang funktionierte ja alles sehr gut. Allerdings gab es in der Testversion die Einschränkung, dass pro Boardstart maximal 50 Befehlserkennungen erfolgten.

Eigene Befehle

Aber der Arduino sollte natürlich auf unsere eigenen, in Deutsch gesprochenen Befehle hören. Dazu musste ein Training erfolgen, und dieses musste tatsächlich online über die Seiten von Cyberon im Browser des Computers erledigt werden. Für das Training eigener Befehle war eine Registrierung auf der Seite https://tool.cyberon.com.tw/ArduinoDSpotterAuth/CTMain.php notwendig, für die eine E-Mail-Adresse, die Seriennummer des Arduinos und das Lesen der Lizenzbedingungen erforderlich ist. Anschließend wählte man die Sprache des neuen Projekts, das übrigens nach Fertigstellung an das Board mit der zuvor eingegebenen Seriennummer gebunden ist und daher auf keinem anderen Arduino Nano RP2040 connect läuft.

Mit „Create" ging es weiter. Jetzt folgte die Festlegung des Trigger-Befehls. Er soll vier bis sechs Silben haben, auf einen Vokal enden und ein möglichst selten benutzter Begriff sein, um Fehlaktivierungen des Kommando-Modus zu vermeiden. „Hallo Nano" schien geeignet zu sein. Mit „Add" wurde der Befehl der Triggerwort-Liste hinzugefügt (Bild 11) und mit „Next" ging es weiter.

Dann gelangte man zur Eingabe der Kommandobefehle. Hier waren bis zu 20 möglich. Für dieses Test-Projekt sollte die LED mit fünf Effekten (rot blinken, grün blinken, blau blinken, Regenbogen und weißes Licht) angesteuert werden. Entsprechend wurden die Kommandos eingegeben (Bild 12).

Ein „Confirm" weiter erschien eine Zusammenfassung der Eingaben. Hier konnte man, falls etwas nicht stimmt, mit „Back" jeweils in die entsprechende Eingabemaske zurückgelangen und Fehler korrigieren. Gute Idee, denn ein einmal fertiggestelltes Projekt lässt sich nicht mehr ändern, so die Angaben auf der Herstellerseite. Da alles in Ordnung war, ging es weiter. Daraufhin wurden die Dateien

- Model_1694078235_Arduino_rp2040.dsproj
- CybLicense_1694078235.h
- Model_1694078235.h
- Info_1694078235.txt

erzeugt und an die zuvor angegebene E-Mail-Adresse gesendet. Das erfolgte im Test auch postwendend. Und dies beendete auch den online-Teil der Spracherkennung. Die praktische Anwendung der neu trainierten Befehle fand wieder ausschließlich offline auf dem Arduino statt.

Die erste der gemailten Dateien ist die Projektdatei, die man später beispielsweise nutzen kann, um dieses Projekt über die Cyberon-Seite auf andere Boards zu übertragen. Die zweite ist die Lizenzdatei speziell für dieses Projekt. In der dritten schließlich ist der Code für die Spracherkennung der eingegebenen Trigger- und Kommandobefehle, allerdings nicht im Klartext, sondern bereits vorkompiliert. Die Textdatei schließlich entsprach der Zusammenfassung aus dem vorangegangenen Training. Insbesondere konnte man hier die Befehlsworte nochmal nachschlagen, was sich später bei der Programmierung als nützlich erwies.

Sketch mit neuen Befehlen

Der Sketch für die LED-Steuerung sollte auf dem zuvor verwendeten „VoiceRecognition.

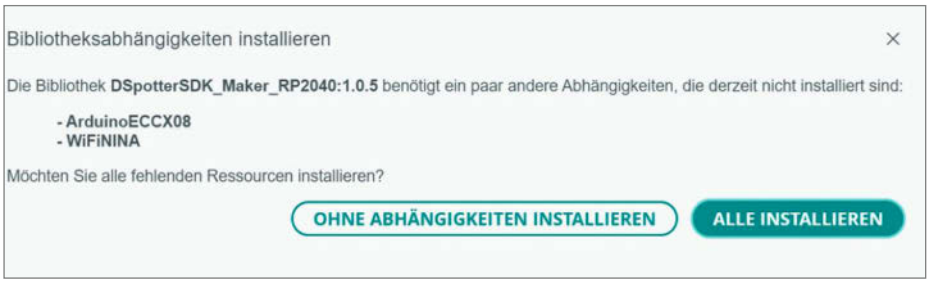


Bild 4: Ohne diese zusätzlichen Bibliotheken läuft nichts.

Bild 5: Die Core-Datei für den Nano connect ist ebenfalls notwendig.

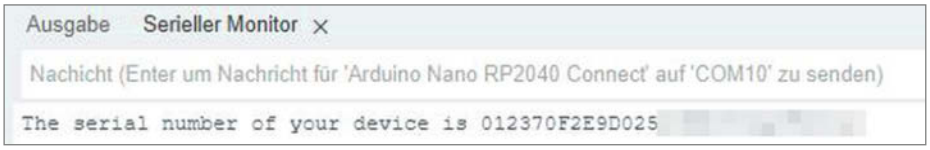


Bild 6: Die 30-stellige Nummer ist für den Download der Spracherkennungs-Bibliothek erforderlich.

© Copyright by Maker Media GmbH.

ino“ basieren. Daher speicherten wir den erstmal mit „Datei/Speichern unter“ unter dem neuen Namen „LEDSteuerung“. Dann mussten zwei der zuvor per EMail erhaltenen Dateien im Ordner „LEDSteuerung“ gespeichert werden, nämlich „CybLicense_1694078235.h“ und „Model_1694078235.h“.

Danach ging es in der Arduino IDE weiter: In LEDSteuerung.ino wurden drei Zeilen angepasst. Aus `#include „CybLicense.h“` wurde `#include „CybLicense_1694078235.h“`. Die Zeilen `#include „Model_L1.h“` und `#include „Model_L0.h“` wurden jeweils zu `#include „Model_1694078235.h“`.

Jetzt folgte der Test der neuen Befehle: Sketch kompilieren und hochladen, den seriellen Monitor starten und schauen, was passiert. Als erstes erstaunte die Meldung des Compilers, dass der Sketch nur 254514 Byte in Anspruch nahm. Das waren gerade mal 1 Prozent des 16 MB großen Speichers. Da bleibt also noch viel Platz für größere Anwendungen. Der Monitor gab auch hier zunächst die Liste mit den Befehlen aus.

Dann folgte der im Laufe des Tests doch etwas nervige Countdown von 20 auf 0, bis der Arduino bereit ist, das Trigger-Wort entgegenzunehmen (mit der kostenpflichtigen Bibliothek entfällt das). Das macht er auch problemlos. Selbst ein unmittelbar neben dem Arbeitsplatz spielendes Radio brachte ihn nicht aus dem Konzept, ob es nun Musik spielte oder die Nachrichten ertönten. Auch alle Kommandos (selbst das „grün blinken“ mit dem Umlaut) verstand er sofort. Wenn eine Zeitlang nicht gesprochen wird, schaltet die Software vom Kommando- zurück in den Trigger-Modus (und der lästige Countdown bei der Testversion beginnt erneut).

Um jetzt der Software beizubringen, was sie bei Erkennen eines Befehls zu tun hatte, brauchte der Sketch zunächst die Liste der Kommandos und Ihrer jeweiligen ID (stehen unter anderem in der Info-Textdatei, die man per E-Mail erhalten hat). Diese IDs wurden zunächst mittels `define`-Anweisungen Konstanten zugeordnet, die dann später im Programm verwendet wurden. Im Prinzip könnte man auch die IDs selbst benutzen, aber dann wäre so ein Programmtext kaum noch lesbar. Die Konstantennamen müssen übrigens nicht mit den Sprachkommandos übereinstimmen. Für die fünf Befehle dieses Tests sah das dann so aus:

```
#define COMMAND_ROT_BLINKEN      10000
#define COMMAND_GRUEN_BLINKEN    10001
#define COMMAND_BLAU_BLINKEN     10002
#define COMMAND_REGENBOGEN       10003
#define COMMAND_WEISSES_LICHT    10003
```

Spätestens jetzt sollte man sich kurz die Arbeitsweise der Spracherkennung zu Gemüte führen. Kernstück ist die Funktion `VRCallback` (siehe Kasten „Die Auswerte-Funktion“ auf

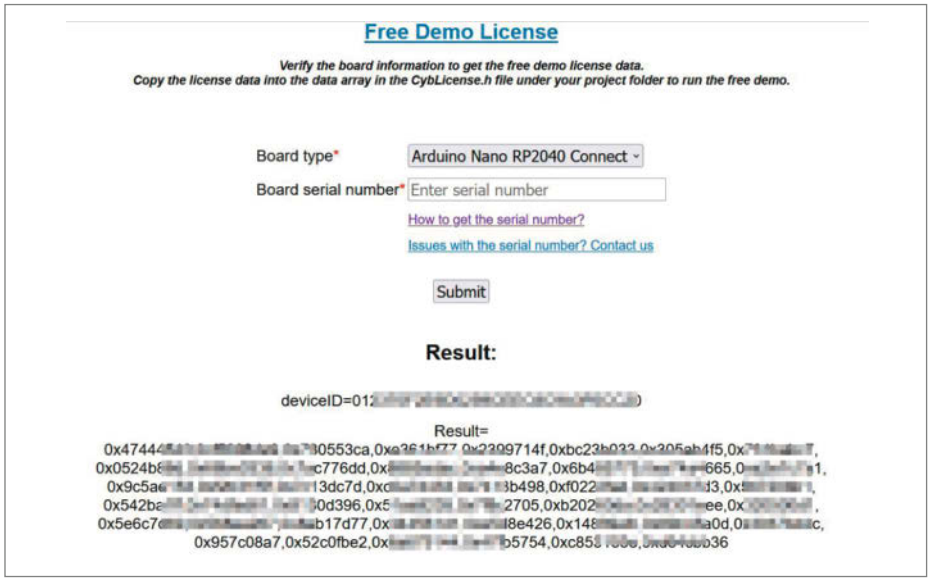


Bild 7: Den Lizenzschlüssel sollte man in eine Textdatei kopieren. Man braucht ihn bei jedem Spracherkennungs-Sketch.

Bild 8: So sieht der korrekt eingefügte Lizenzschlüssel aus.

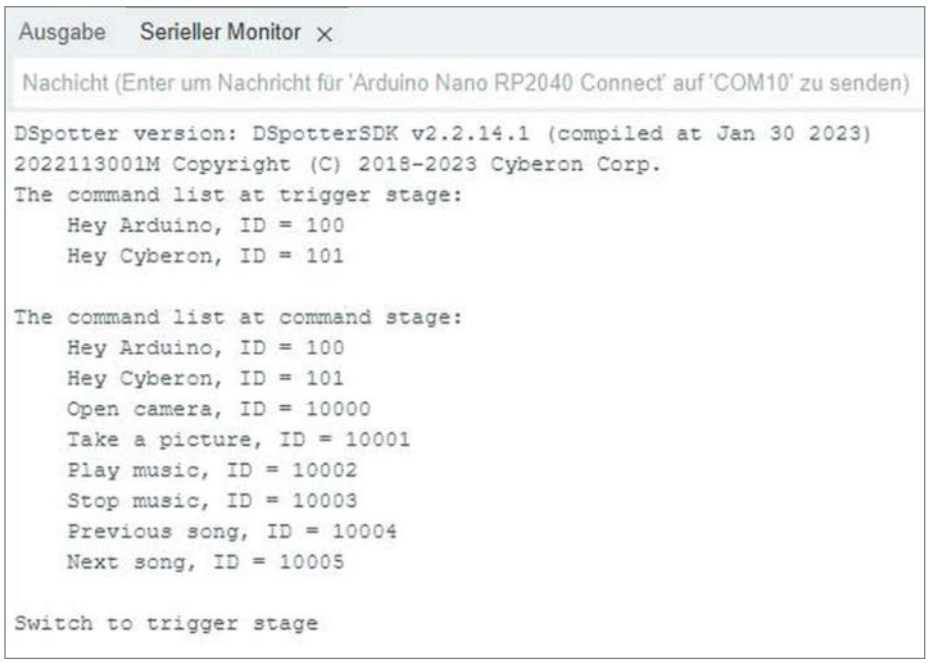


Bild 9: Die Kommandos, die der Arduino mit der Spracherkennungs-Bibliothek von Haus aus versteht, sind allesamt in Englisch.

© Copyright by Maker Media GmbH.

```
Switch to trigger stage
Get command: Hey Arduino, Score=9, SG_Diff=88, Energy=2064, ID=100
Switch to command stage
Get command: Open camera, Score=17, SG_Diff=103, Energy=723, ID=10000
Get command: Take a picture, Score=52, SG_Diff=147, Energy=1083, ID=10001
Get command: Play music, Score=40, SG_Diff=105, Energy=1513, ID=10002
Get command: Next song, Score=37, SG_Diff=103, Energy=1294, ID=10005
Get command: Stop music, Score=18, SG_Diff=78, Energy=1947, ID=10003
Switch to trigger stage
```

Bild 10: Das klappt ja prima. Ganz Sprecher-unabhängig erkennt der Arduino die ihm bekannten Befehle.

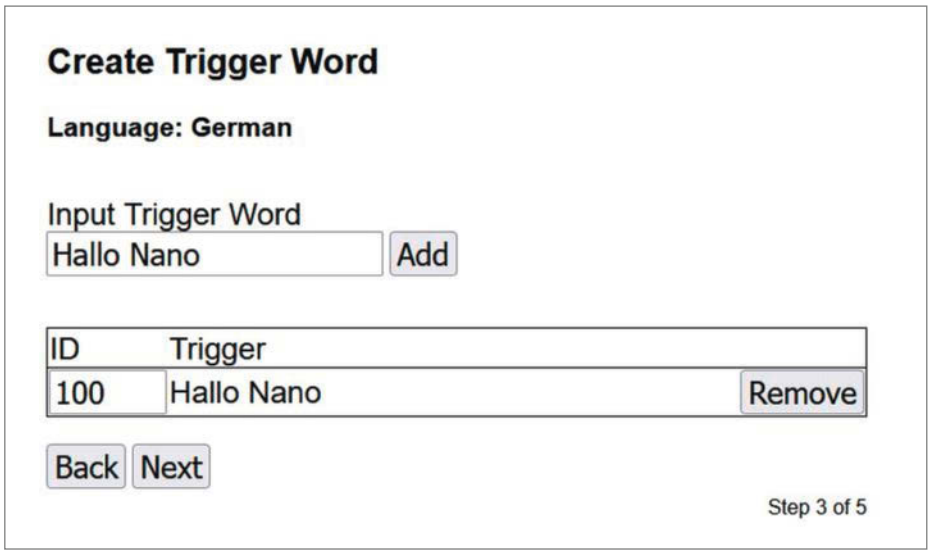


Bild 11: In der Testversion ist nur ein Triggerbefehl möglich.

Bild 12: Die Befehle für die Licherketten-Steuerung

Seite 83). Sie überprüft das vom Spracherkennungsobjekt `DSpotterSDKHL` gelieferte `nFlag`. Das kann sechs Werte annehmen:

```
InitSuccess
GetResult
ChangeStage
GetError
LicenseFailed
LostRecordFrame
```

Wird `InitSuccess` unmittelbar nach dem Start gemeldet, ist alles in Ordnung und keine weitere Aktion notwendig. Bei `ChangeStage` wurde entweder das Triggerwort erkannt und es wird in den Kommando-Modus gewechselt. Oder die Wartezeit im Kommando-Modus ist abgelaufen und es geht zurück in den Trigger-Modus.

`GetResult` bedeutet, dass ein Kommando erkannt wurde. In `nID` steht dann die ID des entsprechenden Befehls. In der dann folgenden `switch`....`case`-Struktur wird dann je nach ID das Entsprechende getan. Im Beispiel folgten im Block nach `case COMMAND_ROT_BLINKEN` die Anweisungen, um die LED rot blinken zu lassen, und so weiter.

Die drei übrigen Möglichkeiten weisen auf Fehler hin: bei `LicenseFailed` stimmt etwas mit der Lizenzdatei nicht, `LostRecordFrame` bedeutet, dass der Nano zu beschäftigt war, um zuhören zu können. Vermutlich ist dann eine der durch Sprachbefehl ausgelösten Funktionen zu zeitintensiv. Auch ein zu langer `delay`-Befehl kann das auslösen. `GetError` schließlich steht für alle anderen Fehler, die im Test aber nicht auftraten.

Fazit

Alles funktionierte richtig gut und tatsächlich offline lokal auf dem Arduino. Das Training neuer Befehlsworte gehört ja nicht zur eigentlichen Spracherkennung. Sogar mein im Hintergrund in den Wänden schlagbohrender Nachbar brachte den Nano nicht aus der Fassung. Das Training auf der Herstellerseite ist einfach, es stehen etwa 40 Sprachen zur Verfügung. Die Programmierung ist auch zu schaffen, wenn man sich die Beispiel-Sketche als Grundlage nimmt. Und mit 16 MB Speicher steht auch größeren Projekten als der hier gezeigten LED-Steuerung nichts im Wege.

Alle Befehle erkannte das kleine System schnell und fehlerfrei. Der geringe Stromverbrauch macht ihn sogar für den Akkubetrieb geeignet. Allerdings sollte man wirklich die 9 Euro für die Volllizenz ausgeben, denn der Countdown vor dem nächsten Triggerwort nervt erheblich. Meine Meinung daher: Da kann man einiges mit machen. Falls Sie auch der Meinung sind und sich damit Loriots Jodeldiplom-Sketch entsprechend „etwas eigenes" schaffen, berichten Sie uns doch bitte darüber. *—hgb*

© Copyright by Maker Media GmbH.

Die Auswerte-Funktion

```
void VRCallback(int nFlag, int nID, int nScore, int nSG, int nEnergy)
{
  if (nFlag==DSpotterSDKHL::InitSuccess)
  {
      //ToDo
  }
  else if (nFlag==DSpotterSDKHL::GetResult)
  {
     switch(nID)
      {
        case COMMAND_ROT_BLINKEN:
          //LED_RGB_Green();
            Serial.println("rot blinken");
            for(int i=0; i<10; i++) {
              analogWrite(LEDR, 0);
              delay(200);
              analogWrite(LEDR, 255);
              delay(200);
            }
          break;
        case COMMAND_GRUEN_BLINKEN:
            Serial.println("grün blinken");
            for(int i=0; i<10; i++) {
              analogWrite(LEDG, 0);
              delay(200);
              analogWrite(LEDG, 255);
              delay(200);
            }
          break;
        case COMMAND_BLAU_BLINKEN:
            Serial.println("blau blinken");
            for(int i=0; i<10; i++) {
              analogWrite(LEDB, 0);
              delay(200);
              analogWrite(LEDB, 255);
              delay(200);
            }
          break;
        case COMMAND_REGENBOGEN:
            Serial.println("Regenbogen");

            for(int i=0; i<10; i++) {
              for(int y=0;y<255;y++) {
                analogWrite(LEDG, R);
                analogWrite(LEDB, G);
                analogWrite(LEDR, B);
                G=G+GI;
                B=B+BI;
                if(R>254) RI=-1;
                if(G>254) GI=-1;
                if(B>254) BI=-1;
                if(R==0) RI=1;
                if(G==0) GI=1;
                if(B==0) BI=1;
                delay(10);
              }
            }
            analogWrite(LEDR, 255);
            analogWrite(LEDG, 255);
            analogWrite(LEDB, 255);
          break;
        case COMMAND_WEISSES_LICHT:
          Serial.println("weisses Licht");
          analogWrite(LEDB, 0);
          analogWrite(LEDR, 0);
          analogWrite(LEDG, 0);
          delay(2000);
          analogWrite(LEDB, 255);
          analogWrite(LEDR, 255);
          analogWrite(LEDG, 255);
          break;
        default:
          break;
      }
  }
  else if (nFlag==DSpotterSDKHL::ChangeStage)
  {
  ...
```

© Copyright by Maker Media GmbH.

Know-How statt Hype

Mit KI-Tools effektiv arbeiten

Heft + PDF mit 29 % Rabatt

Die Nachrichten über revolutionäre KI-Lösungen überschlagen sich täglich. Wie soll man da den Überblick behalten? Mit Tests und Praxistipps erklären wir im c't-Sonderheft, was heute schon geht sowie Ihnen bei der Arbeit hilft und wo Sie den Maschinen noch Zeit zum Reifen geben sollten.

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

shop.heise.de/ct-chatgpt

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

heise Shop

Communicator für Home Assistant

Star-Trek-Fans wissen, worum es geht: Man berührt das Sternenflotten-Enblem auf der Brust und schon kann man mit Gott und der Welt kommunizieren. So etwas ähnliches ist auch dieses Projekt. Allerdings ist die Reichweite aufs Smarthome begrenzt.

von Heinz Behling

© Copyright by Maker Media GmbH.

Tippt sich Captain Picard einmal schwungvoll auf die Brust, hört der Computer oder sogar die ganze Mannschaft der Enterprise auf ihn. Das können Sie auch, allerdings nicht mit der Enterprise (das bleibt immer noch ein Traum), sondern mit Ihrem Smarthome.

Warum ein Sprachassistent, der erst durch Knopfdruck aktiviert werden muss? Vermutlich aus dem gleichen Grund wie in der Enterprise: Solange der Knopf nicht gedrückt wurde, kann niemand mithören. Es ist also nicht wie bei Alexa und Co, also bei Systemen, die ständig lauschen, weil sie auf ihr Aktivierungswort warten oder vielleicht auch aus anderen Gründen. Wie auch immer Sie zu diesem Thema stehen (das können Sie übrigens gerne im Leserforum zu diesem Artikel mit anderen diskutieren): Dieses Projekt hört erst dann zu, wenn Sie es wollen, und steuert dann ihr Smarthome. Allerdings erfolgt auch hier die eigentliche Spracherkennung online in einer Cloud, genauer gesagt in der Cloud von Nabu Casa, der Firma, die hinter der Entwicklung von Home Assistant steckt. Außerdem ist noch KI-Experte OpenAI, bekanntgeworden durch ChatGPT, mitbeteiligt. Dafür ist ein kostenpflichtiger Zugang zur Home-Assistant-Cloud erforderlich. Wie Sie den einrichten, erfahren Sie im Artikel „TV und Heimkino mit Home Assistant" in der Make 5/21.

Gehirn des Sprachassistenten daheim im Smarthome ist ein ESP32 in Form des Atom Echo von M5Stack (Bild 1). Mehr über dieses kleine Board erfahren Sie im Kasten „M5 Atom Echo" auf Seite 88.

Ursprünglich hatte ich ein anderes Board vorgesehen, nämlich ein ESP32 One von Waveshare (Bild 2), weil es bereits ein Mikrofon onboard besitzt. Allerdings müsste dann noch ein Verstärker plus Lautsprecher dazu. Außerdem ist das Board recht stromhungrig. Und ein Gehäuse besitzt es auch nicht.

Kurzinfo

» Sprachassistent für Home Assistant auf ESP32-Basis
» Aktivierung auf Knopfdruck ohne ständiges Mithören
» Mobil durch Stromversorgung Powerbank

Checkliste

Zeitaufwand:
2 Stunden (zzgl. Druckzeit fürs Gehäuse)

Kosten:
25 Euro

Material

» **M5 Atom Echo**
» **USB-Powerbank 5000mAh** (Einschaltstrom muss unter 30mA liegen)

Mehr zum Thema

» Heinz Behling, Intelligentes Heim mit Home Assistant, Make 1/21, S. 100
» Heinz Behling, Controller-Boards fürs Smarthome, Make 4/23, S. 12
» Heinz Behling, TV und Heimkino mit Home Assistant, Make 5/21, S. 92

Alles zum Artikel im Web unter *make-magazin.de/xc71*

Daher fiel meine Wahl auf den M5 Atom Echo. Dies ist ebenfalls ein ESP32, der zusammen mit Mikrofon, Verstärker und Lautsprecher in einem Gehäuse mit nur 2,4 x 2,4 cm Grundfläche und 1,7 cm Höhe sitzt. Einen Aktivierungstaster hat er ebenfalls schon eingebaut und auch eine LED, die die Aktivierung anzeigen kann. Daher habe ich mich dann doch für dieses Board entschieden. Die Stromversorgung erfolgt über seine USB-C-Buchse. Falls Sie den Sprachassistenten stationär (etwa auf dem Wohnzimmertisch) benutzen möchten, reicht ein USB-Netzteil mit 1000mA zur Versorgung völlig aus. Wer es mobil mag, sollte sich eine USB-Powerbank zulegen. Kleine Modelle mit 5000mAh reichen völlig

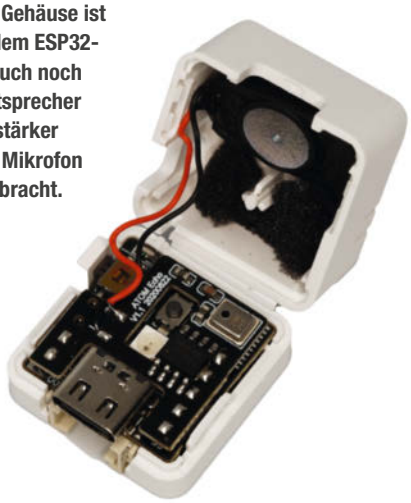


Bild 1: In dem kleinen Gehäuse ist neben dem ESP32-Board auch noch ein Lautsprecher mit Verstärker und ein Mikrofon untergebracht.

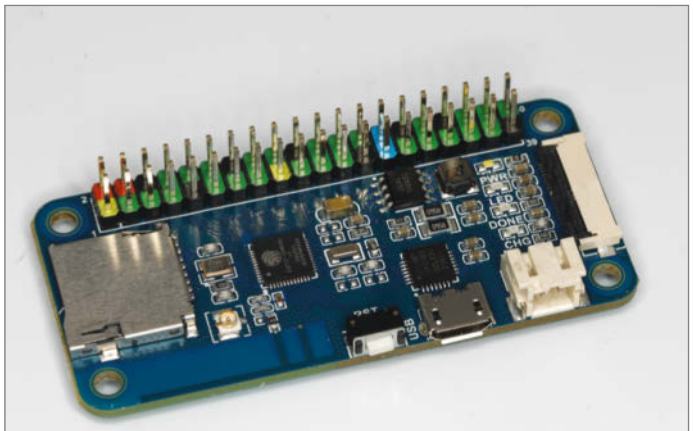


Bild 2: Das war die erste, aber nicht die endgültige Wahl. Das ESP32 One besitzt ebenfalls ein onboard-Mikrofon, aber keinen Verstärker und Lautsprecher.

Bild 3: Die Powerbank mit dem Sprachassistanten

© Copyright by Maker Media GmbH.

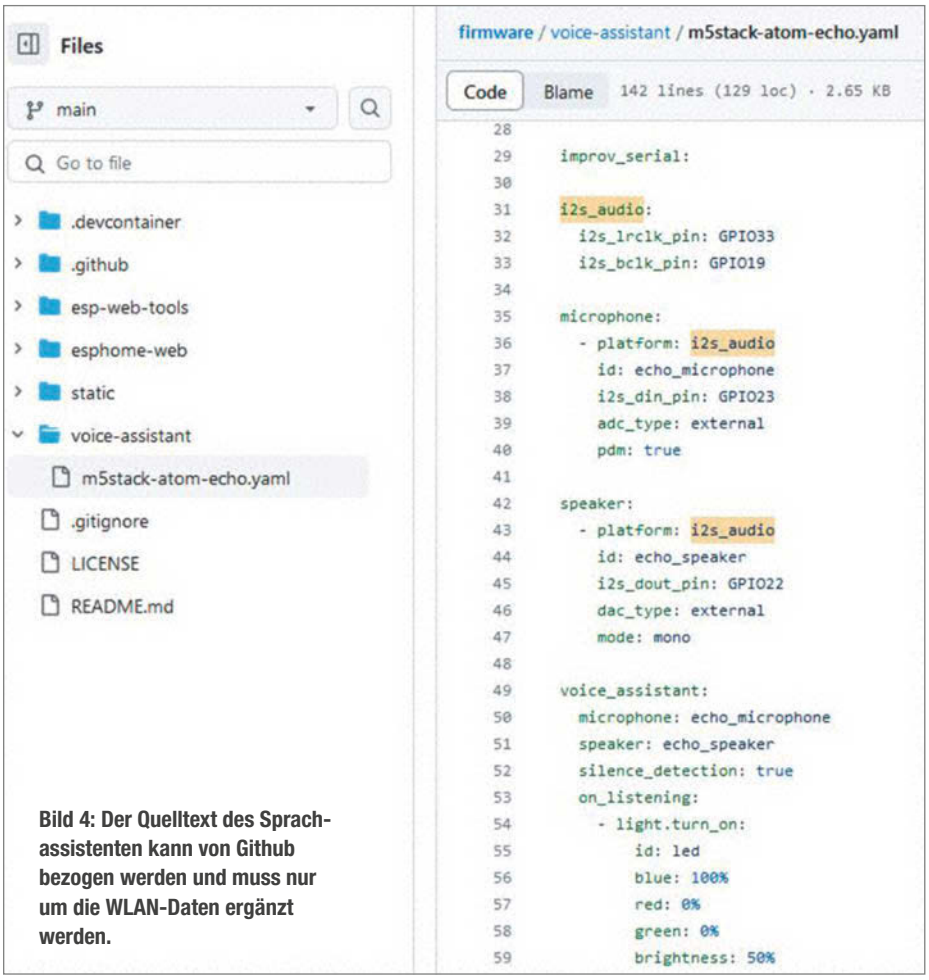


Bild 4: Der Quelltext des Sprachassistenten kann von Github bezogen werden und muss nur um die WLAN-Daten ergänzt werden.

