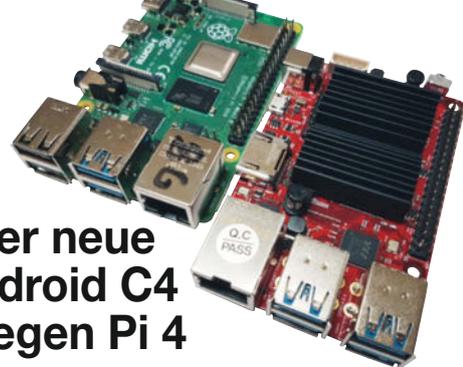


Der neue
Odroid C4
gegen Pi 4



Retrogaming mit dem Raspberry Pi

- ▶ Von Spielhallenklassikern bis Playstation 2
- ▶ Kostenlose Distributionen: Lakka, RetroPi & Co
- ▶ Stilechtes Arcade-Gehäuse im Eigenbau



Projekte

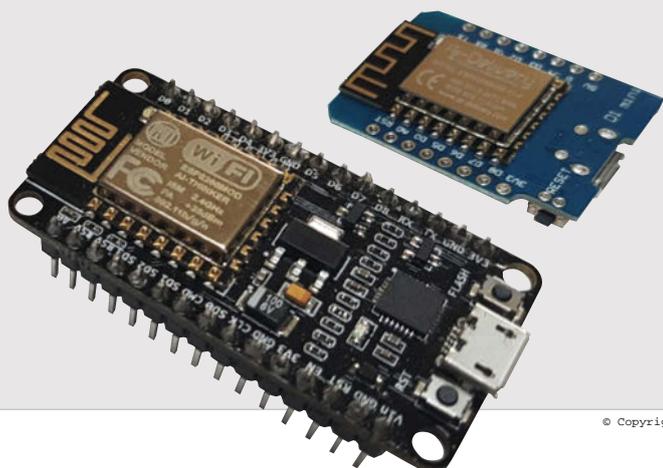
- ▶ Animatronics: Lebensechte Roboter
- ▶ Nabendynamo lädt Smartphone

Werkstatt

- ▶ Stempel konstruieren mit FreeCAD
- ▶ CNC-Fräsen im Workshop bauen

ESP32/8266

- ▶ Schneller zum Smart Home mit alternativer Firmware
- ▶ IoT-Button: Eigene Aktionen über Webdienste auslösen



3D-Druck

- ▶ Test: Gratis-Slicer-Software
- ▶ So geht's: Druckserver OctoPi einrichten



4/20

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

€ 10,90

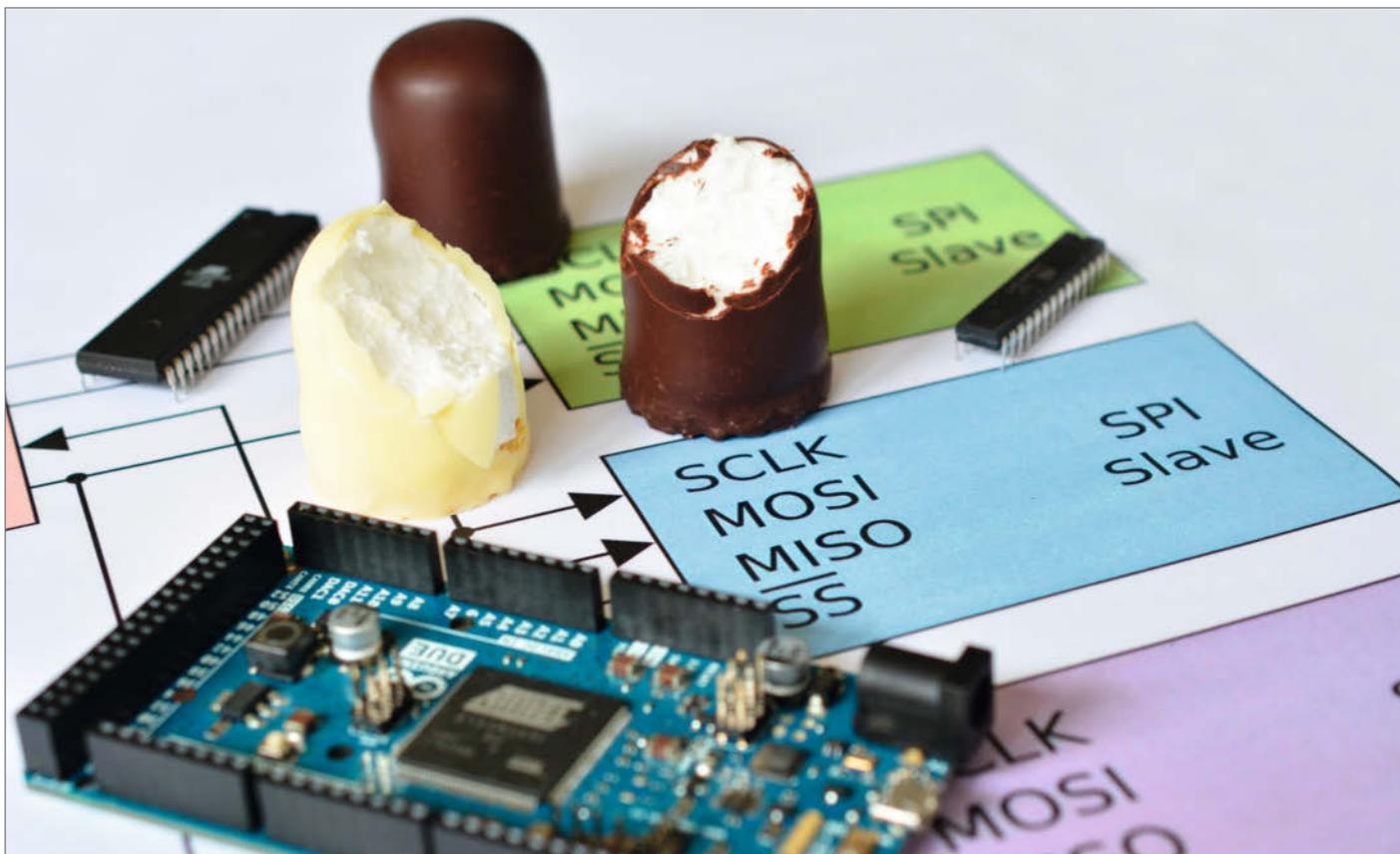


Maker Faire®

Dieses Jahr ist
alles anders – wir
vermissen euch!

**Freut euch
auf 2021.**

maker-faire.de



Ende der Sklaverei

Politisch korrekte Kräfte sorgen sich im Rahmen der BLM-Debatte an ungewohnten Stellen für eine Sensibilisierung im Umgang mit der Sprache: Die Open Source Hardware Association hält die bisherige Benennung der SPI-Signale (eine serielle Schnittstelle an vielen Mikrocontrollern und Peripheriebausteinen) für nicht mehr zeitgemäß und schlägt eine Umbenennung vor. Statt MOSI (Master out, Slave in) und MISO (Master in, Slave out) sollen die Signale nun SDI und SDO (Serial Data in/out) heißen, auch das (mir schon aus anderen Gründen zuwider) SS-Signal nennt sich nun CS (für Chip Select statt Slave Select). Alternativ sollen CIPO und COPI Controller und Peripherie designieren.

Die Streitfrage, ob bestimmte Begriffe der IT-Welt diskriminierend sind, entzündete sich bereits Anfang Juli an den Hackerbezeichnungen White Hat und Black Hat, manche halten gar den Begriff Server (von lat. servus = Diener, Sklave) für ablösenswert. Finde ich diese Diskussion übertrieben? Nein, sie zeigt, wie unüberlegt wir manchmal mit unserer Sprache umgehen. Ich unterstelle einmal, dass Jack Kilby in seinem JK-Flipflop keineswegs aus rassistischen Motiven einen Master- und Slave-Teil vorgesehen hat und auch Motorola bei der SPI-Spezifikation einfach nur gedankenlos bestehende Termini weiterverwendete.

Doch genau diese Gedankenlosigkeit ist der Stein des Anstoßes: Warum auch sollten harmlose Schaltungen und Datenstrukturen an das finstere Kapitel der Sklaverei erinnern? Nicht umsonst haben

Sprachwissenschaftler des texanischen Global Language Monitor die Master/Slave-Terminologie bereits 2004 zum Unwort des Jahres erkoren, was sich zuallererst einige Open-Source-Projekte zu Herzen nahmen.

Das bedeutet nicht, dass ich nun hunderte meiner Schaltbilder und zehntausende Zeilen Quellcode nach historisch vorbelasteten Signal-, Variablen- und Funktionsnamen durchforste – das wird keinen rassistischen Cop und keinen identitär Bewegten sinneswandeln. Es bedeutet auch nicht, überstürzt neue Nomenklaturen vorzuschreiben, die dann allzu leicht in eine Euphemismus-Tretmühle geraten: Hat nicht auch das Substitut Parent/Child einen autoritären Beigeschmack?

Es hilft, behutsam und ohne viel Aufhebens neutrale Begrifflichkeiten einzuführen – „Controller“ und „Peripherie“ sind schon mal ein brauchbarer Ansatz, weil sie keinen Bezug auf Menschen und unrühmliche Ereignisse nehmen. Allzu lautes Säbelrasseln jedoch wird nur wieder jene Kleingeister auf den Plan rufen, die bei Facebook „Teile diesen Negerkuss, um linksgrünversifft Gutmenschen zu ärgern!“ posten.

Carsten Meyer

Carsten Meyer

Sagen Sie uns Ihre Meinung!

mail@make-magazin.de

Inhalt

DIY-Animatronics

Mit einem Kurzbesuch im Bastelkeller ist es bei unserem spektakulären Animatronics-Projekt nicht getan: Die künstlich intelligente, animierte Tesla-Büste entstand in ungezählten Arbeitsstunden. Deutlich weniger Arbeitseinsatz verlangt unser kleiner Minion.

- 68 Report: Animatronics
- 70 Animatronischer Tesla
- 78 Animatronic Minion



RasPi-Retrogaming

Um stilechte Arcade-Automaten selbst zu bauen, benötigt man keinen Hochleistungs-PC – für die Emulation fesselnder Retro-Spiele reicht auch ein RasPi-Einplatinenrechner, dem man eine Gaming-Distribution verpasst. Wir zeigen, welche Komponenten man braucht und wie das passende Gehäuse entsteht.

- 8 Retro-Gaming auf Raspberry Pi
- 14 Bartop Arcade mit Raspberry Pi

- 3 Editorial: Ende der Sklaverei
- 6 Leserforum
- 8 **Retro-Gaming auf Raspberry Pi**
- 14 **Projekt: Bartop Arcade mit Raspberry Pi**
- 22 **Test: Odroid C4 gegen Pi 4**
- 27 3D-Kurs für Maker: Konstruieren mit Blender 2.8
- 28 Werkstattberichte: Neues aus der Szene, Comic
- 30 Was uns inspiriert: Messing-Gameboy, Lichtuhr, mechanisches Orchester
- 34 **Smarthome-Firmware für ESP-Module**
- 44 Leuchtendes LED-Herz aus Acrylglas
- 50 **Schalter zur Welt: Der IoT-Button**
- 58 **Power aus dem Nabendynamo**
- 64 **Makerinnen: Die CNC-Dozentin**
- 68 **Report: Animatronik**
- 70 **Animatronischer Tesla**

Wochenend-Projekte

Maken von verspielt bis praktisch: Ein raffinierter Trick bei unserem LED-Herz lässt auch die Augen des Betrachters leuchten, und unsere Nabendynamo-Ladeschaltung erfreut den Radfahrer selbst auf ausgedehnten Touren mit einem stets proper geladenen Smartphone.

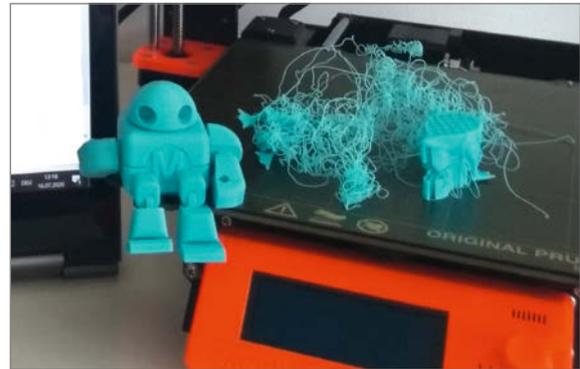
- 44 Leuchtendes LED-Herz
- 58 Power aus dem Nabendynamo



3D-Druck

Ob ein 3D-Druck zum Erfolgserlebnis wird, entscheidet auch der Slicer – ein Programm, das den Entwurf in Scheiben zerlegt und Stützstrukturen einbaut. Wir haben drei prominente Kandidaten verglichen. Vieldruckern zeigen wir, wie man einen 3D-Druckserver einrichtet.

- 90 Slicer-Shootout
- 98 Druckserver für 3D-Drucker
- 106 Stempeln mit FreeCAD



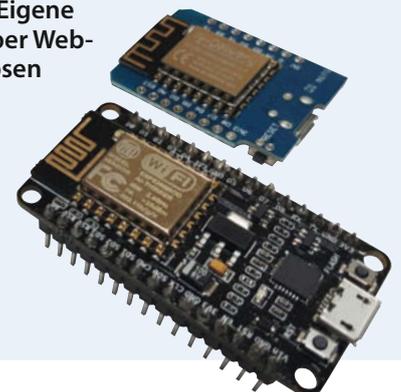
- 78 **Animatronic Minion**
- 82 Please Leave a Message: Schafenster-Plotter
- 84 UV-Kammer für Resindrucker
- 86 MicroscopY: das Lego-Mikroskop
- 88 Reingeschaut: Super8-Tonfilm-Kamera
- 90 **Vergleichstest: Slicer Shootout**
- 98 **Druckserver für mehrere 3D-Drucker**
- 106 **Stempeln mit FreeCAD**
- 116 Kurzvorstellungen: 3D-Drucker, CNC-Fräsen, Oszilloskop, CAD, Raspizubehör, Tastatur-Controller-Boards
- 120 Bücher: CAD, Roboter mit Fischertechnik, 3D-Scannen, Küche als Labor, Modellbau-Kniffe
- 122 Nachgefragt
- 122 Impressum

Themen von der Titelseite sind rot gesetzt.

Smart-Home-Ideen

Wenn man Web-Dienste geschickt einbindet, reicht für das eigene IoT-Device auch ein ganz simples Programm für den ESP8266 – oder Sie spielen einfach eine der fertigen Firmware-Distributionen auf, die wir hier vorstellen: Dann wird das Home auch ohne Coden smart.

- 34 Smart Home mit alternativer ESP-Firmware
- 50 IoT-Button: Eigene Aktionen über Webdienst auslösen



Leserforum

Widerstand fehlt

LCD-Oszilloskop für Digital-Signale,
Make 3/20, S. 110

In der Materialliste zum Artikel fehlt ein 10-Kiloohm-Widerstand. Der sitzt als Pullup-Widerstand zwischen der Reset-Leitung und Vcc. Im Schaltplan auf Seite 112 ist er korrekterweise enthalten. Wir bitten, diesen Fehler zu entschuldigen.

Bahnhof?

Hardware steuern mit TouchOSC,
Make 3/20, S. 94

Da es um ein generelles Problem geht, schicke ich dies an die Redaktion und nicht den Autor. TouchOSC soll nur als Beispiel dienen. Es ist ja nun im Leben immer so: Wenn man etwas verstanden hat, ist alles ganz einfach. Vorher versteht man nur 'Bahnhof'. Viele Autoren tun sich nun schwer damit, den Leser im Status 'Bahnhof' abzuholen. Ist doch alles so klar. Wird doch alles bis ins Detail beschrieben. Richtig. Aber mir fehlt der Überblick. Wie hängt das alles zusammen?

TouchOSC hat sofort mein Interesse geweckt. Ich wollte schon immer mal mit dem Smartphone meinen Arduino steuern. Also, da gibt es eine Software fürs Smartphone für ganz tolle Bedieneroberflächen. Aber halt, das sieht ja aus wie ein Windows-Programm. Arti-

kel nochmal gelesen. Aha: Unter Windows erstellen, dann hochladen aufs Smartphone. Ist doch klar. Und dann braucht man noch ein Programm auf dem Arduino. Kein Problem, da kenne ich mich aus. Aber was zum Teufel macht das HTML in einem Arduino-Sketch?? Artikel nochmal gelesen. Aha, da läuft ein Web-Server auf dem Arduino. Ist doch klar. Und wo rufe ich die Webseite jetzt auf? Auf dem PC? Auf dem Smartphone? Und warum eigentlich? Da werde ich den Artikel wohl noch mal lesen müssen ... Und dann gibt es da noch den *TouchOSC Tester* und die *TouchOSC Bridge Configuration* und den *TouchOSC Bridge File Manager*. Ich fürchte, ich werde den Artikel noch öfter lesen müssen.

Ihr habt das Problem für einen Neuling verstanden? Ich glaube, wichtig ist das große Ganze, das Prinzip, die Zusammenhänge. Eine kleine Skizze, wie alles zusammenspielt, wäre Gold wert gewesen. Die Kleinigkeiten und Details kann man meist schnell im Netz finden. Also bitte in Zukunft an den 'Bahnhof' denken.

Jan Meyer

Sie sprechen ein Problem an, das mit jedem etwas anspruchsvolleren Artikel einhergeht: Was muss ich als Autor des Beitrags beim Leser vor-aussetzen? Was kann ich weglassen, damit ich den in der Thematik vorgebildeten Leser nicht langweile und von der Länge des Artikels her im Rahmen bleibe? Im vorliegenden Fall haben wir schlicht vorausgesetzt, dass sich der Leser bereits

mit dem IoT-Chip ESP8266 beschäftigt hat und weiß, dass man diesen sehr bequem mit der (über die erwähnten Libraries erweiterten) Arduino-IDE programmieren kann; zu diesem Thema haben wir bereits unzählige Artikel und sogar ein Sonderheft veröffentlicht. Passende Hardware ist im Kurzinfo-Kasten aufgezählt. Mit der klassischen Arduino-Hardware hat der Artikel nichts zu tun, diese wurde im Beitrag auch an keiner Stelle erwähnt. Aber Sie haben recht, eine Übersichtsskizze hätte dem Artikel nicht geschadet. (Carsten Meyer)

Tolles Heft

Makers care!, Make 3/20, S. 80 und
Community-Projekt: Bobby-Car mit
Hoverboard-Antrieb, Make 3/20, S. 88

Was für ein tolles Heft! Hab ich mich vor einigen Monaten noch beschwert, weil ich bestimmte Schwerpunkte vermisst habe und auch die durchgängige Nachvollziehbarkeit der Projekte für mich nicht immer erkennbar schien, finde ich nun zwei an meine „Wünsche“ angelehnte Projekte. Zum einen das ESP-Dashboard, zum anderen das Analysieren des IKEA-Wifi-Lautsprecher!

Seite 80 und 88 werde ich sinngemäß bei Gelegenheit verknüpfen – also den Hack eines medizinischen Gerätes mit Teilen eines Hoverboards. Es geht um einen von der Krankenkasse bereitgestellten Rollstuhl, der nicht umgebaut werden darf. Für hügeliges Gelände wäre aber eine kleine elektrische Unterstützung hilfreich. Vielleicht gelingt es mir, ein altes Hoverboard beziehungsweise den Antrieb am Rollstuhl einzuhängen und gegen die Reifen des Rollstuhls zu pressen, um dann für eine Unterstützung beim Schieben sorgen zu können. Wenn Sie hier hilfreiche Ideen hätten, könnte das von mir bald umgesetzt werden.

Mark Haacke

Vielen Dank für Ihre Mail und das Lob! Ein Hinweis zu Rollstühlen: Der Verein be-able hat vor einigen Jahren mit Made for my Wheelchair einige Nachbauhacks entwickelt und veranstaltet regelmäßig Hackathons, auf denen weitere Projekte entwickelt werden. Diese sind leider nicht so ausführlich dokumentiert, aber vielleicht kann man Ihnen beim Verein trotzdem weiterhelfen:

- ▶ <http://www.madeformywheelchair.de>
- ▶ <https://be-able.info/de/be-able/>

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.



Jetzt
bewerben!

www.zfuw.de

FERNSTUDIUM NEBEN
DEM BERUF

SCIENCE & ENGINEERING

- Brandschutzplanung
- Financial Engineering
- Medizinische Physik /
Medizinische Physik
und Technik
- Nanobiotechnology
- Psychologie kindlicher
Lern- und Entwicklungs-
auffälligkeiten
- Software Engineering
for Embedded Systems
- Technoethik

Alles ist gut

Prime und Netflix mit dem Raspberry Pi,
Make 3/20, S. 60

Ich hatte in letzter Zeit schon öfter überlegt, die Make zu kündigen. Ist mein letztes Heise-Abo nach iX und c't. Es sind für mich zu wenig interessante Elektronik-Basteleien drin. Aber dann kommt ein Heft wie dieses, und alles ist gut! Ab Seite 60 habe ich nicht weitergelesen, sondern habe mit RasPi 4 meine 4 Jahre alte Kodi-Installation (auf RasPi 2) erneuert. Die Anleitung für Netflix und Amazon prime hätte ich mir sonst mühevoll aus den entsprechenden Foren zusammensuchen müssen. Hat auf Anhieb geklappt – ich kann die kleinen Eigenheiten von Kodi ja schon. Die File-Server-Einbindung kann manchmal nerven. Also kein Gedanke mehr an Kündigung!

Otto Fürcho

Der Kracher

Tongeber, Make 2/20, S. 98

Ohne Erwartungen an das Ergebnis habe ich mich an den Bau der Lautsprecher gemacht. Einfach weil es eine interessante Herausforderung war – kaum Erfahrung mit dem Löten oder im Arbeiten mit Holz – so konnte ich nur etwas lernen. Das Ergebnis – der Kracher! Nie hätte ich gedacht, dass man soviel Klang für so wenig Geld bekommt. Einfache Holzarbeiten, Frequenzweiche löten – kann ich nun auch. Bin begeistert! Ich habe die Boxen gleich weiter modifiziert, mit einem Dayton Audio WF60PA Plate Amplifier, passt super.

Marten Neubauer



Bild: Marten Neubauer

Seine ersten Boxen hat Marten Neubauer nach unserer Anleitung gebaut – und war vom Klang überrascht.



Bild: Michael Stamm

Michael Stamm hat sein Tomatengießer-Projekt mit Display und Arduino Uno realisiert.

Tomatenautomat

Der Pflanzengießer für die Ferien hat mich dazu gebracht, es noch mal mit der Aufzucht von Tomaten zu versuchen. Da ich sonst das regelmäßige Gießen vergessen habe, und entsprechend die Ernte dann gering war, ist das Problem jetzt hoffentlich gelöst. Mit den Messführlern und den Pumpen aus dem Artikel ist die automatische Bewässerung von drei Tomatenpflanzen entstanden. Damit nicht nur an einer Stelle bewässert wird, habe ich drei Wasserverteiler gedruckt. An die wird der 9/6 Schlauch direkt aufgesteckt.

Michael Stamm

Die STL-Datei für den Wasserverteiler gibt es jetzt über unsere Github-Seite zum Projekt zum Download, vielen Dank, dass sie uns und den Lesern die Datei zur Verfügung gestellt haben!

► make-magazin.de/xhju

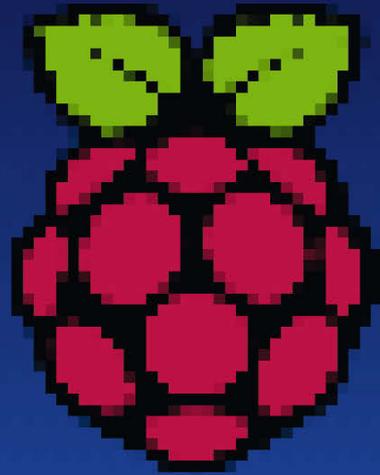
Phantastische Comics

Comic: Kolophonium, zuletzt in
Make 3/20, S. 32

Und hier endlich ein großes Dankeschön von mir zu den phantastischen Comics von Kolophonium in der Make! Möge er noch lange für uns weiter zeichnen!

Peter Leutwiler

Dem Wunsch und dem Lob schließt sich die Redaktion von ganzem Herzen an.



Grafik: Peter König

Retro-Gaming auf Raspberry Pi

Möchte man Spieleklassiker vergangener Jahrzehnte spielen, gibt es viele Optionen – günstig und vielseitig ist es, einen Raspberry Pi als Retro-Konsole zu benutzen. Wir haben vier darauf spezialisierte Distributionen für das Raspi-Betriebssystem mal ausprobiert.

von Carsten Wartmann

Mehr zu Retro-Gaming

Bartop Arcade mit Raspberry Pi

Seite 14

Emulatoren für alte Software gibt es viele, doch die Rasperry Pis verschiedener Generationen sind wohl die flexibelsten und am besten unterstützten Minicomputer fürs Retro-Gaming. Deshalb konzentrieren wir uns im Folgenden auf diese Plattform. Dennoch: Fast alle im Artikel beschriebenen Distributionen unterstützen auch diverse andere Unterbauten und laufen etwa auf PCs (mit Intel/AMD-, aber auch Mac-Hardware), weiteren Single-Chip-Computern wie Odroid und oft auch Android-Set-Top-Boxen und ähnlichem. Genaueres dazu finden Sie in unserer Übersichtstabelle. Und falls Retro-Spiele, speziell auf dem Raspi, für Sie Neuland sind, haben wir ein kleines Glossar zusammengestellt, in dem die Begriffe kurz erklärt werden, die beim ersten Auftreten im Artikel fett gesetzt sind.

Welchen Raspi braucht man?

Es muss nicht immer das neueste Modell sein: Ein Raspi Zero, 1 oder 2 **emuliert** viele Systeme der 8- und 16-Bit Konsolen-Ära, ein Raspi 3 hat genug Power für PS1-Spiele und auch für einiges von N64, Dreamcast und PSP. Der neue

Glossar

Bezel: Grafische Rahmen um das eigentliche Spiel herum, die Bildschirmrahmen wie früher zeigen, oder auch das System selbst oder zum Spiel passendes Artwork

Core: Emulator für spezielles Spielesystem
EmulationStation: Grafische Oberfläche, aus der heraus Spiele gestartet und das System konfiguriert werden kann

emulieren, Emulator: Simuliert die Hardware einer anderen (oft veralteten) Computerhardware, sodass alter Code auf neuen Systemen läuft

GPIO: Frei benutzbare Ein-/Ausgabe-Pins am Raspberry Pi, um etwa Joysticks, Schalter, aber auch originale Hardware anzusprechen

Homebrew: Von Privatpersonen erstellte Spiele auf Systemen, die vom Hersteller nicht zur Ausführung selbst erstellter Programme gedacht sind

Hotkey: Taste/Button, die oder der zusammen mit anderen Buttons oder

dem Joystick das Menü oder spezielle Funktionen des Emulators aufruft (Spielstände, Reset, Cheats etc.)

Libretro: API/Library, um auf einfache Weise Spiele und Emulatoren zu schreiben, die sich wie Plug-ins in die Retro-Distributionen einbinden lassen

RetroArch: Generalisierte Oberfläche, um Emulatoren aus dem Spiel heraus zu bedienen und zu konfigurieren

ROM: Bezeichnung für eine Kassette, die auf ROM-Chips die Spiele enthält; in der Emulatorszene auch zum Beispiel für C64-Spiele auf Diskette verwendet

Shader: Bearbeiten das vom Emulator-Core ausgegebene Bild. Das Ziel kann eine Verbesserung durch Interpolation, aber auch das Simulieren einer Bildröhre, eines Gameboy-Displays oder der Farbfehler auf alten TVs sein.

Themes: Verändern das Aussehen der Menüs von Distribution und Emulatoren

Raspi 4 sollte hier noch etwas Luft nach oben haben, die Unterstützung in den Distributionen ist aber noch sehr frisch.

Die Raspi-Versionen 1 und 2 neu zu kaufen lohnt sich nicht mehr, aber wenn man so ein Board gebraucht bekommt oder aus der Schublade zieht, ist es immer noch gut für ein paar Spiele zwischendurch. Der Raspi Zero ist wegen seines Formfaktors für Handhelds gut zu gebrauchen. Zu beachten ist, dass der Raspi 1 noch SD-Karten benötigt, alle anderen Micro-SD. Eine Class-10-Karte ist kein Luxus, was die Geschwindigkeit angeht, dies macht sich aber fast nur beim Booten, Beschreiben und Backup der Karten bemerkbar.

Da der Raspi 4 praktisch das gleiche wie ein Raspi 3+ kostet, sollte man bei einer Neuanschaffung für Spielezwecke kurz prüfen, ob es bei der gewünschten Distribution noch Probleme mit dem neuesten Modell gibt. Der heftige Aufpreis für einen Raspi 4 mit mehr als 2GB Speicher lohnt sich für unseren Zweck übrigens nicht.

Das Drumherum

Einen Raspi Zero oder 1 sollte man übertakten, sonst ist er zu lahm. Dies geschieht bei manchen Distributionen per Menü, ansonsten über die Datei `config.txt` auf der auch unter Windows lesbaren ersten Partition der (Micro-)SD-Karte. Diese Datei ist ebenfalls die erste Anlaufstelle, falls kein Full-HD-Monitor, sondern stiehlt ein VGA- oder Composite-Monitor oder ein LCD

RetroPie

RetroPie ist die Retro-Umgebung für fast alle modernen Computer. Entweder installiert man RetroPie auf einem vorhandenen Linux oder benutzt eine komplette RetroPie-Distribution, die es für den für Raspi und viele andere Plattformen gibt.

Nachdem man den Pi beim Booten auf der Textkonsole beobachten konnte, wird gleich das Gamepad (sofern eingesteckt und erkannt) konfiguriert. Hierbei sollte man die Erklärung beachten, was man bei nicht vorhandenen Tasten machen soll (lange auf eine bereits konfigurierte Taste drücken).

Danach ist man im Konfigurationsmenü und kann das System anpassen. Ich empfehle hierfür eine Tastatur anzuschließen, denn einige Konfigurationstools arbeiten im Textmodus und sind mit Gamepads nur umständlich bedienbar ❶. Dies ist vor allem bei der `raspi-config` der Fall – die braucht man aber, wenn man etwa auf einem Raspi 1 unterwegs ist und den übertakten will.

Vordefinierte Inhalte oder andere Menüpunkte gibt es zu diesem Zeitpunkt noch

Kurzinfo

- » Retro-Gaming-Distributionen im Kurzporträt
- » Spielen wie früher, heute auf dem Einplatinencomputer
- » Stärken und Schwächen der Distributionen im Vergleich

Mehr zum Thema

- » Heinz Behling, Prime und Netflix mit dem Raspberry Pi 4, Make 3/20, S. 60

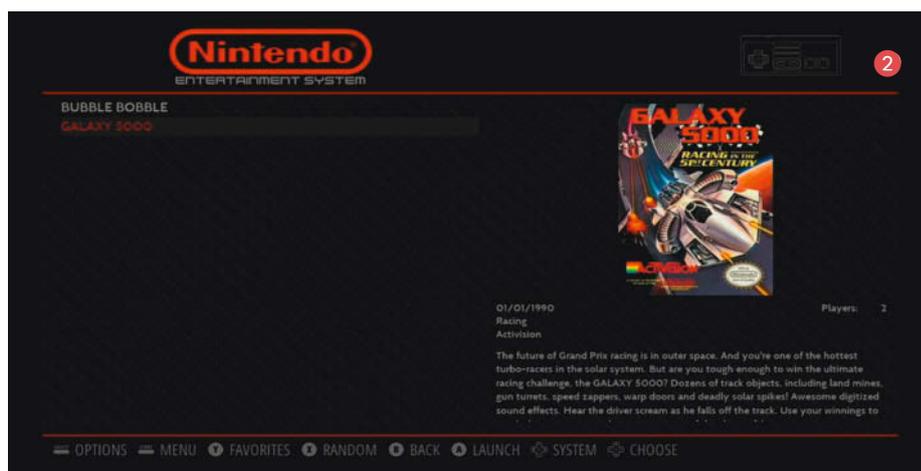


angeschlossen werden soll. Für einen Einsteiger und nicht Linux-affinen Maker mag das erst mal abschreckend erscheinen, es ist aber ganz einfach und gut dokumentiert (siehe Link in der Kurzinfo). Benutzt man hingegen einen Raspi 3 und ein HDTV, so braucht man sich um all das nicht weiter zu kümmern.

Für die ersten Tests der Installation und deren eventuelle Konfigurationen ist eine USB-Tastatur angebracht, mit ihr kann man auch die Spiele ausprobieren. Auf Dauer wird man aber ein USB-Gamepad anschließen wollen. Je nach Distribution wird dieses automatisch erkannt und konfiguriert oder man muss mit der Tastatur die Konfiguration starten. Weitere Möglichkeiten sind drahtlose Gamepads oder Arcade-Hardware, die per USB oder die **GPIO**-Pins des Raspi angeschlossen wird.

Die Software

Alle getesteten und hier vorgestellten Distributionen sind Open Source und können frei heruntergeladen, genutzt und kopiert werden. Die Mehrheit der **ROMs** beziehungsweise Spiele sind allerdings durch Copyright und Urheberrecht geschützt und dürfen nicht weitergegeben werden. Der einzig legale Weg ist hier, die eigenen ROMs oder Disks auszulesen, ohne einen Kopierschutz zu umgehen. Aber dieses Problem stellt sich ja auf allen Retro-Gaming-Plattformen und wird an dieser Stelle nicht vertieft. Manche der im Folgenden vorgestellten Distributionen haben zudem einige ROMs freier Spiele gleich mit an Bord – zum ersten Ausprobieren reicht das in jedem Fall.

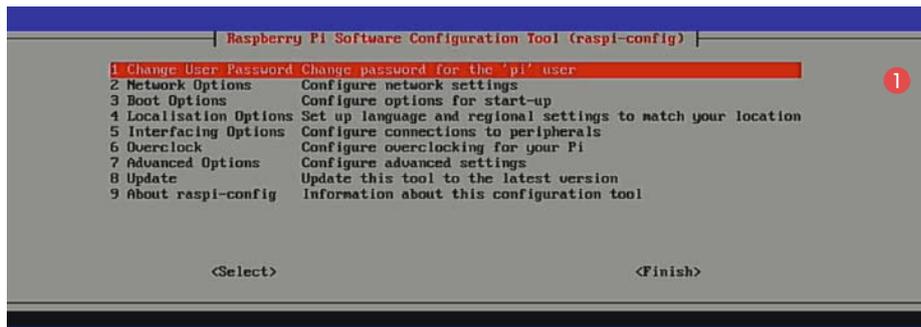


nicht. Also lädt man vom PC aus auf die Laufwerksfreigabe des Pi (unter `\retropi` oder dessen IP-Adresse) seine Game-ROMs in die passenden Ordner, etwa `\roms\nes` hoch und startet dann neu (zumindest die **Emulation Station**).

RetroPie ist die Distribution, bei der man so ziemlich alles selbst einstellen und konfigurieren kann, sei es an der GUI oder für jedes Spiel bis auf CPU-Level einzeln. An vielen Stellen scheint aber der Linux-Charme durch, etwa in Gestalt des Textmodus, den man auch zu

sehen bekommt, wenn man Games lädt. So etwas kann den Eindruck eines System sehr schmälern, gerade wenn man drumherum ein hübsches Retro-Gehäuse baut. Ja, irgendwie kann man das wegkonfigurieren, aber dazu muss man sich in die Tiefen der *config*-Files begeben. Die Hardware-Unterstützung ist so gut wie die der jeweiligen Linux-Distribution unter RetroPie beziehungsweise entspricht den eigenen Linux-Systemkenntnissen. Bei der Raspberry-Pi-Version ist das Basissystem Raspian, was natürlich der eine oder andere Raspi-Besitzer schon gut kennt.

Nach der Installation hat man die Aufgabe vor sich, alle Emulatoren und die einzelnen Spiele zu konfigurieren ② – und das kann sehr



aufwändig werden, weil man keine direkte Unterstützung durch die Distribution bekommt. Sowohl die Dokumentation als auch

die Community von RetroPie sind aber umfangreich, deshalb findet man fast immer eine passende Lösung – irgendwann.

Recalbox

Recalbox (ja, mit nur einem „l“) kommt einem sehr bekannt vor, wenn man schon RetroPie kennt, nutzt doch auch diese Distribution EmulationStation als Unterbau. Aber die Linux-Basis versteckt sich hier wesentlich besser hinter diversen Splash- und Lade-Screens ③. Wenn es um die Installation und Konfiguration geht, bekommt auch ein absoluter Linux-Laie keine Panik, alles geht geführt und ohne Editor von sich. Selbst wenn das Gamepad nicht erkannt wurde, wird einem spätestens nach ein paar panischen Tastendrücken vom System aus die Möglichkeit angeboten, das Pad zu konfigurieren. Eine positive Überraschung!

Insgesamt ist Recalbox besser für Einsteiger geeignet als RetroPi. Leider wird der Raspi 4 noch nicht unterstützt. Wenn man möchte, kann man fast jeden Aspekt (Buttons, Darstel-



lungsqualität, etc.) der Emulationen pro Emulator oder sogar pro Spiel einstellen. Dazu ruft man mit **Hotkey** und B-Button das **RetroArch**-Menü auf (wobei man den Hotkey selbst definiert, üblicherweise *Select*). Bei Recalbox geht die Konfiguration – wie auch bei anderen modernen Distros – über die Bühne, ohne je

einmal den Textmodus oder einen Editor benutzen zu müssen.

Zum schnellen Start und Erfolgserlebnis kommt Recalbox gleich mit einer guten Auswahl an freien ROMs. Ich empfehle hier auf jeden Fall *Blade Buster* für das NES zu testen – ein klasse Shooter ④.



Lakka

Lakka hat eine sehr moderne Oberfläche und erinnert so eher an eine moderne Spielekonsole ⑤. Die GUI basiert auf dem **Libretro**-Frontend und ist im Gegensatz zu EmulationStation nur wenig „themebar“, allerdings mit dem Vorteil, dass man auch nur eine Bedienung verinnerlichen muss und nicht geistig zwischen RetroArch, Libretro und EmulationStation umschalten muss. Mit Hotkey+Start kommt man aus dem Spiel in das Menüsystem des Emula-

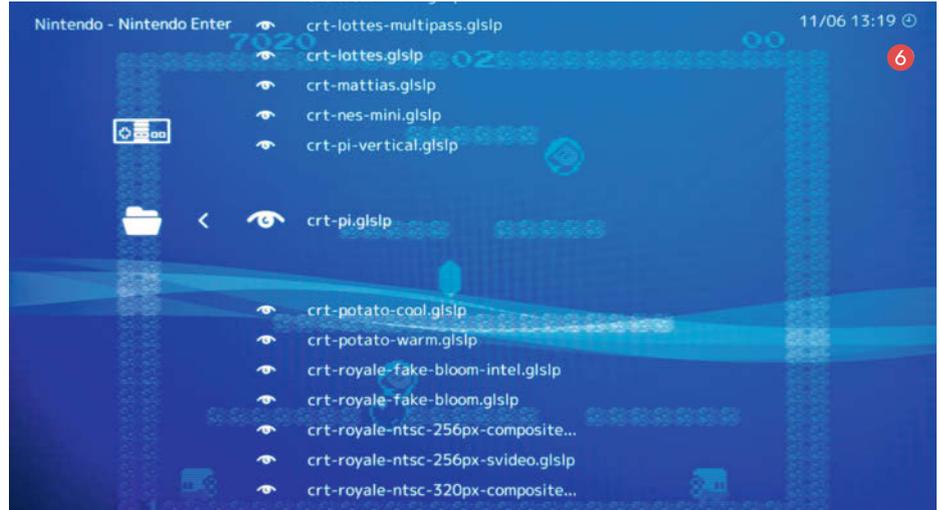
tors (das Spiel ist im Hintergrund abgeblendet zu sehen) und kann so einsteigerfreundlich Screenshots/Spielstände anlegen oder das Spiel und die **Shader** konfigurieren ⑥. Zwar ist man mit den Button-Kombinationen bei anderen Distributionen schneller, aber man kann sich diese in Lakka auch selbst als *Hotkey Binds* definieren. Insgesamt ist bei Lakka am wenigsten zu konfigurieren, bis man endlich mit dem Spielen loslegen kann, weil vieles sinn-

voll voreingestellt ist oder automatisch eingerichtet wird.

Durch die schlanke Architektur bootet Lakka auch auf einem Zero oder Raspi 1 schnell. Die Dokumentation ist gut und die Foren sind ergiebig, ich konnte mit dieser Hilfe alle Startschwierigkeiten lösen. Man muss etwa ssh und Samba nach der Installation einschalten, was fundamental anders ist als bei anderen Distributionen, wo zumindest Samba Standard ist.

Anfangs muss man einmal den zu verwendenden Emulator (bei Lakka als **Core** bezeichnet) pro Spiel aussuchen. Leider gibt es dabei keine direkte Hilfe, wann man welchen wählen sollte, geschweige denn, dass immer die richtige Plattform vorgeschlagen wird. Also muss man selbst probieren, falls das nicht aufgrund eigenen Retro-Wissens klar ist. Gerade auf langsameren CPUs muss man gegebenenfalls die Grafik- oder Soundfehler einiger Cores hinnehmen, um auf spielbare Geschwindigkeiten zu kommen. Ein Raspi lässt sich unter Lakka, anders als bei den anderen Distributionen, nur per Editieren der *config.txt* übertakten. Das ist nicht sehr kompliziert und auch hauptsächlich für einen Raspi Zero oder 1 interessant.

Homebrew und sonstige freie Spiele werden leider standardmäßig nicht in der „Playlist“ angezeigt, der Liste der spielbaren Games, falls sie nicht in einem von Lakka gepflegten Katalog der kompatiblen Spiele stehen. Man kann diese dennoch spielen und zu den Favoriten



hinzufügen, automatisch geht das aber nicht. Ansonsten ist es angenehm, dass die installierten ROMs automatisch gefiltert und sortiert werden, so dass man nur selten auf Probleme wie nicht lauffähige Spiele stößt. Außerdem ist dadurch auch die starre Verzeichnisstruktur nicht nötig, die bei anderen Distros für Übersicht sorgen soll.

Lakka setzt auf einige moderne Ansätze, etwa die automatische Erkennung und Konfiguration von Controllern und das datenbankbasierte Ausliefern von fehlerhaften ROMs. Aber gerade die automatische Konfi-

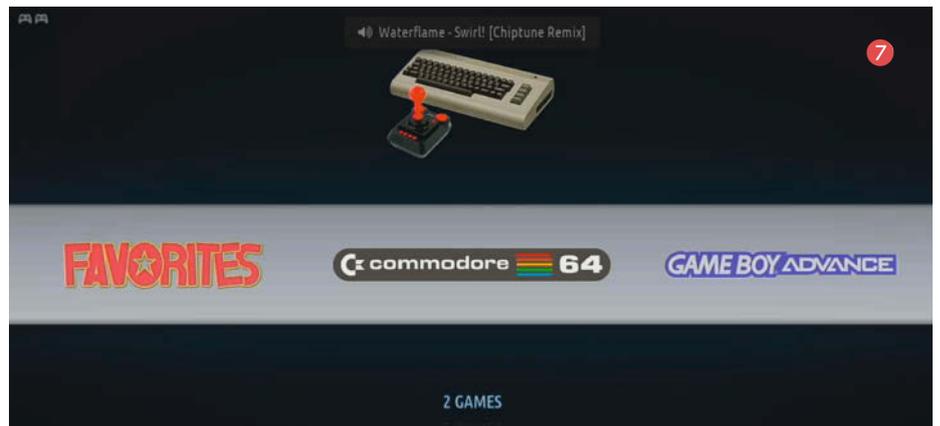
guration der Eingabe-Hardware führt bei exotischer Bestückung (wie bei meinem Arcade Bartop, siehe Seite 14) schnell zu der Situation, dass man doch eine Tastatur anschließen muss. Die händische Konfiguration ist dann komplizierter und durch einen Timeout stressiger als die idiotensichere Konfiguration bei den anderen Distributionen.

Lakka startet erst mal komplett ohne Spiele. Eine Netzwerkverbindung vorausgesetzt kann man mit dem Lakka Menü über *Online Updater/Content Downloader* einige legale ROMs und Demos laden.

Batocera

Batocera begrüßt den Anwender zuerst mal mit einer Text-UI und – für Raspi-unerfahrene Retro-Fans vielleicht beängstigend – einem Filesystem-Resize, was allerdings nur bedeutet, dass die gesamte Kapazität der SD-Karte verfügbar gemacht wird. Nach diesem ersten Booten bekommt man aber die Textkonsole nicht mehr zu sehen.

Dann dauert es etwas und man kommt nach schickem Splash-Screen in ein sehr bekanntes grafisches Menü – kein Wunder, denn es wird wie so oft EmulationStation verwendet. Aber anders als bei RetroPie und Recalbox ist hier alles schick und bunt (Geschmackssache, es gibt aber viele weitere alternative Themes zum Download) und man kann sofort ein paar Emulatoren mit vorinstallierten freien Games ausprobieren **7**. Ernüchterung bis Frust stellt sich ein, wenn dann das C64-Game genauso lange lädt wie im Original. Oder ist es doch hängengeblieben?! Wenn man nur einen einfachen Arcade-Controller angeschlossen hat, gibt es eventuell keinen Weg, das Game zu verlassen und man muss den Raspi hart booten und dann den vergessenen Hotkey definieren ...



Klasse ist, dass schon entsprechende **Bezels** für die vorinstallierten Games vorhanden sind – so bekommt man eine Idee, was möglich ist, um den sonst entstehenden schwarzen Trauerband bei 4:3-Games auf modernen 16:9 Displays zu nutzen **8**. Batocera macht allerdings auf einem Raspi 1 nur bedingt Spaß, wenn man nicht rigoros herunterkonfiguriert (gerade bei den Bezels und Shadern). Bei dieser Distro gilt eher das Motto „alles ist schon für maximales Aussehen konfiguriert“, aber

auf langsamerer Hardware wird es anstrengend, all diese Ressourcenfresser zu zähmen.

Auf meinem Bartop mit 4:3-Monitor und Raspi 3 hatte ich etwas Probleme mit der Bildschirmkonfiguration: Zwischen Hauptmenü und Games wurde immer die Auflösung gewechselt und ich habe bisher keine Möglichkeit gefunden, das zu beheben. Ich vermute, dass dies durch eine falsche Auflösung beim ersten Start hervorgerufen wurde. Für Systeme, die einen Libreto-Core benutzen, ist die



Konfiguration pro Spiel sehr gut integriert. Durch das Menü (Hotkey-B) sieht man noch das Spiel durchscheinen und kann so Änderungen, etwa von Skalierung oder Shadern, direkt beurteilen 9.

Media Center

Wenn man sowieso schon ein Media Center auf Basis von Kodi im Wohnzimmer hat (siehe auch Make 3/20, S.60), gibt es noch die Möglichkeit, mit dem Add-on *Kodi RetroPlayer* Emulatoren fürs Retro-Gaming zu nutzen. Dies ist seit Kodi 18.x möglich, soll hier aber nur der Vollständigkeit halber erwähnt werden.

Was für wen?

Ist man auf ein spezielles System fixiert, dann sollte man sich genau anschauen, ob dies von

der ins Auge gefassten Distribution (gut) unterstützt wird. Gerade bei einigen klassischen Heimcomputern (C64, Amiga, Sinclairs) ist dies nicht immer der Fall. Systeme mit Tastaturen sind daher oft besser auf einem Desktop-Rechner zu simulieren.

Mein Favorit bei den Distributionen ist im Moment Lakka, eine sehr schick aussehende, gut funktionierende, teils aber auch karge Oberfläche, die sich prima einer modernen Wohnlandschaft mit riesigem TV als Zentrum anpasst, aber dank Reduktion auch nicht fehl am Platz auf einer Arcade-Maschine oder einer Handheld-Konsole ist. Batocera ist bunter und „lauter“ und passt gefühlt mehr in eine Gamer-Bude, je nach Geschmack und Anfälligkeit animiert aber eventuell gerade diese Buntheit dazu, noch eine Runde mehr zu spielen.

In Sachen Funktionalität nehmen sich diese beiden Distributionen nichts. Batocera lässt



mehr eigene Konfiguration zu und lässt sich durch Themes sehr weit grafisch und in der Bedienung anpassen. Lakka sperrt sich da ein wenig. Mit Standard-Hardware (Raspi 3/4 an HDTV) ist die Entscheidung eher Geschmacksache. Auf meinem Bartop Arcade mit 4:3-Monitor hat aber Lakka die Nase vorn: Hier war ich viel schneller bei einer spielbaren Konfiguration, während Batocera nach wie vor noch viel Arbeit erwarten lässt. Auch auf dem Raspi Zero oder 1 ist Lakka durchweg schneller, beim Booten sowie in den Menüs. Ein Media Center bietet Lakka aber nicht.

Möchte man enger an Linux bleiben, so ist auch Recalbox gut geeignet. Das Linux versteckt sich hier zwar perfekt unter der Oberfläche, wenn man es aber bewusst heraus lässt, kann man von dort noch viel anpassen und experimentieren. RetroPie schließlich ist die richtige Wahl für Spezialfälle und wenn man gerne *alles* selbst im Griff hat. Durch seine Architektur läuft es ja praktisch als Aufsatz auf diversen Linux-artigen Systemen (und Windows). Dieser Tatsache und der langen Entwicklungszeit geschuldet ist auch die größte Menge an Dokumentation vorhanden – aber auch oft nötig, wenn man mit dem Text-Editor *.conf*-Dateien editiert. —pek

Distribution	RetroPi	Recalbox	Lakka	Batocera
Version	4.6	6.1.1	2.3.2	5.26
URL	https://retropie.org.uk/	https://www.recalbox.com/	http://www.lakka.tv/	https://batocera.org/
Basis	Raspian	Linux	LibreELEC	Linux
Unterstützte Raspberry Pis	Zero/1/2/3/4	Zero/1/2/3	Zero/1/2/3/4	Zero/1/2/3/4/GPi Case
Weitere unterst. Hardware	Linux (alle auf Debian-Basis)	Odroid XU4/C2, x86-PC (32/64 Bit)	Odroid XU4/C2, x86-PC (32/64 Bit)	Odroid C2/Xu4/N2/GO-Advance, Rockchip, PC, Mac, Intel Nuc
Frontend	EmulationStation	EmulationStation 2	Libretro Frontend	EmulationStation
Environment	RetroArch/Diverse	RetroArch/PiFBA	RetroArch	RetroArch
Systeme	> 50	> 80	ca. 50	> 70
Media Center	Kodi (17/18 je nach Linux)	Kodi 17.6	–	Kodi 18.6 Leia
Filesystem-Zugriff	USB/sFTP/Samba/	Samba/Webserver/ssh/scp	Samba/direkt/scp/USB/NAS	Samba/direkt/USB/ssh/scp/NAS
Netplay	über RetroArch	✓	✓	über RetroArch
Controller	Tastatur/USB/PS3/Xbox/GPIO/Wireless	Tastatur/USB/PS3/Xbox/GPIO/Wireless/Virtual	Tastatur/USB/PS3/Xbox/GPIO/Wireless	Tastatur/USB/PS3/Xbox/BT/GPIO
Shader	per Config-Editor oder in RetroArch GUI	Per RetroArch-Menü oder Shader Sets aus der Community	per Hauptmenü für jedes Spiel oder Sets pro Core oder Spiel	per RetroArc oder Shader Sets pro System oder Spiel
✓ ja/vorhanden – nein/nicht vorhanden				



1	VCC	OUT	8
2	IN	S/H	7
3	COMMON	NC 1	6
4	NC	VEE	5



1	1	2	2
3	3	4	4
5	5	6	6
7	7	8	8
9	9	10	10
11	11	12	12
13	13	14	14
15	15	16	16
17	17	18	18
19	19	20	20
21	21	22	22
23	23	24	24
25	25	26	26
27	27	28	28

GRATIS Download

- Schaltplansymbole
- PCB-Footprints
- 3D-Modelle



mouser.de/ecad



MOUSER
ELECTRONICS

Bartop Arcade mit Raspberry Pi

Spielhallen-Feeling mit einem handlichen Gerät, das immer verfügbar ist, weniger kostet als ein Original-Spielautomat und doch fast alle Klassiker der Computerspielgeschichte verfügbar macht – wir zeigen, wie das geht!

von Carsten Wartmann



Wenn man mit den klassischen „Telespielen“ aufwächst – vom Atari VCS2600 über VIC-20 bis zum Amiga – dann stammt man aus einer Generation, in der es noch Spielautomaten in Restaurants gab und Spielhallen, in die man auch als Minderjähriger kam. Was genau den Reiz von Arcade-Automaten für uns Kids damals ausmachte, kann ich gar nicht genau sagen – spätestens mit den Konsolen und Heimcomputern sollte sich der Spieltrieb doch eigentlich zu Hause befriedigen lassen. Vielleicht war es der Look der Maschinen oder der Wettbewerb, wer länger für die Mark spielen konnte?

Irgendwann ließ die Faszination nach und als die Geräte aus den Kneipen und Spielhallen flogen, wollte kaum jemand so eine Kiste kaufen und sich in die Wohnung stellen. Daher wurden fast alle originalen Spielmaschinen verschrottet und sind heutzutage nur noch schwer oder sehr teuer zu bekommen. Zudem sind sie groß und üblicherweise nur für ein einziges Spiel ausgelegt.

Dann kam der Raspberry Pi und es wurde nun einfach und günstig, alte Spiele zu emulieren (siehe dazu auch unsere Übersicht ab Seite 8). Da zudem die Retrowelle anschwellte, wurde in mir der Wunsch immer stärker, so ein Arcade-System zu besitzen – die Klassiker am TV oder Monitor mit einem modernen Game-Controller zu spielen, fühlte sich einfach nicht richtig an. Und heute, da ich so ein selbstgebautes Retro-Arcade-Geräte habe, stelle ich fest: Fast jeder Besucher möchte mal daran spielen, wenn er es sieht.

Kurzinfo

- » **Eigenes Design: Retro-Charme bei handlichen Abmessungen**
- » **Gehäuse aus Plattenmaterial und 3D-Druck-Teilen**
- » **Alten Monitor recyceln und stilecht ansteuern**

Checkliste



Zeitaufwand:

etwa 20 Stunden Design und Elektronik, 5 Stunden 3D-Druck, gut 20 Stunden Gehäusebau und Endmontage



Kosten:

50 bis 150 Euro, je nach Inhalt der Bastelkiste und Recyclingquote



Entwerfen:

Konstruktion des Gehäuses nach vorhandenen Hardware-Komponenten



3D-Druck:

für Verbindungsteile und Halterungen (optional)



Hardware:

Raspberry Pi einrichten, Buttons und Joystick anschließen

Alles zum Artikel
im Web unter
[make-magazin.de/xdt5](https://www.make-magazin.de/xdt5)

Material (Auswahl)

- » **Raspberry Pi** je nach zu emulierender Hardware Version 1-4, hier ein Pi 3A+
- » **SD-Karte** mit passender Retro-Distribution (siehe auch Seite 8)
- » **Arcade-Hardware** Anschluss über USB oder GPIO
- » **Netzteil(e)** für Raspi und Monitor
- » **Step-Down-Konverter** optional, 12V auf 5V für den Raspi
- » **Monitor oder TV** mit HDMI-Eingang, ideal mit Seitenverhältnis 4:3
- » **Verstärker und Lautsprecher** optional, etwa Chipverstärker mit PAM8403
- » **Solid State Relais (SSR)**, etwa PVT312PTF für die Monitorschaltung
- » **Elektronische Kleinteile** etwa Widerstände, Lochrasterplatte

Mehr zum Thema

- » Daniel Bachfeld, Emulator-PC im Spielhallengehäuse, c't Hacks 3/12, S. 100
- » Carsten Wartmann, 3D-Kurs für Maker: Konstruieren mit Blender 2.8, Video-Tutorial-Serie von Make

Design und Planung

Ziemlich klar war, dass ein Full-Size-Standgerät aus Platzgründen für mich nicht in Frage kam. Also sollte es ein sogenannter Bartop werden, den man natürlich nicht nur auf eine Bar, sondern auch auf einen Tisch stellen kann. Eine kurze Recherche ergab, dass nichts fertig zu kaufen war, was mir gefiel und bezahlbar gewesen wäre. So entschied ich mich leichtherzig für einen kompletten Eigenbau.

Auch wenn ich als Dipl. Ing. mit Skizzen und Maßzeichnungen umgehen kann, bin ich ein sehr visueller Mensch. Zudem war bei diesem Projekt ein gefälliges, wohnzimmer-taugliches Design erforderlich. Deshalb habe ich schon vom Start an meine Ideen in der kostenlosen 3D-Software Blender modelliert, dazu später mehr. So konnte ich mir gleich einen dreidimensionalen, realistischen Eindruck verschaffen, wie das Gerät später aussehen würde. Mein Design sollte einen Retro-Charme haben, aber weniger eckig als die klassischen Arcade-Maschinen ausfallen.

Monitor als Ausgangspunkt

Die Maße eines solchen Geräts werden hauptsächlich durch den verwendeten Monitor vorgegeben, den man von seinem Gehäuse befreit. Mein erster Entwurf entstand um einen alten 17-Zoll-LCD-Monitor mit Seitenverhältnis 4:3. Die allererste Idee, ganz klassisch einen Röhrenmonitor zu verwenden, wurde aufgrund des Gewichts schnell verworfen. Auch ergaben Tests an solchen Monitoren, dass deren Optik heutzutage doch nur noch sehr schwer zu ertragen ist und ich hatte mir schon in den 80ern mit Heimcomputern an Röhrenfernsehern meine Augen verdorben.

Das moderne Seitenverhältnis 16:9 ist eher ungeeignet für die Spieleklassiker, die alle für 4:3-Monitore entworfen wurden. Bei einem 16:9-Monitor entstehen dann unschöne Verzerrungen oder es werden schwarze Balken am Rand sichtbar. Viele Spielhallen-Arcades benutzen übrigens auch um 90 Grad gedrehte Monitore mit einem Seitenverhältnis von dann 3:4.



Materialfrage

Meine ersten Versuche, an den Maßen und damit beim Material für die Kiste drumherum zu sparen und trotzdem die Optik von klassischen Automaten mit *Marquee* (dem oft beleuchteten Kasten mit Artwork und Namen des Spiels) über dem Bildschirm zu kreieren, gelangen – gelinde gesagt – ziemlich dürftig **1**. Beim Suchen nach geeignetem Plattenmaterial für das Gerät fand ich dann Multiplex-Platten, die beidseitig tiefschwarz und glatt beschichtet und damit optisch den klassischen schwarzbraunen Siebdruckplatten haushoch überlegen sind, die ja zudem noch eine störende raue Seite haben. Die Quadratmeterpreise für dieses Multiplex ließen mich allerdings schnell an ein kleineres Gerät den-

ken. Deshalb ruhte das Projekt nun erst einmal.

In der Zwischenzeit probierte ich mich an einem Handheld-Emulator auf Basis eines Raspberry Pi Zero, den ich in Blender entworfen und auf meinem Prusa i3 MK2 ausgedruckt habe und der viel Aufmerksamkeit auf der Maker Faire in Berlin 2018 erzielte, unter anderem beim Prusa Printers Chef Jo Prusa persönlich. Bei Interesse beschreibe ich diesen Handheld-Emulator gerne ausführlich in einem eigenen Artikel, lassen Sie es die Redaktion per Mail wissen (mail@make-magazin.de).

Durch einen glücklichen Zufall kam ich kurz nach der Messe an einen 15-Zoll-LCD-Fernseher, der sehr modular aufgebaut war und neben einem HDMI-Eingang auch ein Tonteil hatte. Zudem war er viel leichter und natürlich

kleiner als mein erster Monitor. Leider waren die Preise der von mir favorisierten Platten inklusive Zuschnitt inzwischen utopisch hoch geworden und eine überzeugende Alternative ließ sich nicht finden. Es wurden gedanklich viele Ideen durchgespielt (aufgeben, doch lackieren, lackieren lassen, Lottogewinn ...), keine davon ließ sich aber mit meinen Vorstellungen und finanziellen Mitteln vereinen. In nahezu letzter Sekunde bevor ich aufgeben oder das Projekt wieder vertagen wollte, habe ich dann ein Werbevideo über Folie zur Beschichtung von Flächen (*d-c-fix*) gesehen und schnell war so eine Rolle im Baumarkt gekauft. Beim Ausprobieren stellte sich heraus, dass sich die Folie sehr gut verarbeiten ließ und so konnte nun endlich die Umsetzung starten: Mit der Folie ließen sich präsentable Seiten erreichen und an den Kanten meine geliebte Multiplex Optik erhalten. Das gab dem Projekt neuen Schwung.

2 /boot/config.txt

```
01 # uncomment if you get no picture on HDMI for a default "safe" mode
02 #hdmi_safe=1
03
04 disable_overscan=0      # 1 wenn Zeilen oder Spalten abgeschnitten
05 overscan_scale=1
06
07 # uncomment to adjust overscan. positive numbers if console
08 # goes off screen, and negative if there is too much border
09 overscan_left=4
10 overscan_right=-8
11 overscan_top=-8
12 overscan_bottom=4
13
14 # uncomment to force a specific HDMI mode (this will force VGA)
15 # Mode für PAL 720x576
16 hdmi_group=1
17 hdmi_mode=17
18
19 # Sound output. Set to 0 for autodetect, 1 for DVI, 2 to force HDMI.
20 hdmi_drive=0          # hier über Klinke
21
22 # Using /etc/modules is deprecated
23 # So manually enable audio
24 dtparam=audio=on
25 audio_pwm_mode=2      # Verbessert Ausgabe
26 config_hdmi_boost=0
27
28 # force hdmi while the tv can take time before sending the signal
29 hdmi_force_hotplug=1
30 #hdmi_ignore_hotplug=1
31
32 # uncomment for composite PAL
33 #sdtv_mode=2
34
35 # if you plug your tv at the same time as your rpi and that the rpi
    switches from the hdmi or give a low resolution because tv had no
    enough time to initialize it
36 boot_delay=3
37
38 # uncomment if you don't want the rainbow at startup
39 disable_splash=1
40
41 # default CEC name
42 cec_osd_name=recalbox
43
44 # Overclock
45 gpu_mem_256=128
46 gpu_mem_512=256
47 gpu_mem_1024=384
48
49 avoid_safe_mode=1
50
51 kernel=zImage
```

Die Elektronik

Noch während der Designphase (dazu im Anschluss mehr) wurden die fehlenden Elektronikkomponenten gekauft und gebaut und immer wieder zusammen mit dem Monitor getestet. Die Installation einer Emulator-Distribution auf dem Raspberry Pi (hier ein Raspi 3 B+) ist heutzutage recht einfach. Die größten Probleme verursachten die Einstellungen für den Monitor, der leider kein Full HD bietet, aber dennoch möglichst gut ausgenutzt werden sollte. Hier waren einige Versuche nötig (die jeweils zum erneuten Booten zwangen), um eine optimale Kombination aus Overscan und Auflösung einzustellen. Am Ende hatte ich eine Kombination von Werten gefunden, die gut für den Bildschirm mit Seitenverhältnis 4:3 in PAL-Auflösung (720 × 576) funktioniert.

Um überhaupt zu erfahren, was für ein Monitor an dem Raspi hängt, kann man bei HDMI (und DVI, eingeschränkt auch VGA) den Befehl `tvservice -s` benutzen. Es folgt eine Ausgabe etwa der Form

```
state 0x120005 [DVI DMT (35) RGB full
5:4], 1280x1024 @ 60.00Hz,
progressive
```

was bei unbekanntenen Monitoren erste Hinweise gibt und bei der Fehlersuche hilft.

Im Kasten **2** sieht man meine Boot-Konfiguration. Ich habe den Grundaufbau intakt gelassen, damit Sie sich zurechtfinden. Die Kommentare in Deutsch sind meine Änderungen und Anmerkungen.

Leider stellte sich erst jetzt heraus, dass der Tonteil des TV defekt war. Nach Fehlersuche und Test mit einem neuen Audiochip ging es immer noch nicht und schlussendlich wurde ein externer Verstärker (2x3W PAM 8403) zusammen mit den originalen Lautsprechern

des Fernsehers benutzt. Das zog nach sich, dass ich den Ton von der Kombibuchse des Raspi holen muss, was eine spezielles Kabel erfordert, da hier sowohl Stereo- als auch Video-signale anliegen und die Buchse leider nicht 100 Prozent standardkonform ist (Video und GND sind vertauscht!). Entweder man besorgt sich das passende spezielle Kabel dafür oder man benutzt ein gewöhnliches Stereo-Klinkenkabel, was zwar dazu führt, dass Video-Out zu GND kurzgeschlossen wird, meinen Pi scheint das aber nicht zu stören.

Stromversorgung

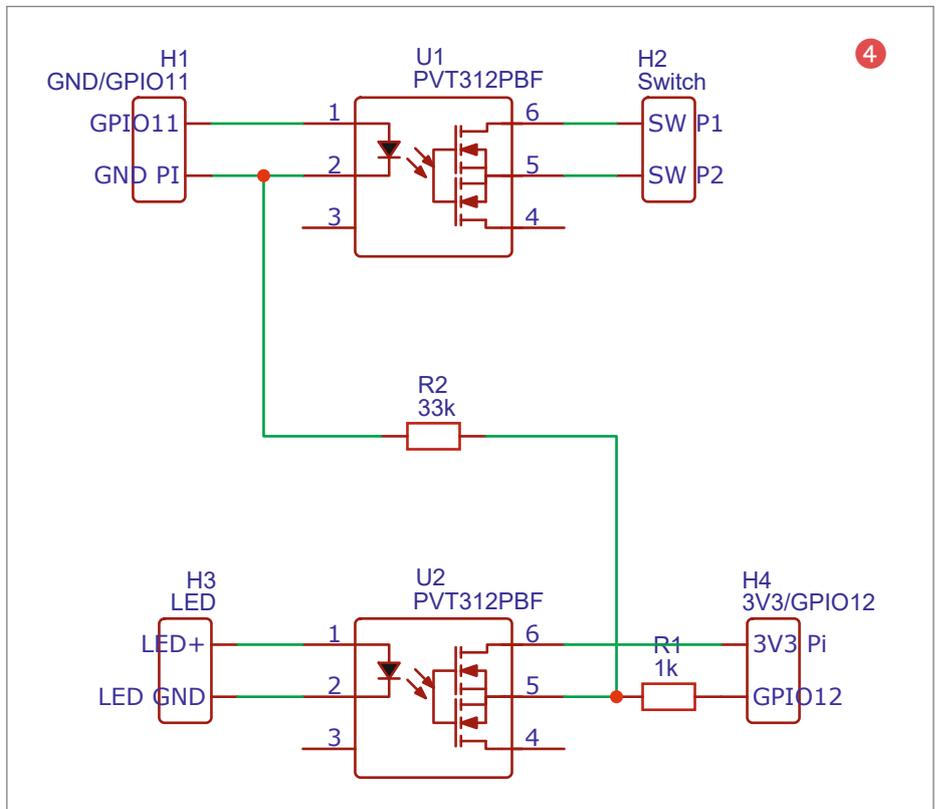
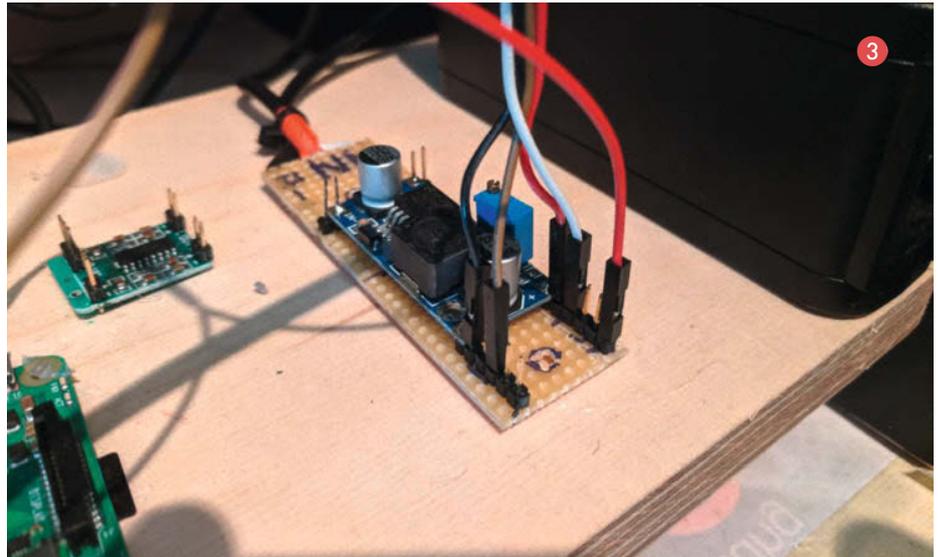
Die Stromversorgung sollte eigentlich das originale Netzteil des TVs sicherstellen, das genügend Leistung bei 12V bereitstellt, um noch einen Raspi zu versorgen, wenn man die 12V in die nötigen 5V für den Einplatinenrechner konvertiert. Hierfür kam ein Step-Down Wandler mit LM2596-Chip zum Einsatz. Die Spannungsversorgung sollte direkt über die GPIO-Ports erfolgen, weil ich kein USB-Kabel zerschneiden wollte. Das hat bei älteren Raspis immer gut funktioniert.

Der verwendete Raspi 3 ist aber leider sehr stromhungrig und so bekam ich öfter das Icon mit dem Blitz zu sehen, das vor Problemen mit der Spannungsversorgung warnt, als mir lieb war. Ich erhöhte die Spannung vom Wandler ein wenig auf 5,2V, aber auch probeweise parallel geschaltete Elkos waren nicht in der Lage, die Spannung konstant zu halten. Ich konnte die Problemquelle schließlich auf die verwendeten Patchkabel vom Wandler zum Raspi einengen, die einen zu hohen Übergangswiderstand aufwiesen. Die schlichte Verdoppelung der Leitungen löste das Problem erst einmal **3** – zumindest sah ich später, während des Betriebs meines Bartops über zwei Tage Maker Faire hinweg, keine Spannungswarnungen mehr. Mittlerweile haben die Kontakte aber durch die Lagerung gelitten. Inzwischen würde ich stattdessen entweder direkt an die Buchsenleiste gute Kabel anlöten oder doch den USB-Port für die Spannungsversorgung nutzen, was offiziell auch so empfohlen ist, aber in der Praxis etwas frickelig werden kann.

Der Chipverstärker für den Sound wurde zuerst auch mit 5V vom Wandler versorgt, was aber ein furchtbares Brummen zur Folge hatte, da das Audiosignal auf einer anderen GND-Ebene lag. 5V waren am Raspi nicht mehr verfügbar, aber den Verstärker mit dem GND vom Raspi zu verbinden brachte die Lösung – kein Brummen mehr.

Monitor steuern

Der Monitor hat neben der Fernbedienung aus seiner TV-Vergangenheit auch ein Panel mit Knöpfen, das die wichtigsten Bedienfunk-



tionen zur Verfügung stellt. Nachdem man den Monitor mit Strom versorgt hat, muss man ihn per Tastendruck einschalten und – je nach Monitor – eventuell auch noch den richtigen Video-Eingang wählen. Dies ist natürlich für so ein Arcade-Gerät ziemlich uncool. Ich verpflichtete daher meinen Raspi dazu, diese Aufgabe beim Booten und Herunterfahren zu erledigen. Dazu überwacht der Raspi per GPIO und Start(Python)-Skript die Status-LED am Monitor und schaltet per GPIO elektronische Schalter (SSRs, Solid State Relays), die parallel zu den Tastern der Bedienpanels liegen **4**. Beim Herunterfahren schaltet ein weiteres Skript den Monitor aus.

Weil die SSRs parallel zu den noch vorhandenen Tasten liegen, kann man den Monitor immer noch über das Panel bedienen. Weil meine Konstruktion aber zuverlässig arbeitet, braucht man das nicht oft und daher habe ich das Panel einfach innen im Bartop-Gehäuse befestigt. Die Überwachung der Status-LEDs per Optokoppler oder SSR stellt sicher, dass ein bereits aktiver Monitor nicht wieder ausgeschaltet wird und umgekehrt. Diese SSRs kosten nur wenig mehr als Optokoppler und beeinflussen im Gegensatz zu diesen die fremde Schaltung nicht durch eine Widerstandserhöhung oder Kapazitäten, die am Ausgangstransistor des Optokopplers auftreten können.

5 TVPower.py

```

01 #!/usr/bin/python2
02
03 import sys
04 import time
05 import RPi.GPIO as GPIO
06
07 if len (sys.argv) != 2 :
08     print "Usage: python TVPower.py 0|1"
09     sys.exit (1)
10
11 # RPi.GPIO Layout verwenden (wie Pin-Nummern)
12 GPIO.setwarnings(False)
13 GPIO.setmode(GPIO.BOARD)
14
15 inPin = 12
16 outPin = 11
17
18 # Set Pins
19 GPIO.setup(outPin, GPIO.OUT)
20 GPIO.setup(inPin,GPIO.IN)
21
22
23 if sys.argv[1] == "1":
24     if not(GPIO.input(inPin)):
25         print("TV is Off, switching to ON!")
26         # Toggle Power Switch
27         GPIO.output(outPin, GPIO.HIGH)
28
29         # Warte 500 ms
30         time.sleep(0.6)
31
32         # Reset Pin
33         GPIO.output(outPin, GPIO.LOW)
34     else:
35         print("TV already ON!")
36
37 elif sys.argv[1] == "0":
38     if (GPIO.input(inPin)):
39         print("TV is ON, switching to OFF!")
40         # Toggle Power Switch
41         GPIO.output(outPin, GPIO.HIGH)
42
43         # Warte 500 ms
44         time.sleep(0.6)
45
46         # Reset Pin
47         GPIO.output(outPin, GPIO.LOW)
48     else:
49         print("TV already OFF!")
50 else:
51     print "Usage: python TVPower.py 0|1"
52     sys.exit (1)

```

6 Start-Skript

```

01 #!/bin/bash
02
03 #/usr/bin/python2 /recalbox/share/system/TVPower.py
04
05 case "$1" in
06     start)
07         /usr/bin/python2 /recalbox/share/system/TVPower.py 1
08         ;;
09     stop)
10         /usr/bin/python2 /recalbox/share/system/TVPower.py 0
11         ;;
12     restart|reload)
13         ;;
14     *)
15     esac
16
17 exit $?

```

Galvanisch getrennt sind sie natürlich auch und man kann damit sogar Wechselspannung schalten.

Das Python-Skript **5** erstellt man in /root oder einem User-Verzeichnis lässt es durch ein Start-Skript **6** in /etc/init.d/S32TVOn ausführen (bei Raspian-basierten Distributionen, bei anderen ist das gegebenenfalls anders!). Die Zahl bestimmt, wann das Skript in Relation zu den weiteren Startskripten bei Start und Shutdown ausgeführt wird.

Die eigentliche Arcade-Hardware wie Joystick und Buttons habe ich per USB angeschlossen. Der Raspi 3 hat ja zum Glück genügend USB-Ports, so dass noch Ports für WLAN-Dongle, weitere Controller oder eine Tastatur und Maus übrig bleiben. Die Verkabelung der Buttons erfolgt mit den mitgelieferten Kabeln und ist sehr einfach und verpolungssicher. Anschließend wird der ganze Aufbau als „Controller“ in *EmulationStation* konfiguriert – und damit ist das Innere des Bartop Arcade fertig.

