DEUTSCHLANDS GEFÄHRLICHSTES DIY-MAGAZIN

c't Make:

Make:

So geht's:
Firewall und VPN mit dem Pi

ToDo-Leuchte

- Erinnert blinkend an Aufgaben
- Multiuser-fähig
- Mit NeoPixeln, ESP32 und MicroPython

Licht-Projekte

- 3D-Druck-Filament als Lichtleiter
- Holografische Bilder in der Flasche
- Belichtungsmesser mit Pi Pico

Know-how

- WSL2 installieren und nutzen
- WLAN in Wokwi simulieren
- RAM, EPROM und Flash verstehen

Kampfroboter

- Einfach nachzubauen
- Preiswert und 3D-gedruckt
- Ferngesteuert

Präzisionslasern

- Lightburn installieren
- Mit Kamera millimetergenau positionieren

1/24
9.2.2024
CH CHF 26.50
AT 14,90
Benelux 15,90
€ 13,50

4 192314 113500 01

www.make-magazin.de

Linux ist geiler

Bislang kennen nur wenige Eingeweihte den Scoop meines c't-Kollegen Peter Siering in der c't-Ausgabe c't 19/04 im Artikel „XP-Schluckimpfung". Der Artikel dreht sich zwar um das Service Pack 2 für Windows, eine damals bahnbrechende Verbesserung der Sicherheitsfunktionen. Aber im Text ist eine subliminale Botschaft versteckt. Reiht man die Anfangsbuchstaben der einzelnen Absätze aneinander, so ergibt das den Satz „Linux ist geiler". Wer es nicht glaubt: Prüfen Sie selbst (siehe Link).

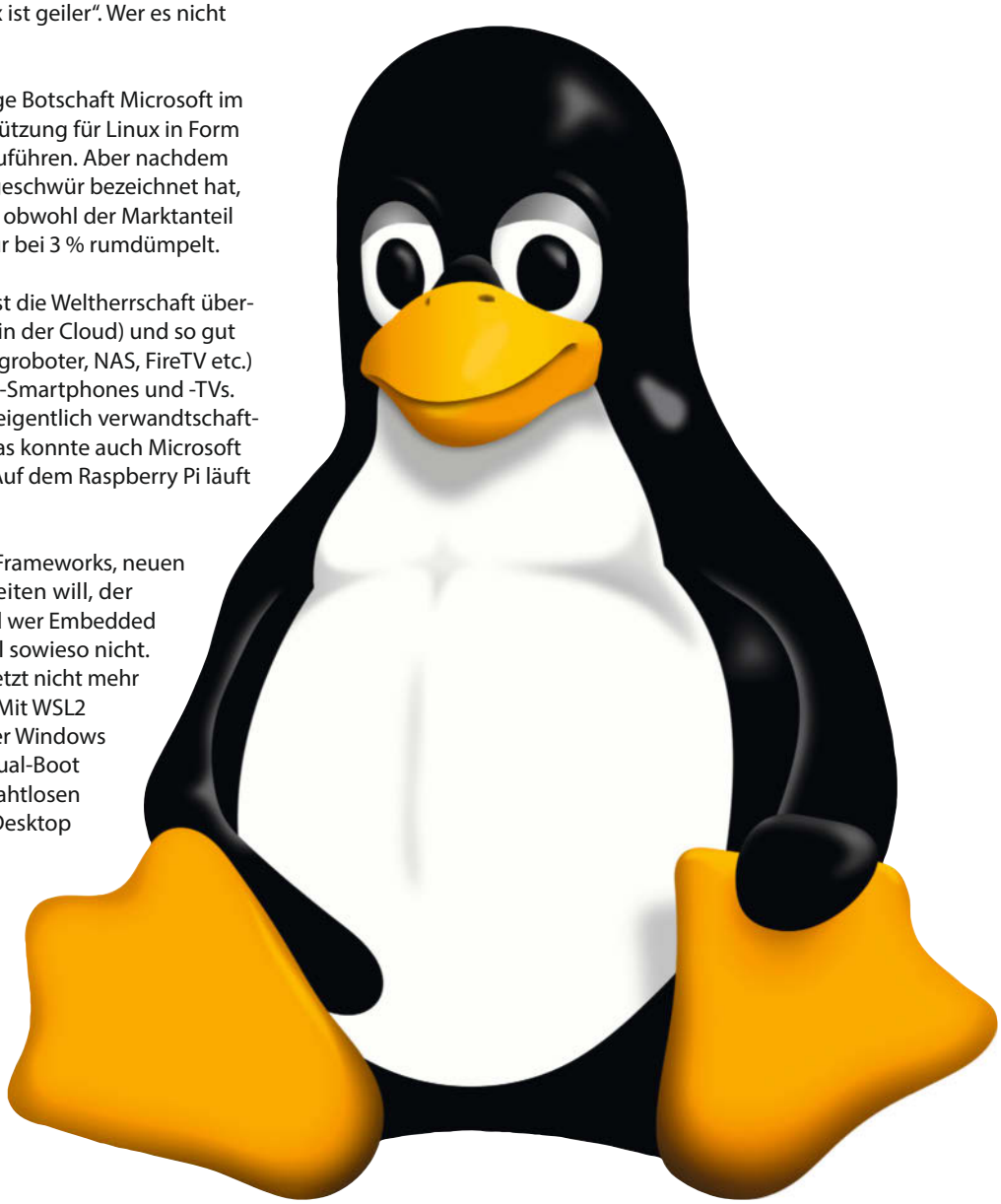
Wir wissen nicht, ob diese unterschwellige Botschaft Microsoft im Jahr 2016 dazu bewogen hat, die Unterstützung für Linux in Form des „Windows Subsystem for Linux" einzuführen. Aber nachdem 2001 Steve Ballmer Linux noch als Krebsgeschwür bezeichnet hat, ist dies ein respektabler Schritt. Und das, obwohl der Marktanteil von Linux auf dem Desktop weiterhin nur bei 3 % rumdümpelt.

Dennoch hat Linux im Hintergrund längst die Weltherrschaft übernommen. 80 % aller Server (einzeln und in der Cloud) und so gut wie alle Embedded Systems (Router, Saugroboter, NAS, FireTV etc.) laufen mit Linux. Dazu kommen Android-Smartphones und -TVs. Und mit seinem BSD ist iOS auf iPhones eigentlich verwandtschaftlich auch ganz nah dran an Linux. Und das konnte auch Microsoft nicht ignorieren. Ach ja, fast vergessen: Auf dem Raspberry Pi läuft natürlich auch ein Linux.

Wer mit coolen Software-Tools, offenen Frameworks, neuen Programmiersprachen und Docker arbeiten will, der kommt meist nicht um Linux herum. Und wer Embedded Systeme verstehen und beherrschen will sowieso nicht. Eingefleischte Windows-Nutzer müssen jetzt nicht mehr bangen, sich ihr System zu zerschießen. Mit WSL2 kann man frickeln ohne Reue: Linux unter Windows installieren, nutzen und lernen – ohne Dual-Boot oder Virtualbox. Und trotzdem mit der nahtlosen grafischen Integration in den Windows-Desktop und USB-Anbindung. Wie das geht, zeigt unser Artikel auf Seite 94. Wem es dennoch nicht gefällt, der kann es ohne Rückstände wieder deinstallieren.

Happy Hacking!

▶ **make-magazin.de/xea7**



Daniel Bachfeld

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

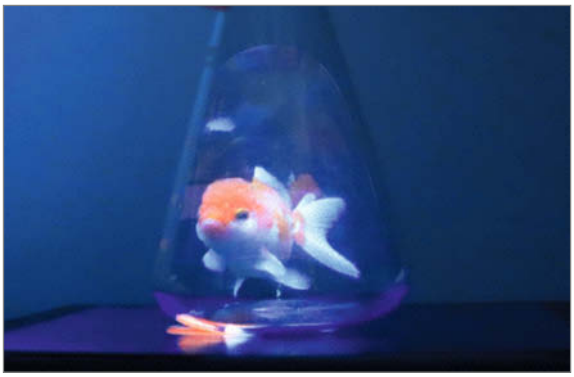
ToDo-Leuchte

Aufgaben, die man nicht im Blick hat, können schnell mal unter den Tisch fallen. Den blinkenden Task-Reminder kann man jedoch kaum übersehen. Er weiß genau, welche Aufgabe als Nächstes ansteht, symbolisiert das auf leuchtenden Täfelchen und die Leuchtfarbe zeigt an, wer sie erledigen muss. Widerstand und Ausreden sind daher zwecklos.

Inhalt

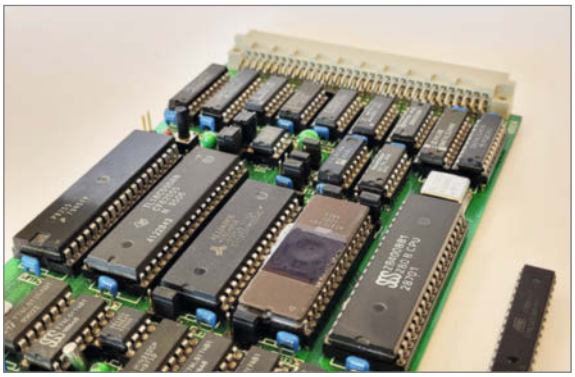
Licht-Projekte

Bei diesem Goldfisch im Erlenmeyer-Kolben handelt es sich um ein raffiniertes Spiegeltrick-Projekt, das den Wasserbewohner scheinbar als Hologra-Fisch erscheinen lässt. Falls Sie Licht eher praktisch verwenden, nützt Ihnen der DIY-Belichtungsmesser bei der Ausleuchtung Ihrer Fotos oder ein Lichtleiter aus 3D-Druck-Filament bei Leuchtanzeigen.



Know-how

Der Online-Simulator Wokwi kann auch WLAN nachbilden. Damit lassen sich dann auch IoT-Projekte testen. Falls Sie Linux verwenden möchten, aber „nur" einen Windows-PC besitzen, zeigen wir Ihnen, wie man darauf das Windows-Subsystem für Linux (WSL) installiert. In die Tiefen von Mikrocontroller-Boards führt Sie der Artikel über die verschiedenen Speicherarten, die dort Dienst tun.



Kampfroboter

Wollen Sie einen eigenen Kampfroboter bauen? Dann ist es Zeit für Kerfuffle – einen einsteigerfreundlichen Mini-Bot, der seinen Kontrahenten im Käfig ordentlich einheizt, obwohl er nur zur Kunststoff-Ameisengewichtsklasse gehört. Aber genau deshalb kann man ihn leicht selbst bauen, 3D-Drucker vorausgesetzt. Wie, das zeigt dieser Artikel ganz genau.

Präzisionslasern

Die Software, mit der Lasercutter gesteuert werden, ist oft grottenschlecht, besonders bei billigen fernöstlichen Geräten. Da gibt es besseres: Lightburn. Wir zeigen Ihnen nicht nur, wie sie die Software installieren und bedienen, sondern auch, wie man mithilfe einer Kamera über dem Lasercutter Schnitte und Gravuren zehntelmillimetergenau auf dem Schnittmaterial platziert, sodass selbst Zahnstocher beschriftet werden können.

Lupenbild auf dem Titel: Bohdan L / Shutterstock.com, Vitaly Korovin / Shutterstock.com

Themen von der Titelseite sind rot gesetzt.

Leserforum

Diffusor für den LavaFrame

Der LavaFrame, Make 3/23, S. 36

Der Lava Frame ist gebaut und funktioniert einwandfrei. Tatsächlich hat sich als schwierigste Aufgabe das Beschaffen der transluzenten Folie herausgestellt. Diese ist auch als Tiefziehfolie zur Zeit nirgends zu bekommen. Wenn man aber durch Probieren feststellt, dass flexible farbige Schneidbretter mit 1 mm Dicke auch ihren Dienst tun, dann ist das Projekt schnell ein Erfolg. Diese bekommt man z.B. im Tedi zu drei Stück für 2,50 Euro, auch preislich einfach gut. Ein sehr schönes Projekt, welches viel Spaß gemacht hat.

Dirk Hertel

Überzogene Reaktionen

Editorial, Make 7/23, S. 3

Ich denke mir auch manchmal, dass die Reaktionen zu solchen Artikeln höchst überzogen sind. Ich, männlich, 58 Jahre alt, Akademiker, aber ohne formelle Ausbildung als Elektrotechniker, schließe seit ca. 44 Jahren so ziemlich alle elektrischen Geräte, von Lampen, Steckdosen bis hin zu Drehstrom-Herdplatten alles selber an, und die zwei Male, wo ich tatsächlich einen Stromschlag gekriegt habe, waren einmal ein Defekt in einem Aufzug, mit dem ich selber (außer dass ich damit fahren wollte) nichts zu tun hatte, und das zweite mal meine Frau, die eine von mir ausgeschaltete Sicherung ohne Rückfrage mit mir wieder eingeschaltet hat. Das war zwar nicht lustig, aber ich habe es ohne weitere Schäden (denke ich zumindest) überstanden.

Da ja, formal gesehen, der nicht ausgebildete Laie nicht einmal eine neue Lampe anschließen darf (die mit Stecker mal ausgenommen) denke ich, dass die Innung dahinter steckt, die wie Kerberos der Höllenhund darüber wacht, dass auch das letzte Elektron nur dann fließen darf, wenn der ausgebildete und beurkundete Elektriker seinen Obolus für den Dienst des 220V-Litzenanschlusses kassiert hat. Es ist nicht so schwer, Sicherung oder Schutzschalter aus, im Zweifelsfall misst man nach, ein Stromprüfmessgerät oder ein Multimeter sollten so ziemlich alle Make-Leser ja haben, und ein wenig Menschenverstand, dann geht das auch immer gut.

Nicos Charara

VDE-Gläubigkeit

Hi, ihr fragt Euch warum Diskussionen unter Elektrikern oft mit Schaum vorm Mund geschehen. Nun schaut Euch mal die Elektriker-Facebookgruppen an (siehe Link) an. Was da an Diskussionen auftaucht ist für mich schon ziemlich lustig. Für mich liegt das an der Ausbildung der jüngeren Zeit. Da scheint eine VDE-Gläubigkeit anerzogen worden zu sein, die jeder Beschreibung spottet. Ich sehe das bei Elektrikern bei uns, die so ab 1995 ihre Lehre hatten. Die sind deutlich unentspannter als die alten Hasen.

Als ich das Fach lernte, gab es noch DIN, TGL und Physik, sonst nichts. Man war Funkenhascher oder Steckdosenkasper und nicht Elektrikergeselle! Wir haben Litzen verzinkt und Ösen gelötet! Heute ein Sakrileg. Ich bin mir des Unterschiedes durchaus bewusst. Ich bin mir auch bewusst, dass vieles heute sicherer ist und besser. Aber zu oft schaut man heute auf andere Länder und lächelt über deren Normen. Dabei haben wir es noch nicht mal geschafft eine Norm für eine verpolungsfreie Steckdose zu schaffen. In Frankreich und England bei Schuko fast undenkbar. Deshalb leiten bei uns schaltbare Steckdosen aus anderen Ländern den Strom zum Verbraucher nicht ab. Sie haben keinen Doppelschalter. Das wird als gegeben akzeptiert.

Wenn aber ein Laie mit physikalischer Schulbildung und Zusatzwissen in Elektrik eine Steckdose wechselt, schäumt der Elektromeister, der im Zweifelsfall für so einen Auftrag gar nicht rausfährt. Er belächelt Sonoff und Shelly-Enthusiasten auch weil er lieber was Teureres verkauft! Shellys kann man doch nicht supporten, da verdient man ja nichts. Der Größenwahn dieser Gilde wird nur noch von dem in der Politik übertroffen. Das geht dann so weit, dass wir als Bürgerwerkstatt (Repair Café) von Miele keine Kohlehalter für einen Staubsaugermotor bekommen, sondern nur den ganzen Motor, welcher dann teurer ist als der Staubsauger.

Matthias Zwerschek

▶ **make-magazin.de/xnfu**

Wer bastelt, der blutet

Ich habe eben die Make 7/23 aufgeschlagen und die Überschrift German Angst im Blickwinkel erhascht. Das hat mich neugierig gemacht und ich habe mir tatsächlich einmal das Editorial durchgelesen. Ich frage mich, welche Ängste bei einigen Lesern der Make bestehen, dass ein Artikel derartige Beschwerden verursachen kann. Ist es der Versuch seinem Kind schonend beizubringen, dass die Herdplatte heiß ist? Sind es übervorsichtige Wichtigtuer? Wer weiß!

Ich stimme Ihnen in allen Punkten zu und bedanke mich für die ehrlichen Worte. Hoffentlich lassen Sie (und das Team der Make) sich nicht von derartigen Shitstorms davon abbringen uns weiterhin mit spannenden Artikeln zu bereichern. Die Artikel regen die Phantasie an und begeistern zum Ausprobieren neuer Dinge. Die Make ist für Bastler und solche die es werden wollen! Wer bastelt, der blutet auch schon mal!

Helge Korz

Warnungen ausreichend

Nach dem Lesen des Vorwortes und des Leserforums muss ich Ihnen nun meine Bewunderung aussprechen! Scheinbar gibt es eine Vielzahl an Besserwissern und Schlaumeier, welche glauben, dass nur ihre Meinung die Wahre ist. Anstatt ein gemeinsames Lernen von- und miteinander, wird die eigene Meinung als der Stein des Weisen dargestellt. Auch die Wortwahl sagt oft mehr über den Schreiber, als über das

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.

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.

Thema. Daher meine Bewunderung für eure Redaktion für den professionellen Umgang mit solchen Wortmeldungen!

Aus meiner Sicht ist die Make ein Magazin für Leute, welche sich für das Thema Elektronik, Basteln und sonstige „digitalen" Themen interessieren. Die Warnhinweise sind absolut ausreichend und jeder kann/muss eigenverantwortlich(!) mit den Informationen umgehen. Wenn man eine bessere Lösung hat, als die gezeigte, würde sich jeder freuen, wenn man dies als Optimierungsvorschlag oder Gedankenanregung einbringt. Wenn man mit etwas nicht einverstanden ist (Titelbild, ...) kann man dies als Hinweis nehmen, dass man es selber anders lösen würde. Solches Vorgehen wäre verbindend und Sie müssten nicht soviel Diplomatie walten lassen. Vielen Dank für die immer wieder sehr interessanten Beiträge und Anregungen.

Fritz Bischoff

Never give up!

Editorial, Make 6/23, S. 3

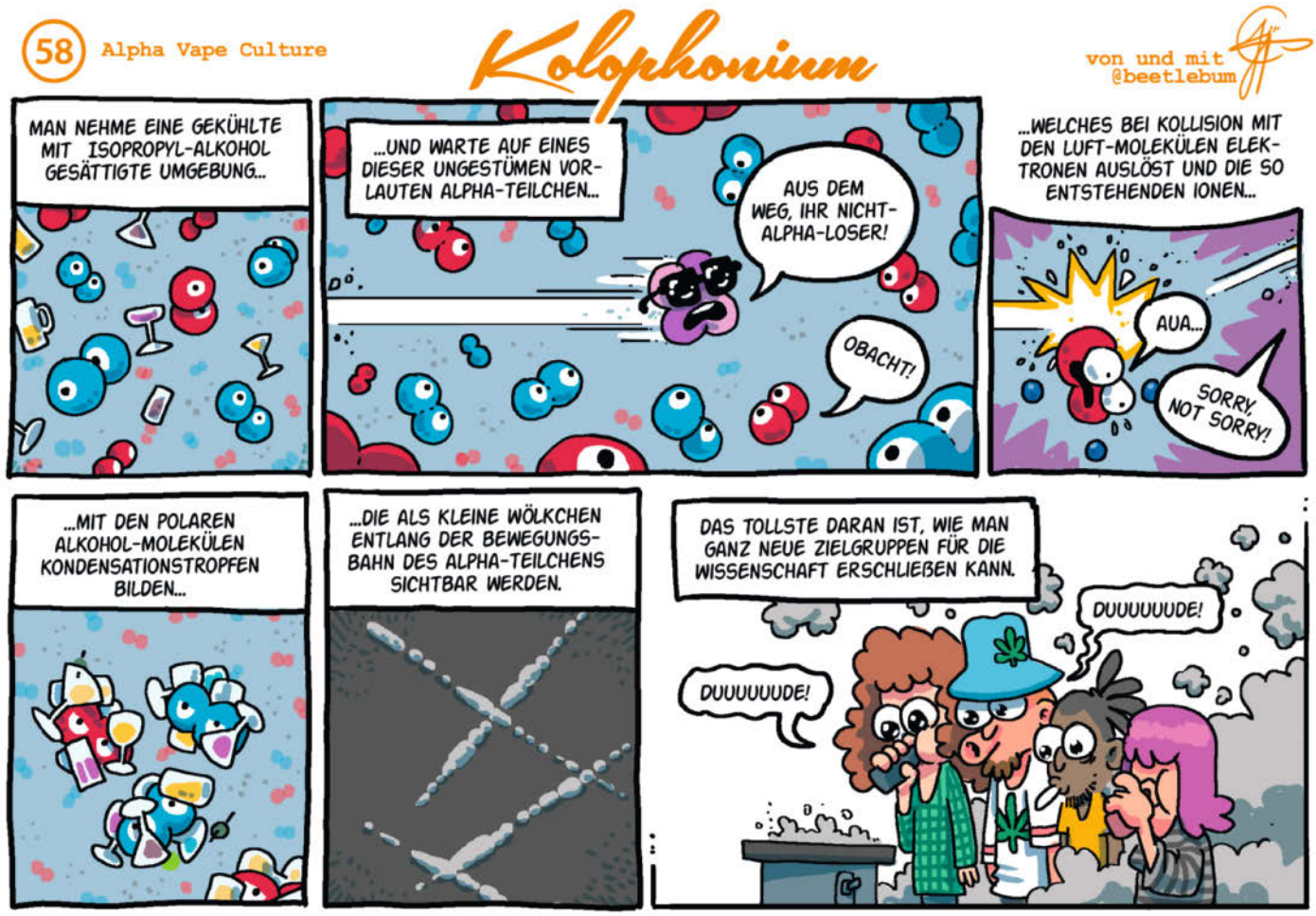
Das kann ich nur unterstreichen, mittlerweile bin ich 70 Jahre alt, aber schon in meiner Jugend habe ich gelernt, wenn man sich informiert und vorsichtig vorgeht, kann man sehr vieles reparieren. Als gelernter Maschinenschlosser und späterer Maschinenbauingenieur habe ich immer versucht, über den Tellerrand hinaus zu schauen, habe schon während meines Studiums noch einen Abendkurs „Industrielle Elektronik" besucht. Dies hat mir immer geholfen das verbreitete Spartendenken: Mechanik - Elektrik zu überwinden.

Habe meine Freunde und deren Fähigkeiten beobachtet, der eine hatte eine enorme Fähigkeit auf dem Bereich der Elektrik/Elektronik, er reparierte bereits im ganzen Dorf die damaligen Röhrenfernseher, bevor er überhaupt eine Ausbildung zum Radio- und Fernsehmechaniker angefangen hatte. Später lötete er eigenständig Transistoren, Dioden und mehr zusammen, erläuterte uns Vorstufe und Endstufe, warum dieser Transistor besser als jener geeignet sei … und mehr.

Irgendwann stand er mit Panik in den Augen vor meiner Tür und fragte:„Kann ich ein paar Sachen bei dir abstellen?" Es war ein kompletter selbstgebauter UKW-Sender mit der teuren Senderröhre QQE 03/12, sie kostetet damals 30 DM – das war ein halber Monatslohn von seiner Ausbildungsvergütung 1970.

Aber wenn ein Skalenseil gerissen war und er ein neues aufziehen musste, kam er völlig verzweifelt zu mir, er wusste, das war für mich kein Problem. Deswegen habe ich in meinem gesamten Berufsleben immer wieder versucht, mir die elektrische/elektronische Seite zu erschließen und daher habe ich das Make-Sonderheft 2023 gekauft. Also: Never give up!

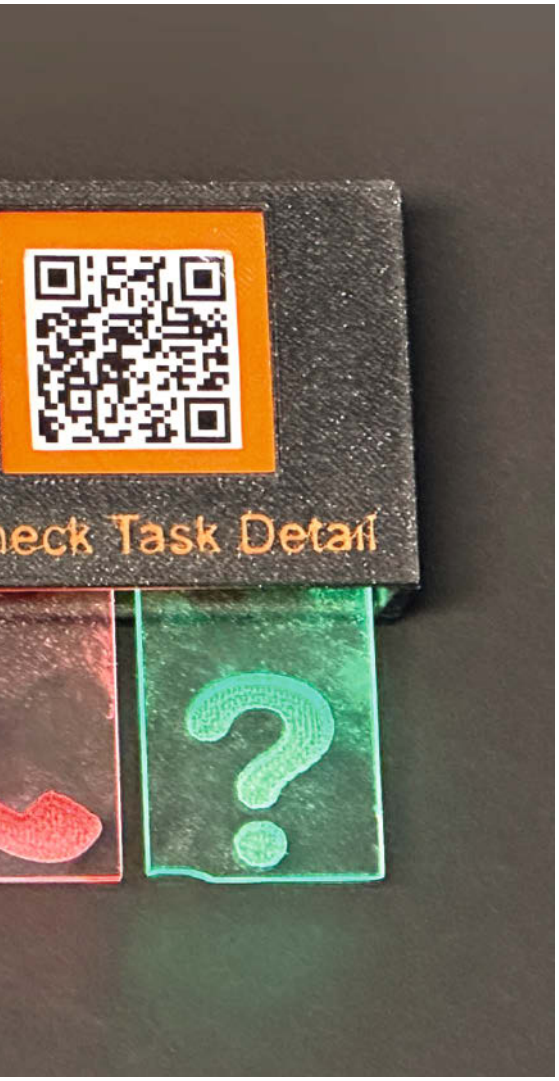
K.H. Rathert



Task-Reminder

Aufgaben, die man nicht im Blick hat, können schnell mal unter den Tisch fallen. Den blinkenden Task-Reminder kann man jedoch kaum übersehen. Er weiß genau, welche Aufgabe als Nächstes ansteht und leuchtet unermüdlich, bis man sie erledigt hat.

von Bernd Heisterkamp



Kurzinfo

» Aufgaben im Blick behalten mit NeoPixeln, ESP32 und MicroPython
» Google-Sheets-Tabellen über Formulare befüllen und filtern
» Mit Apps Script von außen mit Tabellen interagieren

Checkliste

Zeitaufwand: 5 bis 8 Stunden

Kosten: ca. 25 Euro

Material

» **ESP32 ePulse Feather**
» **NeoPixel-LED-Streifen mit 10 LEDs,** 60 LEDs/m, 5V
» **5 Taster**
» **5 1kΩ-Widerstände**
» **5 100nF-Kondensatoren**
» **Platinen-Rohmaterial**
» **Header,** einreihig
» **Kabel, Schrauben und Unterlegscheiben**

Werkzeug

» **3D-Drucker**
» **CNC-Fräse (für die Platine)**
» **Lötkolben**
» **Seitenschneider**

Mehr zum Thema

» Bernd Heisterkamp, Smarte Werkstattboxen, Make 5/23, S. 38
» Gustav Wostrack, Der Weg zur Platine, Teil 1, Make 6/21, S. 104
» Carsten Wartmann, Disco is back – Lichtshow mit WLED, Make 2/23, S. 8

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

Es gibt gewisse Aufgaben, die man trotz Outlook und Handys immer wieder vergisst, insbesondere, wenn die Familie oder die FabLab-Mitglieder nicht den gleichen Kalender teilen. Um dieses Problem zu lösen, habe ich meinen Familien-Tagesplaner aus dem Make IoT-Special 2016 als Task-Reminder neu aufgelegt. Dieser zeigt an, wer als Nächstes mit einer wichtigen Aufgabe an der Reihe ist, indem er mithilfe von jeweils zwei NeoPixel-LEDs fünf Icons in unterschiedlichen Farben (je nach Person) leuchten oder blinken lässt. Die Icons selbst geben einen Hinweis darauf, um welche Aufgabe es sich handelt. Weitere Details liefert ein Webserver, der auf dem ESP32 läuft, der die Elektronik steuert.

Wie man den Task-Reminder nachbaut und wie die Komponenten im Detail zusammenhängen, zeige ich in diesem Artikel. Im GitHub-Repository des Projekts (siehe Link in Kurzinfo) befinden sich die benötigten Tabellenformeln, Programmcodes und ein Platinenlayout.

Das Prinzip

Damit mehrere Personen Aufgaben unkompliziert und unabhängig voneinander erstellen können, nutze ich für den Task-Reminder eine Google-Tabelle (Google Sheet), die man über ein Formular mit Aufgaben füttern kann (Bild 1). Eine zweite Tabelle prüft anschließend, ob eine Aufgabe heute erledigt werden muss. Diesen Status ruft ein ESP32 über ein MicroPython-Programm einmal pro Minute ab und lässt die entsprechenden Icons aufleuchten. Drückt man eines der Icons, löst man über einen darunter befindlichen Taster einen Interrupt aus und markiert die Aufgabe als erledigt. Daraufhin schickt der ESP32 den neuen Status an das Google Sheet und die LEDs erlöschen.

Da der ESP32 über URLs mit der Aufgabenliste kommuniziert, lässt sich das System auch mit weiteren Schnittstellen erweitern. Man kann etwa andere ESPs einbinden, die mithilfe von Sensordaten automatisch Aufgaben erzeugen, sodass der Task-Reminder sich auch melden kann, wenn eine Pflanze dringend Wasser benötigt.

Tabelle und Formular anlegen

Damit man nicht aus Versehen private Tabellen über den Task-Reminder freigibt, empfiehlt es sich, als Erstes einen eigenen Google-Account für dieses Projekt anzulegen. Danach erstellt man eine neue Tabelle und ändert im Menü Datei/Einstellungen die Anzeigesprache auf Englisch (ganz unten im Einstellungsfenster), damit die Vorlagen und Funktionen, die ich für diesen Artikel erstellt habe, richtig funktionieren. Zudem muss man noch die automatische Neuberechnung auf minütlich einstellen. Im nächsten Schritt fügt man über das Menü Tools ein Formular hinzu (Bild 2), über das man neue Aufgaben hinzufügen wird. Es sollte folgende Felder enthalten:

– Aufgabe: (Feldtyp: Short answer),
– Fällig am: (Feldtyp: Date),
– Fällig um: (Feldtyp: Time, optional),
– Icon: (Feldtyp: Multiple-Choice, nach dem Schema 1-Lueften, 2-Giessen oder 3-Recycling),
– Farbe: (Feldtyp: Multiple-Choice, z.B. red (Thomas), green (Melanie) oder blue (Fritz)).

Die Ziffern vor den jeweiligen Aufgabenkategorien im Feld Icon sind notwendig, damit das System die Icons später richtig zuordnen kann. Ebenso stehen standardmäßig red, green, blue und yellow als Farben zur Auswahl, da diese als Variablen im MicroPython-Programm deklariert sind. Sofern man eigene Farben verwenden möchte, muss man nur darauf achten, dass sie im Formular und im Programm übereinstimmen.

Hat man die Felder konfiguriert, wählt man in dem Reiter Settings bei „Responses/Collect E-Mail Addresses“ den Punkt Verified aus. So

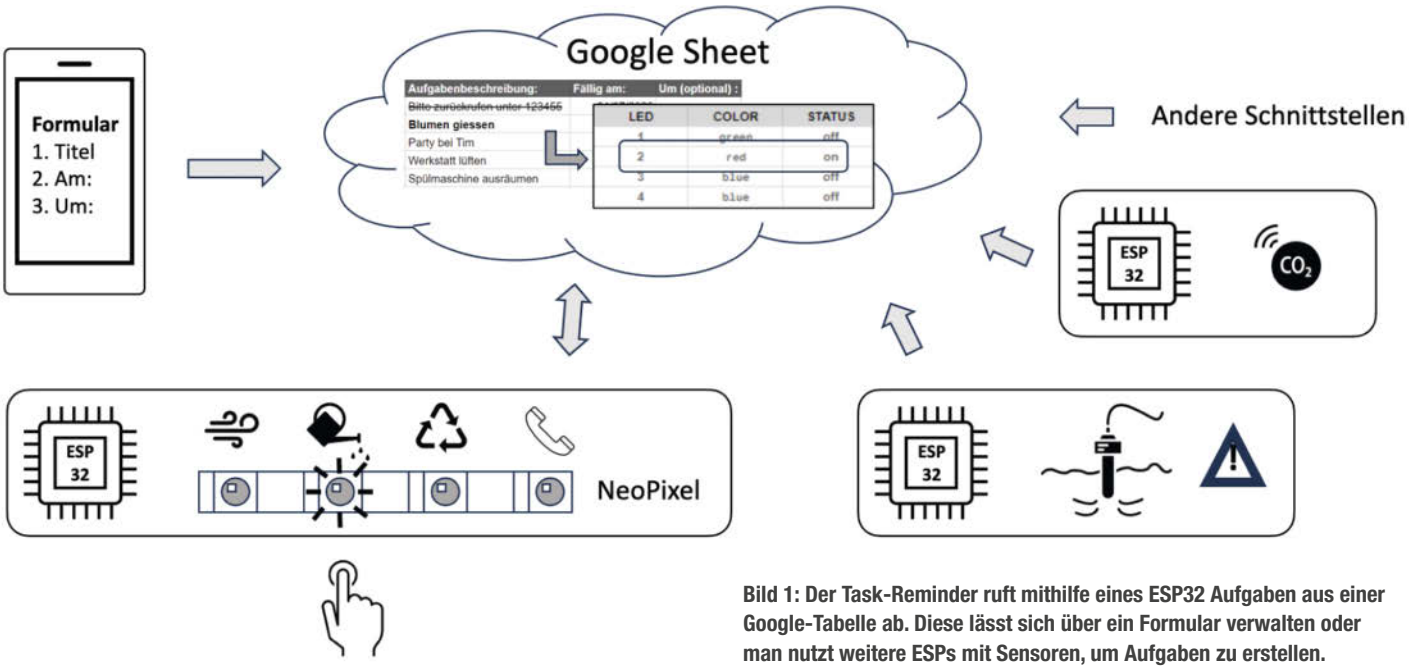


Bild 1: Der Task-Reminder ruft mithilfe eines ESP32 Aufgaben aus einer Google-Tabelle ab. Diese lässt sich über ein Formular verwalten oder man nutzt weitere ESPs mit Sensoren, um Aufgaben zu erstellen.

sieht man später, wer den Task erstellt hat. Über das Augensymbol am oberen Bildschirmrand kann man das Formular ausprobieren. Füllt man dieses aus und sendet es ab, erstellt Google automatisch ein neues Tabellenblatt für die Antworten. Dieses benennt man in Aufgabenliste um. Das Tabellenblatt 1 kann man löschen.

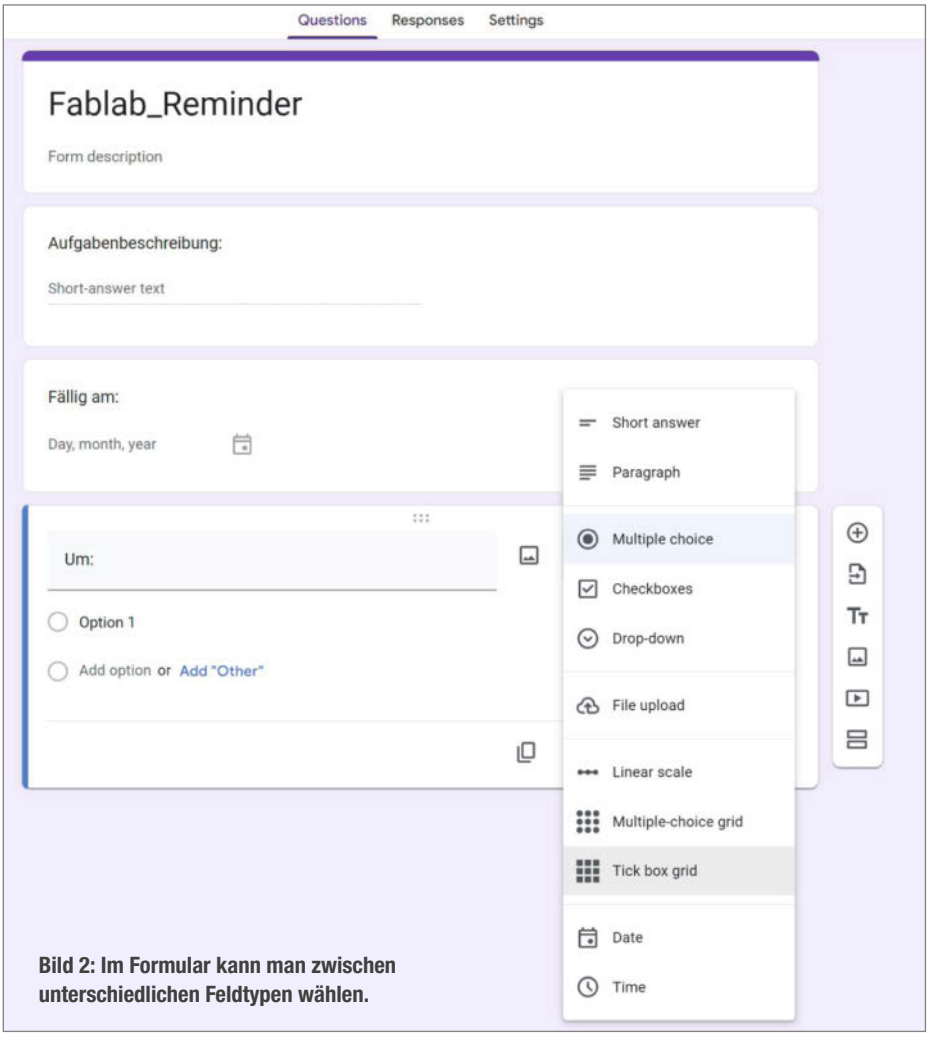


Bild 2: Im Formular kann man zwischen unterschiedlichen Feldtypen wählen.

Die Aufgabenliste

Anders als Excel erlaubt Google Sheets sogenannte Array-Formulas. Das sind Formeln, die man in der ersten Zeile einer Spalte einträgt, die aber für die gesamte Spalte gültig sind. Diese Funktion verwende ich in der Aufgabenliste, um automatisch zu ermitteln, ob eine Aufgabe bereits erledigt wurde oder noch aussteht. Dazu muss man die Tabelle zunächst um drei Spalten zwischen Zeitstempel und Aufgabenbeschreibung ergänzen, wie in Bild 3 zu sehen, und am Ende zwei weitere Spalten (Task_Status und WatchDog) hinzufügen. Anschließend kopiert man in die Felder C1, D1 und E1 die Formeln aus der Datei Google_Aufgabenliste_Formeln.txt. Diese haben folgende Aufgaben:

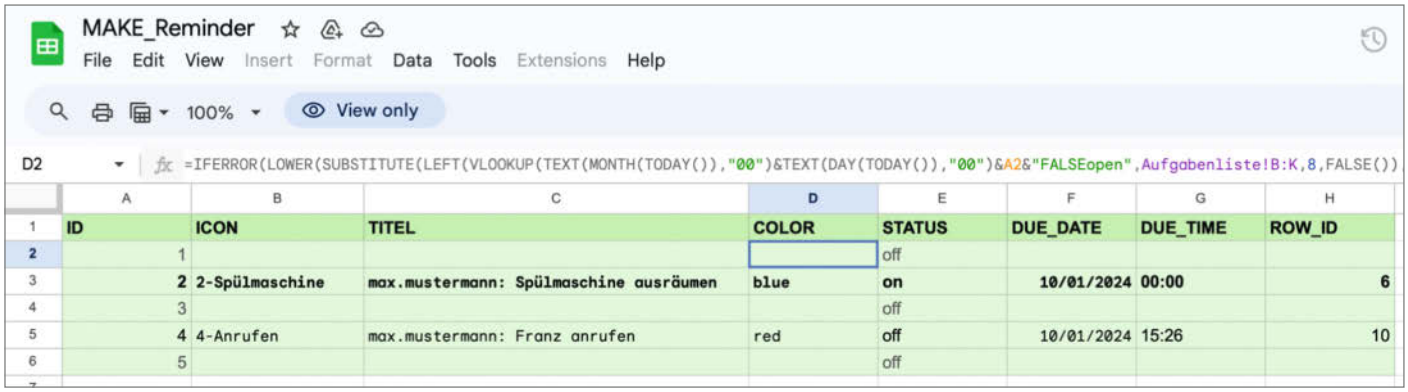
- Die Formel in C1 prüft, ob das Datum in Spalte F vor dem aktuellen Datum liegt. Wenn ja, wird der Wert in Spalte C auf TRUE gesetzt, ansonsten auf FALSE.
- Die Formel in B1 erzeugt einen Referenzwert, den man im zweiten Tabellenblatt (dazu gleich mehr) benötigt, um offene Aufgaben zu finden. Dieser besteht aus dem Fälligkeitsdatum einer Aufgabe, der Icon-Nummer und dem aktuellen Status.
- In D1 erzeugt die Formel eine Kombination aus E-Mail-Adresse und Aufgabenbeschreibung. Das Resultat kann man später über den Webserver abrufen.

Offene Aufgaben finden

Weiter geht's mit einem neuen Tabellenblatt, das man in LED_Status umbenennt (Bild 4). In dieser werden Aufgaben angezeigt, die heute erledigt werden müssen. Hierzu verwende ich die Funktion VLOOKUP, die im Tabellenblatt Aufgabenliste in Spalte B nach

	A	B	C	D	E	F	G	H	I	J	K	L
1	Timestamp	Lookup	Past	Summary	Aufgabenbeschreibung:	Fällig am:	Um (optional) :	Icon:	Farbe:	Email address	Task_Status	WatchDog
2												
3												

Bild 3: Das Tabellenblatt Aufgabenliste muss dieser Anordnung entsprechen.



	A	B	C	D	E	F	G	H
1	ID	ICON	TITEL	COLOR	STATUS	DUE_DATE	DUE_TIME	ROW_ID
2	1				off			
3	2	2-Spülmaschine	max.mustermann: Spülmaschine ausräumen	blue	on	10/01/2024	00:00	6
4	3				off			
5	4	4-Anrufen	max.mustermann: Franz anrufen	red	off	10/01/2024	15:26	10
6	5				off			

Bild 4: Die Funktionen in dem Tabellenblatt LED_Status suchen sich Aufgaben aus der Aufgabenliste.

Referenzen sucht, die dem heutigen Datum entsprechen und bisher nicht erledigt sind. Die Formeln für die zweite Zeile habe ich in der Datei Google_LED-Status_Formeln.txt hinterlegt. Diese muss man mit entsprechend angepasster Zeilenreferenz auch in die Zeilen 3 bis 6 kopieren. In der Spalte DUE_DATE stellt man anschließend noch über das Menü das Format Date ein.

Hat man alles richtig gemacht, zeigt die Tabelle LED_Status nur diejenigen Aufgaben an, die heute fällig sind. Die übrigen Zeilen bleiben leer, bis auf die STATUS-Spalte, in der „off" steht. Ist eine Aufgabe heute fällig, aber die vorgewählte Uhrzeit noch nicht erreicht, erscheint die Aufgabe zwar in der Tabelle, aber der STATUS steht auf „off".

Zugriff über Apps Script

Um das Google Sheet von außen – also auch mit dem ESP32 – abfragen und verändern zu können, benötigt man Google Apps Script, eine webbasierte JavaScript-Plattform. Dazu klickt man im Menü Extensions auf „Apps Script". Über die linke Seitenleiste wählt man anschließend den Punkt Editor aus und sieht unter Files bereits ein leeres Skript. Damit der ESP32 die Tabellendaten lesen und entsprechend die LEDs ansteuern kann, füllt man das Skript mit dem Code aus der Datei Apps_Script_DoGet.txt und nennt es DownloadDataInJSONformat.gs. Der Code beinhaltet die Funktion `DoGet` (siehe Listing), der einen HTTP-Endpoint erzeugt. Dieser gibt die Daten des Tabellenblatts LED_Status in einem JSON-Format zurück.

Als Nächstes erstellt man über das Plus-Icon ein weiteres Skript, kopiert den Code aus Apps_Script_DoPost.txt hinein und nennt es UploadDataInJSONformat.gs (Bild 5). Das im Code befindliche `DoPost` erlaubt es, das Tabellenblatt Aufgabenliste mithilfe einer JSON-Tabelle um neue Aufgaben zu ergänzen. Dadurch kann der ESP32 später auch Aufgaben

Apps Script: DoGet

```
function doGet() {
  return fetch_led_status();
}

function fetch_led_status() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("LED_Status");
  var range = sheet.getRange("A1:H6");
  var values = range.getValues();
  var headers = values[0];
  var led_status = values.slice(1);
  // Convert data into JSON format
  var jsonData = [];
  for (var i = 0; i < led_status.length; i++) {
    var row = {};
    for (var j = 0; j < headers.length; j++) {
      row[headers[j]] = led_status[i][j];
    }
    jsonData.push(row);
  }
  return ContentService.createTextOutput(JSON.stringify(jsonData)).setMimeType(ContentService.MimeType.JSON);
}
```

Apps Script: DoPost

```
function doPost(e) {
  var data = JSON.parse(e.postData.contents);
  var sheetName = "Aufgabenliste";
  var StartColumn = 5;
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName(sheetName);
  var headers = sheet.getRange(1, StartColumn, 1, sheet.getLastColumn()).getValues()[0];
  var newRowValues = headers.map(function(header) {
    return data[header] || '';
  });
  // Prüfen, ob eine ROW_ID in den JSON-Daten existiert
  if (data.hasOwnProperty('ROW_ID') && data['ROW_ID'] !== '') {
    var rowId = parseInt(data['ROW_ID']);
    var currentRowValues = sheet.getRange(rowId, StartColumn, 1, newRowValues.length).getValues()[0];
    for (var i = 0; i < newRowValues.length; i++) {
      if (newRowValues[i] !== '') {
        currentRowValues[i] = newRowValues[i];
      }
    }
    // Die Zeile aktualisieren
    sheet.getRange(rowId, StartColumn, 1, currentRowValues.length).setValues([currentRowValues]);
    return ContentService.createTextOutput("Row updated with values: " + JSON.stringify(currentRowValues));
  } else {
    // oder eine neue Zeile erstellen
    var lastRow = sheet.getLastRow();
    sheet.getRange(lastRow + 1, StartColumn, 1, newRowValues.length).setValues([newRowValues]);
    return ContentService.createTextOutput("New row added with values: " + JSON.stringify(newRowValues));
  }
}
```

als erledigt markieren, indem er den Begriff `done` an die Spalte Task_Status schickt. Der Wert `StartColumn = 5` stellt sicher, dass man nicht aus Versehen die Spalten zwei bis vier überschreibt, in denen sich die Array-Formulas befinden.

Skripte bereitstellen

Damit man beide Skripte verwenden kann, muss man sie separat bereitstellen (engl. deploy). Dies geht über das „Deploy/New deployment"-Icon oben rechts. Hier wählt man als Typ zunächst „Web app" aus und dass jeder (anyone) auf diese zugreifen darf. Ansonsten kann der ESP32 nicht mit der Schnittstelle kommunizieren (Bild 6). Das Deployment wird noch von etlichen Nachfragen begleitet, mit denen Google sicherstellen will, dass man wirklich weiß, was man tut. Da wir hier mit einem neuen Account arbeiten und (hoffentlich) keine sensitiven Daten speichern, ist das Risiko eines Hacks wohl akzeptabel. Am Ende des Deployment-Prozesses erhält man für jedes Skript eine „Web app"-URL (Bild 7). Diese muss man später im MicroPython-Programm einsetzen. Allerdings kann man die Schnittstelle bereits testen, indem man die URL im Browser einfügt. Als Ergebnis erhält man eine JSON-Liste des LED_Status-Tabellenblatts.

Elektronik

Weiter geht's mit der Verschaltung des ESP32 (Bild 8). Das ePulse Feather wird später über den USB-C-Port dauerhaft mit Strom versorgt. Dieser reicht auch für die benötigten NeoPixel aus. Nutzt man ein Board mit einem Micro-USB-Anschluss, sollte man die NeoPixel separat mit Strom versorgen. Die Datenleitung der Neopixel liegt am GPIO-Pin 33 an.

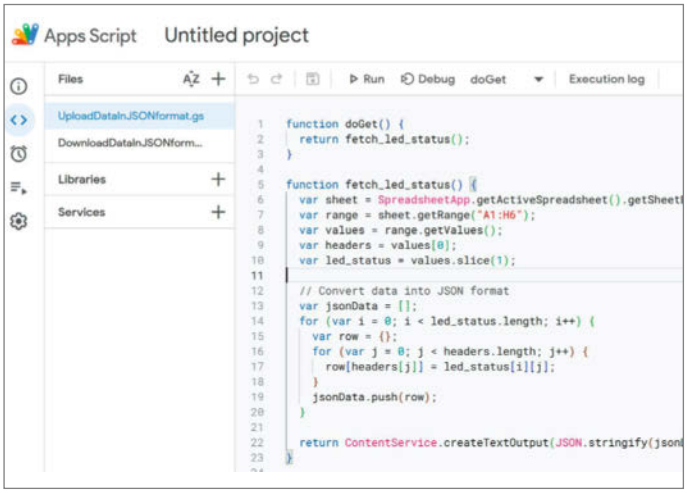


Bild 5: Google Apps Script ermöglicht es, Google-Tabellen von außen abzufragen und zu bearbeiten.

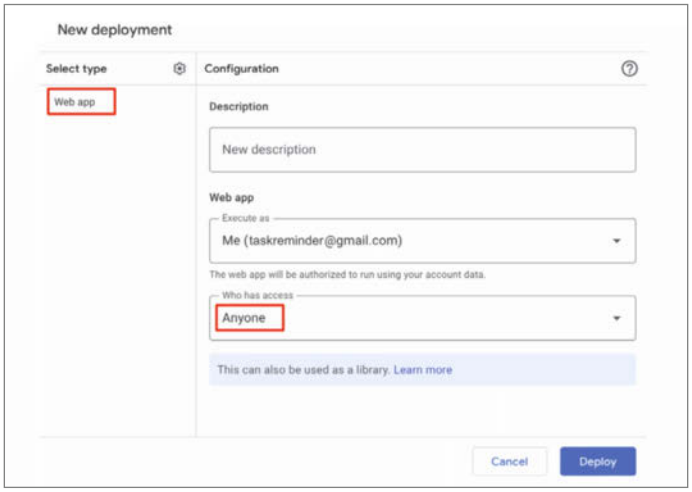


Bild 6: Damit man weniger Zugriffsrechte verwalten muss, darf jeder die Aufgabenliste ergänzen.

Damit die Taster nicht prellen, habe ich kleine 100nF-Kondensatoren verwendet. Die Vorwiderstände mit 1kΩ vermeiden Stromspitzen bei der Ladung der entleerten Kondensatoren. Ich empfehle, die Schaltung mit einem einzelnen Taster und den NeoPixeln auf einem Steckbrett aufzubauen und dann bereits den MicroPython-Code zu testen, bevor man sich an die Arbeit macht und die Platine erstellt.

Der MicroPython-Code

Im Internet findet man diverse Artikel zur Einrichtung von MicroPython auf dem ESP32. Im Folgenden führe ich daher nur die wichtigsten Schritte auf:

1. Eine MicroPython-IDE auf dem Computer installieren. Für dieses Projekt verwende ich die Thonny IDE.
2. Die MicroPython-Firmware mithilfe des Programms esptool.py oder direkt in Thonny flashen.
3. Die drei Programme boot.py, secrets.py und main.py mit Thonny auf den ESP32 kopieren.
4. In secrets.py die Netzwerk-SSID(s) und Passwörter anpassen und speichern.
5. Die „Web app"-URLs für den Datenup- und download in den Zeilen 15 und 16 in main.py einfügen und speichern.

Die Datei boot.py wird nach jedem Neustart des ESP32 einmal ausgeführt. Sie verwendet die Passwörter aus der Datei secrets.py, um eine WLAN-Netzwerkverbindung herzustellen. Anschließend startet das Hauptprogramm main.py.

Die dort eingestellte Pin-Belegung in den Zeilen 19 (für die Taster) und 38 (NeoPixel) muss man nur anpassen, wenn man ein anderes ESP32-Board als das ePulse Feather nutzt oder wenn man andere Pins belegen möchte. Die Variablen und ihre Werte in den Zeilen 24 bis 34 sollte man nicht verändern, da ansonsten auch Begriffe in der Tabelle angepasst werden müssen.

Der Programm-Code ist weitgehend kommentiert. Die Webseite, die angezeigt wird, wenn man die IP-Adresse des ESP32 über den Standard-Port 80 öffnet, ist in den Zeilen 78 bis 123 definiert. Die Zeilen 225 und 226 initialisieren den Timer, der die LEDs blinken lässt. Sollen die LEDs konstant leuchten, kann man diese Zeilen auch auskommentieren.

Im Hauptteil ab Zeile 237 durchläuft das Programm permanent folgende Schritte:

– Zuerst prüft es, ob es einen Taster-Interrupt gegeben hat. Wenn ja, wird das Ergebnis des entsprechenden Tasters auf `done` gesetzt, was die Aufgabe im Google Sheet als erledigt markiert. Hierfür wird in der Funktion `call_google_apps_script` ein HTTP-Post-Befehl an den ESP32 geschickt.
– Danach stellt das Programm fest, ob das Zeitintervall zur Abfrage des LED-Status im Google Sheet bereits überschritten wurde oder ob es seit dem letzten Aufruf einen Interrupt gegeben hat. Wenn dem so ist, wird über die Funktion `fetch_led_status` der aktuelle Zustand der LEDs abgefragt und die NeoPixel werden entsprechend mit `set_neopixel_color` angesteuert.
– Als Letztes wird geprüft, ob jemand die Website aufgerufen hat, die entsprechend angezeigt wird.

Platine und Gehäuse

Funktioniert soweit alles, kann man die Platine aufbauen. Ich habe ihr Layout so gestaltet, dass man sie auch gut durch Isolationsfräsen erstellen kann (Bild 9). Der ESP32 sitzt etwas höher auf den Lötstiften der Platine. So kann man darunter noch die Keramikkondensatoren unterbringen (Bild 10). Diese sollte man nach dem Löten aber flach auf die Platine drücken.

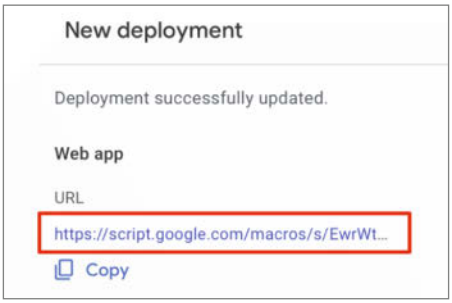


Bild 7: Diese „Web app"-URLs benötigt man zur Kommunikation mit der Google-Tabelle.

Das Gehäuse ist komplett aus PLA gedruckt. Der Deckel besteht aus zwei Teilen, die man später aufeinander klebt. So kann man auf Stützen beim Drucken verzichten. Bei der Oberseite des Deckels habe ich einen manuellen Farbwechsel vorgenommen, um den Text andersfarbig zu drucken. Ihr findet die STL- und STEP-Files im GitHub-Repository des Projekts. Die Icons habe ich aus 3 mm dickem Plexiglas gefräst. Aber man kann diese auch mit einem Laser schneiden, je nachdem, was man (oder das lokale FabLab) an Geräten zur Verfügung hat.

Den NeoPixel-Streifen setzt man vertikal in den Gehäuseboden ein, sodass die LEDs von oben in die Plexiglas-Plättchen leuchten (Bild 11). Diese sind nur am oberen Ende befestigt und liegen direkt über den Tastern. Drückt man auf das Icon am unteren Ende, wird dadurch der Taster betätigt. Der Erdungspin der Taster wird über Silberdraht durch alle Taster durchgeschleift und zusammen mit der GND-Verbindung des NeoPixel-Streifens auf der Platine an J1 mit einer Steckhülse befestigt. Die anderen Kontaktseiten der fünf Taster verbindet man über Kabellitzen mit den Verbindern J3 sowie mit den Pins 3 und 4 von J4. Zuletzt muss man noch den 5V-Eingang des

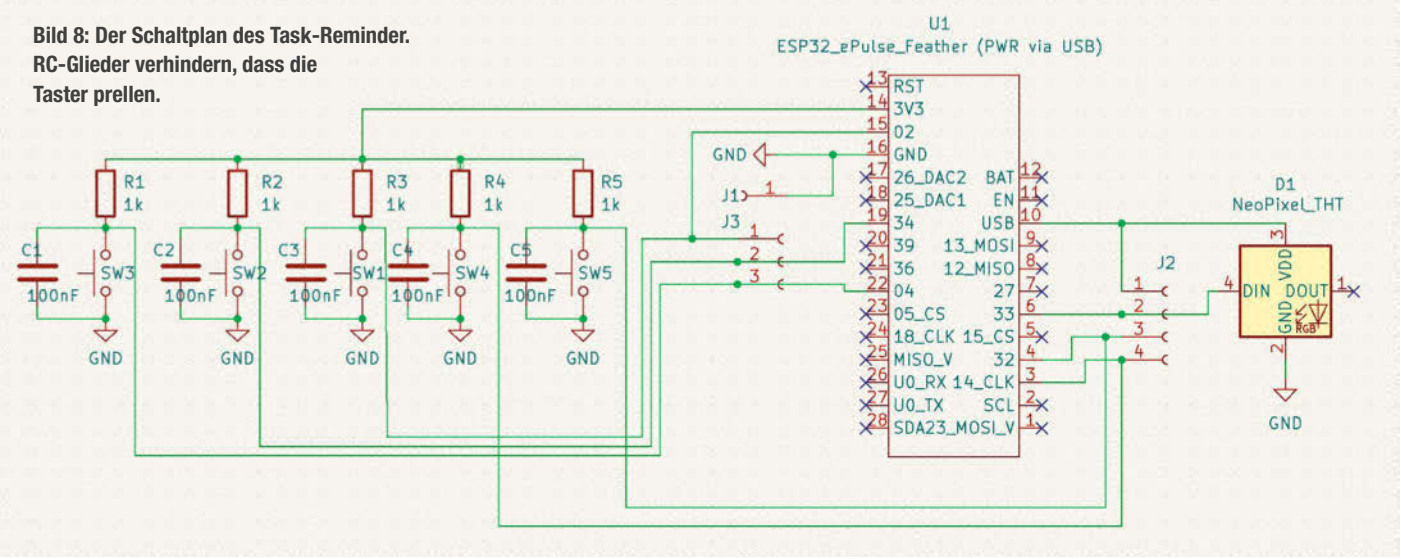


Bild 8: Der Schaltplan des Task-Reminder. RC-Glieder verhindern, dass die Taster prellen.

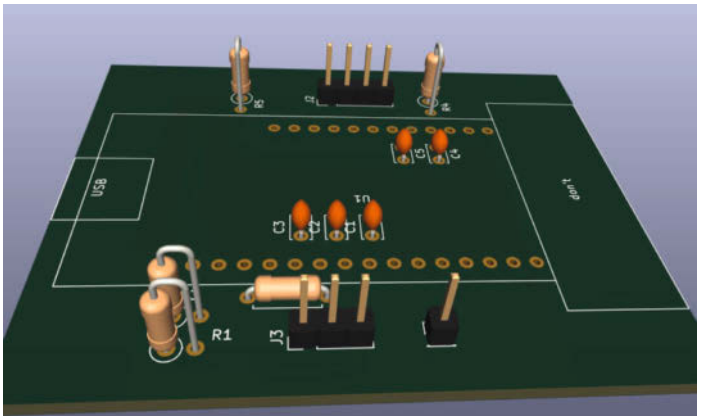


Bild 10: Der ESP32 steckt später über den Keramikkondensatoren.

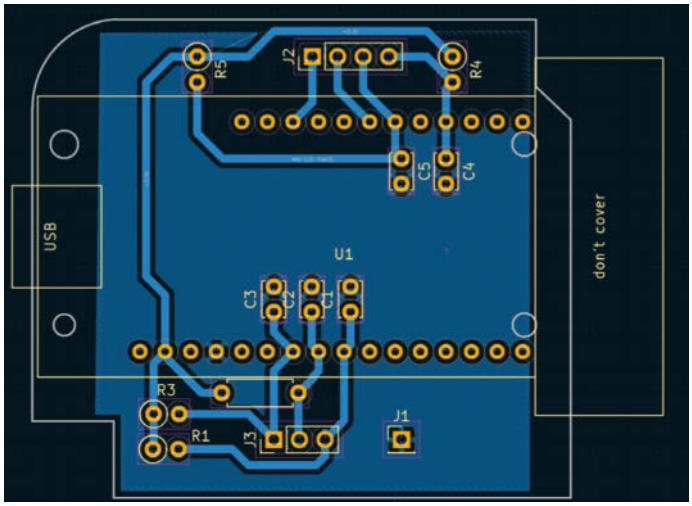


Bild 9: Das einseitige THT-Platinenlayout lässt sich auch fräsen.

NeoPixel-Streifens mit Pin 1 und den Dateneingang mit Pin 2 auf J2 verbinden. Somit ist die Zentrale des Reminders fertiggestellt.

Erweiterungen und Ausblick

Wie eingangs erwähnt, kann man weitere Sensoren in den Task Reminder integrieren. Das dafür benötigte Programm befindet sich ebenfalls in dem GitHub-Repository des Projekts. Mit ihm wird ein ESP32 zu einem Sensor-Satelliten, der die Tabelle mit Aufgaben bestücken kann. In den Zeilen 72 und 73 wird dafür die bereits bekannte Funktion `call_google_apps_script` aufgerufen. Wenn man den Mikrocontroller um einen Sensor erweitert, muss man diesen natürlich noch entsprechend in den Code einbinden. Alternativ kann man auch darüber nachdenken, als Erweiterung einen Dienst zu verwenden, dessen Daten man über das Internet bezieht, z.B. einen Wetterdienst. Dann kann der Task-Reminder je nach Wetterlage signalisieren, dass man etwas erledigen muss. *—akf*

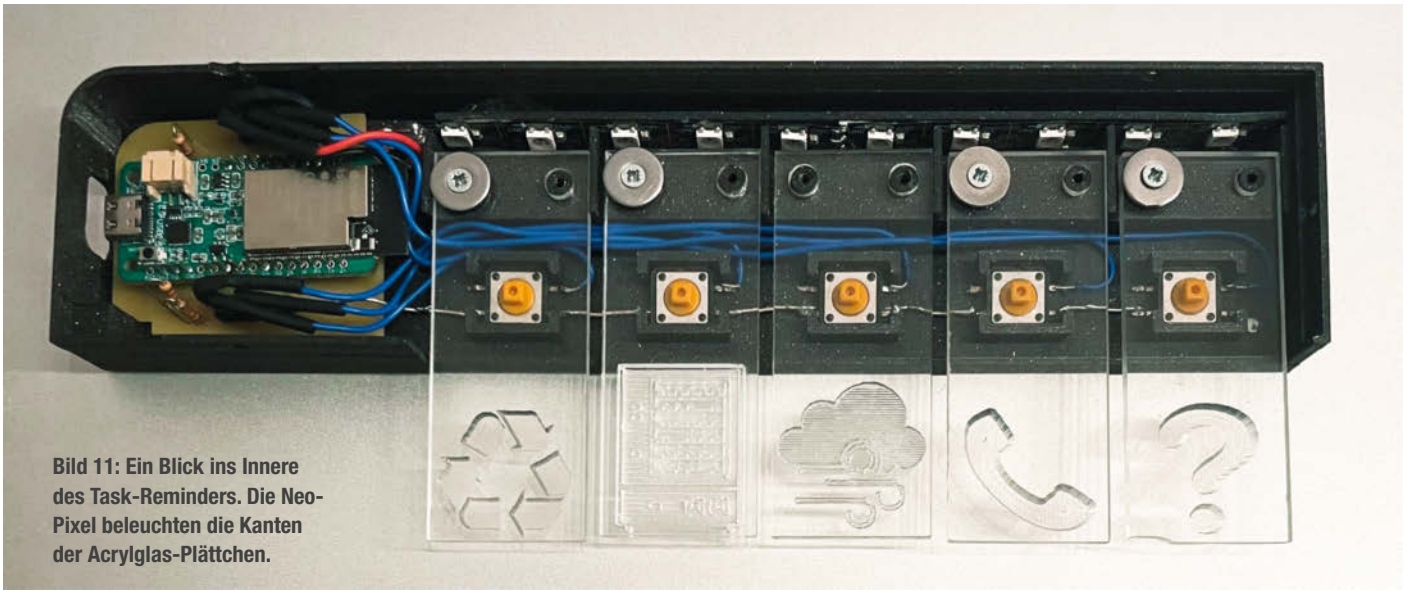


Bild 11: Ein Blick ins Innere des Task-Reminders. Die NeoPixel beleuchten die Kanten der Acrylglas-Plättchen.

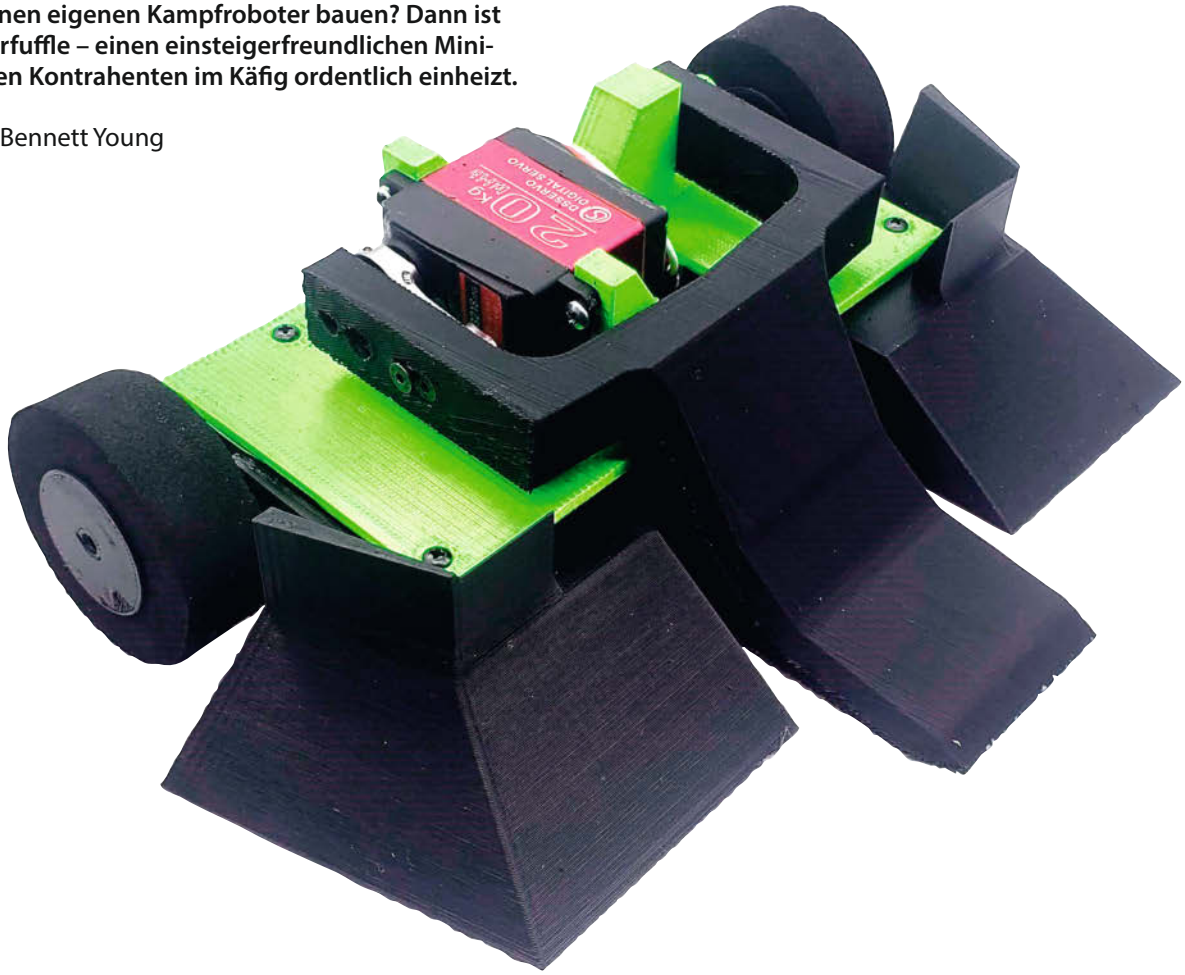
Main Loop des Sensor-Satelliten

```
while True:
    moisture = read_soil_moisture()
    print("Soil Moisture: {:.2f}%".format(moisture))
    if (moisture > 50):
        # Erstelle neue Aufgabe in Aufgabenliste mit folgenden Werten:
        WriteResponse = call_google_apps_script({ HEADER_ICON: "3-Giessen", HEADER_TITEL: "Rosen giessen",
                                                  HEADER_DATE: Today_as_String(), HEADER_EMAIL: "ESP32@Rosen",
HEADER_COLOR: "blue"})
        print(WriteResponse.text)
        time.sleep(60*60*12) # Aufgabe wurde erstellt, warte 12h vor der nächsten Messung
    time.sleep(3)
```

DIY-Kampfroboter

Wollen Sie einen eigenen Kampfroboter bauen? Dann ist es Zeit für Kerfuffle – einen einsteigerfreundlichen Mini-Bot, der seinen Kontrahenten im Käfig ordentlich einheizt.

von Brandon Bennett Young



Wenn Sie mal die Sendung BattleBots (Discovery Channel oder Comedy Central) gesehen haben, dann haben Sie schon eine gute Vorstellung davon, wie solche Roboter funktionieren. Kerfuffle ist ein Roboter in der Kunststoff-Ameisengewichtsklasse. Das bedeutet, er ist nicht aus Metallen oder Kunststoffen wie Stahl oder Nylon gefertigt, wie man das bei den Schwergewichtsrobotern im Fernsehen sieht.

Kerfuffle ist als Einstiegsroboter konzipiert und verwendet preiswerte, 3D-druckbare Materialien, die es vielen Menschen ermöglichen, ihre ersten Schritte in die Welt der Kampfroboter zu machen. Seine Waffen sind keilförmig, um sich unter den Gegner zu schieben, und es gibt einen Hebearm, um ihn umzudrehen. Auch Anfänger können gefahrlos mit ihm üben und sogar gegen andere Kerfuffles kämpfen, ohne dass eine schützende Abgrenzung erforderlich ist.

Ich habe den Roboter ursprünglich entwickelt, um bei einem Wettbewerb meiner Schule zu kämpfen, und er hat sich gut bewährt. Seit der ersten Version im Jahr 2019 wurde Kerfuffle optimiert, um noch wettbewerbsfähiger zu werden, was zur erfolgreichen Version 2 geführt hat, die Sie mit dieser Anleitung bauen können.

Drucken der Teile

Das erste und wichtigste Werkzeug, das Sie für diesen Bau benötigen, ist ein 3D-Drucker. Für den Kerfuffle ist ein Ender 3 mehr als ausreichend, da wir PLA- oder PLA+-Filament verwenden. Diese Sorten sind preiswert und erfordern kein hohes Maß an Tuning zum Drucken.

Neben dem Drucker sollten Sie auch eine Rolle PLA+-Filament in der von Ihnen bevorzugten Farbe kaufen. Normales PLA kann auch verwendet werden, aber ich rate zu dem zäheren PLA+. Als Druckeinstellungen empfehle ich etwa 4 Wände und 50 % Füllung. Diese Einstellungen lassen sich mit der Zeit variieren, wenn Sie mehr Erfahrung mit dem Drucken von Gegenständen und Kampfrobotern haben, aber sie sind ein guter Ausgangspunkt.

Die 3D-Dateien zum Drucken finden Sie über den Link in der Online-Info.

Baue deinen Kerfuffle-Kampfroboter

Sobald die Teile gedruckt und die Komponenten angekommen sind, können wir zum spaßigen Teil übergehen: dem Bau!

Die Abbildung auf S. 17 zeigt eine typische Verdrahtung mit einem Getriebemotor und einem Servo. Aber wir werden unsere Verbindungen löten und auch einen JST-Batterieanschluss hinzufügen.

Die Silver Spark-Motoren haben einen Kontakt in der Nähe eines roten Punktes und einen anderen Kontakt ohne Punkt. Dies entspricht der Polarität des Motors. Jeder tiny-ESC (Nummer 5 in der Abbildung) hat ein lilafarbenes und ein blaues Kabel. Löten Sie zuerst einen dieser Drähte (welcher, ist an dieser Stelle egal) an einen Kontakt und anschließend die andere Farbe an den anderen Kontakt.

Gehen Sie genauso vor, um den anderen tinyESC mit dem anderen Motor zu verbinden, aber vertauschen Sie die Polarität der Drähte. Wenn Sie zum Beispiel den blauen Draht an den Kontakt in der Nähe des roten Punktes des einen Motors gelötet haben, dann löten Sie den violetten Draht an den Kontakt in der Nähe des roten Punktes des anderen Motors. Ich empfehle das Hinzufügen von Schrumpfschlauch, um die Verbindungen vor einem möglichen Kurzschluss zu schützen und die Festigkeit der Verbindung zu erhöhen.

Löten Sie alle roten Drähte von den tinyESCs und dem 9-V-Regler (Nummer 6 in der Abbildung) an einen der Kontakte des FingerTech-Mini-Schalters (Nummer 1 in der Abbildung). An den anderen Kontakt löten Sie den roten Draht der JST-Buchse. Dadurch kann der Stecker des Akkus direkt angeschlossen werden und liefert Strom.

Der Mini-Stromschalter funktioniert durch Anziehen einer Schraube, die die beiden Kontakte miteinander verbindet. Sobald diese Schraube Kontakt hat, ist der Stromkreis geschlossen und der Strom kann fließen. Um sicherzustellen, dass alle Komponenten des Roboters ausgeschaltet werden, sobald der Schalter gelöst wird, verbinden Sie alle roten Drähte an einer Stelle, damit es nur einen Weg für den Stromfluss gibt. Schließen Sie den anderen Kontakt des Schalters an die Batterie an.

Löten Sie alle schwarzen Drähte von den tinyESCs und dem 9-V-Regler an den schwarzen Draht der JST-Buchse. Diese Verbindungen müssen nicht durch den Schalter getrennt werden, da wir die roten Drähte bereits getrennt haben. Durch das Verbinden der schwarzen Drähte wird der Stromkreis geschlossen, so dass nur noch der eine Schalter übrig bleibt, mit dem Sie das Gerät ein- und ausschalten können.

Verbinden von Fernsteuerung und Empfänger

Haben Sie die Fernsteuerung und den Empfänger in einem Bundle gekauft, sind die beiden Geräte meistens schon gepaired.

Wenn Sie manuell eine Verbindung herstellen müssen, gehen Sie wie folgt vor:

1. Die Stromversorgung des Empfängers abstellen.
2. Den mitgelieferten „Transmitter Bind Plug“ (ein Stecker mit einer Kabelschlaufe) in den Eingang „BAT“ des Empfängers stecken.
3. Die Stromversorgung des Empfängers wiederherstellen. Am Empfänger blinkt jetzt eine rote LED.
4. Auf der Fernsteuerung den Knopf „Bind Range Test“ gedrückt halten und die Fernsteuerung einschalten.
5. Das rote Licht am Empfänger leuchtet jetzt durchgehend. Die beiden Geräte sind verbunden.

Kurzinfo

» **Komplett 3D-gedruckt**
» **Fernsteuerbar**
» **Für Anfänger geeignet**

Checkliste

Zeitaufwand:
ein Wochenende

Kosten:
300 bis 400 Euro

Material

» **2 FingerTech Silver Spark** 22.2:1 Übersetzungsverhältnis
» **FingerTech tinyESC** V2
» **Akkupack** Galaxy 3S
» **9V 4,5A Regulator**
» **2 Schaumstoffräder** 2,00 × 0,75 Zoll
» **2 FingerTech Lite Hubs**
» **Fernsteuerung** Typ T6A
» **Empfänger** Typ TR6A
» **FingerTech Mini Power Switch**
» **Annimos Servo** 20 kg; Metallgetriebe
» **5 Plastikschrauben** 3 × 9,5 mm
» **4 Plastikschrauben** 3,5 × 12 mm
» **2 Maschinenschrauben** M3 14 mm
» **Schrumpfschlauch**

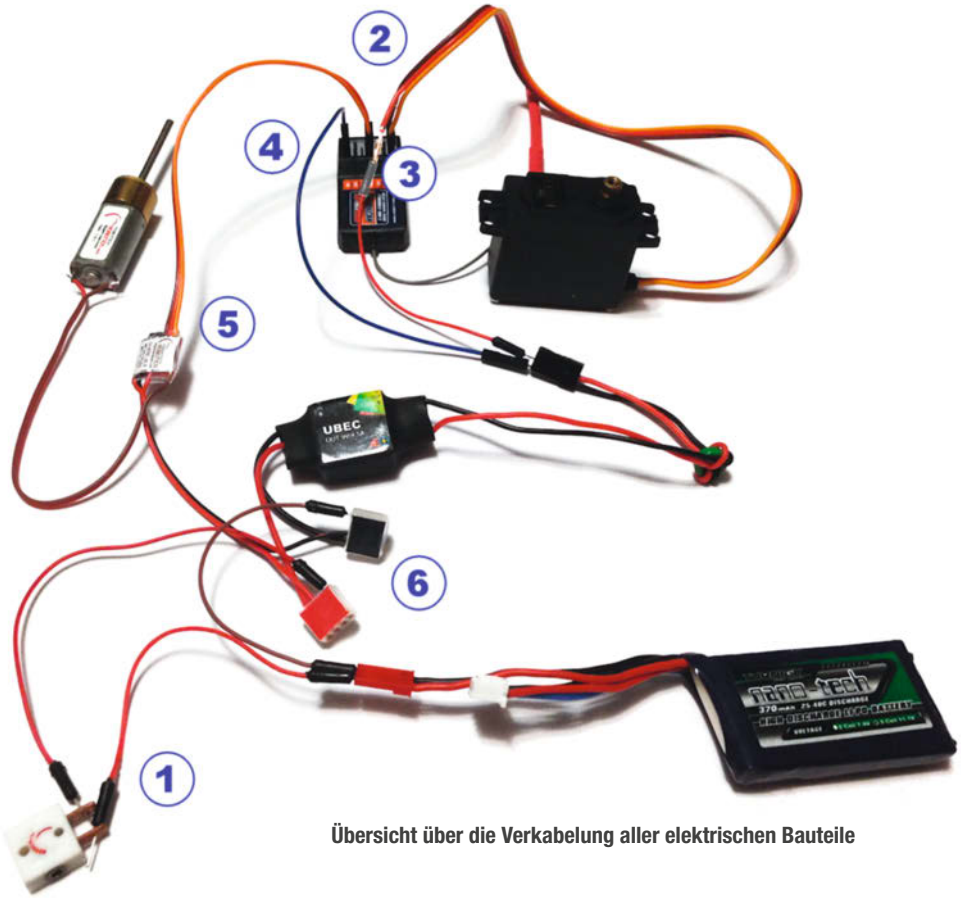
Werkzeug

» **3D-Drucker** Ender 3
» **Schraubenzieherset**
» **PLA+**
» **Heißklebepistole** inklusive Kleber
» **Heißluftpistole**
» **Inbusschlüsselset**
» **Lötkolben** mit Zubehör

Mehr zum Thema

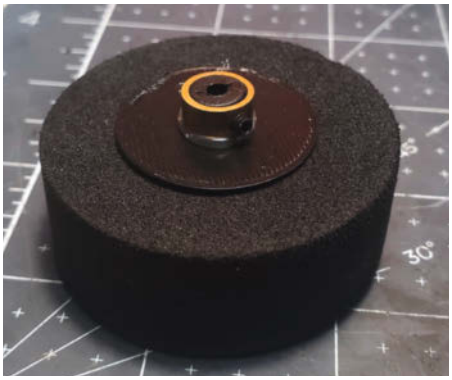
» Marco Düvelmeyer und Ákos Fodor, Roboter fürs Klassenzimmer, Make 7/23, S. 66
» Michael Scheuerl, Smartphone-Roboter für den Unterricht, Make 2/22, S. 62

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



Übersicht über die Verkabelung aller elektrischen Bauteile

In den Schaumstoff werden Plastikteile eingeklebt, um die Motoren befestigen zu können.

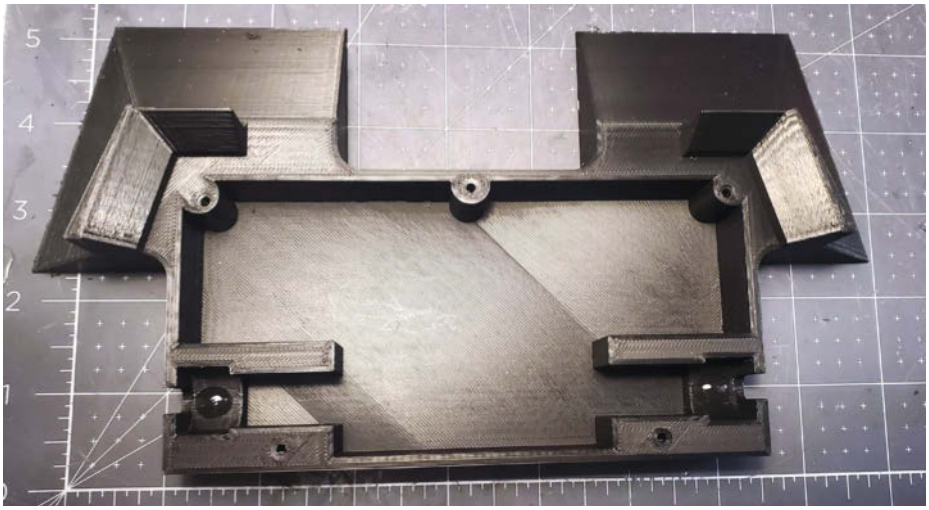


Um die Achsen der Motoren zu befestigen, wird eine Halterung dafür an den Radnaben angebracht.