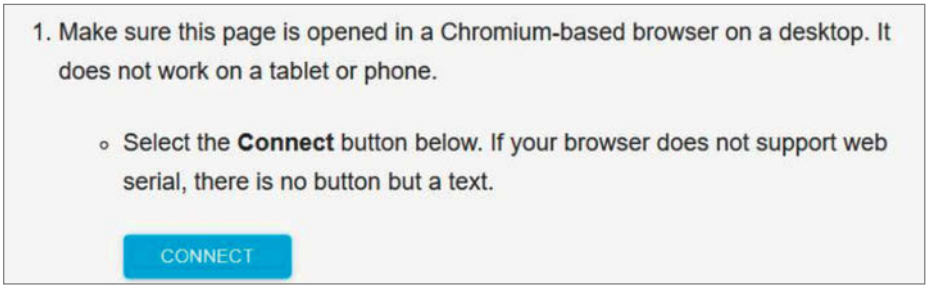


Bild 5: Nur beim richtigen Browser auf dem Computer erscheint dieser Connect-Button.

Bild 6: Hier bitte zuerst auf die Zeile mit der Port-Bezeichnung und dann erst auf Verbinden klicken.

aus, denn der Atom Echo nimmt nur etwa 100mA für einige Sekunden bei Aktivierung auf. Im Ruhezustand sind es etwa 30mA. So reicht die Powerbank etwa eine Woche, bevor sie geladen werden muss. Idealerweise sollte das Powerbank-Modell so klein sein, dass es bequem in eine Hemdtasche passt (Bild 3).

Firmware übertragen per Browser

Die Firmware gibt es fertig im Voicecontrol-Bereich auf den Home-Assistant-Seiten im Netz unter https://www.home-assistant.io/voice_control/thirteen-usd-voice-remote/. Diese Seite sollten Sie mit einem Chrome-kompatiblen Browser auf einem PC öffnen, also entweder mit Chrome selbst oder Microsofts Edge. Auf einem Tablet oder Smartphone funktioniert die im Folgenden beschriebene Installation nicht. Tipp: Wer sich mit ESPHome gut auskennt, kann den Quellcode der Firmware auch auf Github erhalten (Adresse über Kurzinfo-Link). Die Firmware enthält den kompletten Programmcode inklusive der benutzten Ports. Es fehlen aber natürlich die WLAN-Daten (Bild 4).

Hier jedoch gehen wir den einfacheren Weg über den Browser. Verbinden Sie das Atom-Echo-Modul per USB-C-Kabel mit dem Computer. Auf der Internetseite klicken Sie dann auf „Connect" (Bild 5). Sollte dieses Feld nicht sichtbar sein, haben Sie einen nicht-Chrome-kompatiblen Browser benutzt.

Anschließend müssen Sie den Port wählen, an dem der Atom Echo angeschlossen ist, und die Wahl mit einem Klick auf „Verbinden" bestätigen (Bild 6).

Mit „Install Voice Assistant" und der Bestätigung im daraufhin folgenden Fenster „Confirm Installation" geht es weiter. Daraufhin wird der Speicher des ESP32 zunächst gelöscht und dann die neue Firmware installiert. Das dauert ein bis zwei Minuten.

Ist die Installation komplett, geht es mit „Next" weiter zur Eingabe der WLAN-Zugangsdaten (Network-SSID, Password). Die SSID können Sie aus der Liste der vom Modul erkannten Funknetze auswählen. Das Passwort

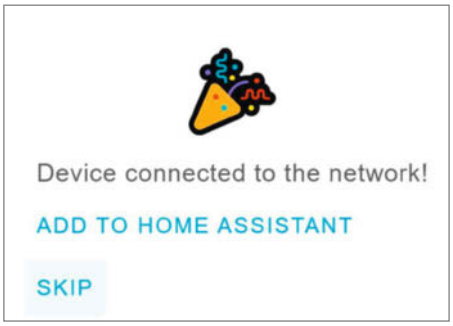


Bild 7: Sobald die WLAN-Verbindung aktiv ist, wird dies auch gemeldet.

© Copyright by Maker Media GmbH.

Bild 8: Der Atom Echo wurde vom Server erkannt.

wie üblich eingeben. Mit „Connect" geht es weiter. Danach kommt ein Klick auf „Add to Home Assistant" (Bild 7).

Dann wird versucht, Sie auf die Web-Oberfläche des Home-Assistant-Servers weiterzuleiten. Das müssen Sie mit einem Klick auf „Open link" und der Eingabe Ihres Home-Assistant-Benutzernamens und Passworts erlauben.

Der Server hat bereits gemerkt, dass es ein neues Gerät gibt. Daher möchte er die ESPHome-Einrichtung starten. Erlauben Sie das mit „OK". Das neue Gerät wird angezeigt (Bild 8). Klicken Sie darauf und beantworten Sie die dann folgende Frage mit Absenden.

Da der neue kleine Sprachassistent ja mobil ist, wäre die Eingabe einer Bereichsangabe (etwa Wohnzimmer) im abschließenden Erfolgsfenster nicht sinnvoll. Schließen Sie das Fenster einfach mit „Fertig".

Jetzt ist der Atom Echo im Prinzip bereits einsatzfähig und zwar überall, wo eine USB-Stromquelle zur Verfügung steht. Für die mobile Nutzung ist eine Powerbank empfehlenswert. Dabei genügt eine mit 5000mAh völlig (siehe Kurzinfo). Er kann zum Beispiel schon alle Geräte in Ihrem Smarthome ein- oder ausschalten. Probieren Sie es mal: Drücken Sie den Taster auf dem Echo und sprechen Sie „Alle Lampen einschalten". Dann wird es hell um sie herum. Mit „Alle Lampen ausschalten" wird es entsprechend wieder dunkel. Auch einzelne Geräte können bedient werden, beispielsweise mit „Schalte die Nachtischlampe ein". Nach jeder Aktion sagt Ihnen der Sprachassistent dann noch, was er getan hat, also zum Beispiel „Nachtischlampe ausgeschaltet".

Einige Dinge kann er aber noch nicht. Haben Sie etwa Skripte angelegt, die mehrere Sachen zusammen erledigen, dann können Sie diese Skripte (noch) nicht mit dem Sprachassistenten starten. Aber das lässt sich ändern. Die vom Assistenten gehörten Worte können Sie nämlich als Trigger für Automationen einsetzen. Ich habe beispielsweise ein Skript mit dem Namen „Kino einschalten", das die Fenster

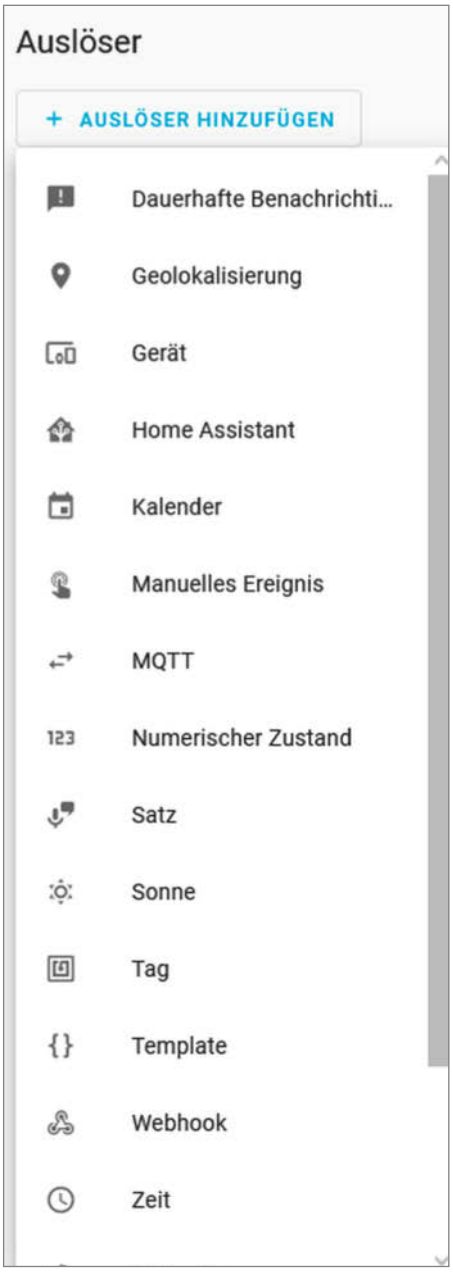


Bild 9: Jetzt können auch gesprochene Sätze als Auslöser in Automationen benutzt werden.

© Copyright by Maker Media GmbH.

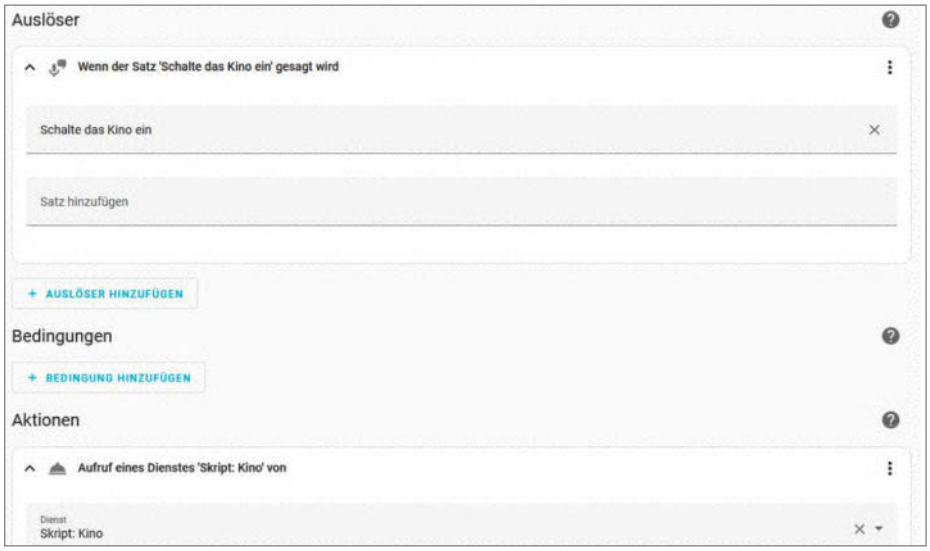


Bild 10: Ein auslösender Satz und eine Aktion müssen mindestens in einer Sprachassistenten-Automatisierung stehen.

verdunkelt sowie den Beamer und die Soundanlage einschaltet. Im Folgenden zeige Ich Ihnen, wie Sie eine Automation anlegen, mit der Sie per Atom Echo das Skript aufrufen.

Startpunkt ist auf der Home-Assistant-Oberfläche unter „Einstellungen/Automatisierungen & Szenen". Mit einem Klick auf „+ Automatisierung erstellen/Neue Automatisierung erstellen" geht es weiter. Nach einem Klick auf „Auslöser" wählen Sie „Satz" (Bild 9).

Als nächstes geben Sie den Text des neuen Befehlssatzes ein, zum Beispiel „Schalte das Kino ein". Jetzt braucht es nur noch die Aktion, die ausgeführt werden soll, wenn der Satz gehört und erkannt wurde. In meinem Beispiel ist es die Ausführung eines Dienstes und der Name des Dienstes ist Skript: Kino (Bild 10).

Ganz unten im Automatisierungsfenster steht das Feld „Antwortvariable". Wenn Sie das Kästchen davor markieren, können Sie danach den Text eingeben, den der Assistent nach getaner Arbeit sprechen soll, also beispielsweise „Kino ist eingeschaltet" (Bild 11).

In der Automatisierung können Sie auch mehrere Aktionen und für jede davon einen eigenen Antwortsatz eingeben. Möchten Sie, dass nicht nur auf genau den Satz „Schalte das Kino ein" sondern beispielsweise auch auf „Bitte schalte das Kino ein" gehört wird, geben Sie den zweiten Satz einfach unterhalb des ersten ein (Bild 12).

Dann ist nur noch ein Klick auf „Speichern" und die Eingabe eines Namens für die Automatisierung (z. B. „Sprachbefehl Kino") erforderlich, und der Sprachassistent versteht den neuen Satz. Der Befehls-Wortschatz des kleinen Boards ist somit nahezu unbegrenzt erweiterbar.

Was jetzt noch fehlt, wäre ein Gehäuse ähnlich des Startrek-Communicators von Captain Picard. Vielleicht hat ja jemand von Ihnen Lust, ein entsprechendes Emblem mit Unterbringungsmöglichkeit für den Atom Echo zu entwerfen. Wir sind neugierig auf Ihre Entwürfe. *—hgb*

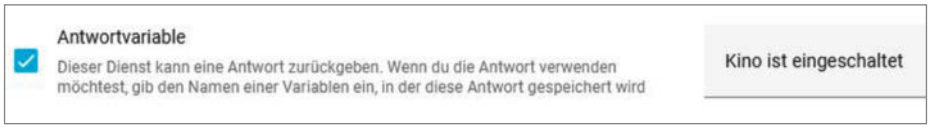


Bild 11: Mit diesen Angaben sagt Ihnen der Assistent, was er getan hat.

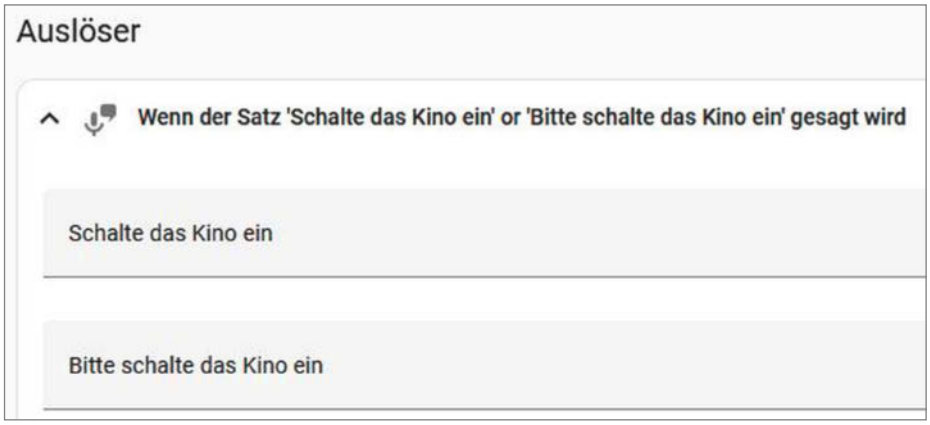


Bild 12: Damit Sie nicht alles wortgetreu wiederholen müssen, können Sie Varianten des Befehlssatzes eingeben.

M5 Atom Echo

Im kleinen weißen Gehäuse sitzen neben einem ESP 32 noch weitere technische Zutaten:

- Taster (an Pin 39)
- RGB-LED
- Infrarot-LED
- NF-Verstärker (über I²S)
- Lautsprecher 0,5 W
- Mikrofon (über I²S)
- I²C-Schnittstelle auf Grove-Buchse nach außen geführt

Außerdem sind die Pins 19, 21, 22, 23, 25, 33, 5V, GND und 3,3V nach außen auf Pinleisten geführt. Unter https://docs.m5stack.com/en/atom/atom-echo?ref=nabucasa gibt es eine recht ausführliche technische Dokumentation zum Board mit einigen einfachen Beispielanwendungen.

© Copyright by Maker Media GmbH.

Horizons #23
by heise

Insights, Inspiration, Networking:
Strategien & Trends für Ihr Business

NEUE DIGITALE ARBEITSWELT

KÜNSTLICHE INTELLIGENZ

CYBER-SICHERHEIT

NEXT GENERATION & RECRUITING

© 3D motion, Adobe Stock

27. NOVEMBER, **Horizons #23 Night**
Cumberlandsche, Hannover

28. NOVEMBER, **Horizons #23 Day**
Künstlerhaus, Hannover

Seien Sie mit dabei!

www.horizons-heise.de

eine Veranstaltung von
Heise Medien

Partner, Sponsoren
chrome · CISO Alliance · DATA academy · hannover impuls · HENCKE · t:n digital pioneers · UVN Unternehmerverbände Niedersachsen e.V.

© Copyright by Maker Media GmbH

Bildschirmspiel 1

Der Konsolenmarkt ist heute sehr übersichtlich und wird im Prinzip durch Sony, Microsoft und Nintendo abgedeckt. Das war aber nicht immer so. In den 70ern begann der Videospiel-Hype und viele sprangen auf. Sogar die DDR.

von Daniel Schwabe

Im Jahr 2021 betrug der erwirtschaftete Umsatz der Software- und Videospielindustrie in Deutschland 56,7 Milliarden Euro. Die Videospielindustrie ist mittlerweile riesig. Aber wie andere Bereiche auch, hat auch sie mal klein angefangen.

Bei diesen kleinen Anfängen in den 70ern wollte natürlich jeder etwas vom Kuchen abhaben. Das führte dazu, dass sogar Unterhaltungsfirmen wie Neckermann Konsolen auf den Markt brachten.

Allerdings haben nicht nur Unternehmen Videospielkonsolen entwickelt. Auch eine Regierung gab den Auftrag dazu. 1979 produzierte die DDR 1000 Einheiten der hauseigenen Videospielkonsole Bildschirmspiel 1 oder kurz: BSS 01. Es handelte sich um 200 weiße und 800 schwarze Exemplare, die

TV-Spiel made in Ostdeutschland

© Copyright by Maker Media GmbH.

hauptsächlich für den Einsatz in Jugendclubs vorgesehen waren.

Vier Spiele konnte man auf dieser Konsole starten: Squash, Fußball, Tennis und Pelota (Letzteres ist in dieser Ausführung Squash für eine Person). Alle waren im Endeffekt eine Abwandlung des bekannten PONG-Spielprinzips, bei denen man mit einem weißen Strich ein weißes Quadrat über den Bildschirm kickt. Angeschlossen wurde das Spielgerät über den Antennenanschluss an ein Fernsehgerät.

Die Schläger wurden mit zwei als „Handregler" bezeichneten Eingabegeräten gesteuert, die jeweils als schwarzes Kästchen mit einem Drehknopf darauf ausgeführt waren.

Hatte man sich zwischen den Spielen entschieden, bot das BSS 01 noch ein paar Modifikatoren für das Spielerlebnis. In zwei Stufen konnten jeweils Ablenkwinkel, Ballgeschwindigkeit und die Schlägergröße eingestellt werden. Sowohl die Wahl der Spiele als auch die Modifikation des Spielerlebnisses erfolgte über Schalter.

Doch wie haben die Ingenieure des Halbleiterwerks Frankfurt (Oder) diese Konsole in der DDR umgesetzt? Indem sie den wohl weit verbreitetsten „All-in-One"-Chip für Videospielkonsolen dieser Zeit verbaut haben. Der AY-3-8500 von General Instrument war seinerzeit in sehr vielen Konsolen integriert. Von Intel bis Universum war der Chip das Herzstück vieler selbstproduzierter oder lizenzierter Konsolen auf dem damaligen Markt.

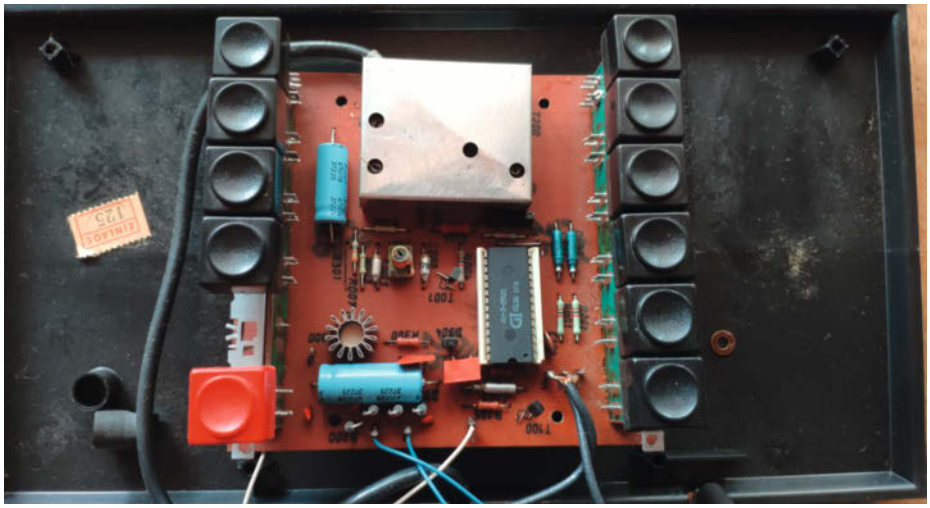
Und das hatte seinen Grund. Der AY-3-8500 brachte alles mit, was man brauchte, um eine eigene Konsole zusammenzubasteln. Es war sogar so leicht, dass seinerzeit in Computermagazinen Anleitungen für den Eigenbau mit diesem Chip veröffentlicht wurden.

Der Blick ins Innere

Schaut man sich die Innereien des Gerätes an, sieht man außer dem großen AY-3-8500 nur ein paar Widerstände, mehr nicht. Nimmt man die Pin-Belegung des Chips zur Hilfe, erkennt man sofort den „All-in-one"-Charakter des Bauteils.

Jedes Spiel, jede Funktion und jeder Modifikator belegt einen Pin am Chip. Im Inneren werden pro Spiel nur bestimmte Teile des Bauteils genutzt. Jede Auswahl hat eigene Logiken. Spätestens jetzt ist klar, dass der AY-3-8500 nicht programmierbar war, sondern alles intern fest verdrahtet wurde.

Auch die Signalerzeugung für die Darstellung auf dem Bildschirm erfolgte auf dem AY-3-8500. Diese Signale werden dann zu einem BAS-Signal weiterverarbeitet. Ebenso die Tonausgabe, für die der Chip drei verschiedene Frequenzen ausgeben konnte, die in unterschiedlicher Länge über einen Lautsprecher im Gerät abgespielt wurden.



Neben dem HF-Modulator ist es ganz schön leer hier. Der AY-3-8500 brachte alles mit, was man brauchte.

Keine Taubenjagd im BSS 01

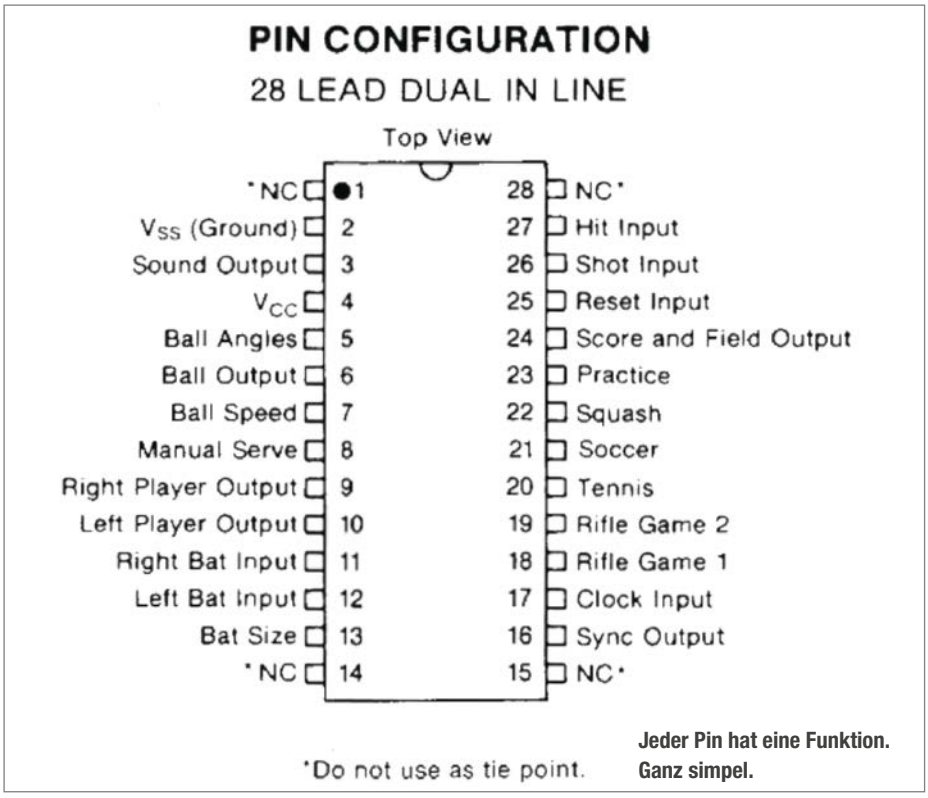
Aufmerksame Beobachter haben auf der Pin-Übersicht bestimmt direkt den Punkt „Rifle Game" gesehen. Tatsächlich verfügte der GI-Chip über mehr als die im BSS 01 angebotenen vier Spiele. Mithilfe von sogenannten Light Guns konnten andere Konsolen noch zwei zusätzliche Schießspiele anbieten. Dabei fliegen weiße Tontauben-Pixel über den Bildschirm und müssen mit einer Plastikkanone abgeschossen werden.

Wie in den Bildern zu sehen ist, verfügte das BSS 01 nicht über solche Lightguns. Dementsprechend wurde die Auswahl für diese beiden Spiele weggelassen.

Der Nachzügler

Im Jahr 1986 begann in der DDR die Produktion eines kompletten Videospielautomaten. Der Poly-Play basierte auf einem Z80-Nachbau und wurde in einer Auflage von 2000 Stück produziert. Er war auch für die öffentliche Nutzung in Jugendclubs oder Ferienheimen vorgesehen und bot mehr Spielspaß. *—das*

▶ make-magazin.de/xk5d



Jeder Pin hat eine Funktion. Ganz simpel.

http://www.pong-story.com/GIMINI1978.pdf

© Copyright by Maker Media GmbH.

Tipps & Tricks

Diesmal gibt es einen Basteltipp für einen Touchsensor, der nur einen Cent kostet, und ein paar Tipps, wie man die vielen USB-Geräte, die wir Maker benutzen, in den Griff bekommt.

von Michael Gaus und Miguel Köhnlein

Touchsensor für 1 Cent

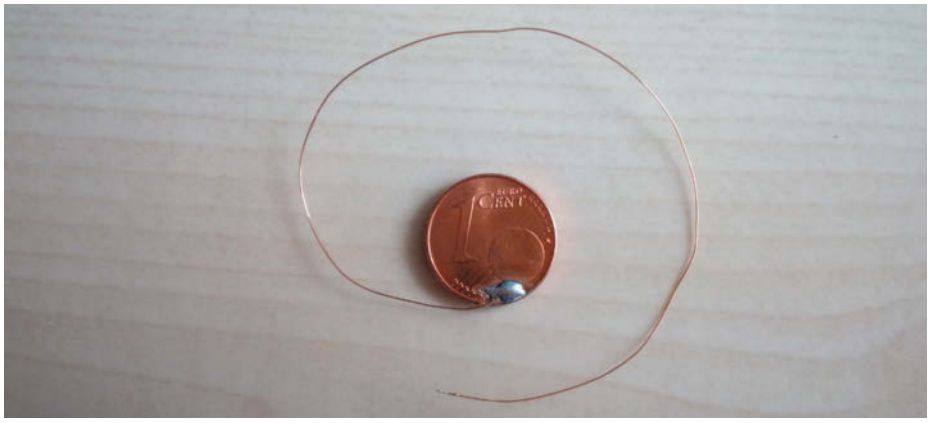
Ein Touchsensor für PICAXE-Mikrocontroller (oder andere Mikrocontroller mit Touchfunktion) ist unschlagbar günstig, denn er kostet buchstäblich nur einen Cent: Man kann einfach eine 1-Cent-Münze verwenden, die in jedem gut sortierten Portemonnaie zu finden sein sollte. Der Sensor muss also nicht online bestellt werden, es fallen keine Versandkosten an und es muss auch nicht auf die Lieferung gewartet werden. Es kann also auch direkt am Wochenende gebastelt werden.

Die Bedienung eines Gerätes erfolgt hier nicht über einen klassischen Taster, sondern durch Berühren eines Touchkontakts mit dem Finger. Hierbei wird die integrierte kapazitive Touchauswertung des PICAXE-Mikrocontrollers ausgenutzt. Wichtig ist, dass der Kontakt elektrisch isoliert ist und nicht direkt mit dem Finger berührt wird. Er kann auf die Innenseite eines Kunststoffgehäuses geklebt werden, z. B. einer großen (leeren) Tic-Tac-Dose. Dies hat einige Vorteile gegenüber einem klassischen Taster: Der Eingabekontakt ist wasserdicht, unempfindlich gegen Schmutz und es sind keine Löcher im Gehäuse erforderlich.

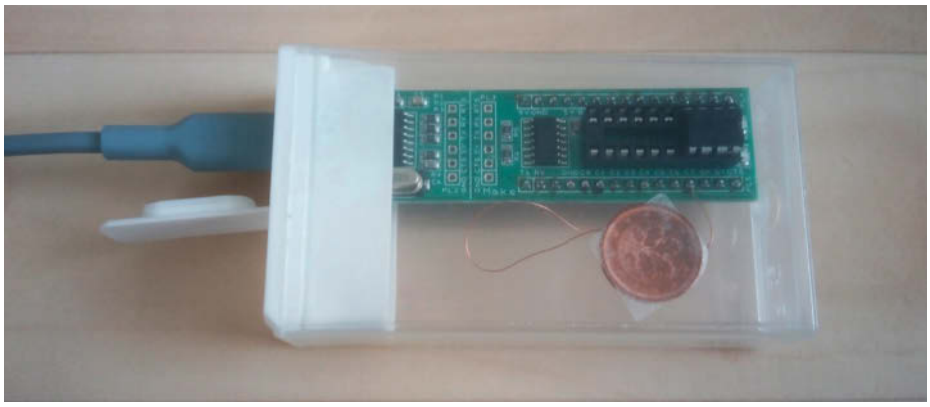
Im Beispiel wird der Touchsensor an Pin C.4 eines PICAXE 08M2 angeschlossen, der auch dem NanoAxe Board beiliegt. Auf dem NanoAxe ist dieser Pin mit C7 beschriftet, da sich die Pinbelegung dort auf den 20M2 bezieht. Weitere mögliche Pins beim 08M2 sind prinzipiell C.1 (auf dem NanoAxe mit B0 beschriftet) und C.2, sodass insgesamt bis zu drei Touchsensoren gleichzeitig an diesen PICAXE-Typ angeschlossen werden könnten. Allerdings funktioniert C.2 beim NanoAxe nicht, da dort die On-Board-LED angeschlossen ist, die die Messung stört.