Das Gehäuse

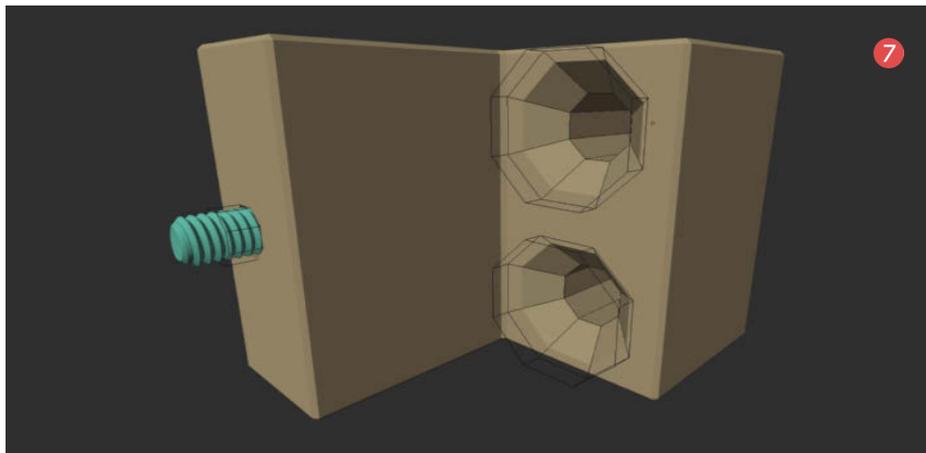
Durch die Entdeckung der *d-c-fix*-Folie gab das Budget als Plattenmaterial unbeschichtetes 15mm Birke-Multiplex her. Gegenüber billigem Sperrholz hat dieses Material weniger Äste und sonstige Macken, sondern schon von Fabrik aus eine sehr glatte Oberfläche und es lässt sich gut bearbeiten. Aber auch hier gibt es verschiedene Qualitäten mit römischen Ziffern oder früher den Buchstaben A-D gekennzeichnet. Bei Online-Bestellungen kann man trotzdem leider meist nur das Beste hoffen, im Baumarkt hingegen kann man sich die beste Platte aussuchen.

15mm Stärke sollten es schon sein, damit man die Teile gut per Schrauben miteinander verbinden kann. Außerdem kommen so die Sichtkanten besonders gut heraus. Nicht zuletzt kann man sich beim Spielen auf den nicht zu dünnen Kanten auch bequem abstützen.

Blender-Entwurf

Den 3D-Entwurf für die Plattenzuschnitte einerseits und die nötigen Verbinder aus dem 3D-Drucker habe ich mit Blender gemacht. Das ist kein CAD-Programm, man kann damit aber sehr wohl mit echten Maßen und teils auch ähnlich wie mit einer parametrischen Konstruktionssoftware arbeiten. Entscheidende Vorteile hat man mit Blender, wenn man schnell Designs probieren und visuell inklusive Material beurteilen will. Dies kam mir bei diesem Projekt besonders entgegen: Sicher müssen die Maße am Ende stimmen, aber um das endgültige Design festzulegen, braucht

ALLNET MAKER & MINT WELT



man eine schicke 3D-Ansicht mit realistisch dargestellten Materialien.

Der vorhandene Monitor bildete die Grundlage für die Konstruktion. Ich nahm ein möglichst orthogonales Foto auf und modellierte anhand dessen in Blender ein einfaches Platzhalterobjekt. An den Stellen, an denen es entscheidend war, nahm ich die Maße direkt am Original ab. So hat der Monitor an beiden Seiten je zwei M3-Gewinde im Rand, für die Halter entworfen wurden, mit denen der Monitor an den Seitenwänden des Korpus befestigt wird **7**. Wo Holzschrauben benutzt werden, wird eine Senkung mitmodelliert, um die Kräfte der Schraube besser in das Material einzuleiten und ein Aufspalten zu verhindern. Trotzdem sollte beim Schrauben in Kunststoff immer mit Gefühl oder der Drehmomentbegrenzung des Akkuschaubers gearbeitet werden.

Bei der Konstruktion wurden viele Modifier benutzt, um eventuelle Änderungen zu erleichtern und die Teile immer wieder verwenden zu können, etwa Boolean-Objekte für die Bohrungen für die Holzschrauben **8**. So entstanden dann auch Verbindungsteile mit verschiedenen Winkeln, Platinenhalter und Lautsprechergritter, die – weil direkt am Gerät konstruiert – sofort passen und schnell eventuelle Denkfehler aufzeigen. Meine 3D-Modelle gibt es übrigens über den Link in der Kurzinformatio zum Download; wer lernen will, Blender für eigene Projekte einzusetzen, findet dort

auch einen Link zur Video-Tutorial-Serie von Make zum Thema.

Virtueller Probestbau

Mit weiteren vorhandenen Bauteilen wie dem Mainboard des Monitors, dem Inverter, dem Netzteil für Monitor und Raspi verfuhr ich ebenso und generierte mit Hilfe von Fotos und dem Meßschieber einfache Platzhalterobjekte für die 3D-Konstruktion **9**. So konnte ich die Platzierung testen und mir gegebenenfalls alternative Befestigungsmöglichkeiten ausdenken, wenn Probleme schon im Vorfeld zu erkennen waren.

Die Seitenteile wurden mittels Bézierkurven in Blender designt, dann in Mesh-Objekte umgewandelt und mit virtuellen Materialien und Texturen versehen, um die beschichteten Multiplex-Platten darzustellen. Mit der so erreichten Darstellung (die auch das Titelbild dieses Artikels ziert) war schnell ein Design gefunden, das in meinen Augen eine gute Mischung aus Möbel und Spielautomat darstellt.

An dem so in Blender entstandenen virtuellen Prototyp wurden dann auch die nötigen Verbindungen festgelegt und wo man welche



• **Snapmaker**
3-in-1 3D Drucker, Fräser & Laser

• **Brick'R'knowledge**
Experimentiersystem Elektronik & Physik

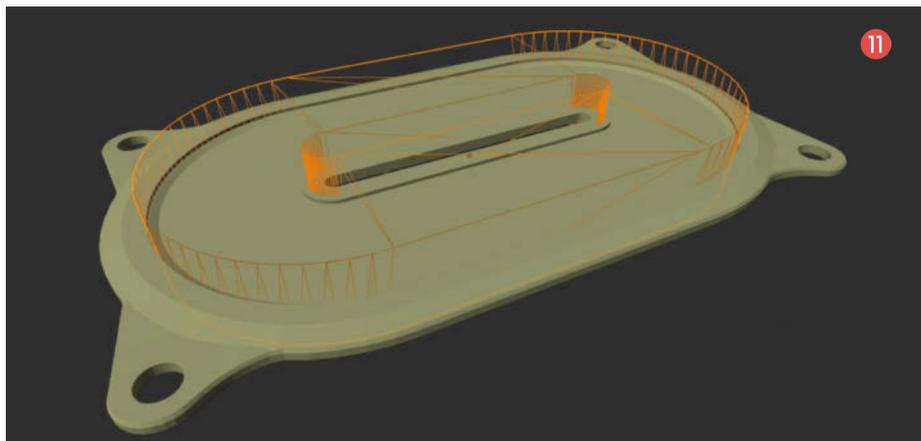
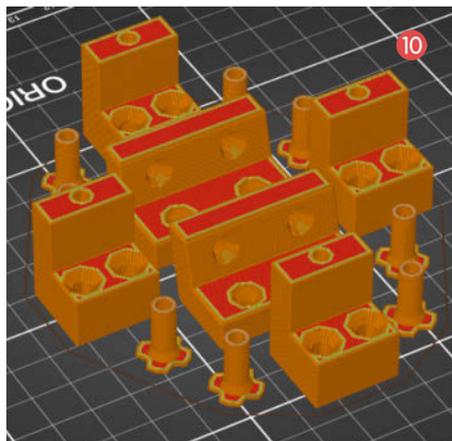
• **iRobot**
Coding Roboter

• **Robobloq**
MINT Roboter

• **MatataLab**
Coding ab 4 Jahren

• **3Doodler**
MINT 3D Stift

• **Arduino**



Schrauben benutzt – im 3D-Raum merkt man dann schnell, ob sich das ganze auch praktisch zusammenbauen lässt. Bei der Befestigung der Seitenwände an den Boden- und Zwischenplatten habe ich mich bewusst für eine von außen sichtbare Schraubverbindung entschieden: Die Edelstahl-Senkkopfschrauben setzen visuell einen schönen technischen Akzent an den ansonsten kahlen Seitenwänden. Sie wären aber auch leicht durch Dekor-Aufkleber (wie im Rendering) oder ein Stück der schwarzen Folie abzudecken. Wer sich sicher ist, das alles stimmt, kann die Schraubenköpfe nach der Montage auch verspachteln und dann die Folie aufbringen.

Für die Knöpfe, den Joystick und die USB-Elektronik wurden wie oben beschrieben Stellvertreterobjekte modelliert, um die Platzverhältnisse im Gerät zu bestimmen und die Montage zu testen. Die Anordnung der Bedienelemente habe ich aber vorab nur grob festgelegt, das tatsächliche Layout wurde fast zuletzt direkt am Bartop und mit realen Bedienelementen ausprobiert und optimiert. Dazu habe ich eine Styrodur-Platte (XPS) verwendet, Löcher an den geplanten Stellen hinein gebohrt und dann die Knöpfe und den

Joystick direkt am Gerät getestet. Erst nachdem ich mit der Platzierung absolut zufrieden war, habe ich die Lage der Bohrungen auf die Multiplex-Platte übertragen.

3D-Druck

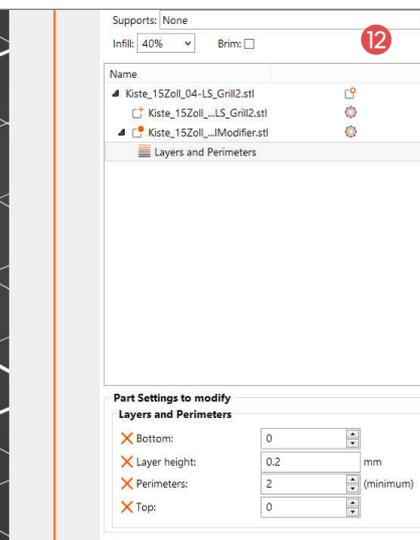
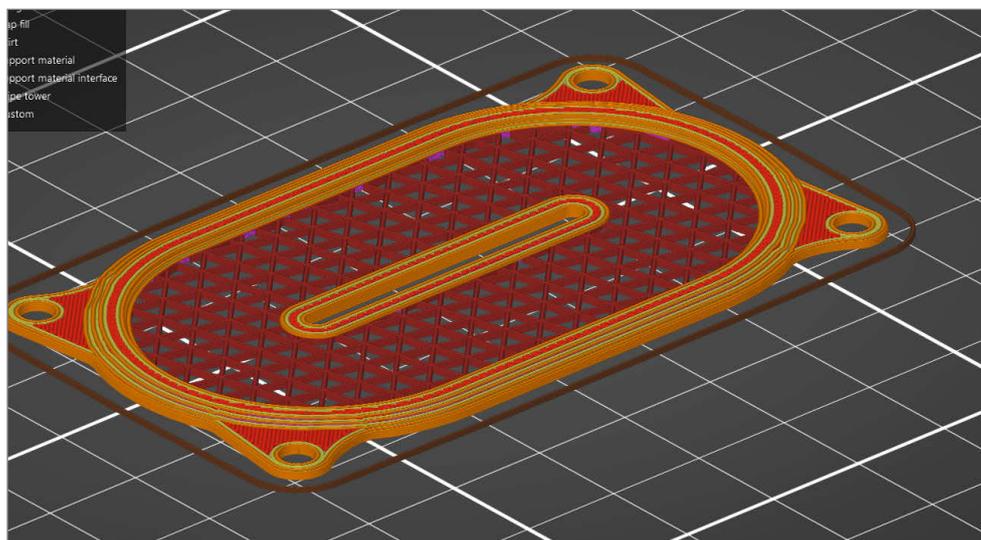
Alle zu druckenden Teile wurde im PrusaSlicer vorbereitet (siehe auch Seite 92) und dann in PETG gedruckt. PETG ist etwas elastischer und zäher als PLA, das gerne mal plötzlich bricht, wenn man eine Schraube zu stark anzieht oder eine Verbindung überlastet. Als Schichtdicke benutze ich 0,2mm für konstruktive Teile, 3 Perimeter, 20% Infill vom Typ *Honeycomb* 10.

Für die Lautsprecherabdeckungen aus dem 3D-Drucker habe ich im PrusaSlicer eine spezielle Funktion verwendet, mit der man durch Modifier-Objekte Aspekte des Slicings steuern kann. Im Bild 11 sieht man das Modifier-Objekt in Drahtgitterdarstellung und das massive Objekt, aus dem durch einen Trick das Lautsprechergitter entsteht, in Flächendarstellung. Man öffnet dieses Teil im Slicer und lädt dann bei selektiertem Bauteil über das Kontextmenü einen Modifier hinzu (Rechtsklick, *Add Modifier/Load...*).

In der Objektliste im PrusaSlicer rechts erscheint der Modifier jetzt in der Hierarchie des Basisobjekts 12. Neben dem Eintrag des Modifiers befindet sich ein kleines Zahnrad-Icon (hier ist es egal, ob Sie das obere oder untere benutzen), mit der Maus darauf klicken und im Menü *Layers and Perimeters* wählen. Unter der Liste erscheint ein Fenster (*Part Settings to modify*) in dem Sie *Top* und *Bottom Layer* auf Null setzen. Wenn Sie nun slicen, erzeugt die Software ein Objekt, bei dem die obersten und untersten Layer des Drucks innerhalb des Modifiers weggelassen werden. So entsteht im Druck ein Gitter aus dem Infill-Muster des Slicers, ohne dass man es selbst im Detail modellieren muss -- je nach Komplexität spart das viel Arbeit..

An die Säge!

An dieser Stelle verließ ich Blender und damit meine Komfortzone – die reale Holzbearbeitung stand an, mit Stich- und Kreissäge sowie Oberfräse. Diese Abenteuer beschreibt Teil 2 des Artikels zur Bartop Arcade im nächsten Heft ausführlich, aber es sei jetzt schon verraten – es geht gut aus! —pek



Früher war alles besser!



Willkommen in der Welt der Classic Games, wo Computer- und Videospiele viel Kreativität und Spielspaß versprochen – und bis heute halten. Wir stellen Spiele, deren Entwickler und Plattformen vor. Bei Retro Gamer finden Sie Screenshots, Fakten, Tipps und mehr zu den Hits von damals.

Testen Sie 2 × Retro Gamer mit 30 % Rabatt!
Lesen Sie 2 Ausgaben für nur 18,- Euro* statt 25,80 Euro* im Handel.

Jetzt bestellen und vom Test-Angebot profitieren:
www.emedia.de/rg-mini

© Copyright by Maker Media GmbH.

Telefon: (0541) 800 09 126
werktags von 8 – 20 Uhr,
samstags von 10 – 16 Uhr
E-Mail: leserservice@emedia.de
Post: eMedia Leserservice
Postfach 24 69
49014 Osnabrück

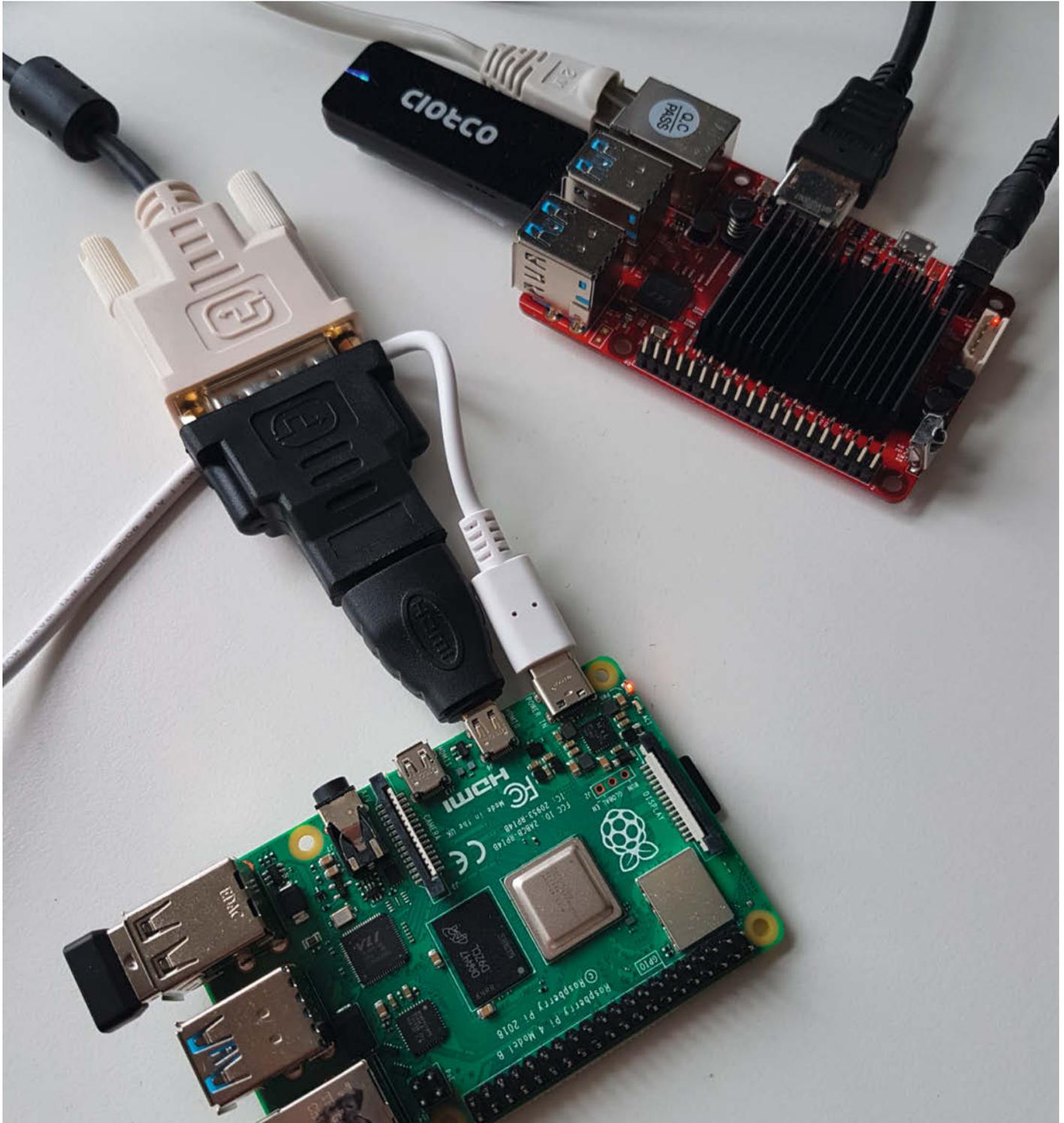
*Preis in Deutschland.

Odroid C4 gegen Pi 4

Unter den Single-Board-Computern ist der Raspberry Pi 4 seit Jahren der Platzhirsch. Mit dem Odroid C4 versucht der Hersteller Hardkernel eine weitere Alternative zu platzieren. Wir haben uns angeschaut, was der C4 leistet und ob er besser als der Pi ist.

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

von Daniel Bachfeld



Hardkernel war einer der ersten Hersteller, die mit eigenen Single-Board-Computern im Scheckkartenformat seit 2013 gegen den Pi antraten. Von den Modellen U2 und U3 über C0 bis C2 zu XU4 bis N2: Immer lagen die Boards hinsichtlich der Speicherausstattung, Prozessorperformance oder Hardware-Unterstützung von Videocodern vor dem Pi. Was die Software-Unterstützung anging, setzte Hardkernel wo es ging auf das populäre Ubuntu in 64-bittigen Versionen. Leider verhinderte die Wahl der populären Linux-Distribution nicht, dass es trotzdem an der reibungslosen Unterstützung mancher verbauter Hardwarekomponenten mangelte. Dennoch war Odroid (auch für den Autoren) in vielen Projekten oft die bessere Wahl.

Mit dem Pi 4B mit 4GByte RAM (für knapp 60 Euro) hatte die Foundation Mitte 2019 aber erstmals leistungsmäßig aufgeholt. Mit der 8GByte-Version (ca. 90 Euro) scheint sie nun die Konkurrenz hinter sich lassen zu wollen. Und nun auch noch die von vielen Makern lang ersehnte 64-Bit-Version eines Raspberry Pi OS. Das sollte erstmal eine Weile reichen! Nun hat aber Hardkernel mit dem C4 (ab 70 Euro) einen potenten Gegner ins Rennen gegen den Pi 4 geschickt.

Die verwendeten Cortex-A55-Kerne des C4 sind nominell nicht so leistungstark wie die A72-Kerne des Pi, etwa weil der A72 Funktionen wie Out-of-Order-Execution mitbringt, in der eventuell anstehende Programmabläufe einfach schon mal vorberechnet werden. Allerdings pumpen die Kerne des C4 die Daten mit 1,9GHz durch ihre Register, wohingegen der Pi nur mit 1,5GHz getaktet ist. Der 4GByte große DDR4-RAM des C4 arbeitet mit 1320MHz und ist nur mit 32 Bit angebunden, statt der bei DDR4-RAM möglichen 64 Bit. Der Pi taktet seine bis zu 8GByte LP-DDR4-RAM mit 3200MHz. Das klingt zwar nach einem mehr als doppelt so hohem Takt, allerdings arbeitet LP-RAM (LP steht für Low Power) mit nur zwei je 16 Bit breiten Channels für den Datentransport, was die Arbeit verlangsamt.

Benchmarks

Ein Vergleich der Spezifikationen lässt nur schwer Schlüsse zu, ob und welches System

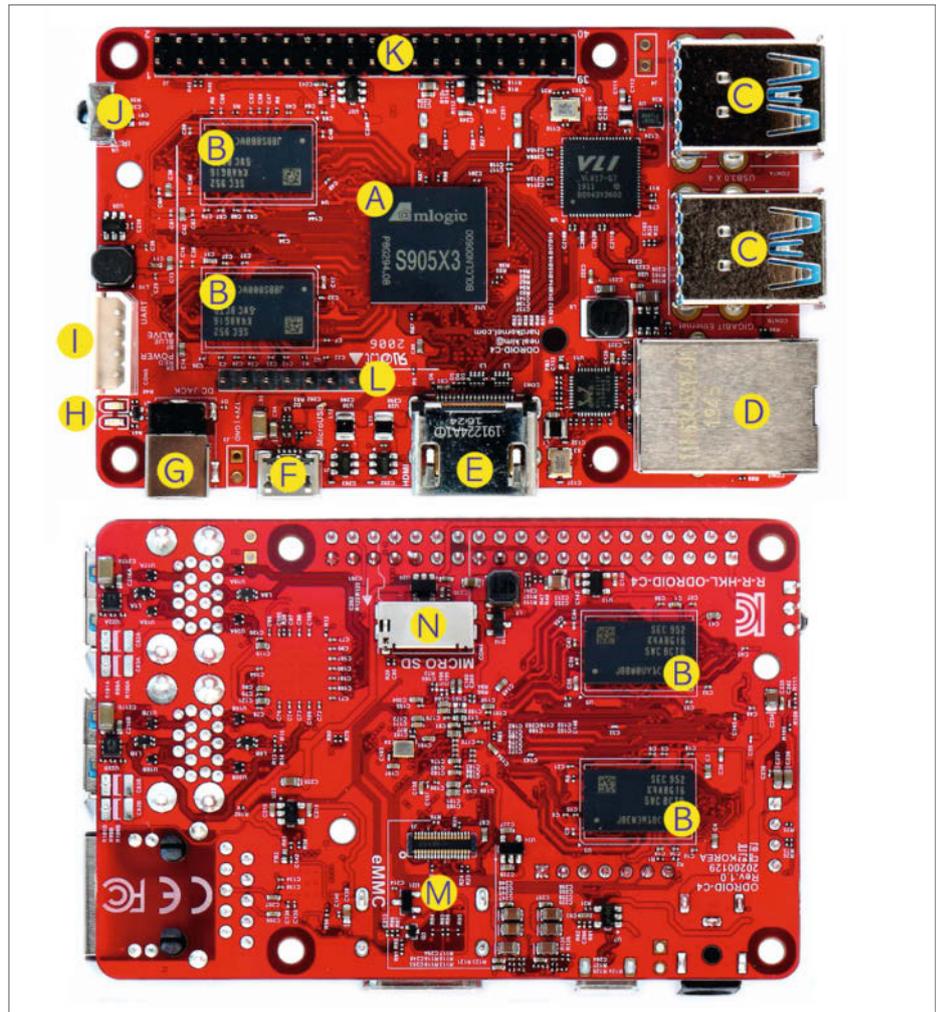
wirklich leistungsfähiger ist. Wir haben einige Benchmarks durchgeführt (siehe Tabelle), in denen wir den C4 mit einem 64-bittigen Ubuntu 20.04 und den Pi 4 mit Raspberry Pi OS in 32 und der neue verfügbaren Version mit 64 Bit verglichen haben. Beim Pi wollten wir wissen, ob die Behauptungen stimmen, er sei mit der 64 Bit in vielen Fällen langsamer als mit der 32-Bit-Version.

Beim C4 haben wir zudem mit MicroSD-Karte und eMMC als Massenspeicher gemessen, um zu sehen, welchen Einfluss die unterstützten Speicherarten auf die Leistung haben. Für

die Benchmarks haben wir den Packer/Entpacker 7zip, OpenSSL, OpenCV und Tensorflow eingesetzt, um die Leistungsfähigkeit des SoC zu ermitteln. Mit gnome-disks, mbw und iotzone haben wir den Durchsatz auf den USB3.0-Ports, zum RAM und zu den Speicherkarten gemessen.

Einpacken

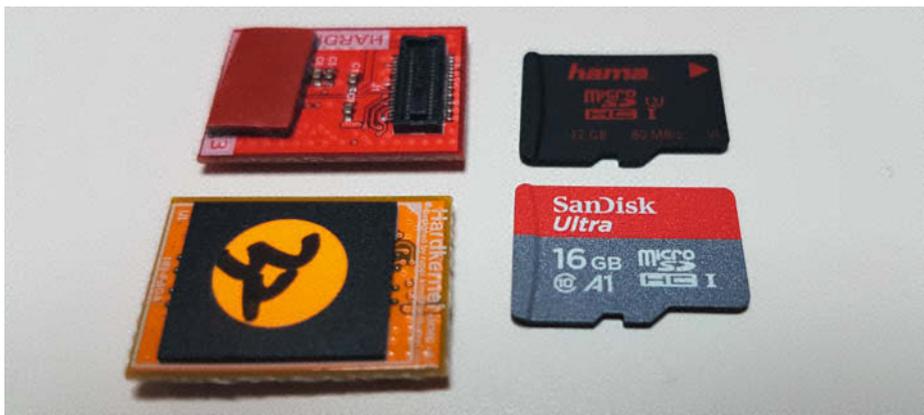
Bei 7zip lag der Pi 4 vor dem C4, wobei der Pi beim Komprimieren mit 64-Bit-OS leicht schneller als die 32-Bit-Version war. Dafür ent-



Der Odroid C4 und seine Komponenten

A	CPU (Amlogic S905X3)	H	2 x System LED indicators
B	DDR4 memory (2GiB or 4GiB)	I	1 x UART for system console
C	4 x USB 3.0 host ports	J	1 x IR receiver
D	1 x RJ45 Ethernet port (10/100/1000)	K	40 x GPIO pins
E	1 x HDMI 2.0	L	7 x GPIO pins
F	1 x Micro USB 2.0 port (OTG)	M	1 x eMMC module socket
G	1 x DC power jack (Outer diameter : 5.5mm, inner diameter : 2.1mm)	N	1 x Micro SD slot

Die Erklärung der Komponenten



Links ein eMMC von oben und unten, rechts herkömmliche MicroSD-Karten

packte der Pi bei 32 Bit erheblich schneller. Beim Verschlüsseln mit dem Blockchiffre AES lag der Pi mit 32-Bit-OS vorne, was umso erstaunlicher ist, da der Pi 4 trotz ARMv8 keine in Hardware gegessene Crpyto-Unterstützung hat. Die Foundation hat diese aus Kostengründen nicht lizenziert. Beim vom Hersteller Hardkernel veröffentlichten Vergleich der Crypto-Funktion überflügelt hingegen der C4 den Pi 4 um Größenordnungen (911 MBit/s zu 126 MBit/s, in unserem Test 830 MBits/s zu 163MBit/s), allerdings dreht es sich dort um den Stromchiffre (Hash-Algorithmus) SHA256. Damit wäre der C4 (wenn es denn Sinn ergeben würde, dies auf SBCs zu machen) eher für Bitcoin-Mining geeignet und der Pi 4 eher für den Einsatz als VPN-Server.

Praxisrelevanz

Da auch synthetische Benchmarks wie Sysbench (den wir diesmal nicht benutzt haben) eher eine grobe Schätzung sind, haben wir zwei aus unserer Sicht praxisrelevantere Szenarien zur Bildverarbeitung mit OpenCV sowie zur Klassifizierung von Objekten mittels des KI-Frameworks Tensorflow Lite als Benchmark eingeführt. Unser OpenCV-Benchmark misst

die Zeit, die Python3 zur Verarbeitung eines Bildes benötigt. Der TF-Lite Benchmark misst die Zeit, um in einem Bild mehrere Objekte zu klassifizieren und das dazugehörige Label anzuzeigen.

In beiden Disziplinen schob sich der Pi mit 64-Bit-OS allen Unkenrufen zum Trotz klar nach vorne. Das liegt unter anderem daran, dass es im 64-Bit-Betrieb neue Befehle gibt, die bestimmte mathematische Operationen beschleunigen, sofern die jeweilige Software mit diesen Features übersetzt wurde. Da OpenCV und Tensorflow davon reichlich Gebrauch machen, bringt das gegenüber 32-Bit Geschwindigkeitsvorteile – und nicht zuletzt auch Genauigkeitsvorteile, weil die FPU bei 64 Bit mit doppelter Genauigkeit rechnet.

Speicher

Mit seiner Unterstützung für eMMC-Speicher (Embedded Multi Media Card) konnte der C4 seine Software im Test mehr als dreimal so schnell laden als der Pi von seiner SD-Karte und fast fünfmal so schnell schreiben. Abgesehen von einer kürzeren Boot-Zeit starteten Anwendungen schneller als beim Pi. Darüber hinaus verpufft dieser Vorteil in der Praxis je-

doch und dürfte erst bei speziellen Anwendungen zum Tragen kommen, die viele Daten zwischen den Karten transferieren müssen.

Ähnlich sah es beim RAM aus: Der ist beim C4 laut Benchmark klar schneller angebunden (gemessen mit mbw 100), in unseren Tests brachte dies kaum Vorteile. Vermutlich profitieren besonders speicherintensive Anwendungen wie gecachte Datenbanken davon. Das ist im übrigen auch die von Pi-Erfinder Eben Upton gegebene Erklärung für das Erscheinen der 8GByte-Version: Datenbankserver würden vom zusätzlichen Speicher profitieren. Mit der Gigabit-Ethernet-Anbindung kann man dann die Daten auch schnell über das Netzwerk ausliefern. Weder C4 noch Pi 4 gaben sich hier eine Blöße, der mit iperf ermittelte Durchsatz war mit 936MBit/s am theoretischen Maximum.

USB 3.0

Anders sah dies bei den USB3.0-Anschlüssen aus, deren Datenrate theoretisch bei 640MByte/s liegt. Wir haben den Speed-Test einmal mit einer SanDisk Extreme Portable SSD (250GB, maximale Leserate 570MByte/s) sowie mit einem SanDisk Extreme Pro USB 3.1 (maximale Leserate 420MByte/s) und dem Tool gnomes-disks durchgeführt. Mit rund 320MByte/s konnte der C4 nur die Hälfte des maximalen erreichen. Der Pi erzielte mit 64-Bit-Betriebssystem 370MByte/s, wohingegen die 32-Bit-Version mit nur 270MByte/s dahintröpfelte. Einmal mehr zeigte sich, dass die 64-Bit-Version des Raspberry Pi OS eine gute Wahl ist.

In zwei Punkten hat diese Version allerdings Nachteile: Erstens wird die PiCam noch nicht fehlerfrei unterstützt und zweitens funktioniert die Hardwarebeschleunigung für Videos in VLC und Chromium noch nicht. Stattdessen sind alle 4 ARM-Kerne ordentlich am Rodeln, um Videos ruckelfrei abzuspielen. Das gleiche Problem hat aber auch der C4, dessen

Benchmarks

	Odroid C4 eMMC (64 Bit)	Odroid C4 SD-Card (64 Bit)	Raspberry Pi 4B 8GB (32 Bit)	Raspberry Pi 4B 8GB (64 Bit)
7z Compress (MIPS)	3216	3085	3512	3711
7z Decompress (MIPS)	7012	6706	7706	6099
mbw 100 (MEMCPY/DUMP/ MCBLOCK) (MiB/s)	2383/2346/3599	2599/2546/4000	1584/1916/2759	1885/1904/2771
USB 3.0 (MByte/s)	324	319	215	370
OpenSSL AES Speed (16384 Bytes) (MByte/s)	78	78	85	76
OpenSSL SHA256 Speed (16384 Bytes) (Mbyte/s)	830	829	163	160
OpenCV (ms)	1205	1213	1726	892
Tensorflow Lite (ms)	280	270	305	240
iperf (Mbit/s)	940	940	940	940

Beschleuniger mangels Treiber auf dem von uns getesteten Ubuntu 20.04 brach lagen. Wer Videos in H.264 und H.265 in 4K schauen will, installiert sich besser die dedizierte Media-player-Distribution CoreElec.

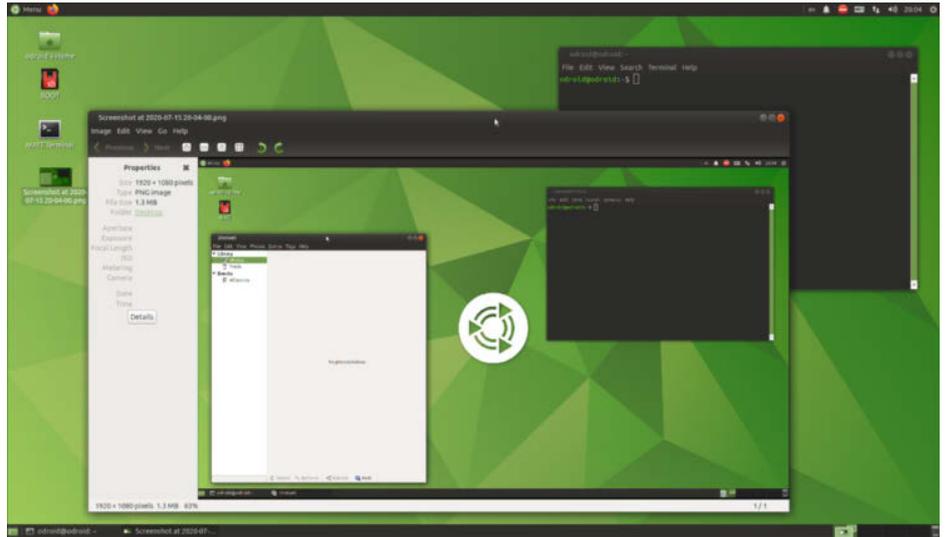
Vergleich

Rein vom Leistungsvergleich zieht der Odroid C4 in den meisten Fällen den kürzeren gegen den Pi mit 64-Bit-OS. Der Pi hätte sogar noch Übertaktungspotenzial von den normalen 1,5GHz auf 2GHz, wofür aber ein Kühlkörper fällig wird. Der C4 ist trotz standardmäßig verbauten Kühlkörpers jedoch schon fast am Ende der Fahnenstange. 2,016GHz hält Hardkernel noch für vertretbar, bei 2,1GHz riskiert man Abstürze. Bei 75 Grad setzt das Throttling ein.

Leider fehlen dem C4 WLAN und Bluetooth. Die Funktionen kann man zwar per Dongle nachrüsten, was aber zusätzlich kostet und zudem einen USB-Port belegt. Gerade mit seinen vier USB3.0-Ports kann der C4 aber auftrumpfen, sofern man ihn mit externen Platten als NAS einsetzen will. Dazu kommt auch noch ein besseres Spannungsversorgungskonzept als beim Pi 4: Die Spannungen für 5V, 3,3V etc. werden von einzelnen Reglern erzeugt, die von einer externen Quelle zwischen 6 und 17V an die Hohlbuchse geliefert werden. Damit ist beispielsweise die 5V-Versorgung der USB-Ports stabil - was für externe Platten substanzial ist - solange das externe Netzteil die Spannung über 6V hält. Das schafft man auch mit Wald- und Wiesennetzteilen.

Spannungsversorgung

Beim Pi 4 steht und fällt alles mit dem externen 5V-Netzteil (das eigentlich 5,1V liefert) und dessen Zuleitung. Zieht der Pi zuviel Strom durch ein schlechtes Kabel, sinkt die ankommende Spannung schnell unter 5V, der Pi schlägt mit seinem Blitzsymbol Alarm und Spannungen an USB-Ports brechen ein. Da die



Ubuntu 20.04 mit Mate-Desktop auf dem C4

Spannung beim C4 irgendwo zwischen 6 und 17V liegen darf, kann man alternativ LiPo-Akku mit 7,4V oder 11,1V zur Versorgung einsetzen. Das ist für manche mobilen Projekte nützlicher als der Griff zur USB-Powerbank. Bei der Leistungsaufnahme lag der C4 mit 4,6W bei Vollast unter dem Pi 4, der 6W aufnahm. Im Leerlauf lag der C4 hingegen bei 3,1W, der Pi 4 nur bei 3W.

Zur Kommunikation mit seiner Umwelt hat der C4 diverse GPIOs und sogar einen IR-Port zum direkten Empfang der Signale von Fernbedienungen. Das kann dann nützlich sein, falls ein billiges HDMI-TV keine Unterstützung für CEC mitbringt und die Signale der TV-Be-

dienung nicht weiterleitet, etwa um LibreElec/Kodi zu steuern. Auf seiner Pin-Leiste bietet der C4 25 GPIOs an, darunter PWM-Ausgänge, A/D-Eingänge, CAN, SPI und I²C. Ein zusätzlicher Audio Expansion Header führt einen I²S-Bus sowie SPDIF nach Außen, an denen man etwa Sound-Karten oder D/A-Wandler anschließen kann.

Software

Hardkernel stellt ein Ubuntu 20.04 für den Odroid C4 mit einem angepasstem Kernel und Mate-Desktop als Oberfläche bereit. Die Oberfläche läuft flüssig und Fenster lassen sich

Flashspeicher (iozone 16374k)

	C4	Pi 4
SanDisk Ultra A1 (read/write)	77951/16162 kBytes/s	44216/18535 kBytes/s
Hama Memory Pro 4K (read/write)	77972/23964 kBytes/s	45062/21383 kBytes/s
eMMC (read/write)	168944/109872 kBytes/s	-/-



neu. innovativ. professionell.

Mobile CNC-Fräse

- + Großformatige Bearbeitung bis 2.500 x 1.250 mm
- + Intuitive Bedienung über Touch-Pad
- + Flexible und kompakte Bauform, Auf- und Abbau in 3:00 Minuten
- + Bequeme Datenübertragung via WLAN und USB

www.sauter-shop.de/cnc-fraese/

+49 (0)8143 / 99 129 0

ohne Ruckeln und Schlieren verschieben, was auf eine funktionierende 2D-Beschleunigung hinweist. Anwendungen wie Firefox und LibreOffice laden schneller als auf dem Pi und reagieren auf Klicks prompt. Für einfache Ein-sätze kann der C4 gut als PC-Ersatz herhalten. Das angepasste Ubuntu hat Hardkernel-typisch aber leider wegen spezieller Treiber und angepasster Module ein paar Macken, sodass sich bestimmte Pakete nicht ohne erhebliche Frickelarbeit nachinstallieren lassen; die Video-player VLC und mpv gehören beispielsweise dazu.

Beim Pi 4 mit Raspberry OS läuft unter 32 Bit alles problemlos, bei 64 Bit gibt es noch die oben erwähnten Macken, die aber dem-nächst ausgegübelt sein sollten. Die GPIOs

lassen sich aber wie gewohnt ansteuern. Der Desktop läuft in beiden Versionen etwas träger als beim C4. Einen Unterschied zwischen der 4Gbyte und der 8Gbyte-Version konnten wir nicht feststellen. Als Datenbankserver oder als Docker-Plattform mit vielen parallel laufen-den Containern könnte der doppelte Speicher aber Nutzen bringen.

Fazit

Ein pauschales Urteil fällt schwer. Der C4 dürf-te wegen der vielen USB-Ports wohl eher NAS-Fans und wegen der etwas besseren Video-hardware Kodi-Enthusiasten ansprechen. Mit der flexibleren Spannungsversorgung wird der Autor ihn in weiteren mobilen Robotikpro-

jekten einsetzen und dort die USB-Powerbank weglassen. Wer gerne mit Ubuntu arbeitet, dürfte sich auf dem C4 wohl fühlen.

Der Pi 4 bringt bereits WLAN und in allen Punkten ausgereifte Software mit. Im 64-Bit-Modus zieht er richtig durch. Als Desk-top-Ersatz lässt er sich dank der zwei HDMI-Ports mit 2 Monitoren betreiben. Die Platzie-rung der Buchsen ist allerdings nervig eng und erfordert im Dual-Mode Spezialkabel. Die Pi-xel-Bedienoberfläche wirkt weiterhin altba-cken, aber es gibt ja alternative Distributionen. Durch seine große Community und dem dar-aus resultierenden fantastischen Support sowie der leistungsfähigen Hardware bleibt der Pi aber gerade für Einsteiger die erste Wahl. —dab

Hardware-Übersicht

	Odroid C4	Raspberry Pi 4B
SoC		
Typ	Amlogic S905X3	Broadcom BCM2711
Prozessor-Kerne	4 × A55	4 × A72
max. Taktfrequenz (GHz)	2.0	2.0 (1.5 Standard)
Crypto-Unterstützung	ja	nein
Architektur/Befehlssatz	ARM v8	ARM v8
GPU-Typ	Mali-G31P	VideoCore VI
RAM (GByte)	4	2/4/8
Datenübertragung		
USB	4 × USB 3.0, 1 × USB 2.0 OTG	2 × USB 2.0, 2 × USB 3.0
Ethernet	Gigabit	Gigabit
WLAN	nur mit Dongle	Dual-Band 802.11 b/g/n/ac
Bluetooth	nur mit Dongle	Bluetooth 5.0 + BLE
Audio/Video		
Monitor	1 × HDMI 2.0a (HDR, CEC)	2 × MicroHDMI 2.0 (CEC)
LCD/Analog-Video	nein/nein	ja/ja
Kameraanschluss	nein	ja
Audio-Ausgänge	nein	3,5mm Klinke
Audio-Eingänge	nein	nein
Video		
H.264 Decode/Encode	4Kp30/1080p60	1080p60/1080p30
H.265 Decode/Encode	4Kp60/1080p60	2160p60/-
GPIO		
Digital I/O	25	28
AD/DA/PWM	2/0/4	0/0/2
IR	1	0
LEDs	Power (rot), Heartbeat (blau)	Power (rot), Kartenzugriff (grün)
Busse	I ² C, CAN, SPI, I ² S	I ² C, SPI, I ² S
Massenspeicher		
microSD	ja	ja
eMMC	ja	nein
Energieversorgung		
Netzteil	5.5V bis 17V (12V, 2A empf.)	5V, 3A
Stromanschluss	Hohlbuchse 5.5/2.1mm	USB-C
PoE	nein	ja
Preis	ca. 70 Euro	4GByte 60 Euro / 4 GByte 90 Euro

3D-Kurs für Maker: Konstruieren mit Blender 2.8

Finale bei unserem Videokurs: Wir beenden im Herbst unser Angebot beim Videoportal *Vimeo*. Doch unsere Tutorialfilme zum Einsatz von Blender als 3D-Konstruktionswerkzeug kann man weiterhin kaufen – als komplettes Paket im *heise shop*.

von Peter König

Im Dezember des vergangenen Jahres hat Make zusammen mit dem Blender-Tutor und Buchautor Carsten Wartmann einen Videokurs gestartet, um Makern zu helfen, Vorlagen für ihre Fräs- und 3D-Druck-Projekte mit der kostenlosen 3D-Software *Blender* selbst zu konstruieren. Die Idee: Wir wollten unsere Einstiegshilfe anhand verschiedener kleiner, in sich abgeschlossener Projekte geben. So können sich alle gezielt die für sie passenden Lektionen herausuchen und müssen nur für jene Tutorial-Videos je ein paar Euro investieren, die ihnen direkt nützen.

Nach und nach kamen mehr Video-Lektionen hinzu – insgesamt neun reguläre Folgen sowie zwei Bonus-Episoden mit insgesamt 5 Stunden Spielzeit liefern Makern inzwischen das nötige Rüstzeug, um mit Blender erfolgreich mit 3D-Eigenkonstruktionen loszulegen.

Ende bei Vimeo ...

Bisher war der Kurs ausschließlich über das Video-on-Demand-Portal der Online-Platt-

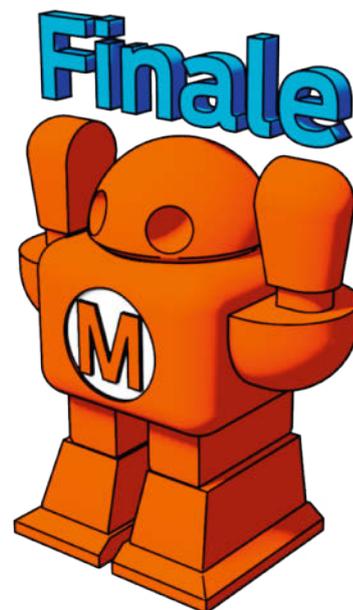
form *Vimeo.com* zu bekommen. Mit Ablauf des 15. Oktober 2020 endet allerdings unser Angebot bei *Vimeo*. Wer vorher noch seine Sammlung unserer Blender-Tutorials vervollständigen will, kann bei *Vimeo* bis zu diesem Zeitpunkt einzelne Folgen des 3D-Kurses kaufen, im Browser anschauen und vor allem: auf den eigenen Rechner herunterladen und sie sich damit dauerhaft sichern. Dazu raten wir dringend (siehe Kasten). Wer gleich alle Episoden haben will, bekommt ab jetzt auch den gesamten Kurs auf einmal für 19,90 Euro. Gegenüber dem Einzelkauf aller kostenpflichtigen Tutorials spart man dabei rund 4 Euro.

... weiter im heise Shop

Alternativ dazu und ebenfalls zum Preis von 19,90 Euro bekommt man den kompletten Kurs neuerdings auch im *heise shop* (Links zu allen Angeboten siehe Kurz-URL am Ende des Artikels). Allerdings kann man dort keine einzelnen Folgen mehr erwerben, es gibt nur noch den gesamten Kurs mit seinen neun Tu-

torials und zwei Bonus-Clips am Stück. Die bisher kostenlosen Episoden *Was ist neu in Blender 2.8?* und das Projekt: *Fingerring mit Beschriftung* (siehe auch *Make 2/20*, S. 114) sowie die Bonus-Videos gibt es übrigens weiterhin gratis zu sehen: Wir haben sie in die Episodenliste auf der Webseite zum Kurs eingebunden, auf der man auch die Beispielfdateien und Templates zu den einzelnen Folgen bekommt.

Wenn die Software-Entwickler im Zeitplan bleiben, erscheint übrigens noch im August die Blender-Version 2.9. Anders als beim Schritt auf 2.8 wird sich aber diesmal die Bedienoberfläche des Programms kaum ändern, sodass unsere Tutorials weiter zu den Funktionen im aktuellen Blender passen. Außerdem ist im Juni mit Blender 2.83 erstmals ein *Long-Term-Support-Release* erschienen, das noch zwei Jahre lang mit Wartungsupdates versorgt wird. So bleibt unser *3D-Kurs für Maker: Konstruieren mit Blender 2.8* noch lange aktuell. —pek



Bisher bei *Vimeo* zu sehen, ab Oktober ausschließlich im *heise shop* erhältlich: Der *Make*-Videokurs mit Blender-Tutor Carsten Wartmann.

Wichtiger Hinweis

Wer einzelne Folgen unseres Blenderkurses über die Videoplattform *Vimeo* gekauft hat, kann sich diese **nur noch bis einschließlich 15. Oktober 2020 herunterladen!** Da wir unser Angebot dort einstellen, ist es später auch nicht mehr möglich, die Videos bei *Vimeo* im Browser abzuspielen.

Alles zum Artikel
im Web unter
make-magazin.de/x7p9

Räume gesucht in Mannheim

Nach 10 Jahren muss das Mannheimer *RaumZeitLabor* zum 31. Oktober aus seinen Räumen in der Boveristraße ausziehen. Derzeit ist man auf der Suche nach einer neuen Unterkunft.

raumzeitlabor.de

Umzug in Chemnitz

In Chemnitz ist das *Holz-kombinat* umgezogen und seit kurzem in der Schiersandstraße 21 beheimatet.

holzkombinat.com

Maker-Termine

Bornhack

11.–18. August

Fünen, Dänemark
bornhack.dk/bornhack-2020

„Zukunft für Alle“

25.–30. August

online und in Leipzig
zukunftfueralle.jetzt

Make Your School

„Maker-Woche“

8.–11. September

online
makeyourschool.de/maker-festival

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



OpenBike

Bikesharing-Plattform sucht Hackerspaces

Im Ulmer *Verschwürhaus* arbeitet man an einer Open-Source-Plattform für Bikesharing. Unterstützt von der Stadt entsteht ein System, das die nötige Hard- und Software umfasst und aus einfachen Zweirädern App-gesteuerte Sharing-Räder macht. Dank offener Daten und standardisierter Schnittstellen sollen andere Kommunen und Einrichtungen es möglichst einfach übernehmen und anpassen können. Auch die Anbindung an andere Mobilitätsangebote wie den ÖPNV und Carsharing soll möglich sein. Erste Prototypen gibt es bereits, sodass das OpenBike-Team nun Hackerspaces sucht, die Teile oder sogar die ganze Plattform austesten wollen.

Dafür sind eine Reihe an Problemen zu lösen: Welche Räder eignen sich? Welches Schloss und welche Hardware zum Tracken sollen genutzt werden? Und wie landen die Trackingdaten in einer Verwaltungssoftware? Aktuell arbeitet man in Ulm mit einfachen Stadträdern, an denen ein Hinweisschild angebracht

ist. Abgeschlossen werden sie mit schweren Zahlenschlössern, da die existierenden Smart Locks schwierig zu befestigen sind. Grundlage für das Tracking ist der LoRaWAN-GPS-Tracker Dragino LGT-92 – langfristiges Ziel ist aber ein eigener Tracker auf Basis eines ESP12-Moduls. Für die Positionierung werden die umliegenden WLANs genutzt und die Daten schließlich über das *The Things Network* versendet. Wo die Räder gerade unterwegs sind, können Interessierte und das OpenBike-Team auf einer Karte und App verfolgen.

Derzeit testen bereits Beschäftigte der Stadt Ulm die OpenBikes und in Halle hat der Hackerspace *Eigenbaukombinat* ein OpenBike-System mit einer Tracking-App statt Hardware-Tracking in Betrieb genommen. Um die Plattform weiter zu verbessern sucht das Team nun mehr Hackerspaces, die ebenfalls testen und mitentwickeln. —hch

► ulm.dev/projects/openbike

#HackTheSummer

Programmier-Wettbewerb für Jugendliche

Die *Gesellschaft für Informatik* hat mit dem Bildungsministerium einen Sommer-Wettbewerb gestartet, bei dem Kinder und Jugendliche digitale Ideen für eine bessere Welt entwickeln sollen. Jeweils bis zu fünf Schülerinnen und Schüler können als Team digitale Prototypen entwerfen, die eines oder mehrere der 17 Nachhaltigkeitsziele der Vereinten Nationen im Blick haben. Die Ziele sind in fünf Kategorien organisiert: People (Menschen), Planet (Erde), Prosperity (Wohlstand), Peace (Frieden) und Partnership (Partnerschaft). Unter diesen Oberbegriffen versteckt sich jeweils eine Handvoll konkreter Aspekte. So sind unter dem Punkt „Erde“ der Klimawandel, die Rettung von Meeren und Ozeanen, der

Erhalt von Ökosystem und der nachhaltige Umgang mit natürlichen Ressourcen gelistet. Hier könnten Projektideen rund ums Reparieren, Weiterverwenden und Recyclen entstehen.



Wer über 18 Jahre alt ist, kann ebenfalls mitmachen – als Mentorin oder Mentor. Alle zwei Wochen können sich die Teilnehmenden mit diesen in Online-Meetups austauschen und beraten lassen. Die besten Teams sollen im Oktober zu einem Prototyping-Wochenende nach Berlin kommen. Das erarbeitete Projekt sollte in einem bis zu dreiminütigen Video oder dreiseitigen Textdokument mit Fotos eingereicht werden. Teilnahmeschluss ist der 19. September 2020. —hch

► hack-the-summer.de

Make Your School Remote

Auch die Hackdays sind nun Online-Workshops

Zu ihren Hackdays an Schulen bringt die Initiative „Make Your School“ normalerweise Materialkoffer mit, die alles enthalten, was man für kreative Digitalprojekte braucht: Mikrocontroller, Sensoren, Motoren und Breadboards. Nachdem durch die Corona-Pandemie bereits zahlreiche Hackdays abgesagt werden mussten, stellte sich die Frage, wie Remote-Hackdays für das Homeschooling aussehen könnten. Ende Juni fand ein Test am Hannoveraner Gymnasium Goetheschule statt. Wie die Präsenzveranstaltungen sind die Remote-Hackdays für 15 bis 30 Teilnehmende ausgelegt und erstrecken sich über zwei Tage. Die Elektronikteile aus den Material-Kits verteilen die Schulen dafür an die Jugendlichen.

Mentoren und Mentorinnen aus verschiedenen technischen Fachbereichen begleiten die Remote-Hackdays weiterhin inhaltlich, nur schalten sie sich eben virtuell zu: Es wird ein Videokonferenztool verwendet, wobei die Schüler in Breakout-Rooms die Möglichkeit erhalten, gemeinsam zu brainstormen und zu tüfteln. Die Ideen erstellen sie an gemeinsamen virtuellen Pinnwänden. Oliver Burmeister, begleitender Lehrer an der Goetheschule, zieht ein positives

Fazit: „Bei den Hackdays in der Schule übernehmen alle die Aufgaben, die sie am Besten können. Die einen programmieren und die anderen bauen das Gehäuse. Das ging hier nicht. Wer mitmachen wollte, konnte sich nicht an einem Punkt rausziehen, sondern alle mussten alles machen und hatten eine hohe Verantwortung für ihr eigenes Projekt. Das hat eigentlich ganz gut geklappt.“ Den ausführlichen Bericht vom Remote-Hackday finden Sie online.—esk



► heise.de/-4840597

Aktionswoche Meet and Code

Die Europäische Aktionswoche *Meet and Code* wird dieses Jahr zwischen dem 1. September und 31. Oktober digital stattfinden. Bis zum 10. September können Vereine sich um einen Zuschuss von 500 Euro für Codingprojekte mit Kindern und Jugendlichen bewerben.

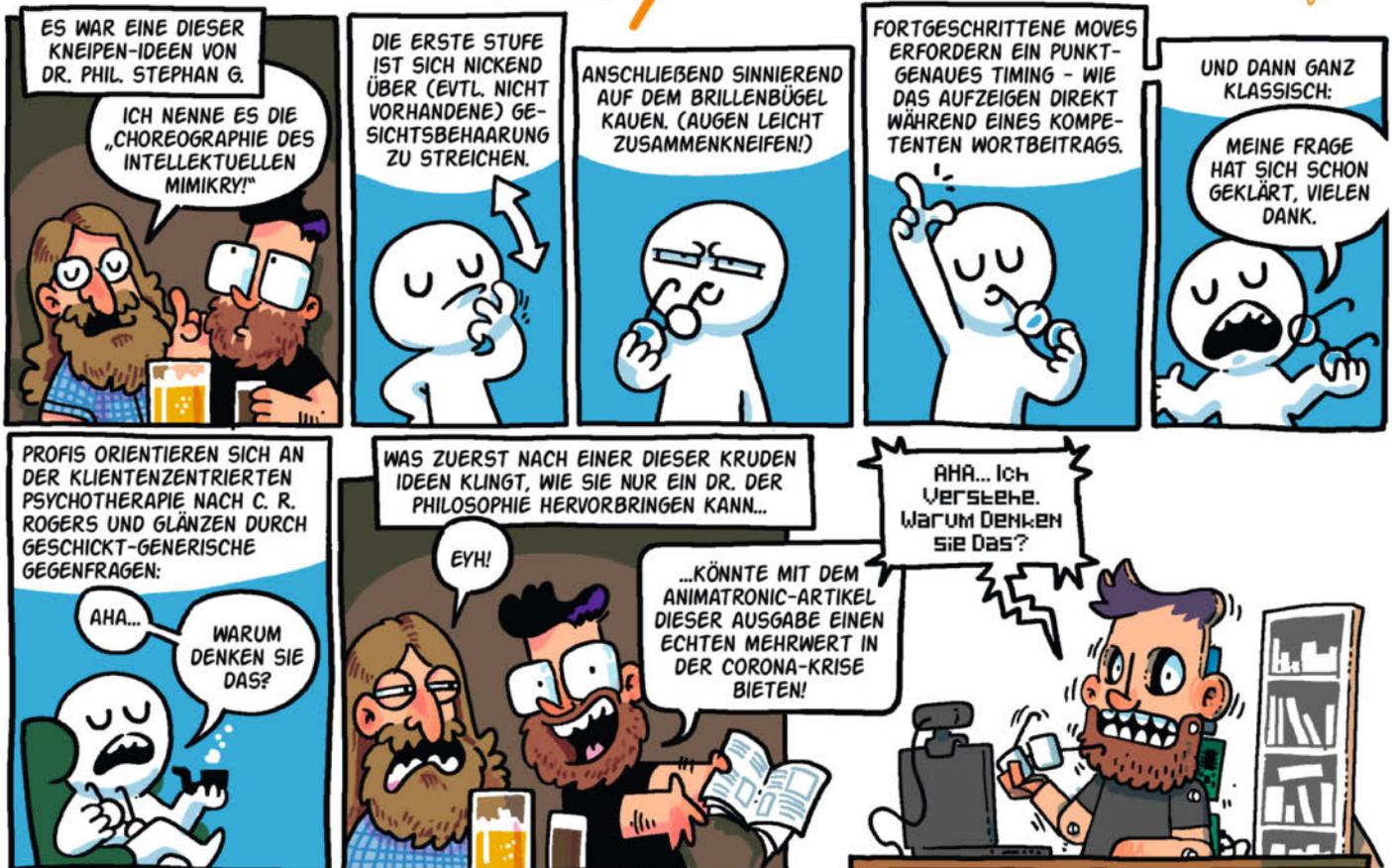
meet-and-code.org/de/de

38

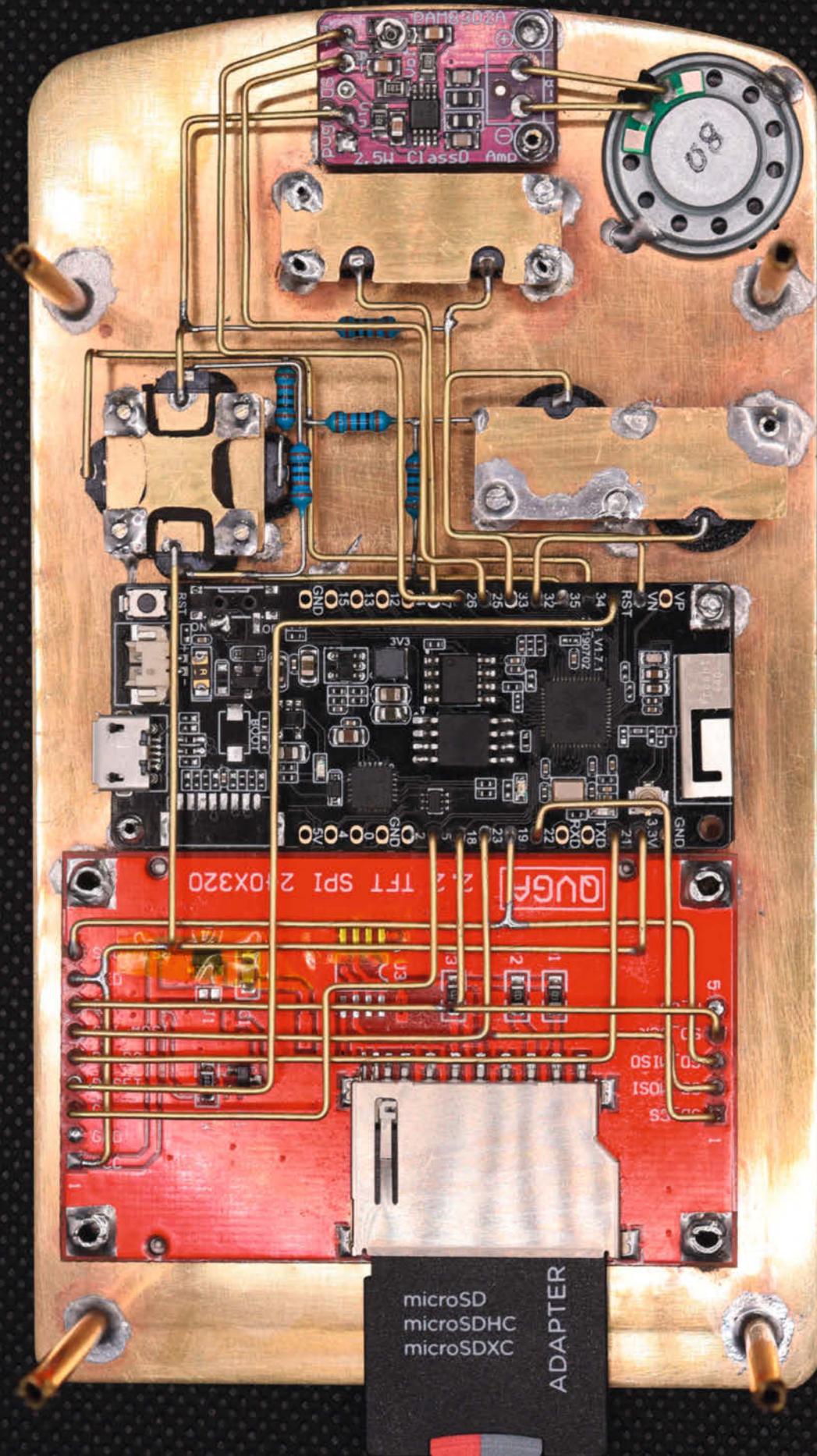
Intellektuelles Mimikry

Kolophonium

von und mit @beetlebum



Was uns inspiriert





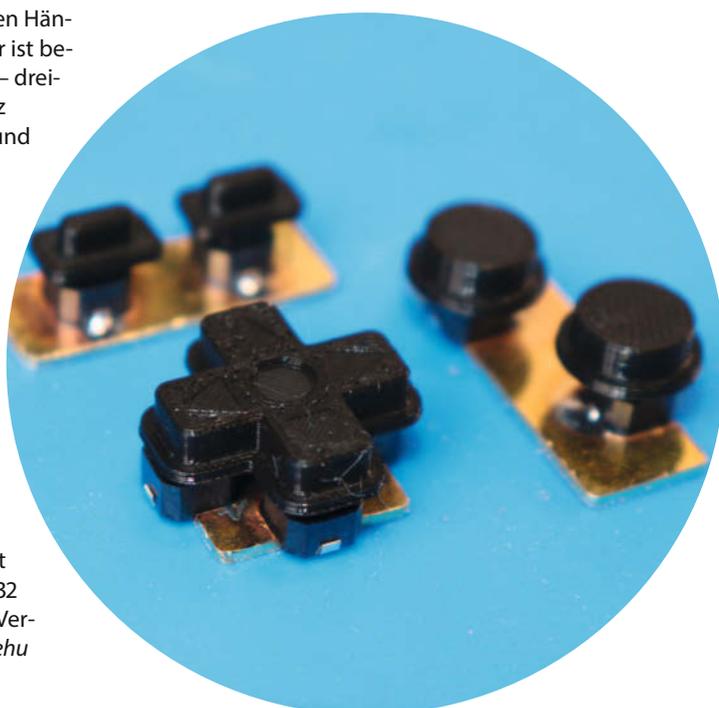
Bilder: Jiří Praus

Steampunk-Gameboy aus Messing

Als Jiří Praus den *GameBoy Color* seiner Kindheit wieder in den Händen hielt, überkam ihn eine Welle der Nostalgie. Der Künstler ist bekannt für seine skulpturhaften – und dennoch funktionalen – dreidimensionalen Schaltungen aus Messingdraht, die eine ganz eigene Kunstform sind. Aus dieser Mischung von Nostalgie und Schaltkreiskunst ist der *BrassBoy* entstanden: Ein Gameboy mit einer glänzenden Messingfront und einer transparenten Rückseite, durch die man sorgfältig gebogene Messingverbindungen bewundern kann.

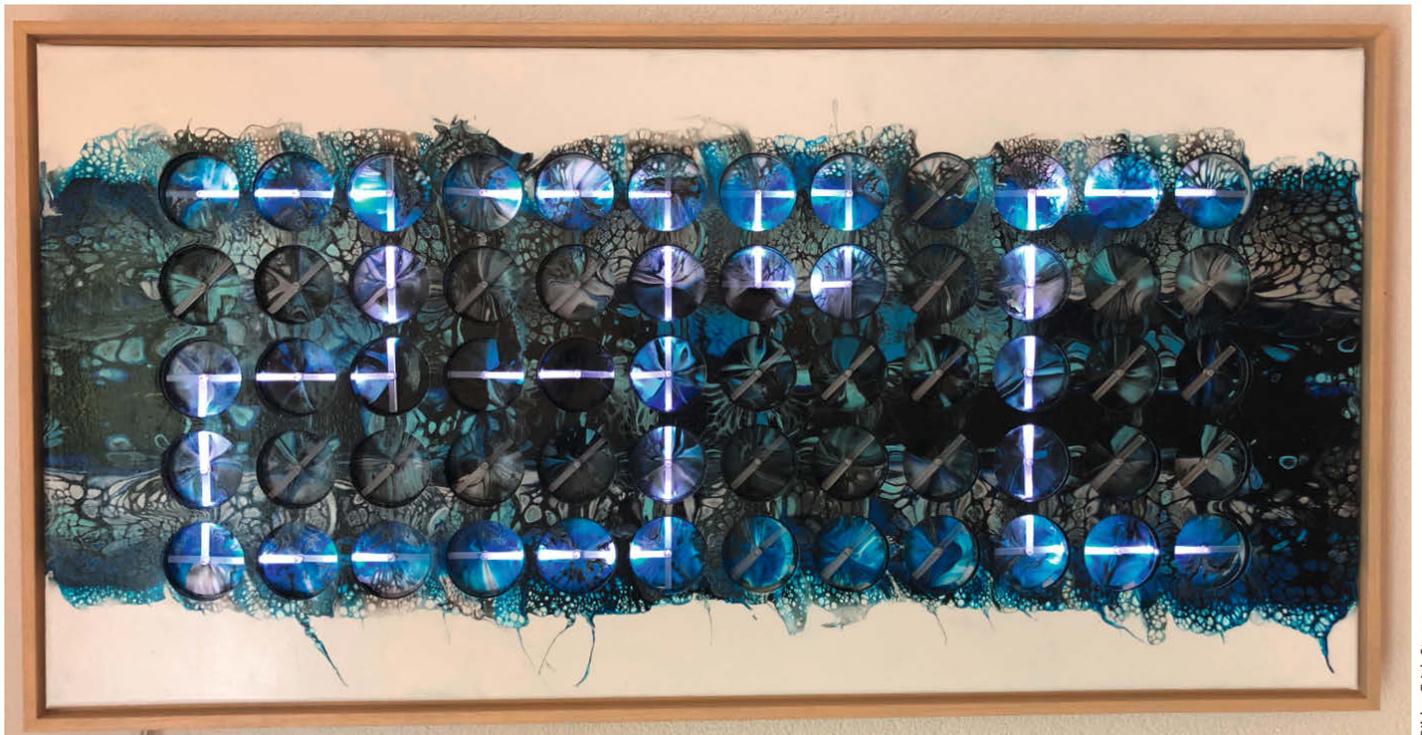
Statt der Original-Hardware hat Praus einen Emulator verbaut und kann so die alten Klassiker spielen. Das Innenleben des BrassBoy enthält einen ESP32 in Form eines LilyGo TTGO T8 Dev-Boards. Dazu kommt ein QVGA-TFT-Display mit einer Auflösung von 320 × 240 Pixeln und ein Klasse-D-Verstärker mit 2,5 Watt. Die wahre Elektronik-Hexerei ist allerdings die Konstruktion: Die Bauteile sind in einem Käfig aus präzise geformten Messingleitungen eingeschlossen – und dieser elegante Schaltkreis funktioniert natürlich nur mit viel Planung. So hat Praus zum Beispiel die Belegung der GPIO-Pins auf dem ESP32 optimiert und genau darauf geachtet, dass alle kreuzenden Verbindungen gut isoliert sind.

—rehu



► jiripraus.cz

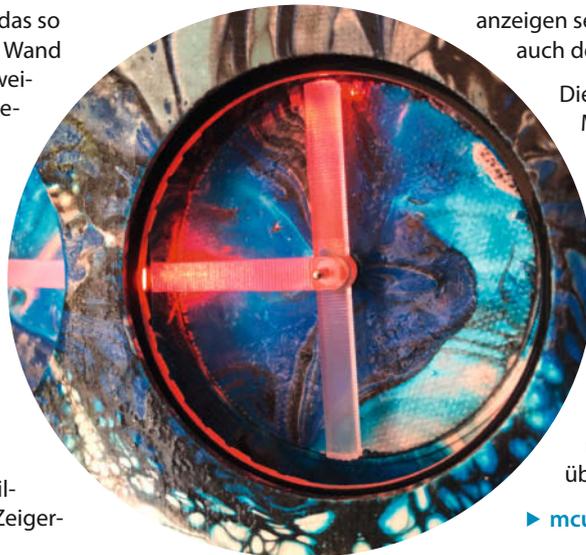
► youtube.com/watch?v=nxa5WelKBLw



Bilder: Erich Styger

60 Milliarden Lichter

Ein technisch anspruchsvolles Projekt, das so schön aussieht, dass man es gut an die Wand hängen kann – dieses Ziel hat der Schweizer Ingenieur Erich Styger mit seiner mechatronischen Kunstinstallation *60 Billion Lights* erreicht. Die Wandanzeige ist gleichzeitig Uhr, Messstation und Spielekonsole, sieht aber auch unbeleuchtet äußerst dekorativ aus. Sie ist 103cm x 53cm groß und besteht aus 60 runden Mini-Anzeigen. In jedem der runden Pixel drehen sich drei transparente Arme aus dem Lasercutter. Insgesamt stecken 2400 RGB-LEDs und 120 Schrittmotoren in dem Projekt, die jeweils einzeln angesteuert werden können. Mehr als 60 Milliarden Kombinationen von Licht und Zeiger-



anzeigen seien damit möglich, so Styger – woher auch der Name des Projekts rührt.

Die Steuerung der Elektronik übernehmen 15 Mikrocontroller (15 NXP LPC845), die über das RS-485-Protokoll kommunizieren. Neben einer Echtzeituhr ist auch ein Sensor verbaut, so dass die Installation Zeit, Temperatur und Luftfeuchte anzeigen und dabei auf die Helligkeit der Umgebung reagieren kann. Über das Adafruit-Bluetooth-Modul BLE SPI Friend und eine App von Adafruit kann die Installation bequem gesteuert werden. Zusätzlich gibt es auch ein Command Line Interface zur Kommunikation über die serielle Schnittstelle. —hch

► mcuoneclipse.com/2020/05/24/60





Bilder: Chris Janka

Ein mechanisches Orchester

Humanizing durch Maschinen ist der Untertitel zu diesem Projekt, das Tüftler und Musiker um den experimentierfreudigen Wiener Studio-Betreiber Chris Janka entwickelt haben: Das *totally mechanized MIDI-Orchestra* ist ein bis zu 100-köpfiges Orchester, aus Motoren, Relais, Elektromagneten und pneumatischen Bauteilen, das jedes beliebige MIDI-File auf seine Weise interpretieren kann. Es besteht aus Instrumenten mit so klingenden Namen wie *The Stepper Motor Organ*, *The Xylocloud*, *The Beer Bottle Organ* und *The Drumulator*. Im Juni konnten Musiker eigene Kompositionen auf der Webseite des Projekts hochladen, die dann vom MIDI Orchestra umgesetzt wurden. Die Aufnahme und Steuerung läuft über einen RasPi mit MIDI-Output und Kamera, an dem zwei USB-Soundkarten hängen – eine für den Audio-stream und eine für die Aufnahme.

MIDI ist ein digitales Protokoll zur Musikprogrammierung, das Anfang der Achtziger Jahre entwickelt wurde. Die Abkürzung steht für *Musical Instrument Digital Interface*. Obwohl das Protokoll MIDI als Grundlage für einen Großteil der populären Musik dient, da es zum Beispiel von Synthesizern verwendet wird, klingt es vielen Kritikern zu mechanisch und programmiert. Daraufhin wurden sogenannte *Humanizing*-Algorithmen eingeführt, um wieder eine gewisse „menschliche“ Ungenauigkeit in die Musik zu bringen. Doch das MIDI Orchestra *humanised* nicht – es *mechanised*! So klingt es, wenn analog-unge-naue Roboter digital programmierten MIDI-Files wiedergeben. —*rehu*

► midi-orchestra.net/



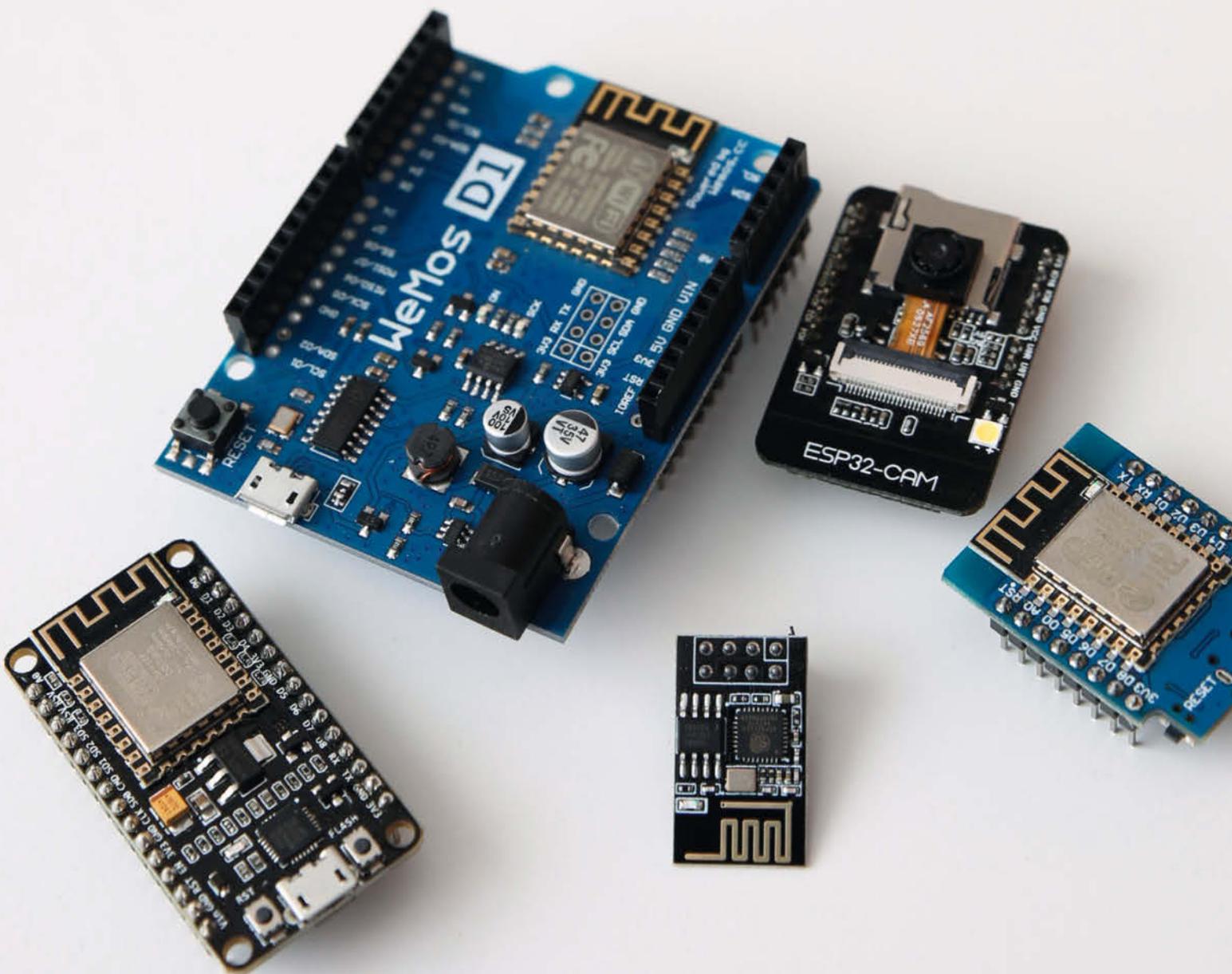
Was inspiriert Dich?

