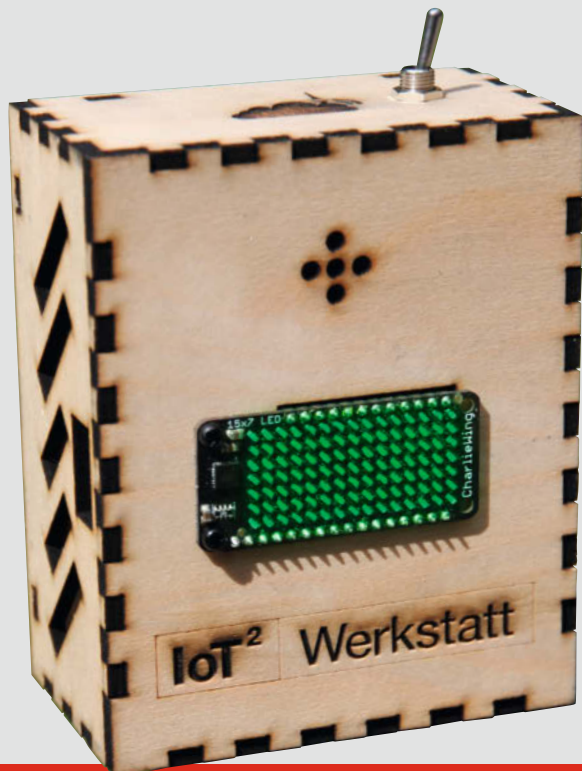




Nixie-Look:
Mini-Displays hinter Glas



CO₂-Ampel gegen Corona

- ▶ Luftqualität messen im Hörsaal, Büro und Klassenzimmer
- ▶ Schritt-für-Schritt-Anleitung für Einsteiger im Heft
- ▶ Ab 40 Euro, in 2 Stunden fertig

DIY-Projekte

- ▶ Pi-hole: Adblocker fürs ganze LAN
- ▶ Zuchtstation für Urzeitkrebse
- ▶ Dias scannen mit Pi und Projektor

Brandneu

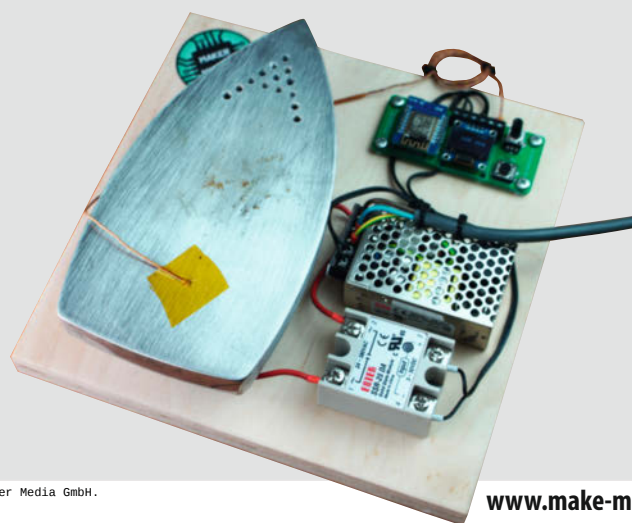
- ▶ Fräst, lasert, druckt: Snapmaker 2.0
- ▶ Einsteiger-3D-Drucker: Prusa Mini
- ▶ Wie früher: Basic-Computer CMM2

Smart Home für die Fritzbox

- ▶ Tür per Telefon öffnen
- ▶ Stromverbrauchsmeldungen per Telegram



SMD-Löten mit dem Bügeleisen

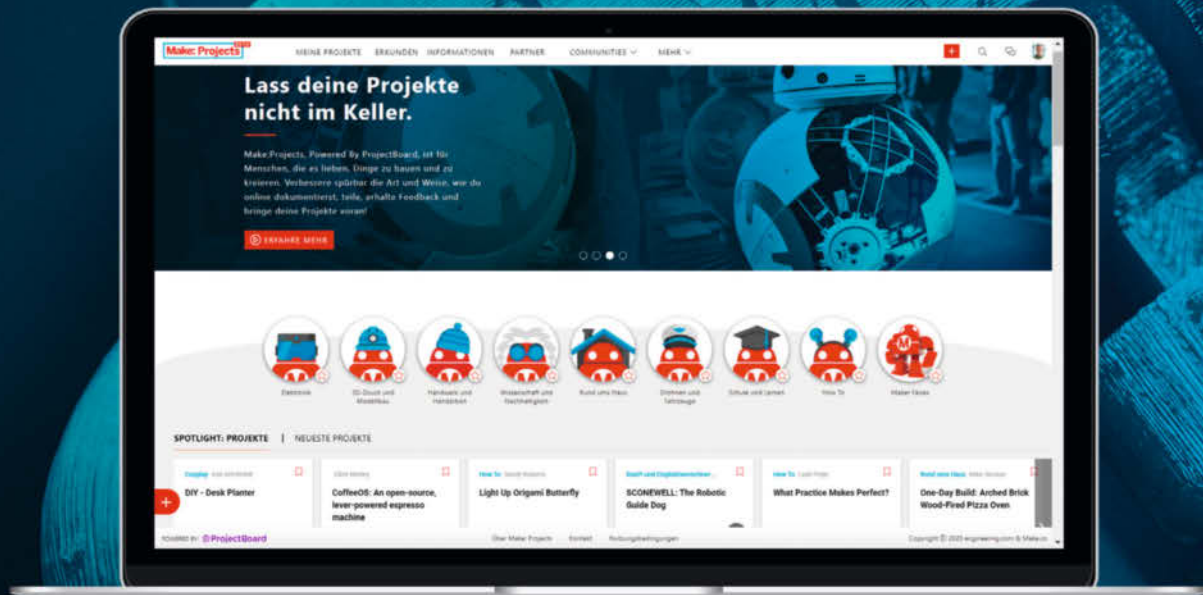


5/20

CH CHF 18,00
AT, Benelux,
IT, ES € 11,90
€ 10,90



Make: Projects



Werde Teil einer weltweiten Maker-Community – online!

Zeige auf der neuen, deutschsprachigen Online-Plattform allen, woran du gerade arbeitest. Teile deine Projektidee, dokumentiere deinen Arbeitsfortschritt und profitiere dabei von dem Feedback anderer Maker. Finde Inspiration für neue Projekte aus vielfältigen Kategorien. Erweitere deine Fähigkeiten durch Schritt-für-Schritt-Anleitungen anderer Maker oder gestalte gemeinsam in Projektteams etwas völlig Neues – egal, wo auf der Welt ihr euch befindet.

STARTE JETZT DEIN PROJEKT!

www.makeprojects.com/de

© Copyright by Maker Media GmbH



Empathie im Neuland

Was habe ich 2013 gelacht, als Kanzlerin Merkel ihren mittlerweile legendären „Das Internet ist für uns alle Neuland“-Brüller anlässlich des Besuchs von Barack Obama raushaute. Vor kurzem hatte ich dann mein eigenes Neuland-Desaster, ausgerechnet vor meinem 13jährigen Sohn. Der spielt nämlich seit einigen Monaten E-Gitarre und saugt sämtliche Infos rund um deren Aufbau, Technik und Geschichte auf – und zwar wie ein Schwamm. Und das erzählt er mir dann.

Pick ups, Single Coils (S), Humbucker (H), P-90, North Start, South Finish, Neck, Bridge, Alnico inklusive der möglichen Konfiguration H-H, H-S, H-S-H usw.: Mir fliegen plötzlich Begriffe um die Ohren, mit denen ich als notorisch Unmusikalischer bislang nichts anzufangen wusste. Falls Sie auch gerade auf dem Schlauch stehen: Es geht um die Tonabnehmer, deren Aufbau und Positionen auf der Gitarre und welchen Einfluss das alles auf den Sound hat.

Gut, als Ingenieur für Nachrichtentechnik verstehe ich natürlich die technischen Grundlagen dahinter und kann es mir vom Sohn erklären lassen. Durch intensives Studium verschiedener Belegungspläne konnte ich ihm sogar dabei helfen, die Humbucker seiner Gitarre auszutauschen, weil die neuen ja einen „High Output“ hätten.

Trotzdem fühle ich mich abgehängt, weil ich auch nach mehrmaligen, ungeduldigen Erklärungen nicht kapiert habe, warum manche Gitarren nur Single Coils oder zwei Humbucker oder beliebige andere Kombinationen für einen bestimmten Klang haben – und warum dann im (Röhren)-Gitarrenverstärker das Signal doch wieder bis zur Unkenntlichkeit verzerrt werden soll.

Was hat das alles mit der Make zu tun? Wir versuchen Sie, liebe Leserinnen und Leser, in jeder Ausgabe ebenfalls an neue Themen und Projekte heranzuführen. Die Redaktion gibt sich große Mühe, sich in ihre Leser hinein zu versetzen, um das richtige Niveau und die richtige Sprache zu finden und die notwendigen Erklärungen und Hinweise zu geben, damit Sie ein Erfolgserlebnis haben und sich nicht abgehängt fühlen. Sollte uns das nicht gelingen, schreiben Sie uns. Aber haben Sie ein wenig Nachsicht, Sie wissen ja, in manche Themen arbeiten wir uns auch erst ein...

Daniel Bachfeld
Daniel Bachfeld

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



CO₂-Ampel gegen Corona

Lüften statt leiden: In Örtlichkeiten mit erhöhter Ansteckungsgefahr ist ein regelmäßiger Luftaustausch das Mittel der Wahl. Schlechte Luft lässt sich leicht über den CO₂-Gehalt in Messwerte fassen – so macht es auch unser Projekt, das zum Öffnen der Fenster mahnt, wenn die Luft allzu dick geworden ist.

10 Der CO₂-Warner für die Schule

Inhalt

Bügeleisen-Lötstation

Sie haben noch ein altes Bügeleisen übrig? Machen Sie daraus eine Reflow-Lötstation für SMD-Platinen – die gelingen damit genauso gut wie mit einem teuren Lötöfen. Unsere Steuer-elektronik kann sogar die geforderten Lötprofile der Bauteile einhalten.

116 Reflow-Löten mit dem Bügeleisen



3	Editorial
6	Leserforum
10	Der CO ₂ -Warner für die Schule
18	3D-Einsteigerdrucker Prusa Mini
20	Fräs-Gravier-3D-Drucker Snapmaker 2.0
24	So geht Heimcomputer im Jahr 2020
30	Werkstattberichte: Termine und Fablab-News
32	Ankündigung: Anleitungen auf Make Projects veröffentlichen
34	Pi-hole: Das schwarze Loch für Werbung
40	Dias scannen mit dem Raspberry Pi
46	Dixies: TFT-Displays im Stil von Nixie-Röhren
54	Was uns inspiriert: Schaltkreis-Skulpturen, Slothbot, gestickte Tastaturen
58	Eigene Knoten für Node-RED
66	Hausgeräte-Überwachung mit Fritzbox und Telegram
72	Fritzbox öffnet die Tür
78	Zuchtstation für Urzeitkrebse
84	Mondphasenanzeige mit ESP32

Brandneu

Bemerkenswerte neue Produkte finden regelmäßig den Weg in unsere Testräume: Etwa der günstige 3D-Drucker Prusa Mini, das 3D-Universalgenie Snapmaker 2.0 oder der rund-erneuerte BASIC-Computer Color Maximite 2, an dem nicht nur Retro-Fans Gefallen finden werden.

- 18 3D-Einsteigerdrucker Prusa Mini
- 20 Fräs-Gravier-3D-Drucker Snapmaker
- 24 So geht Heimcomputer im Jahr 2020



DIY-Projekte

Ein umgebauter alter Diaprojektor könnte Ihre Umkehrfilm-Schätze in das digitale Zeitalter retten, der RasPi-Adblocker Ihr LAN vor überbordender Werbung bewahren und unsere Zuchtstation für Sea Monkeys alte Yps-Gimmicks wiederaufleben lassen – es gibt viel zu tun!

- 34 Pi-hole: Schwarzes Loch für Werbung
- 40 Dias scannen mit dem Raspberry Pi
- 78 Zuchtstation für Urzeitkrebse



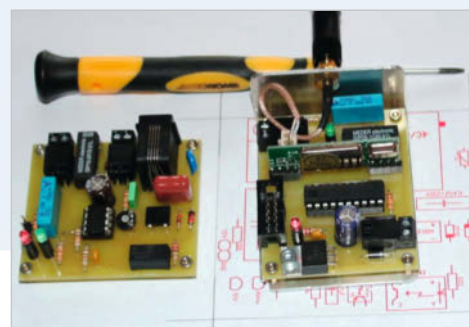
- 90 Reingeschaut: Berührungsloser Seifenspende
- 92 Berührungsloser Musik-Seifenspende-Handwasch-Timer
- 96 Make Family: Mobiltelefone und Mobilfunk
- 102 Reportage: Wearables aus der KOBA Maßschneiderei
- 110 Community-Projekte: Automatische Fahrradschaltung, Binär-Uhr mit Kniwwelino, TV-Lift
- 116 **Reflow-Löten mit dem Bügeleisen**
- 122 Bartop Arcade mit Raspberry Pi, Teil 2
- 130 Tipps&Tricks: Arduino-Pins doppelt nutzen
- 132 Kurzvorstellungen: Neue Anet-Drucker, Seeed Odyssey, Calliope 2.0, ESP32-Audiokit
- 136 Bücher: Stop Motion, Git, Erfinderwerkstatt, Minecraft und Python, 3D Printing
- 138 Impressum

Themen von der Titelseite sind rot gesetzt.

Smart Home mit Fritzbox

Sicher ersetzt die beliebte Fritzbox keine Smart-Home-Zentrale, aber für ein paar Aufgaben kann man sie schon heranziehen: Zusammen mit etwas DIY-Hardware kann sie Hausgeräte überwachen, per DECT-Telefon die Haustürklingel ersetzen und auch gleich die Tür öffnen.

- 66 Hausgeräte-Überwachung mit Fritzbox und Telegram
- 72 Fritzbox öffnet die Tür



Leserforum

Indoktrination

Editorial: Ende der Sklaverei, Make 4/20, S. 3

Natürlich ist es bedauerlich, wenn man wie Sie im MINT-Bereich arbeitet, und trotzdem Schwierigkeiten hat, einfache Konzepte wie die Relation Begriffsbedeutung-Kontext praktisch anzuwenden. Ich sehe keinen Grund, diesen Mangel auch noch zu veröffentlichen. Es gibt gute, anerkannte wissenschaftliche Praktiken, um damit fertig zu werden :-). Über krude Rassismustheorien, Gender, Identitäre Politik, Quote bin ich zwangsläufig ausreichend informiert, und halte das für unwissenschaftlich bis abwegig bis schwachsinnig.

Ich persönlich möchte diesen Mist in einer von mir mitfinanzierten technischen Zeitschrift, wenn überhaupt, ausschließlich nur dann vorfinden, wenn im Artikel eine fundierte wissenschaftliche Auseinandersetzung stattfindet. Halbideologische persönliche Meinungsäußerungen, auch in Editorials, ist für mich Indoktrination.

Bitte nehmen Sie es nicht persönlich! Ich hoffe inständig, dass Sie massenhaft solche Breitseiten bekommen, damit jedenfalls unser MINT-Bereich seine Reputation behält. Schließlich haben viele Menschen diesen Bereich auch deshalb gewählt, weil sie keine Lebenszeit in Theorien investieren wollen, deren Absurdität man in überschaubarer Zeit und abschließend nachweisen kann.

Nix für ungut, danke für Ihr Verständnis, und technisch weiter so!

Wolf Ludwig

Wichtigere Dinge

Wir sollten uns, statt unsere Sprache zu verhunzen, lieber um wichtigere Dinge kümmern, nein, nicht um den überbewerteten Klimawandel, sondern um die Zerstörung unserer Umwelt, wie der des Regenwaldes, wie das Insektensterben wegen der ganzen Pestizide, wie unsere schreckliche Massentierhaltung. Und weil wir schon bei der Sklaverei sind, etwas gegen die moderne Sklaverei durch Scheinselbstständige, rechtlose Angestellte bei Sub-Sub-Unternehmern tun. Siehe die aktuellen Skandale bei Tönnies, auf dem Bau oder bei Zustelldiensten. Hier wird es keinem besser gehen, nur weil es statt Master oder Slave sonstwie heißt. Das Problem liegt nicht bei den Bezeichnungen, sondern in den Köpfen der Menschen.

Rainer Müller-Knoche

Technisch korrekt

Real erlebe ich gerade eine lautstark geforderte Kulturrevolution – Bildersturm, Denkmal-

zerstörung und Geschichtsverfälschung jeder Art inbegriffen und auch von manchen Medien gerne polarisierend thematisiert. Die „Sprachrevolution“ von Gender-Sprech bis BLM ist da nur ein Teil dieser Entwicklung, aber eben ein wichtiger Teil, weil er Standards in der Gesellschaft verändert.

Im rein technischen Zusammenhang, den die darin lautesten Zeitgenossen vielleicht nicht gänzlich überblicken, ist z.B. die in Ihrem Editorial genannte Master-Slave-Beziehung zweier Bauteile oder Geräte aber eine exakte Beschreibung der Abhängigkeiten beider voneinander. Controller und Peripherie bedeutet noch lange nicht, dass ein peripheres Bauteil oder Gerät eine direkte und strikte Abhängigkeit vom Controller hat – Peripherie ist durchaus eigenständig und hat damit gerade kein explizites Master-Slave-Verhältnis zum Controller. Die deutsche Sprache ist sehr beschreibend und würde hier eine ihrer größten Stärken – die Beschreibungspräzision und -eindeutigkeit verwaschen. Ein „Slave“ (Sklave) ist ein in allen Daseinsbereichen vom „Master“ („Herren/Beherrscher“) Abhängiger.

Das MENSCHEN unter schlimmsten Bedingungen zu Sklaven gemacht wurden ist unbestreitbar unmenschlich – vor Hunderten von Jahren genauso wie heute. Kein Mensch sollte jemals in einem solchen Verhältnis existieren müssen! Der Slave eines JK-FlipFlops steht aber genau in diesem Verhältnis zu seinem Master – er ist von dessen Zustand elementar abhängig und kann auch durch Umbenennung nicht aus diesem strikten Verhältnis entfliehen. Hat er damit nicht sogar das Recht, als entrechteter Sklave der Digitalwelt wahrgenommen zu werden? Wir können ihn aber nicht befreien! Die Befreiung aller JK-FlipFlop-Slaves würde mit einem Totalausfall aller Digitaltechnik inklusive aller liebgewonnenen Smartphones einhergehen – auch die der lauten Verfechter der eingangs genannten Revolution.

J. Däbritz

Kontakt zur Redaktion

Leserbriefe bitte an:

mail@make-magazin.de

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

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

www.make-magazin/forum

 www.facebook.com/MakeMagazinDE

 www.twitter.com/MakeMagazinDE

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

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

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

Korrekturen

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

Abschaffung der Sprache

Danke für den in mehrerer Hinsicht erhellenden Beitrag. Bisher hatte ich solche Begriffe als technisches Fachwort gedankenlos verwendet. Nun plagen mich Reue und arge Zweifel, ob überhaupt passender Ersatz möglich ist. SDI und SDO bezeichnen nicht die

technisch notwendige Rangfolge der beteiligten Komponenten. Auch Begriffe wie „Mutter“ verwirren mit „Schraube“ als Gegenstück; wohl ob der Problematik von „Vaterdreher“, vulgo „Vaterzieher“. Die für sensible, auch fachfremde Menschen einzig befriedigende Methode ist die Abschaffung der Sprache. Vorher aber möchte ich noch ein technisches Gerät oder Prinzip finden, dem die Bezeichnung „linksgrünversiffter Gutmensch“ zuzuordnen wäre, um es für alle Zukunft aus meinen Entwicklungen zu verbannen – auch nachdem es seinen Namen verloren hat.

Peter Loster

Humorverbot

Unsere Moralisten haben neue Methoden entwickelt, die keiner Beweise bedürfen und auch die Unschuldsumutung ist ihnen fremd. In Coronazeiten mussten sie neue Tätigkeitsfelder entdecken und jetzt sind halt die Techniker dran. Sklave, welches Unwort, die Verwendung des Ausdrucks gehört gestrichen und verboten. Überhaupt sind diese Techniker so überkreativ, dass ein Kontakt männlich oder weiblich sein kann, ist doch eine Diskriminierung der Frau. Da gibt es auf den Mainboards so lange IDE-Steckerleisten, die einst der Kabarettist Willi Astor eine Frau zur Beschreibung ihres PC sagen ließ: „mein Steckplatz ist noch frei“. Soll das auch verboten werden, wie jede Art von Humor?

Wehrt euch. Wehret nicht den Anfängen, sondern das Treiben muss ein Ende haben. Keine Umbennungen von MOSI und MISO. Auch der Franzose bleibt ein Franzose!

Norbert Strauss

Danke!

Was für ein Editorial! Da fühlt man sich doch gleich besser. Die Werkätigen werden endlich aufgeklärt, ebenso wie die Nörds, dass das mit dem Slave und Master KEINE Verherrlichung oder Verharmlosung der Sklaverei darstellen soll und dass man ihnen zukünftig ihre schmutzigen Gedanken nicht einfach mehr durchgehen lassen wird. So fühlen wir uns alle reiner und irgendwie großeistiger! Wahnsinn, bald ist der Faschismus besiegt. Immer bereit!

Flo Brockmann

Erst Cutter, dann Arcade

Bartop Arcade mit Raspberry Pi, Make 4/20, S. 18

Die Bartop-Emulatorstation finde ich klasse. Ich habe mir aus einigen Pappersperrholzplatten vor Jahren schon mal eine gebaut, allerdings eine Nummer kleiner und eher „very quick & very dirty“ ①. Spaß hat das gemacht, aber die „Guerilla-Lackierung“ und das gefrickelte Erscheinungsbild gefiel mir nie. Da ich damals noch kein gescheites Holzbearbeitungswerkzeug hatte, brauchte ich eine Möglichkeit, Pappersperrholz sauber zuzuschneiden. Dafür hab ich dann einen Lasercutter selbst entworfen und gebaut ② – nach Eurer Vorstellung eines ähnlichen Geräts von Eleks-laser dachte ich: „Das bekomme ich selbst hin“ (youtu.be/PzsGBYa63Hk). Mittlerweile ist auch eine Absaugung und ein Schutzgehäuse dran. Damit konnte ich dann das Holz für meinen neuen Arcad-o-mat zuschneiden (③, thingiverse.com/thing:3287810).

Danke für die vielen Inspirationen und die guten Artikel im Heft. Auch wenn man nicht alles nachbauen kann, hat doch so gut wie jeder Artikel das Potenzial, eine Idee keimen zu lassen, auch wenn das vorgestellte Projekt selbst nicht in Frage kommen würde.

Marco Ding

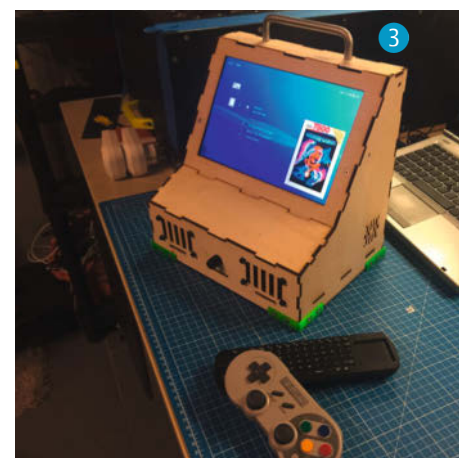
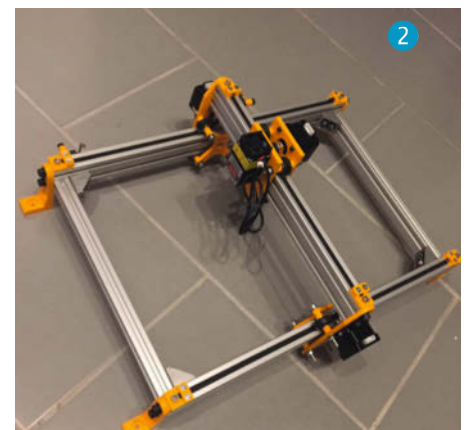
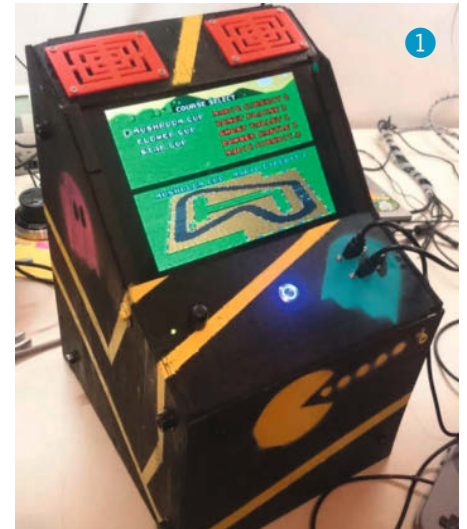
Lieblingswerkzeug vermisst

Lieblingswerkzeug, zuletzt in Make 6/19, S. 121

Wir sind seit langer Zeit begeisterte Leser des Make Magazins und basteln gemeinsam viel. Leider gab es schon lange kein Lieblingswerkzeug und auch keine Seiten mit Tipps&Tricks mehr im Heft. Diese Seiten fanden wir immer klasse, da waren viele tolle Anregungen und hilfreiche Tipps für uns dabei. Warum gibt es das nicht mehr? Wir wünschen uns diese Seiten wieder!

Andrea, Christine, Daniel, Florian, Kerstin, Mario, Markus, Rebecca, Roland, Thomas

Der Input für die Lieblingswerkzeuge wie für die Tipps&Tricks stammte überwiegend aus unserer Leserschaft – und leider erreichten uns da im letzten Jahr immer weniger Zuschriften. Am Ende stammte das meiste in beiden Rubriken aus



der Feder unseres rührigen Autorenduos Köhnlein/Gaus (vielen Dank übrigens!). Aber wir bringen natürlich ähnliche Inhalte weiter, so etwa in diesem Heft auf Seite 130.

Wassersensor-Widerspruch

Pflanzengieß für die Ferien, Make 2/20, S. 34, Ikea-Gewächshaus mit Arduino, Make 2/20, S. 40 und Community-Projekt: Blumengießanlage mit Multi-Function-Shield, Make 2/20, S. 90

In der Make-Ausgabe 2/10 werden drei Smart-Home-Projekte für Zimmerpflanzen vorgestellt. Für den Pflanzengieß werden kapazitive Bodenfeuchtigkeitssensoren empfohlen, die ohne direkten Wasserkontakt weniger von Zersetzungsreaktionen betroffen sind und daher Langlebigkeit bieten (S. 35, letzter Absatz). Gleich im nächsten Artikel über das autonome Gewächshaus wird der SMT50 beworben und mit einem „Chinasensor“ verglichen. Letzterer entspricht in Aussehen und Beschriftung dem vorher per Link gezeigten Bodenfeuchtigkeitssensor, welcher nach einem halben Jahr bereits starke Abnutzungserscheinungen zeigt (S. 43, Abb. 2).

Meines Verständnisses nach handelt es sich um das gleiche Produkt mit der gleichen Anwendungsweise (in die Erde des Blumentopfs stecken). Warum wird er einmal als langlebig und ein anderes Mal als kaputt nach 6 Monaten beschrieben? Erst benutzt und dann davon abgeraten? Ich hoffe, ihr könnt mich in meiner Verwirrung aufklären.

Lena Höding

Es stimmt, für den Selbstbau-Bewässerung haben wir die beim Ikea-Gewächshaus bemängelten Chinasensoren verwendet. Der Grund dafür war, dass der Selbstbau-Bewässerung nicht zur dauerhaften Bewässerung gedacht ist, sondern nur zum Überbrücken der Ferienzeit, also einiger weniger Wochen. Einfache Widerstandssensoren halten nach eigener Erfahrung noch nicht einmal diese Zeit durch und kamen daher nicht in Frage. Die kapazitiven „Chinasensoren“ erreichen insbesondere zusammen mit dem Tipp, die Sensoren mit Klarlack zu schützen, durchaus einige Monate Lebenszeit. Genug für die Urlaubsbewässerung und preiswert genug, um eine Selbstbau-Lösung für 30 Euro zu ermöglichen.

Das Gewächshaus mit Arduino hingegen ist als Dauer-Lösung im Ganzjahres-Betrieb gedacht. Daher müssen hier zuverlässige Sensoren mit sehr langer Lebensdauer verwendet werden. Die haben dann aber auch ihren Preis mit etwa 70 Euro/Stück.

Vier statt drei

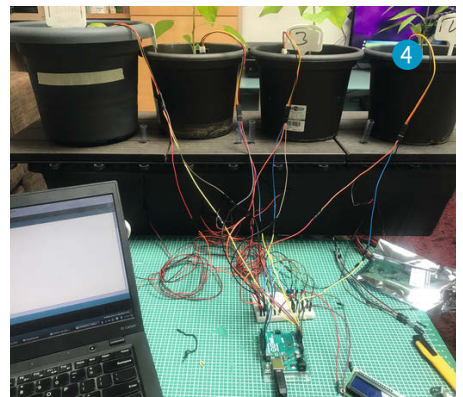
Ich habe mich ein wenig mit dem Gießautomat auseinandergesetzt und eine leicht abgewandelte Lösung überlegt. Ich habe einen

normalen Blumenkasten genommen und drei Kunststoff-Terrassenfliesen aus dem Baumarkt (4 Euro pro Stück) geholt. Ich habe unter den Fliesen links und rechts eine Reihe Standfüße mit dem Dremel weggesägt und sie passen sehr gut drauf. Somit muss ich auch keine Angst haben, dass der Blumenkasten auseinandergedrückt wird. Drumherum wird noch mit Silikonabgrenzung gebastelt und die Löcher für die Töpfe gesägt 4.

Das Programm habe ich für vier Töpfe angepasst. Zusätzlich habe ich noch ein LCD Display mit I²C Controller drangepackt und lasse mir dort die Wasserwerte ausgeben. Damit kann ich in den nächsten Tagen einfach beobachten, wie viel Wasser die Pflanzen verbrauchen. Dann muss ich nicht immer mein Laptop anschließen.

Dirk Gödde

Die Idee mit den Terrassenfliesen ist sehr gut. Wichtig ist vor allem, immer einen Rand zum Schutz gegen Überlaufen zu verwenden. Ihre Lösung mit Silikon ist da sicher gut geeignet. Da



die Pumpen nie gleichzeitig laufen, spielt der zusätzliche Stromverbrauch bei Erhöhung der Pumpenzahl keine Rolle. Im Grunde bestimmt die Anzahl der Ausgangs-Pins des Arduino die maximale Pumpenzahl. Allerdings würde so ein Blumenkasten doch recht unhandlich. Wir denken, maximal sechs Pumpen sind sinnvoll, um den Bewässerung auch einmal transportieren zu können.

—pek

Stephan Noller @holadiho · 4. Aug.

Für ein Technik-Magazin ist die @MakeMagazinDE immer wieder eine angenehme Überraschung.

2

2

8

Antworten

Hilmar Bunjes @hbunjes · 4. Aug.

Antwort an @holadiho und @MakeMagazinDE

Auch das Posting bzw die Mail von Linus Torvald fand ich sehr angenehm zu dem Thema incl der möglichen Alternativen.

Benedikt Koehler @furukama · 5. Aug.

Antwort an @holadiho und @MakeMagazinDE

Sehr schön!

Während uns per Mail zum Editorial vor allem kritische bis aufgebrauchte Stimmen erreichten, gab es auf Twitter auch Lob.



UNSER SORTIMENT VON TECHNIKERN FÜR TECHNIKER

The best part of your project: www.reichelt.de/sortiment

Nur das Beste für Sie – von über 900 Markenherstellern.

Unsere Produktmanager sind seit vielen Jahren bei reichelt tätig und kennen die Anforderungen unserer Kunden. Sie stellen ein breites Spektrum an Qualitätsprodukten zusammen, optimal auf den Bedarf in Forschung & Entwicklung, Instandhaltung, IT-Infrastruktur und Kleinserienproduktion sowie auf Maker zugeschnitten.

Das reichelt Raspberry Pi 4 B 8 GB All-In-Bundle

Das Raspberry Pi 4 Bundle beinhaltet alles Nötige um sofort starten zu können. Setzen Sie damit Projekte wie HD-Mediaplayer, Internetradio, Webcam-Server, Hausautomatisierung um oder nutzen Sie es einfach zum allgemeinen Surfen im Internet!

- mit Raspberry Pi 4 B mit 8 GB Arbeitsspeicher
- hochwertiges und trendiges Gehäuse
- leistungsstarkes USB-C-Netzteil
- 16 GB microSD-Karte (Class 10) mit vorinstalliertem NOOBS
- 4-teiliger Raspberry Pi Kühltasch, microHDMI-auf-HDMI-Kabel, 1 m



RaspberryPi

Bestell-Nr.: RPI 4B 8GB ALLIN

106,74

Raspberry Pi HQ-Kamera, 12,3 Megapixel

Hohe Empfindlichkeit, auch bei schlechten Sichtverhältnissen

- kompatibel mit allen Raspberry Pi Modellen
- Sensor: Sony IMX477R, Diagonale: 7,9 mm
- integrierter IR-Sperrfilter
- für Objektive mit C-/CS-Fassungen
- USB Type-C (5V/3A)



Bestell-Nr.:

RASP CAM HQ 53,13

Kameraobjektive für Raspberry Pi

Diese Objektive sind für die Original Raspberry Pi HQ-Kamera konzipiert.

- Teleobjektiv: 16 mm
- Weitwinkelobjektiv: 6 mm



Bestell-Nr.:

RPIZ CAM 16MM TO 52,15

RPIZ CAM 6MM WW 25,05

Objektiv

Tele

Weitwinkel



- Top Preis-Leistungs-Verhältnis
- über 110.000 ausgesuchte Produkte

- Zuverlässige Lieferung – aus Deutschland in alle Welt.

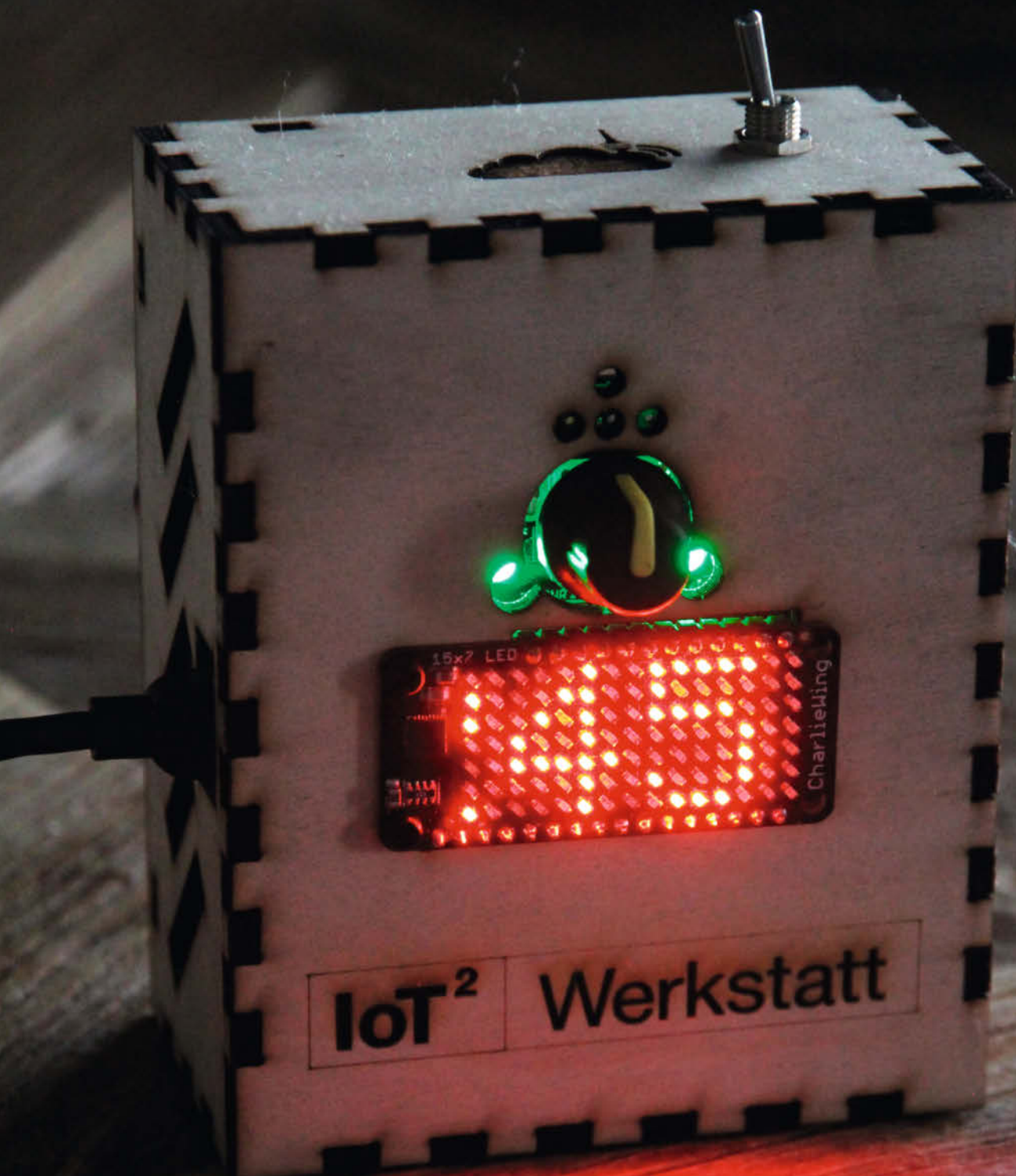
Bestellservice: +49 (0)4422 955-333

www.reichelt.de

reichelt
elektronik – The best part of your project

Es gelten die gesetzlichen Widerrufsregelungen. Alle angegebenen Preise in € inkl. gesetzlicher MwSt. zzgl. Versandkosten für den gesamten Warenkorb. Es gelten ausschließlich unsere AGB (unter www.reichelt.de/agb, im Katalog oder auf Anforderung). Abbildungen ähnlich. Druckfehler, Irrtümer und Preisänderungen vorbehalten. reichelt elektronik GmbH & Co. KG, Elektronikring 1, 26452 Sande, Tel.: +49 (0)4422 955-333

TAGESPREISE! Preisstand: 7. 9. 2020



Der CO₂-Warner für die Schule

Lüften, lüften, lüften – um in Coronazeiten das Infektionsrisiko in kleinen Räumen zu begrenzen, muss regelmäßig die Luft ausgetauscht werden. Wann es Zeit ist, die Fenster aufzureißen, kann eine selbstgebaute CO₂-Ampel anzeigen.

von Guido Burger, Richard Fix und Klaus-Uwe Gollmer

Kaum ein Thema bewegt unsere Gesellschaft im Moment mehr als die COVID-19-Pandemie, besonders der Start des Regelbetriebs an Schulen und Hochschulen ist eine Herausforderung. Hier gilt es, die Ausbreitung von SARS-CoV-2, wie das Virus genannt wird, durch organisatorische Maßnahmen zu verhindern. Inzwischen ist klar, dass die Viren vor allem über Tröpfchen übertragen werden, die beim Atmen, Sprechen und Husten entstehen. Große Tropfen fallen schnell zu Boden, sodass Abstand halten hilft. Mittlere Tropfen werden durch das Gewebe von Masken an der Ausbreitung gehindert. Aber was ist mit den feinsten Tröpfchen?

Diese Aerosole sind unsichtbar, schweben lange Zeit in der Luft und werden durch Verdunstung immer kleiner, bis nur noch die festen Bestandteile übrig bleiben. In den Tröpfchen vorhandene Viren können so bis zu Stunden infektiös bleiben. Kritisch ist dies vor allem in kleinen Innenräumen mit vielen Personen. Hier hilft nur moderne Klimatechnik oder lüften, lüften, lüften. Aber auch Lüften muss gelernt sein, denn zu viel Lüftung sorgt im Winter für unnötigen Wärmeverlust und eine schlechte Klimabilanz. Darüber hinaus möchte niemand unnötig frieren.

Mitte August empfahl das Umweltbundesamt daher, in jeder Unterrichtspause intensiv zu lüften. Außerdem, so das Amt, könnten „CO₂-Ampeln als Anhaltspunkt für gute oder schlechte Lüftung dienen“. Fertige Ampeln gibt es im Preisrahmen von 80 bis 170 Euro. Wir setzen aber auf DIY-Lösungen, an deren Bau und Programmierung gleich in mehreren Schulfächern angeschlossen werden kann – von der Informatik bis zur Chemie. Unser Modell arbeitet mit dem IoT Octopus als Mikrocontroller, der vielfältige Auswertungen unterstützt, von einer bunten Neopixel-LED-Ampel über Trendanzeigen auf einer LED-Matrix bis hin zu IoT-Plattformen wie Thingspeak und Grafana.

CO₂ und VOC

Um herauszufinden, wann wir lüften müssen, nutzen wir einen biologischen Zusammenhang: Ausatemluft enthält mehr CO₂ und volatile organic compounds (VOC, auf deutsch flüchtige organische Verbindungen) als die eingeatmete Luft. Deren Konzentrationen werden üblicherweise in ppm (parts per million) gemessen. Mit ca. 400 ppm CO₂ (0,04 vol%) und in der Regel weniger als 0,5 ppm VOCs enthält die normale Frischluft nur sehr wenig dieser unsichtbaren Gase. Die Konzentration in der Ausatemluft beträgt dagegen 40.000 ppm oder 4 vol% CO₂ und damit deutlich mehr. Da sich auch die potentiell gefährlichen Aerosole in der Ausatemluft befinden, können wir deren Ausstoß nun abschätzen, indem wir die anderen Gase messtechnisch erfassen.

Kurzinfo

- » Luftqualität verstehen
- » Richtig lüften
- » CO₂-Ampel mit IoT Octopus bauen

Checkliste



Zeitaufwand:

2 Stunden



Kosten:

ab 40 Euro



Holzbearbeitung:

Schneiden mit dem Lasercutter

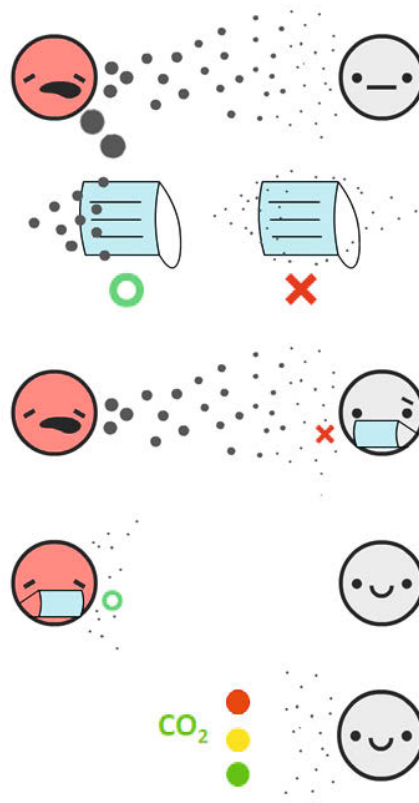
Material

- » IoT Octopus mit BME680 (oder anderes ESP8266-basiertes Board)
- » Grove CO₂-Sensor (SCD-30) mit Kabel
- » Adafruit Charlie Wing
- » Micro-USB-Kabel (5 Volt)
- » 3mm Birken-Sperrholz
- » 9x Nylon-Schrauben und Muttern (2,5mm)
- » 4x Schrauben

Mehr zum Thema

- » Guido Burger, IoT.Octopus, Make-Sonderheft IoT, S. 16
- » Florian Schäffer, Umweltsensoren für Mikrocontroller, Make 1/19, S. 8
- » Helga Hansen, Umweltsensoren einfach eingesetzt, Make 1/20, S. 16

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



Große Tropfen fallen zu Boden (Abstand), mittlere Tröpfchen werden beim Wegfliegen durch Verdunstung kleiner (Aerosole).

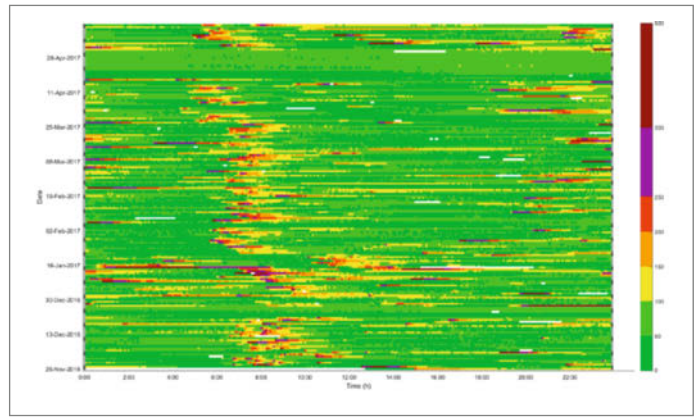
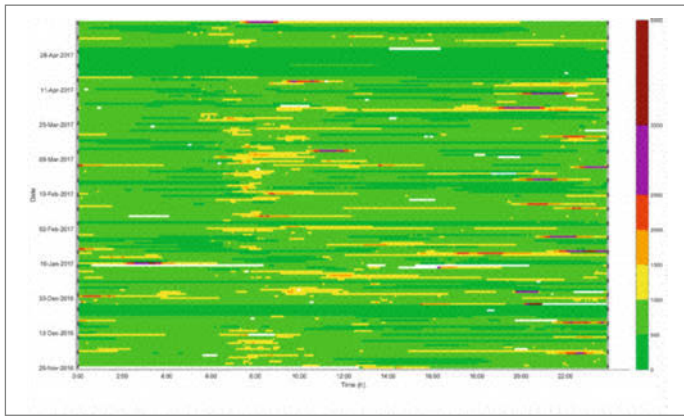
Masken halten größere Tröpfchen besser auf, kleinere schlechter.

Deshalb sind Masken weniger gut dazu geeignet, sich vor einer Erkrankung zu schützen.

Aber effektiv dabei, dir zu helfen, andere vor einer Ansteckung zu schützen.

Eine CO₂-Ampel dagegen kann dich vor möglichen Aerosolen warnen.

Adaptiert von tquev.github.io/covid-19, einer spielbaren COVID19-Simulation



Vergleich von Sensordaten in einem Badezimmer (links CO₂, rechts VOC)

Beim Bau einer Ampel, die zum Lüften auffordert, sind prinzipiell sowohl Messungen von CO₂ als auch von VOCs nutzbar. Der grundlegende Unterschied ist, dass CO₂-Messungen nur ein Gas in der menschlichen Ausatemluft ermitteln, während VOC-Sensoren fast alle Verunreinigungen der Raumluft durch chemische Stoffe erfassen. Sind allein Menschen der Grund für schlechte Luft, führt dies zu äußerst

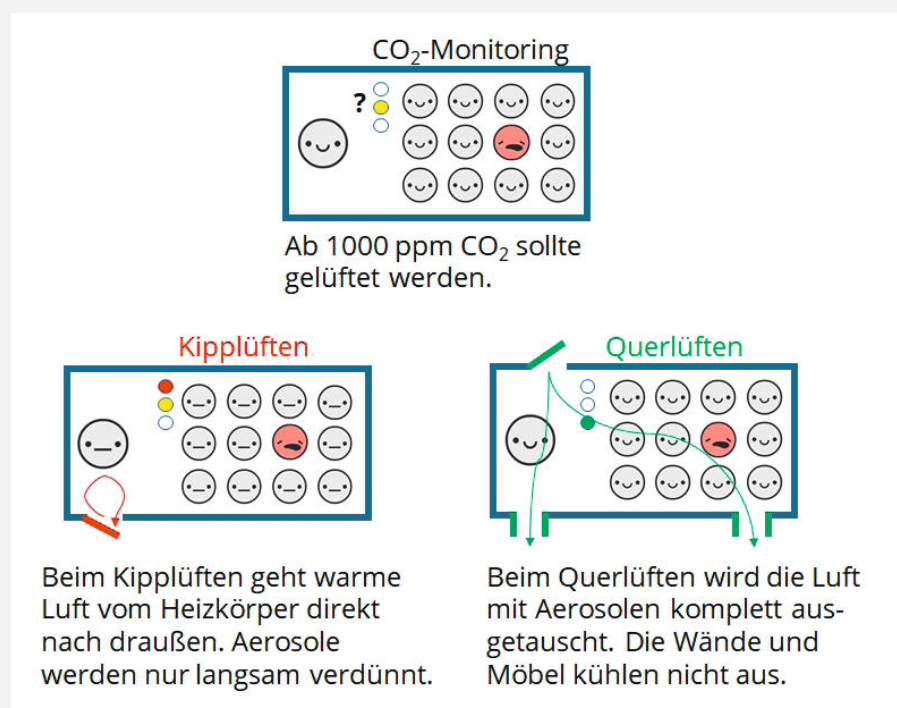
ähnlichen Ergebnissen, etwa im Schlafzimmer. Einen deutlichen Unterschied gibt es dagegen in Badezimmern, Küchen und Esszimmern oder wenn schlechte Luft von außen kommt, beispielsweise Verbrennungsgase. Auch Ausdünstungen aus Teppichen und Gegenständen oder der Einsatz von Putzmitteln wird von VOC-Sensoren erfasst. Für eine automatische Lüftungsregelung in Häusern sind sie deshalb

zu bevorzugen, weil sie die allgemeine Luftqualität erkennen. Außerdem sind sie deutlich günstiger, kleiner und bringen oft eine Temperatur-, Feuchte- und Luftdruckmessung mit, was ebenfalls der Einschätzung der Luftqualität dient. Da die meisten Schulen keine Lüftungsanlagen haben, ist hier eine Anzeige nötig, die ein Zeichen liefert, wenn das Lüften notwendig ist bzw. beendet werden kann.

Richtig lüften

Kippen oder ganz öffnen, ein oder mehr Fenster – wie lüftet man nun richtig? Hier gibt es viele Möglichkeiten und viele Fehlerquellen. Das Ziel ist der möglichst komplette Austausch der belasteten Luft durch frische Luft von draußen. Am sinnvollsten ist eine kurze Querlüftung, bei der zwei gegenüberliegende Fenster weit geöffnet werden und kurz Durchzug erlauben. Dadurch wird eine hohe Wechselrate erzielt und wir haben schnell wieder frische Luft. Ist Querlüftung nicht möglich, sollten so viele Fenster wie möglich weit geöffnet werden. Außerdem empfiehlt das Umweltbundesamt, in Räumen, in denen Sport getrieben wird, fünfmal pro Stunde auf diese Weise zu lüften und nach Auftreten von Krankheitssymptomen wie wiederholtem Husten oder Niesen.

Gerade im Winter werden die Fenster oft etwas länger auf Kipp gestellt, damit niemand frieren muss. Dabei hat Luft eine vergleichsweise geringe spezifische Wärmekapazität und speichert nur wenig Wärme. Kalte Luft wird im Raum nach kurzer Querlüftung daher sehr schnell erwärmt. Das warme Mauerwerk und die Möbel dienen quasi als große Flächenheizung. Bei Kipplüftung ist der Luftaustausch



dagegen sehr schlecht. Im schlimmsten Fall tauschen wir nur die warme Luft über der Heizung, während die aerosolhaltige Luft im Raum stehen bleibt. Da die Warm-

luft vom Heizkörper nach draußen strömt, wird es im Raum kälter und wir denken, wir hätten genug gelüftet – in Corona-Zeiten ein gefährlicher Trugschluss.

Dafür gilt die CO_2 -Konzentration bereits seit langem als guter Indikator mit klaren Grenzwerten. CO_2 -Werte sind außerdem einfach zu verstehen und durch das lineare Messprinzip der Sensoren kann man auf die Anzahl der Personen im Raum schließen, was in Corona-Zeiten ebenfalls sinnvoll sein kann.

Messprinzipien

Moderne CO_2 -Sensoren messen die Lichtabsorption von CO_2 -Molekülen im Infrarotbereich, bei $4,3 \mu\text{m}$, und werden daher auch nichtdispersive Infrarotsensoren (NDIR) genannt. Da dies eine entsprechende optische Weglänge und Lichtleistung erfordert, sind selbst miniaturisierte CO_2 -Sensoren noch einige Kubikzentimeter groß und haben eine mittlere Stromaufnahme von 20 mA. Um verlässliche Messergebnisse zu liefern, müssen sie regelmäßig kalibriert werden. Dies ist an der Außenluft meist einfach möglich.

Seit 2017 gibt es miniaturisierte, digitale VOC-Sensoren, die flüchtige organische Verunreinigungen messen. Sie sind nur wenige Kubikmillimeter groß, günstig herzustellen und haben eine mittlere Stromaufnahme von 0,09 bis 0,9 mA, je nach Betriebsmodus. Das zugrundeliegende Messprinzip ist das der halbleitenden Metalloxide, deren elektrische Leitfähigkeit sich beim Kontakt mit VOCs ändert. Die geringste Leitfähigkeit zeigt das Metalloxid in reiner Luft. Je mehr Verunreinigungen dazukommen, umso höher wird die Leitfähigkeit. Sie steigt aber nicht linear, sondern stark bei Veränderungen kleinster Gesamtkonzentrationen und langsamer bei entsprechend großen Konzentrationen. Die Sensoren linearisieren das Signal dann.

VOC-Sensoren messen außerdem kein einzelnes Gas, sondern nahezu alle reaktiven Verbindungen in der Luft in Summe. Aus dem Messwert kann somit nicht die Konzentration einzelner Moleküle abgeleitet werden. Man-

che VOC-Sensoren errechnen aus dem VOC-Signal noch CO_2 -Äquivalente ("e CO_2 "). Falls Ausatemluft die wesentliche Quelle ist, stimmen die Werte mit CO_2 -Messungen gut überein. Im Falle anderer Quellen ist das CO_2 -Äquivalent deutlich höher – gelüftet werden muss dann aber auch.

Sowohl die Messungen von CO_2 wie auch von VOCs sind von Einflüssen wie Luftfeuchte, Luftdruck und Temperatur abhängig. Einige Module bringen daher entsprechende Sensorelemente zur Kompensation gleich mit. In fertigen CO_2 -Ampeln ist übrigens nicht immer ersichtlich, welcher Sensor zum Einsatz kommt: Ist ein optischer NDIR-Sensor verbaut oder wird ein e CO_2 berechnet? Auch die Angabe, wie die Geräte zu kalibrieren sind, fehlt manchmal. In solchen Fällen ist es schwer einzuschätzen, wie zuverlässig die Ampeldata eigentlich sind.

Grenzwerte

Das Umweltbundesamt hat allgemeine Leitlinien zur "Gesundheitlichen Bewertung von Kohlendioxid in der Innenraumluft" verfasst, an denen wir uns orientieren, um die Schwellwerte für die Ampel festzulegen. Demnach ist eine CO_2 -Konzentration unter 1000 ppm hygienisch unbedenklich. Eine Konzentration zwischen 1000 und 2000 ppm stuft die Leitlinie als bedenklich und alles darüber als inakzeptabel ein. Ab 1000 ppm sollte danach mit dem Lüften begonnen werden und über 2000 ppm muss gelüftet werden. Führt dies zu Störungen des Unterrichts, sollten Maßnahmen wie eine Verringerung der Personenzahl im Raum oder der Einbau einer Lüftungsanlage überprüft werden.

Der CO_2 -Wert ist dabei nicht nur Indikator für die Verbreitung von Aerosolen, sondern auch für das allgemeine Wohlbefinden. Allerdings beeinflussen dies vor allem die chemisch reaktiven VOCs, welche mit dem menschlichen

SCD-30-Anschlüsse

SCD-30	Grove-Kabel
VIN	rot
GND	schwarz
SCL	gelb
RX/SDA	weiß

Organismus wechselwirken und somit beispielsweise zu Unwohlsein führen können. In einer weiteren Handreichung empfiehlt das Umweltbundesamt, die Summe der organischen Verunreinigungen immer unter 10 bis 25 mg/m^3 (entspricht ca. 5 bis 10 ppm, abhängig vom jeweiligen VOC) zu halten. Schließlich sollte die relative Luftfeuchtigkeit im Raum bei 40 bis 60 Prozent liegen, da Aerosole bei geringerer Luftfeuchte länger in der Luft bleiben. Diese Empfehlungen gelten aber vor allem für den Einsatz von Lüftungsanlagen.

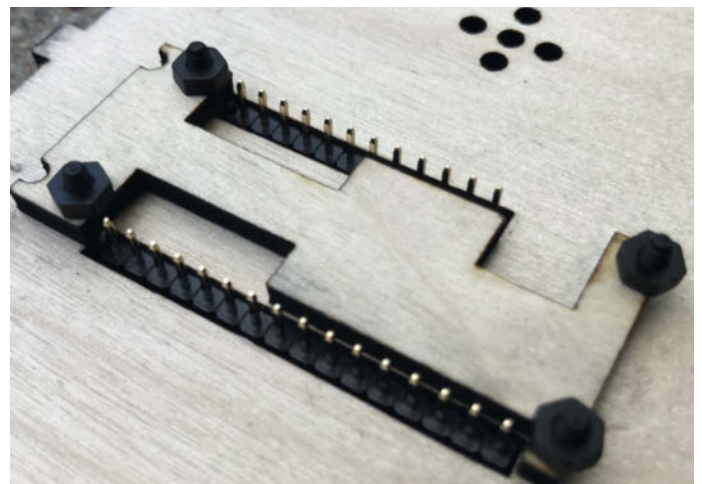
Elektronik

In unserer Ampel nutzen wir den IoT Octopus, den wir für den Einsatz im Schulunterricht konzipiert und bereits im Make IoT-Sonderheft 2016 vorgestellt haben. Die quelloffene Platine basiert auf dem WLAN-Mikrocontroller ESP8266, und verfügt neben dem VOC-Sensor BME680 und zwei Neopixel-RGB-LEDs über zwei Grove-Buchsen und einen Feather-Steckplatz, an denen weitere Hardware schnell angesteckt werden kann. Darüber hinaus haben wir den Bau mit alternativen, ESP8266-basierten Platinen, wie WeMOS D1, Adafruit Feather Huzzah oder NodeMCU getestet. Weitere Hinweise dazu finden Sie online (siehe Link in der Kurzinfo).

Mit dem integrierten VOC-Sensor BME680 lässt sich bereits näherungsweise ein CO_2 -Wert ermitteln und mit den LEDs in Ampelfarben ausgeben. Der BME misst außerdem Luftfeuchte, Temperatur und Luftdruck und kom-



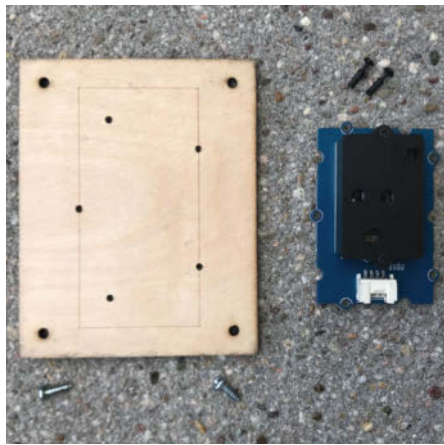
Der Sperrholzrahmen zusammengesteckt und das Display bereits montiert



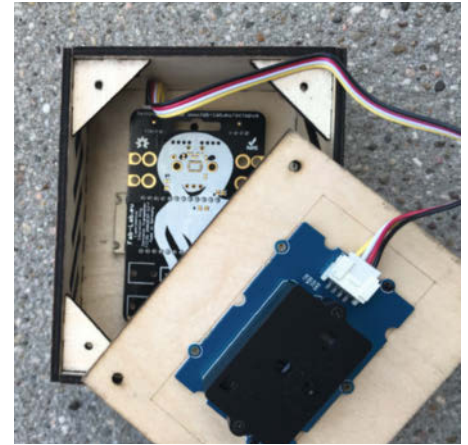
So ist das Holzstück hinter dem Display zu verschrauben.



Die Stromversorgung erfolgt über die seitliche USB-Steckbuchse.



Rückwand mit Bohrungen für die Aufnahme des CO₂-Sensors



Ein Grove-Kabel verbindet den CO₂-Sensor mit dem IoT Octopus.

pensiert so deren Einflüsse auf Messwerte. Zusätzlich verwenden wir einen NDIR-Sensor, den Sensirion SCD-30. Sein Messbereich liegt zwischen 400 und 10.000 ppm, mit einer Genauigkeit von ± 30 ppm plus 3% vom Messwert. Bei einer Messung pro zwei Sekunden benötigt er 19 mA Strom. Er besitzt eine automatische Temperaturkompensation über einen eingebauten Sensor. Die Version des chinesischen Hardwareherstellers Seeed Studio lässt sich am einfachsten in eine Ampel bauen, da diese Platine einen Grove-Anschluss mitbringt. Mit einem Grove-Kabel lässt sich der Sensor einfach mit dem Mikrocontroller verbinden. Wer eine günstigere Version kauft, muss das Kabel anlöten. Nach dem Abschneiden eines Steckers werden die vier Leitungen angeschlossen, wie in der Tabelle SCD-30-Anschlüsse angegeben.

Auf dem IoT Octopus kann optional ein Knopf als Drehregler und Button montiert werden, um den CO₂-Sensor später auf Knopfdruck zu kalibrieren. Schließlich brauchen wir noch eine Anzeige, auf der wir unsere Messwerte ausgeben. Ohne Extraaufwand geht dies mit den Neopixel-LEDs, die auf dem Board bereits verbaut sind. Sie können grün, gelb oder rot aufleuchten. Für eine detaillierte Anzeige des Werteverlaufs setzen wir eine 15x7 LED-Matrix von Adafruit ein, die auf dem IoT Octopus einfach eingesteckt werden kann. Weitere mögliche Anzeigen sind zum Beispiel

der Neopixel Featherwing von Adafruit, ein Neopixelring oder LC-Display, die alle von der verwendeten Programmierumgebung unterstützt werden.

Gehäuse

Für den dauerhaften Einsatz unserer CO₂-Ampel benötigen wir eine Verkleidung, welche die sensible Elektronik schützt und das Aufstellen erleichtert. Dazu verwenden wir mit einem Laser geschnittene 3mm-Sperrholzplatten. Die Vorlagen für verschiedene Ampeln gibt es kostenlos zum Download (siehe Kurmlink). Die Teile sind mit Verzahnungen gestaltet, so dass man sie einfach zusammenstecken kann. Dabei beginnen wir mit den Seitenteilen und den rechteckigen Halterungen der Rückseite. Das linke Seitenteil hat neben vier Lüftungsschlitzen noch eine Aussparung für das USB-Kabel, je nach Variante ist im oberen Seitenteil oder der Front ein Loch für den optionalen Schalter. Ein Tropfen Holzleim in den Ecken versteift die Konstruktion und sorgt so für bessere Haltbarkeit. Die Variante mit dem Adafruit Feather Huzzah passt übrigens genauso in dieses Gehäuse.

Das Display wird auf der Vorderseite mit einer kleinen Hilfskonstruktion verschraubt. Dazu sind die vier 2,5mm-Plastikschrauben und Muttern nötig. Die richtige Lage des Boards und des kleinen Holzstücks (siehe Bild) ist wichtig, damit der IoT Octopus gleich

darauf passt. Stehen die Schraubengewinde zu weit über, sollten sie gekürzt werden.

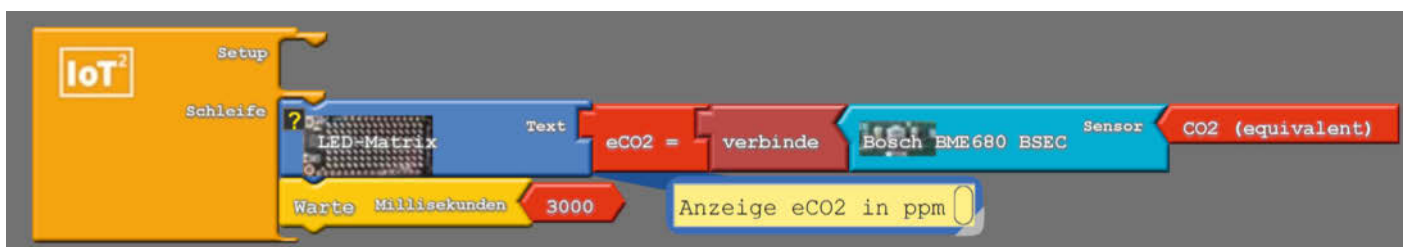
Auf die Steckerleisten kann nun der IoT Octopus aufgesteckt werden, wobei ebenfalls auf die Orientierung zu achten ist: Der BME680-Sensor gehört hinter die fünf Löcher der Frontseite. Nach dem Aufstecken der Front sollte der USB-Anschluss hinter der Aussparung auf der linken Seite zu sehen sein.

Im letzten Schritt montieren wir den Sensor. Hierzu sind in der Rückwand des Gehäuses Bohrungen vorgesehen, in denen der SCD-30 in der empfohlenen Grove-Version sehr leicht verankert werden kann. Hierzu gibt es kleine Plastikdübel von Grove. Alternativ kann man wieder 2,5mm-Schrauben nehmen, wofür die Bohrungen auf der Platine eventuell erweitert werden müssen. Ein Stück Draht oder doppel-seitiges Klebeband erfüllen die Aufgabe letztlich ebenso.

Das Grove-Kabel muss in die rechte Buchse des IoT Octopus gesteckt werden und verbindet so den CO₂-Sensor mit dem IoT Octopus. Hierbei wird der I²C-Datenbus verwendet. Mit vier Schrauben wird die Rückwand nun verschlossen.

Die Steuerung

Für die Programmierung nutzen wir IoT-Ardu-block, unsere Variante der bekannten Programmierumgebung für den Arduino. Sie



Unser erstes Programm

funktioniert mit Blöcken und umfasst spezielle Erweiterungen zum Thema Internet der Dinge (IoT). Die aktuelle Version für Windows und MacOS können Sie kostenlos auf unserer Seite herunterladen (siehe Kurzinfo). Dort finden Sie weiter eine Anleitung zur Inbetriebnahme und zur Installation des gegebenenfalls nötigen USB-Treibers, wobei es unter Windows schneller als unter MacOS geht. Gestartet wird zunächst die Datei *IoT-Werkstatt.bat*. Falls der Windows Defender dies verhindert, geht es mit Klick auf *Weitere Informationen* und *Trotzdem Ausführen* weiter. Außerdem muss der Zugriff auf „Java(TM) Platform SE binary“ zugelassen werden. Aus der Arduino-IDE heraus geht es über *Werkzeuge* schließlich zu *Ardublock*.

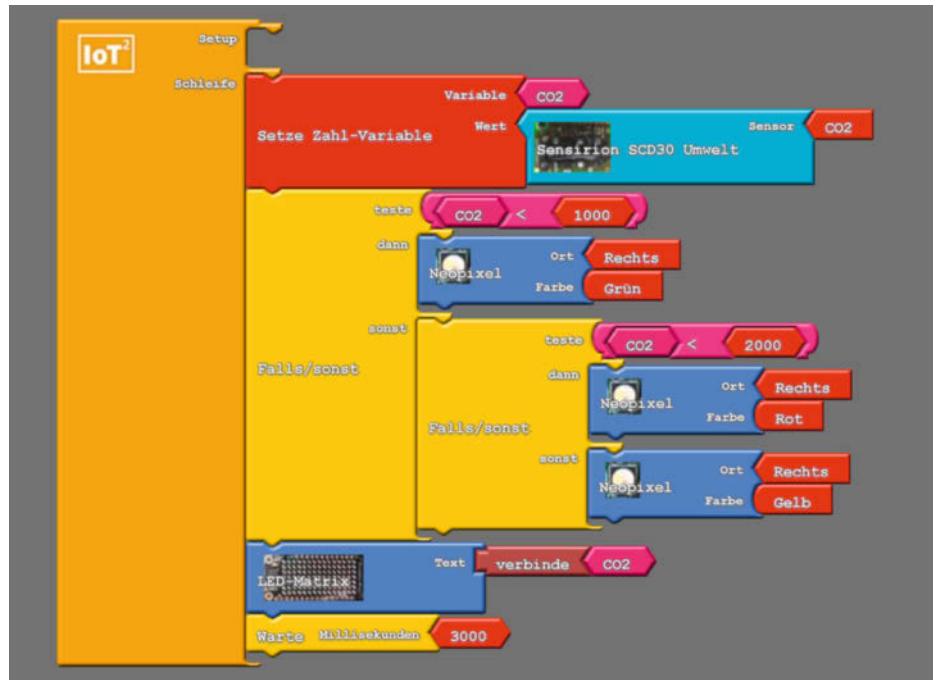
Die Ardublock-Programme bestehen, wie die übrigen Arduino-Programme, grundsätzlich aus zwei Teilen. Im *Setup* werden Anweisungen eingetragen, die der IoT Octopus nur einmal, nach dem Start, zu erledigen hat. Alle Aufgaben in der *Schleife* laufen dagegen solange wieder und wieder ab, wie der IoT Octopus angeschaltet ist. Die möglichen Anweisungen finden wir als bunte Funktionsblöcke in der linken Seitenleiste der IoT-Werkstatt. Nach dem Schlüssel-Schloss-Prinzip sind die Übergänge zwischen den Blöcken in verschiedenen Formen gehalten, sodass nur die passenden Datentypen angeschlossen werden können.

In unserem ersten Programm testen wir die LED-Matrix, den Charlie-Wing, und geben den eCO_2 -Wert des eingebauten BME680-Sensors aus. Im Setup sind keine Angaben nötig. In der Schleife setzen wir zunächst den blauen Block für die LED-Matrix (aus *Anzeigen/Aktoren*) mit dem Text *eCO=*. Der Baustein *verbinde* (aus *Kommunikation (M2M)*) sorgt dafür, dass der Sensor-Messwert in den passenden Datentyp umgewandelt wird, weshalb man beim Auswählen auf die passenden Formen achten muss. Daran wird der Sensor (*BOSCH BME680 BSEC*) angeschlossen und in seinem roten Block als auszugebender Wert *CO2 (equivalent)* ausgewählt. Nach jeder Messung soll der Octopus eine Pause von 3000 Millisekunden machen, den passenden Baustein finden wir unter *Kontrollstrukturen*.

Mit dem Einsetzen der Bausteine für die LED-Matrix und den Sensor bindet die Programmierungsumgebung automatisch die Treiber und Bibliotheken ein, mit denen die Bauteile angesprochen werden. Jetzt müssen wir den IoT Octopus nur noch mit einem USB-Kabel an den Rechner anschließen und das Programm mit einem Klick auf *Hochladen auf den Octopus* auf den Mikrocontroller laden.

Ausgabe als Ampel

Jetzt wollen wir Ampelfarben sehen. Dazu verwenden wir den externen CO_2 -Sensor SCD-30 und lassen die auf dem Octopus vorhandenen



Zwei geschachtelte Fallunterscheidungen – und fertig ist die Ampel

Neopixel in rot, gelb oder grün aufleuchten. Der Messwert wird wieder auf der Matrix angezeigt. Auch dieses Programm hat kein Setup. Wir beginnen mit *Setze Zahl-Variable* und nennen die Variable *CO2*. Um den vorgegebenen Wert 0 gegen den Block *Sensirion SCD30 Umwelt* (aus *Externe Interfaces*) austauschen zu können, muss man die 0 noch anklicken. Aus den *Kontrollstrukturen* holen wir dann eine *Falls/sonst-Unterscheidung* und setzen unter *teste* einen Vergleich mit *<* (aus *Log. Operatoren*). Unsere Variable *CO2* können wir mit einem Rechtsklick klonen und links im Vergleich einfügen. Rechts setzen wir die Zahl 1 (aus *Variablen/Konstanten*) und verändern ihren Wert auf 1000 – den Schwellenwert des Umweltbundesamtes. Ist dies der Fall, leuchtet über *dann* der rechte Neopixel (aus *Anzeigen/Aktoren*) grün auf. Unter *sonst* schachteln wir eine zweite *Falls/sonst-Unterscheidung* und vergleichen nun gegen die zweiten Schwellenwert von 2000 ppm. Mit der Klonen-Funktion ist dies schnell gebaut. Schließlich ergänzen wir den *LED-Matrix-Block* und lassen den Wert der Variable *CO2* über den passenden *verbinde* einlaufen. Den Abschluss macht eine Pause über 3000 Millisekunden.

Trend-Anzeige

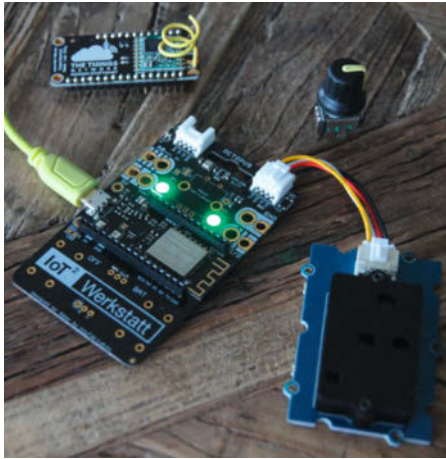
Auf der LED-Matrix können die Sensordaten über die Zeit sogar mit kleinen Balkendiagrammen dargestellt werden. So können wir einen einfachen Trend, zum Beispiel über die letzte Stunde, ablesen. Hierzu dient der Block *Schreibe Array* (aus *Variablen/Konstanten*). Das Array umfasst 15 Stellen, von 0 bis 14, der *index*, der jeweils mit einem *wert* gefüllt wird. Der

untere Block zeigt, dass diese Werte wieder vom CO_2 -Sensor kommen und befüllt den *index* mit 14. Damit die Daten mit der Zeit nach links über das Display wandern, beginnen wir das Programm insgesamt mit einer *Zählschleife* (aus *Kontrollstrukturen*). Unsere *Variable* nennen wir *i* und bei *Mal* setzen wir 14 ein. Hier schiebt *Schreibe Array* nun den Wert von Stelle 14 auf Stelle 13. Der Block *Lese Array* ist unter *Variablen/Konstanten* zu finden und auch die Klon-Funktion ist wieder hilfreich. Damit die Daten des Arrays auf der LED-Matrix erscheinen, fehlt nur noch *Anzeige Array* (aus *Anzeigen/Aktoren*) und eine Pause von fünf Minuten (300.000 Millisekunden). Mit den Angaben bei *min* und *max* begrenzen wir die angezeigten Konzentrationen auf Werte zwischen 380 und 1200 ppm, sodass jeder Pixel 117,14 ppm entspricht.

Nun ist unsere Ampel bestens gerüstet für den Einsatz im Klassenraum, bei dem sich die Neopixel und die Anzeigematrix quasi ständig im Blick des Publikums befinden.

Anbindung an Thingspeak

Wie sieht das Tagesprofil aus, welche maximalen Werte haben wir heute gesehen, wie dokumentieren wir den Lüftungserfolg? All diese Punkte lassen sich mittels IoT-Funktionalität quasi auf einmal klären. Dazu verbinden wir uns mit dem (hoffentlich) vorhandenen WLAN und speichern die Daten in der Cloud. Hier spielt unsere IoT-Werkstatt ihre größten Stärken aus. Ob MQTT, InfluxDB, IFTTT, oder Blynk – für alle großen Maker-Plattformen und -Dienste existieren grafische Blöckchen zur einfachen Integration.



Die einfachste Ampel: Platine mit angeschloss-
nem CO₂-Sensor



Die Trendanzeige in Aktion

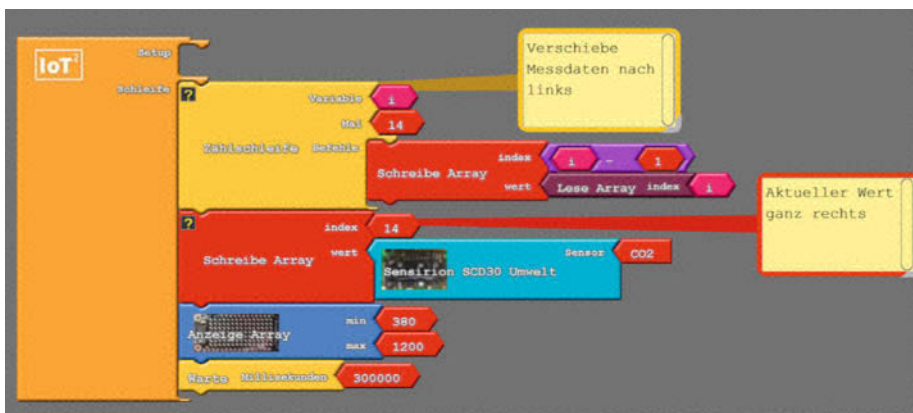
Das folgende Programm nutzt die IoT-Platt-
form Thingspeak. Die Zugangsdaten zum
WLAN tragen wir über den Block aus *IoT: Inter-
net-Zugang* ein. Da Thingspeak in der kosten-
losen Variante nur ein Update-Intervall von 15
Sekunden zulässt, aktualisieren wir mit *Wieder-
hole* erst vier Mal allein die Anzeige der
LED-Matrix, bevor wir die verschiedenen Mess-
werte mit *Send Thingspeak* in die Cloud schi-
cken. Der API-Key sorgt für die Verbindung
zum richtigen Thingspeak Channel. Sie be-
kommen ihn nach der Anmeldung auf der
Plattform. Im Make IoT-Sonderheft finden Sie
auf S. 77 eine Anleitung dazu. Die vielfältigen
Export- und Visualisierungsmöglichkeiten von
Thingspeak ermöglichen den schul- und auf
Wunsch weltweiten Zugriff auf Zeitreihen. Die
Thingspeak-Serversoftware gibt es auch als
Open-Source-Version zur Selbstinstallation
auf einem lokalen Raspberry Pi oder anderen
Rechner. Weitere IoT-Beispiele finden Sie on-
line (siehe Kurzinfo).

Kalibrierung

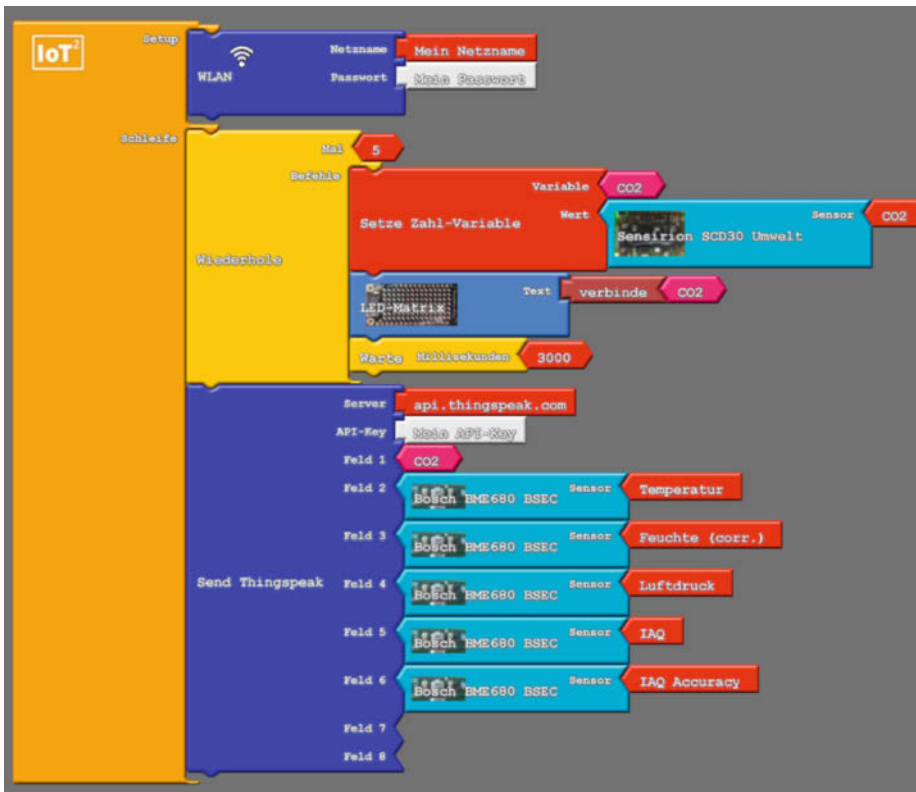
Die NDIR-Sensoren kommen mit einer werk-
seitigen Kalibrierung und können direkt ein-
gesetzt werden. Einmal im Jahr oder wenn die
Messergebnisse unpassend erscheinen, ist
eine Kalibrierung notwendig, um die vom
Hersteller angegebene Genauigkeit aufrecht
zu erhalten. Die typische Abweichung pro
Jahr liegt beim SCD30 bei 30 ppm und maxi-
mal 80 ppm. Um die Messergebnisse zu über-
prüfen, kann man den Sensor für einige Stun-
den die frische Luft messen lassen, zum
Beispiel an einem offenen Fenster. Hier sollte
ein Wert um 400 ppm (± 50 ppm) vorliegen,
dem normalen CO₂-Anteil. Die Außenluft dient
auch als Kalibriergas. Dabei gibt es zwei Mög-
lichkeiten:

1. Automatische Kalibrierung: In diesem
Modus muss der Sensor sieben Tage lau-
fen und in dieser Zeit jeweils für eine be-
stimmte Zeit an der "frischen" Luft sein.
Beim SCD-30 ist es täglich eine Stunde.
Der Sensor sucht dann selbständig seine
Basislinie und speichert diese auf dem
Sensor.
2. Erzwungene Kalibrierung: Hierbei wird
der Sensor einer definierten CO₂-Konzen-
tration (wie 400 ppm) ausgesetzt und der
Basiswert angepasst. Der SCD-30 muss
der Konzentration mindestens 2 Minuten
ausgesetzt werden, auf jeden Fall solange,
bis er einen stabilen Messwert anzeigt. In
der IoT Werkstatt haben wir dafür einen
Block vorbereitet: *Calibrate SCD30 to fresh
air* (zu finden unter *Externe Interfaces*).

Um den Sensor mit einem Knopfdruck zu
kalibrieren, muss nur eine *Falls*-Abfrage (aus
Kontrollstrukturen) in das eigene Programm
eingebaut werden. In unserer Komfort-Aus-
führung leuchtet nach dem Drücken des



Programm zum Erstellen der Trendanzeige



Zwei blaue IoT-Superblöcke und fertig ist der internetbasierte Datenlogger.

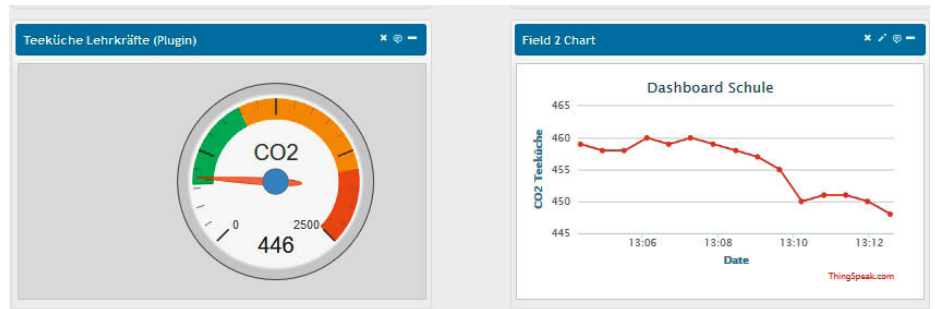
Buttons (aus Sensoren) erst die rechte Neopixel-LED blau auf und nach einer Pause von einer Sekunde beginnt die Kalibrierung.

Wer es ganz genau haben möchte, muss bei der Messung auch den Luftdruck berücksichtigen. Dafür kann man im Kalibrierungsblock die Höhe über Meereshöhe (*Altitude*) eintragen.

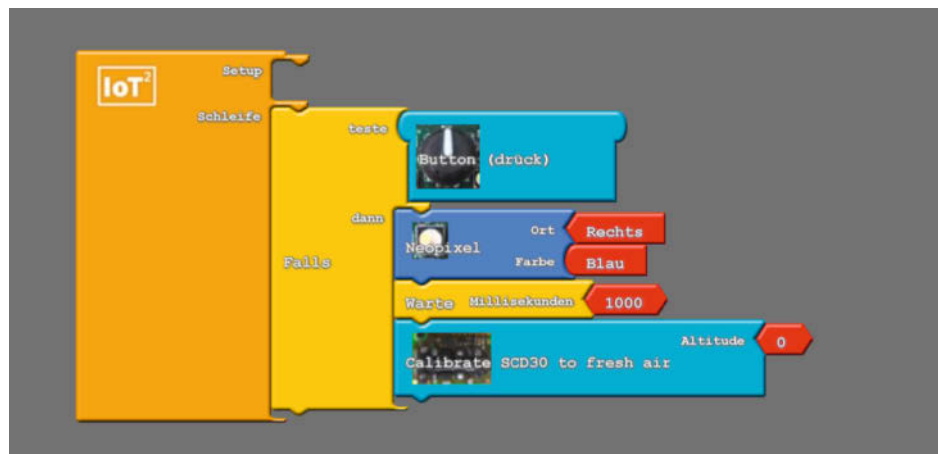
Einsatz (nicht nur) in der Schule

In Schulklassen bietet der Nachbau einer CO₂-Ampel Anknüpfungspunkte in vielen Fächern, über die Erläuterung der Programmierung im Informatik-Unterricht hinaus. So kann man im Physik-Unterricht klären, wie lange Tröpfchen im Raum schweben und warum die Luft beim Lüften auskühlen kann, die Wände dies aber nicht tun sollten. Die Zeitreihendarstellung auf Cloud-Plattformen macht die Auswertung und den Vergleich verschiedener Sensoren beziehungsweise Räume zum Kinderspiel und lädt zum Experimentieren ein. Wie unterscheidet sich Kipplüften vom Querlüften? Verändert die Anwendung eines Luftreinigers die Messwerte? Selbst die Kunst kann man mit ins Boot holen und kreative Alternativen für die Darstellung der Messergebnisse entwerfen – etwa auf einem Neopixelring anstelle der LED-Matrix. Weitere Ideen für den Unterricht sammeln wir auf unserer Projekt-Webseite, zu erreichen über den Link in der Kurzinfo. Schließlich sind Gemeinschaftsprojekte mit engagierten Eltern, Sponsoren aus der lokalen Wirtschaft und dem Schulträger denkbar.