Zuerst wird ein Stück Kupferlackdraht (oder Fädeldraht) auf eine 1-Cent-Münze gelötet. Dabei muss man mit dem Lötkolben sehr heiß arbeiten und unbedingt auf eine hitzebeständige Unterlage achten!

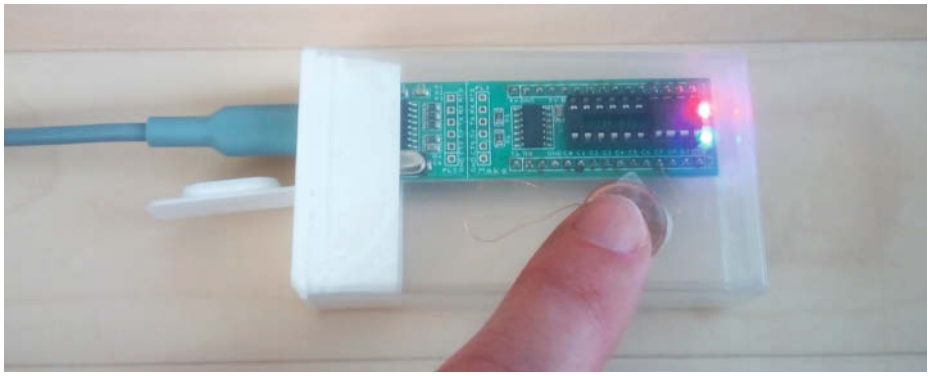
Nach dem Abkühlen wird die andere Seite der Münze mit doppelseitigem Klebeband an der gewünschten Stelle in das Kunststoffge-



An der 1-Cent-Münze wird ein Fädeldraht angelötet.



Die Münze ist mit transparentem Doppelklebeband innen im Gehäuse festgeklebt.



Beim Berühren des Sensors leuchtet die blaue On-board-LED auf.

© Copyright by Maker Media GmbH.

häuse geklebt. Da der Münzrand leicht erhöht ist, sollte etwas dickeres Doppelklebeband verwendet werden. Durch Auftragen von Heißkleber oder einem anderen geeigneten Klebstoff rund um die Münze wird das Cent-Stück dann dauerhaft im Gehäuse fixiert.

Auf der Außenseite des Gehäuses kann an der entsprechenden Stelle ein passendes Symbol aufgeklebt werden.

In der Firmware für den PICAXE wird beim Start im Unterprogramm `touch_calibrate` zunächst eine automatische Kalibrierung des Touchsensors durchgeführt, indem der Ruhepegel des unbetätigten Sensors gemessen wird. Dazu werden in einer Schleife mit den Befehlsaufrufen `touch16` insgesamt zehn Touchwerte nacheinander eingelesen, wobei jeder Wert mit dem Vorgängerwert gemittelt wird, um einen stabilen Messwert zu erhalten.

In der Variablen `touchThreshold` steht nach Durchlaufen der Schleife zunächst der gemittelte Ruhepegel des nicht gedrückten Touchsensors, der nach der `for`-Schleife um 300 erhöht wird, um den Schwellenwert für einen aktiven Touchkontakt festzulegen. Abhängig von den verwendeten Materialien und den Abständen für die Montage des Touchsensors kann eine individuelle Anpassung erforderlich sein. Die automatische Kalibrierung ist sinnvoll, da die Touchanalyse von äußeren Einflüssen wie elektromagnetischen Feldern, Umgebungstemperatur etc. abhängig ist. Durch Aus- und Wiedereinschalten ist eine erneute Kalibrierung jederzeit möglich, falls die Toucherfassung nicht mehr einwandfrei funktioniert.

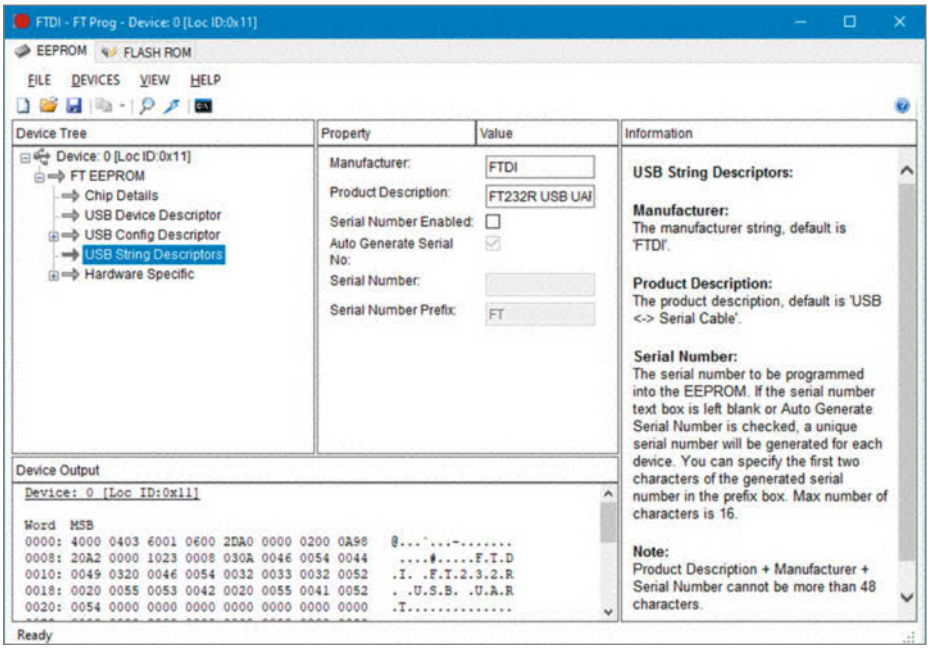
Im Beispielquellcode wird in der Schleife `loop1` zyklisch mit dem Befehl `touch16` der Sensorwert eingelesen und geprüft, ob dieser größer ist als der in der Variablen `touchThreshold` gespeicherte Schwellwert. Ist dies der Fall, so wurde eine Betätigung des Touchsensors erkannt und es wird eine Meldung auf der seriellen Schnittstelle zusammen mit dem Messwert ausgegeben. Zusätzlich wird 100 ms gewartet und während dieser Zeit die On-Board-LED des NanoAxe aktiviert, sodass eine Touchauslösung direkt visuell wahrgenommen werden kann. Danach beginnt die Schleife wieder von vorne. So kann man experimentell den Erfassungsbereich des Sensors bestimmen und erkennen, welchen Einfluss z. B. Handschuhe auf das Messergebnis haben.

Listing: Beispielcode

```
#picaxe 08M2
#terminal 4800
symbol TOUCHPIN = C.4
symbol LED = C.2
#define LED_OFF high LED
#define LED_ON  low LED
symbol cnt = b0
symbol touchValue = w1
symbol touchThreshold = w2

main:
   pause 2000 ; give Terminal some time to open
   LED_ON
   sertxd("Calibrating...", cr, lf);
   gosub touch_calibrate
   sertxd("Threshold: ", #touchThreshold, cr, lf);
   LED_OFF
loop1:
   touch16 TOUCHPIN, touchValue
   if touchValue > touchThreshold then
      sertxd("Touch detected: ", #touchValue, cr, lf);
      LED_ON
      pause 100
      LED_OFF
   endif
goto loop1

touch_calibrate:
; read initial/calibration touch value when untouched
   touch16 TOUCHPIN, touchThreshold
   for cnt = 1 to 10
      touch16 TOUCHPIN, touchValue
      touchThreshold = touchThreshold + touchValue / 2
   next cnt
   touchThreshold = touchThreshold + 300 ; level for "detected"
return
```

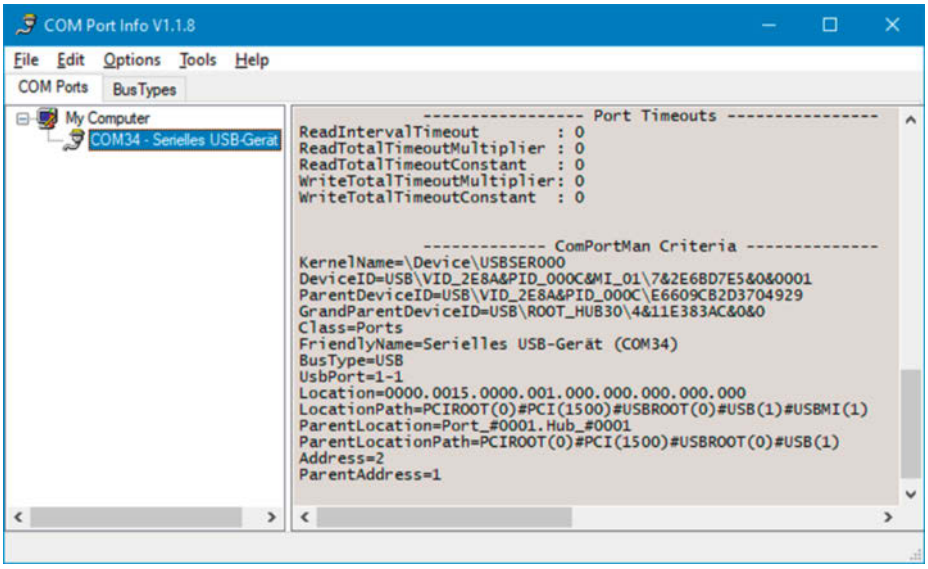
COM-Port mit physikalischem USB-Port verknüpfen

Viele Geräte und Boards, die der Maker verwendet, kommunizieren per USB-Schnittstelle über einen virtuellen COM-Port. Häufig werden dafür USB-UART-Wandlerchips verwendet, z. B. von FTDI. Steckt man mehrere Geräte mit dem gleichen Konverter-Chip nacheinander an den gleichen physikalischen USB-Port eines PCs, so vergibt Windows in den meisten Fällen für jedes Gerät eine neue COM-Port-Nummer. Bei vielen Adaptern mit gleichem Chipsatz kann das schnell lästig werden. Praktisch wäre es, wenn sich diese Geräte immer mit der gleichen virtuellen COM-Portnummer am selben physikalischen USB-Port anmelden würden.

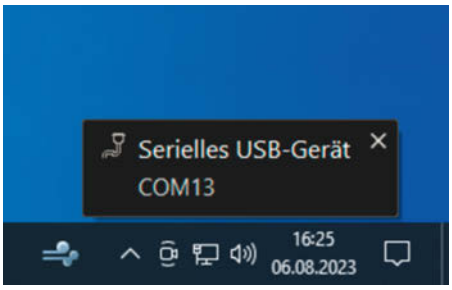


FTDI-Programmierapp

© Copyright by Maker Media GmbH.



Das Tool COM Port Info zeigt mehr Informationen als Windows.



Nie mehr den passenden COM-Port suchen.

Windows unterscheidet Geräte anhand der Vendor ID (VID), der Product ID (PID) und der individuellen Seriennummer. Anhand dieser drei Kriterien wird ein Gerät identifiziert, unabhängig davon, an welchem USB-Port es angeschlossen ist. Beim ersten Anstecken vergibt Windows die nächste freie COM-Port-Nummer. Es ist aber auch erlaubt, auf die Seriennummer zu verzichten. In diesem Fall merkt sich Windows neben VID und PID auch den USB-Port, an dem das Gerät eingesteckt wurde. Wird ein anderes Gerät mit gleicher VID und PID ohne Seriennummer am selben physikalischen USB-Port erkannt, so erhält dieses Gerät keine neue COM-Portnummer, sondern die diesem Port bereits zugewiesene Nummer. Wird ein solches Gerät ohne Seriennummer an einem anderen USB-Port angesteckt, erhält es eine neue COM-Portnummer, die dem entsprechenden anderen USB-Port zugeordnet ist. Wenn es also gelingt, die Seriennummer abzuschalten, ist das Ziel erreicht.

Je nach Chipsatz gibt es die Möglichkeit, dies direkt in der Chip-Hardware zu konfigurieren, was den Vorteil hat, dass auf anderen Ziel-PCs keine zusätzliche Software installiert werden muss, wenn sich die Geräte dort gleich verhalten sollen. Es gibt aber auch eine universelle Softwarelösung, die als Windows-Dienst auf jedem Ziel-PC installiert werden kann.

Die verbreiteten CH340G-Chips haben ab Werk keine Seriennummer, sodass sie sich bereits so verhalten, wie wir es uns wünschen. Werden z. B. zwei Arduinos mit CH340G nacheinander an den gleichen USB-Port angeschlossen, vergibt Windows automatisch jeweils die gleiche COM-Port-Nummer, was nicht nur beim Firmware-Upload auf mehrere Boards ein wahrer Segen ist.

FTDI-Chips mit On-Chip-EEPROM (wie z. B. der bekannte FT232R) oder extern angeschlossenem EEPROM haben in der Regel vom Hersteller eine individuelle Seriennummer. Es ist jedoch möglich, diese abzuschalten. Dazu gibt es das Tool FT_Prog des Herstellers. Am besten schließt man zunächst nur ein Gerät mit FTDI-Chip an den PC an, um Verwechslungen zu vermeiden und geht wie folgt vor:

- FT232R an USB anschließen. Da bei einem fabrikneuen FT232R die Seriennummer aktiv ist, meldet sich dieser zunächst als virtueller COM-Port mit der nächsten freien COM-Port-Nummer an.
- Tool FT_PROG starten. Achtung: Der virtuelle COM-Port des FT232R darf nicht durch ein Terminalprogramm oder eine andere Anwendungssoftware geöffnet sein.
- Unter Devices auf „Scan and Parse“ klicken.
- Auf „USB String Descriptors“ klicken und den Haken bei „Serial Number Enabled“ entfernen.
- Bei Devices auf Program klicken, dann wieder auf Program.
- Nach erfolgreicher Programmierung auf „Cycle Ports“ klicken, dadurch wird der FT232R am USB neu nummeriert. Der FT232R meldet sich nun als virtueller COM-Port entsprechend der zugewiesenen COM-Port-Nummer des verwendeten physikalischen USB-Ports an.

Möchte man die von Windows vergebene COM-Port-Nummer ändern, so hat man dazu im Gerätemanager von Windows die Möglichkeit. Unter „Anschlüsse (COM & LPT)“ öffnet ein Doppelklick auf den zu ändernden COM-Port den Eigenschaften-Dialog. Dort klickt man unter Anschlusseinstellungen auf Erweitert und kann über die Dropdown-Box die COM-Portnummer ändern. Sollten dort noch COM-Port-Nummern auftauchen, die laut Windows bereits verwendet werden, aber in der Praxis nicht mehr benötigt werden, können diese wie folgt aus dem System gelöscht werden: Im Gerätemanager unter Ansicht die Option „Versteckte Geräte anzeigen“ aktivieren. Nun werden unter Anschlüsse alle von Windows vergebenen COM-Port-Nummern angezeigt. Aktuell nicht angeschlossene Geräte werden mit einem grauen statt einem schwarzen Symbol dargestellt. Mit einem Rechtsklick kann das gewünschte Gerät über das Kontextmenü deinstalliert werden.

Für virtuelle COM-Ports, bei denen das direkte Abschalten der Seriennummer im Chip nicht möglich ist, gibt es mit dem Freeware-Tool ComPortMan eine universelle Softwarelösung. Dies funktioniert auch mit Geräten, die virtuelle COM-Ports per USB-CDC zur Verfügung stellen, wie z. B. ein Raspberry Pi Pico. Ein weiteres nützliches Feature ist die Anzeige von Balloontips, sodass man direkt beim Anschließen und Entfernen eines Gerätes über die zugeordnete COM-Port-Nummer informiert wird.

Das Tool läuft als unsichtbarer Windows-Dienst. Zur Installation wird am einfachsten die msi-Datei heruntergeladen und per Doppelklick gestartet. Alternativ kann die zip-Datei genommen, nach C:\Programme entpackt

```
[OnArrival10]
UsbPort=1-1
open="%ProgramFiles(x86)%\teraterm\ttermpro.exe" /c=%ComPortNumber% /BAUD=115200
```

Programme können so beim Erkennen von USB-Geräten automatisch gestartet werden.

© Copyright by Maker Media GmbH.

und anschließend im Installationsverzeichnis die Batch-Datei _service_register.cmd gestartet werden. Bei jedem Systemstart wird der Service dann automatisch geladen.

Die Konfiguration erfolgt über die Datei ComPortMan.ini. Die Einstellmöglichkeiten sind in der Online-Hilfe sehr gut beschrieben. Zum Editieren startet man als Administrator die Batch-Datei _edit_ini.cmd. Falls noch keine ComPortMan.ini Datei vorhanden ist, wird diese mit einer Beispielkonfiguration angelegt. Über verschiedene Abschnitte mit dem Namen [ComPortsXX] können die gewünschten Behandlungen definiert werden, wobei XX eine Zahl zwischen 1 und 99 ist. Über entsprechende Kriterien wird festgelegt, welcher Abschnitt für einen COM-Port zuständig ist. Die verwendbaren Kriterien sind im mitgelieferten Tool namens COM Port Info aufgelistet, z. B. die zugeordnete Nummer für USB-Port oder die entsprechende DeviceID.

Für unseren Anwendungsfall wollen wir jedes Gerät, das an einem bestimmten physikalischen USB-Port angeschlossen ist, auf den noch freien COM13 konfigurieren. Um die benötigte USB-Port-ID herauszufinden, schließen wir beispielhaft einen Raspberry Pi Pico an und wählen per Mausklick den passenden COM-Port, hier COM34, aus. Auf der rechten Seite wird im unteren Bereich „ComPortMan Criteria“ die zugewiesene Nummer für USB-Port angezeigt, hier 1-1.

Nun passen wir die ComPortMan.ini entsprechend an:

```
[ComPorts10]
UsbPort=1-1
PortName=COM13
```

Im Abschnitt `[BalloonTips]` ist bereits vorkonfiguriert, dass beim Anschließen und Entfernen von COM-Ports entsprechende Hinweismeldungen erscheinen. Nach dem Ab- und wieder Anstecken des Geräts ist direkt sichtbar, dass sich der Pi Pico nun wie gewünscht als COM13 meldet.

Im Tool COM Port Info wird nun ebenfalls COM13 angezeigt (sonst Taste F5 drücken). Im rechten Abschnitt wird im oberen Bereich unter „COM Port“ angezeigt, dass der INI-Abschnitt [ComPorts10] hierfür zuständig ist.

Von nun an wird jedes weitere Gerät mit einem virtuellen COM-Port, das an den USB-Port 1–1 angeschlossen wird, von ComPortMan automatisch als COM13 konfiguriert. Es ist in jedem Fall darauf zu achten, dass der gewählte COM-Port nicht schon vorher von Windows verwendet wurde, da es sonst zu Zugriffskonflikten kommen kann, wenn mehreren gleichzeitig angeschlossenen Geräten die gleiche COM-Port-Nummer zugewiesen wird.

Zur Bereinigung der COM-Port-Reservierungen gibt es noch das Tool „COM Name Arbiter“ des gleichen Entwicklers. Damit kann man z.B. auch die bestehende Nummerierung der COM-Ports komplett zurücksetzen, indem man mit dem Button „Remove non-present devices“ alle momentan nicht angeschlossenen COM-Ports deinstalliert.

Ein weiteres interessantes Feature von ComPortMan ist das automatische Starten von Programmen beim Anschließen oder Entfernen von COM-Ports. Dazu gibt es die Sektionen `[OnArrivalXX]` und `[OnRemovalXX]`. Über Variablen kann z.B. auch die jeweilige COM-Port-Nummer übergeben werden. Im folgenden Beispiel (siehe Bild) wird das Terminalprogramm Teraterm mit einer Baudrate von 115200 Baud automatisch gestartet, wenn ein Gerät am UsbPort 1-1 angeschlossen wird. *—caw*

Alles zum Artikel im Web unter *make-magazin.de/x3vr*

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, Upcyling …).

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

Generell portofreie Lieferung für Heise Medien- oder Maker Media Zeitschriften-Abonnenten oder auf einen Einkaufswert von 20 € (innerhalb Deutschlands).
Nur solange der Vorrat reicht. Preisänderungen vorbehalten.

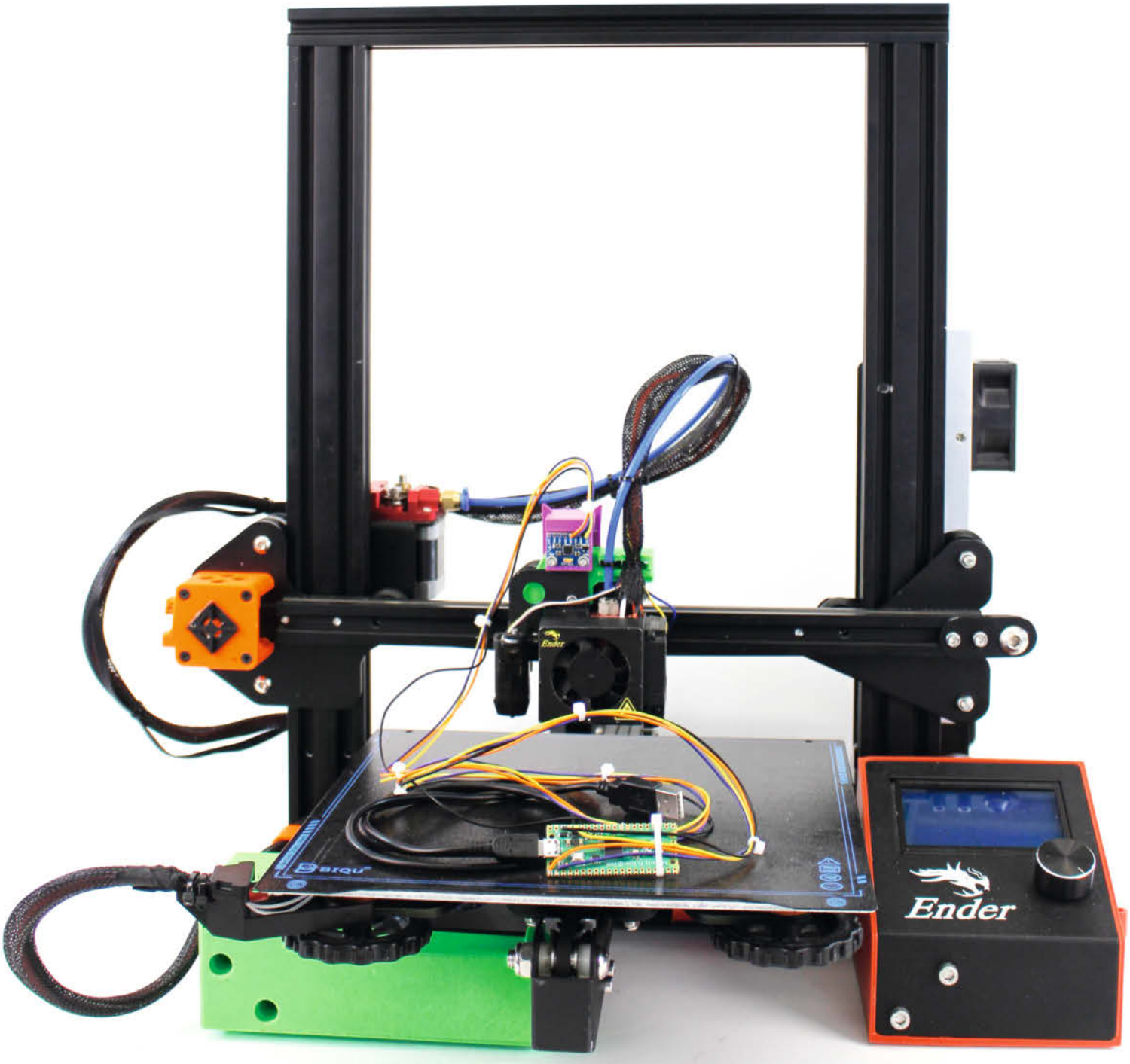
© Copyright by Maker Media GmbH

heise Shop

3D-Drucker mit Klipper optimieren

Eine der wichtigsten Funktionen der 3D-Druck-Software Klipper ist die Kompensation der geräteabhängigen Resonanzfrequenzen, die die Druckqualität verringern und die maximale Geschwindigkeit begrenzen. Hier erfahren Sie, wie man die Frequenzen per Beschleunigungssensor messen und die optimalen Konfigurationseinstellungen ermitteln kann.

von Heinz Behling



© Copyright by Maker Media GmbH.

Falls Sie schon einmal der fast unwiderstehlichen Verlockung nachgegeben und einem 3D-Drucker bei der Arbeit zugeschaut haben, wissen Sie: Hotend und Drucktisch sind ständig in Bewegung, um Schicht für Schicht etwas Neues entstehen zu lassen. Dabei wechseln sie häufig die Richtung, müssen beschleunigen und abbremsen und das auch noch möglichst schnell. Dabei kommt der Printer oft heftig ins Schwingen, was man als Wellenmuster (ringing) erkennen kann. Damit diese Resonanzschwingungen nicht überhandnehmen, arbeiten die Drucker von Haus aus nicht mit der maximal möglichen, sondern einer deutlich geringeren Geschwindigkeit, die eine Kompromisslösung aus Druckqualität und Druckzeit darstellt. Daher dauert ein Druck für gewöhnlich viele Stunden.

Mit der 3D-Druck-Firmware Klipper kann die Druckzeit erheblich verkürzt werden, denn damit kann der Drucker mit erheblich höherem Tempo betrieben werden. Wie man einen Drucker auf Klipper umrüstet, haben Sie bereits in der Make 3/23 erfahren. Das Schwingungsproblem umgeht Klipper dadurch, dass es die den Druck steuernden GCODE-Dateien analysiert und Bewegungen erkennt, die zur Schwingungsanregung führen würden. Die Steuerbefehle für solche Bewegungen werden dann geändert an den Drucker geleitet und so die Resonanzerregung verhindert oder zumindest stark verringert. Diese Funktion heißt input shaper und ist ziemlich rechenintensiv. Deshalb ist für Klipper auch ein leistungsstärkerer Prozessor notwendig, meist wird dazu ein Raspberry Pi benutzt.

Dazu muss Klipper jedoch die Resonanzfrequenzen der X- und Y-Achsen des Druckers kennen (die Z-Achse spielt keine Rolle, da hier beim Druck kaum Bewegungen stattfinden). Dabei genügen keine Werksangaben, denn die Frequenzen sind Geräteindividuell: Vor allem die Gewichte der bewegten Teile, die Reibung in den Lagern und auch die Standfläche des Druckers beeinflussen die Resonanz. Wie Sie diese Frequenzen durch Probedrucke und Messen per Lineal bestimmen können, haben Sie bereits in der Make 3/23 erfahren. Dieses Verfahren ist zwar einfach und genügt bereits, deutlich schneller mit dem Printer zu drucken. Die Genauigkeit ist aber nicht hoch genug, um wirklich das Optimum aus dem Gerät herauszuholen.

Hier zeige ich Ihnen nun, wie Sie die Resonanzen elektronisch mithilfe von Beschleunigungssensoren messen können. Außerdem kann Klipper aus den Messwerten die optimalen Einstellungen für die Konfigurationsdatei bestimmen. Dafür ist ein wenig Lötarbeit erforderlich, um den Sensor mit dem Raspberry zu verbinden. Bevor Sie jedoch mit Löt- und

Kurzinfo

» Resonanzfrequenz messen mit Beschleunigungssensor
» Datenauswertung per Skript
» Druckqualität verbessern Konfigurationseinstellungen für input shaper

Checkliste

Zeitaufwand:
2 bis 3 Stunden

Kosten:
ab 20 Euro

Mehr zum Thema

» Heinz Behling: Sensorloses Homing, Make 1/23, S. 78
» Heinz Behling, Schneller drucken mit Klipper, Make 3/23, S. 10

Material

» Beschleunigungssensor MPU 6050
» Raspberry Pi Pico
» Schaltlitze ca. 5 m
» 3D-Druck-Filament
» 2 Schrauben M3×6
» 1 Schraube M3×16
» 1 Mutter M3

Werkzeug

» Lötausrüstung

Alles zum Artikel im Web unter *make-magazin.de/x4jt*

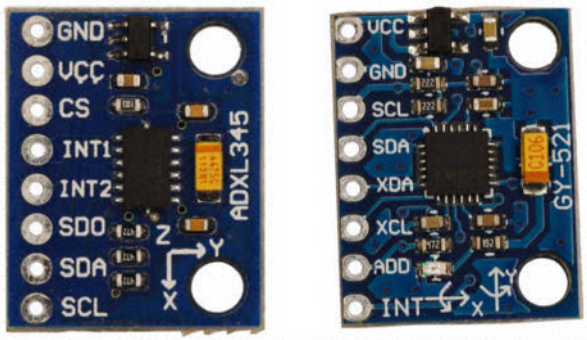


Bild 1: Da die beiden Beschleunigungssensoren ADXL345 (links) und MPU6050 die gleichen mechanischen Abmessungen haben, können auch dieselben Halter benutzt werden.

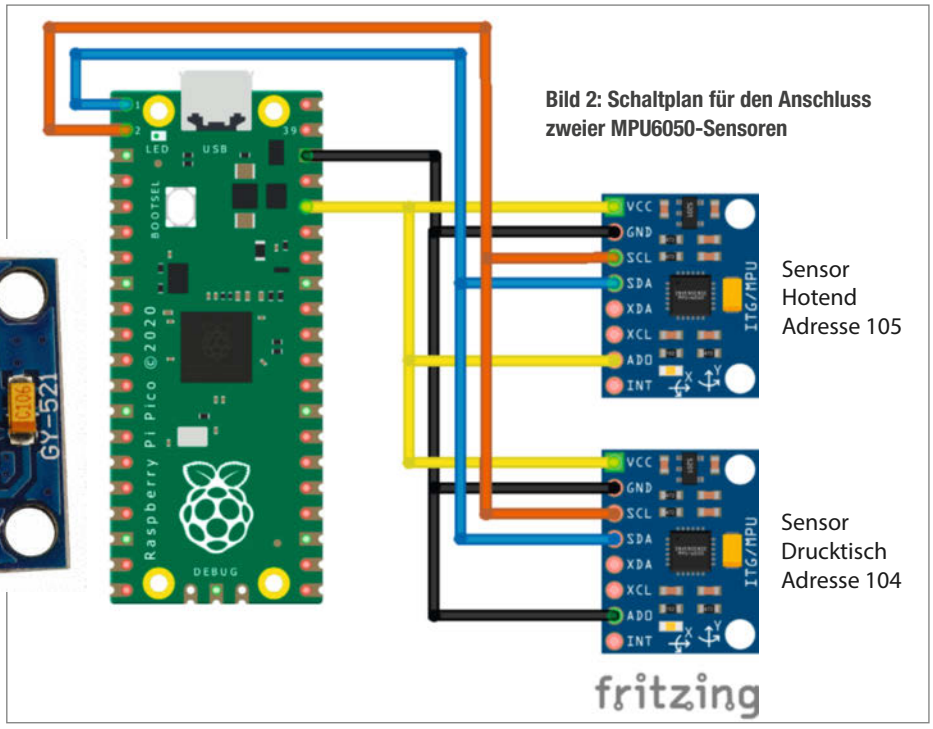


Bild 2: Schaltplan für den Anschluss zweier MPU6050-Sensoren

© Copyright by Maker Media GmbH.