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

Smarthome-Firmware für ESP-Module

Module mit dem ESP8266/32-Prozessor sind klein, preiswert und können mit der richtigen Firmware eine Menge Aufgaben übernehmen, etwa um Ihr Haus smart zu machen. Eine Reihe fertiger Firmware-Versionen nimmt Ihnen das Programmieren ab, sodass Sie sich auf die Konfiguration von Sensoren und Aktoren konzentrieren können. Wir stellen Ihnen vier Lösungen vor.

von Heinz Behling



Ein Smarthome, in dem man sich selbst um fast nichts mehr kümmern muss, sondern in dem viele dienstbare, elektronische Geister die Arbeit und Überwachung übernehmen, wer wünscht sich das nicht? Doch diese Geister fertig kaufen, das geht richtig ins Geld. Da braucht man eine Zentrale, die meist schon einen deutlich dreistelligen Betrag kostet. Möchte man Lampen steuern, kostet jeder elektronische Schalter nochmal so zwischen 30 und 50 Euro. Dann kommen noch Temperatursensoren und elektrisch betätigte Heizkörperventile dazu. Ganz zu schweigen von Sicherheitseinrichtungen gegen Glasbruch, Einbruch, Hochwasser... Sehr viele fangen daher gar nicht erst mit der Automatisierung des eigenen Heimes an. Und von denen, die bereits unter erheblichem Geldeinsatz damit begonnen haben, geben viele auf, wenn bei ihnen (oder dem jeweiligen Ehepartner) der Finanzierungswille parallel zum Kontostand schrumpft. Aber wir sind doch technisch ein wenig versiert. Das muss doch per Selbstbau deutlich billiger gehen, oder? Und wenn ja, kann ich doch hoffentlich trotzdem die bereits gekauften Bestandteile weiterverwenden?

Ja und ja! Das ist heute kein Problem mehr. Die dienstbaren Geister stehen in Form von Platinchen mit ESP8266/32-Chips zur Verfügung. Diese WLAN-fähigen Boards enthalten nahezu alles, was man zur Smarthome-Steuerung braucht. Nahezu heißt: Sensoren, die zum Beispiel die Temperatur messen, oder Relais, die eine Steckdose schalten, müssen noch hinzugefügt werden. Und wie jede andere Hardware auch sind die kleinen Boards zunächst einfach nur dumme, nutzlose Stromverbraucher. Erst die richtige Software erweckt ihre Fähigkeiten zum Leben, also beispielsweise die WLAN-Funktionen oder die Steuerung und Abfrage diverser Ein- und Ausgänge.

Da diese Software fest im Modul eingebettet ist, nennt man sie *Firmware* (englisch *firm* = fest). Dabei bedeutet fest allerdings nicht, dass diese Software für immer und ewig ins Gedächtnis des Moduls gebrannt wurde, sie lässt sich durchaus austauschen. Anders wären ESP-Module ja auch nicht an all die vielen Aufgaben anpassbar, die sie zu lösen imstande sind. Allerdings ist dieser Vorgang nicht einfach durch Wechsel eines Datenträgers zu erreichen, wie etwa beim Betriebssystem eines Raspberry Pi, das auf einer wechselbaren Speicherkarte sitzt.

Und es kommt noch besser: Da viele der kommerziellen Smarthome-Produkte ebenfalls mit den ESP-Chips arbeiten, kann man die Firmware häufig auch darauf installieren, vorausgesetzt, man kommt an die entsprechenden Anschlüsse heran. Glücklicherweise gibt es dazu zahlreiche Anleitungen und Videos im Internet, sodass auch bereits angefangene Projekte große Aussichten haben, erfolgreich im Selbstbau erweitert und vollendet zu wer-

Kurzinfo

- » **Smarthome-Firmware für ESP8266/32-Module**
- » **Vorhandene Firmware aus Modulen sichern**
- » **Firmware ins Modul schreiben**

Mehr zum Thema

- » Eine Anleitung, wie man ESP-Module mit Hilfe eines Arduinos programmiert, finden Sie im Online-Artikel „ESP-Boards mit der Arduino-IDE programmieren“
- » Informationen zum Thema MQTT gibt es im aktuellen Make-Sonderheft „Make Node-RED Special“
- » Wissenswertes zum Thema IoT gibt es im Artikel „IoT-Alleskönner“ in der Make 6/19 ab Seite 8
- » Wie Sie Steckdosen smart machen, lesen Sie im Artikel „Smarte Steckdosen mit neuer Firmware“ im Make-Sonderheft 2018 ab Seite 80

Alles zum Artikel im Web unter make-magazin.de/x1yu

den. Aber warum sollte man die Firmware solcher Produkte überhaupt wechseln, schließlich funktionieren sie doch.

Wie ein Smarthome arbeitet

Da gibt es eine Reihe von Sensoren (Temperatur, Helligkeit, Bewegungssensoren usw.) und etliche Aktoren, also zum Beispiel Schalter, Ventile an der Heizung, Lüfter, Lampen und vieles mehr. In der Mitte sitzt eine Zentrale, die zum einen Messwerte von den Sensoren entgegennimmt und zum anderen entsprechend die diversen Aktoren steuert. Ist die Temperatur im Wohnzimmer zu niedrig, dreht die Zentrale das Ventil am Heizkörper der guten Stube auf.

Nun wollen die Hersteller dieser Anlagen natürlich Geld verdienen, es sei ihnen ja auch gegönnt. Daher halten sie die Art der Kommunikation der einzelnen Bestandteile untereinander oft geheim. Da kommt man dann mit Selbstgebautem nicht heran. Ausweg: Eine eigene Zentrale, von der man weiß, wie sie sich mit den anderen Teilen unterhält. Dann würden aber die bereits vorhandenen Sensoren und Aktoren nur noch Bahnhof verstehen, es sei denn, sie erhielten eine neue Firmware, die sie wieder ins System integriert.

Falls sich der vorhandene Smarthome-Server nicht mit einer Software wie OpenHAB oder FHEM auf solch eine barrierefreie Kommunikation umrüsten lässt, kommen auch selbstgebaute Zentralen auf Basis eines Raspberry Pi in Frage. Es reicht aber auch ein einfacher MQTT-Server, dessen Einrichtung und Benutzung im Make-Sonderheft „Node-RED Special“ ausführlich beschrieben wird. Mit der grafischen Programmiersprache Node-RED

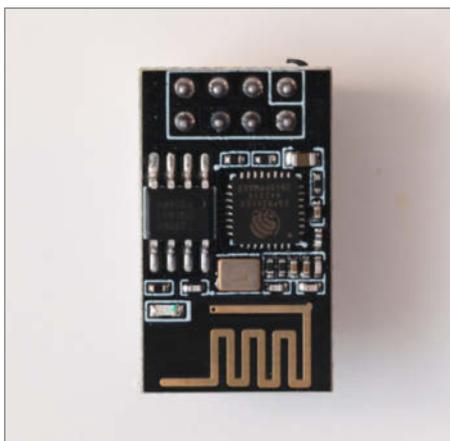
können Sie auch Regeln festlegen, was beim Eintreten bestimmter Ereignisse (Wohnzimmer zu kalt) zu tun ist (Heizung aufdrehen).

Bleibt nur noch die Aufgabe, die richtige Firmware zu finden und sie in die ESP-Module zu bringen. Für gewöhnlich wird die ESP-Firmware auf einem PC mit einem passenden Entwicklungssystem programmiert. Bekanntestes Beispiel ist die *Arduino IDE*, die entgegen ihrem Namen nicht nur für Arduino-kompatible Boards, sondern oft auch für ESP-Module benutzt wird. In diesem System schreibt man die Software (in einer C/C++-ähnlichen Programmiersprache). Sie wird daraufhin von der IDE in eine Form übersetzt (kompiliert), die für den jeweiligen Prozessor auf dem Board verständlich ist, die sogenannte Maschinensprache. Erst dieser Maschinensprache-Code wird dann als Firmware in den Flash-Speicher des Moduls geschrieben. Aber wer will (oder kann) schon eine komplette Firmware von Grund auf selbst schreiben (und debuggen)?

Den Software-Selbstbau haben schon viele andere erfolgreich in Angriff genommen. Glücklicherweise waren einige sogar bereit, ihre Ergebnisse online gratis als fertige Firmware verfügbar zu machen. Diese Smarthome-geeigneten Firmware-Versionen stellen wir Ihnen auf den folgenden Seiten vor.

Geeignete Module

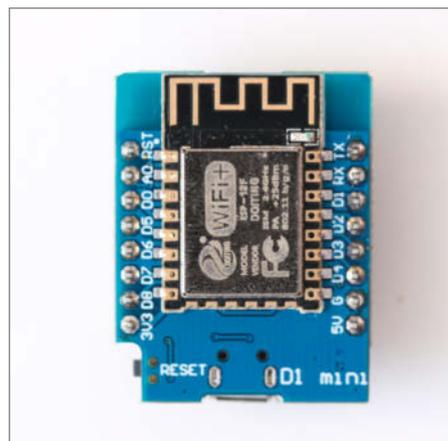
Module mit dem ESP8266- beziehungsweise dem ESP32-Chip gibt es in vielen Bauformen. Die kleinsten sind die ESP-01-X-Module **1**. Die Winzigkeit dieser Boards wird unter anderem dadurch erreicht, dass nur 3 GPIO-Pins nach außen geführt werden. Für einfache Schaltaufgaben (Steckdose) reicht das aber völlig.



1 Die ESP-01-Module sind klein und nur für einfache Aufgaben geeignet.



2 Module ohne USB-Anschluss erfordern einen Programmer zum Übertragen der Firmware.



3 Das D1-Mini-Modul mit USB-Anschluss hat 9 IO-Pins und einen analogen Eingang.

Diese Boards gibt es mit 512 KiB (ESP-01) und 1 MiB (ESP-01S) Flash-Speicher. Boards mit der Bezeichnung ESP-01M enthalten statt des ESP8266 den ESP8285 mit 1MiB Flash. Der ist Software-kompatibel zum 8266er, verbraucht aber etwas weniger Strom.

Etwas kompliziert sind die ESP-01-X-Module beim Übertragen der Firmware, da sie keine USB-Schnittstelle besitzen. Hier braucht man einen Programmer, zum Beispiel wie in Bild 2, oder einen Seriell-/USB-Wandler. Aber auch ein Arduino lässt sich dazu verwenden (Anleitung siehe Kurzinfor-Link). Bitte beachten, dass diese Module mit 3,3V Betriebsspannung arbeiten, mehr vertragen sie nicht!

Module mit USB-Schnittstelle sind wesentlich komfortabler in der Handhabung. Zur Übertragung der Firmware werden sie direkt am PC angeschlossen, der dann auch über USB die Stromversorgung (5V) übernimmt. Die IO-Pins vertragen aber auch bei diesen Modulen

nur 3,3V! Der PC sollte dem Modul dann einen COM-Port zuordnen. Falls das nicht funktioniert, fehlt der erforderliche Treiber (Download siehe Kurzinfor-Link).

USB-Module gibt es in zahlreichen Bauformen. Weit verbreitet ist das D1-mini-Modul 3.

NodeMCU-Boards besitzen noch mehr Anschlüsse, unter anderem auch einen I²C-Bus, sind aber auch deutlich größer 4.

Wer bislang Projekte mit dem Arduino Uno gebaut hat, für den könnte die Bauform Wemos D1 interessant sein, da sie die Maße des Arduino-Boards und nahezu die gleiche Anschlussbelegung besitzt. Damit wird die Weiterverwendung von Arduino-Uno-kompatiblen Shields möglich.

Großer Bruder ESP32

Auch den größeren Bruder des 8266er-Chips, den ESP32, gibt es auf einer ganzen Reihe von

Boards. Die auffälligsten Unterschiede sind die höhere Taktfrequenz (160-240MHz) der bis zu 2 Prozessorkerne und damit die entsprechende größere Arbeitsgeschwindigkeit gegenüber dem 8266 (80-160MHz). Interessanter fürs Smarthome ist jedoch die bessere Schnittstellenausstattung. So ist hier neben WLAN (mit doppelter Bandbreite und bis zu 150MBit/s) auch Bluetooth als drahtlose Schnittstelle vorhanden, und zwar in der energiesparenden BLE-Variante. Außerdem gibt es ein Ethernet-Interface (das auf vielen Boards allerdings nicht benutzt wird), einen Magnetfeld-Hall- sowie einen Temperatursensor. 10 der GPIO-Pins sind für Berührungssensoren geeignet. Es gibt jetzt analoge Ein- und Ausgänge, 2 I²C-Busse und 16 PWM-fähige Ausgänge.

Ein relativ neues Board mit diesem Chip ist das ESP32-CAM 5. Es verfügt zusätzlich über einen Mikro-SD-Kartenslot sowie über einen Eingang für ein Kameramodul. Damit lässt sich im Smarthome recht einfach eine Überwachungskamera ins System integrieren. Allerdings unterstützt nur ESPHome diese Funktion.

Von den vier hier besprochenen Firmware-Versionen kann nur eine (ESPurna) nicht mit dem ESP32-Prozessor arbeiten.

Ob in Ihren bereits vorhandenen kommerziellen Smarthome-Produkten ein ESP-Chip werkelt und die neue Firmware darauf installiert werden kann, erfahren Sie im Internet. Zum Abdrucken hier wären die Listen schlicht zu lang. Die meisten erfordern aber etwas Bastelarbeit, um den Anschluss an den Computer zu bewerkstelligen. Beachten Sie bitte, dass Sie mit dem Überspielen einer Firmware auf einen ESP die dort vorhandene Firmware unwiederbringlich löschen. Sie sollten daher vorher die alte Firmware sichern, damit Sie sie im Bedarfsfall wieder zurückschreiben können. Mehr dazu lesen Sie auf Seite 42.



4 Das NodeMCU-Board stellt noch mehr Anschlüsse zur Verfügung.



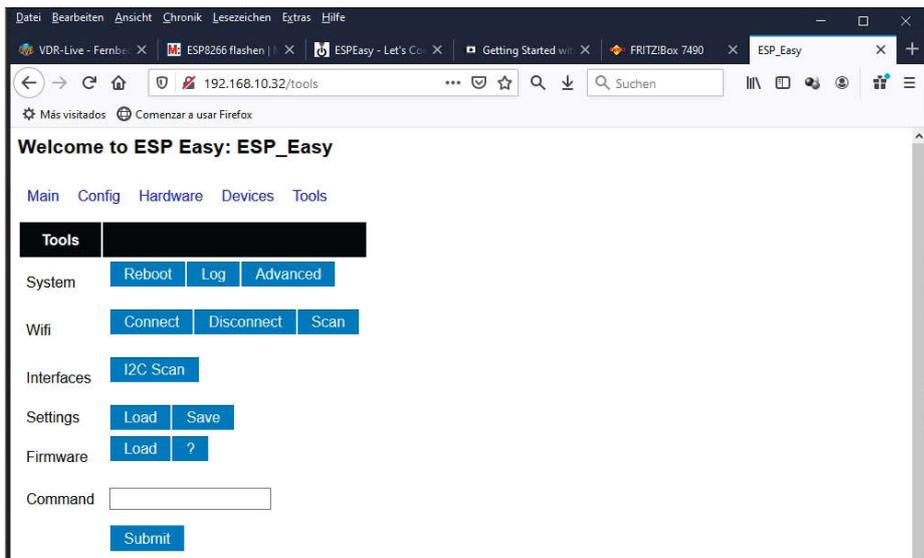
5 Das ESP32-Modul mit 2-Megapixel-Kamera ist kaum größer als ein D1-Mini-Board.

ESPEasy

Die Installation von ESPEasy ist – wie der Name vermuten lässt – einfach: Im downloadbaren ZIP-File ist nicht nur die fertige Firmware in Form dreier BIN-Dateien (je eine für 512, 1024 und 4096KiB Speichergröße des Moduls) enthalten, sondern auch gleich das Tool zum Übertragen der Datei in den Flash-Speicher des Moduls. Die gesamte Konfiguration geschieht über eine Web-Oberfläche. Beim ersten Start stellt ESPEasy dazu einen eigenen WLAN-Accesspoint bereit (ssid: ESP_EASY_0, Passwort: configesp). Falls Sie ein gebrauchtes ESP-Modul verwenden, auf dem ESPEasy schon einmal installiert war, sollten Sie zuvor allerdings den Modul-Flash komplett löschen, da die alte WLAN-Konfiguration beim Überspielen erhalten bleiben kann.

Nach dem Verbinden mit diesem Funknetz steht das Modul unter der IP-Adresse 192.162.4.1 zur Verfügung und präsentiert seine Web-Oberfläche. Dort zeigt es Ihnen die empfangbaren Funknetze an. Wählen Sie Ihres aus und geben Sie das Passwort ein. Nach einem Klick auf *Connect* zeigt Ihnen das Modul seine IP-Adresse, unter der Sie dann die Web-Oberfläche zur weiteren Konfiguration erreichen **6**.

Die Firmware kommt mit allen im Maker-Bereich gängigen Sensor- und Aktor-Modulen zurecht, die direkt an die GPIO-Pins, am I²C-Bus oder einer UART-Schnittstelle arbeiten und enthält entsprechende Steuerroutinen. SPI-Verbindungen sind leider nicht möglich. ESPEasy lässt sich auch auf zahlreichen kommerziellen WLAN-Smarthome-Komponenten installieren, beispielsweise auf preiswerten Schaltsteckdosen von Pearl (Markenname



6 ESPEasy wird über die Web-Oberfläche konfiguriert.

Luminea) oder aus dem Baumarkt OBI (siehe Tabelle auf Seite 41). Mit den entsprechenden Sendemodulen ausgestattet, kann sie sogar Geräte mit Funksteuerung im 433MHz-Bereich steuern, Infrarot-Signale lassen sich allerdings nur empfangen.

Eine Menge an Informationen steht auf der Internet-Seite von ESPEasy zur Verfügung. Gut geordnet sind insbesondere die Listen der verwendbaren kommerziellen Smarthome-Komponenten sowie der Sensoren/Aktoren **7**.

Und falls man dort ein gewünschtes Teil nicht findet: Eine Suche im Internet hilft oft

weiter, denn ESPEasy verfügt über eine große Community. Irgendjemand hat das Problem vielleicht schon gelöst.

Die Zusammenarbeit mit Smarthome-Controllern erfolgt über HTTP- und MQTT-Protokolle. Die Konfiguration der entsprechenden Server-/Broker-Daten erfolgt ebenfalls einfach in der Web-Oberfläche.

Falls später einmal ein Update der Firmware notwendig sein sollte, muss man dazu nicht erneut das Modul aus seinem jeweiligen Gerät ausbauen und verkabeln, denn Updates erfolgen übers WLAN.

	generic ESP8266 or ESP8285	1MB	None	1, 3, (14)	0 (button), 12 (Relay 10A@230VAC), 13 (LED)	on board mains AC to 5DVC + ? 3.3V Voltage Regulator Attention! The DC power is NOT galvanically decoupled from AC power!	90-250VAC	onboard PCB	88mm x 38mm x 23mm
	generic ESP8266 or ESP8285	1MB	None	1, 3, (14)	0 (button), 12 (Relay 10A/16A@230VAC), 13 (LED), 14 (AM2301)	on board mains AC to 5DVC + ? 3.3V Voltage Regulator Attention! The DC power is NOT galvanically decoupled from AC power!	90-250VAC	onboard PCB	88mm x 38mm x 23mm
	generic ESP8266 or ESP8285	1MB	None	1, 3, (14)	0 (button), 12 (Relay 10A@230VAC), 13 (LED)	on board mains AC to 5DVC + ? 3.3V Voltage Regulator	90-250VAC	onboard PCB	?
	generic ESP8285	1MB	None	2, (7, 8)	0, 9, 10, 14 (Buttons), 4, 5, 12, 15 (Relay 10A@230VAC), 13 (LED blue)	on board mains AC to 5DVC buck converter + 3.3V Voltage Regulator	90-250VAC	onboard PCB	145mm x 90mm x 41mm
	generic	1MB	None	-	0 (button), 12 (Relay 2A@230VAC), 13 (LED)	on board mains AC to 5DVC buck converter + 3.3V Voltage	90-250VAC	onboard PCB	86mm x 86mm x 37mm (EU)

7 ESPEasy trennt sauber zwischen kommerzieller Smarthome-Hardware und Selbstbau-Sensoren und -aktoren.

ESPHome

ESPHome arbeitet mit der Smarthome-Controller-Software *Home Assistant* ⁸ und OpenHAB zusammen, was auch problemlos funktioniert. Der Weg dahin ist allerdings etwas steinig.

ESPHome geht bei der Installation nämlich einen anderen, komplizierteren Weg. Das Firmware-Paket muss zunächst auf einem PC in einem Konsolenfenster installiert werden. Dazu ist Python notwendig.

Ebenfalls im Konsolenfenster erfolgt dann die Konfiguration der Firmware für das jeweilige Modul. Sie müssen zunächst eine Konfigurationsdatei für jeden Raum anlegen ⁹, also beispielsweise fürs Wohnzimmer mit

```
esphome livingroom.yaml wizard
```

Anschließend müssen Sie unter anderem den verwendeten Chip, die genaue Boardbezeichnung sowie die WLAN-Daten eingeben und erhalten als Belohnung eine Konfigurationsdatei, die Sie dann aber mit einem Editor erst vervollständigen müssen. Und zwar fehlen noch die Daten der angeschlossenen Sensoren/Aktoren sowie die Anschlüsse, an denen sie hängen. Erst, wenn Sie das alles nachgetragen haben, können Sie mit

```
esphome livingroom.yaml run
```

die Firmware kompilieren lassen. Dieser Vorgang kann je nach Computer bis zu zehn Minuten dauern. Danach wird die Firmware ins Modul übertragen. Das ist gegenüber den Web-Oberflächen der anderen Firmware-Versionen doch ein recht umständliches Verfahren.

Sobald das jedoch geschehen ist, steht das Modul in Home Assistant zur Verfügung.

Auf den Internet-Seiten von ESPHome herrscht in den Listen der benutzbaren Komponenten ein wenig Unordnung, da nicht



ESPHome

Quick search

Table of Contents

- ESPHome
- Guides
- Devices
- Core Components
- Sensor Components
- Binary Sensor Components
- Output Components
- Light Components
- Switch Components
- Fan Components
- Display Components
- Cover Components
- Text Sensor Components
- Climate Components
- Misc Components
- Additional Custom Components
- Cookbook

 SHT3X-D	 STS3X	 SGP30
 TCS34725	 Template Sensor	 Total Daily Energy
 TSL2561	 TX20	 Ultrasonic Sensor
 Uptime Sensor	 VL53Lox	 WiFi Signal Strength
 Xiaomi CGG1	 Xiaomi HHCCJCY01	 Xiaomi LYWSDo2
 Xiaomi LYWSDCGQ	 ZyAura	 Custom Sensor

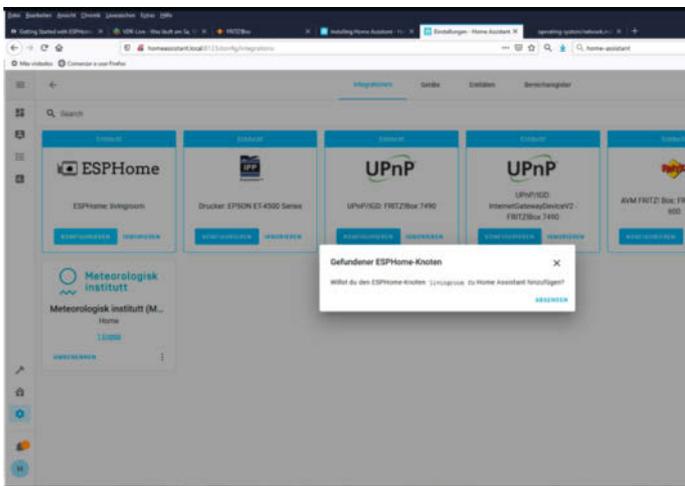
Looking for a sensor that outputs its values as an analog voltage? Have a look at the

¹⁰ Die gut gefüllte Liste der in ESPHome verwendbaren Komponenten

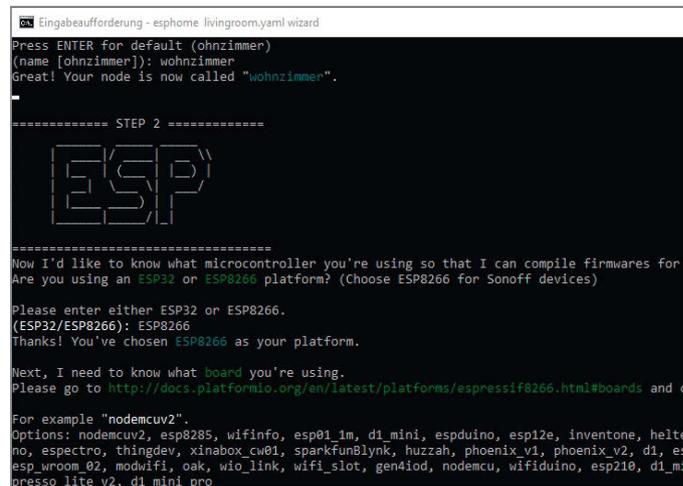
zwischen kommerziellen Smarthome-Produkten und Selbstbau-Modulen unterschieden wird ¹⁰. Daran gewöhnt man sich aber rasch. Das Angebot ist groß. Auch diese Firmware kann mit den Pearl-Steckdosen umgehen, mit denen von OBI jedoch nicht. Dafür kann es sowohl mit 433MHz als auch Infrarot senden und empfangen, auch SPI-Module benutzen und kommt sogar mit Bluetooth BLE (nur auf ESP32-Platinen enthalten) zurecht. Auch die ESP-Kamera darf benutzt werden. Die gängi-

gen Maker-Module für GPIO- und I²C-Anschluss sind ebenfalls verwendbar.

Wenn man sich in dieses System einmal eingearbeitet hat, ist es trotz der komplizierten Einrichtung durchaus sinnvoll, denn so bietet sich die Möglichkeit, vor dem Kompilieren noch Programmänderungen oder -erweiterungen vorzunehmen. Wer programmieren kann und ein künftig an neue Hardware anpassbares System möchte, sollte zu ESPHome greifen.



⁸ Der Smarthome-Controller Home Assistant läuft auch auf einem Raspberry Pi ab Version 3.



⁹ Sämtliche Einstellungen in ESPHome nehmen Sie in einem Konsolenfenster vor. Einfach, aber optisch nicht so ansprechend.

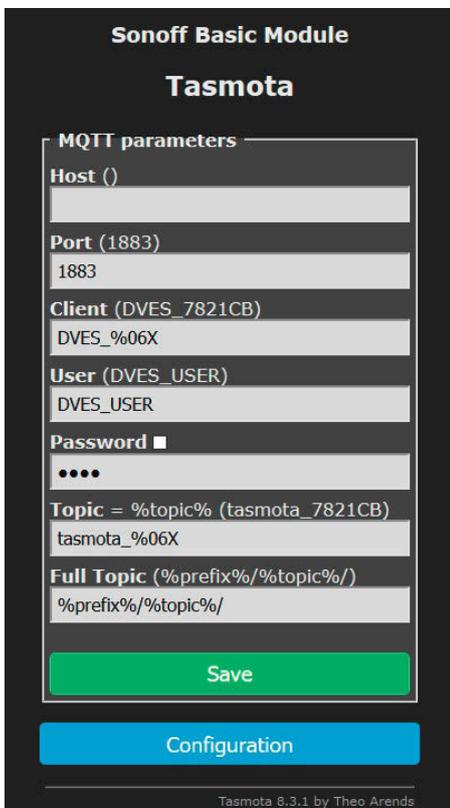
Tasmota

Tasmota ist, was die Zusammenarbeit mit Controllern angeht, deutlich flexibler als ESPHome. Es akzeptiert jeden MQTT-Broker **11**.

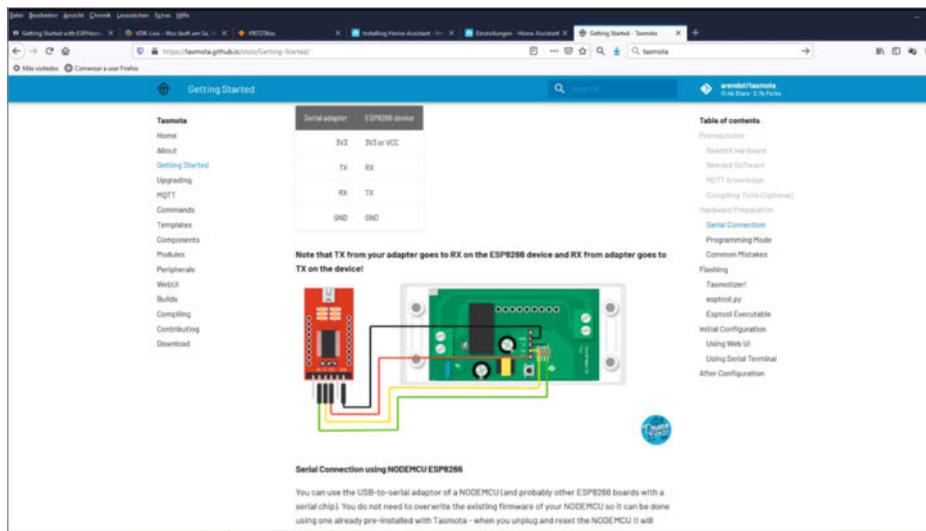
Die Installation ist wie bei ESPEasy sehr einfach: Sie müssen nur die Firmware-Datei tasmota.bin in den Flash-Speicher des Moduls schreiben, der Rest geschieht online per Web-Oberfläche. Sehr schön: Auf den Tasmota-Seiten im Netz erhält man auch Anleitungen, wie man ESP-Module ohne USB-Buchse mit einem Programmer oder Seriell-/USB-Wandler anschliessen muss **12**. Auch Tasmota stellt beim ersten Start einen Accesspoint bereit (ssid: tasmota_XXXXXXXXXX, Passwort: ohne) und kann nach dem Einloggen in dieses WLAN mit der IP 192.168.4.1 erreicht werden. Auch hier müssen zuerst die Daten des eigenen WLANs eingegeben werden. Danach wechselt Tasmota automatisch ins neue Netz, zeigt aber leider nicht die neue IP an und benutzt auch keinen lokalen Namen. Es bleibt also nur übrig, die IP am Router nachzuschauen.

Die Konfiguration ist dann aber einfach, die Weboberfläche ist sehr übersichtlich und sogar in einem Smartphone-tauglichen Format **13**.

Im Internet finden Sie die Listen mit den verwendbaren Komponenten, und zwar ge-



11 Die Web-Seite zur Konfiguration (hier die MQTT-Einstellungen) ist in Tasmota sehr übersichtlich und auch vom Smartphone aus problemlos bedienbar.

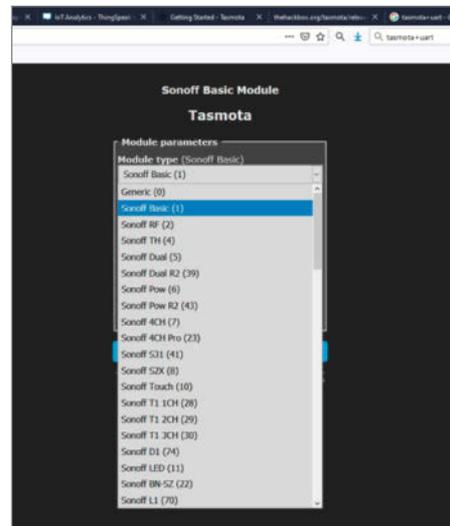


12 Tipps zum Anschluss USB-freier ESP-Module gibt es bei Tasmota im Netz.

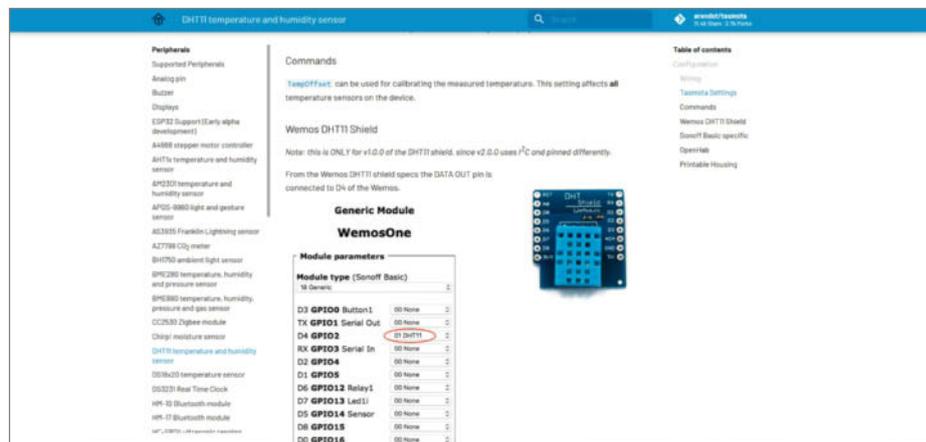
ordnet nach Typenbezeichnung. Erst nach Klicken auf solch eine Bezeichnung erhalten Sie dann aber detaillierte Informationen und in den meisten Fällen auch ein Bild der Hardware **14**.

Darunter sind nahezu alle gängigen Maker-Module für GPIO, I²C und SPI-Anschluss. Lediglich die ESP-Kamera haben wir vermisst. Bei den kommerziellen Komponenten hat es sogar die größte Auswahl aller vier Kandidaten dieses Artikels. Tasmota kann infrarot senden und empfangen, sodass auch der Integration von auf diese Weise ferngesteuerten Geräten ins Smarthome nichts im Wege steht. Dasselbe gilt für Funkfernbedienungen im 433 MHz-Bereich. Spätere Änderungen der Firmware kann Tasmota ebenfalls kabellos übers WLAN empfangen.

Alles in allem ist Tasmota durch seine detaillierten Anleitungen und den Funktionsumfang sehr gut auch für Anfänger geeignet.



13 Im Modulfenster wählen Sie etwa Hersteller und Typ der angeschlossenen Sensoren.



14 Auf den Internetseiten von Tasmota erhält man zu den meisten Sensoren und Aktoren Detailinformationen inklusive Bild.

ESPurna

Auch ESPurna kann mit den gängigen Open-Source-Smarthome-Controllern zusammenarbeiten, aber auch mit dem Google-Sprachassistenten.

Die Installation kann wie bei ESPEasy und Tasmota ebenfalls sehr einfach sein: Wenn es auf der Downloadseite eine geeignete Firmware-Datei für das verwendete Board und die daran angeschlossenen Sensoren/Aktoren gibt, genügt es, die Datei in den Flash-Speicher des ESP-Moduls zu schreiben.

Kompilieren erforderlich

Doch gibt es eben nur wenige solcher fertiger Dateien. In allen anderen Fällen müssen Sie den Quellcode herunterladen (als Arduino- oder platformIO-Projekt) und die Einstellungen in den Konfigurationsdateien selbst vornehmen. Diese Dateien sind aber recht gut kommentiert. Meist beschränkt sich die Tätigkeit darauf, die WLAN-Daten einzugeben und hinter den entsprechenden Namen eines Sensors/Aktor das *FALSE* auf *TRUE* zu ändern. Danach ist noch das Kompilieren und die Übertragung der Firmware ins Modul nötig. Läuft es dann aber wegen eines Fehler nicht, beginnt die Prozedur von vorn. Da zeigen die anderen Kandidaten dieses Artikels, dass es einfacher geht.

Ist diese Hürde aber genommen, besitzen die Module dann eine recht komfortable Web-Oberfläche, über die die weiteren Einstellungen, etwa für den MQTT-Broker, vorgenommen werden **15**.

Die Auswahl verwendbarer Komponenten, sei es kommerziell oder Selbstbau-Module, ist etwas geringer als bei den anderen dreien. SPI-Module sind nicht möglich, Bluetooth BLE wird ebenfalls nicht unterstützt. Auch der bei den anderen dreien mögliche *DeepSleep*-Zustand, der kräftig Strom spart, ist mit ESPurna nicht möglich. Das schränkt die Verwendung batteriebetriebener Geräte ein. Die Internet-Seiten listen die verwendbaren Teile übersichtlich und bebildert auf **16**.

Auch zur Konfiguration gibt es dort viele Tipps und Anleitungen, ein eigenes Wiki steht zur Verfügung.

ESPurna erscheint auf den ersten Blick abschreckend, insbesondere wegen der Art der Installation. Aber darin stecken enorme Entwicklungsmöglichkeiten: So kann man die Firmware auf relativ einfache Weise um die Unterstützung neuer Hardware erweitern, ein Einsetzen der entsprechenden Bibliothek zum Beispiel in der Arduino-IDE genügt da oft schon, um die notwendigen Steuerbefehle verfügbar zu machen. Damit ist aber die Grenze der hier eigentlich besprochenen fertigen Smarthome-Firmware erreicht.

ESPURNA-DCBC15

MQTT

Configure an MQTT broker in your network and you will be able to change the switch status via an MQTT message.

STATUS

GENERAL

DOMOTICZ

HASS

LED

MQTT

NTP

SCHEDULE

SWITCHES

THINGSPEAK

WIFI

ADMIN

DEBUG

Save

Reconnect

Reboot

© 2016-2019
Yoss Pérez
@yosperez
http://tinkerman.cat
ESPurna @ GitHub
GPLv3 license

MQTT

Configure an MQTT broker in your network and you will be able to change the switch status via an MQTT message.

Enable MQTT NO YES

MQTT Broker

MQTT Port

MQTT User

You can use the following placeholders: {hostname}, {mac}

MQTT Password

MQTT Client ID

If left empty the firmware will generate a client ID based on the serial number of the chip. You can use the following placeholders: {hostname}, {mac}

MQTT QoS

MQTT Retain NO YES

MQTT Keep Alive

MQTT Root Topic

This is the root topic for this device. The {hostname} and {mac} placeholders will be replaced by the device hostname and MAC address.

- <root>/relay/#set Send a 0 or a 1 as a payload to this topic to switch it on or off. You can also send a 2 to toggle its current state. Replace # with the switch ID (starting from 0). If the board has only one switch it will be 0.

- <root>/status The device will report a 1 to this topic every few minutes. Upon MQTT disconnecting this will be set to 0.

- Other values reported (depending on the build) are: firmware and version, hostname, IP, MAC.

MQTT User

MQTT Password

MQTT Client ID

MQTT QoS

MQTT Retain NO YES

MQTT Keep Alive

MQTT Root Topic

This is the root topic for this device. The {hostname} and {mac} placeholders will be replaced by the device hostname and MAC address.

- <root>/relay/#set Send a 0 or a 1 as a payload to this topic to switch it on or off. You can also send a 2 to toggle its current state. Replace # with the switch ID (starting from 0). If the board has only one switch it will be 0.

- <root>/status The device will report a 1 to this topic every few minutes. Upon MQTT disconnecting this will be set to 0.

- Other values reported (depending on the build) are: firmware and version, hostname, IP, MAC.

15 ESPurna wird zum Teil über Konfigurationsdateien vor dem Kompilieren und zum Teil über die Web-Oberfläche konfiguriert.

The screenshot shows the ESPurna website's hardware compatibility page. It features a grid of nine product images with their names below them:

- Lohas 9W
- Xiaomi Smart Desk Lamp
- iWoole LED Table Lamp
- Itead Sonoff LED
- Itead BN-SZ01
- Lombex LUX Nova 2 (white and color)
- Arilux AL-LC01 (RGB)
- Arilux AL-LC02 (RGBW)
- Arilux AL-LC06 (RGBWWCW)

16 ESPurna zeigt auf seiner Internetseite, mit welcher Hardware es zusammenarbeiten kann.

Smarthome-Firmware: Vergleichstabelle

Diese Tabelle zeigt Ihnen die wichtigsten Informationen zu den vier Smarthome-Firmware-Versionen für ESP-Module. Die aufgeführten kommerziellen Komponenten stellen nur ein Auswahl dar. Auf den Internet-Seiten der

Firmware erfahren Sie, welche (zum Teil in Deutschland nicht erhältlichen) Produkte noch unterstützt werden.

Das Flashen solcher fertigen Smarthome-Komponenten sollte insbesondere bei mit

Netzstrom betriebenen Teilen nur von Fachleuten ausgeführt werden, da die Geräte dazu geöffnet werden müssen. Zuvor bitte die Original-Firmware wie auf Seite 42 beschrieben sichern.

Smarthome-Firmware für ESP-Module

Name der Firmware	ESPEasy	ESPHome	Tasmota	ESPurna
verfügbar als	Sourcecode/Binärdatei	Binärdatei	Binärdatei	Sourcecode für Arduino-IDE, PlatformIO/Binärdatei für einzelne Sensoren/Aktoren
geeignete ESP-Module	ESP8266/ESP32	ESP8266/ESP32	ESP8266/ESP32	ESP8266/ESP8285
Anwendungsgebiet	IoT	IoT	IoT	IoT
Konfiguration über	Web-Oberfläche	Wizards im PC-Konsolenfenster	Web-Oberfläche	Konfigurationsdatei in der Programmierumgebung
Protokolle zur Zusammenarbeit mit Smarthome-Controllern	Domoticz (MQTT + HTTP), OpenHAB (MQTT + HTTP), PiDome MQTT, FHEM HTTP, UDP, HTTP	OpenHAB, Home Assistant	all MQTT-Broker	Domoticz, Homeassistant, Thingspeak, Google Assistant
MQTT-fähig	ja	ja	ja	ja
Verwendbare Sensoren	sehr viele (GPIO/I ² C/UART)	sehr viele (GPIO/I ² C/UART/SPI)	sehr viele (GPIO/I ² C/UART//SPI)	sehr viele (GPIO/I ² C, UART)
Installierbar auf diesen kommerziellen Produkte				
Genio	ja	ja	-	-
Sonoff	ja	ja	ja	ja
Zemismart	ja	ja	-	ja
Teckin	-	ja	ja	ja
Huafan	ja	ja	ja	-
AiLight	ja	ja	ja	-
Supla Espablo	-	-	ja	-
Magic Home	ja	ja	ja	ja
Arilux	-	ja	ja	ja
Xiaomi Philips	-	ja	ja	ja
Neo Coolcam	-	-	ja	-
Obi Socket 2	ja	-	ja	ja
Luminea (Pearl)	ja	ja	ja	ja
Hue	ja	-	-	-
Hama	-	-	-	ja
Blitzwolf	ja	-	ja	ja
Renkforce	ja	-	-	-
433MHz-Schalter	ja	ja	ja	ja
Bluetooth-BLE	-	ja	ja	-
Infrarot-Fernbedienung				
senden	-	ja	ja	-
empfangen	ja	ja	ja	ja
Datenbus-Ansteuerung				
I2C	ja	ja	ja	ja
SPI	-	ja	ja	-
UART	ja	ja	ja	ja
sonstiges				
Stromspar-Modus (Deep Sleep)	ja	ja	ja	-
Over the Air Updates	ja	ja	ja	ja
ESP32-Kamera	-	ja	-	-
Lizenz	GNU	MIT/GNU	GNU	GNU

Backups der Original-Firmware

Bevor Sie ein ESP-Modul mit einer neuen Firmware ausstatten, ist es empfehlenswert, die darin enthaltene FW-Version zu sichern. Dann kann man sie später wieder zurückschreiben, auch wenn man deren Quellcode nicht (mehr) besitzt. Insbesondere vor der Modifizierung der Firmware kommerzieller Geräte (WLAN-Schalter, Steckdosen, Lampen usw.) ist das ratsam, um sie später wieder in den Ursprungszustand zurückversetzen zu können, falls etwas nicht klappt.

Das passende Software-Tool heißt *esptool*. Sie installieren es unter Windows im Eingabeaufforderungsfenster (Start durch Eingabe von CMD in der Suchezeile). Wechseln Sie darin in das Python-Verzeichnis mit dem Befehl:

```
cd c:\Python27\Scripts
```

Der Installationsbefehl lautet

```
pip install esptool
```

Falls es dabei unter Windows zu Fehlern kommen sollte, kann das an einer unsauberen (oder fehlenden) Python-Installation liegen. In diesem Fall am besten Python 2.7.X deinstallieren und die Version 2.7.18 neu installieren

(Download-Adresse siehe Kurzinfor-Link). Mit Python 3 gibt es auf Windows-Systemen manchmal Probleme, wenn zuvor bereits Python 2.x installiert wurde. Das Modul muss entsprechend seiner Bauweise mit einem USB-Kabel oder einem geeigneten Programmier an den Computer angeschlossen und der passende Treiber installiert sein.

Werte des ESP-Moduls ermitteln

Bevor Sie mit dem Auslesen der Firmware beginnen, sollte Sie zunächst einmal die wichtigsten Werte des ESP-Moduls ermitteln. Das geschieht mit dem Befehl

```
esptool.py flash_id
```

Die Ausgabe sieht dann etwa so aus wie in Bild 17.

Notieren Sie sich diese Werte. Bei kommerziellen Geräte kann es vorkommen, dass die Original-Firmware mit der MAC-Adresse des Geräts verheiratet ist, also nur mit diesem einen Gerät läuft. Selbst bei gleichartigen Modellen müssen Sie deshalb von jedem einzel-

nen Gerät ein Backup ziehen und entsprechend kennzeichnen.

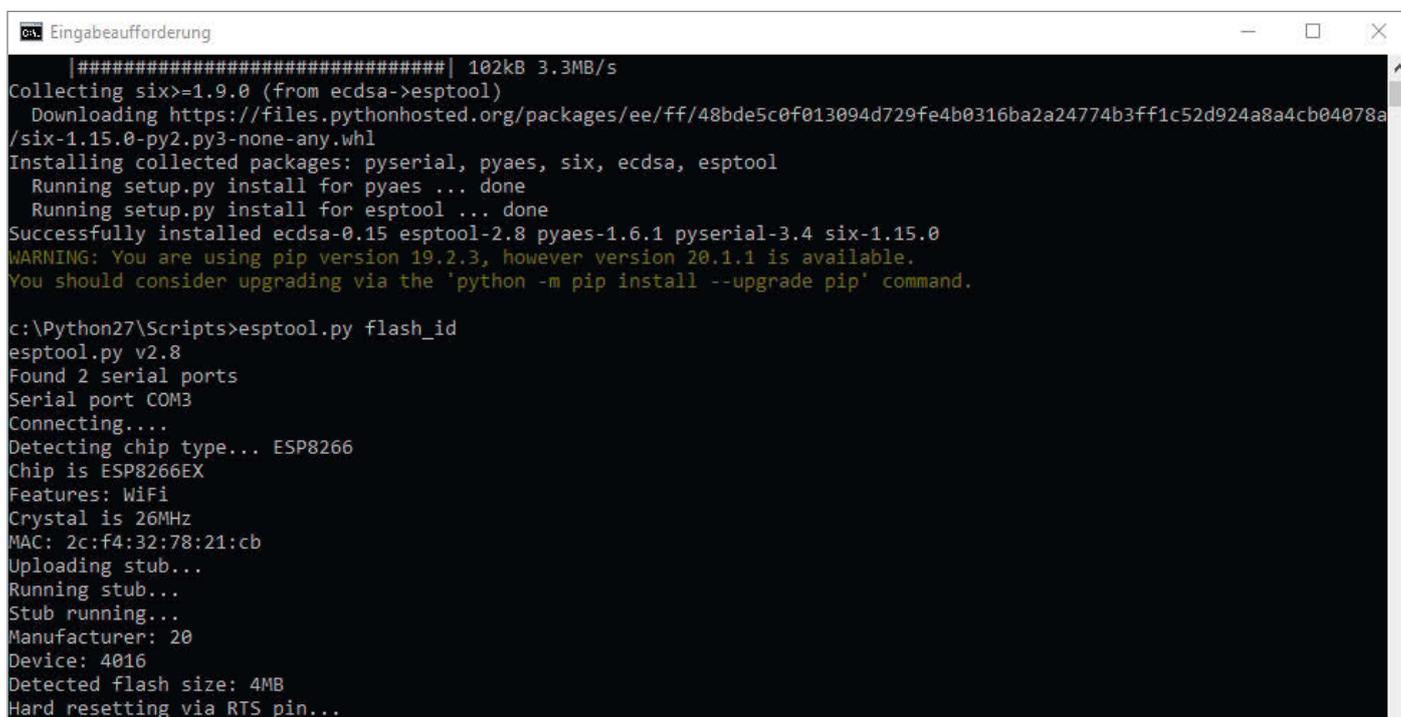
Firmware auslesen

Das Auslesen der Firmware geschieht mit demselben Programm 18. Das X hinter COM müssen Sie durch die zuvor ermittelte Nummer des COM-Ports ersetzen. Der Eintrag 0x400000 gilt für 4MB Speichergröße, bei 2 MB lautet er 0x200000, bei 1 MB 0x100000. Der letzte Parameter des Befehls ist der Dateiname, unter dem die Firmware gespeichert wird. Setzen Sie hier statt MAC die Mac-Adresse des Moduls und statt 4MB die Speichergröße ein.

Das Auslesen ist recht langsam. Glücklicherweise wird Ihnen der Fortschritt dabei als Prozentwert angezeigt 19.

Firmware zurückschreiben

Der Befehl zum Zurückspielen der Firmware auf ein Modul ist dem zum Auslesen sehr ähnlich 20. Hier müssen Sie natürlich ebenfalls den richtigen COM-Port angeben. Ansonsten



17 Dieses ESP-Modul ist vom Typ ESP8266EX, sitzt an COM3 und besitzt 4MB Flash-Speicher sowie die MAC-Adresse 2c:f4:32:78:21:cb.

18 Befehl zum Auslesen der Firmware

```
01 esptool.py --port COMX --baud 115200 read_flash 0x00000 0x400000 Firmware_MAC_4MB.bin
```

```
esptool.py v2.8
Serial port COM3
Connecting....
Detecting chip type... ESP8266
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 2c:f4:32:78:21:cb
Uploading stub...
Running stub...
Stub running...
442368 (10 %)
```

```
esptool.py v2.8
Serial port COM3
Connecting....
Detecting chip type... ESP8266
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 2c:f4:32:78:21:cb
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 4MB
Compressed 4194304 bytes to 340853...
Writing at 0x00028000... (52 %)
```

19 Nur Geduld: So zwischen 2 und 10 Minuten kann das Auslesen dauern.

21 Das Zurückschreiben erfolgt deutlich schneller als das Auslesen.

sind nur die Startadresse im Flash-Speicher (0x00000) sowie der Dateiname der Firmware nötig. Die Endadresse ergibt sich automatisch aus der Größe der Firmware-Datei. Auch der Fortschritt beim Zurückschreiben wird per Prozentzahl angezeigt 21. —hgb

20 Befehl zum Zurückschreiben der Firmware

```
01 esptool.py --p COMX read_flash 0x00000 Firmware_MAC_4MB.bin
```

ANZEIGE

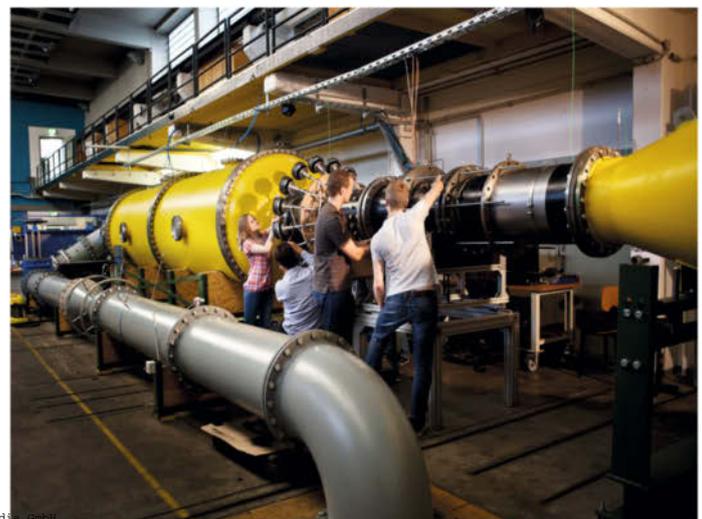
Sichere Dir einen Platz in einem der **spannendsten** und **vielseitigsten technischen Studiengänge** mit hervorragenden Jobaussichten. Lebe in einer der grünen Städte Deutschlands.

Bewirb Dich jetzt für den Bachelor **Maschinenbau** oder **Produktion & Logistik!**

Bewerbung: Bis zum 30. September 2020

Wo: Leibniz Universität Hannover

www.uni-hannover.de



Leuchtendes LED-Herz

Displays mit indirekt beleuchteten Kanten machen alle Projekte etwas magischer – und wir zeigen eine neue Technik, mit der es ganz einfach geht.

von Debra Ansell (Übersetzung: Niq Oltman)



Bilder: Debra Ansell

Um den Rand eines Displays zu beleuchten, verwendet man üblicherweise eine gravierte Acrylglasplatte und lässt eine ihrer Kanten in ein Gehäuse ein. In das Gehäuse platziert man eine Lichtquelle. Das Licht streut durch das Acrylglas und lässt die Gravur aufleuchten. Bemalt man die Kanten zusätzlich farbig, werden die Farben innen sichtbar.

Für dieses Projekt habe ich eine neue Technik ausprobiert: Ich habe das Acrylglas in puzzleförmige Teile geschnitten und die Lichtquelle zwischen den Puzzleteilen angebracht. Auf diese Weise scheinen die resultierenden Formen von innen heraus zu leuchten. Beleuchtet man jedes Teil individuell mit einer anderen Farbe, entstehen harte Übergänge zwischen den Lichtfeldern. Mit diesem Effekt kann man sehr viel farbenfrohere Lichtinstallationen gestalten. Wer einen Lasercutter zur Verfügung hat, kann das leuchtende Herz für etwa 30 Euro bauen.

In den Links in der Kurzinformatio befinden sich zwei weitere Designs für die Gravur der Herzfläche. Die Vektordatei kann man zudem leicht den eigenen Wünschen anpassen.

Basis aus Holz bauen

Zuerst baut man eine Box aus Holz, in der man später die Elektronik verstecken kann. Die Holzteile für die Basis schneidet man mit einem Lasercutter aus 3mm dickem Holz zu 1. Die Schnittvorlage im Vektor-Format ist über den Link in der Kurzinformatio zu finden. Dann wird die Box zusammengesteckt und verklebt: Zunächst streicht man Holzleim auf die unteren Kanten aller Seitenteile. Dann kann man die Teile in die Bodenplatte pressen. Dort sind bereits kleine Aussparungen ausgeschnitten, in die die kleinen Holzzähnen genau passen 2. Das Oberteil nicht festleimen. Es bleibt lose, damit man später den Mikrocontroller in die Box stecken kann.

Acrylglas mit dem Lasercutter zuschneiden

Vorweg: Wir empfehlen gegossenes Acryl, da es schöner leuchten kann, als extrudiertes

Kurzinformatio

- » Herzform aus Acrylglas schneiden
- » LED-Stripes verlöten
- » LED-Farben mit Adafruit Trinket M0 programmieren

Checkliste



Zeitaufwand:
1 bis 2 Stunden



Kosten:
30 bis 60 Euro



Programmieren:
CircuitPython



Löten:
einfache Lötarbeiten



Maschinen:
Lasercutter

Material

- » 6mm dicke Acrylglasplatte transparent, mind. 15cm × 20cm
- » 1,5mm dicke Acrylglasplatte transparent, mind. 15cm × 25cm
- » 3mm dickes Sperrholz mindestens 15cm × 15cm
- » 4–6mm breiter RGB-LED-Strip Typ WS2812B oder SK6812 3535, 60 LEDs pro Meter, z.B. Adafruit NeoPixel-Strips (Art.-Nr. 2959)
- » Adafruit Trinket M0
- » Maschinenschraube M2 × 10mm mit Mutter
- » 2 Maschinenschrauben M3 × 20mm mit Mutter
- » Kupfer- oder Polyesterfolie-Klebeband 5mm breit
- » Holzleim
- » Schrumpfschlauch

Mehr zum Thema

- » Florian Schäffer, LED-Nixies, Make 4/18, S. 20
- » Guido Körber, Plexiglas sandstrahlen, Make 1/19, S. 112

Alles zum Artikel im Web unter make-magazin.de/xh3f

Werkzeug

- » Lasercutter oder Online-Laserschnitt-Service
- » Seitenschneider und Abisolierzange
- » LötKolben und Lötzinn
- » Heißluftpistole
- » Filzstifte
- » Schraubenzieher

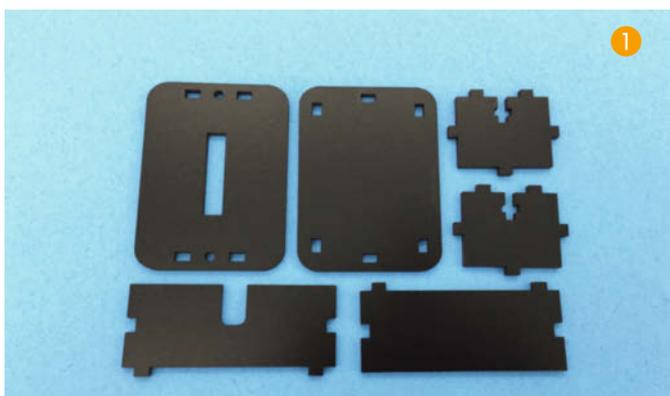
Acryl. Das leuchtende Herz besteht aus drei Schichten Acrylglas. Die äußeren Schichten in 1,5mm Dicke umschließen die 6mm dicke innere Schicht und fixieren so den LED-Streifen. Das Herz wird zusammengeschichtet auf einer Grundplatte aus 6mm dickem Acrylglas fixiert.

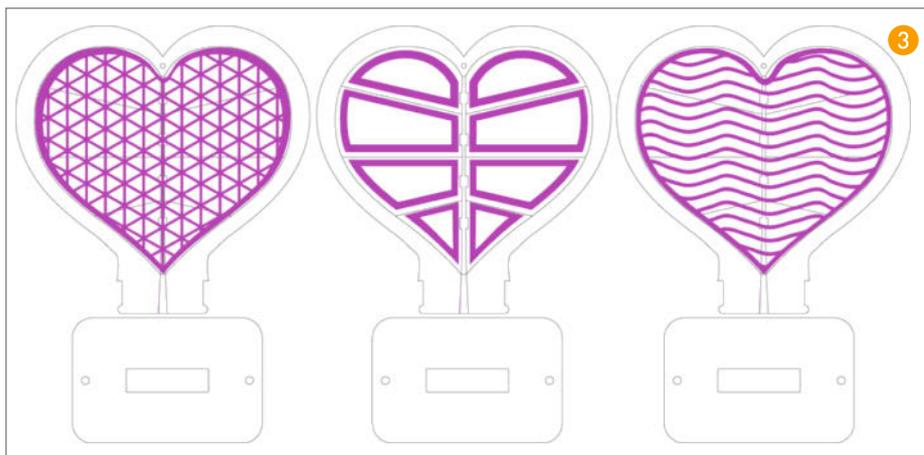
Mit dem Lasercutter schneidet man die beiden Motive für die drei Schichten zu. Bei der 6mm dicken, inneren Platte sind in der Vektordatei rosa Linien zu sehen 3. Diese sind zum Gravieren gedacht. Die schwarzen Konturen sind die Schnittlinien. In der Vorlage

finden sich verschiedene Gravurmuster zur freien Auswahl – alternativ kann man natürlich auch ein eigenes Muster entwerfen. Anschließend kann man die Schutzfolie vom Acryl abziehen.

LED-Streifen vorbereiten

Als nächstes schneidet man ein Stück LED-Strip so zu, dass sich auf ihm acht LEDs befinden. Drei 10cm lange Stücke weißen Draht zuschneiden, je ein Ende abisolieren, und auf





Trinket-GND und LED-Signal auf Trinket-Pin D1. Wenn der Strip leuchtet, kann es weitergehen **6**.

Kanten der Acrylglas-elemente abkleben

Da die Acrylglas-Puzzleteile so klein sind, kann das Licht durch mehrere auf einmal scheinen. Um klar definierte Leucht-Abschnitte mit harten Kanten zu bekommen, muss man die Außenkanten mit reflektierendem Klebeband abkleben. Dazu eignet sich 5mm breites Kupfer- oder Mylar-Klebeband. Die gravierten Teile dann an allen Kanten mit dem reflektierenden Klebeband abkleben **7**. Die Kante mit der Aussparung für die LED muss man dabei frei lassen.

Das Herz zusammenbauen

Eines der 1,5mm dicken Acrylglassherzen flach auf den Tisch legen. Dann eine 10mm M2-Schraube so einsetzen, dass sie senkrecht aus dem kleinen Loch an der Oberseite ragt. Dann setzt man die innere Schicht zusammen. Dazu legt man die Herzkontur aus 6mm dickem Acrylglas so auf die äußere Schicht, dass man die Schraube durch das Loch stecken kann **8**. Dann legt man die gravierten Puzzleteile im Herzrahmen zusammen, so dass alle gravierten Seiten nach oben zeigen. Wenn man die Teile ordentlich in den Rahmen gelegt hat, bildet sich in der Mitte ein Spalt, in den der LED-Strip passt.

LED-Streifen einsetzen

Den gefalteten LED-Strip vorsichtig in den Spalt stecken, so dass die LEDs genau in den Aussparungen sitzen **9**. Jetzt die letzte 1,5mm dicke Schicht Acrylglas auflegen, um die losen Innenteile zu fixieren. Dann eine M2-Mutter auf die Schraube drehen, damit nichts mehr verrutschen kann.

Wenn alles richtig zusammengesetzt ist, passen die Verlängerungen am unteren Ende



die drei Lötflächen an die Dateneingangsseite des LED-Strips löten. Es geht leichter, wenn man viel Flussmittel auf die Kontakte gibt. Der weiße Draht wird hinter dem klaren Acrylglas nahezu unsichtbar. Das ist zwar ästhetisch ansprechend, allerdings kann man die Versorgungs-, Masse- und Signalleitungen jetzt nicht mehr so gut unterscheiden. Daher markiere ich die Drähte mit verschiedenfarbigen Filzstiften **4**. Zur Zugentlastung decke ich die Lötverbindungen mit etwas Schrumpfschlauch ab.

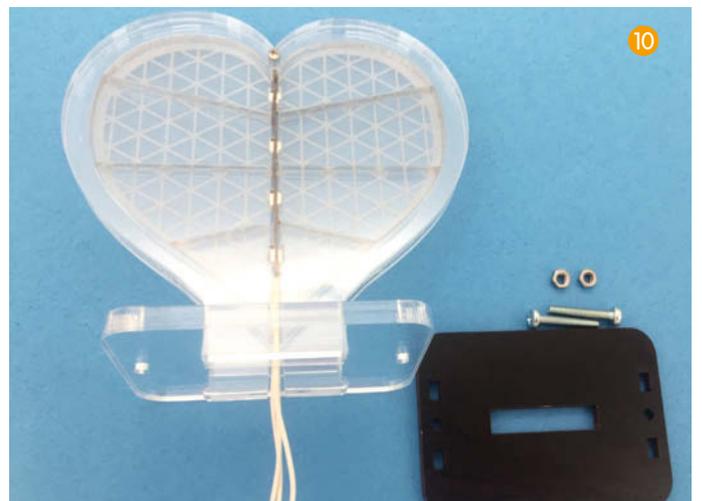
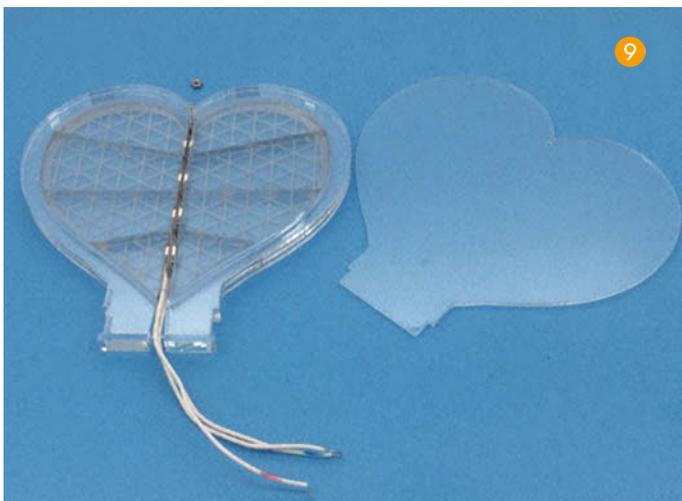
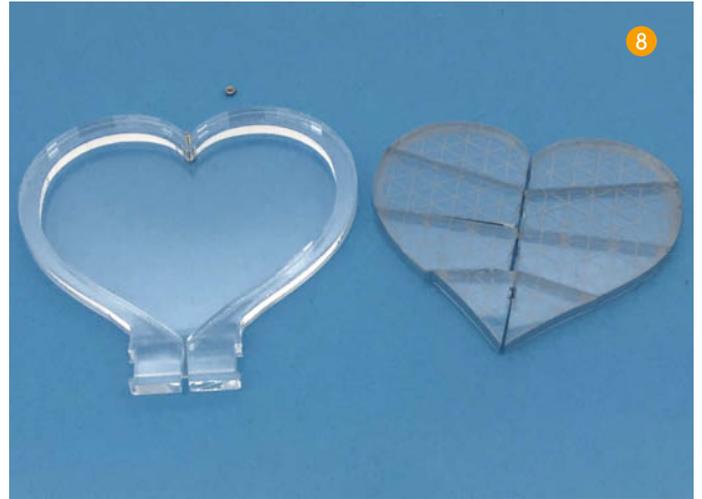
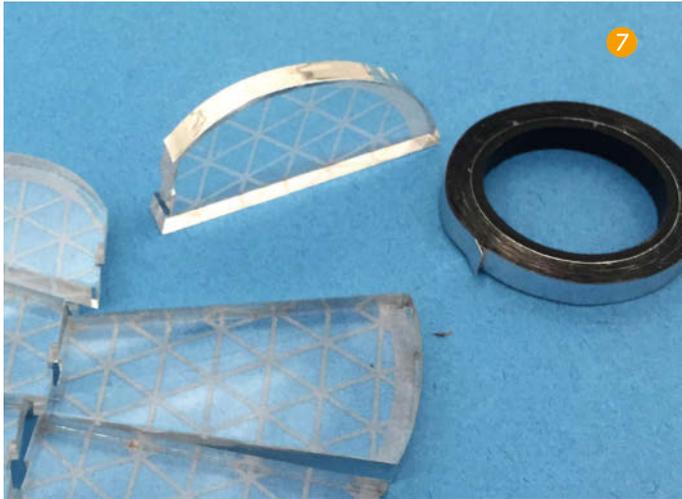
Jetzt biegt man den LED-Strip in der Mitte um, so dass die LEDs beider Hälften Rücken

an Rücken liegen. Wenn sie richtig ausgerichtet sind, kann man die Schutzfolie von der Klebefläche abziehen und die beiden Hälften fest zusammenkleben **5**.

ACHTUNG: Den LED-Strip nicht flach knicken, nur biegen. Es macht nichts, wenn an der Biegung eine kleine Schlaufe entsteht.

Jetzt sollte man sicherstellen, dass der LED-Strip leuchtet. Dafür die freien Enden der Drähte abisolieren und mit dem Mikrocontroller verbinden. Beim Trinket M0 kann man dafür die Drähte einfach provisorisch in die zugehörigen Pins stecken: Die LED-Versorgung (Power) auf Trinket-3V, LED-Masse (GND) auf





der Herzen bündig aufeinander. Die Kabel und die Acrylglaschichten schiebt man vorsichtig in das Loch in der Grundplatte **10**. Der mittlere Steg ist etwas breiter und hat zum Ausgleich einen Spalt in der Mitte; um ihn einzupassen, muss man seine Kanten leicht zusammendrücken. Nicht mit Gewalt – einfach die Grundplatte leicht hin- und her bewegen, bis das Herz einrastet.

Jetzt vorsichtig die Kabel und den mittleren Steg in den Deckel der Holzbox schieben. Hier muss der Steg wieder sanft zusammengedrückt werden, damit er in die Aussparung passt. Die 20mm M3-Schrauben in die entsprechenden Öffnungen der Basis setzen, aber noch nicht mit den Muttern verschrauben **11**.

Mikrocontroller anschließen

Jetzt ist es Zeit, die Kabel vom LED-Strip an den Mikrocontroller Trinket M0 zu löten. Ein Mikro-USB-Kabel für die Stromversorgung und Programmierung am Trinket anbringen. Den Mikrocontroller so in das Holzgehäuse einsetzen, dass das USB-Kabel aus dem Loch

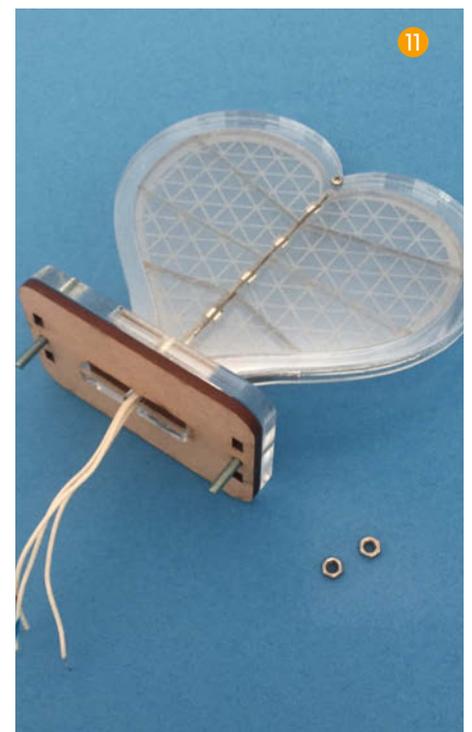
auf der Rückseite herausgeführt werden kann **12**.

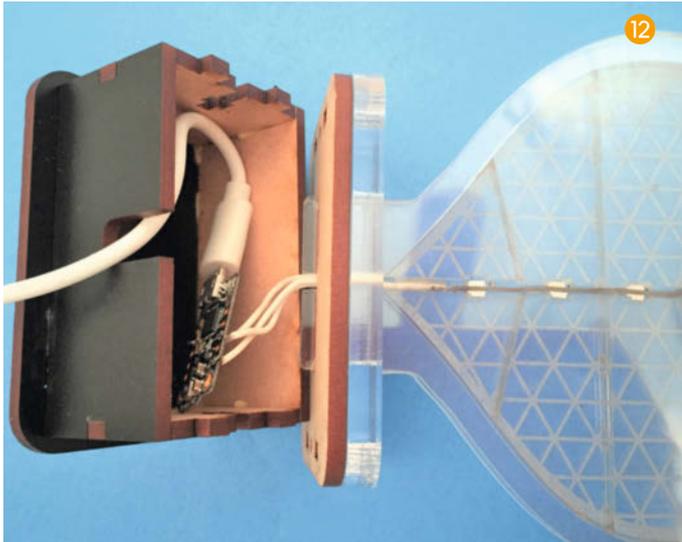
Das Gehäuse verschließen

Jetzt kann man die M3-Muttern auf die 20mm-Schrauben aufsetzen und den Deckel auf der Basis platzieren, so dass die Muttern genau in die Einkerbungen passen **13**. Zum Abschluss zieht man dann die Schrauben gut fest, um den Acrylglas-Aufbau und den Deckel bündig auf der Holzbox zu fixieren.

Mikrocontroller programmieren

Es werde Licht! Mikrocontroller wie der Trinket M0 unterstützen CircuitPython – damit lassen sich schön bunte und dynamische LED-Muster in nur wenigen Codezeilen realisieren. Um schnell zum Ziel zu kommen, habe ich einfach den NeoPixel-Code für CircuitPython von Adafruit angepasst. Der Link zum Download befindet sich in der Kurzinfor. Nur eine Codezeile muss man dazu anpassen, da der Bei-





spielcode für eine Reihe aus acht LEDs geschrieben wurde. In Zeile 6 `pixel_pin = board.A1` muss der LED-Signal-Pin auf `board.D1` geändert werden: `pixel_pin = board.D1`.

Das innere Leuchten

Sobald der Code auf den Mikrocontroller übertragen wurde, sollte das Acryl-Herz aufleuchten und Lichtfolgen aus einzelnen Farben und Regenbogenmustern darstellen.

Wenn man sich ein wenig in die NeoPixel-Lernanleitung für CircuitPython einliest, kann man auch bald ganz eigenen Lichtmuster programmieren, neue Leuchtabschnitte festlegen und sich so ein ganz individuelles Herz erschaffen.

Eigene Anpassungen

Mit dieser Methode kann man die unterschiedlichsten Formen und Motive indirekt in Regenbogenfarben beleuchten. Ich habe nach

diesem Muster zum Beispiel auch einen festlichen Weihnachtsbaum gebaut, dessen gravierte Ornamente in verschiedenen Farben leuchten. Genauso ist auch der farbenfrohen Schmetterling entstanden ¹⁴.

Solange die Aussparungen für die LEDs an der gleichen Position bleiben, kannst du dieses Design nach Belieben anpassen. Wie wäre es zum Beispiel mit Weihnachtsdekorationen, Geschenken oder Trophäen – oder was immer dir gefällt. —*rehu*



Neue Cyberangriffe – Wie können Unternehmen sich schützen – Kritische Infrastrukturen & Industrie 4.0 im Fokus

27. August 2020, 9 – 16 Uhr

Neben der Notwendigkeit von Cyberabwehrmaßnahmen werden auf dieser Online-Konferenz auch verschiedene Konzepte und Vorgehensweisen zur Sicherung der unternehmensinternen IT- und Prozessnetze dargelegt.

