DEUTSCHLANDS GEFÄHRLICHSTES DIY-MAGAZIN

Cit Make:

Retrievent of the selbst betreiben und remote nutzen

Smartes Spielbrett bauen

- Memo-Spiel mit Lichtmustern
- Mit Arduino und Reflexlichtschranke
- Clevere Spielsteinerkennung

lest: alliope mini 3

Projekte

DIY-Röhrenvorverstärker bauen
 LoRaWAN: Nistkästen überwachen
 Smart Home als Messlabor

Know-how

Materialkunde f
ür Lasercutter
 ATtiny programmieren und flashen
 Checkliste: Smart-Home-Wartung

Thermostate ins Netz bringen

 Integration in ioBroker
 Spart viel Energie



IKEA-Würfel hacken

Aufrüsten mit ESP8266
 Smart-Home-fähig mit MQTT



2/24 5.4.2024 CH CHF 26.50 AT 14,90 Benelux 15,90



Ein Veranstaltungsformat der Make:



Maker Faire

Das Format für Innovation & Macherkultur

Die nächsten Events



0



HANNOVER Maker Faire C



... weitere folgen.

maker-faire.de



Ohne mich, Patronenmafia!

Druckerhersteller sind geübt darin, uns das Leben schwer zu machen: Jedes Druckermodell hat seine eigene Patronenbauform, eine leere Farbpatrone sperrt auch den Schwarz-Weiß-Druck und der Chip in den Patronen soll günstige Fremdtinte und Nachfüllen verhindern. In den Druckern sind Verschleißteile wie die Kissen zum Reinigen der Düsen und der Druckkopf wartungsfeindlich verbaut.

Der neuste Streich der Patronenmafia kommt von HP: Böse Hacker könnten Viren auf den Kopierschutzchips von Dritthersteller-Patronen verstecken, die sich dann in den Drucker schleichen, um von dort Schabernack im heimischen Netzwerk zu treiben, warnt der Hersteller (siehe Link). Davon abgesehen, dass HP selber die Idee zu diesem Angriffsszenario geliefert hat, ist HPs Schlussfolgerung nicht etwa, die eigenen Patronen nicht länger mit eigentlich unnötigen Chips zu verdongeln, und dieser Masche so einen Riegel vorzuschieben.

Stattdessen hält man es für angemessen, bereits verkaufte Drucker einfach per Firmware stillzulegen, wenn sie einmal mit einer Fremdpatrone in Berührung gekommen sind. Zum Schutz der Verbraucher, natürlich. Nebenbei gibt HP-Chef Enrique Lores ganz offen zu, dass sie mit jedem Druckerverkauf Verlust machen und die Kohle über die völlig überteuerten Patronen reinholen: "Das ist eben das Geschäftsmodell".

Ich habe auf das Drucker-Drama keine Lust mehr. Unser Tinten-Multifunktionsgerät habe ich zum Scanner degradiert. Für die paar Ausdrucke, die ich im Jahr benötige, habe ich einen billigen SW-Laserdrucker ohne verdongelten Toner gebraucht gekauft, für Text reicht das allemal. Fotos schaue ich mir in Zukunft lieber digital an und die Großeltern bekommen einmal im Jahr ein Fotobuch.

Wenn die Angebote des freien Marktes so bewusst darauf abzielen, mich abzuzocken, übe ich mich lieber in Verzicht. Stattdessen werde ich versuchen, unserem Nachwuchs beizubringen, sich vor dem Kauf über Produkt und Geschäftsmodell zu informieren. Ich hoffe, ihn dafür begeistern zu können, langlebige Dinge wertzuschätzen und im Falle eines Defektes Geräte aufzuschrauben, zu reparieren, anzupassen, alternative Lösungen zu finden oder gar die Firmwares von Clouds und Knebeln zu befreien. Wir können es uns nicht länger leisten, alles wegzuwerfen, was kaputtgeht. Erst recht nicht, wenn der Defekt durch geplante Obsoleszenz simpler Bauteile ausgelöst ist. Daher: Repariert, was nicht repariert werden soll! Nehmt auseinander, kombiniert neu, setzt wieder zusammen! Teilt 3D-Druck-Daten für Ersatzteile im Netz! Hackt den Herstellern den Angstschweiß auf die Stirn! Schreibt eure lokalen Politiker an, sie mögen doch bitte das im Koalitionsvertrag vereinbarte Recht auf Reparatur voranbringen.

Um so mehr hat mich die rege Nachfrage nach unserem Reparatur-Heft im Herbst gefreut. Herzlichen Dank! Nicht wegen der verkauften Hefte (okay, das auch), sondern weil ihr euch für das Reparieren interessiert. Diesen Weg sollten wir gemeinsam weitergehen.

Zuversichtliche Grüße

hamos Bornsen

Johannes Börnsen

make-magazin.de/xyhy

Apropos "gemeinsam weitergehen"

Wir möchten wissen, was Sie sich in Zukunft von der Make wünschen – welche Themen und Projekte Sie besonders interessieren oder was sie nicht mehr lesen wollen. Bis Ende Mai können Sie uns das in einer Umfrage mitteilen. Nutzen Sie die Chance und gestalten Sie die Make mit. Den Link zur Umfrage finden Sie nach der Short-URL oder über den QR-Code.





Smartes Spielbrett bauen

Memory einmal anders, mit Kugeln statt Karten und Auswerte-Elektronik auf einem schicken Spielbrett. Merken Sie sich, welche Lichtmuster beim Auflegen der Kugeln erscheinen. Haben Sie zwei gleiche Muster erwischt, lassen Sie die Kugeln liegen und sie werden mit der dem Spieler zugeordneten Farbe dauerhaft eingefärbt. Falls nicht: Kugeln wieder runternehmen und der Nächste macht weiter. Umweltfreundlich ist das Ganze auch, denn die Kugeln sind Reste aus alten Deorollern.

8 Licht-Memo-Spiel

Inhalt

Projekte

Auf den guten, alten Röhren-Sound schwören ja viele. Hier erfahren Sie, wie man mit den elektrisch beheizten Verstärker-Elementen einen hochwertigen Vorverstärker für die E-Gitarre baut. Haben Sie jedoch eher einen Sinn für Kameras und andere Akku-betriebene Geräte? Mit dem Smart Home können Sie deren Energiespeicher testen. Per Akku läuft auch der LoRa-WAN-Nistkasten, der per Waage die Belegung checkt.

- 16 DIY-Röhrenvorverstärker bauen
- 62 Smart Home als Messlabor

88 LoRaWAN: Nistkasten überwachen



- 3 Editorial: Ohne mich, Patronenmafia!
- 6 Leserforum
- 8 Projekt: Licht-Memo-Spiel
- 16 Report: DIY-Röhren-Vorverstärker
- 24 Projekt: IKEA-Leuchtwürfel-Hack
- 32 Make Online
- 34 Projekt: Thermostate clever hacken
- 42 Maker Faire Update
- 46 Test: Calliope mini 3 ausprobiert
- 50 Test: Die Jeder-kann-Drucker: Prusa MK4 vs. Bambu Lab P1S
- 56 Report: Frühjahrsputz im Smart Home
- 62 Projekt: Messen mit dem Smart Home
- 72 Workshop: ATtiny statt Arduino
- 80 Report: Making unter Palmen
- 84 Community-Projekt: USB Sleuth – selbstgebauter USB-Kabeltester
- 86 Community-Projekt: MuPiBox Musikplayer

Know-how

Hausputz im Smart Home hat nichts mit Staubsaugen zu tun, sondern es geht um die Entfernung überflüssiger Geräte-Einträge u. ä. aus der Datenbank des Servers. Hinterher ist dann alles viel übersichtlicher. Übrigens muss es bei vielen Projekten nicht immer ein Arduino sein: ATtinys reichen oft auch aus und sind echt winzig. Lasercutter-Besitzer erfahren schließlich, welches Material verwendbar ist, welches sie meiden sollten und was Babycreme mit Stahl zu tun hat.

56 Checkliste: Smart Home-Wartung
72 ATtiny programmieren und flashen
92 Materialkunde für Lasercutter



- 88 Community-Projekt: Nistkasten mit Anwesenheitsmeldung
- 90 Report: LoRaWAN-Outdoor-Gateway
- 92 Know-how: Material für Lasercutter
- **100** Projekt: Autofokus für Lasercutter nachrüsten
- 108 Reingeschaut: Überwachungskamera
- 110 Workshop: Private Maker-Tools auf dem Pi
- 116 Workshop: Reverse Proxy auf dem Raspberry Pi
- 122 Know-how: Speicheraufteilung bei Mikrocontrollern
- 126 Kurzvorstellungen: Bücher: Logik, Funktionale Programmierung verstehen, Produkte: Xinfrared T2S Plus, Synth Unit, Alpakka Controller, Califlower, KiCad 8, Cardputer, PaHUB2, ePulse Feather ESP32-C6
- 131 Impressum/Nachgefragt

Themen von der Titelseite sind rot gesetzt.

Thermostate ins Netz bringen

Ja, es gibt sie noch, sogar in relativ neuen Häusern: nicht-Smart-Home-fähige Raumthermostate. Mittels ESP32 und MQTT werden die jedoch zu maßgeschneiderten Wächtern über die Raumtemperatur mit schicken Touchkeys, die keine Smarthome-Wünsche mehr offen lassen. Obendrein sparen Sie damit auch noch zweifach Strom.

34 Thermostate clever hacken



Ikea-Würfel hacken

Die IKEA-Würfelleuchte FREQVENS kann nicht allzu viel: Geräuschabhängig zeigt sie ein paar Animationen auf

ihrer LED-Matrix an, mehr nicht. Ein ESP8266-Board bringt den kleinen Kubus aber auf Trab und via MQTT sogar ins Smart Home. Dann kann er als Uhr arbeiten oder alle möglichen Texte und Messwerte anzeigen. So wird er sicher öfter "FREQVENtiert".



24 IKEA-Leuchtwürfel-Hack

Leserforum

Keine Internetverbindung

Universal-IR-Fernbedienung, Make 1/24, S. 72

Die Lösung mit dem ESP als WiFi-AP hatte ich für andere Dinge auch schon im Sinn, hat aber einen Nachteil: Wenn sich das Smartphone mit dem WLAN verbindet, erwartet es auf diesem Wege auch ins Internet zu kommen. Die meisten Geräte nutzen nur entweder WLAN oder mobile Daten. D.h., während das Handy mit der Fernbedienung verbunden ist, geht kein Threema/Signal/WhatsApp usw. und natürlich auch sonst nichts, was eine Internetverbindung braucht.

Außerdem hatte ich auch schon Smartphones, die sich eigenständig aus WLANs wieder abmelden, wenn sie merken, dass sie darüber nicht ins Netz kommen. Dann bevorzugen sie mobile Daten ...

Karl Fischer

Je nach Smartphone lässt sich das zweite Problem umgehen, da man z. B. Android anweisen kann, sich auch dann mit einem WLAN zu verbinden, wenn es keinen Internetzugang bietet. Aber Sie haben recht, in diesem Fall ist man vom Internet abgeschnitten. Möchte man darauf nicht verzichten, kann man die Fernbedienung auch in ein bestehendes WLAN einbinden, das Internetzugang bietet.

Zu wenig Automation

So interessant und nützlich die vorgestellte Lösung ist, wenn die Prämisse tatsächlich ist: " ... wenn es endlich geschafft ist, renne ich zur Tür. Der Besuch ist längst wieder gegangen", setzt sie viel zu spät und an der falschen Stelle an. Wenn es beim Kinoabend klingelt und ich Störungen überhaupt zulassen möchte, hat der Film von alleine zu pausieren, eine Beleuchtung geht gedimmt an, das Bild der Haustürkamera wird eingeblendet und ich kann per Sprache entscheiden, ob ich die Tür öffne, den Film abbreche und mich auf den Weg mache oder ob ich weiter schaue. Unter Umständen wird sogar die Verbindung zur Sprechanlage aktiviert. Und das alles, ohne dass ich eine Fernbedienung oder das Handy in die Hand nehme. Oder ich habe entschieden, dass beim Start des Filmes die Klingel erst mal stumm geschaltet wird und frühestens auf das zweite Klingeln innerhalb einer bestimmten Zeit reagiert. Niemand muss immer erreichbar sein.

Bei der Haus-Automation geht es um Automation, also darum, dass mir so viele Routine-Dinge wie möglich automatisch abgenommen werden – nicht, dass ich im Normalfall von Hand möglichst viel selber machen kann. Die Universalfernbedienung ist dabei nur ein kleiner Baustein, der zudem noch meist im Hintergrund bleibt und das oben angesprochene automatische Pausieren, Abbrechen oder Fortsetzen ermöglicht.

Andre Schröter

Brillenqualität

Die neue 40-Watt-Klasse, Make 1/24, S. 110

Hier würde mich generell mal die Qualität der Brillen interessieren, da sie für die Sicherheit des Benutzers elementar sind. Bei mir hinterlässt eine Schutzbrille ohne Zertifikat ein mulmiges Gefühl, wobei mich wahrscheinlich ein chinesisches Zertifikat auch nicht wirklich beruhigen würde ;-). Wenn ihr das in Tests angeben könntet, wäre das super, einmal ein Roundup verschiedener Modelle wäre auch interessant. So hätte man zumindest mal einen Überblick, wie die Lage ist (auch wenn es nur eine Momentaufnahme ist) und ob es empfehlenswert ist, nur zertifizierte Modelle zu nutzen, wobei ich dazu tendieren würde.

Eric Keim

Da können wir konkret noch nichts zu sagen, aber wir nehmen das mal in unsere To-Do-Liste auf.

Brücke

Lasern mit Lightburn, Make 1/24, S. 24

Zum "Sonderfall K40" (S. 32): Es gibt mit MeerK40t (MeerKat gesprochen) eine Open-Source-Software für K40-Laser (https://github. com/meerk40t/meerk40t). Dazu gehört ein kleines Programm namens "ruidacontrol". Wenn dieses läuft, tut es so, als sei es ein Ruida-Controller, der dann z. B. von Lightburn angesprochen werden kann. Damit funktioniert MeerK40t als Brücke zwischen Lightburn und dem K40. Angeblich geht das auch ohne GUI z. B. auf einem Raspi. Vielleicht ließe sich mit dieser Lösung ein Controller-Austausch umschiffen. Das MeerK40t-Readme listet die unterstützten Controller auf.

Peter Eisner

Herbe Enttäuschung

Make 1/24

Als langjähriger Leser eurer Zeitschrift bin ich jedes Mal gespannt auf die jeweils neue Zeitschrift. Ich habe bisher immer interessante Projekte oder zu mindestens Inspirationen gefunden. Dieses Mal wurde ich herb enttäuscht. Den Inhalt des aktuellen Heftes empfinde ich gelinde gesagt als Schrott! Fast nichts, was man sinnvoll verwenden kann. Auch wenn Lightburn eine gute Software zu sein scheint, finde ich es übertrieben, hier so viele Seiten darüber zu schreiben. Im DIY-Bereich sollte es doch viel mehr Open Source sein!

Auch dass für die Aufmacher mit riesigen Bildern eine komplette Seite verschwendet wird, finde ich eher Ressourcenverschwendung. Ich hoffe, dass im nächsten Heft wieder interessante Beiträge enthalten sind. Wobei mir schon klar ist, dass die Vorstellungen verschiedener Leser schon sehr unterschiedlich sind.

Early-slf

Intermodellbau

Editorial: German Angst, Make 7/23, S. 3

Wie Sie den passenden Faden mit den Nörglern in der Make zum Thema Umgang mit Strom spinnen, da muss ich ihnen ein exorbitantes Lob aussprechen. Was mir nicht gefallen hat: Sie haben den Messeauftritt auf der Intermodellbau in Dortmund 2023 kurzfristig abgesagt - dabei waren sie einer der Sponsoren für die Messe. Ich fahre seit 30 Jahren zu dieser Messe – alleine, mit Freunden und Vereinskollegen – für mich ist es ein Muss, einmal im Jahr da gewesen zu sein. Die Make ist ein TüftlerMagazin, also gehören Sie und Ihr Team auch auf diese Messe. Sie zeigen Präsenz und was es in der Make gibt, in der sich der Tüftler verewigen kann, und Sie finden vielleicht Nachwuchsautoren für Ihr Magazin.

M. (per Brief)

Wir sind eine kleine Redaktion und Messen inklusive Standaufbau und Anreisen sind für uns immer sehr aufwändig. Daher konzentrieren wir uns eher auf Auftritte auf den Maker Faires. Aber wir nehmen Ihren Impuls gerne nochmal auf.

Schaltplan

The Undertoner, Make 1/24, S. 50

Können Sie einen Schaltplan für den Undertoner-Synthesizer aus Make 1/24 auf der Make-Magazin-Homepage zur Verfügung stellen oder mir per Mail zusenden? Obwohl ich schon oft Schaltungen auf Steckbrettern und Lochrasterplatinen aufgebaut habe, fällt es mir ohne Schaltplan schwer, die Funktion des Undertoners zu verstehen.

Christian Keck







Licht-Memo-Spiel

Statt Karten umzudrehen und sich Bilderpaare zu merken, erscheinen bei diesem Spiel unterschiedliche Lichtmuster, wenn man lichtdurchlässige Kugeln auf das Spielbrett legt. Wer die meisten Paare gefunden hat, gewinnt. Das Spiel kann man selber bauen und es hat durchaus einen gewissen Suchtfaktor.

von Ralf Stoffels



Kurzinfo

- » Gedächtnisspiel, bei dem Paare von bunten Blinkmustern gefunden werden müssen
- » Umsetzung mit Arduino, WS2812-LEDs (NeoPixel) und Reflexlichtschranken
- » Zum Nachbauen f
 ür Maker, die gerne und gut l
 öten

Checkliste



ein Wochenende Kosten:

- ca. 80 Euro (inklusive Glaskugeln statt Deoroller-Kugeln)

Material

- » 36 WS2812-LEDs auch NeoPixel genannt
- » 36 Reflexlichtschranken ITR8307 oder ähnlich
- » 6 PNP-Transistoren BD170 » Widerstände 6 × 4,7 kΩ, 36 × 22
- kΩ, 36 × 150 Ω
- » Arduino Nano
- » Steckernetzteil 5 V/2 A mit Stecker und passender Hohlbuchse
- » Litze und Draht für Verbindungen, optional auch Platinenmaterial oder Streifenrasterplatine
- » 36 Deorollerkugeln oder Glaskugeln oder Gummibälle (Durchmesser 25 bis 30 mm)
- » Holz für das Spielbrett, etwa 40 × 40 cm, 15 mm dick, dazu Leisten für einen Sockelrahmen

Werkzeug

übliches Maker-Werkzeug wie Lötkolben, Seitenschneider etc. Bohrmaschine und andere Holzbearbeitungswerkzeuge je nach Ausführung des Spielbretts

Mehr zum Thema

» Carsten Wartmann, Disco is back – Lichtshow mit WLED, Make 2/23, S. 8 Video: Das Spiel in der Praxis, vorgestellt vom Autor





iele meiner Maker-Projekte werden durch das Vorhandensein des Baumaterials inspiriert. Mal sind es elektronische Bauteile aus der Bastelkiste, ein ausgeschlachtetes Gerät oder ein Anzeigemodul, das ich auf der Maker Faire einfach kaufen musste. Bei diesem Projekt waren es die Kugeln von Deorollern, die in unserer vierköpfigen Familie über die Zeit anfielen und mir zu wertvoll zum Wegwerfen waren. Diese halbtransparenten Kugeln luden ein, mit Licht durchstrahlt zu werden. Ausgehend davon entstand ein Gedächtnis-Spiel mit erheblichem Suchtfaktor, dass sich übrigens auch mit anderen Kugeln nachbauen lässt - dazu mehr am Ende des Artikels.

Das Spiel

Jeder kennt vermutlich das klassische Memory-Spiel, bei dem es darum geht, aus vielen Karten, die verdeckt auf dem Tisch ausliegen,

Paare von Symbolen oder Bildern zu finden. Es dürfen immer nur zwei Karten aufgedeckt werden. Stimmen die Bilder der beiden Karten nicht überein, so werden sie wieder herumgedreht. Sobald ein passendes Paar gefunden wurde, darf der Spieler oder die Spielerin das Kartenpaar vom Tisch nehmen. Man muss also ein gutes Gedächtnis für die Kartenpositionen haben – ein Grund, warum meistens die jüngsten Kinder gewinnen.

Bei meinem Licht-Gedächtnisspiel ist das Spielfeld hingegen am Anfang leer. Die Spieler und Spielerinnen legen nun abwechselnd jeweils zwei Kugeln in freie Vertiefungen des Bretts. Jeder Vertiefung ist auf zufällige Weise ein charakteristisches Blinkmuster zugeordnet, das direkt nach dem Legen der Kugel zu sehen ist: Das kann z. B. die Folge Rot-Blau-Grün, Grün-Gelb oder weiß blinkend sein.

Über die 36 Felder sind so 18 Paare von unterschiedlichen Blinkmustern verteilt, die es zu finden gilt. Passen die beiden Blinkmus-





Bild 2: Schema der Reflexlichtschranke ITR8307



Bild 3: Löcher für LED (hell und mittig) und Reflexlichtschranke (dunkel und exzentrisch) braucht es in jeder einzelnen Vertiefung.

ter der gelegten Kugeln nicht zueinander, so nimmt die Spielerin diese wieder vom Brett. Hat sie ein passendes Paar gefunden, so färben sich die Kugeln in der Farbe der entsprechenden Spielerin und bleiben auf dem Brett liegen. So füllt sich nach und nach das Brett. Im YouTube-Video (siehe Link in der Kurzinfo) wird der Spielverlauf anschaulich.

Während beim Karten-Memory der Spieler gewinnt, der die meisten Karten vom Tisch nehmen konnte, gewinnt hier die Spielerin mit den meisten Kugeln auf dem Brett. Im Titelbild dieses Artikels hat also Türkis gewonnen.

Die Firmware des Spiels sorgt dafür, dass die Blinkmuster nach jedem Spiel zufällig neu verteilt werden. Jedes Spiel ist also eine neue Herausforderung für das visuelle Gedächtnis.

Vorversuche – Fehlversuche

Damit das Spiel ohne irgendwelche Bedienelemente wie Taster funktionieren kann, muss das Brett die Kugeln selbst erkennen und natürlich beleuchten. Da es 36 Kugeln gibt, musste ich eine Lösung finden, die möglichst wenige Bauteile pro Kugel braucht, um die Gesamtzahl der Teile nicht explodieren zu lassen.

Mir kam die zunächst bestechende Idee, RGB-LEDs zu verwenden, die als Lichtquelle und Detektor gleichzeitig funktionieren sollten. LEDs haben generell die Eigenschaft, dass sie nicht nur Licht emittieren können, sondern einfallendes Licht in eine elektrische Spannung wandeln, ähnlich einer Solarzelle. In einer RGB-LED sind drei verschiedenfarbige LEDs verbaut. Verwendet man die rote LED als Sender und misst die Spannung an der blauen LED, so erhält man tatsächlich eine Reflexlichtschranke (Bild 1). Solche Reflexlichtschranken senden Licht aus und registrieren, wie viel davon reflektiert zurückkommt. Sie werden beispielsweise in einfachen Robotern als Hindernis- und Distanzsensor eingesetzt.

Leider war die Idee in der Praxis doch nicht so genial. Mit einem weißen Blatt Papier als reflektierendem Objekt konnte ich zwar ca. 80 mV auf der Empfangsseite messen und mit ein wenig Verstärkung könnte das zur Erkennung reichen. Nur sind die Deokugeln im Unter-



Bild 4: Vereinfachtes Schaltbild mit Lichtschranken und Neopixel-LEDs

schied zum Papier transluzent – soll heißen, sie lassen viel Licht durch und reflektieren nur einen kleinen Teil, auch wenn sie nicht wirklich transparent sind. Dadurch verringert sich der Effekt auf wenige Millivolt, was vom Rauschen nicht mehr unterscheidbar ist. Also keine passende Lösung – aber vielleicht mal eine Idee oder Inspiration für ein anderes Projekt.

Es lief dann am Ende doch auf die Verwendung je einer separaten Reflexlichtschranke in jeder Vertiefung des Spielbretts hinaus. Ich habe mich für die ITR8307 entschieden, die im Elektronikversand für ca. fünfzig Cent pro Stück bestellbar sind. Diese enthalten eine Infrarot-LED und einen entsprechenden Fototransistor. Infrarotlicht hat den Vorteil, dass die Empfindlichkeit gegenüber Störlicht gering ist.

Achtung: Bei den ITR8307, die ich verwendet habe, gab es keine Gehäusemarkierung wie im Datenblatt und hier in Bild 2 gezeigt. Bei meiner Version war Pin 2 nur dadurch gekennzeichnet, dass dessen Anschlussdraht länger als die anderen drei Anschlüsse war. Hier sollte man gegebenenfalls die Diode mit einem Multimeter nachmessen, um sicherzugehen, dass man sie richtig herum anschließt.

Hätte meine erste Idee geklappt, das Lichtmuster und die Kugelerkennung mit einem einzigen Bauteil zu realisieren, hätte ich in jede Kugel-Kuhle des Spielbretts nur je ein Loch für die RGB-LED bohren müssen. Für die praxistaugliche Lösung braucht es hingegen pro Kugelplatz jeweils zwei Löcher: eines für die RGB-LED für das Lichtmuster und eines für die Reflexlichtschranke (Bild 3). Und natürlich sind damit insgesamt doppelt so viele Bauteile zu verdrahten.

Multiplexing

Wie bereits erwähnt, reflektieren die Kugeln das Licht der Reflexlichtschranke nur zu einem kleinen Teil. Das meiste Licht strahlt durch sie hindurch. Eine sichere Erkennung der Kugeln funktioniert daher nur, wenn der analoge Spannungswert am Fototransistor der Lichtschranke gemessen und mit der gespeicherten vorigen Messung verglichen wird. Denn hier kommt es darauf an, eine Änderung zu detektieren – die im Spiel dadurch passiert, dass eine Kugel hingelegt wird. Dafür muss laufend jeder Spannungswert der Lichtschranke ermittelt werden. Verringert sich die Spannung am Fototransistor um etwa zwei Prozent, wertet der Code das als hinzugelegte Kugel - mehr dazu später im Abschnitt zur Spielelogik.

Da die üblichen Mikrocontroller weit weniger als 36 analoge Eingänge haben, konnte ich nicht jeden der Fototransistoren einzeln anschließen. Der hier eingesetzte Arduino Nano stellt nur sieben Eingänge mit ADC (Analog Digital Converter) zur Verfügung. Diese reichen dennoch aus, weil ich jeweils alle Fototransistoren einer *Spalte* des Spielfelds parallel geschaltet habe. Jede der sechs Spalten ist dann mit einem der analogen ADC-Eingänge A0 bis A5 verbunden (Bild 4).

Ich verwende sechs Ausgänge des Arduino (D2 bis D7), um alle die Sende-LEDs der Reflexlichtschranken in je einer *Zeile* zu schalten. Wenn also zum Beispiel nur die Sende-LEDs der Zeile 3 aktiviert sind und der ADC von Spalte 5 ein Signal misst, dann gibt es eine Reflexion auf der Position x=5 und y=3. Auf diese Weise können alle 36 mögliche Kugelpositionen auf dem Feld mit nur sechs Eingängen und Ausgängen erfasst werden.

Die Infrarot-LEDs der Lichtschranken benötigen einzeln einen Strom von ca. 30 mA. Das bedeutet, dass jede Zeile mit 6×30 mA = 180 mA belastet wird – zu viel für die Ausgänge des Arduino. Ich habe daher jede Zeile mit einem PNP-Transistor BD170 geschaltet. HIGH am jeweiligen Arduino-Ausgang sperrt den Transistor, LOW schaltet ihn durch.

Das vereinfachte Schaltbild zeigt den Aufbau in Zeilen, die durch Arduino-Ausgänge aktiviert werden, und Spalten, die von den ADC-Eingängen des Arduino abgefragt werden. Vereinfacht ist das Schaltbild in der Hinsicht, dass zwischendrin die meisten Zeilen und Spalten weggelassen wurden, die aber exakt identisch aufgebaut sind.

Neopixel-Gefunkel

Um den Verdrahtungsaufwand für die Beleuchtung der 36 Kugeln möglichst klein zu halten, habe ich RGB-Leuchtdioden vom Typ WS2812 verwendet, die auch unter dem Namen Adafruit Neopixel bekannt sind und in schon vielen Projekten hier im Make-Magazin verwendet wurden. Besonders beliebt sind dabei die Neopixel-Streifen mit vielen, dicht aufeinander folgenden LEDs. Bei diesen reicht eine 5-V-Spannungsversorgung und ein ein-



Bild 5: WS2812-Neopixel-LED

ziger digitaler Ausgang eines Mikrocontrollers aus, um jede der LEDs des Streifens individuell in Farbe und Helligkeit zu programmieren. Möglich wird das dadurch, dass in jeder der Neopixel-LEDs ein eigener Mikrocontroller integriert ist. Die Daten werden dabei von einer LED zur nächsten "weitergereicht".

Leider konnte ich diese fertigen Streifen für mein Spielbrett nicht verwenden. Wer sich für das Brett aber eine andere Konstruktion überlegt, schafft es vielleicht, die handelsüblichen Streifen zu nutzen und spart sich dann viel Lötaufwand. Ich bin stattdessen auf einzelne WS2812-LEDs im Standard-5-mm-Gehäuse ausgewichen (Bild 5), die in die entsprechend gebohrten Löcher des Spielbretts gesteckt und verdrahtet werden (Bild 6). Da man jeweils den Daten-Ausgang (Dout) einer LED mit dem Daten-Eingang (Din) der nächsten verbindet, entsteht eine Verkettung, die mit den üblichen RGB-LED-Streifen technisch identisch ist. Das bedeutet: Auch in meinem Fall reicht ein einziger Pin (D10), um mithilfe der Neopixel-Bibliothek von Adafruit jede einzelne der 36 LEDs gezielt anzusteuern.

Hinweis: Alternativ sind auch die LEDs mit der Bezeichnung PL9823 einsetzbar. Die sind kompatibel zu WS2812. Allerdings sind die Farben Rot und Grün vertauscht, wenn man wie ich die Neopixel-Arduino-Bibliothek ver-



Bild 6: Testaufbau mit Neopixel-LEDs - serielle Datenleitung mit orangefarbener Litze





Bild 7: Die Maße der Platte und die Nummerierung der LEDs und Sensoren im Code

Bild 8: Das Anleimen der Kanten auf den Hirnholzseiten sorgt dafür, dass alle Seiten des Spielbretts ähnlich aussehen.

wendet. Bei meinem Spiel habe ich tatsächlich PL9823-LEDs verwendet und daher diese Vertauschung in den Code eingebaut. Dies lässt sich aber mit wenig Aufwand auf WS2812 umstellen, indem man die Zeile #define PL9823 im Code löscht (Zeile 21).

Tischlerarbeiten

In meinem Anwendungsfall sollte das Spiel wohnzimmerkompatibel aus Holz bestehen. Aber natürlich sind viele andere Ausführungen und Fertigungsweisen denkbar wie 3D-Druck, geschichtete Plexiglasplatten oder vielleicht sogar Gips oder Ton. Wenn man sich für Holz entscheidet, dann hilft es, wenn es Hartholz ist. Ich habe eine Leimholzplatte aus Akazienholz aus dem Baumarkt verwendet, die zwar erst mal hübsch aussah, aber leider in sich eine sehr unterschiedliche Holzqualität hatte. So sind einige Bohrungen etwas ungenau geworden, da das Holz stellenweise sehr weich war.

Ich habe mich für eine Grundfläche von 40 \times 40 cm entschieden (Bild 7). Wer es kleiner mag: Die 36 Kugeln mit maximal 30 mm Durchmesser lassen sich auch mühelos auf 30 \times 30 cm unterbringen.

Mit der Kreissäge habe ich einen 5 mm breiten Streifen aus der Platte geschnitten, der dann als Anleimer für die Hirnholzkanten diente. So sehen alle vier Seiten des Spielbretts gleich aus (Bild 8).

Um die Vertiefungen möglichst genau zu platzieren, habe ich zunächst die 36 5-mm-Löcher für die LEDs gebohrt. Die Vertiefungen für die Kugeln konnte ich dann mit einem Kegelsenker anhand der vorhandenen Löcher herstellen, da ich keinen passenden Kugelfräser beschaffen konnte. Dass die Vertiefungen nicht kugel-, sondern kegelförmig sind, hat keinen Einfluss auf ihre Funk-



Bild 10: Gefräste Nuten und Rahmen auf der Unterseite des Spielbretts



Bild 9: Bohren der Löcher und der Vertiefungen

tion, die Kugeln am vorgesehenen Platz zu halten.

Nach dem Bohren der Vertiefungen platzierte ich mit einem 4-mm-Bohrer dicht neben das LED-Loch in jede Vertiefung ein zweites Loch für die Reflexlichtschranke.

Ideal ist ein Grundbrett von 15 mm Dicke, da dann die LEDs genau in die mittige Bohrung passen und die Abschlussdrähte der Lichtschranken ausreichen, damit diese bündig zur Oberfläche der Vertiefungen sind (Bild 9). Liegen die Lichtschranken zu tief im Loch, erkennen sie die Kugeln nicht zuverlässig genug.

Da das von mir verwendete Brett allerdings 20 mm dick war, musste ich im Bereich der LEDs und Lichtschranken mit der Oberfräse breite Nuten fräsen. Auf jeden Fall braucht man noch einen Holzrahmen auf der Brettunterseite, der für genügend Platz für die Verdrahtung und das Mikrocontroller-Board sorgt. Ich habe mich entschieden, diesen Rahmen gegenüber dem Grundbrett deutlich kleiner zu gestalten, sodass der Rand des eigentlichen Spielbretts später scheinbar über dem Tisch schwebt (Bild 10).

Lötübungen

Ich löte eigentlich sehr gerne, weil es irgendwie etwas Meditatives hat. Allerdings muss man hier 36 × (4 LED-Pins + 4 Lichtschrankenpins) = 288 Lötstellen bewältigen. Diese sollten auch wirklich gut sein, weil man sonst am Ende sehr lange mit der Fehlersuche beschäftigt ist. Hinzu kommt, dass es gilt, jede Menge Schaltlitze abzulängen und abzuisolieren. Bei mir ging dabei die meditative Entspannung dann doch etwas verloren.

Da ich noch größere Reste von kupferkaschierter Hartfaserplatte hatte, entschied ich mich, auf meiner CNC-Fräse einfache Streifenplatinen zu fertigen und so die meisten Kabel einzusparen. Aber keine Bange: Auch ohne CNC-Fräse kann man sich die Lötarbeit deutlich einfacher machen, indem man längliche Stücke von Lochstreifenplatinen verwendet, deren Leiterbahnen man an den passenden Stellen durch Wegkratzen der Kupferschicht unterbricht (Bild 11). Diese gibt es in einer maximalen Größe von 50 × 10 cm aus Hartpapier zu kaufen. Selbst für ein Spielbrett in meiner Größe braucht man für die einzelnen Längen nur eine Länge von 30 cm. Die Lücken setzt man dann analog zur Vorlage fürs Fräsen.

Isolationsfräsen

Man braucht für solche einfachen Streifenplatinen noch nicht mal ein Platinenlayout- oder ECAD-Programm zu bemühen, da es ja nur darum geht, lange Leiterbahnen und ein paar Unterbrechungen zu fräsen. Ich habe daher die Fräsbahnen mit Inkscape gezeichnet und



Bild 11: Alternative zum Fräsen: Lochstreifenplatinen, deren lange Leiterbahnen man durch Wegkratzen der Kupferschicht trennen kann



Bild 12: Isolationsfräsen der Streifenplatinen auf meiner Eigenbau-CNC-Fräse



Bild 13: Fertige gefräste Platine für die Verbindungen der LEDs und Lichtschranken. Diese Platine habe ich später zwecks einfacherer Montage mit der Dekupiersäge längs getrennt und die weiteren Platinen gleich getrennt gefräst.



Bild 14: Fertig verdrahtetes Spielbrett

mit Estlcam zu G-Code verarbeitet. Gefräst habe ich das dann auf meiner Selbstbau-CNC-Fräse (Bild 12). Das geht aber sicherlich auch auf den einschlägigen China-Fräsen, die für kleines Geld zu bekommen sind.

Die Inkscape-Dateien, passend für mein Brett mit 40 × 40 cm stehen unter dem Link zum Artikel zum Download zur Verfügung. Die Bohrlöcher habe ich mit dem Frässtichel nur markiert und dann auf der Ständerbohrmaschine manuell gebohrt. Der Verdrahtungsaufwand hat sich durch die Platinen erheblich reduziert (Bild 13). Die 150-Ohm-Widerstände zwischen der Anode der Reflexlichtschranke (Pin 2) und dem Kollektoranschluss der Transistoren habe ich SMD-Bauform verbaut, weil man sie prima über die gefräste Lücke löten kann. Das funktioniert aber natürlich auch mit bedrahteten Widerständen.

Elektronik komplettieren

Neben den Platinen für die Verbindung der LEDs und Lichtschranken habe ich den Arduino Nano auf eine Lochrasterplatine gesetzt, um dort die sechs BD170-Transistoren unterzubringen, die die LEDs der Reflexlichtschranken ein- und ausschalten (Bild 14). Übrigens sind die Anforderungen an den Mikrocontroller so gering, dass man statt des Nano auch irgendein anderes Board verwenden kann. Es muss nur je sieben Ausgänge und analoge (ADC-) Eingänge haben.

Die Stromversorgung hingegen muss für den Worst Case ausgelegt sein: Wenn 36 RGB-LEDs weiß mit voller Helligkeit leuchten und 6 LEDs in den Lichtschranken aktiv sind. Das schafft der USB-Ausgang des Computers nicht unbedingt. Daher sollte man auch zur Programmierung des Arduinos das externe Steckernetzteil anschließen. Ich habe dafür ein Netzteil (5 V/2 A) mit Hohlstecker verwendet. Die passende Buchse findet im Holzrahmen unterhalb des Bretts ihren Platz (Bild 15).

In den Rahmen habe ich noch einen inneren Rahmen mit um 3 mm kleinerem Querschnitt eingebaut. Dieser dient als Auflage für den Holzdeckel, unter dem die Elektronik verschwindet (Bild 16).

Die Spiellogik

Das Programm besteht aus zwei Arduino-Sketches. In dem Sketch DeoMemory.ino befindet sich das Hauptprogramm, während die Lichteffekte der Übersichtlichkeit halber in eine zweite Datei LightEffects.ino ausgelagert wurden.

Da eine zeilenweise Erklärung des Source-Codes diesen Artikel bei Weitem sprengen würde, geheich nur kurz auf die Funktionsweise ein – ich hoffe, die Kommentare im Code verhelfen dann zu tieferem Verständnis. Den Code gibt es zum Download unter dem Link in der Kurzinfo.

Wie bei allen interaktiven Spielen gibt es einen Game-Loop, der in diesem Fall alle zehn Millisekunden auf eine Änderung lauert. Dazu aktiviert er zeilenweise die Sende-LEDs der Reflexlichtschranken und ermittelt das Empfangssignal jeder der sechs Spalten.

Die analogen Pins des Arduinos liefern einen Wert zwischen 0 und 1024 und die Werte der vorangegangenen Messung sind als Vergleichswerte für alle 36 Positionen im Array



Bild 15: Buchse für externes 5-V-Steckernetzteil

Bild 16: Ein Deckel aus 3mm Pappelsperrholz deckt die Elektronik ab.



Bild 17: Hier kommen die Kugeln meines Spiels her: mit 29-mm-Standardkugel



Bild 18: Sprungbälle ("Flummies") eignen sich als Deokugelersatz.



Bild 19: Glaskugeln als Spielsteine funktionieren auch und sehen schick aus.

1astSenseArray gespeichert. Ändert sich der aktuell gemessene Wert um mehr als 20 gegenüber dem zuletzt gemessenen an dieser Position (was ca. 2 % vom Maximalwert 1024 entspricht), interpretiert die Software das als eine neu gelegte Kugel an dieser Stelle. Im Kopf des Programms ist dieser Schwellwert mit #define THRESHOLD 20 festgelegt und kann dort gegebenenfalls geändert werden. Im Folgenden gibt es ein paar Plausibilitätsprüfungen und dann wird der zu diesem Feld gehörende Lichteffekt abgespielt.

Das Programm unterscheidet dabei genau, ob die gelegte Kugel die erste oder die zweite des Spielers in dieser Runde war. Denn wenn die zweite neue Kugel gelegt wurde, checkt das Programm nach Abspielen des zweiten Lichteffekts, ob die zur Identifizierung dienenden Nummern der beiden hintereinander erfolgten Lichteffekte gleich waren und somit ein Paar gefunden wurde. In diesem Fall werden beide Kugeln in der Farbe des entsprechenden Spielers dauerhaft beleuchtet, wenn nicht, dann bleiben sie dunkel und müssen wieder vom Spielbrett genommen werden. Die Spielernummer wechselt dann zum anderen Spieler.

Auch wenn die Kugeln auf dem Brett Matrix-artig angeordnet sind, habe ich mich für die interne Verwaltung durchweg für eindimensionale Arrays entschieden, die die Informationen über die aktuellen Sensorwerte, Lichteffekte oder die bereits gelegten Kugeln speichern. So ist die Parameterübergabe einfacher und es bleibt kompatibel mit der Verschaltung der RGB-LEDs, die ja auch einfach linear aufgereiht sind.

Zufälliger Zufall

Zufallsgeneratoren in Computern werden üblicherweise durch rückgekoppelte Schieberegister realisiert, die dafür sorgen, dass jeweils der nächste Zahlenwert unvorhersagbar ist. So entsteht eine scheinbar zufällige Zahlenfolge. Ist aber der erste Wert dieses Generators gleich, dann läuft die Zahlensequenz danach auch immer gleich ab. In einem PC mit Betriebssystem kann man diesen "Seed-Wert" aus der Uhrzeit, den Sekunden seit Start des Rechners oder anderen veränderlichen Werten ableiten und erhält immer wieder andere Sequenzen von Zufallszahlen bei jedem Lauf des Programms.

Der Arduino startet aber immer sehr deterministisch gleich, sodass die zufällig verteilten Lichteffekte bei jedem Neustart immer gleich waren, wie ich feststellte.

Gelöst habe ich das Problem dadurch, dass ich den siebten, noch freien, ADC-Eingang lese und als "Seed-Wert" verwende. Das Rauschen des offenen Eingangs erzeugt so einen wirklich zufälligen Anfangswert, wodurch auch die folgenden Werte immer wieder anders sind und jedes Spiel mit einer zufälligen Neuverteilung der Blinkmusterpaare beginnt.

Blick in die Glaskugel

Wie eingangs erwähnt, haben die Spielsteine meines Spiels vorher mal als Kugeln in Deorollern ihren Dienst getan. Es fiel uns dabei auf, dass der Durchmesser von 29 mm sehr gebräuchlich ist und auch bei unterschiedlichen Marken durchweg Verwendung findet (Bild 17). Wenn man den Kunststoffrand des Deorollers mit einer Wasserpumpenzange greift, lässt er sich leicht abziehen und man kann die Kugel anschließend von innen herausdrücken.

Da sicherlich nicht jeder erst 36 Roll-Deos verbrauchen möchte, bis er dieses Projekt beginnen kann oder sowieso lieber Sprühdeos benutzt, habe ich mich auch nach anderen Kugeln umgesehen, die genauso funktionieren und die man fertig kaufen kann.

Im Spielwarenhandel gibt es transparente Flummies – Handelsbezeichnung "Sprungbälle" – die ähnlich schöne Leuchteigenschaften haben. Meistens haben sie kleine eingegossene Partikel, die das Licht streuen.

Im Hobby- und Dekoversand bin ich dann noch auf Glaskugeln gestoßen. Entscheidet man sich für die einfache Industriequalität und nicht die handgeschliffene Version, dann sind 45 Kugeln mit 25 mm Durchmesser für ca. 12 Euro zu bekommen.

Glaskugeln in Verbindung mit dem dunklen Holz erzeugen ein noch edleres Erscheinungsbild des Spiels (Bild 19).

Ausblick

Hat man die Lötarbeit erfolgreich absolviert und die Kugelerkennung in der Software zuverlässig ans Laufen bekommen, ist der Rest nur noch Software. Das bedeutet: Mit neuem Code kann man nun auch andere Spielideen realisieren. Jede Kugel kann schließlich individuell beleuchtet werden und das Brett "weiß" immer, wo Kugeln liegen.

Zu einem Vier-Gewinnt-Spiel ist es so zum Beispiel nur ein kleiner Schritt. Vielleicht gelingt aber mit dieser Hardware auch die Programmierung von Spielen, bei denen Spielsteine übersprungen werden müssen, wie zum Beispiel Solitär.

Man könnte eventuell auch mehrere Spiele parallel in den Arduino laden – der Code zum Download belegt nur rund ein Viertel des Speichers auf dem Board und die bereits vorliegenden Lichteffekte lassen sich zudem in anderen Spielen weiter nutzen. Die Auswahl des jeweils gewünschten Spiels könnte man durch das Legen von Kugeln in bestimmte Vertiefungen beim Einschalten realisieren – das wäre ein minimalistisches User-Interface ohne Knöpfe und Schalter.

Viel Spaß beim Nachbauen und Weiterentwickeln – die Redaktion freut sich wie immer über Mails mit neuen Spielideen und deren Umsetzungen. —pek

DIY-Röhren-Vorverstärker

Wenn man sich nach einem dreikanaligen Röhren-Vorverstärker für seine E-Gitarre sehnt, aber keinen im Handel findet, bleibt nur der Selbstbau. Dafür muss man aber ordentlich Zeit, Geduld und fundierte Elektronikkenntnisse mitbringen. Ein Bausatz wie der SLO-Highgain hilft, um nicht bei null anzufangen. Dennoch gibt es einiges zu beachten, wie dieser Erfahrungsbericht zeigt.

15 R37 (1111

12

von Detlef Grell

S chon lange hege ich den Wunsch, mir einen "echten" dreikanaligen Röhren-Vorverstärker zuzulegen, der pro Kanal einen eigenen Gain- und Volume-Regler und vor allem eine amtliche Klangregelung für Höhen, Mitten und Bässe hat. Es gibt diese legendären Soundmaschinen aber nirgends zu kaufen. Also habe ich mir selbst einen Röhren-Vorverstärker gebaut – und zwar den SLO-Highgain von Andy Hoppe. So ein Projekt ist nicht trivial und mit einer Betriebsspannung von 350 Volt auch gefährlich. Zudem sollte man neben Geduld ein paar Messgeräte parat haben und nicht zuletzt das Wissen, wie man damit eine systematische Fehlersuche betreibt.

Den SLO-Highgain gibt es derzeit als Bausatz bei "Das Musikding" für 279 Euro – allerdings ohne die vier benötigten ECC83-Röhren, dafür aber mit einem passenden 19-Zoll-Gehäuse. Das ist zwar von hoher Qualität, aber keineswegs vorgebohrt oder gar beschriftet. Links zur Projektseite und dem Produkt finden Sie am Ende des Artikels. Da es sich drucktechnisch nicht anders lösen lässt, sind in diesem Artikel nur Ausschnitte der Schaltpläne abgedruckt, auf die der Text eingeht. Laden Sie sich für das bessere Verständnis die Anleitung auf der Projektseite herunter.

Wenn es gut werden soll

Meine Empfehlung vorweg: Zeit einplanen, viel Zeit – wenn es denn gut werden soll. Ich habe mir insgesamt etwa vier Monate Zeit gelassen, aber natürlich nicht Vollzeit jeden Tag und auch nicht jedes Wochenende dran gesessen. Ersatzbauteile oder Alternativen zu beschaffen, kostet auch immer ein paar Tage. Allein fürs Gehäuse sind einige Wochen draufgegangen – dazu gleich mehr.

Mir war von Anfang an klar, dass sich dieses Projekt nicht "mal eben" in ein oder zwei Wochen stemmen lässt. Das wird allerdings nicht jedem sofort einleuchten – doch auch darum geht es in diesem Artikel. Ich habe seit meinem 15. Lebensjahr Audio- und Computer-Projekte zusammengebaut, Platinen selbst layoutet und geätzt, danach die Bauteile eingelötet, mit Netzteilen und Gehäusen verdrahtet. Aber ich hatte bisher noch kein Röhrenprojekt dabei. Als jemand, der auch schon gelegentlich 230 Volt abbekommen hat, hatte ich jedenfalls den notwendigen Respekt vor Röhren-Betriebsspannungen deutlich über 300 Volt.

Wer sich an den SLO-Highgain dransetzt, der sollte schon einige ernsthafte Selbstbau-Gerätschaften zum Laufen gebracht haben. Er sollte die Schaltung im Detail verstehen, wenn er später herausfinden will, warum sie auf seinem Werktisch nicht funktioniert. Fehlersuche und -behebung erfordern nicht nur Messgeräte wie Multimeter, Signalgenerator und Scope, sondern auch das Wissen, wie man damit umgeht. Vielleicht kennen Sie ja einen

Ergänzende Artikel auf heise+

Dieser Baubericht ist der dritte Teil einer Artikelreihe. Wer mehr über die Hintergründe von Röhrenverstärkern und die Besonderheiten des SLO-HighgainBausatzes erfahren möchte, findet am Ende des Artikels Links zu den jeweiligen heise+-Artikeln. Als Make-Abonnent können Sie diese kostenfrei lesen.

versierten Elektroniker, der Ihnen zur Hand gehen kann – dann stehen die Chancen schon deutlich besser. Und an die Leser, denen ich jetzt den Mund wässrig gemacht habe, weil sie die astronomischen Preise für Soldano-Verstärker kennen, ganz klar die Ansage: Dieses Projekt hier funktioniert nicht wie Malen nach Zahlen. Es ist bedeutend komplexer als den Tonabnehmer in der Gitarre auszutauschen.

Während die eigentliche Röhrenschaltung relativ simpel ist, muss bei diesem Projekt zuvor die digitale Drumherum-Logik funktionieren, ehe man das erste Sound-Erlebnis genießen kann. Wo da Tücken auftauchen können, lesen Sie im weiteren Verlauf.

Das Bohren und das Sägen

Bereits das Gehäuse hat es in sich. Ich besitze zwar eine recht hochwertige Ständerbohr-

maschine, aber ich gestehe: Ich bin selbst damit kein Präzisionshandwerker. Auch das Anbringen von filigranen Beschriftungen ist durchaus herausfordernd. Unter den Bauplänen gibt es ordentliche Vorlagen, die man etwa auf "Waterslide Decal Paper" drucken kann. Hinter "Waterslide Decal" verbergen sich die Abziehbilder, die Sie vielleicht noch aus Ihrer Kindheit kennen: Ins Wasser legen und vorsichtig auf das Zielgebiet schieben, etwa auf Ostereier.

Aber: Wer das noch nie gemacht hat, sollte das vorher gründlich üben. Schon die Vorbereitung ist nicht trivial. Das Papier sollte entweder speziell für Tinten- oder für Laserdrucker ausgewählt werden. Meine ersten Versuche, mit Tintendruck ohne hernach dicke Schichten Klarlack aufzutragen, gingen auch prompt schief. Der Druck löste sich einfach auf.



Bild 1: Die eigentlich schwarz lackierte Frontplatte hat ein Fachbetrieb gebohrt und darauf eine beschriftete Alu-Folie geklebt.



Bild 2: Die Rückseite ist nach demselben Muster bearbeitet worden wie die Frontplatte, allerdings musste der Dienstleister ein wenig mit meinen nicht ganz exakten Bohrungen kämpfen.



Bild 3: Bestückungsseite: Es gilt, einen Haufen Bauteile fein säuberlich auf der Platine zu platzieren und zu verlöten.

Ein alternativer Versuch mit transparenter Klebefolie war nicht wischfest, und die A4-Folien lassen sich auch nur abschnittsweise bedrucken und aufbringen – selbst A3-Folie deckt keine 19-Zoll-Frontplatte komplett ab. So bleiben folglich immer Folienabschnitte erkennbar. Abgesehen davon: Ein weißer Tintendruck für eine schwarze Frontplatte ist ein interessantes Problem.

Da heftiges Lackieren – nicht zuletzt auch der Frontplatte – einem unserer Familienmitglieder wegen der Lösungsmittel-Dämpfe sehr schlecht bekommt, habe ich dann die Reißleine gezogen und einen Fachbetrieb beauftragt. Der hat auch gleich das punktgenaue Bohren übernommen und auf die Frontplatte (Bild 1) und Rückwand (Bild 2) eine bedruckte Alufolie (champagnerfarben) geklebt. Das war nicht billig, aber dafür sieht es wirklich professionell aus.

Damit waren aber noch nicht alle Mechanikprobleme gelöst, denn die mitgelieferte Kaltgerätebuchse passte leider nicht zum Ausschnitt in den Vorlagen. Nachdem ich die Potis eingeschraubt hatte, fehlten ja noch die Knöpfe. Die passenden gab es gerade nicht beim Bausatzanbieter "Das Musikding", und blöderweise habe ich den Hinweis übersehen, dass die Potis eine 1/4-Zoll-Achse (6,35 mm und nicht die üblichen 6,00 mm) mitbringen. Glücklicherweise besitze ich einen 1/4-ZollBohrer und konnte alle 16 Knöpfe passend machen.

Glänzende Lötstellen

Löten kann ich eindeutig besser als Gehäuse bearbeiten. Ich lehne mich jetzt mal etwas aus dem Fenster, aber RoHS-Lötzinn, also das bleifreie, und Handlöten sind für mich nicht vereinbar. Fürs Auge produziert man mit bleifreiem Lot eigentlich nur kalte Lötstellen, also keine, die glänzen. So kann man nie wirklich erkennen, ob Lötstellen gut oder kalt sind. Meine Empfehlung: Stannol HS10 S-Sn60Pb39Cu1 (60 Prozent Zinn, 39 Prozent Blei und 1 Prozent Kupfer).

Leider ist der Verkauf bleihaltigen Lötzinns durch Händler an Privatkunden in Deutschland nicht mehr erlaubt, sodass man jemanden mit Gewerbeschein einspannen muss. Ich warne auch gleich vor dem Versuch, bleihaltiges Lötzinn von chinesischen Händlern zu ordern. Meine drei Versuche bescherten mir völlig untaugliches Zeug, das nur sehr widerwillig zurückgenommen wurde.

Komplett-Bausatz ohne Röhren

Der Bausatz ist sehr liebevoll zusammengestellt, auch die Bauteilqualität ist hoch. Ich kann zwar mit einem Blick die Widerstands-Codes der 5- und 10-Prozenter mit den vier Ringen lesen – aber die fünf Ringe der 1-Prozenter mit dem selten weit genug abgesetzten braunen Toleranz-Ring sind bei mir nie richtig im Langzeitgedächtnis angekommen. Freundlicherweise sind aber alle Bauteile an ihren Gurten von Hand mit dem Wert beschriftet – was spätestens bei nur mit der Lupe lesbaren Werten auf Kondensatoren sehr willkommen ist.

Trotzdem: Wenn man unsicher ist, sollte man ein Multimeter zur Hand nehmen, das Widerstand und Kapazität messen kann. So schön eine doppelseitige Platine (Bild 3) mit Durchkontaktierungen auch ist, hier Bauteile wieder auslöten zu müssen, macht die Platine nicht schöner.

Wenn's doch mal nötig wird, dann kann man die Bauteilbeinchen abkneifen und die Rest-Pins einzeln auslöten. Das verbliebene Lötzinn aus Durchkontaktierungen zu entfernen ist eigentlich nur mit einer Entlötpumpe hinzubekommen – wenn die Platine von hoher Qualität ist. Die hier mitgelieferte Platine ist vergleichsweise robust, sodass beim Einsatz der Pumpe nicht gleich die Lötaugen abreißen sollten. Im Regelfall benutze ich jedoch Entlötlitze und lasse das Lötzinn in den Löchern.

Show Stopper

Der Bausatz ist sehr umfangreich, und folglich passieren Fehler. Darauf sollte man unbedingt

gefasst sein. Als ich das erste Mal Strom auf die Niederspannungsseite gegeben habe, die 12 Volt für die Röhrenheizung und 5 Volt für das Digitale erzeugt, passierte – nichts.