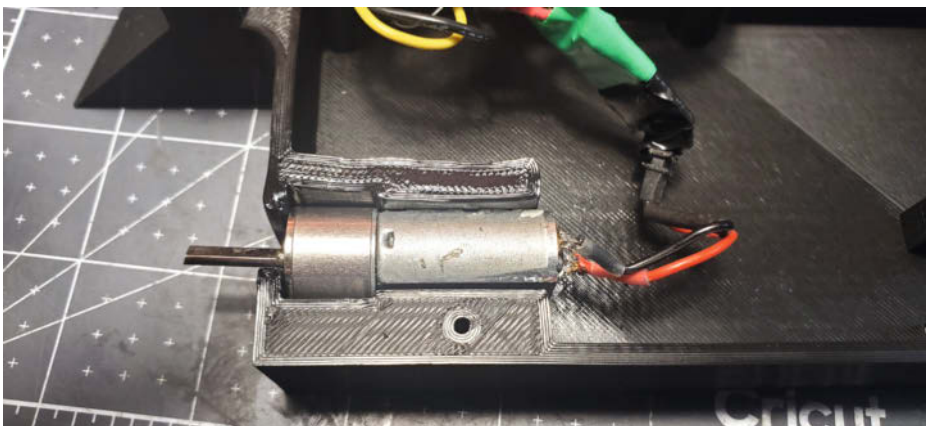
Anschluss des Empfängers und des Servos

Verbinden Sie die Empfängerkabel der tiny-ESCs mit den Kanälen 1 und 2 des Empfängers (Nummer 3 in der Abbildung). Dadurch lässt sich der Antriebsstrang mit dem rechten Bedienhebel der Fernsteuerung steuern.

Die Empfängerkabel vom Servo (Nummer 2 in der Abbildung) werden an den Kanal 3 des Empfängers angeschlossen. Dadurch kann der Hubarm über den zweiten Bedienhebel der Fernbedienung gesteuert werden. Danach schließen Sie das Empfängerkabel des 9-V-Reglers (Nummer 4 in der Abbildung) an den Kanal 4 des Empfängers an. Damit kann der Regler den Servo mit Strom versorgen.



Die Karosserie des Kerfuffle



Die Motoren werden auf beiden Seiten mit Pressfit und Heißkleber befestigt.

Bauen der Räder

Mit einem Klecks Heißkleber auf die größere Hälfte einer 3D-gedruckten Radnabe kleben Sie diese in die Mitte eines Schaumstoffrades.

Auf der anderen Seite des Rades kommt wieder ein Klecks Heißkleber auf die kleinere Hälfte der Nabe und diese stecken Sie wieder in die Mitte der größeren Hälfte. Die Teile müssen gut zusammengedrückt werden, damit auch wirklich alles hält.

Danach kommen die Lite Hubs. Geben Sie etwas Heißkleber auf einen Lite-Hub und setzen Sie ihn in die Mitte der Nabe im Rad ein.

Montage der Antriebsmotoren

Geben Sie einen Klecks Kleber auf die Unterseite jeder Motorhalterung im Fahrgestell.

Nun erhitzen Sie den hinteren Teil des Fahrgestells, um den Silver-Spark-Getriebemotor einzusetzen. Schalten Sie die Heißluftpistole ein und lassen Sie sie auf Temperatur kommen. Konzentrieren Sie sich jeweils auf einen Motorschacht: Sobald der Kunststoff weich wird, setzen Sie den Antriebsmotor in den Schlitz ein. Vergewissern Sie sich, dass der Motor gut montiert ist und im Heißkleber sitzt. Biegen Sie dann den Kunststoff leicht nach innen, damit er den Motor stützt. Lassen Sie dann den Kunststoff abkühlen und aushärten.

Wiederholen Sie diesen Vorgang für den zweiten Getriebemotor.

Anbringen des Hubarms

Schieben Sie den Hubarm über seinen Stützarm auf der oberen Platte, wie in der Abbildung gezeigt.

Bringen Sie den Servohebel an der Ausgangswelle des Servos an. Der Servohebel ist ein Bauteil, das mit der Verzahnung der Aus-

gangswelle verbunden wird (wie ein Zahnrad), sodass Sie die Kraft des Servos auf etwas anderes übertragen können. Schieben Sie den Servo direkt in seinen Einbaubereich. Er sollte genau neben dem Hubarm sitzen.

Befestigen Sie M3-Schrauben durch den Hubarm in dem Servoarm. Aufgrund des Designs des Servoarms müssen Sie zuerst eine Schraube einführen und dann die zweite Schraube anbringen. Möglicherweise müssen Sie die Schrauben kürzen, damit sie genau passen.

Schrauben Sie den Servo mit den vier 3,5 × 12 mm Plastik-Schrauben in die Befestigungslöcher in der oberen Platte. Befestigen Sie die Räder an den Wellen der Getriebemotoren, indem Sie die Lite-Hub-Gewindestifte mit dem Inbusschlüssel festziehen. Befestigen Sie schließlich die obere Platte mit den fünf 3 × 9,5 mm Plastik-Schrauben am Roboterfahrgestell. Viel Spaß mit dem fertigen Kerfuffle!

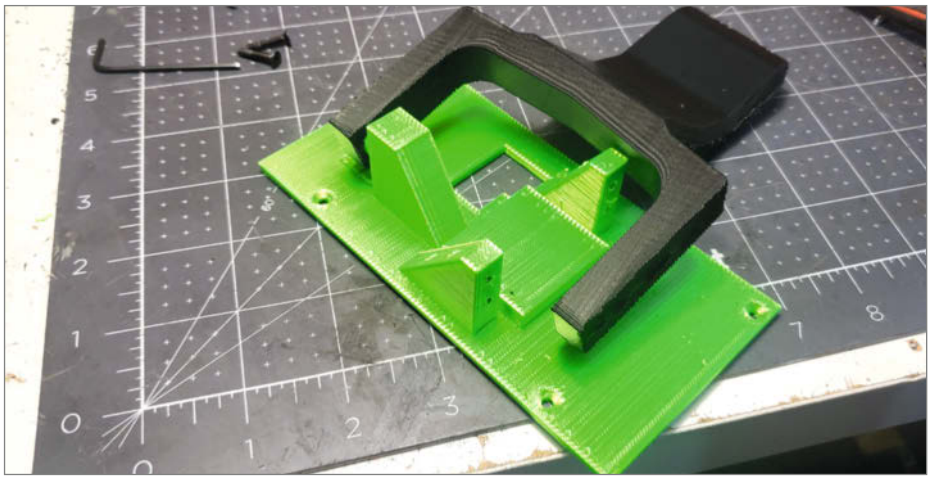
Auf in den Kampf

Was macht man nun mit der fertigen Maschine? Üben Sie zunächst mit Ihrem Roboter. Schalten Sie ihn ein, indem Sie den Mini-Stromschalter mit dem Inbusschlüssel festschrauben. Der Roboter wird mit dem rechten Steuerknüppel (vorne/hinten und links/rechts) bewegt und der linke steuert den Servo des Hebearms (hoch/runter). Machen Sie sich mit der Steuerung Ihres Roboters vertraut und gewöhnen Sie sich daran, wie er sich bewegt.

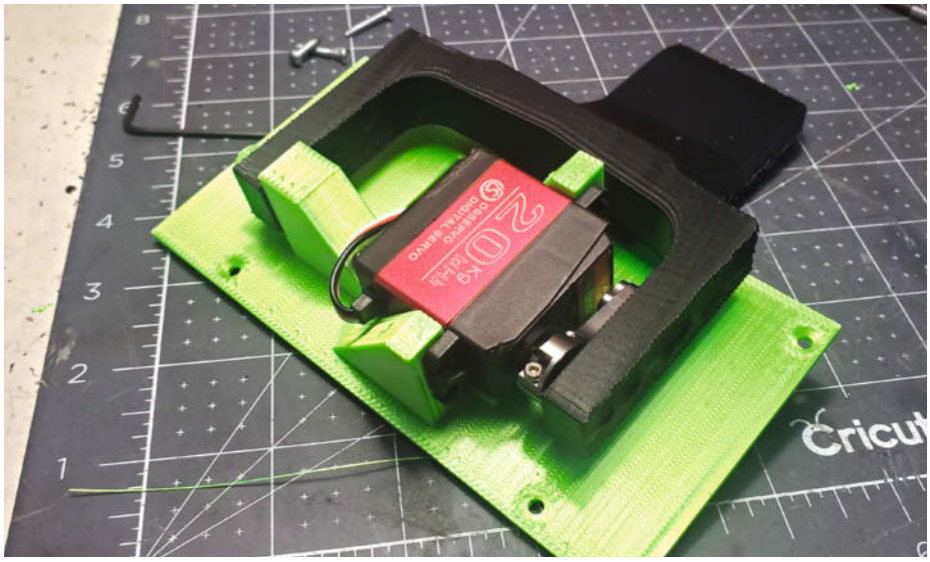
- Fahren Sie Achten – das ist eine gute Möglichkeit, sich an die Steuerung des Roboters zu gewöhnen und seine Geschwindigkeit einzuschätzen.
- Üben Sie das Kämpfen. Da diese Maschine keine gefährlichen drehenden Elemente hat, können Sie ganz einfach Mini-Wettkämpfe veranstalten, bei denen Sie sich in zeitlich begrenzten Scharmützeln gegenseitig zu übertrumpfen und zu bekämpfen versuchen.
- Um anzugreifen, müssen Sie sich einem erhöhten Bereich Ihres Gegners nähern. Da Kerfuffle auf das Anheben angewiesen ist, muss er unter den anderen Roboter gelangen. Suchen Sie also nach Bereichen am Gegner, unter die der Arm gelangen kann.
- Warten Sie mit dem Anheben, bis Sie fest unter dem Gegner stehen. Wenn Sie versuchen, zu früh zu heben, sind Sie angreifbar – also seien Sie geduldig. Wenn Kerfuffle umgekippt ist, können Sie ihn mit dem Hebearm wieder auf die richtige Seite drehen!

Mit Kerfuffle aktuell bleiben

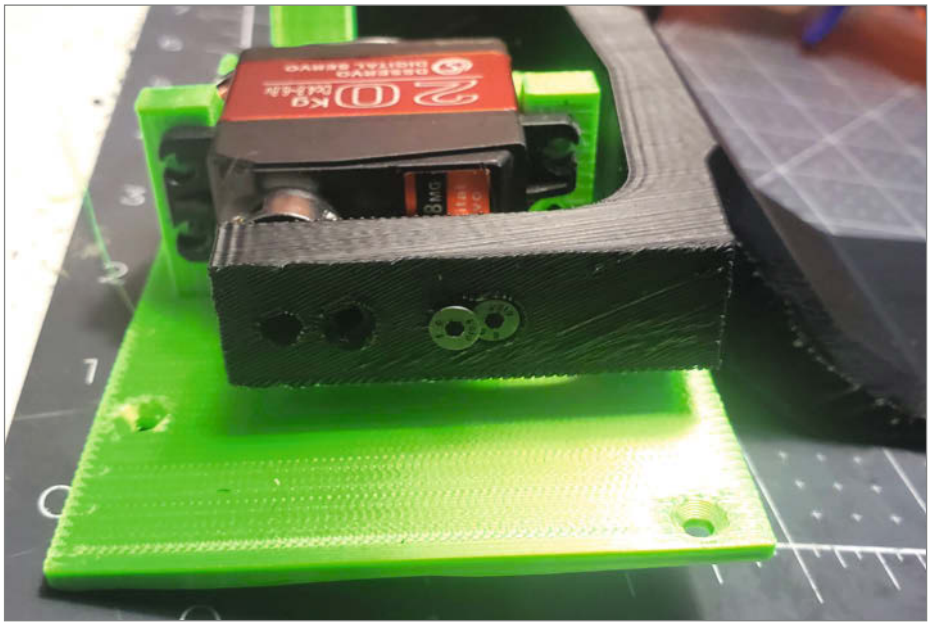
Kerfuffle verändert sich stetig. Neue Entwürfe können Sie auf der Facebook-Seite meines Teams „Bone Dead Robotics“ auf oder auf Instagram unter @bonedeadrobotics finden. *—das*



Der Hubarm wird an der linken Seite eingesteckt.



Die andere Seite des Hubarms wird durch den Servo festgehalten.



Zur Befestigung des Hubarms am Servo werden zwei Schrauben benötigt.

WLAN in Wokwi nutzen

Dank der WLAN-Funktion lassen sich auch IoT-Projekte mit Wokwi simulieren. Mit ihrer Hilfe können die virtuellen Mikrocontroller aus dem Simulator heraus mit externen Servern oder dem eigenen Netzwerk kommunizieren. Wir haben das mit dem ESP32 und dem Pi Pico W ausprobiert.

von Ákos Fodor

Mit dem Online-Simulator Wokwi kann man eine ganze Reihe an Mikrocontroller-Projekten bequem im Browser nachbauen, programmieren und testen (siehe Artikel in der Make 3/23). Dafür stehen Arduino-, STM32-, ESP32-Boards und der Raspberry Pi Pico bereit, an die man virtuelle LEDs, Displays, Servos, Sensoren anschließen kann, um sie etwa über Taster, Joysticks oder Schieberegler zu steuern.

Wer IoT-Projekte entwickelt, will einen Mikrocontroller in der Regel aber auch mit anderen Geräten, einem Smarthome oder dem Internet verbinden. Auch ein einfacher, auf dem Mikrocontroller gehosteter Webserver, auf den man über ein per WLAN verbundenes Smartphone zugreifen kann, ist eine praktische Interface-Lösung.

Das wissen auch die Wokwi-Entwickler und so hat Uri Shaked die WLAN-Funktion des ESP32 in den Simulator implementiert. Dazu musste er allerdings mittels Reverse-Engineering die interne Arbeitsweise des WLANs analysieren, weil Espressif dies nicht offengelegt hat (siehe Video-Link in Kurzinfo). Auch mit dem Raspberry Pi Pico W kann man innerhalb gewisser Grenzen experimentieren. Welche

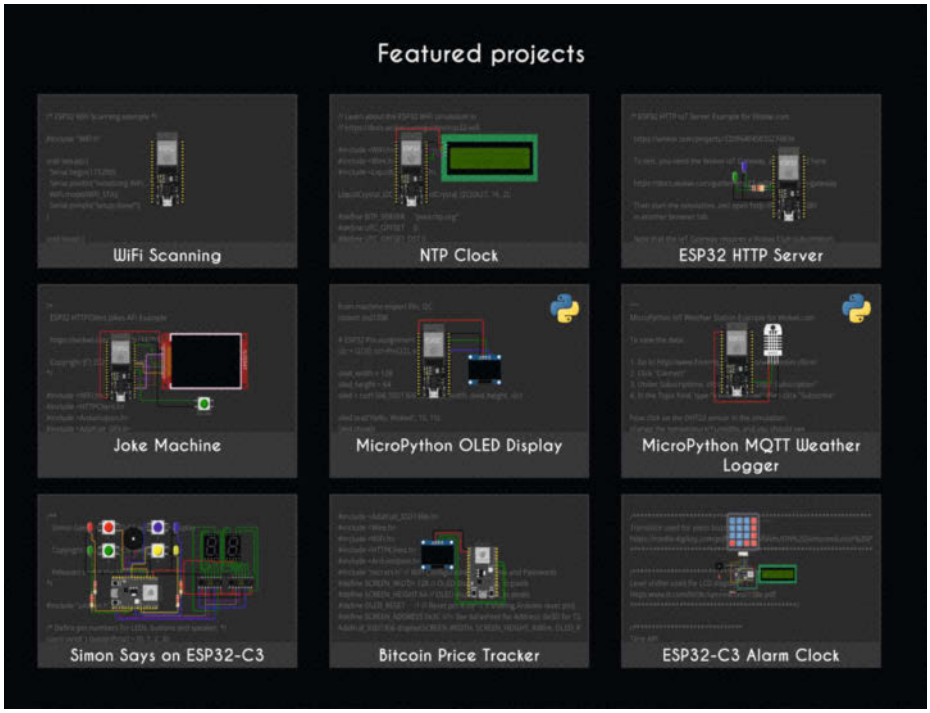


Bild 1: Für den ESP32 hält Wokwi einige Beispiele bereit, die WLAN verwenden.

das sind und wie man die WLAN-Funktion verwendet, haben wir uns angeschaut und auch anhand eines Heft-Projekts getestet.

Kurzinfo

» Projekt mit WLAN in Wokwi simulieren
» Das öffentliche und das private Gateway nutzen
» Task-Reminder in Wokwi nachgebaut

Mehr zum Thema

» Ákos Fodor, Mikrocontroller-Projekte simulieren mit Wokwi, Make 3/23, S. 88

Alles zum Artikel im Web unter
make-magazin.de/x56u

Virtuell verbunden

In den eigenen vier Wänden regelt für gewöhnlich der Switch im WLAN-Router das Netzwerk, den Internetverkehr (per Modem) und die Geräte, die sich mit ihm verbinden dürfen. Zu diesen gehört auch der Computer, in dessen Browserfenster man Wokwi ausführt. Das erlaubt dem virtuellen Mikrocontroller jedoch nicht, sich mit dem lokalen WLAN zu verbinden. Stattdessen erzeugt Wokwi einen virtuellen Access Point auf dem PC, auf dem der Simulator läuft. Da aber selbst dieser an den Browser und dessen Kommunikationsprotokolle gebunden ist, benötigt die Simulation ein zusätzliches Programm, ein Gateway, um mit anderen Diensten und Netzwerken kommunizieren zu können. Dieses verbindet sich via Websocket mit Wokwi und implementiert einen DHCP-Server. Als Nutzer kann man entscheiden, ob das Gateway diese Aufgabe auf dem Wokwi-Server oder dem eigenen Computer erledigen soll.

Öffentliches und privates Gateway

Wer einen kostenfreien oder gar keinen Wokwi-Account besitzt und Projekte mit WLAN-Funktion ausprobiert, nutzt automatisch das öffentliche Wokwi-Gateway (Public Gateway). Da es auf dem Wokwi-Server liegt, gibt es allerdings ein paar Einschränkungen. Die Entwickler behalten sich etwa das Recht vor, den durchgeschleusten Traffic aus Sicherheitsgründen zu überwachen. Daher weist die Online-Dokumentation von Wokwi darauf hin, dass man keine sensiblen (z.B. API-Schlüssel) oder privaten Daten an das öffentliche Gateway verschicken sollte. Außerdem hat man durch die Umleitung über den Server eine etwas langsamere Verbindung und das Gateway erlaubt keine eingehenden Internetverbindungen, weil man dafür direkten Zugriff auf das Wokwi-Netzwerk bräuchte, auf dem das Programm arbeitet.

Für solche Aufgaben bietet sich das private Gateway an, das man über das GitHub-Repository von Wokwi herunterladen kann. Mit ihm lässt sich etwa ein Webserver testen oder mit einem lokalen MQTT-Broker kommunizieren. Ins Internet gelangt der virtuelle Mikrocontroller dann direkt über den eigenen Computer und nicht mehr über den Wokwi-Server. Dadurch überwacht Wokwi auch nicht mehr, welche Daten vom Gateway verarbeitet werden.

Das private Gateway stellt Wokwi als Binary und Source-Code bereit, falls man das Programm anpassen und selbst kompilieren möchte. Damit man es nutzen kann, muss man allerdings ein Wokwi-Club-Mitglied sein, was im Monat etwa 7 Euro kostet und weitere Features beinhaltet. So kann man etwa seine Projekte auf dem Wokwi-Server privat halten und die zukünftige Entwicklung des Simulators beeinflussen.

Leichter Einstieg

Eigentlich ist es ganz einfach: Um sich mit dem virtuellen Access Point zu verbinden, benötigt man lediglich dessen SSID (Wokwi-GUEST).

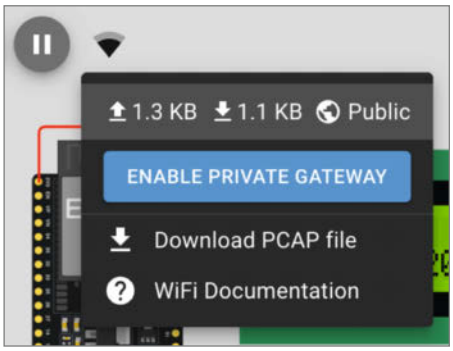


Bild 2: Sobald die Simulation läuft, kann man die Verwendung des privaten Gateways aktivieren.

Ein Passwort gibt es nicht. Ansonsten verhält sich die simulierte WLAN-Verbindung wie eine echte. Wenn man also einen funktionierenden Code hat und in Wokwi kopiert, muss man nicht viel anpassen – und andersherum genauso.

Für all diejenigen, die noch nicht so recht wissen, was man mit der WLAN-Funktion eines ESP32 machen kann, hält Wokwi außerdem ein paar einfache Beispiele parat, die fast alle über das öffentliche Gateway – also kostenfrei – funktionieren (Bild 1). Diese zeigen, wie man nach WLAN-Netzwerken Ausschau hält, die aktuelle Uhrzeit über NTP (Network Time Pro-

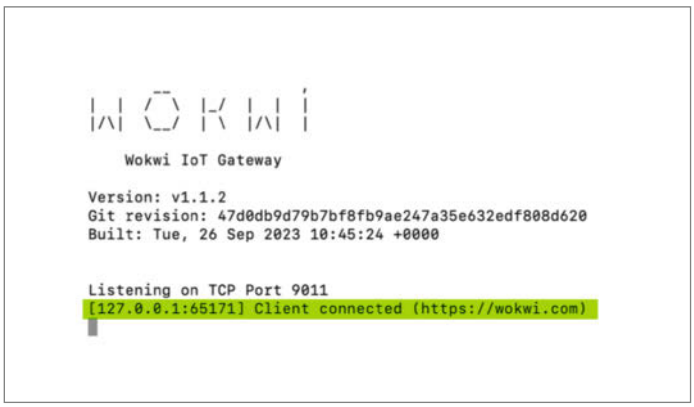


Bild 3: Das private Gateway hat sich erfolgreich mit der Wokwi-Simulation verbunden.

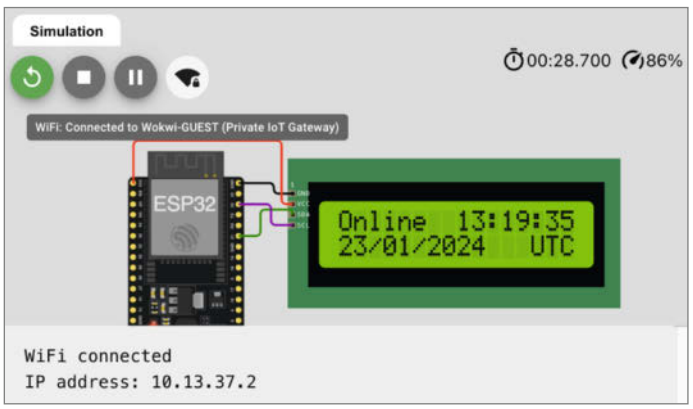


Bild 4: Die IP-Adresse und das Symbol mit dem Schloss verraten: Wokwi nutzt jetzt das private Gateway.

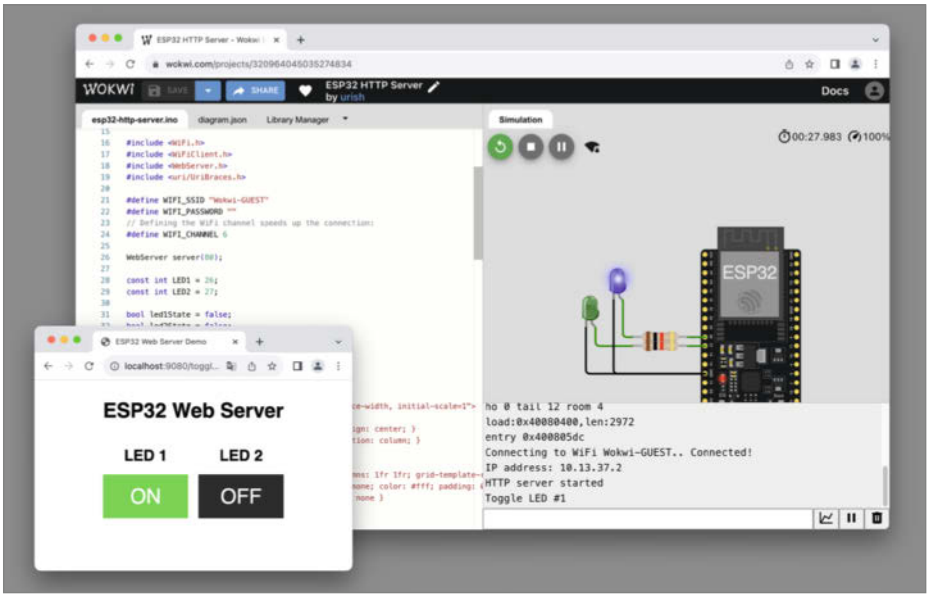


Bild 5: Mit dem privaten Gateway kann man auch einen ESP32-Webserver testen.

tocol) bezieht oder Witze über die JokeAPI abfragt, um sie anschließend auf einem Display darzustellen. Wer für Testzwecke oder generell gern Online-MQTT-Broker nutzt (z.B. HiveMQ), kann selbst das mit einem gratis Wokwi-Account testen.

Lokal und privat

Anders sieht es bei dem Webserver-Beispiel (ESP32 HTTP Server) aus, bei dem man zwei LEDs über ein Webinterface ein- und ausschalten kann. Dieser Sketch funktioniert nur, wenn man das private Gateway verwendet, denn der Webserver wird über localhost aufgerufen.

Dazu startet man das heruntergeladene private Gateway. Sobald die Botschaft „Listening on Port 9011" erscheint, wechselt man wieder zum Wokwi-Projekt zurück. Der Simulator weiß zu diesem Zeitpunkt noch nicht, dass man sich über ein privates Gateway verbinden will. Sobald man den Sketch ausführt, gibt der serielle Monitor daher die IP-Adresse 10.10.0.2 aus, die das öffentliche Gateway vergibt. Um zum privaten Gateway zu wechseln, klickt man im oberen Bereich des Bildschirms auf das WLAN-Symbol neben der Pause-Taste und dann auf „enable private gateway" (Bild 2). Anschließend muss man die Simulation stoppen und neu starten.

Schaut man jetzt auf das Fenster des Gateways, ist dort eine neue Zeile aufgetaucht, die mit „Client connected" bestätigt, dass sich Wokwi erfolgreich mit ihm verbunden hat (Bild 3). Im seriellen Monitor gibt der Simulator jetzt auch nicht mehr die IP-Adresse 10.10.0.2, sondern 10.13.37.2 aus und das WLAN-Symbol ist zusätzlich mit einem Schloss versehen – zumindest solange die Simulation läuft (Bild 4). Stoppt man sie, verschwindet manchmal das WLAN-Symbol oder das Schloss und Wokwi gibt an, dass es keine Verbindung zum privaten Gateway herstellen konnte. Diese Meldung kann man ignorieren, denn Wokwi hat sich die Einstellungen gemerkt und kann sich darüber hinaus gar nicht mehr verbinden, wenn das private Gateway fehlt.

Mit dem aktivierten privaten Gateway kann man nun auf den simulierten Webserver zugreifen. Dazu öffnet man ein neues Browserfenster – keinen Tab, weil sonst die Simulation pausiert wird – und gibt in die Adresszeile `localhost:9080` ein. Daraufhin erscheint ein einfaches Interface mit zwei grauen Buttons, über die man die beiden LEDs, die mit dem ESP32 verbunden sind, steuern kann (Bild 5).

Wer den Datenaustausch mit einem lokalen MQTT-Broker ausprobieren will, sollte sich den freien MQTT-Server Mosquitto auf einem Raspberry Pi installieren und konfigurieren (z.B. per SSH). Wenn man anschließend das „MicroPython MQTT Weather Logger"-Beispiel in Wokwi öffnet und die IP-Adresse des Raspi bei `"MQTT_BROKER"` sowie die Nutzerdaten einträgt, sendet der Simulator die Werte des DHT22 über das private Gateway direkt an den Raspberry Pi (Bild 6).

Task-Reminder in Wokwi

Um herauszufinden, wie gut sich ein Projekt aus dem Heft mit Wokwi simulieren lässt, haben wir den Task-Reminder von Bernd Heisterkamp (S. 8 ff.) nachgebaut. Dieser unterstützt dabei mithilfe von NeoPixel-LEDs, Aufgaben nicht zu vergessen und gleicht sich dafür mit einem Google-Server ab. Zur Kommunikation nutzt das Projekt URLs, die es prinzipiell jedem erlauben können, Änderungen in der Projektdatenbank vorzunehmen. Für diesen Test ist das private Gateway also bestens geeignet.

Als Erstes muss man das Backend und die URLs erstellen, wie es im Task-Reminder-Artikel beschrieben ist. Danach verkabelt man einen ESP32 in Wokwi mit den LEDs und den Tastern. Da der Task-Reminder MicroPython verwendet, bietet sich als Basis die Wokwi-Vorlage „Neopixel Ring Rainbow" an, die man in der Rubrik MicroPython unter „Basic Examples" findet. Den LED-Ring kann man löschen und sich danach einen LED-Streifen aus einzelnen NeoPixeln aufbauen. Zu diesen gibt es leider keine Informationen in der Wokwi-Dokumentation, aber im Grunde muss man nur darauf achten, dass man den Datenausgang (DOUT) jeweils mit dem Dateneingang (DIN) der nächsten LED verbindet. Anschlie-

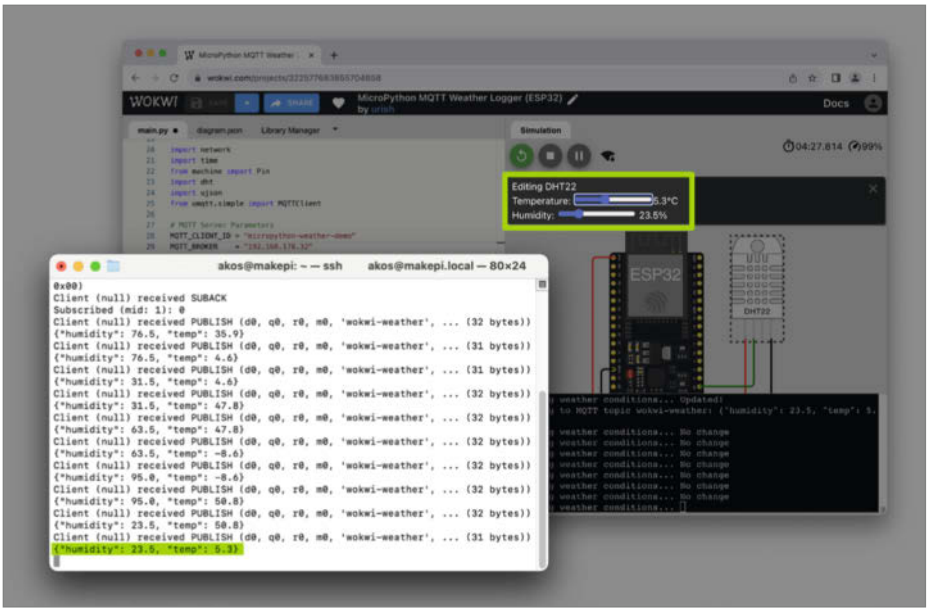


Bild 6: Durch die lokale Anbindung kann Wokwi auch mit einem lokalen MQTT-Broker kommunizieren.

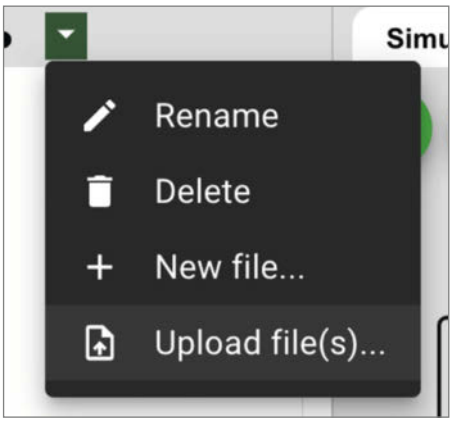


Bild 7: Statt Copy und Paste kann man auch Code hochladen.

ßend verkabelt man die Taster. Diese kann man über eine Schaltfläche softwareseitig entprellen. Da Wokwi keine analogen Schaltungen simulieren kann, entfallen hier nämlich die Kondensatoren und Widerstände des ursprünglichen Entwurfs.

Danach lädt man die drei Python-Programme des Task-Reminders hoch, indem man auf den kleinen Abwärtspfeil über dem Editor klickt und„Upload file(s)“ auswählt (Bild 7). Schließlich muss man nur noch die SSID in secrets.py sowie die URLs, die zu Google führen, in main.py einfügen (siehe Anleitung des Task-Reminder).

Jetzt sollte alles funktionieren. Je nach Vorlage kann aber passieren, dass das REPL stattdessen eine Fehlermeldung ausgibt. In unserem Fall etwa `NotImplementedError: Redirects not yet supported`. Sie deutet darauf hin, dass das Projekt eine veraltete MicroPython-Version oder veraltete Module nutzt. Tatsächlich ist es in der Vorlage mit dem NeoPixel-Ring die Version 1.18 von Anfang 2022. Das lässt sich zum Glück schnell beheben, denn man kann die Firmware mit einem einzigen Eintrag auf den neusten Stand bringen. Dazu wechselt man oberhalb des Editors zur Datei diagram.json und sucht nach `"micropython-20220117-v1.18"`. Diesen Eintrag ersetzt man dann etwa durch die Version `"micropython-20231227-v1.22.0"` (Bild 8).

Nach dieser Anpassung lief das Programm erfreulicherweise genauso, wie erhofft (Bild 9). Je nachdem, auf welcher Vorlage man sein Projekt aufbaut, sollte man hier also etwas genauer hinschauen. Ansonsten stehen die Wokwi-Entwickler über Discord mit Rat und Tat zur Seite, falls mal etwas hakt. *—akf*

```
  "editor": "wokwi",
  "parts": [
    {
      "type": "wokwi-esp32-devkit-v1",
      "id": "esp",
      "top": 119.9,
      "left": 4.6,
      "attrs": { "env": "micropython-20231227-v1.22.0" }
    },
```

Bild 8: Bei MicroPython-Projekten sollte man unbedingt auf die verwendete Version achten.

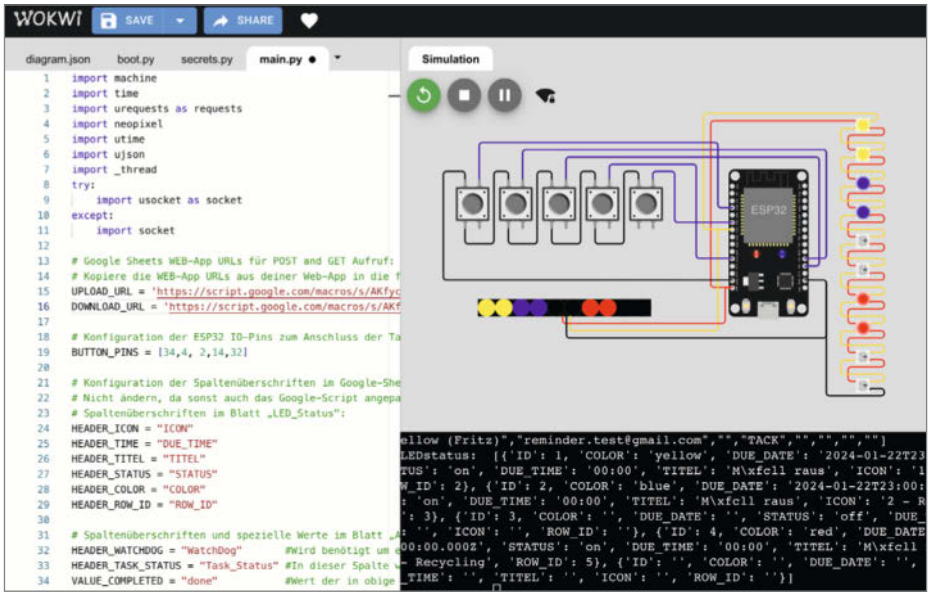


Bild 9: Auch im simulierten Aufbau kann der Task-Reminder Daten mit dem Google-Server abgleichen.

Pico W (beta)

Wokwi hat auch einen Raspberry Pico W im Repertoire, der ebenfalls mit einem virtuellen WLAN-Chip ausgestattet ist. Allerdings befindet sich das Board noch in der Beta-Phase und man kann nicht so leicht in WLAN-Projekte einsteigen wie mit dem ESP32. Das liegt einerseits daran, dass man vergeblich nach Beispielen auf der Wokwi-Website sucht. Zum Zeitpunkt des Artikels bietet Wokwi lediglich eine Vorlage für Pico SDK an. Andererseits funktioniert beim leeren MicroPython-Template für den Pico W, das man unter „MicroPython/Starter Templates/New Project“ findet, das WLAN-Modul nicht. Man muss also entweder in den Nutzerprojekten stöbern oder Google befragen.

Dort findet man recht schnell ein funktionierendes MicroPython-Beispiel (siehe Link in Kurzinfo). Vergleicht man dessen diagram.json-Datei mit der des leeren Templates, sieht man, dass bei Letzterem der Eintrag `"cyw43" = "1"` fehlt, der offensichtlich den WLAN-Chip (CYW43439) des Pico W aktiviert (Bild 10). Gleichzeitig scheint es nicht möglich zu sein, die WLAN-Funktion und eine selbstgewählte MicroPython-Version zu kombinieren. In dem verlinkten Beispiel ist man dadurch derzeit an die Firmware 1.19.1 von März 2023 gebunden. Immerhin funktioniert die WLAN-Verbindung auch mit dem privaten Gateway, das von Wokwi in der Dokumentation als Feature für den ESP32 aufgeführt ist.

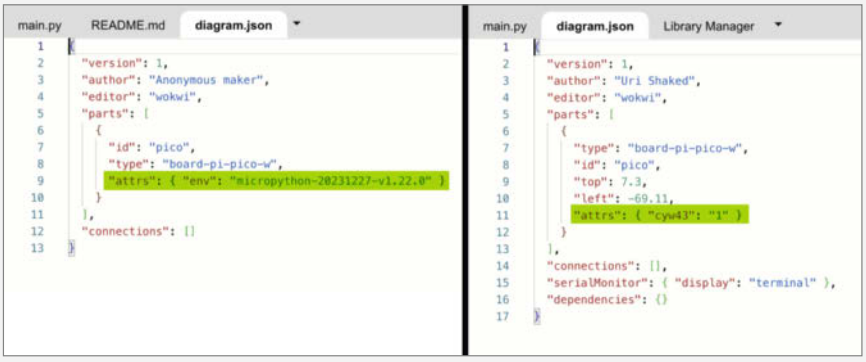


Bild 10: Die json-Konfiguration eines gegoogelten Beispiels (rechts) kann sich per WLAN verbinden – die Wokwi-Vorlage links kann das nicht.

Lasern mit Lightburn

Gerade preiswerte Lasercutter aus China werden meist mit grottenschlechter Software geliefert. Es gibt Besseres: Lightburn ist beim Lasern State of the Art. Hier erfahren Sie, wie man es installiert und damit arbeitet.

von Heinz Behling

Bild 13: So sieht die fertige Platte aus.

Lasercutter und 3D-Drucker haben eines gemeinsam: Sie werden von einer Software gesteuert, die aus den Konstruktionsdateien die zur Ansteuerung der Maschine notwendigen Befehle generiert. Was bei 3D-Printern zum Beispiel Cura übernimmt, erledigen bei Billig-Lasern aus China Programme wie RDWork oder modifizierte alte CorelDraw-Versionen bzw. bei teureren Geräten eine herstellereigene Software.

Besonders die Ausgaben aus China haben zu Recht einen schlechten Ruf: So stürzt beispielsweise das alte CorelDraw 12 auf aktuellen Windows-Versionen häufig ab. Und RDWorks ist äußerst holprig aus dem Chinesischen übersetzt worden, vergisst gerne mal die Lasercutter-Konfiguration und hat beim Dateiimport von Inkscape oft Probleme. Beiden gemein ist eine recht komplizierte Bedienung.

Es gibt Besseres: LightBurn. Dieses Programm ist aktuell der „state of the art". Es ist deutlich einfacher zu bedienen, was nicht nur an der besseren Übersetzung liegt, sondern auch an einer deutlich intuitiveren Menüstruktur und vor allem auch an erheblich einfacheren Einstellmöglichkeiten der Laserparameter. Außerdem arbeitet es mit einer Vielzahl von Lasercuttern zusammen (siehe Kasten „Auswahl der richtigen Lizenz") und steht außer für Windows auch für MacOS und Linux zur Verfügung. Lightburn gibt es jedoch nicht gratis, die Preise liegen je nach Lizenz zwischen 56 und 195 Euro für die Installation auf bis zu drei Computern inklusive Updates für ein Jahr. Allerdings gibt es eine voll funktionsfähige Testversion, die 30 Tage lang gratis benutzt werden kann.

Dieser Artikel zeigt Ihnen die grundlegende Bedienung und gibt Tipps zur Installation und Einstellung. Das erfolgt am Beispiel der Windows-Version. Auf Apple- und Linux-Computern sieht die Oberfläche aber genauso aus. Sie erfahren, wie man in Lightburn selbst konstruiert, was beim Import von Dateien aus anderen Konstruktionsprogrammen zu beachten ist, und wie man in nur einer Datei gleichzeitig Daten für Schnitte und Gravuren unterbringt.

Installieren Sie zunächst Lightburn (siehe Kasten auf Seite 30). Eine Lizenz ist dafür für die ersten 30 Tage zunächst einmal nicht notwendig.

Wenn Lightburn gestartet ist, sollte es auf die Sprache des auf dem Computer installierten Betriebssystems eingestellt sein. Das funktioniert jedoch bei Mehrsprachen-Versionen von Windows nicht immer zuverlässig, Lightburn startet dann in der englischen Version. Falls Sie eine andere Sprache benutzen möchten, wählen Sie unter „Language" die gewünschte Sprache aus.

Bedienoberfläche

Die Oberfläche von Lightburn ist klar aufgeteilt (Bild 1).

Kurzinfo

» **Konstruieren, Lasern und Gravieren mit Lightburn**
» **Einstellen der Schneide- und Gravierparameter**
» **Tipps zur Installation und geeignete Lasercutter**

Mehr zum Thema

» Heinz Behling: Modernisierung für den China-Laser, Make 1/21, S. 112
» Heinz Behling: Lasercutter K40 justieren, Make 4/21, S. 112
» Heinz Behling: Abgas-Reinigungsanlage für Lasercutter, Make 4/18, S. 120

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

In der Mitte steht der Arbeitsbereich, dessen Achsen jeweils mit Zahlenskalen versehen sind. Deren Maximalwerte entsprechen der Größe des Schneidetisches des aktiven Lasercutters. Falls Sie mehrere Lasercutter in Lightburn installiert haben, können Sie in der Lasersteuerung (rechts neben dem Arbeitsbereich) neben „Geräte" das jeweilige Gerät auswählen.

Im Arbeitsbereich können Sie Teile mithilfe der Konstruktionswerkzeuge (Leiste links neben dem Arbeitsbereich) selbst zeichnen, modifizieren, beschriften, verschieben usw. Hier erscheinen auch die Inhalte aus importierten Dateien.

Die aktuelle Position und Größe eines zuvor mit der Maus ausgewählten Teiles wird links oberhalb des Arbeitsbereichs angezeigt. Die neun Kreise daneben legen den Bezugspunkt fest, auf den sich die X- und Y-Koordinaten beziehen. Im Bild 1 ist es die Mitte der rechten Seite des gewählten Objekts. Rechts daneben finden Sie Einstellungen für die Textfunktion für Beschriftungen innerhalb Ihrer Konstruktion: Die wichtigsten sind Schriftart (jede auf dem Computer installierte Schrift) und -größe sowie der horizontale und vertikale Abstand der Zeichen und Zeilen.

Über dem Schriftbereich steht die übliche Werkzeugleiste mit Symbolen beispielsweise fürs Öffnen von Dateien, Undo, Copy, Paste, usw. Wenn Sie den Mauszeiger auf einem Symbol stehen lassen, erscheint nach kurzer Zeit ein Erläuterungstext.

Unter dem Arbeitsbereich steht die Farbauswahl. Durch Klicken auf eines der Farbfelder können Sie dem zuvor ausgewählten Teil Ihrer Konstruktion die entsprechende Farbe zuordnen. Das dient nicht ästhetischen Gesichtspunkten, sondern dient dazu, den Teilen unterschiedliche Laser-Parameter zuordnen zu können. Dazu gibt es später mehr Infos.

Der Bereich rechts ist für zusätzliche Einstellungs- und Bedienfenster vorgesehen und kann sein Aussehen daher ändern. Direkt nach

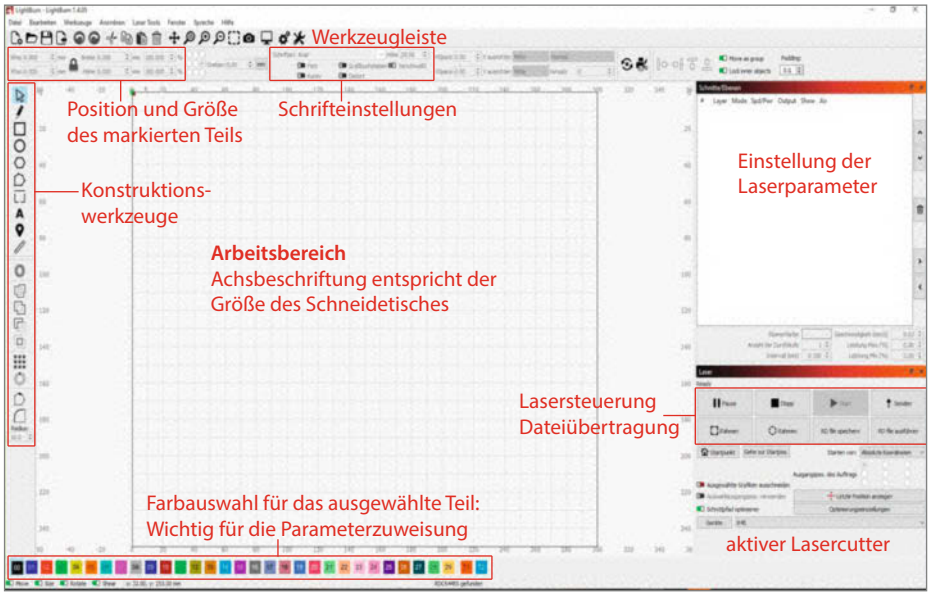


Bild 1: Die Aufgabenbereiche der Lightburn-Oberfläche

der Installation finden Sie dort wie im Bild das Schnitt-/Ebenenfenster oben (dient dem Einstellen der Laser-Parameter) und das Laser-Fenster unten. Im Laserfenster starten oder stoppen Sie das Gerät. Sie können aber auch mit „Senden" die Schnittdatei lediglich auf den Laser übertragen, ohne ihn zu starten. Neben „Starten von" stellen Sie ein, von wo aus der Schnitt beginnen soll. „Absolute Koordinaten" beispielsweise platziert die Teile auf dem Schneidetisch an den Stellen, an denen sie auch auf der Lightroom-Oberfläche liegen.

Welche Fenster im rechten Bereich erscheinen, legen Sie unter „Fenster" fest. Mit „Fenster/Kamerasteuerung" schalten Sie hier das Fenster mit dem Kamerabild ein (Bild 2, siehe Artikel „Kamera für Lightburn" auf Seite 34).

Die Fenster können Sie verschieben, indem Sie in deren roten Titelbalken klicken und sie mit gedrückter Maustaste an den gewünschten Platz schieben. Der kann übrigens auch außerhalb des Lightroom-Fenster liegen, sogar auf einem anderen Bildschirm. Wenn Sie alle Fenster auslagern, erscheint der Arbeitsbereich formatfüllend.

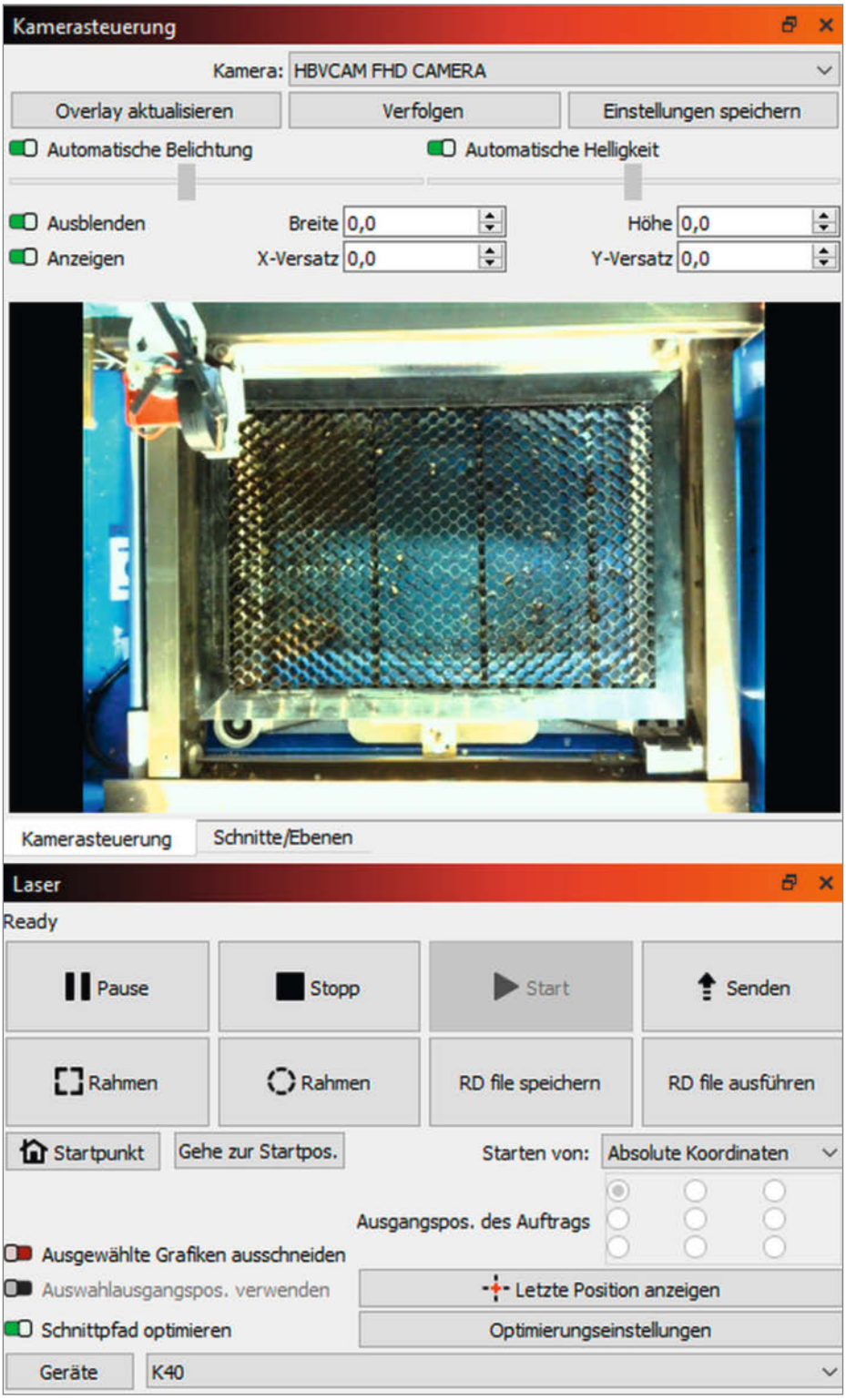


Bild 2: Ist im rechten Bereich nicht genug Platz, um alle zusätzlichen Fenster anzuzeigen, erscheinen Auswahlreiter, hier zum Beispiel für „Kamerasteuerung" und „Schnitte/Ebenen".

Konstruieren mit Lightburn

Um mithilfe von Lightburn zu lasern, gibt es zwei grundlegende Möglichkeiten: selbst konstruieren in Lightroom oder Dateien aus anderen Programmen importieren.

Zunächst zur ersten Möglichkeit. Am Beispiel einer Frontplatte mit einer Bohrung für eine Potenziometer-Achse, einer dazugehörenden Skala sowie Beschriftung zeige ich Ihnen, wie man Lightroom bedient.

Die Frontplatte soll 8 × 10 cm groß sein. Das 8mm-Loch der Achse soll 4 cm vom rechten Rand auf mittlerer Höhe (4 cm) liegen. Außerdem soll ein Logo auf der Platte prangen und unter dem Poti soll der Schriftzug „Lautstärke" stehen.

Beginnen wir mit dem Umriss der Platte: Mit dem Viereck-Werkzeug zeichnen wir zunächst ein Rechteck auf der Arbeitsfläche. Dessen genaue Maße geben wir anschließend als Zahlenwerte in die entsprechenden Felder oberhalb des Arbeitsbereichs ein (Bild 3). Achten Sie dabei darauf, dass das Schlosssymbol neben den Breite- und Höhenfeldern geöffnet ist. Andernfalls bleiben Höhe und Breite im selben Verhältnis zueinander.

Jetzt schieben wir dieses Rechteck auf die Koordinaten 0,0. Aber bitte nicht mit der Maus, das ist viel zu schwierig. Setzen Sie stattdessen in die Felder XPos und YPos jeweils 0 ein. Sie werden feststellen, dass Ihnen das Rechteck nun nach links außerhalb des Arbeitsbereiches entfleucht (Bild 4).

Das liegt daran, dass der falsche Bezugspunkt eingestellt ist. Klicken Sie im Feld mit den neun Kreisen (siehe Beschreibung Bild 1) auf den oben links und geben Sie erneut die X- und Y-Koordinaten ein. Dann erscheint das Rechteck an der gewünschten Stelle (Bild 5).

Jetzt kommt das Loch für die Achse, dessen Mittelpunkt jeweils 4 cm vom rechten und unteren Rand des Rechtecks entfernt liegen muss. Mit dem Kreis-Werkzeug zeichnen Sie das Loch zunächst an beliebiger Stelle. Tipp: Wenn Sie beim Aufziehen des Kreises die Umschalt-Taste gedrückt halten, wird es auch wirklich ein Kreis und kein Oval. Die genaue Größe stellen Sie in den Feldern Breite und Höhe ein (je 8 mm).

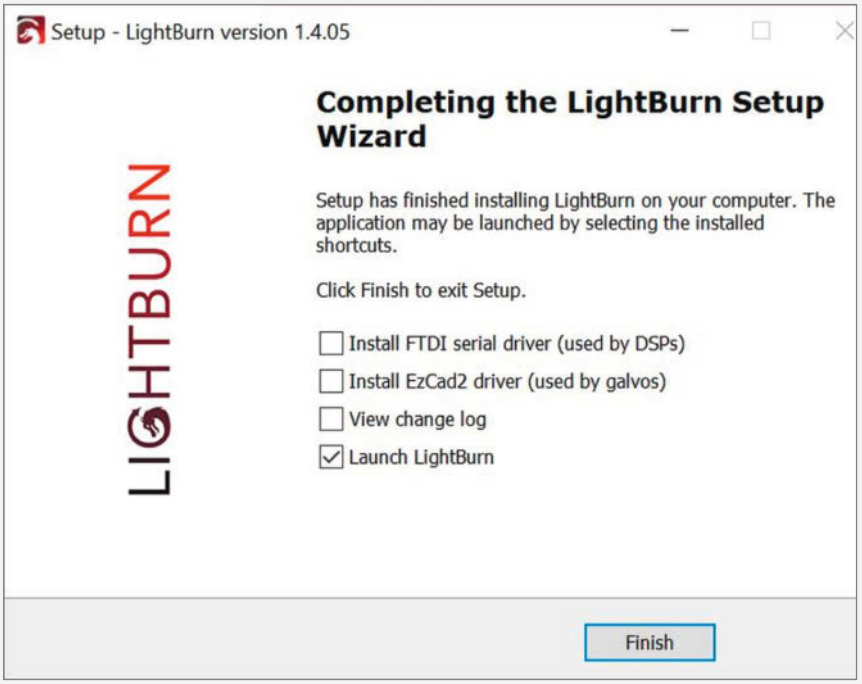
Jetzt zur Position des Mittelpunkts. Da müssen Sie zunächst auch den entsprechenden

Installation von Lightburn

Gleichgültig welchen Lasercutter und dementsprechend welche Lizenz Sie benutzen möchten, das Programm ist stets dasselbe. Nach dem Download (Adresse siehe Kurzinfo-Link) der Programmversion für Ihr Betriebssystem schalten Sie den Lasercutter ein und starten Sie die erhaltene Datei. Die Installation ist weitestgehend selbsterklärend. Bejahen Sie die Fragen, ob Sie die erforderlichen Visual-C-Versionen installieren möchten. Benutzen Sie einen Laser mit DSP- oder Galvo-Controller, müssen auch die FTDI- und EzCad-Treiber installiert werden.

Nach der Installation startet Lightburn. Falls auf Ihrem Computer eine Firewall aktiv ist, kann die sich melden und Ihre Zustimmung verlangen. Anschließend müssen Sie entweder den 30tägigen Testzeitraum starten oder einen gültigen Lizenzschlüssel eingeben.

Lightburn kann nur arbeiten, wenn es mit einem Lasercutter verbunden ist. Das liegt unter anderem daran, dass es wichtige Parameter wie Arbeitstischgröße und die Lage des Nullpunktes vom Lasercutter braucht. Ist der Laser über USB angeschlossen, wird er automatisch gefunden und Sie müssen das lediglich bestätigen. Alle erforderlichen Angaben werden daraufhin aus dem Lasercutter ausgelesen und in Lightburn gespeichert.

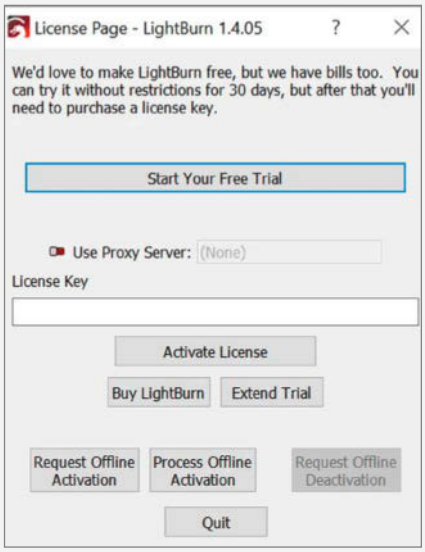


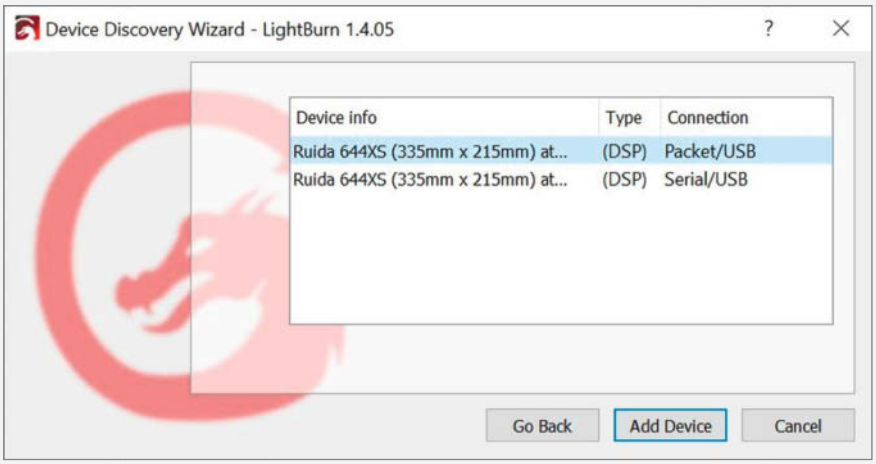
Für DSP- und EzCad-Geräte müssen Sie an dieser Stelle die entsprechenden Treiber aktivieren.

Soll die Verbindung übers Netzwerk erfolgen, wird der Lasercutter nicht automatisch erkannt. In diesem Fall ist nach einem Klick auf „Create Manually“ die Auswahl des Controller-Typs und die Eingabe der IP-Adresse des Lasercutters notwendig. Die können Sie entweder am Router des Netzwerks oder am Laser selbst auslesen. Wie das geht, erfahren Sie in den Handbüchern der Geräte. Mit Lightburn können Sie übrigens beliebig viele Lasercutter benutzen, sofern deren Typ der Lightburn-Lizenz entspricht.

Zum Ende der Installation erscheint das Lightburn-Fenster mit einer Arbeitsfläche entsprechend der Schneidetischgröße Ihres Cutters.

Um Lightburn zunächst kostenlos auszuprobieren, klicken Sie hier auf „Start Your Free Trial“.





Erscheinen mehrere Einträge für einen DSP-Laser wählen Sie den oberen.

Bezugspunkt einstellen (mittlerer Kreis) und dann als YPos (senkrechte Achse) 40 mm und als XPos 60 mm (Breite des Rechtsecks abzüglich Abstand des Mittelpunkts vom rechten Rand, Bild 6).

Jetzt kommen wir zum Logo (was könnte da anderes in Frage kommen als das Make-Logo). Das haben wir als Bilddatei, die nun eingefügt werden soll. Mit „Datei/Importieren“ öffnen wir ein Explorer-Fenster und wählen die Bilddatei aus. Das Logo erscheint grau auf der Arbeitsfläche. Mit dem Verschiebe-Werkzeug (Pfeil) schieben Sie es an die gewünschte Position auf der Platte und passen außerdem die Größe an. Ändern Sie außerdem die Farbe auf rot. Warum? Das Logo soll ja nicht, wie die bislang eingesetzten Teile, geschnitten, sondern graviert werden. Es braucht also andere Laser-Parameter. Unterschiedliche Parameter ordnet man in Lightburn durch verschiedene Farben zu. Daher das rot! Wundern Sie sich übrigens nicht, wenn das Logo selbst weiterhin grau erscheint. Lediglich der Markierungsrahmen darum wird rot gestrichelt (Bild 7).

Blicken Sie jetzt einmal auf das Fenster „Schnitte/Ebenen“ am rechten Rand: Dort sind nun zwei Einträge vorhanden, einer für einen schwarzen, der zweite für einen roten Layer (Bild 8).

Nun folgt als letztes Element noch der Schriftzug „Lautstärke“. Er soll mittig 2 cm unter

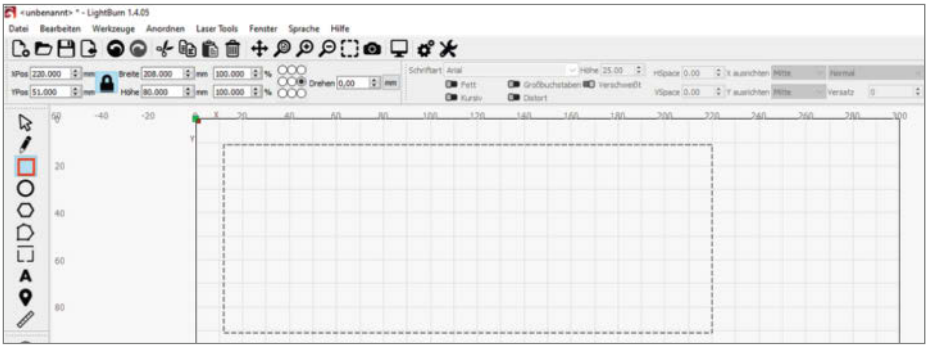


Bild 3: Versuchen Sie nicht, Maße genau mithilfe der Maus einzustellen. Das geht besser per Zahleneingabe.

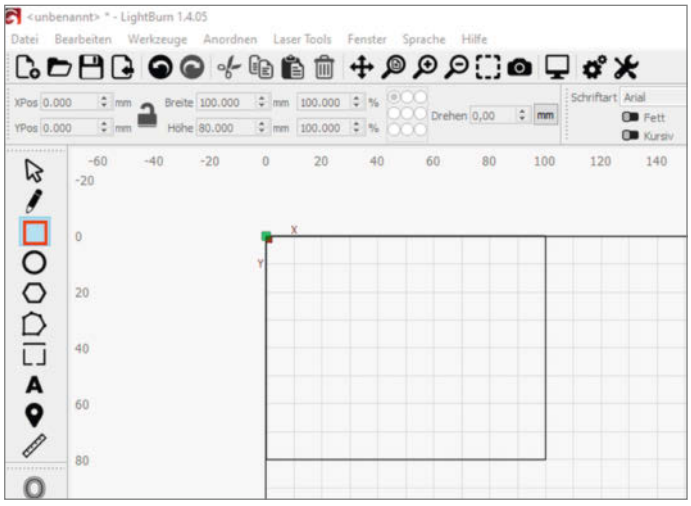


Bild 4: Das Rechteck an der richtigen Stelle

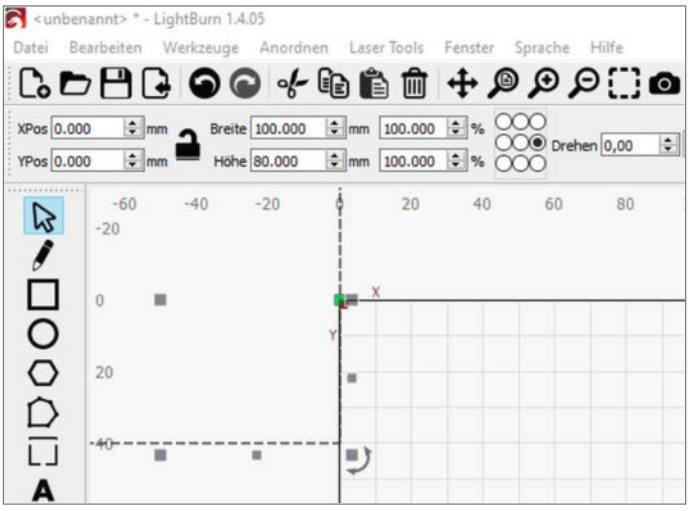


Bild 5: Das Rechteck (gestrichelt) landet zunächst außerhalb des Arbeitsbereiches.

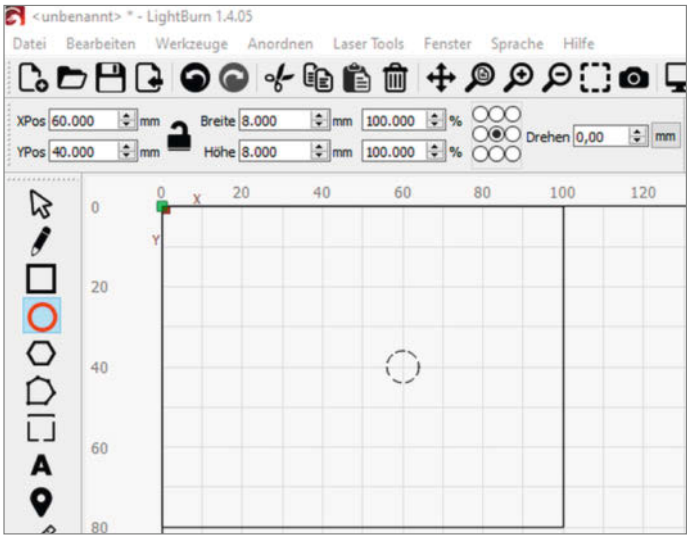


Bild 6: Das künftige Loch an der richtigen Position t

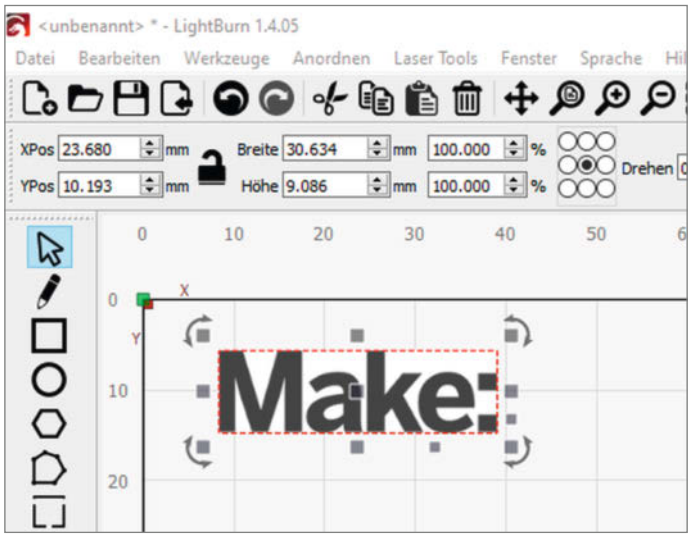


Bild 7: Der rote Rahmen zeigt, dass dem Logo eine neue Farbe zugeordnet wurde.

Bild 8: Jetzt gibt es zwei Layer

dem Achsloch stehen. Wählen Sie das Textwerkzeug, stellen eine geeignete Schrift (zum Beispiel „Arial“) und Größe (etwa 5 mm) ein. Die Position stellen Sie ähnlich wie beim Achsloch ein: Bezugspunkt Mitte, XPos 60 mm, YPos 60 mm. Die Schrift soll aber nicht graviert werden (also vollflächig gelasert), sondern nur als Umriss der einzelnen Buchstaben gezeichnet werden. Das ist im Prinzip dasselbe wie beim Schneiden einer Öffnung, aber mit geringerer Leistung, damit nur eine Linie eingebrannt, aber nicht durchgebrannt, wird. Geringere Leistung heißt aber: andere Laser-Parameter. Damit braucht der Schriftzug auch eine eigene Farbe! Stellen Sie zum Beispiel grün ein (Bild 9).

Jetzt haben wir alle Teile der kleinen Frontplatte konstruiert und es kann ans Lasern gehen. Zuvor sind aber noch die richtigen Laser-Parameter einzustellen. Das geschieht mithilfe der drei Einträge im Ebenen-Fenster. Beginnen wir mit dem schwarzen Layer, der ja für die Schnitte steht. Klicken Sie doppelt auf den Eintrag. Das Fenster des Schnitteinstellungs-Editors erscheint (Bild 10).

Stellen Sie hier entsprechend dem benutzten Material und Gerät die Leistung und Geschwindigkeit ein. Gerne vergessen wird hier das Einschalten des Airassist. Das ergäbe später Schmauchspuren auf dem Schnitt.

Tipp: Wenn Sie „Tabs/Brücken“ einschalten, werden die Teile nicht komplett ausgeschnitten, sondern es bleiben kleine Brücken (hier 0,5 mm breit), die das ausgeschnittene noch stabil im Material festhalten. Das ist sehr nützlich, da man später die Platte komplett mit allen Teilen aus dem Lasercutter entnehmen und weitergeben kann, ohne dass alles auseinanderfällt. Falls Sie das möchten, sollte die Tab-Generierung automatisch und mit gleichmäßigem Abstand erfolgen.

Schließen Sie dieses Einstellungsfenster dann mit „OK“ und klicken Sie doppelt auf den nächsten Layer-Eintrag, den für das Logo. Beim Gravieren ist erheblich weniger Leistung notwendig. Außerdem ist hier zwischen minimaler und maximaler Leistung zu unterscheiden. Beide Werte legen fest, mit welcher Leistung die helleren (minimale Leistung) bzw. dunklen (maximale Leistung) Teile des Bildes gelasert werden. Die Geschwindigkeit, mit der der Laserstrahl über das Material bewegt wird, muss beim Gravieren erheblich höher als beim

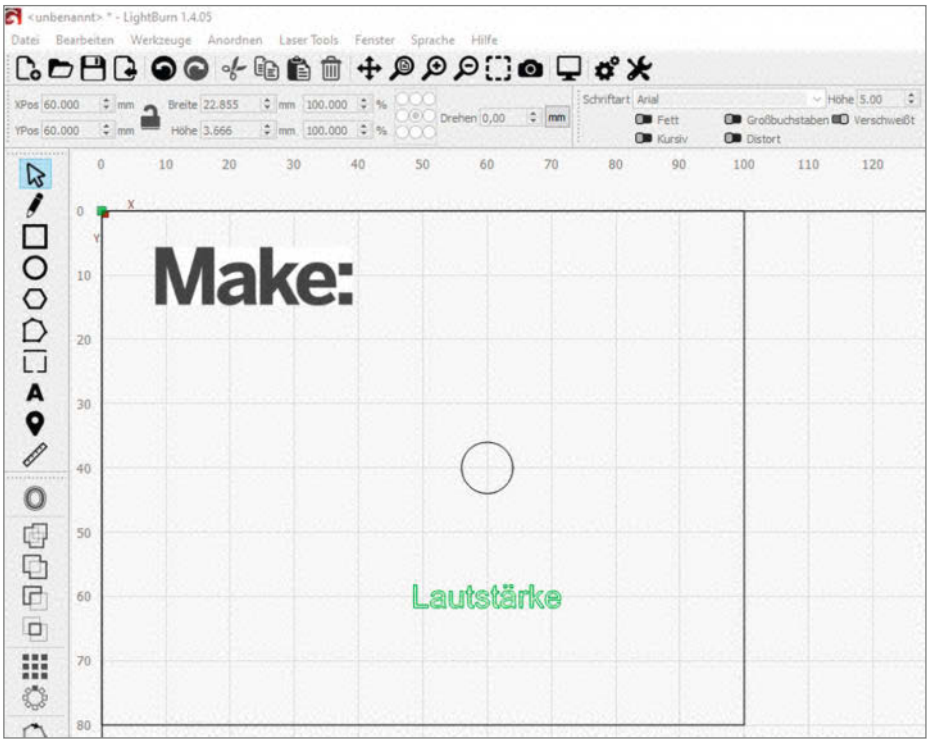


Bild 9: Die Beschriftung an der richtigen Stelle mit eigener Farbe

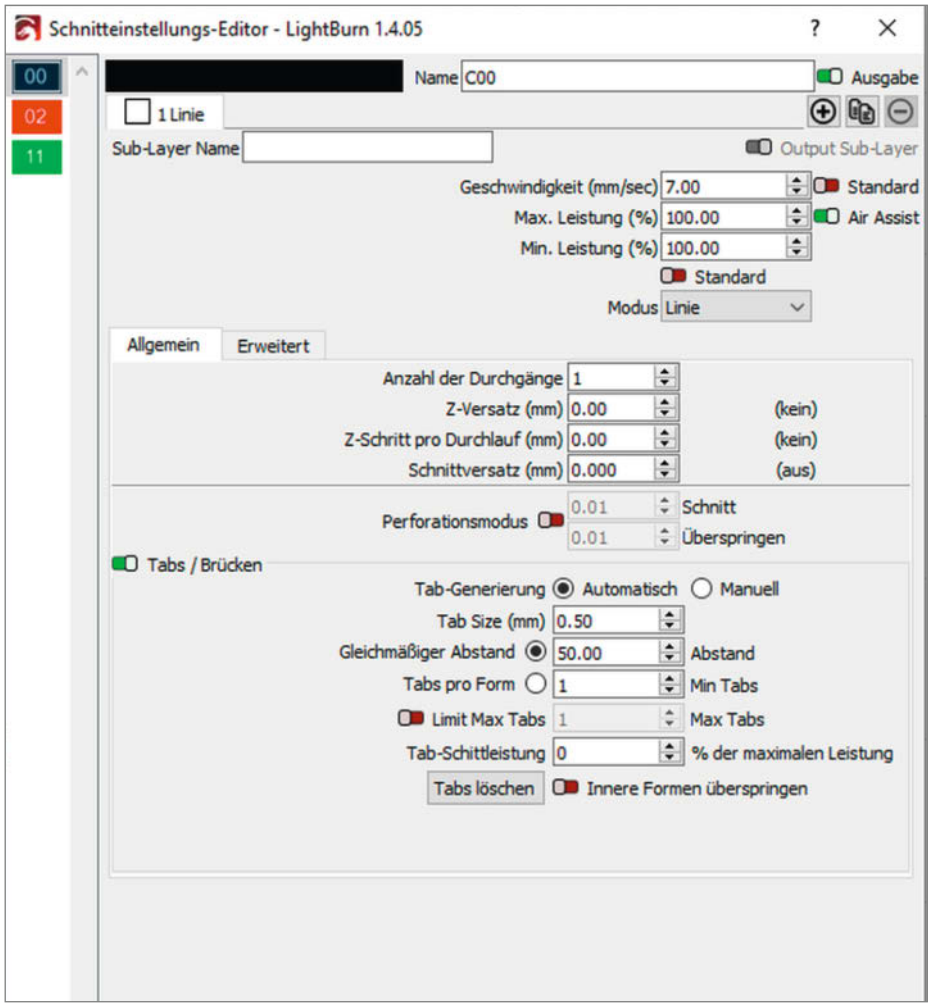
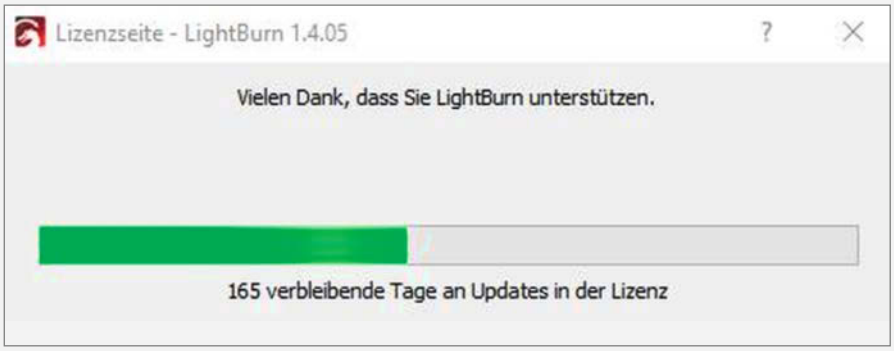


Bild 10: In diesem Fenster stellen Sie sämtliche Laser-Parameter ein.

Auswahl der richtigen Lizenz

Der Download der Lightburn-Software ist gratis. Allerdings läuft diese Version nur 30 Tage lang, um Gelegenheit zum Testen zu bekommen. Möchten Sie sie länger nutzen, ist der Erwerb eines Lizenzschlüssels im Onlineshop erforderlich. Der Schlüssel berechtigt dazu, Lightburn auf bis zu drei Computern (auch mit unterschiedlichen Betriebssystemen) zu installieren. Es sind beliebig viele Lasercutter verwendbar. Falls Sie sich einen neuen Computer kaufen, können Sie Lightburn auf dem alten deaktivieren und die Lizenz auf dem neuen weiterverwenden.



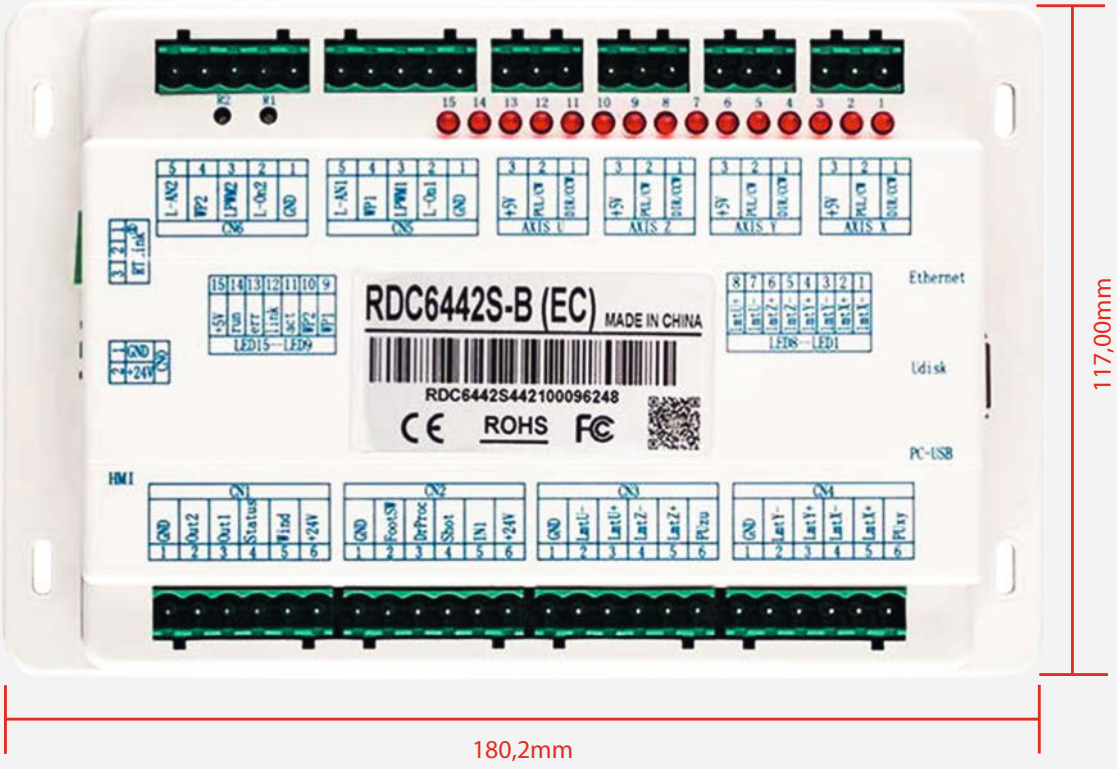
Die Software läuft ohne Zeitbegrenzung, allerdings gibt es nur ein Jahr lang Updates. Nach diesem Jahr kann man die zuletzt installierte Version beliebig lang weiternutzen. Möchte man jedoch wieder Updates erhalten, ist ein Renew der Lizenz (28 Euro) notwendig. Unter „Hilfe/Lizenzverwaltung“ erfahren Sie, wie lange die Update-Möglichkeit noch besteht.