AUSZUG AUS DEM PROGRAMM:

- // Schutz vor Cyberangriffen im Energienetz – was ist erforderlich?
Dr. Tobias Pletzer, Schleswig-Holstein Netz AG
- // Was tun, wenn's brennt? Anatomie einer Datenschutzkatastrophe
Joerg Heidrich, Heise Medien
- // Blockchain sicher gestalten – die Perspektive des BSI
Christian Berghoff, BS
- // Der Mythos vom geschlossenen Netz
Christian Schlehuber, Deutsche Bahn AG
- // Lösungsansätze zum Aufbau resilienter
Sicherheitsarchitekturen im Prozessnetz-Umfeld
Martin Baumgartner, Fortinet

Preis: 159,00 Euro

In Kooperation mit



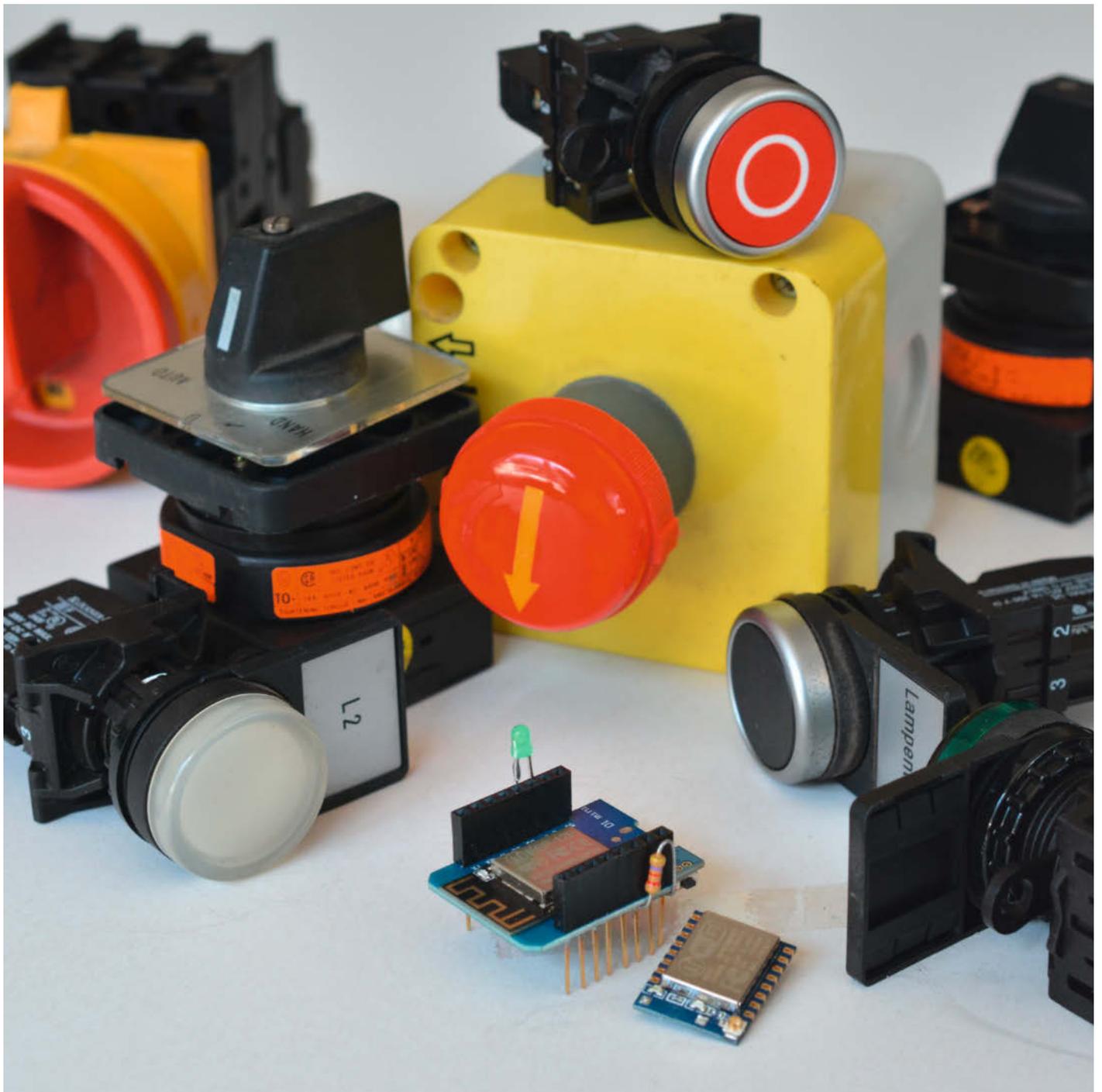
Partner



Schalter zur Welt

Ein Druck auf unseren IoT-Button und es wird eine voreingestellte Nachricht an die Familie gesendet – zum Beispiel, dass man zu Hause eingetroffen ist. Vielleicht koppeln wir den Button aber auch mit einem Smart-Home-System und fahren aus der Ferne die Rollläden herunter: Mit einem WLAN-Button sind sowohl ernsthafte als auch spielerische Anwendungen denkbar.

von Veikko Krypczyk



Eine solche Anwendung ist unser IoT-Button, ein physischer – gern auch etwas größerer und gut sichtbarer – Taster, der drahtlos mit dem Internet verbunden ist. Eine Betätigung wird registriert und löst dann eine vorbestimmte Aktion aus. Die möglichen Anwendungen sind vielfältig: Wir können mit dem Druck auf einen solchen Button etwas starten oder stoppen, vielleicht jemanden anrufen, etwas zählen, einen Notruf auslösen, eine Nachricht an einen bestimmten Empfängerkreis versenden oder einen Tweet absetzen und die Welt damit über den Eintritt eines spannenden Ereignisses informieren.

Sicher haben Sie noch viel bessere Ideen, welches Feuerwerk an Aktionen mit Druck auf den IoT-Knopf ausgelöst werden können. In diesem Beitrag wollen wir zeigen, wie wir einen solchen programmierbaren Button selbst bauen – vor allem aber, welche Cloud-Dienste man geschickterweise kombiniert, um mit möglichst wenig Arbeit davanzukommen.

Funktion im Überblick

Es handelt sich um einen physischen Druckknopf, der über eine minimale Hardware drahtlos mit dem Internet (über ein bestehendes WLAN) verbunden ist. Mit Druck auf dem Button wird ein Signal (Message) versendet. Damit man möglichst flexibel bleibt, welche Aktionen mit dem magischen Knopf ausgelöst werden soll, wird eine vermittelnde Zwischenschicht verwendet, ein cloudbasierter spezialisierter Dienst, der die eingehenden Signale von der IoT-Hardware nach eigenen Wünschen an die Zieldienste weiterleitet. So kommen wir ohne die zeitaufwändige und bei Anfängern auch fehlerträchtige Installation eines eigenen Server-Dienstes wie *ioBroker* aus **1**.

Dazu eignet sich zum Beispiel der Dienst *ThingSpeak* (alle Downloads zum Artikel über den Link in der Kurzinformatio). Mit Hilfe von *ThingSpeak* können Daten aus unterschiedlichen IoT-Anwendungen (Devices) entgegengenommen, gespeichert, ausgewertet und weitergeleitet werden. Für eine private oder eingeschränkte kommerzielle Nutzung ist der Dienst kostenfrei. Dabei können die Aktionen weitgehend durch Konfiguration festgelegt werden, das heißt, es ist keine Programmierung notwendig. Für eine Vielzahl von Diensten stehen einzubindende Plug-ins bereit.

Die IoT-Zwischenschicht kommuniziert ihrerseits wiederum über das Internet mit dem Zielservice. Dabei ist man flexibel und es richtet sich nach der IoT-Anwendung. Es kann sich beispielsweise um einen Internet-Service wie Twitter oder um eine Smart-Home-Anwendung handeln. Das Spektrum ist weitreichend und kann an die individuellen Anforderungen angepasst werden. Neben spielerischen Anwendungen und privaten

Kurzinformatio

- » Selbstgebauter IoT-Button mit ESP8266-Modul
- » Programmieren mit Arduino-IDE
- » Datenaustausch mit REST-API über HTTP
- » Geeignete Server-Dienste auswählen und konfigurieren

Checkliste

-  **Zeitaufwand:**
2 bis 3 Stunden
-  **Kosten:**
10 Euro
-  **Programmieren:**
Arduino-IDE bedienen, C++-Kenntnisse hilfreich
-  **Elektronik:**
Taster anschließen, Stromversorgung

Material

- » ESP8266-Modul z.B. NodeMCU oder Wemos D1
- » Taster in beliebiger Ausführung
- » Micro-USB-Kabel zur Programmierung
- » Batterie oder Akku optional

Mehr zum Thema

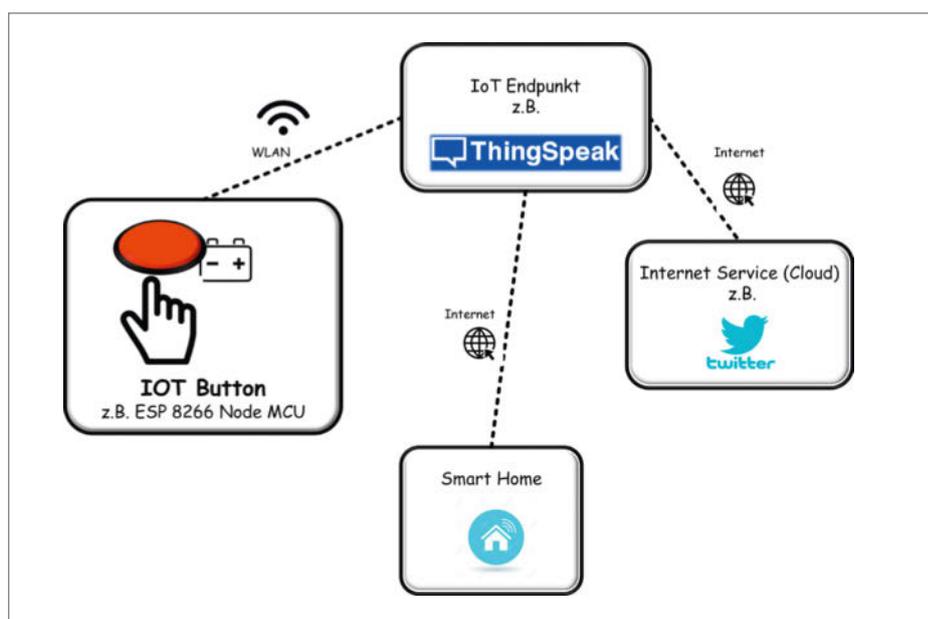
- » Carsten Meyer, Hardware steuern mit TouchOSC, Make 3/20 S. 94
- » Pit Noack, Smartphone als Fernsteuerung, Make 4/18, S. 68

↓ Alles zum Artikel im Web unter make-magazin.de/xu3s

Services sind auch eine Vielzahl von kommerziellen und industriellen Anwendungen denkbar.

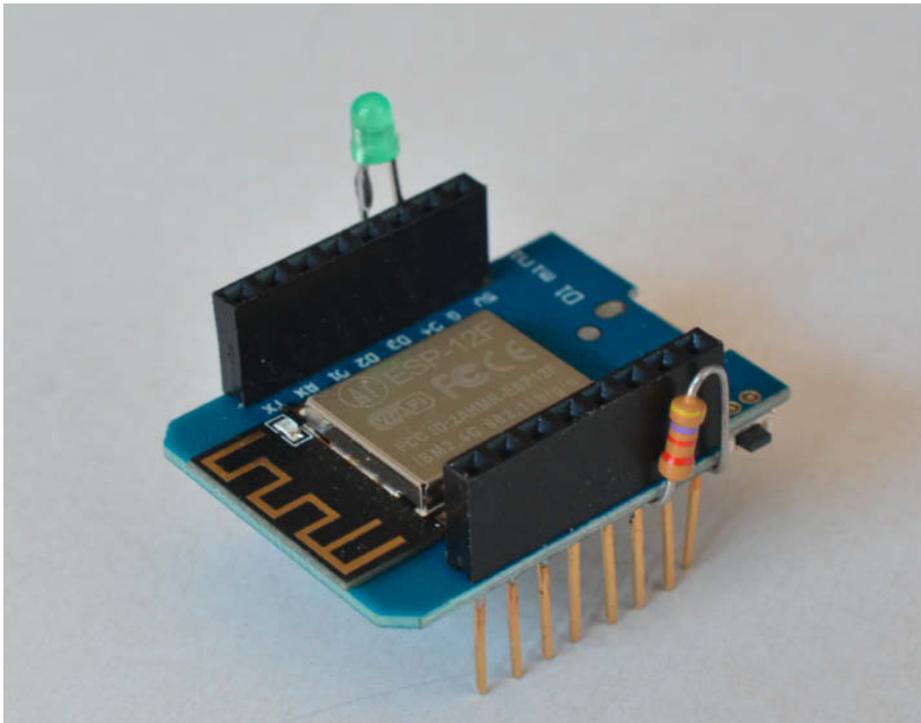
Welche Hardware-Anforderungen bestehen? Die Elektronik im IoT-Button sollte natürlich wenig Platz brauchen; das gilt insbesondere dann, wenn der Button mobil verwendet werden soll. Eine WLAN-Verbindung ist notwendig, und die Energieversorgung sollte

autark über einen Akku oder eine Batterie erfolgen können. Um einen langfristigen Betrieb zu gewährleisten, darf unsere Hardware nur sehr wenig Energie verbrauchen, in inaktiven Zeiten ist ein stromsparender Sleep-Modus von Vorteil. Um den Aufwand in Grenzen zu halten, sollte nach Möglichkeit ein fertiger IoT-Baustein zum Einsatz kommen, dessen Programmierung man leicht selbst ändern kann.



1 Funktionsweise des IoT-Buttons: Der Trick ist die Verwendung externer Dienste.

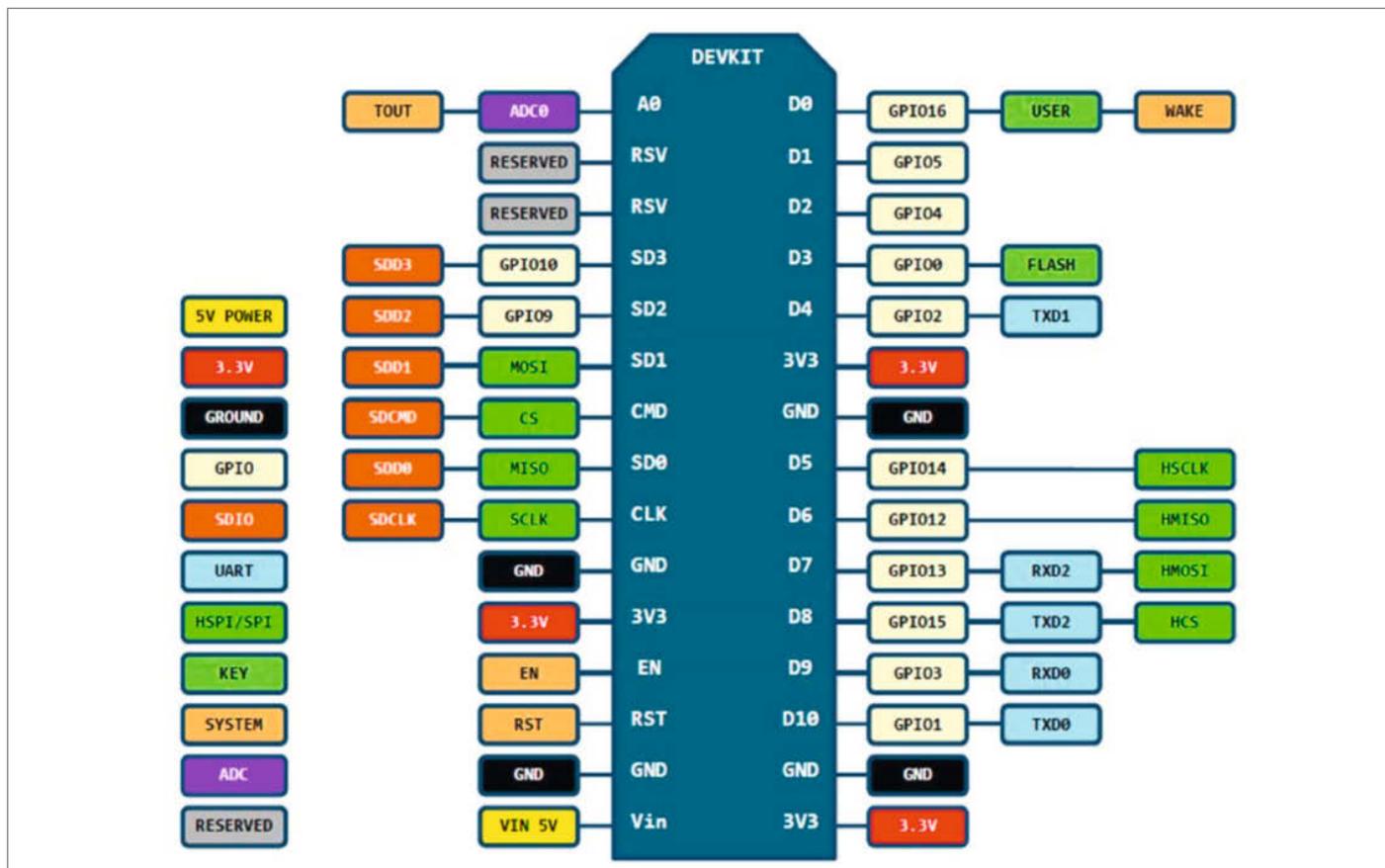
Pflichtenheft



2 Für den Aufbau empfiehlt sich ein fertig aufgebautes Modul wie das Wemos D1. Im Unterschied zum gleich gut geeigneten NodeMCU-Board muss man hier einen Pull-up-Widerstand von 4k7 vom Tastereingang D7 zur 3,3V-Versorgungsspannung vorsehen.

WLAN-fähige Kleinstrechner gibt es zur Genüge, jedoch scheint hier selbst ein Raspberry Pi Zero W überdimensioniert, wobei auch der relativ hohe Stromverbrauch eine unbeschwertere mobile Anwendung ausschließt. Besser geeignet sind kompakte ESP8266-Module wie das ESP-12 oder gleich fertig aufgebaute, Breadboard-kompatible Boards mit diesem Prozessor, zum Beispiel der *ESP8266 NodeMCU* 2 oder der vergleichbare *Wemos D1*, die oft für weniger als 7 Euro angeboten werden, selbst bei deutschen Händlern. Wer zwei bis drei Wochen Zeit hat, bekommt derlei Boards in China noch günstiger.

Den Taster schließen Sie an Pin D7 (GPIO 13) und den Masseanschluss (GND) an . Beim Wemos-Modul müssen Sie noch einen Pull-up-Widerstand 4k7 zwischen D7 und 3V3 vorsehen. Die Stromversorgung erfolgt (zunächst) über das USB-Kabel. Auf dem Mikrocontroller-Board befindet sich eine Status-LED (Anschluss: GPIO2 = D4) 3 . Weitere Bauelemente sind erst einmal nicht notwendig. Für den praktischen Einsatz können später ein optisch attraktiver Button und ein passendes Gehäuse für die Platine, den Taster und Akku oder Batterie ausgewählt werden.



3 Pinbelegung des ESP8266 NodeMCU. Das kleinere Board Wemos D1 hat weniger Pins herausgeführt, der für den Taster verwendete Pin GPIO13 ist dort ebenfalls mit D7 bezeichnet.

https://www.handsontec.com/pdf_learn/esp8266-v10.pdf

Kleine Hürden

Die aktuelle Version der Arduino-IDE 1.8.12 scheint leider zum Zeitpunkt der Erstellung des Artikels (Anfang Juni 2020) nicht ganz mit der Erweiterung zum ESP8266 Board kompatibel zu sein. Beim Versuch, das Board wie beschrieben zu installieren, ist uns auf einem Windows-Rechner die Entwicklungsumgebung abgestürzt. Dabei ist nicht klar, ob es an der Erweiterung oder an der IDE selbst lag. Möglicherweise hilft es, wenn Sie eine vorherige Version der Arduino-IDE installieren. Sollte die IDE nach einem solchen Fehler nicht mehr starten, dann löschen Sie alle Konfigurationsdateien im Verzeichnis `\Benutzer\AppData\Arduinio15`, sie werden bei einem Neustart der IDE neu angelegt.

ESP8266 flashen

Das Programmieren gestaltet sich relativ einfach. Wir verwenden dazu die Arduino-IDE, benötigen also keine speziellen Werkzeuge. Wie man den passenden Treiber für den USB-Chip CH340 installiert und die Arduino-IDE für den ESP8266 erweitert, zeigten wir bereits mehr-

fach in älteren Artikeln (siehe Link in der Kurzinfo).

Stellen Sie nun unter *Werkzeuge/Board* Ihr Modell ein (*Wemons R1D1* oder *NodeMCU 1.0*) und kontrollieren Sie die eingestellte Schnittstelle (die Windows nach Gutdünken verteilt - notfalls hilft ein Blick in den Geräte-Manager, auf welchem COM-Port der CH340 gelandet ist). Damit sind wir schon mit den Installationsarbeiten fertig und können mit der Entwicklung der Firmware beginnen.

Verbindung zur Welt

Die entscheidende Funktion des IoT-Buttons ist das Senden des Signals über die gedrückte Taste mittels WLAN an einen Server in der Cloud. Das Übermitteln der Daten mittels Netzwerk (erst recht eines WLANs) ist eigentlich eine recht komplexe Angelegenheit. Dank einer cleveren Bibliothek sind wir mit wenigen Zeilen Code fertig. Eine Bibliothek bindet man im Quellcode über den Befehl `#include` ein, sie heißt `ESP8266WiFi.h`. Ebenso benötigen wir die Zugangsdaten zum WLAN, also die SSID und das Passwort. Die Platzhalter (siehe Listing) müssen Sie durch die konkreten Angaben Ihres WLAN ersetzen. Kompilieren Sie das Programm: Es gibt über die serielle Schnittstelle *Ich verbinde...* aus, bis es sich an Ihrem Router angemeldet hat. Sie können die Ausgabe über den *Seriellen Monitor* der Arduino-IDE mitverfolgen.

The screenshot shows the Beeceptor website interface. At the top, there is a navigation bar with links for Usage, FAQ, Pricing, and Customer. The main content area features the Beeceptor logo and the text: "Rest API mocking and intercepting in seconds. Replace the endpoint in the code and you are ready. It's that simple!". Below this, there is a form with two input fields: "make-magazin" and ".free.beeceptor.com". A message below the form states: "A sub-domain will be created for this endpoint where you can send requests. Your endpoints: #make-magazin". A blue "Create Endpoint" button is positioned below the form. Underneath the button, there are sections for "Use cases" and "Features".

Use cases

- Build a mock Rest API in a few seconds. Free, No coding required.
- Inspect payloads of any HTTP request (GET, POST, PUT, PATCH, DELETE, etc).
- Customizable responses to simulate API response and failures.
- When load testing your API, do you really need to pass on the load to downstream APIs?
- Simulate latencies, timeouts and slow-responses of downstream APIs (validate rarely reachable code paths)
- A/B testing by switching API endpoints or versions without any redeployment. (prod code vs new code)
- Don't block your UI devs when backend APIs are still in development. Just mock it!
- Create webhook endpoints and simulate responses.

Features

- HTTP request intercepting for debugging and inspecting payload.
- Get named endpoints/sub-domains - easy to replace base URL in your code.
- Proxy Setup: Rule based approach to mock a few calls and hit real API for the rest.
- Capturing HTTP requests in real time.
- Support for CORS & preflight requests (OPTIONS) is out of the box.
- Keep Calm & Get 200 OK.
- All endpoints are HTTPS enabled.
- Requestbin + Mocky.io + Http-proxy => Beeceptor ❤️

4 Denken Sie sich einen schönen Host-Prefix für Beeceptor aus, und schon ist der Endpunkt fertig.

NEU
IM SORTIMENT



PETG

- SCHLAGZÄH
- MECHANISCH BELASTBAR
- LEBENSMITTELTECHNISCH UNBEDENKLICH ZERTIFIZIERT

impak

- SPEZIELL ENTWICKELT FÜR EXTREME MECHANISCHE BELASTUNGEN
- HOHE CHEMISCHE BESTÄNDIGKEIT
- LEICHT ZU DRUCKEN
- SPANBAR

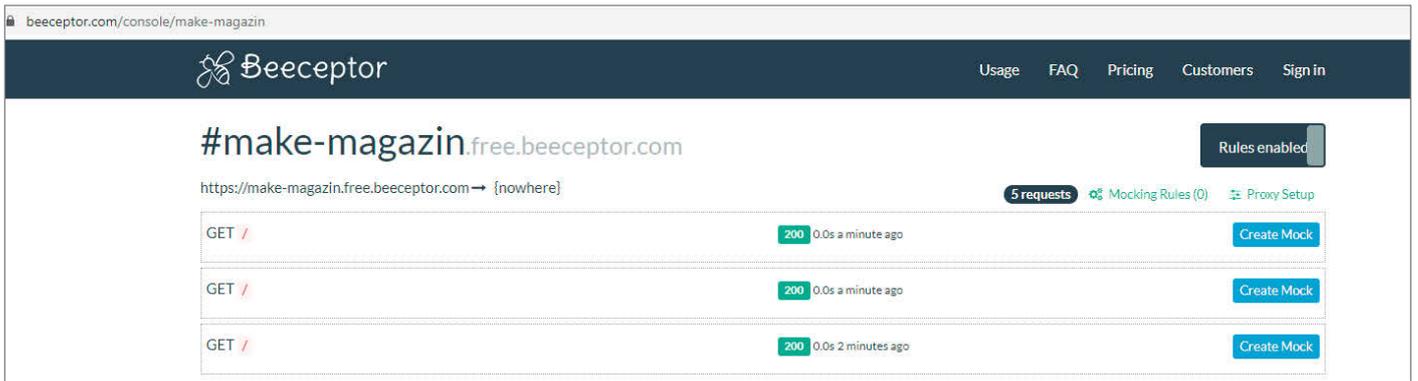


3dk.berlin
Kunststoffe für 3D Drucker



www.3dk.berlin

3dk.berlin



5 Erster Test: Unsere Messages vom IoT-Button kommen beim Cloud-Service Beeceptor an.

Wenn keine Meldung mehr erfolgt, hat sich der ESP8266 erfolgreich mit Ihrem Router verbunden. Doch wo sollen die Daten ankommen und wie funktioniert die Kommunikation? Dazu nutzt man am besten einen Service in der Cloud. Dabei handelt es sich um einen Server, welcher für die Aufgaben des Datenaustauschs mit IoT-Geräten spezialisiert ist. Die Kommunikation erfolgt über das Web via http-Protokoll und REST.

Gib dir den REST

Werden Geräte, beispielsweise ein IOT-Gerät und ein Server, über das Netzwerk (Internet) miteinander verbunden und sollen darüber Daten ausgetauscht werden, dann basiert das auf der Basis von REST (REpresentational State Transfer). RESTful-APIs basieren auf standardi-

sierten Verfahren und Datenformaten wie HTTP/S, URI, JSON oder XML und bieten damit eine universell nutzbare Schnittstelle. Dabei kann man die http-Methoden GET, POST, PUT/PATCH und DELETE nutzen.

Kleiner Exkurs: Bei einer REST-Anfrage (Request) sendet der Client (wie eben ein IoT-Gerät) etwas an den Server und erhält eine Antwort (Response). Eine Anfrage besteht aus den Bestandteilen Endpunkt (endpoint), Methode (method), Header (header) und Daten (data). Der Endpunkt setzt sich aus dem root-endpoint, dem path und möglichen Parametern zusammen. Beispielsweise ist `https://api.github.com` der root-endpoint von GitHub und `/users/veikkoef/repos` der path zu den Repositories des Autors. Zusammen ergibt dieses `https://api.github.com/users/veikkoef/repos`. Es können noch Parameter folgen, zum Beispiel `?query1=value1&query2=value2`, die ein Skript auf dem Server auswerten muss.

Der Header dient zur Bereitstellung von Informationen für den Client und den Server. Dieser kann für viele Zwecke verwendet werden, zum Beispiel zur Authentifizierung und Bereitstellung von Informationen über den nachfolgenden Inhalt. Als letztes Element einer Anfrage folgen die Daten, welche man an den Server übermitteln möchte. Der Body-Bereich ist nur relevant für die Operationen POST, PUT, PATCH oder DELETE. POST wird für Create-Operationen (also das Erzeugen eines Datensatzes) eingesetzt, die Daten sind nicht in der URL sichtbar. PUT/PATCH verwendet man für Update-Operationen und DELETE für das Löschen von Daten.

Einfach geht auch

Aber keine Angst, wir beschränken uns hier auf den einfachsten Fall, das Senden eines GET-Befehls, der für Read-Operationen steht. Obwohl GET-Zugriffe eigentlich nur lesend auf den Server zugreifen, kann man über die in der URL kodierte Anfrage-Zeichenkette natürlich auch ein paar Daten an den Server übermitteln - etwa mit der typischen Anfrage GET

`/?suche=meyer`, bei der ja auch der zu suchende Name übermittelt wird. Statt `?suche=meyer` könnte man auch `?knopf_zustand=1` oder `?knopf_zustand=0` senden - Sie sehen hier schon, worauf wir hinauswollen.

Einen eigenen Server zum Empfangen der Daten der IoT-Geräte könnte man selbst aufsetzen, was aber für den Anfang zu aufwändig und auch nicht notwendig ist: Man nutzt einen Cloud-Service. Zum Test verwenden wir den Dienst `https://beeceptor.com/`. Dort können wir einen so genannten REST-Endpunkt definieren, eine Empfangsstelle für unsere Daten vom IoT-Gerät. Die Konfiguration ist auf der Webseite in wenigen Augenblicken erledigt **4**. Wir verwenden hier `https://make-magazin.free.beeceptor.com`, die URL ist natürlich Ihrem Endpunkt entsprechend anzupassen. Wenn Sie diese Adresse über Ihren Browser aufrufen, sollte die unter „Rules“ vorgegebene Antwort erscheinen und der Aufruf im Beeceptor-Browserfenster protokolliert werden – genau wie Anfragen unseres IoT-Buttons.

Firmware

An diesen Endpunkt wollen wir nun eine Message vom ESP8266 via WLAN und das Internet versenden (siehe Listing). Wenn eine Internetverbindung besteht (`if (WiFi.status() == WL_CONNECTED)`), dann wird der Status der Taste überwacht und bei einem Tastendruck eine GET-Anfrage an den definierten Endpunkt gesendet. Die Adresse für den Endpunkt im Quellcode ist natürlich auf den von Ihnen gewählten Beeceptor-Endpunkt anzupassen.

Das Senden der GET-Anfrage führt zu einer Antwort des Servers. War es erfolgreich, so lautet der Antwortcode 200 und wir zeigen das im Terminal der IDE an (*Werkzeuge/Serieller Monitor*); gegebenenfalls wird der Fehlercode ausgegeben. Die Überwachung des Tastendrucks, der auch von der LED auf dem Board angezeigt wird, erfolgt in einer Endlosschleife (loop), das Programm endet also nur, wenn der ESP 8266 von der Spannungsversorgung getrennt wird. Ob die Daten beim

New Channel

Name:

Description:

Field 1:

Field 2:

Field 3:

Field 4:

Field 5:

Field 6:

Field 7:

Field 8:

Metadata:

Tags:
(Tags are comma separated)

Link to External Site:

Link to GitHub:

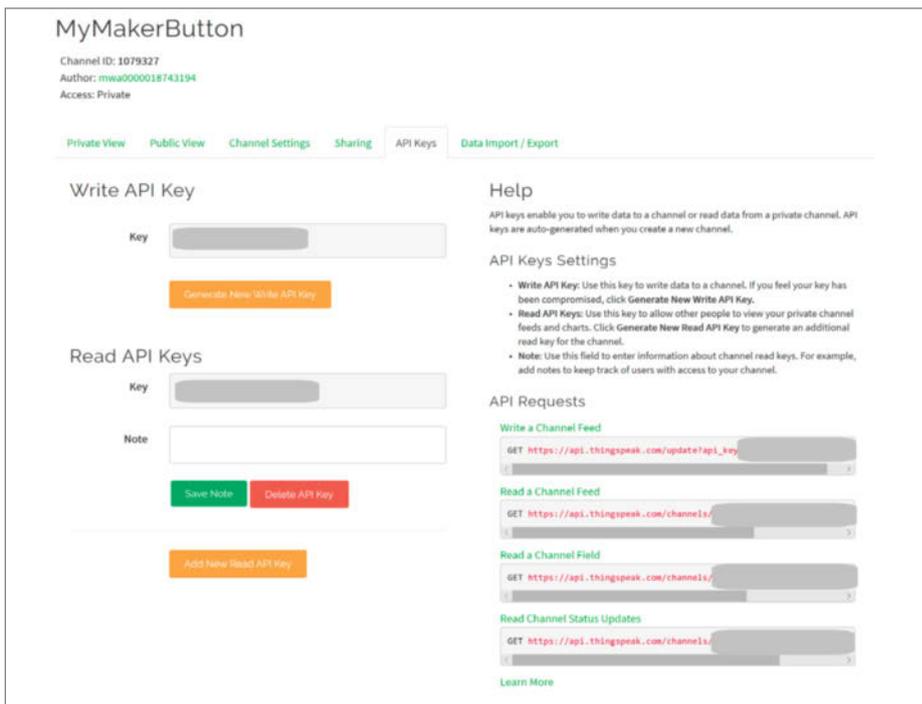
6 Bei Thingspeak deklarieren wir einen neuen Channel (Endpunkt).

Dienst angekommen sind, lässt sich direkt online nachverfolgen. Gehen Sie zu Ihrem Endpunkt der Seite und verfolgen Sie die Anfragen **5**.

Daten verarbeiten

Wenn „die Technik steht“ und die Daten in der Cloud ankommen, stellt sich die Frage, was man nun in der Praxis damit macht. Wir probieren dazu den Dienst ThingSpeak aus. Dieser (registrierungspflichtige, aber für nicht-kommerzielle Nutzung kostenfreie) Dienst dient als vermittelnde Zwischenschicht zu weiteren über das Web erreichbaren Services. Er nimmt die gesendete Nachricht entgegen und löst daraufhin eine Aktion aus. REST-Endpunkte heißen hier Channels **6**.

Jeder Endpunkt hat einen eigenen Namen (hier *MyButton*) und eine eigene ID (automatisch vergeben). Zusätzlich können wir noch Felder für Parameter definieren, bei einem einfachen IoT-Button (GET-Methode, siehe oben) ist das nicht notwendig. Wir werden dennoch Feld 1 verwenden, um eine Aktion auszulösen. Für uns ist hier der *Write-Key* wichtig; die zugehörige URL wird in *Write a Channel Field* angezeigt **7**, diese haben wir im Listing



7 Die API-Keys sichern den Zugang zum Dienst ThingSpeak. Über die angezeigten „API Requests“ lassen sich Werte ändern oder auslesen. Für eine einfache Visualisierung im Web-Browser verwenden wir die „Write a Channel Feed“-URL.

Jetzt durchstarten mit dem ESP32!
 Als Heft oder Digital: shop.heise.de/esp-special

INKLUSIVE:
 Original ESP32-Modul

PROJEKTE
 Mini-Webserver mit Temperaturmessung • Bluetooth-Sensoren • Funktionsgenerator • Piano mit Toucheingabe • Türüberwachung

PORTOFREI
 AB 15 €
 BESTELLWERT

heise shop
shop.heise.de/esp-special

Generell portofreie Lieferung für Heise Medien- oder Maker Media Zeitschriften-Abonnenten oder ab einem Einkaufswert von 15 €. Nur solange der Vorrat reicht. Preisänderungen vorbehalten.

Bühne frei für die besten Hacks von **Make Your School!**

Maker Woche
 8. bis 11. September 2020

jeden Tag ab 16 Uhr live auf YouTube

Trefft auf Maker*innen, Ideenentwickler*innen und Akteur*innen aus dem Bildungsbereich. Taucht bei der digitalen Maker Woche in ihre Welt ein – in **Tutorials, Vorträgen** und **Live-Performances**. Seid mit dabei, bei der Woche für alle Schulverbesser*innen, Prototyp-Entwickler*innen, Hacker*innen und Neugierige.

Livestream, Programm und weitere Infos:
www.makeyourschool.de/maker-festival

8 Werte, die man ThingSpeak im Query-Teil der URL mitgibt, erscheinen in einem Diagramm. Hier haben wir verschiedene Werte für „field1“ ausprobiert.

im Abschnitt THINGSPEAK eingetragen. Wenn Sie diese URL mit verschiedenen Werten für `field1` testweise im Browser aufrufen, sollte sich im *Private View*-Tab Ihres Channels bereits etwas tun: Die der URL mitgegebenen Werte (`field1=1`, `field1=5` usw.) erscheinen in einem Diagramm **8**.

Beachten Sie, dass ThingSpeak in der kostenlosen Version nur 8200 Messages pro Tag verarbeitet, das sind weniger als 6 pro Minute.

Übrigens lassen sich die Visualisierungen (wie das gezeigte Diagramm) über ein `HTTP-iframe` auch in eigene Webseiten einbinden.

Neben den hauseigenen Services aus dem Bereich Analytics und Visualisierung sind insbesondere die folgenden Optionen interessant, bei ThingSpeak als *Apps* bezeichnet: Mit *ThingTweet* zum Beispiel können wir uns direkt mit einem aktiven Twitter-Konto verbinden. ThingTweet verwendet einen eigenen

API-Key und eine andere URL (siehe Abschnitt „THINGTWEET“ im Listing). Den API-Key bekommen Sie, wenn Sie in der ThingTweet-App-Konfiguration auf *Link Twitter Account* klicken und einige Angaben zu Ihrem Twitter-Account machen.

Für ThingTweet ist eigentlich eine POST-Message vorzuziehen, es geht aber auch mit der einfacheren GET-Message, indem man die Nachricht in der URL kodiert: Mit

```
https://api.thingspeak.com/apps/thingtweet/1/statuses/update?api_key=XXXXXXX&status=Hallo+Welt
```

wird „Hallo Welt“ auf Ihrem Twitter-Account gepostet. Beachten Sie, dass Leerzeichen nicht übertragen werden, man muss sie durch ein „+“-Zeichen substituieren (diese kleine Sorge hätte man mit einer POST-Message allerdings nicht). Natürlich müssen Sie auch die „XXXXX“ durch die ThinkSpeak-API-ID ersetzen. Wie solche Variablen bequemer über ein Web-Interface geändert werden können, haben wir im Artikel „ESP8266-Dashboard“ (siehe Info-Kasten) gezeigt.

Immer wenn der Button gedrückt wurde, wird ThingTweet die API von Twitter aufrufen, und wir können eine voreingestellte Nachricht („Hallo, ich bin zu Hause“) versenden. Kleine Anmerkung: Das ginge im Prinzip auch direkt über die Twitter-eigene API, was aber wegen der recht umständlichen *oauth*-Authentifizierung sehr viel komplizierter ist.

Über *React* kann man eine Aktion auslösen, wenn eingehende Daten, zum Beispiel aus Feld 1, einen bestimmten Wert überschreiten. Wenn etwa eine Temperatur gemessen wird und diese einen bestimmten Wert überschreitet, wird eine Aktion ausgelöst. Das kann zum Beispiel der Aufruf eines weiteren Dienstes sein, aber auch die ThingSpeak-eigene ThingTweet-App selbst. Somit müssten Sie noch nicht einmal den Quellcode für ThingTweet anpassen, es genügt, in der React-App als die Bedingung `Field 1 contains 1` einzutragen, da unsere Firmware diesen Wert in der URL mitliefert (`field1=1`).

ThingHTTP dürfte die komplexeste, aber auch interessanteste App sein **8**. Hierüber können Sie über das Internet beliebige andere Geräte und Services anbinden, etwa ein bestehendes Smart-Home-System. Das erfordert aber umfassende Kenntnis des einzubindenden Systems, weshalb wir hier nicht näher darauf eingehen können.

Hackpflicht

Die Kreativität besteht in der Auswahl und der geschickten Nutzung von Cloud-Diensten, wobei wir nur einige Möglichkeiten skizziert haben. Eine Alternative ist zum Beispiel auch der Dienst *IoT-Hub* von Microsoft (siehe Link im Info-Kasten). —cm

Weniger Energie

Es gibt die Möglichkeit, den Energieverbrauch des ESP8266 im Deep-Sleep-Modus erheblich zu senken, das heißt auf wenige Mikroampere. Damit eignet sich das NodeMCU-Board auch dazu, den Mikrocontroller unter Umständen mehrere Monate aus einer Batterie (oder Akku) zu versorgen. Eine externe Spannung wird an die Pins +5V und GND angeschlossen, zum Beispiel in Form einer Power-Bank. Es gibt zwei Varianten für den Deep-Sleep-Modus.

Beim Software-Deep-Sleep wacht der ESP8266 nach einer bestimmten Zeit eigenständig wieder auf. Dazu muss eine Verbindung von GPIO 16 (D2) zum Anschluss RST hergestellt werden. Nach der eingestellten Zeit wird ein Signal gesendet, womit der ESP eigenständig neu startet. Die maximale Zeitdauer ist jedoch auf

etwa 71 Minuten begrenzt. Diese Variante eignet sich hier nicht, da wir kein festes Zeitintervall für das Auslösen des Ereignisses (Tastendruck) haben.

Der Interrupt-Deep-Sleep weckt den ESP8266 dagegen über einen Interrupt aus dem Deep-Sleep-Modus. Er startet nicht eigenständig neu. Diese Form des Deep-Sleep-Modus wird mit dem Befehl `ESP.deepSleep(0)` gestartet. Sobald das Ereignis ausgelöst wurde, sendet man ein Low-Signal (Verbindung von RST mit GND). Dadurch startet der ESP8266 neu und die Software wird ausgeführt – und zwar genau so lange, bis er auf den `sleep`-Befehl trifft. Der IoT-Button wird dann nicht an Pin D7 angeschlossen, sondern schlicht am Reset-Eingang RST. Das angepasste Listing finden Sie unter dem Link im Info-Kasten.

IoT-Button

```

01 #include <ESP8266WiFi.h>
02 #include <ESP8266HTTPClient.h>
03 // #define BEECEPTOR
04 #define THINGSPEAK
05 // #define THINGTWEET
06 int button = D7;
07 int status= 1;
08 String APIkey_ThingTweet = "II4I97696990TAW8"; // durch Ihren ThingTweet-API-Key ersetzen
09 String APIkey_ThingSpeak = "2S7PDUJH9876HCAM"; // durch Ihren ThingSpeak-API-Key ersetzen
10 HTTPClient https;
11 void setup() {
12   Serial.begin(115200);
13   pinMode(LED_BUILTIN, OUTPUT);
14   digitalWrite(LED_BUILTIN, HIGH);
15   pinMode(D7, INPUT);
16   digitalWrite(D7, HIGH);
17   WiFi.begin("<Ihre WLAN-SSID>", "<Ihr WLAN-Password>");
18   delay(1000);
19   Serial.println("");
20   Serial.print("Ich verbinde...");
21   while (WiFi.status() != WL_CONNECTED)
22   {
23     digitalWrite(LED_BUILTIN, LOW);
24     delay(500);
25     digitalWrite(LED_BUILTIN, HIGH);
26     delay(500);
27     Serial.print(".");
28   }
29   Serial.println(" Done!");
30 }
31 void loop() {
32   String payload, tweet, button_value;
33   if (WiFi.status() == WL_CONNECTED) {
34     status = digitalRead(button);
35     if (status == LOW) {
36       BearSSL::WiFiClientSecure client;
37       client.setInsecure();
38       Serial.println("Button betätigt!");
39       digitalWrite(LED_BUILTIN, LOW);
40 #ifndef BEECEPTOR
41       // Abschnitt mit URL zum Testen per Beeceptor
42       // Auf eigene Beeceptor-Seite anpassen!
43       https.begin(client, "https://make-magazin.free.beeceptor.com");
44 #endif
45 #ifndef THINGSPEAK
46       // Abschnitt für Visualisierung von Thingspeak
47       button_value = "1";
48       https.begin(client, "https://api.thingspeak.com/update?api_key="
49         + APIkey_ThingSpeak + "&field1=" + button_value);
50 #endif
51 #ifndef THINGTWEET
52       // Abschnitt für Twitter-API von Thingspeak
53       tweet = "Der+Button+wurde+betätigt";
54       https.begin(client, "https://api.thingspeak.com/apps/thingtweet/1/statuses/update?api_key="
55         + APIkey_ThingTweet + "&status=" + tweet);
56 #endif
57       Serial.println("Antwort des Dienstes:");
58       int httpCode = https.GET();
59       if (httpCode == 200) {
60         payload = https.getString();
61         Serial.println("Event: " + payload);
62       } else {
63         Serial.print("Fehler: ");
64         Serial.println(httpCode);
65       }
66       https.end();
67       // Warten, bis Taste wieder losgelassen. LED blinkt schnell
68       while (digitalRead(button) == LOW) {
69         digitalWrite(LED_BUILTIN, LOW);
70         delay(150);
71         digitalWrite(LED_BUILTIN, HIGH);
72         delay(150);
73       }
74       digitalWrite(LED_BUILTIN, HIGH);
75     }
76   }
77 }

```

Power aus dem Nabendynamo

Einen Laderegler für den Anschluss von Smartphones oder Tablets an den Nabendynamo von Fahrrädern kann man sich mit wenigen Bauelementen selbst bauen. Wir zeigen, wie Nabendynamos funktionieren und mit welchen Mitteln man die erzeugte Spannung regelt.

von Andreas Gräber



Viele Radfahrer haben mittlerweile eine richtige Lichtmaschine an Bord, einen Nabendynamo. Diese sehr zuverlässigen Dynamos haben einen Wirkungsgrad, mit dem sie nicht nur die Fahrradbeleuchtung versorgen, sondern auch den Saft für Smartphones oder Tablets liefern können. So kann man auf jeder Fahrradtour Navigations-Apps mit stromfressender GPS-Unterstützung laufen lassen, ohne fürchten zu müssen, dass der Akku nicht ausreicht.

Für diesen Zweck gibt es sehr viele fertige Geräte – meist in Form von USB-Laderegler – mit deren Hilfe man die am Nabendynamo anstehende Spannung in die für das Smartphone passende Form bringen kann. Es ist aber auch möglich, einen solchen Laderegler selbst sehr kostengünstig zu bauen. Die entsprechende Schaltung ist ganz einfach und kommt mit sehr wenigen Bauelementen aus. Bevor wir die Ladeschaltung vorstellen und erläutern, zeigen wir wie ein Nabendynamo eigentlich funktioniert.

Der Nabendynamo

Der Nabendynamo wandelt mechanische Energie in elektrische Energie um. Er funktioniert auf der Basis der magnetischen Induktion. Die magnetische Induktion umfasst das Motor- und das Generatorprinzip: Durch die richtige Anordnung von Magneten und einer Spule kann entweder Bewegung durch Stromfluss oder Stromfluss durch Bewegung ausgelöst werden. Hier geht es um letzteres: Durch die Bewegung von Magneten um eine Spule herum wird Stromfluss induziert.

Der Nabendynamo besteht aus einem äußeren Ring aus Magneten, der mit den Speichen des Vorderrades verbunden ist und sich deshalb beim Fahren dreht. Dieser Ring heißt Rotor. An der Vorderradachse und -gabel – also exakt in der Mitte des Rotors – befindet sich eine auf einen Eisenkern gewickelte Spule. Dieser Eisenkern dreht sich nicht, genau wie die Achse und die Gabel, und wird deshalb als Stator bezeichnet. Am Stator sind außerdem so genannte Klauen oder Polschuhe aus Eisen befestigt, die vom Stator bis in unmittelbare Nähe zum Rotor reichen.

Zwischen den Magneten im Rotor befindet sich ein Magnetfeld. Durch die Klauen werden die magnetischen Flüsse der Magnetpole zur Spule geleitet. Sie wirken über die Lo-

Der Nabendynamo funktioniert durch Induktion: Magnete werden um eine Spule herum bewegt und erzeugen so Stromfluss.

Kurzinfo

- » Mobiltelefon oder Powerbank unterwegs auf dem Fahrrad laden
- » Induktion und Generatorprinzip anhand eines Nabendynamos erklärt
- » Aufbau eines Ladereglers

Checkliste



Zeitaufwand:
2 bis 3 Stunden



Kosten:
circa 10 Euro ohne Befestigung am Fahrrad



Elektronik:
einfache Löt- und Messarbeiten

Material

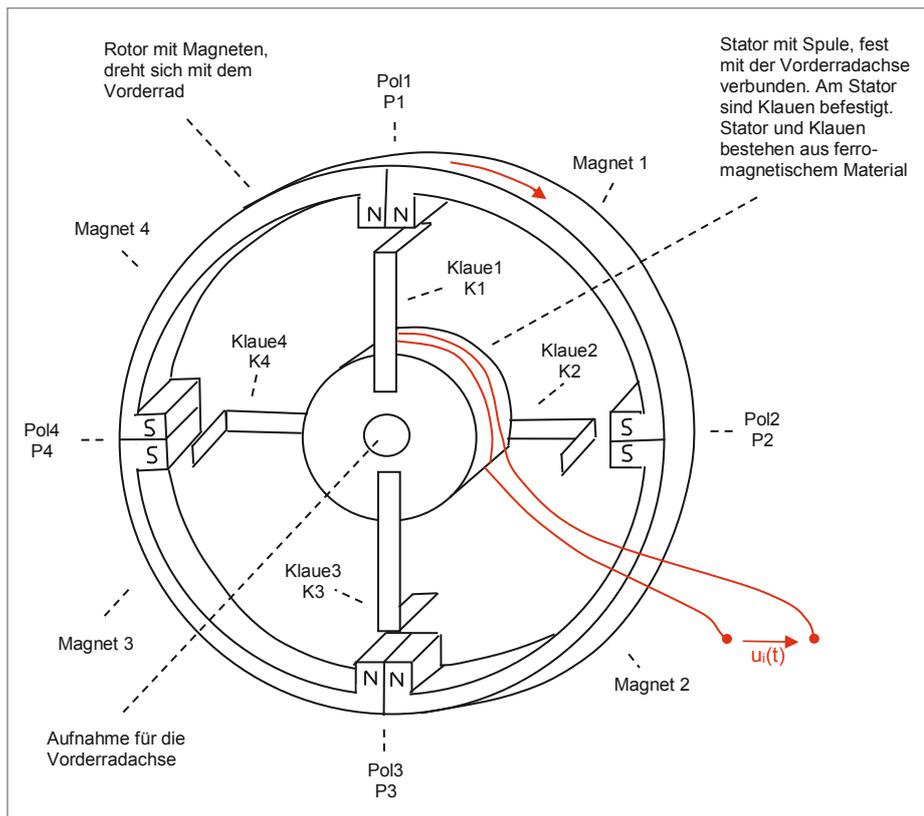
- » 4 Schottky-Dioden SB140
- » unpolarisierter Kondensator 330µF oder 2 in Reihe geschaltete Elektrolytkondensatoren von jeweils 660µF
- » Elektrolytkondensator 2200µF
- » DC-DC-Step-Down-Modul LM 2596
- » Lochrasterplatte für den Versuchsaufbau
- » Stecker für den Nabendynamo
- » USB-zu-Mikro-USB-Kabel zur Verbindung der Ladeschaltung mit dem Smartphone

Mehr zum Thema

- » Elke Schick, Die Gesetze der Anziehung, Make 1/2015, S. 126
- » Florian Schäffer, Spannungsumwandlung, Make Sonderheft 2018, S. 64
- » Carsten Meyer, Spannungsregler, Make 6/2015, S. 116

Alles zum Artikel
im Web unter
make-magazin.de/xr9y





Prinzipieller Aufbau eines Nabendynamos

rentz-Kraft auf die in der Spule vorhandenen frei beweglichen Elektronen, wodurch in der Spule ein Plus- und ein Minuspol entstehen. Der Eisenkern verstärkt die Ladungstrennung, weil sich in ihm Elementarmagnete befinden, die sich in Richtung des Magnetfeldes ausrichten.

In der im Bild dargestellten Stellung des Rotors verläuft der magnetische Fluss wie folgt:

Verlauf des von Pol 1 ausgehenden Flusses:
 Pol 1 → Klaue 1 → Spule → Klaue 2 → Pol 2
 Pol 1 → Klaue 1 → Spule → Klaue 4 → Pol 4

Verlauf des von Pol 3 ausgehenden Flusses:
 Pol 3 → Klaue 3 → Spulenkern → Klaue 2 → Pol 2
 Pol 3 → Klaue 3 → Spulenkern → Klaue 4 → Pol 4

Die Flussrichtung durch die Spule ist dabei jeweils gleich. Sie weist in die Zeichenebene hinein, also von der Front zur Rückseite der Spule.

Wenn der Rotor mit der Drehbewegung des Lauftrahrs um 90° nach rechts gedreht wird, nehmen die magnetischen Flüsse andere Verläufe:

Verlauf des von Pol 1 ausgehenden Flusses:
 Pol 1 → Klaue 2 → Spule → Klaue 3 → Pol 2
 Pol 1 → Klaue 2 → Spule → Klaue 1 → Pol 4

Verlauf des von Pol 3 ausgehenden Flusses:
 Pol 3 → Klaue 4 → Spule → Klaue 3 → Pol 2
 Pol 3 → Klaue 4 → Spule → Klaue 1 → Pol 4

Die Flussrichtung durch die Spule ist dabei wieder in beiden Fällen gleich, sie weist jedoch aus der Zeichenebene heraus, hat sich also verändert.

Jede weitere Drehung des Rotors um 90° nach rechts bewirkt wiederum eine Änderung des Flusses und der Flussrichtung durch die Spule. Und weil die frei beweglichen Elektronen in der Spule bewegt werden, wird so gemäß des Induktionsgesetzes eine Spannung induziert, die an den Spulenanschlüssen abgenommen werden kann. Die induzierte Spannung wird durch die ständig wechselnde Bewegungsrichtung des Stromflusses mal stärker und mal schwächer. Diese Schwankungen haben einen angenähert sinusförmigen Verlauf.

Der hier der Übersicht halber sehr vereinfacht dargestellte Generator – er wird auch als Klauenpolgenerator bezeichnet – besitzt vier Pole und vier Klauen. Bei einer Umdrehung des Rotors wird die Flussrichtung also viermal geändert. Die induzierte Wechselspannung durchläuft dann zwei Perioden. Wie sieht es nun mit der Frequenz der induzierten Spannung an der Spule aus? Da müssen wir ein wenig rechnen. Wir gehen davon aus, dass wir ein Vorderrad mit einem Durchmesser von circa 700mm (28“) benutzen. Damit ergibt sich ein Umfang von rund 2,2m. Wir nehmen nun einmal an, dass wir 15km/h fahren. Für eine Strecke von 15 km braucht

unser Rad etwa 7000 Umdrehungen (15.000m/2,2m = 6818,181818).

Da wir 15 km/h fahren, müssen wir diese 7000 Umdrehungen in einer Stunde durchlaufen. Pro Sekunde bedeutet das etwa 2 Umdrehungen (7000/(60·60) ≈ 2). Bei einem Generator wie in der Zeichnung dargestellt hätte die induzierte Wechselspannung dann eine Frequenz von etwa 4Hz. Das ist natürlich viel zu gering, das Lampenlicht würde dann fürchterlich flackern. Deshalb enthalten reale Nabendynamos meist 28 Magnete beziehungsweise 28 Pole. Damit ergibt sich bei einem 28“-Fahrrad und einer Geschwindigkeit von 15 km/h eine Frequenz der induzierten Spannung von 28Hz, das ist akzeptabel.

Die Ladeschaltung

Wie kann man nun mit dem Nabendynamo eine Powerbank, ein Smartphone oder ein anderes Gerät aufladen? Wir stellen hier eine Schaltung vor, die sich durch eine erfreuliche Einfachheit und Effizienz auszeichnet.

Den mit dem Dynamo erzeugten Strom leiten wir zuerst durch den Kondensator C_K. Kondensatoren bestehen üblicherweise aus zwei Metallplatten, die durch eine Isolationschicht (Dielektrikum) voneinander getrennt sind. Durch den Stromfluss wird eine der Platten negativ aufgeladen, so dass eine Potentialdifferenz zur anderen Platte entsteht. So werden Kondensatoren meist eingesetzt, um elektrische Ladung zu speichern.

Der Kondensator hier dient aber als kapazitiver Blindwiderstand zur (zumindest teilweisen) Kompensation der Induktivität des Nabendynamos. Damit verringert er den komplexen Gesamtinnenwiderstand des Generators. Der Nabendynamo hat einen Innenwiderstand, der durch eine Reihenschaltung aus einer Induktivität (Spule) und einem Ohmschen Widerstand (Spulendraht) dargestellt werden kann. Wenn man an den Dynamo einen passenden Kondensator anschließt, wird die Induktivität kompensiert und der Strom wird nur noch durch den Ohmschen Widerstand begrenzt.

Das liegt daran, dass Kondensatoren sich im Wechselstromkreis nicht aufladen. Da die Elektronen sich dort nicht in eine Richtung bewegen, sondern ihre Flussrichtung regelmäßig ändern, werden die Metallplatten im Wechsel negativ und positiv aufgeladen. Der Kondensator nimmt also zyklisch Leistung auf und gibt sie anschließend wieder an die Quelle ab.

Diese Kompensation ist, wie Messungen ergeben haben, ziemlich wirksam. Für C_K schlagen wir einen unpolarierten Kondensator von 330µF vor. Solche Kondensatoren sind aber nicht so leicht zu beschaffen, deshalb kann man beispielsweise auch zwei gegensinnig in Reihe geschaltete Elektrolytkondensatoren von 660µF verwenden. In diesem Fall

sind entweder die Pluspole der beiden Kondensatoren miteinander verbunden oder die Minuspole.

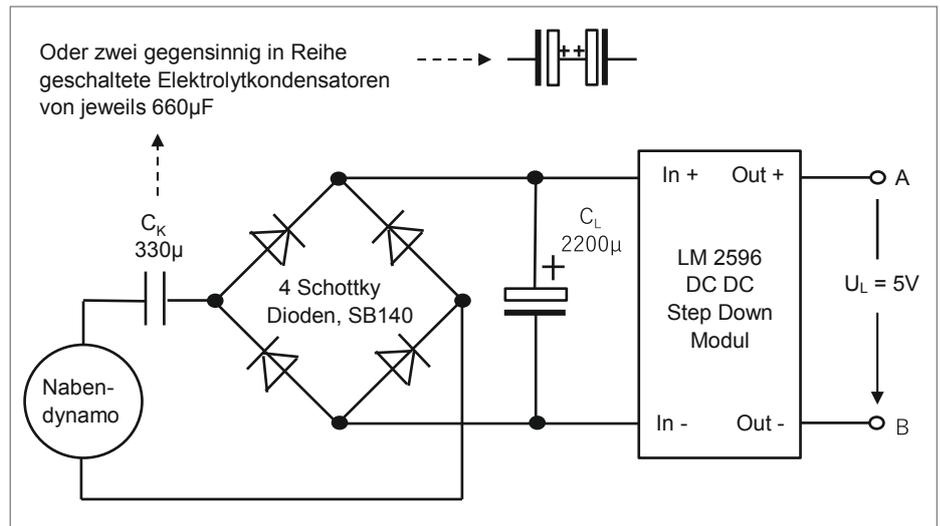
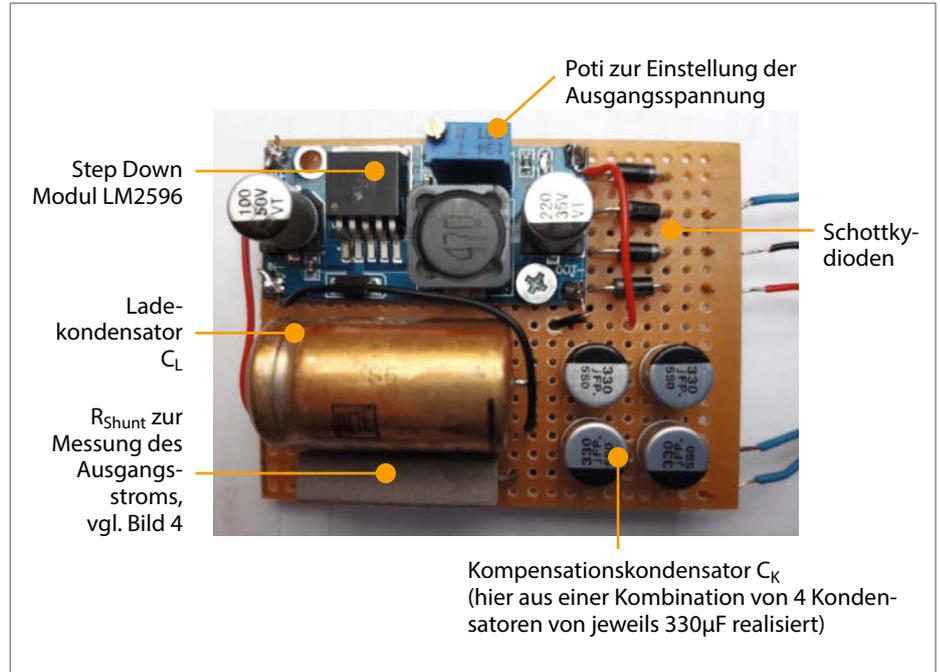
Die am Nabendynamo anstehende Spannung wird mit den 4 Schottky-Dioden zunächst gleichgerichtet. Durch Dioden kann Strom nur in eine Richtung fließen, sie haben eine sogenannte Durchlassrichtung und eine Sperrichtung. Mit ihnen wird der Wechselstrom, den der Dynamo erzeugt und dessen Flussrichtung sich immer wieder ändert, in Gleichstrom umgewandelt wird, der nur in eine Richtung fließt. Nach den Dioden folgt in unserer Schaltung ein Ladekondensator, der als Zwischenspeicher dient. Anschließend kommt das Herz der Schaltung, ein Step-Down-Modul mit dem IC LM2596.

Am Eingang dieses Abwärtswandlers kann man eine Spannung zwischen 3,2V und 36V anlegen. Die auszugebende Ausgangsspannung kann man mit einem auf dem Modul integrierten Poti einstellen. Sie kann zwischen 1,25 und 30V liegen, bei einem maximalen Ausgangsstrom von 2A. Die Eingangsspannung muss allerdings immer etwa 1V höher als die Ausgangsspannung sein.

Die Spannung am unbelasteten Nabendynamo kann über 30V liegen, sie sinkt aber bei Belastung weit unter die maximal zulässigen Eingangsspannung des LM2596-Moduls. An unsere Ladeschaltung sollten vorrangig Geräte angeschlossen werden, die über einen USB-Standard Anschluss verfügen. Deshalb muss die Ausgangsspannung des LM2596-Moduls mit Hilfe des auf dem Board befindlichen Potis möglichst exakt auf 5V eingestellt werden.

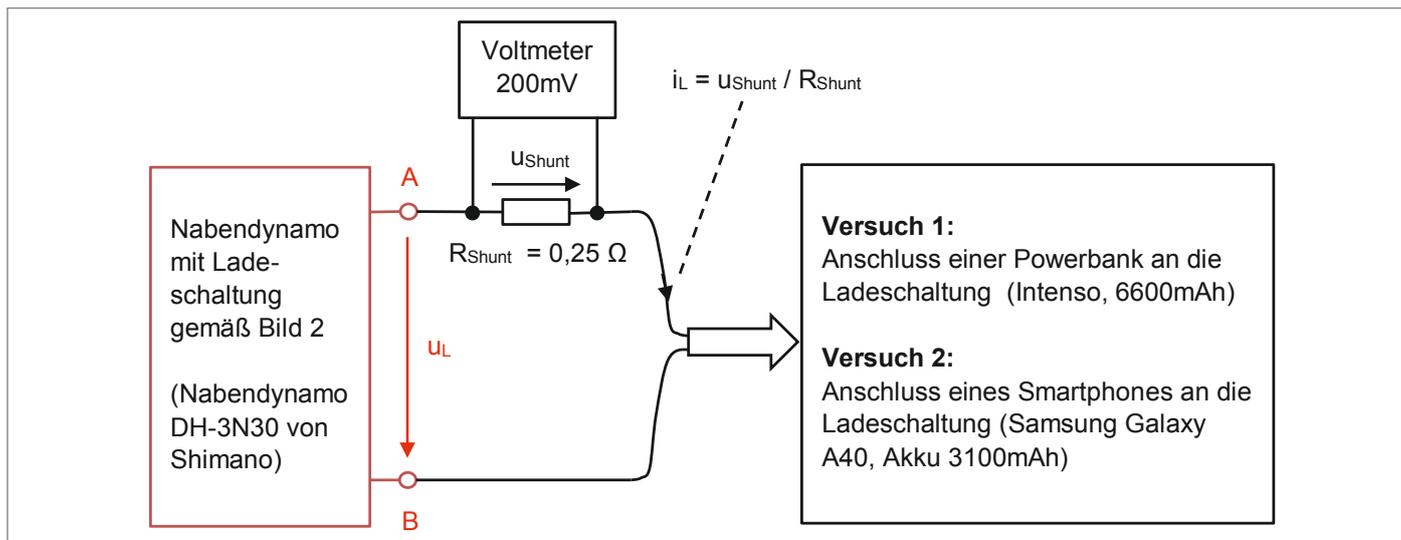
Was liefert die Ladeschaltung?

Wir haben zunächst eine Powerbank und anschließend ein Smartphone an die Ladeschaltung angeschlossen. Dann haben wir jeweils den Ladestrom i_L in Abhängigkeit von der



Das ist die gesamte Ladeschaltung, viel einfacher geht es nicht!





Die einfache Messschaltung für die Überprüfung unseres Laderreglers

Fahrradgeschwindigkeit v_{Fahrrad} gemessen. Am Ausgang der Ladeschaltung befindet sich ein Shunt als Messwiderstand. Dort haben wir mit Hilfe eines hochohmigen Messgerätes die Spannung gemessen. Als Shunt haben wir

einen ziemlich kleinen Widerstand von $0,25\Omega$ gewählt, man könnte dafür auch vier parallelgeschaltete 1Ω -Widerstände verwenden.

Die Ergebnisse der Messungen haben wir in der rechts oben stehenden Grafik erfasst. Zu-

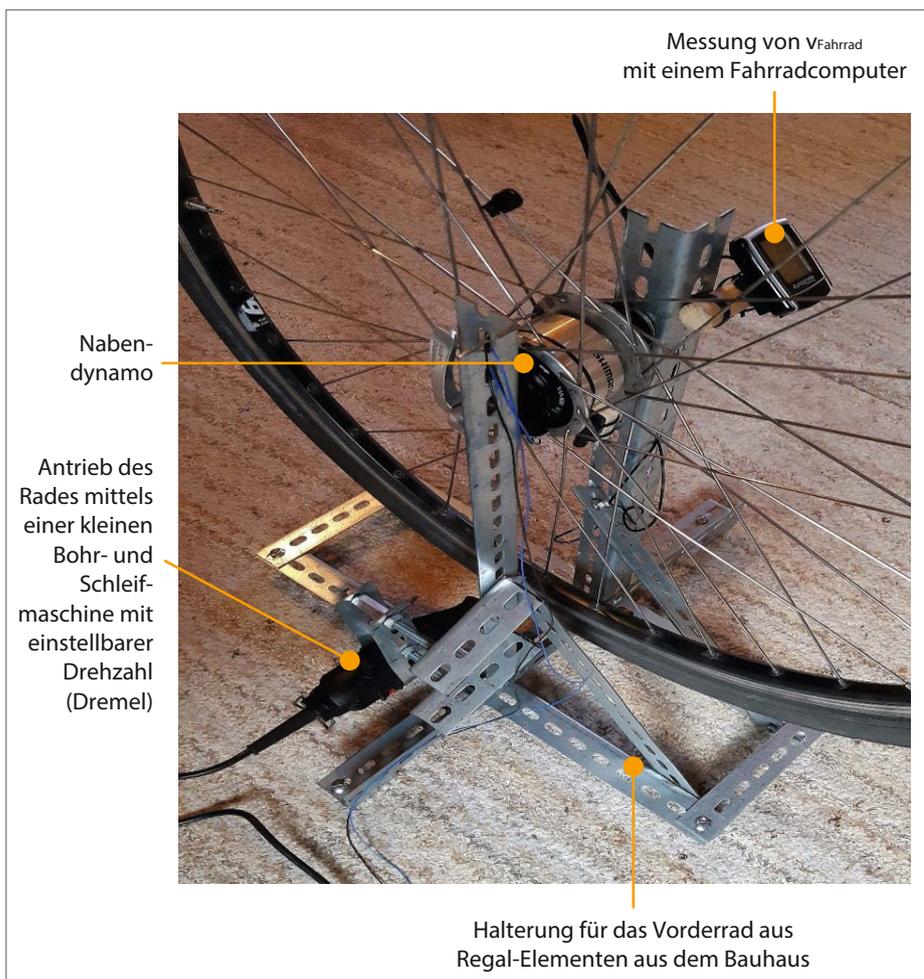
nächst ein paar Worte zur roten Kurve (Versuch 1, Anschluss einer Powerbank an die Ladeschaltung). Die Funktion $i_L = f(v_{\text{Fahrrad}})$ steigt hier zunächst ziemlich steil an und fällt dann mit zunehmender Fahrradgeschwindigkeit wieder ab. Verantwortlich dafür sind vermutlich Eisen- und Hystereseverluste, die durch den größer werdenden Strom immer mehr ins Gewicht fallen. Letztere steigen ja mit zunehmender Frequenz der induzierten Spannung. Und die wird ja mit zunehmender Fahrgeschwindigkeit immer größer! Aber das ist nicht so schlimm. Ein Radfahrer, der nicht gerade bei der Tour de France mitmischt, bewegt sich ja oft in einem Geschwindigkeitsbereich zwischen 15km/h und 25km/h . In diesem Bereich wird die Powerbank immerhin mit über 400mA geladen. Das ist schon ganz gut.

Die Versuche wurden mehrfach bei verschiedenen Ladezuständen der Powerbank wiederholt, die Ergebnisse waren immer fast identisch. Wenn der Akku voll geladen ist, stellt sich eine Erhaltungsladung von circa 8mA ein. Die in der Powerbank enthaltene Lade- und Überwachungsschaltung funktioniert wohl recht gut.

Nun zur grünen Kurve (Versuch 2, Anschluss eines Smartphones an die Ladeschaltung). Die Funktion $i_L = f(v_{\text{Fahrrad}})$ verläuft ähnlich wie die rote Kurve, weist aber insgesamt eine etwas flachere Form auf. Das liegt wahrscheinlich an der im Telefon enthaltenen Lade- und Überwachungsschaltung. Die Messungen haben wir jeweils an einem stationären Prüfstand und während einer Radtour durchgeführt. Die Messergebnisse stimmten weitgehend überein.

Handy-Akku schonen

Die Messergebnisse und die Versuchsfahrten haben gezeigt, dass es mit dem Laderegler



Messungen an einem stationären Prüfstand

möglich ist, unterwegs mit dem Handy das GPS zu benutzen, ohne dass sich der Akku des Handys entleert. Allerdings gibt es dabei eventuell ein Problem: Wenn das Smartphone direkt mit der Ladeschaltung verbunden ist, wird die Ladung naturgemäß sehr oft ein- und ausgeschaltet. Man denke nur an Ampelstops, Trinkpausen und so weiter. Das ist nicht gut für den Akku.

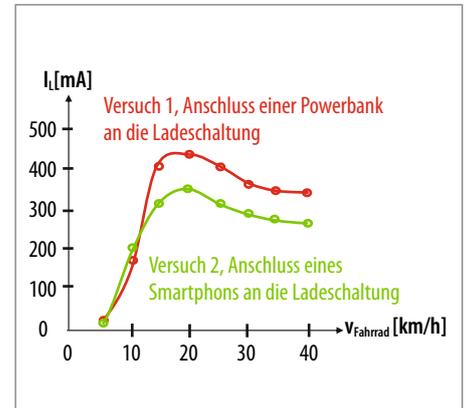
Es wäre deshalb sicherlich sinnvoller, zwischen Telefon und Ladeschaltung einen Energiespeicher zu schalten. Dazu könnte der Einfachheit halber eine Powerbank dienen. Aber da gibt es eine Schwierigkeit: Die Powerbanks kann man in den meisten Fällen nicht gleichzeitig laden und entladen. Nur die als charge-through/pass-through gekennzeichneten Powerbanks sind dafür geeignet und die gibt es ziemlich selten und sie sind auch relativ teuer. Eine mögliche Lösung wäre die Verwendung von zwei kleineren Powerbanks (etwa Varta 57959, 2600mAh). Zunächst wird Powerbank 1 an die Ladeschaltung angeschlossen und Powerbank 2 wird mit dem Telefon verbunden. Nach einem Zeitintervall wird umgeschaltet, so dass 1 zur Speisung des Telefons dient und 2 aufgeladen wird. Diese Umschalerei kann periodisch fortgesetzt wer-

den. Dazu bräuchte man nur einen Wechselrichter. Sicherlich könnte ein solcher Schaltvorgang auch automatisiert werden.

Fazit

Ganz am Anfang dieses Artikels haben wir bereits erwähnt, dass man für Nabendynamos geeignete USB-Laderegler in allen möglichen Preisklassen fertig kaufen kann. Viele dieser Geräte beinhalten allerdings keine Powerbank, genauer gesagt, keinen Pufferakku zwischen Ladeschaltung und Smartphone. Wenn man sich also für den Kauf eines Ladereglers entscheidet, sollte man darauf achten, ob er um einen Pufferakku ergänzt werden muss.

Der Selbstbau eines USB-Ladereglers lohnt sich selbstverständlich nur dann, wenn man Spaß an der Bastelei und am Experimentieren hat. Beim hier vorgeschlagenen Konzept sind zwar die Kosten für die Bauelemente sehr niedrig, aber man muss schon etwas Zeit investieren, bis der Laderegler zuverlässig am Fahrrad funktioniert. Dazu gehören auch die Befestigung am Rad und die Verkabelung, die nicht mehr Teil dieser Anlei-



Ladestrom i_L als Funktion der Geschwindigkeit

v_{Fahrrad}

tung sind. Der Autor wollte zunächst nur erkunden, ob die *Power aus dem Nabendynamo* grundsätzlich möglich ist. Die Ladeschaltung wurde deshalb einfach in ein kleines Plastikgehäuse gequetscht und unter die Telefonhalterung montiert. Das reichte für die Testfahrten aus, sollte aber keine endgültige Lösung sein. —esk

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.codemercs.com/shop

Die CNC-Dozentin

Mehrmals im Jahr findet an verschiedenen Orten in ganz Deutschland der CNC14-Fräsenbau-Workshop statt: Die Teilnehmer bauen in drei Tagen unter Anleitung ihre eigenen Fräsen. Wir haben der Initiatorin Birgit Hellendahl über die Schulter geschaut.

von Carsten Meyer



Kursteilnehmer können ihr eigenes (garantiert funktionsfähiges) CNC14-Exemplar in knapp drei Tagen zusammenbauen und lernen am letzten Tag auch schon erste Schritte mit der Steuerungssoftware. Die Fräse kennen vielleicht einige CNC-affine „Makende“ bereits: Sie ist die Weiterentwicklung der „Sperrholzfräse“ aus Ausgabe 1/14 der c't Hacks (der Vorgängerin des Make-Magazins) mit größeren Abmessungen. Ihre Grundmaße von 1500mm x 940mm orientierten sich anfangs am Bedarf, einen Gitarren-Korpus auf ihr fräsen zu können, erwiesen sich aber auch für sonstige Aufgaben als guter Kompromiss aus Platzbedarf und Anwendungsspektrum.

Einige Verbesserungen gegenüber dem ursprünglichen Design flossen schon recht schnell in das Design ein. Neben der größeren Arbeitsfläche ist vor allem das höhere Portal und die geänderte Führung der Zahnriemen zu erwähnen, das Flaschenzug-Prinzip blieb natürlich. Im Workshop erfahren die Teilnehmer auch, wie man Energieketten, Kamera, Staubabsaugung oder den wichtigen Not-Aus-Schalter nachrüstet.

Die Schrittmotorensteuerung basiert auf einer TB6550-Vierfach-Endstufe und einem Arduino Nano mit GRBL-Firmware. Im Rahmen der Workshops wird auch auf Sonderwünsche eingegangen, zum Beispiel einer Kamera zum Anfahren des Werkstück-Nullpunkts. Die Konstruktion zeichnet sich wie schon das Original durch ihre Einfachheit, ihre Stabilität und ihre Modifizierbarkeit aus. Sperrholz, Balsaholz, Hartholz, Styrodur, Kunststoffe wie Acryl oder POM können problemlos bearbeitet werden, mit Einschränkungen auch Aluminium. Birgit Hellendahl stellte nun eine deutlich kleinere Variante der Sperrholz-Fräse vor, und Make hatte Gelegenheit, mit ihr zu sprechen. Das Interview führte Carsten Meyer, Erfinder der Sperrholzfräse.

Make: Birgit, du entsprichst so gar nicht dem Stereotyp eines gestandenen CNC-Profis, eines stattlichen Herrn im Blaumann. Wie bist du auf das Thema gekommen?

Birgit Hellendahl: Früher habe ich Dioramen und Landschaftsmodelle gebaut, von Hand mit Säge, Dremel und Feilen. Aber die Ergebnisse waren nie so, wie ich es mir vor meinem inneren Auge vorgestellt hatte. Dadurch kam mein Wunsch nach einer computergesteuerten Fräse auf. Also begann ich, mich in das Thema CNC einzulesen, denn einfach kaufen, ohne die Technik zu verstehen, kam für mich nicht in Frage. Während meiner Recherche stieß ich unter anderem auf eine Bauanleitung von euch, die Sperrholzfräse von Carsten Meyer. Um ein erstes Verständnis zu bekommen, baute ich diese Fräse nach.

Dabei machte ich aber schon einige Modifikationen an der Konstruktion, um sie besser auf meine Bedürfnisse anzupassen. So habe ich zum Beispiel die Maße der Fräse vergrößert

und den Lauf des Riemenantriebes optimiert. Und so ist letztendlich die Urform der CNC14 dabei herausgekommen. Und damals wie heute bin ich von der Einfachheit, der Stabilität, der Modifizierbarkeit und Präzision dieser Fräse so begeistert, dass ich bei diesem Konzept geblieben bin.

Make: Klingt, als wärst du mit Technik groß geworden!

Hellendahl: Mit meinem Vater habe ich von klein auf viel gewerkelt, aber ich bin keine studierte Maschinenbau-Ingenieurin, sondern habe mich früher als Diplom-Oecotrophologin mit der Planung von Großküchen beschäftigt. Meine Leidenschaft, mit Holz zu arbeiten, war allerdings immer größer als die für Ernährungswissenschaften.