Relativ schnell ließ sich erkennen, dass der mitgelieferte LM2990-12 ein Stabi-IC für minus 12 Volt war – leider war mir der Typ nicht so geläufig wie die 78er/79er. Folglich musste ich ihn wieder auslöten und durch den korrekten 2940 ersetzen (Bild 4). Immerhin entstand dadurch kein Folgeschaden.

Nächster Stopper: Die 1,4-A-Sicherung im 12-Volt-Zweig verstarb unmittelbar nach dem Einschalten. Eine 2,5-A-Sicherung verrichtet





Bild 5: Genau so soll man die BC517 einlöten, aber leider waren bei den mitgelieferten Transistoren Kollektor und Emitter vertauscht.



Bild 6: Der schwarze Draht liegt am Pin 1 des Steckers, aber leider nicht am Pin 1, wie es der Aufdruck vorsieht.



Bild 7: Zum Glück kann man die Steckerführung leicht abziehen und umgedreht wieder aufsetzen.



Bild 8: Oder man nimmt einen Stecker mit der korrekten Konfektionierung.



seither den Dienst. Danach leuchtete immerhin die Betriebsleuchte, aber nicht eine der LEDs, die den benutzten Kanal anzeigen. Folglich blieb mir nichts anderes übrig, als mich in den Teil der Schaltung hineinzuknien, der mich bei diesem Projekt am allerwenigsten interessierte, nämlich der ganze Digitalkram.

Sämtliche Kanalumschaltungen steuert ein ATmega8-Prozessor, der fertig programmiert mitgeliefert wird. Zwischen den Kanälen wird über 12 Reed-Relais umgeschaltet. Als Treiber für die Relais und die LEDs dient pro Kanal ein BC517, ein sehr gängiger Darlington-Transistor. Außerdem kümmert sich der Prozessor noch um die MIDI-Logik (nur zur Kanalumschaltung) und er sorgt dabei auch noch für die Stummschaltung des Audio-Ausgangs während der Umschaltungen.

Jetzt habe ich alle Bauteile und alle Lötstellen noch mal überprüft – und ich hatte tatsächlich einen einzigen Pin nicht angelötet. Aber mein Problem war nicht gelöst. Ich konnte nur noch die Funktion des ATmega8 anzweifeln. Von Klaus Brunner, dem Inhaber des Musikdings, kam ohne Lamento ein Ersatzprozessor, zudem fehlte auch noch ein Chip für die Stummschaltung. Doch auch daran lag es nicht.

BC517 falsch herum ...

Ich habe schätzungsweise zehn verschiedene Datenblätter zum Pinout des BC517 befragt,



Bild 10: Ordentlich Gain aufgedreht: Die Dachschräge zeigt an, dass die Tiefen angehoben sind, die Spitzen kommen über die Höhen.

die alle bestätigten, dass die Transistoren das klassische E-B-C-Pinout wie der bekannte BC107 haben – was letztlich auch der Bestückungsaufdruck zu verstehen gab. Nachdem ich mich endlich durchgerungen hatte, die BC517 doch mal auszulöten und durchzumessen, waren bestimmt drei Wochen vergangen. Und Treffer: Diese besondere Charge war tatsächlich falsch herum belegt, also Emitter und Kollektor vertauscht (Bild 5).

Die Kanalumschaltung funktionierte trotzdem noch nicht, weil der Lieferant der Platinenstecker diese sozusagen doppelt falsch konfektioniert hat. Diese Stecker haben eine aufgedruckte 1, an denen der schwarze Draht liegt. Genauso soll es sein – wie im Forum noch mal bekräftigt wird. Da der Bestückungsaufdruck vorgibt, wo die Führung der Stecker liegen soll, liegt Pin 1 dann aber doch an der falschen Stelle. Das ist bei allen Positionen kein Problem, wo nicht die Stiftnummer, sondern die Signalbezeichnung aus dem Schaltplan direkt an die Pins auf dem Bestückungsplan gedruckt wurde. Aber immer dort, wo nur 1-2-3 angegeben ist, waren bei mir stets 1 und 3 vertauscht (Bild 6). Das betrifft CH_SW, FROM_ VOL und FROM GAIN.

... und die Stecker auch

Wer die Stecker schon falsch eingelötet hat, kann vorsichtig die Steckerhalterung aus Kunststoff nach oben abziehen und gedreht wieder aufschieben, muss also nichts umlöten (Bild 7 und 8). Zu diesem Zeitpunkt hatte ich die zugehörige Hilfsplatine (Output-Board) für den zweiten Eingang, den eigentlichen Ausgang und die Send-Return-Loop bislang nicht angeschlossen, sondern einfach das Out-Signal auf eine Klinkenbuchse geschickt.

Jetzt endlich Röhren einsetzen und etwas hören! Wieder nix. Die Buchse für den Fußschalter fehlte noch. Daher wirkte der Default-Zustand "Fußschalter angeschlossen", weil FS_ENABLED nicht auf Masse lag (Bild 9).

Endlich war was zu hören. Das war aber noch nicht so doll, und da ich bis dahin nicht entdeckt hatte, dass die Stecker für die Gainund Volume-Potis falsch angeschlossen waren, funktionierte nur der mittlere Kanal – bis ich am Lead-Kanal den Clean-Channel zum Leben erwecken konnte und umgekehrt.

Dann war der Sound schon irgendwie da, aber eher so "naja". Vor allem der Bassregler wirkte weniger auf die Bässe als die Mittenregler, alles sehr schräg. Bingo: Ich hatte tatsächlich alle Bass-Potis falsch angeschlossen. Soweit funktionierte nach der Korrektur alles und auch der Sound war endlich so wie erhofft.

You won't get the swing

Jetzt passierte das, was ich eigentlich von vornherein erwartet hatte: Einer der Kanäle

INNOVATION

Report



Bild 11: Sieht kaputt aus, ist aber nicht ungewöhnlich für eine Verzerrung nahe Fuzz.



Bild 12: Aber so soll es auf keinen Fall aussehen, denn hier sitzen kräftige Hochfrequenzschwingungen auf den Amplitudenspitzen.

(Rhythm/Crunch, also der mittlere) oszillierte. Das ist bei hochohmigen Schaltungen fast schon normal – die fangen leicht mal Signale am Ende einer Verstärkerstufe ein und koppeln sie auf den Eingang zurück. Das passiert meist ab einer bestimmten Gain-Einstellung. Hier war es besonders schlimm, wenn der Bright-Schalter, also die Höhenanhebung, nicht aktiv war – erwartet hätte ich es in der anderen Stellung.

Hier noch ein Tipp dazu, wie man feststellt, dass ein Verstärker schwingt: Wenn sich der Klang ab bestimmten Poti-Stellungen krass verändert, hört man meist sehr eklige, hochfrequente Verzerrungen. Wer ein Scope hat, kann dann sehen, wie das Nutzsignal von hässlichen Störsignalen überlagert wird (Bild 12).

Abhilfe bringt bei diesem Projekt mit etwas Glück bereits eine geänderte Anordnung der Leitungen zwischen Platine und Frontplatte, also den Potis. Ich habe zur Sicherheit besonders lange Gain-Leitungen geschirmt und das Problem war vom Tisch (Bild 13).

Änderungen, die ...

Wir reden hier von einem Projekt, dessen Entwickler die Früchte seiner Arbeit kostenlos an die Community weitergereicht hat. Ihm fehlendes Versions-Management vorzuwerfen, ist eigentlich frech. Aber über die Jahre wurde





OXOCARD



AIR Cartridge

Einstecken und die Connect wird zum Raumsensor





Überwache deine Umgebung mit der Tiefenbildkamera (8×8-Pixel)



Browserbasierte Scripting-Umgebung mit Debugging! Enthält über 100 fertige Beispiele.







Jetzt im heise shop bestellen

In der Schweiz bei Brack www.oxocard.ch



Bild 13: Auch in meinem Aufbau kam es zu ungewollten Oszillationen. Um auf Nummer Sicher zu gehen, habe ich alle Leitungen abgeschirmt, die rückkopplungsgefährdet waren.

das doch zu einem Problem, weil einige kleine Änderungen sich nicht in der Dokumentation finden beziehungsweise sich alte und neue Angaben direkt widersprechen.

Das Projekt wurde 2008 veröffentlicht und intensiv im Forum betreut, auch von Forenmitgliedern untereinander. Aber es sind 40 Seiten mit je 25 nicht immer kurzen Einträgen, die es durchzuarbeiten gilt, wenn man alle Änderungen mitbekommen will. Es fing schon damit an, dass das Forum scheinbar nicht gepflegt wurde – wenn man auf musikding.de versuchte, über das Menü das Forum aufzurufen, erhielt man eine Fehlermeldung, gelangte aber trotzdem dorthin, wenn man eine der Topic-Leisten anklickte. Mittlerweile ist das repariert.

Innerhalb des Forums existieren allerdings noch viele inzwischen unbrauchbare Links,

leider auch die zu wichtigen Dokus. Die URL andyszeugs.de führte zwischendurch zu einem Getränkemarkt. Inzwischen hat Andy das konsistent überarbeitet.

... dem Fortschritt dienen

Hier kurz einige Korrekturen, die man sonst nur übers Forum herausfindet. Als Muting-Chips, also die Opto-Relais, die Schaltknackser bei der Kanalumschaltung unterdrücken, waren AQY214 vorgesehen, also Relais, die aktiv schließen. Inzwischen werden im Bausatz AQY410 (Öffner) geliefert und die aktuelle ATmega-Programmierung ist darauf ausgelegt.

Auf kleine Stolperfallen traf ich noch, als ich die Hilfsplatine in Betrieb nehmen wollte. Die Buchsen in der Gehäuserückwand sind mit Send-Return beschriftet. Üblicherweise deklariert man auf diese Art die Möglichkeit, über "Send" das Signal an externe Effektgeräte zu übergeben, um es von deren Ausgang in "Return" wieder einzuspielen. Letztlich ist die Bezeichnung korrekt, die Funktion aber "andersherum" gemeint: Darüber lässt sich der Vorverstärker seinerseits in die Effekt-Loop eines anderen Verstärkers einschleifen und kann diesem auf die Art drei zusätzliche Pre-Amp-Kanäle spendieren.

Daher gibt es auch einen virtuellen vierten Kanal: Wenn man die Kanäle durchtackert, leuchtet in der vierten Position keine Kanal-LED und der Vorverstärker wird komplett überbrückt, ist also im echten Bypass-Betrieb. So kann man einerseits den SLO-Highgain abschalten. Steckt er andererseits in einer Effekt-Loop, kommt wieder die eigentliche Vorstufe des benutzten Verstärkers zur Geltung.

Ebenfalls Kopfzerbrechen machte mir der Anschluss der Input-Buchse an der Frontplatte. Das lag an widersprüchlichen Aussagen in der Bausatzbeschreibung und im Forum. Da der SLO in ein 19-Zoll-Rack-Gehäuse eingebaut wird, wird der Eingang an der Rückseite benutzt. Schließt man aber eine Gitarre an die Front-Buchse an, dann klemmt diese die Rückseiten-Buchse ab und übernimmt an der Front. Ferner ist der separate 1,8-kOhm-Widerstand, den man in die externe Mute-Leitung einfügen soll, bereits im aktuellen Layout der Hilfsplatine berücksichtigt, ebenso der zweite Muting-Chip.

Ansonsten hat das Projekt viel Spaß gemacht, ich habe wieder einiges gelernt und besitze jetzt einen amtlichen dreikanaligen Röhrenvorverstärker. —*akf*

$\left[\right]$	Alles zum Artikel	
	make-magazin.de/xq7k	



Bild 14: Blick auf den fertigen Röhren-Vorverstärker. Der Aufbau mit den vier Röhren ist fast zu schön, um ihn in einem Gehäuse zu verstecken.

Mastering GitOps 2024

Stabil deployen, sicher automatisieren

16. & 18. April 2024 • online

GitOps für Devs, DevOps und Ops

Wollen Sie Ihre **Softwareanwendungen ohne Handarbeit deployen** oder Ihre **Infrastruktur sicher automatisieren?** Dann empfiehlt sich die deklarative Herangehensweise von GitOps. Als "Single Source of Truth" erlaubt ein Git-Repository den kontiunierlichen Abgleich von Ist- und Zielzustand für Deployment-Pipelines, Anwendungen und Infrastruktur.

 16. April – Dev Day: Im Mittelpunkt steht das kontinuierliche Deployment von Anwendungen

• 18. April – Platform Day: Die Rolle von GitOps im Bereich Platform Engineering und für die Automatisierung von IT-Infrastruktur

Jetzt Tickets sichern!

mastering-gitops.de

Veranstalter



dpunkt.verlag

betterCode()

API 2024

Design, Entwicklung und Management von APIs

14. und 16. Mai 2024 • online

In der Softwareentwicklung kommt man nicht um APIs herum: Ob Private oder Public APIs – schwierige Entscheidungen sind zu fällen. Welche **Architekturparadigmen**, welche Verfahren, welche **Protokolle** nutze ich, wann welche besser nicht? Wie gewährleiste ich **Qualität**, **Kompatibilität** und **Sicherheit**?

Folgende Themen erwarten Sie:

- Fortgeschrittene Entwurfstechniken
- Entwicklungstools
- Versionierung
- Testen

Workshops am 4. und 6. Juni

Veranstalter



API Management

- Data Contracts
- API Sprawl und Great Unbundling

Jetzt Frühbuchertickets ^{sichern!}

api.bettercode.eu

dpunkt.verlag

g

IKEA-Leuchtwürfel-Hack

Sobald die LED-Leuchte FREKVENS Geräusche wahrnimmt, spielt sie Animationen ab – mehr kann sie aber leider nicht. Tauscht man den Original-Controller jedoch gegen einen ESP8266 aus, lässt sich die LED-Matrix des Würfels intelligent steuern, etwa als autonome Uhr oder MQTT-gesteuertes Lauflicht mit Smarthome-Anbindung.

von Dr. Ing. Armin Zink





n der Make 6/23 habe ich beschrieben. wie leicht man die IKEA-Wandleuchte OBEGRÄNSAD aufwerten kann, indem man die ursprüngliche Steuerplatine gegen einen WEMOS D1 Mini Pro tauscht - und die LED-Matrix so in eine praktische Uhr verwandelt. Da es im Sortiment des schwedischen Möbelhauses mit FREKVENS eine weitere LED-Leuchte gibt, die technisch sehr ähnlich, aber deutlich kleiner ist, musste ich ausprobieren, ob sich diese ebenfalls hacken lässt. In diesem Artikel erkläre ich, wie es geht.

Ausgangssituation

Der kleine LED-Würfel kommt mit einem eingebauten Netzteil in einem niedlichen $10 \times 10 \times 10$ cm großen Gehäuse daher und ist im Vergleich zum großen Bruder mit 40 Euro ein echtes Schnäppchen. Die eingebaute Elektronik reagiert mithilfe eines verbauten Mikrofons auf Geräusche und zeigt auf 256 weißen LEDs unterschiedliche Muster an, die sich bewegen, sobald man etwa Musik abspielt oder spricht – angelehnt an eine Lichtorgel aus den 80er Jahren. Man kann aus einer Reihe an Mustern wählen, dazu gehören Balkendiagramme, ein Sternenhimmel oder ein Pseudo-Oszillogramm.

Kurzinfo

» IKEA-LED-Leuchte in ein smartes Gerät verwandeln » 16 × 16 LED-Matrix mit WEMOS D1 Mini Pro steuern » Uhrzeit und MQTT-Daten anzeigen



- » IKEA FREKVENS
- » WEMOS D1 Mini Pro
- » Litze, verschiedene Farben
- » Optional für die automatische Helligkeitseinstellung: LDR-Sensor
- und 100-kΩ-Widerstand mit 0,25 W

Werkzeug

» Lötkolben Schraubendreher Kleine Finger

Mehr zum Thema

- » Dr. Ing. Armin Zink, IKEA-Matrix gehackt, Make 6/23, S. 64
- Dr. Ing Armin Zink, Airsign, Make 4/22, S. 52 Ákos Fodor, Adafruit GFX Library: Malen mit Zahlen, Make 2/23, S. 64
- Peter Tschulik, LED-Laufschrift mit ESP32, Make 4/21, S. 8



Uhrzeit auf der Matrix

Die LED-Anzeige der Leuchte besteht aus 16 × 16 weißen LEDs, die auf einer guadratischen Fläche verteilt sind. Damit reicht die Breite der Matrix nicht aus, um vier Ziffern samt Trennzeichen nebeneinander darzustellen, wie man auf Bild 1 erkennen kann. Aus diesem Grund habe ich die Anzeige auf zwei Zeilen verteilt: Die obere stellt die Stunden und die untere die Minuten dar (Bild 2).

Der benötigte Schriftsatz besteht aus einer monochromen Bitmap-Datei, die ich mithilfe des Online-Tools image2cpp in Byte-Arrays umgewandelt habe (Bild 3). Wenn Sie ihn anpassen oder selbst einen Schriftsatz erstellen wollen, finden Sie eine Anleitung in der Make 6/23, die den Ablauf Schritt für Schritt erklärt (siehe Link in der Kurzinfo). Die Bitmap mit dem Beispiel-Zeichensatz liegt im GitHub-Repository des Projekts. Natürlich kann man auch die Byte-Arrays direkt innerhalb des Codes ändern, in einem Grafikprogramm wie MSPaint ist das allerdings etwas einfacher.



Bild 1: Die Matrix ist mit 16 LEDs zu schmal, um die vollständige Uhrzeit mit Trennzeichen abzubilden.





Bild 3: Mit image2cpp kann man Bitmaps in Byte-Arrays umwandeln.

3

Bild 2: Wenn man die Stunden und Minuten auf zwei Zeilen verteilt, passt es.



Bild 4: Die LED-Leuchte besteht aus einem Kunststoffgehäuse und vielen Metall-Muttern, damit man mehrere Würfel miteinander verbinden kann.



Bild 5: Wenn man eine Schraube in eine der Muttern schraubt (links unten), lässt sich das Gehäuse leichter öffnen.

Die Musterauswahl lässt sich aber nicht erweitern, wodurch der Nutzwert ebenfalls vergleichsweise niedrig ausfällt. Hier hat IKEA meiner Meinung nach Potenzial verschenkt, um etwas Sinnvolleres aus dem Produkt zu machen. Aber dafür gibt es ja uns Maker. Also ran ans Werk: Bauen wir uns eine smarte Leuchte!

Umbau mit Potenzial

Wie auch beim Vorgänger-Projekt kommt beim FREKVENS-Mod ein WEMOS D1 Mini Pro als Austausch-Controller zum Einsatz. Nach dem Umbau zur Y-Clock, wie ich sie genannt habe, reagiert die kleine Leuchte zwar nicht mehr auf Geräusche, verbindet sich aber per WLAN mit dem Internet, ruft die aktuelle Uhrzeit über einen NTP-Server ab und gibt sie auf der LED-Matrix aus. Über einen der Taster auf der Geräterückseite lässt sich zudem die Helligkeit der LEDs dimmen. Wer das automatisieren möchte, kann zusätzlich einen LDR-Helligkeitssensor verbauen.

Außerdem gibt es eine Firmware-Alternative, die es ermöglicht, die Leuchte per MQTT ins Smarthome einzubinden und Sensordaten oder Informationen von Internetdiensten auszugeben. Diese ruft der Mikrocontroller in festgelegten Zeitabständen ab und lässt bis zu zwei Werte gleichzeitig über den Bildschirm laufen.

Verwenden Sie den hier gezeigten Hardware-Hack gern als Ausgangspunkt für eigene kreative Ideen. Ist der Mikrocontroller erst mal eingebaut, lässt sich die LED-Matrix frei bespielen.

Gerät aufschrauben

Die Demontage und vor allem der erneute Zusammenbau der Leuchte ist etwas kompliziert. Das Gerät besteht einerseits aus vielen Einzelteilen, die gesteckt, geschraubt und teilweise zusätzlich mit Silikon verklebt sind, sodass man beim Auseinandernehmen genau aufpassen sollte, wie mal alles zusammengehört hat (Bild 4).

Um an das Innere zu gelangen, löst man als Erstes die vier Schrauben auf der Rückseite des Geräts, bis sie an die schwarzen Muttern stoßen. Danach kann man die Rückwand entfernen, die eigentlich das Netzteil ist. Da sie mit Silikon verklebt ist, benötigt man etwas Kraft. Wenn man eine der mitgelieferten Schrauben in eine Mutter dreht, ist es leichter, mit einer Schere oder Zange an der Rückwand zu ziehen (Bild 5).

Danach löst man die vier Kunststoff-Halter aus den Ecken, mit denen die verklebten acht Muttern im Gehäuse gehalten werden. Führen Sie dazu einen Schraubendreher wie in Bild 6 hinein und hebeln sie das Bauteil mit mehre-



Bild 6: Mit dem Schraubendreher lassen sich die Mutter-Halter leicht heraushebeln.



Bild 7: Zum Hebeln muss man den Schraubendreher zwischen Ecke und Strebe schieben.



Bild 8: Die Löcher an der Gehäuseseite helfen beim Hebeln, sind aber recht empfindlich.

ren Bewegungen heraus. Drücken Sie anschließend die acht Muttern ins Gehäuse und nehmen Sie sie heraus.

Am Boden befinden sich vier Schrauben. die Sie entfernen müssen, um den Innenrahmen anzuheben. Dieser sitzt ziemlich fest im Gehäuse. Um ihn zu lösen, müssen Sie den Schraubendreher zwischen die Ecken und Streben schieben und hebeln (Bild 7). Sobald sich der Rahmen etwas herausbewegt hat, schieben Sie ihn durch die seitlichen Löcher weiter nach oben (Bild 8). Aber Vorsicht: Der Kunststoff des Gehäuses ist ziemlich weich. Wenn Sie die Löcher nicht beschädigen wollen, legen sie etwas Stoff dazwischen.

Abschließend müssen Sie jeweils zwei von vier Muttern an der Gehäuse-Vorderkante, also am LED-Display, herausdrücken und einsammeln. Dann können Sie die zwei schwarzen Halteelemente entnehmen. Nachdem die Halter entfernt sind, kann man die Platine samt des LED-Panels herausnehmen. In manchen der von mir geöffneten FREKVENS war auch die LED-Platine mit Klebstoff befestigt.

Sie können auch die Diffusor-Scheibe entnehmen. Ohne sie ist die Anzeige klarer und streut weniger auf die benachbarten Pixel, auch wenn die innenliegenden LEDs nicht



Bild 9: Die Original-Elektronik der LED-Leuchte ist mit sieben Lötstellen an der weißen Platine befestigt.

mehr aus jedem Winkel gleich gut erkennbar sind. Probieren Sie es am besten nachher beim Zusammenbau aus. Letztlich bleibt es Ihnen überlassen, ob ihnen die Uhr mit oder ohne Diffusor besser gefällt.

Mikrocontroller vorbereiten

Im fertigen Aufbau versorgt die Platine der LED-Matrix den WEMOS mit 5 V. Allerdings laufen die LEDs mit 4 V. Verlötet man den



2. Auflage · 2022 · 382 Seiten · 34,90 € ISBN 978-3-86490-866-8



2023 · 236 Seiten · 24,90 € ISBN 978-3-86490-914-6



3. Auflage · 2022 · 366 Seiten · 36,90 € ISBN 978-3-86490-867-5



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



2023 · 264 Seiten · 29,90 € ISBN 978-3-86490-951-1



2. Auflage 2022 · 198 Seiten · 19,95 € ISBN 978-3-86490-859-0



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

Noch mehr LEGO®, Make & Elektronik: dpunkt.de





Bild 10: Verlöten Sie den WEMOS mit der Leuchte und den Tastern.

WEMOS mit dem Board der Leuchte und schließt dann ein USB-Kabel an das Mikrocontroller-Board an, kann es passieren, dass 5 V durch den WEMOS in die LED-Steuerung gelangen und diese beschädigen. Um das zu vermeiden, sollte Sie den Mikrocontroller flashen, bevor Sie ihn einbauen. Wenn Sie ihn

LED-Würfel verkabeln

FREKVENS	WEMOS D1	Farbe im Bild
VCC	5V	Rot
EN	D5	Rosa
DI	D6	Gelb
CLK	D7	Lila
CLA	D8	Blau
GND	GND	Schwarz
Roter Taster	D4	Weiß
Gelber Taster	D2	Schwarz
Beide Taster	GND	Rot

später neu programmieren möchten, sollten Sie sicherheitshalber das VCC-Kabel ablöten oder einen Schalter anbringen. Für häufigere Anpassungen empfiehlt sich auch, ein USB-Kabel aus dem Gehäuse herauszuführen.

Laden Sie sich zum Flashen zunächst die Arduino IDE und die Sketches aus dem Git-Hub-Repository des Projekts herunter (siehe Link in der Kurzinfo). Wie man ESP-Boards in die Entwicklungsumgebung einbindet, erklären wir Ihnen in einem Online-Artikel, der frei zugänglich und ebenfalls verlinkt ist.

Öffnen Sie danach den Arduino-Sketch mit oder ohne MQTT-Unterstützung. Anders als beim Vorgänger-Projekt verwendet die Y-Clock den WiFiManager von tzapu, um eine Verbindung zu einem WLAN herzustellen. Daher muss man seine WLAN-Zugangsdaten nicht im Sketch eintragen, sondern erledigt das später im laufenden Betrieb. Der WEMOS startet dafür nach dem Einschalten einen Access-Point, über den man die Verbindung zum



Bild 11: Für die Variante, die automatisch die LED-Helligkeit regelt, benötigt man noch den LDR-Sensor und einen 100-kΩ-Widerstand.

WLAN herstellen kann. Für diese Funktion müssen Sie in der Arduino IDE noch die Bibliothek WiFiManager ergänzen. Wenn Sie MQTT nutzen wollen, benötigen Sie außerdem die Bibliothek PubSubClient von Nick O'Leary und die Daten Ihres MQTT-Brokers, um sie im Sketch zu ergänzen (mehr dazu am Ende des Artikels).

Für die Variante mit dem Helligkeitssensor müssen Sie manuell noch ein paar Zeilen zum Sketch hinzufügen. Definieren Sie zunächst mit

#define P_LDR A0

den Port, an dem der Sensor angeschlossen wird. Fügen Sie danach im void setup() die Zeile

pinmode(P_LDR, INPUT);

hinzu, um den Sensor als Eingangssignal festzulegen. Die Helligkeit für die LEDs stellen Sie dann nicht mehr über den roten Taster ein, sondern berechnen sie wie folgt:

brightness = map(analogRead(P_LDR), 1024, 0, 240, 254);

Der Befehl map() wandelt das Analogsignal zwischen 0 und 1024 auf den Bereich von 240 bis 254 um. Geschickterweise fügt man diese Zeile im void loop() in die if-Schleife ein, die minütlich die Zeit aktualisiert.

Stellen Sie schließlich als Board "LOLIN (WEMOS) D1 mini Pro" ein, wählen Sie den Port aus, an dem der Mikrocontroller angeschlossen ist und flashen Sie den Sketch.

Elektronik austauschen

Um das Mikrocontroller-Board mit der LED-Matrix zu verbinden, muss man zunächst die grüne Platine auslöten, auf der sich der Original-Controller befindet (Bild 9). Lösen Sie dafür den einzelnen Mikrofon-Pin am oberen Ende der Platine, heben sie diese danach leicht an und entfernen Sie die übrigen sechs Lötstellen. Trennen Sie zum Schluss die drei Taster-Kabel vom Original-Controller.

Verlöten Sie dann den WEMOS mit der weißen Platine wie im Bild 10 und der Tabelle angegeben. Auf dem IKEA-Board sind nur die Pins für VCC und GND aufgedruckt. Die Reihenfolge der Pins EN, DI, CLK und CLA ist jedoch identisch mit der fritzing-Grafik. Verwenden Sie für die Verbindungen am besten Kabel mit einer Länge von etwa 10 cm. Sind sie kürzer, kann man später nur noch sehr schwer ein USB-Kabel einstecken.

Die Farben der Taster-Kabel haben leider keine sinnvolle Zuordnung: Es gibt ein rotes Kabel, das von beiden Tastern zu GND führt. Den Port D4 verwenden wir für den roten Taster mit dem weißen Kabel, der die Helligkeit regelt und verbinden D2 über das schwarze Kabel mit dem gelben Taster.



Bild 12: Wo sich die Oberseite befindet, erkennt man mithilfe der Mikrofon-Aussparung an der Front.

Variante mit LDR-Sensor

Wenn Sie eine automatische Anpassung der Helligkeit wünschen, löten Sie zusätzlich den LDR-Sensor mit langen Kabeln entsprechend dem Schaltplan (Bild 11) an die Pins A0 und 3V3 des WEMOS, und einen 100-k Ω -Widerstand zwischen GND und A0.

Bohren Sie nach dem Löten ein kleines Loch in eine der Vertiefungen an der Oberseite des Gehäuses (Bild 12). Ich empfehle ein Loch in der zweiten Reihe von vorne. Die Oberseite des Gehäuses erkennen Sie daran, dass das Mikrofon-Loch (von vorn gesehen) sich auf der linken Hälfte befindet. Fixieren Sie danach den LDR-Sensor mit einem Klebeband auf der Innenseite des Gehäuses.

Das Gerät wieder zusammensetzen

Der Zusammenbau ist eine echte Herausforderung, die etwas Fingerspitzengefühl erfordert. Sie möchten das nicht öfter machen als notwendig. Testen Sie deshalb unbedingt, ob der Aufbau funktioniert, bevor sie den Würfel zusammensetzen (Bild 14). Achten Sie dabei darauf, das Gerät **nicht** über das USB-Kabel des Mikrocontrollers mit Strom zu versorgen. Die zu hohe Spannung könnte die LED-Treiber schädigen.

Wenn die Buchstaben AP auf der LED-Matrix erscheinen, ist das ein guter Indikator dafür, dass alles geklappt hat. Setzen Sie die Leuchte danach schrittweise zusammen:

 Das Display passt nur in einer Ausrichtung in das Gehäuse. Achten sie auf das kleine Loch für das Mikrofon.

Vereinfachte Installation

Da ich das Projekt über diesen Artikel hinaus weiterentwickle und um das Flashen des Mikrocontrollers zu vereinfachen, biete ich neben der hier gezeigten Version der Y-Clock-Firmware auch kompilierte Varianten über einen Web-Installer an. Diese sind zwar nicht Open Source, enthalten aber schon die benötigten Bibliotheken und weitere Features (Bild 13). Dazu gehören verschiedene Schriften und Einstellungsmöglichkeiten über ein Web-Interface sowie eine Update-Funktion, mit der man immer auf dem neusten Stand meiner Firmware ist.

Um den WEMOS D1 Mini Pro über den Web-Installer zu flashen, muss man das Board lediglich per USB mit einem Computer verbinden, die Website mit dem Google Chrome Browser aufrufen (siehe Link in Kurzinfo) und die passende Firmware auswählen.

> Bild 13: Der Web-Installer vereinfacht nicht nur das Flashen. Über ihn erhält man auch den aktuellsten Stand meiner Firmware und deren Features.





Bild 14: Die Buchstaben AP zeigen an, dass der WEMOS richtig mit der LED-Matrix verbunden ist und den Access Point gestartet hat.



Bild 15: Die seitlichen Kunststoffbrücken muss man parallel zu den länglichen ICs einsetzen.



 Setzen Sie anschließend nacheinander die Kunststoff-Brücken parallel zu den länglichen IC-Chips auf der weißen Platine ein (Bild 15). Fädeln Sie danach die beiden Muttern ein. Das geht am besten, indem man jeweils eine herunterrutschen lässt und mit einem Schraubendreher durch die Gehäuseseite drückt. Tipp: Sie können die Muttern auch mit den beiliegenden Schrauben fixieren.

- Schieben Sie danach den Innenrahmen ein und drücken Sie ihn gut herunter. Befestigen Sie ihn mit den vier Schrauben.
- Wenn Sie die LDR-Version verwenden, prüfen Sie den Sitz des LDR-Sensors.
- Setzen Sie danach die acht Muttern in die Ecken des Gehäuses und fixieren Sie diese mit den Kunststoff-Haltern.
- Verschrauben Sie die Rückwand mit dem Logo nach oben.

Y-Clock in Betrieb nehmen

Da die Y-Clock die Uhrzeit aus dem Internet bezieht, muss sie sich mit Ihrem WLAN-Router verbinden können. Dazu startet der WEMOS den eingangs erwähnten Access Point, der wie das Gerät ebenfalls Y-Clock heißt und zeigt auf der LED-Matrix die Buchstaben AP an. Navigieren Sie in Ihrem Smartphone zu den WLAN-Einstellungen und verbinden Sie sich mit dem Access Point. Daraufhin erscheint die Benutzeroberfläche vom WiFiManager (Bild 16). Wählen Sie Configure-WiFi und danach Ihr Netzwerk aus und geben Sie das Kennwort für Ihr WLAN ein. Sobald sich die Y-Clock erfolgreich mit Ihrem Netzwerk verbinden konnte, zeigt die Leuchte kurz ein OK und danach die aktuelle Uhrzeit.

Das tut sie ab jetzt jedes Mal, wenn man sie anschaltet. Befindet sich die Y-Clock aber an einem Ort, an dem Sie das gespeicherte WLAN nicht findet, bleibt sie im Access-Point-Modus, sodass man sie erneut konfigurieren kann. Die Zugangsdaten bleiben auf dem ESP in einem gesonderten Speicher. Das hat den Vorteil, dass man den WiFiManager nicht



Bild 16: Über den WiFiManager kann man sich komfortabel mit neuen WLANs verbinden.

Bild 17: Vereinfachte Darstellung des Publisher-Subscriber-Prinzips von MQTT.

jedes Mal neu konfigurieren muss, wenn man etwas im Sketch ändert und den Mikrocontroller flasht. Wenn Sie dennoch mal den Speicher vollständig löschen möchten, wählen Sie in der Arduino IDE im Menü Werkzeuge die Option "Erase Flash/Sketch + WiFi Settings" oder "All Flash Contents" aus, bevor Sie den Sketch übertragen.

Y-Clock als MQTT-Client

Natürlich kann ein vernetztes Display auch mehr anzeigen als nur die Uhrzeit. Wie wäre es zum Beispiel mit der aktuellen Raumtemperatur oder einem Hinweis, der daran erinnert, ein geöffnetes Fenster zu schließen?

Damit IoT-Geräte solche Informationen schnell untereinander austauschen können, bietet sich das MOTT-Protokoll an. Dieses System arbeitet nach einem Publisher-Subscriber-Prinzip (kurz pubsub), d.h. es besteht aus einem oder mehreren Clients, die Nachrichten mit einem Schlagwort (Topic) an einen zentralen Server (Broker) senden (publish), aber auch empfangen können, sofern sie das Topic abonniert haben (subscribe). Die Clients kommunizieren also nicht direkt miteinander und müssen sich auch nicht kennen, sondern erhalten ihre Informationen entsprechend ihres Abos durch den Broker (Bild 17), der beispielsweise lokal auf einem Raspberry Pi (z. B. Mosquitto) laufen kann.

Mit dem Online-Dienst HiveMQ lässt sich sogar mit MQTT experimentieren, ohne sich dafür registrieren zu müssen. Öffnen Sie dazu den Sketch YClock-WiFiManager-MQTT.ino, der die LED-Leuchte in einen MQTT-Client verwandelt und denken Sie sich zwei Namen für Topics aus:

```
#define MQTTSRVR "broker.hivemq.com"
#define MQTTPORT "1883"
#define MQTTTOPIC1 "Topic for Line1"
#define MQTTTOPIC2 "Topic for Line2"
```

Flashen Sie den Sketch danach auf den Mikrocontroller. Sobald die Y-Clock eingeschaltet und mit dem Internet verbunden ist, nimmt sie mit dem Befehl client.setServer() den Kontakt zum Test-Broker von HiveMO auf und abonniertmitclient.subscribe() die beiden Topics - und wartet auf eingehende Nachrichten. Um welche zu senden, öffnen Sie als Nächstes den Web-Socket-Client von HiveMQ in Ihrem Browser (siehe Link in der Kurzinfo). Klicken Sie danach auf Connect, tragen Sie einen der ausgedachten Topics in das Topic-Feld ein und drücken Sie auf Publish. Je nachdem, ob Sie das ersten oder zweiten Topic gewählt haben, erscheint Ihre Nachricht daraufhin mithilfe der Callback-Funktion client.setCallback() unmittelbar in der oberen oder unteren Matrix-Hälfte der LED-Leuchte.

Code für die Laufschrift

```
void p_scrollText(int8_t xs, uint8_t ys, String s) {
    Serial.println("Scrolltext:");
    Serial.println(s);
    for (int i=0; i<s.length(); i++) {</pre>
         p_scrollChar(xs, ys, s[i]);
}
void p_scrollChar(int8_t xs, uint8_t ys,char ch) {
    uint8_t d;
    p scroll():
    p scan(1):
    delay(SCROLL_SPEED);
    for (uint8_t x=0; x<6; x++) {
    p_scroll(); // wait to scan here</pre>
         p_scroll();
         d = pgm_read_byte_near(System6x7+(ch-32)*6+x);
         if ((d& 1) == 1)
                               p_drawPixel(xs, 0+ys, 0xFF);
         else p_drawPixel(xs, 0+ys, 0x00);
if ((d& 2) == 2) p_drawPixel(xs, 1+ys, 0xFF);
                else
         if ((d\& 4) == 4)
         else
                p_drawPixel(xs, 2+ys, 0x00);
                               p_drawPixel(xs, 3+ys, 0xFF);
         if ((d\& 8) == 8)
                p_drawPixel(xs, 3+ys, 0x00);
         else
                               p_drawPixel(xs, 4+ys, 0xFF);
         if ((d&16) == 16)
         else
                p_drawPixel(xs, 4+ys, 0x00);
                               p_drawPixel(xs, 5+ys, 0xFF);
         if ((d\&32) == 32)
         else p_drawPixel(xs, 5+ys, 0x00);
         if ((d\&64) == 64)
                               p_drawPixel(xs, 6+ys, 0xFF);
         else p_drawPixel(xs, 6+ys, 0x00);
         p_scan(1);
         delay(SCROLL_SPEED);
    }
    p_scroll();
    p_scan(1);
    delay(SCROLL_SPEED);
}
void p_scroll() {
     for (uint8_t x=1; x<16; x++)</pre>
    for (uint8_t y=0; y<16; y++)
    p_drawPixel(x-1,y,p_getPixel(x, y));</pre>
    for (uint8_t y=0; y<16; y++)
    p_drawPixel(15, y, 0);</pre>
}
uint8_t p_getPixel(int8_t x, int8_t y) {
    #ifdef H_FREKVENS
         if (x > 7) { y += 0x10; x &= 0x07; }
return p_buf[ (y*8+x)] ;
    #endif
}
```

Falls Sie das mit einem lokalen Broker ausprobieren möchten, geben Sie für MQTTSRVR die IP-Adresse Ihres Computers an, auf dem der Broker läuft. Im GitHub-Repository des Projekts finden Sie auch einen Sketch, der eine Verbindung mit Benutzername und Passwort unterstützt. Abgesehen davon ist die Verbindung über den Port 1883 standardmäßig unverschlüsselt.

Die Laufschrift

Damit die Y-Clock auch längere MQTT-Nachrichten ausgeben kann, habe ich ein paar Funktionen ergänzt, mit denen sich Texte als Laufschrift anzeigen lassen (siehe Listing). Sobald der WEMOS ein Topic über MQTT erhalten hat, initiiert die Callback-Funktion mgtt_callback() die Ausgabe des Textes mithilfe von p_scrollText(int8_t xs, uint8_t ys, String s). Diese zerteilt den String in einzelne Zeichen und gibt diese als Variable s sowie ihre Koordinaten als xs, ys nacheinander an die Funktion p_scrollChar() weiter. Mit ihrer Hilfe taucht ein Buchstabe am rechten Bildschirmrand spaltenweise auf und wird mit p_scroll() Stück für Stück in der Geschwindigkeit SCROLL_SPEED nach links geschoben. Darauf folgt eine leere Spalte und dann der nächste Buchstabe. –akf

Make: Online

Beliebt auf heise 🕂





Aber grundsätzlich machen **Röhrenverstärker** für die Gitarre den Sound, den der Gitarrist hören will....

DIY-Röhrenverstärker, Teil 1: Was den Klang bei

19.02.2024 🕩 | Make

Röhrenverstärkern auszeichnet

Diodenlaser mit 40 Watt Leistung: Creality Falcon 2 im Test

Auch der P2 erzeugt einen breiteren Schnitt als der Diodenlaser....

01.03.2024 🕩 Make



Universelle Infrarot-Fernbedienung mit WLAN bauen

Dann beginnt die Suche nach der **Fernbedienung** des Blu-Ray-Players. Weil die **Fernbedienung** nicht beleuchtet ist und alle Lampen aus sind, ist auch der Pause-Knopf schwer zu finden. Wenn es endlich geschafft ist, renne ich zur Tür.... 14.02.2024 Make

So kommen Sie als Make-Abonnent an Artikel hinter der Paywall von heise+: https://heise.de/-7363373

Große CAD-Serie rund um 3D-Druck, Möbel, Lasern und CNC auf YouTube!

Es gibt eine neue Serie auf unserem YouTube-Kanal! Ein CAD-Kurs, der in vier Teilen den CAD-Workflow vom ersten Strich bis zum fertigen 3D-Modell, Schnittlisten, STL-Dateien für den 3D-Druck, SVG-Dateien für Lasercutter und bemaßte Detailzeichnungen behandelt. Im ersten Teil zeichnet Joey in SketchUp den Korpus einer Kommode. Um die Kommode mit Kumiko-Schiebetüren auszustatten, zeichnet er im zweiten Teil die Türen und das Untergestell. Teil 3 beschäftigt sich mit der Installation von OpenCutlist und der Erstellung einer Schnittliste. Die Videoreihe endet mit der Erstellung von bemaßten Zeichnungen und

Weitere Folgen der Serie:







UR MAKER

Bleib informiert:

www.make-magazin.de

@makemagazinde





Bluesky: @makemagazin.de



WhatsApp Channel: Make Magazin Deutschland



GitHub: MakeMagazinDE



Threads: @makemagazinde

in

LinkedIn: linkedin.com/ company/maker-media-gmbh/



Kontakt zur Redaktion Leserbriefe und Meinungen an: heise.de/make/kontakt/



-dus

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



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



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

DER GROSSE CAD-KURS (1/4)

dem Export der 2D/3D-Daten, so dass alles für

www.youtube.com/@MakeMagazinDE

den Bau der Kommode bereit ist.

S2N Storage Server Network

Die neue heise-Konferenz für Storage, Server, Network

Die S2N ist der Branchentreff für IT-Administratoren, Systemarchitekten, CTOs und alle IT-Spezialisten. Die zweitägige Konferenz informiert über die neuesten Trends und Technologien in den Bereichen **Netzwerkarchitektur und** -management, Serverarchitekturen und moderne Speicherlösungen.

Die **S2N** ist hervorgegangen aus der storage2day, der heise-Konferenz für Speichernetze und Datenmanagement, und erweitert deren Themenspektrum um Server und Netzwerke.

Call for Proposals läuft bis zum 19. April:

Haben Sie IT-Know-how, das Sie mit Fachkolleginnen und -kollegen teilen wollen? Möchten Sie von eigenen Erfahrungen berichten und sie mit einem Fachpublikum diskutieren? **Wir freuen uns über Ihre Themeneinreichungen!**



Premiere in Hamburg am 9. & 10. Oktober





s2n-heise.de

© Copyright by Maker Media GmbH.

Thermostate clever hacken

Bestehende, nicht Smarthome-fähige Thermostate werden zu maßgeschneiderten Wächtern über die Raumtemperatur. Außen schick und mit Touchkeys, innen ein ESP32, der keine Smarthome-Wünsche offen lässt. Obendrein spart es zweifach Strom.

von Alexander Neubauer



34 | Make: 2/2024

© Copyright by Maker Media GmbH.

n unserem recht neuen und energetisch modernen Haus von 2019 sind alle Räume mit Fußbodenheizung ausgestattet und im Garten arbeitet eine Luftwärmepumpe, um alles zu heizen. Herkömmliche Raumthermostate steuern die Fußbodenheizung in jedem Raum individuell und sorgen so überall für die Wohlfühltemperatur.

Als bekennender Smarthome-Enthusiast war es nur eine Frage der Zeit, bis ich diese wenig smarten Thermostate nicht mehr ertragen konnte. Die naheliegende Lösung wäre eigentlich, die vorhandenen Thermostate durch halbwegs preiswerte, smarte Varianten aus Fernost zu ersetzen. Eigentlich. Denn als eingefleischter Maker stellte ich mir natürlich sofort die Frage, wie ich es schaffen könnte, die bestehenden Thermostate so zu modifizieren, dass ich sie via MQTT ins Smarthome-Zeitalter befördern könnte.

Smarte Thermostate

Unsere Wärmepumpe füllt im Keller einen sogenannten Schichtpufferspeicher. Dieser ist so ausgelegt, dass nur das Wasser im oberen Bereich auf die benötigte Trinkwassertemperatur (in meinem Fall 50 °C) erwärmt wird. Im mittleren Bereich herrscht im Normalbetrieb eine Temperatur von 35 °C für die Fußbodenheizung und im unteren Bereich ist das Wasser kalt (etwa 20 °C).

Seit wir eine Fotovoltaik-Dachanlage unser Eigen nennen, habe ich die Heizung um eine Funktion ergänzt: Sobald während der Heizperiode die Sonne genug Strom für die Wärmepumpe liefert, schalte ich diese an, um den Puffer auch unten zu erwärmen. Diese Wärme

Kurzinfo

- » Billige (aber schicke) Raumthermostate hacken und mit eigener Elektronik versehen
- » ESP32 für die Integration ins Smarthome, LCD-Ansteuerung, Touch-Sensor-Steuerung
- » Speicherung von Solarenergie im Warmwasserspeicher und im Fußboden

Checkliste	Mehr zum Thema	
Zeitaufwand: Nachbau etwa ein Wochenende	 » Mark Liebrand, Heizungslüfter, Make 1/23, S. 24 » Klaus Peters, Heizungslogger, Make 1/23, S. 18 » Jan Kipping, Smarte Bad-Sparheizung mit ESP, Make 6/22, S. 20 	
Kosten: ca. 30 Euro pro Thermostat		
Löten: SMD-Löten		
Programmieren: Visual Studio Code mit PlatformIO		
Sonstiges: Fertigung von Platinen bei einem Dienstleister		
Hochspannung: Arbeiten in unmittelbarer Nähe von 230 Volt Netzspannung	Alles zum Artikel im Web unter make-magazin.de/xj86	

nutze ich dann nachts zum Heizen. An einem sonnigen Tag muss ich die Wärmepumpe allerdings wieder abschalten, sobald der Pufferspeicher aufgeheizt ist (die Wärmepumpe erreicht nur 55 °C). Eine interessante Idee eines Freundes brachte mich dazu, den Estrich im Haus als erweiterten Wärmespeicher zu nutzen: In Räumen, in denen wir eine leicht höhere Temperatur tolerieren, schalten smarte Thermostate die Fußbodenheizung ein. So



Abbildung 1: Originale Leistungseinheit



Abbildung 2: Originale Displayeinheit



Abbildung 3: Einzelteile der Displayeinheit: (von links) Selbst entworfene Steuerplatine, LED-Hintergrundbeleuchtung, LC-Display und Touch-Key-Scheibe

Programmiermodus

Um den ESP32 auf seiner Platine in den Programmiermodus zu bringen, muss man den Reset-Knopf drücken, während der Boot-Knopf gedrückt gehalten wird. Nachdem man den Reset-Knopf losgelassen hat, kann auch der Boot-Knopf losgelassen werden. Der ESP32 befindet sich nun im Programmiermodus und die neue Software kann hochgeladen werden. Nach erfolgreichem Upload muss man den Reset-Knopf erneut drücken, um das neue Programm zu starten. Auf handelsüblichen Entwicklungsplatinen übernehmen diese Aufgabe in der Regel zwei Transistoren, die vom USB-to-Serial-IC angesteuert werden.



Abbildung 4: Design der Leiterplatte in EasyEDA

speichern wir einfach und effektiv diese kostenlose Energie. Die in den Fußböden der Räume gespeicherte Wärme wird nachts abgegeben, die Wärme im Heizungssystem wird hier nicht benötigt und kommt so den anderen beheizten Räumen zugute und unterstützt den Pufferspeicher.

Zielsetzung und Machbarkeit

Aufgrund der Komplexität des Projekts kann und will ich keine Schritt-für-Schritt-Anleitung zum Nachbau geben. Vielmehr möchte ich das Gesamtprojekt im Überblick vorstellen und auf wesentliche Aspekte näher eingehen, in der Hoffnung, andere Maker zu ähnlichen Projekten zu ermutigen:

- eigene Ansteuerung eines Segment-LCD-Displays,
- Nutzung der Touch-Eingänge des ESP32,
- Multithreading auf dem Mikrocontroller, um eine jederzeit reaktionsfähige Benutzerschnittstelle zu gewährleisten.

Für die "Smartification" der Thermostate habe ich mir folgende Ziele gesetzt:

- Ansprechende Optik, d. h. weitestgehende Beibehaltung des Originalgehäuses, vor allem für den guten Wife Acceptance Factor (WAF),
- Beibehaltung der Touch-Bedienbarkeit des Originals,
- Integration in ein beliebiges Smarthome-System, hier über MQTT im ioBroker,
- niedriger Energieverbrauch, möglichst nicht höher als der des Originalthermostats.

Was folgte, war erst einmal ein vorsichtiger Blick in ein überzähliges Thermostat. Was ich dort vorfand, stimmte mich für mein Vorhaben recht positiv. Die Thermostate (Typ: C07:H3) sind modular aufgebaut. Sie bestehen aus einer Display- und einer Leistungseinheit, die untereinander mit einem 8-poligen Flachbandkabel verbunden sind (Abbildung 1 und 2).

Die Leistungseinheit beinhaltet dabei die Anschlussklemmen für den Fußboden-Tempe-
raturfühler, die 230 Volt Versorgung und den Ventilantrieb im Heizverteilerkasten. Außerdem sind in ihr das Relais zum Schalten der Ventilantriebe sowie zwei Spannungswandler von 230 Volt zunächst auf 12 Volt und dann weiter auf 5 Volt untergebracht.

Die Anzeigeeinheit (Abbildung 3) ist ebenfalls modular aufgebaut. Sie besteht aus einer Steuerplatine, einer LED-Displaybeleuchtung, einer LCD-Segmentanzeige und einer Glasscheibe, in die die Touch-Sensoren integriert sind. Platine, LED und Display lassen sich durch Entlöten leicht trennen, der Touch-Sensor ist sogar nur gesteckt. Die Hauptaufgabe wird also sein, das Display mit eigener Hardware anzusteuern und die Touch-Sensoren auszulesen. Alle anderen Funktionen des Thermostats erschienen mir zu diesem Zeitpunkt relativ einfach realisierbar.

Entwurf der Steuerplatine

Als Mikrocontroller habe ich mich schnell für den ESP32 entschieden. Er kann von Haus aus an vielen Pins die Touch-Signale erkennen, über WLAN und MQTT mit meinem Smarthome kommunizieren, ist günstig, gut verfügbar und hat geeignete Schlafmodi zum Energiesparen. Bei den engen Platzverhältnissen im Thermostat wurde mir aber schnell klar, dass ich mit den üblichen ESP32-Entwicklungsplatinen an dieser Stelle nicht zum Ziel kommen würde. Also musste eine eigene Steuerplatine (Abbildung 4) her, die alle notwendigen Komponenten beherbergt. Folgende Funktionen sollten abgedeckt werden:

- Ansteuerung der LED-Hintergrundbeleuchtung,
- Ansteuerung der Relais zum Schalten der Ventilantriebe,
- Ansteuerung der LC-Segmentanzeige,
- Auslesen der Sensortasten,
- Auslesen des Bodentemperatur-Sensors,
- Erfassung der Raumtemperatur und Luftfeuchtigkeit mit dem Sensor DHT20,
- Montagemöglichkeit für einen DS18B20-Sensor (siehe unten),
- Spannungsversorgung des ESP32 mit 3,3 V

Meine Steuerplatine verbinde ich mit dem originalen 8-poligen Flachbandkabel über einen SMD-Wannenstecker mit der Leistungseinheit. Diese gibt folgende Signale/Spannungen an die Steuerplatine weiter:

- 5 Volt,
- Masse,
- Anschluss Bodentemperatur-Fühler (anderer Anschluss liegt auf Masse),
- Anschluss Relais (auf Masse geschaltet),

– 12 Volt (über selbst eingelötete Brücke).

Die restlichen drei Pins sind nicht belegt. Die +12 Volt werden im Originalaufbau nicht bereitgestellt. Da der ESP32 aber gerade während der Datenübertragung zu viel Strom vom originalen 5 Volt DC-DC Wandler benötigt, habe ich den Spannungsregler auf meiner Platine mit 12 Volt versorgt. Da das Relais aufgrund des Aufbaus des Netzteils mit einem Transistor gegen Masse geschaltet werden muss, gehe ich bei der Displaybeleuchtung analog vor. Wenn ein Transistor als Schalter zwischen Last und Masse eingesetzt wird, sollte immer ein NPN-Transistor verwendet werden. Zwischen Versorgungsspannung und Last werden dagegen PNP-Transistoren verwendet. Bei der Auswahl des Transistortyps (BCW 65C) orientiere ich mich einfach an der Originalplatine.

Für die Ansteuerung der LC-Segmentanzeige habe ich einen Universal-LCD-IC (PCF8576C) der Firma NXP gewählt. Dieser verfügt über ausreichend Backplane- und Segmentanschlüsse, benötigt keine externe Beschaltung und kann über I²C angesteuert werden.

Die übrigen Bauteile sind überschaubar. Ein Low-Dropout-Regulator AMS1117-3.3 übernimmt die Spannungsversorgung des ESP32 mit 3,3 Volt. Dieser zeichnet sich durch minimale externe Beschaltung aus. Sie besteht lediglich aus zwei 22µF-Kondensatoren. Zur Messung von Temperatur und Feuchte ist ein DHT20-Sensor an Bord. Man kann auch den potenziell genaueren DS18B20-Sensor einbauen. Bei Tests konnte ich jedoch keine nennenswerten Unterschiede zwischen den beiden Sensortypen feststellen. Ansonsten sind nur noch ein paar Vor- und Pull-up-Widerstände, zwei Kondensatoren am Reset- und Boot-Pin des ESP32 und eine Diode in Reihe mit der LED-Hintergrundbeleuchtung verbaut. Diese Diode schützt die LED im Falle einer versehentlichen Verpolung, da LEDs nur eine geringe Sperrspannung vertragen.

Da sich vor allem das Display von den originalen Steuerplatinen kaum ohne Zerstörung der Platine auslöten lässt, habe ich die Buchse für das Flachbandkabel des Touchsensors und den Wannenstecker für die Verbindung zum Netzteil von der originalen Platine recycelt.

Zu Diagnosezwecken und um den ESP32 auch nach dem Auflöten auf meine Steuerplatine programmieren zu können, habe ich die benötigten Pins (Masse, +3,3 Volt, +5 Volt, +12 Volt, TXD, RXD, Reset und Boot) auf Pads herausgeführt. Diese Pads sind in einem Raster von 2,54 mm angeordnet und zum Flashen habe ich einen einfachen Adapter mit Pogo-Pins gebaut, der auf die Pads der Platine gedrückt wird. Auf dem Adapter befinden sich noch zwei Knöpfe, die die Boot- und Reset-Pins auf Masse ziehen, sowie Schraubklemmen, an die ein USB-zu-Seriell-Adapter vom Typ HW-417 angeschlossen werden kann.

Für den Schaltplan und das Layout der Platine habe ich die kostenlose Software EasyEDA (Standard) von JLCPCB verwendet.

Multiplexing

Die LCD-Segmentanzeige, die ich in meinem Projekt verwende, ist ein sogenanntes Passiv-Matrix-Display. Bei diesem Displaytyp könnte prinzipiell jeder Bildpunkt bzw. jedes Segment über eigene Leiterbahnen angesteuert werden. Dies würde jedoch zu einer hohen Anzahl von Anschlüssen am Display bzw. Ausgängen an der Ansteuerhardware führen. Um dies zu vermeiden, wird üblicherweise die sogenannte Matrix-Ansteuerung mit Multiplexing verwendet. Jedes Segment befindet sich an einem Kreuzungspunkt der Leiterbahnen.