Für alle, die in anderen Innenräumen arbeiten, bietet sich die CO₂-Ampel natürlich auch an. Gerade im Home Office vergisst man gerne das Lüften, dabei ist es nach langen Telefonaten dringend nötig. Und selbst Funktionsräume wie Labore, Teeküchen und Toiletten, die von wechselnden Nutzern frequentiert wer-



Die Thingspeak-Plattform visualisiert Zeitreihen – ein perfektes Tool für das Schul-Dashboard



Mit diesem Programm wird die Ampel auf Knopfdruck kalibriert.

den, können mit einer Ampel überwacht werden, um das Infektionsrisiko zu senken und trotzdem die Klimabilanz zu schonen.

Dabei kann die Ampel beim Nachbau individuell angepasst werden. Sinnvoll wäre zum Beispiel die Kombination mit einem Pax-Counter (Make 1/19, S. 32) oder zusammen mit einer lokalen Cloud. Über IFTTT können wir VoIP-Nachrichten versenden, wenn die Raumtemperatur zu niedrig und das Fenster zu lange

auf ist oder die CO₂-Konzentration zu hoch und damit das Infektionsrisiko. Code-Beispiele und Bauteillisten für Varianten mit Neopixelring oder 7-Segment-Anzeige haben wir online bereits vorbereitet (siehe Kurzinfo). Einen besonders schönen Nachbau hat Netzbastler Moritz Metz vor kurzem im Deutschlandfunk vorgestellt: Er hat unser Projekt als Inspiration für den CO₂narienvogel genommen, der bei schlechter Luft von der Stange kippt. —hch

NEU: c't DSGVO – was 2020 wirklich wichtig ist

c't DSGVO 2020 – Neuauflage!

Auf 148 Seiten erfahren Sie, was 2020 wirklich wichtig ist: Ende der Schonfrist, DSGVO in der Praxis, Bußgelder, aktuelle Urteile und Umsetzung der Richtlinien. Dazu: FAQs, Anleitungen, Checklisten, Muster, Video-Tutorials für Admins und Vorlagen für Datenauskünfte.

Auch digital mit DVD-Download erhältlich!

shop.heise.de/dsgvo20

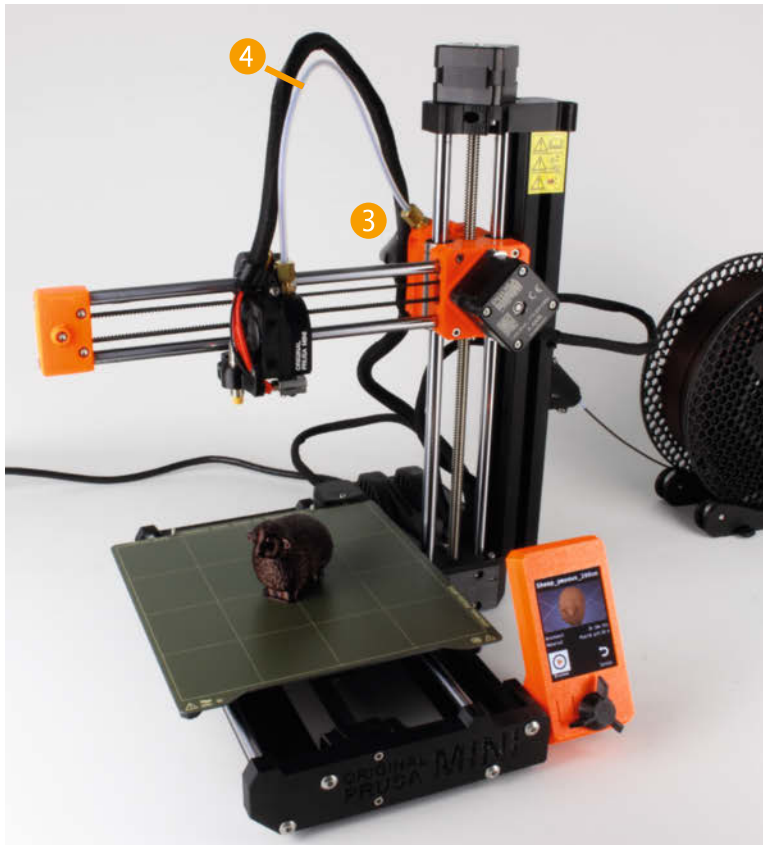
19,90 € >

NEU



heise shop

shop.heise.de/dsgvo20 >



3D-Einsteigerdrucker Prusa Mini

Der Mini ist der kleine Bruder des Original Prusa i3 MK3S, den wir in Make 1/20 (S.120) begeistert getestet haben. Wer keine ganz so große Maschine braucht und weniger als 400 Euro ausgeben will, findet jetzt im Mini den idealen 3D-Drucker – gerade auch als Einsteiger.

von Peter König



Den Prusa Mini bekommt man für 379 Euro direkt im Online-Shop des Herstellers Prusa Research. Während man sich bei den größeren FDM-Druckern aus dem gleichen Haus entscheiden kann, ob man ein Fertiggerät oder einen Bausatz kauft (und über 200 Euro spart), wird der Mini stets als ein Mittelending geliefert. Sprich: Man muss ein paar vormontierte Baugruppen zusammenmontieren, wobei man insgesamt vier (!) Schrauben eindrehet und ein paar Kabel einsteckt. Dass wir am Ende dann aber doch vom Auspacken über die Montage und das erste Kalibrieren bis zum ersten 3D-Druck gut anderthalb Stunden beschäftigt waren, liegt daran, dass der Zusammenbau an manchen Stellen doch etwas Geduld erfordert, etwa wenn das Flachbandkabel des Displays durch einen schmalen Schlitz gefummelt oder mit einer langen Gewindefschraube hinter einem ausladenden Gehäuse in einen frei rutschenden Nutenstein gezielt werden muss. Dank angeformter Anschläge gibt es aber kein Rätselraten in Bezug auf die vorgesehene Position, man kann da nicht viel falsch machen. Die deutsche Bauanleitung liegt gedruckt bei und lässt keine Fragen offen, wenn auch die Bilder durch das A5-Format ein wenig klein geraten sind. Wer lieber Videos schaut, kann sich aber auch auf YouTube den Aufbau von Joe Prusa persönlich zeigen lassen – das dauert keine Viertelstunde (siehe Link).

Startbereit

Beim ersten Start führt der Drucker einen Selbsttest durch, um mögliche Fehler zu finden, die beim Aufbau passiert sind. Der Test wird auf dem großen Farbdisplay ausführlich dokumentiert, dessen Menü sich auf diverse Sprachen einstellen lässt, unter anderem auf Deutsch. Zur Steuerung des Druckers gibt es ein einziges Dreh-Drück-Rad, das mit seinen drei Armen etwas gewöhnungsbedürftig gestaltet ist, erinnert es doch zusammen mit dem Display ein klein wenig an einen altertümlichen Safe aus einem Cartoon. Aber es lässt sich gut bedienen und man läuft keinesfalls Gefahr, versehentlich zu drücken, wenn man nur drehen will (was bei den eleganter gestylten Dreh-Drück-Rädern anderer Drucker durchaus passiert).

Gedruckt wird auf eine mit PEI beschichtete Bauplattform aus Federstahl, die von Magneten auf dem eigentlichen Drucktisch gehalten wird und sich auf 100 °C aufheizen lässt. Die Drucke halten darauf zuverlässig und lassen sich gleichzeitig hinterher einfach lösen, indem man die abgenommene Platte biegt. Vor jeden Druck wird die Bauplattform an 16 Stellen automatisch eingemessen, damit eventuelle Unebenheiten ausgeglichen werden können. Der Druckkopf lässt sich auf 280 °C aufheizen, der Hersteller nennt als unterstützte Druckmaterialien unter anderem



PLA, PETG, ASA und ABS. Der Bauraum misst 18cm in alle drei Richtungen.

Innere Werte

Der Drucker ist im Betrieb so leise, dass ihn sogar die (zugegeben etwas aufdringliche) Lüftung im Make-Labor übertönt. Prusa baut in das Gerät speziell leise Treiber für die Schrittmotoren ein. Im Test erwies sich die Druckqualität ① ② praktisch als ebenso hervorragend wie beim großen Prusa i3 MK3S (Details in unserer Online-Tabelle), auch wenn sich bei den Standardeinstellungen bei 0,15mm Schichtdicke gelegentlich dünne Materialfäden zwischen Objekten zeigten, wenn man mehrere parallel druckt.

Der Mini hat einen Bowden-Extruder, das heißt, der Vorschubmotor fürs Material sitzt am Rahmen des Druckers (in diesem Fall am Kreuzungspunkt von X- und Z-Achse ③) und

wird durch einen Teflonschlauch in den eigentlichen Druckkopf geführt. Trotz des je nach Druckkopfposition recht engen Biegeradius dieses Schlauchs ④ konnten wir problemlos einen Roboterreifen aus dem gummiartigen Material Ninjaflex drucken (Shore-Härte 85A), der in der Mitte der Bauplattform lag – bei Bowden-Extrudern in anderen Druckermodellen klappt die Verarbeitung von solch flexiblem Material oft nicht problemlos.

Sonderausstattung

Die mit dem kostenlosen Prusa Slicer (Make 4/20, S.90) vorbereiteten Druckdateien bringt man entweder über einen USB-Stick (statt der allgemein üblichen SD-Karte) oder per Ethernet-Kabel auf den Drucker. Bei der Auswahl einer Druckdatei zeigt das Display eine schicke 3D-Voransicht – sehr praktisch, wenn man sich nicht mehr genau erinnert, welches Modell sich hinter welchem Dateinamen verbirgt. Wer drahtlos seine Vorlagen auf den Drucker funken will, braucht als Upgrade ein WLAN-Modul auf ESP32-Basis, was zwar schon angekündigt ist, es aber noch nicht zu kaufen gibt.

Für 20 Euro Aufpreis kann man einen optischen Filamentsensor mitbestellen, der den Druck unterbricht, sobald das Material ausgeht. Schiebt man frisches nach, kann der Druck weitergehen. Als optionales Zubehör für 30 Euro gibt es auch eine Bauplattform mit rauer statt glatter Oberfläche, was den Werkstücken auf der Grundfläche eine eigenwillig strukturierte Optik verleiht, die auch bei diversen Bauteilen des Druckers selbst zu entdecken ist – diese Teile werden auf anderen Prusa-Druckern produziert. Die raue Oberfläche soll zudem bei manchen Materialien die Haftung auf dem Druckbett verbessern.

Fazit

Anfang des Jahres war noch der Prusa i3MK3S der beste FDM-Drucker, den wir je im Testlabor hatten – jetzt muss er sich den ersten Platz mit dem Prusa Mini teilen. Denn mit diesem gelingen 3D-Drucke genauso einfach und genauso gut und dank des Farbdisplays mit der Datei-Voransicht sogar noch einen Tick komfortabler.

Der Bauraum des Mini ist zwar nur halb so groß wie der des i3, sein Preis ist es aber auch. Die etwas niedrigeren Maximaltemperaturen

für Kopf und Bett sowie der Bowden-Extruder schränken die Materialauswahl zwar etwas ein, aber eigentlich nur im direkten Vergleich mit dem i3 MK3S aus dem gleichen Haus, denn der Mini wird immer noch deutlich heißer als viele Drucker anderer Hersteller. Und das bei einem Preis, der gerade mal 100 Euro über dem eines Billig-Klassiker wie dem Creality Ender 3 V2 (Make 4/20, S. 116) liegt.

Ein wenig unpraktisch ist der würfelförmige Bauraum mit 18cm Kantenlänge – ein Werkstück, was in der Breite nicht passt, passt auch längs nicht, mit einer Plattform von 16cm x 20cm wäre man vielleicht ein Stück flexibler (wir wollten etwa die Teile eines OpenScan Mini drucken, doch der zentrale Ring der Konstruktion passt in keiner Ausrichtung auf die Plattform). Aber das ist wirklich Jammern auf hohem Niveau, denn eigentlich ist das der 3D-Drucker, den man haben will – speziell für Einsteiger ist er ideal, weil er vom Fleck weg sehr schöne Drucke zu einem vernünftigen Preis liefert.

—pek

Alles zum Artikel
im Web unter
[make-magazin.de/x2h9](https://www.make-magazin.de/x2h9)



Bücher für Alles-Erfinder

IoT-Projekte mit dem ESP32, Roboter-Autos, maschinelles Lernen mit dem Arduino – unsere Autoren erleichtern Ihnen den Einstieg in die Maker-Welt. Mit allen Grundlagen zu Linux, Programmierung und Elektrotechnik. Hochexplosiver Zündstoff für Ihre Ideen!

Alle Handbücher auch als E-Book und im Bundle

www.rheinwerk-verlag.de

© Copyright by Maker Media GmbH.

 Rheinwerk
Technik

Snapmaker 2.0: Der Fräs-Gravier-3D-Drucker

Für das äußerst erfolgreiche Kickstarter-Projekt Snapmaker gibt es gleich drei „2.0“-Nachfolger: Die Geräte A150, A250 und A350 unterscheiden sich nur in Baugröße und Bauraum. Eines der ersten Vorab-Exemplare des Snapmaker A350 erreichte uns zum Test.

von Carsten Meyer



Geblieden ist das modulare Konzept: Das Gerät ist gleichzeitig 3D-Drucker, Laser-Gravierer und CNC-Fräse. Beim A250 und A350 ist die Werkzeugtraverse nun als Portal ausgeführt, der A150 und der Vorgänger (jetzt als „Snapmaker Original“ bezeichnet) müssen mit einer „einarmigen“ Konstruktion auskommen. Der A150 (160mm x 160mm x 145mm Bauvolumen) entspricht in etwa dem Snapmaker Original. Dann gibt es den A250 (230mm x 250mm x 235mm), der von der Größe her in etwa dem 3D-Drucker Prusa i3 MK3S vergleichbar ist, und schließlich den großen A350 (320mm x 350mm x 330mm), der schon mit einem großen Creality CR-10 gleichzusetzen ist.

Das doppelte Z-Portal wurde schon wegen des erheblich größeren Bauraums nötig, auch der Y-Schlitten (beim Snapmaker wird statt des Portals das Werkstück selbst bewegt) wird nun von zwei Lineareinheiten angetrieben. Die Kopplung ist rein elektrisch, die Schrittmotoren sind einfach parallel geschaltet.

Würde Apple 3D-Drucker bauen, wären sie wahrscheinlich ebenso sorgfältig und ansprechend zusammengestellt wie der Snapmaker: Alle Baugruppen sind einzeln verpackt, jedes Schraubentütchen ist beschriftet, Werkzeug zum Zusammenbau ist ebenfalls im Lieferumfang. Ein Sortimentskästchen nimmt die beim Umbau anfallenden übrigen Teile auf. Im Unterschied zu Apple liegt aber eine gedruckte A4-Hochglanz-Anleitung bei.

Ikea-Effekt

Laut Ikea-Effekt bringt man etwas selbst Zusammengebautem mehr Wertschätzung entgegen als einem Fertigprodukt. Snapmaker hat das Verhältnis von vorgefertigten Teilen und Eigenanteil gut abgestimmt; die Endmontage ist nicht schwierig, aber auch nicht allzu trivial. Für den grundsätzlichen Aufbau (einschließlich Auspacken!) benötigten wir eine gute Stunde, das Umrüsten für den jeweiligen Verwendungszweck nimmt dann nochmal 10 bis 15 Minuten in Anspruch.

Die Teile sind äußerst präzise gefertigt, alle Aluminium-Oberflächen sauber eloxiert und durchaus robust, sichtbar etwa an der schweren, verwindungssteifen Druckguss-Grundplatte – kein Vergleich mit den Billig-Maschinen von AliExpress: 3D-gedruckte Plastikteile oder roh gesägte Alu-Profile findet man beim Snapmaker nicht.

Bemerkenswert sind die gekapselten Lineareinheiten: Ein flexibler Edelstahlstreifen deckt die Führung so ab, dass nirgendwo Staub oder Späne eindringen können. Die Schrittmotoren sind innenliegend, ebenso die Endschalter. Die Verkabelung beschränkt sich auf ein paar leicht zu montierende Spezialkabel.

Zur Inbetriebnahme muss man nun noch die Steuerungssoftware *Snapmaker Luban 3.8.0*

herunterladen – nachdem man herausgefunden hat, dass das im Handbuch noch erwähnte Programm *Snapmakerjs* durch ebendiese ersetzt wurde. Snapmaker Luban ist gleichzeitig Slicer, G-Code-Player und CAM-Programm. Es wirkt mit diesen vielen Aufgaben etwas überfrachtet, dazu aber später.

3D-Druck

Für den 3D-Drucker-Betrieb montiert man das beheizte Druckbett (wobei wir die hierfür nötigen 22 Schrauben etwas übertrieben finden) und den Filament-Druckkopf für 1,75mm-Filament, der von weiteren vier Schrauben gehalten wird. Verbesserungsvorschlag: Ein paar Passstifte würden die bei jedem Umbau nötige Kalibrierung der Z-Höhe erübrigen. Die Druckbett-Nivellierung über 20 Punkte geht dank des integrierten Abstandssensors aber recht flott vonstatten. Lediglich der letzte Kalibrierpunkt muss manuell angefahren werden, als Abstandslehre dient hierbei ein Blatt Papier.

Das Aufheizen des Druckbetts dauert etwa fünf Minuten, der Druck selbst bereitet mit Snapmaker Luban keine Probleme. Die Grundeinstellungen führten zu wenig rekordverdächtigen Druckzeiten, auch wenn die Ergebnisse auf Anhieb überzeugen konnten – bei kleineren Objekten störte lediglich die etwas grobe Oberflächenstruktur. In der „feinen“ Voreinstellung gelingen dem Snapmaker makellose Oberflächen – allerdings auf Kosten der Druckzeit, die wegen der auf 0,08 mm halbierten Schichtdicke auf rund das Doppelte steigt.

Die Filament-Zuführung geschieht direkt von der Rolle zum Druckkopf; das hat leider zur Folge, dass bei der initialen „Nullfahrt“ einzelne Windungen von der Rolle springen können und sich das Material verheddert. Eine einfache, an passender Stelle montierte Drahtschleife würde das verhindern.

Lasereinheit

Eine bemerkenswerte Verbesserung gegenüber dem Vorgänger gab es beim Lasermodul: Statt des schwachbrüstigen 200mW-Lasers gibt es nun eine 1,6W-Diode, womit sich Gravurarbeiten nun in einem Viertel der Zeit erledigen lassen. Wunder darf man von der gestiegenen Laser-Leistung aber immer noch nicht erwarten: Selbst 2mm dünnes Sperrholz benötigt zwei Durchgänge, um es zuverlässig zu durchtrennen. Für Papp- oder Folienschablonen reicht die Leistung indes aus. Das blaue Licht eignet sich übrigens nicht zum Gravieren und Schneiden von transparenten Materialien. Wer farbloses Plexiglas schneiden will, muss nach wie vor zu einem CO₂-Laser greifen.

Neu im Lasermodul ist die eingebaute Kamera, mit der sich das Werkstück millimetergenau positionieren lässt, um beispielsweise eine Gravur an einer genau bestimmten Position aufzubringen. Die Firmware nutzt sie, um



Die fertig aufgebauten Lineareinheiten sind rasch mit den M4-Edelstahlschrauben montiert; sie werden mit fertig konfektionierten Spezialkabeln mit der Steuerung verbunden. Die Anleitung ist dabei eine gute Hilfestellung.

den Fokus nach Werkzeugkopfwechsel schnell einzustellen. Die Kamera lässt sich zur Positionierung allerdings nur im WLAN-Betrieb nutzen; man muss einen „Schnappschuss“ des Werkstücks anfertigen, diesen dann in Snapmaker Luban importieren und sein Motiv dann wie gewünscht darauf positionieren.

Für den Laser-Betrieb ist der Drucktisch gegen vier gerippte, schwarz eloxierte Aluminiumtafeln zu tauschen, die den durch das Werkstück gedungenen Laserstrahl „vernichten“. Für den Betrieb des Klasse-4-Lasers ist unbedingt die dem Gerät beiliegende Schutzbrille zu verwenden; der Laser ist (ohne Brille) noch aus einiger Entfernung gleißend hell, und vagabundierende Strahlen können zu ernststen Augenschäden führen.

CNC-Betrieb

Im Vergleich zum Snapmaker Original ist der Fräskopf deutlich aufgewertet worden: Statt der popeligen Bohrfutter, wie man sie von Mini-Bohrmaschinen kennt, gibt es nun eine präzise ER-11-Spannzange, in der Größe vergleichbar mit der von kleinen handgeführten Oberfräsen. Mitgeliefert wird allerdings nur ein Einsatz für 1/4"-Werkzeuge (3,175mm Schaftdurchmesser). Die Frässpindel weist kein fühlbares Spiel auf.

Als maximale Drehzahl gibt der Hersteller 12.000 Umdrehungen pro Minute an, was zum Isolationsfräsen und Bohren von Platinen gerade so ausreicht. Der DC-Motor dürfte nach unserer Schätzung etwa 60W haben, genaue Angaben zur Spindel finden sich leider nicht.



Das Druckbett oder die Opferplatte zum Fräsen ruht auf einem verwindungssteifen Skelett aus Alu-Druckguss.

Snapmaker liefert Spannmaterial und eine MDF-Arbeitsplatte mit, die sich als „Opferplatte“ leicht auswechseln lässt.

Im Unterschied zum 3D-Druck gibt es hier (noch) keine automatische Nivellierung, was beispielsweise beim Isolationsfräsen extrem sinnvoll wäre; kleinere Platinen sollten mit sorgfältiger Einstellung der Frästiefe aber möglich sein.

Gelungen fanden wir die mitgelieferte MDF-„Opferplatte“ (dafür ist sie fast zu schade) mit aufgedrucktem Zentimeter-Raster, die für die mitgelieferten Spannpratzen genügend Befestigungsbohrungen bereithält. Durch den

üppigen Z-Weg könnte auch ein flacher Maschinenschraubstock montiert werden, der die Einspannung kleinerer Teile erleichtert.

Der CNC-Abschnitt im Handbuch schaut noch etwas dürrig aus, doch gerade bei der spannenden Bearbeitung gibt es reichlich Erklärungsbedarf. Wer hier schlampft, wird umgehend mit zerstörten Werkzeugen oder Werkstücken bestraft.

Das Fräsen macht nicht nur deutlich mehr Lärm als der 3D-Druck, es produziert auch Späne und Staub. Ein Staubsauger sollte also in greifbarer Nähe sein, und man sollte über das geschlossene Gehäuse als nächste Anschaffung

nachdenken, das der Hersteller als Zubehör anbietet. Wir vermissten auch einen Not-Aus-Knopf, denn das Fräsen ist (ebenso wie das Arbeiten mit dem Laser) recht unfallträchtig.

Software

Momentan sieht die CAM-Unterstützung seitens Snapmaker Luban noch etwas dürrig aus. Unterstützt werden 3D-Modelle von Fusion360 (Autodesk), weitere G-Code-Postprozessoren sollen in Vorbereitung sein. Gegenüber der Software für den Original-Snapmaker hat sich leider nicht viel getan. So wird der Werkzeugdurchmesser bei der Anzeige der 3D-Vorschau nicht berücksichtigt (das schaffen inzwischen auch CAM-Programme für den Amateur-Bereich), und bei manchen Maschinen-Einstellungen wird kein G-Code erzeugt - mit nicht näher konkretisierter Ursache.

Der Luban-CNC-Editor bietet als Import nur die Vektorformate DXF und SVG an, womit sich zumindest Teile aus flachem Material ausfräsen oder gravieren lassen (2,5D). Zusätzlich kann man im Editor-Modus Rechtecke, Ellipsen/Kreise und Text hinzufügen; bei allen Objekten lässt sich über den „Process“-Modus eine Werkzeugdurchmesser-Korrektur (Konfiguration „Pfad“) hinzurechnen.

Der Editor importiert zwar Bilder (BMP, JPG oder PNG), die dann als Grauwerte-Relief gefräst werden, aber echte 3D-Dateien (z.B. im STL-Format) oder die zum Platinenfräsen wichtigen Bohrdaten (üblicherweise im Excellon-Format) lassen sich nicht einfügen. Auch ein Werkzeugwechsel im laufenden Betrieb ist nicht vorgesehen.

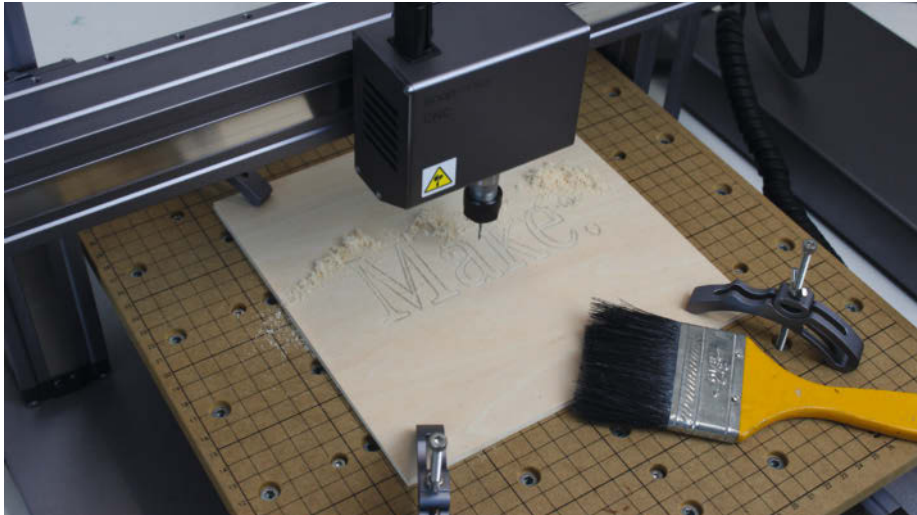
Möglicherweise ist eine gemeinsame Software für alle drei Einsatzzwecke doch nicht das Gelbe vom Ei: Funktionen und Einstellungen, die für den 3D-Druck gebraucht werden,



Die hinteren drei Testobjekte wurden in der etwas grob auflösenden Standard-Einstellung (0,16mm Schichtdicke) gedruckt, die Druckzeit betrug für jedes Objekt mehr als eine Stunde. Beim fein aufgelösten Makey vorn ist die Schrift besser lesbar, dafür war er nach einer Stunde erst zur Hälfte fertig.



Im Lasermodul ist nun eine 1,6W-Diode verbaut. Die Kamera dient zum Positionieren des Gravurmotivs.



Die kompakte Frässpindel ist mit einer hochwertigen Spannzange in der Größe ER-11 ausgestattet. Der Rundlauf war bei unserem Testgerät mit über 20µm „Schlag“ am Werkzeugschaft nicht ganz einwandfrei.



Die lokale Steuerung mit dem Android-Touchscreen ist äußerst komfortabel, allerdings sind Anpassungen während des Betriebs hiermit nicht möglich. Das 320W-Netzteil rechts ist übrigens mit einer dezent pulsierenden Innenbeleuchtung versehen – wir hätten uns aber lieber einen Not-Aus-Schalter gewünscht.

stören beim Lasergravieren oder beim CNC-Fräsen – und umgekehrt. Etwas verwirrend ist auch, dass sich Snapmaker Luban eigene Einstellungen nur dann merkt, wenn man jedes Mal ein neues Profil dafür anlegt.

Das auf Android basierende Touchscreen-Bedienteil lässt sich nur sinnvoll nutzen, wenn keine WLAN- oder Kabelverbindung aktiv ist; man kann damit keine Druck- oder Gravurparameter während des Betriebs einstellen. Gegenüber den vierzeiligen Displays vieler 3D-Drucker hat die ergonomische Touch-Bedienung allerdings in jeder Hinsicht die Nase vorn.

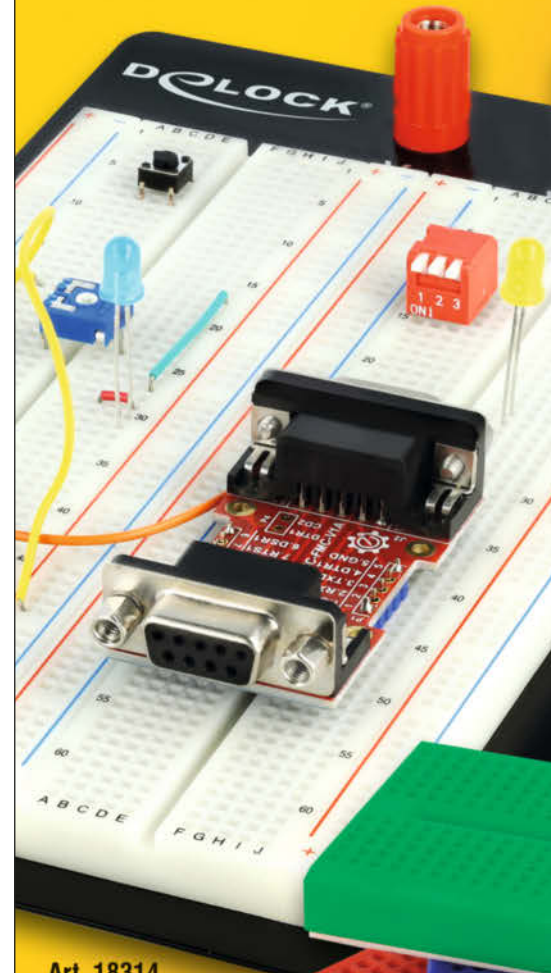
Insgesamt ist der neue Snapmaker 2.0 A350 ein durchaus gelungenes und her-

vorragend verarbeitetes Gerät, bei dem der Fokus eindeutig auf den 3D-Druck gelegt wurde. Schnitt- und Gravurleistung des Lasers sind gegenüber den Vorgänger deutlich gestiegen, mit reinrassigen Lasercuttern kann das Gerät aber nicht mithalten. Beim CNC-Fräsen sind die Möglichkeiten dagegen eher durch die Software begrenzt, hier ist noch Luft nach oben. —cm

Hersteller
URL
Preis

Snapmaker
www.snapmaker.com
1559 Euro bei Vorbestellung,
Auslieferung Dezember 2020

FÜR MAKER: STECKBOARDS



Art. 18314
mit Bodenplatte

Art. 18316-22
Mini in 7 Farben

Gute Idee

Steckbretter

für lötfreie Schalttests +
Versuchsaufbauten +
PCB- & Bastelprojekte +
Labor | Industrie | Hobby

So geht Heimcomputer im Jahr 2020

Selbstbau-Computer gibt es wie Sand am Meer, aber die meisten kommen nicht übers Prototypen-Stadium hinaus. Oft sind sie ein Haufen Kabelsalat auf Breadboards, der von leidlich guter Software angetrieben wird. Dagegen spielt der Colour Maximite 2 (CMM2) in einer ganz anderen Liga.

von Maik Schmidt

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



Der CMM2 ist ein Klein-Computer zum Selberbauen, der aber professionellen Maßstäben genügt. Das liegt insbesondere daran, dass seine Entwickler über ein gehöriges Maß an Erfahrung verfügen und in der Vergangenheit schon einige erfolgreiche Elektronik-Projekte auf die Beine gestellt haben.

Genau genommen entstammt der CMM2 einer ganzen Linie von Maximite-Computern. Es gab nämlich schon einen Monochrome Maximite und einen Original Colour Maximite (siehe auch c't Hacks 4/13, S. 116, Basic-Touchpanel-Computer). Beide waren ebenfalls ausgereifte Mini-Computer, die aber deutlich weniger leistungsstark waren.

Die Hardware

Das aktuelle Modell schöpft aus dem Vollen und basiert auf der STM32H743IIT6-MCU von STMicroelectronics. Die beherbergt eine ARM-Cortex-M7-CPU, die mit 32 Bit operiert und mit bis zu 480MHz taktet.

Auf dem Board gibt es 2MB Flash-Speicher, um persistente Daten und Programme abzuliegen und dazu noch 1MB RAM. Das war dem CMM2-Entwickler aber nicht genug und so hat er weitere 8MB externes RAM spendiert. Dieses ist hauptsächlich für Datenstrukturen in Programmen und für Video- und Audio-Daten gedacht.

Der zusätzliche Speicher hilft enorm, denn der CMM2 liefert gestochen scharfe VGA-Grafik in hoher Auflösung (800 × 600 Pixel) mit bis zu 65536 Farben. Die belastet den Prozessor in keiner Weise, weil sich ein separater LCD-TFT-Display-Controller darum kümmert.

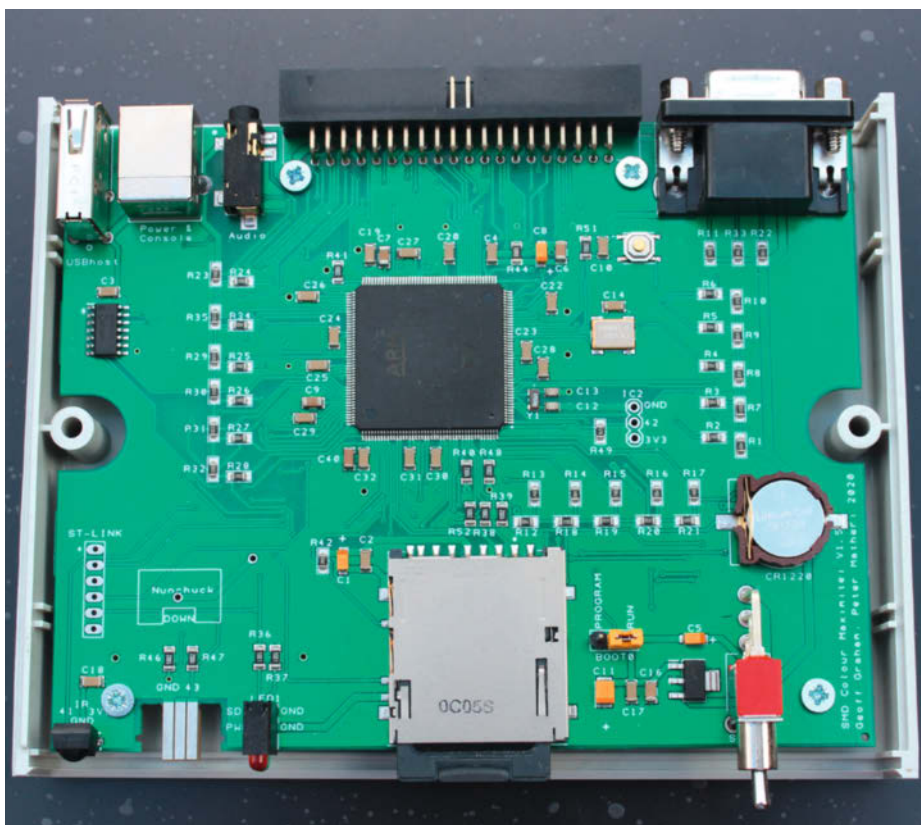
Zusätzlich gibt es einen Blitter, einen speziellen Logik-Controller, der Speicherbereiche rasend schnell verschiebt und jede Menge Sprites. Die Ausgabe knackiger Stereo-Sounds erfolgt über eine 3,5mm-Klinkenbuchse.

Als Eingabegerät dient eine USB-Tastatur und der CMM2 hat darüber hinaus einen Anschluss für einen Wii Nunchuk.

Der CMM2 bietet zu all dem manches Schmankele für Bastler. Unter anderem gibt es zwei UART-, zwei I²C-, zwei SPI- und eine One-Wire-Schnittstelle(n). Die werden alle zusammen mit weiteren Pins über einen Anschluss mit 40 Pins nach außen geleitet. 28 dieser Pins fungieren als digitale I/O-Pins und 12 kommen mit analogen Signalen zurecht.

Der Anschluss hat das Format einer IDE-Schnittstelle und ist kompatibel mit dem Raspberry Pi und den dazu gehörenden Pi-HATs.

Auf der Platine sind ferner Anschlüsse für einen IR-Empfänger/-Sender und einen DS18B20-Tempersensor vorgesehen. Übrigens gibt es zwei Varianten der Platine, nämlich eine, die mit SMD-Bauteilen bestückt wird, und eine bastelfreundliche für Through-Hole-Teile.



Die Platine des CMM2 ist aufgeräumt und übersichtlich.

Als Stromversorgung reicht ein USB-Anschluss, denn der CMM2 begnügt sich mit knapp 300mA. Eine Knopfzelle sorgt dafür, dass die Echtzeit-Uhr ihr Gedächtnis behält und sie hält auch einen 4KB großen RAM-Bereich am Leben, der zur Speicherung von

Optionen und Variablen dient. Im Regelbetrieb sollte die Batterie für drei bis vier Jahre reichen.

Etwas schade ist, dass der CMM2 keinerlei Netzwerk-Technologie, zum Beispiel Ethernet, WLAN oder Bluetooth, mitbringt.



Auf der Vorderseite bietet der CMM2 Platz für einen IR-Empfänger, einen Wii Nunchuk und eine SD-Karte.



Der GPIO-Anschluss dominiert die Rückseite des Geräts.

```

if tott=255 then
  if retr=1 then stopVGM: UnloadVGM
  Cls: Exit Sub
endif
endif
Pause(1)
Loop
if retr=1 then UnloadVGM
end sub

' SHOW MAIN CHARACTER
Sub PutSpriteChar(spr,x,y)
  Local XT,YT
  spr=spr+(PTYPEA-1)*40
  YT=FIX(SPR*17/386)*17
  XT=(SPR MOD 18)*17
  PAGE WRITE 11: BLIT XT,YT,X,Y,16,16,12,4
end sub

' SHOW MAIN CHARACTER
Sub PutSpriteEnemy(spr,x,y)
  'box x,y,16,16,1,rgb(255,0,0),rgb(255,0,0)
  'return
  Local XT,YT,PG
  if spr>=5000 then spr=spr-5000
  SELECT CASE spr
    CASE 0 TO 999: spr=spr-200:PG=5
    CASE 1000 TO 1999: spr=spr-1200:PG=4
    CASE 2000 TO 2999: spr=spr-2200:PG=9
  END SELECT
  PG=1
  YT=FIX(SPR*17/386)*17
  XT=(SPR MOD 18)*17
  BLIT XT,YT,X,Y,16,16,PG,4
end sub

' REFRESH STATUS PANEL WITH CHARACTER PHOTO FOR PLAYER N
Sub PanelStatus(type,n)
  Local X,Y,R5
  X=WRD,WRD2=34: Y=40+(n-1)*76
  PAGE WRITE 0: BLIT (PTYPEA-1)*34,170,X,Y,30,30,12,4
  BOX X-33,Y+2,0,30,0,rgb(0,0,0),rgb(0,0,0)
  BOX X-13,Y+2,0,30,0,rgb(0,0,0),rgb(0,0,0)
  PrintG(x-38,y+3,STR$(Cmag))
  PrintG(x-38,y+13,STR$(Csh))
  PrintG(x-38,y+23,STR$(Cfig))
  PrintG(x-18,y+3,STR$(Cspd))
  PrintG(x-18,y+13,STR$(Carm))
end sub
Gauntlet.bas= 2437 Lines
ESC=Quit, F1=Save current, F2=Run, F3=Find, F5=Select, FV=Paste

```

Der eingebaute Editor reicht für viele Zwecke völlig aus.

Immerhin gibt es schon YouTube-Videos, in denen findige Bastler WLAN nachrüsten.

Einschalten

Der CMM2 hat kein Betriebssystem im eigentlichen Sinn und muss beim Einschalten nichts von irgendwelchen Datenträgern lesen. Daher startet er in der Regel schneller als der angeschlossene Monitor.

Nach dem Start wechselt das Gerät sofort in den BASIC-Interpreter, der geduldig auf Kommandos wartet. Beispielsweise gibt der Befehl MEMORY einen Überblick über die aktuelle Auslastung des Speichers und PRINT "Hallo" gibt den Text *Hallo* aus.

```

COMB - Tera Term VT
File Edit Setup Control Window Help
<DIR>
19:05 19-07-2020      23 Gauntlet
19:53 28-07-2020     1843 first.bas
14:42 08-05-2020     12709 ghost.bas
22:53 03-07-2020     45240 hybris-1ngame.mod
21:32 28-07-2020      132 line.bas
14:43 08-05-2020     14506 Part02.png
21:49 28-07-2020       55 stars.BAS
21:01 28-07-2020       48 template.bas
14:14 17-07-2020       32 test.bas

```

Der eingebaute Datei-Manager ist überaus praktisch.



Wie zu Zeiten des Commodore C64 grüßt der CMM2 mit dem Prompt des BASIC-Interpreters.

Nostalgische Gefühle kommen gewiss dann auf, wenn man den Befehl POKE BYTE &H2400000, 255 absetzt. Der färbt nämlich das Pixel ganz links oben auf dem Bildschirm weiß. Ja, der CMM2 hat eine statische Memory-Map, die zu Peek & Poke-Experimenten wie in guten alten Heimcomputer-Zeiten einlädt.

Im Gegensatz zum Vorgänger hört der Spaß hier aber schon fast auf, solange keine SD-Karte im Slot steckt. Der CMM2 setzt zwingend eine SD-Karte voraus, das heißt, ohne lässt sich nicht viel mit dem Gerät anstellen.

Immerhin akzeptiert der CMM2 an SD-Karten so gut wie alles, was nicht mehr als 128GB hat und mit FAT16, FAT32 oder exFAT formatiert ist. Es irritiert etwas, dass die Software unterschiedliche Timing-Schemata für den Zugriff auf die Karten vorsieht und daher nicht jede Karte gleich erkannt wird. Das äußert sich in der Meldung Check Disk am unteren Rand des Displays und zumeist schafft die Anweisung OPTION SD TIMING NORMAL Abhilfe.

Stöbern in Dateien

Die Arbeit mit Dateien auf der Kommandozeile geht einfach von der Hand, weil das eingebaute BASIC einige Befehle zur Verarbeitung von Dateien mitbringt. Beispielsweise listet ls alle Dateien und Verzeichnisse im aktuellen Verzeichnis auf und CHDIR wechselt in ein neues Verzeichnis.

Dabei ist zu beachten, dass diese Befehle tatsächlich BASIC-Befehle sind und daher den Konventionen der Sprache gehorchen müssen. In der Praxis bedeutet das, dass an vielen Stellen Anführungszeichen gesetzt werden müssen, die in herkömmlichen Kommandozeilen-Interpretern nicht notwendig sind. Um beispielsweise in ein Verzeichnis namens Test zu wechseln, muss die Anweisung CHDIR "Test" ausgeführt werden.

Für kleinere Operationen ist das kein Problem, aber bei der Suche nach Dateien auf vollen SD-Karten kann das schon mal lästig sein. Dankenswerter Weise verfügt der CMM2 über einen komfortablen Datei-Manager. Gestartet wird er mit der Anweisung FILES und bietet eine effiziente Navigation durchs Dateisystem mit den Cursor-Tasten.

Darüber hinaus gibt es eine Such-Funktion, Kopier- und Löschoptionen und die Möglichkeit, Dateien zu editieren oder direkt auszuführen.

Insbesondere die letzte Funktion erweist sich als überaus praktisch, weil der CMM2 erstaunlich viele Audio- und Grafik-Formate von Haus aus unterstützt. Formate wie WAV, FLAC und MP3 spielt das Gerät automatisch im Hintergrund ab. Selbst MOD-Dateien, die auf dem Commodore Amiga das bevorzugte Format waren, bereiten keine Probleme.

Ähnlich gut sieht es bei Bildern aus, die als BMP, GIF, JPEG oder PNG daherkommen können.

Zur Bearbeitung von Texten und Programmen steht ein Editor zur Verfügung, der Funktionen wie Ausschneiden, Kopieren und Einfügen beherrscht. Ferner bietet er Syntax-Highlighting für BASIC-Code und ist einigermaßen flott. Um mal eben kleinere Programme zu schreiben oder anzupassen, reicht er allemal.

Workflow

Umfangreichere Programme schreibt man besser mit den gewohnten Editoren auf dem PC und transferiert sie anschließend auf den CMM2. Dazu gibt es einige Möglichkeiten.

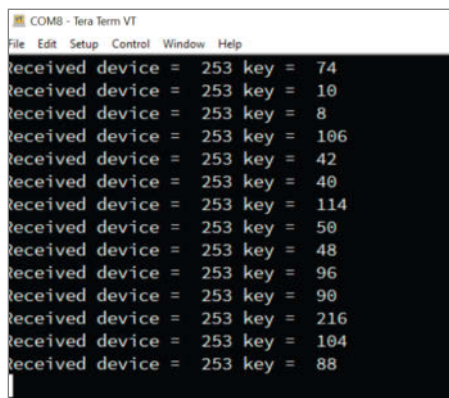
Selbstverständlich ist der Austausch über die SD-Karte eine Option, aber das ständige Wechseln vom PC zum CMM2 wird schnell lästig.

Besser ist da schon der Einsatz eines seriellen Terminals, wie zum Beispiel TeraTerm. Damit lässt sich der CMM2 komplett fernsteuern, das heißt, er benötigt nicht einmal eine eigene Tastatur. Jeder Tastendruck im seriellen Terminal wird auch sofort im CMM2 erkannt. Ferner gibt es die Möglichkeit, Dateien auf vielfältige Weise zu transferieren.

Noch etwas bequemer ist MMEdit, ein Text-Editor mit vielen nützlichen Funktionen, der sich mit allen Vertretern der Maximite-Familie verbindet. Damit kann man Programme auf dem PC editieren und sie per Knopfdruck auf dem CMM2 ausführen. Puristen bevorzugen vermutlich aber eher das blanke Metall.

Basisarbeit

Sobald die Umgebung eingerichtet ist, kann der Spaß beginnen und das bedeutet, BASIC-Programme zu schreiben.



Der Infrarot-Empfänger funktioniert mit vielen handelsüblichen Fernbedienungen.

In der Ära der Heimcomputer war das nichts Besonderes, aber heute ist die Sprache einigermaßen verpönt, weil sie zu unstrukturierter Programmierung verführt und den allseits verhassten Spaghetti-Code produziert.

Das BASIC des CMM2 hat glücklicherweise Weise nichts mit dem zu tun, was früher so auf C64, Schneider CPC & Co lief. Der Dialekt heißt MMBasic und er ist eine fast vollständige Implementierung des ANSI-Standards für Full Basic. Ferner ist es abwärtskompatibel zu Microsofts MBASIC.

MMBasic läuft nicht nur auf den Maximite-Rechnern, sondern auch auf vielen anderen Mikrocontrollern und auch auf Microsoft Windows. Das Projekt steht nicht unter einer Open Source-Lizenz, aber wer MMBasic für nicht-kommerzielle Zwecke einsetzt, bekommt Zugriff auf den Quelltext.

Zeilennummern sind in MMBasic zwar erlaubt, werden aber selten verwendet, weil die Sprache ordentliche Funktionen unterstützt und so nicht auf das leidige GOTO angewiesen ist. Ein Include-Mechanismus erlaubt die Verteilung von Code auf mehrere Dateien und mit ein wenig Disziplin bleiben so auch größere Programme wartbar.

Einer der großen Vorteile von MMBasic ist, dass es einfach zu erweitern ist. Daher spielt die CMM2-Version perfekt mit der Hardware zusammen und es gibt eigene Befehle für die speziellen Video-, Audio- und GPIO-Eigenschaften des Geräts.

Der Zugriff auf die GPIO-Pins ist mindestens so einfach wie beim Arduino. Zentral ist dabei der Befehl SETPIN, der die einzelnen GPIO-Pins konfiguriert. Beispielsweise macht die Anweisung SETPIN 16, DOUT den GPIO-Pin 16 zu einem digitalen Ausgabe-Pin. Insgesamt stehen sechs Modi zur Verfügung. Darunter sind Klassiker wie AIN (Analog Input), aber auch spezielle Modi, wie zum Beispiel FIN, der die Frequenz des Signals an einem Pin misst.

Den aktuellen Zustand eines Pins liest die Funktion PIN, das heißt, nach der Anweisung

Sonderheft Make Special PICAXE

Nicht nur Heimcomputer lassen sich in BASIC programmieren, auch Mikrocontroller wie der Picaxe steuern ihre Pins und Funktionen mit BASIC-Befehlen. Wie leicht das geht, zeigt unser Sonderheft und erklärt die Software, Hardware und die Befehle, die dafür nötig sind.

So gelingt die Programmierung eigener Projekte noch leichter als beim Arduino-C. Statt void() und Pointerarithmetik bietet das Picaxe-BASIC auch blutigen Anfängern verständliche Befehle mit vielen Funktionen zum Steuern der Hardware. Das Heft führt in die Grundlagen der BASIC-Programmierung und die interne Hardware des Chips ein. Anhand verständlicher und praktischer Projekte lernt man, die Befehle konkret einzusetzen.

Das Heft wird im Bundle mit einem Picaxe-Mikrocontroller und unserem eigens dafür entwickelten Programmier- und Test-Board *Nano Axe* geliefert. Damit lässt sich der Picaxe leicht per USB programmieren und debuggen.



Das Heft kostet 24,95 Euro und ist versandkostenfrei über den heise Shop zu beziehen.

`x = PIN(5)` enthält die Variable `x` den aktuellen Wert an GPIO-Pin 5.

Das Schreiben von Werten ist ähnlich simpel und so setzt `PIN(3) = 0` den GPIO-Pin 3 auf 0V. Für kompliziertere Ausgaben eignet sich die Funktion PULSE, die Pulse von bis zu wenigen Mikrosekunden Länge senden kann.

Die meisten Befehle akzeptieren als Parameter den Namen einer Callback-Funktion, die im Falle bestimmter Ereignisse aufgerufen wird.

Listing `ir_receiver.bas` zeigt das anhand eines Programms, das einen Infrarot-Empfänger kontrolliert. Der Code ist an Effizienz kaum

zu überbieten und beginnt mit einer IR-Anweisung, die den Empfänger initialisiert. Die legt gleichzeitig fest, dass Informationen über das sendende Gerät in die Variable `DevCode` gehören und der tatsächlich empfangene Code in der Variablen `KeyCode` landet. Darüber hinaus verknüpft der Aufruf den Empfänger mit der Funktion `IR_Int`. Die wird aufgerufen, wenn Daten empfangen wurden und sie gibt die Daten auch aus.

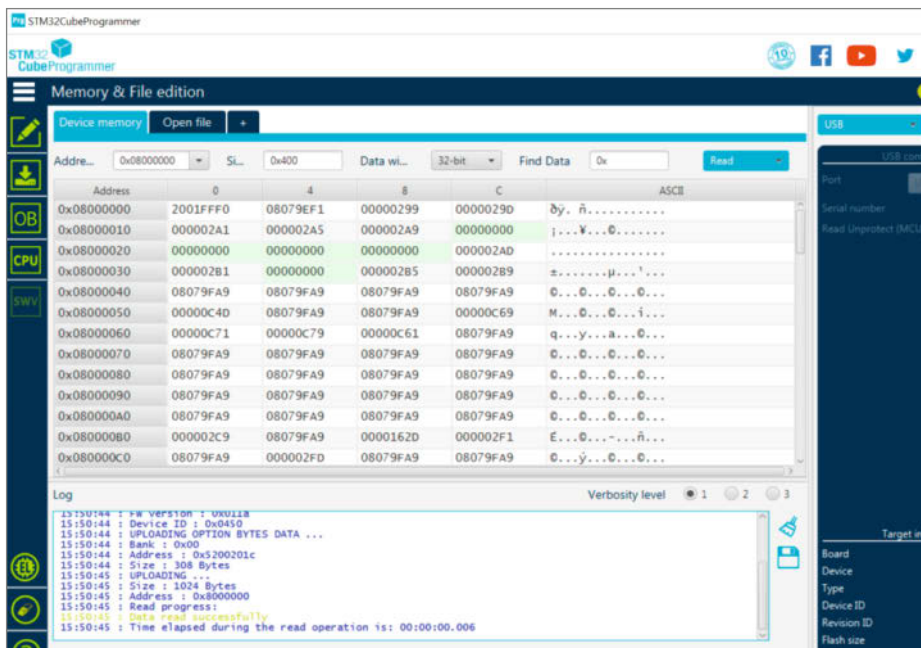
Dieses Programmier-Modell ist ungewöhnlich für BASIC, aber es passt hervorragend zu einem Gerät, das mit Sensoren aller Art operieren soll. Das Programm fragt nicht unentwegt nach neuen Informationen beim Sensor,

ir_receiver.bas

```
IR DevCode, KeyCode, IR_Int

DO
  ' Hier kann man irgendwas machen.
LOOP

SUB IR_Int
  PRINT "Received device = " DevCode " key = " KeyCode
END SUB
```



Die Installation einer neuen Firmware ist schnell erledigt.

sondern wird informiert, wenn es etwas Interessantes gibt. Das vereinfacht die Programmierung enorm.

Was geht sonst?

Um mal eben schnell ein kleines Elektronik-Projekt aufzusetzen oder ein paar Bausteine auszuprobieren, ist der CMM2 eine gute Plattform. Der BASIC-Interpreter fördert die Interaktion und die vielen speziellen Kommandos machen selbst komplexe Abläufe zu einem Kinderspiel.

Es gibt aber noch ein weiteres Anwendungsfeld, auf dem der CMM2 ganz weit vorne liegt, und das ist die Spieleprogrammierung. Es ist kein Zufall, dass die unscheinbare Box einen Nunchuk-Anschluss hat und die Grafik-Hardware schreit geradezu nach ein paar Action-Titeln.

Dafür ist wirklich alles da. Das System hat einen Blitter und jede Menge Sprites. Es unterstützt Multi-Buffering und liest problemlos gängige Grafik- und Sound-Formate. Bis zu drei Wii Nunchuk oder drei Wii Classic-Controller sorgen für Multiplayer-Spaß.



Ein inoffizieller Gauntlet-Port in interpretiertem BASIC steht dem Original in nichts nach.

Wenig überraschend gibt es auch schon die eine oder andere interessante Portierung und eine Umsetzung des Klassikers *Gauntlet* ist besonders beeindruckend. Es ist kaum zu glauben, dass dieses Spiel in interpretiertem BASIC läuft.

Geht immer noch besser

Die CMM2-Entwickler verbessern und erweitern die Firmware des Systems kontinuierlich. Zum Beispiel ist es seit kurzem möglich, in C geschriebene und übersetzte Erweiterungen als ARM-Assembler-Code in BASIC einzubinden, um die Performanz bei Bedarf noch weiter zu steigern.

Von diesen Verbesserungen profitieren CMM2-Nutzer ohne großen Aufwand, denn sie können neue Versionen der Firmware über den USB-Anschluss und mit kostenloser Software einspielen.

Die Software heißt STM32CubeProgrammer und kommt von STMicroelectronics. Jeder kann sie kostenlos für alle gängigen Betriebssysteme herunterladen, aber der Hersteller möchte zumindest die E-Mail-Adresse seiner Kunden kennen.

Auf dem CMM2-Board sind vor dem Firmware-Upload ein paar Vorbereitungen zu treffen. Unter anderem muss ein Jumper versetzt werden, der das Gerät in den Programmier-Modus versetzt. Dann kann die Aktualisierung über einen der beiden USB-Ports erfolgen.

Shut up and take my money!

Momentan gibt es noch keinen deutschen Händler, der den Colour Maximize 2 vertreibt und der nächste Shop befindet sich in England. Dort sind sowohl die vormontierte SMD-Variante mit Gehäuse für 85 britische Pfund als auch ein Selbstbau-Kit für 55 Pfund erhältlich.

Fazit

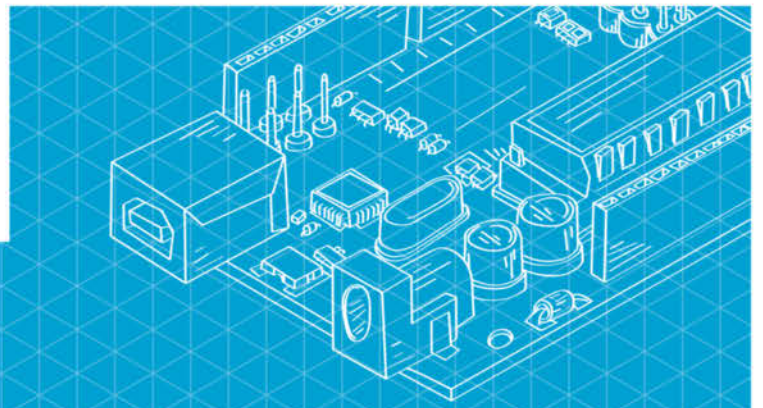
Mit dem CMM2 wird ein Traum für Nostalgie-Fans wahr. Einen solchen Computer wünschte sich jeder, der in den Achtzigern aufgewachsen ist. Kein langes Booten, sofort mit BASIC loslegen und trotzdem Spiele auf dem Niveau von ehemaligen Spielhallen-Titeln programmieren.

Das System überzeugt durch die Liebe zum Detail und wirkt trotz der vielen Funktionen nicht überfrachtet. Die ausgezeichnete (englische) Dokumentation und ein aktives Forum sorgen für schnelle Erfolgserlebnisse sowohl bei der Programmierung als auch bei der Realisierung von Elektronik-Basteleien.

Wenn man die Handbücher ausdruckt und mit einer Spiralbindung versieht, ist der CMM2 genau das Richtige für lange Winterabende.

—dab

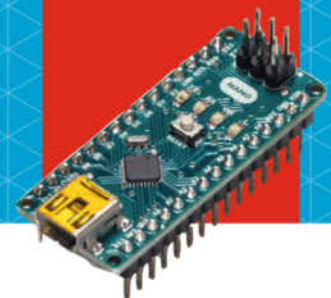
Make:



DAS KANNST DU AUCH!



GRATIS!



2× Make testen und 6 € sparen!

Ihre Vorteile:

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

Für nur 15,60 Euro statt 21,80 Euro.

* Für die Laufzeit des Angebotes.

Jetzt bestellen: make-magazin.de/miniabo

© Copyright by Maker Media GmbH.

Verteiltes Chaos

Der Chaos Communication Congress wird in diesem Jahr eine Remote Chaos Experience (rC3). Derzeit sind die Hackspaces aufgerufen, sich Ideen für lokale Events mit einem gemeinsamen Programm zu überlegen. events.ccc.de

Umzug in Mannheim

Das Mannheimer Raum-ZeitLabor hat endlich ein neues Zuhause gefunden und bleibt in Käfertal. Künftig wird es in der Weinheimer Straße 58-60 unterkommen. Zunächst muss allerdings noch renoviert werden.

twitter.com/RaumZeitLabor

Maker-Termine

ÜsiZUGkunft
28. Oktober – 11. November
FabLab Zug
fablab-zug.ch

Roboexotica
19.–22. November
Freiraum Wien
roboexotica.at

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

Veranstalten Sie selbst?

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

mail@make-magazin.de



Bild: Section77

Mitmachwerkstatt Offenburg eröffnet

Der neue Raum liegt direkt im Bahnhof

Mitte August fiel in Offenburg der Startschuss für die Mitmachwerkstatt des örtlichen Chaostreffs Section77. Zum Start waren bei einem Tag der offenen Tür unter besonderen Hygienebedingungen alle Interessierten eingeladen, den frisch renovierten und teilweise ausgestatteten Raum zu besichtigen und mit den Vereinsmitgliedern ins Gespräch zu kommen. Ein Jahr zuvor wurde die Mitmachwerkstatt gegründet, um auch längerfristige Projekte zu ermöglichen. Das was in der vorigen Situation als Mitnutzer der Gasträume eines Studentenlokals nicht möglich. Der neue Vereinsraum im nördlichen Teil des Offenburger Bahnhofsgebäudes stand zuvor längere Zeit leer und wurde von den Vereinsmitgliedern umfassend renoviert. Zu Beginn sind ein 3D-Drucker, ein Lötarbeitsplatz und eine Tischbohrmaschine eingezogen. Weitere Werkzeuge und Ma-

schinen aus den Bereichen Elektronik, 3D-Druck und CNC-Fräsen werden noch beschafft.

Falls künftig Platzmangel droht, ist bereits die zusätzliche Nutzung eines angrenzenden Raumes in Aussicht. Vereinsmitglieder haben rund um die Uhr Zugang zum Raum und dessen Ausstattung. Für interessierte Besucher finden in jeder ungeraden Woche eines Monats jeweils am Dienstag ab 20 Uhr die regulären, offenen Vereinstreffen statt. Zusätzlich gibt es alle zwei Monate Samstags ein „freies Hacken“ und einmal im Monat ein Tech-Event mit einem bestimmten Thema. Aufgrund der aktuell bestehenden Hygieneregeln ist derzeit die maximale Anzahl an Teilnehmenden begrenzt und eine Anmeldung vorab notwendig. —hch

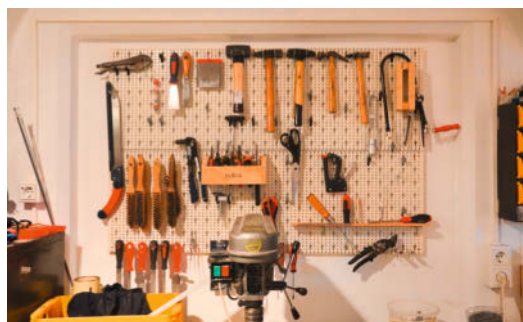
► section77.de

Reparaturmonat Oktober

Weltweite Veranstaltungen werben für das Reparieren

Der Oktober ist seit einigen Jahren der Zeitpunkt für Aktionen rund ums Reparieren und das deutschlandweite Vernetzungstreffen der Reparatur-Initiativen. Durch die Corona-Pandemie muss das Programm des Vernetzungstreffens dieses Jahr allerdings digital stattfinden. Daher sind vom 6. bis 25. Oktober eine Reihe an Webinaren und Arbeitsgesprächen im Internet geplant. Los geht es am 6. Oktober mit einem Gespräch darüber, wie Reparatur-Teams zusammenarbeiten und sich organisieren. In den weiteren Arbeitsgesprächen stehen die Frage, wie Selbsthilfe eigentlich funktioniert, was degrowth bedeutet sowie das Einbinden von Jugendlichen in Reparatur-Initiativen und das Recht auf Reparatur in Europa auf der Agenda. Außerdem sind zwei Webinare geplant, die Rohstoffe in der Elektronik und effizientes Reparieren beleuchten.

Rund um den Internationalen Tag der Reparatur, den 17. Oktober, wird es weltweit zusätzlich viele Veranstaltungen von lokalen Initiativen geben. Das diesjährige Motto lautet „Reparatur ist notwendig“ und soll auf die Bedeutung des Reparierens in Zeiten der



Pandemie hinweisen. Bereits im September starteten schließlich die ersten 3D-Reparatur-Stammtische, in denen der Einsatz von 3D-Druck in Reparatur-Cafés diskutiert wird. Die Idee zu den Treffen stammt aus dem Projekt „3D-Druck & Reparatur“, das vor einem Jahr mit einer Onlinesammlung von Ersatzteilmodellen für den 3D-Druck abgeschlossen wurde. —hch

► reparatur-initiativen.de

Neue Projektplattform in der Beta

Make: Projects

von Rebecca Husemann

Unsere neue Onlineplattform Make Projects ist gestartet. Helft uns, sie mit Leben zu füllen! Ihr findet sie unter makeprojects.com/de. Hier könnt ihr eure Projekte mit anderen teilen, die Anleitungen anderer Maker nachbauen und euch miteinander austauschen. Ladet euer Projekt hoch und zeigt allen, was ihr zu Hause selbst gebaut habt.

Manch einer erinnert sich sicher an die Rubrik *Community-Projekte* auf unserer Webseite, auf der ihr eure Maker-Projekte selbst veröffentlichen konntet. Mit Make Projects geht das jetzt viel einfacher.

Auch wir als Redaktion zeigen dort unserer kleineren Projekte und Anleitungen. Schaut mal auf unserem Profil @MakeMagazinDE vorbei. Viele ältere Make-Projekte könnt ihr dort bereits finden.

Wir lieben Open-Source: Schreibt Schritt-für-Schritt-Anleitungen auf Deutsch,

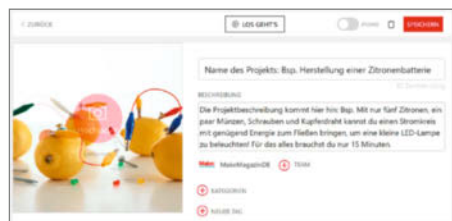
damit andere eure Projekte nachbauen können. Vernetzt euch weltweit: Gebt Feedback zu den Projekten anderer, tauscht euch aus oder helft euch gegenseitig während des Projektbaus weiter.

Veröffentliche dein Projekt auf Make Projects

Dein eigenes Projekt auf Make Projects mit anderen zu teilen, ist ganz einfach: Nach einem Klick auf das Plus-Symbol kannst du auswählen, ob du mit einem leeren Projekt starten oder eine der vorgegebenen Projektvorlagen benutzen möchtest. Danach kannst du verschiedene Teilprojekte für dein Projekt anlegen. Ob work-in-progress, Prototyp oder lang erprobtes Lieblingsprojekt – auf Make Projects ist es richtig aufgehoben. Zum Abschluss kannst du es der passenden Kategorie zuordnen. Fehlt deine Lieblingskategorie noch? Dann sag uns Bescheid. Wer etwas schüchtern ist, kann sein Projekt auch erstmal auf *privat* stellen und veröffentlichen, sobald es bereit für fremde Augen ist.

Mit Make Projects könnt ihr eure Ergebnisse allerdings nicht nur besonders einfach, sondern auch besonders ansprechend online stellen: Eure Listings werden automatisch erkannt und eingefärbt, ihr könnt Videos hochladen, blitzschnell Bildergalerien erstellen und andere Projekte mit Vorschau verlinken.

Wer bereits einen Account bei der englischsprachigen Version von Make Projects ange-



Das Projekt braucht natürlich einen Namen, ein Bild und eine kurze Beschreibung.

legt hat, kann diesen einfach weiterbenutzen und auf makeprojects.com/de auf Deutsch weitermachen.

Project Boards

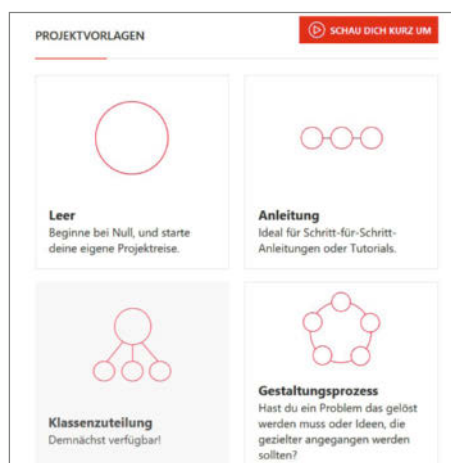
Mit den praktischen Project Boards könnt ihr in einer Gruppe sogar größere Projekte planen, Dateien hochladen und euren Fortschritt miteinander teilen. Hier seid ihr ganz unter euch und könnt gemeinsam kreativ sein: Nutzt das Project Board, um zusammen Projekte nach vorne zu bringen – egal, wo ihr auf dem Globus seid!

Wir freuen uns auf eure Projekte und euer Feedback!

—rehu



Mit einem Klick auf Plus könnt ihr loslegen.



Mit der leeren Vorlage könnt ihr euer Projekt ganz frei präsentieren, mit den anderen sind schon einige Abschnitte für euch vorbereitet.

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



Mit einzelnen Projektschritten ist es ganz einfach, auch bei umfangreichen Projekten die Übersicht zu behalten.

Maker Faire®

TECHNOLOGIE.
INNOVATION.
COMMUNITY.

Wie morgen, nur heute.

Präsentieren Sie Ihr Unternehmen auf der
Maker Faire, entdecken Sie neue Talente, Ideen und
bereichern Sie Ihre Unternehmenskultur.

Die Maker Faires sind die wichtigsten Plattformen der Maker-Szene. Als Aussteller präsentieren Sie sich technologiebegeisterten Menschen, treffen potentielle Mitarbeiter und schwimmen in einem Ideen-Pool. Sprechen Sie uns an, um schon bald die Unternehmenskultur von morgen zu leben.



Ihr Ansprechpartner

Marcel Ossenkop

Sales Account Manager

Tel.: +49 511 5352-133

maos@maker-media.de

Werden Sie Aussteller!

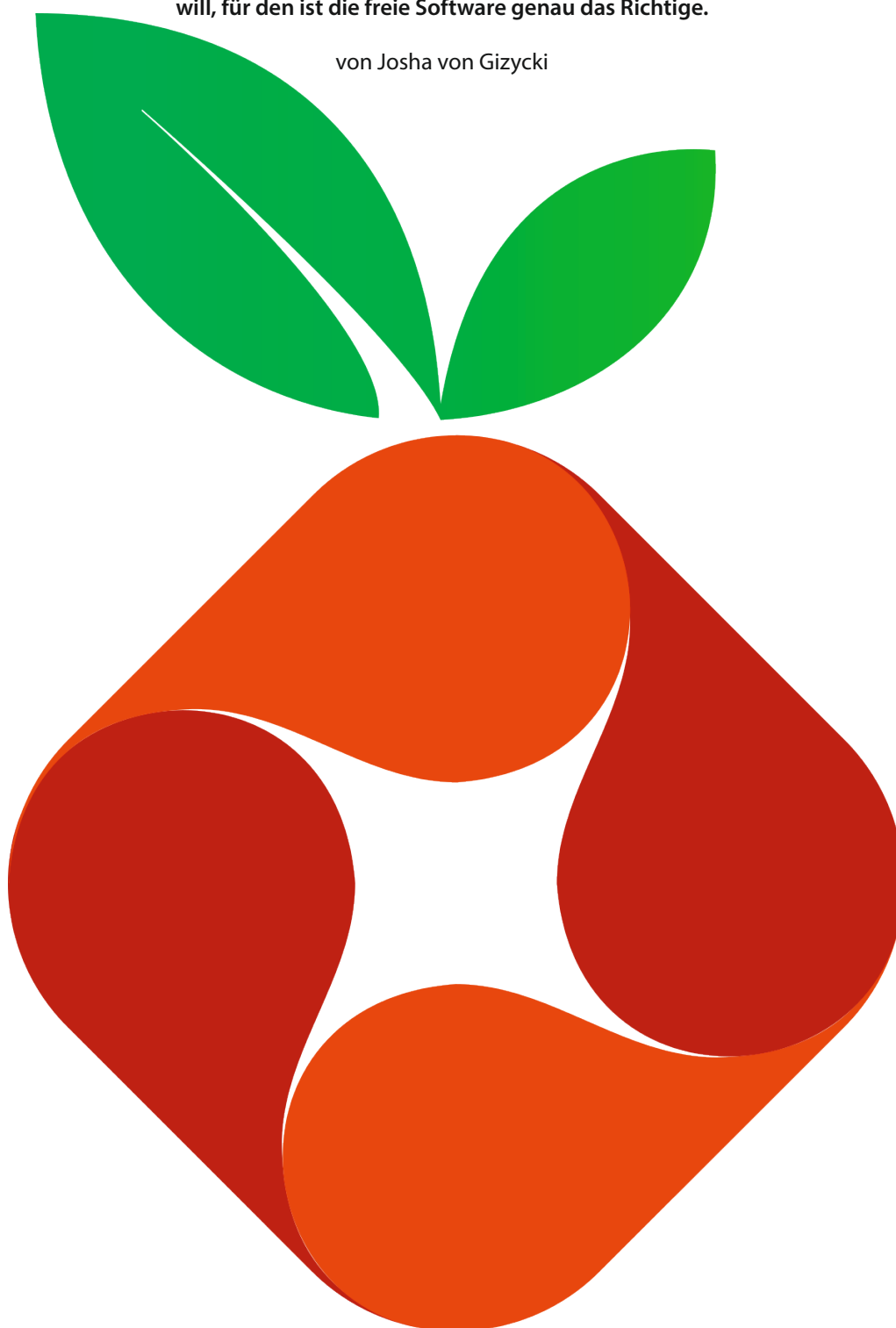
Lernen Sie alle Vorteile kennen:

© Copyright by Maker Media GmbH. www.maker-faire.de/Aussteller

Pi-hole: Das schwarze Loch für Internetwerbung

Pi-hole ist ein Werbeblocker für das heimische Internet, der auf dem Raspberry Pi läuft. Wer von Werbung beim Surfen genervt ist und den Trackern von Google und Co. ein Schnippchen schlagen will, für den ist die freie Software genau das Richtige.

von Josha von Gizycki



Nun kann sich jeder im Browser einen Ad-blocker installieren, der die meisten Probleme löst. Wie sieht es aber mit dem Smartphone aus? Mit dem intelligenten Kühlschrank, dem Arbeitslaptop, dem IoT-Device? Werbenetzwerke werden zudem nicht nur vom Webbrowser antelefoniert, sondern auch von anderen Programmen: Treibersuites, Chatprogrammen, sogar Windows selbst bindet Werbung ein. An dieser Stelle wünsche ich mir eine zentrale Lösung: Mein Netzwerk zuhause soll einfach gar nicht mehr mit Werbenetzwerken reden. Schaffe ich es also, dass mein Router, durch den ja sämtlicher Datenverkehr ins Internet fließt, die Kommunikation mit unerwünschten Servern unterbindet, habe ich mein Ziel erreicht.

Hierbei hilft dem datensparsamen Maker die freie Software *Pi-hole*. Durch die konkrete Installation von Pi-hole leite ich weiter unten, vorher erkläre ich das Funktionsprinzip.

DNS steht für *Domain Name System* und wird recht passend als das Telefonbuch des Internets beschrieben. DNS ist nötig, da der Zugriff auf Webdienste streng genommen nicht über eine Adresse, wie wir sie kennen, funktioniert. Stattdessen läuft er über die IP-Adressen der Server, auf denen die Webdienste bereitgestellt werden.

Gebe ich beispielsweise in meinen Browser `https://www.heise.de/make/` ein, startet mein Computer als Erstes eine Anfrage an einen DNS-Server, üblicherweise den meines Internetanbieters. Darüber kann er herausfinden, welche IP-Adresse hinter dieser Adresse steckt. Im Fall von `heise.de` ist dies die IPv4-Adresse `193.99.144.80`. Sollte meine Internetausstattung IPv6 unterstützen, kann ich `heise.de` auch unter der Adresse `2a02:2e0:3fe:1001:302::` erreichen. Damit kann mein Router ins Internet funken, um den angefragten Inhalt abzurufen.

Wenn ich die Kontrolle über einen eigenen DNS-Server besitze, ist es prinzipiell sehr einfach, DNS-Anfragen ins Leere laufen zu lassen und somit effektiv Werbung zu blockieren. Startet ein Gerät eine Anfrage für eine unerwünschte Webseite, kann der DNS-Server schlicht mit „diese Webseite existiert gar nicht“

- » Werblocker für Browser und smarte Geräte
- » Installation von Pi-hole auf dem Raspberry Pi
- » Wahl der richtigen DNS-Server

Zeitaufwand:
etwa 2 Stunden



Kosten:
60 Euro



Programmieren: Raspbian aufsetzen

- » **Raspberry Pi** Modell 1B bis 4B oder neuer
- » **SD-Karte** mind. 4GB
- » **Netzteil**

Alles zum Artikel
im Web unter
make-magazin.de/xd1a

antworten. Auf diese Weise kann das Gerät keine Werbung anzeigen und das Werbenetzwerk wird gar nicht erst aufgerufen.

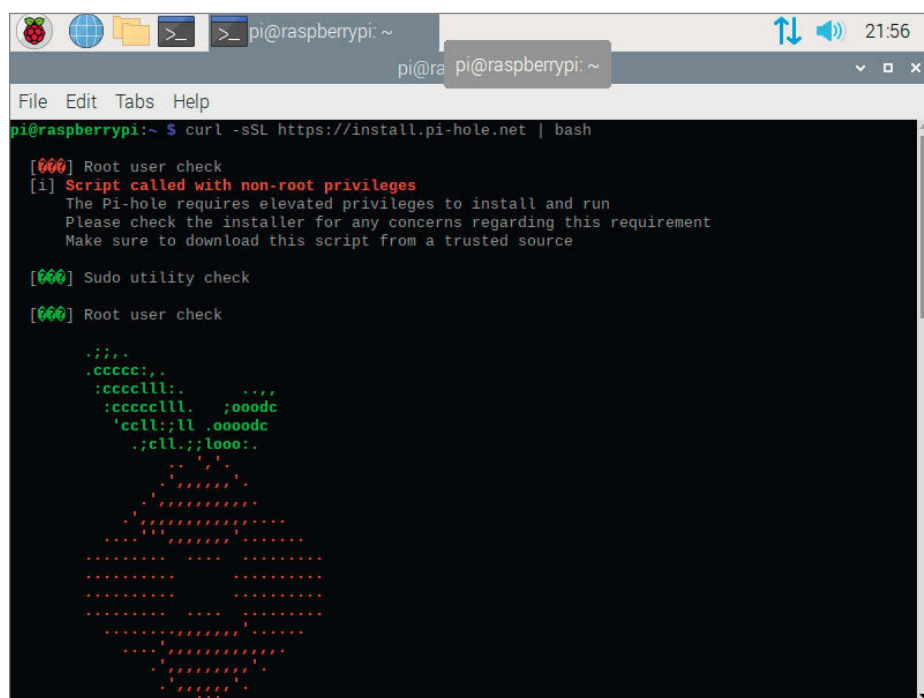
Dazu muss ich allerdings wissen, hinter welchen Adressen Werbenetzwerke stecken. Glücklicherweise gibt es Listen mit solchen Adressen, die von Freiwilligen stets aktuell gehalten werden. Genau diese Listen verwenden auch Ad-Blocker-Browser-Plug-ins, um ihre Aufgabe zu absolvieren.

Strom verbraucht, eignet er sich hervorragend als Server, der rund um die Uhr im Betrieb ist. Was läge also näher, als auf dieser günstigen Plattform einen eigenen DNS-Server zu betreiben? Einen, der nicht nur Werbung, sondern auch anderen unerwünschten Netzwerkverkehr auf DNS-Ebene weafiltriert?

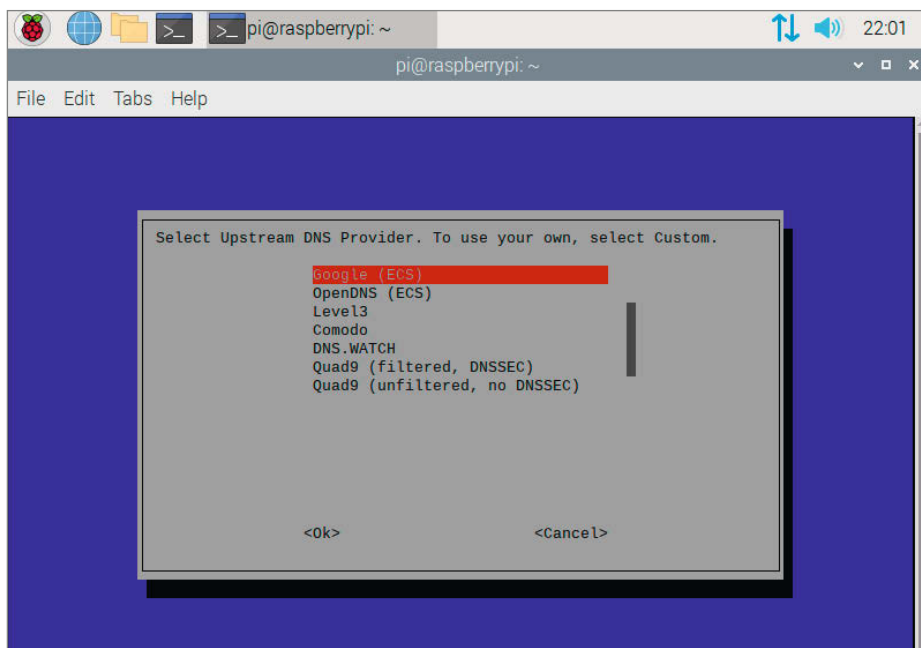
Genau das dachte sich auch das Pi-hole Team und hat gleich eine ganze Suite an passenden Tools entwickelt.

Die maßgeblichen Komponenten sind dabei FTLDNS (Faster Than Light DNS, der DNS-Server von Pi-hole) und eine Weboberfläche mit integriertem Webserver. Über die Weboberfläche kann man beobachten, was

Der Raspberry Pi bietet die perfekte Plattform für meinen Werbefblocker. Da er sehr wenig



1 Ein schneller Root-Check, dann werde ich schon mit einem schicken ASCII-Logo begrüßt.



2 Der Upstream DNS Provider liefert dem Pi-hole die nötigen Daten, um seine Arbeit zu machen.

alles im Netzwerk passiert und das FTLDNS administrieren.

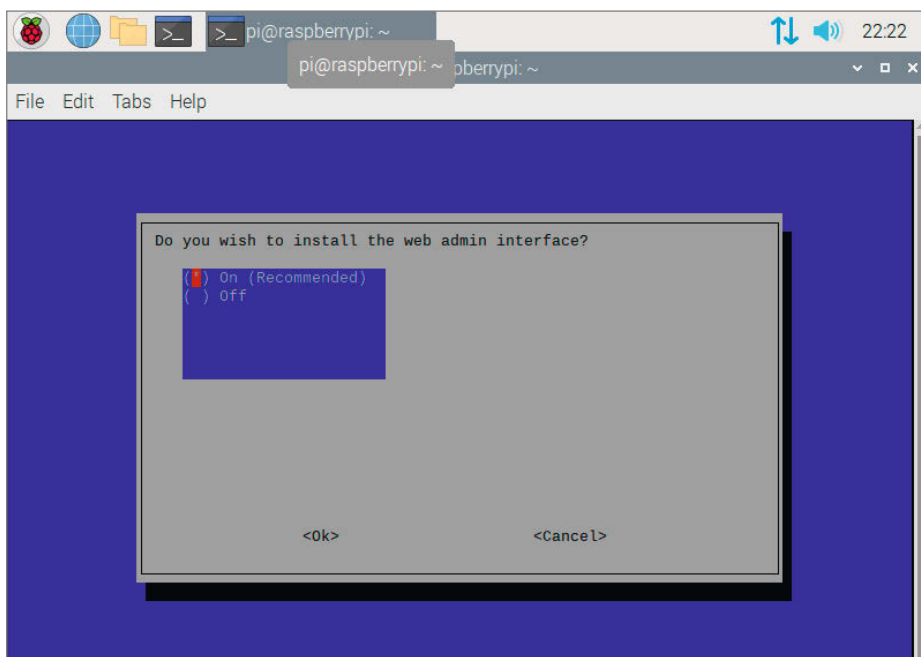
Vor der Installation

Am besten verwendet man einen unbenutzten Raspberry Pi, um Pi-hole zu installieren. Es wird ausdrücklich davor gewarnt, Pi-hole auf einem bestehenden System zu installieren, das bereits andere Netzwerkaktivitäten ausübt. Der Grund dafür ist, dass der Linux-eigene DNS-Dienst dnsmasq deaktiviert wird,

womit bereits existierende Konfigurationen verloren gehen.

Außerdem sollte der Pi im späteren Betrieb für eine möglichst sichere Verbindung per LAN-Kabel ans Netzwerk angeschlossen sein. Nach gut einem Jahr Betrieb im WLAN-Netz kann ich allerdings berichten, dass die Verbindung auch so stabil genug war.

Neben Raspbian, dem empfohlenen Betriebssystem für Raspberries, werden auch Fedora, Debian, Ubuntu und CentOS unterstützt, jeweils mit unterschiedlichen Prozes-



3 Das Web Admin Interface macht die Administration sehr einfach.

sorarchitekturen. Das Betriebssystem muss man vor Pi-hole installieren: In meinem Fall habe ich Raspbian verwendet. Der Bedarf an freiem Festplattenplatz für die Pi-hole-Installation ist mit ungefähr 50 Megabyte vernachlässigbar. Die empfohlene Größe des Arbeitsspeichers ist 512 Megabyte, der wird allerdings im Betrieb kaum belegt.

Installation

Über ein simples Kommandozeilen-Skript, `curl -sSL https://install.pi-hole.net | bash`, wird der textbasierte Installer gestartet 1. Das führe ich im Terminal des RasPis aus. Dann werde ich auf Englisch durch die Installation geführt. Nebenher wird mir der Prozess erklärt: Welche Schritte muss man außerhalb des Pis – vor allem am heimischen Router – vornehmen, um die gewünschte Blockwirkung zu erzielen?

Jetzt kommt die erste nennenswerte Entscheidung: die Wahl des Upstream DNS Providers 2. Ohne den geht es nicht, denn Pi-hole besitzt nicht selber alle „Telefonbucheinträge für das Internet“. Es muss sich die IP-Adressen für Internetdienste von außerhalb holen. Ich habe hier die Möglichkeit, zwischen verschiedenen Diensten auszuwählen. Dazu gehören unter anderem Google, OpenDNS und DNS.WATCH.

Google würde sich nur zu sehr freuen, meine Surfgewohnheiten weiter zu analysieren (mehr dazu im Kapitel Datensicherheit). Somit scheidet die Firma als Anbieter aus. OpenDNS ist ein weiteres Unternehmen aus den USA, welches verspricht, Phishing- und andere bössartige Webseiten zu blocken. DNS.WATCH sagt zu, keine Zensur oder anderweitiges Blocken vorzunehmen und DNS-Anfragen von unserem Pi-hole nicht zu speichern. Dazu gibt es noch weitere vordefinierte Optionen. Zu guter Letzt kann ich eigene IP-Adressen für DNS-Server eintragen.

Ich möchte mit meinem Pi-hole selber Blocklisten verwalten und entscheiden, welche Seiten ich zu sehen bekomme. Daher wähle ich erstmal DNS.WATCH als offenen Upstream Provider. In der Web-Oberfläche kann ich später immer noch sehr bequem die Einstellungen verändern.

Blocklisten

Im nächsten Schritt gibt es die Wahl der Blocklisten. Zur Auswahl stehend und bereits angehakt sind *StevenBlacks's Unified Hosts List* und eine Liste mit Malware-Domains. StevenBlacks Liste ist eine Art Best-Of der Werbenetzwerke und wird von der Community aktuell gehalten. Ich lasse beide angehakt und gehe zum nächsten Schritt. Hier kann ich auswählen, ob IPv4 und IPv6 behandelt werden sollen. Natürlich sollen beide Versionen greifen: IPv4,

weil es noch immer sehr stark verbreitet ist, und IPv6, weil es die Zukunft ist.