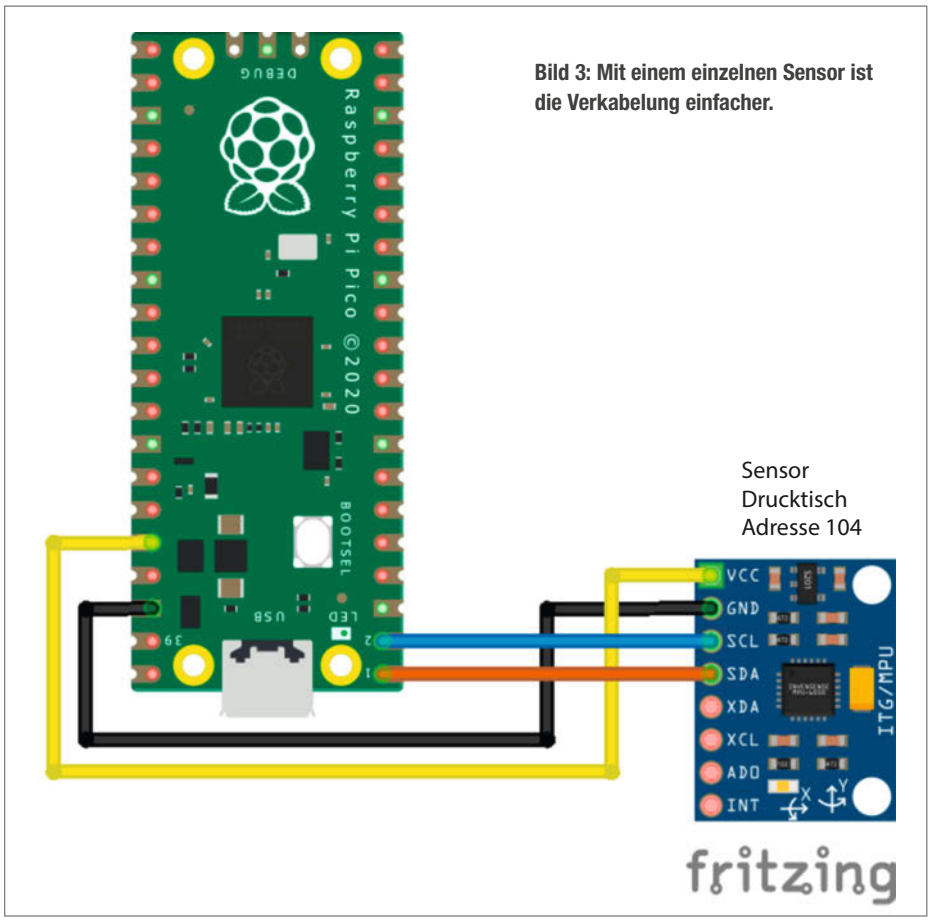


Bild 3: Mit einem einzelnen Sensor ist die Verkabelung einfacher.

Installationsbefehle

```
sudo apt update
sudo apt install python3-numpy python3-matplotlib libatlas-base-dev
~/klipper/venv/bin/pip install numpy
cd ~/klipper
git pull
make clean
make menuconfig
```

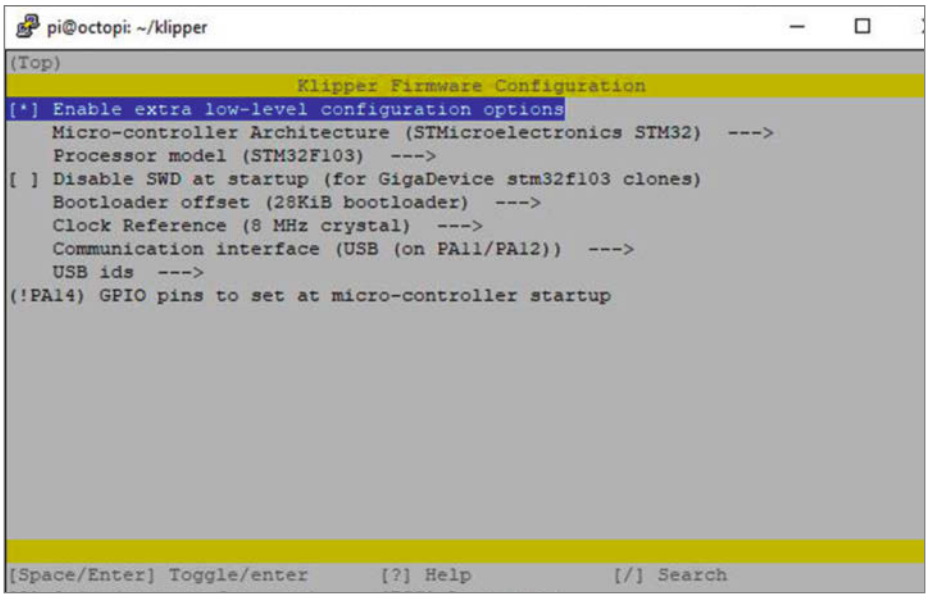


Bild 4: Zur Erinnerung: Dies war die Konfiguration für den Raspi in Ihrem Drucker.

Konfigurationsarbeiten beginnen, drucken Sie bitte den Sensorhalter. Die Druckdateien finden Sie mithilfe des Kurzinfo-Links.

Verbindung zum Beschleunigungssensor

Die Sensoren gibt es auf kleinen, mit Anschlussleisten versehenen Platinen zu kaufen. Für Klipper werden vor allem die zwei Typen mit den Bezeichnungen ADXL345 und MPU6050 (einem MPU9250-kompatiblen Modell) benutzt (Bild 1). Ich habe beide ausprobiert und mich schließlich für den MPU6050 entschieden. Begründung: Beide Sensoren können zwar per SPI- und I²C-Bus mit dem Raspberry verbunden werden. Die I²C-Verbindung des ADXL-Sensors ist aber nicht schnell genug, um die Frequenzen des Druckers zu messen. Daher kommt für ihn nur der (sechspolige) SPI-Anschluss infrage. Der MPU-Sensor hingegen hat eine (vierpolige) Hochgeschwindigkeits-I²C-Verbindung, sodass die Verkabelung einfacher ist. Außerdem kann man die Adresse des Sensors mithilfe einer Lötbrücke ändern, sodass sich am I²C-Bus gleichzeitig zwei Sensoren betreiben lassen (je einen für Drucktisch und Hotend).

Die Sensoren können direkt an den Raspberry angeschlossen werden, der Ihren Drucker steuert. Dazu müsste auf dem aber eine zweite Klipper-Instanz eingerichtet werden, die sich nur um die Frequenzmessung kümmert. Ich habe mich für eine andere Methode entschieden: Um die Sensoren kümmert sich eine eigene Klipper-Installation auf einem Raspberry Pi Pico, der per USB-Kabel an den Drucker-Raspi angeschlossen wird. Vorteil dieser Lösung: Der Pico mitsamt den Sensoren kann jederzeit auch an andere Klipper-Drucker angeschlossen werden, um dort die Frequenzen zu messen. (Sie müssen dann allerdings einige Zeilen in der Konfigurationsdatei auskommentieren, siehe Abschnitt „ohne Pico drucken“). Denn einmal gemessen ändern sich die Frequenzen nicht, solange sie keine Änderungen am Drucker vornehmen, die das Gewicht der bewegten Teile ändern. Die Masse des Sensors inklusive Halter selbst ist gegenüber der sonstigen Mechanik so gering (5 g), dass es keine messbaren Unterschiede bei den Frequenzen mit oder ohne sein Gewicht gibt. Dabei setze ich die Verwendung des über den Kurzinfo-Link bereitgestellten Halters voraus. Falls Sie jedoch mehrere Druckplatten mit unterschiedlichem Gewicht (etwa Stahl- und Glasplatten) verwenden, empfehle ich den Festeinbau zweier Sensoren. So kann bei einem Plattenwechsel der Drucker schneller an die neuen Verhältnisse angepasst werden.

Die Verkabelung der Sensoren mit dem Pico ist in den beiden Schaltplänen zu sehen. Bild 2 zeigt den Anschluss zweier Sensoren an den Pico.

© Copyright by Maker Media GmbH.

Einer der Sensoren hört auf dem I²C-Bus auf die Adresse 104, der andere auf 105. Die Adressänderung wird durch die Beschaltung des A0-Anschlusses bewirkt.

Möchten Sie nur einen Sensor benutzen, ist auch das möglich (Bild 3). Dann bleibt A0 unbeschaltet. Sie müssen dann später nach der Messung der ersten Resonanzfrequenz (zum Beispiel am Druckbett) den Sensor dort entfernen und am Hotend befestigen. Die sonstige Arbeitsweise bei der Messung bleibt gleich. Im Folgenden habe ich die Lösung mit nur einem Sensor benutzt.

Software-Installation

Für den Sensorbetrieb ist einiges an zusätzlicher Software auf dem den Drucker steuernden Raspberry Pi erforderlich. Dazu brauchen Sie eine SSH-Verbindung zwischen Computer und dem Raspi, beispielsweise über das Programm Putty (Download s. Kurzinfolink). In Putty müssen Sie die IP-Adresse des Raspis sowie den Benutzernamen und das dazugehörende Passwort eingeben. Vorgabe sind „pi" und „raspberry", falls Sie bei der Installation nichts anderes eingegeben haben.

Im Terminal-Fenster geben Sie dann nacheinander einzeln die Befehle aus dem Kasten Installationsbefehle ein. Bei einigen Befehlen müssen Sie das Passwort erneut eingeben oder die Befehlsausführung mit Y bestätigen. Einzelne Befehle brauchen außerdem einige Minuten zur Ausführung. Nach dem `menuconfig`-Befehl erscheint das Fenster mit den Einstellungen, wie Sie sie beim Kompilieren für Ihren Drucker (siehe Make 3/23) gewählt haben (Bild 4).

Da nun aber ein Klipper für den Raspberry Pi Pico kompiliert werden soll, müssen einige Zeilen geändert werden. Dazu setzen Sie mit den Cursortasten den blauen Markierungsbalken jeweils auf die zu ändernde Zeile, drücken die Eingabetaste und wählen die entsprechende Einstellung (Bild 5).

Mit Q beenden Sie das Konfigurationsmenü und bestätigen mit Y das Speichern der geänderten Werte. Jetzt ist es an der Zeit, den Raspberry Pi Pico mit dem Raspberry des Druckers über ein USB-Kabel zu verbinden. Halten Sie dabei den Taster auf dem Pico-Board (BOOTSEL) gedrückt. Erst dadurch ist der Pico imstande, über USB eine neue Firmware entgegenzunehmen. Mit dem Befehl

```
sudo make flash FLASH_DEVICE=2e8a:0003
```

starten Sie den Compiler. Anschließend sollte Ihnen eine Erfolgsmeldung angezeigt werden (Bild 6).

Nun müssen Sie noch herausfinden, auf welchen Namen der Pico hört, also unter welcher Kennung er sich am USB-Anschluss meldet, denn diese Angabe braucht der Raspberry Pi des Druckers, um auf den Sensor zugreifen zu können. Geben Sie dazu den folgenden Befehl ein:

```
ls /dev/serial/by-id/*
```

Als Ergebnis erhalten Sie etwas wie in Bild 7. Notieren Sie sich die Zeile, die mit `usb-Klipper_rp2040` beginnt.

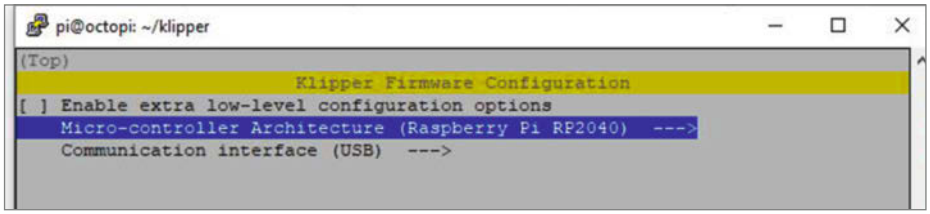


Bild 5: Der Pico braucht hingegen diese Werte.

Sensor-Befestigung

Der Sensorhalter hat die Form einer Schraubklemme. Seine Spannweite reicht aus, um ihn am Hotend und am Druckbett der meisten Drucker zu befestigen. Falls er an Ihrem Printer nicht passt, finden Sie auf Thingiverse eine Menge weiterer Halter. Suchen Sie einfach nach der Druckerbezeichnung und „Acceleration".

Wichtig: Wenn Sie die Resonanz des Druckbetts messen, muss der Halter unmittelbar an der Druckplatte befestigt werden. Die meisten Platten sind über elastische Elemente (Gummi, Silikon oder Stahlfedern) mit dem Wagenschlitten verbunden. Befestigen Sie den Halter am Schlitten statt an der Platte, messen Sie dessen Resonanz und lassen den Teil bestehend aus Federn und Druckplatte außen vor. Das Ergebnis kann dann natürlich nicht stimmen. Dasselbe gilt für die inzwischen öfter benutzten Hotends mit Wägezellen als Level-Sensor. Auch diese Zelle hat eine gewisse Elastizität. Daher muss der Sensor in solchen Fällen direkt am Hotend befestigt werden. Bei kaltem Hotend passt der Halter meist auf den Heizblock.

Drucker-Konfiguration anpassen

Dem Raspberry Pi des Druckers müssen wir nun noch die Existenz des beziehungsweise der neuen Sensoren und seines Pico-Helfers mitteilen. Das geschieht in der Klipper-Konfigurationsdatei des Druckers. Da der hier verwendete Drucker über Octoprint gesteuert wird, können Sie die Datei per Klick auf „Öffne Editor" auf der Oktoprint-Weboberfläche öffnen. In den Kästen zum Konfigurationseintrag für einen beziehungsweise zwei Sensoren finden Sie die Zeilen, die Sie hinzufügen müssen.

Hier ein paar Erläuterungen: Im Abschnitt `[mcu pico]` wird dem Raspi mitgeteilt, dass es einen neuen Prozessor mit dem Namen `pico` gibt. Direkt darunter finden Sie die zuvor

```
pi@octopi:~/klipper $ sudo make flash FLASH_DEVICE=first
  Creating symbolic link out/board
  Flashing out/klipper.uf2 to first
lib/rp2040_flash/rp2040_flash out/klipper.uf2

Loaded UF2 image with 86 pages
Found rp2040 device on USB bus 1 address 18
Flashing...
Resetting interface
Locking
Exiting XIP mode
Erasing
Flashing
Rebooting device
pi@octopi:~/klipper $
```

Bild 6: Falls diese Meldung ausbleibt, haben Sie vermutlich vergessen, den Taster auf dem Pico beim Einstecken des USB-Kabels zu drücken.

```
pi@octopi:~/klipper $ ls /dev/serial/by-id/*
/dev/serial/by-id/usb-Klipper_rp2040_E6616407E33C5F27-if00
/dev/serial/by-id/usb-Klipper_stm32f103xe_35FFD7054E4B323517881143-if00
```

Bild 7: Die erste Kennung gilt für den Raspberry Pi Pico, die zweite ist der ebenfalls per USB angeschlossene Drucker.

© Copyright by Maker Media GmbH.

Konfigurationseintrag für einen Sensor

```
[mcu pico]
serial: /dev/serial/by-id/usb-Klipper_rp2040_E6616407E33C5F27-if00

[mpu9250]
i2c_mcu: pico
i2c_bus: i2c0a
i2c_address: 104

[resonance_tester]
accel_chip: mpu9250
probe_points: 117, 117, 20

[static_digital_output pico_3V3pwm] # Improve power stability
pins: pico:gpio23
```

Konfigurationseintrag für zwei Sensoren

```
[mcu pico]
serial: /dev/serial/by-id/usb-Klipper_rp2040_E6616407E33C5F27-if00

[mpu9250 hotend]
i2c_mcu: pico
i2c_bus: i2c0a
i2c_address: 104

[mpu9250 bed]
i2c_mcu: pico
i2c_bus: i2c0a
i2c_address: 105

[resonance_tester]
accel_chip_x: mpu9250 hotend
accel_chip_y: mpu9250 bed
probe_points:
    117, 117, 20  # an example

[static_digital_output pico_3V3pwm]
pins: pico:gpio23
```

notierte Anschlusskennung. Die kann bei Ihrem Gerät anders aussehen. Setzen Sie das Entsprechende ein.

Unter [mpu9250] werden die Daten des Beschleunigungssensors bekannt gegeben: Er hängt am Prozessor namens `pico`, darauf folgen die Adressen des I²C-Busses und des Sensors. In der Zwei-Sensor-Lösung gibt es natürlich zwei Sensoren, die jeweils den Namenszusatz `hotend` bzw. `bed` erhalten.

Anschließend folgen die Koordinaten des Messpunkts `probe_points`. Auf diesen Punkt wird die Düsenspitze vor der Messung gefahren. Er sollte in der Mitte über dem Druckbett liegen. Die Düsenspitze darf die Druckplatte nicht berühren.

In der Klipper-Dokumentation wird noch der Absatz `[static_digital_output pico_3V3pwm]` empfohlen. Das soll die Spannungsversorgung des Sensors stabil halten. Bei meinen Tests ging es auch ohne, aber offenbar nicht immer. Deshalb sollten Sie auch diesen Eintrag verwenden.

Nachdem Sie alle Zeilen eingetragen haben, speichern Sie die Datei und starten Sie den Drucker erneut.

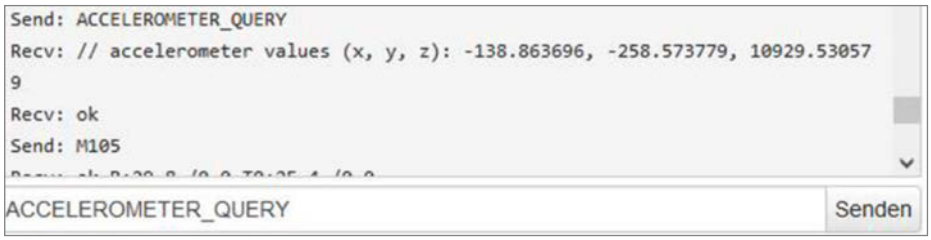


Bild 8: So sieht ein erfolgreicher Sensortest aus.

Resonanzfrequenzen messen

Bevor Sie mit dem Test beginnen, müssen Sie ein Homing durchführen. In Octoprint geben Sie dazu unter dem Reiter Terminal G28 ein. Danach testen wir, ob Klipper den oder die Sensor(en) auch erkennt. In Octoprint geben Sie dazu diesen Befehl ein:

```
ACCELEROMETER_QUERY
```

Falls Sie die Zwei-Sensor-Lösung verwenden, sind dementsprechend auch zwei Tests erforderlich. Die Befehle dazu lauten:

```
ACCELEROMETER_QUERY CHIP=bed
ACCELEROMETER_QUERY CHIP=hotend
```

Achtung: Hier und bei nahezu allen im Terminal eingegebenen Befehlen ist es wichtig, alles groß zu schreiben! Klipper sollte daraufhin eine Erfolgsmeldung ausgeben wie in Bild 8.

Ist das nicht der Fall, steckt der Fehler vermutlich in der Verdrahtung zwischen Sensor und Pico oder in einem Tippfehler bei der Änderung der Konfigurationsdatei. Korrigieren Sie das und testen Sie erneut.

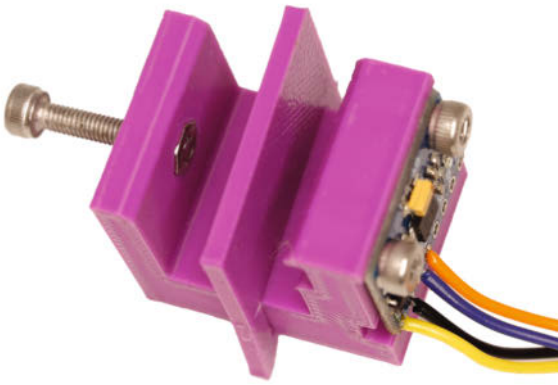
Jetzt kann es an die erste Messung gehen. Falls noch nicht geschehen, schrauben Sie den Sensor mit zwei M3×6-Schrauben auf den Halter. Die untere Zange des Halters hat innen eine sechseckige Öffnung, in die eine M3-Mutter passt. Von Außen stecken Sie eine M3×16-Schraube hinein und drehen Sie in die Mutter. Klipsen Sie noch das Schutzstück in den Halter und er ist einsatzbereit (Bild 9).

Zunächst zur Messung am Druckbett. Befestigen Sie den Sensor auf der Druckplatte (Bild 10). Knallen Sie die Schraube nicht zu fest, allerdings darf der Sensor auch nicht wackeln. Das Druckbett bewegt sich nur nach vorn bzw. hinten. Das ist die Y-Achse. Der Befehl zur Resonanzmessung lautet daher:

```
TEST_RESONANCES AXIS=Y
```

Anschließend beginnt der Drucker, den Drucktisch schnell vor- und zurückzubewegen. Beginnend bei 5 Hz arbeitet er sich bis zu 132 Hz

Bild 9: Der komplett zusammengebaute Halter mit Sensor und Halteschraube



© Copyright by Maker Media GmbH.

hoch (Bild 11). Wundern Sie sich nicht, wenn es dabei auch einmal heftig rappelt, insbesondere, wenn der Drucker in einem Gehäuse oder Schrank steht. Immer wenn es besonders laut ist, wurde eine Resonanzfrequenz erreicht. Es kann nämlich durchaus mehrere geben. Übrigens: Die oft als Geräuschdämpfer angebrachten federnden Füße für den Drucker sind kontraproduktiv: Diese schwingfähigen Gebilde verstärken die Schwingungen des Druckers, da sie kaum Bewegungsenergie an die Standfläche ableiten. Wenn Sie also später wirklich schnell und gut mit dem Printer drucken möchten, sollte Sie solche Füße entfernen. Ach ja, auch zu fest gespannte Zahnriemen wirklich ähnlich. Ihre Vorspannung sollte nur so weit gehen, dass die angetriebenen Teile spielfrei arbeiten. Fester angezogen wirken sie wie Gitarrensaiten, die ja prächtig und lange nachschwingen können.

Nach Abschluss des Tests werden die Daten in die Datei /tmp/resonances_y_yyyymmdd_hhmmss.csv geschrieben. In diesem Dateinamen steht yyyy für die Jahreszahl, mm für den Monat, dd für den Tag und entsprechend hhmmss für die Uhrzeit des Tests (Bild 12).

Nun wechselt der Sensor seine Position zum Hotend. Die Spannweite des Halters genügt, um ihn auch dort zu befestigen. Das Hotend bewegt sich in zwei Achsen: In der Höhe, das ist die Z-Achse, und seitlich, das ist die X-Achse. Für die Z-Achse muss keine Resonanzfrequenz ermittelt werden, denn in deren Richtung bewegt sich das Hotend während des Drucks nur sehr wenig. Führen Sie erneut ein Homing durch. Der Testbefehl checkt nur die X-Achse. Er lautet daher:

```
TEST_RESONANCES AXIS=X
```

Das Testresultat landet am Schluss in der Datei /tmp/resonances_x_yyyymmdd_hhmmss.csv.

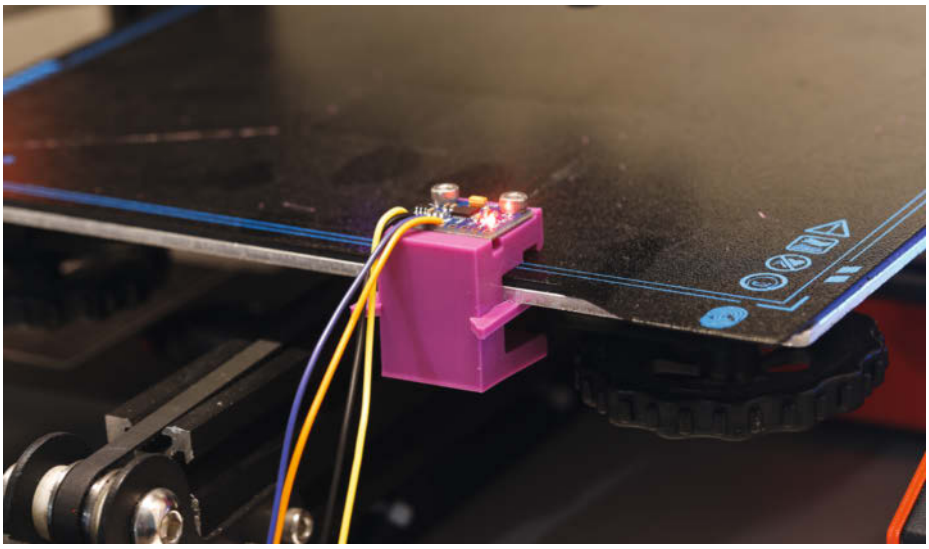


Bild 10: Der Sensor muss fest, aber nicht unbedingt in der Mitte sitzen.

Wärmepumpentechnik für Einsteiger

c't WEBINAR

Technik verstehen, Machbarkeit prüfen, Angebote beurteilen

Das **Webinar** wendet sich an Einsteiger, die das **Thema Wärmepumpe** zwar auf dem Schirm haben, aber noch nicht über weitergehende Erfahrungen und Kenntnisse verfügen. In zwei Stunden vermittelt c't-Redakteur Georg Schnurer fundiertes Wissen: **von Technik bis Effizienz**. Auch **Kosten** und **Praxistipps** inklusive. Das Webinar hilft Ihnen, sich herstellerunabhängig einen ersten Überblick zu verschaffen.

WEBINAR AM 18.10.2023

Jetzt Tickets sichern:
webinare.heise.de/waermepumpen

© Copyright by Maker Media GmbH.

```
Recv: ok B:29.2 /0.0 T0:26.7 /0.0
Send: M105
Recv: ok B:29.3 /0.0 T0:26.7 /0.0
Send: M105
Recv: ok B:29.2 /0.0 T0:26.7 /0.0
Send: TEST_RESONANCES AXIS=Y
Recv: // Disabled [input_shaper] for resonance testing
Recv: // Testing frequency 5 Hz
Recv: // Testing frequency 6 Hz
Recv: // Testing frequency 7 Hz
Recv: // Testing frequency 8 Hz
Recv: // Testing frequency 9 Hz
Recv: // Testing frequency 10 Hz
Recv: // Testing frequency 11 Hz
Recv: // Testing frequency 12 Hz
Recv: // Testing frequency 13 Hz
Recv: // Testing frequency 14 Hz
```

Bild 11: Der Test beginnt von 5 Hz an aufwärts ...

```
Recv: // Testing frequency 120 Hz
Recv: // Testing frequency 121 Hz
Recv: // Testing frequency 122 Hz
Recv: // Testing frequency 123 Hz
Recv: // Testing frequency 124 Hz
Recv: // Testing frequency 125 Hz
Recv: // Testing frequency 126 Hz
Recv: // Testing frequency 127 Hz
Recv: // Testing frequency 128 Hz
Recv: // Testing frequency 129 Hz
Recv: // Testing frequency 130 Hz
Recv: // Testing frequency 131 Hz
Recv: // Testing frequency 132 Hz
Recv: // Testing frequency 133 Hz
Recv: // Re-enabled [input_shaper]
Recv: // Resonances data written to /tmp/resonances_y_20230823_130000.csv file
Recv: ok
Send: M105
```

Bild 12: ... und endet mit 132 Hz. Der Test ist durch, das Ergebnis in eine Datei geschrieben.

```
pi@octopi:~/klipper $ ~/klipper/scripts/calibrate_shaper.py /tmp/resonances_x_*.
csv -o /tmp/shaper_calibrate_x.png
Fitted shaper 'zv' frequency = 109.8 Hz (vibrations = 23.5%, smoothing ~= 0.018)
To avoid too much smoothing with 'zv', suggested max_accel <= 47000 mm/sec^2
Fitted shaper 'mzv' frequency = 69.0 Hz (vibrations = 6.0%, smoothing ~= 0.044)
To avoid too much smoothing with 'mzv', suggested max_accel <= 14000 mm/sec^2
Fitted shaper 'ei' frequency = 96.0 Hz (vibrations = 6.5%, smoothing ~= 0.037)
To avoid too much smoothing with 'ei', suggested max_accel <= 17200 mm/sec^2
Fitted shaper '2hump_ei' frequency = 85.6 Hz (vibrations = 2.2%, smoothing ~= 0.
074)
To avoid too much smoothing with '2hump_ei', suggested max_accel <= 8100 mm/sec^
2
Fitted shaper '3hump_ei' frequency = 79.2 Hz (vibrations = 0.5%, smoothing ~= 0.
131)
To avoid too much smoothing with '3hump_ei', suggested max_accel <= 4600 mm/sec^
2
Recommended shaper is ei @ 96.0 Hz
pi@octopi:~/klipper $
```

Bild 13: Die Empfehlung für die Resonanzfrequenz, den Shaper sowie die maximale Beschleunigung sollten Sie notieren. In diesem Fall sind es 96Hz, „ei“ als Shaper und 17200mm/s² als maximale Beschleunigung.