In der Abbildung ist das Prinzip anhand von neun LEDs dargestellt, die in drei Zeilen und drei Spalten angeordnet sind. Beim zeilenweisen Multiplexen schließt die Ansteuerhardware nacheinander die Schalter, die die Kathoden mit Masse verbinden. Während der Schalter einer Zeile geschlossen ist, sind alle Schalter der Spalten, in denen die LEDs leuchten sollen, ebenfalls geschlossen. Sie verbinden



also die Anode mit der Betriebsspannung, sodass nur die LEDs am Schnittpunkt zweier geschlossener Schalter leuchten. Wenn die Ansteuerhardware diesen Vorgang in schneller Folge wiederholt, blinken die LEDs zwar in Wirklichkeit, unser Auge nimmt dies aber bei hoher Frequenz nicht wahr. Stattdessen erscheinen die LEDs als konstant leuchtend.



Abbildung 5: Zuordnung der einzelnen Segmente der Anzeige zu den Speicheradressen des LCD-Chips (a, b, c, ...). in Tabelle 1

Sowohl den Schaltplan als auch das Layout stelle ich im verlinkten GitHub-Repository (siehe Link in Kurzinfo) zur Verfügung.

Ansteuerung der LC-Anzeige

Die eigentliche Ansteuerung des Displays erfolgt über das sogenannte Multiplexing, das vor allem dazu dient, Anschlusspins zu sparen. Weitere Informationen zum Multiplexing finden Sie im Kasten Multiplexing.

Über die technisch interessante Umsetzung des Multiplexing brauchen wir uns jedoch keine weiteren Gedanken zu machen, da der LCD-Treiber dies selbstständig erledigt. Wir müssen dem Treiber nur über I²C mitteilen, wie viele der vier verfügbaren Backplanes genutzt werden.



Abbildung 6: Zuordnung der einzelnen Segmente einer Ziffer zu den jeweiligen Bits

Beim von mir verwendeten LCD-Treiber werden die Anschlüsse des Displays statt Spalten und Zeilen Backplanes und Segmente genannt. Der eingesetzte PCF8576C kann maximal vier Backplanes mit jeweils 40 Segmenten ansprechen. Mit 44 Kontakten kann er also 160 Segmente adressieren. Analog zum Beispiel mit den LEDs (siehe Kasten), beschaltet er die Segment- und Backplane-Anschlüsse des Displays in schneller Abfolge.

Die elektrische Schaltung des LCD-Treibers sieht in meinem Projekt wie folgt aus: Die Pins SDA und SCL sind die Daten- und Taktpins des I²C-Busses. Diese verbinde ich direkt mit den entsprechenden Pins auf dem ESP32. Der Sync-Pin dient zur Synchronisation, wenn mehrere LCD-Treiber kaskadiert sind. Er bleibt ebenso wie der CLK-Pin als externer Clock-Eingang unbenutzt und daher unbeschaltet. Am Pin VDD wird die Versorgungsspannung des Treibers angelegt. Diese darf laut Datenblatt zwischen +2 Volt und +8 Volt liegen. Da diese Spannung sowohl an den Ausgängen des LC-Displays als auch auf dem I²C-Bus ausgegeben wird, habe ich mich für 3,3 Volt entschieden. Das Display läuft im Original mit 5 Volt, funktioniert aber auch mit 3,3 Volt problemlos. Für die Kommunikation mit dem ESP32 muss bei 3,3 Volt nichts weiter beachtet werden, da er mit 3,3 Volt Versorgungs- und Logikspannung arbeitet.

Der OSC-Pin aktiviert den internen Oszillator und muss gemäß Datenblatt mit VSS verbunden werden. VSS ist die Masse des Treibers. Die Pins A0, A1 und A2 legen die Subadresse des Treibers fest, ebenfalls nur bei Kaskadierung relevant. Ich habe mich entschieden, A0 auf +3,3 Volt zu legen, A1 und A2 auf Masse, was zur Subadresse 1 führt. Pin SAO legt die I²C Adresse fest. Mit ihm kann zwischen den Adressen 00111000 (SA0 auf Masse) und 00111001 (SA0 auf +3,3 V) umgeschaltet werden. Da die Adresse 00111000 bereits durch den DHT20 Sensor belegt ist und dort nicht geändert werden kann, habe ich den Pin SA0 auf +3,3 Volt gelegt. So haben LCD-Treiber und DHT20 unterschiedliche Busadressen und ich kann sie an einem Bus betreiben.

Kommunikation mit dem LCD-Treiber

Leider habe ich keine passende Arduino-Bibliothek für den LCD-Treiber PCF8576C gefunden, sodass ich selbst den Code zur Kommunikation schreiben musste. Hierzu war es nötig zunächst zu verstehen, wie der Treiber die Daten erwartet und speichert. Sein Speicherlayout ist in Tabelle 1 für den Betrieb mit vier Backplanes dargestellt.

In den Spalten stehen die einzelnen Segmentausgänge (n, n+1, n+2, ...), in den Zeilen die vier Backplanes (1 bis 4). Sendet man Daten an den Treiber, so erwartet dieser mindestens 1 Byte, also zwei Spalten und einen Data-Pointer (Zeiger auf den RAM Speicher im Treiberchip), der auf die erste Spalte zeigt. Hier muss man aufpassen, denn es können auch Pointer übergeben werden, die auf die zweite Spalte eines Bytes zeigen. Man übergibt also guasi zwei halbe Bytes. In welcher Anwendung das Sinn ergibt, kann ich nicht sagen. In meinem Code verwende ich jedenfalls nur ungerade Pointer. Um den Treiber nun dazu zu bringen, Daten anzuzeigen, müssen einige Bytes gesendet werden. Listing 1 zeigt eine Funktion aus meinem Code, die genau ein Byte sendet und anzeigt. Es hat sich für mich bewährt, die Wire-Transmission vor jedem Senden zu starten und danach wieder zu beenden.

In der Funktion wird zunächst die Wire-Transmission mit der l²C-Adresse busAddress des LCD-Treibers gestartet. Das folgende Byte modeSet legt fest, wie viele Backplanes verwendet werden, ob der Stromsparmodus aktiviert werden soll und ob das Display aktiv ist. Im nächsten Byte wird bei Verwendung mehrerer Treiber festgelegt, für welchen Treiber die folgenden Daten bestimmt sind. In meinem Fall ändern sich die ersten drei Bytes nie, da es nur einen Treiber gibt und ich auch keinen Moduswechsel vornehme. Das Byte pointer (der Daten-Pointer) legt schließlich fest, an welche Stelle im Speicher des Treibers das folgende Byte data geschrieben wird.

Zuordnung der LC-Segmente

Um die entsprechenden Bits (a, b, c, ...) in Tabelle 1 den einzelnen Segmenten der Anzeige zuzuordnen, habe ich ein Programm geschrieben, das die einzelnen Bits (ab a) nacheinander einschaltet. Jedes Mal, wenn eine "1" an die serielle Schnittstelle gesendet wird, wird das eingeschaltete Bit um eine Position im Byte nach links geschoben. So bleibt nach jedem Bit/Segment Zeit, die Zuordnung zu notieren. In Abbildung 5 ist die Zuordnung der Segmente auf meinem Display mit meiner Schaltung dargestellt.

Um nun mit den einzelnen Segmenten auch Ziffern darstellen zu können, müssen noch jeder Ziffer von 0 bis 9 die entsprechenden Segmente zugeordnet werden. Bei meiner Anzeige haben sich die Konstrukteure offensichtlich Gedanken gemacht, denn alle sieben Ziffern haben die gleiche Segmentzuordnung.

In der Abbildung 6 ist die Zuordnung zu den einzelnen Bits jedes Bytes mit nebenstehender Tabelle der binären Codierung der einzelnen Ziffern dargestellt. Dabei werden die Bits von links (MSB = 1) nach rechts (LSB = 8) gezählt.

Diese Belegung wird natürlich bei jedem Displaytyp unterschiedlich sein, ganz zu schweigen von zusätzlichen Symbolen. Die Belegung lässt sich jedoch mithilfe des oben

Listing 1: LCD-Treiber Kommunikation

```
void display_byte(int pointer, int data) {
    Wire.beginTransmission(busAddress);
    Wire.write(modeSet);
    Wire.write(deviceSelect);
    Wire.write(pointer);
    Wire.write(data);
    Wire.endTransmission();
}
```

genannten Codes, der auf GitHub hinterlegt ist, leicht herausfinden.

Touch-Keys auslesen

In der Glasscheibe vor dem LC-Display befinden sich links und rechts insgesamt fünf Touch-Felder. Da der ESP32 von Haus aus kapazitive Änderungen an den Pins GPIO 2, 4, 12, 13, 14, 15, 27, 32 und 33 erfassen kann, lassen sich die Leiterbahnen der Touch-Sensoren einfach direkt an diese Pins anschließen. Die gemeinsame Leitung der fünf Sensoren habe ich auf Masse gelegt. Der Codeschnipsel zum Auslesen der Touch-Eingänge ist in Listing 2 dargestellt. Mit touchAttachInterrupt wird ein Interrupt an den als TouchPin angegebenen Touch-Pad-GPIO (T0 bis T9) angehängt, der die Funktion functionToCall aufruft, wenn eine Berührung erkannt wird. Der Parameter threshold definiert den Schwellwert, d. h. die Empfindlichkeit der Erkennung. Den für Ihre Anwendung passenden Schwellenwert müssen Sie einfach durch Ausprobieren herausfinden. Höhere Werte bedeuten höhere Empfindlichkeit. In der Schleifenfunktion kann der aktuelle Wert jedes Sensors auch einfach mit touchRead (Pin#) ausgelesen werden. Diese Werte können als Anhaltspunkt für den Schwellwert dienen.

Energiesparmaßnahmen

Im Dauerbetrieb mit aktiviertem WLAN verbraucht ein ESP32 etwa 0,5 Watt. Das klingt zunächst nach wenig, aber ein Jahr hat etwa 8760 Stunden. Der Verbrauch für ein einziges Fußbodenthermostat beträgt also ca. 4,3 kWh pro Jahr. Bei mindestens fünf Thermostaten, die ich einsetzen möchte, sind das bei den



Tabelle 1: Speicheraufbau des LCD-Treibers

heutigen Energiepreisen immerhin 21,5 kWh pro Jahr, also gut 8 Euro, die man leicht vermeiden kann. Da die Fußbodenheizung ein sehr träges System ist, brauche ich weder Messwerte mit hoher Wiederholrate zu senden, noch zu prüfen, ob das Relais geschaltet werden muss. Ich habe für mich entschieden, dass ein Intervall von fünf Minuten für beide Anwendungen ausreicht. Deshalb schicke ich den ESP32 immer dann in den Schlafmodus, wenn die Sensoren ausgelesen, das Relais bei Bedarf geschaltet, die Messwerte gesendet und alle Befehle vom MQTT-Broker abgeholt wurden. Für mich hat sich der "leichte Schlaf" (Light-sleep bei Espressif) als pragmatische Lösung erwiesen. Meine gesamte Displayeinheit (inkl. Low-Dropout-Regulator, LCD-Treiber und schlafendem ESP32) verbraucht in diesem Modus 18 mW.

Wohlgemerkt bleibt die Darstellung auf dem Display dabei erhalten, da der LCD-Treiber aktiv bleibt. Während der aktiven Phasen des ESP32 werden im Mittel etwa 425 mW verbraucht. Da dies aber nur ca. 10 Sekunden während eines 5-Minuten-Zyklus auftritt, ergibt sich ein mittlerer Verbrauch von rechnerisch 32 mW. Das ist weniger als die Hälfte des Leistungsbedarfs der originalen Displayein-

```
Listing 2: Auslesen der Touch-Eingänge
(gekürzt)
void setup() {
    ...
    touchAttachInterrupt(TouchPin, functionToCall, threshold);
    ...
void IRAM_ATTR functionToCall () {
    ...
}
```

```
Listing 3: Taskmanagement (Auszüge)
xTaskCreate(TaskConnectWiFI, "TaskConnectWiFi",3000, NULL,0, &Task1);
...
void TaskConnectWiFI( void * pvParameters ) {
  for (;;) {
    delay(200);
    client.loop(); //ständiges abrufen von Nachrichten via MQTT
    publish_json();//Daten via MQTT versenden
...
  }
}
```

heit. Pro Thermostat und Jahr macht das einen Verbrauch von 0,28 kWh, für alle fünf Einheiten also 1,4 kWh oder nur noch gute 50 Cent an Stromkosten!

Neben dem zyklischen Erwachen aus dem Schlaf habe ich alle Touch-Sensoren ebenfalls als Aufweck-Quellen (in Espressif-Sprech "wakeup sources") definiert. Somit kann das Thermostat jederzeit durch Berühren aktiviert werden und es reagiert binnen Sekundenbruchteilen auf Eingaben des Nutzers (zum Beispiel zum Einstellen der gewünschten Temperatur).

WLAN und MQTT im Hintergrund

Das regelmäßige Schlafen des ESP32 bedingt es, dass jedes Mal, wenn der Controller erwacht, die Verbindung zum WLAN und MQTT-Broker neu aufgebaut werden muss. Daneben fallen für den ESP noch weitere Aufgaben an. Neben dem Abholen der MQTT-Daten zur Steuerung des Thermostats müssen auch noch alle ausgelesenen Sensordaten an den MQTT-Broker abgeliefert werden. Ich habe mich für beide Wege dazu entschieden, die Daten als JSON-Strings zu übertragen. Dies bedeutet jedoch, dass alle empfangenen Daten aus dem JSON-String in Variablen abgelegt und alle zu versendenden Daten in einen JSON-String verpackt werden müssen. Damit all diese Tätigkeiten samt einem ggf. auch einmal scheiternden Verbindungsaufbau nicht die Reaktion der Touch-Sensoren blockieren, habe ich sie in einen eigenen Task ausgelagert. Dazu bietet das auf dem ESP32 standardmäßig laufende Echtzeit-Betriebssystem FreeRTOS zahlreiche Möglichkeiten. Ich erzeuge darin mit der Funktion xTaskCreate() einen Task. Der Funktion werden dabei die folgenden Parameter übergeben: Task-Funktion, Name des Tasks, Task-Stack-Size, Übergabeparameter an den Task, Task-Priorität und Handle des Tasks.

In Listing 3 ist in Auszügen zu sehen was damit gemeint ist. Die oben genannte Task-Funktion ist hier TaskConnectWiFI, welche die Funktion, die als Dauerschleife läuft, solange der Task ausgeführt wird. In ihr verbinde ich den ESP32 mit dem WLAN, dem MQTT, sende die Daten an den MQTT-Broker und prüfe, ob alle Aufgaben erledigt wurden, sodass das WLAN wieder getrennt werden kann und der ESP32 wieder schlafen gehen darf. Der Name des Tasks ("TaskConnectWiFi") bezeichnet schlicht den Task. Er hat in meinem Programm keine weitere Bedeutung.

Der Parameter Task-Stack-Size (im Listing 3000 Bytes) ist die Größe des Stacks, die der Task für sich allokiert. Ich habe sie durch Trial-and-Error festgelegt. Mittels der Funktion uxTaskGetStackHighWaterMark(NULL) habe ich mir in der Task-Funktion die genutzte Größe des Stacks ausgeben lassen und die Stack-Size so eingestellt, dass das Stack sicher nie ausgeht. Da ich keine Probleme mit zu knappem Speicher habe, war ich hier etwas großzügiger.

Es gibt keine Parameter (NULL) für die Task-Funktion. Die Task-Priorität (0 im Listing) gibt an, mit welcher Priorität der Task behandelt wird. Sie startet bei null und größere Werte bedeuten eine höhere Priorität. Versuche mit einer Priorität von 1 haben bei mir dazu geführt, dass den Betriebssystem-Tasks nicht mehr genug Ausführungszeit zur Verfügung stand, was zu Systemabstürzen geführt hat. Mit einer Priorität von 0 funktioniert hingegen alles zu meiner vollsten Zufriedenheit.

Das Handle des Tasks (&Task1) dient mir schließlich dazu, mich in anderen Funktionen auf den Task zu beziehen, also insbesondere, um ihn anzuhalten und wieder zu starten. Den Task halte ich mittels der Funktion vTaskSuspend(Task1) an, bevor sich der ESP in den Schlaf verabschiedet. Nach dem Aufwachen setze ich den Task mittels vTaskResume(Task1) wieder fort.

Achtung: Anfänglich dachte ich, vTask-Suspend(Task1) löscht den Task, sodass ich nach jedem Aufwachen mittels xTaskCreate() einen neuen erzeugte. Dies führte binnen kürzester Zeit dazu, dass dem ESP sein dynamischer Speicher (Heap) ausging und er abstürzte. Nach einem Reset funktionierte er wieder tadellos, bis der Speicher erneut ausging. Diesem Problem bin ich mit der Funktion esp_get_minimum_free_heap_size() auf die Schliche gekommen, die den zum Ausführungszeitpunkt freien Heap ausgibt.

Unergründliches Speicherleck

Das schnelle Schrumpfen des freien Heaps durch die vielfach erzeugten Tasks konnte ich mittels oben genannter, korrekter Verwendung von Tasks lösen. Es zeigte sich jedoch, dass ein kleines Memory-Leak verblieben ist. In meinem Code nimmt der freie Heap sehr langsam, aber kontinuierlich ab. Dies führt wiederum zu einem Absturz des ESP, sobald der Heap zur Neige geht. Ich habe festgestellt, dass bei meiner Lösung, die nur alle fünf Minuten aus dem Schlaf erwacht, der Heap für etwa 30 Tage reicht. Da ich auch nach tagelanger Suche dem Leck nicht auf die Schliche kommen konnte, entschied ich mich für einen etwas unschönen, aber funktionierenden Workaround. Ich überprüfe einfach in der loop()-Funktion mittels oben erwähnter Funktion esp_get_minimum_free_heap_ size(), ob der Heap sich einem kritischen Wert nähert. In meinem Fall sind das, mit etwas Sicherheit, 5000 Byte. Werden diese unterschritten, löse ich mittels ESP. restart() einen Reset aus. Da dieser Fall, wie erwähnt, nur etwa alle 30 Tage eintritt, kann ich damit gut leben.

Sollte Ihr ESP32 sich nach einer mehr oder weniger langen Periode einwandfreier Funktion plötzlich aufhängen, lohnt sich ein Blick auf den Heap allemal. Vielleicht ist Ihre Software auch von einem Memory-Leak geplagt.

Kuschelig sparen

Die fünf Thermostate in unserem Haus sind jetzt seit Ende November 2023 in Betrieb und funktionieren einwandfrei. Auch von der Wächterin über Ästhetik im Haus (meine Frau) gab es bisher keine Klagen. Die Thermostate kommunizieren via MQTT mit ioBroker und ermöglichen mir die Auswertung meiner Messwerte in Grafana (siehe Aufmacherbild).

Das Wichtigste für mich ist jedoch, dass die Laufzeit der Wärmepumpe bei Sonnenschein erhöht und nachts reduziert wurde. An ganztägig sonnigen Tagen, die es seit November tatsächlich vereinzelt gab, musste ich die Wärmepumpe nicht mehr vorzeitig abschalten und konnte viel zusätzliche Energie im Estrich speichern. Je nach Außentemperatur reichte diese Energie teilweise für die ganze Nacht. Die dadurch vor allem in den späten Nachmittagsstunden um ca. 1-2 °C erhöhte Raumtemperatur mindert unser Behaglichkeitsempfinden nicht. —caw

Noch bis Pfingsten 10% Rabatt auf alle Besuchertickets.

Am 09.06. ist Anmeldeschluss für Aussteller.

Hannover Maker Faire[®]

Make:

17.–18. Aug. 2024

Zeigt eure Projekte!

jie is

uhat you

Private Maker, Bildungseinrichtungen, Makerspaces, offene Werkstätten, Vereine, uvm.

Wir haben für euch kostenfreie Standflächen.

Jetzt informieren und anmelden!

www.maker-faire.de/hannover

Maker Faire: Was war, was wird

Das Jahr ist noch jung und trotzdem gab es schon zwei Maker Faires. Für die kommende Hannover Maker Faire ist sowohl der Call for Maker, als auch der Ticketverkauf weiterhin im vollen Gange. Und die Redaktion hat sich auf der Maker Faire Ruhr getummelt.

von Daniel Schwabe

Die Maker Faire in Heilbronn und die Maker Faire Ruhr in Dortmund sind erfolgreich über die Bühne gegangen. Das heißt, dass es schon mit großen Schritten auf das Maker Faire-Highlight des Jahres zugeht: auf die Maker Faire Hannover vom 17. bis 18. August 2024.

Call for Makers und unschlagbare Rabatte

Bis zum 9. Juni 2024 können sich Aussteller noch anmelden, die ihre spannenden Projekte auf der Maker Faire Hannover zeigen wollen. Auf der offiziellen Maker-Faire-Website ist der Call for Makers geöffnet. Privatpersonen, offene Werkstätten, Schulen, Hochschulen und Universitäten, Makerspaces, Fablabs sowie gemeinnützige Vereine und Einrichtungen können sogar kostenlos ausstellen. Also los!

Haben Sie ein spannendes Projekt, das Sie präsentieren möchten, melden Sie sich an! Je mehr Aussteller mit interessanten Präsentationen dabei sind, um so bunter und inspirierender wird die Maker Faire!



Hector 9000

Die Make-Redaktion und die Shop-Besatzung motiviert für die Maker Faire Ruhr. Das Bild zeigt von links Oliver Hofgärtner und Mark Strothoff (Shop), Ákos Fodor, Johannes Börnsen, Peter König, Heinz Behling, Dunia Selman, Daniel Schwabe und Carsten Wartmann. Vorne in der Mitte: Chefredakteur Daniel Bachfeld.

Maker Faire



Makey war natürlich auch dabei und ganz verzückt von all den kreativen Projekten.

Wer ausschließlich als Gast kommen möchte, um sich von all den tollen Ideen inspirieren zu lassen und die neuesten Entwicklungen in der Maker-Szene zu sehen, kann sich jetzt noch einen saftigen Rabatt von 10 Prozent auf die Eintrittskarten sichern. Dieser Rabatt gilt bis zum 21. Mai.

Maker Faire Ruhr

Vom 16. bis 17. März 2024 fand in Dortmund auf dem Gelände der DASA-Arbeitswelt-Ausstellung die Maker Faire Ruhr statt. Über 50 Aussteller haben dort ihre Projekte gezeigt, rund 5000 Besucher und andere Maker inspiriert und selbst auch die ein oder andere neue Idee mit nach Hause genommen.

Natürlich war auch die Redaktion des Make Magazins mit dem heise-Shop dabei! Am Make-Stand konnten sich Besucher bekannte Projekte aus dem Magazin ganz genau anschauen: gehackte IKEA-Produkte, voll ausgestattete Maker-Koffer in verschiedenen Größen und natürlich die redaktionseigene Posteule! Erfolgreich testete die Redaktion auch ein neues Standkonzept, das auf Euroboxen und Holzplatten beruht. In Hannover bauen wir damit einen großen Stand. Zu den Projekten standen dann auch die jeweiligen Redakteure bereit, um Fragen zu beantworten, oder sogar Verbesserungsvorschläge entgegenzunehmen. Alles wurde fleißig von unseren Social-Media-Kanälen begleitet, damit auch die Maker zu Hause einen Einblick in das Geschehen bekamen (oder sogar noch zum Besuch motiviert wurden).

Aber die Redaktion besteht natürlich auch aus Makern, die mit neugierigem Blick über die Messe gegangen sind, um sich inspirieren zu lassen oder vielleicht das nächste große Titelprojekt zu finden. Denn bei der Make kann jeder Autor werden. Der stellv. Chefredakteur Peter König hielt an zwei Tagen einen Vortrage dazu, wie man einen Artikel im Magazin veröffentlichen kann.

Ob automatische Cocktail-Mixer, ein Rudel von Star Wars Robotern oder Experimente an gezüchteten Lungenzellen, auf der Maker Faire Ruhr gab es viel zu sehen. Und auch zu lernen! In verschiedenen Vorträgen konnten interessierte Maker in unterschiedliche Themen reinschnuppern.

Die Redaktion freut sich schon auf die nächsten Maker Faires und auf alle Maker, die tolle Projekte vorstellen, Fragen stellen oder sich in die Riege der Make-Autoren einreihen. —das

OXOCARD



Die Oxoard Science+ ist ein leistungsfähiger Raumsensor, eine Experimentierplatine und dank der offenen Programmierschnittstelle, kann man damit auch hinter die Kulissen schauen, die Programme bei der Ausführung beobachten und alles verändern.

8 Eingebaute Sensoren

- SENSIRION SHT40 - SENSIRION SCD41 - SENSIRION SGP41 - TM MS5607-028A03 - BROADCOM APDS-9251 - KNOWLES MK-SPK0641 - STM VL5SL5CX - MEMSIC MC3479

Computer in Kreditkartengrösse - ESP32 2 MB RAM, 8 MB Flash - TFT-Display 240×240 Pixel - Fünf Taster - Piezo-Lautsprecher



0

Browserbasierte Scripting-Umgebung mit Debugging! Enthält über 100 fertige Beispiele







Jetzt im heise Shop bestellen

In der Schweiz bei Brack www.oxocard.ch





Entspannen ist nicht! Für Social-Media mussten die Redakteure ihre Maker-Skills zeigen.

In den industriellen Hallen der DASA haben sich die Maker superwohlgefühlt.



Gefährlich wie das Magazin. Brandheiße Vorträge wie diesen gibt es nur auf der Maker Faire.



Die spannenden Projekte der Make warten auf die Besucher.



Ohne blinkende Lichter ist eine Maker Faire undenkbar.

TECHNIKUNTERRICHT MACHT ENDLICH SPAB!

Spannende Unterrichtsmaterialien GRATIS

Make: Education

25% - 0.25

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



Jetzt kostenlos downloaden: make-magazin.de/education

Calliope mini 3 ausprobiert

Mit dem Calliope mini 3 ist eine neue Version des bekannten Mikrocontroller-Boards für Schülerinnen und Schüler erschienen. Welche neuen Features er mitbringt, haben wir uns in einem ersten Test angeschaut.

von Ákos Fodor



er Kinder im Grundschulalter großzieht oder sie gar unterrichtet, ist vielleicht schon mal mit dem Calliope mini in Berührung gekommen. Dieses Mikrocontroller-Board ist nämlich speziell für den Unterricht ab der dritten Klasse konzipiert und soll Kinder an die Themen Elektronik und Programmierung heranführen. Jetzt ist das Board in der dritten Version erschienen und der Hersteller hat uns freundlicherweise ein Exemplar zugesendet, sodass wir einen Blick auf seine neuen Funktionen werfen konnten.

Für Anfänger entwickelt

Als deutsche Weiterentwicklung des BBC micro:bit, einem britischen Entwickler-Board für den Bildungsbereich, ist auch der Calliope mini von Werk aus mit Sensoren, LEDs, Sound-Ein- und Ausgabe-Funktion, einem Motortreiber und weiteren Features ausgestattet. Das ermöglicht direkt nach dem Auspacken einfache Elektronik-Experimente durchführen, ohne dass man zusätzliche Komponenten benötigt – ein PC oder Tablet zum Programmieren mal ausgenommen.

Charakteristisch für den Calliope mini ist seine sechseckige Form, die ihn nicht nur optisch von anderen Mikrocontroller-Boards unterscheidet, sondern auch einen funktionalen Hintergrund hat: An den sechs Zacken des Calliope mini befinden sich nämlich übergroße Touchpins (VCC mit 3,3 V, GND und 4 GPIOs), die groß genug sind, damit man etwas mit Bananensteckern oder Krokodilklemmen daran befestigen kann - und weit genug voneinander entfernt sind, damit man dabei nicht so leicht einen Kurzschluss verursacht. Außerdem besitzt der Calliope mini zwei Grove-Stecker, über die man I²C- und SPI-Module anschließen kann, etwa Displays, Sensoren, Relais oder andere.

Um den Calliope mini zu programmieren, kann man das blockbasierte MakeCode von Microsoft verwenden, wahlweise im Browserfenster am PC oder eingebettet in die mobile Smartphone-App, die den Calliope mini sogar per Bluetooth flashen kann. Das Board unter-

Kurzinfo

Mehr zum Thema

- » Lisa Ihde, Ein Piano mit Knete-Tasten programmieren, Make 1/22, S. 62
- » Elisabeth Giesemann und Natalia Prost, Verschlüsselter Bluetooth-Chat, Make 5/18, S. 62
- » Elisabeth Giesemann und Nora Perseke, Turnbeutel mit Leuchtzeichen, Make 4/18, S. 58

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

stützt aber auch JavaScript, MicroPython und die Arduino IDE für Fortgeschrittene sowie weitere IDE-Schnittstellen.

Neue Komponenten, neue Möglichkeiten

Auf den ersten (flüchtigen) Blick sieht der Calliope mini 3 seinem Vorgänger ziemlich ähnlich (Bild 1). Es gibt nach wie vor zwei Grove-Anschlüsse, einen für das externe Batterie-Pack, die LED-Matrix mit 5 × 5 LEDs und die beiden Taster (A und B). Allerdings sitzt jetzt alles auf einer mattschwarzen Platine – und wenn man genauer hinschaut, bemerkt man auch, dass die Calliope-Entwickler ein paar Komponenten ausgetauscht, ergänzt und das Layout angepasst haben.

Da wäre zunächst der neue USB-C-Anschluss, der den MicroUSB-Stecker des Vorgängers ersetzt. Mit ihm unterstützt der Calliope mini 3 jetzt auch WebUSB im Google-Chrome-Browser. Damit kann man das Mikrocontroller-Board direkt über MakeCode flashen, ohne dass man die Firmware als HEX-Datei herunterladen und über den Dateibrowser auf den Calliope schieben muss – auch wenn der Button in MakeCode dann nach wie vor "Herunterladen" heißt (Bild 2).

Ebenfalls neu ist der Mikrocontroller Nordic nRF52833, den man bereits vom micro:bit v2 kennt. Mit seinen 64 Mhz, 128 kB RAM und 512 kB Flash ist er vielfach leistungsstärker als der vorherige nRF51822 (16 Mhz, 32 kB RAM und 256 kB Flash). Dafür ist der externe Flashspeicher IS25LP weggefallen, auf dem sich zuvor als eine Art Bibliothek bis zu 25 HEX-Dateien speichern ließen.

Mit dem Chip-Upgrade lässt sich der Calliope mini jetzt auch in den stromsparenden Sleep-Mode versetzen und über ein Event (z. B. per Tastendruck) aufwecken. Dazu gibt es in MakeCode die neue Erweiterung namens Power, die derzeit noch in der Entwicklung ist (Bild 3). Daher ist die Anleitung, die erklärt, wie man die Funktion mit Blöcken programmiert, noch unvollständig, sodass man etwas experimentieren muss (Bild 4). Auch die Aufzeichnungsfunktion von Audiodateien, mit der man ca. drei Sekunden Ton aufnehmen kann, ist noch in der Beta-Phase. Aber sie funktioniert und dank der selbsterklärenden Bausteine wie "Audioclip aufnehmen" oder "setze Mikrofonempfindlichkeit auf niedrig" auch ziemlich einfach. Damit sich aufgezeichnete Sounds besser anhören, sollte man in jedem Fall einen Lautsprecher anschließen (Bild 5), da der Piezo sich eher für einfache Melodien eignet.

Einfacher stecken mit Jacdac

Auch eine neue Anschlussmöglichkeit hat Einzug auf dem Calliope mini erhalten. Sie nennt sich Jacdac, stammt aus dem Hause Microsoft (siehe Link in der Kurzinfo) und befindet sich an der unteren Kante des Calliope-Boards in Form zweier PCB-Edge-Anschlüsse (Bild 6). Mit diesem System lassen sich Sensoren, Potis, Knöpfe, LEDs oder andere Module mit einem Mikrocontroller verbinden,







Bild 3: Über das Zahnrad in MakeCode gelangt man zu den Erweiterungen. Welche für den Calliope mini 3 gedacht sind, steht in der jeweiligen Beschreibung.

Bild 2: In MakeCode kann man den Calliope mini 3 jetzt auch direkt über WebUSB flashen.

ohne dass man dabei als Anwender groß etwas beachten oder konfigurieren muss. Dadurch, dass alle Stecker identisch funktionieren, kann man (auch beim Calliope mini) jeden Jacdac-Stecker anschließen, wie man möchte. Das geht beispielsweise mit Grove nicht: Denn auch wenn die beiden Anschlüsse auf dem Calliope mini gleich aussehen, ist der linke für I²C- und der rechte für die Kommunikation über UART gedacht. Außerdem sind Jacdac-Module meist so gebaut, dass sie mehrere Anschlüsse besitzen, sodass man sie parallel oder aneinandergereiht verwenden kann.



Bild 4: Nach dem Einschalten geht der Calliope schlafen. Währenddessen kann man die LEDs nicht mit Taste A auf Grün schalten. Das geht erst, wenn der Mikrocontroller mit B wieder aufgeweckt wurde.



Für die Verbindung nutzt Jacdac drei Pins, einen für den Datenaustausch sowie zwei weitere für die Stromversorgung. Diese läuft auf dem Calliope mini 3 über einen separaten Schaltkreis, sodass stromhungrige Module nicht mal aus Versehen den Mikrocontroller des Calliope beschädigen können – damit bietet das Board noch eine Stolperfalle weniger für angehende junge Entwickler.

Eine weitere Besonderheit von Jacdac: Sobald man ein Modul anschließt, erkennt der Calliope mini, was es kann. Das funktioniert, weil die Module selbst auch einen kleinen Mikrocontroller besitzen, der über einen Bus und das eigens entwickelte Jacdac-Protokoll mit dem Calliope mini, anderen Modulen und sogar dem Simulator in MakeCode kommunizieren kann. Dazu muss man die Jacdac-Erweiterung (Bild 7) zum Projekt hinzufügen und den Code auf den Calliope übertragen. Danach erkennt das Board, wenn man ein Jacdac-Modul ansteckt oder austauscht.

Auf der Calliope-Website findet man passende Module in Form eines Jacdac-Starter-Kits von KittenBot für 49 Euro, das 8 Module und 5 Kabel umfasst (Bild 8 und Link in der Kurzinfo). Im Mai soll ein weiteres Kit erscheinen, das Jacdac-Projekte u. a. mit zusätzlicher Energie versorgen soll. Wer neben dem Calliope noch einen micro:bit besitzt, findet auf der Website von KittenBot übrigens auch ein Set mit Adapter (Jacdaptor), der ihn Jacdac-fähig macht. Ansonsten ist Jacdac Open Source und Microsoft bietet für die gängigen PCB-Programme Vorlagen als Teil eines Development Kits an, sodass es auch möglich ist, eigene Module zu entwickeln. Wofür das Akronym Jacdac steht, behält Microsoft allerdings für sich.

GPIO-Header, Motoren-Treiber und LEDs

Wollte man bei den vorherigen Calliope-Modellen die GPIO-Pins nutzen, musste man sich entweder einen Header auf das Board löten oder nutzte den Vorschlag der Entwick-

1 Date

RGB

Learn More



Bild 6: Das Layout des Jacdac-Anschlusses. Auf der Rückseite sind die Pins gespiegelt, damit man den Stecker immer richtig steckt.

Bild 7: Für Jacdac gibt es bei den Erweiterungen eine eigene Rubrik. Auch ein paar Grove-Module sowie die Onboard-Sensoren unterstützen Jacdac über Software.

ler, die Kabel temporär mit Zahnstochern festzustecken. Mit Version 3 ist beides nicht mehr notwendig, denn die Boards werden direkt mit einem verlöteten GPIO-Header ausgeliefert, in den man Jumperkabel von beiden Seiten stecken kann. So rutscht selbst bei Prototypen, die sich viel bewegen, kein Kabel mehr heraus. Auf der Rückseite sind zudem alle Pins beschriftet (Bild 9) und durch den etwas flacheren Header sieht man gleich, welche Pins nach dem Stecken herausragen.

Die GPIO-Leiste ist außerdem etwas länger geworden, denn die sechs Pins für die Motorsteuerung sind ans Ende der Leiste gewandert. An diese kann man nach wie vor zwei Motoren anschließen, jedoch erlaubt es der neue Motortreiber (DRV8835) mit einer weiteren H-Brücke auch, beide Motoren unabhängig voneinander zu steuern - also auch in unterschiedliche Richtungen. Die Pins A und B sind in diesem Zuge M0+, M0-, M1+ und M1- gewichen (Bild 10). Damit lassen sich beispielsweise einfache fahrende Roboter bauen, die man – mit weiteren Sensoren ergänzt – so programmieren kann, dass sie selbstständig ihre Umgebung erschließen und Hindernissen ausweichen. Für stromhungrige Motoren lässt sich über die letzten beiden Pins (VM+ und GND) an der Leiste eine externe Stromquelle (bis 9 V) anschließen. Diese versorgt dann sogar das gesamte Board mit Strom und ermöglicht kompaktere mobile Projekte - auch wenn man keine Motoren verwendet.

Zu der einzelnen RGB-LED (WS2812C) der 2er-Version sind zwei weitere hinzugekommen, die jetzt dort platziert sind, wo sich vorher die Pins für die Motor-Steuerung befanden. Das sorgt natürlich ordentlich für Party-Stimmung. Man kann sie aber selbstverständlich auch sinnvoll einsetzen und für Feedback mit Sensoren koppeln oder mit ihnen feststellen, was das Programm auf dem Board gerade tut wie im Sleep-Mode-Beispiel auf Bild 4.

Learn More

Climate

Jacdac

iacdac

Jacdac for MakeCode

Display

Motor

grove-ultrasonic-

Jacdac for Grove ultrasonic Sensor at C16

jacdac

Audio

Learn More

AI

IoT

grove-soilmoisture-

Jacdac for Grove soil

moisture at C16

jacdac

Fazit

Mit den Upgrades der dritten Version haben die Calliope-Entwickler ihr Board technisch weiter an den micro:bit angeglichen, aber eigene Funktionen auch sinnvoll ausgebaut. Dass der mini jetzt standardmäßig einen verlöteten Header besitzt und die GPIOs beschriftet sind, vereinfacht das Anschließen von Jumperkabeln ungemein und senkt die Einstiegshürde. Zudem könnten Hersteller mit der vereinheitlichten GPIO-Leiste leichter Shields für den Calliope mini produzieren. Dafür wäre es gut, wenn die aktuelle Pin-Anordnung so bleibt. Auch die integrierten Motortreiber und die neue Jacdac-Schnittstelle (zu der es übrigens noch keinen Wikipedia-Eintrag gibt) sind praktische Features, für deren Verwendung man keine zusätzlichen Boards anschließen muss wie beim micro:bit.



Bild 8: Im Jacdac-Starterkit sind neben Knöpfen, NeoPixel-LEDs und Sensoren auch fünf Kabel enthalten.

Das macht sich allerdings auch im Preis bemerkbar: Der Calliope mini 3 ist auf der Hersteller-Website einzeln für 39,90 Euro erhältlich. Ein BBC micro:bit V2 liegt bei etwa 19 Euro. —*akf*



Bild 9: Die Beschriftung der GPIO-Pins auf der Rückseite des Calliope mini



Bild 10: Die Anschlüsse für die Motoren sind in die lange GPIO-Leiste gewandert und weiß hervorgehoben.

Die Jeder-Kann-Drucker Prusa MK4 und Bambu Lab P1S im Test

3D-Druck war lange Zeit ein Hobby für Spezialisten. Mit dem Prusa MK4 und dem Bambu Lab P1S beweisen zwei Drucker, dass es einfacher geht. PLA druckt auf beiden gut, bei anderen Kunststoffen gibt es aber Unterschiede.

von Johannes Börnsen und Pina Merkert



ange war 3D-Druck ein schwieriges Hobby. Es gab zwar diverse günstige Drucker, die liefen aber nicht immer rund und die Menge an Erfahrung, um die Maschinen in den Griff zu bekommen, überforderte viele Makerinnen und Maker. Jede neue Generation versprach, nun endlich benutzerfreundlich zu sein, konnte das Versprechen aber nicht ganz einhalten.

Um wirklich perfekte Ergebnisse zu erhalten, kommt es auf eine ganze Reihe von Parametern an. Die Düsentemperatur muss auf etwa 5 °C genau stimmen. Außerdem sollte ein 3D-Drucker ein beheiztes Bett haben, dessen Temperatur ähnlich exakt stimmen muss und das auch gleichmäßig warm wird. Der wichtigste weitere Parameter ist der Abstand zwischen Düse und Bett, der auf hundertstel Millimeter genau stimmen muss, damit sich Drucke nicht von der Druckfläche lösen. Die Mechanik muss ebenfalls auf Hundertstel positionieren können und der Extruder muss das Filament so fördern, dass der Plastikfaden nicht verrutscht und exakt mit der eingestellten Geschwindigkeit durch die Düse gedrückt wird. Außerdem muss der Drucker das Filament schnell genug zurückziehen können, wenn der Druckkopf ohne zu drucken zwischen Bauteilen hin und her fährt, damit der heiße Kunststoff keine Fäden zieht. In der Praxis wird man diese Punkte natürlich nicht bei jedem Fliamentwechsel neu überprüfen, Erfahrungswerte und Workarounds wie ein Papier zwischen Düse und Druckbett erleichtern das Einstellen.

All diese Parameter per Hand einzustellen wäre ziemlich anspruchsvoll und eine Kunstform, deren Sinn man hinterfragen darf. Denn inzwischen gibt es Sensoren und Voreinstellungen, die einem diese Arbeit abnehmen können. Die beiden von uns getesteten Drucker bewiesen im Testlabor, dass sich Makerinnen und Maker beim Druck von PLA nicht um kompliziertes Einstellen sorgen müssen. In Kombination mit ausgereifter Slicer-Software gelingen so auch 3D-Druck-Neulingen innerhalb der ersten Stunde schöne Drucke. Zusätzlich arbeiten beide Maschinen ausgesprochen rasant und drucken beispielsweise das Bötchen Benchy in weniger als einer halben Stunde.

Solide konstruiert

Der Bambu Lab folgt im Prinzip der Konstruktion edler Open-Hardware-Drucker wie dem Voron. Der Direct-Drive-Druckkopf wird im CoreXY-Prinzip von zwei mit dem Rahmen verschraubten Motoren beschleunigt. Das Portal besteht aus zwei parallelen Rundstangen, die runden Lager können Kräfte in alle Richtungen aufnehmen. Der Rahmen besteht aus gestanztem Blech, was ihn verwindungssteifer als die meisten Konstruktionen aus Aluminium Extrusionen macht. Der Extruder setzt auf doppelseitige Förderräder. Heizblock, Kühlkörper und

Kurzinfo

- » CoreXY-Drucker Bambu Lab P1S und Bettschubser Prusa MK4 im Vergleich
- » Vorteile von geschlossenen Bauräumen und Tipps für Materialien abseits von PLA
- » Einsteigerfreundliche Slicer, gute Presets, gute Automatiken zur Kalibrierung

Mehr zum Thema

- » Test: Textilgehäuse für Bettschubs-3D-Drucker, Make 4/24, S. 78
- » 3D-Drucker tunen Z-Achsen-Waagesensor beim Ender 3 nachrüsten, Make 2/23, S. 94
- » Testvideo zum Bambu Lab P1S





Als Bausatz kostet der Prusa MK4 rund 900 Euro, fertig montiert zahlt man 1200 Euro.



Den Bambu Lab P1S erhält man fertig montiert für 750 Euro. Im Set mit dem hier aufgesetzten Materialwechsler werden 1000 Euro fällig.



Market-Hotends bei Aliexpress.



Zwei Drucker, zwei Benchies, wenig Unterschied: Beide Drucker erzeugen das Bötchen in unter 30 Minuten. Links der Ausdruck des PS1, rechts des MK4.

größte Neuerung ist der "Nextruder", ein Direct-Drive-Extruder mit Planetengetriebe und einem großen Förderrad ohne Rillen. Dazu später mehr. Die Düse ist eine auf die Edelstahl-Heatbreak aufgeschrumpfte Baugruppe. Prusa bietet aber einen Adapter an, den man zusammen mit normalen M6-Düsen anderer Hersteller verwenden kann. Das Mainboard trägt nun endlich auch einen 32-Bit-Controller, an der Vorderkante hängt nun ein Farbdisplay.

Düsenabstand

Der Bambu Lab P1S kommt mit einem induktiven Bettsensor und werksseitig eingestelltem Düsenabstand. Da Bambu Lab das Düsen-Ökosystem kontrolliert, sollte es nicht nötig sein, diesen Abstand noch nachzustellen. Der Sensor zeigte in unserem Test keinen temperaturabhängigen Drift, was bedeutet, dass Bambu Lab diesen in der Firmware kompensiert haben muss. In der Praxis kann man den Bambu Lab auspacken, anschließen und ohne Kalibrierung der ersten Schicht sofort losdrucken. Die Firmware-Einstellung zum Nachkalibrieren ist aber erreichbar, was entscheidend ist, um Hotend-Baugruppen anderer Hersteller einsetzen zu können.

Im Nextruder hat Prusa eine Wägezelle verbaut, die die Kraft messen kann, mit der der Drucker seine Düse auf die Druckoberfläche



Präzise und flexibel, aber schwer: der Druckkopf des Prusa MK4. Mit Adapter lassen sich hier handelsübliche Düsen nutzen.



Leicht und schnell, aber proprietär: Bambu möchte das Extruder-Geschäft gerne für sich behalten und bietet beispielsweise gehärtete Düsen als Upgrade.

auftippt. Diese Lösung zum Einstellen des Bettabstands ist genial! Der Drucker misst mit der tatsächlichen Düse und der tatsächlichen Druckplatte. Unterschiedlich dicke Druckplatten oder unterschiedlich lange Düsen spielen keine Rolle. Der MK4 misst seine Druckhöhe einfach jedes Mal selbst aus, ohne manuellen Eingriff. In unserem Test hatten wir mit PLA bei jedem Druck eine perfekte erste Schicht. Übrigens haben wir in Make 2/2023 bereits eine Anleitung (siehe Kurzlink) gezeigt, wie sie diese Lösung bei einem Ender 3 nachrüsten können.

Beim Drucken von PETG kamen uns aber leichte Zweifel an dem System. Damit die Messung durch die thermische Ausdehnung der Hotend-Komponenten nicht verfälscht wird, misst der MK4 mit warmem Bett und einer Düsentemperatur von ungefähr 40 °C unter der Drucktemperatur. Das ist warm genug, dass eine winzige Menge Plastik bereits flüssig wird und Richtung Düse wandert. Bei PLA hinterlässt der MK4 deswegen an den Messpunkten des Bed-Meshs winzige Punkte auf der Druckfläche. Diese sind aber so klein, dass wir sie selbst an der glatten Unterseite von PLA-Drucken nicht erkennen konnten. Sie nerven nur ein klein wenig beim Reinigen der Druckplatte. PETG klebte allerdings weniger gut an der Druckfläche, weshalb der kleine Punkt nicht kleben blieb, sondern sich als winziger Tropfen an der Düsenspitze sammelte, der zur Seite weggedrückt wird. Die anschließend gedruckte erste Schicht hätten wir per Hand minimal näher (0,02 bis 0,035 Millimeter) eingestellt, um die Betthaftung zu erhöhen. Um das nachzustellen, kann man den Knopf mehr als zwei Sekunden gedrückt halten, was ein geheimes Menü zum Nachstellen des Düsenabstands öffnet.

Gehäuse

Trotz unserer kleinen Meinungsverschiedenheit mit der Automatik des MK4 hätten wir die erste Schicht von Drucken mit PETG oder ASA weiterhin als "perfekt" bezeichnet. Probleme gab es jedoch, als weitere Schichten dazu kamen. Der MK4 wird ohne Gehäuse geliefert und war den minimalen Luftströmungen unseres knapp 20 °C warmen Büros schutzlos ausgeliefert. Unsere Testdrucke mit ASA hafteten unter diesen Bedingungen im Prinzip gar nicht an der texturierten Druckplatte, die Prusa im Handbuch empfiehlt. Auch mit Klebestift wurde keiner der Drucke fertig. Etwas mehr Erfolg hatten wir mit einer angetrockneten Schicht Haarspray, einem Trick, den wir bereits auf der Glasplatte von Ultimaker-Druckern erfolgreich nutzen konnten. Trotzdem würden wir einen MK4 ohne Gehäuse nur für PLA empfehlen. Schon PETG neigte ohne Gehäuse zu sichtbarem Warping. Abhilfe schafft vermutlich das offizielle Prusa Enclosure für 370 Euro, mit der beliebten Umbauanleitung für zwei Lack-Tische von IKEA haben wir zusammen mit dem MK3S gute Erfahrungen gemacht.

Der Bambu Lab P1S kommt serienmäßig mit Gehäuse. Für rund 150 Euro weniger gibt es aus dem gleichen Hause auch den P1P, was im Grunde der gleiche Drucker ohne Gehäuse ist – in unseren Augen lohnt sich der moderate Aufpreis jedoch definitiv.

Bambu Lab X1C

Bambu Lab stieg mit einer Kickstarter-Kampagne für den X1C in den Markt für 3D-Drucker ein. Die 1200 Euro teure Maschine bietet gegenüber dem später erschienenen P1S nur leichte Vorteile: Das Gehäuse besteht aus Aluminium, das Display ist ein größerer Touchscreen, das Bett heizt bis 120 °C und neben der Düse ist ein Lidar verbaut. Mit diesem Sensor kann der X1C sein Bett ausmessen und Pressure Advance automatisch kalibrieren. Beides scheint in der Praxis nicht wichtig zu sein, weil die mitgelieferten Voreinstellungen beim P1S ausreichend gut funktionieren. Wir waren davon überrascht, dass der technisch interessante Sensor so wenig bringt und würden wegen des besseren Preis-Leistungs-Verhältnisses den P1S empfehlen.

Extrusions-Konsistenz

Extruder nach dem Prinzip von Bondtechs BMG sind beliebt, weil sie das Filament von zwei Seiten und damit im Zweifelsfall mit viel Gewalt in die Düse drücken können. Die Förderräder sind dafür gleichzeitig Zahnräder, die sich gegenseitig antreiben, das Filament wird durch eine Rille geführt. Prusas MK3S nutzt dieses Prinzip und auch sämtliche Bambu Lab-Drucker arbeiten danach.

Die 3D-Druck-Community diskutiert seit einer Weile, ob genau dieses Prinzip für inkonsistente Extrusionsmengen sorgt. Das Argument ist, dass die Förderräder wegen der Rille nicht überall den gleichen Umfang haben. In der Mitte der Rille ist der Umfang kleiner als an den Rändern und prozentual ist der Effekt größer, wenn die Förderräder kleiner sind. Sollte das Filament beim Drucken von der Mitte zur Seite wandern, würde etwas mehr gefördert, was den Druck in der Düse erhöht, wodurch der Drucker minimal breitere Druckbahnen hinterlässt. Wandert es zurück, verkleinert sich der Umfang wieder und die Extrusion wird wieder schmaler. In der Praxis sollte sich das in Form von nicht ganz perfekt glatten Wänden zeigen. Wissenschaftlich quantifizieren konnte die Community das Problem bisher nicht. Minimale Inkonsistenzen bei der Extrusionsmenge könnten nämlich auch viele andere Gründe wie schwankende Durchmesser des Filaments oder eine labbrige Z-Achse haben.



Wem das monochrome Display des P1S zu spartanisch ist, der kann alternativ alle Funktionen des Displays auch über die App auf dem Smartphone nutzen.

Prusa lässt es beim MK4 nicht auf Vermutungen ankommen und eliminiert das Problem konstruktiv: Das Extrusionsrad hat keine Rille, stattdessen drücken zwei glatte Führungsrollen das Filament an. Sollte es wandern, bleibt die Extrusionsmenge gleich, da der Umfang des Rades überall gleich ist. Der Preis dafür ist eine etwas geringere Maximalkraft, die der Extruder auf den Filamentfaden ausüben kann. In unserem Test war diese jedoch völlig ausreichend und das Filament rutschte nicht durch.

Im Prinzip ist die Lösung beim MK4 elegant, weil sie ein mögliches Problem sicher löst. Andererseits konnten wir beim P1S keine sichtbaren Inkonsistenzen in der Extrusion erkennen. Beim P1S scheint das Filament innerhalb so enger Toleranzen geführt, dass es nicht weit genug in den Extruder-Rillen wandern kann, um die Druckqualität sichtbar zu beeinträchtigen.

Geschwindigkeit

Mit dem Voron sind FDM-3D-Drucker entscheidend schneller geworden. Zum einen erlaubt die CoreXY-Konstruktion eine erheblich geringere Masse zu bewegen, was rein physikalisch Vibrationen reduziert. Zum anderen begannen die Entwickler der Drucker-Firmware Klipper mit"Input Shaping" die Vibrationen vorherzusagen und die ungewollten Bewegungen mit



Das Farb-LCD des Prusa macht einen etwas zeitgemäßeren Eindruck als das monochrome Display des P1S, in der Praxis funktionieren aber beide Bedienkonzepte gut. Beide Drucker werden aber jeweils über den Knopf neben dem Display bedient und müssen auf einen Touchbildschirm verzichten.



PLA-Drucke haften auf der Druckplatte des MK4 gut, die Voreinstellungen führen zu sehr guter Druckqualität bei kurzen Druckzeiten.

gewollten Bewegungen auszugleichen. Die CoreXY-Drucker von Bambu Lab (X1C, P1S und P1P) replizieren beide Innovationen und gehören damit zu den schnellsten 3D-Druckern, die man fertig montiert kaufen kann.

Input Shaping funktioniert aber genauso bei Bettschubsern. Das Bett ist schwerer als das Portal eines CoreXY, weshalb rein physikalisch nicht die gleichen Beschleunigungen möglich sind, aber der Abstand ist nur etwa



Bei CoreXY-Druckern wie dem P1S bewegt sich das Druckbett lediglich Schicht für Schicht nach unten, die X- und Y-Achse übernimmt der Druckkopf. Bei Bettschubsern wie dem MK4 wird das schwere Druckbett dagegen nach vorne und hinten bewegt. Faktor 2. Seit Firmware-Version 5 beherrscht der MK4 auch Input Shaping. Interessanterweise misst er die eigenen Resonanzfrequenzen nicht wie der Bambu Lab mit einem Beschleunigungssensor am Kopf selbst aus. Prusa vertraut stattdessen darauf, dass alle MK4 exakt gleich schwingen. Trotz des fehlenden Sensors haben wir im Test keine Nachteile der simplen Voreinstellung erkennen können.

Als Messgröße für die Geschwindigkeit hat sich die #SpeedBoatRace durchgesetzt, also die Druckzeit eines Benchy-Boots mit 0,25 Millimetern Schichthöhe. Der Prusa MK4 druckt in 28 Minuten im leicht angepassten Speed-Profil ein fehlerloses Benchy. Der Bambu Lab P1S erledigt den Druck in 22 Minuten bei vergleichbar guter Qualität.

Die hohe Geschwindigkeit beider Drucker verbessert ihre Alltagstauglichkeit erheblich. Die sonst üblichen Druckzeiten können beide meist ungefähr vierteln. Da wir oft Drucke über Nacht laufen gelassen haben, war die genaue Druckzeit aber in vielen Fällen nicht wichtig. Der leichte Geschwindigkeitsvorteil vom P1S ist zwar messbar, im Maker-Alltag macht er aber höchstens bei sehr langen Drucken einen Unterschied.

Fazit

Sowohl der Prusa MK4 als auch der Bambu Lab P1S lösen das Versprechen ein, einfach zu funktionieren. Das ist für Anfänger essenziell, aber auch für alle attraktiv, deren Projekt gerade etwas anderes ist als ein Tuning des 3D-Druckers. Beide Maschinen drucken richtig flott und produzieren dabei eine vergleichsweise hohe Druckqualität. In unserem Video zum P1S (siehe Link in der Kurzinfo) haben wir verwirrende Einstellungen in Bambu Labs Slicer bemängelt, diese sind inzwischen aber durch ein Software-Update behoben.