Beim nächsten Schritt wird mir angezeigt, welche IP-Adresse mein Pi vom Router bekommen hat. Genau diese wird der Pi in Zukunft auch bei sich als statische Adresse hinterlegen. Später werde ich sie als DNS-Server im Router eintragen müssen. Es folgt eine Belehrung über IP-Konflikte und wie ich sie vermeiden kann. Auch diese Information wird später bei der Router-Konfiguration noch nützlich sein.

Nachdem mir mitgeteilt wird, welche Adresse hinter meiner Upstream-Auswahl steckt, werde ich gefragt, ob ich das Web Admin Interface installieren möchte **3**. Da es ein echtes Sahnestück für Statistikbegeisterte und Kommandozeilenmuckel darstellt, bleibt der Haken bei *On* drin. Um die Oberfläche auch ohne weitere Konfiguration sofort nutzen zu können, sollte im nächsten Schritt auch der dazugehörige Webserver mitinstalliert werden.

Privacy Modes

Bei den darauffolgenden Fragen, ob Queries geloggt werden sollen und welchen Privacy Mode man aktiviert haben möchte, kann ich getrost die vorausgewählten Optionen benutzen. Wer allerdings sein Pi-hole mit größerem Fokus auf Datensparsamkeit betreiben möchte, sollte sich die vorgestellten Optionen genauer angucken. Speziell die Privacy Modes sind eine gute Möglichkeit, mit einem Klick einzustellen, welche Informationen zu den Surfgewohnheiten der Netzwerkbenutzer sich das Pi-hole so merkt. Dazu gehören zum Beispiel die Optionen, dass zwar alle DNS-Anfragen gespeichert, aber nur anonymisiert ausgegeben werden, oder dass rein gar keine Daten historisiert werden. Mehr zu diesem Thema gibt es im Abschnitt *Privatsphäre* zu lesen.

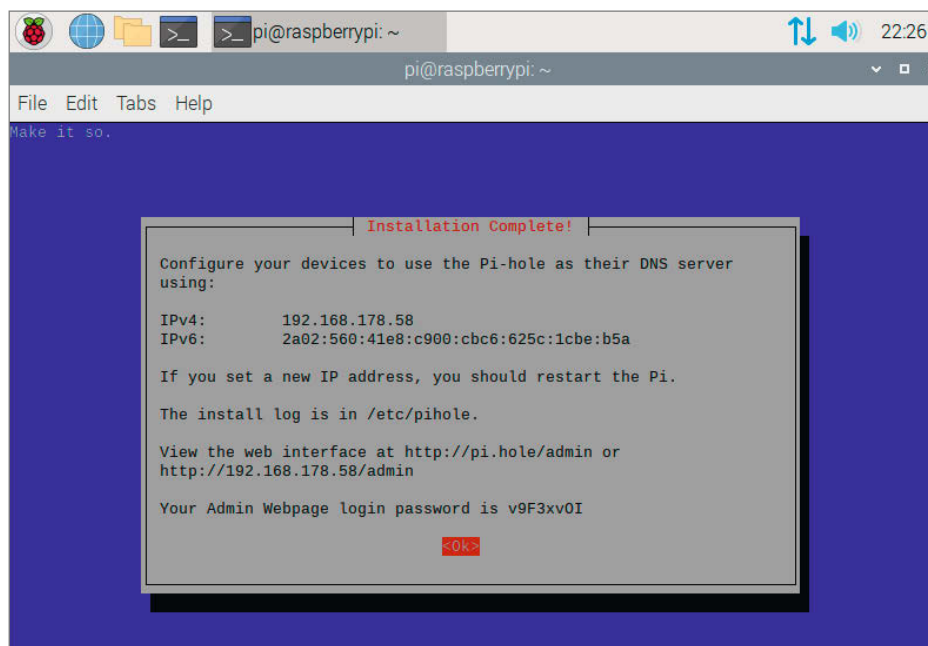
Darauf folgt die Linux-typische vorbeisrollende Textwand, die mir auf verschiedene Art und Weise erklärt, was gerade so vor sich geht. Details interessieren mich da wenig. Viel wichtiger ist, dass ich nach einer kurzen Wartezeit ein letztes Fenster mit dem Titel „Installation Complete!“ angezeigt bekomme.

Hiervon sollte man sich einen Screenshot machen **4**, denn es werden wichtige Informationen angezeigt: Die vom Router zugewiesenen IPv4- und IPv6-Adressen des Rasperrys, wie ich auf die Web-Admin-Oberfläche zugreifen kann und das dazugehörige Passwort.

Herzlichen Glückwunsch an ... mich, es scheint alles gut gegangen zu sein!

Pi mit der Fritzbox verbinden

Das war aber noch nicht alles: Schön ist es schon, dass ich jetzt meinen eigenen DNS-Server mit integriertem Ad-Blocker habe, aber er wird noch nicht genutzt. Dann muss ich den



4 Hier Screenshot machen oder kopieren; diese Infos brauchen wir noch.

Pi noch im Router hinterlegen. Ich erkläre den Prozess für meine Fritz-Box, die viele andere Haushalte auch besitzen. (Andere Router funktionieren natürlich auch.)

Dafür navigiere ich mit meinem Browser zur Adresse <http://fritz.box> und melde mich dort an. Dort teile ich dem Router als erstes mit, dass mein Pi immer dieselbe IP-Adresse erhalten soll. Ist er per WLAN verbunden, geschieht dies über *WLAN/Funknetz*. Dort suche ich das entsprechende Gerät heraus, klicke auf *Bearbeiten* und setze einen Haken bei „Diesem Netzwerkgerät immer die gleiche IPv4-Adresse zuweisen“. Ist der Pi per LAN-Kabel verbunden, mache ich das Gleiche über *Heimnetz/Netzwerk*.

Nun erhält mein Raspberry Pi immer dieselbe IP-Adresse vom Router. Damit kann ich ihn bei der Fritz-Box über *Internet/Zugangsdaten/DNS-Server* als DNS-Server eintragen **5**. Hier gebe ich sowohl die IPv4-Adresse als auch die IPv6-Adresse als bevorzugten Server ein. Als alternativen Server – der Pi kann ja auch mal ausfallen – trage ich den Standardserver meines Internetanbieters ein.

Unter Umständen muss ich dann noch am Gerät physisch bestätigen, dass die Änderung umgesetzt werden soll. Bei meiner Fritz-Box musste ich auch einen beliebigen Knopf des Geräts drücken – eine sinnvolle Vorsichtsmaßnahme.

DNSv4-Server

☐ Vom Internetanbieter zugewiesene DNSv4-Server verwenden (empfohlen)
 ☒ Andere DNSv4-Server verwenden

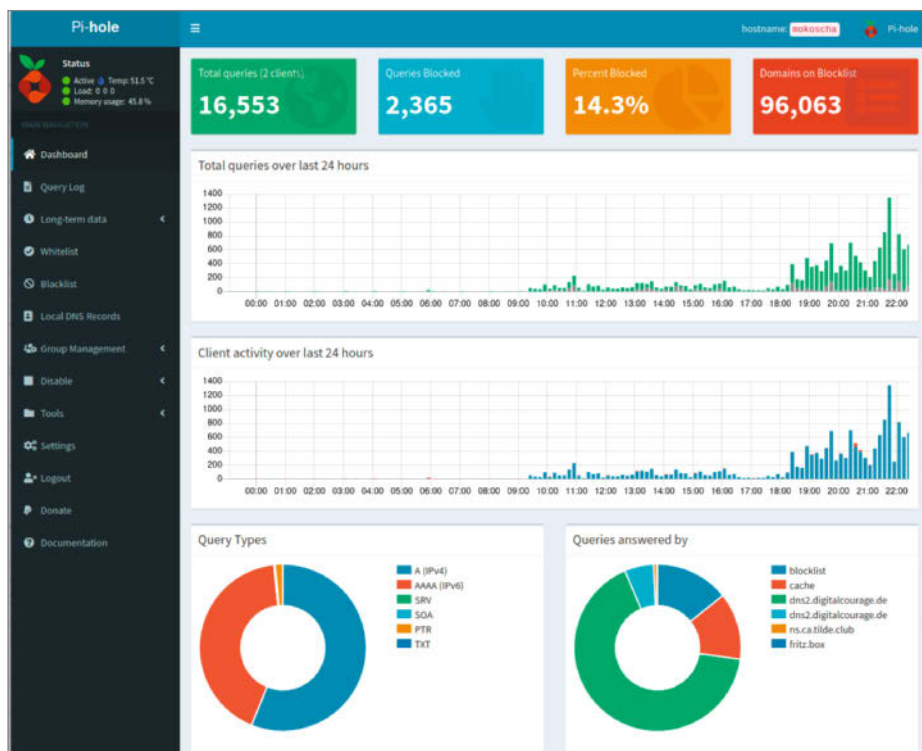
Bevorzugter DNSv4-Server: 192 . 168 . 178 . 2
 Alternativer DNSv4-Server: 84 . 200 . 69 . 80

DNSv6-Server

☐ Vom Internetanbieter zugewiesene DNSv6-Server verwenden (empfohlen)
 ☒ Andere DNSv6-Server verwenden

Bevorzugter DNSv6-Server: 2a02:560:416c:3e00:4e98:e442:b1ea:a3c9
 Alternativer DNSv6-Server: 2001:1608:10:25::1c04:b12f

5 Jetzt geht es ans Eingemachte: Der Pi-hole wird in Betrieb genommen.



6 Sehr übersichtlich, alle wichtigen Informationen auf einen Blick: Das Dashboard meines Pi-holes, das schon seit geraumer Zeit im Betrieb ist.

Aber jetzt will ich mir das schicke Web Interface angucken!

Das Web-Interface

Nun sollte ich das Web Interface über die angegebene Adresse <http://pi.hole/admin> öffnen können. Sollte dies jedoch nicht funktionieren, kann ich es über die IP-Adresse versuchen, die auch in der Abschlussmeldung des Installers angegeben wurde. Ich

werde mit einem fast leeren Dashboard begrüßt und habe relativ wenig Interaktionsmöglichkeiten. Nach einem Klick auf *Login* werde ich zur Passwordeingabe aufgefordert. Das Passwort stammt aus dem Installer, es gibt aber auch eine Zurücksetzen-Funktion.

Nun steht mir die Tür zum Herumklicken, Erforschen und Rumbasteln offen 6. So kann mir Pi-hole zum Beispiel mittels eines schicken Säulendiagramms anzeigen, wie viel DNS-Anfragen in frei wählbaren Zeitintervallen be-

FTL Information	
FTL version:	v5.1
Process identifier (PID):	23952
Time FTL started:	19:39:56
User / Group:	pihole / pihole
Total CPU utilization:	0.1%
Memory utilization:	0.6%
Used memory:	5.51 MB
DNS cache size:	10000
DNS cache insertions:	1477
DNS cache evictions:	0

See also our [DNS cache documentation](#).

7 Schön schlank: FTL

antwortet wurden. Das bringt allerdings bei einer frischen Instanz wenig, da noch kaum Datenverkehr durch meinen Server geflossen ist. Das kann ich natürlich etwas beschleunigen, indem ich möglichst viele unterschiedliche Webseiten öffne. Danach gibt es auch etwas zu gucken.

Interessant sind beispielsweise die Informationen zu FTL, der DNS-Server Implementierung des Pi-hole 7. Man kann gut erkennen, dass er sehr sparsam ist; er verbraucht vernachlässigbare CPU- und Arbeitsspeicherressourcen. Schön, dass die Macher großen Wert auf ein schlankes Programm gelegt haben.

In den Einstellungen finde ich die Maske, in der ich die Upstream-DNS-Server verändern kann 8. Die Liste kenne ich schon aus der Installationsprozedur, aber auf der rechten Seite kann ich meine eigene Adresse eingeben. Ich mache mich auf die Suche nach geeigneten Kandidaten, denen ich vertrauen kann. Schließlich wird mein Pi-hole, wenn es eine angefragte Adresse nicht kennt, eine Anfrage an einen dieser Server starten. Auf diese Weise weiß dann der angefragte DNS-Server, welche Webseiten ich so besuche. Daher fallen eigentlich schon alle vorgeschlagenen Anbieter raus: Hier handelt es sich entweder um US-amerikanische Unternehmen oder Server, die selber schon filtern. Ich möchte aber kein vorzensiertes Internet erleben, sondern, wenn überhaupt, selber filtern.

Meine Wahl für den primären Server fällt auf den DNS-Server des Digitalcourage e.V.: Kein Filtern, gemeinnütziger Verein, kein Logging, klingt sehr gut. Den sekundären Server habe ich aus einer Liste von öffentlichen Servern. Es handelt sich um einen kleinen, nicht loggenden Server, der von Enthusiasten be-

Upstream DNS Servers

IPv4	IPv6	Name
☐	☐	Google (ECS)
☐	☐	OpenDNS (ECS)
☐	☐	Level3
☐	☐	Comodo
☐	☐	DNS.WATCH
☐	☐	Quad9 (filtered, DNSSEC)
☐	☐	Quad9 (unfiltered, no DNSSEC)
☐	☐	Quad9 (filtered + ECS)
☐	☐	Cloudflare

ECS (Extended Client Subnet) defines a mechanism for recursive resolvers to

Upstream DNS Servers

Custom 1 (IPv4)

☒

Custom 3 (IPv6)

☒

Custom 2 (IPv4)

☒

Custom 4 (IPv6)

☒

Interface listening behavior

☐ Listen on all interfaces
 Allows only queries from devices that are at most one hop away (local devices)

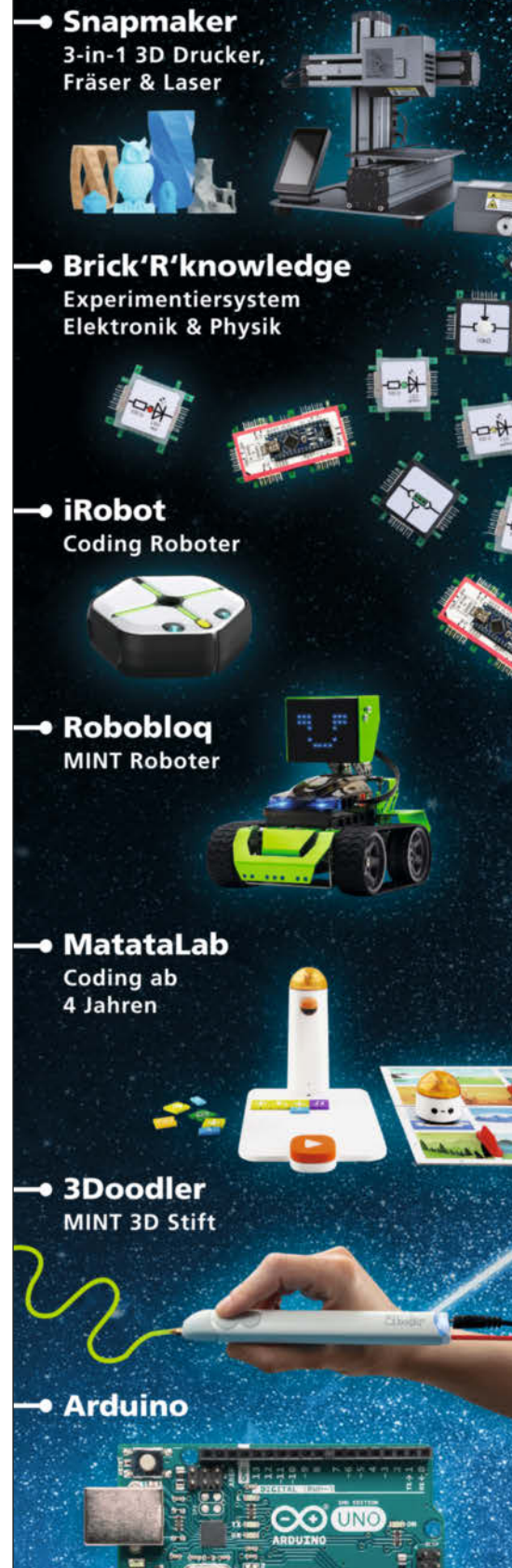
☒ Listen only on interface wlan0

☐ Listen on all interfaces, permit all origins

Note that the last option should not be used on devices which are directly connected to the Internet. This option is safe if your Pi-hole is located within your local network, i.e. protected behind your router, and you have not forwarded port 53 to this device. In virtually all other cases you have to make sure that your Pi-hole is properly firewalled.

8 Voreingestellte DNS-Server? Ich gebe lieber meine eigenen ein.

ALLNET MAKER & MINT WELT



Privacy settings

DNS resolver privacy level

Specify if DNS queries should be anonymized, available options are:

☒ **Show everything and record everything**

Gives maximum amount of statistics

☐ **Hide domains: Display and store all domains as "hidden"**

This disables the Top Domains and Top Ads tables on the dashboard

☐ **Hide domains and clients: Display and store all domains as "hidden" and all clients as "0.0.0.0"**

This disables all tables on the dashboard

☐ **Anonymous mode: This disables basically everything except the live anonymous statistics**

No history is saved at all to the database, and nothing is shown in the query log. Also, there are no top item l

The privacy level may be increased at any time without having to restart the DNS resolver. However, note that the privacy level. This restarting is automatically done when saving.

9 Wie viel Daten soll mein Pi-hole aufbewahren?

trieben wird. Um solche Server zu finden, sollte man sich an die Ratschläge von bekannten Vereinigungen wie dem Chaos Computer Club oder dem Digitalcourage e.V. halten.

Privatsphäre

Wie oben schon angeschnitten, weiß der angefragte DNS-Server, welche Webadressen ein Client ansteuert. Daher ist es wichtig, dass man „seinem“ DNS-Server vertrauen kann. Dies gilt jedoch nicht nur für den Upstream-Server: Ich habe mir ja meinen eigenen DNS-Server ins Netzwerk gestellt, für den gilt das auch. Daher ist es wichtig, dass ich mit den Daten, die der Pi-hole sammelt, verantwortungsvoll umgehe.

Ich konnte bei meinem Pi-hole bisher keine Beziehung von DNS-Anfragen zu Endgeräten herstellen, da der Client immer die Fritz-Box ist. Wer aber Surfgeohnheiten der Nutzer kennt, kann ziemlich genau herauslesen, wer wann welche Webseite aufgerufen hat. Und Nutzer können in diesem Fall zum Beispiel Mitbewohner, Partner und vollpubertäre Jugendliche mit großem Bedarf nach Privatsphäre sein.

Ich kann aber auch aus einem anderen Blickwinkel das Surfverhalten anderer Menschen analysieren: Als meine Wohnung während eines Urlaubs für eine Woche leerstand, haben für eine Nacht zwei Bekannte bei mir übernachtet. Die beiden hatten die Zugangsdaten zum WLAN. In der Statistik tauchte dann deutlich eine Nutzungsspitze auf. Über diese hätte ich herausfinden können, welche Webseiten aufgerufen wurden und dieses Surfverhalten so den Gästen zuordnen können.

Wer dieser Versuchung nicht widerstehen kann, kann in den *Privacy settings* 9 die zweite oder dritte Option anwählen. Dadurch kann man noch immer interessante Diagramme erzeugen, läuft aber nicht Gefahr, aus Versehen anderen hinterherzuspüffeln.

Schlussbemerkung

Pi-hole ersetzt prinzipbedingt keinen Ad-Blocker im Browser. Es kann nicht das Verhalten des Web-Browsers beeinflussen oder gezielt Skripte deaktivieren. Dafür werden unter Umständen aber Funktionen von erwünschten Onlinediensten gestört: Ich kann beispielsweise keine Suchergebnisse von Google Shopping mehr aufrufen. Diese werden aber eh von Ad-Blockern geblockt. Die von Google verlinkten Webseiten kann ich immerhin händisch aufrufen, ohne dass Google meine Klicks nachvollziehen kann.

Dies ist ein kleiner Preis – und ich bin gewillt, ihn zu bezahlen, denn die Vorteile sind groß: Pi-hole kennt eine lange Liste von bekannten Webseiten, die Malware verbreiten und blockt entsprechende DNS-Anfragen. So bin ich vor leichtfertigen Klicks auf dubiose Webseiten geschützt. Zusätzlich schützt es mich vor eigentlich rechtschaffenen Webseiten, die böswillig gehackt wurden und mich nun auf eine andere Webseite umleiten wollen. Gegen diese Angriffe und die am weitesten verbreiteten Werbenetzwerke bietet Pi-hole eine solide Abwehr. Und wenn ich gezielt Webseiten, wie beispielsweise heise.de, unterstützen will, indem ich die verknüpften Werbeanbieter freischalte, geht das natürlich auch. —rehu

Dias scannen mit dem Raspberry Pi

Scanner für Dias gibt es zwar schon lange zu kaufen, aber die Digitalisierung eines großen Lichtbildarchivs geht damit oft nur mühsam voran. Schneller geht es mit dieser per Raspberry Pi automatisierten Kombination aus modifiziertem Diaprojektor und Digitalkamera.

von Dominik Laa



Neue Dia-Aufnahmen haben die meisten vermutlich schon länger nicht mehr gemacht, doch bei vielen schlummern noch hunderte oder sogar tausende Aufnahmen in Lagerräumen – oftmals mit wertvollen Erinnerungen. Das Problem: Die Digitalisierung von Diaaufnahmen bei Firmen ist nicht wirklich günstig, insbesondere bei hohen Stückzahlen. Diascanner, die man kaufen kann, funktionieren wie herkömmliche Scanner, mehr als einige Dias kann man da nicht auf einmal bearbeiten, dadurch wird der Zeitaufwand extrem. Somit bleibt nur mehr eine DIY-Lösung: ein Diaprojektor mit vorgeschalteter Kamera und zentraler Steuerung durch ein Raspberry Pi Zero. Das Ganze wird eher funktional und mit minimalem Aufwand aufgebaut, denn in der Regel digitalisiert man nur einmal sein Dia-Archiv und kann anschließend den Aufbau wieder zerlegen.

Das Prinzip

Wer Diaaufnahmen in großen Mengen zu Hause hat, besitzt in der Regel auch noch den dazugehörigen Projektor. Dieser funktioniert eigentlich nach einem einfachen Prinzip: Die Diaaufnahme wird von einer starken Lichtquelle angeleuchtet und das entstandene Bild wird auf eine Leinwand projiziert. Statt einer Leinwand wird für den Diascanner im Eigenbau eine Kamera mit Makroobjektiv benötigt oder zumindest ein Objektiv mit großer Brennweite und Zwischenringen (um die Makrofähigkeit zu gewährleisten).

Das Bild, das eigentlich projiziert werden würde, wird durch die Projektorlampe von hinten beleuchtet und kann einfach mit der Kamera als digitales Foto aufgenommen und bei Bedarf nachbearbeitet werden. Damit das gesamte Bild direkt möglichst formatfüllend aufgenommen werden kann, muss das Objektiv der Kamera natürlich stark genug vergrößern und nahe genug am Projektor befestigt werden, so wie das auch bei Makroaufnahmen der Fall ist.

Der große Vorteil an dieser Methode ist, dass der Vorschub des Diaprojektors für den Bilderwechsel genutzt werden kann und damit eine Automatisierung überhaupt erst möglich wird. Außerdem können die Dias direkt aus den Magazinen heraus verarbeitet werden, in denen sie in der Regel ohnehin gelagert werden – das ist gerade bei den günstigen Diascannern, die es zu kaufen gibt, oft anders.

Licht dimmen

Damit eine Automatisierung funktionieren kann, muss jedoch der Diaprojektor modifiziert werden. Bei herkömmlichen Projektoren ist eine Halogenlampe mit einer Leistung von etwa 150W oder sogar noch mehr verbaut.

Kurzinfo

- » Dias direkt aus dem Magazin heraus digitalisieren
- » Raspberry Pi steuert automatischen Bilderwechsel
- » Stapel-Nachbearbeitung mit Adobe Lightroom oder ähnlicher Software

Checkliste



Zeitaufwand:

Aufbau und Feintuning etwa ein Tag, Scan-Dauer je nach Größe der eigenen Dia-Sammlung



Kosten:

von 0 bis rund 100 Euro, je nach vorhandener Hardware und Inhalt der Bastelkiste (ohne Projektor und Kamera)



Fotografische Ausrüstung:

digitale Spiegelreflexkamera mit Makro-Objektiv oder Objektiv längerer Brennweite (für die Vergrößerung) und Zwischenringen (um die Makrofähigkeit zu gewährleisten)



Computer:

mit Bildbearbeitungssoftware für RAW-Entwicklung und Stapelverarbeitung, etwa *Adobe Lightroom* oder *Darktable*

Material

- » Diaprojektor mit Fernbedienung (kabelgebunden oder Funk)
- » Projektor-Ersatzlampe Halogen, 50–75 Watt / 24 Volt
- » Stativkopf mit Montageplatte
- » Diffusormaterial etwa Diffusorfolie aus dem Fotobedarf, PTFE-Platte (Teflon) mit 0,5–1mm Stärke oder dünnes Milchglas
- » Raspberry Pi mit Netzteil und SD-Karte mit Standard-Raspi-Betriebssystem, ggf. Adapterkabel für USB-Anschluss beim Pi Zero
- » Servo
- » Material für den Unterbau Aluprofile, Bretter, Winkel oder ähnliches
- » Verbindungsmaterial Schrauben, Kabelbinder und ähnliches

Mehr zum Thema

- » Peter König, Alternative Optik für die ESP32-CAM, Make 2/20, S. 8
- » Peter König, Der einfachste Plotter der Welt, Make 1/20, S. 8

Alles zum Artikel
im Web unter
make-magazin.de/xh3n



1 Ein Makro-Adapter kann ein herkömmliches Teleobjektiv zu einem eingeschränkten Makro-Objektiv machen – für den DIY Diascanner reicht das auf jeden Fall aus.



Trifft diese enorme Leistung auf den empfindlichen Bildsensor der Kamera (oder gar auf das menschliche Auge, das durch den optischen Sucher der Spiegelreflexkamera blickt) drohen Schäden. Da außerdem die Lebensdauer von Halogenlampen mit derart hoher Leistung beschränkt ist, sollte lieber eine Lampe mit niedrigerer Leistung verbaut werden (50W bis 75W ²). Die Farbtemperatur sollte dabei ähnlich wie die originale Lampe sein (etwa 2900K), geringfügige Abweichungen können aber später bei der Bildbearbeitung entfernt werden. Die Besorgung der Lampe stellt vermutlich auch den schwierigsten Teil des gesamten Projektes dar, derartige Halogenlampen sind mittlerweile selten geworden.

Theoretisch kann man die Lichtquelle auch gegen eine ganz andere austauschen und etwa LEDs einbauen. Dabei ist aber die Farbtemperatur zu beachten, außerdem stehen zur Spannungsversorgung durch den Projektor von Haus aus nur 24V zur Verfügung, was die Auswahl noch stärker einschränkt, weshalb

ich der Einfachheit halber lieber nur die Lampe getauscht habe.

Diffusor benötigt

Um eine absolut gleichmäßige Beleuchtung sicherzustellen und zudem noch etwas mehr von der Helligkeit abzuschwächen, setzt man zusätzlich noch ein Diffusor ein. Entweder benutzt man dafür mehrere Lagen von herkömmlicher Diffusorfolie (aus dem Fotobedarf), eine dünne PTFE-Platte (alias Teflon, 0,5mm oder 1mm stark) oder ein Milchglas mit passenden Abmessungen. Nachdem PTFE für den Einsatz als Dichtung günstig zu haben ist und einfach zugeschnitten werden kann, habe ich mich dafür entschieden.

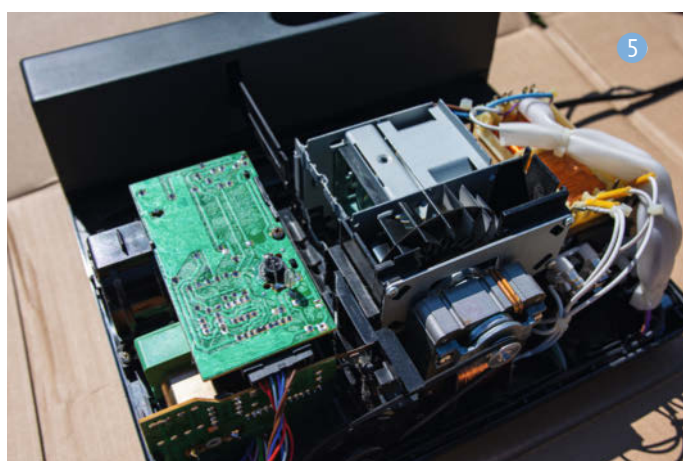
Projektorumbau

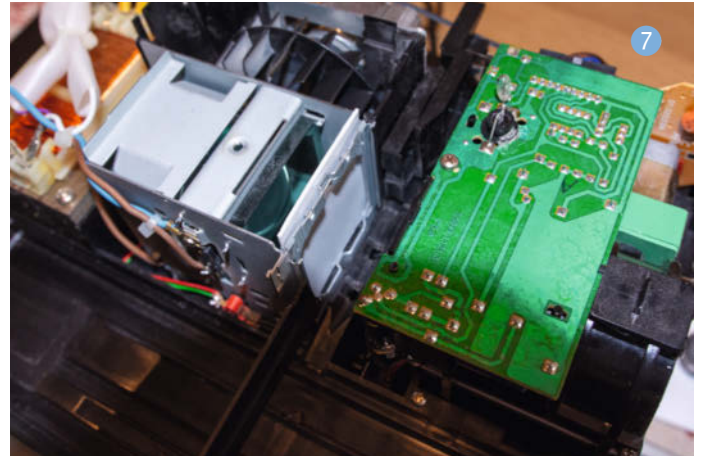
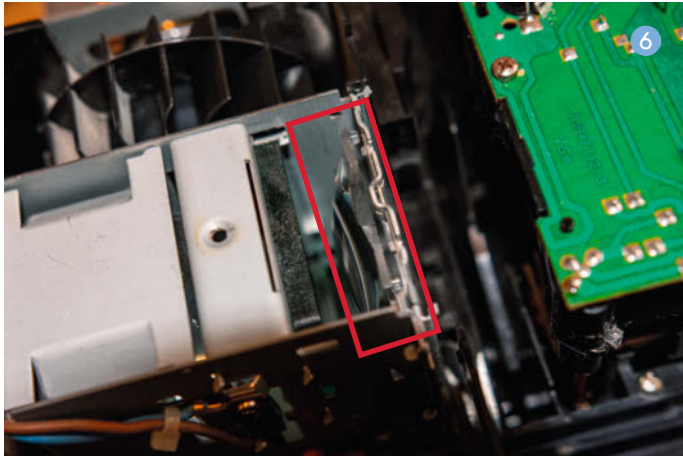
Um den Projektor für den Scanner-Einsatz umzubauen, muss er unbedingt von der Netzspannung getrennt sein. **Vorsicht:** Auch danach können spannungsführende Bauteile noch gefährliche Restspannung gespeichert haben!

Im ersten Schritt muss das Objektiv am Projektor entfernt werden ³, es wird einfach linksherum herausgeschraubt. Dieses Objektiv wird fürs Scannen nicht benötigt, weil später das Objektiv an der Kamera zum Aufnehmen der Fotos ohnehin möglichst knapp vor dem Projektor montiert wird (siehe Aufmacherbild zum Artikel).

Dann öffnet man den Deckel (im Bild ⁴ blau markiert); je nach Modell müssen dafür auch Schrauben gelöst werden (rot markiert). Außerdem entnimmt man den Dia-Halter (grün markiert) – er dient im normalen Betrieb dazu, das aktuell eingelegte Dia aus dem Projektor zu ziehen, etwa um es umzudrehen, wenn es seitenverkehrt einsortiert war. Schließlich nimmt man das Gehäuse ab ⁵.

Anschließend kann man im Inneren des Projektors vorsichtig das in Projektionsrichtung von der Lampe aus gesehen weiter vorne (also unmittelbar am Dia) liegende flache Glaselement entfernen, den Kondensor – im Bild ⁶ rot markiert. Das senkt zwar die Lichtausbeute noch etwas weiter, was aber aus den





oben genannten Gründen gerade erwünscht ist. Statt dem Kondensor passt man ein Stück Diffusor an dieser Stelle ein ⁷. Gerade bei dünnem PTFE oder Diffusorfolie sollten dabei 2 oder sogar 3 Lagen verwendet werden. Achtung: Der Diffusor soll nicht nur das Licht gleichmäßig verteilen, sondern auch die Wärme der Projektorlampe vom Dia fernhalten. Deshalb ist Papier und ähnliches Material als Diffusor nicht geeignet, hier droht Feuergefahr!

Zuletzt wird die originale Lampe ausgebaut und durch die neue, schwächere Lampe ersetzt. Die originale Lampe sollte man aber auf jeden Fall behalten, um den Diaprojektor später wieder zurückbauen zu können. Ist alles soweit vorbereitet, schließt man das Gehäuse des Projektors wieder.

Aufbau

Grundsätzlich könnte der Projektor jetzt schon aufgestellt und die Kamera mit einem Stativ davor platziert werden. Das Problem dabei ist nur: Der Bereich, in dem das Objektiv der Kamera wirklich perfekt scharfe Fotos aufnimmt, ist bei solchen Makroaufnahmen meist kleiner als 1mm. Das heißt, wenn man aus Versehen den Diaprojektor verschiebt oder die Kamera bewegt, sind alle nachfolgenden Aufnahmen unbrauchbar.

Damit das nicht geschehen kann, baut man eine einfache Vorrichtung – je nach eigenen Möglichkeiten kann die aus dem 3D-Drucker kommen oder aus Aluprofilen, Holzbrettern oder sogar (CNC-)gefrästen Bauteilen bestehen. In jedem Fall wird ein Stativkopf benötigt, auf dem man Objektiv und Kamera befestigt, die dann im richtigen Winkel zum Projektor ausgerichtet werden.

Ich habe meine Konstruktion mit Aluminiumprofilen umgesetzt. Da die Platte, auf der der Projektor sitzt ⁸, nur an zwei Punkten befestigt ist, kann sie einfach geneigt werden, dadurch wird die Einstellarbeit vereinfacht. Kernstück meines Aufbaues ist die Adapterplatte ⁹, die ich aus Aluminium gefräst habe. Natürlich kann man sie auch einfach aus

Kunststoff oder hartem Holz zuschneiden und mit den passenden Bohrungen versehen. Der Stativkopf wird von unten auf die Adapterplatte geschraubt ¹⁰, die wiederum den Stativkopf auf dem Aluprofil befestigt. Auf diesem kann der Stativkopf samt Adapterplatte einfach nach vorne und hinten geschoben und dann in perfekter Entfernung zwischen Projektor und Kamera festgeschraubt werden.

Fernbedienung

Was noch gebraucht wird, ist eine Halterung für das Bedienteil oder die Fernbedienung des Diaprojektors. Um dieses nicht weiter modifizieren zu müssen, löst ein Servo den Tastendruck für den Wechsel zum nächsten Dia aus. Eine Vorrichtung dafür kann schnell und einfach aus Restmaterialien gebastelt werden: einfach ein Reststück einer Platte nehmen und einige Löcher bohren, durch die später Kabelbinder gefädelt werden, die das Bedienteil halten ¹¹. Das Servo findet anschließend auf einem kleinen Sockel Platz. Der Anschluss des Servos an den Raspberry Pi Zero ist ebenfalls einfach, wie das Diagramm zeigt ¹².

Software

Die Hardware ist nun abgeschlossen, also geht es weiter mit der Software. Dafür werden auf

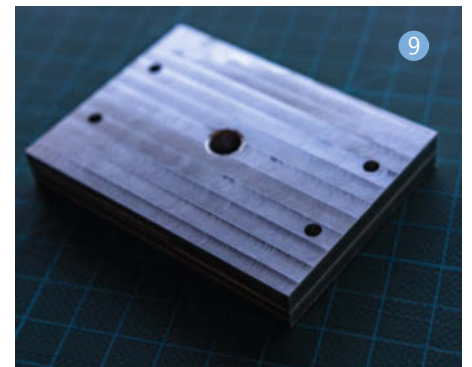
dem Pi zwei Programme benötigt: *gphoto2* und *pigpio*. Mit Hilfe von *gphoto2* können unterstützte Kameras angesteuert werden – das Programm kann nicht nur die Aufnahme von Fotos triggern, sondern auch zahlreiche weitere Parameter setzen. Die Liste der unterstützten Kameras ist erstaunlich lang (siehe Link in der Kurzinfo) und keineswegs auf neueste Modelle beschränkt. Wer möchte, kann die Fotos von der Kamera auch sofort auf den Raspberry Pi laden – nachdem das aber gerade bei dem Pi Zero etwas dauert, würde ich empfehlen, stattdessen einfach die SD-Karte in der Kamera zu benutzen.

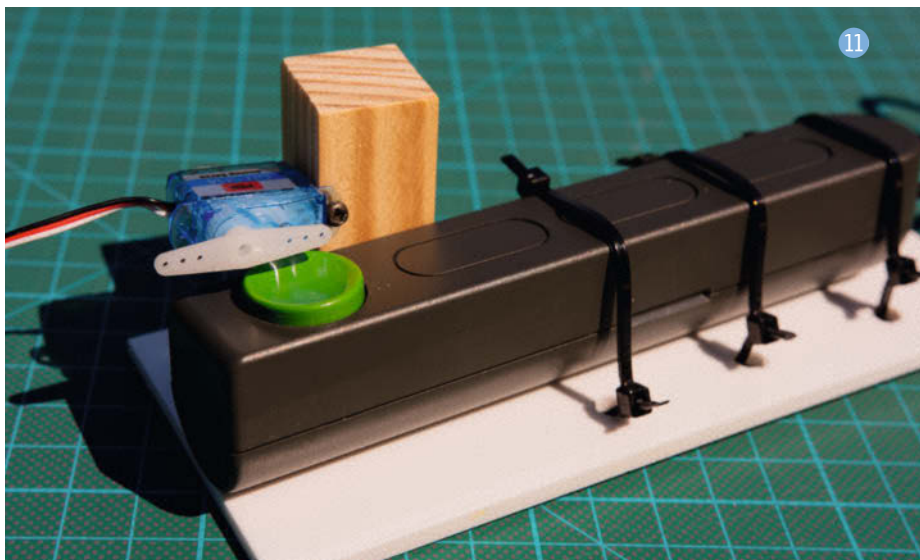
Die zweite Software ist der *pigpio Daemon*. Mit diesem ist es möglich, die elektrischen Ausgänge des Raspberry Pi anzusteuern, sogar die Steuerung von Servos per PWM-Signal funktioniert damit ohne Probleme (siehe auch Make 1/20, S.8).

Für die Installation der beiden Tools gibt man einfach folgende Befehle auf der Kommandozeile ein:

```
sudo apt-get install pigpio
sudo apt-get install gphoto2
```

Damit ist die Installation fertig und man kann erste Tests machen. Erst einmal muss jetzt die Kamera per USB-Kabel an den Raspberry Pi angeschlossen werden. Im meinem Fall, beim Raspberry Pi Zero, braucht man dazu einen zusätzlichen Adapter. Mittels *gphoto2* prüft





man jetzt, ob die Kamera korrekt erkannt wurde [13](#).

Hat das wie gezeigt funktioniert, kann direkt die Fernauslösung der Kamera getestet werden:

```
gphoto2 --trigger-capture
```

Automatik

Damit die Dias später automatisch eins nach dem anderen gescannt werden können, muss das Servo regelmäßig aufs Knöpfchen drücken und dazu genau definierte Positionen ansteuern. Dafür kommt der pigpio-Demon

zum Einsatz, gestartet über:

```
sudo pigpiod
```

Als nächstes muss unser angeschlossenes Servo (auf Ausgang 14 des Raspberry Pi) auch als solches konfiguriert werden:

```
pigs pfs 14 50
```

Jetzt heißt es, erst einmal die richtigen Werte für die gewünschten Positionen herauszufinden. Da hilft nur probieren. Mit dem Befehl

```
pigs servo 14 <WERT>
```

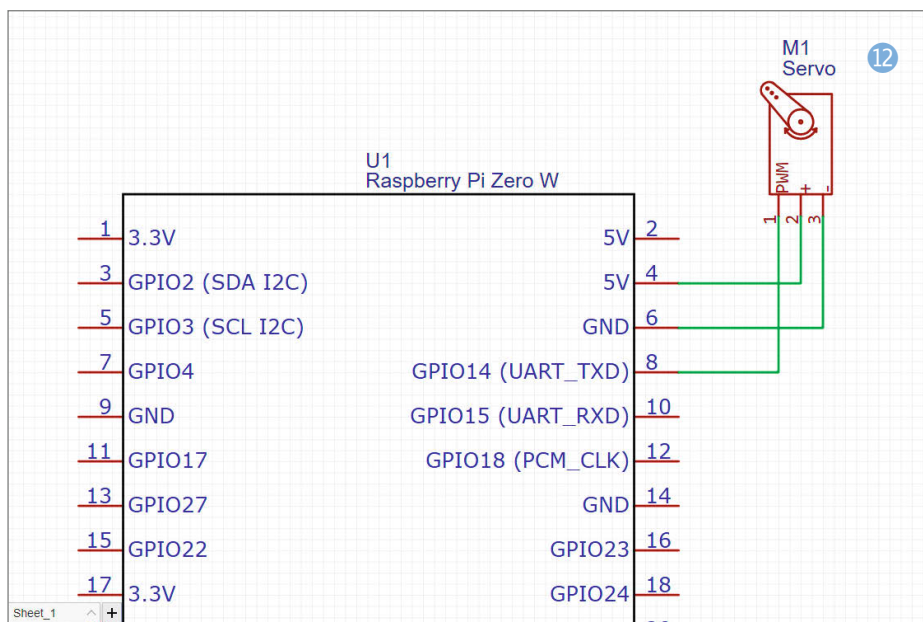
bewegt man das Servo. Zwei Stellungen und die dazugehörigen Werte sind dabei von Interesse: Einen Wert, bei dem der Knopf zum Weiterschalten sicher nicht berührt wird, und einer, bei dem der Knopf sicher (aber ohne Gewalt) ausgelöst wird. In meinem Fall waren das die Werte 1400 (für nicht gedrückt) und 1750 (für gedrückt).

Damit der Vorgang jetzt automatisiert abläuft, reicht ein einfaches Skript [14](#) (Download siehe Link in der Kurzinfo). Um es zu editieren, reicht jeder beliebige Editor, etwa der vorinstallierte *nano*:

```
nano aufnahme.sh
```

Das Skript sorgt dafür, dass ein Magazin mit Dias durchläuft und von jedem Bild automatisch ein Reproduktionsfoto geschossen wird. Das genaue Timing und die nötigen Wartezeiten hängen dabei wesentlich von der verwendeten Kamera und dem Diaprojektor ab – ich konnte eine Geschwindigkeit von etwa 3 Sekunden pro Dia erzielen (ein Video findet man über den Link in der Kurzinfo).

Wie viele Dias in das Magazin passen, steht in der Variable *IMGs*, die die Durchläufe der Schleife begrenzt. Ich würde empfehlen, zusätzlich am Ende ein leeres Foto zu machen, damit später noch klar ist, wann die Kassette



13 Kamera erkennen

```
gphoto2 --auto-detect
```

```
pi@raspberrypi:~ $ gphoto2 --auto-detect
```

Model	Port
Nikon DSC D3200	usb:001,003

gewechselt wurde, sodass die Dias in entsprechende Alben abgelegt werden können.

Das erstellte Skript macht man über den Befehl

```
chmod +x aufnahme.sh
```

ausführbar. Man startet es über

```
sudo ./aufnahme.sh
```

Kamera-Einstellungen

Bevor jetzt aber Hunderte Fotos automatisch aufgenommen werden, gilt es erst, die richtigen Einstellungen zu finden. Dafür schaltet man den Projektor (und falls erforderlich, zusätzlich dessen Lichtquelle) ein und legt ein Dia ein – am besten eines, das möglichst scharf ist und starke Kontraste bietet. Die Kamera sollte für alle Dia-Reproduktionsfotos im manuellen Modus sein, etwa in Bezug auf den Autofokus – einzig die Belichtungsdauer kann man bei gewählter Blende und ISO-Empfindlichkeit der Kamera überlassen.

Dann muss der Fokus richtig eingestellt werden, dafür das Objektiv etwa auf Blende 4 (F4) stellen, bei Zoom-Objektiven die Brennweite etwa in die Mitte stellen und dann versuchen, das Dia scharf zu bekommen. Dabei sollte das gesamte Dia in den Sucher der Kamera passen und auch ein Dia im Hochformat sollte aufgenommen werden können. Je nach Objektiv können dafür ein oder mehrere Makroadapter oder Zwischenringe notwendig sein.

Parallel zur Kamera stellt man in diesem Schritt auch den Stativkopf richtig ein. Wenn das Dia augenscheinlich scharf erscheint, können alle Schrauben festgezogen werden und ab diesem Zeitpunkt sollte die Kamera nicht mehr verschoben werden.

Jetzt stellt man die Blende in den Bereich F8 bis F12 ein, um durch einen vergrößerten Tiefenschärfebereich die optimale Schärfe zu gewährleisten (die Blende weiter zu schließen ist aber nicht notwendig). Als nächstes gilt es die Belichtungsdauer einzustellen, das funktioniert wie bei regulären Fotos auch und auch auf die Automatik der Kamera kann man dabei setzen.

Sobald ein Testfoto perfekt scharf abgebildet wird, sollten die Einstellungen beibehalten werden und der automatische Serien-Scan von kompletten Diamagazinen kann losgehen.

Vorher entstauben

Die Qualität der so aufgenommenen Dia-Scans hängt natürlich stark von der Qualität des Ausgangsmaterials ab. Je größer das Korn des originalen Films war, desto schlechter ist die tatsächliche Auflösung des Dias – vergleichbar mit dem Bildrauschen bei hohen ISO-Werten heutiger Digitalkameras.

Ein weiteres Problem stellt Staub dar, er lagert sich gerne auf den Dias ab und fällt dort

14 aufnahme.sh

```
01 #!/bin/bash
02
03 # Deklaration von Variablen
04 n=0
05 IMGS=36
06 BUTTON_NOT_PRESSED=1400
07 BUTTON_PRESSED=1750
08
09 # pigpiod starten und Ausgang konfigurieren
10 pigpiod
11 pigs pfs 14 50
12
13 # Schleife zum Aufnehmen der Fotos
14
15 while [ $n -le $IMGS ]
16 do
17     echo "Getting image $n"
18
19     # Druckknopf auslösen
20     pigs servo 14 $BUTTON_PRESSED
21
22     # 0.5 Sekunden warten
23     sleep 0.5
24
25     # Servo wieder in Neutralstellung bringen
26     pigs servo 14 $BUTTON_NOT_PRESSED
27
28     # Foto mit gphoto2 aufnehmen
29     gphoto2 --trigger-capture
30
31     # Zählervariable erhöhen
32     n=$(( n+1 )) # increments $n
33 done
34
35 # pigpiod wieder beenden
36 pkill pigiod
```

im Normalfall auch nicht weiter auf. Wenn dann allerdings ein Foto mit hoher Qualität gemacht wird, sieht man jedes Staubkorn durchaus. Als Abhilfe kann man die Dias vorab mit reiner Luft (ohne Ölzusatz) abblasen – diese Luft gibt es in Dosen zu kaufen, aber auch kleinste Kompressoren, wie sie etwa für Airbrush eingesetzt werden, liefern Luft ohne Ölzusatz. Nicht geeignet sind hingegen normale Kompressoren, in ihrer Luft befindet sich eine kleine Menge Öl und diese sollte keinesfalls auf den wertvollen Diaaufnahmen landen. Außerdem sollte der Vorgang im Freien durchgeführt werden, sonst verteilt sich der Staub nur weiter im Raum und landet am Ende dann doch wieder auf den Diaaufnahmen. Sind nur einzelne Staubkörnerchen auf manchen Dias, können diese im Bedarfsfall auch einfach in der Nachbearbeitung entfernt werden – in Lightroom muss man dafür im Entwicklungsmodus einfach nur die Taste Q drücken und entsprechende Stellen auswählen.

Nachbearbeitung

Die Fotos landen beim automatischen Dia-scannen mit dem beschriebenen Skript einfach auf der SD-Karte der Kamera, doch damit

ist der Prozess noch nicht zu Ende. Die Fotos enthalten in der Regel noch Teile des Rahmens und sollten allgemein auch noch etwas verbessert werden. Es spart eine Menge Arbeit, wenn man nicht jedes Foto einzeln feintunen muss, sondern eine Software benutzt, bei der man einmal gefundene Einstellungen auf einen ganzen Stapel von Bildern übertragen kann. Dafür empfehle ich die Verwendung der Software *Lightroom* von Adobe, mit der kann man das Zuschneiden der Fotos beinahe komplett automatisieren, da ja alle Bilder aus exakt derselben Kameraposition aufgenommen worden sind. Lightroom kann man sieben Tage kostenlos testen, will man die Software länger benutzen, muss man anschließend ein Abo abschließen. Die einzelnen Schritte der Nachbearbeitung mit Lightroom zeigen wir in einer Online-Bilderstrecke (siehe Link in der Kurzinfor), ganz ähnliche Schritte funktionieren aber auch mit kostenlosen Programmen wie *Darktable*. Um mehr Luft bei der Nachbearbeitung ohne Qualitätsverlust zu haben, sollten die Fotos unbedingt als Rohdaten (im RAW-Format) gespeichert werden; die gewählte Software muss mit diesen also umgehen können. —pek

TFT-Displays im Stil von Nixie-Röhren

In den 1950er Jahren, als sich die Leuchtdiode noch in der Entwicklungsphase befand, hielt die Nixie-Röhre Einzug in die digitale Signalverarbeitung. Inspiriert von dieser wohl elegantesten Form zur Darstellung von Zahlen und Symbolen entstand meine Idee für die Display-basierte Dixie-Röhre.

von Matthias Helneder



Die Dixie-Uhr



Zur Nachbildung von Nixies gibt es bereits zahlreiche Ansätze. Meiner „Röhre“ soll aber ein kleines TFT-Display als Basis dienen und nicht, wie oft im Netz zu finden, ein EL-Draht oder mittels LED beleuchtete Acrylplatten mit Gravur. Angelehnt an die Namensgebung des Vorbilds aus dem Jahre 1954 habe ich mein Projekt „Display Indicator eXperimental No. 1“ genannt und daraus das Akronym „Dixie“ abgeleitet.

Ziel war es, fünf „Dixie-Röhren“ anzufertigen und diese mit einem Arduino Nano als Uhr zu betreiben. Die Röhren habe ich in ein Holzgehäuse im Retrostil gebaut, in dem auch die überschaubare Steuerelektronik Platz findet. Zum Einstellen der Uhr habe ich drei Tasten eingebaut. Mit einem SD-Karten-Shield und einer SD-Karte lassen sich die Bilder austauschen und für die Spannungsversorgung der Displays ist noch ein Spannungsregler nötig.

Röhren-Konstruktion

Im Wesentlichen ist die „Dixie-Röhre“ eine mechanische Konstruktion mit einem TFT-Farbdisplay ① als elektronischer Komponente. Kernstück jeder Röhre ist der Holzsockel der Kunststoffglocke, den ich als Bodenplatte nutze ②. Auf ihm habe ich einen Aluminiumrahmen ③ befestigt, an dem das Display mit Abstandhaltern ④ angeschraubt ist. Zusammen mit acht Bananensteckern ⑤, die als Anschlusspins dienen, bildet die Bodenplatte den Stecker zum Anschluss an die Elektronik. Alle mechanischen Komponenten sind im gut sortierten Baumarkt erhältlich. Die Kunststoffglocke und das Display sowie weitere elektromechanische Teile, beispielsweise die Stecker, sind im Internet zu finden.

Im ersten Schritt habe ich alle Einzelteile für den Rahmen aus Aluminiumprofilen gefertigt. Hierbei habe ich eine Bügelsäge, Feile, Bohrmaschine und einen M3-Gewindeschneider verwendet. Wer besseres Equipment zur Metallverarbeitung hat, kann selbstverständlich darauf zurückgreifen. Das Drahtgitter, welches als dekoratives „Aondengitter“ Verwendung findet, lässt sich mit dem Seitenschneider zurechtschneiden und im Schraubstock biegen. Die Konstruktionszeichnungen mit allen Bemaßungen können online heruntergeladen werden (siehe Link in der Kurzinfor).

Anschließend habe ich kurz den Lötcolben bemüht, um acht Kupferlitzen an die Anschlüsse des Displays zu löten (Länge jeweils ca. 10cm). Sinnvoll ist es, dabei verschiedene Farben zu verwenden. So ist später nachvollziehbar, welche Leitung zu welchem Signal gehört. Bei einfarbigen Leitungen muss die Signalzuordnung sonst mit dem Durchgangsprüfer ermittelt werden. Nun geht es an die Montage des Rahmens.

Kurzinfor

- » Dixie-Röhre bauen
- » TFT-Display über SPI ansteuern
- » Zusammenbau im Holzgehäuse

Checkliste



Zeitaufwand:
2 Wochenenden



Kosten:
ca. 190 Euro



Löten:
Grundkenntnisse



Holzbearbeitung:
Sägen und kleben



Metallbearbeitung:
Fortgeschrittene Kenntnisse



Programmieren:
Arduino-Grundkenntnisse

Material

- » Arduino Nano mit Netzteil
- » SD-Karten-Shield
- » SD-Karte
- » 5 × SPI-TFT-Display (1,8 Zoll, 128 × 160 Pixel)
- » 5 × Kunststoffglocke mit Holzsockel
- » Eine vollständige Liste mit Bezugsquellen finden Sie online.

Mehr zum Thema

- » Florian Schäffer, LED-Nixies, Make 4/18, S. 20
- » Florian Schäffer, Anzeigen am Arduino, Make 4/18, S. 26
- » Hans Borngräber, Glimmziffern hinter Glas, Make 3/15, S. 36

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

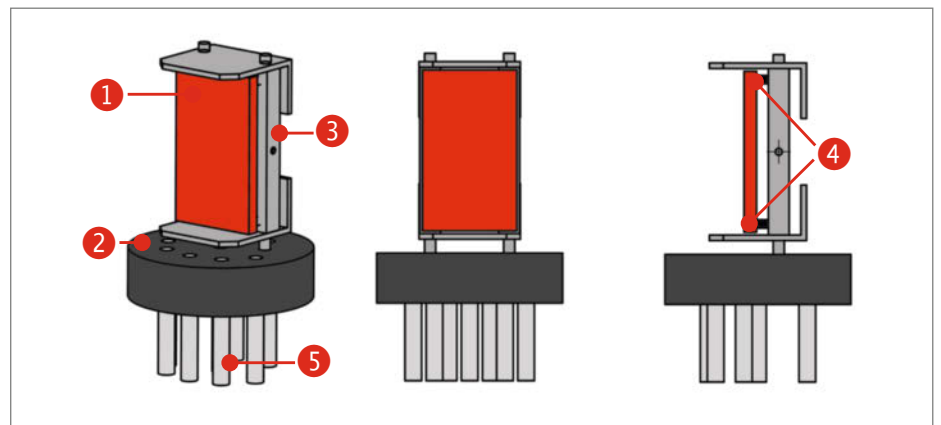
Zusammenbau

Als erstes habe ich das Display mit den Abstandhaltern an den beiden Quadratrohren befestigt. Die Gewinde der Abstandhalter sind gegebenenfalls zu kürzen, da sie nicht weiter als 1mm ins Rohr stehen sollten. Weiter ist zu beachten, dass die M3-Gewinde seitlich am Quadratrohr von einander abgewandt nach links und rechts zeigen müssen.

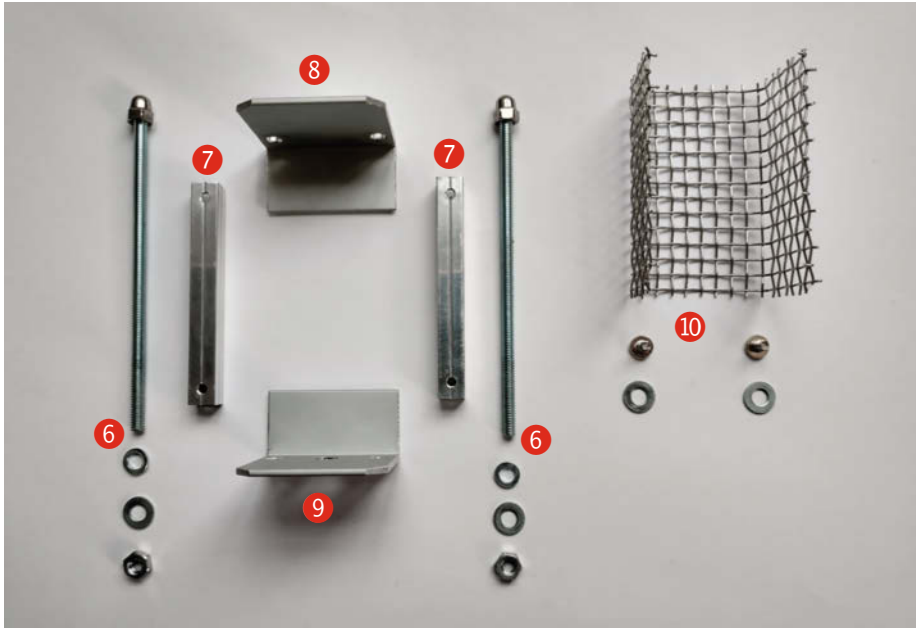
Anschließend habe ich die zwei M4-Gewindestangen durch die Bohrungen des oberen Winkels gesteckt und durch die zwei Rohre geführt. Nun muss man den unteren Winkel

aufstecken und das Ganze fest verschrauben. Die Anschlusslitzen des Displays sind durch die mittige Bohrung des unteren Winkelprofils zu führen. Wegen der schöneren Optik habe ich mich dazu entschlossen, am oberen Winkel Hutmuttern anzubringen. Zum Schluss wird das Drahtgitter über den Rahmen geschoben und an den zuvor genannten M3-Gewinden festgeschraubt.

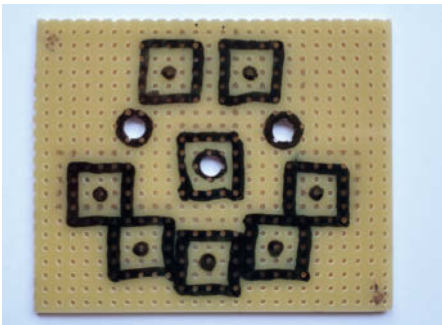
Im zweiten Schritt folgt das Bohren der Bodenplatte. Da diese Bohrungen bei allen fünf Röhren an gleicher Position sein müssen, ist es ratsam, sich hierfür eine Schablone anzufertigen. Sie ist auch später bei der Erstel-



Entwurfszeichnung der Dixie-Röhre



Rahmenteile: 6 Gewindestange, 7 Quadratrohr, 8 Winkelprofil oben, 9 Winkelprofil unten, 10 Drahtgitter



Meine Bohrschablone

lung der Röhren-Sockel sehr hilfreich. Eine Vorlage ist in den Downloads zu finden (siehe Kurzinfo).

Für die Schablone habe ich ein Stück Lochrasterplatte mit einem Raster von 2,54mm verwendet, daher stammen auch die etwas krummen Maße in der Zeichnung. Passenderweise war auf dem Holzsockel bereits eine zentrierte Bohrung vorhanden, an der ich die Schablone leicht ausrichten konnte.

Wer zum Anfertigen der beiden Senkungen an den 4,5-mm-Bohrungen kein passendes Werkzeug wie Flachsener oder einen Forstner-

bohrer zur Hand hat, kann diese weglassen. Allerdings müssen dann die M4-Gewindestangen des Rahmens um ca. 8mm länger sein, da sie andernfalls nicht weit genug aus der Platte ragen, um sie festschrauben zu können. Die acht Bohrungen mit 5mm Durchmesser dienen später zur Befestigung der Bananenstecker. Je nach verwendeten Steckern muss der Bohrungsdurchmesser ggf. anders gewählt werden.

Gestaltung

Sind alle Bohrarbeiten abgeschlossen, geht es an die farbliche Gestaltung der Bodenplatte mit Buntlack oder Ähnlichem. Ein helles Grau erschien mir für eine Röhre weitaus passender als der ursprüngliche braune Farbton. Während der Lack trocknet, kann man die Zeit nutzen, um einen Standard für die Verteilung der Displaysignale auf die Bananenstecker festzulegen. Wer möchte, kann sich an meinem Beispiel orientieren. Wichtig ist, diesen Standard bei allen fünf Röhren und später bei den dazugehörigen Sockeln im Gehäuse gleichermaßen anzuwenden.

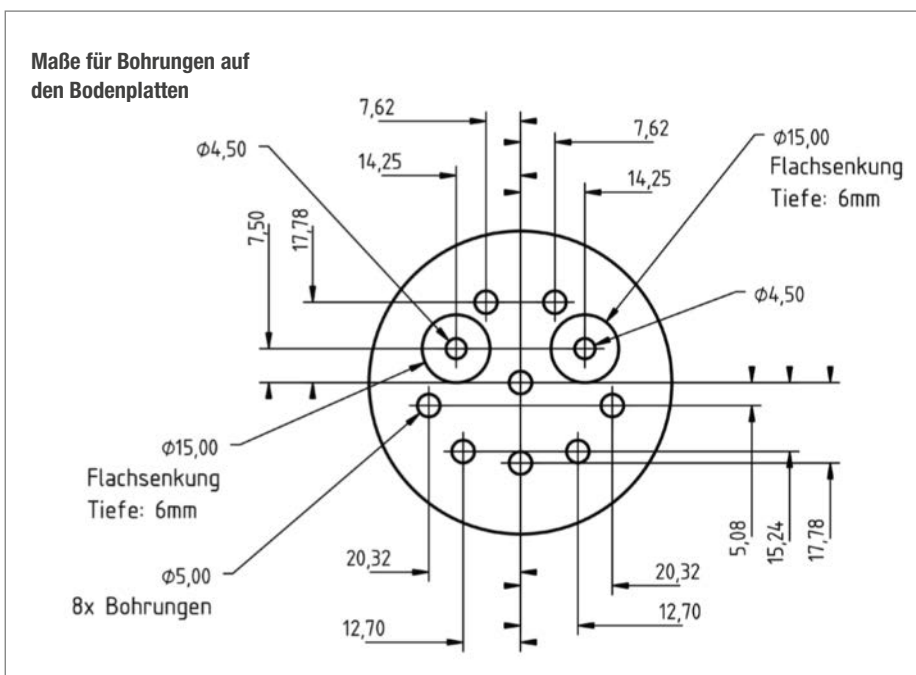
Ist festgelegt, welches Signal an welchen Stecker soll, kommt die Montage der Röhre. Zuerst habe ich alle Leitungen des Displays durch die acht Bohrungen geführt. Danach habe ich die Gewindestangen des Rahmens in die 4,5mm großen Bohrungen gesteckt und festgeschraubt. Anschließend sind die Signalleitungen mit dem Seitenschneider so weit zu kürzen, dass noch ca. 10mm aus der Bohrung ragen. Darauf habe ich die Enden abisoliert und mit etwas Lötzinn verzinnt, um sie an die Bananenstecker zu löten. Dieser Arbeitsschritt gestaltete sich aufgrund der kurzen Litzen leider etwas knifflig.

Nach erfolgreicher Lötarbeit werden die Stecker in die Bodenplatte geklebt. Hierzu kann man etwas Klebstoff in die Bohrungen geben und die Bananenstecker eindrücken. Die meisten handelsüblichen Klebemittel eignen sich zum Kleben von Metall. Wer auf Nummer sicher gehen will, sollte auf Zwei-Komponenten-Kleber zurückgreifen.

Zum Abschluss kommt das Ganze im wahren Sinne des Wortes unter die Haube. Die Kunststoffglocke wird auf die Bodenplatte gesetzt und fertig ist die Dixie-Röhre. Da zwischen Rahmen und Kunststoffglocke nicht viel Freiraum ist, ist darauf zu achten, diese nicht zu verkratzen. Sollte die Haube nicht über den Rahmen passen, kann dieser mit der Feile nachgearbeitet werden. Hierbei können die Außenkanten der Winkelprofile auf der Rückseite des Rahmens großzügig abgefeilt werden.

Elektronik

Der Arduino Nano hat alles an Bord, was zum Betreiben des Displays und später der „Röhre“, notwendig ist. Da ich die Uhr mit Ziffern in



Form von Bildern darstellen will, brauche ich nur ein SD-Card-Shield, um die Bilder von einer SD-Karte zu laden. Zur Ansteuerung des ST7735-Bildcontrollers, welcher im Display verbaut ist, sowie zum Einbinden des Shields gibt es Bibliotheken mit Beispielprogrammen für die Arduino IDE. An diesen habe ich mich beim Schaltplan orientiert.

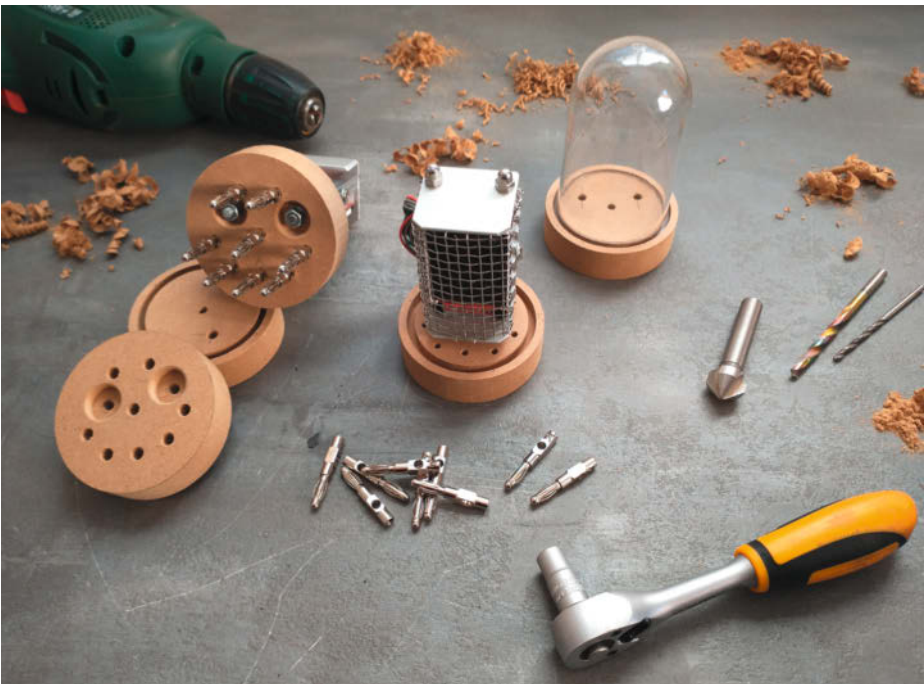
Werfen wir zunächst einen Blick auf den allgemeinen Ablauf, wie Grafiken ihren Weg von der Speicherkarte zum Display finden: Bilder legen wir als Bitmap-Datei auf der SD-Karte ab. Nach einem entsprechenden Funktionsaufruf lädt der Arduino ein Bild von der Karte und schreibt es in das Display. Die Kommunikation erfolgt über die Schnittstelle SPI (Serial Peripheral Interface) – ein Bus-System, wobei der Arduino der Controller ist und Röhre sowie Speicherkarte die Peripherie. Die drei Signale SCK, MOSI und MISO werden gemeinsam genutzt und liegen beim Nano fest an den angegebenen Pins (siehe Tabelle SPI-Anschlüsse).

Zudem hat jedes Peripheriegerät eine Chip-Select-Leitung (CS), um festzulegen, mit welchem Gerät der Controller gerade kommuniziert. Beim Arduino sind diese Signale nicht fest vorgegeben und es kann praktisch jeder Digital-Pin als Chip-Select verwendet werden. Außerdem sind die Anschlüsse für Masse (GND) und die Spannungsversorgung zu verbinden. Der Displaycontroller (VCC) wird mit 5 Volt und die Hintergrundbeleuchtung (LED) mit 3,3 Volt vom Arduino betrieben.

Test-Aufbau

Mit einem Funktionstest nach der Montage kann man prüfen, ob das Herausführen der Signale vom Display auf die Bananenstecker geklappt hat. Bei eventuellen Problemen später ist somit ein Fehler seitens der Röhre auszuschließen. Für den Test habe ich den Nano samt SD-Shield zunächst auf einem Steckbrett untergebracht und ein Testprogramm aufgespielt. Die Röhren lassen sich dafür einfach verbinden, indem Drähte an die acht Bananenbuchsen gelötet, diese an die Röhren-Pins gesteckt und die Drähte mit dem Steckbrett verbunden werden. Die Verdrahtung aller Komponenten erfolgte mit Steckdrähten gemäß der Zuordnung im Schaltplan.

Wer an dieser Stelle mehr Zeit investieren und sich zudem handwerklich austoben



Bodenplatten, teilweise bearbeitet

möchte, kann einen provisorischen Sockel aus Holz oder Ähnlichem anfertigen, um die Buchsen dort zu befestigen. Mit der Bohrschablone gelingt dies relativ schnell und einfach.

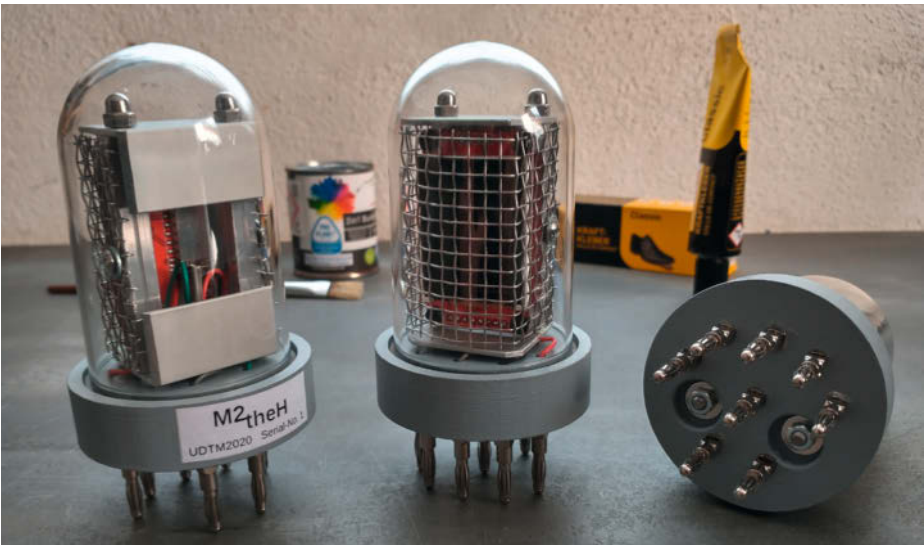
Software für den Röhren-Test

Wechseln wir nun zur Programmierumgebung des Arduino. Über die Bibliotheksverwaltung sind die folgenden Bibliotheken zu suchen und installieren, mit denen das Display und die SD-Karte angesprochen werden: Adafruit GFX Library, Adafruit ST7735 and ST7789 Library, Adafruit SPIFlash, SdFat Adafruit Fork und Adafruit ImageReader Library. Sofern nicht

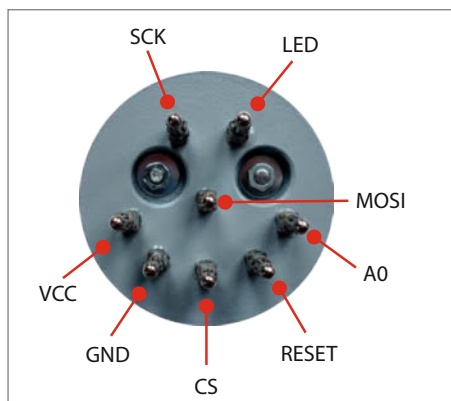
bereits vorhanden, verlangt jede dieser Bibliotheken noch weitere Libraries, die mit installiert werden müssen. Nötig ist auch TimerOne, die aber erst bei der Uhr-Software Verwendung findet. Über den Link in der Kurzinformat finden Sie weitere Infos zu den Bibliotheken und ihrer Einbindung in die Arduino IDE.

Als nächstes wird die SD-Karte vorbereitet und mit dem Dateisystem FAT32 formatiert. Bilder müssen direkt im Hauptverzeichnis abgelegt sein und folgende Kriterien erfüllen: Das Dateiformat ist Bitmap (*.bmp), die Farbtiefe 24 Bit und die Auflösung 160 x 128 Pixel groß – dies entspricht der Auflösung des Displays. Die Dateinamen sind frei wählbar. Für den folgenden Test habe ich *Testbild.bmp* ver-

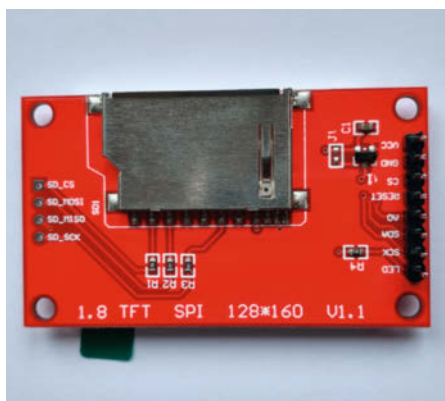
SPI-Anschlüsse		
Signal	Aufgabe	Arduino-Pin
SCK	Takt	D13
MOSI	Daten vom Controller an Peripherie	D11
MISO	Daten von Peripherie an Controller	D12



Fertige Dixie-Röhren



Pinbelegung der Röhre



Anschlüsse Display



Anschlüsse SD-Karten-Shield

wendet. Mit einem Grafikprogramm, wie beispielsweise dem kostenlosen GIMP, kann jedes beliebige Bild in das benötigte Format gebracht werden.

Steckt die SD-Karte mit Testbild im Shield und ist der Arduino mit dem Computer verbunden, kann der Test beginnen. Hierzu eignet sich das Beispielprogramm BreakoutST7735-160x128 (zu finden unter Datei/Beispiele/Adafruit ImageReader Library*). Im Programmcode muss man in der Zeile 94 noch den Dateinamen parrot.bmp zu Testbild.bmp ändern. Sinnvoll ist auch das Einfügen einer Verzögerung (delay(30000);) in Zeile 96 (s. Seite 50). Damit bleibt das Bild für

30 Sekunden im Display sichtbar, bevor das Beispielprogramm weiter läuft.

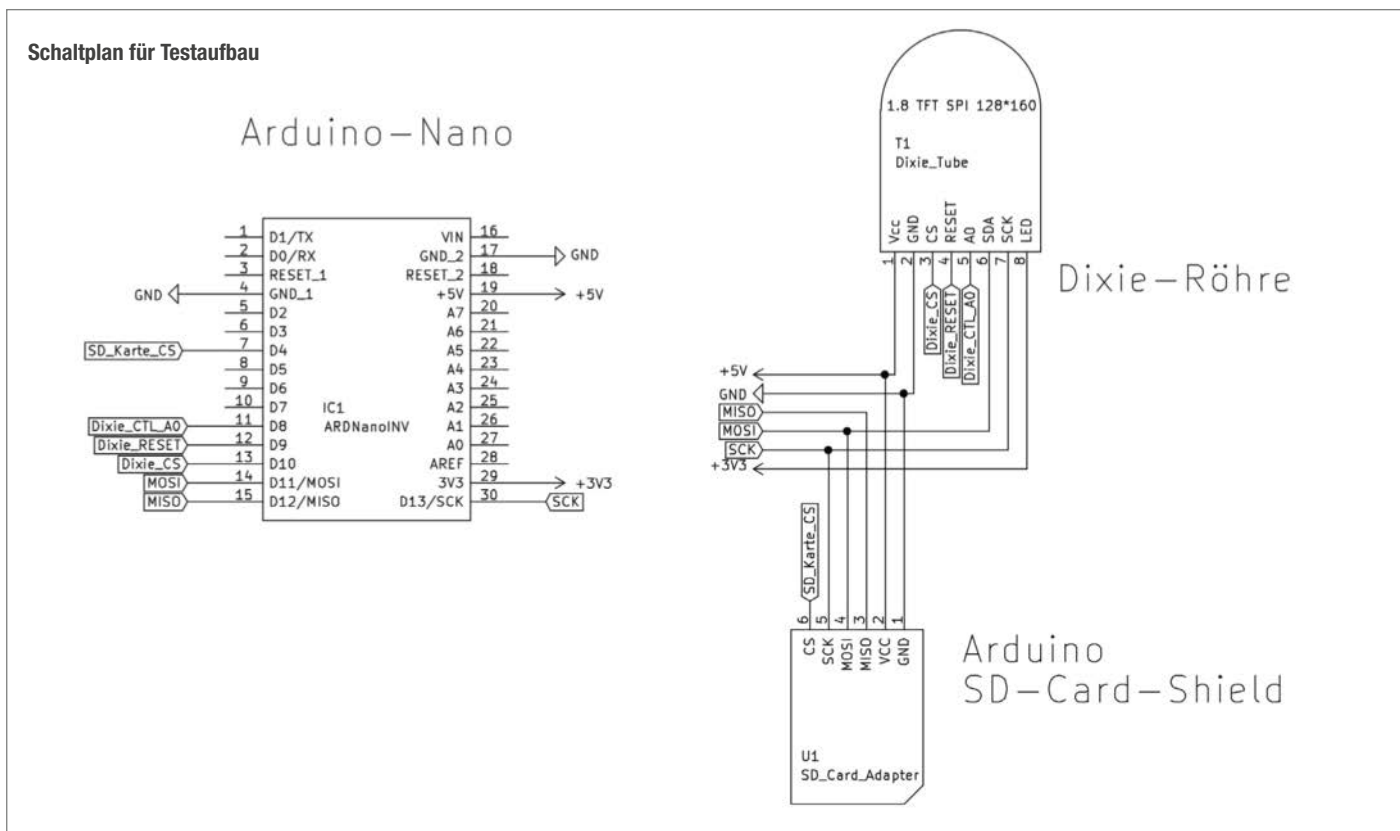
Abschließend kann das Programm hochgeladen werden. Wenn sich kein Fehler eingeschlichen hat, zeigt die Röhre das Testbild an. Wären noch mehr Bilder auf der Speicherkarte, würden sie nach den 30 Sekunden geladen. So bleibt das Display schwarz und das Programm stoppt.

Die Elektronik

Um die fünf Röhren als Uhr zu betreiben, habe ich die Schaltung des Testaufbaus um vier Röhren und zwei Taster erweitert und sie auf

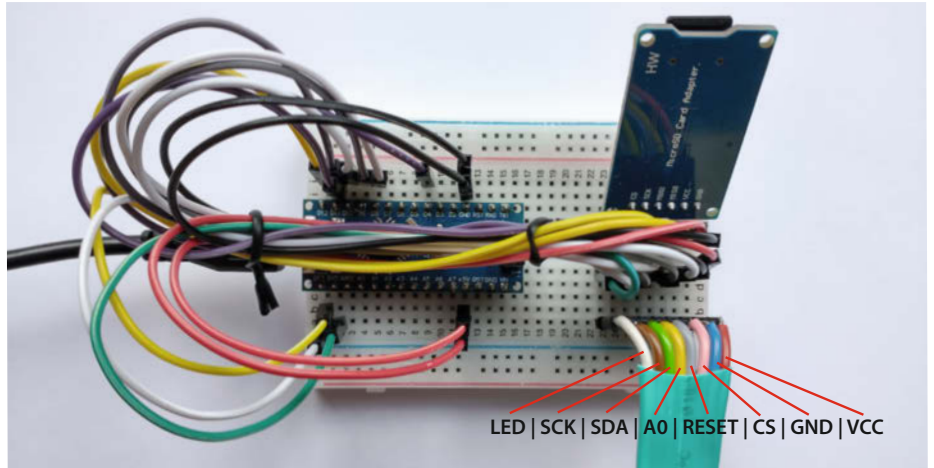
einer kleinen Lochrasterplatine (Rastermaß 2,54mm; 24x18 Bohrungen) aufgebaut. Den vollständigen Schaltplan finden Sie online (siehe Kurzinfo).

Die Spannungsversorgung erfolgt über den Arduino Nano. Betrieben am Computer oder später an einem USB-Netzteil mit mindestens 500mA Ausgangsstrom, ist für die 5-Volt-Aufgaben ausreichend Strom verfügbar. Problematisch sieht es hingegen auf der 3,3-Volt-Seite aus. Da fortan fünf Hintergrundbeleuchtungen zu versorgen sind, kommt ein maximaler Strom von circa 250mA zusammen. Der Nano schafft lediglich 50mA. Daher habe ich einen LD33V-Spannungsregler eingebaut,



der aus den 5 Volt des Arduinos 3,3 Volt bereitstellt, dabei aber bis zu 800mA Strom liefert. Somit hat die 3,3-Volt-Versorgung mehr Power und das Projekt bleibt während des Betriebs auch ohne Kühlkörper wohltemperiert.

Zum Einstellen der Uhrzeit nutze ich drei Taster, die jeweils über einen Kondensator und zwei Widerstände entprellt werden. Die Signale für die Röhren sind einheitlich auf Stiftleisten geführt, ebenso die des SD-Karten-Shields und die der Taster. Da die Digital-Pins des Nanos für meinen Aufbau nicht ausreichen, verwende ich auch Analog-Pins als digitale Ein- und Ausgänge. Das ist bei allen Analog-Pins mit Ausnahme von A6 und A7 möglich, die nur als analoge Eingänge nutzbar sind. Im Programmcode setze ich die Nummerierung der Digital-Pins dabei einfach fort (A0 = D14; A1 = D15; A2 = D16; usw.).



Testaufbau am Steckbrett mit Arduino Nano und SD-Karten-Shield: Die markierten Leitungen führen zur Dixie-Röhre.

Anfertigen der Platine

Für den Nachbau der Platine habe ich Bestückungspläne vorbereitet, die über den Kurzlink heruntergeladen werden können. Die gelben und grauen Leiterbahnen kennzeichnen jeweils isolierte Drahtbrücken auf der Platinenoberseite. Der Arduino kann in Buchsenleisten auf die Platine gesteckt oder direkt

eingelötet werden. Ich habe mich für die Steckvariante entschieden. Wer kein großer Freund von Lötarbeiten dieses Ausmaßes ist, kann an dieser Stelle auch mit einem ECAD-Programm zu Werke gehen und die Leiterplatte bei einem Hersteller bestellen.

Beim Löten ist es am besten, mit den Leiterbahnen zu beginnen und verzinnten Kupfer-

draht zu nutzen. Löcher, in die Bauteile gesteckt werden, dürfen natürlich nicht mit Lötzinn verschlossen werden. Als nächstes folgt die Bestückung der Stift- und Buchsenleisten. Hier ist es hilfreich, bei den Pins, an denen die Leiterbahnen auf der Platinenoberseite liegen, mit Skalpell oder Elektronikseitschneider Teile der Isolierung zu entfernen. So

Make:markt

BÜCHER / ZEITSCHRIFTEN



Der Verlag für kreative Köpfe!

Informatik und Elektronik können komplex, theoretisch und anstrengend sein. Es geht aber auch einfach, anschaulich und leicht nachvollziehbar – wenn man die Dinge in die eigenen Hände nimmt und zum »Maker« wird: Mit Büchern vom dpunkt.verlag.

www.dpunkt.de

ELEKTRONIK / ZUBEHÖR



Mein Gehäuse des Monats!

[https://bloess.berlin/...](https://bloess.berlin/)



... /aktuell



Was Maker schon alles geschaffen haben!

Die Antwort und viele Beispiele finden Leser in unseren Zeitschriften „Space – das Weltraum Magazin“, „Wissen 2020“ und dem „Urknall“ vieler Computer- und Make-Enthusiasten – dem „Retro Gamer“.

www.emedia.de



Code Mercenaries
Hard- und Software GmbH

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

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

Komponenten für intelligentes Licht:

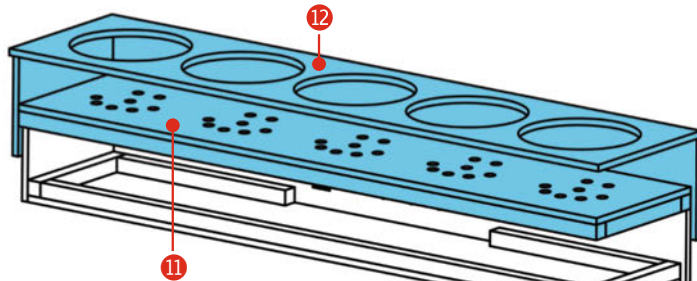
DMX, IEC62386

Ständig Angebote im Webstore:

www.codemerces.com/shop

Geänderter Beispiel-Code

```
Serial.print(F("Loading parrot.bmp to screen..."));
stat = reader.drawBMP("/Testbild.bmp", tft, 0, 0);
reader.printStatus(stat); // How'd we do?
delay(30000);
```



Das Gehäuse besteht aus zwei Hälften: oben (blau) und unten.

lassen sich diese leichter festlöten. Schließlich sind die übrigen Bauteile zu bestücken. Bei den beiden Elektrolytkondensatoren ist auf korrekte Polarität zu achten und beim Spannungsregler die Pinbelegung zu berücksichtigen. Zum Abschluss wird der Arduino Nano in die Buchsenleisten gesteckt.

Das Gehäuse

Wie in den 50er Jahren üblich, soll die Uhr ganz im Stil von „Brauner Ware“ – in

einem Gewand aus Holz – die Zeit verstreichen lassen. Dafür habe ich einen eher schlichten Aufbau gewählt. Das Gehäuse besteht aus zwei Hälften, die aufeinander gestellt werden. Im unteren Teil wird die Elektronik eingebaut, im oberen die Rohre. Als Ausgangsmaterial eignen sich 4mm Sperrholzplatten. Sie lassen sich mit einem Teppichmesser und einem Stahllineal gut auf Maß schneiden.