Datenauswertung

Falls Sie nun befürchten, dass Sie den Inhalt dieser Dateien selbst auswerten müssten: Nein, dazu haben Sie ja zwei kleine Diener namens Raspberry und Klipper. Genauer gesagt befindet sich im Klipper-Verzeichnis auf dem Raspberry ein Skript, das Ihnen die Datenauswertung abnimmt, die Resonanzfrequenz in einer Grafik darstellt und Empfehlungen für die Einstellungen des input shapers gibt. Die können Sie dann einfach in die Konfigurationsdatei eintragen und fertig.

So geht's: Geben Sie im Putty-Terminalfenster nacheinander die beiden Befehle aus dem Kasten zu den Skript-Befehlen für die Datenauswertung ein. Jeder Durchgang dauert etwa eine Minute. Schließlich erhalten Sie eine Schlussmeldung (Bild 13) mit den Empfehlungen für die Resonanzfrequenzeinstellung, den Beschleunigungswert und den zu wählenden Shaper für diese Achse.

Diese Shaper sind unterschiedliche Algorithmen, um die Resonanzen vorauszuberechnen und zu vermeiden. Außerdem wird die Bilddatei /tmp/shaper_calibrate_x.png erzeugt, die Sie mithilfe eines FTP-Programms wie Filezilla auf Ihren Computer übertragen und anzeigen lassen können (Bild 14). Für die Y-Achse heißt das Bild entsprechend /tmp/shaper_calibrate_y.png.

Eintrag in die Konfigurationsdatei

Öffnen Sie erneut die Konfigurationsdatei und suchen Sie darin nach dem Abschnitt `[printer]`. Für die X- und die Y-Achse gibt es nur einen gemeinsamen Wert für die maximale Beschleunigung `max_accel:`. Tragen Sie dahinter den kleineren der beiden zuvor ermittelten Beschleunigungswerte ein. In diesem Beispiel ist es 4000.

Als nächstes kontrollieren Sie, ob es in der Datei bereits einen Abschnitt namens `[input shaper]` gibt. Falls nicht, tragen Sie ihn ein. Darunter folgen diese Zeilen:

```
shaper_freq_x: 96.0
shaper_type_x: ei
shaper_freq_y: 46.4
shaper_type_y: ei
```

Speichern Sie die Datei und führen Sie einen Neustart des Druckers durch.

Drucken ohne Pico

Den Pico mitsamt dem Sensor brauchen Sie zum Drucken nicht und können ihn an anderen Druckern einsetzen. Allerdings würde Ihr Drucker beim Start den fehlenden Pico bemerken und den Dienst verweigern.

Das umgehen Sie, indem Sie vor die in den beiden Kästen für die Konfigurationseinträge genannten Zeilen jeweils eine Raute # setzen.

© Copyright by Maker Media GmbH.

Skript-Befehle für die Datenauswertung

```
# für die X-Achse
~/klipper/scripts/calibrate_shaper.py /tmp/resonances_x_*.csv -o /tmp/shaper_calibrate_x.png
#für die Y-Achse
~/klipper/scripts/calibrate_shaper.py /tmp/resonances_y_*.csv -o /tmp/shaper_calibrate_y.png
```

Die Zeilen für den input shaper lassen sie hingegen, wie sie sind.

Jetzt können Sie es wagen und mal mit deutlich höherem Tempo drucken. Sie werden sich wundern, was sich noch alles aus Ihrem Printer herauskitzeln lässt. Dafür werden am Anfang aber noch einige Experimente notwendig sein. Erhöhen Sie im Slicer zunächst die Geschwindigkeit bei Bewegungen und Füllungen, dann die für Außen- und Innenwände sowie andere Oberflächen. Viel Spaß dabei! *—hgb*

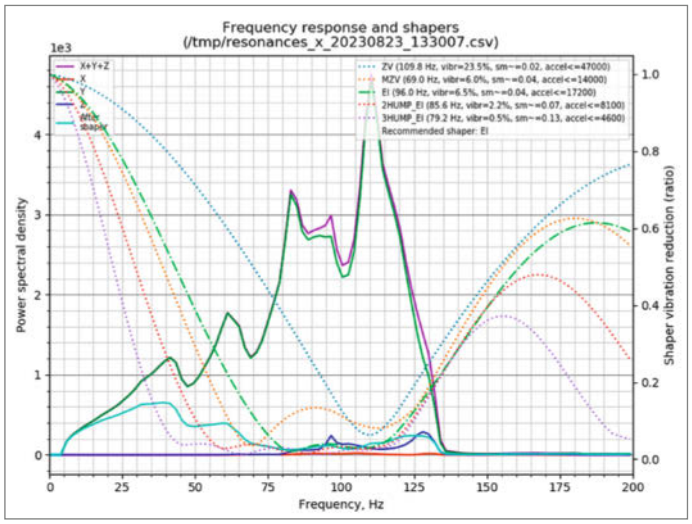


Bild 14: Das sieht zunächst verwirrend aus. In der Grafik des Skripts ist das wichtigste die hellblaue Linie: Sie zeigt die Schwingungen nach Anwendung des input shapers.

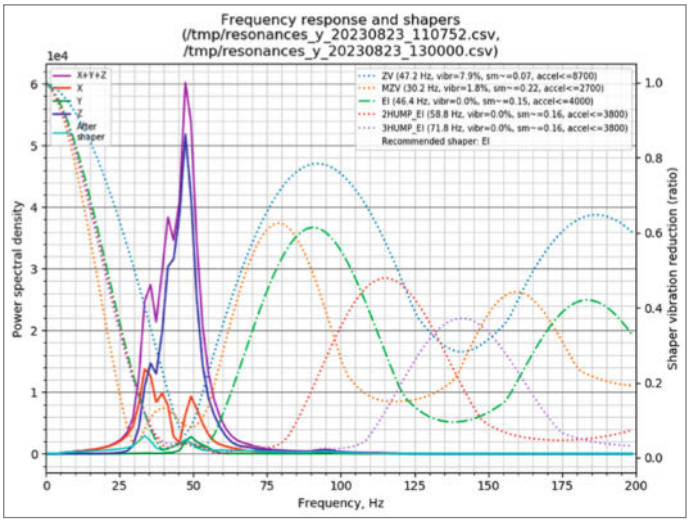


Bild 16: Für die Y-Achse sieht das Ergebnis ganz anders aus. Hier wird als Resonanzfrequenz 46,4Hz, als Shaper „ei“ und für die Beschleunigung maximal 4000mm/s² empfohlen.

Generell portofreie Lieferung für Heise Medien- oder Maker Media Zeitschriften-Abonnenten oder ab einem Einkaufswert von 30 € (innerhalb Deutschlands). Nur solange der Vorrat reicht. Preisänderungen vorbehalten.

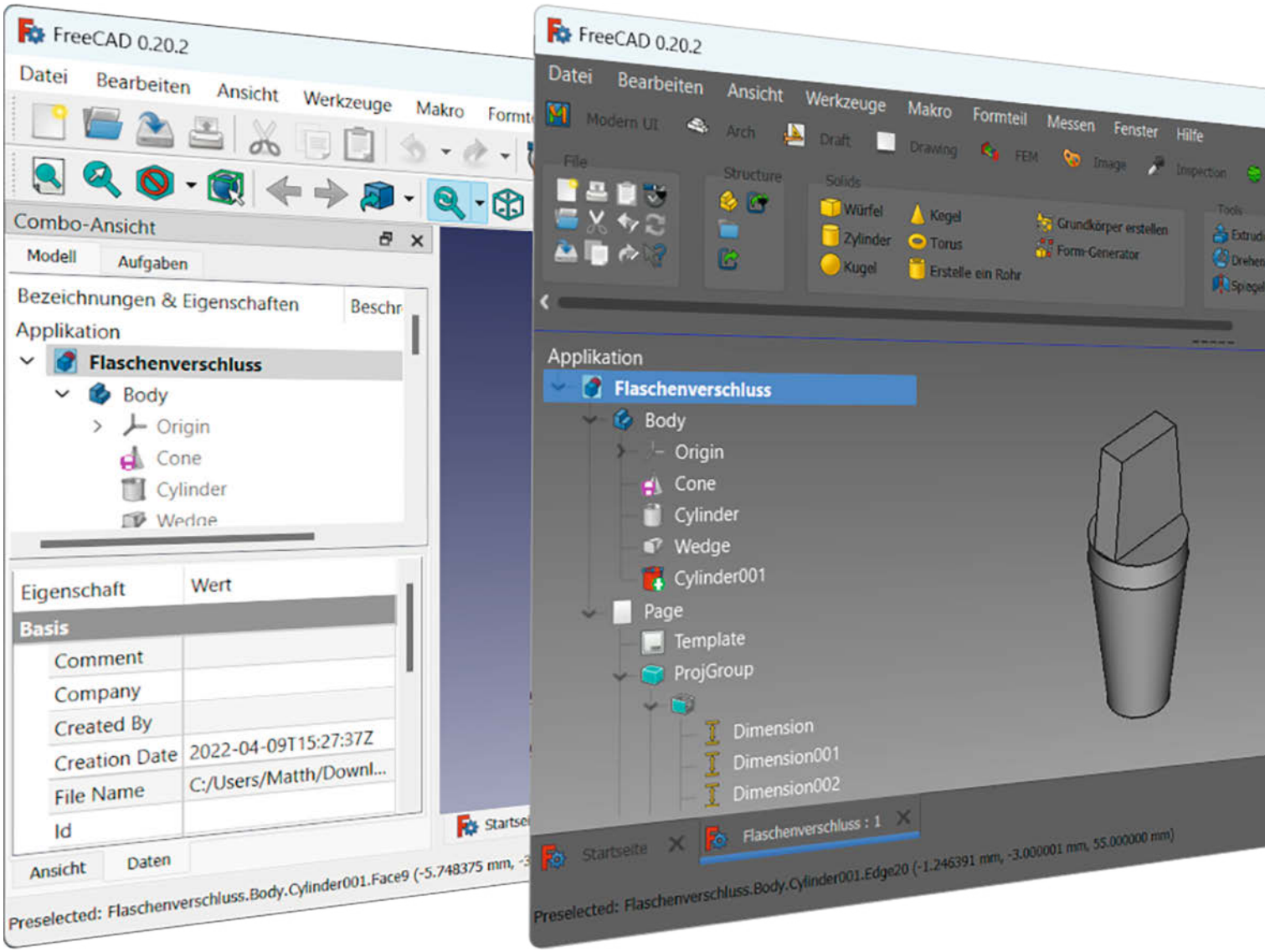
© Copyright by Maker Media GmbH.

heise Shop

PowerToys für FreeCAD

Ist das wirklich FreeCAD? Ausgesuchte Add-ons für die kostenlose Konstruktionssoftware beschleunigen und erleichtern nicht nur die Arbeit mit dem Programm, sondern krempeln sogar die Bedienoberfläche komplett um. Wir zeigen eine Auswahl interessanter Add-ons und wie man diese installiert.

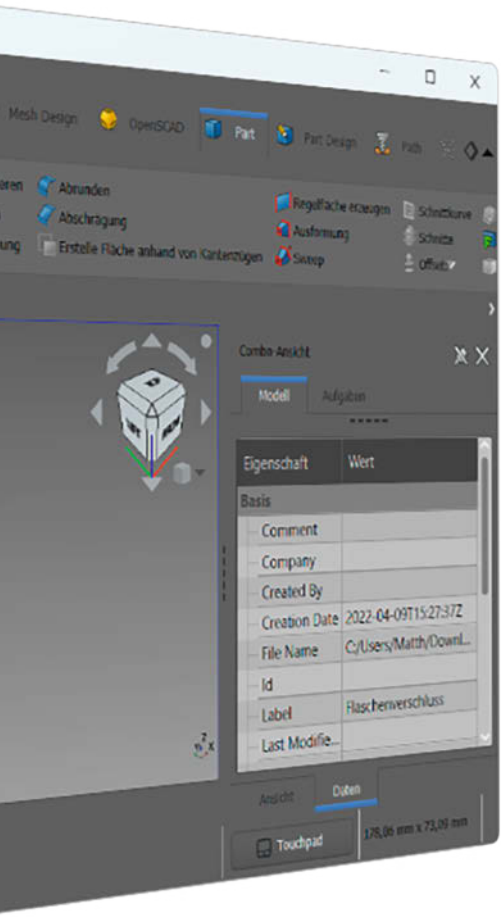
von Matthias Mett



© Copyright by Maker Media GmbH.

Was unter Windows die Hilfsprogramme Microsoft PowerToys sind, gibt es in ähnlicher Form auch für FreeCAD – in Form zusätzlich installierbarer Add-ons. Hier stehen eine Reihe von kostenlosen Zusatzprogrammen zur Wahl, die sich mit dem eingebauten Erweiterungsmanager (auch Add-on-Manager genannt) einfach installieren lassen. Wir haben verschiedene Add-ons herausgesucht, die Ihre Konstruktionsroutinen vereinfachen und auch die Standard-Oberfläche von FreeCAD stark verändern.

Alle in diesem Artikel aufgeführten Add-ons installieren Sie einfach über den Add-on-Manager, indem Sie ihn im Menü unter Werkzeuge aufrufen. Das jeweilige Add-on lässt sich im Filter – also im oberen rechten Eingabefenster – durch Eingabe des Namens suchen. Mit einem Doppelklick auf das entsprechende Add-on öffnet sich die Add-on-Beschreibung und man kann das Tool mit dem Installieren-Button oben rechts zu FreeCAD hinzufügen. Der Add-on-Manager kümmert sich auch um Updates, indem er später bei Aufruf nach Updates sucht und eventuelle Aktualisierungen links unten im Add-on-Manager vorschlägt.



Kurzinfo

» Die kostenlose Konstruktionssoftware FreeCAD mit Add-ons anpassen und erweitern
» Alternative Bedienoberfläche frischt das Design auf und erleichtert den Umstieg
» Launcher und Shortcuts für schnelleres Arbeiten

Mehr zum Thema

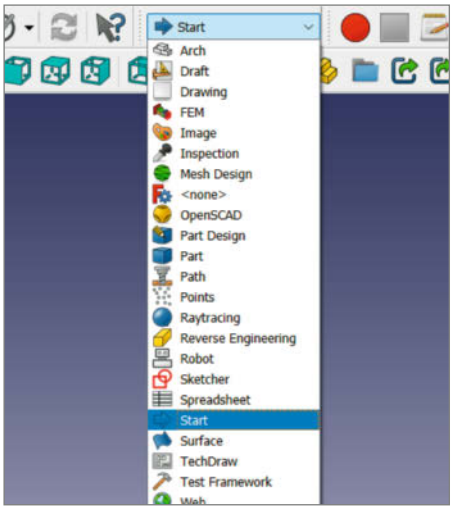
» Eine verlinkte Liste aller bisherigen FreeCAD-Artikel aus der Make gibt es online unter der Kurz-URL.

Alles zum Artikel im Web unter *make-magazin.de/xr5q*

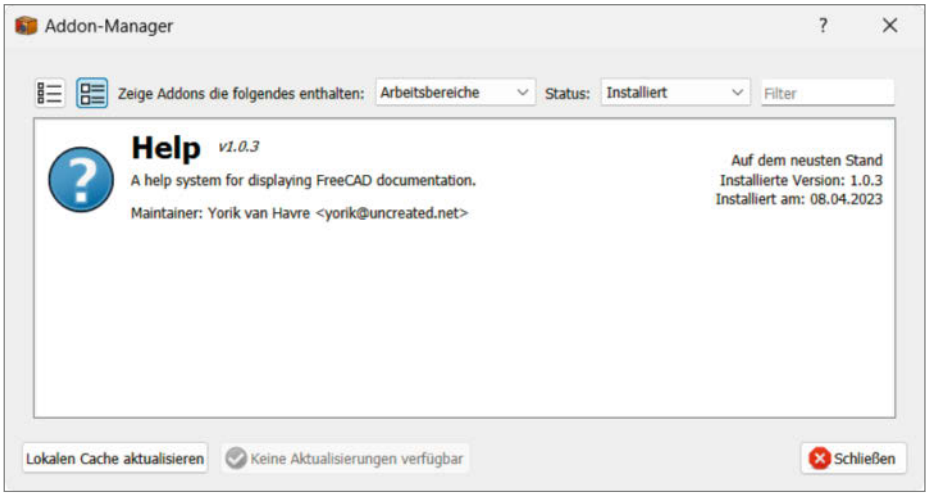
Hilfe

Wie jede gute Software hat FreeCAD auch einen Menüeintrag namens Hilfe – doch falls Sie FreeCAD frisch installiert haben und diese Hilfe aufrufen, erscheint nur eine Meldung, dass man die Erweiterung namens Hilfe über den Erweiterungsmanager installieren soll. Praktischerweise können Sie direkt im Meldungsfenster auf eine Schaltfläche klicken, um dorthin zu wechseln und die Hilfe zu installieren. Eventuell müssen Sie beim ersten Start des Erweiterungsmanagers noch eine Warnung abnicken, dass es sich bei Add-ons um Angebote von dritter Seite handelt und anhaken, dass die Erweiterungen automatisch auf Stand gehalten werden sollen. Dann geht es los. Nach der Installation der Hilfe (und anderer Add-ons) muss FreeCAD neu gestartet werden.

Der alternative Weg: Im Menü auf Werkzeuge/Addon-Manager klicken und dann

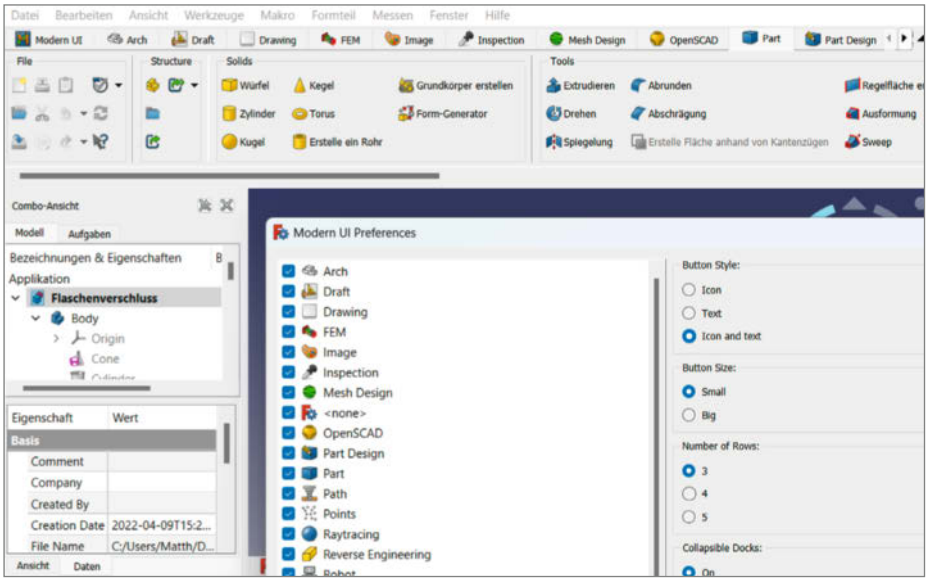


FreeCAD klassisch: Alle Workbenches aufgelistet im Auswahlmenü

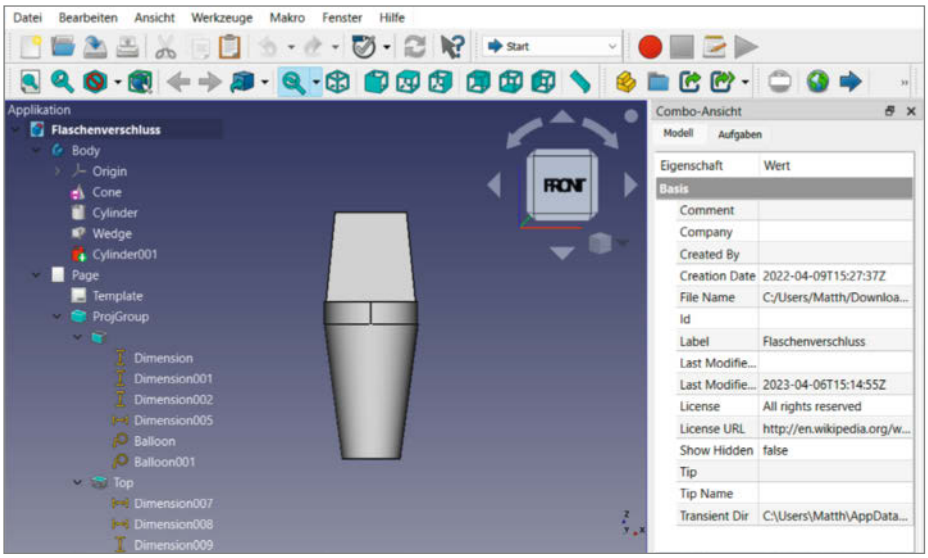


Add-on-Manager mit installiertem Help-Modul, gefiltert nach allen installierten Modulen und Arbeitsbereichen

© Copyright by Maker Media GmbH.



Die Ribbon-Struktur des Add-ons Modern-UI mit dem Einstellungsfenster Modern UI Preferences



Glass-Add-on mit vergrößertem Eigenschaften- und Aufgabenfenster auf der rechten Seite

oben rechts im Fenster „Help“ eingeben. Wählen Sie in den Suchergebnissen das gleichnamige Modul aus und installieren es mit dem Installieren-Button. Wenn Sie nach der Installation die Hilfe aufrufen, erscheinen die entsprechenden Hilfeseiten im Arbeitsbereich in einem neuen Tab.

Workbench-Ribbon

Wer schon einmal mit anderen CAD-Programmen gearbeitet hat, ist meist eine andere Arbeitsweise gewohnt. Das liegt vor allem daran, dass FreeCAD aus Modulen besteht, die sich Workbenches oder auch Arbeitsbereiche nennen. Bereits in der Standardinstallation enthält FreeCAD eine Reihe Workbenches mit zum Teil grundlegend verschiedenen Funktionen. Beim Starten der Software erscheint automatisch die Start-Workbench mit den ihr zugeordneten Symbolen in der Menüleiste. Über der Ausklappliste kann man in eine andere Workbench und damit einen anderen Werkzeugsymbolsatz umschalten, beispielsweise Part Design für die 3D-Modellierung oder Sketcher für 2D-Skizzen.

Das Add-on Modern UI hingegen stellt die einzelnen Arbeitsbereiche als Karteireiter nebeneinander dar und listet alle Buttons in Bereiche gruppiert auf. Diese Darstellung kennt man zum Beispiel aus dem Ribbon-Menüband von Microsoft Office oder anderen CAD-Programmen wie DesignSpark Mechanical oder Solid Edge.

Je nach Bildschirmauflösung passen in vielen Arbeitsbereichen die Buttons jedoch nicht direkt auf den Bildschirm. Hier erscheint dann eine horizontale Scrollleiste, mit der man das Menüband entsprechend verschieben kann. Weitere Möglichkeiten zum Einstellen des Menübands befinden sich auf der linken Seite im Karteireiter Modern UI unter Modern Settings. Dort lassen sich unter anderem der Button-Stil und die Button-Größe einstellen, außerdem einzelne Arbeitsbereiche ein- oder ausblenden, was das Ribbon deutlich übersichtlicher macht.

Das Add-on ist auf der Add-on-Seite noch mit dem frühen Alpha-Status angegeben. In dem dort angegebenen Link kann man auf die Diskussion im FreeCAD-Forum wechseln, um weitere Informationen über das Projekt zu erhalten. Generell fiel uns nach der Installation des Add-ons eine längere Startzeit von FreeCAD auf. Falls man das Add-on wieder deinstallieren möchte, kann es sein, dass nach der Deinstallation beim Start von FreeCAD keine Symbolleisten mehr erscheinen. Diese lassen sich mit dem auf der Add-on-Seite unter Deinstallation gezeigten Makro wieder sichtbar machen.

Objektbaum-Overlay

Ein weiteres Add-on mit dem Namen Glass stellt den Objektbaum transparent im Hauptansichtenbereich dar, ähnlich wie bei anderen CAD-Programmen wie Fusion 360 oder Solid Edge. Die hauseigene Combo-Ansicht links im Programmfenster mit dem Objektbaum und dem Aufgabenbereich sowie dem Eigenschaften-Editor bleiben dabei unverändert. Um den transparenten Objektbaum im Hauptansichtenbereich ein- oder auszuschalten, drückt man die Tasten Q+1. Damit der Objektbaum nicht doppelt erscheint, kann man die komplette Combo-Ansicht auf die rechte Seite des Bildschirms verschieben und das Eigenschaftenfenster ganz nach oben ziehen. Damit blendet man den Objektbaum aus und kann zwischen dem Eigenschaften- und dem Aufgabenfenster hin und her schalten.

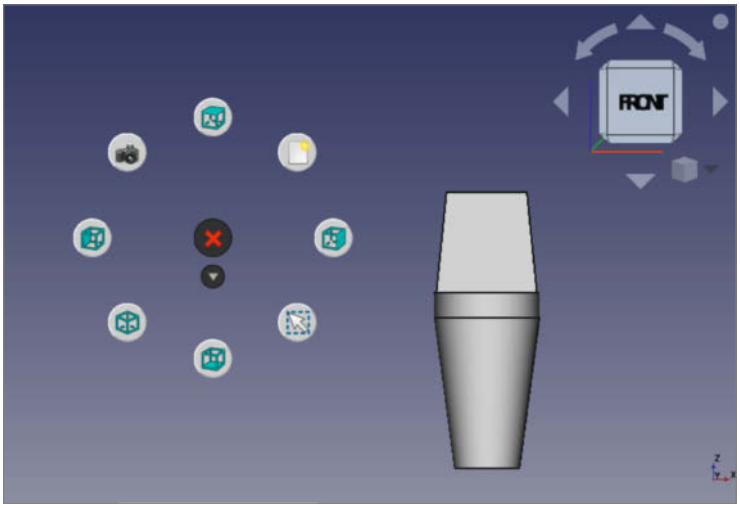
Durch diese Menüaufteilung mit der ausgeblendeten Baumansicht und dem großen Eigenschaftenfenster hat man mehr Übersicht, falls man eine Aufgabe bearbeitet und gleichzeitig der Objektbaum auf der linken Seite vorhanden ist. So entfällt das Umschalten in den Tabs zwischen dem Objektbaum und dem Aufgabenfenster und man vergisst nicht, eine noch nicht abgeschlossene Aufgabe zu beenden (was normalerweise bei FreeCAD eine häufige Fehlerquelle ist). Das Eigenschaftenfenster bleibt jedoch bei einem Neustart von FreeCAD nicht maximiert, die Baumansicht ist dann wieder sichtbar.

© Copyright by Maker Media GmbH.

Pie-Menüs

Ein weiteres Add-on zur Produktivitätssteigerung ist PieMenu, welches eine Reihe von Funktionen für den schnellen Zugriff in Kreisform darstellt. Die einzelnen Buttons lassen sich anpassen, sodass sich das Tool als grafische Shortcut-Sammlung nützlich macht.

Nach der Installation lässt sich PieMenu mit der Tabulator-Taste aufrufen und erscheint an der Stelle, wo sich der Mauszeiger im Zeichenbereich befindet. Zum Anpassen des Tools ist rechts in der Menüleiste der Punkt Accessoires mit dem Untermenü PieMenu hinzugekommen. In den Einstellungen lassen sich die einzelnen Buttons mit anderen Funktionen belegen und auch die Erscheinung des Tools ändern, wie beispielsweise die Größe der Buttons. So kann man sich oft verwendete Funktionen zum schnellen Zugriff auf das PieMenu legen.



Grafische Shortcut-Sammlung: PieMenu nach Aufruf mittels Tabulator-Taste

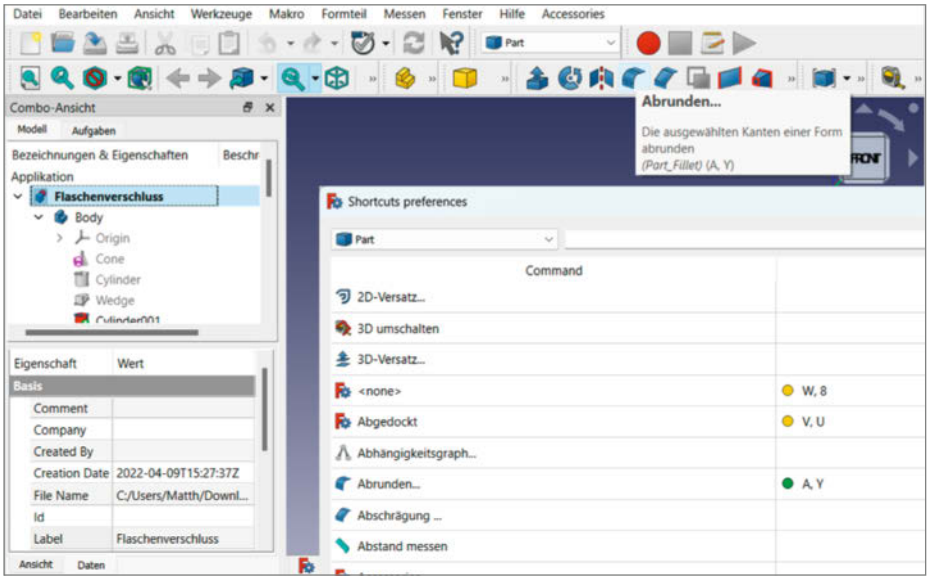
Tastenkürzel

Richtige Shortcuts, also Tastenkürzel für Funktionen, erstellt man mit dem Add-on ShortCuts. Nach der Installation befindet sich in der Menüleiste unter Accessoires der Menüpunkt ShortCuts, wo sich die Tastenkürzel verwalten lassen. In den Einstellungen sind die Funktionen und bereits vorbelegten Kürzel aufgelistet; den Werkzeugen ohne Eintrag in der Spalte Shortcut kann man nach Belieben eigene Tastenkombinationen zuweisen. Achtung: Auch hier sortiert FreeCAD die Funktionen nach Arbeitsbereichen!