Make: Wo gab es denn die Idee, die Initialzündung zu deinen Workshops?

Hellendahl: In der Dingfabrik e.V., dem Kölner FabLab, bin ich schon seit vielen Jahren Mitglied. Meine erste funktionierende CNC sprach sich dort schnell herum. In der Folge haben mich zwei Besucher, ihnen doch beim Bau einer eigenen Fräse zu helfen, und daraus wurde noch in 2014 ein erster kleiner und noch reichlich improvisierter Workshop.

Dann nahm das Projekt seinen Lauf, weitere Interessenten kamen auf mich zu und es wurden stetig mehr. Auch ich lerne natürlich mit jedem Workshop dazu, optimiere die Abläufe und verändere auch immer wieder Details in der Konstruktion, um die CNC14 noch besser zu machen. Inzwischen sind in über 20 Workshops mehr als 140 Fräsen entstanden, die alle eine personalisierte CNC14-Gravur tragen.



Die Fräse CNC14 ging aus der Make-Sperrholzfräse hervor und erhielt einige Verbesserungen.

Make: Alle Achtung! Die ersten Kurse gab es ja nur in Köln, mittlerweile veranstaltest du sie im ganzen Bundesgebiet.

Hellendahl: Ja, die Teilnehmer kommen inzwischen auch aus Österreich, Schweiz, Italien, Slowakei, Belgien ...

Make: Hast du dabei Hilfe? Wir wissen ja aus eigener Erfahrung, wie zeitraubend das Organisieren von Veranstaltungen ist.

Hellendahl: Ich habe ein paar Mitstreiter, die nach und nach zu meinem Projekt als Unterstützer dazugekommen sind: Alois ist eine Kapazität im Fräsen-Bau und hat schon viele unterschiedliche Konzepte aus Holz und Alu erfolgreich umgesetzt. Er ist aber auch durch seine große Erfahrung in der Holz- und Metallbearbeitung die perfekte Unterstüt-



Das CNC14-Team: Alois, Birgit, Marian und Uli



Die neue Z-Achse wird aus Aluminium gefertigt.

zung, um die Teilnehmer beim Aufbau der vorbereiteten Teile mit Rat und Tat zu unterstützen.

Marian ist unser Soft- und Hardware-Spezialist. Er steckt sein Wissen und Können in die Weiterentwicklung der Steuerelektronik der CNC14. Marian fertigt eigens entworfene Platinen, mit denen die CNC14 einfacher angesteuert werden kann. Und nicht zuletzt Uli. Er unterstützt uns besonders im Hintergrund durch Ideen und die Fertigung von Spezialwerkzeugen und Messlehren, um die Genauig-

keit beim Aufbau der Fräse im Workshop nochmal zu steigern. So ist ein wirklich tolles Team entstanden, wir ergänzen uns wunderbar.

Make: Welche Vorkenntnisse brauchen deine Teilnehmer? Was sind das für Leute?

Hellendahl: Alles, was man zum Bau und Bedienung der Maschine wissen muss, lernt man innerhalb des Kurses. Zum Workshop sollten ein Akkuschauber und ein Laptop mitgebracht werden. Die Bauteile der Fräse sind vorgefertigt, also passgenau gesägt oder gefräst. Vorkenntnisse sind eigentlich nicht erforderlich, Erfahrung mit dem Werkstoff Holz aber sicher vorteilhaft. Die ist in der Regel vorhanden.

Die Teilnehmer setzen sich aus einer bunten Mischung von Schreibern, Hobbybastlern, Modellbauern, Gitarrenbauern, Makern, Künstlern und Designern zusammen. Das macht aus jedem Workshop ein ziemlich interessantes Wochenende. Jeder bringt einen eigenen Input aus seinem speziellen Bereich mit. So ist inzwischen über die Jahre eine lebhaft Community entstanden. Und die hält über das jeweilige Wochenende hinaus, so dass Erfahrungen zum Beispiel in unserem CNC14-Forenbereich miteinander geteilt werden. Unsere Freude ist, wenn sich die Teilnehmer bei uns wohlfühlen.

Make: Machst du im Workshop auch etwas in Sachen CAD- und CAM-Software?

Hellendahl: Wir geben den Teilnehmern kostenfreie Programme an die Hand. Im Workshop zeigen wir die ersten Schritte zur Erstellung einer Zeichnung und geben Tipps zu Programmen. So geben wir eine Basis, auf die der Teilnehmer nach dem Workshop aufbauen kann. Das Lernen beginnt anschließend und



Alles dabei: Die Fräsen-Workshops können auch in größeren Seminarräumen stattfinden.

endet nie. Jedes neue Projekt ist eine neue Herausforderung mit neuen Anforderungen. Ich sage immer: Man nimmt sich keine neue Maschine mit nach Hause, sondern ein neues Hobby!

Make: Ich habe gesehen, dass du die bewährte Z-Achse aus unserem Bauvorschlag nicht mehr verwendest.

Hellendahl: Seit Ende 2019 haben wir unsere neue Z-Achse, die wir im Team und mit der Unterstützung unseres Zulieferers Kamp & Kötter entwickelt haben. Damit konnten wir den Z-Verfahrweg auf 140mm verdoppeln, was das Handling bei dicken Werkstücken enorm erleichtert. Eigentlich waren wir mit der alten Z-Achse ganz zufrieden, aber sie ist schlicht nicht mehr lieferbar. Ideen für weitere Modifikationen und Erweiterungen gehen uns so schnell nicht aus.

Make: Wie zum Beispiel für das kleinere Modell ...

Hellendahl: Gestatten: „Dötzchen“. Die ist geeignet für alle, die nicht so viel Platz haben. Ihr Arbeitsbereich umfasst 290mm x 600mm und 120mm in der Höhe, die Werkstücke können eine Breite bis 390mm haben. Dötzchen wird wie auch ihre Vorgänger aus Multiplex-Platten, wellenunterstützten Linearschienen, drei NEMA23-Schrittmotoren und einer Menge Schrauben gebaut und erhielt eine etwas kleinere Z-Achse. Wir wollten sie schon auf der Maker Faire Ruhr vorstellen, aber die fiel ja leider aus.

Make: Apropos: Wie erging es dir während der Corona-bedingten Einschränkungen?

Hellendahl: Natürlich wurden einige Veranstaltungen abgesagt, das war schon schade. Eine Zeit lang waren keine Workshops möglich und Anfragen seltener, aber dann wurde es den Leuten anscheinend langweilig und viele haben sich wieder angemeldet. Ende Mai durften wir in Hamm wieder loslegen. Den Kurs haben wir dann zweigeteilt, um die Personenzahl zu begrenzen und den Abstand gewährleisten zu können. Termine werden immer unter <https://cnc14.de/termine> bekannt gegeben.

Make: Birgit, wir danken dir für die Einblicke! —cm



Für Anwender mit begrenztem Platz ist „Dötzchen“ gedacht, die kleine Schwester der CNC14.



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

ct

ct.de/angebot

Jetzt gleich bestellen:

 ct.de/angebot

 +49 541/80 009 120

 leserservice@heise.de

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

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

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

© Copyright by Maker Media GmbH.



Animatronik

Sie sehen wie Menschen, Tiere oder Fabelwesen aus, bewegen sich am Schnürchen wie Marionetten, sind im Kern aber ausgekugelte Roboter: Animatronische Figuren haben vor allem in Filmen, Museen und Vergnügungsparks ihren großen Auftritt, sind aber auch für Maker ein reizvolles Projekt.

von Daniel Springwald

Animatronik selber machen

Sprechende Tesla-Büste im Eigenbau **Seite 70**

Animatronischer Minion für Einsteiger **Seite 78**

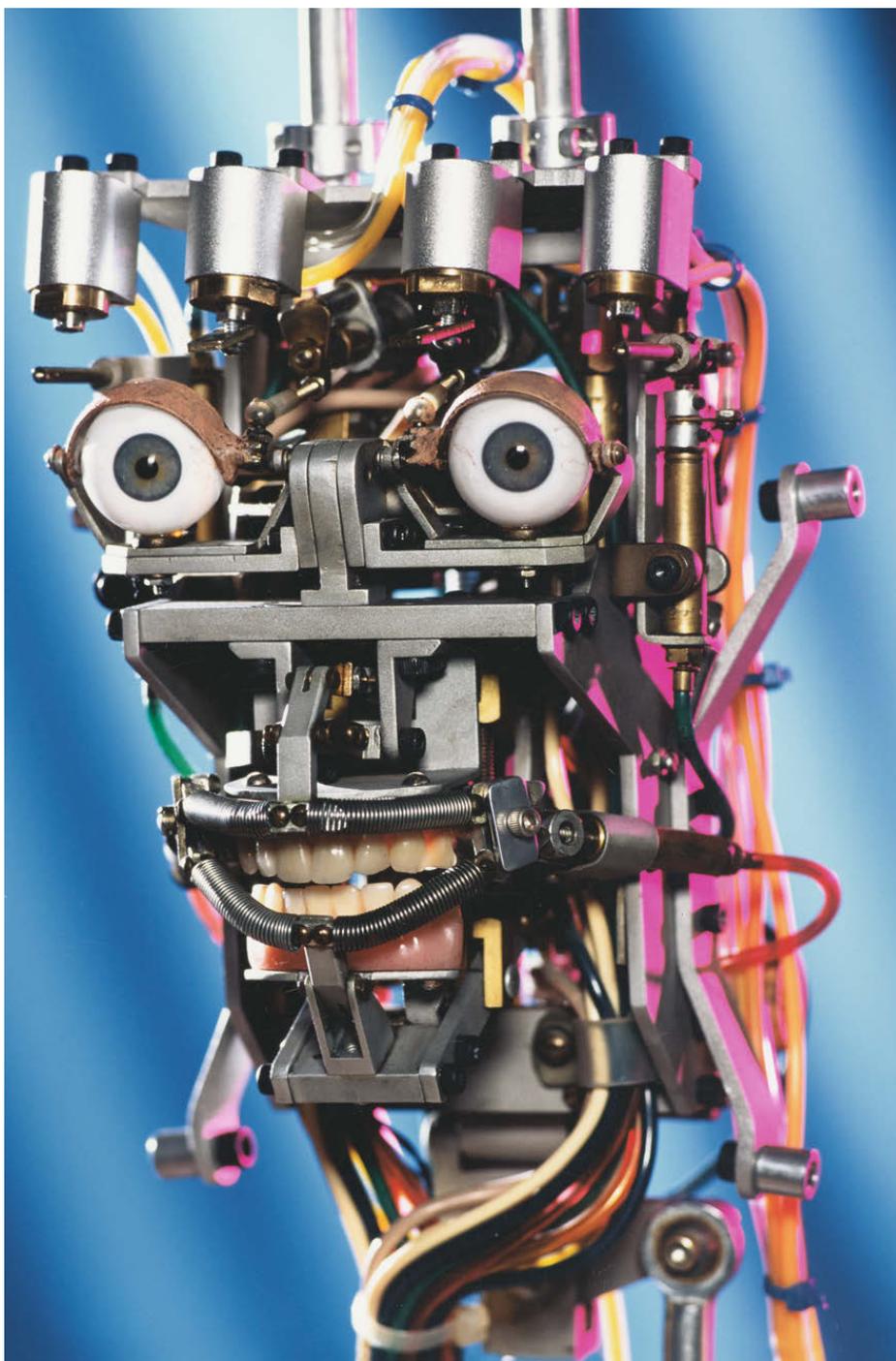


Bild: Courtesy of Garner Holt Productions, Inc.

In der Animatronik geht es darum, künstliche Figuren ferngelenkt oder programmgesteuert, aber möglichst lebensecht zu bewegen – ganz gleich, ob es sich um ein kleines Tier, ein Alien-Monster oder auch eine menschliche Figur handelt. Animatronische Figuren werden entweder live von einem Puppenspieler gesteuert oder die Bewegungen werden nur einmal aufgezeichnet oder programmiert und dann beliebig oft abgespielt.

Je nach Einsatzzweck muss übrigens nicht die komplette Puppe animatronisch sein – teilweise steckt auch ein Mensch in einem Kostüm und bewegt den Körper, während die Mimik des Kopfes von außen ferngesteuert wird. Ein bekanntes Beispiel dafür sind die Figuren der Fernsehserie „Die Dinos“ aus den 1990er Jahren, was man in einem YouTube-Video gut sieht. Sie finden es über die die Kurz-URL am Ende des Artikels – ebenso wie zahlreiche andere Videos der erwähnten Figuren, da sich ihr Reiz erst dann vollends erschließt, wenn man sie in Aktion sieht.

Einsatzgebiete

Animatronik trifft man heute hauptsächlich in Freizeitparks und auf Filmsets an, letzteres zum Beispiel bei Spezialeffekten: Wo Verletzungen oder Körpermodifikationen der Schauspieler die reine Maskenbildnerei überfordern, werden oft bewegte Puppen verwendet. Diese müssen meist nur aus bestimmten Blickwinkeln realistisch wirken oder es werden nur einzelne Körperteile gezeigt. Unter Umständen kommen deshalb für verschiedene Kameraeinstellungen mehrere unterschiedliche Modelle derselben Filmfigur zum Einsatz.

Animatronik wird in Filmen darüber hinaus aber auch eingesetzt, um ganz eigenständige Darsteller zu erschaffen und zum Leben zu erwecken. Ein Beispiel dafür ist die Figur „Teddy“ aus dem Film „AI Artificial Intelligence“ (Video siehe Kurzinfo). Durch immer leistungsfähigere Computer ersetzt zwar heute oft computergenerierte Grafik (CGI) die Animatronik im Kino, der Anteil „echter“ Figuren bei Dreharbeiten ist aber – je nach Film – immer

noch überraschend hoch. Denn auch die Animatronik hat sich in den letzten Jahren weiter entwickelt und ein so hohes Niveau erreicht, dass sie an vielen Stellen durchaus mit Computergrafik mithalten kann.

Animatronische Puppen werden beim Film naturgemäß nur für einen begrenzten Zeitraum verwendet und nach den Dreharbeiten meist nicht mehr benötigt. Sie müssen daher nicht besonders verschleißfrei konstruiert werden. Teilweise werden sie wieder zerlegt, um die Bauteile für neue Projekte zu verwenden, oder während der Dreharbeiten als „Stunt-Double“ sogar zerstört.

In Freizeitparks sind animatronische Figuren hingegen meist im Dauereinsatz und spielen dort im Minutenabstand immer wieder dieselbe Szene vor. In Konstruktion und Aufwand gleicht ihr Innenleben inzwischen nicht selten dem von Industrie-Roboterarmen.

Obwohl es sich um ein eher spezielles Metier handelt, gibt es zahlreiche kommerzielle Hersteller für Animatronik. Zu den bekannteren zählen zum Beispiel im Filmgeschäft *Jim Henson's Creature Shop* und *Weta Workshop*, für den Bedarf von Freizeitparks liefern etwa *Disney* und *Garner Holt Productions* die passenden Figuren.

Es entstehen aber auch immer mehr animatronische Figuren, die nicht nur rein vordefinierte Bewegungen abspielen, sondern mit den Besuchern interagieren. Hier kann man bereits von „interaktiven Robotern“ sprechen. Beispiele gibt es etwa auf der Webseite der britischen Firma *Engineered Arts* zu sehen. Auch der interaktive Roboterkopf SEMMI der Deutschen Bahn fällt in diese Kategorie (siehe Link).

Technik

Animatronik macht ein Zusammenspiel verschiedener Gewerke wie Maskenbildnerie, Modellierung, Maschinenbau, Elektronik und Programmierung notwendig.

Es können verschiedene Antriebssysteme eingesetzt werden. Eine der ersten Steuerungen für Figuren beim Film erfolgte über Drahtseilzüge. Dabei sitzt der Puppenspieler versteckt außerhalb des Sichtbereiches und bewegt mit seinen Händen einen Joystick oder eine Kopie der Puppe. Diese Bewegungen werden dann über ein Seilzugsystem auf die Figur übertragen. Als Weiterentwicklung werden inzwischen auch hydraulische oder pneumatische Systeme als Antrieb benutzt. Solche animatronische Figuren agieren meist ortsgebunden und sind nicht sehr beweglich, da sie über einen Strang Kabel oder Schläuche mit einer Steuereinheit verbunden sind.

Soll die Puppe frei im Raum bewegt werden, kann man Komponenten aus dem RC-Modellbau verwenden – in die Figur baut man dann Servos ein, die man über eine ganz normale Funk-Fernsteuerung kontrolliert, nur



Bild: Courtesy of Garner Holt Productions, Inc.

Dieser animatronische Kopf von Abraham Lincoln kann beeindruckend viele Emotionen darstellen (Video siehe Link am Ende des Artikels).

dass ein Knüppel des Senders dann die horizontale Bewegung der Augen steuert statt das Ruder eines Flugzeugmodells. Auch dieses Steuerungsprinzip lässt sich noch weiter professionalisieren, etwa über stärkere elektrische Motoren unterschiedlichster Arten.

Je nach Konstruktion und eingesetzter Technik kann Animatronik durchaus die Lautstärke eines Industrie-Roboterarmes entwickeln. Das stellt aber meist kein Problem dar: Beim Film kann dieser Ton herausgeschnitten werden und in Vergnügungsparks wird die Lautstärke durch entsprechenden Abstand zum Publikum und laute Umgebungsgeräusche oder Musik übertönt. Es gibt zudem moderne Figuren, die durch speziell entwickelte Motoren nahezu geräuschlos arbeiten.

Um das Gesicht einer animatronischen Figur zu bewegen, ist ein feinmechanisches System mit mehreren Kleinmotoren nötig. Gerade bei den Augen und Augenlidern kann es anspruchsvoll sein, diese Konstruktion lebensecht, aber verschleißfrei zu gestalten. Als künstliche Haut kommt meist Gummi oder Silikon zum Einsatz, das aufwändig mit Haaren ausgestattet und Schminke behandelt wird. Mit der Zeit und im permanenten Einsatz können diese Materialien Alterungserscheinungen wie Risse aufweisen. Deshalb und wegen der aufwändigen Mechanik wird in Freizeitparks immer häufiger darauf verzichtet, das Gesicht der Figuren motorisch zu bewegen. Statt dessen setzt man kleine Farbbildschirme für die Augen ein oder stellt das komplette Gesicht per Videoprojektion dar. Das funktioniert vor allem dann gut, wenn der Abstand zum Betrachter entsprechend groß ist. Früher geschah das durch externe Film-Projektoren – so etwas konnte man zum Beispiel in der Geisterrikscha des *Phantasialands* seit den 1980er Jahren bestaunen. Miniatur-Beamer machen es inzwischen aber auch möglich, das Bild von innen auf ein Gesicht aus transluzentem Material zu projizieren. Dadurch muss der Kopf nicht mehr auf den Projektor ausgerichtet sein, sondern kann sich frei bewegen.

Animatronik in Spielzeugen

Der eine oder andere kennt vielleicht noch das Spielzeug Furby aus den 1990er Jahren. Auch dieses elektronische Plüschtier kann guten Gewissens bereits als animatronische Figur bezeichnet werden, da es Augen, Augenlider, Mund, Ohren und Körper bewegen kann. Zudem ist es noch eine technische Meisterleistung, da all dies mit nur einem einzigen Motor bewerkstelligt wurde. In den neueren Furby-Generationen wurden die Augen durch kleine Graustufen-Bildschirme ersetzt – bei der in Deutschland wenig bekannten Version *Furby Connect* sind diese sogar in Farbe.

Solche Spielzeuge mit animatronischen Komponenten zu hacken ist auch ein beliebter Maker-Sport – und von da aus ist der Weg zum ersten eigenen animatronischen Projekt gar nicht mehr weit. Wie man so etwas selbst umsetzen kann, zeigen unsere beiden folgenden Artikel. —pek

Videos der Figuren im Web unter make-magazin.de/xerw



Ein moderner Furby mit Bildschirm-Augen

Animatronischer Tesla

Beim Wort „Tesla“ denken heute viele nur an eine Automarke, wissen aber wenig über den namensgebenden Erfinder. Jetzt kann sich Nikola Tesla selbst zu Wort melden – in Form dieser lebensgroßen, animatronischen Büste.

von Daniel Springwald



Animatronische Büsten zu bauen, begeistert mich bereits seit den 1990er Jahren. Damals als Schüler waren die Möglichkeiten allerdings sehr bescheiden: Die Motoren wurden per C64 gesteuert und die künstliche Haut aus Baumarktsilikon in der Farbe „Bahama-beige“ geformt. Seitdem sind die Möglichkeiten durch 3D-Druck, weltweiten Versand von Spezialmaterial wie Kunsthaut und günstige Elektronik sehr viel größer worden.

Meine Motivation

Meine bisherigen Animatronik-Projekte waren inspiriert von Filmfiguren wie *RoboCop* oder *Edward Scissorhands*. Für diese Büste diente mit Nikola Tesla erstmals eine reale Person als Vorlage. Seit der Lektüre seiner Biografie vor einigen Jahren begeistert mich dieser Erfinder, der vielfach überraschend unbekannt ist und dessen Name aktuell eher mit Elektroautos assoziiert wird, als mit seinen bahnbrechenden Erfindungen im Bereich des Wechselstroms. Zudem werden viele seiner Ideen und Errungenschaften in der allgemeinen Wahrnehmung oft fälschlicherweise anderen Personen zugeschrieben. Das brachte mich auf die Idee, eine lebensgroße Büste Teslas zu bauen, mit der man sich über dessen Leben und seine Erfindungen unterhalten kann. Und „unterhalten“ ist tatsächlich wörtlich gemeint: Man kann die Büste ansprechen, wird von ihr angeschaut und hört eine Sprachantwort.

Hals und Nacken

Kern der Büste ist ein mechanisches Skelett, welches das Gewicht des Kopfes trägt und die Bewegungen von Nacken und Hals umsetzt. Damit diese Bewegungen natürlich aussehen, sollte die Geschwindigkeit nicht zu gering sein. Gleichzeitig hat der Kopf mit einem Gewicht von 2,4kg auch eine gewisse Massenträgheit. Die Konstruktion muss daher sowohl stabil als auch kräftig ausfallen.

Für die Neigung des Kopfes dient ein Kardangelen. In einer ersten Version bestand dieses Halsgelenk aus kleinen Kugellagern, die in einer 3D-gedruckten Halterung eingefasst waren. Der eingesetzte PLA-Kunststoff erwies sich aber für die Belastung langfristig nicht als optimal. Die aktuelle Version des Kardangelenks besteht daher primär aus stabilem Holz und Makerbeam-Aluminium-Profilen.

Für die Neigung des Halsgelenks sind grundsätzlich nur zwei Servos nötig. Die Drehbewegung des Kopfes nach links und rechts geschieht durch einen dritten Servo, der oben auf dem Gelenk montiert ist.

Die Servos

Normale Modellbau-Servos haben in der Regel eine feste Geschwindigkeit, mit der sie

Kurzinfo

- » Lebensgroße, animatronische Büste des großen Erfinders
- » Computergesteuert mit Sprach- und Gesichtserkennung
- » Komplette Eigenkonstruktion mit neun Bewegungsachsen

Checkliste

-  **Zeitaufwand:**
etwa 10 Monate
-  **Kosten:**
etwa 1800 Euro
-  **Holzbearbeitung:**
sägen, bohren, schleifen, kleben, lackieren
-  **Löten:**
elektronische Komponenten zu einem komplexen System verbinden
-  **Entwerfen:**
3D-Druckvorlagen konstruieren mit OpenSCAD
-  **3D-Druck:**
Einzelteile aus PLA drucken, nachbearbeiten, lackieren
-  **Programmieren:**
Arduino, C#, OpenCV, Artificial Intelligence Markup Language (AIML)

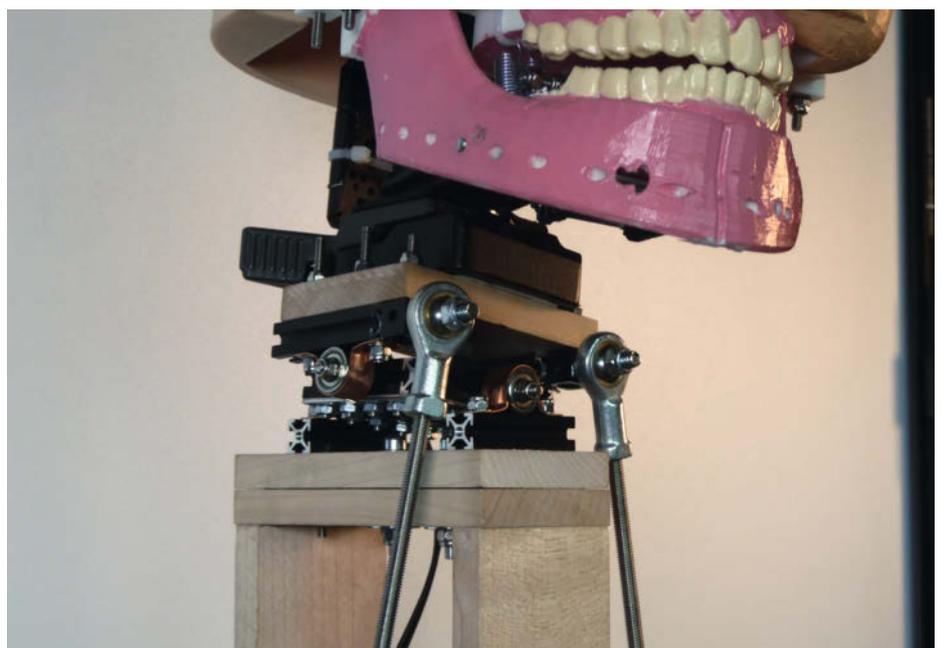
Material (Auswahl)

- » Silikon-Maske
- » Perücke, Wimpern und Kunstbart
- » künstliche Augen
- » 9 Dynamixel-Servos mit Controller und Netzteil
- » Lüfter zur Kühlung der Servos
- » 3D-gedruckte Teile
- » Montagematerial wie Holz, Kugellager, Kabel, Makerbeam-Profile
- » Lautsprecher und Räummikrofon
- » Mini-Touchscreen, Webcam, Bewegungsmelder
- » USB-Docking-Station mit Sound- und Grafikkarte
- » externer PC per USB verbunden
- » Arduino Uno
- » Regalbrett Ikea Lack als Grundplatte
- » Neopixel-Band unter der Grundplatte

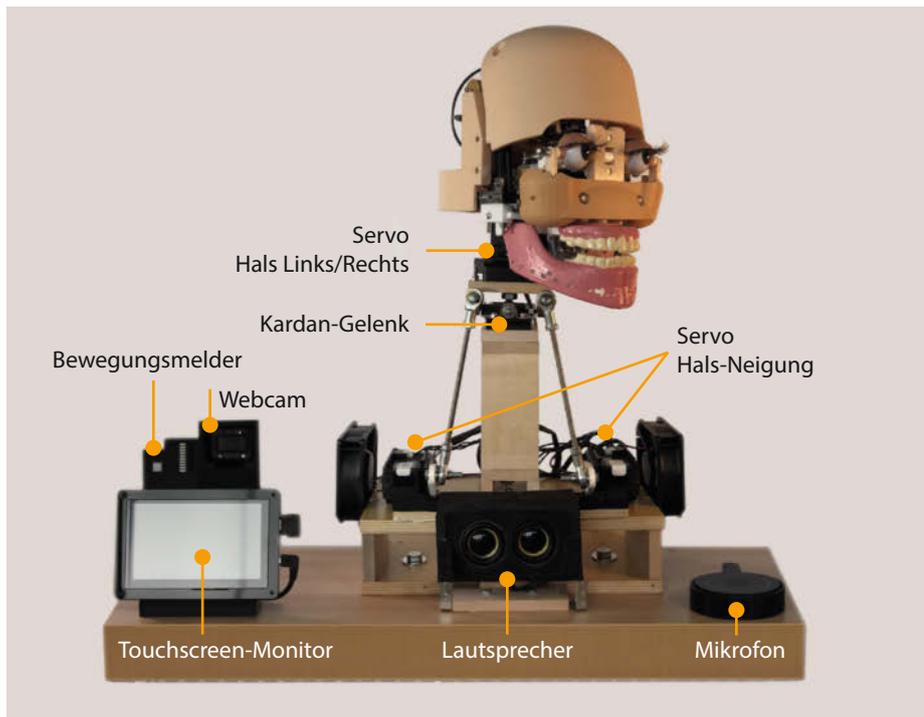
Mehr zum Thema

- » Florian Schäffer, Servos für jeden Zweck, Make 3/18, S. 46
- » Florian Schäffer, Dynamixel: Die Profi-Servo-Liga, Make 3/18, S. 54
- » Imen Mguedmini, Bobby – Sitzhaltungsbot mit Persönlichkeit, Make Sonderheft „Robotik“ 2019, S. 76

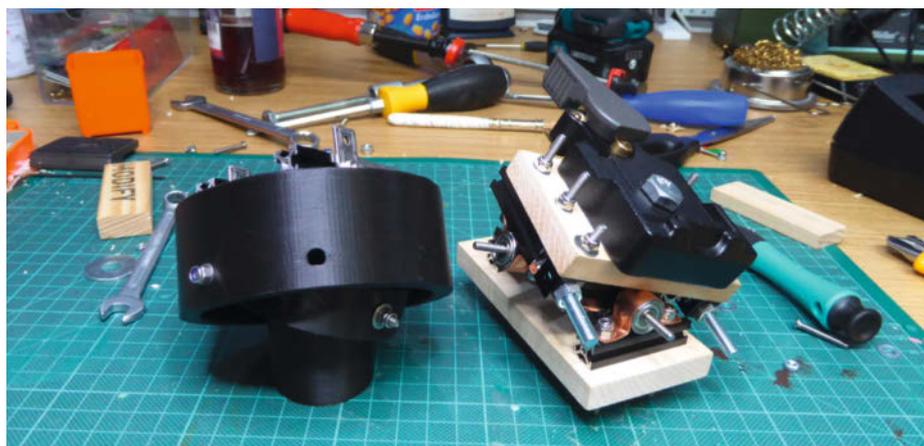
Alles zum Artikel im Web unter make-magazin.de/xu1r



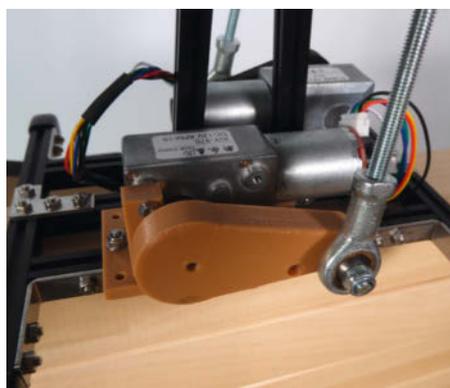
Das Halsgelenk muss den Kopf tragen und bewegen können.



Das Innenleben der Büste

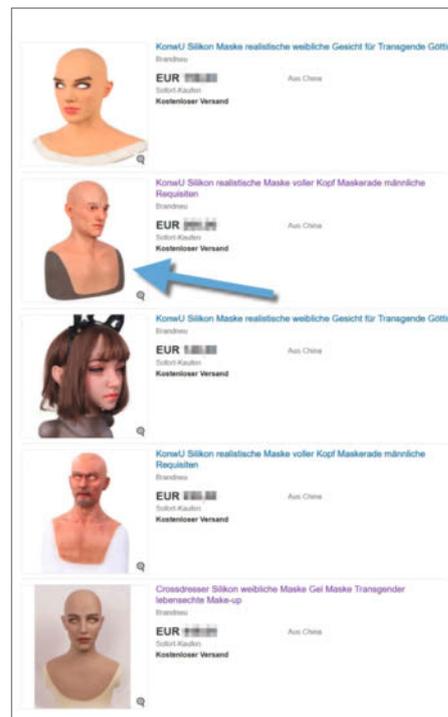


Beide Versionen des Hals-Kardangelenks. Links die erste Version aus 3D-gedruckten Teilen, rechts die solidere Variante aus Holz und Metall.



Der Eigenbau-Servo entstand auf Basis einer beliebigen Motor-Getriebe-Kombination, die in vielen mobilen Robotern die Räder antreibt.

möglichst schnell den Zielwinkel anfahren. Zum Bewegen von animatrischen Figuren sind sie deshalb nur bedingt geeignet. Wenn etwa der Kopf zu schnell bewegt wird, könnte die Halskonstruktion Schaden nehmen und die Bewegungen sehen unnatürlich aus. Dieses Problem lässt sich umgehen, indem nicht direkt der Zielwinkel an den Servo gesendet wird, sondern die Bewegung per Software durch kontinuierliche Anpassung des gewünschten Winkels über die Zeit interpoliert wird. Beim Einschalten fahren die Servos allerdings dennoch einmal sehr schnell auf ihre Standardposition. Wenn diese sich stark von der aktuellen Position des Servos unterscheidet, wird der Kopf hier wieder unnatürlich abrupt bewegt.

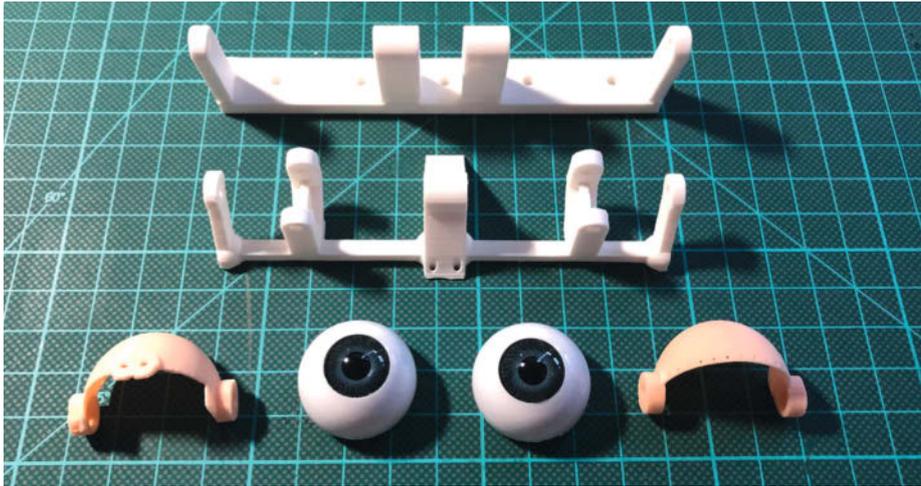


Täuschend echte „Haut“ gibt es im Internet.

Eine mögliche Lösung dafür sind serielle Servomotoren. Bei herkömmlichen Servos erfolgt die Steuerung auf einer Einbahnstraße per Pulsweitenmodulation (PWM), ohne dass es eine Rückmeldung über den aktuellen Zustand gibt. Digitale Servos, die man in Serie hintereinanderschaltet, können hingegen bidirektional kommunizieren und so ihre Position, Temperatur und andere Telemetrie-Daten zurücksenden. Dadurch ist beim Start die echte Position des Servos bekannt, man kann ihn von dort aus langsam und kontrolliert bewegen. Zudem kann man vielen seriellen Servos per Befehl sogar eine Beschränkung der maximalen Geschwindigkeit vorgeben und spart sich die Interpolation per Software.

Leider kosten serielle Servos ein vielfaches von normalen Modellbau-Servos gleicher Leistung und Qualität. Es schien mir daher einen Versuch wert, für den Hals der Büste einen eigenen Antrieb zu konstruieren. Als Grundlage dient dabei zwei der für den Bau einfacher Fahrroboter so beliebten JGY-370-Motoren mit einer Getriebe-Untersetzung auf 60 Umdrehungen pro Minute bei 12 Volt. Die Geschwindigkeit kann dabei per Motorsteuerung geregelt und der aktuelle Winkel durch eingebaute Dreh-Encoder ermittelt werden. Für die einmalige Eichung dient pro Servo-Horn ein Hall-Sensor.

Unglücklicherweise war der JGY-370-Motor in dieser Getriebeuntersetzung zu schwach und in der anschließend ausprobierten Variante 10 Umdrehungen pro Minute hingegen zu langsam für realistische Bewegungen. Also musste ich schließlich doch auf serielle Servos



Die künstlichen Augen wurden zugekauft, die ganze filigrane Mechanik drumherum ...

... stammt aus dem 3D-Drucker.

zurückgreifen. Da zur Bewegung der Augen und des Mundes inzwischen bereits Servos vom Typ Dynamixel MX12W zum Einsatz kamen, war es naheliegend, beim Hals deren „größeren Bruder“ MX64AT zu benutzen. Dadurch konnte für die Ansteuerung aller Servos dann dasselbe Bussystem verwendet werden, was die Programmierung vereinfacht.

Die Haut

Das Herstellen von Abdruckformen und Gießen eines Silikonkopfes ist eine Kunst für sich, die spannend zu erlernen wäre – daran habe ich mich bisher aber noch nicht gewagt. Glücklicherweise kann man entsprechende Ganzkopf-Masken aber bereits fertig erwerben. Wichtig ist dabei, dass diese aus Silikon bestehen - "einfache" Karnevalsmasken aus Latex eignen sich meiner Erfahrung nach nicht so gut. Gut gemachte Silikonmasken sind leider nicht billig. Selbst direkt in China bestellt kosten sie schon über 200 Euro und nach oben hin sind bis zu vierstelligen Eurobeträgen kaum Grenzen gesetzt. Dafür gibt es sie aber in nahezu jeder Form und Farbe und auch bereits mit Gesichtsbehaarung wie Augenbrauen oder Bart versehen. Für meine Nikola-Tesla-Büste konnte ich als Grundlage eine halbwegs neutrale, männliche Maske erwerben.

Die Augen

Während beim Nacken beziehungsweise Hals primär die Kraft für die Bewegung zur Verfügung stehen muss, sollte die Mechanik der Augen möglichst klein, schnell und exakt sein. Bei dieser Büste ist die Aufhängung der Augen mit der Software OpenSCAD designt, 3D-gedruckt und wird mit fünf Servos angetrieben. Lediglich die Augäpfel mit den Pupillen sind als fertiges Plastikbauteil dazugekauft.

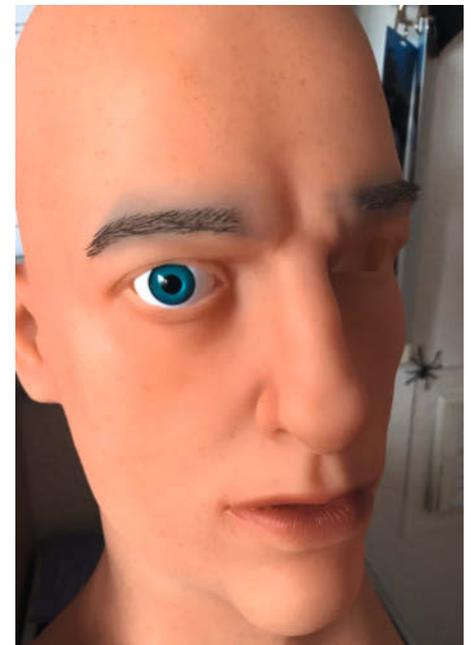
Für ein natürliches Erscheinungsbild sollte jedes Auge drei Bewegungen unterstützen:

Drehung links/rechts, auf/ab und auf/zu. Wenn man beide Augen mechanisch miteinander verbindet, kann man dies bereits mit insgesamt drei Servos für beide Augen gemeinsam umsetzen. Dann wäre aber keine Feinabstimmung per Software möglich, das heißt, beide Augen müssen rein mechanisch immer identisch weit geöffnet sein und man müsste sich für eine Entfernung entscheiden, auf die die Augen scheinbar fixiert sind. Mir war es wichtig, dass die genaue Stellung der Augen auch im Betrieb noch kurzfristig über die Programmierung angepasst werden kann. Daher ist bei Nikola Teslas Büste nur die Auf/ab-Bewegung der Augen über einen gemeinsamen Servo realisiert, daneben verfügt jedes Auge über zwei eigene Servos für die Links/rechts- und Auf/zu-Bewegung.

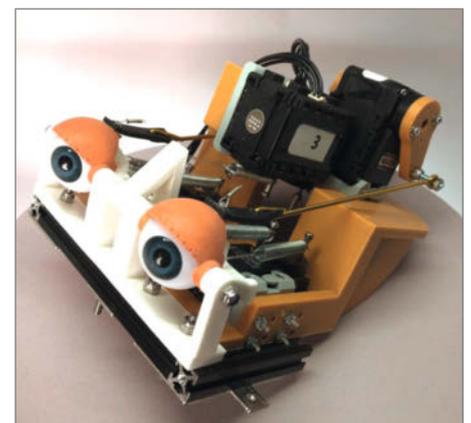
Zum Schließen der Augen wäre es optimal, wenn die Augenlider oben und unten noch Bestandteil der Haut der Maske wären. Dann müsste das Material aber sehr dehnbar sein und direkt mit den Servos verbunden werden können, ohne dass diese Verbindung später abreißt. Bei der von mir gekauften Maske war dies leider nicht mehr möglich, da deren Lider bereits herausgeschnitten waren.

Die Augenlider sind daher als getrenntes 3D-Druck-Bauteil in Form eines hohlen Kugelausschnittes umgesetzt. Zum Schließen des Auges wird die Schale an der horizontalen Achse gedreht und damit vor den Augäpfeln heruntergeklappt. Hier ist die Passform sehr wichtig, damit sich die Lider zum einen frei bewegen können, ohne die Haut und oder die Augäpfel zu berühren. Zum anderen sollte der Abstand aber nicht zu groß sein, damit die Augen nicht insgesamt zu tief in die Augenhöhlen rutschen.

Zunächst hatte ich befürchtet, dass diese Konstruktion wie Klimperaugen bei einer Kinderspielpuppe wirkt. Aber durch das passgenaue Einsetzen und das Bemalen mit einem Farbton möglichst nah an der Farbe der künst-



Die künstlichen Augäpfel in Kombination mit der Silikonhaut verbergen gut, ...



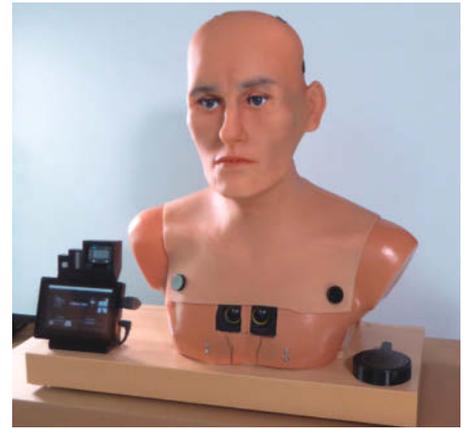
... dass die Augen im Inneren von insgesamt fünf Servos bewegt werden.



Im Mund sitzt ein einzelner Servo.



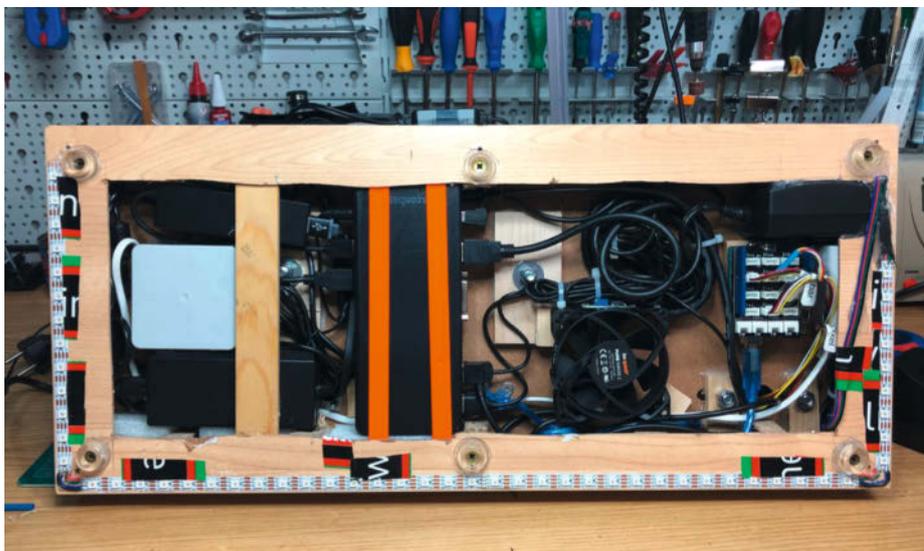
Die Haare werden streifenweise angeklebt.



Alles wird mit Magneten am Körper befestigt.



Nicht alles passt direkt beim ersten Versuch – im Lauf der Entwicklung sammelt sich eine Menge an Ausschussteilen an.



Das Innenleben der Basis

lichen Haut fällt es kaum auf, dass die Augenlider nicht Bestandteil der Haut sind.

Der Mund

Um aufwändige, animatronische Bewegungen möglich zu machen, wären – ähnlich wie bei den Augen – beim Mund ebenfalls mehrere Servos sinnvoll. Aus Platzgründen konnte ich die Mundbewegung in der aktuellen Konstruktion leider nur mit einem einzelnen Servo lösen. Der Mund kann sich daher ausschließlich öffnen und schließen. Ein Spitzener der Lippen oder ein breites Lächeln können damit (zur Zeit) noch nicht dargestellt werden.

Obwohl die Mechanik des Mundes damit nur aus einem Scharnier besteht, funktionierte hier nicht alles auf Anhieb. Der eingesetzte Servo schien zwar bei ersten Tests zunächst mehr als genug Kraft zu besitzen, denn er musste zunächst in seiner maximalen Leistung beschränkt werden, damit die Zähne nicht zu hart aufeinander schlagen. Nachdem dann im nächsten Schritt aber die Haut am Kopf angebracht war, bot sich bereits ein anderes Bild: Jetzt musste der Servo zusätzlich das Gewicht von Kinn und Hals mit anheben. Das funktionierte gerade noch, aber nur, bis die Haut der Maske unten an Brust und Sakko fixiert wurde. Plötzlich musste der Servo zusätzlich die künstliche Haut am Hals dehnen. Da der Mund die meiste Zeit geschlossen ist, bedeutet dies permanentes Nachregeln und damit Kraftaufwand für den Servo. In der Folge überhitzte er nach kurzer Zeit und schaltete sich ab. Das Problem konnten aber Zugfedern innerhalb des Mundes beheben, die den Unterkiefer immer wieder nach oben ziehen und damit den Servo unterstützen.

Die Haare

Erste Versuche, als Frisur einfach eine fertige Perücke zu verwenden, sahen nicht überzeugend aus und das ganze verrutschte schnell. Zudem wollte ich den Kopf zum Erreichen der

Augenmechanik schnell öffnen können, ohne dass anschließend die Haare aufwändig wieder aufgesetzt und frisiert werden müssen.

Als Lösung besteht der mit Haaren bedeckte Teil des Kopfes nun aus einer festen Form, die aus Drahtnetz und Leicht-Modelliermasse gefertigt ist. Günstige Kunststoffperücken sind aus einzelnen Streifen hergestellt, die man mit einer Schere wieder gut zerteilen kann. Klebt man solche Streifen Reihe für Reihe von unten nach oben auf die "Kopfhaut", entsteht am Ende durchaus realistische Haare, die man sogar frisieren kann.

Die Kleidung

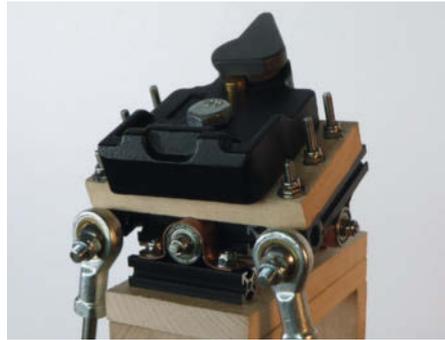
Da die Büste lebensgroß ist, war die Kleidung verhältnismäßig einfach zu beschaffen. Das Sakko konnte ich auf einem Flohmarkt kaufen; das Hemd mit Vaternörder (steifer Stehkragen) im Internet ersteigern. Beides sollte möglichst plan auf der Bodenplatte der Büste aufliegen und bei Bewegungen nicht hochrutschen. Dafür ist in die unteren Säume von Hemd und Sakko Bleiband eingnäht. Damit die Kleidung und Haut insgesamt die Form eines menschlichen Oberkörpers erhalten, ist darunter der Korpus einer Schaufensterpuppe aus Kunststoff eingebaut. An diesem sind die Kleidung und die Silikonhaut des Kopfes mit Magneten befestigt.

Die Basis

Der Sockel besteht aus einem Regalbrett von Ikea. Da die Teile aus der Lack-Serie zwar dick aussehen, aber praktisch hohl sind, finden in diesem „Brett“ Netzteile, eine USB-Docking-Station mit USB-Hub, Grafikkarte und Soundkarte sowie ein Arduino Uno Platz. Durch diese integrierten Komponenten hat die Büste lediglich einen Strom- und USB-Anschluss nach außen. Gleichzeitig bekommt die Basis dadurch das notwendige Gewicht, um bei den Bewegungen des Kopfes nicht zu wackeln. Unter dem Sockel sind noch RGB-LED-Streifen angebracht, mit denen zusätzlich noch Zustände angezeigt oder Aufmerksamkeit erregt werden kann.

Modularität

Bei der Hobby-Entwicklung animatronischer Figuren funktioniert – zumindest bei mir – nicht alles gleich auf Anhieb. Man muss zum Beispiel Achsen, Motoren und Bewegungsrichtungen immer wieder überarbeiten. Die wichtigste Erfahrung, die ich daher in den letzten Jahren bei diesem Hobby gemacht habe, ist, möglichst modular zu bauen: Die Konstruktionen sind beweglich und die Einzelteile vielfältig miteinander verbunden - da geht irgendwann im Betrieb mal etwas kaputt. Deshalb ist es hilfreich, wenn man die Figur



Die aktuelle Hals-Mechanik mit der Schnellwechsellplatte, die von einem Foto-Stativ stammt



Die Kabel zum Kopf können über Steckverbindungen schnell getrennt werden.

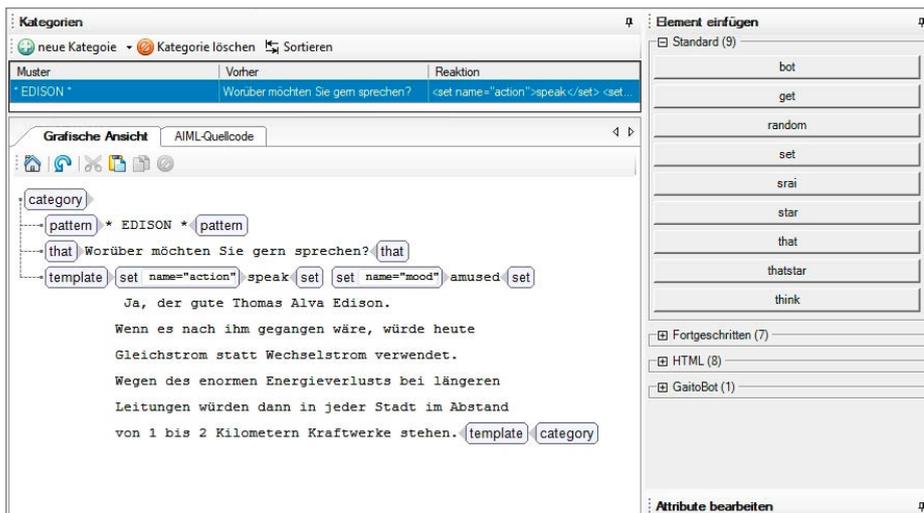
dann schnell und einfach in ihre Bestandteile zerlegen und wieder zusammenbauen kann. Meine ersten animatronischen Figuren habe ich noch an den meisten Stellen fest verbunden und zum Beispiel die künstliche Haut direkt mit den Motoren verklebt. Irgendwann

war dann einer der Motoren zur Steuerung der Augen defekt. Dieser war leider fest in der Tiefe des Kopfes verbaut und dadurch nicht zerstörungsfrei zu erreichen.

Inzwischen versuche ich möglichst durchweg Schraubverbindungen zu verwenden.

Servo	Temp	Load	Target Pos	Actual Pos
LeftEyeLid [1]	37° from 70	14 [0 - 1024]	212°	211°
RightEyeLid [2]	39° from 70	26 [0 - 1024]	157°	157°
EyesVertical [3]	35° from 70	-	150°	152°
RightEyeHorizontal [4]	38° from 70	-	165°	168°
LeftEyeHorizontal [5]	39° from 70	-	177°	180°
Mouth [7]	44° from 75	128 [0 - 1024]	144°	142°
NeckLeftRight [6]	31° from 80	32 [0 - 1024]	190°	189°
NeckUpDown1 [10]	34° from 80	144 [0 - 1024]	145°	146°
NeckUpDown2 [11]	32° from 80	40 [0 - 1024]	201°	200°

Die Software in Aktion



Das „Wissen“ des animatronischen Nikola Tesla ist in AIML gespeichert.

Manchmal ist aber auch das noch nicht optimal. Um zum Beispiel den Kopf der Nikola-Tesla-Büste vom Hals zu lösen, musste man anfangs vier kleine Schrauben und Muttern lösen. Eine Hand muss dabei den Kopf festhalten, was das Einsetzen und Anziehen der Schrauben mit der verbleibenden Hand ziemlich herausfordernd machte. Später habe ich an dieser Stelle eine Schnellwechselplatte für Fotostative eingesetzt. Dadurch klappt das Lösen und Verbinden von Hals und Kopf jetzt mit nur einem Handgriff.

Bei der Verkabelung ist es hilfreich, modular zu arbeiten und zum Beispiel Steckverbinder zu verwenden. An der Schnittstelle zwischen Kopf und Hals sind die Verbindungskabel in der Mitte noch einmal durch Stecker unterbrochen und damit schnell zu trennen. Anderfalls liefe man Gefahr, bei der Demontage Kabel abzureißen, weil die Einzelteile noch damit verbunden sind.

Ein mehrpoliges Kabel durch eine Steckverbindung zu unterbrechen bedeutet einen gewissen Aufwand: Ein passendes Steckerpärchen muss gekauft werden und beim Anlöten muss man auf beiden Seiten die gleiche Reihenfolge der Adern beachten. Eine ansehnliche Optik bei der Isolation und Zugentlastung der Kabel hinzubekommen, kann zudem eine Kunst für sich sein.

Ich habe dafür einen Kniff für mich entdeckt: In meinem Keller finden sich viele alte, nicht mehr genutzte Verlängerungskabel für Steckerformate wie Firewire, SCSI oder Parallel Port. Wenn man bei einem solchen Kabel die Stecker an den Enden zum Kreis verbindet und es dann in der Mitte durchschneidet, erhält man ein Kabel, das bereits eine Steckverbindung besitzt. Um die Reihenfolge der einzelnen Adern muss man sich dann keine Gedanken mehr machen, es reicht, sich an den Farben der Adern zu orientieren. Je nach Anzahl der benötigten Adern funktioniert dieser

Trick auch mit aktuellen Kabelformate wie USB- oder Servo-Kabelverlängerungen. Man sollte bei noch gebräuchlichen Steckerformaten aber die Stecker unbedingt besonders beschriften, damit man das vermeintliche USB-Kabel am Ende nicht doch aus Versehen in einen PC oder ein Netzteil einsteckt.

Die Software

Die Software der Büste ist eine C#-Anwendung mit grafischer Oberfläche (WPF), läuft auf einem externen Laptop und ist in mehrere Funktionsbereiche gegliedert. Die grundlegendste Aufgabe ist dabei die Steuerung und Überwachung der einzelnen Servos. Darauf baut eine Gestensteuerung auf, die Gesten wie "blinzeln", "Person anschauen" oder "Mund synchron zur Sprachausgabe bewegen" definiert. Mehrere Gesten können zu Aktionen wie "Sage Guten Tag" oder "Leerlauf" kombiniert werden, wobei die Software immer nur eine Aktion gleichzeitig ausführt.

Auf der obersten Ebene der Software werden die Daten aus Gesichtserkennung, Spracherkennung und Bewegungsmelder interpretiert und in Aktionen umgesetzt. Für die Spracheingabe wird wie bei den bekannten Heim-Sprachassistenten Amazon Echo & Co. ein Mikrofon-Array verwendet, das vorne rechts auf dem Sockel montiert ist. Welche Antworten Nikola Tesla dabei auf Basis der erkannten Spracheingabe gibt, ist in AIML (einem Auszeichnungsformat für Chatbots) definiert. Der für die Tesla-Büste verwendete AIML-Interpreter funktioniert komplett lokal, so dass das Gesagte nicht zwecks Interpretation in die Cloud geschickt werden muss.

Ich mag AIML, da man damit einerseits leicht erste Ergebnisse erzielen kann, das Format aber dennoch geeignet ist, komplexe Dialoge zu definieren. Die Grundlage bildet hierbei immer die Übersetzung einer (Sprach)

Eingabe des Benutzers in eine Antwort des Computers. Im einfachsten Fall wird dies 1:1 abgebildet, das heißt, sowohl Eingabe als auch die Antwort werden fest vorgegeben. Mit etwas Übung kann man mittels Übersetzungs- und Normierungslisten beziehungsweise Platzhaltern aber auch Unschärfe in die Erkennung bringen. Die getätigte Eingabe muss dann das hinterlegte Muster nicht zu hundert Prozent genau treffen, sondern das System erkennt auch sinnvolle Varianten. Wenn man richtig tief in AIML einsteigen möchte, kann man damit komplexe Dialoge mit Themenschwerpunkten, Rückfragen und Verwendung externer Datenquellen erstellen.

Bei verfügbarem WLAN wird die vom AIML-Interpreter gelieferte Antwort per Sprachsynthese über einen Cloud Service erzeugt und klingt sehr natürlich. Ohne Netzwerkverbindung kann die Büste immer noch Sprache bereitstellen, der man aber anmerkt, dass sie künstlich erzeugt wurde.

Neben dem Mikrofon sind auf dem Sockel der Büste noch ein kleiner Touchscreen, ein Bewegungsmelder und eine Webcam angebracht. Um dem Besucher ein Verständnis der Funktionsweise zu vermitteln, werden auf dem Bildschirm die aktuellen Aktionen und Gesten sowie das Kamerabild der Webcam angezeigt. Darauf sind die per OpenCV erkannten Gesichter entsprechend markiert.

Auf Messen und Veranstaltungen funktioniert die Spracherkennung aufgrund der Umgebungslautstärke häufig nicht gut, daher hat die Software zusätzlich einen manuellen Modus. Dann kann man auf dem Touchscreen des angeschlossenen Notebooks auswählen, welche Antworten die Büste als nächstes geben und welche Reaktionen sie zeigen soll. Das eigenständige Verhalten wird dabei dann auf das Verfolgen von Gesichtern und Anschauen der Besucher reduziert.

Ausblick

Insgesamt bin ich mit der aktuellen Version meiner animatronischen Büste zufrieden. Als nächstes wäre es aber schön, dem Mund mehr Bewegungsfreiheit zu geben, damit Tesla zum Beispiel lächeln oder mit den Lippen einen O-Laut darstellen kann. Dabei könnte der schwache Servo durch ein leistungsstärkeres Modell ersetzt werden. Bei der Software liegt noch viel Potenzial im Chatbot-Wissen. Viele Bereiche von Nikolas Leben und Wirken sind dort aktuell noch nicht abgebildet und werden von mir nach und nach aufgefüllt.

Am meisten freue ich mich aber darauf, wenn Nikola Tesla auf seiner ersten Maker Faire auf Besucher trifft und sich mit ihnen über sein Leben austauscht. Bis dahin gibt es auf meinem Blog (siehe Link in der Kurzinfor) schon mal Videos und noch weitere Infos zur Büste zu finden.

—pek

VOICE ENTSCHEIDERTALK

Digitales Europa – Daten, Infrastruktur und Wettbewerb

16. September 2020,
14 – 18 Uhr

ONLINE- KONFERENZ

Der Austausch zwischen IT-Anwendern, Politik, Wissenschaft und IT-Anbietern ist wichtig wie eh und je. Unter dem Titel **VOICE ENTSCHEIDERTALK** organisieren VOICE, der Bundesverband der IT-Anwender e.V., und heise Events eine vierstündige digitale Konferenz. Sie setzt ihren Fokus auf die Digitalisierung Europas.

Die Anstrengungen für einen gemeinsamen Digitalen Binnenmarkt dürfen nicht nachlassen. Die von verschiedenen EU-Kommissaren vorgelegten Strategiepapiere zur Künstlichen Intelligenz, Datenstrategie oder Digitalen Souveränität dürfen nicht zur Makkulatur verkommen. Sie müssen nicht nur mit umsetzbaren Inhalten gefüllt, sondern auch im alltäglichen Umgang zum Beispiel mit Cybersecurity, Daten, Wettbewerb und den großen EU-Projekten wie GAIA X zum Leben erweckt werden.

Deshalb planen wir für den **VOICE ENTSCHEIDERTALK** vier Themenblöcke:

- Datenstrategie
- Cybersecurity
- Datensouveränität und GAIA X
- Offener Wettbewerb

Preis: 99,00 € zzgl. MwSt.



Animatronic Minion

Um eine animatronische Figur zu bauen, muss man nicht immer so viel Aufwand treiben wie bei der Tesla-Büste von Seite 70. Für wenig Geld und schon innerhalb eines Tages kann man mit etwas Basterei ein ausgemustertes Spielzeug zum Leben erwecken.

von Daniel Springwald

Bei der Auswahl der Spielzeugfigur ist ein großer Körper von Vorteil. Die benötigten Servos in einen schlanken Körper einzubauen – etwa den einer Barbie-Puppe – ist schwierig. Das hier verwendete Minion-Spielzeug bietet sich neben seinen vorteilhaften Proportionen auch dadurch zum Umbau an, dass das Gehäuse durch Schrauben verschlossen ist und somit leicht und zerstörungsfrei zu öffnen. Zudem sind der Mund, die Arme und das Auge bereits beweglich, auch wenn statt der ursprünglichen Mechanik jetzt Servos verwendet werden sollen ①.

Servos gibt es in verschiedenen Größen, die nicht alle die typische Form haben müssen, wie der große und der mittelgroße Servo auf dem Bild ②. Die Preisspanne bei der Beschaffung von Servos ist beachtlich: So bekommt man günstige Exemplare schon für fünf bis zehn Euro, man kann aber auch mehr als hundert Euro für einen optisch nahezu identischen Servo ausgeben. Dessen Qualitäten stecken im Inneren: Für den höheren Preis erhält man in der Regel mehr Drehmoment,

höhere Geschwindigkeit und/oder bessere Genauigkeit. Beim animatronischen Minion ist jedoch nicht viel Masse zu bewegen, daher genügen hier die preisgünstigen Exemplare der sogenannten 9g-Klasse. Einer der drei auf dem Bild ② gezeigten Servotypen ist tatsäch-

lich winzig - im direkten Vergleich zur Münze ist das gut zu sehen. Bauartbedingt eignet er sich im Gegensatz zu den größeren Servos gut für lineare Bewegungen, aber nicht für Drehungen, da er das oben herausragende Horn nur entlang einer Achse verschiebt.



① Die originale Mechanik der Minion-Figur: Das Auge bewegt sich, wenn man auf das Logo auf der Brusttasche der Latzhose drückt.

Einbau der Servos

Nach dem Öffnen des Minion-Gehäuses können die beweglichen Teile in Augenschein genommen werden: Bei den Armen bietet es sich an, die Schultergelenke zu entfernen und als Achse direkt einen Dreh-Servo zu verwenden. Dazu müssen die Arme jeweils mit dem kleinen Plastik-Ärmchen des Servos (dem so genannten Servohorn) verbunden werden. Diese Verbindung könnte man zum Beispiel sehr professionell mit einem selbst gestalteten Verbindungsstück aus dem 3D-Drucker herstellen. Hier soll aber eine schnelle Lösung genügen, indem Minion-Arm und Servo-Horn einfach mit etwas Draht verbunden und umwickelt werden. Anschließend wird das Ganze noch mit Heißkleber fixiert, damit es sich später nicht mehr löst **3**.

Die meisten Servos können keine 360°-Drehung durchführen, sondern decken nur einen Bereich von 90° oder 180° ab. Beim Einsetzen des Servo-Horns sollte dies berücksichtigt werden, sonst kann der Minion seine Arme nur – anatomisch eher unerwartet – schräg nach hinten bewegen. Diese Grenzen des Servos sollte man nicht durch manuelles Drehen ermitteln, denn dabei kann sein Getriebe beschädigt werden. Statt dessen ist es ratsam, alle Servos vor dem Einbauen einmal elektronisch anzusteuern und dabei die End- und Neutral-Position zu bestimmen. Das geht entweder per Fernsteuerung oder einem Arduino, wie gleich beschrieben, oder aber etwas komfortabler mit einem Servo-Tester – einem sehr nützlichen Werkzeug, das es schon für etwa 10 Euro zu kaufen gibt.

Nachdem der Arm am Servo-Horn befestigt ist, kann der Servo in den Körper geklebt werden, wobei seine Achse in etwa dem Schultergelenk der Figur entsprechen sollte.

Beim Auge und Mund können die ursprünglichen Bewegungsachsen unverändert bleiben – man muss nur die Mechanik entfernen, die bisher das manuelle Drücken der Latzhose auf die Augen übertragen hat. Hier kommen insgesamt drei der winzigen Linear-Servos zum Einsatz, um die beiden Augen-

Kurzinfo

- » Animatronisches Projekt für den Einstieg
- » Altes Spielzeug zu neuem Leben erwecken
- » Ferngelenkt oder per Arduino gesteuert

Checkliste



Zeitaufwand:

etwa ein Tag



Kosten:

10 bis 50 Euro, je nach Inhalt der Bastelkiste (ohne Fernsteuerung)



Feinwerkzeug:

Basteln auf engstem Raum



Programmieren:

einfache Arduino-Programmierung (optional)

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

Material (Auswahl)

- » Spielzeugfigur mit möglichst viel Platz im Inneren für Einbauten, idealerweise mit beweglichen Teilen oder Gliedmaßen
- » Servos Zahl und Größe je nach Figur und Vorhaben
- » Fernsteuerung mit Empfänger und Akkus/Batteriehalter
- » Arduino alternativ zur Fernsteuerung
- » Servo-Karte optional bei Arduino-Einsatz
- » Kleinteile Kabel, Draht, Zahnstocher und ähnliches

Mehr zum Thema

- » Florian Schäffer, Servos für jeden Zweck, Make 3/18, S. 46

lider und den Mund zu bewegen. Grundsätzlich könnte man auch beide Augenlider gleichzeitig mit nur einem gemeinsamen Servo steuern, das dürfte allerdings mechanisch eine Herausforderung darstellen.

An den Augenlidern sind zur Bewegung bereits kleine Plastikhebel vorhanden, an denen das Horn des kleinen Servos gut befestigt werden kann – wieder mit Draht und Heißkleber **4**. Auch beim Mund funktioniert ein kleiner, linearer Servo gut. Um diesen im Körper zu befestigen, ist zum Beispiel ein Zahnstocher hilfreich oder man lötet eine kleine Halterung aus Draht **5**.

Sind alle Servos platziert und befestigt **6**, können die Servo-Kabel nach außen geführt und das Gehäuse wieder geschlossen werden. Mindestens bei den kleinen Servos sind die

Kabel meist sehr kurz, so dass es sich anbietet, die Kabel aller Servos zu verlängern. Das geht zum Beispiel durch Zerschneiden der Kabel und Anlöten von Flachbandkabeln, wie auf den Fotos zu sehen ist. Alternativ kann man dreipolige Verlängerungskabel erwerben, die speziell für Servos gemacht sind.

Funkfernsteuerung

Die hier verwendeten Servos stammen aus dem funkferngesteuerten (RC-)Modellbau und bewegen dort normalerweise Achsen oder Ruder von Fahr- oder Flugmodellen. Es liegt daher nahe, eine solche Fernsteuerung auch für die animatronische Figur zu verwenden **7**. Günstige Sets mit Fernsteuerung, passendem Empfänger und Batteriehalter



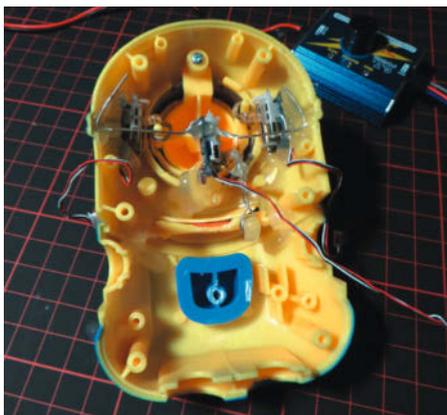
2 Servos in drei verschiedenen Größen – nur die beiden kleineren Typen wurden tatsächlich in die Minion-Figur eingebaut.



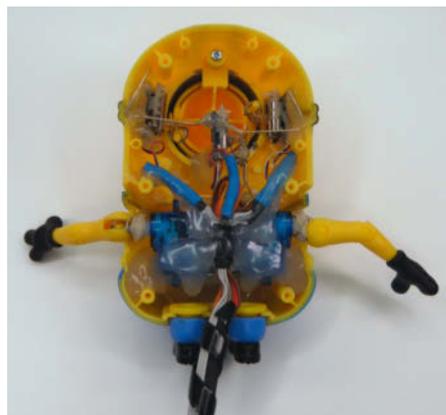
3 Die Arme sind mit jeweils einem 9g-Servo verbunden; dessen Drehachse bildet direkt das Schultergelenk.



4 Die beiden Augenlider werden durch zwei der ganz kleinen Servos bewegt.



5 Der dritte Servo, ebenfalls ein Miniatur-Exemplar, bewegt den Mund.



6 Alle fünf Servos sind montiert – viel Platz bleibt da nicht mehr.

sind im Internet oft für unter 60 Euro zu bekommen. Technische Kenntnisse braucht es für die Verwendung meist keine – lediglich auf die richtige Drehung der Servostecker an Hand der Beschriftung beziehungsweise der Farben ihrer Kabel muss man achten. Da alle Verbindungen nur gesteckt werden, kann man Fernsteuerung und der Empfänger anschließend für weitere Projekte und Basteleien mit Servos wiederverwenden.

Sobald alle Servo-Kabel mit der Fernsteuerung verbunden sind, kann die Figur direkt gesteuert werden: Bewegt man die Hebel der Fernsteuerung, bewegt der Minion Arme, Mund und Auge.

Steuerung per Arduino

Alternativ kann man die Bewegungen auch programmieren und dann immer wieder abspielen. Dazu bietet sich der Einsatz eines Arduinos an. Dieser kann die Modellbau-Servos bereits von Haus aus steuern, da er über entsprechende PWM-Ausgänge verfügt. Die meisten Arduino-Modelle wie der Arduino UNO können bis zu 12 Servos kontrollieren. Mit der *Arduino Servo Library* braucht man nur wenige Zeilen Code, um einen Servo auf einen gewünschten Winkel einzustellen 8.

Eine Alternative zur direkten Steuerung per Arduino stellt der Einsatz einer speziellen Servo-Karte dar, mit der man bis zu 16 Servos

steuern kann und die eine deutlich übersichtlichere Verkabelung ermöglicht. Solche Karten sind ab rund 15 Euro im Handel und werden per I²C mit nur vier Kabeln an den Arduino angeschlossen 9. Die digitalen Pins des Arduino werden dann nicht von den Servos belegt 10 und können für andere Dinge wie Sensoren oder LEDs verwendet werden.

Ausblick

Die Minion-Figur hier ist natürlich nur ein mögliches Beispiel – bei anderem, ausrangierten Spielzeug bieten sich wahrscheinlich andere animatronische Umbauten an. Für Figuren mit kleinem Kopf ergibt es vielleicht Sinn, den Nacken durch einen Servo zu ersetzen, damit der Kopf nach links und rechts schauen kann. Ist die Figur etwas größer, könnte man

versuchen, die Arme an mehreren Gelenken zu bewegen. Je nach Gewicht der Körperteile sind dabei aber kräftigere Servos ratsam.

Bei der Verwendung eines Arduinos gibt es noch weitere spannende Möglichkeiten der Interaktion: Durch einen Bewegungsmelder kann man sich zum Beispiel von der Figur winkend begrüßen lassen. Oder das Spielzeug spricht über ein MP3-Board und einen kleinen Lautsprecher zuvor aufgezeichnete Sätze. Beim hier gezeigten Minion wäre eine weitere Idee ein zusätzlicher, großer Servo unter den Füßen (etwa der schwarze aus Bild 2), mit welchem er sich komplett nach links und rechts drehen kann. Bei der ferngesteuerten Version hingegen könnte man die Figur auf einem RC-Modellauto montieren und sie ebenfalls ferngesteuert auf Spritztour schicken.

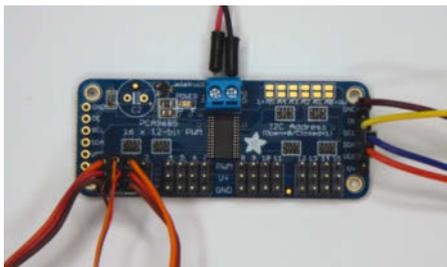
—pek

8 Servosteuerung (Auszug)

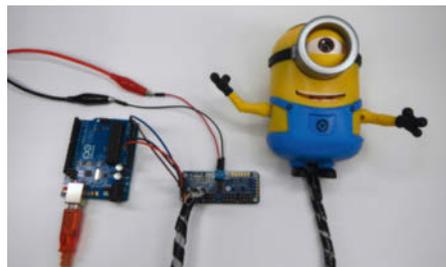
```
01 #include <Servo.h> // Die Arduino Servo Library einbinden
02
03 Servo servo; // Ein Servo-Objekt erzeugen
04
05 void setup() {
06     servo.attach(9); // Ordnet dem Servo-Objekt den PIN 9 zu
07     servo.write(60); // Sendet 60° als Zielwert an den Servo
08 }
```



7 Eine Funkfernsteuerung nebst Empfänger, Batterie und drei Servos



9 Eine Servo-Karte steuert bis zu 16 Servos über nur 4 Arduino-Pins.



10 Hier wird der Minion testweise mit der Servo-Karte gesteuert.

NEU

Raspberry Pi-Projekte zum Basteln, Steuern, Vernetzen

Auch als Heft + PDF erhältlich mit 29% Rabatt



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 Smart-home-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 oder ab einem Einkaufswert von 15 €. Nur solange der Vorrat reicht. Preisänderungen vorbehalten.

© Copyright by Maker Media GmbH.

 **heise shop**

shop.heise.de/ct-raspi20 >



Please Leave a Message

Wer in den letzten Monaten in Berlin an meiner Werkstatt in der Scheffelstraße 9 vorbeikam, konnte meiner Nachbarschaft und mir auf dem Plotter im Schaufenster eine kleine Nachricht hinterlassen.

von Niklas Roy



Please Leave A Message ist eine Art experimentelle, öffentliche Mailbox, die ich im Schaufenster meiner Werkstatt aufgebaut habe, einem alten Ladengeschäft im Osten Berlins. Passanten können mit ihren Smartphones darauf zugreifen, um Anderen eine nette Nachricht zu hinterlassen. Dabei macht es Spaß zuzusehen, wie die Nachrichten geschrieben werden, denn dies erledigt die solide Mechanik eines 30 Jahre alten *Aritma Colorgraf*-Plotters mit höchster Präzision.

Als ich mich dem Projekt widmete, befanden sich weite Teile der Welt im Corona-Lockdown. In Berlin durften wir draußen spazieren gehen, trotzdem fühlte es sich recht komisch an. Mit einer positiven Botschaft im Schaufenster wollte ich die Stimmung aufhellen, aber mir fehlte die zündende Idee. Deshalb dachte ich mir, dass sich die Menschen nette Nachrichten auch selbst schreiben könnten und ich nur die Infrastruktur bereitstelle.

Im Keller hatte ich einen alten DIN-A3-Plotter, den mir mein Nachbar Günter geschenkt hatte. Nach jahrelangem Lagern hatte zwar die Elektronik gelitten, doch die robuste Mechanik war noch völlig intakt. Daher beschloss ich, den Plotter zu zerlegen, nur die Mechanik zu behalten und die alten Platinen durch neue Elektronik zu ersetzen. Nun steuert ein Arduino Mega die beiden Schrittmotoren, welche den Stift in der X- und das Papier in der Y-Achse bewegen. Über einen Leistungstransistor schaltet der Arduino den Elektromagneten, der den Stift auf das Papier drückt. Ein paar Testzeichnungen später war der Plotter wieder einsatzfähig!

