



Der neue
Raspi 5
im Test

Bausätze für Maker

- Aus Holz, Karton und zum Löten
- Roboter und Funktionsmodelle



Workshop

- GUIs für ESP32 programmieren
- Saugerkupplungen auf Maß drucken
- Pop-up-Karten selber machen

Projekte

- IKEA-PUGG-Uhr aufmotzen
- Mit Pi: Ambilight für TVs
- Head-up-Display für die Skibrille

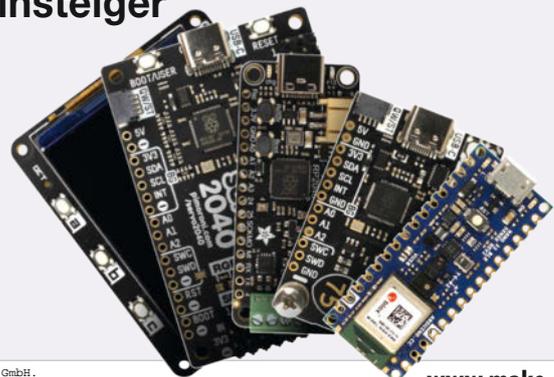
WLAN-Waage

- Billigwaage mit ESP32 aufwerten
- Mit Farbdisplay ausstatten
- In Home Assistant einbinden



Boards für MicroPython

- Mit Display, Sensoren und Servocontroller
- Test & Praxis-Tipps für Einsteiger



7/23
8.12.2023
CH CHF 26.50
AT 14,90
Benelux 15,90
€ 13,50





Life is
what you
Make:
it

Hannover

Maker Faire®

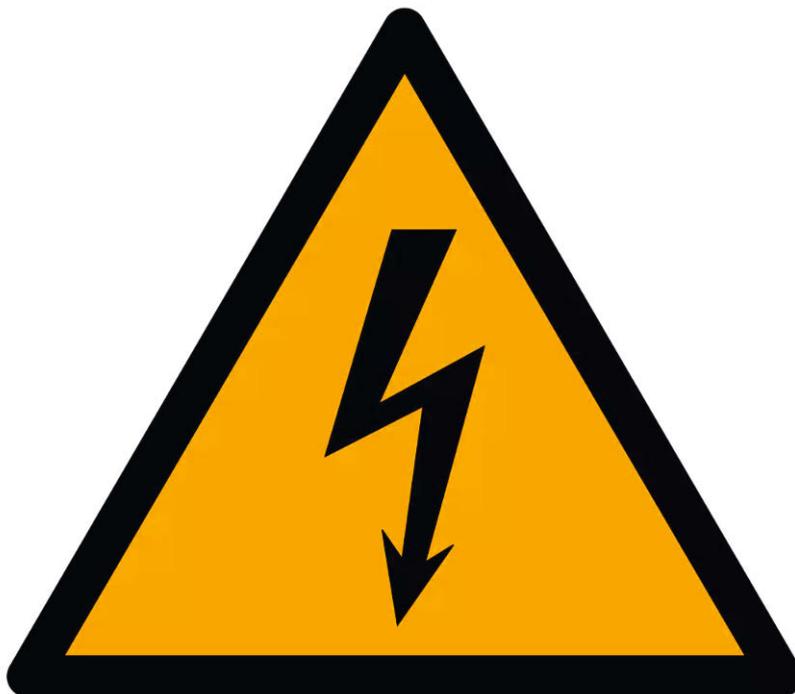
17.–18. Aug. 2024

Sei dabei!

Bewirb dich jetzt als Maker
für eine kostenfreie Standfläche!

Werde Teil
unserer
Community!

www.maker-faire.de/hannover



German Angst

Eigentlich war es mit Ansage: Unser Reparatur-Sonderheft hat wieder viele Bedenkenträger dazu veranlasst, unsere ihrer Meinung nach unverantwortlichen Anleitungen und Tipps als lebensgefährlich zu brandmarken. Zwar warnen wir in den Artikeln, in denen Gefahr besteht, mit 230 Volt in Berührung zu kommen, und erklären, wie man sich schützen kann und welche Vorsorgemaßnahmen man ergreifen sollte. Und dass man im Zweifel lieber eine Fachkraft ranlassen soll. Trotzdem gibt es immer einen Aufstand. Am schlimmsten hat es diesmal den Artikel zu Schaltnetzteilen in Waschmaschinen im heise-Forum abbekommen: Tod, Verderben und Siechtum drohen allen, die versuchen würden, den einen Kondensator auszutauschen.

Wir bei der Make nehmen das Thema nicht auf die leichte Schulter, dennoch denke ich, dass die Reaktionen völlig überzogen sind. Erstens traue ich unseren Lesern durchaus zu, ihr Können und die Gefahren richtig einzuschätzen. Und vorher den Stecker zu ziehen bzw. nicht mit der bloßen Hand zu prüfen, ob Spannung anliegt. Zweitens stelle ich mir die Frage, warum bei Themen rund um 230 V die Reaktionen so heftig ausfallen. Beruhen sie auf persönlichen Verlusten von Freunden oder Angehörigen im eigenen Umfeld? Aus Erfahrungen am eigenen Leibe? Oder weil man von anderen gehört hat, die jemanden kennen ...?

Schaut man sich die zurückliegenden Statistiken an, gab es laut VDE im Jahr 2019 32 Stromunfälle mit Todesfolge, davon entfielen 7 auf Industrie und Gewerbe und 25 auf „Sonstiges“. Sonstiges teilte sich wiederum in private Haushalte und andere Orte auf. Zur Orientierung: 2016 wurden 7 Todesfälle in Haushalten angegeben. Tendenz leicht steigend. Laut VDE sind oft Billigprodukte aus China im Spiel, deren mangelnde Qualität, zu Fehlfunktionen, Kurzschlüssen und

Sicherheitsproblemen führen können“. Der VDE vermutet sogar eine vermehrte Nutzung von Ladegeräten in Badezimmern. Nach Einschätzungen von Medizinern führen Unfälle im Niederspannungsbereich (< 1000 Volt) in 3 Prozent der Fälle zum Tod. Verkehrsunfälle führen zwar nur in rund 1 Prozent zum Tod (2022: 360.000 Unfälle, 2800 Tote). Dennoch ist das Risiko allein wegen der Größenordnung ungleich größer - ohne dass ständig jemand darauf hinweist, wie unverantwortlich Autofahren sein.

Damit wir uns nicht falsch verstehen: Es geht hier nicht darum, eine potenzielle Gefahr durch Strom zu Gunsten des Bastelns oder des Reparierens zu banalisieren. Es geht darum, das Risiko auf Basis von Zahlen richtig einzuordnen. Und wie es in der Statistik aussieht, muss die Reparatur der Waschmaschine nicht automatisch im sicheren Tod enden. Warum der Umgang oder Nichtumgang mit 230 V in unseren Foren regelmäßig mit Schaum vorm Mund endet, ist mir schleierhaft.

Aber tun Sie mir trotzdem den Gefallen, weiterhin unsere Warnhinweise in Artikeln zu beachten und einzuhalten. Und geben Sie Tipps zum sicheren Umgang gerne weiter. Ohne Schaum vorm Mund.

Happy Hacking

Daniel Bachfeld

► make-magazin.de/xemj

Sagen Sie uns Ihre Meinung!

mail@make-magazin.de

Inhalt

Workshops

3D-Pop-up-Weihnachtskarten können Sie mit unserer Anleitung ganz individuell selber basteln. Grafik geht aber auch elektronisch: Peppen Sie Ihre Programme für ESP32 und Co. doch mit einer grafischen Oberfläche auf. Und staubfrei bohren, sägen etc. können Sie künftig durch maßgeschneiderte Staubsauger-Adapter.

- 32 Pop-up-Karten selber machen
- 92 GUIs für ESP32 programmieren
- 102 Saugerkupplungen auf Maß drucken



Bausätze für Maker

Früher war nicht nur mehr Lametta, sondern die Vorweihnachtszeit auch Bastelzeit. Inzwischen findet das eher nach der Bescherung statt. Unsere Geschenkideen für Maker sind diesmal Bausätze, die alles Notwendige aus Holz, Karton oder an Elektronik enthalten, um daraus richtige funktionierende Planetenmodelle, Uhren, Dampfmaschinen, Roboter oder Murbelbahnen zu bauen.

- 22 Bausätze für Maker

- 3 Editorial
- 6 Leserforum
- 8 **Test: Raspi 5 im Kurztest**
- 12 **Report: MicroPython-Boards**
- 20 Jubiläums-Maker-Faire in Hannover
- 22 **Report: Bausätze für Maker**
- 32 **Workshop: Alles auf eine Karte**
- 42 In eigener Sache
- 44 **Projekt: DIY-TV-Beleuchtung**
- 50 **Projekt: Head-Up-Display für Skifahrer**
- 56 **Projekt: IKEA-Uhr LEDifiziert**
- 62 **Community-Projekt: Gaszähler auslesen und Daten sichern**
- 64 **Community-Projekt: Upcycling-Bartop: Arcade-Spaß mit Scratch-Spielen**
- 66 **Community-Projekt: Roboter für das Klassenzimmer**
- 70 **Projekt: Der Rossmann-Waagen-Hack**

Projekte

Leuchten wird demnächst nicht nur der Weihnachtsbaum, sondern auch Ihr Fernseher mit selbstgebauter Effekt-Beleuchtung. Aber auch eine schlichte IKEA-Uhr kann mit einem LED-Kranz zu einem echten Hingucker werden. Abnehmen nach den Feiertagen können Sie beim Skifahren mit Head-up-Display.

- 44 Mit Pi: Ambilight für TV
- 50 Head-Up-Display für Skibrille
- 56 IKEA-PUGG-Uhr aufmotzen



Waage smart machen

Die Feiertage sind gefährlich fürs Gewicht. Wir machen daher eine preiswerte Personenwaage fit fürs Smarthome mit dem Ziel, dass Sie das Erreichen ihrer gravitationsabhängigen Vorsätze (Wunschgewicht) fürs nächste Jahr besser überwachen (lassen) können. Teil 1 bindet die Waage mit neuer ESP32-Elektronik und Farb-Display in Home Assistant ein.

- 70 Der Rossmann-Waagen-Hack



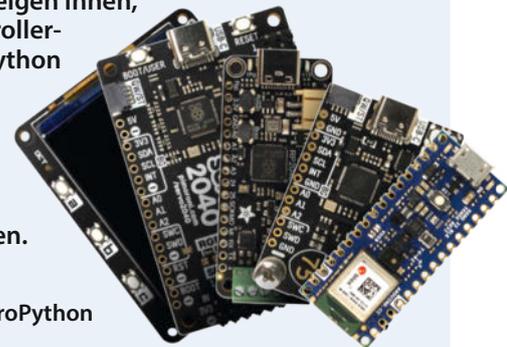
- 78 Know-how: Holzisenbahn mit 3D-Druck erweitern
- 80 Projekt: KI-Sprachassistent mit eigenem sprechenden Avatar
- 86 Projekt: DIY-Timeout-Uhr
- 90 Reingeschaut: Laminiergerät
- 92 **Workshop: UI-Design mit LVGL**
- 98 Projekt: Lötén mit einer Hand
- 102 **Projekt: Staubfrei arbeiten mit osVAC neo**
- 110 Workshop: Makros für FreeCAD in Python programmieren
- 118 Kurzvorstellungen: Recycle-Nylon-Filament, 3D-Drucker Bambulab P1S, Licht- und Entfernungssensor, LibrePCB V1.0, Mikrowellen-Bewegungsmelder, MotionGen Pro, Joysticks per USB, Raspberry-Pi-Alternative Alta A1, USB-C Power Delivery, Rockchip SBC,
- 122 Impressum/Nachgefragt

Themen von der Titelseite sind rot gesetzt.

Boards für MicroPython

Wer die leicht erlernbare Programmiersprache Python ausprobieren möchte, kommt um passende Hardware nicht herum: Wir zeigen Ihnen, welche Mikrocontroller-Boards für MikroPython geeignet sind und außerdem LEDs, Sensoren oder Displays zum Ausprobieren gleich onboard mitbringen.

- 12 Boards für MicroPython



Leserforum

Elektronikfragen

Schicker Streamer, Make Magazin 4/23, S. 48

Mechanisch wirklich erste Sahne, aber elektrisch? Bei den Netzteilen habe ich Fragen: Längsregler mit Drossel ist ein außergewöhnliches Konzept. Wurde das berechnet oder mindestens vermessen? Bei 7000 uF am Ausgang sollte über dem Regler eine Schutzdiode sein. Laut Datenblatt benötigt der Regler auf der Eingangsseite einen 10uF Kondensator nah beim Regler! Statt Folienkondensatoren sollten Keramiktypen eingesetzt werden (induktionsarm). Welchen Vorteil bringt so ein „audiophiles“ Netzteil für den Display-Controller? Da der USB-DAC laut Hersteller bereits seine Versorgung filtert, kann auch für diesen Zweig ein weniger aufwendiges Netzteil ausreichend sein? Optimal wäre es gewesen, den DAC direkt an der Rückwand zu installieren, um dessen On-board-Buchsen erreichen zu können. Ansonsten ist das Steckerkabel ausgangsseitig direkt an die Buchsen zu löten.

Otto Fürcho

Antwort des Autors: Die Drossel wurde berechnet, nicht von mir aber in einem ähnlichen Netzteillayout. Zur Schutzdiode: Ja, im Falle eines Kurzschlusses am Eingang des Reglers könnte diese ohne Diode beschädigt werden. Danke für den Tipp zu den Keramikkondensatoren, probiere ich in einem Folgeprojekt. Da der Displaycontroller über HDMI mit dem Raspberry verbunden

ist, sollte die Wahl des Netzteils zur Störungsvermeidung beitragen. Das Khadas Board bekommt seine Spannung vom Raspberry über USB. Zur Montage des DAC: Dann wäre leider das USB-Kabel aus der Rückwand heraus und wieder zum Raspberry hinein verlaufen. Das fand ich optisch nicht so gut. Ein direktes Anlöten am Khadas-Board habe ich aus Garantiegründen vermieden.

Zuviel Smarthome und IoT

Für die Ausgabe Make 4/23 muss ich euch mal ein großes Lob aussprechen. Endlich mal wieder was zum Basteln und jede Menge Informationen zu ESP und Co. Auch wenn ich mit Smarthome nicht viel anzufangen weiß, waren diese Infos mal wieder richtig erfrischend. Basteln ist eher mein Ding, daher habe ich den Galagino komplett nachgebaut und mit ein paar kleinen Modifikationen am Gehäuse sauber zum Laufen bekommen. Der zugehörige Online-Artikel zur Z80-Emulation hat mich stark beeindruckt. Da glaubt man, man hätte Ahnung von Soft- und Hardware und auch vom Programmieren, und dann beschreibt der Autor wie er mit Hilfe eines „fremden“ Z80-Emulators das ganze Mini-Arcade-Gerät entwickelt und zum Laufen bringt. Respekt. Und ein 3D-gedruckter Joystick mit Mikrotastern in einem so kleinen Gerät ist schon der Hammer. Da kam auch mein selbstgebauter Laser (MaXYposi mit 7W-LED-Laser)

mal wieder zum Einsatz. Solche Bastelbeiträge wünsche ich mir öfters, die Make ist in letzter Zeit eher zur Smarthome und IoT-Zeitschrift verkommen.

Hans-Peter Dietrich

Newsletter

Als Elektroniker-Bastler habe ich Ihre interessante Zeitschrift schon des Öfteren gekauft, wenn tolle Projekte zum Nachbauen darin waren, auch wenn Technikwissen vermittelt wurde. Nun, da bei meinem Heimatkiosk Eure Zeitschrift nicht mehr, also nur auf Bestellung erhältlich ist und ich keine „Katz im Sack“ kaufen möchte, wäre es echt super, wenn Ihr in den Newslettern die Inhaltsartikel im jeweiligen aktuellen Magazin veröffentlichen würdet sowie wann die nächste neue Ausgabe erscheint.

Othmar Berbig

Schade, dass ihr Kiosk uns aus dem Sortiment genommen hat. Welcher Kiosk in Ihrer Nähe uns noch führt, finden Sie unter www.mykiosk.de. Daneben finden Sie den Inhalt der aktuellen Hefte auch auf unseren Shop-Seiten shop.heise.de.

Klebealternative

Schweißdraht, Make 6/23, S. 19

Anstatt des „Schweißdraht aus der Flasche“ für 25€, geht auch Sekundenkleber mit Soda/Natron. Ich habe damit auch schon Muttern wie in dem Beispiel aus 6/23 oder auch andere Plastik-Gehäuseteile günstiger wieder repariert. Gibt auch mehr als genug Videos auf YouTube darüber.

Thomas Liebscher

Wenn man dem Hersteller des Superklebers glauben mag, geht das zwar irgendwie auch mit Natron, aber hält nicht so gut wie das eigene Produkt. Er hat auch ein Video dazu gemacht: <https://hgpowerglue.com/faq/>. Eine (bessere) Alternative ist laut unserem Forums-User LBOF Edelkorund in Verbindung mit Epoxidharz. Ob man damit in der Summe deutlich günstiger wegkommt als mit dem im Heft erwähnten Kleber, können wir leider nicht sagen.

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



www.facebook.com/MakeMagazinDE



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



www.twitter.com/MakeMagazinDE



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

Korrekturen

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

Geht's noch?

Titelbild, Make 6/23

Das Foto auf der Titelseite des aktuellen Make-Sonderhefts 06/23 ist ja hoffentlich als Aprilscherz gedacht. Ich würde ja nichts schreiben, wenn es sich um irgend eine hirn- und sinnlose Verwendung in einem beliebig schlechten Bericht der Bild-Zeitung handeln würde, aber in einem technisch orientierten und auch an (hoffentlich) technisch versierte Leser gewandten Bastler-(ok, „Maker“-)Magazin ein Foto zu benutzen, auf dem völlig sinnfrei auf der Bestückungsseite einer Platine mit dem LötKolben herumgestochert wird... also bitte. Wollt ihr wirklich so tief sinken?!

Robert Klenk

Das Bild war eher als Symbolbild gedacht, um Bauteile und LötKolben zusammen auf einem Bild zu bringen. Die Platine von unten abzubil-

den war uns zu langweilig. Dass wir damit Leser irritieren, war nicht beabsichtigt.

Fehlende Bezugsquellen

Tipps & Tricks, Make 6/23, S. 124

Ich suche die Bezugsquellen für den im Reparaturheft beschriebenen ATX-Power-Adapter.

Hartmut Boss

Die Bezugsquellen befinden sich normalerweise immer in den Kurz-Links eines Artikels. Leider hat die ein Fehlerteufel gefressen. Der Kurzlink lautet <http://make-magazin.de/xj2a>

Feuchte Filamente

Filamenttest, Make 5/23, S. 8

Meiner Erfahrung nach ist die Feuchtigkeit in Filament hauptsächlich schlimm, weil beim

Drucken dann bei >200°C Wasserdampfblasen entstehen, die sozusagen „Sollbruchstellen“ im Druck bilden. Extrem ist das bei Nylon aber auch PETG. Ich habe schon total sprödes PLA gedruckt, das im gedruckten Zustand wieder ganz normal war... Dazu kommt noch, dass Filament in seiner Komposition für 3D-Druck etwas anderes sein kann als das PETG, das in der Industrie zum Herstellen von Flaschen genutzt wird.

Rolf Pürckhauer

Grundsätzlich sind fast alle gängigen 3D-Druck Filamente hygroskopisch und damit feuchtigkeitsempfindlich. So auch PLA und PETG. Im Heft wurde eher von Erfahrungen gesprochen, da PETG dazu neigt, schneller Feuchtigkeit bei offener Lagerung aus der Luft zu ziehen. Dafür kann PETG jedoch auch wieder gut getrocknet werden. PLA hält oft viele Jahre, ohne Effekte von Feuchtigkeit zu zeigen.

57 Struwelkolben



In einem kleinen Werkstattzimmer,
Da stand ein LötKolben gar immer.
Mit glühendem Blick, so heiß und rot,
Von ihm erzähl ich nun in diesem Bonmot.

Kolophonium



Ein unvorsichtiger Bastler,
voller Eifer,
Träumte viel
von elektrischem Geifer.



Er griff nach dem Kolben
ohne Bedacht,
Seine Finger
schnell zur Glut gebracht.



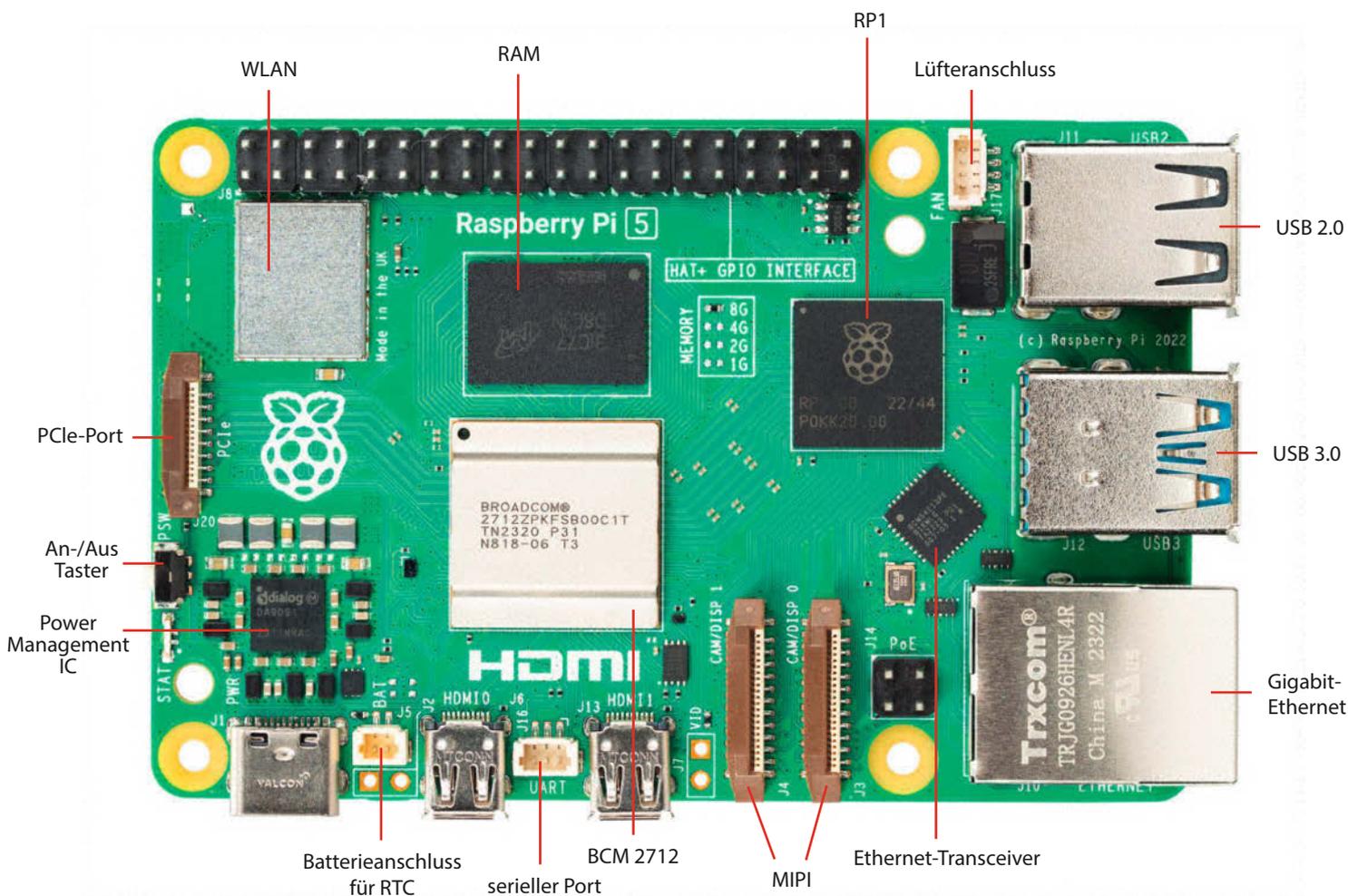
Des Bastlers G'schrei,
der Finger qualmte,
Die Hitze,
die sein Fleisch umwallte.



Der Struwelkolben zischte
höhnisch und froh,
"Lehre dir Respekt,
so ist das eben so."



Die Moral dieser Verse, wahr und klar,
Mit neuem Kolben wird Handwerk wunderbar.
Für Heim und Werkstatt, der Held am Ort,
Einhand-LötKolben, verliere nie dein Wort!



Raspi 5 im Kurztest

Die neue Version ist viel schneller als der Vorgänger und bringt neue praktische Features mit. Wir haben uns das mal angeschaut.

von Daniel Bachfeld und Heinz Behling

Auf der Maker Faire im August tat Eben Upton in unserem Interview mit Make-Kollege Ákos noch so, als wüsste er von nichts. Knapp vier Wochen später kündigte die Raspberry Pi Foundation den Pi 5 für Ende Oktober an und veröffentlichte erste Spezifikationen. Leider war dann Ende Oktober bei keinem deutschen Händler der Pi 5 offiziell verfügbar, was bis Redaktionsschluss Ende November anhält. Immerhin fand ein handverlesenes Exemplar aus dem Raspishop mit 8 GByte

den Weg in die Make-Redaktion, welches wir in der Praxis getestet haben.

Alles neu

Im Grunde ist bei der Entwicklung des Pi 5 kein Stein mehr auf dem alten geblieben, fast alles ist neu und schneller. Die vier A76-Kerne verarbeiten mehr Befehle parallel, haben größere Caches und arbeiten mit einem um 600 MHz höheren Takt. Der Videoprozessor ist um eine

Generation weiter und 500 MHz schneller. Das RAM schaufelt die Daten doppelt so schnell zur CPU. Die SD-Karte liefert über eine modernere Anbindung (SDR104 statt DDR50) ihre Bits doppelt so schnell.

Das WLAN-Modul ist vom Pi 4 übernommen, allerdings wurde die Datenverbindung auf den DDR50-Standard beschleunigt, so dass Daten per Funk doppelt so schnell fliegen. Auch USB 3.0 ist nochmal etwas flinker geworden. Summa summarum kann man

sagen, dass der Pi 5 zwei- bis dreimal schneller geworden ist als sein Vorgänger. Einen Vergleich der Daten des Pi 5 mit dem Pi 4 finden Sie in den Tabellen zu Spezifikation sowie Benchmarks.

Nützliche Sachen sind dazu gekommen: ein Ein/Aus-Taster sowie eine herausgeführte PCIe-Schnittstelle zum Anschluss etwa von NVMe-SSD und SATA. Allerdings benötigt man dafür Adapter. Offiziell ist die Schnittstelle auf dem Pi als PCIe 2.0 x1 angegeben, der Blogger Jeff Geerling hat Videos und Messungen veröffentlicht, die zeigen, dass die Schnittstelle den schnelleren Standard PCIe 3.0 unterstützt. So wie es aussieht, wird das Booten von NVMe jedoch nicht möglich sein.

Endlich hat die Pi Foundation die Krypto-Engine im SoC lizenziert, sodass nun Verschlüsselungsfunktionen in Hardware laufen. Unter anderem profitieren OpenSSL und Anwendungen wie WireGuard davon. Sogar eine Echtzeituhr (RTC) mit optionalem Akku-Anschluss und ein eigener, temperatureregelter Lüfteranschluss sind vorhanden.

Einige insbesondere für Maker interessante Funktionen fehlen weiterhin, etwa A/D- und D/A-Wandler. Für Medienwiedergabe wären auch mal ein paar andere Hardware-Decoder nett gewesen, etwa VP9 und AV-1. Es sind sogar einige Funktionen weggefallen: Der H.264-Hardware-Decoder ist nicht mehr dabei. Weil die Ethernet- und USB-Anschlüsse vertauscht sind, passen alte Gehäuse nicht mehr.

Energie

Der Pi 5 erwartet weiterhin 5V über seine USB-C-Buchse, das bisherige offizielle Netzteil mit 5,1V und 3 A für den Pi 4 funktioniert auch am Pi 5 noch. Allerdings gibt der nach dem Booten die Warnung aus: „This power supply is not capable of supplying 5A. Power to peripherals will be restricted“. Soll heißen: Solange man keine externen, stromhungrigen Geräte an USB 3.0 oder PCIe anschließt, sollte alles reibungslos funktionieren. Für mehr Power gibt es ein neues Netzteil mit 5 A (ca. 13 Euro), sodass auch zwei USB-Geräte mit je 4,5 W plus NVMe problemlos versorgt werden.

Laut Dokumentation sollten Netzteile für den Pi 5 zwar USB Power Delivery (USB-PD) beherrschen, doch offenbar nicht, um die Spannung zu erhöhen, sondern nur den Strom. Dazu müssen aber die Kabel auch ausgelegt sein, um Verluste und damit Spannungsabfälle zu verhindern. Im Moment sind die offiziellen Netzteile zwar die beste Wahl, Probleme könnte jedoch das Power-Routing auf dem Pi 5 selbst verursachen. Für die Spannungen auf dem Board ist der neue Power-Management-IC (PMIC) DA9091 verantwortlich, der speziell für den Pi 5 entwickelt wurde und diverse Spannungen für die verschiedenen Komponenten erzeugt.

Spezifikationen Raspberry Pi

| | Pi 5 | Pi 4 |
|-----------------------------|----------------------------------|---|
| SoC | BCM2712, 64 Bit, 16 nm | BCM2711, 64 Bit, 28 nm |
| CPU | 4 × Cortex A76@2,4GHz | 4 × Cortex A72@1,5GHz |
| L2-/L3-Cache | 4 × 512 KByte/2 MByte shared | 1 MByte shared/ – |
| Krypto-Beschleuniger | Ja | Nein |
| GPU | VideoCore 7, 1 GHz | VideoCore 6, 500 MHz |
| RAM | DDR4X (4266 MHz) | LPDDR4 (2400 MHz) |
| H.265 (HEVC) De-/Encoding | 4K, 60 fps/ k.A. | 4K, 60 fps/ – |
| H.264 (MP4) De-/Encoding | – / – (Software) | 1080p, 60 fps / 1080p, 30 fps |
| I/O-Chip | RP1 | für USB VIA VL805 |
| SD-Karte | MicroSD, DDR104 | Micro-SD, DDR50 |
| WLAN/BT/Anbindung | 2,4/5 GHz, BT 5.0 mit BLE, DDR50 | 2,4/5 GHz, BT 5.0 mit BLE, SDIO prop. |
| HDMI (micro) | 2 × 4K/60 Hz, HDR | 2 × 4K/60 Hz, HDR |
| PCIe 2.0 x1 | 1 Lane | - |
| USB 3.2 Gen 1 / 2.0 | 2/2 | 2/2 |
| MIPI CSI und DSI | 2 × 4 Lanes, 1,5 Gbit/s | 1 × CSI + 1 × DSI, je 2 Lanes, 1 Gbit/s |
| USB-C (Spannungsversorgung) | 5 V/5 A (27 W), USB-PD | 5 V/3 A (15 W) |

In unserem Test wurde die WLAN-Verbindung beim Abspielen eines Videos von einer USB-SSD nach etwa 30 Sekunden instabil bis hin zu völligem Aussetzen. Das Entfernen des USB-Sticks beseitigte das Problem. Vermutlich verursachen die hohen Ströme zu große Spannungsabfälle. Jeff Gerling berichtet in seinen Tests von ähnlichen WLAN-Ausfällen in Zusammenhang mit NVMe-SSDs.

Sehr praktisch ist der Power-Taster. Drückt man die Taste einmal, so erscheint auf der Bedienoberfläche das bekannte Menü zum Neustarten bzw. Herunterfahren. Zweimal gedrückt, fährt der Pi 5 automatisch herunter, egal ob gerade noch Anwendungen offen sind oder nicht. Im Headless-Server-Betrieb sollte das jedoch kein Problem darstellen. Ein erneuter Druck auf die Taste fährt den Pi 5 wieder hoch.

Neue Architektur

Während beim Pi 4 die CPU, GPU und Peripherie-Schnittstellen samt GPIO alle in einem einzigen SoC (BCM2711) stecken, sind diese Funktionen nun auf den Application Processor (AP, BCM2712 mit CPU und GPU) und den RP1 aufgeteilt.

Der RP1 (auch RP1-Southbridge wie bei PC-Mainboards genannt) ist eine Eigenentwicklung, der die Peripherie-Aufgaben des Pi 5 für GPIO, schnelles Ethernet, USB3.0, etc. übernimmt. AP und RP1 sind über PCIe verbunden. Für kommende Pi-Modelle benötigt die Foundation neben dem RP1 also nur noch etwas, was schnell rechnet und Grafik und Speicher hat. Mit der Auslagerung versetzt sich die Pi Foundation in die Lage, unabhängiger von SoC-Herstellern wie Broadcom zu werden.

Benchmarks

| | Pi 5 | Pi 5 ohne Kühlkörper | Pi 4 |
|------------------------------------|-------------|----------------------|-------------|
| mbw 100, memcopy, MiB/s | 3.872.097 | 3.299.753 | 1.784.099 |
| 7z Decomp (4 Kerne) | 12809 | 8947 | 7147 |
| 7z Comp (4 Kerne) | 9088 | 7503 | 3634 |
| OpenSSL, AES-128-cbc, 256 Byte | 1686393.34k | 1686393.34k | 49824.09k |
| iozone SD random read/write | 89901/27655 | | 45095/21656 |
| USB 3.0 Read, MByte/s | 415 | 415 | 385 |
| Python OpenCV, Blurring, ms | 31 | 40 | 61 |
| Python Tensorflow, Inferencing, ms | 44 | 50 | 148 |



Unser Pi 5 im Praxisbetrieb mit Kühlkörper. Der USB-C-Stecker des Netzteils kommt sich mit dem Micro-HDMI-Adapter ins Gehege. Besser sind spezielle Micro-HDMI-zu-HDMI-Kabel.

Womöglich designed die Foundation bald selbst eine ARM-CPU und kombiniert diese mit dem RP1. Schon mit dem RP2040 (RP2) zeigte die Foundation ihr Potenzial als Chip-Designer und Hersteller. Ob der Umbau beim Pi 5 allerdings voll kompatibel zu allen bisherigen Anwendungen ist, muss sich noch zeigen. Mangels verkaufter Exemplare fehlen Erfahrungen in der Breite.



Das neue Netzteil liefert bei 5,1 V 5 A. Das unterstützt nicht jedes PD-fähige Netzteil. Also Augen auf beim Netzteilkauf!

Fehlertolerant

Der RP1 wickelt neben USB und Ethernet auch die Steuerung der beiden MIPI-Schnittstellen ab, an die sich jeweils sowohl Kameras (CSI) als auch Displays anschließen lassen. Die GPIOs sind neuerdings „fault-tolerant“, was soviel heißen soll, dass sie auch Spannungen jenseits von 3,3 V überleben. In der ersten Version des RP1-Datenblatt stand noch explizit „5V-tolerant“, warum der Passus geändert wurde, ist unklar. Immerhin macht es den Umgang mit den GPIOs und externen Break-out-Boards und Sensoren, die mit 5 V Ausgangspegel arbeiten, weniger gefährlich. Die Anordnung und Funktion der GPIO-Leiste selbst bleibt unverändert.

Der RP1 bringt laut Datenblatt sogar je vier A/D- und D/A-Wandler mit, die sind jedoch nicht auf die Pin-Leiste geführt. Vielleicht werden ja auf einem Pi 6 diese Funktionen mal verfügbar sein.

Software

Für den Pi 5 hat die Pi Foundation extra das Pi OS auf die Debian-Version Bookworm für 32 Bit und 64 Bit aktualisiert. Im Großen und Ganzen funktioniert alles wie es soll. Einzig die H.265-Unterstützung in Hardware scheint noch nicht implementiert zu sein, dazu gleich mehr. Beim Durchführen unserer Python-Benchmarks mussten wir neuerdings in virtuellen Umgebungen (venv) arbeiten, weil aktuelle Debian-basierte Distributionen die Installation von Python-Modulen aus Fremddatenquellen (sprich per pip) nicht mehr ohne Weiteres zulassen. Gleiches gilt für Ubuntu 23.10, das es ebenfalls als Desktop- und Server-Version für den Pi 5 gibt.

Anwendungen

Für unseren Test wollten wir uns anschauen, ob auch die 3D-Druck-Software Klipper profitiert, die beim Raspberry Pi 4 schon an Grenzen stößt: Nutzt man den input shaper zur Verbesserung der Druckqualität, die GCode-Vorschau und das Video-Streaming gleichzeitig, kommt der Pi 4 ins Stolpern. Meist fällt zunächst die Video-Übertragung aus. An den Betrieb mehrerer Drucker an einem Raspi unter voller Feature-Nutzung ist da gar nicht zu denken.

Wir haben daher auch versucht, den 5er unter Klipper zu betreiben. Allerdings sind selbst die zur Verfügung stehenden Entwickler-Images noch so unausgereift, dass eine sichere Inbetriebnahme nicht möglich war. Daher muss man sich noch etwas gedulden, bis der 5er auch bei den Klipper-Entwicklern in ausreichender Stückzahl vorhanden ist und sie Gelegenheit haben, die notwendigen Anpassungen vorzunehmen.

Kodi

Da der Pi häufig als Mediaplayer genutzt wird, haben wir uns die speziell für den Raspi 5 programmierte Kodi-Version aus dem neuesten Entwickler-Image der Mediaplayer-Distribution Librelec angeschaut (LibreELEC-RPi5.aarch64-12.0-nightly-20231114-d7616c5). Installiert wurde es auf einer per USB3.0 angeschlossenen SSD, auf der sich auch die Testvideos befanden.

Die Wiedergabe von HD-Videos (1920 × 1080 Pixel in H264, H265, VP9 und AV1) war tadellos und ruckelfrei, gleichgültig, ob die Ausgabe auf einem HD- oder 4K-Monitor in Vollbild erfolgte. Die Framerates erreichten 30 bzw. 60 Hz. Auch die Tonwiedergabe war einwandfrei. Anders sah es aus bei 4K-Videos: In VP9, H.264 und H.265 kodierte Filme ruckelten selbst, wenn die Ausgabe nur in HD-Auflösung erfolgte. Forderte man die volle Auflösung solcher Videos auf einem 4K-Monitor, waren nur noch Framerates unter 20 Hz erreichbar. Das Ruckeln war entsprechend deutlich bemerkbar. Bei besonders dynamischen Bildinhalten ging die Framerate sogar noch weiter runter bis auf Werte knapp über 10 Hz. Wie oben schon erwähnt, fehlen offenbar noch Treiber für H.265.

Da der Pi per Software dekodieren musste, fing er schnell an zu kochen. Ohne Kühlkörper erreichte der Raspi 5 bereits im Leerlauf schon Temperaturen von 52 bis 54 °C. Bereits kurz nach dem Start einer Video-Wiedergabe schnellte die Temperatur auf 85 °C hoch und die Taktfrequenz des Prozessors wurde sicherheitshalber gedrosselt. Mit dem Original-Lüfter (Preis ca. 7 Euro) blieb die Temperatur deutlich unter 60 °C, der Prozessor kann ungebremst arbeiten.

Fazit

Der Pi wurde ursprünglich als günstiger Computer für Schulen entwickelt, um wieder mehr Kinder ans Programmieren heranzuführen. Dafür eignet sich der Pi 5 mit Pi OS auch weiterhin. Aus der Community kommt allerdings Kritik, ob die Foundation mit den Preisen noch auf dem richtigen Weg befindet. Rund 70 Euro für die 4-GB-Byte-RAM-Version und 95 Euro für die 8-GB-Byte-Version sind schon ein Brett und zusätzlich Zubehör von NUCs gar nicht mehr so weit entfernt. In unserem Kasten Pi 5 oder NUC erklären wir, ob es sich noch lohnt.

Durch den Performanceschub des neuen Pi profitieren anspruchsvollere Programmierprojekte, etwa KI- und Robotikanwendungen, insbesondere wenn sie in Python programmiert sind. Wer nicht so viel Wumms benötigt oder weniger Geld ausgeben will, der kann auch mit dem Pi 4 immer noch prima arbeiten und spannende Projekte umsetzen. Als Mediaplayer mit Kodi reicht auch der Pi 4 noch dicke. —dab

Raspberry Pi 5 oder NUC?

Der neue Raspberry nähert sich leistungsmäßig den kleinsten PC-Modellen, den sogenannten NUCs (Definition von Intel: **N**ext **U**nit of **C**omputing). Diese Winzlinge wurden ursprünglich von Intel entwickelt, waren mit einem Mainboard-Maß von 10 × 10 cm für damalige Verhältnisse sehr klein, besaßen aber dennoch einen recht leistungsfähigen Intel-Prozessor, großen Arbeitsspeicher, eine recht leistungsfähige Grafikeinheit on Board und Festplatten-Speicher. Inzwischen gibt es NUCs auch von zahlreichen anderen Herstellern, was zu einem deutlichen Preisabfall führte. Für unter 200 Euro kann man heute schon einen kaufen.

NUCs waren und sind voll Windows-taugliche Geräte, können aber auch andere Betriebssysteme verwenden. So gibt es beispielsweise vom Smarthome-Server-BS Home Assistant auch eine für NUCs geeignete Version.

Der Raspi 5 nähert sich nun leistungsmäßig den NUCs an, kann aber bei der Prozessor-Leistung sicher nicht ganz mithalten. Auch preislich liegen NUCs und Raspi 5 bei ähnlicher Ausstattung nicht allzu weit auseinander. Daher stellt sich die Frage, zu welchem Gerät man greifen soll.

Beide Geräte habe jeweils Vor- und Nachteile: Bei der Grafik sind sie in etwa gleich aufgestellt. Aber in anderen Punkten gibt es mehr oder weniger deutliche Unterschiede. So liegt der Raspberry beim Leistungsbedarf etwas tiefer, was sich bei Geräten im 24-h-Dauerbetrieb bemerkbar machen kann. Außerdem besitzt er frei programmierbare IO-Ports, mit denen ein NUC nicht dienen kann, es sei denn, man rüstet eine entsprechende USB-IO-Lösung nach, für die es dann auch noch zum Betriebssystem passende Software geben muss. Das ist sicher für einen „Bastelcomputer“ ein großer Nachteil; bei einem Smarthome-Server, der nur via Netzwerk mit Sensoren, Aktoren und allen anderen Geräten kommuniziert, stellt das jedoch kein Problem dar.

Ausstattungsvergleich Raspberry Pi 5/Geekom Air 11

| | Raspberry Pi 5 | Geekom Air 11 |
|---|---|--|
| Prozessor/Taktfrequenz | BCM2712 (4 × ARM Cortex A76)/ 2,4 GHz | Intel Celeron N5095 /bis 2,9 GHz |
| Arbeitsspeicher | 4 oder 8 GB | 8 GB, erweiterbar auf 32 GB |
| Massenspeicher | Mikro-SD-Karte (nicht mitgeliefert) | M.2-SSD 256 GB (max. 1 TB) |
| Grafik | 4K/2 × Mikro-HDMI/ 2 × LCD-Port | 4K/1 × HDMI/ 1 × Mini-Displayport |
| Ports/Netzwerk | 2 × USB 2, 2 × USB 3, 1 × USB-C/1-GB-Ethernet, WiFi 802.11ac, Bluetooth 5 | 3 × USB 3, 2 × USB-C, 1 × 1-GB-Ethernet, WiFi 5, Bluetooth 4.2 |
| frei programmierbare I/O-Anschlüsse | 26 | – |
| Leistungsaufnahme Leerlauf/ Vollast | 3,1 W/15 W (mit USB-Geräten bis 27 W) | 5,1 W/20,3 W |
| mitgeliefertes Zubehör | – | Arbeitsspeicher, SSD, Gehäuse, Netzteil, Lüfter/Windows 11 |
| Preis | 93 € | 179 € |
| zum Betrieb noch erforderliches Zubehör/Preis | Netzteil, Massenspeicher, Lüfter, Gehäuse, Betriebssystem/ab 55 € | GPIO-USB-Karte/14 € |
| Gesamtpreis für betriebsbereites Gerät (8 GB RAM/ 256 GB Massenspeicher) | ca. 150 € | 193 € |

Der hier zum Vergleich benutzte NUC (siehe Tabelle) wird von Haus aus mit einer 256-GB-SSD geliefert. Auch um Kühlung, Gehäuse und Netzteil muss man sich keine Sorgen machen, da alles dabei ist. Der Arbeitsspeicher sitzt zudem auf einem Steckmodul und kann durch Ersatz auf bis zu 32 GB gebracht werden.

Im Moment kann keine klare Empfehlung für den einen oder anderen gegeben werden. NUCs sind im Grunde Büro-PCs, die man mit geringem Aufwand auch zum

Basteln verwenden kann. Umgekehrt hat der Raspi 5 genug Potenzial, um auch als einfacher Bürocomputer seinen Dienst zu verrichten. Sobald er in größerer Stückzahl verfügbar sein wird und sich Software-Entwickler wirklich damit befassen können, wird sich herausstellen, was er wirklich gut kann. Besonders das Gebiet der künstlichen Intelligenz dürfte da interessant sein. Das ist auch sehr stark von der jeweiligen konkreten Anwendung und vor allem auch von der Preisentwicklung beim Raspi 5 abhängig.



GEEKOM MiniAir 11 Mini-PC
Intel Celeron der 11. Generation, Windows 11 Pro vorinstalliert

★★★★★ 514 Bewertungen

299,00€ **179,00€**

- Windows 11 Pro vorinstalliert und aktiviert
- Ultradünn und platzsparend
- Intel® Celeron® Prozessor bietet außergewöhnliche Leistung und Wert
- Intel® UHD Graphics unterstützt zwei 4K-Displays

GRÖSSE

N5095 8GB RAM+256GB SSD

AUSWAHL ZURÜCKSETZEN

Kostenloser Standardversand aus dem EU-Lager

Lieferzeit: Versand innerhalb von 1-2 Arbeitstagen

MicroPython-Boards

Wie startet man am besten mit MicroPython, wenn man Projekte umsetzen will, die leuchten, Geräusche machen oder ein Display benötigen? Zum Beispiel mit speziellen Boards, die genau für solche Szenarien konzipiert sind. Wir haben ein paar davon ausprobiert.

von Ákos Fodor



Die Programmiersprache Python erfreut sich großer Beliebtheit, denn sie ist für Anfänger leicht zu erlernen. Gleichzeitig kann man es für physikalische Simulationen oder Machine-Learning-Projekte wie Stable Diffusion nutzen. Seit 2014 gibt es Python als MicroPython auch für Mikrocontroller. Diese Implementierung ermöglicht es, schnell ins Physical Computing einzusteigen und in vereinfachter Form spielerische, experimentelle und praktische Projekte umzusetzen.

Zu den offiziellen Pyboards, mit denen das MicroPython-Projekt ursprünglich startete, haben sich über die Jahre viele weitere kompatible Mikrocontroller-Boards gesellt. Und da die Pyboards gerade kaum erhältlich sind, habe ich die Gelegenheit genutzt und mich nach interessanten Alternativen umgeschaut. Dabei haben es mir hauptsächlich spezielle Varianten angetan, die wesentlich mehr mitbringen als ihre reine Rechenleistung. Falls ihr also noch keine Gelegenheit hattet, in MicroPython reinschnuppern, ist jetzt vielleicht ein guter Zeitpunkt. In der Kurzinformatio findet ihr Links zu allen getesteten Boards.

Minimales Set-up

Grundsätzlich kann man MicroPython-Projekte mit Entwickler-Boards starten, auf denen sich im Wesentlichen (neben Spannungsreglern) nur ein Mikrocontroller befindet, dessen GPIO-Pins auf die Platine hinausgeführt werden. Von MicroPython unterstützte Chips sind z.B. der RP2040 der Raspberry Pi Foundation, der ESP8266 und diverse ESP32-Varianten von Espressif, SAMD-Chips von Microchip, STM32-Chips und viele mehr. Diese sitzen wiederum auf den vielen Boards von Adafruit, Arduino, Seeed Studio und Co. Das bekannteste Boards mit dem RP2040 ist wohl der Raspberry Pi Pico bzw. Pico W (mit WLAN).

Kurzinformatio

- » Spezielle MicroPython-Boards für den leichten Einstieg in Projekte
- » Displays, Servos und NeoPixel steuern
- » Mit Sensoren experimentieren

Mehr zum Thema

- » Helen Leigh Steer, Python auf Hardware, Make 6/20, S. 90
- » Stefan Draeger, Whack-A-Mole mit dem Raspberry Pico, Make 6/22, S. 56
- » Carsten Wartmann, Pybricks: Python steuert Lego-Roboter, Make 3/23, S. 78



Einfacher mit Spezial-Boards

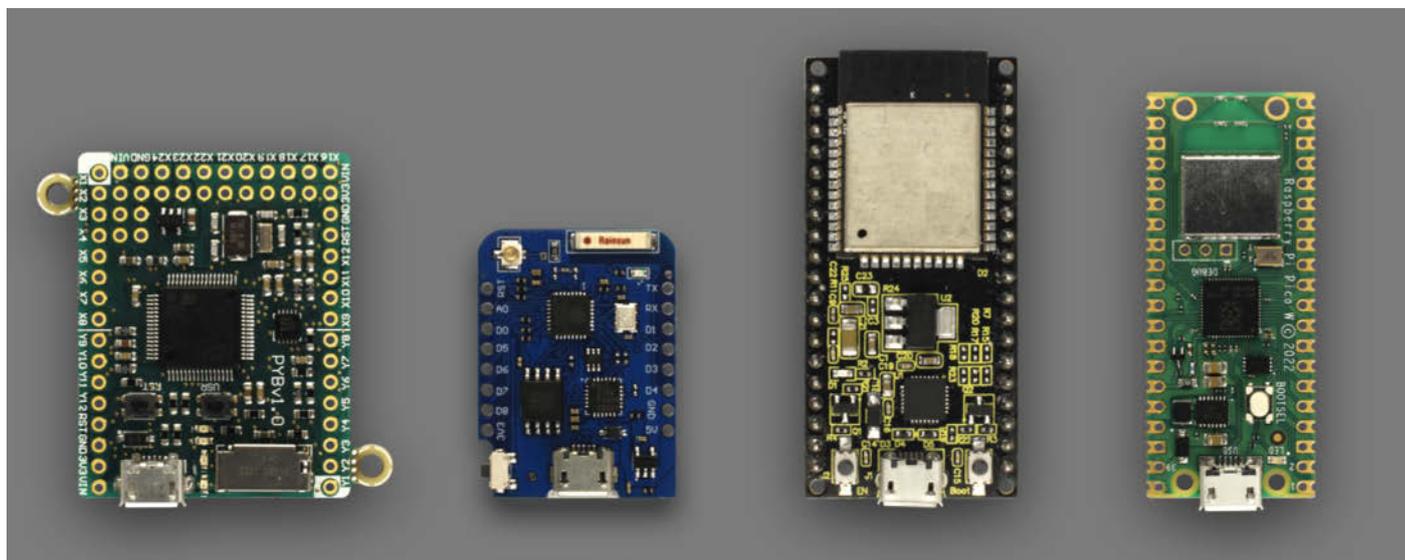
Für manche Vorhaben kann ein solches Entwickler-Board völlig ausreichen, z.B. wenn man MicroPython von Grund auf lernen möchte oder lediglich mit anderen Geräten kommunizieren will. Hier hilft sicher auch ein Blick in unser Make MicroPython Special mit vielen Tipps und Tricks (siehe Link in Kurzinformatio). Soll der Mikrocontroller aber Teil eines Projekts sein, das sich bewegt, leuchtet, Geräusche macht oder mit dem man haptisch interagieren kann, kommen schnell weitere Komponenten wie Displays, Sensoren, Lautsprecher oder Joysticks ins Spiel. Diese muss man aber erst richtig verkabeln und mit den passenden Bibliotheken zum Laufen bekommen. Für manche ist das ein Kinderspiel, für andere ein Grund, Projekte auf die lange Bank zu schieben – was dazu führen kann, dass sich über die Jahre viele tolle Elektronikbauteile

ansammeln, aber es fehlt dann doch immer die Zeit, sich mit ihnen auseinanderzusetzen.

Hier kommen spezielle Boards ins Spiel, die bereits mit zahlreichen Komponenten ausgestattet und vorkonfiguriert sind, sodass man sich direkt in spannende Projekte stürzen kann. Ob das wirklich immer so einfach ist, habe ich mit ein paar Modellen für euch ausprobiert.

Pimoroni Tufty 2040

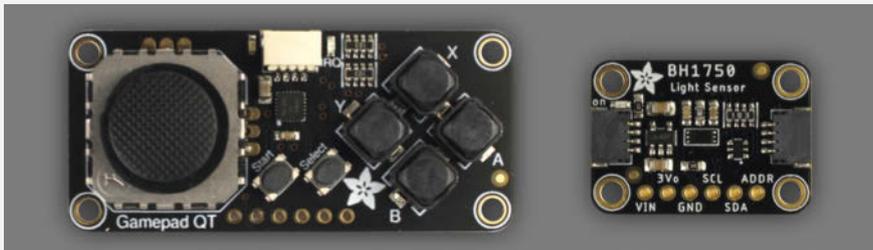
Wer lernen möchte, wie man einen kleinen Bildschirm mit MicroPython programmiert, findet einen schnellen Einstieg mit dem Tufty 2040 von Pimoroni. Ursprünglich als digitales Namensschild (engl. badge) für Veranstaltungen entwickelt, vereint das Board einen RP2040 mit einem gestochenen scharfen 2,4-Zoll-IPS-Display mit 320 x 240 Pixeln. Damit man es auch unterwegs nutzen kann, lässt es sich



MicroPython unterstützt mittlerweile viele gängige Entwicklerboards. Hier von links: Pyboard V1, WeMos D1 mini Pro, ESP32-WROOM, Raspberry Pi Pico.

CircuitPython

Wenn man mit MicroPython kompatible Boards sucht, stolpert man recht schnell auch über CircuitPython von Adafruit. Dabei handelt es sich um eine Abzweigung (engl. fork) von MicroPython, die Adafruit für sein eigenes Ökosystem entwickelt und mit ein paar Änderungen versehen hat, die das Programmiererlebnis noch weiter vereinfachen sollen. Dazu gehört etwa, dass ein Mikrocontroller, auf dem CircuitPython läuft, als Laufwerk auf dem Computer-Desktop erscheint – sofern er eine native USB-Schnittstelle besitzt (z.B. RP2040 oder ESP32-S3). Dadurch kann man Programme in einem einfachen Texteditor erstellen und direkt auf das Laufwerk kopieren. Außerdem ist standardmäßig ein Soft-Reboot aktiviert, mit dessen Hilfe sich das Board jedes Mal neu startet, sobald man sein Programm in der IDE (engl. Integrated Development Environment) speichert. Für diese Funktion sollte man den empfohlenen Mu Editor verwenden.



Adafruit-STEMMA-Module wie dieses Gamepad oder der BH1750-Lichtsensor lassen sich mit CircuitPython einfacher in Betrieb nehmen als mit MicroPython.

Ein weiteres großes Feature, das CircuitPython besonders für Anfänger interessant macht, sind die zahlreichen Bibliotheken, die Adafruit für sein Ökosystem bereitstellt. Neben Mikrocontrollern gehören nämlich auch etliche Bildschirme, Sensoren, LED-Streifen und STEMMA-Komponenten zum Sortiment. Da Adafruit in der Regel auch eine CircuitPython-Bibliothek für diese anbietet, kann man sie sehr

unkompliziert mit einem CircuitPython kompatiblen Board in Betrieb nehmen.

Der einzige Nachteil: Da die Bibliotheken auf möglichst vielen Adafruit-Boards funktionieren sollen, laufen manche Lösungen in CircuitPython nicht ganz so schnell oder effizient wie mit einem speziell auf einen Mikrocontroller angepassten MicroPython (siehe Pimoroni Servo 2040 und Interstate 75).

zudem über einen JST-PH-Stecker auf der Rückseite mit einer Batterie oder einem 3,7-V-Akku (LiPo/Lilon) betreiben. Da der Tufty keine Schutzschaltung hat, sollte man einen Akku wählen, der selbst eine hat. Außerdem muss man ihn separat aufladen oder man verbindet den Tufty mit einem kleinen Breakout-Board wie dem Lipo Amigo Pro.

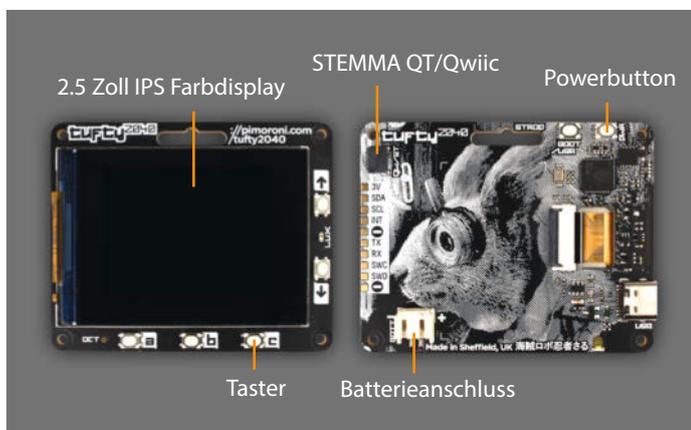
Auf der Platine befinden sich neben dem Display noch fünf Taster auf der Vorderseite, die man frei mit Funktionen belegen kann, sowie eine Boot- und eine Power-Taste auf der Rückseite, mit der sich Tufty ausschalten lässt, wenn man etwas Strom sparen möchte. Auch

einen Sensor hat Pimoroni in Form eines Fotowiderstandes verlötet, mit dem man z.B. die Hintergrundbeleuchtung des Displays über PWM regeln kann. Für STEMMA-QT/Qwiic-Module hat das Board ebenfalls einen Steckplatz.

Der Tufty 2040 kommt bereits mit einem vorinstallierten MicroPython und weil es sich anbietet, hat Pimoroni gleich ein paar Beispiele auf den 8 MB (vgl. 2 MB beim Raspberry Pico) großen Flash-Speicher geladen. So kann man gleich nach dem Auspacken sehen, was das kleine Gerät auf dem Kasten hat. Nicht erschrecken: Das Tufty-Board startet erst, nachdem man die Power-Taste gedrückt hat. Daraufhin

erscheint ein animiertes, farbenfrohes Menü, das auf dem scharfen Display wirklich eine gute Figur macht. Mit den Tastern kann man sich eine der verschiedenen Demos auswählen und starten. Darunter befinden sich Namensschilder, animierte Schriftzüge und sogar ein kleines Spiel. Um zum Menü zurückzugelangen, muss man den Tufty neu starten.

Sobald man den Tufty mit einer MicroPython IDE (z.B. Thonny) verbindet, kann man die gespeicherten Beispiel-Codes studieren und anpassen – oder man baut sich selbst aus Einzelteilen etwas zusammen. Sofern man den Flash-Speicher nicht löscht, erscheinen neue



Das tragbare Pimoroni Tufty 2040 hat nicht nur einen schönen Bildschirm, sondern auch ein paar Taster und einen Batterie/Akku-Anschluss.



Ein Bild als Badge ist schnell erstellt, nur muss man die Bildqualität beim Export etwas senken, damit es in den verfügbaren RAM passt.

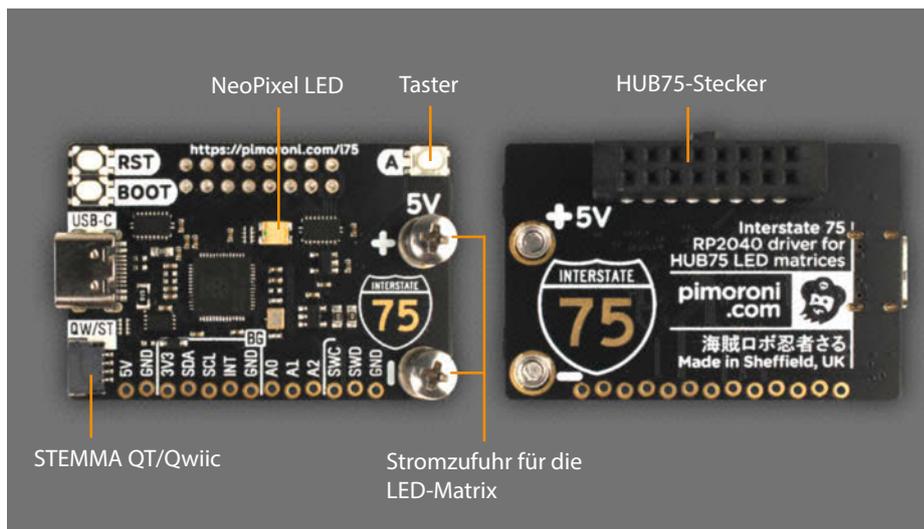
Programme nach dem Upload automatisch auch in dem animierten Menü. Um den Tufty kennenzulernen, bietet Pimoroni Anleitungen auf seiner Website an und auf GitHub gibt es auch ein paar weitere Beispiel-Codes, z.B. um zu demonstrieren, wie sich die Bildschirmhelligkeit mithilfe des Fotowiderstandes regeln lässt.

Um auszuprobieren, wie leicht ich mir ein eigenes Badge erstellen kann, habe ich am Computer eine JPEG-Datei erstellt, eines der vorhandenen Beispiele dupliziert und so weit gekürzt, bis es nur noch das Bild anzeigen sollte. Allerdings passierte das nicht. Wie ich recht schnell herausfand, war das Bild zu groß für den 264 kB großen RAM des RP2040, der zum Teil bereits damit beschäftigt war, MicroPython auszuführen. Also habe ich die Datei mit einem verschmerzbaaren Qualitätsverlust verkleinert und erneut hochgeladen. Vielleicht baue ich noch einmal eine Version mit Grafikelementen und füge nur JPEG-Inhalte ein, wo es unbedingt notwendig ist.

Am Tufty 2040 gefällt mir besonders gut, dass man direkt in ein funktionierendes System einsteigt und viele Codes mitgeliefert bekommt, mit denen man gleich losspielen kann. Abgesehen davon finde ich die Displayqualität überzeugend. Einzig die CircuitPython-Bibliothek für meinen Adafruit-Gamecontroller, den ich über den STEMMA-QT/Qwiic-Anschluss verwenden wollte, konnte ich nicht zum Laufen bringen.

Pimoroni Interstate 75

Habt ihr vielleicht auch schon länger eine LED-Matrix in der Schublade liegen, aber bisher nicht verwendet, weil der richtige Mikrocontroller oder die Zeit fehlte? Sofern diese LED-Matrix einen Hub75-Anschluss besitzt, hat Pimoroni das perfekte Board für euch, um



Reinstecken, fertig, los: Mit dem Pimoroni Interstate 75 kann man LED-Matrizes ganz leicht ansteuern.

z.B. schnell eine Laufschrift zu bauen: den Interstate 75. Dieses Board kann eine einzelne RGB-Matrix mit bis zu 64 x 64 Pixeln oder bis zu vier verbundene LED-Matrizes steuern, die nebeneinander aufgereiht sind. Dafür nutzt Pimoroni die PIO-Funktion (Programmable Input Output) des RP2040, der sich ebenfalls auf dem Board befindet. Über PIO kann der Mikrocontroller unabhängig vom Hauptprozessor mit neun Assembler-Befehlen blitzschnell Programme ausführen und in diesem Fall die Daten für die LED-Matrizes verarbeiten. Praktischerweise muss man sich damit aber gar nicht beschäftigen, sondern lediglich den Interstate 75 über den Stecker auf der Rückseite in den Anschluss auf der LED-Matrix stecken. Danach bringt man die Stromkabel, die mit der LED-Matrix mitgeliefert werden, an den Schrauben auf dem Interstate 75 an und steckt das andere Ende in die LED-Matrix.

Wer möchte, kann über die STEMMA-QT/Qwiic-Buchse noch einen Sensor oder einen Poti anschließen, um etwa die Helligkeit der LEDs darüber zu steuern. Danach verbindet man den Interstate 75 mit einem USB-C-Netzteil und startet die MicroPython-IDE seiner Wahl (z.B. Thonny), um ein paar Beispiel-Codes auf das Board zu laden. Das dauert im besten Fall keine fünf Minuten.

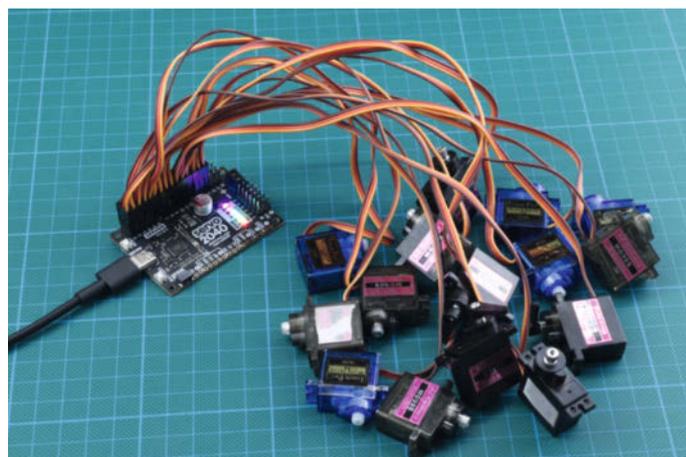
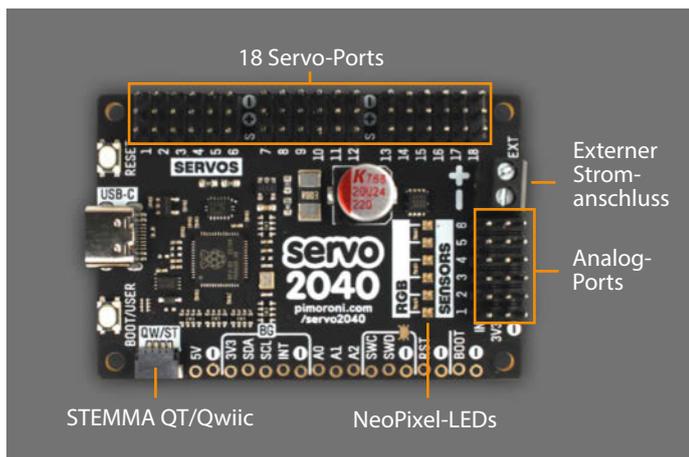
Pimoroni gibt an, dass die Stromversorgung über USB-C mit 5 V und maximal 3 A für eine LED-Matrix bis 64 x 64 LEDs ausreicht, sofern man nicht alle LEDs bei voller Helligkeit in Weiß leuchten lässt. Auch wenn dieses Szenario unwahrscheinlich ist, würde die Waveshare P3-64x64-LED-Matrix, die ich mit dem Interstate 75 ausprobiert habe, dafür 4 A benötigen. Bei besonders leuchtstarken Projekten oder spätestens, wenn man mehrere Matrizes miteinander verschaltet, sollte man



Strom erhält die LED-Matrix über die zwei Schraubklemmen.



In dieser animierten Demo prallen die Kreise von den Wänden ab.



An dem Pimoroni Servo 2040 lassen sich bis zu 18 Servos anschließen.

Jetzt muss man sich nur noch überlegen, was man mit den vielen Servos baut.

an den Schrauben am Interstate 75 eine zusätzliche externe 5-V-Stromquelle anschließen, um sicherzugehen, dass das Board keinen Schaden nimmt. Die Beispiel-Codes (springende Bälle und ein Schriftzug), die Pimoroni auf seiner Website anbietet, kann man ohne Bedenken über ein USB-C-Netzteil abspielen.

Den Pimoroni Interstate 75 gibt es übrigens auch als etwas größere WLAN-Variante mit einem aufgesteckten Raspberry Pico W. Dadurch könnte man eine Laufschrift z.B. über WebREPL programmieren.

Pimoroni Servo 2040

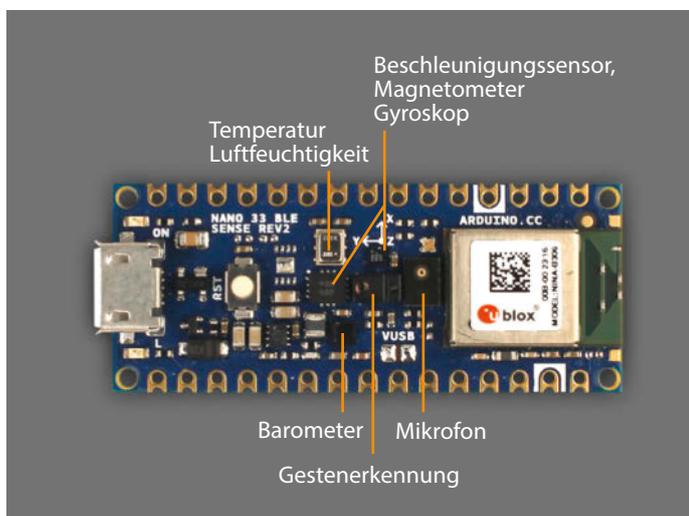
Wer Roboter oder Animatronics mit vielen Servos bauen will, verbindet diese in der Regel nicht direkt mit einem Mikrocontroller, sondern nutzt dafür einen zusätzlichen Servo-Controller, z.B. den PCA9685. Dieser koordiniert den reibungslosen Ablauf der vielen Servos und versorgt sie gleichzeitig mit Strom.

Pimoroni hat zu diesem Zweck ein Board entwickelt, das beides kann und dafür lediglich einen RP2040 benötigt: den Servo 2040. An ihn lassen sich insgesamt 18 Servos anschließen. Wie auch beim Interstate 75 verwendet Pimoroni für die Hintergrundsteuerung die PIOs des RP2040. Das Board lässt sich optional auch mit CircuitPython betreiben, allerdings lassen sich dann nur 16 Servos anschließen, da die PIOs nicht verwendet werden. Über einen analogen Pin-Header an der Seite kann man zusätzliche Sensoren in sein Projekt einbinden. Neben diesem Header sind sechs Neo-Pixel-LEDs verlötet, die man z.B. dafür nutzen kann, mithilfe von Farbcodes Feedback zu geben, wenn gerade kein Computer angeschlossen ist. Außerdem hat auch dieses Board einen Anschluss für STEMMA QT/Qwiic.

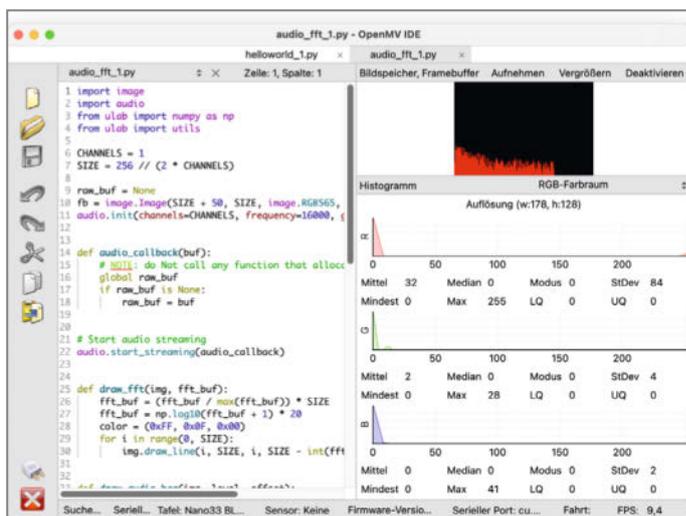
Sowohl das Board als auch die Servos lassen sich über den USB-C-Anschluss mit 5V und maximal 3 A betreiben. Außerdem kann man eine externe Stromquelle über einen Schraub-

klemmblock anschließen. Um herauszufinden, ob das überhaupt nötig ist, kann man sich über REPL ausgeben lassen, wieviel Strom das Board für jeden einzelnen Servo misst. Bei einem MG90s sind es im Schnitt 100 mA ohne größeren Widerstand, kann aber je nach Konstruktion und Belastung variieren. Erhöht sich dieser, benötigt man etwas mehr. Das hängt stark von dem jeweiligen Anwendungsfall ab. Insofern ist es praktisch, dass man diese Werte direkt vom Board abfragen kann. Benötigt man mehr als 5 V für seine Servos, sollte man unbedingt auf der Rückseite des Boards eine Lötbrücke durchtrennen, mit der man die Stromversorgung zum RP2040 und den übrigen empfindlichen Komponenten kappt.

Das Board besitzt neben den NeoPixel-LEDs keine dedizierte Status-LED, was man spätestens merkt, wenn man den Servo 2040 das erste Mal mit dem Computer verbindet. Das hat mich erst mal irritiert. Nachdem ich aber die Firmware geflasht und den Beispiel-



Der Arduino Nano 33 BLE Sense ist mit allerhand Sensoren ausgestattet.



In der OpenMV IDE findet man fertige Beispiele für die jeweiligen Sensoren.

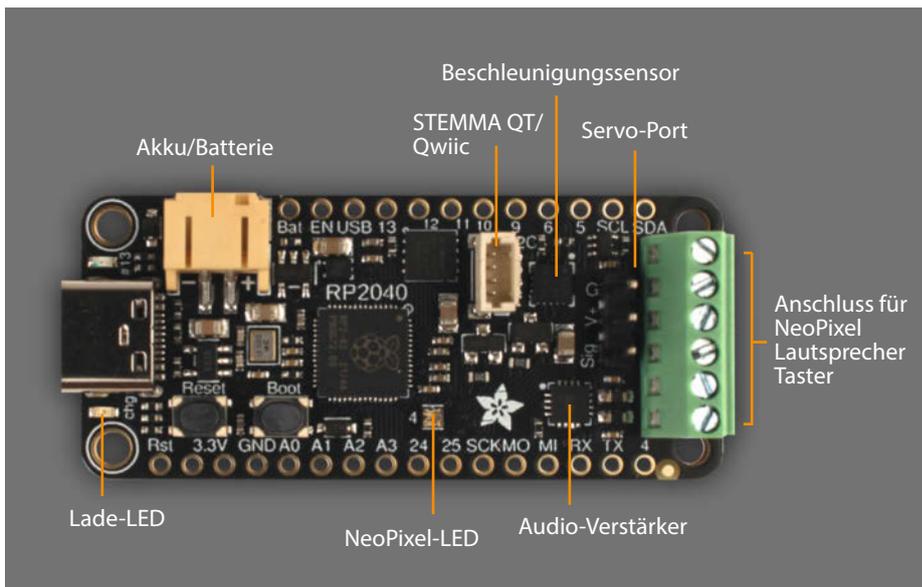
Code led_rainbow.py aufgespielt hatte, erstrahlten die LEDs in einem animierten Regenbogen.

Der eigentliche Star sind aber natürlich die Servos und für deren Programmierung liefert Pimoroni sowohl eine umfangreiche Dokumentation, als auch genug Beispiele, durch die man sich mithilfe der Kommentare hangeln kann. Servos lassen sich über Werte (z.B. Winkel), prozentual oder PWM steuern. Ich habe MG90s- und MG996R-Servos mit dem Board ausprobiert und bis auf einen, der in den äußeren Winkeln ein wenig ächzte, bewegten sich alle wie gewünscht. In solchen Fällen kann man Servokanäle separat kalibrieren und z.B. die äußeren Winkel begrenzen.

Das Einzige, was ich mir bei diesem Board noch gewünscht hätte, wäre eine WLAN- oder Bluetooth-Schnittstelle, damit man sein Projekt z.B. über einen Webserver fernsteuern oder mit dem Internet verbinden kann.

Arduino Nano BLE 33 Sense (Rev2)

Dieses Board im Arduino-Nano-Format läuft mit einem nRF52840-Chip von NXP und ist mit einer Vielzahl von Sensoren gespickt. Das



Mit dem Adafruit Prop-Maker Feather kann man Projekte umsetzen die leuchten, Geräusche machen und auf Bewegung reagieren.