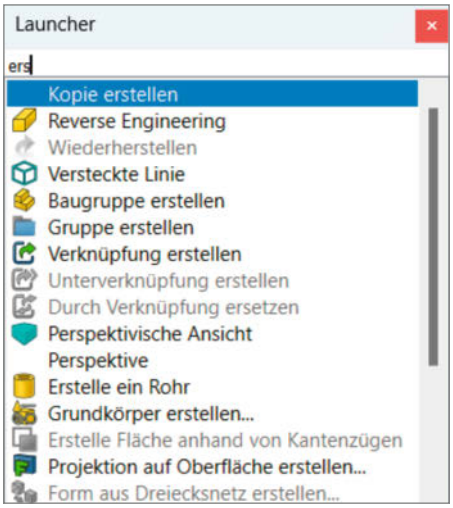
Als Beispiel soll der Befehl Abrunden aus dem Part-Arbeitsbereich die Tastenkombination a und y bekommen. Ob Sie die Buchstaben klein oder groß eintippen, spielt keine Rolle, in der Liste erscheinen sie stets groß. Ein Komma dient zur Trennung zwischen mehreren Tasten.

Wenn man nun im Part-Arbeitsbereich beispielsweise einen Würfel erstellt und eine Kante markiert, lässt sich mit Drücken der Tasten a und y nacheinander das Abrunden-

© Copyright by Maker Media GmbH.



Das mithilfe von ShortCuts eingerichtete Tastenkürzel des Abrunden-Buttons mit erweitertem Tooltipp und die dazugehörigen Einstellungen in den ShortCut-Preferences



Der Laucher liefert nach wenigen eingetippten Zeichen anklickbare Suchergebnisse.

Menü im Aufgabenbereich aufrufen. Die bisher zugewiesenen Tastenkombinationen lassen sich leicht feststellen, indem man den Mauszeiger über einem Werkzeug-Button schweben lässt und so den Tooltipp aufruft – allerdings erst nach einem Neustart von FreeCAD.

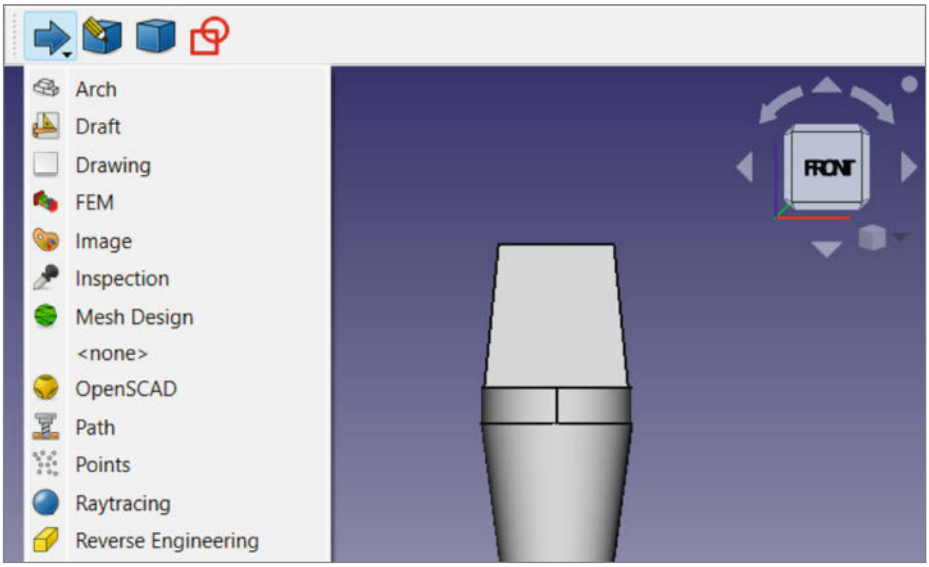
Launcher

Das Add-on Launcher stellt eine Eingabezeile bereit, die man in der Combo-Ansicht andocken, in die Symbolleiste verschieben oder als eigenständiges Fenster verwenden kann. Falls das Tool nach der Installation nicht sichtbar sein sollte, setzen Sie einen Haken dafür unter Ansicht/Paneele/Launcher.

Wenn man in die Eingabezeile einen Begriff eintippt, listet das Tool alle Funktionen auf, die es dazu findet. Mit einem Klick lässt sich die Funktion aus den Suchergebnissen heraus aufrufen. Auch Änderungen von Einstellungen wie beispielsweise das Umschalten des Navigationsstils lassen sich damit bewerkstelligen. Das Tool ist gut geeignet für diejenigen, welche die Funktionen suchen und durch Tippen von ein paar Anfangsbuchstaben schnell aufrufen wollen.

Arbeitsbereich-Buttons

Wenn Sie trotz schicker Ribbons- und Overlay-Add-ons lieber mit der Standard-Bedienoberfläche von FreeCAD arbeiten möchten, gibt es auch hierfür Erweiterungen, welche die Arbeit vereinfachen. Eine davon ist Selector Toolbar, wodurch eine neue Werkzeugleiste hinzugefügt wird, die für ausgewählte Arbeitsbereiche einen Button enthält, sodass man mit einem Klick dorthin umschalten kann. Die Toolbar enthält außerdem auf der linken Seite eine weitere Ausklappliste. Was wo erscheint, legt man im Menü „Accessoires/Selector Toolbar" fest: Ein Haken in der Checkbox steht für einen eigenen Button, ein Minus für einen Eintrag in der Ausklappliste, eine leere Checkbox blendet den Arbeitsbereich aus beidem aus.



Selector Toolbar mit den wichtigsten Workbenches Part, Part Design und Sketcher als Button sowie mit ausgeklappten weiteren Arbeitsbereichen

Mehr Stil

Bei dem Add-on ToolbarStyle handelt es sich um eine einfache Erweiterung mit vier Einstellungen für das Aussehen der Standard-Buttons: als Icon, als Text, als Icon mit Text oder mit Text unterhalb des Icons. Auch hierfür findet man die Einstellungen im Menü unter Accessoires bei ToolbarStyle.

Man kann FreeCAD mit zusätzlichen Themes erweitern, indem man im Add-on-Manager im oberen linken Drop-Down-Menü Einstellungs-Pakete auswählt. Nach der Installation eines solchen Themes, etwa ProDarkThemePreferencePack, findet man dieses im Menü unter Bearbeiten/Einstellungen/Allgemein/Präferenz-Pakete. Dort lässt sich das Theme mit dem Button Übernehmen aktivieren, wobei sich das Aussehen von FreeCAD direkt verändert (ohne Neustart!).

Screenshots

Zusätzlich zu Arbeitsbereichen und Themes können Sie in FreeCAD Makros installieren. Um diese anzuzeigen, wählen Sie im Add-on-Manager im oberen linken Drop-Down-Menü die Rubrik Makros aus, die daraufhin in der

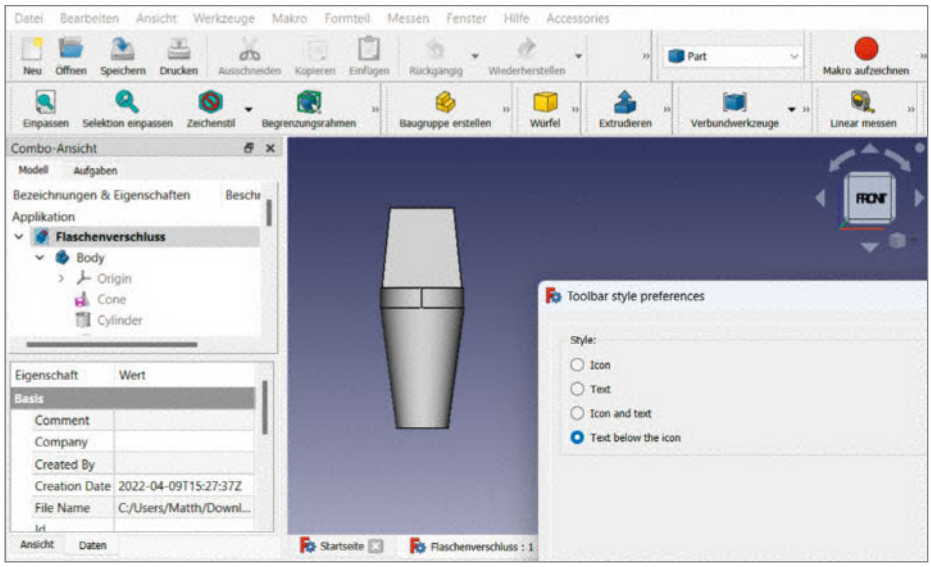
© Copyright by Maker Media GmbH.

Übersicht erscheinen. Mit dem Makro Snip lassen sich aus FreeCAD heraus Screenshots erstellen, ohne dass man ein zusätzliches Programm benötigt, was ursprünglich dafür gedacht war, Screenshots für das FreeCAD-Forum aufzunehmen.

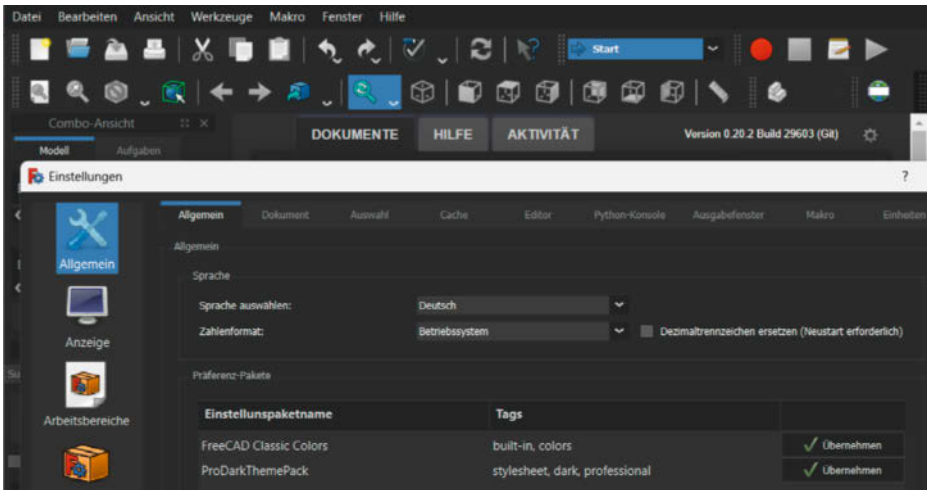
Bei der Installation fragt FreeCAD nach, ob es einen Button in der Buttonleiste hinzufügen soll. Beim Aufrufen des Tools erscheint ein Fenster, das man passend zieht, um den Bildschirmausschnitt zu wählen. OK schießt einen entsprechenden Screenshot.

Weiter stöbern

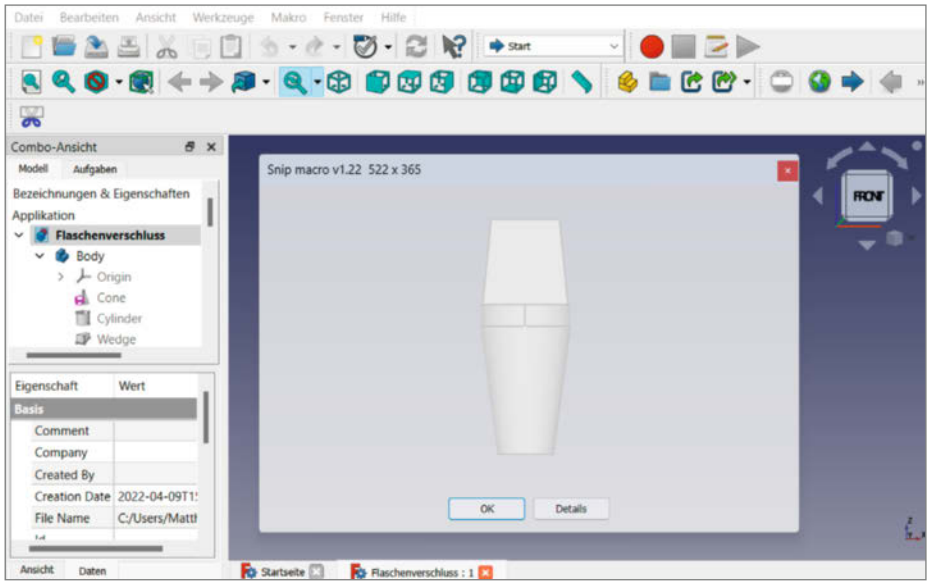
Die vorgestellten Tools lassen sich auch problemlos kombinieren. So kann man beispielsweise ModernUI und Glass parallel installieren. Die hier vorgestellten Zusatzprogramme stellen natürlich nur eine sehr kleine Auswahl der verfügbaren Add-ons in FreeCAD dar, es finden sich noch viele weitere Zusatzfunktionen im Add-on-Manager wieder – von Zahnrad-Generatoren über 3D-Druck- und Animationswerkzeuge bis hin zu spezialisierten Funktionen für den Gitarrenbau. Daher lohnt es sich, ausführlich im Add-on-Manager einmal zu stöbern – vielleicht ist ja genau das passende für Ihr Projekt mit dabei. *—pek*



ToolbarStyle-Einstellungen, hier gewählt: Text unterhalb der Buttons



Angewendetes ProDarkThemePreferencePack mit geöffneten Einstellungen



Das Screenshot-Makro Snip mit zusätzlicher Symbolleiste und Ausschnitt-Auswahlfenster

Alle FreeCAD-Artikel aus Make als PDF

Seit 2020 zeigen wir in loser Folge den praktischen Umgang mit der kostenlosen Software FreeCAD. Alle elf bisher in Make erschienenen Praxisartikel unseres Autors Matthias Mett bekommt man jetzt auch als elektronisches Dossier zusammengefasst. Auf 58 Seiten werden dort fünf kleinere und größere Projekte vom Stempel bis zum Raspberry-Pi-Gehäuse Schritt für Schritt und Klick für Klick beschrieben; zusätzlich gibt es insgesamt zwei Stunden Video-Tutorials direkt am digitalen Zeichenbrett.

Das Dossier ist ab sofort im heise shop für 7,90 Euro zu kaufen, Make-Abonnenten zahlen nur 5,90 Euro. Am einfachsten finden Sie das Dossier über diesen Link:

▶ make-magazin.de/xr5q

© Copyright by Maker Media GmbH.

Bücher zu FreeCAD

Die kostenlose 3D-Konstruktionssoftware FreeCAD ist leider alles andere als selbsterklärend. Doch deutsche Anleitungsbücher suchte man in der Vergangenheit vergeblich. Das hat sich glücklicherweise geändert – wir stellen aktuelle Titel vor.

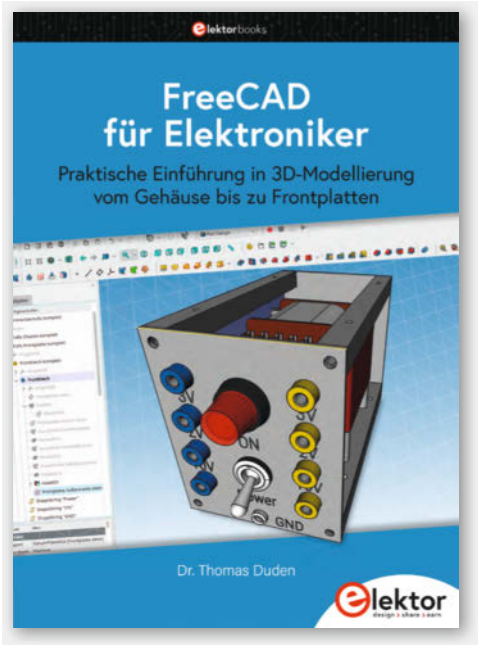
von Peter König

Praxiseinstieg

Den umfassenden Praxiseinstieg in FreeCAD soll laut Untertitel das Buch von Detlef Ridder im mitp-Verlag bieten. Nach einem Kapitel zur Installation folgt die Übersicht über ein paar ausgesuchte Arbeitsbereiche (Workbenches) der Software, die wichtige Werkzeuge fürs Zeichnen von 2D-Skizzen, deren Ableitung in die dritte Dimension sowie für deren Verfeinerung eines nach dem anderen ausführlich beleuchtet. Dieses Kapitel dürfte Einsteiger ziemlich ermüden, da sich aus Vorstellung von Werkzeugen kaum erschließt, wie man mit ihnen etwas baut; zum (späteren) Nachschlagen ist das Kapitel aber durchaus nützlich.

Das dritte Kapitel führt dann (endlich) Klick für Klick durch ein Beispielprojekt: Anhand der Konstruktion eines Schraubenschlüssels bekommt man einen guten Eindruck davon, wie man mit FreeCAD arbeitet. Vor das folgende Beispiel, die Planung eines zweistöckigen Hauses, ist wieder eine ausführliche Übersicht der dafür nötigen Arbeitsbereiche geschaltet. Steht das Haus (im Rechner), lernt man für das Gebäude wie den Schraubenschlüssel technische Zeichnungen abzuleiten und Maße einzufügen. Zum Schluss des Buchs wird noch angerissen, wie man einzelne konstruierte Teile in FreeCAD zusammenbauen kann, eine FEM-Festigkeitsanalyse durchführt, 3D-Modelle für den 3D-Druck exportiert und CNC-Fräswege erstellt.

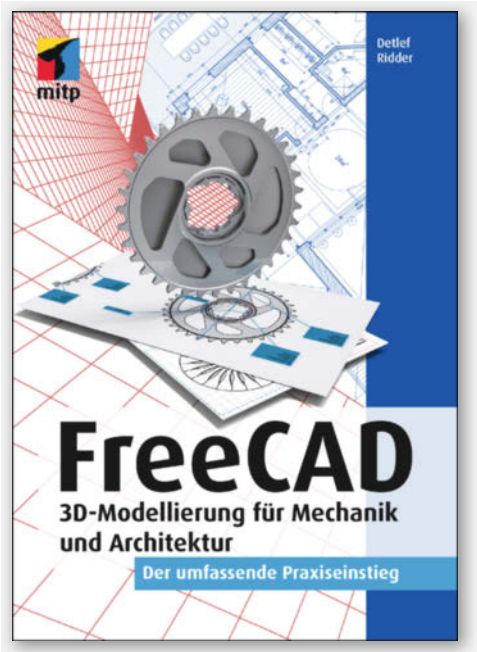
Der große Minuspunkt des Buchs sind die Abbildungen: Wo die GUI von FreeCAD mit Farben zur Markierung arbeitet, sieht man im Buch nur grau in grau. Viele Screenshots wimmeln von Markierungen, Hinweisziffern, Pfeilen und vergrößerten Elementen – was Zusammenhänge deutlich machen soll, kippt manchmal ins Unübersichtliche. Schade, denn der Autor hat offensichtlich viel Akribie in die Bearbeitung der Bilder gesteckt. Vielleicht entschließt sich der Verlag ja, die zweite Auflage komplett in Farbe zu drucken. Das würde dieses Buch sehr voranbringen.



Für Elektroniker

Das FreeCAD-Buch für Elektroniker aus dem Elektor-Verlag ist durchgehend farbig bebildert – sogar die dünnen roten und grünen Linien auf dunkelblauem Hintergrund, die FreeCAD seinen Nutzern zumutet, stechen hier gut voneinander ab. Doch auch hier gibt es ein Problem mit den Bildern, denn fast alle zeigen die komplette FreeCAD-Bedienoberfläche, aber auf maximal zwölf Zentimeter Breite gedruckt. Der reiferen Jugend ab vierzig sei der Griff zur Lupe geraten, dann lassen sich die Parameter in der Combo-Box und die Menüeinträge durchaus lesen ...

Inhaltlich geht es hier auf deutlich mehr Seiten als beim zuvor besprochenen Band um ein entschieden schmaleres Anwendungsgebiet: Elektronik-Bastler lernen, wie sie Frontplatten, Gehäuse und Montageteile für ihre Projekte in 3D planen – und zwar stets Schritt für Schritt an ganz konkreten Beispielen. So zeigt der Autor Thomas Duden den Umgang mit dem Blech-Arbeitsbereich (Workbench:



FreeCAD-Bücher

Titel	FreeCAD	FreeCAD für Elektroniker	FreeCAD for Makers
Untertitel	3D-Modellierung für Mechanik und Architektur – Der umfassende Praxiseinstieg	Praktische Einführung in 3D-Modellierung vom Gehäuse bis zu Frontplatten	Create 3D prints, laser cuts, folded sheets and more with free design software
Autor	Detlef Ridder	Dr. Thomas Duden	Jo Hinchliffe
Verlag	mitp	Elektor	Raspberry Pi Press
Umfang	296 Seiten	463 Seiten	104 Seiten
Abbildungen	schwarzweiß	Farbe	Farbe
Sprache	Deutsch	Deutsch	Englisch
ISBN/URL	978-3-74750-488-8	978-3-89576-543-8	hackspace.raspberrypi.com/books/freecad
Preis	24,99 € (Buch oder E-Book), 29,99 € (beides zusammen)	39,80 € (Buch), 32,80 € (PDF)	gratis (PDF)

© Copyright by Maker Media GmbH.

Sheet Metal) anhand eines einfachen Halters für eine 9-Volt-Blockbatterie, der am Ende als technische Zeichnung für den Zuschnitt exportiert wird.

Der Batteriehalter kommt im folgenden Beispiel, der Frontplatte eines einfachen Messgeräts, gleich zum Einsatz. Dazu Schalter, Buchsen, Schrauben, Anzeigen oder bestückte Platinen, die fertig importiert und platziert werden und für die man vor allem die nötigen Ausschnitte anlegt. Der Clou: Für viele der im Buch benutzten „Fertigteile" gibt es aber ebenfalls ausführliche Klickanleitungen, und zwar im Anhang des Buches, der ein gutes Drittel(!) des gesamten Seitenumfangs füllt. Durch diese Struktur ist das Buch einerseits gut nutzbar für Elektroniker, die nur das Nötigste in FreeCAD selber machen wollen und vielleicht auf fertige 3D-Komponenten aus dem Netz zurückgreifen, andererseits ist es aber auch für alle hilfreich, die tiefer in die mechanische 3D-Konstruktion einsteigen wollen.

Außer Konkurrenz

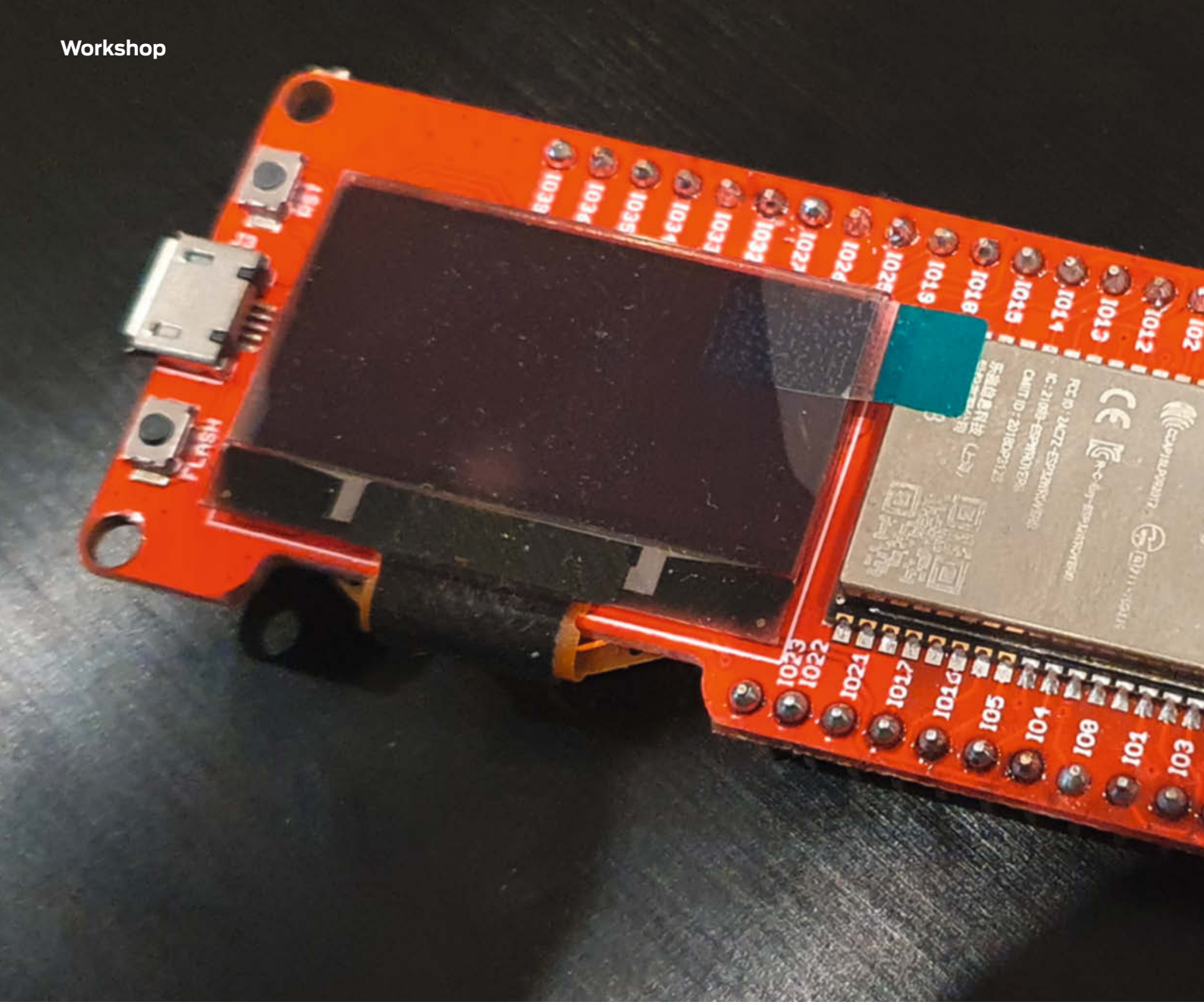
Wir möchten hier noch ein weiteres „Buch" kurz erwähnen, weil es sich zum einen explizit an Maker richtet und zum anderen kostenlos zu bekommen ist. Allerdings gibt es FreeCAD for Makers nur auf englisch und die Anführungszeichen haben wir deshalb gesetzt, weil man dieses „Buch" nicht gedruckt bekommt, sondern nur als Download. Außerdem entspricht seine ganze Aufmachung eher einem Sonderheft des ebenfalls kostenlos herunterzuladenden HackSpace-Magazins. Tatsächlich ist FreeCAD for Makers eine Sammlung aller Tutorials zur kostenlosen Konstruktionssoftware aus den vergangenen Ausgaben der HackSpace – ganz ähnlich, wie unser eigenes E-Dossier alle FreeCAD-Artikel aus Make bündelt (siehe Kasten auf Seite 109).

Auch wenn manche der Tutorials in der HackSpace-Sammlung sich explizit an Einsteiger richten (sollen), was die Einfachheit der Beispiel-Werkstücke angeht – wir würden das Gratis-PDF eher Leuten mit etwas FreeCAD-Grundkenntnissen empfehlen, die nicht jeden einzelnen Klick im Bild sehen müssen. Aber: Reinschauen kostet ja nichts. —*pek*

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

© Copyright by Maker Media GmbH.

heise Shop

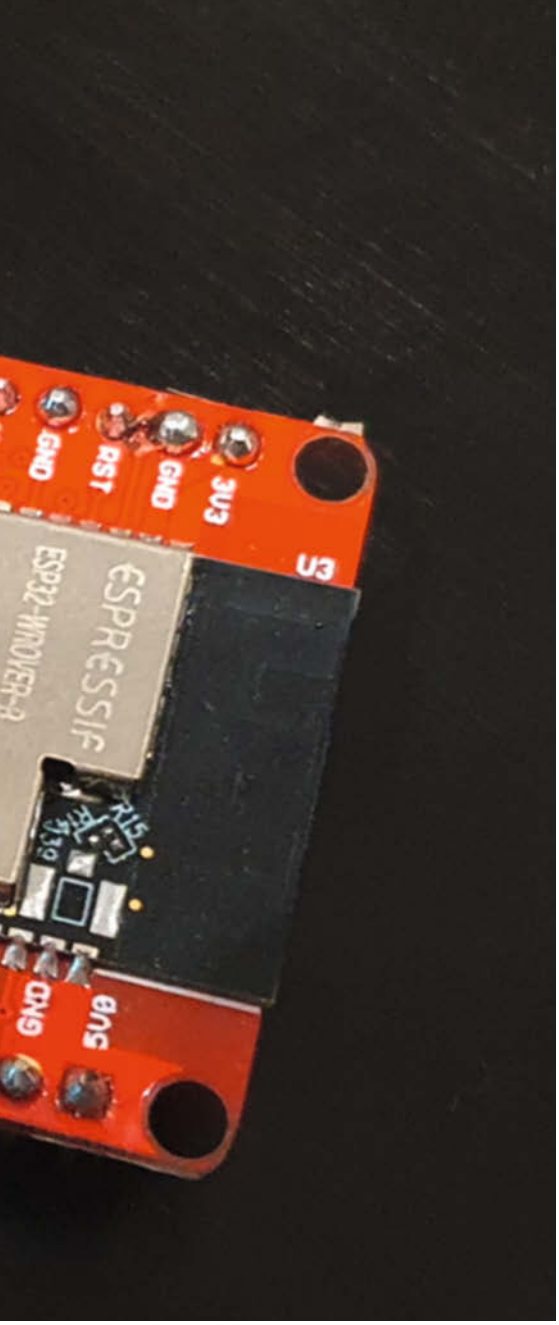


ESP32:
Wie ULP den Stromverbrauch reduziert

Der ULP-Coprozessor kann den recht hohen Stromverbrauch des ESP32-Mikrocontrollers senken. Ganz einfach ist das aber nicht – unser Ratgeber zeigt, wie es geht.

von Tam Hanna

© Copyright by Maker Media GmbH.