Für Lightburn gibt es mehrere Lizenzarten, die sich nicht nur im Preis, sondern vor allem in den jeweils unterstützten Geräten unterscheiden. Genauer gesagt geht es um die in den Geräten benutzten Controller. Hier wird zwischen GCode-, DSP- und Galvo-Controllern unterschieden. Falls Sie später einmal den Lasercutter wechseln und der neue mit einer anderen Elektronik arbeitet, kann man die Lightburn-Lizenz entsprechend erweitern.

GCode-Controller benutzen eine Elektronik, die der von 3D-Druckern sehr ähnlich ist, wie diese oft auf Arduino- oder moderneren 32-Bit-Chip-Boards basiert und häufig auch dieselbe Firmware (Marlin) oder GRBL benutzt. Typische Bords sind das Smoothieboard oder Cohesion 3D. Auch die Lasercutter von Creality arbeiten mit solch einem Controller. Für diese Geräte braucht man entsprechend den GCode-Lizenzschlüssel von Lightburn (56 Euro). Bitte beachten Sie den Sonderfall K40, der keinen der gängigen Controller enthält (siehe Kasten „Sonderfall K40“).

Die meisten China-Laser verwenden DSP-Controller von Ruida oder Trocen. Die brauchen die DSP-Lizenz (112 Euro), die übrigens auch für GCode-Laser geeignet ist.

Haben Sie einen ganz modernen Faser-Laser, der bislang mit der Software EZCad2 arbeitet, ist die Galvo-Lizenz erforderlich (140 Euro).



So sieht ein Ruida-Controller aus.

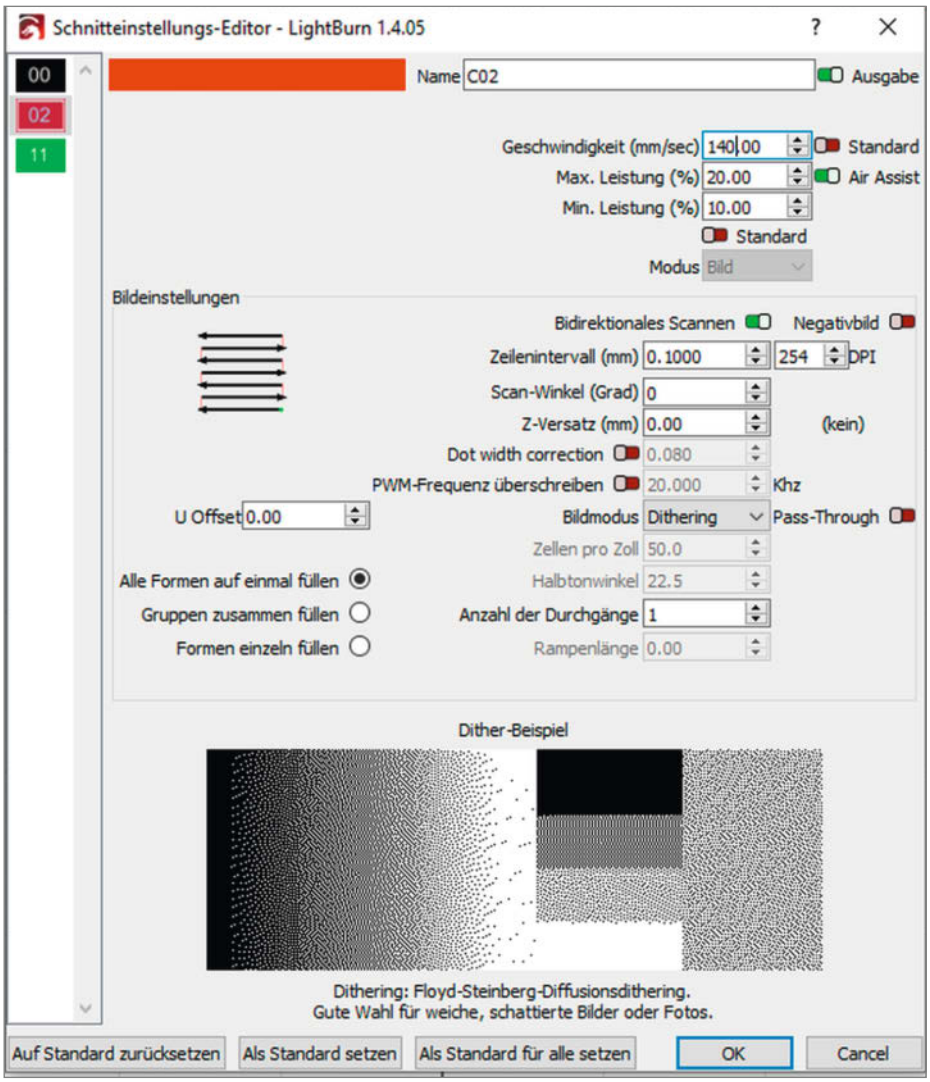


Bild 11: Die Parameter fürs Gravieren sind extrem stark von der Vorlage und dem Material abhängig.

Schneiden sein. Ein Faktor von etwa 15 bis 20 ist da ein guter Anhaltspunkt. Wenn Sie das Material also mit 7 mm/s schneiden, dann probieren Sie zum Gravieren Werte von etwa 120 bis 140 mm/s aus (Bild 11).

Die Einstellung „Bildmodus“ legt fest, wie das Bild in Laserpunkte umgesetzt wird. Hier kann man keine allgemein gültige Empfehlung aussprechen. Probieren Sie für den Anfang „Dithering“. Das liefert meist brauchbare Ergebnisse. Das Thema Bilder gravieren ist so komplex, dass wir dazu demnächst noch einen eigenen Artikel bringen werden.

Schließen Sie auch dieses Fenster mit „OK“ und öffnen Sie die Einstellungen für den dritten Layer, die Schrift. Die Linien der Schrift sollen nur angeritzt, aber keineswegs geschnitten werden. Daher sind hier für Leistung und Geschwindigkeit ähnliche Werte wie für das Gravieren erforderlich. Geben Sie nur die maximale Leistung ein, der Wert für die minimale wird automatisch entsprechend gesetzt (Bild 12).

Schließen Sie nun auch das dritte Einstellungsfenster. Jetzt wird es heiß. Legen Sie das Material in den Laser ein und starten Sie die Bearbeitung per Klick auf den Start-Button. Der Laser setzt sich in Gang und produziert die kleine Platte (Bild 13, siehe Titel).

Datei-Import

Über „Datei/Importieren“ können Sie auch Konstruktionsdateien aus anderen Quellen einlesen. Insgesamt 23 Dateiformate stehen dazu zur Verfügung, unter anderem diverse Bildformate wie jpg und png sowie Vektorformate wie svg, dxf oder ai.

Wenn Sie eine solche Datei öffnen, die mehrere Teile enthält, lassen die sich oft nur gemeinsam markieren und bearbeiten. In solchen Fällen hilft „Anordnen/Gruppierung aufheben“ weiter. Danach lassen sich die Einzelteile einzeln markieren und können so auch jeweils eigene Farben/Laser-Parameter erhalten (Bild 14). Ansonsten ist die Vorge-

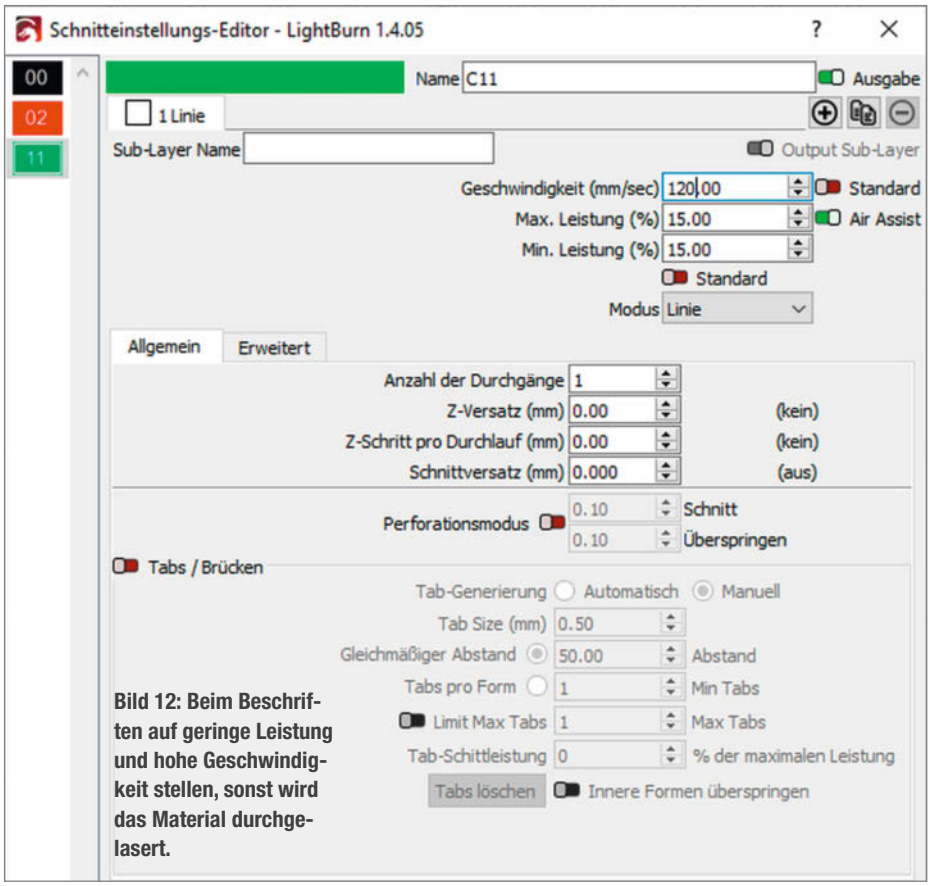


Bild 12: Beim Beschriften auf geringe Leistung und hohe Geschwindigkeit stellen, sonst wird das Material durchgelasert.

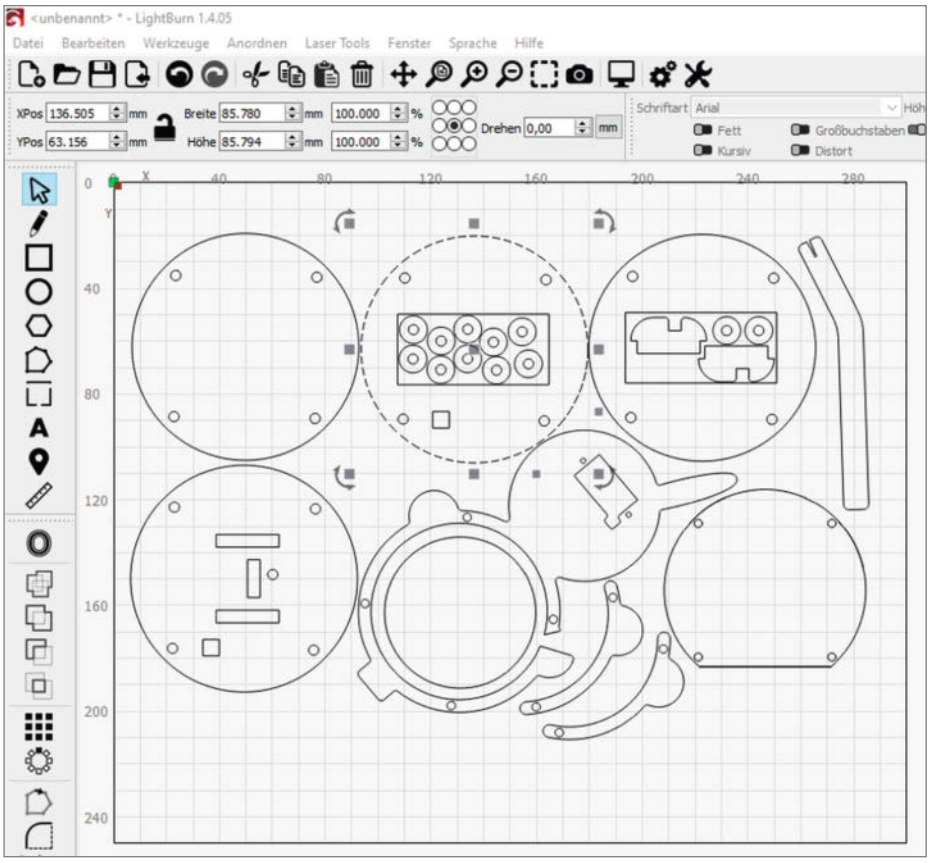
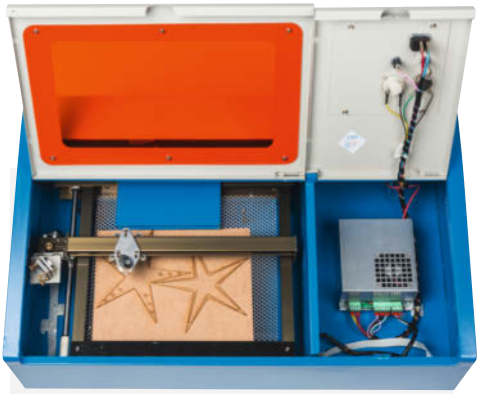


Bild 14: Wenn man die Gruppierung aufhebt, können auch in importierten Dateien die Teile einzeln bearbeitet werden und jeweils eigene Laserparameter erhalten.

hensweise mit der bei eigenen Konstruktionen identisch.

Damit haben Sie nun die grundlegende Bedienung von Lightburn kennengelernt. Doch die Software steckt voller Möglichkeiten. Ich hoffe, dass Sie möglichst viele davon bei eigenen Projekten kennenlernen können. Die würden wir dann auch gerne hier in der Make vorstellen. Scheuen Sie sich also nicht, zu konstruieren, zu lasern und als Autor bei der Make Ihr Projekt einzureichen. Viel Spaß dabei! *—hgb*

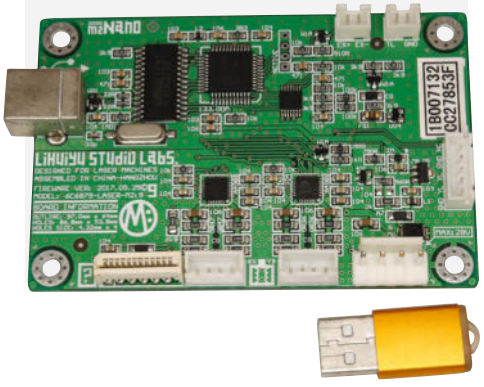


Sonderfall K40

Der billige Lasercutter K40 mit seinem etwa DIN-A4-großen Arbeitstisch arbeitet mit einem exotischen Controller, den Lightburn nicht unterstützt.

Möchten Sie solch einen Cutter mit Lightburn benutzen, kommen Sie um den Austausch des Controllers nicht herum. Im Artikel „Modernisierung für den Lasercutter“ (siehe Kurzinfo) finden Sie eine Umbauanleitung auf einen Ruida-Controller, den ich in diesem Artikel auch benutzt habe. Im Netz gibt es aber auch zahlreiche Anleitungen zum Umbau auf einen preiswerteren GCode-Controller. Sie brauchen für einen so umgebauten K40 dann entsprechend die DSP- oder GCode-Lizenz.

Der serienmäßige Controller des K40 braucht zum Betrieb der Software auch noch einen USB-Dongle.



Make: Digital

Beliebt auf heise+

Rossmann-Hack: Personenwaage smart machen

Übliche Haushaltsgeräte ins Smart Home einbinden: Wir zeigen Ihnen, wie Sie eine handelsübliche Personenwaage Schritt für Schritt smarter machen.

heise+ 03. Januar 2024, 09:00 Uhr 94

Garagentorantrieb mit ESPHome und Home Assistant smart machen

Da oftmals noch alte Garagentorantriebe im Einsatz sind, zeigen wir Ihnen, wie man so einen Antrieb für die Hausautomatisierung mit Home Assistant fit macht.

heise+ 17. Januar 2024, 10:00 Uhr 82

Bastelprojekt: Taupunkt-Lüftungssystem bauen und Keller trockenlegen

Viel besser ist es daher, bei der Entscheidung über den Lüftzeitpunkt auf den **Taupunkt** zu schauen, denn der spiegelt den absoluten Wassergehalt einer Luftmasse

YouTube: Einfach eine andere CPU in die IKEA-Leuchte einbauen

In der Make-Ausgabe 6/23 haben wir verschiedene Projekte rund um das Thema Reparieren und Pflegen vorgestellt. Ein besonders schönes Beispiel war der Ikea Hack, bei dem aus einer OBEGRÄNSAD Lampe eine faszinierende Pixelclock entstand. Dieses Projekt wurde von Johannes in unserem derzeit sehr beliebten YouTube-Video nachgebaut. Auch ohne viel Löt- und Elektronikerfahrung lässt sich die IKEA-Lampe zur Pixelclock umbauen. Im Video zeigen wir Schritt für Schritt, wie man mit einer Heißluftstation den eingebauten IC einfach auslötet und durch einen eigenen ESP ersetzt. —*dus*

▶ www.youtube.com/@MakeMagazinDE

Weitere aktuelle Videos:

Bleib informiert:

www.make-magazin.de

@makemagazinde

Instagram: @makemagazinde
@makerfairedeutschland

Facebook: @makemagazin.de

X (Twitter): @MakeMagazinDE

YouTube: @MakeMagazinDE

TikTok: @makemagazinde

Bluesky: @makemagazin.de

WhatsApp Channel:
Make Magazin Deutschland

Mastodon: @MakemagazinDE
@social.heise.de

GitHub: MakeMagazinDE

Threads: @makemagazinde

oder disktutiere in unseren Foren online über Themen und Artikel:
www.make-magazin.de/forum

Nichts mehr verpassen: Abonniere unseren Newsletter! weitere Infos unter:
www.maker-faire.de

Wo finde ich die Make am Kiosk?
www.mykiosk.com

Kamera für Lightburn

Ob man nun eine Sperrholzplatte bis zum letzten Millimeter ausnutzen, ein bereits bearbeitetes Teil um Gravuren und/oder zusätzliche Schnitte ergänzen oder winzige Teile mit dem Laser beschriften möchte, es kommt auf Präzision an. Hier hilft eine Kamera über dem Schneidetisch, deren Bild in der Lasercut-Software Lightburn eingeblendet werden kann. Damit schieben Sie dann Schnittlinen und Gravuren millimetergenau aufs Material.

von Heinz Behling

Lasercutter sind an sich hochpräzise Werkzeuge, mit denen sich Schnitte, Beschriftungen und Gravuren mit einer Genauigkeit von deutlich unter einem Millimeter ausführen lassen. Nur das Einlegen des zu bearbeitenden Materials und damit die Positionierung der Schnitte und Gravuren darauf geschieht mit der Methode Versuch und Irrtum. Selbst wenn der Lasercutter ein rotes Laserlicht als Positionierhilfe anbietet, zeigt dieses nur selten wirklich auf den Punkt, den der Schneidstrahl treffen wird.

Dabei kann das genaue Positionieren den Verschnitt minimieren. Und das nachträgliche Bearbeiten bereits vorgefertigter Teile, ob Handy-Hülle oder Gehäuse, ist nur dann erfolgreich, wenn die nachträglichen Schnitte und Gravuren auch genau dort sitzen, wo man sie sich wünscht. Sonst landet eine Gehäuseöffnung für einen zusätzlichen Schalter kaum in derselben Linie wie die bereits vorhandenen. Und ganz kleine Teile treffsicher zu lasern ist so gut wir unmöglich.

In diesem Artikel gibt es den Ausweg: Eine Kamera wird über dem Arbeitstisch des Lasercutters fest angebracht. Ihr Bild wird dann in den Arbeitsbereich der Lasercutter-Software Lightburn eingeblendet, sodass man gleichzeitig die Konstruktionszeichnung und den Schneidetisch mit dem darauf liegenden Material sehen kann. Das Ausrichten ist so wirklich einfach.

Dieser Artikel geht davon aus, dass Lightburn bereits auf Ihrem Computer installiert und für Ihren Lasercutter konfiguriert ist. Falls noch nicht geschehen, finden Sie Informationen darüber im Artikel auf Seite 24.

Kurzinfo

» Kamera über Lasercutter-Schneidetisch montieren
» Auswahl der Kamera und Bestimmung des Abstands
» Justierung des Kamerabildes in die Lightburn-Oberfläche

Checkliste

Zeitaufwand:
2 bis 3 Stunden

Kosten:
50 Euro

Material

» **USB-Kameramodul 5 Megapixel**
Bezugsquellen siehe Kurzinfo-Link
» **1 m Alu-Profil 20 × 20 mm**
» **Kunststoff-Verbindungswinkel**
passend zu Alu-Profil
» **Schrauben M4 × 30 mit Muttern**
» **Sperrholz oder Finnpappe**
3 mm für Kamerahalter oder
» **optional: 3D-gedruckter Kamerahalter**

Werkzeug

» **Metallsäge**
» **Bohrmaschine mit Metall-Bohrern**
» **optional: 3D-Drucker**
» **Hammer**

Mehr zum Thema

» Heinz Behling: Autofokus-Wabentisch für Lasercutter K40, Make 2/21, S. 110
» Heinz Behling: Modernisierung für den China-Laser, Make 1/21, S. 112
» Heinz Behling: Laser für Leser, Make 3/18, S. 16

Alles zum Artikel im Web unter
make-magazin.de/xx8u

Die richtige Kamera

Bei der Auswahl des Kameramoduls bietet uns Lightburn eine Hilfe an (unter „Hilfe/Hilfe zur Kameraauswahl", Bild 1). In der dort angezeigten Tabelle erfahren Sie unter Berücksichtigung der Größe des Arbeitsbereichs Ihres Lasercutters, welche Kamera (Auflösung/Blickwinkel) in welcher Höhe über dem Schneidetisch zu montieren ist. Damit die Sache gut funktioniert, sollte die Kamera mindestens 5 Megapixel Auflösung haben. In meinem Fall hat der Laser einen Arbeitsbereich von 300 × 250 mm. Die Kamera hat eine Auflösung von 2592 × 1944 Pixeln. Waagerecht entspricht somit ein Pixel etwa 300 / 2592 = 0,16 mm, senkrecht sind es 250 / 1944 = 0,13 mm. Es sollte also später möglich sein, das Schnittmaterial mit einer Genauigkeit von unter einem Millimeter auf der Arbeitsfläche zu positionieren. Dies ermöglicht dann sogar die Beschriftung solch kleiner Dinge wie Zahnstocher oder Ähnliches.

Zudem ist ein fest einstellbares Objektiv notwendig. Autofokus ist nicht verwendbar, denn durch das Scharfstellen wird der Abbildungsmaßstab geringfügig verändert. Damit ist dann aber die genaue örtliche Zuordnung zwischen Kamerabild und Lightburn-Arbeitsfläche nicht mehr möglich.

Diese Hilfe hat Lightburn übrigens nicht so ganz uneigennützig ins Programm eingebaut: Wenn Sie auf einen der Tabelleneinträge klicken, gelangen Sie zum Kamerashop von

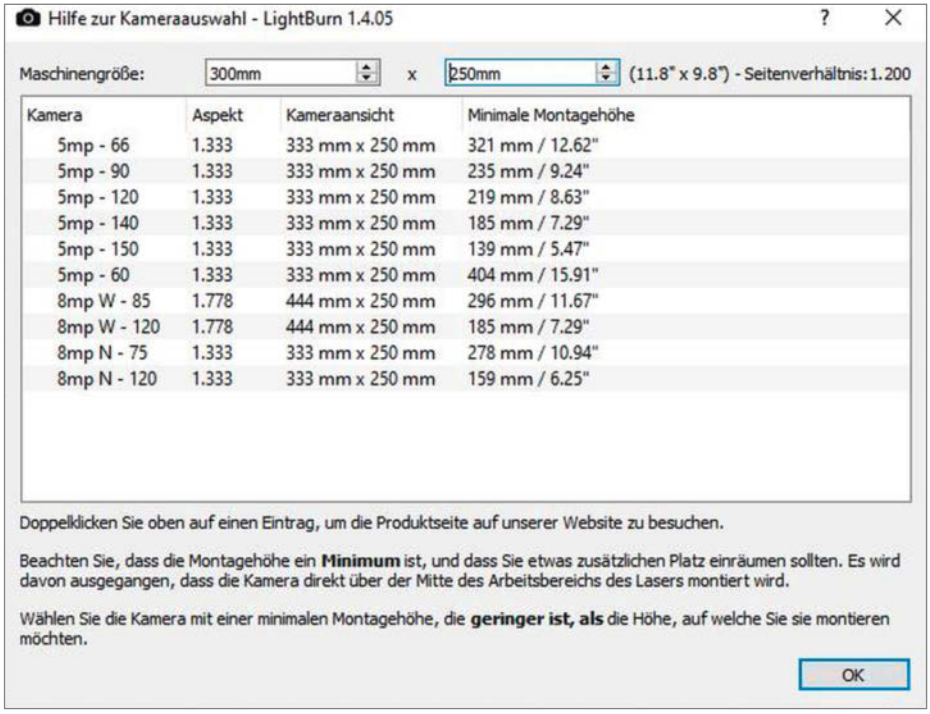


Bild 1: Diese Tabelle zeigt, wie viel Platz Sie über dem Schneidetisch brauchen, damit die Kamera den kompletten Arbeitsbereich sehen kann.

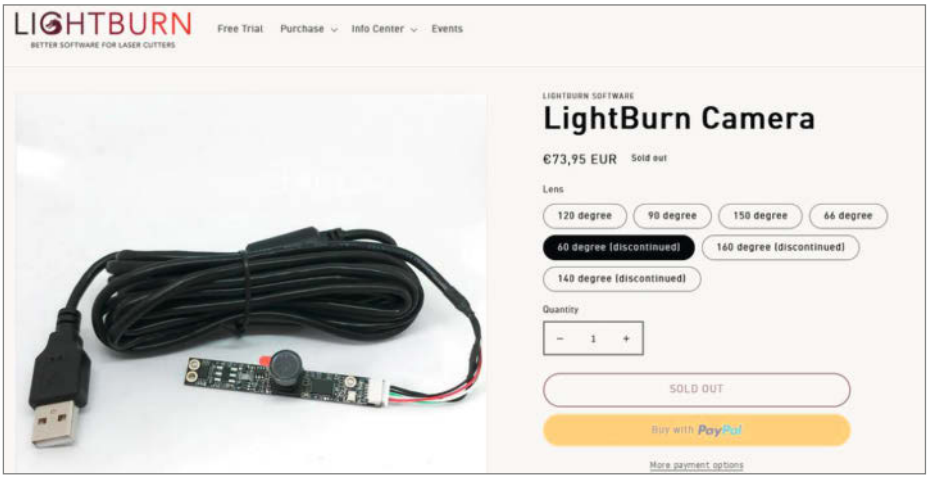


Bild 2: Die Kameramodule sind im Lightburn-Shop recht teuer.

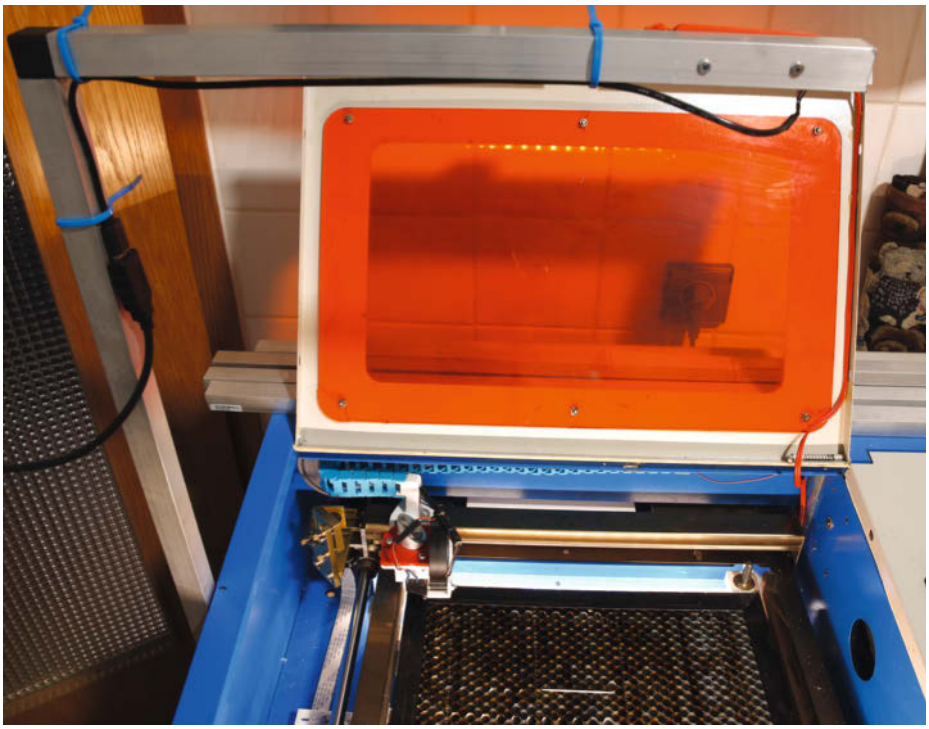


Bild 3: Der Kameragalgen aus Vierkant-Alurohr

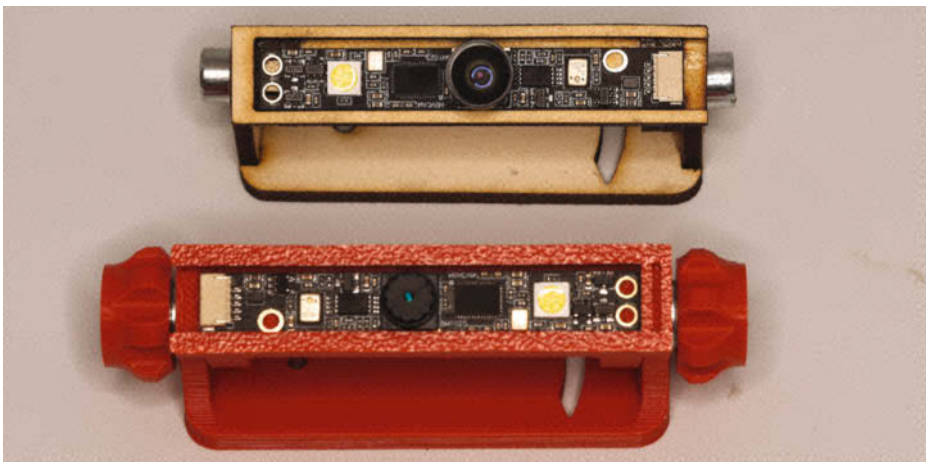


Bild 4: 3D-Druck- und Lasercut-Variante des Kamerahalters

Lightburn, wo Sie das jeweilige Kameramodul bestellen können. Die Preise sind aber recht hoch (Bild 2) und liegen inklusive Versandkosten bei 80 Euro. Gleichwertige Module (Bezugsquellen sie Kurzinfo-Link) kosten im Online-Handel nur die Hälfte.

Der Blickwinkel der Kamera sollte möglichst klein sein, denn das ergibt die geringsten optischen Verzerrungen im Randbereich. Die kann Lightburn zwar in gewissem Maße ausgleichen, aber das bringt in den Außenbereichen eine etwas geringere Genauigkeit. Um eine konkrete Kamera auszuwählen, müssen Sie als Erstes ausmessen, wie viel Platz über dem Lasercutter zur Verfügung steht, um die Kamera an einem Stativ zu befestigen. Wählen Sie dann das Kameramodul mit dem kleinsten Blickwinkel, das sich noch über dem Laser unterbringen lässt. Eine Auflösung von 5 Megapixeln reicht bis zu einer Schneidetischgröße von 50 × 70 cm völlig aus. Bei größeren Geräten sollten 8 Megapixel verwendet werden.

Falls Sie jetzt auf die Idee kommen sollten, das Kameramodul an der Innenseite des Lasercutter-Deckels zu montieren, vergessen Sie es. Mehrere Gründe sprechen dagegen. Die Scharniere und Anschläge des Deckels sind nicht spielfrei. Das bedeutet, die Kameraposition wäre bei jedem Öffnen anders. Damit ließe sich keine exakte Positionierung des Kamerabilds in Lightburn erreichen. Zweitens läge die Kamera beim Betrieb des Lasers im Inneren des Schneideraums und würde den Rauch dort mitbekommen. Dessen Ablagerungen auf den Objektivlinsen würden das Kamerabild unscharf und kontrastarm machen. Und zum dritten klappt der Deckel meist nach hinten und seine Vorderkante steht oft nicht über der Mitte des Schneidetisches.

Deshalb sollte nur ein externes Stativ verwendet werden, das am Lasercutter befestigt wird. Das Stativ am Tisch/Regal/Schrank unter dem Cutter anzuschrauben, ist schlecht, da der Lasercutter sich gegenüber diesem Untergrund verschieben kann und dann die Kameraposition verschoben würde. Ich empfehle ein aus Alu-Vierkantrohren und passenden Kunststoff-Steckverbindern (Bezugsquellen siehe Kurzinfo-Link) gefertigtes L-förmiges Stativ, das am unteren Ende mit stabilen Schrauben (M5) fest an der Seite des Lasercutters verschraubt ist und dessen waagerechter Schenkel mittig über dem Arbeitsbereich steht (Bild 3).

Die Maße der beiden Alu-Schenkel und deren Position sind vom jeweiligen Lasercutter abhängig. Das Ausmessen der Mitte des Schneidetisches ist aber einfach. Das waagerechte Rohr muss mindestens 5 cm über die Mitte des Schneidetisches hinausragen. Die Länge des senkrechten Rohres sollte so sein, dass das obere Ende mindestens 10 cm über der mindestens erforderlichen Montagehöhe

der Kamera über dem Schneidetisch liegt. Diese Mindesthöhe finden Sie in der Tabelle aus Bild 1. Außerdem darf das Öffnen des Deckels nicht behindert werden. Bei meinem Cutter verwende ich eine 5-Megapixel-Kamera mit 60° Blickwinkel. Die Höhe über dem Schneidetisch beträgt 53 cm, also knapp 13 cm über der Mindesthöhe.

Die Kamera selbst wird dann mit einem Selbstbau-Halter (Bild 4, Lasercut- und 3D-Druck-Dateien siehe Kurzinfo-Link) am Stativ befestigt.

Das Kameramodul wird mit dem Computer, auf dem Lightburn läuft, per USB-Kabel verbunden. Dann muss es so über dem Lasercutter ausgerichtet werden, dass der Schneidebereich scharf, gerade und vollständig im Bild steht (Bild 5). Das geht auf Windows-PCs beispielsweise gut mit der Kamera-App. Zum Scharfstellen muss das Objektiv geringfügig gedreht werden.

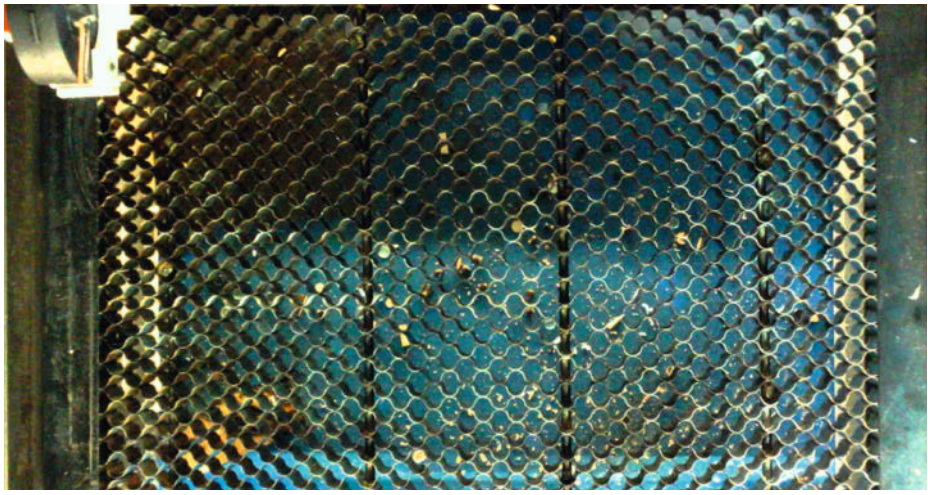


Bild 5: Fürs Ausrichten und Scharfstellen mit der Kamera-App sollten Sie sich Zeit nehmen, damit wie hier der Wabentisch vollständig und gleichmäßig scharf erscheint.

Abbildungsfehler messen in Lightburn

Hat die Kamera den Arbeitsbereich des Lasers im Blick, können wir nun in Lightburn weiterarbeiten. Zunächst müssen die Verzerrungen gemessen werden, die das Kameraobjektiv verursacht. Die ermittelten Verzerrungen werden dann später berücksichtigt, wenn im Programm das Kamerabild über den Arbeitsbereich des Programms gelegt wird.

Lightburn stellt dazu ein Assistentenprogramm bereit, das Sie unter „Laser Tools/Kameralinse kalibrieren“ finden. Im Assistentenfenster wählen Sie zunächst die richtige Kamera aus. Daraufhin erscheint das aktuelle Kamerabild (Bild 6). Falls Sie die richtige Kamera nicht finden können, kontrollieren Sie das USB-Kabel. Nach Verbinden mit der Kamera kann es außerdem bis zu einer Minute dauern, bis die Kamera eingerichtet ist. Haben Sie also ein wenig Geduld.

Bild 6: Wählen Sie die Laser-Kamera, stellen Sie das richtige Objektiv ein und klicken Sie auf weiter.

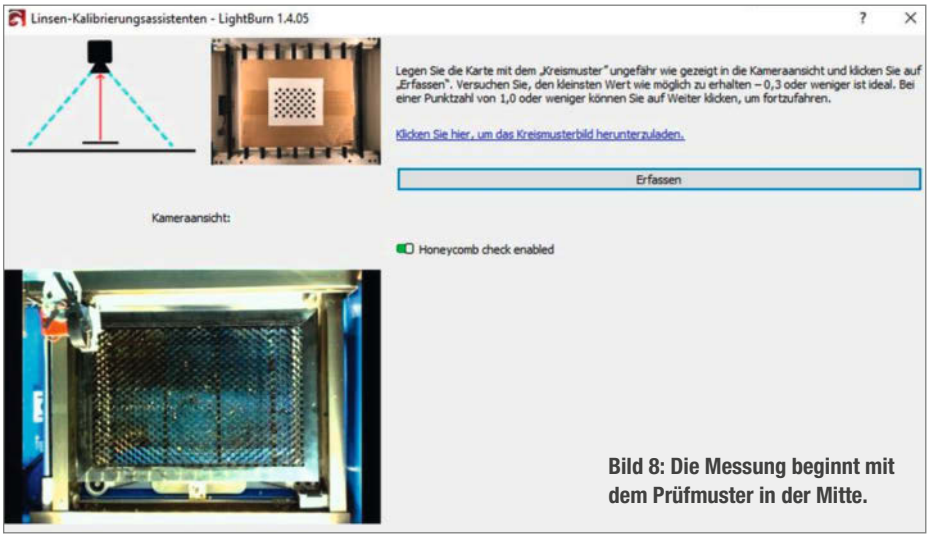


Bild 8: Die Messung beginnt mit dem Prüfmuster in der Mitte.

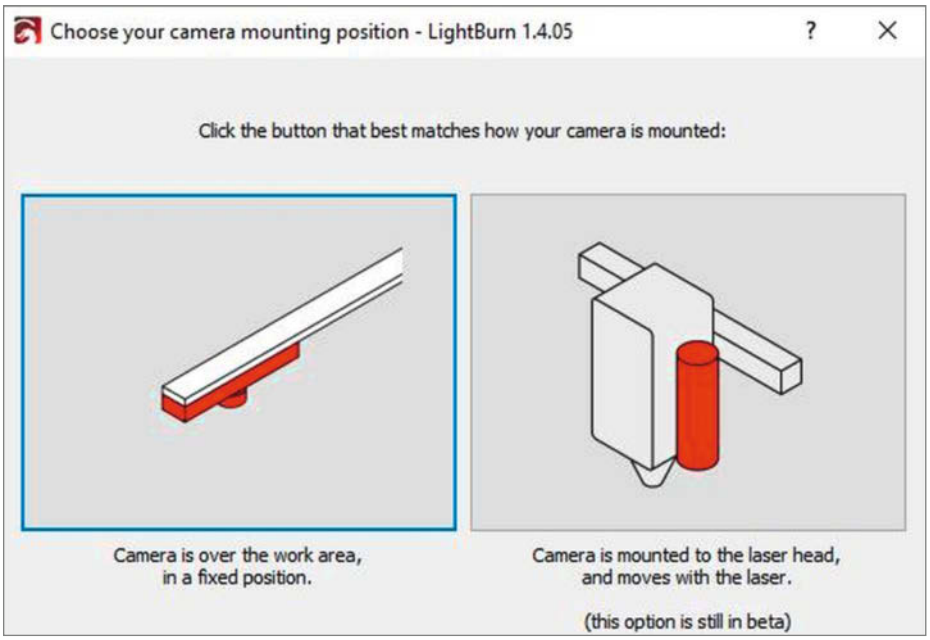


Bild 9: Die Kamera ist an einem Stativ über dem Lasercutter befestigt. Daher ist das linke Feld anzuklicken.

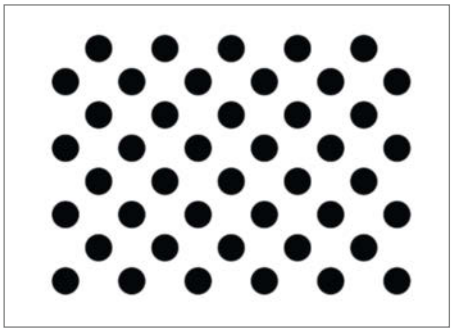


Bild 7: Diese Punkte sind das Prüfmuster für die Kalibrierung.

Zur Verzerrungsmessung benötigt man ein Prüfbild, das Ihnen der Assistent zum Download anbietet (Bild 7). Nach dem Download müssen Sie es in guter Qualität drucken (am besten auf Papier mit 160g/m², aber kein glänzendes Papier). Die Datei ist aber für große Lasercutter mit einem Arbeitsbereich von etwa 50 × 70 cm gedacht. Für kleinere Cutter muss das Prüfmuster kleiner gedruckt werden. Faustregel: Die Breite des Prüfmusters sollte maximal ein Drittel der Breite des Schneidetisches sein. Über den Kurzinfo-Link können Sie ein für den K40 passendes Format erhalten.

Falls Ihr Lasercutter einen Wabentisch hat, müssen Sie den mit Pappe o. ä. abdecken, denn das Wabenmuster würde die folgenden Messungen stören. In neun Schritten erfolgt die Verzerrungsmessung. Dazu muss das Prüfmuster jeweils neu platziert werden, beginnend in der Mitte des Schneidetisches. Der Assistent zeigt Ihnen jeweils die erforderliche Position (Bild 8).

Mit einem Klick auf „Erfassen" beginnt die Messung. Als Ergebnis erscheint ein Score, der unter 0,3 liegen muss, besser noch unter 0,2. Falls das nicht erreichbar ist, stimmt meist die Beleuchtung nicht. Am besten ist eine indirekte Beleuchtung, wie sie in vielen Lasercuttern eingebaut ist. Am schlechtesten ist eine Beleuchtung von oben. Verbessern Sie, falls nötig, die Beleuchtung und wiederholen Sie die Messung bis der Score ausreicht. Erst dann klicken Sie auf „Weiter".

Der Assistent zeigt Ihnen dann die nächste Position fürs Prüfmuster. Gehen Sie so alle neun Messungen durch. Die Reihenfolge der 9 Messstellen ist:

1. Zentrum,
2. Mitte vorn,
3. Mitte links,
4. Mitte rechts,
5. Mitte hinten,
6. rechts vorn,
7. links vorn,
8. links hinten,
9. rechts hinten.

Sollte Ihnen dabei der Schneidekopf im Weg sein, fahren Sie ihn zur Seite. Achten Sie darauf,

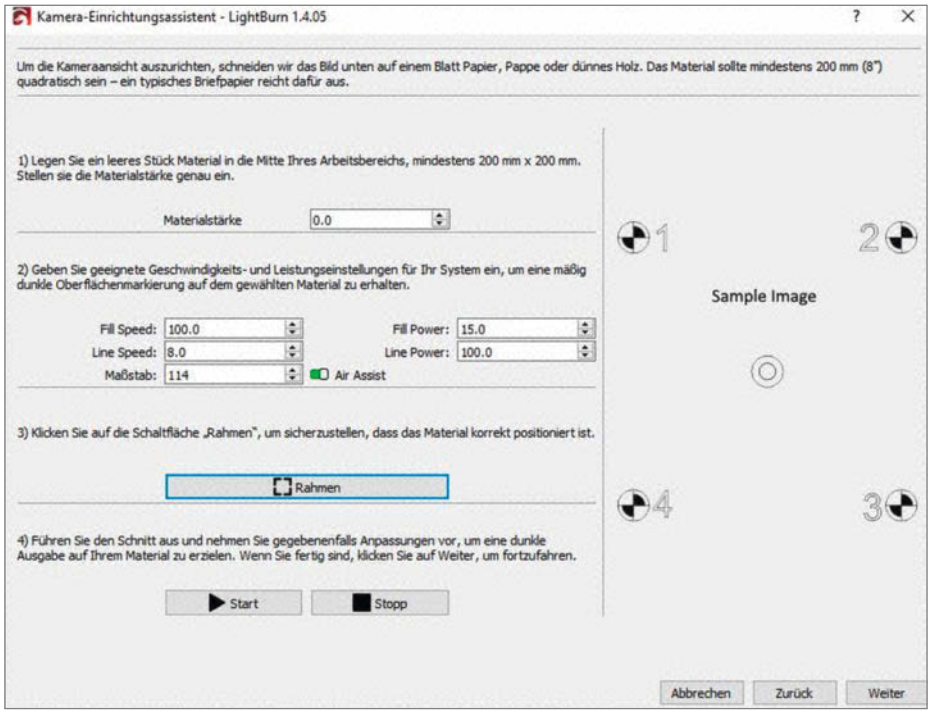


Bild 10: Beim K40 muss das zweite Prüfmuster auf 80 Prozent verkleinert werden.

dass auch die Leitungen zum Kopf nicht über dem Prüfmuster liegen. Schließlich erhalten Sie die Erfolgsmeldung. Beenden Sie den Assistenten.

Kamerabild ausrichten

Auch dafür hat Lightburn einen Assistenten, den Sie unter „Laser Tools/Kameraausrichtung kalibrieren" finden. Zunächst müssen Sie wählen, ob Ihre Kamera über dem Cutter oder direkt am Schneidekopf befestigt ist. In diesem Beispiel ist die Kamera über dem Gerät (Bild 9). Danach ist noch die richtige Kamera zu wählen und das von ihr gesendete Bild erscheint im Fenster. Nach einem Klick auf „Weiter" kommen Sie in das Einstellungsfenster für das Ausrichtungs-Prüfmuster. Das müssen Sie mit dem Laser erst noch gravieren. Dazu ist eine laserfähige Platte mit mindestens 200 × 200 mm Größe erforderlich. Ich empfehle dazu Finnpappe, denn darauf ist

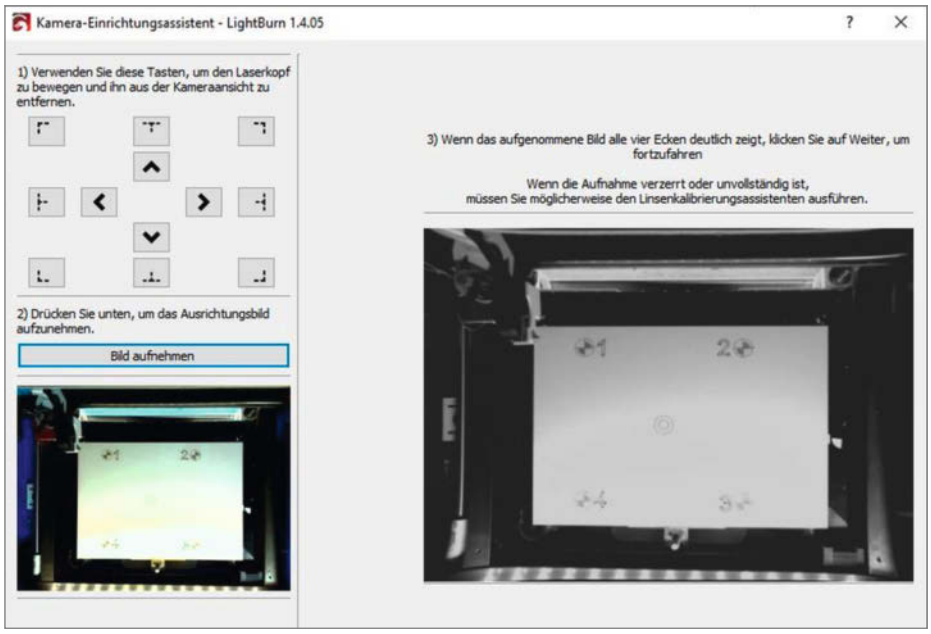


Bild 11: Falls der Laserkopf Teile verdeckt, verschieben Sie ihn mit den Pfeiltasten.

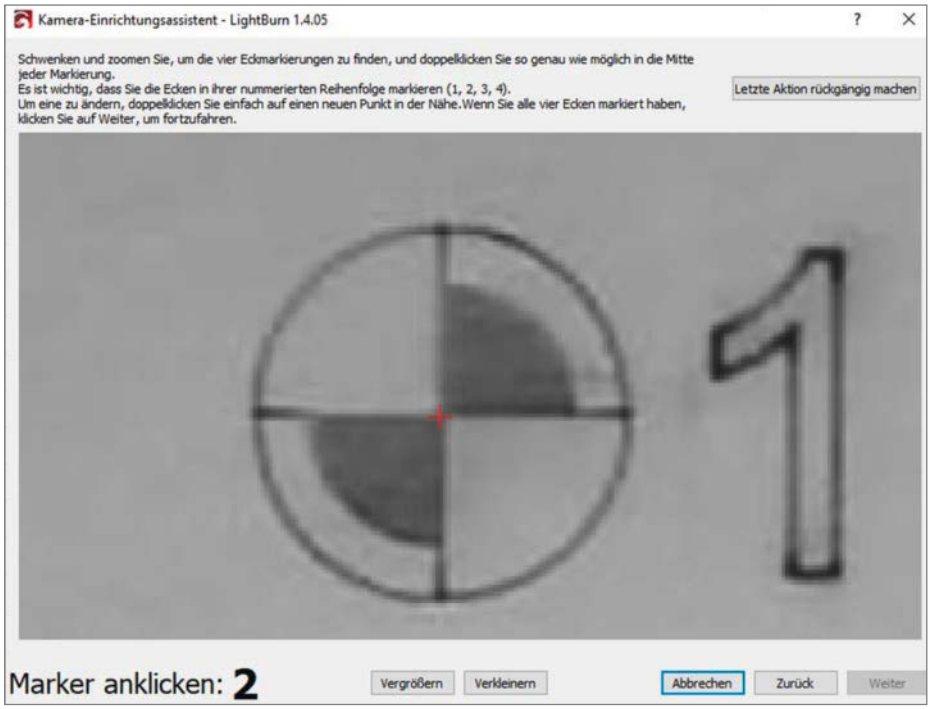


Bild 12: Die Mitte muss möglichst genau getroffen werden.

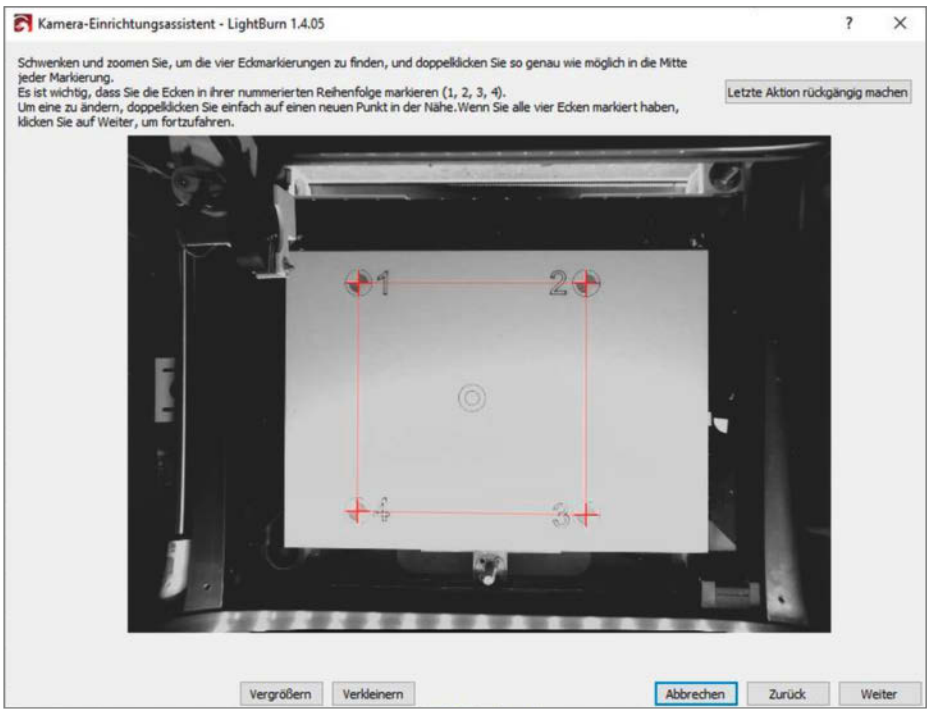


Bild 13: Sobald dieses Quadrat erscheint, ist die Ausrichtung kalibriert.

Bild 14: Das Kamerasymbol rechts blendet das Bild aus dem Laser ein.

das Gravurmuster gut zu erkennen und sie liegt eben.

Zu den Einstellungen (Bild 10): Da ist zunächst das Feld für die Materialstärke. Haben Sie einen Lasercutter, bei dem die Fokussierung durch vertikales Verschieben des Laserkopfes geschieht (z. B. Diodenlaser), müssen Sie hier die Stärke der Platte eingeben, denn die Oberfläche der Platte liegt ja hier einige Millimeter näher an der Kamera. Findet die Fokussierung hingegen durch Absenken des Arbeitstisches statt, bleibt der Abstand der Plattenoberfläche zur Kamera gleich. Deshalb muss hier eine 0 stehen.

Für kleine Cutter wie dem K40 muss das Prüfmuster verkleinert werden. Das geschieht mit dem Wert 80 beim Eintrag „Maßstab". Umgekehrt sollte man für große Geräte den Wert erhöhen. Dann braucht man aber auch eine größere Materialplatte zum Lasern.

Schließlich stellen Sie noch die Geschwindigkeiten und Leistungswerte fürs Gravieren und Linienzeichnen ein. Die entsprechen dem, was Sie auch bislang bei diesen Funktionen auf Ihrem Laser benutzt haben. Ist alles eingestellt, kontrollieren Sie, ob Laser, Kühlung und Abluftanlage eingeschaltet sind und der Laser geschlossen ist. Per Klick auf „Start" wird das Prüfmuster gelasert.

Mit „Weiter" erhalten Sie nun das Kamerabild vom gerade gefertigten Muster. Falls der Laserkopf Teile davon verdeckt, können Sie ihn mit den Pfeiltasten oben links aus dem Weg räumen (Bild 11). Dann geht es weiter.

Auf dem Bild sehen Sie die Platte mit dem Prüfmuster. Zoomen Sie nun auf die mit 1 gekennzeichnete Ecke (mit Mausrad zoomen). Ihre Aufgabe: Klicken Sie doppelt auf den Schnittpunkt der senkrechten und waagerechten Linie im Kreis (Bild 12). Dort erscheint ein rotes Kreuz.

Das wiederholen Sie dann mit den anderen drei Ecken. Welche gerade dran ist, steht im Fenster. Nach den vier Klicks erscheint im Kamerabild ein rotes Quadrat, dessen Ecken genau auf den markierten Punkten liegen (Bild 13).

Mit „Weiter" und „Abschließen" wird der Assistent beendet.

Anwendung

Jetzt ist die Kamera einsatzbereit. Sie können es testen. Ich habe dazu als Material wie zu Beginn bereits erwähnt, einen Zahnstocher verwendet und ihn mit dem Schriftzug „Make:" beschriftet. Den Zahnstocher einlegen und in der Menüzeile von Lightburn auf das Kamerasymbol klicken (Bild 14).

Dadurch wird das Kamerabild unter das Arbeitsfeld in Lightburn eingeblendet (Bild 15).

Jetzt kann man alles, was man schneiden oder gravieren will, passgenau platzieren, zum Beispiel die von mir gewollte Beschriftung. Mit

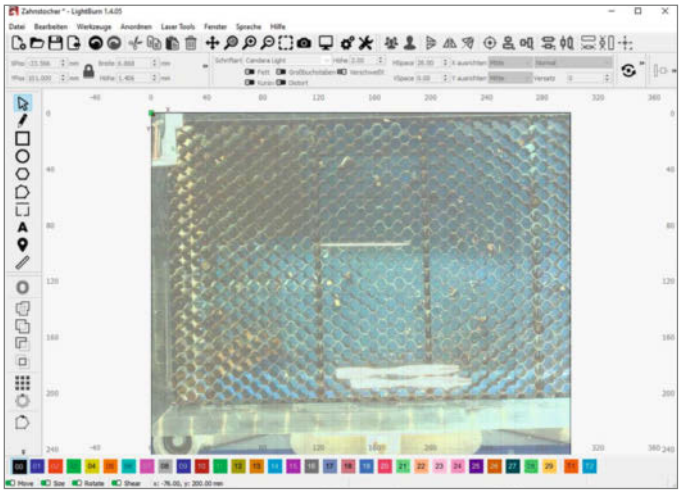


Bild 15: Das Bild des Schneidetisches ist im Arbeitsbereich von Lightburn eingeblendet.

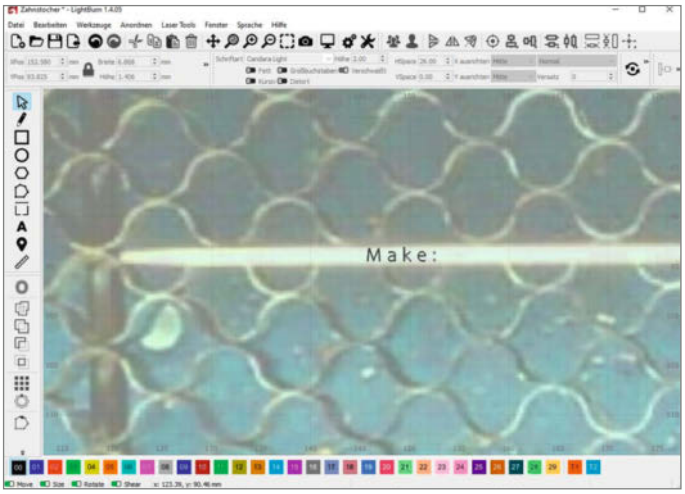


Bild 16: Die Schrift ist auf Zehntel-Millimeter genau auf dem Zahnstocher platziert.

dem A-Symbol aktiviere ich die Textfunktion, stelle in der oben Einstellleiste die Schrifthöhe auf 2.00 mm und wähle eine Schrift mit möglichst dünnen Linien, zum Beispiel „Candara light“. Mit einem HSpace von 26 sorge ich für etwas mehr Abstand zwischen den Buchstaben. Anschließend aktiviere ich mit dem Pfeil die Markierungs- und Verschiebefunktion. Mit der Maus setze ich die Schrift genau auf den Zahnstocher (Bild 16). Dabei sollte man die STRG-Taste gedrückt halten, denn nur so lässt sich die Schrift in Millimeter-Bruchteilen verschieben. Mit dem Mausrad kann man das Bild vergrößern.

Jetzt kann gelasert werden. Stellen Sie die Leistung und Geschwindigkeit entsprechend ein (Bild 17). Dann starten Sie den Laser und einige Sekunden später erhalten Sie einen beschrifteten Zahnstocher. Mit etwas Übung können Sie sogar längere Text exakt platzieren (Bild 18).

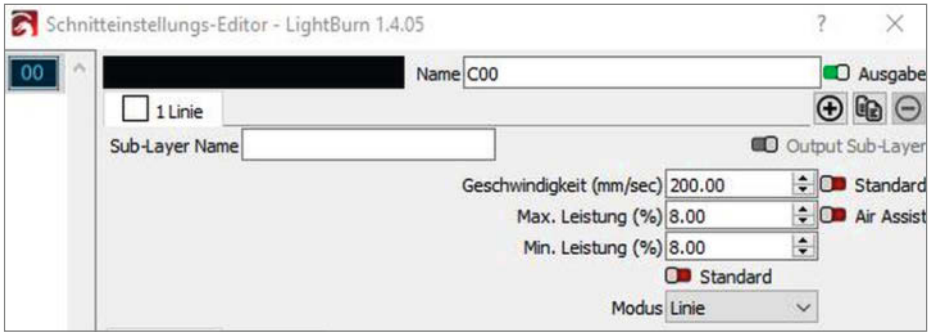


Bild 17: Bei solch kleinen Objekten sollten Sie die Leistung sehr gering einstellen und ein Air Assist ausschalten, sonst weht der Zahnstocher weg.

Wozu das Ganze?

Zugegeben, man kommt nur selten in die Verlegenheit, Zahnstocher beschriften zu müssen. Aber er erschien mir als Beispiel für die Möglichkeit der präzisen Platzierung von Schnitten und Gravuren auf dem Schnittmaterial geeignet. Die eigentlichen Hauptanwendungen bestehen aber darin, auch kleine Materialreste noch für die Herstellung gelaserter Teile verwenden zu können. Das schont Umwelt und Geldbeutel.

Zum anderen kann man bereits bearbeitete Teile nun noch per Laser ergänzen: Sicher ist es jedem schon einmal passiert, dass man nach dem Lasern feststellte: Es fehlt noch ein Schnitt. Den nachträglich passgenau einzusetzen, ist ohne Kamera kaum möglich. Jetzt geht es problemlos. So vermeidet man also Verschnitt. *—hgb*

Bild 18: Ist der Text länger und der Zahnstocher liegt schräg im Cutter, muss man die exakte Drehung der Schrift etwas üben.

secIT 2024:

Cyberangriffe ins Leere laufen lassen

Praxiswissen aus erster Hand, mit dem Unternehmen Cyberattacken vorbeugen oder sie stoppen können, bietet die secIT. Die Kongressmesse findet vom 5. bis 7. März 2024 statt.

Wie können sich Unternehmen und Organisationen vor Cyberattacken schützen oder die Folgen solcher Angriffe minimieren? Antworten auf diese Fragen geben die rund 50 praxisorientierten Workshops, Vorträge und Deep-Dive-Sessions auf der secIT 2024.

Cyberangriffe sind für Unternehmen und öffentliche Einrichtungen ein ernsthaftes Problem. Das belegt unter anderem die Studie „Wirtschaftsschutz 2023" des Digitalverbands Bitkom. Ihr zufolge wurden im vergangenen Jahr 70 Prozent der deutschen Firmen Opfer von Datendiebstählen. In mehr als 60 Prozent kam es zur digitalen Sabotage von Informations- und Produktionssystemen. Die Folge: Schäden von rund 150 Milliarden Euro.

Schwerpunkt auf Praxis

CISOs, IT-Sicherheitsexperten und Administratoren müssen daher in der Lage sein, Cyberattacken einen Riegel vorzuschieben oder zumindest den Schaden durch Angriffe zu begrenzen. Wie sie das bewerkstelligen können, erfahren sie auf der IT-Security-Kongressmesse secIT 2024 by heise im Hannover Congress Centrum (HCC).
Den Großteil des Programms haben die Redaktionen von c't, iX und heise Security zusammengestellt. Die Besucher erwartet daher kein „Marketing-Blabla". Sie bekommen vielmehr wertvolle Informationen und Praxiswissen aus erster Hand.

„Teamwork ist in der Incident Response unverzichtbar"

Incident Response ermöglicht es Unternehmen, die Auswirkungen von Cyber-Angriffen zu minimieren. Doch damit dieser Ansatz den vollen Nutzen bringt, sind auch „menschliche Faktoren" wichtig, etwa Teamwork, Empathie und Rücksichtnahme, so Lisa Lobmeyer, Team-Managerin bei der HiSolutions AG.

Frau Lobmeyer, warum ist Incident Response so wichtig?
Incident Response kann die Folgen eines IT-Sicherheitsvorfalls abfedern. Dazu gehört ein gutes Krisenmanagement. Es hilft, die Lage zu beruhigen und entscheidungsfähig zu werden. Ein weiteres Element ist die forensische Aufklärung. Sie gibt Aufschluss darüber, was eigentlich passiert ist und ermöglicht es, die weitere Vorgehensweise im Krisenstab zu bestimmen.

Lisa Lobmeyer, Team-Managerin bei der HiSolutions AG

Ist Unternehmen und Organisationen bewusst, wie wichtig Incident Response ist?
Gerade dem Mittelstand fehlen häufig die Kapazitäten für Incident Response, aber auch das Bewusstsein für dessen Notwendigkeit. Im Idealfall können Unternehmen vorbereitende Maßnahmen treffen, um die Reaktions- und Wiederanlaufzeit zu verkürzen. Dazu gehören Ransomware-sichere Backups sowie Wiederanlaufpläne, Notfallkontaktlisten und Infrastrukturpläne, die in gedruckter Form vorliegen, aber auch eine Übersicht über Notfallmaßnahmen, die helfen, IT-Ausfälle zu überbrücken.

Ist es ratsam, in Eigenregie ein Krisenmanagement durchzuführen?
Organisationen können Maßnahmen im Krisenmanagement vorbereiten, um schneller Ruhe ins Team zu bringen und Entscheidungen zu ermöglichen. Weniger zu empfehlen ist, in der Chaosphase und der damit verbundenen Überforderung das Krisenmanagement ohne Vorbereitung selbst in die Hand zu nehmen. Ohne Vorkehrungen und das nötige Vorwissen ist es nur schwer möglich, die nötige Klarheit und Führung zu bieten, die ein Teamwork ermöglicht. Deshalb ist es im Regelfall anzuraten, auf spezialisierte Fachkräfte zurückzugreifen.

Ransomware? Nein danke!

Ein Schwerpunkt der secIT sind Angriffe mit Erpressersoftware – laut dem Bundesamt für Sicherheit in der Informationstechnik (BSI) nach wie vor die größte Bedrohung für Firmen. Stefan Strobel von cirosec stellt auf der secIT aktuelle Ransomware-Trends vor. Dieses Wissen ist sehr wichtig, um die Security-Strategien anzupassen.

Auf welche Schwachstellen Cyberkriminelle es besonders abgesehen haben, zeigt Dr. Jörg Schneider in einem Vortrag. Dabei stellt er die Top-Ten-Lücken aus Pentests des vergangenen Jahres vor und was man dagegen tun kann.

Cyberattacken haben verheerende Folgen, wie Volker Kozok vom Netzwerk für Cyber Intelligence am Beispiel des Landkreises Anhalt-Bitterfeld erläutert. Die Verwaltung des Kreises musste nach einem Angriff mit einem Verschlüsselungstrojaner monatelang in einem „Katastrophenmodus" arbeiten.

Active Directory und Entra ID schützen

Bei vielen Cyberattacken ist das Active Directory ein Einstiegspunkt für Angreifer. Denn fehlerhafte Konfigurationen und Sicherheitslücken im AD ermöglichen es Angreifern, sich Zugang zu Servern und Clients zu verschaffen. Im Rahmen eines Workshops und einer Live-Demo erläutert Robert Flosbach von Neodyme, wie Angreifer im Detail vorgehen und wie sich Unternehmen davor schützen können.

Ebenfalls um die Absicherung von lokalen AD-Instanzen sowie von Entra ID (ehemals Azure Active Directory) drehen sich zwei redaktionelle Workshops von Frank Ully und Tim Mittermeier, beide von Oneconsult. Ully konzentriert sich auf die On-Premises-Version von AD, insbesondere die Härtung solcher Umgebungen. Tim Mittermeier erläutert, welche Angriffspfade zwischen Entra ID, Azure-Services und On-Premises-Umgebungen vorhanden sind und wie sie sich unterbrechen lassen.

Es ist passiert – was nun?

Doch was tun, wenn es doch zu einem sicherheitsrelevanten Vorfall (Incident) kam? Mit diesem Thema beschäftigen sich mehrere Workshops, redaktionelle Vorträge und Keynotes. So führt Joshua Tigao von circosec eine Schulung zum Thema Incident Handling & Response durch. Die Basis bildet dabei der ISO-Standard 27035. Das Rahmenwerk unterstützt IT-Sicherheitsfachleute und Administratoren dabei, sicherheitsrelevante Vorfälle zu erkennen und Gegenmaßnahmen einzuleiten.

Damit sich der Schaden durch eine Cyberattacke in Grenzen hält, sind organisatorische Vorkehrungen und exakt definierte Prozesse erforderlich. In einem Workshop spielt Maik Würth von HiSolutions mit den Teilnehmern durch, wie diese Strukturen aussehen sollten.

Keine Entwarnung

Weitere Themen der secIT sind die Absicherung von Cloud-Umgebungen und der Schutz von IoT-Komponenten (Internet of Things). Im Workshop „Practical Ethical Hacking" von Jan-Tilo Kirchhoff (Compass Security) wiederum schlüpfen die Teilnehmenden in die Rolle von Hackern und lernen, wie sie ein komplettes Firmennetzwerk unter ihre Kontrolle bringen können, um Schwachstellen aufzudecken und Gegenmaßnahmen einzuleiten.

Das Fachwissen, das die Besucher von der secIT mitnehmen, ist wichtiger denn je. Claudia Plattner, die Präsidentin der obersten deutschen IT-Sicherheitsbehörde BSI, stuft die Lage im Bereich Cybersecurity in Deutschland nach wie vor als „besorgniserregend" ein.

Welche Faktoren entscheiden über den nachhaltigen Erfolg von Incident-Response-Einsätzen?

Das Wichtigste nach einem IT-Security-Vorfall ist, dass die Betroffenen als Team an dessen Bewältigung arbeiten können – mit Geduld und Struktur. Ein größerer Vorfall erledigt sich selten innerhalb weniger Stunden oder Tage. Zusätzlich hilft eine Portion Pioniergeist, und zwar sowohl bei der Führung als auch den Mitarbeitern. Denn ein Incident tritt schließlich nicht jeden Tag auf.

Wie kann die Zusammenarbeit von Incident-Response-Teams verbessert werden?

Im Ernstfall braucht das Team einen „Leuchtturm" in Form einer Führungskraft. Speziell die Führungsebene, insbesondere die Krisenstabsleitung, benötigt Mut zur Veränderung, um die Cyber-Attacke zu bewältigen. Zu beachten ist zudem, dass ein solcher Vorfall viel Kraft kostet. Daher sollte das gesamte Team ein hohes Maß an Rücksicht und Empathie aufbringen, speziell das Management. Werden diese Faktoren berücksichtigt, kann sich die erfolgreiche Bewältigung eines Cyber-Angriffs in einem Unternehmen sogar zu einer „Teambuilding-Maßnahme" entwickeln, wie etliche unserer Kunden bestätigt haben, allerdings eine, auf die die meisten gerne verzichtet hätten.

Peltier-Nebelkammer

Radioaktivität ist allgegenwärtig, mit bloßem Augen jedoch nicht zu erkennen. Diese selbstgebaute Diffusions-Nebelkammer macht ionisierende Strahlung aber sichtbar. Dafür verwendet sie ausgediente Computertechnik und ein kühlendes Peltier-Element.

von Matthias Rosezky

Unsichtbare Strahlung kann man mit einer vergleichsweise einfachen Technik sichtbar machen: der sogenannten Diffusions-Nebelkammer. In ihr wird ionisierende Strahlung mit freiem Auge als Spuren in einem feinen Nebel aus Alkoholtröpfen sichtbar. Das setzt eine ausreichend niedrige Temperatur von etwa –30 °C voraus. Am einfachsten erreicht man diese mit Trockeneis. Allerdings kann man die Nebelkammer damit nicht lange betreiben. Schließlich ist das Trockeneis nach einiger Zeit aufgebraucht. Eine Lösung hierfür bieten Peltier-Elemente, mit denen man (Strom vorausgesetzt) beliebig lange kühlen kann. In diesem Artikel zeige ich euch, wie ihr damit relativ einfach eine eigene Nebelkammer bauen könnt.

Grundlagen

Um den Nebel für eine Diffusions-Nebelkammer zu erzeugen, benötigen wir ein möglichst hohes Temperaturgefälle und einen Stoff wie Isopropanol oder Ethanol. Dieser verdampft bereits bei Raumtemperatur und kondensiert bei ca. –30 °C schon wieder ausreichend, sodass sich eine dünne Nebelschicht bildet. Dieses übersättigte Luft-Alkohol-Gemisch wartet praktisch nur darauf, endlich an einer Stelle zu kondensieren.

Tritt jetzt etwa ein hochenergetisches Alpha- oder Beta-Teilchen in diese Schicht ein, so ionisiert es einzelne Atome im Gemisch entlang seiner Bahn. Diese Ionen sind der Ausgangspunkt oder auch die Kondensationskeime für die Bildung von Alkoholtröpfchen, die man als Spur im Nebel wahrnehmen kann. Je nachdem, welche Strahlungsart auftritt, ergeben sich unterschiedliche Bahnen: Schwere Alpha-Teilchen bilden kurze und sehr kräftige Spuren, leichtere Beta-Teilchen längere und zudem deutlich dünnere. Legt man außerdem noch einen starken Magneten in die Kammer, krümmen sich die Bahnen der unterschiedlichen Teilchen gemäß ihrer Ladung und Masse. Je höher die Strahlung ist, desto mehr könnt ihr sehen (Bild 1). Damit ihr euch das besser vorstellen könnt, findet ihr in der Kurzinfo einen Link zu einem Video meiner Nebelkammer.

Besonderheiten des Entwurfs

Als Kühlfläche dient ein einfaches Kupferblech, unter dem ein Peltier-Element befestigt ist (Bild 2). Man könnte auch Aluminium verwenden, doch Kupfer weist eine deutlich bessere Wärmeleitung auf und sorgt dadurch für gleichmäßigere Temperaturen auf der gesamten Platte. Um die heiße Seite des Peltier-Elements zu kühlen, nutze ich einen handelsüblichen CPU-Kühler. Dieser ist mit Nylonschrauben am Blech befestigt, damit diese keine Hitze zum Blech zurückführen.

Kurzinfo

» **Diffusions-Nebelkammer selbst bauen**
» **Ionisierende Strahlung sichtbar machen**
» **Experimente mit Peltier-Elementen**

Checkliste

Zeitaufwand:
ca. 4 bis 5 Stunden

Kosten:
150 bis 200 Euro

Material

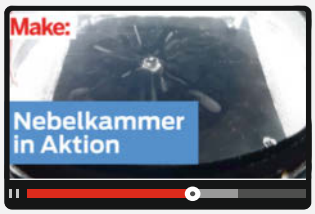
» **Thermalright Peerless Assassin 120 CPU-Luftkühler** oder vergleichbar
» **PC-Netzteil mit mindestens 250 Watt,** teilmodular
» **Peltier-Element TEC2-25408**
» **Kupferblech,** 100 x 100 x 2 mm
» **Acrylglas,** min. 160 x 160 mm
» **LED-Streifen 50 cm,** 5V oder 12V
» **Glasglocke,** ca. 14 cm Durchmesser und 15 bis 20 cm Höhe
» **Wärmeleitpaste**
» **2 Stücke dünner Filz,** ca. 100 x 100 mm
» **Schwarzes Elektro-Isolierband**
» **Selbstklebendes PE-Isolierband,** 2 mm Stärke
» **4 M2,5-Nylon-Abstandhalter,** 8 mm, Female-Female
» **4 M2,5-Nylon-Schrauben,** 10 mm
» **4 M2,5-Nylon-Schrauben,** 6 mm
» **4 Unterlegscheiben für M2,5-Schrauben**
» **4 kleine Magnete**
» **4 68-Ohm Widerstände,** 0,5 Watt
» **2 dünne Kabel,** ca. 30 cm
» **24-Pin-ATX-Mainboard-Verlängerungskabel**
» **8-Pin-ATX-CPU-Verlängerungskabel**

Werkzeug

» **Lötkolben**
» **Bohrer** oder **Dremel**
» **Seitenschneider**
» **Schere**
» **Schraubendreher**
» **Tacker**
» **Permanentmarker**

Mehr zum Thema

» Heinz Behling, Eigenbau-Geigerzähler mit Alarm, Make 3/22, S. 8
» Marcel Dierig und Tobias Lohf, Daten aus der Stratosphäre erheben, Make 5/19, S. 64
» Ulrich Schmerold, Peltier-Leselampe, Make 6/15, S. 12
» Video: Peltier-Nebelkammer in Aktion



Alles zum Artikel im Web unter
make-magazin.de/x1dh

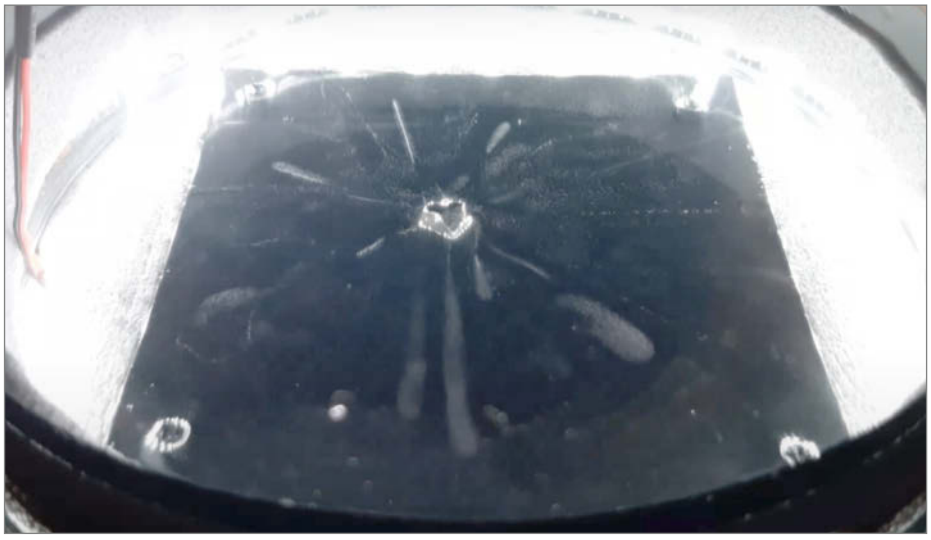


Bild 1: Dank des Nebels wird das Strahlen-Feuerwerk radioaktiver Stoffe sichtbar.

Für eine ausreichend große Kühlwirkung verwende ich ein TEC2-Peltier-Element. Dieses hat intern bereits zwei übereinandergelegte Halbleiterschichten und wirkt dadurch genauso wie zwei einzelne Peltier-Elemente. Dass man aber nur eines benötigt, vereinfacht den Aufbau und erspart einiges an Arbeit und Tüftelei.

Dadurch, dass die wichtigsten Teile der Nebelkammer verschraubt sind, lässt sich alles unkompliziert zerlegen und wieder zusammenbauen, ohne dass man etwa Kleber lösen müsste. Gleichzeitig kann man mit Schrauben den Anpressdruck von Kühler, Peltier-Element und Kupferplatte deutlich leichter einstellen. Durch die bessere mechanische Integrität ist der gesamte Aufbau später auch sehr stabil und braucht keine weiteren Stützen.

Der einzige Verbrauchsstoff in diesem Aufbau ist das Isopropanol (kurz IPA), das sich in einem Stück Filz an der Decke einer Glasglocke befindet. In diesen Filz habe ich eine Heizung aus vier Widerständen verbaut, sodass das Temperaturgefälle nochmals ein wenig größer wird und mehr Nebel entsteht. Damit man später auch gut sieht, was in der Kammer passiert, leuchtet ein LED-Streifen die Bodenschicht und den Nebel aus.

Mit dem Kühler beginnen

Da ihr jetzt wisst, wie die Peltier-Nebelkammer funktioniert und aus welchen Bestandteilen sie besteht, ist es Zeit für den Nachbau. Diesen beginnt man am besten mit dem CPU-Kühler. An diesem bringt man zunächst die Lüfter an und schraubt anschließend die beiden Montageklemmen fest, mit denen man den Kühler normalerweise auf einem Mainboard befestigt. Ich habe zusätzlich ein paar Schaumstoffstücke zwischen Kühlplatte und Kühlrippen gesteckt, um den Luftzug in diese Richtung zu verringern.