Für die meisten Projekte ist PLA das ideale Material, weil es praktisch kein Warping gibt und es wegen seiner hohen Steifigkeit die stabilsten Teile bildet. PLA druckt auf beiden Druckern exzellent. Auch flexible Filamente funktionieren, solange die Shore-Härte nicht zu niedrig wird. Wir kennen jedoch auch keine kommerziell vertriebenen FDM-Drucker, die mit besonders flexiblem Filament entscheidend besser zurechtkämen. Besser dürfte nur der Riemen-Extruder des YouTubers "Proper Printing" funktionieren.

Für PETG, ASA, ABS, CPE und Nylon würden wir den Prusa MK4 immer in ein Gehäuse stellen wollen, das leider nicht dabei ist. Der Bambu Lab P1S hat von sich aus ein Gehäuse, was Warping deutlich reduziert. Beide Drucker haben aber durchaus trotz Gehäuse noch mit Warping zu kämpfen. Im Test lösten sich auf beiden Druckern Drucke vom Bett, weil Ecken sich so hochwölbten, dass sie mit der Düse kollidierten. Auch wenn sich Drucke nicht lösten, sorgte Warping öfter für hässliche Wände und nicht maßhaltige Kanten.

Als positive Überraschung sahen wir im Test keine schwankenden Temperaturen, ein gleichmäßig beheiztes Druckbett, rutschfrei haftende Federstahlplatten als Druckoberflächen und vor allem keine einzige erste Schicht mit deutlich falsch eingestelltem Düsenabstand. Diese Verlässlichkeit und Wiederholbarkeit ist entscheidend, um den 3D-Drucker zum alltagstauglichen Werkzeug zu machen. Beide Gräte können in diesem Punkt überzeugen. Der Bambu Lab ist dabei günstiger, hat die bessere App und druckt schneller. Der teurere Prusa wird weitgehend in der EU produziert, ist leiser und kann leicht repariert werden. Außerdem gab es für alle bisherigen Prusa i3 ein Upgrade-Kit auf die neue Revision.

Während für den MK4 das Mehrfarbdrucksystem MMU3 zwar angekündigt, aber noch nicht verfügbar ist, ist das Automatic Material System (AMS) von Bambu Lab längst erhältlich und ermöglicht den Druck mit bis zu 16 Filamenten (bei Kombination von vier AMS) im automatischen Wechsel. In unseren Tests lief das in den meisten Fällen problemlos, gelegentlich schaffte es das AMS jedoch nicht, das Filament beim Wechsel auf eine andere Rolle vollständig aus dem P1S zu ziehen. Nach kurzer manueller Hilfe konnte der Druck aber fortgesetzt werden.

Wer auf einen Einsatz ohne Cloudanbindung oder gänzlich ohne Netzwerkzugriff angewiesen ist, ist mit dem MK4 besser bedient. Zwar lassen sich auch beim P1S Druckdaten auf der MicroSD speichern und über das Menü am Drucker starten, ein Firmwareupdate ist jedoch nur übers Netzwerk möglich. —jom



Das digitale Abo für IT und Technik.

Exklusives Angebot für Make-Abonnenten:

Sonderrabatt für Magazinabonnenten

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

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

heise.de/plus-testen



🔀 leserservice@heise.de 🛛 📞 0541 80009 120

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

Frühjahrsputz im Smart Home

Natürlich macht es Spaß, immer neue Komponenten, Skripte und Module dem eigenen Smart-Home-System hinzuzufügen. Aber einmal im Jahr sollte sich der Administrator hinsetzen und gezielt das System ausmisten. Und was weiterläuft auf den aktuellsten Stand bringen. So klappt's systematisch und tut auch nicht weh.

von Robert Kränzlein

ngefähr 2015 fing es an: "Schatz, ist es in Ordnung, wenn ich drei smarte Heizkörperthermostate installiere?" Die Gattin gab ihre Zustimmung, ohne zu ahnen, dass da noch das eine oder andere Gerät im Nachgang hinzukommen könnte. Schnell gesellten sich ein Dutzend weiterer Heizkörperthermostate hinzu, gefolgt von einigen Fenstersensoren, Wandthermostaten und einer FHEM-Zentrale. Der Kaminkehrer bestand auf einer Abschaltung des Dunstabzugs, wenn die Küchenfenster geschlossen waren, und wollte bei dieser Gelegenheit ein mehrere hundert Euro teures System mit einem einzigen Fenstersensor an den Mann bringen. Grund genug für weitere Fenstersensoren und die erste Schaltsteckdose. Die Heizung machte immer wieder Zicken, weshalb Vor- und Rücklauf ein Messelement erhielten und die Störmeldung des Brenners wurde bald durch einen LDR an einem Arduino überwacht.

Wer ein Smart-Home hat, kennt das: Auch dabei blieb es nicht. Immer wieder ergeben sich neue Möglichkeiten. Im Make-Magazin finden sich Bauanleitungen für eigene Sensoren und Geräte und dann war da noch ein Tutorial, das man auch unbedingt ausprobieren musste. Irgendwann wurde FHEM abgelöst durch ioBroker, Homematic durch HomematicIP ergänzt und fast alle Lampen im Haus waren irgendwie smart. Von IKEA gab es diese praktischen kleinen Buttons, von Aquara einen Würfel, die Shellies stecken in jeder zweiten Steckdose, ein Rasenmäherroboter, ein Balkonkraftwerk, ein Staubsaugerroboter und so weiter und so fort wurden nach und nach im Smart Home vereint.

222

19:00 pm 🛛 🤶 🎟 📊

2023 sah das ganze dann so aus wie im Diagramm dargestellt – ioBroker attestierte 17.126 Objekte mit 15.163 Zuständen. Außer einem Neubau, der von A bis Z mit international standardisierten Modulen mit dem Gebäudeautomations-Feldbus KNX ausgestattet ist, werden wohl viele Smart Homes früher oder später in etwa diesen Umfang erreichen. Würde man einem Fachmann diese Anzahl an Datenpunkten nennen, würde er wohl fragen, wie viele Stockwerke das zu steuernde Krankenhaus hat.

Systematisch durchgehen

Jedes der uns zur Verfügung stehenden Geräte und Module überschüttet uns mit Datenpunkten, von denen nur ein Bruchteil jemals tatsächlich benötigt wird. Für gewerblich genutzte Anlagen zum Messen, Steuern und Regeln gibt es die VDMA 24186-4, die Richtlinien zur Wartung solcher Anlagen festlegt. Die Richtlinie enthält Checklisten zur Wartung, deren einzelne Punkte periodisch bzw. bei Bedarf ausgeführt werden. Üblicherweise erfolgt eine solche Wartung jährlich, sofern nicht besondere Ansprüche an Ausfallsicherheit oder erschwerte Betriebsbedingungen gegeben sind.

Für den eigenen Haushalt sind diese Checklisten nur bedingt geeignet, da z. B. kaum jemand Druckluftkomponenten verbaut hat oder Schaltschränke mit Lüfter und Filter. Sinnvoll wäre es andererseits schon, einmal im Jahr im eigenen Smart Home systematisch nach dem Rechten zu sehen. Lässt sich das Licht im Esszimmer nicht mehr einschalten. wird dies schnell auffallen, aber was ist mit dem Wassersensor unter der Küchenzeile? Hat er noch ausreichend Batteriespannung oder ist er seit Monaten offline? Funktioniert das Skript noch, das eine Warnmeldung an alle verschickt, wenn ein Fenstersensor geöffnet wird, obwohl sich niemand der Bewohner im Haus befindet, oder hat der Dienstbetreiber die Schnittstelle geändert oder abgeschaltet? Haben die Familienmitglieder noch die gleichen Mobilfunkgeräte wie bei der Einrichtung der Präsenzerkennung oder gab es da die eine oder andere Änderung?

Wir alle haben viel Spaß daran, Neues zu installieren und auszuprobieren, aber wer hat noch den Überblick, das alles auf Dauer nachzuhalten? Einmal im Jahr sollten wir uns hinsetzen und das System mal auf Herz und Nieren prüfen. Dabei hilft hoffentlich meine Checkliste von Seite 60, die es über den Link in der Kurzinfo auch zum Download gibt.

Backup ist Pflicht

Zu Beginn der Wartung ist die Erstellung eines Backups unbedingt erforderlich. Werden bei Updates oder Tests einzelne Komponenten oder Installationen beschädigt, gibt es dann die Möglichkeit, alles wieder auf den Ursprungszustand zurückzusetzen. Aber halt – genügt eigentlich *ein* Backup? Bei mir sind z. B. zwei Raspberries tätig. Einer davon hat einen USB-Stick, auf den ioBroker regelmäßig Backups schreibt. Nur wenn beide SD-Karten vollständig als ISO-Datei und der USB-Stick z. B. auf die Festplatte des NAS gesichert werden, lässt sich das System wieder komplett herstellen.

Zweiter Schritt, bevor die Arbeiten am System beginnen, ist die wichtige Frage: Sind eigentlich alle Benutzer- und Anmeldedaten noch vorhanden und auffindbar? Nicht nur der Login zur Smart-Home-Zentrale, sondern auch zu Apps von Drittanbietern, Cloud-Systemen, Schnittstellen wie TheThingsNetwork oder API-Schnittstellen zu Telegram, IFTTT, etc.? Falls nicht, sollte man sich vorab über die jeweilige Passwortwiederherstellung Zugang verschaffen. Übrigens: Es ist eine gute Idee, wenn die Benutzernamen und Passwörter irgendwo niedergeschrieben und diese Liste dann sicher verwahrt wird, für den Fall, dass der Smarthome-Administrator mal unvorhergesehen nicht erreichbar ist ...

Kurzinfo

» Leitfaden zur j\u00e4hrlichen Wartung am Smart Home
 » Erfahrungen aus der Praxis
 » Checkliste zum Abhaken als Download



Ans Eingemachte

Als Nächstes sollte man sich die Frage stellen, ob es nicht etwas auszumisten gibt. Alle Module und Plots, die im letzten Jahr nicht benutzt wurden, könnten deinstalliert oder sollten zumindest hinterfragt werden. Die Herstellerschnittstelle zu meinem Auto musste ich unbedingt ausprobieren, habe diese aber kein einziges Mal irgendwie genutzt, also weg

SHSW-1#F4CFA276492E#1	Garagen 🖬 device			
🖼 Relay0	Channel	4		
AutoTimerOff	Auto Ti O state	leveltimer		80
🗋 AutoTimerOn	Auto Ti O state	level timer		80
ButtonReverse	Button T O state		normal(0)	80
🗅 ButtonType	Button T O state		action	E0
ChannelName	Channel (C) state		(null)	80
Event	Event O state			8.0
EventCount	Event C 🔿 state			80
linput	Input / () state		false	80
D Switch	Switch 🖸 state	mitch	false	80
D Timer	Duration O state	level timer		80
longpush	Longpush 🔘 state		false	E 0
D longpushtime	Longpus O state		800 ms	E 0
🚹 source	source a O state		timer	80
📟 ext	Channel 🚺 channe			
humidity1	External _ (C) state	value humidity		8 0
humidity2	ExternalO state	value humidity		E 0
humidity3	External _ O state	value humidity		I 0
switch1	Switch 🔘 state			E 0
temperatureC1	External O state	value temperature		10
temperatureC2	External O state	value.temperature		E 0
temperatureC3	External O state	value.temperature		E :0
temperatureF1	External _ O state	value.temperature		0
temperatureF2	External	value.temperature		E 0
temperatureF3	External O state	value temperature		E 0
factoryResetFromSwitch	Factory rO state		true	E 0
🔓 firmware	New fir O state		false	0
firmwareupdate	Update O state			8.0
hostriame	Device _ O state		192.168.178.34	E 0.
🖨 id	Device Id 🔘 state		shelly1	E0
D name	Device _ 🔘 state		(null)	80
protocol	Protocol 🔘 state		соар	E0
C reboot	Reboot 🔘 state			E 0
🕼 rssi	Device RO state		-84 db	B (0
type	Device L., O state		SHSW-1	
D uptime	Uptime 🔘 state		108D23-29:38	
version	Firmwar [0] state		20210909-144114	10

Ausmisten ungenutzer Datenpunkte am Beispiel eines Shelly-Relais

Erst denken, dann updaten

Je nach den Gegebenheiten des eigenen Smart Homes kann ein Update in Ausnahmefällen für einzelne Komponenten auch kontraproduktiv sein. In Home Assistant kann man z. B. die Cloud-Steuerung von Tuya-Zigbee-Geräten per Hack umgehen, um sie ausschließlich lokal zu steuern (siehe

damit. In meinem gewachsenen System gibt es ein Trådfri-Gateway für die IKEA-Lampen und Shortcut-Buttons, einen deCONZ-ConBee-II-Stick, an dem Aquara-Komponenten und eine Philips-Hue-Leuchte angebunden sind und die in fast jedem Raum vorhandenen Echo-Geräte sind ja inzwischen auch in der Lage, mit ZigBee-Geräten zu kommunizieren. Mindestens eines, wenn nicht sogar zwei dieser Gateways könnte man sicher außer Betrieb nehmen.

Dabei ist zu berücksichtigen, dass Geräte, die bisher über Gateway A und B angebunden waren, und zukünftig über Gateway C kommunizieren, wie neue Geräte zu behandeln sind. Die mir bekannten Systeme erkennen in der Regel nicht, dass dieses Endgerät bereits Artikel unter "Mehr zum Thema" in der Kurzinfo). Tuya arbeitet da aber ständig gegen, indem sie per Firmware-Updates solche von der Cloud getrennten Geräte modifizieren, sodass diese ohne Cloud nicht mehr laufen. In solchen Fällen lässt man die Firmware lieber auf einem älteren Stand.

früher mit dem System verbunden war und ordnen Verknüpfungen und Skripte neu zu. Es ist empfehlenswert, sich hier vorher Notizen zu machen, mit welchen Partnern das Endgerät kommuniziert und welche Skripte darauf zugreifen. Dann wird das auszurangierende Gateway mitsamt der Endgeräte gelöscht, die Endgeräte mit Gateway C verbunden und alle benötigen Verknüpfungen und Skripte werden entsprechend angepasst.

Updates ziehen

Nach dem Ausmisten starte ich mit den Updates, beginnend über Putty mit den Updates auf dem Linux-Grundsystem, gefolgt von den Updates für die Installation von ioBroker, RaspberryMatic, PiHole und Node-Red. Dann folgen alle Module innerhalb des ioBrokers und zu guter Letzt ist zu prüfen, inwiefern die einzelnen Smart-Home-Komponenten ein Update bekommen (können). Ich verbiete z.B. meinen Shellies und smarten Steckdosen im Normalbetrieb die Kommunikation mit dem Internet. Nur bei der Wartung schalte ich in der FritzBox die Kindersicherung für diese Geräte ab, führe Firmware-Updates durch und sperre anschließend die Kommunikation nach außen wieder. Selbst Homematic-Fenstersensoren können über die eigene Zentrale Updates erhalten.

Funktionsprüfung

Eine VDMA-Vollwartung sieht vor, dass jeder einzelne Sensor und Aktor auf Funktion geprüft wird. Hierzu sind meist zwei Personen notwendig – einer an der Schaltzentrale und ein weiterer im Gebäude, der z. B. mit einem geeichten Thermometer einen Vergleich mit den Messungen der Sensoren herstellt.

Diesen Aufwand würde ich zu Hause nicht treiben. Denn ein defekter Sensor liefert meist ein Messergebnis, das am untersten oder obersten Ende des möglichen Messbereiches liegt. Wenn ein Raumthermostat zu Hause -15 °C oder



Schematische Darstellung meines Smart-Home-Systems!

© Copyright by Maker Media GmbH

+70 °C anzeigt, benötige ich kein Thermometer, um dieses Messergebnis als falsch einzustufen. Aber anschauen muss ich jeden Messwert zumindest einmal, ebenso, wann zuletzt ein Ergebnis übertragen wurde. An den Fenstern ist es dagegen sinnvoll, wenn eine zweite Person durchs Haus läuft und auf Zuruf ein Fenster nach dem anderen kippt und wieder schließt.

Ebenso wie die Sensoren sollten alle Aktoren geprüft werden. Schließt z. B. das Magnetventil noch zuverlässig oder bleibt es immer mindestens 10% offen stehen? Nicht nur physikalische Aktoren sollten geprüft werden, sondern auch logische, wie das Verschicken von Warnmeldungen über WhatsApp, Telegram und IFTTT. Auch hier habe ich mir bei der Gelegenheit übrigens die Frage gestellt, ob alle drei Dienste wirklich notwendig sind, oder ob nicht einer davon entfallen kann.

Logik-Prüfung

Im Anschluss an die Ein- und Ausgabegeräte geht es an die Logik. In meinem ioBroker gibt es eine ganze Vielzahl von Skripten, die Batteriezustände überwachen, Warnmeldungen generieren, Berechnungen anstellen oder einfach morgens die Kaffeemaschine starten. Laufen alle dieser Skripte noch oder benutzen sie verwaiste Variable? Ideal wäre es, künstlich unterschiedliche Ausgangsszenarien für die Skripte zu schaffen und dann zu beobachten, ob diese das gewünschte Resultat ausgeben, was allerdings einen gewissen Aufwand bedeutet.

Aufschlussreich sind zur Logik-Prüfung auch die Logfiles der diversen Systeme. Kaum jemand wird von sich behaupten können, alle Einträge in diesen Logfiles zu verstehen und bewerten zu können. Aber Fehlermeldungen, die gehäuft oder zumindest regelmäßig zur gleichen Zeit auftreten, wären es schon wert, ihnen weiter nachzuspüren. Gibt man den Text der Fehlermeldung bei Google ein, findet sich sicher eine Reihe von Foreneinträgen, die das Problem und dessen Lösung näher einzugrenzen helfen.

Hardware-Check

Nach der Softwareseite überprüfe ich die Hardware aller Geräte, also sowohl der im Haus verteilten Sensoren und Aktoren als auch der Smart-Home-Zentralen und Gateways. Sind Beschädigungen am Gehäuse sichtbar? Sitzen alle Stecker fest? Gibt es geknickte Antennen? Besonderes Augenmerk ist hierbei auf den Zustand der Batterien zu richten. Gerade bei Sensoren, die vor Gas- oder Wasseraustritt warnen, sollten die Batterien spätestens bei 20% der Restkapazität getauscht werden. Bei Heizkörperthermostaten halte ich es persönlich hingegen für vertretbar, diese erst bei Ausfall zu tauschen. Aber hier muss jeder seine eigene Strategie entwickeln – abgestimmt auf die eigene Smart-Home-Landschaft und die eigenen Bedürfnisse.

Finales Backup

Nach Abschluss der eigentlichen Wartung steht nun wieder ein Backup des kompletten Systems an. Ich würde empfehlen, dabei das eingangs gemachte Backup nicht zu überschreiben, sondern beide Backups parallel zu behalten. Somit ist nach ein paar Tagen immer noch ein Rollback möglich, falls sich Inkompatibilitäten zwischen einzelnen Updates der Einzelsysteme ergeben.

Auf der nächsten Seite folgt eine Checkliste für die vorstehend beschriebenen Arbeiten. Probieren Sie es aus! Ich bin mir sicher, in jeder Smart-Home-Installation finden sich defekte Geräte, leere Batterien oder Skripte, die nicht mehr tun, was sie eigentlich sollen. Ein regelmäßiger Frühjahrsputz lohnt sich also auch im Smart Home. —pek

Mac&i Wissen erfahren

Webinar – 16. Mai 2024

Apple Vision Pro im Unternehmen einsetzen

Tauchen Sie ein in die revolutionäre Welt der Apple Vision Pro und entdecken Sie, wie Spatial Computing den professionellen Alltag transformieren kann.

Erfahren Sie in diesem Webinar, wie sich Apples Headset im Firmennetzwerk integrieren und verwalten lässt. Entdecken Sie das Potenzial der Vision Pro für den professionellen Einsatz und profitieren von praktischen Erfahrungen.



Jetzt Ticket sichern: heise-academy.de/webinare/apple-vision-pro



Checkliste Smart-Home-Wartung

Position	Tätigkeit	erledigt	
1	Backup		
1.1	Backup aller Smart-Home-Zentralen		f-fr
1.2	Backup aller weiteren Geräte, sofern hier ein Backup möglich ist		
1.3	Sicherheitskopie von USB-Sticks und SD-Karten, die im Smart Home als Speicher dienen		19:00 PM
1.4	Überprüfung auf Füllstand der Speichermedien. Freie Kapazität min. 20%		
2	Datensparsamkeit		
2.1	Entfernen nicht benötigter oder bereits demontierter Geräte		
2.2	Entfernen nicht benötigter Datenpunkte		
2.3	Entfernen unbenutzter Module		
2.4	Entfernen nicht benötigter Aufzeichnungen/Plots		
3	Update		
3.1	Update des Linux-Grundsystems		
3.2	Update der Smart-Home-Zentralen auf aktuellen Softwarestand		
3.3	Update von Mobil-Apps oder Smart-Home-Touch-Bildschirmen		
3.4	Update aller Module und Plug-ins		
3.5	Update aller Feldgeräte		
4	Funktionsprüfung		
4.1	Funktionsprüfung aller Eingangsmeldungen selten benötigter Sensoren wie Wassermelder. Bei ständig laufenden Sensoren nur Logikprüfung		
4.2	Logikprüfung aller Eingangsmeldungen laufend genutzter Sensoren: Datum letztes Eingangssignal? Werte plausibel?		
4.3	Funktionsprüfung aller Ausgaben (Telegram, WhatsApp, Warnleuchten, Displays, LEDs,)		
4.4	Funktionsprüfung aller Skripte		
5	Hardwareprüfung		
5.1	Sichtprüfung der Geräte auf Beschädigungen		
5.2	Sichtprüfung und Reinigung der Smart-Home-Zentralen		
5.3	Prüfung von Antennen		
5.4	Prüfung von Steckverbindungen		
5.5	Prüfung aller Batteriezustände, ggf. Austausch der Batterien/Akkus		
6	Abschluss		
6.1	nochmaliges Backup alle Komponenten		
6.2	Beschriftung und sichere Ablage der Backups		
6.3	ggf. Ersatzteilbeschaffung und Austausch defekter Bauteile		

CE WIR TEILEN KEIN HALBWISSEN WIR SCHAFFEN FACHWISSEN



Passkeys statt Passwörter

Das Webinar erläutert, was Passkeys sind, wie sie funktionieren und wie man sie im Alltag nutzen kann. Ebenso werden Fragen hinsichtlich der Sicherheit von Passkeys geklärt.



Einführung ins Threat Modeling

In diesem Workshop lernen Sie Threat Modeling als leichtgewichtige Methode kennen. Sie erfahren, wie Sie zu einem "Positive Threat Mindset" kommen, sodass Sicherheitslücken proaktiv vermieden werden.



Node.js intensiv

Der Workshop bietet einen umfassenden Einstieg in die Entwicklung serverseitiger Anwendungen mit Node.js. Sie lernen die wichtigsten Basics, die Architektur und Best Practices kennen.



Einführung in Power Bl

Der Workshop stellt praxisnah die wichtigsten Grundlagen des BI-Konzeptes und die verschiedenen Komponenten vor. Außerdem lernen Sie die Vorteile von Power BI gegenüber dem klassischen Excel kennen.



Einführung in den Kea DHCP Server Der Workshop gibt eine vollständige Einführung in die neue Kea-DHCP-Software auf Unix- und Linux-Systemen. Sie lernen, wie man das Kea-DHCP-System installiert, konfiguriert und wartet.



Hacken für Anfänger Im Webinar erfahren Sie, wie Sie in einer virtuellen Trainingsumgebung erste Gehversuche unternehmen. Sie lernen außerdem, wie Sie Kali Linux einrichten und den anzugreifenden Server bereitstellen.

Sichern Sie sich Ihren Frühbucher-Rabatt: heise.de/ct/Events

Messen mit dem Smart Home

Eine Smart-Home-Software wie ioBroker kann nicht nur ein Haus steuern, sondern auch als vielseitiges Messsystem dienen – hier für die Kapazitätsmessung von Kamera-Akkus.

von Erik Golz

.

Vor dem letzten Urlaub kam die Frage auf, ob ich genügend Akkus für die Kamera habe. Also schnell einen Originalakku des Herstellers und zwei Fremdfabrikate samt Ladeschale gekauft. Nach dem Urlaub blieben Fragen offen: Sind die Originalakkus besser als die Fremdfabrikate? Gefühlt waren sie es. In den letzten Jahren hatten sich zudem einige Akkus angesammelt. Wie gut sind die alten Akkus noch? Welche sollten aussortiert werden?

So entstand die Idee, einen Prüfstand zu bauen: kontrollierte Entladung über einen definierten Widerstand, Messung der Akkuspannung während der Entladung. Natürlich mit automatischer Abschaltung bei Erreichen einer vorgegebenen Entladeschlussspannung. Dabei sollte aus der gemessenen Spannung bei bekanntem Widerstand der entnommene Strom bis zum Abschalten berechnet werden und die kumulierte Addition dann die entnommene Kapazität in mAh oder mit der Spannung multipliziert in mWh ergeben.

Ziel war es nicht, die absolute Kapazität der Kamera-Akkus wissenschaftlich zu bestimmen, sondern die eigenen Akkus unter definierten Bedingungen zu testen und zu vergleichen. Wie kann man also diese Messung einfach und ohne großen Aufwand durchführen?

Grundsätzliche Überlegungen

Von einem anderen Projekt war noch ein Shelly Uni übrig. Dieses vielseitige WLAN-Modul kann Spannungen messen und zwar umschaltbar im Bereich 0–12 Volt oder 0–30 Volt. Zusätzlich kann es zwei Ausgänge schalten, ist also wie geschaffen für dieses Projekt. Alle weiteren Funktionen werden dann mit der Smart Home-Software ioBroker ausgeführt. Eine Software, die eigentlich für die Steuerung eines Smart Homes gedacht ist, aber durch ihre extreme Vielseitigkeit bestens dazu taugt, den Shelly Uni zu steuern und die Messwerte zu verarbeiten.

Kurzinfo

» Messstation für Akkus mit Shelly Uni bauen
 » Smart-Home-System ioBroker als Messsoftware nutzen
 » Analyse und Darstellung mit ioBroker

Checkliste	Werkzeug
Zeitaufwand: etwa 5 Stunden für Aufbau und KonfigurationKosten: ca. 25 Euro ohne Stromversorgung	 » Lötkolben und Zubehör » Multimeter » Übliches Maker-Werkzeug Zangen, Schraubendreher, …
	Mehr zum Thema
 Material » Ladeschale passend für die Akkus » Breadboard » 10 Lastwiderstände je nach Akku, hier 10 Ω, 2 W » Shelly Uni » Relais 12 V 1×UM » LED grün » Vorwiderstand für LED 1 kΩ » Werkzeug-Akku 18 V mit Adapter und 12-V-Ausgang oder Labornetzteil » Hardware-Plattform für ioBroker Raspberry Pi oder Host mit Proxmox o. ä. » Kleinteile Kabel, Schrumpfschlauch, Jumperkabel, etc. 	 Andreas Koritnik, Smarter Gaszähler mit Smarthome-Integration, Make 3/23, S. 20 Andreas Koritnik, Temperatursensoren für den ioBroker Gaszähler, Make 5/23, S. 48 Daniel Bachfeld, Docker für Raspberry Pi, Make 1/21, S. 92 Gerd Michaelis, Der Akku-Lagerungs-Lader, Make 6/17, S. 94 Docker auf dem Raspberry Pi installieren (Einsteiger-Tutorial)
	Alles zum Artikel

Der Vorteil dieser Vorgehensweise liegt darin, dass alle Schritte nacheinander und in sich abgeschlossen durchgeführt werden können. So wird zuerst die Hardware bestehend aus Akku und Widerständen zusammengebaut und anschließend der Shelly Uni mit Spannungsmessung und angesteuertem Relais in den Stromkreis integriert. Dies alles kann über die eigene Webseite des Shelly Uni kontrolliert und bedient werden.

im Web unter

make-magazin.de/xd1c



Wenn dann alles zur Zufriedenheit funktioniert, geht es daran, den Testaufbau mit der Software ioBroker zu steuern und dort die eigentliche Messroutine in Form einiger kurzer, leicht verständlicher Skripte zu erstellen.

Da für alle Akkus nur eine einzige Messung geplant war, muss nicht unbedingt eine aufwendige Visualisierung programmiert werden. Es genügt, die Messung mit dem entsprechenden Skript zu starten und nach Abschluss der Prüfung die entnommene Kapazität in der entsprechenden Variablen abzulesen. Die Dokumentation der Messwerte für jeden Akku kann dann mit Block und Bleistift erfolgen.

Weil es aber später ohne großen Aufwand möglich war, eine Visualisierung für das Smartphone zu erstellen und den Spannungsabfall des Akkus bis zur Endspannung zu speichern und grafisch darzustellen, habe ich auch das gemacht. Für die eigentliche Fragestellung wäre dies jedoch nicht notwendig gewesen.

Aufbau des Prüfstands

Am Anfang stand die Überlegung, mit welcher Last die Prüflinge entladen werden sollten und wie lange die Messung mit dieser Last somit dauert. Die Sony NP-FW50-Akkus bzw. deren Nachbauten sind mit 7,2 V und 1020 bis 1080 mAh spezifiziert. Bei einer Last von 100 Ohm fließt je nach genauer Spannungslage ein Strom von ca. 72 mA. Die spezifizierte Kapazität wäre damit in ca. 14,5 Stunden entladen. Natürlich kann der Akku nicht bis zur Entladeschlussspannung von 2,5 V pro Li-Ion-Zelle, bei zwei Zellen im Akku also bis 5 V, entladen werden, da die Kamera bereits bei ca. sieben Volt Akkuspannung einen erschöpften bzw. leeren Akku anzeigt und den Betrieb einstellt. Die 7 V waren also auch hier der festgelegte Abschaltwert, der eine Messung beendet.

Der Akku sollte auch nicht kontinuierlich entladen werden, da sich die Akkuspannung nach jeder Lastunterbrechung wieder erholt und ohne diese Erholung der Abschaltwert zu schnell erreicht wird. Um eine praxisnahe Belastung zu simulieren, wurde daher nach jeweils vier Minuten Entladung eine Pause von einer Minute eingelegt, was in etwa der Nutzung des Akkus beim Fotografieren entspricht.

In der Durchführung ergab sich damit je nach Akkuqualität eine Messdauer von 10 bis 12 Stunden, was problemlos über Nacht durchgeführt werden kann.

Shelly Uni



Der Shelly Uni ist eines der vielseitigsten Geräte von Shelly. Es kann entweder mit 12–24 Volt Wechselspannung oder mit 12–36 Volt Gleichspannung betrieben werden. In diesem Projekt funktionierte es jedoch bereits ab einer Gleichspannung von 9 Volt zuverlässig. Vier Grundfunktionen stehen zur Verfügung:

- » Der Analog-Eingang misst eine analoge Gleichspannung im Bereich 0-30 V oder 0-12 V (über Webmenü konfigurierbar).
- » Die Eingänge Input 1 und 2 erfassen logische Schaltzustände (high/low).
- » Bis zu drei Temperatursensoren DS18B20 oder ein Umweltsensor DHT22 können an den DATA-Sensor-Port angeschlossen werden.
- » Zwei Open-Collector-Schaltausgänge steuern externe Geräte.

Nach dem ersten Einschalten baut der Shelly Uni ein eigenes (ungeschütztes) WLAN auf, in dem er mit einem Webbrowser unter der Adresse 192.168.33.1 angesprochen wird. Unter "Internet&Security/WiFi mode – Client" wird der Shelly im lokalen WLAN angemeldet, entweder mit einer festen IP-Adresse oder einer per DHCP zugewiesenen Adresse. Über diese Adresse erfolgt dann die weitere Konfiguration, z. B. wählt man unter "Settings/ADC Range" den Messbereich des ADC-Analogeingangs.

Der Shelly kann auch über eine App angesprochen werden, in diesem Projekt war jedoch die Funktionalität des eingebauten Webservers ausreichend. Der gesamte Aufbau wurde auf einem Breadboard durchgeführt. Eine alte Ladeschale für ein Universalladegerät wurde zerlegt und an den Plus- und Minuskontakten Leitungen angelötet. Mithilfe dieser Ladeschale kann der zu prüfende Akku schnell gewechselt werden. Als Last werden zehn in Reihe geschaltete Metallschichtwiderstände mit je 10 Ohm (2 W, 5%) verwendet. Über diese Last wird der Akku entladen. Bei einer Spannung von 7,2 V und einem Strom von 72 mA (I = U / R, I = 7,2 V / 100 Ω = 72 mA) nehmen die Widerstände zusammen eine Leistung von ca. 0,5 W auf. Da die zehn Widerstände zusammen 20 W vertragen können, ist die Belastung hier unkritisch.

Der Aufbau mit in Reihe geschalteten Widerständen ermöglicht es, durch Verringerung der Anzahl der Widerstände den resultierenden Lastwiderstand zu verringern und damit die Belastung des Akkus zu erhöhen, was die Messdauer verkürzen würde. Dieser Lastwiderstand wird einmal gemessen und in ioBroker als Wert in einem der Skripte gespeichert.

Bei diesem Aufbau würden die Akkus nun vollständig und unkontrolliert bis auf 0 V tiefentladen. Hier kommt der Shelly Uni (siehe Kasten) ins Spiel. Der ADC-Messeingang (Analog-Digital-Wandler) wird parallel zur Spannung des Prüfakkus geschaltet und kann im Webinterface des Shelly auf einen Messbereich von 0-30 V oder 0-12 V (wie hier) konfiguriert werden. Ein Schaltausgang des Shelly Uni könnte nun in Reihe zur Last geschaltet werden, um diese abzuschalten. Obwohl laut Datenblatt der Schaltausgang mit bis zu 100 mA belastet werden kann, habe ich für eine saubere Systemtrennung ein Relais in Reihe zum Lastwiderstand geschaltet und dieses Relais mit dem Ausgang des Shelly Uni angesteuert. Eine LED mit 1000 Ohm Vorwiderstand parallel zum Eingang des Relais zeigt an, ob das Relais geschaltet ist, d. h. ob gerade entladen wird.

Als Spannungsversorgung des Shelly Uni dient ein Werkzeugakku mit 18 V / 5 Ah , der über einen USB-Ladeadapter mit 5,5 mm-Hohlbuchse 12 V zur Verfügung stellt. Damit ist eine sichere Spannungsversorgung des Messaufbaus auch über einen längeren Zeitraum gewährleistet, wobei der zu prüfende Akku nur mit der Prüflast belastet wird, die Messung also nicht verfälscht wird.

Auf der eigenen Webseite des Shelly Uni (Abbildung 11) können nun die Schaltzustände der beiden Ausgänge angezeigt bzw. umgeschaltet werden, ebenso wird der Messwert des ADC angezeigt. Wenn alles wie beschrieben funktioniert, kann es mit ioBroker weitergehen.

ioBroker-Installation

Wenn die Testhardware steht, kann ioBroker installiert und für die Messung konfiguriert werden. Wer bereits eine ioBroker-Installation z. B. für sein Smart Home hat, ist auf der sicheren Seite und kann die zusätzlichen Funktionen direkt ausführen. Aber auch sonst ist es nicht schwer, ioBroker auf geeigneter Hardware zu installieren.

ioBroker läuft auf einer Vielzahl von Plattformen bzw. Betriebssystemen, darunter Linux und Windows. Optimal ist natürlich Hardware, die rund um die Uhr im Einsatz ist, etwa ein Raspberry Pi oder ein vergleichbarer Kleincomputer.

Eine weitere Möglichkeit ist die Virtualisierung: Hier läuft ioBroker als virtuelle Instanz auf einem Host, das kann Docker oder jeder andere Virtualisierer sein. Gerade Docker läuft auf vielen NAS von Synology oder QNAP und unter NAS-Software wie Unraid und ist dort schnell installiert.

Obwohl meine ioBroker-Installation im Smart Home seit Jahren zuverlässig auf einem Odroid xu4 (ähnlich dem Raspberry Pi) läuft, habe ich mich bei diesem Projekt entschieden, einen virtuellen Linux-Container (LXC) auf meinem Proxmox-Server aufzusetzen und io-Broker darin zu installieren. Zum einen wollte ich die Installation sauber vom Smart Home trennen, zum anderen hätte ich für die Anforderungen der Shelly-Anbindung (mindestens Node.js 16.0, js-controller 3.3.22, Admin Adapter 6.0.0) mein System updaten müssen.

Ein LXC-Container ist eine virtuelle Umgebung, in der ein isoliertes Linux installiert ist. Da der LXC-Container den Kernel des Hosts (hier Proxmox) nutzt, ist der Overhead gering und die Effizienz sehr hoch. Der LXC-Container ist unter Proxmox mit dem Button, Create CT", einem Debian-Template und den Parametern 2 GByte RAM, 32 GByte Root Disk und feste IP-Adresse schnell und ohne großen Aufwand installiert. Nach dem Start des LXC-Containers wird über die Konsole innerhalb von Proxmox (oder später über Putty) die eigentliche ioBroker-Installation entsprechend der Anleitung von ioBroker mit drei Befehlen (siehe Kasten) installiert.

Die erste Zeile aktualisiert den virtuellen Linuxrechner. Die zweite Zeile installiert, curl", ein Kommandozeilen-Programm, mit dem Dateien in Rechnernetzen übertragen werden. Dieses "curl" wird dann mit der dritten Zeile aufgerufen und installiert ioBroker inklusive aller notwendigen zusätzlichen Programme wie Node.js. Die Ausgabe:

ioBroker was installed successfully. Open http://192.168.2.23:8081 in a browser and start configuring!

ist dann der Lohn der Installation. Wie immer gilt aber, reboot tut gut.

Diese Installation kann statt über die Konsole alternativ über Putty erfolgen, dazu muss jedoch innerhalb des LXC-Containers in der Datei /etc/ssh/sshd_config der SSH-Zugang mit PermitRootLogin yes freigeschaltet wer-

Installation von ioBroker

apt-get update && apt-get upgrade --yes
apt-get install curl
curl -sLf https://iobroker.net/install.sh | bash -

den (nano /etc/ssh/sshd_config, anschließend mit STRG+O, STRG+X speichern).

Anschließend kann mit einem Webbrowser und der angegebenen lokalen Adresse des LXC-Containers und dem Port 8081 ioBroker gestartet werden. Wer Docker benutzt, startet nach Aufsetzen des Docker-Containers direkt mit der entsprechenden Adresse des Gastsystems und dem Port 8081.

ioBroker starten

Nach dem ersten Aufruf der ioBroker-Webseite startet der Konfigurationsassistent. Zuerst werden die Lizenzbedingungen akzeptiert, danach wird das Admin-Passwort festgelegt. Sofern der Zugriff lokal bleibt, also kein Port-Forwarding ins Internet geplant ist, muss die Authentifizierung nicht aktiviert werden. Anschließend wird noch der Standort festgelegt, die automatische Suche nach Geräten und Diensten kann übersprungen werden.

Nun gilt es, sich im ioBroker (Abbildung 2) zurechtzufinden. Wichtig für den Anfang sind die Einträge für Adapter, Instanzen und Objekte in der linken Spalte. Später kommen dort noch die Einträge Skripte und ggf. iQontrol hinzu. ioBroker ist, wie der Name schon sagt, ein Vermittler, und zwar einer zwischen Geräten und Funktionen. Geräte sind im Prinzip alles, was extern angeschlossen werden kann und Daten liefert, sei es ein physikalisches Gerät (wie der Shelly Uni) oder ein Dienst wie z. B. eine Wetterabfrage. Für die Weiterverarbeitung der erfassten Daten (Speicherung, Visualisierung, ...) werden entsprechende Adapter ähnlich wie bei einem App-Store hinzugefügt. Die große Anzahl an Adaptern, sei es zur Anbindung unterschiedlichster Hardware, sei es zur Aufbereitung und Weiterverarbeitung der Daten, macht die Leistungsfähigkeit von ioBroker aus.

Ein Klick auf den Eintrag "Adapter" öffnet im rechten Feld die installierten Adapter (mit blauer Kopfzeile und Angabe der installierten Version) und die verfügbaren Adapter (mit grauer Kopfzeile und Angabe der verfügbaren Version).

Für das Akkutest-Projekt werden folgende Adapter benötigt (Auswahl des entsprechenden Adapters, Klick auf die drei Punkte, Klick auf das Plus-Symbol):

- "shelly" – Adapter zum Anbinden des Shelly Uni. Die Konfiguration erfolgt später.



Abbildung 2: Die ioBroker-Hauptseite

C	v6.13.16 <	C) 4	Ð	4							
	Übersicht	=	:=	C	•	т					Filter	
-	Adapter	Freier Fe	stplattensp	eicher: 93	3%, Gesa	imte RAN	/I-Auslastur	g: 1264	Mb /	Frei: 40	% = 810 Mb [Se	rver: ioBroker-make
	Instanzen		0	admir	n.0		н	4	C	€	Admin	
=	Objekte		0	backit	tup.0			4	C		BackltUp	
83	Aufzählungen		* *	igonti	rol.0			2	C	Ð	iQontrol Vis	
	Protokolle			javaso	ript.0		п	2	C	_	Skriptausführur	ng
Po	Benutzer											
	Hosts		6	shelly	.0		н	2	C		Shelly	
	Dateien		shelly.0)			v6.6.1				ø	\$9.85 MB
0	Backup	Š	Lebens	zeicher	Host							
35	Skripte	0	Verbund Dienst	den mit	Gerä	l oder						
0	iQontrol			web.0)		п	4	C	Ð	WEB-Server	
			198	0.0000			1.00		1883			

Abbildung 3: Starten des Shelly-Adapters

II Übersicht	C III 🖕 🖿 🖕 🖿 🚺 Tr 🕂	Ŧ±	Objekte: 361, Zustande: 286 🏼 🔌
Adapter	ID Raum VI Funktio	on 👻 Wert	Einstellun 👻
Instanzen	🖿 backitup 🛞		
	iscovery		Î
Objekte	iqontrol 🖸		1
E Aufzählungen	iavascript 😈		ii .
	script		Π.
Protokolle	📂 shelly		1
Benutzer	🖕 o 🏀		8
	SHUNI-1#C45BBE6BE85D#1		8
Hosts	ADC ADC		1 I
Dateien	Power	2 V	E 0
	Range	12 V	目立
Backup	Cloud		10 m
Skripte	Mqtt		1 I
	Relay0		1
iQontrol	🗁 Relay1		ii ii
	AutoTimerOff	0 s	■○
	AutoTimerOn	0 s	E 0
	ButtonReverse	normal(0)	E 0
	ButtonType	toggle	E O
	ChannelName	(null)	E 0.
	Event		E 0
	EventCount	0	E 0
	linput	false	1 0
	Switch	false	E 0
	longpush	(null)	■ ○
	longpushtime	800 ms	B (1)

Abbildung 4: Der Shelly-Zweig in ioBroker

- "javascript" die Plattform f
 ür die zu erstellenden Skripte, keine Konfiguration erforderlich, also schlie
 ßen.
- Optional kann installiert werden:
- "iQontrol" sofern eine einfache Visualisierung z. B. auf dem Smartphone gewünscht wird. Die Demokonfiguration sollte geladen werden und erleichtert die spätere Konfiguration.
- "history" sofern gewünscht wird, die gemessenen Akkuspannungen nicht nur in den Skripten zu verarbeiten, sondern in einer JSON-Datenbank für weitere grafische Darstellung zu speichern.
- "flot" die Grafikausgabe der gespeicherten History-Daten.

Im Eintrag "Instanzen" in der linken Spalte werden die installierten Adapter aufgeführt, konfiguriert, gestartet oder gestoppt. Eine ampelähnliche Statusanzeige zeigt, ob ein Adapter inaktiv ist, eine Störung vorliegt oder läuft.

Der Shelly-Adapter kann mit einer größeren Anzahl von Shelly Uni oder anderen Shelly-Geräten verbunden werden. Es gibt aber Adapter, die genau einem Endgerät zugeordnet sind und daher mehrfach installiert werden, wenn dieses Endgerät mehrfach vorhanden ist. (Z.B. Fritzboxen, SAT-Receiver, Messwertaufnehmer, etc.) Im Eintrag "Instanzen" kann man auch nicht mehr benötigte Adapter löschen.

Im Eintrag "Objekte" sind alle Zustandswerte aufgelistet. Dies können erfasste Datenpunkte von Adaptern sein, aber auch selbst definierte Variable, am besten unter Ø_userdata abgelegt. Im Eintrag "Objekte" befindet sich auch ein Zweig für Shelly-Geräte, der bei Installation des Shelly-Adapters automatisch angelegt wurde. Vor der Konfiguration des Shelly Uni befinden sich hier allerdings noch keine Einträge zu diesem.

MQTT-Verbindung

Die Hardware um den Shelly Uni ist komplett. Der ioBroker ist aktiv. Beide müssen jeweils eine IP-Adresse aus einem gemeinsamen Adressraum haben, z. B. 192.168.2.168 für den Shelly Uni und 192.168.2.23 für den ioBroker (Port 8081). Jetzt werden die beiden Mitspieler miteinander bekannt gemacht. Dazu wird der Shelly Uni über das MQTT-Protokoll mit dem Shelly-Adapter im ioBroker verbunden.

Im ioBroker führt ein Klick auf das Tool-Icon von shelly.0 im Eintrag "Instanzen" zur Konfigurationsseite des Shelly-Adapters. Im Reiter "Allgemeine Einstellungen" können die Einträge unverändert bleiben, im Reiter "MQTT Einstellungen" werden der MQTT-Benutzername und das MQTT-Passwort festgelegt. In meinem isolierten Projekt wähle ich der Einfachheit halber als Kombination shelly/shelly.

Nach Aufruf der Webseite des Shelly Uni werden diese MQTT-Einstellungen unter "Internet & Security/Advanced -Developer Settings/Enable MQTT" übernommen: Haken bei "Enable MQTT" setzen, Benutzername und Passwort wie vorher gewählt. Unter "Server:" steht die Serveradresse der ioBroker-Installation, mit dem MQTT-Port 1882, sofern in ioBroker nicht geändert.

Anschließend mit "Save" die Konfiguration des Shelly-Uni abschließen und diesen einmal neu starten.

Im ioBroker kann nun der shelly.0-Adapter im Eintrag, Instanzen" (Abbildung 3) gestartet werden, der dann nach kurzer Zeit bei der Statusanzeige von grau bzw. rot auf grün wechselt. Klappt man das Menü des Adapters auf, sieht man drei grüne Häkchen für, Verbunden mit Host", "Lebenszeichen" und "Verbunden mit Gerät oder Dienst". Bleibt der Adapter rot oder springt auf gelb, müssen die eingestellten Parameter überprüft werden. Meistens handelt es sich um einen Zahlendreher in der Konfiguration, z. B. bei der eingestellten Serveradresse.

Zusätzlich erscheint nun unter "Objekte/ shelly/0/" der Shelly Uni (Abbildung 4). Hier wird u. a. die gemessene Spannung im Zweig "ADC" und der Zustand der Schalter im Zweig "Relais0" bzw. "Relais1" angezeigt.

"Relais0/Switch" (bzw. "Relais1/Switch") zeigt nicht nur den Ausgang des entsprechenden Schalters an. Hier schaltet ein Klick auf den Eintrag "true" bzw. "false" diesen Ausgang direkt um (was man im Webbrowser des Shelly Uni überprüfen kann: Dort kennzeichnet ein blauer Ring, dass der Ausgang eingeschaltet ist).

Die Skripte

Wenn alles soweit grün ist, geht es an die eigentliche Programmierung des Akkutesters. Die grafischen Skripte steuern die eigentliche Mes-

Übersicht C III 🖿 🛸	11 Tr + Tr ±	Objekte: 81	3, Zustande: 658
Adapter ID	Typ • Rolle • Raum	✓ Funktion	
Instanzen			
Objekte Skkutest	folder		28
Aufzählungen / Objekt bearbeite + Erstellen	n Alt+0 O state state	0	/80
Benutzer	Booleschen Zustand erstellen Nummern-Zustand erstellen	101,16 true 7,68	/80
Hosts Du_min	String-Zustand erstellen Zustand erstellen	7,68 (null)	/80
Backup alias	Kanal erstellen Gerät erstellen		
Skripte discovery iQontrol flot	Flet B meta		
Info	0		ii ii

Abbildung 5: Variablen-Definition

sung. Im Prinzip kann alles in ein großes Skript gepackt werden, aber aus Gründen der Übersichtlichkeit läuft hier für jede Teilaufgabe ein eigenes Skript. Dadurch ist es einfacher, die einzelnen Aufgaben zu implementieren und ggf. Fehler zu suchen und zu lokalisieren.

Zunächst werden im ioBroker (Abbildungen 5 und 12) einige Variable definiert. "Objekte/0_userdata/0" anklicken, dann auf das +-Symbol klicken. Als Typ "Verzeichnis" wählen und den Namen "Akkutest" eintragen.

Nun das neue Verzeichnis Akkutest anklicken, dann +-Symbol, diesmal "Typ: Zustand, Zustandstyp: Logikwert, Name: Messung" definiert die Binärvariable Messung, die später anzeigt, ob gerade eine Messung läuft.

Wieder "Akkutest" anklicken, dann +-Symbol, diesmal "Typ: Zustand, Zustandstyp: Zahl, Name: Lastwiderstand" nennen. Dies definiert die numerische Variable namens Lastwiderstand, die den Wert des Lastwiderstandes am Prüfaufbau speichert.

Die weiteren benötigten Variablen Kapa_ mAh, Kapa_mWh für die Ergebnisse sowie U_max, U_min für die maximalen und minimalen Spannungswerte während einer Messung werden, wie die Variable Lastwiderstand, als numerische Variable angelegt.

Ein Klick auf den Eintrag "Skripte" in der linken Spalte der ioBroker-Hauptseite öffnet den Skripte-Bereich. Mit dem +-Symbol wird ein neues Skript vom Typ "Blockly" angelegt. Nach Vergabe eines Skriptnamens öffnet sich das Bearbeitungsfenster (Abbildung 6), in dem die Einzelschritte jeweils in Form von Blöcken zusammengestellt werden. Im grau hinterleg-



Abbildung 6: Das Start-Skript im Blockly-Editor



ten Zwischenbereich befinden sich die verfügbaren Programmblöcke, thematisch gruppiert von "System" bis "Funktionen". Ein Klick auf eine Gruppe öffnet die Blöcke, die dann nach rechts in den Arbeitsbereich gezogen und verbunden werden. Sofern Werte zu ergänzen sind, öffnet ein Klick auf das entsprechende Feld im jeweiligen Block die Liste der verfügbaren Objekte zur Auswahl, etwa die Variablen wie den Lastwiderstand. Dies wird für jedes der folgenden Skripte wiederholt.

Das Skript "Start_Messung" (Abbildung 6) speichert den Messwert des Lastwiderstandes, belegt die Variablen für Maximal- und Minimalspannung mit der aktuell gemessenen Akkuspannung, löscht die Ergebnisvariablen und startet die Messung, indem die Variable Messung auf logisch wahr gesetzt wird und der Datenpunkt Schalter am Shelly-Uni ebenfalls auf, logisch wahr" gesetzt wird. Dadurch schaltet das Relais durch und die Prüflast liegt am Akku an. Bei jedem Stopp und Neustart des Abbildung 7: Skript für den Neustart einer Messung

Skripts werden die Ergebnisvariablen gelöscht und die Messung neu gestartet.

Das Skript "Restart_Messung" (Abbildung 7) setzt eine Messung fort, wenn sie aus irgendeinem Grund unterbrochen wurde, ohne die bis dahin gemessenen Werte zu löschen.

Das Skript "Unterspannung" (Abbildung 8) prüft über einen Trigger (Auslöser bei bestimmten Bedingungen), ob sich der Wert von Power_Measurement, dem Datenpunkt des Shelly-ADC (Spannungsmessung), geändert hat. Fällt der Wert unter die eingestellte Grenzspannung von 7 V, wird die Messung beendet, indem die Variable Messung auf logisch falsch gesetzt wird. Gleichzeitig wird auch der Ausgang des Shelly-Schalters auf logisch falsch gesetzt, wodurch das Relais geöffnet und damit der Akku von der Last getrennt wird.

Alle Skripte laufen zu diesem Zeitpunkt weiter. Da aber die Zustandswerte Messung und Schalter auf logisch "falsch" stehen, ist der Akkutest praktisch beendet. Eine weitere



Abbildung 8: Das Skript "Unterspannung" schaltet den Akku bei Unterspannung ab.



Abbildung 9: Das "Takten"-Skript

Entladung der Akkus findet nun nicht mehr statt.

Das Skript "Takten" (Abbildung 9) prüft alle 5 Minuten, ob gerade eine Messung läuft. Wenn ja, wird der Lastwiderstand eingeschaltet und nach 4 Minuten wieder ausgeschaltet. Dadurch wird die Entladung der Akkus im Verhältnis 4:1 (vier Minuten entladen, eine Minute aus) gesteuert. Über das Zeitprogramm (*/5 * * * *) und die Abschaltverzögerung (240 Sekunden im Skript) können auch andere Taktverhältnisse eingestellt werden.

Das Skript "Messung" (Abbildung 10) ist das eigentliche Herzstück des Akkutesters. Es prüft alle drei Sekunden, ob eine Messung vorliegt (Variable Messung) und ob aktuell eine Last an dem Akku anliegt. Beide Bedingungen müssen erfüllt sein und sind daher logisch im falls-Zweig mit und-verknüpft. Ist eine der beiden Bedingungen nicht erfüllt, geschieht nichts, sonst sieht der Ablauf wie folgt aus:

- Die gespeicherten Werte f
 ür maximale und minimale Spannung werden aktualisiert.
- Die vergangenen drei Sekunden werden als Belastungsdelta betrachtet und dieses Delta numerisch zu den Variablen der Kapazität in

mAh und mWh (Kapa_mAh, Kapa_mWh) addiert. Dazu wird die am Shelly-Uni-ADC gemessene Spannung in Volt durch den voreingestellten Widerstandswert in Ohm (Variable Lastwiderstand) dividiert, was den aktuellen Stromfluss in Ampere ergibt. Würde die Messung stündlich erfolgen, wäre dieser Wert gleichzeitig die aus dem Akku entnommene Strommenge in Amperestunden (Ah). Da die Messung jedoch alle drei Sekunden erfolgt, wird der berechnete Stromfluss durch 1200 geteilt (3600 s / 3 s = 1200) und für die Ausgabe in Milliamperestunden (mAh) noch mit 1000 multipliziert. Dies ergibt den Teiler 1.2 im Skript (siehe auch gelber Kommentarblock).

Diese Strommenge wird in der lokalen Variablen mA_deltazwischengespeichert und zu Kapa_mAh addiert bzw. mit der aktuellen Spannung multipliziert zu Kapa_mWh addiert.

Messung

So vorbereitet kann die Messung durchgeführt werden. Dazu wird der Ausgang des Shelly Uni auf false/aus gesetzt z. B. über die Shelly-Website (Abbildung 11) und ein voller Akku in die Ladeschale gelegt. Die Ruhespannung eines frisch geladenen Akkus liegt bei ca. 8 V oder darüber.