Board wird beispielhaft in vielen Büchern rund um die Frameworks TinyML und TensorFlow Lite eingesetzt. Man kann mit den Messwerten des Nano smarte Geräte entwickeln, die auf

Stimmen, Gesten oder andere Bewegungen reagieren. Die IMU-Einheit (engl. Inertial Measurement Unit) LSM9DS1 vereint einen Beschleunigungsmesser, ein Magnetometer



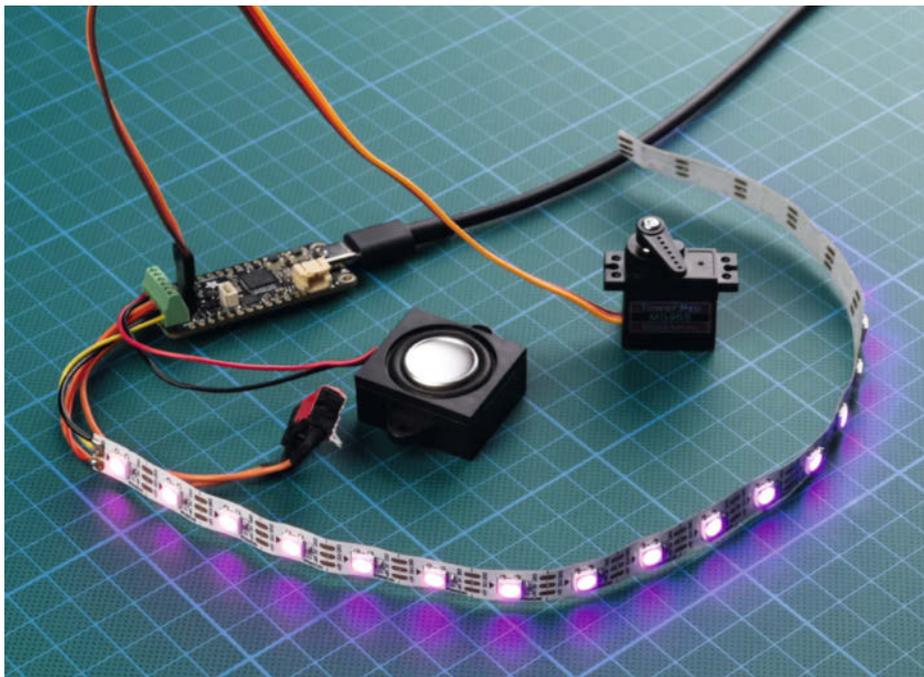


Bastle die Zukunft nach deinen Visionen.

DO WHAT YOU LOVE.

ifmjobs.de

Vom Tüfteln zum Zukunftsgestalten: Entwickle und implementiere mit deinen Hardwarelösungen die DNA unserer Sensoren und Geräte. Arbeite am Puls der industriellen Evolution – bei ifm. **Bewirb dich als Hardwareentwickler:in!**



Erst wenn der (hier abgeklebte) Taster gedrückt ist, geht die Party los. Fehlt nur noch ein passender Akku.

und ein Gyroskop und kann dadurch die relative Bewegung und Ausrichtung des Nano bestimmen. Weiter erkennt der APDS9960-Sensor, wenn sich etwas vor dem Nano bewegt oder nähert. Das LPS22HB-Barometer misst den Luftdruck und der HS3003-Sensor die Temperatur sowie die Luftfeuchtigkeit. Außerdem verfügt der Nano über ein Mikrofon und lässt sich per Bluetooth mit einem Smartphone verbinden, um die Sensordaten auszuwerten oder mit ihnen zu interagieren.

Wenn man den Nano 33 BLE Sense in MicroPython programmieren will, empfiehlt Arduino den Fork OpenMV, der eigentlich für Machine-Vision entwickelt wurde, aber alle wichtigen Module beinhaltet, die man benötigt, um mit den Sensoren in MicroPython zu experimentieren. In einer detaillierten Anleitung erklärt Arduino erst, wie man den ursprünglichen Bootloader im Flash-Speicher vom Anfang ans Ende versetzt und was das für die weitere Handhabung bedeutet. Danach lädt man OpenMV mit der zugehörigen OpenMV IDE auf das Board.

Anschließend kann man sich in der IDE ein paar Beispiele anschauen, die für den Nano 33 BLE Sense konzipiert sind. Öffnet man den seriellen Monitor, erscheinen im REPL (je nach Beispiel) etwa die Messwerte der einzelnen Sensoren, während man den Nano dreht und schwenkt. Es gibt auch eine FFT-Demo (engl. Fast Fourier Transformation), die das Mikrofon des Mikrocontrollers nutzt und die verarbeiteten Signale live in der IDE rechts oben anzeigt. Nur den ADPS9960-Sensor konnte ich mit OpenMV Version 4.5.0 leider nicht aus-

probieren, da die Module in dieser Firmware-Version fehlen. Das Team von OpenMV arbeitet aber gerade an einer Lösung.

Sobald man beabsichtigt, ein MicroPython-Programm auf den Nano zu überspielen, wird es aber knifflig, denn die OpenMV IDE sieht das gar nicht vor – es geht aber trotzdem. Dafür muss man nur eine andere IDE verwenden. Allerdings ist die Handhabung der Dateien aufgrund der besonderen Speicheraufteilung des Nano (siehe Link in Kurzinfor) nicht so unkompliziert, wie man es sonst von MicroPython kennt. Dateien, die man auf den Mikrocontroller überspielt, werden an einem Ort im Flash-Speicher abgelegt (vermutlich hinter der OpenMV-Firmware), an den man nicht so einfach gelangt. Vor einem Update von OpenMV sollte man daher in jedem Fall alle Daten löschen, weil es sonst passieren kann, dass sich das Board aufgrund von übrig gebliebenem Datenmüll nicht mehr mit IDEs verbinden mag (ist mir zumindest passiert).

Arduino hat mit dem Nano 33 BLE Sense ziemlich viel Technik auf engstem Raum gepfercht und sie funktioniert auch mit MicroPython. Ich hätte mir nur ein Dateisystem wie beim RP2040 oder ESP32 gewünscht. Mit den beschriebenen Herausforderungen ist der Nano eher ein Board für Fortgeschrittene.

Adafruit Prop-Maker Feather

Wenn man sich SciFi-Requisiten (engl. props) anschaut, bestehen sie oft aus einer Mischung von Licht, Sounds und Bewegungen. Denkt man beispielsweise an die Geisterfalle aus den

Ghostbusters-Filmen oder den Bewegungsmelder aus Aliens, wären diese vermutlich ziemlich langweilig, wenn sie nur eines dieser drei Dinge könnten. Für alle, die solche Requisiten oder Spielzeuge bauen möchten, hat Adafruit das RP2040 Prop-Maker Feather entwickelt. Mit diesem Board, das man mit CircuitPython programmieren kann, lassen sich gleichzeitig ein NeoPixel-LED-Streifen, ein Lautsprecher und ein Servo betreiben. Außerdem ist auf dem Prop-Maker Feather ein Drei-Achsen-Beschleunigungssensor verlötet, der schon reagiert, wenn man ihn antippt.

Damit die selbst gebauten Requisiten auch portabel sind, kann man an der JST-PH-Buchse einen 3,7-V-Akku (LiPo/Lilon) anschließen, der mit 200 mA geladen wird, sobald man das Mikrocontroller-Board über USB-C mit einer Stromquelle verbindet. Man sollte nur unbedingt abgleichen, ob die Polung am Akku identisch mit der Buchse auf dem Board ist, um nicht aus Versehen die Elektronik zu beschädigen. Wer lieber Batterien verwenden möchte, muss auf der Rückseite des Boards die Lötbrücke des LiPo-Chargers trennen. Außerdem kann man noch ein bisschen Strom sparen, indem man die Stromzufuhr des Audio-Verstärkers (MAX98357) sowie für den Neo-Pixel-LED-Streifen per Code ein- oder ausschaltet.

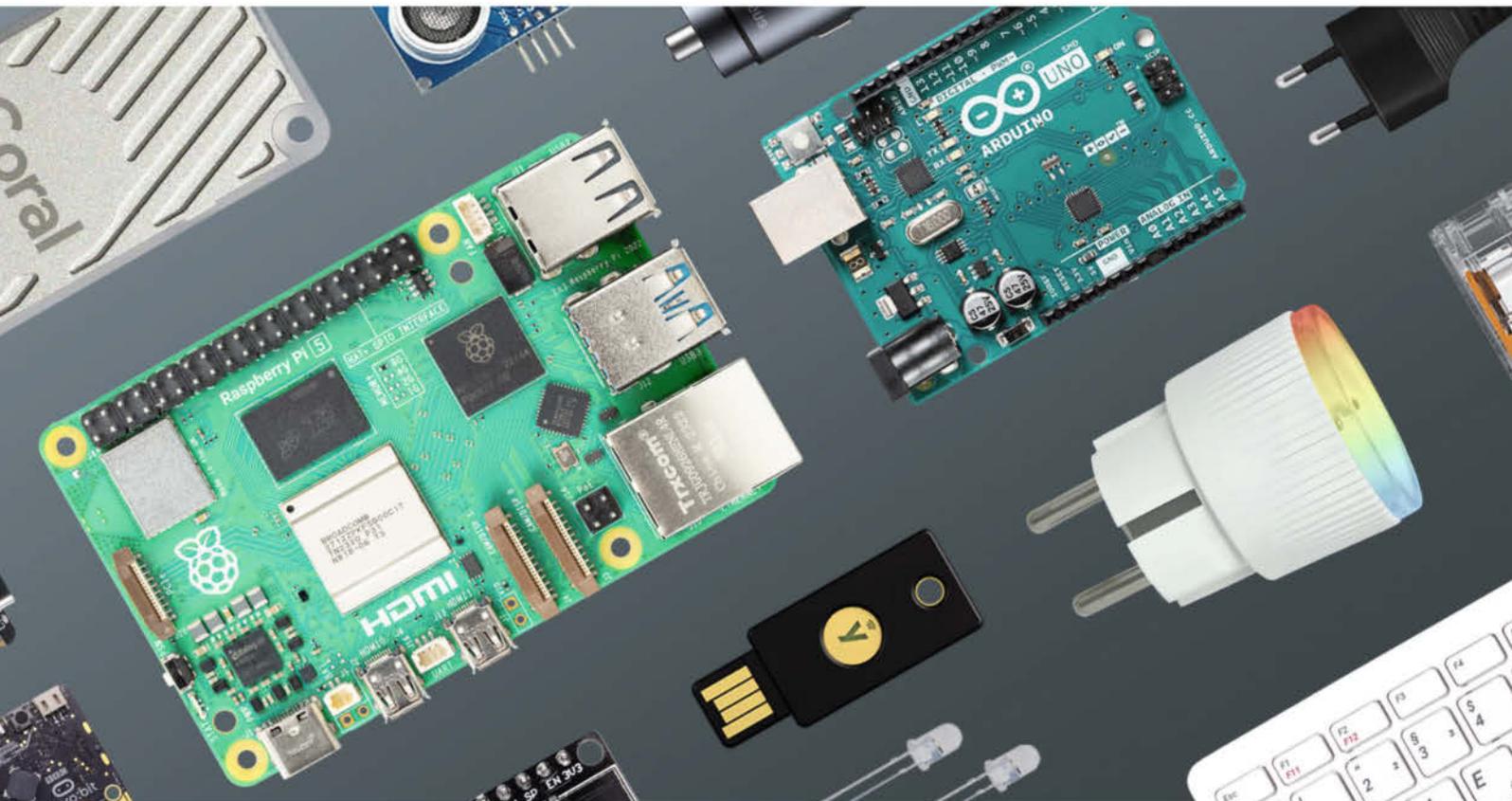
Wie bei Adafruit-Produkten üblich, findet man auf der Hersteller-Website ausführliche Informationen zu allen Anschlüssen und Funktionen des Boards, sowie eine Schritt-für-Schritt-Anleitung, wie man es in Betrieb nimmt. Um mit dem Board loszulegen, musste ich zunächst CircuitPython neu flashen. Die ausführliche CircuitPython-Einführung habe ich anschließend übersprungen und mir direkt die Beispiele in der Rubrik „CircuitPython Essentials“ angeschaut. Diese führen nacheinander durch die speziellen Funktionen des Boards, angefangen bei blinkenden LEDs, über die Soundausgabe bis zu einem Projekt, das alles miteinander verbindet. Und wenn am Ende die lustige Musik aus einem angeschlossenen Game-Boy-Lautsprecher trällert und gleichzeitig die LEDs blinken, fängt man an, herumzuspinnen, was man wohl als Nächstes mit dem Board umsetzen wird.

Fazit

Auch wenn ich am liebsten von Grund auf verstehen mag, wie etwas funktioniert und ich mich jedes Mal über selbst zusammengebaute Hardware freue, kostet das doch viel Zeit und ist mit vielen Herausforderungen verbunden. Mit den hier getesteten Boards war ich dagegen unmittelbar im Thema und konnte einfach ausprobieren, was man mit der fertigen Hardware machen kann. Dass die Boards außerdem MicroPython unterstützen, hat das Ganze noch vereinfacht. —akf



BERRYBASE
The Maker Shop



RASPBERRY PI | DEV. BOARDS | SENSOREN / MODULE | SMART LIVING | BAUELEMENTE | U. V. M.

ALLES FÜR DEIN NÄCHSTES PROJEKT!

Du möchtest mit dem Raspberry Pi einen Magic Mirror erschaffen, mit dem ESP32 eine Wetterstation verwirklichen oder mit dem Arduino deine eigene CO2-Ampel bauen? Oder planst du vielleicht ein komplettes Smart Home? Kein Problem: Du hast die Ideen, und wir haben alles, was du für dein nächstes Projekt brauchst! Besuche uns auf Berrybase.de.

ENTDECKE MEHR ALS 7.000 PRODUKTE AUF [BERRYBASE.DE](https://Berrybase.de)



Jubiläums-Maker-Faire in Hannover

Neuer Termin für die zehnte Maker Faire in Hannover und Fahrplan für Maker Faires im deutschsprachigen Raum für 2024.

von Daniel Schwabe

Seit 2013 treffen sich Maker aus der ganzen Welt einmal jährlich im Hannover Congress Centrum (HCC). Selbst während der Corona-Pandemie gab es zwei digitale Veranstaltungen. Nun ist es soweit, die zehnte Maker Faire Hannover in Präsenz findet 2024 statt – allerdings schon eine Woche früher als ursprünglich geplant: am 17. und 18. August. Für das Jubiläum laufen die Vorbereitungen bereits auf Hochtouren, um ein ganz besonderes Rahmenprogramm für die ganze Familie zu präsentieren.



2013 hieß die Maker Faire Hannover ihre ersten Gäste willkommen.

Wie kann man teilnehmen?

Die Stände auf der Maker Faire sind für private Maker, Vereine, Hochschulen, Unis und Forschungsinstitute kostenfrei. Über den bereits eröffneten „Call for Makers“ können sich Interessierte schon anmelden (siehe Link).

Unternehmen, die ihre Angebote und Produkte auf der Maker Faire präsentieren wollen, können bis Ende dieses Jahres noch von besonderen Frühbucherkonditionen profitieren.

Und wer als Besucher auf die Messe kommen möchte, kann bereits ab Dezember 2023 seine Tickets kaufen. Das ist übrigens das perfekte Weihnachtsgeschenk für die Maker in der Familie.

Noch mehr Messen im deutschsprachigen Raum

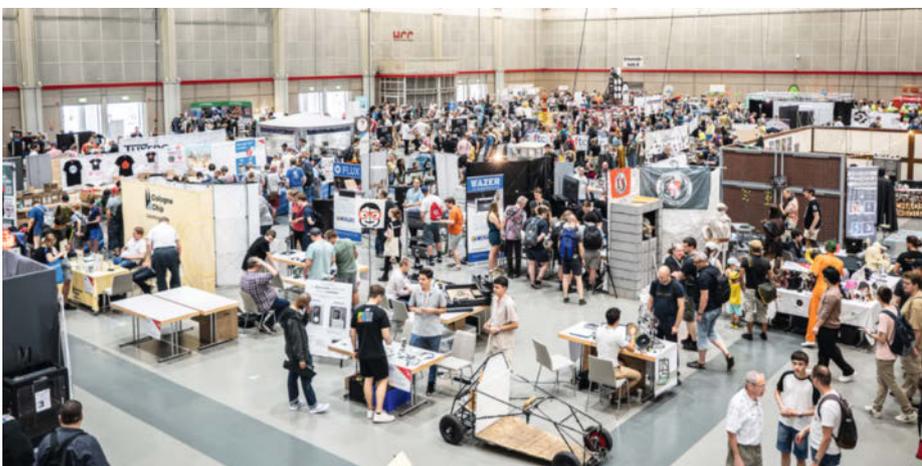
Seit 2013 hat sich die Maker Faire in ganz Deutschland, Österreich und der Schweiz etabliert. Für 2024 stehen bereits drei Termine im deutschsprachigen Raum neben Hannover fest:

- Maker Faire Heilbronn am 24. Februar 2024
- Maker Faire Ruhr in Dortmund am 16. und 17. März 2024
- Maker Faire Vienna vom 25. bis 27. Oktober 2024

Weitere Maker Faires sind geplant, Lizenzen für die Regionen Köln/Düsseldorf, Bayern und die Schweiz sind noch zu vergeben.

Wer immer die neusten Informationen zu Maker Faires im DACH-Raum und darüber hinaus haben möchte, kann sich für den zweiwöchentlichen Newsletter auf der Homepage der Maker Faire anmelden (siehe Link). —das

► make-magazin.de/x2kw



Jedes Jahr füllt die Maker Faire die Hallen mit Makern aus der ganzen Welt.



Beeindruckende Shows, bei denen Maker ihre Kreationen vorstellen, sind wichtiger Bestandteil der Maker Faire.



DIGITAL DESIGN & UX NEXT

Produktentwicklung, Technologiepotenziale und Gestaltung zusammendenken

Konferenz • München • 16. – 18. April 2024

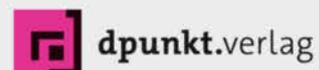
Ganzheitliches Design und nahtlose User Experience sind die Bausteine für erfolgreiche Produkte.

In Vorträgen und Workshops erfahren Sie, wie Sie **UX Design, Produktmanagement** und **Technologiekompetenz** in multidisziplinären Teams integrieren können. Unsere Konferenz bietet Ihnen Einblicke in die **aktuellen Trends** und zeigt praktische Ansätze und **Best Practices**, die Sie in Ihrem eigenen Unternehmen anwenden können.

Digital Design & UX Next – das Event-Ereignis für Usability- & UX-Profis, Digital Designer, Requirement Engineers und Product Owner.

www.dd-ux.de | Jetzt Frühbucherticket sichern!

Veranstalter



Bausätze für Maker

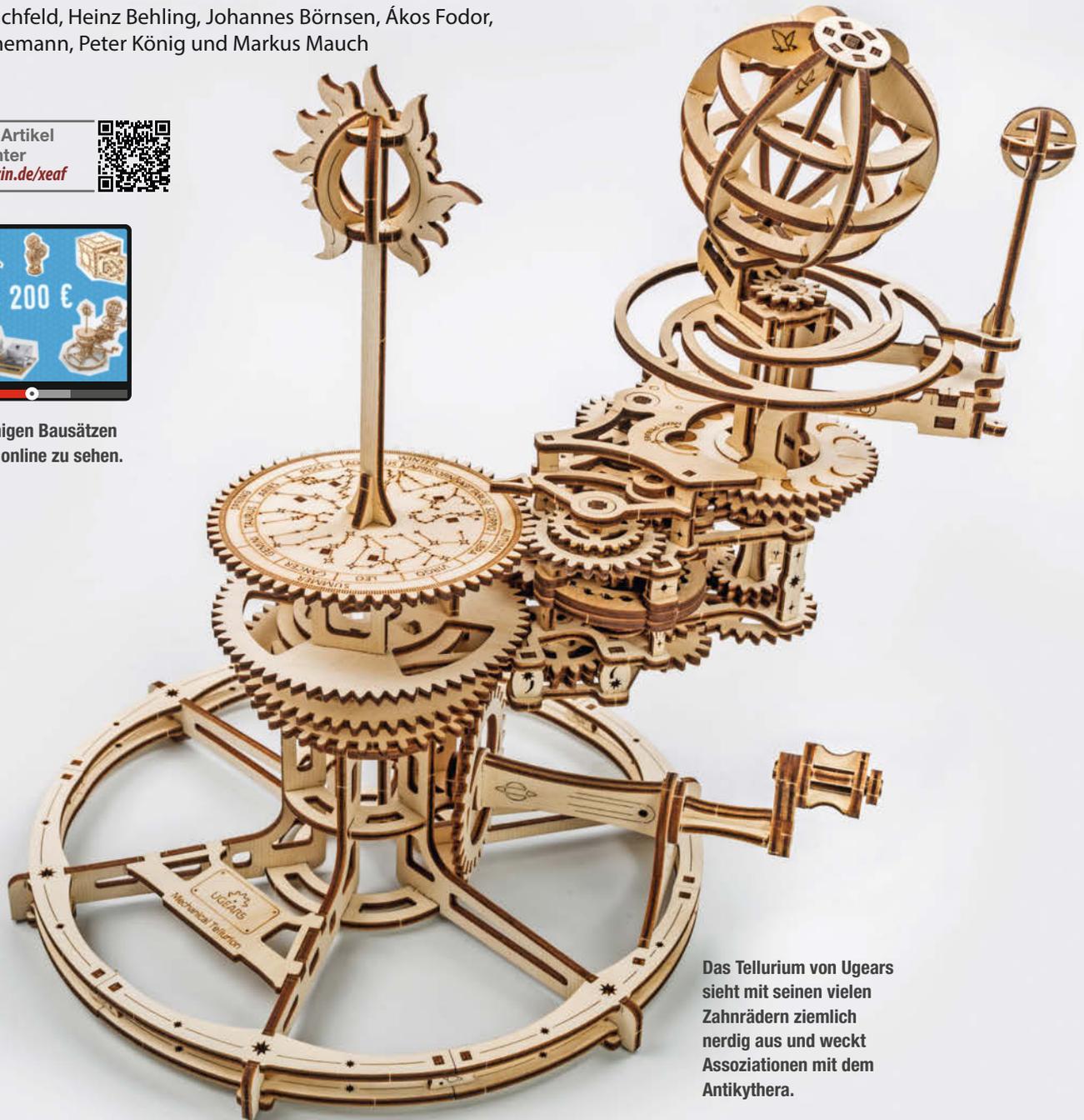
Make hat Bausätze und Experimentierkits aus Holz, Pappe, Kunststoff und Platinen getestet, die sich gut als Weihnachtsgeschenke für Maker eignen.

von Daniel Bachfeld, Heinz Behling, Johannes Börnsen, Ákos Fodor, Kathrin Grannemann, Peter König und Markus Mauch

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



Ein Video von einigen Bausätzen
in Aktion gibt es online zu sehen.



Das Tellurium von Ugears sieht mit seinen vielen Zahnrädern ziemlich nerdig aus und weckt Assoziationen mit dem Antikythera.

Weihnachten versetzt viele Menschen in Panik. Was soll ich denn nur schenken? Und dann soll es auch noch sinnvoll sein? Notoriker schenken jedes Jahr Socken und ein Buch aus der Spiegel-Bestseller-Liste. Stoiker respektive Maker verschenken spannende Bausätze. Die Make-Redaktion und ihre Autoren haben sich ein paar Kits rausgesucht und getestet, die man ohne Bedenken verschenken und sich selber gönnen kann. Für jeden Geldbeutel,

jede Geschicklichkeit und jede Werkzeugausstattung bzw. Nicht-Ausstattung ist etwas dabei: Maschinen aus Sperrholz, physikalische Anwendungen aus Pappe/Karton, Elektronik, Roboter und Fischertechnik.

Auf den folgenden Seiten finden Sie unsere Tests und Aufbauverfahren zusammengefasst. Ein Video von einigen der vorgestellten Kits in Aktion gibt es über den Kurzlink.
—dab/—pek

AstroMedia

Die Spezialität des Verlags AstroMedia sind Bausätze funktionsfähiger Versionen klassischer und wissenschaftlicher Apparate vom Teleskop über den Sextant bis hin zu Stirling-Motor und Dampfmaschine. Wo immer es geht, ist Karton das Material der Wahl, sinnvoll ergänzt durch Einzelteile aus Kunststoff, Silikon, Glas oder Metall.

Mit manchen AstroMedia-Bausätzen ist man schnell mit dem Aufbau fertig, andere beschäftigen einen viele Stunden lang. Wir haben mit dem Handspektroskop und der Dampfmaschine einen einfachen und einen komplexen Bausatz ausprobiert.

Handspektroskop

Das Handspektroskop (7,90 Euro) ist zusammengebaut in etwa so groß wie die Hand eines Erwachsenen und in rund einer Stunde einsatzbereit. Der Bausatz besteht aus einem bedruckten A4-Kartonbogen sowie einer Okularlinse, einem Beugungsgitter und einem Stück Film mit dem Lichtspalt und der Skala zum Ablesen der Licht-Wellenlängen.

Die Bauanleitung ist gleich mit auf freien Flächen des Kartonbogens aufgedruckt, aus dem man die drei Gehäuseteile mit Schere oder Cutter ausschneidet. Falzlinien rillt man mit einem Messerrücken oder einer Stopfnadel an einem Lineal entlang vor, für Klebungen gibt es Laschen – wer schon mal Kartonmodelle gebaut hat, für den ist hier nichts Überraschendes dabei. Etwas Vorsicht muss man beim Einkleben von Linse, Gitter und Film walten lassen, damit nicht verkehrt herum montiert oder durch Klebstoff an den falschen Stellen trübe wird.

Fertiggestellt kann man mit Hilfe des handlichen Geräts das Spektrum unterschiedlicher künstlicher Lichtquellen erkunden. Zur Zeit klassischer Glühlampen war das wahrscheinlich weniger spannend als heute, wo sich bei verschiedenen LED-Lichtquellen recht unterschiedliche und manchmal überraschende Lichtspektren zeigen. Am dankbarsten als Forschungsobjekt sind helle Lampen mit großen Milchglashauben etwa in Küche und Bad. In die Sonne sollte man mit dem Gerät allerdings auf keinen Fall schauen, das steht auch extra auf dem fertigen Spektroskop aufgedruckt. Wer das Sonnenspektrum sehen will, peilt besser einen weißen Vorhang an, der von der anderen Seite von der Sonne beschienen wird.

Dampfmaschine

Die Dampfmaschine (39,90 Euro) ist eines der aufwändigeren Modelle des Herstellers, deren Aufbau durchaus zwei lange Tage ausfüllt.

Anders als beim Taschenspektroskop braucht es hier weder Schere noch Cutter fürs Ausschneiden der Bauteile, die sind auf allen acht Kartonbögen vorgestanzt und die Falzlinien genietet.

Das ist eine große Hilfe, weil man sehr viele Bauteile der Dampfmaschine letzten Endes aus jeweils vier Pappschichten zusammenklebt – auf zwei innere Schichten, oft mit Aussparungen für die gelochten Kunststoff-Achslager-Scheiben, kommen je zwei Deckschichten, meist aufwendig mit Goldfarbe und schwarzen Verzierungen bedruckt. Die exakte Stanzung der äußeren Form erleichtert es hier deutlich, die Teile wirklich passgenau aufeinander zu fügen. Diese Sandwich-Bauweise führt zu hoher Steifigkeit der Teile, aber auch zu einem erheblichen Klebstoffverbrauch.

Vom Karton als Hauptmaterial sollte man sich nicht täuschen lassen: Das ist kein Bausatz für Kinder, eher ein schönes Projekt, was man zusammen mit bastelaffinen und geduldrigen Kindern in Angriff nehmen kann. Auf dem Weg zur fertigen Maschine sind ein paar Herausforderungen zu meistern: Arbeitet man etwa beim Zusammenkleben der zentralen Säule, die den Dreh- und Angelpunkt für den Balancier bildet, nicht penibel, stehen die Achsen der ganzen Kraftübertragung später vielleicht nicht im rechten Winkel zueinander, was im



AstroMedia

Das Handspektroskop gehört zu den einfacheren Bausätzen von AstroMedia. Ein Beugungsgitter mit 1000 Linien pro Millimeter im Inneren zerlegt einfallendes Licht in sein Spektrum.

Betrieb für unnötige Reibung sorgt. An einigen Stellen muss man vom Alleskleber zum Zwei-Komponenten-Kleber wechseln, ein Loch in den Dosendeckel bohren, dünnes Blech zuschneiden und biegen und weitere Kartonferne Aufgaben lösen. Wer das nicht scheut, wird damit belohnt, dass unter den eigenen Händen allmählich eine ausgeklügelte Konstruktion heranwächst, bei der geschickt eingesetzte dampfesistente Materialien die Pappe vor zu viel Hitze und Feuchtigkeit schützen. —pek

Die Bausätze wurden uns von AstroMedia für den Test zur Verfügung gestellt.



Für den Aufbau der funktionsfähigen Dampfmaschine braucht man ein langes Wochenende, an dem man sonst nichts vorhat. Die ausgeklügelte Konstruktion fasziniert aber garantiert.

AstroMedia

Ugears

Ugears ist ein ukrainisches Unternehmen, das Lasercut-Bausätze aus Sperrholz herstellt. Die Modelle besitzen zahlreiche mechanische Funktionen wie bewegliche Räder oder Kolben in Motoren bis hin zu voll funktionsfähigen Pendeluhrn. Die Bausätze werden in der Ukraine gefertigt, aber durch Händler in Deutschland verkauft und versendet.

Falls beim Aufbau oder später einmal etwas zu Bruch geht: Es gibt einen kostenlosen Ersatzteil-Service. Dazu muss auf der Internet-Seite des Händlers (siehe Kurz-URL) die Bausatzbezeichnung und die Teilenummer angegeben werden. Kostenlos heißt übrigens, dass Ugears weder etwas für das Teil selbst noch für den Versand berechnet. Die Ersatzteile werden als Geschenk deklariert von Kiev aus verschickt. Der deutsche Händler weist aber darauf hin, dass trotzdem ab und zu Zollbehörden Gebühren in Rechnung stellen würden.

Steampunk-Uhr

Wer in Lasercut-Modelle reinschnuppern möchte, ist bei der Steampunk-Uhr von Ugears für 11,90 Euro richtig: Mit 43 Teilen und einer Aufbauzeit von nur rund einer Stunde ist sie ein ideales Einsteigermodell. Über einen Antrieb verfügt sie nicht, die Zeiger bewegen sich beim manuellen Antrieb an den Zahnrädern jedoch im richtigen Geschwindigkeitsverhältnis auf einer gemeinsamen Achse. Die Bauteile kommen auf drei kleine Platten verteilt. Die Aufbauanleitung ist auf den ersten Blick etwas knapp, lässt sich aber Schritt für Schritt gut nachvollziehen. —jom

Tresor

Deutlich komplexer wird es beim Tresor (64,90 Euro): 179 Teile ergeben einen Safe, dessen Tür sich mit einem dreistelligen Code öffnen lässt. Der gewünschte Code kann beim Zusammenbau definiert werden, ist nachträglich jedoch nicht mehr zu verstellen. Ist die Tür geöffnet, liegt der Blick auf die Verriegelungsmechanik offen. Drei Räder verhindern das Zurückschnappen der Verriegelung. Bringt man die auf den Rädern befindlichen Aussparungen mit dem Code in die richtige Position, wird der Riegel von einem Gummiband zurückgezogen und die Tür kann geöffnet werden.

Unser Testmodell hat sich gegen die Montage jedoch etwas gewehrt: Die Ecken des Tresorkorpus werden mit Fingerzinken verbunden, zu deren Montage leichte Überzeugungsarbeit mit einem Gummihammer notwendig war. An vielen Teilen haben wir mit einer Schlüsselfeile die dunklen Laser-Schnittkanten der Steckzapfen entfernt, was das In-

einanderstecken deutlich erleichterte. Auch eine kleine Zange, ein Modellbauseitenschneider, ein Splinten-Austreiber zum Herausdrücken der Teile aus ihrem Bogen und ein Stechbeitel zum Entfernen von Graten an schwer zugänglichen Stellen haben sich im Laufe der Aufbauzeit von etwa acht Stunden zu uns gesellt. Und noch ein wichtiger Tipp aus eigener Erfahrung: Achten Sie beim Zusammenbau penibel auf die roten Ausrufezeichen in der Anleitung. Sie markieren die korrekte Ausrichtung von Teilen, die sich auch verdreht montieren lassen. Da die Drehachsen der Code-Scheiben mit Widerhaken ineinander gesteckt werden, ist nachträgliche Demontage zwecks Fehlerkorrektur schwierig. —jom

Murmel-Kettenbahn

Bei dieser Murmelbahn (es gibt noch drei andere Modelle) werden die Kugeln mit einer per Kurbel angetriebenen Kette nach oben transportiert. Jede Kugel schaltet den Weg für die jeweils folgende um, drei Wege stehen zur Verfügung.

Der Bausatz (52,90 Euro) war sehr präzise gefertigt, sämtliche Teile konnten problemlos aus den 5 1/2 Sperrholzplatten herausgelöst werden. Die Platten und die Teile sind num-



43 Teile ergeben eine Uhr ohne Antrieb. Lehrreich ist der kleine Bausatz trotzdem. Oder haben Sie schon mal ein Uhrwerk gebaut?



Der Sperrholztresor schützt die Schokoladenvorräte vor ungewolltem Zugriff dank frei einstellbarem Code.

meriert. Das ausführlich bebilderte Handbuch zeigt bei jedem Bauabschnitt, wo und auf welcher Platte das jeweilige Teil zu finden ist; außerdem, welche Teile mit dem beiliegenden Schmirgelpapier geglättet werden müssen und/oder eine Schmierung mit dem ebenfalls enthaltenen Wachsstäbchen brauchen. Rote Ausrufezeichen zeigen auch hier an, wenn es auf die genaue Ausrichtung eines Teils ankommt. Für einige empfindliche oder besonders kleine Teile liegen Ersatzteile bei, die auf den Holzplatten mit einem Pluszeichen gekennzeichnet sind.

Am kompliziertesten war die namensgebende Transportkette, die aus über 70 Einzelteilen besteht. Das Zusammenstecken ist problemlos, erfordert aber manchmal etwas Kraft, da einige Teile auf Presspassung gearbeitet sind, um dauerhaft zusammenzuhalten. Insgesamt braucht man für das Zusammenfügen der 400 Bauteile gut sieben Stunden, die aber richtig Spaß machen.

Diese und die anderen Murmelbahnen von Ugears haben noch eine Besonderheit: Bis zu vier Bahnen (gleiche oder unterschiedliche) können miteinander kombiniert werden, sodass die Bahnen von einer einzigen Kurbel angetrieben werden und die Kugeln nacheinander alle Bahnen durchlaufen, bevor sie wieder am Startpunkt anlangen. Die entsprechenden Verbindungsteile sowie Wegeumsteller sind im Bausatz enthalten. —hgb

Tellurium

Astronomische Geräte haben es mir schon immer angetan und so entschied ich mich für den Bau des Telluriums (von Tellus, Erde, siehe Titelbild des Hefts und dieses Artikels). Es demonstriert die Bewegungen der Erde (inklusive deren Drehung um die eigene Achse) um die Sonne und des Mondes um die Erde, sodass sich die Entstehung der Jahreszeiten, Mondphasen und Finsternisse veranschaulichen lässt. Für Kenner: Ein Tellurium ist ein Sonderfall einer Planetenmaschine, die alle Planeten enthält. Das Tellurium wird mittels Kurbel bedient, wobei eine Kurbeldrehung einem Tag entspricht. Für ein Jahr muss man also 365 mal kurbeln. An einer Stelle des Geräts kann man die jeweilige Mondphase ablesen, an einer anderen Stelle das Sternzeichen.

Der Bausatz besteht aus 6 Brettchen mit 249 durchnummerierten, fein gelaserten Teilen, die sich leicht herauslösen ließen. Die gedruckte, farbige Anleitung ist klar strukturiert und die jeweiligen Schritte sind sehr gut erkennbar. Die Anleitung erinnert an Lego-Aufbauanleitungen, bei denen in der Regel auch nicht viel schief gehen kann. Der Aufbau ist ein No Brainer und hat fast schon meditativen Charakter. Und: Es hat einen gewissen

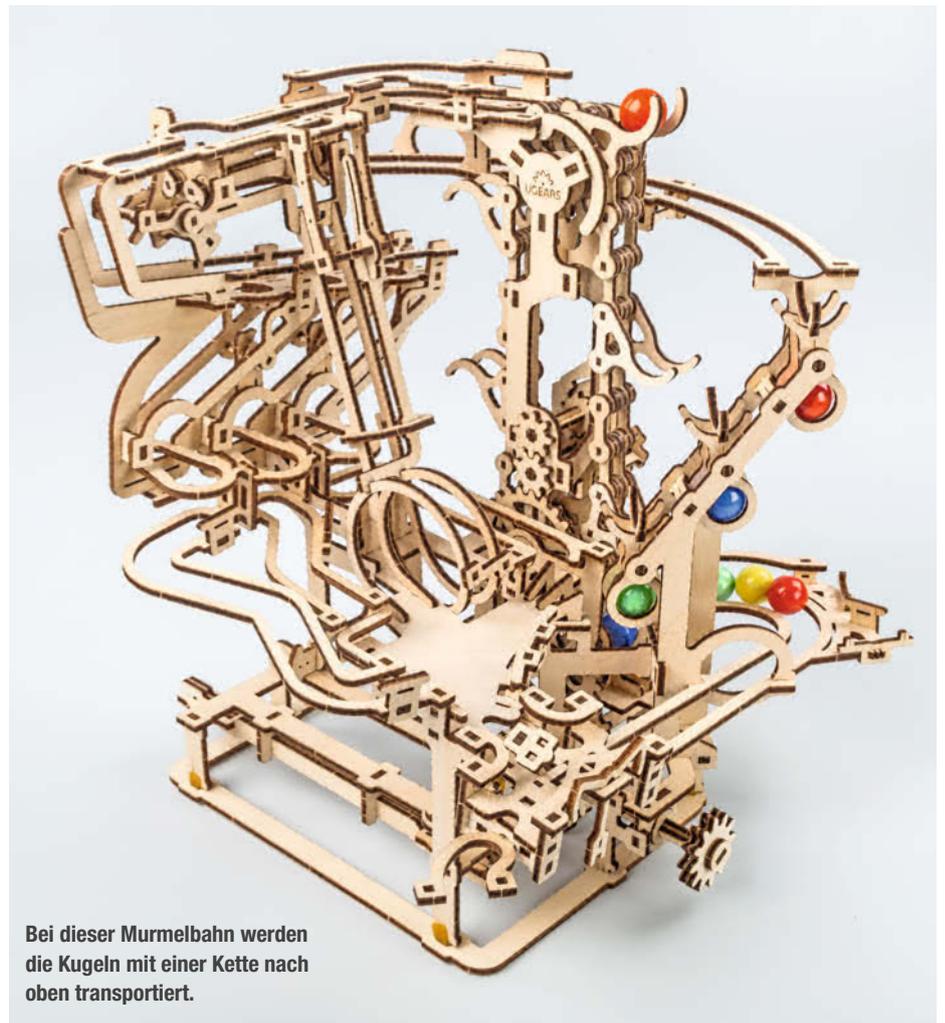
Suchtfaktor, die Teile nach und nach zu etwas größerem zusammenzufügen. Für letzteres benötigt man mitunter etwas Kraft, weil der Bausatz auf Presspassungen beruht und damit Klebstoff überflüssig macht.

Das Tellurium verwendet viele Zahnräder, die zu Getrieben und sogar einem kleinen Planetengetriebe zusammengefügt werden. Bei den Zahnrädern muss man peinlichst darauf achten, dass keine Holzreste zwischen den Zähnen hängen bleiben, um Blockaden zu vermeiden. Als Lager der Wellen und Achsen dienen Holzscheiben, die man mit dem beiliegenden Schleifpapier entgraten und mit dem ebenfalls beiliegenden Kerzenwachs schmieren muss. Damit erhält man später ein für Holz erstaunlich leichtlaufendes Modell.

Insgesamt hat der Aufbau rund fünf entspannte Stunden gedauert. Die Altersempfehlung liegt bei 14 Jahren, was aber weniger der kognitiven Leistung als vielmehr dem Kraftaufwand geschuldet ist. Für knapp 53 Euro bekommt man mehrere Stunden Bastelspaß und ein faszinierendes Hingucker-Modell (Höhe 32 cm, Durchmesser 33 cm) fürs Regal. —dab



Die Mondphasenanzeige des Telluriums



Bei dieser Murmelbahn werden die Kugeln mit einer Kette nach oben transportiert.

Blinkyparts

Auf der Suche nach dekorativen, lehrreichen Bausätzen kommt man hierzulande kaum an Blinkyparts vorbei. In diesem Shop findet man allerlei bunt Blinkendes in unterschiedlichen Schwierigkeitsstufen und Preisklassen, mit der sowohl Kinder als auch Erwachsene ihren Spaß haben und dabei noch etwas über Schaltungen und die Funktion der Bauteile lernen. Viele der Sets stammen aus der Binary Kitchen, dem Regensburger Hack- und Makespace.

Cat Lamp Soldering Kit

Wir haben für euch zwei Bausätze genauer angeschaut und zusammengebaut. Das Cat Lamp Soldering Kit von Steph Piper aka Maker Queen aus Australien beinhaltet eine wunderschön ausgestaltete Platine sowie eine Handvoll Bauteile und kostet 19,89 Euro. Lötet man Fotoresistor, Transistor, Widerstand, Batteriehalter, Schalter sowie eine LED in die katzenförmige Platine ein, entsteht binnen einer halben Stunde eine dekorative Lampe, die bei Dunkelheit automatisch anspringt. Eine großformatige Anleitung erklärt jeden Schritt detailliert und gibt Tipps zur Verarbeitung von Bauteilen und dem Löten an sich. Solltet ihr keine Dritte Hand als Lötthilfe zur Verfügung haben, nutzt die Umverpackung des Bausatzes beim Anlöten der einzelnen Platinenteile.



Der LED Cube bringt 27 bunte LEDs in Würfelform.

Über einen QR-Code kommt man zu einem Video, das jeden der Schritte ausführlich beschreibt. Ein Hinweis sei an dieser Stelle gegeben: Die Bauanleitung ist derzeit nur englischsprachig verfügbar.

LED Cube

Wer etwas mehr Zeit hat und ein bisschen kleinteiliger arbeiten möchte, für den ist der LED Cube für 11,49 Euro etwas. Der Bausatz enthält 27 LEDs, einen Batteriehalter, einen Schalter sowie ein Holzgehäuse, das während des Baus als Halterung für die LEDs nutzbar ist und hinterher die Basisplatte für den Würfel darstellt.

Die im Shop angegebene Bauzeit von zwei bis vier Stunden erschien mir im Vorfeld als recht großzügig kalkuliert, der Teufel steckt bei diesem Bausatz aber im positiven Sinne im Detail. Die Ebenen des Würfels zusammensetzen erfordert ein bisschen Geschick und Geduld - und damit auch Zeit. Hat man den Dreh einmal raus, wie man die Beinchen der LEDs zu einem Quadrat zusammensetzt, ist es eine machbare Aufgabe. Dieser Bausatz entspringt übrigens direkt der Binary Kitchen, dem Regensburger Hack- und Makespace.

Weitere Kits

Neben den hier vorgestellten Bausätzen gibt es bei Blinkyparts noch weitere Lötkits in unterschiedlichen Schwierigkeitsstufen. Für Einsteiger finden sich einfache LED-Schaltungen in Form eines Katzen-Ansteckers oder eine Eule mit leuchtenden Augen, die als „Joule Thief“ auch noch den letzten Rest aus vermeintlich leeren Batterien holt.

Etwas herausfordernder wird es mit dem SMD-Lauflicht, für das 20 0805-SMD-LEDs verarbeitet werden müssen und



Kleine Nachtleuchte in Form einer Katze: Cat Lamp Soldering Kit

sehr kompliziert mit dem legendären „Nibble++“, einer echten Herausforderung für Lötexperten mit SMD-Bauteilen bis zu einer „Größe“ von 01005 (0,4 × 0,2mm). Hier braucht man eine ruhige Hand und idealerweise eine Lupe zum Löten.

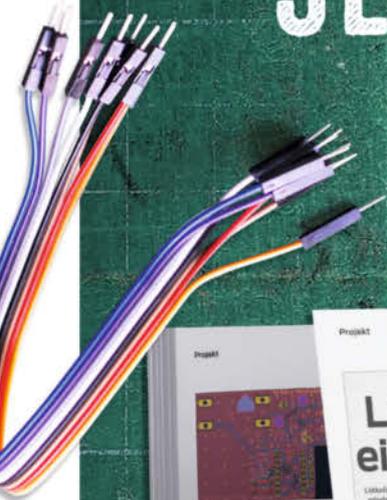
Neben blinkenden Dingen findet sich im Shop auch nützliches: Für knapp 80 Euro bietet Blinkyparts einen Bausatz für einen Feinstaubsensor, der in das Citizen Science-Projekt sensor.community eingebunden werden kann. Das Set enthält alle nötige Technik, lediglich um das Gehäuse und die Stromversorgung muss man sich selbst kümmern. Und wem bisher noch ein Labornetzteil fehlt, der kann sich mit dem ATX Breakout Board eins selbst bauen. Dieses bietet nach erfolgreicher Lötarbeit Festspannungen von 3,3, 5 und 12 Volt sowie eine variable Ansteuerung über ein Potentiometer. Zusätzlich benötigt man ein einfaches ATX-Netzteil. Wem das grundlegende Material zum Löten fehlt, der findet im Shop Zubehör wie Lötzinn, Lötunterlagen sowie den legendären Open-Source-Lötkolben Pinecil.

Zu guter Letzt noch ein Hinweis zum Shop: Ein Großteil der Gewinne vom Verkauf fließt laut eigener Aussagen in gemeinnützige Organisationen und Vereine. Zudem können Bildungseinrichtungen, Vereine und Hackspaces im Shop zu günstigeren Preisen einkaufen. Solltet ihr also bei der nächsten Veranstaltung einen Workshop geplant haben, lohnt sich eine Nachfrage per Mail.

—Kathrin Grannemann/dab

Make:

JETZT IM ABO GÜNSTIGER LESEN



GRATIS!



2x 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



Roboter

Variobot variAnt

Endlich mal kein Fahrroboter: Die stilisierte Ameise variAnt wird zwar auch von zwei Motoren angetrieben, allerdings drehen die hier mal keine Räder. Vielmehr treiben die Arduino-gesteuerten Mini-Getriebemotoren über einen raffinierten Mechanismus jeweils links und rechts drei Beine an. Das funktioniert mit dem zusammengebauten Modell in der Praxis auch überraschend gut. Bis dahin ist es allerdings ein teils steiniger Weg.

Vorab: Wer meint, Anleitungen seien nur für Warmduscher, der wird mit dem Bausatz seine Frustrationstoleranz in ungeahnte Höhen treiben. Das vorherige und gründliche Durchlesen aller Schritte verhindert viele Fehler und Rückbauten. Zudem trainiert der Bausatz die Feinmotorik: Kleine Inbusschraubchen halten alles zusammen und provozieren Menschen mit Wurstfingern aufs Äußerste. Wer also Lust auf echte Herausforderungen hat: Hier ist der richtige Bausatz.

Bei der Montage geht es im schwarzen Acryl-Ameisenkörper ziemlich eng zu. Irgendwo scheint immer ein Kabel im Weg zu sein. Manche Kabel sitzen sehr stramm auf der Steckerleiste und drohen sich im späteren Betrieb zu lösen – es wird nämlich nur gesteckt und nicht gelötet. Die Beinmechanik erfordert besondere Aufmerksamkeit, da die Antriebsstangen in unterschiedlichen Winkeln gebogen sind und die aufgesteckten Zahnradchen in dieser Hinsicht markiert werden müssen, um sie später einzubauen. Pro Seite müssen drei Beinchen und ein Motor mit insgesamt fünf Zahnradchen richtig ausgerichtet werden – ein Geduldsspiel. Als Beingelenke fungieren kleine, geschmierte Messingkugeln.

Im großen und ganzen ist die nur als PDF verfügbare, deutschsprachige, farbige Anleitung leicht verständlich und detailliert. An mehreren Stellen sind die Bilder aber aus einer ungünstigen Perspektive aufgenommen, etwa bei der Verkabelung der Mini-Breadboards. Ein Schaltplan à la Fritzing wäre da zielführender. Bei der Darstellung zur Montage von Messinghülsen für die Beinantriebe sind Knicke abgebildet, die es real in den Hülsen gar nicht gibt.

Zur Steuerung liegt dem Bausatz ein Board mit Motor-Treibern sowie ein Arduino-kompatibler Mikrocontroller bei. Die Motorumdrehungen und somit die gelaufene Distanz ermittelt der Controller mittels Reed-Relais, die durch auf den Motorachsen haftende Neodym-Magnete öffnen und schließen. Hindernisse erkennt die Ameise mittels mehrerer Fototransistoren, die an verschiedenen Stellen des Körpers verbaut werden. Mehrere LEDs signalisieren den Zustand der Ameise. Der



Die Roboterameise variAnt sieht dank des Acrylglass-Körpers nicht nur schick aus, sondern fasziniert auch mit ihrem geschmeidigen Gang.

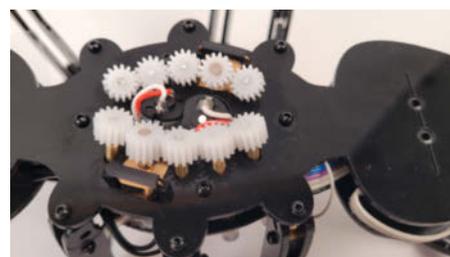
Hersteller bietet für die Arduino IDE eine Bibliothek zum Download an, die das Programmieren vereinfacht: `ant.setup()`, `ant.setSpeed(160)` und `ant.stepBack(5,10)` lassen die Ameise links 5 und rechts 10 Schritte rückwärts gehen.

Eine Menge raffinierter Einzellösungen sind in dem 345-teiligen Bausatz vereint, für den wir rund sieben Stunden plus eine Stunde Fehlersuche zum Betrieb benötigten. Wer das Teil zum Laufen bekommt, sollte vom Hersteller in die ewige Ruhmeshalle aufgenommen werden. Kollege Wolfgang Stieler von der Technology Review hat variAnt ebenfalls zusammengebaut, leider lief die Ameise anschließend nicht (TR 7/23, S.104). Der Bausatz ist eine Challenge und mit 200 Euro nicht gerade günstig. Er könnte sich unter Enthusiasten allerdings den Ruf eines einzigartigen und mit viel Liebe zum Detail konstruierten Roboters erarbeiten. —*dab*

Calli:bot

Inzwischen gibt es eine ganze Reihe von Mikrocontrollern, die speziell für Einsteiger, Kinder und Jugendliche sowie Bildungszwecke entwickelt wurden. Der Calliope mini ist hierbei der leuchtende Stern am Himmel dieser kompakten Einplatinencomputer. Nicht ohne Grund, denn er bringt bereits zahlreiche Sensoren und Aktoren onboard mit. Dazu gehören beispielsweise Tasten und Pins, ein Temperatursensor, ein Lautsprecher, ein 5x5-LED-Feld, eine RGB-LED, ein Lautstärke- und Helligkeitsmesser und vieles, vieles mehr.

Mit der großzügig bestückten, kleinen sternförmigen Platine allein lassen sich bereits Dutzende Programmierprojekte umsetzen: vom „Hello World“-Klassiker über ein Mini-



Die Ausrichtung der Zahnräder und der Wellen von variAnt ist ein Geduldsspiel.

Klavier bis hin zu einer „Fang das Ei“-Spielekonsole. Hierfür stehen intuitiv zu bedienende visuelle Programmierumgebungen zur Verfügung. Zu den bekanntesten zählen dabei das Open Roberta Lab, welches aus einer Initiative des IAIS des Fraunhofer Instituts entstanden ist, sowie MakeCode von Microsoft.

Das Schöne am Calliope mini ist, dass er durch verschiedene Bauteile noch erweitert werden kann. Hat man sich also am „Mini-Computer in der Hosentasche“ in der Basisvariante satt getüftelt, können separate LED, ein Ultraschall-, ein Feuchtigkeits- oder ein Farbsensor für frischen Wind und neue Möglichkeiten sorgen. Die anfängliche Insellösung Calliope mini wird so zur Steuerzentrale für externe Komponenten und bietet damit auch schon jüngeren Makern die Möglichkeit, Luft aus den Sphären von großen Geschwistern wie beispielsweise dem Arduino zu schnupern.

Eine ganz besondere Art und Weise, den Calliope mini zu erweitern, ist, ihn mit einem geeigneten Kit zu einem echten, fahrbaren Roboter werden zu lassen. Bemüht man die Suchmaschine, wird man zwar schnell fündig, doch tatsächlich gibt es nur wenige Erweite-

rungssets zum Fahrroboter, die auch tatsächlich halten, was sie versprechen. Als besonders empfehlenswert ist hier der Calli:bot 2 von Knotech für 57 Euro zu nennen: ein wohl durchdachter, kompakter und preiswerter fahrbarer Untersatz speziell für den Calliope mini, der in Deutschland gefertigt wird. Der Calli:bot 2 verfügt über zwei Metallgetriebemotoren mit Rädern, vier RGB- und zwei rote LEDs, zwei IR-Linienfolgesensoren, zwei Taster, einen Ultraschallsensor zur Abstandsmessung sowie zwei Ausgänge für Servomotoren.

Diese Aufzählung allein hört sich natürlich schon einmal gut an. An einem simplen Beispiel aber lässt sich aufzeigen, dass Knotech viel Verstand, Wissen und Herzblut in die Konzeption und Umsetzung des Calli:bot 2 gesetzt hat. Er macht nämlich das, was einige der Sets, mit denen ich in den vergangenen Jahren gearbeitet habe, nicht zuverlässig gemacht haben: Er fährt geradeaus, wenn er dies soll. Das klingt zwar vermutlich zunächst profan, doch ein 08/15-Erweiterungskit, welches das nicht tut, erzeugt Frust und verstaubt ziemlich schnell in irgendeiner dunklen Ecke. Auch alle anderen verbauten Sensoren und Aktoren arbeiten zuverlässig, sodass das Programmieren, Tüfteln, Spielen und Maken mit dem Calli:bot 2 wirklich eine Freude ist. —Markus Mauch/dab

Lego Spike Prime

Schon seit Jahrzehnten begeistert Lego mit seinen programmierbaren Robotersets Menschen aller Altersgruppen gleichermaßen. Der dänische Spielwarenhersteller ist Pionier im Bereich der „educational robots“ und gehört zu den Marktführern. Mit dem 2021 platzierten „Spike Prime“-Set als Nachfolger der weitverbreiteten „Mindstorms“-Serie wurde mit wohl-durchdachter Konsequenz die nächste Evolutionsstufe erklommen, auch wenn natürlich vieles von Spike Prime an Mindstorms erinnert.

Neben dem quaderförmigen Steuerhub finden sich in der stabilen Aufbewahrungsbox ein Abstands-, ein Kraft- und ein Farbsensor sowie insgesamt drei Motoren und jede Menge Kleinteile zum Bauen. Der Hub kann über eine eigene Entwicklungsumgebung blockbasiert programmiert werden. Eingebettet findet man hier zudem gut durchdachte Lerneinheiten, welche sich thematisch in Bereiche wie „Erfinderteam“ oder „Alltagshelfer“ unterteilen.

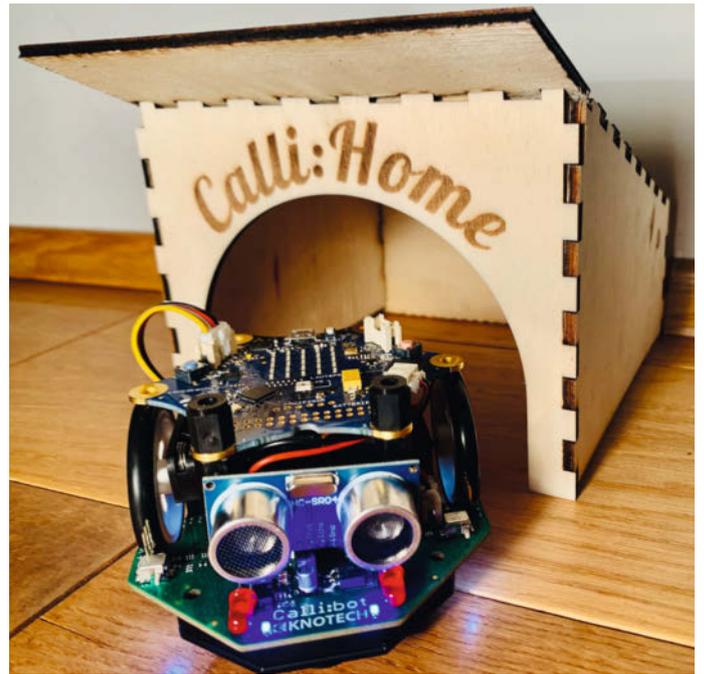
Beobachtet man Heranwachsende im Umgang mit einem Spike-Prime-Set, so ähneln sich die Vorgehensweisen hierbei häufig. Die meisten jungen Tüftler bauen zunächst einmal ein Modell nach dem anderen entsprechend der Anleitungen aus der Entwicklungsumgebung nach und bespielen dieses mit einem vorgefertigten Skript, welches an manchen Stellen noch erweitert oder fertiggestellt wer-

Mit dem Chassis wird der Calliope mini zum Calli:bot, hier in einem selbstgebauten Häuschen artgerecht gehalten.

den muss. Schritt für Schritt gehen sie dann aber dazu über, eigene Modelle zu bauen. Damit beginnt häufig der eigentlich spannende Teil. In vielen Fällen sind die Modelle zunächst Fahrroboter, die in ihrem Verhalten erst einmal strecken- bzw. zeitbasiert programmiert werden. Erst mit der Zeit entwickelt sich dann eine komplexere Vorgehens- und Denkweise, bei der dann auch die Sensoren mit einbezogen werden.

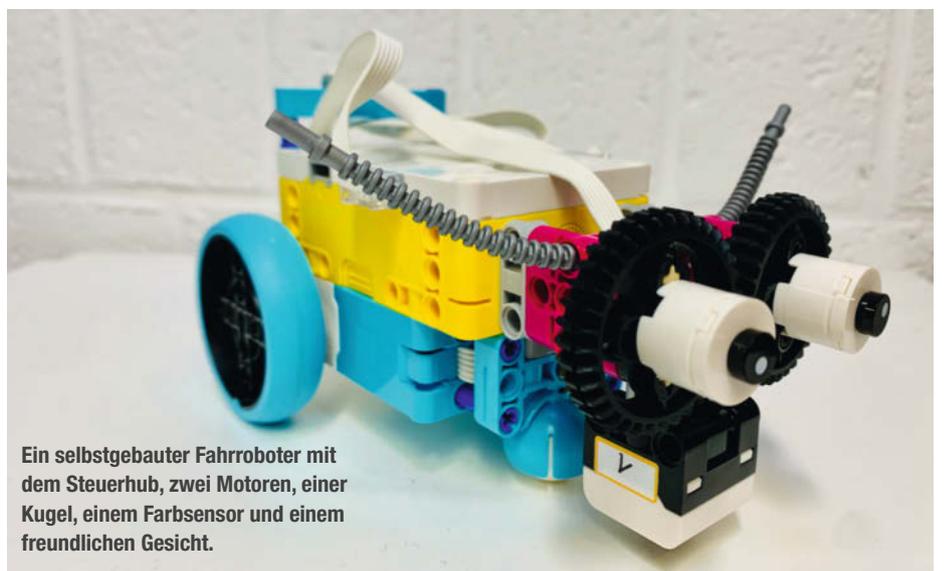
Das Tüfteln auf eigene Faust bringt aber auch immer wieder Rückschläge mit sich; beispielsweise dann, wenn die erdachte Konstruktion zu instabil ist oder die selbst konzipierte Mechanik nicht ganz das tut, was sie soll. Manchmal bedarf es im Übrigen auch Herausforderungen, die gestellt werden, um die Kreativität anzuregen; beispielsweise einen mit farbigem Malercrepp aufgeklebten Labyrinthgang, der von einem Fahrroboter mit Farbsensor durchfahren werden soll.

Bei aller Qualität des Sets und dessen Gesamtkonzeption sollte nicht außer acht gelassen werden, dass dieses durchaus einen stolzen Preis hat: ab 330 Euro Straßenpreis. Für einen ähnlichen Betrag könnte man alternativ



gleich mehrere Robotiksets anderer Hersteller erwerben.

Zusammenfassend lässt sich aber festhalten, dass Lego Spike Prime ein innovativer Klassiker ist, der es Kindern und Erwachsenen ermöglicht, sich spielerisch gemeinsam mit verschiedenen MINT-Perspektiven auseinanderzusetzen und in der vereinten Tüftelei zusammenzufinden. Die Qualität und Konzeption des Sets lässt sich der Hersteller zwar gut bezahlen, bereitet aber auch lange Freude und lädt immer wieder aufs Neue zum Experimentieren, Knobeln und Erfinden ein. Nicht ohne Grund ist Lego Spike Prime auch bei (inter-)nationalen Roboterwettbewerben für Kinder und Jugendliche wie beispielsweise der WRO (world robotic olympics) im Einsatz. —Markus Mauch/dab



Ein selbstgebauter Fahrroboter mit dem Steuerhub, zwei Motoren, einer Kugel, einem Farbsensor und einem freundlichen Gesicht.

fischertechnik – Smart Robots Pro

Bausätze von fischertechnik sind für viele junge Menschen der Einstieg in die Welt der Mechanik und Elektronik. Mit dem Set Smart Robots Pro (199 Euro) hat das deutsche Traditionsunternehmen seine Robotics-Sparte um einen neuen Bausatz erweitert, der sich an Kinder ab 8 Jahren richtet. Das Komplettpaket umfasst neben den bewährten fischertechnik-Bausteinen einen Mikrocontroller namens „BT Smart Controller“ und elektronische Komponenten wie Taster, Motoren, Fototransistoren und LEDs, die sich in zahllosen Varianten miteinander verschalten lassen. Programmiert werden die Modelle mit visuellen Bausteinen in der App „ROBO PRO Coding“, die optisch an Scratch oder Robotical erinnert.

Da das Einstiegsvideo, das die Programmierung und den „BT Smart Controller“ erklären soll, derzeit nicht verfügbar ist, muss man sich nach dem Aufbau der Modelle ein paar Inhalte selbst erschließen oder bereits Erfahrungen mitbringen. Fischertechnik arbeitet aber zurzeit an einer Lösung.

Aufbau leicht gemacht

Der Bausatz beinhaltet 362 Bausteine und bietet 12 Bauanleitungen für fahrende und tanzende Roboter, Maschinen und miniaturisierte Freizeitpark-Attraktionen. Drei Projekte kann man mit dem beiliegenden Heft nachbauen, weitere Anleitungen findet man im e-Learning-Portal, auf das man über einen QR-Code gelangt. Diese sind im Vergleich zur gedruckten Version sogar animiert, dreidimensional und interaktiv und auch auf einem Smartphone oder Tablet nutzbar. Das ist sehr hilfreich, da sich manche Bausteine nur geringfügig unterscheiden.

Programmieren mit Bausteinen

Damit sich der gebaute Roboter auch bewegen kann, nachdem man ihn zusammgebaut hat, lädt man sich die „ROBO PRO Coding“-App auf den Computer, das Smartphone oder Tablet. Um den Einstieg zu erleichtern, hat fischertechnik für fast jedes der zwölf Projekte ein paar Code-Beispiele vorbereitet, mit denen man die Roboter gleich testen kann, sobald der Roboter mit der App verbunden ist. Dann startet man das Programm direkt aus der App heraus. Ein Überspielen (Flashen) ist nicht notwendig und auch nicht möglich. Das heißt zwar, dass die „ROBO PRO Coding“-App immer verbunden sein muss, damit sich der Aufbau bewegen kann, gleichzeitig ist der Mikrocontroller dadurch weniger fehleranfällig. Wer seine Programme für einen späteren Zeitpunkt speichern oder mit anderen teilen



Der Bausatz, davor der Bauteilespender, der mit Hilfe einer Lichtschranke erkennt, wann er ein Bauteil in Richtung Rampe schieben darf.

möchte, kann sie auf einen Datenträger oder in fischertechniks GitLab exportieren.

Einschätzung

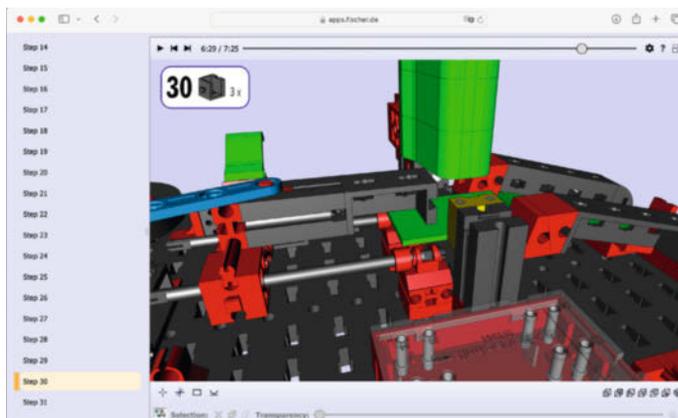
Für diesen Test habe ich die Modelle aus dem mitgelieferten Heft nachgebaut sowie den Bauteilespender, der wie eine kleine Fertigungsstraße aussieht. Auch wenn ich mir gewünscht hätte, schon vor dem Zusammenbau zu erfahren, was ich lernen werde, überraschten mich die getesteten Konstruktionen mit ihrer kreativen Bauweise. Der kleine Drivebot nutzt etwa die mit Tastern versehenen Stoßstangen, um nach einem Hindernis die Richtung zu ändern. Beim Ballspiel lernt man, wie man eine Abschussrampe samt Lademechanismus bauen kann – auch wenn die Kugeln nach meinem Geschmack noch etwas weiter fliegen dürften. Der Bauteilespender hat mich mit seiner Lichtschranke und dem Kurbelantrieb samt Sicherheitsmechanismus beeindruckt.

Die interaktiven Bauanleitungen lassen sich pausieren, drehen und schwenken.

Eines ist klar: Dieses Set lebt von den Herausforderungen, die der Aufbau und die Programmierung mit sich bringen und das „Pro“ in „Smart Robots Pro“ sollte man auf jeden Fall ernst nehmen. Während der Bausatz mit den interaktiven Anleitungen äußerst einsteigerfreundlich beginnt, hakelt es etwas am Übergang zur Programmierung. Hat man diese (nicht allzu große) Hürde aber überwunden, darf man sich über ein Set freuen, das viel Inspiration liefert und mit den großartig verarbeiteten und vielseitigen Bausteinen stundenlangen Spaß bescheren kann – auch großen Kindern.

—akf

Der Baukasten wurde uns von fischertechnik für den Test zur Verfügung gestellt.



heise +

ct

iX

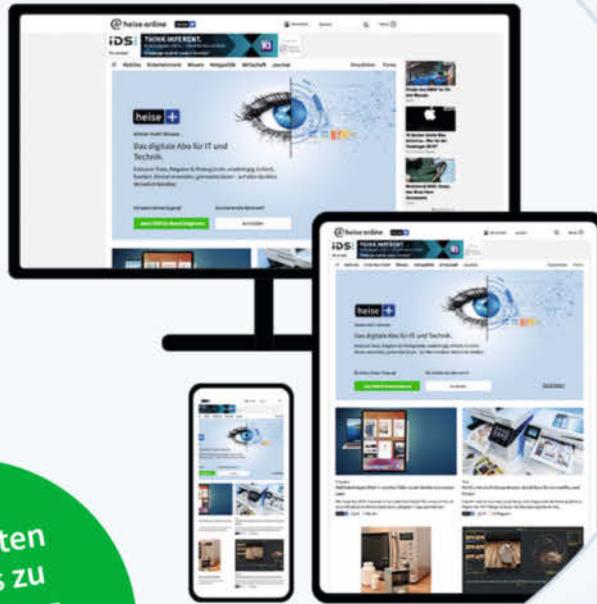
Mac&i

Make:

MIT
Technology
Review
Das Magazin für Innovation von Heise

ct *Fotografie*

Make-
Abonnenten
lesen bis zu
60%
günstiger



Das digitale Abo für IT und Technik.

Exklusives Angebot für Make-Abonnenten:
Sonderrabatt für Magazinabonnenten

- ✓ Zugriff auf alle Artikel von heise+
- ✓ Jeden Freitag exklusiver Newsletter der Chefredaktion
- ✓ Alle Heise-Magazine online lesen: c't, iX, MIT Technology Review, Mac & i, Make und c't Fotografie
- ✓ 1. Monat gratis lesen – danach jederzeit kündbar

Sie möchten dieses Exklusiv-Angebot nutzen? Jetzt bestellen unter:

heise.de/plus-testen

✉ leserservice@heise.de ☎ 0541 80009 120

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

Alles auf eine Karte

Weihnachten steht vor der Tür. Sorgen Sie dafür, dass der Briefträger sich dazu gesellt und Weihnachtskarten zustellt. Von Ihnen selbst gefertigte mit 3D-Effekt oder sogar Sound kommen sicher besonders gut an. Mit dieser Anleitung werden Sie zum Profi-Kartenmacher.

von Heinz Behling



Morgen, Kinder, wird's was geben... Klar, denken jetzt die (fast immer) lieben Kleinen: Geschenke! Aber auch Grußkarten sind gefordert, denn ohne Fleiß, kein Preis! Und da kann man mit den Kindern zusammen echt was selbst machen. Schicke Karten, die beim Aufklappen ein dreidimensionales Innenleben präsentieren, sogenannte Popup-Karten, hinterlassen Eindruck. Und wenn die Großeltern dann auch noch per Soundmodul beim Aufklappen einen von den Enkeln gesprochenen Gruß erhalten, ist die weihnachtliche Stimmung perfekt. Selbst Geldgeschenke lassen sich so deutlich stilvoller überreichen.

Im Folgenden werden wir drei unterschiedliche Arten solcher 3D-Karten basteln: Eine einfache, vertikal gefaltete mit zentral symmetrisch angeordneten Aufklapp-Objekten, in diesem Fall Geschenkpakete (Bild 1).

Die zweite ist horizontal gefaltet und enthält Aufklapp-Objekte in mehreren Ebenen (Bild 2). Hier bringen wir einen zweigeteilten Schriftzug, einen kleinen Weihnachtsbaum und bei Bedarf auch einen Halter für ein Geldgeschenk unter.

Schließlich gibt es noch eine Karte mit Soundmodul (Bild 3), auf das Sie einen bis zu 20 Sekunden langen Gruß aufnehmen können. Solche Module gibt es einzeln im Online-Handel (für ca. 8 Euro). Deutlich preiswerter ist es aber, wenn Sie eine fertige Soundkarte kaufen und die umarbeiten (ab ca. 2,50 Euro) oder ein Modul aus einer gebrauchten Soundkarte ausbauen und wiederverwenden. Die darin befindliche Elektronik kann nämlich beliebig oft Ton aufnehmen. Und die Batterien halten meist recht lang, da das Modul nur sehr kurzzeitig in Betrieb ist.

Gedruckt werden die Karten mithilfe eines Tintenstrahl- oder Laserdruckers. Die Druckseiten, insbesondere die einzelnen Schnitt- und Faltlinien, müssen aber millimetergenau konstruiert werden. Doch keine Angst, der Einsatz eines komplizierten CAD-Programms ist dafür nicht erforderlich. Stattdessen verwenden wir hier eine Software, die Sie vielleicht sogar schon auf Ihrem Computer haben, aber noch nie benutzten: OpenOffice Draw. Sollten Sie dieses Büroprogramm-Paket noch nicht haben, finden Sie die Download-Adresse über den Kurzinfo-Link. Danach müssen Sie zur Installation lediglich die überspielte EXE-Datei starten, der Rest ist selbsterklärend. Das Programmpaket steht auch für Linux und Mac-Betriebssysteme zur Verfügung.

Das Programm bringt außerdem Weihnachts-relevante Symbole (auch Sterne genannt) mit, die damit ganz einfach in unterschiedlichen Größen und Farben gedruckt werden können. Oder Sie drucken lediglich die Umrisse und lassen die dann farblich vom Nachwuchs per Buntstift ausmalen.

Unsere Aufklappkarten bestehen immer aus zwei Seiten aus stärkerem Papier. Ich ver-

Kurzinfo

- » Einfache vertikal-gefaltete Karten
- » Karten mit mehreren Ebenen, auch für Geldgeschenke
- » Karten mit Sound und Papiertüten-Schneeflocken

Checkliste



Zeitaufwand:
ab einer halben Stunde



Kosten:
ab 0,10 Euro

Mehr zum Thema

- » Heinz Behling: Sterne aus der Tüte, Make 6/17, S. 66

Alles zum Artikel
im Web unter
make-magazin.de/xce4



Material

- » Papierbögen DIN A4 mit 160 oder 200g/m² weiß oder farbig
- » Papiertüten ca. 4,5 × 6 cm
- » Weihnachtliches Dekomaterial Sterne, Schneeflocken etc.
- » Klebestift
- » Soundmodul aus gebrauchter oder neuer Grußkarte

Werkzeug

- » Schere
- » Cutter und Schneidematte
- » Lineal
- » Farbstifte/Farbdrucker

wende Papier mit 200 g/m². Solches mit 160 g/m² sollte ebenfalls genügen, falls Ihr Drucker das stärkere nicht verarbeiten kann. Eine der beiden Seiten bildet die äußere Hülle der Karte mit dem Deckblattmotiv. Auf der zweiten Seite befinden sich die herausklappenden 3D-Objekte. Diese Seite muss daher mit Ausschnitten versehen werden, die ganz leicht mit einem Cutter (eine preiswerte Version wie in Bild 4 aus dem Discount-Laden reicht) und einem Lineal angefertigt werden können. Vorsicht: Diese Arbeit sollten Sie kleineren Kindern nicht überlassen, die Cutter sind scharf wie ein Skalpell!

Zuvor ist noch eine kleine Vorarbeit erforderlich: Finden Sie den Druckbereich Ihres Druckers heraus. Nur wenige Modelle können eine DIN-A4-Seite randlos bedrucken. Die meisten Geräte lassen an allen vier Kanten einige Millimeter frei. Wie groß diese Ränder sind, steht in

Bild 1: Die einfachste Weihnachtskarte ist ideal zum Basteln und Malen mit Kindern.



Bild 2: Mit Nachthimmel und Weihnachtsbaum erscheint hier auch noch ein Gruß in zwei Ebenen. Außerdem wurde hier ein Geldschein untergebracht.



den technischen Unterlagen des Geräts. Falls die nicht mehr zur Verfügung stehen, helfen meist die Internetseiten der Hersteller weiter. Einige Drucker arbeiten übrigens nicht symmetrisch, das heißt, die Ränder an den gegenüberliegenden Seiten sind nicht gleich breit (bei meinem Epson ist dies beispielsweise der Fall).

Einfache, vertikal gefaltete Karte

Damit kommen wir zur Bauanleitung der ersten Karte: Wie in Bild 1 gezeigt soll sie vertikal

in der Mitte gefaltet sein und innen drei übereinander gestapelte Geschenkpakete enthalten. Sie können aber auch andere Objekte verwenden. Einzige Voraussetzung bei dieser einfachen Karte: Die Objekte müssen vertikal achsensymmetrisch sein und an beiden Seiten senkrechte Kanten haben.

In OpenOffice Draw müssen Sie zunächst die Seite einrichten: Das geschieht unter Format/Seite. Setzen Sie dann die Ränder des Druckbereichs. Die Ränder sollten symmetrisch sein. Ist das bei Ihrem Drucker nicht der Fall, dann setzen Sie die vier Werte auf den Wert des größten Randes, den Ihre Drucker an

einer der vier Seiten macht. In meinem Beispiel ist das 1 cm (Bild 5).