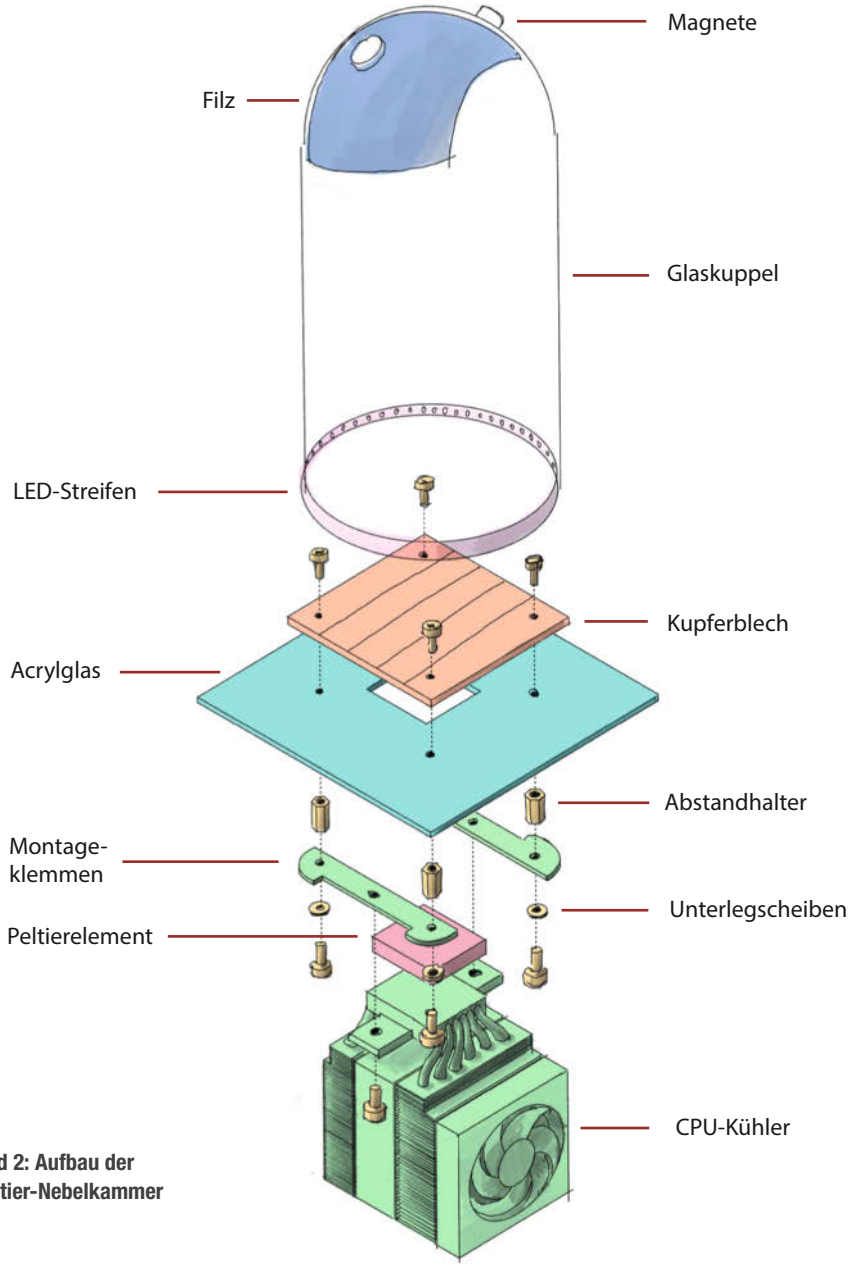


Bild 2: Aufbau der Peltier-Nebelkammer

Im nächsten Schritt montiert man das Peltier-Element am CPU-Kühler (Bild 3). Dazu trägt man erst eine gleichmäßige Schicht Wärmeleitpaste auf die Kontaktfläche des Kühlers auf. Danach presst man das Peltier-Element schön mittig darauf, sodass dessen kühlende Seite nach oben (vom CPU-Kühler weg) zeigt. Diese wird später am Kupferblech anliegen. Bei meinem TEC2-25408-Element ist die kühlende Seite jene mit der Modellnummer-Beschriftung. Im Zweifelsfall kann man das Element auch kurz an 12 Volt anschließen (siehe Abschnitt „Richtig verkabeln"), um sicherzustellen, dass man auch wirklich die richtige Seite erwischt hat.

Kupferblech vorbereiten

Damit man später die Teilchen im Nebel besser erkennen kann, empfiehlt es sich, eine Seite des Kupferblechs mit schwarzem Isolierband (oder einer schwarzen Kunststofffolie) abzukleben. Dabei lässt man das Material ein paar Millimeter über den Rand hinausragen, klappt es anschließend um und fixiert es mit Kapton-Tape auf der Unterseite des Blechs (Bild 4). Dadurch löst sich das Isolierband nicht so schnell.

Als Nächstes muss man vier Löcher in das Kupferblech bohren, damit man es später mit einer Acrylplatte und dem CPU-Kühler samt Peltier-Element verbinden kann. Als Schablone für die Bohrlöcher habe ich die Montageklammern des Kühlers verwendet. Dieser Schritt ist etwas fummelig, denn sobald man den Kühler mittig auf der Unterseite des Kupferblechs positioniert hat, schweben die Klammern ein paar Millimeter in der Luft (Bild 5). Man muss die Löcher also gedanklich auf das Kupferblech projizieren und mit einem Permanentmarker anzeichnen. Da die Schrauben aber ohnehin deutlich kleiner sind als die Löcher der Montageklammern, müssen die Markierungen nicht allzu genau sein. Außerdem zeichnet man noch den Umriss des Peltier-Elements auf das Kupferblech und markiert sich die Seite, auf der sich dessen Kabel befinden. Danach nimmt man den Kühler vom Blech und schraubt die Klemmen wieder ab.

Nachdem alles markiert ist, fixiert man das Kupferblech auf einer Arbeitsplatte, sodass es nicht wegrutschen kann und stanzt mit einem Körner die Bohrlöcher vor. Danach bohrt man sie mit einem 3mm-Metallbohrer in das Blech und prüft, ob sich die M2,5-Nylon-Schrauben hindurchstecken lassen.

Nun kommt die erste Wärmedämmung auf das Kupferblech. Dazu klebt man eine Schicht PE-Isolierband so auf die Unterseite des Blechs, dass die ganze Fläche bedeckt ist, die Bohrlöcher und Platz für das Peltier-Element aber frei bleiben (Bild 6). Die Dämmung sollte zudem ein paar Millimeter Abstand zum Element halten, sodass man es später exakt auf das Blech pressen kann, ohne dass man die Isolierung dazwischenklemmt.

Acrylglas bearbeiten

Zwischen dem Kupferblech und den Montageklemmen sitzt später ein Acrylglas, das als Auflagefläche für die Glasglocke und die LEDs dient. Um es mit dem Kupferblech verschrauben zu können, benötigt man im Acrylglas zunächst ebenfalls vier Löcher sowie einen rechteckigen Ausschnitt, durch den das Peltier-Element passt.

Nachdem man das Acrylglas gebohrt und geschnitten hat, legt man es auf die gedämmte Seite des Kupferblechs und richtet zunächst die Löcher aus. Danach fixiert man beide Teile miteinander, indem man nacheinander einen der vier 8-mm-Abstandhalter auf das Acrylglas setzt und von hinten mit einer M2,5×10-Nylonschraube durch das Kupferblech verschraubt. Dabei sollte man die Schrauben aber nicht zu fest anziehen. Es reicht, wenn die Dämmung zwischen Acrylglas und Kupferblech ein wenig zusammengedrückt wird.

Die gesamte Fläche auf der Unterseite des Acrylglases deckt man danach mit mindestens einer Schicht PE-Isolierband ab (Bild 7). Auch hier lässt man wieder Platz für das Peltier-Element und die Abstandhalter frei. Mehr als zwei oder vielleicht drei Schichten darf man jedoch nicht anbringen – sonst überragt die Dämmung die Abstandhalter.

Wenn man die verschraubten Platten umdreht, sieht man, dass es zwischen der Kupferplatte und dem Acrylglas einen kleinen Versatz gibt. Um diesen auszugleichen und gleichzeitig eine weiche Oberfläche zur Abdichtung der Glasglocke zu bekommen, klebt man eine Schicht PE-Isolierband auf die Oberseite des Acryls rund um das Blech (Bild 8). Das sollte schon ausreichen und verhindert gleichzeitig eine Kondensation an den Rändern des Kupfers.

Kühler montieren

Als Nächstes montiert man den Kühler an die Kupferblech-Acrylglas-Konstruktion. Dazu schraubt man zunächst die Klemmen des CPU-Kühlers mit vier M2,5×6-Nylonschrauben und Unterlegscheiben an den Abstandhaltern fest – gerade fest genug, dass nichts mehr wackelt (Bild 9). Danach trägt man eine gute Schicht Wärmeleitpaste auf das Peltier-Element auf, streicht sie glatt und setzt den Kühler langsam auf die Unterseite der Kupferblech-Konstruktion. Jetzt muss man nur noch die Klemmen mit dem CPU-Kühler verschrauben, bis sich nichts mehr bewegt. Das ist wichtig, damit eine gleichmäßige Auflagefläche gegeben ist und sich die Wärmeleitpaste gut verteilen kann.

Licht und Schatten

Um die Nebelfläche zu beleuchten, habe ich einen 50 cm langen LED-Streifen von außen um die Glasglocke gelegt und beide Enden mit Kapton-Tape am Glas festgeklebt. So kann man ihn recht einfach entfernen, ohne zu stark am Glas zu scheuern. Damit man einen besseren Blick auf den Isopropanol-Nebel hat und nicht so stark von den LEDs geblendet wird, habe ich den Streifen zudem mit einem Sichtschutz versehen. Das lässt sich relativ einfach bewerkstelligen, indem man direkt oberhalb des LED-Streifens sowohl innen als auch außen auf der Glasglocke ein bis zwei schmale Schichten des PE-Isolierbands aufklebt. Dadurch scheinen die LEDs nur mehr auf das schwarz abgeklebte Kupferblech und es ist kein Streulicht mehr von oben sichtbar. Ich musste zusätzlich das äußere Band noch einmal ringsum mit Kapton-Tape umkleben, damit es auch verlässlich hält.

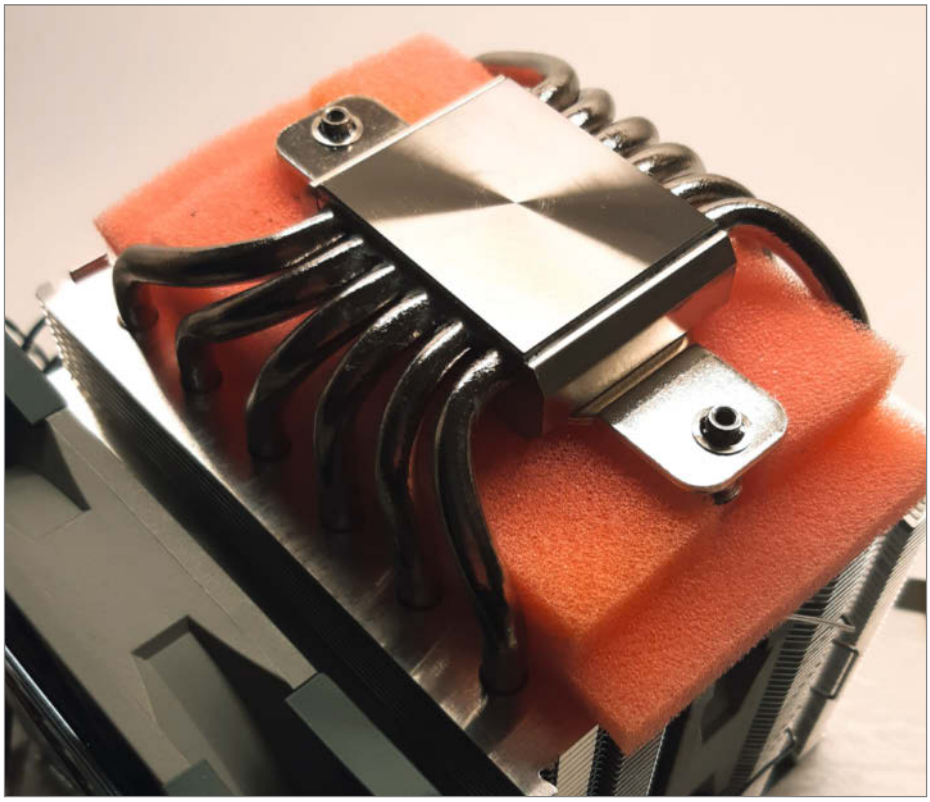


Bild 3: Auf die Kontaktfläche des CPU-Kühlers kommen Wärmeleitpaste und das Peltier-Element.

Nebel-Beschleuniger

Damit der Nebel schneller entsteht, lötet man vier 68-Ohm-Widerstände in Reihe und formt sie, so gut es geht, zu einem geöffneten Kreis. Für die Stromzufuhr befestigt man an beiden Enden jeweils ein 30 cm langes Kabel. Nun legt man den Filz um die Widerstände und tackert sie ein (Bild 10), sodass sich die Enden später auf keinen Fall berühren können. Ein entstehender Funke könnte sonst den IPA-Dampf entzünden.

Den getackerten Filz setzt man anschließend innen an der Deckelfläche der Glasglocke ein und fixiert ihn mittels Magneten durch das Glas. Zwei Haltepunkte können je nach Stärke der Magneten schon ausreichen.

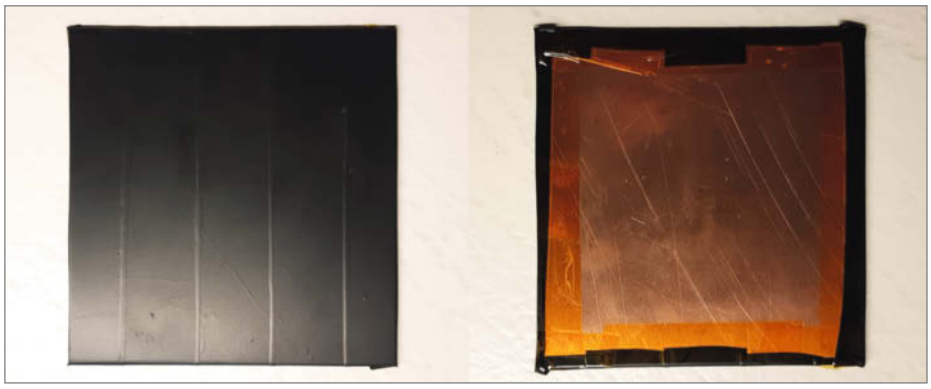


Bild 4: Die Verkleidung des Kupferblechs (links) geht bis auf die Rückseite (rechts).

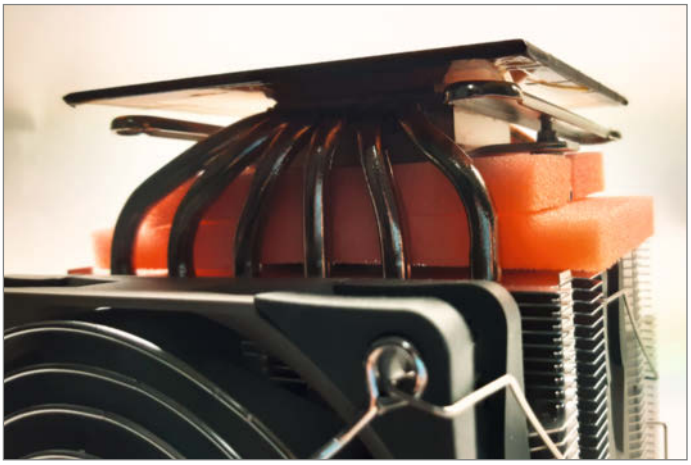


Bild 5: Etwas fummelig: Um die Bohrlöcher zu markieren, muss man mit dem Permanentmarker zwischen das Kupferblech und die Montageklemmen.

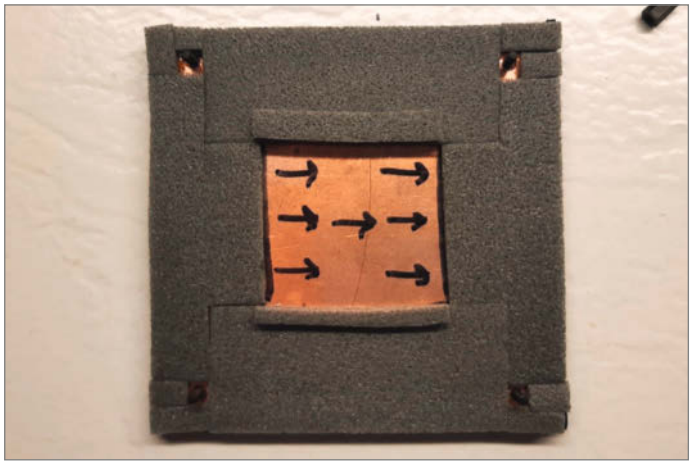


Bild 6: Die Pfeile auf der gedämmten Unterseite zeigen an, wo später die Kabel des Peltier-Elements liegen.

Wer den Filz stärker fixiert haben möchte, kann auch gerne mehr Magnete verwenden. Die beiden Kabel kann man optional mit Kabelbindern oder Schrumpfschläuchen bündeln und unter dem Glasrand hindurchführen.

Richtig verkabeln

Als Stromversorgung verwende ich ein altes, nicht mehr gebrauchtes PC-Netzteil von Corsair (CS450M) mit 450 Watt. Wie schon in der Kurzinfo angemerkt, reicht aber auch ein Netzteil mit weniger Leistung (ab 250 Watt). Das Peltier-Element verbraucht selbst etwa 96 Watt. Um die Kabel, die am Netzteil hängen, nicht zerschneiden zu müssen, nutze ich ein 8-Pin-ATX-CPU- und ein 24-Pin-ATX-Mainboard-Verlängerungskabel (Bild 11). Bei beiden habe ich die Stecker (male) abgeschnitten, sodass die Buchse (female) und die herausragenden Kabel übrig bleiben.

Wie man die einzelnen Komponenten der Nebelkammer mit den Verlängerungen verbindet, zeigt der schematische Schaltplan in Bild 12. Das Peltier-Element verbindet man mit der 8-Pin-ATX-Verlängerung. Den Strom für die restlichen Komponenten liefert die 24-Pin-ATX-Verlängerung, die abhängig von der Ader entweder 3,3 V, 5 V oder 12 V Spannung bereithält. Je nach verwendetem LED-Streifen benötigt man für diesen 5 oder 12 Volt. Die in Filz eingebettete Heizung sowie

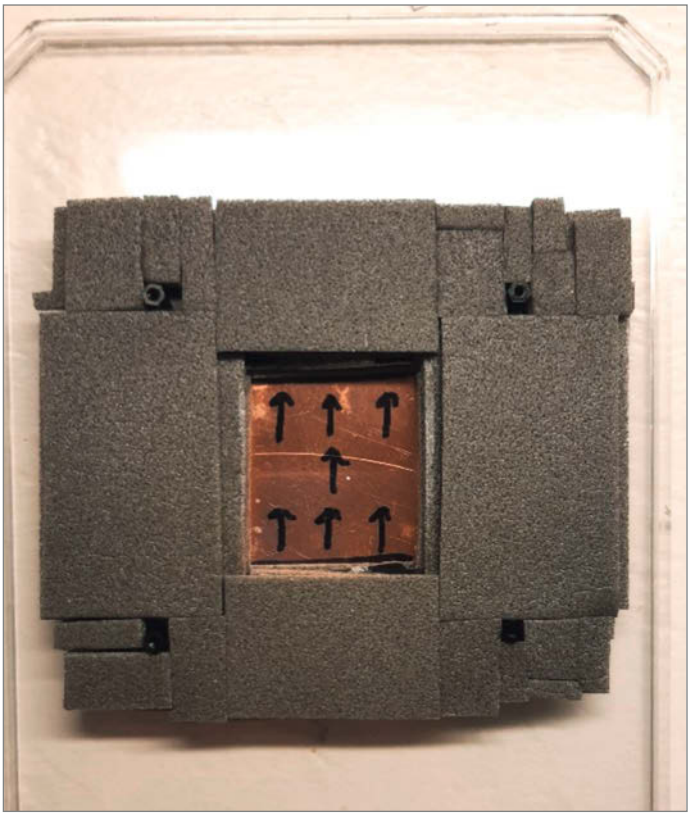


Bild 7: Die Dämmung auf dem Acrylglas sollte bündig mit den Abstandhaltern abschließen.

Bild 8: Mit PE-Isolierband verkleidet man zum Schluss die Oberseite des Acrylglases.

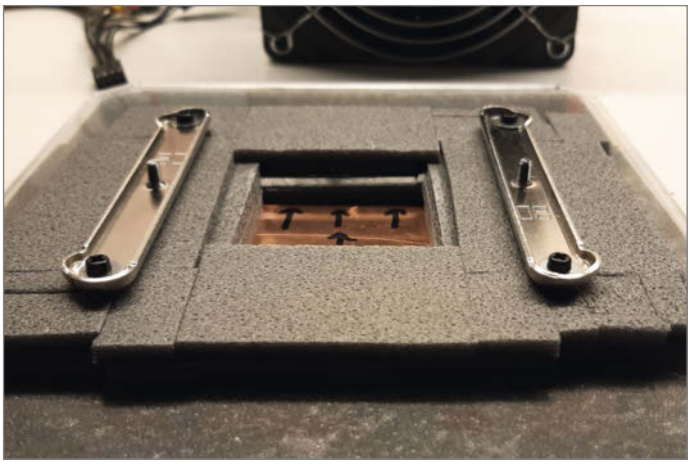


Bild 9: Man muss erst die Klemmen montieren und danach den Kühler an ihnen festschrauben.

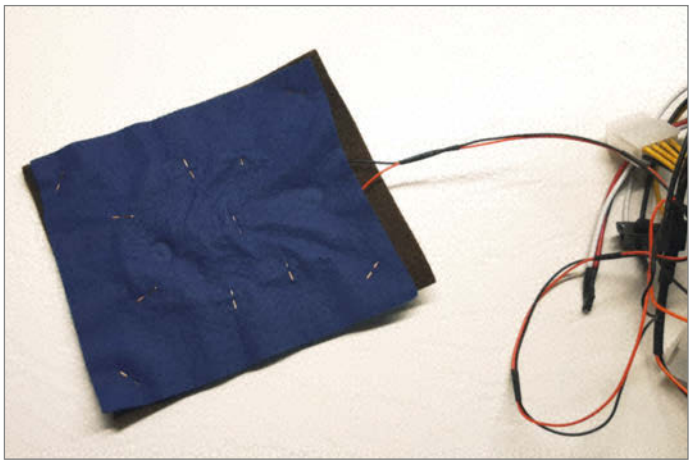


Bild 10: Die improvisierte Heizung aus Widerständen ist in den Filz eingetackert. Dadurch entsteht der Nebel schneller.

die Lüfter des CPU-Kühlers sind an 12 Volt angeschlossen.

Damit sich das Netzteil überhaupt ohne Mainboard einschaltet, muss man außerdem das einzige grüne Kabel an dem 24-Pin-Stecker auf Masse legen. Am besten schließt man es nicht kurz, sondern lötet einen 68-Ohm-Widerstand dazwischen. Danach kann man das Netzteil mit der Steckdose verbinden und einschalten. Herzlichen Glückwunsch, damit ist eure Nebelkammer fertig! Alles, was ihr über den Betrieb wissen müsst, folgt jetzt.

Betrieb und Wartung

Die Peltier-Nebelkammer ist sehr einfach zu benutzen. Falls die Kupferplatte nicht kalt genug wird, sollte man den Anpressdruck und die Wärmeleitpaste prüfen. Es kann aber auch an der Dämmung oder den Lüftern liegen. Ein Infrarot-Thermometer ist extrem hilfreich, um zu prüfen, ob die Temperatur an der Oberseite bei –30 °C liegt.

Um den Nebel zu erzeugen, tränkt man den Filz gut mit Isopropanol, sodass er überall gleichmäßig nass ist. Dann stellt man die Glasglocke auf das Acrylglas und schaltet die Nebelkammer ein. Jetzt ist ein wenig Geduld gefragt, denn es dauert etwas, bis die korrekte Temperatur erreicht wird. Nach ein paar Minuten bildet sich eine feine Nebelschicht direkt über der schwarzen Kupferplatte. Kurz darauf blitzen in der Schicht schon die ersten weißen Linien auf.

Sollte sich zu Beginn Eis auf der Oberseite der Kupferplatte bilden, kann man vor dem Einschalten auch eine dünne Schicht Isopropanol mit einem Tuch auf diese auftragen. Je nachdem, wie gut der Filz durchtränkt war, lässt die Stärke des Nebels nach längerer Zeit wieder nach. Dann muss man wieder ein wenig Alkohol nachgeben. Dabei sollte man in jedem Fall vermeiden, die Dämpfe einzuatmen, weil das zu gesundheitlichen Schäden führen kann.

Nach dem Betrieb sollte man das flüssige Isopropanol von der Kupferplatte wischen, damit es nicht in alle Ecken und Spalten kriecht. Wenn man plant, die Nebelkammer länger nicht zu benutzen, sollte man die Konstruktion samt Glasglocke außerdem ein paar Stunden im Freien offen liegen lassen, damit der restliche Alkohol verdampfen kann. Mehr ist nicht zu tun.

Eine mögliche Erweiterung

In Zukunft könnte man das Design noch um eine sogenannte Saugspannung ergänzen. Das ist eine Spannung von ungefähr 1000 Volt, die die freien Ionen der alten Spuren in der Nebelkammer absaugt und dadurch schneller wieder neue Spuren sichtbar werden lässt. Für eine solche Änderung müsste man aber den oberen Teil der Nebelkammer ein wenig umstrukturieren. Die Diffusions-Nebelkammer funktioniert aber auch ohne diese Modifikation schon ausgezeichnet. —*akf*

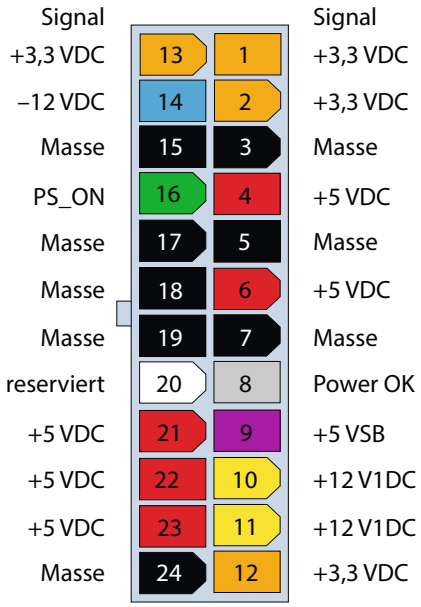


Bild 11: Anschlüsse an einem X-2.0-Stecker mit 24 Pins

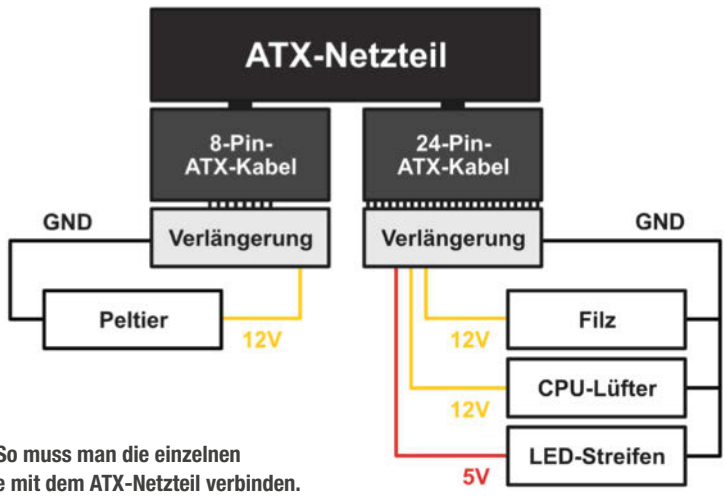
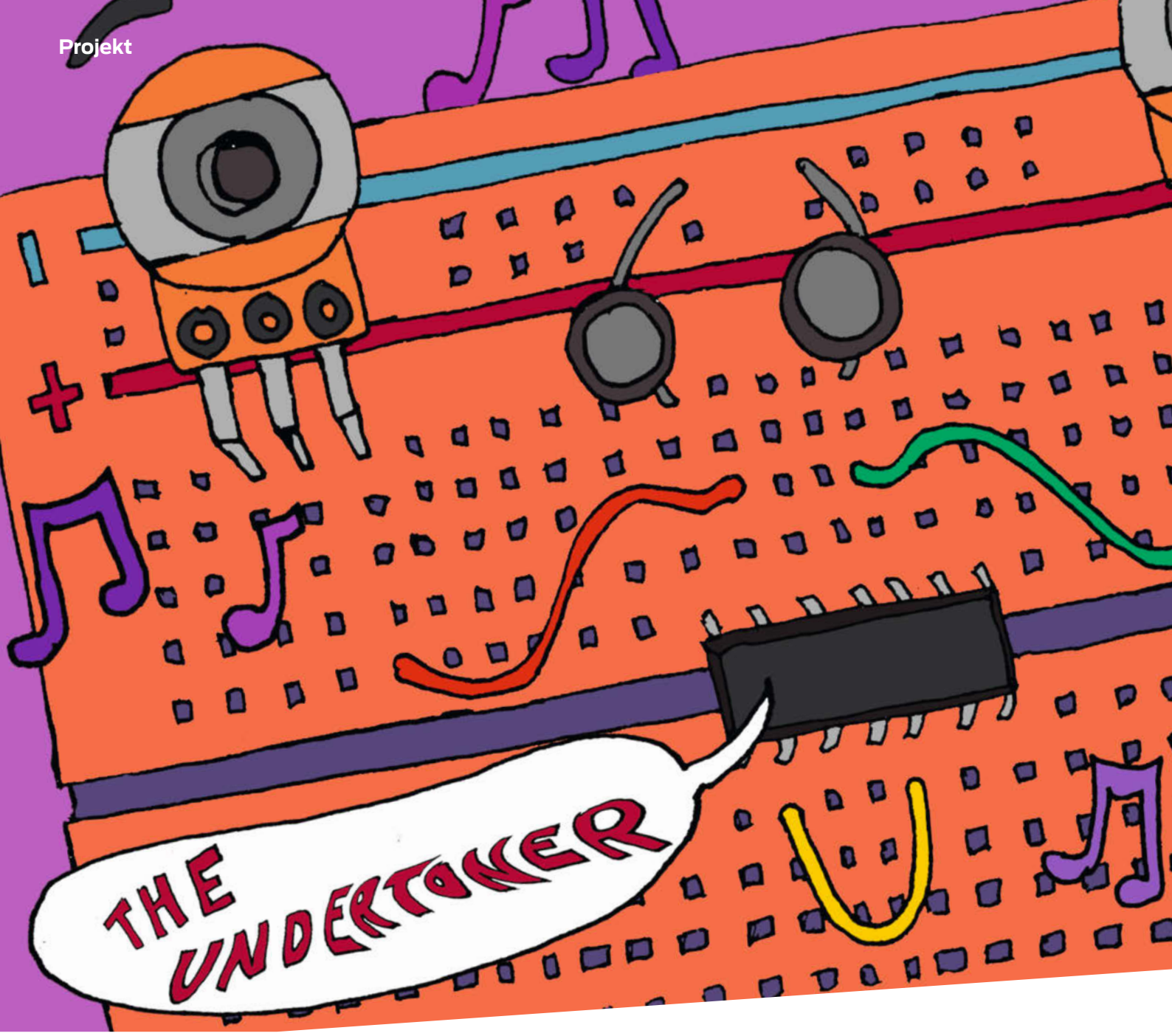


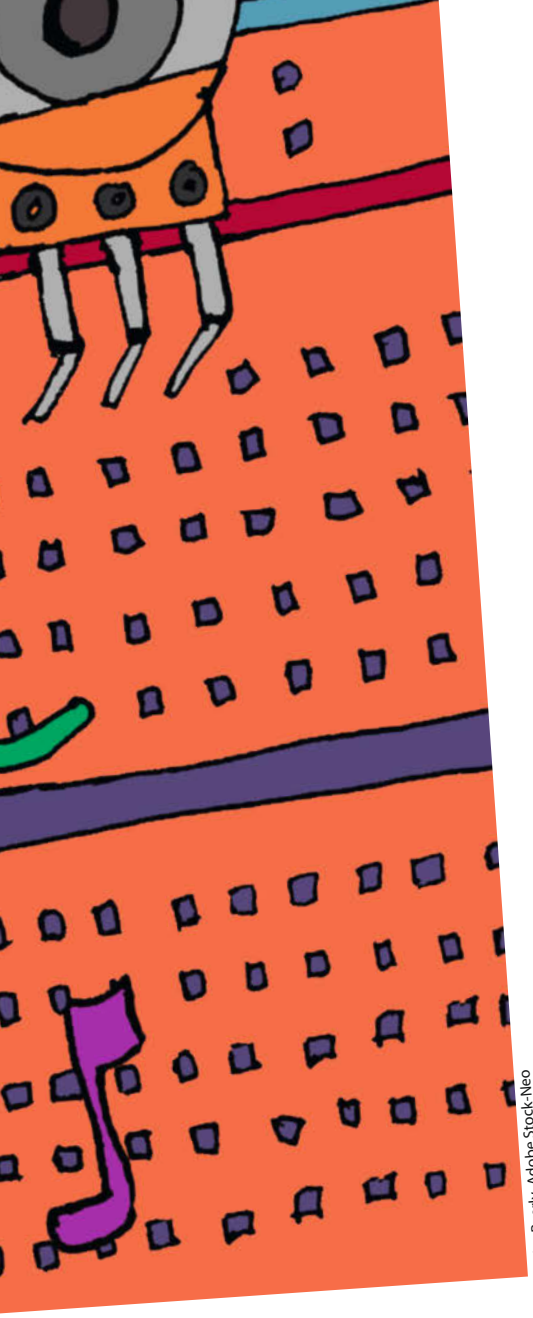
Bild 12: So muss man die einzelnen Elemente mit dem ATX-Netzteil verbinden.



Synthesizer mit perfekter Stimmlage

Der Undertoner ist ein einfacher Synthesizer, bei dessen Bau man einiges über Oszillatoren, Musiktheorie, Physik und den verwendeten CMOS-Chip erfährt.

von Sean Hallowell, Kirk Pearson



Maisy Byerly, Adobe Stock-Neo

Kurzinfo

» Erstaunlicher Synthesizer, der automatisch Tonleitern spielt
» Vielfältig nutzbarer Schaltkreis
» Basis für eigene Experimente

Checkliste

Zeitaufwand:
1 Stunde

Kosten:
ca. 10 Euro

Material

» **4 × CD4093** NAND-Gatter mit Schmitt-Trigger
» **Jumperkabel** isolierter Draht
» **3 × Potentiometer** 250kΩ
» **Kondensatoren** 0,1µF (100nF, Keramik o.ä.), 1µF (Elektrolyt), 10µF (Elektrolyt)
» **Breadboard**
» **Batterie** 9V-Block mit Anschlussclips
» **2 × Krokoklemmenkabel**
» **3 × Fotowiderstände** (LDR) für Experimente
» **LEDs** rot, für Experimente
» **Widerstände** 100Ω bis 1kΩ für Experimente

Werkzeug

» **Aktiver Lautsprecher** mit Line-In und passendem Klinkenkabel
» **Maker-Werkzeug** Seitenscheider, Abisolierzange, ggf. Multimeter

Mehr zum Thema

» Carsten Wartmann, Make-Breadboard++ Experimentierset zum Selbstbau, Make 2/23, S. 82
» Carsten Meyer, Mal rantasten: Wie Sie mit dem MIDI-Protokoll Klänge produzieren, Make 6/17, S. 106
» Kurt Diedrich, Analoger Synthesizer nach Mini-Moog-Vorbild, Make 5/17, S. 88
» Felix Pfeifer, Der Synthesizer-Punk, Make 4/16, S. 84
» Florian Fusco, Space-Sounds aus dem Siliziumchip, Make 1/18, S. 48

Alles zum Artikel im Web unter
make-magazin.de/xg28

Dogbotics Labs

Dogbotic Lab ist ein Audiolabor in der San Francisco Bay Area, eine Gemeinschaft, in der Menschen aus allen Gesellschaftsschichten gemeinsam experimentieren und lernen können. Die Autoren Sean Hallowell und Kirk Pearson sind Geschäftsführer von Dogbotics. Sie bieten DIY-Synthesizer-Work-shops für Anfänger und verrückte virtuelle Kurse zu Themen wie DIY-Drum-Maschinen, Audio- und Videosynthese, experimentelle Fotografie, tragbare elektronische Instrumente und vieles mehr.

dogbotic.com

Der Undertoner ist ein Synthesizer-Projekt, das sich perfekt für Anfänger eignet, die von Grund auf in die Welt der elektronischen Musik eintauchen wollen. Diese überraschend simple musikalische Schaltung besteht aus einem einzigen integrierten Schaltkreis, der sicherlich nicht in erster Linie für künstlerische Zwecke entwickelt wurde. Doch dank einer interessanten Eigenschaft der in ihm enthaltenen NAND-Gatter können wir aus diesem Chip eine Folge von Musiktönen zaubern. Irgendwie weiß die hirnlose Handvoll Drähte, die Sie gleich bauen werden, wie man eine tonale Melodie spielt!

Zuerst zeigen wir Ihnen, wie Sie diese einfache Schaltung bauen können, ohne dass Sie Erfahrung mit Elektronik haben müssen. Dann erklären wir Ihnen, wie diese wunderbare musikalisch-elektronische Besonderheit funktioniert, und führen Sie sogar durch eine Variante, bei der ein flackerndes Licht vor Ihren Augen Melodien komponiert! Gerade rechtzeitig für Ihren nächsten Abend bei Kerzenschein.

Der NAND-Chip

Darf ich vorstellen: der integrierte Schaltkreis CD4093B. Dieses kleine Kerlchen ist in der Funktion bemerkenswert einfach. Es enthält spezielle NAND-Gatter mit zwei Eingängen und einem Ausgang. Wenn beide Eingänge mit 0 Volt bzw. GND oder auch nur ein einzelner Eingang mit GND verbunden ist (der IC muss dafür mit der Versorgungsspannung verbunden sein), wird am Ausgang Strom ausgegeben. Wenn jedoch beide Eingänge mit der positiven Spannung verbunden sind, schaltet der Ausgang den Strom abrupt ab.

In Ihrem Computer befinden sich wahrscheinlich Millionen von NAND-Gattern – sie sind ein wesentlicher Bestandteil der digitalen Logik. Der CD4093D kommt als Dual-Inline-

Package (DIP)-Formfaktor mit 14 Beinchen daher, welche exakt in ein Breadboard passen. Weiterhin hat Ihr CD4093B nicht nur einen, sondern vier dieser NAND-Gatter in sich. Während man sich in einer herkömmlichen digitalen Schaltung, etwa einem Computer, darauf verlassen kann, dass die Schaltspannungen immer konstant sind, ist dies in der Welt außerhalb des Computers nicht immer der Fall. Daher besitzen die Eingänge der NAND-Gatter hier noch sogenannte Schmitt-Trigger (siehe Kasten), die dafür sorgen, dass die Gatter nicht in undefinierte Schaltzustände geraten. Zusätzlich verträgt der Chip Spannungen bis etwa 20 V, ist also sehr flexibel im Einsatz.

Schmitt-Trigger

Benannt nach seinem Erfinder Otto Schmitt ist der Schmitt-Trigger eine Schaltung, die dafür sorgt, dass Ein- und Ausschalten nicht gleichzeitig stattfinden können. Dafür sorgt eine Schalt-Hysterese. Wenn man etwa einen analogen Eingang überwachen möchte und bei über 2,5 V den Zustand AN erkennen möchte, die Spannung aber aufgrund äußerer Einflüsse leicht schwankt (Rauschen), dann kann es zu schnellen AN/AUS-Wechseln kommen, die in ungünstigsten Fällen zu Schwingungen führen. Der Schmitt-Trigger vergleicht nun mit zwei Spannungen, zum Beispiel mit unter 2,1 V (AUS) und mit über 3,0 V (EIN). Werden diese Schwellen nicht erreicht, beharrt der Ausgang in seinem vorherigen Zustand.

Beliebte Anwendungen des Schmitt-Triggers sind z. B. die Signalaufbereitung von digitalen Signalen, die über lange Leitungen übertragen wurden und deren Flanken nicht mehr steil genug für die übliche digitale Elektronik sind. Aber auch analoge Signale können so praktisch mit einem Bit abgetastet werden. Zusammen mit einem Tiefpass-Filterglied (Kondensator/Widerstand) können Tasten sehr effektiv entprellt werden. Und durch Rückkopplung von Ein- und Ausgang über einen Widerstand und einen Kondensator parallel zum Eingang lassen sich Schwingstufen (Oszillatoren) bauen, wie wir in diesem Artikel sehen werden.

Der Gate-Oszillator

In der ursprünglichen Schaltung sah das vierte (unbenutzte) NAND-Gatter ein wenig traurig aus, also haben wir es in einen Oszillator verwandelt, um unserer Schaltung einen rhythmischen Effekt zu verleihen! Wenn Sie die Töne ohne diesen Stakkato-Effekt (Gated) hören wollen, ersetzen Sie die Verbindung an Pin 11, die zum mittleren Pin des ersten Potentiometers führt, durch eine Verbindung zur Masseschiene. Mit einem einfachen Wechselschalter wird es dann noch flexibler.

Stromversorgung des Chips

Stecken Sie den 4093 in das Breadboard wie in Bild 1 zu sehen. Achten Sie darauf, dass die Halbkreis-Kerbe im IC-Gehäuse nach links zeigt. Der IC überbrückt so die Teilung in der Mitte des Breadboards. Wie in Bild 1 eingesteckt, sind die IC-Pins von links unten gegen den Uhrzeigersinn ringsherum nach links oben nummeriert. Bei einem 14-poligen IC wie dem 4093 ist also Pin 1 unten links, Pin 7 unten rechts, Pin 8 oben rechts und Pin 14 oben links.

Um Ihre NAND-Gatter korrekt mit Strom zu versorgen, müssen sie an eine Stromquelle angeschlossen werden. Eine 9-V-Batterie ist eine sehr einfache Möglichkeit. Sie können aber auch ein 9-V-Gleichstrom-Netzteil benutzen oder vielleicht leichter verfügbare 5 V. Allerdings sind die Frequenzen, die wir erzeugen, von der Spannung abhängig. Daher müssen bei 5 Volt die Werte der Bauteile angepasst werden, was wir nur fortgeschrittenen Makern empfehlen würden.

Diese Versorgungsspannung wird schließlich in die Strom- und Masseschienen des Breadboards geleitet, dies sind die oberen und unteren Reihen, die mit positiv (+, oft rot) und negativ (-, blau oder schwarz) gekennzeichnet sind. Um die Verdrahtung einfacher zu machen, sollten die Stromversorgungsreihen oben und unten am Breadboard miteinander verbunden werden (Bild 1). Während des gesamten Aufbaus sollte die Spannungsversorgung nicht verbunden sein!

Verbinden Sie dann die Pins 14, 12 und 8 mit einer positiven Spannungsschiene (+) und Pin 7 mit einer Masseschiene (-), wie in Bild 1.

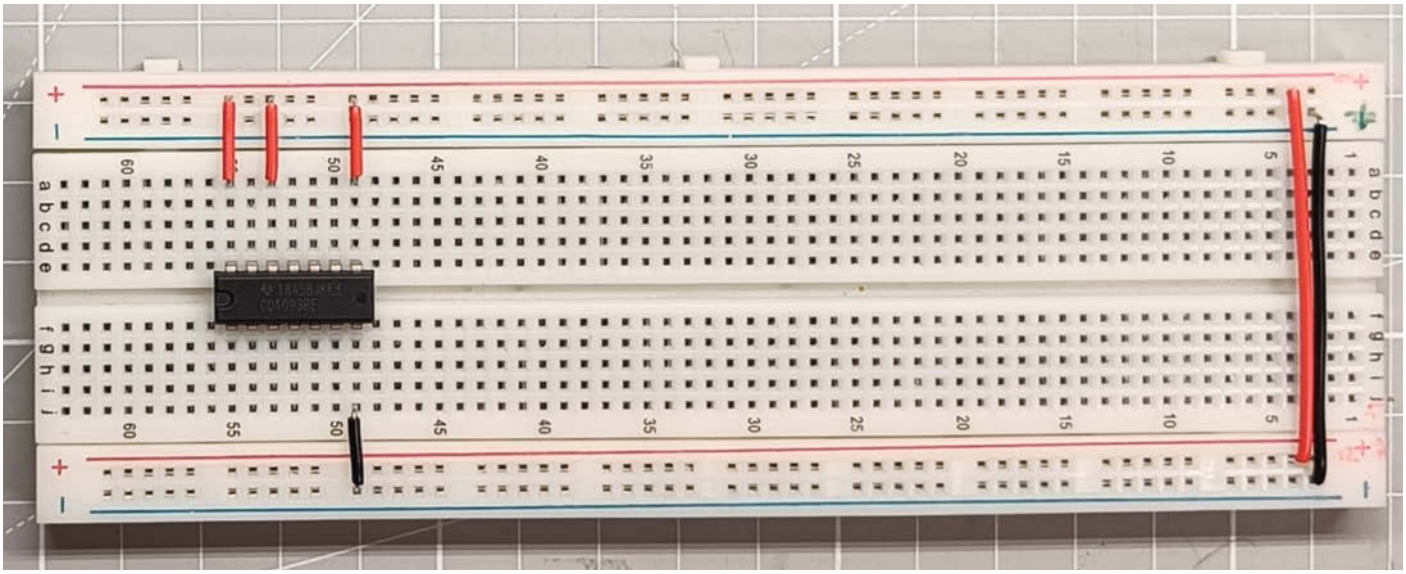


Bild 1: IC und Stromversorgung

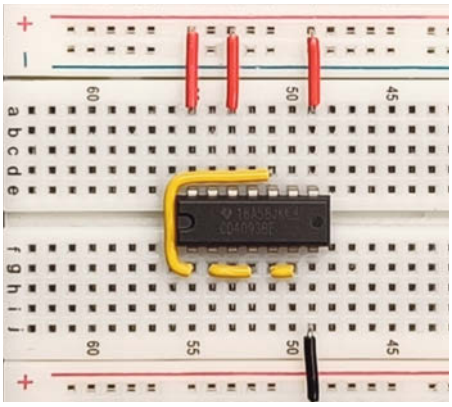


Bild 2: Die ersten Verbindungen

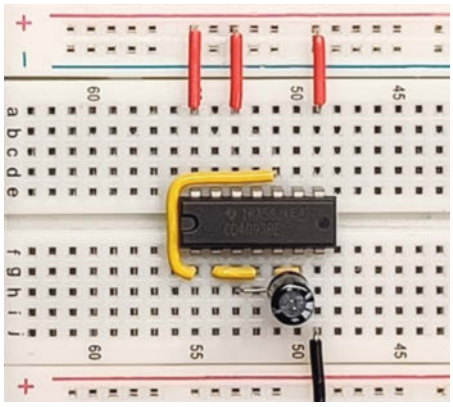


Bild 3: Kondensator für den Unterton-Generator

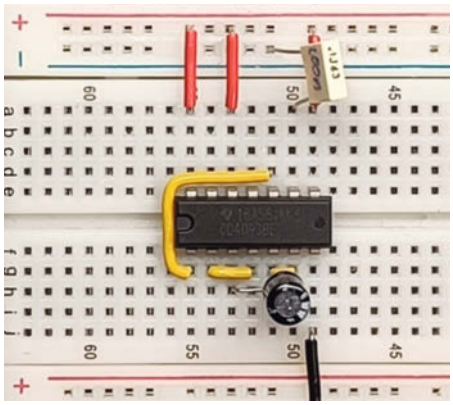


Bild 4: Kondensator für den Hauptoszillator

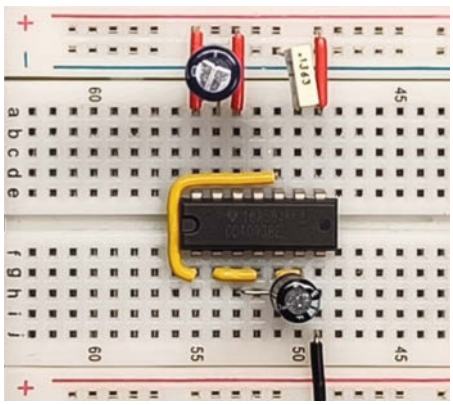


Bild 5: Kondensator für den Gating-Oszillator

Stellen Sie nun die internen Verbindungen des Chips her, wie in Bild 2 dargestellt: Pin 1 geht an Pin 10 des ICs, Pin 2 geht an Pin 4 und Pin 5 geht an Pin 6.

Die Kondensatoren

Nehmen Sie einen 1 µF-Kondensator und platzieren Sie den langen Anschluss (Anode oder Pluspol genannt) in der Reihe, die von Pin 3 geteilt wird, und den kurzen Schenkel (Kathode oder Minuspol) in der Reihe, die von Pin 6 geteilt wird (Bild 3). Unser nächster Kondensator hat eine Kapazität von 0,1 µF. Legen Sie den positiven Anschluss in die Reihe, die von Pin 9 geteilt wird, und den negativen Anschluss an Masse (Bild 4). Unser letzter Kondensator ist 10 µF. Er (Bild 5) befindet sich zwischen Pin 13 (langer Anschluss) und Masse (kurzer Anschluss).

Die Potentiometer

Das erste Potentiometer regelt den Stakkato-Effekt und wird zwischen den Pins 11 und 13 des ICs angeschlossen (Bild 6). Einer der Drähte wird an den mittleren Anschlusspin (das zum Schleifer führt) des Potentiometers angeschlossen. Der andere Draht wird an einen der seitlichen Pins angeschlossen. Welchen der beiden Pins Sie anschließen, bestimmt, ob das Potentiometer in die eine oder andere Richtung gedreht werden muss, um den Widerstand zu erhöhen und damit die Frequenz zu vermindern. Auch ein Vertauschen der beiden Pins führt zu einer Drehrichtungsänderung.

Das zweite Potentiometer wird zwischen den Stiften 9 und 10 angeschlossen (Bild 7) und regelt die Tonhöhe des primären Oszillators.

Das dritte Potentiometer (Bild 8) steuert die Untertöne. Einer der seitlichen Pins muss an Pin 6 oder 5 des ICs, der Mittelpin wird mit Pin 11 des ICs verbunden. Auf meinem Breadboard wurde die Verbindung zu Pin 11 über den Mittelpin des ersten Potis durchgeführt, um den Anschluss nicht auf die schon sehr volle Seite des ICs führen zu müssen.

Einstecken und lostönen

Jetzt kommt der Moment der Wahrheit! Nehmen Sie das Kabel, das an Ihren Verstärker oder aktiven Lautsprecher angeschlossen ist, und setzen Sie zwei Krokodilklemmen auf den Stecker: Eine an der Spitze und eine an der Hülse des Steckers (Bild 9). Der hier abgebildete Klinkenstecker hat eine Spitze, einen Ring und eine Hülse, wobei der Ring der mittlere Teil ist (Stereo-Stecker). Verbinden Sie mit den Krokodilklemmen (grün im Bild) die Spitze des

Steckers und über ein Kabel mit Pin 4 Ihres CD4093. Nehmen Sie das andere Klemmenkabel (schwarz) und verbinden Sie die Hülse des Steckers mit einem Massenpunkt auf dem Breadboard.

Die Potentiometer sollten nicht an einem der Anschlagpunkte stehen, sondern irgendwo um die Mitte herum. Drehen Sie bei angeschlossener Batterie die Lautstärke Ihres Verstärkers langsam auf. Sie sollten nun kräftige Rechteckwellen hören, die sich in einem gleichmäßigen Rhythmus ein- und ausschalten, der durch Potentiometer 1 (in den Bildern das linke) gesteuert werden kann. In den Links finden Sie ein kurzes Video dazu.

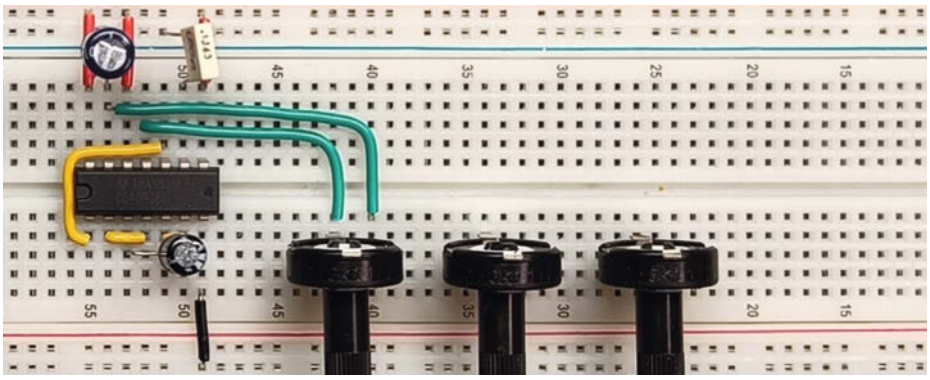


Bild 6: Kabel zum Poti für den Gating- oder Stakkato-Effekt

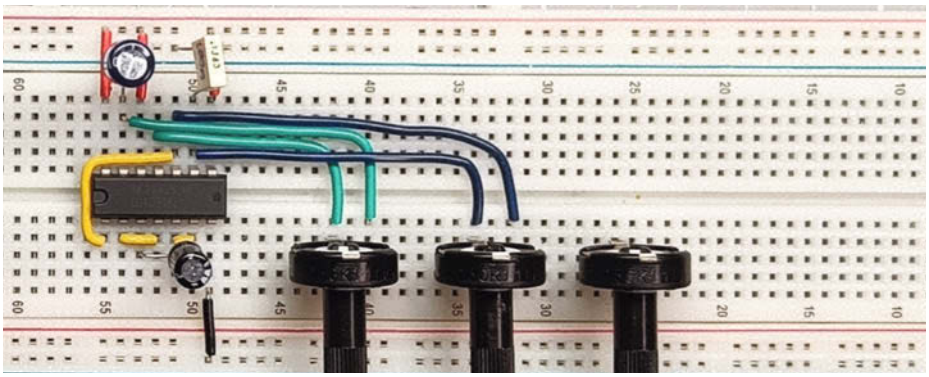


Bild 7: Poti 2 regelt die Frequenz des Hauptoszillators.

Drehen Sie jetzt Potentiometer 2 – je nachdem, in welche Richtung Sie es drehen, sollten Sie allmählich aufwärts oder abwärts gleitende Töne hören, die regelmäßig zurückgesetzt werden (d. h. wieder nach unten/oben springen), bevor sie wieder ansteigen/abfallen. Stellen Sie Poti 2 so ein, dass Sie eine relativ hohe Tonhöhe erhalten (eher ein Quietschen als ein Brüllen).

Nehmen Sie nun Potentiometer 3 und beginnen Sie, dieses langsam zu drehen. Sie sollten eine Reihe von Tönen hören, die sich in der Frequenz deutlich vom Nachbarton abheben. Einen nach dem anderen, fröhlich auf und ab, auf eine geheimnisvoll harmonische Weise miteinander verbunden. Unser Schaltkreis spielt nicht nur irgendwelche Tonhöhen, sondern Musiknoten auf einer Tonleiter.

Funktionsweise

Der Undertoner ist, in einem Wort: verblüffend. Ein paar Logikgatter liefern uns irgendwie eine Sammlung von Tonhöhen, die, egal wie sehr sie sich anstrengen, einfach nicht „verstimmt“ klingen können. Wie um alles in der Welt können ein paar subatomare Teilchen ein musikalisches Phänomen mit so tiefen kulturellen Wurzeln verstehen?

Lassen Sie uns diesen Kreislauf aufbrechen, um ihn zu verstehen. Zunächst wollen wir uns ansehen, wie man ein NAND-Gatter zum

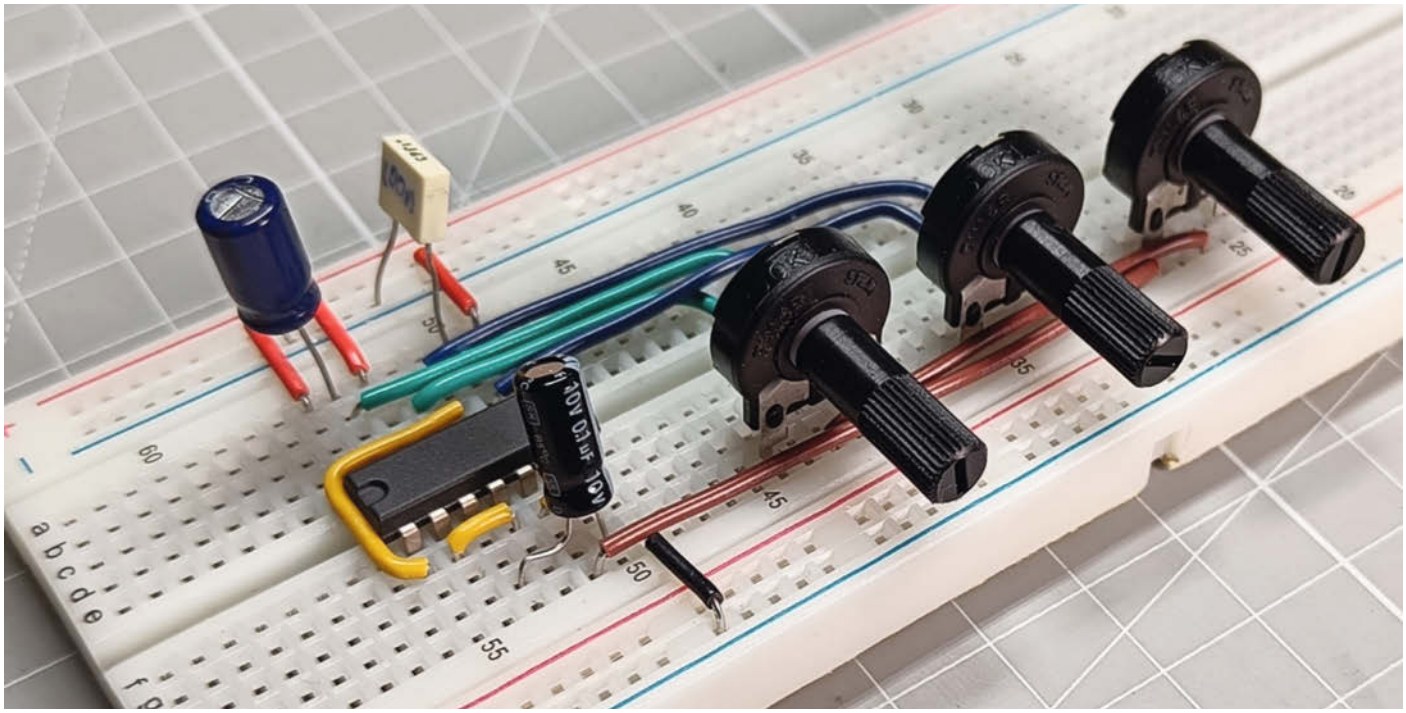


Bild 8: Das Poti 3 regelt die Pulsweite und ändert beim Drehen die Untertöne.

Schwingen bringt. Wenn wir über NAND-Gatter sprechen, hilft uns das, was Informatiker eine Wahrheitstabelle nennen. Sie zeigt uns, wie der Ausgang unseres NAND-Gatters aussehen wird, wenn unsere Eingänge EIN (durch eine 1 dargestellt) oder AUS (durch eine 0) sind.

Wie Sie aus dieser Tabelle ersehen können, haben wir, wenn wir einen der Eingänge unseres NAND-Gatters nehmen und ihn HIGH/EIN schalten (d. h. mit dem positiven Batteriepol verbinden), am anderen Eingang einen Inverter (eine logische 0 wird zu einer 1 und eine 1 zu einer 0). Stellen Sie sich nun vor, dass wir den invertierenden Eingang des NAND-Gatters mit dem Ausgang verbinden und so eine Rückkopplungsschleife über einen Widerstand schaffen. Indem wir einen Kondensator an den sogenannten „Rückkopplungseingang" des NAND-Gatters anschließen, fügen wir damit ein Gefäß für Elektronen hinzu. Genauer gesagt dient dieser Kondensator als Behälter, der die Zeit verzögert, die die Spannung am Ausgang braucht, um am Eingang registriert zu werden. Dies ist von Bedeutung, da der Eingang darauf wartet, dass die Spannung an seinem Pin einen bestimmten Schwellenwert erreicht, bevor sein Zustand als 1 oder 0 definiert wird. Der Kondensator bestimmt die Dauer dieses Prozesses, indem er sich füllt und leert. Ein größerer Kondensator bedeutet mehr Zeit zwischen den Zyklen, d. h. eine langsamere Frequenz und damit eine niedrigere Tonhöhe.

Indem wir ein Potentiometer, in der Wasseranalogie einen Hahn, zwischen Ausgang und Eingang schalten, können wir die Frequenz über den Widerstand steuern und die Zeit, die der Kondensator zum Auffüllen braucht, verlangsamen oder beschleunigen. Oder stellen Sie sich den Kondensator wie einen Eimer und den Widerstand wie einen Schlauch vor. Größerer Schlauch/kleinerer Eimer bedeutete schnelleres Füllen und umgekehrt. Mit anderen Worten: Das Potentiometer ermöglicht uns eine variable Steuerung der Tonhöhe. Wer sagt denn, Logik sei langweilig! Das NAND-Gatter, das wir angeschlossen haben, gibt einen pulsierenden Wechsel von 9 V und 0 V in einem Muster aus, das wir gerne als die Zähne eines Halloween-Kürbis sehen. Der Fachausdruck dafür ist Rechteckwelle. Werfen Sie einen Blick auf Bild 11.

Betrachten Sie die Rechteckwelle mit der Beschriftung „50 % Tastgrad". Die Ausgabe dieses NAND-Gatters besteht aus gleichmäßigen, abwechselnden EIN- und AUS-Momenten (der Kürbis hat regelmäßig angeordnete Zähne). Es steht jedoch nirgendwo geschrieben, dass unsere EIN- und AUS-Momente gleich sein müssen. Tatsächlich können wir die Impulsbreite (wie lang der EIN-Teil ist) mit dem 3. Potentiometer in dem Schaltkreis variieren, aber nicht so direkt, wie man vielleicht denken würde.

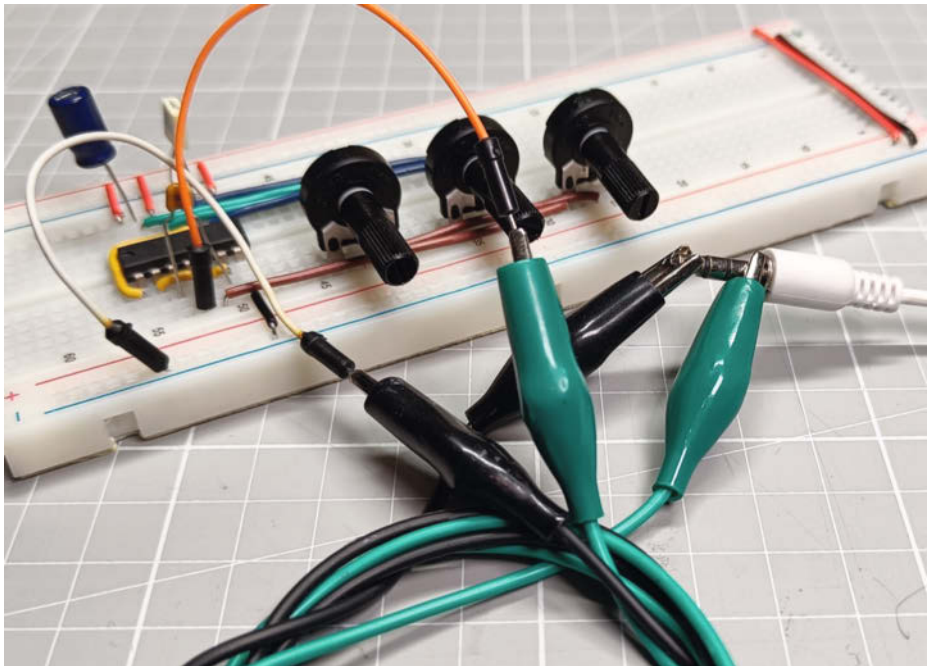


Bild 9: Anschluss des Verstärkers

Untertöne

Nehmen Sie sich einen Moment Zeit, um zu überlegen, was passieren könnte, wenn wir die Impulsbreite länger machen als die tatsächliche Periode der Schwingung. Jetzt wirkt die Pulsbreiten-Steuerung auch als Signalblocker. Wenn wir die Impulsbreite über die Periode der Oszillatorfrequenz hinaus erhöhen, wird jede zweite Schwingung blockiert. Es stellt sich heraus, dass durch das Blockieren jedes zweiten EIN-Moments der Welle die ursprüngliche Frequenz effektiv halbiert wird. In Musikerkreisen ist dies als „dieselbe Note eine Oktave tiefer spielen" bekannt. Wenn wir unsere Impulsbreite weiter verlängern, hören wir jedes Mal, wenn sie einen neuen ganzzahligen Schwellenwert überschreitet, eine neue Unterteilung unserer ursprünglichen Frequenz. Auf diese Weise kommen wir von der Frequenz x zu x/2, x/3, x/4, x/5 und so weiter.

Wir nennen diese Sammlung von Tonhöhen, die sich aus der Division einer ganzen Zahl durch eine Reihe von ganzen Zahlen ergeben, die Untertonreihe (subharmonisch). Das heißt, während in der Natur ein Grundton eine Reihe von Obertönen enthält, also sowohl ihn selbst als auch alle schnelleren (d. h. höher klingenden) Frequenzen, die den ganzzahligen Vielfachen entsprechen, geschieht in unserer Elektronik das Gegenteil. Anstelle von Oktave aufwärts, Quinte aufwärts, Quarte aufwärts, große Terz aufwärts und so weiter, erhalten wir auf unserem Steckbrett also Oktave abwärts, Quinte abwärts von dort, Quarte abwärts von dort, große Terz abwärts von dort

Atari-Punk-Candle

Wenn Sie experimentierfreudig sind, kann man Poti 3 gegen einen LDR, also einen lichtabhängigen Widerstand (englisch Photoresistor), auszutauschen. Auf diese Weise können Sie die Übergänge zwischen den Tönen mit wechselnder Lichtintensität steuern! Schnappen Sie sich eine Kerze, gehen Sie in einen dunklen Raum und versuchen Sie, den Punkt zu finden, an dem das Flackern der Kerze Ihren Undertoner wie einen Disco-Hit der späten 70er Jahre arpeggieren lässt. Generell funktionieren LDRs mit Dunkel-Widerständen von um 100 kOhm am besten. Sie können auch alle drei Potis durch LDRs ersetzen und dann den Undertoner mit drei Fingern berührungslos wie ein Theremin spielen.

Wenn Sie ein angehender Synthesizer-Nerd sind, kennen Sie den Namensgeber des Projekts vielleicht von einem anderen klassischen Einsteigerprojekt – dem sogenannten „Stepped Tone Generator" oder der „Atari Punk Console".

Like a vactrolling stone

Wie wäre es, wenn wir statt eines NAND-Gate-Oszillators für das Gate unseres Synthesizers einen Arpeggio-Oszillator für unsere Tonleiter verwenden würden? Diese Variante ist der Atari Punk Candle sehr ähnlich, aber mit einem kleinen Unterschied: Anstatt den Oszillator an den Pins 11, 12 und 13 zu verwenden, um den Synthesizer zu steuern, können Sie eine rote LED in Reihe mit einem Widerstand an- und ausschalten lassen. Platzieren Sie die LED so vor dem Fotowiderstand, dass die maximale Lichtmenge der LED auf den Fotowiderstand trifft. Wenn nun das Licht aus- und eingeschaltet wird, hören Sie, wie der Schaltkreis schnell durch die Noten der Untertonreihe schaltet. Diese lichtgesteuerte Modulationsquelle wird „Vactrol" genannt, und ob Sie es glauben oder nicht, es gibt sie überall. Angefangen bei alten Synthesizern für die Übertragung von analogen Spannungen an Schaltkreise bis zu MIDI-Kabeln zwischen zwei Musikinstrumenten: Hier werden analoge bzw. digitale Informationen übertragen. In moderner Inkarnation wird der lichtempfindliche Widerstand durch einen Fototransistor ersetzt.

Da die Informationen hier optisch – als Photonen und nicht Elektronen – übertragen werden, sind die kommunizierenden Systeme elektrisch völlig getrennt (galvanische Trennung). Auf diese Weise können Sie mit Ihrem MIDI-Kabel zwei beliebige Geräte miteinander verbinden, ohne befürchten zu müssen, dass sich Brumm- und Potenzialspannungen aufbauen, die den Klang stören und sogar Geräte zerstören können.

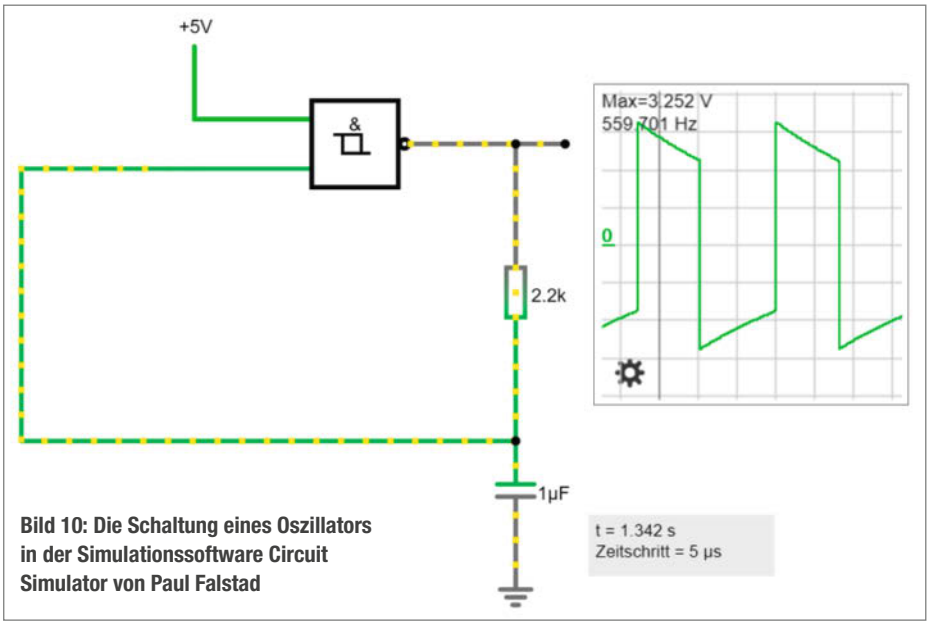


Bild 10: Die Schaltung eines Oszillators in der Simulationssoftware Circuit Simulator von Paul Falstad

falstad.com/circuit

NAND Wahrheitstabelle

Eingang A	Eingang B	Ausgang Y
0	0	1
0	1	1
1	0	1
1	1	0

und so weiter. Diese Untertoner-Schaltung spiegelt also die natürliche Obertonreihe wieder, indem sie unsere Eingangsfrequenzen durch ganze Zahlen dividiert! Das ist ein schöner Denkanstoß, dass die Prinzipien der Musik in der Welt der Elektronik genauso gelten wie in der Welt der Physik von schwingenden Körpern.

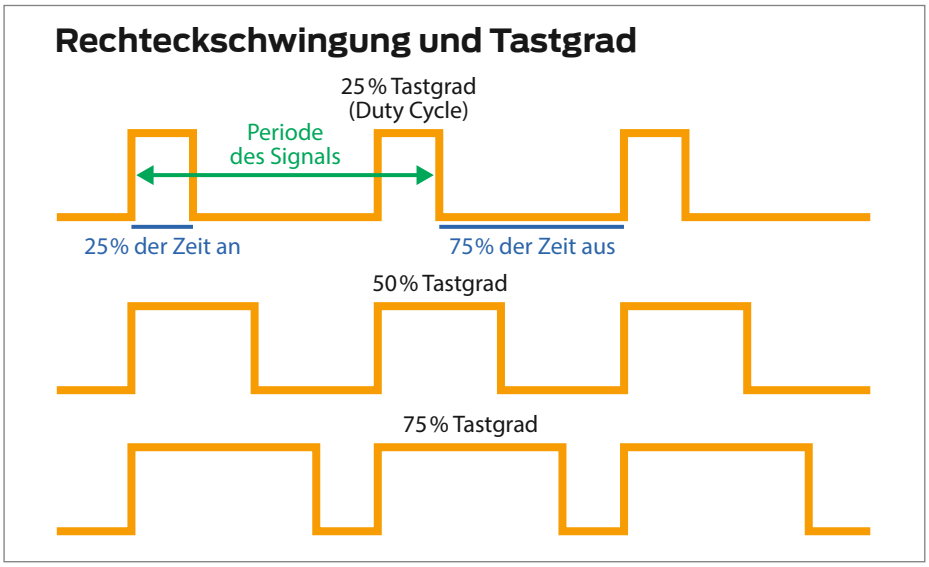


Bild 11: Eine Rechteckwelle mit Pulsweitenmodulation

Intonationale Angelegenheiten

Es mag schwerfallen, mit dieser Schaltung zusammen mit klassischen Instrumenten zu spielen. Seit Anfang des 18. Jahrhunderts hat die westeuropäische Kompositionstradition dieses mathematisch abgeleitete Stimmsystem, das als reine Stimmung bekannt ist, zugunsten der sogenannten gleichschwebenden oder gleichstufigen (temperierten) Stimmung weitgehend aufgegeben. Kurz gesagt: Dieses neue System macht es für verschiedene Instrumente einfacher, in der richtigen Stimmung zusammenzuspielen. Und obwohl der Undertoner im Gegensatz zu einem Klavier oder einer Gitarre ein synthetisches Instrument ist, sind die Tonleitern, die er spielt, dennoch mathematisch abgeleitet. Wenn man bedenkt, dass diese akustischen Instrumente im Gegensatz zum Undertoner auf unnatürliche Weise angepasst wurden, um eine andere, „künstlichere" Tonleiter zu erzeugen, fragt man sich, wer denn nun der wahre „Synthesizer" ist! *—caw*

Maker-Faire-Auftakt für 2024

Auch das Jahr 2024 hat wieder diverse Maker Faires zu bieten. Diese Termine dürfen in keinem Maker-Kalender fehlen.

von Daniel Schwabe

Alte Maker-Hasen wissen, dass auf den Maker Faires tolle Projekte, viele Produkte und Ideen von Makern, Firmen und Vereinen vorgestellt werden. Dazu gibt es ein abwechslungsreiches Rahmenprogramm für die ganze Familie. Und das Beste daran: für private Maker und Vereine sind die Stände vor Ort kostenfrei.

Heilbronn öffnet am 24. Februar 2024 seine Tore

Unter dem Motto „Learning by making" findet am 24. Februar 2024 von 10:00 Uhr bis 18:00 Uhr die Maker Faire Heilbronn im Maker Space experimenta statt.

Neben diversen Mitmach-Aktionen, bei denen z.B. mit Textilfolie eigene Motive auf Stofftaschen gedruckt werden können, mit dem Lötkolben Silberschmuck hergestellt wird, oder kleinen Holzwölfen gefährlich-rote LED-Augen verpasst werden, gibt es ab 12:00 Uhr auch wieder diverse Vorträge:

- PlayStation 2 and Reverse Engineering von Roland Lückl,
- Bau eines Minisatelliten für CanSat 2024 von Aurélie Wittmer,
- Drawbot – Ein Bausatz für Roboter-Workshops von Hans-Günther Nusseck,
- Mit Kopf, Herz und Hand – ganzheitliches Lernen in Makerspaces für Menschen mit und ohne Handicap von Simone Dinse de Salas.

Wer noch einen eigenen Vortrag für die Messe beisteuern möchte, kann sich noch über den „Call For Participation" als Referent anmelden.

Maker Faire Ruhr am 16. und 17. März 2024 in Dortmund

Vom 16.03 bis 17.03.2024 findet in der DASA – Arbeitswelt Ausstellung die Maker Faire Ruhr statt.

Mit dabei ist diesmal: die gesamte Make-Redaktion inklusive Shop. Am Make-Stand können aus der Zeitschrift bekannte Projekte hautnah erlebt und die Redaktion mit Fragen gelöchert werden. Wer selber mal Autor eines Make-Artikels werden möchte, kann gern auf das Team zukommen und sich persönlich eine Visitenkarte mit den Kontaktdaten schnappen.

Die 10. Maker Faire in Hannover am 17. und 18. August 2024

Dieses Jahr knallen am 17. und 18. August 2024 die Sektkorken in Hannover, denn es handelt sich um das zehnjährige Maker-Faire-Jubiläum in Präsenz – wie immer im Congress Center Hannover. Wer sich dieses Event nicht entgehen lassen möchte, kann sich im bereits eröffneten Ticketshop noch bis Ostern 20 % Rabatt auf die Eintrittskarten sichern (siehe Link). Günstiger werden die Tickets nicht mehr.

Wer als Aussteller an der Maker Faire Hannover teilnehmen will, hat bis zum 9. Juni 2024 Zeit, sich über den „Call for Makers" zu melden.

Selbstverständlich wird auch hier wieder die Make-Redaktion vor Ort sein und mit Makern, Lesern sowie kleinen und großen Fans zwei tolle Tage genießen.

Der Newsletter

Wer immer Up to date sein möchte, kann sich auf der offiziellen Maker Faire Homepage für den Newsletter-Verteiler anmelden und bekommt so immer die neuesten Informationen. —*das*

Fühlt sich an wie gestern. Auch dieses Jahr freut sich die Make-Redaktion auf die Maker Faires des Jahres.

Flaschengeist

In diesem Erlenmeyerkolben spukt ein sogenannter Pepper's Ghost, ein halbtransparentes Bild mit holografischer Anmutung. So eine gespenstische Erscheinung lässt sich in einem beliebigen Glas leicht selbst beschwören. Man braucht zusätzlich nur ein Stück Polycarbonat und ein Display – egal, ob Handy, Röhrenfernseher oder Oszilloskop.

von Joshua Ellingson

Kurzinfo

- Klassischen Spiegeltrick, mit einfachen Mitteln und großer Wirkung selber umsetzen
- Klappt mit unterschiedlichsten Glasgefäßen in allen Größen und Formen
- Inspiration: Zu Besuch im Makerspace und privaten Fernseher-Museum des Autors

Checkliste

Zeitaufwand:
1 bis 2 Stunden

Kosten:
etwa 15 Euro (ohne Bildschirm)

Material

- **Erlenmeyerkolben** 1000 ml oder ähnliches Glasgefäß
- **dünne Pappe** für die Schablone, etwa 12 × 15 cm, etwa aus einer Nudel- oder Müslipackung
- **Polycarbonatplatte** oder ähnlicher transparenter Kunststoff, 0,75 bis 1 mm dünn, etwa 12 × 15 cm
- **Bildschirm** auf dem Videos oder ähnliches abgespielt werden können: Smartphone, Tablet, Röhrenfernseher, Oszilloskop ...

Werkzeug

- **Computer und Drucker** für die Schablone
- **Schere**
- **Permanentmarker**
- **lange Pinzette** oder Spitzzange (optional)

Mehr zum Thema

- Carsten Wartmann, Vektorgrafik auf dem Oszilloskop, Make 5/17, S. 16
- Philip Steffan, Synthesizer im Eigenbau, c't Hacks 1/13, S. 86, im Volltext online
- Kurt Diedrich, Analoger Synthesizer nach Mini-Moog-Vorbild, Make 5/17, S. 88
- Florian Fusco, Space-Sounds aus dem Siliziumchip, Make 1/18, S. 48
- Daniel Springwald, Im Maker-Fotostudio, Make 4/22, S. 100

Alles zum Artikel im Web unter *make-magazin.de/x7ry*

Die klassische Bühnenillusion namens „Pepper's Ghost" lernte ich in einer Folge der Fernsehsendung „Mr. Wizard's World" auf dem Kindersender Nickelodeon kennen, irgendwann in den späten 1980er Jahren. Viele der dort gezeigten wissenschaftlichen Experimente für den Hausgebrauch waren überschaubar und leicht zu Hause nachzumachen, aber gelegentlich zeigte Mr. Wizard seinen jugendlichen Gästen im Studio etwas wirklich Spektakuläres. In diesem speziellen Beitrag ließ er die Erscheinung eines Skeletts auf einem Stuhl in einem Raum erscheinen, in dem sich definitiv kein Skelett befand, bevor er das Licht ausschaltete. Er erklärte, wie die Spiegeltäuschung funktionierte, mit einer großen, schrägen Glasscheibe zwischen dem leeren Stuhl und der geöffneten Tür. Es war erstaunlich, aber ich wusste, dass ich diesen Effekt zu Hause nicht nachmachen konnte. Außerdem: Woher sollte ich ein Skelett nehmen?

Dreißig Jahre später, im Jahr 2019, sah ich mir „The Imagineers" an, eine Dokumentation auf Disney+ über die Geschichte von Disneys experimenteller Imagineering-Abteilung und die Entwicklung der Themenparks des Konzerns. In den ersten Episoden wird Pepper's Ghost mehrfach erwähnt, vor allem in Bezug auf die erstaunliche Wirkung dieses Effekts in der Geistervilla-Attraktion (The Haunted Mansion) in den Parks. Das weckte eine alte Neugierde und jetzt als Erwachsener wurde mir klar, dass ich wahrscheinlich alles Material griffbereit hatte, das ich brauchte, um meine eigene Pepper's-Ghost-Illusion zu bauen.

Also schob ich ein Stück gerahmtes Plexiglas schräg in ein Aquarium, stellte ein altes LCD-Display oben auf das Becken, das nach unten gerichtet war, und spielte darauf ein Video eines Goldfisches vor schwarzem Hintergrund ab. Es funktionierte großartig! Ich war erstaunt, wie einfach es war, eine überzeugende Illusion eines schwimmenden Fisches zu erzeugen, also machte ich damit weiter. Schließlich probierte ich den Effekt mit einer Glasglocke auf meinem iPad aus, mit einem runden Stück transparenten Polycarbonats als Reflektor. Dies war der Beginn einer ganz neuen Reise, die mich tief in die Welt der Fotografie, der Kunststoffe, der Synthesizer und der Echtzeit-Grafik führte.

War ich nach über zwanzig Jahren als Werbegrafiker auf einmal ein angehender audio-

Keith Hanmond

Retro-Fernseher leuchten besser: Links ein massiver, aber theoretisch tragbarer RCA Sportable von 1958, rechts ein Philco Predicta Holiday aus derselben Ära.

visueller Installationskünstler? Ich hatte mich vorher schon mit Videokunst beschäftigt, aber meine früheren Experimente mit Schwarz-Weiß-Fernsehern schienen jetzt parallel zur Erforschung von Pepper's-Ghost-Illusionen auf einmal mehr Sinn zu ergeben.

Keith Hanmond

Wer mit alten Fernsehern arbeitet, braucht neben Adaptern auch jede Menge Kabel.