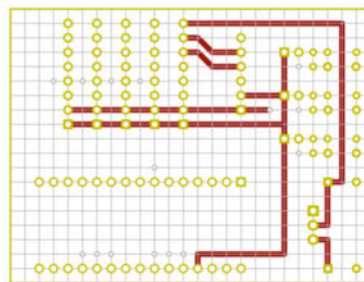
Der obere Teil bildet einen Sockel für die Rohre. Hier erweist sich die Bohrschablone

ein weiteres Mal als sehr hilfreich. Mit ihr kann man alle Bohrungen für die 40 Bananen-Buchsen auf der Sockelplatte (11) anzeichnen. Die runden Ausschnitte in der Deckplatte (12) habe ich mit einer Laubsäge geschnitten. Die vorbereiteten Platten werden dann mit Holzleim verklebt: Als erstes habe ich die Vierkanthölzer, auf denen die Sockelplatte aufliegen soll, auf die Seitenteile geklebt. Die Höhe ist so gewählt, dass von den Röhren nur die Glocken aus der Deckplatte ragen. Als nächstes muss man die vier Seitenplatten rechtwinklig zueinander ausrichten und verleimen, so dass die Vierkanthölzer innen liegen. Während der Trocknungszeit ist es sinnvoll, die Platten mit Schraubzwingen zu fixieren.

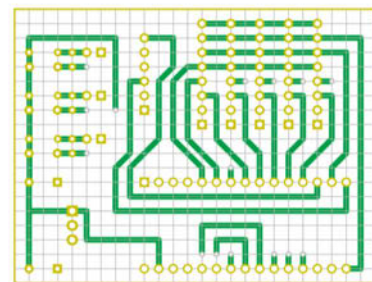
Im Anschluss kann die Sockelplatte auf die Vierkanthölzer geklebt werden. Wer das Gehäuse lackieren oder Möbelversiegelung auftragen möchte, sollte nun mit den innen liegenden Plattenseiten beginnen. Darauf folgend muss man die Bananen-Buchsen einschrauben und die Deckplatte aufkleben. Anschließend kann man mit feinem Schleifpapier die Platten außen bearbeiten und Klebekanten abrunden, bevor auch die Außenseiten lackiert werden.

In der unteren Gehäusehälfte sind alle elektronischen Bestandteile untergebracht. Die Platine und das SD-Card-Shield habe ich mit Distanzbolzen an der Bodenplatte befestigt, alternativ können sie auch eingeklebt werden. Rückseitig sind die drei Taster und eine USB-B-Buchse eingebaut. Letztere ist mit Zwei-Komponenten-Kleber an die Holzplatte

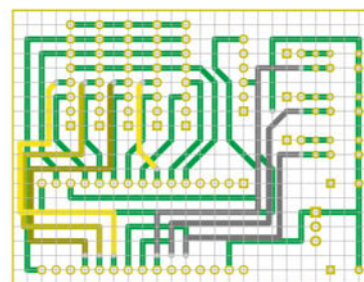
Bestückungspläne der Platine



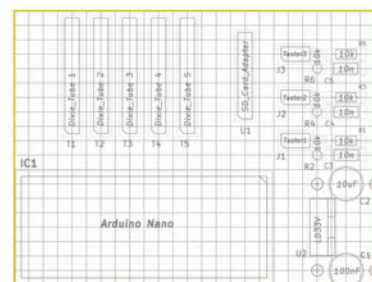
Oberseite – Leiterbahnen



Unterseite – Leiterbahnen



Oberseite – Drahtbrücken



Bauteile

geklebt und führt den Mini-USB-Anschluss des Arduinos aus dem Gehäuse.

Auch hier habe ich mit den Vierkanthölzern begonnen und sie bündig mit den Kanten auf die Bodenplatte geklebt. Danach habe ich die Seitenteile mit der Bodenplatte verleimt und die Platten außen mit feinem Schleifpapier bearbeitet. Ist der Leim trocken, können alle elektronischen Komponenten montiert und anhand des Schaltplans miteinander verdrahtet werden. Zwischen dem Arduino und der USB-B-Buchse steckt ein Mini-USB Kabel, wobei ich einen Stecker abgezwickelt und das Kabel direkt an die Buchse gelötet habe. Zum Verbinden der Bananen-Buchsen eignet sich ein Netzwerkkabel. Dies hat acht farblich unterschiedliche Adern, die sich während des Lötens gut zuordnen lassen.

Software

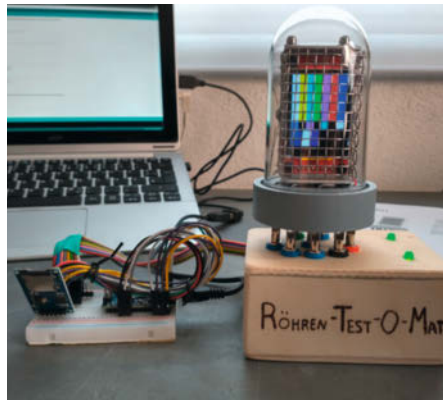
Beim Programm der Uhr habe ich mich zum Laden und Anzeigen der Bilder an dem zuvor verwendeten Beispielpogramm orientiert. Für die Uhrzeit kommt ein Timer-Interrupt zum Einsatz. Jede Millisekunde erfolgt der Aufruf einer Interrupt-Funktion. Hier werden die Variablen für Sekunden, Minuten und Stunden entsprechend hochgezählt. Zu jeder vollen Minute erfolgt nacheinander die Aktualisierung der Röhre Nummer vier und fünf. Bei jedem Stundenwechsel aktualisiert der Mikrocontroller außerdem die Röhren Nummer eins und zwei.

Einmal geladen, zeigt Röhre drei dauerhaft den Doppelpunkt an, der Stunden und Minuten voneinander trennt. Zudem wird in ihr das Menü zum Einstellen der Uhrzeit dargestellt. Die Abfrage der dafür notwendigen drei Taster erfolgt zyklisch alle 250 Millisekunden. Den Arduino-Sketch und die Ziffer-Bilder finden Sie über den Link in der Kurzinfor. Um die Software in die Uhr zu laden, nutzen wir ein letztes Mal die Arduino IDE. Nach dem Einschalten zeigt die Uhr neben dem Doppelpunkt zunächst nur Striche an, solange die Zeit nicht eingestellt ist.

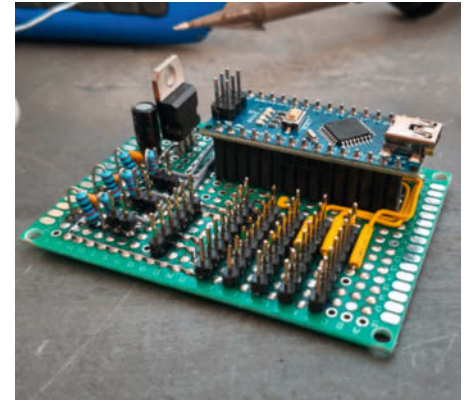
Das Menü zum Einstellen der Uhrzeit öffnet sich durch das Betätigen von Taster 3 (Pin A5/D19). Mit Taster 2 (Pin A4/D18) und Taster 1 (Pin A3/D17) können zunächst die Minuten vor- bzw. zurückgestellt werden. Erneutes Drücken von Taster 3 ermöglicht das Einstellen der Stunden. Dies geschieht wie zuvor über die beiden anderen Taster. Ist die Uhrzeit korrekt eingestellt, kann man mit einem weiteren Druck auf Taster 3 das Menü verlassen. Nun wird die Uhrzeit nacheinander in die Dixie-Röhren geschrieben und die Uhr taktet stoisch vor sich hin.

Fazit

Die Dixie-Röhre hat ihren eigenen Charme, dennoch bleibt das Vorbild unerreicht. Damit



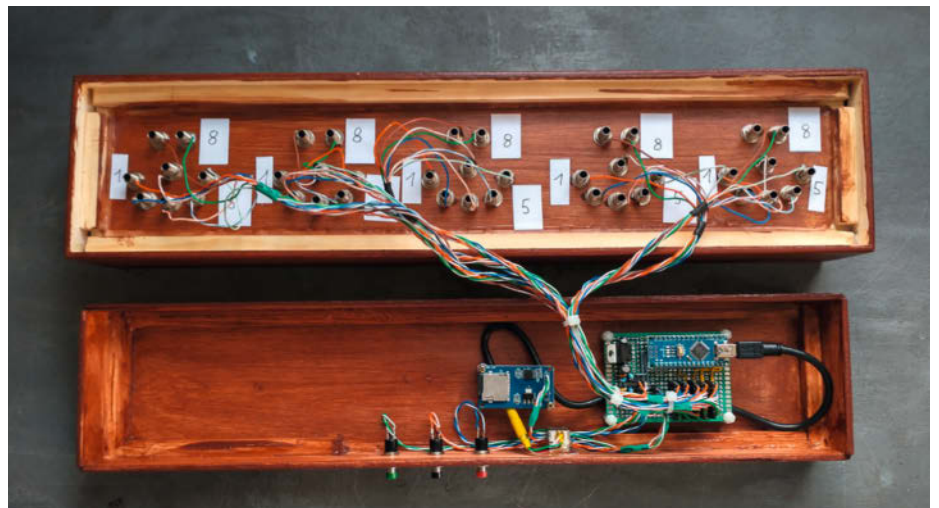
Röhre mit Testbild: Der provisorische Sockel ist aus Sperrholz und mit grünen Deko-LEDs versehen.



Fertig bestückte Platine der Dixie-Uhr

die Uhr sich von alleine stellt, könnte man sie mit einem DCF-Empfangsmodul zur Funkuhr umbauen oder mit einem Echtzeit-Modul für mehr Genauigkeit sorgen. Schließlich bietet die SPI-Schnittstelle der Röhre ganz neue Möglichkeiten. In unserer schnelllebigen Welt

vergeht die Zeit wie im Flug und ehe man sich versieht, steht wieder Weihnachten vor der Tür. Dann kann die Dixie-Röhre beispielsweise als CO₂-schonender und flammenloser Ersatz für Kerzen verwendet werden. —hch

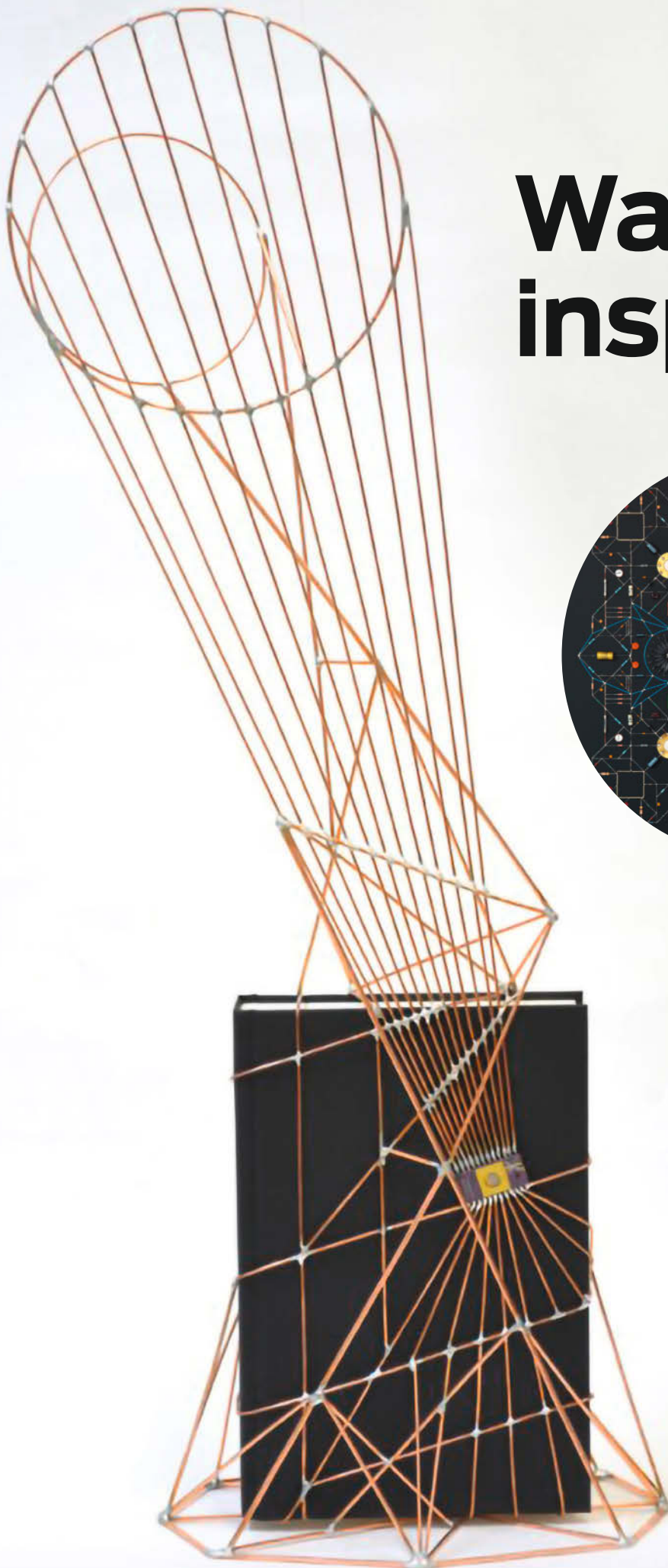


Blick ins Gehäuseinnere (fertig montiert und verdrahtet)



Dixie-Röhren als Kerzenersatz

Was uns inspiriert

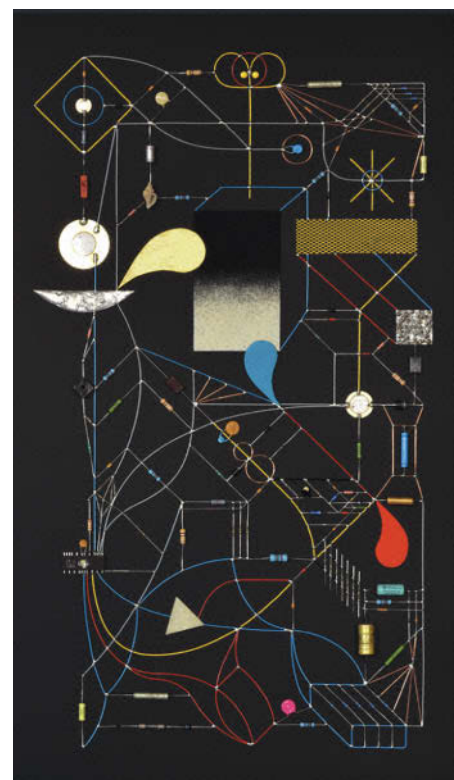


Schaltkreis-skulpturen

Der Künstler Leonardo Ulian liebt es, elektronische Geräte auseinander zu nehmen und sich in der Betrachtung der Muster von Schaltkreisen zu verlieren. Er verwendet diese Inspiration, um abstrakte Kunstwerke und skulpturhafte Formen aus Kupferdraht und elektronischen Bauelementen zu weben. Seine gelöteten Kupferverbindungen nehmen die Form von tibetischen Mandalas und fraktalen Mustern an.

Ulian versteht Elektronikschaltungen und kann auch die Funktionsweisen der Bauelemente genau erklären. Dennoch sind seine Kunstwerke eher Patchworks elektronischer Komponenten, die man als Phantomplatinen bezeichnen könnte. Dabei bindet er gerne thematisch fremde Objekte – wie Bücher – ein und erweckt im Betrachter den Eindruck ferner, futuristischer Welten. —*rehu*

► leonardoulian.com



Was inspiriert Dich?

Wir freuen uns über Vorschläge an:
mail@make-magazin.de



Bilder: Allison Carter, Georgia Tech

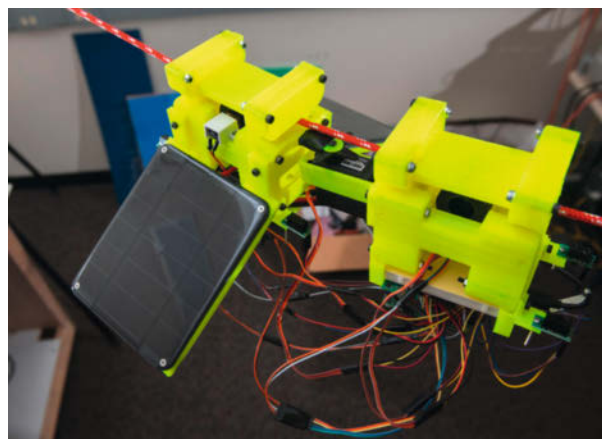
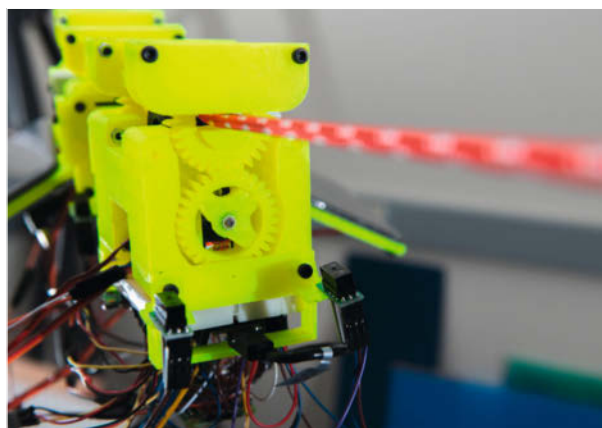
Faultier als Vorbild

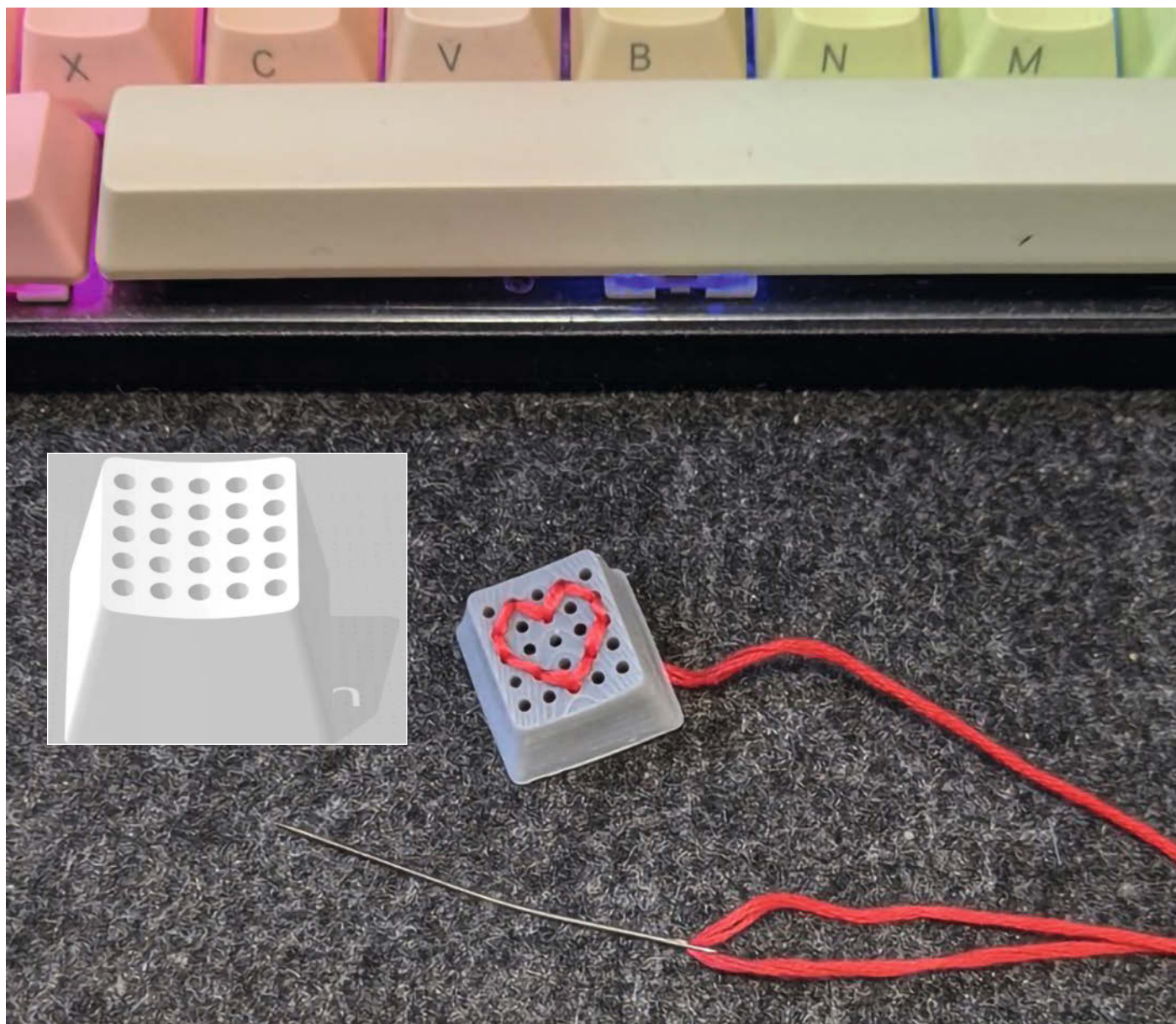
Die Überwachung von Tieren und Pflanzen im Botanischen Garten von Atlanta übernimmt derzeit der *Slothbot*, der sich dabei so wenig wie möglich bewegen soll. Seine Form ist daher vom Faultier inspiriert. Unter einer 3D-gedruckten Hülle verstecken sich Sensoren, die die Temperatur, den CO₂-Anteil in der Luft und weitere Daten aufzeichnen. Zwei Solarmodule versorgen den Roboter mit Spannung. Der Slothbot soll sich möglichst nur bewegen, wenn er Sonnenlicht zum Laden braucht. Und je weniger er sich bewegt, desto weniger Energie ist nötig. So kann er auch lang andauernde Fernüberwachungen übernehmen.

Ein wichtiger Aspekt dieses Projekts der Universität von Georgia Tech ist die Fortbewegung des Bots auf Stahlseilen. Gleichstrommotoren treiben seine Räder an, während ein Mechanismus aus zwei Zahnrädern und zwei Servos das Manövrieren über verzweigte Seile ermöglichen. Wo andere Roboter am Boden leicht feststecken oder an Hindernissen scheitern, vereinfacht die Fortbewegung an Outdoor-Seilen den Betrieb.

—hch

► news.gatech.edu/2020/06/16/slothbot-garden-demonstrates-hyper-efficient-conservation-robot





Bilder: Billie Ruben

Tastatur mit Kreuzstich

Fans mechanischer Tastaturen beeindrucken regelmäßig mit ihren kreativen Ideen. Wo bisher Farbkompositionen, individuell bedruckte Tastaturkappen und Beleuchtung die Hauptrolle spielten, kommt jetzt der gute alte Kreuzstich ins Spiel. Bei der Arbeit an einer selbst gestalteten Tastatur kam die Makerin Billie Ruben auf die Idee, Tastenkappen herzustellen, die man besticken kann.

Die ersten 3D-Druckvorlagen bietet sie bereits zum Download an. Die Tasten haben 5x5 Löcher auf der oberen Seite, so dass man die Belegung der Taste mit Garn aufsticken kann.

Wie Billie Ruben selbst schreibt, wird die fertige Tastatur wohl weniger ein Alltagsgegenstand als ein hübsches Vorführprojekt. Falls es allerdings möglich sein sollte, die bestickbaren Tasten so weit zu verkleinern, dass man eine durchsichtige Schutzkappe darüber setzen kann, wären Kreuzstich-Tastaturen tatsächlich benutzbar. —esk

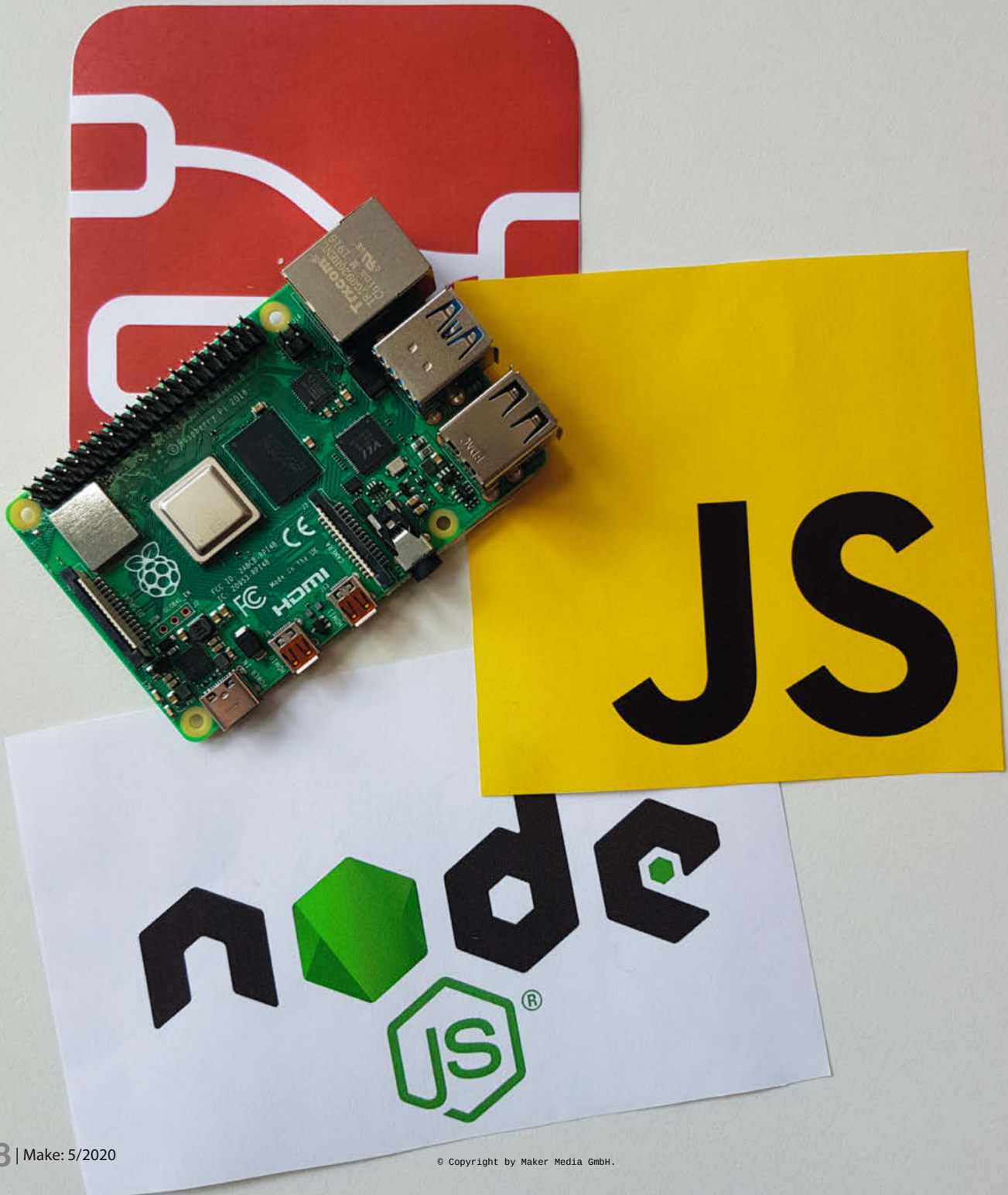
► prusaprinters.org/social/84905-billie-ruben/prints



Eigene Knoten für Node-RED

Node-RED ist bei Bastlern äußerst beliebt, weil es Nutzern ohne große Programmierkenntnisse die Erstellung von IoT-Anwendungen ermöglicht. Das liegt nicht zuletzt an vielen nützlichen Erweiterungen, die unzählige Internet-Dienste und Geräte ohne großen Aufwand einbinden.

von Maik Schmidt



Mit Node-RED ist es zum Beispiel kein Problem, Sensoren auszulesen, die mit den GPIO-Pins eines Raspberry Pi verbunden sind, und die ermittelten Daten per Twitter oder per E-Mail zu versenden. Die Verteilung von Informationen im lokalen Netz über das IoT-Protokoll MQTT ist ebenfalls ein Klacks und Node-RED unterstützt sogar die Erstellung von Web-Anwendungen mit durchaus schicken Oberflächen.

Das funktioniert alles deshalb so gut, weil es Erweiterungen gibt, die komplizierte Details verstecken, so dass Nutzer komplexe Anwendungen mit wenigen Handgriffen zusammenklicken können. Bei Bedarf lassen sich solche Erweiterungen aber auch leicht selbst entwickeln. Wir zeigen an dieser Stelle, wie das geht.

Schnelleinstieg

Wer eine Erweiterung für Node-RED schreiben möchte, sollte verstehen, wie Node-RED unter der Haube funktioniert. Im Grunde ist die Software recht übersichtlich und besteht aus einem Editor und einer Laufzeitumgebung. Mit dem Editor können Nutzer verschiedene Knoten zu so genannten Flows kombinieren. Knoten sind eine Art Miniprogramme, die einzelne Aufgaben übernehmen und einen Output erzeugen, den sie an andere Knoten weiterleiten. Alles zusammen ergibt dann einen Flow.

Die Laufzeitumgebung sorgt anschließend dafür, dass die Knoten eines Flows im Hintergrund ausgeführt werden und ihre Dienste verrichten.

Die technologische Basis all dessen ist Node.js. Node.js ist eine JavaScript-Implementierung, die auf der des Google Chrome-Browsers basiert. Sie erlaubt nicht nur die Ausführung von JavaScript-Code außerhalb eines Browsers, sondern bietet eine asynchrone und ereignisgesteuerte Plattform mit nicht-blokkierenden Ein- und Ausgabe-Operationen.

Das klingt alles ein wenig akademisch, ist im Kern aber ganz simpel. Programme, die unter Node.js laufen, warten per Voreinstellung nicht auf die Ergebnisse von Ein- und Ausgabe-Operationen, sondern geben wäh-

rend der Wartezeit die Kontrolle an andere Programme beziehungsweise Programmabschnitte ab. Diese Vorgehensweise nutzt die Ressourcen eines Rechners optimal und passt hervorragend zur IoT-Welt, die zum Großteil durch Ereignisse gesteuert wird.

Neben vielen anderen Aufgaben eignet sich Node-RED daher hervorragend zur Integration von Web-Diensten aller Art, wie zum Beispiel solchen für Wetter- oder Pollenflug-Vorhersagen. So gut wie alles, was das

Kurzinfo

- » Überblick Node-RED
- » Eigene Nodes programmieren
- » COVID-19-API nutzen

Checkliste



Zeitaufwand:

1 Tag



Kosten:

0 bis 50 Euro, je nachdem, ob man einen Pi benutzt



Programmieren:

JavaScript, Node.js

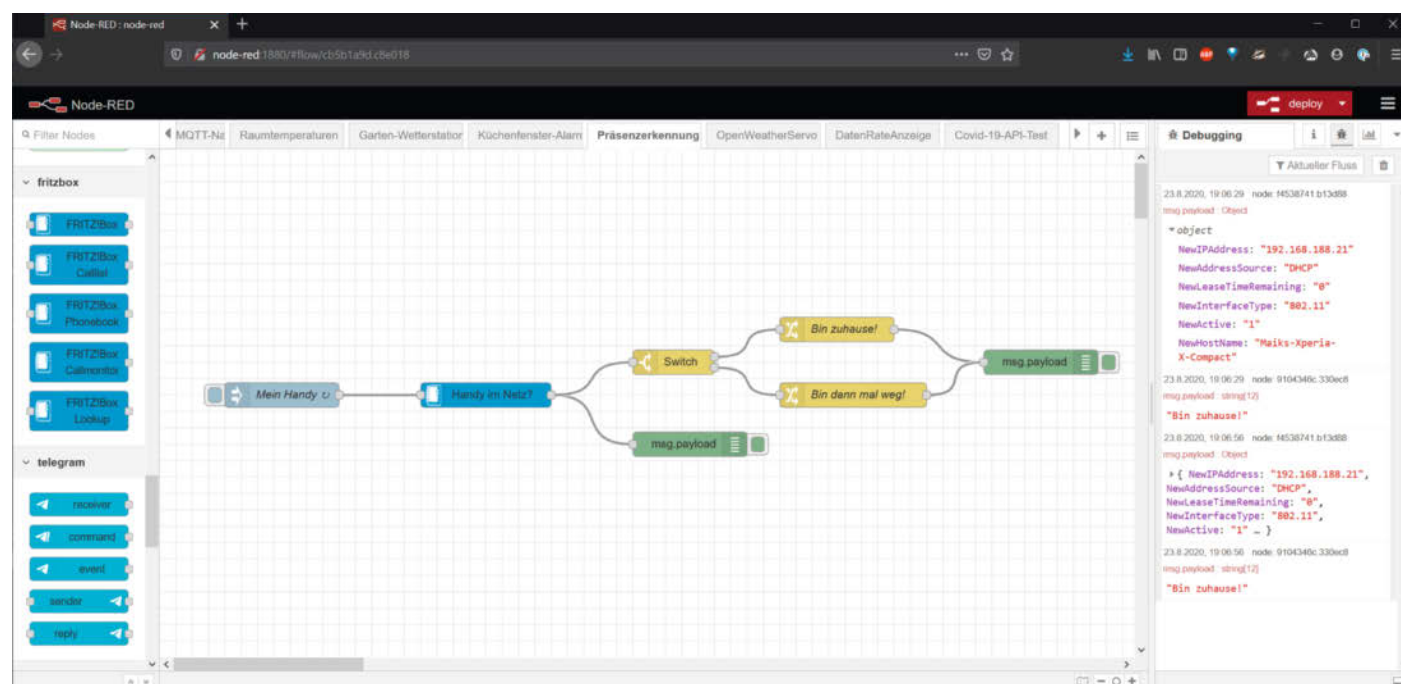
Material

- » Raspberry Pi 2-4 (plus Zubehör)

Mehr zum Thema

- » Markus Ulsass, Alles im Fluss, Make IoT Special 2016, S. 58
- » Markus Ulsass, Wolfhüllambiente mit Luftgütemessung, Make IoT Special 2016, S. 66

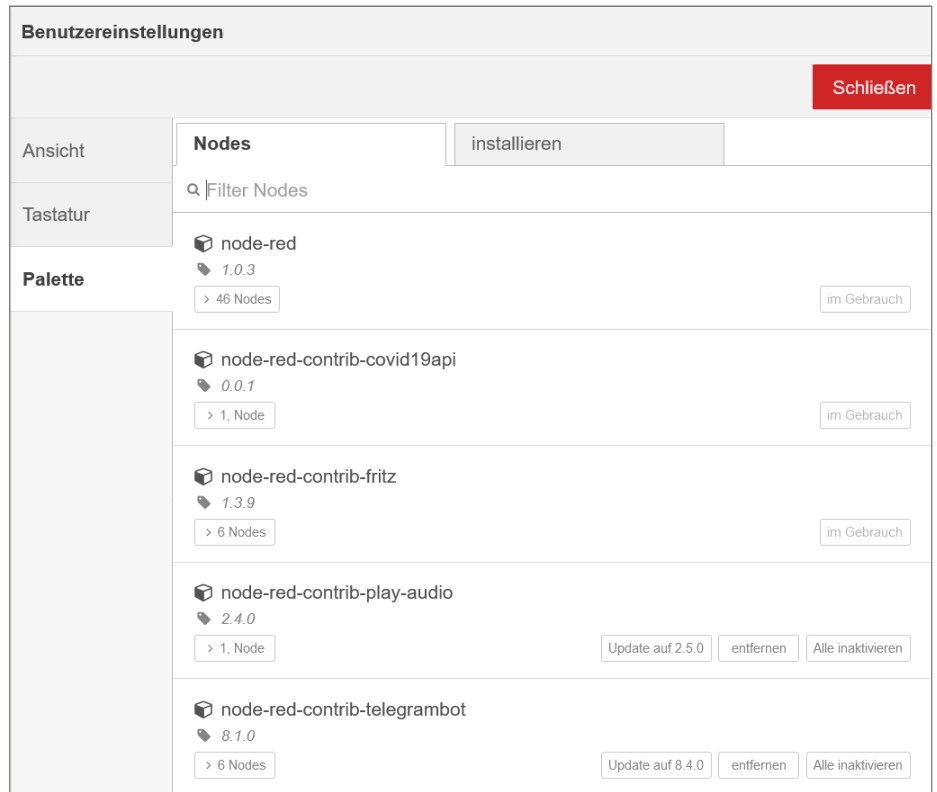
Alles zum Artikel
im Web unter
make-magazin.de/xuec



Der Node-RED-Editor ist aufgeräumt. Links sind die Knoten, in der Mitte der aktuelle Flow und rechts diverse Ausgabe-Fenster.



Die COVID-19-API bereitet die wichtigsten Informationen für die meisten Länder kompakt auf.



Normalerweise administrieren Nutzer ihre Pakete über den Editor und nicht über npm.

Internet zu bieten hat, kann Node-RED anzapfen und in Flows verwenden. Das gilt auch für Daten über die Corona-Pandemie und es bedarf keines großen Aufwands, sie mithilfe eines eigenen Knoten-Typs zu verarbeiten. In einer ersten Ausbaustufe soll ein solcher Knoten die neu gemeldeten Infektionen des Vortages für ein vorgegebenes Land ermitteln.

Qual der Wahl

Zunächst steht die Auswahl eines passenden Anbieters an, denn aktuelle Fallzahlen gibt es an vielen Stellen im Netz. Nicht alle sind jedoch akkurat, leicht maschinell zu verarbeiten und obendrein auch noch kostenlos. Aus der Menge sticht die COVID-19-API hervor, denn sie stellt die Daten der Johns Hopkins-Univer-

sität tagesaktuell und größtenteils kostenlos über eine REST-Schnittstelle zur Verfügung. Es gibt auch ein kostenpflichtiges Premium-Angebot, aber das ist nur für professionelle Anwender interessant.

Das Schöne an REST-Schnittstellen ist, dass es so einfach ist, sie auszuprobieren. Für Experimente reicht ein Web-Browser oder ein Kommandozeilen-Werkzeug wie *curl* oder *wget*. Das gilt auch für die COVID-19-API, denn sie basiert auf simplen JSON-Dokumenten und verfügt über eine umfangreiche Dokumentation.

Der wichtigste Anlaufpunkt der API ist die Tageszusammenfassung, die alle wichtigen Daten für jedes Land enthält, das seine Corona-Situation veröffentlicht. Unter der URL liegt ein circa 40KB großes JSON-Dokument, das

ein Feld von Objekten repräsentiert. Jedes dieser Objekte beschreibt die aktuelle Corona-Situation eines einzelnen Landes. Es enthält unter anderem die insgesamt bestätigten Infektionen (*TotalConfirmed*), die Anzahl der bisher an COVID-19 verstorbenen Personen (*TotalDeaths*) und die am Vortag gemeldeten Neuinfektionen (*NewConfirmed*).

Um sich zum Beispiel ein Bild über die Lage in Deutschland zu machen, reicht es, das Summary-Dokument zu laden und nach dem Objekt zu suchen, dessen *Country*-Eigenschaft den Wert *Germany* hat.

Knoten entwirren

Prinzipiell lässt sich das schon mit den Bordmitteln von Node-RED erledigen, denn es gibt Knoten für das Versenden von HTTP-Anfragen und auch welche zur Verarbeitung von JSON-Dokumenten. Hier geht es aber um die beispielhafte Erstellung eines eigenen Knoten-Typs, der gegebenenfalls um Eigenschaften erweitert werden kann, die sich mit den eingebauten Knoten wiederum nur umständlich umsetzen lassen. Darüber hinaus kann ein eigener Knoten-Typ anderen Anwendern zur Verfügung gestellt werden, so dass diese sich nicht selbst in die API einarbeiten müssen.

Weil Node-RED von Anfang an für Erweiterungen konzipiert wurde, ist die Implementierung eines eigenen Knotens kein Hexen-

package.json

```
01 {
02   "name": "node-red-contrib-covid19api",
03   "version": "0.0.1",
04   "dependencies": {
05     "node-fetch": "~2.6.0"
06   },
07   "node-red": {
08     "nodes": {
09       "covid19api": "covid19api.js"
10     }
11   }
}
```

werk. Deshalb gibt es im Netz eine Unmenge von Erweiterungen und zwar sowohl Knoten als auch ganze Flows.

Das Teilen von Flows ist besonders unkompliziert, weil sie nur JSON-Dokumente sind, die über die Export- und Import-Funktionen von Node-RED erzeugt und wieder eingelesen werden können. Das ist bei Knoten etwas komplexer, aber nicht sonderlich schwierig. Dazu gleich mehr.

Node.js-Pakete haben großen Anteil an der Popularität von Node.js, denn sie bilden eine umfangreiche Bibliothek an vorgefertigten, kostenlosen Lösungen, auf die Entwickler jederzeit zurückgreifen können. Diese Pakete werden mit einem Werkzeug namens *npm* (Node Package Manager) verwaltet. Wie die meisten Paket-Manager bietet npm Funktionen zum Installieren, Deinstallieren und Aktualisieren von Paketen an.

Node-RED-Nutzer kommen in der Regel nicht direkt mit npm in Kontakt, weil der Editor die entsprechenden Operationen im Menü *Palette verwalten* versteckt. Dieser Menüpunkt delegiert im Hintergrund aber alle Arbeiten an npm.

Pakete packen

Im Grunde ist jeder Node-RED-Knoten ein npm-Paket, das in der Regel mindestens drei Dateien enthält. Zentral ist die Datei *package.json*, die alle Meta-Informationen des npm-Pakets enthält. Dazu zählen unter anderem der Name des Pakets, die Versionsnummer, etwaige Abhängigkeiten und die Lizenz, unter der das Paket verwendet werden darf. Pflichtfelder sind allerdings nur der Name und die Version.

Listing *package.json* zeigt den zum COVID-19-Knoten gehörenden Code. Der Name des npm-Pakets ist *node-red-contrib-covid19api* und entspricht somit der Konvention, dass Node-RED-Erweiterungen mit dem Präfix *node-red-contrib-* beginnen. Die Version des Pakets ist 0.0.1 und entspricht damit dem Schema des Semantic Versioning, dem alle npm-Pakete folgen müssen.

Weil der COVID-19-Knoten per HTTP auf die COVID-19-API zugreift, definiert die Datei *package.json* eine Abhängigkeit zum npm-Paket *node-fetch*. Dieses wird dann bei der Installation automatisch mitinstalliert.

Bis hierhin entsprechen alle Einträge in der Datei den Standard-Einträgen, die in so gut wie jedem npm-Paket zu finden sind. Speziell für Node-Red sind die Angaben unterhalb des Attributs *node-red*.

Sie teilen der Außenwelt mit, welche Knoten das Paket bereitstellt und in welchen JavaScript-Dateien deren Implementierung zu finden ist. In diesem Fall wird nur ein neuer Knoten mit dem Namen *covid19api* definiert und dessen Quelltext enthält die Datei *covi-*

covid19api Node bearbeiten

Löschen

Abbrechen

Fertig

⚙ Properties

⚙ 📄 🖨

📌 Name

Name

Der neue Knoten bekommt automatisch einen Konfigurationsdialog.

Knoten-Generator

Die meisten Web-Dienste basieren auf dem Architekturstil REST (Representational State Transfer) und folgen daher alle ähnlichen Mustern. Da liegt es nahe, den Code für Node-RED-Knoten, die auf solche Dienste zugreifen, zu generieren und nicht jedes Mal von Hand zu schreiben.

Das Werkzeug Node Generator erledigt genau das und erzeugt zum Beispiel Code

anhand einer OpenAPI (Swagger)-Definition oder sogar aus einem Node-RED-Funktionsknoten.

Wer vorhat, einen neuen Knoten für einen Internet-Dienst zu entwickeln, sollte also zuerst prüfen, ob der Node Generator den Entwicklungsprozess nicht erheblich beschleunigen kann, statt alles manuell einzugeben.

inject Node bearbeiten

Löschen

Abbrechen

Fertig

⚙ Properties

⚙ 📄 🖨

✉ Nutzdaten

▼ a_z Germany

📄 Topic

☐ Einmal nach der Injektion

0.1

Sekunden, dann

🔄 Wiederholen

keine

▼

📌 Name

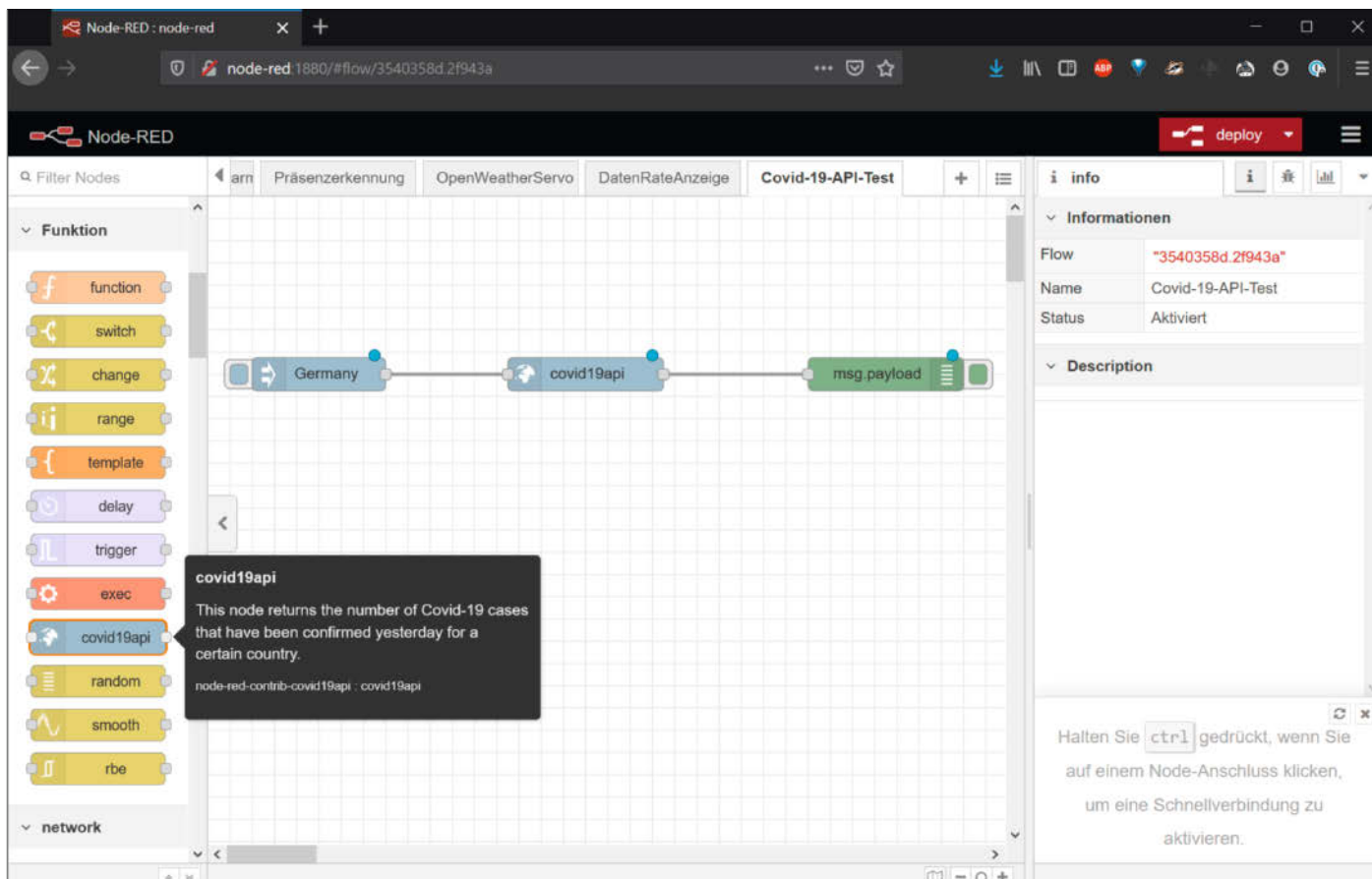
Name

Hinweis: "Intervall zwischen den Uhrzeiten" und "zu einem bestimmten Zeitpunkt" verwendet "cron".

"Intervall" sollte weniger als 596 Stunden betragen.

Siehe Informationsfenster für Details.

Im Inject-Knoten wird das zu prüfende Land hinterlegt.



Der neue Knoten reiht sich optisch nahtlos in die vorhandenen ein.

d19api.js. Ein Paket kann beliebig viele Knoten definieren und die dazu gehörigen Quelltext-Dateien müssen sich nicht im selben Verzeichnis befinden wie die Datei *package.json*. Sie können auch in Unterverzeichnissen liegen.

Listing *covid19api.js* zeigt den Inhalt der Quelltext-Datei. Es sind nur wenige Zeilen, aber die haben es in sich, denn sie übernehmen nicht nur die Integration mit Node-RED, sondern auch die Auswertung der COVID-19-API.

covid19api.js

```
01 const fetch = require('node-fetch');
02 const Covid19Api = 'https://api.covid19api.com/summary';
03
04 module.exports = function(RED) {
05   function Covid19ApiNode(config) {
06     RED.nodes.createNode(this, config);
07     const node = this;
08     node.on('input', function(msg) {
09       fetch(Covid19Api)
10         .then(response => response.json())
11         .then(data => {
12           const country = msg.payload
13           const summary = data.Countries.find(c => c.Country === country);
14           return summary;
15         })
16         .then(summary => {
17           msg.payload = summary ? summary.NewConfirmed : -1;
18           node.send(msg);
19         })
20         .catch(error => console.log(error));
21     });
22   }
23   RED.nodes.registerType('covid19api', Covid19ApiNode);
24 }
```

Zuerst bindet der Code die *node-fetch*-Bibliothek mittels der *require*-Anweisung ein und weist das Ergebnis der Konstanten *fetch* zu. Die Konstante *Covid19Api* enthält die URL der täglichen Zusammenfassung, die die COVID-19-API anbietet.

Als nächstes definiert das Programm eine anonyme Funktion, die vom Paket exportiert wird. Diese Funktion ruft Node-RED später auf und übergibt dabei eine Referenz auf das Laufzeitsystem, die konventionsgemäß RED heißt. Die anonyme Funktion definiert wiederum eine Funktion mit dem Namen *Covid19ApiNode*, welche die eigentliche Geschäftslogik des Knotens implementiert. Am Ende der Datei meldet dann ein Aufruf der Funktion *registerType* diese Funktion bei Node-RED an und assoziiert sie mit dem Knoten-Typ *covid19api*.

Der Ablauf ist also wie folgt: *package.json* spezifiziert den Knoten-Namen und die dazu gehörende JavaScript-Datei. Die Datei exportiert eine Funktion, die den Knoten-Namen mit dieser Funktion assoziiert und sie bei Node-RED registriert. Node-RED ruft diese Funktion auf, wenn der Knoten verwendet wird. Darüber hinaus kann die Datei natürlich noch viel mehr Code mit der eigentlichen Implementierung des Knotens enthalten.

Sonderheft Make Special Node-RED

Das Sonderheft führt in die Grundlagen von Node-RED und MQTT sowie die Installation und Konfiguration der freien Software auf dem Raspberry Pi ein. Durch den mitgelieferten WLAN-Mikrocontroller ESP8266 D1 mini können Sie ihre Sensoren und Aktoren in eigenen Projekten drahtlos per MQTT und Node-RED kontrollieren. Drei Projekte zeigen exemplarisch, wie man Sensordaten abfragt, Daten aus dem Internet holt, weiterverarbeitet und etwa einen Motor fernsteuert.

Projekte mit der Fritzbox

Auch aus der Fritzbox lässt sich mit Node-RED mehr rausholen: Dank einer offenen Programmierschnittstelle und vorgefertigter Node-RED-Bausteine kann

man auf einfache Weise auf Systemdaten zugreifen und diese auswerten. Eine Präsenzerkennung zeigt an, wenn sich das Smartphone ins heimische WLAN einbucht und eine Bandbreitenanzeige mit ESP8266 und LED-Pixel-Streifen visualisiert die aktuelle Auslastung des Internetzugangs. Mit speziellen Knoten lassen sich zudem die DECT-fähigen Steckdosen und Thermostate von AVM steuern. Alle Projekte gibt es mit Beispielcode und den verwendeten Flows zum Download. Ein abschließender Artikel führt in die Programmierung mit JavaScript und den Umgang mit Node.js ein.

Das 80 Seiten umfassende gedruckte Heft mit dem ESP8266 D1 mini gibt es exklusiv im heise shop (<https://heise.de/-4836285>) für 24,95 Euro versandkostenfrei zu bestellen. Als PDF ohne Board gibt es das Heft für 14,95 Euro.



Daten sammeln

Jetzt gilt es herauszufinden, wie die Funktion `Covid19ApiNode` ihre Aufgabe erledigt. Sie startet mit einem Aufruf der Funktion `createNode` und erzeugt somit einen neuen Knoten in der aktuellen Node-RED-Instanz. Als Parameter übergibt sie dabei eine Referenz auf das aktuelle Objekt (`this`) und die Konfiguration des Knotens. Diese Konfiguration enthält alle Daten, die dem Knoten zum Beispiel über den Konfigurationsdialog hinzugefügt wurden.

Nach diesem Aufruf ist das aktuelle Objekt ein vollwertiger Node-RED-Knoten und wird der Konstanten `node` zugewiesen. Die `on`-Funktion bringt dem Knoten dann bei, wie er sich im Falle einer eingehenden Nachricht zu verhalten hat. Das ist klassisches JavaScript, denn hier wird ein Ereignis (`input`) mit einer Callback-Funktion verknüpft, die immer dann aufgerufen wird, wenn das Ereignis eintritt. Als Parameter erhält die Callback-Funktion die Nachricht, die der Knoten empfangen hat.

Die folgenden Anweisungen greifen dann auf die COVID-19-API zu und werten deren Antwort aus. Den Anfang macht die Funktion `fetch`, die das aktuelle Summary-Dokument per HTTP abfragt. Der Rückgabewert von `fetch` ist nicht das Dokument selbst, sondern ein `Promise`-Objekt. Das ist in modernem JavaScript-Code ein beliebtes Muster, denn es verhindert allzu tief verschachtelte Auswertungen von Callback-Funktionen.

Dazu bieten `Promise`-Objekte die Methoden `then` und `catch` an. Diese werden jeweils im Erfolgs- beziehungsweise im Fehlerfall auf-

gerufen. Wenn also der Aufruf von `fetch` geklappt hat, wird anschließend die nachfolgende `then`-Methode aufgerufen und bekommt als Argument das zur HTTP-Antwort gehörende `Response`-Objekt.

Dieses Verhalten zieht sich im weiteren Verlauf durch. Die `json`-Methode wandelt zum Beispiel die JSON-Nutzlast des Objekts `response` in ein JavaScript-Objekt um. Wenn diese Umwandlung erfolgreich war, geht es mit der nächsten `then`-Methode weiter und die bekommt das JavaScript-Objekt übergeben.

Das Objekt ist ein Array von `Summary`-Objekten und nun gilt es, das richtige zu finden. Die Nachricht, die dem Knoten im Falle eines `input`-Ereignisses übermittelt wird, hat die Eigenschaft `msg.payload`. Sie enthält den Namen des Landes, dessen neue Fallzahlen zu ermitteln sind. Die `find`-Methode prüft daher, ob es ein `Summary`-Objekt gibt, dessen `Country`-Eigenschaft mit dem gesuchten Land übereinstimmt.

Besonderes Augenmerk gebührt der Anweisung `return summary`, denn sie führt nicht etwa zum Verlassen der Funktion `Covid19ApiNode`. Sie ist vielmehr Bestandteil der anonymen Funktion, die an die aktuelle `then`-Methode übergeben wurde und so sorgt sie dafür, dass der Inhalt der Konstanten `summary` als Argument an die nachfolgende `then`-Methode übergeben wird.

Der letzte `then`-Aufruf prüft, ob überhaupt ein passendes `Summary`-Objekt gefunden werden konnte und überschreibt dann den Inhalt der eingegangenen Nachricht entweder mit der Anzahl der neuen Corona-Infektionen oder mit dem Wert `-1`. Schließlich sendet die Funktion `send` die Nachricht an den Folge-Knoten.

Die `catch`-Funktion kommt zum Zuge, wenn in irgendeinem der `then`-Zweige ein Fehler auftritt. Fehler werden momentan nur auf der Konsole vermerkt.

Ein bisschen Farbe noch

Die beiden bisherigen Dateien beschreiben nur die Meta-Informationen des neuen Knoten-Typs und die Programmlogik, die ausgeführt wird, sobald der Knoten eine Nachricht enthält. Sie legen nicht fest, wie der Knoten innerhalb des Node-RED-Editors aussieht. Darum kümmert sich die Datei `covid19api.html`.

Im Grunde ist die Datei eine Mogelpackung. Es steht zwar HTML drauf, aber es steckt auch einiges an JavaScript drin. Im Wesentlichen enthält die Datei drei `<script>`-Blöcke, die alle unterschiedliche Aufgaben haben. Der erste enthält einen Aufruf der Funktion `registerType`, die in diesem Fall den Knoten beim Node-RED-Editor registriert. Dazu bekommt die Funktion den Namen des Knotens übergeben und ein Objekt mit allerlei Attributen, die den Knoten genauer beschreiben.

Das Attribut `category` legt fest, in welcher Knoten-Gruppe der Knoten auf der linken Seite des Editors erscheint. In diesem Fall ist es die Gruppe namens `function`. Die Eigenschaft `color` legt die Hintergrund-Farbe des Knotens fest und `icon` das Icon, das auf dem Knoten zu sehen ist. Es gibt eine Reihe von Standard-Icons, aber hier kann man bei Bedarf auch eigene Grafik-Dateien angeben. Bevorzugt wird dabei das skalierbare Vektor-Format SVG

covid19api.html

```
01 <script type="text/javascript">
02   RED.nodes.registerType('covid19api', {
03     category: 'function',
04     color: '#a6bbcf',
05     icon: 'white-globe.png',
06     defaults: {
07       name: { value: '' }
08     },
09     inputs: 1,
10     outputs: 1,
11     label: function() {
12       return this.name || 'covid19api';
13     }
14   });
15 </script>
16
17 <script type="text/html" data-template-name="covid19api">
18   <div class="form-row">
19     <label for="node-input-name"><i class="fa fa-tag"></i> Name</label>
20     <input type="text" id="node-input-name" placeholder="Name">
21   </div>
22 </script>
23
24 <script type="text/html" data-help-name="covid19api">
25   <p>
26     This node returns the number of Covid-19 cases that have
27     been confirmed yesterday for a certain country.
28   </p>
29 </script>
```

und auch wenn bei den Standard-Icons noch die Extension .png verwendet wird, lädt Node-RED automatisch passenden SVG-Ersatz.

Sehr wichtig ist der Abschnitt `defaults`, der Vorbelegungen für alle Eigenschaften des Knotens enthält. In diesem Fall ist das nur der Name des Knotens und der ist per Voreinstellung leer. Wenn der Knoten mehr Eigenschaften hat, sollten diese hier unbedingt vorbelegt werden.

Knoten können mehrere Ein- und Ausgänge haben und die Attribute `inputs` und `outputs`

legen deren Anzahl fest. Der *covid19api*-Knoten empfängt Nachrichten über genau einen Eingang und sendet Nachrichten über genau einen Ausgang.

Schließlich beschreibt die Eigenschaft `label` noch, welches Etikett der Knoten trägt, wenn er Bestandteil eines Flows ist. Wurde kein Name explizit angegeben, trägt er die Zeichenkette `covid19api`, sonst den angegebenen Namen. Die `label`-Eigenschaft wird daher mittels einer Funktion

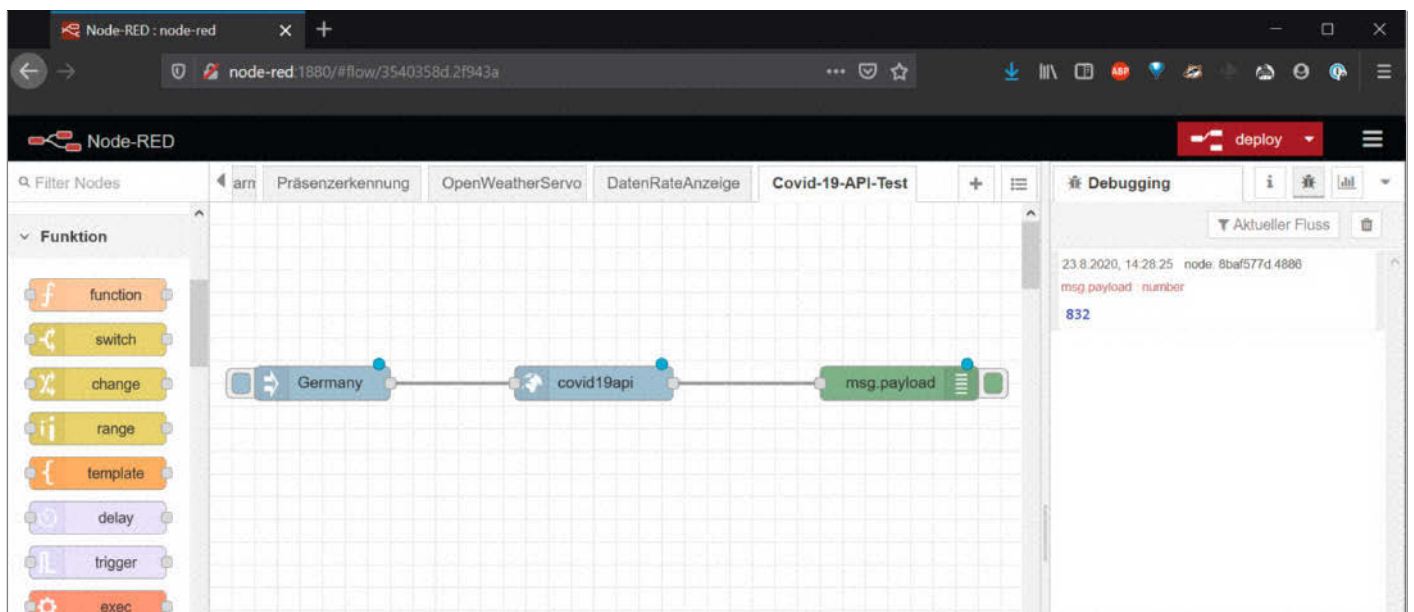
umgesetzt und nicht über ein statisches Literal.

Offen bleibt bisher die Frage, wo eine Eigenschaft wie der Name des Knotens eigentlich herkommt? Die klärt sich schnell bei einem Blick in den zweiten `<script>`-Block. Der definiert nämlich den Konfigurationsdialog des neuen Knotens, indem er das Attribut `data-template-name` auf den Namen `covid19api` setzt.

Innerhalb des `<script>`-Elements gibt es dann endlich auch reguläres HTML und zwar ein Text-Eingabe-Element für den Namen des Knotens. Node-RED stellt diverse CSS-Klassen zur Formatierung der Elemente bereit und *form-row* dient der einheitlichen Darstellung der einzelnen Formular-Elemente. Wichtiger sind aber die IDs der `<input>`-Tags, denn die müssen mit dem Präfix `node-input-` beginnen. Alles, was darauf folgt, definiert den Namen des Konfigurationselements. In diesem Fall ist das die Zeichenkette `name` und die legt den Namen des Attributs im Knoten fest. Aufgrund dieser Konvention ist es möglich, dass der JavaScript-Code zu Anfang der Datei auf das Attribut mit dem Namen `name` zugreifen kann. Der Code in der Datei *covid19api.js* kann auf diese Attribute über den `config`-Parameter zugreifen.

Node-RED integriert automatisch die Font Awesome-Bibliothek, die eine große Anzahl nützlicher Icons bereitstellt. Diese stehen in Form von CSS-Klassen zur Verfügung und in diesem Fall dient die Klasse *fa-tag* zur Darstellung eines Etiketten-Icons vor dem Eingabefeld des Knotennamens.

Das letzte `<script>`-Element setzt das Attribut `data-help-name` auf den Wert `covid19api` und wird dadurch zum Hilfe-Text des Knotens. Der gesamte Inhalt dieses Elements



Ein erster erfolgreicher Testlauf

wird als Hilfe-Text zum Knoten im Node-RED-Editor angezeigt und wer eine eigene Knoten-Bibliothek veröffentlicht, tut gut daran, möglichst hilfreiche Dokumentation mitzuliefern.

Entwicklungshilfe

Als nächstes gilt es, den neuen Knoten endlich auszuprobieren und der dazu gehörende Prozess unterscheidet sich kaum von der Installation anderer Knoten. Zunächst einmal müssen alle Dateien, die zum neuen Knoten gehören, in einem gemeinsamen Verzeichnis liegen, das den Namen des Pakets hat. In diesem Fall ist das der Name `node-red-contrib-covid19api` und die weiteren Schritte setzen voraus, dass dieses Verzeichnis im Home-Verzeichnis des Nutzers liegt.

Um das Paket zu installieren, wechselt man in das Heimverzeichnis des Nutzers, unter dem Node-RED läuft und von dort in das Verzeichnis `.node-red`. Auf einem Raspberry Pi ist das in der Regel das Verzeichnis `/home/pi`, so dass das Kommando `cd ~/.node-red` in den entsprechenden Ordner wechselt.

Hier findet sich das `node_modules`-Verzeichnis, in dem alle Node.js-Module liegen, auf die Node-RED zugreifen kann. Hierhin wird das Paket für den neuen Knoten mit dem Kommando

```
npm install ~/.node-red-contrib-covid19api
```

installiert. Die Installation erzeugt dabei einen Link auf das Original-Verzeichnis und installiert gleichzeitig alle Abhängigkeiten.

Node-RED lädt neue Module nicht automatisch und muss daher neu gestartet werden. Auf einem Raspberry Pi erledigt das Kommando

```
sudo service nodered restart
```

den Job. Anschließend erscheint der `covid19-api`-Knoten auf der linken Seite in der Funktion-Gruppe.

Wer Node-RED bereits produktiv einsetzt, um zum Beispiel das eigene Heim zu automatisieren, sollte die Installation einer weiteren Node-RED-Instanz zu Entwicklungszwecken erwägen. So werden laufende Flows beim ständigen Durchstarten nicht beeinträchtigt.

Laufen lassen

Ein erster Test-Flow ist mit Node-RED schnell gebaut und benötigt gerade einmal drei Knoten. Ausgangspunkt ist ein `Inject`-Knoten, dessen Nutzdaten die Zeichenkette `Germany` enthalten. Dazu muss der Nutzdaten-Typ mit der Dropdown-Box im Konfigurationsdialog entsprechend angepasst werden.

Der `Inject`-Knoten wird anschließend mit einem `covid19api`-Knoten verbunden, der bei Bedarf im Konfigurationsdialog einen neuen

Namen bekommen kann. Schließlich wird der `covid19api`-Knoten mit einem `Debug`-Knoten verbunden, der alle eingehenden Nachrichten ins `Debug`-Fenster auf der rechten Seite schreibt. Um die Nachrichten zu sehen, muss man das kleine Käfer-Icon rechts oben anklicken.

Um den Flow zu aktivieren, muss man ihn veröffentlichen. Das geht mit dem roten Deploy-Knopf rechts oben. Sobald der Flow aktiv ist, reicht ein Klick auf den kleinen Knopf ganz links am `Inject`-Knoten, um die aktuellen Corona-Infektionen für Deutschland zu ermitteln und im Debug-Fenster auszugeben. Übrigens muss der Flow nach jeder Änderung neu veröffentlicht werden.

Wurde bei der Ausführung die Zahl -1 zurückgegeben, konnte in der Regel das gewünschte Land nicht gefunden werden. Alle anderen Fehler schreibt Node-RED in die Datei `/var/log/syslog`.

Der eigene Node-RED-Knoten läuft und kann ab sofort in beliebigen Flows zum Einsatz kommen. Möglich ist beispielsweise die Erstellung eines Dashboards, das die Informationen in einem Web-Browser darstellt. Alternativ kann ein Flow die Informationen per MQTT an einen ESP8266 oder einen ESP32 senden. Der kann sie dann in Form einer LED-Ampel, mithilfe eines Displays oder auf viele andere Arten visualisieren. Wie man LED-Streifen zur Visualisierung benutzt, erklärt auch unser Node-RED-Sonderheft (siehe Kasten).

Fazit

Wie so oft im Leben ist aller Anfang schwer. Sobald man aber eine funktionierende Version einer Node-RED-Bibliothek hat, ist es recht einfach, sie schrittweise zu erweitern. Zum Beispiel ist es kein großes Problem, den Covid19-Knoten um eine Drop-Down-Box mit den unterstützten Ländern zu erweitern. Diese Liste könnte der Knoten sogar bei der COVID-19-API abfragen.

Verbesserungspotenzial gibt es auch auf technischer Ebene, denn der Knoten muss nicht jedes Mal das ganze Summary-Dokument laden, sondern könnte es durchaus zwischenspeichern und nur einmal täglich laden. Solche Mechanismen sieht Node-RED vor und sie sind gut dokumentiert.

Dokumentation ist ein gutes Stichwort, denn die ist reichlich vorhanden. Ferner lohnt sich das Studium bestehender Erweiterungen, denn die meisten sind naturgemäß nicht allzu umfangreich.

Als größte Hürde erweist sich letzten Endes die Programmiersprache JavaScript, denn die muss man halbwegs beherrschen, um seine eigenen Node-RED-Ideen umsetzen zu können. Allzu schwierig ist die Sprache aber nicht und sie eignet sich gut für Einsteiger. —dab

NEU

Auch
Heft + PDF
erhältlich mit
29 % Rabatt

Raspberry Pi- Projekte zum Basteln, Steuern, Vernetzen



NEU

c't RASPI 2020

In diesem Sonderheft hat die c't-Redaktion die besten Artikel rund um den Raspi aus dem vergangenen c't-Jahrgang zusammengetragen und überarbeitet. Damit verschaffen Sie sich zusätzliche Sicherheit in Ihrem Netzwerk, setzen den Kleincomputer als Multimedia-Server oder Netzwerkspeicher ein oder bauen sich damit einen Internet-Radiowecker oder eine Smarthome-Zentrale und vieles mehr.

shop.heise.de/ct-raspi20

Einzelheft
für nur

14,90 € >

Generell portofreie Lieferung für Heise Medien- oder Maker Media Zeitschriften-Abonnenten. Nur solange der Vorrat reicht. Preisänderungen vorbehalten.

 **heise shop**

shop.heise.de/ct-raspi20

Hausgeräte-Überwachung mit Fritzbox und Telegram

Ist der Haussegen aus dem Gleichgewicht, weil Sie wieder einmal vergessen haben, Ihren Anteil an der Hausarbeit durch Leerung von Waschmaschine oder Trockner zu leisten? Mit einem Raspberry Pi, ein paar smarten Steckdosen und etwas Software zeige ich Ihnen, wie Sie das nie mehr vergessen.

von Stefan Wollner



Egal ob Sie zur Miete, in einer Eigentumswohnung oder einem Einfamilienhaus wohnen, Haushaltsgeräte wie Waschmaschinen oder Trockner befinden sich oft außerhalb der Wohnung in Zwischenetagen, Gemeinschaftskellern oder Waschküchen. Und selbst wenn nicht, treten dennoch die gleichen Probleme auf: Man vergisst die Wäsche aus der Maschine zu nehmen und aufzuhängen, die Spülmaschine auszuräumen oder man wartet unnötigerweise vor dem Bullauge der Waschmaschine, bis diese den Magnetverschluss freigegeben hat.

Um dem aus dem Weg zu gehen, suchte ich eine unauffällige Lösung, ohne Hand an die Gebäudeverkabelung zu legen, in die Elektronik der Geräte einzugreifen oder wie beim *Wäsche-Fertig-Melder* aus der Make 3/20, Sensoren an die Geräte kleben und lästige Kabelstrippen zum Kontroll-Baustein ziehen zu müssen. Meine Wahl fiel auf die smarten, per DECT kommunizierenden Steckdosen der Firma AVM ¹ vom Modell DECT200 in Verbindung mit einem Raspberry Pi und meiner Fritzbox. Drei dieser Steckdosen hatte ich noch aus einem vorherigen Projekt übrig.

Der Raspberry Pi fragt über die *AVM Home Automation API* den aktuellen Stromverbrauch der über die Steckdosen angeschlossenen Geräte bei der Fritzbox ab. Werden festgelegte Schwellenwerte über- beziehungsweise unterschritten, versendet er im Namen der jeweiligen Steckdose eine Nachricht in eine Telegram-Gruppe ². Die Software des Projektes ist in Python geschrieben und nutzt eine Bibliothek, um die Kommunikation mit der Fritzbox durchzuführen. Um den Programmcode so einfach wie möglich zu halten, verwende ich für den Versand der Telegram-Nachrichten bewusst keine der vorhandenen Bibliotheken.

Anmeldung DECT-Steckdosen

Um das Projekt nachzubauen, müssen Sie als erstes die Steckdosen an der Fritzbox anmelden. Wie dies durchgeführt wird, kann den beiden Steckdosen-Schnellanleitungen entnommen werden, die Sie über den Link in der Kurzinfo erreichen.

Nachdem die Steckdosen angemeldet wurden, können Sie ihnen auf der Weboberfläche einen Namen geben. Bevor Sie die Steckdosen hinter Wasch- oder Spülmaschine verbauen, notieren Sie sich deren **AIN** (**A**ktor **I**dentifikations**n**ummer). Diese eindeutige Bezeichnung benötigen Sie später für die Abfrage. Sie finden diese auf der Rückseite der Steckdosen und in der Weboberfläche der Fritzbox ³.

Anmeldeverfahren Fritzbox

Prinzipiell ist es möglich, sich ausschließlich per Kennwort an der Fritzbox anzumelden. Dennoch empfehle ich aus Sicherheitsgrün-

Kurzinfo

- » Feststellen des Geräte-Status durch Auswertung des aktuellen Stromverbrauchs
- » Erstellung Telegram-Bots und -Gruppe
- » Versand von Telegram-Nachrichten anhand von Schwellenwerten

Checkliste



Zeitaufwand:

60 Minuten für die Installation



Kosten:

ab etwa 25 Euro für den Raspberry zzgl. 42 Euro pro überwachter Steckdose



Programmieren:

Grundkenntnisse Raspberry Pi



Computer:

Desktop oder Notebook mit Windows, macOS oder Linux

Material

- » Raspberry Pi (ab Zero)
- » SD-Speicherkarte passend zum Raspberry
- » Netzteil passend zum Raspberry
- » pro überwachtem Gerät 1 Steckdose AVM DECT 200 oder AVM DECT 210
- » AVM Fritzbox

Mehr zum Thema

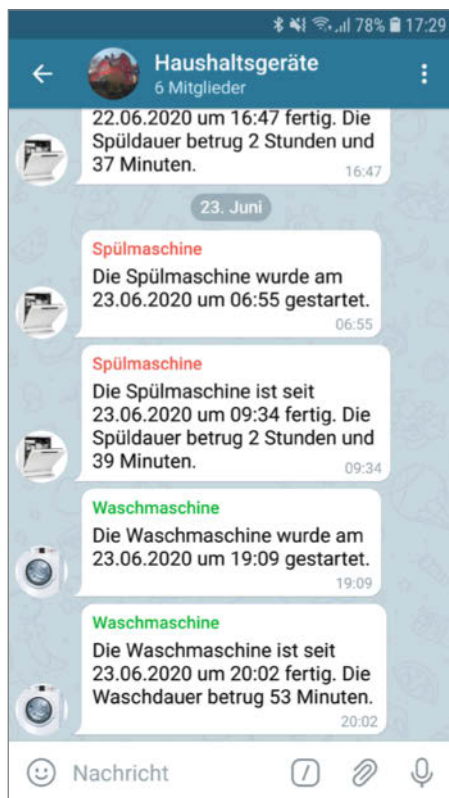
- » Wie Sie Hausgeräte durch Abfragen ihrer LED-Anzeigen überwachen, lesen Sie im Artikel „Wäsche-Fertig-Melder“ im Make-Magazin 3/20 ab Seite 18
- » Infos zur Einrichtung eines Telegram-Bots finden Sie im den Artikel „Digitaler Bilderrahmen mit Messenger-Funktion“ aus der Make 5/19 ab Seite 32
- » und in „Smarter Bewegungsmelder“ aus der Make 4/19 ab Seite 66
- » Wie man Steckdosen anderer Hersteller mit neuer Firmware flasht, erfahren Sie in „Smarte Steckdosen mit neuer Firmware“ im Make Sonderheft 2018 ab Seite 80



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



- ¹ Die beiden DECT-Steckdosen-Modelle von AVM kosten jeweils etwa 42 Euro: Links die DECT 200 (nur für innen, maximal 2300W/10A) und rechts die DECT 210 (spritzwassergeschützt nach IP44, maximal 3450W/15A).



2 So sehen Nachrichten in der Gruppe *Haushaltsgeräte* aus.

den, zwei Benutzerkonten anzulegen. Das erste erhält dabei Administrationsrechte. Für das zweite Konto werden nur Berechtigungen für den Bereich *Smart-Home* eingerichtet. Diese Zugangsdaten werden später für die Software auf dem Raspberry Pi benötigt, damit der über die Fritzbox auf die Steckdosen zugreifen kann.

Zunächst melden Sie sich wie gewohnt an Ihrer Fritz!Box an und wechseln in der Benutzeroberfläche auf *System* und *Fritz!Box Benut-*

zer. Dort betätigen Sie den Button *Benutzer hinzufügen*. Anschließend geben Sie einen Benutzernamen sowie ein Kennwort ein. Das Feld der E-Mailadresse kann freigelassen werden. Bevor Sie das Benutzerkonto mit dem Button *Übernehmen* endgültig anlegen, wählen Sie noch die zum jeweiligen Benutzer passenden Berechtigungen aus 4.

Nachdem beide Benutzerkonten angelegt wurden, wechseln Sie auf *Anmeldung im Heimnetz*. Dort aktivieren Sie *Anmeldung mit Fritz!Box-Benutzernamen und Kennwort* und speichern die Eingabe mit einem Klick auf *Übernehmen*. Sie werden nun von Ihrer Fritzbox abgemeldet und können sich mit einem der beiden neu angelegten Benutzerkonten erneut anmelden.

Telegram-Bots und -Gruppe

Als nächstes sollten Sie sich um die Einrichtung der notwendigen Telegram-Bots kümmern. Diese versenden später unter Ihren eigenen Namen Nachrichten der Steckdosen in eine Telegram-Gruppe. Während der späteren Installation der Software werden die jeweiligen Tokens der Bots sowie die Gruppen-ID benötigt.

Die Erstellung eines Telegram-Bots wurde bereits in den Artikeln „Digitaler Bilderrahmen mit Messenger-Funktion“ (den Link zu diesem Artikel finden Sie in der Kurzinfor) ausführlich beschrieben. Daher werde ich nur auf die Besonderheiten zu meinem Projekt eingehen. In meinem Github-Repository (siehe Kurzinfor-Link) finden Sie Bilder der Erstellung eines Telegram-Bots.

Nachdem für jede Steckdose ein eigener Bot erstellt wurde, ist es an der Zeit, eine Telegram-Gruppe zu gründen. Hierzu klicken Sie in der Handy-App auf die drei Striche und wählen den Punkt *Neue Gruppe* aus. In dem sich öffnenden Dialog können Sie mit *@Bot_*

Name Ihre Bots suchen und der Liste hinzufügen. Mit einem Klick auf das Pfeilsymbol wechselt man in das nächste Fenster. In diesem vergeben Sie den Gruppennamen. Weiterhin besteht die Möglichkeit, ein Bild hinzuzufügen bevor Sie die Gruppe mit einem Klick auf das Hakensymbol erstellen 5.

Sie haben sicherlich bemerkt, dass für die Aufnahme Ihrer Bots in die Telegram-Gruppe weder ein Benutzername noch ein Passwort benötigt wurde. Grundsätzlich besteht für jeden Telegram-Nutzer die Möglichkeit, einen oder mehrere Ihrer Bots in seine eigene Gruppe aufzunehmen. Sie können diese Voreinstellung jedoch ändern, indem Sie nochmals in den Chat mit dem *Botfather* wechseln und das Kommando */setjoingroups* eingeben. In dem folgenden Dialog wählen Sie für jeden Ihrer Bots den Status *DISABLED* aus. Zuletzt kann mit dem Kommando */setuserpic* jedem Bot ein individuelles Bild zugeordnet werden. In meinem Github-Repository finden Sie auch hier weitere Bilder.

Zur Ermittlung der Gruppen-ID wechseln Sie in den Chat der erstellten Gruppe. Mit einem Klick auf den Gruppennamen wird eine Übersicht aller Mitglieder angezeigt. Mit dem Button *Mitglied hinzufügen* öffnet sich ein Suchdialog. In diesem geben Sie *@getidsbot* ein und fügen das neue Mitglied mit einem Klick auf das Hakensymbol Ihrer Gruppe hinzu. Zurück in der Übersicht, wird das neu hinzugefügte Mitglied *GetIDs Bot* mit einem langen Klick auf dessen Namen direkt wieder entfernt. Im Gruppenchat finden Sie eine Nachricht mit vielen Informationen. In dieser wird auch die Gruppen-ID angezeigt 6.

Testnachricht

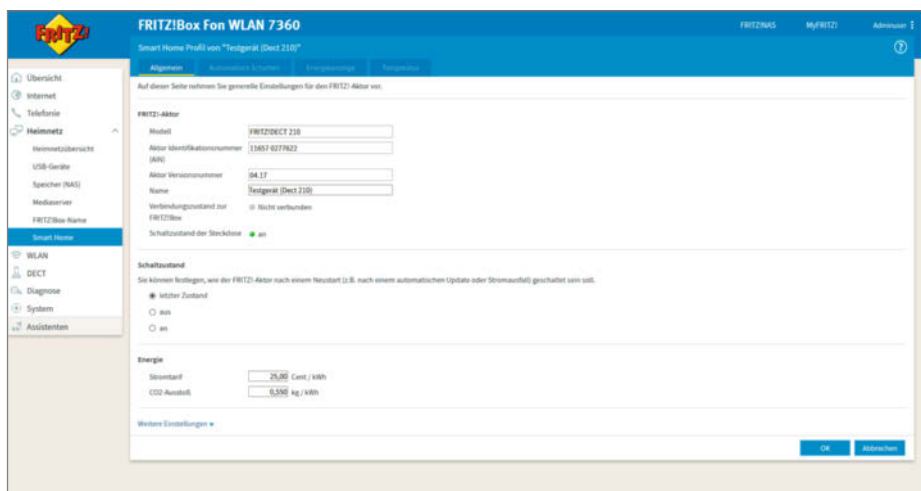
Nachdem alle Vorarbeiten abgeschlossen wurden, ist es an der Zeit, den Nachrichtenversand zu testen. Hierzu geben Sie in einem Browser Ihrer Wahl das folgende Kommando ein 7.

Den *BOT-TOKEN* sowie die *GRUPPEN-ID* ersetzen Sie mit Ihren eigenen Angaben. Nach circa einer Sekunde sollte die Nachricht *Testnachricht* in Ihrem Gruppenchat erscheinen 8.

Vorbereitung der SD-Karte

Für die Installation des Projektes empfehle ich ein *Raspberry Pi OS Lite*. Sie können allerdings auch eine Version mit Desktopumgebung einsetzen. Eine Installationsanleitung sowie den Download des Images finden Sie unter dem Link in der Kurzinfor.

Nach der Installation muss man eine leere Datei mit dem Namen *ssh* auf der *BOOT*-Partition der SD-Karte erstellen. Diese aktiviert den SSH-Server, über den Sie sich nach dem Start des Raspberry Pis mit einem Programm wie *putty* verbinden.



3 Das Smart Home Profil einer angemeldeten AVM-Steckdose mit dem Namen *Testgerät* und der AIN 11657 0277622

Installation Projekt

Die Installationsdateien befinden sich in einem Github-Repository. Damit Sie mit Github kommunizieren können, müssen zuerst die folgenden Befehle ausgeführt werden:

```
sudo apt update
sudo apt -y install git
```

Nun können Sie das Repository mit den folgenden Befehlen auf Ihren Raspberry Pi klonen ⁹.

Um die Installation zu starten, geben Sie die folgenden Kommandos ein:

```
chmod +x install.sh
./install.sh
```

Während der Installation werden Sie unter anderem dazu aufgefordert, das Login-Passwort des Raspberry Pi, die Bot-Tokens sowie die Gruppen-ID einzugeben. Zum Abschluss der Installation muss der Raspberry Pi mit dem folgenden Befehl neu gestartet werden:

```
sudo reboot
```

Danach verbinden Sie sich erneut mit ihm und wechseln mit dem folgenden Kommando in das Verzeichnis des Haushaltsprogramms:

```
cd /opt/haushalt
```

Schwellenwerte

Für den späteren Nachrichtenversand werden für jede Steckdose zwei Schwellenwerte benötigt. Dabei handelt es um einen Start- bzw. einen Stopp-Wert in Milliwatt. Zur Verdeutlichung ein kurzes Beispiel.

Liegt der Start-Schwellenwert bei 5.000mW, wird eine Nachricht mit dem Inhalt „Das Gerät <GERÄTENAME> wurde gestartet“ versendet, sobald das Gerät diesen Stromverbrauch überschreitet. Liegt der Stopp-Schwellenwert bei 2.500mW, wird eine Nachricht mit dem Inhalt „Der Vorgang des Gerätes <GERÄTENAME> ist fertig“ versendet, sobald das Gerät diesen Stromverbrauch unterschreitet.