Das Skript "Start_Messung" wird neu gestartet. Es setzt dabei den Wert des Lastwiderstandes, löscht die Ergebnisvariablen, füllt U_max und U_min mit den Anfangswerten und startet die Messung am Prüfstand. Die LED leuchtet jetzt grün. Alle drei Sekunden werden die Variablen Kapa_mAh und Kapa_mWh aktualisiert (Abbildung 12). Der Wert von U_min sinkt allmählich während der Messung. Wenn die

Projekt

Messung al	lle 3 Sekunden		er e ser e s 2 - 2 - 2 - 2 -	n nan nan r Distant				त के त्वर दे के इस	Shally UNI	🍝 🗢 🔒	Time: 17:4
Zeitplan										Shellyuni-01	
falls	und 🔹	Wert vom O	bjekt ID Schalt	ing					Channel 1 😨		Ø
mache	falls mache akt	Wert • vom	Objekt ID Pow	ver Measurement	Wert •	vom Objekt ID	U_max		Channel 2 🧔		۲
	falls	Wert • vom	Objekt ID Pow	ver Measurement	Wert	vom Objekt ID	U_min	3 8 8 4 9 9 5 1 9		ADC Voltage 7.73V	
	setze mA_d	etta · auf ·	+ • • • • • • • • • • • • • • • • • • •	Vert • vom Objekt IC	Power Measureme Lastwidersta	urement			Internet & Security	Sensor URL actions	(i) Settings
	aktualisiere (Kapa_mAh] mit Kapa_mWh] mit	Wert + · · · · · · · · · · · · · · · · · · ·	vom Objekt ID letta v t vom Objekt ID Wert vom Ob Ma_delta v	(apa_mAh) m Kapa_mWh jekt ID (Power	it Verzögerung	mit Verzöger	rung 🔳	supporti	Alterco Robotics Ltd.	
					42 9 40 A		8. St. 8.	34 - E - S 40	Abbildung 11: St	euern der Schalt	tkanäle am

Abbildung 10: Skript für die Messung

Abbildung 11: Steuern der Schaltkanäle am Shelly Uni



Die Konferenz für Enterprise-JavaScript

Mainz • 7. - 8. Mai 2024

^{Jetzt} Frühbuchertickets ^{sichern!}

KH/



Workshops am 6. Mai • Barrierefreiheit • React • Angular

Veranstalter







🖕 0_userdata		1
b 0	E	Î
Akkutest		/ 1
🗋 Kapa_mAh	0,26253789	280
🗅 Kapa_mWh	2,09177129	ノ目立
Lastwiderstand	101,16	280
Messung	true	乙酮原
🗋 U_max	8,34	280
🗋 U_min	7,96	ノ言意
	 O_usserdata O Akkutest Kapa_mAh Kapa_mWh Lastwiderstand Messung U_max U_min 	O_userdata Image: Constraint of the second sec

Abbildung 12: Eingehende Messwerte leuchten kurz grün auf.

Visualisierung mit iQontrol

Mit dem Adapter iQontrol lässt sich der Status des Akkutests auf einen Blick überschauen. Ein Klick auf iQontrol in der linken Spalte der ioBroker-Hauptseite öffnet die Konfiguration. Im Reiter "Geräte" wird im Drop-down-Menü "Ansicht" der Eintrag "Home" ausgewählt. Jede Zeile entspricht dabei einer Kachel von iQontrol. Die erste Zeile mit dem Namen "Clock" kann gelöscht werden.

Die Einträge in der Zeile "iobroker uptime", (angelegt von der Demokonfiguration) werden nun geändert zu Wert: "Messung läuft", Überschrift:"Akkutester". Ein Klick auf das Stiftsymbol öffnet ein weiteres Menü. Darin den Gerätetyp auf "Schalter" ändern und den Wert des Datenpunkts über das Stiftsymbol auf die Variable Messung in den Objekten 0_userdate/0/Akkutest setzen. Weiter unten bei "Verhalten der Gerätekachel (allgemein): Aktion beim Klick auf das Icon" und "Aktion beim Klick auf die Kachel" auf "Schalten" stellen. Weiter kann bei "STATE, LEVEL und Zeitstempel" die Anzeige von "Status" auf "Status+Zeitstempel" geändert werden, damit später jedes Feld auch die Uhrzeit der letzten Änderung anzeigt.

Für die nächste Kachel wird die erste Zeile dupliziert, der Name auf "Last anliegend" geändert, der Eintrag für die Überschrift gelöscht und unter dem Stiftsymbol als Wert der Eintrag von "shelly.0.SHUNI-1/#xxxxxxxx/#1.Relay0(1). Switch" in den Objekten gewählt.

Die dritte Zeile wird von der zweiten kopiert. Name: "Spannung [V]", "Gerätetyp" auf "Wert" ändern, Wert aus dem



Auswahlmenü der Objekte: "shelly.O.SHUNI-1/#xxxxxxx/#1.ADC. Power". Die weiteren Zeilen für die Kacheln "U_max", "U_min", "Kapa_mAh" und "Kapa_mWh" werden jeweils aus der vorherigen dupliziert und lediglich Name und einzutragender Wert im Editor (Stiftsymbol) abgeändert.

Nach Speichern der Konfiguration öffnet ein Klick auf das lila unterlegte Q ein neues Browserfenster mit der Visualisierung. Ein paar Konfigurationseinstellungen reichen also aus, den eigenen Prüfstand aufs Tablet oder Smartphone zu bringen.

Die Möglichkeiten von iQontrol sind umfangreich. Es ist zum Beispiel möglich, die Skripte zu erweitern, um über iQontrol die Messung zu starten. Spannung am ADC des Shelly Uni 7V erreicht, wird die logische Variable Messung auf false gesetzt und die Messung am Prüfstand beendet. Das Resultat kann in den Variablen Kapa_mAh und Kapa_mWh abgelesen werden. Auf diese Weise werden so alle vorhandenen Kamera-Akkus nacheinander getestet.

Die Ergebnisse

Nachdem alle Akkus einmal den Prüfstand durchlaufen hatten, herrschte Klarheit: Der neu gekaufte Originalakku war der leistungsstärkste. Ein Akku eines Fremdherstellers mit mehr als 10 Jahren auf dem Buckel schied sofort aus. Aber Originalakkus, die schon einige Jahre ihren Dienst verrichteten, zeigten mit 70 % der ursprünglichen Kapazität noch gute Werte. Zwei neu gekaufte Fremdakkus brachten dagegen nur zwei Drittel der Kapazität des neuen Originals, aber da sie weniger als die Hälfte kosteten, war es trotzdem ein gutes Geschäft. Ob das auf Dauer so bleibt, wird die Zeit (und weitere Tests) zeigen. Nebenbei lieferten diese Fremdakkus nur die aufgedruckte Kapazität, wenn man sie bis auf 6 V entlädt, was aber praxisfern ist, wenn die Kamera mindestens 7 V zum Betrieb benötigt.

Weitere Tests wären nun möglich, z. B. das Verhalten bei niedrigen oder hohen Temperaturen oder bei kleineren oder größeren Entladeströmen. Aber die Zeit, die diese Tests kosten würden, nutze ich lieber zum Fotografieren.

Ausblick

Das Projekt des Akkutests mit Smart-Home-Technik ist ein reines Hobbyprojekt, zeigt aber, wie schnell ein Messaufbau mit Sensorik verbunden und wie einfach dieser in eine übergeordnete Datenerfassung, -verarbeitung und -speicherung integriert werden kann. Ein Blick in die Liste der verfügbaren Adapter unter dem Eintrag "Adapter" im ioBroker zeigt das ganze Potenzial.

Von Shelly beispielsweise gibt es nicht nur Sensoren für Kleinspannungen (Shelly Uni), sondern auch größere Sensoren bis hin zu Dreiphasenwechselstrom. Vergleichbar dazu können mit einem Siemens Sentron Messwertaufnehmer über das Protokoll Modbus RTU over TCP Leistungs- und Verbrauchsdaten im industriellen Umfeld erfasst werden. Dies ist bis zur direkten Anbindung einer Siemens S7 Prozesssteuerung möglich.

Hardwaretechnisch stellt die Installation von ioBroker als virtuelle Instanz unter Proxmox auf einem Mini-PC, NUC oder Thin Client eine moderne, stabile und wartungsfreundliche Lösung dar. Ergänzt durch influxDB/Grafana in einer weiteren VM auf der gleichen Hardware kann ein autarkes Messsystem unabhängig von bestehenden Firmennetzwerken aufgebaut werden, das auch kritischen Sicherheitsaspekten gerecht wird. —*caw*

Schnelle Diagramme: History und Flot

Für größere Projekte mit iobroker, vor allem wenn die Anzahl der darzustellenden Variablen steigt, empfiehlt sich influxDB als Datenbank und Grafana für die grafische Darstellung. Diese beiden Dickschiffe erfordern allerdings eine aufwändigere Einarbeitung. Die Adapter "history" und "flot" für Datenbank und Diagramme sind einfacher zu handhaben. Nach der Installation von History und Flot über den "Adapter" in der linken Spalte von ioBroker erscheinen die entsprechenden Anwendungen sofort unter "Instanzen". Sofern der Speicherort der History-Daten in der Voreinstellung belassen wurde, ist keine weitere Konfiguration notwendig. Die Auswahl der zu speichernden Datenpunkte erfolgt über den Eintrag "Objekte". Ein Klick auf das Zahnrad der Zeile



" ../ADC/Power" im Shelly-Zweig öffnet ein Menü, in dem die Speicherung in der History aktiviert werden kann. Auch hier müssen die Standardwerte nicht verändert werden. Damit werden nun alle von der Shelly Uni gemessenen Spannungswerte in der History-Datenbank gespeichert.

Die Konfiguration der Flot-Diagramme erfolgt über den Eintrag "Übersicht" oder als Weiterleitung des Flot-Adapters aus den "Instanzen". Ein Klick darauf öffnet die Konfiguration von Flot in einem neuen Fenster. Im Reiter "Eingangsdaten" müssen nur die ID des Datenpunktes durch Klick auf das Ordnersymbol direkt neben dem ID-Feld und die Min/Max-Bereichsgrenzen eingestellt sowie der Haken bei "Auto-update" gesetzt werden.

Ein Klick auf die Schaltfläche "Zeige im Fenster" öffnet ein neues Fenster im Browser, in dem das ausgewählte Diagramm angezeigt wird. Die Einstellmöglichkeiten von Flot sind sehr umfangreich und werden allen Anforderungen gerecht.



🚽 shop.heise.de/make-esp32cam





ATtiny statt Arduino

Wenn ein Arduino-Board für ein Hardware-Projekt zu ausladend ist und die Leistung auch gar nicht benötigt wird, greifen Sie doch mal zu einem ATtiny. Wir zeigen Ihnen, wie man diese winzigen Controller mit Microchip Studio unter Windows programmiert und nutzen dafür die UPDI-Schnittstelle der neusten Modelle.

von Florian Schäffer
Die ATtiny-Familie ist schon lange am Markt und stellt im Grunde eine abgespeckte Version der ATmega dar, die man von diversen Arduino-Boards kennt. Die internen Funktionen sind vergleichbar, aber es gibt weniger Speicher, weniger I/O-Pins (jedenfalls in der Regel) und teilweise weniger Geschwindigkeit aufgrund kleinerer Taktraten.

Für manche Projekte reicht das aber völlig aus. Wenn der Mikrocontroller nämlich nur kleine Aufgaben übernehmen soll, dann brauchen Sie viele Extras vielleicht gar nicht (Bild 1). Auch der deutlich günstigere Preis (oft weniger als ein Euro) und der geringe Platzbedarf sind manchmal wichtiger. Zudem sind die ATtiny genügsamer beim Stromverbrauch und für batteriebetriebene Projekte prädestiniert. Und wenn Sie ein eigenes Hardwareprojekt entwickeln, bei dem ein Mikrocontroller neben anderen Komponenten seinen Dienst verrichtet, spart der ATtiny Platinenplatz, Strom (bis hinab zu 0,1µA sind möglich) und Geld im Vergleich zu einem Arduino.

Also werfen wir in diesem Artikel einen Blick auf die aktuellen ATtiny-Serien und zeigen, wie Sie diese mit AVR-GCC in Microchip Studio unter Windows programmieren können. Damit lässt sich beispielsweise eine Lautstärkeregelung mithilfe eines echten Analogausgangs realisieren oder wie hier im Artikel eine 6-V-Glühbirne mit einer variablen Spannung flimmerfrei dimmen.

Familientreffen

Die Auswahl der ATtiny ist mittlerweile fast unübersichtlicher als bei den ATmega – auch aufgrund der sich immer wieder ändernden Bezeichnung und Systematik seitens des Herstellers. Um etwas Ordnung ins Chipchaos zu bringen, hat Microchip im Jahr 2016 damit begonnen, die ATtiny in Serien zu unterteilen. Seitdem sind die tinyAVR 0er-, 1er- und 2er-Serie er-

Kurzinfo

- » ATtiny kennenlernen
- » Microchip Studio und AVRDude verwenden
- » ATtiny mit dem Unified Program and
 - Debug Interface (UPDI) flashen



schienen. Im Vergleich zu den vorherigen ATtiny bieten die neuen Modelle mehr Flash und SRAM, haben standardmäßig I²C, SPI und UART, mindestens zwei 16-Bit-Timer sowie eine Geschwindigkeit von bis zu 20 MHz. Darüber hinaus teilen sich die Chips innerhalb ihrer Serie gewisse Eigenschaften, was den späteren Wechsel auf andere Modelle mit alternativer Speicher- oder Pin-Konfiguration vereinfacht, z. B. wenn das Programm anwächst. Außerdem sind die neuen Serien alle über ein Kabel per Unified Program and Debug Interface (UPDI) programmierbar, auf das wir später noch genauer eingehen. Dafür findet man die aktuellen ATtiny nicht mehr als steckbares DIP-Package, sondern mindestens in der SO-Bauform (siehe Kasten).



Bild 1: Ein ATtiny13 (links) zählt in diesem Aufbau lediglich die Impulse an einem Eingang und variiert abhängig von deren Anzahl die Zeitspanne für ein High-Signal. Einen größeren Mikrocontroller benötigt man dafür nicht.



Bild 2: An der Bezeichnung der neuen ATtiny erkennt man die Serie und weitere Spezifikationen. Je nach Pinanzahl gibt es die Chips in unterschiedlichen Bauformen, DIP ist allerdings nicht dabei.





Bild 4: Die ATtiny der 1er-Serie generieren ein echtes analoges Ausgangssignal (hier in Schritten von $5 \times 16,7$ mV).

Bild 3: Die ATtiny der 1er-Serie haben mindestens einen DAC. Ab den Modellen mit 16kB-Flash (ATtiny1614) besitzen sie sogar drei DACs.

Gehäusebauform

Das Dual in-line package (DIP oder DIL) ist mit seinen Anschlussbeinchen für Hobbyentwickler sicherlich das angenehmste, weil die Beine gut in Breadboards passen und sich einfach löten lassen. Hauptabnehmer von Chips ist aber die Industrie und die mag es gerne klein, z. B. im SMD-Format (Surface Mounted Device), da es sich besser industriell verarbeiten lässt. Deshalb stirbt die DIL-Größe langsam aus. Bei den ATtiny gibt es nur noch ältere Typen wie den ATtiny85 oder ATtiny13 mit Beinchen.

Die Größen von SMDs haben sich leider etwas unübersichtlich entwickelt. So gibt es Small Outline (SO), was von einigen Herstellern auch als SO-IC oder SOIC genannt wird. Bei Small Outline Package (SOP) sind die Gehäuse etwas breiter, sodass die zwei Pinreihen etwas weiter entfernt sind, was auch als SO-W bekannt ist. Der Abstand zwischen zwei Pins beträgt jeweils 1,27 mm. Haarig im wahrsten Sinne des Wortes wird es bei SSOP (Shrink Small Outline Package), denn hier beträgt der Pinabstand oft nur noch 0,65 mm. Es geht aber auch noch kleiner: Chips im Format QFN (Quad Flat No Leads Package) oder MLF (Micro Lead Frame) kommen etwa ganz ohne Anschluss-Pins aus. Da hört der Lötspaß echt auf.

Zu den ATtiny in SOP-Bauform gehören beispielsweise der ATtiny1614, ATtiny414,



Bild 5: SMD-Adapterplatinen mit aufgelötetem S08-IC (links) und S0P16 rechts. Auf der Rückseite befinden sich Lötpads für SSOP mit 0,65 mm Pitchabstand.

ATtiny412 und der ATtiny212. Damit man diese auch auf einem Breadboard in der Prototypen-Entwicklung benutzen kann, gibt es kleine Adapterplatinen (Bild 5). Auf diese werden die Chips gelötet und alle Anschlüsse führen zu den Seiten an Stiftleisten im Raster von 2,54 mm. Einmal müssen Sie also zum Lötkolben greifen. Achten Sie aber sorgfältig darauf, wie herum das SMD-Bauteil aufgelötet werden muss, damit Pin 1 vom IC auch mit dem (meistens rechteckigen) Pin 1 der Stiftleiste verbunden ist. Folgen Sie dazu der Leiterbahn oder messen Sie mit einem Durchgangsprüfer nach.

Grundsätzlich sollte man einen sehr genauen Blick ins Datenblatt werfen und sich die Package-Maßzeichnungen ansehen, denn bei Microchip wird alles als SO bezeichnet und erst im Detail sieht man, ob das Gehäuse beispielsweise 3,9 mm oder 7,5 mm breit ist. Die Adapterplatinen sind in der Regel so gestaltet, dass beide Formen Platz finden, wozu die Lötflächen extra lang sind. Ob es sich um ein Modell der neuen Serien handelt, erkennt man an der Chip-Bezeichnung (Bild 2), die zudem die Größe des Flash-Speichers und die Anzahl der Pins angibt. Man muss aber genau hinschauen, denn während ein ATtiny827 zur 2er-Serie gehört, ist der ATtiny828 ein viel älteres Modell.

Da es in diesem Artikel zu weit führen würde, auf die Eigenschaften und Unterschiede aller neuen ATtiny einzugehen, finden Sie dazu Links in der Kurzinfo. Für dieses Projekt habe ich aber bewusst einen Chip der 1er-Serie gewählt.

DAC on Board

Eine Besonderheit der ATtiny-1er-Serie sind die integrierten Digital-Analog-Wandler (DAC: Digital Analog Converter), von denen die größeren Modelle bis zu drei beherbergen (Bild 3). Die andere Richtung konnten AVR-Chips schon immer: aus einem externen analogen einen digitalen Wert erzeugen und verarbeiten (ADC: Analog Digital Converter). Aber ein DAC ist bei den AVRs (relativ) neu und ein echtes Highlight.

"Moment mal", werden Sie jetzt eventuell sagen. "Der Arduino hat doch sogar sechs Analogausgänge und nicht nur einen. Was soll daran so toll sein?". Tatsächlich tun die Analogausgänge am Arduino aber nur so, als ob sie welche wären. Der darauf sitzende ATmega328 bietet diese Funktion nicht wirklich. Stattdessen erzeugt die Arduino-Software ein PWM-Signal (Pulsweiten-Modulation), das wie ein analoges Signal wirken kann, aber keines ist: Der Signalpegel ist immer 0 V oder 5 V, wechselt nur sehr schnell.

Ein vollwertiger DAC kann hingegen wirklich eine beliebige Spannung zwischen 0 V und der Betriebsspannung (oft 5 V) erzeugen. Der ATtiny kann mithilfe eines internen Widerstandsnetzwerks je nach Konfiguration bis zu 4,3 V generieren (Bild 4).

AVR-GCC statt Arduino

Einzelne ATtiny-Chips lassen sie sich nicht ganz so komfortabel mit der Arduino IDE programmieren, als würden sie direkt auf einem Arduino UNO oder Nano sitzen. Abgesehen von der fehlenden USB-Schnittstelle müsste man den Controller nämlich zuvor mit einem passenden Bootloader präparieren, der im Speicher der ganz kleinen Typen keinen Platz hat. Zudem verbietet es die geringe Anzahl der Pins in den meisten Fällen, permanent zwei davon allein für die serielle Schnittstelle zu blockieren. Daher schauen wir uns in diesem Artikel auch die offizielle Entwicklungsumgebung für AVRs an, mit der sich die Chips direkt flashen lassen: Microchip Studio für Windows (ehemals AVR Studio).

Wenn Sie bisher nur für Arduinos programmiert haben, wird es aber eventuell zu einem kleinen Kulturschock kommen: Bekannte Befehle wie pinMode(), digitalWrite() und Serial.print() gibt es in Microchip Studio nicht. Auch viele Bibliotheken, die das Leben so vereinfachen können, sind nicht verfügbar.

Das hört sich erst mal nicht gerade nach einem Gewinn an. Wenn Sie sich mit der Welt

hinter Arduino befassen wollen, lohnt es sich aber. Irgendjemand hat nämlich Funktionen wie pinMode() und die Bibliotheken erstellt - und zwar in AVR-GCC. Das ist ein freier C-Compiler, der sich darum kümmert, aus dem Quellcode die Binärdatei zu erstellen, die auf den Mikrocontroller übertragen und dort ausgeführt wird. Um also unter die Haube zu schauen und eventuell selbst etwas Neues oder Besseres zu entwickeln (etwa für einen neuen, bisher nicht unterstützten Sensor oder ein Display), müssen Sie wissen, was dahinter steckt und wie es funktioniert. Viele Programmierer wollen sich selbst mit dem Datenblatt eines Bauteils beschäftigen, es verstehen und ihre Programmierkenntnisse erproben oder verbessern - oder sich mit anderen messen. Immerhin gibt es auch für Arduino oft mehrere Bibliotheken für ein und dasselbe Bauteil, weil jeder Entwickler andere Wünsche oder Fähigkeiten hat.

Natürlich existieren für die meisten Anwendungen, die man mit Microchip Studio umsetzen will, auch bereits Lösungen von anderen Nutzern. Denn schon lange, bevor es sie für Arduino gab, haben sich Programmierer daran gesetzt, Bibliotheken für die ATtiny zu entwickeln. Diese müssen Sie allerdings selbst im Web suchen und sie lassen sich nicht so spielerisch einbinden wie mit der Arduino IDE.

Der Code, den Sie direkt für AVR-GCC erstellen, ist immer sparsamer beim Speicherbedarf und manchmal sogar schneller. Denn obwohl sich die Arduino-Funktionen zur I/O-Steuerung usw. einfach nutzen lassen, blähen sie den Code

AVRs mit UPDI programmieren

ATmegas wie der auf dem Arduino UNO (bis Rev 3) werden üblicherweise über die ISP-Schnittstelle (In-System-Programmierung) und mit einem extra Programmieradapter beschrieben (geflasht oder gebrannt), wofür dieser vier Leitungen benötigt. Danach nutzt das Board ein Verfahren über einen Bootloader per USB, weshalb Sie etwa keinen Adapter benötigen, um einen Arduino mit Programmen zu bespielen.

Inzwischen gibt es alternative Methoden und die mehr oder weniger neuste ist das UPDI. Hierbei wird nicht einmal eine separate Datenleitung benutzt, sondern der Reset-Anschluss des Chips, sodass man keinerlei Pins verschwendet. Ein ausgeklügeltes Protokoll sorgt dafür, dass der Prozessor in den Programmiermodus wechselt und auch die Baudrate des Kommunikationssignals erkennt. Nach dem Verbindungsaufbau kann die einzelne Leitung für den bidirektionalen Datenaustausch genutzt werden. Das spart Platz, weil man keinen Pinheader für den Programmieradapter benötigt.

Der Clou ist zudem, dass kein spezieller Programmieradapter notwendig ist, sondern ein einfacher USB-Seriell-Wandler (auch als FTDI-Adapter bezeichnet) ausreicht, wie er bei fast jedem Maker vorhanden sein dürfte oder der für ein paar Euros leicht beschafft ist. Benutzen Sie sicherheitshalber lieber einen Adapter mit echtem FT232-Chip, auch wenn der etwas teurer ist als die billigen Nachbauten mit CH340/CH9340, die eher Treiberprobleme bereiten.

Weil die beiden seriellen Datenleitungen verbunden sind, benötigt man eine Ent-

kopplung. Hierfür kann man einfach einen Widerstand mit 4,7 kΩ nutzen. Das bezeichnet man oft als PyUPDI-Schaltung, weil es vermutlich das erste Mal zusammen mit der Python UPDI-Programmiersoftware gezeigt wurde. Bei den Links (siehe URL in der Kurzinfo) finden Sie eine Webseite zu SerialUPDI, auf der weitere Beschaltungen (auch mit Dioden) gezeigt werden. Dort wird behauptet, dass die meisten USB-Adapter einen internen Widerstand in der Tx-Leitung besitzen und deshalb eine Diode besser sei. Dem Autor sind solche Interfaces allerdings noch nie begegnet. Sollten Sie mit dem Widerstand Probleme haben (ein Schaden entsteht nicht), probieren Sie eine der gezeigten Alternativen. Nach dem Flashen können Sie den Widerstand R3 und den USB-Adapter entfernen.

auf und wenn Flash-Speicher wie bei den ganz kleinen ATtiny knapp ist, dann zählt irgendwann jedes Byte. Der Unterschied ist oft nicht riesig, aber eben doch vorhanden. Auch gibt es Funktionen, die sich gegenseitig behindern. Ein typisches Beispiel sind die Timer, die teilweise von der Arduino-Software für die Erzeugung von PWM-Signalen blockiert werden. Das ist kaum dokumentiert und führt immer wieder zu überraschenden Seiten-



Bild 8: Breadboard-Ansicht der Schaltung mit dem SO8-Adapter für den ATtiny212. Die Anschlüsse an Ihrem FTDI-Adapter können anders angeordnet sein.

Spannungsversorgung

Der Schaltplan zeigt eine externe 5-V-Versorgung. Diese ist nur notwendig, wenn kein FTDI-Adapter angeschlossen ist. Ansonsten kann der USB-Adapter den minimalen Strom für den ATtiny und die LED liefern und die zusätzlichen 5 V sind nicht notwendig. Sie sollten auch immer nur eine der beiden 5-V-Quellen anschließen, damit kein Strom rückwärts in eine der Spannungsquellen fließen kann, wenn dort eventuell eine Diode fehlt. effekten, wenn man als Anwender unbedarft damit programmiert.

Allerdings muss man bei AVR-GCC etwas mehr auf seine Schreibweise achten. Das hat aber auch Vorteile, wie man bei der Variablen-Deklaration sieht: In Arduino reicht es, einen Integer mit int zahl; zu deklarieren. Damit ist aber nicht genau gesagt, wie groß der Integer ist - Sie müssen dafür erst in die Dokumentation schauen (es sind 16 Bit). Anders sieht es mit uint16_t zah1; aus. Jetzt ist klar, dass es sich um einen vorzeichenlosen (unsigned) Integer mit 16 Bit handelt. So kann man Überlauffehler vorab erkennen. Wenn Sie sich auf einen Integer mit nur 8 Bit beschränken (int8_t), sparen Sie sogar Speicherplatz und erhöhen die Ausführungsgeschwindigkeit bei manchen Operationen (immerhin handelt es sich bei den AVRs um 8-Bitter). Diese exakte Deklaration ist übrigens optional auch in der Arduino IDE möglich, weil sie auf AVG-GCC aufsetzt.

Microchip Studio Features

Ein weiterer Vorteil von Microchip Studio ist, dass es wirklich alle Prozessoren von Microchip unterstützt und sie nicht erst über irgendwelche Quellen nachgeladen werden müssen. Ein direkter Zugriff auf die Datenblätter rundet das Angebot zudem ab.

Und dann sind da natürlich noch die Dinge, die die Arduino IDE in dieser Form nicht kann: Simulation und Debugging. Microchip Studio kann Code ausführen und die Ergebnisse anzeigen, ohne dass dazu ein physischer Mikrocontroller vorhanden sein muss. Natürlich bewegen sich dann keine Servos, aber die Ausgänge zeigen Signalpegel etc. an. Das kann schon sehr hilfreich sein.

Beim In-System-Debugging ist der Controller mit der IDE verbunden und beide tauschen sich aus. So kann man in der Software Haltepunkte (Breakpoints) definieren, bei denen die Programmausführung wirklich stehen bleibt, um zum Beispiel zu prüfen, ob das angeschlossene Relais wirklich wie gewünscht geschaltet hat. Aus dem Mikrocontroller kann man zudem jederzeit die Register auslesen, um zu sehen, was genau passiert. Überläufe von Variablen, sich nicht verändernde Variablen oder auch der Zustand der I/O-Pins lassen sich so einfach finden und analysieren – bei großen Projekten eine enorme Arbeitserleichterung.

Im Rahmen dieses Artikels können wir darauf aus Platzgründen nicht eingehen, werden uns aber möglicherweise in einem späteren Beitrag damit beschäftigen. Zudem sind die ATtiny hierfür nicht vollumfänglich geeignet.

Wenn Sie auf die Funktionen der Arduino-Welt nicht verzichten wollen und einfach nur einen leistungsfähigen Editor suchen, dann können Sie auch in Microchip Studio Arduino-Projekte erstellen – vorausgesetzt, die Arduino IDE ist bereits installiert.

Elektronik verbinden

Für die erste Kontaktaufnahme wird ein einfaches Projekt herhalten: Ein ATtiny212 soll über den DAC eine sich verändernde Spannung ausgeben, solange ein Taster gedrückt wird (Bild 7 und 8). Damit lernen Sie gleich kennen, wie digitale Signale eingelesen und ausgegeben werden und Sie den DAC steuern.

Wie bei jedem Mikrocontroller ist es sehr empfehlenswert, die Spannungsversorgung mit einem Kondensator (100 nF) nahe am Bauteil zu stabilisieren. Dies ist die einzig notwendige externe Beschaltung. In der Praxis (oder bei Batteriebetrieb) können Sie darauf auch oft verzichten, aber dann kann es zu merkwürdigen Fehlfunktionen kommen – es lohnt sich nicht, hier zu sparen.

Die LED dient wie bei allen Einstiegsprojekten als "hello world"-Anzeige und bietet die Gelegenheit, im Code zu zeigen, wie digitale Ausgänge angesprochen werden. Für den Eingang des Tasters wird der interne Pull-Up aktiviert, sodass der Taster Low-aktiv arbeitet.

Der Analogausgang am ATtiny darf nur hochohmig belastet werden. Das bedeutet, er kann so gut wie keine Leistung liefern – nicht einmal für eine LED. Mit einem Multimeter oder Oszilloskop könnten Sie die Spannung messen, aber das wäre nun doch etwas langweilig. Aus diesem Grund kommt noch ein Transistor (BC337/BC547 o.Ä.) als

Microchip Studio for AVR and SAM Devices
Microchip Studio
Release Notes
Upgrade to PRO license to unlock the full potential of MPLAB & XCB C compiler's advanced-level optimizations. More information
Back Install Cancel

Bild 9: Um ATtiny zu programmieren, reicht es, wenn Sie die AVR-Unterstützung installieren.

primitiver Treiber zum Einsatz. Dieser steuert eine kleine Glühbirne (E10-Fassung), sodass die analoge Ausgangsspannung den Transistor schrittweise immer weiter durchschaltet und die Lampe dimmt. Das klappt natürlich nur in gewissen Grenzen und ist nicht optimal, weil hier nicht auf die Transistor-Kennlinie geachtet wird etc. Aus diesem Bild 10: Installieren Sie auch den XC8 C Compiler, um später lästige Hinweise zu vermeiden.

Grund schadet die Spannung von etwa 12 V der 6-V-Glühbirne auch nicht.

Microchip Studio installieren

Um das Beispielprojekt in Betrieb zu nehmen, müssen wir als Nächstes Microchip Studio installieren. Den Link zur Software finden Sie in

AVRDude Arguments

-c serialupdi -p t212 -P COM4 -V -e -U flash:w:"\$(TargetDir)\$(TargetName).hex":a

Menu contents.					
AVRDude			Add		
			Delete		
			Movel		
			Move Dov	wn	
Title	AVRDude				
Command:	C:\tools\avrdude\avrdude.exe -c serialupdi -p t212 -P COM3 -V -e -U flash:w:"\$				
Arguments:					
Initial directory:				•	
Use Output window	Pr	ompt for argume	nts		
Treat output as Unico	e 🗹 CI	ose on exit			
	OK	Cancel	Apply	,	



Bild 11: Finden Sie heraus, an welchem COM-Port der FTDI-Adapter angeschlossen ist.

Bild 12: Über "External Tools" fügen Sie AVRDude zu Microchip Studio hinzu.



Bild 13: Microchip Studio mit dem Grundgerüst eines neuen Projekts

main.c #define F_CPU 3333333UL //3,3 MHz #include <avr/io.h> #include <util/delay.h> int main(void) { PORTA.DIRSET = (1 << 3);PORTA.DIRCLR = (1 << 2)PORTA.PIN2CTRL = PORT_PULLUPEN_bm; VREF.CTRLA |= VREF_DAC0REFSEL_4V34_gc; _delay_us(25); PORTA.PIN6CTRL &= ~PORT_ISC_gm; PORTA.PIN6CTRL |= PORT_ISC_INPUT_DISABLE_gc; PORTA.PIN6CTRL &= ~PORT_PULLUPEN_bm; DAC0.CTRLA = DAC_ENABLE_bm | DAC_OUTEN_bm | DAC_RUNSTDBY_bm; uint8 t DACwert = 0: while (1) { if (PORTA.IN & (1 << 2)) { PORTA.OUTCLR |= (1 < < 3);_delay_ms(150); } else { PORTA.OUTSET |= (1 < < 3);DACwert += 5 DAC0.DATA = DACwert;delay_ms(50); PORTA.OUTCLR |= (1 < < 3);_delay_ms(50); } } }

der Kurzinfo, Wählen Sie entweder den Komplett-Download für eine Installation von Festplatte oder den Web-Installer, der die notwendigen Dateien während der Installation nachlädt. Während des Installationsprozesses werden Sie gefragt, welche Architekturen unterstützt werden sollen. Für das Beispiel und alle AVR-Controller genügt es, wenn Sie AVR auswählen (Bild 9). Wenn Sie ausreichend Speicherplatz haben, können Sie die erweiterten Funktionen und Beispiele installieren. Sie sind aber nicht erforderlich. Installieren Sie am besten auch den "XC8 C Compiler" (Bild 10). Diesen benötigen Sie zwar nicht, da wir uns auf AVR-GCC beschränken, der immer installiert wird. Allerdings kommt es später sonst bei jedem Start von Microchip Studio zu einer lästigen, aber unbedeutenden Warnung, die man wegdrücken muss. Folgen Sie den nächsten Schritten und installieren Sie alle empfohlenen Treiber, weil sie für spätere Projekte sinnvoll sein könnten. Schließen Sie zum Schluss das Dialogfenster, ohne Microchip Studio zu starten.

AVRDude einbinden

Um eine kompilierte Programmdatei (als HEX-File vorliegend) mit dem FTDI-Adapter auf den ATtiny zu übertragen, benötigt man ein zusätzliches Programm. Profis verwenden dafür schon lange AVRDude. Wie man es zu Microchip Studio hinzufügt, zeige ich Ihnen hier.

Zuerst einmal müssen Sie den virtuellen COM-Port für Ihren USB-Seriell-Adapter ermitteln (Bild 11). Stecken Sie ihn an einen USB-Port und öffnen Sie dann den Geräte-Manager (z. B. mit Kurzbefehl Windows+x und dann im Menü auswählen). Im Abschnitt Anschlüsse finden Sie den Eintrag, USB Serial Port" mit der Angabe des COM-Ports in Klammern. Bedenken Sie, dass sich die Portzuordnung ändert, wenn Sie später einen anderen Adapter oder USB-Port benutzen.

Laden Sie danach das für Ihr Windows passende ZIP-Archiv, z.B. avrdude-v7.2-windowsx64.zip von der AVRDude-Website herunter (siehe Link in der Kurzinfo) und entpacken Sie den Inhalt des Archivs in einen beliebigen Ordner. Das Programm benötigt keine weitere Installation.

Starten Sie jetzt Microchip Studio und klicken Sie auf den Menüeintrag "Tools/External Tools". Wählen Sie bei Command den Pfad zu avrdude.exe auf Ihrem System aus. Tragen Sie die Werte in die Felder ein, wie im Bild 12 zu sehen und aktivieren Sie die Option "Use Output Window". Klicken Sie auf Apply und dann auf OK. Den COM-Port müssen Sie an Ihre Gegebenheiten anpassen. Beim Zusammenstellen der Zeile in Arguments hilft Ihnen die Auswahl, die erscheint, wenn Sie auf den kleinen Pfeil nach rechts neben der Eingabe-

Workshop



Bild 14: Pinout des ATtiny212

zeile klicken. Die vollständige Zeile muss wie im Kasten gezeigt lauten.

Der Code

Öffnen Sie als Nächstes das Beispielprogramm in Microchip Studio. Die Datei attiny_make. atsln befindet sich in dem Projektordner, den Sie aus dem GitHub-Repository des Artikels herunterladen können (siehe Link in der Kurzinfo). Öffnet sich nach dem Laden des Projekts nicht automatisch die Codeansicht (Bild 13), klicken Sie am rechten Rand im "Solution Explorer" auf main.c. Der Code ist in der Datei ausführlich kommentiert, damit Sie ihn leichter nachvollziehen können.

Wenn Sie sich den Quellcode anschauen, werden Sie zahlreiche Unterschiede zur Programmierung mit der Arduino IDE erkennen. Markant ist, dass bei den meisten Programmen für den GCC die Bibliothek avr/io.h eingebunden wird. Diese ist dafür zuständig, dass die benutzten Referenzen den speziellen Registern des jeweils benutzten AVRs zugewiesen wurden.

Je nach Mikrocontroller gibt es zwei verschiedene Techniken, wie die I/O-Pins als Ein- und Ausgänge etc. deklariert werden. Die neueren ATtiny nutzen die hier benutzte Syntax mit DIRSET und DIRCLR: Eine logische Eins in den einzelnen Bits des Registers setzt den jeweiligen Pin als Eingang bzw. Ausgang. Bit 0 entspricht dem Pin 0 des entsprechenden Ports und Bit 7 steht folglich für Pin 7.

Für den DAC muss man die Referenzspannung einstellen, aus der sich die möglichen Spannungsschritte ergeben. Für das Beispiel nutzen wir die interne Spannung von 4,3 V, womit sich 255 mögliche Schritte à ca. 16,7 mV ergeben. Der DAC benötigt noch ein paar Registereinträge, um bereit zu sein und anschließend wird der mit DAC0.DATA übergebene Byte-Wert als äquivalente Spannung am Ausgang PA6 erzeugt (siehe Listing).

Solange Sie den Taster drücken, wird der DAC-Wert kontinuierlich hochgezählt, bis die 8-Bit-Variable überläuft und wieder bei 0 beginnt. Dadurch wird die Lampe schrittweise heller und dann wieder ausgeschaltet. Zur Kontrolle blinkt die LED.

Kompilieren und übertragen

Um den C-Code zu kompilieren, brauchen Sie nur "Build/Build Solution" zu wählen oder F7 zu drücken. Im unteren Fensterbereich erscheinen die Ausgaben des Compilers und eventuelle Fehlermeldungen. Der Kompiliervorgang dauert nach dem ersten Mal deutlich weniger lang, weil nicht alle Abhängigkeiten immer neu übersetzt werden, sondern nur die geänderten Dateien.

Haben Sie AVRDude eingerichtet und den FTDI-Adapter wie im Schaltplan gezeigt angeschlossen, dann genügt ein Aufruf von "Tools/AVRDude", um die erzeugte Hex-Datei zu übertragen. Im unteren Fensterbereich sehen Sie den Fortschritt. Kompiliert belegt der Code nur 212 Byte und damit 10,4 Prozent des Speichers eines ATtiny212. Mit der Arduino IDE beansprucht ein vergleichbares Projekt mit PWM als Analogsignal etwa 1310 Byte.

Nachdem Sie ihr Programm getestet haben, bekommen Sie hoffentlich Lust, weiter zu experimentieren. Modifizieren Sie das Verhalten des Tasters, wie er die Helligkeit ändert oder lassen Sie die LED permanent leuchten, solange der Taster gedrückt wird. Fortan können Sie andere Bastler damit überraschen, wie klein die Hardware von Mikrocontroller-Projekten ausfallen kann. — akf



Mit Künstlicher Intelligenz produktiv arbeiten



- ► KI-Programme anwenden
- ► Grenzen der Sprachmodelle erkennen
- Was Unternehmen rechtlich beachten müssen
- Die eigene Sprach-Kl betreiben
- Wo KI-Assistenten tatsächlich helfen
- Wie KI Schule und Arbeit verändert



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



Making unter Palmen

Rauchende Köpfe und surrende Technik: Studierende von über 60 Universitäten haben sich auf Gran Canaria beim 7. Smart Green Island Makeathon getroffen. Dort versuchten sie, in 4 Tagen verschiedene Maker-Projekte zum Thema Umwelt im Team umzusetzen. Die Make-Redaktion war vor Ort und hat sich die spannenden Projekte angesehen.

von Daniel Schwabe



om 28. Februar bis zum 2. März fand auf dem Messegelände Infecar Feria de Gran Canaria in Las Palmas der 7. Smart Green Island Makeathon statt. Bei diesen Events haben Studierende (und deren Professoren) aus der ganzen Welt die Möglichkeit, unter Anleitung und mit Hilfestellung von Industriefirmen ein technisches Projekt im Team umzusetzen. Das haben wir uns einmal genauer angeschaut.

Die jungen Leute müssen sich in Gruppen zusammenfinden, das Projekt planen, realisieren und jeden Abend ihren Fortschritt präsentieren. Das ist sportlich, aber alle Teams waren hoch motiviert und ausnahmslos alle haben unserem Redakteur vor Ort gesagt, dass sie keine Bedenken hätten, ihr Projekt in dieser Zeitspanne fertigzustellen.

Veranstaltungsvater und Geschäftsführer von ITQ, Dr. Rainer Stetter, sieht die Veranstaltung als Möglichkeit für Studierende, aus dem theoretischen Unialltag herauszukommen und direkt praktisch zu lernen. In einem Gespräch hat er begeistert erzählt, wie dieses Event jedes Jahr größer wird und immer mehr Studierende teilnehmen. "Bald reicht die Halle nicht mehr" sagte er begeistert, während er auf die Studenten um sich herum zeigte.

Grün, grün, grün sind alle meine Projekte

Wie der Name der Veranstaltung andeutet, drehen sich die Themen um grüne und nachhaltige Technologien. Das Sparen von Ressourcen und der Umweltschutz sind zentral für alle Projekte. Neben dieser grundlegenden Ausrichtung ist Automatisierung ein großes Thema und die Verbindung beider Bereiche führte zu sehr interessanten Ideen.

Unter dem Namen "Trash Terminators" hat sich ein Team der Herausforderung gestellt, ein autonom fahrendes Auto (im Endeffekt ein großes Skateboard) mit einem Roboterarm zu versehen. Dieser Arm erkennt mithilfe von KI herumliegenden Müll über einen Live-Video-Feed und sammelt ihn vom Boden auf, klassifiziert ihn und trennt den Abfall dann direkt.

Generell waren Roboterarme und KI-Anwendungen auf dem Event sehr beliebt. Vier Teams haben an einer automatisierten und



Dieser Traktor soll am Ende des Makeathons elektrisch fahren? Da bin ich aber mal gespannt.

Ein Symbolbild für den Makeathon: Roboterarme und grüne Pflänzchen. Das hier ist der Prototyp für eine aeroponische Farm.

smarten aeroponischen Farm gearbeitet. Auch hier kamen wieder eine Kamera und ein Roboterarm zum Einsatz. Der Arm hebt die Pflanze aus einem Regal auf ein Fließband und die Kamera kann dort den Status der Pflanze erkennen – ob sie genug Wasser hat, ob sie von Schädlingen befallen ist und wie entsprechend gehandelt werden muss. Der Arm platziert die Pflanzen am Ende wieder an ihren vorgesehenen Orten. Wieder andere haben mit Roboterarm plus KI diverse Kabel sortiert und andere mit einem Roboterarm Einkäufe aus einem Regal entnehmen lassen.

Ein Team, das an der Automatisierung von Bananenplantagen gearbeitet hat, konnte sogar vor Ort an einer echten, nahegelegenen Bananenplantage arbeiten, um das Projekt so nahe am echten Use Case zu entwickeln wie möglich.

Maker-Paradise

Neben den oben genannten Armen und Kameras, hatten die Teilnehmer für ihre Projekte Zugriff auf alle Werkzeuge, die das Maker-Herz begehrt. In einem eigenen Bereich war alles für Holz- und Metallverarbeitung bereitgestellt. Das konnte man auch hören, denn es wurde fleißig gehämmert, gesägt und gefeilt.

Des Makers liebstes Hilfsmittel, der 3D-Drucker, fehlte natürlich auch nicht. Eine ganze Farm – ein Rudel möchte man schon sagen – druckte den tüftelnden Makern alles, was sie zur Rettung der Umwelt benötigten. Alle anderen Teile wie Kabel, Mikrocontroller, Holz und alle möglichen Bauteile haben entweder die anwesenden Firmen mitgebracht, oder wurden von den Teams bestellt und im lokalen Baumarkt besorgt.



Von den Meistern lernen

Fachleute der betreuenden Firmen standen nicht nur beim Maken mit Rat und Tat zur Seite. Während des Events wurden Workshops angeboten, die direkt von den Unternehmen gehalten wurden. Dort konnten sich die Teil-



Pfiffig: Wer einen automatischen Müllsammler baut, muss sich um die Ordnung am Projekttisch keine Sorgen machen.



Natürlich dürfen auf einem Maker-Event die 3D-Drucker nicht fehlen.



Stifte raus: In den Workshops gab es interessante Maker-Themen.



Alle Teams haben ordentlich rangeklotzt und in kurzer Zeit ein vorzeigbares Projekt auf Beine – oder Räder – gestellt.



Denkt an die Kinder: Mit Lego in den Umgang mit Technik einsteigen.

nehmer dann in Maker-Themen wie 3D-Druck, die Nutzung von MQTT oder Schrittmotoren einweisen lassen. Dieses neue Wissen konnten sie anschließend direkt in den Projekten anwenden.

Auch etwas für die Kleinen

Neben dem großen Hauptevent gab es auch was für Nachwuchs-Maker. Mit Lego-Mindstorm-Robotern und anderen Technikbaukästen wurden Kinder von lokalen Schulen an den schaffenden Umgang mit Technik herangeführt. Ältere Kinder und Jugendliche konnten sich dann an den allseits beliebten Roboterarmen ausprobieren und lernen, was man mit ihnen Tolles machen kann.

Die letzten Schrauben festziehen

Am Ende der Tüfteltage hielten alle Gruppen eine Präsentation, um ihren Fortschritt vorzustellen. Was geklappt hat, was nicht geklappt hat, wo es Rückschläge gab und was für den nächsten Tag geplant ist. Und auch das E-Traktor-Team hat vorgestellt, was sie geschafft haben.

Wer Maken kann, kann auch feiern

Die oberste Regel beim Tüfteln und Bauen ist ja, dass alles mit einem vollen Magen besser klappt. Neben einem üppigen Buffet zur Mittagszeit und mehr oder weniger gesunden Snacks über den ganzen Tag verteilt, gab es natürlich zum Abschluss des Tages, nach den absolvierten Vorstellungen, eine sehr verdiente Feier!

Hands-on?

Über ein Event wie diesen Makeathon zu lesen, ist interessant, Bilder der Projekte zu sehen noch besser. Am interessantesten ist es aber, die Ergebnisse in Aktion zu sehen. Dazu gibt es auf der Maker Faire Hannover am 17. und 18. August 2024 die Gelegenheit. So wie die Make auf dem Makeathon vor Ort war, wird der Veranstalter des Smart Green Island Makeathons, das Unternehmen ITQ, in Hannover mit einem Stand vertreten sein, an dem einige der erarbeiteten Projekte ausgestellt werden. — das



Makeathon und Make Magazin passen gut zusammen. Unser Redakteur für Themen über 23°C (Daniel Schwabe) war vor Ort.

Übersicht über die Projekte

- » Batteriezellen recyceln
- » Smart-Aeroponic-Farm
- » Kabelmanagement mit Roboterarm
- » Smart-Farming
- » Automatisierungen mit Roboterarmen
- » Autonomer Müllsammler und -sortierer » Augmented-Reality-Unterstützung für
- Maintenance-Arbeit
- » Optimierung der Bewegungsabläufe von Roboterarmen
- » Das Erkennen von kaputten Lüftern per Machine-Learning
- » Umbau eines Traktors auf Elektroantrieb
- » Wissenschaftliche Baukästen für Kinder
- » Hydrogen-Detektor
- » Überwachung des Bierbrauprozesses
- » Ein weiterer Müllsammel-Roboter
- » loT in Gebäuden
- » Unkrauterkennung



Schlummertrunk nach einem produktiven Tag



Mit Elektromotor, Batterien, Solarzelle und H₂O Range Extender sieht dieser Traktor schon sehr viel grüner aus.



Maken macht hungrig. Schnell einen Burger schnappen und weiter tüfteln!

USB Sleuth selbstgebauter USB-Kabeltester

Auf Basis eines ARM-Mikrocontrollers STM32C031K6T kann man sich selber einen Tester für USB-Kabel bauen.

von Daniel Schwabe



it einem ARM-Mikrocontroller vom Typ STM32C031K6T und einem selbst designten PCB hat Maker Spencer Maroukis einen USB-Kabeltester entwickelt.

Damit kann man beispielsweise reine Stromkabel von Datenkabeln unterscheiden. Außerdem können Kabel darauf überprüft werden, ob sie für bestimmte Verwendungszwecke geeignet sind: etwa, ob es sich um ein USB-on-the-Go (OTG) oder ein aktives USB-Kabel handelt und welche Geschwindigkeiten damit erreicht werden können.

Diese Tests können für folgende Kabel durchgeführt werden: USB-C auf USB-C, USB-C auf USB-A (Desktop-USB-Anschluss), USB-C auf Mikro-USB, USB-C auf Mini-USB. Es können nur Stecker-zu-Stecker-Kabel getestet werden, keine Verlängerungen.

Über LEDs werden die Testergebnisse bzw. technischen Spezifikationen der Kabel angezeigt. Aufgeschlüsselt werden die technischen Spezifikationen auf dem PCB nach folgenden Gruppen: USB 2.0, USB 3.0, USB 3.1, Orientierung, Widerstände. Beispielsweise besteht die USB-2.0-Sektion aus einem Test der beiden Datenleitungen, der Ground- und der VBUS-Leitung. Jeder dieser Aspekte wird in einer LED abgebildet, die bei bestandenem Test leuchtet. Bei USB-C-Kabeln ist in diesem Fall die Orientierung der Datenleitungen wichtig.

Bei einem Test wird ein Kabel an zwei Seiten des Testers angeschlossen und der Tester eingeschaltet. Dieser startet dann automatisch alle Tests auf diesem Kabel und gibt die Ergebnisse dann mit den oben beschriebenen LEDs aus.

Neben diesen Testmöglichkeiten, die der Tester ohne Zusatz-Equipment erledigt, sind noch große Kontakt-Pads auf dem PCB angelegt, um Kabel von Hand mit einem Multimeter durchzumessen. Dafür sind auch extra Schalter für Stromisolation auf dem PCB aufgebracht, um die Testströme des USB-Testers bei Multimetertests abzustellen.

Der Tester kann ohne einen angeschlossenen Computer oder ein Netzteil verwendet werden und wird nur von einer 3-Volt-CR2032-Knopfzelle gespeist. Die Platine ist ungefähr handtellergroß.

In seinem Blogpost geht Spencer Maroukis genauer auf den Entwicklungsprozess und die Testverfahren sowie deren technische Umsetzung ein. Dazu zählen auch detaillierte Schaltpläne und Erklärungen von USB-Standards und dem Aufbau von USB-Kabeln. Er erläutert, welche Widerstände für welchen Standard vorgesehen sind und wie man diese auch manuell mit einem Multimeter testen kann.

In diesem Blogpost findet sich auch ein Link zum GitHub-Repository des Projektes, auf dem sowohl der Code, Gerber- und KiCad-Dateien heruntergeladen werden können. —*das*

https://maroukis.net/usb-sleuth/

<page-header><page-header>

So sieht das Ergebnis eines USB-Tests aus.



Vom Prototypen ...



MuPiBox Musikplayer

Über die MuPiBox können lokale Musikdateien, Spotify und Streams aus dem Internet abgespielt werden. Die Bedienung per Touchscreen ist für Jung und Alt kinderleicht.

von Olaf Splitt



ittlerweile gibt es eine Vielzahl von Hörspielboxen für Kinder auf dem Markt. Teilweise sind Abonnements nötig oder die einzelnen Folgen sind vergleichsweise teuer. Wer seine CDs oder Kassetten digitalisiert hat oder einen Spotify-Account bezahlt, wird hier zweimal zur Kasse gebeten, um auf beiden Geräten die gleichen Inhalte zu hören.

Durch Zufall habe ich Erik kennengelernt und gemeinsam haben wir auf Basis des Sonos Kids Controllers eine Musikbox mit Touchscreen-Steuerung entwickelt. Herzstück der Box ist ein Raspberry Pi.

Die Box kann Inhalte wie MP3s, Spotify-Medien, Streams und Podcasts abspielen. Die Bedienung ist einfach und intuitiv, sodass sich Kinder schnell zurechtfinden. Nach Antippen von Interpreten oder Alben werden diese abgespielt. Die einfache Bedienung der Box hat nicht nur meinen großen Sohn überzeugt, auch mein Fünfjähriger forderte eine eigene Box.

Wir erkannten das Potenzial der Box und tauften das Projekt MuPiBox. Aus einem Bastelprojekt für die eigenen Kinder wurde ein Projekt, das für jeden umsetzbar sein sollte. Dabei war es uns wichtig, dass man die MuPi-Box auch ohne IT-Erfahrung oder Kenntnisse zum Raspberry Pi bauen kann.

Auch unser Team wurde größer und das Ergebnis war ein modulares 3D-gedrucktes Gehäuse. Damit können auch weniger handwerklich Begabte ein Gehäuse nach ihren Bedürfnissen drucken (lassen). Audiophile Anwender tunen die Box, um den Klang zu perfektionieren. Mittlerweile sind verschiedene Gehäuse im Einsatz, einige können in unserem virtuellen Showroom auf der Projektseite besichtigt werden.

Seit der ersten Version hat sich einiges getan. Der ursprünglich langwierige Installationsprozess wurde durch ein fertiges Image ersetzt. Das Image wird einfach auf eine MicroSD oder einen USB-Stick geschrieben. Vor dem Start werden noch die WLAN-Daten hinzugefügt und schon ist die Box installiert. Im Webinterface können die Eltern die Box dann weiter modifizieren. Neuerdings können auch LED-Streifen angeschlossen werden. Die LEDs reagieren dann auf die Musik und die Box wird zur mobilen Disco.

Inzwischen wurde der MuPiHAT für den Raspberry Pi entwickelt. Dieser bietet die Möglichkeit, den Raspberry Pi kontrolliert ein- und auszuschalten. Es verfügt über eine integrierte Soundkarte, die einen Mono- oder Stereobetrieb ermöglicht. Weiterhin kann die Box mit Batterien betrieben werden und verfügt so immer über eine stabile Stromversorgung.

Die Box ist so individuell wie der Erbauer. Mittlerweile überzeugt die Box auch in Seniorenheimen, Kindertagesstätten oder im heimischen Wohnzimmer. —*caw*



Der gedruckte Halter für Display, Raspi und Elektronik



Das zweigeteilte und modulare Gehäuse lässt viel Spielraum für eigene Ideen.



Die MuPiBox in Standardausführung mit gedrucktem Gehäuse

Nistkasten mit Anwesenheitsmeldung

Wir wollen im Wald verteilt für die heimischen Vogelarten Nistkästen aufhängen, um das Brutverhalten der Vögel unauffällig zu studieren. Wann wird das Nest gebaut, wie entwickeln sich die Jungvögel und wird der Kasten ein zweites Mal belegt? Unser Nistkasten meldet das über LoRaWAN.

von Manfred Caspar und Dominik Kuhn



s gibt bei uns immer weniger Vögel. Dieses Projekt soll helfen und Nistmöglichkeiten für unsere heimischen Vögel im Wald schaffen. Und nicht nur das: Wir Maker von der Linux User Group Saar haben die auf Basis von NaBu-Bauanleitungen gebastelten Nistkästen besonders ausgestattet, denn wir möchten zusätzlich wissen, ob der Kasten genutzt wird, und zwar ohne die Vögel beim Brüten zu stören.

Die Überwachung der Nistkästen abseits im Wald soll von zu Hause aus und ohne Batteriewechsel vor Ort möglich sein. Das erfordert Sparsamkeit beim Stromhunger der Elektronik und der Datenübertragung. Eine Kamera hätte viele Ressourcen (Strom) gefressen, ebenso wie die Datenübertragung übers Mobilfunknetz. So haben wir uns entschieden, die Nutzung des Nestes mithilfe von Gewichtssensoren zu erfassen. Eine Seite des Sensors ist mit dem Nistkasten, die andere Seite mit einer Plattform verschraubt, auf der das Nest sitzt. Das geringe Gewicht eines Rotkehlchen (12 g) reicht aus, die Wägezelle so weit zu verformen, dass ein deutliches Signal entsteht.

