DEUTSCHLANDS GEFÄHRLICHSTES DIY-MAGAZIN

Iake

Der neue Pico 2 im Schnelltest

Projekte mit ATtiny

Schimpfwortgenerator
 433-MHz-Fernsteuerung

d E

Jake:

Funk im Smart Home

Techniken im Überblick
So steht's um Matter
Ratgeber: Was eignet sich wofür

Siebdruck mit Laser & Leim

Rahmen günstig selbst bauen
Holzleim statt Foto-Emulsion
Motiv freilasern und drucken

Reports

Rückblick auf Apple II
 Maker-Paradies in der Wüste

IoT-Projekte Tibber-Relais Wasserstandsmelder

() 10:10



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.





Alles nur geklaut?

Programmiersprachen waren noch nie mein Steckenpferd, aber ohne sie sind manche Projekte schlicht unmöglich. Also habe ich mir entsprechende Bücher gekauft, zehn Stunden lange YouTube-Tutorials geschaut und wer weiß wie oft LEDs mit Arduinos zum Blinken gebracht. Trotzdem bin ich nie so weit gekommen, Code für einen Arduino ohne ständiges Nachschlagen einfach runterschreiben zu können oder mir wenigstens die unterschiedlichen Variablenarten zu merken.

Doch meine Rettung naht! Seit einer Weile lässt sich viel schönerer Code, als ich ihn jemals werde schreiben können, kinderleicht per KI generieren. ChatGPT macht da im 4o-Modell einen passablen Job. Endlich kann ich die Absauganlagensteuerung bauen, die mir schon so lange im Kopf umhergeistert. Arduino, Stromsensoren und ein paar Relais sind schnell zusammengesucht. Soll doch das Kunsthirn seinen Job machen, ich lehne mich zurück.

Wenig später habe ich die Steuerung noch um ein Display erweitern lassen, das den Status der einzelnen Absperrschieber grafisch darstellt, und ein Nachlauf für den Staubsauger ist auch eingebaut. Rund 25-mal habe ich die KI den Code anpassen lassen, immer wieder im Online-Simulator Wokwi ausprobiert, stückweise erweitert und neue Funktionen eingebaut. Zurückgelehnt habe ich mich also gar nicht wirklich. Wenn ich Teile des Codes nicht verstanden habe, erklärte die KI ihn mir geduldig. Wie praktisch, geht mir durch den Kopf: Endlich habe ich jemanden, dem ich völlig ungeniert auch die dümmsten Fragen stellen kann! Ganz nebenbei habe ich so bei diesem Projekt wohl mehr übers Programmieren gelernt als je zuvor. Ob der fertige Code schön und effizient ist, kann ich zwar nicht beurteilen, aber er funktioniert. Das hätte ich so niemals selbst hinbekommen!

Doch während der Code nebst auskommentierten Erklärungen auf dem Bildschirm erscheint, kommen mir Zweifel. Ich fühle mich, als wäre ich wieder 15 Jahre alt und würde in der Bioklausur vom Tischnachbarn abschreiben. Wo bleibt denn das Selbstgemachte, wenn der Code von einem Computer erstellt wird? Ist der generierte Code nicht ein Innovationskiller, gar geklaut? Und ist das denn wirklich noch MEIN Projekt, wenn die KI den Code herbeizaubert?

Beim Zeichnen der Platine wird mir klar: Natürlich ist das mein Projekt. Die Idee für diese Steuerung ist von mir, die Hardware habe ich ausgesucht und in ein selbst gebautes Gehäuse gesetzt, die Platine layoute ich und die pneumatisch bewegten Absperrschieber sind ebenfalls selbst konstruiert und gebaut. Diese Steuerung gibt es so kein zweites Mal. Und auch den generierten Code gibt es so nicht noch einmal, dafür haben die KI und ich zu viele Details besprochen und angepasst.

Ich finde es nicht verwerflich, sich von einer KI bei Teilen eines Projektes helfen zu lassen, die man selber nicht kann oder die einem keinen Spaß machen. Schließlich geht es bei unseren Hobbyprojekten nicht darum, den effizientesten Code der Welt zu schreiben, sondern darum, unsere Ideen zum Leben zu erwecken. Es ist nur konsequent, wenn wir die Aufgaben, die die KI besser und schneller erledigen kann, auch von ihr erledigen lassen.

In diesem Sinne: Macht das Beste aus der modernen Technik und lasst eurer Kreativität freien Lauf. Die Welt des DIY bleibt spannend und voller Möglichkeiten – und mit der Unterstützung von KI vielleicht sogar auch innovativer.

Happy Making!

Johannes Kornsen

Johannes Börnsen

make-magazin.de/xn91



Schimpfen und fernsteuern mit ATtiny

Warum zu viel Rechenkraft einsetzen, wo sie nicht nötig ist? Für manche Projekte reicht der winzige ATtiny gut aus. Er ist preiswert, robust und bei Arduino-Fans sehr beliebt. In diesem Heft widmen wir ihm gleich zwei Artikel. Im ersten geht es um einen handlichen Schimpfwortgenerator und im zweiten darum, wie der ATtiny zusammen mit einem 433-MHz-Funkmodul etwa zur Steuerung eines Modellbootes eingesetzt werden kann.