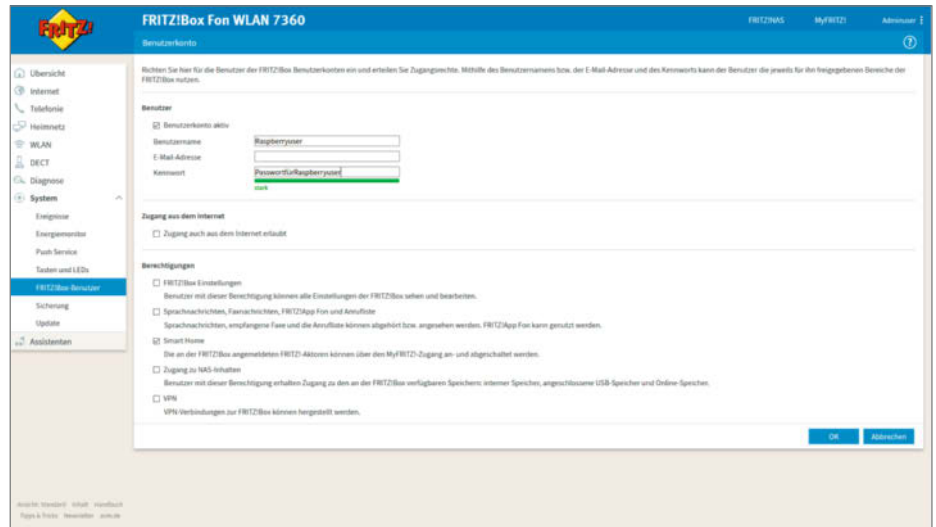
Um die Schwellenwerte zu ermitteln, können Sie das Programm *aufzeichnung.py* verwenden. Dabei gehen Sie nach der folgenden Methode vor:

Start Aufzeichnung -> Start Maschine -> Ende Maschine -> Ende Aufzeichnung

Das Programm starten Sie mit dem folgenden Kommando:

```
/opt/haushalt/aufzeichnung.py
```

Die Aufzeichnung beginnt, sobald die aufzeichnende Steckdose (Gerät) ausgewählt wurde. Mit der Tastenkombination *Strg + C* kann die Aufzeichnung beendet werden. Im Terminal wird ein Pfad zu einer Textdatei ausgegeben.



4 Bei der Neuanlage eines Benutzerkontos bitte die Checkbox **Smart Home** auswählen.

Darin befinden sich die aufgezeichneten Werte. Sie können die Textdatei mit dem folgenden Befehl ausgeben, wobei <PFAD> mit dem vorher angezeigten Pfad ersetzt werden muss:

```
cat <PFAD>
```

Jede angezeigte Zeile entspricht dabei einer Messung und ist wie in der Tabelle ¹⁰ aufgebaut:

Sie müssen nun für jedes Gerät herausfinden, ab welchem Stromverbrauch die beiden Nachrichten versendet werden sollen. In meinem Repository stelle ich die vollständigen Beispielaufzeichnungen meiner Wasch- und Spülmaschine sowie meines 3D-Druckers inklusive einer Erklärung der von mir festgelegten Schwellenwerte zur Verfügung.

Konfigurationsdateien

Nachdem für alle Geräte die Schwellenwerte ermittelt wurden, müssen Sie diese in die Konfigurationsdatei *schwellenwerte.json* eintragen. Alle für das Projekt notwendigen Konfigurationsdateien befinden sich im Verzeichnis:

```
/home/pi/.credentials
```

Das Eintragen kann per Hand erfolgen und muss wie in ¹¹ aussehen.

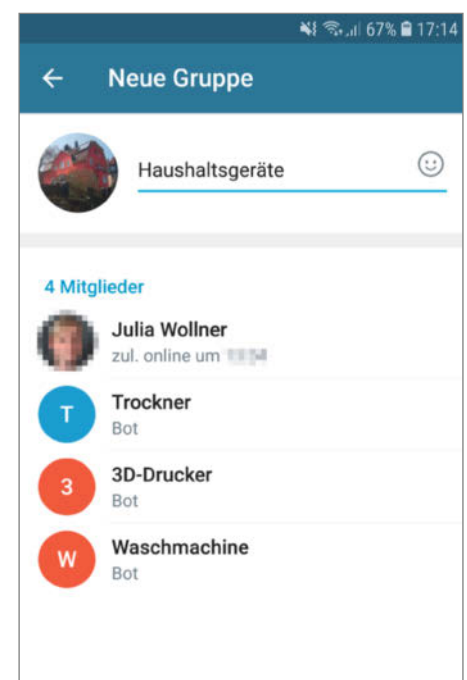
Oder Sie nutzen das von mir geschriebene Programm, um die Schwellenwerte maschinell in die entsprechende Konfigurationsdatei einzutragen. Der Befehl dazu lautet:

```
/opt/haushalt/schwellenwerte.py
```

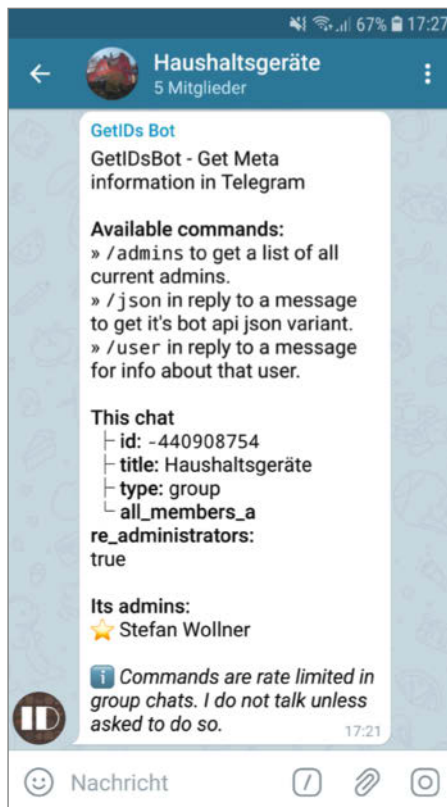
Unabhängig für welche Variante Sie sich entscheiden, wichtig ist die Einhaltung der Notation. In den Dateien können Sie Änderungen an den Bot-Tokens, den Anmeldedaten der Fritzbox, den Schwellenwerten, den Gerätenamen sowie den AIN-Nummern der Steckdosen durchführen.

Programm haushalt.py

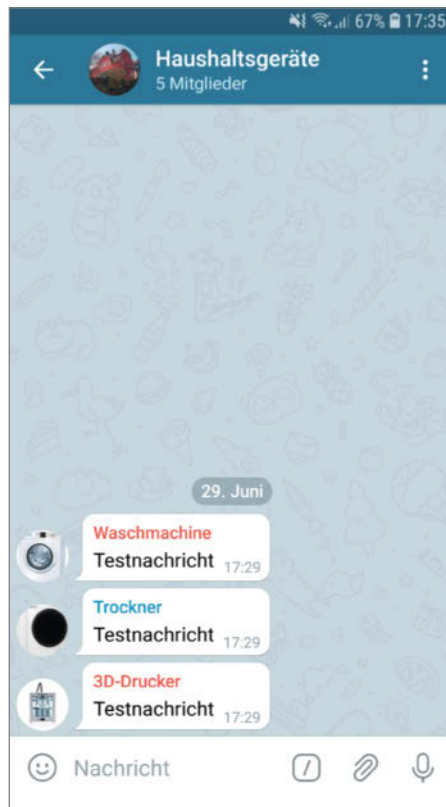
Mit dem vorangegangenen Schritt haben Sie die letzten Vorarbeiten abgeschlossen. Dem eigentlichen Hauptprogramm *haushalt.py* liegen nun alle notwendigen Informationen für einen reibungslosen Betrieb vor. Dieses Programm kümmert sich um die Kommunikation mit der Fritzbox und den Versand der Telegram-Nachrichten. Weiterhin ist es für den Informationsaustausch mit der lokal angelegten Datenbank zuständig. Darin werden entsprechende Datensätze für jeden Vorgang angelegt. Damit ist es dem Programm möglich, auch nach einem Neustart zu erkennen,



5 Die neue Telegram-Gruppe **Haushaltsgeräte** mit den erstellten Bots und weiteren Personen



6 Die Chat-ID wird für die spätere Installation benötigt.



8 Das Ergebnis der versendeten Testnachrichten für die angelegten Telegram-Bots

10 Messwerte		
ID	mW	Timestamp
4	0	2019-11-09 09:26:15
5	0	2019-11-09 09:26:35
6	2570	2019-11-09 09:26:56
7	2570	2019-11-09 09:27:17
8	2570	2019-11-09 09:27:37
9	15020	2019-11-09 09:27:58
10	15020	2019-11-09 09:28:19
11	15020	2019-11-09 09:28:39
12	15020	2019-11-09 09:29:00
128	182740	2019-11-09 10:08:57
129	182740	2019-11-09 10:09:18
130	168440	2019-11-09 10:09:38
131	168440	2019-11-09 10:09:59
132	168440	2019-11-09 10:10:20
321	19740	2019-11-09 11:15:24
322	2430	2019-11-09 11:15:44
323	2430	2019-11-09 11:16:05
324	2430	2019-11-09 11:16:26
325	2430	2019-11-09 11:16:46
326	2430	2019-11-09 11:17:07
327	2430	2019-11-09 11:17:28
328	2430	2019-11-09 11:17:48
329	2430	2019-11-09 11:18:09
330	1930	2019-11-09 11:18:30
331	1930	2019-11-09 11:18:50

7 Testbefehl für den Telegram-Bot

```
https://api.telegram.org/bot<BOT-TOKEN>/sendMessage?chat_id=<GRUPPEN-ID>&text=Testnachricht
```

ob etwa die Waschmaschine noch läuft und wann der Waschvorgang gestartet wurde. Das Programm versendet deshalb keine erneute Start-Nachricht.

Die angelegten Datensätze können später auch zu Auswertungszwecken herangezogen werden. Wer eine solche Erweiterung für das Projekt programmiert, kann gerne einen Pull Request an das Github Repository senden. Insgesamt habe ich versucht, den Programmcode so einfach wie möglich zu halten, damit auch fortgeschrittene Anfänger Änderungen an diesem vornehmen können.

Systemd Service Unit

Das Programm *haushalt.py* wird über eine Systemd Service Unit gestartet. Diese ist mit einem Windows-Dienst vergleichbar. Den Aufbau der Unit *haushalt.service* sehen Sie im Listing 12. Weiterführende Informationen zum Thema finden Sie in der Kurzinfor.

Haben Sie Änderungen an den Konfigurationsdateien durchgeführt, muss der Dienst neu gestartet werden, um die neuen Werte zu übernehmen. Sie können hierzu diesen Befehl verwenden:

```
sudo systemctl restart
haushalt.service
```

Alternativ besteht die Möglichkeit, den Raspberry Pi vollständig neu zu starten.

Hausautomatisierung

Natürlich gibt es bereits viele Projekte, die sich mit dem Thema Hausautomatisierung befassen und sogar verschiedene Systeme und Protokolle miteinander verbinden. Hier sind insbesondere der *IO-Broker* sowie *FHEM* zu erwähnen (Links in der Kurzinfor).

In beiden Projekten besteht die Möglichkeit, das *AVM DECT Ökosystem* zu integrieren. Zusätzlich können auch Steckdosen anderer Hersteller mit einer alternativen Firmware wie etwa *Tasmota* geflasht und eingebunden werden. Falls diese Steckdosen ebenfalls über Stromsensoren verfügen (das ist nicht bei allen Smart-Home-Steckdosen der Fall), erfolgt die Übermittlung des aktuellen Strom-

9 Klonen des Repository

```
01 cd ~
02 git clone https://github.com/Blackbox-git/HousekeepingPi
03 cd HousekeepingPi
```

11 Eintrag in der Konfigurationsdatei

```
01 {
02 "AIN1start": 5000,
03 "AIN1stop": 2100,
04 "AIN2start": 2000,
05 "AIN2stop": 600,
06 "AIN3start": 160000,
07 "AIN3stop": 15000
08 }
```

12 Die Systemd Service Unit im Detail

```
01 [Unit]
02 Description=Steckdosenüberwachung AVM Fritz!Box
03 After=mariadb.service
04
05 [Service]
06 Type=simple
07 ExecStart=/opt/haushalt/haushalt.py
08 Restart=always
09
10 [Install]
11 WantedBy=multi-user.target
```

verbrauches dann meist mit dem Protokoll *MQTT* über WLAN. Der Artikel *Smarte Steckdosen mit neuer Firmware* beschreibt den Vorgang ausführlich. Auch die Open-Source-Software *Homeassistant*, eine leistungsfähige Smart-Home-Zentrale, kann mit kommerziellen Geräten wie der Fritzbox und deren Zubehör, aber auch mit Selbstbauadaptern auf Basis von ESP-Mikrocontrollern zusammenarbeiten. Falls Sie neugierig geworden sind, finden Sie über den Kurzinfo-Link mehr Infos dazu.

Ich habe mich aufgrund des hohen Installations- und Konfigurationsaufwandes allerdings bewusst gegen ein solch komplexes System entschieden. Zudem konnte ich mir durch die Entwicklung neue Fertigkeiten aneignen.

Fazit

Mein Projekt wird das Leben vieler Menschen nicht radikal verändern. Dennoch besitzt es das Potential, dieses zumindest in einem Teilbereich zu erleichtern. Vielleicht hilft es, Beziehungs-

problemen aus dem Weg zu gehen oder sorgt für ein harmonischeres Zusammenleben in einer WG oder einem Mehrgenerationenhaus.

In Zukunft kann ich mir eine Verbindung zu anderen Projekten vorstellen. Denkbar wäre, eine Nachricht an den Digitalen Bilderrahmen (Make 5/2019, S. 32) zu versenden, sobald etwa die Waschmaschine ihre Arbeit beendet hat. Dadurch kann sichergestellt werden, dass die Großeltern den beschwerlichen Weg in den Waschkeller nicht umsonst antreten müssen. —hgb

Einsteigen und durchstarten!



A. Sweigart

Routineaufgaben mit Python automatisieren

Praktische Programmierlösungen für Einsteiger

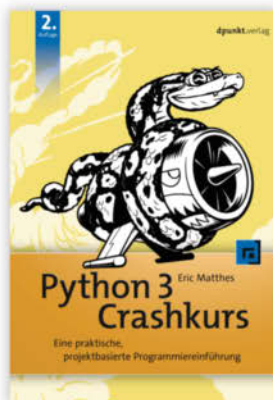
Aktualisiert auf Python 3

2. Auflage

2020, 696 Seiten

€ 34,90 (D)

ISBN 978-3-86490-753-1



E. Matthes

Python 3 Crashkurs

Eine praktische, projektbasierte Programmier Einführung

2. Auflage

2020, 620 Seiten

€ 32,90 (D)

ISBN 978-3-86490-735-7



M. Inden

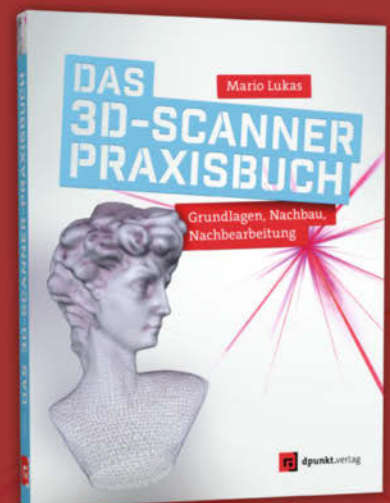
Java Challenge

Fit für das Jobinterview und die Praxis – mit mehr als 100 Aufgaben und Musterlösungen

2020, 600 Seiten

€ 34,90 (D)

ISBN 978-3-86490-756-2



M. Lukas

Das 3D-Scanner-Praxisbuch

Grundlagen, Nachbau, Nachbearbeitung

2020, 352 Seiten

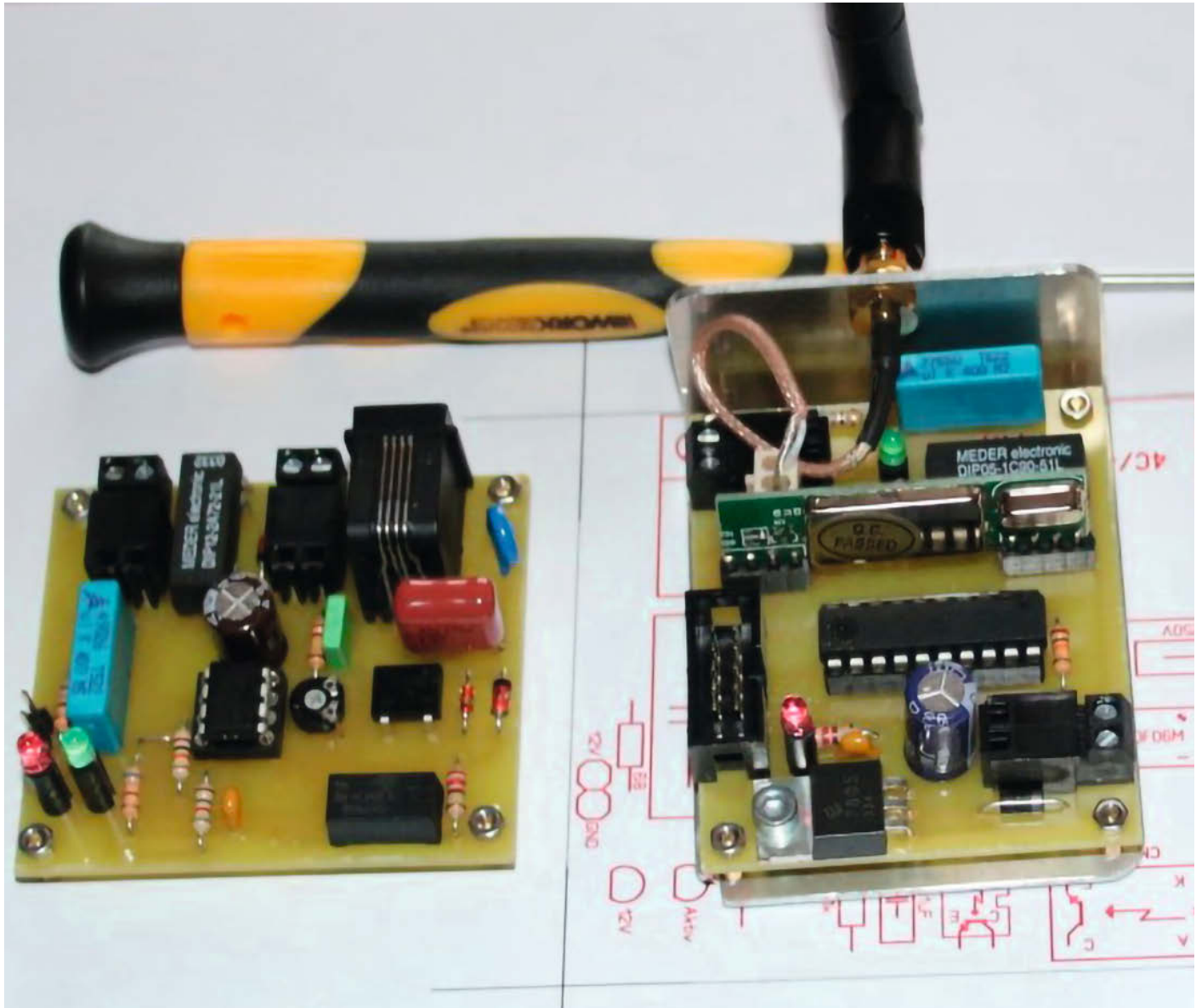
€ 34,90 (D)

ISBN 978-3-86490-640-4

Fritzbox öffnet die Tür

Lassen Sie das Telefon schon zusammen mit der Türklingel bimmeln? Schön, aber das ist ja nur eine halbe Lösung. Dieses Leserprojekt zeigt, wie man per DECT-Telefon und Fritzbox die Tür per Telefon auch öffnen kann. Interessant nicht nur, wenn Sie bettlägerig oder schlecht zu Fuß sind.

von Hans Borngräber



Seit der Fritzbox-Hersteller AVM das TR064-Management-Protokoll im FritzOS dokumentiert hat, gibt es einige Lösungen, die eine Umleitung des Haustür-Klingelsignales zu einem DECT-Telefon ermöglichen. Dadurch klingelt das Telefon, wenn jemand den Klingelknopf betätigt. In der Kurzinfo finden Sie unter "Mehr zum Thema" Anleitungen dazu.

Wenn Sie solch eine Schaltung benutzen, ist es schön und gut: Es klingelt an der Haustür und am Telefon. Aber wie bekommt man jetzt die Tür auf? Gerade für Behinderte oder ältere Personen wäre eine solche Fernschalt-Funktion über das Telefon sehr hilfreich, um zum Beispiel in Mehrparteienmietshäusern den Briefträger den Zugang zu den Postkästen im Hausflur zu ermöglichen oder bei Einfamilienhäusern die Gartentür zu öffnen. Mit dem TR064-Protokoll geht das jedoch nicht. Und da setzt mein Projekt ein.

Analog-Anschluss zweckentfremdet

Die Lösung sieht so aus: Jede Fritzbox besitzt auch zwei analoge, meist ungenutzte Telefonanschlüsse, die vom DECT-Anschluss aus per Kurzwahl angerufen werden können. Einen davon verwendet man nun zur Steuerung des elektrischen Türöffners an der Haustür. Der Anschluss steuert das Basis-Modul dieses Projekts, das, sobald es angerufen wird, über ein Relais den Türöffner betätigt. Am DECT-Telefon muss man dazu nur die zum Anschluss (FON1 oder FON2) gehörende Kurzwahl-Nummer **7 beziehungsweise **2 wählen. Das Modul wird mit dem Türöffner-Taster per Kabel verbunden. Optional bei größerer Entfernung geht das auch über eine Funkfernsteuerung (im 433MHz-Bereich). Die ist aber nur gegen versehentliche Bedienung durch einen der bei Funk-Steckdosen gebräuchlichen Handschal-

Kurzinfo

- » Türöffnung durch interne und externe Anrufe
- » Türöffner über Kabelverbindung betätigen
- » Funkfernsteuerung mit individuellem Code-Schlüssel

Checkliste


Zeitaufwand:
3 bis 4 Stunden (ohne Platinenherstellung)


Kosten:
circa 20 bis 60 Euro (ohne/mit Funkfernsteuerung)

Werkzeug

- » Lötkolben
- » Kleiner Schraubendreher
- » Nur bei Selbstherstellung der Platinen: Belichtungs- und Ätzanlage

Material

Die Bauteil-Listen stehen wegen ihrer Länge nur online zur Verfügung

Mehr zum Thema

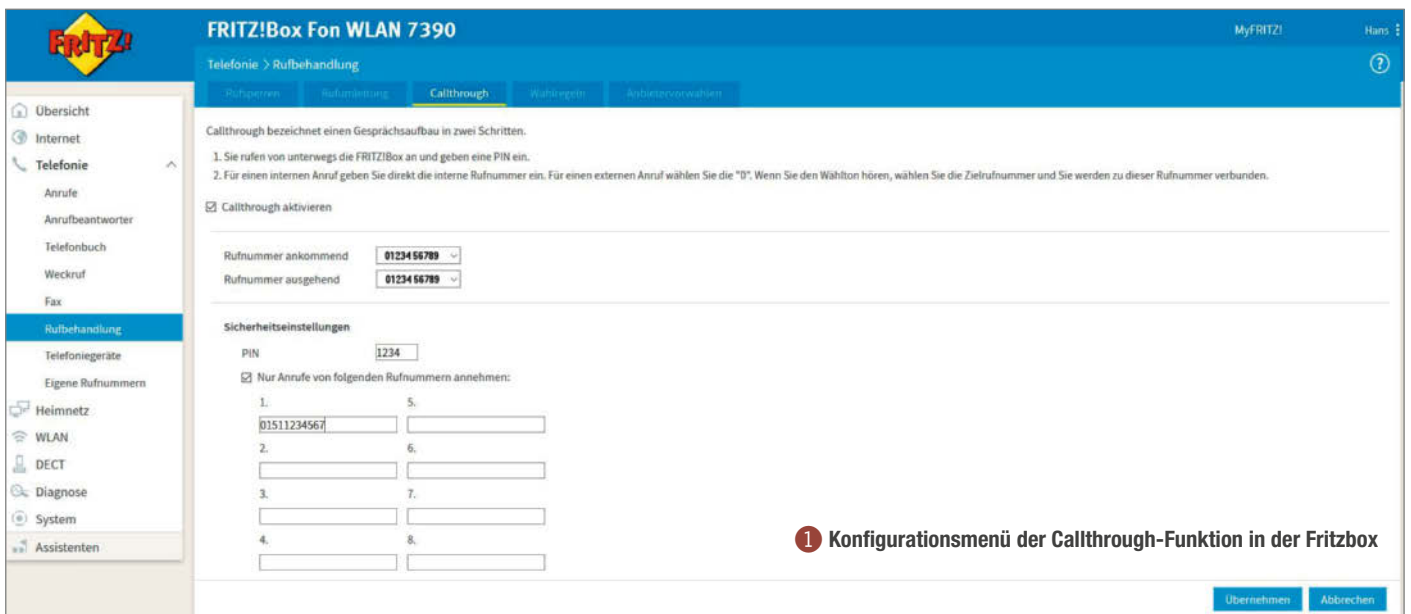
- » Wie man das Türklingel-Signal an die Fritzbox weiterleitet, erfahren Sie im Make-Artikel „Türklingel mit Arduino“
- » Eine weitere Methode zur Verbindung lesen Sie im Online-Artikel „Eine smarte Türklingel mit dem ESP8266 und Fritzbox selber bauen“


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

ter, nicht aber gegen das Abhören mittels eines Sniffers geschützt (siehe Kasten „Vorsicht, Abhörgefahr“ auf Seite 78). Das kann man zwar durch Verwendung von Funkmodulen ändern, die ihr Signal verschlüsselt übertragen, ist in meinem Projekt aber noch nicht enthalten.

Die Einrichtung des Anschlusses an der Fritzbox entspricht der Einrichtung eines Analogtelefones. Da hilft notfalls die Online-Hilfe

der Box weiter. Achten Sie unbedingt darauf, dass dem analogen Telefon keine ihrer externen Rufnummern zugeordnet wird, denn dann würde jeder eingehende Anruf auf dieser Leitung bei Ihnen zu Hause die Tür öffnen. Es sei denn, dass Sie den analogen Anschluss über die *Callthrough*-Funktion der Fritzbox ① absichern (mit PIN und Angabe der berechtigten Rufnummer, zum Beispiel die eines Handys). Dann kann man auch ohne Beden-



FRITZ!Box Fon WLAN 7390 MyFRITZ! Hans

Telefonie > Rufbehandlung

Rufnummern Rufumleitung **Callthrough** Rufregeln Rufnummernverwaltung

Callthrough bezeichnet einen Gesprächsaufbau in zwei Schritten.

1. Sie rufen von unterwegs die FRITZ!Box an und geben eine PIN ein.
2. Für einen internen Anruf geben Sie direkt die interne Rufnummer ein. Für einen externen Anruf wählen Sie die "0". Wenn Sie den Wählton hören, wählen Sie die Zielrufnummer und Sie werden zu dieser Rufnummer verbunden.

☒ Callthrough aktivieren

Rufnummer ankommend:

Rufnummer ausgehend:

Sicherheitseinstellungen

PIN:

☒ Nur Anrufe von folgenden Rufnummern annehmen:

1. 01511234567	5.
2.	6.
3.	7.
4.	8.

Übernehmen Abbrechen

① Konfigurationsmenü der Callthrough-Funktion in der Fritzbox

ken eine externe Rufnummer dem Türöffner zuordnen. Großer Vorteil: Ist das alles korrekt eingerichtet, kann man mit dem Handy ebenfalls die Tür öffnen. Sehr nützlich bei einem Notfall, falls im Inneren niemand mehr dazu in der Lage sein sollte.

Das Basis-Modul

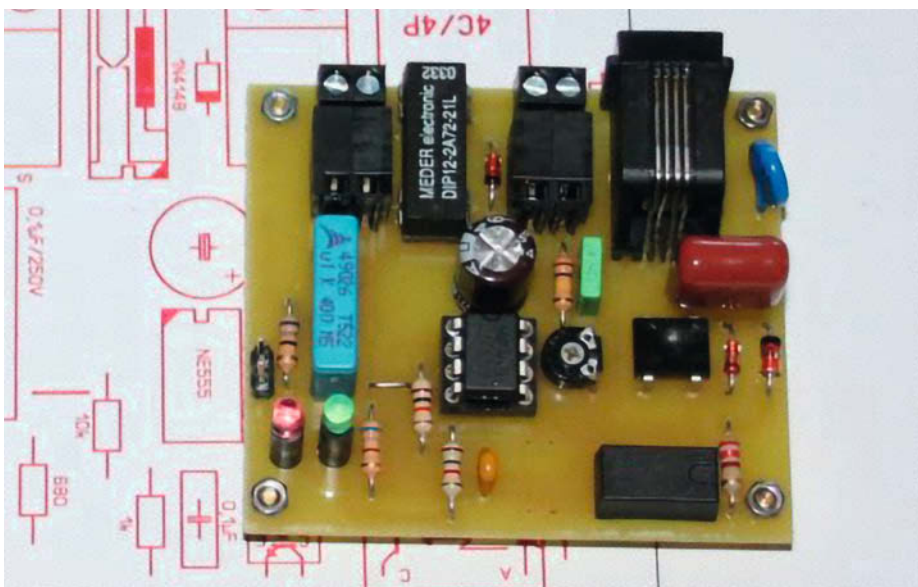
Nun zur eigentlichen Elektronik.

Sie besteht aus drei Funktionsblöcken:

- Analog-Interface mit Schalt-Aktor für den Türöffner (Basis-Modul) ②
- ISM-Sender für die kabellose Übertragung des Türöffner-Kommandos (optional)
- ISM-Empfänger für den Empfang des Türöffner-Kommandos (optional)

Die Schaltung des Basis-Moduls ③ kann auf einer Lochraster-Platine oder auf einer selbst geätzten Platine ④ erfolgen.

Die Layout-Dateien zur Platinen-Herstellung ⑤ stehen über den Kurzinfo-Link zum Download bereit. Der Schwierigkeitsgrad der Schaltung ist gering, die Schaltung ist also auch für Anfänger geeignet. Mein erster Prototyp war nach zwei Stunden aufgebaut (Lochrasterplatine). Die Kosten für die Bauteile belaufen sich auf etwa 20 Euro. Für die Bauteile (Basis-Modul und ISM-Platinen) gibt es über den Kurzinfo-Link erreichbare fertige Warenkörbe, falls Sie die Bauteile bei Reichelt bestellen möchten. Achtung: C5 wurde im Warenkorb durch einen 15nF-Kondensator ersetzt, da 12nF nicht lieferbar sind! Die Opto-

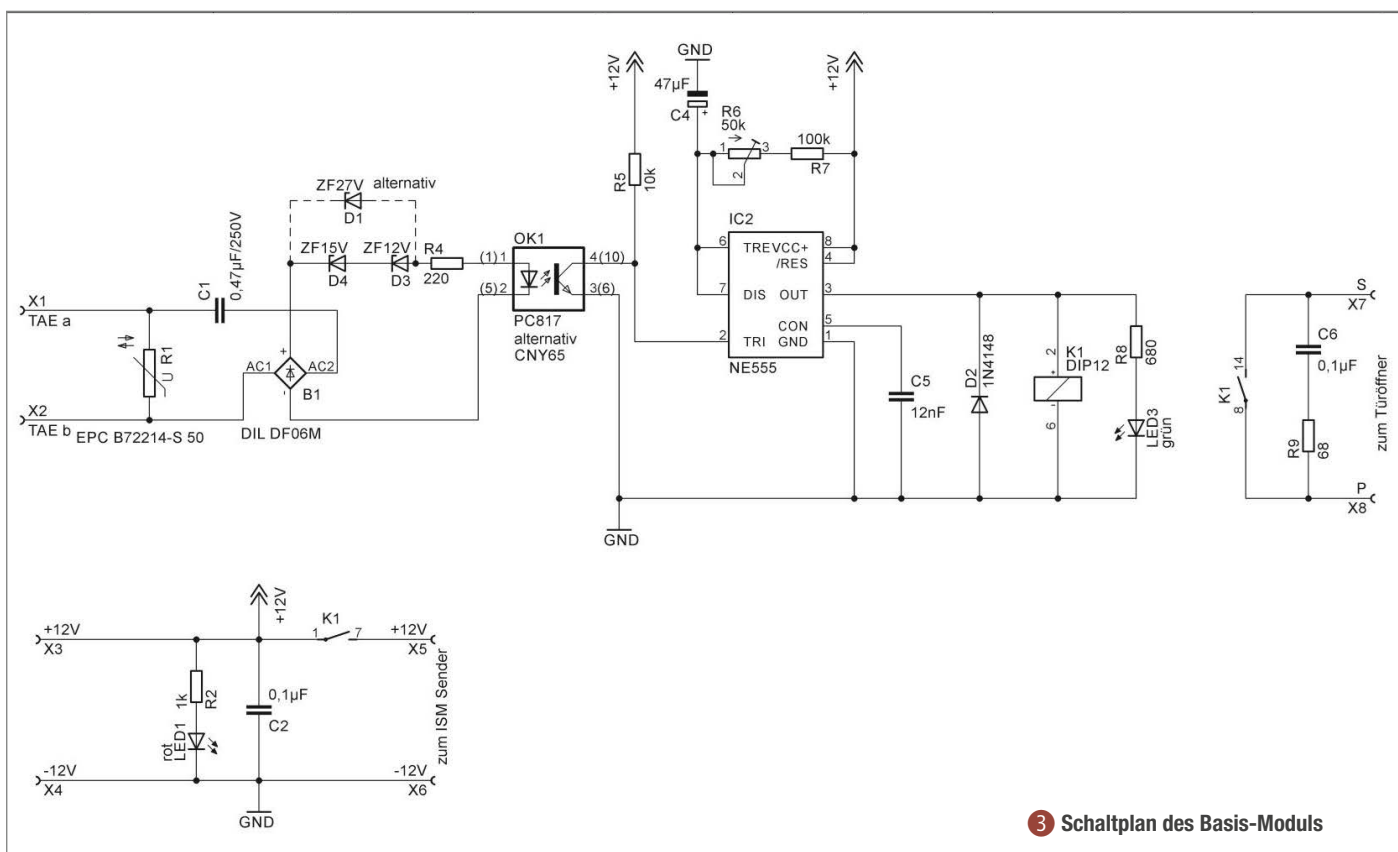


② Das Basis-Modul ist das Verbindungsglied zwischen Fritzbox und Türöffner.

koppler werden alternativ bestückt, also entweder der CNY65 oder der PC817. Der CNY65 hat wegen seiner Bauform eine höhere Isolationsklasse. Er ist deshalb auch für den Betrieb an einer alten Telefonanlage oder einem analogen Anschluss der Telekom geeignet.

Eigentlich ist die Installation des Basis-Moduls recht einfach, wenn da nicht die Zweidraht-Leitung zum Türöffner-Taster wäre. Bei geringem Abstand zwischen Fritzbox und

Taster ist das kein Problem. Wenn aber mehrere Meter Kabel quer durch die Wohnung gelegt werden müssten, kann es zu Akzeptanzproblemen kommen. Aber auch dafür gibt es eine Lösung, die da heißt: Funkfernbedienung. Das Basis-Modul hat einen geschalteten 12V-Anschluss, an den ein 433MHz-ISM-Sender ⑥ angeschlossen werden kann. Dieser funkt immer dann, wenn das Relais für den Türöffner anzieht. Der Empfänger sitzt direkt neben dem



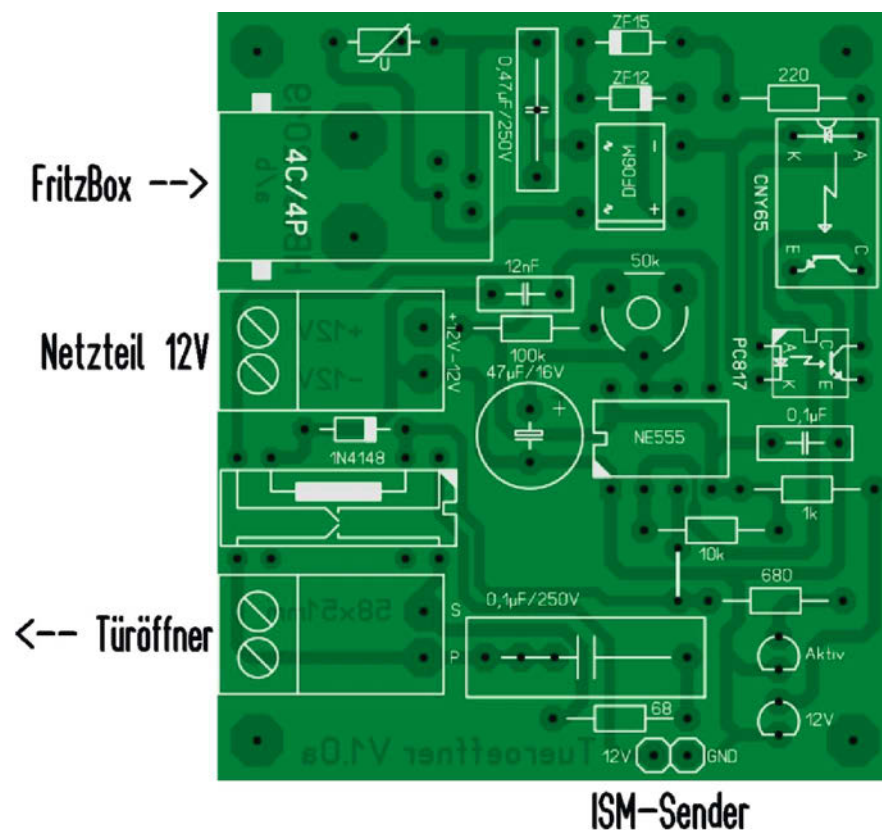
③ Schaltplan des Basis-Moduls

Listing 1: Türöffner-Sender

```

01 $regfile = "attiny85.dat"
02 $crystal = 8000000
03 $hwstack = 40
04 $swstack = 16
05 $framesize = 32
06
07 Config Portb.3 = Output          ' Sende Port aktivieren
08 Portb.3 = 1
09 Config Portb.4 = Output          ' Aktivitäts LED
10 Portb.4 = 1
11
12 Const Serout_extpull = 0         ' Pull up Seriell Port einschalten
13
14 Dim S As String * 10             ' Variable Ausgabe String
15 Dim Baudrate As Long             ' Übertragungsgeschwindigkeit
16
17 S = Chr(13) + "***Sesam" + Chr(13) ' Ausgabe String
18 Baudrate = 1200                  ' Übertragungs Geschwindigkeit
19
20 Do
21   Portb.3 = 1                    ' Ausgabe Port einschalten
22   Portb.4 = 0                    ' Aktivitäts LED ein
23   Waitms 10                      ' 10ms warten Wird als Break erkannt
24   Serout S , 0 , Portb , 3 , Baudrate , 0 , 8 , 1 ' String ausgeben
25   Portb.3 = 0                    ' Ausgabe Port ausschalten
26   Portb.4 = 1                    ' LED aus
27   Waitms 300                     ' 300msec bis zur nächsten Sendung
28 Loop
29
30 End
31

```

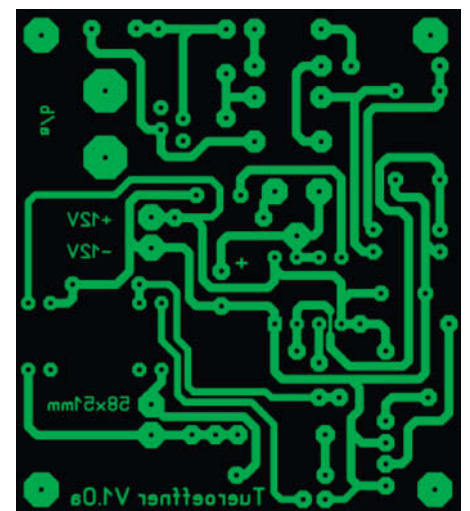


4 So sitzen die Anschlüsse auf der selbstgeätzten Platine.

Türöffner-Taster und braucht daher nur noch eine kurze Kabelverbindung.

Der Sender

Da das frei-benutzbare ISM-Band mit vielen Signalen vollgestopft ist, muss man das ausgestrahlte Signal individualisieren, damit die Tür nicht versehentlich geöffnet wird, wenn der Nachbar zum Beispiel eine seiner Funksteckdosen schaltet. (Lesen Sie dazu aber unbedingt auch den Textkasten „Vorsicht, Abhörgefahr!“ auf Seite 78) Das ist recht einfach und übernimmt ein Mikrocontroller vom Typ ATTI-



5 Die Vorlagen zur Platinen-Belichtung für das Basis-Modul

Listing 2: Türöffner-Empfänger

```

01 $regfile = "attiny2313a.dat"
02 $crystal = 8000000
03 $baud = 1200
04 $hwstack = 24
05 $swstack = 16
06 $framesize = 32
07
08 Config Input1 = Cr , Echo = Cr          ' Konfiguration Einlese Parameter
09
10 Config Porta.0 = Output                ' Relais Port
11 Porta.0 = 1
12 Config Porta.1 = Output                ' Led Port
13 Porta.1 = 1
14
15 Dim Na As String * 10                  ' Empfangsstring
16 Dim Pos As Byte                        ' Position des Schlüsselwertes
17 Echo On                                ' Serieller Input Echo aus
18 Print "Tueroeffner V1.0a"              ' Versionsmeldung
19
20 '***** Hauptprogramm
21 Do
22   If Ischarwaiting() = 1 Then           ' enthält UART ein Zeichen? enthält
23     Input Na                             ' Zeichenkette einlesen.
24                                           ' Ende wird mit CR markiert
25     Pos = Instr(na , "Sesam")           ' Empfangene Daten mit
26                                           ' dem Schlüsselwert vergleichen
27     If Pos > 0 Then                     ' Wenn Pos ungleich 0 ist
28                                           ' der Schlüsselwert gefunden
29       Print Na ; " " ; Pos              ' Empfangenen String ausgeben
30       Porta.1 = 0                       ' LED ein
31       Porta.0 = 0                       ' Relais ein
32       Wait 3                            ' 3 sec. warten
33       Porta.0 = 1                       ' Relais aus
34       Porta.1 = 1                       ' LED aus
35     ElseIf Pos <= 0 Then                 ' Kein sauberes Signal empfangen
36       Print "Schrott"                   ' 3 sec. warten
37       Wait 3
38     End If
39   End If
40 Loop
41 End

```

NY85 mit einem kleinen BASCOM-Programm (Listing 1). Zum Programmieren reicht die Free-ware-Version von BASCOM aus, ein entsprechender Programmierer ist in den Online-Warenkörben enthalten. Immer wenn der Sender aktiviert wird, schickt er im 300ms-Takt eine Schlüsselsequenz zum Funkmodul und dieses

sendet den Schlüssel zum Empfänger. Nur der eigene Empfänger kennt die Schlüsselsequenz und reagiert darauf. Für andere Informationen, die im ISM-Band rumschwirren, ist der Empfänger taub.

Beim Aufbau des Sende-Moduls ist einiges zu beachten. Wichtigster Punkt ist die Anten-

ne: Ohne Antenne haben die 433MHz-Module fast keine Reichweite, 2 Meter gehen gerade so. Will man quer durch die Wohnung durch Türen und Wände, geht es ohne Antenne nicht. Hier bieten sich die kleinen Stummelantennen für den 433MHz-Bereich an. Sie sind intern als Dipol geschaltet und benötigen keine Grundplatte als Gegenpol. (Ein einfaches Stück Draht mit der richtigen Länge (174mm) wäre nur eine halbe Antenne. Um auf volle Leistung zu kommen würde der Draht einen Gegenpol aus einer runden Metallscheibe mit 174mm Durchmesser benötigen. Das wäre eine klobige Angelegenheit.) Der nächste Punkt ist der mechanische Aufbau von Platine und Antenne. Die Stummelantennen werden als Vertikal-Strahler senkrecht montiert. Das geht am besten mit einem Alu-Winkel, auf dem Platine und Antenne montiert werden. Mit dem richtigen Aufbau schafft man dann in einem Gebäude locker 20-30m Funkstrecke.



Vorsicht, Abhörgefahr!

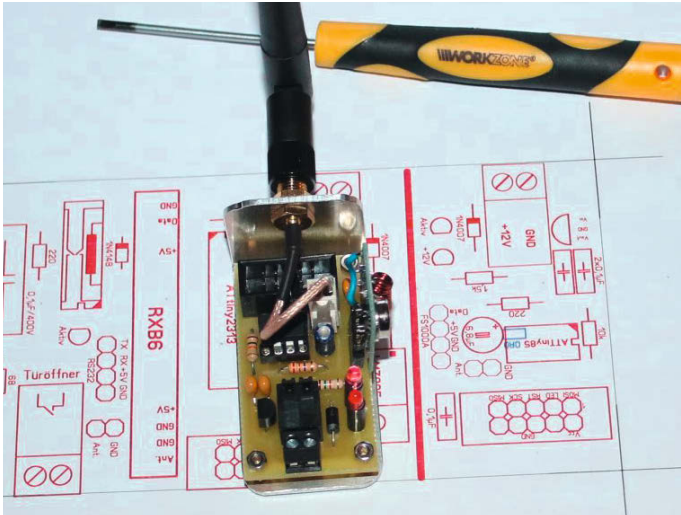
Die Übertragung des Steuercodes zum Türöffner im 433MHz-Bereich ist nicht abhörsicher: Die durch die Mikro-Controller gesendeten Steuersequenzen verhindern lediglich, dass der Türöffner auf Funksteckdosen-Sender oder ähnliches reagiert. Das Mitschneiden des Öffnungssignals verhindert sie nicht! Schon ein Arduino mit Empfängermodul und geeigneter Sniffer-Software kann das Signal mithören und den Steuercode anzeigen. Der kann dann von Unbefugten zum Öffnen der Tür benutzt werden. Betrachten Sie daher die hier verwendete Schaltung

nur als „Proof of Concept“. Sicher wird es erst durch Verwendung von Funkmodulen mit verschlüsselter Datenübertragung, zum Beispiel dem RFM69W. Oder Sie verbinden das Basis- und das Türöffnermodul mittels ESP-Modulen über WLAN. Auch diese Verbindung ist dann sicher, solange kein Unbefugter Zugriff auf das Funknetz erhält.

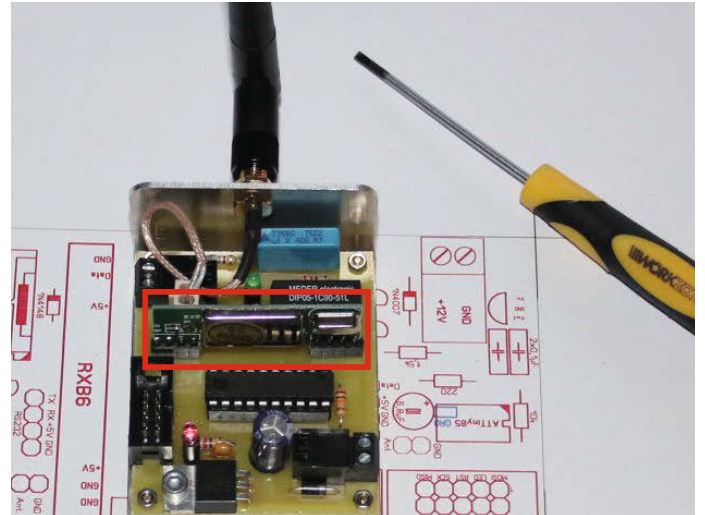
Wenn Ihnen eine passende Lösung dazu einfällt, dann senden Sie uns Ihr Erweiterungsprojekt bitte per E-Mail. Vielleicht können Sie ja bald Ihren Artikel hier lesen.

Der Empfänger

Die Auswahl des richtigen Empfängers **7** ist nicht so einfach. Die simplen und billigen Pen-



6 Das Sendemodul ist sehr kompakt.



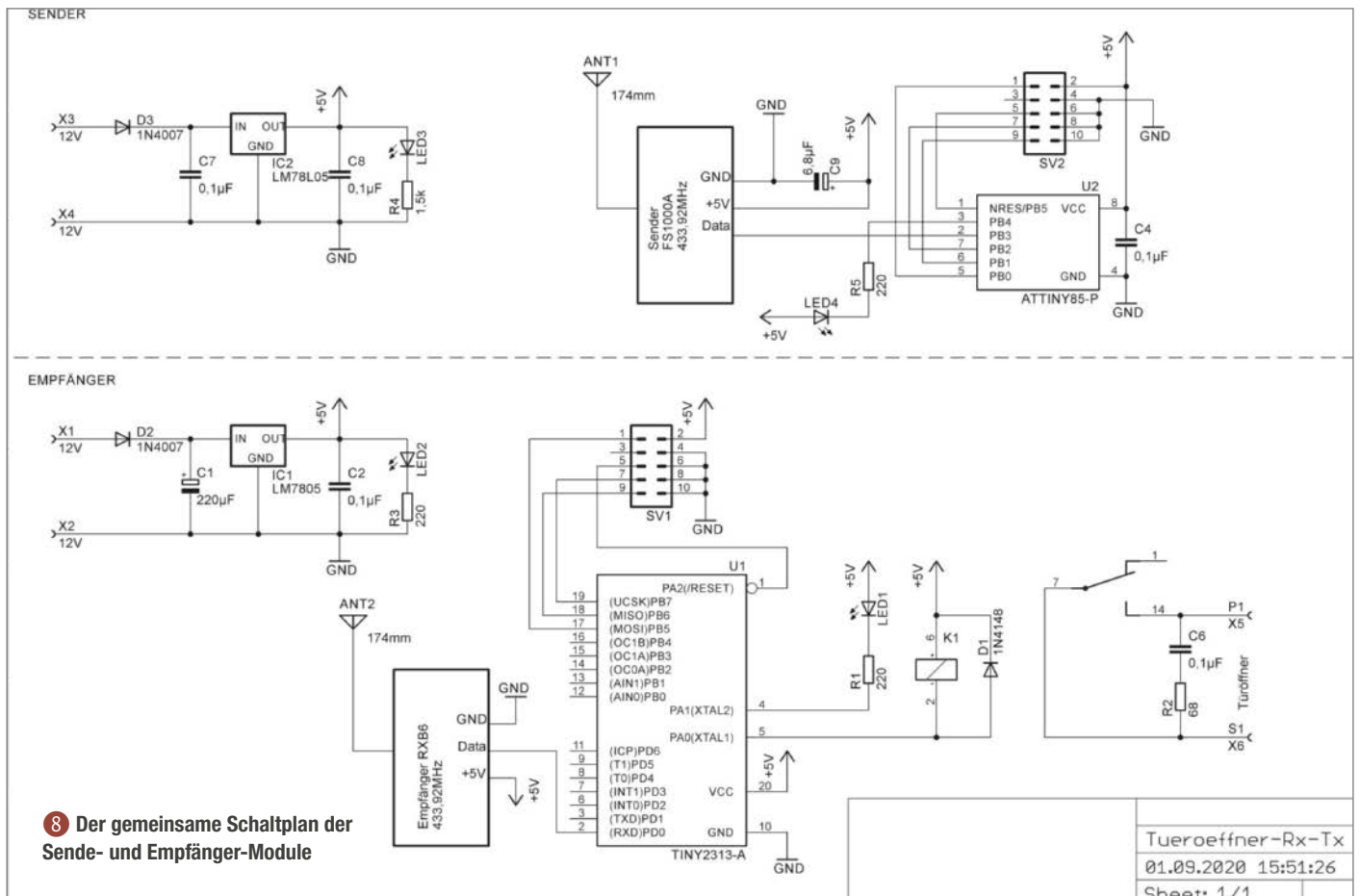
7 Das Empfangs-Modul mit dem Empfänger in der Mitte.

delempfänger, die zusammen mit den Sendern ausgeliefert werden, sind zu unempfindlich, um eine sichere Übertragung zu garantieren. Nur die teureren Superheterodyn-Empfänger sind zu gebrauchen. Auch eine ordentliche Antenne, die ebenfalls senkrecht montiert wird, ist Pflicht, genau wie für den Sender.

Zum Dekodieren des Sendesignal wird ein ATTINY2313 eingesetzt 8. Der Grund hierfür ist der UART, der im ATTINY85 nicht vorhanden ist. Nur mit einem UART kann das empfangene Signal sauber dekodiert werden. Das kleine BASCOM-Programm (Listing 2) lauert ununterbrochen auf das Schlüsselwort, um dann ein Relais anzusteuern. Das

Relais zieht für 3 Sekunden an und betätigt den Türöffner. Eine Pause von 3 Sekunden verhindert dann, dass der Türöffner zu oft eingeschaltet wird.

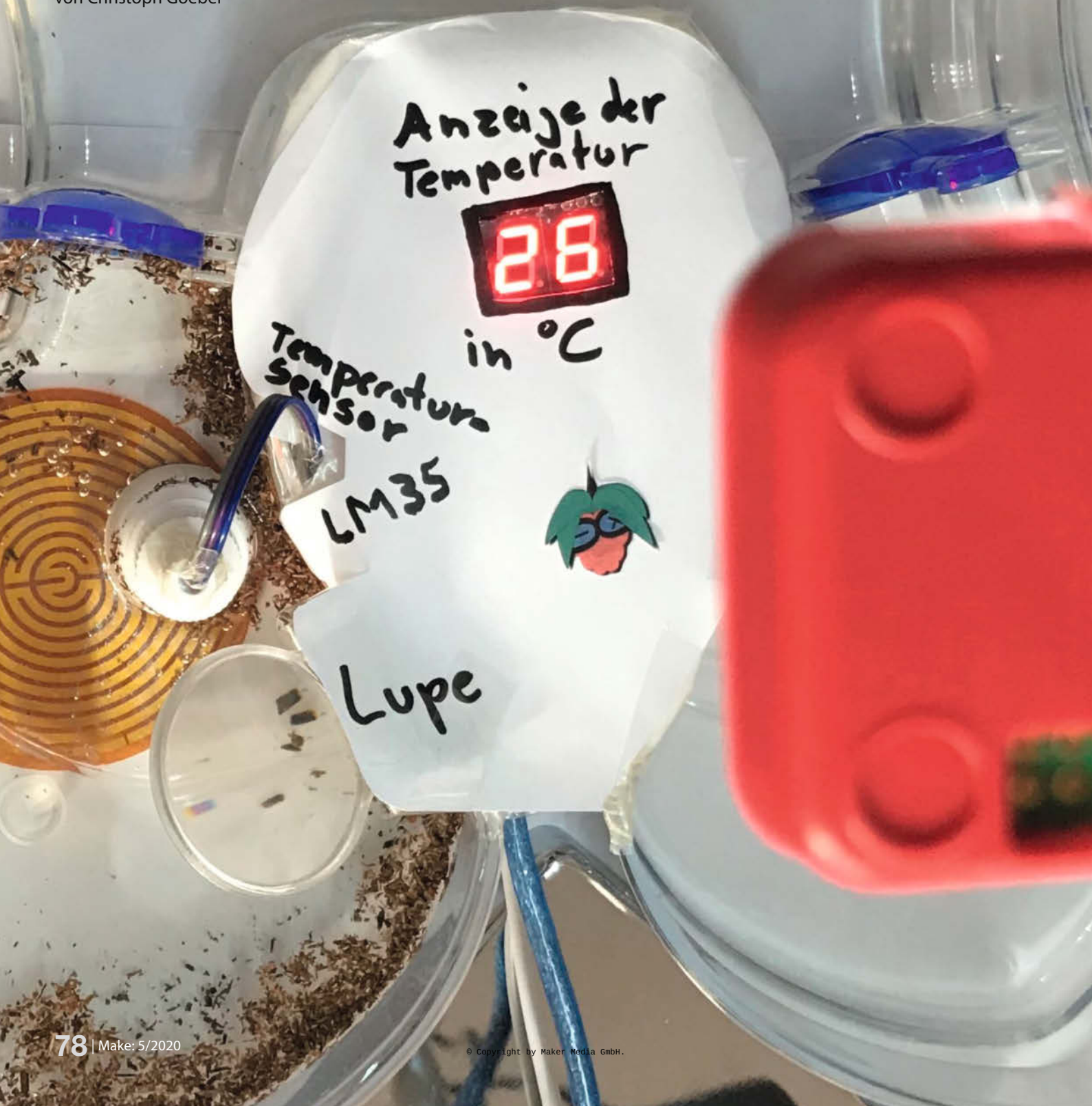
Auch für die Platinenherstellung der Sende- und Empfänger-Module stehen Dateien zum simplen Download zur Verfügung. —hgb



Zuchtstation für Urzeitkrebse

Mit Hilfe eines Arduino Uno und eines LM35 Temperatursensors wird die Wassertemperatur eines Triops-Aufzuchtbeckens ermittelt und das Wasser bei Bedarf über ein Heizpad aufgewärmt. Ein Raspberry Pi Zero hält mittels Kamera den Zuchtfortschritt fest.

von Christoph Goebel



Auf der Suche nach einem Geschenk für meine Nichte und Neffen fiel mein Blick auf ein Urzeitkrebsaufzuchtbecken. Mein Interesse für die kleinen Triops-Krebse war geweckt und nachdem ich das Becken verschenkt hatte, studierte ich Webseiten mit Aufzuchtstipps. Die Triops-Krebse gelten als älteste bekannte Tierart der Welt, die sich seit circa 200 Millionen Jahren kaum verändert hat. Damit sind sie sogar noch ein Stück älter als ihre bekannten Verwandten, die Salinenkrebse, die früher oft der Zeitschrift YPS beigelegt wurden.

Für ihre Aufzucht ist eine konstante Temperatur das A und O, aber gerade in den kalten Monaten kann es recht aufwendig sein, die Temperatur mit einer Wärmequelle wie einer Schreibtischlampe über Nacht konstant zu halten. Das muss doch anders gehen, dachte ich mir. Einen Tag später war ich Besitzer meines eigenen Aufzuchtbeckens und das Becken-Upgrade konnte starten.

Das durchsichtige Becken hat eine U-Form, die ich ausgenutzt habe, um die Arduino-Steuerung für eine Wasserheizung einzuklemmen. Außerdem habe ich die Aufzucht mit einer Raspberry-Pi-Kamera aufgenommen. Die Heizfolie wird von einem Arduino Uno über einen MOSFET betrieben und die aktuelle Temperatur mittels eines LM35-Sensors überprüft. Wie warm das Becken gerade ist, steht außerdem auf einer selbstgebauten 7-Segmentanzeige.

Anzeige mit eigenem Shield

Für das Anzeigen von zweistelligen Sensordaten habe ich mir bereits vor einiger Zeit ein eigenes Shield gebaut. Dazu verwendete ich eine Dual Prototyp Universal PCB Platine (6cm × 4cm). Das Board kann von beiden Seiten mit steckbaren Bauteilen bestückt werden. Ich habe zwei 7-Segmentanzeigen und die dazugehörige Ansteuerung aufgelötet. Per Multiplexing sorgen zwei Transistoren dafür, dass die Anzeigebausteine im Wechsel den Einer- bzw. Zehnerwert zur Anzeige bringen.

Jede 7-Segmentanzeige besitzt, wie der Name bereits vermuten lässt, mindestens 7 Leucht-Segmente. Nimmt man den Punkt noch hinzu, kommt man sogar auf 8 separat ansteuerbare Leuchtfelder. Bei zwei Ziffern und weiteren Bauteilen wie dem Temperatursensor ergeben sich so mehr benötigte Pins, als der Arduino zur Verfügung stellt. Daher setze ich ein Multiplexing-Verfahren ein, bei dem ich Signale für die beiden unterschiedlichen Segmente über den gleichen Pin sende – so schnell, dass das menschliche Auge nicht hinterher kommt. Auf diese Weise reichen mir zehn Arduino-Pins für die Shield-Ansteuerung.

In der Anzeigegruppe habe ich dafür zwei Segment-Bausteine mit gemeinsamer Kathode (K) verbaut. Die einzeln ausgeführten

Kurzinfo

- » **Temperaturanzeige mit Arduino Uno, LM35 und 7-Segment-Shield**
- » **Aufwärmen des Wassers über ein MOSFET-gesteuertes Heizpad**
- » **Regelmäßige Beckenaufnahmen mit Raspi Zero W**

Checkliste



Zeitaufwand:
5 Stunden



Kosten:
180 Euro



Programmieren:
Arduino- und Python-Grundkenntnisse



Löten:
optional

Mehr zum Thema

- » Florian Schäffer, Anzeigen am Arduino, Make 4/18, S. 26
- » Christoph Goebel, Internet-Radio mit Pi Zero, Make 1/19, S. 38
- » Felix Pfeifer, Raspberry Pi einrichten, online

Material

- » Arduino Uno
- » Triops-Becken
- » LM35-Temperatursensor
- » 12-V-Heizfolie (an 9 Volt betrieben)
- » MOSFET (IRF520)
- » DIY-7-Segmentanzeige Arduino Shield
- » Raspberry Pi Zero W
- » Pi-Kameramodul V2
- » Kunststoffbox 8,5cm × 5,5cm × 3cm
- » Eine vollständige Liste mit allen Kleinteilen finden Sie online.

Werkzeug

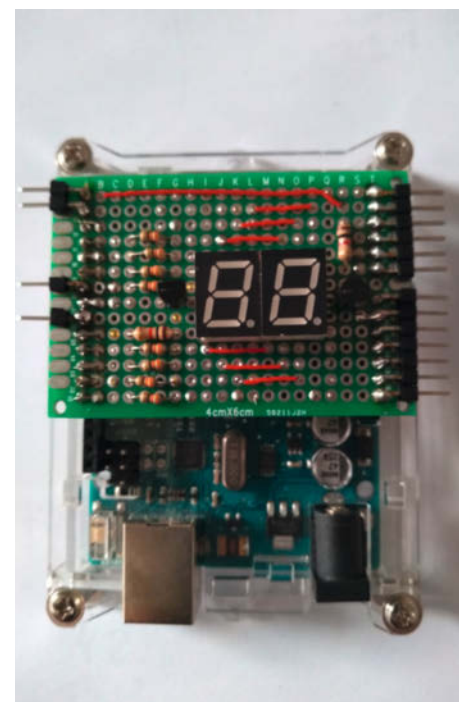
- » Spitzzange
- » Seitenschneider
- » Lötstation

Alles zum Artikel
im Web unter
make-magazin.de/x8sk

Anoden werden über die Output-Pins des Arduino mit dem Segmentmuster belegt. Jeder Ausgangspin eines Segments ist mit den dazugehörigen Eingangspins beider Segmentanzeigen verbunden. Somit sind die Verbindungen beider Bausteine parallel zueinander angelegt.

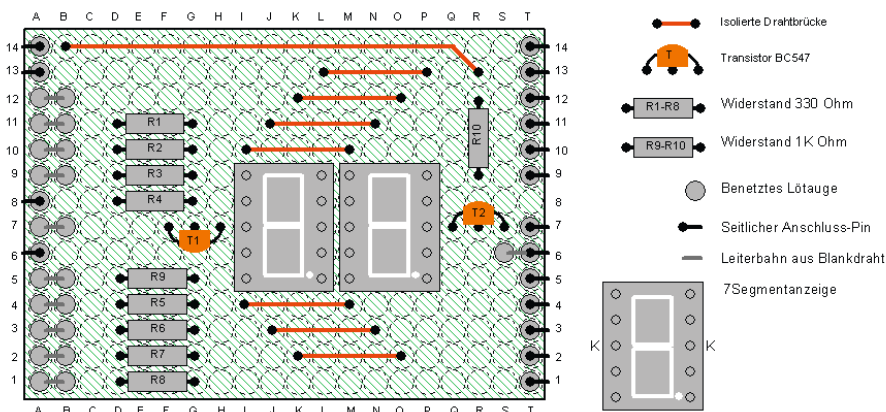
Liegt an einem Eingangspin eines Segments nun eine Spannung an, kommt es durch die Verbindung von Kathode zu GND (Masse) zum Stromfluss und damit zum Aufleuchten. Mittels der zwei Transistoren werden die zwei Steuerausgänge der Bausteine daher immer im Wechsel auf Masse gezogen. Der jeweils gerade aktive Baustein zeigt innerhalb der Aktivierungszeit das Ausleuchtmuster. Der andere Baustein leuchtet nicht, da ohne Masseverbindung kein Stromfluss stattfinden kann.

So kommt es, dass bei 21 Grad Celsius der linke Baustein das Muster für die „2“ und der rechte Baustein das Muster für die „1“ anzeigt. Passiert dieser Aktivierungswechsel in einer ausreichend schnellen Frequenz, ist das menschliche Auge nicht in der Lage, das Wechselspiel wahrzunehmen. Jede LED besitzt auf

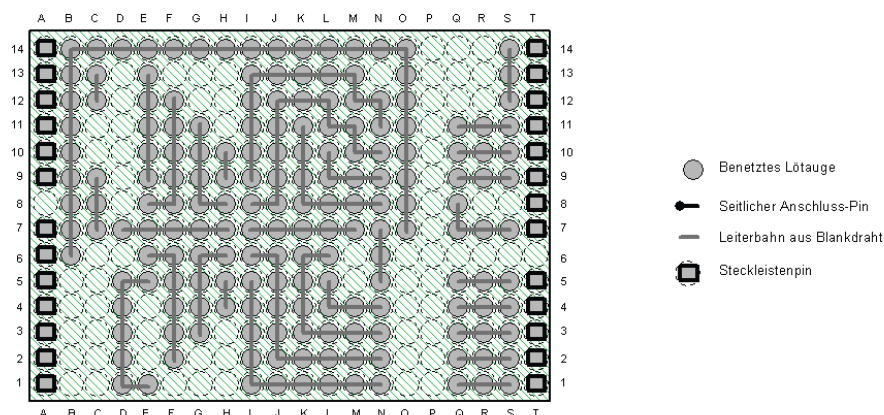


Shield auf Uno aufgesteckt

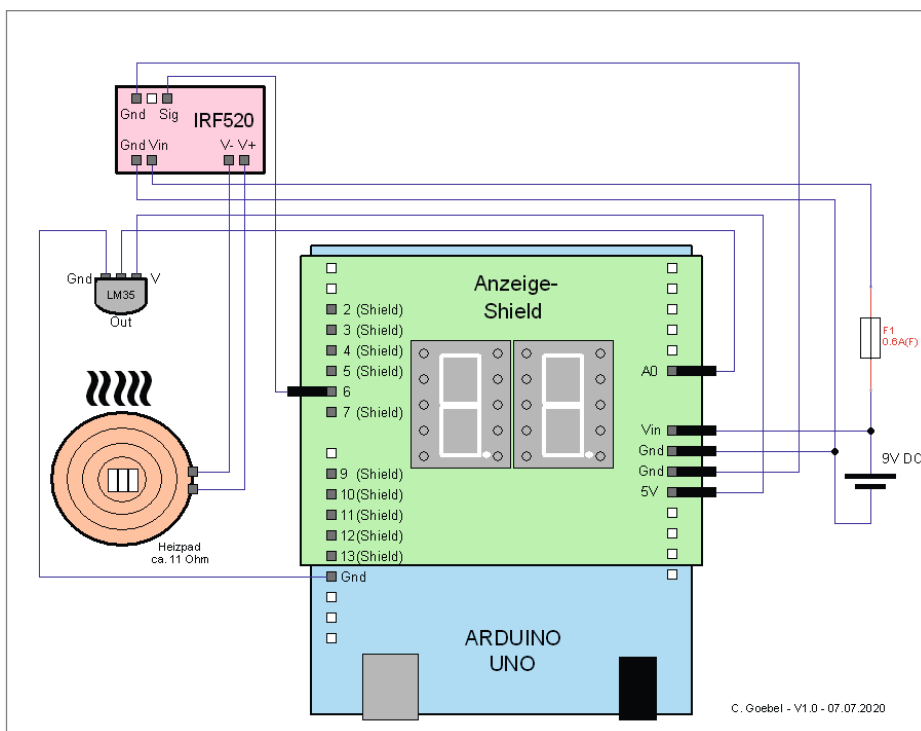
Platinenoberseite



Platinenunterseite (Oberseite nach rechts gekippt)



Aufbau meines Arduino-Shields mit 7-Segmentanzeige



Schaltplan des Projekts

der Platine schließlich ihren eigenen Vorwiderstand. Möchte man die Anzeige um einen dritten Baustein erweitern, benötigt man einen dritten Transistor inklusive Arduino-Steuerepin und muss die Verbindungen anodenseitig ebenfalls parallel und mit eigenen LED-Vorwiderständen ausführen.

An den Seiten habe ich Pinleisten verlötet, über die weitere, abgesetzte Bauteile, Baugruppen und Sensoren angeschlossen werden können. Auf der Unterseite können die Leisten auf den Arduino Uno gesteckt werden – da die Pinleisten des Arduino nicht ganz zum Raster der Platine passen, musste ich die Pins mit einer Spitzzange noch vorsichtig in Position biegen.

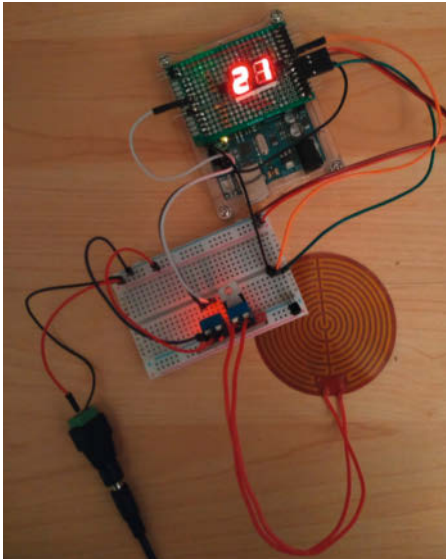
Neben meiner DIY-Anzeige-Platine gibt es von diversen Herstellern auch 7-Segmentanzeigen, die zum Beispiel mit integriertem Controller über die I²C-Schnittstelle angeschlossen werden können. Mit einem eigenen Aufsatz-Shield lernt man aber, wie Multiplexing funktioniert. Außerdem ist es schnell gebaut und individuell anpassbar.

Die Uno-Versorgung erfolgt mittels 9-Volt-Netzteil über den Pin *Vin* des Arduino, der über das Shield verbunden ist. Der 5-Volt-Pin versorgt den LM35 mit Spannung. Dieser speist seine analogen Signale über den Pin A0 ein. Der Arduino wertet diese Daten aus und steuert über Pin 6 per Pulsweitenmodulation das MOSFET-Board an, welches für die Regelung des Heizelements zuständig ist. Die Pins 2, 3, 4, 5, 7, 9, 10, 11, 12, 13 sind für das Anzeige-Shield zuständig. Über die *Gnd*-Pins beziehungsweise den Minuspol wird der Stromkreis geschlossen.

Wasserdichter Aufbau

Ich habe alle zur Schaltung gehörenden Komponenten so verpackt, dass weder ein sorgloser Wasserwechsel noch das Umkippen des Beckens zum Kurzschluss oder zum Wässern der Platine führen können. Das MOSFET-Modul samt Spannungseinspeisung, Sicherung und Anschlüssen steckt geschützt in einem Kunststoffgehäuse. Aufmerksame Leser kennen es von meinem Bürothermometer aus der Make3/20. Die Schaltung habe ich einfach auf einem Breadboard per Steckverbindern zusammen mit dem Arduino nochmal in einen Gefrierbeutel gepackt. Zum Glück passte alles in den Raum zwischen den beiden kleinen Becken. Auf der Oberseite verkleidete ich die Elektronik noch mit einer beschrifteten Pappe, wobei ich einen Ausschnitt für die beiden Anzeigebausteine anfertigte. Auch eine Lupe zum Vergrößern der Krebse habe ich später daran angeklebt.

Für den LM35 mit seiner kompakten Bauform, die einem Transistor zum Verwechseln ähnlich sieht, habe ich auch schnell ein was-



Vor dem Einbau habe ich die Schaltung bereits zusammengesteckt und getestet.



Arduino und Kunststoffbox in der Plastikhülle

serdichtetes Gehäuse gefunden. Ich schnappte mir ein altes Augentropfenfläschchen: In den Schraubdeckel habe ich eine kleine Öffnung gebohrt und die Anschlussverkabelung des LM35 hindurchgeführt. Damit der Sensor in die Flasche passt, habe ich außerdem den Tropfenspender mit einem Cutter-Messer abgeschnitten. Ein Silikontropfen half später, das verschraubte Fläschchen nach außen hin wasserdicht zu versiegeln. Um zu überprüfen, ob die Temperaturmesswerte trotz des Gehäuses noch stimmen, habe ich später mit einem Fleischthermometer parallel gemessen. Erstaunlicherweise waren die Werte sehr verlässlich und wichen nur selten voneinander ab. Zu guter Letzt wählte ich einen stabilen Aufstellungsort und klebte unten von außen die Beckenfüßchen und das Heizelement an.

Bei 9-Volt-Spannungsversorgung und einem 11-Ohm-Heizpad sollte sich bei voller Ansteuerung ein maximaler Heizstrom von 0,82 Ampere ergeben. In der Praxis betreibe ich die Pulsweitenmodulation (PWM) aber nur mit 180, damit sich das Wasser nicht zu schnell erwärmt und die Krebse kocht. So stellt sich nur ein gemessener Gesamtschaltungsstrom von 0,55 Ampere ein und ein 1-A-Netzteil ist ausreichend, wenn es für den Dauerbetrieb ausgelegt ist.

Arduino-Programm

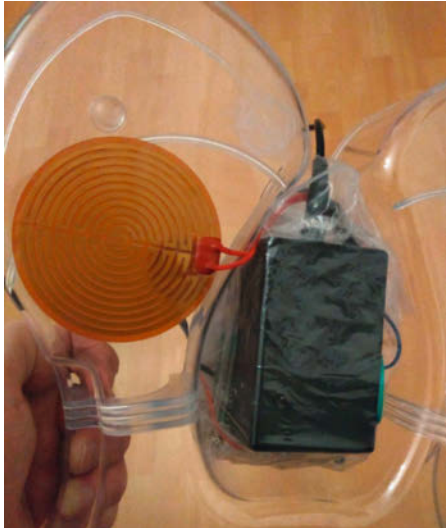
Das Erfassen der vom LM35-Sensor ermittelten Temperatur geschieht über den Analogpin 0 des Uno. Dieser ist dank eines 10-Bit Analog-Digital-Konverters in der Lage, Eingangswerte zwischen 0 und 5 Volt in Dezimalwerte von 0 bis 1023 umzuwandeln. Dabei bedeutet ein Grad Temperaturveränderung

einen Anstieg der Spannung um 10 Millivolt. Um die Spannungswerte in Grad Celsius umzurechnen, dividiere ich sie zunächst durch den Maximalwert 1023. Anschließend multipliziere ich sie mit der Maximalspannung von 5 Volt. Die Ergebnisse vervielfache ich im Programm noch um den Faktor 100, da der Sensor ja in 10-Millivoltsschritten misst. So sieht die Funktion aus:

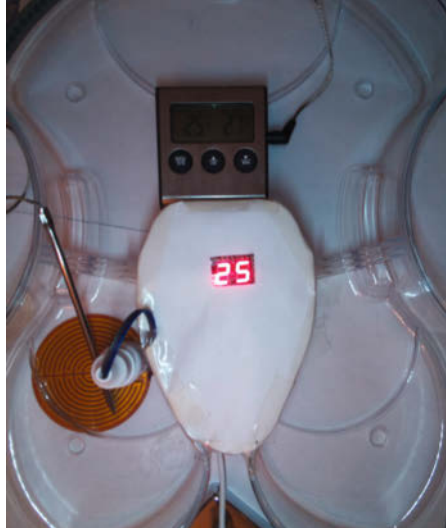
```
float TemperaturCelsius(){
    float AD_Wert = analogRead(lmPin);
    float Prozent = AD_Wert/1023.0;
    float Volt = Prozent*5.0;
    return 100.0*Volt;
}
```

Segmentmuster ansteuern

```
01 int lmPin = 0;
02 int PinArray[] = {10, 12, 11, 5, 4, 7, 3};
03 byte EinerPin = 2, ZehnerPin = 9;
04
05 byte segmentmuster[10] = {
06     B01110111, // 0
07     B00100100, // 1
08     B01011101, // 2
09     B01101101, // 3
10     B00101110, // 4
11     B01101011, // 5
12     B01111011, // 6
13     B00100101, // 7
14     B01111111, // 8
15     B01101111}; // 9
16
17 //EinerPin schalten
18 digitalWrite(EinerPin, HIGH);
19 digitalWrite(ZehnerPin, LOW);
20 aktivierung(einer);
21
22 //ZehnerPin schalten
23 digitalWrite(EinerPin, LOW);
24 digitalWrite(ZehnerPin, HIGH);
25 aktivierung(zehner);
26 }
27
28 void aktivierung(int a){
29     for(int j = 6; j >=0; j--){
30         digitalWrite(PinArray[6 - j], bitRead(segmentmuster[a], j) ==
31             1?HIGH:LOW);
32         delay(12);
33     }
34 }
```



Die Elektronik passt in den Beckenzwischenraum, das Heizelement habe ich daneben angeklebt.



Überprüfung der gemessenen Temperatur mit einem Fleischthermometer

Die Temperaturen übergebe ich an eine Anzeigefunktion, die die Einerstelle von der Zehnerstelle trennt:

```
void anzeige(int wert){
    byte einer, zehner;

    zehner = int(wert / 10);

    einer = wert - zehner * 10;
```

Die jeweils zugehörige 7-Segmentanzeige wird mit Hilfe des korrespondierenden Transistors angesteuert (Pin 2 und Pin 9) und das passende LED-Byte-Segmentmuster ausgegeben. Die sieben Segmente sind dafür wie folgt aufgeteilt: Der obere Strich an Pin 3 (Bit 0), die oberen Senkrechten an Pin 7 (Bit 1) und Pin 4 (Bit 2), der mittlere Strich an Pin 5 (Bit 3), die unteren Senkrechten an Pin 11 (Bit 4) und Pin 12 (Bit 5) und schließlich Pin 10 (Bit 6). Der Punkt kommt nicht zum Einsatz (Pin 13, Bit 7). Jedes Segmentmuster repräsentiert dann eine Ziffer.

Temperaturstufen

```
01 void loop(){
02   int temperatur = (int)TemperaturCelsius()+.5;
03   Serial.println(temperatur);
04   if ((temperatur > 14) and (temperatur < 18)){
05     heizung_stufe_6();
06     Serial.println("HS6");
07   }
08   else if ((temperatur > 17) and (temperatur < 21)){
09     heizung_stufe_5();
10     Serial.println("HS5");
11   }
12   else if (temperatur == 21){
13     heizung_stufe_4();
14     Serial.println("HS4");
15   }
16   else if ((temperatur == 22) or (temperatur == 23)){
17     heizung_stufe_3();
18     Serial.println("HS3");
19   }
20   else if ((temperatur > 23) and (temperatur < 25)){
21     heizung_stufe_2();
22     Serial.println("HS2");
23   }
24   else if ((temperatur > 24) and (temperatur < 26)){
25     heizung_stufe_1();
26     Serial.println("HS1");
27   }
28   else{
29     heizung_stufe_0();
30   }
31   void heizung_stufe_0(){
32     analogWrite(PwmPin, 0);
33   }
34   void heizung_stufe_1(){
35     analogWrite(PwmPin, 100);
36   }
37   void heizung_stufe_2(){
38     analogWrite(PwmPin, 140);
39   }
40   void heizung_stufe_3(){
41     analogWrite(PwmPin, 165);
42   }
43   void heizung_stufe_4(){
44     analogWrite(PwmPin, 170);
45   }
46   void heizung_stufe_5(){
47     analogWrite(PwmPin, 175);
48   }
```

Heizelement ansteuern

In der Sketch-Hauptschleife fällt die Entscheidung darüber, ob und wie stark das Heizpad das Wasser aufwärmen muss. Meine Triops-Art fühlt sich in der Aufzuchtphase bei 23 bis 26 Grad Celsius pudelwohl. Durch mehrere Testläufe ohne Tierchen kam ich zu dem Schluss, dass sechs verschiedene Heizstufen genügen sollten: Nach einem Wasserwechsel muss das Becken zügig, aber nicht zu schnell, aufgewärmt und im Normalbetrieb auf konstanter Temperatur gehalten werden. Das destillierte Wasser, mit dem ich das Becken fülle, hat dabei meist eine Temperatur von 18 Grad.

Die Werte der PWM-Ausgabe können zwischen 0 und 255 liegen und werden von Pin 6 an den MOSFET ausgegeben. Hier hätte ich die Leistungsstufen auch berechnen können, aber habe mich auf die Erfahrung aus meinen Probeläufen verlassen. Als höchsten Wert nutze ich 180, um das Wasser aufzuheizen, wenn es unter 18 Grad warm ist. Der PWM-Wert wird dann in Fünfer-Schritten auf bis zu 165 heruntergeregelt, sobald höhere Temperaturen erreicht werden. Zwischen 23 und 26 Grad heizt die Folie nur noch etwas nach, darüber hinaus gar nicht mehr. Insgesamt dauert das Aufheizen so zwischen 30 Minuten und einer Stunde, je nach Außen- und Wassertemperatur.

Zeitgesteuerte Fotos

Um von meinen Triops-Krebsen regelmäßig Bilder aufnehmen zu können, platzierte ich einen Schwenkarm mit Pi Zero und passender Kamera über dem Becken. Ein Cronjob sorgt dafür, dass mein kleines Python-Skript alle 15 Minuten Aufnahmen macht und diese auf



Raspberry Pi Zero W mit Kameramodul V2 in Gehäuse

einer Speicherkarte sichert, über die ich per Samba-Netzwerkfreigabe zugreifen kann.

Auf dem Raspi nutze ich die aktuelle Version von Raspbian Lite, das wenig Platz auf der Speicherkarte in Anspruch nimmt, da es keine Desktop-Oberfläche besitzt. Nach dem Entpacken des heruntergeladenen Images benutzte ich den Win32 Disk-Imager unter Windows und installierte das Betriebssystem auf eine 16GB Micro-SD-Karte. Eine ausführliche Anleitung zur Installation und weiteren Einstellungsmöglichkeiten finden Sie online (siehe Download-Link).

Auf der Boot-Partition habe ich eine leere Datei `ssh` ohne Dateiendung angelegt, um den Zugang per SSH zu aktivieren. Eine zweite Datei mit dem Namen `wpa_supplicant.conf` enthält die passenden WLAN-Einstellungen, um sich nach dem Erststart sofort mit meinem Netzwerk verbinden zu können. Schließlich steckte ich die Speicherkarte in den Kartenslot des Raspberry Pi und verstaute ihn mit angeschlossener Kamera in einem Pi-Zero-Gehäuse. Nach dem ersten Starten stellte der Zero W eine Verbindung zu meinem Netzwerk her. Auf den neuen Netzwerkteilnehmer kann ich nun mit dem Programm Putty zugreifen. Der Hostname ist anfangs `raspberrypi`, der Benutzername ist `pi` und das Passwort lautet `raspberrypi`.

Via SSH verbunden landet man direkt im `pi`-Heimverzeichnis. Dort benötigte ich zwei neue Ordner, `pictures` und `scripts`. Ich legte sie über das Putty-Terminalfenster mit dem Befehl `mkdir -p pictures scripts` an. Ein `sudo apt update` und `sudo apt upgrade` brachte den Pi auf den neusten Stand und ein `sudo apt-install python3-picamera` installierte die aktuelle Python-Kameraanbindung. Das Aufnahme-Skript für Videos und Fotos (`make_picture.py`) erstellte ich im Unterordner `scripts` mit `nano ~/scripts/make_picture.py`. Das

Aufnahme-Skript

```
01 #!/usr/bin/python3
02 import picamera, time
03 cam = picamera.PiCamera()
04 cam.resolution = (2592,1944)
05 t = time.strftime("%Y%m%d_%H%M")
06 cam.capture('/home/pi/pictures/foto_%s.jpg' %t)
07 time.sleep(2)
08 cam.resolution = (1024,768)
09 cam.start_recording('/home/pi/pictures/video_%s.h264' %t)
10 time.sleep(10)
11 cam.stop_recording()
```

Skript lädt die Python-Pakete `picamera` und `time` und nimmt erst ein Foto und anschließend ein 10-sekündiges Video auf. Die Dateien werden im Ordner `pictures` gespeichert und jeweils mit einem Zeitcode versehen.

Damit ich die Aufnahmen später bequem vom PC aus ansehen und herunterladen konnte, installierte und konfigurierte ich den Raspi als Samba-Server (eine ausführliche Erklärung und die vorbereitete Konfigurationsdatei finden Sie online, siehe Kurzinfo).

```
sudo -s apt install samba
cd /etc/samba
mv /etc/samba/smb.conf /etc/samba/smb.conf.org
grep -Ev '^#|^;'|^$' smb.conf.org > smb.conf
```

Nach dem Einrichten startete ich den Pi und damit den Samba-Service neu. Zuletzt machte ich mich an die `crontab`-Datei, damit der Pi mein Aufnahme-Skript alle 15 Minuten ausführt. Ich fügte die folgende Zeile mittels `sudo nano /etc/crontab` am Ende der Datei an:

```
0,15,30,45 * * * * pi /home/pi/scripts/make_picture.py
```

Schließlich aktivierte ich die Pi-Kamera mit dem Terminalbefehl `raspi-config`. Da meine

offizielle Raspberry Pi V2 Kamera das bekannte Fokusproblem hatte, nutzte ich eine alte Kreditkarte, um die Fokussierung anzupassen – eine Anleitung dazu finden Sie online (siehe Downloadlink). Danach konnte ich die Triopskrebse mit vielen Bildern und Videos beim Aufwachsen beobachten.

Erweiterbarkeit

Mein Triopsbecken verfügt über drei Bereiche, die mit kleinen Wänden voneinander abgetrennt werden können. Bestimmt ließen sich auch alle drei Bereiche individuell überwachen und beheizen. Dazu wären mehr Temperatursensoren und Heizpads nötig, als ich bisher installiert habe. Außerdem sollte man dann die Heizströme im Auge behalten und das Netzteil entsprechend wählen.