Alles fühlte sich wie eine große Entdeckung an, und das alles geschah kurz vor dem Ausbruch der Pandemie. Es stellte sich bald heraus, dass ich jetzt sehr viel Zeit für mich allein hatte, um zu experimentieren. Viele meiner Ergebnisse sieht man auf meiner Webseite ellingson.tv (siehe auch Link in der Kurzinfo). Wer sofort damit loslegen will, seinen eigenen Pepper's Ghost zu bauen, kann gleich zu Seite 63 weiterblättern, alle anderen lade ich vorher zu einem Besuch in meiner Werkstatt ein.

Fernseher-Faible

So ziemlich jede Art von Bildschirm kann gut mit einem Pepper's Ghost kombiniert werden, aber ich mag Kathodenstrahlröhren am liebsten. Das Leuchten eines Röhrenfernsehers hat etwas Faszinierendes, vor allem bei Schwarz-Weiß-Geräten aus den 1960er und 70er Jahren. Sie haben eine fast neonartige Intensität, die mit LCDs und anderen moderneren Techniken unmöglich zu reproduzieren ist. Ich habe eine wachsende Sammlung von alten Fernsehern in verschiedenen Größen, Formen und Altersklassen. Sie reicht von einem RCA Sportable von 1958, der massiv wie ein alter Buick aufgebaut ist, bis zu einem winzigen Symphonic Minni Portable von 1969. Ich habe auch Röhrenbildschirme, die keine Fernseher sind, etwa einen Monitor, der ursprünglich mit einer Schwarz-Weiß-Videokamera für die Mikroskopie verwendet wurde. Vor allem aber sammle ich Transistorfernseher aus den 1970er Jahren von Marken wie Zenith und General Electric. Ihr Bild ist auch nach fast 50 Jahren noch sehr gut und manchmal sind sie auch schick. Ich habe zum Beispiel einen Zenith Sidekick, der komplett mit Jeansstoff überzogen ist, passend zur Blue-Jeans-Welle von 1974.

Ich versuche, die Rückseite meines Fernsehers nach Möglichkeit nicht zu öffnen, also schicke ich das Video von meinem Computer durch eine Reihe von Adaptern, die das digitale Signal in ein analoges Format umwandeln, das mit dem VHF/UHF-Antennenanschluss meines Fernsehers kompatibel ist. Es gibt zwar All-in-One-Geräte für die Umwandlung von HDMI in VHF, aber ich habe sie noch nicht ausprobiert. Sie scheinen teurer zu sein als ein einfacher HDMI-zu-Composite-Adapter in Verbindung mit einem HF-Modulator. Erinnerst du dich an HF-Modulatoren, mit denen man etwa Heimcomputer oder Videogeräte über den Antennenanschluss an alte Fernseher ohne SCART-Eingang anschloss? Heutzutage gibt es sie oft in der Elektronikecke auf dem Flohmarkt oder im Gebrauchtkaufhaus zu einem günstigen Preis.

Wenn man wie ich mit Fernsehern arbeitet, benötigt man neben den Adaptern allerlei Kabel wie HDMI, Composite-Video (gelber Cinch-Stecker) und Kupfer-Koaxialkabel. Und wenn der Fernseher nur über Schraubklemmenanschlüsse verfügt, braucht man noch einen passenden Transformator mit Koaxial-zu-Gabelkabelschuh-Verbindungen. Wer ein gewisses Alter hat, bei dem kann dies Erinnerungen an das Anschließen eines Atari 2600 wecken. Wenn aber erst alles angeschlossen ist und funktioniert, fühlt es sich wie Magie an, Fenster vom Desktop auf die Röhre des Fernsehers zu ziehen.

Synthesizer, Ton und Video

Ursprünglich habe ich meine Live-Video-Experimente mit Klängen aus meiner Plattensammlung online gestellt, aber ich habe schnell gemerkt, dass ich mich bei meinen eigenen Inhalten nicht zu sehr auf die Musik anderer Leute stützen sollte. Ich begann, mit einem Spielzeug-Stylophon zu arbeiten, einem tragbaren Plastikdings, das man durch Tippen eines Stiftes auf eine Metall-Klaviatur spielt, oder indem man den Stift darüber zieht: Es klingt wie ein betrunkenes Huhn. Irgendwann bin ich dann auf ein modernes Stylophon mit eingebautem Hall und einigen regelbaren Filtern umgestiegen. Diese einfachen Regler waren mein Einstieg in die Konzepte der subtraktiven Synthese und damit der Grundlage von Synthesizern – und ich musste noch mehr lernen.

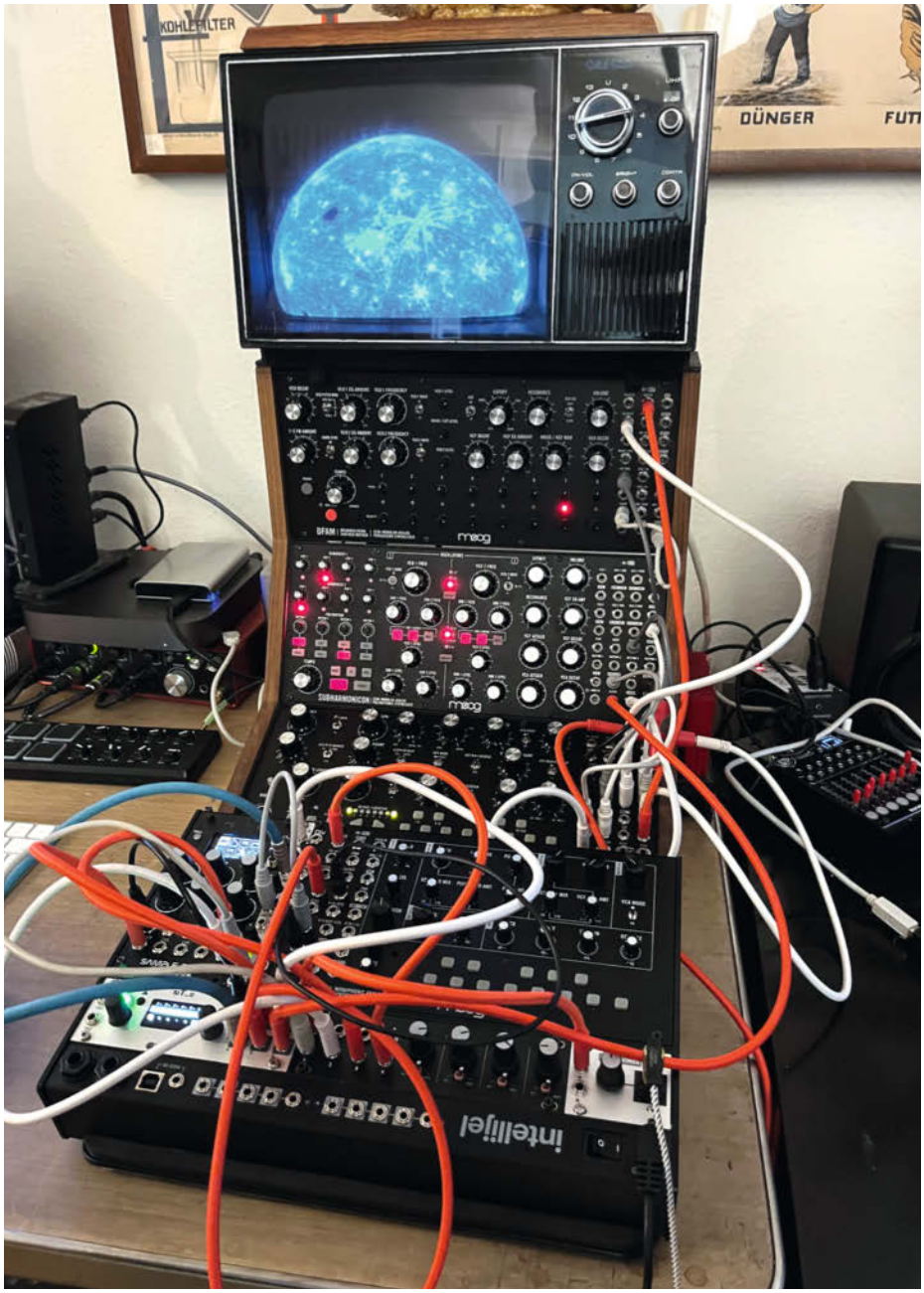
Es gibt viele Optionen für Einsteiger-Synthesizer und mein erstes Instrument war schließlich der Moog Mother-32. Ich kaufte ihn, weil er mir gut organisiert und ablesbar erschien, auch wenn ich noch keine Ahnung

hatte, was irgendwelche Begriffe und Regler bedeuteten. Er hat einen Vintage-Look, keine Spur von irgendeinem Bildschirm, Flanken aus Echtholz und rote Blinklichter – Teil meiner Überlegungen war, dass er zumindest vor der Kamera gut aussehen würde, falls ich ihn nicht gut klingen lassen könnte. Mit der Zeit erwarb ich noch ein Moog Subharmonicon und ein DFAM (Drummer From Another Mother), um das dreiteilige „Sound-Studio"-Set zu vervollständigen. Jedes dieser Geräte hat seinen eigenen Fokus und jedes eröffnet neue Möglichkeiten für die anderen.

Wie sich herausstellte, machen Synthesizer irgendwie süchtig, und es ist leicht, viel Zeit und Geld in dieses höchst unterhaltsame Spiel mit Audio-Logikrätseln zu investieren (siehe auch die weiteren Make-Artikel zum Thema Synthesizer in der Kurzinfo). Wenn du die Konzepte von Synthesizern ausprobieren willst, ohne Dein Konto zu überziehen, empfehle ich VCV Rack. Dabei handelt es sich um eine Art Emulationssoftware für analoge Synthesizer, die in der Grundversion kostenlos ist und eine große Gemeinschaft von Entwicklern und Benutzern hat. VCV Rack kann MIDI-Signale von Hardware-Controllern empfangen und ich verwende es ständig, um MIDI zur Steuerung anderer Dinge auszugeben, etwa für die Videosoftware VDMX.

VDMX wiederum ist eine Live-Performance-Videosoftware für den Mac, mit der ich Videos bearbeite, die ich an meine Fernsehgeräte sende. Es ist eine merkwürdige Software, aber auch sehr leistungsfähig. Die Benutzeroberfläche scheint in den späten 1990er Jahren stehengeblieben zu sein – es gibt immer noch keine Rückgängig-Funktion und die Fenster schweben losgelöst von jeglicher Art irgendeines übergreifenden Rahmens. Dem Uneingeweihten mag das veraltet erscheinen, aber es gibt viele Gründe, diesen Impuls zu überwinden und ich bin froh, dass ich dabei geblieben bin. Denn VDMX kann problemlos System-Audio- und MIDI-Geräte einbinden, es bietet einfache Optionen, Bilder auf mehrere Displays auszubreiten und es hat einen modularen Ansatz, der es jedem Slider oder Button erlaubt, mit jedem anderen Slider oder Button innerhalb von VDMX zu kommunizieren. Von der Adobe-Welt ist es meilenweit entfernt und ich genieße es sehr, seine Geheimnisse zu ergründen.

Ich versuche, für die meisten meiner Projekte Videos zu verwenden, die als Public Domain zur Verfügung stehen. Das Prelinger-Archiv etwa verfügt über eine ausgezeichnete und umfangreiche Sammlung solcher Videos. Ursprünglich war ich nur auf der Suche nach interessantem Filmmaterial, das nicht auf YouTube zu finden ist, aber es dauerte nicht lange, bis ich mich in diese wie in Zeitkapseln gespeicherten Stücke einst flüchtigen Films verliebte. Es gibt dort etwa alte Zigarettenwerbung, in der die Frische einer Marke gegenüber einer anderen angepriesen wird, peinliche Benimm-Dich-Filme aus den 1950er Jahren, Amateurclips von Roadtrips quer durchs Land und Werbefilme von Unternehmen über die neuesten Fortschritte in der einen oder anderen Technologie. Es ist eine wahre Fundgrube an bewegten Bildern, aus der man schöpfen kann, und auch für den Ton ist das Material hervorragend geeignet.



Die eingesetzten analogen Synthesizer produzieren nicht nur Klänge, sondern steuern über MIDI auch die Videosoftware VDMX.

Mein Studio

Wie bei vielen Menschen, die in San Francisco arbeiten, ist mein Arbeitsbereich auch mein Lebensraum. Ich habe eine Studiowohnung im Erdgeschoss mit viel Stauraum gemietet. Zugegebenermaßen ist das nicht die praktischste Situation für ein Vintage-Fernseher-Museum und eine Maker-Werkstatt, aber ich arbeite mit dem, was ich habe. Neben der Küche gibt es eine Speisekammer, die ich in ein Mini-Atelier verwandelt habe und in der ich derzeit meine Emaille-Pins verpacke, die ich bei Etsy verkaufe und die (wie könnte es anders sein) Fernseher zeigen. Im großen Raum verwende ich einen Mac Mini mit Intel-Prozessor, um VDMX zu betreiben, da es in der Vor-Apple-Silizium-Welt am stabilsten zu laufen scheint, im Speisekammer-Atelier habe ich

Keith Hammond

Links auf dem Monitor verteilen sich locker die Einzelfenster von VDMX vor die Webseite der US-Make.

Keith Hammond

Das Speisekammer-Atelier ist der Ort für eher chaotische Kreativität ...

zusätzlich einen einfachen M1-Mac-Mini zum Bearbeiten und Rendern meiner kurzen Social-Media-Clips.

Manchmal zeichne und male ich auch noch, und das Chaos, was ich dabei verursache, bleibt ebenfalls im Speisekammer-Atelier. Der Wohnbereich des Hauptraums beherbergt jetzt eine riesige Polycarbonat-Kuppel auf einem 55-Zoll-Flachbildfernseher für Pepper's-Ghost-Zwecke. Ursprünglich war es ein Designer-Sessel, den ich beim Trödler bei den Second-Hand-Sofas gefunden habe und fand, dass er stattdessen besser neben meiner Couch Platz finden sollte. Mein Schreibtisch ist ein alter Tanker Desk aus Metall, den ich von einem befreundeten Videoredakteur geerbt habe. Er hatte eine große Ablagefläche, die ich komplett mit Computer- und Audiogeräten gefüllt habe.

Der Wohnbereich ist notwendigerweise auch ein Fotostudio mit ein paar Licht- und Stativständern, die hier und da aufgestellt sind. Ich nutze eine Canon M6 Mark II, die seitlich auf einem Stativ steht, um vertikale Videos aufzunehmen. Ich verwende einen Ninja-V-Recorder, um das Beste aus dem HDMI-Feed der Kamera herauszuholen. Ein Diffusionsfilter in Kombination mit einem Graufilter (ND-Filter, Neutraldichte-Filter) auf dem Sigma-16-mm-Objektiv hilft, die wilde Strahlkraft der Kathodenstrahlröhren und LEDs in den Griff zu bekommen. Manchmal verwende ich ein altes russisches Objektiv mit einem Adapter und einem Speedbooster – einem Zwischenring, der die Brennweite verkürzt und die Lichtstärke erhöht –, um Weitwinkel- oder Makroaufnahmen zu machen.

Beim Filmen von Pepper's-Ghost-Erscheinungen gibt es eine Menge auszuprobieren. Je nach Bildschirm und gezeigtem Bild muss man die Umgebungsbeleuchtung entsprechend anpassen. Damit der Effekt lebendig wirkt, muss es insgesamt immer ein wenig dunkel sein. Gleichzeitig ist es aber wichtig, dass die Umgebung um den Effekt herum zumindest etwa zu sehen ist, daher kann es erforderlich sein, diverse kleine Beleuchtungskorrekturen vorzunehmen, bis alles passt. Ich verwende hinterher auch subtile Farb- und Beleuchtungsanpassungen in der kostenlosen Videobearbeitungssoftware DaVinci Resolve, um das Video so nachzubearbeiten, wie es in Wirklichkeit aussieht.

Keith Hammond

... während im großen Zimmer eher große Installationen ihren Platz finden, etwa die Pepper's-Ghost-Kuppel aus der Schale eines alten Designer-Sessels in Form einer transparenten Blase.

Selbstgebauter Flaschengeist

Du brauchst keine riesige Glaskuppel, um deine eigene Pepper's-Ghost-Illusion zu kreieren. Es ist ein sehr skalierbarer Effekt, für den nur einfache Materialien benötigt werden, die leicht zu beschaffen sind. Ich habe sogar Demo-Videos gemacht, in denen ich zeige, wie man einen Pepper's Ghost mit einem gewölbten transparenten Deckel eines Coffee-To-Go-Bechers, einer Snack-Verpackung und dem eigenen Telefon improvisiert.

Da Halloween eigentlich rund ums Jahr immer vor der Tür steht, wollen wir einen Pepper's Ghost in einem Erlenmeyerkolben beschwören. Der passt perfekt in in das Labor jedes verrückten Wissenschaftlers, in ein Spukhaus oder auf eine Gruselparty.

Vorlage drucken: Lade die PDF-Datei mit der Schablone für den Reflektor über den Link in der Kurzinfo herunter und schneide sie aus (Bild 1). Diese Form liegt später schräg in der Flasche, um das Bild auf dem Display zu reflektieren. Die Form sollte der Geometrie des Kolbens entsprechen, damit das Plastik nicht zu sehr sichtbar ist und gerade sitzt. Um das hinzubekommen, müssen wir uns mit Hilfe von dünnem Karton an die Form herantasten.

Vorlage ausschneiden: Übertrage die gedruckte Vorlage auf den dünnen Karton und schneide die Form aus diesem aus. Pappkarton ist steifer als Druckerpapier und entspricht damit eher der Form des fertigen Reflektors im Kolben (Bild 2). Auf der anderen Seite ist er viel besser sichtbar als das durchsichtige Plastik, sodass du besser erkennen kannst, wo du dein Ergebnis noch verfeinern musst. Und Pappe ist viel billiger, falls Du feststellst, dass die Vorlage nach den Korrekturen zu klein geworden ist.

Probesitzen im Kolben: Rolle die Kartonvorlage um die Längsachse, sodass sie durch den Hals passt, und schiebe sie dann in den Kolben. Bewege sie mit einer Pinzette oder Zange, um sie in einem Winkel von etwa 45° zu positionieren (Bild 3). Prüfe, ob sich die Pappe eventuell durchbiegt oder der Winkel nicht passt. Wenn das Pappstück nicht gerade im gewünschten Winkel sitzt, hole es wieder raus und schneide es weiter zurecht oder

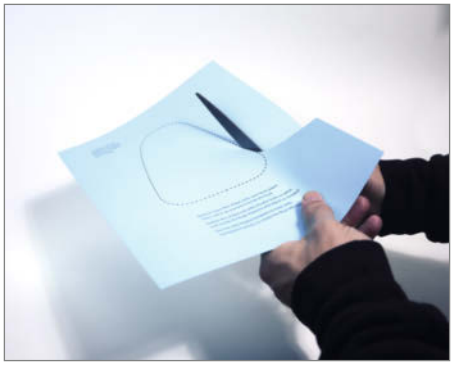


Bild 1

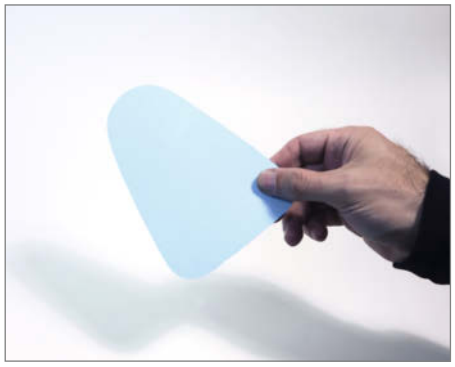


Bild 2

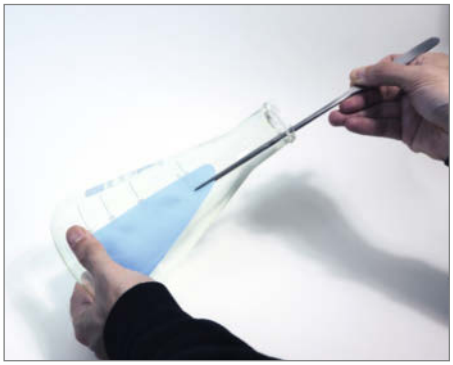


Bild 3

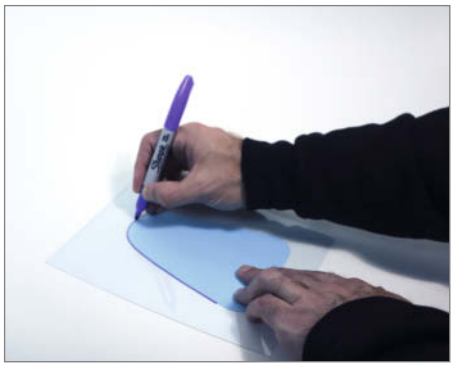


Bild 4

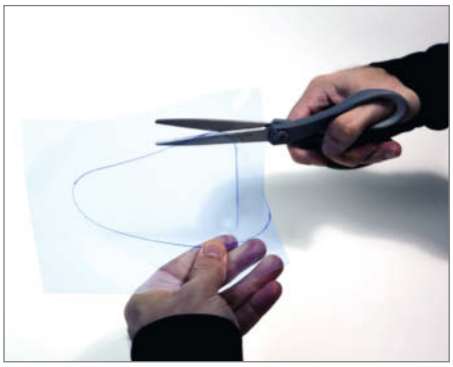


Bild 5

Bild 6

Bild 7

Bild 8

Bild 9

starte mit einem frischen Stück Pappe neu. Falls nötig, lohnt es sich, es auch ein paar mal zu versuchen, das verbessert die Wirkung deutlich.

Sollte die Form deines Gefäßes stark von dem hier benutzten Kolben abweichen, ist die Anpassung aufwändiger; eventuell musst du die Pappvorlage auch etwas größer zuschneiden, damit sie zu deinem Glas passt.

Reflektor zuschneiden: Sobald du einen akzeptablen Zusschnitt für die Pappe gefunden hast, verwende diese als Schablone für die klare Polycarbonatplatte. Wenn dein Kunststoff eine dünne Schutzfolie hat, lasse diese noch drauf, während du die Schablone mit einem Permanentmarker nachzeichnest (Bild 4); auf diese Weise verschwindet jede Spur des Markers zusammen mit der Folie, wenn du sie nach dem Schnitt abziehst. Schneide den Kunststoff vorsichtig mit einer Schere entlang der Markierung aus (Bild 5).

Einrollen: Der Kunststoff ist sehr flexibel, zumindest, wenn es Polycarbonat ist: Der Reflektor sollte sich daher leicht zusammenrollen lassen, ohne zu knicken oder zu brechen (Bild 6).

Einschieben: Schiebe den aufgerollten Kunststoff mit dem breiten Ende zuerst in den Kolben (Bild 7). Wenn alles richtig gelaufen ist, sollte der Kunststoffreflektor dann in der gleichen Position aufspringen, an der die Schablone vorher saß (Bild 8).

Testvideo suchen: Bevor du dich für die endgültige Gestalt des Geistes in Deinem Erlenmeyerkolben entscheidest, solltest du testen, ob alles funktioniert, wie es soll. Suche dir dafür ein Video oder ein Bild auf schwarzem Hintergrund. Alles, was schwarz ist, wird bei der Pepper's-Ghost-Illusion transparent, sodass das hellere Objekt im Fokus des Videos zu schweben scheint.

Bei einer Google-Suche findet man schnell eine Vielzahl von Videos mit verschiedenen Kreaturen und Objekten vor schwarzem Hintergrund. Wähle ein Video mit möglichst hellen Farben vor einem sehr dunklen Hintergrund. Ich verwende persönlich meist ein bestimmtes Video eines Goldfisches dazu, inzwischen so oft, dass ich für eine hochaufgelöste Version davon sogar bezahlt habe. Der Goldfisch ist sozusagen mein Maskottchen geworden.

Generalprobe: Dimme das Licht im Raum und stelle dein vorbereitetes Gefäß auf den Bildschirm, auf dem das Video läuft (Bild 9). Es lebt!

Vielleicht stellst du fest, dass das Bild auf dem Kopf steht. In diesem Fall drehst du den

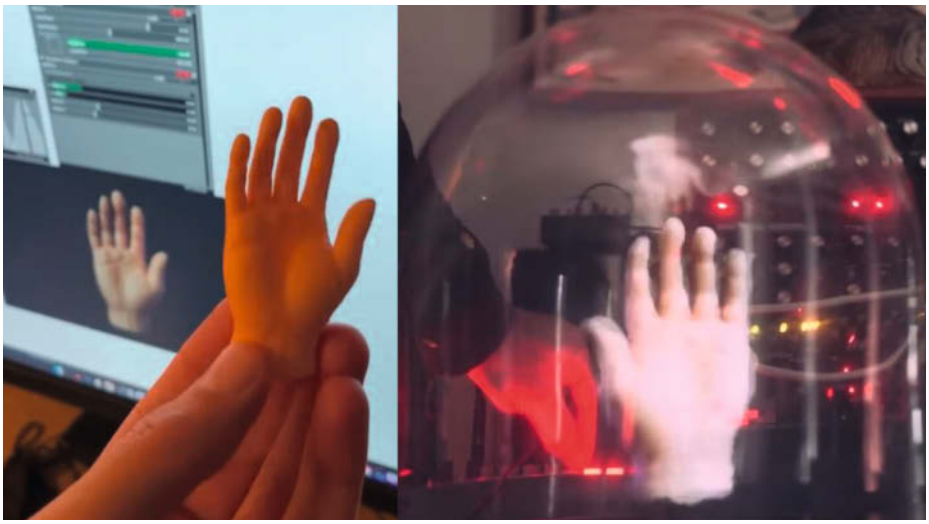


Bild 10

Bildschirm einfach so, dass er in die andere Richtung zeigt.

Denke daran, dass alles, was sich im Kolben befindet, ein Spiegelbild dessen ist, was er reflektiert, sprich: In der Flasche erscheint das Video spiegelbildlich. Dies ist vor allem wichtig, wenn du Text im Kolben anzeigen willst. In diesem Fall kannst du das Video mit einer Videobearbeitungssoftware horizontal spiegeln, damit es richtig reflektiert wird.

Feinschliff

Jetzt, wo du einen funktionierenden Flaschengeist hast, kommt der kreative Teil: Du musst dein Geschöpf nur noch weiter ausarbeiten. Wie würde es aussehen, wenn der Kolben mit grüner Flüssigkeit gefüllt wäre? Mit seltsamen Kreaturen, Geistern, Augäpfeln oder Körperteilen (Bild 10)? Hast du noch ein anderes Chemieglas oder besonderes Gefäß, in das ein schwimmendes oder schwebendes Experiment passen würde? Möchtest du vielleicht eine eigene Animation dafür erstellen?

Optionen für Animationen: Wenn du ein iPad hast, bietet die beliebte Zeichen-App Procreate Werkzeuge für einfache Animationen. Damit ist es recht simpel, eine Einzelbild-Animation vor einem schwarzen Hintergrund zu erstellen und sie dann als Animation im Kolben in einer Schleife laufen zu lassen. Wenn du dich mit anderen Grafikanwendungen wie Blender oder Adobe After Effects auskennst, noch besser – dann gibt es viel Raum für selbst ausgedachte Kreationen verrückter Wissenschaftler.

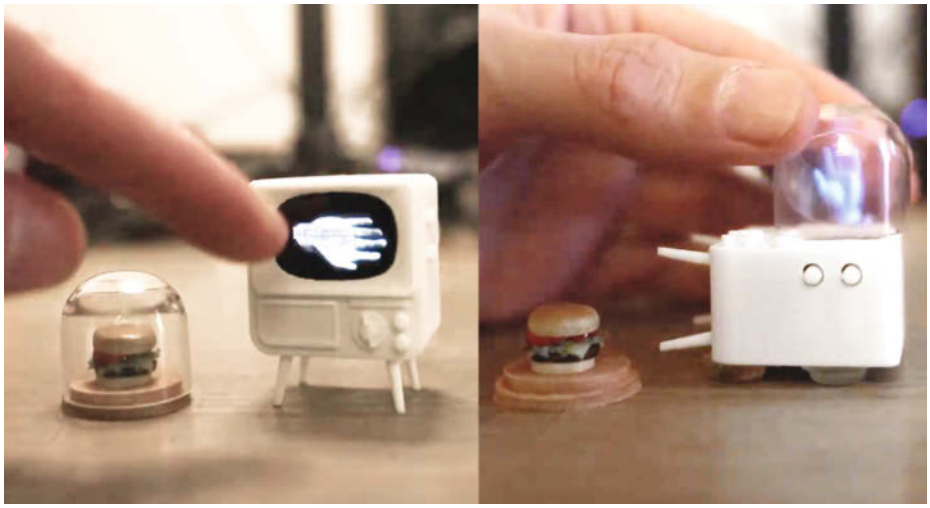


Bild 11

iPad als externes Display: Manchmal benutze ich über die Anzeigeeinstellungen meines Mac das iPad als externes Display (die Funktion heißt Sidecar). Dann verwende ich die Echtzeit-Videosoftware VDMX, um audioreaktive Pepper's-Ghost-Effekte vom Bildschirm meines iPads in einen Glasbehälter zu übertragen. Obwohl Sidecar eine hervorragende macOS-Integration aufweist, kann es bei der Ausführung von Videosoftware zu unerwartetem Verhalten kommen. Eine gute Alternative könnte die App YAM (Yet Another Monitor) für Mac und iOS sein. YAM scheint eine sehr geringe Latenzzeit zu haben und unterstützt das iPhone und andere Geräte.

Andere Gläser und Displays aller Größen: Für dieses Projekt kannst du fast alle Größen und Formen von Gläsern oder anderen transparenten Behältern verwenden. Ich habe schon billige Schneekugeln und Glockengläser verwendet, von riesigen bis hin zu dem winzigen aus Bild 11, das auf mein TinyTV (von tinycircuits.com) passt. *—pek*

Pepper's Ghosts, von Charles Dickens bis Tupac Shakur

Die berühmte Technik zur Geisterprojektion wurde 1858 vom englischen Ingenieur Henry Dircks erfunden, um Spiritisten zu entlarven und Besseres zu bieten als deren Laterna-Magica-Phantasmagorien-Spiele. Die Illusion wurde 1862 von John Henry Pepper im Theater eingeführt – die beiden Männer teilten sich ein Patent – und war bald darauf mit Erlaubnis des Autors Charles Dickens beim Stück „The Haunted Man" auf der Bühne sehen (Bild 12).

Steige in deinen Doom Buggy und lass' dich in Disneylands 54 Jahre alter Geistervilla-Attraktion von einer Vielzahl von Gespenstern erschrecken – die meisten von ihnen werden immer noch von animatronischen Puppen und der Dircks/Pepper-Illusion dargestellt (Bild 13).

Die Digital Natives entdeckten Pepper's Ghost wieder, als die animierten Avatare der Gorillaz 2005 und 2006 bei Musikpreisverleihungen per 3D-Projektion auf der Bühne erschienen (Bild 14) – und erneut, als der verstorbene Tupac Shakur 2012 mit Snoop Dogg und Dr. Dre beim Coachella „live" auftrat und damit eine verrückte Mode der „Hologramm-Touren" von noch lebenden, bereits toten und völlig virtuellen Künstlern auslöste. *—Keith Hammond*

Wikimedia – gemeinfrei

Bild 12

Gorillaz

Bild 14

Disney

Bild 13

Smarter Garagentorantrieb
mit ESPHome und Home Assistant

Bei einer Hausautomatisierung sollte die Steuerung des Garagentors nicht fehlen. Da oftmals noch alte Garagentorantriebe im Einsatz sind, zeigen wir in diesem Artikel, wie man so einen Antrieb für die Hausautomatisierung mit Home Assistant fit macht.

von Tim Riemann

Zu einer Garage, die einen Garagentorantrieb besitzt, gehört im Normalfall eine Funkfernsteuerung, mit der man den Antrieb ohne Probleme bedienen kann, falls die sich nicht wieder einmal unter einen der Autositze verabschiedet hat. Leider ist die Reichweite, je nach Umgebung, begrenzt und je nach Alter des Antriebs kann man ihn auch nicht so ohne Weiteres in die Hausautomatisierung einbinden, um z. B. auch unterwegs zu kontrollieren, ob man das Garagentor auch wirklich geschlossen hat. In der Garage des Autors verrichtet eine Hörmann ProMatic 2 ihren Dienst und soll smart gemacht werden. Das Prinzip lässt sich aber sehr einfach auf Antriebe anderer Hersteller übertragen.

Hardware

Um eine passende Hardware zu bauen, wird zuerst ein Überblick über die zu steuernde Toröffneranlage benötigt. Ein Blick in die Bedienungsanleitung zeigt, dass es die Möglichkeit gibt, den Garagentorantrieb über einen nachrüstbaren Schlüsselschalter zu schalten. Die entsprechenden Anschlüsse sind auf der Tor-Elektronik auf Steckleisten geführt (Bild 1). Dabei wird der Kontakt für den Impuls über den Schlüsselschalter auf 0 V gezogen und das Garagentor daraufhin geöffnet. Die Verwendung der zwei Kontakte ist natürlich denkbar einfach und lässt sich entweder über ein Relais oder einen Optokoppler realisieren. Darüber hinaus liegen an dem gleichen Stecker auch noch 24 V Spannung an, die als Versorgungsspannung für den Mikrocontroller verwendet werden kann. Hier kann ein Standard-Step-Down-Konverter verwendet werden, der eine Eingangsspannung von 24 V unterstützt und 5 V Ausgangsspannung zur Verfügung stellt.

Da die Einbindung des Garagentors in Home Assistant über ESPHome geschehen soll, fiel die Wahl auf einen Mikrocontroller, der von ESPHome bereits gut unterstützt wird: einen Wemos D1 mit ESP8266. Die Anzahl der freien Pins ist für diese Anwendung vollkommen ausreichend.

Für den Anwendungsfall soll mindestens das Garagentor geöffnet und geschlossen werden können und detektiert werden, in welchem Zustand sich das Garagentor befindet. Es wird daher ein Pin benötigt, der einen Optokoppler vom Typ PC817 ansteuert, der dann den Impuls auf den Eingang des Garagentorantriebs weitergibt. Hier fiel die Wahl auf den Pin D6. Zur Detektion des Garagentorstatus kommt ein Magnetschalter zum Einsatz, der am oberen Ende des Garagentors montiert wird. Ist das Garagentor zu, so ist auch der Kontakt geschlossen. Wird das Garagentor geöffnet (einige wenige Zentimeter reichen dabei), wird auch der Kontakt geöffnet und der Zustand entsprechend gemeldet. Für

Kurzinfo

- Herkömmlichen Garagentor-Antrieb mit ESP8266 steuern
- Übermittlung von Torstellung, Temperatur und Feuchtigkeit in der Garage an Home Assistant
- Steuerung des Tores über Home-Assistant-Oberfläche oder per Android Auto

Checkliste

Zeitaufwand:
6 bis 8 Stunden

Kosten:
50 Euro

Material

- **ESP8266-Board Wemos D1 mini**
- **Temperatur-/Feuchtesensor BME280**
- **Step-Down-Wandler** Eingangsspannung mind. 24 V, Ausgangsspannung 5 V
- **Feuchtraum-Aufputzdose**
- **Magnetkontakt**
- **Optokoppler PC817**
- **Schrauben/passende Dübel für Aufputzdose**

Werkzeug

- **Lötausrüstung**
- **Bohrmaschine**

Mehr zum Thema

- Heinz Behling: Marktübersicht für Smart Home Boards, Make 4/23, S. 12
- Heinz Behling: Intelligentes Heim mit Home Assistant, Make 1/21, S. 100
- Heinz Behling: Smarthome-Firmware für ESP8266/32-Module sichern und flashen, Make 4/20, S. 34

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

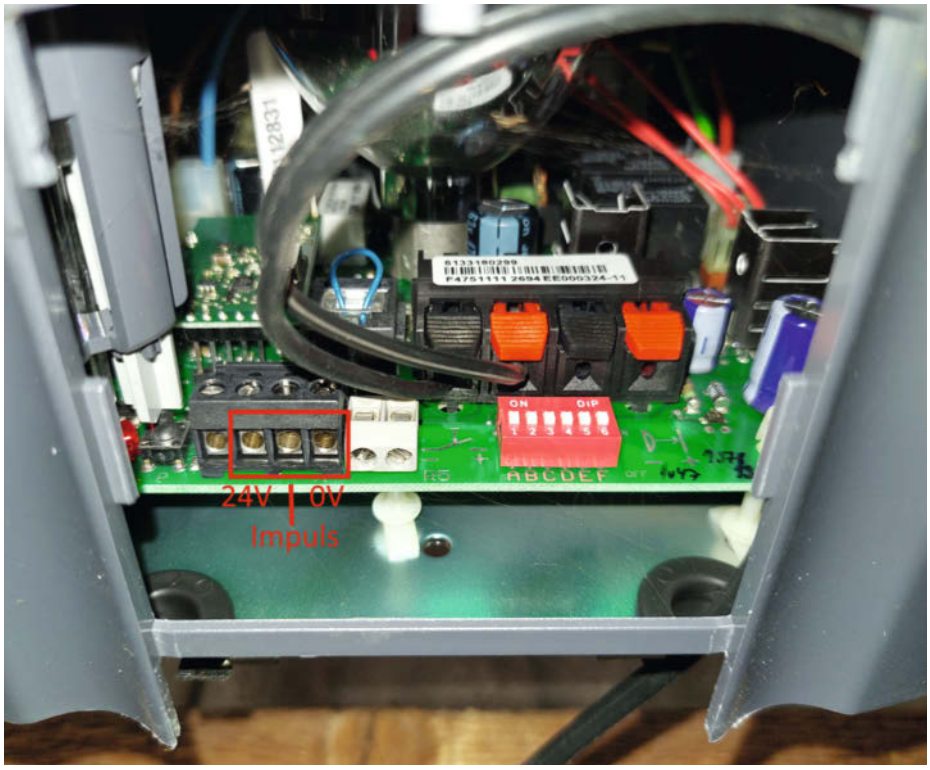


Bild 1: Steckerbelegung eines Hörmann-ProMatic-2-Garagentorantriebs

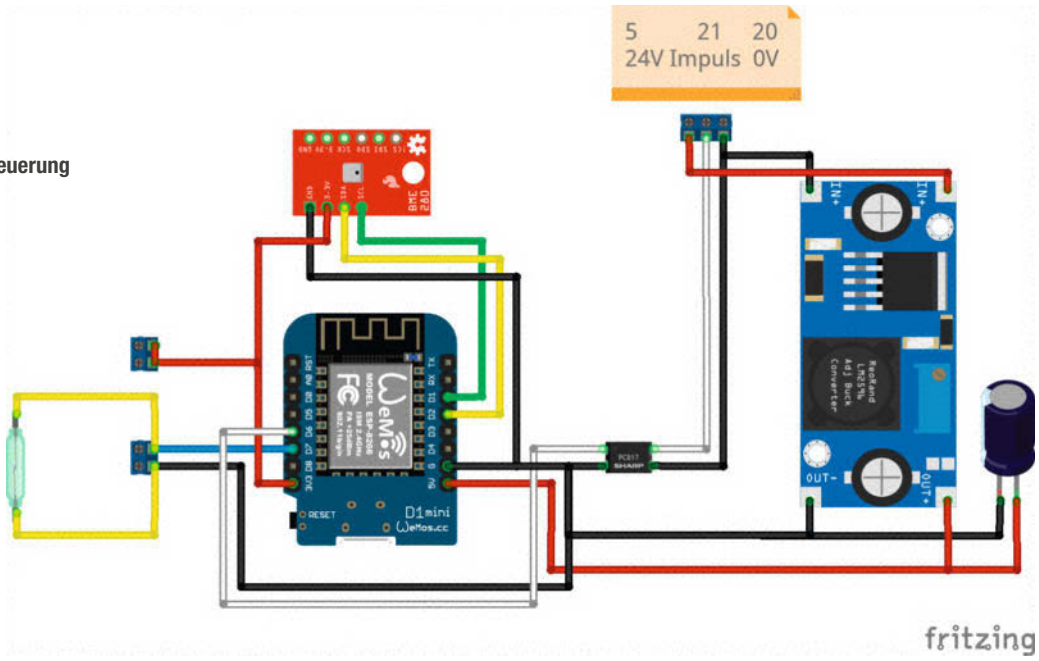


Bild 2: Schaltplan der Garagentorantrieb-Steuerung

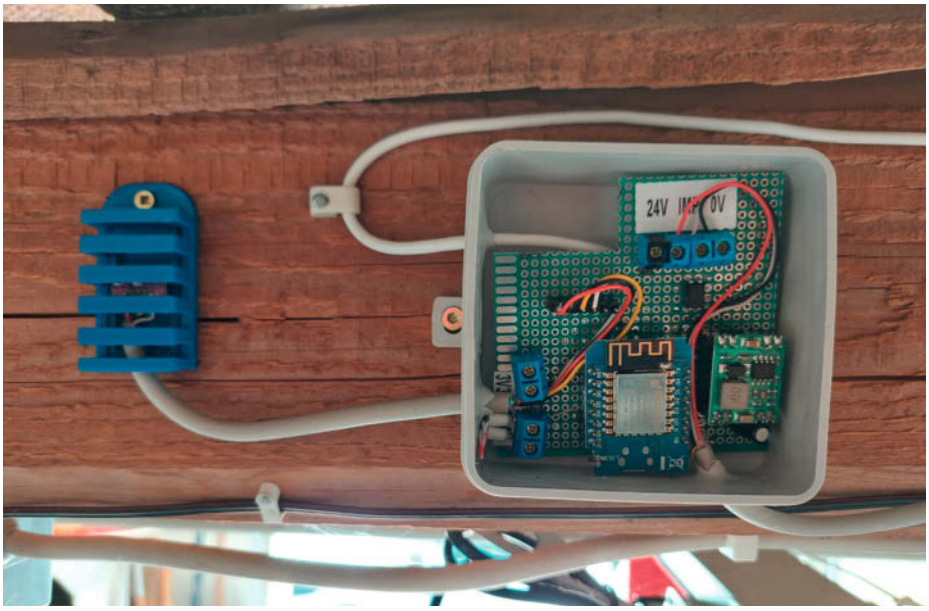


Bild 3: Einbau der Schaltung in einer Kabel-Abzweigdose

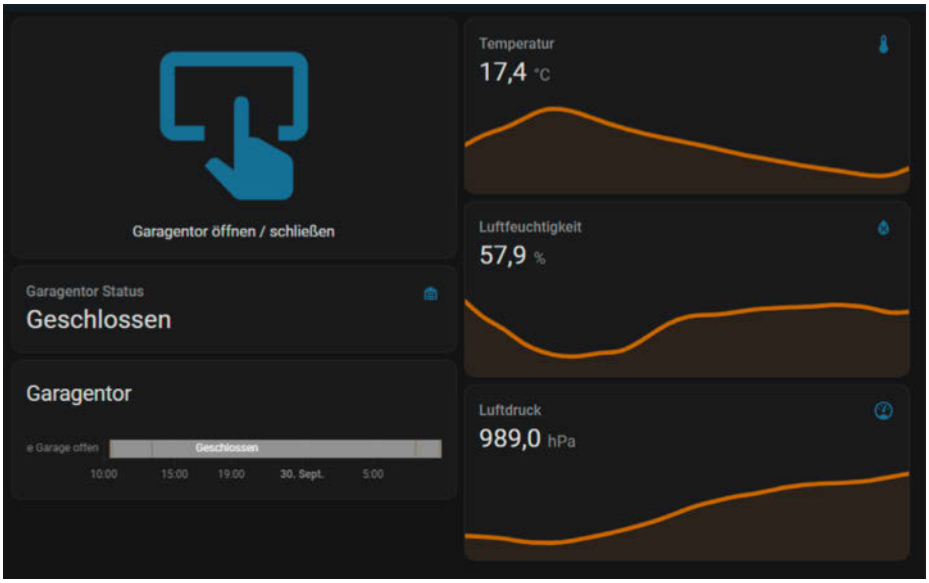


Bild 4: Einfaches Kontrollfenster in Home Assistant

den Magnetschalter ist der Pin D7 vorgesehen, der gegen GND gezogen wird.

Zum Schluss sollte zusätzlich auch noch die Temperatur und Luftfeuchtigkeit in der Garage überwacht werden. Daher fiel die Wahl auf einen BME280-Sensor, der bereits in vielen Projekten eingesetzt wurde. Neben der Versorgungsspannung benötigt der BME280 noch zwei weitere Pins für die Steuerung über den I^2C Bus. Dafür werden die Pins D1 für SCL und D2 für SDA verwendet. Der Schaltplan (Bild 2) zeigt die Verbindungen zwischen den einzelnen Komponenten. Auf der rechten Seite ist ein Step-Down-Konverter zu sehen, der die Versorgungsspannung für den Mikrocontroller zur Verfügung stellt, und ein 1000-µF-Kondensator, der Lastspitzen abpuffern soll.

Der Aufbau der Schaltung erfolgte auf einer einfachen Lochrasterplatine, die bequem in eine Feuchtraumkabel-Abzweigdose hineinpasst (Bild 3). Der BME280 wurde nicht darin verbaut, sondern in einem eigens gedruckten Gehäuse untergebracht. Hier finden sich einige gute Designs auf den Seiten mit 3D-Vorlagen (z. B. bei Thingiverse, Link siehe Kurzinfo).

ESPHome-Konfiguration

Nachdem die Hardware aufgebaut ist, kann die Konfiguration der Firmware für den Wemos D1 in ESPHome erfolgen. Da in den vorherigen Ausgaben bereits genauer auf das Flashen der Firmware mit dem ESPHome-Dashboard eingegangen wurde (siehe „Mehr zum Thema“ in der Kurzinfo), erklären wir hier nur kurz den Ablauf. Im Dashboard wird ein neues Device vom Typ ESP8266 unter einem beliebigen Namen angelegt. Im Anschluss wird die neu angelegte Konfiguration editiert und um die Definition der Ein- und Ausgänge (siehe Listing) ergänzt. Danach muss nur noch die Firmware auf den ESP8266 übertragen werden und die Hardware an den Garagentorantrieb angeschlossen werden.

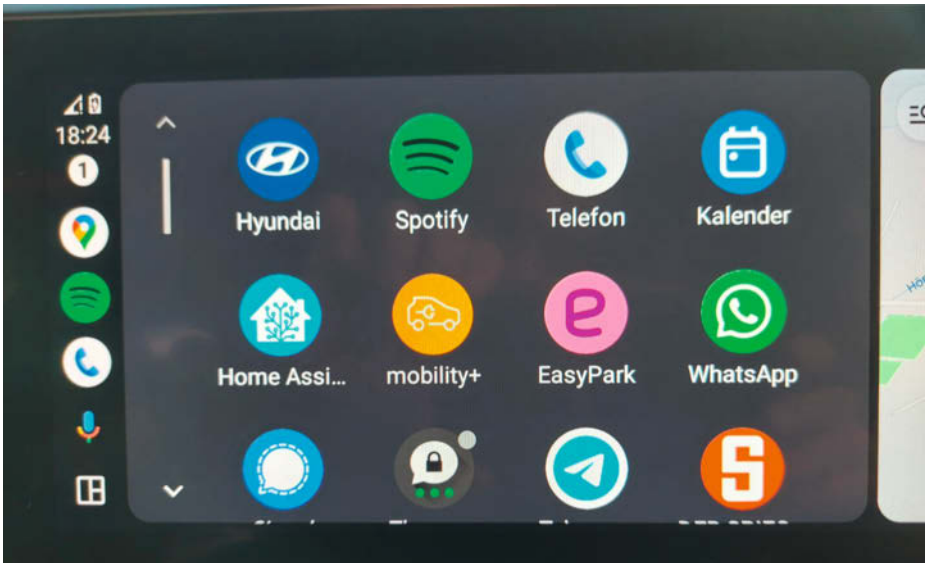


Bild 5: Wenn Sie im Auto ein Android-Gerät haben, können Sie dort die passende Home-Assistant-App installieren.

Im Listing wird zuerst definiert, an welchen Pins des Wemos D1 der I²C-Bus angeschlossen ist. Dieser wird für den BME280-Sensor benötigt, der im Anschluss definiert wird. In diesem Bereich werden die Namen für den Sensor vergeben, mit denen er in Home Assistant auftaucht. Auch das Update-Intervall kann für den Sensor eingestellt werden, wobei 60 Sekunden ausreichend sein sollten.

Als Nächstes wird der Pin für den Optokoppler definiert, der letztendlich den Garagentorantrieb steuert, und eine eindeutige ID vergeben. Diese ID ist für die Verknüpfung des Buttons notwendig, der danach konfiguriert wird. Genau wie der BME280 bekommt der Button einen Namen zugewiesen. Zusätzlich wird noch eine Zeitdauer (Duration) für die Dauer des Schaltimpulses angegeben. 500 Millisekunden sind für die ProMatic ausreichend. Bei anderen Antrieben können andere Werte notwendig sein. Dann muss man ins entsprechende Handbuch schauen oder einfach ausprobieren.

Zum Schluss folgt noch die Definition des Eingangs für den Magnetschalter. Auch der erhält einen Namen, der später in Home Assistant angezeigt wird. Zusätzlich legt der Befehl `pullup true` fest, dass ein interner Pull-Up-Widerstand verwendet werden soll (der Magnetschalter schaltet gegen GND). Die `Device-Class` lautet `garage_door`. Über die `Device-Class` kann Home Assistant direkt erkennen, um welche Art von Sensor es sich handelt und zeigt dann den Status des Garagentors mit den passenden Piktogrammen (Tür offen oder geschlossen) an.

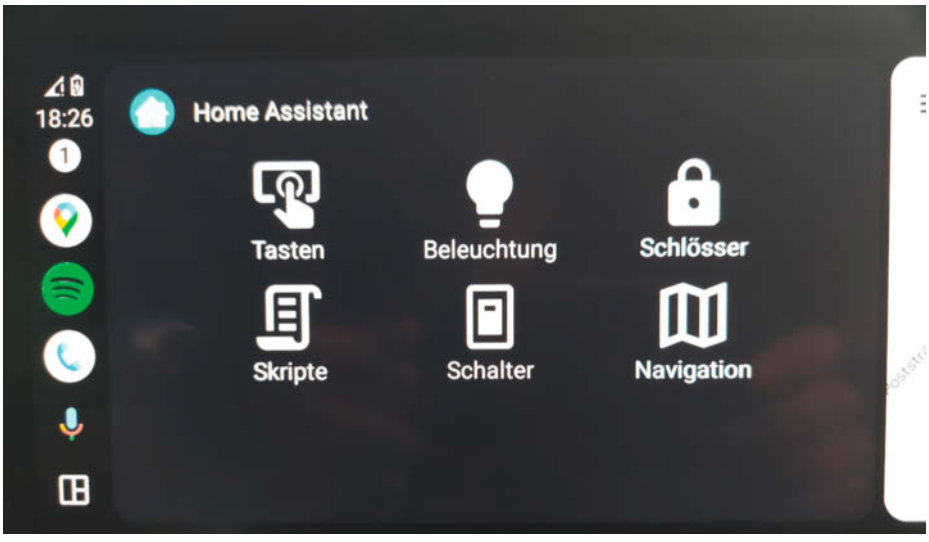


Bild 6: Die Steuerung des Garagentors wird automatisch ins Tasten-Menü eingebaut.

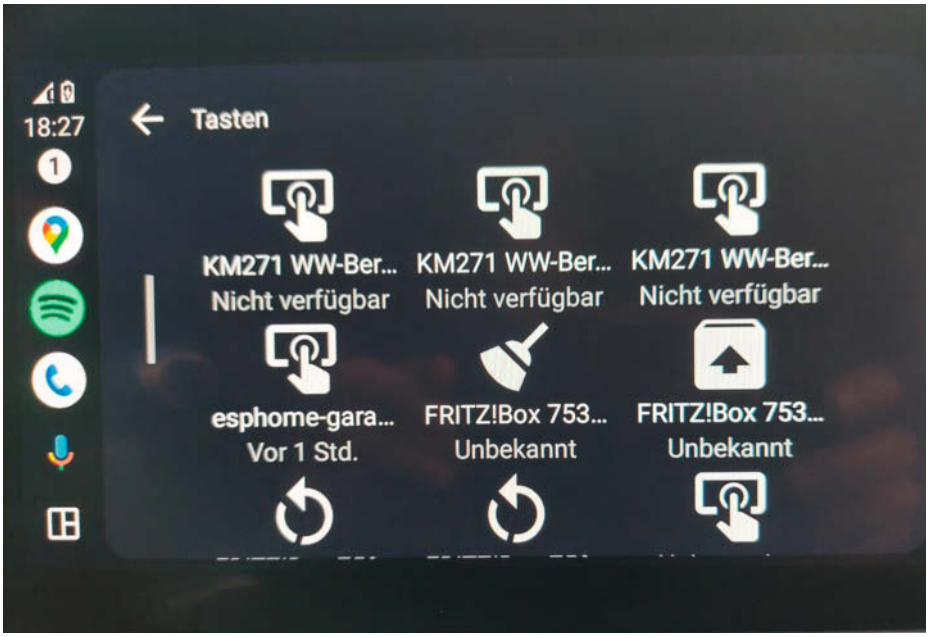


Bild 7: Der Schalter fürs Tor verbirgt sich unter „esphome-gara...“.

Listing 1: Die Firmware

```
# Default Bus for the BME280 sensor (SCL = 1, SDA = D2)
i2c:
  sda: GPIO4
  scl: GPIO5
  scan: true
  id: sensor_bus

# BME280 sensor definition (Address = 0x76)
sensor:
  - platform: bme280
    temperature:
      name: "Garage Temperatur"
      oversampling: 16x
    pressure:
      name: "Garage Luftdruck"
    humidity:
      name: "Garage Luftfeuchtigkeit"
    address: 0x76
    update_interval: 60s

# Button configuration (D6)
output:
  - platform: gpio
    pin: GPIO12
    id: door_open

button:
  - platform: output
    name: "Tor öffnen / schließen"
    output: door_open
    duration: 500ms

# Input for reed relais (door open / closed - D7)
binary_sensor:
  - platform: gpio
    pin:
      number: GPIO13
      inverted: false
      mode:
        input: true
        pullup: true
    name: "Garage offen"
    device_class: garage_door
```

Einbinden in Home Assistant

Nachdem die Firmware auf den ESP8266 geflashed wurde, startet der Mikrocontroller neu und meldet sich automatisch im konfigurierten WLAN an. Wenige Sekunden später findet Home Assistant das neue Gerät bereits: Es taucht zusammen mit den dazugehörenden Entitäten unter „Einstellungen/Geräte & Dienste/ESPHome“ auf.

In der Menüleiste auf der linken Seite der Web-Oberfläche des Home-Assistant-Servers setzen Sie nun einen Eintrag für die Garage ein (in Home Assistant Dashboard genannt). Dazu auf der Übersichtsseite zunächst auf die drei Punkte in der rechten oberen Ecke klicken und mit „Dashboard bearbeiten“ und danach am selben Ort mit „Dashboard verwalten“ den Eintrag anlegen. Als Name eignet sich zum Beispiel „Garage“. Falls ein Pictogramm gewünscht ist: `mdi:garage` passt.

Nach Bestätigung durch „Erstellen“ erscheint der Eintrag bereits im linken Menü. Klicken Sie darauf und bearbeiten Sie das neu erstellte Dashboard mithilfe des „Raw-Konfigurationseditor“ aus dem 3-Punkte-Menü. Das ist einfach, denn der vorhandene Text muss lediglich durch den aus Listing 2 ersetzt werden. Nach dem Speichern erscheint dann der Fensterinhalt (Bild 4).

Steuerung mit Android Auto

Die Home-Assistant-App gibt es seit Kurzem auch für Android Auto. Falls Sie ein Auto mit Android-fähigem Gerät besitzen und diese App installieren (Bild 5), können Sie künftig auch über den Touchscreen des Fahrzeugs auf den Garagentor-Öffner zugreifen. Leider ist dieser Modus momentan noch etwas umständlich zu benutzen, da es in der Auto-App noch keine Dashboards gibt. Der Zugriff auf den Garagentorantrieb ist nur über das Tasten-Menü (Bild 6) möglich. In Zukunft kann dann das Garagentor ganz bequem direkt aus dem Auto heraus per Tipp auf den Touchscreen geöffnet werden (Bild 7). Das mühsame Suchen nach dem irgendwo verschwundenen Fernsteuergeber fürs Tor entfällt.

Fazit

Auch alte Geräte lassen sich relativ einfach in Home Assistant integrieren, wenn man sich ein wenig mit der Hardware des zu steuernden Gerätes befasst. Mit ESPHome und einer kleinen Konfigurationsdatei kommt man dabei um das Schreiben von kompliziertem Quellcode herum. Die neue Hardware wird dann automatisch in Home Assistant gefunden und kann sofort in den Dashboards eingesetzt werden. Über Android Auto kann man sogar aus dem Auto heraus die neuen Funktionen nutzen. Mittels des hier gezeigten Weges lassen sich auch noch viele andere Geräte smart machen und in Home Assistant integrieren. *—hgb*

Listing 2: Dashboard

```
views:
  - title: Garage
    path: garage
    icon: mdi:garage
    badges: []
    cards:
      - type: vertical-stack
        cards:
          - show_name: true
            show_icon: true
            type: button
            tap_action:
              action: toggle
            entity: button.esphome_garage_tor_offnen_schliessen
            name: Garagentor öffnen / schließen
            show_state: false
            icon: ''
          - type: entity
            entity: binary_sensor.esphome_garage_garage_offen
            state_color: true
            name: Garagentor Status
          - type: history-graph
            entities:
              - entity: binary_sensor.esphome_garage_garage_offen
            title: Garagentor
      - type: vertical-stack
        cards:
          - graph: line
            type: sensor
            detail: 1
            entity: sensor.esphome_garage_garage_temperatur
            name: Temperatur
          - graph: line
            type: sensor
            detail: 1
            entity: sensor.esphome_garage_garage_luftfeuchtigkeit
            name: Luftfeuchtigkeit
          - graph: line
            type: sensor
            detail: 1
            name: Luftdruck
            entity: sensor.esphome_garage_garage_luftdruck
```

Universal-IR-Fernbedienung

Infrarot-Fernbedienungen versenden in Signalen verschlüsselte Zeichenketten als Befehle. Mit Mikrocontroller und IR-Empfänger oder -Transceiver lassen sich diese entschlüsseln wie auch nachbilden. Ein Smartphone und ein WLAN-fähiger Mikrocontroller ersetzen so eine ganze Sammlung alter IR-Fernbedienungen.

von Kai Schauer

guteksk7 / Shutterstock.com, lorenzo gambaro / Shuttestock.com, Montage: Make

Wenn ich abends einen Film über den Beamer sehen will, sind verschiedenste Geräte im Spiel: Beamer, Blu-Ray-Player, Soundbar und Leinwand. Am Ende liegen vier Fernbedienungen auf dem Couchtisch. Dann läuft der Film. Das Licht ist aus. Und wenn's so richtig spannend ist, klingelt jemand an der Haustür. Dann beginnt die Suche nach der Fernbedienung des Blu-Ray-Players. Weil die Fernbedienung nicht beleuchtet ist und alle Lampen aus sind, ist auch der Pause-Knopf schwer zu finden. Wenn es endlich geschafft ist, renne ich zur Tür. Der Besuch ist längst wieder gegangen.

Irgendwann ist der Film zu Ende. Dann müssen wieder alle vier Fernbedienungen bedient werden, um die Geräte auszuschalten und die Leinwand hochzufahren. Kommt Euch die Situation bekannt vor? Mir ist darüber hinaus auch noch aufgefallen, dass ich bei jeder Fernbedienung nur ein paar wenige Knöpfe regelmäßig benutze.

Da muss Abhilfe geschaffen werden. Neben den vier Fernbedienungen liegt auch noch das Handy auf dem Couchtisch. Es wäre doch großartig, wenn ich mit dem Handy gleich auch noch alle anderen Geräte bedienen könnte. Das Display des Handys ist hervorragend beleuchtet. Wenn es gelingt eine Oberfläche zu programmieren, kann auch die Schriftgröße eingestellt werden. Im Idealfall ließen sich auch beliebige Funktionen ergänzen, löschen oder ändern.

Die Idee der hier vorgestellten Lösung ist es, eine universelle Infrarot-Fernbedienung mit WLAN zu bauen. Doch da stellt sich eine weitere Herausforderung: Ich habe kein WLAN. Die meisten Lösungsansätze, die im Internet zu finden sind, basieren auf einem vorhandenen WLAN-Router, der das Management übernimmt. Ich entscheide mich dafür, ein WLAN-Modul zu verwenden und es als Access Point zu betreiben. So kann auch jeder Besucher, der von mir das Passwort erhält, diese Lösung auf seinem eigenen Smartphone verwenden.

Hardware und Programmierung

Bei dem WLAN-Modul habe ich mich für ein ESP01 entschieden. Zwei GPIO-Pins genügen für mein Vorhaben: Ich möchte ja nur eine IR-LED ansteuern und zusätzlich ein Relais schalten. Das Relais brauche ich, um eine Lampe ein- und ausschalten zu können, bevor ich beim Klingeln zur Tür renne, um dabei nicht über die Latschen zu stolpern.

Die Programmierung erfolgt über die Arduino IDE. Für die Verwendung von IR-LEDs und die Steuerung über WLAN greife ich auf

Kurzinfo

» So entschlüsselt man IR-Signale
» IR-Fernbedienung über ESP8266-WLAN-Modul als Server
» HTML-Seite mit CSS-Buttons auf dem Smartphone als Bedienoberfläche

Checkliste

Zeitaufwand:
etwa ein Wochenende

Kosten:
30 bis 40 Euro

Programmieren:
Arduino-IDE und ESP8266 (ESP01 und NodeMCU)

Löten:
Lochrasterplatine mit Bauelementen

Hochspannung:
optional mit 230-V-Installation

Werkzeug

» **Multimeter**
» **Lötkolben** und Zubehör
» **Seitenschneider**
» **Oszilloskop** optional
» **LCR-Tester** optional

Material

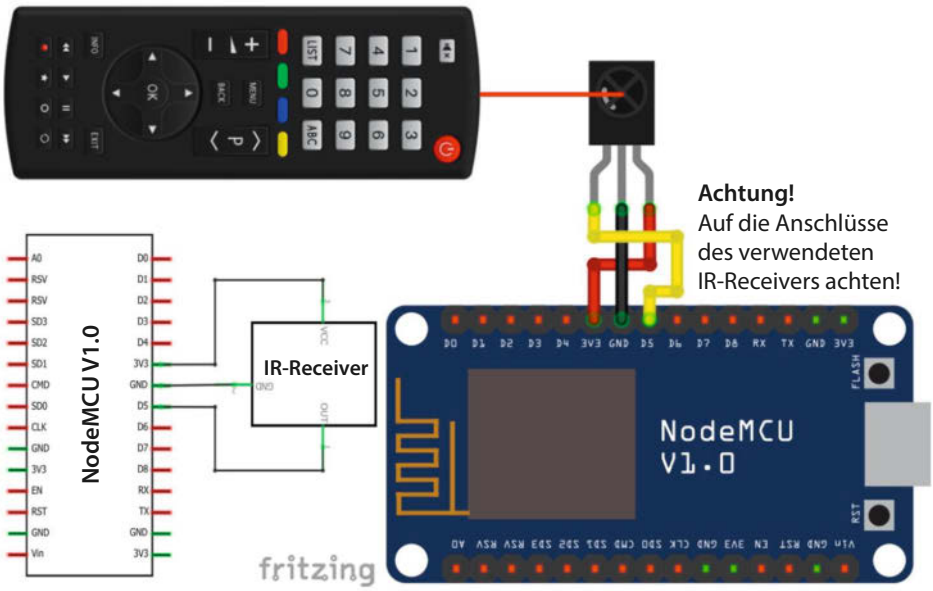
Signale scannen:

» **Infrarot-Empfänger** etwa IR1261-38khz-3-5v
» **Mikrocontroller-Board NodeMCU ESP8266** oder ähnlich

Signale senden:

» **Transistor BC307B**
» **Transistor BC337**
» **2 IR-LED**
» **2 Widerstände 1 kΩ**
» **Mikrocontroller-Board ESP01** mit WLAN
» **USB-Programmer** ESP8266 ESP01 UART Serial Adapter
» **Taster/Druckschalter** für den Programmer (optional)
» **Relais** z.B. 5 V KY-019 (optional)
» **Steckernetzteil 5 V** z.B. ein altes Handy-Ladegerät
» **Spannungswandler Step-Down 5 V zu 3,3 V** z.B. AMS1117
» **Lochrasterplatine**
» **Kabel**

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



Achtung!
Auf die Anschlüsse des verwendeten IR-Receivers achten!

Bild 1: Der IR-Receiver, hier mit NodeMCU als Controllerboard, andere Boards gehen natürlich auch.

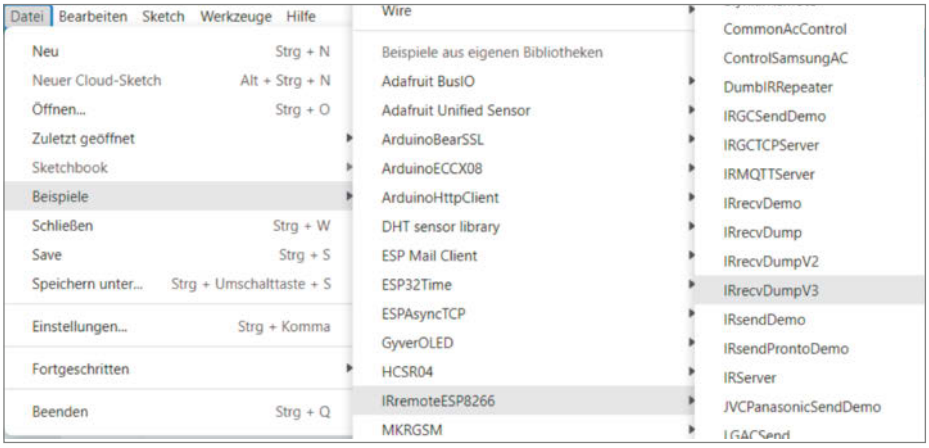


Bild 2: Zum Entschlüsseln der IR-Signale braucht man in der Arduino IDE das Programm IRrecvDumpV3 aus den Beispielen für die Bibliothek IRremoteESP8266.

bewährte Bibliotheken zurück. Wie Bibliotheken generell in die Arduino-IDE eingebunden werden, beschreiben wir online (siehe Link in der Kurzinfo). Benötigt werden die Bibliotheken ESP8266WiFi sowie IRremoteESP8266.

IR-Signal entschlüsseln

Zuerst muss ich die Funktionen der verschiedenen IR-Fernbedienungen entschlüsseln. Dafür verwende ich einen IR-Receiver und einen Mikrocontroller. In meinem Beispiel habe ich mich für ein NodeMCU ESP8266 entschieden. Es sind aber auch die meisten anderen Mikrocontroller für diesen Zweck geeignet. Wichtig ist, dass der Out-Pin des IR-Receivers mit einem Interrupt-fähigen Pin des Mikrocontrollers verbunden wird. Beim NodeMCU können dies D1, D2 und D5 bis D7 sein. Dann braucht es noch GND und 3.3V, die wir vom Mikrocontroller bekommen (Bild 1).

Ist der Aufbau erledigt, beginnt die Entschlüsselung. Dazu rufen wir über die Arduino IDE aus den Beispielen für IRremoteESP8266 das Programm IRrecvDumpV3 auf (Bild 2). Dort muss noch gegebenenfalls der verwendete Out-Pin des selbstgebauten IR-Receivers aktualisiert werden. In meinem Beispiel habe ich D5 verwendet. D5 ist beim NodeMCU GPIO14. Also wird im Programm `kRecvPin=14` eingetragen (Bild 3). Nun wird der Mikrocontroller per Upload mit dem Programm geflasht. Bei vielen NodeMCUs muss zum Flashen der Flashmode eingeschaltet werden: Bei gedrücktem Flash-Taster den Reset-Taster betätigen. Den Flashtaster erst nach Loslassen von Reset ebenfalls wieder loslassen.

Ist dies geschehen, wird der Serial Monitor in der Arduino IDE geöffnet. Jetzt ist darauf zu achten, dass die Baud-Rate wie im Code von IRrecvDumpV3 eingestellt wird (115200 Baud). Anschließend nehme ich eine meiner IR-Fernbedienungen und drücke eine der von mir gewünschten Tasten. Bild 4 zeigt das Ergebnis exemplarisch für die On/Off-Taste einer Panasonic-IR-Fernbedienung. Der Einfachheit halber kopiere ich mir das Ergebnis in eine txt-Datei.

In Bild 4 sind in der Anzeige des seriellen Monitors die Parameter unterstrichen, die ich konkret für diese Panasonic-IR-Fernbedienung später im Programm meiner HandyFB benötige. Jede IR-Fernbedienung sendet mit dem ersten Wert die Adresse des Geräts und dann den verschlüsselten Wert für eine Funktion. Die vollständig versendete Zahlenfolge ist im seriellen Monitor in der Zeile protokolliert, die mit `unit16_t rawData[99]` beginnt. Das ist ganz schön kryptisch, zumal die Zahlen eigentlich gemessene Zeiten darstellen, die wiederum schwanken können. Wer tiefer einsteigen will: Mithilfe eines Oszilloskops ist das gut zu sehen...

Nach demselben Muster wiederhole ich den Prozess für alle mir wichtigen Tasten aller meiner IR-Fernbedienungen.

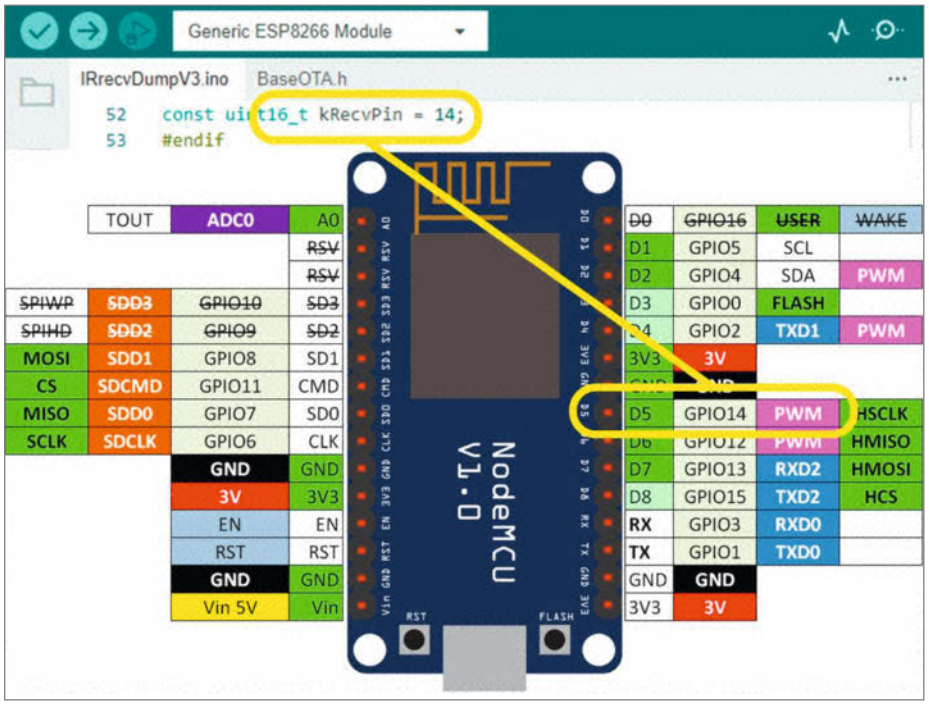


Bild 3: Aus dem Pinout-Diagramm des Boards (hier NodeMCU) lässt sich ablesen, welche interne GPIO-Nummer im Code von IRrecvDumpV3 für den Auslese-Pin eingetragen werden muss.

Schaltung der IR-Fernbedienung

Die gezeigte Schaltung ist das Ergebnis einiger Vorversuche. Beispielsweise hatte ich eine große Auswahl verschiedener IR-LEDs, Transistoren und Widerstände zur Verfügung. Experimente sind nötig, schließlich muss die Reichweite des IR-Signals groß genug sein, um alle Geräte zu erreichen, die im Wohnzimmer an unterschiedlichen Orten stehen. Bei den LEDs kommt es vor allem auf den Öffnungswinkel und die Wellenlänge an: Es gibt IR-LEDs mit Wellenlängen von 840 bis zu 950 nm. Für Fernbedienungen sollten es da die größeren Wellenlängen sein (940 oder 950 nm). Öffnungswinkel gibt es ab 5° bis über 100°. Bei 5° muss man aber schon sehr genau zielen und die Bedienung mehrerer auseinanderstehender Geräte ist damit unmöglich.

Zum Probieren hatte ich mir z.B. zwischenzeitlich eine sehr einfache Schaltung mit einem Arduino Uno aufgebaut. Diese hatte einen Taster und hat per Knopfdruck nur ein einzelnes IR-Signal versendet (konkret, um die Soundbar ein- und auszuschalten. Anstelle des Festwiderstands habe ich ein Potentiometer

eingesetzt. Damit konnte ich den Widerstand so lange verstellen, bis das Signal aus ca. 5 m Entfernung sicher und reproduzierbar geschaltet hat.

Ein elementares Bauteil dieser Schaltung ist der Spannungskonverter. Das Relais braucht 5 V, um schalten zu können. Das ESP01-WLAN-Modul verträgt aber nur maximal 3,3 Volt, höhere Spannung führt zur Beschädigung. Darum geht es nicht ohne Spannungskonverter. Ich bevorzuge den AMS1117, es sind aber auch beliebige andere möglich.

Die vollständige Schaltung ist in Bild 5 zu sehen. Als 5-V-Stromversorgung habe ich ein altes Steckernetzteil genommen.

Jetzt geht es darum, den ESP01 mit einem Programm flashen zu können. Hierfür verwende ich einen klassischen ESP01-Programmer. Es ist wichtig daran zu denken, dass beim ESP01 der GPIO 0 auf GND gezogen werden muss, um ihn anschließend programmieren/flashen zu können. Hierfür habe ich den ESP01-Programmer mit einem kleinen Taster ergänzt (siehe Bild 6). Achtung: Damit das klappt, ist darauf achten, dass man einen Taster nimmt, der im Normalzustand geöffnet ist.

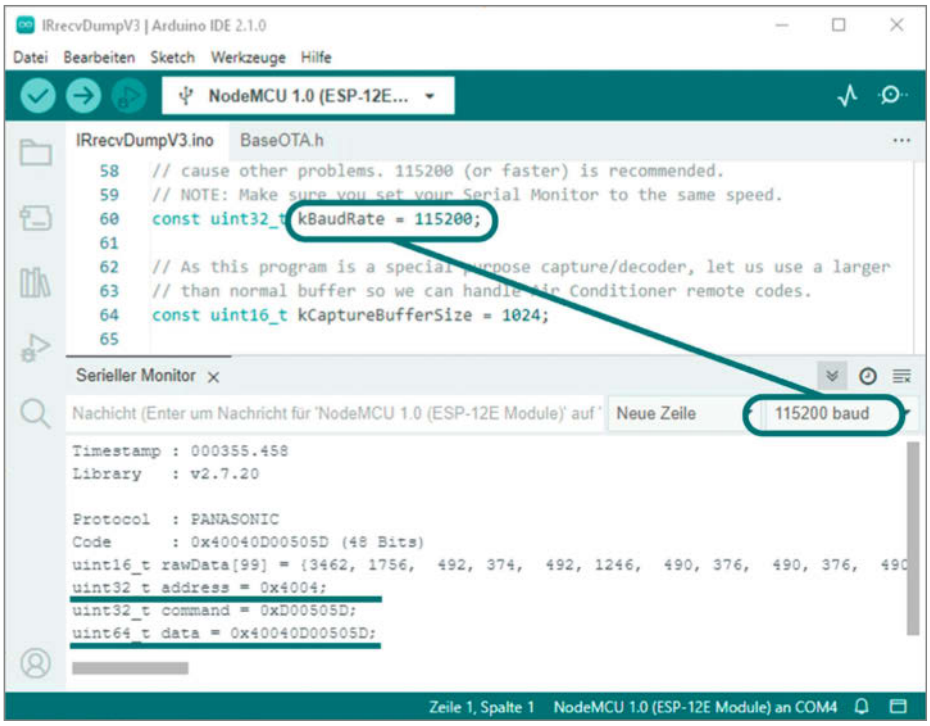


Bild 4: Protokoll im seriellen Monitor nach dem Scannen eines IR-Signals

Programmierung des ESP01

Das Procedere zum Programmieren des Boards ist wie folgt:

- ESP01 im stromlosen Zustand aus der IR-Fernbedienung-Schaltung ziehen und in den ESP01-Programmer stecken.
- Den Taster für GPIO 0 auf GND drücken und unbedingt gedrückt halten, während man den ESP01-Programmer in den USB-Anschluss des PCs steckt. Alternativ vereinfacht ein USB-Hub mit schaltbaren USB-Buchsen die Arbeit und schont die USB-Buchsen am PC vor Verschleiß. Bei so einem USB-Hub muss der Taster für GPIO 0 beim Einschalten der Buchse gedrückt gehalten werden.
- Dann kann man die Taste wieder loslassen.
- Nun ist der ESP01 zum Flashen mit dem Programm bereit. Durch Betätigung des Upload-Buttons in der Arduino-IDE wird das Programm auf das Board geladen.
- Hat der Upload geklappt, wird der ESP01-Programmer aus der USB-Buchse herausgezogen und wieder hineingesteckt, ohne den Taster zu drücken. Beim USB-Hub mit Schalter wird einmal aus- und dann wieder eingeschaltet. So wird der ESP01 bestromt, kann aber nicht mehr geflasht werden. Er ist jetzt im Betriebszustand. Die Prüfung der WLAN-Funktionalität ist möglich.

Programmablauf

Kommen wir zum Programmablauf der IR-Fernbedienung. Der Code ist zu lang, um ihn

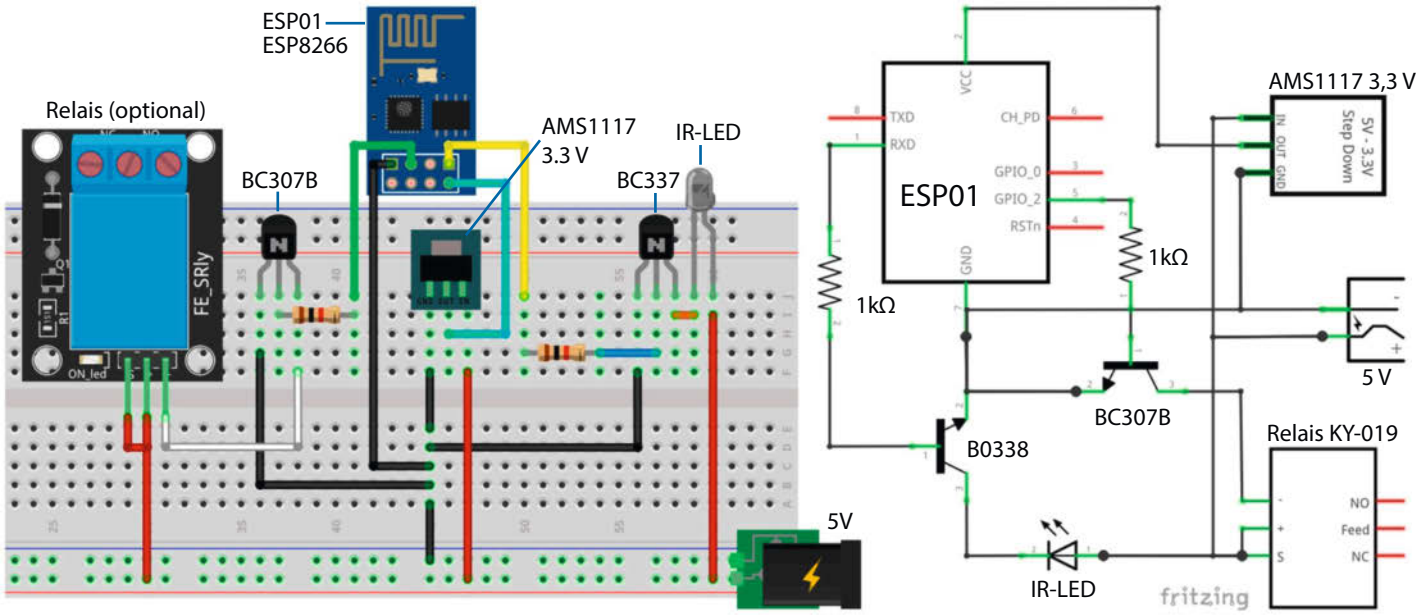


Bild 5: Aufbau und Schaltplan der IR-Control-Schaltung

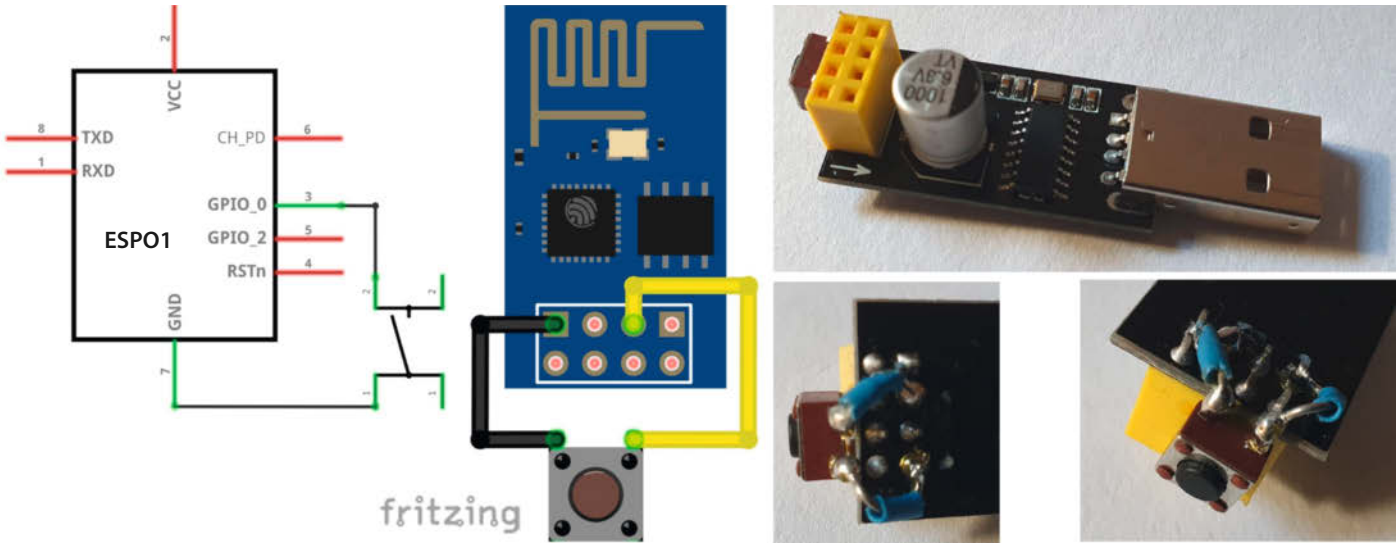


Bild 6: Modifikation des ESP01-Programmers, um D0 zum Programmieren mit GND zu verbinden

hier komplett abzudrucken, aber es gibt ihn über den Link in der Kurzinfo zum Download. Ausführliche Kommentare im Code sollten die Orientierung erleichtern, daher hier nur der Ablauf in groben Zügen:

Nach Anlegen der Variablen, Aufruf der Bibliotheken und allen weiteren Initialisierungen folgen die für Arduino-Code üblichen Abschnitte `setup()` für den Start und dann die Dauerschleife (`loop`).