Kurzinfo

» Stromverbrauch des ESP32 senken
» ULP-Coprozessor verstehen
» Compiler und Optionen

Checkliste

Zeitaufwand:
1 Stunde

Kosten:
unter 10 Euro

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

Mehr zum Thema

» Dieser Artikel erschien ursprünglich auf heise+
» Daniel Bachfeld, RISC-V für Maker, Make 4/23, S. 94
» Daniel Bachfeld, Strom sparen bei ESP-Mikrocontrollern, Make 3/21, S. 84
» Maik Schmidt, ESP32 – Vollzugriff, Make 3/19, S. 94

Der ESP32-Mikrocontroller eignet sich bestens für die Umsetzung von Systemen, die über WLAN und Bluetooth kommunizieren. Der Haken: Die Leistungsaufnahme des Espressif-Controllers ist alles andere als moderat.

Nun könnte man einfach einen stromsparenden Mikrocontroller einsetzen, den ESP32 als Modul verwenden und nur bei Bedarf zuschalten. Allerdings ist es ziemlich umständlich, einen solchen zweiteiligen Aufbau umzusetzen.

Zum Glück gibt es einen einfacheren Weg: Die meisten ESP32-Varianten haben einen ULP-Coprozessor (Ultra Low Power). Dabei handelt es sich um einen winzigen Prozessor, der in den ESP32 integriert ist – aber unabhängig von den Main-Cores arbeiten kann. Er hat zudem Zugriff auf die GPIOs und den I2C-Controller. Den ULP-Prozessor kann man nutzen, um den ESP32 in einen energiesparenden Stand-by-Modus zu versetzen. Die Verwendung des ULP hat außerdem den Vorteil, dass der ESP32 auch im Ruhezustand auf eingehende Ereignisse reagieren kann.

Der ULP ist ein primitiver 8-Bit-Coprozessor, den man mit einer Assembler-Sprache programmiert. Detaillierte Systeminformationen finden Sie in dem ESP-IDF-Programming-Guide. In diesem Ratgeber demonstrieren einige Experimente den praktischen Workflow für die Arbeit mit dem ULP. Dabei kommt eine ESP32-Variante zum Einsatz, die eine klassische ULP-Zelle aufweist. Daneben gibt es etwa noch den ESP32-S2, der über einen leistungsstärkeren ULP-Coprozessor auf Basis der RISC-V-Architektur verfügt. Zudem können Sie auf ein größeres Angebot an herstellerspezifischen Platinen zurückgreifen.

Im Espressif-Zoo gibt es verschiedene Platinen, die diverse Peripherie mitbringen, etwa ein Display oder den USB-Seriell-Wandler, der für die Kommunikation mit dem PC notwendig ist. Diese Peripheriegeräte benötigen ebenfalls Energie.

Das Projekt starten

In den folgenden Schritten verwenden Sie die folgende Version des ESP-IDF (IoT Development Framework):

```
/esp4/esp-idf$ idf.py --version
ESP-IDF v4.4.1-dirty
```

Da Espressif in Version 5.0 von IDF jegliche Unterstützung für klassische Makefiles wegfallen lässt, führen Sie die Konfiguration am besten mithilfe von CMake durch. Zuerst rufen Sie ein neues Projektskelett ins Leben:

```
/esp4/heisespace$ cp -r $IDF_PATH/
     examples/get-started/hello_world
```

Das Hello-World-Beispiel pflegt die ULP-Unterstützung in ein vorhandenes Projektskelett ein. Öffnen Sie anschließend die Datei hello_world/main/CmakeLists.txt, die als Steuerungsdatei über die Kompilierung des Projektskeletts entscheidet. Dort sind Anpassungen wie in Kasten „CmakeLists.txt" durchzuführen.

Über `idf_component_register` teilen Sie der Toolchain mit, welche optionalen Komponenten bei der Kompilierung einzubinden sind. Das Code-Fragment bietet normalerweise alles, was Sie für die Arbeit mit der ULP benötigen. Dann folgen einige `set`-Befehle, bei denen es sich im Allgemeinen um übliches CMake handelt. Wichtig ist der Aufruf `set(ulp_exp_dep_srcs`: Er sorgt dafür, dass die Datei hello_world_main.c bei Änderungen des ULP-Programms ebenfalls neu kompiliert wird. Die Steuerungsdatei geht eine sehr enge Beziehung mit einigen Primitiva ein, die der ULP-Assembler im Rahmen der Verarbeitung der Datei generiert.

Legen Sie dann ein neues Arbeitsverzeichnis an, in dem die per `set(ulp_s_sources "ulp/tamsulp1.S")` angeforderte .s-Datei mit dem eigentlichen ULP-Quellcode unterkommt (im Pfad `/esp4/heisespace/hello_world/main$`):

```
mkdir ulp
cd ulp
tamsulp1.S
gedit tamsulp1.S
```

Die Kompilation der .s-Datei erfolgt unter Nutzung des Präprozessor, der auch für C-Dateien vorgesehen ist. Aus diesem Grund müssen Sie einige Header-Dateien einbinden, ohne die eine Kompilation scheitern würde:

```
#include "sdkconfig.h"
#include "soc/rtc_cntl_reg.h"
#include "soc/rtc_io_reg.h"
#include "soc/soc_ulp.h"
#include "soc/sens_reg.h"
```

© Copyright by Maker Media GmbH.

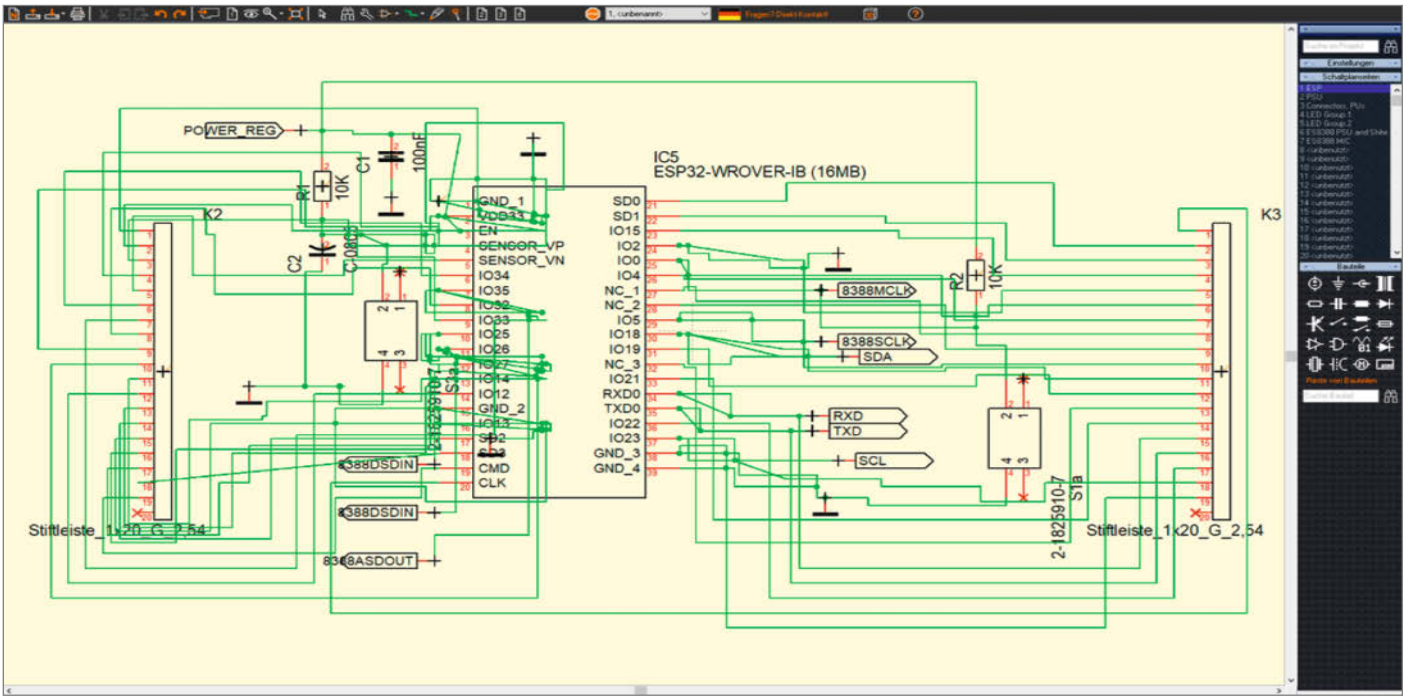


Bild 1: Der in den folgenden Tests und Experimenten verwendete Schaltungsaufbau.

Es folgt die Deklaration einer globalen Variablen, die später sowohl für den ESP32 als auch für den ULP-Teil der Lösung gleichermaßen ansprechbar sein muss. Schließlich folgt der eigentliche Einsprungspunkt:

```
.bss
    .global acti_count
acti_count:
    .long 0

.text
    .global entry
entry:
    move r3, acti_count
    ld r0, r3, 0
    add r0, r0, 1
    st r0, r3, 0
```

Die erste Aktion dieser Routine ist es, den Wert zu laden, der in der globalen Variablen enthalten ist und im Register R3 unterkommt. Dann wird dieser Wert in das Arbeitsregister verschoben, wo mit dem `add`-Befehl eine Inkrementierung erfolgt. Der `st`-Aufruf schreibt den Wert danach zurück.

Die erste Aktion dieser Routine ist es, den Wert zu laden, der in der globalen Variablen enthalten ist und im Register R3 unterkommt. Dann wird dieser Wert in das Arbeitsregister verschoben, wo mit dem `add`-Befehl eine Inkrementierung erfolgt. Der `st`-Aufruf schreibt den Wert danach zurück.

Problematisch ist allerdings, dass es im Rahmen der Übergabe zwischen dem ESP32 und dem ULP-Kern mitunter zu Race Conditions kommen kann. Der ULP-Kern darf den ESP32 nämlich erst dann neu anwerfen, wenn dieser seinen (etwas zeitaufwendigen) Sleep-Prozess hinter sich gebracht hat. Dies lässt sich durch Überprüfen des entsprechenden Bits (Bild 2) ermitteln.

Speichern Sie die .s-Datei und öffnen Sie danach menuconfig. Aktivieren Sie dort die Option und stellen Sie in der Rubrik „RTC slow memory reserved for coprocessor" den Speicherbereich auf 512 ein. Danach kompilieren Sie das Programm wie gewohnt:

```
/hello_world$ idf.py build
```

Interessant ist daran, dass die Toolchain darüber informiert (Bild 3), dass nun einige zusätzliche Dateien Teil des Build-Prozesses geworden sind.

Den C-Teil einrichten

Das erfolgreiche Kompilieren der .s-Datei stellt sicher, dass die assemblierten Programminformationen für den ULP Teil der .bin-Datei sind. Diese brennt später die Espressif-Toolchain auf den Chip. Von Haus aus ist die ULP allerdings deaktiviert – ihre Nutzung setzt voraus, dass sie in der C-Datei eingebunden ist. Hierzu müssen Sie in die im Projektskelett enthaltene Datei hello_world_main.c wechseln und einige zusätzliche Header einfügen:

```
#include "esp32/ulp.h"
#include "ulp_main.h"
#include "esp_sleep.h"
```

Besonders wichtig ist die Datei ulp_main.h. Der ULP-Teil generiert sie automatisch und stellt verschiedene Informationen bereit, wo

CmakeLists.txt

```
idf_component_register(SRCS "hello_world_main.c"
    INCLUDE_DIRS ""
    REQUIRES driver soc nvs_flash ulp)

set(ulp_app_name ulp_${COMPONENT_NAME})
set(ulp_s_sources "ulp/tamsulp1.S")
set(ulp_exp_dep_srcs "hello_world_main.c")
ulp_embed_binary(${ulp_app_name} "${ulp_s_sources}" "${ulp_exp_dep_srcs}")
```

```
exit:
    READ_RTC_REG(RTC_CNTL_DIAG0_REG, 19, 1)
    and r0, r0, 1
    jump exit, eq
    wake
    WRITE_RTC_FIELD(RTC_CNTL_STATE0_REG, RTC_CNTL_ULP_CP_SLP_TIMER_EN, 0)
    halt
```

Bild 2: Ist der Prozessor aufgewacht?

© Copyright by Maker Media GmbH.

die für das C-Programm relevanten Teile anzusprechen sind. Der Name stammt übrigens aus der Deklaration in der CMake-Datei, die in diesem Fall folgendermaßen aussieht:

```
set(ulp_app_name
    ulp_${COMPONENT_NAME})
set(ulp_s_sources "ulp/tamsulp1.S")
```

Im nächsten Schritt sind einige Überlegungen dazu anzustellen, wie der Schlafmodus des ESP32 eigentlich funktioniert. Aus der Logik folgt, dass der Schaltungsteil, der am wenigsten Energie verbraucht, von der Stromversorgung getrennt ist. Beim ESP32 teilen sich ULP und Hauptkern einen kleinen Speicherbereich (Bild 4).

Die Aktivierung des Stromsparmodus erfolgt dadurch, dass der ULP-Kern zu rechnen beginnt. Die restlichen RAM-Speicherbereiche und sonstigen Teile des Controllers werden dann über integrierte Schalter von der Versorgungsspannung getrennt, was den Strombedarf reduziert. Der von der Stromversorgung getrennte Arbeitsspeicher verliert seinen Inhalt.

In unserem Experiment entspricht eine Aktivierung des ESP32-Hauptkerns nach einer ULP-Sitzung prinzipiell der Reaktivierung eines komplett initialisierten Controllers. Deswegen beginnt die Programmausführung abermals in der Methode `app_main` (Kasten „app_main Methode“).

Um Informationen über die Ursache der Aktivierung zu bekommen, nutzen Sie die Methode `esp_sleep_get_wakeup_cause`: Mit ihr lässt sich ein Statusbit abfragen, das darüber informiert, ob der vorliegende Aufruf der Methode `app_main` entweder direkt durch Einschalten der Versorgungsspannung oder aber durch eine vom ULP-Teil befohlene Aktivierung des Hauptkerns erfolgte.

Erkennt die Methode einen kompletten Start, ruft das Programm die Methode `init_ulp_program` auf. Andernfalls wird der Wert des in der .s-Datei angelegten Zählers ausgegeben. Wichtig ist hierbei, dass die ULP-Engine – um Strom zu sparen – mit nur 16 Bit Wortbreite arbeitet. Die Nutzung des Statements `ulp_acti_count & UINT16_MAX` sorgt dafür, dass Teile der Register, die ULP nicht bearbeitet hat, nicht für die Verarbeitung genutzt werden.

Zu guter Letzt findet sich dann noch ein Aufruf der Methoden `esp_sleep_enable_ulp_wakeup`, `ulp_run` und `esp_deep_sleep_start()`. Die erste Methode hat die Aufgabe, den ESP32-Hauptkern für den ULP-Kern ansprechbar zu machen. Die beiden anderen Methoden sorgen dann dafür, dass der Hauptkern in den bereits beschriebenen stromsparenden Schlafmodus wechselt.

Im nächsten Schritt wenden Sie sich der Methode `init_ulp_program` zu. Sie hat die Aufgabe, die Binärdateien aus dem Arbeits-

```
-- Generating done
-- Build files have been written to: /home/tamhan/esp4/heisespace/hello_world/build/esp-idf/main/ulp_main
[647/1045] Performing build step for 'ulp_main'
[1/7] Generating esp32.ulp.ld
[2/7] Generating tamsulp1.ulp.S
[3/7] Building ASM object CMakeFiles/ulp_main.dir/tamsulp1.ulp.S.obj
[4/7] Linking ASM executable ulp_main
[5/7] Generating ulp_main.sym
[6/7] Generating ulp_main.bin
[7/7] Generating ulp_main.ld, ulp_main.h
[900/1045] Building C object esp-idf/libsodium/CMakeFiles/__idf_libsodium.dir/libsodium/src/libsodium/crypto_core/salsa/ref/core_s
```

Bild 3: ULP verwendet einen separaten Kompilationsprozess.

Name	Size	Address range		Governed by
		From	To	
ROM0	384 KB	0x4000_0000	0x4005_FFFF	Static MPU
ROM1	64 KB	0x3FF9_0000	0x3FF9_FFFF	Static MPU
SRAM0	64 KB	0x4007_0000	0x4007_FFFF	Static MPU
	128 KB	0x4008_0000	0x4009_FFFF	SRAM0 MMU
SRAM1 (aliases)	128 KB	0x3FFE_0000	0x3FFF_FFFF	Static MPU
	128 KB	0x400A_0000	0x400B_FFFF	Static MPU
	32 KB	0x4000_0000	0x4000_7FFF	Static MPU
SRAM2	72 KB	0x3FFA_E000	0x3FFB_FFFF	Static MPU
	128 KB	0x3FFC_0000	0x3FFD_FFFF	SRAM2 MMU
RTC FAST (aliases)	8 KB	0x3FF8_0000	0x3FF8_1FFF	RTC FAST MPU
	8 KB	0x400C_0000	0x400C_1FFF	RTC FAST MPU
RTC SLOW	8 KB	0x5000_0000	0x5000_1FFF	RTC SLOW MPU

Bild 4: Die beiden RTC-Speicherbereiche sind geteilt

app_main Methode

```
void app_main(void)
{
    printf("Hello world!\n");
    esp_sleep_wakeup_cause_t cause = esp_sleep_get_wakeup_cause();
    if (cause != ESP_SLEEP_WAKEUP_ULP) {
        printf("Normaler Start\n");
        init_ulp_program();
    } else {
    printf("ULP-Start\n");
    printf("Counter %u \n", ulp_acti_count & UINT16_MAX);
    }
    ESP_ERROR_CHECK( esp_sleep_enable_ulp_wakeup() );
    ESP_ERROR_CHECK( ulp_run((&ulp_entry - RTC_SLOW_MEM) /
sizeof(uint32_t)));
    esp_deep_sleep_start();
```

```
extern const uint8_t ulp_main_bin_start[] asm("_binary_ulp_main_bin_start");
extern const uint8_t ulp_main_bin_end[] asm("_binary_ulp_main_bin_end");
void init_ulp_program()
{
    esp_err_t err = ulp_load_binary(0, ulp_main_bin_start, (ulp_main_bin_end -
    ulp_main_bin_start) / sizeof(uint32_t));
    ESP_ERROR_CHECK(err);
}
```

Bild 5: Diese Routine setzt Start- und Endadresse des ULP-Programms.

```
/esp4/heisespace/hello_world$ idf.py flash -p /dev/ttyUSB1
Executing action: flash
[...]
/esp4/heisespace/hello_world$ idf.py monitor -p /dev/ttyUSB1
```

Bild 6: Der Flash-Prozess

© Copyright by Maker Media GmbH.

```
W (301) spi_flash: Detected size(8192k) larger than t
I (315) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
Hello world!
ULP-Start
Counter 1
ets Jun  8 2016 00:22:57
```

Bild 7: Nach der Aktivierung von ULP inkrementiert sich der im Counter gespeicherte Wert permanent.

speicher des Hauptkerns in den Programmspeicher der ULP zu übertragen. Die Konstanten `ulp_main_bin_start` und `ulp_main_bin_end` sind dabei von der Toolchain generierte Start- und Endstellen, die der Methode `ulp_load_binary` ermöglichen, das Kompilat zu lokalisieren (Bild 5).

Um das Ergebnis des Experiments zu testen, genügt es, einen Zweikampf zwischen Flash-Befehl und Aktivierung des Monitors durchzuführen. Achten Sie darauf, die mit den Pins verbundenen Knöpfe in der korrekten Reihenfolge zu aktivieren, damit der Bootloader die Kompilate entgegennimmt (Bild 6).

Abschluss im .s File

```
entry:
    move r3, acti_count
    ld r0, r3, 0
    add r0, r0, 1
    st r0, r3, 0

tamslabel:
    READ_RTC_REG(RTC_GPIO_IN_REG, RTC_GPIO_IN_NEXT_S, 16)
    move r1, r0
    rsh r0, r1, 7
    AND R0, R0, 1
    JUMP tamslabel, eq //if ALU result of and is zero
exit:
```

ULP-Initialisierung

```
void init_ulp_program()
{
    esp_err_t err = ulp_load_binary(0, ulp_main_bin_start, (ulp_main_bin_
end - ulp_main_bin_start) / sizeof(uint32_t));
    ESP_ERROR_CHECK(err);

    rtc_gpio_init(PINNO_ESP);
    rtc_gpio_set_direction(PINNO_ESP, RTC_GPIO_MODE_INPUT_ONLY);
    rtc_gpio_pulldown_dis(PINNO_ESP);
    rtc_gpio_pullup_dis(PINNO_ESP);
    rtc_gpio_hold_en(PINNO_ESP);
    rtc_gpio_isolate(PINNO_ESP);
}
```

```
I (316) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
Hello world!
ULP-Start
Counter 1
```

Bild 8: Die Statusmeldung informiert über das erfolgreiche Einlesen des GPIO-Pins.

Nach einem letzten Reset beginnt der ESP32 mit der Programmausführung. Wundern Sie sich nicht, wenn durch die wiederholten Aufrufe des Startcodes immer wieder verschiedene Statusmeldungen im Monitor erscheinen. Wichtig ist vor allem, dass die Inkrementierung der Zähler-Variablen (Bild 7) erfolgt.

Sensor-Interaktion im Stand-by

Mit den bisher umgesetzten Maßnahmen sind Sie lediglich in der Lage, den ULP als einen intelligenteren Zähler zu verwenden. Die wirkliche Schlagkraft bekommt das Teil, wenn es mit externer Hardware interagieren kann – zumindest in einem eingeschränkten Rahmen.

Es ist etwa möglich, auf Signale der GPIO-Pins zu reagieren. Außerdem kann der ULP eingeschränkt mit verschiedenen I2C-Sensoren interagieren. Damit setzen Sie einen kombinierten Prozessrechner um, in dem der Niedrigenergie-Teil den Hochenergie-Teil nur bei Bedarf aktiviert. Aufgrund des Zusammenhangs zwischen Chipfläche und Stromverbrauch gilt allerdings auch hier, dass nur ein vergleichsweise kleiner Teil des GPIO-Angebots nutzbar ist.

Um die Testimplementierung abzuschließen, müssen Sie in die .s-Datei zurückkehren und dort wie im Kasten „Abschluss im .s File" das Markup platzieren.

Besonders wichtig ist der Teil um `tamslabel`: Im ersten Schritt nutzen Sie das Makro `READ_RTC_REG`, um die Informationen im GPIO-Register in das Arbeitsregister der ULP zu laden. Danach verschieben Sie diese in ein anderes Register und führen einen Right Shift durch, um den für unser Zielbit relevanten Wert einfach ansprechbar zu machen. Der `and`-Vergleich prüft dann den Wert des Zielbits und sorgt dafür, dass das EQ-Statusbit gesetzt beziehungsweise gelöscht wird.

Sinn dieser Vorgehensweise ist, dass Sie im nächsten Schritt einen konditionalen Sprungbefehl einsetzen können. Der Sprung zu „tamslabel" wird immer dann ausgeführt, wenn der `and`-Befehl `null` zurückliefert – das GPIO-Pin also nicht hochgezogen wird. In diesem Fall springt der Miniprozessor zum Label zurück und der Lese-Prozess in Bezug auf das GPIO-Pin beginnt von Neuem.

Sofern der `and`-Befehl nicht `null` zurückliefert, wird der `JUMP`-Befehl ignoriert. Die ULP rennt daraufhin zum bekannten Aufwach-Code weiter, der den Hauptkern reaktiviert und die normale Programmausführung fortsetzt.

C-Teil anpassen

Im C-Teil sind ebenfalls Anpassungen notwendig. Wichtig ist, dass ULP und Hauptkern dasselbe GPIO-Pin über zwei unterschiedliche Konstanten ansprechen:

© Copyright by Maker Media GmbH.

```
#define PINNO_ULP 7
#define PINNO_ESP GPIO_NUM_26
```

Bei der Initialisierung der ULP sind dann einige Manipulationen erforderlich, um das Pin zur Verfügung zu stellen (Kasten „ULP-Initialisierung“).

Außerdem rufen Sie `ulp_set_wakeup_period` auf, um die Periode der ULP-Aktivierungen zu bestimmen:

```
void init_ulp_program()
{
    [...]
    ulp_set_wakeup_period(0,50*1000);
}
```

Zu guter Letzt benötigen Sie noch einige weitere Header-Dateien:

```
#include "driver/gpio.h"
#include "driver/rtc_io.h"
```

An dieser Stelle fehlt ein erneuter Aufruf von `make flash`, um das aktuelle Kompilat in den Speicher des ESP32 zu bringen. Wenn Sie den Reset-Knopf nach dem Start des Monitors drücken, stellen Sie fest, dass der ESP32 die Programmausführung nun an der folgenden Stelle anhält:

```
I (0) cpu_start: Starting
scheduler...
Hello world!
Normaler Start
```

Dies ist insofern logisch, als Sie nach dem ersten Start sofort die ULP loslassen. Diese betritt die Warteschleife, und da keine Spannung anliegt, verbringt der ESP32 den Rest in den periodischen Aktivierungen beziehungsweise der Warteschleife. Sofern Sie den Wert des GPIOs (mit einem Vorwiderstand) auf High ziehen, springt das Programm weiter und führt zum gewünschten Ergebnis (Bild 8).

Warteschleife

```
void app_main(void)
{
    printf("Hello world!\n");
    esp_sleep_wakeup_cause_t cause = esp_sleep_get_wakeup_cause();
    if (cause != ESP_SLEEP_WAKEUP_ULP) {
        printf("Normaler Start\n");
        init_ulp_program();
    } else {
    printf("ULP-Start\n");
    printf("Counter %u \n", ulp_acti_count  & UINT16_MAX);
    }

    for (int i = 10000; i >= 0; i--) {
        printf("WARTEN...\n");
    }
```

Stromverbrauch abschätzen

Wegen der Verwendung eines FTDI-Moduls als Stromversorgung ist es einfach, ein Multimeter zwischen 3V-Pin und Stromversorgung der Platine einzuschleifen. Achten Sie darauf es in einen vernünftigen Strommessbereich zu setzen und die Autorange-Funktion zu deaktivieren. Beim Messen weit divergierender Werte kommt es sonst nämlich mitunter zu einer Umschaltung des Messbereichs, was in vielen Fällen die Eingangsimpedanz verändert und zu einem seltsamen Verhalten der geprüften Schaltung führen kann.

Im Schlafmodus sehen Sie einen durchschnittlichen Verbrauch von 0,38 mA. Da der aktive Zustand des ESP32 hier sehr kurz ist, bietet es sich noch an, im Rahmen des Starts eine zusätzliche Warteschleife unterzubringen (Kasten Wartschleife).

Eine erneute Messung zeigt nun einen ungefähren Stromverbrauch von 36 mA.

Eine Reduktion von 36 mA auf 0,38 mA entspricht einem Faktor 100 – eine durchaus erhebliche Verringerung des Energieverbrauchs, die vor allem bei der Arbeit mit batteriebetriebenen Systemen sehr hilfreich sein kann.

Außer Frage steht einerseits, dass die Programmierung der ULP wegen der verwendeten Assembler-Sprache wesentlich unbequemer ist als das normale Programmieren des ESP32. Andererseits steht dem eine Reduktion des Stromverbrauchs um den Faktor 100 entgegen.

Der Fairness halber muss man hier allerdings anmerken, dass dedizierte Low-Power-Controller von Anbietern wie GigaDevice oder STMicroelectronics mit Schlafstrom im Nanobereich auskommen. Dies ist natürlich noch eine Größenordnung effizienter, erfordert aber einen Wechsel der Systemplattform. —*caw*

TECHNIKUNTERRICHT MACHT ENDLICH SPAß!

Make: Education

Mit **Make Education** erhalten Sie jeden Monat kostenlose Bauberichte und Schritt-für-Schritt-Anleitungen für einen praxisorientierten Unterricht:

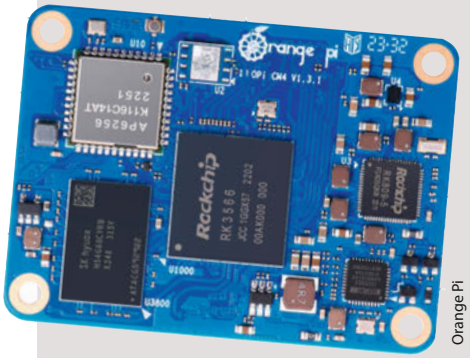
Für alle weiterführenden Schulen
Fächerübergreifend
Digital zum Downloaden
Monatlicher Newsletter

Jetzt kostenlos downloaden:
make-magazin.de/education

© Copyright by Maker Media GmbH

Orange Pi Compute Module 4

Alternative zum Raspi CM


Orange Pi

Orange Pi hat ein neues Compute Module auf den Markt gebracht. Basierend auf dem ARM Quad-Core Rockchip RK3566, der mit 1.8 GHz getaktet ist, fungiert das Modul als Drop-in-Ersatz für ein Raspberry Pi Compute Module. Offiziell ist das Modul mit den Betriebssystemen Ubuntu 22.04 und 20.04, Debian 11 und 12, OpenHarmony 4.0 Beta1 sowie Orange Pi OS und Android 11 kompatibel.

Das Orange Pi Board ist mit WiFi 5 und Bluetooth 5.0 Low Energy ausgestattet. Es werden Versionen mit 1 GB, 2 GB, 4 GB und 8 GB LPDDR4 RAM angeboten. Der integrierte eMMC-Speicher ist von 8 GB bis 128 GB verfügbar.

Als Compute Module verfügt das Board über zwei 100-polige Board-to-Board-Steckverbinder, über die Internet-, Display-, SATA-, USB- und Audio-Verbindungen zwischen Compute Module und dem Basismodul, auf das es aufgesteckt wird, hergestellt werden. Welche physikalischen Verbindungen letztendlich genutzt werden können, hängt vom gewählten Basisboard ab. Ohne ein Basisboard, das zumindest die Stromversorgung und den Speicherzugriff sicherstellt, kann das Compute Module nicht verwendet werden.