Schließlich noch ein Hinweis zur Krebshaltung: Jedes Haustier, und sei es noch so klein, bedeutet Verantwortung. Triops-Krebse sind zwar keine Katzen oder Hunde, aber auch sie benötigen regelmäßig Licht, Futter, eine geeignete Umgebung sowie hin und wieder einen Wasserwechsel. All das sollte man bedenken, wenn man seinen Kleinen solch ein Becken als Geschenk macht. Erreichen die fleißigen Schwimmer ein Alter von zwei bis drei Monaten, hat man ganze Arbeit geleistet. —hch



Aufnahme eines Krebses mit dem Kameramodul



1

Die Mondphasenanzeige

Es ist nie zu früh, den Nachwuchs für Astronomie zu interessieren. In meinem Fall mündete die akute Mondbegeisterung meiner Tochter darin, dass ich für die Wand eine leuchtenden Mondphasenanzeige baute, kurz *MoPhA* (sprich: Mofa) genannt.

von Alexander Wankerl

Eigentlich ist Astronomie mein Hobby, aber momentan komme ich arbeitsbedingt, aufgrund des Hausbaus und letzten Endes wegen meiner Tochter, der kleinen Ideengeberin für dieses Projekt, nicht wirklich zum „Spechteln“, wie die Himmelsbeobachtung unter Astronomiebegeisterten oft genannt wird. Neben der Astronomie beschäftige ich mich jedoch auch – letzten Endes wegen dem Hausbau und auch beruflich – mit Mikrocontroller-Spielereien.

Nun ist es so, dass – ohne mein Zutun – unsere kleine Maus total fasziniert vom Mond war und teilweise noch ist. Ein Buch mit dem Titel „Der kleine BUH und der große Mond“ von Paloma und Ulises Wensell hat mich dann auf die Idee gebracht, meine MoPhA zu bauen, die auch bei bewölktem Himmel oder sehr spätem Mondaufgang stets zeigt, in welcher Phase der Mond ist.

Das Konzept

Am Anfang stand ich vor der Wahl, eine Drehscheibe mit den Mondphasen – ähnlich eines

Chronographen mit Mondphasenanzeige – und entsprechendem Drehmechanismus zu bauen oder das ganze einfach mit einer Illumination verschiedener Bildchen der Mondphasen zu machen ¹. Aufgrund der leichteren Umsetzbarkeit wurde es die zweite Variante.

Da die Konstruktion energiesparend sein sollte und um eine zusätzliche Hitzeentwicklung zu vermeiden, kamen zur Beleuchtung nur LED-Module in Frage. Zudem sollte die Anzeige zeitlich begrenzt sein. Daher wird momentan mittels einer Zeitschaltuhr die MoPhA nur zwischen sechs Uhr und acht Uhr morgens sowie zwischen 18 Uhr und 21 Uhr eingeschaltet; eine mögliche Erweiterung ist, den Einschaltzeitpunkt mit dem Mondaufgang zu koppeln. Die Bauzeit betrug etwas über einem Monat, da ich immer nur abends ein bis zwei Stunden arbeiten konnte, wenn mein Töchterchen bereits im Bett war.

Der Aufbau war damit eigentlich weitestgehend klar, als Basis sollte ein kundenspezifisch gefertigter Bilderrahmen mit Fotos der Mondphasen dienen. Aufgrund der leichten



² Eine Auswahl der verwendeten Baumaterialien – es wurde viel wiederverwendet, was beim Hausbau so anfiel.



Verfügbarkeit entschied ich mich, neun Bilder auf drei Fotos zu je 30cm × 20cm aufzuteilen, das fertige Motiv hätte somit Abmessungen von 90cm × 20cm. So sieht es zumindest in der Theorie aus – da die Fotos aber immer ein wenig größer sind, ergab sich am Ende ein Maß von 91,5cm × 20,3cm. Für den restlichen Korpus wurden weitestgehend Restmaterialien aus dem Hausbau verwendet: Styrodur aus Dämmelementen, Verpackungsmaterial der Dachziegel, Spachtelmasse, Montagekleber, Leerrohrabschnitte und anderes ².

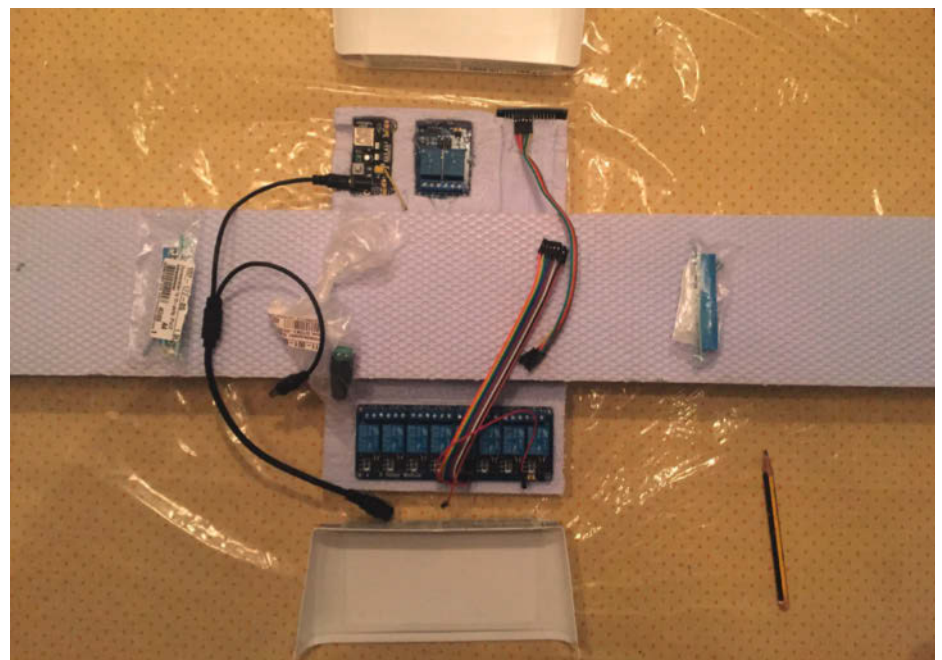
Der Unterbau

Als erstes prüfte ich die Anordnung der einzelnen Komponenten durch bloßes Positionieren auf den Styrodurelementen. Danach wurden die obere und untere Aufnahme der Elektronik (Stromversorgung, Relais-Module, Entwicklerboard) zur Aufnahme der Gerätschaften vorbereitet, das heißt, die Aussparungen ausgeschnitten beziehungsweise -gekratzt ³, nicht unbedingt die schönste Arbeit. Anschließend ging es an den 6cm dicken Hauptkörper und die Aussparungen für die LED-Module. Diese sind bis auf die drei roten am hinteren Ende platziert um eine möglichst gleichförmige Ausleuchtung zu erzielen ⁴. Als Reflektoren kam einfach Alufolie zum Einsatz, die LED-Module wurden in G4-Fassungen gesteckt, die wiederum durch kurze Kabelisolierungen (auch vom Hausbau übrig) fixiert wurden – insgesamt eine etwas knifflige Angelegenheit.

Die Kabellage verläuft weitestgehend in ausgeschnittenen Rinnen ⁵ ⁶ ⁷. Die Zielsetzung dieses Aufbaus war es, die Zugäng-

lichkeit für spätere Modifikationen oder Reparaturen zu erhalten. Daher sind auch die schwarzgrauen Verblendungen nur mittels Klettbanden fixiert. Bei dem zylinderförmigen Teil der Konstruktion handelt es sich um die Kappe einer PU-Schaumdose, die Ersatzteile (2× 3-LED-Modul, 1× 6-LED-Modul und

G4-Fassung) beinhaltet. Die Abdeckungen des Entwicklerboards, der Relaismodule und der Stromversorgung wurde durch eine halbierte Eisverpackung umgesetzt, welche mit schwarzer Folie beklebt wurde. Die restliche Dämmung wurde durch Überbleibsel der Dachziegelverpackung verkleidet.



³ Von links oben nach rechts unten: Stromversorgung, 2-Kanal-Relais, ESP32, 8-Kanal Relais und Steckbrücken

Kurzinfo

- » Beleuchtetes Wandbild zur Mondphasenanzeige
- » ESP32-Modul zur Steuerung
- » Ermittlung von Mondalter und Finsternisanzeige

Checkliste



Zeitaufwand:
30 bis 40 Stunden



Kosten:
90 bis 110 Euro



Löten:
einfache Lötarbeiten, Litze und LEDs in Rasterplatinen einlöten



Programmieren:
mit der Arduino-IDE

Material

- » Die vollständige Materialliste gibt es online (siehe Link)

Werkzeug

- » Laubsäge
- » Lochsäge
- » Stemmeisen
- » Lötkolben
- » Heißklebepistole
- » Cuttermesser und Schere
- » Mess- und Zeichenwerkzeug Bleistift, Lineal, Meterstab, Winkel

Alles zum Artikel
im Web unter
make-magazin.de/x5wg



4 Hier sind die Alureflectoren und die roten LEDs für die Mondfinsternisanzeige zu sehen.

Die Bilder

Am genauesten musste der Rahmenrücken aus MDF bearbeitet werden: Hier wurden Kopien der Fotos als Schablonen 8 verwendet, um die Silhouetten der Mondbilder genauestens ausschneiden zu können, damit später auch wirklich nur der Mond auf den Bildern leuchtet. Dies erfolgte mittels einer Laubsäge und klappte bis auf die abnehmende Sichel, bei welcher ich durch schwarze Folie Fehler korrigieren musste, auch sehr gut. Ein nettes Detail beinhaltet die obere Abdeckung für den ESP32: Das Board und die LEDs, welche *Power On* (rot) und bestehende WLAN-Verbindung (blau) anzeigen, sind durch eine transparente Acrylscheibe von oben sichtbar 9.

Die Elektronik

Als elektronische Bauteile kamen neben den bereits erwähnten LED Modulen (6x3 LEDs, 40lm, 0,5W, 3000K und 3x6 LEDs, 80lm, 0,9W, 3200K) auch noch 3 rote LEDs (100lm, diffus)

zur Anzeige einer Mondfinsternis zum Einsatz, ebenso zwei Relaismodule (einmal 8-Kanal und einmal 2-Kanal) sowie das eben schon erwähnte ESP32-WLAN/BT-Mikrocontrollerboard und diverse Steckbrücken sowie Adapterkabel. Die Stromversorgung erfolgt durch ein 12V/2A Standardsteckernetzteil. Prinzipiell besteht auch über einer USB-Buchse auf der Rückseite die Möglichkeit, ein Smartphone oder kleines Tablet zu laden, jedoch ist der Ladestrom auf lediglich 200mA oder weniger begrenzt. Darüber hinaus führt der Anschluss eines zusätzlichen Verbrauchers zu Instabilitäten.

Das exakte Mondalter (siehe auch Kasten) kann zudem über Bluetooth abgerufen werden. Zwar unterstützt das ESP32 sowohl WLAN als auch Bluetooth, jedoch habe ich hier mit dem HC-05 ein weiteres Modul verwendet, welches lediglich als *Serial-to-Bluetooth*-Schnittstelle fungiert. Über eine entsprechende App (*Bluetooth Serial Terminal* oder etwas ähnliches) kann das Mondalter einfach auf dem Tablet oder Handy ausgelesen werden. Ursprünglich sollte mit dem AT-09 ein BLE-Modul

zum Einsatz kommen, um den Strombedarf niedrig zu halten. Allerdings war dieses mit meinem Android-8.0-Tablet nicht kompatibel, so wurde es Bluetooth Classic. Der Mehrbedarf an Strom machte allerdings ein weiteres Stromversorgungsmodul notwendig. Der komplette Aufbau kann dem Schaltplan 10 entnommen werden, wer eine Verdrahtungsdarstellung von Fritzing bevorzugt, findet die ebenso online zum Download wie die beiden Versionen meiner Firmware.

Die Funktion

Das anfängliche Funktionsprinzip der MoPhA mit der ersten Firmwareversion 11 war dabei wie folgt: Das ESP32-Board ist mit dem hausinternen WLAN verbunden und ruft über einen GET-Befehl die Seite <https://www.astronomie.de/der-himmel-aktuell/> auf. Der Inhalt der Webpage wird nach dem String, welcher das Mondalter angibt, durchsucht und je nach Alter das entsprechende LED-Modul via Relais geschaltet. Rund um Neumond bleiben die Lichter bis auf die oben erwähnte orange Signalleuchte aus. Falls eine Mondfinsternis auftritt (egal ob partiell oder total) werden statt des 6-LED-Moduls die drei roten LEDs aktiviert 12.

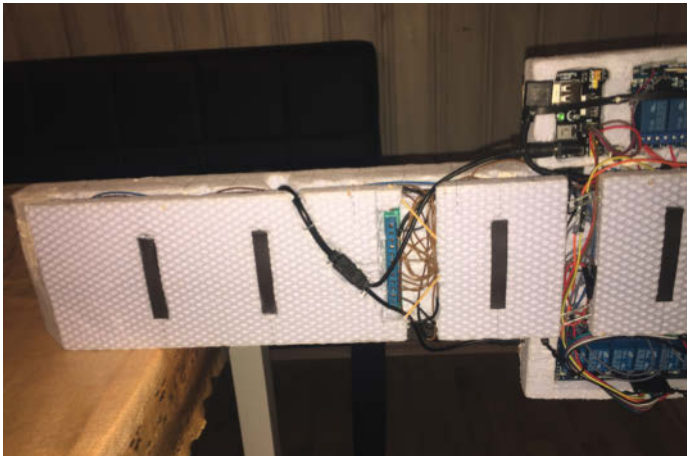
Allerdings gab es auf der oben genannten Seite zwischenzeitlich einen Bug und der entsprechende Subframe mit dem Mondalter war nicht mehr auffindbar (das wurde mittlerweile korrigiert). Da ich nicht bis zur Behebung des Problems warten wollte, entschied ich mich kurzerhand, die Programmierung abzuändern: Nunmehr lässt sich der ESP32 von einem NTP-Zeitserver das aktuelle Datum durchgeben 13 und berechnet mittels des Julianischen Datums (siehe Kasten) direkt das Mondalter 14. Der Algorithmus gibt momentan nur einen genäherten Wert aus, jedoch hat diese Version einen entscheidenden Vorteil: Der Sketch läuft wesentlich schneller und das System darüber hinaus stabiler. Das Durchsuchen des HTML-Strings nach dem Zielbegriff und das Auslesen des Mondalters beanspruchte eben doch deutliche Systemressourcen. Daher bleibt der Sketch erst mal so bestehen und ich schwenke nicht zur ersteren Version über, sondern optimiere eventuell noch den Algorithmus, um einen genaueren Wert für das Mondalter zu erhalten.

Der Feinschliff

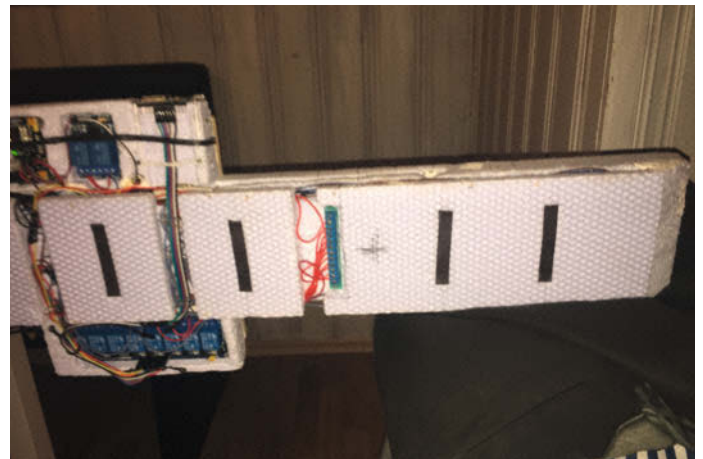
Darüber hinaus wurde die Abfrage der Mondfinsternis präzisiert: In der ursprünglichen Version war ich davon abhängig, dass eine Mondfinsternis auf der Hauptseite angekündigt wurde, eine Boolesche Variable würde beim Auffinden des Begriffs „Mondfinsternis“ auf TRUE gesetzt und der Vollmond rot illuminiert. In der neuen Version wird das Datum mit einer Liste zukünftiger Mondfinsternisse (bis zum Jahr 2050) abgeglichen und nur bei Überein-



5 Kompletter Korpus ohne Bilder von vorne gesehen – hier sieht man die Kabelrinnen.



6 Der Korpus in der linken ...



7 ... und der rechten Rückansicht.

stimmung das Bild rot beleuchtet. Zwar wird hier auch keine Unterscheidung zwischen Halbschatten-, partieller und totaler Finsternis unternommen, jedoch bin ich unabhängig (bis auf die Datums- und Zeitabfrage mittels eines NTP-Servers) von externen Datenquellen und deren Funktionstüchtigkeit.

Natürlich gab es während der Entwicklung auch eigene Bugs, die teilweise erst im Betrieb auffielen: So vergaß ich zunächst bei der Erst-

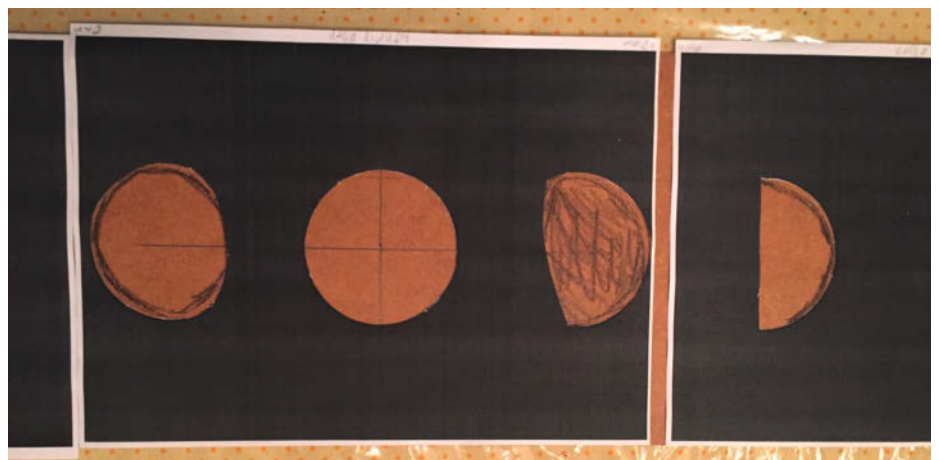
version, einen Abschaltbefehl für die vorangegangene Leuchtphase einzuplanen, was zu dem netten Effekt von zwei gleichzeitig beleuchteten Mondphasen führte ¹⁵. Ein weiteres Problem struktureller Natur stellte die Stromversorgung des ESP32 dar. Zunächst dachte ich, dass eine direkte 3,3V-Einspeisung ausreichend sein sollte. Jedoch zeigten sich hier deutliche Instabilitäten. Ein Umschwenken auf 5V und Verwendung des Board-eige-

nen DC/DC-Wandlers behob dieses. Kniffliger war hier bei der zweiten Version die Tatsache, dass trotz anstehender Mondfinsternis (Halbschattenfinsternis am 5. Juni 2020) das Vollmondbildchen nicht rot werden wollte. Zwar wurde die Mondfinsternis korrekt erkannt, jedoch hatte ich innerhalb der `if`-Schleife ein `break` vergessen, so dass die übergeordnete `for`-Schleife durchlief und den Boolean-Wert wieder auf `FALSE` setzte.

Das Julianische Datum

Das Julianische Datum (in allen Sprachen abgekürzt JD für englisch *Julian Date*) ist eine in den Naturwissenschaften, besonders der Astronomie, gebräuchliche Tageszählung. Die Zählung erfolgt in Tagen und Tagesbruchteilen, die seit dem 1. Januar des Jahres -4712 (4713 v.Chr.), 12:00 Uhr UT (*Universal Time*) vergangen sind. Dass dieses als Jahr 1 der Julianischen Periode mit ihrerseits einer Länge von 7980 Jahren gewählt wurde, hat seine Ursache darin, dass hier gemeinsam ein neuer Durchlauf der traditionell drei wichtigsten Zyklen der Kalenderrechnung begann: Des 28-jährigen Sonnenzyklus, nach dem etwa das Osterdatum berechnet wird, des dafür ebenfalls wichtigen 19-jährigen Mondzyklus und des 15-jährigen Zyklus der *Indiktion* (einer Jahreszählung aus der Antike).

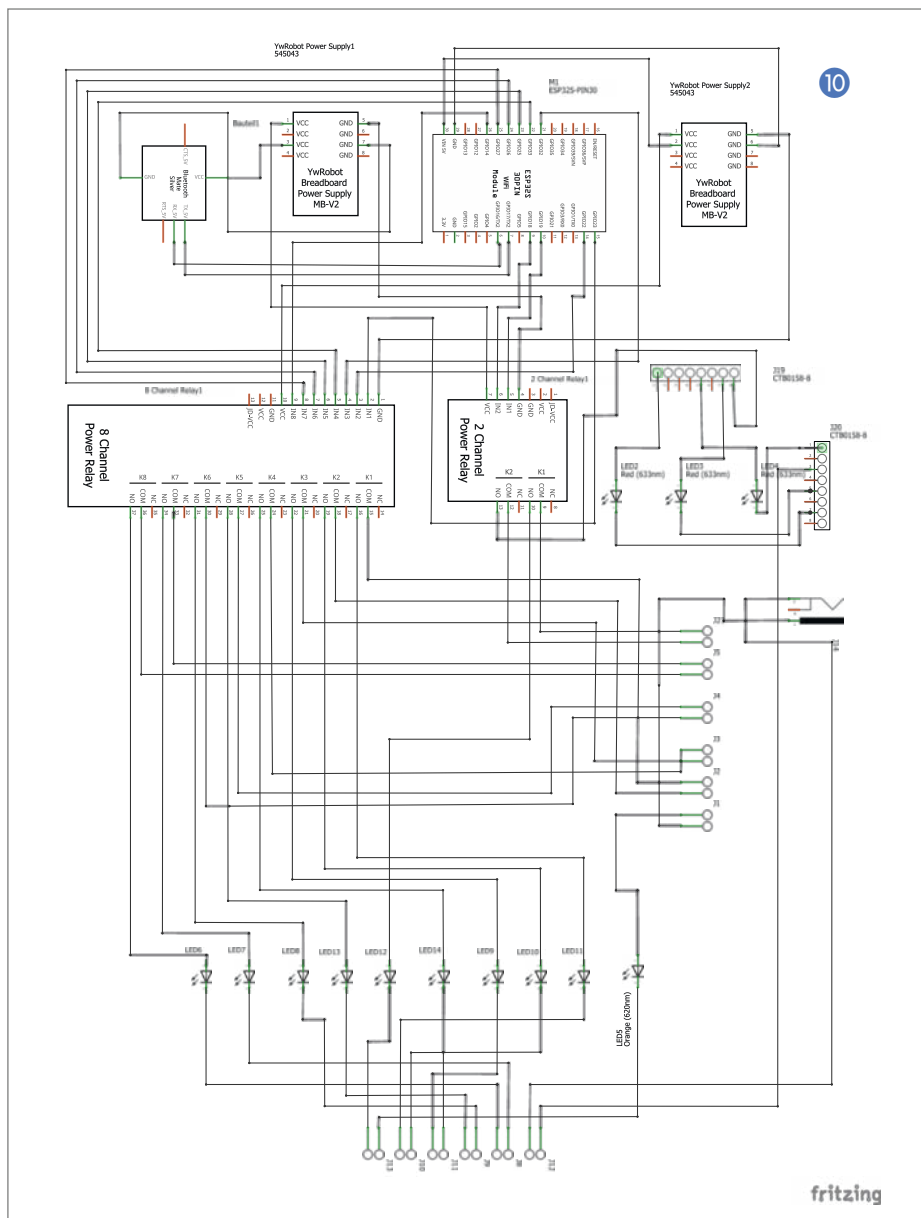
Zum Beispiel war der 8. August 2020 um 12:28 Uhr MESZ (Mitteleuropäischer Sommerzeit) das Julianische Datum 2459069.93611 (und nebenbei der 39ste Geburtstag des Autors).



8 In die Rückwand hinter den Bildern werden Löcher in Form der Mondphasen geschnitten, damit auf den Bildern später nur der jeweilige Mond von hinten beleuchtet ist.



9 Kompletter Korpus mit Verkleidung und Sichtfenster für den ESP32



Mondlauf und Mondalter

Bekanntlich benötigt der Mond fast 30 Tage, um einmal die Erde zu umrunden – diese Zeitspanne kennen wir als Monat. Leider hat der Mond keine Umlaufzeit von genau 30 Tagen, sondern eigentlich sogar zwei unterschiedliche: die *siderische* Umlaufzeit von 27,32 Tagen und die *synodische* Umlaufzeit von 29,53 Tagen. Erstere nimmt als Bezugspunkt einen Fixstern, letztere bezeichnet die Spanne von Neumond zu Neumond und definiert unseren Monat. Das Mondalter bezeichnet die Zeitspanne, welche seit dem letzten Neumond verstrichen ist und folgt hierbei der synodischen Umlaufzeit.

Ein Mondjahr hat demnach zwölf mal $29,53 = 354,36$ Tage und ist damit rund elf Tage kürzer als das eigentliche Kalenderjahr, welches sich wie die siderische Umlaufzeit des Mondes nach den Fixsternen richtet. Diese elf Tage beziehungsweise zwölf Nächte sind gemeinhin als „die Zeit zwischen den Jahren“ oder auch als Rauhnächte bekannt.

Eine Bestätigung, dass die Korrektur erfolgreich war, erhielt ich bei der Halbschattenfinsternis am 5. Juli 2020. Zunächst dachte ich bei der überraschenderweise roten Vollmondscheibe abermals an einen Bug, da ich diese Finsternis nicht im Kopf hatte. Schon wieder eine? Hat sich die MoPhA um einen Monat geirrt? Ist der Sketch etwa wieder fehlerhaft? Nichts davon, ein Blick in mein astronomisches Jahrbuch belehrte mich eines Besseren – die MoPhA hatte recht und funktionierte tadellos. Die Finsternis gab es, aber sie war nur in Nord- und Südamerika zu sehen.

Ausblick

Zukünftige Erweiterungen könnten eventuell weitere Ansteuerungsmöglichkeiten via Bluetooth etwa für Testläufe oder ein Lernprogramm sein oder das zeitgesteuerte Ein- und Ausschalten der MoPhA gemäß der Auf- und Untergangszeiten des Mondes. Doch, wie heißt es so schön – *never change a running system*. Leider zeigt die kleine Ideengeberin (zumindest momentan) wieder geringeres Interesse am Mond, jetzt sind eher die Raumfahrer und deren Anzüge interessant. Aber vielleicht ändert sich das ja genau so schnell wieder wie die Mondphasen. —pek



12 Rote Illumination des Vollmonds anlässlich der partiellen Mondfinsternis vom 16. Juli 2019

11 Erste Firmware (Auszug)

```

01 http.begin("https://www.astronomie.de/der-himmel-aktuell/?no_cache=1");
02
03 int httpCode = http.GET(); //Make the request
04
05 if (httpCode > 0) { //Check for the returning code
06
07     String line = http.getString();
08     Serial.println(httpCode);
09
10     if (line.indexOf("Mondalter") >= 0) { // Searchstring exists?
11         Serial.println();
12         int vonPos = line.indexOf("Mondalter ");
13         int bisPos = line.indexOf(" Tage");
14
15         Serial.print("Das aktuelle Mondalter ist ");
16         Serial.print(line.substring(vonPos + 10, bisPos));
17         Serial.println (" Tage");
18
19         // convert to float
20         String MA = line.substring(vonPos + 10, bisPos);
21         char char1[8];
22
23         MA.toCharArray(char1, MA.length() + 1);
24         MAWert = atof(char1);
25         Serial.println(MAWert);

```

13 Zweite Firmware (Auszug): Ermittlung des Datums

```

01 void loop() {
02     while (!timeClient.update()) {
03         timeClient.forceUpdate();
04     }
05
06     // The formattedDate comes with the following format:
07     // 2018-05-28T16:00:13Z
08     // We need to extract date and time
09
10     formattedDate = timeClient.getFormattedDate();
11     // formattedDate = "2024-03-26";
12     // only for eclipse test
13
14     Serial.println(formattedDate);
15
16     // Extract date
17     int splitT = formattedDate.indexOf("T");
18     dayStamp = formattedDate.substring(0, splitT);
19
20     Serial.print("DATE: ");
21     Serial.println(dayStamp);
22
23     decadestring = dayStamp.substring(2, 3);
24     YString = dayStamp.substring(0, 4);
25     MString = dayStamp.substring(5, 7);
26     DString = dayStamp.substring(8, 10);
27
28     Serial.println(decadestring);
29     Serial.println(YString);
30     Serial.println(MString);
31     Serial.println(DString);
32
33     decade = decadestring.toInt();
34     D = DString.toInt();
35     M = MString.toInt();
36     Y = YString.toInt();
37
38     Serial.println(Y);
39     Serial.println(M);
40     Serial.println(D);

```

14 Zweite Firmware (Ausschnitt): Berechnung des Julianischen Datums und Mondalters

```

01 double getPhase(int Y, int M,
02     int D) {
03     double AG, IP;
04     byte phase;
05     long YY, MM, K1, K2, K3, JD;
06
07     // calculate julian date
08     YY = Y - floor((12 - M) / 10);
09     MM = M + 9;
10     if (MM >= 12)
11         MM = MM - 12;
12
13     K1 = floor(365.25 * (YY +
14         4712));
15     K2 = floor(30.6 * MM + 0.5);
16     K3 = floor(floor((YY / 100) +
17         49) * 0.75) - 38;
18
19     JD = K1 + K2 + D + 59;
20     if (JD > 2299160)
21         JD = JD - K3;
22
23     // calculation of lunar age
24     IP = normalize((JD -
25         2451550.1) / 29.530588853);
26     AG = IP * 29.53;

```



15 Doppelte Beleuchtung aufgrund eines Fehlers in der ersten Firmwareversion

Berührungsloser Seifenspendender

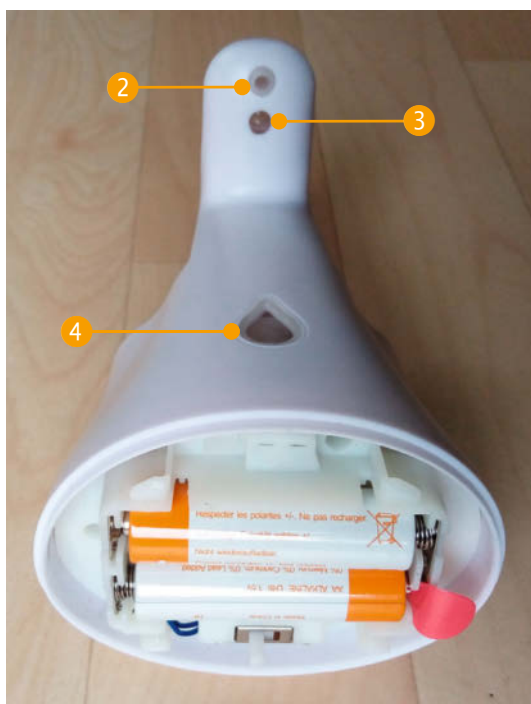
Für knapp 7 Euro ist ein batteriebetriebener automatischer Seifenspendender mit Bewegungssensor erhältlich, der erstaunlich gut ohne Tropfen und Kleckern funktioniert. Wir haben reingeschaut.

von Michael Gaus und Miguel Köhnlein

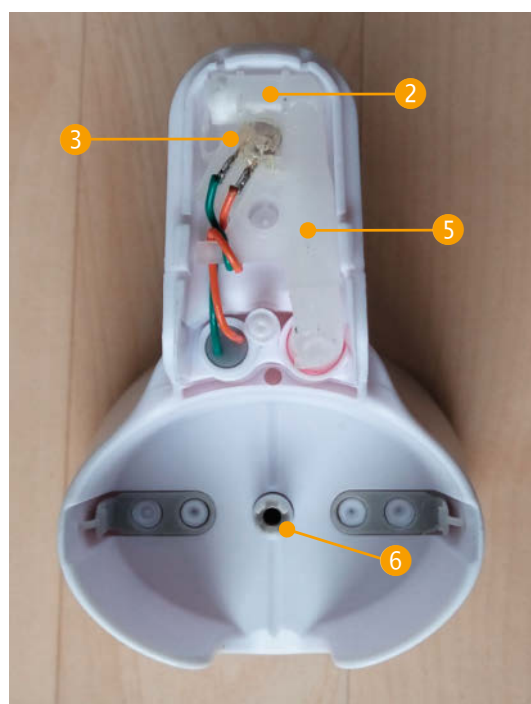


- ① Behälter mit Flüssigseife
- ② Ausgabe der Flüssigseife
- ③ IR Sendediode
- ④ IR-Empfänger hinter Sichtfenster
- ⑤ Transport-Kanal für Seife
- ⑥ Eintrittsöffnung für Seife aus dem Behälter

Durch Unterbrechung der Sichtverbindung zwischen IR-Sendediode ③ und IR-Empfänger ④ durch die vorgehaltene Hand wird die berührungslose Seifenausgabe ausgelöst.



Die IR-Sendediode strahlt Richtung Sichtfenster



Die Seife wird von ⑥ nach ② hochgepumpt.

Der hier gezeigte Seifenspender wird unter dem Namen *Sagrotan No Touch* vermarktet. Die benötigten zwei Mignon-Batterien sowie 250ml Flüssigseife sind bereits im Lieferumfang enthalten. Erhältlich sind die Sets mit unterschiedlichen Duftnoten ab 6,95 Euro in Drogeriemärkten, Apotheken und im Online-Versandhandel. Trotz des geringen Preises steckt in dem kleinen Gerät erstaunlich viel Technik.

Der Seifenbehälter ① wird kopfüber mit dem Deckel nach unten in das Gerät eingesteckt, bis er mit einem leichten Klicken hörbar einrastet. Die Seife wird über eine Öffnung ② am Ausleger des Gehäuses direkt in die darunter gehaltene Hand gepumpt. Um eine berührungslose Auslösung zu ermöglichen, sendet die Infrarot-Sendediode ③ zyklisch Infrarot-Impulse aus. Diese treffen normalerweise auf einen IR-Empfänger ④, der sich hinter dem tropfenförmigen Sichtfenster befindet.

Sobald man jedoch eine Hand unter die Ausgabeöffnung des Seifenspenders hält, wird der IR-Strahl unterbrochen und dies dann von der Elektronik erkannt. Dadurch wird der Motor gestartet, der über Zahnräder die Pumpe antreibt. Diese transportiert über einen Transport-Kanal ⑤ die durch die Eintrittsöffnung ⑥ angesaugte Seife nach oben bis zur Austrittsöffnung.

Eines der Zahnräder betätigt nach einer vollen Umdrehung einen Mikroschalter, in dessen Folge der Motor stoppt ⑦. Auf diese Weise wird eine definierte Seifenmenge abgegeben. Der Mechanismus funktioniert in der Praxis sehr gut und sauber, es tropft nichts nach.

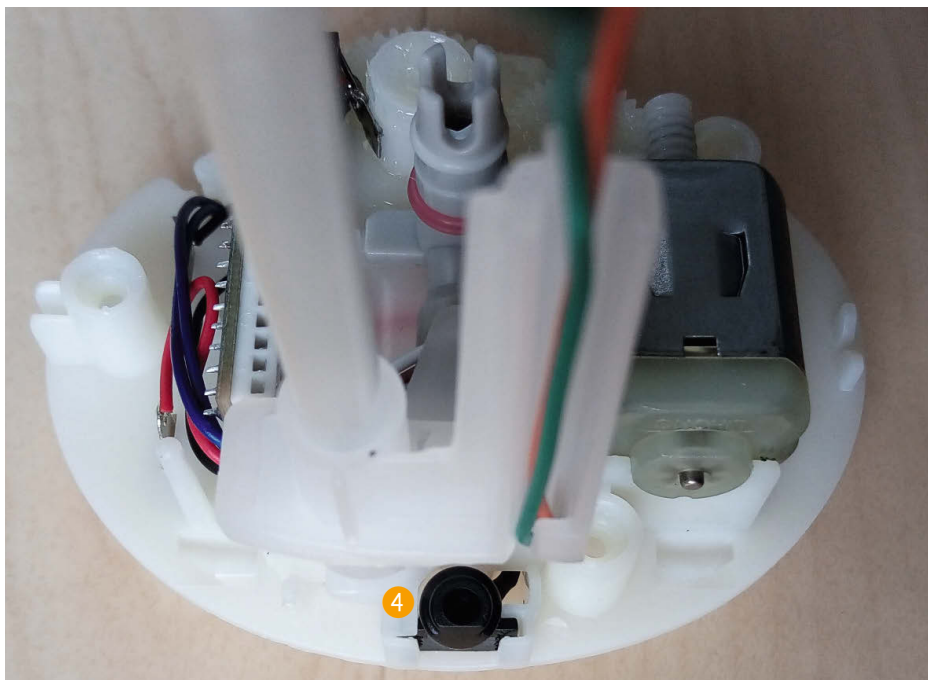
Auf der Steuerplatine befindet sich ein Mikrocontroller mit 8 Pins, leider ohne Aufdruck auf dem Gehäuse. Daher wird man ihn kaum für andere Zwecke benutzen können. Der Platz reicht aber für andere Controllerplatinen aus.

Die Elektronik wird direkt mit der 3V-Batteriespannung aus den beiden Mignon-Zellen versorgt. Vor dem Wechsel des Seifenbehälters sollte man den Ausschalter betätigen, damit während des Austauschs nicht aus Versehen ein Trigger-Impuls ausgelöst wird.

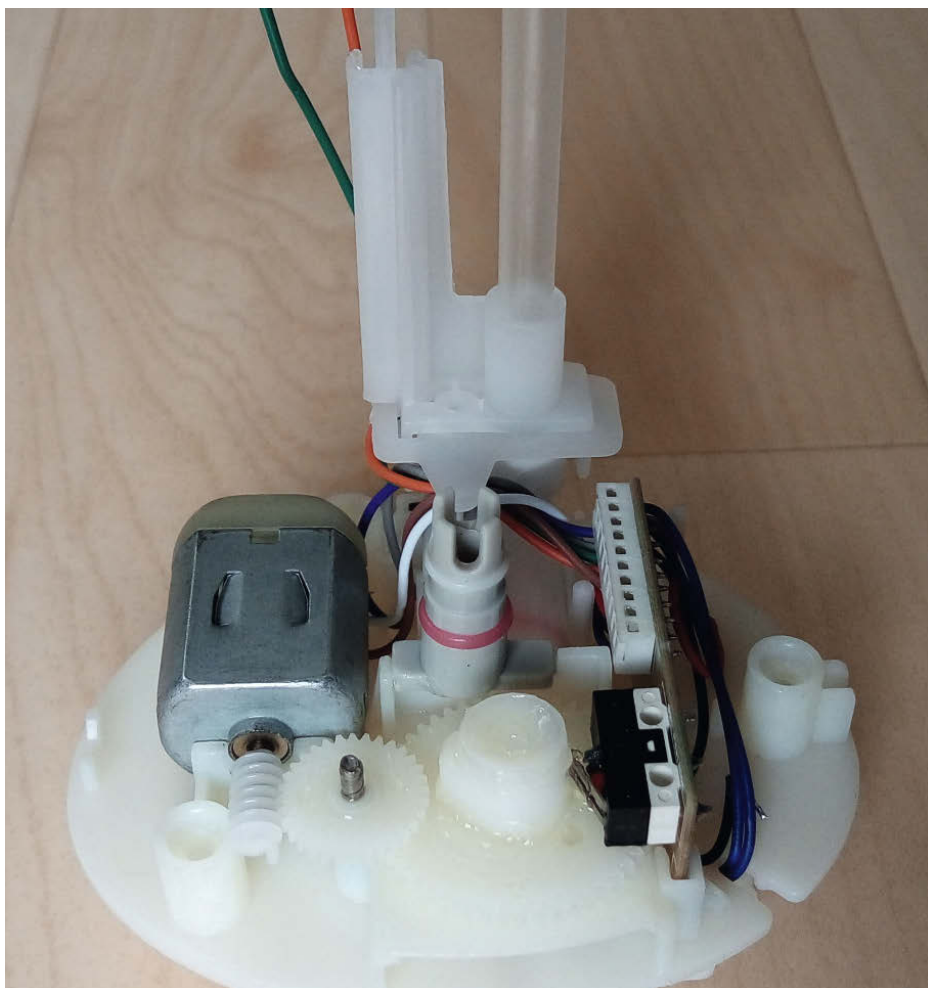
Im Grunde enthält der Seifenspender viele Bauteile, die auch für den berührungslosen Handwasch-Timer aus der Make 3/20 (ab Seite 68) nötig sind. Da bietet sich an, beides zu kombinieren. Noch verlockender wird dies, wenn man das herstellerseitig nicht vorgesehene Nachfüllen des leeren Seifenbehälters durch einen kleinen Umbau eines leeren Behälters nachrüstet. Dann kann man sogar die persönliche Wunsch-Flüssigseife verwenden. Die genaue Vorgehensweise ist im Artikel „Berührungsloser Seifenspender mit Handwasch-Timer“ ab Seite 92 beschrieben.

—hgb

► make-magazin.de/xxz8



Das kleine schwarze Teil vorn ④ ist der IR-Sensor.



⑦ Der Motor (links) treibt über Zahnräder die Pumpe an. Sie saugt über das mittlere Rohr die Seife an und gibt sie über den Schlauch wieder von sich. Der Mikrocontroller sitzt zusammen mit dem Mikroschalter zur Erkennung einer Zahnradumdrehung auf einer kleinen Platine.

Berührungsloser Musik-Seifenspender-Handwasch-Timer

Den (Handwasch-)Timer mit berührungslosem Schalter aus Heft 3/20 halten wir für eine sehr gute Idee. Damit jedoch auch die Seife ohne Viren-Übertragungsmöglichkeit entnommen werden kann, stellen wir hier unseren Erweiterungsvorschlag vor.

von Miguel Köhnlein und Michael Gaus



Nicht nur im Bad und zu Corona-Zeiten, auch in der Küche ist ein berührungsloser Seifenspender oft Gold wert, da man dort beim Kochen und Backen meistens ziemlich klebrige Hände hat und somit dann eine Verschmutzung des Seifenspenders vermeiden kann. Und wenn dann während des Timer-countdowns noch Musik abgespielt wird, sorgt das Händewaschen gleichzeitig auch noch für gute Laune.

Seifenspender aus der Drogerie

Wir verwenden einen batteriebetriebenen Seifenspender, der für 6,99€ (mit Batterien und 250ml Seife) unter der Bezeichnung *Sagrotan No Touch* in Drogerien erhältlich ist (siehe Reingeschaut auf Seite 90). Es gibt mehrere Ausführungen mit unterschiedlichen Seifen-Duftnoten sowie eine spezielle Kids-Edition, die aber alle baugleich sind. Der Seifenspender verfügt bereits über entsprechende Sensorik, um berührungslos Seife entnehmen zu können. Die benutzen wir statt des im Artikel in Heft 3/2020 beschriebene Näherungssensormoduls, sodass der Handwasch-Timer automatisch startet, sobald Seife entnommen wird. Außerdem verwenden wir ein MP3-Player-Modul, das ebenfalls durch den Sensor gesteuert, Musik abspielt.

Gerät öffnen

Zum Öffnen des Seifenspenders schalten wir zuerst das Gerät über den Schalter ① auf der Rückseite aus, damit beim Hantieren nicht der Motor versehentlich gestartet wird. Außerdem entfernen wir einen eventuell bereits eingesetzten Seifenbehälter.

Danach nehmen wir das komplette Unter-teil ab, darunter kommen die Batterien zum Vorschein. Die drei nun sichtbaren Schrauben ② werden entfernt.

Anschließend nehmen wir das Plastikteil oberhalb der Seifenöffnung ab. Dieses ist mit Rastnasen geklippt und muss vorsichtig mit Fingernagel und einer kleiner Schraubendreherklinge abgehoben werden.

Darunter ist die Infrarot-Sendediode zu sehen. Die dort angeschlossenen Litzen (grün und orange) löten wir ab (Polung merken!) und wirren die Kabelschlaufe ③. Nun können wir das Innenleben als komplette Einheit inklusive der beiden Litzen vorsichtig von unten am Batteriefach herausziehen.

Umbaumaßnahmen

Auf der Platine sitzt ein Mikroschalter, der über eine Kunststoffnase auf dem danebenliegenden Zahnrad betätigt wird und so pro kompletter Umdrehung einmal ausgelöst wird. Die beiden äußeren Pins liegen an GND, der mittlere Pin ist an einem Eingangspin des Mikro-

Kurzinfo

- » Handwasch-Timer mit Seifenspender steuern
- » Timer-gesteuerte Musik-Wiedergabe mit MP3-Player
- » Seifenspender mit beliebiger Seife nachfüllen

Checkliste



Zeitaufwand:
ein bis zwei Stunden



Kosten:
etwa 25 Euro



Löten:
Litze an Kontaktstifte löten



Programmieren:
Arduino-IDE

Material

- » Arduino Uno oder kompatibles Board
- » Multifunction-Shield
- » Seifenspender Sagrotan No Touch
- » Soundmodul DFPlayer Mini
- » Widerstand 1 Kiloohm
- » Micro-SD-Speicherkarte
- » Lautsprecher 8Ω, 0,5 W
- » Steckernetzteil 9V, 1000mA, 5,5mm Hohlstecker
- » Schalllitze 0,14mm²
- » Isolierband

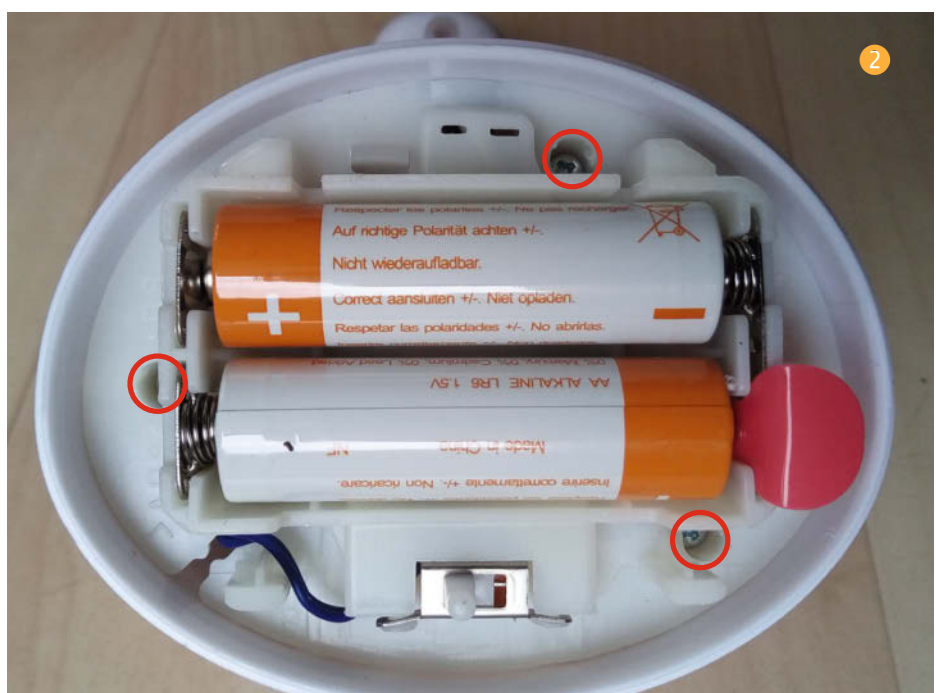
Mehr zum Thema

- » Den Artikel „(Handwasch-)Timer mit berührungslosem Schalter“ finden Sie in Make 3/20 ab Seite 68.
- » Informationen zum Display erhalten Sie im Artikel „Multi-Function-Shield“ in Make 2/18 ab Seite 74.

Alles zum Artikel
im Web unter
make-magazin.de/xcl5

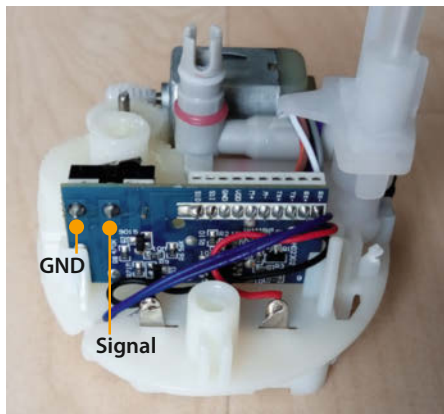


① Zuerst das Gerät ausschalten





3 Die beiden Litzen an der IR-Sendediode müssen abgelötet werden.



4 Am Mikroschalter wird ein zweiadriges Flachbandkabel angelötet.



5 Das Flachbandkabel des Seifenspenders wird am MFS angeschlossen.

controllers angeschlossen. Im Ruhezustand liegt dort ein HIGH-Pegel (3V) an. Sobald der Seifenspender ausgelöst wird, wechselt der Pin kurzzeitig auf LOW. Dies verwenden wir zur Triggerung des Arduinos im Handwasch-Timer, der an den Eingangspins bereits ab einer Spannung von 3V einen HIGH-Pegel erkennt. Wir müssen nur zwei zusätzliche Litzen am äußeren und mittleren Pin des Mikroschalters anlöten 4 und nach außen führen. Am besten eignet sich ein dünnes Flachbandkabel mit einer markierten Ader, das dann einfach beim späteren Aufdrücken des Batteriefachdeckels eingeklemmt wird.

Anschließend schieben wir die Einheit wieder ins Gehäuse und führen die grüne und orange Litze wieder nach oben Richtung IR-Sendediode. Nach dem Verschrauben der Einheit werden die Litzen wieder richtig gepolt an die Sendediode gelötet und die Plastikabdeckung aufgeklipst. Danach kann der Batteriefachdeckel wieder aufgedrückt werden,

wobei das Flachbandkabel leicht eingeklemmt und nach außen geführt wird.

Anschluss am Handwasch-Timer

Die beiden herausgeführten Adern des Flachbandkabels schließen wir auf dem Multifunction-Shield statt des Näherungssensormoduls an der Stiftleiste ~5 an (GND links, Signal rechts) 5. Am Arduino-Sketch brauchen wir zunächst nichts zu ändern, da dort auf eine HIGH-LOW-Flanke am Pin 5 gewartet wird.

Für einen ersten Funktionstest ohne Seifenbehälter schalten wir den Seifenspender ein und versorgen den Arduino per Netzteil mit Strom. Sobald wir die Hand unter die Öffnung des Seifenspenders halten, sollte der Motor zu hören sein und der Timer-Countdown auf der Siebensegmentanzeige des Timers starten. Nach erfolgreichem Test setzen wir den Seifenbehälter mit dem Deckel nach unten ein (ein-

rasten lassen!). Beim ersten Mal sind drei bis vier Auslösungen notwendig, bevor Seife aus dem Spender herauskommt.

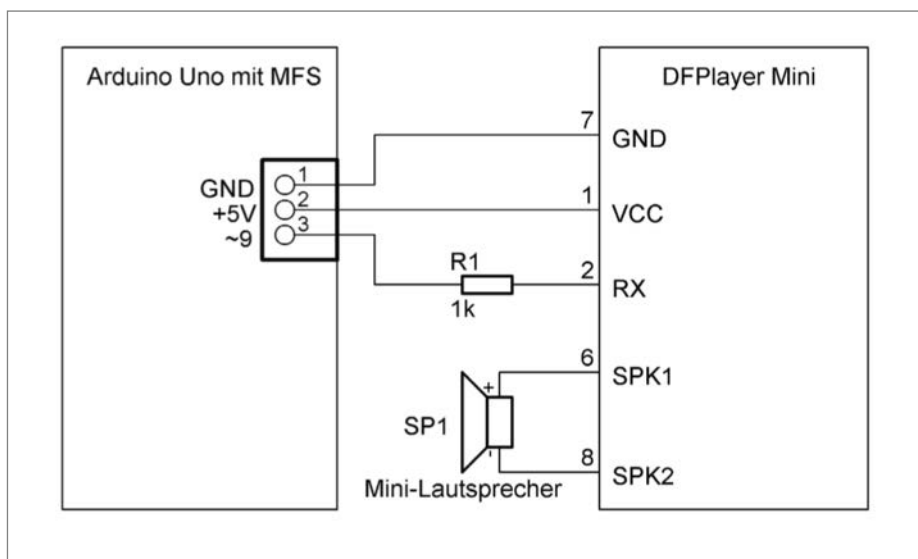
Nachfüll-Sperre

Der Seifenbehälter auf dem Spender ist nicht dafür vorgesehen, ihn zu öffnen und nachzufüllen. Man soll halt Ersatzseife nur vom Hersteller kaufen. Aber Maker finden eine Lösung und können so vom günstigen Kaufpreis des Geräts profitieren. Nach einem kleinen Umbau kann man auch seine persönliche Wunsch-Flüssigseife nachfüllen.

Der Seifenbehälter-Deckel ist nicht per Hand aufzukriegen, da zwei Kunststoffnasen dies verhindern 6. Durch Erwärmen des Deckels werden die Haltenasen jedoch weich, sodass sich der Deckel mit ein wenig Kraftaufwand aufschrauben lässt. Man kann entweder den Deckel kurz in ein Gefäß mit heißem Wasser eintauchen oder ihn mit einem Heißluftgebläse



6 Die beiden Plastiknasen werden mit einem Seitenschneider abgezwickt.



7 So werden der MP3-Player und der Arduino verbunden.

erwärmen. Sobald er warm genug ist, dreht man ihn mit einem Tuch auf. Damit wir künftig beim Nachfüllen leichteres Spiel haben, zwicken wir die beiden Rastnasen mit einem Seitenschneider ab und entgraten bei Bedarf noch mit einer kleinen Nagelfeile. Anschließend wird die gewünschte Flüssigseife eingefüllt und der Deckel wieder aufgedrückt. Ab jetzt lässt sich der Deckel einfach per Hand aufschrauben.

Probier's mal mit Musik

Um das richtige Händewaschen vor allem auch für Kinder interessanter zu gestalten, möchten wir den Waschvorgang mit Musik untermalen, sodass es nicht als lästige Pflicht empfunden wird. Das kleine und kostengünstige Soundmodul *DFPlayer Mini* eignet sich prima hierfür. Die Musik liegt im MP3-Format auf einer Micro-SD-Karte.

Der Anschluss am Arduino erfolgt über einen Portpin 7, der als *SoftwareSerial* konfiguriert und mit dem RX-Pin des Players verbunden wird. Zwischen Player-RX und dem Arduino muss noch ein 1kOhm-Serienwiderstand geschaltet werden, da der Arduino 5V-Pegel und der Player nur 3,3V-I/O-Pegel hat. Die Verbindung von Player-TX zum Arduino wird nicht benötigt.

Das Soundmodul verfügt über einen internen Verstärker, sodass ein kleiner Lautsprecher mit 4 bis 8 Ohm Impedanz direkt angeschlossen werden kann. Je nachdem, wo dieser platziert werden soll, kann man auch einen wasserdichten Duschlautsprecher verwenden.

Im angepassten Arduino-Sketch *Handwaschtimer_Musik.ino* (Download siehe Kurzinfo-Link) wird zusätzlich die *SoftwareSerial*-Bibliothek sowie die für den DFPlayer Mini passende Bibliothek *DFRobotDFPlayerMini* eingebunden 8. Wir benötigen zwar nur die TX-Funktion von *SoftwareSerial*, allerdings muss dennoch ein Pin für RX angegeben werden. Hier haben wir in Zeile 4 des abgebildeten Listingausschnitts den unbenutzten Pin 6 angegeben.

Im *setup* wird in der Funktion *initMp3(void)* mit dem Befehl *mp3.volume(15)* die Lautstärke auf 50 Prozent des Maximums eingestellt 9. Ohne diesen Aufruf würde das Modul leider mit der größtmöglichen Lautstärke 30 starten, was so gut wie immer viel zu laut ist. Die damit erzielte Lautstärke ist vor allem vom Wirkungsgrad des Lautsprechers abhängig. Falls Ihnen das Ergebnis zu leise oder immer noch zu laut ist, können Sie den Zahlenwert entsprechend ändern.

Zufällige Song-Auswahl

Nachdem der Trigger durch den Seifenspende erkannt wurde, wird durch den Befehl *mp3.play(songNumber)* ein zufälliges ausgewähltes

8 Die benutzten Bibliotheken

```
01 #include "SoftwareSerial.h"
02 #include "DFRobotDFPlayerMini.h"
03 #define NUMBER_OF_SONGS 10
04 SoftwareSerial mySoftwareSerial(6, 9); // RX, TX on Arduino
05 DFRobotDFPlayerMini mp3;
06 bool randomGeneratorInitialized = false;
```

9 MP3-Player-Initialisierung

```
01 void initMp3(void)
02 {
03     mySoftwareSerial.begin(9600);
04     if (!mp3.begin(mySoftwareSerial, false, false))
05     {
06         //Use softwareSerial to communicate with mp3.
07         Serial.println(F("Unable to begin:"));
08         Serial.println(F("1.Please recheck the connection!"));
09         Serial.println(F("2.Please insert the SD card!"));
10         while(true);
11     }
12     Serial.println(F("Delay for DFPlayer init ..."));
13     delay(3000);
14     Serial.println(F("DFPlayer Mini online."));
15     mp3.setTimeout(500); //Set serial communication time out 500ms
16     mp3.volume(15); //Set volume value
17 }
```

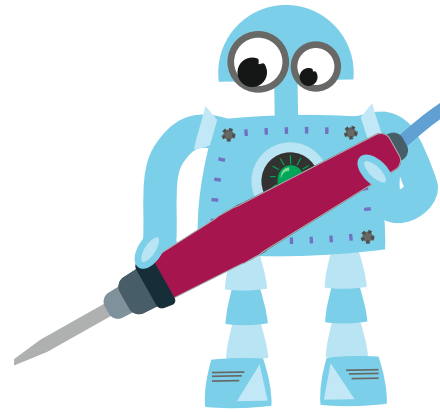
10 Zufallsgesteuerte Song-Auswahl

```
01 if(!randomGeneratorInitialized)
02 {
03     randomSeed(millis()); // use elapsed time as first start value for
    random
04     randomGeneratorInitialized = true;
05 }
06 songNumber = random(1, NUMBER_OF_SONGS+1);
07 mp3.play(songNumber);
08 /*
09 digitalWrite(BuzzerPin,LOW);
10 delay(100);
11 digitalWrite(BuzzerPin,HIGH);
12 */
13 for(int i=Zeit;i>0;i=i-1)
14 {
15     DisplayValue = i;
16     delay(1000);
17 }
18 DisplayValue = 0;
19 mp3.stop();
```

Lied abgespielt, siehe Zeile 7 in Listing 10. Hierfür wird die *random*-Funktion auf dem Arduino verwendet, die hier einen Wert zwischen 1 und *NUMBER_OF_SONGS* liefert. Dieser *define* (er steht in der Zeile 3 von Listing 8) muss ganz oben im Sketch entsprechend an die Anzahl Ihrer Lieder angepasst werden, die auf die Micro-SD-Karte kopiert wurden. Sobald die erste Triggerung nach dem Einschalten erkannt

wurde, wird mit *randomSeed(millis())* der Zufallsgenerator mit der bis dahin abgelaufenen Zeit in Millisekunden initialisiert.

Die Songs auf der Speicherkarte müssen sich im Verzeichnis *mp3* befinden und die Namen mit einer 4-stelligen Nummer beginnen (*0001xxx.mp3* usw.). Nachdem der Countdown abgelaufen ist, wird die Soundausgabe mit *mp3.stop()* gestoppt. —hgb



Mobiltelefone und Mobilfunk

In diesem Artikel geht es um ganz alltägliche Technik: Tragbare Telefone, beziehungsweise Handys oder Smartphones. Damit du weißt, was du da eigentlich benutzt, erzählen wir dir, wie Mobiltelefone entstanden sind und zeigen, aus welchen Komponenten sie bestehen. Dafür stellen wir ein paar DIY-Mobiltelefone vor, die du auch selbst bauen kannst.

von Elke Schick



Mobile Telefone gehören inzwischen so sehr zum täglichen Leben, dass es beinahe unvorstellbar erscheint, dass sich bis vor gut zwanzig Jahren alle Bewohner eines Haushalts ein einziges Telefon teilten. Und nicht nur das, wenn man keinen Anrufbeantworter besaß, konnte man damals ganze Anrufe verpassen und nie von ihnen erfahren. Oder man wusste, wenn das Telefon klingelte, nicht, wer da gerade anrief. Ganz zu schweigen davon, wo man überall nicht erreichbar war. Nirgendwo eigentlich, sobald man das Haus verließ.

Spätestens seit Beginn des aktuellen Jahrtausends kommunizieren aber die allermeisten von uns über Mobilfunk mit dem Telefonnetz, versenden und empfangen Textnachrichten und nehmen nicht ab, wenn auf ihrem Display keine Telefonnummer angezeigt wird. Smartphones, die seit rund 10 Jahren auf dem Vormarsch sind, können sich nicht nur mit dem Telefonnetz, sondern auch noch mit dem Internet verbinden. Eigentlich müssten wir sie Taschencomputer mit Telefonfunktion nennen, wenn wir mal ehrlich sind.

Geschichte des Mobilfunks

Angefangen hat das alles 1926 mit einem Telefondienst für Fahrgäste der ersten Klasse in Zügen zwischen Hamburg und Berlin. Über diesen Zugpostfunk konnten Passagiere über das Postnetz Telefonate ins Festnetz führen. Die Verbindung mit dem Festnetz wurde über die parallel zur Zugstrecke verlaufenden Telegraphenmasten hergestellt. Die zwei Zugtelefone waren durch eine Antenne auf dem Dach zweier Waggons mit den Masten verbunden. Über die Festnetze der beiden Städte an den Enden der Zugstrecke konnten Verbindungen nach ganz Deutschland hergestellt werden.

Nach den Zugtelefonen kamen die Autotelefone. Ab 1946 war es in den USA möglich, Autos mit fest installierten Telefonen zu kaufen, die über mobile Telefonnetze von der Illinois Bell Telephone Company in St. Louis und Chicago mit dem Festnetz verbunden werden konnten. Ab 1958 konnte man solche Telefone auch in Deutschland nutzen. Ihr Preis lag bei etwa der Hälfte dessen, was man bereits für das Auto zahlen musste. Das Mobilfunk-Netz, das sie nutzten, nannte man das A-Netz. Die Telefonate aus diesem

Kurzinfo

- » Entstehung des Mobilfunks
- » Hardware für DIY-Mobiltelefone

Checkliste



Zeitaufwand:

3 bis 5 Stunden



Kosten:

circa 100 Euro

Mehr zum Thema

- » Daniel Bachfeld, Radiosender bauen, Make 4/19, S. 16
- » Matthias Wendt, 100 Jahre Rundfunk – von Hertz bis Gigahertz, Make 4/19, S. 8



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

Material

- » Mikrocontroller Arduino Uno für das Arduino-Phone, ATmega 1284P für das DIY Cellphone oder Raspberry Pi B für das PiPhone
- » GSM-Modul für den SIM800-Standard
- » GSM-Klebeantenne für Mobilfunk
- » SIM-Karten-Halter
- » Display LC- oder TFT, optional mit Touch-Funktion
- » Zifferntasten falls das Display keine Touch-Funktion hat
- » Ein-/Ausschalter
- » Soundkarte falls die entsprechenden Funktionen nicht in das GSM-Modul integriert sind
- » Mikrofon Electret-Kondensator mit Anschlussleitungen oder Lötkontaktstiften
- » Lautsprecher 8 Ohm
- » Akku 3,7V, 1200 oder 1300mAh
- » Platine nur für das DIY Cellphone
- » Gehäuse optional, je nach Projekt, kann 3D-gedruckt oder gefräst werden

Werkzeug

- » Lötkolben
- » Fräse optional
- » 3D-Drucker optional



Das Zugtelefon befand sich in einem Schreibabteil, in dem eine Zugsekretärin arbeitete.

Bild: Reinhard Dietrich CC0 01 Universal Public Domain Dedication



Bild: svlambo CC Attribution 2.0 Generic License

Ein Autotelefon in einem 1959er Ford Fairlane 500

Netz wurden „handvermittelt“, das heißt, man erreichte von den Telefonen aus eine Person in der Telefonzentrale, die die Verbindung herstellte. Erst ab 1972 konnte man mit dem neuen B-Netz Selbstwählverbindungen herstellen.

Wirkliche Mobiltelefone - also tragbare Telefone, ohne Auto dazu - konnten ab 1985 mit dem C-Netz genutzt werden. Das C-Netz war wie das A- und das B-Netz analog, aber es bestand aus wesentlich kleineren Funkzellen als die Vorgängermodelle. Wegen der damit verbundenen kürzeren Funkstrecken brauchten die Telefone mit diesem Netz eine wesentlich geringere Sendeleistung. Außerdem war es das erste Netz, bei dem die Telefonnummer nicht über das Telefon selbst, sondern über eine Karte vergeben wurde, die man in das Telefon steckte.

Das ab 1992 in Deutschland verfügbare digitale D-Netz wurde nach dem so genannten GSM-Standard entwickelt. GSM ist die Abkürzung für „Global System for Mobile Communications“ - es ist ein Mobilfunkstandard, der von einem Komitee beim Europäischen Institut für Tele-

kommunikationsnormen (ETSI) entwickelt wurde. So sollte gewährleistet werden, dass es in ganz Europa ein einheitliches Funknetzwerk gab.

Wie Mobilfunk funktioniert

Von den Telegraphen über die Festnetz-Telefone bis zum Mobilfunk beruhen alle Formen der Kommunikation über große Distanzen auf der Erzeugung und Ausrichtung von elektromagnetischen Wellen. Diese Wellen können in unterschiedlichen Frequenzen erzeugt werden. Die Einheit für die Frequenzen heißt Hertz. Damit sich die unterschiedlichen Arten von Funk nicht in die Quere kommen, hat jede Anwendung ihren eigenen Frequenzbereich, genannt Frequenzband. Diese Frequenzbänder sind wiederum in Kanäle unterteilt.

Elektromagnetische Wellen können sich nicht unendlich weit verbreiten und werden mit steigender Entfernung zum Sender schwächer. Damit Mobilfunknetze möglichst flächendeckend sein können, beste-

hen sie aus vielen kleinen Zellen. In jeder Zelle gibt es eine so genannte Basisstation beziehungsweise Antenne, die Mobilfunksignale in der direkten Umgebung empfängt und über das Netzwerk des jeweiligen Anbieters dorthin weitergibt, wo sie empfangen werden sollen. Für das Senden der Signale vom Sender zur Basisstation (Uploading) und wiederum von der Basisstation zum Empfänger (Downloading) werden verschiedene Kanäle innerhalb der Frequenzbänder genutzt.

Wenn du mit deinem Mobilfunkgerät eine Telefonnummer wählst, wird diese zuerst in Funksignale umgewandelt. Dasselbe gilt natürlich auch für die Gespräche selbst. Was du sagst, wird von dem Mikrofon in deinem Handy aufgenommen. Diese analogen Daten müssen zuerst in digitale Daten umgewandelt werden. Für die Übertragung der Daten werden so genannte Sprachcodecs eingesetzt. Codecs sind Algorithmenpaare, die die analogen Signale auf digitale Symbole abbilden. Außerdem dienen sie dazu, die Datenmenge zu reduzieren. Die digitalen Sprachdaten sollen für die Übertragung

so weit wie möglich komprimiert werden. „So weit wie möglich“ bedeutet in diesem Fall, dass die Verständlichkeit der Sprache bei der Übertragung erhalten werden soll. Dass das nicht immer gelingt, hast du bestimmt schon einmal erfahren, wenn du eine schlechte Verbindung hattest.

Neben der Sprache übertragen Mobiltelefone auch Daten. Das sind SMS und andere Chatnachrichten, der Standort, Webseiten et cetera, et cetera. Mit der Einführung des D-Netzes wurde erstmals auch das Versenden von Textnachrichten mit Telefonen möglich. Dass das Senden von SMS beliebter sein würde als das Telefonieren war für die Hersteller eine Überraschung. Bei den auf GSM aufbauenden Mobilfunktechniken UMTS und 4G bauten sie daher besonders die Kapazitäten

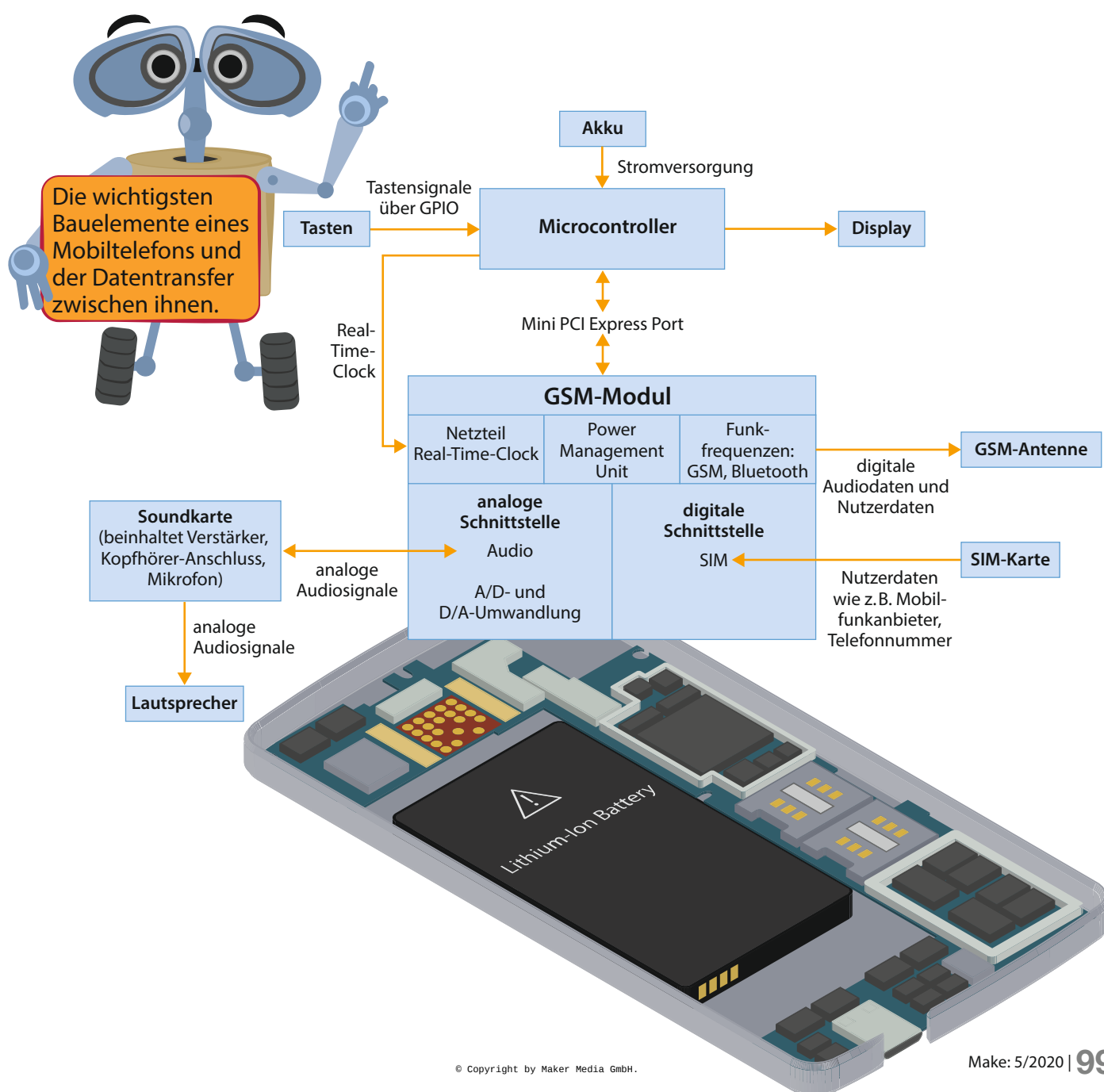
der Datenübermittlung aus und ermöglichen den Zugang zum Internet.

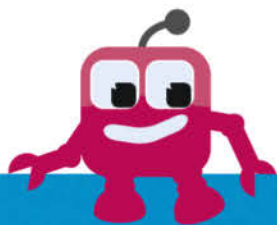
DIY-Telefone

Der beste Weg, mehr über Mobiltelefone zu lernen, ist, selbst eines zu bauen. Dazu gibt es verschiedene Möglichkeiten und Projekte. Sowohl auf der Basis eines Arduinos (Arduin-O-Phone) als auch auf der Basis eines Raspberry Pis (PiPhone) gibt es DIY-Phones. Eines der ersten Selbstbau-Mobiltelefon-Projekte war das DIY Cellphone von David Mellis vom MIT. Von den hier vorgestellten Projekten ist es mit Abstand das komplizierteste, das am meisten Vorwissen und Erfahrung voraussetzt. In den Links zu diesem Artikel findest du die entsprechenden Dokumen-

tationen, mit deren Hilfe du die Projekte nachbauen kannst.

Eine weitere Option ist Ringo, das DIY-Phone-Kit der Firma Circuitmess. Hier erhältst du alle notwendigen Komponenten inklusive einer ausführlichen Anleitung. Mit allen Projekten erhältst du Mobiltelefone, mit denen du telefonieren kannst. Das Ringo ist außerdem internetfähig. Es ist ein Komplettpaket, mit dem du auch die Programmierung der Mobiltelefon-Anwendungen lernen kannst. Für Anfänger ist es mit Sicherheit am Besten geeignet. Das PiPhone und das Arduin-O-Phone sind einander relativ ähnlich, wobei das Arduin-O-Phone besser dokumentiert ist. Anhand des Ringos zeigen wir die Bauelemente, die du für ein DIY-Mobiltelefon brauchst. —esk





Vorderseite

Das Display

Auf dem Display wird dir angezeigt, was du gerade mit dem Telefon machst. Bei den Projekten, die mit Tasten ausgestattet sind, werden LC-Displays verwendet. Alle Eingaben, die du mit den Tasten machst, werden über den Mikrocontroller in Befehle für das Display und die Netzwerkkarte umgewandelt.

Beim Arduin-O-Phone und dem PiPhone kommen TFT-Touch-Displays zum Einsatz. Wie bei Smartphones auch bedienst du dein Telefon dann über Tasten, die auf deinem Display angezeigt werden. Über die Wahl des Displays kannst du also entscheiden, ob du Hardware wie Tasten oder einen Joystick zusätzlich einbauen willst oder nicht.

Die Tasten

Wenn du kein Touch-Display verwendest, musst du dein Telefon mit Tasten ausstatten. Du brauchst einen Ein-/Aus-Schalter und Tasten, um dich durch die Software-Menüs zu bewegen. Beim Ringo sind das der Joystick und die A- und B-Taste. Dann brauchst du noch Tasten zum Eingeben von Zahlen und Buchstaben.

Soundkarte

Um die analogen Signale vom Mikrofon zu digitalisieren, brauchst du eine Soundkarte. Beim Ringo sind das Mikrofon und der Kopfhörer-Anschluss in die Soundkarte integriert.

Der Controller

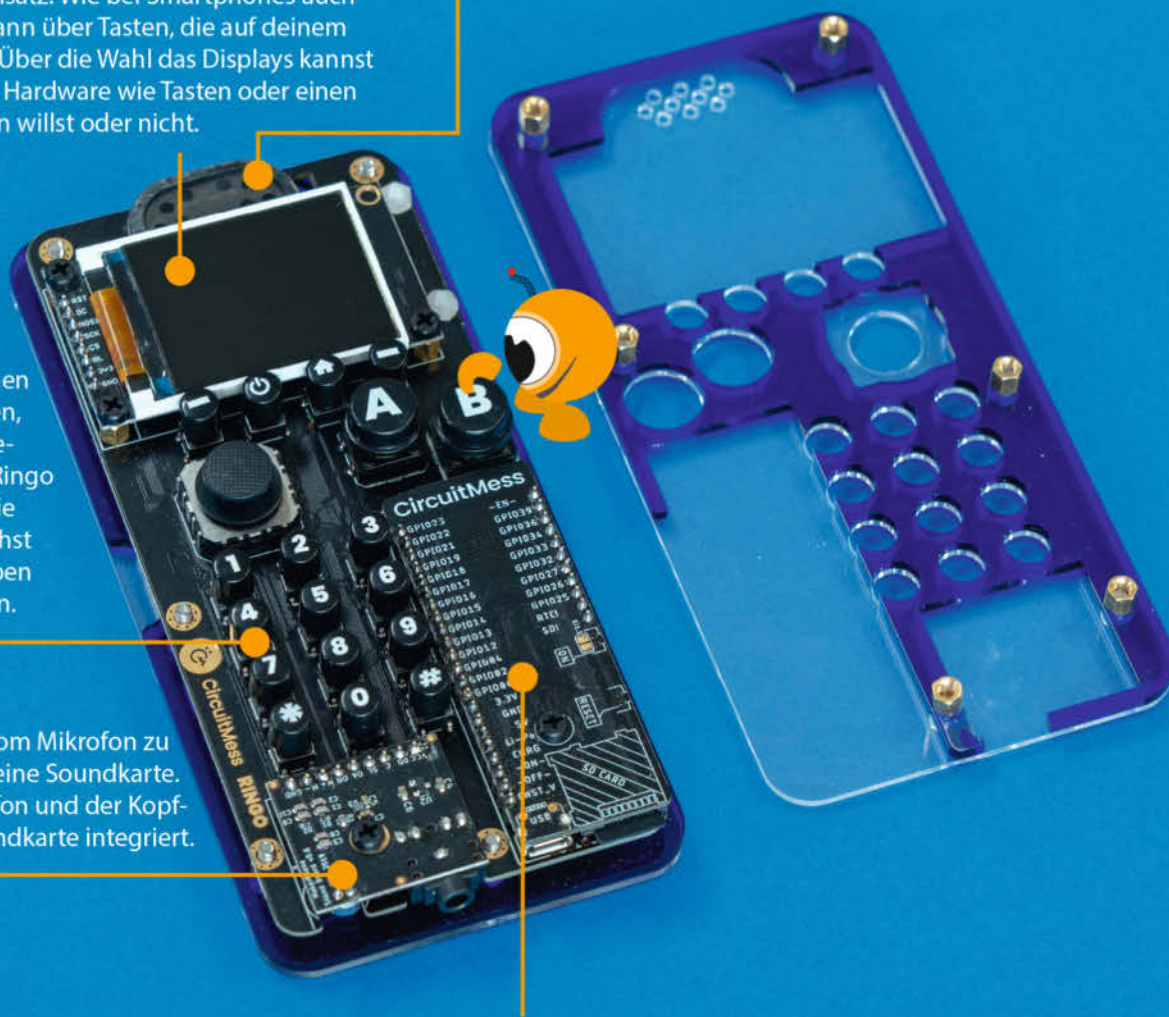
Damit dein Telefon seine einzelnen Bauteile miteinander verbinden und mit anderen Geräten kommunizieren kann, braucht es ein Gehirn. Dafür benutzt du einen Mikrocontroller wie zum Beispiel den Raspberry Pi, den Arduino oder den ATmega.

Der Mikrocontroller ist die Schaltzentrale deines Telefons. Er wird immer, wenn du es benutzt, aktiviert. Sobald

du Tasten drückst oder über das Touchpad eine Telefonnummer eingibst, übersetzt er deine Eingaben in Funksignale und sorgt dafür, dass sie über die GSM-Antenne ausgesendet werden. Außerdem übermittelt es die gedrückten Ziffern an das Display, so dass du sehen und überprüfen kannst, welche Telefonnummer du gewählt hast. Wenn die Verbindung hergestellt ist, aktiviert der Controller das Mikro-

Lautsprecher

Damit du bei deinen Telefonaten auch hören kannst, was gesagt wird, brauchst du einen Lautsprecher. Für ein Mobiltelefon reicht ein Kleinlautsprecher, den es für etwa 5 Euro im Elektronikhandel gibt.



fon und den Lautsprecher, so dass du dein Telefonat führen kannst.

Für die hier gezeigten mobilen Telefone wurden verschiedene Mikrocontroller benutzt. Das PiPhone beruht auf dem Raspberry Pi Model B, das Arduin-O-Phone auf dem Arduino Uno, das Ringo auf einem ESP32 und das DIY Cellphone auf einem ATmega128.

Rückseite

GSM-Antenne

Für das Senden und Empfangen von Daten brauchst du noch eine passende Antenne. GSM-Antennen zum Aufkleben gibt es im Elektronikhandel.

GSM-Modul

Das GSM-Modul stellt die Verbindung zum Mobilfunk her. In den meisten Fällen sind diese Module mit GSM dafür UMTS und LTE kompatibel. Alle Algorithmen für das Senden und Empfangen von Daten wie SMS und GPRS sind auf diesem Modul gespeichert. Die GSM-Module sind auf einen bestimmten SIM-Standard ausgelegt. Beim Ringo und beispielsweise dem Adafruit Fona GSM-Breakout ist es der aktuelle SIM800-Standard.

SIM-Karten-Halter

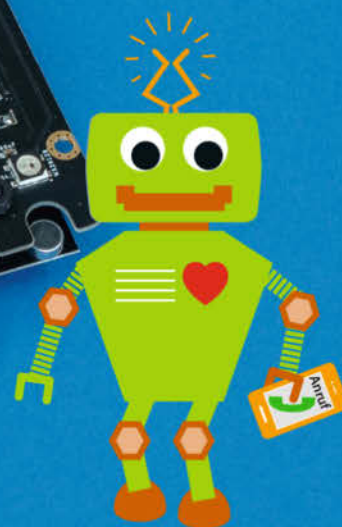
Um über den Mobilfunk zu telefonieren, brauchst du immer eine SIM-Karte. SIM steht für „Subscriber Identity Module“, diese Chipkarte enthält die Informationen zur Identifikation des Benutzenden. Dazu gehören die Telefonnummer und der Mobilfunkanbieter, dessen Netzwerk genutzt wird.

Akku

Für die Stromversorgung des Mobiltelefons brauchst du einen Akku. Hierfür reicht ein LiPo-Akku mit 3,7V und ungefähr 1200mAh.

Platine

Um alle Bauteile miteinander kombinieren zu können, brauchst du in fast allen Fällen eine Platine. Auf der Platine ist vorgegeben, welches Bauelement, wo angebracht wird und alle Anschlüsse und Verbindungen, die die Bauteile benötigen, sind dort bereits angebracht. Das PiPhone und das Arduin-O-Phone kommen ohne Platine aus und beim Ringo ist die Platine im Kit enthalten. Bei diesem Projekt werden die Bauteile mit Kabeln verbunden und mit Kabelbindern zusammengehalten.



Wearables aus der KOBA Maßschneiderei

In der KOBA Maßschneiderei fertigte das Kollektiv KOBAKANT ein Jahr lang individuelle Wearables an. Entstanden sind dabei viele unglaubliche Kleidungsstücke – von intelligenten Prothesen bis zu atmenden Jacken.

von Hannah Perner-Wilson und Mika Satomi



Bild: Xavier Antoinet

Die Koba Maßschneiderei für E-Textilien tauchte ganz plötzlich in Berlin Kreuzberg auf und blieb für ein Jahr. Der kleine Laden wurde vom Künstlerinnen-Kollektiv Kobakant von Dezember 2017 bis Januar 2019 geführt. Kobakant, das sind Mika Satomi und Hannah Perner-Wilson. Das Projekt wurde zum Teil aus Mitteln des EU-Innovationsprogramms „Horizont 2020“ finanziert.

Ein Jahr lang konnte jeder in unseren Laden spazieren und ein E-Textil-Projekt in Auftrag geben. E-Textilien sind Kleidungsstücke, die mit Elektronik versehen neue Funktionen erhalten und interaktiv werden. Für uns gab es in der Schneiderei nur eine Bedingung: Das Projekt musste persönlich sein, ein Bedürfnis des Trägers erfüllen. Daher durfte es nur für den Auftraggeber selbst sein. Aus diesem Grund hatte unsere Koba Maßschneiderei auch keinen Online-Shop. Alles sollte sich im physischen Raum abspielen, von der Bestellung bis zur Anprobe.

In dieser experimentellen Schneiderei wollten wir ergründen, welche individuellen Bedürfnisse und Anforderungen Menschen an Wearables stellen – und wie sie sich erfüllen lassen. Wir wollten nicht einfach nur ein neues Produkt auf den Markt werfen, sondern E-Textilien personalisieren und die Wünsche der Menschen ins Zentrum stellen.

Was wurde in der Schneiderei gefertigt?

In jenem Jahr sind in der Koba-Maßschneiderei viele E-Textilien entstanden. Wir verwenden für unsere Arbeit ganz unterschiedliche Techniken. Meist kombinieren wir klassische Stoffbearbeitungs-Techniken wie Nähen, Sticken, Stricken, Weben, Häkeln, Filzen und Flechten mit moderner Elektronik.

Um E-Textilien herzustellen, kann man zum Beispiel elektrisch leitenden Stoff mit einer Schere, einem Schneideplotter oder einem Lasercutter in jede beliebige Form schneiden und auf andere Stoffe aufbügeln. Einige leitfähige Fäden und Gewebe haben einen so hohen Kupfergehalt, dass man direkt auf ihnen löten kann. Man kann auch vernähbare, flexible Platinen herstellen: Dafür schneidet man Kaptonfolie in Form und fügt Verbindungen aus Kupferband hinzu. Bei Kobakant arbeiten wir mit einer Kombination aus Textilverkzeugen (Nähmaschinen, Strickmaschinen, Bügeleisen, Nadel und Schere) und Elektronikwerkzeugen (LötKolben, Multimeter, Mikrocontroller).

Koba ist zwar ein Projekt von uns, Kobakant, doch wir hatten in unserer Maßschneiderei wunderbare Künstler zu Gast. Ben Munday entwickelt beruflich Schnittmuster und hat uns dabei unterstützt, für die verschiedenen Projekte Schnittmuster zu erstellen. In der Koba Maßschneiderei hat er uns beigebracht,



Bild: Irene Posch

Die Koba Maßschneiderei

wie wir Schnittmuster selbst entwickeln und wie aus den abstrakten Formen ein Kleidungsstück entsteht. Ohne ihn wären wir wohl keine echte Maßschneiderei geworden. Im Mai ist dann auch unsere Häkel-Meisterin Mariam Arid zu uns gestoßen. Sie hat den Löwenanteil an Dominique's Rüstung geleistet – eines der beiden Projekte, die wir hier vorstellen.