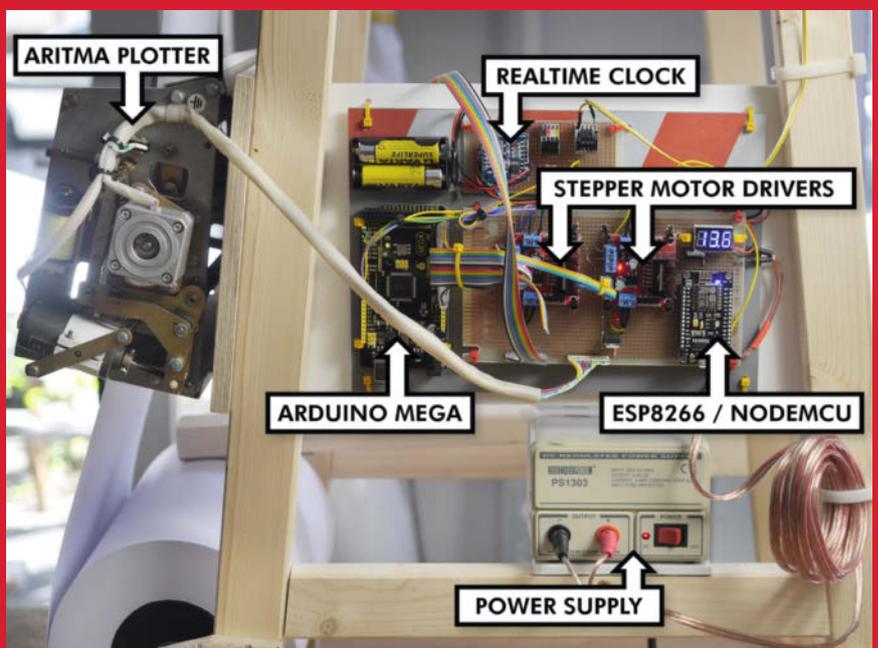
Den Plotter habe ich mit einem NodeMCU-Board mit ESP8266 verknüpft. Wer plotten möchte, muss sich zunächst ins Netzwerk des ESP einloggen und eine lokale Webseite aufrufen. Der Weg scheint etwas umständlich, hat aber den Vorteil, dass man nur auf den Plotter zugreifen kann, wenn man in direkter Umgebung des Ladens ist. Außerdem ist er nicht mit dem Internet verbunden. Die einfache Webseite stellt ein Formular mit Textfeld zur Verfügung, in das man seine Nachricht schreiben kann. Drückt man auf den Knopf *Drucken*, wird der Text vom Smartphone an den ESP8266 geschickt und von dort über die serielle Schnittstelle an den Arduino Mega weitergeleitet. Dieser steuert den Plotter und druckt die Botschaft aus.

Zuletzt habe ich einen großen Holzrahmen gebaut und den Plottermechanismus aufgeschraubt. Unten im Rahmen hängt eine dicke Papierrolle. Das Papier bewegt sich durch den Plotter zur Oberseite des Rahmens, läuft dort über ein abgeschnittenes PVC-Abflussrohr als Walze und wird hinten von einem Motor wieder aufgewickelt. Weitere Details der Installation, die Quellcodes sowie einige der Nachrichten sind auf meiner Webseite zu finden. —hch

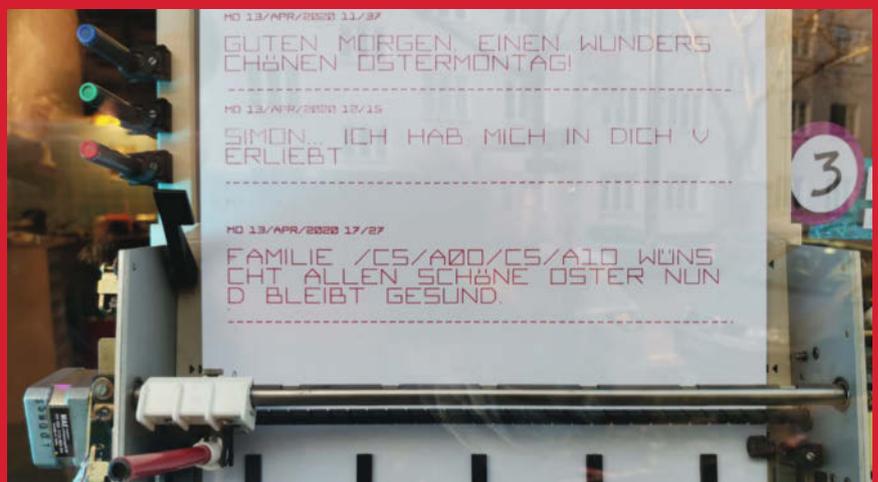
► niklasroy.com/pleaseleaveamessage



Das Holzgerüst nimmt später die Papierrolle auf, auf der die Nachrichten geplottet werden.



Grundlage des Projekts ist ein alter Aritma Colorgraf, den ich von seinem Gehäuse befreit habe. Nun wird er von moderner Elektronik gesteuert.



In der ersten Version konnte man dem Plotter Textnachrichten schicken, nach einem Update sind nun auch Zeichnungen möglich.

UV-Kammer für Resindrucker

Werkstücke aus dem Harzdrucker müssen nach dem Druck noch aushärten.
Schneller geht der Prozess mit dieser selbstgedruckten UV-Kammer mit
LED-Streifen.

von Werner Fugmann



Größer, schneller und günstiger – so lautet derzeit die Devise für Resin-3D-Drucker, die Werkstücke aus fotosensitivem, flüssigem Harz drucken und besonders feine Auflösungen ermöglichen. Ich habe früh zugeschlagen und mir einen kleinen LCD-UV-Drucker zugelegt. So schön die Drucke auch sind, so stellt sich doch schnell eine Frage: Wie und wo kann ich die fertigen Harz-Drucke vollständig aushärten? Denn im Gegensatz zum geläufigen Schichtdruck mit Filament ist nach dem Drucken ein weiterer Prozessschritt notwendig.

Den Druck tagelang in die Sonne zu legen ist in Deutschland für mich keine Option. Auch teure, fertige UV-Kammern lassen mein Maker-Herz nicht höher schlagen. Eine eigene Lösung musste her. Gesagt, getan! Mit der kostenlosen Konstruktionssoftware Fusion 360 entwarf ich eine einfache Box zum Selberdrucken. Die Box wird mit einer Tür mit Magnetverschluss und Scharnieren geschlossen. Ein Drehteller lässt das gewünschte Objekt in den Strahlen rotieren. Die Dateien für den Druck des Gehäuses und der Tür können umsonst heruntergeladen werden (siehe Link).

Nach dem Druck habe ich außen noch für einen rostigen „used look“ gesorgt. Dabei werden in einzelnen Arbeitsschritten verschiedene verdünnte Acrylfarben aufgetragen, verwischt und verschmiert. Dazu gehören braun, rot, orange und natürlich schwarz. Eine silberne Schicht aus Acryl-Sprühlack sorgt für eine gute Grundierung und den richtigen Metall-effekt. Innen habe ich alle sechs Seiten der Kammer mit Spiegelfolie beklebt – alternativ kann man Aluminium-Klebeband verwenden. Wichtig ist, dass alle Innenflächen, auch die Innenseite der Tür, am Ende mit einem reflektiven Material vollständig beklebt sind. Dies sorgt dafür, dass die UV-Strahlen gleichmäßig reflektiert werden und der 3D-Druck optimal von allen Seiten bestrahlt wird.

Als UV-Quelle empfehle ich einen LED-Streifen mit LEDs des Typs 2835. Sie haben eine Wellenlänge von 385 bis 400nm, was bereits geeignet ist, um 405-nm-Harz auszuhärten. Ein selbstklebender Streifen von drei bis fünf Metern Länge reicht aus. Ich habe ihn in Schlangenlinien auf die Seitenwände, Oberseite und die Rückwand direkt auf die Spiegelfolie geklebt. In der Rückwand befindet sich in einer Ecke ein kleines Loch, um das Kabel für den Stromanschluss herauszuführen. Am besten funktionieren die Kammer und der Härteprozess mit einem rotierenden, solarbetriebenen Drehteller. Das UV-Licht betreibt den Drehteller, sodass keine weitere Energiequelle nötig ist. Den gereinigten Resindruck muss man schließlich für circa eine Stunde in der Kammer aushärten lassen. —hch



Die abgemessene Spiegelfolie habe ich passend zugeschnitten.



Die LED-Streifen sind selbstklebend.



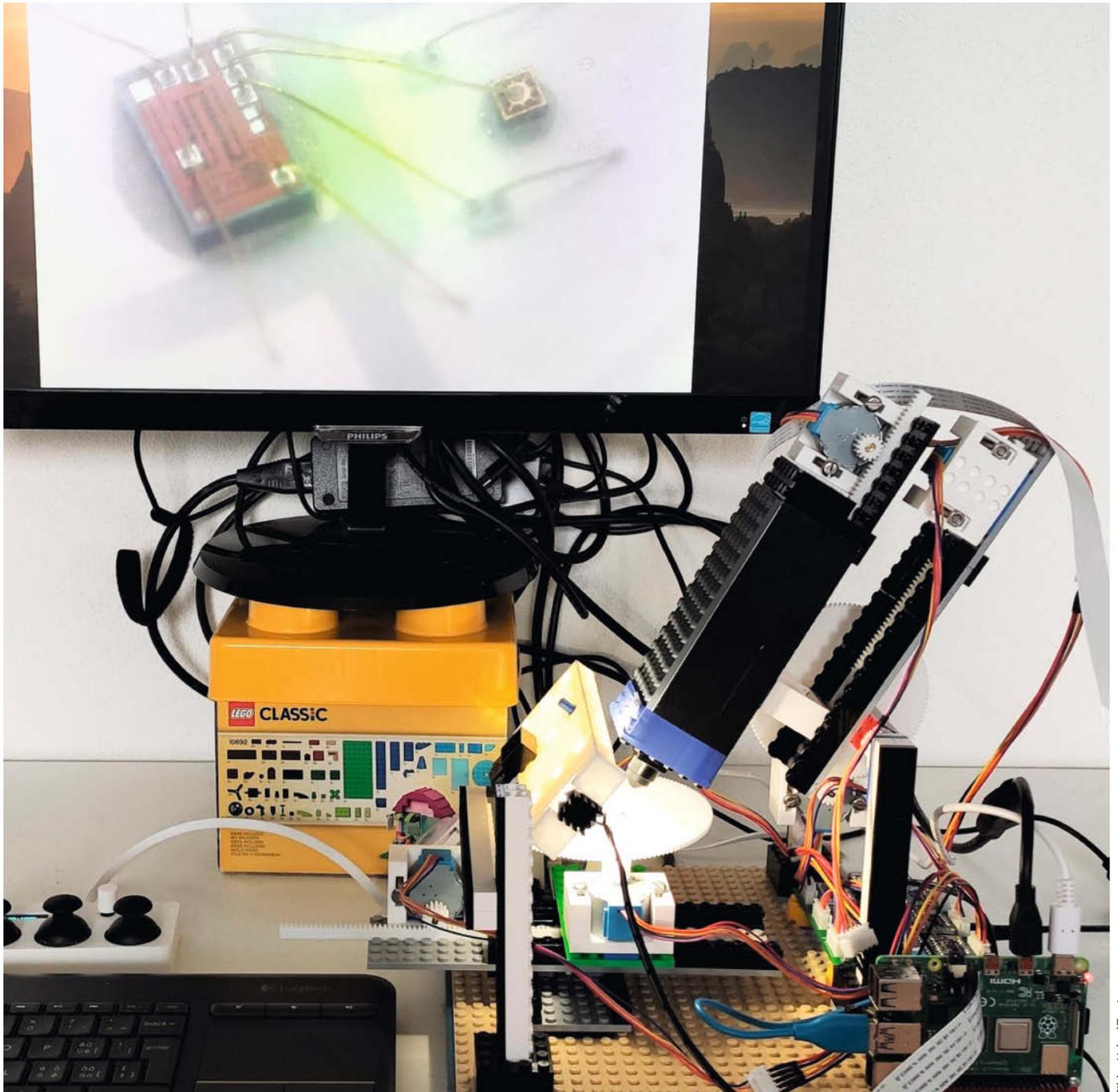
Dank bunter Farben bekommt die Kammer einen rostigen Look.

► heise.de/-4768962
 ► youtu.be/7SArh9Wjapw

MicroscopPy: das Lego-Mikroskop

3D-gedruckte Teile, zwei Arduinos und ein Raspberry Pi stecken in dem motorisierten Open-Source-Mikroskop MicroscopPy – und ganz viele Legosteine. Damit sind beeindruckende Bilder möglich.

von Helga Hansen



Bilder: Yuksei Terniz

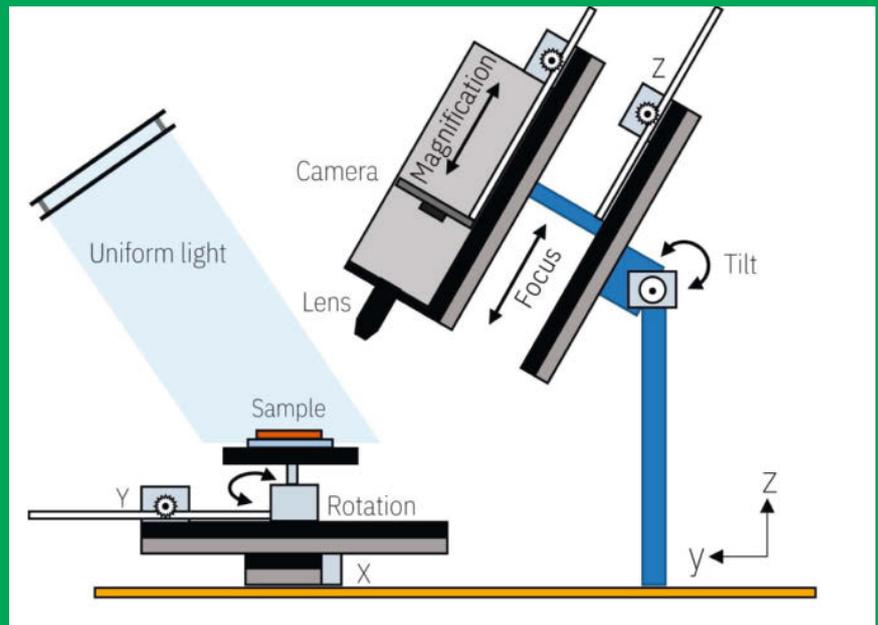
Mikroskopaufnahmen, die wie Hochglanzfotos aussehen: Das war das Ziel von Yuksel Temiz, der bei IBM in Zürich forscht und eigentlich winzige Chips für medizinische Anwendungen entwickelt. Um die Mikrochips für Präsentationen ins rechte Licht zu rücken, machte er aber einen Umweg über die Mikroskopentwicklung. Die Anleitung zum Bau des MicroscoPy hat er vor kurzem online veröffentlicht und setzt dabei auf Bauteile, die nicht nur viele Maker bereits zu Hause haben dürften: Legosteine.

Das Grundprinzip ist einfach. Das zu vergrößernde Teil wird auf den Drehteller gelegt, der von einer Hochleistungs-LED ausgeleuchtet wird. Den Teller kann man dabei nicht nur drehen lassen, sondern auch in der X- und Y-Achse verschieben. Die Kamera sitzt in einem Gehäuse auf einem Arm, der ebenfalls drei Bewegungsmöglichkeiten mit sich bringt. Der Aufbau kann um seine Aufhängung gekippt und zum Einstellen des Brennpunkts auf seiner Achse verschoben werden. Schließlich lässt sich auch der Abstand von Kamera und Linse im Gehäuse motorisiert einstellen und so die Vergrößerung beeinflussen. Ohne Objektivwechsel sind so Strukturen von wenigen Zentimetern bis zu wenigen Mikrometern erkennbar.

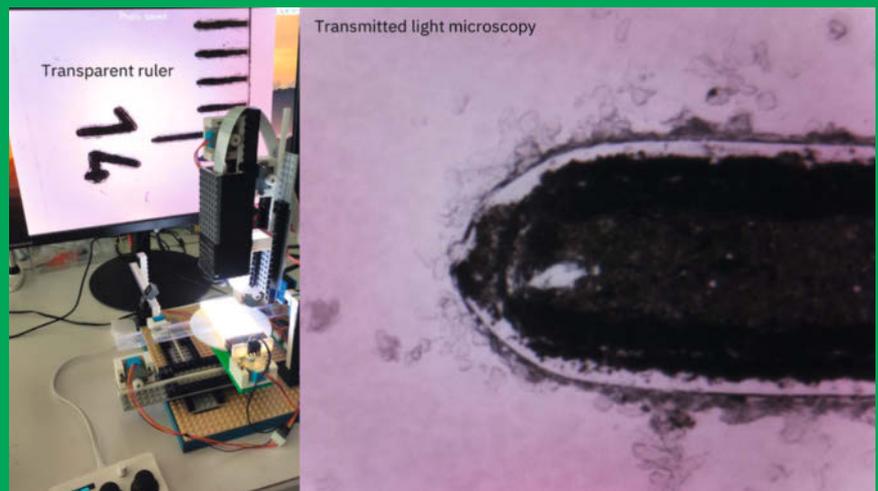
In seinem Mikroskop stecken dafür ein Raspberry Pi, der die 8-Megapixel-Kamera steuert, und zwei Arduino auf eigens entworfenen Platinen. Einer kümmert sich als Mainboard um die sechs Steppermotoren hinter den beweglichen Teilen und die LED-Beleuchtung. Über die serielle Schnittstelle (UART) kommuniziert er mit dem zweiten Arduino, der die Joysticksignale im kleinen Controllerpad verarbeitet. Nötig ist das allerdings nicht unbedingt, denn mit einem HDMI-Display und einer Tastatur für den Raspi kann das Gerät genauso gesteuert und die Bilder können dabei auch großformatig angeschaut werden. Für die Spannungsversorgung reicht das Original-Raspi-Netzteil mit 5 Volt und 3 Ampere, wobei ein zusätzlicher Spannungsregler die nötigen 12 Volt für die LED-Beleuchtung zur Verfügung stellt. Nur der Monitor benötigt dann noch ein eigenes Netzteil.

Um die Elektronik in Position zu halten, nutzte Temiz zunächst selbstentworfene Bauteile, die er mit dem 3D-Drucker hergestellt hat. Wer diese Variante nachbauen will, findet die Druckvorlagen aus FreeCAD zum Selberdrucken auf Github. Um Schulen und privaten Makern den Eigenbau zu ermöglichen, hat er die meisten Teile inzwischen durch Legosteine ersetzt. So kostet der Nachbau je nach gewünschten Features und Bezugsquellen zwischen 175 und 350 Euro. Die Dokumentation mit einigen Videos, Code für die Arduinos und den Raspi, Platinenlayouts und allen Anleitungen gibt es auf Github zum Herunterladen. —hch

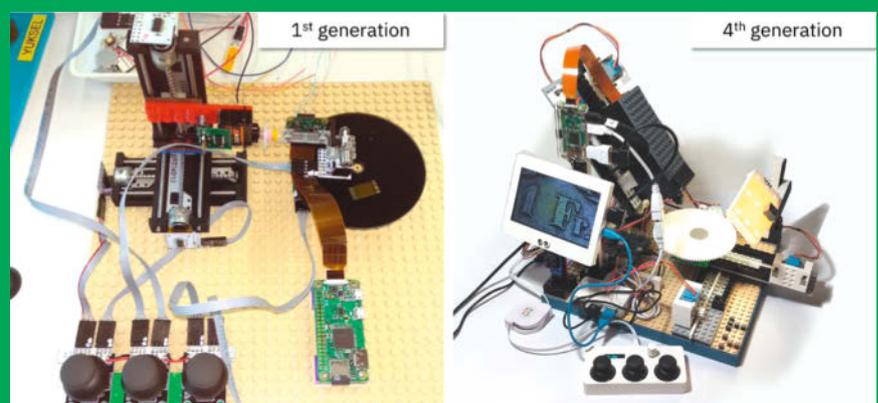
► github.com/IBM/MicroscoPy



Neben dem Basisaufbau mit reflektierendem Licht für Makrofotos sind auch weitere Aufbauvarianten möglich.



Hier etwa durchleuchtet das Licht die Probe, wie in einem klassischen Durchlichtmikroskop.

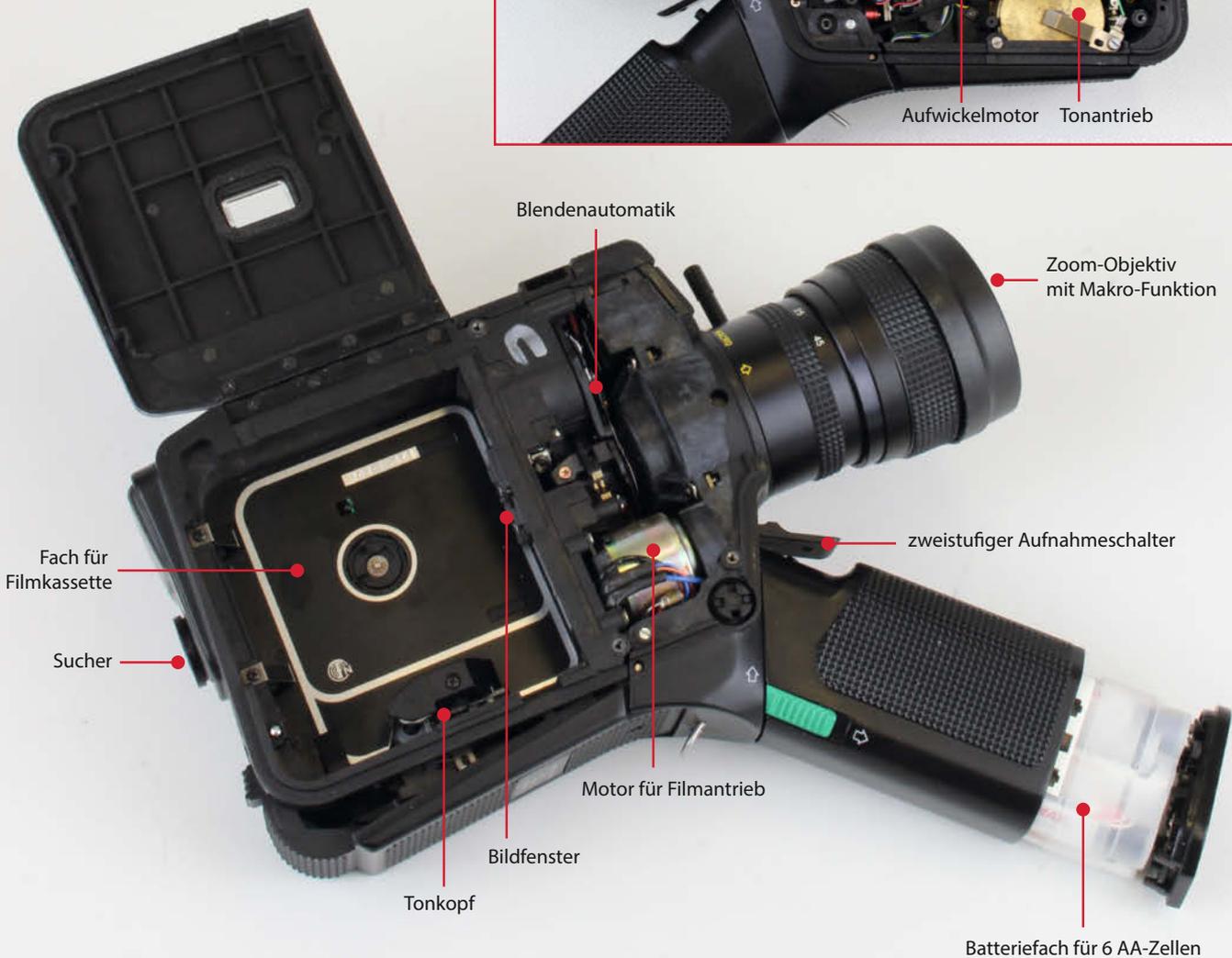
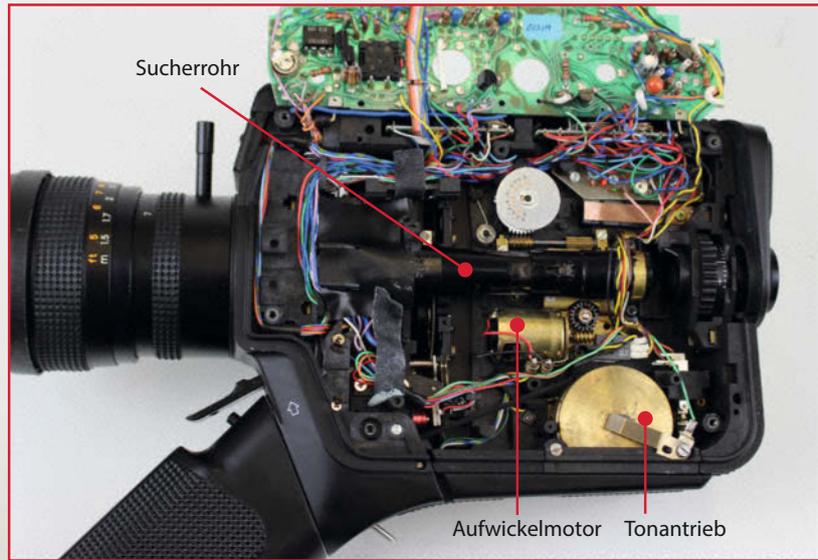


Über zwei Jahre Arbeit hat Temiz inzwischen in das Projekt gesteckt.

Super8-Tonfilm-Kamera

Einst teure, feinmechanische Optik-Objekte der Begierde, heute billig online verschertelt: Super8-Kameras beinhalten interessante Technik, aus der sich einiges Neues machen lässt.

von Heinz Behling



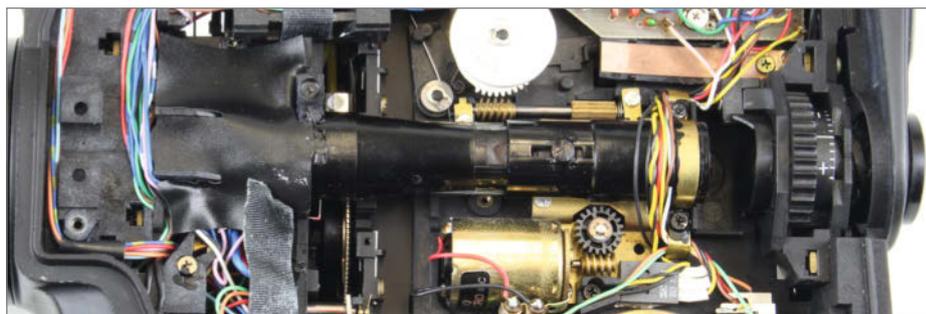
Da konnte ich nicht widerstehen, als ich diese Kamera auf eBay sah: Die Bauer 265 XL, eine von Bosch in Malaysia gebaute Tonfilm-Kamera im Super8-Format. Ein paar Tage später hielt ich das inzwischen 40 Jahre alte technische Meisterwerk in den Händen. Dank äußerst stabiler Bauweise, trotz Kunststoffgehäuse, hatte es die Zeit nahezu spurlos überstanden bis auf ein paar kleine, abgewetzte Stellen an der Seite. Das Innenleben erschien wie frisch aus der Fabrik. Diese Kamera mit ihrem sicher einmal deutlich vierstelligen D-Mark-Preis war nun für 10 Euro mein Eigen.

Der erste Blick ins geöffnete Filmfach zeigte, dass das Bildfenster mit $5,69\text{mm} \times 4,22\text{mm}$ nur ein wenig größer ist als die Sensoren gängiger Raspberry-Kameras ($3,67\text{mm} \times 2,76\text{mm}$).

Da könnte man doch vielleicht einmal Kamera nebst Raspberry Zero direkt ins Gehäuse integrieren. Das Filmkassetten-Fach böte genügend Platz dafür. Und auch sonst hat diese Kamera einiges zu bieten, was sich wiederverwenden ließe. Da Super8-Filme nicht sehr lichtempfindlich waren, wurden meist lichtstarke Objektive eingebaut. So auch hier: ein Motor-Zoomobjektiv ($7\text{mm} - 45\text{mm}$ Brennweite) mit Lichtstärke 1:1.2 und das auch noch mit Makrofunktion und Schnittbild-Entfernungsmesser. Wer hat sowas schon an einer RasPi-Kamera?

Die eingebaute Stromversorgung (6 AA-Zellen) ist im Griff untergebracht, der zugleich den Auslöser trägt. Zusätzlich gibt es zur Kamera eine Fernbedienung, wengleich *fern* hier sehr relativ ist, da deren Kabellänge mit etwa 60cm eher kurz ist. Sie hilft aber, die Kamera bei Stativmontage vor Erschütterungen durch Betätigen des Auslösers zu schützen. Die Kamera lässt sich übrigens nicht nur im üblichen Quer-, sondern auch im Hochformat aufs Stativ montieren.

Ein Belichtungsmesser im Inneren regelt die Blendenöffnung. Dieser Mechanismus sitzt direkt hinter dem Objektiv und teilt sich den Platz mit dem Motor für den Filmtransport (der auch die Sektorblende vor dem Bildfenster



Hinter der Platine verstecken sich das Sucherrohr und die weiteren Motoren.



Zwei ICs auf der Platine, das war 1980 ein wichtiges Verkaufsargument in Katalogen.

bewegt) und dem Prisma, das ein Teil des Lichts zum Sucher leitet.

Hinter der linken Seitenwand der Kamera sitzt die Elektronik, die nicht nur die Filmdrehzahl und die Belichtung steuert, sondern vor allem auch den Tonteil enthält. Die Platine lässt sich service- (und Maker-) freundlich ausklappen. Sensationell für Baujahr 1980: Auf der Platine sitzen sogar schon zwei integrierte Schaltungen, der Mikrofoneingang sowie eine Kopfhörerbuchse.



Die automatische Blende wird vom Belichtungsmesser gesteuert oder manuell eingestellt.

Unter der Platine kommen dann unter anderem das Sucherrohr und die Motoren zum Filmaufwickeln und für den Antrieb des Tonteils zum Vorschein.

Das wird sicher eines meiner nächsten Projekte: Die Super8-RasPi-Kamera. Platz für die Elektronik ist da, Strom steht zur Verfügung, das Objektiv ist Spitze und sogar Sound ist möglich. Jetzt muss ich dann nur noch das Schmalfilm-typische Flimmern hinbekommen. —hgb



Das Bildfenster bei Super8 ist $5,69\text{mm} \times 4,22\text{mm}$ groß und sitzt zusammen mit dem Filmtransport-Greifer in einer Führungsschiene.



Das lichtstarke Objektiv besitzt eine Makrofunktion, die mit dem Zoomring eingeschaltet wird.

Slicer Shootout

Zum 3D-Drucken braucht man einen Slicer. Wir haben drei quelloffene und kostenlose Slicer-Programme auf Herz und Nieren geprüft und in verschiedenen Disziplinen verglichen: Das Urgestein Slic3r, Ultimakers Cura und Prusas Slic3r-Weiterentwicklung namens PrusaSlicer.

von Rebecca Husemann



Wer sich einen nagelneuen 3D-Drucker bestellt hat, ist meistens ganz scharf darauf, ihn nur noch zu kalibrieren und dann dem Wunder des ersten richtigen 3D-Drucks beiwohnen zu dürfen. Ein 3D-Modell ist im Internet schnell gefunden (Was nützliches? Was niedliches? Hihi, erstmal diese lustige Büste von Lenin) und dann wird es plötzlich kompliziert: Das ist ja eine STL-Datei. Kann man die so drucken? Nein, dazu muss man sie erst in einem Slicer in G-code umwandeln.

Was ist eigentlich ein Slicer?

Eine Slicer-Software wandelt ein 3D-Modell – meist im OBJ- oder STL-Format – in Anweisungen um, die ein 3D-Drucker verstehen kann. Das Ergebnis ist G-code, also Maschinensprache. Dem 3D-Drucker werden mit dem G-code maschinenrelevante Daten übergeben, die genau für den verwendeten 3D-Drucker definiert sind. Und ohne G-code läuft erstmal gar nix.

Ein vom G-code gesteuerter Fertigungsvorgang läuft ab Start vollautomatisch. Alle Arbeitsschritte sind vordefiniert und eigentlich ist kein Eingreifen von Menschen mehr nötig. Steuert man einen Drucker mit Programmen wie Octoprint und Repetier, kann man dem G-code beim Ablaufen zusehen, sogar übers Netz. Mehr zu solchen Druckservern erfahren Sie ab Seite 98.

Und warum gibt es so viele unterschiedliche Programme, die G-code generieren? Das liegt an der Vielzahl verschiedener Druckerhersteller und engagierter Open-Source-Projekte, die versuchen, die Grenzen des Druckbaren immer weiter zu verschieben. Das ist ja eigentlich super, führt aber zu vielen unterschiedlichen Produkten auf dem Markt. Und dazu, dass man oft mehrere Druckversuche braucht, bis man die Einstellungen für den G-code des eigenen Druckermodells perfektioniert hat.

Anfängern reicht es meist schon, wenn das Objekt gleichmäßig und ordentlich gedruckt wird. (Seien wir ehrlich: ist schon toll, wenn es überhaupt gedruckt wird.) Mit der Zeit kann man die Werte allerdings immer weiter optimieren. Ob man nun dafür sorgen will, dass Drucke bei gleichbleibender Qualität schneller laufen oder man den Materialverbrauch senken möchte: Profis freuen sich über mehr Detail-Einstellungen. Doch auch für Anfänger sollte ein Slicer alle Interface-Einstellungen bieten, um schnell zu einem befriedigenden Ergebnis zu kommen.

Wie ein Slicer den Druck optimiert

Aber slicen ist noch mehr als nur die Umwandlung in Maschinensprache. Eine STL-Datei von einem 3D-Modell enthält zunächst alle räum-

Kurzinfo

- » **Drei Slicer-Programme im Vergleich**
- » **Welche Software sich für den Einstieg eignet**
- » **Slicer-Einstellungen für Profis**

Mehr zum Thema

- » Sebastian Müller, Schöner leben – mit 3D-Druck, Make 1/20, S. 108
- » Heinz Behling, Versteckte Cura-Funktionen, Make 1/19, ab S. 90
- » Heinz Behling, 30 Filamente für jeden Zweck, Make 2/19, ab S. 8
- » Dominik Laa, Richtig stützen bei Resin-3D-Druck, Make 3/20, ab S. 102

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

lichen Daten, die man für den 3D-Druck braucht. Der Slicer rechnet das Modell dann in einzelne Schichten um, aus denen sich das Modell aufbaut und übersetzt das Modell so in lineare Bewegungen auf den drei Achsen.

An diesem Punkt geschieht das, was die Slicer voneinander unterscheidet: Das Programm optimiert die Wege, die der Extruder nimmt, nach verschiedenen Gesichtspunkten. Wie lange benötigt die Schicht zum Abkühlen? Wie kann man unschöne Narben an Verbindungsstellen vermeiden? Wie wird der Druck am stabilsten?

Zudem gibt der Slicer noch Informationen zu Variablen wie Betttemperatur, Extrudertemperatur und Lüfterbewegung für die verschiedenen Schichten mit. Auch eventuelle Abstandssensoren und andere Besonderheiten werden vom G-code kontrolliert. Je besser die Voreinstellungen im G-code sind, desto besser ist die Druckqualität.

Dazu kommen die Anpassungen, die man selbst im Slicer vornehmen kann. Slicer bieten ein grafisches Interface, in dem man erkennen kann, wo ein oder mehrere Objekte auf der Druckplatte positioniert sind, wie groß sie sind und ob man Überhänge abstützen muss. Manche Slicer haben besondere Stärken, wie den Druck von Brücken, besonders sparsames Infill oder elegante Stützstrukturen. Die Schwächen reichen von unregelmäßigen Oberflächen zu unschönen Nähten oder dünnen Deckflächen.

Welche Slicer vergleiche ich?

Ich betrachte drei beliebte kostenlose und quelloffene Slicer: Die Urgesteine Slic3r 1.3.0 und Cura 4.6.1 sowie den Neuzugang Prusa Slicer 2.2.0.

Cura und **Slic3r** werden jeweils seit 2011 entwickelt. Slic3r wird größtenteils von der Community geschrieben, während die Cura-Entwickler bei Ultimaker in Vollzeit die Entwicklung der Software vorantreiben können. Der **PrusaSlicer** hat den Original-Slic3r als Ausgangsbasis verwendet und wird seit ungefähr 2017 vom Prusa-Team weiterentwickelt.

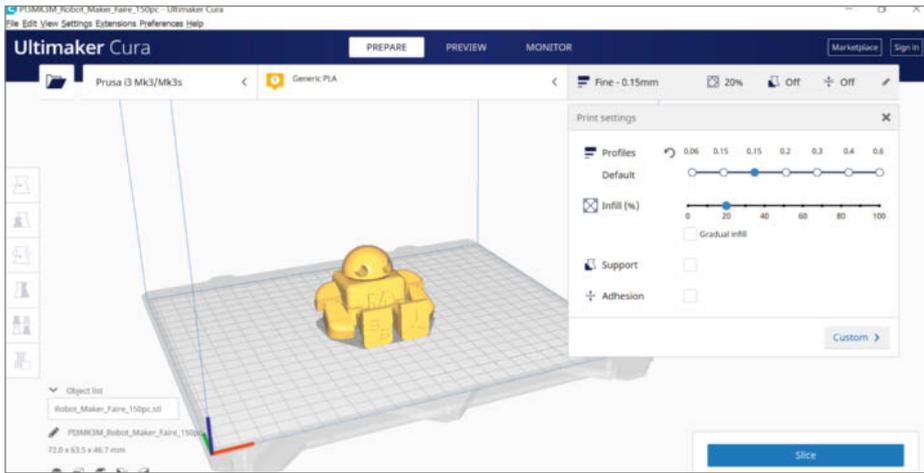
Slic3r hatte als Projekt von freiwilligen Entwicklern lange den Vorteil, dass es besonders experimentelle Features zuerst implementieren konnte – Features, die später zum Indust-

Honorable Mention

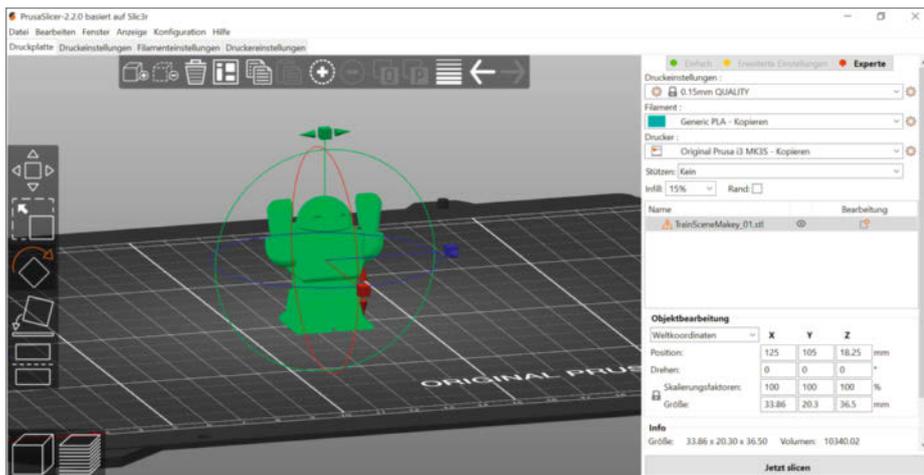
Ich habe mich für den Test auf kostenfreie Slicer beschränkt. Viele 3D-Drucker-Profis schwören allerdings auf Simplify3D. Dieses Programm ist zwar kostenpflichtig, bietet dafür aber zuverlässig tolle Druckergebnisse. Wer zum Beispiel 3D-Drucke verkaufen möchte, für den kann sich die Investition lohnen.

Slicer

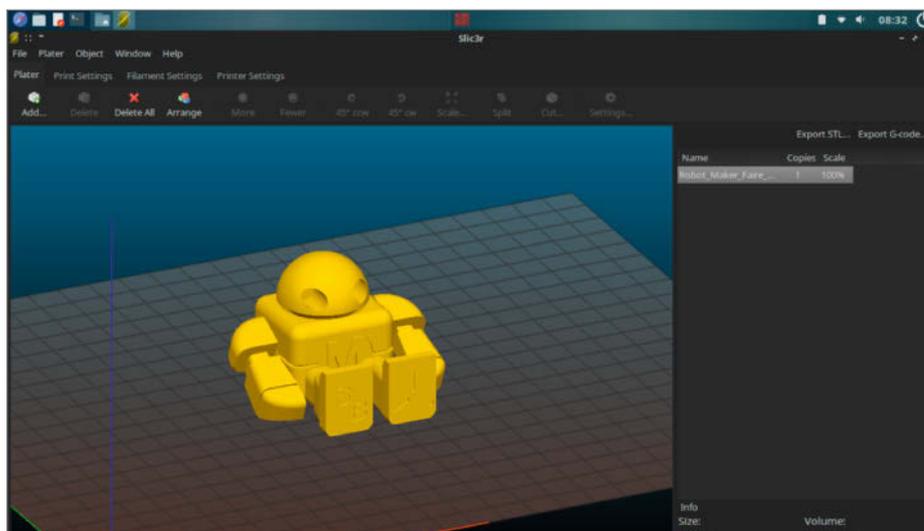
Software	Cura	Slic3r	PrusaSlicer
Hersteller	Ultimaker	unabhängig	Prusa Research
Sprachen	Englisch, Deutsch und 14 weitere	Englisch	Englisch, Deutsch und 10 weitere
getestete Version	4.6.2	1.3.0	2.2.0
Betriebssysteme	Windows, macOS, Linux	Windows, macOS, Linux	Windows, macOS, Linux
Lizenz	frei und quelloffen	frei und quelloffen	frei und quelloffen
Kosten	kostenlos	kostenlos	kostenlos



Das Interface von Cura ist sehr aufgeräumt und einladend.



PrusaSlicers Icon-Auswahl macht Lust zum Rumprobieren.



Slic3rs User-Interface ist zwar nicht modern, aber trotzdem übersichtlich.

riestandard wurden. Optionen wie Brim, variable Schichthöhen und die Ansteuerung mehrerer Extruder haben wir dem quelloffenen Projekt zu verdanken. In letzter Zeit ist die Entwicklung allerdings ins Stocken geraten.

Beim PrusaSlicer handelt es sich um den weiterentwickelten Slic3r und am Vergleich der beiden kann man schön erkennen, wie schnell sich die Slicer-Welt weiterentwickelt (besonders, wenn ein bezahltes Entwickler-

team neue Funktionen in kürzester Zeit raus-hauen kann).

Cura ist ein zuverlässiges Arbeitstier, das seit 2011 kontinuierlich weiterentwickelt wird. Hersteller Ultimaker stellt auch Einstellungen für Konkurrenzdrucker bereit: Das würde nicht jede Firma machen. Gekoppelt mit der Open-Source-Philosophie der Entwickler macht diese Konsistenz Cura bisher für viele Anwender zum Goldstandard.

Im Test bewerte ich den Druck auf Basis der serienmäßigen Druckprofile und prüfe, wie viele Optionen die Software einem erfahrenen Anwender fürs Finetuning an die Hand gibt.

Die Installation

Zur Installation kann man nicht viel sagen: Sie ist bei allen Programmen einfach. Die Installationsdateien und Anleitungen sind auf den Webseiten hinterlegt und aktuell (alle Links siehe Kurzinfor). Bei Beginn der Installation wird man schon nach ersten Voreinstellungen und Präferenzen gefragt. Bei Cura und Prusa Slicer kann man voreingestellte Druckerprofile laden, während man im Slic3r beim ersten Start unter Anleitung einen Drucker manuell konfiguriert.

Die Bedienoberfläche im Vergleich

Damit eine Software gut angenommen wird, braucht sie eine intuitive und benutzerfreundliche Bedienoberfläche. Einfache Aufgaben sollte man ohne viel Suchen oder Googeln erledigen können.

Der Grundaufbau der Bedienoberfläche ist bei allen drei Programmen ähnlich: Es gibt eine Darstellung vom Druckbett im 3D-Raum und daneben ein Menü für die Einstellungen und die Druckerauswahl.

Bei **Cura** merkt man, dass der Slicer im Tandem mit einem Drucker entwickelt wird, der Anfänger ansprechen soll. Es sieht auf den ersten Blick einladend, übersichtlich und modern aus. Man startet mit den anfangergefreundlichen Einstellungen, die *Custom-Einstellungen* verstecken sich dann einen Klick weiter.

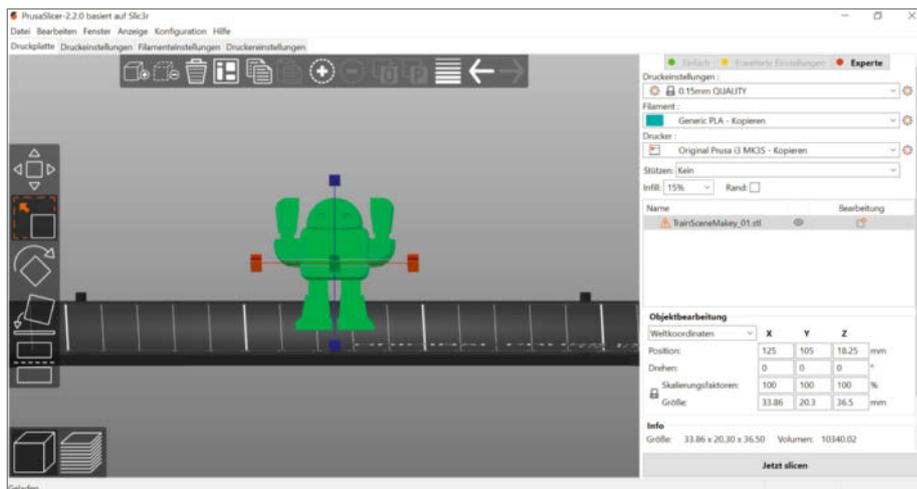
Aber ehrlich gesagt: Das Design ist verwirrend und die Entwickler haben es mit dem Minimalismus etwas zu gut gemeint. Wo sind denn hier die Funktionen? Die „richtigen“ Profi-Einstellungen muss man erst in der Konfiguration extra und einzeln aktivieren (oder als Plugin nachinstallieren). Dort ist die Auswahl dann wirklich überfordernd. Das ist so ähnlich wie ein aufgeräumtes Zimmer, in dem aber alles nur in den nächsten Schrank gestopft und die Tür vernagelt wurde. Nicht umsonst haben wir bereits einen Artikel namens „Versteckte Cura-Funktionen“ veröffentlicht. Für Einsteiger definitiv optimal, für Profis etwas umständlich.

„Mit Cura ist 3D-Drucken so einfach wie 2D-Drucken“, lese ich in einer Review. Der Kritiker hat sich offenbar noch nie mit 2D-Druckertreibern rumgeärgert. Ich würde sagen: Mit Cura ist 3D-Drucken sogar einfacher als 2D-Drucken.

Der **PrusaSlicer** ist in den Händen des Prusa-Teams einfach schön geworden und besonders die klaren Icons lassen Designerherzen höher schlagen. Die Auswahl für Druckermodele sieht noch fast genauso wie beim Slic3r aus, doch dank dieser vertrauten Ansicht findet man sich umso schneller zurecht. Man kann unkompliziert zwischen *Einfach*, *Erweiterter Modus* und *Expertenmodus* wechseln. Mit einem Klick auf die drei Kategorien *Druckereinstellungen*, *Filamenteinstellungen* und *Drucker-einstellungen* kann man beliebig tief einsteigen. Im Bauraum bietet der PrusaSlicer ein paar Optionen-Knöpfe mehr als Cura, das kommt aber sehr aufgeräumt daher.

Hätte ich nicht eine gewisse Nostalgie für den **Slic3r**, würde ich ihn vielleicht hässlich nennen. Allerdings ist er sehr funktional und übersichtlich aufgebaut. Das Programm bündelt alle Einstellungen in drei Kategorien und macht den Überblick für Neueinsteiger möglich, während es gleichzeitig viele Anpassungsmöglichkeiten für Profis bietet. Früher gab es wie im PrusaSlicer die Option zwischen Profi- und Fortgeschrittenen-Optionen zu wechseln, doch das wurde beim letzten Release abgeschafft: „Simple/Expert Mode is no more, we are all experts now.“ Die Buttons für die Bearbeitung des Modells im Bauraum sind alle beschriftet und zusätzlich mit einem Icon versehen – womit er vermutlich von den dreien der barrierefreieste Slicer ist.

Bei allen drei Programmen kann man sich das Modell in einer farbkodierten Layer-Simulation ansehen. Cura bietet zudem einen *X-Ray-Modus*, in dem man gut erkennen kann, ob das Modell an sich fehlerhaft ist.



Funktionen wie Skalieren und Rotieren sind im PrusaSlicer schön eindeutig dargestellt.

Objekte verschieben, skalieren und rotieren

Alle drei Slicer bieten die Möglichkeit, ein Objekt mit einem Klick zu importieren, per Drag&Drop auf dem Druckbett zu positionieren, zu skalieren, zu rotieren und flach auf das Druckbett zu legen. Sie können mehrere Objekte automatisch optimal auf dem Druckbett verteilen. Dabei gibt es in jedem Slicer die Option, die Drucke nacheinander statt parallel durchzuführen – sie ist nur unterschiedlich gut versteckt. (Ja, Cura, du bist gemeint.)

Beim **PrusaSlicer** kann man Änderungen am Modell sowohl mit der Maus nach Augenmaß durchführen, als auch punktgenau im Textfeld eingeben. Beim Drehen können in der visuellen Darstellung zu Beispiel einzelne Drehachsen angeklickt werden. Diese Navigation ist im PrusaSlicer sehr intuitiv zu bedienen.

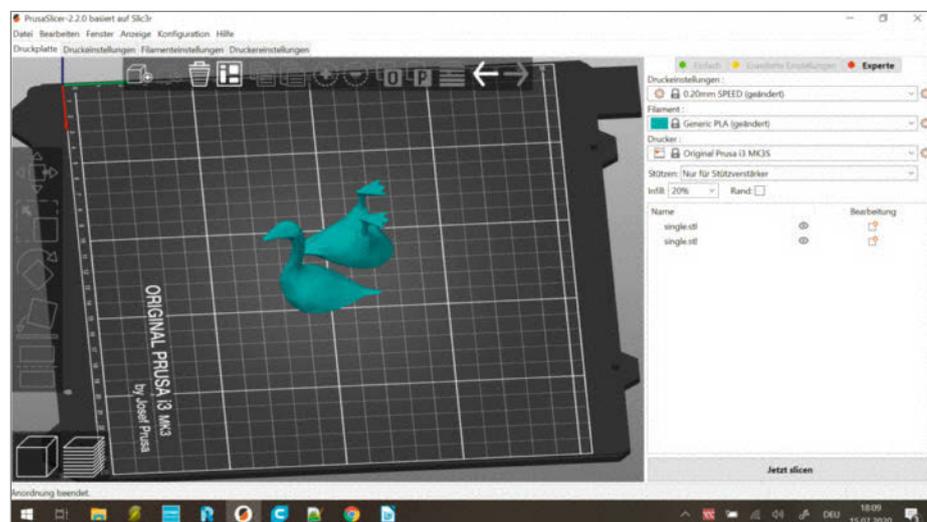
Bei **Cura** sind die Funktionen genauso wie beim PrusaSlicer: Visuell eindeutig und

sie gehen ohne Nachdenken von der Hand. Allerdings kann die Rotation nur in 15 Grad-Schritten erfolgen, außer man hält beim Rotieren mit der Maus die Shift-Taste gedrückt. So kann es recht mühselig werden, ein Modell genau so zu rotieren, wie man es möchte.

Im **Slic3r** kann man das Modell per Knopfdruck in 45 Grad-Schritten um die Z-Achse drehen, doch das winkelgenaue Rotieren muss man erst im Kontextmenü suchen. Skalieren kann man das Modell mit einem Klick – in Prozentangaben. Eine metrische Wunschhöhe kann man so nicht angeben, dafür muss man die etwas versteckte „scale to size“-Funktion verwenden.

Modelle im Slicer zerschneiden

Die Schneiden-Funktion sticht beim **Prusa Slicer** heraus: Das Zerteilen eines Objektes in mehrere Teile ist geradezu verdächtig einfach. Man kann gleich mit auswählen, ob man die



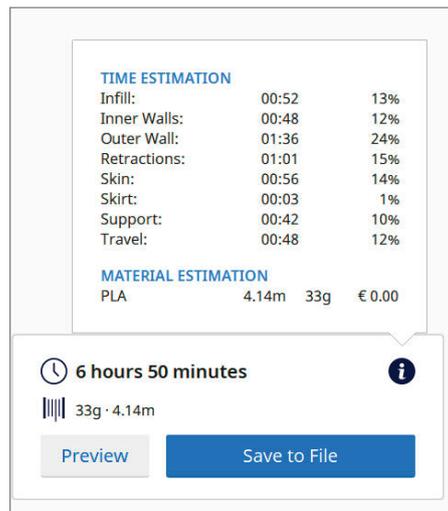
Die Zerschneiden-Funktion im PrusaSlicer



Die beiden Teile lassen sich ordentlich drucken und passen bündig aufeinander.

Kompatibilität

Software	Cura	PrusaSlicer	Slic3r
Materialprofile	62	110	Keine Voreinstellungen
Unterstützte Drucker	Über 280 Modelle	Prusa, Creality, BIBO, LulzBot	Keine Voreinstellungen
Eingabeformat	STL, OBJ, X3D, 3MF, BMP, GIF, JPG, PNG	STL, OBJ, AMF, XML, 3MF, PRUSA	STL, OBJ, AMF, XML, 3MF
Ausgabeformat	G-code, 3MF, STL, UFF, OBJ	G-code, STL, AMF, OBJ	G-code, SVG, STL



Cura bietet ausführliche Angaben zur Druckzeit einzelner Komponenten.

untere Hälfte mit der Schnittfläche nach unten positionieren möchte. **Slic3r** hat die Vorgängerversion von PrusaSlicers Schneid-Feature, die nur ein bisschen weniger schick aussieht.

Cura bietet diese Funktion leider nicht. Es ist auch nicht geplant, sie in naher Zukunft zu implementieren.

Voreingestellte Filamente und Modelle

Cura hat viele voreingestellte Druckprofile für Drucker und Filamente, mit denen man ohne Nachdenken durchstarten kann. Von Anet über Creality3D bis zu Renkforce ist die Auswahl beeindruckend. Zudem kann man auch Druckerprofile vom Hersteller importieren. Auch cool: Cura kann PNG-Dateien importieren und in den 3D-Raum umwandeln.

Bei **PrusaSlicer** stehen die hauseigenen Druckermodelle im Vordergrund. Daneben unterstützt er nativ Modelle von Creality, BIBO und LulzBot. Bei den Filamenten lässt Prusa dagegen nichts zu wünschen übrig, die Auswahl vordefinierter Filamente von verschiedenen Anbietern ist überwältigend.

Beim **Slic3r** ist das Anlegen neuer Drucker noch echte Handarbeit – und im Vergleich nicht sehr komfortabel. Allerdings kann man auch im Slic3r Druckerprofile von Herstellern direkt importieren.

Druckdauer und Filamentverbrauch

Die Angaben eines Slicers zu Druckdauer, Filamentverbrauch und Materialkosten sind wich-

tig, besonders wenn es sich um kostenpflichtige Gemeinschaftsgeräte in Bibliotheken, Hacker- und Makerspaces handelt. Auch Personen, die Auftragsdrucke ausführen, benötigen Eckdaten für die Kostenkalkulation.

Cura liefert dem Benutzer Informationen zur Druckdauer, Filamentverbrauch in Länge und Gewicht und eine Kostenkalkulation. Schön ist, dass Cura genau aufschlüsselt, wie viel der Druckzeit für einzelne Komponenten – wie Infill, Stützen und Wände – aufgewendet wird. So weiß man, bei welchen Einstellungen noch Beschleunigungspotential ist.

PrusaSlicer gibt ebenfalls Druckdauer, Filamentverbrauch in Länge und Gewicht und eine grobe Kostenkalkulation aus. Bei der Druckdauer wird auch die Druckzeit für die Option *Stealth Mode* angegeben. Wählt man diese Option aus, wird der Druck besonders leise – dafür dauert er ein paar Minuten länger.

Slic3r ist in dieser Hinsicht sehr unkomfortabel. Es gibt lediglich den voraussichtlichen Filamentverbrauch an. Definiert man die Kosten für ein Filament selbst, erhält man immerhin eine Kostenabschätzung. Doch dass die Druckdauer nicht im Slic3r ersichtlich ist, ist eigentlich kaum zu glauben. Wie konnte ich jemals so arbeiten?

Slice-Dauer

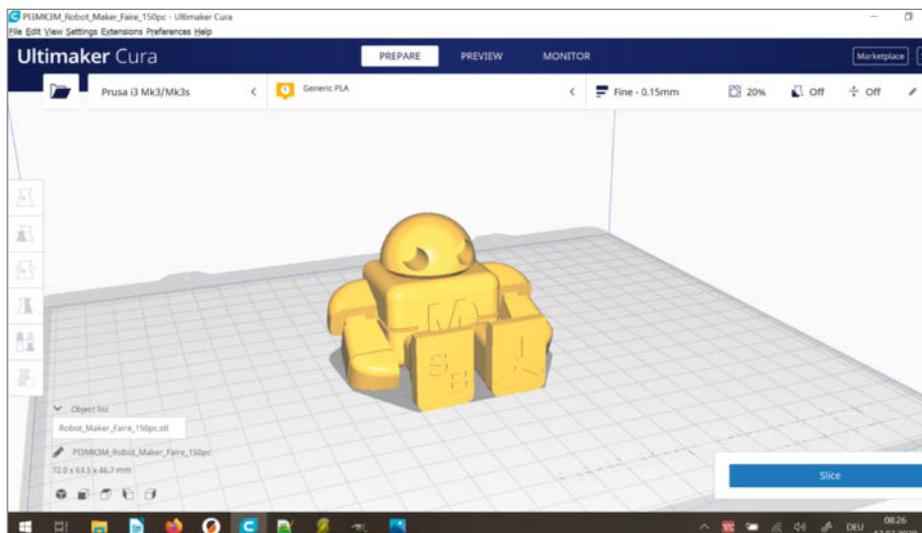
Wie schnell eine Slicer-Software ein Modell in G-code umwandeln kann, macht bei kleinen Modellen kaum einen Unterschied. Doch wer regelmäßig große Projekte druckt, sollte eindeutig den schnellsten Slicer bevorzugen – wie man im folgenden Test auf einem Lenovo Thinkpad X200s deutlich sehen kann.

Als Testobjekt nehmen wir unseren Roboter Makey in zwei Ausführungen: Die „kleine“ Version ist 2,4MB groß, die „große“ Version dagegen 5MB.

Cura bietet ein solides und reproduzierbares Ergebnis. Der kleine Makey ist in gut 15 Sekunden gesliced, der Große braucht nur 20 Sekunden. Auch bei weiteren Testdurchläufen bewegt Cura sich immer grob um die 16 Sekunden-Marke. Hier geht an Violdrucker eine klare Empfehlung für Cura raus.

Der **Slic3r** ist das genaue Gegenteil: Bei einfachen Modellen ist es die schnellste Software, doch bei größeren Modellen skaliert die Berechnungszeit überhaupt nicht mit. Der kleine Makey ist in 8 Sekunden berechnet, doch beim großen Makey bin ich nach über 90 Sekunden Slicezeit beinahe auf dem warmen Druckbett eingeschlafen.

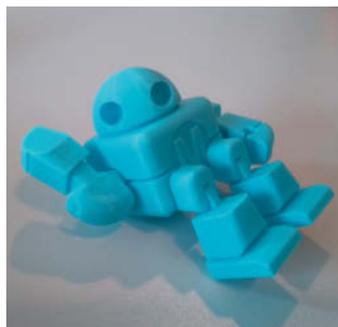
PrusaSlicer scheint das Skalierungsproblem vom Slic3r übernommen zu haben, ist aber längst nicht so extrem. Für den kleinen Makey benötigt er nur 10 Sekunden und der große Makey ist nach 40 Sekunden bereit zum Druck.



Der Makey war sowohl Testobjekt für die Slice-Dauer als auch für die Basiseinstellungen.



Leider sehr feste Gelenke, aber dafür unglaublich sauber gedruckt: Cura auf dem Ultimaker.



Cura auf dem Prusa i3 MK3S bietet auch ein sehr sauberes Druckergebnis.



Sehr beweglich und perfekt gedruckt: Das Ergebnis von Prusa auf Prusa.



PrusaSlicer auf dem Ultimaker? Viel Stringing und eine raue Oberfläche. Genauso sieht's beim Slic3r aus.

Drucktest mit Basiseinstellungen

Um zu testen, wie gut die mitgelieferten Basiseinstellungen der Slicer sind, habe ich unseren Roboter Makey auf zwei verschiedenen 3D-Druckern gedruckt.

Makey ist ein sogenanntes Print-in-Place-Modell (PIP). Das Modell auf Thingiverse finden Sie in den Links in der Kurzinfo. Er enthält verschiedene Arten von Verbindungselementen wie bewegliche Teile und Scharniere, die direkt mitgedruckt werden. Ich habe nicht in den Druck eingegriffen, nichts verstellt, nur im Prusa und Cura das Filament und das Druckermodell gewählt und im Slic3r die Nozzle und Bettgröße manuell eingetragen. Diesen Test habe ich mit PLA auf zwei verschiedenen 3D-Druckern durchgeführt: Dem Prusa i3 MK3S und dem Ultimaker 2+.

Cura liefert auf dem Ultimaker den saubersten Druck. Der Roboter hat eine sagenhaft ebene Oberfläche. Allerdings sitzen die Gelenke und der drehbare Kopf sehr fest. Wo Cura Stützfilament angelegt hat, ist allerdings eher fragwürdig: Aus den Augen kriegt man es nicht so einfach raus. Ein Cura-Druck fühlt sich auf dem Ultimaker natürlich heimisch, doch auf dem Prusa schneidet er auch gut ab. Die Gelenke sitzen lockerer, nur die Oberfläche ist nicht ganz so makellos wie auf dem Ultimaker. In beiden Fällen wäre nur ein Eingriff in die Support-Einstellungen nötig, um den Druck zu perfektionieren.

Vom **PrusaSlicer** habe ich mir auf dem Ultimaker mehr erhofft. Der Druck ist sehr ungleichmäßig, der Filamentfluss scheint nicht gepasst zu haben und die Gelenke sind nur schwer beweglich. Auf dem Prusa glänzt hingegen der eigene Slicer dafür: Der Druck ist sehr sauber und alle Gelenke können sich frei bewegen.

Nicht so toll: Per Default ist das Bett im PrusaSlicer bei PLA auf 60 Grad eingestellt, doch die erste Schicht wollte nicht haften. Erst nach Änderung auf 70 Grad läuft der Druck. Eine kleine Anpassung, doch dafür muss der

Anwender erstmal wissen – oder wahlweise recherchieren – was der Fehldruck zu bedeuten hat.

Kann mich meine Erinnerung an zuverlässige Druckergebnisse mit dem **Slic3r** denn so täuschen? Der Druck auf dem Ultimaker ist rillig und ungleichmäßig – und der Test auf dem Prusa ein glatter Reif. Schon nach wenigen Schichten gibt es reproduzierbare Stepverluste. Nach über 90 Sekunden Rechenzeit für einen dritten Versuch hebt sich der Druck direkt vom Bett ab und ich gebe auf. Hier müsste man noch viel Feinjustierung (zum Beispiel an den Lüftereinstellungen) betreiben, um einen funktionierenden Makey in Händen zu halten.

Was kann ich im Slicer als „Profi“ einstellen?

Nachdem nun klar ist, wie die drei Slicer mit den Standardeinstellungen laufen, kann ich mich in die Profieinstellungen wagen. Ich schaue mir die Features an, die bei den gewählten Slicern besonders vielseitig sind: Infill, Support und Skirt/Brim/Raft. Die Filamenteinstellungen klammere ich aus: Sie haben zwar einen großen Einfluss auf das Druckergebnis auf dem Prusa i3 MK3S, sind aber längst nicht so starr vorgegeben wie die genannten Features.

Infill

Slicer können massive Formen in Hohlkörper umrechnen, damit man Material, Zeit und Gewicht beim Drucken spart. Die Hohlform wird dann vom Slicer mit einer luftigen Struktur namens *Infill* gefüllt, um die Stabilität des Drucks und eine gleichmäßige Deckschicht zu gewährleisten. Dem Infill kann man Strukturen unterschiedlicher Stabilität zuweisen, von Wabenmuster über Spiralen bis zu Dreiecken.

PrusaSlicer und **Slic3r** stehen beim Infill noch auf dem selben Stand mit vielen kleinen Optimierungsoptionen: Neben den üblichen Füllmustern kann man einstellen, dass das

Infill mit einer höheren Schichtdicke als die Außenhülle gedruckt wird und so nicht in jeder Schicht gedruckt werden muss. Man kann *only infill where needed* einschalten, wenn die Füllung nicht durchgehend, sondern nur an einigen Stellen als Stützmaterial gedruckt werden soll. Zudem können die beiden Slicer kleine Teilflächen massiv drucken, genauso wie regelmäßige massive Zwischenschichten.

Cura hat diese Features ebenfalls und noch viel mehr: Mit *gradual infill steps* wird das Infill nach oben hin immer dichter, so dass man Filament spart, ohne Stabilität einzubüßen. Dazu gibt es in Cura einen praktischen Hack, um das Infillmuster und die Dichte an einzelnen Stellen zu ändern, etwa rund um ein Schraubloch, das eine stabile Umgebung benötigt. Dazu kann man einfach eine geometrische Form in Cura erzeugen, sie in das Modell setzen und ihr eigene Infill-Eigenschaften zuweisen. Für eine höhere Stabilität kann Cura das Infill in einer Schicht auch breiter (in mehreren Bahnen) drucken.

Support

Supportmaterial fügt ein Slicer dann in den Druck ein, wenn es freischwebende Überhänge und Brücken gibt, die der Drucker nicht bewältigen könnte. Die Stützstrukturen sollen



Stepverluste und Warping beim Slic3r: Das sieht aus wie der Tatort eines Verbrechens.



Der Tree Support ist nicht nur sehr elegant, sondern auch hohl und damit besonders materialsparend. (Modell: Entitled Goose von Ginny/Thingiverse)

möglichst wenig Material verbrauchen, optimal stützen und sich später leicht vom Werkstück trennen lassen. Beim Supportmaterial treibt die 3D-Drucker-Community die Entwickler besonders intensiv an. Sie wollen Materialverbrauch, Druckzeit und vor allem das fummelige Entfernen von Supportstrukturen immer weiter optimieren.

Bei **Curas** Standard-Stützstruktur hat man die Wahl, ob sich die Stützstruktur vom Druckbett oder von überall aufbaut. Daneben gibt es Stützblocker, mit denen man Bereiche markieren kann, an denen kein Support zugefügt werden soll. Individuellen Support kann man nur mit dem Plug-in *Custom Supports* einstellen. Zum Ausgleich bietet Cura den experimentellen *tree support* an. Er imitiert die Struktur eines Baumes und umschließt das Modell mit seinen Ästen. Das ist besonders bei organischen Formen praktisch.

Der **PrusaSlicer** bietet vier Support-Modi: *Keine Stützstrukturen*, *Überall Stützstrukturen*, *Stützstrukturen nur auf dem Druckbett* und *Stützstrukturen nur für Stützverstärker*. Stützverstärker sind ein praktisches Feature, mit denen man Stützstrukturen gezielt an einzelne Stellen am Modell setzen kann. Bislang geht das über geometrische Formen, die man dem Modell unterschiebt – da muss man aber etwas rumprobieren. Genauso kann man Stützblocker einbauen, wenn man an Stellen auf Stützstrukturen verzichten möchte.

Im **Slic3r** kann man auswählen, ob man Support generieren möchte und welchen Schwellenwert der Slicer berücksichtigen soll. Zudem hat man die Wahl, ob sich die Stützstruktur vom Druckbett oder von überall aufbaut.

An der Tabelle „Stützstrukturen im Vergleich“ kann man nicht nur erkennen, welchen Einfluss die Stützstrukturen haben, sondern auch, wie unterschiedlich die Druckzeiten der Modelle bei verschiedenen Slicern ausfallen. Die Prognosen sind relativ zuverlässig und Curas Drucke sind durch die Bank am schnellsten.

Rafts, Skirts und Brims

Rafts, skirts und brims sollen die Haftung der ersten Schicht auf dem Druckbett optimieren. Der Skirt ist ein einzelnes Band mit etwas Abstand um die Basis des Objekts. Ein Brim sind mehrere Linien um die Basis des Objekts, die die Auflagefläche erhöhen. Raft ist eine abnehmbare Basis, auf der das Objekt gedruckt wird.

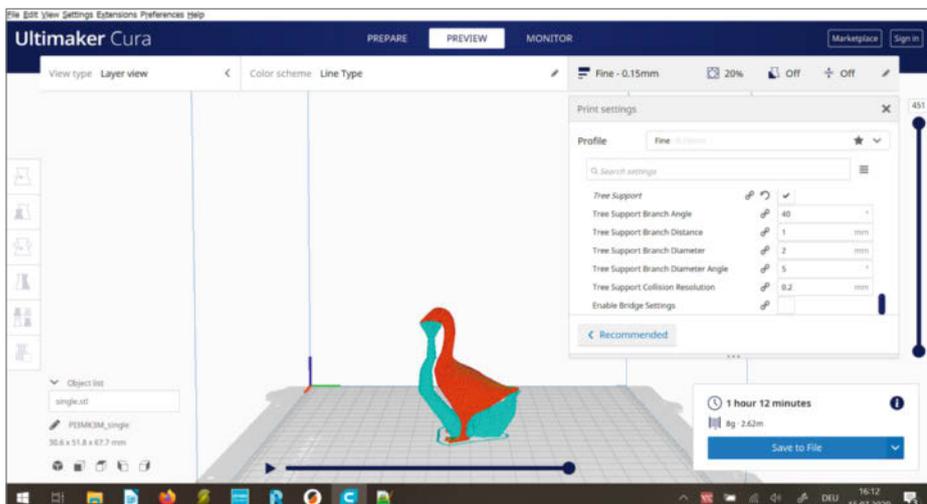
Der **PrusaSlicer** baut hier ganz auf die Erfahrung der **Slic3r-Entwickler**. Die haben zum Beispiel die niedlichen *Brim Ears* implementiert: Runde Bärenohrchen an den Ecken eines Objekts. Mit ihnen kann man den Druck bei minimalem Brim leicht vom Bett lösen. Man kann zudem die Breite des Brim festlegen. Beim Skirt gibt es bei beiden die Möglichkeit, Höhe und Abstand zum Druckobjekt zu definieren. PrusaSlicer kann dazu noch ein Draft Shield generieren: Eine Mauer um das Objekt, die den Druck vor Luftzug und Klimaänderungen schützt. PrusaSlicer und Slic3r können zwar Raft anlegen, Slic3r rät aber selbst im Handbuch davon ab. Nötig ist dieses Feature dank besserer Modelle und Drucker heutzutage kaum noch.

In Cura sind die Skirt- und Brim-Einstellungen großteils ein und derselbe Button. Man kann wie bei den anderen beiden Slicern Breite, Höhe und Abstand zum Objekt vorgeben. Bei Cura kann man zudem das praktische Feature *Brim nur auf der Außenseite des Objekts* aktivieren. Auch Cura kann Raft generieren.

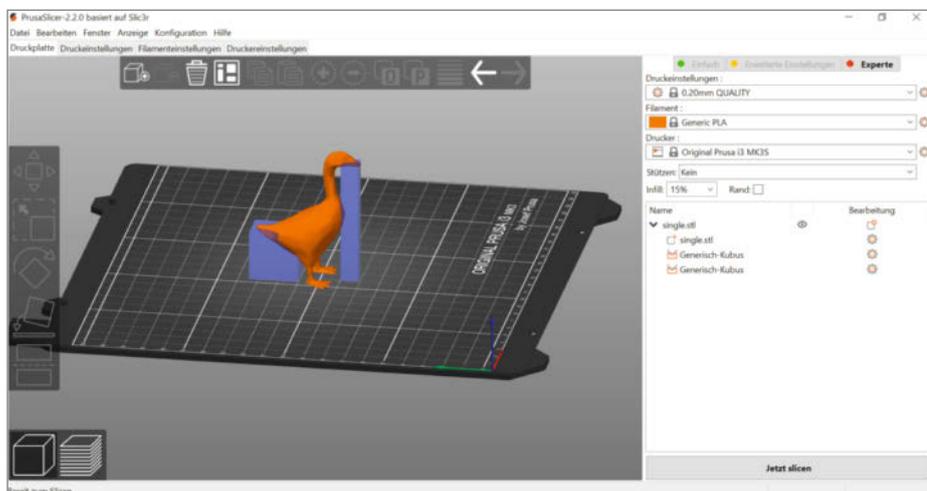
Überhang und Brücken

Überhang bewältigen alle drei Slicer gut, der Slic3r nur etwas gröber als die anderen beiden. Den Testdruck zum Bridging haben zwar auch alle Slicer gemeistert, doch Cura hat hier etwas schlechter abgeschnitten: Das Modell hat sich während des Drucks vom Bett gehoben und ein paar lose Filamentschnüre auf der Unterseite hinterlassen.

Cura hat dafür einige experimentelle Bridging-Features im Angebot. Im Internet raten viele User davon ab, sie zu einschalten, denn meist soll das Ergebnis nur schlechter werden. Wir haben das natürlich getestet: Die Brücken sehen im experimentellen Modus zwar anders aus, aber tatsächlich nicht besser.



Der Tree Support in Cura. Vorsicht: Vorher den normalen Support ausstellen, sonst gibt es einfach nur doppelte Stützen.



Hier habe ich im PrusaSlicer zwei individuelle Blöcke Supportmaterial eingefügt.

Stützstrukturen im Vergleich

Software	Cura	Cura	Cura	PrusaSlicer	PrusaSlicer	PrusaSlicer
Support-Modus	Kein Support	Standard Support	Baumstruktur	Kein Support	Standard Support	Manuelle Stützverstärker
Dauer	58min	82min	72min	70min	119min	117min
Materialverbrauch	6g/2,17m	9g/3,09m	8g/2,26m	7g/2,47m	11g/3,66m	10,5g/3,56m

Reparatur-Funktionen

Modelliert man eigene Modelle, kann es passieren, dass man unmögliche Geometrien generiert. An diesen Stellen sieht das Modell zwar richtig aus, es überlagern sich aber mehrere nicht verbundene Vektorpunkte, die einen 3D-Drucker gehörig irritieren würden. Gab es früher noch Zusatzsoftware, die das Druckobjekt korrigiert hat, ist diese Funktion mittlerweile in vielen Slicern integriert. Ein Dienst hierfür ist zum Beispiel NetFabb. Für diesen Test habe ich ein selbstmodelliertes 3D-Modell mit mehreren Fehlern verwendet. Beim Druck auf dem Prusa i3 MK3S schneiden alle drei Slicer überraschend gut ab:

Der **Slic3r** bietet ein sehr akzeptables Ergebnis. Die obersten Schichten sind etwas löchrig geworden und die Seitenwände haben ein paar Macken, aber die fehlerhafte Mittelachse der Treppenstufen kann man nur erahnen.

Der **PrusaSlicer** hat das schönste Ergebnis produziert. Die Oberflächen sind ebenmäßig und man kann dem Druck auf keinen Fall ansehen, dass der Slicer mit einem fehlerhaften Modell gefüttert wurde.

Cura schneidet ebenfalls gut ab. Die Oberfläche ist nicht ganz gleichmäßig und man kann wie beim Slic3r erahnen, dass die Mittelpunkte der Stufen sich überlappen, doch das sind schon sehr feine Details.

Fazit

Für mich gewinnt der PrusaSlicer in der Rubrik „Bedienoberfläche“. Beim Druckergebnis haben alle Slicer ihre Stärken und Schwächen, doch der Cura überzeugt mit schneller Rechen- und Druckzeit.