Da die gefaltete Karte später in einen Briefumschlag (Format DIN lang) passen soll, darf nicht die gesamte Höhe der A4-Seite verwendet werden. Die Höhe wird daher auf 21 cm eingestellt. Zusammen mit den Rändern gibt das eine Druckfläche mit den Maßen 19 x 19 cm, auf der wir unsere Karte konstruieren können. Mit OK bestätigen Sie die Einstellung.

Jetzt setzen wir die erste Faltlinie, die Mitte. Dazu wählen wir zuerst die Linien-Funktion mit dem zweiten Symbol von links unten im OpenOffice-Fenster (Bild 6).

Jetzt könnte man mit der Maus die Linie millimetergenau in die Zeichnung einsetzen. Aber das ist viel zu kompliziert und kaum genau zu machen. Es gibt einen einfacheren Weg. Zeichnen Sie auf der Seite eine kurze vertikale Linie. Dazu die Maus auf den Startpunkt setzen, Maustaste gedrückt halten und bei ebenfalls gedrückter Shift-Taste bis zum Endpunkt führen, bevor Sie die Tasten wieder loslassen. Durch die zusätzliche Shift-Taste wird die Linie automatisch senkrecht! Anschließend setzen Sie den Mauszeiger auf diese Linie, klicken mit rechts und wählen Position und Größe.

Zuvor erst einmal ein paar Gedanken dazu, wo diese Linie hin muss und wie lang sie werden soll: Die Länge oder besser gesagt Höhe ist klar: Der Druckbereich ist 19 cm hoch. So lang muss auch die Linie sein. Also tragen wir bei Höhe 19 cm ein.

Zur Position: Sie soll in der Mitte der Druckfläche vom oberen bis zum unteren Rand führen. Daraus ergeben sich die X-(waagerechte) und Y-(senkrechte) Koordinaten des Linienstartpunkts: 9,50 cm für X und 0 cm für Y (Bild 7). Achten Sie dabei darauf, dass die Basispunkte jeweils oben links sitzen.

Bei unseren Karten werden wir drei verschiedene Linienarten zeichnen: Schnittlinien, Tallinien (an denen das Papier wie bei der Mittellinie nach oben gefaltet wird) und Berglinien (hier wird nach unten, von der Linie weg gefaltet). Damit wir das später auseinanderhalten können, geben wir ihnen unterschiedliche Farben. Die Tallinien habe ich türkis (cyan) gefärbt (Klick mit rechts auf die Linie, im Menü Linie wählen und per Klick das Farbfeld cyan auswählen).

An dieser Stelle sollten Sie die Karte das erste Mal speichern, denn diesen Kartenrohbau brauchen Sie später noch für das Deckblatt der Karte, also die äußere Hülle. Wählen Sie daher einen entsprechenden Namen, zum Beispiel „Geschenke_aussen“.

Jetzt können wir auch schon mit dem Zeichnen der später herausklappenden Geschenke beginnen. Es soll ja wie in Bild 1 ein Stapel von drei unterschiedlich großen, würfelförmigen Paketen werden. Ich beschreibe hier ausführlich die Konstruktion des ersten,

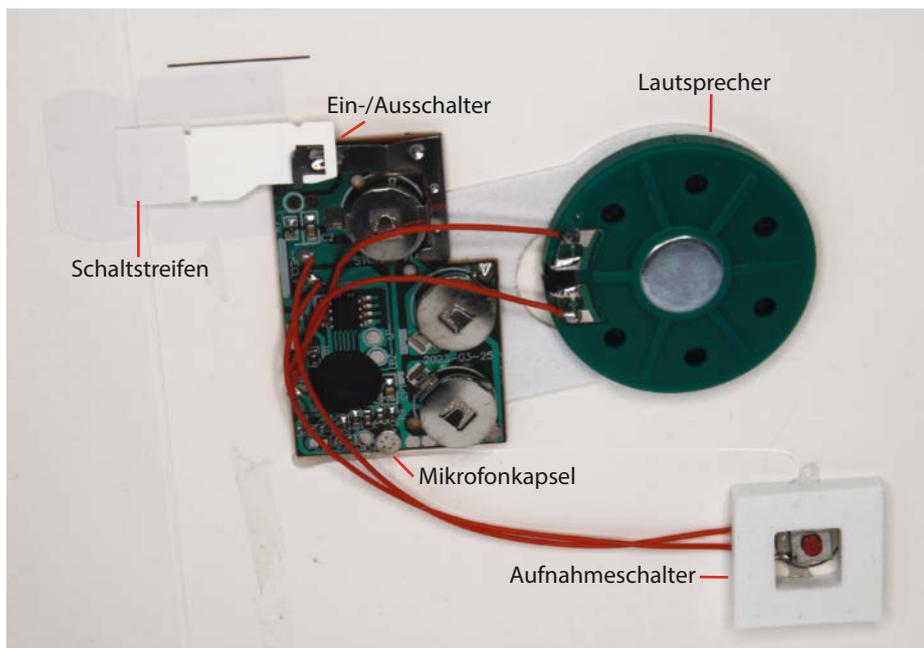


Bild 3: Erstaunlich, wie viel Elektronik in einer 2,50-Euro-Soundkarte steckt.

für die anderen gebe ich Ihnen dann nur noch die Positionen und Längen an.

Das obere Paket soll einen Abstand von 2 cm vom oberen Rand und eine Kantenlänge von ebenfalls 2 cm haben. Wir brauchen also zwei waagerechte Schnittlinien mit jeweils 4 cm Länge (zweimal die Kantenlänge, da ja zwei Würfelflächen aus der Karte herausklappen sollen) und zwei senkrechte Tallinien mit je 2 cm Länge. Das hört sich jetzt kompliziert an, ist mit OpenOffice Draw aber einfach zu konstruieren: Für die erste (obere) Schnittlinie zeichnen Sie zunächst eine waagerechte Linie an beliebiger Stelle. Dann stellen Sie wie bereits beschrieben deren Position ein. X ergibt sich aus Seitenmitte (9,5 cm) minus der Kantenlänge des Würfels (2 cm), also insgesamt 7,5 cm. Y entspricht dem Abstand vom oberen Druckrand, also 2 cm. Die Länge (Breite) der Linie muss der doppelten Kantenlänge des Würfels entsprechen, also 4 cm. Und da es sich hier um eine Schnittlinie handelt, lassen wir deren Farbe auf schwarz.

Die untere Schnittlinie ist noch einfacher zu zeichnen, denn wir kopieren einfach die gerade gezeichnete Linie (anklicken, STRG+C und anschließend STRG+V tippen) und schieben Sie mit der Cursor-Taste nach unten. Jeder Druck der Cursor-Taste verschiebt die Linie um 0,1 cm. Es muss also 20mal getippt werden. Falls Sie sich nicht sicher sind: Linie mit rechts anklicken, Position und Größe wählen und kontrollieren. Ihre Karte sollte nun aussehen wie in Bild 8.

Nun brauchen wir noch drei Falmlinien: Zwei Tallinien verbinden jeweils die Enden der beiden Schnittlinien, und eine Berglinie in der Mitte des Würfels. Zunächst die Täler: Zeichnen Sie eine kurze senkrechte Linie. Im Einstellungsfenster geben Sie dann als Startpunkt dieselben Koordinaten wie die der oberen Schnittlinie ein (X=7,5 cm, Y=2 cm). Die Höhe beträgt 2 cm (Kantenlänge des Würfels). Und da es sich um eine Tallinie handelt, setzen Sie deren Farbe auf cyan. Die rechte Tallinie kopieren Sie wie bereits beschrieben und setzen Sie mit der Cursortaste um 4 cm (40mal tippen) nach rechts. Sieht schon fast fertig aus (Bild 9).

Aber nur fast, denn die Mittellinie erscheint hier noch als Tallinie, sie muss aber eine Berglinie sein. Sie müssen also einen Teil der ganz zu Beginn angelegten Mittel-Linie durch eine Berglinie überdecken. Auch das ist wieder schnell erledigt: Kopieren Sie nochmal eine der beiden senkrechten Tallinien und schieben Sie sie in die Mitte (20mal Cursortaste). Ändern Sie dann deren Farbe. Ich habe für Berglinien rot gewählt. Schon haben Sie aus einem Tal einen Berg gemacht (Bild 10). Speichern Sie nun die Karte per Datei/ Speichern unter und wählen Sie einen anderen Dateinamen, zum Beispiel „Geschenke_innen“, damit Sie die Vorlage für das Deckblatt nicht überschreiben.



Bild 4: Solch ein preiswerter Cutter ist völlig ausreichend.

Damit sind die Linien des ersten Würfels fertig. Der zweite soll direkt darunter angeordnet werden und 4 cm Kantenlänge haben. Die Werte für seine obere Schnittlinie sind daher: X = 5,5 cm (Seitenmitte minus Kantenlänge), Y = 4 cm (Abstand des ersten Würfels von der Oberkante plus Höhe des ersten Würfels) und Breite = 8 cm (2mal Kantenlänge). Die untere kopieren Sie aus der ersten und setzen sie 4

cm nach unten. Die drei Falzlinien zeichnen sie analog zum ersten Würfel.

Der dritte Würfel hat 5 cm Kantenlänge und wird genauso angelegt (Startpunkt erste Schnittlinie X = 4,5 cm, Y = 8 cm, Breite = 16 cm). Das sieht schließlich aus wie in Bild 11.

Falls Sie die Karten später selbst oder von Ihren Kindern farbig ausschmücken möchten, sind Sie nun bereits fertig. Sie können das aber

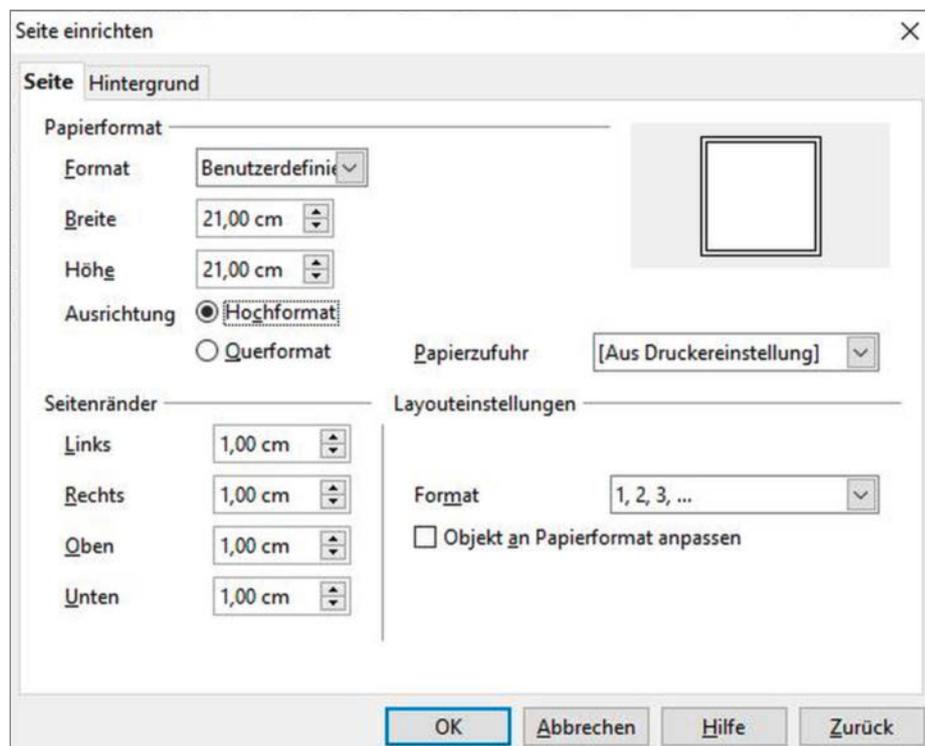


Bild 5: Die vier Ränder sollten gleich breit eingestellt werden. Das wirkt später optisch besser.



Bild 6: Die Linienfunktion steckt hinter der schräggestellten Linie.

auch mit OpenOffice Draw machen und gleich mit drucken. Ich zeige Ihnen dafür, wie Sie das Design meiner Karte anlegen.

Zunächst färben wir das obere Paket rot. Dazu werden wir ein Rechteck innerhalb der äußeren Linien des Würfels zeichnen, das wir dann entsprechend einfärben. Das Symbol für die Rechteck-Funktion finden Sie neben dem

Linien-Symbol. Zeichnen Sie ein beliebiges Rechteck und klicken Sie dann mit der rechten Maustaste darauf. Auch hier geht es jetzt mit Position und Größe weiter. Die Position entspricht dem Startpunkt der oberen Linie des ersten Würfels ($x = 7,5 \text{ cm}, Y = 2 \text{ cm}$). Die Breite des Rechtecks muss 4 cm, seine Höhe 2 cm betragen.

Jetzt überdecken aber die Begrenzungslinien des Rechtecks unsere bereits angelegten Linien. Das können Sie ändern: Klicken Sie mit rechts auf das Rechteck, wählen Sie Linie und stellen im nächsten Fenster die Transparenz auf 100 Prozent. OK zur Bestätigung, fertig.

Für die Farbeinstellung des Rechtecks klicken Sie erneut mit rechts darauf, dann auf

Fläche und wählen Sie die gewünschte Kolorierung, im Beispiel rot. Diesmal verdeckt die Farbe die mittlere Falzlinie. Das lösen Sie ge- konnt per Rechtsklick aufs Rechteck, Anord- nung und Ganz nach hinten.

Auf dieselbe Art färben sie die beiden an- deren Würfel. Schließlich sieht das ganze aus wie in Bild 12.

Lassen Sie nun Ihrer Fantasie freien Lauf und schmücken Sie Ihr Werk noch mit Sternen oder anderen Symbolen und Schriftzügen. Die ent- sprechenden Funktionen finden Sie wieder in der Symbolleiste unten im Draw-Feld. Klicken Sie auch einmal auf die kleinen Pfeile neben den Symbolen. Dann erscheinen weitere Motive wie etwa Herzen u. ä. Sind Sie mit dem Ergebnis zufrieden, speichern Sie die Karte und drucken Sie sie. Das Papier sollte dicker als gewöhnliches Briefpapier sein. Mit Papieren mit 160 oder 200 g/m² habe ich die besten Erfahrungen gemacht. Auch Fotopapier kann verwendet werden. Es sieht besser aus, ist aber etwas schwieriger zu falten. Die Druckqualität sollten Sie möglichst hoch einstellen.

Anschließend geht es ans Schneiden: Mittels Cutter und Lineal schneiden Sie alle schwarzen Linien. Achtung: Die Linien sind nur schwach zu erkennen, sie sollen ja schließlich nicht das Design stören. Danach fahren Sie mit dem stumpfen Rücken des Cutters einmal über jede Falzlinie. Das macht diese Bereiche geschmei- dig, sodass sie sich besser falten lassen. Schließ- lich falten Sie die Karten entsprechend den Berg- und Tallinien in die jeweilige Richtung. Das Innenteil ist damit fertig (Bild 13).

Anschließend geht es an die Gestaltung des Deckblatts: Öffnen Sie dazu in OpenOffice Draw die zuerst gespeicherte Version der Karte namens Geschenke_aussen. Als erstes ändern Sie die Farbe der Falzlinie: Auf der Aussenseite ist es ja keine Tal- sondern eine Berglinie. Setzen Sie die Farbe daher auf rot.

Die Außenseite enthält bislang nur die Mittellinie, sonst (noch) nichts. Das können Sie nach Belieben gestalten. Wie wäre es mit einem weihnachtlichen Bild von Ihnen? Setzen Sie Ihrer Kreativität keine Grenzen. Falls Sie ein Bild verwenden: Setzen Sie es nur auf die rechte Seite der Karte. Sie wird ja vertikal zugeklappt. Die Rückseite ist also zusammengeklappt nicht sichtbar. Fotos können Sie mit „Einfügen/Bild/ Aus Datei“ einsetzen. Wählen Sie dann die ent- sprechende Datei und positionieren Sie diese auf der Karte. Sollte das Bild zu groß sein, kön- nen Sie unerwünschte Bereiche mit weißen Rechtecken abdecken (Bild 14).

Speichern Sie Ihr Werk und drucken Sie es. Falten Sie danach die Mittellinie mit dem Rü- cken des Cutters und falten Sie die Seite. Der Zeitpunkt naht, wo beide Teile der Karte ver- eint werden. Mithilfe von Klebestiften werden sie zusammengefügt. Tragen Sie den Klebstoff auf die Rückseite des Innenteils auf. Achten Sie darauf, dass kein Klebstoff auf die heraus-

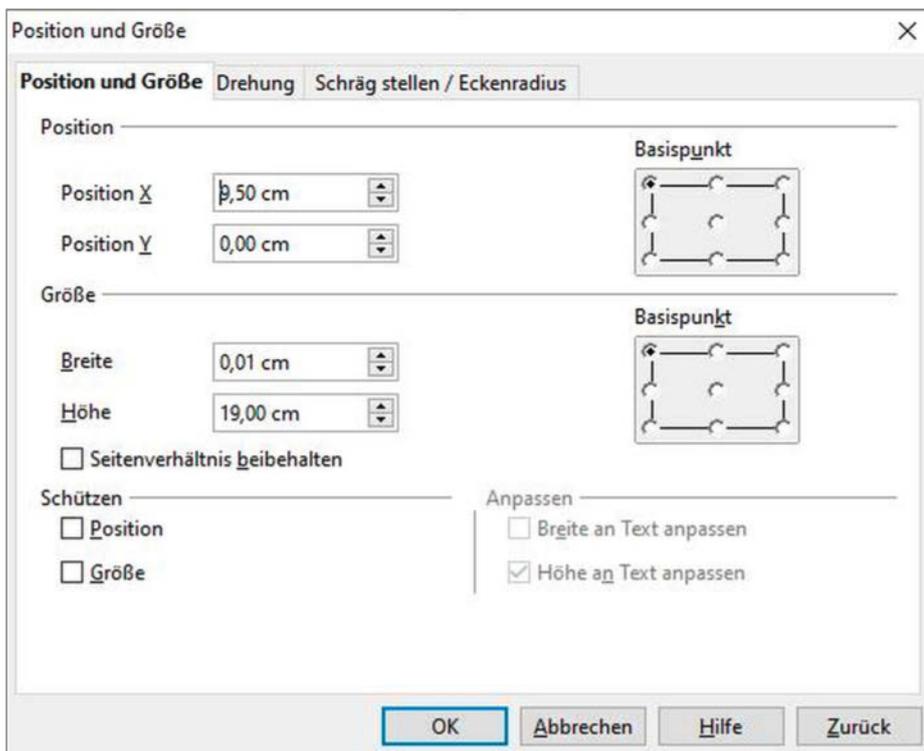


Bild 7: Die Werte für die mittlere Falzlinie

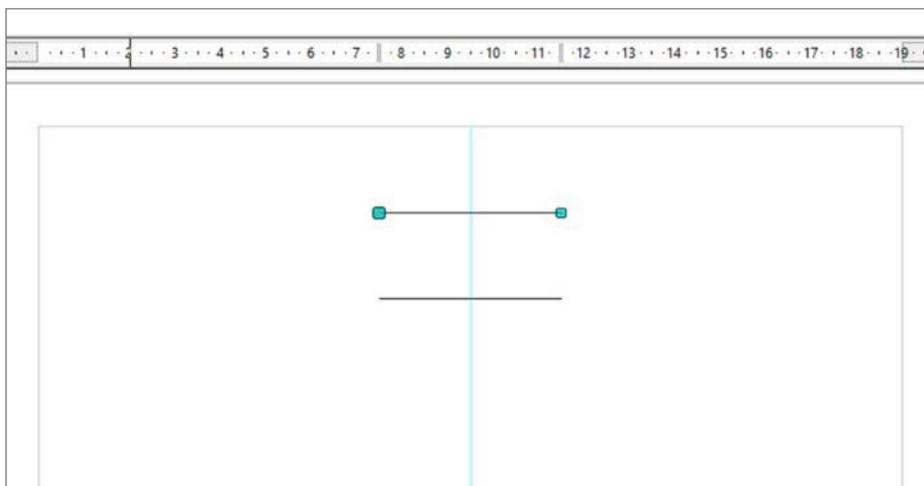


Bild 8: Die beiden Schnitlinien des oberen Geschenkewürfels



Bild 9: Der erste Würfel ist fast fertig, aber da stimmt etwas noch nicht.

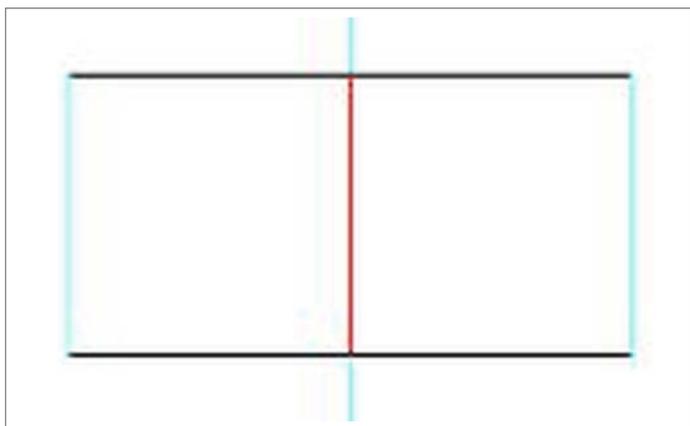


Bild 10: Die zuletzt gezeichnete überdeckt die bereits vorhandene Linie.

klappbaren Teile gelangt. Danach legen Sie den Innenteil mit dem Klebstoff auf die Rückseite des Außenteils. Drücken Sie sie fest und falten Sie die Karte noch vor dem Austrocknen des Klebers einmal zusammen. So zusammengeklappt legen Sie einen schweren Gegenstand (Buch) auf die Karte und lassen den Kleber trocknen. Anschließend schneiden Sie die Karte auf Maß, entfernen also den überflüssigen unteren Teil, sodass die Außenmaße

der Karte 21 x 21 cm betragen. So passt sie in einen Briefumschlag und ist einsatzbereit.

Mehr-Ebenen-Karte

Eine Karte, auf der die herausklappenden Objekte in mehreren Ebenen stehen, können Sie ebenfalls mit OpenOffice Draw konstruieren. Hier habe ich übrigens die komplette DIN-A4-Seite benutzt. Durch die Querfaltung passt diese

Karte dann nicht mehr in einen DL-Umschlag, stattdessen muss ein C5-Kuvert benutzt werden.

Die Vorgehensweise bei der Konstruktion ist dieselbe, allerdings ist bei den Maßen und der Positionierung einiges zu beachten. Außerdem sollten solche Karten nur horizontal geklappt werden, da nur dann der richtige 3D-Effekt entsteht. In Bild 15 habe ich Ihnen den Innenteil der in Bild 2 gezeigten Karte detailliert dargestellt inklusive der Bemaßungen.



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



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



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

Experimentieren,
basteln, bauen:
Zeit für dein
Feiertagsprojekt!



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



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



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



2022 · 240 Seiten · 24,90 €
ISBN 978-3-86490-908-5

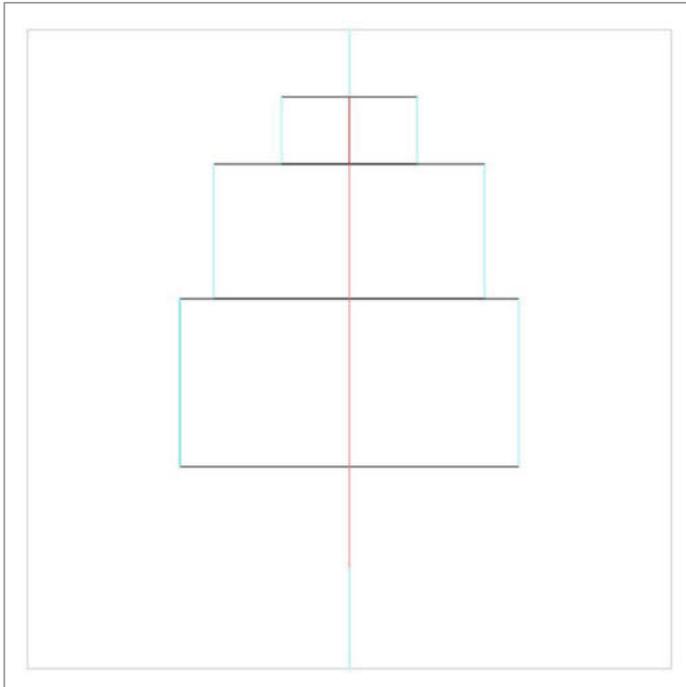


Bild 11: Die Linien aller drei Würfel

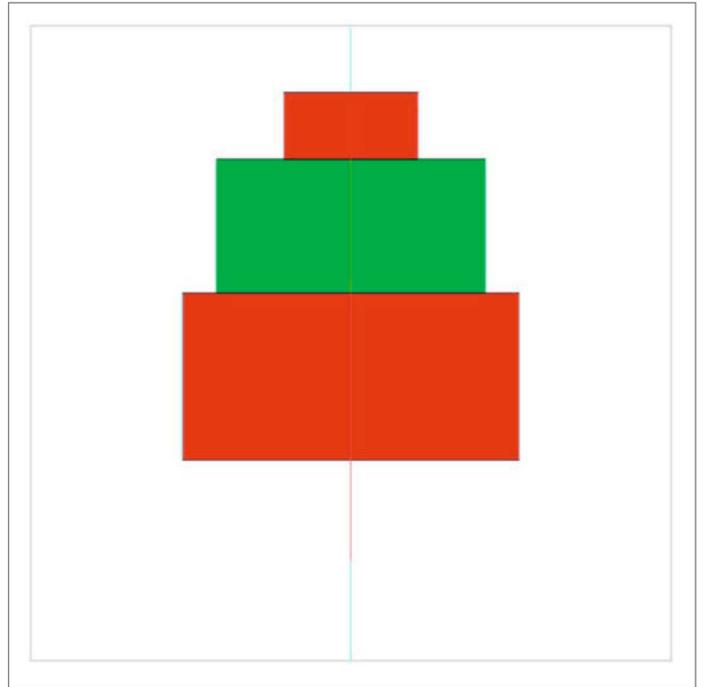


Bild 12: Der Würfelstapel in Farbe und demnächst 3D

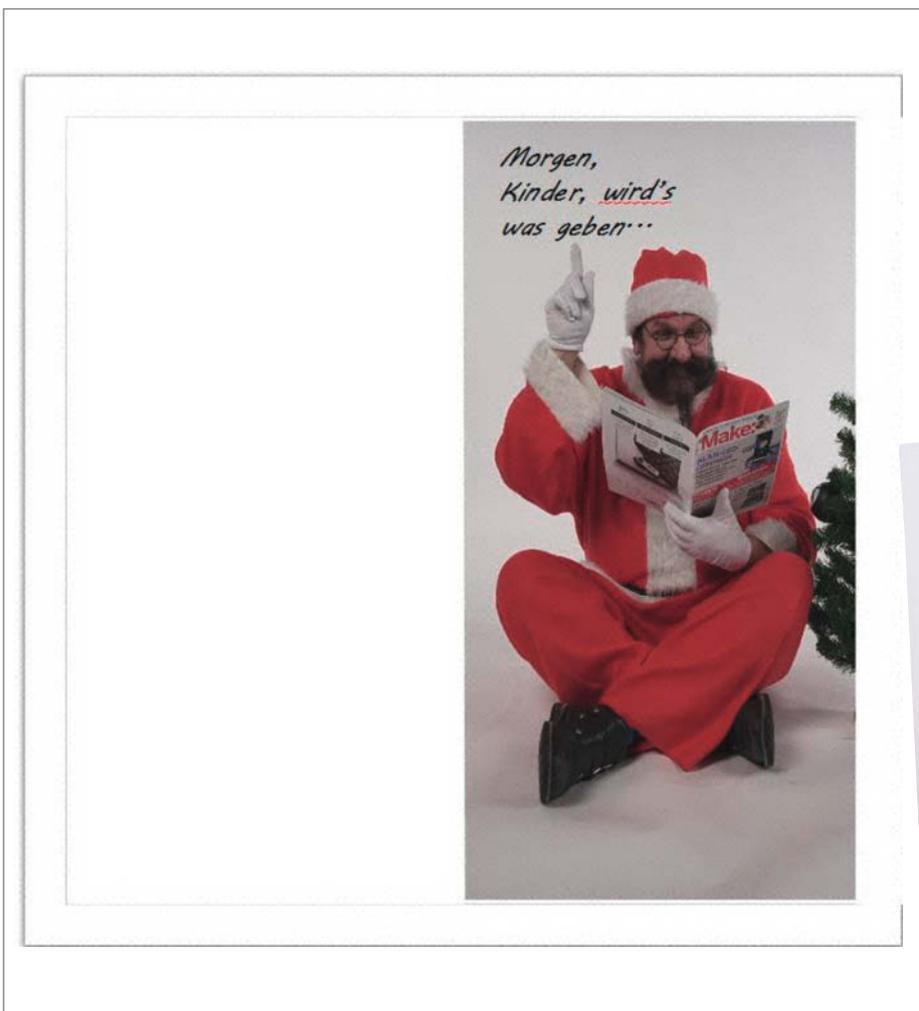
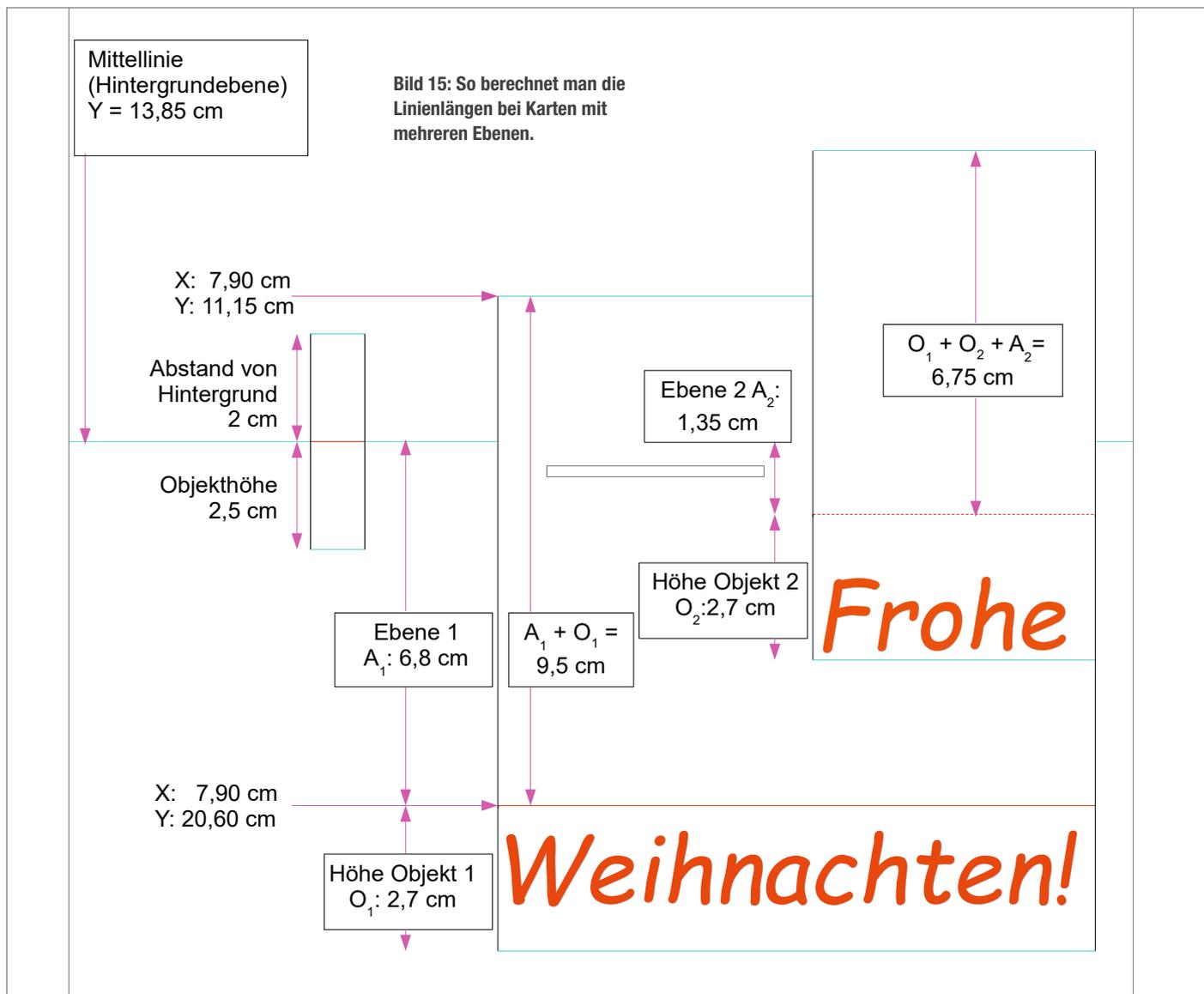


Bild 14: Die Außenseite der Karte

Online (siehe Kurzinfo-Link) zeige ich Ihnen noch eine detailliertere Anleitung für die Mehrebenen-Karte. Übrigens enthält diese Karte links oberhalb des Schriftzugs „Frohe“ einen 2 x 40 mm großen Schlitz. Der ist für Geldgeschenke vorgesehen. Möchten Sie die Banknoten nicht einfach nur zusammenrollen, sondern etwas mehr gestalten, finden Sie unter den Links auch einige Anleitungen, wie man Geldscheine beispielsweise zu kleinen Tannenbäumen faltet. Ist kein Bargeschenk vorgesehen, lassen Sie diesen Schlitz einfach weg.



Bild 13: Das Innenteil der Karte geschnitten und gefaltet



TECHNIKUNTERRICHT MACHT ENDLICH SPÄß!



Make:Education

Mit **Make Education** erhalten Sie jeden Monat kostenlose Bauberichte und Schritt-für-Schritt-Anleitungen für einen praxisorientierten Unterricht:



Für alle weiterführenden Schulen



Fächerübergreifend



Digital zum Downloaden



Monatlicher Newsletter

Jetzt kostenlos downloaden:

make-magazin.de/education

Bild 16: Solche Pergamenttütchen kosten nur etwa 2 Euro für 25 Stück.



Online gibt es zum Download auch die fertigen OpenOffice-Dateien für die beiden hier konstruierten Karten.

Tüten-Schneeflocke mit Sound

Für die Soundkarte können Sie die Elektronik aus gebrauchten Grußkarten (siehe Bild 3) verwenden. Sie müssen sie nur vorsichtig inklusive des meist als Papier- oder Kunststoffstreifen gebauten Ein-/Aus-Schalters ausbauen und in gleicher Weise in Ihre eigene Karte einsetzen. Dazu müssen Sie auf der Rückseite mithilfe eines 3-mm-Locheisens einige Schallöffnungen für den Lautsprecher einfügen. Das ist durchaus machbar und aus Sicht der Müllvermeidung auch empfehlenswert.

Einfacher geht es aber mit fertig gekauften, aufnahmefähigen Soundkarten, die im Moment (Stand Anfang November 23) von einem Billigladen für 2,50 Euro angeboten werden (Bezugsquelle siehe Kurzinfor-Link). Aber auch diese Karten kann man pimpen: Wie wäre beispielsweise eine Schneeflocke, die sich beim Öffnen der Karte entfaltet? Wenn Sie oder Ihr Nachwuchs dann dazu auch noch „Leise rieselt der Schnee“ aufs Soundmodul aufnehmen (einfach die Karte aufschlagen, den Taster der Karte drücken und Sie können 20 Sekunden Ton aufnehmen), wird der oder die Beschenkte sicher zu Tränen gerührt sein.

Die Schneeflocke fertigen wir aus kleinen Pergament-Tütchen in der Größe 4,5 x 6 cm (Bild 16). Das kann man wieder gut mit dem bastelwütigen Nachwuchs erledigen.

Sie brauchen für eine Schneeflocke etwa acht bis zehn Tütchen. Tragen Sie am unteren geschlossenen Rand und an der Mittelnah der Tüten jeweils eine dünne(!) Linie mit dem Klebestift auf und legen Sie die Tüten übereinander. Achten Sie darauf, dass die Tüten genau übereinander liegen. Danach gehen Sie dem Tütenstapel mit einer kräftigen Schere an die Seiten. Schneiden Sie Spitzen und Dreiecke aus. Dazu braucht man allerdings ziemlich viel Kraft. Also sollten Sie diesen Schritt besser nicht den Kindern überlassen. Die Schnitte ergeben die Spitzen und Zacken der Schneeflocke. Sobald Sie sie in die Mitte der Karte eingeklebt haben, öffnet sie sich beim Aufklappen und leise rieselt der Schnee (Bild 17).

Übrigens gibt es ähnliches auch im Laden bei der Weihnachtsdekoration. Meist sind es aber aufklappbare Weihnachtsbäume. Auch die eignen sich prima, um Karten aufzupeppen (Bild 18).

Ich hoffe, Sie können diesem Artikel Anregungen für eigene Bastel-Karten abgewinnen. Übrigens: Auch die Make-Redaktion ist empfänglich für Weihnachts-Grußkarten. Die Adresse finden Sie im Impressum. Und damit wünsche ich Ihnen allen fröhliche Weihnachten und Frieden auf Erden für alle Menschen, welchem Glauben sie auch angehören mögen. —hgb

Bild 17: Der Schnee kann kommen.



Bild 18: Hier sollten Sie eher „Oh Tannenbaum“ singen.



secIT by heise

HANNOVER 2024



meet.
learn.
protect.

05. - 07. MÄRZ 2024, HANNOVER

Die Kongress- messe für Security-Profis



Bis 31.12.2023
kostenloses Sponsored-Ticket
sichern oder von unseren
Frühbucherrabatten profitieren.

secit-heise.de

© Copyright by Maker Media GmbH.

Offizieller Eventpartner



Ab sofort: Alle Online-Artikel auf der Make-Webseite

Wer die Make im Abo bezieht und seinen kostenlosen Account bei heise online mit seinen Abo-Daten verknüpft, kann bereits seit rund einem Jahr alle Artikel aus der Make-Redaktion ohne Aufpreis lesen, die hinter der Bezahlschranke von heise+ liegen. Wer das noch nicht wusste oder bisher die Möglichkeit nicht genutzt hat, findet über den Link eine ausführliche Anleitung dazu, wie man diese Verknüpfung einrichtet.

Das lohnt sich jetzt besonders, denn ab dem Verkaufsstart dieses Hefts am 8. Dezember findet ihr alle heise+-Artikel aus der

Make-Redaktion auch direkt in der Liste unserer Online-Meldungen auf unserer Webseite www.heise.de/make. Der Vorteil für Make-Abonnentinnen und Abonnenten: Dort erscheinen manche Artikel früher als im Heft – so konnte man unsere Tests zu Bausätzen für Maker (ab Seite 22 in diesem Heft) etwas ausführlicher und bereits eine Woche früher online lesen und zeitiger die Wunsch-Kits einkaufen. Doch nicht nur Heft-Artikel erscheinen dort. Wir veröffentlichen auch regelmäßig Projektbeschreibungen und Anleitungen, die man in keiner

gedruckten Make-Ausgabe findet: in der Vergangenheit etwa eine dreiteilige Serie zum Eigenbau einer Solathermie-Dusche, den Test eines Bausatz-Synthesizers oder eine Anleitung, wie man Explosionszeichnungen mit der kostenlosen Software FreeCAD erstellt. Und für die Vorweihnachtszeit bereiten wir gerade noch den einen oder anderen DIY-Geschenktipp vor. —pek

► make-magazin.de/xrga

YouTube: Von Carsten Meyer löten lernen

In der Make-Ausgabe 5/23 haben wir an dieser Stelle über unseren Video-Redakteur Johannes Börnsen geschrieben: „Unterstützt wird er bei seinen Videos von den anderen Redaktionsmitgliedern, von denen Johannes sich gerne erklären lässt, was er selbst nicht weiß.“ Nun ist es keine Schande, manches (noch) nicht zu wissen, oder zumindest weniger als andere – ganz besonders, wenn der Lehrmeister eine Koryphäe wie unser ehemaliger und inzwischen seinen wohlverdienten Ruhestand genießender Kollege Carsten Meyer ist.

Den hat Johannes jüngst besucht, um von ihm richtig löten zu lernen. Herausgekommen ist das mit Abstand bisher längste Video auf unserem YouTube-Kanal: Fast eine Stunde lang zeigt Carsten Tricks und gibt Tipps zum Löten, speziell von winzigen SMD-Bauteilen,

die für viele Maker immer noch Angstgegner sind. Wie klappt es bei „normalen“ SMD-Widerständen, wie bei einem Chip mit einem Beinchenabstand von nur 0,4 Millimetern? Und was tun, wenn man aus Versehen doch mal ein Beinchen verbogen hat oder ein Bauteil wieder auslöten muss? Die Antworten gibt es im Video, zu erreichen über den Link unten.

Neue Videos auf unserem YouTube-Kanal erscheinen weiterhin laufend und immer freitags. Und sie sind stets kostenlos und ohne Abo abrufbar.

—pek

► www.youtube.com/@MakeMagazinDE



Einblick in die Redaktion: Make auf WhatsApp

Einen direkten Einblick in die Arbeit der Make-Redaktion ist jetzt auch über WhatsApp möglich: In der Übersicht der Messenger-App auf „Aktuelles“ tippen und danach nochmal neben „Kanäle suchen“ auf „Alle anzeigen“. Dann auf die Lupe und lostippen – die Eingabe der vier Buchstaben „Make“ sollte derzeit reichen, um uns zu finden (mit vollem Namen firmieren wir unter „Make Magazin Deutschland“). Nach einem letzten Tipp auf das Pluszeichen ist unser Kanal dann abonniert und bietet einen tieferen Blick hinter die Kulissen der Produktion unseres Hefts, der Online-Artikel und YouTube-Videos, den man auf andere Weise nicht bekommt. Wer besonders dicht

dran sein will, tippt in der Kanalsicht oben auf die durchgestrichene Glocke, um die Benachrichtigungen zu aktivieren.

Wir nutzen hier eine ziemlich frische Funktion von WhatsApp, die erst im Oktober überhaupt freigeschaltet wurde und sind gespannt auf unsere und eure Erfahrungen damit. Aktuell können wir Bilder und Videos senden und ihr könnt mit Emojis darauf reagieren, eine direkte Kontaktaufnahme ist (derzeit) nicht möglich. Dafür bleibt euer Abo komplett anonym, wir haben keinerlei Zugriff auf Daten, wer unserem Kanal folgt. Wir wissen nur, dass bereits fast 2000 Nutzerinnen und Nutzer dabei sind. —pek



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



DIY-TV-Beleuchtung

Filme, die über den Fernseher hinaus erstrahlen: Mit einem Raspberry Pi und LED-Leisten wird Fernsehen noch immersiver.

von Mike Senese



Fernsehgeräte sind heutzutage technologische Wunderwerke – riesige, aber hauchdünne Bildschirme mit kristallklarer 4K-Auflösung, Bildwiederholraten von bis zu 120 Hz, HDR-Farb- und Luminanzbereich und anderen Funktionen, die Wohnzimmer in persönliche Kinos verwandeln. So großartig diese neuen Fernseher auch sind, die Magie endet immer noch an der dünnen Blende, die den Bildschirm einrahmt. Aber was wäre, wenn man Sendungen, Filme oder Spiele über den Fernseher hinaus ausstrahlen könnte?

Mit einem Raspberry Pi, einer HDMI-Capture-Card und einem Streifen mit einzeln adressierbaren RGB-LEDs können Sie genau das tun. Die Hardware verarbeitet die Videoinhalte, erkennt die spezifischen Farben, die den Rändern des Bildschirms am nächsten sind, und steuert dann die LEDs so an, dass sie die gleichen Farben hinter dem Fernseher

Kurzinfo

- » Offene statt proprietäre Lösung wie bei kommerziellen Ambient Lighting-Produkten
- » Konfiguration über Weboberfläche
- » Individuell anpassbar

Checkliste

-  **Zeitaufwand:**
2 Stunden
-  **Kosten:**
120 Euro

Werkzeug

- » Seitenschneider oder Abisolierzange
- » Klebestreifen doppelseitig

Mehr zum Thema

- » Daniel Springwald, Der LavaFrame, Make 3/23, S. 36
- » Carsten Wartmann, Disco is back – Lichtshow mit WLED, Make 2/23, S. 8

Alles zum Artikel
im Web unter
make-magazin.de/xfs2



nach außen projizieren. All dies geschieht dynamisch und in Echtzeit, wodurch ein toller immersiver Beleuchtungseffekt entsteht.

Diese Beleuchtung wird als „Ambient Lighting“ oder „Bias Lighting“ bezeichnet. Es gibt eine Reihe von All-in-One-Ambient-Beleuchtungssystemen – beispielsweise von Philips, die diesen Effekt bieten, aber sie sind teuer und proprietär. Zum Glück ist es nicht schwer, ein eigenes System zusammenzustellen.

Die Raspberry-Pi-Hardware

Damit die LED-Hintergrundbeleuchtung die Farbtöne anzeigt, die auf dem Bildschirm zu sehen sind, greift man das Videosignal von einem externen HDMI-Gerät (Apple TV, Chromecast, Roku usw.) ab. Es wird dann nicht nur per HDMI an den Fernsehbildschirm weitergeleitet, wo es normal angezeigt wird, sondern auch per USB an den Raspberry Pi geschickt, wo die HyperHDR-Software das Signal auf die Videoinformationen am äußeren Rand des Fernsehbildschirms reduziert und dieses Signal dann an die LED-Lichtleiste sendet.

Vorbereiten der LEDs

Auf dem Markt gibt es eine Vielzahl von LED-Lichtleisten, und HyperHDR kann mit den meisten von ihnen umgehen. Über die Hue-Bridge können sogar externe Philips Hue-Lampen ferngesteuert werden, sodass man zusätzliche Lampen neben dem Fernseher platzieren kann, um eine noch breitere Streuung des Umgebungslichts zu erzielen.

Für meine Konstruktion habe ich WS2812B-LED-Streifen verwendet. Sie sind billig und werden mit 5 V betrieben, sodass man sogar einen Stromkreis bauen kann, der sie und den Raspberry Pi mit nur einem Stecker versorgt. Ich habe eine 5-Meter-Lichtleiste mit 60 LEDs pro Meter gekauft. Mehr LEDs bedeuten feinere Übergänge zwischen den Farben. Allerdings bedeuten mehr LEDs auch einen höheren Stromverbrauch. Es gibt auch Vorteile bei der Verwendung von 12-Volt-Lichtbändern wie der WS2815-Version. Diese leiden nicht unter Spannungsabfällen bei größeren Längen, wie es bei den 5-Volt-Leuchten der Fall sein kann. Aber sie bieten nicht die Möglichkeit, den Pi über dieselbe Stromquelle zu betreiben.

Apropos Stromquelle: Sie brauchen eine, die für die Beleuchtung all dieser LEDs ausreicht. Wir sprechen hier von 150 bis 300 LEDs, die möglicherweise ein rein weißes Licht ausstrahlen, was bedeutet, dass sie einen hohen maximalen Stromverbrauch haben können. Ein normales Handy-Ladegerät reicht dafür nicht aus. Man braucht ein Netzteil, wie es für Laptops genutzt wird, das der Spannung der Lichterkette entspricht und eine ausreichende Stromstärke hat. Meins hat 5 V/10 A, also 50 Watt Leistung, aber ich würde zu einem stärkeren raten, das bis zu 300 W leisten kann, nur zur Sicherheit. Wer es genau wissen will, findet über die URL in der Kurzinfo einen Online-Rechner für den Energiebedarf von LED-Streifen.

Um die LED-Lichtleiste an die Stromquelle anzuschließen, verwende ich einen 5-V-Buchsenadapter. Einige Netzteile sind bereits mit



Die Capture-Card greift das Videosignal ab und sendet es per USB an den Raspberry Pi.

einem solchen Adapter ausgestattet, man kann ihn aber auch nachkaufen. Die Adapter werden normalerweise in einem kleinen Set geliefert. Man muss darauf achten, dass die „weibliche“ Seite dabei ist. Man entfernt ein kleines Stückchen Isolierung von den Stromkabeln (Rot und Weiß) der LED-Leuchten und schließen sie diese an die Schraubklemmen der Buchse an. Das rote Kabel geht an den Pluspol (+), das schwarze oder weiße Kabel an den Minuspol (-).

Verbinden Sie schließlich mit einem männlich/weiblich Überbrückungskabel die Datenleitung des LED-Streifens (bei den meisten Streifen der mittlere Pin) mit GPIO-Pin 18 des Raspberry Pi (der sechste Pin von der rechten unteren Ecke; Pin 12 in Pin-Out-Diagrammen), wie in der Abbildung gezeigt.

Wichtig: Wenn der Raspberry Pi separat mit Strom versorgt wird, muss man auch eine Brücke zwischen einem der GND-Pins am Raspberry Pi und dem Erdungsdraht (Weiß) des LED-Streifens setzen, sonst wird es nicht funktionieren. Ein Ground-Pin liegt direkt auf dem siebten Pin von rechts unten, links neben dem Signal-Pin in der Abbildung (Pin 14 in Pin-Out-Diagrammen.)

Jetzt schließt man das externe HDMI-Gerät an die HDMI-Capture-Karte an, ein HDMI-Kabel von dort an den Fernseher und ein USB-Kabel an einen USB-Anschluss des Raspberry Pi. Die USB-Buchse meines Raspberry Pi versorgt die Capture-Karte mit Strom.

Wenn alles mit Strom versorgt und der richtige HDMI-Eingang gewählt ist, sollte jetzt der normale Inhalt auf dem Fernsehbildschirm zu sehen sein.



Die Stromkabel der LED-Ketten sind doppelt herausgeführt: einmal in einen Stecker und einmal lose. Die losen Enden kommen an den 5-V-Stromstecker.

Montieren der LEDs

Um ein optimales Ergebnis zu erzielen, sollte man ein Rechteck aus LED-Leuchten um die hintere Kante des Bildschirms herum und in der Nähe der Seiten anbringen. Manche LED-Leisten verfügen über eine Klebefläche und können mit dem eingebauten Klebstoff und Falten an den Ecken angebracht werden.

Andere bauen einen speziellen Rahmen mit vorgefertigten Eckverbindungen, um die Lichter in einem idealen Winkel zu halten.

Wie auch immer man vorgeht, man muss die LED-Leuchten auf jeder Seite zählen. Man benötigt diese Zahlen für die HyperHDR-Konfiguration.

Den Beginn der LED-Kette (die Stelle, an der das Signalkabel an den Raspberry Pi angeschlossen ist), muss man sich notieren. Der Anfang wird später per Software konfiguriert. Man kann in der Software auch einen Abstand eingeben, gemessen in LED-Pixeln, wenn z. B. für den Standfuß des Fernsehers eine Lücke in der Lichtleiste entsteht.

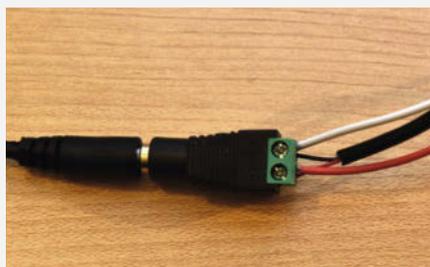
Software herunterladen und installieren

Für dieses Projekt wird HyperHDR verwendet, eine Variante der Open-Source-Software Hyperion. Damit kann man die Helligkeit der LED-Leuchten an die Darstellung des heimischen HDR-Fernsehers anpassen, ohne dass man einen teuren HDMI-Splitter benötigt, der das HDR-Signal auf beiden Kanälen überträgt. Die meisten HDMI-Splitter und Capture-Karten stufen das gesplittete Signal auf SDR herunter, was zu einer weniger farbenfrohen Übertragung der umgebenden Lichter führt. HyperHDR behebt dieses Problem.

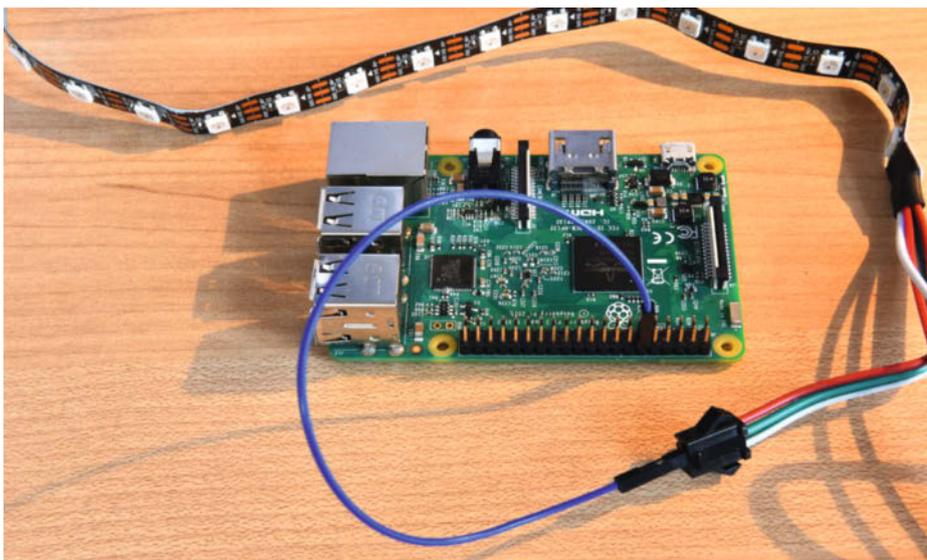
Die neueste Version der Software gibt es über den Link in der Kurzinfor. Ich verwende die 64-Bit-Beta-Version von 19.0.0.0 auf einem Raspberry Pi 3. Die Versionen für einen Raspberry Pi

Hinweis zur Raspberry Pi Stromversorgung

Wenn man 5-V-Lampen verwendet und den Raspberry Pi mit derselben Stromversorgung nutzen möchte, geht das natürlich auch. Man schneidet das große Ende (USB-A) eines Micro-USB- oder USB-C-Kabels ab (je nachdem, welches Pi-Modell man verwendet). Suchen sie die roten und schwarzen Drähte. Man entfernt einen Teil der Isolierung und steckt diese zusammen mit den Drähten, die den LED-Streifen mit Strom versorgen, in die entsprechenden Plus- und Minus-Steckplätze.



Eine Stromversorgung für LEDs und Raspberry Pi.



Das grüne Kabel des LED-Streifens ist für die PWM-Steuerung der LEDs verantwortlich und wird mit einem Verbindungskabel (im Bild blau) an den Raspberry Pi angeschlossen.

folgen einem Namensschema: SD-card-image-(Versionsnummer)-aarch64.img.xz.

Auf der Release-Seite befinden sich auch Versionen, die auf dem Pi 4, Pi Zero 2 W und der Revision 1.2 des Pi 2 funktionieren.

Sobald die Datei auf dem Computer gespeichert ist, verwendet man ein Imaging-Programm, um eine bootfähige SD-Karte zu erstellen.

Ich mag Etcher, aber auch andere Optionen wie Rufus funktionieren hervorragend. Sie können auch das Tool Raspberry Pi Imager verwenden, mit dem man viele verschiedene Arten von Images mit den gewünschten Einstellungen erstellen kann. Die verwendete SD-Karte muss mindestens eine Kapazität von 8 GB haben.

Sobald das Image erstellt ist, kann man der Karte eine Datei hinzufügen, um dem Raspberry Pi direkt Zugang zum WLAN zu geben. Der Raspberry Pi benötigt Zugang zum Netzwerk, weil die Einstellungen an der LED-Software über eine Weboberfläche vorgenommen werden.

Dazu schreibt man in einem Texteditor den Text aus dem Kasten WLAN-Konfiguration und passt ihn an die eigene WLAN-Umgebung an.

Diese Datei speichert man unter dem Namen wpa_supplicant.conf in dem Ordner boot auf der SD-Karte, während sie sich noch im Computer befindet.

Die SD-Karte wird dann in den Raspberry Pi eingelegt und der Computer mit einem HDMI-Kabel an den Fernseher angeschlossen. Sobald der Raspberry Pi mit Strom versorgt wird, fährt er automatisch hoch. Nach ein paar Minuten kommt die Aufforderung, sich anzumelden (wenn während des Flash-Vorgangs nichts anderes eingestellt wurde: Benutzer: Pi, Passwort: Raspberry).

Ich verwende eine Tastatur und Maus, um diese Daten direkt auf dem Bildschirm des Pi einzugeben. Man kann sich auch per SSH von einem separaten Computer aus in den Pi einloggen, aber da dieses Projekt bereits einen Fernsehbildschirm beinhaltet, ist es genauso einfach, die Daten direkt einzugeben. In Make 5/23 wird im Artikel „Verbindungen zum Raspberry Pi Server“ dieses Vorgehen detaillierter beschrieben.

Als Nächstes braucht man die IP-Adresse des Raspberry Pi. Ermittelt wird diese mit dem Befehl `ifconfig` im Terminal des Raspberry Pi.

Die IP-Adresse sieht in etwa so aus: 192.168.xx.xx. Jetzt kann man die USB-Peripheriegeräte entfernen und den Pi von der HDMI-Verbindung abtrennen. Von hier aus sollte alles andere über SSH und die HyperHDR-Benutzeroberfläche erledigt werden.

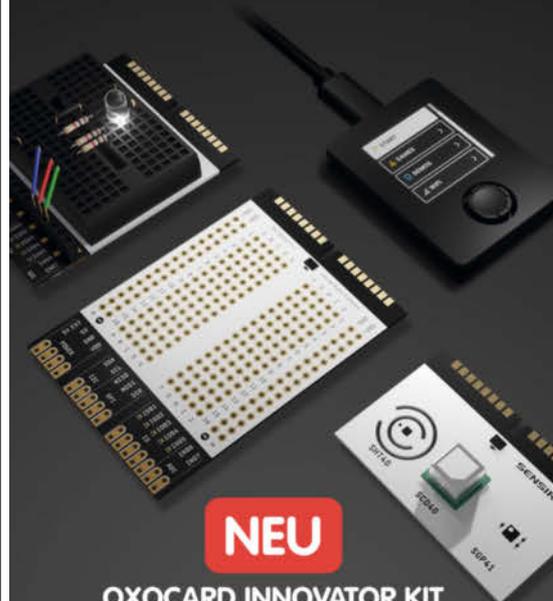


Die LED-Leisten müssen so weit außen wie möglich angebracht werden. Dabei ist eine Neigung hilfreich.

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



BESCHENK DICH SELBST

Kostenloser Elektronik- und Programmierkurs mit 15 Experimenten (14-99J)



Jetzt im heise Shop bestellen

In der Schweiz bei Galaxus oder Brack

www.oxocard.ch

WLAN-Konfiguration

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
land=DE
network={
  ssid="Name Ihres WLAN"
  psk="Passwort für Ihr WLAN"
}
```