Dazu müssen nur jeweils ein paar Bytes in größeren Abständen übertragen werden. Zur Übertragung setzen wir auf das energiesparsame **Lo**w Power Long **Ra**nge **W**ide **A**rea **N**etwork (kurz: LoRaWAN), ein weltweites offenes und freies System aus Software- und Hardware-Modulen sowie öffentlich zugänglichen Gateways, welche dem Open-Source-Gedanken folgen. Der Energieverbrauch ist sehr niedrig, die Datenübertragung liegt bei maximal 50 kbit/s bei einer Reichweite in ländlichen Gebieten von bis zu 10, manchmal sogar 40 km.

Wir integrierten unser Projekt in diese Infrastruktur. Die von unserer LoRa-Node übertragenen Daten werden von den öffentlichen Gateways in Reichweite empfangen und von dort an das "The Things Network" (TTN) weitergeleitet. Dort können wir die Daten einsehen, mit MQTT abrufen und weiter verarbeiten.

Die Hardware ist ein Heltec LoRaWan-Board, das neben dem LoRa-Sender einen ESP32, ein kleines Display sowie die Energieverwaltung für einen LiPo-Akku enthält. Der Controller wird in einer wasserdichten Verteilerdose aus dem Baumarkt seitlich am Nistkasten befestigt. In der Dose sitzt auch der Akku. Die Software überträgt alle vier Minuten Daten. Das spart Strom und überlässt auch anderen Anwendern genug Gateway-Kapazität.

Falls jemand den Nistkasten nachbauen möchte: Software und weitere Infos gibt es auf unserer Github-Seite.

Eine ausführliche Bauanleitung für Makeund heise+-Abonnenten erscheint demnächst online, sie wird dann über den Link erreichbar sein. —hgb

make-magazin.de/x9du



Im Nistkasten sitzt über dem Boden eine Zwischenschicht, auf der später das Nest ruht. Darunter befindet sich die Wägezelle. Der Controller ist seitlich in einer Verteilerdose angebracht. Nestbau, Brüten und Füttervorgänge können durch die Gewichtsänderungen erkannt werden.



So funktioniert LoRaWAN: Von der Messstation (dem LoRa Node) werden die Daten per Funk (lizenzfreie Frequenzen regional verschieden zwischen 433 und 928 MHz, Reichweite bis 10 km) mit sehr geringer Leistung an die öffentlich zugänglichen LoRa-Gateways gesendet. Das Gateway hat eine Verbindung zum Internet und leitet darüber die Daten zu einem zum Beispiel von The Things Network betriebenen Server weiter. Von dort aus können User ihre Daten dann z. B. über MQTT abrufen und benutzen.

	hlaumainen						
	ID: blaumeisen						
	↑1 ↓1 • Last activity 8 minu	tes ago 🗇					
	Overview Live data Messa	aging Lo	cation Pay	load form	sutters General settings		
Time	Туре	Data previ	rw.		Verbor	e denam 🕥	🛓 Esport as JSC
4-20101-07	Schedule data downlink for transmiss.	Devidence :	26 08 71 73	0.	Sul Delay: 5		
P 20-01-06	Forward uplink data message	Deutotri	26 88 71 71	0.	Paylond: { mestgewicht_in_grame: 44.4 } 01.00 0 % (Pourt: 2 Data vate: 5P8	98125 3581 -2.25	HILL -334
t in an in	Successfully processed data message	Developer	28 00 71 71	0 🎼			
4-20-00-00	Schedule data dowilink for transmiss.	Deviking;	28 08 73 75	0.	Bul Delay: B		
P. 20100-48	Forward uplink data message	Revisite)	26 08 71 71	0.6	Period: [nestgestint_in_grame: 44.3] 01.00 O 🐞 (Port: 2 Data rate: 598	B4125 S101 +7.25	1051 -115
P 20-201-48	Successfully processed data message	Deskittet	28 88 71 71	0.6			
↑ 20:00:23	Forward join-accept message	Deukonz:	26 88 71 71	0.8			

Diese Daten liefert der LoRaWAN-Server. Aus der Gewichtsveränderung kann man ableiten, ob der Nestbau begonnen hat (langsames Ansteigen), ob Eier gelegt wurden (stufenweises Ansteigen über mehrere Tage) oder wie die Gewichtszunahme der Küken ist.

Die Birkenfelder Box

Was macht ein Schaltschrank im Walde, ganz still und stumm, nur mit Solarpaneelen drumherum? Es ist eine "Birkenfelder Box", ein gemütliches und Energie-autarkes Zuhause für LoRa-Gateways & Co. in der Wildnis.

von Rainer Maria Kreten

Antenne

Standrohr (Steck- oder Teleskopmast)



Ableitung Antenne

Batterie

Hinweisschild

FORSCHUNG BRAUCHT INTERNET

Dieses Gerät sorgt dafür, dass Messwerte von Sensoren ins Internet und welter auf unsere Server gelangen. Sa erforschen wir hier in Zusammenarbeit mit Hochschulen und Forschungseinrichtungen nicht nur das Verhalten von Rehen, die mit speziellen Halsbändern ausgestattet sind. Auch Messwerte zu Temperatur und Feuchtigkeit werden über dieses Gerät übertragen.

Darüber hinaus kann jedermann diese Station kostenfrei mitnutzen. Fragen Sie uns einfach!

Weitere Infos: www.nlphh.de/forschung-und-monitoring/

Inlagenbetreiber:

Nationalparkamt Hunsrück-Hochwald Brückener Strasse 24 | 55765 Birkenfel

-Mail poststelle@nlphh.de elefon 06131 884 152-0 Nationalpark Hunsrück-Hochwald

Schaltkasten mit Stromversorgung

Erdung

Solarpanels

6

er Umwelt-Campus Birkenfeld der Hochschule Trier liegt am Rande des Nationalparks Hunsrück-Hochwald. Dieser dient vielen Forschern aus dem ganzen Bundesgebiet als riesiges Freiluft-Labor. Wo geforscht wird, da fallen Daten an, die auch nahezu in Echtzeit auf unseren Servern landen sollen. Wieso geht es den jungen Fichten im Vordergrund des Fotos scheinbar so gut, während ihre alten Eltern im Hintergrund bereits von Borkenkäfern und Klimawandel dahingerafft wurden? Was machen Temperaturverläufe und Wasserversorgung und welche Auswirkungen hat das auf das Rehwild, aber auch auf Moose, Flechten und andere Kleinstlebewesen? Für alle möglichen Parameter gibt es Sensoren, die per LoRaWAN ihre Werte in die Welt funken. Also gingen wir daran, zusammen mit der örtlichen Kommunalverwaltung am Aufbau des LoRa-Netzes zu arbeiten. "Wir" sind in diesem Zusammenhang die Nationalparkverwaltung gemeinsam mit der Hochschule.

LoRaWAN gilt als flexibel, unkompliziert und preiswert. Das ist es aber nur solange, wie wir in unseren eigenen Häusern und Wohnungen agieren. Kommen Fachunternehmen ins Spiel, die sich um die Montage der Gateways an öffentlichen Gebäuden kümmern, die Leitungen verlegen, Antennen montieren und sich um Dinge wie Blitzschutz und Zugangstüren kümmern, steigen die Kosten leicht um das Zehnfache der verbauten LoRa-Hardware. Darüber hinaus gibt es nicht überall geeignete Standorte, um auch örtlich wechselnde Forschungsprojekte mit einer stabilen Netzabdeckung zu versorgen.

Flexibel und nachbausicher

So kam der Wunsch auf, flexibel einsetzbare Gateway-Stationen zu haben, die im Wesentlichen aus einer wetterfesten Kiste mit Antenne und ausreichend Photovoltaik für Energie-



Ab vier Meter Höhe muss der Mast verspannt werden. Der Abspannkranz wird in Bodennähe montiert und dann beim Ausziehen des Teleskopmastes in die Höhe geschoben.

Kurzinfo

- » Router, LoRaWAN-Gateways und andere Technik in der Wildnis nutzen
- » Insel-Photovoltaik-Lösung abseits von Fahrzeugen und Gebäuden
- » Erfahrungen bei Montagearbeiten in freier Natur

Mehr zum Thema

» Einen ausführlichen Artikel zur Birkenfelder Box gibt es im Volltext gratis online über den Link

Autonomie bestehen. Platz sollte darin nicht nur für das LoRaWAN-Gateway sein, unsere Birkenfelder Boxen sollten auch weitere Technik im Nationalpark beherbergen können.

Und: Es sollte sich um eine Lösung handeln, die mit den Mitteln einer Hobbywerkstatt sicher nachgebaut werden kann und dabei funktional und kostengünstig ist. Abgesehen von den LoRa-Komponenten wurde daher ein örtlicher Baumarkt als Hauptlieferquelle genutzt. In der Konstruktion kommt daher etwa eine Boden-Einschraubhülse zum Einsatz, wie man sie auch für Wäschespinnen verwendet. Ein Schaltschrank bildet die eigentliche Box, die wiederum mit Auspuffschellen am Pfosten befestigt wird.

Alle Details online

Wie die Birkenfelder Box im Detail aufgebaut ist, haben wir ausführlich online beschrieben (zu finden über den Link in der Kurzinfo). Dabei berichten wir nicht nur über die technischen Herausforderungen und die gefundenen Lösungen, sondern verraten auch, bei welchem Arbeitsschritt Praktikantenblut floss, warum beim Aufstellen der Box im Gelände ein Badelaken unverzichtbar ist und auf welche Spezies von Waldbewohnern die Installation rätselhafterweise besondere Anziehungskraft auszuüben schien.

Der Materialeinkauf für eine Box schlug am Ende mit rund 1300 Euro zu Buche. Es ist also kein billiges Projekt (die Materialliste gibt es ebenfalls im Online-Artikel). Zudem waren für uns als Behörde einschlägige Internetplattformen als Lieferquellen ausgeschlossen, ein Tribut an das Landeshaushaltsrecht. Die Kosten relativieren sich aber sehr, wenn man Alternativen zum Aufspannen des Funknetzes durchkalkuliert. Alternativen zu betrachten lohnt sich auch mit Blick auf die Materialliste. Vieles ist ersetzbar oder es findet sich ein brauchbares Teil im Keller oder in der Garage. Gerade in Hinblick auf die aktuell oft gestörten Lieferketten sind wir gut beraten, wenn wir



Alles zum Artikel

im Web unter

wissen, wie wir unsere Projektziele auch unter nicht immer optimalen Rahmenbedingungen realisieren können. —pek



Das aktuelle Innenleben der Birkenfelder Box. Das LoRaWAN-Gateway selbst ist ein Outdoor-Modell und sitzt inzwischen mit dem LTE-Modul außerhalb der Box am Mast.



Die Bleibatterie passt exakt in eine wasserdichte Munitionskiste. Zur Verbindung mit der Box dient eine schwere Gummischlauchleitung.

Material für Lasercutter

Kistchen, Kästen, Formen, Stempel, Bilder, Schilder.... Was kann man nicht alles mit einem Lasercutter machen. Doch woraus? Welches Material ist geeignet, welches nicht? Das erfahren Sie in diesem Artikel. Und auch, wie Sie scheinbar widerspenstigen Stahl doch zur Zusammenarbeit mit einem CO₂-Laser bringen können.

von Heinz Behling



S perrholz, Plexiglas oder doch massives Buchenholz? Pappe oder Papier? Was kann man mit einem Lasercutter schneiden und/oder gravieren? Und funktioniert das dann sowohl mit einem Röhren-CO₂-Laser als auch mit einem Dioden-Gerät? Diese Übersicht sagt Ihnen, was möglich ist und was Sie besser nicht versuchen sollten.

Bei den Lasercuttern, um die es hier geht, handelt es sich um CO₂-Laser bis zu 100 Watt Leistung und Dioden-Laser mit maximal 40 Watt, da diese Geräte im Maker-Bereich und FabLabs hauptsächlich verbreitet sind.

Die Leistung des Geräts bestimmt vor allem die maximale Materialstärke sowie die Geschwindigkeit, die damit geschnitten werden kann. Die Materialart ist hingegen vor allem von der Wellenlänge des Laserstrahls abhängig (siehe Kasten "Begriffserklärung: Transmission, Reflexion, Absorption" auf Seite 62).

Kurzinfo

» Schneide- und Gravur-Materialien für Dioden- und CO₂-Laser » Material, das nicht in den Lasercutter sollte

» Tipps zum Lasern

Mehr zum Thema

- » Andreas Bochmann: Lasercutter belichtet Platinen, Make 5/18, S. 98
- » Heinz Behling: Schneiden und Gravieren, Make 4/18, S. 126
- » Heinz Behling: Laser-Fliesenschneider, Make 6/18, S. 114
- » Heinz Behling: Lasern mit Lightburn, Make 1/24, S. 24
- » Heinz Behling: Kamera für Lightburn, Make 1/24, S. 34



Laserfähiges

Im Folgenden zeige ich Ihnen eine Übersicht von laserfähigen Materialien. Meist werden Sie das Material als Platten bei einem Händler vor Ort oder online kaufen. Viele Stoffe gibt es auch in anderen "Bauformen". Plexiglas beispielsweise wird auch als Rohre oder in Kugelform angeboten.

Achten Sie bei der Lagerung darauf, dass Platten sich durch ihr Eigengewicht verziehen können. Liegen Sie dann im Lasercutter nicht mehr eben, ist eine korrekte Verarbeitung kaum möglich. Lagern Sie Platten daher liegend übereinander, am besten noch durch ein Gewicht beschwert.

Die Übersicht beruht auf meinen Erfahrungen mit CO₂- und Diodenlasern. Falls Sie hier konkrete Einstellungsempfehlungen für Leistungen/Geschwindigkeiten beim Schneiden oder Gravieren erwarten, muss ich Sie enttäuschen. Meiner Erfahrung nach streuen die Werte zwischen den einzelnen Lasercuttern sehr stark, selbst, wenn sie dieselbe Leistung versprechen. Konkret vorgegebene Werte würden Sie daher mit hoher Wahrscheinlichkeit noch mehr enttäuschen. Wenn Sie das nicht glauben: Schauen Sie einfach ins Internet, welche Empfehlung da gegeben werden und wie weit die fürs gleiche Material beim gleichen Laser auseinander liegen können.

Empfehlenswerter ist es, sich für den jeweils eigenen Laser eine Sammlung von Materialproben anzulegen, auf denen Sie beispielsweise mit dem Material-Test von Lightburn die Leistungs-/Geschwindigkeitskombinationen ausprobiert haben.

Sperrholz

Dies ist wohl das meistverwendete Lasermaterial, das sich mit unseren Geräten in Stärken bis 10 mm verarbeiten lässt. Sperrholz besteht aus mindestens drei aufeinander geklebten Schichten, bei denen der Faserverlauf um jeweils 90 Grad verdreht ist. Das erhöht die Stabilität des Holzes enorm. Dennoch reagiert Sperrholz insbesondere auf Feuchtigkeit. Es verzieht sich, kann sich in Extremfällen sogar um mehrere Zentimeter wölben. Solche Platten sind dann im Lasercutter kaum noch brauchbar. Bewahren Sie Sperrholz daher am besten liegend in einer vor der Luftfeuchtigkeit schützenden Verpackung auf.

Allerdings ist Sperrholz nicht gleich Sperrholz. Da gibt es unterschiedliche Holzarten, die sich im Preis, vor allem aber in ihrer Widerstandsfähigkeit unterscheiden. Außerdem wird zwischen mehrere Qualitätsstufen unterschieden. Die Stufe AA zum Beispiel bedeutet, dass auf beiden Oberflächen des Sperrholzes maximal 3 mm große Äste vorhanden sein



Bild 1: Willy zwölfmal in 3-mm-Birkensperrholz graviert



Bild 2: In der Seitenaufnahme des angeschnittenen Bär-Reliefs ist die Tiefe der Gravur zu erkennen.



Bild 3: Dieselbe Datei einmal auf Plexiglas XT (links), das andere Mal auf Plexiglas GS: Die Gravur ist auf XT deutlich besser zu erkennen. Beschriftungen hingegen gelingen auf GS besser.

dürfen. Außerdem darf summiert auf einen m² maximal 6 mm² an Ästen vorhanden sein. Bei AB darf eine Seite des Sperrholzes schon Äste bis 15 mm Durchmesser sowie Kittstellen als Lochfüllung enthalten.

Am einfachsten ist Birke und Pappel zu lasern, auch aus Balsa wird Sperrholz angeboten, das sich lasern lässt. Harthölzer wie Buche und Eiche hingegen fordern dem Laser richtig Leistung ab. Da können bei schwächeren Lasern mehrere Schneidedurchgänge erforderlich werden. Das Gravieren ist deutlich anspruchsloser. Hier können auch mit 5-W-Diodenlasern brauchbare Ergebnisse erzielt werden. Sperrholz lässt sich mittels Gravur nicht nur einfärben, sondern sogar reliefartig bearbeiten. Im Bild 1 wurde mein Teddy-Fotomodell (er heißt übrigens Willy) zwölfmal ins Sperrholz graviert.

So wurde an den dunklen Stellen über die Hälfte des Holzes abgetragen (Bild 2). Dadurch ergab sich nicht nur eine deutliche Färbung, sondern der Bär erschien auch als Relief mit Tiefenwirkung.

Tipp: In Baumärkten wird 3- oder 4-mm-Birkensperrholz oft als 5er-Pack in Folie eingeschweißt im Format A4 und A3 angeboten. Ich habe da immer wieder die Erfahrung gemacht, dass die beiden sichtbaren Oberflächen des Pakets in Ordnung waren, im Inneren sich jedoch Minderwertiges versteckte. Lassen Sie sich dann besser Material aus einer großen Platte zuschneiden, auch, wenn das ein paar Cent teurer sein sollte.

Massivholz

Wie bei Sperrholz gilt auch hier: Weiche Hölzer wie Birke oder Pappel sind einfacher zu bearbeiten als harte Hölzer. Da es viele Massivhölzer nur in Stärken ab 15 mm gibt, fällt das Schneiden von Eiche oder Buche sowie vieler anderer Holzarten ohnehin weg.

Holzfurniere oder Balsaholz hingegen lassen sich sehr gut schneiden. Birken- und Pappelbretter bis zu 5 mm Stärke sind für den CO₂-Laser gut geeignet. Der Diodenlaser sollte für diesen Zweck schon ein 20- oder 40-W-Modell sein.

Begriffsklärung: Reflexion, Transmission, Absorption

Damit ein Material mit einem Laserstrahl bearbeitet werden kann, muss es die Energie des Laserstrahls aufnehmen. Dabei spielen drei Begriffe eine wichtige Rolle:

1. Reflexion: Dies ist die Eigenschaft, eine Strahlung an der Oberfläche zurückzuwerfen, etwa sichtbares Licht an einer Metalloberfläche. Auch die Reflexion ist wellenlängenabhängig. Ist die Reflexion bei der Wellenlänge eines Laserstrahls groß (bei glänzenden Metallen nahe 100 Prozent), gelangt beim Lasern kaum Energie ins Material. Daher findet bestenfalls eine geringfügige Erwärmung um einige wenige Grad statt. Für Gravuren oder Schnitte ist das bei der Leistungsklasse unserer Laser aber viel zu wenig. Da braucht man schon Laser mit Strahlleistungen im Kilowattbereich.

2. Transmission: Darunter versteht man die Eigenschaft eines Materials, eine Strah-

lung hindurchzulassen. Meist wird immer ein ganzer Wellenlängenbereich durchgelassen, bei herkömmlichem Glas zum Beispiel der erweiterte Bereich um das sichtbare Licht herum zwischen 380 und 780 nm Wellenlänge. Diesen Durchlassbereich nennt man das Transmissionsfenster eines Materials.

Damit ein Material mit dem Laser bearbeitet werden kann, muss es die Laserenergie aufnehmen. Die Wellenlänge des benutzten Lasers muss also außerhalb des Transmissionsfensters liegen. Der Stoff ist dann für den Laserstrahl undurchsichtig. So werden die Wellen eines CO₂-Lasers (10.600 nm) in Glas schon in den ersten Millimeterbruchteilen zu über 99 Prozent aufgenommen und in Wärme umgesetzt, was zum Absprengen kleiner Partikel an der Oberfläche führt. Das blaue Licht eines Diodenlasers hingegen geht zu über 95 Prozent durch solches Glas hindurch und hinterlässt keine Wirkung, da die Strahlenergie nicht ausreichend vom Material aufgenommen wird.

3. Absorption: Die Absorption kennzeichnet den Anteil der Strahlungsenergie, die ins Material eindringt, es aber nicht wieder verlassen kann. Beim Lasercutten liegt die Absorption meist bei über 90 Prozent, geeignete Laser-/Materialkombination vorausgesetzt. Diese Energie wird zum weitaus größten Teil in Wärme umgesetzt, wobei Temperaturen von mehreren Tausend Grad entstehen können. Das Material wird dabei nicht verbrannt, sondern verdampft. Dieser Dampf muss möglichst schnell aus dem Schnitt entfernt werden, damit er sich bei Kontakt mit dem Luftsauerstoff nicht entzündet und die Schnittkanten verbrennen und/oder verrußen. Dazu dient der Luftstrom des Airassist-Systems.

Das Gravieren hingegen ist unabhängig von der Materialstärke bei fast allen Holzarten möglich, allerdings ist der Effekt bei ohnehin dunklen Holzarten kaum noch erkennbar.

Plexiglas

Dieser Kunststoff aus Polymethylmethacrylat (PMMA) ist ebenfalls sehr beliebt beim Lasern. Auf CO₂-Geräten (40W) sind alle Pexiglas-Arten bis 10 mm Stärke zum Schneiden und Gravieren verwendbar. Diodenlaser mit ihrem meist blauen Licht kommen nur mit schwarzem, gelbem oder rotem Plexiglas zurecht.

Plexiglas gibt es im Handel als gegossenes (Plexiglas GS) und extrudiertes Material (Plexiglas XT). Äußerlich ist beides außer an den Aufdrucken der Schutzfolie nicht zu erkennen. Beim Gravieren zeigen sich aber deutliche Unterschiede. Plexiglas XT reagiert deutlich schwächer auf den Laserstrahl. Dadurch bleibt die Gravur transparenter und detaillierter als beim gegossenen Plexiglas (Bild 3). Schnittkanten sind bei XT-Platten außerdem deutlich glatter, bei GS hingegen sind meist Riefen erkennbar, die die Passgenauigkeit bei zusammengesteckten Teilen beeinträchtigen können.

Finnpappe

Finnpappe sieht bereits aus geringen Abstand fast aus wie helles Sperrholz (Bild 4). Dennoch handelt es sich um Pappe, die aus reinem Holzschliff gefertigt wird und sehr stabil ist. Die Färbung ist aber nicht lichtecht und verändert sich insbesondere bei Sonneneinstrahlung. 3 mm dicke Pappe kann man kaum knicken. Damit lassen sich durchaus ähnlich stabile Produkte herstellen wie mit entsprechendem Sperrholz bei deutlich geringerem Gewicht. Außerdem reagiert Finnpappe im Gegensatz zu Sperrholz kaum auf Temperatur- und Feuchtigkeitsänderungen. Sie bleibt schlicht eben ohne jede Wölbung.

Zum Schneiden ist aber viel weniger Energie erforderlich. Selbst 5-W-Laser kommen damit zurecht. Ein 40-Watt-CO₂-Laser kann Finnpappe mit mehr als 30 mm/s schneiden, sofern er richtig fokussiert ist.

Die Pappe ist im Handel mit Stärken von 0,5 mm bis zu 10 mm erhältlich.

Graupappe

Diese Pappe ist eines der preiswertesten Materialien. Sie besteht meist aus Altpapier-Fasern, die mithilfe von unterschiedlichen Bindemitteln (Leim, Kunststoffharze u. ä.) unter Druck und Hitze zu Platten geformt werden. Die Frage nach den Verarbeitungsmöglichkeiten von Graupappe in Lasercuttern ist nicht eindeutig zu beantworten. So habe ich selbst schon Graupappe mit 4 mm Stärke problemlos geschnitten, während Material mit 2 mm sich



Bild 4: Finnpappe (oben) und Sperrholz (unten) sehen sich sehr ähnlich.

erfolgreich wehrte. Ebenso kann es in Diodenlasern funktionieren, in CO₂-Laser hingegen nicht oder umgekehrt. Die Ursache für dieses widersprüchliche Verhalten liegt vermutlich in den verwendeten Bindemitteln.

Graupappe sollten Sie daher nur aus Quellen beziehen, die Ihnen bestätigen, dass die Pappe für Ihren Laser geeignet ist. Aber auch dann müssen Sie damit rechnen, dass Graupappe anders als Finnpappe beim Schneiden stark rauchen kann und dementsprechend die Staubfilter in der Abgas-Anlage des Lasercutters belastet.

Übrigens: Das Innenleben von vielen Ikea-Möbeln besteht ebenfalls aus Graupappe. Im Gegensatz zu den dünnen äußeren Dekorschichten der Möbel ist die Graupappe aber das eigentliche, tragende Teil dieser Möbel und dementsprechend stabil. Ich konnte beispielsweise beim Anfertigen von Luftdurchlässen in einer aus Ikea-Schrankteilen gefertigten 3D-Druck-Vitrine nur die Außenteile lasern, für die Graupappe im Inneren brauchte ich eine Stichsäge.

Alu eloxiert

Aluminium lässt sich nur dann lasern (genauer gesagt: nur gravieren), wenn die Oberfläche beschichtet ist. Meist ist das Aluminium eloxiert, das heißt, die Oberfläche wurde elektrochemisch oxidiert. Die so gebildete, wenige Mikrometer dicke Oxidschicht kann dann noch eingefärbt werden. Deshalb gibt es im Handel eloxiertes Aluminium in zahlreichen Farben. Beim Gravieren wird die Farbstoffschicht abgetragen, sodass das metallische Aluminium zum Vorschein kommt.

Das Material ist für CO₂- und Diodenlaser geeignet. Bei Diodenlasern können aber bei hellen Farben, Gold- oder Silbertönen eine höhere Leistung oder mehrere Durchgänge notwendig sein.

Durch das Gravieren wird das Aluminium übrigens nicht korrosionsanfälliger, denn durch die Hitze oxidiert das Aluminium an den Gravurstellen augenblicklich wieder. Diese farblose Oxidschicht schützt das Metall vor weiterer Oxidation.



Bild 5: Solche Taschenlampen aus eloxiertem Alu sind problemlos mit Diodenund CO,-Lasern zu gravieren.



Bild 6: Beim Spiegel links wurde nur die Glas-Vorderseite mit einem CO₂-Laser graviert, beim rechten fand die Bearbeitung auf der Rückseite statt. An den dunklen Stellen ist hier sogar die Spiegelschicht mehr oder weniger abgetragen.

Eloxiertes Aluminium gibt es als Blechplatten. Häufiger jedoch werden daraus gefertigte Gegenstände graviert, etwa Taschenlampen (Bild 5), Behälter und vieles mehr.

Leder

Dünneres (bis 3 mm), weiches Leder (vor allem Wildleder) lässt sich mit beiden Laserarten gut lasern. Allerdings kann es sich durch die dabei entstehende Wärme verziehen und noch während der Bearbeitung Wellen oder Falten schlagen. Leder sollte daher auf dem Schneidetisch mit kleinen Gewichten oder Magneten außerhalb des Schnitt-/Gravierbereichs fixiert werden. Stabilere, weil stärker gegerbte Lederarten (wie steifes Rindsleder) müssten selbst bei voller Leistung in mehreren Durchgängen geschnitten werden. Allerdings sollte man volle Leistung vermeiden, da es dadurch zu Verkohlungen an den Schnittkanten oder zumindest zu Verfärbungen kommen kann. Das erfordert dann natürlich noch mehr Schneidedurchgänge.

Glas/Spiegel

Handelsübliches Klarglas (auch Fensterglas genannt) kann mit einem CO₂-Laser hervorragend graviert werden, da es die Laserenergie in dem Wellenbereich zu fast 100 Prozent auf-



Bild 7: Mit dem Laser angeritzte Kacheln lassen sich gerade oder sogar in Kurven brechen, solang die Radien nicht zu klein sind.

nimmt. Da dies aber bereits in den ersten Millimeterbruchteilen an der Oberfläche geschieht, gelangt keine Energie in größere Tiefen. Das bedeutet zum einen, dass Glas sich nicht lasercutten lässt, zum anderen, dass es beim Lasern zu starken Temperaturunterschieden an der Oberfläche kommt. Glas reagiert darauf mit Sprüngen. In der Tat läuft der Gravurprozess beim Glas nicht durch Verdampfen, sondern dadurch, dass winzige Glassplitter von der Oberfläche abplatzen.

Glasbearbeitung mit Diodenlasern ist kaum möglich. Selbst farbiges Glas nimmt zu wenig Energie auf. Im Netz hört man manchmal vom Edding-Trick: Dabei wird die Glasoberfläche mittels schwarzem Eddingstift eingefärbt und diese Oberfläche dann mit dem Diodenlaser bearbeitet. Ich habe es probiert, es gelang mir nicht. Das mag aber auch am Eddingstift liegen. Vielleicht hat einer von den Lesern da eine Empfehlung, mit der es funktioniert.

Auch Spiegel bestehen in der Regel aus Klarglas, daher gilt für die eine Oberfläche das bereits gesagte. Die Rückseite jedoch ist anders aufgebaut, nämlich aus der eigentlichen Spiegelschicht aus Silber- oder Aluminium direkt auf dem Glas. Dahinter ist dann noch eine lichtundurchlässige Lackschicht aufgebracht. Diese Lackschicht kann mit beiden Laserarten graviert werden, sogar bis zur völligen Entfernung (Bild 6). An solchen Stellen wird der Spiegel dann teildurchlässig wie ein Spionspiegel.

CO₂-Laser können mit Leistungen von 40 Watt und geringer Geschwindigkeit sogar die Spiegelschicht gravieren. An diesen Stellen findet dann keine Spiegelung mehr statt, sondern das Glas wird einfach durchscheinend. Kombiniert man beides, sind beeindruckende Effekte erzielbar.

Keramik

Keramik (Porzellan, Steingut, Kacheln etc.) ist ein äußerst widerstandsfähiges Material. Daher lässt es sich mit keinem der beiden Laserarten gravieren oder schneiden.

Allerdings ist Keramik meist glasiert. Und für diese Glasurschicht gilt dasselbe wie für Glas: Die Gravur mittels CO₂-Laser ist möglich, der Diodenlaser muss meist passen, es sei denn, es handelt sich um dunkel eingefärbte Glasuren. Da kann es funktionieren, ist aber stark einzelfallabhängig, denn die verwendeten Materialien sind sehr unterschiedlich.

Zum Thema Schneiden: Wenn zum Beispiel ein Fliesenleger eine Kachel zuschneidet, dann ritzt er lediglich die Glasuroberfläche an. An dieser geschwächten Stelle bricht dann die Kachel, sobald man sie entsprechend biegt. Das Anritzen kann auch mithilfe eines CO₂-Lasers erledigt werden. Ich hab das in einem früheren Artikel auch schon in der Make

Know-how



Bild 8: Solch ein Materialtest mit unterschiedlichen Laserleistungs- und Schnittgeschwindigkeits-Kombinationen ist bei Faserplatten ratsam. Die Schmauchspuren lassen sich mit einem feuchten Tuch und etwas Spülmittel beseitigen.

beschrieben (Bild 7, siehe auch "Mehr zum Thema" in der Kurzinfo). Das gilt übrigens auch für Glas.

Stein

Steine schneiden ist mit Lasern dieser Leistungsklasse nicht möglich, gravieren aber meist sehr wohl. Besonders Schieferplatten sind geeignet, da sie in der Regel eine ebene Oberfläche aufweisen, auf die der Laser einfach zu fokussieren ist. Auch ist Schiefer ziemlich weich, sodass der Gravureffekt recht deutlich ist. Ebenso ist Marmor gut zu gravieren.

Bei Natursteinen wie Kiesel, Geröll oder ähnlichem sollte Sie es einfach ausprobieren. Die Zusammensetzung der Steine kann sehr unterschiedlich sein. Daher kann ich hier keine allgemein gültige Empfehlung für einzelne Mineralien geben. Weiches Material wie Bims oder lockeren Sandstein sollte Sie aber meiden. Bei der Bearbeitung dieser Steinarten löst sich viel scharfkantiger Staub, der sich in der Mechanik des Lasercutters festsetzen kann. Das bekommt Gleit- und Kugellagern nicht gut.

Beachten Sie auch, dass größere Steine/ Platten ein beträchtliches Gewicht haben können. Das kann eine elektrische Höhenverstellung des Arbeitstisches oder einen Autofokus eventuell überfordern. Dadurch entsteht kein unmittelbarer Schaden, aber es ist möglich, dass der Laser falsch fokussiert.

Papier

Um Papier zu schneiden oder zu gravieren reicht die Leistung der Lasercutter, ob CO₂ oder

Diode, meist nicht nur völlig aus, sondern ist bei dünneren Papieren oft zu hoch. Insbesondere CO₂-Laser sollten auf die geringstmögliche Leistung (ca. 10 Prozent) und hohe Geschwindigkeit eingestellt werden. Ein Airassist-System sollte besser abgeschaltet bleiben, da es das Papier vom Arbeitstisch wehen kann. Außerdem muss Papier immer am Tisch befestigt werden (Gewichte, Magnete), da es sich beim Schneiden meist stark wölbt.

Dioden-Laser müssen entsprechend der Papierfarbe in der Leistung unterschiedlich justiert werden: Weiß und blau erfordern hier etwas mehr Power.

Stempelgummi

Im Handel gibt es speziell fürs Lasern geeignete Stempelgummi-Platten. Nur die sollten Sie benutzen. Anderes Material ist entweder nicht zu verarbeiten oder kann schädliche Stoffe freisetzen. Rechnen Sie aber auch bei Laser-geeignetem Material trotzdem damit, dass Stempelgummi fast immer mehrere Bearbeitungsdurchgänge braucht, auch beim Gravieren.

Die Platten haben meist eine Stärke von 8 oder 10 mm. Um damit vernünftige Stempel herausarbeiten zu können, muss etwa die halbe Höhe weggraviert werden. Das kann schon bis zu fünf Wiederholungen erfordern. Das Material wird im Gegensatz zu den anderen Stoffen nicht verdampft, sondern als Staub weggesprengt. Die dabei entstehenden Staubmengen können beträchtlich sein. Doch nur ein geringer Anteil davon landet über die Abgasabsaugung im Filter.

Hier kommt Verstärkung



Das Make-Sonderheft bietet einen praxisorientierten Einstieg in Schaltungen mit Operationsverstärkern inkl. Experimentierset.

Will man Sensorsignale verarbeiten oder verstärken, Spannungen überwachen oder Audiosignale filtern: Mit geringem Aufwand und ohne komplizierte Berechnungen setzt man Operationsverstärker ein. Das Heft erklärt, wie alle Schaltungen funktionieren.

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

Heft + Experimentierset für nur 49,95 €



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





Bild 9: Dieser Aufkleber wurde aus Laserfolie gefertigt, deren obere Schicht gelb ist.

Das Material ist nämlich recht schwer, sodass sich der Staub rasch im Lasercutter absetzt. Nach dem Lasern von Stempelgummi ist es daher empfehlenswert, den Schneideraum auszusaugen.

Faserplatte (MDF/HDF)

Faserplatten gibt es als mitteldichte (MDF) und hochdichte (HDF) Platten. Sie werden bei-

spielsweise für Möbelrückwände, Verkleidungen u. ä. verwendet. Diese Platten gibt es auch mit Folien- oder Laminatbeschichtung (HPL, "Resopal") in unterschiedlichen Farben und Dekoren. Sogar Holzdesigns sind lieferbar. Diese Beschichtungen können auf einer oder beiden Seiten aufgebracht sein.

Unbeschichtete MDF-Platten bis 3 mm können von beiden Laserarten geschnitten und graviert werden. HDF-Platten erfordern schon einen CO₂-Laser oder einen starken Dioden-Laser (20 oder 40 W) bei voller Leistung und geringer Geschwindigkeit. Ähnlich wie bei Graupappe gilt aber auch hier, dass auch scheinbar gleiche Platten sich im Laser unterschiedlich verhalten können. Daher ist die Anfertigung eines Materialtests (unter Lightburn in "Laser-Tools/Material-Test") sehr zu empfehlen (Bild 8).

Plattenbeschichtungen können je nach Farbe in Dioden-Lasern Probleme bereiten, wenn sie dessen Licht zu stark reflektieren. Falls die Beschichtung nur einseitig ist: Lasern Sie von der unbeschichteten Seite aus.

Kraftplex

Die graubraunen Kraftplex-Platten (ab 0,8 mm) bestehen lediglich aus Holzfasern, die unter Hochdruck gepresst wurden. Beide Laserarten können dieses Material schneiden und gravieren. Besonderer Vorteil: Kraftplex raucht nahezu nicht. Da keine Harze als Bindemittel enthalten sind, entstehen auch keine

gesundheitsschädlichen Stoffe. Es ist meiner Meinung nach das einzige Material, bei dem man die Abgase des Lasercutters auch ohne Rauchfilter direkt ins Freie leiten kann, ohne Nachbarn zu sehr zu belasten (ich empfehle dies aber nicht).

Beim Schnitt sollte das Airassist-System eingeschaltet sein und mit relativ hoher Geschwindigkeit gearbeitet werden, damit die Ränder der Schnittkanten nicht verkohlen. Kraftplex sollte vor Nässe geschützt werden, da es sonst aufquellen kann. Auf Luftfeuchtigkeit reagiert es kaum. Leicht angefeuchtet lässt es sich sogar verformen, muss aber bis zum Trocknen in Form gehalten werden.

Laserfolie

Der Name Laserfolie führt eigentlich in die Irre. Es handelt sich hier um Platten aus zwei Kunststoffschichten unterschiedlicher Farbe. Die obere Schicht ist meist deutlich dünner als die untere. Wenn man die Oberseite per Gravur weglasert, erscheint die Farbe der unteren Schicht. Damit lassen sich sehr schöne Schilder (z. B. Namenschilder an Klingelknöpfen) oder Warnhinweise lasern (Bild 9). Da es diese Kunststoffplatten auch mit nur 0,1mm Gesamtstärke gibt, die dann sehr flexibel sind und oft auch noch eine selbstklebende Rückseite haben, nennt man sie auch Laserfolie.

Sehr oft wird der Begriff aber auch für Transparentfolie verwendet, die für Laserdrucker geeignet ist. Achten Sie beim Kauf darauf!

Das sollten Sie nicht lasern

Nicht alles, was auf den ersten Blick nach Laser-Material aussieht, sollten Sie auch verwenden. Es gibt einige Stoffe, die fürs Gerät oder Sie als Benutzer nicht sehr gut sind. Im Folgenden nenne ich Ihnen die vier Stoffe, die besonders gerne als laser-geeignet verdächtigt werden, es aber nicht sind. Diese Liste ist aber bei weitem nicht vollständig.

Makrolon

Polycarbonat (Makrolon) ist ein Kunststoff, aus dem zum Beispiel DVDs gefertigt werden, der sich aber nur sehr schwer lasern lässt, also hohe Strahlintensitäten und geringe Geschwindigkeiten erfordert. Dabei rußt das Material extrem stark. Diese Rußpartikel können an der Oberfläche zahlreiche gesundheitschädliche, sogar krebserregende Stoffe tragen, die beim Einatmen ungesund sind. Daher ist ein wirksamer Feinstfilter für die Laser-Abluft notwendig. Der aber verstopft durch die große Rußmenge sehr schnell. Schon ein einziger Lasercut kann die Filter verstopfen, was dazu führt, dass die Abgase nicht mehr aus dem Laser abgesaugt werden, der Ruß sich dort ablagert oder beim Öffnen des Geräts entweicht.

Makrolon wird jedoch oft zurecht als Sichtfenster in CO₂-Lasercuttern eingesetzt. Diese Anwendung ist durchaus sinnvoll, da es dem Laserstrahl lange widersteht und dabei deutlich zeigt, ob irgendwo infolge einer Fehljustierung der Laser aufs Fenster auftrifft (Bild 10)!

Polystyrol

Ähnliche Wirkung auf Filter hat auch Polystyrol, das in Baumärkten oft direkt neben Plexi-



© Copyright by Maker Media GmbH.

Know-how

glasplatten als deutlich billigeres, sogenanntes Bastlerglas verkauft wird. Es entwickelt beim Lasern Dämpfe, die sich als extrem klebriger Niederschlag im Gerät und insbesondere in den Filtern niederschlagen. So werden Staub- und Aktivkohlefilter schon nach wenigen Minuten unbrauchbar, da die Poren in den Filtermedien verstopfen. Daher Hände weg davon.

PVC

PVC (Polyvinylchlorid), auch oft schlicht Vinyl genannt, kann beim Lasern Chlorwasserstoff (HCl) entwickelt, das in Verbindung mit Wasser (Luftfeuchtigkeit, Haut-Oberfläche) zu Salzsäure wird. Im Inneren des Lasercutters kann dies zu übermäßiger Korrosion (Rost) führen. Schlimmer ist jedoch die Wirkung auf den menschlichen Körper, insbesondere beim Einatmen des gasförmigen HCI: In der Lunge kann es starke Reizungen auslösen. Daher sollten Sie PVC aus dem Lasercutter verbannen.

Teflon

Auch Teflon (Polytetrafluorethylen) verhält sich ähnlich wie PVC: Statt HCl entsteht jedoch Fluorwasserstoff (HF), das mit Wasser Flusssäure bildet, die noch aggressiver ist als Salzsäure. Daher sollten Sie auch diesen Kunststoff nicht lasern, also weder schneiden noch gravieren. —hgb



Der Trick mit der Baby-Creme

Stahl lässt sich mit den hier behandelten Lasern normalerweise nicht bearbeiten, da die Reflexion viel zu groß ist und keine Energie aufgenommen wird.

Mit einem Trick geht es aber doch, und zwar verblüffend gut. Dazu ist auch kein teures Zusatzmaterial wie die sogenannten Markierungssprays (zu Preisen um 100 Euro pro 300 ml) erforderlich. Stattdessen genügt Babycreme (Penaten-Creme).

Solche Creme (von der ich noch eine viele Jahre alte Dose in meinem Badezimmerschrank fand) enthält unter anderem Zinkoxid-Pulver. Und das ist der interessante Bestandteil.

Professor Heinz Kalt und Professor Claus Klingshirn, Forscher am Karlsruher Institut für Technologie (KIT), fanden heraus, dass Zinkoxid-Pulver als sogenannter Zufallslaser arbeiten kann. Es nimmt dabei Licht auf, das aber aufgrund der Reflexion an den zahlreichen Bruchkanten der Oxid-Körnchen dieses nur schwer wieder verlassen kann. Das kann dazu führen, dass die Oxid-Kristalle nun selbst als Laser arbeiten und so UV-Laserlicht ausstrahlen.

Ich vermute, dass dies auch bei meinen Versuchen mit Stahl-Schlüsselanhängern,

Inisch erprobt und bewährt utverträglichkeit dermatologisch bestätigt utverträglichkeit dermatologisch bestätigt Petrolatum, Zinc Oxide, Lanolin, Taic, Aqua, Panthenol, Hansmells Virginiana, Alantoin, Sorbitan Sequioleate, Certybridenne Chende, Paraffinum Liquidum, Circine Seja, Alcohol denai, Partum, CirSiso- (PPT0647) 150 ml e Million Million Million Bei Zimmertemperatur Zinkoxid ist einer der Hauptinhaltsstoffe. Penaten-Creme: Wer hätte gedacht, dass darin Laser stecken.

einem 40-W-CO₂-Laser und Penaten-Creme stattfand. Jedenfalls erzielte ich damit eine deutliche Wirkung: Der Stahl ließ sich nicht nur gravieren, sondern nahm dabei soviel Energie auf und erhitzte sich so stark, dass sogar an der Rückseite noch deutliche Spuren der Gravur sichtbar waren.

Auch Sprühöl mit MoS₂ aus dem Autozubehörhandel zeigt eine Wirkung, aber deutlich geringer.



Der obere Schlüsselanhänger wurde mithilfe von Penaten-Creme gelasert. Infolge ungleichmäßiger Schichtdicke war die Wirkung rechts deutlich stärker. Unten wurde Molybdän-Disulfid-haltiges Sprühöl verwendet. Der Effekt war deutlich schwächer, aber gleichmäßiger.

FREITAG IST C'T-TAGI*



*Endlich Wochenende! Endlich genug Zeit, um in der c't zu stöbern. Entdecken Sie bei uns die neuesten Technik-Innovationen, finden Sie passende Hard- und Software und erweitern Sie Ihr nerdiges Fachwissen. **Testen Sie doch mal unser Angebot: Lesen Sie 5 Ausgaben c't mit 30 % Rabatt – als Heft, digital in der App, im Browser oder als PDF. On top gibt's noch ein Geschenk Ihrer Wahl.**

Jetzt 5 × c't lesen für 23,25 € statt 30,25 €*

** im Vergleich zum Standard-Abo

Jetzt bestellen:

ct.de/meintag



Autofokus nachrüsten

Die Schneidleistung von Lasercuttern hängt sehr davon ab, ob sich die Schnittfläche im Brennpunkt des Laserstrahls befindet. Bei billigen Geräten ist da manuelles und daher ungenaues Einstellen üblich. Hier erfahren Sie, wie man einen Lasercutter mit motorisch verstellbarem Arbeitstisch auf Autofokus umrüstet und so immer auf den Punkt kommt.

von Heinz Behling



ochpräzise schneiden oder auch Bilder und Logos gravieren, in diesen Disziplinen ist ein Lasercutter Spitze, vorausgesetzt, er ist richtig eingestellt. Schließlich funktioniert das alles nur dann, wenn man den Laserstrahl so bündelt, dass seine Energie beim Auftreffen auf das zu bearbeitende Material ausreicht um es zu verdampfen.

Dazu wird der ohnehin schon energiereiche Strahl mithilfe einer Linse gebündelt (siehe Kasten "Laseroptik" auf Seite 105), sodass sich die Energie in deren Brennpunkt auf einer Fläche mit nur etwa 0.1 mm Durchmesser konzentriert. Bei einem Laser mit 40 Watt Strahlleistung, ein gängiger Wert bei CO₂-Laserröhren, ergibt das eine Energiedichte von über 10 Kilowatt/mm². Dagegen ist die 24-cm-Kochplatte Ihres Elektroherds mit 2000 Watt Leistung eine kühle Angelegenheit, denn sie bringt es nur auf 44 Milliwatt/mm².

Aber diese hohe Energiedichte des Lasers wird nur im Brennpunkt der Linse erreicht. Die Oberfläche des Schnittmaterials muss sich daher genau dort befinden. Da die Materialien jedoch unterschiedliche Stärken aufweisen können, muss der Laser bei jedem Cut- oder Graviervorgang fokussiert werden. Dazu wird meist der Arbeitstisch des Cutters in der Höhe verstellt. Bei preiswerten Geräten geschieht dies per Hand durch zwei Taster, die mittels Motor den Tisch auf- und abwärts bewegen. Der Fokusabstand wird in der Regel durch eine Abstandslehre ermittelt, die zwischen die Spitze des Schneidekopfes und die Oberfläche des Materials gelegt wird. Passt die Lehre genau dazwischen, soll der Brennpunkt exakt auf dem Material liegen.

Voraussetzungen

Das kann man automatisieren, zumindest bei Lasercuttern, die ohnehin einen motorisch betriebenen Arbeitstisch (Bild 1) und einen Ruida-Controller (Bild 2) haben, was bei sehr vielen, preiswerten China-Geräten der Fall ist.

Auch in unserer Redaktion verrichtet solch ein Cutter seinen Dienst (siehe Titelbild des Artikels). Fokussiert wurde mithilfe eines kleinen, 17 mm hohen Plexiglasstücks als Fokuslehre.

Wie man Cutter umrüstet, die diese Voraussetzungen nicht erfüllen, haben wir ja bereits einmal anhand des K40 gezeigt (siehe Artikel, Modernisierung für den China-Laser" in der Kurzinfo).

Der Umbau bei unserem Redaktions-Cutter, der auch unter der Bezeichnung 7050 (wegen der Arbeitsbereichgröße von 70 × 50 cm) gehandelt wird, erfordert den Austausch des 230-V-Wechselstrommotors gegen einen Schrittmotor, sowie den Einbau eines entsprechenden Treiber-Moduls und eines Tastsensors am Laserkopf. Die Bauweise der billigen China-Geräte ähnelt sich sehr. Daher

Kurzinfo

6 bis 8 Stunden

Material

8-mm-Welle

DMA 860H

der Fokusebene

Kosten:

ca. 120 Euro

- » Autofokus im China-CO₂-Lasercutter per Schrittmotor und Höhensensor
- » Steuerung per Ruida-Controller und Lightburn-Software » Ermitteln der richtigen Fokusebene



Make 1/24, S. 24



gilt diese Anleitung auch für andere Geräte. Vielleicht müssen Sie an der ein oder anderen Stelle leichte Anpassungen vornehmen, insbesondere, wenn es um die Befestigung der Teile geht.

ähnliches (3 mm, zum Ermitteln

» Holzklötze etwa 5 cm hoch

Der Schrittmotor wird nicht mehr über die beiden Taster an der Geräteseite, sondern über das Menü und die Tasten des Ruida-Controllers im Lasercutter gesteuert. Dort gibt es auch den Menüpunkt "Autofokus". Außerdem ist dann auch die Bedienung vom PC aus zum Beispiel mittels Lightburn möglich.

Ein Sensor am Kopf schaltet durch, wenn er die Oberfläche des Schnittmaterials berührt. Das geschieht während des Autofokus-Prozesses: Dabei wird der Arbeitstisch durch



Bild 1: Der 230V-Motor der Tischverstellung ist hinter der Heckklappe des Lasercutters gut versteckt.



Bild 2: Sieht das Display Ihres Lasercutters so oder ähnlich aus, ist meistens ein Ruida-Controller eingebaut. Weiteres Erkennungszeichen ist die Steuersoftware RDWorks.

den Motor nach oben gefahren, bis der Sensor durchschaltet. Jetzt ist dem Controller die Höhe der Schnittfläche bekannt und er fährt den Tisch wieder ein Stück nach unten, bis die Schnittebene auf Höhe der Fokusebene liegt. Wieweit heruntergefahren werden muss, ermitteln wir später nach Einbau der Hardware einmal und speichern den Wert dauerhaft in der Firmware des Controllers.

Nun zu den Neuteilen (Bild 3): Der Motor muss ein kräftiges Exemplar (Typ 57HS22) sein, der genügend Kraft entwickelt, um den Tisch anzuheben. Dazu braucht er einen Betriebsstrom von 4 A und hat annähernd die Maße des alten Motors und den gleichen Achsdurchmesser. Dem hohen Strom entsprechend ist auch das Treibermodul DMA860H etwas größer als die bereits im Cutter eingebauten und hat sogar einen eigenen Lüfter an der Rückseite.

Der Sensor ähnelt einem Kugelschreiber und hat eine ringförmige Halterung mit 24 mm Durchmesser, mit der er am Laserkopf befestigt wird. Zusätzlich habe ich mir noch ein Riemen-Zahnrad für die 8-mm-Achse des Schrittmotors sowie einen Näherungssensor besorgt, der eingebaut werden kann, um ein zu tiefes Absenken des Tisches zu verhindern. Wie sich aber später bei der Konfiguration der Firmware herausstellte, ist der nicht unbedingt nötig, da der maximale Verfahrweg des Tisches auch in der Konfiguration begrenzt werden kann.



Bild 3: Die Teile des neuen Tischantriebs: Steppertreiber-Modul (hinten) und Schrittmotor (rechts), davor der Sensor für den Laserkopf, das neue Zahnrad und links der Näherungssensor für die untere Wegbegrenzung

Austausch-Motor

Nun zum Umbau: Bevor Sie beginnen, müssen Sie eventuell noch einen Adapterring (siehe Bild 3) für die Sensorhalterung mit dem Lasercutter herstellen. Seine Halterung hat einen Durchmesser von 24 mm. Ist der Durchmesser des Laserkopfes geringer, brauchen Sie diesen Ring. Über den Kurzinfolink können Sie Schnittdateien für verschiedene Ringgrößen erhalten. Schneiden Sie den passenden zu. Als Material ist Plexiglas zu empfehlen. Es geht aber auch mit anderem, wenn es nur quetschstabil und bruchfest genug ist.

Fahren Sie danach den Arbeitstisch wie bislang mithilfe der beiden Taster nach unten, damit Sie genügend Platz haben, um am Laserkopf zu arbeiten. Dann schalten Sie den Lasercutter aus und ziehen den Netzstecker. Auch der Ausbau der vorderen, hinteren und rechten Geräteklappen ist ratsam. Dazu muss nur einer der jeweiligen Achsstifte nach innen geschoben werden.

Der Halter des 230-V-Motors ist mit vier Schrauben (Inbus) am Bodenblech des Lasercutters verschraubt (Bild 4). Drehen Sie die heraus und schrauben Sie auch den Motor vom Halteblech ab.

Vom Motor führen drei Drähte ins Elektronikfach des Cutters. Die sollten Sie nicht einfach abschneiden, denn sie führen Netzspannung und können bei Berührung anderer Teile heftige Reaktionen (Kurzschluss) hervorrufen. Zwei der Drähte führen zu den beiden Tasten und sind dort mit Steckern befestigt. Die müssen Sie einfach nur abziehen. Der dritte Draht führt zum Netzspannungsanschluss am 24-V-Netzteil. Verfolgen Sie ihn vom Motor aus in diese Richtung. Irgendwo im Kabelkanal des Elektronikfachs ist er mit einem anderen Kabel verbunden. An dieser Stelle schneiden Sie die Motorader durch und isolieren das blanke Ende des anderen Kabels mit Isolierband. Den Motor samt Kabelstrang können Sie dann aus dem Gerät entfernen.

Der Schrittmotor passt in die Öffnung des alten Halteblechs. Leider stimmen aber die Schraubenlöcher nicht überein. Daher müssen neue gebohrt werden (5 mm, Bild 5). Durch die Schraubenlöcher am Motor kann man deren Lage recht gut am Halteblech anzeichnen.

Achten Sie darauf, dass die Achse des Schrittmotors in der Mitte der Öffnung im Halteblech liegt. Dann passt nachher auch die Länge des Zahnriemens. Deshalb habe ich auch das Zahnrad des alten Motors verwendet, denn das neue hat einen geringeren Durchmesser, was die Zahnriemenspannung erschweren kann. Zum Anschrauben des Schrittmotors können Sie die Schrauben des alten Triebwerks verwenden, brauchen aber passende Muttern dazu (Bild 6).

Den Motor setzen Sie dann an die Position des alten Motors, achten darauf, dass der



Bild 5: Die neuen Löcher dürfen nicht zu dicht an den alten liegen, sonst reißt das Blech.



Bild 4: Der alte Motor muss raus, das Halteblech wird weiterverwendet. Achtung: An der Achse läuft gerne die Fettfüllung heraus.

Zahnriemen um das Zahnrad geführt ist, und schrauben das Halteblech fest. Der Verkabelung widmen wir uns später.

Steppertreiber

Als nächstes kommt das Stepperdriver-Modul in den Lasercutter. Vor dem Einbau stellen Sie die DIP-Schalter ein, und zwar auf diese Stellungen:

- 1 off,
- 2 off,
- 3 on,
- 4 off,
- 5–8 on.

Dadurch wird der Treiber auf einen Motorstrom von 4,45 A und 400 Schritte/Umdrehung eingestellt.

Das Modul ist recht groß und passt nicht auf den freien Platz vor dem Laser-Netzteil (Bild 7). Andererseits ist links von den beiden schon vorhandenen, kleineren Modulen noch Platz. Allerdings passt das neue Modul auch dort nicht. Ich habe daher die beiden kleineren Module etwas nach links gesetzt, sodass zwischen ihnen und dem Laser-Netzteil genügend Platz für das neue große Modul entstand. Für alle drei Module müssen dann aber neue Löcher gebohrt werden.

Die Schraubenbefestigung in den Billig-China-Lasern ist übrigens eine typische Sparlösung: Die Löcher werden etwas knapp im Durchmesser gehalten und die Schrauben (ganz normale M4-Gewindeschrauben) müssen sich ihr Gewinde beim ersten Hineindrehen selbst schneiden. Dieses Verfahren habe ich leider auch anwenden müssen, da das Bodenblech unter dem Elektronikfach doppelwandig ist. Also die Löcher mit einem 3,5-mm-Bohrer bohren und die M4-Schrauben mit etwas Kraftaufwand (daher wohl die Inbus-Schrauben) dort hineinzwingen. Übrigens ist dabei ein kleiner Magnet sehr hilfreich, mit dem man die beim Bohren entstehenden Metallspäne wieder sicher aus dem Elektronikfach herausfischen kann. Schließlich sitzen alle drei Module im Gerät (Bild 8).