Besonders empfehlenswert für Einsteiger und Vieldrucker mit wenig Interesse an Profieinstellungen ist der Cura-Slicer. Wer sich dagegen gerne richtig reinfuchst, dem lege ich

den PrusaSlicer ans Herz. Der Slic3r müsste allerdings ordentlich aufholen, um sich mit Cura und PrusaSlicer messen zu können.

Man kann deutlich erkennen, wie groß die Entwicklung von Slicer-Software in den letzten Jahren war – und wie schlecht heutzutage die Drucke wirken, die vor zwei Jahren noch ganz akzeptabel waren. Als ich mich bei einer Freundin über die schlechte Druckqualität eines Slic3r-Drucks beklagte, hat sie mich prompt als „verwöhnt“ bezeichnet. Umso schöner ist es, wie wenig Nachbearbeitung heute noch nötig ist, um einen 3D-Druck vorzeigbar zu machen.

Ob und wie der Slic3r nach Adoption durch Prusa noch eigenständig weiterentwickelt wird, ist nicht klar, aber ich bin sehr gespannt, was sich die Teams von Cura und der Prusa Slicer in der Zukunft ausdenken.

Weitere Entwicklungen

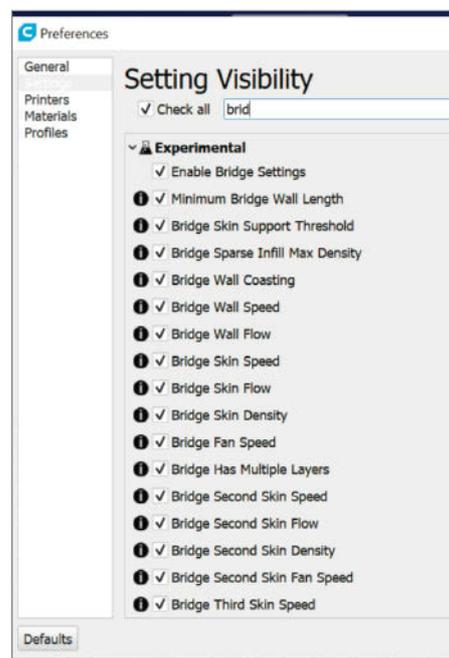
Die Slicer-Teams haben bereits neue Features für die Zukunft angekündigt. Besonders angetrieben werden sie von der 3D-Druck-Community, die klare Vorstellungen hat, welche Funktionen ihr fehlen (wie individueller Brim bei mehreren Objekten).

Der PrusaSlicer 2.3.0 soll bald erscheinen (Stand Juli 2020) und wird noch ausgefeiltere Funktionen für individuellen Support haben. In Zukunft kann man sich anzeigen lassen, welche Bereiche des Modells ab einem gewählten Schwellwert Stützmaterial benötigen und mit einem Pinsel direkt Flächen ergänzen oder entfernen.

Cura präsentiert schon in der neuen Beta 4.6 von April 2020 ihre erweiterten *Intent Profiles*. Mit diesen Profilen kann man das Ziel eines Drucks (etwa besonders glatt oder besonders präzise Maße) angeben und schon passen sich die Feineinstellungen dieser Vorgabe an. Bisher gab es dieses Feature nur für Ultimaker-Drucker, in Zukunft wird die Bandbreite ordentlich vergrößert. —rehu



Das im PrusaSlicer manuell angelegte Supportmaterial sieht gut aus.



Die experimentellen Bridge-Settings in Cura.



Das Druckergebnis nach der STL-Reparatur: Cura, PrusaSlicer und Slic3r

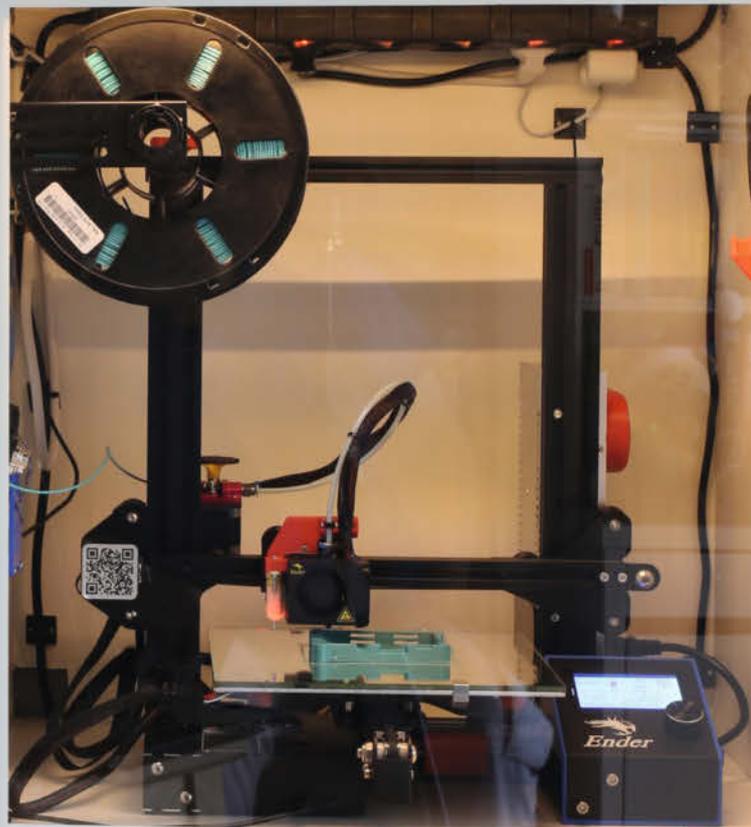
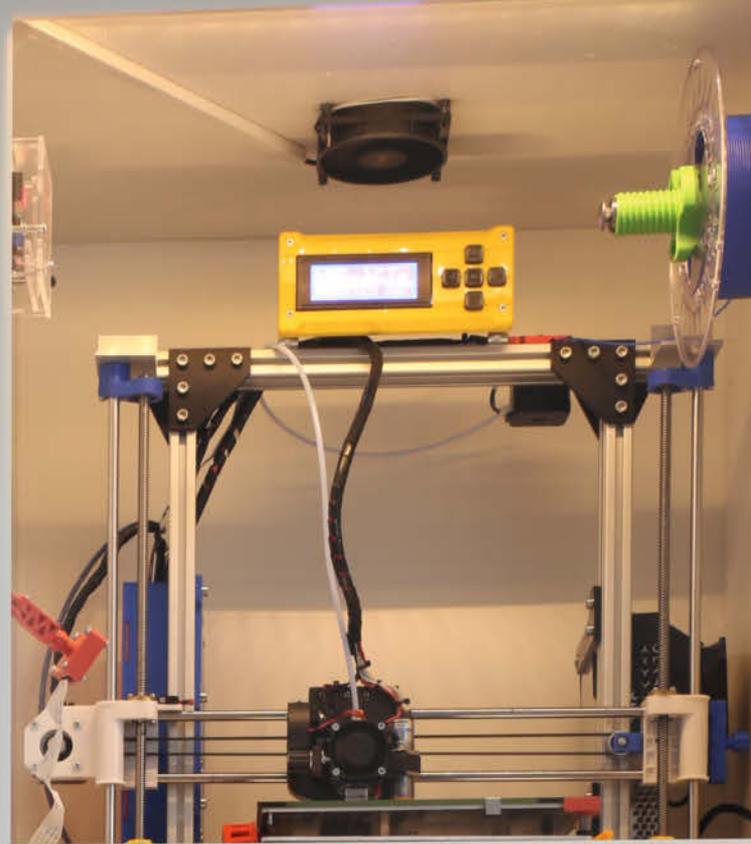
Features

Software	Cura	PrusaSlicer	Slic3r
STL-Reparatur	Ja	Ja	Ja
Infill-Muster	13	13	13
Raft/Brim/Skirt	Ja/Ja/Ja	Ja/Ja/Ja	Ja/Ja/Ja
Individuelle Stützen/Stützoptionen	Ja	Ja	Nein
Anzeige Materialverbrauch/Druckdauer	Ja/Ja	Ja/Ja	Ja/Nein

Druckserver für mehrere 3D-Drucker

Nicht nur Fablabs, Schulen und Firmen besitzen mehrere 3D-Drucker, auch daheim kommt das, wie wir von zahlreichen Lesern wissen, immer öfter vor. Praktisch wäre es doch, alle zentral über ein Netzwerk mit Druckdaten zu versorgen, zu steuern und zu kontrollieren. Wir zeigen, welche Lösungen es dafür gibt, was sie können und wie man sie einrichtet.

von Heinz Behling



Seitdem es 3D-Drucker-Bausätze bereits ab 100 Euro gibt, findet man nicht nur in Fab-labs und anderen offenen Werkstätten, sondern auch im Privatbesitz häufig mehrere Drucker. Schließlich ist dies die effektivste Methode, bei mehrteiligen Druckobjekten die Herstellungszeit drastisch zu verkürzen: So kann ein Drucker zum Beispiel das Gehäuseunterteil, der andere den Deckel formen und 16 Stunden Druckzeit schrumpfen zu nur noch 8.

Doch gerade preiswerte Drucker erfordern mangels Netzwerk-Anschluss mühsames Hantieren mit Speicherkarte oder USB-Sticks, um die Druckdateien in den Printer zu bekommen. Und ständig der Gang zum Gerät, nur um zu schauen, ob noch alles richtig und gefahrlos läuft, das ist lästig und im Netz-Zeitalter geradezu altertümlich. Es gibt jedoch Abhilfe: Druckserver, also kleine Computer, an denen die Drucker angeschlossen sind, und die nicht nur als Netzwerk-Verbindungsbrücke dienen, sondern auch zur Kontrolle und Überwachung der Geräte aus der Ferne (Arbeitsplatz oder Wohnzimmer), idealerweise mit Live-Kamerabild vom Ort des Druckgeschehens.

Kleiner Computer? Da drängt sich doch ein Raspberry Pi 3 geradezu auf. Zum Druckserver wird er aber erst durch die richtige Software. Vorherrschend in diesem Bereich sind *Octo-Print* und *Repetier-Server*. Beide gibt es nicht nur für Linux (inkl. Raspbian), sondern auch für Windows und macOS. Ein großer Windows-PC oder ein Mac wären für diese Aufgabe aber deutlich überdimensioniert, es sei denn, Sie nennen eine Druckerfarm Ihr Eigen. Solche Computer würden auch nur unnötig Strom verbrauchen. Daher halten wir nur die Raspi-Versionen für sinnvoll, also *OctoPi* und *Repetier-Server 0.94.0 for Raspberry Pi*, die jeweils außer der eigentlichen Server-Software auch noch ein Raspberry-Betriebssystem enthalten und als Speicherkarten-Image zum Download zur Verfügung stehen (siehe Kurzinfo-Link).

Aber welche Software ist die bessere? Wir habens ausprobiert. Im Folgenden schildern wir kurz die Vor- und Nachteile der beiden Kandidaten und zeigen, wie man sie im Hinblick auf den Anschluss von mehr als einem Drucker zum Laufen bringt.

Die grundlegenden Aufgaben, also die Verbindung der Drucker mit den Benutzern übers Netz, erfüllen beide problemlos. Auch die dauerhafte Speicherung der Druckjobs sowie das Führen einer Druck-Historie beherrschen sie. Unterschiede gibt es aber bei der Installation, in den Überwachungsfunktionen und in den Erweiterungsmöglichkeiten. Sie sind dadurch für unterschiedliche Ansprüche und Vorkenntnisse geeignet. Im Folgenden schildern wir ihre Fähigkeiten und zeigen danach, wie Sie die Druckserver-Distributionen zur Zusammenarbeit mit mehreren Druckern installieren und konfigurieren müssen.

Kurzinfo

- » Vergleich OctoPrint und Repetier-Server
- » Printserver-Installation für mehrere Drucker
- » Zwei oder mehr Webcams einrichten

Checkliste

-  **Zeitaufwand:**
2 Stunden
-  **Kosten:**
100 Euro (für 2 Drucker)
-  **Programmieren:**
Anpassen von Skript-Dateien
-  **3D-Druck:**
Gehäuse aus Download-Druckdateien

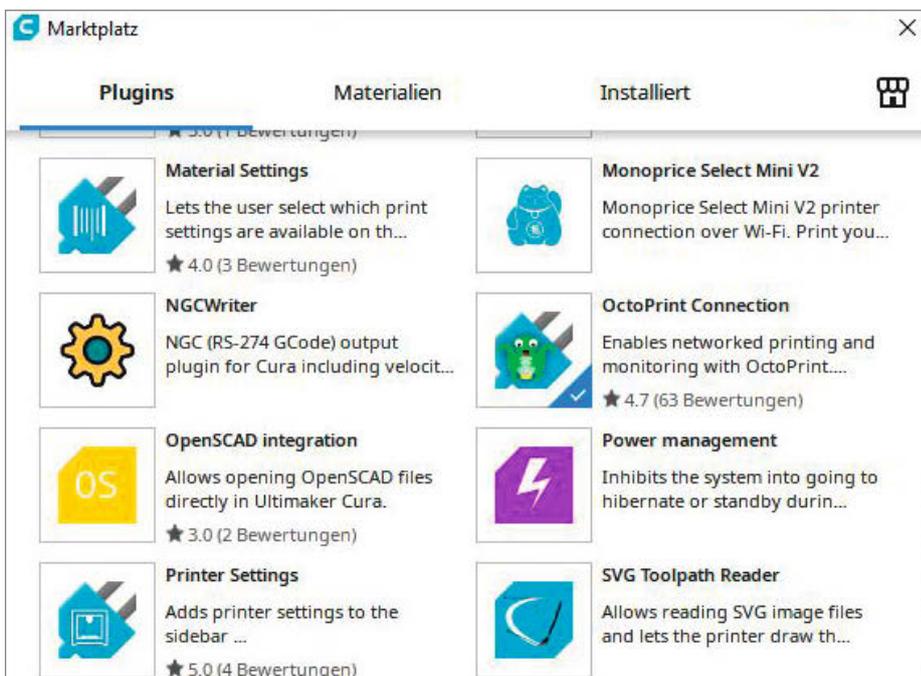
Material

- » Raspberry Pi empfohlen Pi 3B, 3B+, 4B 1/2/4GB, kein Zero
- » Micro-SD-Speicherkarte mind. 8GB, empfohlen 32GB
- » Netzteil 5V/3000 bis 4000mA, Micro-USB-Stecker, für RasPi 4: USB-C-Stecker
- » USB-Kabel USB-A-Stecker auf Druckeranschluss, pro Drucker ein Kabel
- » Raspi-Kameramodul mit Kabel passender Länge
- » USB-Kamera

Mehr zum Thema

- » Mehr zum Thema 3D-Druckserver erfahren Sie im Artikel „Komfortdrucker“ in der Make 2/18 ab Seite 120.
- » Mehr zum Thema Netzwerk-Sicherheit von 3D-Druckservern lesen Sie im Online-Artikel „OctoPrint-Chefentwicklerin: 3D-Drucker gehören nicht ins öffentliche Netz“
- » Eine Anleitung zum Anlegen von udev-Regeln finden Sie im Online-Artikel „RasPi: Feste USB-Schnittstellen-Namen durch udev-Regeln“

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



Im Marktplatz des Slicers Cura gibt es ein Plug-in, mit dem Sie Druckdateien direkt zum Octo-Print-Druckserver senden und rudimentäre Funktionen des Druckers in Cura überwachen können. Ein entsprechendes Repetier-Plug-in steht auf Github bereit (siehe Kurzinfo-Link).

OctoPi

OctoPi ist eine kostenlose Distribution, die den Open-Source-Druckerserver OctoPrint enthält. Er nimmt Druckdateien per Datei-Upload in seiner Web-Oberfläche oder direkt vom Slicer-Programm wie Cura entgegen und speichert sie dauerhaft auf der Speicherkarte des RasPi oder des Druckers. Überwacht wird der Drucker durch Temperaturdiagramme ①, Anzeige des gerade gedruckten Layer und ein Live-Kamerabild, das auf Wunsch auch als Zeitraffer-Film gespeichert werden kann. Die Temperatur und Versorgungsspannung des Rasperrys werden überwacht und gegebenenfalls Alarmsymbole angezeigt. Durch die mitgelieferte Cura-Engine kann die Software auch slicen.

Das zum Mitmachen einladende Open-Source-Konzept (siehe Slicer-Vergleich auf Seite 90) macht sich bei den Erweiterungen bemerkbar: Eine Menge von Anwendern entwickelter Plug-ins steht zur Verfügung, die den Druckserver und auch den Drucker um interessante Funktionen erweitern, etwa zur Kalibrierung, Filamentwechsel, Ein-/Ausschalten des Druckers, Anzeige von IP-Adressen oder Druckzeiten, Sprachausgabe übers Web und vieles mehr. Außerdem gibt es auf den Octo-

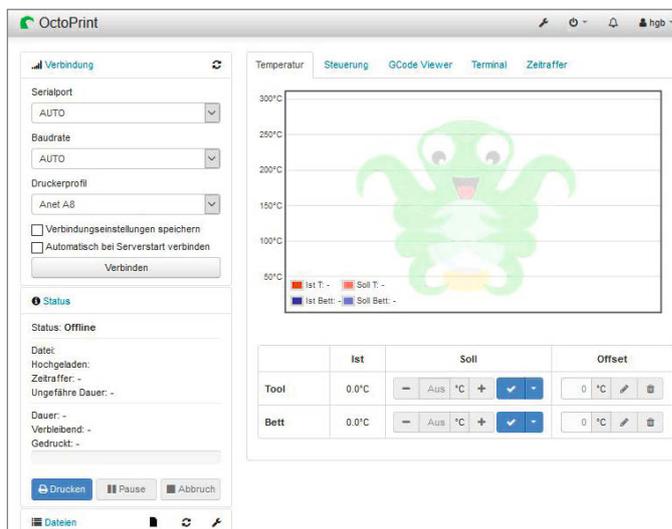
print-Seiten einen Workshop, mit dessen Hilfe man eigene Plug-ins programmieren kann. Als Programmiersprache wird Python benutzt, das relativ leicht erlernbar ist.

Der Zugriffsschutz wird durch Benutzernamen und Passwort verwirklicht. Das ist natürlich durch Brute-force-Angriffe knackbar. Deshalb sollte der Drucker nicht für den Netzzugang von außen freigegeben werden.

Größter Nachteil aber: OctoPi kann von

Haus aus nicht mit mehreren Druckern zugleich arbeiten. Erst durch die nachträgliche Installation je einer zusätzlichen OctoPrint-Instanz pro zusätzlichem Drucker von Hand wird dies möglich. Schwierig, aber durchaus lohnend – auch ohne Computer-Erfahrung, denn hier erfahren Sie, wie es geht.

① Das OctoPrint-Fenster ist übersichtlich. Die Bedienung erfolgt per Maus. Eine Touchscreen-fähige Webseite wird bei Erkennen eines solchen Endgeräts automatisch erzeugt.



Repetier-Server

Was Dateiübertragung, Drucker- und RasPi-Überwachung und Bedienung angeht, ist Repetier-Server durchaus gleichwertig mit OctoPi. Mit Ausnahme der Kamerafunktion: Die beherrscht die kostenlose Version dieser Software nämlich nicht. Erst die für 60 Euro erhältliche Kaufversion kann das und ist dann selbst für mehrere Drucker sehr einfach einzurichten. Repetier-Server ist nämlich von Haus aus fähig, nicht nur mit einem Drucker zu arbeiten. Dafür sind bei der Installation keine Klimmzüge nötig. Dementsprechend ist es schnell installiert und in diesem Punkt OctoPi deutlich überlegen.

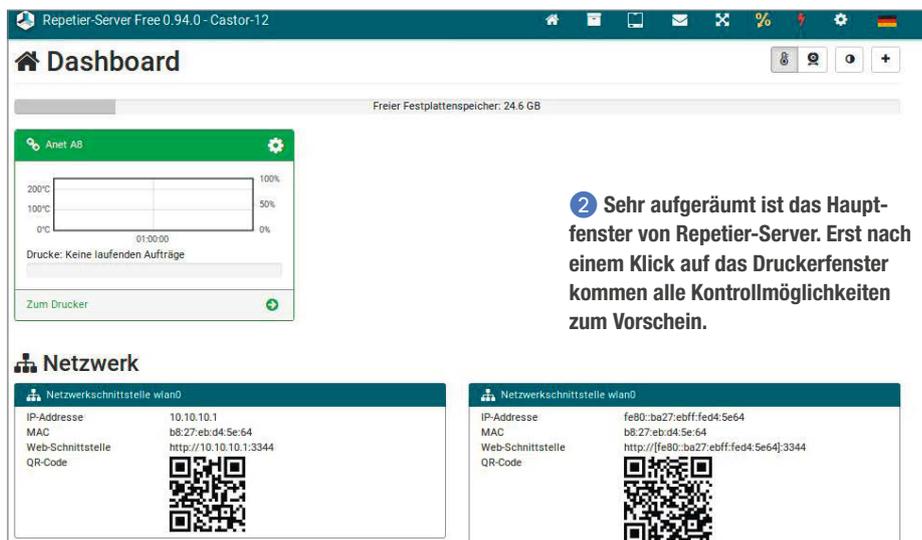
Beim ersten Start stellt der Druckserver einen eigenen WLAN-Accesspoint bereit, über den man sich einloggen und die Installation inklusive der Einstellung der eigenen WLAN-Zugangsdaten vornehmen kann. All das geht wirklich intuitiv vor sich, da sämtliche Schritte gut erklärt sind. Aber die Hinweise und Fragestellungen muss man auch sorgsam lesen. Beispiel: Wenn man beim Anlegen eines Druckerprofils einfach die vorgegebene Repetier-Firmware übernimmt, statt die korrekte Version des Geräts zu wählen, darf man sich über Kommunikationsprobleme nicht wundern.

Die Software ist nicht Open Source. Wohl mit ein Grund, warum es bei Erweiterungen deutlich schlechter aussieht als OctoPrint. Repetier-Server erlaubt sie zwar auch, nämlich als Skripte,

die in LUA geschrieben sind und im Verzeichnis `var/lib/Repetier-Server/lua` abgelegt werden. Der Server führt sie dann laufend aus (Herstellerangabe: 100mal pro Sekunde). Auf diese Weise lassen sich beispielsweise per GPIO-Pins zusätzliche Schalter abfragen, LEDs oder Relais schalten, Druckerdaten abfragen und vieles mehr. Allerdings gibt es kein offizielles Plug-in-Repository. Man muss daher die Internet-Suche bemühen oder selbst programmieren.

Die Web-Oberfläche ist schlicht, aber funktionell ②. Einen mitgelieferten Slicer hat Repetier nicht, lediglich ein Cloud-Slicer soll laut Internet-Seite des Herstellers kommen. Da aber die meisten 3D-Druckerbesitzer ohnehin ein entsprechendes Programm besitzen, ist das kein großer Nachteil.

Wer schnell und auf möglichst einfachem Weg einen Druckserver in Betrieb nehmen möchte, dem sei Repetier aber ans Herz gelegt.



② Sehr aufgeräumt ist das Hauptfenster von Repetier-Server. Erst nach einem Klick auf das Druckerfenster kommen alle Kontrollmöglichkeiten zum Vorschein.

Installationsvorbereitung

Bevor Sie mit der Installation einer der beiden Server beginnen, sollten Sie die Drucker und Kameras (Liste mit geeigneten Kameras siehe Kurzinfo-Link) mit dem Raspberry verbinden und für eine ausreichende Stromversorgung sorgen. Der Druckerserver ist da nämlich anspruchsvoll. Durch die Belastung der USB-Ports und den Rechenaufwand für die Videos läuft das Netzteil des Rasperrys nahezu permanent mit hoher Last. Ist es zu schwach oder das USB-Kabel hat einen zu kleinen Leiterquerschnitt, kommt es später immer wieder zu Verbindungsabbrüchen im Netzwerk und zu



3 Nicht gerade VDE-gerecht, aber mit 4A leistungsfähig genug für den Druckerserver-RasPi.

den Druckern. Das kann Druckaufträge empfindlich stören. Bei meinem Drucker-Ensemble brachte erst ein 4A-Netzteil 3 mit einem nur 30cm langen USB-Kabel eine ausdauernde Stromversorgung zustande. Auch für eine gute Kühlung, am besten mit einem Lüfter, sollte man daher sorgen, insbesondere, falls der Drucker zusammen mit den Druckern in einem Schrank untergebracht ist. Sowohl OctoPrint als auch Repetier-Server zeigen Unterspannung wie auch erhöhte Temperaturen mit Warnsymbolen an.

Dann sollten Sie das entsprechende Drucker-server-Image auf eine Speicherkarte schreiben. Dieser Punkt unterscheidet sich nicht von der Installation anderer Distributionen für Rasperrys. Die gepackten Images sind über den Download-Link erhältlich. Nach dem Übertragen auf Ihren Computer und dem Entpacken schreiben Sie das Image der gewünschten Distribution auf die Speicherkarte. Wir verwendeten den *Raspberry Pi Imager*, den es für alle drei Betriebssysteme gibt. Stecken Sie die Micro-SD-Karte in den Leser, wählen Sie im Programm die Karte und das Image aus und starten Sie den Schreibvorgang mit einem Klick auf *write* 4.



4 Der RaspberryPi Imager ist das Universalwerkzeug für Speicherkarten.

Auf den nächsten Seiten zeigen wir die Installation beider Distributionen. Warum beide? Nun, beide funktionieren problemlos. Ich persönlich bevorzuge zwar OctoPi wegen der vielen Plug-ins. Aber dieses Urteil ist rein subjektiv. Falls Sie auf solchen „Schnickschnack“ und auch auf die kostenlose Kameraunterstützung verzichten können, ist der Repetier-Server das Mittel der Wahl, denn einfacher geht es kaum. Das entsprechende Installations-Kapitel ist angenehm kurz, da nahezu alles selbsterklärend ist. OctoPi erfordert da deutlich mehr Lektüre. Falls Sie sich dafür entschieden haben, machen Sie bitte auf Seite 102 weiter.

Repetier-Server-Einrichtung

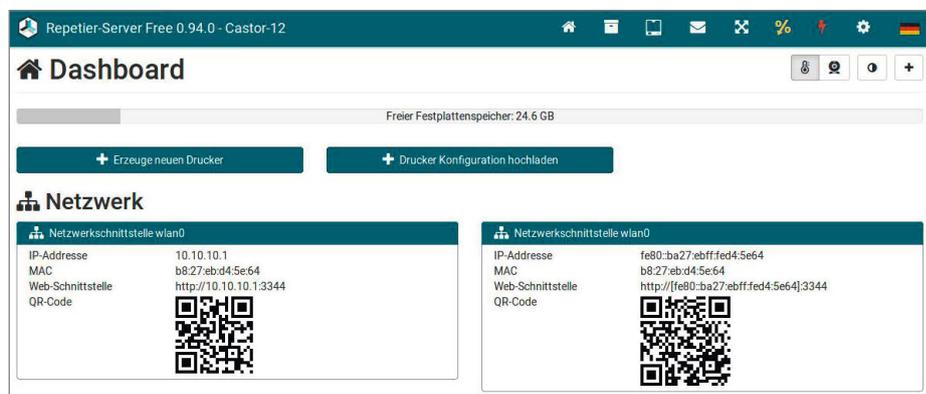
Starten Sie den Raspberry mit der Speicherkarte. Beim ersten Start arbeitet er als WLAN-Accesspoint. Falls Sie also mit Ihrem Computer über WLAN aufs Netz zugreifen, wechseln Sie zum Netz mit der ssid *RepetierServer* und dem Passwort *12345678*. Andernfalls verbinden Sie den RasPi per Netzwerkkabel oder arbeiten mit angeschlossenem Monitor, Tastatur und Maus. Dass war aber auch schon die schwierigste Hürde, denn jetzt können Sie bereits im Browser über die Adresse

<http://repetier-server.local/>

auf die Weboberfläche zugreifen 5.

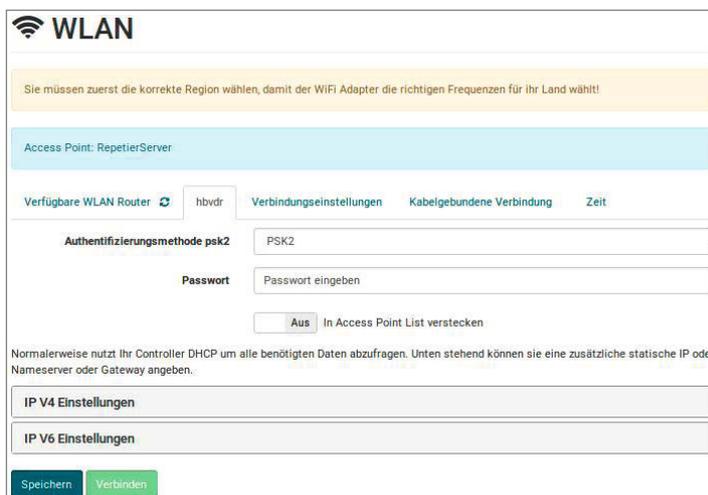
Hinter dem Zahnrad in der Symbolleiste finden Sie die *Globalen Einstellungen* und darin die Netzwerk-Konfiguration. Geben Sie dort Namen und Passwort Ihres WLANs ein und ganz wichtig: Wählen Sie die korrekte Region (Land), sonst startet der WLAN-Zugang nicht 6.

Danach können Sie den RasPi neu starten und dann im eigenen WLAN über die oben genannte http-Adresse auf den Server zugreifen und die Druckerprofile anlegen. Auch das ist selbsterklärend. Achten Sie aber darauf, dem Drucker die jeweils richtige Firmware-Version zu wählen. All dies bedarf keiner großen Anleitung, weshalb wir uns hier auch zurückhalten.



5 Der Accesspoint des Repetier-Servers ermöglicht sofort den Zugang zur Web-Oberfläche.

6 Beim WLAN-Zugang unbedingt unter Region das richtige Land wählen.



Einrichten von OctoPi

Falls Sie in OctoPi WLAN benutzen möchten, sollten Sie die Speicherkarte nach dem Schreiben des Images noch nicht entnehmen, sondern zunächst die Daten Ihres Netzwerks in die Datei `octopi-wpa-supPLICANT.txt` eintragen. Es geht dabei um den Netzwerknamen `ssid` und das Passwort `psk` 7. Entfernen Sie auch die Doppelkreuze vor den Zeilen 7 und 8.

Dann legen Sie die Karte in den Raspi ein und starten ihn. Nach etwa ein bis zwei Minuten sollte er Verbindung mit dem Netzwerk haben. Sie können ihn nun vom PC oder Mac mit dieser Adresse per Browser erreichen:

`http://octopi.local`

In der Web-Oberfläche unseres Druckers erscheint der Installations-Wizard mit dem Begrüßungsbildschirm. Klicken Sie auf *Weiter* und machen Sie die jeweils geforderten Angaben. Wir empfehlen dringend, im zweiten Schritt die Zugangsbeschränkung zu aktivieren und Benutzernamen sowie Passwort einzugeben, denn schließlich wird der Server übers Netz für jeden Benutzer zugänglich sein. Im Router Ihres Netzes sollte der Zugang vom Internet aus nicht gestattet werden. Da solch ein Zugangsschutz relativ leicht zu hacken ist, könnten dann andere die Kontrolle über Ihre Drucker übernehmen.

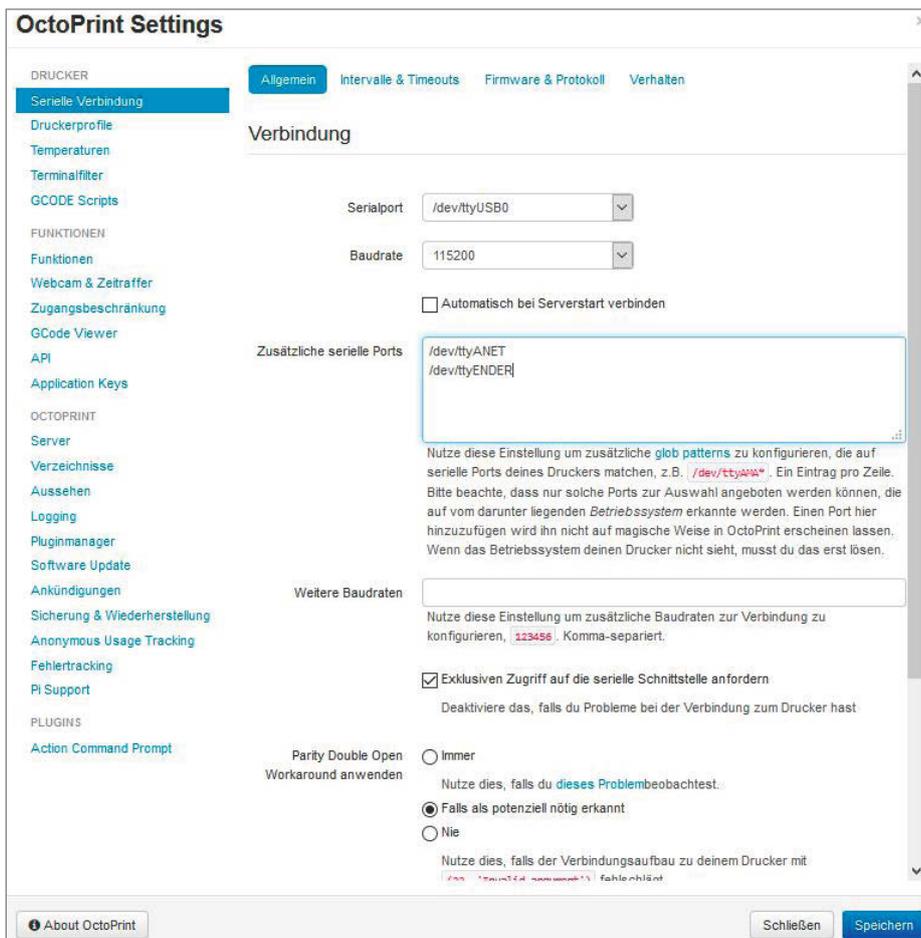
Im sechsten Fenster müssen Sie ein Druckerprofil für den ersten Drucker eingeben. Beachten Sie die zusätzlichen Reiter für *Druckbett & -volumen*, *Achsen* sowie *Hotend & Extruder*. Sobald das alles erledigt ist, gelangen Sie ins OctoPrint-Hauptfenster. Mit einem Klick auf den Schraubenschlüssel in der Symbolleiste wechseln Sie zu den *Settings*. Dort legen Sie unter *Druckprofile* nun den zweiten Drucker an. Vergessen Sie das *Speichern* am Schluss nicht.

```
## WPA/WPA2 secured
network={
  ssid=""
  psk=""
}
```

7 Entfernen Sie die Doppelkreuze vor den Zeilen von `network` bis zur schließenden geschweiften Klammer. Ergänzen Sie die Angaben zu Ihrem WLAN.

```
# Uncomment the country your Pi is in to activate Wifi in RaspberryPi 3 B+ and above
# For full list see: https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2
country=GB # United Kingdom
#country=CA # Canada
#country=DE # Germany
#country=FR # France
#country=US # United States
```

8 Auch vor der entsprechenden `country`-Zeile entfernen Sie das Doppelkreuz.



9 Die neuen Schnittstellen müssen OctoPrint erst bekannt gemacht werden.

Nun wird es Zeit, den Druckern eindeutige Schnittstellennamen zuzuordnen, also Bezeichnungen, die sich nicht bei jedem Bootvorgang oder jedem Neuanschluss ändern. Dazu öffnen Sie über ssh ein Konsolenfenster. Als ssh-Client empfehlen wir das kostenlose Programm *putty* (Download siehe Kurzinformativ-Link). Als Adresse gilt auch hier `octopi.local`. Im Konsolenfenster müssen Sie nun eine `udev`-Regel entwickeln. Eine Anleitung dazu finden Sie im Online-Artikel „RasPi: Feste USB-Schnittstellen-Namen durch `udev`-Regeln“ (siehe Kurzinformativ-Link). Als Schnittstellenbezeichnung verwende ich im Folgenden `ttyANET` und `ttyENDER`.

Diese Schnittstellen müssen Sie OctoPrint nun noch unter *zusätzliche serielle Ports* mitteilen 9. Speichern nicht vergessen.

Danach sollten sich im Verbindungsbereich die neuen Schnittstellen auswählen lassen. Wählen Sie beispielsweise das Druckerprofil *Creativity Ender* und die dazugehörige Schnittstelle 10.

Ob dahinter auch der entsprechende Drucker steckt, lässt sich schnell testen. Klicken Sie dazu auf *Verbinden* und wechseln Sie in den Reiter *Terminal*. In die Eingabezeile links von *Senden* geben Sie den Befehl `G28` ein. Das sollte ein Drucker-Homing bewirken, in diesem Fall am Ender. Falls der falsche Drucker reagiert, haben Sie sich beim Anlegen der `udev`-Regel geirrt. Stimmt diese Zuordnung, trennen Sie die Verbindung und probieren Sie den (oder die) anderen Drucker.

Zweite Instanz

Die Schwäche, dass OctoPrint nicht gleichzeitig mit mehreren Druckern arbeiten kann, umgehen wir geschickt mit der Installation



zusätzlicher Instanzen im Konsolenfenster. Mit 11 stellen Sie eine Kopie der OctoPrint-Konfigurationsdateien für die zweite Instanz her.

Anpassen müssen Sie sie aber noch. Mit

```
sudo nano /etc/default/octoprint2
```

öffnen Sie diese Konfigurationsdatei. Ändern Sie dort die Port-Adresse in

10 Nun wird getestet, ob die Schnittstellen den richtigen Drucker zugeordnet wurden.

```
PORT=5001
```

und erweitern Sie die `DAEMON_ARGS`-Zeile um das Verzeichnis der neuen Instanz 12.

Für die erste Instanz müssen wir das auch machen. Also mit

```
sudo nano /etc/default/octoprint
```

deren Konfigurationsdatei öffnen und darin die Zeile so ändern wie in 13.

Start-Automatik

Jetzt müssen wir noch dafür sorgen, dass die zweite Instanz auch beim Booten mit gestartet wird. Dazu kopieren wir das Startscript der ersten Instanz mit

```
sudo cp /etc/init.d/octoprint /etc/init.d/octoprint2
```

und passen es an:

```
sudo nano /etc/init.d/octoprint2
```

Außer in der Zeile

```
DAEMON=/usr/bin/octoprint
```

ergänzen wir überall

```
octoprint
```

zu

```
octoprint2
```

Dieses neue Script müssen wir dem System noch bekannt machen mit

```
sudo systemctl daemon-reload
```

```
sudo update-rc.d octoprint2 defaults
```

Wir können es gleich ohne Booten starten:

```
sudo /etc/init.d/octoprint2 start
```

Für jeden weiteren Drucker müssen Sie diese Prozedur wiederholen, die neuen Instanzen aber anders nennen (*octoprint3* usw.).

Jetzt haben wir eine zweite (dritte, ...) OctoPrint-Instanz laufen, von der allerdings noch nichts nach außen dringt. Dazu müssen wir den in OctoPi arbeitenden HaProxy anpassen. Wir machen ein Backup der Konfiguration.

```
sudo cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.old
```

Dann beginnt die Änderung mit

```
sudo nano /etc/haproxy/haproxy.cfg
```

11 Kopieren der OctoPrint-Konfiguration

```
cp -R /home/pi/.octoprint /home/pi/.octoprint2
sudo cp /etc/default/octoprint /etc/default/octoprint2
```

12 Anpassung für die zweite Instanz

```
DAEMON_ARGS="--host=$HOST --port=$PORT --config /home/pi/.octoprint2/config.yaml --basedir /home/pi/.octoprint2/"
```

13 Änderung für die erste Instanz

```
DAEMON_ARGS="--host=$HOST --port=$PORT --config /home/pi/.octoprint/config.yaml --basedir /home/pi/.octoprint/"
```

Tools für Maker



PORTOFREI
AB 15 €
BESTELLWERT

 heise shop

shop.heise.de/tools



© Copyright by Maker Media GmbH.

Bestellen Sie ganz einfach online unter shop.heise.de oder per E-Mail: service@shop.heise.de

14 Backend für den ersten Drucker (Anet)

```
backend anet
  reqrep ^([\^ :])\ /anet/(.) \1\ /\2
  option forwardfor
  server octoprint 127.0.0.1:5000
  acl needs_scheme req.hdr_cnt(X-Scheme) eq 0
  reqadd X-Scheme:\ https if needs_scheme { ssl_fc }
  reqadd X-Scheme:\ http if needs_scheme !{ ssl_fc }
  reqadd X-Script-Name:\ /anet
```

15 Backend für den zweiten Drucker (Ender)

```
backend ender
  reqrep ^([\^ :])\ /ender/(.) \1\ /\2
  option forwardfor
  server octoprint2 127.0.0.1:5001
  acl needs_scheme req.hdr_cnt(X-Scheme) eq 0
  reqadd X-Scheme:\ https if needs_scheme { ssl_fc }
  reqadd X-Scheme:\ http if needs_scheme !{ ssl_fc }
  reqadd X-Script-Name:\ /ender
```

Für jeden Drucker müssen wir hinter

`option forwardfor except 127.0.0.1`
eine Zeile nach diesem Muster einfügen:

```
use_backend Druckerbezeichnung if {
  path_beg /Druckerbezeichnung/ }
```

Für *Druckerbezeichnung* setzen Sie bitte etwas Sinnvolles ein, etwa *anet* oder *ender*.

Die beiden Zeilen

```
use_backend webcam if { ... }
```

```
default_backend octoprint
```

ersetzen wir durch

`default_backend webcam`

Danach definieren wir die Backends für jeden Drucker nach diesem Muster für den Anet 14 und für alle weiteren Drucker, also zum Beispiel für den Ender 15.

Die Druckerbezeichnungen müssen dabei dieselben sein wie in den *use-backend*-Zeilen weiter oben. Die Bezeichnung hinter *server* muss dem Namen der jeweiligen Octoprint-Instanz entsprechen. In diesem Beispiel ist es *octoprint2*.

Das Backend *octoprint* entfernen wir ganz. Dann bitte die Datei speichern. Damit wir alle OctoPrint-Instanz(en) notfalls über die Web-

Oberfläche neu starten können, müssen wir pro Instanz noch eine Datei bearbeiten:

```
nano /home/pi/.octoprint2/config.yaml
```

Darin ändern wir die Zeile mit dem *serverRestartCommand* in

```
serverRestartCommand: sudo service octoprint2 restart
```

Nach einem Neustart des RasPis steht nun die zweite Instanz zur Verfügung. Probieren Sie es aus: Starten Sie im Browser die erste mit

```
http://octopi.local/anet/
```

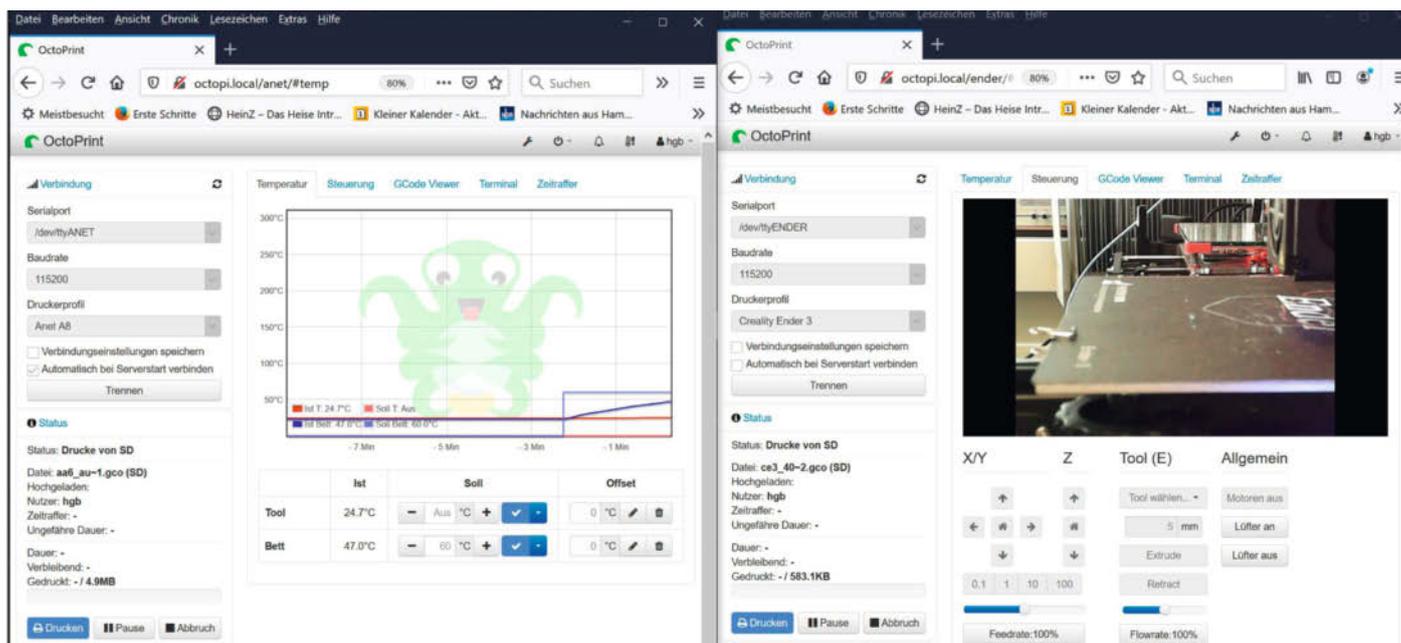
und die zweite in einem eigenen Fenster mit

```
http://octopi.local/ender/
```

Achten Sie darauf, dass am Ende der Adressen jeweils noch ein Schrägstrich stehen muss, sonst funktioniert es nicht! Wählen Sie dann in jedem Fenster die Schnittstelle und das dazugehörigen Druckerprofil aus, markieren Sie *Automatisch beim Serverstart verbinden* und klicken Sie auf *Verbinden*. Beim nächsten Aufruf der entsprechenden Druckeradresse wird die Verbindung zum entsprechenden Printer dann automatisch hergestellt.

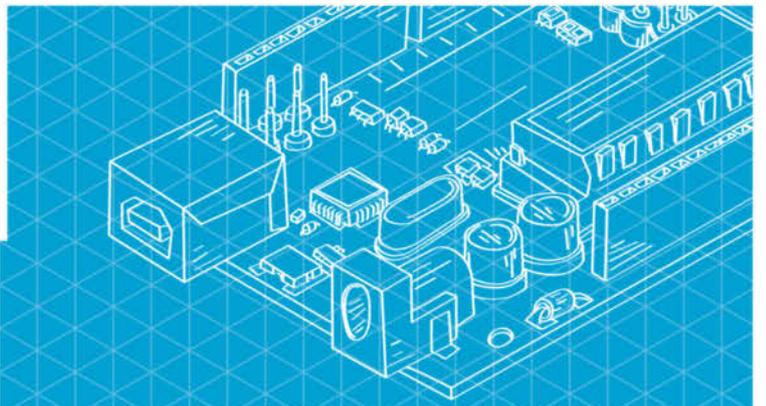
Bleibt noch die Kamerafrage. Aus Platzgründen haben wir diese Anleitung online gestellt (siehe Kurzinfo-Link).

So, der Weg war etwas mühsam, aber er lohnt sich. Denn jetzt können Sie alle OctoPrint-Vorteile nutzen, insbesondere die Erweiterungen. Und demnächst mit Video haben Sie wirklich alles unter Kontrolle. Viel Spaß mit dem Multi-Drucker-fähigen OctoPrint 16. —hgb



16 Das doppelte OctoPrint, allerdings im Moment mit nur einer Kamera.

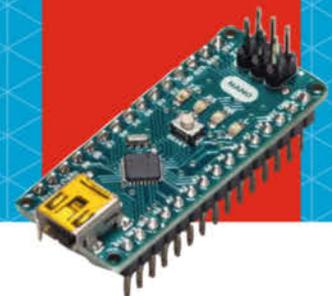
Make:



DAS KANNST DU AUCH!



GRATIS!



2x Make testen und 6 € sparen!

Ihre Vorteile:

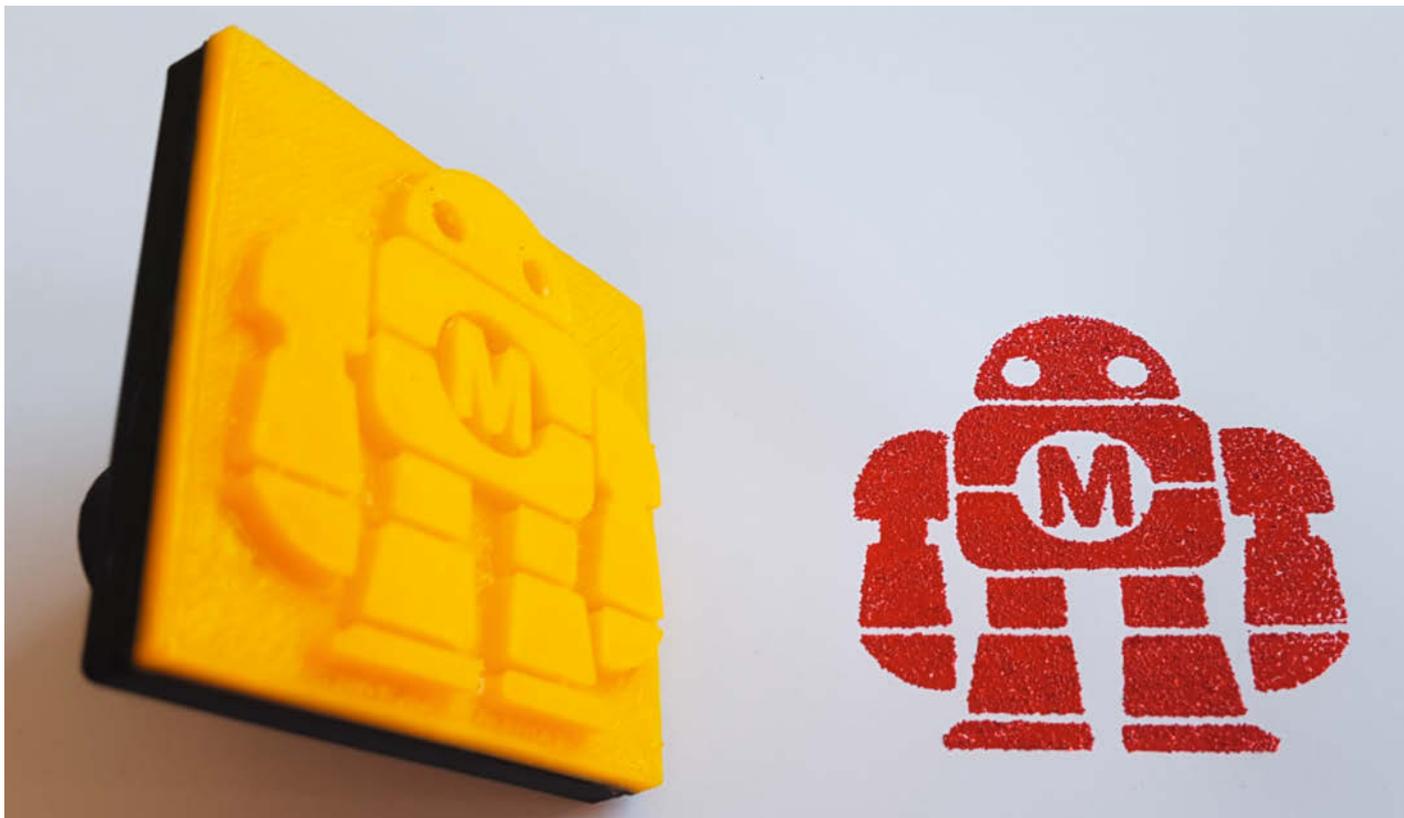
- ✓ **GRATIS dazu:** Arduino Nano
- ✓ **NEU:** Jetzt auch im Browser lesen!
- ✓ Zugriff auf Online-Artikel-Archiv*
- ✓ Zusätzlich digital über iOS oder Android lesen

Für nur 15,60 Euro statt 21,80 Euro.

* Für die Laufzeit des Angebotes.

Jetzt bestellen: make-magazin.de/miniabo

© Copyright by Maker Media GmbH.



Stempeln mit FreeCAD

Diesmal bauen wir mit der kostenlosen Konstruktionssoftware FreeCAD einen Stempel und fertigen ihn auf dem 3D-Drucker mit dem geeigneten elastischen Material. Auch der Stempel selbst kann 3D – wenn man ihn zum Prägen mit Hilfe von Embossing-Pulver benutzt.

von Matthias Mett

Bei FreeCAD handelt es sich um eine kostenlose Open-Source-Software zum Erstellen von dreidimensionalen Modellen. In der Make-Ausgabe 2/20 haben wir bereits gezeigt, wie man mit FreeCAD einen Fingerring mit eingepprägter und aufgesetzter Beschriftung konstruiert und beim Dienstleister im Metall-3D-Druck fertigen lässt. Die Zeichenwerkzeuge von FreeCAD kamen dabei nur kurz zur Sprache, deshalb wird diesmal mit deren Hilfe ein Stempel nach einer Bildvorlage konstruiert und ebenfalls 3D-gedruckt. Dabei entsteht die Stempelplatte aus flexiblem, gummiartigen Material, der Griff hingegen aus festem Kunststoff.

FreeCAD lässt sich über die Webseite freecadweb.org für die Betriebssysteme Microsoft Windows, macOS und Linux herunterladen. Nachdem Sie das Programm installiert haben, beginnen Sie über das *Dashboard* ein neues Dokument. Speichern Sie es zuerst unter *Datei/Speichern unter* mit dem Namen „Makey“

ab, da in diesem Beispiel die Konturen des Maker-Faire-Maskottchens, des Roboters Makey nachgezeichnet werden sollen.

Auf der linken Seite des Programms befindet sich eine Baumansicht, in der die Dokumentenstruktur dargestellt ist. Sobald man ein neues Objekt erstellt, findet es sich dort wieder. Klickt man es an, öffnet sich unterhalb der Baumansicht das Eigenschaftfenster, in dem sich die Parameter des Objektes anpassen lassen.

Startbild

Für unseren Stempel wollen wir nach einer Vorlage die Konturen von Makey zweidimensional nachzeichnen und daraus ein 3D-Objekt erstellen. Wechseln Sie dazu in der Symbolleiste den Arbeitsbereich vom voreingestellten mit der Bezeichnung *Start* zum Arbeitsbereich *Image* ①. In der jetzt erscheinenden *Image*-Symbolleiste betätigen Sie den Button

mit dem Tooltip *Eine planare Grafik im 3D Raum erzeugen* ② und wählen die Datei mit der Bildvorlage aus. Das Makey-Beispielbild finden Sie über den Link in der Kurzinfo.

Lassen Sie im nächsten Fenster die Orientierung der Bildebene auf *XY-Ebene* stehen und bestätigen Sie mit *OK*. Sollten Sie jetzt nichts sehen, beachten Sie den Kasten *Bild unsichtbar?* Andernfalls wählen Sie im Navigation Cube ③ in der rechten oberen Ecke des Zeichenbereichs *Top* aus, um in die Draufsicht zu gelangen. Klicken Sie nun auf das Bild ④, worauf es etwas seine Farbe verändert, und schauen sich auf der linken Seite die Bildeigenschaften an.

Unter *Image Plane* ist *XSize* mit 25mm und *YSize* mit 17mm angegeben ⑤. Der Stempel würde so viel zu klein. Daher können Sie die Skalierung des Bildes anpassen, indem Sie in der *Image*-Symbolleiste den Button *Skalieren einer Bildebene durch Festlegen des Abstandes zweier Punkte* ⑥ auswählen. Wählen Sie als

ersten Punkt die untere Kante des Fußes **7** in der Mitte mit der Maus aus und als zweiten Punkt die höchste Mitte des Kopfes **8**. Klicken Sie auf das Bild, um diese Bildebene auszuwählen, wie im Dialog gefordert. Geben Sie nun unter *Distanz* 50mm ein und bestätigen Sie mit *OK* **9**. Heben Sie die Markierung des Bildes mit einem Klick in den freien Bereich auf, worauf es in der ursprünglichen Farbe erscheint.

2D-Skizze

Um ein wenig die Arbeit zu erleichtern, zeichnen wir nur die rechte Hälfte des Logos und spiegeln diese auf die andere Seite. Wechseln Sie zum Zeichnen in der oberen Symbolleiste in den Arbeitsbereich *Sketcher* **10**. Suchen Sie in den neu erschienenen Symbolleisten den Button mit dem Tooltip *Neue Skizze erstellen* **11**. Im darauffolgenden Fenster bestätigen Sie die Voreinstellung der Skizzenorientierung auf der XY-Ebene mit *OK*. Auf der linken Seite in den *Aufgaben* haken Sie unter *Meldungen des Löfers* die beiden Punkte *Auto remove Redundants* und *Auto Update* an **12**. Unter dem Punkt *Bedienelemente bearbeiten* **13** nehmen Sie den Haken bei *Raster anzeigen* raus, falls das Kontrollkästchen nicht ohnehin schon leer ist.

Der Ursprung der Skizze sollte automatisch in der Mitte des Bildes liegen, die vertikale Achse sollte genau mittig durch das Bild verlaufen. Beginnen Sie nun mit der ersten Linie, der Rundung Kopfes. Wählen Sie aus der Symbolleiste den Kreisbogen mit der Bezeichnung *Endpunkte und Punkt auf Kreisbogen* aus **14** (falls bei Ihnen an der im Bild markierten Stelle ein anderes Kreiswerkzeug zu sehen ist, können Sie das gewünschte über einen Klick auf das kleine Dreieck daneben zum Vorschein bringen). Starten Sie den Bogen am obersten Punkt des Kopfes an der vertikalen Achse **15** (erster Klick) und ziehen Sie ihn herunter auf die untere rechte Ecke des Kopfes **16** (zweiter Klick). Den dritten Klick setzen Sie in der Mitte des Kreisbogens, nachdem der Bogenradius richtig sitzt **17**. Eine vertikale Hilfslinie am anderen Ende des Bogens hilft beim Setzen des Radius.

Zeichnen Sie nun mit dem Werkzeug *Linie in der Skizze erstellen* **18** die untere Linie des Kopfes wieder von der vertikalen Achse aus in Richtung des Kreisbogens. Der Endpunkt der vertikalen Hilfslinie des Radius **19** kann als Startpunkt dienen. Bevor Sie die Linie mit dem zweiten Klick beenden, bewegen Sie den Cursor über den Endpunkt des Kreisbogens **20**. Wenn beide Punkte gelb sind, verbindet FreeCAD beide Punkte, damit keine Lücken entstehen, die später das Extrahieren als 3D-Objekt verhindern könnten.

Sie können die beiden Endpunkte auch verbinden, indem Sie mit einem Klick der rechten Maustaste aus dem Zeichenmodus in den Auswahlmodus wechseln. Markieren Sie dann

Kurzinfo

- » Konstruieren der Stempelplatte nach einer Bildvorlage in FreeCAD
- » Konstruieren des Stempelgriffs für den 3D-Druck aus festem Material
- » Papier prägen mit dem selbstgedruckten Stempel und Embossing-Pulver

Checkliste



Zeitaufwand:

etwa 2 Stunden (ohne Druckzeit)



Kosten:

bis etwa 30 Euro (ohne Embossing-Material)



Computer:

Desktop oder Notebook mit Windows, macOS oder Linux



Entwerfen:

CAD, aber keine Vorkenntnisse nötig



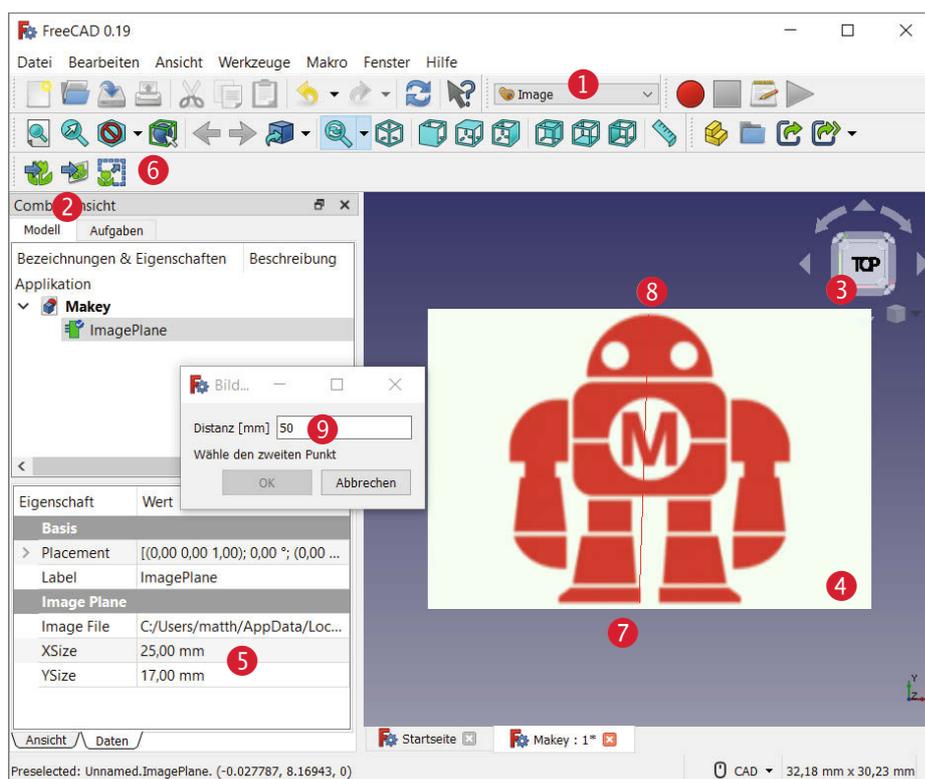
3D-Druck:

mit flexiblem und festem Material, auf dem eigenen Drucker oder beim Dienstleister

Mehr zum Thema

- » Matthias Mett, Ring frei für FreeCAD, Make 2/20, S. 114
- » Peter König, Gratis-CAD für Maker, Make 4/18, S. 90

Alles zum Artikel im Web unter make-magazin.de/xw3z



die beiden Endpunkte mit dem Lasso (gedrückt gehaltene linke Maustaste ziehen), aber ohne die Linien mit auszuwählen. Versuchen Sie, das Lasso größer oder in eine andere Richtung zu ziehen, falls sich die Punkte nicht markieren lassen. Verwenden Sie nun aus der Symbolleiste mit den Skizzen-Beschränkungen *Eine Koinzidenzbeschränkung für das gewählte Element setzen* 21. Diese verbindet

dann beide Endpunkte. Falls bei Ihnen die beiden Endpunkte automatisch aneinander eingerastet sind, fügt FreeCAD eine solche Beschränkung übrigens von sich aus hinzu.

Beschränkungen

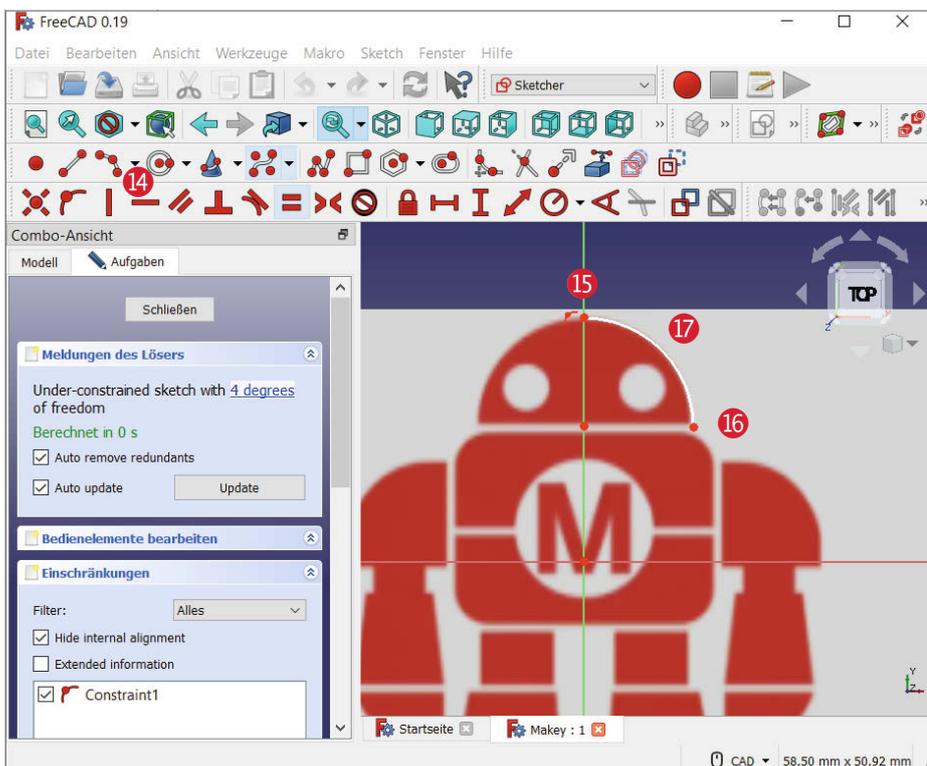
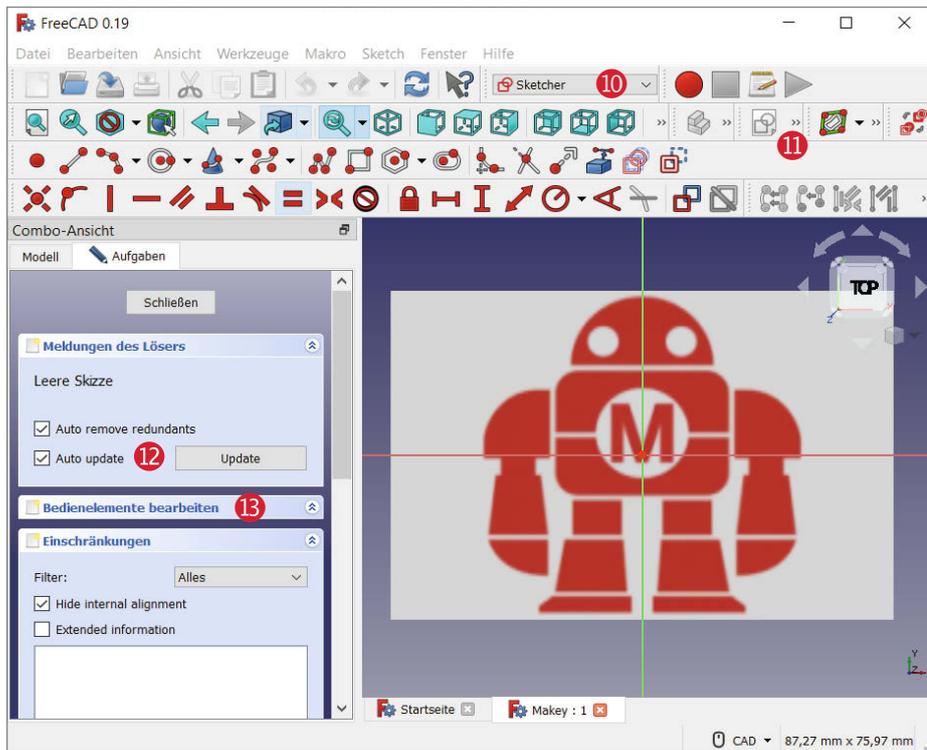
Sogenannte Beschränkungen oder Constraints verleihen der Skizze Präzision: Falls die untere

Linie etwas schief verläuft, lässt sie sich mit der Einschränkung *Eine horizontale Beschränkung für das gewählte Element setzen* 22 begradigen. Eventuell hat FreeCAD die Beschränkung automatisch hinzugefügt, das können Sie an der kurzen Linie oberhalb der gezeichneten Linie erkennen, oder in der Eigenschaftenbox nachsehen 23. Durch An- und Abhaken einer Beschränkung aus dieser Liste sehen Sie meist, wo diese in der Skizze sitzt.

Falls Sie an Ihren Linien und Kurven noch etwas ändern wollen, können Sie am oberen Punkt des Bogens mit gedrückter Maustaste den Radius anpassen (falls noch ein Werkzeug aktiv ist, wieder durch Rechtsklick in den Auswahlmodus wechseln). Das komplette Element verschieben Sie mit gedrückter Maustaste, nachdem Sie den Mauszeiger auf eine Linie davon geführt haben.

Abrundungen

Zeichnen Sie dann das Auge mit dem Button *Einen Kreis in der Skizze erstellen* 24. Den Durchmesser des Kreises können Sie durch Anfassen der äußeren Linie anpassen, die Position über den mittleren Punkt. Dann fahren Sie mit der oberen Linie des Körpers fort. Zeichnen Sie hier wieder von der vertikalen Hilfslinie aus eine Linie, diesmal aber über die obere Abrundung hinaus 25. Zeichnen Sie nun eine vertikale Linie von der unteren rechten Ecke nach oben, ebenfalls über die Rundung hinaus



GUI-Tricks

Falls FreeCAD die Symbolleisten alle in einer Reihe anzeigt, können Sie sie an den gepunkteten Anfassern entweder eine Ebene nach unten ziehen und damit den Platz für Symbole vergrößern oder die Leiste auf der linken und rechten Seite des Fensters andocken. In der Standard-Navigationseinstellung CAD können Sie mit dem Mausrad das Bild größer zoomen. Wenn sich das Bild in eine falsche Richtung verschiebt, holen Sie es mit der Steuerungstaste und der rechten Maustaste wieder in das Zentrum. Falls sich das Bild dabei verdreht, genügt ein Klick auf *Top* des Navigation Cubes. Verhält sich Ihr FreeCAD anders als gerade beschrieben, können Sie die *Navigationseinstellung* prüfen und gegebenenfalls ändern. Sie finden die Einstellung als Eintrag in dem Kontextmenü, das ein Rechtsklick auf den leeren blauen Zeichnungshintergrund öffnet – allerdings nicht, sofern Sie sich gerade in einer 2D-Skizze befinden.

26, sodass beide Linien sich kreuzen. Positionieren Sie die Linien mit Hilfe ihrer Endpunkte, damit diese richtig sitzen.

Suchen Sie in der Symbolleiste mit den Skizzengeometrien den Button *Erstellen einer Abrundung zwischen zwei Geraden oder an einen anliegenden Punkt* 27. Wenn Sie die beiden sich überkreuzenden Linien nacheinander markieren, erstellt der Befehl eine Abrundung. Deren Lage können Sie durch Verschieben des eingblendeten Mittelpunkts der Abrundung anpassen 28. Falls der Radius zu eng sein sollte, können Sie ihn durch Anpassen der Länge der oberen oder seitlichen Linie vergrößern. Zeichnen Sie nun noch die untere Linie des Körpers in Richtung des inneren Kreises und den inneren Viertelkreis 29.

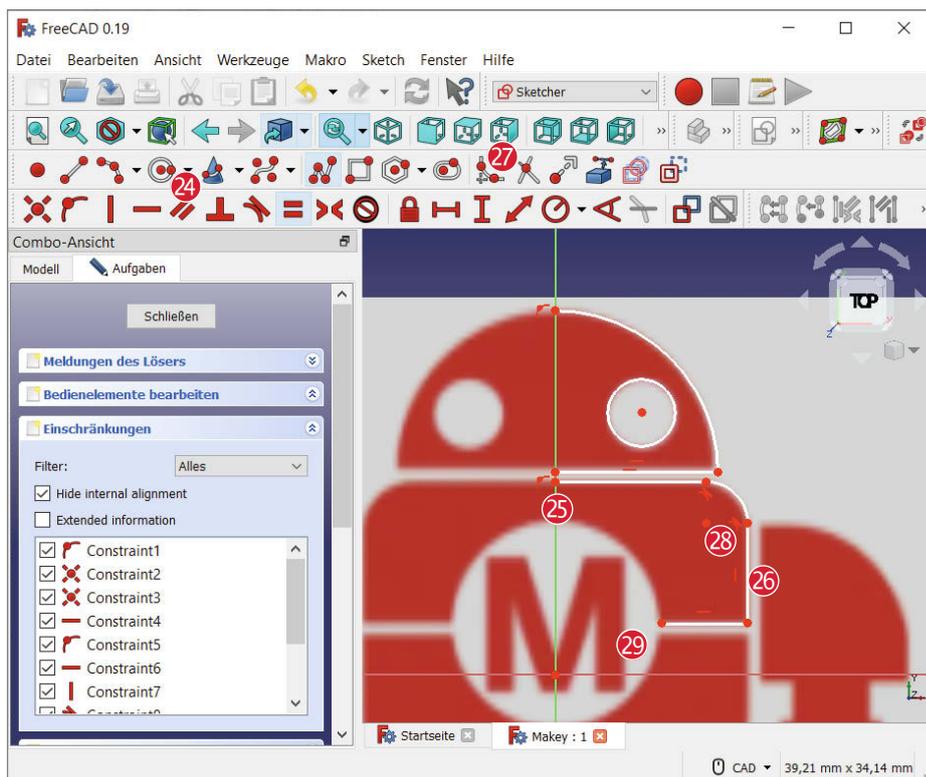
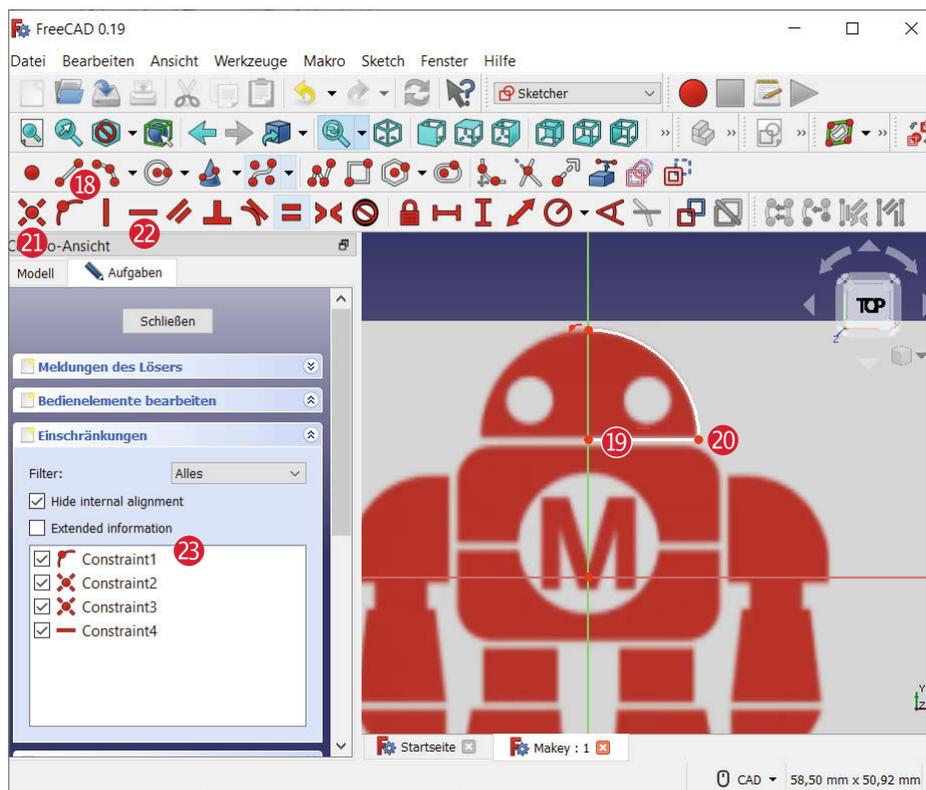
Spiegeln

Da der obere rechte Teil des Körpers gleich dem des unteren rechten Teils ist, spiegeln wir ihn. Markieren Sie dazu alle seine Teile durch Anklicken mit der Maus, bis alle grün eingefärbt sind. Markieren Sie dann mit gedrückter Steuerungstaste die horizontale Achse. Suchen Sie nun in der Symbolleiste der Skizzier-Werkzeuge den Button *Erstellt eine symmetrische Geometrie mit Bezug auf die zuletzt ausgewählte Linie oder Punkt* 30.

Der obere Teil des Körpers ist nun nach unten gespiegelt. Markieren Sie den unteren neuen Teil des Körpers komplett und positionieren Sie ihn anhand des Befehls *Verschieben* aus der gleichen Symbolleiste 31. Am Einfachsten geht es, wenn Sie mit dem Lasso den unteren Teil des Körpers so auswählen, dass der zuletzt markierte Punkt der Mittelpunkt des Viertelkreises um das M auf Makeys Brust ist. Denn nach dem Betätigen des Buttons erscheint eine Hilfslinie, die vom zuletzt ausgewählten Punkt ausgeht. Mit deren Ende markieren Sie den Punkt in der Bildvorlage, der damit korrespondieren soll – hier der Mittelpunkt des Kreises um das M – woraufhin sich der untere Teil des Körpers verschiebt.