Im allerersten Programmteil vor `setup()` werden vor allem die für das Versenden der individuellen IR-Befehle notwendigen Werte definiert. Dazu gehören die Adressen, gegebenenfalls Frequenzen und Befehle im Hexadezimalcode der IR-Fernbedienungen.

Im `setup()` startet dann unter anderem der WLAN-Access-Point und die IR-LED wird für das Senden vorbereitet.

Im `loop()` gibt es zwei wesentliche Teile: Der erste Teil erstellt die HTML-Seite, die für den User dann auf dem Handy sichtbar wird und die vom User ausgelöste Funktionen erkennt. Der größte Teil des Programms formatiert das Erscheinungsbild auf dem Handy-Display.

Im zweiten Programm-Teil werden entsprechend der vom Handy empfangenen Funktionen die Befehle zum Versenden der IR-Signale ausgeführt. Die dafür erforderlichen Werte wurden im ersten Programmteil schon definiert. Im Wesentlichen werden einfache `if-else`-Abfragen umgesetzt.

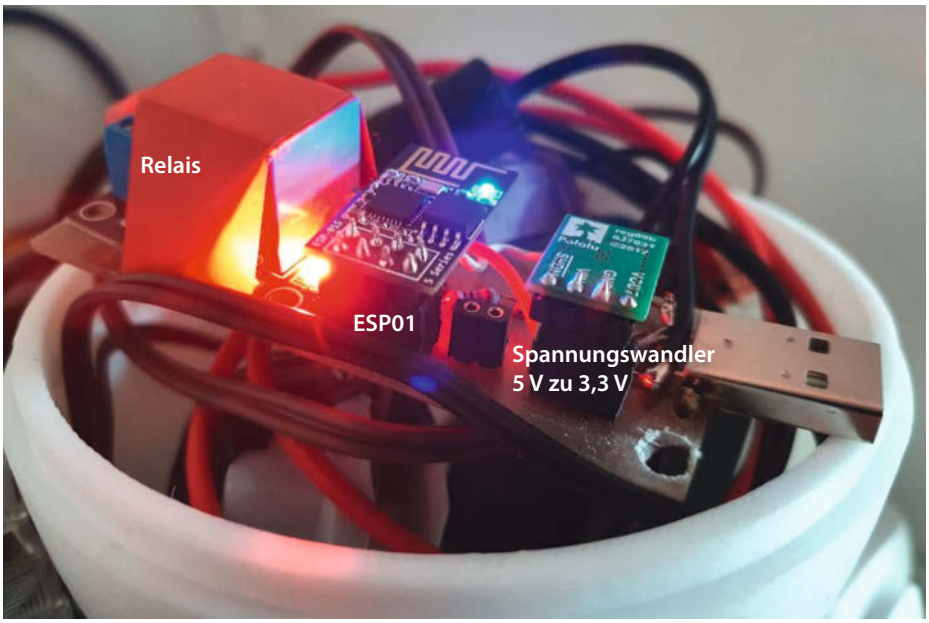


Bild 7: Der Aufbau meiner Universalfernbedienung auf einem Stück Lochrasterplatine

Konkreter Aufbau

Die Platine von meiner IR-Fernbedienung ist schließlich 8 × 3 cm groß geworden (Bild 7). Die gesamte Kabelei ist in einem überdimensional großen Überraschungsei untergebracht, das ich noch in meinem Lager fand. Natürlich ist auch jedes andere Gehäuse denkbar. Weil die farbigen LEDs auf der Platine (rot und blau) sehr hell leuchten, ist jederzeit zu erkennen, ob die IR-Fernbedienung betriebsbereit ist (Bild 8).

Bei mir sind alle Geräte an einer schaltbaren Steckdosenleiste angeschlossen, so auch die IR-Fernbedienung. Wenn ich also mal einen Film sehen will, wird alles eingeschaltet und danach wieder ausgeschaltet. Die farbigen LEDs sorgen jetzt zusätzlich dafür, dass ich es nicht vergesse.

Aus dem Überraschungsei kommen ein 230-V-Anschluss, eine Steckdose und eine lange Litze mit den IR-LEDs. Mit der Steckdose wird die kleine Lampe versorgt, die mit dem Relais auf der Platine geschaltet werden kann.

In meinem Aufbau und bei meiner Anordnung der Geräte im Wohnzimmer komme ich mit zwei IR-LEDs aus. Eine ist auf den Soundbar ausgerichtet, die andere auf Beamer und Blu-Ray-Player. Das Drahtgestell (Bild 9) ist erforderlich, um die LED weit genug vor den Beamer zu hängen. Der Beamer wird nach Benutzung in den Schrank geschwenkt. Eine Schiebetür lässt ihn dann ganz verschwinden.

Inbetriebnahme am Handy

Inzwischen ist alles so weit vorbereitet, dass die Inbetriebnahme mit dem Handy erfolgen kann. Das ESP01-Modul ist als Access Point konfiguriert. Es wird also als eigener Webserver betrieben. Wenn ich mich mit diesem über

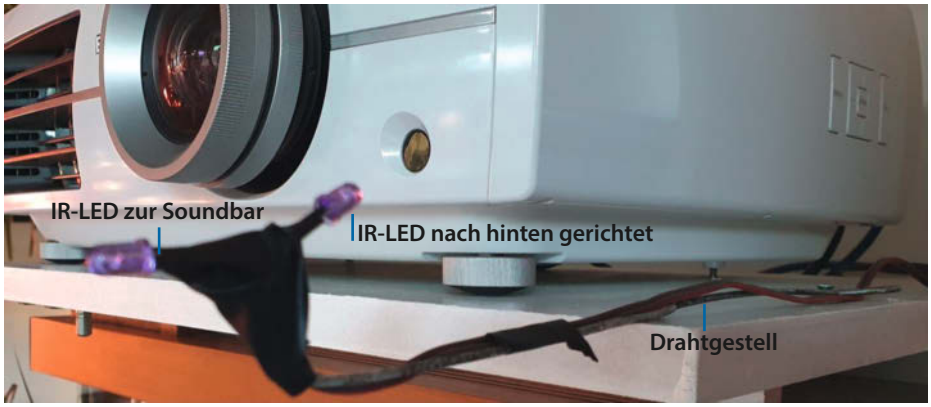


Bild 9: Zwei passgenau ausgerichtete Infrarot-LEDs schicken Signale an den Beamer und an die Soundbar.

Bild 8: Die LEDs auf der Platine im Inneren schimmern durch die Hülle des Eis, in das ich meine Fernbedienung eingebaut habe.

mein Handy verbinde, fungiert das Handy als Client des vom ESP01-Modul errichteten Netzwerks und erhält dafür eine IP-Adresse. Wenn ich dann die IP-Adresse des Webservers im Browser des Handys aufrufe, wird die HTML-Seite zur Steuerung der Geräte auf dem Display meines Handys sichtbar.

Als erstes schalten wir (falls noch nicht aktiv) das WLAN am Handy ein. Die IR-Fernbedienung mit der SSID „HandyFB" wird als WLAN erkannt und in der Übersicht verfügbarer WLAN-Netzwerke angezeigt (Bild 10). Wir klicken es an und werden dann nach dem Passwort gefragt. Hier ist das Passwort einzugeben, wie es im Programm definiert wurde.

Das Handy verbindet sich dem Webserver. Das kann ein paar Sekunden dauern. Wenn die Verbindung hergestellt ist, gehen wir auf die Einstellungen der Verbindung. Dann wird neben anderen Parametern auch unsere IP-Adresse in diesem Netzwerk angezeigt. In Bild 10 endet die IP-Adresse auf 2. Das Handy ist also das zweite Gerät im Netzwerk.

Jetzt öffnen wir den Browser unseres Handys. In der Adresszeile wird die IP-Adresse des Webservers eingetragen. Die ist mit der IP-Adresse, die das Handy zugewiesen bekommen hat, weitgehend identisch – nur dass sie auf „1" enden muss. Diese IP-Adresse des Webservers bleibt immer die gleiche. Wir können sie also als Lesezeichen im Browser speichern. Sollten sich mehrere Handys am Webserver anmelden, erhalten diese die verschiedensten Nummern. Das spielt aber für den Webserver selbst keine Rolle. Wenn alles funktioniert hat, sollte es so aussehen wie in Bild 10 ganz rechts.

Jetzt lassen sich bequem über Tippen auf die Schaltflächen im Handy-Browser alle Geräte zentral steuern. Licht aus, Film ab! —*pek*

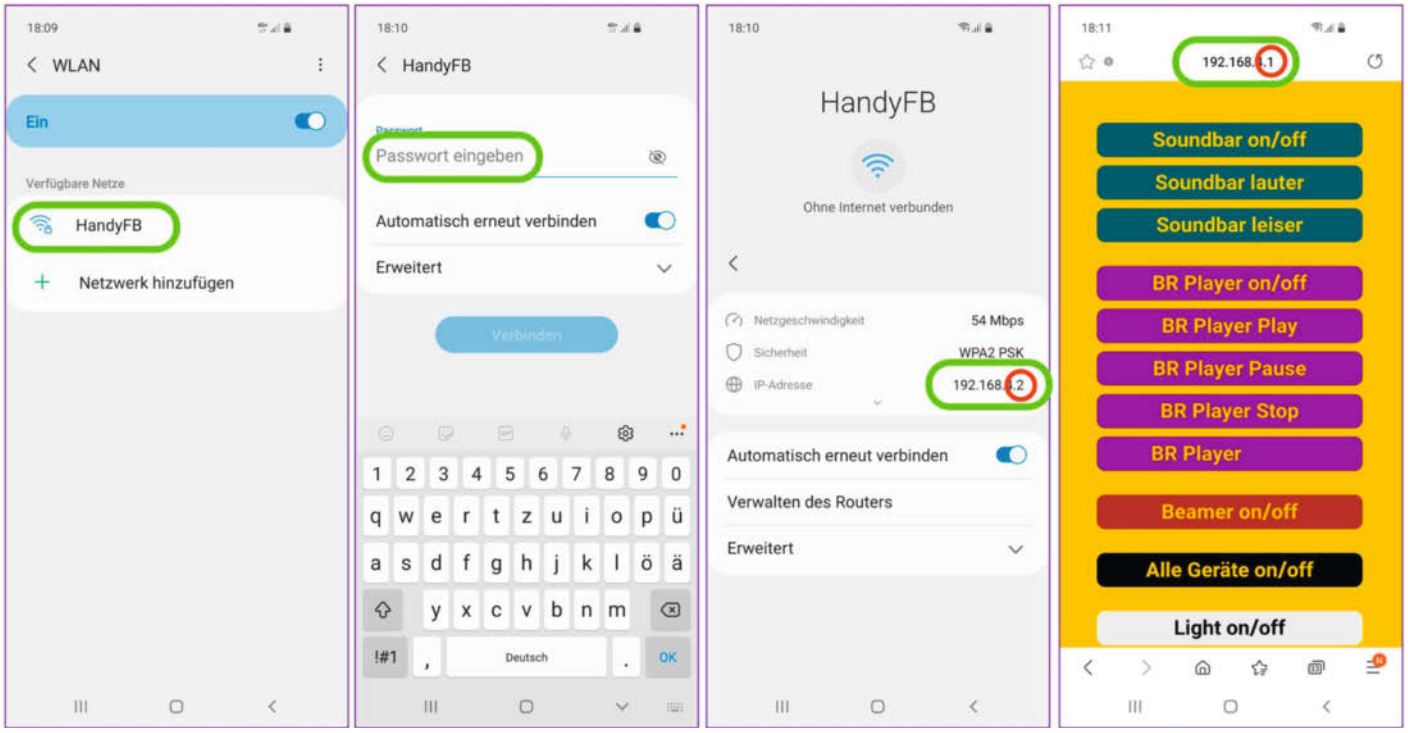


Bild 10: So kommt die Webseite mit der zentralen Bedienoberfläche in den Handy-Browser.

Photon

Ein selbstgebauter Open-Source-Belichtungsmesser

Analoge Fotografie erlebt eine Renaissance. Falls die Vintage-Kamera aus dem eigenen Bestand oder vom Flohmarkt keinen eingebauten Belichtungsmesser hat, wird die Wahl von Blende und Verschlusszeit aber zum Glücksspiel. Es sei denn, man benutzt einen externen Belichtungsmesser, den man sich auch selber bauen kann. So geht's.

von Martin Spendiff

Hier ist ein Raspberry-Pi-Pico-Projekt, das Ihnen ein paar Tränen ersparen kann, wenn Ihre analogen Filme nach dem Entwickeln aus dem Labor zurückkommen: Wir haben einen Open-Source-Belichtungsmesser namens Photon entwickelt.

Ein solcher Belichtungsmesser ist beim Fotografieren ein unverzichtbares Werkzeug, besonders bei alten Kameras, die keinen eingebauten Belichtungsmesser haben. Moderne Kameras wenden eine Menge Rechenarbeit auf, um zu ermitteln, wieviel Licht auf das Motiv fällt. Wenn man sich aber direkt dem Motiv zuwendet und auf dem Belichtungsmesser ablesen kann, welche Belichtung bei welcher Blende und welchem ISO-Wert angemessen ist, ist kein Raten mehr nötig und alles wird einfacher!

Zur Übersicht haben wir ein (englisches) Video produziert, dass nicht nur die Bedienung von Photon zeigt, auch mit dessen Hilfe geschossene Beispielfotos präsentiert und einen Vergleich mit kommerziellen Geräten bietet (siehe Link in der Kurzinfo).

Photon reproduziert (derzeit) einige der Funktionen von teureren kommerziellen Geräten (Bild 1) unter Verwendung einiger preiswerter, leicht erhältlicher Teile. Er misst die Helligkeit des Umgebungslichts sowie die Rot-, Grün- und Blauanteile des Lichts, was in zukünftigen Versionen auch noch einen Weißabgleich ermöglichen könnte – Photon ist ein Open-Source-Projekt in Entwicklung und die Möglichkeiten der Hardware reizt die Software derzeit noch nicht aus. Aber auch schon jetzt gilt: Sie können für einen neuen Belichtungsmesser entweder ab gut 100 (aber auch über 1000) Euro ausgeben – oder sich für aktuell rund 60 Euro selbst einen löten. Das ist gar nicht schwer.

Kurzinfo

- **Praxistauglicher Belichtungsmesser für analoge Fotos aus wenigen Komponenten zusammenbauen**
- **Open-Source-Projekt, eigene Erweiterungen und Updates aus der Community möglich**
- **Blitzmodus ist in Arbeit**

Checkliste

Zeitaufwand:
ein Wochenende

Kosten:
60 Euro (Stand Januar 2024)

Programmieren:
Micro-Python-Code auf Raspberry Pi Pico spielen

3D-Druck:
für das Gehäuse, optional

Alles zum Artikel im Web unter
make-magazin.de/xm1g

Material

- **Raspberry Pi Pico** Standard-Board oder Pimoroni Pico LiPo
- **LiPo SHIM for Pico** von Pimoroni (PIM557) oder ähnliches Breakout-Board für den Akkuanschluss mit Laderegler (nicht nötig, wenn als Board ein Pico LiPo verwendet wird)
- **Farb-OLED-Bildschirm** 128 × 128 Pixel, etwa Waveshare-1,5-Zoll-Modul
- **Drehgeber** zum Anpassen der Einstellungen und Ändern des Prioritätsmodus; Iduino SE055 oder ähnlich
- **Tastaturschalter** für die Lichtmessung; je nach Geschmack linear, taktil oder clicky
- **Mikrotaster**, 6 × 6 mm für den ISO-Modus
- **BH1745 Helligkeits- und Farbsensor** als Breakout-Platine von Pimoroni (PIM375)
- **LiPo/Li-Ionen-Akku** 3,7 V, 1100 mAh, mit zum Board passenden JST-Stecker
- **Kabel** zur Verbindung der Komponenten
- **Gehäuse** aus dem 3D-Drucker (optional, Download siehe Link)

Hardware

Falls Sie einen Standard-Pico und eine Laderegler-Platine wie Pimoronis LiPo SHIM verwenden, verbinden Sie den Laderegler mit dem Pico, wie in der Anleitung dazu beschrieben. Bequemer geht es mit dem Pico LiPo von Pimoroni als Platine, denn der hat den Laderegler gleich mit an Bord. Schließen Sie dann den Akku an. Kontrollieren Sie zuvor unbedingt, ob die Polung des Akkusteckers mit der des Boards übereinstimmt.

Schließen Sie dann den Lichtsensor, das OLED-Display, den Drehgeber und die beiden Schalter an die GPIO-Pins des Pico an (Bild 2 und Tabelle). Die Versorgungsspannung (mit VCC, + oder 3.6V beschriftet) und GND aller Komponenten sind ebenfalls mit dem Pico verbunden, in der Tabelle aber weggelassen. Auf dem Bild sitzen die Teile bereits im Gehäuse aus dem 3D-Drucker (herunterzuladen über den Link in der Kurzinfo). Die Technik lässt sich aber problemlos auch in jedes andere passende Gehäuse einbauen.

Software

Für die Pico-Firmware laden Sie bitte das aktuelle UF2-Image aus dem Github-Repository von Pimoroni herunter (Link in der Kurzinfo) und installieren es auf dem Pico. Sie benötigen es, um die Pimoroni-Treiber für den Lichtsensor zu verwenden.

Wir haben den Photon-Code in Micro-Python geschrieben, der auf einfacher Mathematik basiert, und die vom Lichtsensor zurückgegebene Beleuchtungsstärke in einen Belichtungswert umwandelt (Code zum Download siehe Link in der Kurzinfo). Eine etwas ausführlichere Darstellung der Mathematik dahinter steht am Ende der Readme-Datei in Github.

Bild 1: Der selbstgebaute Belichtungsmesser Photon braucht den Vergleich mit käuflichen Geräten nicht zu scheuen.

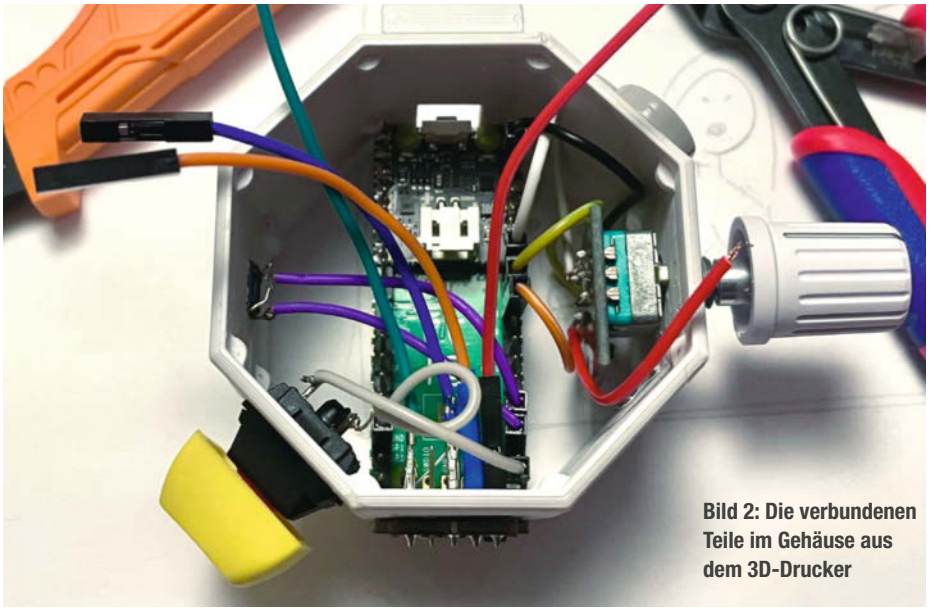


Bild 2: Die verbundenen Teile im Gehäuse aus dem 3D-Drucker

Kopieren Sie den Code von Github auf Ihren Rechner – entweder als Zip-Archiv, wie in unserer Online-Anleitung zu GitHub beschrieben, oder per Kommandozeile, falls Sie Git auf Ihrem Rechner installiert haben:

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

Ermitteln Sie dann den Port, über den der Pico vom Rechner aus erreichbar ist:

```
python -m serial.tools.list_ports
```

Mit dem ermittelten Pfad (bei mir zum Beispiel unter Linux `/dev/ttyACM0`, unter Windows etwa `COM2`) kopiert man den Code mit ampy (Adafruits Kommandozeilen-Tool für Kommunikation mit MicroPython-Boards über eine serielle Verbindung, siehe Link in der Kurzinfo) und den `put`-Befehlen auf den Pico:

```
ampy -p /dev/ttyACM0 put drivers/
ampy -p /dev/ttyACM0 put gui/
ampy -p /dev/ttyACM0 put color_
        setup.py
ampy -p /dev/ttyACM0 put main.py
```

Geschafft! Jetzt wird das Python-Skript automatisch gestartet, sobald der Pico mit Strom versorgt wird.

Anschlüsse

Pico-Pin-Bezeichnung	Pico-Pin-Nummer	Sensor/Modul-Anschluss
Lichtsensor BH1745		
GP4	6	SDA
GP5	7	SCL
OLED-Display		
GP19	25	DIN/MOSI
GP18	24	CLK/SCK
GP17	22	CS
GP20	26	DC
GP21	27	RST
Drehgeber		
GP6	9	CLK
GP7	10	DT
GP8	11	SW
Lichtmessungs-Schalter		
GP15	20	der eine
GND	z.B. 18	der andere
ISO-Taster		
GP22	29	der eine
GND	z.B. 28	der andere

Bild 3: Noch in Entwicklung: der Blitz-taugliche Sensor für Photon

Gehäuse

Sie können unsere STL-Dateien aus dem Verzeichnis„cases“ aus dem Github-Repository in 3D drucken. Es gibt dort verschiedene Varianten zur Auswahl. Wer keinen 3D-Drucker hat, kann seinen Belichtungsmesser aber auch in jedes handliche, aber genügend große fertige Gehäuse einbauen, da ist dann die eigene Kreativität gefragt.

Einsatz und Ausblick

Drücken Sie die Taste auf dem SHIM-Board oder dem Pico LiPo zum Einschalten. Drehen Sie den Drehgeber, um Blende oder Verschlusszeit vorzuwählen. Was Priorität hat, wechselt bei Druck auf den im Drehgeber integrierten Druchschalter. Drücken Sie die große Taste, um die Lichtmessung vorzunehmen und auf dem Display erscheint auf Grundlage der Messung eine passende Kombination aus Blende und Verschlusszeit. Um die vorgegebene Filmempfindlichkeit zu verändern, drücken Sie den kleinen Taster einmal, stellen den gewünschten ISO-Wert mit dem Drehgeber ein und betätigen den kleinen Taster zum Bestätigen nochmals. Das alles ist im verlinkten Video auch noch mal zu sehen.

Photon ist Work in Progress und lebt vom Mitmachen. Wenn Sie Photon noch besser machen können, forken Sie bitte das Github-Repository und verwenden Sie einen Feature-Zweig. Die Voraussetzungen für Weißabgleich-Messungen sind da, werden nur noch nicht ausgewertet. Die am häufigsten geforderte Verbesserung ist aber ein Blitz-/Stroboskopmodus – da arbeiten wir gerade dran. Leider erwies sich der hier beschriebene BH1745-Sensor als zu träge, um Blitzlicht zu erfassen. Deshalb sind wir dabei, von Grund auf einen neuen Sensor zu entwickeln und rechnen mit einem Blitz-tauglichen Update für Photon bis Mitte 2024 (Bild 3).

Wer oft blitzt, sollte mit seinem Photon-Nachbau also noch etwas warten; wir hoffen aber, schon jetzt den Anstoß dazu gegeben zu haben, Photon zu einer offenen Community-Ressource zu machen. *—pek*

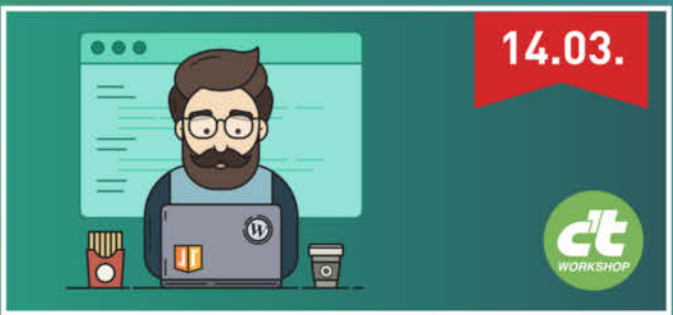
WIR TEILEN KEIN HALBWISSEN
WIR SCHAFFEN FACHWISSEN

KI für den Unternehmenseinsatz – vertraulich und sicher

Das Webinar stellt verschiedene Konzepte vor, KI im Unternehmen zu nutzen und vergleicht diese in Hinblick auf Technik, Kosten und Datenschutz.

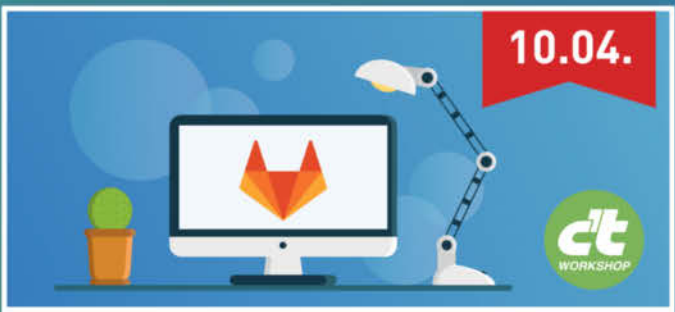
Datenschutz in Arztpraxen

Das Webinar beleuchtet in drei Sitzungen die wichtigsten Themen aus dem Telematik- und Datenschutz-Alltag einer Arztpraxis und gibt konkrete, praktische Tipps.



WordPress für Einsteiger

Der praxisorientierte Workshop richtet sich an Neu- und Quereinsteiger in WordPress und bietet eine grundlegende und fundierte Einarbeitung in die aktuelle Version des populären CMS.



Einführung in GitLab

Der Workshop bietet einen Einstieg in den Betrieb einer eigenen Gitlab-Instanz. Sie lernen GitLab initial aufzusetzen, sowie Ihre Instanz zu konfigurieren und an eigene Anforderungen anzupassen.

CI/CD mit GitLab

Der zweitägige Workshop bietet eine praktische Einführung in die GitLab-CI-Tools und zeigt, wie man damit Softwareprojekte baut, testet und veröffentlicht.

Kluge Strukturen für Microsoft 365 entwickeln

Lernen Sie in dem Workshop, wie Sie gemeinsam mit Ihrem Team Leitlinien entwickeln, um in Zukunft das volle Potenzial für die Zusammenarbeit auszuschöpfen.

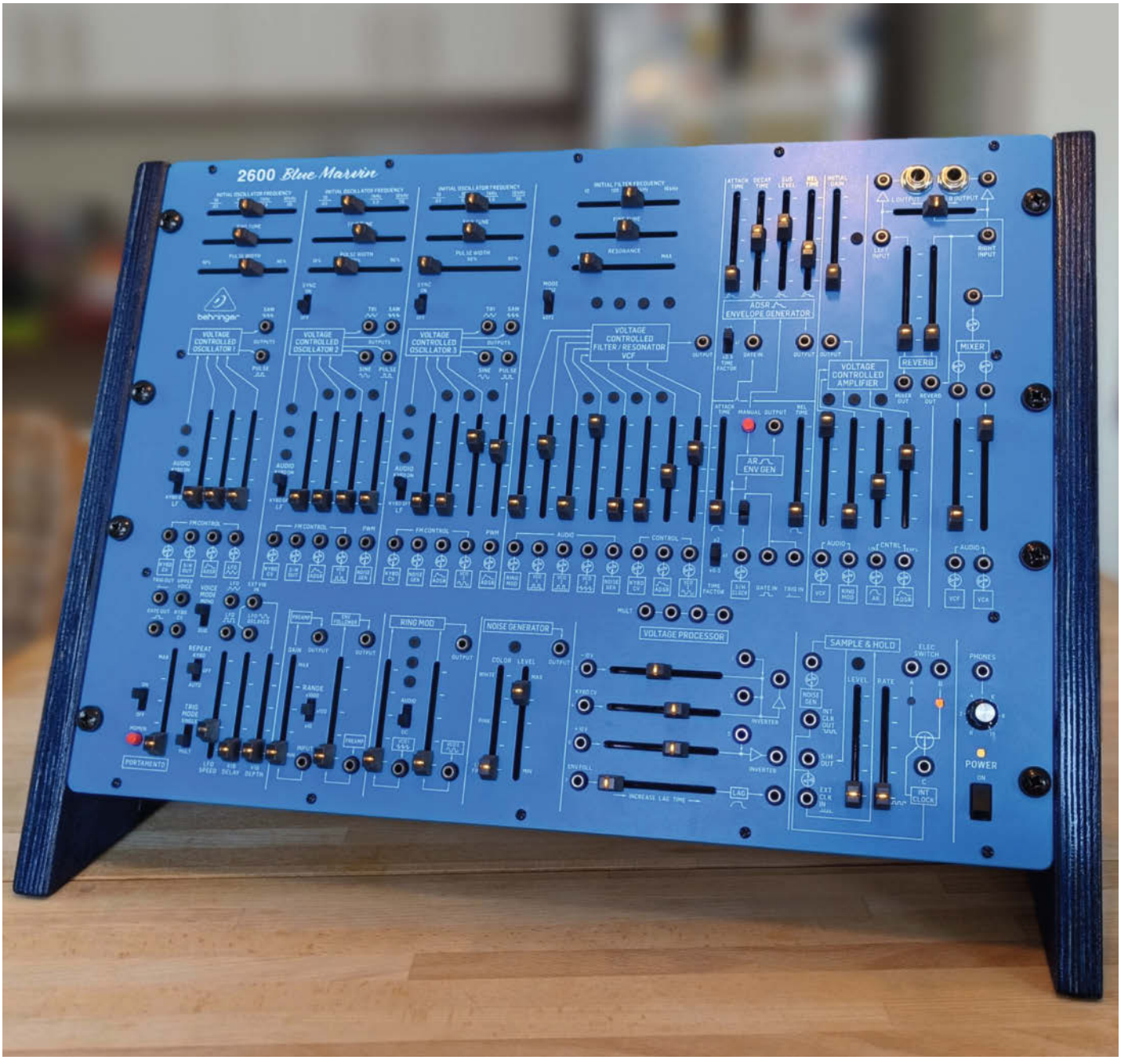
Sichern Sie sich Ihren Frühbucher-Rabatt:

heise.de/ct/Events

19"-Rack-Ständer für den Schreibtisch

Neues „Gear“ zieht ein und braucht einen Ständer. Also schnell etwas ohne Elektrowerkzeuge aus Altholz gebaut und per 3D-Druck funktionsfähig gemacht. Und ganz hässlich ist es am Ende auch nicht.

von Carsten Wartmann



Der Behringer 2600 ist ein moderner und recht preiswerter Nachbau des Synthesizer-Klassikers ARP 2600, mit dem unzählige Hits und Soundeffekte erzeugt wurden, allen voran die Sounds von R2-D2. Als dieser Synthesizer für einen guten Preis in den lokalen Kleinanzeigen auftauchte, überlegte ich nicht lange. Nur passt er mit seinem 19-Zoll-Rackformat nicht zu meinen Desktop-Synths und nimmt auf dem Tisch liegend viel Platz weg. Metallgestelle und schicke Holzständer gibt es zwar zu kaufen, aber mit etwas Altholz und Muskelschmalz sollte das auch im Selbstbau für wenig Geld möglich sein?

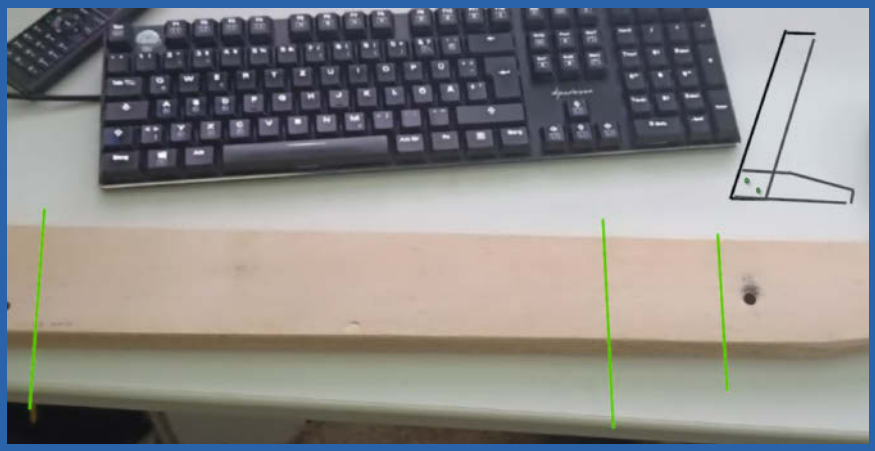
Von einem alten Sofa, das 15 Jahre und zwei Kinder überlebt hatte, bis es ersetzt wurde, waren noch zwei massive Multiplexstreben übrig. Das Holz ist weder schön noch makellos, aber ich wollte die Macken als Erinnerung behalten. Bei einer Videokonferenz mit meinem Kollegen und Holzversteher Johannes bekam ich wie immer gute Tipps. Schnell war eine erste Skizze angefertigt und besprochen. Mit zwei oder drei Sägeschnitten sollte es zu schaffen sein. Zuerst wollte ich die Teile mit Versatz verschrauben, aber Johannes machte mir Mut, dass ich das auch mit Leim und einer Schraube schräg von unten versenkt hinbekomme.

Entsprechend der gewünschten Neigung des Ständers habe ich den dort gemessenen Winkel auf die Latten übertragen und konnte mir so einen weiteren Schnitt sparen. Meine gute Japansäge geht durch Multiplex wie durch Butter und mittlerweile kann ich auch einigermaßen gerade sägen, sodass dies schnell erledigt war.

Die Verbindung mit der versenkten Schraube habe ich noch mit einem Stahlstift (abgeknipster Nagel) gegen Verdrehen gesichert, nicht so sehr für den endgültigen Halt, sondern weil ich meinen Zwingen nicht so ganz traue. Nachdem der Leim fest war, wurde geschliffen und nach sorgfältigen Entstauben mit blauer Beize (Clou) gefärbt. Danach kamen noch ein paar Lagen Klarlack aus der Dose drüber, was einen leichten Glanz beschert und die Beize fixiert.

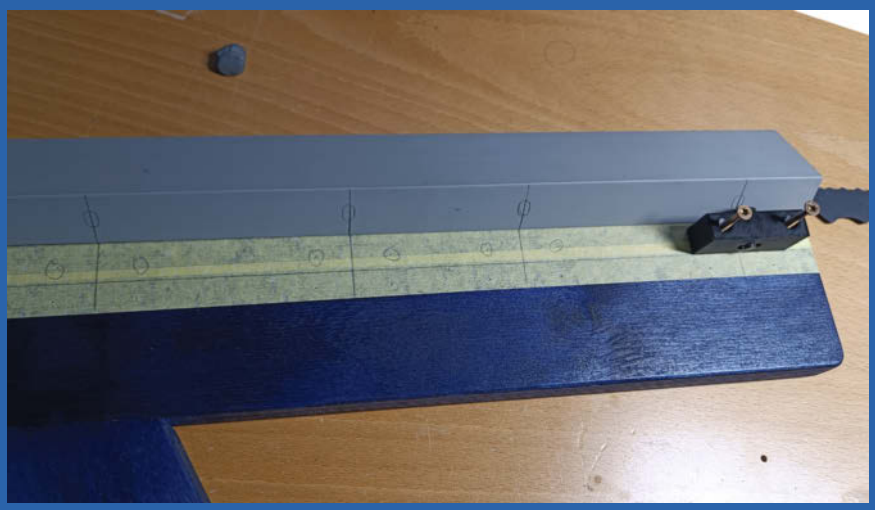
Die Winkel zur Befestigung des Synthesizers mit den Rackschrauben habe ich in Blender entworfen und in PETG ausgedruckt. Die Abstände der Befestigungslöcher am Gerät waren schnell mit etwas Klebeband übertragen und stimmten beim zweiten Versuch fast sofort. Noch ein paar Silikonfüße drunter und das Ganze steht sicher und rutschfest. Die nach hinten ragenden Streben könnten etwas kürzer sein, aber ich werde sie wohl nicht mehr kürzen, irgendwie möchte ich noch ein Rack bauen, um noch eine Reihe Eurorack-Module darüber zu montieren … —*caw*

▶ printables.com/model/725587



Diese Skizze entstand im Videochat.

Die Version links wurde es dann. Im Rückblick sind die Füße daher etwas zu lang geraten.



Die Maße der Montagelöcher wurden mit dem Kunststoffwinkel abgenommen und auf die Seitenteile übertragen, um die gedruckten Montagewinkel richtig zu platzieren.

Bombenspaß mit Arduino

Mit 12 Arduinos wird eine aus 11 verschiedenen Minispielen bestehende „Bombe“ im gemütlichen Wohnzimmer realisiert. Diese gilt es zu entschärfen.

von Daniel Schwabe

Heath Paddock

Basierend auf dem Spielkonzept des Computerspiels „Keep Talking and Nobody Exploades“ muss ein zweigeteiltes Team asymmetrisch einen DIY-Bombenkoffer innerhalb von 5 Minuten entschärfen.

Der eine Teil des Teams muss direkt Hand anlegen und die Anweisungen der anderen Team-Hälfte umsetzen, die das Benutzerhandbuch der Bombe vor sich hat.

Dazu müssen 11 verschiedene Minispiele, basierend auf 11 Arduinos gemeistert werden, um die Bombe zu entschärfen. So muss an einem Modul ein leuchtender Punkt mit einem Steuerkreuz durch ein unsichtbares Labyrinth geleitet werden. Durch zwei weitere grüne LEDs kann das Handbuch-Team herausfinden, wie das Labyrinth aussieht, und das Entschärfungs-Team dann navigieren.

Wer hier nicht bei der Kommunikation einen kühlen Kopf bewahren kann, dem fliegt das Wohnzimmer um die Ohren. Ein Display zählt in roten Lettern die Zeit herunter, während zusätzlich noch das Ticken der Uhr zu hören ist. Drei Leben haben die Spieler zur Verfügung. Je mehr Fehler man macht, desto schneller zählt der Countdown herunter.

Technisch verfügt jedes der 11 Minispiel-Module über einen eigenen Arduino Pro oder Arduino Pro Micro. Dazu kommt ein Arduino Mega 1280, der die Spielüberwachung übernimmt.

Die einzelnen Module sind, zusammen mit dem Arduino Mega, in einer Schleife geschaltet. Ist ein Modul korrekt„entschärft“ worden, oder haben die Spieler einen Fehler gemacht, sendet das betreffende Modul eine Nachricht an sein Nachbarmodul, das die Nachricht wiederum weiterschickt, bis die Übertragung wieder am Anfang ankommt.

Die mit einem selbstgeschriebenen Protokoll über Hardware Serial verschickte Nachricht wird dann von den Teilen der Bombe, die sie betreffen, ausgelesen und umgesetzt. Beispielsweise um eines der drei Leben bei einem Fehler abzuziehen.

Diese Elektronik verschwindet im Koffer fast komplett unter extra 3D-gedruckten Abdeckungen, in denen die Bedienelemente der Module, wie beispielsweise Knöpfe oder Displays, eingelassen sind. Die 3D-Dateien dafür sowie der Code und weitere Informationen, findet man im GitHub-Repository zu dem Projekt von Heath Paddock. Diese Abdeckungen können ohne Aufwand entfernt werden, um ein Modul aus dem Spiel herauszunehmen. Für die Zukunft gibt es hier auf jeden Fall Erweiterungspotenzial.

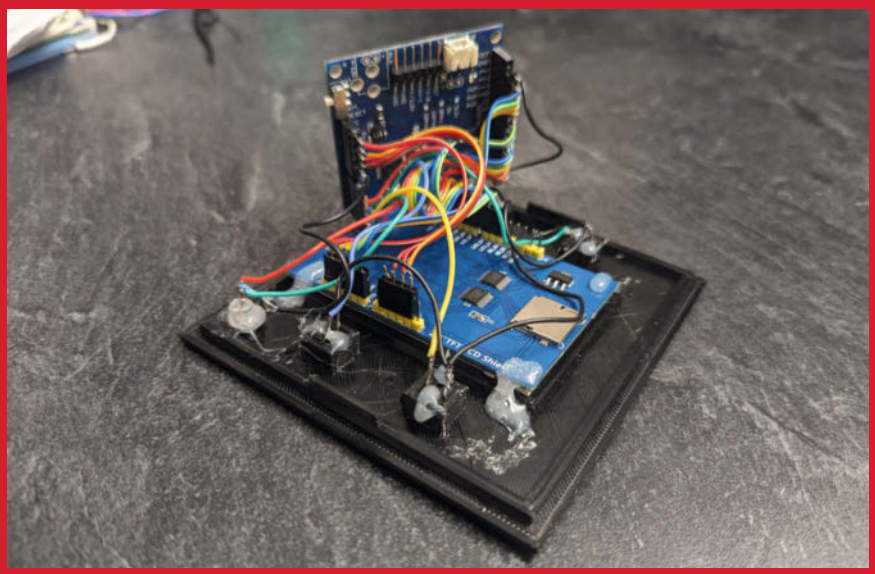
Auf Heath Paddocks YouTube-Channel befindet sich außerdem ein Hands-On zu dem Koffer mit einem Einblick in eine Spielsession.

Unter folgendem Link kommt man auf die Projektseite, auf der detailliert aufgeschlüsselt ist, wie die Spielmodule funktionieren und wie technisch vorgegangen wurde. —*das*

▶ **https://heathbar.github.io/keep-talking/**

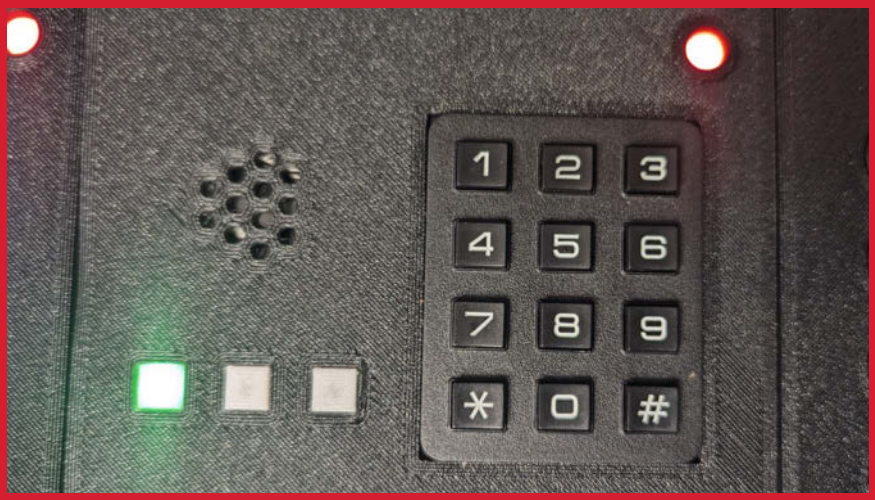
Heath Paddock

Was man hier wohl machen muss? Ohne Anleitung kommt man nicht weiter.



Heath Paddock

Das Innenleben eines der Module.



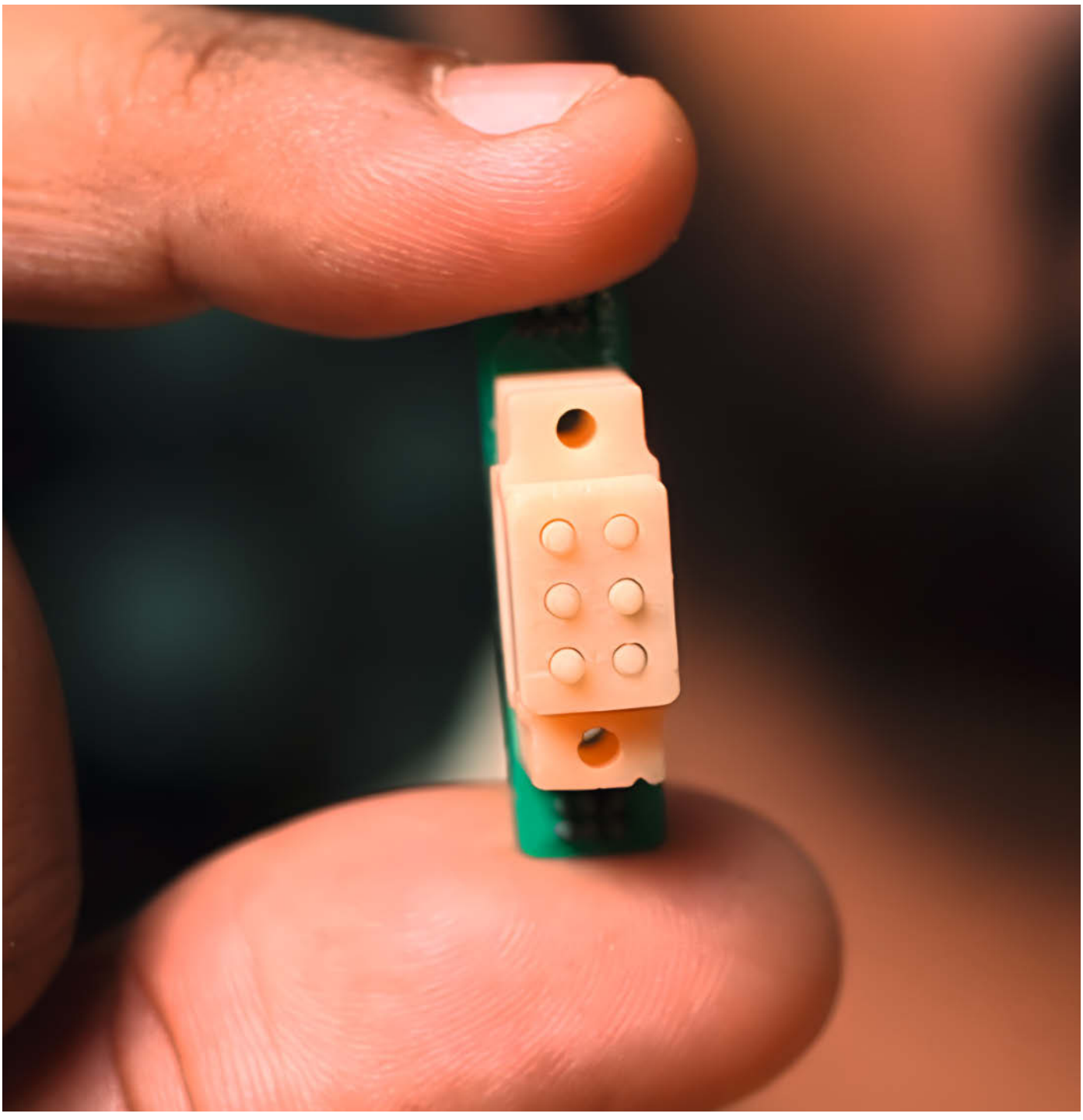
Heath Paddock

Die Minispiele unterscheiden sich alle voneinander.

DIY-Braille-Modul

Blinde und sehbehinderte Menschen können mithilfe einer Braillezeile fühlen, was auf einem Computerbildschirm passiert. Solche Geräte sind aber recht teuer und nicht für jeden zugänglich. Dieser ausgezeichnete Open-Source-Entwurf könnte vielen Menschen helfen.

von Ákos Fodor



Bilder: Vijay Raghav Varada

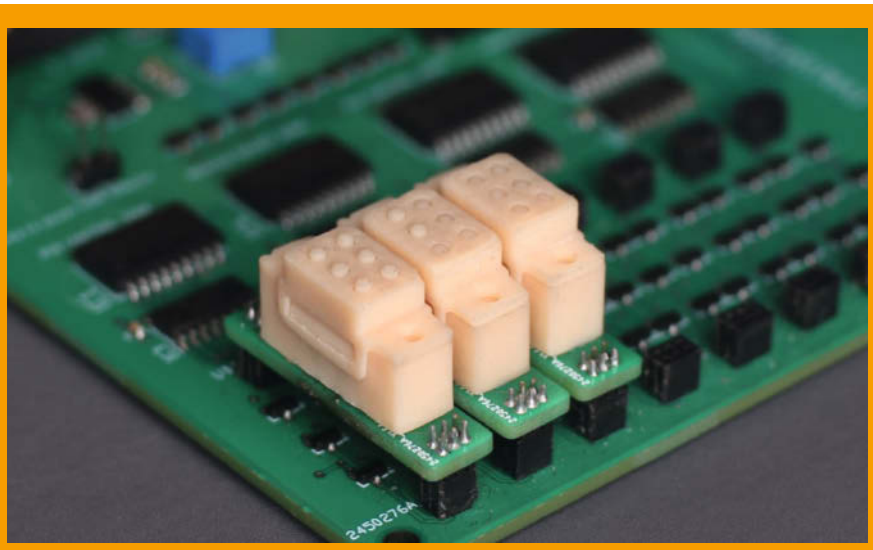
Der indische Ingenieur und Maker Vijay Raghav Varada hat mit einem selbst entwickelten Braille-Modul den Hackaday Prize 2023 gewonnen. Das kleine Gerät verfügt über bewegliche Stifte, mit deren Hilfe sich die unterschiedlichen Zeichen der Brailleschrift abbilden lassen. Für gewöhnlich findet man mehrere dieser Module in sogenannten Braillezeilen, einer Art haptischem Display. Diese unterstützen blinde und sehbehinderte Menschen dabei, mit Computern, Tablets und Smartphones zu interagieren, indem sie Bildschirminhalte dynamisch in Computerbraille wiedergeben.

Varadas Entwurf ist nicht nur kompakter als kommerzielle Modelle, die für die Bewegung ausladende Piezo-Biegewandler verwenden, sondern auch günstiger und Open Source. Dafür kann das Modul mit 6-Punkt-Braille bestimmte Sonderzeichen nicht abbilden. Die meisten Komponenten stammen aus dem 3D-Drucker und sind aus Kunstharz (Resin) gefertigt. Während handelsübliche Braillezeilen aufgrund der komplexen Bauweise ihrer Module meist mehrere Tausend Euro kosten, ist es Varada gelungen, die Materialkosten eines einzelnen Moduls auf unter einen US-Dollar zu senken. Dadurch erhalten viele Menschen einen bezahlbaren Zugang zu dieser Technik, die es sich sonst nicht leisten könnten – z.B. weil ihre Krankenkasse die Kosten nicht übernimmt oder sie gar nicht versichert sind.

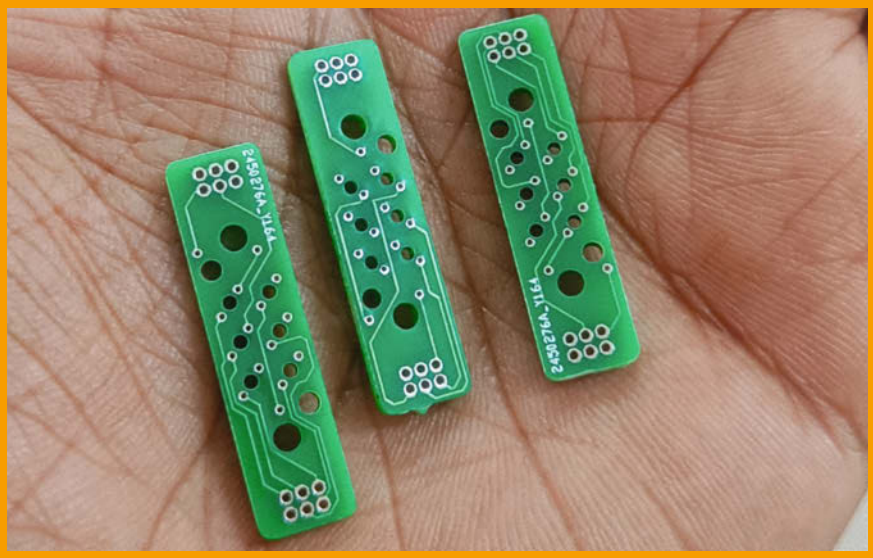
Tatsächlich kam Varada bereits vor 10 Jahren das erste Mal in Berührung mit dynamischen Braillezeilen und hat auch schon 2016 einen Entwurf für den Hackaday-Wettbewerb eingereicht. Damals bewegten sich die Stifte mithilfe kleiner Motoren auf und ab. Sein aktuelles Design ist von Flip-Dot-Displays inspiriert und nutzt winzige Nocken aus 3D-gedrucktem Kunstharz, die zwischen zwei stabilen Positionen wechseln können. Dafür befindet sich in jedem Nocken ein ebenso kleiner Permanentmagnet und darunter ein Elektromagnet auf einem Treiber-Board. Dieses wird von einem Arduino-Nano angetrieben. Sobald das Board den Elektromagneten aktiviert, dreht sich der Nocken und schiebt den zugehörigen Braille-Stift entweder aus dem Gehäuse hinaus oder lässt ihn wieder zurückfahren. Außerdem verharrt der Stift in der gewünschten Position, ohne dass permanent Strom benötigt wird. Um die winzigen Elektromagnete herzustellen, hat Varada sich mithilfe seines 3D-Druckers eine kleine Maschine gebaut, die den Draht um einen Ferritkern wickelt.

Weitere Informationen zum Projekt, eine detaillierte Bauanleitung sowie alle benötigten Dateien für die Treiber-Platinen und den 3D-Druck gibt es unter dem folgenden Link. *—akf*

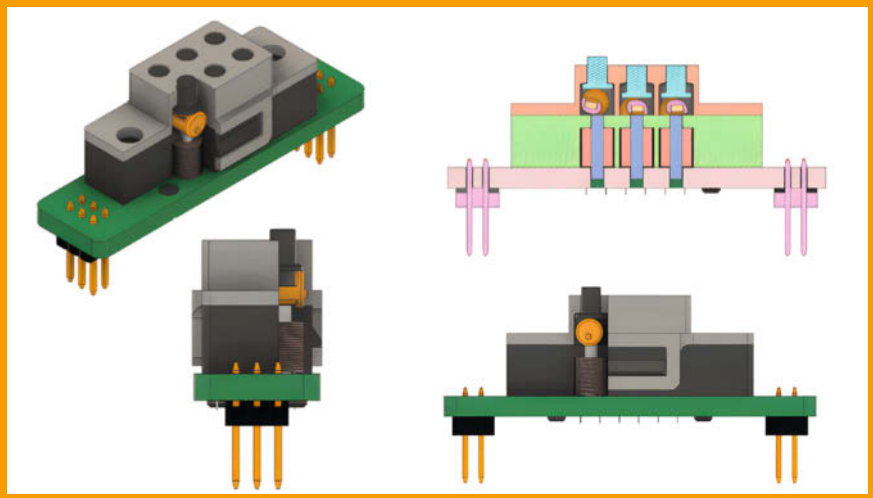
▶ make-magazin.de/x7x7



Mithilfe eines speziellen Treiberboards lässt sich aus mehreren Modulen eine Braillezeile bauen.



Die Platinen für die einzelnen Braille-Module kosten nur wenige Cent. Die Entwürfe gibt es kostenfrei als Download.

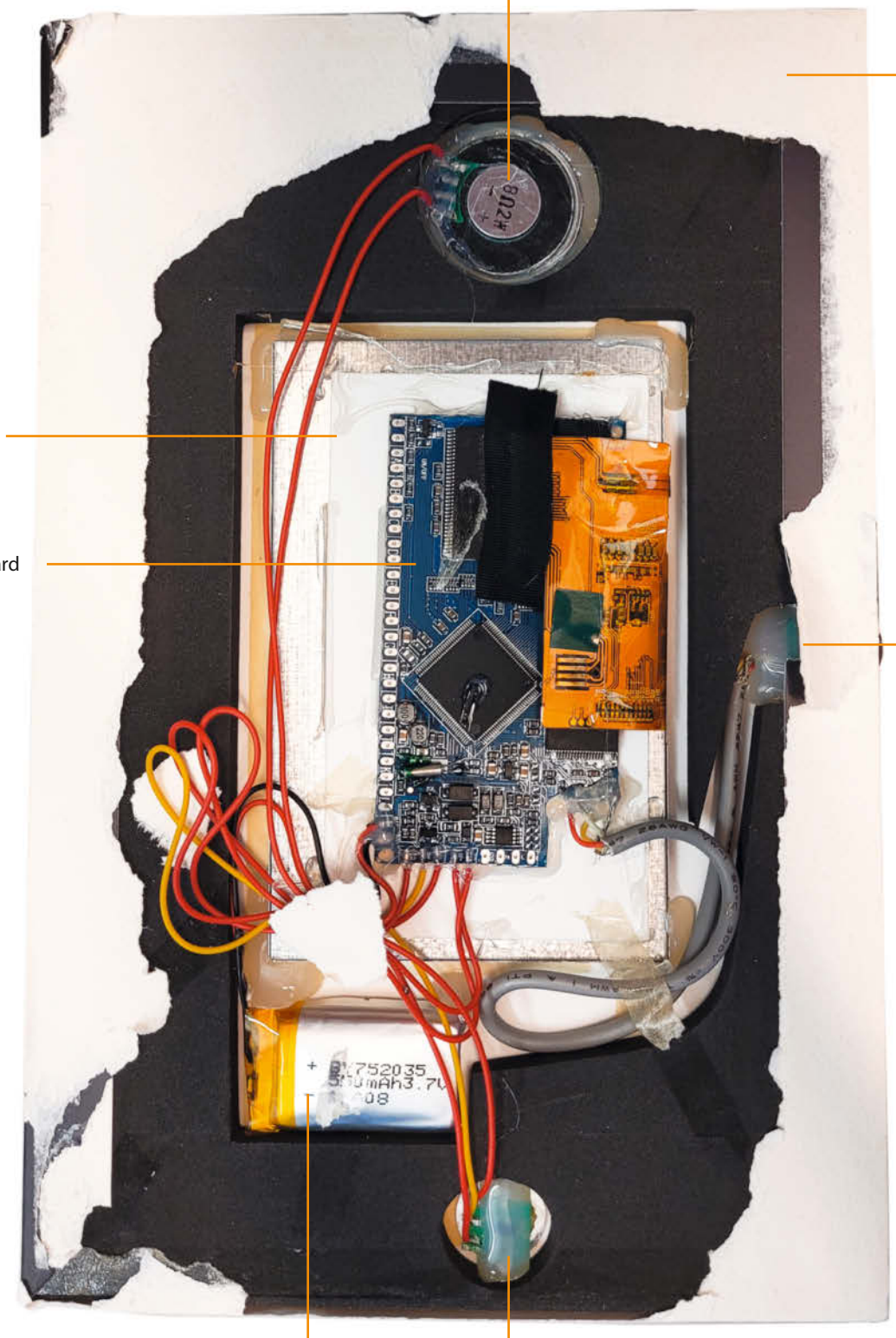


Ein sich drehender Nocken kann die Braillestifte einzeln aus dem Gehäuse drücken. Der Mechanismus ist von Flip-Dots inspiriert.

Videokarte

Grußkarten, die beim Aufklappen nervige Musik in grausamer Qualität abspielen, hat vermutlich jeder privat schon mal bekommen. Auf teure Videokarten stößt man wohl eher bei Werbekampagnen im Unternehmensumfeld. Wir haben uns das Innenleben mal näher angeschaut.

von Daniel Bachfeld



Ganz schön viel Technik hinter der Pappe.

In der Redaktion erhalten wir solche mit Video-Display ausgestatteten Karten in der Regel zur Ankündigung von Produkten eines Herstellers – Kostenpunkt je nach Größe des Displays zwischen 25 und 60 Euro.

An das Innere der Videokarte kommt man leider nur maximal invasiv: Man muss die verklebte Pappe auf der Rückseite abreißen. Da offenbart sich dann etwas, das durchaus auch so in der Make als Projekt zu finden sein könnte: bestehend aus Akku (550 mAh), Lautsprecher, Controller-Board und farbigem TFT-Display mit 4,3-Zoll. Zum Abspielen des Videos muss man die Karte aufklappen, wobei sich ein Magnet von einem Hallsensor entfernt und alles einschaltet!

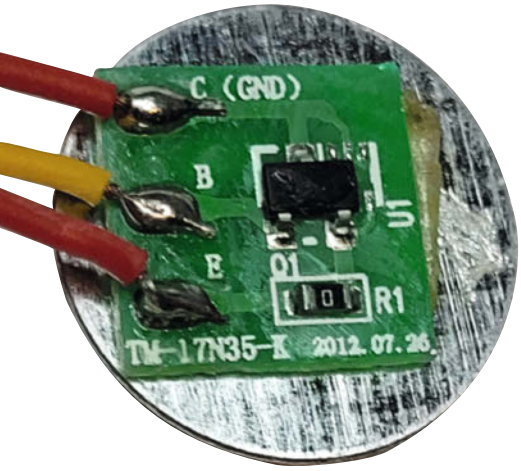
Schließt man die Videokarte per USB an den PC an, so meldet sie sich als mit FAT32 formatiertes USB-Drive mit 75 MByte an. Der auf dem Board verbaute NAND-Flash HY27UF081G2A von Hynix (siehe auch den Artikel über Speicherarten auf S. 104) ist zwar 128 Mbyte groß, der Controller scheint jedoch nicht die volle Größe bereitzustellen. Der Controller ist vom Typ CC1783 und hat den Aufdruck „China Chip“(?). Datenblätter sind leider keine zu finden. Ihm ist ein DRAM HY56DU1622 mit 16 MByte zur Seite gestellt.

Offenkundig kann der CC1783 H.264-Videos mit 480p inklusive ihrer dazugehörigen Audiospur abspielen. Er kann auch mehrere in seinem Flash abgelegte Videos (.mov und .mp4) in der Reihenfolge ihres Dateinamens abspielen.

Das Innenleben der Karte lässt sich prima in eine andere Hülle verfrachten und mit anderen Videos wiederverwenden. Daneben bietet die Karte reichlich Material für das Ersatzteilregal. Das Display bringt eine flexible Leiterplatte mit, deren Anschluss in einen FPC/FFC (ähnlich dem Steck-Zuklapp-Verbinder wie beim Pi) passt und damit über Adapter noch in eigenen Projekten eine Heimat findet.

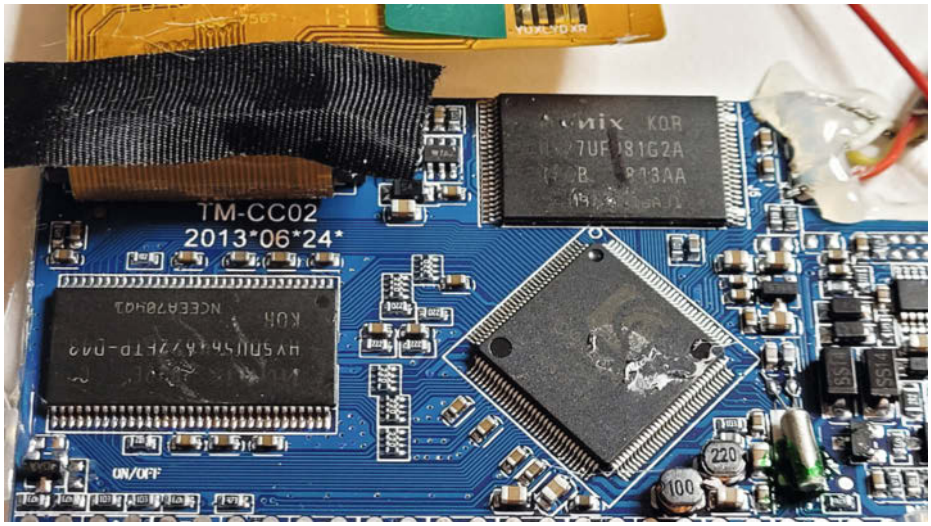
Abgesehen vom CC1783 (mangels Dokumentation) lassen sich auch alle Komponenten des Boards in anderen Projekten einsetzen, so man denn eine Heißluftentlötstation sein Eigen nennt. —*dab*

▶ make-magazin.de/xswy

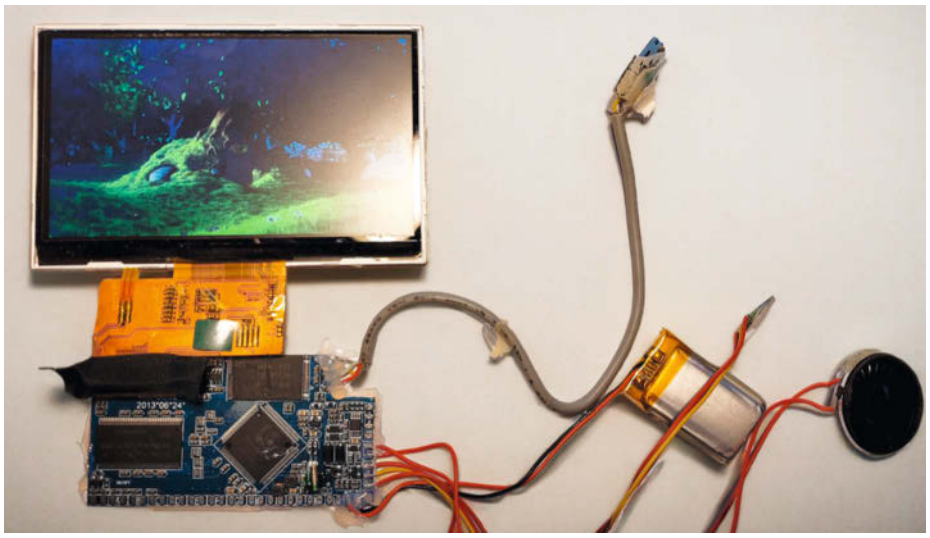


Über den Hallsensor bemerkt der Controller, ob der Deckel der Karte geöffnet wurde. Dann beginnt er mit dem Abspielen der Videos.

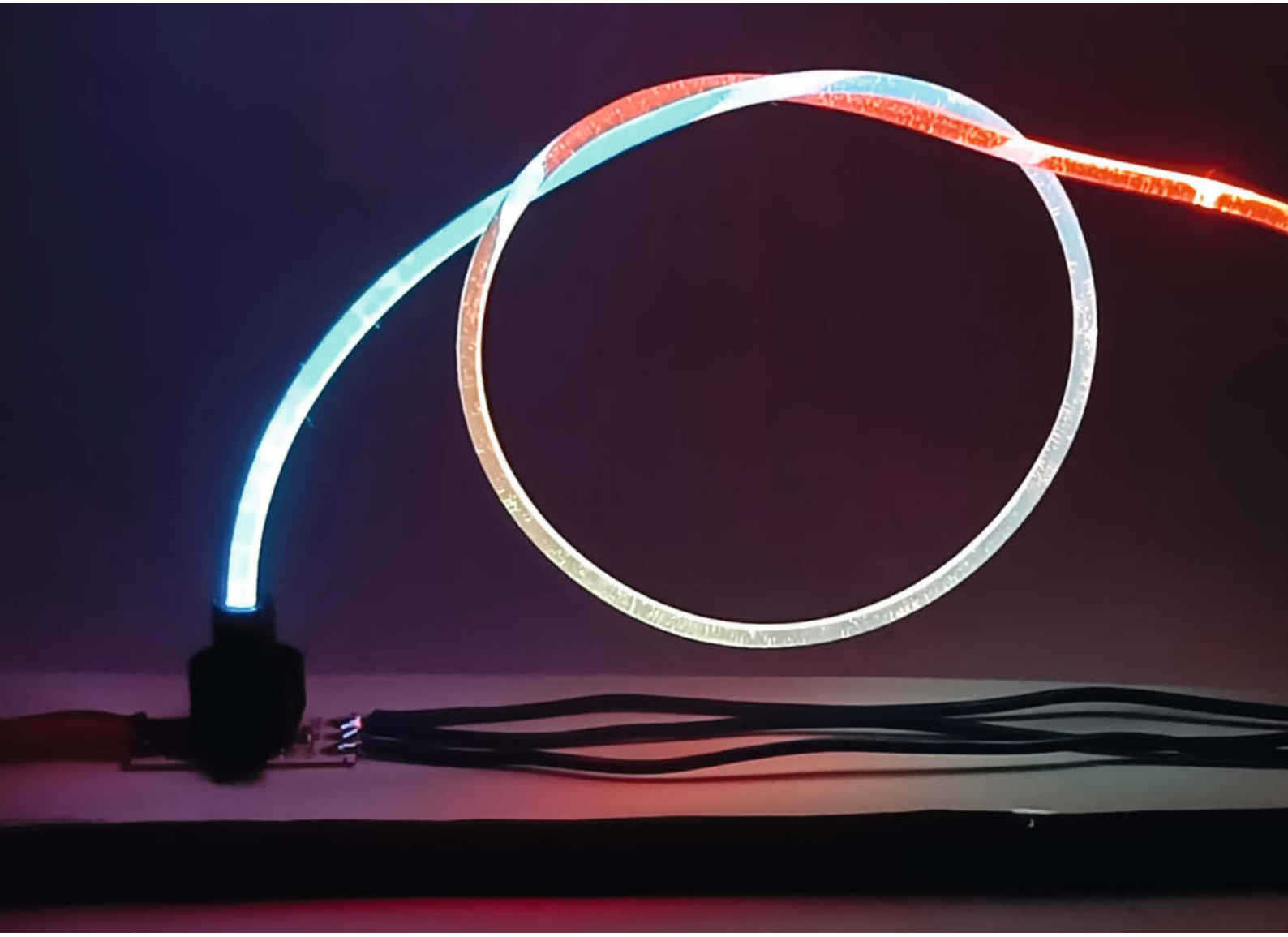
Die Videokarte, bevor wir reingeschaut haben!



Das Board beherbergt den Controller, Flash und RAM sowie Audioausgabe und Displayansteuerung.



Organentnahme: Die Videokarte lässt sich leicht in andere Hüllen verfrachten.



Lichtleiter aus 3D-Druck-Filament

Bei Lichtleitern denkt man an Glasfasern und schnelle Datenübertragung. Dabei hat wahrscheinlich fast jeder schon einmal Lichtleiter aus Kunststoff in elektronischen Geräten gesehen und nicht wahrgenommen: sei es in der Fritzbox oder als Status-LED im ultraflachen Rahmen eines Fernsehers. Maker haben oft auch schon die Zutaten für solche Lichtleiter zu Hause: klare 3D-Druck-Filamente.

von Carsten Wartmann

Kurzinfo

» Klares Filament leitet Licht wie eine Glasfaser
» Beispiele, Experimente und Ideen
» Tipps und Tricks

Checkliste

Zeitaufwand:
1 Stunde

Kosten:
1 Euro

Material

» **Transparentes Filament**
PLA, PETG, ABS, TPU
» **Helle LEDs** und
Stromversorgung

Werkzeug

» **Cutter** oder Skalpell mit schmaler, dünner Klinge
» **Lackmalstifte** oder Acrylstifte

Mehr zum Thema

» Alexander Ehle, LEGO-Grusellabor mit Licht und Sound, Make 2/20, S. 54
» Eva Ismer, Glasfasern in Textilien, Make 6/19, S. 50
» Guido Körber, Plexiglas sandstrahlen, Make 1/19, S. 112

Alles zum Artikel im Web unter
make-magazin.de/xu5q

Lichtleiter bringen Licht ohne Kabel an unzugängliche Stellen. Soll das Licht nicht wie in Glasfasern zur Datenübertragung dienen, sondern einfach nur leuchten, können auch preiswerte Kunststoffe verwendet werden. So wurden Lichtleiter für Geräte populär, in die man nur eine Platine waagerecht einbauen und darauf einfach (SMD-)LEDs löten konnte: Ein durchsichtiges Kunststoffteil leitet das Licht dann um die Ecke zur senkrechten Frontplatte. Bei den heutigen sehr hellen LEDs spielt auch der Lichtverlust im Kunststoff keine Rolle mehr und schließlich können auch spezielle Endkappen (Kreise, Quadrate, Dreiecke, Telefonhörer etc.) beleuchtet werden, ohne speziell geformte LEDs verwenden zu müssen.

Als ich für den Video-Kurs „Konstruieren mit Blender 2.8“ die Augen einer kleinen Statue beleuchten wollte (kostenlose Folge, siehe Links), stand ich vor dem Problem, kleine LEDs genau in die Augenkanäle einbauen zu müssen. Inklusive Kabel schien mir das in dem recht kleinen Innenraum zu fummelig. Ein kurzer Test mit einem Stück transparentem PLA-Druckfilament zeigte mir, dass die Lichtleiteigenschaften dieses Druckmaterials ausreichend waren und so setzte ich dieses Prinzip zum ersten Mal um.

Für eine Handheld-Konsole habe ich dann etwas später vier Stücke Filament verwendet, um die vier SMD-LEDs eines Ladecontrollers für LiPo-Akkus nach außen sichtbar zu machen.

Leuchtende Knöpfe

Während ich vor kurzem Bedienknöpfe für beleuchtete Schieberegler (Bild 3) entwarf, habe ich dieses Prinzip wieder verwendet. Ein kurzes Stück Filament leitet das Licht der Regler-LED an die Oberseite des Knopfes. Das sah gut aus, passte aber nicht so recht zum Design des Gerätes und markierte die Position des Reglers auf seinem Weg nicht so gut. Also probierte ich aus, wie ein von hinten beleuchtetes waagrechtes Stück funktioniert, und das gefiel mir.

Die ersten Prototypen wurden auf einem FDM-Drucker erstellt, das geht schneller als mit einem Resin-Drucker und verursacht auch keine Sauerei. Da mein Prusa MK2 sehr genau eingestellt ist und ich die Toleranzen für die Passungen gut einschätzen kann, passte das Filament nach wenigen Testdrucken in die Bohrung: Das waagerechte Filamentstück rastet regelrecht ein, das senkrechte Stück hält ohne Kleber. Auch das endgültige Design, gedruckt auf dem Harzdrucker (Prusa SL1), passte sofort.

Die Variante mit dem Lichtpunkt brauchte nicht geklebt zu werden, die Querstücke wurden mit einem winzigen Tropfen CA-Kleber (Superkleber), der mit einem Zahnstocher aufgetragen wurde, vorsichtig fixiert.

Anschließend wurde der Überstand mit einem Skalpell parallel zu den Ober- und Sei-

Bild 1: Lichtleiter beleuchten die Augen.

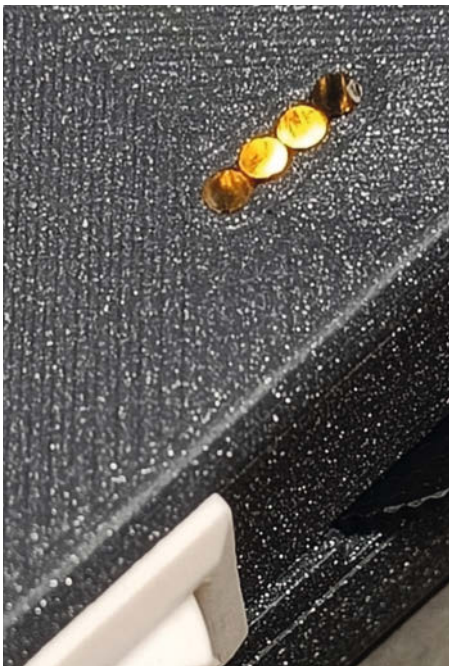


Bild 2: Klassische Anwendung, um das Licht der LEDs von der Platine nach außen zu bringen

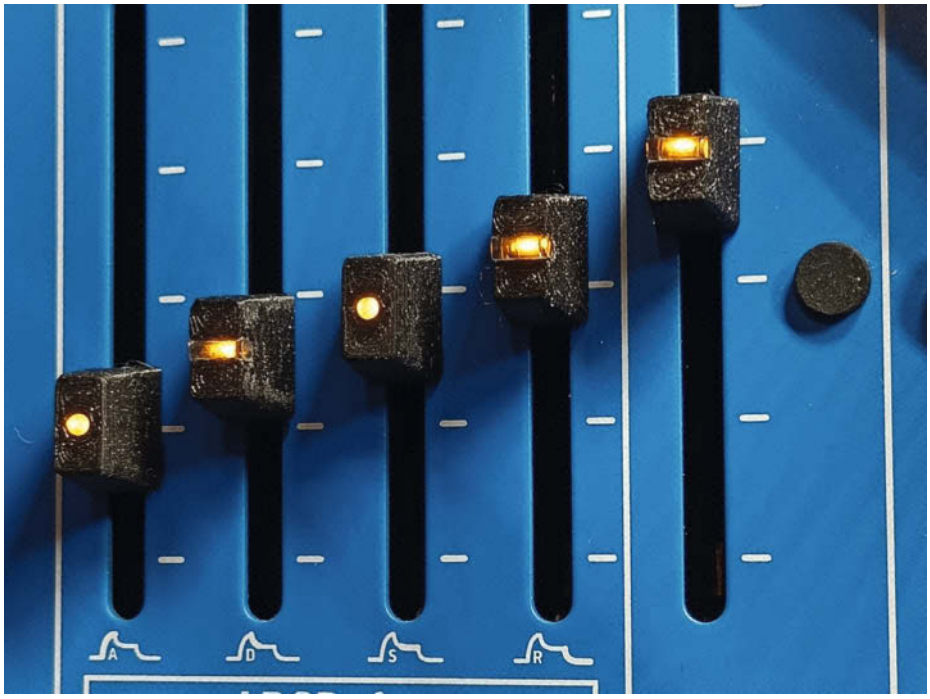


Bild 3: Prototypen der Knöpfe in PLA (Galaxy Black) gedruckt

Bild 4: Einkleben des Lichtleiters mit einem winzigen Tropfen CA-Kleber. Das endgültige Design wurde aus Resin (Prusament Anthracite Grey) gedruckt.

tenflächen der Knöpfe sauber abgeschnitten. Damit erhält man bereits recht saubere Lichtaustrittsflächen. Wer ein diffuses Erscheinungsbild wünscht, kann die Ein- bzw. Austrittsflächen auch mit feinem Schmirgel aufrauen oder für eine hohe Lichtausbeute polieren.

Die quer montierten Filamentstücke wirken nicht wie Lichtleiter, sondern eher wie Linsen, die die dahinter liegende LED vergrößern und in Filamentrichtung strecken, sodass ein kleiner Lichtstreifen entsteht.

Da ich für meinen Synthesizer (ein ARP 2600 Clone) fast 60 Knöpfe brauchte, musste ich zwei Sätze drucken, dann kleben und abschneiden. Aber nach den ersten Teststücken habe ich für beide Arbeitsschritte zusammen nicht mehr als 30 Sekunden pro Knopf gebraucht.

Die Arbeit hat sich gelohnt, die Regler sind jetzt viel besser zu bedienen, sehen gut aus und die Reglerstellung ist auch im Dämmerlicht gut zu erkennen (Bild 6a und 6b).

Bild 5: Die Filamentstücke sind bereits verklebt, danach werden die Enden bündig mit einem Cutter abgetrennt.

Geht es auch länger?

Ja! Im Vergleich zu nicht ganz billigen, handelsüblichen Lichtleitern ist das 3D-Druck-Filament nicht ganz so klar und enthält gerne mal winzige Luftbläschen oder Materialschlieren, sodass ein Teil des Lichts vom Filament gestreut wird. Aber wie man auf Bild 7 sehen kann, kommt am Ende immer noch genug Licht an, um eine blendfreie Anzeige auf einer Frontplatte zu erhalten.

Wenn man das Streulicht nicht sehen will, kann man den Lichtleiter auch lackieren oder mit einem Schrumpfschlauch versehen. Beim Schrumpfen muss man natürlich aufpassen, dass man nicht zu viel Hitze einbringt. Oder man verwendet PETG oder ABS. PLA hingegen kann natürlich mit Hitze in wilde Formen gebracht werden, wenn man es z. B. in heißes Wasser legt.

Das Leuchten entlang des Filaments kann natürlich auch ein guter Effekt sein, wie bei der interessanten Uhr aus der US-Make-Ausgabe 86 (Online-Artikel für Make- und heise+-Abonnenten siehe Kurzlink), die käuf-

liche Lichtleiter verwendet, die absichtlich auch radial Licht aussenden.