Kurzinfo

- » Was ist die Koba Maßschneiderei?
- » Wie entstehen E-Textilien?
- » Materialien und Techniken

Mehr zum Thema

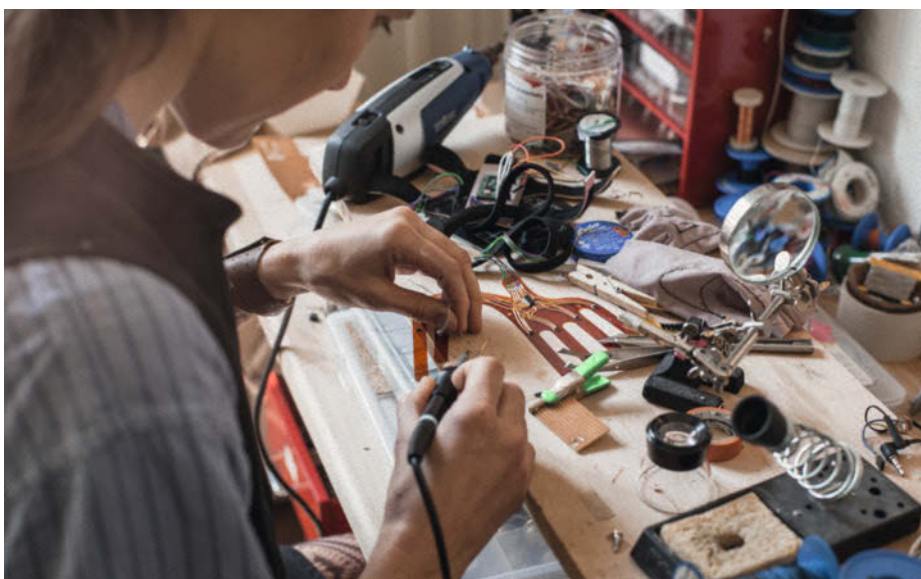
- » Eva Ismer, Glasfasern in Textilien, Make 6/2019, S. 50
- » Helga Hansen, Schöner leuchten, Make 04/2015, S. 20

Alles zum Artikel
im Web unter
make-magazin.de/xm7x



Bilder auf dieser Seite: Aimee Shirley von Freunden

Kreative Atmosphäre in der Schneiderei



Koba verbindet Handarbeit mit Elektronik.

Materialien für E-Textilien

Leitende Garne und Gewebe können Strom leiten. **Leitendes Garn** besteht zum Beispiel aus silberbeschichteten Kunstfasern oder aus einem dünnen Metallfaserstrang, der mit Kunstfaser überzogen ist. **Leitende Gewebe** können synthetische Gewebe mit einer Metallbeschichtung (wie Kupfer, Silber oder Nickel) sein. Andere bestehen aus leitendem Faden, der zusammen mit nicht leitendem Faden zu Stoff gewebt wurde. Manchmal wird auch leitender Faden mit nicht leitendem Faden zu Strickstoffen verstrickt.

Widerstandsgarne und -gewebe zeigen unter Druck einen piezoresistiven Effekt. Man kann sie verwenden, um Textilsensoren herzustellen. Zur Herstellung von

Widerstandsgewebe werden die Eigenschaften des Ausgangsmaterials verändert, in diesem Fall die elektrische Leitfähigkeit. Das geschieht durch die Dotierung des synthetischen Gewebes mit einem organischen Polymer. Diese Gewebe haben einen zunehmenden Widerstand über die Entfernung und eignen sich hervorragend zur Herstellung von Potentiometern und Positions-Sensoren. **Widerstandsgarn** wird aus einer Mischung von Stahlfasern und nicht leitenden Fasern gesponnen.

Wenn man versteht, wie elektrische Standardkomponenten funktionieren, kann man sie mit leitfähigen Textilmaterialien neu erfinden. Viele Schalter werden zum



Bild: Thomas Straub

Leitende Garne und Bänder

Beispiel aus zwei Leitern hergestellt, die bei Aktivierung des Schalters miteinander Kontakt haben – das lässt sich mit E-Textil-Materialien unkonventionell umsetzen.

Ein Sensor für eine Fingerprothese

Tom hat unsere KOBASchneiderei kontaktiert, kurz nachdem er bei einem Unfall die Fingerkuppe seines linken Ringfingers verloren hatte. Als er nach dem Unfall seine erste Prothese erhielt, war er ziemlich enttäuscht: Es gab keine Möglichkeit, sie nach seinen Vorstellungen seinen Bedürfnissen anzupassen. Daraufhin beschloss er, seine eigene DIY-Prothese herzustellen.

Tom kündigte sich per Mail bei uns an. Er ließ uns wissen, dass er sich von uns einen Textilsensor für seine selbst hergestellte Fingerprothese wünscht. Und er erzählte uns, dass er sich darüber ärgert, dass die meisten kommerziellen Prothesen patentiert und teuer sind. Tom überlegte, Prothesen selbst herzustellen und daraus ein Open Source-Projekt zu entwickeln.

Zu diesem Zeitpunkt hatte Tom bereits eine eigene experimentelle Fingerprothese mit

einem kleinen Servomotor auf einem 3D-Drucker hergestellt. Sie sollte sich bewegen, wenn er seine Hand öffnet und schließt – genau wie seine anderen Finger. Bei uns hat er sich dann einen textilen Fingersensor bestellt, damit er die Prothese mit der Gelenkbewegung seines verbliebenen Ringfingers steuern kann.

Dieses Projekt befindet sich noch in der Entwicklung. Auftraggeber Tom lebt in Süddeutschland. Wenn wir also einen Prototyp fertig gestellt haben, senden wir ihn per Post an ihn und warten, bis wir ein Feedback von ihm erhalten. Bis dahin wissen wir nicht, ob unsere Ideen ihm gefallen und mit seinem Prothesenfinger kompatibel sind. In naher Zukunft werden wir (hoffentlich) erleben, wie sich seine DIY-Prothese genau so bewegt, wie er es möchte. Und vielleicht können wir sie auch dazu bringen, Dinge zu tun, die andere Finger nicht können.

KOBAs Lösung

Bisher haben wir folgende Lösung für Tom entwickelt: Der *textile bend sensor finger sleeve* ist ein Handschuh für seine linke Hand, in dem ein textiler Drucksensor auf dem mittleren Knöchel seines linken Ringfingers eingebettet ist. Der Sensor erkennt die Beugung der Verbindung durch den Druck auf die Stoffschichten. Der Gewebedrucksensor besteht aus zwei Schichten leitenden Stoffs, zwischen denen sich eine Schicht piezoresistiven Gewebes befindet. Der piezoresistive Effekt beschreibt die Veränderung des elektrischen Widerstands eines Materials durch Druck oder Zug.

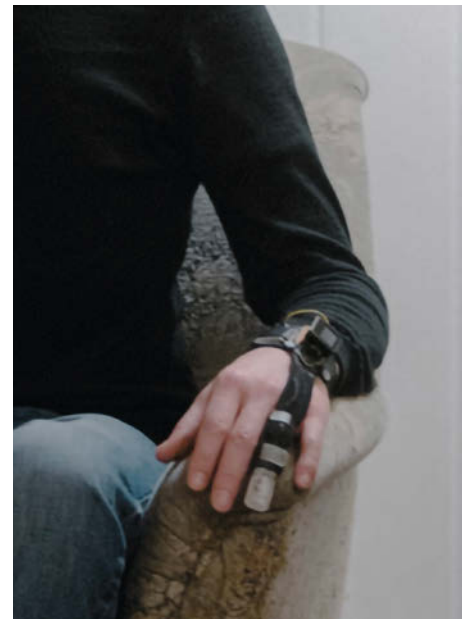


Bild: Xavier Antoinet

Tom trägt seine Prothese mit textilem Biegesensor.

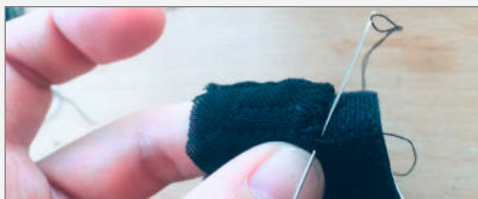
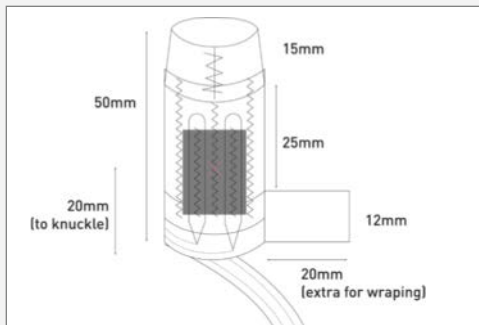
Der Sensor ist mit der am Armband angebrachten *microview* verbunden. Sie besteht aus einem Arduino und einem OLED-Display. Die *microview* liest den Sensor aus, zeigt den Messwert an und steuert den Servomotor. Ein Knopf am Armband ermöglicht es Tom, den Sensor zu kalibrieren. Der Stoffsensor hält auch die 3D-gedruckte kinetische Fingerprothese am Finger und führt die elektrische Verbindung (Leistung, Masse, Daten) zum Servomotor in der Prothese.



Bild: Kobakant

Der textile Sensor mit Kontrollstation *microview*

Making-of



1. Dieses Diagramm zeigt den Aufbau der Fingerhülse des Biegesensors. Er besteht aus zwei Schichten Power-Mesh-Gewebe, die druckempfindliches Widerstandsgewebe der Marke Eeonyx einschließen und mit drei Zick-Zack-Stichen befestigt sind. In die Schichten werden zwei leitende Fäden eingeführt, die als Elektroden dienen, um die Änderung des Eeonyx-Gewebes zu erfassen.

2. Das Power-Mesh-Material und das dehnbare Eeonyx-Widerstandsmaterial ist auf die passende Größe zugeschnitten.

3. Wir schichten Power Mesh, Eeonyx und Power Mesh in drei Lagen aufeinander. Mit einer Nähmaschine heften wir die Schichten im Zickzackstich zusammen. Die mittlere Naht verläuft über die gesamte Länge des Stoffstücks – so können wir vermeiden, dass sich die leitenden Fäden berühren. Die anderen beiden Nähte fixieren später das leitende Garn im Stoff.

4. Hier nähen wir die Kanten des Stoffstücks aufeinander, um einen Schlauch zu formen. An Ober- und Unterkante des Schlauchs befestigen wir eine medizinische Fingerschleife. Diese Schleife sorgt später dafür, dass der Stoffschlauch nicht vom Finger rutschen kann.

5. Wir führen das leitende Garn durch die Kanäle, die wir im dritten Schritt genäht haben. Die Garn-Enden schauen am unteren Rand des Stoffschlauchs raus. Das Garn liegt auf dem Eeonyx-Material auf.

6. Dann fädeln wir das leitende Garn in Paracord-Nylonseil ein – so kann man das leitende Garn isolieren. Das wiederholen wir, bis wir genug Verbindungen haben, um das microview-Armband und den Servomotor von Toms 3D-gedruckter Prothese anzuschließen. Die Nylonstränge flechten wir dann zu einem kompakten Seil.

7. Wir haben mit der gleichen Technik als Alternative auch einen Handschuh aus doppelschichtigem Power-Mesh hergestellt, der den Biegesensor und die Verbindungen enthält.

Eine Liste mit allen verwendeten Materialien finden Sie in den Links in der Kurzinfo.

Tanzoutfit mit Bewegungssensoren

Dominique ist eine Performance-Künstlerin. Als sie auf KOBA zukam, entwickelte sie gerade eine Performance zum Thema Freizeit und erforschte dabei den *Flow*, in den man beim Handarbeiten schnell gerät. Dazu übte sie sich in Entspannungsübungen, um ihre innere Anspannung zu lösen, bis sie ganz weich und „flüssig“ in ihren Bewegungen wurde. Sie wünschte sich von uns ein Kostüm, das den Fluss ihrer Bewegungen unterstreicht. Nachdem wir gesehen hatten, wie sich ihr Körper bewegt, beschlossen wir, unsere Aufmerksamkeit auf ihren Nacken, ihre Hände und den Druck, den ihre Füße auf den Boden ausüben, zu richten.

Während des gesamten Prozesses gab es nur eine Konstante: Alles sollte gehäkelt sein. Dominiques erste Skizze zeigte noch ein gehäkeltes Kostüm, das wie eine altertümliche Kettenrüstung aussah. Wir waren von der Herausforderung, ein komplett gehäkeltes Elektronikprojekt anzugehen, begeistert.

Innovative Kunst oder Omas Topflappen?

In diesem Projekt haben wir viel über Ästhetik diskutiert: Wie kann man etwas häkeln, das nicht automatisch nach Oma aussieht? Wie können wir Häkeltechniken entwickeln, deren Ergebnisse vielleicht technisch, organisch, biologisch und vor allem ungewöhnlich sind? Nachdem wir viele Treffen lang über Kunst, Handarbeit und organische Formen gesprochen haben, sind wir bei einer gemeinsamen Inspiration gelandet: Schleimpilze. Wir wollten ein gehäkeltes Set aus Umhang, Handschuhen, Schuhen und Haube anfertigen, dessen Aussehen an die organischen Muster von Schleimpilzen erinnert.

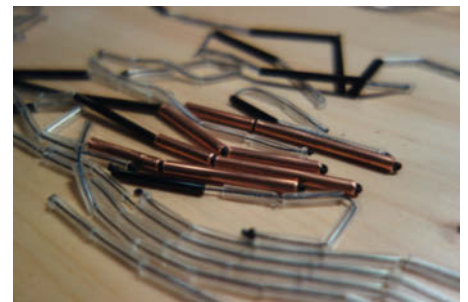
Häkeln ist eine dieser Handarbeiten, für die es keine Maschine gibt. Daher waren wir dank-



Schleimpilz-Häkelpfen

bar, dass Mariam dabei war. Mariam ist eine Freundin von uns, die hervorragend häkeln kann - und kurz zuvor aus Aleppo nach Deutschland geflohen war. Während der Arbeit an diesem Projekt lernte sie Deutsch, was den Prozess sehr amüsant gemacht hat. Was Mariam wohl gedacht hat, wenn wir ihr mit Händen und Füßen unsere verrückten Experimente erklärt haben? Es war manchmal nicht einfach, sie zu überzeugen, dass sie möglichst unsauber und organisch häkeln soll - wo sie doch so wunderbar gleichmäßig häkeln kann. „Was trägt Dominique eigentlich drunter?“ war eine Frage, die Mariam wirklich unter den Nägeln brannte.

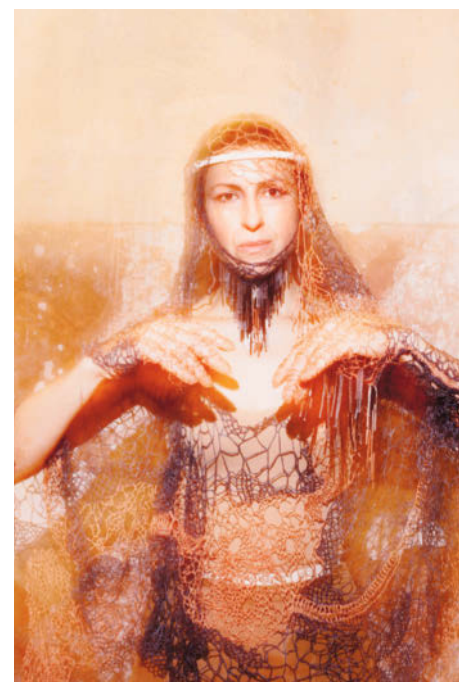
Doch das Aussehen war nicht die einzige Herausforderung: Wir mussten auch Wege



Kupfer- und Glasröhrchen auf leitendem Garn



Die Bewegungssensoren der gehäkelten Handschuhe



Dominiques Kostüm

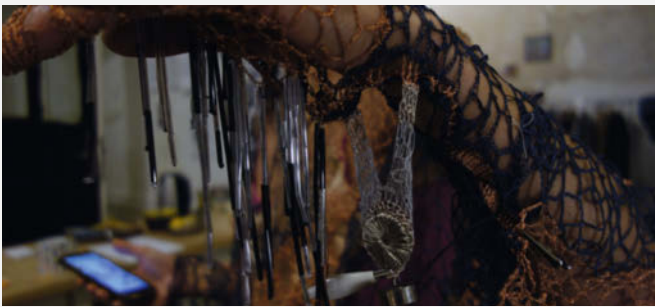
Bilder auf dieser Seite: Kobakant

Making-of

Handschuhe



1. Die kupfernen Fingerspitzen sind unregelmäßig aus festen und doppelten Maschen gehäkelt. So können wir ein organisches Spitzenmuster erzeugen, das aussieht, als wäre der Häkelstoff in einer symbiotischen Beziehung über den Körper gewachsen.



2. Die Kupferfinger gehen in einen blauen Ärmel über. Der Ärmel isoliert zwei Kupferstränge, die zu den perlenbesetzten Sensoren an der Handinnenfläche führen.

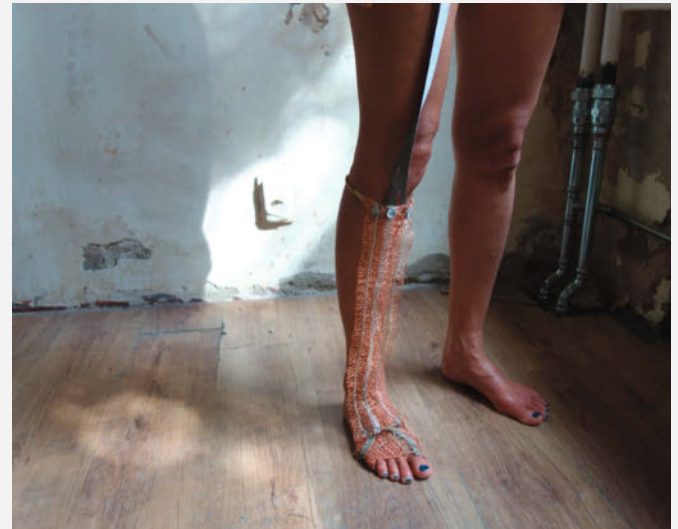


3. Die Perlen baumeln als Bewegungssensor von der Handfläche.

Schuhe



1. Zwei graue Häkelsensorkreise sitzen auf dem Fußballen. Sie erkennen Druckveränderungen, wenn Dominique ihr Gewicht verlagert.



2. Der Fußsensor mit Verbindungswegen, die über das Bein zum WLAN-Gürtel an der Taille führen

finden, um Sensoren zu häkeln. Wie können wir Kontaktschalter und Widerstände konstruieren, mit denen wir die Bewegungen von Dominique durch physische Änderungen im Material des Kostüms erfassen können?

KOBAs Lösung

Der gehäkelte Schleimpilz-Umhang hat einen Durchmesser von 1,5m und ist vollständig aus leitendem und nicht leitendem Garn gehäkelt.

Auf seiner Oberfläche befinden sich Cluster aus Glas- und Kupferperlen, die als Bewegungssensoren fungieren. Die Bewegung wird erkannt, weil die Kupferperlen auf leitendes Garn gefädelt sind. Wenn sich die Perlen bewegen, öffnen und schließen sie ständig den Stromkreis, wie ein Schalter. Je ungleichmäßiger und „gestresster“ Dominiques Bewegungen sind, desto instabiler ist die Verbindung. Je flüssiger sie sich bewegt, desto gleichmäßiger ist die elektrische Verbindung

zwischen den baumelnden Perlen. So entsteht ein Signal, das Dominique bei ihrem Auftritt in Musik und Geräusche umwandeln kann.

Die Handschuhe und die Haube sind ebenfalls mit Perlen-Bewegungssensoren versehen. Diese Sensoren haben wir gefertigt, indem wir ein Stück dünnes Kupferrohr auf das Ende eines versilberten Nylonfadens gefädelt haben. Den Rest des Fadens haben wir mit Hilfe von Glasperlen isoliert. Wenn sich die Kupferröhrchen berühren, schließen sie den

Kontakt zwischen den Häkelverbindungen. Die Verbindung führt zu GND und einem GPIO-Pin. Wenn die Kupferröhrchen sich berühren, zeigt der GPIO-Pin LOW an, ansonsten wird HIGH angezeigt (da der Pin durch einen internen Pull-up-Widerstand nach HIGH gezogen wird).

Unter die Fußballen der Schuhe haben wir aus einem Garngemisch mit 20 Prozent Edelstahl und 80 Prozent Wolle einen Kreis gehäkelt. Dieses spezielle Garn wird beim Zusammendrücken leitfähiger, weil es aus kurzen Stahlfasern besteht, die im entspannten Zustand einen hohen elektrischen Widerstand aufweisen, aber umso leitfähiger werden, je mehr die Fasern im Garn zusammengedrückt werden. So können wir den Druck des Fußes auf dem Boden erfassen. Diese einfache Materialeigenschaft des Garns liefert überraschend konsistente Sensorergebnisse.

Um die Sensordaten auszulesen, haben wir einen WLAN-Gürtel gebaut. Ein kabelloses x-OSC I/O-Board ist an Dominiques Taille angebracht und mit ihrem Outfit verbunden. Es liest die Sensorinformationen aus und sendet sie drahtlos als OSC-Pakete (Open Sound Control) an einen nahe gelegenen Computer. Dieser empfängt die Sensordaten und bereitet sie auf. Dann werden die Signale der Bewegungs-

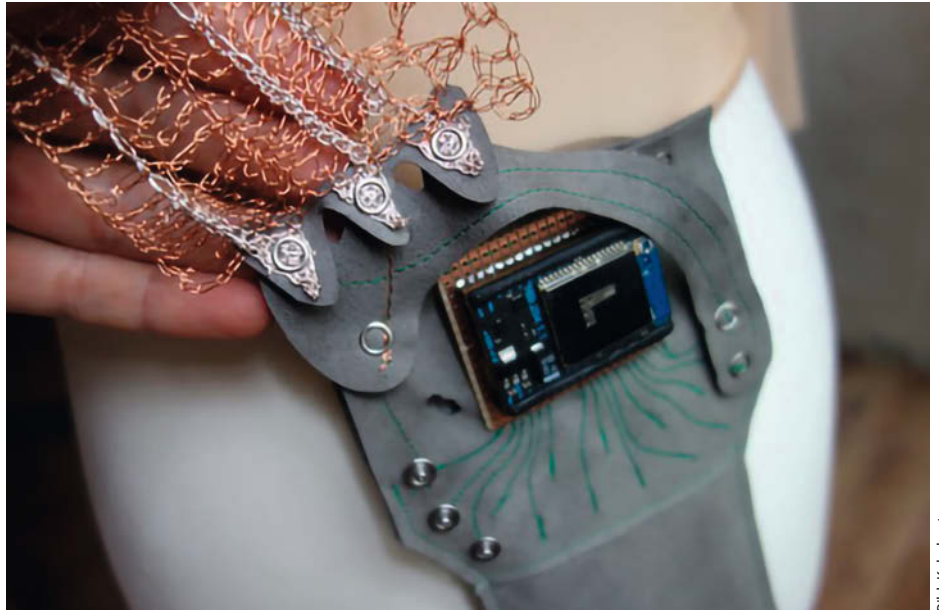


Bild: Kobakant

Die Steuereinheit an der Taille

sensoren in verschiedene Klänge umgewandelt. Der Drucksensor der Schuhe löst beispielsweise ein tiefes, bebendes Geräusch aus.

Erst wenige Tage vor Dominiques Auftritt hatten wir den Umhang, die Haube, die Hand-

schuhe und die Schuhe fertiggestellt. Dominique war begeistert, als sie das gehäkelte Outfit trug: Es fühlte sich an, als sei ihr Körper ein Instrument, auf dem nur sie spielen kann.

Weitere KOBA-Projekte

Insgesamt hatte die KOBA-Maßschneiderei 14 Aufträge innerhalb eines Jahres. Hier sind ein paar weitere Projekte, die in der Zeit entstanden sind:

Geometrische Partyjacke

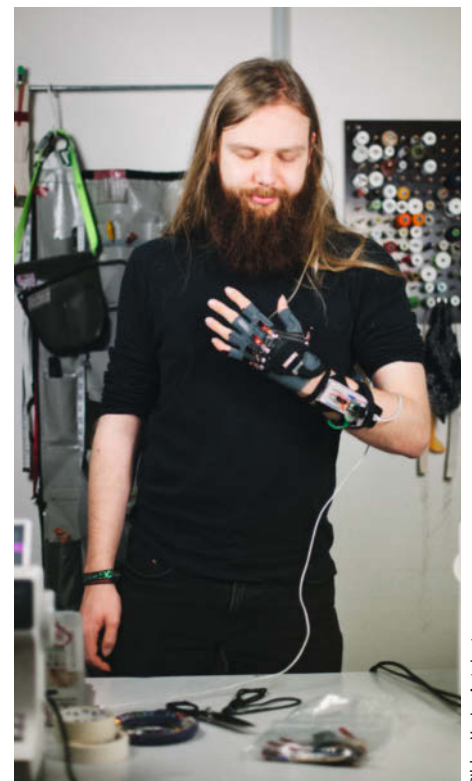
Sebastian wollte seinen Lieblingshoodie zu einer Partyjacke umfunktionieren. Sie sollte mit leuchtenden Polygonen überzogen sein, deren Umrisse individuell steuerbar sind. Wir haben seine Lieblingsjacke nachgeschneidert und eine individuelle Platine zur Steuerung des Lichts entwickelt. So konnten wir mit 42 LEDs 42 Glasfasern zum Leuchten bringen. Sebastian kann die Animation über ein *Particle Argon IoT Development Board* in seiner Jackentasche steuern.

Programmierhandschuhe

Arne hat sich bei unserer Schneiderei Programmierhandschuhe „für faule Programmierer“ bestellt. Mit diesen Handschuhen möchte er ohne Tastatur an seinem Computer tippen können – und im Idealfall seine Finger möglichst nur leicht auf und ab bewegen müssen. Für uns war es eine ziemlich große technische Herausforderung, die winzigen Bewegungen seiner Finger exakt zu registrieren.



Sebastians Partyjacke leuchtet in geometrischen Formen.



Bilder: Xavier Antoinet

Arne betrachtet zufrieden seine Programmierhandschuhe.



Wims Weste darf nicht zu sehr ablenken.



Boris leuchtet beim Posaunspielen auf.

Bilder auf dieser Seite: Kobakant

Posaunenjacke

Boris hat bei uns eine Posaunenjacke bestellt. Während er seine Posaune spielt, leuchtet sie mit jedem Ein- und Ausatmen rhythmisch auf. Normalerweise kann man nur erahnen, wie die Posaunisten in ihre Instrumente atmen und so Geräusche erzeugen. Diesen Vorgang visualisiert zu sehen, ist total spannend. In

diesem Outfit tritt er mit einem Straßenmusik-orchester auf.

Erzählerweste

Wim ist ein Geschichtenerzähler. Bei uns hat er eine Weste bestellt, die aufleuchtet, wenn er seine Geschichten erzählt. Dabei sollte die Technik möglichst unsichtbar sein. Obwohl

die Weste die unterschiedlichsten Leuchteffekte anzeigen kann, verwendet er pro Geschichte immer nur einen Effekt. „Sonst starren alle nur auf das Licht und hinterher erinnert sich keiner mehr an die Geschichte“, sagt er. E-Textilien zu Schneidern, die wirklich die Bedürfnisse des Einzelnen erfüllt ist ein friemelig und langwieriger Prozess mit vielen Irrwegen – und geduldigen Kunden. —rehu

In die Cloud – aber sicher

8. Oktober 2020, 11.00 – 12.30 Uhr



Immer mehr Firmen suchen nach Wegen, ihre IT durch stärkere Nutzung der Cloud zu optimieren. Diese nutzen dann fertige Cloud-Dienste (Software-as-a-Service) oder bauen auf Basis-Komponenten eigene Funktionen und Anwendungen (Platform- oder Infrastructure-as-a-Service).

Ziele wie Kosteneinsparung, Flexibilität und Skalierbarkeit lassen sich dabei aber nur erreichen, wenn man die Sache richtig angeht, sich realistische Ziele setzt und die neuen Herausforderungen der Architektur versteht.

Inés und Manuel Atug beschreiben anhand ihrer reichhaltigen Erfahrungen mit Cloud-Projekten nicht nur, was da so typischerweise schief geht, sondern natürlich auch, wie Sie Ihr Cloud-Projekt richtig angehen.

Preis: 149,00 Euro inkl. MwSt.



www.heise-events.de/webinare/cloud-aber-sicher

Automatische Fahrrad-schaltung mit Arduino

Selbst schalten war einmal, auch für Fahrräder gibt es jetzt eine Automatikschaltung. Shift4Me kann man einfach nachbauen und an fast jedes Rad montieren.

von Helga Hansen



Bilder: Jan Oelbrandt

Viele Leute nutzen die Gänge ihres Rads nicht aus und fahren deswegen langsamer oder unbequemer, als sie eigentlich müssten, glaubt der Fahrradfahrer Jan Oelbrandt. Daher hat der Bastler beschlossen, eine automatische Lösung für das Schalten zu schaffen, die einem wirklich alle Arbeit abnimmt. Nur selber lenken und treten muss man dann noch.

Sein Shift4Me-System wiegt nur wenig und kann mit grundlegenden Elektronikkenntnissen und Zugang zu einem 3D-Drucker selbst gebaut und installiert werden. Es soll sowohl mit Ketten- wie auch Nabenschaltungen funktionieren, solange die Schaltung mit einem Schaltzug betrieben wird. Auch Elektroräder können damit ausgerüstet werden. Bei ihnen ließen sich sogar die höchsten Verbesserungen durch automatisches Schalten erzielen, indem die Batterie geschont und der Verschleiß verringert wird, erklärt Oelbrandt.

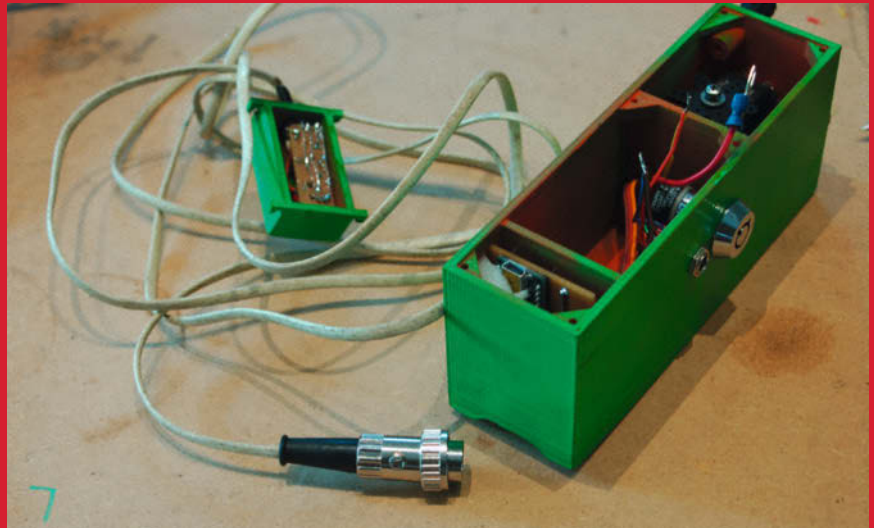
Das System besteht aus vier Komponenten: einem Magnetsensor, der die Trittfrequenz misst, einem Arduino nano für die Steuerung, einem PowerHD 1510MG Servo-Motor, der am Schaltzug zieht sowie einem Akku-Pack. Der Sensor wird in der Nähe der Pedale installiert, während der Arduino, Akku und der Servo in einem 3D-gedruckten Kasten unterkommen, der am Rahmen befestigt wird. Dabei muss auch der Schaltzug ausgetauscht und durch den Kasten geführt werden. So kann der Servo statt der Hebel am Lenker den Schaltzug bewegen. Zusätzlich gibt es einen Knopf, der am Lenker angebracht wird, um das System zeitweilig auszustellen und das Rad in den ersten Gang zurück zu setzen.

Alle Teile, inklusive der 3D-Drucke, kosten circa 150 Euro, so Oelbrandt. Mit dem vorgeschlagenen Akku seien acht Stunden Fahrzeit möglich. Hat das System keinen Saft mehr, bleibt das Rad im letzten benutzten Gang stecken. Vorgaben wie die gewünschte Trittfrequenz, die das Shift4Me-System beim Fahren umsetzen soll, können im Arduino-Programm eingestellt werden.

Wer das Gerät nachbauen möchte, sollte sich im Shift4Me-Forum anmelden. Die Anleitung samt 3D-Druckvorlagen, Schaltplan und Programm von Oelbrandt können nach der Anmeldung kostenlos heruntergeladen werden. Sollten beim Nachbau Fragen oder Probleme auftauchen, können sie ebenfalls im Forum gestellt werden. Inzwischen gibt es zusätzlich ein Github-Repository, das von einem Shift4Me-Fan eingerichtet wurde. Einigermaßen wasserfest soll das System übrigens auch sein, solange das Rad nicht ständig draußen steht. Wer das Upgrade wieder loswerden möchte, kann die Shift4Me-Teile übrigens einfach ausbauen und den ursprünglichen Schaltzug wieder einsetzen und nutzen.

—hch

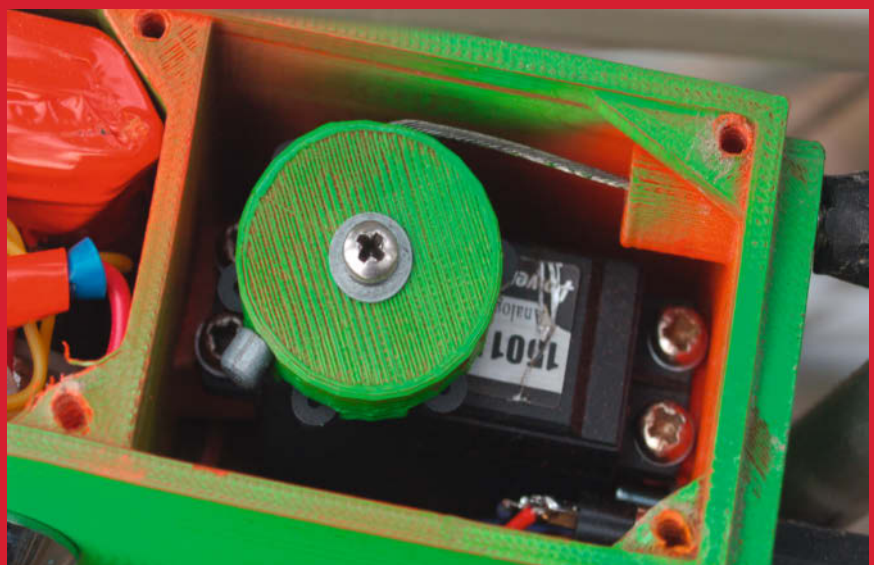
► copynotes.be/shift4me/index.php



Im grünen Kasten werden der Arduino Nano, die Akkus und der Servo untergebracht.



Zum zeitweiligen Ausschalten der Automatik gibt es einen Knopf am Lenker.



Der Servo zieht am Schaltzug, um die Gänge automatisch zu setzen.

Binär-Uhr mit dem Kniwwelino

Bei dieser Uhr mit Mikrocontroller und LED-Streifen ist Kopfrechnen angesagt. Wie spät es ist, wird nur mit Nullen und Einsen dargestellt.

von Christian Moll

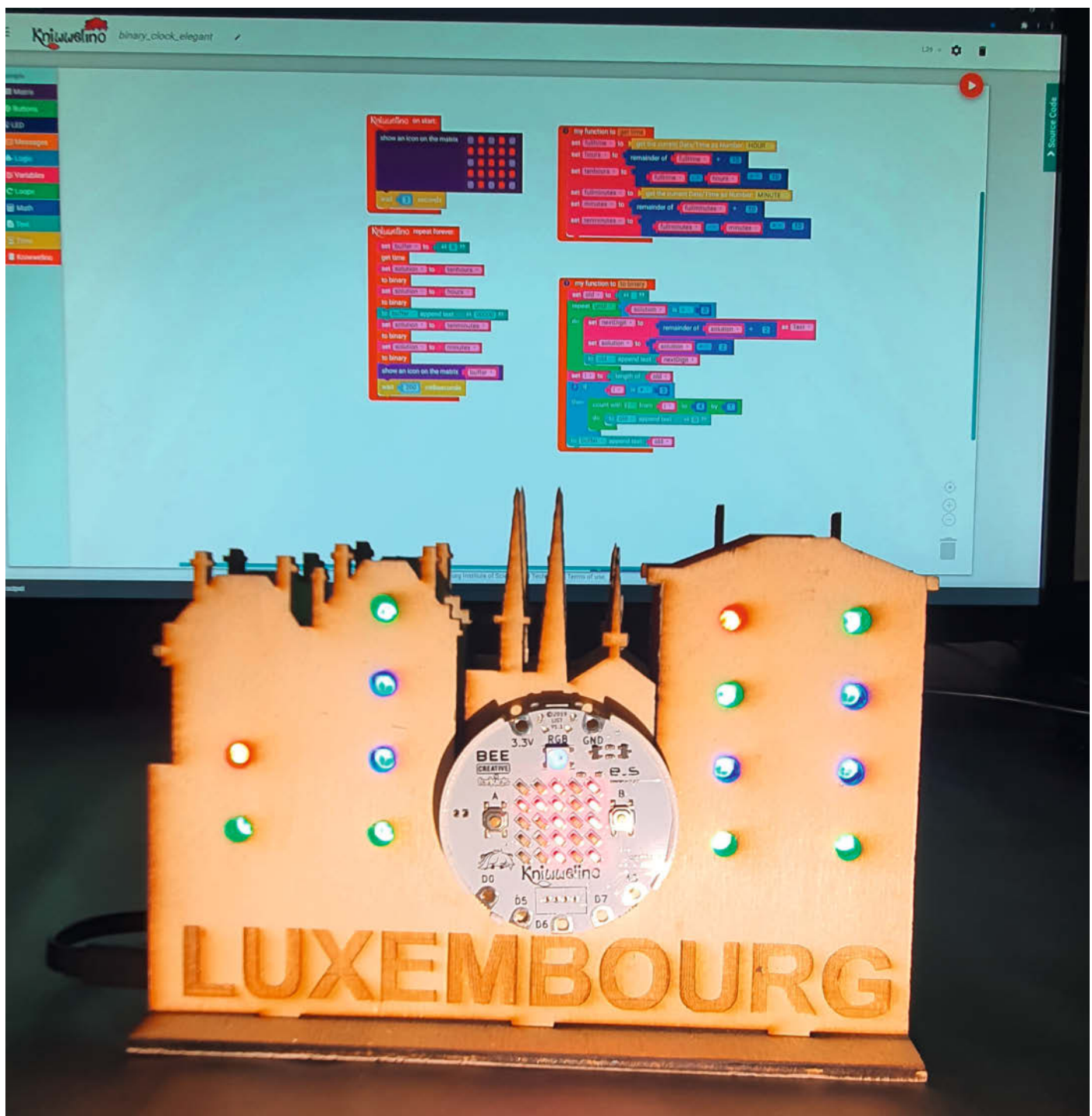


Bild: Tim Schmitz

Eine Binär-Uhr gehört wohl zu den geekigsten Arten, die Uhrzeit anzuzeigen, aber sie ist auch hilfreich, um die elementare Grundlage von allen Computern und Mikrocontroller-Projekten zu erklären und zu verstehen. Es geht um das Bit, die 1 und die 0. Dafür wird die Uhrzeit meist erst in ihre Ziffern und dann in deren Binärwerte zerlegt. Aus 14:52 Uhr wird 14 5 2 und daraus 01 0100 101 0010. Der Unterschied zwischen 1 und 0 kann durch an- und ausgeschaltete LEDs dargestellt werden. Seit Frühjahr 2018 hängt eine große Binär-Uhr in Bahnhof von St. Gallen in der Schweiz, die seither für rauchende Köpfe sorgt.

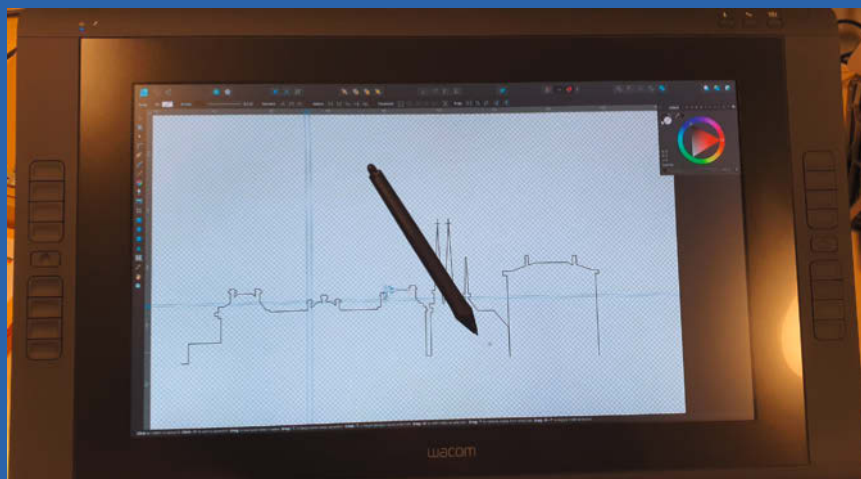
Um unsere Binär-Uhr mit dem Mikrocontroller Kniwwelino nachzubauen, muss man nicht einmal eine Programmiersprache beherrschen. Stattdessen nutzen wir eine grafische Programmierumgebung, in der man Code aus leicht verständlichen Blöcken zusammenschiebt – so erinnert das Programmieren eher an das Bauen mit Lego. Wir haben sogar zwei Varianten gebaut und lassen die Zeit auf dem Kniwwelino oder in der LED-Skyline aus dem Lasercutter anzeigen. Dabei muss man von unten nach oben rechnen. Während die LED-Matrix auf dem Mikrocontroller nur über rote LEDs verfügt, geht es auf dem Streifen bunt zu: Alle Einsen werden auf grün geschaltet, während die Nullen blau angezeigt werden. Um alle LEDs leuchten zu lassen, werden die übrigen Pixel in rot aufgefüllt. Praktisch ist, dass der Kniwwelino, sobald er eine Verbindung zum Internet hat, von einem NTP-Server (Network Time Protocol) die aktuelle Uhrzeit abrufen und sie regelmäßig abgleicht.

Unsere erste Programmvariante teilt die Uhrzeit mit ineinander geschachtelten Wenn-Dann-Abfragen in die einzelnen Ziffern auf und überprüft, ob diese gerade oder ungerade sind. Wesentlich eleganter und viel kürzer wird der Code, wenn man die Zehner- und Einerstellen mit Modulo und Division durch 2 bearbeitet – also die Ziffern solange durch 2 teilt, bis als Rest (Modulo) eine 1 oder 0 herauskommt. Dieses Programm lässt sich mit wenigen Veränderungen nutzen, um sowohl die LEDs der Matrix als auch des LED-Streifens zu schalten.

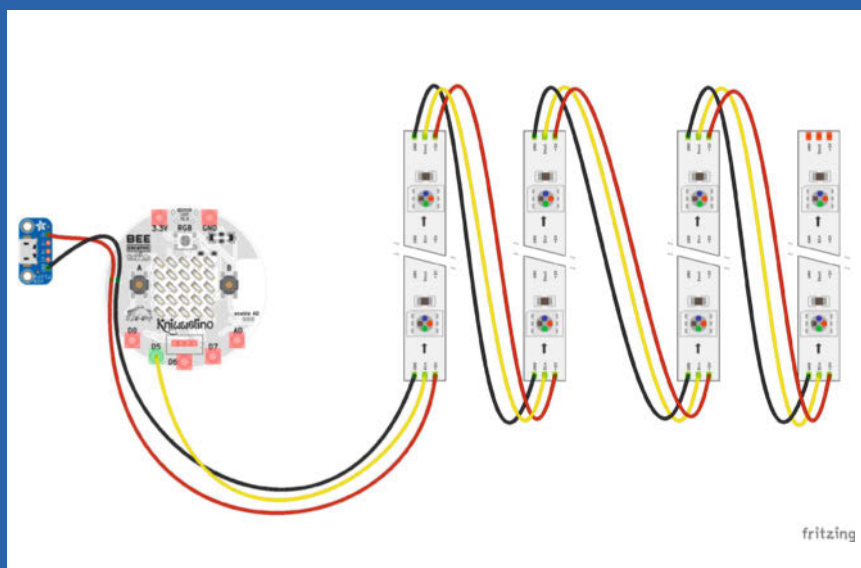
Für das Gehäuse haben wir die Skyline von Luxemburg genommen und etwas angepasst. Die einzelnen Teile haben wir zunächst aus Karton gelasert und nach einem kurzen Probelauf der LED-Streifen aus Sperrholz gefertigt. Der Kniwwelino wird am besten in einer 3D-gedruckten Hülle montiert. Damit der LED-Streifen passt, haben wir ihn in einen Abschnitt mit zwei LEDs und drei Abschnitte mit vier LEDs zerschnitten und anschließend die Abschnitte wieder mit dünnen Kabeln verlötet. Die vollständige Anleitung mit den Schnittvorlagen und Programmen zum Herunterladen finden Sie online.

—hch

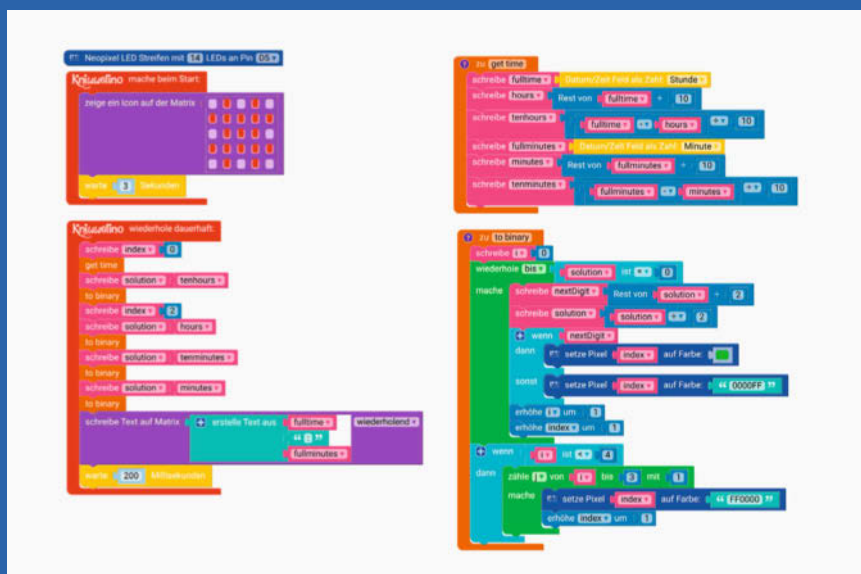
► heise.de/-4850868



Entwurf der Skyline für den Lasercutter



Schaltplan mit Kniwwelino und LED-Streifen



Die Programmierung erfolgt mit bunten Blöcken.

TV-Lift

Nachdem ich viel Geld für einen neuen Fernseher im Wohnzimmer ausgegeben hatte, war der alte, circa 22 Zoll große Fernseher übrig. Er sollte nun einen Platz im Schlafzimmer finden, allerdings war dort kein Platz. Als Lösung habe ich mir einen Fernsehlift gebaut.

von Matthias Jünemann



Der Fernseher „liegt“ nun auf dem Kleiderschrank und kann bei Bedarf ausgefahren werden. Dazu hängt er unter einer Schublade, die mit einem Linearmotor nach vorne fährt. Der Fernseher wird dann mit einem ausgedienten Rolladenmotor nach unten ausgeschwenkt. Zurück geht es natürlich analog, in umgekehrter Reihenfolge. Ein Arduino Uno mit einem selbstgebauten Shield mit Transistor-Transistor-Logik (TTL) überwacht alle Bauteile und für die bequeme Infrarot-Fernsteuerung habe ich meine TV-Fernbedienung mit eingebunden.

Als Werkstoff habe ich mich für 12mm starke Multiplexplatten entschieden, weil sie sehr stabil sind und sich gut verarbeiten lassen, und habe sie im Baumarkt zuschneiden lassen. Ebenfalls aus dem Baumarkt ist eine Rollschubführung für die Schublade. Der Linearmotor der Schublade nennt sich „ECO-Worthy Linear Actuator Eco-400“ auf eBay, hat eine Schublänge von 400mm und wird mit 12V DC betrieben. Das Aus- und Einschwenken des Fernsehers erfolgt mit einem handelsüblichen Jalousiemotor von Julius Mayer „JM35“, der mit 230 Volt betrieben wird: Dabei werden zwei Seile auf dem Jalousiemotor auf- bzw. abgerollt. Die Richtung wird über zwei Eingänge gesteuert, von denen immer nur einer angesteuert werden darf. Zur Ansteuerung der Motoren nutze ich vier gesockelte Relais des Herstellers Finder. Drei Endschalter überwachen die Lage der Schublade und des TVs.

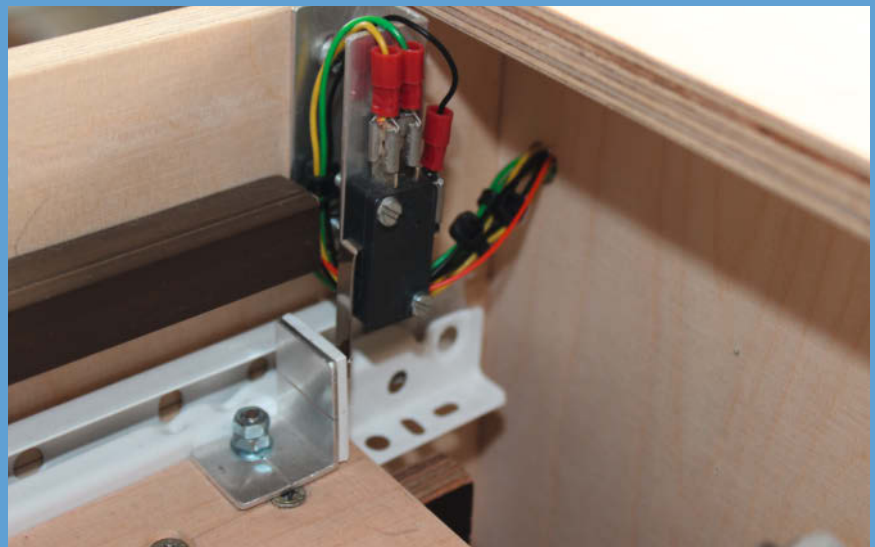
Vorn am Gerät habe ich zwei Taster und einen Infrarot-Empfänger montiert, um das Ein- und Ausfahren zu starten. Für die Arduino-Steuerung habe ich das erwähnte Shield selbst entwickelt. Die Endschalter gehen über LEDs auf dem Shield an die Eingänge, damit man die Zustände der Leitungen direkt erkennen kann. Zwischen das Relais und den Arduino habe ich vier UND- und ein ODER-Gatter gesetzt. Sie verknüpfen die Zustände der Arduino-Ausgänge und der Endschalter und steuern mit ihren Ausgängen den „8CH HIGH-Voltage Source Driver“-IC TD62783AP als Ausgangstreiber, um die Relais zu schalten. Weder die TTL-ICs noch der Arduino sind ansonsten geeignet, um genügend Strom auf das Relais zu geben.

Damit ich meine Infrarot-Fernbedienung von LG nutzen kann, habe ich auf dem Shield noch das Infrarot-Modul TSOP 4836 verbaut. Zur Auswertung der Signale nutze ich die Arduino-Bibliothek IRremote. Über den seriellen Monitor lassen sich die übertragenen Codes beim Druck auf die gewünschten Tasten ermitteln und in das eigene Programm einbinden. Den ganzen Baubericht, inklusive Schaltplänen und Programmcode zum Herunterladen und Nachbauen, gibt es online.

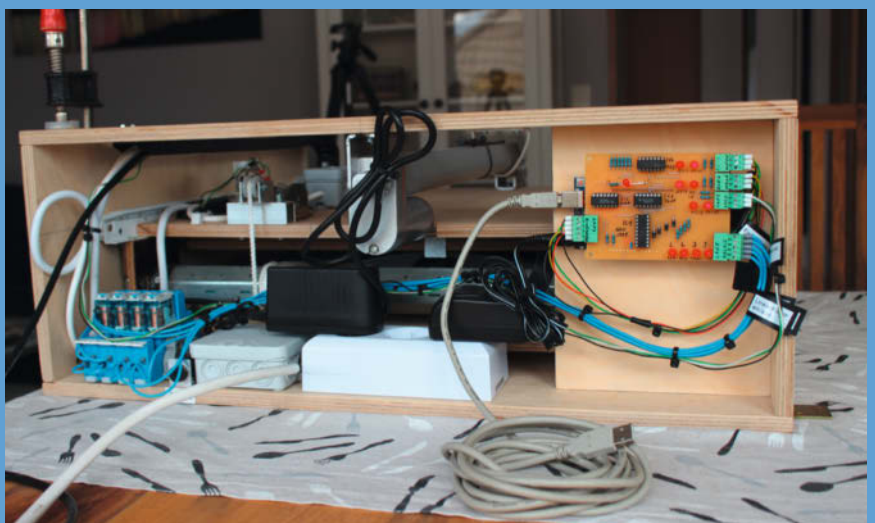
—hch



Ausgefahrener Fernseher in der frisch gebauten Schublade



Endschalter überprüfen die Lage der Schublade und der Jalousie.



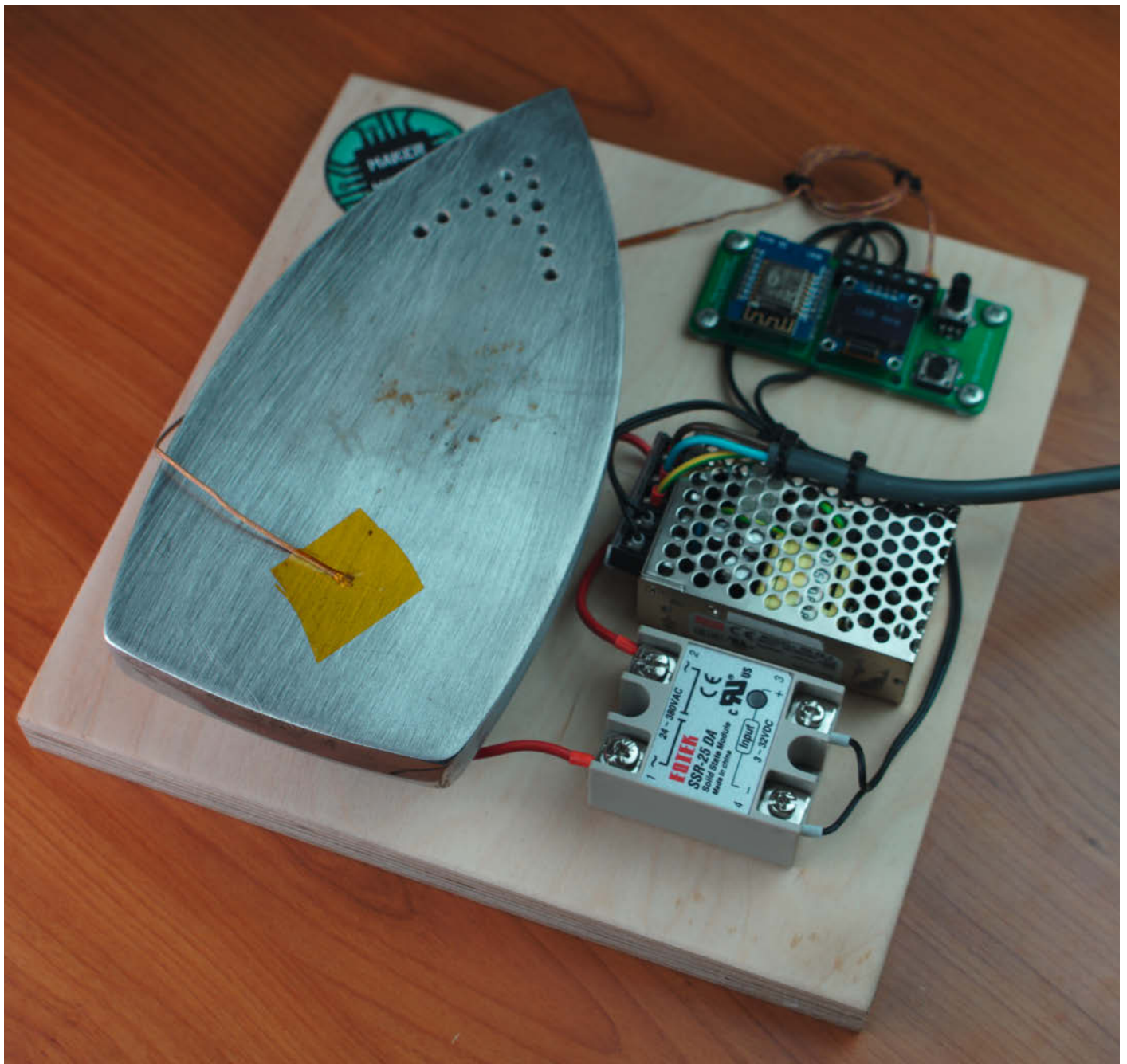
Ein Arduino mit selbstentwickeltem Shield steuert das Projekt.

► heise.de/-4893679

Reflow-Löten mit dem Bügeleisen

Bei Musterplatten ist das SMD-Löten von Hand noch praktikabel, aber schon für eine Kleinstserie lohnt das Reflow-Verfahren mit Pastenschablone und Lötöfen oder Heizplatte. Für letztere lässt sich prima ein altes Bügeleisen „upcyclen“: Wenn man etwas Steuer-Elektronik nachrüstet, kann man sogar mit bauteilschonenden Temperaturprofilen löten.

von Moritz König



Kommerzielle Reflow-Öfen und Platinen-Heizplatten gibt es genügend, aber eine Lösung, die kostengünstig ist und aus „haushaltsüblichen“ Dingen gebaut werden kann, ist viel reizvoller: Das alte, kaputte Bügeleisen aus der Garage war einfach zu schön, um es auf den Schrottplatz zu bringen. So kam uns die Idee der Bügeleisenlötstation. Nachdem das Schätzchen zerlegt worden war, ging es sofort daran, die Sohlentemperatur zu messen, und schnell stellte sich heraus, dass die zum Löten benötigte Temperatur sicher erreicht wird.

Anders als bei den DIY-Pizzaofenlötstationen kommt die Wärme hierbei von unten. Ob es schonender für die Bauteile ist, weil sie zuletzt der Wärme ausgesetzt sind, lässt sich nur vermuten, aber zumindest ist es der ungestörte Anblick der schmelzenden Lötpaste wert, sich so eine tolle Bastelhilfe nachzubauen.

Die richtige Temperatur

Woher weiß man denn, wie der richtige Temperaturverlauf beim Platinenlöten aussieht? Häufig werden dazu Informationen in den Datenblättern zu den jeweiligen Bauteilen, die sich auf der Platine befinden, angegeben. Das sogenannte *Reflow Profile* (oder *Solder Profile*, Lötprofil) gibt den empfohlenen Temperaturverlauf während des Lötens an, wobei sich die grundlegende Kurve von Bauteil zu Bauteil nur geringfügig unterscheidet (was plausibel ist, da sonst einzelne Bauteile beschädigt werden können). In Bild 1 ist beispielsweise das Lötprofil des Time-of-Flight-Sensors VL53L0X abgebildet. Grundsätzlich können solche Kurven in vier Teilabschnitte unterteilen werden:

- **Vorheizen:** Beim Vorheizen wird die Platine auf ungefähr die halbe Maximaltemperatur gebracht. Dies dient vor allem zum Verflüchtigen von Lösungsmitteln und zum gleichmäßigen Erwärmen von Komponenten, Platine und Lötpaste. Die Temperatur steigt hierbei bis etwa 120°C bis 150°C an.
- **Aktivierungszone** (Soak-Zone): Wie der Name schon sagt, dient diese Zone zur Aktivierung des Flussmittels, welches ein Hauptbestandteil der Lötpaste ist. Das Flussmittel wechselt dabei in den flüssigen Zustand. Die Temperatur steigt hier nur minimal an und sollte in etwa bei 150°C bis 160°C liegen. Im unten abgebildeten Diagramm ist dies der Bereich zwischen T_{smin} und T_{smax} .
- **Reflow-Zone:** In diesem Bereich findet der eigentlich Lötprozess statt. Die Lötpaste schmilzt und verbindet sich mit den Kontaktstellen der Bauteile. Der Temperaturbereich liegt hier zwischen 200°C und 250°C (Diagramm: T_{p-10} bis T_p).
- **Abkühlzone:** Die Temperatur sinkt wieder unter die Liquidustemperatur (Temperatur, bei der ein Material von dem flüssigen in den festen Zustand übergeht), das Lötzinn wird fest.

Kurzinfo

- » Bügeleisen-Sohle als Löt- und Heizplatte
- » Temperaturprofile und Lötverlauf
- » Platinen mit Paste beschichten und bestücken

Checkliste



Zeitaufwand:

4 Stunden



Kosten:

30 Euro



Maschinen:

Bohrmaschine oder Akkuschrauber mit Holzbohrern



Programmieren:

Arduino-IDE bedienen



Hochspannung:

Erfahrung mit 230V-Installationen



Warnhinweis

Achtung! Die vorgestellte Schaltung arbeitet mit lebensgefährlicher Netzspannung und sollte nur von damit vertrauten Personen in Betrieb genommen werden. An der Heizplatte besteht Verbrennungsgefahr.

Material

- » Altes Bügeleisen, möglichst unbeschichtet
- » Solid-State-Relais für 1kW-Lasten
- » MAX6675 Sensor-IC, ggf. auf Breakout-Board
- » K-Thermoelement (Temperatursensor)
- » Wemos D1 Mini
- » OLED-Display mit I²C-Schnittstelle
- » 5V- oder USB-Netzteil
- » Kabel, Taster, Poti laut Stückliste auf Github (siehe Link)
- » Holzschrauben
- » Doppelseitiges Klebeband
- » Kapton-Klebeband
- » Multiplex- oder HDF-Platte (Rest)

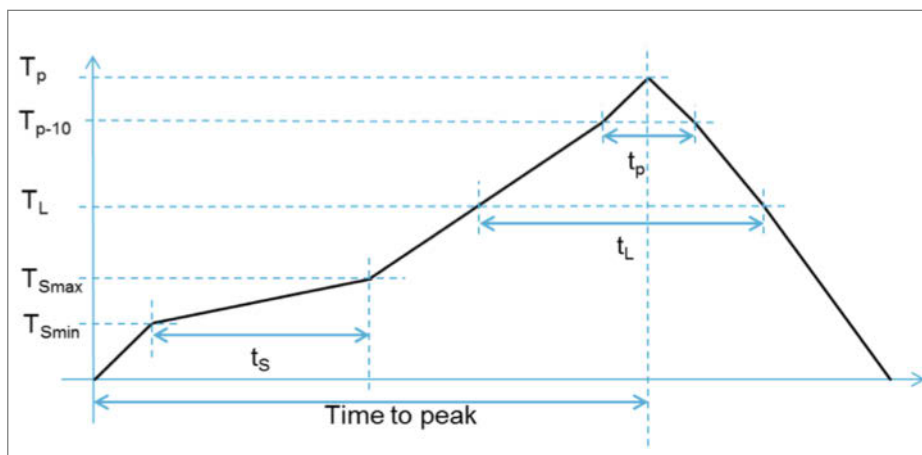
Mehr zum Thema

- » Löten in der Pizzapfanne, c't Hardware Hacks 1/2012, S. 40
- » Schablonendrucker für SMD-Löter, Make 2/2019, S. 112

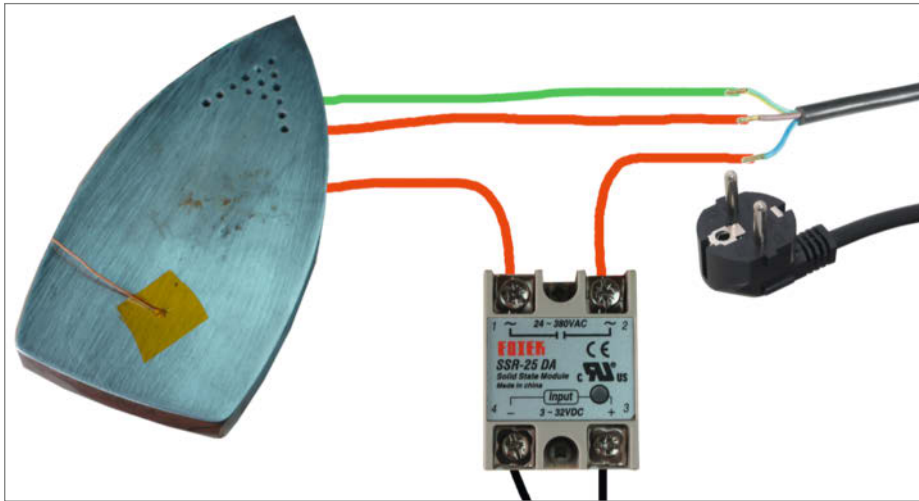
Alles zum Artikel
im Web unter
make-magazin.de/xkzs

Bügel-Praxis

Das oben besprochene Temperaturprofil sieht in der Bügel-Praxis natürlich wieder etwas anders aus. Besonders beim Bügeleisen, aber



1 Der beispielhafte Temperaturverlauf (Reflow-Profil) eines VL53L0X Time-of-Flight Sensors.



➊ Anschluss des Relais SSR-25 DA, Pin 1: Bügeleisen Anschluss 1, Pin 2: 230V-L, Pin 3: Wemos D1 Mini Pin D8, Pin 4: Wemos D1 Mini Masse. Zusätzlich wird der Anschluss 2 des Bügeleisens mit 230V-N und der Masseanschluss der Sohle mit dem Schutzleiter (grün/gelb) verbunden.

auch bei DIY-Reflow-Öfen können teils nicht alle Temperaturen bzw. Temperatur-An- und -Abstiegszeiten exakt eingehalten werden, was aber in der Regel nicht so kritisch ist. Das hier verwendete Bügeleisen besitzt einen sehr dicken Eisenkern, was eine ziemlich träge Temperaturveränderung mit sich bringt. Beim Aufheizen kann das Bügeleisen fast noch mithalten, aber besonders das Abkühlen unter die Liquidustemperatur dauert mehrere Minuten, weshalb die Platine stets nach dem Verflüssigen der Lötpaste von dem Eisen genommen werden sollte, um Schäden an Platine und Komponenten zu vermeiden.

Vereinfacht können hier also nur zwei Zustände angenommen werden. Die Gradienten, also die Temperatur-An- und -Abstiege, werden hierbei vernachlässigt und bilden sich durch die Trägheit des Eisens von ganz allein. Der erste Zustand ist die Soak-Zone bei 150° C. Bei dieser Temperatur sollte die Platine eine kurze Zeit verweilen. Den zweiten Zustand, die Reflow-Zone, wird nur kurz angefahren und bildet zugleich auch fast das Ende des Lötprozesses. Sobald also das Bügeleisen die Maximaltemperatur erreicht hat, schaltet man es ab und nimmt die Platine vom Eisen.

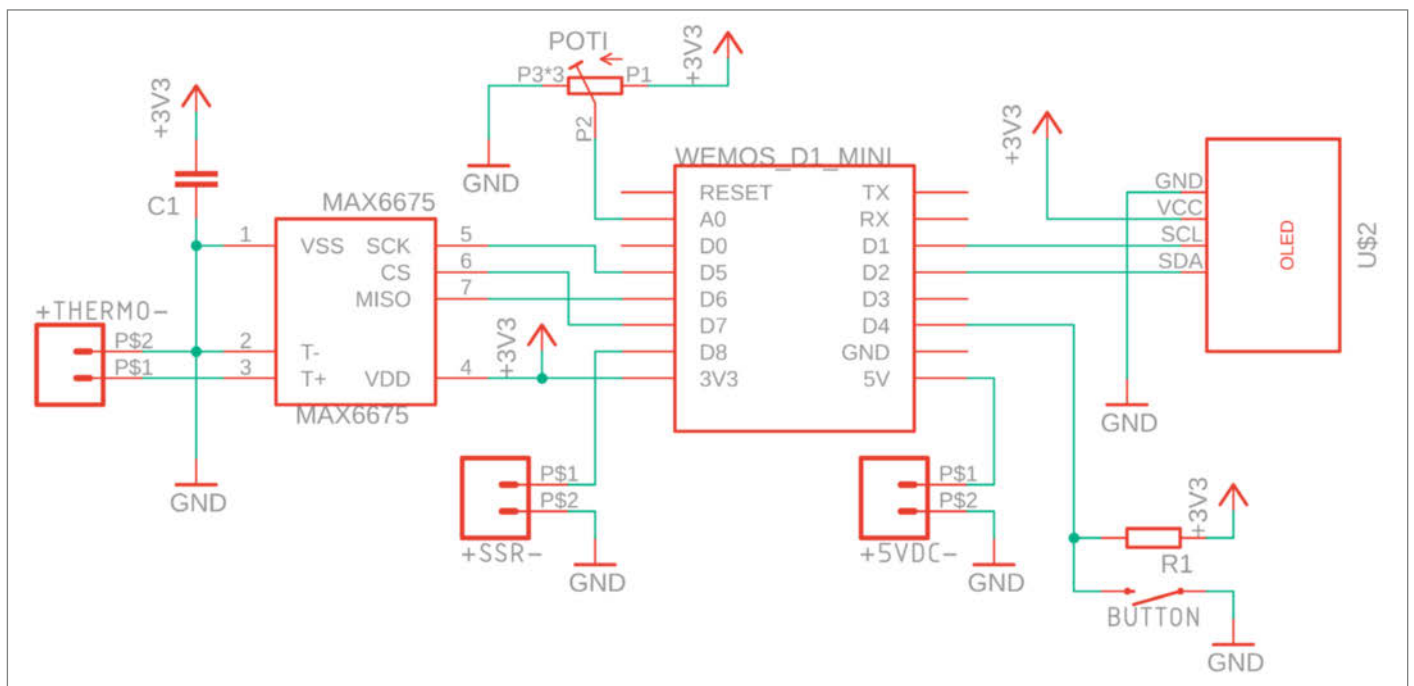
Elektronische Ansteuerung

Je nach Trägheit des verwendeten Bügeleisens funktioniert das Platinenbügeln auch ohne Steuerung. Da es aber durchaus nützlich ist, über die aktuelle Temperatur sowie den Zustand während des Lötens Bescheid zu wissen, haben wir für diese Konstruktion eine Steuerung aus einem Solid State Relay und einem Wemos D1 Mini als Controller entwickelt. An dieser Stelle möchten wir nochmals darauf hinweisen, dass für Aufbau und Verdrahtung Erfahrungen mit 230V-Installationen erforderlich sind – hier herrscht definitiv Lebensgefahr.

Da der Wemos D1 Mini natürlich selbst keine 230V schalten kann, wird der Heizdraht des Bügeleisens über ein Solid-State-Relais geschaltet, das nur einen geringen Steuerstrom benötigt ➋. Das kleine OLED-Display wird über die I²C-Schnittstelle angesteuert, während das Temperatur-Modul über SPI angesprochen wird ➌. Außerdem befindet sich auf der Station ein kleines 5V-Netzteil für den Wemos D1 Mini (der Mikrocontroller kann aber auch über ein externes USB-Netzteil verwendet werden). Der Temperatursensor wird mit hitzebeständigem Kapton-Klebeband auf das Eisen geklebt. Um die richtige Temperatur des Eisens zu ermitteln, sollte darauf geachtet werden, dass der Sensor nicht am Rand platziert wird, da es dort am längsten dauert, bis es warm wird.

Mechanischer Aufbau

Der mechanische Aufbau besteht aus einer Multiplex-Holzplatte, die alle elektronischen



➋ Schaltplan der Steuerungsplatine mit Wemos D1 Mini, MAX6675 Temperatur-IC sowie Taster, Potentiometer und Mini-OLED-Bildschirm. Der Autor hat hierfür eine Platine entworfen, der Aufbau auf einer Lochrasterplatine ist aber ebenso möglich.

Komponenten an Ort und Stelle hält; noch besser geeignet (aber nicht so einfach zu bearbeiten) sind HDF- oder Konstruktionsplatten mit 6mm Stärke (Baumarkt). Die Basisplatte ist dem Bügeleisen angepasst und in unserem Fall 28cm × 28cm groß **4**. Die Steuerungsplatine und das Solid-State-Relais können von oben mit vier bzw. zwei Schrauben fixiert werden. Beim Bügeleisen kommen die Schrauben von unten; damit sie nichts verkratzen, sollte man dafür Senkkopfschrauben verwenden. Zusätzlich kann man in den Ecken noch Möbelgleiter aus Filz als Füße anbringen.

Je nach Bügeleisen müssen eventuell zusätzliche Löcher gebohrt oder gefräst werden, um an die Anschlüsse des Heizdrahtes zu gelangen **5**. In unserem Fall haben wir die Multiplexplatte mit einem großen Kegelsenker an der linken Anschlussklemme vertieft. Alternativ kann man die Sohle auch mit metallenen Abstandshülsen montieren.

Firmware

Damit die Bügelstation auch das macht, was sie soll, muss der Wemos D1 Mini noch mit unserer Firmware (siehe Link in der Kurzinfor) programmiert werden. Wie man mit Hilfe der Arduino-IDE einen Sketch auf ESP8266-basierter Controller bekommt, war schon oft Thema in Make und soll deshalb hier nicht wiederholt werden. Unter dem Link finden Sie nähere Hinweise.

Wie zuvor erwähnt, wird aufgrund der thermischen Trägheit des Bügeleisens kein PID-Regler oder vergleichbares benötigt. Eine simple Zweipunktregelung reicht hierbei vollkommen aus, was die Regelschleife stark vereinfacht. Der Sketch aktualisiert alle 200ms das OLED-Display mit Temperatur, Zustand und Solltemperatur, wartet auf einen Tastendruck oder eine Potentiometeränderung und schaltet gegebenenfalls das Solid-State-Relais, sollte sich die Station im Bügelmodus befinden. Im Bügelmodus wird außerdem eine Prozentanzeige eingeblendet, die Auskunft über den Verlauf des aktuellen Zustandes gibt.

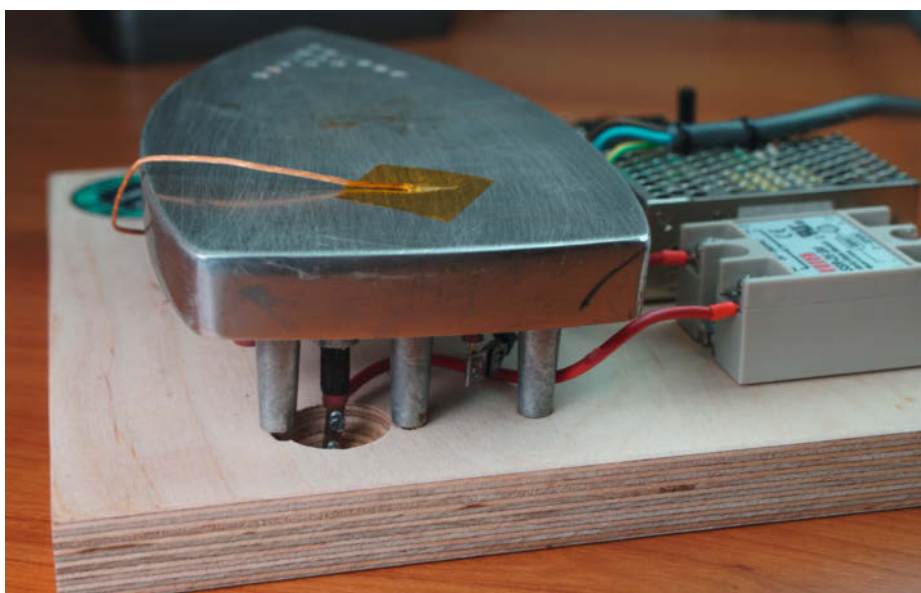
Pastendruck

Nachdem man die Platinen gereinigt hat (hier eignet sich am besten Isopropanol), wird die Lötpaste auf den Pads verteilt. Dafür gibt es ebenso viele Verfahren wie für das Löten selbst. Bei Einzelstücken und sehr kleinen Platinen kann man die Lötpaste mit Spritze und Dosierkanüle manuell auftragen. Unser persönlicher Favorit (und in der Industrie üblich) ist allerdings das Stencil-Verfahren.

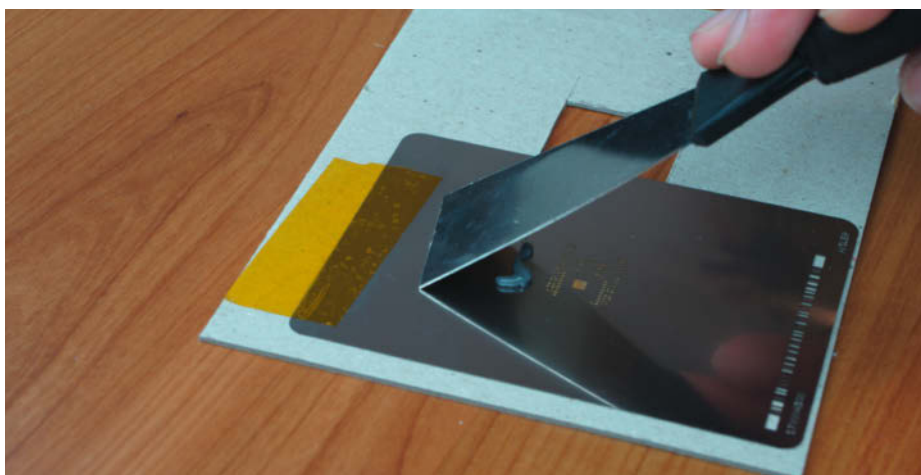
Hierbei werden Schablonen mit Ausschnitten für die Pads benutzt, die man bei fast allen Platinen-Dienstleistern parallel zur Platine in Auftrag geben kann; Voraussetzung ist nur,



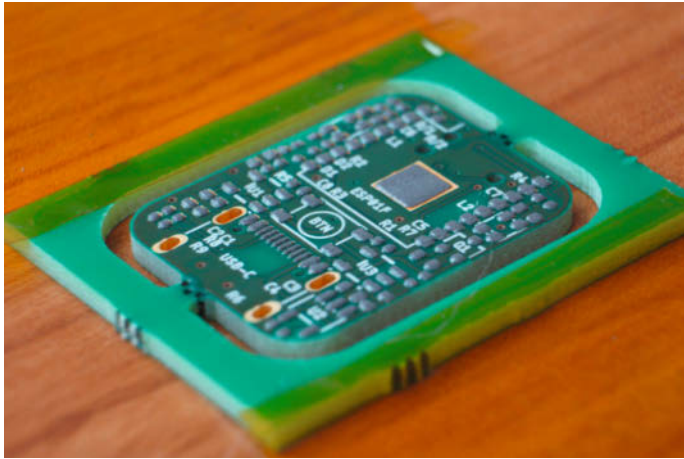
4 Die einzelnen Komponenten (hier das Halbleiter-Relais) wurden mit Schrauben auf der Grundplatte montiert. Die 230V-Anschlüsse (oben) sollten später berührungssicher abgedeckt werden.



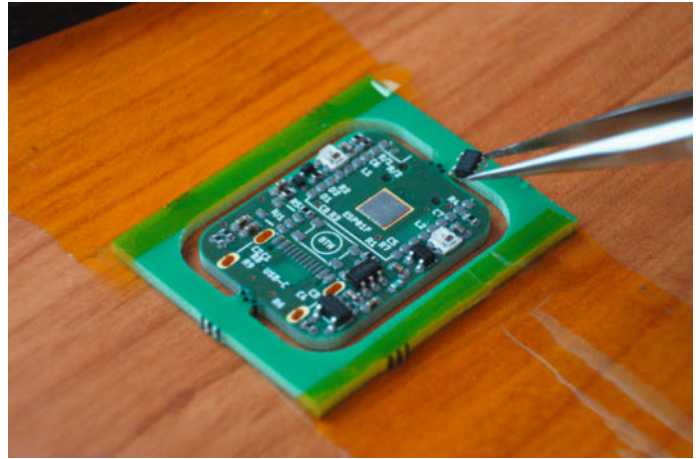
5 Ausfräsungen in der Multiplexplatte für die Anschlusskontakte des Bügeleisens erleichtern die plane Montage der Sohle.



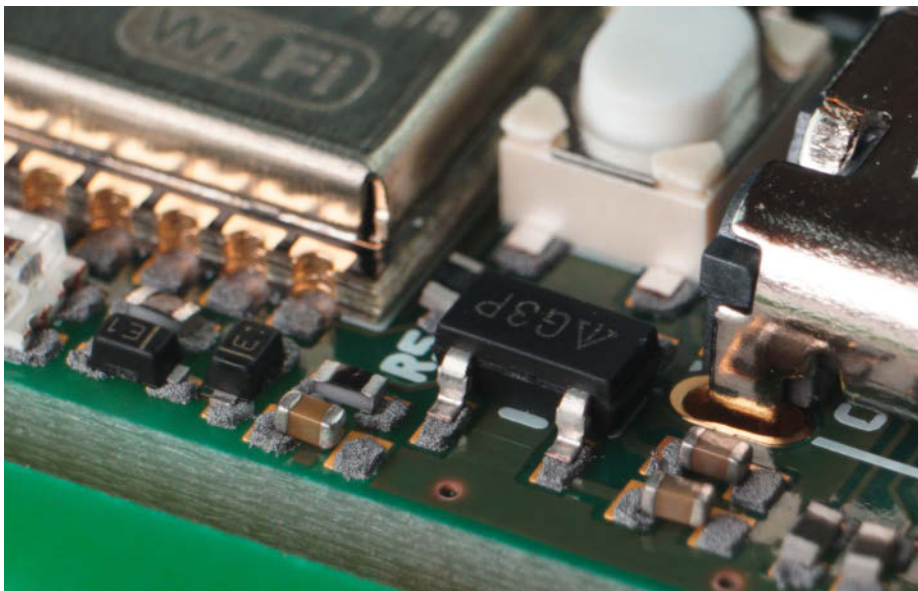
6 Das Rakeln der Platine bringt Lötpaste auf die Pads der Komponenten.



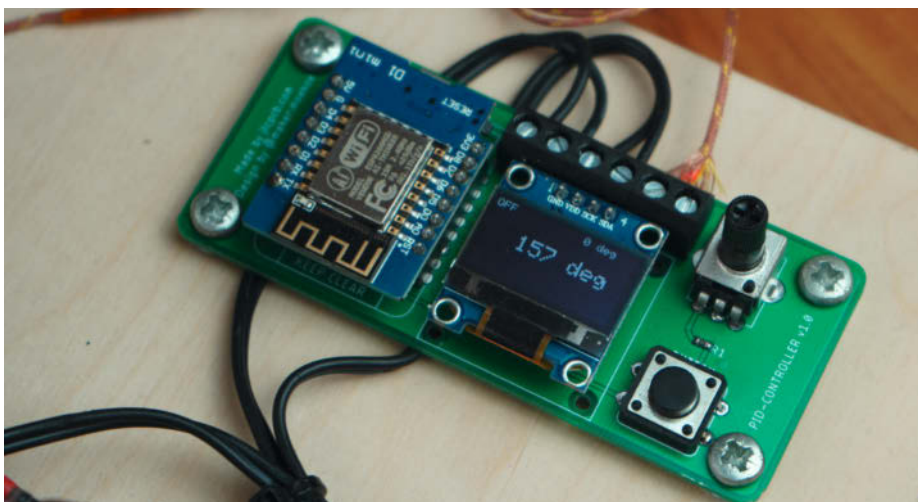
7 Nach Abheben der Schablone sollte die Lötpaste so verteilt sein wie hier zu sehen.



8 Die Bauteile werden mit Hilfe einer Pinzette oder einer Vakuum-Pipette bestückt.



9 Achten Sie darauf, dass alle Bauteilanschlüsse Kontakt mit der Lötpaste haben. Der kleine SMD-Kondensator im Vordergrund ist nur einseitig verbunden, er musste später von Hand nachgelötet werden.



10 Nach Erreichen der Zieltemperatur aktiviert die Firmware eine Zweipunkt-Regelschleife.

dass man eine Gerber-Datei für die SMD-Pads mitliefert. Schablonen mit 100 oder 125µm Stärke sind überall Standard und eignen sich auch für die manuelle Bedruckung mit der Rakel. Die Schablonendicke wählt man nach der Feinheit der Pad-Strukturen: Bei eher groben Layouts wählt man eine dickere Schablone, es gelangt dann auch mehr Paste auf die Pads.

Vor einigen Ausgaben haben wir einen Schablonendrucker zum Selbstbau vorgestellt, es geht für kleine Stückzahlen aber auch einfacher: Man klebt Platinenreste so auf eine Grundplatte, dass ein fester, nur an einer Seite offener Rahmen für die zu bedruckende Platine entsteht. Die Edelstahl-Schablone wird einseitig mit Klebeband befestigt.

Wird nun über die Schablone gerakelt, verteilt sich nur an diesen Stellen die Lötpaste; es sollten keine nebeneinanderliegenden Pads mit Lötpaste überbrückt sein **6**. Ein Ergebnis, mit dem sich arbeiten lässt, ist in Bild **7** zu sehen.

Nach spätestens drei beschichteten Platinen sollte die Rückseite der Schablone gereinigt werden, da die Paste mit unserem einfachen Aufbau leicht verschmiert. Apropos: Für private Zwecke darf man ruhigen Gewissens bleihaltige Legierungen nehmen, die einen niedrigeren Schmelzpunkt aufweisen als bleifreie.

Nach dem Rakeln werden die Bauteile mit einer feinen Pinzette aufgesetzt **8**; das sollte man tunlichst nicht nach dem hastigen Genuss von fünf Tassen Kaffee tun, denn allzu leicht wischt man bereits aufgesetzte Bauteile versehentlich von den Pads. Die Bauteile haften durch die Adhäsion der Paste ansonsten so gut auf der Platine, dass man sie gefahrlos vom Bestückungsplatz zur Bügeleisen-Lötstation tragen kann **9**.

Nach dem Einschalten sollte die aktuelle Temperatur des Bügeleisens auf dem OLED-Display angezeigt werden.