```
[pi@hyperhdr:~] $ sudo systemctl status hyperhdr@root.service
• hyperhdr@root.service - HyperHdr ambient light systemd service for user root
   Loaded: loaded (/etc/systemd/system/hyperhdr@root.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2022-12-19 01:50:10 GMT; 27s ago
     Main PID: 2454 (hyperhdr)
        Tasks: 11 (limit: 779)
             CPU: 6.931s
   CGroup: /system.slice/system-hyperhdr.slice/hyperhdr@root.service
           └─2454 /usr/bin/hyperhdr
```

active (running) zeigt an, dass die Software korrekt läuft.

Wechsel zum ROOT-Benutzer

Wenn man WS2812B-LEDs verwendet, werden diese wahrscheinlich mit PWM über GPIO-Pin 18 gesteuert (der Standard-PWM-Pin für HyperHDR und viele andere Raspberry-Pi-LED-Projekte).

Es gibt eine Sache, die in den Beschreibungen von Hintergrundbeleuchtungsprojekten nicht immer erwähnt wird: Die PWM-Funktion des Pi ist nur dann zugänglich und kann genutzt werden, wenn sich der Raspberry Pi im Root-Modus befindet.

Im Allgemeinen wird im Normalbetrieb davon abgeraten, da dies einen direkten Zugriff auf kritische Dateien ermöglicht. Allerdings kann bei diesem Projekt der Raspberry

Pi nach der Konfiguration vom Netzwerk getrennt werden.

Der Root-Zugriff wird folgendermaßen aktiviert:

1. Verbinden per SSH mit dem Pi unter Verwendung der IP-Adresse
2. Passwort eingeben.
3. Deaktivieren des Benutzer pi:
sudo systemctl disable --now hyperhdr@ pi.service
4. Aktivieren des Benutzer root:
sudo systemctl enable --now hyperhdr@root.service
5. Überprüfen, ob root aktiv ist, indem man den Statusbefehl verwendet:
sudo systemctl status hyperhdr@root.service

Konfigurieren der Benutzeroberfläche

In einem Browser kommt man über die URL hyperhdr:8090 oder <Raspberry-Pi-IP>:8090 auf die HyperHDR-Benutzeroberfläche.

Es gibt eine ganze Reihe von Einstellungsmöglichkeiten in der Software, folgend die Minimaleinstellungen für dieses Projekt:

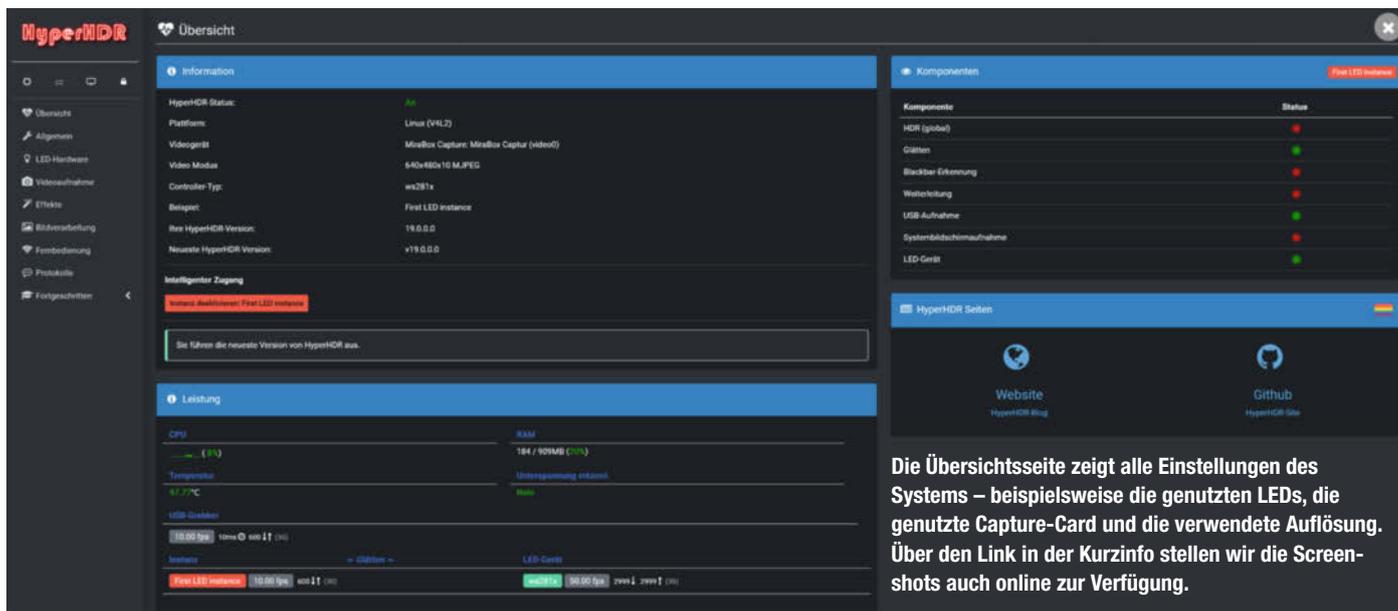
Videoaufnahme: Der HDMI-Grabber sollte hier automatisch unter den Geräteeinstellungen auftauchen. Man kann die Auflösung und die Bildrate auf „Automatisch“ belassen, aber um die Dinge flott zu halten, sollte man die Einstellungen auf 640x480 und 20 FPS reduzieren – wir haben es hier nicht mit feinen Pixeln zu tun. Die Einstellung „Tone-Mapping HDR zu SDR“ kann man auch aktivieren. Ganz wichtig: Das Klicken auf „Einstellungen speichern“ unten auf der Seite nicht vergessen.

Auf demselben Bildschirm im Abschnitt „Instanz USB Capture“ das Kontrollkästchen „USB-Aufnahme“ aktivieren und dann wieder auf „Einstellungen speichern“.

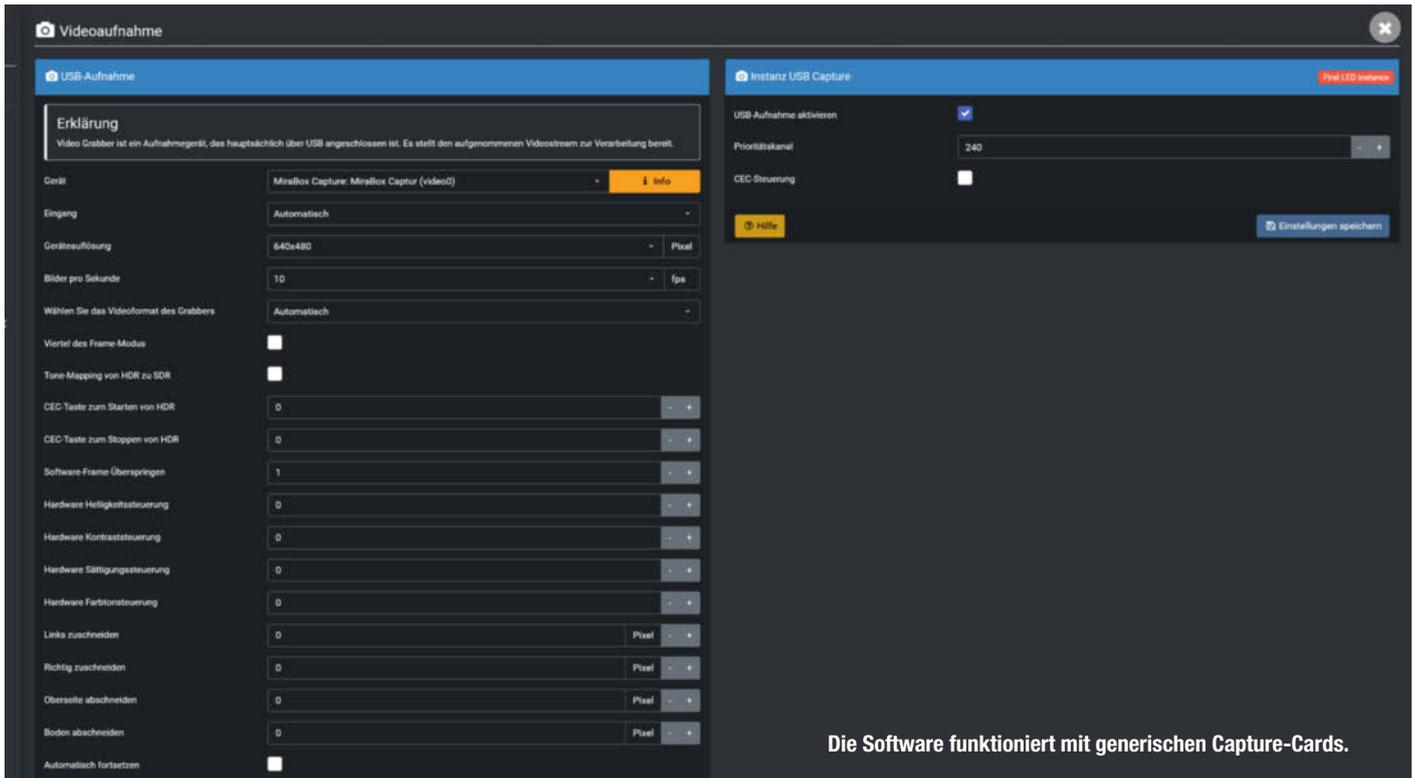
LED-Hardware: Auf der Registerkarte „LED-Controller“ aus dem Dropdown-Menü „Controllertyp“ die Option ws281x auswählen. Möglicherweise muss man die RGB-Reihenfolge ändern (dafür gibt es einen Assistenten im Menü „Fortgeschritten“).

Unter „Maximale LED-Anzahl“ muss mindestens die Anzahl der verwendeten LEDs eingetragen werden. Nach einem Klicken auf „Einstellungen speichern“ beginnen jetzt möglicherweise einige der LEDs zu leuchten.

Oben auf diesem Bildschirm gibt es den Reiter „LED-Layout“. Hier werden nochmal die Anzahl der LEDs, die Länge der Lücke und die Eingangsposition eingetragen. Sobald auf „Einstellungen speichern“ geklickt wird,



Die Übersichtsseite zeigt alle Einstellungen des Systems – beispielsweise die genutzten LEDs, die genutzte Capture-Card und die verwendete Auflösung. Über den Link in der Kurzinfo stellen wir die Screenshots auch online zur Verfügung.



sollte sich das komplette LED-Array mit Licht füllen.

Völlig eingetaucht

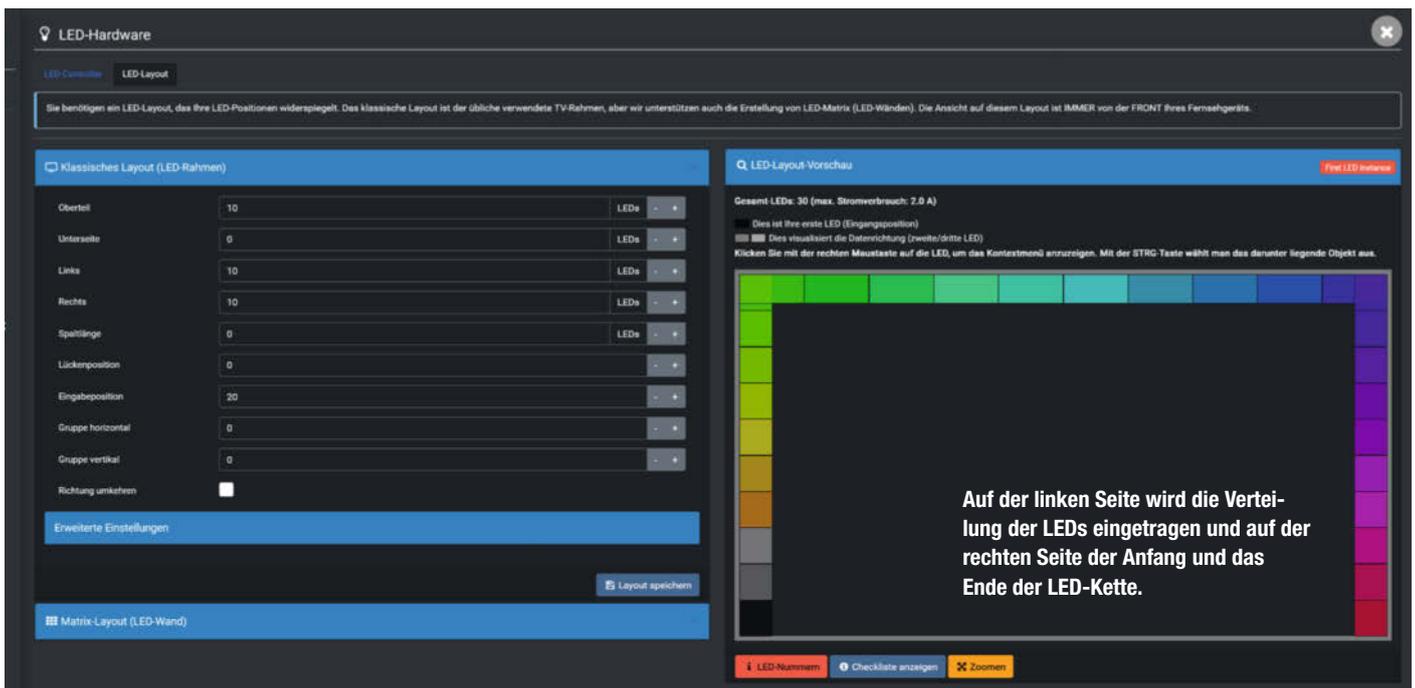
Wenn man sich nun Inhalte mit dem HDMI-Streaming-Gerät anschaut, sollte man die entsprechenden Umgebungslichter sehen können. Cool, nicht wahr? Möglicherweise muss

man jetzt noch etwas in der Software feinjustieren. Beispielsweise mit LED-Platzierung, der Farbanpassung oder der Erkennung von schwarzen Balken.

Natürlich ist der Effekt in einem abgedunkelten Raum oder bei Nacht am deutlichsten. Möglicherweise muss man den Fernseher neu positionieren, um eine bessere Lichtstreuung zu erreichen oder um zu verhindern, dass die

hellen LEDs von Personen, die neben dem Fernseher stehen, zu sehr wahrgenommen werden.

Im Hyperion-Forum findet man Hinweise zur Einrichtung und zu Problemen. Nutzer mit A/V-Receiver, Surround-Sound und CEC-verbundenen Geräten müssen hochpreisige HDMI-Hardware verwenden, um diese Funktionen zu erhalten. —das





Head-Up-Display für Skifahrer

Ein Cockpit für die Skibrille, das Geschwindigkeit, Höhe und Kurs in Echtzeit anzeigt und deine Sicht nicht stört – so etwas kann man ab rund 500 Euro kaufen, oder aber selber bauen!

von Georg Jacobs

Tolle Apps, die uns nach der Fahrt den Skitag bis ins Kleinste analysieren und Daten wie Top-Geschwindigkeit und Strecke anzeigen, gibt es zuhauf. Aber was kann man nutzen, während man auf der Piste ist?

Schon vor Jahren wollte ich eine Oakley-Skibrille mit Minidisplay haben, das mir Ort und Geschwindigkeit anzeigt. So etwas war aber viel zu teuer. Dann erhielt ich die DIY-Idee für ein Head-Up-Display aus einem YouTube-Video (Link in der Kurzinfor) und mir war klar, dass so ein Projekt machbar ist.

Das Design aus dem Video habe ich angepasst, sodass aus Sicherheitsgründen keine Fremdkörper im Brilleninnenraum platziert werden müssen und auch am Helm sollte keine Veränderung die Stabilität gefährden. Das fertige Gerät besteht aus zwei Teilen: einem Visier für OLED-Minibildschirm, Blendschutz und Optik sowie einem Steuerkästchen mit der Steuerelektronik und der Batterie.

Grundkonzept

Ein Head-Up-Display hat inzwischen fast jedes moderne Auto. Die Windschutzscheibe dient als halbtransparenter Spiegel und als Bildschirm, der Informationen mitten ins Sichtfeld einspielt. Das Ganze wird mit einer Sammellinsenoptik auf einen bequemen Sehabstand des Bildes (> 2 m) gebracht.

Doch die immense Lichtstärke auf der Skipiste – 100.000 Lux an einem sonnigen Tag – macht dort sehr leuchtstarke Displays nötig. Also habe ich die Anzeige in einen umschließenden schwarzen Sichtkanal (den Tubus) eingepasst und damit die Helligkeitsunterschiede kompensiert. Das Ganze ist am unteren und äußeren Rand des Sichtfeldes platziert, wo es die Übersicht nicht stört.

Zweck und Ergonomie

Was aber soll das Teil denn anzeigen? In der Konzeptionsphase wollte ich noch Temperatur, Feuchte und Druck mit aufnehmen und die Anzeige jeweils automatisch oder tastengesteuert wechseln. Doch die eingespielte Information muss intuitiv erfasst werden, darf keine ablenkenden Details enthalten und ruhig soll das Bild auch sein.

Nach ein paar Spielereien habe ich mich daher für eine Anzeige von folgenden Daten entschieden: gerundete Höhenmeter, Geschwindigkeit in m/s, Zahl der vom NEO-6M GPS-Modul gerade empfangenen Satelliten und Kurs in Grad, unterstützt durch einen Ausschnitt einer stilisierten Kompassrose als Laufband. Die Legende für die Daten ist ein Buchstabe, dessen Schriftgröße kleiner als die angezeigten Zahlen ist.

Wer mag, baut noch einen DHT22-Sensor (für Temperatur und Luftfeuchtigkeit) und einen Barosensor ein, der Arduino Nano packt die Daten locker.

Kurzinfor

- » Zeigt auf der Piste Geschwindigkeit und Höhe an
- » Kostengünstige Alternative zu kommerziellen Produkten
- » Umsetzung mit NEO-6M GPS-Modul

Checkliste



Zeitaufwand:
ein Wochenende



Kosten:
40 Euro

Werkzeug

- » 3D-Drucker
- » Lötkolben

Mehr zum Thema

- » Benno Lottenbach, Ping-Pong-Trainingsroboter selber bauen, Make 6/22, S. 36
- » Fabian Bräunlein, ESP32-Projekt: Tracker für Apples Find-My-Netzwerk bauen, Make 1/23, S. 48
- » 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/xmue

Material

- » Arduino Nano
- » Kippschalter 2 Pole
- » Klinkestecker 4 Pole
- » Klinkebuchse 4 Pole
- » NEO-6M GPS-Modul
- » OLED-Monochrom-Display 0,91 Zoll
- » Sammellinse f ~ 50 mm
- » PLA

Der zweite Schritt in Sachen Ergonomie war die Schriftgröße. Die 128 x 32 Pixel lassen wenig Platz. Zwei bibliotheksseitig eingebaute einfache Systemschriften helfen weiter. Ein größeres Display hätte 128 x 64 Pixel, doch wird damit der Tubus schon wieder so hoch, dass es zu einer deutlichen Sichtfeldeinschränkung kommen kann.

Auch bezüglich der Anzeigefarbe sollte so wenig Aufmerksamkeit wie möglich auf das Display gezogen werden. Für mich und die meisten anderen ist Skifahren ja ein Augensport. Zudem verfälschen die getönten Brillenscheiben das Ergebnis einer farbigen Anzeige. Deshalb werden die Display-Inhalte in weißer Schrift auf schwarzem Hintergrund angezeigt. Das sorgt für den höchsten Kontrast und minimale Störung.

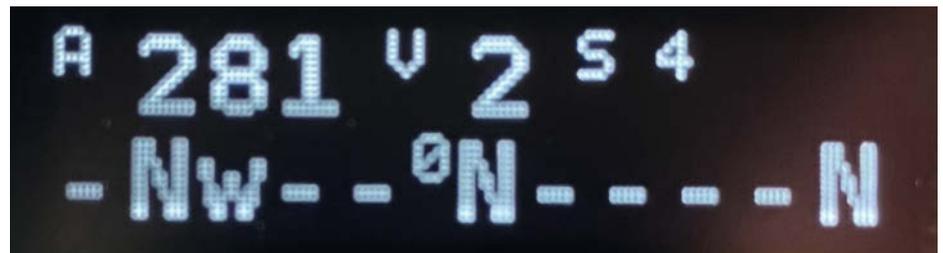
Wer sich mit dem Thema tiefergehend beschäftigen mag, der lese die Dissertation von Nataša Miličić, die alle relevanten Themen von

Optik über Ergonomie bis hin zum Thema Ablenkung aus der Sicht eines Kfz-Ingenieurs ausführlich beschreibt. Den Link dazu gibt es in der Kurzinfor.

Anpassung und Design

Die optische Anpassung erledigt eine Sammellinse, das Feintuning zur scharfen und leicht lesbaren Darstellung dazu funktioniert über die Tubuslänge. Da habe ich ein paar 3D-Drucke mehr gebraucht. Dieses Element habe ich im FreeCAD aus zwei Keilen modelliert, deren eine Koordinate die Tubuslänge definiert. So kann die vorhandene Linse und der ausgleichende Sehfehler angepasst werden – wie bei einem Monokular – ohne dass die anderen Passungen beeinträchtigt werden. Mehr Details zum Anpassungsprozess folgen später.

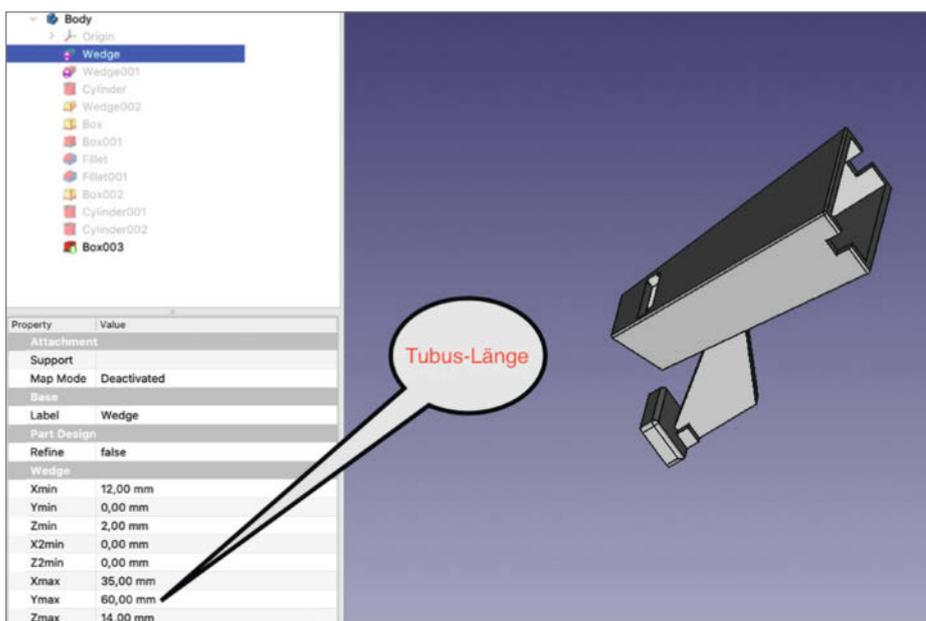
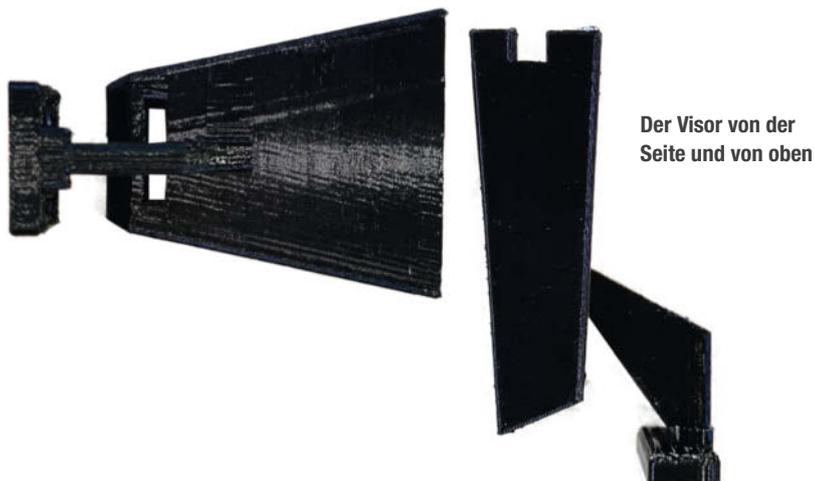
Am Ende, das mit der Brille verbunden wird, ist der Tubus gerundet (mittels abzuziehenden



Die Anzeige: Oben sieht man Höhenmeter, Geschwindigkeit und empfangene Satelliten, unten einen Kompass.



Display ohne den Tubus



In FreeCAD bestimmt die Variable Ymax die Tubuslänge.

dem Zylinder in FreeCAD) und passt sich dadurch der äußeren Form der Brille an. Fixiert wird das Ganze mit feinem Draht oder einem Faden an den schaumstoffverkleideten Lüftungsgittern unter der Brille. Ein Gummi spannt einen Nylonfaden. Dieser zieht den Aufbau in die Höhe und wird an der Oberseite befestigt.

Damit sind keine Bohrungen oder ähnliches an der Brille nötig und der Aufbau kann sich beim Sturz ohne Kraft lösen. Materialverlust ist mir im Zweifelsfall weniger wichtig als mein Augenlicht.

Die Steuereinheit kommt in ein 75×75 mm großes Kästchen, das Batterien, Arduino, Schalter, Buchse und GPS beherbergt. Auch hier ist eine Seite an die Helmrandung angepasst. Ich habe erst mit einer Formlehre gearbeitet, aber am Ende war es genauer, einfach einen Kreisbogen zu nehmen, womit ich in FreeCAD abzuziehende Zylinder definieren konnte.

An dünnen Metallstiften (ich habe Deko-Stecknadeln genommen), die ich durch die passend liegenden Lüftungsschlitze zwischen Helm und Schaumstoff eingeschoben habe, konnte ich mit Gummizügen die Box ohne Struktureingriff am schützenden Helm anbringen. Auch hier gilt, beim Unfall lieber Materialverlust als Verletzungen in Kauf zu nehmen.

Entwicklungs- und Bauzeit

Beginnen hatte ich mit dem Projekt Ende November und der Skiurlaub war in der ersten Märzwoche geplant. Während des Urlaubs habe ich nur noch am Code gebastelt. Da das Projekt aber mein erster komplexerer 3D-Druck war, musste ich viel Lernzeit für FreeCAD aufbringen. Dafür ist der Code sehr simpel.

Der Code

Das Programm ist extrem simpel aufgebaut. Eine Endlosschleife liest die Daten aus dem GPS-Controller über `SoftwareSerial` und gibt die Information mit einer im Programmtext vor dem Kompilieren fest einzugebenden Refresh-Zeit aus. Wer also noch weitere Features reinpacken will, hat selbst im Nano genug Platz. Der Controller ist ohnehin die meiste Zeit mit „Daumendrehen“ beschäftigt. Konkret verwendet der Sketch 16348 von 30720 Bytes (53%) des Programmspeicherplatzes und die globalen Variablen verwenden 723 von 2048 Bytes (35%) des dynamischen Speichers. Laut Compiler verbleiben 1325 Bytes lokaler Variablenspeicherplatz.

Von Vorteil ist die recht kleine Bibliothek `TinyGPSPlus`, die sehr auf Speichersparsamkeit hin geschrieben wurde.

Verwendet wurden insgesamt die Bibliotheken:

- `TinyGPSPlus.h`
- `SoftwareSerial.h`



Die Befestigung am Helm mithilfe von Gummis und Stecknadeln sorgt im Falle eines Sturzes dafür, dass die Technik vom Helm abfällt, ohne den Nutzer zu verletzen.

- Wire.h
- SSD1306Ascii.h
- SSD1306AsciiWire.h

Wie diese Bibliotheken eingebunden werden, steht in einem verlinkten Online-Artikel in der Kurzinfo.

Optik

Das System entspricht einer Nahlupe. Das Auge liegt innerhalb des f-Abstands (also innerhalb der Brennweite; hier ~52 mm) der Linsenebene und das Display ist knapp unter einer Brennweite weit weg. Damit ergibt sich ein fast im Unendlichen gesehenes, virtuelles, aufrechtes und seitenrichtiges Bild. Der Bildausschnitt ist ca. 10° hoch und 30° breit. Im Sichtbereich liegt das Display am Unterrand des erweiterten Sichtfeldes seitlich, also deutlich außerhalb der Region, die für ein rasches Erkennen von Hindernissen auf der Piste oder seitlich herannahenden anderen Skifahrern ungestört bleiben muss.

Sollte der Betrachter eine Korrektur für das Sehen in der Ferne benötigen (Brille, Kontaktlinsen) und diese unter der Skibrille nicht tragen, so kann die Stärke der Korrektur mit der Linse verrechnet werden. Als grobe Berechnung nimmt man die Formel mit f in mm und Brillenstärke in Dioptrien. Wer unter der Skibrille eine Brille trägt, braucht die Korrektur nicht zu beachten.

Mit diesem Maß kann der Tubus gedruckt werden. Die nötigen Feineinstellungen habe ich mit einem Millimeter-Maßband bewerkstelligt, das ich in den Tubus eingelegt habe. Die genau scharf abgebildete Millimeter-Marke zeigt mir dann die Displayebene an. Das Display ragt ca. 1 mm in den Tubus hinein.

$$f_{\text{neu}} \approx \frac{1000}{\frac{1000}{f_{\text{Linse}}} - \text{Brillenstärke}}$$

Berechnung der Korrektur der Linse

Anpassung der Konstruktionsdateien

In der Datei zum Visierträger ist das Maß für die Länge des Tubus im allerersten Bauteil „Wedge (Keil)“ unter YMax abgelegt (hier: 60 mm). Den Innenraum definiert ein zweiter Keil, der Überlänge hat – was nicht stört, da keine Kollision mit anderen Bauteilen droht. Wird die Tubuslänge in dieser Variable angepasst, ändert sich das Außenmaß für den Deckel nicht, andere Anpassungen entfallen also. Ggf. ist dann nur der Ausschnitt für die Kabelführung des Deckels zu verlegen (Box003; Z-Direction; hier -60 mm).

Die Batterien sind ohne Spiel eingeklemmt. Wer etwas mehr Platz möchte, ändert in der Datei namens Helmaufsatz das Außenmaß im Abschnitt Pad/Sketch und dann in Pocket/Sketch001 die Länge des Ausschnittes.

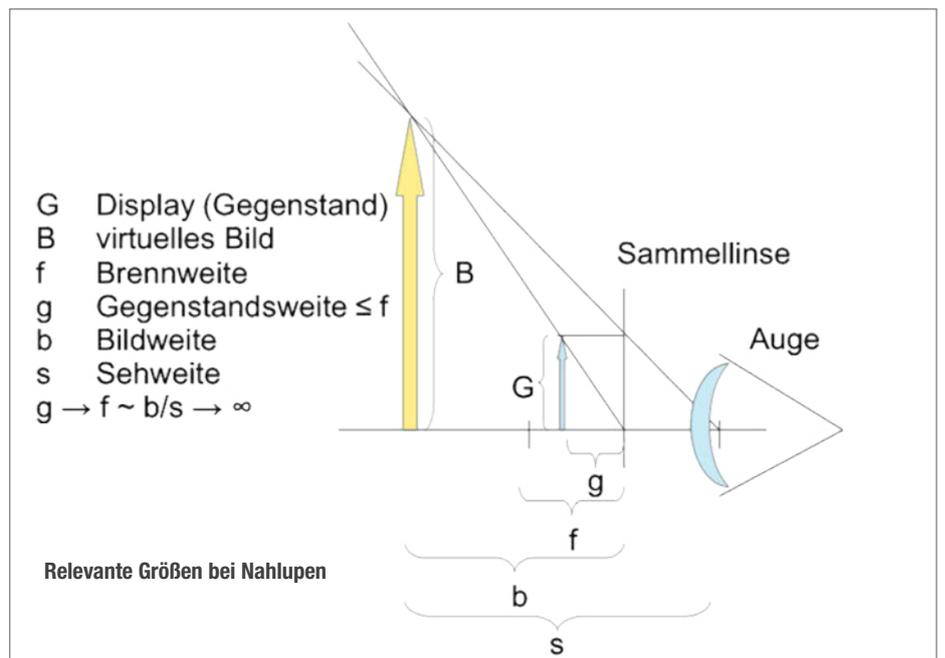
Um Material in der Testphase zu sparen, kann bei der Anpassung der Kreisausschnitte

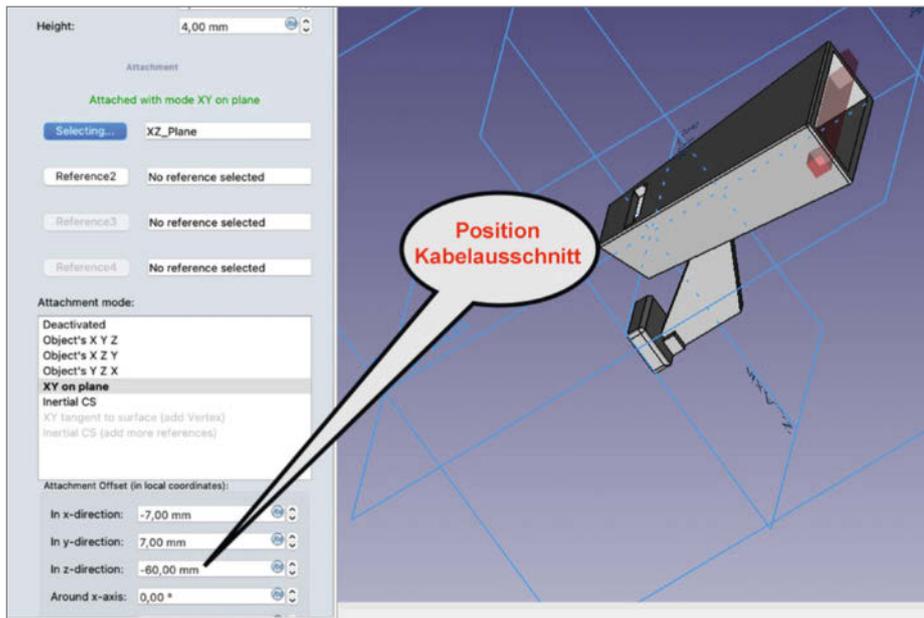
(Cylinder.003) auf die Datei namens *_Helm-anpassung zurückgegriffen werden, in der der überwiegende Teil der Box abgeschnitten ist. Nach endgültiger Maßfindung kann die letzte abzuziehende Box gelöscht werden und das Kistchen ist vollständig.

Die Bohrlöcher für Klinkenstecker, Schalter und die Befestigung des Tubus am Brillenunterrand habe ich nicht konstruiert. Die Festigkeit des Materials gibt ein simples Bohren der Aussparungen her. Der Helmaufsatz aus PLA mit 15 Prozent Füllung ist so fest, dass ich mich draufstellen kann.

Verkabelung

Der doppelpolige Kippschalter trennt die Batterien vom Arduino Nano (Eingang VCC und GND). Vom Arduino aus werden GND und 5V an den GPS-Controller und das Display geleitet. A3/4 bedienen SoftwareSerial, D4/5 den I²C-Bus. In der Testphase habe ich





In der FreeCAD-Datei lässt sich der Kabelführungsausschnitt beliebig variieren.



Mit dem HUD hat man auf der Piste alles im Blick.

den Arduino nicht fest verbaut, damit der USB-Port zugänglich ist. Im Testbetrieb läuft die Stromversorgung dann entsprechend über die USB-Speisung und die Batterien sind abgetrennt.

Die Software wird über USB von einem anderen Computer aufgespielt. Nachdem man den Arduino verbunden hat, öffnet man die Datei `GPS_OLED.ino` mit der Arduino IDE. In dem Programm unter „Board/Arduino AVR Boards“ Arduino Nano auswählen und unter Ports den verbundenen Arduino Nano einstellen. Dann auf Upload klicken. Eine genau-

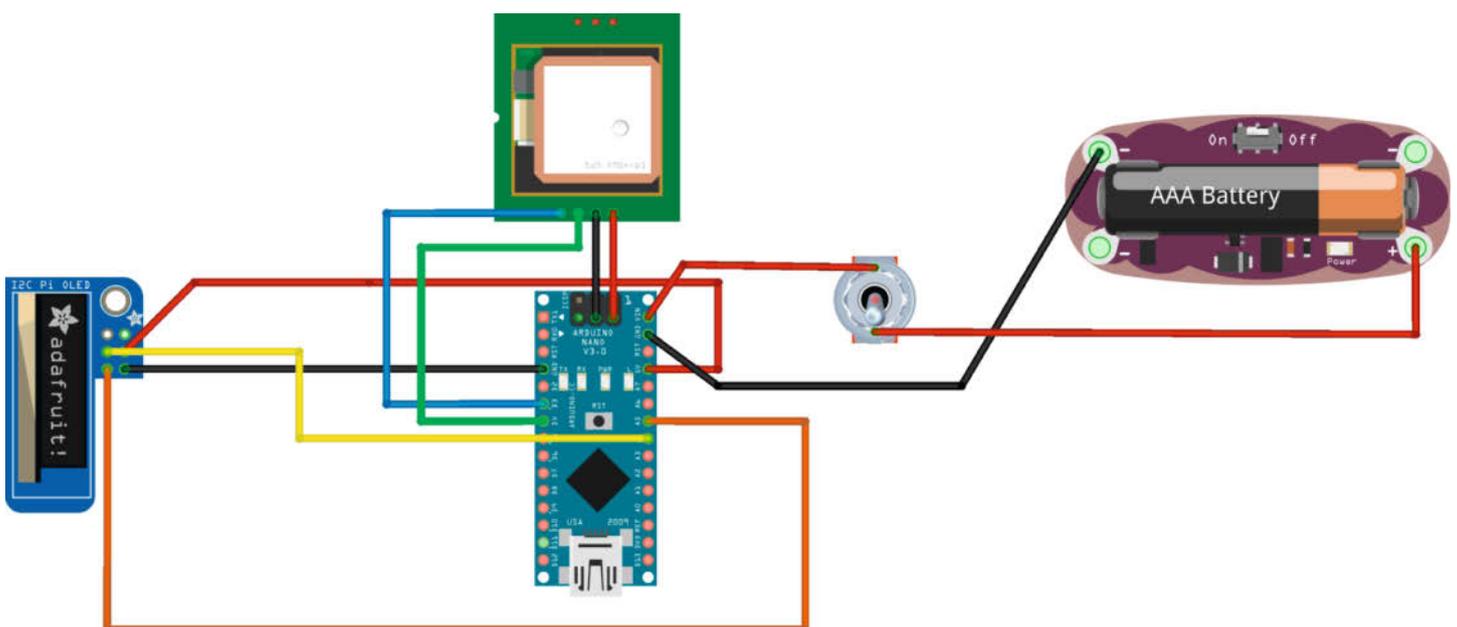
ere Erklärung für diesen Vorgang ist in der Kurzinfo verlinkt. Dort befindet sich auch der Link zum Code.

Wo gab es Probleme?

Obwohl die TinyGPSPlus eine gute Bibliothek ist, musste ich etwas tüfteln, damit das Auslesen des NEO-6M über `SoftwareSerial` klappte (ich habe ewig gebraucht, bis auch die Höhe gelesen wurde). Es war ein Timing-Problem. Daher habe ich die standardmäßig eingebaute Funktion `delay()` durch eine ei-

gens definierte Funktion ersetzt, die während der Pausen zwischen den Display-Aktualisierungen auch die serielle Verbindung abfragt. Demgegenüber war das OLED-Display unkompliziert.

Die Befestigung an der Brille ist ein Kompromiss, denn zu viel Zug am Gummi drückt den Brillenkunststoff ein. Bei zu wenig Zug fällt das Gerät auf der Piste ab. Ich habe mit Haushaltsgummis als Federelement gespielt und die Befestigung zur Brillenoberseite mit einem dünnen Nylonfaden geführt; der stört den Blick überhaupt nicht.



Verbindung von Arduino, Display und GPS-Modul. Die Stromversorgung kann auch über USB erfolgen.

fritzing

Tipps

Ich habe auf einen externen Spannungswandler verzichtet und mich auf den des Arduino Nano verlassen und das klappte.

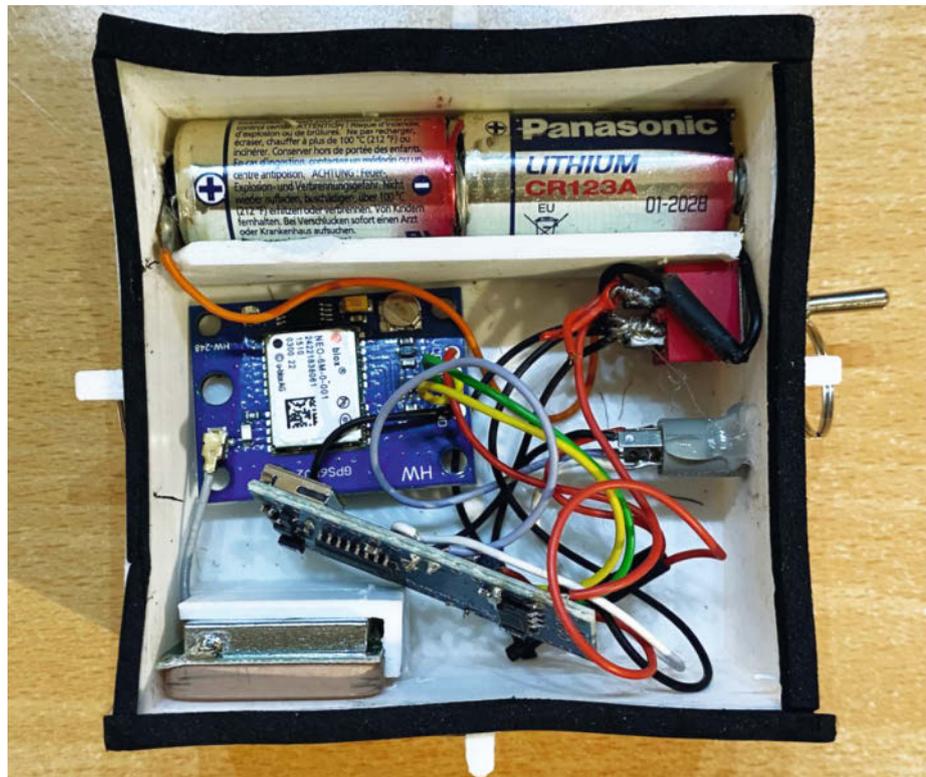
Die mechanische Anpassung der Rundungen des Helms ist schwierig zu konstruieren, sodass ich einen kleinen Rahmen gedruckt habe, an dem ich die Passung zum Helm testen konnte, ohne viel Zeit oder PLA zu investieren. Diese, sowie alle weiteren 3D-Dateien, sind in der Onlineinfo verlinkt.

Bei Modifikationen am Tubusaufbau muss darauf geachtet werden, dass die Linse und das Display in einer Ebene liegen, sonst leidet die Ablesbarkeit sehr. Damit war auch meine

Ursprungsidee, eine Linse zwischen den zwei Brillenkunststoffen platzsparend einzuschleiben oder eine Fresnellinse zu verwenden, schnell als optisch unzureichend ausgeschieden.

Verbesserungsideen

Wer gut konstruieren kann, findet sicher einen Weg, einen Rahmen an die Brille anzupassen und so zu drucken, dass der Tubus schöner angeflanscht werden kann. Eine Erweiterung des Codes auf ein Halten der Geschwindigkeit mit Anzeige des lokalen Maximums für 2 bis 5 Sekunden – wie bei einer Aussteuerungsanzeige im A/V-Bereich – wäre auch eine gute Idee! —das



Stromversorgung, Mikrocontroller und GPS-Antenne

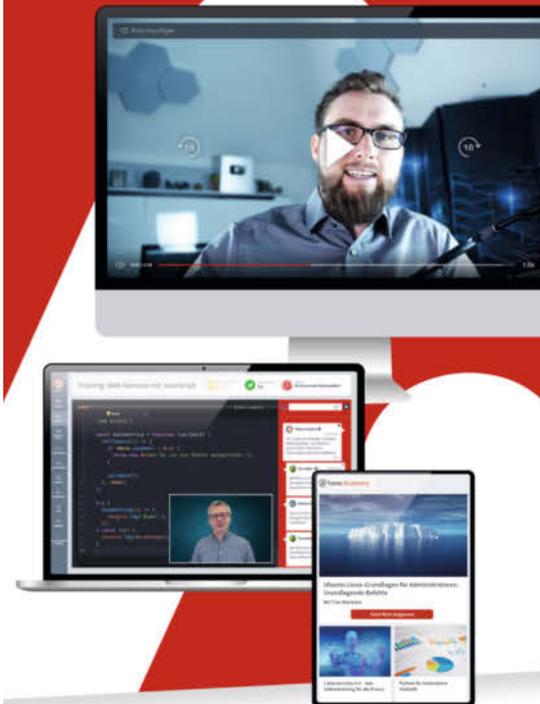
Hinweis zum GPS-Modul

Das GPS-Modul benötigt beim ersten Einschalten – insbesondere nach einem Ortswechsel – bis zu zehn Minuten, um sich geografisch zu finden. Geduld, es ist nicht kaputt; später ist es in der Regel in deutlich unter einer Minute parat.

Je nach Variante hat die Controllerplatine eine LED, die die sichere Verfügbarkeit der Positionsdaten anzeigt. Damit kann im Testmodus vor dem Zusammenbau die sichere Funktion geprüft werden. Diese LED zeigt die Funktionsbereitschaft an,

auch wenn keine Datenverbindung aufgebaut ist. Je nach Bezugsquelle kommt keine Stützbatterie mit. Aber auch falls sie mitgeliefert wird, ist deren Erneuerung gut investiertes Kleingeld. Die beiliegende Batterie ist oft schnell leer.

Die Antenne wird mit einem Koaxial-Clipverbinder auf den Controller montiert, das ist sehr empfindliches dünnes Material. Im gedruckten Gehäuse ist die Antenne dann mit Blick zum Himmel eingebaut.



Qualifizieren Sie Ihre Fachkräfte für die Zukunft der IT

Mit Ihrem Partner für digitale IT-Weiterbildung

- 80 relevante IT-Themen von über 100 renommierten IT-Experten
- Jeweils über 100 Webinare und digitale Kurse
- Interaktives Lernen durch Features wie Übungsaufgaben und Wissenstests
- Individuelle Lernumgebung für jeden Mitarbeiter
- Uneingeschränkter Zugriff und volle Kostenkontrolle

**JETZT
KOSTENLOS
TESTEN**



**Mehr Infos unter
heise-academy.de**

IKEA-Uhr LEDifiziert

Als hätte IKEA seine PUGG-Wanduhr dafür konstruiert, lässt sie sich ideal mit einem LED-Streifen nachrüsten. Von einem PICAXE-Mikrocontroller gesteuert, kann man mit den 60 LEDs tolle Lichtspiele für die Uhr programmieren und diese sogar fernsteuern.

von Michael Gaus und Miguel Köhnlein



Mit LED-Streifen lässt sich so manches Wohn-Accessoire aufmotzen. So auch IKEAs Edelstahl-Wanduhr PUGG. Sie verfügt über einen Gehäusedurchmesser von etwa 32 cm, woraus sich ein Umfang von ungefähr einem Meter ergibt. Perfekt also, um die Uhr ringsherum mit einem ebenso langen LED-Streifen mit 60 LEDs zu versehen. So kann man zum Beispiel eine leuchtende Sekundenanzeige nachrüsten oder die RGB-LEDs programmieren, um zahlreiche Lichteffekte darzustellen und sie per Fernbedienung umzuschalten. Wie das geht, zeigen wir euch in diesem Artikel. Die benötigte Firmware findet ihr im GitHub-Repository des Projekts (siehe Link in Kurzinfo).

Wahl der Hardware

Als Leuchtmittel verwenden wir einen LED-Streifen mit APA102-LEDs und steuern sie mit dem Mikrocontroller PICAXE 14M2 an, der in einem Nano-Axe-Board steckt. Das Board ist einzeln und in Kombination mit dem Make PICAXE Special 2020 im heise Shop erhältlich. Da die Daten für die LEDs über die Signale Data und Clock eingespeist werden, reichen Rechenleistung (bis 32 MHz), Speicher (2 kB) und RAM (512 Bytes) des PICAXE 14M2 völlig aus. Für eine RGB-LED benötigen wir z.B. nur 3 Bytes Arbeitsspeicher, bei 60 LEDs also insgesamt 180 Bytes.

LED-Streifen montieren

Um einen LED-Streifen mit einem Meter Länge an der PUGG-Uhr anzubringen, benötigt man rein rechnerisch einen Durchmesser von 31,8 cm (100 cm/Pi). Da der wahre Durchmesser mit 31,6 cm knapp darunter liegt, befestigen wir zuerst einen ca. 1 mm dicken Kartonstreifen rings um den Rand, der zur Wand zeigt (Bild 1) und nutzen dafür doppelseitiges Klebeband. Das bringt den Durchmesser auf das passende Maß. Außerdem dient der Bastelkarton gleichzeitig als Isolierung zwischen dem leitenden Edelstahlgehäuse der Uhr und dem LED-Streifen. Die Klebeschicht allein reicht als Isolierung nämlich nicht aus, sodass es ohne den zusätzlichen Karton zu Kurzschlüssen kommen könnte.

Auf den umlaufenden Karton kleben wir dann den selbstklebenden LED-Streifen. Die erste LED, deren Pfeil in Richtung der restlichen LEDs zeigt, muss sich bei Sekunde 1 befinden, die restlichen LEDs bringt man mit den Pfeilen im Uhrzeigersinn an. Je nach Kartondicke kann der Streifen am Ende etwas zu lang sein, sodass man eventuell hinter der letzten LED vorsichtig ein paar Millimeter abschneiden muss. An die vier Anschlüsse GND, CI, DI und 5V der ersten LED löten wir Litzen an, die wir auf die Rückseite der Uhr führen und dort mit der Steuerplatine verbinden.

Kurzinfo

- » IKEA-Uhr um LED-Streifen und PICAXE ergänzen
- » Eigene Lichtmuster programmieren
- » LED-Streifen mit einer Infrarot-Fernbedienung steuern

Checkliste



Zeitaufwand:
2 bis 3 Stunden



Kosten:
ca. 90 Euro

Material

- » IKEA PUGG-Wanduhr
- » IKEA SMÅHAGEL 5V-Netzteil mit 2,4A
- » Nano-Axe-Board
- » PICAXE 14M2
- » APA102-LED-Streifen 5V, 1 Meter, 60 LED/m
- » Infrarot-Modul IR1261
- » IR-Fernbedienung mit NEC-Protokoll, siehe Kasten im Artikel
- » 100Ω-Widerstand
- » 4,7µF-Kondensator
- » Micro-USB-Kabel
- » Lochrasterplatten
- » Kabel, Lötzinn, doppelseitiges Klebeband, Pappkarton

Mehr zum Thema

- » Michael Gaus und Miguel Köhnlein, Für Halloween: Beleuchtetes Kürbis-Gesicht mit Nano-Axe-Board
- » Helga Hansen, PICAXE-Einstieg: Vorbereitung zum Programmieren
- » Carsten Wartmann, Disco is back – Lichtshow mit WLED, Make 2/23, S. 8
- » Michael Gaus und Miguel Köhnlein, Smart Home: IKEA-Pfefferkuchenhäuser mit NanoAxe-Board basteln
- » Video: LED-Mod in Aktion



Werkzeug

- » Lötkolben
- » Seitenschneider
- » Schere

Alles zum Artikel im Web unter make-magazin.de/xu8a

Steuerplatine

Die Steuerung haben wir auf einer kleinen Lochrasterplatine mit Löttraster aufgebaut (Bild 2). Mit einer Laubsäge kann man

ein passendes Loch aus dem oberen Teil der Steuerplatine sägen, sodass sich diese über den runden Kunststoffblock für die Uhraufhängung schieben lässt. Der untere Platinenrand liegt direkt am Kunststoffsteg an, der



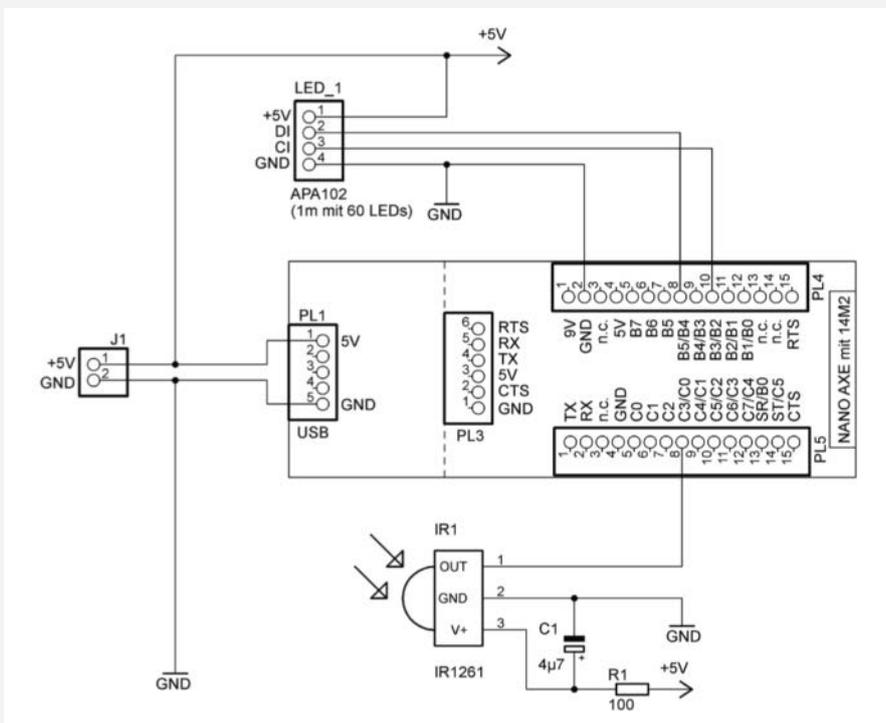
Bild 1: Der LED-Streifen sitzt auf einem isolierenden Bastelkarton.

Schaltplan

Die im Nano-Axe-Schaltplansymbol innenliegenden Pins kennzeichnen die Port-Nummern des hier verwendeten PICAXE 14M2. Dieser verfügt über 12 I/O-Pins B0 bis B5 sowie C0 bis C5. Die Bezeichnungen an den Kontakteleisten PL4 und PL5 stimmen mit dem Aufdruck auf der Platine überein und kennzeichnen die Pins eines PICAXE 20M2.

Die Stromversorgung kann über ein USB-Netzteil mit 5V Ausgangsspannung am Steckverbinder J1 erfolgen. Man sollte darauf achten, dass das Netzteil den benötigten Strom liefern kann. Eine einzelne weiße LED benötigt bei voller Helligkeit nämlich ca. 60 mA, wenn alle 3 RGB-Farbwerte im Programm auf 0xFF gesetzt sind – bei 60 LEDs, die gleichzeitig leuchten, ergibt das einen Strombedarf von 3,6 A. Lässt man diese aber z.B. mittels 0x22 weniger hell leuchten oder nur ein paar gleichzeitig auf voller Stärke, reicht auch ein Netzteil mit 2,4 A, wie in der Materialliste angegeben.

Das RC-Glied R1/C1 dient dazu, die Spannungsversorgung des IR-Empfängers IR1261 zu entkoppeln, sodass keine Stör-



impulse eingefangen werden. Man sollte das RC-Glied möglichst dicht am IR-Empfänger platzieren. Deshalb löten wir es

nicht auf die Steuerplatine, sondern auf eine separaten kleine Lochrasterplatine, die näher am IR-Empfänger montiert ist.

sich beim Uhrwerk befindet. Ein Kabelbinder rund um den Uhraufhängungs-Bolzen verhindert, dass die Platine herausrutschen kann (Bild 3).

Die Verbindungen zum LED-Streifen, dem 5-V-Netzteil und einer weiteren Platine haben wir mit drei Stiftleisten hergestellt. Auf der separaten Zusatzplatine befindet sich

das RC-Glied R1/C1 und der Anschluss des IR-Empfängers, der die Signale der Fernbedienung empfängt.

Die Nano-Axe-Platine wird über ein Micro-USB-Kabel mit 5V Spannung versorgt, das wir am anderen Ende abgeschnitten und direkt in die Lochrasterplatine eingelötet haben. Möchte man das Board an einen PC anschließen, um z.B. die Firmware aufzuspielen, sie zu verändern oder Debug-Terminalausgaben anzusehen, kann man das weiße Micro-USB-Kabel entfernen und den Nano-Axe über ein anderes USB-Kabel mit dem PC verbinden. Dieser übernimmt dann automatisch auch die Spannungsversorgung des Nano-Axe-Boards. Aufgrund des relativ hohen Strombedarfs muss der LED-Streifen aber immer über das separate 5-V-Netzteil mit Strom versorgt werden. Der 5V-Anschlusspin an der Stiftleiste PL4 des Nano-Axe-Boards ist unbeschaltet und wird nicht genutzt. Durch diese Art des Aufbaus lässt sich mechanisch verhindern, dass die vom PC gelieferte 5-V-Spannung gegen die 5-V-Netzteilspannung treiben kann.

Den IR-Empfänger bringen wir unterhalb der Ziffer 6 am Uhrengehäuse an (Bild 4). Die Anschlussdrähte biegen wir um 90 Grad und

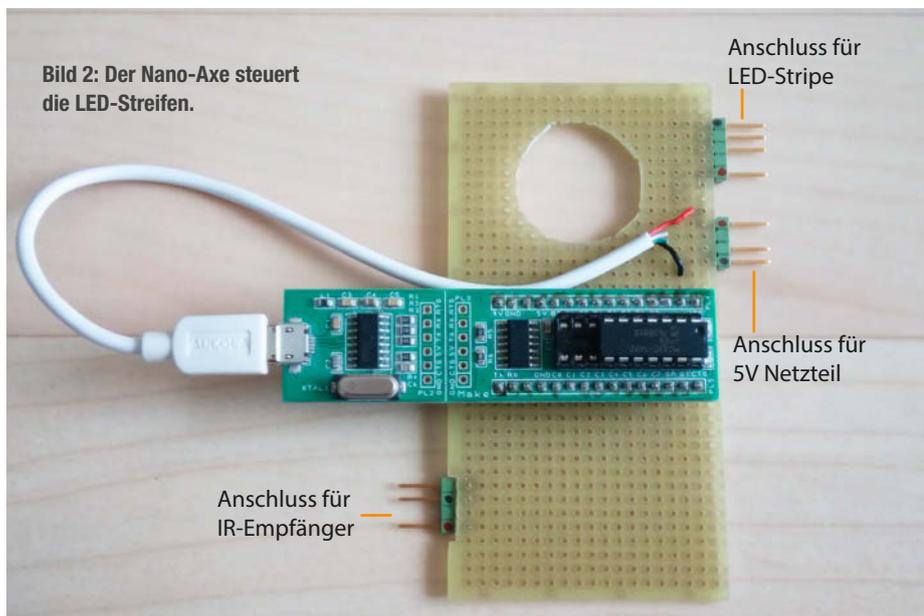


Bild 2: Der Nano-Axe steuert die LED-Streifen.

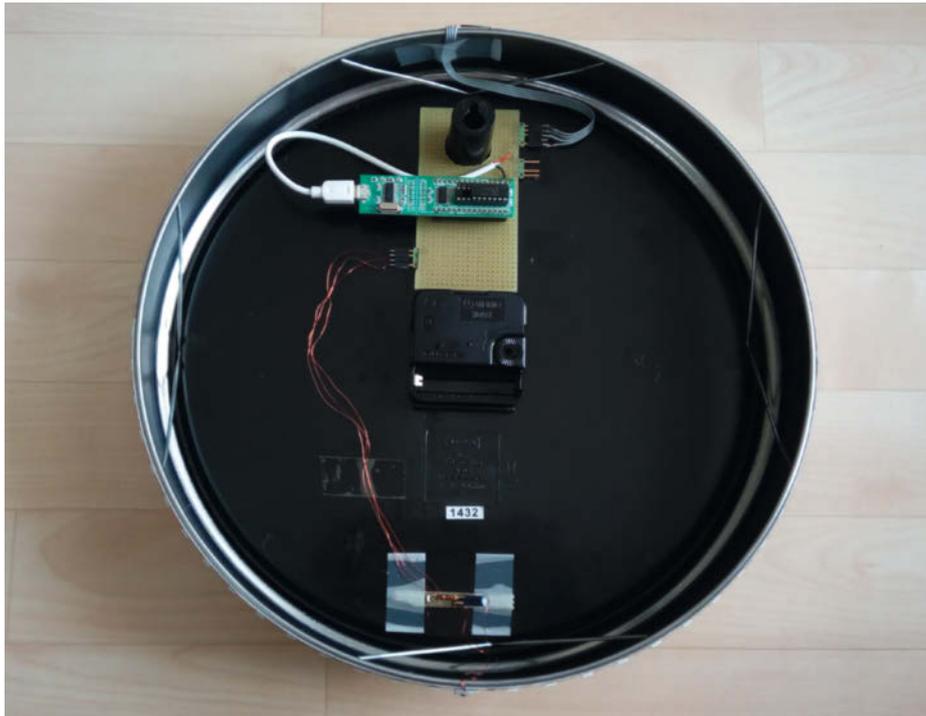


Bild 3: Steuerplatine (oben) und die Platine mit RC-Glied (unten) auf der Rückseite der Uhr

kürzen sie so weit, dass sie gerade in ein Stückchen Lochrasterplatine mit einzelnen Lötstellen passen, ohne auf der anderen Platinenseite herauszuschauen. Die Platine sägen wir danach so zurecht, dass sie die Größe des liegenden IR1261 hat. Dann löten wir die drei Anschlussbeinchen des IR-Empfängers auf der Lötseite der Platine in den Lötaugen fest. Außerdem löten wir noch drei Fädeldrähte an jeden Pin. Mit dieser Montageart bleibt die eigentliche Bestückungsseite der Lochrasterplatine frei und kann später plan auf dem Uhrengehäuse aufliegen (Bild 5).

Anschließend kleben wir mit doppelseitigem Klebeband ein Stück Papier um die IR-Platine, sodass nur die Frontseite des IR-Empfängers offen bleibt. Das Ganze muss man dann wiederum mit doppelseitigem Klebeband an der Uhr befestigen. Die Fädeldrähte legt man über den LED-Streifen hinweg seitlich auf die Rückseite der Uhr, fixiert sie dort mit Klebeband und verlötet sie an die Platine mit dem RC-Glied.

PICAXE-Firmware

Die Firmware haben wir mit dem PICAXE-Editor erstellt und sie ist an das Beispiel aus dem Artikel „Für Halloween: Beleuchtetes Kürbis-Gesicht mit Nano-Axe-Board“ (siehe Link in Kurzinfor) angelehnt. Mit der Zuordnung `symbol LED_DEFAULT_BRIGHTNESS` kann man die Helligkeit der LEDs anpassen. Viele LED-Streifen sind so hell, dass man sie dimmen sollte, um die Augen zu schonen. Der Wert

muss im Bereich zwischen 0 (aus) und 0xFF (maximale Helligkeit) liegen.

Um ein LED-Muster zu generieren, ruft man eine beliebige Kombination aus Makros mit den gewünschten LED-Daten solange nacheinander auf, bis das gewünschte Muster im RAM erzeugt ist:

– `SetLed(number, r, g, b)` setzt für eine beliebige LED mit der Nummer `number` im Bereich von 0 bis `Anzahl_LEDs - 1` den gewünschten Farbwert.



Bild 4: Der IR-Empfänger sitzt unterhalb der 6 des Zifferblatts.

Hier kommt Verstärkung



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
- ▶ **Inklusive 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.

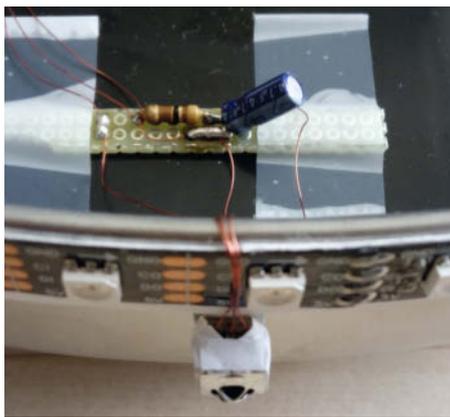


Bild 5: IR-Empfänger und RC-Glied sind mit Fädeldrähnen verbunden.

- `SetNextLed(r, g, b)` setzt für die nächste LED nach der zuvor gesetzten LED den gewünschten Farbwert.
- `SetNextMultipleLeds(quantity, r, g, b)` setzt für die nächsten `quantity` LEDs nach der zuvor gesetzten LED den gewünschten Farbwert.

Wenn man diese Makros verwendet, wird automatisch der bei `LED_DEFAULT_BRIGHTNESS` angegebene Wert für die Helligkeit mit eingerechnet. Der Befehl `gosub updateLeds` sendet das im RAM vorhandene Muster an den LED-Streifen. Das nullt auch die Nummer der nächsten zu setzenden LED. Es ist jedoch darauf zu achten, dass man die Makros `SetNextLed` und `SetNextMultipleLeds` nur solange aufruft, bis die maximal zulässige Anzahl an LEDs gesetzt wurde, da ansonsten ein Überlauf im RAM erfolgt und es zum Absturz der Firmware kommen kann.

In der Beispiel-Firmware ist die Sekundenanzeige direkt nach dem Einschalten des Mikrocontrollers aktiv. In dieser Konfiguration werden zunächst der Reihe nach alle LEDs im Sekundentakt eingeschaltet, bis sie nach einer Minute alle leuchten – jede fünfte rot, die restlichen grün. Anschließend schaltet das Programm die LEDs der Reihe nach im Sekundentakt wieder aus, bis nach einer Minute keine LED mehr leuchtet. Danach wiederholt sich der Ablauf.

Mit der IR-Fernbedienung kann man weitere Lichteffekte auswählen. Mit den Tasten 1 bis 9 stehen neun Lichteffekte zur Verfügung:

- Taste 1: Automatik-Mode. Hier laufen die Lichteffekte der Tasten 2 bis 9 zyklisch nacheinander ab.
 - Taste 2 bis 7: Diese Muster zeigen verschiedene Lichtpunkte und -ketten, die nacheinander in den Farben Rot, Gelb, Grün, Cyan, Blau und Magenta am Streifen entlangwandern.
 - Taste 8: Lässt die LEDs stufenweise heller dimmen, nacheinander in den Farben Rot, Gelb, Grün, Cyan, Blau und Magenta.
 - Taste 9: Ein Regenbogenverlauf, der langsam den LED-Streifen entlangwandert.
 - Taste 0: Sekundenanzeige
- Mit den Vorlauftasten rechts und links kann man die Lichtprogramme auch schrittweise nacheinander auswählen.

Für jeden Lichteffekt gibt es eine eigene Funktion namens `ledprog_xxx`. Diese wählt die `main`-Routine per `select case` abhängig von der Variablen `progNumber` aus. Für eigene Lichteffekte kann man diese Auswahl ergänzen und jeweils eine eigene Funktion anlegen und aufrufen. Um diese zum Automatik-Mode hinzuzufügen, muss man die Funktion `ledprog_autoMode` mit einem weiteren Makroaufruf von `REPEAT_LEDPROG` ergänzen und die gewünschte Funktion und Wiederholungszahl angeben. Als lokale Variablen kann man die Bytes `localByte1` bis `localByte3` sowie die Worte `localWord1` bis `localWord3` verwenden, die bereits in den vorhandenen Lichteffekten benutzt werden. Ebenso kann man die Variablen `state`, `color_r`, `color_g` und `color_b` dort verwenden. Nach dem Aufruf von `gosub updateLeds` startet das Makro `CheckForProgChange`. Bei einem Tastendruck auf der Fernbedienung wechselt das Programm zum nächsten Effekt.

Infrarot-Kommunikation

Viele Low-Cost-Fernbedienungen verwenden das Infrarot-Protokoll der Firma NEC (Bild 6), um mit Empfängern zu kommunizieren. Die dafür verwendete IR-Pulsfrequenz beträgt 38 kHz. Eine Folge solcher Pulse bezeichnet man auch als Burst. Am Beginn einer Übertragung sendet das Gerät zunächst einen 9 ms langen Burst, gefolgt von einer 4,5 ms langen Pause.

Anschließend übermittelt es 16 Adress- und 16 Datenbits. Hierbei hat ein Bit für die Kodierung einer logischen Eins die Länge von 2,25 ms und für eine logische Null die Länge von 1,12 ms. Dabei wird jeweils zunächst ein 560 µs langer Burst gesendet, gefolgt von einer Pause von 1690 µs (logische 1) bzw. 560 µs (logische 0). Die Pause legt also fest, ob es sich um ein Eins- oder Null-Bit handelt.

Der Tastencode wird nur einmal ausgesendet. Hält man die Fernbedienungstaste aber gedrückt, sendet die Fernbedienung zyklisch ca. alle 110 ms einen Repeat-Code. Dieser besteht aus einem 9 ms langen Burst, gefolgt von einer 2,25 ms langen Pause sowie einem 560 µs langen Burst. Wer mehr Details zum NEC-Protokoll benötigt, findet sie über den Link in der Kurzinfo.

In der Firmware konfiguriert `setint` einen Interrupt auf die fallende Flanke am Pin C0, an dem der Ausgang des IR-Empfängers angeschlossen ist. Wenn der IR1261 einen Burst erkennt, wird der Ausgang auf einen Low-Pegel und bei einer Pause auf einen High-Pegel gezogen. Die Routine `interrupt` wertet dann in einer Schleife mit dem `pu1sin`-Befehl die jeweilige Zeitdauer des High-Pegels aus und speichert ihn per `bptr` nacheinander als 16-Bit-Werte in den RAM des PICAXE. Dies geschieht so lange, bis entweder `pu1sin` eine Null zurückliefert, da ein Timeout ($65536 \cdot 1,25 \mu s \approx 82 \text{ ms}$) erreicht wurde oder bis eine bestimmte Anzahl an Werten in den RAM geschrieben wurde.

Eine passende Fernbedienung kann man z.B. mit dem Suchbegriff „ir remote nec“ bei eBay finden. Teilweise sind diese auch im Set mit dem IR-Empfänger HX1838 erhältlich, der kompatibel zum IR1261 ist. Meistens haben diese Fernbedienungen 21 Tasten, die in einer 3 x 7 Matrix angeordnet sind. Die gesendeten IR-Codes können je nach Modell unterschiedlich sein. Diese lassen sich jedoch einfach mit der PICAXE-Firmware ermitteln, per Terminal ausgeben und entsprechend im Quellcode anpassen.



Bild 6: IR-Fernbedienung mit NEC-Protokoll

Tastencodes ermitteln

Die IR-Tastencodes sind über die Defines `IR_xxx` zugeordnet. Falls eure Fernbedienung nicht funktioniert, weil sie andere Codes liefert als im Beispielprogramm, muss man diese zunächst ermitteln. Hierzu aktiviert man das Makro `DEBUG_PRINT` ganz oben im Code, so dass es den Befehl `ser_txd` anstatt eines Strichpunkts als Kommentarzeichen im Quelltext einfügt. Um die Terminalausgaben zu aktivieren, muss der Code also wie im Listing aussehen.

Anschließend prüft man über den Check-Button im PICAXE-Editor die Firmware auf Syntaxfehler und flasht sie per Programmierkabel auf den PICAXE. Wenn man danach ein Terminal mit 38400 Baud öffnet, sollte bei jedem Tastendruck auf der Fernbedienung eine Terminalausgabe mit den entsprechend gemessenen Pulszeiten sowie dem entsprechenden Tastencode erscheinen (Bild 7). Die Anzahl bei `Cnt` sollte 32 sein. Unter `Data` wird der gewünschte Code angezeigt. Das NEC-Protokoll überträgt 16-Bit-Daten, wobei das erste Datenbyte den inversen Inhalt des zweiten hat. Hier interessiert uns nur der zweite Wert, der dann bei dem entsprechenden `define IR_xxx` im Quellcode eingetragen wird. Möchte man zusätzlich noch die Adresse der Fernbedienung auswerten, so steht diese in der Variablen `irAddr` zur Verfügung. Nachdem man die benötigten Tasten ermittelt hat, ändert man `DEBUG_PRINT` wieder im Code und überträgt die Firmware erneut auf den PICAXE. Jetzt lässt sich der LED-Streifen auch per Fernbedienung regeln. —akf

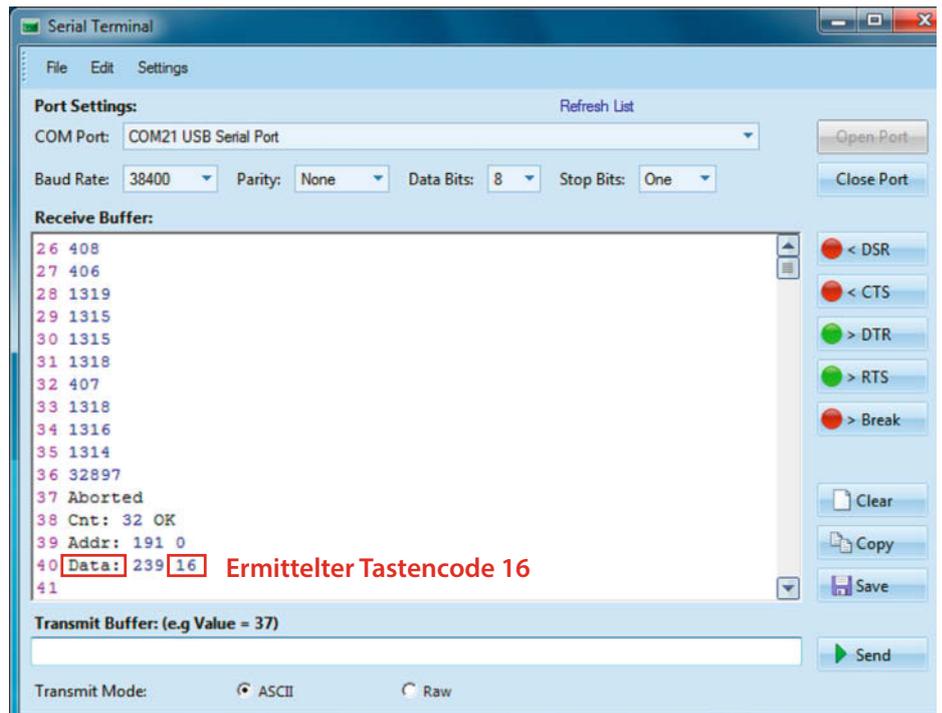


Bild 7: Die DEBUG-Ausgabe im Terminal gibt den Code der gedrückten Taste aus.

Debug-Zeilen anpassen

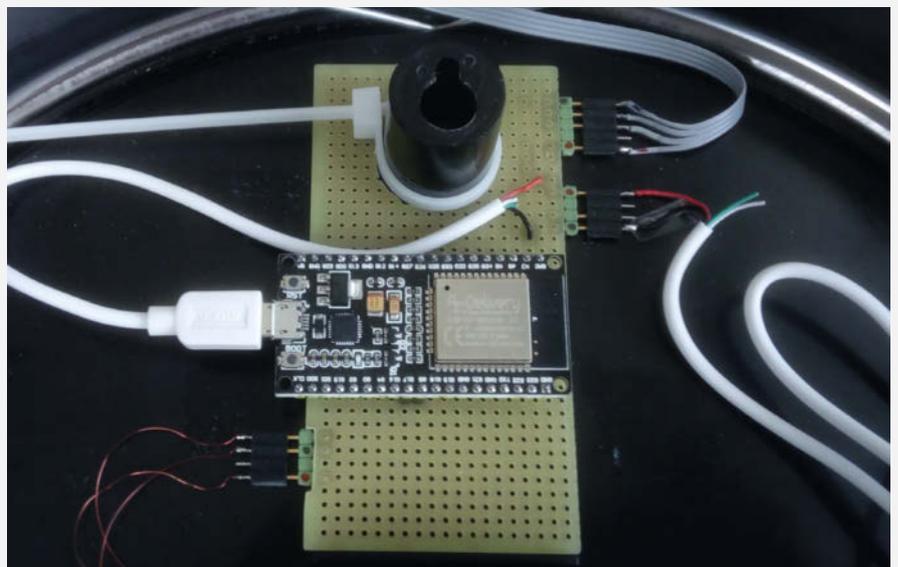
```
; fuer Debug-Terminalausgaben das Semikolon vor folgender Zeile entfernen:
#define DEBUG_PRINT ser_txd