Passend zum Compute Module bietet Orange Pi ein Basisboard mit Standard-, LAN-, USB-, Audio-, GPIO-, HDMI-, Kamera- und LCD-Anschlüssen an. —*das*

Hersteller	Orange Pi
URL	www.orangepi.org
Preis	55 € (Amazon/AliExpress)

4inch DSI Round Touch Display

Rundes Display mit Touchscreen für Raspberry Pi

Runde Displays gibt es bislang nur wenige und dann meist nur in einer Größe von zwei Zoll. Touchscreens oder größere Durchmesser suchte man bislang vergebens. Waveshare hat nun ein IPS-Display mit 4 Zoll Durchmesser im Programm, das außerdem Eingaben per Touch unterstützt. Das Display wird an den DSI-Connector der Raspberry-Pi-Einplatinencomputer angeschlossen, der Touchscreen kommuniziert über I²C. Auf der Rückseite des Displays findet sich neben den Steckanschlüssen auch noch die Möglichkeit, die I²C-Adresse einzustellen. So können am selben Bus eines Raspis zwei dieser Anzeigeeinheiten angesteuert werden.

Die Auflösung des Displays beträgt 720 × 720 Pixel bei maximal 60 Hertz Bildfrequenz und 170 Grad Betrachtungswinkel. Die Oberfläche besteht aus gehärtetem Glas. Der Touchscreen arbeitet kapazitiv und mit 10-Punkt-Erkennung. Der Außendurchmesser beträgt 128 mm bei einer Tiefe von 17 mm. Auf der Platine an der Rückseite befinden sich vier Abstandshalter, auf die man wahlweise einen Raspi 3 oder Raspi 4 schrauben kann.

Waveshare

Die Stromversorgung des Bildschirms erfolgt über den Raspi. Waveshare stellt ein ausführliches Wiki zur Hard- und Software-Installation bereit. Außerdem gibt es vorkonfigurierte Linux-Images für Raspberry Pi 3 und 4. Helligkeit und Ausrichtung des Bildes lassen sich per Software und über den Touchscreen steuern. —*hgb*

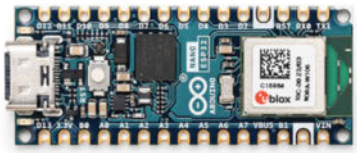
Hersteller	Waveshare
URL	make-magazin.de/xk5r
Preis	90 US-$, ca. 120 € bei Amazon

Arduino Nano ESP32 für IoT

Nachwuchs für die Nano-Familie

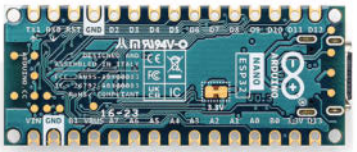
Frisch aus dem Hause Arduino und damit großteils aus Europa kommt der Arduino Nano ESP32. Das beliebte Nano-Format gibt es nun auch in einer Version mit dem verbreiteten ESP32. Das Nano ESP32 kommt mit Wi-Fi/Bluetooth, 8 MB internem und 16 MB externem Flash, 512 kB RAM, USB-C (HID-fähig) und einer RGB LED.

Die Familie der pin-kompatiblen Nano-Boards reicht vom Klassiker mit ATMega328 über das Nano Every mit ATMega4809, das mit 5 V arbeitet, bis hin zu Boards mit nRF52840, RP2040 und jetzt ESP32 Mikrocontrollern, die mit 3,3 V arbeiten. Das muss man beachten, wenn man einen älteren Nano ersetzen will, um den Chip nicht mit 5-V-Peripherie zu zerstören. Der Nano ESP32 arbeitet auch mit 3,3 V, kann aber über USB-C mit 5 V oder über einen integrierten Spannungsregler mit einer Spannung von 6 bis 12 V versorgt werden.



Der ESP32 ist ein u-blox NORA-W106 und somit ein ESP32-S3, der mit bis zu 240MHz läuft. Mit der ArduinoIDE 2 kann man sich die Bits und Bytes des Boards im Einzelschritt-Debugger ansehen, ohne weitere Hardware zu benötigen.

Neben Arduino-Code wird auch Micro Python offiziell von Arduino unterstützt. Neben dem praktischen Formfaktor, der sowohl für Breadboards als auch für Platinenprojekte geeignet ist, bekommt man natürlich ein Board, das von Arduino perfekt dokumentiert ist und für IoT-Projekte über die Arduino-Cloud oder andere Anbieter mit dem Internet verbunden werden kann. Und das sind dann auch die Gründe, warum man zum Nano greift und nicht irgendein ESP32-Board aus Fernost verwendet. —*caw*


Arduino.cc

Hersteller	Arduino.cc
URL	store.arduino.cc/products/nano-esp32
Preis	18 €

© Copyright by Maker Media GmbH.

Anker Make M5C

Schneller 3D-Drucker mit schönen Ergebnissen

Mit dem AnkerMake M5C hat der chinesische Hersteller Anker seinen zweiten 3D-Drucker auf den Markt gebracht. Optisch und technisch liegt er nah bei seinem Vorgänger M5, verzichtet allerdings auf einen Touchscreen sowie die Kamera samt KI-Überwachung. Außerdem hat er mit 220 × 220 × 250 mm auf der X- und Y-Achse jeweils 15 mm weniger Baufläche. Dafür ist der M5C aber genauso schnell und mit 449 Euro ganze 250 Euro günstiger als der M5. In Filamentform verarbeitet er PLA, PETG, ABS, TPU und sogar Nylon (PA) und wird als vormontierter Bausatz in zwei Teilen geliefert (Werkzeuge inklusive), sodass er schon nach 20 Minuten einsatzbereit ist.

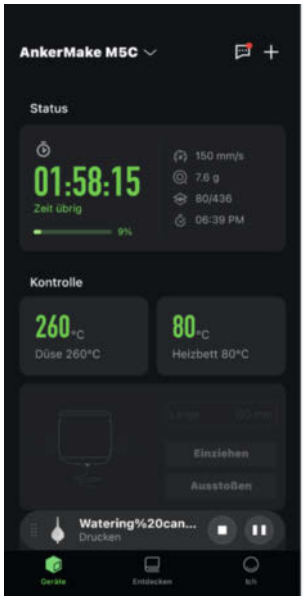
Da der M5C keinen Touchscreen besitzt, benötigt man ein Smartphone und die AnkerMake-App für iOS oder Android, um ihn zu steuern. Außerdem muss man sich bei Anker registrieren, bevor man den M5C koppeln darf. Wer mag, kann den 3D-Drucker beim Einrichten mit seinem WLAN verbinden. Dann hat man auch die Option, von unterwegs oder vom Computer aus über das Internet auf den M5C zuzugreifen. Ist die WLAN-Verbindung unterbrochen oder nicht gewünscht, kann die Smartphone-App den 3D-Drucker auch per Bluetooth steuern. Die App ist einsteigerfreundlich gestaltet und gibt Zugang zu den Grundfunktionen des Druckers sowie einigen Einstellungen, die anders nicht erreichbar sind.

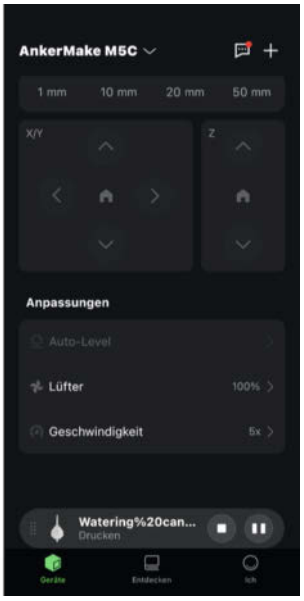
Als einzige physische Interaktionsmöglichkeit – wenn man die versteckte Reset-Taste neben dem USB-C-Anschluss nicht mitzählt – hat Anker einen Start-/Pause-Knopf verbaut, der prominent neben der Druckfläche platziert ist. Diesen kann man auf verschiedene Arten nutzen und einstellen, was er beim einmaligen, doppelten oder langen Drücken tun soll. Ein laufender Druck lässt sich etwa pausieren und fortsetzen oder vollständig abbrechen.

Auch wenn der M5C auf die KI-Funktion des M5 verzichtet, Fehldrucke also weder erkennen noch abbrechen kann, besitzt sein Druckkopf zumindest einen Filament-Sensor, mit dessen Hilfe das Gerät den laufenden Druckauftrag pausieren kann, wenn kein Material mehr beim Extruder ankommt. Nach einem Stromausfall druckt der M5C ebenfalls weiter.

3D-Modelle lassen sich im AnkerMake-Slicer vorbereiten und bequem über den AnkerMake-Account per WLAN an den Drucker schicken. Wer bereits mit Cura oder dem PrusaSlicer arbeitet und diese auch für den M5C nutzen will, kann sich auf der AnkerMake-Website die Druckerprofile für den großen Bruder M5 herunterladen.

Der AnkerMake M5C kann sich sehr schnell bewegen, was sich vor allem bei größeren Modellen auf die Druckzeit auswirkt. Die Geschwindigkeitsunterschiede zwischen dem normalen (250mm/s) und schnellen (500mm/s) Modus sind allerdings nicht ganz so groß wie erwartet. Auch erreicht der 3D-Drucker die beworbenen 500mm/s nur, wenn kein Material extrudiert wird, ansonsten fährt er maximal mit 270mm/s – im Vergleich zu den 70 mm/s eines UltiMaker S5 läuft der M5C aber natürlich immer noch wie ein geölter Blitz und produziert selbst bei diesen Geschwindigkeiten sehr ansprechende Ergebnisse.





Mit seiner Bodenplatte aus Aluguss und den extrudierten Profilen wirkt der M5C nahezu unzerstörbar und ist doch so konstruiert, dass man schnell an die kritisch relevanten Stellen kommt und sich durch die aufgeräumte Kabelführung so schnell nichts verheddert. Ein weiterer Pluspunkt ist außerdem der von AnkerMake entwickelte Direct-Extruder, dessen Zahnräder bündig mit dem Kühlergrill des Hotends abschließen. Dadurch minimieren sich Filamentprobleme im Hotend, zum Beispiel durch eigenwilliges TPU-Filament. Ersatzteile für Reparaturen gibt es auf der AnkerMake-Website zurzeit noch recht wenig, dafür aber zahlreiche Videos und Dokumente mit Anleitungen. Einzig die Lautstärke des M5C hat uns etwas gestört, sodass man ihn trotz der schicken Optik wohl doch lieber in der Werkstatt lassen sollte.

Insgesamt ist der AnkerMake M5C ein solider und schneller 3D-Drucker, der in unseren Tests im „Easy Mode" auf Anhieb zuverlässige Druckergebnisse abgeliefert hat. Schade finden wir, dass man bestimmte Funktionen und Einstellungen nur in der Smartphone-App findet, für dessen Nutzung man sich unbedingt registrieren muss. Wem das nichts ausmacht, der darf sich mit dem AnkerMake M5C auf ein unkompliziertes 3D-Druckererlebnis freuen und erhält für weniger Geld eine gute Alternative zum AnkerMake M5.

Online gibt es für Abonnenten einen ausführlicheren Testartikel zum M5C (siehe Link). Anker hat uns den Drucker freundlicherweise zur Verfügung gestellt. *—akf*

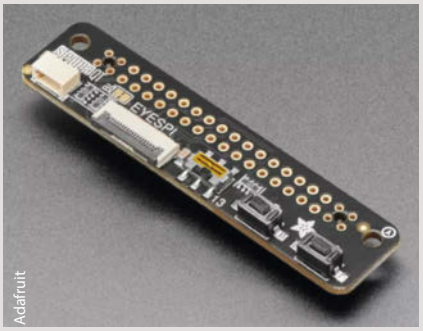
▶ make-magazin.de/xk5

Hersteller	Anker Innovations Limited
URL	www.ankermake.com/eu-de
Preis	449 €

© Copyright by Maker Media GmbH.

Adafruit EYESPI Pi Beret

Flexkabelstandard zur Display-Verbindung



Adafruit

Mit diesem Adapterboard können Displays über lange flexible Kabel am Raspberry Pi betrieben werden. Raspberry Pis werden oft ohne angeschlossene Ein- und Ausgabegeräte betrieben. Ab und zu wird jedoch ein Display zum Betrieb benötigt. Einige können direkt auf die GPIO-Leiste gesteckt werden, andere lassen sich per Flachbandkabel abgesetzt anschließen. Mit EYESPI bietet der Hersteller Adafruit ein komplettes System zur Verkabelung per Flachbandkabel an. Mit dem Adafruit EYESPI Pi Beret HAT kann man passende kleine Displays per Flachbandkabel an den Raspberry Pi anschließen, ohne ihn fest mit dem Computer zu verbinden.

Über einen FPC (Flexible Printed Circuit)-Stecker, den sogenannten EYESPI-Connector, kann ein Display über ein 18-adriges Flexkabel angeschlossen werden. Zusätzlich verfügt der Adafruit EYESPI Pi Beret über zwei frei programmierbare Tasten, einen Schiebeschalter und einen 3-poligen Stemma QT-Stecker. Über diesen können verschiedene Erweiterungen an den Raspberry Pi angeschlossen werden.

EYESPI Displays sind in verschiedenen Größen und Auflösungen im Adafruit Store erhältlich. Es werden sowohl LCD, OLED als auch E-Ink Displays angeboten. —*das*

Hersteller	Adafruit
URL	www.adafruit.com/product/5783
Preis	4,95 US-$

ESP32-S2-DevKit-Lipo-USB

Linux fähiges ESP32 Board

Vor zwei Jahren hat Olimex das ESP32-S2-DevKit-Lipo-USB herausgebracht. Nun ist der Nachfolger mit dem neuen ESP32-S3 erschienen, der auch den Betrieb von Linux auf einem ESP32 ermöglicht.

Das ESP32-S3-DevKit-LiPo ist ein Entwicklungsboard für das ESP32-S3, das mit einer Lithium-Polymer-Ladeschaltung ausgestattet ist. Dies ermöglicht den Betrieb sowohl über USB als auch über einen Lithium-Polymer-Akku, dessen Daten auch direkt über das Dev-Board überwacht werden können. Hierfür ist eine Steckverbindung auf dem Board installiert. Im Gegensatz zum Vorgängermodell verfügt das ESP32-S3-DevKit-Lipo über zwei USB-C-Anschlüsse – einen für die Stromversorgung und Datenübertragung und einen für serielle Verbindungen. Zusätzlich sind 45 GPIO-Pins vorhanden.

Das Board basiert auf einem ESP32-S3-WROOM-Modul mit 8 MB Flash-Speicher und 8 MB RAM. Der integrierte Chip läuft mit 240 MHz und unterstützt WLAN- und Bluetooth-Verbindungen. Das Board kann auch mit Linux ab Kernel 6.3 betrieben werden, ein passendes Image wird auf GitHub zur Verfügung gestellt.

Olimex GitHub

Das Board und die Software stehen unter der Open-Source-Lizenz GPL-3.0, d.h. alle Schaltpläne (KiCad) und die Software sind im Quellcode auf GitHub verfügbar. Bei Olimex ist auch der 8-Bit-Rechner Agon Light erhältlich, der bereits in Make 3/23 ausführlich getestet wurde. —*das*

▶ make-magazin.de/xk5r

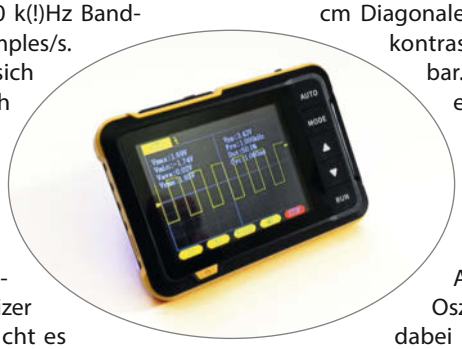
Hersteller	Olimex
URL	https://www.olimex.com/
Preis	12 €

Taschen-Oszilloskop

Fnirsi DSO 152

Ausprobiert —von Make:—

Ein tragbares und universelles Oszilloskop für rund 30 Euro bleibt leider auch mit diesem Gerät ein Traum. Hat man die passenden Messaufgaben für das kleine Ding, bekommt man allerdings viel Funktion für sein Geld. Zunächst die harten Fakten: 200 k(!)Hz Bandbreite, 1 Kanal, 2,5 MSamples/s. Damit disqualifiziert sich das DSO 152 eigentlich schon für viele Messaufgaben. Wenn man es aber nebenbei bei der Arbeit an Audioschaltungen oder wie ich zur Anzeige von Wellenformen am Synthesizer einsetzen möchte, reicht es aus. Die Bedienung ist einfach: Mit einem Dreh-Klick-Schalter auf der Oberseite des Gerätes wechselt man die Parameter, mit den Pfeiltasten nach oben und unten ändert man Parameter wie Zeitbasis, Eingangsverstärkung etc. MODE schaltet zwischen Single, Auto und Normal um, RUN friert das aktuelle Bild ein. AUTO startet einen etwas langsamen Autoadjust-Prozess. Einige seltener benutzte Funktionen verstecken sich hinter längeren Tastendrücken, sind aber alle auf zwei Seiten im winzigen Handbuch nachzulesen und schnell zu merken.



Das Display ist mit 320 × 240 Pixeln bei 7 cm Diagonale ordentlich aufgelöst, kontrastreich und gut ablesbar. Die Stromversorgung erfolgt über einen 1000-mAh-Akku oder über USB-C, über das der Akku natürlich auch geladen wird. Durch die geringen Abmessungen ist das Oszi ideal, um es immer dabei zu haben oder in ein Gerät zu integrieren. Mit dem DSO 138 gibt es auch eine makerfreundliche Variante als Platinensandwich. —*caw*

Hersteller	Fnirsi (Shen Zhen Shi Fei Ni Rui Si Technology Co., Ltd.)
URL	www.fnirsishop.com
Preis	ca. 30 €

© Copyright by Maker Media GmbH.

Inkplate 6plus

E-Ink-Display mit Touch und ESP32

Ausprobiert — von Make: —

Keine andere Technik ist stromsparender als elektronisches Papier und somit ideal, um den Status von Geräten oder Dingen anzuzeigen. Die Inkplate 6plus des kroatischen Herstellers Soldered ist 6 Zoll groß und hat eine Auflösung von 1024 × 768 Pixel, unterstützt Graustufen und lässt sich per Zweifinger-Touch (kapazitiv) bedienen. Die Zeit für einen (partial) Refresh des Bildschirminhalts ist mit 1,3 Sekunden erträglich.

Die Displays stammen aus recycelten Kindles von Amazon, die Platine ist eine Neuentwicklung. Mit dem integrierten ESP32 und seinem WLAN lässt sich das Display leicht für eigene Projekte einsetzen. Zum Reinschnuppern liefert der Hersteller in seiner Arduino-Bibliothek mehrere Beispiele mit, etwa eine Anzeige für Openweather. Zum Kompilieren mussten wir allerdings die Pfade der Include-Dateien anpassen, dann lief es.

Um Touch-Eingaben zu ermitteln, ruft man die Funktion `touchIn-Area` auf. Sie prüft, ob in einem bestimmten Bereich eine Berührung zu messen ist, etwa auf einem grafischen Button. Mit `tsGetData` kann man die aktuellen Koordinaten der berührten Stelle ermitteln. Alternativ zu Arduino gibt es Bibliotheken für MicroPython und eine API für ESP-IDF.

Auf der Platine ist ein Anschluss für ein LiPo-Akku sowie ein SD-Kartenslot vorhanden. Die GPIOs des ESP32 sind an den Rand der Platine geroutet, sodass man eigene Erweiterungen (SPI, I²C) anschließen kann. An eine sogenannte easyC-Buchse kann man Sensor-Boards von Soldered anschließen. Rund 180 Euro sind zwar kein Pappenstiel, dafür wird das Produkt in Europa entwickelt und gefertigt. —*dab*

Hersteller	Soldered, Kroatien
URL	soldered.com
Preis	179 €

Color Kit Grande

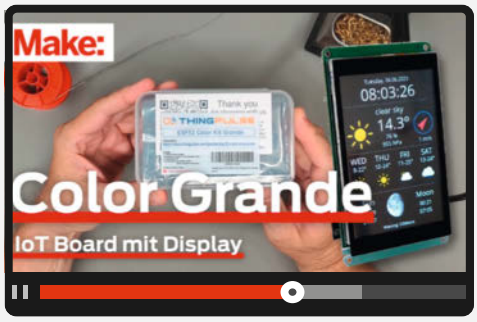
IoT-Starterkit zum selber löten

Ausprobiert — von Make: —

Das Color Kit Grande der Schweizer Entwickler Thing Pulse ist ein ESP32-basierter Bausatz mit einem 320 × 480 Pixel großen Touchscreen, ideal für IoT-Experimente. Das Board basiert auf dem ESP32 ePulse Feather des gleichen Herstellers und wird als Bausatz mit einem Farbdisplay geliefert, das ein kapazitives Touchinterface bietet. Im Gegensatz zu resistiven Touchscreens sind kapazitive Touchscreens wesentlich genauer und besser in der Erkennung und müssen wie bei einem Smartphone kaum berührt werden.

Thing Pulse

Das Board ist für Maker gedacht, die gerne basteln und löten. Aber keine Sorge, die kleinen SMD-Bauteile und die Anschlüsse für das Display sind bereits verlötet und so beschränkt sich das Löten auf ein paar Pin-Header und einen Schalter in Durchstecktechnik (THT) und einen Groove-Connector (SMD, auch für Lötanfänger geeignet). Wir hatten keine Probleme beim Löten (Siehe unser Video in den Links), aber die hochwertigen gedrehten Pin-Header für das Board und den ESP erfordern eine gute Ausrichtung, wenn sie später passen sollen. Alle Informationen gibt es auf der Shop-Seite und auf GitHub (siehe Link). Dort findet man auch die Dokumentation und Videos, die den Aufbau und Beispielprojekte zeigen. Das Demoprojekt ist eine hübsche Wetter-App, die ihre Daten über die API der OpenWeatherMap bezieht. Die Entwicklung und das Hochladen der Demo erfolgt mit Visual Studio Code und der kostenlosen Erweiterung PlatformIO. Eine gute Gelegenheit, einmal diese professionellen Tools jenseits der Arduino IDE auszuprobieren.



Da das in der Schweiz entwickelte Board in China produziert wird, muss man zum Dollarpreis im Shop noch die Steuer und eventuelle Zollgebühren hinzurechnen, was nach Deutschland ca. 53 Euro ergibt. —*caw*

▶ make-magazin.de/xk5r

Hersteller	ThingPulse Ltd.
URL	make-magazin.de/xk5r
Preis	39 US-$ (siehe Text)

© Copyright by Maker Media GmbH.

IMPRESSUM

Make: Nächste Ausgabe erscheint am 27. Oktober 2023

Redaktion

Make: Magazin
Postfach 61 04 07, 30604 Hannover
Karl-Wiechert-Allee 10, 30625 Hannover
Telefon: 05 11/53 52-300
Telefax: 05 11/53 52-417
Internet: www.make-magazin.de

Leserbriefe und Fragen zum Heft: info@make-magazin.de

Die E-Mail-Adressen der Redakteure haben die Form xx@make-magazin.de oder xxx@make-magazin.de. Setzen Sie statt „xx" oder „xxx" bitte das Redakteurs-Kürzel ein. Die Kürzel finden Sie am Ende der Artikel und hier im Impressum.

Chefredakteur: Daniel Bachfeld (dab) (verantwortlich für den Textteil)

Stellv. Chefredakteur: Peter König (pek)

Redaktion: Heinz Behling (hgb), Johannes Börnsen (jom), Ákos Fodor (akf), Daniel Schwabe (das), Dunia Selman (dus, Social Media), Carsten Wartmann (caw)

Mitarbeiter dieser Ausgabe: Debra Ansell, Beetlebum (Comic), Thomas Euler, Michael Gaus, Tam Hanna, Bernd Heisterkamp, Jan Kipping, Andreas Koritnik, Miguel Köhnlein, Matthias Mett, Markus Mierse, Jan Wegener

Assistenz: Susanne Cölle (suc), Christopher Tränkmann (cht), Martin Triadan (mat)

Leiterin Produktion: Tine Kreye

DTP-Produktion: Martina Bruns, Martin Kreft (Korrektorat)

Art Direction: Martina Bruns (Junior Art Director)

Layout-Konzept: Martina Bruns

Layout: Nicole Wesche

Fotografie und Titelbild: Andreas Wodrich

Digitale Produktion: Kevin Harte, Thomas Kaltschmidt, Pascal Wissner

Hergestellt und produziert mit Xpublisher: www.xpublisher.com

Verlag

Maker Media GmbH
Postfach 61 04 07, 30604 Hannover
Karl-Wiechert-Allee 10, 30625 Hannover
Telefon: 05 11/53 52-0
Telefax: 05 11/53 52-129
Internet: www.make-magazin.de

Herausgeber: Christian Heise, Ansgar Heise

Geschäftsführung: Ansgar Heise, Beate Gerold

Anzeigenleitung: Daniel Rohlfing (-844) (verantwortlich für den Anzeigenteil), mediadaten.heise.de/produkte/print/das-magazin-fuer-innovation

Leiter Vertrieb und Marketing: André Lux (-299)

Service Sonderdrucke: Julia Conrades (-156)

Druck: Dierichs Druck + Media GmbH & Co.KG, Frankfurter Str. 168, 34121 Kassel

Vertrieb Einzelverkauf:
DMV DER MEDIENVERTRIEB GmbH & Co. KG
Meßberg 1
20086 Hamburg
Telefon: +49 (0)40 3019 1800
Telefax: +49 (0)40 3019 1815
E-Mail: info@dermedienvertrieb.de
Internet: dermedienvertrieb.de

Einzelpreis: 13,50 €; Österreich 14,90 €; Schweiz 26.50 CHF; Benelux 15,90 €

Abonnement-Preise: Das Jahresabo (7 Ausgaben) kostet inkl. Versandkosten: Inland 80,50 €; Österreich 88,90 €; Schweiz 123.90 CHF; Europa 95,20 €; restl. Ausland 100,80 €

Das Make-Plus-Abonnement (inkl. Zugriff auf die App, Heise Magazine sowie das Make-Artikel-Archiv) kostet pro Jahr 6,30 € Aufpreis.

Abo-Service:

Bestellungen, Adressänderungen, Lieferprobleme usw.:
Maker Media GmbH
Leserservice
Postfach 24 69
49014 Osnabrück
E-Mail: leserservice@make-magazin.de
Telefon: 0541/80009-125
Telefax: 0541/80009-122

Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden. Kein Teil dieser Publikation darf ohne ausdrückliche schriftliche Genehmigung des Verlags in irgendeiner Form reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Alle beschriebenen Projekte sind ausschließlich für den privaten, nicht kommerziellen Gebrauch. Maker Media GmbH behält sich alle Nutzungsrechte vor, sofern keine andere Lizenz für Software und Hardware explizit genannt ist.

Für unverlangt eingesandte Manuskripte kann keine Haftung übernommen werden. Mit Übergabe der Manuskripte und Bilder an die Redaktion erteilt der Verfasser dem Verlag das Exklusivrecht zur Veröffentlichung. Honorierte Arbeiten gehen in das Verfügungsrecht des Verlages über. Sämtliche Veröffentlichungen in Make erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes.

Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.

Published and distributed by Maker Media GmbH under license from Make Community LLC, United States of America. The 'Make:' trademark is owned by Make Community LLC Content originally partly published in Make: Magazine and/or on www.makezine.com, ©Make Community LLC 2023 and published under license from Make Community LLC. All rights reserved.

Printed in Germany. Alle Rechte vorbehalten. Gedruckt auf Recyclingpapier.

© Copyright 2023 by Maker Media GmbH

ISSN 2364-2548

Nachgefragt

Wie hat der 3D-Druck Deine Herangehensweise an Projekte beeinflusst?

Jan Wegener

Worms, testet ab Seite 8 Filamente gründlich durch.

Der 3D-Druck bietet unglaublich viele Möglichkeiten und löst die Abhängigkeiten von fertigen Normteilen. Ich merke, dass ich 3D-druckend „denke", wenn ich etwas neu konstruiere, repariere oder modifiziere.

Markus Mierse

München, stellt auf S. 68 seinen Holo-Projektor vor.

Meine Projekte sind stark durch Funktionsteile geprägt, daher war der 3D-Drucker in meiner Werkstatt eine echte Offenbarung. Vorher war ich auf eine einfache (selbstgebaute) CNC-Portalfräsmaschine und meine Werkzeuge beschränkt.

Thomas Euler

Tübingen, überwacht auf S. 66 sein Balkonsolarkraftwerk.

Früher habe ich eher nach etwas Vorhandenem gesucht, das ich umbauen konnte; mit dem 3D-Drucker bin ich freier in der Projektgestaltung. Aber es ist auch erschreckend, wie viel Müll die Drucker produzieren.

Jan Kipping

Stuttgart, steuert auf S. 54 einzelne Geräte.

Seit ich mich ins Konstruieren von mechanischen Teilen eingearbeitet habe, ist der Teil von Projekten viel einfacher. Vorher habe ich ewig gewerkelt oder passende Gehäuse gesucht. Jetzt geht es nur noch um Software und die Funktion.

Inserentenverzeichnis

AFAG Messen und Ausstellungen GmbH, Nürnberg 17
dpunkt.verlag GmbH, Heidelberg 23
OXON AG, Liebefeld 43
TUXEDO Computers GmbH, Augsburg 124
Weidmüller GmbH & Co. KG, Detmold 2

© Copyright by Maker Media GmbH.

Ein Veranstaltungsformat der Make:
make-magazin.de

VIELEN DANK an alle diesjährigen Teilnehmer!

Wir sehen uns im nächsten Jahr.

Hannover
Maker Faire®

Das Format für Innovation und Macherkultur

maker-faire.de

© Copyright by Maker Media GmbH.

Bürocomputer für die Westentasche

TUXEDO Aura 14 - Gen3

Teilaluminiumgehäuse
2 cm dünn | 1,4 kg leicht

14,0'' Full-HD-Display
1920 x 1080 | 100 % sRGB

Intel Core i5-1235U
10 Kerne | 12 Threads

USB-C-Power-Delivery
49-Wh-Akku | bis zu 5 Stunden

100% Linux

5 Jahre Garantie

Lifetime Support

Gefertigt in Deutschland

Deutscher Datenschutz

Support vor Ort

TUXEDO

tuxedocomputers.com

© Copyright by Maker Media GmbH.