Oder könnte man so auch Miniatur-Neonröhren oder Nixie-Röhren herstellen? Zunächst: Man sollte die hellsten LEDs nehmen, die man bekommen kann. Zwischen verschiedenen Materialien wie PLA, PETG oder TPU gibt es Unterschiede in der Lichtleitung und wie gut oder schlecht das Licht nach außen gestreut wird, was je nach Anwendung erwünscht sein kann. Klares TPU finde ich bisher am besten, aber es ist eher nur als Lichtleiter für Frontplatten geeignet, da es nicht von selbst in der Form bleibt, in die man es biegt.

Einen noch größeren Einfluss hat die Produktion des Filaments: Mikrobläschen und Schlieren im Material variieren teilweise innerhalb weniger Zentimeter. Wenn man ein schönes Stück gefunden hat, muss man es irgendwie in Form bringen. Meine besten Erfahrungen habe ich mit PLA in heißem Wasser gemacht. Scharfe Knicke sollten vermieden werden. Für Displays o. ä. kann man die Rückseite des Filaments mit weißen oder silbernen Lackmarkern bemalen, das erhöht die Lichtausbeute nach vorne erheblich.

Wenn die Filamentstücke zu lang sind, kann man an beiden Enden LEDs anbringen. Ich habe kleine Kupplungen für 5 mm und WS2812b-LEDs entworfen und in PETG gedruckt. Mit RGB-LEDs an beiden Enden erhält man auch interessante Mischfarben in der Mitte des Stranges (Bild 8). Bisher ist es mir aber noch nicht gelungen, etwas herzustellen, das als Display auch komplett tageslichttauglich ist. Irgendwo habe ich bestimmt noch rote Killer-LEDs. Meine Neugier ist jedenfalls geweckt und ich werde weiter experimentieren. —*caw*

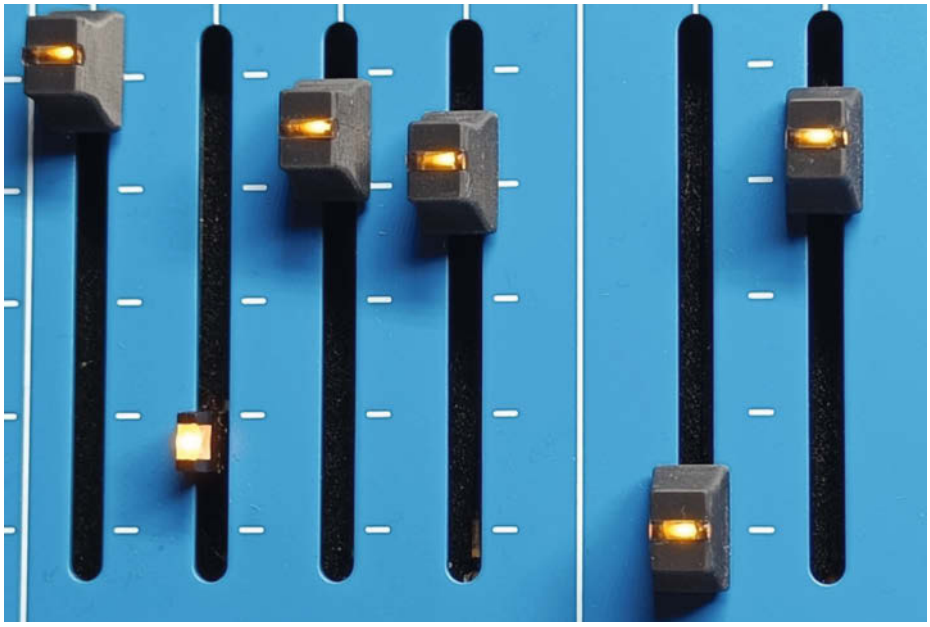
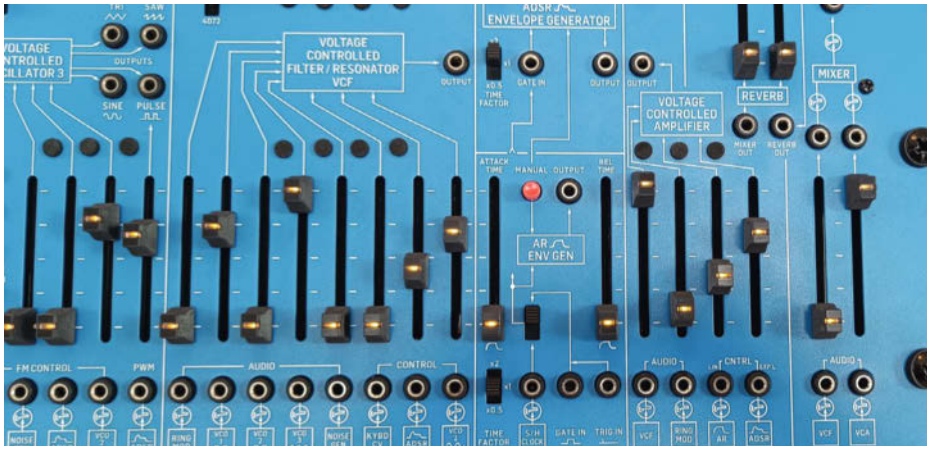


Bild 6a und 6b: Das Ergebnis überzeugt. Der zweite Knopf von links wurde zum Vergleich abgezogen und zeigt den Schieberegler wie er von der Fabrik kommt.



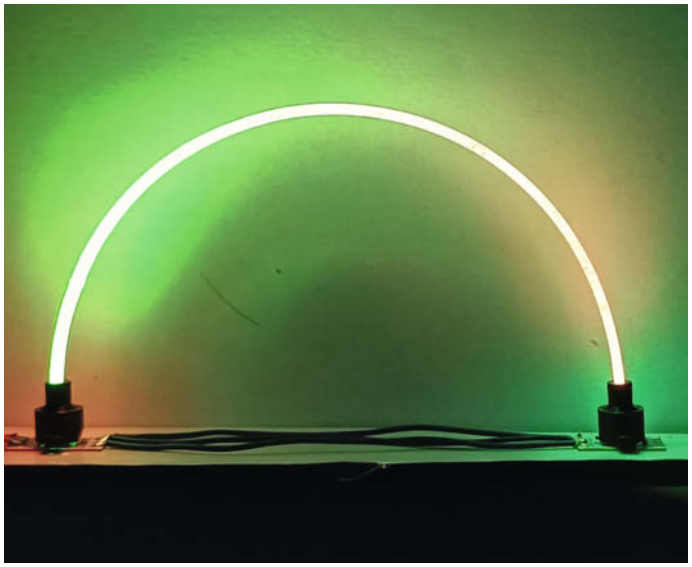


Bild 8: Ein Lichtbogen mit zwei Farben, der Mischeffekt ist im Foto aber nicht so gut zu erkennen wie mit den eigenen Augen.

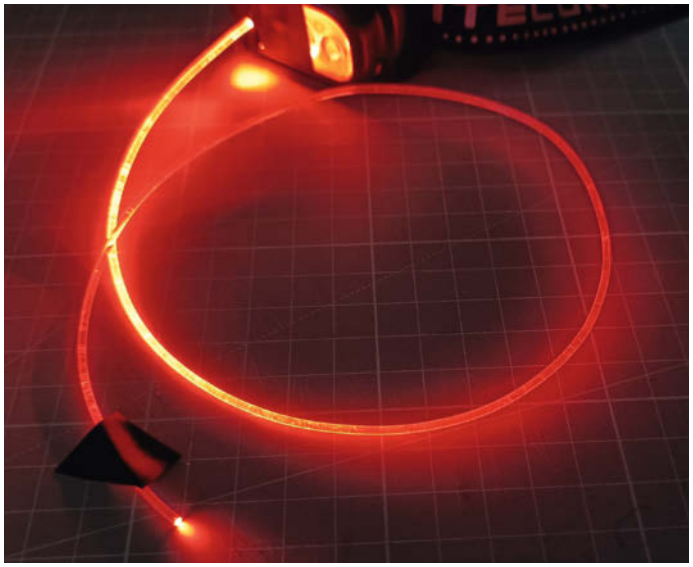
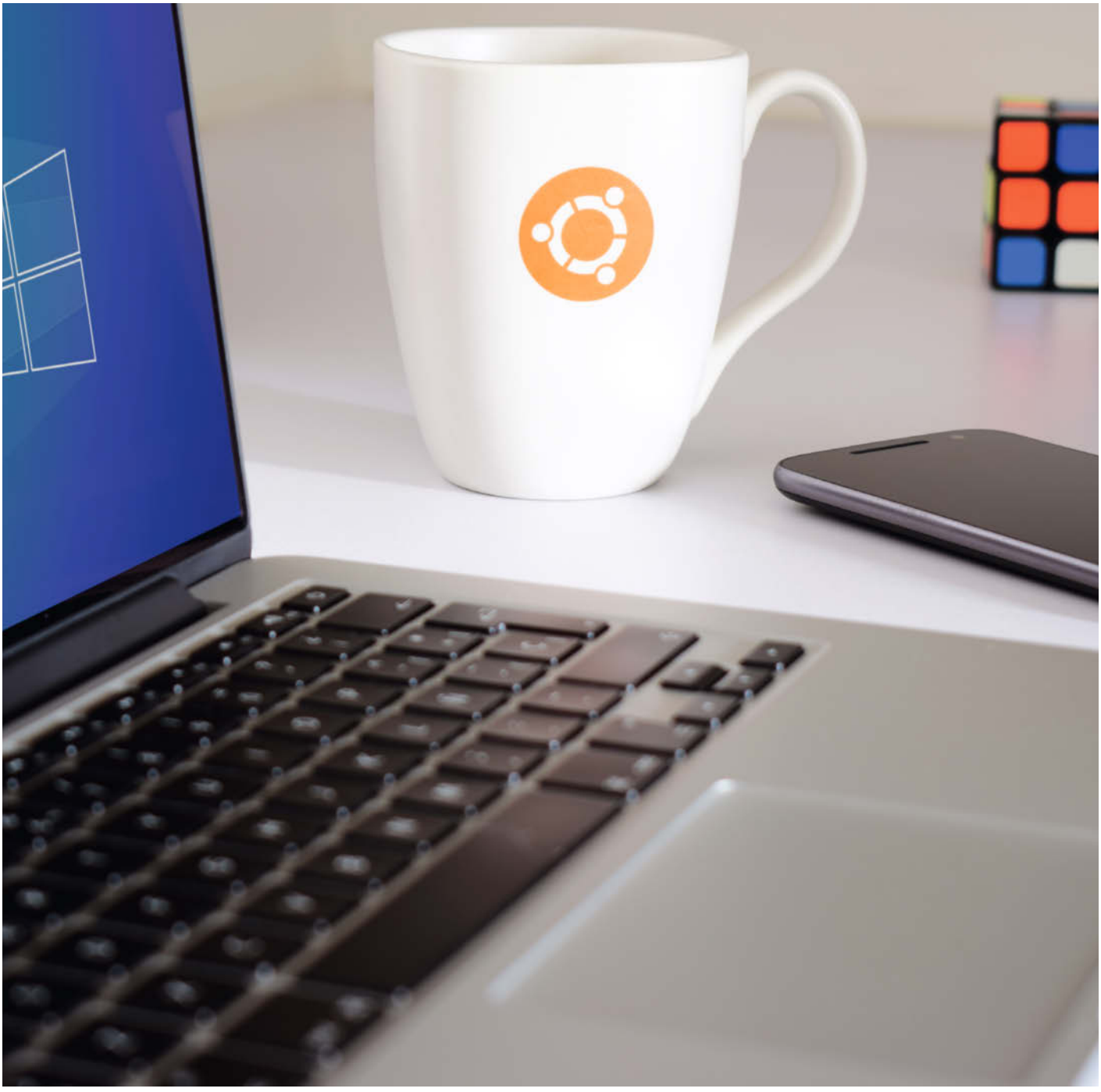


Bild 7: Ein erster Test mit einer hellen Stirnlampe (hier im Astromodus in Rot)

WSL2 für Maker

Manche Dinge funktionieren unter Linux viel einfacher als unter Windows, insbesondere wenn man jenseits der Arduino IDE arbeiten muss. Mit dem Windows Subsystem for Linux kann man ohne Dual Boot oder Virtual Box zum Beispiel Ubuntu direkt in Windows laufen lassen. Wir zeigen wie man das einrichtet und zum Erstellen von Firmware für ESP und Raspberry Pi Pico nutzt.

von Carsten Wartmann



J.D. Canchila/shutterstock.com

Viele Aufgaben, die bei der Entwicklung für Mikroprozessoren anfallen, sind unter Linux einfacher zu erledigen oder sogar nativ für Linux entwickelt. Ein zweiter PC oder Raspi mit Linux, Dual Boot oder eine virtuelle Maschine (VM) sind Lösungsansätze. Für den gelegentlichen Einsatz, wenn die Anwendung für beide Systeme entwickelt wird oder auch für das schnelle Ausprobieren von Linux-Anwendungen und Shell-Skripten bietet sich das „Windows Subsystem für Linux" WSL an. Es bietet eine sehr minimale virtuelle Maschine, auf der eine gängige Linux-Distribution wie z.B. Ubuntu Linux läuft. Und es lässt sich problemlos wieder vom PC löschen, falls man sich mit Linux doch nicht anfreunden kann.

Die erste Version von WSL verfolgte noch den Ansatz, Systemaufrufe von Linux-Binaries (ELF) über eine Art Emulation auf Windows auszuführen, ein Ansatz, der Linux-Benutzern vielleicht von Wine bekannt ist („Wine Is Not an Emulator"). Die neuere WSL2 verwendet eine abgespeckte virtuelle Maschine, die einen normalen Linux-Kernel ausführt.

Wir verwenden hier WSL2, das auch grafische Linux-Anwendungen ausführen kann, seit kurzem neben Windows 11 nun auch für die immer noch beliebte Version 10. Die Installation kann im Microsoft Store oder per Power-Shell gestartet werden. Wir verwenden hier die Power-Shell-Variante, was sich im Artikel viel schneller nachvollziehen lässt als eine lange Bilderstrecke, und am Ende benutzen wir sowieso zu 99 % die Tastatur.

Als Beispiel werden in diesem Artikel MicroPython-Firmwares jeweils für den ESP32 und den Raspberry Pi Pico unter WSL kompiliert. Dies ist sinnvoll, wenn Sie Änderungen an der Firmware vornehmen wollen, spezielle Komponenten für zusätzliche Funktionen oder neue Hardware hinzufügen wollen oder unnötige Komponenten entfernen wollen, die nur Speicherplatz kosten und nicht verwendet werden.

Eine beliebte Erweiterung von MicroPython ist das Modul „ulab", das eine Numpy-kompatible Bibliothek zur schnellen Berechnung von Daten in MicroPython bietet. Anwendungsgebiete sind die Robotik, wie bereits in Make 1/21 gezeigt, oder die Signalverarbeitung. Wenn die Funkverbindung sehr schlecht, zeitlich oder strommäßig begrenzt ist, kann es sinnvoll sein, die Messwerte auf dem Sensorboard zu verarbeiten und nur die ausgewerteten Daten zu senden.

Danach sind Sie bestens gerüstet, um auch eigene Projekte als Firmware auf ein beliebiges Mikroprozessor-Board zu bringen.

WSL2 Installation

Die Bedienung von WSL erfolgt über Befehle in der Windows-Powershell. Diese erreichen Sie am besten über das Suchmenü. Von dort

Kurzinfo

» **WSL mit Ubuntu als Gastsystem einrichten**
» **Die wichtigsten Befehle für WSL**
» **Firmware für Mikrocontroller-Boards kompilieren**

Checkliste

Zeitaufwand:
1 Stunde

Kosten:
0 Euro

Mehr zum Thema

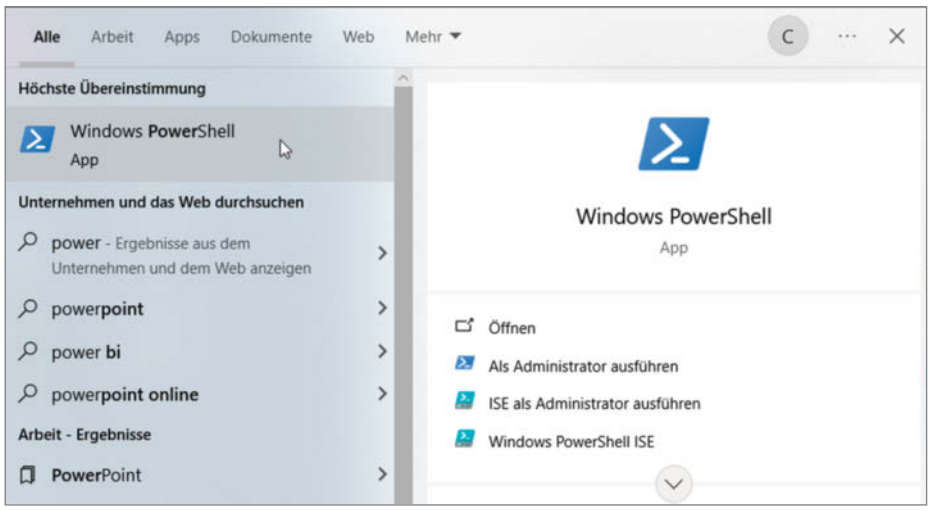
» Ákos Fodor, MicroPython-Boards, Make 7/23, S. 12
» Thomas Euler, MicroPython beschleunigen, Make 1/21, S. 72
» Josef Müller, KI für den ESP32, Make 2/22, S. 86

Alles zum Artikel im Web unter
make-magazin.de/xt7e

WSL1 und WSL2

In der ersten Version von WSL, also WSL1, haben die Entwickler Windows mit einer Abstraktionsschicht versehen: Diese bildet Linux-Systemaufrufe auf Windows-Systemaufrufe ab. WSL1 führt also Linux-Programme aus, ohne einen Linux-Kernel zu benötigen. Für WSL2 hingegen stellt Microsoft einen speziellen Linux-Kernel in einer kleinen virtuellen Maschine (VM) zur Verfügung. Der gleichzeitige Betrieb von WSL1 und WSL2 ist derzeit möglich, allerdings ist unklar, wie lange WSL1 noch von Microsoft unterstützt wird.

Wenn Dienste auf WSL laufen, die auch aus dem lokalen Netz erreichbar sein sollen, ist WSL1 besser geeignet – der Zugriff ist ohne weitere Verrenkungen möglich. WSL1 bietet sich auch an, wenn im Linux-Umfeld häufig auf viele Dateien auf Windows-Laufwerken zugegriffen werden muss – hier ist WSL1 deutlich schneller, da es direkt das Windows-Dateisystem (NTFS) verwendet. Eine besondere Stärke von WSL2 hingegen ist die Möglichkeit, Linux-GUI-Anwendungen auf den Windows-Desktop (Windows 10 und 11) zu bringen. Unter WSL1 funktioniert dies nur, wenn man einen sogenannten X-Server auf der Windowsseite manuell nachrüstet und richtig konfiguriert.



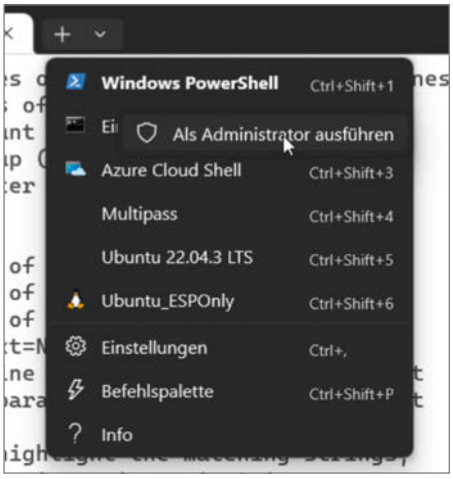
Powershell suchen und finden.

Eingabe der Befehle

Wenn die Befehle für die Powershell gedacht sind, wird dies im Artikel durch PS > in der Zeile angezeigt. Ist die Kommandozeile mit der Shell „Bash“ unter Linux auf WSL gemeint, wird dies durch ein $ angezeigt. Diese Zeichen dürfen nicht in die Kommandozeile eingegeben werden.

Bei Ihnen sieht dieser Prompt (die Eingabeaufforderung) sicher anders aus, da normalerweise noch weitere Informationen wie das aktuelle Verzeichnis oder der Benutzername ausgegeben werden.

Wenn Sie diesen Artikel im Heft lesen, müssen Sie die Zeichen ` (Gravis, Powershell) und \ (Backslash, Bash) nicht eingeben und können einfach in der Zeile weiter tippen. Wenn Sie den Befehl aus dem Online-Artikel kopieren, können Sie Befehle aus mehreren zusammengehörigen Zeilen kopieren und einfügen, die Powershell bzw. Bash fügt die Zeilen dann korrekt zusammen.



Die rechte Maustaste erlaubt die Powershell als Administrator zu starten.

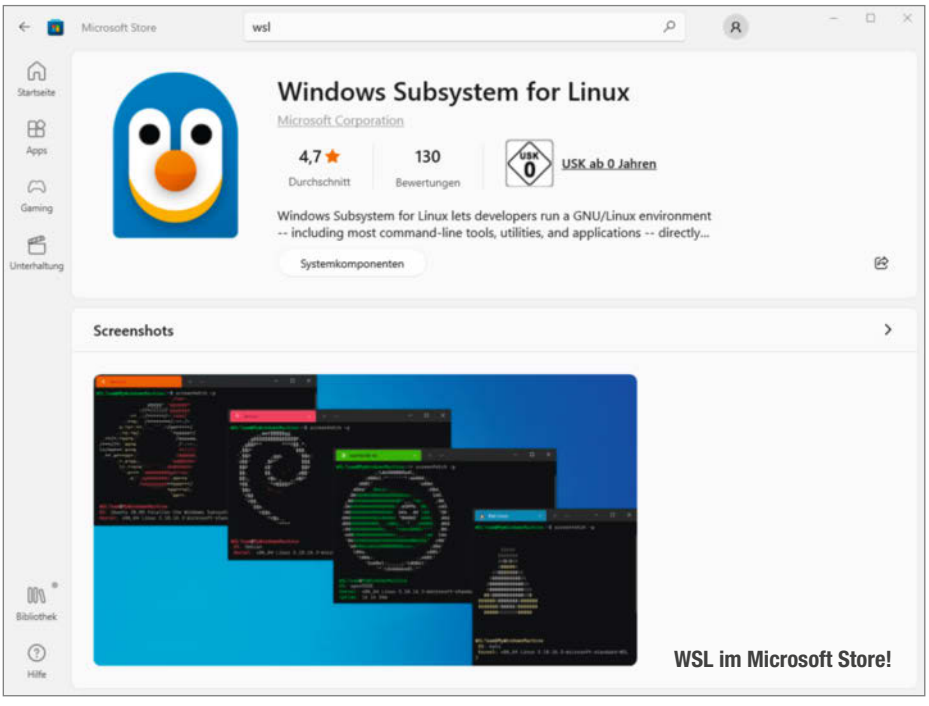
aus kann die Powershell auch auf den Desktop verlinkt oder in die Taskleiste gepackt werden.

Für die Erstinstallation wird eine Powershell mit Administratorrechten benötigt. Diese starten Sie entweder ebenfalls über die Suche (siehe Bild) und dann „Als Administrator ausführen“. Oder, wenn Sie bereits eine Powershell mit normalen Rechten geöffnet haben, über einen Klick mit der rechten Maustaste auf den Powershell-Eintrag im Tab-Menü.

Jetzt sind sie nur noch einen Befehl von einer WSL-Installation entfernt:

```
PS > wsl --install Ubuntu-22.04 `
        --web-download
```

Der Parameter `--web-download` behebt häufige Probleme mit dem Microsoft Store, indem die Distribution über das Web geladen wird. WSL kann auch über den Microsoft Store installiert werden, wenn man es farbig haben möchte, aber danach muss man sowieso wieder auf die Kommandozeile wechseln.



WSL im Microsoft Store!

Nun kann WSL gestartet werden:

```
PS > wsl --distribution Ubuntu-22.04
```

Ein weiterer bequemer Weg führt wieder über das kleine Tab-Menü im Fenster der Powershell, auch die Tastenkürzel sind hier aufgelistet. Und auch im Systemmenü finden sich einige Einträge. Nach ein paar Sekunden Startzeit läuft die Distribution und wir befinden uns in der Linux-Shell.

In einem Powershell-Fenster kann mit

```
PS > wsl --list --verbose
```

eine Liste der verfügbaren Distributionen angezeigt werden und ob sie aktuell laufen.

Firmware für ESP32 Boards

ESP-Chip-basierte Boards erfreuen sich nicht umsonst großer Beliebtheit: Sie sind preiswert, leistungsfähig und bringen drahtlose Netzwerkfähigkeiten mit. Aufgrund ihres Ursprungs als industrielle Mikrocontroller gibt es auch ein sehr ausgereiftes, aber auch komplexes Framework, um sie zu programmieren.

Die wichtigsten Entwicklungspakete sind unter Linux schnell installiert:

```
$ sudo apt update
$ sudo apt install build-essential \
    libffi-dev git pkg-config\
        python3.10-venv cmake
```

Nun zum ESP-Framework von Espressif, dem ESP-IDF (IoT Development Framework): Wechseln Sie in Ihr Heimatverzeichnis, der Download wird das Framework dort installieren:

```
$ cd
$ git clone -b v5.0.4 --recursive  \
      https://github.com/espressif/\
      esp-idf.git
```

Wechseln Sie nun in das neue Verzeichnis „esp-idf“ und führen Sie das Skript „install.sh“ aus. Dies muss nur einmal nach der Installation ausgeführt werden.

```
$ cd esp-idf
$ ./install.sh
```

Nach jedem Login oder Neustart des Linux an der WSL muss jedoch folgendes ausgeführt werden, um die komplexen Abhängigkeiten und Voreinstellungen des ESP-IDF zu setzen:

```
$ source export.sh
```

Wenn man das öfter braucht, kann man diesen Aufruf auch in das Bash-Startskript .bashrc einbauen.

Nun laden wir den MicroPython-Quellcode herunter (clonen):

```
$ git clone https://github.com/\
            micropython/micropython
```

Dann:

```
$ cd micropython
$ make -C mpy-cross
```

Wechseln Sie nun in das Verzeichnis des Boards, in unserem Fall „ports/esp32".

```
$ cd ports/esp32
$ ls boards/
$ make BOARD=ESP32_GENERIC submodules
$ make -j 4 BOARD=ESP32_GENERIC
```

Der Befehl `ls boards/` zeigt an, welche Boards zurzeit unterstützt werden. Sollte das passende Board nicht dabei sein, kann man auch die Boards mit „GENERIC" im Namen verwenden, sollte aber den richtigen Prozessortyp für das Board kennen.

Die make-Befehle kompilieren dann die MicroPython-Firmware auf vier (`-j 4`) CPU-Kernen, diese Zahl können Sie entsprechend Ihrer CPU anpassen. Das fertige „firmware.bin"-Abbild findet man dann in dem zum Board passenden Build-Verzeichnis, hier „build-ESP32_GENERIC/". Dies ist praktisch, falls man es auf einen anderen Rechner mit ESPTool flashen oder weitergeben möchte.

USB-Geräte in WSL

Sobald wir ein USB-Gerät in unseren Windows-Rechner einstecken, wird es von Windows erkannt und eingebunden. Unsere WSL-Distribution hat also keine Chance, an Windows vorbei auf die Hardware zuzugreifen. Abhilfe schafft hier das Windows-Programm „usbipd", welches in der Lage ist, solche USB-Geräte an die Linux-Distribution unter WSL weiterzureichen.

Laden Sie den usbipd-Installer herunter (Link über den Kurzinfolink) und installieren Sie das Programm durch einen Doppelklick auf den MSI-Installer.

Aus einer Powershell heraus kann nun `usbipd list` aufgerufen werden. Es wird eine

```
Windows PowerShell
PS C:\Users\Carsten Wartmann> wsl -l -v
  NAME                   STATE           VERSION
* Ubuntu-22.04           Running         2
  Ubuntu_ESPOnly         Stopped         2
PS C:\Users\Carsten Wartmann>
```

Auflisten der Distributionen auf dem eigenen Rechner

Liste (siehe Bild) mit den vorhandenen Geräten angezeigt. Für uns sind die Geräte interessant, die mit COM bezeichnet sind. In meinem Fall ist es das Gerät mit der Busid 4-2. Im Zweifelsfall das Board ausstecken und schauen, welches Gerät beim Wiedereinstecken erscheint.

Für den nächsten Schritt benötigen wir wieder eine Administrator-Powershell. Passen Sie den Parameter `--busid` entsprechend an:

```
PS > usbipd bind --busid 4-2
```

Rufen Sie den Befehl `usbipd list` erneut auf und das USB-Gerät sollte nun als „Shared" aufgelistet sein. Diese Freigabe überlebt übrigens einen Neustart von Windows und muss daher nur einmal pro Gerät durchgeführt werden. Wenn das Gerät nicht angeschlossen ist, erscheint es unter „Persisted:". Mit `usbipd unbind - - busid x-y` in einer Administrator-Powershell heben Sie diese Zuordnung wieder auf.

Diese Zuordnung kann auch mit der Hardware-ID (`-i VID:PID`) erfolgen, so dass USB-Geräte eindeutig identifiziert werden können, egal wo sie eingesteckt sind. Arbeitet man jedoch mit mehreren gleichen Boards (z.B. mehrere ESP32-Boards), so ist es praktischer, die BUSID zu verwenden.

Ab jetzt reicht ein

```
PS > usbipd.exe attach --wsl -a -b
4-2
```

um das entsprechende Board allen WSL2-Distributionen zur Verfügung zu stellen. Dieser Befehl sperrt die Powershell, wobei der Parameter `-a` das Board automatisch verbindet, wenn es vom USB-Port getrennt und wieder verbunden wird. Strg-C beendet das Programm, wenn es nicht mehr benötigt wird und gibt die Powershell wieder frei.

In der WSL-Distribution können wir nun den USB-Port verwenden. Zuerst müssen wir aber herausfinden, welchem Device-File er zugeordnet ist:

```
$ dmesg
...
[...] usb 1-1: New USB device
[...] usb 1-1: Product:USB..
[...] usb 1-1: SerialNumber:
[...] cdc_acm 1-1:1.0: ttyACM0:USB
ACM
...
```

Wir schauen am Ende der Ausgabe nach „Serial" und „tty"und erkennen, dass unser Board an dem Linux-Device „/dev/ttyACM0" (oder

```
PS C:\Users\Carsten Wartmann> usbipd list
Connected:
BUSID  VID:PID    DEVICE                                                        STATE
1-2    1532:0062  USB-Eingabegerät, Razer Atheris                               Not shared
1-6    0bda:5539  Integrated Webcam                                             Not shared
1-10   8087:0026  Intel(R) Wireless Bluetooth(R)                                Not shared
2-4    0bda:8153  Realtek USB GbE Family Controller                             Not shared
3-1    04e8:4001  Per USB angeschlossenes SCSI (UAS)-Massenspeichergerät        Not shared
4-2    1a86:55d4  USB-Enhanced-SERIAL CH9102 (COM4)                             Not shared
4-5    413c:b06e  USB-Eingabegerät                                              Not shared
5-3    04d9:a232  USB-Eingabegerät                                              Not shared
5-4    0bda:402e  Realtek USB Audio                                             Not shared
5-5    413c:b06f  USB-Eingabegerät                                              Not shared
6-4    046d:085c  c922 Pro Stream Webcam, C922 Pro Stream Webcam                 Not shared

Persisted:
GUID                                  DEVICE
```

Liste der USB-Geräte

Die eigene Firmware

In unserem Artikel zeigen wir, wie man vorhandenen Quellcode kompiliert und gegebenenfalls erweitert. Für komplett eigene Programme, und sei es nur das klassische „Blink.c“, muss man das erledigen, was die Entwickler von Micropython schon gemacht haben. Sowohl das Pico-Entwicklungskit als auch die ESP-IDF erfordern, dass Pfade und Voreinstellungen korrekt gesetzt sind. Für beide gibt es Konfiguratoren auf der Kommandozeile, oder die Integration in die großen Entwicklungsumgebungen stellen Helfer zur Verfügung. Nichtsdestotrotz kann es für den Anfang sehr lehrreich sein, den Urschlamm zu kennen.

Für den Einstieg ist eher ein Projekt mit einem Pico zu empfehlen, die Dokumentation der Raspberry Pi Foundation (siehe Links in Kurzinfo) ist hervorragend und man kommt mit wenigen Schritten, die Sie so ähnlich nun schon kennen, zur ersten blinkenden LED.

Auch für ESP-Boards ist das möglich und sicher auch lehrreich. Espressif empfiehlt aber schon auf den ersten Seiten der ebenfalls hervorragenden Dokumentation doch lieber eine IDE wie Eclipse oder VSCode zu verwenden.

makefile

```
BOARD = ESP32_GENERIC
USER_C_MODULES = /home/cw/ulab/code/micropython.cmake

include Makefile
```

ähnlich, je nach Board und Schnittstelle) angeschlossen ist.

Leider stellen uns Linux und das Espressif IDF noch ein paar Stolpersteine in den Weg: Linux ist sehr auf Sicherheit bedacht und als normaler Benutzer dürfen wir nicht auf serielle Ports zugreifen. Mit „sudo“ würde es gehen, aber der Versuch mit `sudo idf.py -p /dev/ttyACM0 flash` zu flashen schlägt leider fehl. Für den Benutzer „root“ ist die IDF nicht konfiguriert und laut ESP ist das auch so gewollt.

Also müssen wir Linux beibiegen, dass unser normaler Arbeitsaccount (bei mir „cw“, bitte ändern) berechtigt ist. Dies muss nur einmal für das Linux gemacht werden.

```
$ sudo adduser cw dialout
```

Damit gehört der Benutzer „cw“ zur Gruppe `dialout`, die auf die Ports zugreifen darf. Nun müssen wir uns ausloggen und wieder einloggen oder `newgrp dialout` benutzen was allerdings nur für diese neue Shell gilt.

Egal wie, jetzt ist wiederum `source export.sh` aus dem Verzeichnis esp-idf/ auszuführen, um dann endlich mit

```
$ idf.py -p /dev/ttyACM0 flash
```

die Firmware auf das Board zu bekommen. Mit viel Macht entsteht viel Verantwortung.

Möchte man nun über die serielle Schnittstelle auf das Board zugreifen, so ist dies dank „usbipd“ problemlos möglich, dank WSL auch mit grafischen Linux-Programmen. Im Falle von MicroPython könnte man nun Thonny installieren (`sudo snap install thonny`), es auf der WSL-Distribution starten und loslegen. Wer lieber unter Windows weitermachen möchte, muss den Befehl „usbipd.exe attach“ in der Powershell beenden, dann bekommt Windows den seriellen Port wieder.

Achtung Explorer

Für den Windows Explorer ist „makefile“ das Gleiche wie „Makefile“! Besonders in diesem Fall ist das sehr problematisch, da „makefile“ vom make-Befehl für benutzerdefinierte Einstellungen für das eigentliche „Makfile“ verwendet wird. Startet man allerdings aus der Linux-Shell heraus ein notepad.exe mit „makefile“ oder „Makefile“ als Datei, so bekommt man sehr wohl unterschiedliche Dateien.

Ungemach droht auch, wenn man schnell mal mit Notepad.exe eine Datei makefile erzeugen will, dann hängt Notepad nämlich ein .txt an und Linux sucht sich einen Wolf nach „makefile“.

MicroPython erweitern: ulab

Bisher haben wir nur die Standard-MicroPython-Firmware kompiliert. Kommen wir nun zu dem eingangs erwähnten Beispiel der Einbindung von ulab. Viele der Befehle und Prozeduren werden Ihnen jetzt nicht mehr so fremd vorkommen:

```
$ cd ~
$ git clone https://github.com/
v923z/\
      micropython-ulab.git ulab
$ cd ~/micropython/ports/esp32
```

Je nach Boardtyp geht es nun entsprechend weiter, hier für das ESP32 durchgespielt. Für einen Raspberry Pi Pico muss dann im „makefile“ und bei der `BOARD=` Anweisung der Boardtyp durch „PICO_W“ ersetzt werden. Die Pfade müssen je nach Usernamen und Speicherpfad angepasst werden.

Erstellen Sie eine Datei „makefile“ (kleines „m“!) und geben Sie die Zeilen aus dem Kasten „makefile“ ein.

Jetzt zur Sicherheit `make clean` aufrufen. Dann `make -j 4` für die Kompilierung auf vier CPU-Kernen. Die Firmware wird wie im Abschnitt vorher beschrieben auf das Board geflasht.

In REPL in Thonny oder per seriellem Terminal kann man nun überprüfen, ob das ulab-Modul vorhanden ist:

```
MicroPython v1.22.0-preview.246
Type "help()" for more information.

>>> help("modules")
__main__          asyncio/event
hashlib           select
...
aioble/peripheral dht
os                ulab
...
```

Bingo! MicroPython kann nun superschnell mit vielen Daten rechnen.

Backup! WSLs sichern ...

Hat man sich eine schöne WSL-Distribution angelegt und möchte diese sichern, für den Fall das man sie „kaputt spielt“ ist das kein Problem. Der wsl-Befehl möchte allerdings die genaue Bezeichnung der Distribution wissen, also checken wir, wie das WSL benannt wurde.

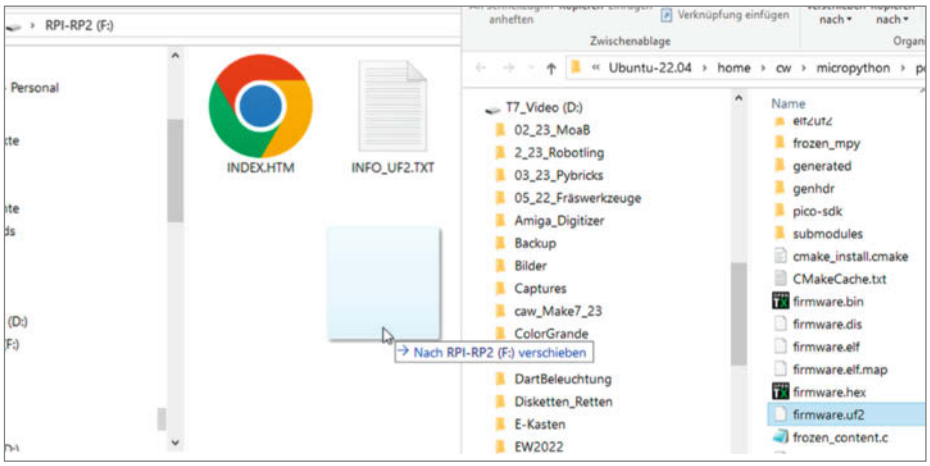
Anschließend muss dann leider das komplette WSL beenden werden, sollten also wichtige Daten auf den Distributionen nicht gespeichert sein, sollten Sie diese zuerst sichern:

```
PS > wsl -list --verbose
NAME            STATE           VERSION
* Ubuntu-22.04    Stopped         2
  Ubuntu_ESP32    Stopped         2
PS > wsl --shutdown
```

Eine Sicherung der WSL2-Linux-Distribution wird dann wie folgt durchgeführt:

```
PS > wsl --export Ubuntu-22.04 `
         D:\Backup\UbuntuTest.tar
```

Das dauert dann etwas, die Daten werden leider unkomprimiert in ein tar-Archiv geschrieben, eine schnelle SSD ist hier Gold wert.



Drag&Drop von Linux nach Windows

... und wiederherstellen

Zum Wiederherstellen einer Sicherung sind folgende Befehle nötig:

```
PS > wsl --set-default-version 2
PS > wsl --import Ubuntu_Test `
     D:\wsl_Ubuntu_Test\ `
     D:\Backup\UbuntuTest.tar
```

Die Default-Version muss gesetzt werden, sonst meckert Windows mit einem recht kryptischen „Wsl/Service/E_FAIL"-Fehler. Der Import dauert je nach Größe der Distribution ein paar Minuten.

Leider „vergisst" WSL nach dem Import wer der Default-User sein soll und man landet als Systemadministrator „root" im Linux. Man kann dann wie folgt (hier mit User „cw") starten:

```
PS > wsl -d Ubuntu_Test -u cw
```

oder man loggt sich als root ein und fügt folgendes zu „/etc/wsl.conf" hinzu:

```
[user]

default=cw
```

Danach `wsl --shutdown` (oder nur die Instanz mit `wsl -t InstanzName` neu starten). Beim nächsten Start wird dann mit dem gewünschten User gestartet.

Pico RP2

Der Prozess eine Firmware für den Pico auf Linux oder auf WSL zu kompilieren ist ähnlich wie beim ESP32. Allerdings ist der Prozess etwas überschaubarer, da die Befehle weniger komplex und zahlreich sind und das Framework (Pico-SDK) nicht so kompliziert ist. Zuerst einmal müssen die passenden Bibliotheken und der Cross-Compiler installiert werden.

Wenn Sie bereits eine Entwicklungsumgebung installiert haben, sind wahrscheinlich schon einige Pakete installiert, aber das macht nichts, apt installiert nichts doppelt.

```
$ sudo apt update
$ sudo apt install gcc-arm-none-eabi \
    libnewlib-arm-none-eabi \
        libstdc++-arm-none-eabi-
newlib \
        build-essential libffi-dev \
        git pkg-config cmake
```

Dann kann es losgehen: Wie und wo man MicroPython herunterlädt steht schon oben im ESP32-Teil. Man kann die gleichen Quellen wie beim ESP32 benutzen und muss es nicht nochmal laden.

Der Parameter `BOARD=RPI_PICO_W` muss je nach Typ des RP2040-Boards in den folgenden Befehlen gesetzt werden. Z.B. „RPI_PICO" für den ersten Pico ohne Wlan oder etwa „ARDUINO_NANO_RP2040_CONNECT" für das Nano Connect. Welche Boards unterstützt werden, kann man im Verzeichnis `ports/rp2/boards` herausfinden. Dann geht es los:

```
$ cd micropython
$ make -C mpy-cross
$
$ cd ports/rp2
$ make BOARD=RPI_PICO_W submodules
$ make BOARD=RPI_PICO_W clean
$ make -j 4 BOARD=RPI_PICO_W
```

Der Parameter „-j 4" startet „make" in vier parallelen Tasks, passen Sie diesen Parameter auf die Anzahl der CPU Kerne in Ihrem PC an.

Wenn alles glatt geht, finden Sie die Firmware nach kurzer Zeit in einem neuen Ordner „build-" mit dem Boardnamen angehängt: Hier also `build-RPI_PICO_W`. Darin befindet sich die Firmware mit dem Namen „firmware.uf2".

Beim Pico wird eine neue Firmware ja üblicherweise installiert, indem man den Boot-Knopf beim Einstecken des USB-Kabels hält. In das nun eingebundene Laufwerk schiebt man die .uf2-Firmware-Datei. Also öffnen Sie doch mal einen Windows-Explorer von Linux aus! Der abschließende „/" vom Pfad-Trenner muss hier weg gelassen werden, oder explorer.exe öffnet das „Dokumente"-Verzeichnis auf Windows.

```
explorer.exe build-RPI_PICO_W
```

Jetzt schieben Sie die .uf2-Datei per Drag&Drop auf dass Pico-Board.

Es ist auch möglich das (Firmware)-Laufwerk des Pico in WSL einzubinden, ebenfalls praktisch für die Programmierung später.

```
$ sudo mount -t drvfs F: /mnt/f
$ ls /mnt/f
INDEX.HTM   INFO_UF2.TXT
    'System Volume Information'
```

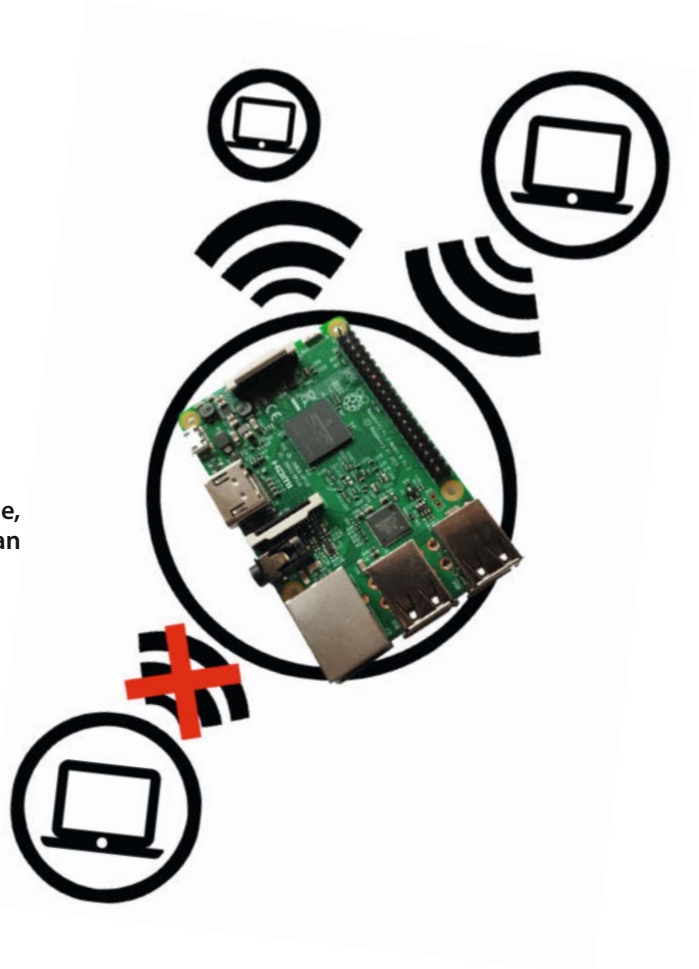
Dann mit „cp" die Datei „firmware.uf2" dorthin kopieren. Allerdings ist das mount-Kommando nach jedem Ab- und Anstecken des Boards wieder nötig. *—caw*

Windows von Linux aus

Alle Windows-Programme und Anwendungen, die sich im Befehls-Suchpfad von Windows befinden, können von Linux aus gestartet werden. So kann man z.B. einen Quellcode oder ein Projekt in VSCode unter Windows bearbeiten, indem man Code.exe (aus dem entsprechenden Pfad) mit der Datei als Parameter aufruft.

Aber auch wsl.exe (das .exe ist hier nötig!) lässt sich statt aus der Powershell direkt aus der Bash aufrufen.

Netzwerk-sicherheit mit Raspberry Pi



Das Internet ist ein Ort, an dem es alles gibt. Auch Kriminelle, die einem schaden wollen. Mit einem Raspberry Pi kann man das heimische Netz jedoch ein wenig sicherer machen.

von Daniel Schwabe

Raspberry Pi als Firewall

Die Software IPFire ist ein mächtiges Tool, das einen Rechner zu einem Router macht. In dem hier dargestellten Verwendungszweck wird ein Raspberry Pi mit IPFire bespielt und in das normale Netzwerk eingegliedert, welches bereits einen Router besitzt. Damit sollen dann bestimmte Geräte im Netzwerk verwaltet werden. Diese Konfiguration ist in erster Linie zum Rumprobieren und Kennenlernen des Einrichtungsprozesses und der Oberfläche gedacht. Aber wenn man z.B. von einem anbietergestellten Router mit wenig Funktionen abhängig ist, kann die Installation auch in diesen Fällen für mehr Komfort bei der Netzwerkkonfiguration sorgen. Oder wenn man sich mit einem anderen Haushalt den Internetzugang teilt und deshalb eigene individuelle Einstellungen braucht.

Flashen des Images

Den Link zu IPFire finden Sie in der Online-Info. Dort erhalten Sie auch einen Link zu einer Kompatibilitätsliste.

In diesem Fall ist es das aarch64-Image. Wichtig ist, dass man die Version mit dem Namen Flash Image herunterlädt. Wie man ein Image auf eine SD-Karten aufspielt erfahren Sie im Artikel Verbindungen zum Raspberry Pi Server in Make 5/23 S. 72.

Bevor man jetzt aber die microSD-Karte aus dem PC nimmt, muss noch eine Datei auf der Karte abgeändert werden. Mit dem Standard-Texteditor öffnet man die Datei uENV.txt und setzt den Wert `SERIAL-CONSOLE=` auf `OFF`.

Jetzt kann die Karte in den Raspberry Pi und dieser mit einer bestehenden Verbindung zu einem Monitor über HDMI und einer Tastatur gestartet werden.

Konfiguration

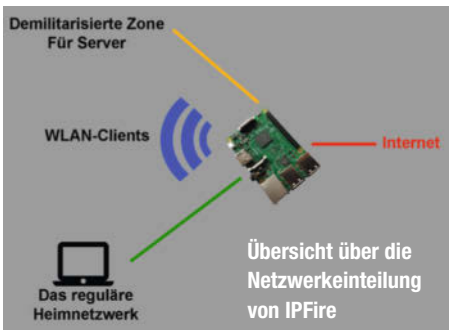
Beim ersten Hochfahren öffnet sich der Einrichtungsassistent von IPFire. Nach grundlegenden Einstellungen wie Tastaturlayout und Zeitzone braucht der Raspberry Pi einen Hostnamen. Das ist im Endeffekt der Name des Computers im Netzwerk. In diesem Beispiel aus Gründen der Originalität pifire.

Danach wird man noch nach einem Domainnamen gefragt. Dieser Domainname wird später an den Hostnamen angehängt. Man kann ihn auf dem Standardwert localdomain belassen. Dadurch lässt sich der Raspberry Pi im Netzwerk später mit `pifire.localdomain` ansprechen.

Als nächstes wird man nach zwei Passwörtern gefragt. Einmal für den root-Account des Raspberry Pi. Damit kann man sich später z.B. über SSH auf dem Gerät einloggen und auf die Kommandozeile zugreifen. Zum anderen nach dem Passwort für den Admin-Account für die grafische Oberfläche von IPFire. Beide Passwörter sollten möglichst sicher sein. Vor allem das Admin-Passwort wird später benötigt.

Die nächsten Schritte betreffen das Einrichten des letztendlichen Netzwerks. In einer Übersicht bekommt man die drei Listenpunkte

- Network configuration type
- Drivers and card assigments
- Address settings



Übersicht über die Netzwerkeinteilung von IPFire

Diese Drei Punkte müssen alle von oben nach unten nacheinander abgearbeitet werden.

Network configuration type

Der erste Punkt legt fest, wie das Netzwerk aufgebaut ist und welche Teile bereitgestellt werden sollen. IPFire nennt die verschiedenen Netzwerkteile Green, Red, Orange und Blue. Wofür die verschiedenen Farben stehen, ist der Grafik zu entnehmen.

Für das hier behandelte Minimalsetup wählt man GREEN + RED. Das bedeutet, dass ein normales Heimnetz aufgebaut ist, in dem dann die PCs verbunden sind (GREEN) und dass der Pi sich dann mit dem Internet verbindet, um den Geräten in diesem grünen Netzwerk Internet zur Verfügung zu stellen (RED).

Drivers and card assigments

Jedes Netzwerk braucht eine eigene Netzwerkkarte. In diesem Beispiel reichen die WLAN- und die LAN-Schnittstelle des Pis. Bei

Bedarf können aber zusätzliche LAN-Schnittstellen über USB hinzugefügt werden.

Für die hier genutzte Konfiguration wird sich der Raspberry Pi über WLAN mit dem richtigen Router verbinden und dann über LAN sein Netzwerk bereitstellen.

Dem Netzwerk GREEN wird dafür das Gerät **usb: Microship Technology, Inc.** zugewiesen. Auch wenn USB im Namen vorkommt, handelt es sich um die fest verbaute LAN-Schnittstelle.

RED bekommt **sdio: brfmfmac** zugewiesen. Die Daten zum Einloggen ins lokale WLAN werden später eingetragen.

Address settings

Dem GREEN-Interface wird hier eine IP-Adresse zugewiesen. In diesem Beispiel ist es 192.169.1.1 Das ist die Adresse, die diesem Interface und damit im Endeffekt dem Raspberry Pi zugeteilt wird. Die Network Mask kann auf dem Standardwert 255.255.255.0 belassen werden.

Für das RED-Interface wird es jetzt spannend. Hier geht es darum, wie der Raspberry Pi eine Verbindung zum Internet herstellt. Wenn nur der Raspberry Pi als Router in diesem Netzwerk verwendet werden soll, kann man hier auch die Login-Daten des Internetanbieters eintragen.

Hier befindet sich der Raspberry Pi allerdings in einem bereits bestehenden Netzwerk. Deshalb stellt man hier DHCP ein und lässt alle anderen Werte unangetastet. Dadurch bezieht der Pi ganz normal über den Router eine IP.

Nachdem alle drei Punkte abgearbeitet sind, geht es mit dem Bestätigen von DONE weiter.

DHCP server configuration

Diese Einstellungen legen fest, in welchem Bereich der Raspberry Pi IP-Adressen an Geräte vergibt, die am GREEN-Interface angeschlossen sind.

Mit der Leertaste muss der DHCP-Server enabled werden. In den beiden Feldern Start address und End address werden die erste und letzte IP eingegeben, die in diesem Netzwerk vergeben werden sollen. Hier z.B. 192.169.1.100 als Start und 192.169.1.200 als Ende. Außerdem kann man hier den Server eintragen. Diesmal 8.8.8.8 von Google.

Wenn man mit OK bestätigt, startet der Raspberry Pi neu. Jetzt kann man die Verbindung zum Monitor trennen und sich mit einem LAN-Kabel direkt anschließen.

Einstellungen in der GUI

Über https://pifire.localdomain:444 kann jetzt die Weboberfläche von IPFire aufgerufen werden. Diese Adresse ändert sich je nach eingegebenen Daten während der Installation.

Um eine WLAN-Verbindung herzustellen, klickt man unter System auf Wireless Client und dort dann auf New Network. Hier müssen dann der Name, der Verschlüsselungsstandard und das Passwort des Netzwerks eingetragen werden. Nach einem Klick auf Add, verbindet sich der Raspberry Pi mit dem Internet. Das kann eine Weile dauern.

Sobald auf der Startseite von IPFire der Status auf Connected steht, kann man lossurfen.

IPFire ist eine umfassende Routersoftware. Über das UI können jetzt neue Firewall-Einstellungen vorgenommen werden, um beispielsweise bestimmte Dienste zu sperren. Über Blacklists lassen sich zudem bestimmte URLs blockieren, beispielsweise um Werbung abzustellen. Außerdem kann man den gesamten Netzwerkverkehr über einen VPN laufen lassen.

Die Firewall-Regeln funktionieren wie folgt: Es gibt immer eine Verbindungsquelle und ein Verbindungsziel. In diesem Beispiel richten wir die Firewall so ein, dass man aus dem Haupt-Router-Netzwerk auf die GUI von IPFire zugreifen kann.

Dafür klickt man unter „Firewall/Firewall Rules“ auf add und bekommt eine Eingabemaske für neue Regeln.

Dafür wählt man als Source unter Standard networks Any und bei Destination Standard networks RED.

Damit nur auf Port 444 (der Port unter dem die GUI erreichbar ist) zugegriffen werden kann, wählt man noch unter Protocol TCP aus und gibt 444 als Destination Port ein.

Darunter noch ACCEPT auswählen und auf add klicken.

Die Konfiguration ist folgendermaßen zu lesen: Aus allen Netzwerken kann jetzt über TCP auf Port 444 des roten Netzwerks zugegriffen werden.

Kurzinfo

» **Mächtige Router-Software zur Heimnetzorganisation**
» **Erweiterung der Möglichkeiten von Standard-Routern**
» **Netzwerkverkehr über VPN umleiten**

Checkliste

Zeitaufwand:
2 Stunden

Kosten:
80 Euro

Material

» **Ein Raspberry Pi** Mindestens Model 3 B mit Netzteil
» **Ein Windows PC**
» **WLAN**
» **LAN-Kabel**

Mehr zum Thema

» Mike Senese, DIY-TV-Beleuchtung, Make 7/23, S. 44
» Daniel Schwabe, Verbindungen zum Raspberry Pi Server, Make 5/23, S. 72
» Josha von Gizycki, Pi-hole: Das schwarze Loch für Internetwerbung, Make 5/20, S. 34

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

DMZ – Demilitarisierte Zone

DMZ steht für demilitarisierte Zone. Damit ist im Rahmen von Netzwerkarchitekturen gemeint, dass innerhalb eines Netzwerks ein Sub-Netz besteht, das ganz normal auf das Internet zugreifen kann, aber innerhalb des lokalen Netzwerkes nicht auf Geräte außerhalb dieser DMZ Zugriff hat. Das nutzt man unter anderem, wenn man einen Dienst (z.B. eine Website) bereitstellt, der von außen zugänglich ist.

Wird jetzt der von Außen zugängliche Server gehackt, auf dem die Website gehostet wird, kann der Angreifer nicht auf andere Geräte außerhalb der DMZ zugreifen. Das ist eine der wichtigsten Sicherheitsvorkehrungen, wenn man Server im eigenen Netzwerk hostet.

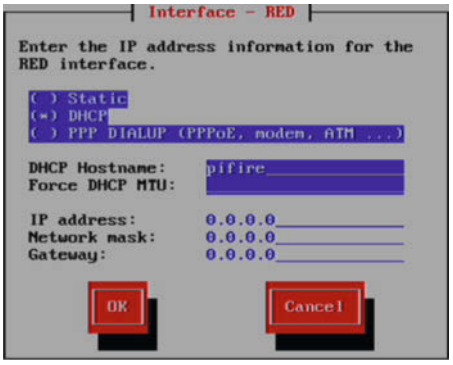
Dadurch kann auch über einen Computer im Netzwerk des Hauptrouters auf die GUI zugegriffen werden.

Das Gleiche funktioniert natürlich auch umgekehrt.

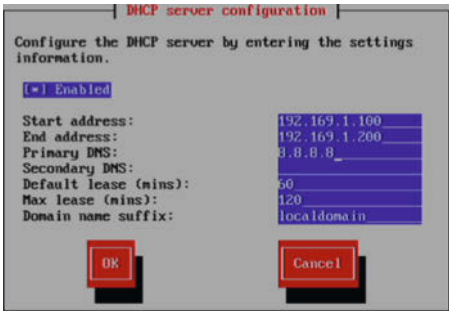
Mit einer neuen Regel, bei der Standard networks GREEN als Source und Standard networks RED als Destination eingestellt werden (diesmal ohne einen speziellen Port) kann man auf Server im Hauptnetzwerk des Basisrouters zugreifen. Damit sind jetzt z.B. NAS-Speicher oder andere Dienste erreichbar.

Diese Regeln können ganz fein eingestellt werden, um bestimmten Computern Zugriff auf andere Teile des Netzwerks zu erlauben oder zu verbieten. Wenn in einer größeren Installation noch WLAN- und DMZ-Netzwerke hinzukommen und im Netzwerk selber Services gehostet werden, ist über diese Zugangsbeschränkungen sicherheitstechnisch viel zu machen. Außerdem können vordefinierte Gruppen von einzelnen Geräten im Netzwerk erstellt werden (unter „Firewall/ Firewall Groups“). Mit diesen Gruppen ist es dann auch möglich, Zugriffe zu gewähren oder zu blockieren.

Hostet man selber für eigene Dienste Server von Zuhause aus, ist auch das Einrichten und Verwalten einer DMZ mit IPFire möglich. Dafür braucht man lediglich einen weiteren LAN-Adapter, der per USB an den Raspberry Pi angeschlossen wird, und die richtigen Einstellungen im Setup.



Die Adressinformationen für die beiden Netzwerke



Die DHCP-Einstellungen sind die letzten Schrauben, die in diesem Setup gestellt werden müssen.

VPN über Wireguard

Ein VPN leitet alle Internetverbindungen zuerst über einen anderen Server um. Dadurch kann man auch auf Geräte zugreifen, die im gleichen Netzwerk wie dieser Server sind (z.B. NAS-Speicher). Das wird auch von Unternehmen genutzt, die Homeoffice anbieten, damit Angestellte von Zuhause auf die Infrastruktur der Firma zugreifen können und das auf einem sicheren Weg. Denn Datenflüsse in und aus dem Firmennetzwerk müssen geschützt sein. Oder man nutzt einen großen VPN-Anbieter mit Servern im Ausland, um dadurch auf Dienste zuzugreifen, die nur dort zugänglich sind. Aber VPNs haben auch für private Nutzung einen gewissen Sicherheitsaspekt. Wenn man zum Beispiel unterwegs ist und sich in

Docker-Compose-Datei

```
version: "3.8"
volumes:
  etc_wireguard:

services:
  wg-easy:
    environment:
      - LANG=de
      - WG_HOST=öffentliche Adresse

      # Optional:
      - PASSWORD=ein-sicheres-Passwort
      - WG_PORT=51820
      # - WG_DEFAULT_ADDRESS=10.8.0.x
      # - WG_DEFAULT_DNS=1.1.1.1
      - WG_MTU=1500
      # - WG_ALLOWED_IPS=192.168.15.0/24, 10.0.1.0/24
      # - WG_PERSISTENT_KEEPALIVE=25
      # - WG_PRE_UP=echo "Pre Up" > /etc/wireguard/pre-up.txt
      # - WG_POST_UP=echo "Post Up" > /etc/wireguard/post-up.txt
      # - WG_PRE_DOWN=echo "Pre Down" > /etc/wireguard/pre-down.txt
      # - WG_POST_DOWN=echo "Post Down" > /etc/wireguard/post-down.txt

    image: ghcr.io/wg-easy/wg-easy
    container_name: wg-easy
    volumes:
      - etc_wireguard:/etc/wireguard
    ports:
      - "51820:51820/udp"
      - "51821:51821/tcp"
    restart: unless-stopped
    cap_add:
      - NET_ADMIN
      - SYS_MODULE
    sysctls:
      - net.ipv4.ip_forward=1
      - net.ipv4.conf.all.src_valid_mark=1
```

ein öffentliches WLAN einwählt, dann aber über einen VPN-Tunnel ins Internet geht, kann ein eventuell heimlich Mitlesender nur verschlüsselte Datenströme sehen.

Diese Anleitung ermöglicht es, von außerhalb auf das Netzwerk zuzugreifen, in das der Raspberry Pi eingewählt ist, und den eigenen Datenverkehr in öffentlichen Netzwerken zu verschleiern, ohne ein Abo bei einem VPN-Anbieter abzuschließen.

Installiert wird Wireguard mit dem Docker Image WireGuard Easy. Dafür ist alles in dem GitHub-Repository in der Kurzinfo zu finden. An dieser Stelle wird eine funktionierende Docker- und Portainer-Installation vorausgesetzt. Dafür gibt es in der Kurzinfo eine Anleitung.

Den Inhalt der Datei docker-compose.yml kopiert man in einen neuen Stack von Portainer. Dafür klickt man auf Stacks und add Stack oben rechts.

Jetzt muss man den Host in Zeile 9 ersetzen. Entweder mit einer Domain, die auf das Netzwerk umleitet, oder der externen IP-Adresse, die man unter https://www.wieistmeineip.de/ abfragen kann. Letzteres ist nicht zu empfehlen, weil sich diese Adresse ändern kann. AVM bietet bei ihren Routern eine sogenannte Myfritz-Freigabe an. Damit erhält man auch eine Adresse, mit der von außen auf das Netzwerk zugegriffen werden kann. In jedem Fall müssen im Router die passenden Ports freigegeben werden. In dieser Konfiguration Port 51820 mit dem UDP Protokoll. Bei Bedarf kann dieser Port auch beim Erstellen des Stacks geändert werden, indem man Zeile 13 modifiziert.

Danach muss in Zeile 12 ein neues Passwort gewählt werden. In Zeile 16 muss man den MTU-Wert (Maximum Transmission Unit) auf den des Pis stellen. Im Normalfall ist dieser Wert 1500, zur Sicherheit kann man sich die Werte für alle Netzwerk-Devices mit dem Kommandozeilenbefehl `ifconfig| grep -i MTU` ausgeben lassen. Ist der Pi über LAN mit dem Netzwerk verbunden, lautet der richtige Eintrag eth0. Für WLAN br-<eine lange Zeichenkette>.

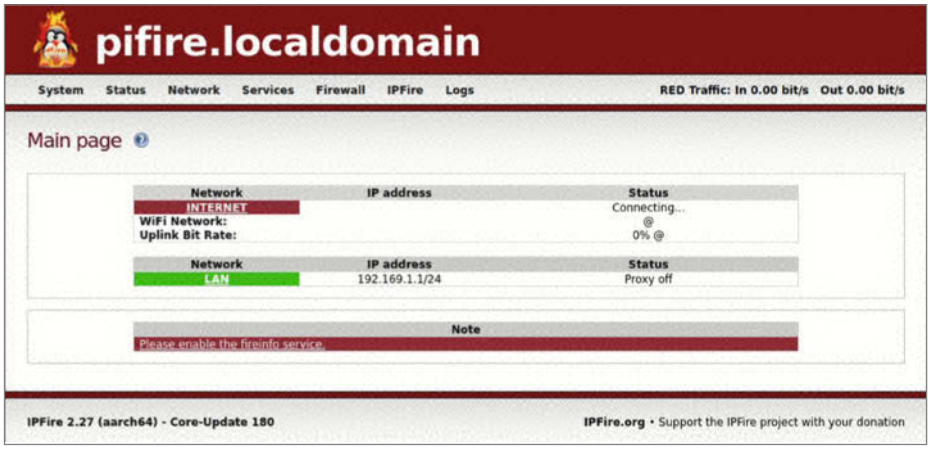
Als Letztes nach unten scrollen und auf „Deploy Stack“ klicken.

Wenn die Container-Erstellung bestätigt wurde, kann man über http://<Lokale-Adresse-des-Pi>:51821/ auf die Bedienoberfläche von Wireguard zugreifen. Der Login geschieht mit dem in Zeile 12 festgelegten Passwort.

Jede Verbindung kann gleichzeitig nur von einem Gerät genutzt werden. Wenn jetzt Handy und Laptop gleichzeitig in den VPN eingeloggt werden sollen, braucht es zwei Verbindungen.

Um sich einzuloggen, benötigt das Endgerät auch einen Client. Passende Client-Software für Mobilgeräte und Desktop sind in der Online-Info verlinkt.

Um sich mit Windows in den neuen VPN einzuwählen, loggt man sich als erstes in die



Noch gibt es kein Internet, weil der Raspberry Pi noch in das WLAN eingewählt werden muss.

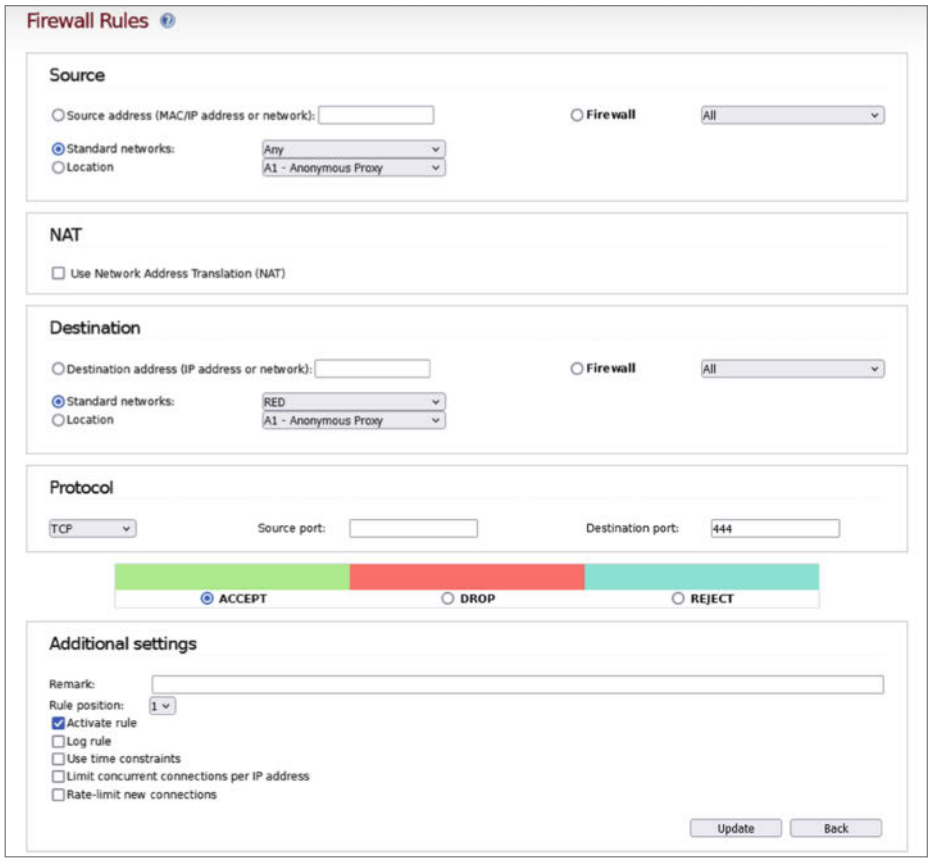
Wireguard-Oberfläche ein. Dort klickt man auf + New und legt eine neue Verbindung an. Diese Verbindung wird dann in einer Liste angezeigt. Dort kann man rechts auf das Download-Symbol klicken und eine <Verbindungsname>.conf-Datei herunterladen.

Nachdem man den Client aus der Online-Info installiert und gestartet hat, kann man dort auf„Importiere Tunnel aus Datei“ klicken und die <Verbindungsname>.conf auswählen. Danach auf Aktivieren klicken und die VPN-Verbindung ist aufgebaut.

Für Handys und Tablets ist das Ganze noch einfacher. Neben der angelegten Verbindung im Wireguard-Dashboard auf das QR-Code-Symbol klicken und diesen über die Wireguard-App mit der Gerätekamera einscannen.

Brain.exe

Diese Tools können helfen, den Netzwerkverkehr sicherer zu machen. Trotzdem ist beim Surfen im Internet das Nachdenken vor jedem Klick das Wichtigste, was man tun sollte. *—das*



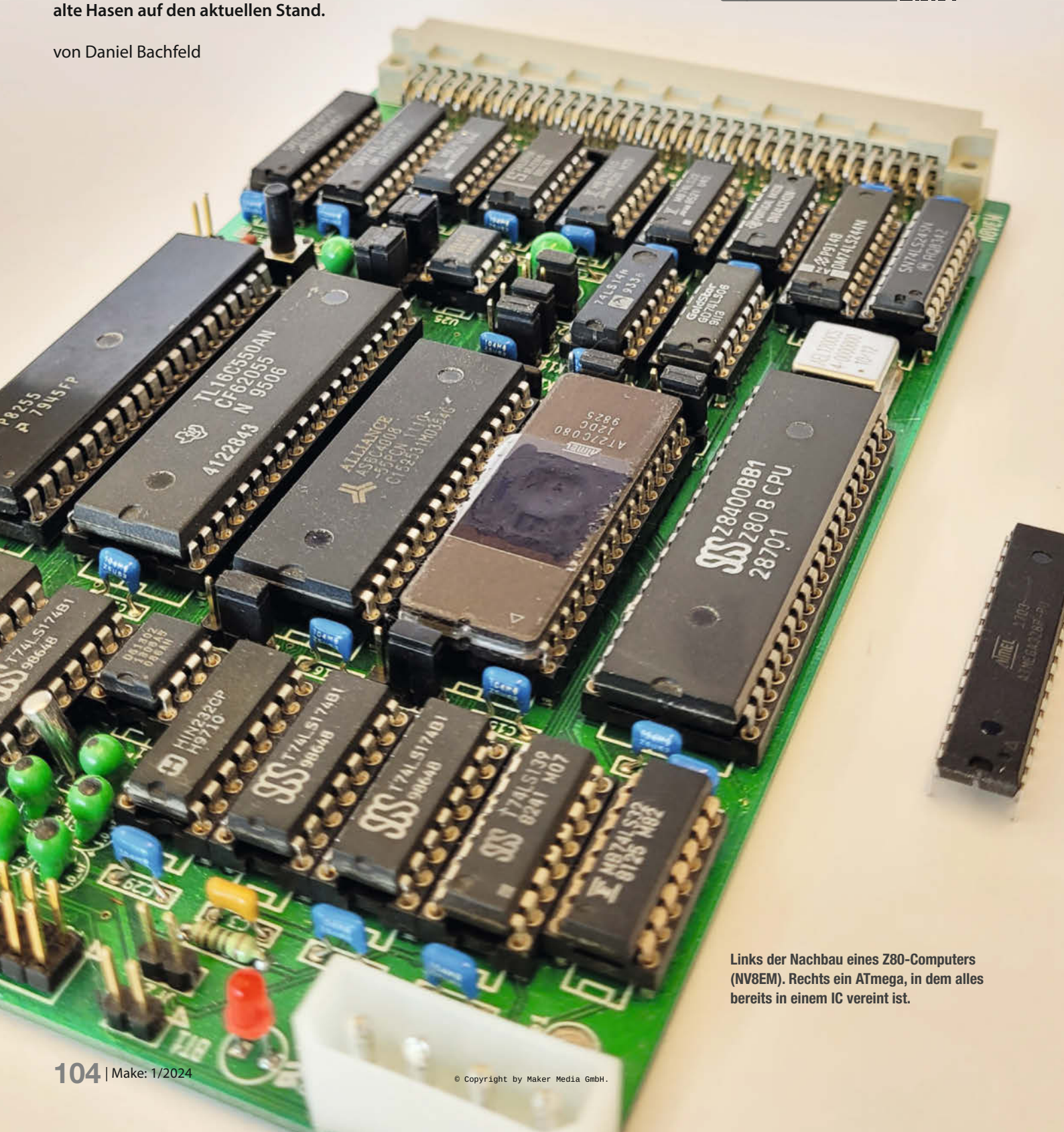
So kann eine Firewall-Regel aussehen. Zugriffe werden nach Quelle und Ziel unterteilt und können so fein eingestellt werden.

Überblick: Speicherarten

In Make-Artikeln werfen wir oft mit Fachbegriffen rund um Speicher um uns: RAM, ROM, Flash, EEPROM, SRAM, PS- und SPI-RAM und so weiter. Unser „Refresher" führt Neulinge in das Gebiet ein und bringt alte Hasen auf den aktuellen Stand.

Alles zum Artikel im Web unter *make-magazin.de/xb9v*

von Daniel Bachfeld



Links der Nachbau eines Z80-Computers (NV8EM). Rechts ein ATmega, in dem alles bereits in einem IC vereint ist.

Ohne Speicher geht nichts: Kein modernes, digitales Gerät kommt ohne ihn aus. Weder PCs noch Smartphones, Smart-TVs oder WLAN-Router funktionieren ohne. Programme beziehungsweise Apps müssen irgendwo dauerhaft (nichtflüchtig) abgelegt und temporäre Daten wie ein Film- oder Daten-Stream müssen zur Verarbeitung zwischengespeichert werden. Damit es nicht zu unübersichtlich wird, konzentrieren wir uns hier vorerst auf das, was man etwa auf beziehungsweise in einem Raspberry Pi, in einem Arduino UNO und einem ESP32 vorfindet.

Stromlos

Generell unterscheidet man zwischen nichtflüchtigem und flüchtigem Speicher. Nichtflüchtiger Speicher gruppiert sich wiederum in permanenten und semi-permanenten Speicher. In einem nichtflüchtigen Speicher bleiben die Inhalte auch ohne Versorgungsspannung erhalten. Permanenter Speicher ist nur lesbar, also Read Only Memory (ROM). Er ist ideal als Befehlsspeicher geeignet, lässt sich aber nur einmalig programmieren – in der Regel über die sogenannte Maskenprogrammierung schon im Halbleiterwerk.

Solch einen ROM findet man etwa in den SoCs der Raspberry Pis. Darin ist der erste Bootloader einprogrammiert, der dann den zweiten Bootloader von SD-Karte (bis Pi 3) oder aus einem EEPROM lädt (ab Pi 4). Fun Fact: Für den Bootprozess ist die Grafikeinheit (GPU) des Pi zuständig, die die ARM-Kerne erst nach dem Kopieren des Linux-Kernels in den DRAM aufweckt, um den Staffelstab zu übergeben. Ok, EEPROM, SD-Karte und DRAM erklären wir gleich.

ROM

Der ESP32 hat ebenfalls einen unveränderlichen ROM an Bord, in dem ein Bootloader ab Werk gespeichert ist. Der Nachteil von ROMs ist, dass man sie nicht mehr umprogrammieren kann. Man muss sich also seiner Sache sicher sein. Im Unterschied dazu erlaubt semipermanenter Speicher auch das Schreiben in die Speicher. Dazu gehören Erasable Programmable ROMs (EPROMS) und Erasable Electrically Programmable ROMs (EEPROMS).

Ältere Leser unter ihnen werden sich noch an EPROMS erinnern. Seine übersetzten Programme schrieb man mit einem speziellen Gerät, dem Eprommer oder EPROM-Brenner, in einem Zug in das EPROM. Der Begriff „Brennen“ wird bereits im Patent zu den nur einmalig beschreibbaren PROMs erwähnt und geht auf das irreversible, gezielte Durchbrennen der haarfeinen Verdrahtungen interner Dioden durch einen hohen Stromstoß zurück. Im englischen nennt man die feinen Drähte Cat's Whisker oder einfach nur Whisker, also Schnurrhaar. Bei EPROMs brennt aber nichts mehr durch. Stattdessen werden mit einer hohen Programmierspannung Elektronen auf die eigentlich isolierten Floating Gates von Feldeffekttransistoren (MOSFETs) geladen. Damit ist die Speicherzelle programmiert. Den Begriff „Brennen“ hat man beibehalten.

EPROM

Das EPROM ist ein physischer Chip im DIP/DIL-Gehäuse mit einem Quarzglas-Fensterchen im Deckel. Will man ein Update seines

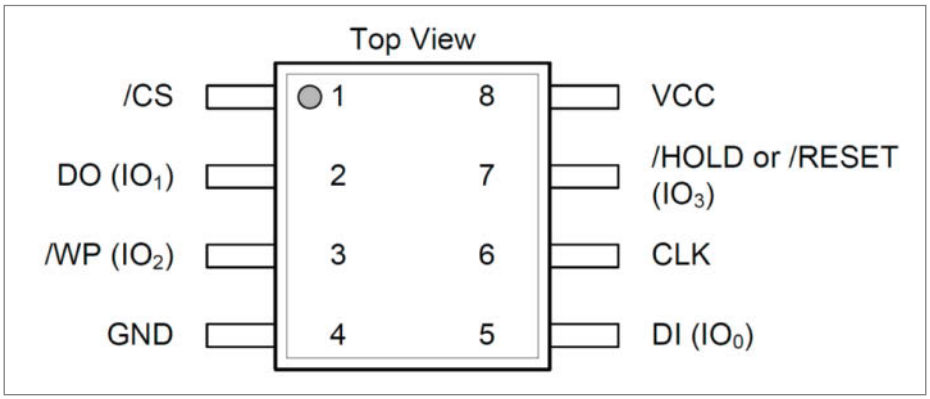


Bild 3: Die Pinbelegung eines seriellen NOR-Flashes mit QSPI-Schnittstelle.

Wikipedia

Bild 1: PROM und EPROM (oben): Das Fenster dient zum Zurücksetzen der Speicherzellen auf logisch 1.

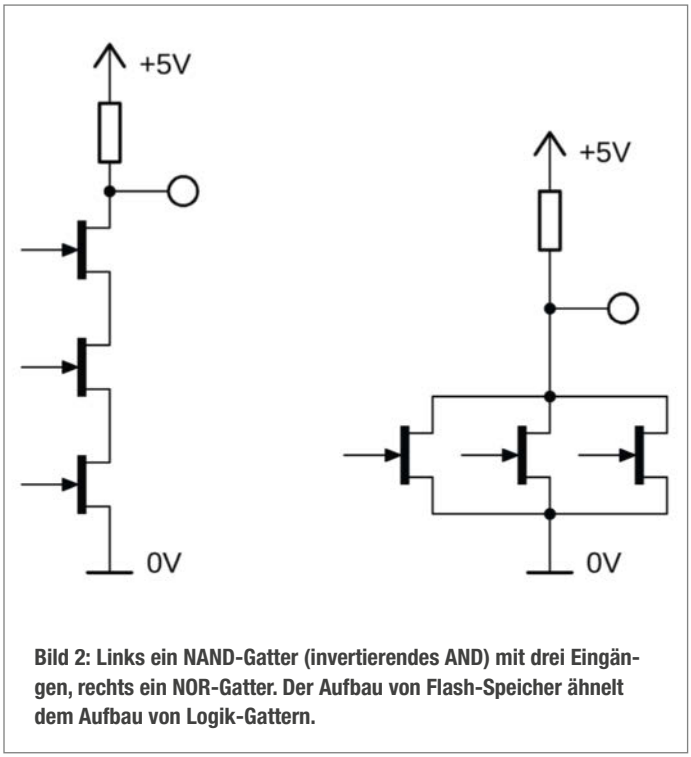


Bild 2: Links ein NAND-Gatter (invertierendes AND) mit drei Eingängen, rechts ein NOR-Gatter. Der Aufbau von Flash-Speicher ähnelt dem Aufbau von Logik-Gattern.

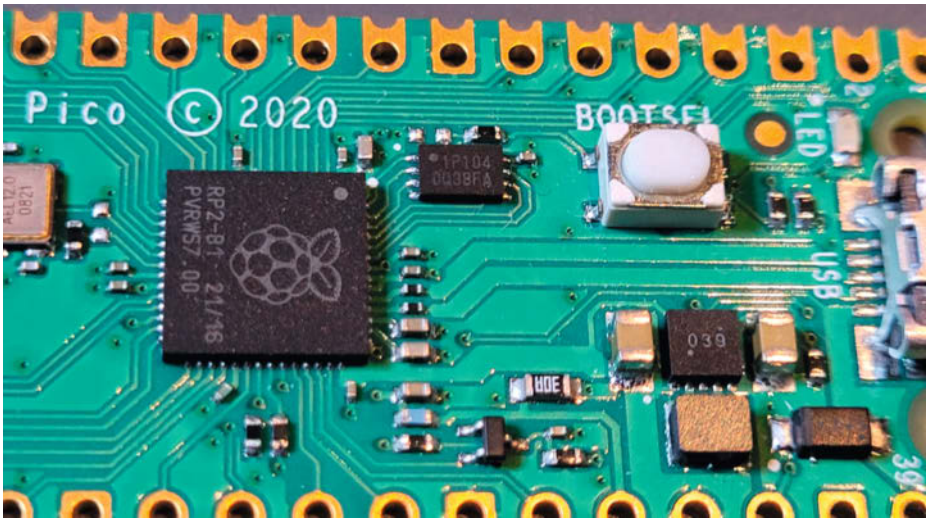


Bild 5: Auch der RP2040 auf dem Pi Pico nutzt einen externen QSPI-Flash als Befehlsspeicher. Wer einen größeren Speicher bräuchte, kann ihn theoretisch ersetzen. Allerdings muss man dann unter Umständen die Timing-Parameter in der PICO-SDK anpassen.