; fuer Debug-Terminalausgaben das Semikolon vor folgende Zeile setzen:
#define DEBUG_PRINT ;
```

ESP32 als Alternative

Wer lieber eine WLAN-Steuerung per WLED-Software über den PC oder die App haben möchte, kann alternativ auch ein ESP32-Board statt eines Nano-Axe verwenden. Wir haben unseren Aufbau mit einem ESP 32 Dev Kit C ausprobiert und die Steuerelektronik in gleichem Stil auf einer Lochrasterplatine aufgebaut. Die Datenleitung DI des APA102-LED-Streifens haben wir mit dem GPIO33 und die Clock-Leitung CI mit dem GPIO32 des Mikrocontrollers verbunden. Zur Konfiguration und Bedienung von WLED gab es bereits in der Make 2/23 einen ausführlichen Artikel von Carsten Wartmann (siehe Link in Kurzinfo).

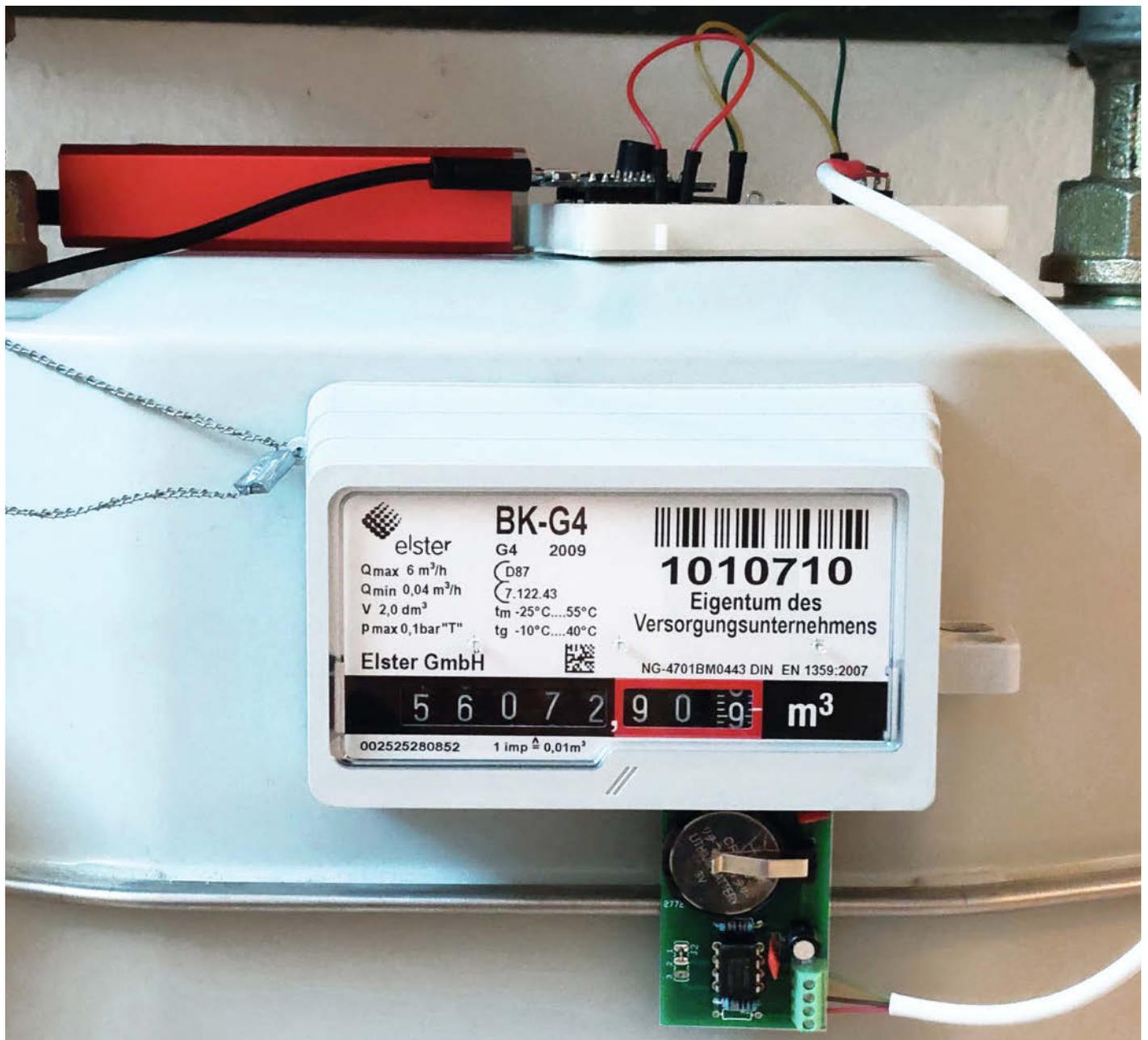
Alternativ kann man auch einen ESP32 verwenden.



Gaszähler auslesen und Daten sichern

Im Gegensatz zu Smart Metern, die elektrische Arbeit messen, speichern herkömmliche Gaszähler den gemessenen Verbrauch mechanisch in einem Zählwerk. In der Make 3/23 haben wir bereits gezeigt, wie man einen Gaszähler selber smart machen kann. Hier präsentieren wir eine Lösung, die auch ohne WLAN die Werte speichert.

von Rainer Goldelius



Viele Gaszähler geben pro Umdrehung des Zählwerks einen magnetischen Impuls ab. Meist ist dies ein Impuls pro 10 Liter. Durch Zählen dieser Impulse kann der Verbrauch ermittelt werden, jedoch nicht der absolute Wert, sondern nur der Wert seit Beginn der Zählung. Um den Verbrauch zu ermitteln, ist es daher notwendig, diese Impulse sicher zu speichern.

ChatGPT hat mir geholfen, den CMOS-Clock-Chip (RTC) PCF8583 von Philips zu finden, der auch als Zähler verwendet werden kann und 240 Byte RAM zur Verfügung stellt. In diesem Speicherbereich können wir Informationen ablegen und so sicherstellen, dass auch dann gezählt wird, wenn das WLAN nicht aktiv ist oder das ESP sogar ausfällt.

Zur Erfassung der Zählimpulse wird ein Hallsensor vom Typ AH3661UA verwendet. Bei diesem Sensortyp sind Maßnahmen zur Kontaktentprellung, wie sie bei Reedkontakten häufig auftreten, nicht erforderlich. Die gemessene Stromaufnahme der Musterplatine lag zwischen 8 und 14 μA , eine 3-V-Knopfzelle (CR2032) reicht also für eine längere Zeit als Backup für die gesammelten Verbrauchswerte aus.

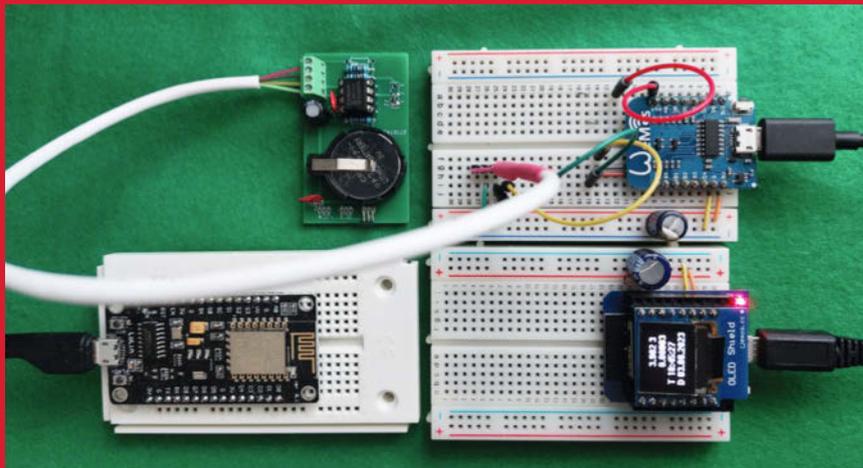
Der RTC-Baustein hat eine I²C-Schnittstelle und kann einfach über einen ESP8266 ausgelesen werden, der dann auch die weitere Verarbeitung übernimmt und die Daten über das lokale WLAN per MQTT zur Auswertung und Visualisierung weiterleitet.

Im RAM des RTC-ICs werden der letzte Zählerstand, die Uhrzeit und das Datum der Auslesung gespeichert. Bei jedem Auslesen der Werte wird die Anzahl der Impulse seit der letzten Messung sowie die Zeit zwischen zwei Zugriffen über I²C berechnet. Zusätzlich werden der Zählerstand bei Inbetriebnahme, der Zählerstand zu Beginn der Abrechnungsperiode und zwölf weitere Werte für den jeweiligen Monat gespeichert. Diese Werte können über MQTT mit dem Subtopic „Init“ initialisiert werden.

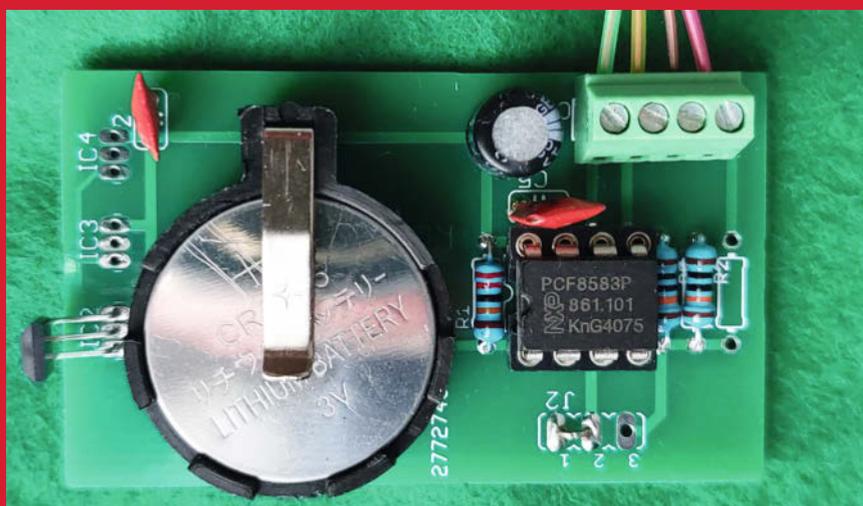
Mangels Stromanschluss wird der ESP von einer kleinen Powerbank versorgt, zur Stromersparnis in den Deep Sleep versetzt und nur alle zwei Minuten aufgeweckt. Die Anlage läuft seit Mai ohne Probleme, auch ein probeweises Abstecken des Controllers für längere Zeit hat zu keinem Datenverlust geführt. Es geht nur die zeitliche Zuordnung der Impulse verloren, diese werden aber dem Zeitraum zwischen dem Ausstecken und dem Wiedereinstecken neu zugeordnet.

Ich habe die Software aus Make 3/23 als Subscriber für die gezählten Impulse modifiziert und auch damit läuft das System einwandfrei. Die Quellen für die angepasste Software und die Platine (Eagle) sind in meinem GitHub zu finden. —caw

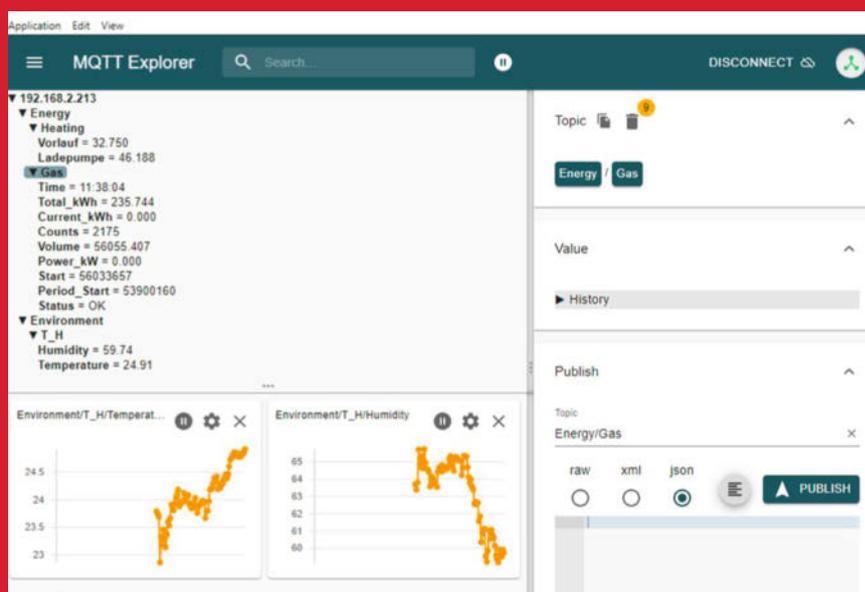
► github.com/Rainer-G/Gas_Meter/



Oben rechts der ESP8266 mit dem Lesekopf, unten links ein ESP8266 als MQTT-Broker, daneben ein externes Display zur Anzeige der Verbräuche



Der Lesekopf mit dem Hallsensor links auf der selbst entworfenen Platine



Die Ergebnisse im MQTT-Explorer

Upcycling-Bartop: Arcade-Spaß mit Scratch-Spielen

Aus ein paar Holzresten, einem ausrangierten Computer, etwas Farbe und im Drogeriemarkt ausgedruckten Bildern wird in Kombination mit einem USB-Arcade-Kit ein Bartop, auf dem sich in Scratch programmierte Spiele spielen lassen.

von Markus Mauch



Anleitungen zum Bau von Arcade-Automaten findet man viele im Netz. Auch in der Make 4/20 (Bartop Arcade mit Raspberry Pi) wurde der Bau eines Bartops mit allen zugehörigen Komponenten ausführlich thematisiert. Inzwischen kann man auch ein fertig konfektioniertes Gehäuse oder gleich die komplette Arcade-Maschine bestellen.

In fast allen Fällen läuft auf diesen Geräten Software von der Stange, beispielsweise RetroPie oder ähnliche Alternativen, die dann in früheren Jahren liebgezwonnene Spiele-Klassiker noch einmal aufleben lassen.

Mit diesem Bartop bin ich einen etwas anderen Weg gegangen: Das Gerät an sich ist in einem Upcycling-Prozess aus bereits vorhandenen Materialien entstanden. Einzig für das DIY-Kit mit den Buttons und dem Joystick sowie für die Fotos mit den Bildern aus dem Spiel habe ich frisches Geld ausgegeben.

Das Spiel für meine Arcade-Maschine ist nicht irgendein Klassiker, der wieder mit ein paar Klicks zum Leben erweckt worden ist, sondern wurde von mir mit allen Details in Scratch selbst programmiert. Scratch ist eine frei verfügbare, blockbasierte Programmierumgebung, die besonders bei Kindern und Jugendlichen, aber auch vielen Erwachsenen sehr beliebt ist. Gemeinsam mit meinen Schülerinnen und Schülern arbeite ich im Unterricht häufig damit. Mit diesem Bartop wollte ich eine ganzheitliche Brücke schlagen: Von ersten einfachen Algorithmen, die den Spieler eine Figur über den Bildschirm steuern lassen, hin zur fertigen Spielekonsole, die vieles bietet, was das Zockerherz höher schlagen lässt.

Inzwischen habe ich sowohl für meine Gartenhütte und meine Schule, als auch für das Medienzentrum, in dem ich arbeite, einen Arcade-Automaten nach ähnlichem Muster gebaut.

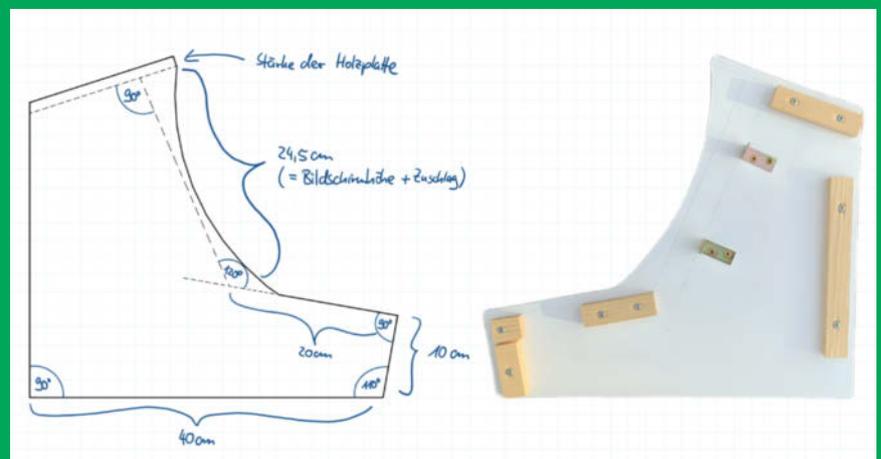
Das Bartop setzt sich aus drei Bestandteilen zusammen: dem aus Holz zusammengezwimmerten Gehäuse, einem alten Laptop, bei dem ich Bildschirm und Rechner getrennt habe, und einem USB-Arcade-Kit. Maßgebend für die einzelnen Gehäuseteile sind die Breite und die Höhe des verwendeten Bildschirms. Bei der Konstruktion der Seitenteile habe ich so lange mit Bleistift, Lineal, Geodreieck und improvisiertem Zirkel experimentiert, bis ich mit der Form zufrieden war.

Das Spiel, das auf meinem Bartop läuft, ist ein klassisches Weltraum-Arcadespiel. Der Spieler steuert ein Raumschiff und muss sich mit Laserstrahlen gegen allerlei Gefahren wie Asteroiden und angreifenden Ufos wehren und sich zu guter Letzt mit einem gefährlichen Endgegner auseinandersetzen. Sowohl für den Bau des Bartops als auch für die Programmierung des Spiels habe ich ausführliche Tutorials erstellt, welche verlinkt sind.

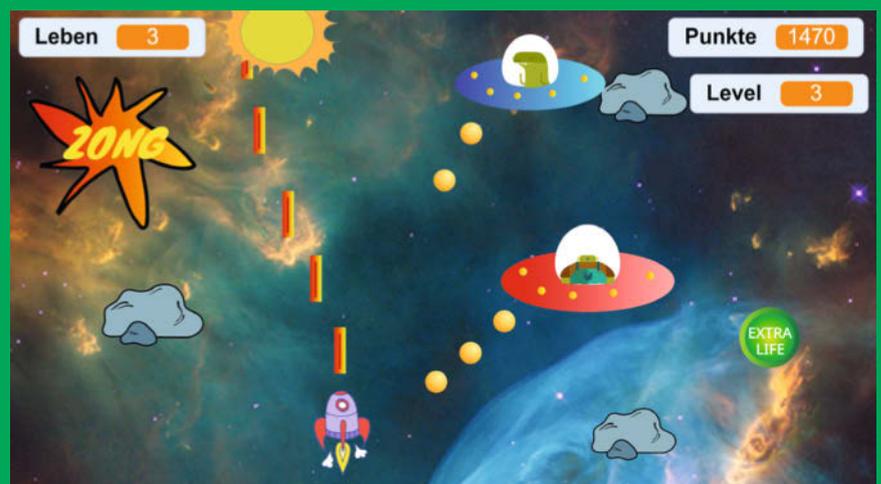
Das Bartop hat bereits mehrere Tage der offenen Tür in verschiedenen Einrichtungen,



Der Weg zum Bartop



Skizze und Abbildung eines Seitenteils



Das in Scratch selbst programmierte Weltraum-Arcadespiel

eine Messeausstellung sowie diverse Besuche von kleineren und größeren Kindergruppen aus der Nachbarschaft in unserer Gartenhütte überstanden und schon für zahlreiche leuchtende Augen gesorgt.

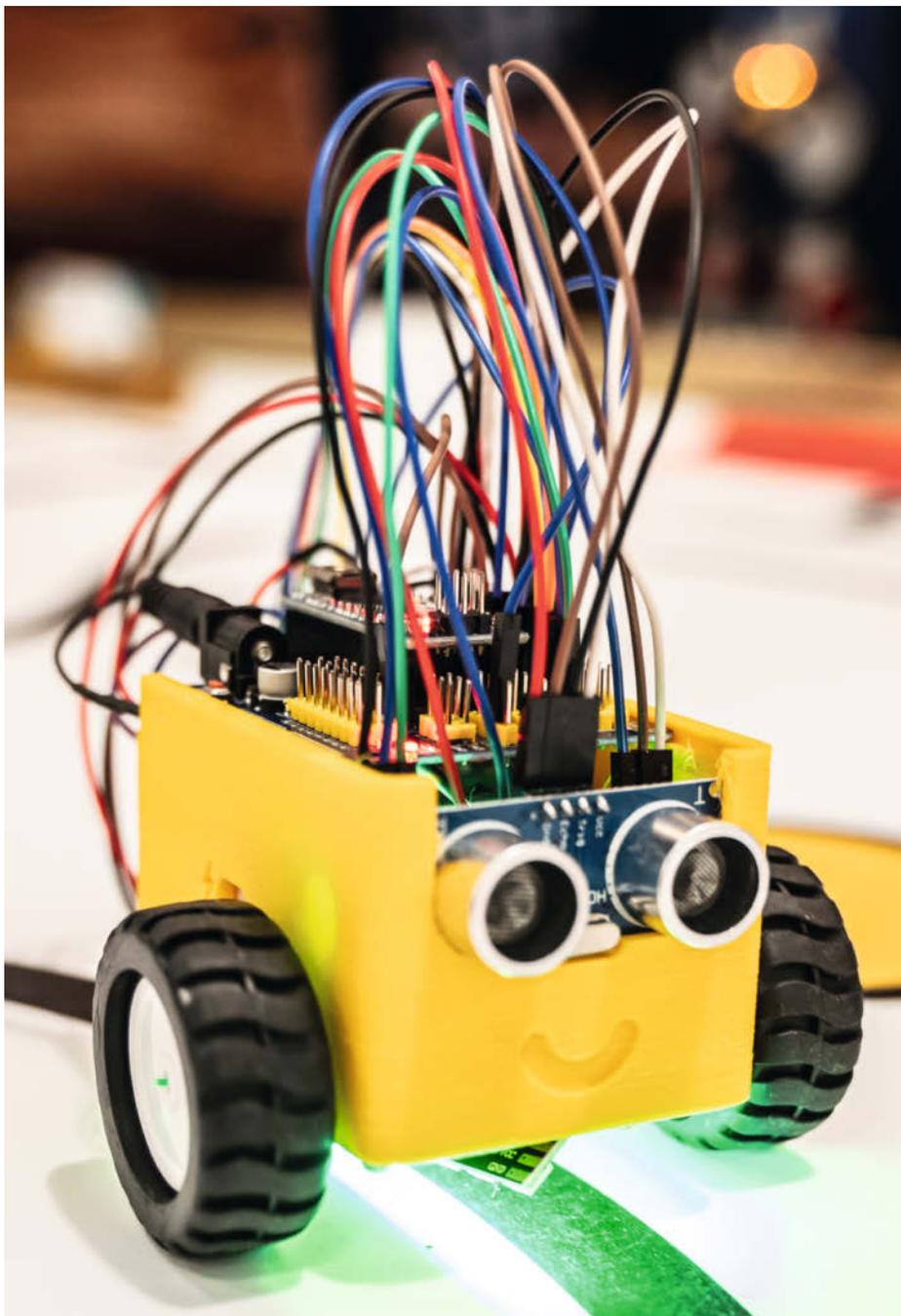
Eine vollständige Anleitung gibt es unter dem folgenden Link. —das

► <https://youtu.be/jSwwjxMcsb4>

Roboter fürs Klassenzimmer

Wer programmieren lernen möchte, kommt um die Theorie nicht herum. Wie man sie vermittelt, steht auf einem ganz anderen Blatt. Mit einem praxisnahen Konzept bringt dieses Projekt den Profis von morgen das Physical Computing näher.

von Marco Düvelmeyer und Ákos Fodor



Wenn Marco Düvelmeyer an seiner Schule Informatik unterrichtet, lernt man bei ihm nicht nur Programmieren, sondern auch CAD-Konstruktion, 3D-Druck und Elektrotechnik. Damit diese Disziplinen sinnvoll ineinandergreifen, hat er ein didaktisches Konzept ausgearbeitet, das in einem kleinen Roboter auf Arduino-Basis seine Form gefunden hat. Dieser ist mit Sensoren und Motoren ausgestattet und lässt sich mit dem blockbasierten Open Roberta programmieren.

Als Bauanleitungen verwendet Düvelmeyer selbst erstellte Videos, die man auf dem YouTube-Kanal „Technikwerkstatt 4.0“ findet. Eine begleitende Lernlandkarte, die man sich über einen Link in der Kanalinfo herunterladen kann, gibt einen Überblick über das umfangreiche Projekt. Sie zeigt anschaulich, welche unterschiedlichen Phasen (Lernsituationen) man auf dem Weg durchläuft und führt über auswählbare interaktive Elemente zu weiteren Informationen, Videos, Zeichnungen und Programmieranleitungen.

Da der Roboter in erster Linie für den Unterricht konzipiert ist, kommt Software zum Einsatz, die für Lehrzwecke kostenfrei erhältlich ist. Das ermöglicht auch anderen Schulen, den kleinen Roboter ohne Zusatzkosten in den Unterricht einzubinden – vorausgesetzt, die notwendige IT-Infrastruktur ist vorhanden. Um mit seinem Projekt ein noch größeres Publikum zu erreichen, setzt Düvelmeyer für seine CAD-Anleitungen zukünftig auf Autodesk Fusion 360 und hat auch schon damit begonnen, seine bisherigen Videos neu zu produzieren. Damit kann sich irgendwann jeder, auch außerhalb des Lehrumfelds, den Roboter nachbauen, denn Autodesk bietet für Bastler eine kostenfreie, nicht kommerzielle Fusion-360-Lizenz an.

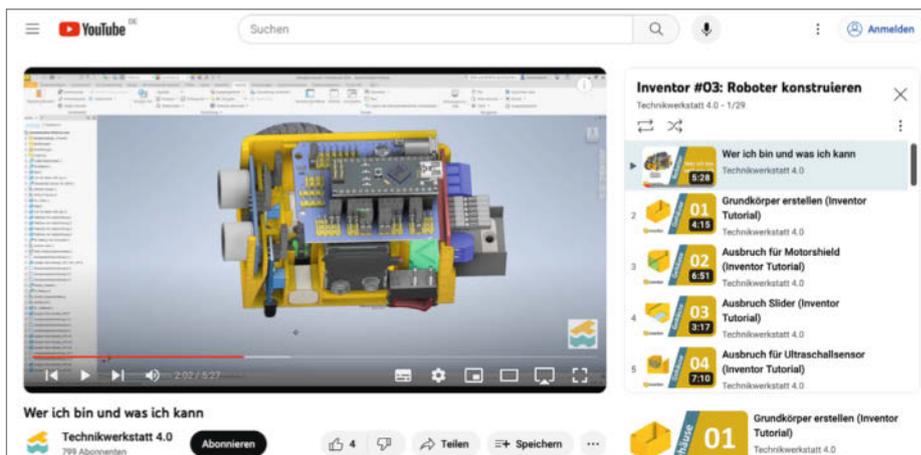
Make: Wie bist du auf dieses Konzept gekommen?

Düvelmeyer: Ich habe vor einigen Jahren an einer Berufsschule in Hamburg als Lehrer gearbeitet und im Bereich Werkzeugmecha-

nik unter anderem auch CAD unterrichtet. Als gelernter technischer Produktdesigner habe ich mich auf diesen Unterricht gefreut, aber schnell gemerkt, dass es eine der anstrengendsten Unterrichtseinheiten überhaupt ist. Denn sobald die Schülerinnen und Schüler (SuS) einen Klick verpassen, kommt Frust auf. Weil CAD-Programme so komplex sind, ist es unvermeidlich, dass 25 SuS hier und da auch mal etwas verpassen. Deshalb habe ich überlegt, wie ich mir selbst am liebsten Programme beibringe, und das ist immer schon mit YouTube gewesen.

Damals musste ich das Programm MegaCAD unterrichten, aber weil es eher ein Nischenprogramm ist, gab es nicht viele Videos dazu, und vor allem hat mir immer ein roter Faden in den vorhandenen Tutorials gefehlt. Deshalb habe ich einfach selbst welche aufgenommen.

Die erste Stunde, in der meine SuS mit den Videos arbeiteten, lief dann plötzlich völlig anders. Alle hatten Kopfhörer auf, konnten zurückspulen, wenn ihnen etwas zu schnell ging und waren vertieft in das



Screencasts sind für CAD-Anleitungen besonders praktisch, weil man mitverfolgen kann, wie ein Modell entsteht.

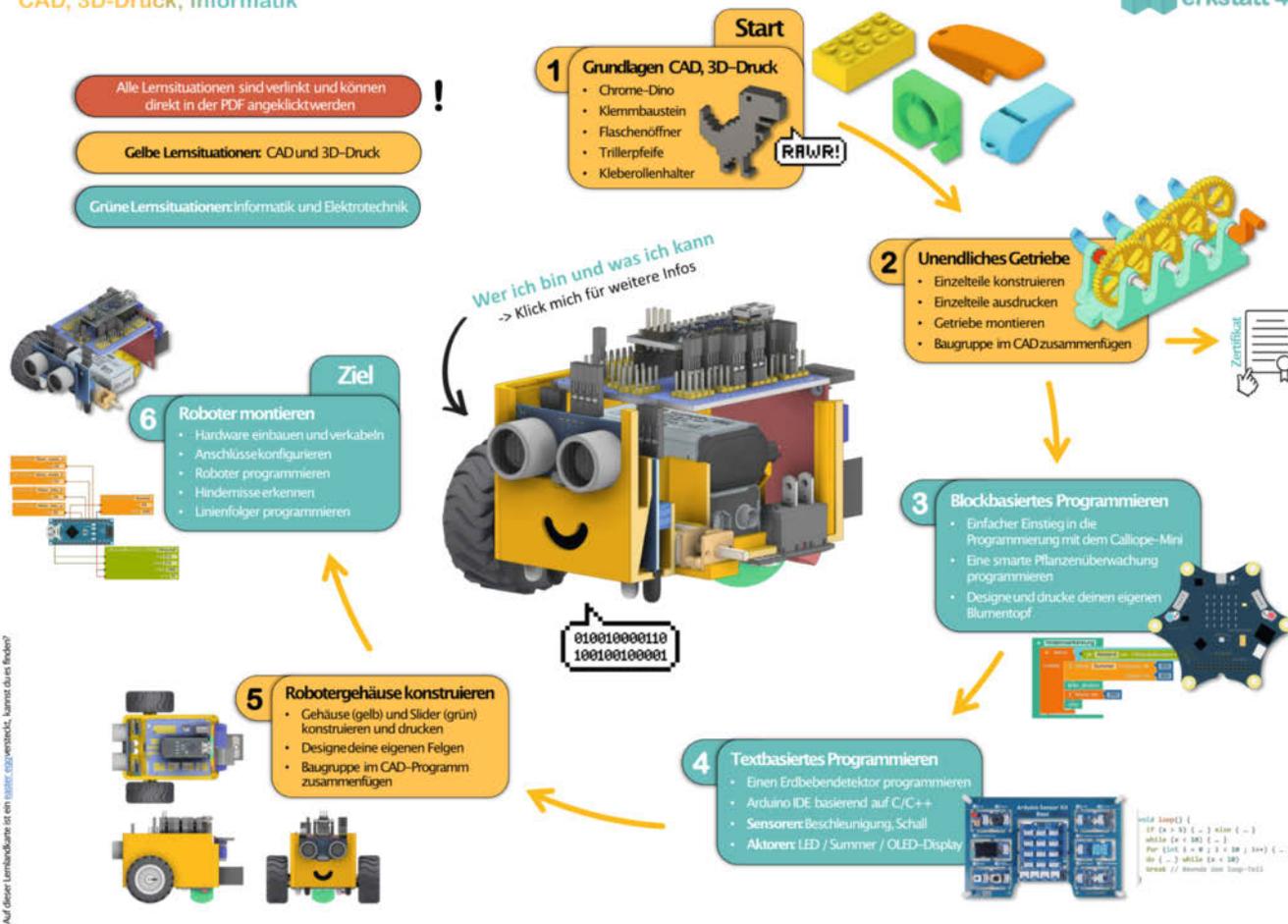
Programm. Viele meldeten mir zurück, dass sie so am besten lernen könnten.

Make: Und wann kam es zu dem Roboter und der Lernlandkarte?

Düvelmeyer: Das war, als ich an meine jetzige Schule (die Thomas-Morus-Schule in Osnabrück) gekommen bin, wo ich vor ca. vier Jahren den Wahlpflichtkurs Industrie 4.0 eingeführt habe. Da hatte ich die Idee,

In sechs Schritten zum eigenen Roboter

CAD, 3D-Druck, Informatik



Die Lernlandkarte leitet über verschiedene Lemsituationen zum Ziel.

Auf dem Weg lernt man, wie man den Roboter in CAD baut, 3D-druckt und programmiert.



Der kleine Roboter während der Bauphase: Sobald die Elektronik funktioniert, muss sie nur noch im Gehäuse Platz finden.



Für die Schülerinnen und Schüler ist es ein besonderes Erlebnis, den Roboter selbst zusammenzubauen und mit nach Hause zu nehmen.

dass meine SuS selber einen Roboter entwickeln und dabei CAD und Programmieren lernen sollten.

„Weil CAD-Programme so komplex sind, ist es unvermeidlich, dass 25 Schülerinnen und Schüler hier und da auch mal etwas verpassen. Deshalb habe ich überlegt, wie ich mir selbst am liebsten Programme beibringe, und das ist immer schon mit YouTube gewesen.“

Als ich anfang, dafür neue CAD-Tutorials aufzunehmen, wurde mir aber schnell klar, dass die SuS sich bei einem so großen Projekt wie der kompletten Entwicklung eines Roboters auf dem YouTube-Kanal nicht zu rechtfinden werden. Man kann zwar Playlisten anlegen, aber diese nicht so sortieren und untereinander verknüpfen, wie ich es mir vorgestellt hatte.

Deshalb begann ich im ersten Schritt, interaktive Arbeitsblätter zu erstellen, durch die sich die SuS einfach durchklicken können und wo ich dann die entsprechenden Videos und Playlisten verlinkt habe. Die Arbeitsblätter verteilte ich dann über den Schulserver, was im Windows-Dateisystem schnell unübersichtlich wurde. Viele PDF-Dokumente in verschiedenen Unterordner führten immer wieder zu Problemen. Es

musste also eine vernünftige Struktur her: die Lernlandkarte.

Make: Wie hast du die Lernlandkarte konzipiert?

Düvelmeyer: Ich habe mir überlegt, wie die Arbeitsmaterialien aussehen müssten, damit ich selber gerne mit ihnen arbeite und meine bisherige Unterrichtserfahrung hinzugezogen. Daraus entwickelte ich die Kriterien für die Lernlandkarte. Einerseits sollte man schnell erfassen können, worum es in dem Projekt geht. Gleichzeitig wollte ich die Lernlandkarte mit Bildern und Grafiken optisch attraktiv gestalten und sowohl Mädchen als auch Jungs ansprechen. Weiterführende Dokumente habe ich dann in dem Dokument verlinkt, sodass die Lernlandkarte einen guten Ausgangspunkt bildet. Mittlerweile befindet sie sich schon in der 14. Version.

Diese und die mit ihr verlinkten Arbeitsmaterialien habe ich dann online gestellt, sodass man alles mit jedem Endgerät und von überall abrufen kann, ohne dass sich die SuS irgendwo anmelden müssen. Das machte den Unterricht nochmal schlagartig übersichtlicher und strukturierter.

Make: Und fand es auch Anklang bei den Schülerinnen und Schülern?

Düvelmeyer: Ja, ich habe gemerkt, dass man Themen wie Mathematik, Physik und Informatik durch ein Projekt wie die Konstruktion eines Roboters viel lebensnäher unterrichten kann. Außerdem hatten die SuS viel Spaß an CAD. Ihre eigenen Projekte virtuell am PC zu designen und dann mit

dem 3D-Drucker in die Realität „teleportieren“ zu können, hat für viele etwas Magisches. Auch wenn die Einstiegshürde hoch ist, überwiegt die Motivation, das eigene Design mit nach Hause nehmen zu können.

Make: Wie hat dich das Projekt selbst weitergebracht?

Düvelmeyer: Durch dieses Unterrichtskonzept bin ich mit vielen anderen Lehrkräften in Kontakt gekommen, die ähnliche Interessen haben und deren Wissen in einigen Teilgebieten weit über mein eigenes hinausgeht. Über eine Fortbildung zum Informatiklehrer und ein Netzwerk für Techniklehrer habe ich zwei Kollegen kennengelernt, die mich bei der Entwicklung des Roboters unterstützt haben. Dadurch habe ich selbst viel Neues dazugelernt. Außerdem musste ich mich auch mit Design, Farbtheorie und Videoschnitt vertraut machen.

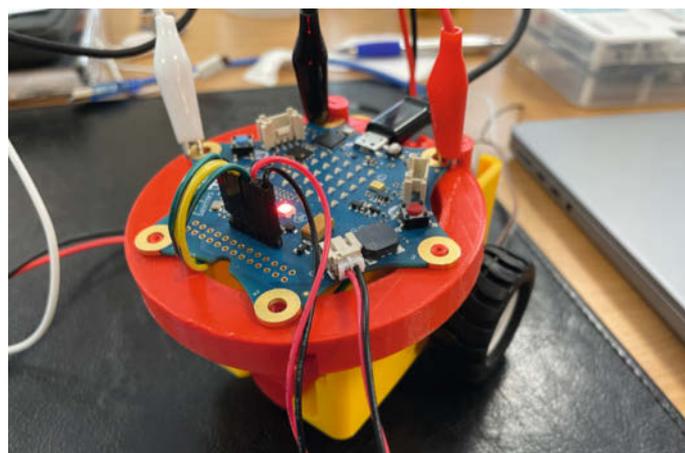
Make: Welche Vorteile siehst du in deinem Konzept?

Düvelmeyer: Das Unterrichtskonzept befindet sich in stetiger Entwicklung. Wenn von den SuS eine Frage kommt wie „Warum müssen wir denn jetzt unbedingt etwas

„Ihre eigenen Projekte virtuell am PC zu designen und dann mit dem 3D-Drucker in die Realität ‚teleportieren‘ zu können, hat für viele etwas Magisches.“



Am Stand der Technikwerkstatt 4.0 auf der Maker Faire zog der kleine Roboter munter seine Kreise.



Auch eine Calliope-Variante des Projekts ist gerade Arbeit. Der größere Mikrocontroller braucht etwas mehr Platz im Gehäuse.

über Strom und Spannung lernen?“, ist das ein Signal, um das Konzept an der entsprechenden Stelle noch mal anzupassen. Gleichzeitig sehen die SuS schnell ein, warum sie erstmal einen kleinen Theorie-Input zur Elektrotechnik benötigen, wenn sich ihre Motoren gar nicht oder in die falsche Richtung drehen. Dabei bekommen sie nicht nur einen Einblick in verschiedene Berufsfelder, sondern müssen über die Theorie hinaus auch viele physikalische und mathematische Grundlagen anwenden und können dabei beobachten, wofür diese in der Praxis gebraucht werden und wie sich eine Änderung der Parameter auf ihr Programm oder den Roboter auswirkt.

Make: Wie ist die bisherigen Resonanz von anderen Schulen?

Düvelmeyer: Die Schulen, von denen ich weiß, dass sie dieses Konzept im Unterricht einsetzen, finden besonders praktisch, dass der Unterricht nicht mehr wirklich vorbereitet werden muss. Kolleginnen und Kollegen, die sich noch nicht mit dem Thema CAD auskennen, können durch die Video-Tutorials selbst niederschwellig in das Thema einsteigen und den SuS im Anschluss einfach den YouTube-Kanal und die Lernlandkarte zeigen. Was vielen Schulen auffällt ist, dass die SuS von Anfang an sehr interessiert und motiviert dabei sind. Die Themen CAD, 3D-Druck und Robotik sind bei den Kids aktuell im Trend. Vielleicht liegt das auch daran, dass sie zum Teil deutsch- und englischsprachigen Makern auf YouTube folgen (etwa Tomary).

Die grünen Lernsituationen zum Thema Informatik und Elektrotechnik waren in der Vergangenheit weniger gefragt, da die meisten Schulen mit der Landkarte erstmal in das Thema CAD und 3D-Druck einsteigen

wollten. Mit der Einführung des Pflichtfaches Informatik in diesem Schuljahr hat sich das allerdings geändert. Viele Schulen sind noch nicht wirklich auf das Pflichtfach Informatik vorbereitet und es kommen vermehrt Rückfragen dazu. Vielleicht kann dieses Unterrichtskonzept dem extremen Mangel an Informatiklehrerinnen und -lehrern ein bisschen entgegenwirken.

„Was vielen Schulen auffällt ist, dass die Schülerinnen und Schüler von Anfang an sehr interessiert und motiviert dabei sind. Die Themen CAD, 3D-Druck und Robotik sind bei den Kids aktuell im Trend.“

Die Informatikthemen aus dem niedersächsischen Curriculum für das Pflichtfach Informatik, wie Programmieren, Codierung, Daten und ihre Spuren, Aufbau von Netzwerken etc. sind für viele Schulen noch echtes „Neuland“. Wir selber unterrichten vieles davon auch zum ersten Mal, verfügen aber zumindest über das erforderliche Fachwissen.

Make: Du hattest einen Stand auf der Maker Faire. Wie kamen deine Roboter bei den Besuchern an?

Düvelmeyer: Auf der Maker Faire habe ich gemerkt, dass diese Themen auch bei Eltern gut ankommen, weil man daran viele andere MINT-Fächer aufhängen und mathematische sowie physikalische Zusammenhänge erklären kann (Kraftübersetzung beim

unendlichen Getriebe, Elektrotechnik, Programmieren, etc.). Die Kids fanden vor allem den Chrome-Dino süß und viele haben ihn im Workshop nachgezeichnet und anschließend ausgedruckt.

Make: Wie geht es mit dem Projekt weiter?

Düvelmeyer: Vom holprigen Anfang mit einem zu komplexen kettengetriebenen Roboter als Lernträger bis hin zum aktuellen Konzept war es ein langer Weg und vieles musste wieder verworfen oder vom Schwierigkeitsgrad her angepasst werden. Mit den in der aktuellen Lernlandkarte aufgeführten Lernträgern haben wir im letzten Schuljahr gute Erfahrungen gemacht, sodass das Grundgerüst unseres Konzeptes nun endlich steht.

Hinter der Lernlandkarte gibt es aber noch einige Arbeitsblätter, Videos und weiteren Content, den man ausarbeiten, aktualisieren oder verbessern kann. Das gilt z.B. für die Fusion-360-Tutorials in den Lernsituationen 2 und 5. Geplant sind langfristig auch noch OnShape-Tutorials.

Für den CAD-Einstieg in Lernsituation 1 ist auch noch ein Einkaufs-Chip geplant, da er noch einfacher zu bauen ist als der Chrome-Dino und den perfekten Einstieg bildet. Diese Idee kam z.B. von einer anderen Schule aus Bochum und mir war sofort klar: Der Dino ist der falsche Einstieg! So wächst dieses Konzept immer weiter an und lebt viel vom Feedback anderer Schulen und der SuS.

—akf

Zum YouTube-Kanal des Projekts geht es über den folgenden Link.



Der Rossmann-Waagen-Hack

Das wollte ich schon lange: Übliche Haushaltsgeräte in mein Home-Assistant-Smarthome einbeziehen. Hier zeige ich Ihnen, wie mein Projekt einer smarten Personen-Waage auf Basis eines Rossmann-Sonderangebots Schritt für Schritt smarter wird.

von Heinz Behling



Mein Smarthome funktioniert eigentlich schon ganz gut: Heizung und Lüftung sowie die Unterhaltungs-Elektronik hat mein Home Assistant voll unter Kontrolle (obwohl ich immer noch glaube, ich wäre die Kontrollinstanz). Doch noch immer gibt es etliche Geräte in meinem Haushalt, bei denen das (noch) nicht der Fall ist.

Und denen möchte ich nun auch ein smartes Bewusstsein einhauchen und Sie daran teilhaben lassen. Mein erstes Opfer soll meine neue Personenwaage sein. Da es sich um ein preiswertes Modell Marke Rossmann handelt und noch nicht einmal 15 Euro gekostet hat, halten sich meine Bedenken wegen eventueller Lebensgefahr für das Gerät in Grenzen. Mir geht es hier auch vor allem darum, Ihnen zu zeigen, was bei so einem Projekt alles zu beachten ist. Mag sein, dass dabei auch mal etwas nicht auf Anhieb so funktioniert, wie es soll. Aber auch ich werde aus Fehlern klug.

Als Ziele für dieses Projekt habe ich:

- Verbindung zwischen Waage und Smarthome-Server per WLAN,
- Akkubetrieb mit langer Laufzeit,
- Einschalten der Waage wie bisher durch Berühren der Wägefläche,
- Speichern der Gewichtsdaten auf dem Home-Assistant-Server nach Personen getrennt,
- Grafische Auswertung der Daten und Möglichkeit, ein Wunschziel zu setzen.

Letztendlich möchte ich die Waage dazu benutzen, die guten Vorsätze, die man an jedem Jahresende fasst ("Im nächsten Jahr nehme ich 10 kg ab!") technisch so gut wie möglich unterstützt in die Tat umzusetzen, ohne am Jahresende feststellen zu müssen: „Ich bin nur noch 11 kg von meinem Ziel entfernt!“ Ich möchte also in meinem Home Assistant ein Wunschgewicht einstellen und bei jedem



Bild 1: Die Waage meiner Wahl

Kurzinfo

- » Elektronik der Personenwaage durch ESP32 ersetzen
- » Lange Akku-Laufzeit durch Deep Sleep und Einschalten per Touch-Kontakt
- » Abgleich der Waage und Datenübernahme in Home Assistant

Checkliste



Zeitaufwand:
3 bis 4 Stunden



Kosten:
55 Euro

Werkzeug

- » Schraubendreher
- » Lötausrüstung
- » Laubsäge, Feile zur Gehäusebearbeitung
- » empfehlenswert: Klein-Bohrmaschine
Dremel o. ä.

Material

- » Glas-Diagnose-Waage von Rossmann oder ähnlich
- » Wemos Lolin32 mini lite
- » 2,2-Zoll-TFT-Display
- » Lipo-Akku 3,7 V, 1200 mAh
- » Schrauben M3 und M2,5 mit Muttern

Mehr zum Thema

- » Kostenlos online: Florian Schäffer, Kirmes-Kraftprotz-Ampel, Make 5/16, S. 64



Alles zum Artikel
im Web unter
make-magazin.de/x7zg



Wiegen erfahre ich, ob und wie viel ich mich meinem Ziel angenähert (oder davon entfernt) habe.

Mit der Verwirklichung dieser Ziele beginne ich in diesem Artikel: Hier ersetze ich die Waage-Elektronik durch ein ESP32-Board mit Farb-Display und Akku. Dazu gibt es eine ESPHome-Firmware, die die Eichung der Waage zulässt und die Daten an meinen Home Assistant übermittelt.

Aber das ist nur der erste Schritt: In weiteren Online- und/oder Heftartikeln werde ich die Waage dann entsprechend der Zielliste weiterentwickeln und Sie daran

teilhaben lassen. Das Projekt ist also noch work in process. Vielleicht können Ihnen aber meine Gedanken weiterhelfen, um eigene Projekte zu entwickeln. Die von mir dabei benutzten Techniken werde ich Ihnen genau erklären. Und falls Sie Ideen zur Waage haben, teilen Sie mir diese mit. Vielleicht kann ich die eine oder andere mit ins Projekt einbauen.

Teileauswahl

Die hier verwendete Waage ist zurzeit bei Rossmann für knapp 15 Euro zu bekommen (Bild 1). Aber auch ähnliche Waagen anderer Anbieter sind geeignet, sogar, wenn sie elektronisch defekt sind. Denn letztendlich brau-

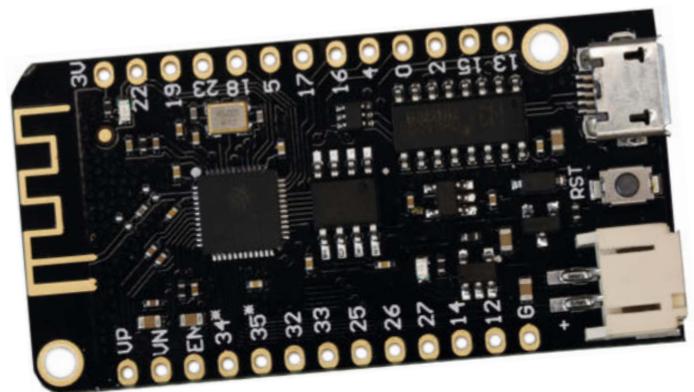


Bild 2: Das Controllerboard Lolin32 mini lite ist klein genug und hat eine Akku-Elektronik.



Bild 3: Hier ist die Verbindung zum GND-Anschluss vorhanden.

chen wir nur die vier Wägezellen, die in den Füßen der Waage stecken. Vorteilhaft sind Modelle, die auch das Körperfett messen können. Dazu dienen zwei Elektroden auf der Wiegefläche, über die der elektrische Widerstand von einem Fuß zum anderen gemessen wird. Diese Elektroden wird meine Waage künftig zum Einschalten verwenden. Damit ich aber auch mit Socken oder gar Schuhen bewaffnet mein Gewicht herausfinden kann, benutze ich auch einen der drei an der Rückseite der Waage serienmäßig angebrachten Taster zum Einschalten. Der mittlere soll das

erledigen. Die Waage wird mittels Akku betrieben.

Damit der nicht nach kurzer Zeit nachgeladen werden muss, schaltet sich die neue Waagen-Elektronik nach 90 Sekunden ab, der ESP32 geht in den Deep-Sleep-Modus. In dieser Betriebsart braucht die Elektronik nur 0,3 mA Strom. Ich wollte, dass die Waage nur etwa alle drei Monate nachgeladen werden muss. Daher entschied ich mich für eine Akkuzelle mit 1200 mAh. Das reicht für 4000 Stunden Standby-Betrieb, also mehr als 160 Tage. Der kurzzeitige Stromverbrauch beim Wiegen (ca. 120 mA) kann vernachlässigt werden.

Für den Akkubetrieb ist auch eine entsprechende Ladeelektronik notwendig. Um nicht zu viele Bauteile im knapp bemessenen Gehäuse unterbringen zu müssen, kam nur ein Controllerboard mit Akku-Anschluss infrage. Außerdem sollte es ein ESP32 sein, damit ich genügend I/O-Pins für den Wiege-Sensor, Schalter und Display zur Verfügung habe. Meine Entscheidung fiel daher auf ein Lolin32-mini-lite-Board (Bild 2).

Das Display sollte gut lesbar sein, wenn man auf der Waage steht. Farbe wäre zwar kein Muss, aber doch schön, denn so kann ich später eine Gewichtszu- oder abnahme mit roter oder grüner Schrift deutlich anzeigen. Die Entscheidung fiel daher auf ein 2,2-Zoll-TFT-Display mit 320 x 240 Pixeln, das recht gut in den Raum des alten Displays passt.

Die Wägezellen sind ja Bestandteil der Waage, aber ein Messverstärker ist noch notwendig, um die Daten fürs Controllerboard aufzubereiten. Da kommt nur ein HX711 infrage, und zwar in möglichst kleiner Bauform (Bezugsquellen siehe Kurzinfo-Link). Mein Modul hatte 16 x 24 mm Größe. Wichtig: Für einwandfreie Messungen müssen auf dem Modul die Kontakte GND und E- miteinander verbunden sein. Das kann man entweder an einer entsprechenden Leiterbahn auf der Platineunterseite erkennen (Bild 3) oder mittels Ohmmeter nachmessen. Fehlt diese Verbindung (solche Module sind tatsächlich in großer Zahl im Handel), hilft eine eingelötete Drahtbrücke.

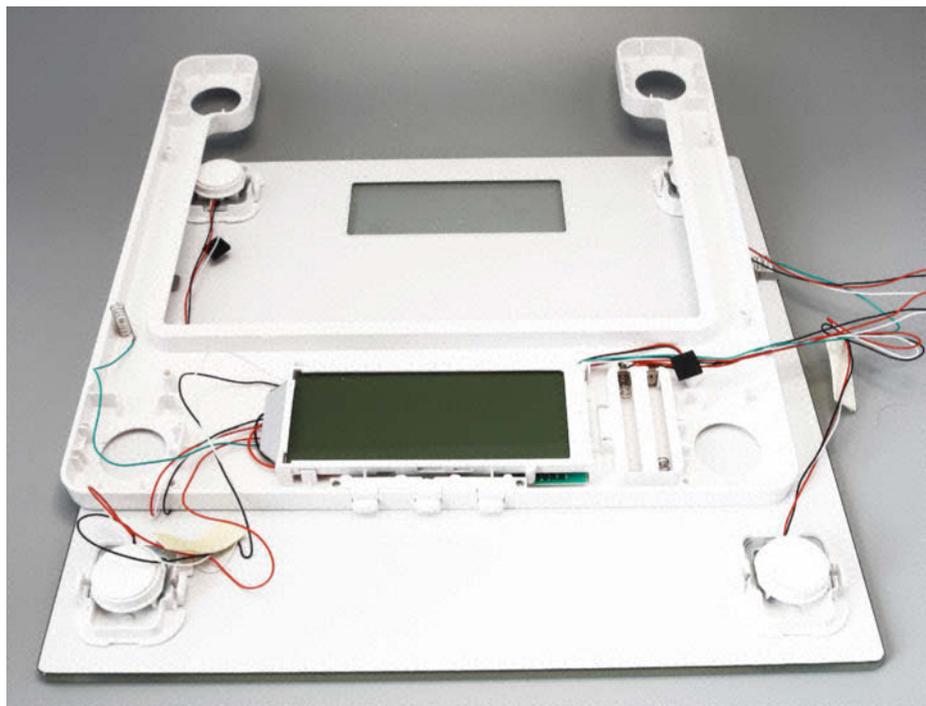


Bild 4: Das geöffnete Gehäuse sollte behutsam behandelt werden, damit kein Draht abreißt.

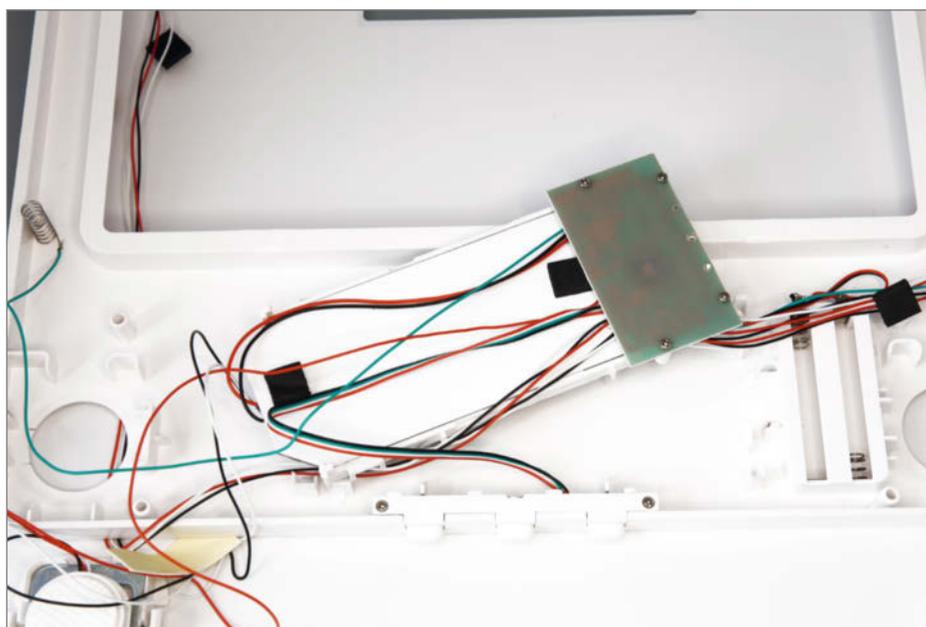


Bild 5: Die Waagen-Elektronik sitzt auf dieser kleinen Platine.

Waage öffnen

Damit schreite ich zur Tat. Falls man keine zweite Waage zur Verfügung hat: Batterien einlegen, Waage auf den Boden stellen und das eigene Gewicht ermitteln und notieren. Das dient nicht nur des Schaffens eines schlechten Gewissens, sondern der Ermittlung eines wichtigen Eichwertes. Am Ende sollte mir die Waage mit der neuen Elektronik ja wieder genau dieses Gewicht anzeigen.

Erst dann musste ich an die Waagen-Elektronik herankommen: Sie sitzt unterhalb des Displays. Wenn man die Waage umdreht, sieht man das Plastikgehäuse. Es ist lediglich im

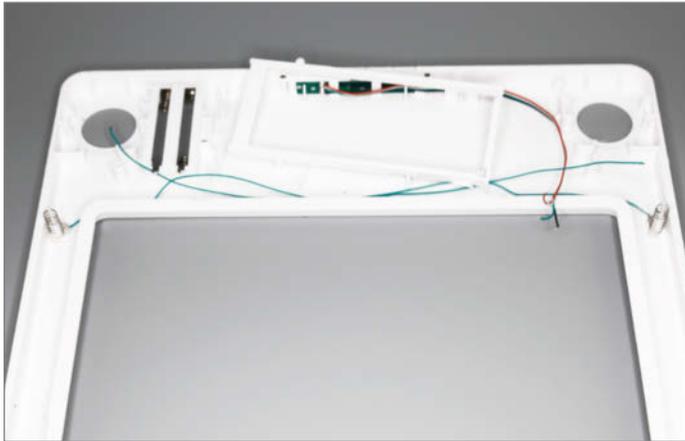


Bild 6: Vom Unterteil werden nur das Gehäuse selbst, die beiden Kontaktfedern zu den Fußelektroden, die Tasterleiste an der hinteren Kante und der kleine Displayrahmen gebraucht.

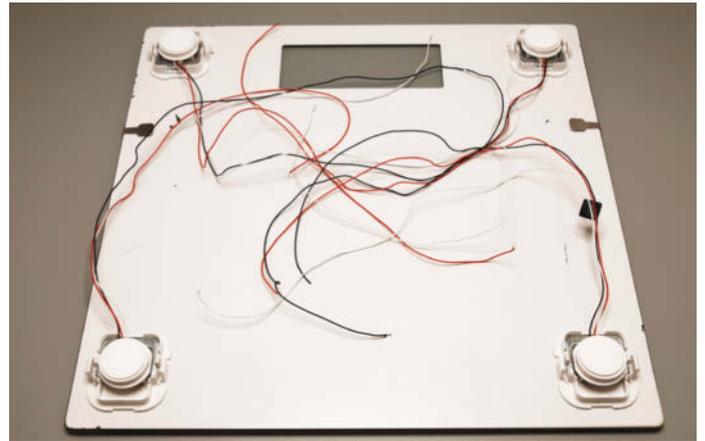


Bild 7: Beim Oberteil mit den vier Wägezellen entwirrte ich zuerst die zwölf Drähte.

Oberteil (einige an der Glasfläche angeklebte Plastikteile) eingeclipst. Mit einem kleinen Schraubendreher konnte ich, an den Ecken beginnend, die eingerasteten Haken vorsichtig lösen. Vorsichtig, weil man dabei zum einen das Kunststoffgehäuse zerbrechen und zum anderen die Lackierung an der Unterseite der Glasfläche beschädigen kann. Das ist mir auch

glatt passiert, was man auf dem ein oder anderen Foto in diesem Artikel auch erkennen kann. Wenn es mich irgendwann zu sehr stören sollte, werde ich das mit einem geeigneten Lackstift wieder ausbessern.

Die Elektronik sitzt unter dem Display im Gehäuseunterteil und ist über zwölf Drähte mit den Wägezellen verbunden, die am Ge-

häuseoberteil befestigt sind (Bild 4). Vorsicht! Diese Drähte sind sehr dünn und dürfen auf keinen Fall an den Wägezellen abreißen, da man sie dort nicht wieder anlöten kann, ohne die Messeigenschaften der Zelle zu verändern.

Das Display ist mittels eines Kunststoffrahmens im Unterteil eingerastet. Der lässt sich

Online-Shopping ohne Probleme: c't hilft.



Heft + PDF mit 29 % Rabatt

Ohne Reue günstig digital einkaufen und zahlen – aber sicher muss es sein. Das c't-Sonderheft gibt Rat, welches Zahlungsmittel Sie wählen sollten, um Ihr Geld zurückzubekommen und Cyberkriminellen nicht auf den Leim zu gehen.

- ▶ Die wichtigsten Regeln für den Onlinekauf
- ▶ Schützen Sie sich vor Betrug
- ▶ Kaufprobleme lösen
- ▶ Käuferschutz richtig einsetzen
- ▶ Digital bezahlen
- ▶ **Auch als Heft + digitale Ausgabe mit 29 % Rabatt**

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

 shop.heise.de/ct-sicher-einkaufen23

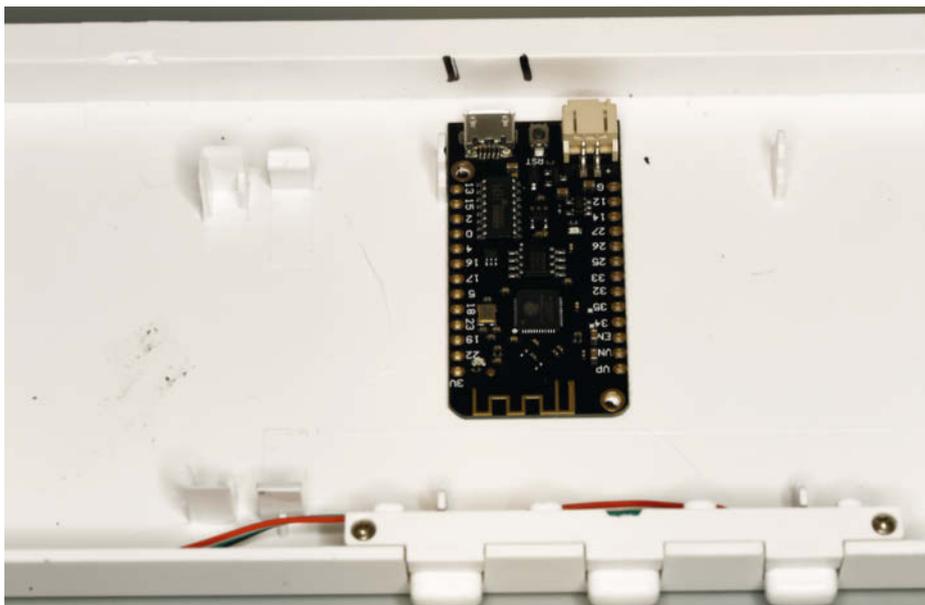


Bild 8: Achten Sie auf den Platz für den Akku-Stecker.

recht leicht lösen. An der Unterseite des Displays ist die Elektronik mittels fünf kleiner Schrauben befestigt (Bild 5). Nachdem ich die gelöst hatte, lötete ich die zwölf Drähte der Wägezellen von der Platine ab. Auch die Drähte am Batteriefach waren nun überflüssig, denn ein Akku sollte ja später den Strom liefern. Die beiden Kontaktfedern zu den Fußelektroden des Oberteils brauchte ich ebenfalls noch. Da die Federn ständig aus dem Gehäuse fielen, habe ich sie mit etwas Heißkleber befestigt.

Schließlich hatte ich das von der Elektronik befreite Unterteil (Bild 6) und das Oberteil bestehend aus Glasplatte und vier Wägezellen zur Verfügung (Bild 7).

Gehäuse anpassen

Controller und Display sollten dort mit Schrauben befestigt werden, wo auch das alte Display saß. Außerdem musste die USB-Buchse des Boards von außen zugänglich sein, damit man den Akku nachladen kann. Ich legte das Board ins Gehäuse und zeichnete die Lage der USB-Buchse und der Schraubenlöcher an. Achtung: Die USB-Buchse darf nicht direkt an der Gehäusewand anliegen. Daneben befindet sich ja der Akkuanschluss, in den im Gehäuseinneren später noch der entsprechende Stecker passen soll. Daher setzte ich das Board etwa 5 mm tiefer ins Gehäuse hinein (Bild 8). Das Board wird später mit zwei

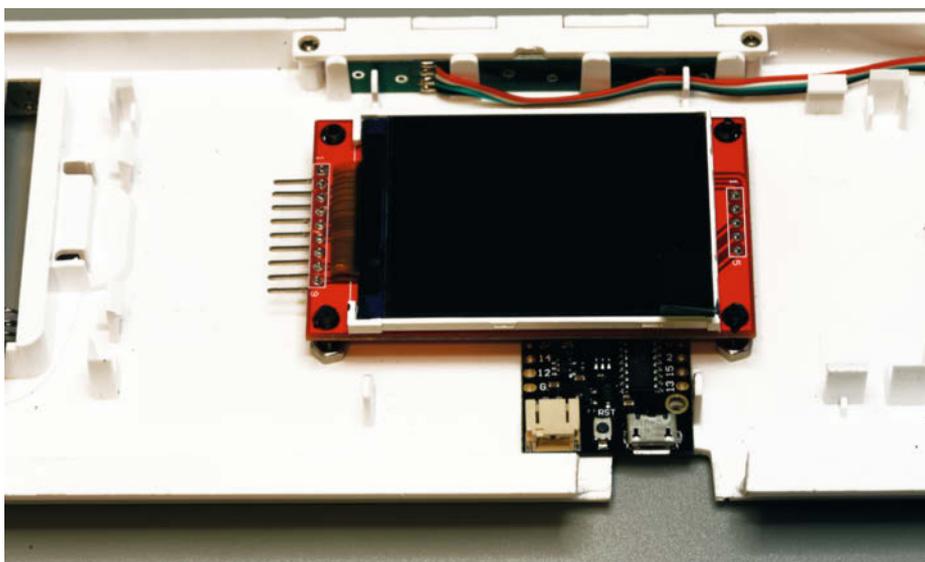


Bild 9: Bei manchen Displays weisen die Anschlussstifte nach unten. Die müssen zur Seite gebogen werden, da das Display sonst zu hoch ist.

M2,5×6-Schrauben und passenden Muttern befestigt.

Mit einer Mini-Bohrmaschine und kleiner Trennscheibe schnitt ich dann den Platz für die USB-Buchse aus. Rechts und links ließ ich die Öffnung jeweils 5 mm breiter, damit dort noch ein USB-Stecker hindurchpasst. Die Bohrlöcher für die Schrauben haben 2,5 mm Durchmesser.

Danach nahm ich für das Display Maß. Es sollte mittig im Displayrahmen sitzen. Die Bohrlöcher haben 3 mm Durchmesser. Die Befestigung erfolgt mit vier M3×12-Schrauben. Die werden zunächst mit Muttern im Gehäuse befestigt. Darauf kommt dann noch eine Mutter, mit der man die Einbauhöhe des Displays einstellt. Es sollte das Board gerade nicht berühren. Schließlich folgt das Display, das dann am Ende mit Muttern gesichert wird (Bild 9). Zunächst nahm ich nur Maß, der endgültige Einbau sollte erst nach der Verdrahtung erfolgen.

Löten

Jetzt kam eine Fleißarbeit: Die Schaltung musste entsprechend dem Schaltplan, der auch zum Download bereitsteht (Bild 10), verbunden werden. Die Drähte dürfen dabei nicht durch die Lötösen des Controllerboards hindurchgesteckt und von unten angelötet werden, da dann das Board zu hoch wird. Stattdessen werden alle Drähte oben auf die Lötösen gelötet. Dabei kann das Board schon im Gehäuse stecken. So kann man auch gut die Drahtlängen zum Display abschätzen. Sie sollten so kurz wie möglich sein, es geht eng im Gehäuse zu.

Störrische Kabel kann man mit einem Tropfen Heißkleber bändigen. Auch den HX711 hab ich damit im Gehäuse befestigt. Zugegeben, es wird etwas eng im Gehäuse, aber es lässt sich alles unterbringen (Bild 11).

Schließlich setzte ich noch das Display auf und schraubte es fest. Dann galt es, beide Gehäuseteile wieder so zu vereinen, dass die Kabel weder eingeklemmt wurden noch herausragten oder vor dem Display entlangliefen. Dabei war eine feine Pinzette sehr hilfreich und schließlich gelang es.

Firmware aufspielen

Die Firmware musste ich natürlich neu anlegen. Falls Sie die Waage nachbauen, finden Sie die Download-Adresse des Quelltexts über den Kurzinfo-Link. Die ersten Schritte sind jedoch gleich, egal ob neu angelegt oder mit von uns kopiertem Quelltext: Ich begann damit, dass ich die Waage per USB-Kabel mit dem Home-Assistent-Server verband und im Browser des Computers die Home-Assistent-Web-Oberfläche öffnete. In der seitlichen Leiste folgte ein Klick auf „ESPHome“. Schließlich ging es mit „+ New Device“ und „Continue“ weiter. Dann brauchte das neue Projekt einen

Namen. Ich entschied mich für „waage1“ (man weiß ja nie, wie viele da noch kommen). Danach wählte ein Klick auf ESP32 den Controller aus. In früheren ESPHome-Versionen konnte man hier direkt die Board-Bezeichnung wählen. Das geht nun leider nicht mehr (Grrr...), sondern muss im nächsten Schritt nachgeholt werden. Deshalb folgte im nächsten Fenster zunächst ein Klick auf „Skip“. Home Assistant konnte nun bereits das neue Projekt (Bild 12).

Mit „Edit“ ging es weiter. Im nächsten Fenster ersetzte ich die Board-Bezeichnung `esp32dev` durch `lo1in32_lite`. In die Zeile hinter `captive_portal`: (unbedingt darauf achten, dass der Cursor exakt am Anfang der neuen Zeile steht!) kam der neuen Quellcode (bitte dort per Kopieren und Einfügen aus der Download-Datei einsetzen). Dann ließ ich per „Install“ die Firmware auf die Waage überspielen. (Eventuell werden Sie noch nach dem richtigen Port gefragt, falls mehrere USB-Geräte am Home-Assistant-Server angeschlossen sind). Das Kompilieren und Übertragen dauerte ein paar Minuten. Dann meldete sich die Waage zum ersten Mal im Display (Bild 13).

Das Gewicht auf dem Display konnte zu diesem Zeitpunkt noch nicht stimmen. Es fehlte ja noch der Abgleich für die Wägezellen und auch für die Touch-Sensoren. Doch zuvor kam erstmal eine Pause, denn zunächst ließ ich den Akku drei Stunden aufladen. Das erledigte ich mit einem alten Handy-Netzteil. Grund dafür: Touch-Sensoren reagieren bei Boards mit Akku-Anschluss unterschiedlich, je nach Spannungsquelle. Da die Waage ja später nur per Akku betrieben werden sollte, musste ich den erstmal sicher einsatzbereit machen.

Da ESPHome mit OTA (Over The Air)-Update arbeitet (also Updates via Funknetz), war auch für die weiteren Kalibrierungsarbeiten an der Firmware keine USB-Verbindung zum Server mehr nötig.

Kalibrierung

Falls eine zweite Waage zur Verfügung steht, geht es nach sieben Kaffee und einem Schnitzel weiter. Andernfalls darf man bis zur Kalibrierung des Mess-

wertes keine Nahrung aufnehmen oder sein Körpergewicht auf andere Art ändern. Ich empfehle daher dringend die Verwendung einer zweiten, notfalls ausgeliehenen Waage!

Schließlich ging es zunächst mit der Touch-Kalibrierung weiter. Im ESPHome-Bereich auf der HA-Web-Oberfläche ließ ich mir nach einem Klick auf Logs im waage1-Fenster die von dort gesendeten Daten anzeigen (Bild 13).

Der Wert fürs Touchpad schwankte um 435 herum (das kann bei Ihrem Board anders sein). Den notierte ich als ersten. Dann berührte ich beide Elektroden auf der Waage mit bloßen Händen (barfüßiges Daraufstellen ist nicht unbedingt notwendig, aber etwas genauer) und notierte auch den dann angezeigten Wert.

Auch der schwankte etwas, lag aber immer unter 10. Das hieß nun zunächst mal, dass der Sensor funktioniert, und zweitens hatte ich die Möglichkeit, daraus den Kalibrierungswert für den Touch-Sensor zu berechnen. Der sollte in der Mitte zwischen beiden Werten liegen, also in meinem Fall bei 223. Diesen Wert setzte ich dann in die Firmware (zuvor im LOG-Fenster auf edit klicken) bei `threshold` und `wakeup_threshold` ein (Bild 15).

In der Download-Firmware ist übrigens die Deep-Sleep-Funktion noch nicht enthalten, da sie beim Überspielen und der Eichung der Firmware stören würde. Jetzt ist es an der Zeit, sie zu aktivieren (siehe Kasten). Das geschieht am Anfang der Firmware unter dem Block

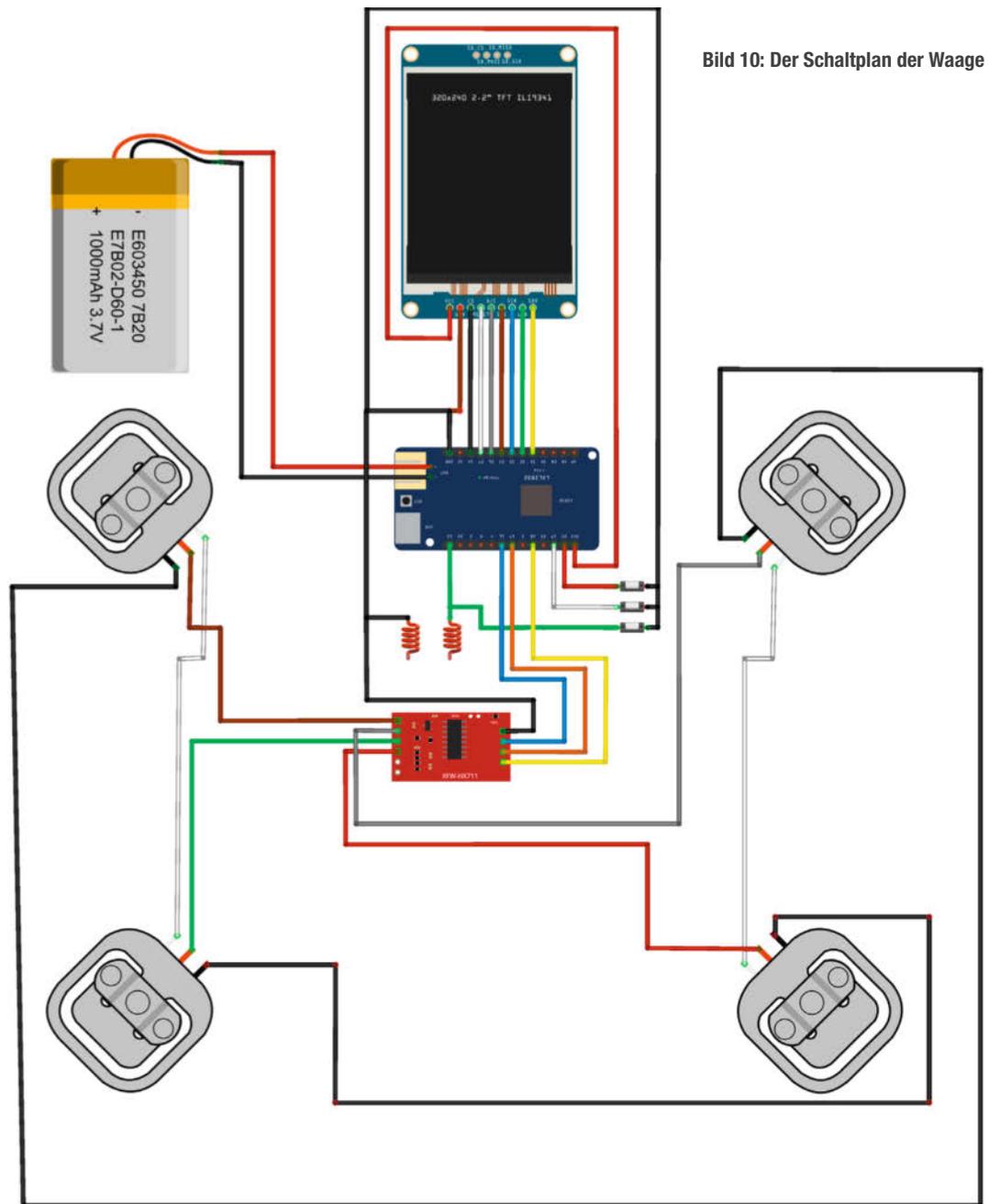


Bild 10: Der Schaltplan der Waage

Bild 11: Machen Sie die Kabel nicht länger als unbedingt notwendig.

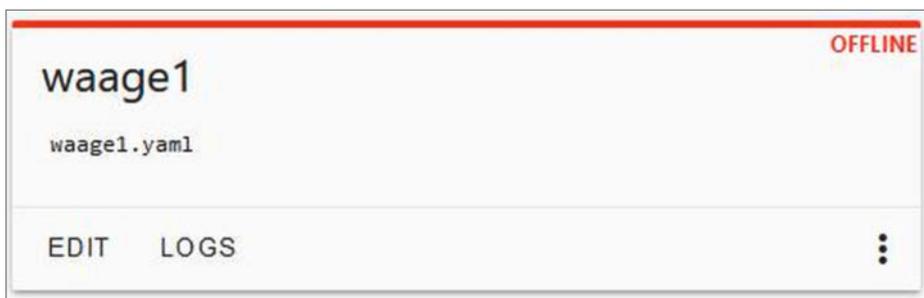


Bild 12: Das Projekt ist angelegt, enthält aber noch keinen Programmcode.



Bild 13: Sie lebt, auch wenn das angezeigte Gewicht noch nicht stimmt.



Bild 14: In den LOG-Daten interessiert der Wert fürs Touch Pad und der HX711 Value.

esp32. Fügen Sie dort die drei Zeilen des Blocks deep_sleep ein, wie im Kasten zu sehen.

Mit „Install“ wurde die Firmware erneut übertragen. Der Deep Sleep und das Aufwachen konnten jetzt überprüft werden. Nach etwa 1,5 Minuten sollte sich das Display abschalten und nach Berühren beider Elektroden wieder aufleuchten. Falls nicht, stimmt der Kalibrierungswert nicht. Also wiederholen. Aber im Allgemeinen ist der Wert recht unkritisch.

Nun ging es an die Gewichtseinstellung: Wieder betrachtete ich die LOG-Daten, diesmal interessierte mich der HX711 Value. Auch dieser Wert schwankt etwas. Hier gilt es, einen mittleren Wert zu finden (Bild 16).

Diesen Wert (197800) setzte ich in die Firmware ein.

```
- calibrate_linear:
  - 197800 -> 0
```

Nach erneutem Installieren der Firmware stimmte der Nullwert. Dann wurde die Prozedur wiederholt. Allerdings stand ich diesmal auf der Waage. Erneut notierte ich den Wert. Der wurde diesmal aber in die Zeile unter dem Nullwert statt der -4000 eingetragen:

```
- -4000 -> 1
```

Die 1 wurde durch mein zuvor ermitteltes Gewicht (bzw. das einer Vergleichswaage) ersetzt. Bei Nachkommastellen bitte einen Punkt statt eines Kommas verwenden. Aus Datenschutzgründen wird das hier aber nicht mitgeteilt. Nochmals wurde die Firmware installiert und dann zeigte die Waage mein Gewicht richtig an.

Die Daten der Waage konnte ich nun auch in Home Assistant verwenden und beispielsweise auf dessen Web-Oberfläche anzeigen. Dazu geht man dort in den Bereich, in den man die Anzeige anordnen möchte (in meinem Fall Bad). Dort klickt man auf die drei Punkte oben rechts und wählt „Dashboard bearbeiten“. Nach „+ Karte hinzufügen“ wählt man „Pegel“ und gibt als Entität „waage1HX711 Value“ ein. Nach Speichern/Fertig erscheint dann die Skala der Waage zusammen mit dem gerade gemessenen Gewicht (Bild 17). Auch diesen Wert habe ich aus Datenschutzgründen etwas verändert.

Damit endet die erste Folge des Waagen-Umbaus. Im Moment kann die Waage ja nicht mehr als sie ursprünglich schon beherrschte. Einziger Unterschied: Die gemessenen Gewichte werden jetzt ans Smarthome übertragen. In den nächsten Folgen zeige ich dann, wie ich die Daten dort auswerte, einzelnen Benutzern zuordne, mit individuellen Zielvorgaben vergleiche und auch grafisch darstelle (schön, wenn die Kurve runtergeht ...). —hgb

```
[15:25:19][D][esp32_touch:298]: Touch Pad 'ESP32 Touch Pad GPIO27' (T4): 434
[15:25:19][D][hx711:031]: 'HX711 Value': Got value 197828
[15:25:19][D][esp32_touch:298]: Touch Pad 'ESP32 Touch Pad GPIO27' (T4): 432
[15:25:19][D][esp32_touch:298]: Touch Pad 'ESP32 Touch Pad GPIO27' (T4): 432
[15:25:19][D][hx711:031]: 'HX711 Value': Got value 197800
[15:25:19][D][sensor:094]: 'HX711 Value': Sending state 0.42428 kg with 1 decimals of accuracy
```

Bild 16: Für die Gewichtskalibrierung ist der Wert hinter Got value wichtig.

```
63 binary_sensor:
64   - platform: esp32_touch
65     name: "ESP32 Touch Pad GPIO27"
66     pin: 13
67     threshold: 223
68     wakeup_threshold: 223
```

Bild 15: Hier wird die Empfindlichkeit des Touch-Sensors festgelegt.

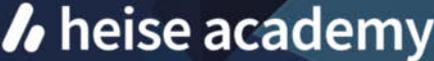


Bild 17: Der Wert auf der Home-Assistant-Oberfläche: Nein, so ein Leichtgewicht bin ich nicht!

Deep-Sleep-Code

```
esp32:
  board: lolin32_lite
  framework:
    type: arduino

deep_sleep:
  run_duration: 90s
  touch_wakeup: true
```



Maschinelles Lernen leicht gemacht!

Webinar-Serie „Einführung in Machine Learning mit Python“

11. Januar – 15. Februar 2024

KI und Machine Learning sind aus der heutigen IT-Welt nicht mehr wegzudenken. Wer den Einsatz von maschinellem Lernen für eigene Projekte prüfen und umsetzen möchte, erhält einen Überblick über die Grundlagen und Grenzen der künstlichen Intelligenz.

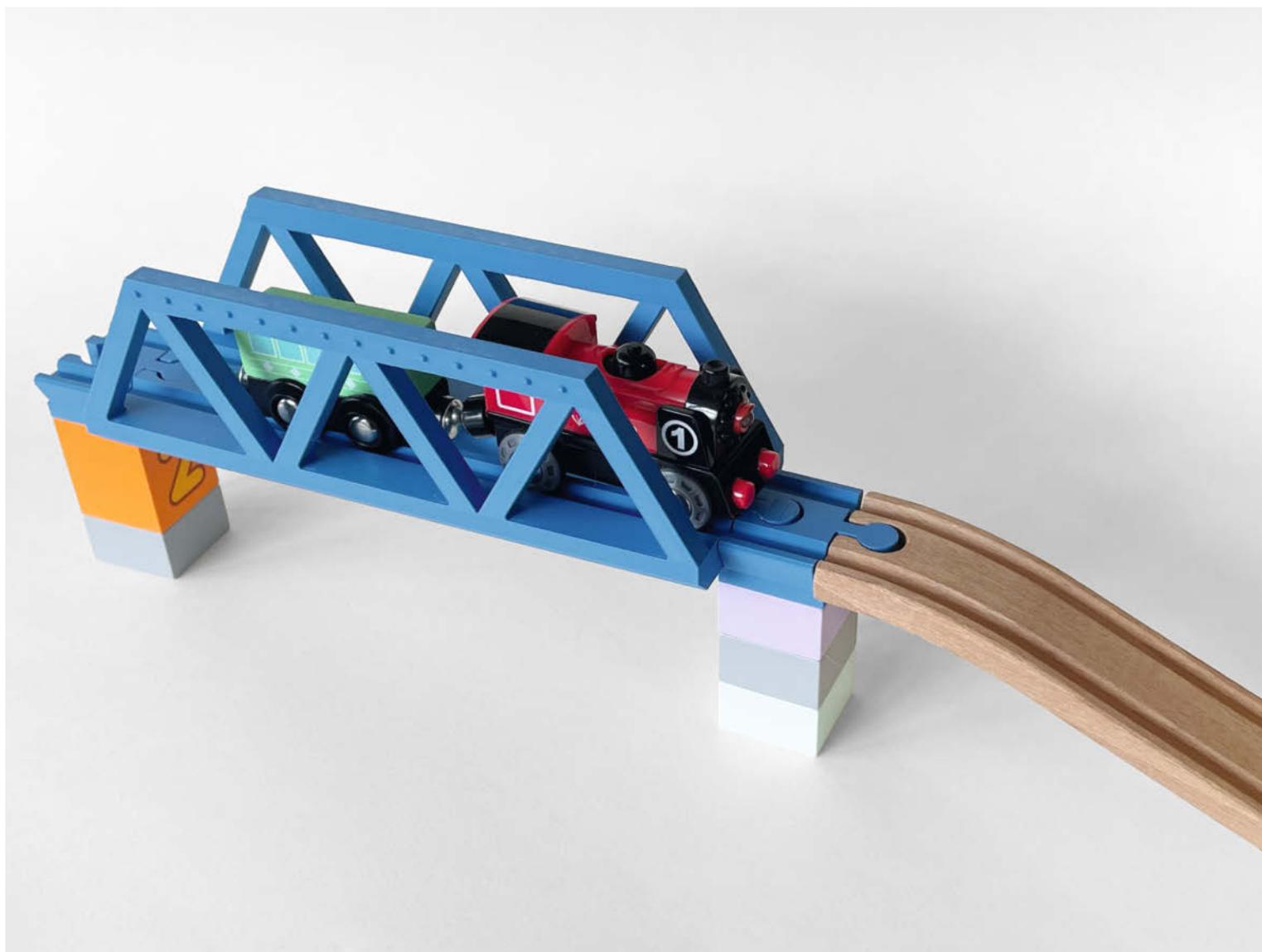
Eigene Programmierübungen in Python werden mit Theorie, Demonstrationen, und Hands-on-Experimenten abwechslungsreich unterstützt.



Holzeisenbahn mit 3D-Druck erweitern

Der gebrauchte gekaufte Holzeisenbahn lagen drei Brückenrampen bei. DREI!? Nichts hätte unseren Nachwuchs mehr auf die Palme bringen können! Ein kurzer Blick in den 3D-Druck-Katalog Thingiverse hat aber Abhilfe geschaffen: Neben Brückenrampen unterschiedlichster Bauart findet man auch Adapter, die die Holzeisenbahnwelt zu anderen Spielzeug-Ökosystemen öffnen.

von Johannes Börnsen



Spielzeughersteller haben in der Regel wenig Interesse, ihre Produkte kompatibel zu anderen Herstellern zu machen. Schließlich soll man Zubehör möglichst nur aus dem eigenen Hause kaufen. Dabei fördern zusätzliche Kombinationsmöglichkeiten die Kreativität bei groß und klein. Dank der 3D-Druck-Community gibt es aber Abhilfe: Etliche kostenlose Modelle erweitern die Brio-Holz-eisenbahn in Richtung Duplo-Klemmbausteinwelt oder bieten Zubehör, das die Hersteller selber nicht im Programm haben. Und wer dort nicht fündig wird, kann die Modelle anpassen oder gleich eigene entwickeln. Dieser Artikel stellt praktische Modelle vor und gibt Tipps zum Anpassen und Selberzeichnen. Links zu den erwähnten Modellen finden Sie in der Kurz-URL am Ende des Artikels.

Am naheliegendsten sind offenbar Brückenpfeiler aus Duplo. Drei 2x2-Steine übereinander (also mit vier Noppen im Quadrat) und ein Brio-Adapter oben drauf ergeben genau die klassische Holz-eisenbahn-Brückenhöhe. Es lassen sich aber auch Zwischenhöhen realisieren. Als Rampe dienen dann gerade Schienen, das Eisenbahn-Stecksystem bietet dafür genug Spiel.

Solche Brückenpfeileradapter gibt es in vielen Ausführungen, gute Erfahrungen habe ich mit einem Modell vom User „Molgan“ auf Thingiverse gemacht. Es ist leicht zu drucken, braucht wenig Support, passt problemlos zu unserem Schienen-Sammelsurium und klemmt wie ein Original-Stein auf den Duplo-Quadern. Neben Duplo-Brücken-Adaptoren gibt es auch solche zu Trackmaster-Schienen oder Zugmaschinen, in denen Playmobil-Figuren Platz nehmen können.

Ein echter Hit sind flexible Schienen. Sie zerteilen den Grundkörper einer Geraden mit einem Zickzack-Muster in ein Gleis, das sich sowohl seitlich, als auch in der Höhe biegen lässt. Das nimmt Druck aus komplexen Gleisanlagen, bietet aber auch die Möglichkeit, beispielsweise Türschwellen oder Teppichkanten zu überbrücken. Das Modell ist ohne Support druckbar, benötigt aber aufgrund der komplexen Form recht lange. Praktisch sind auch Adapter, die zwei offene Schienenenden miteinander verbinden, sowohl männlich/männlich als auch weiblich/weiblich.

Ein weiteres häufiges Modell sind kurze Kurvenstücke mit Winkeln von mehr als 20 Grad auf wenigen Zentimetern. Den Druck dieser Schienen können Sie sich jedoch sparen.

Eine einzelne Lokomotive ohne Anhänger



Wer kleine Physikexperimente zum Entgleisen von Zügen machen möchte, ist mit dem Adapter ganz rechts gut bedient: Der Radius ist zu eng gewählt. Der linke und mittlere Adapter funktionieren dagegen, sind aber – Achtung Wortspiel – weniger flexibel als die biegsamen Gleise.

ger kommt durch die Knicke zwar durch, je mehr Anhänger jedoch angekoppelt sind, desto eher wird der Zug an solch engen Bögen entgleisen, weil die seitlichen Zugkräfte zu groß werden. Dieser sogenannte Entgleisungskoeffizient lässt sich im Übrigen mit der Gleichung nach Nadal berechnen, das ist aber eher etwas für erwachsene Kinder.

Neben solch einfachen Teilen finden sich auch komplexe Modelle wie ganze Bahnhöfe, Verladekräne oder umschaltbare Gleise. Für etwas ältere Kinder sind außerdem zu den Holzgleisen passende Fahrgestelle mit Lego-Noppenoberseite eine Möglichkeit, eigene Lokomotiven und Anhänger zu bauen.

Tipps für den 3D-Druck

Für den Druck der meisten Teile reicht PLA aus. Etwas robuster wird es jedoch mit PETG, was den rauen Kinderzimmeralltag besser wegsteckt als das empfindlichere PLA. Das kommt vor allem flexiblen Teilen wie den biegsamen Geraden zu Gute. Wichtig ist außerdem, auf eine gute Layerhaftung zu achten. Diese können sie mit dem Druck eines Temptowers ausprobieren. Dabei wird die Düsentemperatur innerhalb eines Modells in mehreren Schritten erhöht und Sie können nachher ausprobieren, bei welcher Temperatur die Schichten des Druckes am besten aneinander haften.

Mit den systemübergreifenden Verbindern ist bei uns übrigens eine wichtige Frage aufgetaucht: Gehören die Brückenpfeiler-Adapter beim Aufräumen in die Holz-eisenbahn- oder die Klemmbausteinkiste? —jom

Alles zum Artikel im Web unter make-magazin.de/xwe7

Durch das Zickzack-Muster ist dieses Gleis flexibel, überbrückt Ungenauigkeiten in komplexen Gleisbauten und gleicht auch den einen oder anderen Gleisschubser aus.



KI-Sprachassistent mit eigenem sprechenden Avatar

Ein 10-Zoll-Display, ein Raspberry Pi und ein Mikrofon werden durch etwas Python-Code und viele Webdienste zu einem personalisierten Sprachassistenten, der zu allen Fragen eine (mehr oder weniger richtige) Antwort hat. Die Aufgaben und Fähigkeiten des eigenen Assistenten können je nach Bedarf im Python-Code auch noch erweitert werden.

von Dirk Wahl



Sprachassistenten wie Alexa, Siri, Cortana, etc. gibt es schon in vielen Varianten. Der hier vorgestellte selbstgebaute Sprachassistent kommt in Gegensatz dazu mit einer persönlichen Note daher, denn ein individueller Avatar einer beliebigen Person (in meinem Fall von mir selbst) auf dem Bildschirm gibt Antworten auf die gestellten Fragen.

Dank seiner Verbindung mit Google Speech-to-Text und Text-to-Speech und die API-Schnittstelle zu ChatGPT kann dieser Sprachassistent zudem auch noch andere Dinge als viele seiner eingangs erwähnten digitalen Kollegen, etwa kreativ Rollen spielen oder als Übersetzer agieren, eben alles, was ChatGPT zu bieten hat. Aktuelle Nachrichten wie das Wetter oder die Verkehrslage kann ChatGPT allerdings nicht liefern – ist das gewünscht, kann man sich den offenen Python-Code des Assistenten selbst nach Belieben erweitern und Informationen über andere APIs einholen (ähnlich wie bei der Staumeldung durch KI-Bilder in Make 4/23, siehe „Mehr zum Thema“ in der Kurzinfor).

Ich betreibe meinen Sprachassistenten auf einem Mini-HDMI-Monitor mit Lautsprecher; der Raspberry sitzt hinten auf dem Bildschirm und ein USB-Mikro mit Schwanenhals ragt nach oben heraus. Betrieben wird alles zusammen durch ein 12-Volt-Netzteil. Alternativ ist auch ein Akkubetrieb möglich, unterwegs muss dann das mitgeführte Smartphone per Hotspot für die nötige Internetverbindung sorgen. Im Prinzip kann man aber einen eigenen Assistenten nach diesem Muster auch mit größeren oder kleineren Bildschirmen bauen – ich habe inzwischen noch eine mobile Version mit einem 3,5-Zoll-Touch-Display gebaut. Der lässt sich mit einer Powerbank und Bluetooth-Lautsprecher oder Kopfhörer betreiben; das Mikro steckt im USB Port (Bild siehe am Ende des Artikels).

Eigenen Avatar erschaffen

Beginnen wir mit einem Bild des personalisierten Avatars, der als Personifikation des Assis-

Kurzinfo

- » Individuellen sprechenden Avatar erstellen
- » Wake Word generieren und Anfrage über das Mikrofon aufnehmen
- » API-Verbindungen zu Google und ChatGPT herstellen
- » Ausgabe der ChatGPT-Antwort als Animation mit Sprache auf dem Mini-Monitor

Checkliste



Zeitaufwand:
etwa zwei Stunden



Kosten:
ca. 150 Euro für die Hardware, für die Nutzung der Webdienste fallen ggf. laufende Kosten an

Mehr zum Thema

- » Dirk Wahl, KI auf der Schreibmaschine, Make 2/23, S. 28
- » Dirk Wahl, Staumeldung durch KI-Bilder, Make 4/23, S. 76
- » Video: Der Sprachassistent in Aktion



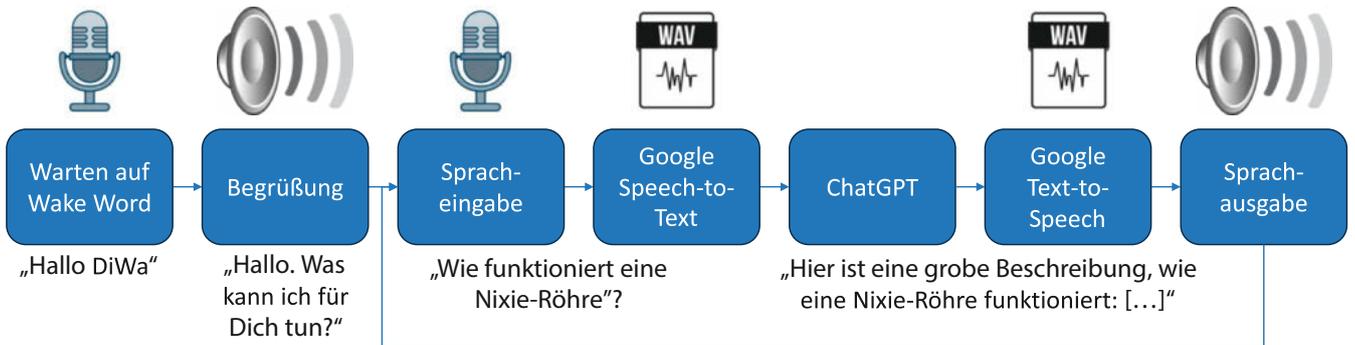
Material

- » Raspberry Pi 4 mit Gehäuse
- » MicroSD-Karte 32 GB
- » 10,1-Zoll-Mini-Monitor mit HDMI-Eingang und Lautsprecher; meiner ist von Aishichen
- » HDMI-Kabel mit Microstecker für den Raspberry Pi 4
- » 12-V-Netzteil und/oder 11,1V-LiPo Akku samt Anschlusskabel mit XT60-Buchse
- » 12-V-DC-DC Step-down-Converter mit USB-C Ausgang für den Raspberry Pi
- » DC-Y-Kabel zum parallelen Verbinden des Monitors und des Step-Down Converters
- » USB-Mikrofon mit Schwanenhals; meines ist von Supvox

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

tenten auf dem Bildschirm erscheinen soll. Ich habe dafür ein Foto von mir in einen Comic-Stil umsetzen lassen. Für so etwas gibt es aktuell eine Vielzahl von Anbietern – ich habe mit der Online-Bildbearbeitung [fotor.com](https://www.fotor.com) ganz gute Ergebnisse erhalten. Mit den 8 Credits, die man zur Anmeldung derzeit gratis erhält, können

ein paar Varianten der Umsetzung des eigenen Porträtbilds in verschiedenen Styles erstellt werden. Diesen und auch die meisten nächsten Schritte habe ich auf dem PC durchgeführt, und die fertigen Dateien dann später per VNC auf den Raspberry kopiert, auf dem der Assistent läuft – dazu später mehr.



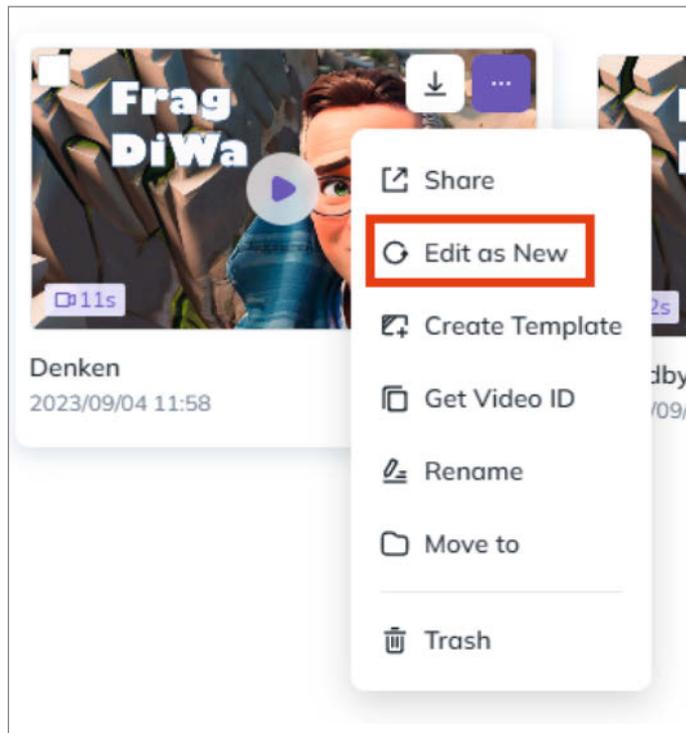
Für den Bau des Sprachassistenten werden einige Tools und API-Services benötigt, die in diesem Artikel Schritt für Schritt erklärt werden – hier die grafische Übersicht.



So sieht meine Verwandlung in fotor.com mit dem Style „Disney 2“ aus.



Das Uhr-Icon steht bei HeyGen bei der Texteingabe für eine Pause von einer halben Sekunde.



Bei HeyGen erzeugt man ein Video mit neuem Text auf Basis des vorangegangenen bequem über das Kontextmenü.

Kurz zusammengefasst war der Ablauf so: Bei fotor.com kostenlos registrieren, Angebot zum Testen der Pro-Version wegklicken, im Menü „Photo Editing Tools/Photo to Cartoon auswählen“ Bild hochladen, Style wählen und links unten auf „1 image“ umstellen, damit pro Durchgang nur ein Bild erzeugt wird.

In PowerPoint habe ich dann noch auf dem heruntergeladenen Bild den Schriftzug „Frag DiWa“ ergänzt und das ganze als .jpg abgespeichert.

Der so erstellte Avatar ist noch ein statisches Bild und muss nun noch sprechen lernen: Dazu habe ich heygen.com verwendet. Für den Sprachassistenten werden vier Versionen von Animationen benötigt. Die zwei freien Credits bei neuer Registrierung entsprechen zwei Minuten Video (es wird jeweils auf

halbe Minuten aufgerundet) und reichen dafür genau aus:

1. Stumm, für das Warten auf das Wake Word oder auf die Antwort von ChatGPT
2. Begrüßung: „Hallo. Was kann ich für Dich tun?“
3. Antwort von ChatGPT, dazu gleich mehr
4. Verabschiedung: „Ich wünsche Dir einen schönen Tag.“

Fangen wir mit 2. an: Nach der Registrierung bei HeyGen auf „Photo Avatar“ und dann „Upload TalkingPhoto“ klicken und das Bild hochladen. Erscheint es dann in der Bilderliste mit der Maus auf das Pfeil-Icon fahren und im aufklappenden Kontextmenü über einen Klick auf „Landscape“ auf Querformat umstellen. Um das Bild im Kreis auf das volle Format 16:9 zu erweitern, im sich öffnenden Editor mit rechter

Maustaste auf das Bild klicken und statt „Circle view“ „Half-body“ selektieren. Dann den gewünschten Text eingeben und rechts eine deutsche Stimme auswählen, bei mir ist das „Killian“. Mit „Play Scripts“ kann das Ergebnis der Spracherzeugung geprüft werden, „Submit“ erstellt dann das Video – erst dann ist die Animation zu sehen.

Aus dem so erstellten Video kann dann im Kontextmenü über die drei Punkte oben rechts „Edit as New“ ausgewählt und mit der geänderten Texteingabe das Video für die Verabschiedung (Punkt 4) erstellt werden.

Für das erste Video in der Liste (Stumm) gibt es unter der Texteingabe ein Uhr-Icon für eine Pause von einer halben Sekunde. Die Textzeile wird damit für etwa 10–15 Sekunden gefüllt. Dass hierfür automatisch eine andere

JSON-Schlüssel für Google TTS/STT