Anschließend folgt die Verkabelung entsprechend dem Schaltplan (Bild 9). Beachten Sie, dass jeweils zwei der acht Motoranschlüsse miteinander verbunden werden.

Achtung: Online-Händler versenden statt des DM860H-Treibermoduls auch gerne einmal einen DM860A. Der kann zwar auch verwendet werden. Allerdings ist die Belegung der Anschlüsse anders. Orientieren Sie sich dann an der Beschriftung. Außerdem muss bei diesem Modul unbedingt auf die Polung der Versorgungsspannung geachtet werden, denn es hat einen Gleichstromeingang, während die H-Ausführung mit Gleich- und Wechselstrom zurechtkommt. Bei Gleichspannung ist der H-Variante die Polung egal.

Sensoreinbau

Dazu sollten Sie zunächst den stiftförmigen Sensor aus seiner Halterung herausnehmen. Er wird lediglich durch eine kleine Inbusschraube gehalten. Der Halter wird am Laserkopf befestigt, und zwar direkt unter dem dreieckigen Spiegelhalter (Bild 10). Zur Montage müssen Sie die untere Röhre, in der die Linse sitzt, aus dem Laserkopf entnehmen. Sie ist mit einer Rändelschraube befestigt. Den Luftschlauch können Sie nach Niederdrücken des blauen Rings am Anschluss herausziehen.

Die Öffnung des Halters hat einen Durchmesser von minimal 24 mm. Die Röhre unter dem Spiegelhalter des Laserkopfes kann je nach Hersteller zwischen 20 und 24 mm liegen. Dementsprechend kann, wie bereit erwähnt, ein Adapterring notwendig sein. Über den Kurzinfo-Link können Schnittdateien für Adapter mit 20, 21, 22 und 23 mm Innendurchmesser

Konfiguration in RDWorks

Falls Sie statt Lightburn noch die Original-Software RDWorks benutzen, können Sie auch dort die Firmware des Lasercutters konfigurieren. Sie finden Sie unter "Datei/ Werkseinstellungen". Danach müssen Sie das Passwort eingeben, dass bei allen Geräten RD8888 lautet.

Schließlich gelangen Sie ins Konfigurationsfenster. Vor dem schlecht übersetzten "enalbe HOME" und vor "Aktivieren Sie den Grenz…" muss ein Häkchen gesetzt werden. Statt "Step length" verwendet RDWorks den Begriff "Schritt L?nge(um)" (Tut mir leid, für die Fehler in RDWorks kann ich nichts). Immerhin wird "Home offset" auch hier so genannt.

RDWorks stellt allerdings keinen Fokustest zur Verfügung. Online gibt es aber eine Reihe von Testmustern, die den gleichen Zweck erfüllen. Die Download-Adresse erfahren Sie über den Kurzinfo-Link.

Ich empfehle Ihnen aber, stattdessen Lightburn zu benutzen. Die kostenlose Testversion können Sie 30 Tage ohne Einschränkung verwenden. Zeit genug, um den Autofokus einzubauen und zu konfigurieren.



Bild 6: Der Steppermotor muss zentrisch auf dem Halteblech sitzen und die Verzahnung des Zahnrades vom Motor weg weisen. heruntergeladen werden. Aber den passenden Ring haben Sie ja (hoffentlich) bereits gelasert.

Bei der Montage des Halters geht es eng zu, wenn am Laserkopf auch noch ein roter Positionier-Laser sitzt. Ideal sitzt der Sensorstift, wenn dessen Spitze auf den Rahmen des Wabentisches zeigt, falls sich der Laserkopf am Nullpunkt der Arbeitsfläche befindet. Dadurch ist gewährleistet, dass der Sensor später bei einem Autofokus nach einem Reset nicht mit der Tastspitze in ein Loch des Wabentisches fährt und so nicht schalten kann! Falls es zu eng wird am Laserkopf, würde ich auf den ohnehin ungenauen Positionier-Laser verzichten.

Dann setzen Sie den Sensorstift wieder ein. Schrauben Sie ihn noch nicht fest. Setzen Sie erst die zuvor entnommene Linsenröhre wie-



Bild 7: Das dicke neue Treibermodul passt nicht vor das Lasernetzteil.

der ein und schrauben die fest. Jetzt muss die Montagehöhe des Stiftes eingestellt werden. Die Unterseite des Taststifts muss unterhalb der Laseraustrittsöffnung liegen, aber oberhalb der Brennpunktebene. Das hört sich jetzt kompliziert an, ist aber relativ einfach: Nehmen Sie die bisherige Fokussierhilfe. Halten Sie diese unter die Laseröffnung. Die Tastspitze stellen Sie so ein, dass sie zwischen Ober- und Unterkante der Fokussierlehre liegt (Bild 11). Schrauben Sie den Sensor fest – fertig.

Kniffeliger ist das Verlegen des Sensorkabels. Es ist etwa 2 Meter lang und muss durch die beiden Kabelketten (die kleine vom Laserkopf zur Y-Achse und die größere von der Y-Achse zum Rahmen) hindurchgefädelt werden. Leider ist insbesondere in der kleinen Kette kaum Platz, da dort bereits das Kabel für den Positionierlaser und der Luftschlauch liegt. Ich habe daher die schwarze Isolierung des Sensorkabels auf kompletter Länge entfernt. Die Enden der drei Drähte habe ich abisoliert und an ein etwa 50 cm langes Stück steifen Schaltdraht (1 mm²) gelötet. Diesen Schaltdraht habe ich dann als Einfädelhilfe genutzt und das Kabel damit durch die Ketten bekommen. So war es recht einfach, das Kabel bis zum Elektronikfach durchzufädeln.

Dann können die drei Adern an den Controller angeschlossen werden. Der braune Draht kommt an +24V, der blaue an GND und der schwarze an LmtZ- (Bild12).

Damit ist der Einbau des Autofokus bereits fertig. Jetzt geht es ans Testen und Konfigurieren.

Erster Test

Verbinden Sie den Lasercutter wieder mit dem Stromnetz und schalten Sie ihn ein. Der Reset wird durchgeführt und der Laserkopf



Bild 8: Die neue Platzverteilung im Elektronikfach



Bild 10: Der rote Sensorhalter muss mit seiner Oberseite direkt am Spiegelhalter anliegen.



Laseroptik

Sind Sie mit der Schärfe Ihres Lasercutters nicht zufrieden, sollten Sie dessen Linse kontrollieren. Als erstes geht es um Sauberkeit: Eine Linsenoberfläche, die mit Ruß oder sonst wie verschmutzt ist, zerstreut das Laserlicht. Der Strahl wird dadurch breiter und die Schärfe nimmt drastisch ab. Schlimmer noch: Die Schmutzschicht absorbiert einen Teil der Laserstrahlung, was zu einer starken, punktförmigen Erhitzung der Linse führen kann. Solche Linsen können ernsthaft beschädigt werden (Verformung, Bruch). Daher sollten Sie die Linse regelmäßig reinigen. Für die Reinigung werden etliche Spezialmittel angeboten. Ich habe mit ganz normaler Spülmittellösung und einem Küchentuch die besten Erfahrungen gemacht.

Wichtig ist auch die Einbaulage der Linse: Meist sind plankonvexe Linsen eingebaut, die an einer Seite eben und an der anderen nach außen gewölbt (konvex) sind. Die konvexe Seite muss zum Laser, die plane zum Schnittgut zeigen.



Ist die Linse verkehrt herum eingebaut, findet an der planen Seite keine Lichtbrechung statt, da der Laserstrahl senkrecht auf die Fläche trifft. Folge: Es gibt keinen eindeutigen Brennpunkt.



Eine plankonvexe Linse in der richtigen Einbaulage bricht das Licht zunächst an der gewölbten Oberfläche und dann wegen des schrägen Einfalls nochmal an der planen Seite. fährt auf die Koordinaten 0,0. Bei meinem Gerät liegt das hinten rechts.

Tippen Sie dann gegen die Tastspitze des Sensors, drücken Sie sie leicht nach oben und lassen sie wieder los. Am Controller muss dann die LED 5 leuchten und wieder erlöschen. Falls nicht, kontrollieren Sie den Anschluss des Sensors am Controller und das Kabel auf



Bild 11: Das Einstellen der Sensorhöhe



Bild 12: GND und die Sensorleitung werden mit dem Anschluss CN3 verbunden. Die Versorgungsspannung +24V kommt von CN2.

Beschädigungen. Beheben Sie etwaige Fehler und wiederholen Sie den Test.

Sobald der LED-Test funktioniert, ist sichergestellt, dass der Controller das Sensorsignal erhält. Dann folgt der Test des Tischantriebs. Tippen Sie auf dem Bedienfeld auf Z/U. Mit den Pfeiltasten rechts und links können Sie den Tisch auf- und abwärts bewegen. Da die Firmware noch nicht konfiguriert ist, kann es dabei zu dem Effekt kommen, dass sich der Tisch nicht bewegt, der Motor aber laute Geräusche von sich gibt. In diesem Fall soll er sich mit zu hoher Geschwindigkeit bewegen. Das beheben wir später. Aber immerhin ist das ein Zeichen dafür, dass die Verkabelung stimmt.

Gibt der Motor hingegen keinen Mucks von sich und bewegt sich auch nicht, stimmt etwas mit den Verbindungen nicht. Schauen Sie am Treibermodul nach, ob an der Oberseite die grüne LED leuchtet. Wenn nicht, fehlt die Versorgungsspannung oder die Masseleitung. Ist die Spannung vorhanden, checken Sie die Verbindungen zum Controller und zum Motor entsprechend dem Schaltplan. Beheben Sie etwaige Fehler.

Bewegt sich schließlich bei Tastendruck der Tisch oder ertönen zumindest laute Geräusche, geht es weiter mit der Firmware-Konfiguration.

Software-Einstellung

Die wird in der Laser-Software Lightburn auf dem PC erledigt. Das ist aber auch mit der beim Lasercutter mitgelieferten Software RDWorks möglich (siehe Kasten auf Seite 103).

Falls noch nicht geschehen, schalten Sie den Lasercutter ein und verbinden ihn mit dem Computer. Starten Sie anschließend Lightburn. Mit "Bearbeiten/Maschineneinstellungen" gelangen Sie ins Konfigurationsfenster. Scrollen Sie darin ganz nach unten und klicken Sie auf den Pfeil vor "Vendor Settings" (Bild 13).

Die folgende Sicherheitsfrage beantworten Sie mit "ja". Dadurch wird das Konfigurationsfenster erweitert. Scrollen Sie bis zu "Z-Axis Settings". Zunächst muss die Autofokus-Funktion eingeschaltet werden. Klicken Sie auf den Schiebeschalter neben "Enable Homing". Auch bei "Limit Trigger" muss der Schalter auf grün stehen, sonst wird das Sensorsignal nicht beachtet. Bei "Home offest" ersetzen Sie den dort vorgegebenen Wert durch 5. Die restlichen Werte stellen Sie zunächst ein wie in Bild 14.

Die Einstellungen werden erst dann zum Lasercutter geschickt und dort gespeichert, wenn Sie auf "Schreiben" klicken. Notieren Sie sich noch den Wert, der bei "Step length (um)" steht, also im Beispiel 13,274999. Den Wert brauchen wir gleich. Das Fenster können Sie dann mit "ok" schließen.

Die Geschwindigkeit des Tischantriebs ist jetzt auf 5 mm/s eingestellt. Das ist ein brauch-

barer Wert bei dem von mir benutzten Gerät. Falls Ihr Lasercutter dabei immer noch Probleme bereitet (Geräusche statt Bewegung), muss dieser Wert verringert werden.

Legen Sie danach ein Stück Finnpappe oder Sperrholz auf den Tisch und führen Sie einen Autofokus durch: Drücken Sie am Cutter-Display auf "Z/U" und wählen Sie mit der Pfeil-nach-oben-Taste "Autofokus". Dann noch einmal auf "Enter" drücken und der Tisch setzt sich in Bewegung.

Als nächstes muss ermittelt werden, welche Weglänge der Tisch pro Schritt des Motors zurücklegt. Das geschieht mithilfe von zwei Holzklötzchen oder ähnlichen Gegenständen, die circa 5 cm hoch sein sollten. Deren genaue Höhe müssen Sie zum Beispiel mit einem Messschieber millimetergenau ermitteln! Im folgenden Beispiel gehe ich von 50 mm Höhe aus.

Legen Sie zunächst einen der Klötze so unter die Austrittsöffnung des Laserkopfes, dass sie genau aufliegt. Dazu müssen Sie den Tisch etwas nach unten fahren. Dazu tippen Sie am Cutter-Display auf Z/U und bewegen den Tisch mit der Pfeilrechts- bzw. Pfeillinks-Taste, bis der Klotz gerade unter die Öffnung passt. Auf dem Display wird die Z-Höhe nun in Millimeter angezeigt. Notieren Sie diesen Wert, z. B. 34,4 mm. Anschließend legen Sie den zweiten Klotz auf den ersten und fahren die Austrittsöffnung auf die Oberfläche des zweiten Klotzes. Auch dazu muss der Tisch nach unten bewegt werden. Wieder wird ein Z-Wert angezeigt, zum Beispiel 79,3 mm. Bilden Sie die Differenz der beiden Werte, also 79,3 - 34,4 = 44,9. Der Controller des Cutters ist also der Meinung, er hätte den Tisch um 44,9 mm nach unten bewegt, obwohl es in Wahrheit 50 mm waren

Mit einer einfachen Dreisatz-Rechnung ermitteln wir die korrekte Schrittweite:

alte Schrittweite / angezeigter Weg = neue Schrittweite / wirklicher Weg.

Aufgelöst nach der neuen Schrittweite ergibt dies:

neue Schrittweite = (alte Schrittweite × wirklicher Weg) / angezeigten Weg.

Im Beispiel ist die neue Schrittweite also (13,274999 \times 50) / 44,9 = 14,782850. Öffnen Sie in Lightburn erneut das Fenster mit den Maschineneinstellungen und tragen Sie diesen Wert bei "Step length (um)" ein. Übertragen Sie die neue Einstellung auf den Cutter und führen Sie dort einen Reset aus.

Als nächstes ermitteln wir, um wie viele Millimeter die Fokusebene unter dem Schaltpunkt des Sensors liegt. Legen Sie dazu die Finnpappe, das Sperrholz oder was immer Sie zum Ermitteln der Fokusebene benutzen möchten, in den Lasercutter ein. Wichtig ist, dass auf dem Material beim nun folgenden Lasern gut sichtbare Schnittspuren entstehen. Transparentes Material wie Plexiglas ist daher nicht geeignet. Die Materialstärke sollte etwa

×

4

?

Eigentum	Wert	Eigent	um	Wert
Speed factor % (0 to 200)	80		Max travel (mm)	500,000000
 Engraving Parameters 			Keypad jumpoff speed (mm/sec)	2,000
X start speed (mm/sec)	15,000		Keypad acceleration (mm/s^2)	3.000,000
Y start speed (mm/sec)	15,000		E-Stop acceleration (mm/s^2)	12.000,000
X acceleration (mm/s^2)	10.000,000		Home offset (mm)	0,000
Y acceleration (mm/s^2)	3.000,000	~ ~	Z Axis Settings	
Line shift speed (mm/sec)	150,000	1	Enable Homing	C Richtig
Facula Size (50 - 99%)	98,000		Limit Trigger	Richtig
Engraving factor % (0 to 100)	100		Invert Keypad Direction	Falsch
 Rotary Parameters 			PWM Rising Edge Valid	Falsch
Enable Rotary	Falsch		Limiter Polarity	Falsch
Pulses per rotation	10.000,000		Direction Polarity	Richtig
Diameter	100,000		Step length (um)	13,274999
 Miscellaneous 			Max speed (mm/sec)	5,000
Focus Distance	0,000		Jumpoff speed (mm/s^2)	5,000
X Axis Backlash (mm)	0,000		Max acceleration (mm/s^2)	1.000,000
Y Axis Backlash (mm)	0,000		Max travel (mm)	150,000000
Enable wireless panel speed shift	Falsch		Keypad jumpoff speed (mm/sec)	1,000
Wireless panel speed fast (mm/sec)	200,000		Keypad acceleration (mm/s^2)	1.000,000
Wireless panel speed slow (mm/sec)	50,000		E-Stop acceleration (mm/s^2)	10.000,000
Engraving Mode	Common Mode		Home offset (mm)	14.800
Return Position	Origin	~ ~	U Axis Settings	

Bild 13: Die wichtigen Konfigurations-Einstellungen stehen unter Vendor Settings.

Bild 14: Mit diesen Werten für die Z-Achse beginnen wir die Konfiguration der Controller-Firmware.

3 mm betragen. Auch Reste sind brauchbar, da für den Test nur eine Fläche von etwa 7 × 3 cm gebraucht wird.

Mit den Pfeiltasten am Bedienfeld des Cutters positionieren Sie den Laserkopf nun so, dass die Tastspitze des Sensors über dem Material liegt. Führen Sie jetzt einen Autofokus durch. Die Sensorspitze steht nun 5 mm über der Oberfläche der Finnpappe. Dort bleibt der Laserkopf stehen. Das ist die Null-Höhe, von der aus wir den Fokustest starten. Zur Kontrolle: Ihre bisherige Fokuslehre darf nun nicht mehr unter die Austrittsöffung passen. Falls doch, muss der Wert für "Home offset" in den Maschineneinstellungen verringert werden.

Im folgenden Test wird eine Folge von jeweils 20mm-langen Linien geschnitten. Nach jeder Linie wird der Tisch um 1 mm nach unten gefahren. Der Test beginnt bei der Null-Höhe des Sensors. Die liegt ja einige Millimeter über der bisherigen Fokusebene (siehe Bild 11).

In Lightburn kontrollieren Sie, ob "Starten von" auf "aktuelle Position"steht. Die Testfunktion finden Sie in Lightburn unter "Laserwerkzeuge/Fokustest". Die Werte im Testfenster habe ich wie in Bild 15 eingestellt.

Ein Klick auf Start setzt den Lasercutter in Bewegung. Der Vorgang dauert einige Minuten. Schauen Sie sich das gelaserte Muster genauestens an (Bild 16). Suchen Sie nach der schmalsten Linie mit der schärfsten Zahl darunter.

Die Zahl steht für den Abstand vom Sensor-Nullpunkt. Dazu müssen Sie noch den bereits eingetragenen Home-Offset addieren (5,0 mm) und erhalten so den Abstand der Fokusebene vom Schaltpunkt des Sensors. Im Beispiel sind es 10,5 mm. Diesen Wert tragen Sie schließlich in die Maschineneinstellungen als "Home offset" ein, schreiben alles in den Cutter zurück und sind fast fertig. Denn es fehlt noch die Begrenzung des Fahrweges nach unten. Bewegen Sie den Arbeitstisch mit, Z/U" und den Pfeiltasten nach unten, bis er einige Millimeter über der mechanischen Begrenzung steht. Lesen Sie dann den Z-Wert am Display ab und tragen Sie diesen Wert in die Maschineneinstellungen als "Max travel (mm)" ein. Das war es nun aber wirklich.

Künftig können Sie den Autofokus über das Lasercutter-Display wie bereits beschrieben auslösen. Es geht aber auch in Lightburn, wenn Sie das Bewegen-Fenster einblenden und darin auf Fokus klicken. Wichtig: Dabei muss unbedingt Schnittmaterial oder der Wabentischrahmen unter der Sensorspitze liegen! — hgb



Bild 15: Die Einstellungen für den ersten Fokustest



Bild 16: Das Schärfemaximum ist hier bei der Linie mit der 5.5.



Überwachungskamera

Auf dem Flohmarkt fiel mir eine sehr robuste Kamera auf, auch das Objektiv sah interessant aus. Also wurde sie gekauft und zerlegt.

von Carsten Wartmann

achdem ich mich schon mit CS-Mount-Objektiven für die moderne Raspberry-Pi-HQ-Kamera beschäftigt hatte (siehe Kurzlink), wusste ich, dass allein das Objektiv die fünf Euro wert sein kann, die der Händler für die ganze Kamera haben wollte. Ursprünglich dachte ich, das Objektiv wäre ein motorisiertes Zoom, aber leider ist nur die Blendensteuerung elektrisch. Immerhin ist es ein 6–12 mm Zoomobjektiv aus Japan mit brauchbarer Lichtstärke von 1:1.14 – zu dem später mehr.

Die Informationen über die Kamera (Grundig FAC 830 I) sind sehr spärlich. Irgendwo fand ich eine Pinbelegung und konnte die Kamera mit Strom versorgen, am Oszilloskop sah ich dann ein PAL-Videosignal. Damit war meine Neugier auf der Videoseite befriedigt. Auch wenn die Kamera damals mit ihrer Digitaltechnik sicher top war, ein analoges PAL-Signal ist heute nicht mehr wirklich brauchbar.

Innereien

Die Kamera stammt aus der Übergangszeit von analoger zu digitaler Videotechnik, der gesamte Signalweg vom CCD-Chip bis zur Steuerung von Belichtung und Blende ist digital. Am Ende wird dann analoges PAL-Video ausgegeben. Deshalb findet sich auf den Platinen auch ein Mikroprozessor, ein Derivat des Motorola MC6811. Apropos Platinen: Da die Elektronik für all das noch nicht genug integriert werden konnte, wurden mehrere Platinen mit Flexkabeln verbunden. Interessanterweise sind die Platinen ein Sandwich aus einer Flexplatine, die sowohl die Platinen verbindet, als auch Leiterbahnen enthält, und zwei normalen einseitigen Platinen. Das Sandwich ist fest miteinander laminiert, sodass eine große, faltbare Platine entsteht.

Die meisten Bauteile sind SMD-Bauteile, nur einige größere Kondensatoren, die Stecker und unerwarteterweise der CCD-Sensor (DIP) sind in Durchstecktechnik (THT) und somit gehen deren Beinchen durch das gesamte Sandwich. Durch Entlöten kann man eine ganze Handvoll, Hühnerfutter" und einige ICs, vor allem aus der 74er-Serie, weiterverwenden. Die Spezialchips von Sony sind eher nicht mehr zu gebrauchen. Hingegen ist der IR-Filter sicher noch brauchbar.

Objektiv

Das Objektiv ist nicht schlecht, nur wie steuert man die Blende? Es gibt Objektive, die ein Vi-


Anschlussseite und Zulassungs-Post-Adler

deosignal annehmen und die Helligkeitsinformation dort herausfiltern. Oder neuere Objektive, die eine Gleichspannung zur Steuerung benutzen. Wenn ich mir die Schaltung im Objektiv ansehe, erkenne ich eine einfache Op-Amp-Schaltung (wie es scheint, ein LM324). Ich vermute also eine DC-Steuerung. Versorgt man die Schaltung mit Strom, so öffnet sich die Blende. Legt man eine Spannung an den Steuereingang, schließt sich die Blende, eine prozentuale Öffnung habe ich aber noch nicht erreicht, auch die Funktion der beiden Potis für ACL und LEVEL sind mir noch unklar. Um das Objektiv zu testen, könnte man die Blende per Hand teilweise öffnen (abgeblendet sind die Abbildungsleistungenbesser) und den Mechanismus blockieren.

Und weiter?

Das massive Metalldruckgussgehäuse kann sehr gut wiederverwendet werden. Ein Raspberry Pi passt wie angegossen hinein, es ist noch genug Platz für Netzteil, Akku und dergleichen. Sogar die Kühlung des Raspis über das Gehäusemetall sollte einfach zu bewerkstelligen sein. Auch das HQ-Kameramodul passt, und die Mechanik zur Verstellung des Sensorabstands könnte weiter verwendet werden, sodass C- und CS-Objektive abwechselnd ohne Zwischenringe verwendet werden können. Eine Abdeckung für die Anschlüsse und eine WLAN-Antenne sind schnell 3D-gedruckt.

Vielleicht finde ich noch eine brauchbare Dokumentation für die Blendensteuerung, dann wäre ein Maximum der alten Kamera wiederverwendbar. Auf jeden Fall war es spannend und lehrreich, sich mit dieser Technik aus den Anfängen des digitalen Video-Zeitalters zu beschäftigen. -caw

make-magazin.de/xj2e



Schutzmaßnahmen beachten !

Note handling precautions !

GRUNDIG

Make: 2/2024 | 109

FRC 830 JA

000

Das Zoom-Objektiv mit CS-Mount



Der IR-Filter und der CCD-Chip

© Copyright by Maker Media GmbH

ALC LEVEL

Private Maker-Tools auf dem Pi

Immer mehr Software für Maker wandert in die Cloud. Wer sie benutzt, teilt aber auch persönliche Informationen und Ideen mit den Anbietern. Privat gehostete Dienste sind da schon sicherer. Mit den hier gezeigten Tools wandert die Maker-Toolbox in die private Cloud.

von Daniel Schwabe



Die hier gezeigten Tools werden in eine Docker-Umgebung auf einem Raspberry Pi installiert. Dafür wird das Docker-Frontend Portainer verwendet. Wie man beides installieren kann, ist in dem verlinkten Video in der Kurzinfo zu sehen. Wie man einen Raspberry Pi aufsetzt und über SSH bedient, steht in Make 5/23 auf Seite 72.

Programmieren im Browser

Visual Studio Code (VS Code) ist ein beliebter Texteditor, der mithilfe von Add-ons zu einer IDE (Integrierte Entwicklungsumgebung) für fast alle Sprachen erweitert werden kann. Mit der hier gezeigten Installation lässt sich VS Code als Webdienst installieren, den man später mit einem Browser aufrufen kann. Die dort erstellten oder bearbeiteten Dateien werden auf dem Server gespeichert und können von jedem Computer aus bearbeitet werden.

Als Erstes legt man einen neuen Ordner mit dem Befehl sudo mkdir /VS-Code an. In ihn werden später alle Dateien gespeichert, die man in VS Code erstellt. Als Nächstes installiert man das eigentliche Programm. Dafür geht man in Portainer auf Stacks und dann auf "Add stack". Dort trägt man in die obere Zeile einen Namen für den Stack ein und fügt darunter den Text aus dem Kasten "VS Code Composer-Datei" ein.

Das hier verwendete Image hat noch viele Optionen. Dazu findet man den Link zum Projekt in der Kurzinfo. Die Optionen PASSWORD und SUDD_PASSWORD müssen unbedingt angepasst werden.

Nach einem Klick auf "Deploy the stack" startet der Server. Unter <Raspberry-pi-IP>: 8443 ist jetzt Visual Code erreichbar. Dort loggt man sich mit dem erstellten Passwort aus der Zeile PASSWORD ein. Die anschließende Frage, ob den Dateien (in dem Ordner, den man bei der Installation festgelegt hat) vertraut werden kann, muss man bestätigen.

Jetzt erscheint die bekannte VS-Code-Oberfläche im Browser. Das Passwort, das man

VS Code Compose-Datei

container_name: code-server

/VS-Code:/config

- 8443:8443 restart: unless-stopped

PASSWORD=MeinStarkesPasswort

image: lscr.io/linuxserver/code-server:latest

SUD0_PASSWORD=MeinZweitesStarkesPasswort
DEFAULT_WORKSPACE=/config/workspace

version: "3" services: code-server:

> environment: - PUID=1000 - PGID=1000 - TZ=Berlin

volumes:

ports:

Kurzinfo

» Programmieren, CAD-Bearbeitung und mehr im Browser
 » Tools unabhängig vom Betriebssystem verwenden
 » Keine Bindung an große Cloud-Dienste



unter "SUDO_PASSWORD" festgelegt hat, ist für die Kommandozeile in VS Code notwendig. Das Terminal erreicht man oben links über das "Burgermenü/Terminal/New Terminal".

Mit ihm hat man direkten Zugriff auf die reguläre Kommandozeile des Docker-Containers. Wichtig hierbei ist, dass alles, was in dem Terminal geschieht, nur im Docker-Container passiert. Installierte Programme etc. verändern nicht das Host-System. Dieses Terminal verwendet man auch, um beispielsweise Unterstützungen für neue Programmiersprachen zu installieren, wie hier am Beispiel von Python erklärt: Als Erstes installiert man das Python Add-on für VS Code. Dafür klickt man auf das Add-on-Symbol auf der linken Seite und sucht nach Python. Das richtige Add-on ist das von ms-python. Durch einen Klick auf "Installieren" installiert man das Paket. Jetzt funktioniert schon das Hervorheben, Formatieren etc. von Python-Dateien (PY), aber das Ausführen noch nicht. Dafür muss man zuerst Python selbst in den Container installieren. Dafür geht man in das Terminal und führt als erstes sudo apt update aus. Bei der Passwortabfrage muss man hier das Passwort aus SUDO_PASSWORD eingeben. Danach installiert man Python mit sudo apt install python3. Anschließend kann man einen Rechtsklick auf eine PY-Datei machen und "Run Python File in Terminal" wählen.

*	local	
⊟	Dashboard	
C	App Templ	ates ~
8	Stacks	
♡	Containers	
≔	Images	
\$	Networks	
₿	Volumes	Das Stacks-Menü
0	Events	befindet sich links im Portainer-Menüband.
P	Host	~



Das Terminal verbindet einen direkt mit dem Docker-Container.

CAD im Browser

JSketcher ist ein Open-Source-CAD-Programm, das im Browser funktioniert. Damit kann man unabhängig vom Endgerät STEP-Dateien für 3D-Drucker erstellen. Außerdem ist es möglich, FreeCAD-Dateien zu importieren und weiter zu bearbeiten. Im Gegensatz zu VS Code werden bearbeitete Dateien allerdings nicht auf dem Server, sondern im lokalen Speicher des Browsers gespeichert, auf den das Tool dann immer wieder zugreifen kann. Man kann also nicht von unterschiedlichen Geräten auf die Projekte zugreifen.

Für dieses Programm gibt es keine vorgefertigte Docker-Compose-Datei und auch kein Docker-Image. Deshalb ist der Installationsvorgang etwas anders als bei Programmen mit Compose-File. Ein Image ist quasi eine Blaupause für einen Container. Darin sind alle Teile, die ein Programm braucht, vorkonfiguriert. Für VS Code gibt es ein fertiges Image, das dann mit einem Docker-Compose-File installiert werden kann. Für JSketcher erstellt man als Erstes ein neues, eigenes Image. Dafür klickt man auf der linken Seite in der Portainer-Menüleiste auf "Images" und dort dann auf der rechten Seite auf "Build a new image".

Im sich öffnenden Dialog gibt man im oberen Feld einen Namen für das Image ein. Dieser muss kleingeschrieben sein, z. B. jsketcher. Im großen Feld darunter trägt man die Anweisung zum Erstellen des Images aus dem Kasten "JSketcher-Image" ein.

Jetzt hat man ein Image, aus dem man nur noch einen Container erstellen muss. In der Image-Übersicht von Portainer gibt es eine Liste mit Images, die bereits auf dem System gespeichert sind. Der oberste Eintrag ist jetzt das JSketcher-Image, das aufgrund seines Namens direkt einen Tag bekommen hat: jsketcher:latest. Diesen Tag muss man kopieren (oder sich merken), über die Portainer-Menüleiste in die Container-Übersicht gehen und dort auf der rechten Seite "Add Container" klicken.

Auch hier gibt man im obersten Feld einen Namen ein, diesmal für den Container. Im Feld "Image" kopiert man nun den Tag des JSket-



VS Code kann auch in der Browser-Version Plug-Ins.

cher-Images, in diesem Fall jsketcher:latest. Dies sagt Portainer, aus welchem Image der Container erstellt werden soll. Im Bereich "Network Ports Configuration" muss man jetzt noch einstellen, über welche Ports das Tool erreichbar sein soll.

Configure Default Build Task...

Diese Option ist in zwei Felder geteilt: einmal in den Host-Port und einmal in den Container-Port. Was genau Ports sind, kann man im Artikel auf Seite 116 nachlesen. JSketcher hört auf den Port 3000. Deshalb gibt man auf der rechten, der Container-Seite, 3000 ein.

Auf der linken Seite muss stehen, über welchen Port der physische Rechner über den Browser angesprochen werden muss, um auf das Programm zuzugreifen. Das kann jeder freie Port sein. Wenn Port 3000 frei ist, kann man auch dort 3000 eintragen. Ist dieser Port aber schon belegt, kann man 3001 oder eine ganz andere Zahl eingeben. Für dieses Beispiel wird weiter Port 3000 verwendet. Nachdem das alles eingetragen ist, klickt man unten auf der Seite auf "Deploy the container".

Jetzt startet der Container und man kann unter <lp-des-Rasperry-PI>:3000 auf JSketcher zugreifen. Über diesen Weg öffnet man die klassische 3D-Ansicht. Gibt man <lp-des-Rasperry-PI>:3000/sketcher.html in die Browserzeile ein, kann man einen 2D-Sketcher nutzen. Wie man JSketcher bedient und welche einzelnen Funktionen es gibt, wird auf dem You-Tube-Kanal des Erstellers erklärt. Der Link zu diesem Kanal ist in der Kurzinfo zu finden.

Schaltungen im Browser

OpenCircuits ist ein Open-Source-Tool, um Schaltkreise zu designen. Es funktioniert ebenfalls im Browser und kann für unterschiedliche

🔮 Test.py

👻 Test.py

Ctrl+Enter

Ctrl+X Ctrl+C

Shift+Alt+C

Delete

Ctrl+K Ctrl+Shift+C

print

JSketcher-Image

FROM node:18-bookworm
WORKDIR /sketch
RUN apt update
RUN yes | apt install git
RUN git clone https://github.com/xibyte/jsketcher.git .
RUN npm install
CMD ["npm", "start"]

	···· 🐠 Test.py 🛛 X	
~ WORKSPACE	• Test.py	
Test.py	1 print©"Hallo Hake")	
	PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS (2)	
	eabc@7ddfdb8b59c7:-/workspace\$ /bin/python3 /config/workspace/Test Hallo Hake o abc@7ddfb8b59c7:-/workspace\$	

Und so erblickt das erste Python-Programm im Browser das Licht der Welt.

Anwender die Projekte direkt auf dem Server speichern. Auch bei diesem Programm muss man wieder ein eigenes Image bauen. Dafür geht man in der Portainer-Menüleiste auf Images und klickt ganz rechts auf "Build a new image". Im oberen Feld trägt man einen kleingeschriebenen Namen für das Image ein und im großen Feld darunter den Inhalt des Kastens "OpenCircuits-Image".

Danach klickt man unten auf "Build the Image". Wenn das Image fertig gebaut ist, erhält es einen Tag. Den kann man entweder der Image-Übersicht entnehmen oder man bastelt ihn sich selbst zusammen. Der Tag ist <Dervergebene-Image-Name>:latest.

Über die Portainer-Menüleiste geht man anschließend in die Container-Ansicht und klickt dort rechts auf "Add Container". Im sich öffnenden Fenster gibt man ganz oben einen Container-Namen ein und in der Zeile "Image" den Tag des erstellten Images.

Nun klickt man im Bereich "Network ports configuration" auf den grauen Button "publish a new network port" und es öffnet sich eine zweiteilige Eingabe. Auf der linken Seite trägt man den Port ein, der am Host-System für diesen Container festgelegt werden soll. Auf der rechten Seite trägt man 3000 ein. Auf diesen Port hört OpenCircuits. In diesem Beispiel trägt man auf beiden Seiten 3000 ein und klickt als Letztes auf "Deploy the Container". Jetzt kann man mit <Raspberry-pi-IP>:3000 auf OpenCircuit zugreifen.

Klickt man oben rechts im Fenster auf "Sign In", kann man sich mit einem Benutzernamen anmelden. Speichert man dann die Sketches über den grünen Save-Button oben links, kann man sich auch von anderen Computern anmelden, bei denen man sich mit demselben Namen einwählt. Interpreter können auch im Browser ausgeführt werden.

Private Code-Verwaltung

Gitea ist wie GitHub oder GitLab eine mächtige Code-Verwaltung. Im Gegensatz zu diesen Alternativen ist Gitea allerdings zum einen privat und zum anderen pflegeleicht.

Um Gitea zu installieren, legt man als Erstes einen neuen Ordner über die SSH-Verbindung

mit dem Raspberry Pi an. Der Befehl dafür lautet sudo mkdir /Gitea. In diesem Ordner werden später alle Gitea-Dateien gespeichert. Als Nächstes installiert man den Container über Docker-Compose. Dafür klickt man in der Portainer-Menüleiste auf Stacks und dann auf der rechten Seite auf "Add Stack". Im sich öffnenden Feld trägt man in der ersten Zeile einen

🚍 Images	
☐ Id⊥l Filter V	Tags
aha256:2b2259c2b0ea322010c16e5f3f7c0d	(jsketcherlatest)

Create container		
Name:	ag sylarana	
image configuration		
Regaty	Dicker Hub lanorymous	
image*	doctancia a.g. my image my tag	in Search
	di Image sprecis vejation.	
· Advanced marks		
Always put the image (3)		
O too are currently using an anarphone account to	pull images from Dasheemuk and will be imited to 100 pulk every 6 hours. You	ari configure Dockerfuly authentication in the Registres View. Remaining justs 190(190
Webhooks		
Create a container webhook ()	(Blancest Fasture)	
Network ports configuration		
Publish all exposed redwork ports to random hos	a paorite (D: C	
Manual network pert publishing ()		
Access control		
Bruille access contour (1)		
Administrators	this represents to activity the setting of the sett	Restricted read to website the newspectral of this resources to a set of users and/or forms
Actions		
Auto reviewa (*)		
Depin for contrast.		So sieht der Container-Dialog in Portainer aus.

Eine lange Nummer und ein Tag: Das sind die Identifikatoren eines Doker-Images.

EXPLORER

✓ WORKSPACE

Test.py

Open to the Side

Select for Compare

Open Timeline

Cut

Copy

Download..

Copy Path

Run Tests

Rename...

Debug Tests

Copy Relative Path

Delete Permanently

Run Python File in Terminal

Open in Integrated Terminal

Open With...

ſΩ

Ω

ç

RG

Д

OpenCircuits-Image

```
FROM node:18-bookworm
WORKDIR /circuit
RUN apt update
RUN yes
          apt install git
        RUN yes
          apt install yarn
        RUN yes | apt install golang-go
RUN export export PATH=$PATH:/usr/local/go/bin
RUN git clone https://github.com/OpenCircuits/OpenCircuits.git .
RUN varn
RUN yes | yarn build
RUN sed -i "49i StartServer();" ./build/scripts/start.js
CMD yes | yarn start
```



Von der Box zum umfangreichen Modell, JSketcher bringt CAD in jeden Browser.

Stack-Namen ein, beispielsweise Gitea. In das Feld darunter kommt der Inhalt des Kastens "Docker-Compose-Datei für Gitea".

Nach einem Klick auf "Deploy the stack" wird der Container installiert und Gitea startet. Mit <Raspberry-pi-IP>:3000 kommt man zur Weboberfläche, wo man weitere Einstellungen vornehmen kann. Die Interessanteste ist der Datenbanktyp. Dieser steht hier standardmäßig auf SQLite3. Für kleine, private Installationen wie dieser hier ist SQLite eine vollkommen valide Option. Für größere Operationen, die dann wahrscheinlich auch nicht nur auf einem kleinen Pi installiert werden, empfiehlt sich ein richtiger Datenbank-Server mit PostgreSQL.

Die Optionen "Gitea Basis URL" und Server-Domain werden automatisch mit dem Hostnamen des Computers ausgefüllt, auf dem Gitea installiert ist. Beides kann man zu einer Domain ändern – wenn man eine besitzt – die auf das lokale Netzwerk und diesen Server umleitet. Ansonsten kann man hier alles so lassen, wie es ist. Die Einstellungen wurden durch die Docker-Compose-Datei mit übertragen. Nachdem man die Daten bestätigt hat, kann man einen neuen User anlegen. Der erste angelegte Account wird zu einem Admin-Konto.

So, wie die Installation jetzt besteht, funktioniert das Klonen von Repositorys über HTTPS. Um SSH zu aktivieren, damit Authentifizierung beim Arbeiten mit den Repositorys nicht mehr über ein Passwort, sondern über einen SSH-Key geschehen kann, muss noch eine Option in der Gitea-Einstellungsdatei geändert werden. Dafür loggt man sich in den Raspberry Pi per SSH ein und führt den Befehl "nano /Gitea/gitea/conf/app.ini" aus. Jetzt öffnet sich ein Textdokument. In der Sektion Server fügt man eine neue Zeile ein, in die man "LFS_START_SERVER = true" schreibt. Mit STRG+O speichert man danach die Datei.

Jetzt muss der Container neu gestartet werden. Dafür geht man in Portainer auf die Container-Übersicht und klickt auf den Gitea-Container und in der oberen Zeile auf Restart. Der Container startet daraufhin neu und jetzt funktioniert das Klonen sowohl über SSH als auch HTTPS.

Bildbearbeitung im Browser

Das Programm miniPaint ist ein umfangreicher Bildeditor. Er unterstützt mehrere Ebe-



Menual network port pu	blishing 🕥 🔹 publich a new network port						
host	a.g. 80, 80-88, (p.80 or (p.80-88 (optional)	•	container	e.g. 80 or 80-88	TCP	UDP	8

Port-Konfiguration beim manuellen Erstellen eines Containers.

nen, das Öffnen von Dateien und bietet zahlreiche Effekte.

Um miniPaint in einer Cloud zu installieren, erstellt man wieder ein neues Image. Dafür klickt man auf Images und dort auf der rechten Seite auf "Build a new image". In die erste Zeile kommt ein kleingeschriebener Name für das Image, in diesem Fall miniPaint. In das Feld darunter kopiert man den Inhalt aus dem Kasten "miniPaint-Image".

Danach muss man den Container einrichten. Dafür klickt man auf der linken Seite auf die Container-Übersicht und dort dann rechts auf "Add Container". In die oberste Zeile trägt man wieder einen Namen ein, diesmal für den Container. In das Feld "Image" kommt der Image-Tag, hier miniPaint:latest.

Im Bereich "Network ports configuration" klickt man auf den grauen Button "publish a new network port". Die Eingabe hier ist zweigeteilt. Auf der rechten Seite trägt man den Port ein, der im Container angesprochen werden soll, für miniPaint der Port 8080. Auf die linke Seite kommt der Port am physischen Computer, von dem dann auf Port 8080 in den Container umgeleitet wird. Ist Port 8080 noch nicht belegt, dann kann man hier auch 8080 eintragen. Als Letztes klickt man unten auf "Deploy the Container".

Jetzt kann man mit <Raspberry-pi-IP>:8080 das Tool im Browser aufrufen. Die Dateien, die in miniPaint erstellt werden, muss man manuell lokal speichern. Es wird nichts auf einem Server gesichert.

Calling Home

Alle hier vorgestellten Services laufen jetzt privat im heimischen Netzwerk. Will man diese Tools auch von außerhalb, also unterwegs benutzen, kann man das mit einer Domain, Port-Forwarding und einem Reverse-Proxy machen. Wie das geht, erfährt man im Artikel auf Seite 116. — das

miniPaint-Image

```
FROM node:18-bookworm
WORKDIR /miniPaint
RUN apt update
RUN yes | apt install git
RUN git clone https://github.com/
viliusle/miniPaint.git .
RUN npm install
CMD ["npm", "run", "server"]
```



Von der blinkenden LED bis zum Volladdierer: Logikgatter beherrscht OpenCircuits.

	Erstkonfiguration		
Wenn du Gitea in einem Docker-Co	ntainer nutzt, lies bitte die Dokumentation, bevor du irgendweiche Einstellungen veränder: Datenbankeinstellungen		
Gitea benötig	t MySQL PostgreSQL MSSQL SQLite3 oder TiDB (MySQL-Protokoll).		
Datenbanktyp *	SQLite3 -		
Pfad	/data/gitea/gitea.db		
	Dateipfad zur SQLite3 Datenbank. Gebe einen absoluten Pfad an, wenn Gitea als Service gestartet wird.		
	Allgemeine Einstellungen		
Seitentitel *	Gitea: Git with a cup of tea		
	Du kannst hier den Namen deines Unternehmens eingeben.		
Repository-Verzeichnis *	/data/git/repositories		
	Remote-Git-Repositories werden in diesem Verzeichnis gespeichert.		
Git-LFS-Wurzelpfad	/data/git/lfs		
	In diesem Verzeichnis werden die Dateien von Git LFS abgespeichert. Leer lassen, um LFS deaktivieren.		
Ausführen als	git		
	Der Nutzer unter dem Gitea ausgeführt wird. Beachte, dass dieser Nutzer Zugriff auf das Repository-Wurzelverzeichnis haben muss.		
Server-Domain *	Domain oder Host-Adresse für den Server.		
SSU-Conver.Bod			
330-361761-7011	222 Der Port deines SSH-Servers. Leer lassen, um SSH zu deaktivieren.		
Gitea-HTTP-Listen-Port	3000		
	Port. unter dem der Gitea-Webserver laufen soll.		
Gitea-Basis-URL*			
	Adresse für HTTP(S)-Klon-URLs und E-Mail-Benachrichtigungen.		

Die meisten Einstellungen sind schon aus dem Docker-Compose-File übernommen.

Reverse-Proxy auf dem Raspberry Pi

Wenn man zu Hause verschiedene Web-Dienste hostet – beispielsweise eine Website und eine private Cloud – dann braucht man eine Möglichkeit, diese Server von außerhalb des Netzwerks mit unterschiedlichen Domains anzusprechen. Mit der Software Caddy lassen sich mit verschiedenen Domains unterschiedliche Server im gleichen Netzwerk aufrufen.

von Daniel Schwabe



m Maker-Haushalt sammeln sich schnell mehrere Server an: Home Assistant für das Smart-Home, eine eigene Website, eine online IDE, eine eigene Nextcloud oder eine smarte Kaffeemaschine, die eine Website hostet, auf der man den Brühvorgang starten kann. Für alle diese Services braucht man eine eigene Domain, um sie aufzurufen – beispielsweise www.home-assistant.meine-Domain.de für das Smart-Home und www.portfolio.meine-Domain.de für die Website. Um auf diese Art und Weise auf alle diese Dienste von außen zugreifen zu können, braucht man zusätzlich noch einen offenen Port am Router und die Software Caddy als Reverse-Proxy zur intelligenten Weiterleitung aller Datenströme.

Was ist ein Reverse-Proxy?

Das World Wide Web – also im Endeffekt das Internet, das sich im Browser abspielt – funktioniert über zwei Ports: Port 80 für HTTP, also unverschlüsselten Datenverkehr, und Port 443 für HTTPS, den verschlüsselten Datenverkehr. Diese beiden Ports sind der Standard. Gibt man eine URL, zum Beispiel https://www.google.de in den Browser ein, versucht dieser automatisch über Port 443 auf den Server zuzugreifen, auf dem die Website liegt.

Gibt es nur einen Dienst im eigenen Netzwerk, auf den man vom Internet aus zugreifen will, ist das ganz einfach: Man öffnet im Router den Port 80 oder 443 und leitet alle Anfragen, die an diesen Port geschickt werden, auf den einen Server im Netzwerk um. Fritz!Box-Besitzer kennen das als Portfreigaben.

Sobald man aber mehrere Subdomains, etwa www, www2, auf unterschiedliche Server in seinem Netz weiterleiten will, kommt



Stacks ist Portainers Bezeichnung für Docker-Compose-Dateien.

Kurzinfo

- » Eigene Domain für zu Hause einrichten
- » Eigene Subdomains mit SSL-Zertifikaten für alle Services
- » Automatisches DNS-Update für Umleitung

auf eigene IP-Adresse



Mehr zum Thema

- » Daniel Schwabe, Verbindungen zum Raspberry Pi Server, Make 5/23, S. 72
 » Daniel Schwabe, Netzwerk-Sicherheit
- mit Raspberry Pi, Make 1/24, S. 100 » Video: Docker auf dem Raspberry Pi
- installieren (Einsteiger-Tutorial)



man mit üblichen Routern nicht weiter. Stattdessen braucht man ein Programm, das die Anfragen an diese beiden Ports annimmt und dann nach Subdomains sortiert, welche Anfragen an den jeweiligen Dienst im Netzwerk gehen. Dieses Programm nennt man einen Reverse-Proxy.

Installation

Die hier vorgestellte Installation von Caddy funktioniert über Docker und wird in der grafischen Docker-Oberfläche Portainer vorgenommen. Wie man Docker und Portainer auf einem Raspberry Pi installiert, erklärt das Video aus der Kurzinfo Schritt für Schritt. Wie man einen Raspberry Pi neu aufsetzt und ihn über eine SSH-Schnittstelle bedient, steht in Make Ausgabe 5/23 auf Seite 72.

Nachdem sowohl Docker als auch Portainer auf dem Raspberry Pi eingerichtet sind, stellt man als erstes eine SSH-Verbindung zu diesem Server-Pi her und erstellt einen neuen Ordner und eine neue Datei. Das geschieht mit folgenden Befehlen: sudo mkdir /caddy/ erstellt einen neuen Ordner im Root-Verzeichnis (die erste Ebene, auf der man Ordner erstellen kann) mit dem Namen caddy. Der Befehl sudo touch /caddy/Caddyfile erstellt eine neue leere Datei mit dem Namen Caddyfile in dem neu erstellten Ordner. Danach kann man den Docker-Container mit Caddy installieren.

Dazu öffnet man über <IP-des-Raspberrys>:9449 die Weboberfläche von Portainer, wählt auf der sich öffnenden Seite die lokale Docker-Umgebung aus und klickt auf der linken Seite auf Stacks, danach oben rechts auf "Add Stack".

In der obersten Zeile der neuen Eingabeaufforderung trägt man einen Namen ein, zum Beispiel caddy. In das große Feld darunter kommt der Text aus dem Kasten Docker-Compose-Datei für Caddy.

Danach muss man unten auf der Seite auf deploy klicken und warten.

Ports

Zwischen dem heimischen Computer und dem Internet steht der Router. Er managed alles, was innerhalb des Netzwerkes passiert, welche Daten rausgehen und welche reinkommen. Datenverkehr mit dem Internet passiert durch sogenannte Ports. Diese heißen so, weil sie wie Türen funktionieren. Standardmäßig sind die meisten von ihnen abgeschlossen. Wenn man aber einen Serverdienst im Netzwerk betreibt, kann man einen dieser Ports öffnen. Dadurch ist es dann möglich, Datenverkehr, der an diesem Port ankommt, auf einen bestimmten Computer im Netzwerk umzuleiten. Auf diesem Computer kann dann die Software für einen Dienst laufen. Man öffnet quasi an einer bestimmten Stelle das eigene Netzwerk, um Daten von außen zu empfangen.

Docker-Compose-Datei für Caddy version: "3" services: web: image: caddy container_name: "caddy" ports: '80:80' - '443:443' '8448:8448' volumes: /caddy/Caddyfile:/etc/caddy/Caddyfile - caddy-data:/data restart: unless-stopped networks: caddv-network networks: caddy-network: volumes: caddy-data:

Wie funktioniert das mit den Domains?

Als Nächstes brauchen wir eine Domain. Sie bestimmt am Ende, was in die URL-Leiste des Browsers eingetippt wird. Beispielsweise ist make-magazin.de eine Domain. So eine Domain kann man bei verschiedenen Anbietern im Internet für einen kleinen Beitrag im Monat mieten.

Hat man eine Domain gemietet, muss der Browser nur noch wissen, wie man sich mit einem Dienst in dieser Domain verbindet. Dafür ist das Domain-Name-System, kurz DNS, zuständig. Es liefert auf eine Anfrage etwa nach www.example.com eine IP-Adresse zurück. Wenn man eine URL in den Browser eingibt, fragt der Browser bei einem DNS-Server nach, welche IP zu der Domain in der URL gehört und setzt automatisch den richtigen Port (80 oder 443) ein. Das funktioniert im Endeffekt wie bei einem Telefonbuch.

Die URL https://google.de verweist am 07.03. 2024 beispielsweise auf die IP-Adresse IPv6 2a00:1450:4016:80c::2003. Genau so muss jetzt die IP-Adresse des eigenen Netzwerks bei einem DNS-Server hinterlegt werden. Allerdings ändert sich diese IP immer mal wieder, weil Internet-Provider meist nachts eine Zwangstrennung der Verbindung durchführen. Beim Wiederaufbau bekommt der Router dann meist eine neue öffentliche IP-Adresse zugewiesen.

Einige Router haben eine Funktion eingebaut, um die eigene IP-Adresse im öffent-

	FRITZ!	FRITZ!Box 7590
۵	Übersicht	Internet > Freigaben
۲	Internet ^	Portfreigaben FRITZIBox-Dienste DynDNS VPN (IPSec) VPN (WireGuard)
Onlin Zuga Filte	Online-Monitor Zugangsdaten Filter	Über einen DynDNS-Anbieter kann ihre FRITZIBox unter einem festen Domainnamen aus dem Internet erreicht wer der FRITZIBox regelmäßig ändert. Ihren Domainnamen und zugehörige Kontoinformationen erhalten Sie bei Ihrem Benutzeroberfläche Ihrer FRITZIBox sowie Anwendungen und Dienste einfach erreicht werden, sobald in der FRITZ eingerichtet wurden.
	Freigaben	
	MyFRITZ!-Konto	DynDNS benutzen
	DSL-Informationen	Geben Sie die Anmeldedaten für Ihren DynDNS-Anbieter an.
5	Telefonie	Update-URL
Ð	Heimnetz	
© WLAN		Domainname
1	Smart Home	
0	Diagnose	Benutzername
۲	System	AVM-Router unterstützen viele
" s i	Assistenten	DNS-Services. Allerdings immer nur für eine Domain.
(?)	Hilfe und Info	

lichen DNS abzulegen. Bei einer FritzBox kann man in der Weboberfläche fritz.box z. B. unter Internet/Freigaben/DynDNS die Zugangsdaten zum Domain-Anbieter eintragen, und der Router kümmert sich dann selbst darum, die IP-Adresse immer zu aktualisieren.

Allerdings funktioniert das nur für eine einzelne Domain. Hier geht es aber darum, mehrere Domains zu verwalten. Dafür kann man ein Programm wie ddns-updater verwenden. Auch ddns-updater lässt sich wieder über Portainer in einen Docker-Container installieren. Aber zuerst muss man mit

sudo mkdir /ddns-updater

wieder einen neuen Ordner anlegen und die Rechte des Ordners mit

sudo chown -R 1000 ddns-updater

anpassen, damit das Programm dort Dateien anlegen kann.

Für die Installation klickt man wieder auf "Stacks/Add Stacks", gibt einen Namen (diesmal z. B. ddns-updater) ein und trägt den Text aus dem Kasten "Docker-Compose-Datei für ddns-updater" in das große Feld darunter ein.

Nach einem Klick auf deploy startet das Programm und legt im vorhin erstellten ddnsupdater-Ordner eine Datei mit dem Namen config.json an. In der Kommandozeile des Raspberry Pi öffnet man diese Datei anschließend mit dem Editor nano, indem man das Kommando nano /ddns-updater/config.json eingibt.

In diese Datei muss man nun die Zugangsdaten für den Domain-Provider eintragen. Auf der Projektseite zum ddns-updater gibt es zu vielen Anbietern eine Beispielkonfiguration. Das Projekt ist in der Online-Info verlinkt. Mit der Tastenkombination STRG+O kann man das Dokument speichern. Eine Beispielkonfiguration für den Domain-Anbieter Strato ist im Kasten "Beispielkonfiguration Strato" abgedruckt.

Sind diese Daten eingetragen, kann man unter <IP-des-Raspberrys>:8000 eine Übersicht der Domains aufrufen, die mit dem Programm upgedatet werden.

Portfreigabe

Eine Domain zeigt jetzt also auf den Router des heimischen Netzwerks und auf einem Computer oder Server läuft jetzt schon Caddy. Jetzt muss man die beiden Ports 80 und 443 am Router auf dem Computer im Netzwerk freigeben, auf dem Caddy installiert ist. Das funktioniert auf jedem Router anders, auf einer Fritzbox beispielsweise über das Webinterface fritz.box.