Programmes brennen respektive installieren, so muss man das EPROM aus seinem Sockel hebeln und den gesamten Inhalt löschen. Dazu beleuchtet man das Fensterchen im EPROM einige Zeit mit UV-Licht, was die Floating Gates wieder entlädt. Anschließend ist das EPROM neu beschreibbar. Natürlich hatte man früher bei der Entwicklung immer mehrere EPROMs parat, um nicht jedes Mal den Löschzyklus abwarten zu müssen.

Ein EEPROM benötigt kein spezielles Programmiergerät und lässt sich elektrisch löschen – praktischerweise jedes Bit in ihm einzeln. Sein Aufbau ist aber erheblich komplexer, weshalb klassische EEPROMs langsamer und teurer sind. In der Regel speichert man in ihnen nur kleinere Datenmengen und selten Programme. Für Seriennummern oder MAC-Adressen gibt es auf dem ESP32 und den SoCs der Raspberry Foundation einmalig programmierbaren Speicher (One Time Programmable, OTP). Dort werden gezielt elektronische Sicherungen (eFuse) durchgebrannt. Dazu liefern die Hersteller spezielle APIs, um den OTP im laufenden Betrieb zu programmieren.

Prinzipiell fallen unter den Begriff des EEPROMs auch Flash-Speicher, die mittlerweile so ziemlich alle anderen Arten von nichtflüchtigen Speichern verdrängt haben. Man findet sie in USB-Sticks, auf SSDs und in SoCs von eingebetteten Systemen wie in Telefonen, Routern, Druckern, TVs, Smartphones und Smarthome-Geräten, also eigentlich in so gut wie allem.

UND, ODER NICHT?

Flash-Speicher wurde in den 1980er Jahren von Toshiba entwickelt und diente zunächst als Alternative zu herkömmlichen Massenspeichermedien wie Festplatten und optischen Datenträgern. Die Legende besagt, dass der Begriff Flash-Speicher von einem Toshiba-Entwickler geprägt wurde, weil ihn der schnelle Löschvorgang an das Blitzen eines Kamera-Blitzes erinnerte. Daran lehnte auch der Men-in-Black-Regisseur 1997 den Begriff „Blitzdingsen“ (engl. flashy-thing) zum Löschen menschlicher Erinnerungsspeicher an.

Zum Aufbau von Flash-Speicher gibt es zwei Techniken: NAND und NOR. Beim NAND-Flash sind mehrere Einzel-Speicherzellen (mit Floating-Gate-Transistoren) seriell verschaltet, was an die serielle Anordnung der Transistoren in einem logischen NAND-Gatter erinnert, wenn man dies um 90 Grad kippen würde (Bild 2). NAND-Speicher sind nur halb so groß wie NOR-Speicher, da nicht jede Zelle (Bit) dedizierte Leitungen zum Adressieren und Auslesen der Bits hat. Stattdessen werden NAND-Speicher immer page- bzw. blockweise ausgelesen. Eine Page umfasst zwischen 512 Bytes bei kleineren Speicherchips und 16 kByte bei großen. Mehrere Pages bilden einen Block. Linuxer kennen den Begriff der blockbasierten Geräte (Block Devices), zu denen Festplatten, CD-ROMs und DVD gehören.

Um einen NAND-Flash zu beschreiben, muss man immer mindestens eine komplette Page schreiben. Auch das Auslesen funktioniert nur über komplette Pages. Das Löschen hingegen erfolgt über den Block, in dem die Page liegt. Das macht die Sache aufwändig: Auch wenn man nur ein Byte im NAND ändern will, muss man den gesamten Block einlesen, das eine Byte ändern, den Block im NAND löschen und den Block neu schreiben.

NAND wird häufig in USB-Sticks, Speicherkarten und Solid-State-Drives (SSDs) verwendet. Er bietet schnelle Schreibgeschwindigkeiten, jedoch langsamere Lesezugriffe im Vergleich zu NOR. NAND-Speicher ist aufgrund seines Aufbaus anfälliger für Fehler und Ausfälle, weshalb im Hintergrund noch eine spezielle Hardware arbeitet, die Fehler korrigiert und Schreiboperationen verteilt, Lesern

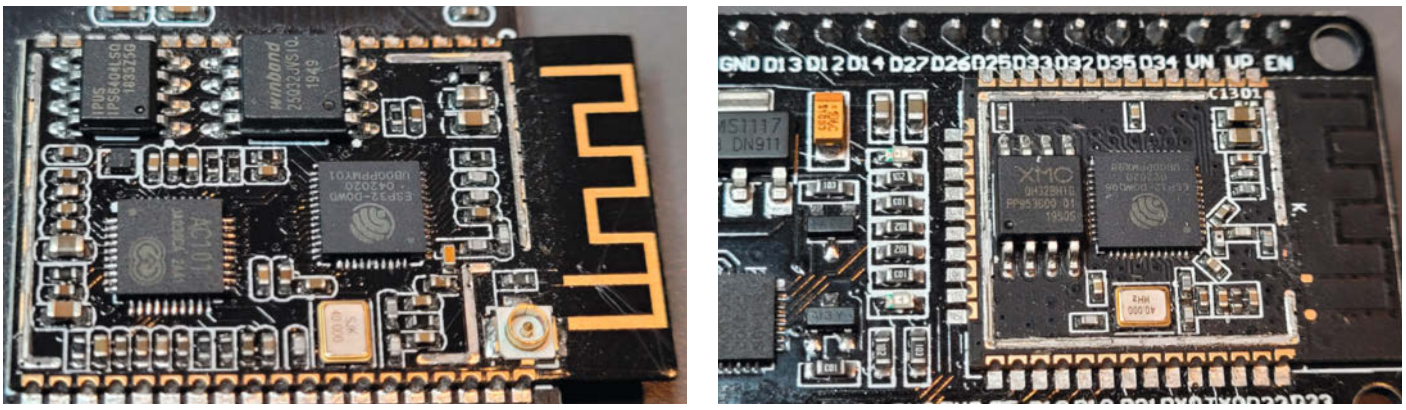


Bild 4: Die meisten ESP32-Devkits sind mit Modulen ausgestattet, in denen ein externer NOR-Flash das Programm enthält. Rechts ein herkömmliches Devkit, links ein Audiokit mit zusätzlichem Audio-IC und RAM-IC. Normalerweise ist der Aufbau unter einer Metallkappe verborgen.

ist das womöglich schon als Wear Leveling vor die Augen gekommen.

Gezielter Zugriff

Beim NOR-Speicher kann man über dedizierte Leitungen beliebige einzelne Bytes adressieren und auslesen. Das ist insbesondere für den Befehlsspeicher von Mikrocontrollern und seinen Program Counter wichtig, der auf den jeweils nächsten Befehl zeigt. Nur mit beliebig wählbaren Adressen sind Sprünge und Verzweigungen (genau genommen Aufrufe) zu anderen Befehlen im Speicher möglich. In diesem Zusammenhang fällt auch der Begriff Execute in Place (XiP), der das Auslesen eines Befehls und seine sofortige Ausführung bedeutet, ohne Umwege über Caches oder RAM. Der ATmega 328 im Arduino hat beispielsweise 32 kByte internen NOR-Flash, aus denen er seine Befehle liest.

Bei NAND-Speicher muss eine CPU zur Befehlsausführung erst die komplette Page in einen RAM kopieren und dann einzelne Bytes auslesen. Genauso macht es auch der PC mit seiner blockbasieren Festplatte oder SSD. Beim Beschreiben und Löschen arbeitet der NOR-Flash jedoch ähnlich wie NAND, nämlich blockbasiert – allerdings erheblich langsamer.

Eine wichtige Gemeinsamkeit haben NOR und NAND und auch die anderen (E)EPROMs: Im Normalzustand sind ihre Zellen immer auf 1 gesetzt. Man kann also keine 1 in eine Speicherzelle schreiben, sondern immer nur eine 0. Beim Löschen werden Zellen von 0 wieder auf 1 gesetzt.

Zufällig beliebig

Auf die Speicherstellen (Bytes) eines Random Access Memory, kurz RAM, kann man sowohl lesend als auch schreibend zugreifen, ohne den Umweg, ganze Pages laden zu müssen. Und das auch noch rasend schnell. Das Random in RAM steht für wahlfreie oder beliebige Adressen und nicht für zufällig – was keinen Sinn ergeben würde, aber womöglich interessant zu beobachten wäre.

Während bei anderen Speicherarten meist der Schreibzugriff mit den einhergehenden Löschungen ein Riesenaufwand ist und zusätzliche Logik erfordert, ist bei RAM lesen und schreiben mehr oder minder gleichwertig. Der große Nachteil von RAM: Sein Inhalt überlebt nur solange, wie eine Versorgungsspannung anliegt.

RAM findet man in Geräten als statischen RAM (SRAM) und dynamischen RAM (DRAM). SRAM ist aus mehreren Transistorzellen aufgebaut (meist sechs), die sogenannte bistabile Kippstufen (engl. Flip Flop) ergeben. Eine gespeicherte 1 bleibt solange darin erhalten, bis man ein 0 hineinschreibt – oder den Strom abschaltet. Man findet diese Speicherart in Mikrocontrollern. Der Arduino Uno mit seinem

Bild 6: Wer sich schon immer gefragt hat, was das alles im Arduino-Board Manager zu bedeuten hat, dürfte nach der Lektüre dieses Artikels die Antworten haben.

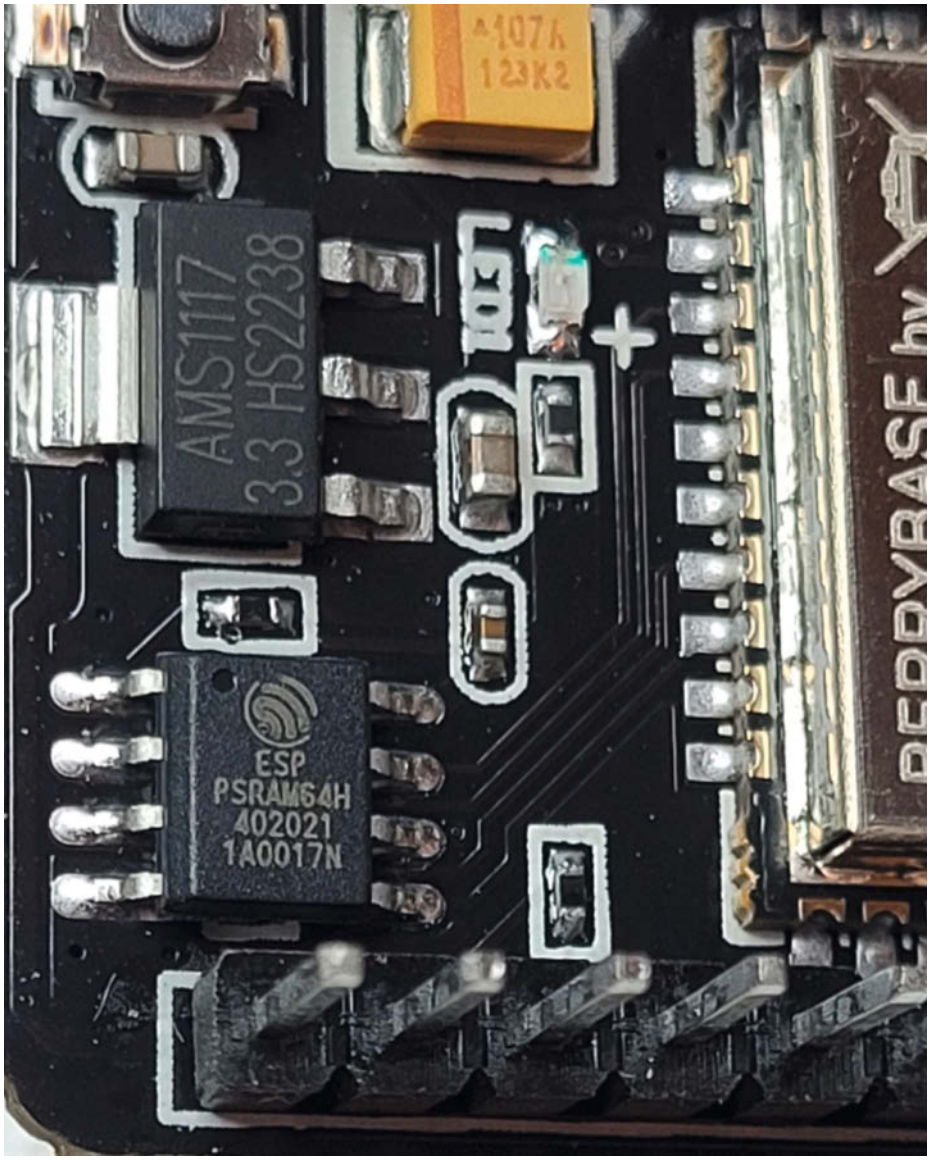
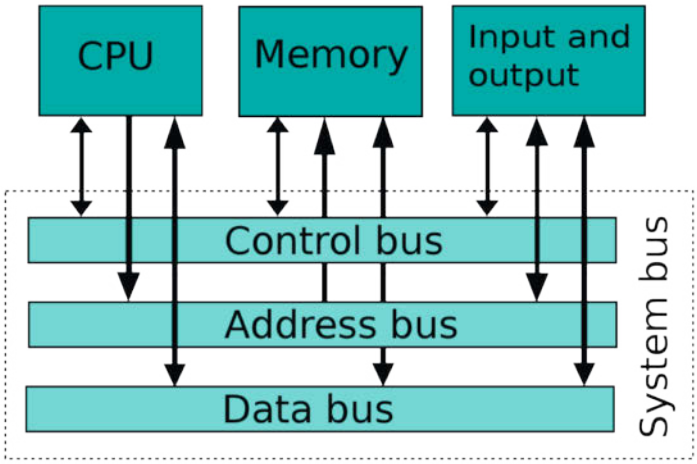


Bild 7: Die ESP32-CAM bringt zusätzlichen RAM in Form eines seriellen ICs mit (links unten).



Wikipedia

Bild 8: Die CPU wickelt den Transport von Daten zwischen sich, Speicher und I/O über Bus-Systeme ab.

ATmega 328 hat beispielsweise 2 kByte SRAM, der RP2040 immerhin 264 kByte, der ESP32 sogar 520 kByte.

DRAM verbraucht mit einem Transistor und einem Kondensator erheblich weniger Platz auf einem Silizium-Die. Eine Spannung auf dem Kondensator bedeutet eine 1, keine Spannung eine 0. Leider muss die Ladung des Kondensators regelmäßig aufgefrischt werden, weil der sich sonst selbstständig entlädt. Eine integrierte Hardware sorgt dafür, dass die Ladung regelmäßig aufgefrischt wird – aber auch nur, bis man den Strom abschaltet.

DRAM ist erheblich günstiger in der Herstellung und wird überall da eingesetzt, wo viel RAM vonnöten ist, etwa PCs oder Spielekonsolen. Allerdings ist DRAM um einiges langsamer als SRAM. Hinweis am Rande: Der Hersteller Espressif verwendet den Begriff DRAM etwas anders: Das interne SRAM lässt sich in Bereiche für Befehle (Instruction RAM, IRAM) und Daten aufteilen. DRAM steht hier für Data RAM.

Cache

Modernen CPUs ist eine besondere Art des SRAM zu Seite gestellt: der Cache. Dorthin kopiert die CPU ganze Programmsequenzen aus dem DRAM, um Befehle schneller lesen und abarbeiten zu können. ARM, Intel und AMD haben bis zu drei Ebenen von Caches in ihre Prozessoren integriert, um Programmabläufe zu beschleunigen. Um immer den richtigen Code im Cache parat zu haben, gibt es aufwändige Techniken zur Vorhersage, welchen Abzweig der Prozessor als Nächstes nehmen wird. In der Regel kommt man damit beim Programmieren jedoch nicht in Berührung.

Die ESP32-Prozessoren haben ebenfalls einen kleinen Cache (32 kByte und mehr), in dem Teile des Codes beim Lesen aus dem langsameren (NOR-)Flash zwischengespeichert werden. Es ist beim ESP32 sogar möglich, Teile des großen internen SRAM als Befehlsspeicher zu nutzen, indem man Code-Sequenzen der Anwendung aus dem Flash dorthin kopiert und danach vom SRAM schneller ausliest. Dann heißt es IRAM. Auch ARM-CPUs wie beim Pi haben Caches, ATmega jedoch nicht.

Drinnen und Draußen

Heutzutage bringen viele Mikrocontroller alle zum Betrieb wichtigen Komponenten wie CPU, RAM, Flash und sogar EEPROM gleich auf einem Chip in einem Gehäuse mit. Man muss nicht mal mehr einen Quarz zur Takterzeugung anschließen. Die ATmegas und PICs von Microchip sind genauso konzipiert. Einzige größere Hürde sind die speziellen Programmiergeräte, um die „nackten" ICs zu programmieren. Seit dem Arduino Uno ist sogar das weggefallen: Dem ATmega wird ein spezieller Bootloader per Programmiergerät eingeflasht. Dieser Bootloader übernimmt die Kommunikation über seine serielle Schnittstelle (USB-zu-seriell) mit der Außenwelt. Beim Start des Mikrocontrollers bzw. Reset nimmt er kurzzeitig Befehle entgegen, mit denen man ihm ein Programm senden kann. Dieses schreibt er im laufenden Betrieb in den Flash und startet es.

Mit der „Alles drin"-Lösung fährt man aber nicht immer gut. Oft wird der Speicher knapp, weil das Programm mit der Entwicklungszeit an Funktionsumfang dazu gewinnt. Bei den ATmegas muss man dann auf ein größeres Modell wechseln. Das wäre dann beispielsweise der ATmega1284, der aber gleich mehr Beinchen hat und vermutlich nicht mehr in die vorhandene Schaltung passt.

Nachrüstbar

Espressif geht einen anderen Weg. In vielen seiner IC-Varianten (ESP32, S1, S2, S3, C2, C3, C6) ist noch kein Flash-Speicher enthalten. Er muss extern nachgerüstet werden und zwar in Form eines sogenannten SPI-Flash (Bild 3 und 4). Per serieller Schnittstelle werden Adressen an den Flash übertragen, der die Daten seriell zurückliefert. SPI ist bereits in den Espressif-ICs integriert. Mit einer speziellen Logik wird der SPI-Flash in den Adressraum der CPU eingeblendet, sodass man sich als Programmierer nicht um das Abholen der Befehle per SPI kümmern muss.

Der serielle NOR-Flash des Pi Pico nimmt 16 MBit auf, also 2 MByte. Sein SPI-Bus kann mit maximal 133 MHz getaktet werden (Bild 5). Bei 4 Bits parallel kommt man auf einen theoretischen Datenstrom von 66 MByte/s.

SPI benötigt vier Leitungen: Clock (CLK), Master Out Slave In (MOSI), Master In Slave Out (MISO) und Chip Select (CS), um jeweils ein Bit zu übertragen. Die ESP32-ICs unterstützen aber noch die schnelleren Varianten Dual und Quad SPI mit mehr Leitungen. Bei Letzterem werden vier Bits parallel übertragen. Die Begriffe QIO und QOUT findet man auch in der Arduino IDE im Board Manager (Bild 6) für Übertragungen mit vier Leitungen, DIO und DOUT für zwei Leitungen. Der schnellste Mode ist der standardmäßig eingestellte QIO-Mode.

Auch der Pi Pico mit seinem RP2040 hat dedizierte QSPI-Ports, um seinen externen, 2 MByte großen Flash (W25Q16) anzusteuern (Bild 5). Maximal unterstützt er 16 MByte.

ESP32-Modelle unterstützen zudem zusätzlichen externen RAM, der ebenfalls per QSPI angesteuert und nahtlos in den virtuellen Adressraum eingeblendet wird. Dabei kommt meist von Espressif zertifizierter Pseudostatic RAM (PSRAM) zum Einsatz. Pseudostatic RAM heißt er, weil er intern als DRAM aufgebaut ist, aber von außen wie ein SRAM wirkt. Auf vielen ESP32-CAMs ist PSRAM mit 8 MByte Speicher verbaut, obwohl der ESP32 nur 4 MByte davon adressieren kann (Bild 7). Über die HIMEM API von Espressif kann man aber zwischen Speicherbänken umschalten, sodass der Zugriff auf die „höheren" 4 MByte möglich ist. Das PSRAM arbeitet mit maximal 133 MHz.

Ansprache

Wenn nun alle wichtigen Speicherarten erklärt sind, wie hängt dann nun alles zusammen? Im Kern geht es bei digitalen Systemen immer darum, eine CPU mit einer Folge von Befehlen zu füttern, um irgendeine Aufgabe zu erledigen. Das ist im PC ebenso wie im kompakten Mikrocontroller, in dem CPU, Flash und RAM gleich in einem Gehäuse untergebracht sind. Um etwa eine profane Aufgabe wie das Blinken einer LED zu bewerkstelligen, benötigt ein ATmega (wie im Arduino) einen Befehlsspeicher. Daraus ruft die CPU die Befehle nacheinander ab und führt sie aus. Aber wie kommt sie an die Befehle?

Dazu legt die CPU die Adresse der Speicherstelle des Befehls auf ihren Adressbus (Bild 8). Welche die richtige Speicherstelle ist, weiß der interne Program Counter. An den Adressbus

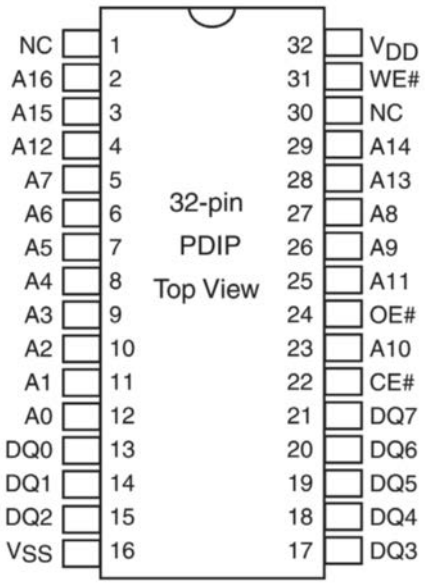


Bild 9: Ein RAM-Baustein mit statischem Speicher im großen DIL/DIP-Gehäuse

ist der NOR-Flash angeschlossen, in Bild 9 ist ein dedizierter IC in einem DIL/DIP-Gehäuse mit 128 kByte abgebildet. Das ist nur ein Serviervorschlag, in ATmegas ist der natürlich so nicht verbaut, aber man bekommt einen Eindruck von den Anschlüssen respektive Schnittstellen.

Über die Leitungen A0 bis A16 adressiert der Program Counter die Speicherstelle. Soll es etwa Speicherstelle 66666 sein, so muss er an seine Adressleitungen folgendes Bitmuster anlegen: 10000010001101010 (ganz links A16, ganz rechts A0).

An den Datenausgängen des ICs DQ0 bis DQ7 erscheint darauf das Byte der gewünschten Speicherstelle als Bitmuster. Das ist der Lesevorgang. Die Kontroll-Leitungen WE#, OE#, CE# steuern, ob die Daten-Pins des ICs Ein- oder Ausgang oder hochohmig sein sollen. Im Prinzip macht sich das IC dann quasi unsichtbar.

Das gelesene Datenbyte landet vereinfacht gesagt im Instruction Register der CPU und wird dekodiert und ausgeführt. Anschließend wird der Programm Counter erhöht und das nächste Byte aus dem Flash gelesen.

Analog dazu wird auch RAM über Adress- Daten- und Kontrollleitungen gesteuert. Will die CPU Daten temporär in einem RAM speichern, etwa berechnete Werte in Variablen, so legt sie an ihren Adressbus die Adresse des gewünschten Speicherortes (den weiß das Programm durch die Initialisierung). Auf den 8 Bit breiten Datenbus legt sie den Wert der Variablen. Über den Kontrollbus (WE, Write Enable) signalisiert sie dem RAM, dass dieser Daten speichern soll. Will die CPU Daten stattdessen aus dem Speicher auslesen, so signalisiert sie dies dem RAM über den Pin OE (Output Enable). Damit sich mehrere Speicherchips am gleichen Adressbus nicht ins Gehege kommen, steuert eine zusätzliche Logik den Pin CE (Chip Enable).

Architekturen

Hängen Datenspeicher (RAM), Befehlsspeicher (ROM; hier der Flash-Speicher) und auch die Peripherieeinheiten wie GPIO, UART etc. am gleichen Adress- und Datenbus (Bild 8), so spricht man von der Von-Neumann-Architektur. Einige kennen das womöglich von älteren Prozessoren wie Z80, 6502/10, 8085, 80486. Aber auch aktuelle x64-kompatible Desktop-Prozessoren von Intel und AMD beruhen darauf. Der Vorteil: Der CPU ist es im Prinzip egal, woher ihre Befehle kommen, sie kann sie sowohl aus dem ROM als auch aus dem RAM lesen. Alle Bausteine und I/Os liegen in einem durchgehenden Adressraum. Logisch kann man sich das wie eine lange Straße vorstellen, in der die ICs an unterschiedlichen Hausnummern wohnen.

Gibt es für RAM und ROM jeweils eigene Busse, spricht man von Harvard-Architektur. Es gibt getrennte Straßen für Befehle und Daten und getrennte Adressräume. Das bietet Geschwindigkeitsvorteile, macht aber die Verwaltung der Adressen unübersichtlich. Reine Harvard-Architekturen findet man nur selten.

Viele CPUs in Mikrocontrollern arbeiten nach dem Modified-Harvard-Prinzip, was die strikte interne Trennung von Befehls- und Datenspeicher aufhebt. Allein die Integration von Caches macht dies erforderlich. Hinzu kommt, dass die zwei getrennten Straßen der puren Harvard-Architektur durch ein Speichermanagement zusammengeführt werden, sodass es für den Programmierer wie zwei oder mehr hintereinander liegende Straßen aussieht (Unified Memory). Die ATmegas sind unter anderem Vertreter dieser Architektur, ebenso die LX6/7 in den ESPs sowie einige ARM-CPUs. Auch die seriellen Speicher SPI-Flash und PSRAM werden über spezielle Logiken in die Adressräume eingeblendet, sodass es für die CPU transparent ist.

Darüber, ob ein System eine Harvard-, von-Neumann oder Modified-Harvard-Architektur ist, gibt es in Foren immer wieder Streit. Die Unterscheidung ist aber meist nur dann relevant, wenn man sich aus Gründen beim Programmieren mit den internen Bussen beschäftigen muss.

Ausblick

Mit der Übersicht über die verschiedenen Speicherarten lassen sich Angaben zur Leistungsfähigkeit von Mikrocontrollern besser einordnen. Auf dieser Grundlage erklären wir in einer der nächsten Ausgaben, wie Mikrocontroller mit ihren Speichern umgehen, wo Variablen und Konstanten gespeichert sind und was Heap und Stack bedeuten. —*dab*

Die neue 40-Watt-Klasse

Aktuelle Diodenlaser bieten mit 40 Watt Laserleistung so viel Power wie ihre CO_2-Kollegen. Aber sind sie eine wirkliche Alternative? Wir haben den Creality Falcon 2 getestet.

von Johannes Börnsen

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

Bild 1: Der ausladende CO_2-Laser in unserer Werkstatt verbraucht dauerhaft viel Platz.

Bisher war die Welt der Hobby-Laserschneider und -gravierer recht eindeutig in zwei Geräteklassen aufgeteilt: Auf der einen Seite CO_2-Laser mit reichlich Leistung, bei denen der Laserstrahl in einer namensgebenden CO_2-Röhre erzeugt wird. Bauartbedingt verfügen diese Geräte über eine Wasserkühlung sowie ein geschlossenes Gehäuse. Auf der anderen Seite standen die im Vergleich schwachen Diodenlaser, bei denen das Laserlicht direkt im Schneidkopf mittels Halbleiter erzeugt wird. Für die Kühlung reicht daher ein in den Laserkopf integrierter Lüfter. In der Regel haben diese Geräte kein Gehäuse und der Rahmen steht einfach offen auf dem Tisch. Während die Diodenlaser durch den günstigen Anschaffungspreis punkten (einfache Diodenlaser sind bereits für unter 200 Euro zu bekommen) und vergleichsweise platzsparend und mobil sind, eigneten sie sich mit Leistungen von 2 bis 10 Watt eher zum Gravieren als zum Schneiden. Ganz anders dagegen die CO_2-Laser: Beim chinesischen Einsteiger-CO_2-Laser K40 für rund 400 Euro stehen bereits 40 Watt als Laserleistung zur Verfügung – genug, um mehrere Millimeter starkes Holz in einem Durchgang zu schneiden.

Dieses Gefüge hat sich nun verschoben: Mehrere Hersteller bieten mittlerweile Diodenlaser mit 22 Watt, 40 Watt oder sogar 44 Watt Laserleistung an, genug um viele Aufgaben ihrer CO_2-Kollegen zu übernehmen. Dabei wird das Licht mehrerer Lasermodule direkt im Schneidkopf zu einem gemeinsamen Laserstrahl gebündelt. Die Schneidköpfe sind dadurch etwas gewachsen. Nach wie vor liefern die Hersteller diese Geräte aber in der Regel ohne Gehäuse aus.

Gestochen scharfer Vogel

Für den Vergleich eines solchen Diodenlasers mit einem ebenbürtigen CO_2-Laser haben wir uns einen Creality Falcon 2 in die Redaktion geholt. Gegen ihn tritt ein Noname-CO_2-Gerät an (Bild 1), das wir bereits seit einigen Jahren in der Redaktion für diverse Projekte im Einsatz haben.

Der Creality Falcon 2 kommt mit fertig montiertem Rahmen. Lediglich der Air Assist (ein kleiner Kompressor, der Schmauchspuren vorbeugen soll) muss noch angeschlossen werden. Neben einer Schutzbrille ohne Zertifikat sind diverse Werkzeuge, etwas Sperrholz für erste Proben, eine Fokushilfe und zusätzliche Füße im Lieferumfang enthalten. Diese legen den Laser um einige Zentimeter höher,

Bild 2: Beim Falcon 2 lässt sich der Schneidkopf in der Höhe verstellen und mit Schrauben arretieren.

Bild 3: Die vom Falcon 2 geschnittene Oberfläche (links) ist frei von Schmauchspuren, der Redaktions-CO_2-Laser (rechts) hat dagegen deutliche Rückstände hinterlassen.

Bild 4: Der Air Assist ist mit dem Schneidkopf verbunden und verringert Schmauchspuren.

Bild 5: Auch bei diesen Versuchen mit Leder sieht man Schmauchspuren auf den Werkstücken des CO_2-Lasers (rechts).

damit man dicke Materialien bearbeiten kann. Eine Schneidunterlage (Honeycomb) und ein Gehäuse gehören dagegen nicht zum Lieferumfang, sondern müssen zusätzlich erworben werden.

Der Falcon 2 ist in einer 22-Watt- und einer 40-Watt-Ausführung zu bekommen. Das spiegelt sich natürlich auch im Preis wider: Stand Januar 2024 schlägt das 22-Watt-Gerät mit 770 Euro zu Buche, das 40-Watt-Gerät ist mit 1600 Euro mehr als doppelt so teuer. Außerdem ist noch eine 12-Watt-Version erhältlich, die mit 620 Euro aber nur wenig günstiger als die 22-Watt-Version und mit einem schlechteren Preis-Leistungsverhältnis eher uninteressant ist.

Die für unseren Test genutzte 40-Watt-Maschine lässt sich via Knopfdruck am Laserkopf auf die Hälfte der Leistung herunterschalten, was einen Teil der Laserdioden deaktiviert. Creality verspricht in diesem Modus eine höhere Präzision durch einen dünneren Laserstrahl. In unseren Tests konnten wir jedoch kaum einen Unterschied feststellen und haben den Laser daher immer mit allen Dioden betrieben, auch um den Vergleich zu unserem CO_2-Laser zu wahren.

Der Creality Falcon 2 bietet mit 40 × 41,5 cm maximaler Bearbeitungsgröße eine klassentypische Schneid- und Gravurfläche. Günstige CO_2-Laser wie der K40 bieten lediglich 20 × 30 cm. Unser Redaktionslaser kann mit 50 × 70 cm auch größere Werkstücke bearbeiten.

Sowohl bei Dioden- als auch CO_2-Lasern muss man den Fokus des Laserstrahls einstellen. Bei beiden Geräten misst man dazu den korrekten Abstand von Schneidkopf zu Werkstück mit einer kleinen Lehre. Zum Verstellen des Fokuspunktes befindet sich beim Falcon 2 der Kopf in einer Schwalbenschwanz-Führung, in der er hoch- und runtergeschoben und mit zwei Rändelschrauben fixiert werden kann (Bild 2). Bei unserem CO_2-Laser ist dagegen eine motorisierte Höhenverstellung des Wabengitters verbaut, auf das das Werkstück für die Bearbeitung gelegt wird. Die motorisierte Verstellung ist natürlich sehr praktisch, in der Praxis funktioniert jedoch beides problemlos.

Für dicke Materialien lässt sich der Falcon 2 außerdem mit schraubbaren Zusatzfüßen in zwei Stufen um jeweils 28 mm höher legen. Die nach unten geöffnete Bauweise bietet darüber hinaus die Möglichkeit, den Lasercutter direkt auf eine Tischplatte oder den Fußboden zu stellen und diese zu gravieren.

Holz, Leder, Acrylglas

Für einen ersten Test haben wir 3 mm dickes Sperrholz verwendet und einen Lampenschirm ausgeschnitten, jeweils zur Hälfte auf dem Creality Falcon 2 und unserem CO_2-Laser. Beide Geräte haben wir auf 100 Prozent

Bild 6: Das Gehäuse lässt sich wie ein Zelt auf- und abbauen und schützt nicht nur vor dem Laser.

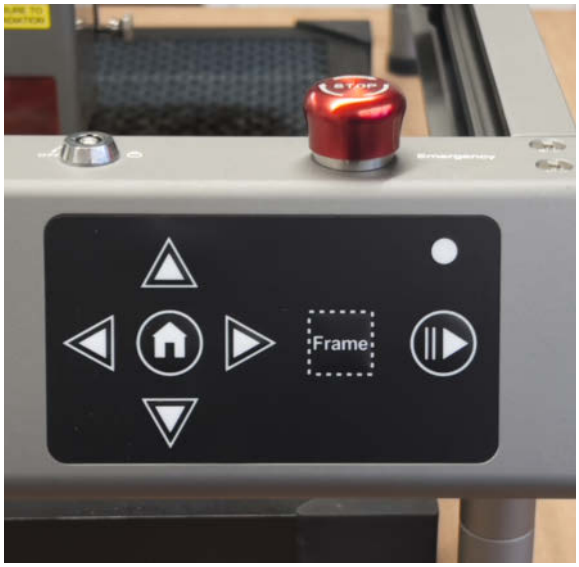


Bild 7: Der rote Ausschalter für Notfälle ist gut erreichbar. Sitzt aber das Zelt auf dem Falcon 2, kommt man nicht mehr ran.

Leistung und mit 600 mm/min laufen lassen. Im direkten Vergleich der Ergebnisse fällt auf, dass der Diodenlaser trotz Bündelung mehrerer Laserdioden einen deutlich feineren Schnitt erzeugt als unser Redaktions-CO_2-Laser. Für einen weiteren Vergleich haben wir außerdem einen Xtool P2 die gleichen Teile schneiden lassen – ebenfalls ein CO_2-Gerät, jedoch deutlich neuer und mit 55 Watt, also etwas leistungsstärker als unser Redaktionslaser. Auch der P2 erzeugt einen breiteren Schnitt als der Diodenlaser.

Außerdem fällt auf, dass der Air Assist des Falcon 2 (Bild 3) ganze Arbeit leistet: Auf der Sperrholzoberfläche sind kaum Schmauchspuren sichtbar, während beide CO_2-Laser deutliche Spuren in Blasrichtung des Air Assist hinterlassen haben. Die Schnittflächen des Falcon 2 sind sauber und sehr fein, insgesamt gefällt uns das Schnittbild gut.

Als nächsten Test haben wir beide Geräte Leder schneiden lassen. Auch hier haben wir wieder die gleichen Einstellungen gewählt: für 1,5 mm starkes Leder 30 Prozent Leistung und 6000 mm/min. Da beim Laserschneiden von Leder gelegentlich feine Fasern stehen bleiben, sind wir insgesamt 5 Durchgänge gefahren. Auch hier sind Schmauchspuren auf der Lederoberseite beim CO_2-Laser zu sehen, aufgrund der Farbe aber deutlich weniger als beim Sperrholz (Bild 5). Erneut macht der Falcon 2 ein gutes Bild, die Schnittflächen sind sauber, auch die kleinen Löcher für die Nähte sind zuverlässig rausgetrennt.

Wenn wir zu Acrylglas (Plexiglas, PMMA) kommen, ist der CO_2-Laser aber klar im Vorteil. Während der Diodenlaser schwarzes Acryl schneiden kann, kann er hellere Farben bestenfalls gravieren. Klares Acrylglas bleibt vom Diodenlaser unbeeindruckt. CO_2-Laser emittieren Licht mit einer Wellenlänge von 10.600 Nanometern, was im unsichtbaren Infrarotbereich des Lichtspektrums liegt. Acrylglas absorbiert Infrarotlicht sehr effizient, daher kann ein CO_2-Laser klares Acrylglas problemlos schneiden und hinterlässt rußfreie Schnittkanten, die aussehen wie poliert.

Diodenlaser hingegen operieren typischerweise im Bereich des sichtbaren Lichts (der Falcon 2 und viele andere bei 455 Nanometern, was im blauen Bereich liegt). Diese Wellenlängen werden von klarem Acrylglas nicht absorbiert und damit auch nicht in Wärme umgewandelt, da das Material sichtbares Licht durchlässt. Ohne ausreichende Absorption kann der Laser das Material nicht effektiv schneiden oder gravieren.

In unserem Test ließ sich dunkelblaues Acrylglas mit einer Stärke von 3 mm bei voller Leistung und 20 Durchgängen entsprechend nicht mit dem Diodenlaser schneiden. Es verformte sich durch die Wärmeeinwirkung lediglich. Das CO_2-Gerät lieferte dagegen nach nur einem Durchgang einen sauberen Schnitt.

Das Problem mit der Sicherheit

Creality bietet für den Falcon ein Gewebegehäuse an, das wie ein Zelt aufgebaut und über das Gerät gestellt wird. Es verhindert, dass reflektiertes Laserlicht (beispielsweise vom Wabengitter) eine Gefahr für das Auge bildet. Um den Laser beladen zu können, lässt sich mit einem Reißverschluss der obere Teil der Umhausung öffnen und nach hinten klappen.

Auch wenn es geht, raten wir in jedem Fall dazu, Diodenlaser nicht offen und ungeschützt zu betreiben. Neben dem Schutz der Augen verringert ein Gehäuse nämlich auch das Problem der giftigen Dämpfe und Gase, die beim Schneiden und Gravieren entstehen. Das Laserzelt von Creality (Bild 6) bietet dazu einen Absauganschluss mit Lüfter, dessen Schlauch man nach draußen führen kann. Bei regelmäßigem Einsatz sollten die entstehenden Dämpfe mit einem Aktivkohlefilter gereinigt werden. Schade ist nur, dass der Ausschalter für Notfälle nicht erreichbar ist, sobald man das Zelt verschlossen hat (Bild 7).

Manche Materialien sollte man sich aber auch mit Filter grundsätzlich verkneifen. PVC bildet beispielsweise Chlorgase und Salzsäure, chromgegerbtes Leder kann Chrom freisetzen, aber auch der Rauch von Sperrholz und dem darin enthaltenen Klebstoff ist nicht nur unangenehm, sondern giftig. Bei CO_2-Lasern stellt sich diese Frage nicht, da sie grundsätzlich über ein festes Gehäuse verfügen.

Eine echte Alternative

Der feine Schnitt des Creality Falcon 2 und der Schmauchspuren effektiv verhindernde Air Assist haben uns im Test sehr gut gefallen. Die Verarbeitungsqualität ist hochwertig, das Zusammenspiel mit der Steuerungssoftware Lightburn (56 Euro, muss separat erworben werden) war problemlos. Gerade in kleineren Werkstätten, in denen man nicht dauerhaft eine Stellfläche für einen Laser reservieren kann, ist der Falcon 2 eine echte Alternative. Lediglich bei Acrylglas sind auch die 40-Watt-Dioden des Falcon 2 machtlos, da führt an einem CO_2-Laser kein Weg vorbei. *—jom*

Kameramodul für Raspberry Pi Pico
mit Personenerkennung

Wir testen ein preiswertes Kameramodul für den Raspberry Pi Pico: Kann es leisten, was der Hersteller verspricht, also die Anwesenheit von Personen im Raum erkennen? Das wäre doch ideal fürs Smarthome.

von Heinz Behling

Zugegeben: Dieses Kamera-Modul ist nicht ganz neu. Jedoch erscheint es interessant genug für Smarthome-Anwendungen, denn es soll etwas können, was sonst nur zu einem erheblich höheren Preis möglich ist: Personen-Detektion.

Bewegungsmelder gibt es viele. Die reagieren auch darauf, wenn eine Person in ihren Sichtbereich tritt, allerdings auch auf Tiere und oft sogar auf eine Änderung des Tageslichts. Insbesondere bemerken sie nicht, wenn eine Person bereits im Raum ist, sich aber nicht oder nur geringfügig bewegt. Wer Beleuchtung und/oder Heizung der Zimmer aber sinnvollerweise abhängig davon einschalten lässt, ob sich auch jemand darin befindet, kommt mit einfachen Bewegungsmeldern nicht aus.

Das Kameramodul Arducam HM01B0 (Bild 1) für den Raspberry Pi Pico und andere Controllerboards mit RP2040-Chip kostet etwas über 20 Euro und nimmt Schwarzweiß-Bilder mit bis zu 320 × 320 Pixel auf. Der Anschluss ist mit 8 Jumperkabeln (liegen dem Modul bei) rasch erledigt (Bild 2).

Für den Pico gibt es auf den Github-Seiten des Herstellers einige Programmbeispiele im Quellcode und auch als fertig kompilierte Binärdateien (mit der Dateiendung uf2), die man unmittelbar auf den Pico kopieren kann, sofern der sich im Laufwerksmodus befindet (beim Verbinden mit dem Computer Boot-Taste gedrückt halten). Allerdings sind nicht alle im Github-Repository vorhandenen Programmbeispiele für dieses Kameramodul geeignet: Einige setzen das Arduino-Mini-Modul voraus, das mit dem Pico über SPI statt I²C kommuniziert. Leider ist das nicht so hundertprozentig gekennzeichnet. So funktioniert die Software zur Gestenerkennung nicht, wohl aber die Personen-Detektion. Einen Link zur entsprechenden Binärdatei finden Sie über den Kurzinfo.

Kurzinfo

» Einfacher Anschluss an RP2040-Boards
» Personen-Detection mithilfe von KI (Tensorflow)
» funktioniert auch über größere Entfernungen

Mehr zum Thema

» Daniel Bachfeld: KI-Komponenten für Maker, Make 6/18, S. 42
» Josef Müller, KI für den ESP32, Make 6/21, S. 48

Alles zum Artikel im Web unter *make-magazin.de/xn9w*

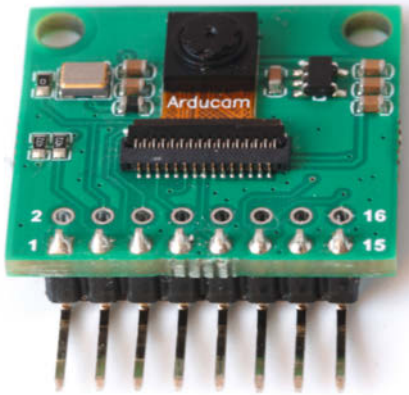


Bild 1: Nur acht Anschlüsse genügen für diese Kamera.

Programmierkenntnisse dringend empfohlen

Auf den Seiten findet man Anleitungen, mit deren Hilfe man unter Linux (oder WSL, siehe Artikel Seite 94) die Programme auch selbst kompilieren kann. Doch stellen Sie sich das nicht so einfach vor: So erfolgt beispielsweise die Aktivierung und Einstellung der On-Board-

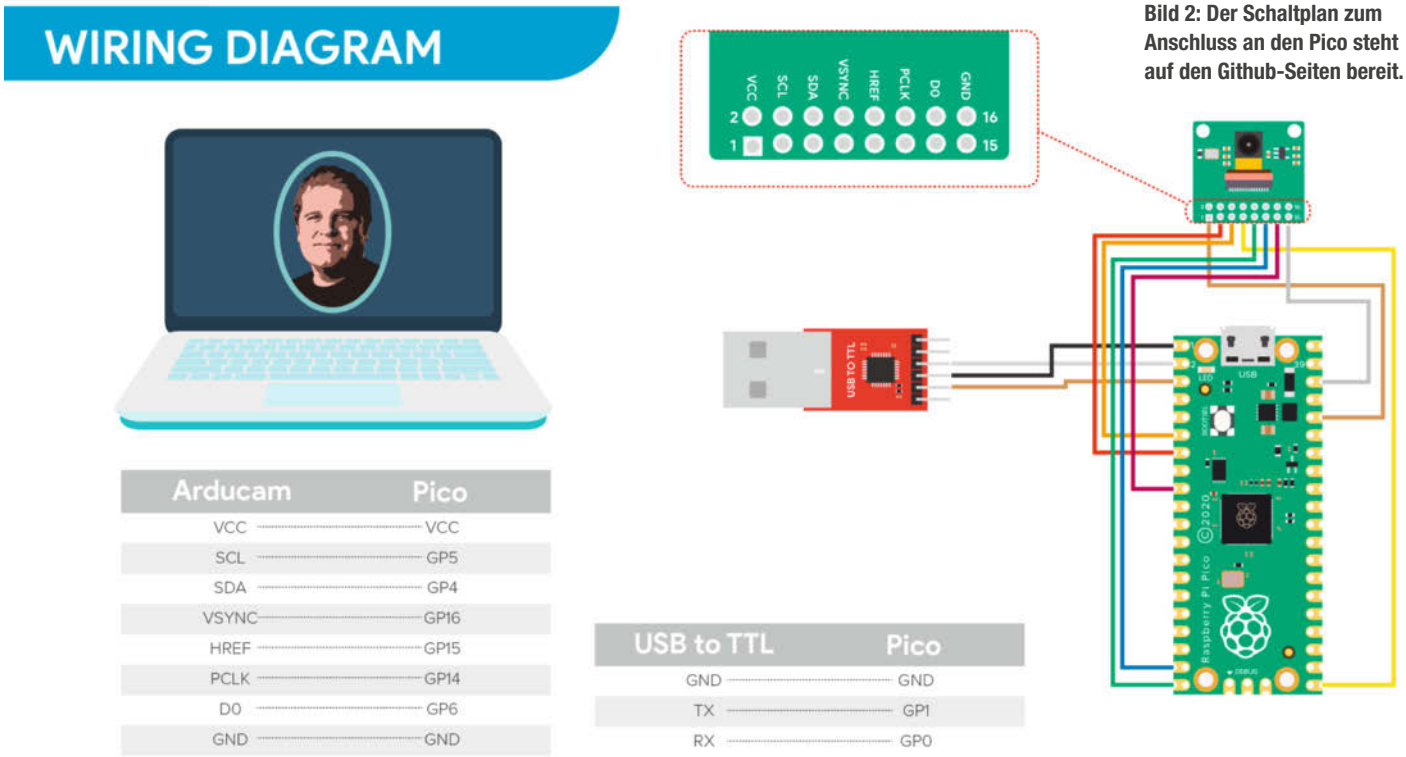


Bild 2: Der Schaltplan zum Anschluss an den Pico steht auf den Github-Seiten bereit.

```
capture_frame took 54898 ticks (54 ms)
Display took 19070 ticks (19 ms)
GetImage took 74636 ticks (74 ms)
Invoke took 864635 ticks (864 ms)
person score:102 no person score -102
**********
```

Bild 3: Der Wert „person score" steht für die Wahrscheinlichkeit, dass sich eine Person im Sichtbereich der Kamera befindet.

Bewegungserkennung (ja, das hat dieses Modul auch), mit deren Hilfe man den im Tiefschlaf befindlichen RP2040 aufwecken kann, durch Beschreiben mehrerer Register (Bild 4) mittels schlecht dokumentierter I²C-Befehle. Eine reizvolle Funktion, weil dann nur das Kameramodul dauernd laufen muss, was bei einem Leistungsbedarf von etwa 1mW lange Akkulaufzeiten ermöglicht. Ebenso per I²C-Befehl geht es bei der Einstellung der Bildwerte wie Helligkeit, Kontrast u. ä. Glücklicherweise arbeitet das Modul aber mit den Standardeinstellungen bereits recht gut.

Das Ergebnis der Personen-Detektion, die mithilfe künstlicher Intelligenz (Tensorflow) erfolgt, wird über den USB-Anschluss ausgegeben. Am einfachsten testet man das mit dem seriellen Monitor der Arduino-IDE (Bild 3). Als `person score` wird hier ein Wert ausgegeben, der ohne erkannte Person in der Nähe von -100 liegt. Wird eine Person bemerkt, steigt der Wert auf deutlich über 0, meist liegt er dann zwischen 80 und knapp über 100. Der Wert wird etwa einmal pro Sekunde ausgegeben.

Diesen Wert kann man dann für entsprechende Smarthome-Anwendungen verwenden, indem man ihn beispielsweise an einen MQTT-Server schickt. Die Software muss man dann aber selbst erweitern. Fertige Bibliotheken etwa für ESPHome/Home Assistant stehen leider (noch?) nicht zur Verfügung.

Die Erkennungsrate ist erstaunlich zuverlässig, sogar bei schlechter Beleuchtung. Selbst über Entfernungen von mehr als 5m erkannte der Testaufbau Personen sicher im Raum. Insofern ist der Test bestanden.

Bei Dunkelheit jedoch sieht das Infrarotblinde Modul nichts. Von außen ist im winzigen Objektiv der rote Glanz des IR-Sperrfilters zu erkennen. Doch da das Objektiv nur etwa 4 mm Außendurchmesser hat, wurde ein Ausbau des Filters nicht gewagt. Das begrenzt die Anwendung etwas. Wer beispielsweise auch schlafende Personen überwachen möchte, der muss aber wohl oder übel das Objektiv modifizieren.

9.2 Sensor mode control

Address	Byte	Register name	Type	Description	CMU	Default (HEX)
0x0100	[2:0]	MODE_SELECT	RW	[2:0] Sensor mode selection 000: Standby 001: Streaming 011: Streaming 2 **(Output N frames)** 101: Streaming 3 **(Hardware Trigger)**	-	0x00
0x0101	[1:0]	IMAGE_ORIENTATIO N	RW	Image Orientation [0]:Horizontal Mirror En [1]:Vertical Flip Enable	Y	0x00
0x0103	[0]	SW_RESET	W	software reset	-	0xFF
0x0104	[0]	GRP_PARAM_HOLD	W	Group parameter hold 0 – consume 1 – hold	-	0xFF

9.3 Sensor exposure gain control

Address	Byte	Register name	Type	Description	CMU	Default (HEX)
0x0202	[7:0]	INTEGRATION_H [15:8]	RW	Coarse integration time in lines **(16-bit UINT)**	Y	0x01
0x0203	[7:0]	INTEGRATION_L [7:0]	RW		Y	0x08
0x0205	[6:4]	ANALOG_ GAIN	RW	Analog Global Gain code **(8-bit UINT)**	Y	0x00
0x020E	[1:0]	DIGITAL_ GAIN_H [9:8]	RW	Digital Global Gain code **(8-bit UINT)**	Y	0x01
0x020F	[7:2]	DIGITAL_ GAIN_L [7:2]	RW		Y	0x00

Bild 4: Wer die Kamera genauer steuern möchte, muss sich die Befehle aus sieben Seiten Registerübersichten heraussuchen.

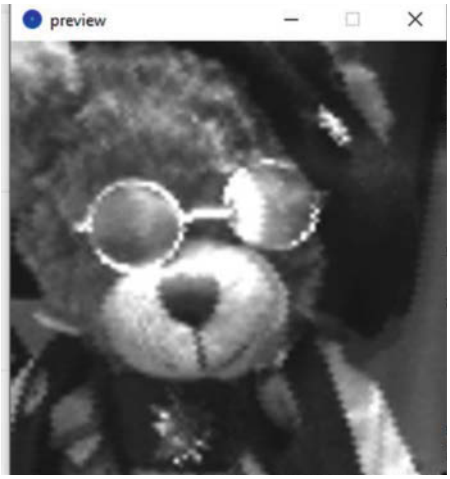


Bild 5: Die Bildqualität der Kamera ist nicht berauschend. Sie reicht aber völlig aus, um auch über mehrere Meter Entfernung zu erkennen, ob eine Person im Blickfeld ist.

Wer bislang eine Bildausgabe vermisst hat: Auch das kann das Modul über den Pico. Da die Datenmenge aufgrund der geringen Auflösung bescheiden ist, funktioniert das sogar über eine serielle Schnittstelle (im Schaltplan Bild 2 bereits enthalten) oder wahlweise auch über den USB-Anschluss. Allerdings folgt die Datenausgabe keinem gängigen Standard. Zur Anzeigt ist ein Skript auf dem PC bzw. Mac erforderlich, das unter der Processing (eine Entwicklungsumgebung für einen Java-Dialekt) läuft (Link siehe Kurzinfo). Die Bildausgabe ist aber alles andere als ein Genuss (Bild 5): Schwarzweiß und geringe Auflösung haben halt ihren Preis. Für die Personen-Detektion ist die Bildanzeige aber nicht nötig.

Fazit

Das Kameramodul HM01B0 zusammen mit einem Raspberry Pi Pico kann zuverlässig die Anwesenheit einer Person erkennen. Jedoch sollte der Hersteller bei der Dokumentation auf seinen Github-Seiten noch nachbessern. Dazu gehört insbesondere eine genaue Kennzeichnung, welche Programme mit welchem Kameramodul arbeiten. Und solche Bugs wie defekte Links zum Schaltbild des Moduls sollten auch nicht vorkommen.

Wer sich zutraut, ein C-Programm anzupassen und auch vor seitenlangen Referenzlisten für I²C-Registerinhalte nicht zurückschreckt, kommt damit sicher gut zurecht. Einsteigern und in der Programmierung Unerfahrenen ist aber eher davon abzuraten. *—hgb*

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 Rasperry 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

FLSUN S1

3D-Drucker mit 1200mm/s und Radarüberwachung

Mit bis zu 1200 mm/s soll der 3D-Drucker S1 des chinesischen Herstellers FLSUN arbeiten. Der Deltadrucker befindet sich zurzeit in der Vorverkaufsphase. Ab Ende Februar soll er in Europa ausgeliefert werden.

Von der Ausstattung bringt der S1 alles mit, um mit solchem Tempo in guter Qualität drucken zu können. Die 39 kg des Vollmetall-Rahmens zusammen mit der Vibration Compensation per Firmware dürften genügen, um den Printer auch bei der maximalen Beschleunigung von 40000 mm/s² schwingungsarm zu halten. Die Closed-Loop-Schrittmotoren melden an die Druckerelektronik (Quadcore A7 mit 1,5 GHz und 16 GB EMMC), wie viele Schritte sie tatsächlich ausgeführt haben. So können Schrittauslassungen vermieden werden.

Der Druckraum hat einen Durchmesser von 320 mm bei einer Höhe von 430 mm. Die Druckbett-Temperatur reicht bis zu 120 °C. Das Bett hat zwei Heizzonen: Eine innere mit 220 mm Durchmesser und eine äußere, die von 220 bis 320 mm Durchmesser reicht. Das 80-W-Hotend erreicht eine maximale Temperatur von 360 °C. Filamente werden in der im Drucker eingebauten beheizten Trocknerkammer von Wassereinschlüssen befreit. Ein Luftfilter entfernt Schadstoffe aus der Luft. Der Druckvorgang wird durch KI mittels HD-Kamera und Lidar (vom Hersteller fälschlicherweise als Radar bezeichnet) überwacht. Das warnt bei Ablösungen und stoppt das Gerät.

Der T1 wird zur Zeit (im Vorverkauf) für 1299 US-Dollar angeboten. *—hgb*

Hersteller	FLSUN
URL	flsun3d.com
Preis	1299 US-$

Playtastic Roboterbausatz

Spielzeug-Roboter mit Bluetooth und App für Programmierung

Ausprobiert —von Make:—

Pearls neuester Fahrroboter kommt als Bausatz mit 52 Teilen und verspricht Spaß am Basteln, Schrauben und Tüfteln. Der Roboter bringt 2 Motoren, ein per LEDs animiertes Gesicht mit Augen und Mund sowie einen Greifarm mit. Der Aufbau gestaltete sich ähnlich simpel wie bei Fischertechnik, nur mit vielen, kleinen Schrauben. Ich (13-jährige Make-Praktikantin) habe ungefähr 45 Minuten dafür gebraucht.

Wenn der Roboter dann fertig aufgebaut ist, gibt es mehrere Möglichkeiten, sich mit ihm zu beschäftigen. Entweder man benutzt die kleinen Knöpfe auf dem Rücken des Roboters oder die kostenlose App, mit der man sich via Bluetooth mit dem Roboter verbinden kann.

Über die Programmierung per App (iOS und Android) kann man ihn in beliebige Richtungen fahren lassen. Mit seinem Lautsprecher spielt er sechs vorprogrammierte Comic-Sounds ab. Zudem kann er acht Gesichtsausdrücke darstellen. Sein Greifarm hält Dinge bis 7 cm Durchmesser fest, etwa Getränkedosen. In der App gibt es mehrere Modi: Entweder man benutzt den Real-Live-Modus, in dem man ihn in Echtzeit steuern kann. Oder man programmiert eine Reihenfolge von Aktionen, welche man dann immer wieder ablaufen lässt.

Der 77 Euro teure Roboter ist simpel aufgebaut und auch seine Bedienung ist leicht zu verstehen. Aber es ist ein großer Spaß, sich damit zu beschäftigen. Ich würde das Aufbauen des Roboters ab etwa neun Jahren empfehlen, da es darunter frustrierend sein kann, wenn eine Schraube nicht auf Anhieb passt oder sogar noch verloren geht.

—Cleo Hohmann/dab

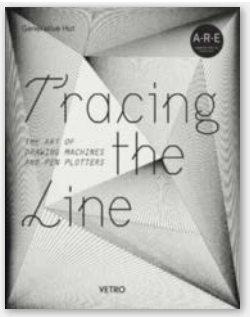
Der Bausatz wurde uns vom Hersteller zur Verfügung gestellt.

Hersteller	Pearl
URL	pearl.de
Preis	76,99 €

Tracing the Line

The Art of Drawing Machines and Pen Plotters

Lange vor Matrix-Nadeldruckern und Tintenstrahlern waren Stiftplotter das Mittel der Wahl, um Computerdaten grafisch auszugeben, etwa als großformatige technische Zeichnungen. Die braucht man in den heutigen Zeiten von 3D-CAD und digitalen Fertigungsmaschinen wie CNC-Fräsen kaum noch, trotzdem erleben Stiftplotter eine Renaissance: als Werkzeug für Kunst. Die knüpft dabei direkt an die Anfänge der Computerkunst überhaupt an, denn erste Werke der damals neuen Richtung entstanden auf Profi-Plottern wie dem Zuse Graphomat Z 64 in den Rechenzentren der 60er Jahre.



Das vorliegende englischsprachige Buch schlägt die Brücke von frühen Pionieren der Plotterkunst wie Georg Nees und Frieder Nake – deren Arbeiten in den einleitenden Überblickstexten kurz gewürdigt werden – zu rund 100 Vertreterinnen und Vertretern der aktuellen Stiftplotter-Szene, denen das Buch jeweils eine Doppelseite widmet. Dort sieht man vor allem fertige Werke, dazwischen aber auch Fotos des jeweiligen technischen Setups; über viele Bilder lassen sich per Augmented-Reality-App fürs Smartphone oder Tablet noch ergänzende Videos legen. Daher kommen viele dieser Doppelseiten praktisch ohne Text aus.

Die meisten Aktiven der Szene arbeiten grafisch und abstrakt, mit dekorativen Ergebnissen. Vom Cover des Buchs sollte man sich nicht täuschen lassen: Nur wenige arbeiten streng in schwarz-weiß und mit einer einheitlichen Linienstärke; die Szene ist bunter, als man zunächst denkt.

Ein inspirierendes und schön gestaltetes Buch, auch wenn die gewählte Schrifttype etwas schwer zu lesen ist: Sie sieht ein wenig so aus, als stamme sie ebenfalls aus einem Stiftplotter. Da die Texte zugunsten der Bilder insgesamt recht kurz sind, fällt das aber weniger ins Gewicht. *–pek*

Herausgeber	Luca Bendandi, Pierre Paslier
Verlag	Vetro Editions
Umfang	240 Seiten
ISBN	978-3-9821664-7-6
Preis	39,90 €

Analogcomputer THAT

Eintauchen in die Welt der analogen Computer

Vinyl-Schallplatten, Nixie-Röhren und Vintage-Computer erfreuen sich nach wie vor großer Beliebtheit. Kein Wunder also, dass auch die analoge Computertechnik wieder aus der Versenkung auftaucht. Aber halt, es gibt noch andere Gründe als Nostalgie: Nicht für alle Probleme ist die Digitaltechnik die beste Wahl, aber Quantencomputer sind für praktische Anwendungen noch in weiter Ferne. So hat sich unter anderem die Firma Anabrid GmbH die Entwicklung eines Analog-Digital-Chips auf die Fahnen geschrieben. Und weil man sowieso schon klassische Analogrechner für Schulen und Universitäten baut, entstand mit THAT, kurz für The Analog Thing, ein relativ günstiger Analaogrechner.

anabrid.com

Sowohl die digitale als auch die analoge Technik haben ihre Stärken. Nur gibt es heute kaum Kurse und Informationen zum analogen Rechnen und wenige haben Hardware zur Verfügung. Aus diesem Grund hat die Anabrid GmbH den THAT entwickelt. Durch dessen modernes Design ist es auch möglich, z.B. einen Arduino anzuschließen. Die Open-Source-Systemarchitektur macht es möglich, den Rechner komplett zu erforschen und zu erweitern.

Der Rechner kommt aus Deutschland und kostet 499 Euro (Bildungseinrichtungen erhalten 10 % Rabatt) plus Versand (DHL in Deutschland). Im Lieferumfang enthalten sind ein USB-Kabel (kein USB-Netzteil), ein Cinch-Cinch-Kabel, Klebefüße, 30 Patchkabel, eine Kurzanleitung und ein Kabel zum Verbinden von THAT-Einheiten. Sinnvolle Erweiterungen sind etwa weitere Patchkabel, ein Oszilloskop (auch Soundkarten, Oszilloskope und einfache Digitaloszilloskope sind verwendbar) und vieles mehr, was auf den Seiten der ausführlichen Dokumentation (online, siehe Link) aufgeführt ist. Man kann aber auch mit dem Basispaket beginnen.

Am Computer stehen Module zum Integrieren, Summieren und Vergleichen, Potentiometer zur Werteingabe, Multiplikatoren und XIRs (Widerstandsnetzwerke) zur Verfügung. Außerdem gibt es Kondensatoren, Dioden (auch Zenerdioden) und ein digitales Voltmeter zur Anzeige der Ergebnisse. —*caw*

▶ make-magazin.de/xqed

Hersteller	Anabrid GmbH
URL	anabrid.com
Preis	499 €

Ausprobiert —von Make:—

DOINOW B1 Pro

Präzisions-Akkuschrauber

Der Akkuschrauber ist für mich ein unentbehrlicher Helfer beim Basteln und Reparieren geworden und mittlerweile besitze ich zwei davon, um vor allem bei Holzarbeiten ohne Bitwechsel vorbohren und schrauben zu können. Wenn es aber mal etwas präziser sein soll, fehlte mir zu meinen Wiha-Feinschraubern ein Akkuschrauber im Kleinformat. Also habe ich mir den DOINOW B1 Pro Präzision zugelegt. Für knapp 50 Euro ist er noch erschwinglich und dazu bekommt man noch 48 hochwertige Bits. Diese passen natürlich auch in andere Bithalter wie von iFixIt (SW 4 mm). Dabei sind fast alle „Sicherheits"-Bits, die man so in Geräten findet, ob Pentalobe oder Triwing.

Was macht den Schrauber so besonders? Ja, er ist schneller, besonders wenn es viele Schrauben sind. Die Abrutschgefahr ist mit etwas Übung geringer als bei Handschraubern. Es gibt eine separat zuschaltbare 3-fach LED-Beleuchtung für dunkle Stellen. Die Drehmomentbegrenzung ist in fünf Stufen einstellbar (0,05/0,08/0,13/0,16/0,2 Nm), besonders wichtig, wenn Schrauben gleichmäßig angezogen werden sollen. Das Getriebe hat es mir bisher auch verziehen, wenn ich für den Motor zu fest angezogene Schrauben kurz von Hand gelöst und dann mit dem Motor weiter „gezogen" habe. Die Grenze würde ich irgendwo bei M3-Schrauben sehen, aber ein PC-Gehäuse kann man noch gut auseinandernehmen.

Das Drehmoment und die LED werden über einen einzigen Taster am Ende des Griffs geschaltet, ansonsten gibt es noch eine Wippe zum Ein- und Ausdrehen der Schraube. Das OLED-Display zeigt die Drehmomentstufe und die Akkuladung als Icon an, ansonsten aber eher wenig wichtige Infos. Der Akku soll 2 bis 3 Stunden Schraubbetrieb ermöglichen, was ich nicht testen konnte.

Der Schrauberkörper und die Transporthülle sind außen aus dickem Aluminium. Die Hülle lässt sich nur mit viel Kraft zusammendrücken und wenn der Einschub eingesteckt ist, kann ich mich darauf stellen. Der Einschub hält die Bits magnetisch und den Schraubendreher durch die Federkraft der Ladekontakte fest und lädt den Schraubendreher über USB-C, egal ob das Gehäuse geschlossen ist oder nicht. Zusätzlich gibt es zwei Öffnungen, mit denen die Bits magnetisiert oder entmagnetisiert werden können. —*caw*

Hersteller	DOINOW
URL	make-magazin.de/xqed
Preis	49,99 €

Tipps und Tricks zu Lightburn

Video aus der Redaktion

Wussten Sie, dass die Verfahrgeschwindigkeiten, mit denen Lightburn die Bearbeitungszeit in der Vorschau berechnet, konfigurierbar sind? Diese lassen sich sogar mit einem Klick von der Lasersteuerung in Lightburn übertragen, die Konfiguration ist nur gut versteckt. Diesen und viele weitere Tipps und Tricks zur Software Lightburn gibt es diese Woche passend zu unseren Artikeln in dieser Ausgabe im Video aus der Make-Redaktion.

Redakteur Johannes Börnsen hat dafür zusammengetragen, was für ihn in der Praxis die größten Arbeitsbeschleuniger waren. Dazu gehört das Einstellen, um wie viele Millimeter Objekte mit den Pfeiltasten verschoben werden und das Anlegen einer Grafikdatenbank mit eigenen, häufig verwendeten Elementen. Das Video zeigt auch, wie man Testmuster schneidet, um möglichst schnell die idealen Lasereinstellungen für das Schneiden und Gravieren mit noch unbekannten Materialien zu finden.

Obwohl viele Laserschneider und Lasergravierer ihre eigenen Programme mitbringen, können die meisten alternativ auch mit Lightburn betrieben werden. Je nach verwendetem Controller sind verschiedene Lizenzen ab 56 Euro erhältlich. Wer einen modernen Faserlaser betreiben möchte, mit dem sich auch Metalle gravieren lassen, muss allerdings 139 Euro einkalkulieren. *—jom*

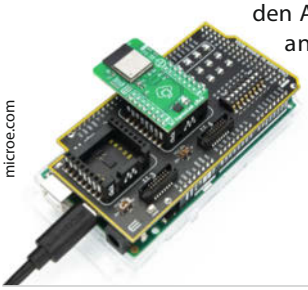
URL youtube.com/@MakeMagazinDE

Click Shield for Arduino Mega

Baukasten für IoT-Profis

Click Shields sind eine Entwicklung der Firma MicroE aus der Tschechischen Republik, die Embedded-Systems entwickelt und vertreibt. Das Click-System vereinfacht die Entwicklung von Systemen durch eine standardisierte Architektur, in die fast alles gesteckt werden kann. Kunden sind vor allem die Industrie (80 %), Hobbyisten und Studenten (je 10 %). Mittlerweile gibt es über 1600 Click-Boards, die praktisch alle Bereiche abdecken. Als Basis für die Boards dienen Click-Shields, die Mikrocontroller tragen. Nun ist auch ein Shield für den beliebten Arduino Mega erhältlich.

Das „Click Shield for Arduino Mega" verfügt über zwei Click-microBUS-Sockel. Damit kann jeweils ein 16-poliges Board der Click-Serie an den Arduino Mega angeschlossen werden. Zusätzlich verfügt das Shield über zwei microBUS Shuttle Ports. Über diese können mit zusätzlichen Adapterboards weitere Click-Boards angeschlossen werden. Diese sind dann aber über ein Kabel verbunden und nicht fest mit dem Shield verbunden. Die Spannung, mit der die Click-Boards betrieben werden, kann für die beiden großen Sockel jeweils mit einem Schalter auf 3,3 oder 5 Volt eingestellt werden. Die Stromversorgung kann über den Arduino oder direkt über USB-C erfolgen.


microe.com

Das Shield selbst verfügt über einen MCP1501-Spannungsreferenzbaustein. Dieser kann aber bei Bedarf mit einem Schalter überbrückt werden, wenn eine externe Referenz verwendet werden soll. Neben den Click-Verbindungen sind auf dem Shield noch 8 programmierbare Schalter und 8 programmierbare LEDs verbaut. Für den ersten microBUS sind zusätzlich Status-LEDs für Power, Transmit, Receive und Pin D13 (LED) vorhanden. *—das*

Hersteller	MicroE Mikroelektronika s.r.o
URL	mikroe.com
Preis	45 US-$

GPIOViewer

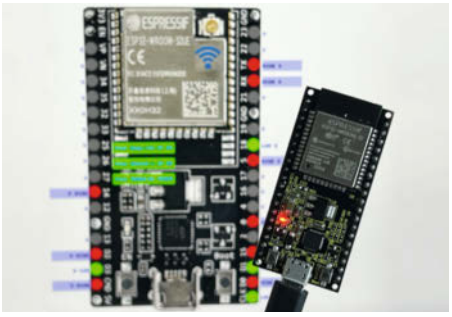
Ausprobiert —von Make:—

Analysetool für ESP32-Projekte

Wenn ein Mikrocontroller-gesteuertes Projekt nicht tut, was es soll, kann man auf den ersten Blick manchmal nicht erkennen, ob es an der Hardware oder an der Firmware liegt. Um Fehler leichter zu identifizieren, hat der GitHub-Nutzer thelastoutpostworkshop daher die Arduino-Bibliothek GPIOViewer für die Arduino IDE 2 entwickelt. Mit ihr kann man in Echtzeit den Status der GPIOs eines ESP32 prüfen. Dafür hostet die Bibliothek einen Webserver auf dem Mikrocontroller, der die GPIO-Aktivität grafisch in einem Browserfenster abbildet.

Um den GPIOViewer in Betrieb zu nehmen, muss man insgesamt drei Bibliotheken in der Arduino IDE installieren. Das GitHub-Repository des Projekts erklärt, welche das sind und wie man es macht. Danach lässt sich das praktische Tool mit wenigen Zeilen in eigene Programme einbinden und bringt auch einen passenden Beispielsketch mit. Hat man diesen auf einen ESP32 übertragen, gibt der serielle Monitor eine IP-Adresse aus, über die man sich mit dem Webserver verbinden kann. Das funktioniert auch auf dem Smartphone oder Tablet.

Dort sieht man dann im 100ms-Takt, was die GPIOs gerade tun, z.B. wenn man mit dem Finger über die Pins fährt. Je nachdem, welches



Board man verwendet, kann man zwischen mehr als zwanzig bekannten ESP32-Boardvarianten oder einem generischen Profil wählen.

Das Analysewerkzeug hat thelastoutpostworkshop ursprünglich für eigene Projekte entwickelt, die mit vielen LEDs und Bildschirmen gespickt sind. Da kann es schnell passieren, dass eine LED verkehrt herum verbaut ist und nicht leuchtet, obwohl das Programm fehlerfrei funktioniert. *—akf*

URL https://github.com/thelastoutpostworkshop/gpio_viewer

NeoRuler

Smartes Messinstrument

Ausprobiert —von Make:—

HOZO Design

Der NeoRuler ist ein elektronisches Lineal mit einer Nutzlänge von etwas über 30 Zentimetern, das die aktuell gemessene Strecke auf einem Display anzeigt und auf Wunsch gleich maßstäblich umrechnet. Dafür stehen vorgefertigt über 90 metrische Maßstäbe und solche aus der Welt von Fuß und Zoll zur Wahl – von 1:1 bis zu atlastauglichen Skalen. Alternativ misst man auf dem Plan eine bekannte Entfernung und gibt dann ein, welchem realen Maß das entsprechen soll.

Ist der Maßstab definiert, benutzt man das Gerät fast wie ein gewöhnliches Lineal: Man führt einen kleinen Schieber ganz nach links für den Nullpunkt der Messung und zieht ihn dann zum gewünschten Endpunkt. Oder man setzt den Nullpunkt per Knopfdruck auf die aktuelle Schieberposition.

Die schicke Skala aus blauen LEDs zeigt dynamisch zum Maßstab passende Unterteilungen an, ist beim genauen Hinsehen aber eine Enttäuschung, weil ihr Raster mit 1 mm Abstand zu grob ist. Eine Unterteilung in Zoll (25,4 mm) ergibt bereits unterschiedliche Abstände von Markierung zu Markierung und ist damit unbrauchbar ungenau. Nützlich hingegen ist die Visualisierung der Strecke zwischen Nullpunkt und aktuellem Messpunkt, gerade bei individuell gewählten Nullpunkten.

Der Hersteller Hozo stellte uns für diesen Test die sogenannte Premium Combo zur Verfügung, die fast das Doppelte des nackten Lineals kostet und noch drei alternative Köpfe für den Schieber umfasst (zwei Stiftführungen und eine Lupe) sowie weitere Teile, um aus dem NeoRuler in wenigen Handgriffen eine digitale Schiebelehre zu machen (siehe Bild). Der praktische Nutzen speziell der letzteren Zutat hält sich unserer Erfahrung nach allerdings in Grenzen: Laut Hersteller liegt die Messgenauigkeit bei lediglich bei 0,1 Millimetern, das kann jede normale digitalen Schieblehre besser. Ein guter Tipp ist aber, den NeoRuler über Bluetooth mit der kostenlosen App namens Meazor für Smartphones und Tablets zu koppeln, denn darüber lässt sich manche Einstellung auf dem Gerät bequemer vornehmen als über das reduzierte Drei-Knopf-Interface auf dem NeoRuler selbst.

So schick das Gerät erst einmal wirkt, es wird in den meisten Maker-Werkstätten verzichtbar sein. Und wer wirklich viel mit Plänen auf Papier in krummen Maßstäben arbeitet, dem reicht in der Regel die günstige Grundversion. Einen ausführlichen Testbericht gibt es für Make- und heise+-Abonnenten online (siehe Link). *–pek*

Das Gerät wurde uns vom Hersteller für den Test zur Verfügung gestellt.

▶ make-magazin.de/xqed

Hersteller	HOZO Design
URL	hozodesign.com
Preis	121,95 € (nur Lineal) bis 234,95 € (Premium Combo)

ENGINEER SS-02

Innovative Entlötpumpe

Ausprobiert —von Make:—

Alle Handentlötpumpen, die ich, seit ich löte, für ein paar Mark gekauft oder im Set bekommen habe, funktionierten eher schlecht als recht und hielten nicht lange, egal ob aus Kunststoff oder Metall. Auch kreative Hilfsmittel wie Heißluftpistole und Entlötlitze konnten oft nicht verhindern, dass die Platine oder Bauteile beschädigt wurden. In der Zeit, in der meine preiswerte Entlötstation aufheizte, hätte ich aber mit einer guten Handentlötpumpe schon einige Bauteile entfernen können.

Die Entlötpumpe ENGINEER SS-02 „Solder Sucker" aus Japan macht da einiges besser: Sie ist komplett aus Aluminium, bis auf die Dichtungen – und den Clou, ein kleines Stück Silikonschlauch (2 mm Innendurchmesser) an der Ansaugdüse. Dieser hält 350 °C (kurzzeitig auch mehr) aus. Das habe ich so noch nicht gesehen, die sonst üblichen Teflondüsen liegen nie perfekt am Board an und ziehen viel Nebenluft, warum ist so ein Stück Schlauch nicht Standard? Ungewöhnlich ist auch die Konstruktion mit dem von allen Seiten drückbaren Kolben, der es im Gegensatz zu herkömmlichen Pumpen erlaubt, den Kolben auch auf den Tisch oder den Oberschenkel zu drücken, ohne einen Krampf im Daumen zu bekommen.

Das schnelle Entlöten im Nachhinein oder wenn man beim Bestücken einen Fehler gemacht hat, ist somit schnell erledigt. Im Lieferumfang ist eine sehr hilfreiche kleine Anleitung in Japanisch und Englisch enthalten, die die Bedienung und Pflege zeigt. Mitgeliefert werden ein paar Zentimeter Ersatzsilikonschlauch, den man auch leicht schräg abschneiden kann, um einen flacheren Winkel zur Platine zu erhalten.

Man sollte übrigens aufpassen, es gibt mittlerweile einige nur wenige Euro billigere Kopien der SS-02. Ich habe mir aus Neugierde mal eine bestellt und der erste Eindruck war gut. Aber die Feder der Kopie war schwächer und nach nur wenigen Entlötungsversuchen sah man schon Verschleiß an der Kolbenachse: Am Gleitlager wurde wohl gespart. Auch der Auslöseknopf ist

sehr kantig und unangenehm zu bedienen. Schwergängig waren auch die Gewinde am Kopf, den man von Zeit zu Zeit zum Reinigen abschrauben muss. Für die geringe Ersparnis lohnt es sich nicht, den Nachbau zu kaufen. *—caw*

Hersteller	Engineer Co., Ltd.
URL	engineer.jp
Preis	etwa 30 €

IMPRESSUM

Make: Nächste Ausgabe erscheint am 5. April 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: Beetlebum, Josh Ellingson, Sean Hallowell, Bernd Heisterkamp, Cleo Hohmann, Kirk Pearson, Tim Rieman, Matthias Rosezky, Kai Schauer, Martin Spendiff, Brandon Bennte Young

Assistenz: Susanne Cölle (suc), Christopher Tränkmann (cht), Martin Triadan (mat)

Leiterin Produktion: Tine „The Rock" Kreye

DTP-Produktion: Martina Bruns, Martin Kreft (Korrektorat)

Art Direction: Martina Bruns (Junior Art Director)

Layout-Konzept: Martina Bruns

Layout: Nicole Wesche

Fotografie und Titelbild: Meike Weiß, 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.

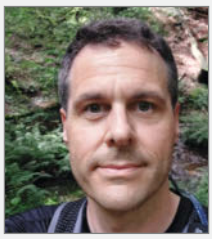
Abo-Service:

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

ISSN 2364-2548

Nachgefragt

Welches PC-Betriebssystem nutzt du bei deinen Projekten am liebsten und warum?



Tim Riemann

Haina, kontrolliert ab S. 66 sein Garagentor per Home Assistant

Meist nutze ich Windows aufgrund seiner Flexibilität und des umfangreichen Softwareangebots. Hier laufen meine Entwicklungsumgebungen und notfalls ist auch schnell eine Linux-Shell über WSL2 gestartet

Ákos Fodor

Redakteur
bei Make

Am liebsten verwende ich macOS. Mir gefällt vor allem, dass die Elemente der Nutzeroberfläche logisch ineinandergreifen und bei einem Update nicht alles über den Haufen geworfen wird, was bereits gut funktioniert.

Daniel Schwabe

Technical Writer
bei Make

Linux Mint ist mittlerweile ein für den GUI-Nutzer freundliches OS, bei dem ich immer das Gefühl habe, dass Sachen einfach funktionieren. Bluetooth-Audio und Druckerinstallationen geschehen mit einem Klick.

Daniel Bachfeld

Chefredakteur
Make

Eigentlich Linux, aber ich arbeite wechselnd mit Windows oder Ubuntu, je nachdem welche Aufgaben gerade zu erledigen sind. Manches funktioniert in der Bash besser, manches auf dem Windows-Desktop.

All-AMD-Gamingnotebook

TUXEDO Sirius 16 - Gen1

Mobilität

Grafikleistung

AMD Power

Hocheffizienter Leichtathlet

TUXEDO Pulse 14 - Gen3

Mobilität

Akkulaufzeit

Linux kompatibel

Bis zu 5 Jahre Garantie

Sofort einsatzbereit

TUXEDO

Gefertigt in Deutschland

Deutscher Datenschutz

Deutscher Tech Support

tuxe.do/make0124