Die Steuerung wechselt beim ersten Tastendruck vom „Off“- in den „Preheat“-Zustand und fährt die Temperatur bis auf 150°C hoch **10**. In unserem Fall werden die 150°C nach circa 70 Sekunden erreicht. Anschließend hält ein simpler Zweistufenregler die Temperatur, und die Firmware wartet auf den nächsten Tastendruck.

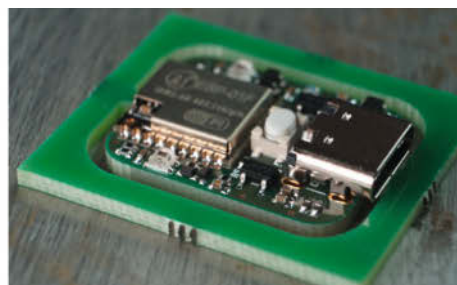
Dann wechselt die Steuerung in den „Reflow“-Zustand und fährt die zuvor mit dem Poti eingestellte Reflow-Temperatur an. Bei 190° dauert dies weitere 60 Sekunden. Nach dem Erreichen der Zieltemperatur wechselt die Steuerung automatisch in den „Cooling“-Zustand und schaltet das Relais aus. Wie

oben erwähnt, sollte die Platine dennoch nicht auf dem Bügeleisen gelassen werden, da es sich viel langsamer abkühlt als verlangt. Hierbei ist Vorsicht geboten, denn die Lötpaste ist noch nicht unter die Liquidustemperatur gefallen, wodurch die Bauteile durchaus noch verrutschen können **11**. Wir benutzen zum Abheben der Platine einen dünnen Spachtel.

Verbesserungen

Abschließend inspiziert man alle Lötstellen und lötet gegebenenfalls von Hand nach, wenn sich das Zinn nur auf einer Seite mit dem Bauteil verbunden hat.

Die Redaktion hat mit dem beschriebenen Verfahren zumindest bei kleinen Platinen hervorragende Ergebnisse erzielt, die sich kaum von Ofenlötungen unterscheiden. Gleichwohl bietet der Aufbau noch Raum für Verbesserungen: Voluminöse Bauteile (z.B. größere Spannungsregler-ICs) werden besser durchgewärmt, wenn man die Platine beim Preheat- und Lötvorgang mit dem Deckel einer Butterdose abdeckt. Wir werden zudem noch mit einem Lüfter unter der Bügeleisen-Sohle experimentieren, um sie nach dem Erreichen der Spitzentemperatur rascher abzukühlen; damit wäre das vorzeitige und „gefährliche“ Abheben mit noch flüssigem Lötzinn unnötig. —cm



11 Lötprozess: Nach dem Aktivieren der Lötpaste schmilzt das Lot bei Erreichen der Spitzentemperatur kurz auf und verbindet die Bauteile mit der Platine.

SMART LOSLEGEN

mit dem Node-RED Special von Make!

Make: Node-RED Grundlagen

Node-RED SPECIAL

inklusive ESP8266 D1 Mini mit WLAN

Jetzt auch **komplett digital** erhältlich!

Für Smart Home & IoT

Projekte

- Temperaturen messen
- Fenstersensoren überwachen
- Servomotor als Anzeige

Für Fritzbox

- Präsenzerkennung in Bandbreitenanzeige
- LED-Streifen
- DECT-Geräte steuern

inkl. D1 Mini

Az-Delivery

CE

Make Special: Node-RED

Für Einsteiger ohne tiefere Programmier-Kenntnisse bietet **Node-RED** die ideale Lösung. Man verknüpft fertige grafische Bausteine zu einem „Flow“ und kann so komplexe Anwendungen für **IoT und Smart Home** entwerfen.

Im neuen **Make Node-RED Special** bietet die deutsche Make-Redaktion Anfängern und Fortgeschrittenen einen leichten Einstieg in die Programmierung von Smart Homes. Das Heft wird **portofrei** **inklusive eines ESP8266 D1 Mini mit WLAN** geliefert!

shop.heise.de/ma-nodered

24,95 € >

Generell portofreie Lieferung für Heise Medien- oder Maker Media Zeitschriften-Abonnenten. Nur solange der Vorrat reicht. Preisänderungen vorbehalten.

© Copyright by Maker Media GmbH.

heise shop

shop.heise.de/ma-nodered

Bartop Arcade mit Raspberry Pi

Sauber planen mit 3D-Software macht Spaß, aber jetzt wird es gefährlich und schmutzig, denn es geht an die Holzbearbeitung, damit die Retro-Spielkonsole mit Raspi-Herz ein stilechtes Gehäuse bekommt.

von Carsten Wartmann



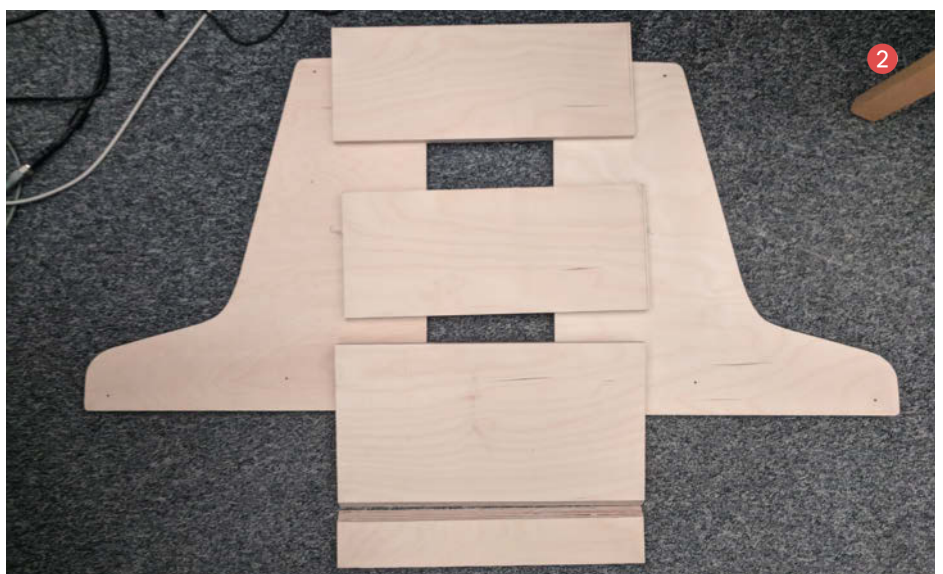
Eigentlich war meine Retro-Bartop-Arcade im Prinzip ja schon fertig, nachdem das Material ausgewählt war, die minutiöse Planung mit der 3D-Software Blender ansprechende Renderings ergab sowie die Elektronik stand und funktionierte (ausführlich beschrieben in Make 4/20 ab Seite 14). Doch dann ruhte das Projekt erst einmal fast ein Jahr – ich hatte einfach keine Ahnung, wie ich die jetzt anstehende Holzbearbeitung angehen sollte, bei der es galt, den passgenauen 3D-Entwurf in reale Bauteile für den Korpus umzusetzen. Ich wollte es natürlich selber machen, ein Tischler mit CNC-Fräse wäre zu teuer geworden und bei Hobbyisten und in Fablabs stehen selten Fräsen, die mit dem 15mm-Multiplex umgehen können, das ich für das Gehäuse ausgewählt hatte.

Nach vielem Hin- und Herdenken war ich der Meinung, einen gangbaren Weg gefunden zu haben. Die geraden Schnitte machte ein Freund mit seiner Handkreissäge mit großer stabiler Anschlagschiene **1**. Eine gute Tischkreissäge hätte hier natürlich die Arbeit noch einfacher gemacht, war aber nicht in Reichweite.

Die Seitenteile mit ihren Rundungen und Schwüngen waren die größere Herausforderung **2**. Mit Stichsäge und viel schleifen wäre es vielleicht möglich, ein schickes Ergebnis zu erzielen, aber die Arbeit erschien nicht sehr verheißungsvoll. Daher entschied ich mich, eine (billige) Oberfräse zu kaufen. Da die Sichtkanten sowieso mit einem Radius versehen werden sollten, wofür eine Oberfräse ja wie geschaffen ist, war das eine weitere Rechtfertigung für den Kauf.

Mit Kopterring

Ich habe dann anhand des 3D-Modells eine Schablone mit entsprechend Untermaß für den Kopterring der Fräse erstellt **3**. Hierbei



Kurzinfo

» Gehäuse aus Plattenmaterial nach Schablonen aus 3D-Software fräsen

» Flächen folieren statt lackieren

» Retro-Charme bei handlichen Abmessungen

Checkliste



Zeitaufwand:

etwa 20 Stunden Design und Elektronik, 5 Stunden 3D-Druck, gut 20 Stunden Gehäusebau und Endmontage



Kosten:

50 bis 150 Euro für die Hardware, je nach Inhalt der Bastelkiste und Recyclingquote; Material für das Gehäuse je nach Größe (siehe dort)



Holzbearbeitung:

Sägen, bohren, fräsen, schleifen, folieren



3D-Druck:

für Verbindungssteile und Halterungen (optional)

Material

- » **Hardware** ausführliche Liste siehe Artikel im vorigen Heft
- » **Multiplex-Platte** etwa Birke, 15mm stark, etwa 20 Euro/m²
- » **Kunststoffolie** d-c-fix, schwarz, matt, etwa 6 Euro/m²
- » **Sperrholz** 5mm, für Bezel (Monitorblende) und Frässhablonen
- » **Superkleber** (CA-Kleber), dünnflüssig
- » **Holzschrauben** Kreuzschlitz oder Torx, 5x60mm, für Verbindung von Seiten und Böden
- » **Holzschrauben** 4x25mm für Winkel
- » **Montagematerial** etwa Maschinenschrauben für den Monitorhalter

Werkzeug

- » **Handkreissäge** mit Schiene
- » **Oberfräse** mit Nut- und Abrundungsfräsern sowie Kopterring
- » **Bohrmaschine** mit Forstnerbohrern im passenden Durchmesser zu den Arcade-Knopfen
- » **Handwerkzeug** zur Holzbearbeitung
- » **Schwingschleifer** (optional), Schleifklötze und Schleifpapier für die Oberflächenbearbeitung

Alles zum Artikel
im Web unter
make-magazin.de/xhf5

Mehr zum Thema

- » Carsten Wartmann, Retro-Gaming auf Raspberry Pi, Make 4/20, S. 8
- » Carsten Wartmann, Bartop Arcade mit Raspberry Pi (Teil 1), Make 4/20, S. 14
- » Daniel Bachfeld, Emulator-PC im Spielhallengehäuse, c't Hacks 3/12, S. 100



ist der Versatz vom Kopiererring zum Fräser einzukalkulieren, das heißt, die Schablone für äußere Umrissse muss an allen Seiten um dieses Maß kleiner sein (im Gegenzug müssen Ausschnitte um genau dieses Maß größer in der Schablone sein). Den Versatz habe ich mit meiner bevorzugten 3D-Software Blender recht trickreich mit Hilfe der *Inset Faces*-Funktion erstellt **4** – ich bin sicher, ein richtiges CAD-Programm kann das mit wenigen Mausklicks. Blender aber kommt bei 2D-Formen an

seine Grenzen, daher wird der Umweg über ein 2,5D-Objekt fällig.

Einfacher wäre gewesen, einen Konturen- oder Bündigfräser mit Kugellager und Anlauferring am Schaft zu benutzen. So etwas erschien mir damals aber noch recht teuer. Inzwischen bekommt man so einen kompletten Fräsersatz in einfacher Qualität für etwa 10 Euro und kann damit ohne Versatz arbeiten: Der Fräser fräst 1:1 die Form heraus, die die Schablone vorgibt. Das ist gerade bei kleineren Teilen und enge-

ren Radien unabdingbar und die Investition absolut wert.

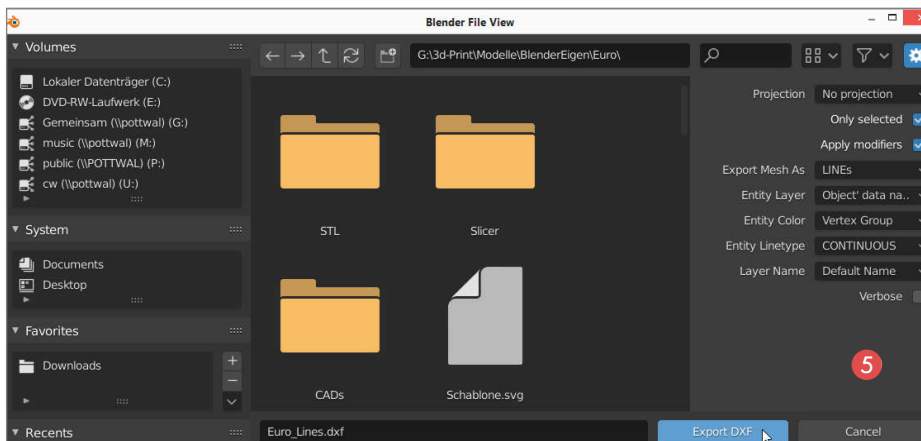
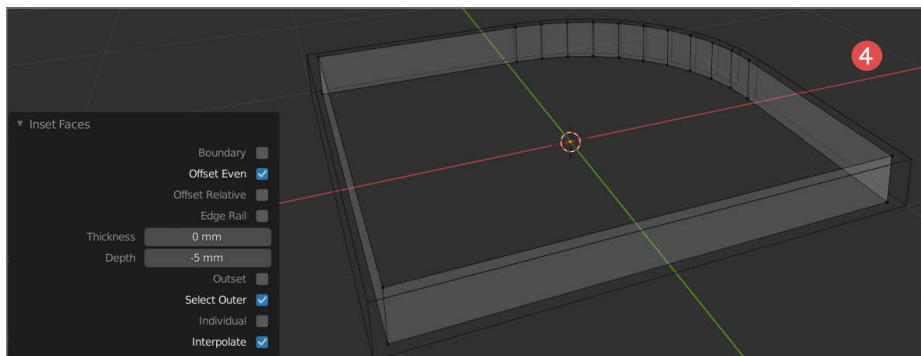
Schablonen-Export

Für den Export so einer Schablone muss man leider auf das in Blender schlecht unterstützte DXF-Format zurückgreifen. Als umständlicher, aber gangbarer Weg hat sich hierbei bewährt, im 3D-Modell die Flächen aus dem Seitenteil zu entfernen, sodass nur noch die Kontur und die Bohrungen als Kanten (Edges) übrig bleiben. Die Kontur muss aus der Draufsicht als DXF exportiert werden. Für meine Belange funktionierende Parameter sieht man auf dem Screenshot **5**. Ist dies geschafft, kann die DXF-Datei in einem Vektorzeichenprogramm wie *Inkscape* geöffnet und bei Bedarf noch weiter bearbeitet werden, um etwa Anmerkungen, Linienstärken und -muster zu ergänzen oder um kleine Fehler zu korrigieren.

Die Schablone habe ich dann mit Hilfe der Freeware *PDF-XChange Viewer* auf A4-Blätter verteilt gedruckt (das Bild **6** zeigt den Vorgang bei einem ähnlichen Projekt) und auf eine 5mm dicke Sperrholzplatte geklebt. Hier merkte ich, wie groß die Verzerrungen beim 2D-Druck sein können, denn die einzelnen Blätter passten nur sehr widerwillig zusammen. In der Rückschau würde ich heute lieber in einem Copyshop die ganze Vorlage am Stück auf A2 oder A1 drucken lassen. Ein anderer Tipp ist, einen anderen Drucker auszuprobieren, denn in meinem Fall brachte ein Tintenstrahler unerwartet wesentlich besser zueinander passende Blätter zustande als unser alter LaserJet.

Wenn möglich, klebt man die Vorlage so auf die Platte, dass zwei gerade und rechtwinklige Kanten der Schablone mit den Seiten der Platte zusammenfallen, damit man sich zumindest hier schon mal die Bearbeitung sparen kann. Dann bohrt man die Löcher, in denen nachher die Schrauben in den Seitenteilen sitzen und schraubt eine zweite 5mm-Platte an die erste, sodass man gleich zwei Schablonen produziert, falls während des Fräsens etwas schief geht und man eine Schablone zerstört. Außerdem gelingt das In-Form-Schleifen der Kanten bei zwei Platten als Sandwich genauer und man schleift weniger Buckel in die Kontur – das ist jedenfalls meine Erfahrung.

Entlang der Vorlage habe ich per Stichsäge die Schablonen ausgesägt und dann an den Kanten mit einem Schleifklotz und einer Schleifmaus (einem Miniatur-Schwingschleifer, nicht zu verwechseln mit dem Lieblingswerkzeug aus Make 3/19, S.18) glatt geschliffen, was dank des dünnen weichen Sperrholzes recht einfach war. An der fertigen Schablone kann man die Kanten noch mit Superkleber (Cyanacrylat, CA) versiegeln, um sie robuster zu machen und kleinere raue Stellen zu füllen.





Fräsen

Da ich vorher noch nie eine Fräse in der Hand hatte, musste ich natürlich erstmal etwas dazu lesen, einige Videos schauen und dann zur Übung das klassische Küchen-Schneidbrett aus einem Stück Leimholz herausfräsen. Jetzt wurde es ernst, denn es ging an die Seitenteile aus dem gutem 15mm-Multiplex. Eine Schablone wurde auf eine Platte aufgeschraubt und dann in mehreren Durchgängen mit einem grundschneidenden Nutenfräser (der nicht nur an den Seiten, sondern auch unten schneidet) mit wachsender Frästiefe pro Durchgang das erste Seitenteil gefräst. Natürlich könnte man auch die Form zuerst mit der Stichsäge grob ausschneiden und dann erst die Kontur fräsen und käme vielleicht mit weniger Durchgängen aus, aber der Vorteil bei meiner Arbeitsweise ist, dass bis zum letzten Durchgang die komplette Platte als Auflagefläche für die Fräse an beiden Seiten des Fräskanals zur Verfügung steht.

Wie tief man in einem Durchgang fräst, hängt sicher vom Geschick des Bedieners, der Kraft der Fräse und dem Fräser ab. Aufpassen sollte man bei den folgenden Durchgängen,

dass sich im Fräsgang vor dem Fräser nicht die Späne ansammeln und festklemmen. Noch ein Trick: Bei den ersten Durchgängen muss man nicht so kräftig den Kopiering gegen die Schablone drücken wie beim letzten Gang, der dann für ein gleichmäßiges Fräsbild sorgt.

Beim Fräsen ist der Grad zwischen „perfekt“ und „Ausschuss“ oder auch „Gefahr“ sehr schmal (siehe Kasten). Kurz vor Ende des letzten Durchganges des zweiten Seitenteils dachte ich: „Ach, die paar Zentimeter noch, da brauch' ich nicht nochmal umgreifen“ und Peng! lief mir der Kopiering über die Schablone und der Fräser in mein Werkstück. Das Ergebnis sieht man im Bild 7.

Frustriert und geschockt habe ich die Fehlstelle mit Holzfüller gespachtelt und geschliffen. Um eine glatte Oberfläche und scharfe Kanten an der Fläche zu erhalten, habe ich das zweite Seitenteil mit Tape versehen und gegen das zu spachtelnde geschraubt 8. Nach dem Abbinden des Holzspachtels wurde mit einem Schleifklotz alles grob geschliffen, dann stimmte zumindest die Form wieder (auch wenn das schöne Linienmuster der Multiplexplatte an dieser Stelle gestört war).

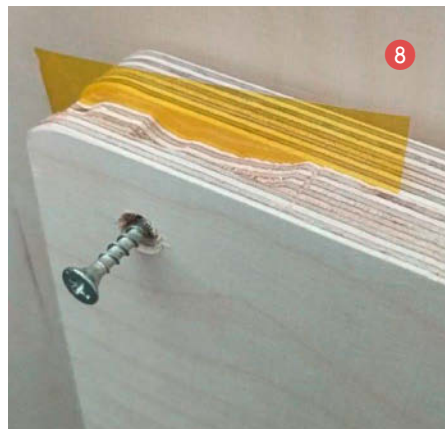
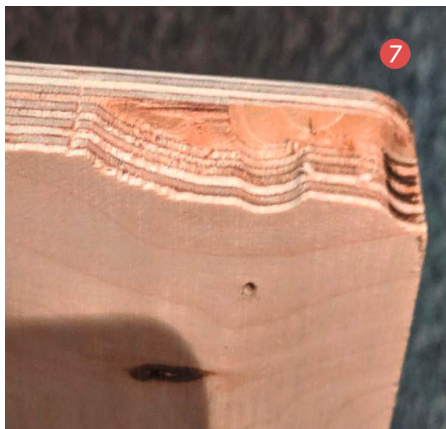


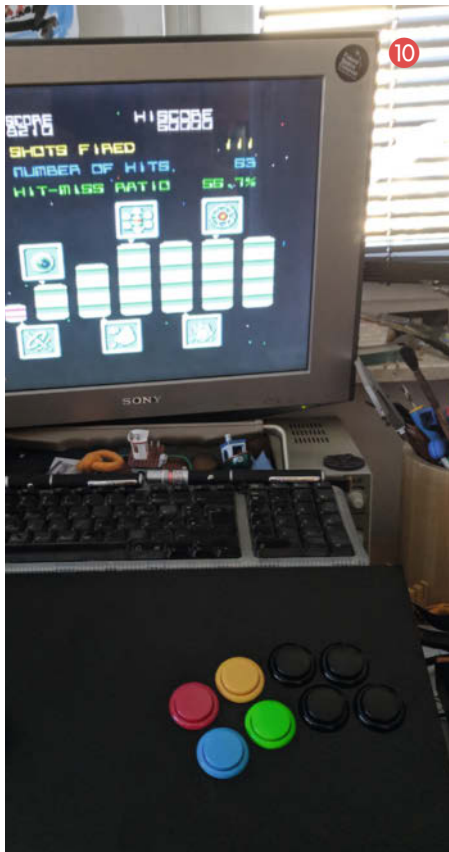
Vorsicht bei der Holzarbeit

- » Beim Sägen und Fräsen unbedingt Gehörschutz und Schutzbrille tragen.
- » Konzentriert und überlegt arbeiten und sich Zeit lassen.
- » Ablenkungen wie Handy, Kinder oder Haustiere ausschalten oder aus der Werkstatt verbannen.
- » Immer die Fräsrichtung beachten – man bewegt die Maschine immer *gegen* die Werkzeugrotation, weil die Rückschlaggefahr in umgekehrter Richtung deutlich größer ist. Wird *außen* gefräst am Werkstück (wie bei unserer Schablone), dann führt man die Fräse *gegen den Uhrzeigersinn*, jedenfalls wenn sich der Fräser im Uhrzeigersinn dreht, wie bei Oberfräsen üblich. Fräst man dagegen eine Tasche *im* Werkstück, führt man die Fräse *im Uhrzeigersinn*.
- » Beim Umgang mit Sekundenkleber unbedingt Augenkontakt vermeiden und darauf achten, sich nicht versehentlich die Finger zusammenzukleben, sie lassen sich nach dem Aushärten des Klebers nur noch unter Verletzungen trennen.

Nachdem die Seitenteile fertig (und repariert) waren, habe ich die Kanten mit einem Abrundfräser bearbeitet. Hierbei muss die Frästiefe genau eingestellt werden, damit man möglichst gleiche Radien auf beiden Seiten der Platte erreicht, sodass beide Viertelrundungen zusammen eine schöne halbe Rundung ergeben. Mit einem neuen und scharfen Fräser sollte so eine Oberfläche entstehen, die kaum mehr einer Nachbearbeitung bedarf.

Auf die Fehlstelle habe ich dann einen Farblaserausdruck der Multiplexschichtung mit der Farbseite zum Holz geklebt. Danach vorsichtig





Mehr Holzarbeit in Make

Die Arbeit mit Holz und Holzwerkstoffen ist Schwerpunkt unserer jährlichen Sonderausgabe, die auf halbem Weg zur nächsten regulären Make-Ausgabe am 5. November 2020 erscheint. Abonnenten erhalten dieses Heft im Rahmen ihres Abos automatisch ins Haus geschickt; sie wird aber auch im Zeitschriftenhandel erhältlich sein.



das Papier abgeschliffen, sodass die Farbe übrig blieb – da hatte ich mir eine echte High-Tech-Methode ausgedacht! Beim Nachbessern des Ergebnisses stellte ich fest, dass es mit paar Buntstiften und Lineal auch gegangen wäre. Aber man lernt ja nie aus 9.

Für die von unten einschiebbare Rückwand habe ich dann noch mit einem 4mm-Nutfräser Nuten gefräst. Diese Rückwand ist allerdings bis heute noch nicht fertig, dazu nachher noch mehr.

Für die Böden war dann nur noch genaues Messen (Tipp: Sägeblattstärke unbedingt mit einrechnen) und Sägen entlang der Schiene nötig, das war nach der Aufregung beim Fräsen schon fast erholend. Spannend wurde nochmal der Winkelschnitt für die untere Frontplatte und die Handauflage, aber auch hier kam der Holzspachtel und ein Schleifblock zur Hilfe, um die Ungenauigkeiten auszubügeln und einen gefälligen Übergang herzustellen.

Knopfleiste

Arcade-Knöpfe sind leider nicht international genormt: Während Japan-Style-Buttons metrische Maße haben, sind jene von US-Herstellern auf Bohrungsdurchmesser in Zoll/Inch-Maßen angelegt und brauchen beispielsweise ein Loch mit 1 1/4" Durchmesser (31,75mm). Die Forstnerbohrer für Löcher der nötigen Größe bekommt man hierzulande aber oft nur mit 30mm oder 32mm Durchmesser. Bei 30mm muss man nach dem Bohren die Löcher von Hand erweitern, was bei 15mm tiefen Bohrungen sehr nervig ist, bei 32mm wird das Loch schon etwas zu groß. Eine Standbohrmaschine ist für diese Arbeit praktisch, mit Vorsicht gelingt es aber auch mit einer Handbohrmaschine oder einem kräftigen Akkubohrer.

Ich habe mit 32mm gebohrt, der Ring am Knopf verdeckt das etwas zu große Loch und den etwas lockeren Sitz kann man mit etwas Heißkleber fixieren. Letzteres war bei mir am Ende gar nicht nötig, weil ich die Flächen meines Gehäuses inklusive der Handauflage foliert habe (dazu gleich mehr) 10. Ich habe die Bohrungen nicht komplett durchgebohrt, sondern noch etwas Auflagefläche gelassen, dies verhindert auch ein hässliches Ausreißen der Platte auf der Unterseite. Die Durchbrüche für die Kabel/Steckkontakte erstellt man genau deshalb auch am besten mit einem Stufenbohrer 11.

Bohrungen für eckige Buttons wie an der Stirnseite werden normal rund gebohrt und dann mit einer eckigen Raspel oder der Stichsäge in Form gebracht.

Monitorblende (Bezel)

Diesen Arbeitsgang habe ich mir bis zum Schluss aufgespart, denn hier kommt es auf jeden Millimeter an und es ist der Arbeitsschritt, der mir vielleicht die meisten Probleme

bereitet hat. Das ganze übrige Kabinett wurde zusammengeschrubt, um die exakten Maße daran abnehmen zu können. Erst in dieser Bauphase wurden dann auch die Klammern exakt dimensioniert und in 3D gedruckt, die den Monitor halten, da hier möglichst wenig Spiel sein sollte ¹².

Ich habe nach langem Messen und Rechnen zuerst einen Prototyp aus 5mm Depron gebaut – einer Art feines Styropor, das bei Modellfliegern beliebt ist und das man bequem mit einem Cutter schneiden kann (Pappe geht sicher auch). Nach diesem Prototypen und mit ein paar Korrekturen habe ich dann die endgültige Version aus Sperrholz gefertigt. Die Außenkanten nur mit einer Handkreissäge und Schiene maßhaltig und winkelig hinzubekommen war sehr aufwändig, da hier kleinste Abweichungen durch unschöne Lücken auffallen. Die Schnitte für die innere Öffnung, durch die man den Monitor sieht, habe ich entlang einer Schiene mit der Stichsäge ausgeschnitten. Danach mit der Fräse und einem 45°-Fasenfräser die Kanten abgeschrägt und dann die Ecken ausgefeilt, weil man ja mit dem runden Fräser nicht dort hineinkommt. Alles in allem ist dieses Teil das schlechteste meiner ganzen Bartop Arcade geworden, nur die aufgeklebte Folie kaschiert die schlimmsten Bausünden.

Das Aussägen der Lautsprecher-Durchbrüche unter dem Bildschirm war gemessen daran eine der einfacheren Aufgaben ¹³. Die muss es auch bei so einem Projekt geben.

Kantenversiegelung und Folieren

Die Sichtkanten der gefrästen Multiplex-Platten ¹⁴ wurden nur kurz mit einem Schleifschwamm (100er Körnung, nass) übergeschliffen. Dann kommt der eigentliche Trick: Die Kanten werden mit dünnflüssigem CA-Kleber getränkt, am besten in zwei Durchgängen. Man sollte nur darauf achten, keine Nasen an den Flächen der Platten zu produzieren, die man sonst schleifen muss. Abgesehen von der zusätzlichen Arbeit verändert der CA-Kleber die Holzstruktur und dadurch würde man diese Stellen später durch die Folie sehen. Ich habe den Kleber tatsächlich mit dem Zeigefinger verstrichen, da alle vorhandenen Handschuhe entweder zu grob waren oder zu gut klebten. Sicher nicht ideal – funktioniert aber, man sollte sich nur nicht an die Nase oder gar die Augen fassen, während der Kleber noch aktiv ist. Die Schicht CA ging im Laufe der nächsten Tage ab, ich wäre allerdings bis dahin durch keine Schleuse mit Fingerabdruck-Scanner gekommen.

Nachdem der Kleber getrocknet ist, schleift man einmal kurz mit dem nassen Schleifschwamm über und poliert danach mit einem Baumwolltuch. Voilà: Mit wenig Aufwand ent-



steht so eine glänzende und robuste Kante, die aussieht und sich anfühlt, als wäre sie lackiert.

Schon im Vorfeld der Planung (siehe Artikel im vorigen Heft) war klar, dass fertig schwarz beschichtete Platten den preislichen Rahmen sprengen würden und ein Lackieren aufgrund mangelnder Fertigkeiten und räumlicher Gegebenheiten nicht in Frage käme.

Nach langem Nachdenken und Recherchieren fand ich dann *d-c-fix*-Klebefolie. Tatsächlich wäre ich mit einer Rolle für etwa 8 Euro ausgekommen, aber Tests und ein Fehler machten dann zwei Rollen nötig (die zweite wurde aber inzwischen bei einem weiteren Projekt aufgebraucht). Letztlich hat sich gezeigt, dass die Folie nicht nur gut aussieht, sondern auch super zu verarbeiten und sehr haltbar ist ¹⁵. Zudem gibt es sie in verschiedenen Farben und Designs, in glänzend oder matt.

Für diesen Schritt ist es fast unabdingbar, eine weitere Person als Hilfe zu haben, denn ab einer gewissen Foliengröße braucht man vier Hände. Die zu folierenden Teile sollte man glatt schleifen (zuerst mit P120 auf dem

Schwingschleifer, dann 100er auf einem Schaumstoff-Schleifblock, dann 100er-Block nass), aber man muss es nicht übertreiben: So glatt wie fürs Lackieren muss es nicht sein, im Gegenteil, wenn man die Anmutung von lackiertem Holz erzielen will, schleift man nicht so penibel und dann zeichnet sich die Maserung des Materials durch die Folie ab.

Für das Folieren ist allerdings absolute Staubfreiheit wichtig, denn jeder noch so kleine Krümel auf der Klebefläche oder dem Holz ist später durch die Folie zu sehen. Ich habe das mittels Handstaubsauger und einem High-Tech-Staubwedel („Staubmagnet“) auf den Platten selbst erzielt und natürlich sollte der Raum, in dem man das macht, ebenfalls möglichst staubfrei sein. Ob man sauber gearbeitet hat, kann man mit einer Probeklebung mit einem kleinen Stück Folie testen, das man gleich wieder abzieht – wenn dann noch Staub an der Folie klebt, muss man nochmal sauber machen. Feucht zu wischen ist übrigens nicht zu empfehlen, da sich hier wieder Holzfasern aufstellen können.





Achtung: Gerade bei der Probeklebung die Folie nach spätestens einer Minute wieder abziehen! Wenn man die Folie erst mal anziehen lässt und nach mehr als ein paar Minuten wieder abzieht (wegen Fehlern, Falten oder Unebenheiten), werden Holzfasern aus der Oberfläche gerissen und man muss neu schleifen!

Die Folie habe ich mit der Hand und einem weichem Tuch aufgestrichen (ein Raket oder eine Folienkarte haben sich nicht bewährt, es fehlt das Gefühl für die recht weiche Folie). An den Rändern wurde die Folie mit einem Plastiklöffel (punktuelles Arbeiten als mit dem Raket) fest angedrückt, so dass man die Kanten beim folgenden Schneiden gut erkennt.

Nach dem Folieren beginnt man am Besten möglichst bald mit dem Schneiden der

Ränder, damit sich die Folie nicht um die Kante festzieht. Mit einer Hand die Folie leicht von der Kante wegziehen und mit der anderen (ruhigen) Hand und einem absolut scharfen Cutter entlang der Kante abschneiden, dabei das Messer auf der Kante abstützen, möglichst wenig absetzen pro Kante und nicht ins Holz schneiden. Klingt nach einem Kunststück, aber keine Angst: Wenn ich das schaffe, kann das jeder. An einem Probestück zu üben ist natürlich immer zu empfehlen. Kleine Wackler-Überstände kann man erst mal ignorieren und die später in Ruhe korrigieren. Kerben, wo zu viel Folie weggenommen wurde, kann man geistig ausblenden und ignorieren – wenn das Ding erst mal ein paar Stunden bespielt wurde, kommen ganz

sicher noch weitere als Gebrauchsspuren dazu.

Die Folie hat sich in der Praxis bewährt: Nach 24 Stunden war sie bombenfest und nach drei strapaziösen Tagen Maker Faire Berlin waren an der Bartop Arcade keine Probleme sichtbar ¹⁶. Eine ähnlich ansprechende Oberfläche per Lackieren hinzubekommen, erfordert sicher Profi-Fertigkeiten. Ich habe matte Folie verwendet und die sich abzeichnende Holzmaserung sieht durchaus gut aus. Möchte man glänzende Folie verwenden, sollte man die Flächen aber sicher noch glatter schleifen.

Aussicht und Fazit

Inzwischen versorgt ein 12V/5V-Kombi-Laptop-Netzteil sowohl den Raspi als auch den Monitor. Dabei wird der Raspi über die Micro-USB-Buchse versorgt, was weniger Probleme macht als über die GPIO-Pins. Weiterhin bin ich in Folge meines Tests der Spiele-Distributionen für Make 4/20, S. 8 zu Lakka als Retro-Distribution gewechselt.

Verbesserungen gäbe es noch einige umzusetzen:

- ein Ein/Aus-Taster, der den Pi sauber hoch- und herunter fährt
- ein solides Rückteil, das eine 230V-Kaltegerätebuchse und solche für USB und Netzwerk aufnimmt
- darüber eine transparente Rückwand, gegebenenfalls noch bunte LEDs oder gar *Ambilight* im Inneren
- ein Lüfter – denn wenn es mal eine Rückwand gibt, wird es drinnen vielleicht zu warm
- alle Spiele konfigurieren, besonders wenn man noch Heimcomputer-Spiele mit nur dem Joystick/Buttons bedienbar machen will (siehe dazu auch Make 4/20, S. 8)
- vernünftige Arcade-Hardware einbauen (meine ist etwas laut und vom Gefühl her klapperig)
- auch Akkubetrieb sollte gut möglich sein, da nur maximal 12V gebraucht werden.

Leider wird der Automat viel zu selten bespielt und ganz billig war es auch nicht, wenn man die eigens angeschafften Werkzeuge wie die Oberfräse mit einrechnet. Aber dennoch war die Bartop Arcade im Eigenbau ein tolles Projekt und ich habe ein Menge gelernt. Aber ein bisschen ist es wie bei einem Kind, das einen heißen Ofen anfasst: Man lernt daraus und macht es nicht wieder, wenn man darüber schlau geworden ist. Obwohl, so ganz stimmt das nicht mehr, denn inzwischen habe ich in einem weiteren Projekt Seitenteile für einen Synthesizer gebaut und dabei sowohl die Erfahrung mit der Plattenbearbeitung als auch die restliche *d-c-fix*-Folie nutzen können.

—pek



DIGITAL

— 2020 —

ITJOB SUMMIT

by Heise

Attraktive Arbeitgeber • Lebenslaufcheck • Spannende Vorträge

SAVE THE DATE

06.10.2020 • 23.10.2020

FINDE DEINEN TRAUMJOB ONLINE!

Aussteller

accenture



Bundeskriminalamt

BWI
IT für Deutschland

DEVK

KLEMM & Cie.
TALENT POOL

methodpark

.msg

QWARE
SOFTWARE ENGINEERING

real.digital
we ♥ digital commerce

SW//M

YESSPRESS

ZITIS

Unser Partner:

 **Jobware**
... da hab' ich den Job her!

JETZT KOSTENFREI REGISTRIEREN

WWW.IT-JOB-SUMMIT.DE

© Copyright by Maker Media GmbH

Tipps & Tricks

Was tun, wenn der Arduino bei einem Ihrer Projekte an akutem Pin-Mangel leidet? Wir wissen nicht, was Ihr Arzt empfiehlt. Wir empfehlen die trickreiche Doppelnutzung der vorhandenen Anschlüsse.

von Martin Ossmann

Taster und LED an einem Pin

Manchmal reichen die GPIO-Pins von einem Board wie dem Arduino nicht aus. Wenn man eine LED ansteuern und einen Taster abfragen will, kann man beide Aufgaben mit nur einem einzigen Pin und geschickter Programmierung erledigen, wie dieser Artikel zeigt. Einziger Nachteil dabei ist, dass die LED kurz erlischt, während man den Taster drückt. Das fällt aber in der Praxis kaum auf.

Die Verdrahtung **1** ist dabei erstaunlich einfach: Die Ausgangsstufe des GPIO-Pins bilden im Inneren des Arduinos zwei Transistoren *T1* und *T2*. Die Eingangsfunktion wird durch einen Schmitt-Trigger *S2* realisiert. Bild **2** zeigt Beschaltung des Arduinos im Probeaufbau.

Geschickter Wechsel zwischen In- und Output

Um die LED zum Leuchten zu bringen, schaltet man die GPIO-Funktion auf *INPUT*. Beide Transistoren sind dann im nichtleitenden Zustand. Die LED wird in dieser Phase über *R1* und *R2* mit Strom versorgt. Bei den angegebenen Werten ergibt sich ein Strom durch die LED von 5 mA, genug, um sie zum Leuchten zu bringen.

Um die LED zum Erlöschen zu bringen, schaltet man den GPIO auf *Output* und gibt ein *LOW* aus. Dabei wird Transistor *T2* in den leitenden Zustand gebracht. Die LED liegt dann mit einem Pol an GND, am anderen herrscht lediglich die an der Sperschicht von *T2* abfallende Spannung und die reicht nicht, um die LED zum Leuchten zu bringen.

Ob der Taster gedrückt wird, kann man feststellen, wenn die LED leuchtet. Das heißt, um den Taster abzufragen, schaltet man kurz die LED ein und fragt dann den Eingangspin ab. Diese Abfrage dauert weniger als 10 Mikrosekunden. Danach schaltet man die LED wieder für mindestens 1 Millisekunde in den ursprünglich gewünschten Zustand. Ist das *aus*, bewirkt die Pinabfrage nur ein Leuchten in einem hundertstel der Zeit (1 Mikrosekunde:

1 Millisekunde). Die LED scheint dann aus zu sein. Ist der ursprüngliche Zustand ein, spielt das ohnehin keine Rolle. Diese Leuchtdauer ist zu kurz, um sie wahrzunehmen.

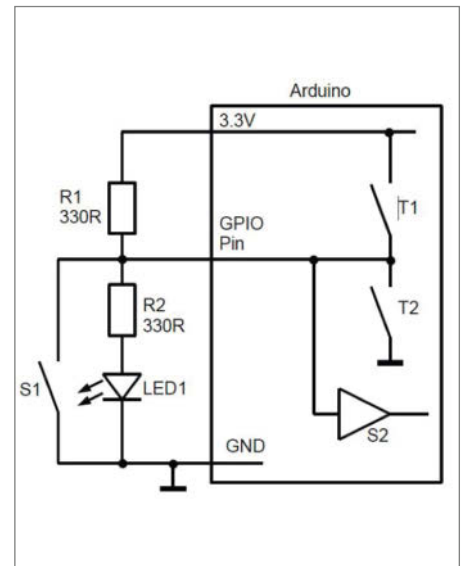
Diese Grundfunktionalitäten sind in **3** dargestellt.

In **4** sieht man ein Beispielprogramm mit einer sehr einfachen Aufgabe, die die Arbeitsweise dieses Tricks aber gut veranschaulicht. Wenn die Taste gedrückt wird, fängt die LED an zu blinken. Wenn man den Taster nochmal drückt, hört die LED auf zu blinken.

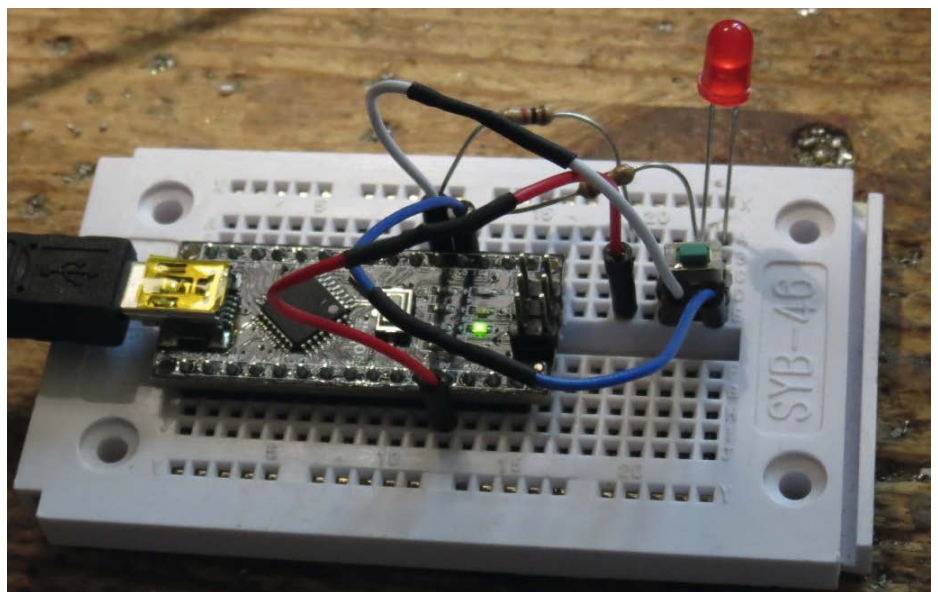
Wie man sieht, kann man die Doppelfunktion des Pins gut benutzen, um Tastaturfunktionalitäten mit Rückmeldung per LED mit nur einem GPIO-Pin realisieren. Selbstverständlich lässt sich diese Schaltung auch an mehreren Pins gleichzeitig verwenden. So kann man relativ großzügige Tastaturen und LED-Anzeigen selbst mit einem Arduino nano verwirklichen.

—hgb

► make-magazin.de/xcu5



1 So muss ein Pin beschaltet werden, damit er zugleich als Ein- und Ausgang verwendet werden kann.



2 Neben Taster und LED mit Vorwiderstand ist nur ein zusätzlicher Widerstand zur doppelten Pin-Nutzung notwendig. Den Rest erledigt die Software.

3 Die Grundfunktionen

```

01 int getDualUsePin1key(){
02   pinMode(DualUsePin1, INPUT);
03   int v=digitalRead(DualUsePin1) ;
04   if(DualUsePin1State){
05     setDualUsePin1() ;
06   }
07   else{
08     clearDualUsePin1() ;
09   }
10   delay(1) ;
11   return v ;
12 }
13
14 void setDualUsePin1(){
15   pinMode(DualUsePin1, INPUT) ;
16   DualUsePin1State=1 ;
17 }
18
19 void clearDualUsePin1(){
20   pinMode(DualUsePin1, OUTPUT) ;
21   digitalWrite(DualUsePin1, LOW) ;
22   DualUsePin1State=0 ;
23 }

```

4 Beispielprogramm

```

01 void loop() {
02   if(blinkMode){
03     blinkCnt++ ;
04     delay(10) ;
05     if(blinkCnt>=time1){
06       blinkCnt=0 ;
07       if(DualUsePin1State==0){
08         setDualUsePin1() ;
09       }
10       else {
11         clearDualUsePin1() ;
12       }
13     }
14     if(getDualUsePin1key()==0){
15       Serial.println("Set blinkmode off !") ;
16       while(getDualUsePin1key()==0) {} ;
17       blinkMode=false ;
18       clearDualUsePin1() ;
19     }
20   }
21   if(! blinkMode){
22     if(getDualUsePin1key()==0){
23       Serial.println("Set blinkmode on !") ;
24       while(getDualUsePin1key()==0) {} ;
25       blinkMode=true ;
26       setDualUsePin1() ;
27     }
28   }
29 }

```



Workshops 2020



Vertriebsperformance 4.0

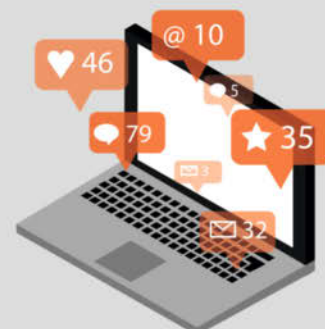
19. Oktober 2020 / Hannover

Erfahren Sie in diesem Workshop, weshalb alte Vertriebssteuerungsmodelle nicht mehr funktionieren werden und was es braucht, damit wir für Vertrieb 4.0 gerüstet sind.

Social Selling im B2B

27. Oktober 2020 / Hannover

In diesem Workshop lernen Sie, Ihre eigene Social Selling-Strategie zu entwickeln und diese erfolgversprechend mit den bestehenden Sales Aktivitäten zu verknüpfen.



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



**Ausprobiert
— von Make: —**

T7

Multi-Function-Tester



Tut es der gerade ausgelötete Transistor vielleicht doch noch? Bedeutet der Scheibenkondensator-Aufdruck „100“ nun 10 oder 100p? Welche Pinbelegung könnte der MOSFET aus der Wühlkiste haben? Solche Fragen beantwortet das pfiffige kleine Testgerät, das derzeit bei eBay & Co. angeboten wird. Es ist kaum größer als eine Zigarettenschachtel, kostet keine 20 Euro und ist mit einem kleinen Farbdisplay ausgestattet, das Messwerte und Pinbelegung übersichtlich anzeigt.

Die Bedienung beschränkt sich auf zwei Schritte (womit die fehlende Anleitung zu verschmerzen ist): Bauteil in der Textool-Fassung festklemmen (oder mit den mitgelieferten kurzen Prüfkabeln anschließen) und den „Start“-Knopf drücken. Bei passiven Bauteilen zeigt es den Wert mit brauchbarer Genauigkeit (2%) an, bei Dioden und Z-Dioden die Durchlassspannung und bei bipolaren Transistoren die Stromverstärkung (hfe), die Basis-Schwellenspannung und, falls messbar, auch den Reststrom (etwa bei den uralten Germanium-Transistoren, die wir ausprobierten). Bei MOSFETs ermittelt es sogar Gate-Kapazität, Gate-Schwellenspannung und On-Widerstand, so dass man den Transistor mit etwas Sachkenntnis grob einschätzen kann. Durch den eingebauten Spannungswandler werden sogar Z-Dioden bis 24V zuverlässig erkannt. Das Gerät bietet einen eingebauten Akku, der über die Micro-USB-Buchse geladen wird.

—cm

Vertrieb etwa eBay, AliExpress,
real.de
Preis 10 bis 20 €

Edu:Bit

Erweiterungsplatine und -Kit für den BBC Micro:Bit

Eine LED-Ampel, ein Mikrofon, Taster ... auf der großen roten Platine des Edu:Bit ist all das versammelt, was bei der Arbeit mit einem Mikrocontroller nützlich sein könnte. Das Erweiterungsboard für den BBC Micro:Bit stammt von der malaysischen Firma Cytron Technologies, misst 157,5mm × 124,5mm und ist damit deutlich größer als der Micro:Bit selbst, der einfach eingesteckt wird. Zur Ausstattung gehören außerdem ein Piezobuzzer, ein Potenziometer, ein Infrarotsensor und vier RGB-LEDs. Dank der zusätzlichen Spannungsversorgung und der drei Anschlüsse für Servos sowie zwei Ports für Gleichstrommotoren können auch gleich größere Projekte in Angriff genommen werden. Von Haus aus sind die Anschlüsse und Bauteile auf dem Board integriert, können aber auch abgebrochen und über Grove-Kabel wieder angesteckt werden.

Das Kit gibt es im Online-Shop von Cytron wahlweise mit oder ohne den Micro:Bit selbst. Es enthält weitere Teile, neben einem Anleitungsbuch einen Servo und einen Motor, Grove-Kabel und Karten für ein Spiel. Die Anlei-

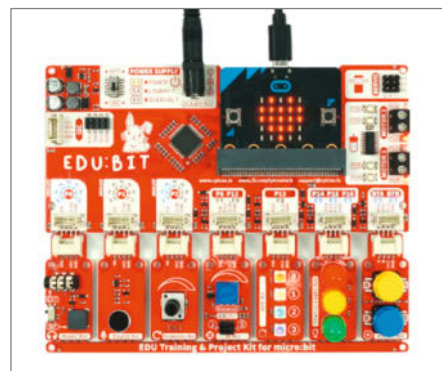


Bild: Cytron Technologies

tung enthält zehn englischsprachige Lektionen, um die Bauteile des Kits kennenzulernen und zu programmieren. —hch

Hersteller	Cytron
URL	cytron.io
Preis	49,75 US-\$ (mit Micro:Bit), 34,75 US-\$ (ohne)

ESP32-Audio-Kit

Sound-Erweiterung für den ESP32

Nachdem mit der ESP32-Cam bereits die Bildverarbeitung Einzug in die ESP-Welt hielt (Make 1/20, S. 28), ist nun auch Sound kein Problem mehr. Das unter verschiedenen Namen bei AliExpress erhältliche ESP32-Audio-Kit bringt alle Zutaten dafür mit: Neben einem ESP32-Modul (240MHz Taktfrequenz) mit WLAN und Bluetooth (inklusive BLE) enthält es auch zwei Mikrofone sowie Eingänge für externe Aufnahmegeräte (Line-In auf 3,5mm-Stereo-Klinkenstecker).

Außerdem sitzt auf der 7cm × 8,5cm großen Platine auch ein Stereoverstärker, der über Dupont-Steckverbinder ein Lautsprecherpaar mit bis zu 3 Watt Leistung (an 4 Ohm) versorgen können soll. Daneben steht ein Headset-Anschluss über die übliche 3,5mm-Klinkenbuchse zur Verfügung. Ein Speichermedium findet im Mikro-SD-Kartenslot (bis zu 64GByte) seinen Platz.

Die Stromversorgung des Audio-Kits kann entweder über eine Mikro-USB-Buchse (5V, 2A) oder über eine 3,7V-Lithium-Akkuzelle erfolgen. Die Ladeelektronik dafür ist ebenfalls auf dem Modul vorhanden. Ergänzt wird alles durch acht aufgelötete Taster, sechs an einem Posten-Steckverbinder anliegenden I/O-An-



Bild: AliExpress

schlüsse sowie eine auf eine zweite USB-Buchse geführte UART-Schnittstelle, über die auch das Flashen des Moduls erfolgen kann. Drahtlose Firmware-Updates soll das Modul ebenfalls beherrschen. —hgb

Vertrieb	AliExpress
Preis	10 bis 15 €

Calliope mini 2.0

Mikrocontroller-Board für die Grundschule

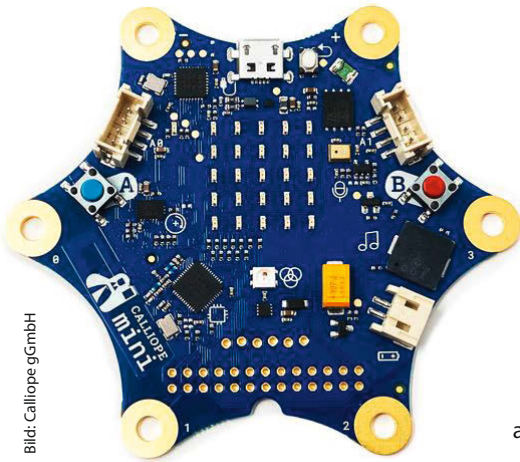


Bild: Calliope gGmbH

Der vor etwa vier Jahren zuerst auf den Markt gekommene Calliope mini (Make 2/17, S. 48) ist seit Mitte August in der Version 2.0 verfügbar. Ergänzt wurde die Platine um einen weiteren, größeren Flash-Speicher, wodurch bis zu 25 Programme auf dem Calliope mini selbst abgespeichert werden. Diese Neuerung trägt vor allem der Realität der IT-Ausstattung an Schulen und dem Lehrmaterial für den Mikrocontroller Rechnung: Denn die Calliope-Leitfäden stellen häufig verschiedene Programme vor, die für den Einsatz in unterschiedlichen Fächern gedacht sind. In Schulen ist aber häufig nur begrenzter Zugang zu PCs

möglich, so dass jede Neuprogrammierung organisatorisch aufwändig ist, und bisher konnte der Calliope mini nur ein einziges Programm zur Zeit speichern.

Die übrige Ausstattung mit Sensoren, Lautsprecher und Mikrofon, RGB-LED und LED-Matrix, Bluetooth und Motortreibern bleibt identisch, weshalb sich auch alle bisher veröffentlichten Calliope-Projekte uneingeschränkt auf der Version 2.0 umsetzen lassen sollten.

—esk

Hersteller Calliope gGmbH
URL shop.calliope.cc
Preis 37,90 €

Seeed Odyssey X86J4105

Single-Board-Computer mit Intel-Prozessor

Elektronik-Basteln wie mit Arduino und Pi, aber mit der Leistung eines Mini-PCs – das verspricht der Seeed Odyssey. Seine vier Intel-Celeron-Kerne (64 Bit) mit bis zu 2,5GHz Takt sind in der Lage, auch Projekte mit höheren Ansprüchen an die Rechenleistung anzutreiben. 8GByte RAM gepaart mit 64GByte sehr flinkem eMMC-Flash (lesen/schreiben 261131/187065 kByte/s) bieten genug Platz für laufende Anwendungen und Daten sowie Betriebssystem und Software. Vorinstalliert war auf unserem Board Windows 10, das wir jedoch durch einen Ubuntu (20.04) Desktop mit Gnome ersetzt, das sehr flüssig lief.

Anders als der Name es vermuten lässt, beginnt man mit dem Board keine Irrfahrt, sondern eher ein vielfältiges Abenteuer. Dazu bietet es neben den vielen PC-üblichen Schnittstellen wie 2 × USB3.1, 2 × USB3.0, 2 × Gigabit-Ethernet, SATA, HDMI und 2 × M.2 PCI (1 × M-Key, 1 × B-Key) insbesondere zwei Arduino- und Pi-kompatible Pin-Leisten. Die Raspberry-Pi-Leiste ist Pin-kompatibel zu herkömmlichen Erweiterungsplatinen (HATs). Hinter den Arduino-Pins steckt ein zusätzlicher ARM-Cortex-Mikrocontroller ATSAMD21G18, wie er etwa auch auf dem Arduino Zero eingesetzt wird. Den programmiert man mit der Arduino-IDE in gewohnter Weise, der Mikrocontroller ist intern fest mit einer seriellen Schnittstelle des Boards verdrahtet.

Die Kombination aus Mini-PC und echtzeitfähigem Mikrocontroller auf nur einer 11cm × 11cm großen Platine macht ihn zu einem praktischen System für Bastler, die eine kompakte Plattform für komplexere Steuerungen oder KI-Projekte suchen. Im Vergleich zu einem Pi 4B mit 4 bzw. 8GByte hat der Odyssey wenig überraschend in allen Belangen die Nase vorn. Mit 188 US-Dollar kostet der Odyssey ohne eMMC aber mehr als doppelt so viel wie ein Pi 4 mit 8GByte. Die Version mit eMMC kostet sogar 218 US-Dollar. Je nach Anwendungsfall lassen sich die höheren Kosten aber durch weniger Probleme bei der Treiberunterstützung und allgemein besseren Support wettmachen. Intel-Prozessoren – egal welcher Art – werden von allen Betriebssystemen und jeglicher Software in der jeweils



Ausprobiert
 —von Make:—

aktuellsten Fassung unterstützt, sodass es selten Inkompatibilitäten gibt, die den Basteltrieb bremsen. Bei Pi&Co. bekommt man meist nur gut abgehangene Software angeboten, die untereinander auch nicht immer kompatibel ist.

—dab

Hersteller Seeed
URL seeedstudio.com
Preis 188 US-\$

Benchmark

	Seeed Odyssey X86J4105	Raspberry Pi 4B 8GB 64 Bit
7z Compress MIPS	6586	3711
7z Decompress MIPS	8460	6099
mbw 100 (MEMCPY/DUMP/MCLOCK) MiB/s	3340/4903/6157	1885/1904/2771
USB 3.0 (MByte/s)	453	370
OpenSSL AES Speed (16384 Bytes) (MByte/s)	144	76
OpenCV (ms)	538	892
Tensorflow Lite (ms)	212	240
iperf (Mbit/s)	940	940

Ausprobiert
— von **tech-stage**

Anet ET5 Pro

FDM-3D-Drucker



Bild: Anet

Der Bauraum dieses 3D-Druckers ist satte 300mm × 300mm × 400mm groß. Neben einem zuverlässig arbeitenden Filament-Sensor ist auch ein fest installierter Näherungssensor für die Druckbettkalibrierung an Bord – dem Bausatz liegen sowohl eine Glasplatte als auch eine Kunststoffauflage für das beheizte Druckbett bei. Ein neues Mainboard und TMC2208-Schrittmotorentreiber sorgen für geringe Geräuscentwicklung. Der Bowden-Extruder erreicht eine maximale Temperatur von 250 °C.

Die kurze Aufbauanleitung ist ordentlich bebildert und leicht verständlich, der Zusammenbau unproblematisch und in einer halben Stunde zu schaffen. Die gesamte Konstruktion ist sehr stabil und hinterlässt einen positiven Gesamteindruck. Die Bedienung per Touchscreen ist dank des übersichtlichen Menüs unkompliziert, auf Deutsch kann man es aber nicht umstellen. Der Anet ET5 Pro liefert Out of the Box ordentliche und brauchbare Druckergebnisse. —pek

Einen ausführlichen Testbericht lesen Sie online.

► make-magazin.de/xmf5

Hersteller	Anet, anet3d.com
Vertrieb	direkt vom Hersteller oder über eBay
Preis	ab 390 €

Anet A8E

FDM-3D-Drucker

Nur noch der Name erinnert an den alten 3D-Drucker-Bausatz (siehe Make 1/18, S.34). Die Hardware des beim Hersteller vorbestellbaren A8-Nachfolgers ist komplett neu: Metall statt Plexiglas, stabil statt wackelig. Der Rahmen besteht aus Aluprofilen und soll dem Drucker die notwendige Stabilität für eine gute Druckqualität geben. Der Bauraum ist mit 220mm × 220mm × 250mm nahezu unverändert gegenüber dem Vorgänger. Der Drucktisch besteht nun jedoch aus einem Alurahmen, der aus einem Stück hergestellt ist. Ständiges Nachleveln sollte damit nicht mehr nötig sein.

Der A8E ist ein Bausatz, der allerdings nur noch aus drei Teilen besteht und in deutlich unter einer Stunde fertig sein soll. Der Drucker arbeitet mit 24V. Das halbiert bei gleichen Heizleistungen gegenüber dem notorisch brandgefährlichen Vorgänger die Ströme, sodass Kabel und Stecker nicht mehr außerhalb ihrer Spezifikationen betrieben werden und sich auch bei Vollast (Tisch und Druckkopf heizen gleichzeitig) kaum erwärmen dürften. Das neue Netzteil soll jetzt besser gegen Überlastung, Überspannung und zu hohe Temperaturen gesichert sein. Die Elektronik sitzt nun



Bild: Anet

in einem gut belüfteten Alugehäuse seitlich am Druckerrahmen. Die beiden Anschlussbuchsen für die Display-/Tastatureinheit sind von außen zugänglich. Das Display ist nicht mehr fest am Rahmen angebaut, sondern frei beweglich. Das erleichtert die Unterbringung des Druckers in Schränken oder Gehäusen erheblich. Der Bedienungskomfort wurde durch eine Filamentwechsel-Automatik deutlich erhöht: Ein Klick soll jeweils für das Ausstoßen beziehungsweise Einziehen des Filaments reichen. —hgb

Hersteller	Anet
URL	anet3d.com
Preis	172 US-\$

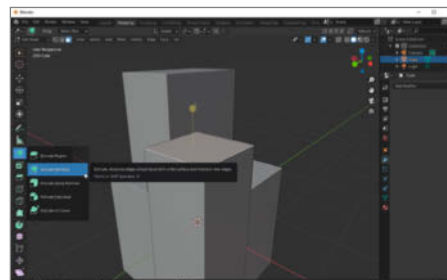
Blender 2.90

Open-Source-3D-Paket

Im Gegensatz zur 2.83 LTS-Version (Long Term Support für 2 Jahre) des Open-Source-3D-Pakets Blender werden in den regulären 2.9x-Versionen größere Veränderungen zugelassen. So ebnet das jetzt erschienene Blender 2.90 den Weg zu Version 3.0, die schon fast genau in einem Jahr erwartet wird. In Sachen Rendering (mit den Engines Cycles und Eevee), Animation und Sculpting gibt es viele Verbesserungen – dazu mehr über den Link unten. Hier hingegen zu den speziell für Maker interessanten Dingen:

An der Oberfläche wurde das Sortieren der Modifier Stacks und Constraints per Drag&Drop eingebaut und die Darstellung von Optionen in Panels lesbarer und kompakter gemacht. Szenenstatistiken (etwa zu Speicherbedarf, Objekt- und Flächenzahl etc.) können jetzt auch prominent als Overlay im Viewport angezeigt werden. In dem neuen Menu-Search (F3) bekommt man nun zusätzlich angezeigt, in welchem Menü sich eine Funktion versteckt.

Für das Modellieren besonders interessant ist das neue Werkzeug Extrude Manifold: Man kann damit ähnlich wie mit SketchUp arbeiten, etwa Teilflächen aus Körpern extrudieren,



wobei überzählige Flächen automatisch vereinigt oder gelöscht werden. Sehr praktisch ist auch das neue Snapping für Edge/Vertex Slide und das Maßband sowie absolute Maße für Bevels (Tool und Modifier). Blender läuft unter Windows, Linux und macOS und steht kostenlos zum Download zur Verfügung. —Carsten Wartmann/pek

► make-magazin.de/xmf5

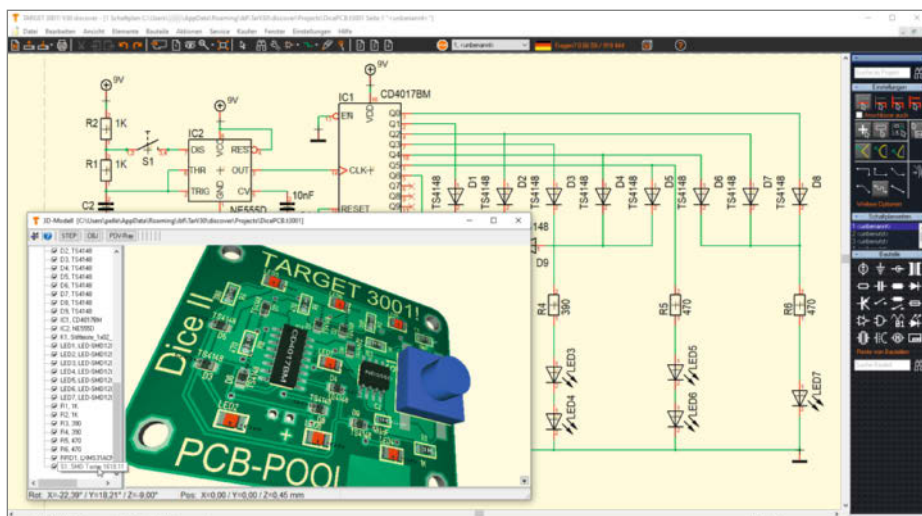
Hersteller	Blender Foundation
URL	blender.org
Preis	gratis

Target 3001! V30

Platinenlayout-Programm

Die neue Version der Software kam anlässlich des 30-jährigen Jubiläums heraus, daher die Versionsnummer – auch wenn nicht so viel neues gegenüber dem direkten Vorgänger V20 dazugekommen ist, wie der Zehnerschritt suggeriert.

Wer gern auf eigenwillige Autorouter verzichtet, wird am Pilot Router gefallen finden: Man gibt den Leiterbahnverlauf grob mit der Maus vor, die Leiterbahn wird dann automatisch unter Einhaltung der Design-Richtlinien kreuzungsfrei verlegt. Ähnlich funktioniert auch das interaktive Mäandrieren: Hochgeschwindigkeits-Signale (z.B. schon bei USB 2.0) verlangen gleiche Leiterbahnlängen, die man durch mäanderförmige Schleifen erzwingen kann. Zu den Arbeitserleichterungen gehört die neue Selektion von Leiterbahnabschnitten auch über Durchkontaktierungen hinweg, ebenso wie die praktische Sidebar mit häufig verwendeten Lieblingsbauteilen oder das automatische „Straffen“ von unnötig oft gewundenen Leiterbahnen. Mindestabstände lassen sich nun ganzen Signalklassen zuordnen, etwa allen Leiterbahnen mit Netzspannung. Und schließlich gibt es noch eine Optimierung beim Im-



port von STEP-3D-Daten, die zum Beispiel für eine detaillierte 3D-Ansicht oder für 3D-gedruckte Modelle der Baugruppe nötig sind: STEP-Bauteilformen mit Rundungen sind oft unnötig komplex, zur Verminderung des Datenaufkommens werden sie nun vereinfacht, ohne dass Details verloren gehen. —cm

Hersteller	Ing.-Büro Friedrich
URL	ibfriedrich.com
Preis	kostenlos (Discover-Version), 67,26 € (Light), 778,86 € (Economy)

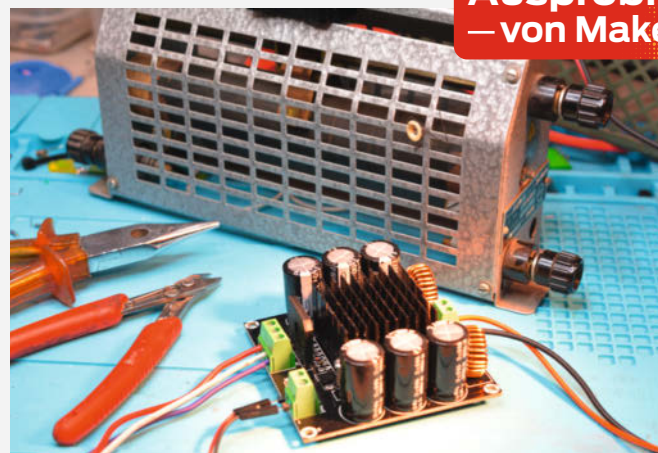
XH-M253

420W-Endstufenmodul

Class-D-Audioverstärker erbringen Höchstleistungen auf kleinstem Raum bei geringer Verlustleistung. Ob das nur 11 Euro teure Modul eines chinesischen Herstellers auch wirklich die versprochenen 420 Watt bringt, sollten unsere Messungen zeigen. Die mit dem ICTDA8954 aufgebaute Endstufe arbeitet im H-Brückenbetrieb (Bridge-Tied Load, BTL) mit einer empfohlenen Versorgungsspannung von $2 \times 25V$ AC; Gleichrichter und Siebkondensatoren (je $3 \times 2200\mu/50V$) sind gleich mit auf dem Baustein drauf. Als Stromversorgung reicht ein dicker Ringkerntrafo, der wenigstens 4A liefert. Wir haben an 8 Ohm gut 150W gemessen, an 4 Ohm waren es knapp 180W Sinusleistung. Kurz vor Begrenzungseinsatz zeigten sich hier Instabilitäten; man ist also gut beraten, sich mit 150W zufrieden zu geben, zumal der dann stark belastete Gleichrichter nicht gekühlt wird. Eine höhere Ausgangsleistung (nur an 8 Ohm!) ließe sich mit einem Trafo von $2 \times 30V/5A$ erzielen, dann würde die Leerlaufspannung aber schon die Spannungsfestig-

keit der Elkos strapazieren.

Bei näherer Betrachtung stellten wir fest, dass sich der Hersteller bei der weitgehend dem Datenblatt folgenden Schaltung einen argen Schnitzer erlaubt hat: Die dort angegebenen Dämpfungskondensatoren Csn1 bis Csn4 am Ausgang waren nicht mit 220p, sondern mit 220n bestückt, dadurch brannten die 10-Ohm-Widerstände Rsn1 und Rsn2 innerhalb von Millisekunden durch. Etwas dubios erscheint uns auch die Verwendung eines winzigen 0805-SMD-Kondensators im Ausgangsfilter, den wir vorsichtshalber durch einen hochwertigen Folienkondensator ersetzen. Mit der geänderten Bestückung erreichten wir einen angemessenen Klirrfaktor (0,06% bei 1W und



**Ausprobiert
— von Make: —**

0,2% bei mittlerer Leistung). Wer ein solches Modul kauft, sollte sich also auf einen gewissen Optimierungsbedarf einstellen und über das entsprechende Fachwissen verfügen. —cm

Vertrieb	Amazon, eBay und AliExpress
Preis	ab 10,99 € inkl. Versand

Praxisbuch Stop Motion Animation

Kreative Filme mit LEGO-Figuren



Einen Trickfilm selber zu machen ist ein ideales Projekt für die Corona-Zeit, nicht nur für Teams aus Eltern und Kindern: Das meiste, was man dazu braucht, ist schon im Haus, man ist lange damit beschäftigt und das Ergebnis lässt sich prima übers Internet verbreiten. Wie man ein solches Filmprojekt aus lauter Einzel-fotos konkret angeht, beschreibt der Autor in diesem Buch anschaulich und mit vielen praktischen Beispielen und Tipps garniert – etwa, dass für den eigenen *Brickfilm* auch schon ein Smartphone reicht, dass man bei Kamerafahrten lieber das Set als die Kamera bewegt und wie man mit etwas Bildbearbeitung Figuren überzeugend per Stop Motion durch die Luft fliegen lässt. Nebenbei lernt man auch erstaunlich viel über Arbeitsschritte und Kameraeffekte, die auch in „echten“ Kinoproduktionen zum Einsatz kommen.

Für alle, die beim Lesen die Lust packt, sofort selbst loszulegen (die zu wecken, schafft das Buch tatsächlich ziemlich gut!), streut der Autor zudem kleine Ideen für erste Szenen oder Kurzfilme ein – denn natürlich muss nicht jeder Trickfilm gleich so episch werden wie der „Untergang der Titanic“ und andere Werke von Hundbrax Lego Movies (siehe Make 6/16, S. 72). Viel Spaß beim Dreh! —pek

Autor	Alexander Altendorfer
Verlag	mitp
Umfang	192 Seiten
ISBN	978-3-7475-0248-8
Preis	19,99 € (Buch), 16,99 € (E-Book), 22,99 € (beides)

Programmieren lernen mit Python und Minecraft

Plugins erstellen ohne Vorkenntnisse

Dass man besser im Kontext persönlicher Interessen lernt, ist eine Binsenweisheit. Dass viele Programmier-Lehrbücher trotzdem für die meisten Leser maximal uninteressante Beispiele zur Erklärung heranziehen, ist ein großes Ärgernis. Das durchgehend farbige Buch von Daniel Braun macht es besser und kombiniert das populäre Aufbauspiel Minecraft mit dem Einstieg in die Programmiersprache Python.

Minecraft unterstützt nämlich in Python geschriebene Plug-ins, mit denen der Spieler neue eigene Funktionen einbauen und Abläufe automatisieren kann. So kann man mit einem Klick beispielsweise Mauern oder komplette Unterkünfte mit Innenausstattung direkt in seiner Minecraft-Welt erschaffen. Dabei führt der Autor mit einer leicht verständlichen und kurzweiligen Sprache nach und nach in alle

Grundelemente der Sprache ein. Zeitgleich lernt man die umfangreiche API von Minecraft kennen und erfährt, wie man Objekte in der Welt und deren Zustände abfragt oder auch Dinge manipuliert. Auch die einhergehende Einführung in die objektorientierte Programmierung, das Erstellen von Klassen und die Erklärung von Vererbung und Zugriffsmethoden ist dem Autor dank der gut gewählten Beispiele gut gelungen. Ob sich das Buch aber wirklich für Leser ohne jegliche Vorkenntnisse eignet, darf man bezweifeln. —dab



Autor	Daniel Braun
Verlag	mitp
Umfang	304 Seiten
ISBN	978-3747502600
Preis	24,99 € (Buch), 21,99 € (E-Book), 29,99 € (beides)

Git

Projektverwaltung für Entwickler und DevOps-Teams

Puh, mit 415 Seiten ist dieses Buch ein ganz schöner Klopfer. Und gleich zu Anfang machen die Autoren Öggl und Kofler klar: Sie kratzen damit gerade erst an der Oberfläche.

Doch keine Sorge: Das Buch ist kurzweilig und einladend geschrieben und das Autorenteam streut gerne interessante Trivia ein, um das trockene Thema Git ein wenig aufzulockern. Es beginnt mit dem Kapitel „Git in zehn Minuten“ (das in Wirklichkeit eher 25 Minuten in Anspruch nimmt) und geht anschließend äußerst detailliert auf die Anwendung von Git ein. Den Start machen klassische Features wie *Repos*, *Commits*, *Branches* und *Merging*, doch die Autoren behandeln auch wichtige – und oft übersehene Themen – wie Git-Netiquette, knackige Commit-Messages und gute Workflows. Dabei kann man alle

Kapitel unabhängig voneinander lesen und sich so die Infos rauspicken, die man gerade braucht. Schön ist der Fokus auf Dienste und

Tools wie *GitHub* und *GitLab* und ihre Alternativen wie *Azure DevOps*, *Bitbucket*, *Gitea* und *Gitolite*. Man erhält zwar einen guten Überblick, allerdings ist hier auch die Gefahr am größten, dass die Informationen im Buch schnell veraltet sind.

Wer nur mal ein bisschen in Git reinschnuppern möchte, wählt besser ein kurzes Video-Tutorial, doch wer beispielsweise in einem neuen Arbeitsumfeld schnell produktiv auf Git umsteigen muss, für den ist das Buch perfekt. —rehu

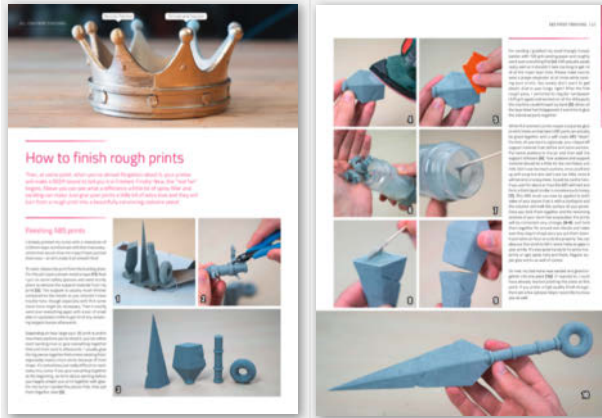


Autoren	Bernd Öggl und Michael Kofler
Verlag	Rheinwerk Computing
Umfang	415 Seiten
ISBN	978-3-8362-7186-8
Preis	39,90 €

The book of 3D-Printing

Modeling, finishing & more

„The book of 3D-Printing“ vom deutschen Cosplayerteam *Kamui Cosplay* richtet sich zwar an Cosplayer, enthält aber Unmengen von Wissen, das für jeden 3D-Druck-Novizen nützlich ist. Nach dem Einstieg in die üblichen Fragen wie „Was ist 3D-Druck?“ erklären die Autoren, wie man passende 3D-Modelle findet oder sie mit Autodesk Fusion und Blender modelliert.



Im Abschnitt zum „perfekten Finish“ kann man von Cosplayern besonders viel lernen: Hier liest man praktische Tricks, um 3D-Drucke zu kleben, Techniken zum Vorbehandeln und Grundieren der Modelle und die passenden Hilfsmittel, um die Drucke perfekt glatt zu zubern. Dabei geht Kamui Cosplay auf die Bearbeitung von ABS, PLA und Resin ein – letzteres ist besonders praktisch, da Resindruck noch so neu ist, dass er in vielen Tutorials keine Erwähnung findet.

Zum Abschluss zeigt das Duo Beispiele, bei denen sich 3D-Druck besonders gelohnt hat und wie sie die Objekte modelliert, gedruckt und nachbearbeitet haben. Lobenswert: Im ganzen Buch wird immer wieder auf Arbeitssicherheit hingewiesen. Die Publikation räumt außerdem mit den Vorurteilen der Cosplay-Community auf, dass 3D-Druck nur eine faule Abkürzung sei. „The book of 3D-Printing“ ist leider bisher nur auf englisch erhältlich. —rehu



Autoren	Svetlana Quindt und Benjamin Schwarz
Verlag	Eigenverlag
Umfang	48 Seiten
Link	www.kamui cosplay.com/product/3dprinting/
Preis	PDF 4,99 US-\$, Softcover 19,99 US-\$

Die Erfinder Werkstatt

Tolle Projekte aus Schaltkreisen und Leuchtdioden

Mit diesem Buch hat man einen Überblick über die Grundlagen von Stromfluss und Elektronik in der Hand (geeignet ab 10 Jahren). Auf den ersten 39 Seiten werden die Werkzeuge, ihre Handhabung und die grundlegenden Informationen über Stromfluss vorgestellt. Erst ab Seite 40 beginnen die Anleitungen für die 18 Projekte.

Diese Projekte sind klassische Elektronik-Anfänger-Projekte. Dazu gehören beispielsweise eine Münzen-Batterie, ein Elek-

tromotor aus einer Batterie, einem Dauermagneten und einer Spule und ein Generator aus einer Kurbel und einem Elektromotor. Die Projekte sind gut konstruiert – der Autor hat für Probleme wie zum Beispiel die Aufhängung einer Spule praktische und saubere Lösungen. So werden für den Bau des Elektromotors zwei Sicherheitsnadeln sehr elegant gleichzeitig als Stromanschlüsse für die Batterie und Aufhängung für die Spule verwendet.

Am Ende jedes Projekts gibt es eine kurze Erläuterung der Grundlagen – der Autor erklärt, wie der jeweilige Aufbau funktioniert. In diesen Erklärungen werden die üblichen Formulierungen und Fachbegriffe verwendet – sie repetieren Informationen, ohne sie wirklich anschaulich zu machen. Das reicht für viele Zwecke aus, trotzdem wäre es schöner, wenn die Erklärungen ein wenig alltags-tauglicher wären. —esk



Autor	Jack Challoner
Verlag	Dorling Kindersley
Umfang	160 Seiten
ISBN	978-3-8310-3917-3
Preis	16,95 €

IMPRESSUM

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), Helga Hansen (hch), Carsten Meyer (cm), Rebecca Husemann (rehu), Elke Schick (esk)

Mitarbeiter dieser Ausgabe: Beetlebum, Hans Borngräber, Guido Burger, Richard Fix, Michael Gaus, Christoph Goebel, Klaus-Uwe Gollmer, Matthias Helneder, Matthias Jünemann, Miguel Köhnlein, Moritz König, Dominik Laa, Christian Moll, Martin Ossmann, Hannah Perner-Wilson, Mika Satomi, Maik Schmidt, Josha von Gizycki, Alexander Wankler, Carsten Wartmann, Stefan Wollner

Assistenz: Susanne Cölle (suc), Christopher Tränkmann (cht), Martin Triadan (mat)

Leiterin Produktion: Tine Kreye

DTP-Produktion: Martina Bruns, Martin Kreft (Korrektur)

Art Direction: Martina Bruns (Junior Art Director)

Layout-Konzept: Martina Bruns

Layout: Nicole Wesche

Fotografie und Titelfeld: Andreas Wodrich

Digitale Produktion: Anna Hager, 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

Geschäftsführer: Ansgar Heise, Dr. Alfons Schröder

Verlagsleiter: Dr. Alfons Schröder

Stellv. Verlagsleiter: Daniel Bachfeld

Anzeigenleitung: Michael Hanke (-167)
(verantwortlich für den Anzeigenteil), mediadaten.heise.de/
produkte/print/das-magazin-fuer-innovation

Leiter Vertrieb und Marketing: André Lux (-299)

Service Sonderdrucke: Julia Conrades (-156)

Druck: Dierichs Druck + Media GmbH & Co.KG,
Frankfurter Str. 168, 34121 Kassel

Vertrieb Einzelverkauf:

VU Verlagsunion KG
Meßberg 1
20086 Hamburg
Tel.: 040/3019 1800, Fax.: 040/3019 145 1800
E-Mail: info@verlagsunion.de
Internet: www.verlagsunion.de

Einzelpreis: 10,90 €; Österreich 11,90 €; Schweiz 18,00 CHF;
Benelux, Italien, Spanien 11,90 €

Abonnement-Preise: Das Jahresabo (7 Ausgaben) kostet
inkl. Versandkosten: Inland 65,10 €; Österreich 66,50 €;
Schweiz/Europa: 72,10 €; restl. Ausland 88,20 €

Das Make-Plus-Abonnement (inkl. Zugriff auf die App, Heise Select sowie das Make-Artikel-Archiv) kostet pro Jahr 6,30 € Aufpreis.

Make: Nächste Ausgabe erscheint
am 10. Dezember 2020

Abo-Service:

Bestellungen, Adressänderungen, Lieferprobleme usw.:

Maker Media GmbH
Leserservice
Postfach 24 69
49014 Osnabrück
E-Mail: leserservice@make-magazin.de
Telefon: 0541/80009-125
Telefax: 0541/80009-122

Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden. Kein Teil dieser Publikation darf ohne ausdrückliche schriftliche Genehmigung des Verlags in irgendeiner Form reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Alle beschriebenen Projekte sind ausschließlich für den privaten, nicht kommerziellen Gebrauch. Maker Media GmbH behält sich alle Nutzungsrechte vor, sofern keine andere Lizenz für Software und Hardware explizit genannt ist.

Für unverlangt eingesandte Manuskripte kann keine Haftung übernommen werden. Mit Übergabe der Manuskripte und Bilder an die Redaktion erteilt der Verfasser dem Verlag das Exklusivrecht zur Veröffentlichung. Honorare Arbeiten gehen in das Verfügungsrecht des Verlages über. Sämtliche Veröffentlichungen in Make erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes.

Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.

Published and distributed by Maker Media GmbH under license from Make Community LLC, United States of America. The 'Make' trademark is owned by Make Community LLC. Content originally partly published in Make: Magazine and/or on www.makezine.com, ©Make Community LLC 2020 and published under license from Make Community LLC. All rights reserved.

Printed in Germany. Alle Rechte vorbehalten.
Gedruckt auf Recyclingpapier.

© Copyright 2020 by Maker Media GmbH

ISSN 2364-2548

Nachgefragt

Welche Retrotechnik sollte man mal neu erfinden?

Josha von Gizycki

Braunschweig, erklärt auf Seite 34, wie man Pi-hole einrichtet

Fahrzeuge stecken voller Firlefanz. Der heutigen Stand der Sicherheit mit dem Funktionsumfang von vor 50 Jahren, reduziert aufs Wesentliche – das könnte ruhig mal wieder erfunden werden.

KOBAKANT:

Hannah Perner-Wilson und Mika Satomi

Berlin, geben auf Seite 102 einen Einblick in die KOBAMAßschneiderei

Die Uhr. Wie wäre es mit einer ganz einfachen Old-School-Uhr, die die Uhrzeit ohne jeden Schnickschnack

anzeigt? Man trägt sie am Handgelenk, stellt die Zeit selbst ein und jede geht ein bisschen falsch, so dass jeder in seiner eigenen kleinen Zeitzone lebt. Vielleicht muss auch gar nicht die Technologie selbst neu erfunden werden, sondern nur die Kultur um sie herum.

Matthias Jünemann

Nordkirchen, baut auf Seite 114 einen Fernsehlift

Ich fände es klasse, einen alten Hebdrehwähler aus der Telefonvermittlung wiederzubeleben oder neu zu bauen. Vielleicht lässt sich ein Projekt finden, in dem man ihn mit einem modernen µC nutzen kann ...

Inserentenverzeichnis

ALLNET Computersysteme GmbH, Germering	39
dpunkt.verlag GmbH, Heidelberg	71
Mouser Electronics Inc., München	140
Reichelt Elektronik GmbH & Co., Sande	9

Rheinwerk Verlag GmbH, Bonn	19
Tragant GmbH, Berlin	23
Make:markt	51

Back to BASIC

Exklusiv
im heise shop!

+ Nano-Axe-
Board mit
PICAXE-08M2



NEU

Make Picaxe Special

Noch einfacher als Arduino: Im neuen PICAXE Special der Make dreht sich alles um den Einstieg ins Programmieren mit BASIC. Dazu gibt es ein neuentwickeltes Programmierboard für den Einsatz von PICAXE-Chips, das Nano-Axe-Board mit USB-Anschluss. Damit können Sie sofort starten!

shop.heise.de/make-picaxe

Einzelheft
für nur

24,95 € >

Generell portofreie Lieferung für Heise Medien- oder Maker Media Zeitschriften-Abonnenten.
Nur solange der Vorrat reicht. Preisänderungen vorbehalten.

© Copyright by Maker Media GmbH.

 **heise shop**

shop.heise.de/make-picaxe >

Innovation, Auswahl und Service in perfekter Harmonie

Millionen Produkte –
unbegrenzte Designoptionen



Noch heute versandfertig
mouser.de/te-selection