```
{
  "type": "service_account",
  "project_id": "PROJECT_ID",
  "private_key_id": "KEY_ID",
  "private_key": "-----BEGIN PRIVATE KEY-----\nPRIVATE_KEY\n-----END PRIVATE KEY-----\n",
  "client_email": "SERVICE_ACCOUNT_EMAIL",
  "client_id": "CLIENT_ID",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://accounts.google.com/o/oauth2/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/SERVICE_ACCOUNT_EMAIL"
}
```

Stimme ausgewählt wird, ist in diesem Fall egal – sie sagt ja nichts.

Und zum Schluss braucht man noch ein Video für Punkt 3 in der Liste, die Antwort von ChatGPT. Eine wirklich lippen-synchrone Ausgabe der Antwort von ChatGPT zu generieren dauert leider zu lange, deshalb habe ich ein Video mit einer Standardantwort von rund 15 Sekunden Dauer erstellt, das ich dann später mit dem Text der echten Antwort unterlege. Das ist leider nicht perfekt, funktioniert aus meiner Sicht aber ganz Okay. Schaut euch mein YouTube-Video dazu an (Link in der Kurzinfo), dann könnt ihr selbst beurteilen, ob diese Lösung für euch taugt. Der Text der Standardantwort ist ziemlich egal, ich habe einen Abschnitt aus Wikipedia über München genutzt.

Nach diesen Schritten und den Downloads der Dateien von heygen.com sollten nun die vier MP4-Dateien (in meinem Fall Frag_DiWa_Hallo.mp4, Frag_DiWa_Denken.mp4, Frag_DiWa_Antwort.mp4, und Frag_DiWa_Goodbye.mp4) auf dem PC vorliegen. Statt einer Comic-Umsetzung des eigenen Gesichts als Avatar kann man je nach Geschmack natürlich auch andere Bilder oder Charaktere verwenden – von der Mona Lisa bis Max Headroom. Der Kreativität sind hier kaum Grenzen gesetzt.

API von OpenAI

Weiter geht es mit den APIs. Um mit ChatGPT zu kommunizieren, sei es über die Web-Oberfläche (siehe Link in der Kurzinfo) oder per API, ist eine Registrierung bei openai.com notwendig. Die Schritte zur Anmeldung hatten wir bereits in den vergangenen Heften beschrieben (siehe „Mehr zum Thema“ in der Kurzinfo), deshalb hier nur ganz kurz: Nach der Registrierung bei OpenAI kann unter „Personal/View API keys“ ein API-Schlüssel für die Kommunikation zwischen dem Raspberry Pi und ChatGPT erzeugt werden. Diesen fügt man später in den Python-Code ein und muss ihn direkt beim Erzeugen speichern, denn er wird später nicht mehr angezeigt! Das initiale (jedoch zeitlich begrenzte) Budget ist bei „Manage Account“ einsehbar, es steht aber nur bei Neuregistrierung zur Verfügung. Für die API-Aufrufe durch den Sprachassistenten reicht der Betrag zum Ausprobieren gut aus.

APIs von Google

Um die Spracheingabe in Text umzuwandeln, und um die Text-Antwort von ChatGPT wieder als Sprache auszugeben, wird Google STT (Speech-to-Text) und TTS (Text-to-Speech) verwendet. Komplette kostenfrei ist das auf Dauer leider nicht, allerdings kann man innerhalb des 90-tägigen Testzeitraums Dienste über ein Testguthaben von derzeit 300 US-Dollar verfügen, ohne was dafür zahlen zu müs-

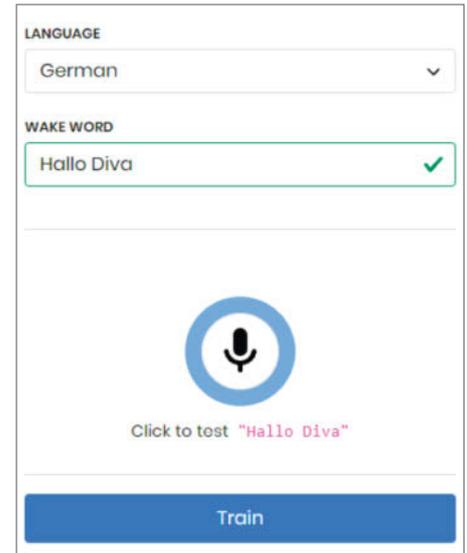
sen, wofür sich schon sehr viele Anfragen starten lassen. Und danach sind die Kosten immer noch recht überschaubar: Je nach verwendeter Stimme sind bei Text-To-Speech eine bis vier Millionen Zeichen pro Monat kostenlos, dann kostet eine weitere Million beispielsweise 16-US-Dollar; die Preise für Speech-To-Text werden nach Minuten berechnet und kosten je nach API-Version ab etwa 1 Cent pro Minute. Die Webseiten mit den Preisinformationen haben wir über die Kurzinfo verlinkt.

Im Netz lässt sich die Google-Sprachausgabe testen (Link in der Kurzinfo). Dort kann man dann eine passende Stimme auswählen, denn diese sollte idealerweise ähnlich der bei HeyGen gewählten Stimme sein.

Prinzipiell sind folgende Schritte zu tun:

1. Bei der Google Cloud Platform registrieren (console.cloud.google.com),
2. Ein neues Projekt erstellen und oben links in der Dropdown-Liste auswählen,
3. Unter „APIs und Dienste“ den Service „Cloud Text-to-Speech“ suchen und aktivieren. Hier muss nun leider bereits ein Abrechnungskonto erstellt, sprich: Kreditkartendaten hinterlegt werden, auch wenn die Karte zunächst nicht belastet wird,
4. Dann noch den Service „Cloud Speech-to-Text“ suchen und aktivieren,
5. Unter „APIs und Dienste/Anmeldedaten“ jetzt noch ein Dienstkonto erstellen,
6. Danach unter „IAM und Verwaltung/Dienstkonten“ beim Dienstkonto den JSON-Schlüssel erstellen und auf den PC downloaden.

Das Ergebnis sieht in etwa so aus wie im Kasten „JSON-Schlüssel“ auf Seite 82.



Das gewünschte „DiWa“ wird als Kunstwort leider nicht erkannt, mit „Diva“ funktioniert das Wake Word aber sehr gut.

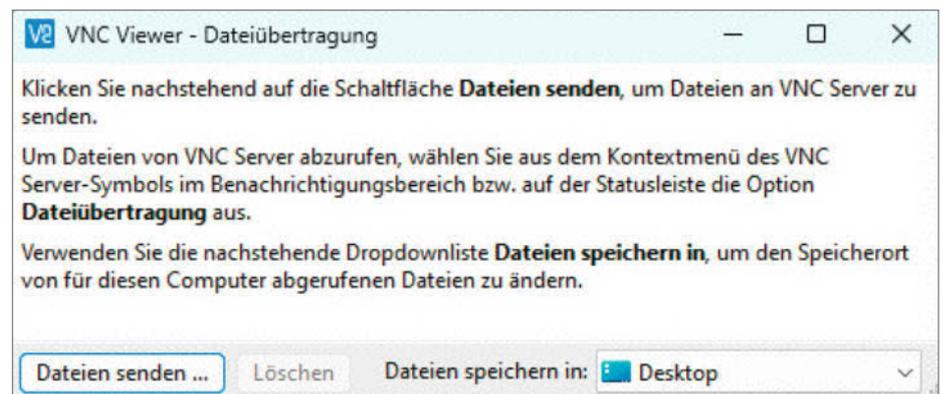
Python-Bibliotheken

```

pip3 install google-cloud-speech
pip3 install google-cloud-texttospeech
pip3 install python-vlc
pip3 install sounddevice
pip3 install pvrecorder
pip3 install pvporcupine
pip3 install picovoice
pip3 install python-dotenv
sudo apt-get install libportaudio2
    
```



Die auf dem PC vorbereiteten Dateien werden per VNC auf den Raspi übertragen.



Rolle und Arbeitsanweisung

```
Dialog_Gesamt = \[{'role': 'system', 'content': 'Du bist DiWa, der hilfreicher Sprachassistent. Wichtig: nenne bei Aufzählungen nicht "Erster", "Zweiter", oder 1., 2., und so weiter, sondern nur die Information durch Komma getrennt.'}\]
```

In die Linksammlung in der Kurzinfo haben wir noch ein ausführliches englisches YouTube-Video aufgenommen, das diese Schritte nochmal detailliert erklärt.

Eigenes Wake Word

Damit nun der Sprachassistent nicht ständig in den Raum lauscht und das gehörte sofort an Google bzw. ChatGPT schickt, wird noch ein Wake Word benötigt. Dieses kann man etwa bei picovoice.ai erzeugen. Für den Python-Code wird sowohl der dort gezeigte API-AccessKey benötigt als auch die PPN-Datei mit dem Wake Word. Diese generiert man durch Klick auf die Schaltfläche „Porcupine Wake Word“ im Abschnitt „Start Building“.

Anschließend wählt man Raspberry Pi als Plattform und kann dann eine Zip-Datei herunterladen, die die nötige PPN-Datei enthält. Falls ein deutsches Wake Word verwendet wird, ist auch noch zusätzlich die Datei `porcupine_params_de` notwendig (Download über den Link).

Start und Stopp

Damit man später weiß, ab wann der Sprachassistent die eigene Frage mitschneidet und ab wann er nicht mehr zuhört, gibt es noch die beiden Dateien „start.wav“ und „stop.wav“ (Download über den Link in der Kurzinfo; dort haben wir auch eine Quelle für alternative Töne verlinkt). Mein Assistent hat eine voreingestellte Dauer für die Spracheingabe von fünf Sekunden, die durch diese beiden Töne eingerahmt werden (siehe Video). Diese kann nach Bedarf im Code aber noch angepasst werden.

Python-Code

Jetzt geht es auf den Raspberry Pi: Erstmal mit dem Pi Imager (siehe Link in der Kurzinfo) das Raspberry Pi OS (64-Bit) auf die SD-Karte aufspielen und in den Raspi einstecken. Es empfiehlt sich die 64-Bit-Version, mit der 32-Bit-Version kann es zu Fehlern kommen. Nach der initialen Konfiguration und dem Update der

Programme wie beim Pi üblich sollte bei den „Einstellungen/Pi Konfiguration/Schnittstellen/VNC“ aktiviert werden. Das hilft sehr, um den Sprachassistenten später vom PC aus fernsteuern zu können und um die bisher auf dem PC gespeicherten Dateien wie die Avatar-Videos einfach übertragen zu können. Rechts unten im Tray ist ein VNC-Icon, dort steht dann auch die IP-Adresse des Raspis. Das Viewer-Programm für Windows gibt es bei RealVNC zum Download (Link siehe Kurzinfo).

Sobald das alles erledigt ist, müssen auf dem Raspberry Pi im Terminal erst einmal etliche Python-Bibliotheken installiert werden (siehe gleichnamiger Kasten auf Seite 83). Die Liste gibt es auch noch mal online für Copy & Paste, sodass man sie nicht abtippen muss.

Sobald das erledigt ist, werden alle zehn bisher erstellten Dateien vom PC mit VNC auf den Raspberry Desktop geladen (vier Videos, der JSON-Schlüssel, die PPN-Datei mit dem Wake Word sowie der Porcupine-Parameter für Deutsch, Start- und Stopp-Sound und



Es geht noch kleiner: Hier die mobile Variante mit einem 3,5-Zoll-Touch-Display.

So sieht mein erster Aufbau des Sprachassistent von hinten aus, mit Akku zur mobilen Stromversorgung. WLAN unterwegs kommt vom Handy mit Hotspot.

schließlich das eigentliche Python-Programm, herunterzuladen über den Link in der Kurzinfo). Dazu wird auf dem PC der VNC-Viewer geöffnet, mit der IP-Adresse des Raspberrys eine Verbindung hergestellt und dann über das Menü des VNC die Dateien auf den Raspberry übertragen. Die Dateien werden auf dem Raspberry Pi einfach auf den Desktop gelegt. Wer sie woanders unterbringen will, muss im Code die Pfade anpassen.

Test des Sprachassistenten

Jetzt muss man im Code des Python-Programms Frag_DiWa.py noch an den entsprechenden Stellen die eigenen API-Keys einfügen und gegebenenfalls die Dateinamen von Videos und Sounddateien anpassen. Ist statt einer männlichen eine weibliche Stimmsender, stellt man unten im Code bei:

```
saml_gender = texttospeech
    .SamlVoiceGender.MALE
```

noch auf FEMALE um.

Dann kann das Python-Programm Frag_DiWa.py mit Thonny auf dem Raspberry Pi testweise gestartet werden.

Man kann dem Assistenten eine initiale Rolle zuweisen sowie eine Arbeitsanweisung für ChatGPT. Wie die in meinem Fall lautet, steht im Kasten.

Wenn alles richtig benannt und verlinkt ist, dann sollte der Sprachassistent nun laufen. Die Videos werden im VLC-Player angezeigt. Nach Nennung des Wake Words sollte der Assistent die Frage verarbeiten und das Ergebnis von ChatGPT ausgeben. Falls man keine Antwort hört, ist möglicherweise noch der Lautsprecher des Raspberry auf AV und nicht auf HDMI eingestellt.

Man kann nun eine Frage nach der anderen stellen. Spricht man nichts ins Mikrofon, geht der Assistent wieder in den Ruhemodus und wartet auf das Wake Word. Sagt man „DiWa beenden“, wird das Programm beendet und der Raspberry automatisch heruntergefahren.

Während bei den Videos für Begrüßung und Abschied der VLC-Player sowohl die Animation anzeigt als auch die von HeyGen dafür erzeugte Sprachausgabe abspielt, wird bei der Antwort auf die Frage die VLC-Lautstärke auf 0 gesetzt und das Video stattdessen mit der von Google Text-to-Speech erzeugten Sounddatei unterlegt. Solange die abgespielt wird,

wiederholt sich das Video dazu in einer Endlosschleife.

Automatischer Start

Soll nun der Assistent direkt nach dem Booten starten, sind noch diese Eingaben im Terminal notwendig (die beiden eingerückten Zeilen bitte direkt an die vorangegangene anhängen):

```
chmod +x /home/pi
    /Desktop/Frag_DiWa.py
sudo nano /etc/xdg/autostart
    /Frag_DiWa.desktop
```

Dort diesen Eintrag ergänzen (die dritte Zeile bitte mit einem Leerzeichen getrennt an die zweite anhängen):

```
\[Desktop Entry\]Name=Frag_DiWa
Exec=/usr/bin/python3
    /home/pi/Desktop/Frag_DiWa.py
```

Danach mit Strg+O speichern und mit Strg+X schließen und anschließend rebooten.

Das war's. Viel Spaß beim Aufbau und beim Unterhalten mit dem selbstgebaute Sprachassistenten. —pek

ICH WARTE NICHT AUF UPDATES. ICH PROGRAMMIERE SIE.



40%
Rabatt!



ct MINIABO PLUS AUF EINEN BLICK:

- 6 Ausgaben als Heft, digital in der App, im Browser und als PDF
- Inklusive Geschenk nach Wahl
- Zugriff auf das Artikel-Archiv
- Im Abo weniger zahlen und mehr lesen

Jetzt bestellen:

ct.de/angebotplus



DIY-Timeout-Uhr

Eine Tischtennis-Uhr kostet im Handel zwischen 400 und 500 Euro. „Zu viel Geld!“, dachte sich unser Autor. Also ist Selberbauen angesagt.

von Jochen Ferger



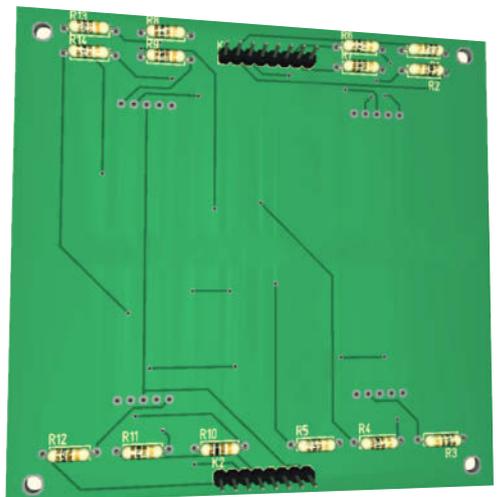
Timeout! ruft eine Spielerin beim Tischtennis. In den Bundes- und höheren Ligen stellt nun der Schiedsrichter eine Timeout-Uhr auf den Tisch. Diese zählt von 60 Sekunden herunter, anschließend geht das Spiel weiter. Zum Einsatz kommt die Uhr ebenso beim Satzwechsel (1 Minute), beim Einspielen (2 Minuten) und beim Spielwechsel (5 Minuten).

Der Autor dieses Artikels ist Abteilungsleiter Tischtennis vom Diezer TSK Oranien und begeisterter Maker. Für seinen Verein baute er eine solche Uhr auf Grundlage eines Arduino Uno selber.

Die dafür eingesetzten Platinen hat der Autor in Target gezeichnet und in China produzieren lassen. Die konstruierte Steuerplatine ist ein Shield.

Die Anzeige besteht aus 7-Segment-Anzeigen, solche mit 56,9 mm Höhe sind gut geeignet. Da sie jedoch eine höhere Spannung benötigen, werden sie nicht direkt von den 5-V-Ausgängen des Arduinos angetrieben, sondern über Treiber-ICs ULN2803 mit 9 V.

Bei den Treibern handelt es sich im Endeffekt um mehrere Schalter, die über einen kleinen Steuerstrom größere Ströme schalten (Darlington Transistoren). Für eventuelle Erweiterungsmöglichkeiten ist das Port-Erweiterungs-IC MCP23017 verbaut. Es verwendet den I²C-Anschluss des Arduinos (zwei Pins auf dem Board) und stellt dafür 16 weitere Ein- und Ausgänge



Achtung: Der gemeinsame Spannungsanschluss der Segmentanzeigen befindet sich rückseitig betrachtet ganz rechts.

zur Verfügung. Die Anzeigeplatine wird dreimal benötigt, das Steuer-Shield einmal.

3D-Druck

Zu Beginn waren nur zwei Zeiten geplant, daher hat die Bedieneinheit im Deckel der Uhr nur zwei Taster. Ursprünglich war vorgesehen, den Arduino samt Shield auf dem Boden der Uhr zu verschrauben, mangels Platz wurde das verworfen. Der Arduino liegt jetzt – ebenso wie der Halter für die 9-V-Block-Batterie – unten in der Uhr. Batteriehalter und Steuerelektronik lassen sich von innen an die Seitenwand kleben.

Die Gehäuseteile der Uhr sind mit Fusion 360 realisiert worden. Deckel und Boden bestehen aus PLA+ und wurden mit einem Anycubic i3 ausgedruckt, ebenso die Standfüße aus TPU. Da man die Uhr bei einem Timeout oder bei Satzwechseln auf den Tischtennistisch stellt, werden so Kratzer auf der Platte vermieden.

Für die drei Seitenteile braucht man aufgrund der Höhe von 25 cm einen größeren Drucker, in diesem Fall einen Anycubic Chiron.

Bestückung der Platinen

Die Anzeigeplatine bekommt zuerst Widerstände, die alle einen Wert von 100 Ohm haben und als 1/4-Watt-Variante ausgelegt sind. Die Widerstände kann man von beiden Seiten auflöten. Im Bild sind sie auf der Rückseite angebracht.

Beim Einsetzen und Anlöten der 7-Segment-Module muss man auf die leicht verbiegbaren Beinchen achten. Zuletzt lötet man die Steckkontakte auf die Rückseite auf. Hier können gewöhnliche Steckerleisten oder PSK- oder JST-Stecker verwendet werden.

Bei der Verkabelung muss man darauf achten, dass die Kabel lang genug sind. Für die

Kurzinfo

- » Selbst erstellte Platinen
- » Verwendung von 7-Segment-Anzeigen
- » Verschiedene Filamente für verschiedene Ansprüche

Checkliste



Zeitaufwand:
6 Stunden



Kosten:
90 Euro

Werkzeug

- » Lötkolben
- » Schraubenzieher
- » 3D-Drucker

Mehr zum Thema

- » Benno Lottenbach, Ping-Pong-Trainingsroboter, Make 6/22, S. 36
- » Ulf Mayer, WLED mittels Herzfrequenz steuern, Make 2/23, S. 40

↓ Alles zum Artikel im Web unter make-magazin.de/x9e1



Material

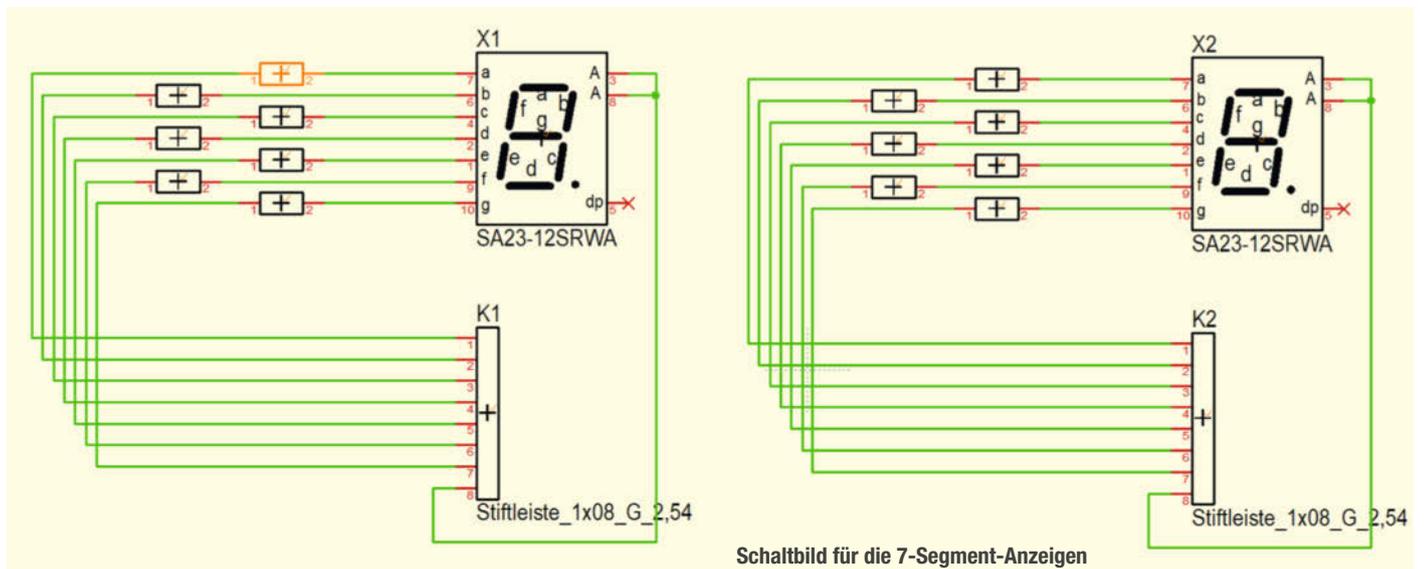
- » Arduino Uno
- » 6 7-Segment-Anzeige Typ SA23-12SRWA
- » 2 Taster schraubbar
- » Kippschalter
- » 9-V-Batteriehalter
- » Filament PLA+ und TPU
- » 2 Transistor-Schalter ULN2803
- » Port-Erweiterungs-IC MCP23017
- » 15 Meter 0,2 mm² Draht
- » Leiterplatten vom Dienstleister
- » 14 Widerstände 100 Ohm
- » Widerstand 4,7 Kiloohm
- » 8 JST-Stiftleiste 8 Pins
- » JST-Stiftleiste 6 Pins
- » 2 Transistor-Array Typ ULN 2803A
- » I/O Expander Typ MCP23017
- » Kippschalter 5 mm
- » 2 Drucktaster 16 mm
- » 12 ABS-Abstandshalter
- » Schrauben M2, M2,5, M3
- » Muttern M2, M2,5, M3
- » TPU
- » PLA+

Kabel der oberen Steckerleiste sollte man mindestens 30 cm einplanen.

Steuer-Shield

Das Steuer-Shield wird von unten nach oben gelötet – zuerst der 4,7-Kiloohm-Wider-

stand, dann die drei ICs. Hierbei muss die Polarität beachtet werden. Zuletzt lötet man die Steckkontakte. Für die Verbindungen zu den Anzeigemodulen kann man normale Steckerleisten oder wieder PSK- bzw. JST-Kontakte verwenden. Die Kontakte für die linken Stellen der Displays sind auf dem



Steuer-Shield K4 bis K6 und auf der Anzeigeplatine die untere Steckkontaktleiste.

Für die Verbindung zum Arduino verwende ich gerne die nicht ganz so häufig verfügbaren Steckerleisten mit 2 mm längeren Kontakten. So entsteht ein Abstand zwischen Arduino und Shield.

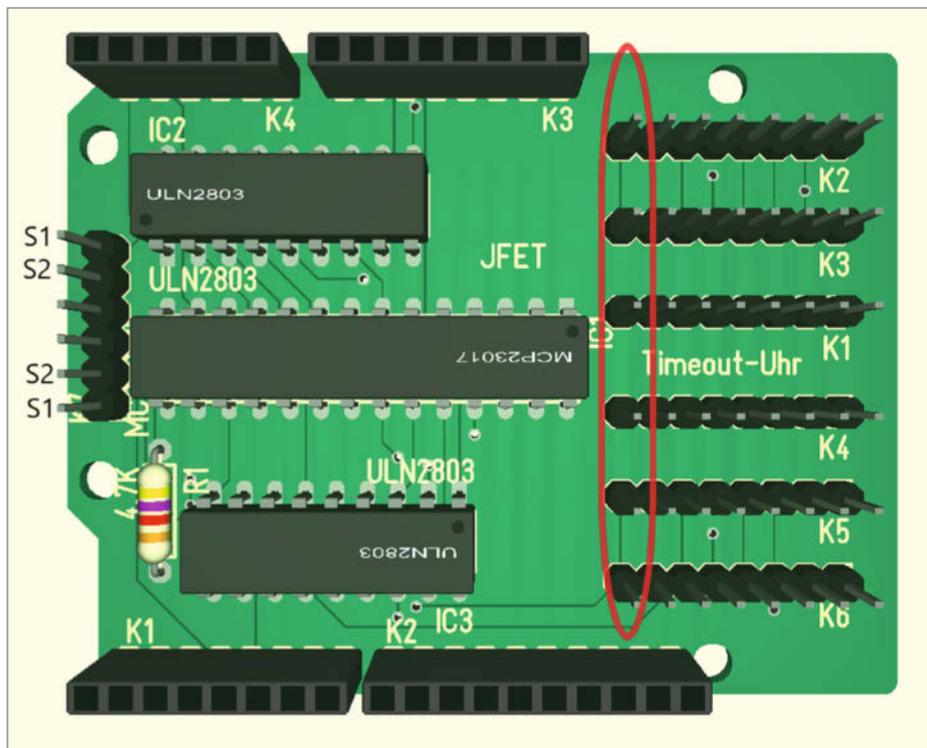
Zusammenbau

Die Taster und der Kippschalter werden mit je ca. 30 cm langen Kabeln verdrahtet. Anschließend setzt man Taster und Kippschalter in den Deckel und verschraubt alles. Zuletzt wird der Griff auf den Deckel gesetzt. Man verschraubt

ihn, indem die Mutter (M3) im Griff mit einer Pinzette fixiert und die Schraube (M3×18) mit Unterlegscheibe unten von der Deckelseite eingeschraubt wird.

Die Anzeigeplatinen montiert man mit einem Abstandshalter und Schrauben (M2×16), Unterlegscheiben und Muttern (M2) in die Seitenteile.

Die Füße werden wie der Griff im Deckel mit Schrauben (M2,5×14) befestigt.



Spannungsversorgung

Die Kontakte des Batteriefachs werden so mit den Kabeln vom Kippschalter verlötet, dass der Plus-Kontakt geschaltet wird. Plus und Minus werden mit dem Stromkabel am Arduino verlötet und mit Schrumpfschläuchen gegen Kurzschluss gesichert.

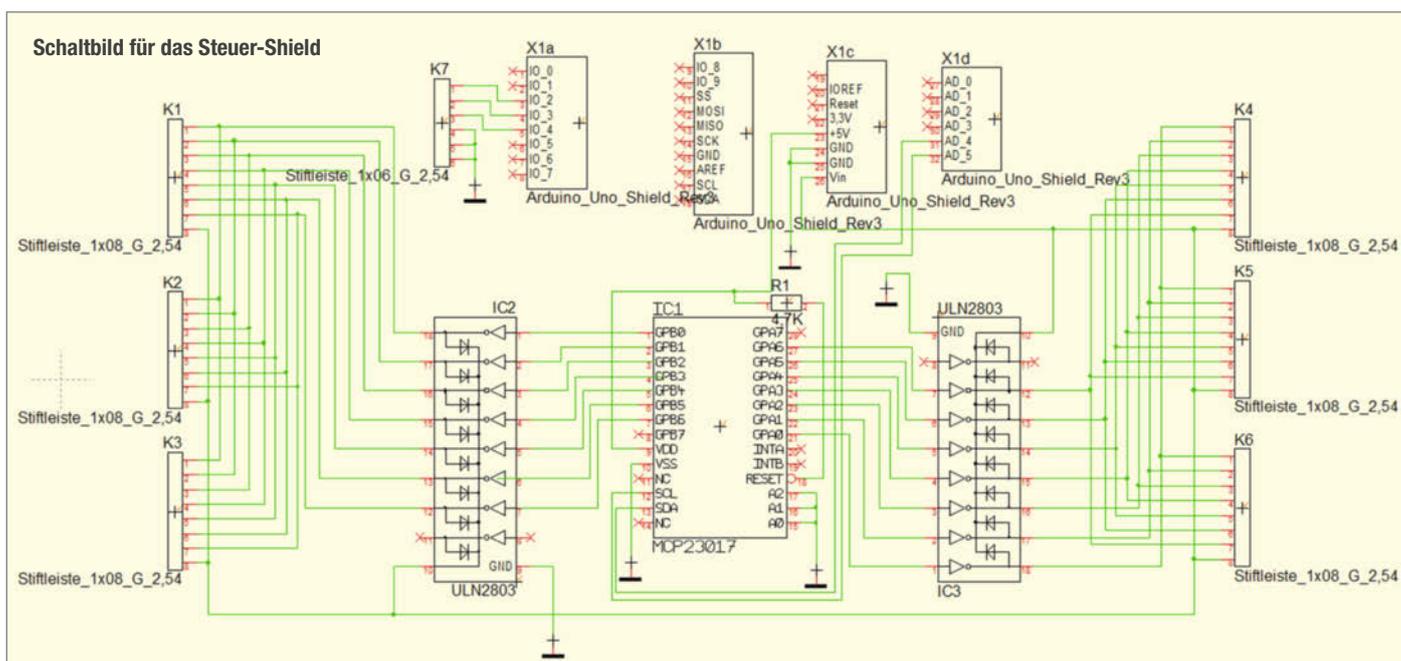
Endmontage

Nach der kompletten Verdrahtung werden die Seitenteile nacheinander an den Deckel geschraubt. In der Konstruktion der Seitenteile sind Gewinde (M3) in den Halterungen konstruiert. Durch Stützmaterial und Ungenauigkeiten des Druckers kann man hier auch oft eine Schraube M2,5 verwenden. Alternativ lassen sich im Deckel leicht Muttern von unten dagegen schrauben. Bevor jetzt der Deckel auf die gleiche Art und Weise wie der Boden verschraubt wird, muss noch die Software auf den Arduino aufgespielt werden.

Der gemeinsame Spannungsanschluss für die Segmentanzeigen ist von oben betrachtet der linke Pin. Die Taster S1 und S2 werden an der Steckerleiste links gemäß der Abbildung verdrahtet. Die Reihenfolge ist hierbei später im Quellcode einstellbar.

Programmierung

Die Software wird über USB von einem anderen Computer aufgespielt. Nachdem man



den Arduino verbunden hat, öffnet man die Datei Timeout_Uhr.ino mit der Arduino IDE. In dem Programm unter „Board/Arduino AVR Boards“ Arduino Uno auswählen und unter Ports den verbundenen Arduino Uno einstellen. Dann auf Upload klicken. Eine genauere Erklärung für diesen Vorgang ist in der Kurzinfor verlinkt. Dort befindet sich auch der Link zum Code.

Danach wird die Funktion geprüft. Sollten die Taster vertauscht sein, kann man einfach in Zeile 22 und Zeile 25 des Quelltextes die Eingangs-Pins 2 und 3 austauschen.

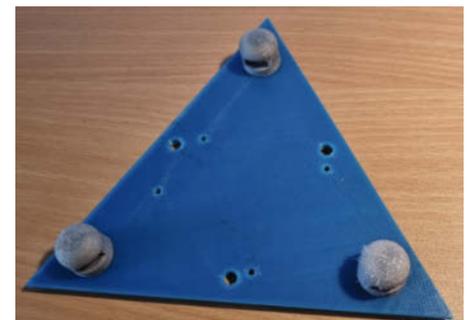
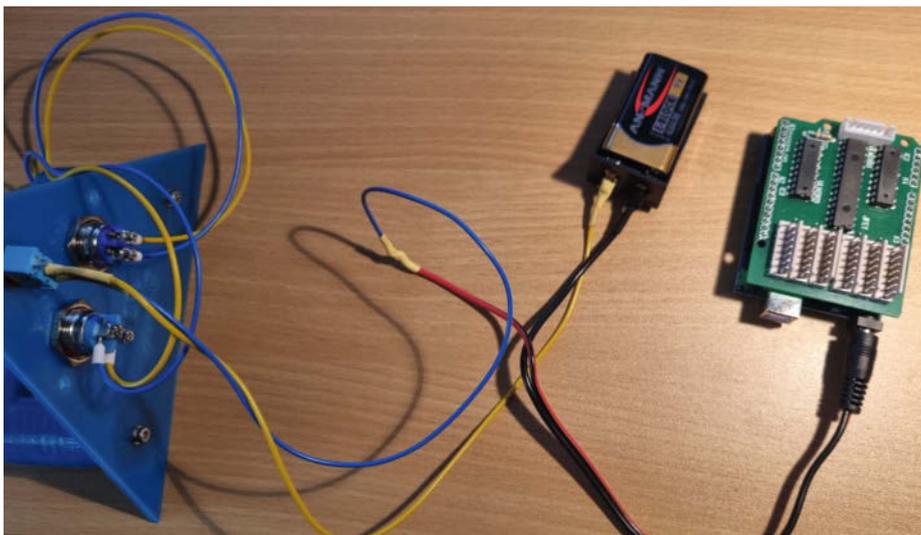
Spiel, Satz und Sieg

Und so ist mit scharfem Auge und ruhiger Hand diese Uhr nun nun ein Hingucker bei Mann-

schaftsspielen. Auch wenn eine solche Uhr in den unteren Spielklassen nicht so oft Anwendung wie in Bundesligen findet, ist sie nun ein Alleinstellungsmerkmal des Diezer TSK Oranien. Aber nicht nur an der Tischtennisplatte geht es weiter, sondern auch an der Maker-Werkbank. Denn nicht nur im Sport kann man sich stetig steigern. Eine weitere Uhr ist als lasergeschnittene Edelstahlvariante in Planung. —das



Die 7-Segment-Anzeigen werden von Innen gegen die Seitenwände geschraubt.



Die weichen TPU-FüÙe schützen die Tischtennisplatte vor Kratzern.

Verkabelung der Spannungsversorgung und der Schalter

IX SPECIAL 2023
Künstliche Intelligenz
 verstehen, anwenden, selbst entwickeln

Was große Sprachmodelle können
 Wie funktioniert GPT-4, Bard und Co. und wie setzt man sie ein?
 Hugging Face: die Zentrale für Open-Source-AI
 Stable Diffusion: so geht's

Eigene Projekte ganz konkret

Heft + PDF mit 29 % Rabatt

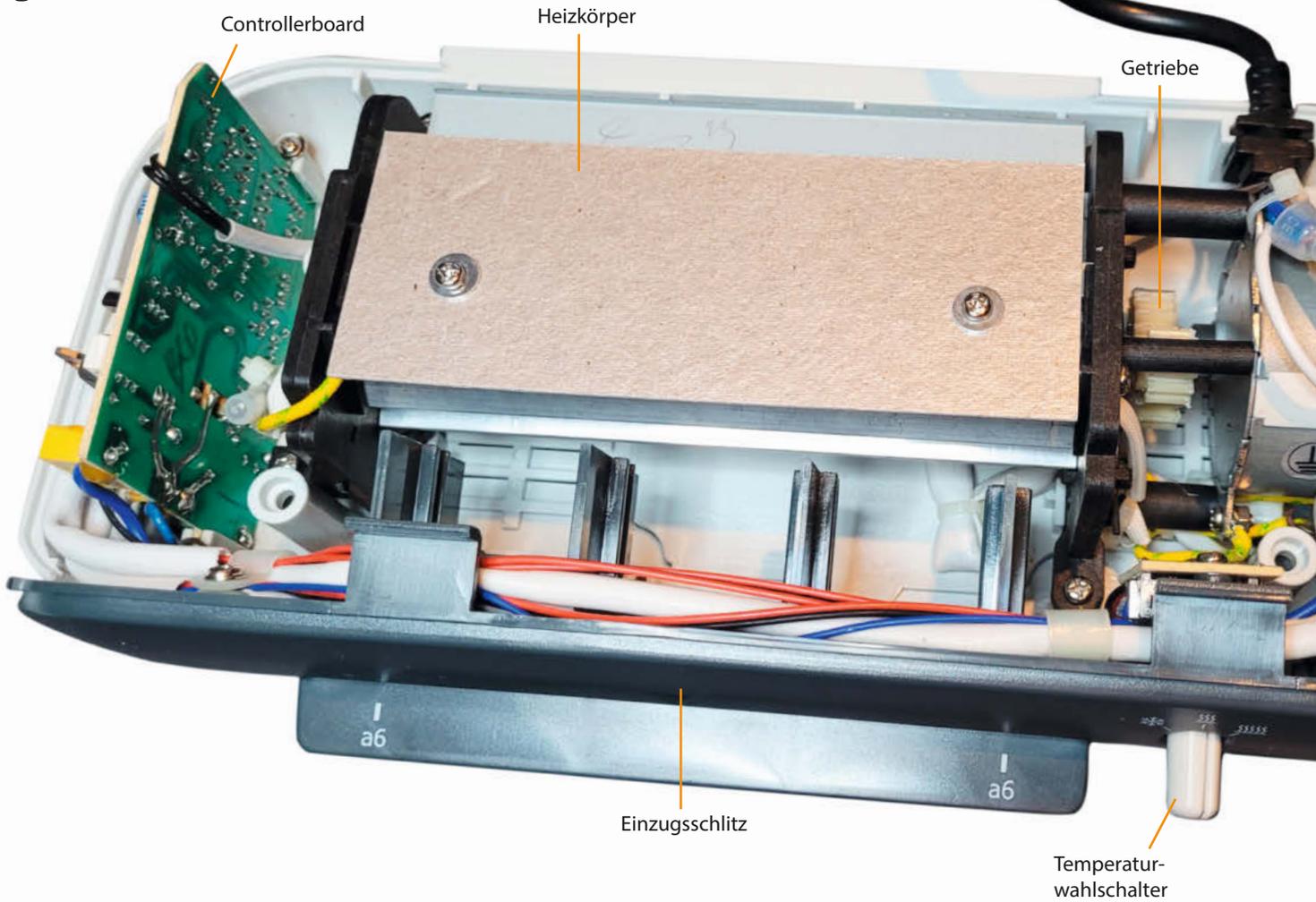
Do KI Yourself!

Modelle anwenden und selberrnachen

Heft für 14,90 € • PDF für 14,90 €
 Bundle Heft + PDF 20,90 €

shop.heise.de/ix-ki

Reingeschaut



Laminiergerät

Sie gehören zu der Klasse von Geräten, die man anschafft, einmal benutzt und dann nie wieder. Zum Wegwerfen zu schade, zum Reinschauen ideal, denn Laminiergeräte verbinden verschiedene Disziplinen der Ingenieurskunst und eignen sich prima zum Ausschlichten.

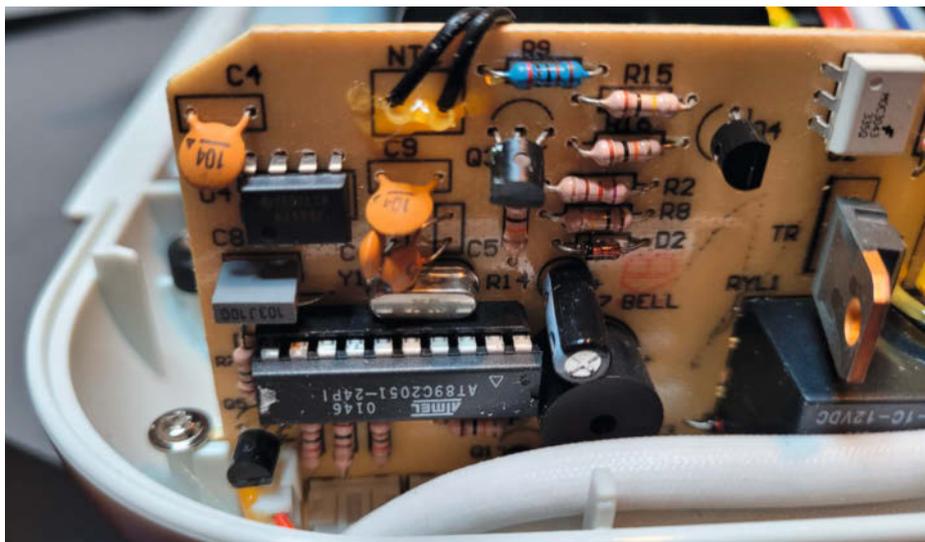
von Daniel Bachfeld

Selbst gedruckte Fotos, wichtige Dokumente, Mitgliedsausweise: Zum Schutz kann man sie in Hüllen stecken oder laminieren. Dazu setzt man in der Regel auf Folien mit einer Klebeschicht, die durch Erhitzen weich gemacht und mit dem Dokument von beiden Seiten verschmolzen respektive verklebt wird. Zum Schmelzen und Zusammenpressen dienen Heißlaminiergeräte, die es unter anderem zu jedem Schulhalbjahr bei einschlägigen Lebensmittelketten gibt. Das hier geöffnete Gerät eignet sich nur zum Laminieren kleiner Dokumente.

Ein Heizelement bringt einen Metallkörper und die in ihm enthaltenen Gummiwalzen auf die notwendige Temperatur. Die Temperatur der unteren Walze wird von einem NTC, also einem temperaturabhängigen Widerstand, aufgenommen und der Spannungsabfall von einem Mikrocontroller AT89C2051 ausgewer-



Unter dem Metallkörper ist das Heizelement angeschraubt. Rechts daneben im Hintergrund die Lichtschranke.



Die Steuerung übernimmt ein 8051-Mikrocontroller von Atmel (heute Microchip).

tet. Der Controller regelt mit einem Triac (MOC3034) die Phasenanschnitte der Netzspannung, um das Heizelement auf die richtige Temperatur zu bringen und dort zu halten. Ist das erreicht, schaltet er kurz den Buzzer an. Schiebt man dann ein Dokument in den Schlitz, so unterbricht es die Lichtschranke. In der Folge schaltet der Controller per Relais den Synchronmotor an, um das Dokument einzuziehen.

Ein Getriebe sorgt für eine Umsetzung des Motorantriebs, um das Dokument langsam durch die Rollen zu führen. So hat die Laminierhülle genügend Zeit, warm zu werden und die Klebeschicht anzuschmelzen.

Der AT89C2051 von Atmel ist im Prinzip ein frei programmierbarer 8-Bit-Mikrocontroller, den es seit 1980 am Markt gibt: MCS-8051 (kurz MCS-51), erfunden von Intel. Allerdings hat die moderne Version 2 KByte eingebauten Flash und 128 Byte RAM an Bord. Sein Flash ist ein „programmable and erasable read-only memory“ (PEROM), also mehrfach wiederbeschreibbar. Dazu benötigt man jedoch einen speziellen Flasher wie den TL866, der eine Programmierspannung von 12 V statt der üblichen 5 V liefern kann, um sie an die vorgesehenen Eingänge des Controllers zu legen.

Der IC ist gesockelt und ließe sich im Fehlerfall leicht austauschen. Falls man den Laminierer ausrangiert, kann man den Mikrocontroller später in eigenen Projekten

weiterverwenden. Mit dem Small Device C Compiler (SDCC) gibt es einen freien Compiler, Platformio unterstützt den Mikrocontroller ebenfalls. Fun Fact: Über die Architektur des Mikrocontrollers gibt es durchaus Diskussionen. Ist es Harvard oder von-Neumann? Platformio schreibt „Harvard“. Was meinen Sie?

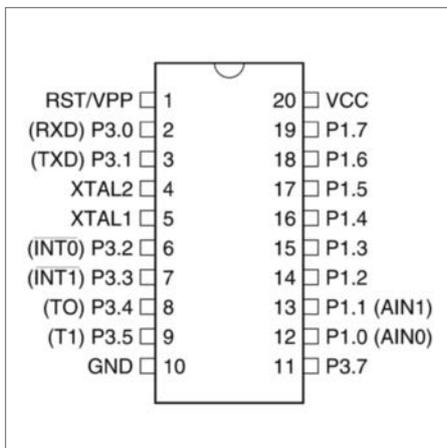
Darüber hinaus hat der Laminierer noch viel mehr Potenzial zum Ausschichten: Relais, Buzzer, Heizelement, Motor, Taster, Drehknopf. Die nicht gesockelten Bauelemente wären leicht auszulöten. Andere Dinge sind nur gesteckt oder verschraubt.

Ohnehin ist der sorgfältige und reparaturfreundliche Aufbau des Gerätes auffällig. Geschraubte Kabelhalter und Kabelbinder halten die Verbindungen an ihren Plätzen, Metallteile sind ordentlich geerdet, Kabelenden sind mit Steckschuhen versehen. Nichts ist mit Heißkleber lieblos an andere Teile geklatscht.

Für ein in China gefertigtes Gerät ist das bemerkenswert. Das Baujahr des Modells schätzen wir auf 2014. Ob früher™ in China alles besser war oder der Hersteller viel Wert auf Haltbarkeit, Wartbarkeit und Sicherheit gelegt hat, wissen wir nicht. Aber diese Qualität finden wir nicht mehr so oft beim „Reingeschaut“.

—dab

► make-magazin.de/x71h



Die Pinbelegung des MCS-8051 kompatiblen Mikrocontrollers. Sieht zwar aus wie ein ATmega, ist aber keiner.



Der NTC wird mit einer Feder an die Gummirolle gepresst.

UI-Design mit LVGL

Für ein modernes Nutzerinterface ist nicht immer ein Raspberry Pi nötig. Dank der LVGL-Grafikbibliothek laufen selbst aufwändige Benutzeroberflächen auf einfachen Embedded-Systemen wie dem ESP32. Wir zeigen, wie der Einstieg in LVGL gelingt.

von Maximilian Kern



Ob Smartwatch, Kopierer oder Waschmaschine: Immer mehr Alltagsgeräte sind mit Grafikdisplays ausgerüstet und bieten moderne, intuitiv bedienbare Nutzeroberflächen. Auch im DIY-Bereich sind hochauflösende Displays mittlerweile erschwinglich geworden. Darauf ein professionell aussehendes User Interface (UI) darzustellen, ist allerdings immer noch eine Herausforderung. Mit den üblichen Grafikbibliotheken wie Adafruit-GFX muss man Bedienelemente oft mühsam zusammensetzen, während es für Kantenglättung und Animationen an Rechenleistung mangelt. Ansprechende UIs waren deshalb lange Zeit Linux-Systemen wie dem Raspberry Pi vorbehalten.

Doch das ändert sich mit der „Light and Versatile Graphics Library“, kurz LVGL. Die effiziente Grafikbibliothek holt das Maximum aus Mikrocontrollern wie dem ESP32 heraus. So kann man aufwändige Benutzeroberflächen aus einer Vielzahl integrierter Widgets für z.B. Knöpfe, Schieberegler oder sogar einer Bildschirmtastatur zusammenstellen. Für all diese Elemente lässt sich die Funktion und das Design individuell anpassen. Selbst verschachtelte Menüführungen über mehrere Seiten und Untermenüs sind möglich.

LVGL ist noch ein recht junges Projekt und umfassende Tutorials sind schwer zu finden. Anhand einfacher Beispiele zeige ich in diesem Artikel, wie der Einstieg in LVGL gelingt und wie man ein eigenes Benutzerinterface entwirft.

Breadboard-Aufbau

Für Projekte mit LVGL sollte man einen Mikrocontroller wählen, der mindestens 180 kB Flash und 48 kB RAM besitzt. Dafür eignen sich zum Beispiel ESP32, STM32F oder Raspberry Pico. Ich verwende für mein Projekt den ESP32 Thing von SparkFun. Als Bildschirm kommt ein 2,8-Zoll großes TFT-Display von Adafruit mit kapazitivem Touchscreen zum Einsatz, das sich mit dem Finger in etwa so bedienen lässt, wie man es von Smartphones gewohnt ist.

Damit man das Adafruit-Display über den SPI-Bus ansteuern kann, muss man zunächst die Lötbrücken IM1, IM2 und IM3 auf der Rückseite des Breakout-Boards kurzschließen. Danach kann man den Lötcolben auch schon wieder weglegen. Alle weiteren Verbindungen zwischen ESP32 und Display geschehen nämlich auf einem Breadboard. Die 5V des ESP32-Boards versorgen das TFT-Display, sodass man den gesamten Aufbau über USB betreiben kann. Zusätzlich zum SPI (MOSI, SCK und CS) verbinden wir noch das D/C-Signal für den Displaycontroller. Um später auch den Touchscreen verwenden zu können, schließen wir noch den I²C-Bus an den ESP32 an (Bild 1). Und damit ist die Hardware auch schon bereit. Zusätzliche Display-Si-

Kurzinfo

- » **Nutzeroberflächen selbst gestalten**
- » **Erste Schritte mit der LVGL-Grafikbibliothek auf dem ESP32**
- » **Kapazitiven Touchscreen als Eingabemethode verwenden**

Checkliste

-  **Zeitaufwand:**
ca. 2 bis 3 Stunden
-  **Kosten:**
ca. 80 Euro

Material

- » SparkFun ESP32 Thing
- » Adafruit 2,8-Zoll-Display mit kapazitivem Touchscreen
Produktnr. 2090
- » Breadboard mit Kabeln

Mehr zum Thema

- » Maximilian Kern, Retro-Audio-Player, Make 5/22, S. 10
- » Ákos Fodor, Adafruit GFX Library: Malen mit Zahlen, Make 2/23, S. 64
- » Carsten Romahn, Touchscreen GUI mit dem Raspberry, Make 6/19, S. 98

Programmieren

- » **Grundkenntnisse in C++ und Arduino**

↓ Alles zum Artikel im Web unter make-magazin.de/xg3w



gnale, etwa für die Reset-Leitung und die Ansteuerung der Hintergrundbeleuchtung, sind auf dem Breakout-Board bereits vorkonfiguriert, sodass es nicht nötig ist, hier weiter zu beschalten.

Arduino IDE vorbereiten

Zum Programmieren verwenden wir die Arduino IDE 2.0. Wer noch nie mit dem ESP32 gearbeitet hat, muss diesen zunächst im Board Manager installieren (siehe Link in der Kurzinfo). Ist das geschafft, muss man noch drei Bibliotheken über den Bibliotheksverwalter installieren: lvgl, TFT_eSPI und Adafruit_FT6206.

Als Nächstes passen wir die Konfiguration der LVGL-Bibliothek an. Hierzu verschieben wir

Verbindungen

| ESP32 Thing | Touchdisplay |
|-------------|--------------|
| GND | GND |
| Pin 22 | SCL |
| Pin 19 | SDA |
| Pin 23 | MOSI |
| Pin 18 | CLK |
| Pin 5 | CS |
| Pin 17 | D/C |
| VUSB | Vin |

die Datei lv_conf_template.h aus dem Verzeichnis der Arduino-Bibliothek (Dokumente/Arduino/libraries/lvgl) eine Ebene nach oben (Dokumente/Arduino/libraries) und benennen

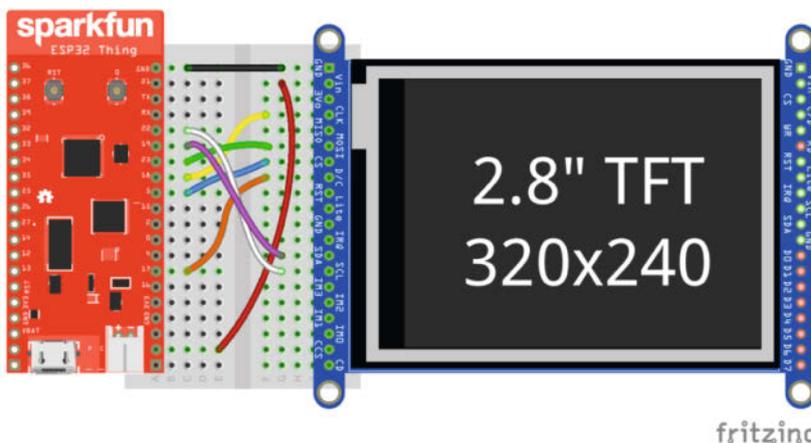


Bild 1: So muss man den ESP32 mit dem Adafruit-Display verbinden.



Hello World!

Bild 2: Hello World in LVGL

sie in `lv_conf.h` um. Innerhalb dieser Datei ändern wir das erste `#if 0` in `#if 1`. Jetzt muss nur noch `#define LV_TICK_CUSTOM` von `0` auf `1` gesetzt werden, damit wir den Arduino-Timer `millis()` für das Zeitmanagement verwenden.

LVGL selbst dient nur zur grafischen Darstellung und steuert Displays nicht direkt an. Zur Kommunikation mit dem Display benötigen wir deshalb noch die Bibliothek `TFT_eSPI`, die sozusagen als Grafiktreiber dient. Wieso ausgerechnet diese Bibliothek und nicht z.B. `Adafruit GFX`? Das hat Performance-Gründe: `TFT_eSPI` ist im direkten Vergleich zur `Adafruit GFX Library` um ein Vielfaches schneller. Und

das macht sich sehr deutlich bemerkbar, wenn man mit bewegten Grafiken arbeitet. Damit `TFT_eSPI` auf das Display zugreifen kann, müssen wir vor der ersten Verwendung noch die Pinbelegung in der Datei `User_Setup.h` im Bibliotheksverzeichnis von `TFT_eSPI` hinterlegen. In der Headerdatei in Sektion 2 unter „ESP32 Dev board ... ILI9341“ bearbeiten wir die folgenden Zeilen und entfernen die Kommentarsymbole:

```
#define TFT_MOSI    23
#define TFT_SCLK    18
#define TFT_CS      5
#define TFT_DC      17
#define TFT_RST     -1
```

Das erste Interface

Nachdem die Hardware, die IDE und die Bibliotheken startklar sind, können wir endlich das erste Benutzerinterface erstellen (Bild 2). Der benötigte Code für ein einfaches „Hello World!“ ist in Listing 1 dargestellt und wie folgt aufgebaut: Zuerst werden die Bibliotheken ein-

gebunden und Konstanten für die Bildschirmgröße eingeführt. Danach erstellen wir ein `TFT_eSPI`-Objekt mit der Bildschirmgröße unseres Displays und nennen es `tft`. Als Zwischenspeicher (Buffer) für das angezeigte Bild legen wir `disp_buf` und `buf` an. Zum Aktualisieren des Bildschirms dient die Funktion `my_disp_flush()`. Um deren Inhalt müssen wir uns dabei nicht weiter sorgen, da es sich um eine LVGL-spezifische Funktion handelt.

Im Arduino-Setup werden nacheinander LVGL, `TFT_eSPI` der Buffer und die Display-schnittstelle initialisiert. Erst danach beginnt der kreative Teil und ein Hello-World entsteht, das auf dem Display erscheinen soll. Hierzu definieren wir ein Label-Objekt. Grundsätzlich haben alle grafischen Objekte in LVGL den Typ `lv_obj_t`. Erst mit `lv_label_create()` wird das Objekt zum Text-Label. Durch die Übergabe von `lv_scr_act()` sagen wir, dass das Objekt auf der untersten Darstellungsebene liegen soll. Würden wir hier stattdessen ein anderes, bereits erstelltes Objekt angeben, verschachteln sich die Grafikobjekte in einer Hierarchie ineinander – mehr dazu später.

Mit `lv_label_set_text()` legen wir den Inhalt des Text-Labels fest. Damit das „Hello World!“ auf dem Display zentriert erscheint, nutzen wir die Funktion `lv_obj_align()` mit dem Parameter `LV_ALIGN_CENTER`. Mit diesen drei Zeilen ist der Textblock auch schon vollständig definiert. Auf diese Weise erstellt man in LVGL übrigens auch alle anderen Grafikobjekte, indem man sie erst initialisiert und anschließend anpasst.

Jetzt fehlt nur noch der Arduino-Loop, der den `lv_timer_handler()` aufruft, um die UI zu aktualisieren. Auch anderer Code, z.B. für Sensorabfragen, kann sich im Loop befinden, wenn es nicht zu lange dauert, diesen auszuführen. Als Nächstes überspielen wir den Sketch auf den ESP32. Daraufhin zeigt unser Bildschirm einen weißen Hintergrund und in der Mitte den Text „Hello World!“ an.

Input und Output

Um einer Benutzeroberfläche näherzukommen, ersetzen wir als Nächstes den statischen Text durch ein interaktives Element. LVGL lässt sich mit verschiedenen Eingabegeräten wie Mäusen, Touchpads und Dreh-Encodern bedienen und auch mit einem Touchscreen. Damit wir diesen verwenden können, müssen wir den Beispielcode ein wenig anpassen.



Bild 3: Dieser einfache Slider lässt sich über den Touchscreen bedienen.

Listing 1

```
#include <TFT_eSPI.h>
#include <lvgl.h>
#define screenWidth 240
#define screenHeight 320

TFT_eSPI tft = TFT_eSPI(screenWidth, screenHeight);

static lv_disp_draw_buf_t disp_buf;
static lv_color_t buf[screenWidth * 10];

void my_disp_flush(lv_disp_drv_t *disp, const lv_area_t *area, lv_color_t *color_p) {
    uint32_t w = (area->x2 - area->x1 + 1);
    uint32_t h = (area->y2 - area->y1 + 1);
    tft.startWrite();
    tft.setAddrWindow(area->x1, area->y1, w, h);
    tft.pushColors((uint16_t*)&color_p->full, w * h, true);
    tft.endWrite();
    lv_disp_flush_ready(disp);
}

void setup() {
    lv_init();
    tft.begin();

    lv_disp_draw_buf_init(&disp_buf, buf, NULL, screenWidth * 10);

    static lv_disp_drv_t disp_drv;
    lv_disp_drv_init(&disp_drv);
    disp_drv.hor_res = screenWidth;
    disp_drv.ver_res = screenHeight;
    disp_drv.flush_cb = my_disp_flush;
    disp_drv.draw_buf = &disp_buf;
    lv_disp_drv_register(&disp_drv);

    lv_obj_t *label = lv_label_create(lv_scr_act());
    lv_label_set_text(label, "Hello World!");
    lv_obj_align(label, LV_ALIGN_CENTER, 0, 0);
}

void loop() {
    lv_timer_handler();
    delay(5);
}
```

Zuerst binden wir die Bibliothek Adafruit_FT6206 ein und erstellen das Objekt touch. Die Funktion my_touchpad_read() bildet die Brücke zwischen LVGL und der Adafruit_FT6206-Bibliothek. Sie fragt ab, wo der Bildschirm berührt wurde und macht die aktuelle Position für LVGL zugänglich. Im Arduino-Setup initialisieren wir den Touchscreen danach mit touch.begin() und mappen ihn mit wire.begin(19, 22) auf die richtigen Pins. Wie schon beim Bildschirm benötigt der Touchscreen einige Zeilen Code, um ihn mit LVGL zu verknüpfen und zu konfigurieren.

Anstelle des „Hello World!“-Labels soll nun ein Slider erscheinen (Bild 3), den wir mit lv_slider_create() erzeugen. Diesen skalieren wir mit lv_obj_set_size() und zentrieren ihn danach mit lv_obj_align() in der Mitte des Bildschirms.

Der Code in Listing 2 zeigt beispielhaft, wie man den Regler grafisch auf dem Display abbildet. Um ihn auch praktisch nutzen und mit dem Finger bedienen zu können, muss man ihn global als lv_obj_t * slider deklarieren. Dadurch kann man über lv_slider_get_value(slider) die aktuelle Position des Reglers auslesen. Im GitHub-Repository des Projekts findet ihr einen erweiterten Sketch, der die zusätzlichen Zeilen beinhaltet. Auf ähnliche Weise kann man die Position des Sliders auch verändern, indem man die Funktion lv_slider_set_value() verwendet. Das Lesen und Schreiben von Eigenschaften lässt sich auch kombinieren, sodass man z.B. die Position des Sliders auslesen und diese in ein Textobjekt schreiben oder den Slider als Fortschrittsbalken benutzen könnte.

Styles und Bilder

Noch sieht der Slider in unserem Interface etwas einsam und langweilig aus. Damit das nicht so bleibt, passen wir ihn optisch an. Mit Styles bietet LVGL dafür unzählige Möglichkeiten. Man kann LVGL-Objekten zum Beispiel benutzerdefinierte Farben, Ränder und Schatten geben. Außerdem lassen sich mit wenig Aufwand Grafiken in die Benutzeroberfläche integrieren. Indem wir die Farben anpassen und ein Symbol einfügen, erweitern wir das einfache Slider-Beispiel im nächsten Schritt zu einer Glühbirnen-Steuerung (Bild 4). In Listing 3 ist der Codeausschnitt angegeben, der anstelle der bisherigen Slider-Definition von Listing 2 eingesetzt werden kann.

Jedes LVGL-Objekt verfügt über Style-Eigenschaften, die zunächst durch Standardwerte festgelegt sind. Analog zu den Funktionen, mit denen man Objekte positioniert und skaliert, gibt es Style-Funktionen, um die optischen Eigenschaften eines Objekts zu bearbeiten. Diese Funktionen sind immer ähnlich aufgebaut und besitzen meist drei Argumente: Das Objekt, einen oder mehrere

Werte für Eigenschaften (z.B. Farbe oder Radius) und einen Selektor, der angibt, auf welchen Bestandteil oder Zustand eines Grafikobjekts der Style angewendet wird. Mit dem Befehl lv_obj_set_style_bg_color() färbt man beispielsweise den Hintergrund des Sliders ein. Welche Eigenschaften und Bestandteile das jeweilige Grafikobjekt besitzt, kann man in der

LVGL-Dokumentation nachlesen (siehe Link in Kurzinfo). Ein Slider verfügt über drei Bestandteile, die man anpassen kann: MAIN, INDICATOR und KNOB.

Nachdem nun der MAIN-Teil des Sliders eingefärbt und mit einem Rahmen versehen ist, erhält der INDICATOR noch einen Farbverlauf. Hierzu nutzen wir die Funktionen

Listing 2

```
#include <TFT_eSPI.h>
#include <Adafruit_FT6206.h>
#include <lvgl.h>
#define screenWidth 240
#define screenHeight 320

TFT_eSPI tft = TFT_eSPI(screenWidth, screenHeight);
Adafruit_FT6206 touch = Adafruit_FT6206();
static lv_disp_draw_buf_t disp_buf;
static lv_color_t buf[screenWidth * 10];

void my_disp_flush(lv_disp_drv_t *disp, const lv_area_t *area, lv_color_t *color_p) {
    uint32_t w = (area->x2 - area->x1 + 1);
    uint32_t h = (area->y2 - area->y1 + 1);
    tft.startWrite();
    tft.setAddrWindow(area->x1, area->y1, w, h);
    tft.pushColors((uint16_t*)&color_p->full, w * h, true);
    tft.endWrite();
    lv_disp_flush_ready(disp);
}

void my_touchpad_read(lv_indev_drv_t *drv, lv_indev_data_t*data) {
    TS_Point touchPoint = touch.getPoint();
    if(touchPoint.x > 0 || touchPoint.y > 0) {
        data->state = LV_INDEV_STATE_PR;
        data->point.x = 240 - touchPoint.x;
        data->point.y = 320 - touchPoint.y;
    }
    else {
        data->state = LV_INDEV_STATE_REL;
    }
}

void setup() {
    lv_init();
    tft.begin();
    Wire.begin(19, 22);
    touch.begin();

    lv_disp_draw_buf_init(&disp_buf, buf, NULL, screenWidth * 10);
    static lv_disp_drv_t disp_drv;
    lv_disp_drv_init(&disp_drv);
    disp_drv.hor_res = screenWidth;
    disp_drv.ver_res = screenHeight;
    disp_drv.flush_cb = my_disp_flush;
    disp_drv.draw_buf = &disp_buf;
    lv_disp_drv_register(&disp_drv);

    static lv_indev_drv_t indev_drv;
    lv_indev_drv_init(&indev_drv);
    indev_drv.type = LV_INDEV_TYPE_POINTER;
    indev_drv.read_cb = my_touchpad_read;
    lv_indev_drv_register(&indev_drv);

    lv_obj_t * slider = lv_slider_create(lv_scr_act());
    lv_obj_set_size(slider, 180, 10);
    lv_obj_align(slider, LV_ALIGN_CENTER, 0, 0);
}

void loop() {
    lv_timer_handler();
    delay(5);
}
```

Abfrage mit Events

Oft möchte man den Wert eines Objekts nicht unentwegt abfragen, sondern nur wissen, wann er sich verändert hat. Hierfür sind Events eine elegante Lösung. Man kann sie z.B. auslösen, indem man einen Knopf drückt, auf einer Seite scrollt oder einen Slider bewegt. Hierzu definiert man einfach eine Callback-Funktion, die das Event aufrufen soll. Mit `lv_obj_add_event()` verknüpft man diese Callback-Funktion mit dem Objekt, wie in dem folgenden Beispiel:

```
//Callback-Funktion
static void bl_slider_event_cb(lv_event_t * e){
    lv_obj_t * slider = lv_event_get_target(e);
    backlight_brightness = lv_slider_get_value(slider);
}

//Setup
lv_obj_add_event_cb(slider,
                    bl_slider_event_cb,
                    LV_EVENT_VALUE_CHANGED, NULL);
```

Ändert sich der Wert des Sliders, wird die Callback-Funktion aufgerufen und die Hintergrundbeleuchtung auf den Wert des Sliders gesetzt. Das erspart, den Slider ständig im Loop abzufragen.

`lv_obj_set_style_bg_grad_color()` und `lv_obj_set_style_bg_grad_dir()`, um Farbe und Richtung des Verlaufs anzugeben. Als Untergrund für den Slider erstellen wir mit `lv_obj_create()` ein Rechteck, das ich im Folgenden als Box bezeichne. Mit den Styles runden wir die Ecken dieser Box ab, versehen ihre Fläche mit einem Farbverlauf und fügen einen leichten Kontaktschatten hinzu.

Damit der Slider vor der Box erscheint, genügt es, die Box als Erstes zu definieren. Im Codebeispiel habe ich die Elemente zusätzlich ineinander verschachtelt, indem ich beim Erstellen des Sliders mit `lv_slider_create(box)` das Box-Objekt als Argument übergeben habe. Der Slider wird dadurch zu einem Teil des Box-Objekts und bewegt sich fortan mit dieser mit.

Eine Grafik einfügen

Für die Glühbirnen-Steuerung fehlt uns jetzt nur noch ein passendes Symbol, das über dem Slider erscheint. In LVGL lässt

sich ein Bild-Objekt mit `lv_image_create()` erstellen. Als Datenquelle für das Bild kann man entweder eine Datei auf einer SD-Karte ablegen oder als Variable direkt im Code definieren. Einige der gängigsten Symbole sind sogar direkt in LVGL eingebaut. Das umfasst unter anderem Icons für eine Batterie- oder Lautstärkeanzeige und zur Mediensteuerung – jedoch keine Glühbirne. Dafür bietet LVGL einen „Online Image Converter“ an, mit dem man PNG-Dateien in ein Byte-Array umwandeln kann (siehe Link in Kurzinfo). Die exportierte C-Datei kopieren wir ohne Änderungen in den Ordner des Arduino-Sketches. Um das Array nutzbar zu machen,

Bild 5: Auch die Open-Source-Fernbedienung OMOTE funktioniert mit LVGL.



Bild 4: So sieht die UI schon ziemlich professionell aus.

muss man im Sketch selbst lediglich die Funktion `LV_IMAGE_DECLARE()` mit dem Namen des Bildes aufrufen. Mit `lv_img_set_src()` legt man dann das Array als Quelle für das Bild-Objekt fest. Damit das Symbol in der passenden Farbe erscheint, kann man es nachträglich sogar noch mit der Funktion `lv_obj_set_style_img_recolor()` einfärben.

Mit der hinzugefügten Grafik und dem Slider, der sich mit dem Finger bewegen lässt, haben wir in wenigen Schritten eine vorzeigbare Benutzeroberfläche gebaut. Diese kann man jetzt noch beliebig ausbauen, mit weiteren Funktionen versehen und individualisieren.

Layouts am PC entwickeln

Während sich einfache UIs mit der Arduino IDE zügig erstellen lassen, wird das mit zunehmenden

Styles als Vorlage

Anstatt die grafischen Eigenschaften der einzelnen Objekte direkt zu bearbeiten, kann man auch eine `lv_style_t`-Variable definieren. In dieser lassen sich z.B. Farbe und Form einmal festlegen und danach mit `lv_obj_add_style()` auf mehrere Objekte anwenden. Das ist vor allem dann hilfreich, wenn viele Objekte das gleiche Aussehen haben sollen. Mehrere Styles kann man auch in einem Theme zusammenfassen. So lässt sich etwa ein heller und dunkler Darstellungsmodus über verschiedene Objekte hinweg systemweit festlegen.

der Komplexität immer mühsamer – auch weil jede Änderung Zeit zum Kompilieren und Flashen benötigt. Wer diesen Vorgang beschleunigen möchte, kann die UI zuerst auf einem PC entwickeln und später auf den Mikrocontroller portieren. Zu diesem Zweck bietet LVGL einen vorgefertigten Simulator für Visual Studio auf GitHub an (siehe Link in Kurzinfor). Alle bekannten Funktionen lassen sich hier plattformübergreifend verwenden und man profitiert von einer besseren Code-Vervollständigung und sehr kurzen Kompilierzeiten.

Wer sich mit dem Code gar nicht erst auseinandersetzen möchte, kann eine noch einfachere Alternative nutzen: die Entwicklungsplattform Squareline Studio von den Machern von LVGL. Sie ermöglicht es, UIs per Drag&Drop zu erstellen und ist für den privaten Gebrauch als eingeschränkte, kostenfreie Version verfügbar (siehe Link in Kurzinfor) oder mit erweiterten Möglichkeiten im Abo. Mit Squareline Studio kann man Objekte bzw. Widgets einfach auf dem Bildschirm anordnen und bearbeiten. Selbst Bitmaps lassen sich einfach einfügen und benötigen keine mühevollen Konvertierung. So kann man unmittelbar erkennen, wie das Interface später aussehen wird. Ist man damit zufrieden, generiert die Software den Quellcode in der gewünschten Programmiersprache.

Das klingt erstmal viel einfacher, als den Code selbst zu programmieren und mag bei manchen Projekten schneller zum Ziel führen. Da Squareline Studio jedoch noch in der Entwicklung ist, unterstützt die Software nicht alles, was man auch im Code umsetzen kann. Spätestens, wenn man Hardwarefunktionen benötigt, die sich nicht am PC simulieren

lassen (z.B. Displayhelligkeit oder Infrarot-Kommunikation), muss man den Code manuell ergänzen und sich dafür erst einmal im exportierten Code vom Squareline Studio zurechtfinden.

Praxis-Projekt

Da mich LVGL mit seiner effizienten Programmierung überzeugt hat, habe ich die Bibliothek auch für ein eigenes Projekt verwendet: die Open-Source Fernbedienung OMOTE, mit der man verschiedene Geräte über einen Touchscreen steuern kann. Die Fernbedienung (Bild 5) greift dabei auf die gleiche

Hardware zurück wie das Beispiel in diesem Artikel, nur dass der ESP32 und das Display auf einer eigenen Leiterplatte anstelle eines Breadboards montiert sind. Das Interface ist so aufgebaut, dass man mit Wischgesten zwischen verschiedenen Bildschirmen bzw. Geräteprofilen wechseln kann. In LVGL werden diese mithilfe von Tabview-Objekten abgebildet, bei dem jeder Tab die entsprechenden Tasten für Fernsteher, Tuner oder Smart-Home-Steuerung beinhaltet. Wenn ihr mehr über die OMOTE erfahren wollt, findet ihr den Source-Code und weitere Informationen auf der Projekt-Website (siehe Link in Kurzinfor). —akf

Listing 3

```
lv_obj_set_style_bg_color(lv_scr_act(), lv_color_hex(0x464B55),
                          LV_PART_MAIN);

lv_obj_t* box = lv_obj_create(lv_scr_act());
lv_obj_set_width(box, 200);
lv_obj_set_height(box, 280);
lv_obj_set_align(box, LV_ALIGN_CENTER);
lv_obj_set_style_radius(box, 25, LV_PART_MAIN);
lv_obj_set_style_bg_color(box, lv_color_hex(0x646464), LV_PART_MAIN);
lv_obj_set_style_bg_grad_color(box, lv_color_hex(0x3C414B), LV_PART_MAIN);
lv_obj_set_style_bg_grad_dir(box, LV_GRAD_DIR_VER, LV_PART_MAIN);
lv_obj_set_style_border_color(box, lv_color_hex(0x2D323C), LV_PART_MAIN);
lv_obj_set_style_shadow_width(box, 40, LV_PART_MAIN);
lv_obj_set_style_shadow_spread(box, 0, LV_PART_MAIN);
lv_obj_set_style_shadow_ofs_y(box, 15, LV_PART_MAIN);

lv_obj_t* slider = lv_slider_create(box);
lv_slider_set_value(slider, 75, LV_ANIM_OFF);
lv_obj_set_width(slider, 142);
lv_obj_set_height(slider, 13);
lv_obj_set_x(slider, 0);
lv_obj_set_y(slider, -21);
lv_obj_set_align(slider, LV_ALIGN_BOTTOM_MID);
lv_obj_set_style_bg_color(slider, lv_color_hex(0x0F1215), LV_PART_MAIN);
lv_obj_set_style_outline_color(slider, lv_color_hex(0x1E242D),
                              LV_PART_MAIN);
lv_obj_set_style_outline_width(slider, 6, LV_PART_MAIN);

lv_obj_set_style_bg_color(slider, lv_color_hex(0xC5C5C5),
                          LV_PART_INDICATOR);
lv_obj_set_style_bg_grad_color(slider, lv_color_hex(0xE9BB00),
                              LV_PART_INDICATOR);
lv_obj_set_style_bg_grad_dir(slider, LV_GRAD_DIR_HOR, LV_PART_INDICATOR);

lv_obj_set_style_bg_color(slider, lv_color_hex(0x5F6464), LV_PART_KNOB);
lv_obj_set_style_bg_grad_color(slider, lv_color_hex(0x3C424B),
                              LV_PART_KNOB);
lv_obj_set_style_bg_grad_dir(slider, LV_GRAD_DIR_VER, LV_PART_KNOB);
lv_obj_set_style_outline_color(slider, lv_color_hex(0x373C46),
                              LV_PART_KNOB);
lv_obj_set_style_outline_width(slider, 2, LV_PART_KNOB);
lv_obj_set_style_shadow_color(slider, lv_color_hex(0x000000),
                              LV_PART_KNOB);
lv_obj_set_style_shadow_width(slider, 10, LV_PART_KNOB);
lv_obj_set_style_shadow_spread(slider, 3, LV_PART_KNOB);

LV_IMG_DECLARE(lightbulb);
lv_obj_t* bulbIcon = lv_img_create(box);
lv_img_set_src(bulbIcon, &lightbulb);
lv_obj_set_style_img_recolor(bulbIcon, lv_color_hex(0x1E242D),
                             LV_PART_MAIN);
lv_obj_set_style_img_recolor_opa(bulbIcon, LV_OPA_COVER, LV_PART_MAIN);
lv_obj_align(bulbIcon, LV_ALIGN_CENTER, 0, -22);
```

Schneller mit diesem Trick

Damit gerade die Wischgesten flüssig laufen, wendet die OMOTE ein paar Tricks zur Optimierung an: Normalerweise besitzt LVGL einen einzelnen Buffer, in den zuerst die Pixel für die UI geschrieben und danach an das Display gesendet werden. Die beiden Schritte lassen sich durch Double-Buffering aber parallelisieren. So kann gleichzeitig ein Buffer beschrieben werden, während im Hintergrund der zweite Buffer per Direct-Memory-Access an den Bildschirm übertragen wird. Um die Bildrate zu verbessern, habe ich zusätzlich die SPI-Taktfrequenz erhöht. Der ESP32 unterstützt hier bis zu 80 Mhz.

Löten mit einer Hand

LötKolben, LötZinn und Bauteil. Zum Löten braucht man mindestens drei Hände. Mit diesem Tool kann man mit einer Hand löten und LötZinn nachreichen.

von Justin Atkins



Dieses Gerät ist ein LötKolbenaufsatz, der die Extrusion von Lötzinn durch ein eingebautes System ermöglicht, sodass man beim Löten die zweite Hand frei hat.

Es handelt sich um ein Open-Source-Projekt, das mit zwei Berührungssensoren für das Ausgeben von Lötzinn arbeitet. Es basiert auf einem Mikrocontroller ESP32-S3. Die Bewegung des Lötzinns durch eine 3D-gedruckte Röhre wird von einem NEMA17-Schrittmotor gesteuert, der von einem A4988-Schrittmotor-Treiber angetrieben wird.

Der Schrittmotor-Treiber unterstützt eine Stromversorgung über USB-C, kann aber auch über den Wechselstrom des LötKolbens mit Hilfe eines integrierten 5-V-DC-Gleichrichters mit Strom versorgt werden. Außerdem ist er mit einem Boost-Konverter ausgestattet, der die 5V für die Schaltung auf 12V für den Motor erhöht.

Der Mechanismus für die Lötzinnausgabe ist federgestützt wie bei einem 3D-Drucker-Filamentextruder. Das gezackte Rad, das durch die eigene Drehung das Lötzinn nach vorn schiebt, wird mit einer Feder gleichmäßig nach unten gedrückt. Das ermöglicht eine konstante Spannung auf das dünne Zinn, wodurch es immer optimal nachgeliefert wird und man die Zinnrolle leicht ersetzen kann.

Die gesamte Vorrichtung wiegt nur 360 Gramm. Im Gegensatz zu kommerziellen Lösungen ist das Rohr des Lötzinnextruders flexibel und leitet das Lötzinn leicht an der LötKolbenspitze vorbei. Dadurch kann mit einem leichten Druck auf die flexible Extruderröhre noch feingranularer bestimmt werden, wann wie viel Lötzinn an die Spitze kommt.

Kurzinfo

- » Löten mit nur einer Hand
- » Open-Source-Konstruktion zum Nachbauen
- » Genauer löten als mit kommerziellen Lösungen

Checkliste



Zeitaufwand:
12 Stunden



Kosten:
140 Euro

Werkzeug

- » LötKolben
- » Schraubenzieher
- » 3D-Drucker

Material

- » [Materialliste online verlinkt](#)

Alles zum Artikel im Web unter make-magazin.de/x4vv

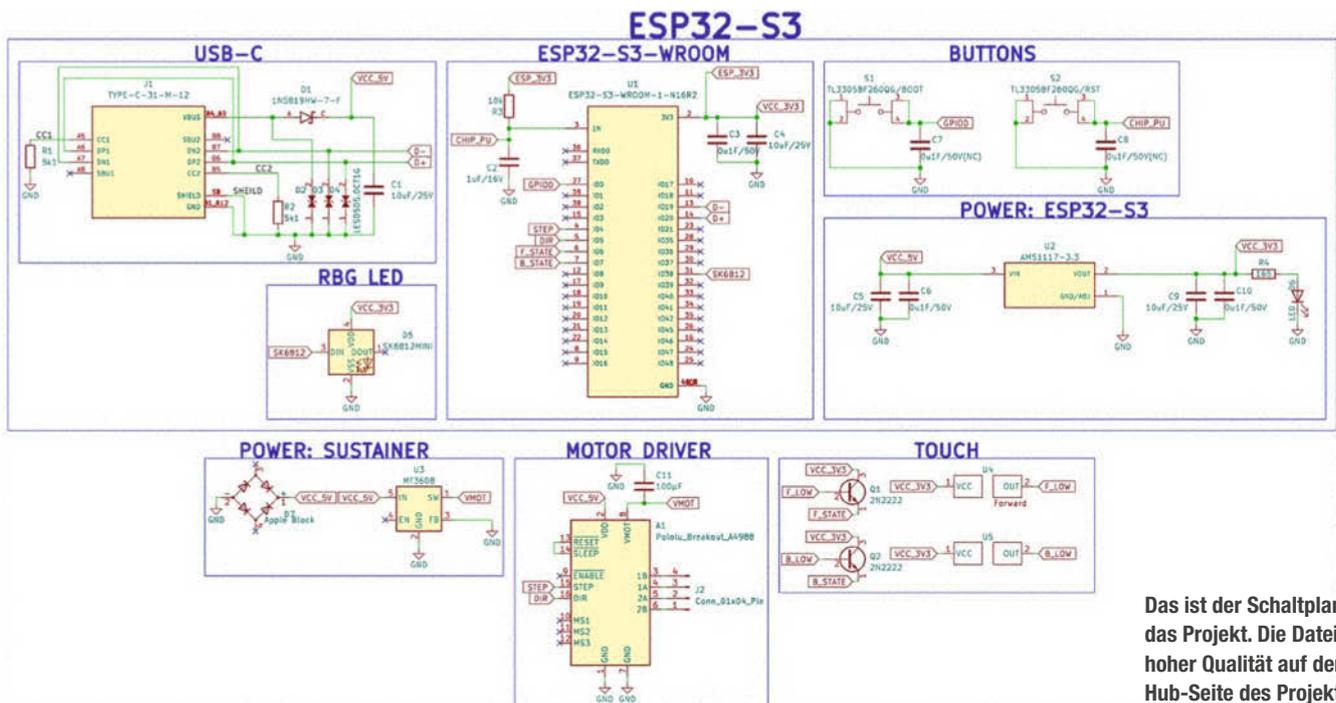
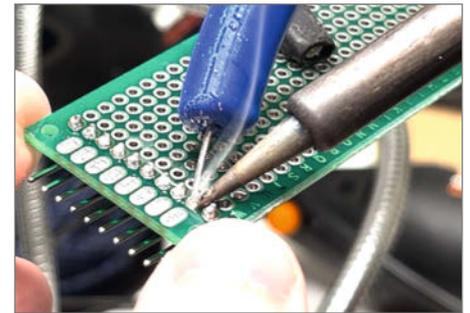


Mehr zum Thema

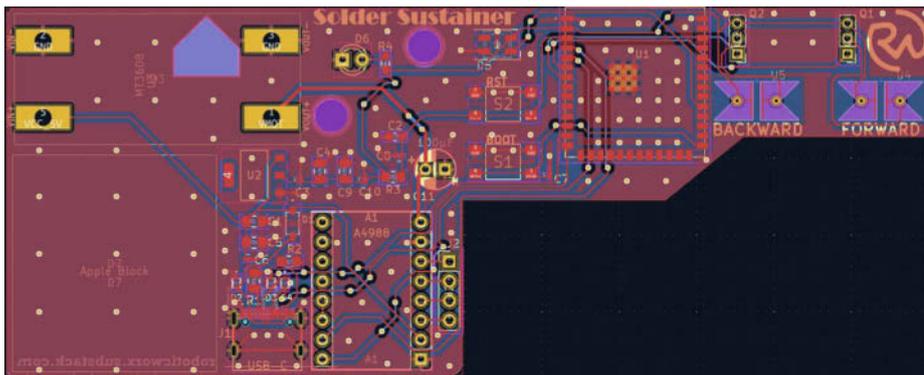
- » Johannes Börsen, Wie man richtig SMD-Bauteile verlötet, YouTube-Video
- » Florian Schäffer, Speichertasten für Weller-Lötstation nachrüsten, Make 6/23, S. 48

Kommerzielle Versionen von Einhand-LötVorrichtungen (beispielsweise das FX8803 von Hakko) sind oft so geformt, dass der LötKolben nach unten gerichtet ist. Das mag zwar nützlich sein, wenn man Drähte an die Stifte eines

Das Extruderröhrchen bringt das Lötzinn an die Spitze des LötKolbens und kann mit einem kleinen Druck auf das Plastik an die Lötspitze gehalten werden.



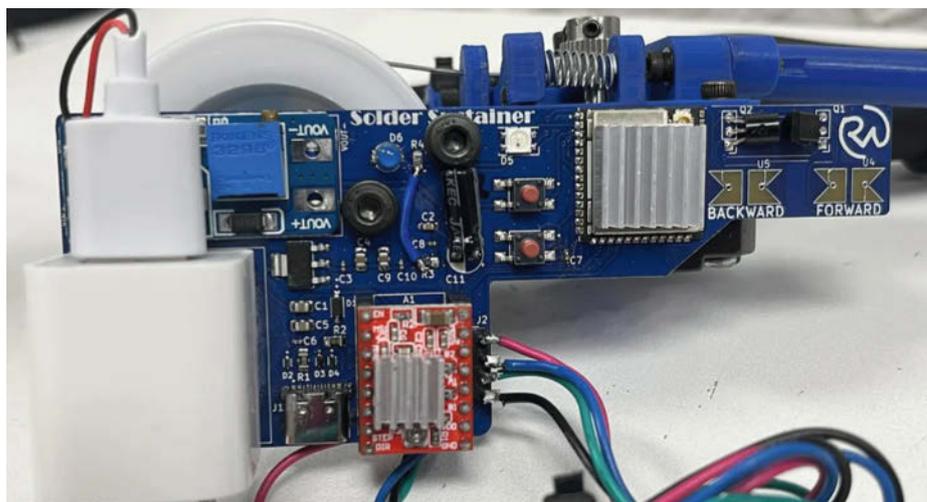
Das ist der Schaltplan für das Projekt. Die Datei ist in hoher Qualität auf der GitHub-Seite des Projektes zu finden (Link in der Kurzinfo).



Bestückungsübersicht. Eine genaue Stückliste bekommt man über den Link in der Kurzinfo.



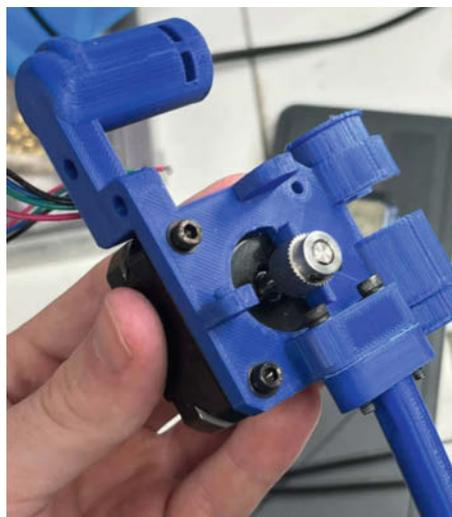
Mithilfe einer solchen Schablone kann Lötpaste über die ganze Platte hinweg in einem Arbeitsschritt aufgetragen werden.



Die Vorlagen für die eigens erstellte Leiterplatte gibt es ebenfalls zum Download.

Motors lötet, aber es ermöglicht kein Löten auf einer ebenen Fläche.

Wenn Sie eine eigene Version dieses Geräts bauen wollen, ist das kein Problem. Das Projekt ist Open Source. Die Links zu



Das gezackte Extruderzahnrad schiebt später das Lötinn nach vorn.

den Projektseiten finden Sie über die Kurzinfo. Dort sind auch Videos von dem Gerät in Aktion verlinkt.

Obwohl es sich bisher nur um einen Prototyp handelt, ist er für mich sehr nützlich und ich habe bereits viele Stunden damit gelötet.

Die Schaltung

Alles in der Box mit der Bezeichnung ESP32-S3 (die obere Hälfte des Schaltplans) ist Teil unserer Mikrocontroller-Schaltung, die den Schrittmotor steuert. Schrittmotoren benötigen sehr spezifische Spannungsimpulse, um sich vorwärts zu bewegen. Ich habe mich in dieser Schaltung für einen Schrittmotor und nicht für einen Gleichstrommotor entschieden, weil er in der Lage ist, sofort zu stoppen.

In Version 2 werde ich wahrscheinlich trotzdem zu einem Gleichstrommotor wechseln, damit ich alles kompakter gestalten kann.

Ich verwende Standard-Keramikkondensatoren zur Oberflächenmontage für die Entkopplungen und habe zwei einpolige SPST-Taster eingebaut, um die BOOT- und RST-Operationen für den Mikrocontroller zu steuern.

Bevor man zum ersten Mal versucht, Code auf den Mikrocontroller zu laden, muss man den ESP32 in den Bootloader-Modus versetzen. Dafür muss man den BOOT-Knopf gedrückt halten, dann den RST-Button einmal drücken und dann den BOOT-Knopf loslassen.

Sonderangefertigte Teile

Für die Technik habe ich eine eigene Leiterplatte designt. Ich empfehle bei einer Bestellung bei einem Produzenten für maßgefertigte Platinen direkt eine Schablone für die Lötpaste mitzubestellen.

Die CAD-Dateien für dieses Projekt sind auf meiner Thingiverse-Seite enthalten. Der Link dafür befindet sich in der Kurzinfo. Um die Teile auszudrucken, braucht man etwa 44 g PLA-Filament, einschließlich Stützstrukturen.

Die Druckeinstellungen, die ich verwendet habe, waren eine 20%ige Füllung bei 0,2 mm Schichthöhe mit einem Rand, um Verformungen zu vermeiden. Diese Einstellungen müssen aber nicht bei allen Druckern optimal sein.

Code

Der Code ist in der Kurzinfo verlinkt. Die betreffende Datei heißt „Solder Sustainer Program“. Wie man diesen auf den ESP kopiert, wird in einem gesonderten Artikel beschrieben, der ebenfalls verlinkt ist.

Die Arduino-Bibliothek AccelStepper ist ebenfalls erforderlich, damit der Code auf meinem GitHub richtig funktioniert.

Zusammenbau

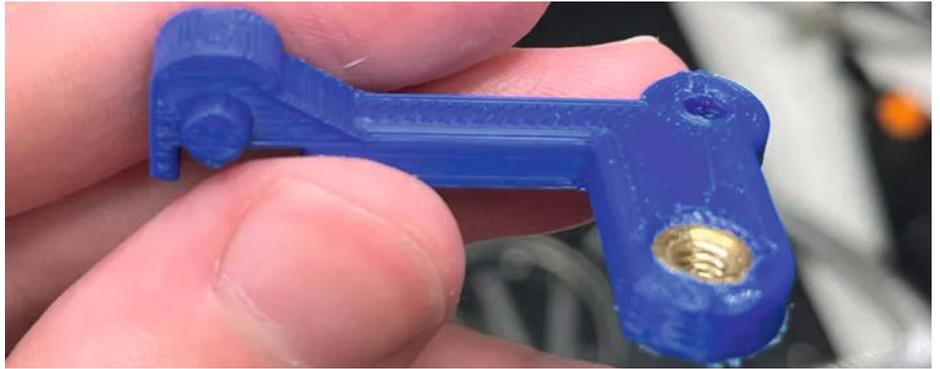
Als Erstes montiert man das Extruderzahnrad für den Motor und schraubt es auf der Motorwelle fest. Anschließend nimmt man zwei M4-Schrauben und schraubt die Lötinnröhre fest an die Motorhalterung.

Als Nächstes passt man einen M4-Gewindeinsatz in den Level-Arm ein. Dafür den Gewindeinsatz auf das Loch legen und mit einem LötKolben erhitzen und so das Plastik schmelzen. Wie in der Materialliste auf meiner Website



Durch diese Rinne im Rad des Kugellagers bleibt das drahtförmige Lötzinn in der Spur.

Dieses Gewinde wird in den Arm hinein geschmolzen.



angegeben, darf dieser Einsatz höchstens 4 mm dick sein, damit er nicht unten herausragt.

Danach nimmt man das Kugellager 4 x 13 x 5 mm und schleift etwas von der Mitte ab. Dieser Schritt ist nicht zwingend, ich würde es aber empfehlen, damit das Lot nicht so leicht verrutscht. Das kann man mit einem Dremel bewerkstelligen.

Jetzt wird eine Feder zwischen den Arm und die Halterung gelegt und das Lager mit einer kurzen M4-Schraube eingeschraubt, die nicht länger als 8 mm sein darf (7 mm, wenn man keine Unterlegscheibe verwendet). Für die Feder kann man eine aus einem alten 3D-Drucker oder einem großen Stift verwenden. Es funktionieren verschiedene Versionen.

Daraufhin werden zwei M3-Muttern durch den Lötzinnrollen-Halter geschoben. So kann man dort später eine Kappe aufschrauben, damit die Lötzinnrolle nicht herunterfällt. Wenn eine Lötzinnrolle einen anderen Durchmesser hat, kann die Halterung in der 3D-Datei angepasst werden.

In die beiden mutterförmigen Löcher neben der Lötzinnhalterung werden zwei M4-Muttern eingesetzt, an denen später von der anderen Seite aus die Leiterplatte befestigt

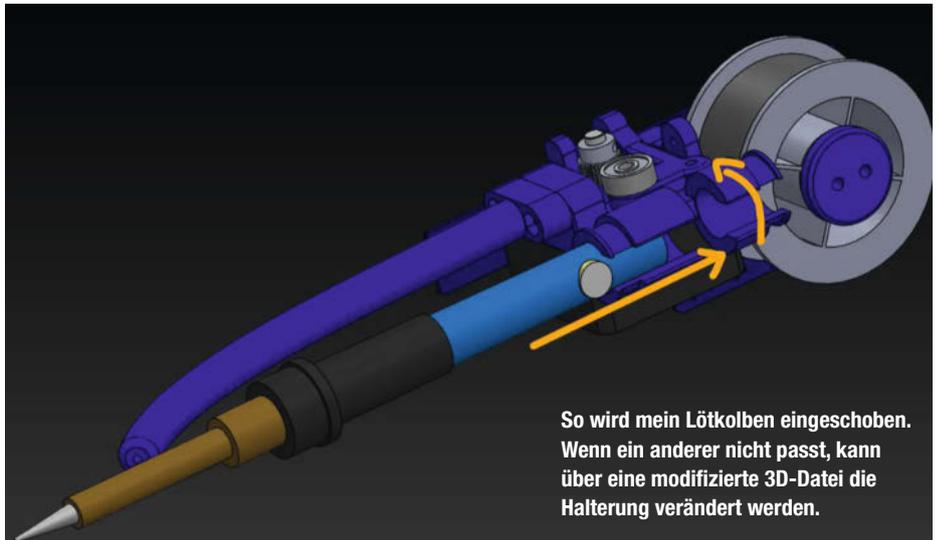
wird. Nachdem die Platine festgeschraubt ist, fehlt nur noch der Lötkolben.

Zum Einsetzen schiebt man ihn einfach von unten ein und dreht ihn, wenn die Nase den Schlitz in der Mitte erreicht.

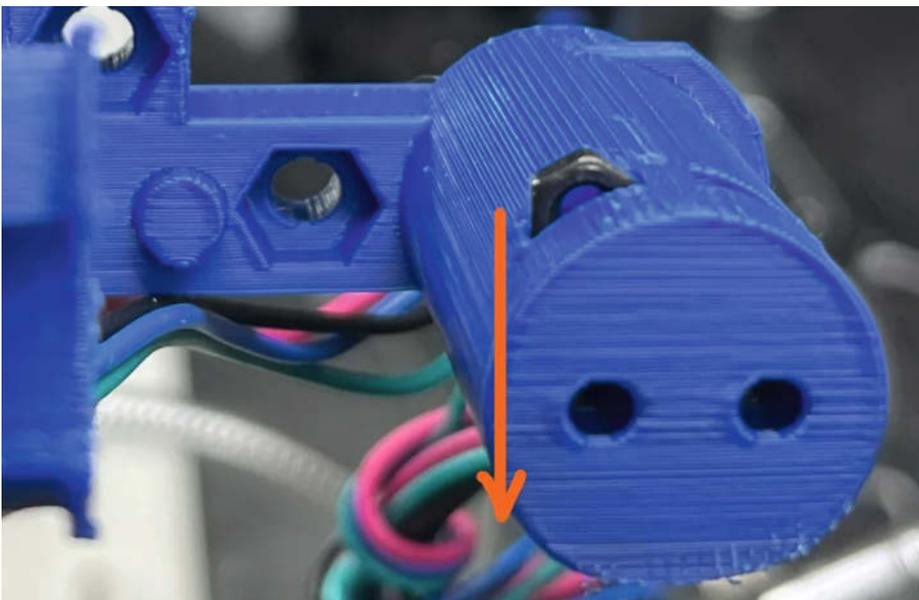
Beachten Sie, dass ich die STEP/SolidWorks-Dateien auf meiner CAD-Seite beigefügt habe,

sodass Sie die Teile bei Bedarf bearbeiten können. Machen Sie sich also keine Sorgen, wenn sie nicht genau die gleiche Ausrüstung haben.

Das war alles! Ich hoffe, dieses Tool ist für Sie genauso nützlich wie für mich und bewahrt Sie vor verbrannten Fingern und herunterfallenden Bauteilen. —das



So wird mein Lötkolben eingeschoben. Wenn ein anderer nicht passt, kann über eine modifizierte 3D-Datei die Halterung verändert werden.



An diesen Muttern wird später von außen ein Stopper für die Lötzinnrolle angeschraubt.



Staubfrei arbeiten mit osVAC neo

Gibt es den universalen Werkstattsauger, der an Elektrowerkzeuge aller erdenklichen Hersteller passt? Natürlich nicht. Aber es gibt ein Adaptersystem zum Selbermachen, das jedes Werkzeug an jeden Saugschlauch bringt. Wir zeigen, wie man den eigenen Maschinenpark mit universellen Steckkupplungen ausstattet.

von Ludo Kerz

Eigentlich könnte alles so einfach sein: Alle einschlägigen Elektro-Werkzeuge haben Anschlüsse zur Staubabsaugung; jeder hat auch einen (Staub)Sauger. Aber in der Praxis ist es die Hölle auf Erden: Nichts passt zu nichts. Also wird fröhlich mit Klebeband improvisiert. Oder der Staub wird einfach aus der Maschine rausgeblasen und setzt sich in jede Ritze der Werkstatt. Im Hobbyhimmel dagegen ist das Problem gelöst: Da gibt es eine einheitliche Steckverbindung für alle Geräte.

Nun, den Hobbyhimmel gibt es tatsächlich: Er ist eine offene Werkstatt in Stuttgart und unter dem Namen osVAC haben sie dort genau das fehlende Verbindungsstück von allem, was Staub ausstößt, zu allem, was Staub saugt, mit Hilfe eines 3D-Druckers realisiert.

Allerdings stellt der Hobbyhimmel von osVAC keine wirklich brauchbaren CAD-Daten zur Verfügung. Um die Anpassung an die eigenen Geräte auf beiden Seiten der Verbindung zu erleichtern, haben wir deshalb unter der Bezeichnung osVAC neo eine Bibliothek von Fusion-360-Dateien entwickelt, die es einem ermöglichen, ohne CAD-Kenntnisse Verbinders zu den eigenen Geräten anzupassen und zu drucken. Und genau darum geht es in diesem Artikel: Wie ihr euch eure individuellen Verbinders für eure Werkstatt mit Hilfe von osVAC neo konfigurieren und drucken könnt.

Unterschiede osVAC und osVAC neo

Selbstverständlich sind osVAC und osVAC neo zu 100 Prozent kompatibel zueinander und von beiden Systemen finden sich im Web auch schon eine Vielzahl von fertigen Verbindern zum Download und zum Drucken, ohne dass man selber die 3D-Modelle anpassen muss. Zusätzlich wurde osVAC neo gegenüber dem originalen osVAC in einzelnen Punkten optimiert:

Kurzinfo

- » Passende Staubsauger-Kupplungen für jeden Saugerschlauch und jeden Werkzeugstutzen
- » Ohne CAD-Vorkenntnisse parametrische Teile in der Gratis-Version von Autodesk Fusion 360 anpassen
- » Adapter selber in 3D drucken

Checkliste

- Zeitaufwand:** rund eine Stunde am Rechner (ohne Druckzeit)
- Kosten:** wenige Euro Materialkosten
- 3D-Druck:** PLA-Teile drucken
- Computer:** mit Windows oder macOS
- Konstruieren:** Parameter anpassen in Fusion 360, keine Vorkenntnisse erforderlich

Mehr zum Thema

- » Gratis-CAD für Maker, Make 4/22, S. 76
- » Video: Spezialadapter im Bewegtbild



Alles zum Artikel im Web unter make-magazin.de/xq1j



- osVAC bietet nur einen Durchmesser von 32 mm für die Steckverbindung. Für Handgeräte sind diese damit aber oft schon recht unhandlich. Andererseits haben aber auch viele stationäre Maschinen Bedarf für größere Durchmesser. osVAC neo kann auf jeden Durchmesser eingestellt werden.
- Die Schlauchverbinder von osVAC neo sind etwas weniger voluminös.
- Die Steckkupplung von osVAC neo kann optional mit einer Gummidichtung ausgestattet werden, um die Dichtigkeit zu erhöhen.

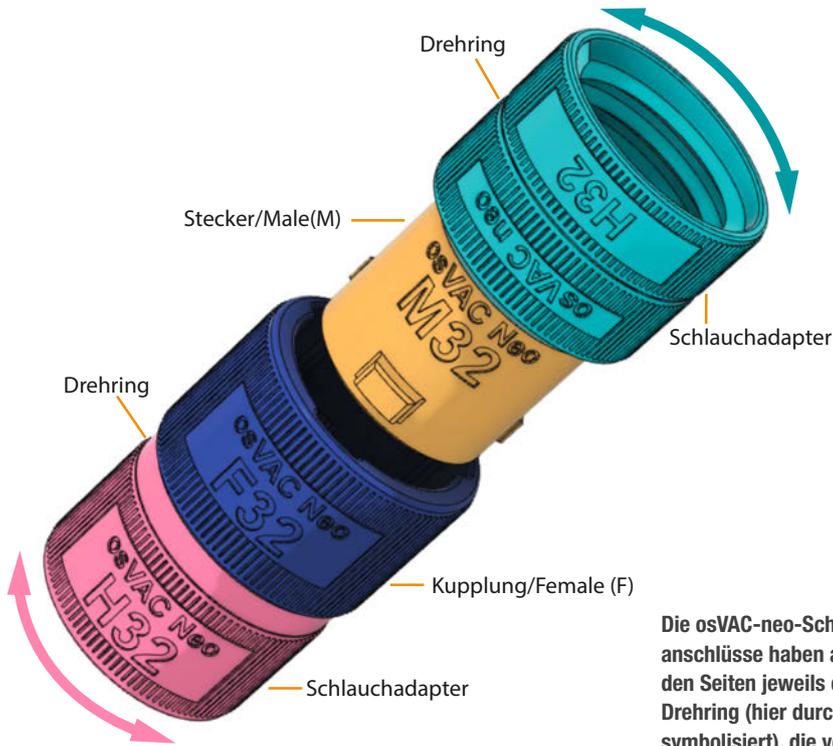
- Der Schlauchadapter wurde enger ausgelegt, um auch hier die Dichtigkeit zu verbessern.
- Im Gegensatz zu osVAC ist für mehr Flexibilität nicht nur die Schlauchkupplung, sondern auch der Schlauchstecker mit einem Drehring ausgestattet.
- Da manche 3D-Drucker Schwierigkeiten mit dem Drehring haben, steht eine geteilte Variante der Schlauchverbinder zur Verfügung.
- Es gibt viele weitere Spezialverbinder.



Staubsaugeranschluss am Rotationsschleifer früher: Mit Klebeband hält es nicht lange und der Schlauch ist starr verbunden.



Mit selbstgedruckten Adaptern ist der Schlauch schnell eingekuppelt und kann dank Drehring flexibel in der Fassung rotieren.



Die osVAC-neo-Schlauchanschlüsse haben an beiden Seiten jeweils einen Drehring (hier durch Pfeile symbolisiert), die verhindern, dass sich der Saugerschlauch bei der Arbeit mit dem Werkzeug störrisch verdreht.

Zur Nomenklatur

Die Grundidee einer Verbindung nach dem osVAC-Prinzip besteht aus einem Stecker und einer Kupplung. Der Stecker wird dabei auch als „male“ (männlich) oder auch einfach als M bezeichnet, die Kupplung als „female“ (weiblich) oder auch einfach als F. Wichtig ist natürlich auch noch der Schlauch-Adapter. Dieser wird auch „hose“ (Schlauch) oder einfach H genannt.

An diese Buchstaben schließt sich jeweils eine Zahl an. Bei M und F beschreibt diese den Innendurchmesser des Steckers in Millimetern. Bei H bedeutet die Zahl den Innendurchmesser des angeschlossenen Schlauchs.

Zum CAD-Werkzeug

Der Hersteller Autodesk stellt Privatanwendern eine etwas eingeschränkte Version seines CAD-Programms Fusion 360 kostenlos zur Verfügung. Für unsere Zwecke ist diese voll und ganz ausreichend. Stand der folgenden Anleitung ist Herbst 2023; da Fusion 360 stetig weiterentwickelt wird, können sich allerdings zukünftig ein paar Abweichungen ergeben.

Um „Fusion 360 für Privatanwender“ zu installieren, muss man sich auf autodesk.de

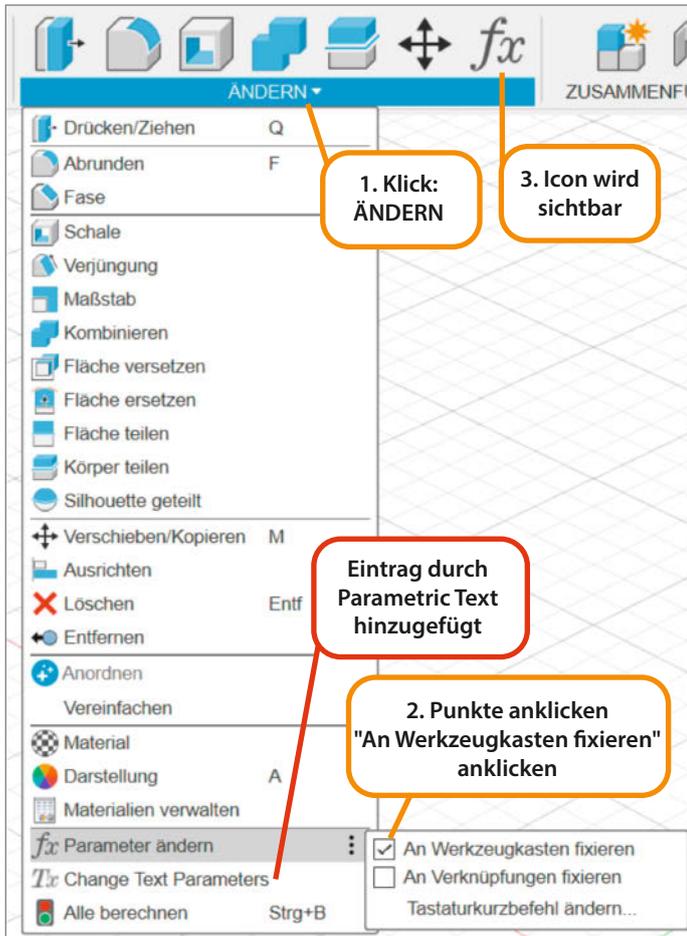


Bild 1

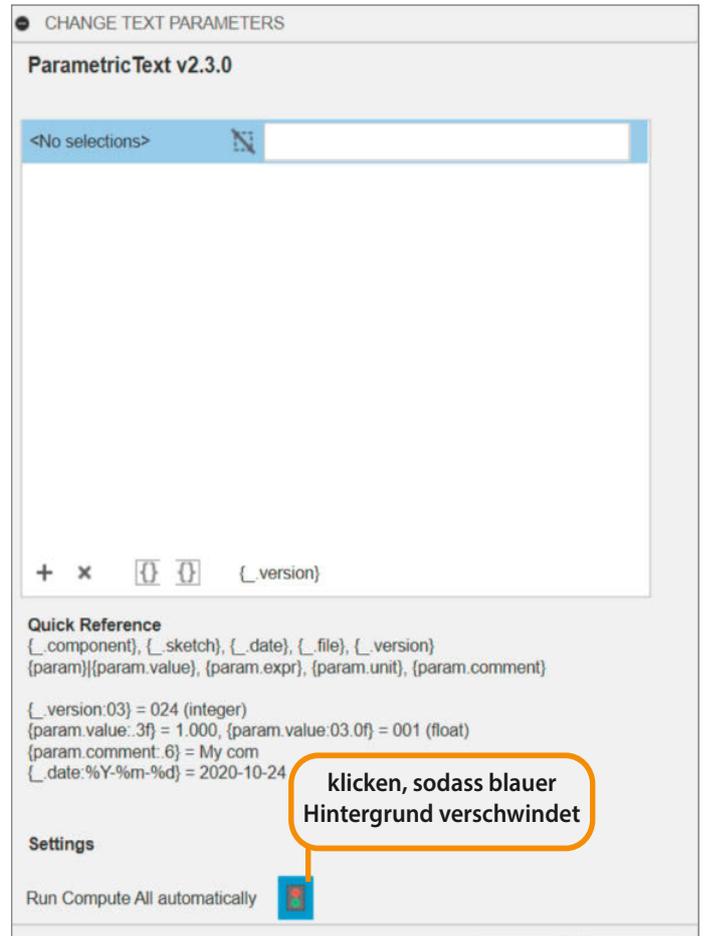


Bild 2

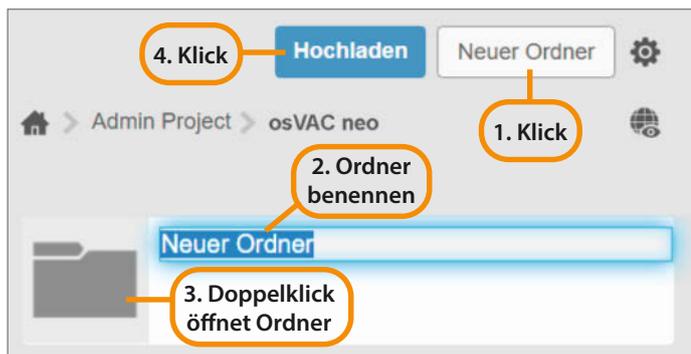


Bild 3

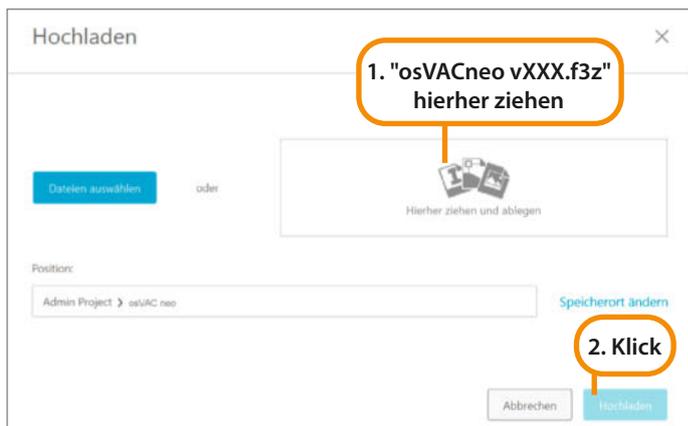


Bild 4

kostenlos registrieren und dann die Software für Windows oder macOS von der Webseite herunterladen (siehe Links in der Kurzinfor). Zusätzlich benötigen wir noch die kostenlose Erweiterung „Parametric Text“ aus dem Autodesk App Store (siehe Link). Diese passt die eingeprägte Beschriftung der Verbindler bei Änderungen automatisch an, sodass jedes Teil stets korrekt beschriftet ist. Nicht irritieren lassen: Um das Add-on zu installieren, beendet man Fusion 360, macht einen Doppelklick auf die Installationsdatei und startet dann Fusion 360 neu.

Nachdem wir Fusion 360 gestartet haben, klicken wir auf „Ändern“ (siehe Bild 1). Erscheint in der Liste ziemlich weit unten der Eintrag „Change Text Parameters“, so ist die Erweiterung korrekt installiert.

Da wir den Menüeintrag „Parameter ändern“ direkt darüber oft verwenden werden, empfiehlt es sich, auf die drei Punkte rechts neben diesem Eintrag zu klicken. In der sich öffnenden Box setzen wir einen Haken bei „An Werkzeugkasten fixieren“. Dadurch erscheint in der Werkzeugleiste oben ein weiteres Icon, über das man Parameter schneller ändern kann.

Als eine weitere Optimierung klicken wir auf „Change Text Parameters“. In dem sich öffnenden Fenster sehen wir unten eine kleine, blau hinterlegte Ampel (Bild2). Hierauf klicken wir einmal, sodass die blaue Hinterlegung verschwindet. Hierdurch werden die häufig zeitintensiven automatischen Neuberechnungen deaktiviert. Wer auf einem leistungsfähigen Rechner unterwegs ist, kann aber durchaus ausprobieren, ob die Maschine in annehmbarer Zeit mit der Automatik klarkommt. Damit sind die Vorbereitungen abgeschlossen und es kann endlich losgehen.

osVAC neo laden

Die benötigte CAD-Datei „osVACneo vXXX.f3z“ gibt es über Link in der Kurzinfor zum Download. Die Datei hat die Endung f3z und entpackt sich später in mehrere Dateien.

Als nächstes legen wir in Fusion 360 in der Seitenleiste links (auch Datenpanel genannt) durch Klick auf die entsprechende Schaltfläche einen neuen Ordner an (Bild 3). Den neuen Ordner nennen wir „osVAC neo“. Durch Klicken auf das Ordnersymbol links gelangen wir in den noch leeren Ordner. Abschließend wählen wir „Hochladen“.

Wir ziehen die Datei „osVACneo vXXX.f3z“ auf das sich öffnende Fenster und klicken nochmals auf „Hochladen“ (Bild 4).

Links im Fenster von Fusion 360 können wir nun beobachten, wie eine Reihe von Dateien entpackt werden, der Fortschritt wird in einem Balken am unteren Rand angezeigt – das ganze kann etwas dauern. Dass die

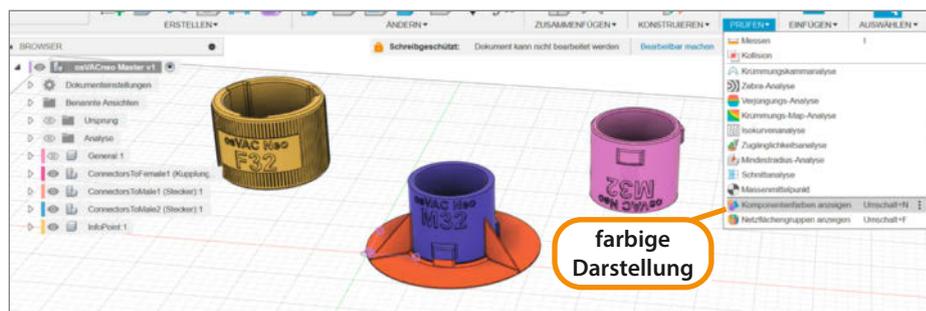


Bild 5

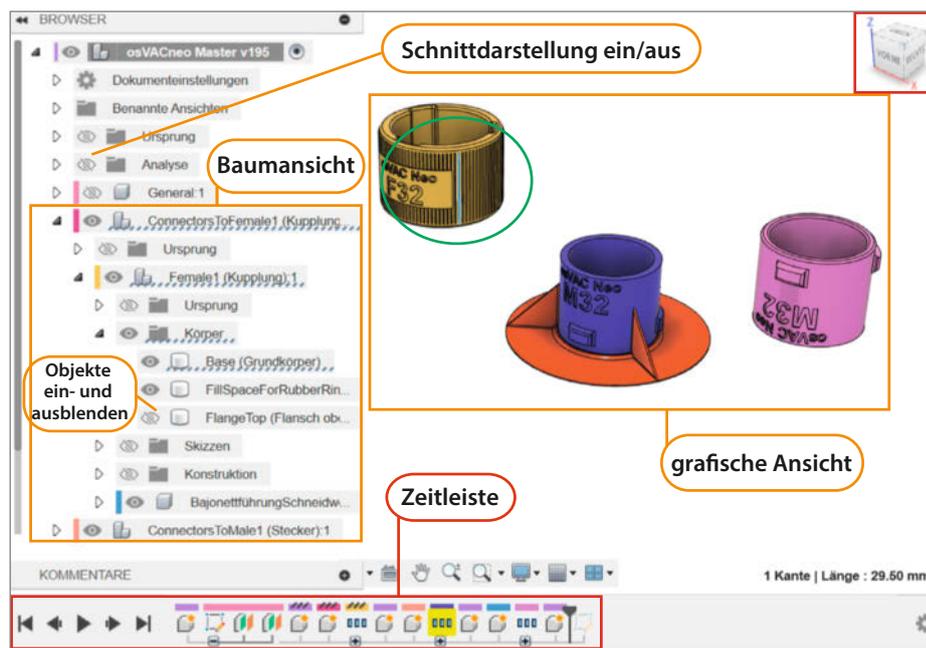


Bild 6

Dateien als schreibgeschützt angezeigt werden, braucht uns nicht zu irritieren, denn bei Änderungen wird der Schreibschutz automatisch aufgehoben und falls nicht, kann man mit einem Mausklick den Status auf „Bearbeitbar“ umschalten. In der kostenlosen Version von Fusion 360 ist die Zahl der gleich-

zeitig bearbeitbaren Dateien auf zehn beschränkt.

Erste Schritte mit Fusion 360

Hierzu öffnen wir zunächst die Datei „osVAC neo Master“ mit einem Doppelklick. Alle Kom-

ponenten werden in einem Einheits-Grau angezeigt. Wer das nicht mag, wählt „Prüfen/ Komponentenfarben anzeigen“ aus (Bild 5).

Mit dem Würfel oben rechts dreht und kippt man die 3D-Ansicht der Objekte beliebig. Hierzu klicken wir entweder auf die einzelnen Flächen des Würfels oder auf die Pfeile oder man dreht und kippt den Würfel mit der Maus bei gleichzeitig gedrückter linker Maustaste. Mit dem Mausrad können wir in der grafischen Ansicht beliebig zoomen, mit gedrücktem Mausrad sie verschieben.

Ein Klick auf das Auge vor der „Analyse“ im sogenannten Browser (Bild 6) aktiviert eine Schnittdarstellung. Zur Beurteilung von Details ist das oft sehr hilfreich.

Links sind alle Elemente der grafischen Ansicht in einer Baumstruktur dargestellt. Klicken wir rechts auf ein 3D-Objekt (in Bild 6 durch einen grünen Kreis markiert), so werden die zugehörigen Einträge in der Baumansicht durch gepunktete Unterstreichungen hervorgehoben. Durch Klick auf die Augensymbole können Objekte ein- und ausgeblendet werden. Es lohnt sich, damit mal etwas zu spielen. So gibt es hier z.B. einen normalerweise versteckten Flansch für die Steckkupplung.

Unten gibt es die Zeitleiste, die die Entstehungsgeschichte aller Objekte Schritt für Schritt speichert. Mit den Bedienelementen links können wir dieses nachvollziehen. Durch Klick auf eines der Pluszeichen in der Zeitleiste werden mehr einzelne Detailschritte angezeigt. Sich durch die einzelnen Schritte einer fertigen Konstruktion zu klicken ist übrigens auch eine gute Methode, um sich in die Arbeitsweise mit Fusion 360 einzudenken; wer nur an passenden Adaptern interessiert ist, kann die Zeitleiste aber getrost unten liegen lassen.

Durchmesser ändern

Der voreingestellte Durchmesser der osVAC neo-Adapter ist 32 mm. Gerade für kleinere Elektro-Werkzeuge bietet es sich aber an, diesen auf 25 mm zu ändern und dann auch einen dünneren und damit flexibleren Schlauch zu verwenden. Für größere Staubsauger hingegen sind 32 mm eher zu wenig. Die Durchmesser der osVAC-neo-Adapter werden zentral in der Datei „osVAC neo Master“ verwaltet und von den anderen Dateien übernommen. Gängige Durchmesser der osVAC neo-Adapter sind 25, 32, 36, 40 und 50 Millimeter. Versuchsweise haben wir auch schon problemlos Varianten mit 80 Millimetern erstellt.

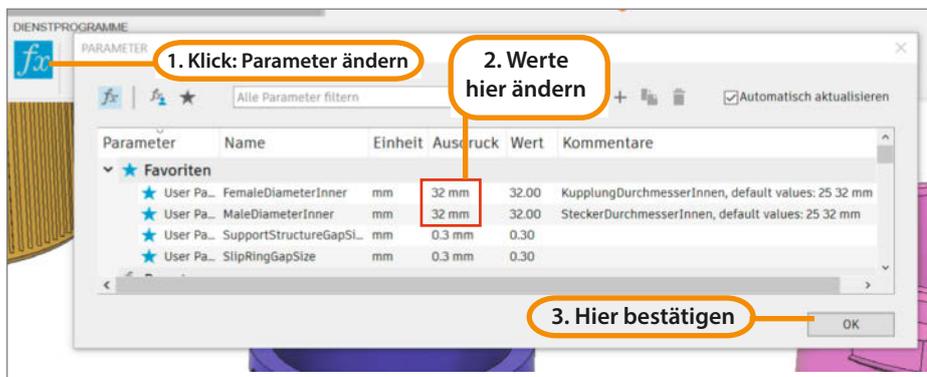


Bild 7

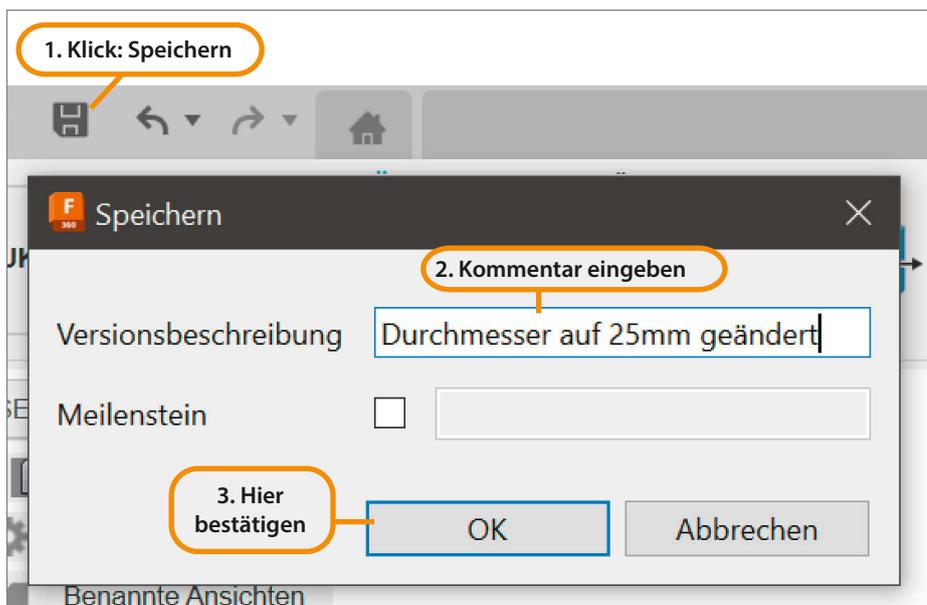


Bild 8

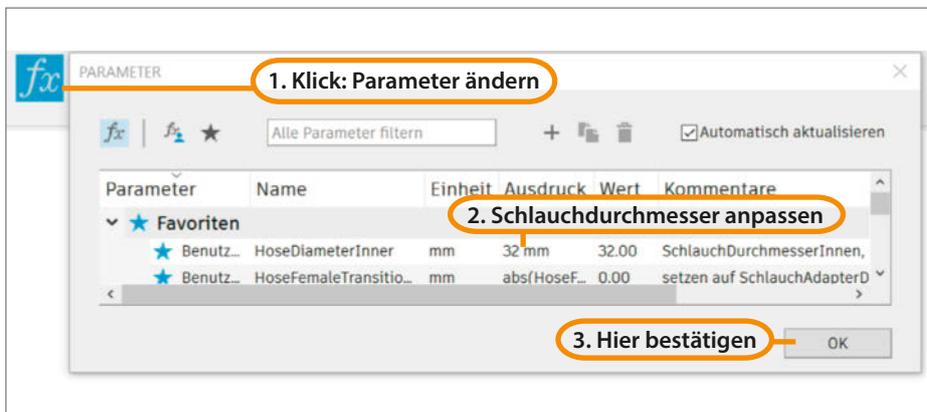


Bild 10



Bild 9

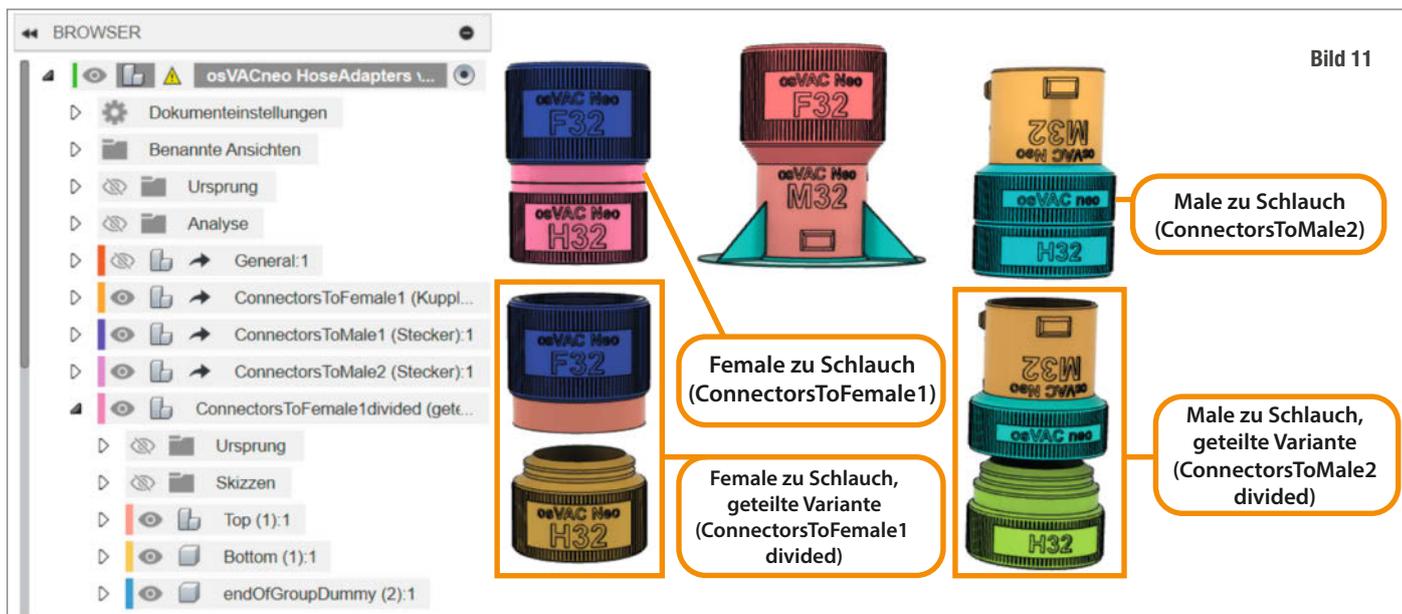


Bild 11

Um die Durchmesser zu ändern, klicken wir in der Datei „osVAC neo Master“ auf „Parameter ändern“. Es klappt ein Fenster mit einer Parameterliste auf. Falls Fusion den Haken bei „Automatisch aktualisieren“ rechts oben automatisch setzt, lohnt es sich, den wegzuklicken, bevor man die Werte FemaleDiameterInner und MaleDiameterInner z.B. auf 25 mm setzt, sonst muss man nach jeder Änderung auf die Aktualisierung des Programms warten. Stattdessen ändert man alles, was man ändern will und bestätigt dann mit OK (Bild 7). Achtung: Je nach Hardware ist Fusion 360 bei der Bearbeitung der CAD-Daten nach Parameteränderungen schon mal eine Minute beschäftigt, bis es eine Reaktion zeigt. Es ist gelegentlich etwas Geduld gefordert!

In der grafischen Ansicht sieht man, dass sich die Form der Adapter geändert hat. Die Beschriftung ist jedoch noch unvollständig, sofern, wie eingangs vorgeschlagen, die automatische Aktualisierung dafür deaktiviert wurde (Bild 2). Gelegentlich verhaspelt sich Fusion 360 auch bei den Neuberechnungen und es kommt zu irreführenden Fehlermeldungen. Alles zusammen wird mit Strg+B korrigiert, was das Tastaturkürzel für „Alle [Parameter] berechnen“ ist und auch den letzten Eintrag im „Ändern“-Menü bildet (vergleiche Bild 1).

Möchten wir, dass diese Änderungen wirksam werden, so speichern wir diese noch ab (Bild 8).

Schlauchverbinder erstellen

In der Praxis sind die häufigsten benötigten Verbinder solche für Schläuche, also die Übergänge von Male/Female auf Schraubgewinde für Staubsaugerschläuche (englisch „hose“, Abkürzung H). Der besondere Hit ist, dass alle

Schlauchverbinder bei osVAC neo mit Drehringen ausgestattet sind, um ein verwindungsfreies Arbeiten zu ermöglichen.

Um Schlauchverbinder zu konfigurieren, öffnen wir die Datei „osVAC neo HoseAdapters“ per Doppelklick. Wenn wir zuvor den Durchmesser der osVAC neo-Adapter geändert haben, sehen wir am oberen Fensterrand ein gelbes Dreieck, auf das man klickt, damit die Änderungen übernommen werden (Bild 9).

Auch in dieser Datei ist es eine gute Idee, sich die Baumansicht mal im Detail anzuschauen und die Analyse (Schnittbildansicht) einzuschalten.

Generell gilt, dass bei Schläuchen deren Innendurchmesser angegeben wird. Hat unser

Schlauch einen anderen Durchmesser als in der Datei voreingestellt, so können wir das anpassen, indem wir den Parameter HoseDiameterInner verändern (Bild 10). Nach Änderung der Parameter wieder Strg+B drücken, damit die Beschriftung aktualisiert wird. Gängige (Innen)Durchmesser für Staubsaugerschläuche sind: 27, 32, 36, 38, 40 und 50 Millimeter. Da die Schläuche sehr flexibel sind und alle in ihrem Mantel ein Gewinde ähnlicher Steigung haben, sind keine weiteren Anpassungen an den Schlauchadaptern notwendig.

Normalerweise drückt man bevorzugt die in Bild 11 gezeigten oberen, einteiligen Varianten (ConnectorsToFemale1 und ConnectorsToMale2). Allerdings macht der integrierte

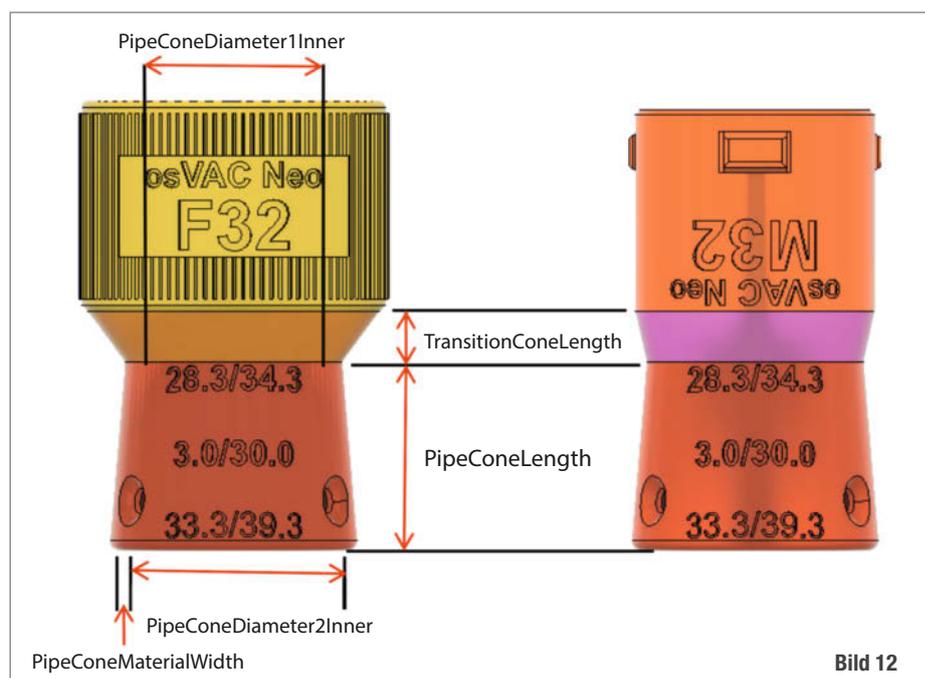


Bild 12

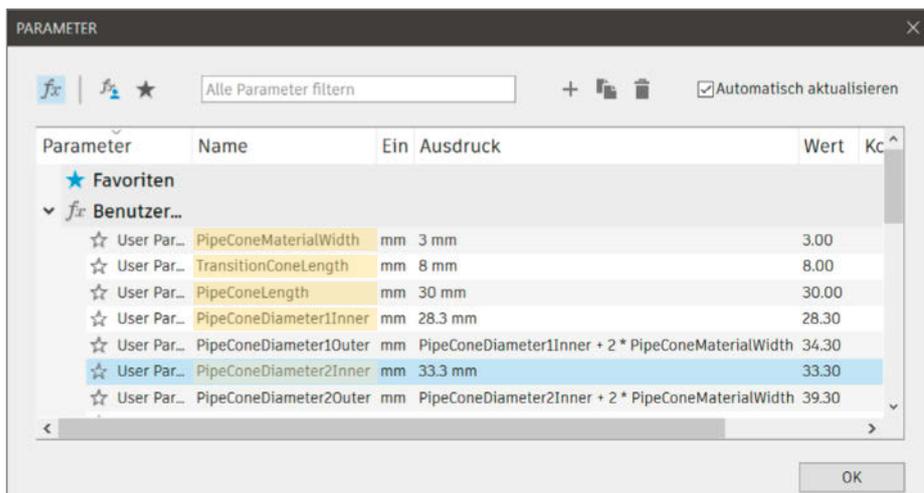


Bild 13

und gleich am richtigen Ort mitgedruckte Drehring auf manchen Druckern Probleme. Daher wurden die geteilten, unteren Varianten geschaffen (ConnectorsToFemale1divided und ConnectorsToMale2divided). Hierbei werden die beiden Hälften getrennt gedruckt. Im Oberteil findet sich ein kleines Loch, durch das man ein Stück Filament einschiebt, um die beiden Hälften zu verbinden.

Verbinder Male/ Female zu Zylinder/Konus

Neben den Schläuchen müssen natürlich auch die Geräte und Sauger angeschlossen werden. Häufig findet man an ihnen zylindrische oder konische Abgänge. Daher werden auch oft Verbinder von Male/Female einerseits auf Zylinder/Konus andererseits benötigt. Diese finden wir neben anderen Verbindern

in der Datei „osVACneo PipesConesFlangesHandles“.

Durch Ändern der in Bild 12 und 13 gezeigten Parameter können die Verbinder beliebig angepasst werden. Verwendet man für PipeConeDiameter1Inner und PipeConeDiameter2Inner (Bild 12) die gleichen Werte, so erhält man einen zylindrischen Adapter, ansonsten einen Konus. Um in der Praxis den Überblick zu behalten, werden die Parameter auf den Adapter eingepreßt (Strg+B zum Aktualisieren drücken).

Export zum 3D-Druck

Klickt man in der Baumstruktur etwas an, so werden die zugehörigen Elemente in der grafischen Darstellung etwas gedimmt dargestellt. Ist es das gewünschte Teil, machen wir einen Rechtsklick in der Baumstruktur und wählen aus dem Kontextmenü „Als Netz speichern“. In dem sich dann öffnenden Fenster haben wir bisher immer als Format „STL (binär)“ ausgewählt, die übrigen Parameter auf Standardeinstellungen belassen und mit OK bestätigt (Bild 14) – das hat sich in der Praxis bewährt. Der folgende Dialog bietet Möglichkeiten, Dateinamen und Pfad anzupassen und die STL-Datei lokal und auch in der Cloud zu speichern.

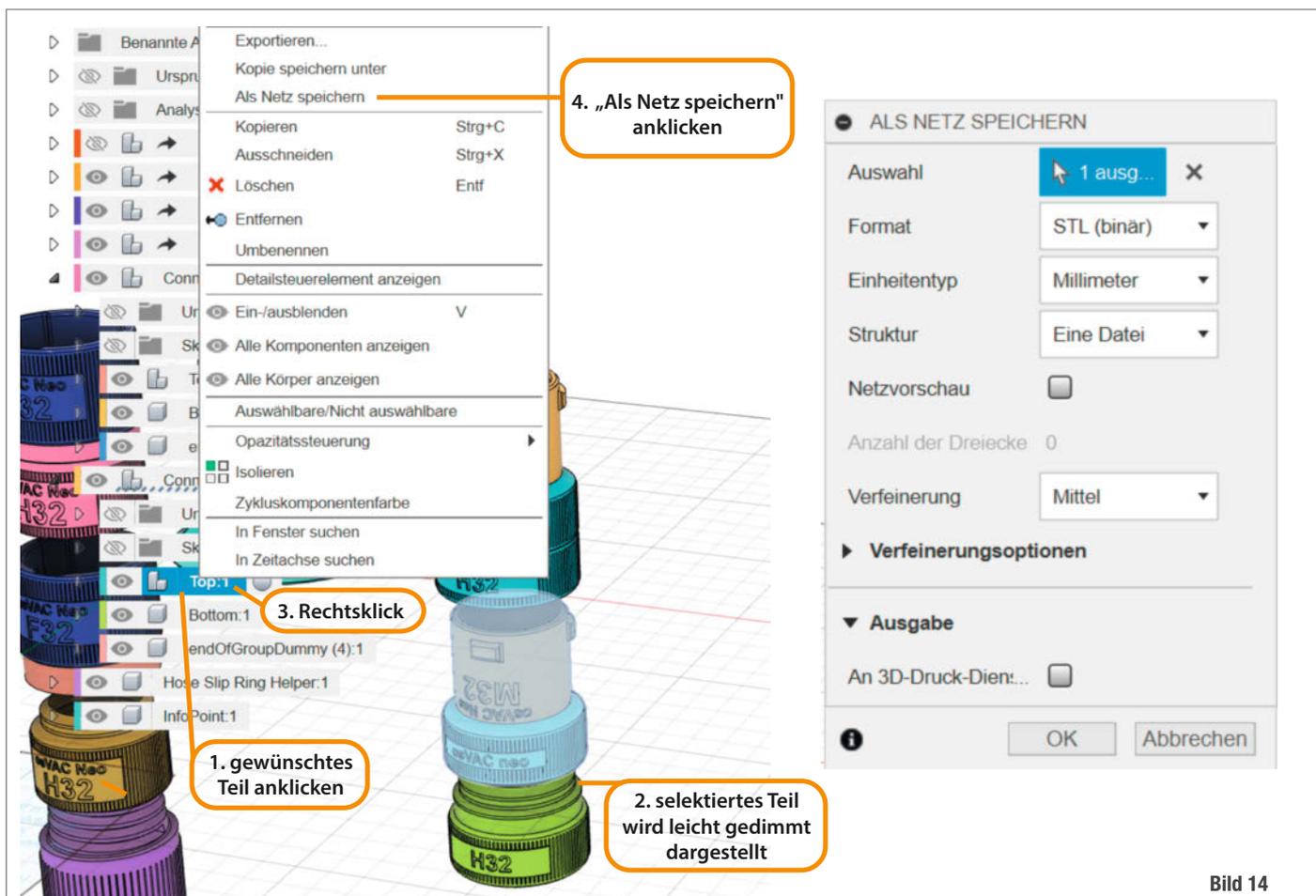


Bild 14

3D-Druck-Parameter

Die Verbinder wurden bereits von vielen Anwendern auf einer breiten Palette von 3D-Druckern erfolgreich gedruckt. Wir hier haben einen Ender-3 V2 im Einsatz, mit dem wir anfänglich mit einer 0,4-mm-Düse und einer Schichtdicke von 0,2 mm gedruckt haben. Vor fast einem Jahr sind wir aber auf eine 0,6er Düse und 0,3 mm Schichtdicke gewechselt und dabei geblieben.

Alle hier besprochenen Objekte können in der Orientierung gedruckt werden, wie sie in der 3D-Darstellung in Fusion 360 zu sehen sind. Da die allermeisten Objekte nur eine kleine Kontaktfläche zum Druckbett haben und manche zudem unten rund sind, empfiehlt es sich, ein kleines Raft (keine Brim!) im Slicer auszuwählen. Support ist bei den meisten Druckern nicht notwendig. Ausnahme sind die geteilten Schlauchverbinder; diese drucken wir meistens mit Support.

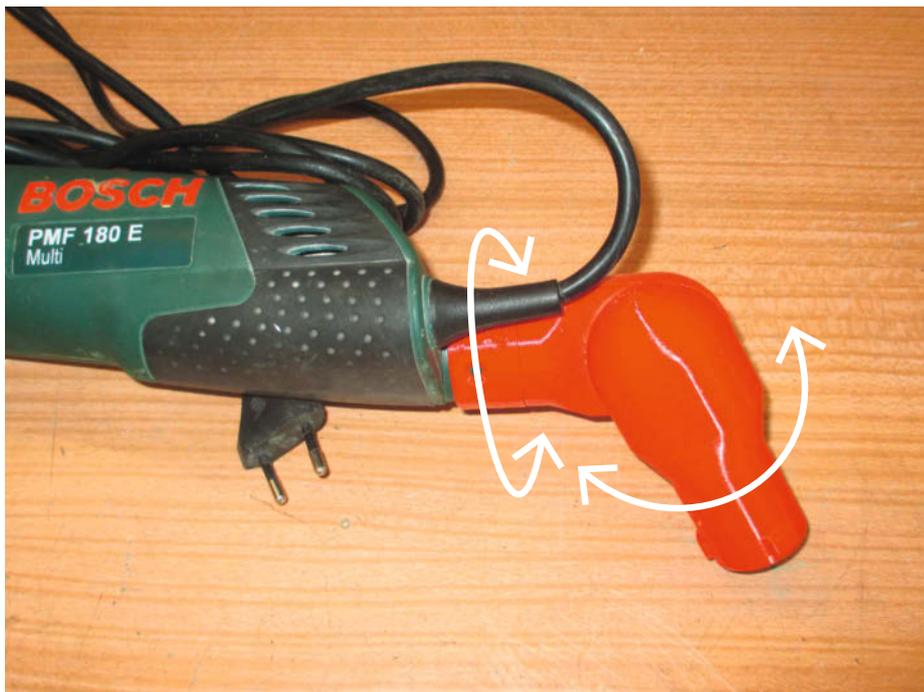
Fazit

Früher war die Absaugung der meisten Elektro-Handgeräte schwierig, da kein Abgang gleich und auch jeder Staubsauger anders ist. Seitdem wir hier unseren privaten Hobbyraum konsequent mit osVAC neo ausgerüstet haben, ist dieses Problem gelöst. Auch für „eben mal schnell“ ist es nicht mehr zu umständlich, eine Absaugung anzuschließen. Die Arbeit mit mehreren Geräten und der Wechsel zwischen ihnen erfordert nur noch ein einfaches Umstecken, der Saugerschlauch passt ja jetzt überall dran. In der Folge saugen wir viel konsequenter ab und folglich ist auch weniger Staub im Raum, der sich überall ablagert.

osVAC neo ist dabei eine große Hilfe, da die Anpassung an viele Szenarien sehr einfach zu realisieren ist. In diesem Artikel wurde auch nur ein Bruchteil der Funktionalität von osVAC neo besprochen, da es sonst den Rahmen des Heftes sprengen würde. So gibt es noch einen Verbinder mit Doppeldrehring und eine Absaugung für Standbohrmaschinen. Hier lohnt sich auch die Erkundung der Dateien auf eigene Faust.

Einige von euch werden jetzt zwar angefixt, aber vielleicht trotzdem enttäuscht sein, da sie einen komplizierteren Adapter bräuchten, den es noch nicht fertig gibt. In dem Fall schaut man zuerst, ob jemand im Netz das Problem schon gelöst hat. Ansonsten hilft dann das Einarbeiten in Fusion 360, um die notwendigen Anpassungen jenseits der hier gezeigten Veränderungen der Parameter vorzunehmen.

In einem vertretbaren Rahmen leisten wir als Initiatoren hier auch gerne Hilfestellung und planen auch, noch weitere Informationen zur Verfügung zu stellen. Leider ist natürlich auch bei uns die Zeit knapp. Wenn sich also jemand so sehr für osVAC neo begeistert,



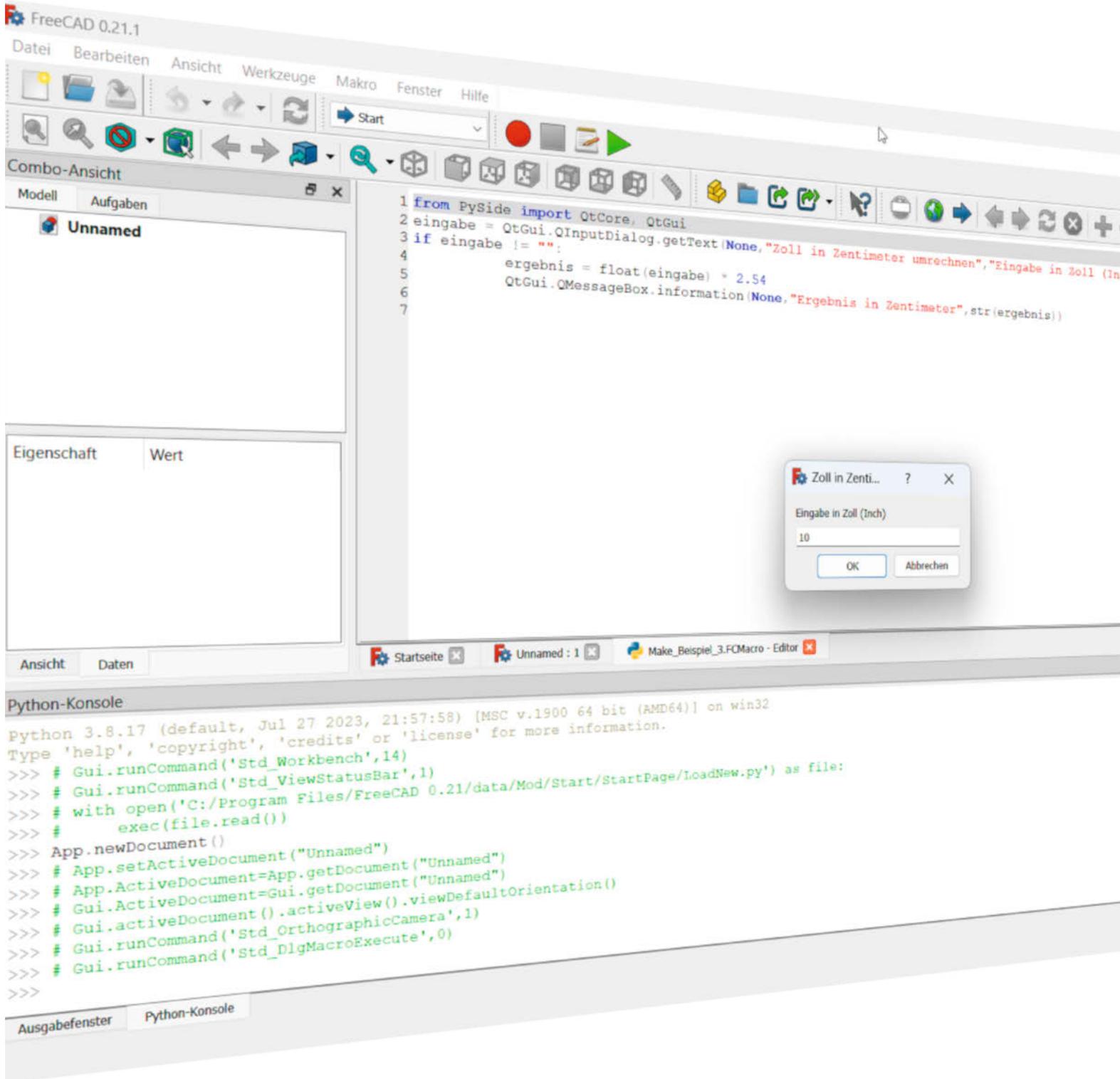
Dieser Spezialadapter verfügt über zwei Drehringe in zwei Ebenen, wodurch sich der Schlauch völlig frei drehen kann (Datei „osVAC neo Special“).

dass sie oder er daran mitarbeiten möchte, freuen wir uns über Kontaktaufnahme über die Make-Redaktion (mail@make-magazin.de). osVAC neo steht unter der Creative-Commons-Lizenz BY-NC-SA 4.0. Das heißt: Auf Basis unserer Arbeit ist es gestattet, genau die Adapter zu konfigurieren, zu verändern und zu drucken,

die ihr gerade benötigt und die STL-Vorlagen dann auch im Netz anderen zur Verfügung zu stellen. Die kommerzielle Nutzung, der Verkauf von Adaptern und Vermarktung des Systems als Produkt auf Basis von osVAC neo ist nicht gestattet. Das ganze soll ein idealistisches Projekt bleiben. —pek



Dieser Aufbau zur Absaugung einer Ständerbohrmaschine war dieses Jahr auf der Maker Faire Hannover in Aktion zu sehen.



Makros für FreeCAD in Python programmieren

FreeCAD lässt sich mit Makros automatisieren und erweitern. Dazu nutzt man die Programmiersprache Python, die sich gut für Einsteiger eignet. Wir zeigen, wie man Makros aufzeichnet, die Programmierung anpasst und erweitert. Außerdem binden wir Makros in Symbolleisten ein, damit sie sich jederzeit einfach starten lassen.

von Matthias Mett



So wie es früher Mode war, Microsoft Excel mit der Skriptsprache Visual Basic for Applications (VBA) zu erweitern, lässt sich das kostenlose Open-Source-Programm FreeCAD mit Makros programmieren und automatisieren. Dazu verwendet FreeCAD die Programmiersprache Python, die eine einfache Syntax verwendet und dadurch auch für Einsteiger gut zugänglich ist. In FreeCAD lassen sich Makros zudem mit einem Makrorekorder aufzeichnen, wodurch das Konstruktionsprogramm auch dann sehr einfach automatisierbar ist, wenn man selber keinen Code schreiben will.

Hello CAD-World

Wenn Sie FreeCAD starten und ein neues Dokument anlegen, können Sie im Menü unter Ansicht/Paneele/Python-Konsole die Programmierkonsole einblenden. Diese erscheint dann im unteren Fensterbereich

Kurzinfo

- » Eigene Makros erstellen und programmieren
- » Makros in Symbolleisten einbauen

Checkliste

- Zeitaufwand:** ein bis zwei Stunden
- Kosten:** keine
- Konstruieren:** CAD, aber keine Vorkenntnisse nötig
- Computer:** Desktop oder Notebook mit Windows, macOS oder Linux

Mehr zum Thema

» Eine verlinkte Liste aller FreeCAD-Artikel aus der Make gibt es online unter der Kurz-URL.

Alles zum Artikel im Web unter make-magazin.de/xp2j

rechts. Wenn sich die Python-Konsole geöffnet hat, sind schon ein paar Programmierbefehle vorhanden, die uns erstmal nicht interessieren. Daher löschen Sie die vorhandenen Inhalte, indem Sie mit der rechten Maustaste innerhalb der Konsole klicken und dann den Menüpunkt „Konsole leeren“ auswählen.

Lassen Sie uns mit einem „Hello World“ beginnen, wie es schon seit den 1970er Jahren in vielen Programmierbüchern als eine der ersten Übungen zu finden ist. Wenn Sie in der Python-Konsole FreeCAD und dann einen Punkt eingeben, erscheint eine Auswahlliste, bei der Sie Console mit den Pfeiltasten auswählen. Wichtig beim Eintippen ist, dass Sie Groß- und Kleinschreibung beachten.

Geben Sie anschließend wieder einen Punkt ein und scrollen Sie wieder mit den Pfeiltasten der Tastatur zu PrintMessage. Wenn Sie den Befehl erreicht haben, erscheint auf der rechten Seite ein Tooltip, der die Syntax erklärt. Nach PrintMessage erwartet Python eine Klammer mit einem String-Objekt. Bestätigen Sie also PrintMessage und geben anschließend eine zu öffnende Klammer ein. Hier kommt nun der Text "Hello World" mit Anführungszeichen hinein und anschließend wieder eine schließende Klammer.

Die Zeile sollte dann so aussehen:

```
FreeCAD.Console.PrintMessage("Hello World")
```

Wenn Sie nun die Eingabetaste betätigen, sehen Sie unter Umständen erstmal nichts. Das liegt daran, dass es in FreeCAD für die Textausgabe eine eigene Konsole gibt. Diese können Sie anzeigen, indem Sie im Menü unter Ansicht/Paneele/Ausgabefenster die Ausgabekonsole einblenden. Dort steht nun mit Ausführungszeitpunkt der Text „Hello World“.

Die Python-Konsole kann immer nur einen Befehl ausführen, der sich im Nachhinein nicht mehr ändern lässt. Wenn Sie den Befehl nochmal ausführen wollen, müssen Sie ihn nochmal eingeben oder kopieren und erneut einfügen.

CAD-Objekte

Python unterstützt eine objektorientierte Programmierung, was bei FreeCAD genutzt wird. Hierbei behandelt Python alles als Objekte, wobei die Programmierobjekte hierarchisch aufgebaut sind. Wenn man das Wurzelobjekt kennt, in unserem Beispiel FreeCAD, dann lassen sich durch die Autovervollständigung die



Autovervollständigung mit PrintMessage und Hilfe im Tooltip



FreeCAD mit eingblendeter Python-Konsole und geöffnetem Ausgabefenster

dazugehörigen untergeordneten Objekte aufrufen. Das Wurzelobjekt kann wiederum Klassen wie Document, Module wie Console und Funktionen wie newDocument enthalten. Diese sind in der Autovervollständigung mit unterschiedlichen Symbolen gekennzeichnet.

Funktionen führen eine Aktion direkt aus. Beispielsweise können wir mit der Funktion newDocument ein neues Dokument erzeugen, dem wir in der darauffolgenden Klammer dessen Namen mitgeben:

```
FreeCAD.newDocument("MeinDokument")
```

Nach der Eingabe auf der Konsole und der Bestätigung mit der Eingabetaste erstellt FreeCAD ein neues Dokument; dieses erscheint im Ansichtenbaum und als Karteireiter im Hauptfenster.

Scrollen Sie die Objekte in der Autovervollständigung einmal durch, viele sind selbsterklärend und enthalten eine kurze Erklärung im Tooltip. Für die weitere Automatisierung machen wir es uns jetzt jedoch einfacher und zeichnen eine Klickfolge mit dem Makrorekorder auf.

Aufnahme läuft

Dazu leeren Sie zuerst die Python-Konsole. Dann wechseln Sie in den Arbeitsbereich

„Part“, um die Symbolleiste mit den Volumenkörpern anzuzeigen. Um die Makroaufzeichnung zu starten, öffnen Sie in der Menüleiste den Menüpunkt „Makro“ und klicken auf den Befehl „Makro aufzeichnen“. Dieser Befehl ist alternativ auch in einer eigenen Symbolleiste (Ansicht/Symbolleisten/Makro) unter einem roten Punkt zu finden.

Starten Sie die Makroaufzeichnung und geben Sie einen beliebigen Namen ein (etwa Make_Beispiel_1) und bei Bedarf auch den Pfad, unter dem die Makro-Datei gespeichert wird. Bestätigen Sie nun den Button „Aufzeichnen“. Erstellen Sie nun einen neuen Würfel durch Klick in die Symbolleiste. Stoppen Sie die Aufzeichnung über das Menü „Makros“ mit dem Button „Aufzeichnung beenden“ oder über die Symbolleiste mittels des grünen Quadrats.

Das aufgezeichnete Makro finden Sie im Menü unter „Makro“ und „Makros...“. Wenn Sie das Makro mit dem zuvor gewählten Namen, etwa „Make_Beispiel_1.FCMacro“ mit der Maus markieren, können Sie es mit dem Button „Ausführen“ nutzen, um den nächsten Würfel zu zeichnen.

Jedenfalls theoretisch, denn wenn Sie es versuchen, bekommen Sie keinen neuen Würfel, sondern im Ausgabefenster eine Fehlermeldung (unicode error). Um dem auf den Grund zu gehen, markieren Sie das Makro wieder unter „Makro/Makros ...“, klicken dann aber auf „Bearbeiten“.

Wenn Sie den Code des Makros mit der Anzeige in der Python-Konsole vergleichen, sind diese fast identisch. Das aufgezeichnete Makro enthält jedoch noch die Befehle import FreeCAD und import Part. Dabei handelt es sich auf Verweise der jeweiligen Bibliotheken, um diese Objekte im Makro verfügbar zu machen. Man könnte aus dem Inhalt der Python-Konsole auch ein Makro erstellen, indem man in der Python-Konsole mit der rechten Maustaste klickt und dann im Kontextmenü auf „Verlauf speichern unter ...“. Hier würden dann jedoch die beiden Importbefehle fehlen.

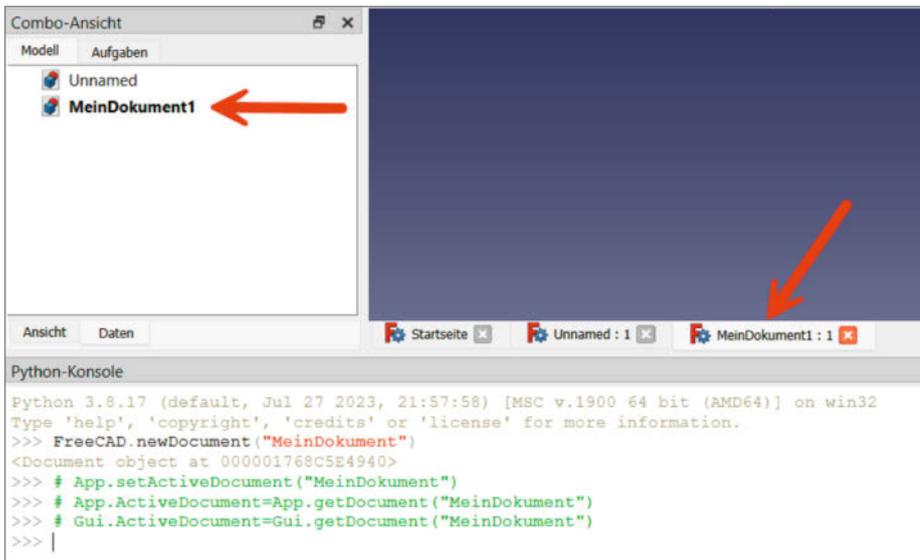
Bei den grün markierten Befehlen mit einer oder mehreren Rauten am Anfang handelt es sich um Kommentare. Diese lässt der Compiler beim Ausführen des Makros aus. Darin befinden sich jedoch oft Hinweise, wofür der nachfolgende Befehl nützlich ist.

Im Wesentlichen besteht das Makro jetzt aus drei Befehlen. Der erste Befehl erstellt den Würfel, wobei der Befehl mit App für Anwendung (Application) beginnt und auf das derzeit aktive Dokument verweist. Hier gibt es die Funktion addObject, die ein Objekt erstellt. In der Klammer ist als erstes angegeben, aus welchem Arbeitsbereich das Objekt stammt, in unserem Fall aus dem Part-Arbeitsbereich. Dieser enthält dann ein Box-Objekt, also einen Würfel:

```
App.ActiveDocument.addObject(
    "Part::Box", "Box")
```

Wenn Sie diesen Befehl jetzt in die Python-Konsole kopieren und ausführen, entsteht ein neues Würfel-Objekt. Damit dieses im Ansichtenbaum besser aufzufinden ist, erhält das Objekt einen neuen Namen mittels fortlaufender Nummerierung. Dies geschieht, indem der Befehl über das aktive Dokument auf das derzeit aktive Objekt verweist und dessen Label ändert.

Im erzeugten Code des Makros fällt allerdings auf, dass der Buchstabe ü in „Würfel“ beim Eintrag für das Label nicht richtig dargestellt wird. Man muss das auf der Ecke stehende Quadrat mit dem Fragezeichen darin hier tatsächlich gegen ein frisch eingetipptes



Mit Makro erzeugtes neues Dokument

ü tauschen (anschließend speichern mit Strg+S), damit nicht wieder die eingangs beobachtete Fehlermeldung auftritt:

```
App.ActiveDocument.ActiveObject
.Label = "Würfel"
```

Zuletzt aktualisiert der dritte Befehl das aktive Dokument, damit alles richtig dargestellt wird:

```
App.ActiveDocument.recompute()
```

Sie können nun im Ansichtenbaum den vorhandenen Würfel mit der rechten Maustaste markieren und über das Kontextmenü löschen. Wenn Sie sich im Makrofenster befinden, können Sie das Makro über das Menü (Makro/Ausführen) oder über die Symbolleiste (grünes Dreieck) ausführen. Auch über „Makro/Makros ...“ können Sie das Makro markieren und mit dem entsprechenden Button starten. Das Makro erstellt nun ein neues Würfel-Objekt mit korrektem Umlaut im Label und hängt eine laufende Nummer an.

Weitermachen

Wenn Sie weitere Eigenschaften ändern wollen, zeichnen Sie einfach ein neues Makro auf. Löschen Sie zuerst wieder den vorhandenen Würfel in der Baumansicht. Starten Sie nun die Makroaufzeichnung und vergeben wieder einen Namen wie „Make_Beispiel_2“. Erstellen Sie einen neuen Würfel und markieren ihn in der Baumansicht. Ändern Sie in den Eigenschaften die Höhe unter „Height“ auf 3,5 und bestätigen Sie mit der Eingabetaste. Beenden Sie anschließend die Makroaufzeichnung.

Hier sind nun zwei Zeilen hinzugekommen, einmal mit 3 mm Höhe und dann mit 3,5 mm Höhe. Die Zeile mit der 3 mm Höhe können Sie einfach löschen. Wenn Sie das Makro erneut ausführen wollen, müssen Sie auch das Sonderzeichen unter Würfel wieder mit einem ü ersetzen, ansonsten erscheint eine Fehlermeldung.

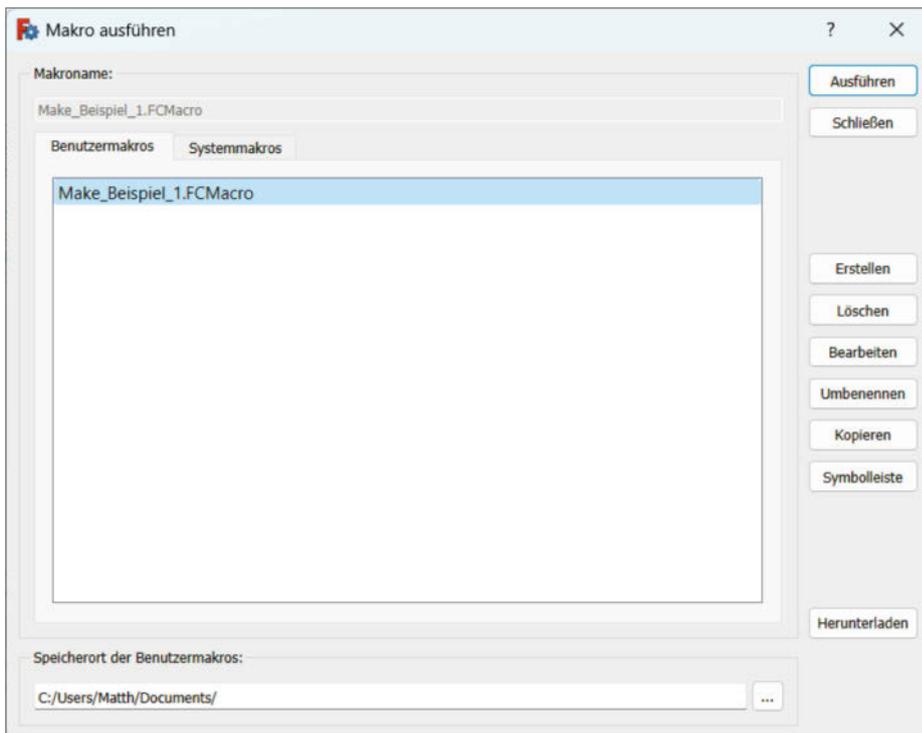
Diesmal holt sich das Makro zuerst das Dokument und dann das Objekt über den Namensverweis und stellt dann die Höhe auf 3,5 mm ein:

```
FreeCAD.getDocument('Unnamed')
.getObject('Box').Height = '3.5 mm'
```

So lässt sich alleine mit der Aufzeichnung von Makros FreeCAD einfach automatisieren.

Zahlen, bitte

Als nächstes Beispiel erstellen wir einen einfachen Rechner mit vier Zeilen Code, der Zoll (Inch) in Zentimeter umrechnet. Beim Starten des Makros wollen wir eine Eingabebox anzeigen, in der man eine Zahl in Zoll eingibt, die man umrechnen möchte. Anschließend soll der Rechner die Zahl nicht im Ausgabefenster anzeigen, sondern in einer MessageBox.



Das Makromenü mit aufgezeichnetem Makro

Der gesamte Code steht im Listing 1, wir gehen ihn aber Zeile für Zeile durch.

Gehen Sie wieder in das Menü „Makro/Makros ...“, wobei Sie den Button „Erstellen“ betätigen und unter dem Namen „Make_Beispiel_3“ eingeben.

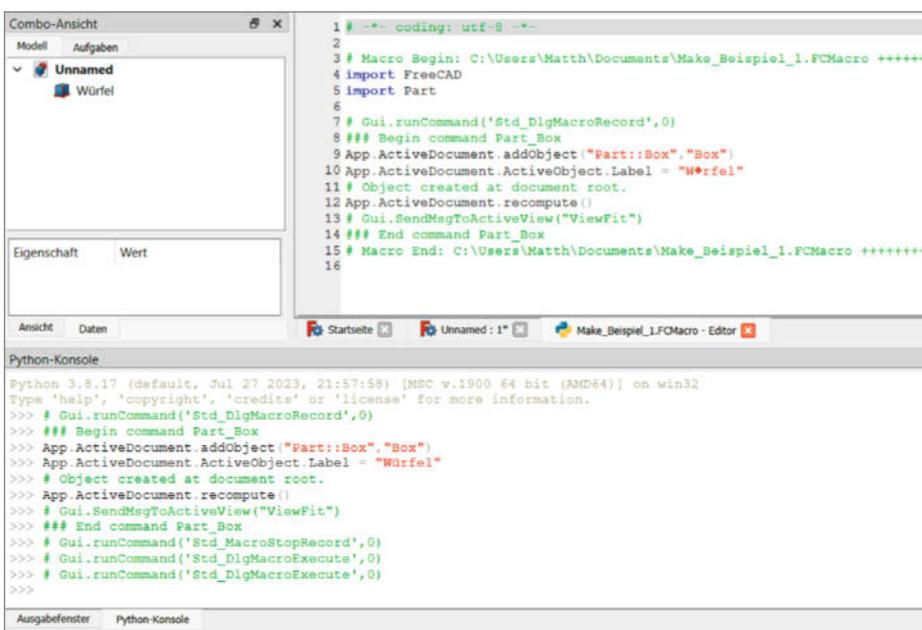
Für die Anzeige der Eingabebox und der MessageBox benötigt man eine Bibliothek, welche diese Objekte enthält. FreeCAD ist in dem Framework Qt programmiert, welches

auch für die Fensterdarstellung zuständig ist. Um diese Objekte nutzen zu können, gibt es eine Bibliothek mit dem Namen PySide, welche die Qt-Objekte in Python nutzbar macht.

Geben Sie daher zuerst folgende Zeile im Makro ein:

```
from PySide import QtCore, QtGui
```

Als nächstes benötigen wir eine Variable, die den eingegebenen Text speichert, damit wir



Vergleich des aufgezeichneten Makros mit der Python-Konsole

```

1 # -*- coding: utf-8 -*-
2
3 # Macro Begin: C:\Users\Matth\Documents\Make_Beispiel_2.FCMacro ++++++
4 import FreeCAD
5 import Part
6
7 # Gui.runCommand('Std_DlgMacroRecord', 0)
8 ### Begin command Part_Box
9 App.ActiveDocument.addObject("Part::Box", "Box")
10 App.ActiveDocument.ActiveObject.Label = "Würfel"
11 # Object created at document root.
12 App.ActiveDocument.recompute()
13 # Gui.SendMsgToActiveView("ViewFit")
14 ### End command Part_Box
15 # Gui.Selection.addSelection('Unnamed', 'Box')
16 FreeCAD.getDocument('Unnamed').getObject('Box').Height = '3 mm'
17
18 FreeCAD.getDocument('Unnamed').getObject('Box').Height = '3.5 mm'
19
20 # Macro End: C:\Users\Matth\Documents\Make_Beispiel_2.FCMacro ++++++
21

```

Aufgezeichnetes Makro mit Würfel und eingestellter Höhe

```

1 # -*- coding: utf-8 -*-
2
3 # Macro Begin: C:\Users\Matth\Documents\Make_Beispiel_2.FCMacro ++++++
4 import FreeCAD
5 import Part
6
7 # Gui.runCommand('Std_DlgMacroRecord', 0)
8 ### Begin command Part_Box
9 App.ActiveDocument.addObject("Part::Box", "Box")
10 App.ActiveDocument.ActiveObject.Label = "Würfel"
11 # Object created at document root.
12 App.ActiveDocument.recompute()
13 # Gui.SendMsgToActiveView("ViewFit")
14 ### End command Part_Box
15 # Gui.Selection.addSelection('Unnamed', 'Box')
16 FreeCAD.getDocument('Unnamed').getObject('Box').Height = '3.5 mm'
17
18 # Macro End: C:\Users\Matth\Documents\Make_Beispiel_2.FCMacro ++++++
19

```

Für die Ausführung korrigiertes Makro

```

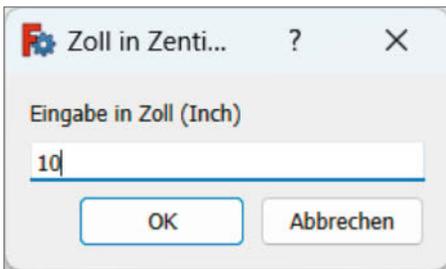
Listing 1

from PySide import QtCore, QtGui
eingabe = QtGui.QInputDialog.getText(None, "Zoll in Zentimeter umrechnen", "Eingabe in Zoll (Inch)")[0]
ergebnis = float(eingabe) * 2.54
QtGui.QMessageBox.information(None, "Ergebnis in Zentimeter", str(ergebnis))

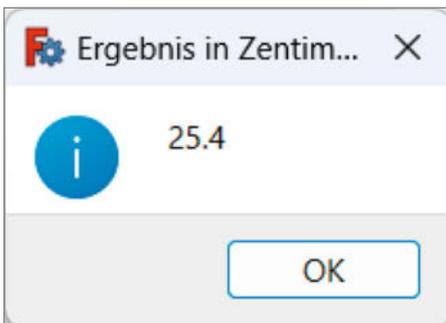
```

ihn für weitere Berechnungen verwenden können. Diese nennen wir einfach `eingabe`. Bei vielen Programmiersprachen gibt es bestimmte Programmierkonventionen, welche den Datentyp vor den Variablenamen vorstellen. Hier brauchen wir uns um den Datentyp zu dem jetzigen Zeitpunkt jedoch noch keine Gedanken zu machen.

Jetzt kommt ein Trick: Führen Sie in der Python-Konsole (nicht im Editor für das Makro)



Eingabedialog des Makros



Messagebox mit Umrechnungsergebnis

den zuvor genannten `import`-Befehl ebenfalls aus. Dann stehen die Bibliotheken in der Konsole zur Verfügung. Nun können Sie dort `QtGui` eintippen und mit einem Punkt die Autovervollständigung aufrufen (das geht im Makro-Editor nämlich nicht). In der Liste finden Sie `QInputDialog`, welches das Eingabefenster darstellt. Da wir die Eingabe in eine Variable holen wollen, gibt es eine Funktion `getText`. Der Funktionsname verrät auch, dass der Inhalt unserer Variablen später ein Text, also ein String sein wird. Dabei ist es egal, ob man Buchstaben oder Zahlen hineinschreibt.

In den Optionen für die Klammer interessieren uns nur die zweite und dritte Stelle, welche der Titel des Fensters und das Label, also die Beschreibung ist. Man muss die erste Stelle jedoch angeben, hier fügen Sie einfach `None` ein. Die zweite Stelle, also der Titel des Fensters lautet, „Zoll in Zentimeter umrechnen“ und als Label fügen Sie „Eingabe in Zoll (Inch)“ ein. Die Ausgabe des Input-Dialoges kann mehrere Werte enthalten. Wir möchten jedoch nur den ersten auslesen. Daher geben wir nach der Anweisung in eckigen Klammern eine `0` ein, die den ersten Wert darstellt.

Auf diese Weise entsteht die zweite Zeile aus Listing 1, die Sie jetzt in das Makro einfügen.

Sie können das Makro einmal ausführen und in den Eingabedialog etwas eintragen. Hier passiert jedoch noch nichts. Daher programmieren Sie als nächstes die Berechnung. Erstellen Sie wieder eine Variable, um das Ergebnis der Berechnung zu speichern, diese nennen Sie einfach `ergebnis`. Der Datentyp ist zu dem jetzigen Zeitpunkt wiederum egal.

Um Zoll (Inch) in Zentimeter umrechnen, verwendet man eine einfache Formel:

$$\text{Zoll} \cdot 2,54 = \text{Zentimeter}$$

Anstelle von Zoll verwenden wir die Variable `eingabe`, die unseren eingegebenen Wert enthält. Da es sich jedoch um einen Text, also einen String handelt, müssen wir ihn erstmal in eine Zahl umwandeln, um damit rechnen zu können. Dafür verwenden wir den Datentyp `float`, der den String in eine Gleitkommazahl umwandelt. In die Klammer kommt dann die String-Variable `eingabe`. Mit dem Komma in der Zahl `2,54` lässt sich auch nicht rechnen, dieses muss man in einen Punkt umwandeln. Das kommt daher, dass im amerikanischen Maßsystem ein Komma einen Punkt darstellt und umgekehrt.

Mit dieser Berechnung als dritte Zeile im Makro erhalten Sie ein Ergebnis:

```
ergebnis = float(eingabe) * 2.54
```

Wenn Sie das Makro jetzt ausführen, errechnet es ein Ergebnis, was aber noch nicht ausgegeben wird und daher nicht sichtbar ist. Dieses Ergebnis wollen wir jetzt in einer Messagebox ausgeben. Hier können Sie wieder in der Python-Konsole `QtGui` mit einem folgenden Punkt eingeben und `QMessageBox` aus der Auswahlliste auswählen. Die Optionen in der Klammer sind ähnlich, wobei wir als Titel „Ergebnis in Zentimeter“ und als Label die Ergebnis-Variable angeben. Da es sich bei der Ergebnis-Variablen um eine Zahl handelt, müssen wir sie wieder in einen Text

Listing 2

```
from PySide import QtCore, QtGui
eingabe = QtGui.QInputDialog.getText(None,"Zoll in Zentimeter umrechnen","Eingabe in Zoll (Inch)") [0]
if eingabe != "":
    ergebnis = float(eingabe) * 2.54
    QtGui.QMessageBox.information(None,"Ergebnis in Zentimeter",str(ergebnis))
```

umwandeln, um den Wert als Label angeben zu können. Hierzu verwenden Sie einfach den String-Befehl `str`, wobei Sie in der Klammer die Ergebnis-Variable angeben. So ergibt sich die 4. Zeile im Makro.

Wenn Sie jetzt das Makro starten, können Sie im Eingabefenster eine Zahl in Zoll angeben, die in Zentimeter umgerechnet als Ergebnis in einer Messagebox ausgehen wird. Damit wäre der Rechner für die Umrechnung von Zoll in Zentimeter mit vier Codezeilen fertig.

Fehler abfangen

Man könnte jedoch in das Eingabefenster auch einfach einen Text eingeben. Dann erscheint beim Versuch der Umrechnung eine Fehlermeldung im Ausgabefenster. Diesen Fehler kann man umgehen, indem man an dritter Stelle vor der Berechnung eine bedingte Anweisung verwendet. Diese funktioniert nach dem Prinzip: falls Bedingung, dann Anweisung.

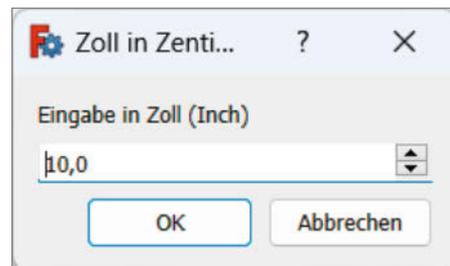
Im ersten Schritt fangen wir den Fall ab, dass die Eingabe-Variable ganz leer ist. Falls nicht, dann führe die Berechnung durch und gebe das Ergebnis aus. Bei einer `if`-Abfrage

muss man in Python die nachfolgenden Anweisungen mit einem Tabulator einrücken, ansonsten erkennt die `if`-Abfrage die Anweisungen nicht und es erscheint eine Fehlermeldung.

Der Code des Makros, das sich nicht von leeren Eingaben verwirren lässt, steht im Listing 2.

Um das Beispiel weiterzutreiben, könnte man jetzt noch bei der Eingabe abfragen, ob es sich um einen String oder eine Zahl handelt, um eine Fehlermeldung zu vermeiden. Dazu ändern Sie im Makro in `getDouble` die Funktion `getText` in `getDouble`. Diese gibt immer eine Zahl zurück. Falls es keine Eingabe gibt, ist die Zahl 0. Daher passen wir noch die `if`-Abfrage an, indem wir die Berechnung nur durchführen, wenn die Zahl größer 0 ist. In der Berechnung brauchen Sie dann die Eingabe-Variable auch nicht mehr in eine Zahl umzuwandeln, dies fällt einfach weg (Listing 3). Die Eingabebox sieht dann ebenfalls leicht anders aus: Es ist eine Zahl mit 0,0 vorgegeben, die sich mit Pfeilbuttons erhöhen und vermindern lässt.

Man kann dem Rechner auch noch weitere Dinge hinzufügen und das Makro ausbauen.



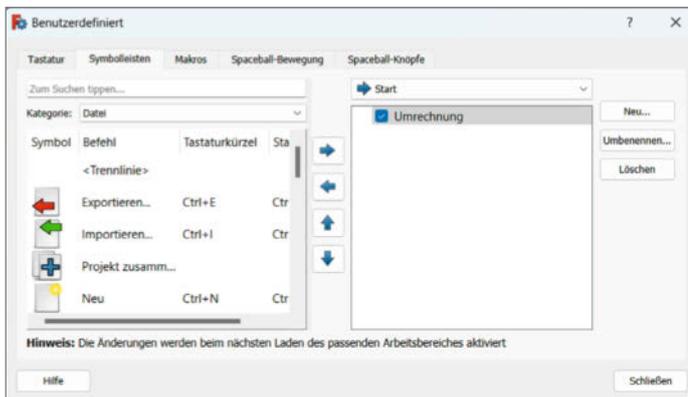
Eingabedialog mit vorgegebenem Zahlenformat

Versuchen Sie selbst einmal, den Rechner nach Ihren Wünschen zu erweitern. Einen ähnlichen Einheitenrechner finden Sie in FreeCAD unter `Werkzeuge/Einheitenrechner`, dieser ist jedoch anders aufgebaut als unser Programmierbeispiel.

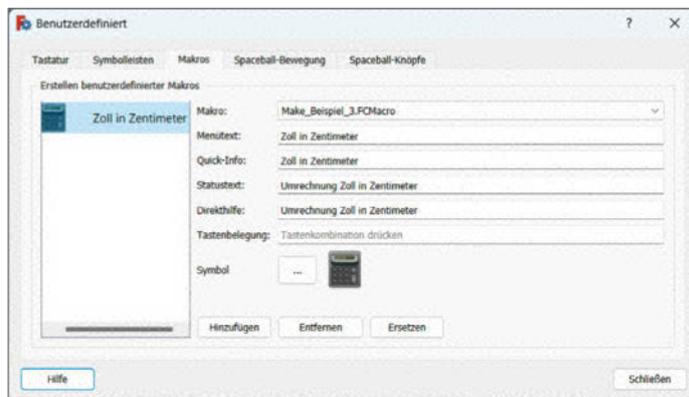
Jetzt können Sie noch das erstellte Makro mit einem Button in die Symbolleiste einbinden, um mit einem Klick die Umrechnung zu starten. Öffnen Sie dazu in der Menüleiste von FreeCAD das Fenster unter `„Werkzeuge/ Benutzerdefiniert ...“` und wählen Sie den Karteireiter `„Symbolleisten“` aus. Mit dem Button

Listing 3

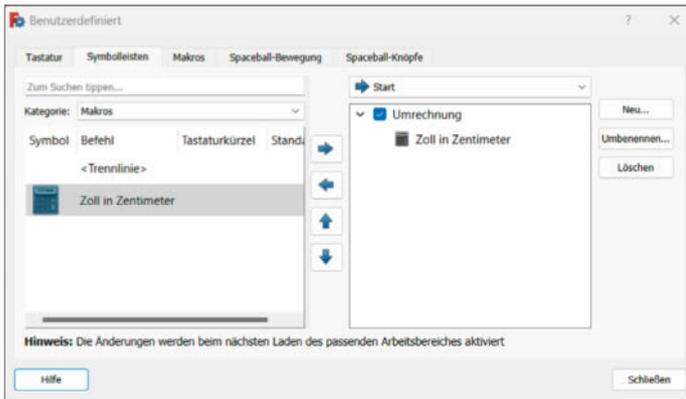
```
from PySide import QtCore, QtGui
eingabe = QtGui.QInputDialog.getDouble(None,"Zoll in Zentimeter umrechnen","Eingabe in Zoll (Inch)") [0]
if eingabe >= 0:
    ergebnis = eingabe * 2.54
    QtGui.QMessageBox.information(None,"Ergebnis in Zentimeter",str(ergebnis))
```



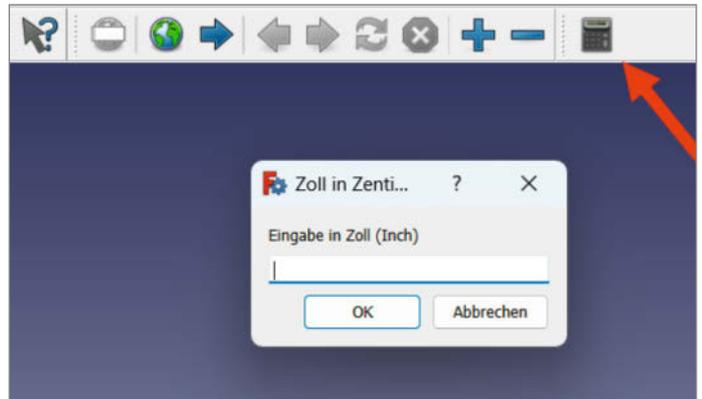
Symbolleiste für Umrechnung erstellen.



Neues Makro „Zoll in Zentimeter“



Zuordnung des Makros zur Symbolleiste



Fertiges Makrosymbol mit Eingabedialog

„Neu“ erstellen Sie eine neue benutzerdefinierte Symbolleiste, wobei Sie ihr den Namen „Umrechnung“ geben.

Wechseln Sie nun in den Karteireiter „Makros“ und wählen unter „Makro“ das eben erzeugte „Make_Beispiel_3.FCMacro“ aus. Geben Sie bei „Menütext“ und „Quick-Info“ jeweils den Text „Zoll in Zentimeter“ ein. Bei „Statustext“ und „Direkthilfe“ können Sie jeweils „Umrechnung Zoll in Zentimeter“ eingeben. Wählen Sie unter „Symbol“ das Taschenrechner-Symbol aus, dies ist das Symbol mit dem Namen /icons/accessories-calculator.

svg. Betätigen Sie nun den Button „Hinzufügen“, damit erscheint auf der linken Seite des Fensters ein Symbol mit dem Namen „Zoll in Zentimeter“.

Wechseln Sie nun zurück in den Karteireiter „Symbolleisten“. Wenn Sie auf der linken Seite unter „Kategorie“ den Eintrag „Makros“ auswählen, erscheint das neu erstellte Makro in der Liste. Wählen Sie es nun mit der Maustaste aus und betätigen Sie in der Mitte den Pfeil nach rechts, um das Makro der Symbolleiste zuzuordnen. Damit erscheint das Makrosymbol auch in der Symbolleiste. Schließen Sie das

Fenster und führen das Makro mit dem neuen Symbol einmal aus.

Wenn Sie sich weiter in die Programmiersprache Python einarbeiten wollen, möchten wir Ihnen ein Buch empfehlen (siehe Kasten). Python taucht in der Maker-Welt in den unterschiedlichsten Kontexten auf und so können Kenntnisse dieser Sprache an vielen Stellen nützlich sein. Das Buch zeigt zwar die Python-Grundlagen in einer anderen Entwicklungsumgebung, beherrscht man die erst einmal, kann man mit den hier vorgestellten Methoden aber auch versiert FreeCAD programmieren. —pek

Buchtipps: Einstieg in Python

Python ist eine Programmiersprache, die einfach zu erlernen ist und auch bei Makern gerne zum Einsatz kommt, beispielsweise bei der Programmierung eines Raspberry Pi. Das Buch „Einstieg in Python“ von Thomas Theis bietet eine Einführung in die Sprach-elemente und Programmier-techniken dieser Sprache, die sich gerade für Anfänger sehr gut eignet. Am Anfang geht es mit „Hallo Welt“ los, um dann jeweils mit kurzen Erklärungen und dazugehörigen Beispielen tiefer einzusteigen. Die vielfältigen Programmierbeispiele sind als Download auf der Webseite des Verlages verfügbar, sodass man nicht alles selbst tippen muss und direkt loslegen kann.

Als Programmierumgebung verwendet der Autor den in Python enthaltenen Editor IDLE, der nicht überladen ist und sich auf das Wesentliche konzentriert. Ein Programmierprojekt in Form eines Spiels begleitet den Kurs, wobei man es im Laufe des Buchs erweitert und verbessert. Der Programmierkurs bleibt jedoch nicht beim Einstieg in die Programmiersprache Python stehen, sondern geht weiter, indem er die Programmierung von Benutzeroberflächen oder auch das Einbinden von Datenbanken erklärt. Wer ein Buch zum Lernen, aber auch zum Nachschlagen sucht, findet hier allein vom Umfang her eine gute Quelle.

Hinweis: Dieses Buch erscheint in einem Verlag, der wie die Maker Media GmbH zur heise group gehört.



| | |
|---------------|--|
| Autor | Thomas Theis |
| Verlag | Rheinwerk |
| Umfang | 455 Seiten |
| ISBN | 978-3-8362-8830-9 |
| Preis | 24,90 € (Buch), 22,16 € (E-Book), 29,90 € (Bundle) |

Neuer Input für Maker

Make Elektronik Special

Make Elektronik Special bietet einen einfachen und praxisorientierten Einstieg in Transistorschaltungen, die Maker in eigenen Projekten einsetzen können. Das mitgelieferte Experimentierset inkl. Breadboard, Kabeln und 45 Elektronikbauteilen enthält alles, um die gezeigten Schaltungen sofort nachbauen und testen zu können.

Heft + Experimentierset für 44,95 €

 shop.heise.de/make-elektronik21



Inklusive Experimentierset und Breadboard

Make Operationsverstärker Special

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.

Heft + Experimentierset für nur 49,95 €

 shop.heise.de/make-opv



Inklusive Experimentierset

Make PI Pico Special

Mit dem Make Special PI Pico steigen Sie ein in die Welt der Programmierung von ARM-Mikrocontrollern. Make zeigt in dem 64-seitigen Special, welche Entwicklungsumgebungen es für den Raspberry Pi Pico gibt, wie man sie installiert und wie man sie nutzt.

Heft + Raspberry Pi Pico für 24,95 €

 shop.heise.de/make-pico



Inkl. Raspberry Pi Pico RP2040

Recyceltes Nylon-Filament

aus alten Fischernetzen



Fillamentum

Polyamid (Nylon) stellt als Druckmaterial seine eigenen Herausforderungen: Ein beheizter Druckraum, Extrudertemperaturen um 240 °C, Druckbett um 100 °C und absolute Trockenheit des Filaments sind notwendig, um sinnvoll drucken zu können. Gelingt es, diese Herausforderungen zu meistern, liefert Nylon sehr haltbare und temperaturstabile Druckteile, die eine hohe Beständigkeit gegenüber vielen Chemikalien und Lösungsmitteln aufweisen.

Fillamentum, ein europäischer Hersteller hochwertiger 3D-Druck-Materialien, bietet nun Nylonfilamente aus recycelten Fischernetzen an. Neben der Einsparung von über 95 % CO₂-Emissionen bei der Herstellung von Nylon wird damit auch ein Zeichen gegen die gefährliche Umweltverschmutzung durch einfach ins Meer geworfene gebrauchte Fischernetze gesetzt. Die Farbe des Porthurno-Filaments erinnert an das Meer. Die schwarze Variante Orca enthält 10 % Kohlefasern und hat noch bessere mechanische Eigenschaften als reine Nylonfilamente. Die Temperaturbeständigkeit von bis zu 180 °C ermöglicht das Autoklavieren (Sterilisieren bei 120 °C in Dampf) und die Kerbschlagzähigkeit bleibt bis -20 °C erhalten.

Die recycelten Rohstoffe erhält Fillamentum vom Gründer und Inhaber der Firma „Fisly Filaments“, Ian Falconer aus Cornwall in England (Links siehe make-magazin.de/x5es). Dort wird das Filament lokal und ohne den Einsatz von Chemikalien und Zusatzstoffen hergestellt

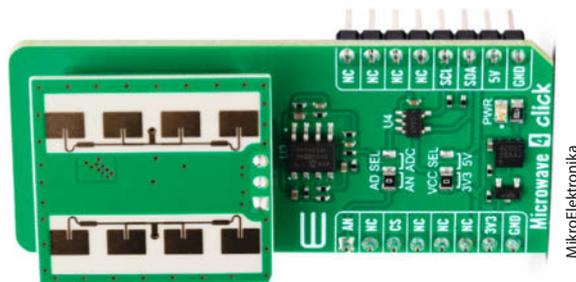
—caw

| | |
|-------------------|--|
| Hersteller | Fillamentum Manufacturing Czech s.r.o. |
| URL | fillamentum.com |
| Preis | Porthurno 88 €/kg (auf 750-g-Spule) Orca 131 €/kg (auf 600-g-Spule) |

Microwave 4 Click

Mikrowellen-Bewegungsmelder

MikroElektronika (MikroE) hat das neue Click-Board „Microwave 4 Click“ vorgestellt. Es handelt sich dabei um einen Bewegungssensor. Die Platine basiert auf dem PD-V12 von pdlux. Der Sensor sendet ein Mikrowellensignal (24,1 GHz, <3 mW nach FCC/CE) aus, das von einem sich bewegenden Objekt reflektiert wird und durch den Dopplereffekt eine Frequenzänderung erfährt. Durch diese Frequenzänderung des empfangenen Signals erkennt das Gerät eine Bewegung. Die Daten des Sensors können entweder analog oder über I²C ausgelesen werden. Damit lassen sich Anwendungen wie Einbruchmeldeanlagen, automatische Türöffner, Anwesenheitserkennung und vieles mehr realisieren.



MikroElektronika

Das Board ist 3,3 und 5 Volt kompatibel, die Antennen auf dem Board arbeiten ab 20 cm Abstand zum Objekt und haben bei Wandmontage nur einen sehr kleinen toten Winkel, was den Sensor für Einbruchmeldeanlagen prädestiniert.

Der Click-Board-Standard von MikroElektronika ermöglicht die Entwicklung von Prototypen ohne Löten von Bauteilen – die einzelnen Boards können zusammengesteckt werden. Derzeit gibt es über 1500 Click-Boards, die mit verschiedenen Basisboards (z. B. Raspberry Pi oder Arduino) verwendet werden können.

Das Microwave 4 Click ist für 21 US-Dollar auf der Seite des Herstellers erhältlich, wo auch eine ausführliche Dokumentation und Downloads zu finden sind.

—das

| | |
|-------------------|--|
| Hersteller | MikroElektronika |
| URL | mikroe.com/microwave-4-click |
| Preis | 21 US-\$ |

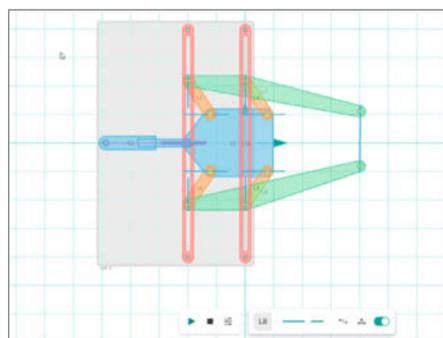
MotionGen Pro

Kinematiken online entwerfen

Die Verbindung eines Kolbens über ein Gestänge zu einem Antriebsrad hat jeder schon einmal bei einer Dampfmaschine oder einer Dampflokomotive gesehen. Hier wird eine lineare Bewegung in eine Drehbewegung umgewandelt, was noch recht einfach zu verstehen und zu visualisieren ist. Mit ein paar mehr Verbindungen sind aber auch Bewegungspfade möglich, die sich jeder Intuition entziehen. Hier setzt MotionGen Pro an. Ursprünglich für die SnappyXO-Roboter entwickelt, hat es sich mittlerweile zu einem universellen Werkzeug entwickelt. Die Nutzung von MotionGen Pro ist kostenlos, die Registrierung erfordert jedoch eine verifizierbare E-Mail-Adresse und einige Angaben wie Name und Organisationsform.

Aufgrund seiner Nähe zu STEM und Bildung ist MotionGen Pro gut mit (englischen) Tutorials und Videos dokumentiert (Links siehe make-magazin.de/x5es). Nach ein paar Videos sollte man schon loslegen können, man kommt aber auch mit etwas Herumprobieren recht weit und simuliert so schnell den ersten kinematischen Apparat oder Roboter.

MotionGen läuft komplett im Browser, man kann seine Kreationen in der Cloud oder lokal



speichern. Der Export ist als Bild, PDF und SVG möglich, aus der 3D-Ansicht auch als STL- oder OBJ-Dateien. So hat man in wenigen Minuten 3D-druckbare Teile, mit denen man seine Konstruktion testen kann.

Aber Vorsicht: Auch wenn man kein Projekt plant, verliert man schnell Zeit beim Versuch, möglichst komplizierte und interessante Bewegungsbahnen zu erzeugen.

—caw

| | |
|-------------------|--|
| Hersteller | Mechanismic Inc. |
| URL | motiongen.io |
| Preis | kostenlos |

JoyWarrior

Joysticks per USB treiberlos und einfach ansteuern

Ausprobiert
— von Make: —

Der Anschluss von Joysticks an Windows-, Linux- oder OSX-Computer über USB ist mit den Chips der JoyWarrior-Serie ohne Programmierung von Mikroprozessoren und ohne Treiberprobleme möglich.

Die eigentlich für industrielle Anwendungen gedachten Chips und Produkte der bei Berlin ansässigen Firma Code Mercenaries wurden schon immer auch gerne von Modellfliegern, Hobby-Flugsimulator-Bastlern und Makern genutzt, um Joysticks und einige Tasten einfach und treiberlos an USB-Ports anzuschließen. Jetzt gibt es einen neuen Chip, der vier analoge Achsen mit 12 oder per Matrix 32 Tasten an ein USB-fähiges Gerät anbinden kann.

Den Chip gibt es im Webshop von Code Mercenaries auf einem bastelfreundlichen Modul für knapp 20 Euro, die einzelnen Chips sind im QFN28-Formfaktor erhältlich. Außerdem gibt es eine breadboardfreundliche Variante mit Adapterplatine auf DIL28 (passt auch in IC-Sockel).

Der Joystick-Teil bietet vier analoge Achsen mit 12 Bit Auflösung pro Achse, 16-fachem Over-

sampling und digitalem Filter. Damit können 1000 präzise Achspositionen pro Sekunde mit minimalem Rauschen ausgelesen werden. Als Positionsgewer kann alles verwendet werden, was Spannungen (0 bis 3,3V) ausgibt. Potentiometer arbeiten dann als Spannungsteiler und Hall-Sensoren oder Hall-ICs als Spannungsquelle, aber auch Abstandssensoren oder ähnliches mit analogem Ausgang sind denkbar.

Als digitale Eingänge können bis zu 32 Taster oder andere Schalter angeschlossen werden. Hier bietet der JoyWarrior28A12-32 zwei Modi: Es können 12Taster direkt angeschlossen werden oder es kann eine 4x8-Tastenmatrix verwendet werden, um bis zu 32 Taster anzuschließen.

Es werden keine speziellen Gerätetreiber benötigt, die JoyWarrior-Chips verwenden die



Code Mercenaries

systemeigenen Treiber (USB 1.1/2.0) und können über die vorhandenen Softwareschnittstellen genutzt werden.

Das Modul und der DIL-Adapter sind im Webshop von Code Mercenaries erhältlich, Einzelchips sind bei Segor gelistet. —caw

| | |
|-------------------|--|
| Hersteller | Code Mercenaries Hard- und Software GmbH |
| URL | make-magazin.de/x5es (Shop- und Infolinks) |
| Preis | Je nach Ausführung ab ca. 15 € |

Bambulab P1S

Schneller 3D-Drucker mit automatischem Filamentwechsel

Ausprobiert
— von Make: —

Der Bambulab P1S bildet zwischen dem frisch vorgestellten A1 mini und dem teureren X1 Carbon einen Kompromiss mit attraktiven Eigenschaften. Neben schnellem Druck mit 500 mm/s, CoreXY-Bauweise und geschlossenem Gehäuse bietet er mit dem Automatic Material System (AMS) einen separat erhältlichen Materialwechsler mit geschlossener Filamentbox. Das AMS lässt sich auf den Deckel des Druckers stellen und bietet in der Basis-Version vier Filamentrollen Platz. In der maximalen Ausbaustufe lassen sich vier AMS kombinieren und so insgesamt 16 Filamentrollen vorhalten.

Im hauseigenen Slicer BambuStudio lässt sich beim Start eines Druckes auswählen, welches Filament genutzt werden soll, manuelles Spulenwechseln ist dann nicht mehr nötig. Es lassen sich aber auch innerhalb eines Druckes unterschiedliche Filamente kombinieren und so mehrfarbige Objekte drucken. Mischen kann der Drucker die Filamente jedoch nicht, aus gelb und blau lässt sich also kein grün erzeugen. Flexible Materialien dürfen nicht ins AMS, diese werden ganz klassisch an die Rückseite des Druckers

gehängt und manuell zum Druckkopf geschoben.

In unserem Test funktionierte der automatische Materialwechsel in den meisten Fällen reibungslos, gelegentlich schaffte es das AMS jedoch nicht, das Filament aus dem Druckkopf zurückzuziehen. Nach kurzer manueller Hilfestellung ließ sich der Druck aber ohne sichtbaren Druckfehler fortsetzen.

Für jeden Materialwechsel verlängert sich die Druckzeit um rund 90 Sekunden, was bei Drucken mit vielen Wechsels die Druckzeit vervielfachen kann. Außerdem muss der Drucker bei jedem Wechsel die Düse entleeren, was jedesmal einige Gramm Filament verbraucht. Dies kann auf Wunsch auch in den Stützstrukturen oder in der Objektfüllung verdrückt werden, landet sonst aber als kleines Filamentknäuel auf dem Tisch hinter dem Drucker.

Die Druckqualität des P1S konnte uns im Test vollständig überzeugen: Dank Schwingungskompensation korrigiert der Drucker Vibrationen und erzeugt makellose Oberflächen, denen man kaum ansieht, dass sie gedruckt wurden. Zu unserem Test haben



Bambu Lab

wir ein Video veröffentlicht, dass Sie auf dem YouTube-Kanal des Make Magazins (@MakeMagazinDE) finden. —jom

| | |
|-------------------|------------------------------|
| Hersteller | Bambu Lab |
| URL | eu.store.bambulab.com |
| Preis | 899 € inkl. AMS (659 € ohne) |

VCNL4020 Proximity and Light Sensor

Licht- und Entfernungssensor auf Breakout-Board



Adafruit hat einen neuen Lichtsensor mit Entfernungsmessfunktion für berührungslose Lichtschalter und ähnliches auf den Markt gebracht. Der VCNL4020 Proximity and Light Sensor ist ein über I²C ansprechbares Breakout-Board, das dem STEMMA-QT-Standard folgt. Natürlich sind auch Lötunkte vorhanden, falls man das Board in einem anderen Kontext verwenden möchte. Es ist der Nachfolger des 2019 erschienenen Adafruit VCNL4040 und bietet einen größeren Bereich der Helligkeitsmessung.

Adafruit VCNL4020 verfügt über eine Reihe konfigurierbarer Parameter wie die Anzahl der Samples pro Sekunde (Integrationszeit zwischen den Messungen) – mehr Samples pro Sekunde bedeuten schnellere Ergebnisse, sind aber bei der Messung von schwachem Licht nicht so genau. Mit 16-Bit-Messungen und 0,25 Lux pro Bit können Lichtstärken von 0,25 bis 16.000 Lux gemessen werden. Zusätzlich können verschiedene Filter eine Tiefpass- oder Mittelwertbildung übernehmen und so die Ergebnisse durch das Herausfiltern von Ausreißern verbessern.

Der Sensor misst Distanzen bis zu 20 cm. Er kann sowohl mit 3,3 Volt als auch mit 5 Volt betrieben werden. Damit ist das Board zu den gängigen Arduino, Raspberry Pi & Co. Mikrocontrollern kompatibel.

Das Breakout-Board hat die Abmessungen 25,3 × 17,8 × 4,6 mm und ist für 5,95 Dollar im Adafruit Shop erhältlich.

—das

| | |
|------------|---------------------------|
| Hersteller | Adafruit Industries, LLC |
| URL | adafruit.com/product/5810 |
| Preis | 5,95 US-\$ |

Alta AI Einplatinencomputer

Raspberry-Pi-Alternative mit KI-Unterstützung

Der Hersteller Libre Computer hat mit dem Alta AI einen Raspberry-Pi-Konkurrenten auf Basis des Amlogic A311D-Prozessors entwickelt. Er soll sich besonders für KI-Anwendungen eignen. Der Alta bietet 4 Gigabyte LPDDR4-RAM, die CPU verfügt über vier ARM Cortex-A73-Kerne für mehr Leistung und zwei sparsamere ARM Cortex-A53-Kerne. Zusätzlich ist eine NPU (Neural Processing Unit) für KI-basierte Aufgaben integriert.

Neben Standard-Ports, wie einer 3,5-mm-Klinkenbuchse für Audio, RJ45 für Gigabit-LAN und vier USB-3.0-Anschlüssen verfügt der SBC im Gegensatz zum aktuellen Raspberry Pi 5 über einen Standard-HDMI-Ausgang. Die 40-polige GPIO-Stiftleiste bietet laut Hersteller eine hohe Kompatibilität zum Raspberry Pi Interface. Die Stromversorgung ist über einen Power-over-Ethernet-Anschluss (PoE) oder über die USB-C-Buchse möglich, die neben USB-Power auch USB-C-Daten-Protokolle spricht. Das Board kann mit Standard-Linux-Betriebssystemen betrieben werden. Im Leerlauf soll das Board nur ein Watt verbrauchen, unter Vollast nicht mehr als zehn Watt.



Libre Computer

Das Alta-AI-Board soll im Dezember 2023 ausgeliefert werden. Libre Computer hat bereits in der Vergangenheit interessante SBCs mit Rockchip-Hardware angeboten, die den am Markt erhältlichen Pis meist leistungsmäßig überlegen waren. Ob das neue Board mit dem Pi 5 mithalten kann und wie gut die NPU unterstützt wird, müssen Tests zeigen, sobald das Board verfügbar ist.

—das

| | |
|------------|-------------------------|
| Hersteller | Libre Computer |
| URL | libre.computer |
| Preis | ca. 56 € (Vorbesteller) |

USB Type C Power Delivery Dummy Breakout

Breakout-Board für Stromversorgung

Dieses neue Produkt von Adafruit basiert auf dem HUSB238-Chip. Das Board wird über USB-C an ein PD-(Stecker)-Netzteil angeschlossen und kann bis zu 100 Watt per Power-Delivery für ein Projekt abrufen. Das Board kann die Spannung auf zwei verschiedene Arten anfordern: fest verdrahtet über Jumper (die Widerstände konfigurieren) oder dynamisch über I²C.

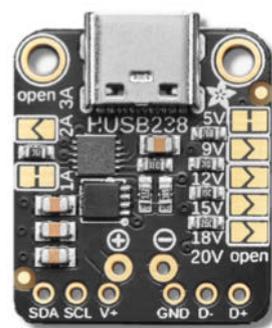
Standardmäßig ist es auf 5 V und 1 A eingestellt, da USB-C diese Spannung immer zuerst liefert. Durch Trennen des 5V-Jumpers und Überbrücken (Löten) der 9V-, 12V-, 15V-, 18V- oder 20V-Jumper kann die Spannung (sofern vom PD-Netzteil bereitgestellt) konfiguriert werden. Für spezielle Anwendungen kann auch der gewünschte Strom von 2 A bis 3 A gewählt werden.

Über I²C kann die Adafruit-HUSB238-Bibliothek verwendet werden, um die verfügbaren Spannungen auszulesen und eine bestimmte Spannung anzufordern. Werden keine I²C-Befehle an den Chip gesendet, werden automatisch die fest verdrahteten Spannungen

verwendet. Je nach eingestellter Spannung gibt das Board also bis zu 20 V aus, was bei der Integration in ein System mit Mikroprozessorboards zu beachten ist und so entsprechende Sicherheitsvorkehrungen getroffen werden müssen, damit das steuernde Board nicht durchbrennt.

Das Board hat die Abmessungen 24,5 × 20,2 × 4,9 mm und ist im Adafruit Store für 5,95 Dollar erhältlich.

—das



Adafruit

| | |
|------------|---------------------------|
| Hersteller | Adafruit Industries, LLC |
| URL | adafruit.com/product/5807 |
| Preis | 5,95 US-\$ |

Einplatinenrechner ROC-RK3588-RT

SBC mit drei Ethernet-Ports und KI-Unterstützung

Der Rockchip SoC RK3588 verfügt über acht Kerne, von denen vier A76-Kerne (2,4 GHz) und vier A55-Kerne (1,8 GHz) sind. Die GPU ist eine Mali-G610 MP4, die neben der Enkodierung und Dekodierung von H.264, H.265, VP8, VP9, VC-1 und AVC auch die Dekodierung für den AV1-Codec unterstützt.

Das Board ist in der Maximalausstattung mit bis zu 32 GB RAM und bis zu 128 GB integriertem eMMC-Speicher erhältlich. Neben einem SD-Kartenslot (bootfähig) zur Speichererweiterung, verfügt das Board über einen NVMe kompatiblen M.2 Slot mit einer PCIe-2.0-

Lane. Dieser Slot kann allerdings nicht für ein Bootmedium verwendet werden. Zusätzlich verfügt das Board über einen zweiten M.2 Slot mit E-Key. Damit kann laut Hersteller ein WLAN- und Bluetooth-Modul nachgerüstet werden. Ob andere Erweiterungen funktionieren, wird nicht angegeben.

Der Rockchip RK3588 besitzt außerdem eine integrierte NPU. Diese ist für die Beschleunigung von KI-Workloads vorgesehen. Auf der Webseite von Firefly wird ein SDK zur Nutzung der NPU zum Download (Links

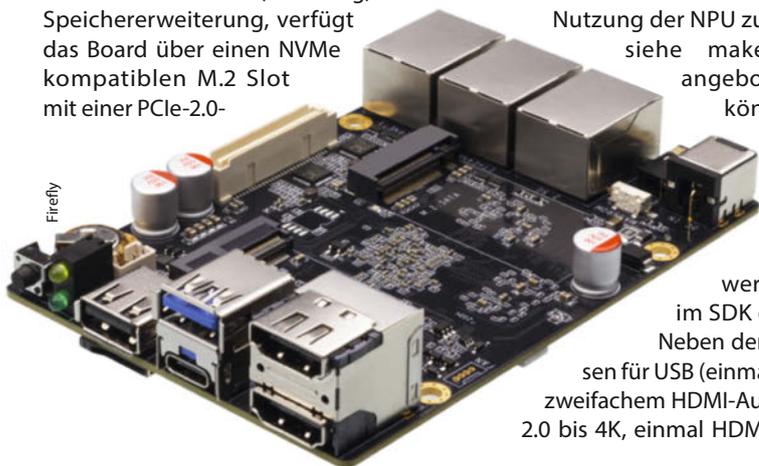
siehe make-magazin.de/x5es) angeboten. Mit dem SDK können verschiedene Modelle anderer Frameworks (TensorFlow, PyTorch, ONNX etc.) für die NPU konvertiert werden, Beispiele sind im SDK enthalten.

Neben den üblichen Anschlüssen für USB (einmal 2.0 und einmal 3.0), zweifachem HDMI-Ausgang (einmal HDMI 2.0 bis 4K, einmal HDMI 2.1 bis 8K), 3,5 mm

Klinkenbuchse und Display-Port-fähigem USB-C (bis 8K) verfügt das Board über drei RJ45-LAN-Anschlüsse (zweimal Gigabit und einmal 2,5 Gigabit).

Der Zugang für Maker zu den Ports für Erweiterungen ist nicht ganz einfach. Statt einer Reihe von GPIO-Pins wie bei ähnlichen Produkten sind PCIe, UART, USB, CAN, SARADC und GPIO-Anschlüsse in einem 60-poligen BTB-Stecker (Board to Board) gebündelt und entsprechend schwer abzugreifen.

Der Rechner wird mit Android 12 ausgeliefert, auf der Downloadseite des Herstellers finden sich auch Debian- und Ubuntu-Images. Ob diese auch alle Treiber für die leistungsstarke Hardware enthalten, müssen weitere Tests zeigen. —das



| | |
|------------|---|
| Hersteller | Firefly Open Source Team |
| URL | firefly.store/goods.php?id=221 |
| Preis | 229 US-\$ (8 GB RAM und 64 GB eMMC) |

LibrePCB Version 1.0

Kostenlose Leiterplatten-Entwurfs-Software

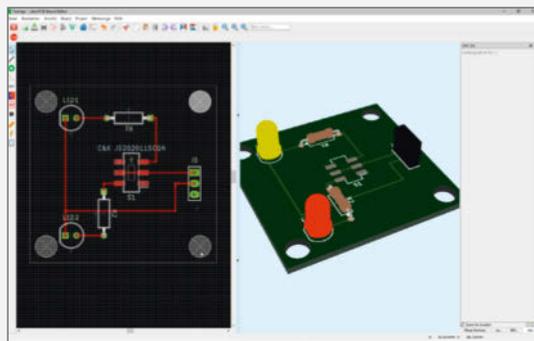
LibrePCB hat sich den Slogan „Create electronics the easy way“ gegeben und vor kurzem die Version 1.0.0 erreicht. Ziel ist es, dem Einsteiger eine logische und leicht erlernbare Oberfläche zu bieten, aber auch dem Fortgeschrittenen ein schnelles Arbeiten zu ermöglichen. Der Wechsel zwischen Schaltplanentwurf, Bauteilanpassung, Leiterplattenansicht und Bestellvorgang soll schnell und reibungslos erfolgen.

Um Anfängern und Umsteigern den Einstieg zu erleichtern, wurde auf bekannte Konzepte zurückgegriffen (Eagle-Umsteiger werden es merken). Tatsächlich braucht man kaum das Handbuch zu Rate zu ziehen, um erste einfache Platinen zu entwickeln, wenn man schon einmal mit einem ähnlichen Programm gearbeitet hat. Bei der Entwicklung wurde aber nicht nur auf das Äußere geachtet, sondern auch der Unterbau sorgfältig entwickelt, um die Bauteilverwaltung und höhere Funktionen nicht zu bremsen. Als Open Source Software (GPLv3) kann LibrePCB für praktisch alle Plattformen und Be-

triebssysteme übersetzt werden. Neben der Selbstkompilierung aus den Quellen gibt es Installer für die gängigsten Betriebssysteme und portablen Versionen, die auch auf Rechnern ohne Systemrechte genutzt werden können.

Die Oberfläche ist modern und an vielen Stellen intuitiver oder durchdachter als andere Open-Source-Alternativen, was LibrePCB auch für Einsteiger attraktiv macht. Besonders hilfreich sind die ersten Tipps, was als Nächstes zu tun ist, die bei einem leeren Projekt angezeigt werden und so manches Tutorial-Video ersparen. Oft findet man zum gewählten Schaltplanzeichen zwar kein passendes Bauteil, aber der Bauteil-Editor ist glücklicherweise sehr einfach zu bedienen.

Auch wenn man schon ein Lieblingsprogramm hat, kann es nicht schaden, einen Blick auf LibrePCB zu werfen. Findet man Fehler, so hilft der recht aktive Issues-Bereich



im GitHub des Projekts (Links siehe make-magazin.de/x5es) weiter. Wurde der Fehler noch von niemand anderem gefunden, so kann man an dieser Stelle auch einen Bugreport loswerden. —caw

| | |
|------------|---|
| Hersteller | Libre PCB (Opensource) |
| URL | librepcb.org |
| Preis | kostenlos |

IMPRESSUM

Make: Nächste Ausgabe erscheint am 9. Februar 2024

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: Justin Atkins, Beetlebum (Comic), Jochen Ferger, Michael Gaus, Rainer Goldelius, Kathrin Granneman, Georg Jacobs, Maximilian Kern, Ludo Kerz, Miguel Köhnlein, Markus Mauch, Matthias Mett, Mike Senese, Dirk Wahl

Assistenz: Susanne Cölle (suc), Christopher Tränkmann (cht), Martin Triadan (mat)

Leiterin Produktion: Tine „The Rock“ Kreye

DTP-Produktion: Martina Bruns, Martin Krefit (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

Welcher Bausatz hat dich als Kind am meisten geprägt?



Jochen Ferger
 Diez, erklärt ab S. 62, wie man eine Tischtennis-Uhr baut
 Mich prägte ein alter Märklin-Metallbaukasten am meisten. Besonders als ein Motor dazukam, konnte ich bewegte Modelle erstellen. Trotzdem glitt ich in die E-Technik ab. Schuld war ein Volkshochschulkurs.



Georg Jacobs
 Saarbrücken, behält auf S. 52 mit seinem Head-up-Display die Übersicht
 Ein Kosmos Chemiekasten, am Ende ein komplett ausgebautes Dachbodenlabor nicht nur für rasch ablaufende exotherme Reaktionen geringer Aktivierungsenergie. Nobody was harmed.



Ludo Kerz
 Makinghausen, baut auf S. 102 Adapter für Staubsauger
 Ich habe förmlich nach den Kosmos-Experimentierkästen und fischertechnik Elektronik gegiert. Ich habe alles aufgesaugt wie ein Schwamm. Und es war immer noch zu wenig. Wie viel einfacher ist das doch geworden!



Markus Mauch
 Radolfzell, beschreibt ab S. 19 seine Erfahrungen mit Robotern
 Ich hatte einen „Kosmos 2000“-Elektronikbaukasten. Eine Art Riesen-Breadboard mit Bauteilen zum Stecken. Hiermit habe ich LEDs zum Leuchten gebracht, Zitronen zu Batterien gemacht und Alarmanlagen gebaut.

Inserentenverzeichnis

dpunkt.verlag GmbH, Heidelberg 37
 ifm group services gmbh, Tett nang 17
 OXON AG, Liebfeld 47

Sertronics GmbH, Hamburg 19
 TUXEDO Computers GmbH, Augsburg 124
 Teile dieser Ausgabe enthalten Beilagen der DIMABAY GmbH, Berlin.

Maker Faire®

Das Format für
Innovation & Macherkultur

Die nächsten Events



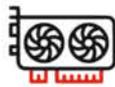
... weitere folgen.



Hocheffizienter Leichtathlet TUXEDO Pulse 14 - Gen3 mit **AMD**



CPU-Leistung



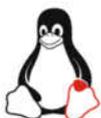
Grafikleistung



Mobilität



Akkulaufzeit



Linux kompatibel



Bis zu 5 Jahre Garantie



Sofort einsatzbereit



Gefertigt in Deutschland



Deutscher Datenschutz



Deutscher Tech Support

TUXEDO

[tuxedocomputers.com](https://www.tuxedocomputers.com)

© Copyright by Maker Media GmbH.