Dort klickt man unter Internet/Freigaben/ Portfreigaben unten auf der Seite auf "Gerät für Freigaben hinzufügen" und wählt aus der

Workshop

Docker-Compose-Datei für ddns-updater

<pre>- PUBLICIP_DNS_TIMEOUT=3s - HTTP_TIMEOUT=10s # Web UI - LISTENING_ADDRESS=:8000 - ROOT_URL=/ # Backup - BACKUP_PERIOD=0 # 0 to disable - BACKUP_DIRECTORY=/updater/data</pre>	<pre>version: "3.7" services: ddns-updater: image: qmcgaw/ddns-updater container_name: ddns-updater network_mode: bridge ports: - 8000:8000/tcp volumes: - /ddns-updater:/updater/data environment: - CONFIG= - PERIOD=5m - UPDATE_COOLDOWN_PERIOD=5m - PUBLICIP_FETCHERS=all - PUBLICIP_HTTP_PROVIDERS=all - PUBLICIPV6_HTTP_PROVIDERS=all - PUBLICIPV6_H</pre>	
<pre>- ROOT_URL=/ # Backup - BACKUP_PERIOD=0 # 0 to disable - BACKUP_DIRECTORY=/updater/data</pre>	- PUBLICIP_DNS_TIMEOUT=3s - HTTP_TIMEOUT=10s # Web UI - LISTENING ADDRESS=:8000	
•	- ROOT_URL=/ # Backup - BACKUP_PERIOD=0 # 0 to disable - BACKUP_DIRECTORY=/updater/data	

Liste im Feld Geräte den Computer aus, auf dem Caddy läuft. Hat man dem Raspberry Pi einen Namen gegeben, wird dort in der Liste dieser Name angezeigt. Wenn nicht (oder sollte der Name aus irgendwelchen Gründen nicht auftauchen) kann man am Raspberry Pi mit dem Befehl ifconfig die aktuelle lokale IP-Adresse anzeigen. Diese kann man anschließend in ein Feld eingeben, wenn man in den Fritzbox-Einstellungen bei "Gerät" ganz nach unten scrollt und "IP-Adresse manuell eingeben" anklickt. Ist das Gerät ausgewählt, lässt man alle anderen Optionen wie sie sind und klickt unten auf "Neue Freigabe". Im sich öffnenden Dialog wählt man nun als Erstes Portfreigabe und dann unter Anwendung HTTP-Server aus. Dann wird alles automatisch ausgefüllt und man kann mit OK bestätigen. Danach macht man das Ganze noch einmal, diesmal allerdings unter "Anwendung" der HTTPS-Server. Dann wird wieder mit OK bestätigt.

Caddy konfigurieren

Jetzt hat man eine Domain, die auf das eigene Netzwerk zeigt und bei Anfragen über HTTP/HTTPS an den Caddy-Server weiterlei-

Beispielkonfiguration Strato { "settings": [{ "provider": "strato", "domain": "example.de", "host": "@", "password": "DasSicherePasswort", "ip_version": "ipv4", "ipv6_suffix": "",

```
"ipv6_suffix": "",
"provider_ip": true
}
```

] }

Ihr Windows-Ratgeber



- Hochsicherheits-Windows
- Notfallsystem bauen, Viren suchen, Probleme lösen
- Unerwünschte Treiber- und Firmware-updates verhindern
- Home- in Pro-Edition umwandeln
- Windows-Umzug mit c't-WIMage
- Whitelisting: Möglichkeiten, SRP/Restric'tor, AppLocker



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



🛱 shop.heise.de/ct-windows24

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



© Copyright by Maker Merke: 2/2024 | 119

Domain	Host	Provider	IP version	Update status	Set IP	Previous IPs (reverse chronological order)
example.de	@	Strato DNS	ipv4	Success (changed to 12.123.1.12), 15s ago	12.123.1.12	N/A

Über die Weboberfläche hat man den Status der Domains immer im Blick.

	Contra-	PRI12:B0X 7550	Hyperical Printanes :
• •	Obersicht Internet	Image: Constraint of the state of the s	?
	Online-Monitor	Gerät Raspberry-pi +	
	Zugangsdaten Filter	IPv4-Adresse 102.108.178.100	
	Freigaben	MAC-Adresse R8-27:58:95:58:54	
	MyFRITZI-Konto DSL-Informationen	iPv6 Interface-ID :: 191b : 8643 : fride : da61	
۲ ۶	Telefonie Heimnetz	iPv4-Einstellungen	
? (d)	WLAN Smart Home	 Dieses Gerät komplett für den internetzugriff über IPv4 freigeben (Exposed Host), Diese Einstellung kann nur für ein Gerät aktiviert werden. 	
0	Diagnose	IPvő-Einstellungen	
۲	System	PING6 freigeben. Firewall für delegierte IPv6-Präftxe dieses Gerätes offnen.	
.1	Assistenten	 Dieses Gerät komplett f ür den Internetzugriff über IPv6 freigeben (Exposed Host). 	
1	Hilfe und Info	Freigaben	
		Status Bezeichnung Protokoll IP-Adresse im Internet	Port extern vergeben
		Es sind keine Freigaben eingerichtet	Neue Freigab

Um in einer FritzBox eine Portfreigabe einzurichten, muss man erst das Gerät für Freigaben konfigurieren.

Subdomains

Hat man sich eine Domain gemietet, kann man sich für diese Domain sogenannte Subdomains sichern. Wenn man die Domain **meine-domain.de** besitzt, wäre **subdomain.meinedomain.de** eine Subdomain. Man kann sich vor die eigentliche Domain, abgetrennt mit einem Punkt, quasi eine neue URL erstellen, ohne eine neue Domain kaufen zu müssen. Wie viele Subdomains man nutzen kann, hängt vom Domain-Anbieter ab. tet. Diesen muss man jetzt noch konfigurieren: Dazu öffnet man die Caddyfile-Datei, die man am Anfang angelegt hat mit nano / caddy/Caddyfile. Falls am unteren Rand des Fensters in Rot, File is unwritable" steht, muss man die Datei mit STRG+X schließen und mit dem Befehl sudo nano /caddy/Caddyfile öffnen. Der Zusatz sudo gibt einem alle Rechte auf dem System und man kann so alle Befehle ausführen. Die Konfigurationen für alle Domains, die über diesen Server verwaltet werden sollen, kommt in diese eine Datei. Ein Konfigurationsblock sieht immer wie folgt aus:

```
Domain.de {
   tls <E-Mail-Adresse>
   reverse_proxy <ZIEL>
   }
}
```

Dieser Block ist wie folgt zu lesen:

Wird dieser Server mithilfe dieser Domain angesprochen, aktiviert die erste Zeile in der geschweiften Klammer HTTPS und legt eine E-Mail für das SSL-Zertifikat fest. HTTPS bedeutet dass der Datenverkehr zwischen Client (z, B, einem Browser) und dem Server (in diesem Fall der Server, auf dem Caddy installiert ist) verschlüsselt wird. HTTPS ist bei Caddv standardmäßig aktiviert. Die E-Mail-Adresse dahinter wird dann in das Zertifikat als Besitzer eingetragen. Dieses Zertifikat wird von einer vertrauenswürdigen Zertifizierungsstelle ausgestellt. Beispielsweise Let's Encrypt. Das stellt sicher, dass sich Verbrecher nicht selbst ein Zertifikat für eine gefälschte Website ausstellen. Solche Zertifikate müssen in regelmäßigen Abständen erneuert werden. Das macht Caddy aber von selbst. Wenn mal eine Domain nicht mehr über Caddy verwaltet wird, läuft das Zertifikat nach spätestens 90 Tagen von selbst aus. Diese Einstellung legt wirklich nur die eingetragene E-Mail fest. Die letzte Zeile ist die wichtigste. Hier wird festgelegt, was Caddy mit den Anfragen macht, die über diese Domain ankommen.

Der Befehl reverse_proxy bestimmt, dass der ganze Verkehr, der mit dieser Domain eröffnet wird, zu einem bestimmten Ziel im Netzwerk umgeleitet werden sollen. Das kann entweder eine lokale IP-Adresse sein – wenn sich der Zielserver auf einem anderen Computer befindet – oder der Name eines Dockercontainers auf demselben Computer.

Wenn das Ziel eine IP-Adresse ist, sieht die Zeile so aus:

reverse_proxy 192.168.178.xxx:Port

Port muss dann natürlich durch den Port ersetzt werden, auf den das Programm hört. Denn Programme überwachen nur bestimmte Ports, damit sie sicher sein können, dass Daten auch für sie bestimmt sind. Das ist immer verschieden, wird aber in den meisten Fällen in der Dokumentation der Programme erklärt.

Connected networks				
Join a network	Select a network		- (Itiko nationala)	
Network	IP Address	Gateway	MAC Address	Actions
bridge	172.17.0.3	172.17.0.1	02:42:ac:11:00:03	B Leave network

Über Portainer kann man ganz einfach Container zu Netzwerken hinzufügen.

MVERITZI-Freigabe			Caddynie beispier
 Portfreigabe 			<pre>home-assistant.meine-domain.de { tls max.mustermann@gmail.com reverse_proxy home-assistant:8123</pre>
Anwendung	HTTP-Server	•	<pre>} portfolio.meine-domain.de { tls max.mustermann@gmail.com</pre>
Protokoll	ТСР	•	reverse_proxy portfolio:80 } ich-brauche-kaffee.meine-domain.de {
Port an Gerät	80 bis Port 80		tls max.mustermann@gmail.com reverse_proxy 192.168.178.xxx:PORT
Port extern gewünscht	80		}
(Nur IPv4)			
Freigabe aktivieren			
Internetzugriff über IPv4	und IPv6		
O Internetzugriff über IPv4			
O Internetzugriff über IPv6			
	OK Abb	rechen	Mit einer Portfreigabe öffnet man ein kleines Türchen am

Muss der Datenstrom auf keinen neuen Port umgeleitet werden, sondern wird direkt von z. B. Port 80 auf Port 80 geleitet, kann man den Port in dieser Zeile auch weglassen.

Wenn sich das Programm, auf das umgeleitet werden soll, auf dem gleichen Computer in einem anderen Docker-Container befindet, ist das Ganze noch einfacher. Bei der Installation von Caddy wurde das virtuelle Netzwerk caddy_caddy-network erstellt. Das ist ein Netzwerk, in dem nur Docker-Container sein können. Dieses Netzwerk existiert lokal auf dem Server.

Diesem Netzwerk kann man ganz einfach neue Container über Portainer hinzufügen. Dafür klickt man auf der Portainer-Übersicht unter Container auf den Container-Namen und scrollt bis ganz nach unten. Dort kann man dann in dem Dropdown-Menü "Join a Network" das Netzwerk auswählen und mit einem Klick auf "Join Network" den Container hinzufügen.

Hat man das getan, kann man im Caddyfile ganz einfach reverse_proxy <Container-Name>:Port eintragen und kann so zu diesem Container umleiten. Den Container-Namen kann man in der Container-Übersicht in Portainer ablesen. Eine Konfiguration für das in der Einleitung aufgestellte Szenario würde dann aussehen wie im Kasten "Caddyfile Beispiel".

Was kann Caddy noch?

Caddy kann aber noch mehr, als nur den Traffic in einem Netzwerk zu verwalten. Es kann nämlich auch als Server für HTML-basierte Websites dienen. Dafür geht man in Portainer unter Container auf Caddy und klickt auf console.

Dort stellt man dann command auf /bin/ ash um und gibt unter user root ein. Nachdem man auf connect geklickt hat, befindet man sich innerhalb des Docker Containers. Jetzt geht man mit dem Befehl cd /data in das Datenverzeichnis von Caddy. Dort erstellt man dann mithilfe von mkdir /website einen neuen Ordner für die Website. Diesen mit cd /website betreten und dort caddy file-server --listen :8976 ausführen. Jetzt wird der Inhalt dieses Ordners mit Caddy auf Port 8976 bereitgestellt. Der Port kann in dem Befehl angepasst werden. Außerhalb des Docker-Systems befindet sich dieser Ordner im normalen Dateisystem des Raspi unter /var/ lib/docker/volumes/caddy_caddy-data/_ data/website. Wenn man dort eine index.html anlegt, hat man eine funktionierende Website. Möchte man auf diese Website über eine Domain zugreifen, ist die Konfiguration folgende:

```
caddywebsite.de {
    tls max.mustermann@gmail.com
    reverse_proxy localhost:8976
}
```

Außerdem kann man Caddy noch mit Add-ons erweitern. Die sind aber nur für sehr spezielle Anwendungen nützlich. Einen Link mit mehr Informationen zu diesen Add-ons ist in der Kurzinfo zu finden. —das

Execute	
Command	∆ /bin/ash
Use custom command	
User (?)	root
Connect	

Klickt man hier auf Connect, verbindet man sich mit der Konsole des Containers. Alle Eingaben geschehen nur in der abgesteckten Containerumgebung und betreffen nicht das Host-System.

Memory Maps verstehen

Die Verknüpfung von CPU, RAM und Flash ist in Mikrocontrollern oft trickreich, mitunter verwirrend. Wir helfen, die interne Verdrahtung und die Memory Maps besser zu verstehen.

von Daniel Bachfeld



n Ausgabe 1/24 haben wir eine Übersicht über verschiedene Speicherarten in modernen Mikrocontrollern veröffentlicht. In diesem Artikel erklären wir die Anordnung der Speicherbereiche und unter welchen Adressen Flash, RAM und die I/O-Bausteine zu erreichen sind.

Sowohl im PC als auch im Mikrocontroller dreht sich alles um die CPU: Sie benötigt Befehle aus dem Flash, RAM zum Ablegen von Daten und Peripherieeinheiten (aka I/O) zur Kommunikation mit der Außenwelt.

Als Maker kennt man bereits viele Kommunikationsbusse wie I²C, 1-Wire, SPI, I²S, CAN, UART, HDMI. Das sind aber nur die seriellen Busse für "nach draußen", zum Anschluss von Geräten, Sensoren, Displays usw.

Mit den Bussen zur Kommunikation mit den Speichern (mal abgesehen von den extern angeschlossenen QSPI-Flashs) und Peripherie-Einheiten kommt man als Maker nur noch selten in Berührung. Dennoch ist ein Grundverständnis der internen "Verdrahtung" und des Datentransports hilfreich, insbesondere wenn man sich mal etwas tiefer in Embedded Systeme einarbeiten will oder sei es nur, die An- und Ausgaben in der Arduino IDE besser zu verstehen.

Transportwesen

Früher war zwar nicht alles besser, aber vieles war übersichtlicher. Boomer erinnern sich gerne an die aus heutiger Sicht lachhafte Simplizität der 8-Bit-Heimcomputer der 80er Jahre. Deren Aufbau kam im Kern der Von-Neumann-Architektur im Bild 1 sehr nahe: Eine CPU ist mit ROM (heute eher Flash), RAM und I/O-Bausteinen über einen Adressbus, einen Datenbus und einen Steuerbus verbunden. Startet die CPU, weil jemand gerade den Strom angeschaltet hat, so geht es bei Adresse 0x0000 los. Das kleine x steht hier für die in solchen Fällen übliche hexadezimale Darstellung. Zur Erklärung oder als Erinnerung: Ein Nibble, also ein halbes Byte (mit den Werten 0 bis 15), lässt sich über die Ziffernfolge 0 bis F darstellen (A = 10, F = 15). Damit kann man 16-Bit-Adressen mit 4 Ziffern darstellen!

Die CPU legt das Bitmuster der Adresse auf seine 16 parallelen Leitungen des Adress-Busses. Erreicht die CPU unter dieser Adresse jetzt das ROM, den RAM oder die I/Os? Es kommt darauf an! Und zwar darauf, wie der Entwickler sein System aufgebaut hat. Eine nicht in Bild 1 eingezeichnete Steuerlogik (dazu gleich mehr) entscheidet, welchen Baustein die CPU zu sehen bekommt. Üblicherweise baut man sein System so auf, dass bei Adresse 0x0000 das ROM oder der Flash liegt. Ist das ROM beispielsweise 32 kByte groß, so kann die CPU einzelne Zellen zwischen 0x000 und 0x7FFF (32766) abfragen und über den Datenbus jeweils 8 Bit, also ein Byte, auslesen.

Geschickterweise beginnt der sogenannte Adressraum des RAMs direkt daran anschließend bei 0x8000 (32769). Wäre das RAM nun ebenfalls 32 kByte groß, würde man seine Speicherstellen von 0x8000 bis 0xFFFF (65535) erreichen. Leider wäre nun keine Adresse mehr für I/O frei. Deshalb wählt man das RAM beispielsweise nur 16 kByte groß (0x8000 bis 0xBFFF). Die I/O-Bausteine wären dann ab 0xC000 erreichbar.

Im Unterschied zu ROM und RAM dienen die I/O-Bausteine in der Regel nicht zum Speichern von Daten oder Lesen von Befehlen, sondern zum Konfigurieren von Registern und der Ein- und Ausgabe. Die Konfigurationsregister legen beispielsweise fest, ob GPIOs als



- » Bussysteme verstehen
- » Memory Maps interpretieren
 » Unterschiede beim ARM, ATmega und ESP



Ein- oder Ausgang arbeiten sollen, ob intern Pull-Down-Widerstände geschaltet werden sollen usw. Man kennt das vom Arduino und Raspberry Pi. Die einzelnen Register sind über Adressen erreichbar, in die die CPU schreibt oder aus denen sie liest, etwa den Status eines Pins oder den Wert eines ADC-Wandlers.

Aneinandergereiht sieht dann die sogenannte Memory Map wie in Bild 2 aus. Insgesamt kann die CPU 65536 (2¹⁶) Speicherstellen adressieren.

Weichenstellung

Mit welchem Baustein (RAM, ROM, I/O) bei welcher Adresse die CPU gerade sprechen

22	20
OxFFFF	0 13
I/O	444 4
0xC000	1100 0000 0000 0000
OxBFFF	<mark>10</mark> 11 1111 1111 1111
RAM	
0x8000	1000 0000 0000 0000
0x7FFF	<mark>01</mark> 11 1111 1111 1111
ROM	
0x0000	<mark>00</mark> 00 0000 0000 0000

Bild 2: Die Übersicht der Speicherstellen (Memory Map) des Mikrocontrollers aus Bild 1.

A14	A15	IC
0	0	ROM
0	1	ROM
1	0	RAM
1	1	I/O

Bild 3: Je nach Kombination der Adress-Bits werden einzelne ICs aktiviert oder deaktiviert.



Bild 1: Der Klassiker! Ein Mikrocontroller nach der von-Neumann-Architektur

darf, entscheidet ein sogenannter Adress-Dekoder. Er besteht aus einer Logikschaltung (früher gerne der TTL-IC 74LS138) der anhand der Bitkombinationen auf den Adressleitungen den richtigen IC ein- und ausschaltet. Wir erinnern uns: die Leitung Chip Select oder Chip Enable veranlasst einen (diskreten) IC, virtuell seine Beinchen zu heben und quasi nicht mehr am Adress- und Datenbus zu hängen. So können mehrere ICs den gleichen Bus nutzen, ohne sich ins Gehege zu kommen. Schaut man sich die Adressbits in Bild 2 an, so fällt auf, dass die Bits A0 bis A13 wiederkehrende Muster annehmen, A14 und A15 aber hochzählen und zwar 00, 01, 10, 11 also isoliert betrachtet von 0 bis 3. Das entspricht 4 × 16 kByte, unserem Adressraum von 64 kByte. Die Logiktabelle und die Zuordnung der IC ist in Bild 3 zu sehen.

Diese Art der Weichenstellung ist bei 16-Bit-Systemen noch ähnlich, wenn auch mit größerem Adressraum. Bei modernen Systemen



Bild 4: Die Advanced Microcontroller Bus Architecture (AMBA), wie sie ARM in seinen SoCs einsetzt.



Bild 5: Am Beispiel des RP2040 im Pi Pico sieht man, wie kleinteilig der innere Aufbau ist.

Bit-Banding-Berechnung

Die Bit-Banding-Adresse berechnet sich nach der Formel

bit_word_addr = bit_band_base + (byte_offset × 32) + (bit_number × 4)

Um beispielsweise Pin 9 des bei Adresse 0x40006040 liegenden Port-Eingangsregister B auszuwerten, liest man (per Pointer) folgende Adresse ein:

 $0 \times 420 \times 0824 = 0 \times 4200000 + (0 \times 40006040 - 0 \times 40000000) \times 32 + (9 \times 4)$

Auf den ersten Blick ist das etwas aufwändig in der Berechnung. Gießt man die Formel aber in C in ein Makro (per #define), wird der Einsatz von Bit Banding zum Kinderspiel. mit einer Memory Management Unit (MMU), etwa dem Raspberry Pi, ist die Lage erheblich komplexer, weil dort virtuelle Adressen ins Spiel kommen, die erst in echte physische Adressen übersetzt werden müssen.

Übrigens: Bei 16-Bit-Computern sind die internen Datenregister der ALU der CPU echte 16 Bit breit. NOR-Flash-Chips gibt es mit 8, 16 und 32 breitem Datenbus. Man könnte also 2 ICs mit je 8 Bit oder 1 IC mit 16 Bit anschließen. Analog gilt das für 32-Bit- und 64-Bit-Systeme.

Prinzipiell kann man die Reihenfolge der Speicherbereiche in einem System oder einem SoC beliebig festlegen und in der Praxis unterscheiden sich Mikrocontroller verschiedener Hersteller auch erheblich in der Abfolge der Bereiche. Das ROM kann beispielsweise bei 0x2000 beginnen, während das RAM bei 0x0000 anfängt. Wichtig ist nur, dass erstens die CPU beim Start (oder Reset) die richtige Adresse auf den Bus legt, um Befehle abzuholen und zweitens der Compiler und der Linker wissen, wo das RAM liegt, um Daten dort abzulegen. Ersteres müssen die Chip-Designer sicherstellen, zweiteres die Compiler-Bauer.

Busse

Das in Bild 1 gezeigte Bussystem besteht aus drei Einzelbussen für Adressen, Daten und



Bild 6: Die Memory Map eines ARM Cortex M3.

Steuerleitungen und es kann jeweils nur ein IC gerade über einen Bus mit der CPU kommunizieren. Gibt es jeweils Bussysteme für RAM und ROM, so spricht man von Harvard-Architektur. Tauchen die dennoch im gleichen Adressraum auf, so spricht man von modifizierter Harvard-Architektur, wie bei den ATmegas. Dort wird es aber noch spezieller: Denn diese als 8-Bit-Architektur angepriesenen Mikrocontroller sind eigentlich 16-Bitter. Die internen Register sind zwar nur 8 Bit breit. Dafür ist der interne Flash über einen 16 Bit breiten Bus angebunden, über den die CPU ihre 16 und 32 Bit breiten Befehle einliest. Das macht ATmega im Veraleich zu anderen 8-Bit-Mikrocontrollern schneller.

Genau genommen ist der Flash etwa beim ATmega328 zwar 32 kByte groß. Aber man kann nur 16384 Adressen aufrufen, weil immer zwei Byte (16 Bit) ausgegeben werden. Das SRAM im ATmega ist hingegen nur mit einem 8 bittigen Datenbus verbunden. Die I/O-Peripherie ist ebenfalls über einen separaten Bus angebunden.

Wildwuchs

Bei ARM-CPUs ist es etwas komplexer. Das gesamte Bussystem nennt sich Advanced Microcontroller Bus Architecture (AMBA) und wird auf SoCs zur Verbindung der vielen internen Komponenten verwendet. Ein typisches AMBA-basiertes System besteht aus einem Hochgeschwindigkeits-Bus ASB (Advanced System Bus) bzw. AHB (Advanced High-performance Bus) in neueren Systemen und einem Peripherie-Bus (APB), die über eine Bridge miteinander verbunden sind. In der dritten Generation der ARM-SoCs gibt es sogar vier Busse: Neben AHB und APB kommen noch Advanced eXtensible Interface (AXI) und der Advanced Trace Bus (ATB) hinzu.

Im RP2040 des Pi Pico kommen AHB-Lite und APB zum Einsatz. In Bild 5 sieht man, wie fragmentiert die innere Struktur des SoC ist. Diverse SRAMs in diversen Größen hängen am AHB und am APB sind die I/Os angeschlossen.

Eine typische Memory Map für ARM-Systeme, also der Speicherplan, sieht dann ähnlich aus wie in Bild 6 für einen Prozessor vom Type ARM Cortex M3. Im Adress-Bereich des SRAM und der Peripherie ist ein kleiner Bereich für Bit Banding reserviert. Dort sind einzelne Bits, etwa I/O-Register der Peripherie, gegen eine 32-Bit-Adresse gemappt. So lassen sich einzelne Bits ohne weitere Operationen lesen und setzen. Beim ATmega etwa muss man immer das ganze I/O-Register und dessen bereits gesetzte Werte berücksichtigen. Um ein Bit zu verändern, muss man dann das ganze Register lesen, ein Bit setzen (OR) und alles zurückschreiben. Wie man das Feature nutzt, erklärt der Kasten "Bit-Banding-Berechnung".



Unübersichtlich

Beim ESP32 ist die Memory Map ziemlich lückenhaft, wie in Bild 7 zu sehen ist. Die grauen Bereiche sind die nicht genutzten, für spätere Zwecke reservierten Bereiche, die weißen sind die benutzten. Eine Memory Management Unit (MMU) führt zumindest die externen Speicherbereiche für die beiden CPU-Kerne zusammen. Nebenbei benutzt der ESP32 (und alle anderen Modelle) getrennte Busse für Instructions und Daten, die aber in einem linearen Adressraum zusammengeführt sind.

Spannend wird es, wenn der ESP32 Teile des Codes, zum Beispiel einzelne Funktionen, aus dem Flash in seinen internen SRAM kopiert. Denn dann bezieht er seine Instruktionen ja theoretisch über den Datenbus. Espressif hat dem ganzen die Krone aufgesetzt und hat Teile des SRAMs (und auch des Flash) sowohl über den Datenbus als auch über den Instruction-Bus (I-bus) angebunden und das ganze jeweils in unterschiedliche Adressregionen gemappt. Um über den I-bus Befehle aus dem SRAM1 (siehe Bild 8) auszulesen, legt die CPU die Adresse 0x400A0000 auf den Bus. Will sie hingegen Daten aus der gleichen Speicherstelle auslesen, legt sie 0x3FFFFFFFF auf den Bus.

Glücklicherweise muss man sich nur in Extremfällen mit der genauen Adressierung beschäftigen. Um Funktionen gewollt in den schnelleren SRAM zu verlagern, insbesondere für Interrupt-Routinen wichtig, stellt man ihr IRAM_ATTR voran. Der Linker sorgt dann später dafür, dass alles an den richtigen Adressen liegt.

Auch die Flashing-Tools wie esptool.py und idf.py wissen, wohin neue Programme gehören. In der Anzeige beim Flashen tauchen die in Bild 7 gezeigten Adressen nie auf. Aber der Offset zur Start-Adresse des Flash wird angezeigt: 0x00010000.

Ausblick

Einfach war früher, immerhin verbergen die Hersteller die Komplexität ihrer Mikrocontroller geschickt vor den Augen des Anwenders. Am Ende ist wichtig, in linearen Adressräumen denken und entwickeln zu können. Damit ist die Grundlage für die Adressierung aller Komponenten gelegt. In einer der nächsten Ausgaben erklären wir dann, wie Software diese wofür nutzt. — dab





Ausprobiert — von Make:—

Xinfrared T2S Plus

Smartphone-Wärmebildkamera mit Makro-Funktion



Die Xinfrared T2S Plus ist eine etwa 3 × 3 × 3 Zentimeter große Wärmebildkamera fürs Smartphone. 256 × 192 Pixel bieten bei 25 fps einen guten Anhaltspunkt, um etwa einem Kurzschluss auf die Schliche zu kommen.

Für die App Xtherm ist kein Account nötig. Neben Aufzeichnung von Bildern und Videos, lässt sich die höchste und niedrigste Temperatur an bis zu drei wählbaren Punkten im Bild auslesen. Auch Durchschnittstemperaturen auf einer Linie oder in einem Rechteck sind möglich. Der Temperaturbereich lässt sich zwischen -20 bis 120 °C und 120 bis 450 °C umschalten.

Für genaue Messungen lässt sich die Umgebungstemperatur, die Luftfeuchtigkeit, der Betrachtungsabstand und ein Korrekturwert einstellen. Auch Temperaturalarme können eingestellt werden.

Die kleine Kamera kommt mit einer Schutzhülle, einem Objektivdeckel sowie einem Handgriff, an den sich Smartphone und Kamera montieren lassen. Es steht eine Version für Android (USB-C) und eine für iOS-Geräte (Lightning) zur Wahl. An Apple-Geräten mit USB-C muss ein Adapterkabel verwendet werden. Die Bildqualität ist gut, die Darstellung flüssig. Für Nahaufnahmen lässt sich das Objektiv an einem Drehring auf einen Betrachtungsabstand von rund zwei Zentimetern einstellen —jom

Hersteller	•
URL	
Preis	

Xinfrared amzn.eu/d/dYSGhCN 339 € (Amazon)

Califlower

3D-Drucker-Kalibrierungssystem

Califlower besteht aus einem Testmodell, Excel/ OpenOffice-Tabellen zur Auswertung und einer ausführlichen englischen Dokumentation. Es ist nicht gedacht für die erste Kalibration des Druckers, kann aber die Maßhaltigkeit des Druckers nochmals deutlich verbessern. Ein genauer Messschieber und 24 g Filament werden benötigt. Inklusive Druckzeit benötigt der Prozess etwa zwei Stunden. Man sollte mit den meistgenutzten Filamententypen drucken, um die verschiedenen Schrumpfungsraten zu ermitteln. Nach dem Druck müssen zehn Messungen entsprechend der Tabelle gemacht und eingetragen werden. Dabei werden Außenund Innenkanten im Wechsel gemessen. Das Califlower-Testobjekt ist so konstruiert, dass sichere und genaue Messungen leicht möglich sind. Die Messwerte werden dann in die Tabelle eingetragen. Dabei ändert sich die Farbe der Tabellenzelle je nach Güte der eingegebenen Werte.

Die Ergebnisse zeigen an, welche Fehlergrößen existieren (X, Y, Skew, also Achsen nicht 90° zueinander, Inner und Outer) und wo die Stellschrauben (Steps per Millimeter, Mecha-

Vector 3D Califlower

Ausprobiert — von Make:—

nik, Über-/Unterextrusion) im System sitzen. Mit den Korrekturwerten kann man dann im Slicer, der Druckerfirmware oder bereits beim Modellierungsprozess in CAD gegensteuern. Je nach System kann dies leicht (Open Source, RepRap oder ähnlich) oder schwierig (spezielle Herstellerfirmware) sein. Das Dokument zu Califlower und auch die Tabellen geben aber die wichtigsten Daten und Vorgehensweisen wieder, der Support findet vorzugsweise über Discord statt. —*caw*

HerstellerVector 3DURLvector3d.shopPreis6,95 €

KiCad 8

Open-Source-PCB-Design-Software

Das Team von KiCad hat Version 8 seiner Open-Source-PCB-Software veröffentlicht. Seit KiCad 7 ist ein jährliches großes Update geplant. Jede Major-Version wird dann ein Jahr gepflegt. Neue Versionen lesen natürlich die alten Projekte, der umgekehrte Weg ist aber nicht möglich, was bei Übernahme von Projekten immer bedacht werden sollte. Neue Austauschformate sind u. a. EasyEDA (JLCEDA), CADSTAR, Solidworks, Altium Designer, EAGLE und LTSpice schematics. Die Offenheit zeigt sich auch in der Unterstützung von Cadence Allegro als Schaltplan-Export; hier bewegt man sich auf die Industrie zu.

Zu den weiteren Neuerungen zählen über 1500 ergänzte Symbole und 760 Footprints, generelle Verbesserungen der Benutzerführung. Weiterhin gibt es unzählige Verbesserungen in den Komponenten wie dem Schaltplan-Editor, PCB-Design und dem Simulator. Für viel Gegenwind sorgte in vorherigen Versionen die Darstellung von "Pin 1" in den Footprints, hier folgt man nun der Ästhetik und nicht veralteten Industriestandards.

Am Simulator hat das KiCad-Team mit den Ngspice-Entwicklern geschraubt und dabei nicht nur die Handhabung vereinfacht, sondern bietet inzwischen auch Pol-Nullstellen, Rau-



Ausprobiert

schen, Streuparameter und Fast Fourier Transformation (FFT). Die Ergebnisse von Arbeitspunkt-Simulationen kann KiCad 8 direkt im Schaltplaneditor anzeigen. Hier in der Redaktion konnten wir uns anhand eines aktuellen Projektes von den Neuerungen überzeugen.

Die KiCad-Entwickler haben auch das Command-Line-Interface (CLI), den Footprint-Editor, den 3D-Viewer und viele weitere Komponenten angepasst und das Programm kann jetzt auch Teilelisten (BOM) ohne externe Tools exportieren. —*akf*

make-magazin.de/xz2m

Hersteller	KiCad
URL	KiCad.org
Preis	kostenlos (GPL3)

Cardputer

Mikrocomputer mit ESP32, Tastatur, Akku und Display

In etwa so groß wie eine Zigarettenschachtel ist der Cardputer von M5Stack. In dem soliden Kunststoffgehäuse mit den Maßen 84 × 54 × 17 mm steckt aber eine Menge drin: Da ist zunächst der von außen zugängliche Mikrocontroller ESP32-S3 mit LX7-Dual-Core-Prozessor und 8 MB Flash, den M5Stack auch schon im CoreS3-Modul einsetzt. Der Controller sitzt in einer Steckfassung, ist aber zusätzlich mit einem nicht lösbaren Folienkabel mit dem Cardputer verbunden. Am auffälligsten ist aber die 56 Mini-Tasten umfassende alphanumerische Tastatur. Darüber thront das 1,44-Zoll-TFT-Display mit 240 × 135 Pixeln.

Außerdem ist der kleine Computer mit einem Mikrofon, einem digitalen Verstärker und einem 1-W-Lautsprecher sowie einem Infrarot-Sender ausgestattet. Für die Stromversorgung sind zwei Akkus zuständig: Ein kleinerer mit 120 mAh steckt im Oberteil auf der Tastatur-Platine, der größere mit 1400 mAh sitzt im nur mit zwei kleinen Inbus-Schrauben (Schlüssel wird mitgeliefert) befestigten Unterteil. Der Stromverbrauch soll laut Hersteller im Standby bei 0,26 Mikroampere liegen und bei Verwendung des IR-Senders bis zu 255 mA betragen. Ein mehrstündiger, ununterbrochener Akkubetrieb ist also möglich.

An der linken Seite ist eine I²C-Schnittstelle herausgeführt, die externe Sensoren mit Spannung versorgen oder aber per Schalter als Spannungseingang für den Cardputer umgeschaltet werden kann. Die Rückseite zieren ein Ein-/Ausschalter, die Taster für Reset und Firmware-Download sowie die winzige IR-LED. Mit ihr lassen sich alle gängigen Unterhaltungselektronik-Komponenten steuern.

Im Gehäuseunterteil sind zwei starke Magnete untergebracht, sodass der kleine Computer an Metallgehäusen oder magnetischen Whiteboards sicher haftet. Außerdem sitzen dort zum Lego-System kompatible Stecklöcher.

M5Stack bietet zum Cardputer eine umfangreiche Online-Dokumentation, Schaltpläne sowie auf Github den Quellcode der bereits auf dem Gerät installierten Software, die u. a.



auch die Programmierung des IR-Senders zeigt. Programmiert wird das Gerätchen mit der Arduino IDE oder der ESP-IDF. Der Cardputer kostet im Shop des Herstellers 30 US-Dollar. In Europa ist er im Online-Handel ab knapp 40 Euro erhältlich. —hgb

make-magazin.de/xz2m

Hersteller	M5Stack
URL	make-magazin.de/xz2m
Preis	29,90 US-\$/ca. 40 €

Funktionale Programmierung verstehen

Konzepte und Entwurfsmuster für guten Code

Funktionale Programmierung verspricht eleganteren Code als der imperative Ansatz und erspart den Overhead der Objektorientierung. Und weil dann noch "Python-Special" auf dem Titel steht, interessiert uns dieses Buch natürlich.

Rundheraus: Um von diesem Buch zu profitieren, muss man sowohl praktische Erfahrung mit (objektorientierter) Software-Entwicklung als auch fundierte Kenntnisse der Konzepte dahinter haben und schon wissen, um was es sich etwa bei Klassen, Instanzen oder Singletons handelt. Zudem gibt es viele Code-Schnipsel in unterschiedlichen Sprachen, darunter in Java, Python (klar, steht ja auch drauf), aber vor allem in Scala. Für alle, die mit letzterem noch nicht in Berührung kamen, gibt es im Anhang auf 20 Seiten einen denkbar kompakten Überblick.

In Kombination mit einschlägigen Vorkenntnissen reicht dieser Anhang aus, dass man der Einführung in Entwurfsmuster, funktionale Datenstrukturen, in das zentrale Konzept der Unveränderlichkeit sowie in Funktionen höherer Ordnung folgen kann, die den rund 120-seitigen Hauptteil des Buchs ausmachen. Dabei fällt auf: Selbst wenn es mit einem Ausflug in die Kategorien-Theorie phasenweise sehr mathematisch und grundsätzlich wird, hat der Autor doch stets vor Augen, dass seine Leser das Gelernte praktisch einsetzen wollen. Das erklärt auch, warum er bei seinen Beispielen auf die Sprache Scala statt auf Haskell setzt – Haskell ist zwar viel strenger funktional, aber Scala unterstützt parallel auch das objektorientierte Programmieren und ist daher praktischer nutzbar.

Für noch mehr Nutzwert zeigt ein zweiter Anhang auf den restlichen 30 Seiten, wie die Scala-Beispiele des Buchs in Python aussehen. Diesen Teil gibt es nur in der deutschen Ausgabe des Buches, nicht im US-Original. Das ist eine Bereicherung, führt in der Praxis allerdings dazu, dass man ziemlich oft dabei ist, hin- und herzublättern. Fazit: Ein Buch für Spezialisten, die ihren Horizont erweitern wollen, für normale Maker aber keine Pflichtlektüre. — pek



Hinweis: Dieses Buch erscheint in einem Verlag, der wie die Maker Media GmbH zu heise medien gehört.

Autor	Jack Widman, Python-Teil von Thorsten T. Will
Verlag	Rheinwerk Computing
Umfang	187 Seiten
ISBN	978-3-8362-9868-1
Preis	29,90 \in (Buch oder E-Book), 34,90 \in (Bundle)

Synth Unit

MIDI-Synthesizer-Modul für Mikrocontroller



Wirklich winzig, gerade mal 10 g schwer, recht günstig – und trotzdem steckt ein kompletter MIDI-Synthesizer in diesem kleinen Gehäuse. Er basiert auf einem MIDI-SAM2695-Chip. Er ist 64-stimmig polyphon, und die Stimmen können auch auf mehrere Instrumente (pro MIDI-Kanal, multitimbral) aufgeteilt werden. Die eingebauten Effekte wie Reverb, Chorus, Mixer, EQ und weiteren, reduzieren die Polyphonie auf 38 Stimmen. Zudem bringt er einen GM-Satz an insgesamt 128 Instrumentenklängen mit.

Da die MIDI-Datenübertragung seriell erfolgt, kann das Modul mit vielen Mikrocontrollern zusammenarbeiten, beispielsweise ESP32, ESP 8266 und Arduinos. Die Sound-Wiedergabe erfolgt über einen Verstärker und Lautsprecher im Modul. Intern bietet der Chip Stereoausgänge, die aber nicht herausgeführt sind.

Programmiert wird der Synthesizer entweder in der Arduino IDE oder mittels UIFlow. Mit gut 40 Zeilen Code lässt man Drums oder Pianomelodien ertönen. Der Hersteller stellt online eine recht ausführliche Dokumentation inklusive Schaltplan, Datenblatt des Chips sowie Programmierbeispielen bereit. -hgb

Hersteller	M5Stack
URL	make-magazin.de/xz2m
Preis	13 US-\$

PaHUB2

1:6-Hub für den I²C-Bus

Der I²C-Bus hat den Vorteil, dass man mehrere Sensoren, Displays oder andere Komponenten über nur zwei Datenleitungen mit einem Mikrocomputer oder -controller verbinden kann. Das spart eine Menge an Kabel und/oder Leiterbahnen. Voraussetzung ist aber, dass die Komponenten unterschiedliche Bus-Adressen besitzen. Bei vielen kleinen Erweiterungsplatinen für Mikrocontroller sind die aber fest eingestellt, sodass beispielsweise alle Displays gleichen Typs auch dieselbe Adresse verwenden. Folge: Entweder darf nur eines dieser Displays verwendet werden oder man braucht für jedes einen eigenen I²C-Bus, womit der Vorteil dahin wäre.

Das Modul PAHub2 von M5Stack umgeht dieses Problem. Der darin arbeitende Chip vom Typ PCA9548APW kommuniziert auf der einen Seite mit einem Mikrocontroller über den I²C-Bus. Auf der anderen Seite steuert er sechs weitere, voneinander unabhängige I²C-Anschlüsse auf der kleinen Hub-Platine, an die beliebige Geräte auch mit gleichen Adressen angeschlossen werden dürfen, ohne sich gegenseitig zu beeinträchtigen.

Die Stromversorgung des Hubs und der daran angeschlossenen Module erfolgt über



den Bus vom Mikrocontroller aus. Der Hub selbst hört auf die hexadezimale Adresse 70. Die kann aber mithilfe dreier Lötbrücken auf 71 bis 77 geändert werden, sodass auch der Betrieb mehrerer Hubs am selben Bus möglich ist. So wären maximal bis zu 48 l²C-Geräte gleicher Adresse denkbar.

Die am Hub angeschlossenen Geräte müssen vom Programm her dann mit der Adresse des Hubs und einer zusätzlichen Kanalnummer von 0 bis 5 adressiert werden, die der Portnummer auf der Hub-Platine entspricht. Auf Github stellt der Hersteller dazu ein kleines Programmbeispiel zur Verfügung, das die Vorgehensweise erklärt. Der Hub (Online-Preis ca. 10 Euro) wird mit einem vierpoligen Anschlusskabel in einem Lego-kompatiblen Gehäuse —hgb geliefert.

Hersteller	M5Stack
URL	make-magazin.de/xz2m
Preis	7,95 US-\$/9,50 €

ePulse Feather ESP32-C6

Zigbee, WiFi, BLE, Thread und Matter

Von Thingpulse kommt das ePulse Feather C6 Dev-Board, das besonders für akkubetriebene IoT-Anwendungen interessant ist. Der ESP32-C6 von Espressif kommuniziert mit Wifi 6, BLE 5, Zigbee und unterstützt Thread und Matter. Wifi 6 ist interessant, weil es mittels "Target Wake Time" Energie spart. Zigbee ist ebenfalls ein spannendes Thema für IoT- und Heimautomation.

Das ESP-Modul (4 MB Flash, C6-MINI-1, ESP32-C6H4) sitzt auf einem Entwicklungsboard nach dem Adafruit-Feather-Standard und lässt auf üblichen Breadboards seitlich genügend Platz für Jumper-Kabel, um auf die 20 GPIO-Pins zugreifen zu können.

Für den Betrieb mit Lithium-Akkus gibt es einen JST-Steckverbinder und einen TP4065-Chip, der für das Laden und Überwachen des Akkus zuständig ist. Ein MAX17048, Fuel Gauge Chip" überwacht die Akkuspannung und liefert zuverlässige Daten über die verbleibende Energie im Akku.

Das Board kann über USB-C oder den VIN-Pin (3,3 V bis 6 V) mit Strom versorgt werden. Im Tief-



schlaf verbraucht das ePulse-Board nur 18 µA. Die Entwicklung für das Board ist natürlich mit der Espressif-IDF in den aktuellen Versionen und damit auf allen IDEs möglich, die diese unterstützen (z. B. PlatformIO). Die ArduinoIDE unterstützt das C6-Modul erst ab der Entwicklerversion 3.0.0-alpha3. ThingPulse hat bereits begonnen, Demo-Code zur Verfügung zu stellen. — caw

Hersteller	ThingPulse Ltd.
URL	Thingpulse.com
Preis	15 US-\$

M5Stack

Ausprobiert

von Make:

Alpakka

Game-Controller mit Gyromaus

Vier Knöpfe rechts, vier Richtungstasten links, vier Schultertasten und zwei Analogsticks. Was bin ich? Ein moderner Game-Controller.

Controller für Videospiele haben mittlerweile ein Standard-Layout, das firmenübergreifend wenig Unterschiede bietet. Der Open-Source-Controller Alpakka der finnischen Firma Input Labs hat aber ein paar Tricks auf Lager. Nicht nur wird der rechte Analogstick von einer Gyromaus ersetzt, anstelle dieses Sticks befindet sich ein Mausrad auf dem Controller. Daneben ist noch ein kleiner C-Stick untergebracht, der in vier Richtungen gedrückt werden kann.

Der Controller basiert auf einem Raspberry Pi Pico, der (zusammen mit Tastern, Potentiometer-Joystick und Dreh-Encoder) auf eine speziell designte Leiterplatte aufgelötet wird. Aufgrund des Open-Source-Charakters des Produktes kann man entweder alle Teile selber kaufen, sich eine Leiterplatte bei einem Leiterplattenhersteller anfertigen lassen und das Gehäuse selber 3D-drucken oder direkt im Input-Labs-Shop die Einzelteile ordern.

Mit einem Micro-USB-Kabel verbindet man den Controller über die USB-Schnittstelle des Raspberry Pi Picos mit dem Computer. Der Controller wird ohne die Installation von Treibern erkannt. Je nachdem, wie das Gamepad eingestellt ist, werden die Eingaben entweder als generische "Tastatureingaben" oder als Xbox-Controller-Eingaben eingestuft. Die Gyromaus wird immer als Maus erkannt.

Die Maus ist nicht immer aktiv. Erst wenn man den Daumen auf ein Stück leitendes Plastik legt, in dem die Facebuttons (die vier Knöpfe an der rechten Vorderseite des Controllers) eingelassen sind, wird die Maus aktiviert.

Der Alpakka ist nicht nur treiberlos nutzbar, sondern auch ohne eine Konfigurationssoftware bedienbar. Auf dem Controller selber sind verschiedene Profile gespeichert, die die Tasten des Controllers unterschiedlich belegen. Zum Beispiel im Profil "FPS WASD" (First Person Shooter) sind WASD (die Bewegungstasten) auf den Analogstick eingestellt, die linke hintere Schultertaste wird mit Rechtsklick und die rechte mit Linksklick genutzt.

Ein anderes Profil (Racing) legt die Beschleunigungsachsen, die ein Xbox-Controller an den hinteren Schultertasten hätte, auf den Analogstick. Diese Profile kann man mit unterschiedlichen Tastenkombinationen direkt am Controller einstellen. Wem das aber zu fummelig ist, der kann auch direkt im Browser ein Tool zum Wechseln der Profile und Einstellungen des Alpakkas benutzen.

Der Controller liegt trotz seines futuristisch eckigen Polygon-Designs sehr bequem in der Hand. Die Tasten auf der Vorderseite haben einen angenehmen Druckpunkt und einen guten Hub.

Bei den Schultertasten sieht das leider etwas anders aus. Die vorderen beiden Schultertasten fühlen sich, so wie die Facebuttons, sehr angenehm an und haben einen guten Hub. Die hinteren Tasten fühlen sich allerdings gar nicht an. Ohne den Klick der Knöpfe würde man nicht bemerken, dass die Buttons gedrückt werden. Auch sehen kann man das kaum. Der Hub ist wirklich sehr minimal. Das ist für komplett digitale Knöpfe okay, aber ein kleines bisschen mehr Weg bei den Tasten würde sich besser anfühlen.

Der Analogstick ist eine Überraschung. Obwohl er über keine Gummierung verfügt (die man natürlich in Eigenregie nachrüsten





kann), schmiegt er sich gut an den Daumen an. Da er ein bisschen länger als andere Analogsticks ist, sind präzisiere Bewegungen möglich. Das Scrollrad sieht aus wie ein Mausrad, fühlt sich an wie ein Mausrad und funktioniert genau so wie ein Mausrad.

Die Gyromaus funktioniert sehr gut. Die Mausempfindlichkeit lässt sich, genau wie die gesamte Controller-Belegung, direkt in drei Stufen am Controller einstellen. Die vordefinierten Sensitivitätsprofile orientieren sich an der Auflösung des Monitors, an dem der Alpakka genutzt wird: 1080p, 1440p und 4k. Wem die Geschwindigkeit in keinem der Profile passt, kann diese in der Webapp manuell einstellen.

Beim Spielen funktioniert das Zusammenspiel zwischen Controller und Maus sehr gut. Spiele wie Civilisation, die ausschließlich mit der Maus gesteuert werden können, werden zum entspannten Couch-Spaß. Aber auch Third-Person-Shooter wie Remnant 2 funktionieren super und man kann sehr intuitiv zielen. Auch wenn es im Spiel hektischer wird, hat man mit der Gyromaus gute Kontrolle. Zwar braucht man natürlich ein bisschen Eingewöhnungszeit, aber danach ist der Alpakka eine tolle Alternative zu den klassischen Gamepads. —das

Hersteller	Input Labs
URL	inputlabs.io/alpakka
Preis	160€

IMPRESSUM

Redaktion

Make: Magazin Postfach 61 04 07, 30604 Hannover Karl-Wiechert-Allee 10, 30625 Hannover Telefon: 05 11/53 52-300 Telefax: 05 11/53 52-417 Internet: www.make-magazin.de

Leserbriefe und Fragen zum Heft: info@make-magazin.de

Die E-Mail-Adressen der Redakteure haben die Form xx@make-magazin.de oder xxx@make-magazin.de. Setzen Sie statt "xx" oder "xxx" bitte das Redakteurs-Kürzel ein. Die Kürzel finden Sie am Ende der Artikel und hier im Impressum.

Chefredakteur: Daniel Bachfeld (dab) (verantwortlich für den Textteil)

Stellv. Chefredakteur: Peter König (pek)

Redaktion: Heinz Behling (hgb), Johannes Börnsen (jom), Ákos Fodor (akf), Daniel Schwabe (das), Dunia Selman (dus, Social Media), Carsten Wartmann (caw)

Mitarbeiter dieser Ausgabe: Beetlebum (Comic), Manfred Caspar, Detlef Grell, Erik Golz, Robert Kränzlein, Rainer Maria Kreten, Dominik Kuhn, Pina Merkert, Alexander Neubauer, Florian Schäffer, Olaf Splitt, Ralf Stoffels, Armin Zink

Assistenz: Susanne Cölle (suc), Martin Triadan (mat)

Leiterin Produktion: Tine "The Rock" Kreye

- DTP-Produktion: Martina Bruns, Martin Kreft (Korrektorat)
- Art Direction: Martina Bruns (Junior Art Director)

Layout-Konzept: Martina Bruns

Layout: Nicole Wesche

Fotografie und Titelbild: Andreas Wodrich

Digitale Produktion: Kevin Harte, Thomas Kaltschmidt, Pascal Wissner

Hergestellt und produziert mit Xpublisher: www.xpublisher.com

Verlag

Maker Media GmbH Postfach 61 04 07, 30604 Hannover Karl-Wiechert-Allee 10, 30625 Hannover Telefon: 05 11/53 52-0 Telefax: 05 11/53 52-129 Internet: www.make-maqazin.de

Herausgeber: Christian Heise, Ansgar Heise

Geschäftsführung: Ansgar Heise, Beate Gerold

Anzeigenleitung: Daniel Rohlfing (-844) (verantwortlich für den Anzeigenteil), mediadaten.heise.de/produkte/print/ das-magazin-fuer-innovation

Leiter Vertrieb und Marketing: André Lux (-299)

Service Sonderdrucke: Julia Conrades (-156)

Druck: Dierichs Druck + Media GmbH & Co.KG, Frankfurter Str. 168, 34121 Kassel

Vertrieb Einzelverkauf:

DMV DER MEDIENVERTRIEB GmbH & Co. KG Meßberg 1 20086 Hamburg Telefon: +49 (0)40 3019 1800 Telefax: +49 (0)40 3019 1815 E-Mail: info@dermedienvertrieb.de Internet: dermedienvertrieb.de

Einzelpreis: 13,50 €; Österreich 14,90 €; Schweiz 26.50 CHF; Benelux 15,90 €

Abonnement-Preise: Das Jahresabo (7 Ausgaben) kostet inkl. Versandkosten: Inland 80,50 €; Österreich 88,90 €; Schweiz 123.90 CHF; Europa 95,20 €; restl. Ausland 100,80 €

Das Make-Plus-Abonnement (inkl. Zugriff auf die App, Heise Magazine sowie das Make-Artikel-Archiv) kostet pro Jahr $6,30 \in$ Aufpreis.



Abo-Service:

Bestellungen, Adressänderungen, Lieferprobleme usw.: Maker Media GmbH Leserservice Postfach 24 69 49014 Osnabrück E-Mail: leserservice@make-magazin.de Telefon: 0541/80009-125 Telefax: 0541/80009-122

Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden. Kein Teil dieser Publikation darf ohne ausdrückliche schriftliche Genehmigung des Verlags in irgendeiner Form reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Alle beschriebenen Projekte sind ausschließlich für den privaten, nicht kommerziellen Gebrauch. Maker Media GmbH behält sich alle Nutzungsrechte vor, sofern keine andere Lizzenz 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 2024 and published under license from Make Community LLC. All rights reserved.

Printed in Germany. Alle Rechte vorbehalten. Gedruckt auf Recyclingpapier.

© Copyright 2024 by Maker Media GmbH

ISSN 2364-2548

Nachgefragt

Welches Brettspiel würdest du am liebsten mal mit Licht- oder Soundeffekten ausstatten?

Manfred Caspar Völklingen, überwacht auf S. 88 Nistkästen.

Ich brauche jetzt endlich mal einen vernünftigen Spielekegel für mein "Mensch ärgere dich nicht", der blinkt, jammert und heult, wenn er vom Spielfeld gekickt wird.

Rainer Maria Kreten

S. 90 ein autarkes LoRaWAN-Gateway. Europareise: Flaggenfarben, knusperndes Baguette, urige Alphörner, mähende Schafherden, exotische Musikinstrumente, so bekommt jede Region ihre Licht- und Soundmarke.



Hannover, programmiert auf S. 72 ATtinys.

Soundeffekte für die unter den Bäumen von Sagaland versteckten Märchen – auch hilfreich bei Sehschwäche – und die jeweiligen Hörbücher zum Abspielen. Die Bäumchen sollten im Dunklen leuchten. Dominik Kuhn Püttlingen, überwacht

auf S. 88 Nistkästen. Beim Rauswerfen bei

"Mensch ärgere Dich nicht" werden zufallsgenerierte passende Buzzertöne abgespielt, um die Stimmung etwas anzuheizen. Die LED der Figur geht aus und das Häuschen fängt wild an zu blinken.

Inserentenverzeichnis

7 TUXEDO Computers GmbH, Augsburg

Ein Teil dieser Ausgabe enthält Beilagen der DIMABAY GmbH, München.

..... 132



DEUTSCHLANDS GEFÄHRLICHSTES BO-ANGEBOT*

*VON DEUTSCHLANDS GEFÄHRLICHSTEM DIY-MAGAZIN (LAUT LESERN)

2× Make testen mit über 30 % Rabatt

Jetzt bestellen: make-magazin.de/abo-angebot

Warum eigentlich gefährlich?

Laut Lesern sind wir das "gefährlichste DIY-Magazin" Deutschlands. Das ist aber natürlich nur Spaß! Sie können unser Magazin ganz unbesorgt lesen und 2 Ausgaben als Heft + digital testen, zusätzlich erhalten Sie ein Geschenk Ihrer Wahl – klingt doch eigentlich ganz ungefährlich.







Warum TUXEDO? Hardware. Software. Service.



Unsere Geräte kommen vorinstalliert mit TUXEDO OS und sind perfekt auf den Einsatz mit Linux optimiert.



Unsere Softwarelösungen stehen allen Linux-Nutzern zur Verfügung und werden kontinuierlich weiterentwickelt.



Der technische Kundenservice von TUXEDO ist stets für Fragen, Wünsche und Hilfestellung erreichbar.











© Copyright by Maker Media GmbH.





Deutscher Tech Support