Zeichnen Sie nun mit Hilfe der bisher benutzten Werkzeuge den rechten Teil des Körpers komplett fertig. Achten Sie dabei darauf, dass alle horizontalen und vertikalen Linien eine automatisch hinzugefügte Beschränkung (Constraint) auf diese Richtung haben (zu erkennen an den eingblendeten waagerechten und senkrechten kurzen Linien) und dass alle Punkte aneinander einrasten, damit die Kurven geschlossen sind.

Markieren Sie den fertigen rechten Teil des Körpers und anschließend mit gedrückter Steuerungstaste die vertikale Achse. Spiegeln Sie alles auf die linke Seite. Bei unserer Zeichnung spiegelte die Seite perfekt, ohne das wir etwas passend schieben mussten – die Figur sah prima aus. Verbinden Sie schließlich die noch offenen Punkte des Kopfes und des Kör-

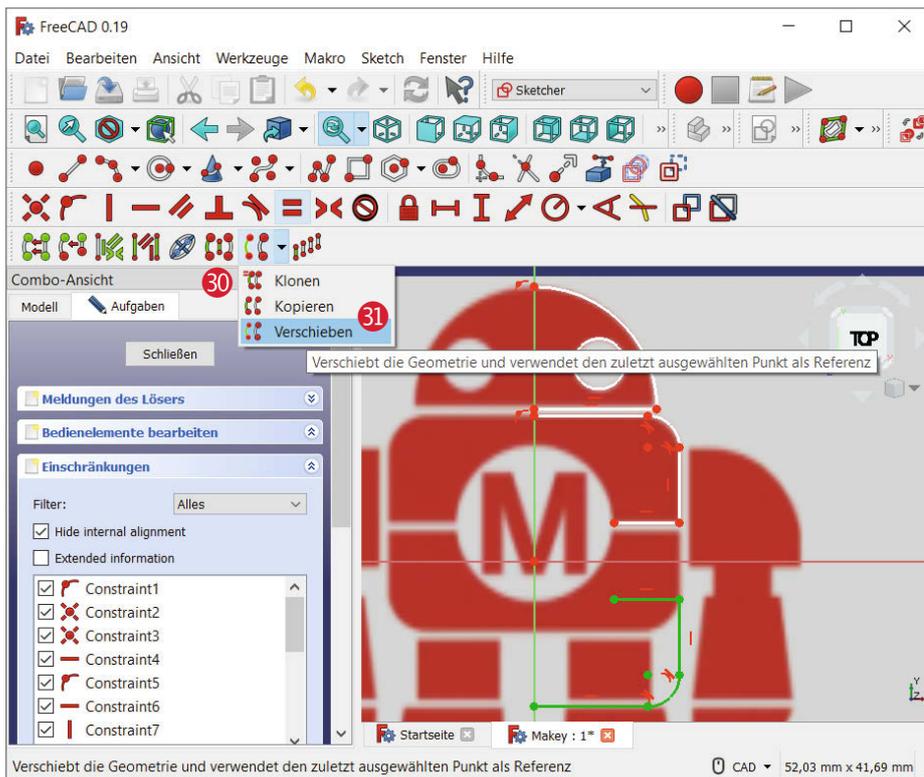


pers auf der vertikalen Achse wieder mit der Koinzidenzbeschränkung 21.

Auf Maß

Maßangaben haben wir bisher nicht verwendet, jedoch sollen die Abstände der einzelnen

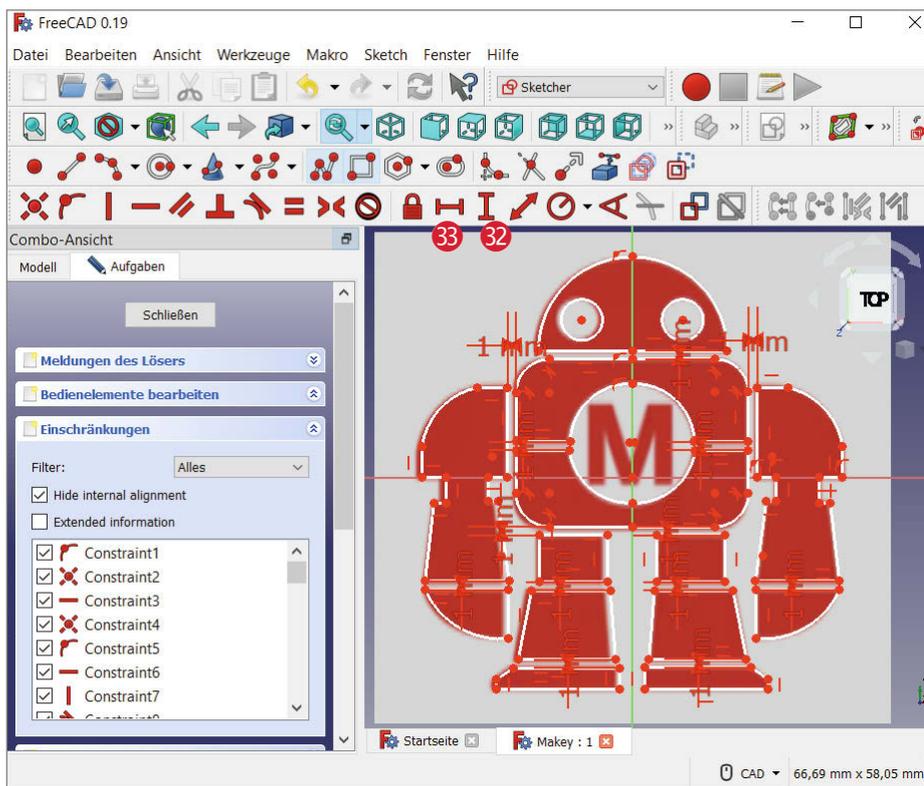
Teilformen, wo sie durch schmale Lücken getrennt sind, exakt gleich sein. Markieren Sie dazu zum Beispiel einen Punkt der unteren Linie des Kopfes und einen Punkt der oberen Linie des Körpers nacheinander bei gedrückter Steuerungstaste. Suchen Sie in der Symbolleiste der Skizzen-Beschränkungen den But-



Flächen füllen

Wechseln Sie nun in den Arbeitsbereich **Part** ³⁴ und rufen Sie in der Menüleiste unter **Formteil** ³⁵ den **Form-Generator** auf. Wählen Sie im Aufgabenbereich **Fläche aus Kanten** ³⁶ aus. Markieren Sie zuerst ein Auge und drücken Sie auf **Erstellen** ³⁷. FreeCAD erstellt eine neue Fläche mit dem Namen **Face**. Fahren Sie mit dem nächsten Auge fort und erstellen Sie wieder eine neue Fläche. Die einzelnen Linien des Kopfes und der anderen Körperteile können Sie bei gedrückter Steuerungstaste mit der Maus nacheinander markieren und jeweils eine Fläche erzeugen.

Wenn Sie alle Flächen erzeugt haben, schließen Sie den Aufgabenbereich mit dem entsprechenden Button im Aufgabenbereich links. Markieren Sie nun die Fläche des Kopfes mit dem Namen **Face002** (also dem Kopf) in der Baumansicht und mit gedrückter Steuerungstaste die Fläche des linken Auges mit dem Namen **Face**. Drücken Sie dann den Button **Schneidet zwei Formen** ³⁸, um das Auge aus dem Kopf herauszustanzeln (die Warnmeldung können Sie ignorieren). Markieren Sie für das zweite Auge das Objekt **Cut**, das sich nun unten in der Baumansicht befindet. Markieren Sie anschließend mit gedrückter Steuerungstaste **Face001** und drücken Sie wieder den Button **Schneidet zwei Formen**.



3D-Extrusion

Markieren Sie die erste Fläche in der Liste mit dem Namen **Face003**. Erzeugen Sie ein 3D-Objekt, indem Sie mit dem Button **Extrudieren einer ausgewählten Skizze** ³⁹ die Achse **Z** auswählen und eine Länge von 6.00 für 6mm angeben. Stellen Sie außerdem die Voreinstellung unter **Länge** im Feld **Entlang** von 10mm auf 0mm zurück. Mit dem Button **Anwenden** erzeugen Sie das 3D-Objekt. Wählen Sie unten in der Auswahlbox des Aufgabenbereichs die nächste Fläche mit dem Namen **Face004** aus und drücken Sie wieder auf **Anwenden**.

Fahren Sie so mit allen Flächen fort, bis jede Fläche zu einem 3D-Objekt umgewandelt ist. Extrahieren Sie jedoch **Face001**, **Face002**, **Face** und **Cut** nicht, sondern nur **Cut001** ⁴⁰, also den fertigen Teil des Kopfes. Das Ergebnis können Sie überprüfen, wenn Sie die Ansicht über den Navigation Cube auf eine Ecke kippen ⁴¹. Aus den Face-Objekten hat FreeCAD jetzt Extrude-Objekte erzeugt. Wenn alles passt, klicken Sie im Aufgabenbereich auf **Schließen**, um die Baumansicht wieder zum Vorschein zu bringen.

Beschriftung

Jetzt fehlt noch der Buchstabe **M** in der Mitte von **Makey**. Blenden Sie zuerst in der Baum-

ton **Den vertikalen Abstand zwischen zwei Punkten oder Streckenenden festlegen** ³². In der sich öffnenden Eingabebox geben Sie als Länge 1,00mm an. Fahren Sie so mit allen Abständen der einzelnen Körperteile fort – mit den vertikalen wie mit den horizontalen, für die es ein eigenes Werkzeug gibt ³³.

Verlassen Sie nun mit dem Button **Bearbeitung der Skizze beenden** den Zeichenbereich. Markieren Sie das Bild in der Baumansicht mit dem Namen **ImagePlane** und verstecken Sie es über den Befehl **Ein/Ausblenden** aus dem Kontextmenü (rechts anklicken). Jetzt sind nur die gezeichneten Linien sichtbar.

ansicht das Bild mit dem Namen *ImagePlane* mit der rechten Maustaste und über das Kontextmenü wieder ein ⁴². Wechseln Sie über den Navigation Cube mittels *Top* wieder auf die Draufsicht und wechseln Sie in den Arbeitsbereich *Draft* ⁴³. In der Symbolleiste mit den *Draft creation tools* wählen Sie den Button mit dem gelben S darauf (Tooltip: *Create a shape from a text string by choosing a specific font and a placement*) ⁴⁴.

Wählen Sie mit der Maus den linken unteren Eckpunkt des Buchstabens aus. Anschließend geben Sie im Aufgabenbereich im Feld *Zeichenkette* den Großbuchstaben „M“ ein. Im

Feld *Schriftartendatei* wählen Sie unter *Windows* im Pfad *C:\Windows\Fonts* die Schriftart *arialbd.ttf* aus drücken Sie auf *OK*, um den Buchstaben zu erzeugen.

Mithilfe des *Verschiebe-Buttons* mit dem Namen *Bewegt das ausgewählte Objekt zwi-*

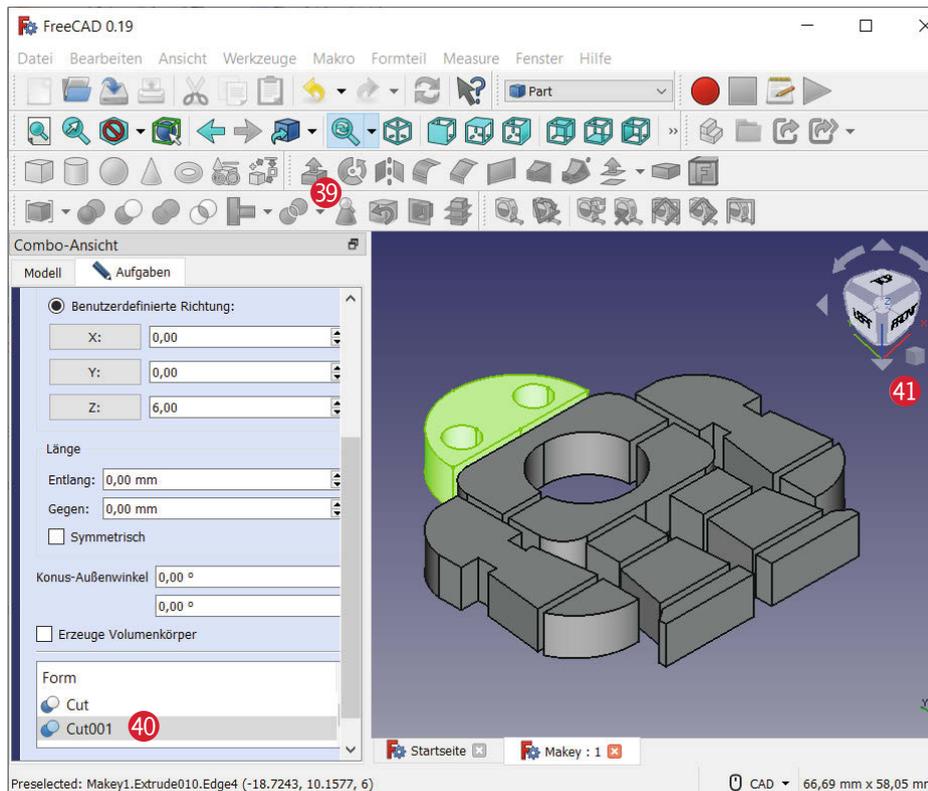
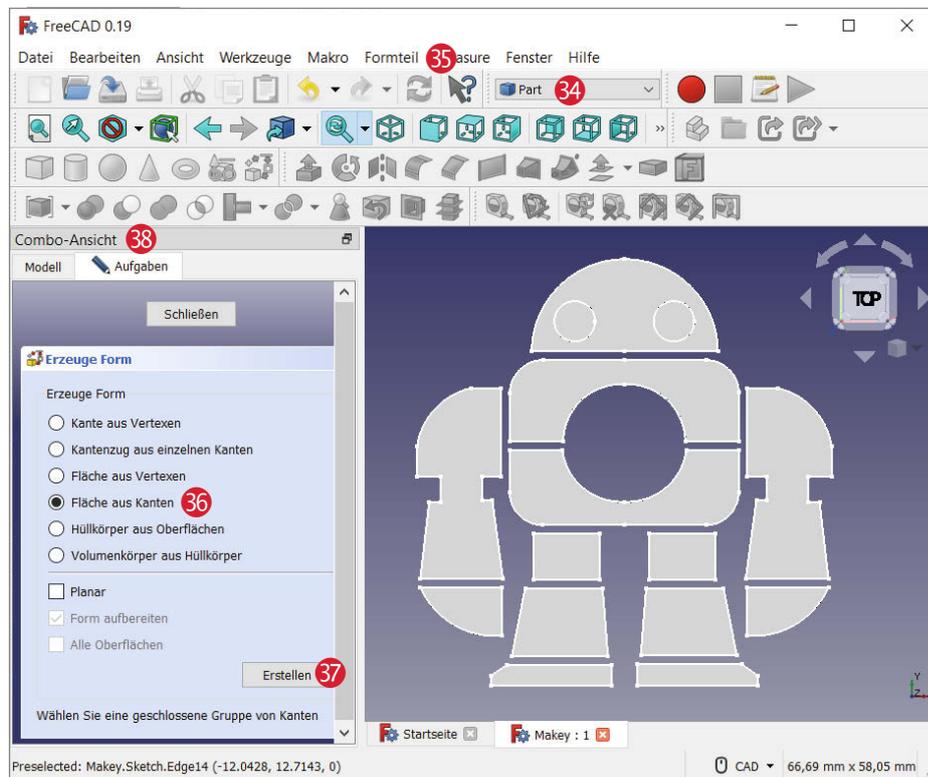
schen 2 Punkten ⁴⁵ können Sie den Buchstaben passend anhand der Bildvorlage verschieben. Falls Sie den Buchstaben nicht sehen, liegt er möglicherweise hinter der Bildebene, Sie können das prüfen, indem Sie die Bildebene in der *Baumansicht* ausblenden. In diesem

Bild unsichtbar?

Bei manchen Grafikkarten kann es in der FreeCAD-Version 0.18 zu Problemen kommen, wenn man eine Bild-datei importiert. FreeCAD zeigt dann das Bild nach dem Import nicht an und es erscheint in der Statusleiste ganz unten im Programmfenster die Fehlermeldung in Rot: „Unhandled Base::Exception caught in GUIApplication::notify.The error message is: Access violation“.

Falls Sie von dem Problem betroffen sind, installieren Sie die neueste Entwicklerversion 0.19, die Sie im Downloadbereich der FreeCAD-Webseite finden. Navigieren Sie dazu auf der Downloadseite weiter herunter zu *Development versions* und klicken Sie auf *FreeCAD releases page*. Weiter unten in den *Assets* finden sich die Versionen für das jeweilige Betriebssystem.

Für Windows laden Sie *FreeCAD_0.19.XXXXX_x64_LP_12.1.2_PY3QT5-WinVS2015.7z* herunter. Zum Entpacken installieren Sie vorher das kostenlose Programm 7-Zip (Download siehe Link in der Kurzinfor). Entpacken Sie das Archiv und benennen Sie den erscheinenden Ordner in *FreeCAD 0.19* um. Verschieben Sie den Ordner in das *Programme-Verzeichnis*, in dem auch das andere FreeCAD-Verzeichnis liegt. Navigieren Sie mit dem *Windows Explorer* in das Verzeichnis und suchen Sie im Unterordner *bin* die Datei *FreeCAD.exe*. Erstellen Sie per rechter Maustaste eine Verknüpfung auf dem Desktop, die Sie ebenfalls *FreeCAD 0.19* nennen. Auf diese Weise haben Sie parallel die stabile Version 0.18 und die Vorabversion 0.19 startbereit.



Fall wechseln Sie bei ausgewählter Bildebene in die Eigenschaften und stellen Sie dort unter *Placement/Position/Z* für die Bildebene den Wert -1 ein, um das Bild um einen Millimeter nach unten zu schieben **46**.

Sitzt der Buchstabe von oben gesehen richtig, blenden Sie das Bild in der Baumansicht

wieder aus und überprüfen Sie die Z-Position des Buchstabens, indem Sie die Zeichnung mit dem Navigation Cube auf eine Ecke kippen. Falls der Buchstabe im oberen Bereich der 3D-Objekte sitzen sollte, überprüfen Sie wieder in den Eigenschaften unter *Placement/Position/Z*, ob dort der Wert 6mm an-

gegeben ist. Falls ja, setzen Sie diesen auf 0mm zurück. Wechseln Sie nun wieder in den *Part*-Arbeitsbereich **47** und extrahieren Sie den Buchstaben wie gehabt, wobei das Extrude-Objekt *Extrude013* **48** entsteht.

Abschluss

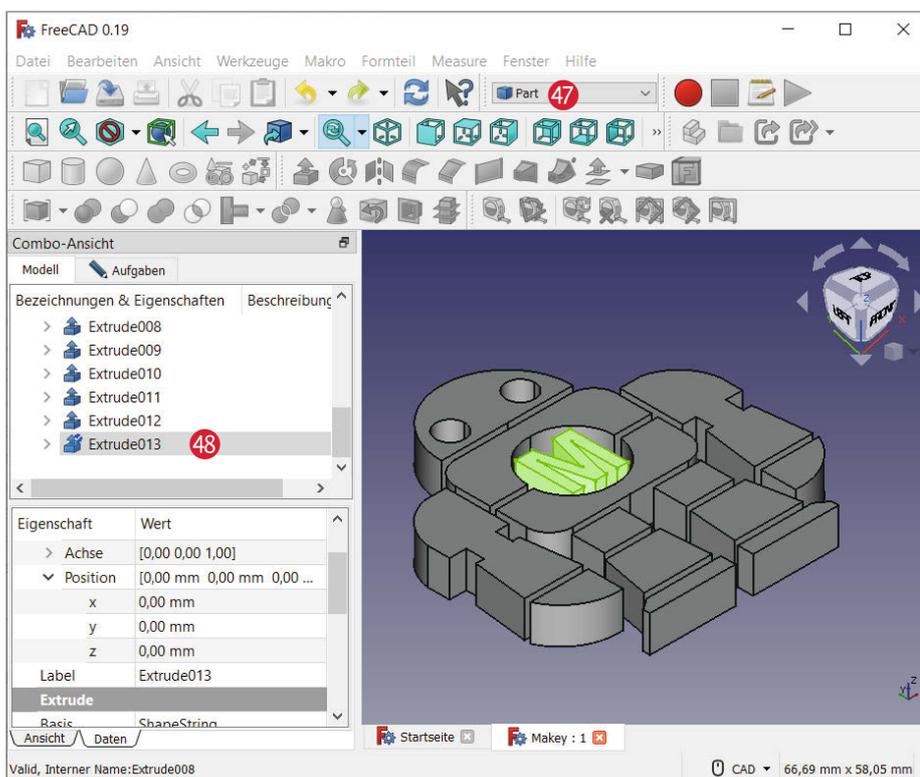
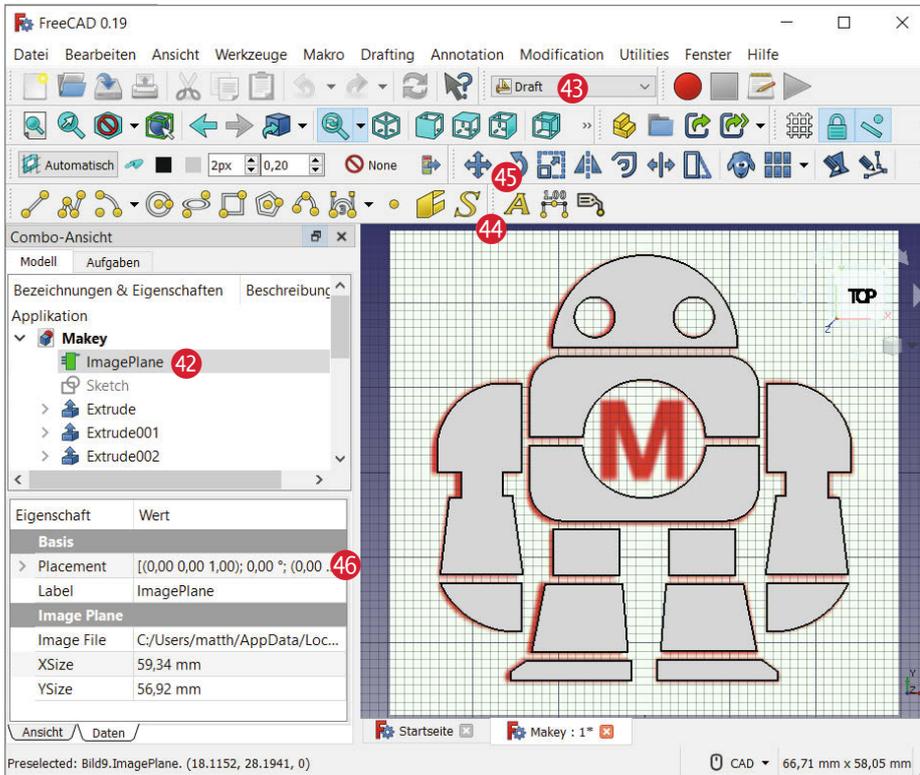
Blenden Sie in der Baumansicht den Buchstaben mit der Bezeichnung *Extrude013* mit der rechten Maustaste aus, um freien Zugriff auf den Schnittpunkt der horizontalen und vertikalen Achse zu haben. Wechseln Sie nun wieder zurück in den *Sketch* in der Baumansicht. Zeichnen Sie für die Stützplatte ein Rechteck mit dem Werkzeug *Rechteck in der Skizze erstellen* um Makey herum. Position und Maße spielen keine Rolle, denn im nächsten Schritt schränken Sie das Rechteck ausgehend vom Schnittpunkt der Achsen zum rechten oberen Punkt horizontal ein und legen für den Abstand den Wert 27mm fest. Fahren Sie mit der Einschränkung auf der linken Seite fort, sowie vertikal oben und unten, wozu Sie wieder die Werkzeuge aus den Schritten **32** und **33** benutzen.

Blenden Sie in der Baumansicht den Buchstaben mit der Bezeichnung *Extrude013* wieder ein. Extrahieren Sie wieder im *Part*-Arbeitsbereich das Rechteck in der Z-Achse um 3mm. Die Körperteile vom Makey ragen jetzt 3mm aus der Stützplatte heraus **49**.

Erhöhen Sie vor dem Export zum 3D-Druck die Mesh-Dichte des zu exportierenden Objektes, um sichtbare Kanten in den Rundungen im Export zu vermeiden. Wechseln Sie dazu im Menü unter *Bearbeiten/Einstellungen...* im Bereich *Part Design* in den Kartenreiter *Form-Ansicht*. Ändern Sie den Wert unter *Setzt die maximale Abweichung abhängig von der Modell-Bounding-Box* von 0,5000 % zum Maximalwert von 0,0100 %. Das erhöht die Mesh-Dichte soweit, dass keine Ecken und Kanten mehr auftreten. Markieren Sie nun wieder in der Baumansicht alle Objekte mit der Steuerungstaste, die mit der Bezeichnung *Extrude* beginnen. **50** Wählen Sie im Menü *Datei/Exportieren...* das Format *STL-Mesh (*.stl,*.ast)*. Geben Sie als Dateinamen „Makey“ an und starten Sie den Export. Damit ist der Teil des Stempels fertig, der aus flexiblem Material gedruckt wird.

Stempelgriff

Für den Stempelgriff legen Sie ein neues Dokument an und nennen es „Stempel“. Erstellen Sie eine neue Skizze und zeichnen Sie ein Rechteck, das Sie wieder auf allen Seiten vom Mittelpunkt der Achsen mit 27mm Distanz einschränken. Wechseln Sie in den Arbeitsbereich *Part* **51** und extrahieren Sie das Rechteck entlang der Z-Achse um 5mm. Fügen Sie den eigentlichen Griff über den Button *Einen Zylinder*

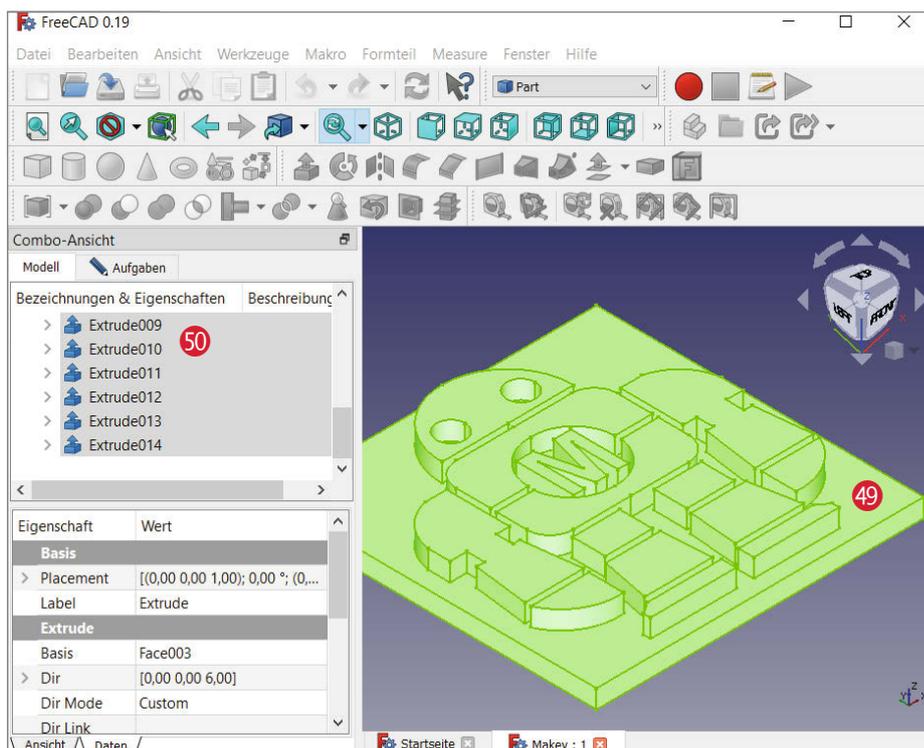


der erstellen ⁵² hinzu. Passen Sie in den Eigenschaften den *Radius* auf 10mm sowie *Height* auf 35mm an. Die oberen Kanten können Sie grifffreundlicher gestalten, indem Sie den Button *Die ausgewählten Kanten einer Form abrunden* ⁵³ verwenden. Markieren Sie dann die beiden *Fillet*-Objekte in der Baumansicht ⁵⁴ mit gedrückter Steuerungstaste und exportieren Sie sie als STL-Datei.

3D-Druck

Wir haben beide Dateien auf einem Ultimaker 2+ gedruckt: die Datei *Makey.stl* für die Stempelplatte aus dem flexiblen Filament Ninjaflex (mit dem Bild nach unten auf der Glasplatte und Stützen) und die Datei *Stempel.stl* aus gewöhnlichem PLA. Die beiden Teile hält handelsüblicher Kleber zusammen, der für Gummi und Kunststoff geeignet ist; auch Doppelklebeband funktioniert für diesen Zweck.

Alternativ könnten Sie die Ausdrücke auch bei einem Druckdienstleister wie *i.materialize* bestellen ⁵⁵. Bei diesem Anbieter kostet die Datei *Makey.stl* im Lasersinterverfahren aus flexiblem Material etwa 12 Euro. Die Datei



ONLINE

Microsoft Teams in der Praxis

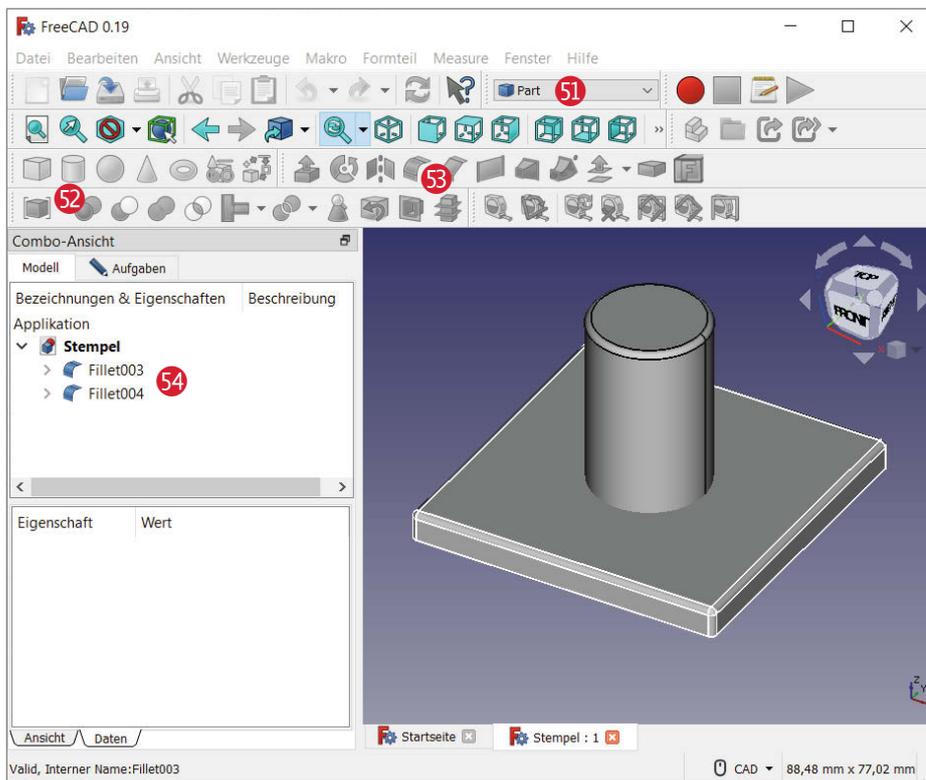
Grundlagen, Funktionen, Praxistipps für den Unternehmenseinsatz

17. September 2020, 9 – 13 Uhr

Microsoft Teams ist das am schnellsten wachsende Tool für kollaboratives Arbeiten. In diesem Webinar lernen Sie, was Teams alles bietet, wie Sie Ihre Prozesse damit effektiv organisieren und wie Sie Ihre Mitarbeiter dabei richtig einbinden. Lassen Sie sich von einem zertifizierten Experten beraten und bekommen Sie Antworten auf Ihre Fragen – ganz entspannt vom eigenen Schreibtisch aus.

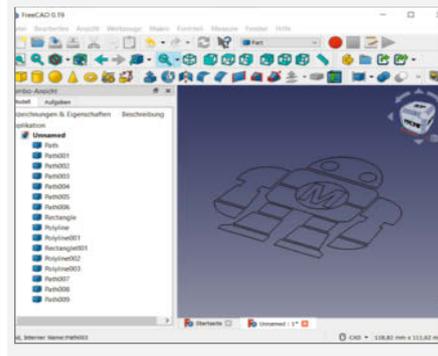
Sichern Sie sich 1 von 50 Spartickets für nur 129,00 Euro!

www.heise-onlinekonferenzen.de/microsoft-teams



Vorhandene Vektorgrafik

Falls Sie Ihr Wunschmotiv für den Stempel schon als fertige Vektorgrafikdatei im SVG-Format vorliegen haben, können diese in FreeCAD importieren und sparen sich das Abzeichnen von Hand. In FreeCAD importieren Sie die Vorlage über *Datei/Importieren.../SVG as geometry (*.svg)*. Beim Import erstellt FreeCAD eine Reihe von Objekten, die Sie im *Draft*-Arbeitsbereich weiterbearbeiten und dann wie beschrieben mit *Extrudieren einer ausgewählten Skizze* in 3D-Objekte umwandeln können. Andere Vektorgrafik-Formate als SVG importiert FreeCAD leider nicht so komfortabel, aber es ist einen Versuch wert, die Datei im kostenlosen Vektorgrafik-Programm *Inkscape* zu öffnen und sie als SVG-Datei abzuspeichern.



☰
i.materialize 55

	<p>Makey.stl 54.00 x 54.00 x 6.00 mm</p> <p> Gummiartig (MJF) Naturgrau</p>	<input type="button" value="-"/> <input type="text" value="1"/> <input type="button" value="+"/>	11.70 €	11.70 € ×
	<p>Stempel.stl 54.00 x 54.00 x 35.00 mm</p> <p> Polyamid (MJF) Natural</p>	<input type="button" value="-"/> <input type="text" value="1"/> <input type="button" value="+"/>	13.72 €	13.72 € ×

EINZELNE DATEI HOCHLADEN

MEHRERE DATEIEN HOCHLADEN

+ FÜGEN SIE AKTUELLE MODELLE HINZU

Voraussichtliches
Mai 12
Zwischensumme
25.42 €

Stempel.stl kommt je nach Material zwischen etwa 13 und etwa 17 Euro. Zu beachten ist bei Dienstleistern ein eventueller Mindestbestellwert, außerdem können Mehrwertsteuer und Versandkosten noch dazukommen. Wie sich die selbst gedruckten und die von i.materialize gefertigten Stempel in der Praxis machen, sehen Sie in einer Online-Bilderstrecke (siehe Link in der Kurzinfo).

Embossing-Test

Zum Ausprobieren der Stempel haben wir Embossing-Pulver verwendet, das nach Erwärmung eine geprägte Oberfläche ergibt (siehe Aufmacherbild auf Seite 116). Bei diesem Verfahren stempelt man zuerst mit einer durchsichtigen und langsam trocknenden Stempelfarbe die Kontur aufs erste Blatt Pa-

pier. Legen Sie zum Wiederverwenden des restlichen Pulvers ein zweites Stück Papier unter, das in der Mitte angeknickt ist.

Anschließend schütten Sie ein wenig Embossing-Pulver auf die noch nasse Stempelfarbe auf das erste Blatt. Nehmen Sie dann das gestempelte Stück Papier vorsichtig hoch und lassen Sie das überschüssige Pulver auf das darunterliegende Stück Papier rieseln. Wenn Sie beide Hälften des Papiers hochnehmen, sammelt sich an dem Knick das Pulver und Sie können es in die Dose zurückschütten.

Auf dem ersten Blatt Papier klebt nun das Pulver auf der durchsichtigen Stempelfarbe. Sie brachen das Pulver nicht auf der Stempelfarbe trocknen zu lassen, sondern können direkt weiterarbeiten. Nehmen Sie nun einen Embossing-Fön und erhitzen Sie damit das Pulver. Durch die Hitze verläuft es und ergibt eine strukturierte Oberfläche. Wir haben zwei verschiedene Embossing-Pulver ausprobiert, einmal *Card Deco Essentials Embossingpulver Glitzer Rot* und *Ranger Embossingpulver Red Tinsel*. Bilder von den Ergebnissen gibt es ebenfalls in unserer Online-Bilderstrecke. —pek

DIGITAL

— 2020 —

ITJOB SUMMIT

by Heise

SAVE THE DATE

06.10.2020 • 23.10.2020

FINDE DEINEN
TRAUMJOB
ONLINE!

Attraktive Arbeitgeber • Lebenslaufcheck • Spannende Vorträge

Unser Partner:  Jobware

JETZT KOSTENFREI REGISTRIEREN

WWW.IT-JOB-SUMMIT.DE

© Copyright by Maker Media GmbH

Ausprobiert
— von Make: —

Nomad 883 Pro CNC-Fräse



Das Gerät richtet sich preislich zwar eher an kleinere Firmen, Makerspaces und FabLabs, ist mit einer Grundfläche von 20cm x 20cm aber sehr kompakt und dank eines Aluminiumrahmens sehr steif – eine grundlegende Voraussetzung für eine präzise Materialbearbeitung. Eine Plexiglas-Klappe, die den Arbeitsraum komplett verschließt, reduziert die Belastung durch Staub und Lärm am Arbeitsplatz auf ein Minimum. Der Spindeltrieb ist eine Eigenkonstruktion aus einem BLDC-Motor und einer Riemenübersetzung.

In unserem Test zeigte sich, dass selbst die Bearbeitung von Stahl mit entsprechenden Einstellungen möglich ist, feinste Details werden präzise abgebildet. Dank der Auflösung von weniger als 0,04mm und einer Wiederholgenauigkeit von etwa 1/10mm ist die Maschine auch gut für die Fertigung von Platinen durch Isolationsfräsen geeignet, auch wenn die Spindeldrehzahl mit 10.000 Umdrehungen pro Minute eher etwas niedrig ist.

Wer keine zentnerschweren Stahlblöcke bearbeiten will und auf der Suche nach einer hochwertigen Fräse für feine Arbeiten an verschiedensten Materialien ist, findet mit der Nomad 883 Pro eine besonders hochwertige Lösung. Einen ausführlichen Test mit mehr Bildern gibt es online. —Roman Radtke/cm

► make-magazin.de/x1vt

Hersteller	Carbide 3D
URL	carbide3d.com/nomad
Preis	ca. 2800 €

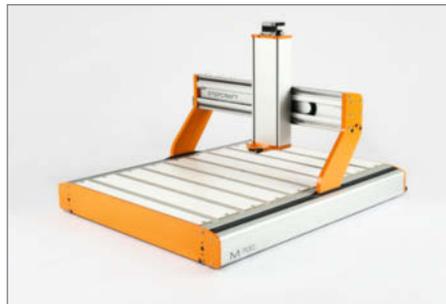
Stepcraft M-System

CNC-Fräse mit Erweiterungen

Bei der neuen CNC-Fräsen-Serie des deutschen Herstellers Stepcraft lässt sich der Maschinentisch vollständig herausnehmen, sodass – in Kombination mit optionalen Vakuumschienen – nicht nur normales, aufgespanntes Plattenmaterial bearbeitet werden kann, sondern auch Tische und Böden, auf denen die Maschine selbst steht. Laut Hersteller soll es sogar möglich sein, die Fräse damit an die Wand zu heften, um diese dann zu bearbeiten.

Für Präzision in der Mechanik sollen Kugelumlaufrollführungen von *Bosch Rexroth* aus gehärtetem Stahl bei allen Achsen sorgen, dazu *dryspin*-Gewindespindeln von *igus*. Optional und gegen Aufpreis gibt es noch hochwertigere Spindeln (*HIWIN*). Wie zuvor die D-Serie des Herstellers gibt es auch die M-Modelle in drei Größen: Der Arbeitsbereich erstreckt sich dabei von 348mm x 543mm x 194mm bei der M.500 bis zu 679mm x 1044mm x 194mm bei der M.1000. Alle Modelle sind entweder als Fertigerät oder als Bausatz erhältlich.

In die Werkzeugaufnahme mit 43mm Durchmesser passen nicht nur die Fräsmotoren des Herstellers, sondern auch die anderer Anbieter. Zudem soll alles Zubehör der D-Serie von Stepcraft mit der M-Serie kompatibel sein, sodass sich auch diese Maschinen nicht



nur zum Fräsen, sondern auch zum 3D-Drucken (siehe Make 2/18, S. 130), Schneiden, Lasern und Gravieren nutzen lassen. Für die D-Serie ist seit Mai neu ein Vakuum-Tisch im Angebot, mit dem sich etwa Platinen zum Isolationsfräsen deutlich einfacher fixieren lassen als mit herkömmlichen Spannvorrichtungen.

Einen ausführlichen Test und Erfahrungsbericht zur M-Serie lesen Sie in der nächsten Make-Ausgabe. —pek

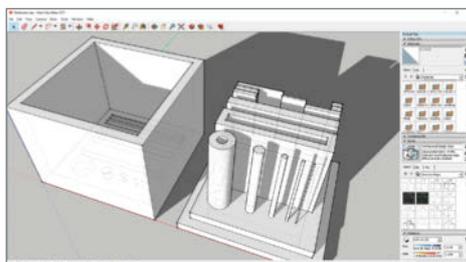
Hersteller	Stepcraft
URL	shop.stepcraft-systems.com
Preis (Bausatz/ Fertig)	M.500: 1899 €/2298 € M.700: 2399 €/2848 € M.1000: 2999 €/3498 €

SketchUp

3D-Software nur noch im Abo

Die Firma Trimble, mittlerweile Besitzerin der bei vielen beliebten 3D-Konstruktionssoftware SketchUp, schwenkt zum 4. November komplett auf ein Abo-Modell um – ab dann zahlt man pro Jahr entweder 119 US-Dollar für die reine Online-Version *SketchUp Shop* oder 299 US-Dollar für *SketchUp Pro* oder 1199 US-Dollar für die *Studio*-Version. Zum Vergleich: Die klassische Vollversion von SketchUp kostet 695 US-Dollar und ist nur noch bis zu diesem Stichtag im Angebot. Wer das nutzt, dem legt Trimble ans Herz, die heruntergeladene Installationsdatei gut zu bunkern – was man als Hinweis darauf verstehen kann, dass es ab diesem Zeitpunkt keine Downloadmöglichkeit mehr jenseits eines Abos geben wird.

Das betrifft wahrscheinlich auch Nutzer der kostenlosen Version von SketchUp. Zwar wird *SketchUp Free* weiterhin angeboten, dabei handelt es sich aber (wie *Shop*) um ein reines Web-Angebot. Wenn man weiß, wo man suchen muss (siehe Link) kann man sich zwar immer noch die letzte kostenlose



SketchUp-Make-Version von 2017 für die Installation unter Windows und macOS herunterladen, aber zu befürchten ist, dass auch diese Download-Seite zum 4. November verschwindet. —pek

► make-magazin.de/x1vt

Hersteller	Trimble
URL	www.sketchup.com/de
Preis	gratis, Abo ab 119 US-\$/Jahr, Pro-Version 695 US-€

Audio Injector Zero

Soundkarte für Raspberry Zero

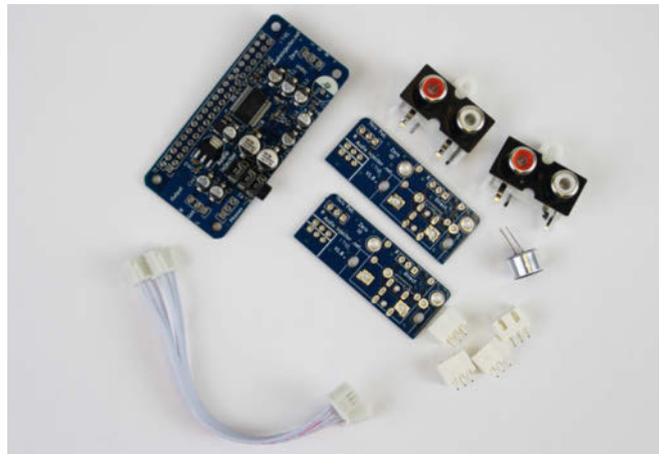
Soundkarten mit Ein- und Ausgängen gab es für den Zero bislang nur als USB-Versionen, die mittels OTG-Adapter an die einzige USB-Buchse des RasPi-Zwerges angeschlossen werden mussten. Dadurch ergab sich das Problem, die üblicherweise über die Buchse erfolgte Stromversorgung anderweitig lösen zu müssen. Aufsteckkarten für den GPIO-Port hingegen besaßen entweder nur Ausgänge oder allenfalls eingebaute Mikrofone als Eingänge. So war eine Direktaufnahme von anderen Tonquellen nicht möglich.

Der Audio Injector Zero hingegen ist eine vollwertige Soundkarte mit Stereo-Ein- und Ausgängen zum Aufstecken oder Einlöten (Steckerleiste für den GPIO-Port wird nicht mitgeliefert). Die technischen Daten (96kHz Samplerate, 32 Bit Auflösung) klingen anspruchsvoll und sollen hochwertige Tonaufnahmen nicht nur mit einem Zero, sondern auch mit den anderen Raspberry-Versionen ermögli-

chen. Einen Mono-Mikrofoneingang besitzt die Karte aber auch, ebenso einen Kopfhöreranschluss in Form einer 3,5mm-Klinkenbuchse.

Die Karte in der Größe eines Zero wird mit zwei Breakout-Boards, darin einlötbaren Stereo-Cinchbuchsen, einer Electret-Mikrofonkapsel sowie den benötigten Verbindungskabeln- und Steckverbindern geliefert. Bei Bedarf kann auf die Breakout-Boards auch noch jeweils ein Stereo-Potenzio- meter (nicht enthalten) eingebaut werden, um die Lautstärken analog einstellen zu können.

—hgb



Hersteller	Audioinjector
URL	www.audioinjector.net/rpi-zero
Preis	21 US-\$

DMSO2D72

Handheld-Oszilloskop

Drei Geräte im Formfaktor eines Digital-Multimeters: Das praktische Gerät von Joy-IT misst nicht nur Spannungen, Ströme und Widerstände, es ist ebenso ein vollwertiges Zweikanal-Oszilloskop und Funktionsgenerator.

Gerade bei Handheld-Oszilloskopen vermisst man ergonomische Drehknöpfe und dedizierte Schalter – beim DMSO2D72 ist das kaum anders. Immerhin bringt das Gerät eine (leider recht langsame) Automatik mit, die auf Knopfdruck schon mal ein brauchbares, stehendes Bild erzeugt. Im Unterschied zu diversen Billigst-Angeboten liegt die Bandbreite bei durchaus respektablem 70 MHz bei 250 MSamples/s Abtastrate (einkanalig), für Messungen an Maker-typischen Schaltungen ist das völlig ausreichend. Eine zusätzliche Messfunktion ermittelt Spitzenwerte sowie die Frequenz des Eingangssignals. Auf die fortgeschrittenen Trigger-Modi, Signalanalyse- und Zoom-Funktionen von Benchtop-Digitaloszilloskopen muss man indes verzichten. Für die Oszilloskop-Funktionen stehen zwei separate BNC-Eingangsbuchsen zur Verfügung, man kann also handelsübliche Prüfspitzen verwenden. Die aufgenommenen Wellenformen lassen sich

über eine Software auch auf den PC laden; leider gehen dabei die eingestellten Parameter verloren.

Der Funktionsgenerator (mit eigener BNC-Ausgangsbuchse) arbeitet grundsätzlich unabhängig von Oszilloskop und DMM; neben Sinus (1Hz bis 25MHz), Rechteck (bis 10MHz) und Dreieck/Sägezahn stehen auch arbiträre Wellenformen zur Auswahl; letztere können mit einem Hilfsprogramm für Windows-PCs angelegt werden. Der Sinus weist einen Klirrfaktor von 0,015 Prozent auf, ein guter Wert, der aber leider im unteren und oberen Pegelbereich auf bis zu 0,24 Prozent ansteigt. Kleine Schönheitsfehler: Die eingestellte Ausgangsspannung (Effektivwert) gilt nur für das Rechtecksignal, beim Sinus muss man den Wert mit dem Faktor 0,707 multiplizieren. Es lässt sich zwar ein Gleichspannungs-Offset addieren, der maximale Ausgangspegel von 2,5V reicht aber nicht aus, um 5V-Digitalschaltungen anzusteuern.

Der Digitalmultimeter-Teil arbeitet mit einer Auflösung von 4000 Counts, was 3,5 gültigen Stellen entspricht; die Abweichungen bei Spannungsmessungen zu unserem jüngst kalibrierten Agilent 34401A (6,5 Stellen)



Ausprobiert
— von Make! —

len) betragen maximal 0,3 Prozent. Die Durchgangsprüfer-Funktion arbeitet angenehm spontan. Vorbildlich war bei unserem Testgerät die Genauigkeit in den Widerstands- und Kondensator-Messbereichen, die Anzeige war „auf den Punkt“. Der Kapazitäts-Messbereich endet (wie bei vielen Multimetern) leider schon bei 100µF, dicke Netzteil-Elkos bleiben also außen vor.

—cm

Hersteller	Joy-IT
URL	https://joy-it.net
Preis	225 €

Creality Ender 3 V2

Günstiger FDM-3D-Druckerbausatz



Seit Juli ist mit dem Ender 3 V2 der Nachfolger des beliebten Bausatz-3D-Druckers Creality Ender 3 (siehe Make 5/19, S.18) erhältlich. Von diesem unterscheidet sich die neue Version durch Verbesserungen an Mechanik und Elektronik. So sind nun die lang vermissten Riemenspanner für die X- und Y-Achse eingebaut. Mittels Rändelschrauben sind so die Antriebsriemen sehr schnell korrekt justiert.

Das neue Druckbett besteht aus einer Glasplatte und soll für eine bessere Haftung des Druckobjekts sorgen. Außerdem soll sich dadurch die Aufheizzeit verkürzen, da es die Wärme besser leitet. Der Printer wird nun serienmäßig mit dem Silent-Mainboard geliefert, das durch verbesserte Schrittmotor-Treiberbausteine die Geräuschentwicklung deutlich verringert; das Nachrüsten dieses Boards (ausprobiert in Make 6/19, S.122) entfällt damit.

Auch das Display und die Bedienoberfläche sind völlig neu: Das höher auflösende Farb-Grafikdisplay zeigt nun eindeutig erkennbare Icons sowie die vergangene und die noch benötigte Druckzeit an. Bedient wird der Drucker nach wie vor über einen Dreh-/Druckknopf. Ergänzt wurde die Überarbeitung noch durch einen Transportknopf für den manuellen Filamenteinzug, eine Werkzeugschublade unter dem Druckbett und eine neue Druckkopfabdeckung.

Der Preis wird im Creality-Shop mit regulär 369 Euro angegeben, bei Redaktionsschluss galt allerdings noch ein Sonderpreis von 275,99 Euro. —hgb

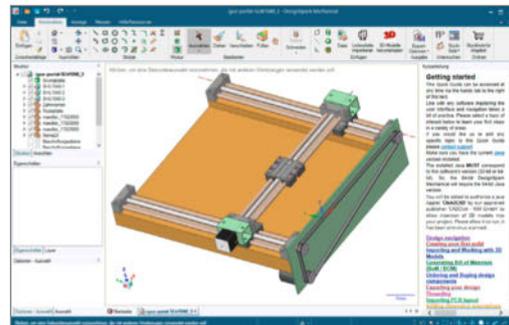
Hersteller	Creality
URL	creality3dshop.eu
Preis	ab 276 €

DesignSpark Mechanical V5.0

Kostenloses CAD-Tool

Hinter dem Konstruktionsprogramm DesignSpark Mechanical (DSM), das der Elektronikhändler RS Components auf seiner Webseite nach Registrierung kostenlos zum Download anbietet, steckt eine funktionsreduzierte Version der Direktmodellierungs-CAD-Software SpaceClaim Engineer (siehe auch Make 4/18, S.90). Jetzt ist Version V5.0 zum Download bereitgestellt worden und bietet etwa neue Zeichenwerkzeuge für die Arbeit mit Beschränkungen, um zum Beispiel parallele Linien zu zeichnen. Komplexe Gebilde wie in sich nochmals gewundene Spiralen kann man jetzt mit Hilfe von Funktionsgleichungen definieren.

Die Software bietet mehr Informationen als bisher über die Eigenschaften von Objekten, etwa das Volumen von Körpern, sowie über deren Qualität, etwa die Krümmung von Oberflächen. Zudem kann man die 3D-Darstellung von Konstruktionen besser steuern, auch auf Ebene einzelner Objekte, um die Balance zwischen Performance und visueller Qualität zu optimieren. Besonders spektakulär ist der neue Flythrough-Modus, bei dem man mit der virtuellen Kamera ins Innere komplexer Konstruktionen hineinfliegen kann.



DesignSpark Mechanical V5.0 läuft unter Windows ab Version 7, allerdings ausschließlich in der 64-Bit-Version. Wer noch mit 32 Bit unterwegs ist, muss sich mit V2.1 zufrieden geben, das immer noch zum Download angeboten wird. —pek

Anbieter	DesignSpark / RS Components
URL	rs-online.com/designspark/mechanical-software
Preis	kostenlos

Camera Scheduler

Kamera-Anschluss-Verdoppler für Rasperrys

Die amerikanische Firma Waveshare bietet ein Adapter-Set zum Anschluss zweier Kameramodule an den CSI-Slot der Rasperrys an. Das Set inklusive Anschlusskabel für Raspberry 1 bis 4 sowie Zero und zweier Jumperkabel zur Verbindung mit dem GPIO-Port kostet 12 Dollar.

Die komplett auf einem Folienleiter aufgebaute Elektronik ermöglicht den Anschluss zweier Kameramodule am Raspi. Die gleichzeitige Übertragung beider Bilder parallel ist nicht möglich, lediglich die Umschaltung zwischen den Kameras. Dazu muss an einem Lötspunkt auf dem Folienleiter eine Schaltspannung (3,6V) angelegt werden, die beispielsweise von einem GPIO-Pin durch den Raspi softwaregesteuert geschaltet werden kann (siehe Link). Da das Platinenmaterial allerdings empfindlich gegen hohe Temperaturen ist, braucht man da schon etwas Lötfernung, um Beschädigungen zu vermeiden.

Die Kameramodule müssen den gleichen Sensortyp enthalten, der gemischte Betrieb beispielsweise von Modulen unterschiedli-



cher Auflösung ist nicht möglich. Das Modul selbst ermöglicht nur einen Abstand von gut 10cm zwischen den Kameras. Für eine längere Distanz sind entsprechende zusätzliche Folienkabel notwendig. —hgb

► make-magazin.de/x1vt

Hersteller	Waveshare
URL	www.waveshare.com
Preis	12 US-\$

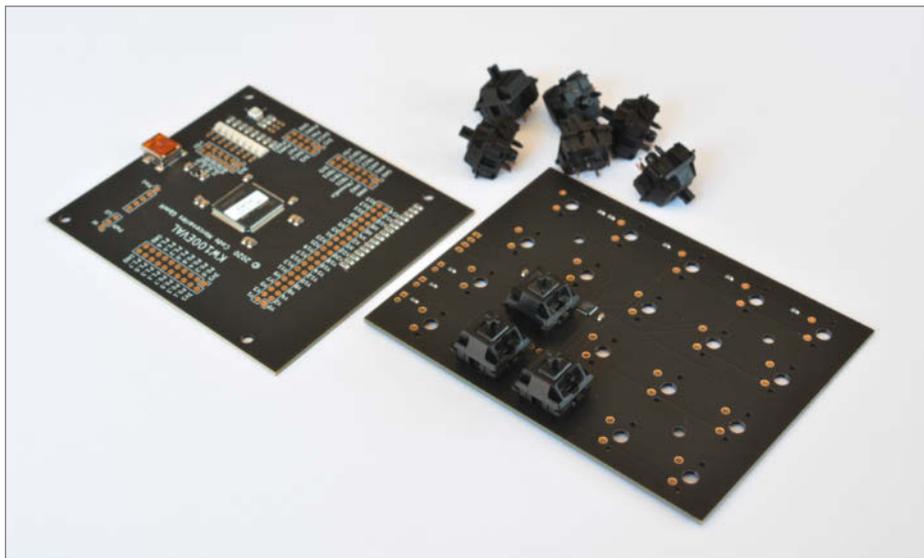
KW100EVAL und KW28EVAL

Starterkits für USB-Tastaturcontroller

Das rund 34 Euro teure Kit KW100EVAL (links im Bild) ist ein 8cm x 8cm großes Breakout-Board für den KeyWarrior100 mit Mini-USB-Buchse, acht Status-LEDs und einer RGB-LED. Alle Signale zum Anschluss selbstgebauter Matrix-Tastaturen sind auf doppelreihige Pads im 2,54mm-Raster geführt. 40 Leitungen sind für die Tastenmatrix (24 Zeilen x 16 Spalten) reserviert, somit können bis zu 384 Tasten abgefragt werden.

Daneben gibt es noch 16 universelle I/Os, analoge Eingänge zur Simulation einer Maus über Potenziometer, Messwertaufnehmer oder analoge Joysticks und sogar PWM- und DAC-Ausgänge zur Ausgabe quasi-analoger Spannungen. An den PWM-Ausgängen ist die RGB-LED angeschlossen, die man über Leiterplatten-Jumper abtrennen kann, wenn die Signale anderweitig verwendet werden sollen. Da sich der KW100 als ganz normales *Human Interface Device* (HID) am Rechner anmeldet, wird zur Ansteuerung kein Treiber benötigt; Libraries für beliebige Programmiersprachen (C, Delphi) gibt es zum Download. Für die Konfiguration bietet Code Mercenaries ein kostenfreies Tool an, mit dem alle Einstellungen und Tastaturdaten angepasst werden können.

Bald lieferbar ist auch das KW28EVAL, mit dem kleinere Tastaturen (zum Beispiel für Ar-



cade-Automaten) realisiert werden können. Praktischerweise können 20 Cherry-MX-Taster direkt auf die Platine gelötet werden. Der USB-Chip KeyWarrior28 unterstützt Matrix-Tastaturen im Raster 8 x 8, was Tastaturen mit bis zu 64 Tasten ermöglicht. Wie beim KW100 kann man Tastaturmakros definieren, die beim Betätigen eine Tastensequenz abspielen. Den

Chip gibt es bereits auf einem kompakten Breakout-Board, die Platine mit den Cherry-Tastern ist in Vorbereitung. —cm

Hersteller	Code Mercenaries
URL	shop.codemercs.com
Preis	34 € (KW100EVAL), 17,50 € (Board mit KeyWarrior28)

Tronxy X5SA

Großformat-3D-Drucker

Das Auffälligste an diesem FDM-Drucker-Bausatz sind die ungewöhnlich großen Abmessungen von 66cm x 58cm x 64cm der fertigen Maschine, die im Vergleich zu anderen Bausatz-Druckern der Preisklasse um 300 Euro enormen Bauraum von 33cm x 33cm x 40cm bietet. Zur Ausstattung gehören ein farbiger 3,5-Zoll-Touchscreen, ein beheiztes Druckbett sowie ein Leveling- und Filament-Sensor. Die maximale Temperatur der Druckplattform beträgt 100 °C, die des Druckkopfes 270 °C. Die Kunststoff-Druckauflage ist auf einer zusätzlichen Aluplatte aufgeklebt und sorgt für eine gute Haftung des Filaments.

Wie die Kollegen von TechStage in ihrem intensiven Test der Maschine feststellten (ausführlich online nachzulesen, siehe Link

am Ende), ist der Aufbau in etwa sechs Stunden zu bewältigen – könnte aber auch in der halben Zeit zu schaffen sein, wenn die Anleitung besser wäre, daher ist das kein Gerät für Einsteiger. Die Druckqualität erwies sich von Anfang an als sehr ordentlich, aber der Druckkopf-Lüfter lärmt unangenehm. Das fällt umso störender ins Gewicht, weil die Druckgeschwindigkeit des Geräts mit 20-100mm/s nicht den Rahmen des üblichen sprengt – in Kombination mit der Düsenbohrung und damit maximalen Schichtdicke von 0,4mm ergeben sich zwangsläufig Druckzeiten von vielen Stunden oder gar Tagen, wenn man den großen Bauraum wirklich ausnutzen will. —pek

► make-magazin.de/x1vt

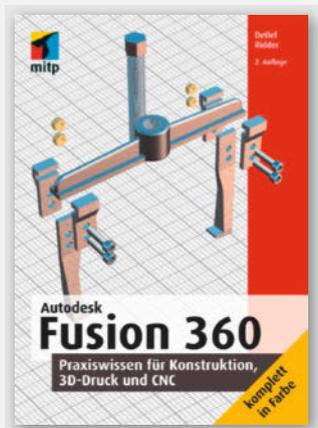


Hersteller	Tronxy
URL	tronxy.com
Preis	ab 300 €

Bild: Tronxy

Autodesk Fusion 360

Praxiswissen für Konstruktion, 3D-Druck und CNC



Ein Buch, das wir bereits in der ersten Auflage besprochen haben, kommt hier nur ausnahmsweise noch mal zur Sprache – wie in diesem Fall. Denn die erste Fassung dieses Buches zur CAD-Anwendung *Fusion 360* stuften wir seinerzeit als wenig brauchbar ein, was vor allem an den durchweg schwarzweißen Bildern lag (siehe Make 5/19, S.126). Die jetzt erhältliche zweite Auflage ist hingegen reich mit farbigen Bildern bestückt, was nicht nur hübscher aussieht und mehr Lust aufs Lesen und Konstruieren weckt, sondern die Anleitungen überhaupt erst nachvollziehbar macht. Darüber hinaus ist auch der Inhalt überarbeitet und neu systematisiert worden: So findet man alle Informationen zu CNC-Arbeitsgängen jetzt in einem Kapitel (statt wie bisher auf zwei unterschiedliche Blöcke im Buches verteilt); Flächen-, Freiform- und Netzmodellierung haben eigene Kapitel bekommen und werden durch zusätzliche Beispiele erläutert.

Damit hinterlässt das Buch einen grundsätzlich anderen Eindruck als in der ersten Auflage: Es gibt einen brauchbaren und breiten Überblick über *Fusion 360*, kombiniert mit detaillierten Anleitungen für Aufgaben von der 2D-Skizze über 3D-Modellierung bis hin zu CNC, 3D-Druck, Blechbearbeitung, Rendering und Animation. —pek

Autor	Detlef Ridder
Verlag	mitp
Umfang	400 Seiten
ISBN	978-3-7475-0158-0
Preis	29,99 € (Buch), 25,99 € (E-Book), 34,99 € (Bundle)

Die Ideenwerkstatt - Band 3

Scale-Modellbau 2

Zwar gibt es zwischen Makern und Modellbauern eine größere Schnittmenge, dennoch wird das auf RC-Helikopter spezialisierte *Rotor-Magazin* nicht jedem geläufig sein, der *Make* liest. Das vorliegende Bändchen versammelt ausgewählte und aktualisierte Grundlagen-Artikel aus dem *Rotor-Magazin*, etwa zum Drehen, Löten und Bohren sowie eine Airbrush-FAQ für Einsteiger. Der Arduino-Einsatz für die Servo- oder Beleuchtungssteuerung ist ebenso Thema wie CNC-gefräste Instrumententafeln im Maßstab 1:12 – zugegeben, andernfalls wären wir auf dieses Buch wohl gar nicht aufmerksam geworden.

Da andere Kapitel sich mit speziellen Flugmodellthemen wie dem Bau von Einziehfahrwerken beschäftigen, liegt der Nutzwert für alle Nicht-Modellpiloten vor allem im Blick über den Tellerand. Liebevolle Maker, für die

auch die Optik ihrer Projekte eine entscheidende Rolle spielt, finden hier Inspiration: Sie lernen in diesem Buch etwa, wie man aus Getränkedosenblech mit einem Pausrad aus dem Nähbedarf Nieten *en miniature* imitiert oder durch Farbe für das authentische *Weathering* sorgt, eine maßstabsgetreue Patina. Ganz ähnlich vom Konzept und Umfang her ist auch der noch erhältliche Band 1 der *Ideen-Werkstatt* gestaltet, der unter anderem Kleben, Tiefziehen und Gewindeschneiden behandelt und zeigt, wie man Hitzeverfärbungen auf Abgasrohren simuliert. —pek



Autor	Gunther und Frederick Winkle
Verlag	MSV Medien, Rotor Edition
Umfang	98 Seiten
ISBN	978-3-923142-95-8
Preis	19,90 €

Das 3D-Scanner-Praxisbuch

Grundlagen, Nachbau, Nachbearbeitung

Mario Lukas, der Autor dieses Buches, ist in Maker-Kreisen als Kopf des *FabScanPi*-Projekts bekannt, eines Open-Hardware-3D-Scanners auf Basis des Raspberry Pi. Insofern ist es kaum verwunderlich, dass sich im Buch auch eine ausführliche Bau- und Bedienungsanleitung für diesen Scanner findet. Dennoch belegt dieser Abschnitt weniger als ein Zehntel der Seiten des Buches, denn neben der dort benutzen Linienlaser-Methode gibt es noch andere Verfahren, um ein dreidimensionales Objekt in einen Datensatz zu verwandeln, etwa die Photogrammetrie oder den Scan mit Hilfe von Tiefensensoren und Lichtmuster-Projektion.

Das Buch stellt alle diese Verfahren gleichberechtigt und ausführlich vor, und zwar stets in Form praktischer Anleitungen, die mit Maker-Mitteln nachvollziehbar sind. Dazu ist es gespickt mit vielen Tipps und Tricks – so er-

fährt man etwa, wie man den proprietären Kinect-2.0-Tiefensensor für die Xbox fit für den Anschluss an einen Computer über USB 3.0 macht. Ein weiterer Pluspunkt: Fürs Scannen und den fast ebenso wichtigen Nachbearbeitungs-Workflow kommt in diesem Buch ausschließlich kostenlose Software zum Einsatz.

Dass jemand ein praktisch nutzbares Buch übers 3D-Scannen auf Deutsch schreibt, war schon lange mal fällig – Mario Lukas hat diese Aufgabe hervorragend gelöst und damit wahrscheinlich einen echten Maker-Buch-Klassiker geschaffen. —pek



Autor	Mario Lukas
Verlag	dpunkt
Umfang	352 Seiten
ISBN	978-3-86490-640-4
Preis	34,90 € (Buch), 27,99 € (PDF, ePub, Mobi)

Bauen, erleben, begreifen: fischertechnik-Roboter mit Arduino

Modelle, Steuerung, Programmierung

Eigene Roboter zu bauen ist lehrreich und macht Spaß – besonders dann, wenn man alle benötigten Teile parat hat und nicht lange suchen und improvisieren muss. Mit einem Produkt wie *fischertechnik* ist das möglich, denn es bietet hohe Freiheitsgrade und ist einsteigerfreundlich.

Die Robotiksysteme des Herstellers sind aber recht teuer und so haben Dirk Fox und Thomas Püttmann beschlossen, ihren Projekten mit einem Arduino und günstigen Bauteilen Leben einzuhauchen. Außer C/C++-Kenntnissen setzen die Autoren nichts voraus und sie beginnen mit einer kurzen Einführung in die Hard- und Software des Arduino. Dabei gehen sie insbesondere auf die Aspekte ein, die im Zusammenspiel mit den fischertechnik-Bauteilen wichtig sind, zum Beispiel die Stromversorgung und die Befestigung des Boards.

Anhand von fünf Robotik-Projekten zeigen sie, was alles mit wenig Aufwand möglich ist.

Neben zwei intelligenten Fahrzeugen basteln sie einen Plotter, einen Greifer und einen Roboter, der Objekte sortieren und Tic-Tac-Toe spielen kann. Dank der vielen Abbildungen und Fotos sollte der Nachbau auch Anfängern leicht von der Hand gehen. Die Beschaffung aller Bauteile kostet ein paar Euro, liegt aber gewiss unter dem Preis des Original-Zubehörs. Auf der Webseite zum Buch gibt es den Quelltext für alle Projekte, detaillierte Teile-Listen, 3D-Bauanleitungen, Datenblätter und vieles mehr.

—Maik Schmidt/dab

Autoren	Dirk Fox, Thomas Püttmann
Verlag	dpunkt
Umfang	456 Seiten
ISBN	978-3-8649-0426-4
Preis	32,90 € (Buch), 25,99 € (PDF, ePub, Mobi)



Das total verrückte Küchenlabor

Unser Essen unter der Lupe

Der britische Autor Stefan Gates hat ein Buch geschrieben, das – wie der Titel schon verrät – genauso gut in der Küche wie im Chemielabor liegen kann. Auf knapp 100 reich bebilderten Seiten erklärt er die chemischen und biologischen Grundlagen von Essen. Die Informationen erstrecken sich von den Prozessen im menschlichen Körper, also zum Beispiel unserer Wahrnehmung von Essen, über den Nährwertgehalt oder die Geschichte von verschiedenen Lebensmitteln bis hin

zu chemischen Prozessen beim Reifen, Verschimmeln oder Garen.

Die 20 Experimente im Buch umfassen beispielsweise das Sichtbarmachen von Eiseiteilchen in Müsli, das Auflösen einer Eierschale und ein Rezept für Cola. Alle Experimente lassen sich mit Haushaltsgegenständen durchführen und sind für Kinder leicht nachvollziehbar. Schön übersichtlich und informativ ist auch der mehrseitige Abschnitt zum Für und Wider von Ernährung mit oder ohne Fleisch.



Die Lektüre bringt viele neue Erkenntnisse über Essen mit sich – besonders die Seiten über exotische Lebensmittel wie die Stinkfrucht oder tausendjährige Eier und die Beschreibungen derer Gerüche sind informativ – und sehr lustig. Außerdem findet sich

auf der Doppelseite über Verdauungsprodukte der wunderschöne Satzanfang „Jeder Furz ist anders ...“. Wer dabei oder bei dem Experiment mit Speichel im Vanillepudding nicht neugierig wird, für den ist das Buch aber vielleicht eher doch nichts.

—esk

Autor	Stefan Gates
Verlag	Dorling Kindersley
Umfang	96 Seiten
ISBN	978-3-8310-3921-0
Preis	14,95 €

IMPRESSUM

Make: Nächste Ausgabe erscheint am 1. Oktober 2020

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: Debra Ansell, Beetlebum, Tyler Capps, Werner Fugmann, Andreas Gräßer, Veikko Krypczyk, Matthias Mett, Niq Oltman, Roman Radtke, Niklas Roy, Maik Schmidt, Daniel Springwald, Carsten Wartmann

Assistenz: Susanne Cölle (suc), Christopher Tränkmann (cht), Martin Triadan (mat)

Leiterin Produktion: Tine (The Rock) Kreye

DTP-Produktion: Nicole Judith Hoehne (Ltg.), Martina Bruns, Martin Krefit (Korrektur)

Art Direction: Martina Bruns (Junior Art Director)

Layout-Konzept: Martina Bruns

Layout: Nicole Wesche

Fotografie und Titelbild: 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: Daniel Rohlfing (-844) (verantwortlich für den Anzeigenteil), www.heise.de/mediadaten/make

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
 Meißberg 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.

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 Verlages in irgendeiner Form reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Alle beschriebenen Projekte sind ausschließlich für den privaten, nicht kommerziellen Gebrauch. Maker Media GmbH behält sich alle Nutzungsrechte vor, sofern keine andere Lizenz für Software und Hardware explizit genannt ist.

Für unverlangt eingesandte Manuskripte kann keine Haftung übernommen werden. Mit Übergabe der Manuskripte und Bilder an die Redaktion erteilt der Verfasser dem Verlag das Exklusivrecht zur Veröffentlichung. Honorierte Arbeiten gehen in das Verfügungsrecht des Verlages über. Sämtliche Veröffentlichungen in Make erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes.

Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.

Published and distributed by Maker Media GmbH under license from Make Community LLC, United States of America. The 'Make:' trademark is owned by Make Community LLC. Content originally partly published in Make: Magazine and/or on www.makezine.com, ©Make Community LLC 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



Was für ein Roboter wolltest du schon immer mal sein?

Birgit Hellendahl

Wesseling, gibt auf Seite 64 einen Einblick in ihren Fräsenbau-Workshop

Ich wäre gerne ein Roboter, der blitzschnell von einem Ort zum anderen reisen kann, damit ich jederzeit bei meinen Freunden sein und überall Workshops geben kann.

Carsten Wartmann

Berlin, hat sich ab Seite 8 ausgiebig mit Retro-Gaming beschäftigt

Ich wäre gerne ein Roboter, der auf dem Saturnmond Enceladus landet, sich durchs Eis schmilzt und im Ozean nach Leben sucht. Dieser Ort ist ein guter Kandidat für außerirdisches Leben.

Anna Ludwig

Hannover, organisiert und plant begeistert unsere Maker Faires

Ich wäre gerne ein Roboter, der das Meer von Müll und Plastik befreit. Dann könnte ich den Lebensraum der Meeresbewohner schützen und dabei zusehen, wie sich das Meer wieder erholt.

Niklas Roy

Berlin, hat den Schaufenster-Plotter auf Seite 82 gebaut

Eigentlich will ich gar kein Roboter sein. Sonst müsste ich mir ja von Isaac Asimov Vorschriften machen lassen!

Inserentenverzeichnis

ALLNET Computersysteme GmbH, Germering	19	Sauter GmbH, Inning	25
BERNHARDT Kunststoffverarbeitungs GmbH 3dk.berlin, Berlin	53	Technische Universität Kaiserslautern, Kaiserslautern.....	7
Fachhochschule Wedel Gemeinnützige Schulgesellschaft mbH, Wedel	61	Wissenschaft im Dialog gGmbH, Berlin.....	55
Leibniz Universität Hannover, Garbsen.....	43	Make:markt	63
Mouser Electronics Inc., München	13		

SMART LOSLEGEN

mit dem Node-RED Special von Make!

dt Make: **Node-RED SPECIAL**

Grundlagen

- › Einführung in Node-RED
- › MQTT verstehen
- › Installation auf dem Pi
- › Flows erstellen
- › ESP8266 einrichten & programmieren
- › Sensoren auslesen & Motoren steuern
- › Bedienoberflächen entwerfen
- › Einstieg in JavaScript

Für Smart Home & IoT

Projekte

- › Temperaturen messen
- › Fenstersensoren überwachen
- › Servomotor als Anzeige

Für Fritzbox

- › Präsenzerkennung im WLAN
- › Bandbreitenanzeige mit LED-Streifen
- › DECT-Geräte

inklusive ESP8266 D1 Mini mit WLAN

NEU

Make Special: Node-RED

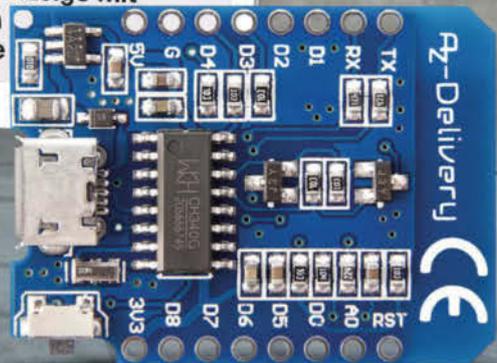
Für Einsteiger ohne tiefere Programmier-Kenntnisse bietet **Node-RED** die ideale Lösung. Man verknüpft fertige grafische Bausteine zu einem „Flow“ und kann so komplexe Anwendungen für **IoT und Smart Home** entwerfen.

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 € >

inkl. D1 Mini



 **heise shop**

shop.heise.de/ma-nodered >

› Generell portofreie Lieferung für Heise Medien- oder Maker Media Zeitschriften-Abonnenten oder ab einem Einkaufswert von 15 €. Nur solange der Vorrat reicht. Preisänderungen vorbehalten.

© Copyright by Maker Media Club



I WANT
YOU

Maker Faire®

Vernetzen • Austauschen • Präsentieren

**Werde Teil von Deutschlands größtem
Maker-Community-Netzwerk!**

Nutze unsere Plattform und präsentiere
kostenfrei deine Idee auf einer der
deutschlandweiten Maker Faires.

maker-faire.de/call-for-makers

© Copyright by Maker Media GmbH.