8 Der Schimp

433-MHz-Fernsteuerung im Eigenbau

Inhalt

"Der beste PC aller Zeiten"

Einer der langlebigsten und einflussreichsten Computer aller Zeiten ist der Apple II. Die Zeitschrift PC World kürte ihn sogar zum "besten PC aller Zeiten". Nächstes Jahr im Juni wird er 50 Jahre alt. Wir begeben uns auf eine Nostalgiereise und schauen uns die damals bahnbrechende Hardware an. Außerdem erzählt die Spieleentwickler-Legende John Romero in einem Interview, was der Apple II für ihn bedeutet hat.

98 Apple II – Die Nostalgie-Zeitreise103 Zeitzeugen-Interview mit John Romero



- 3 Editorial
- 6 Leserforum
- 8 Der Schimpfroboter
- 14 Verkaufsautomat mit ESP
- 24 433-MHz-Fernsteuerung im Eigenbau
- 34 KI erobert die Maker Faire
- 38 Oxocard Connect
- 42 Unser Team
- 44 AVR-Programme debuggen, Teil 2
- 52 Wassertiefe messen und digital abfragen
- **58** Automatisiert Strom sparen
- 64 Ein Maker-Paradies in der Wüste
- 76 ESP32-Platine für Schrittmotoren
- 78 Interaktives schwebendes Display
- 80 Elektromechanischer Handheld
- 82 Siebdruck mit Leim und Laser

Das neue Kraftpaket Pico 2 im Test

Für rechenintensive Projekte könnte der neue Pico 2 der Raspberry Foundation interessant sein. Bereits zum Börsengang im Juni wurde der vorgestellt – und passenderweise auch schon auf der Maker Faire in Hannover gezeigt. Die größte Neuerung sind die zwei RISC-V-Kerne, die zusätzlich zu den zwei ARM-Kernen eingebaut sind. In unserem Test konnten wir feststellen, dass der Pico 2 doppelt so schnell ist wie sein Vorgänger.





- 86 Sandbilder erstellen Sandmaltisch, Teil 2
- 92 Die Soft-Matrix
- 98 Apple II Die Nostalgie-Zeitreise
- 103 Zeitzeugen-Interview mit John Romero
- 106 Portraitobjektiv aus Holz
- 116 Funktechniken fürs Smart Home
- 124 Pi Pico 2
- 128 Kurzvorstellungen
- 130 Impressum/Nachgefragt

Themen von der Titelseite sind rot gesetzt. Grafik Titelthema Funk: 62357/freepik.com

Textilien bedrucken wie die Profis

Braucht der Kleiderschrank etwas Neues? Wir zeigen euch, wie ihr T-Shirts und andere Textilien zu Hause mit professionellen Ergebnissen im Siebdruckverfahren gestalten könnt. Der klassische Siebdruck ist aufwendig, weil man eine Dunkelkammer und Chemikalien braucht. Wer einen Lasercutter hat, kann es einfacher machen. Nur ein einfach zu bauender Siebdruckrahmen und wasserfester Holzleim werden noch benötigt.

82 Siebdruck mit Leim und Laser



Smart-Home-Funktechnik im Überblick

WLAN, Bluetooth, Zigbee, Z-Wave, LoRaWan oder Matter. Wer das eigene Zuhause in ein Smart Home

verwandeln möchte, dem steht eine große Auswahl an drahtlosen Techniken zur Verfügung. In unserer Übersicht zeigen wir, welche Smart-Home-geeigneten Funktechniken es gibt und was deren Vorzüge und Nachteile sind.



116 Funktechniken fürs Smart Home

Leserforum

Sehr hilfreich

Modulare Frontplatten, Make 3/24, S. 16

Ihr Artikel von Gert Michaelis über das Mosaik-Frontplattensystem hat mir bei einem Projekt sehr weitergeholfen. Seit Jahren möchte ich ein Anzeigentableau bauen, das ähnlich aussieht wie in großen Leitwarten. Ich hatte nur keine Idee, wie ich die Anzeigenfelder konstruieren soll.

Das Mosaik-System hat mir nun eine Idee zur Umsetzung gegeben, siehe Fotos anbei. Ich habe das Skript angepasst für M2/M3-Zylinderkopfschrauben und das Ganze dann mit einem selbst konstruierten Modulargehäuse kombiniert. Die Anzeige selbst ist mit RGB-Streifen realisiert und wird von einem Arduino kontrolliert. Ein Raspberry Pi führt im Hintergrund die Überwachung aus, pingt Systeme, fragt Status-Webseiten und MQTT-Broker ab und färbt dann die Felder passend ein.

Der erste Gehäuseentwurf war mit einer Art Diarahmen, in dem das Anzeigenfeld eingebaut ist. Das habe ich dann noch vereinfacht mit milchigem Acryl, mit dem Laser auf Fenstergröße passend geschnitten und mit Klebefolie beschriftet.

Andreas

Fehler beim Preisrahmen

Strompreise im Auge behalten, Make 4/24, S. 8

Der Fehlerteufel hat sich eingeschlichen. Auf der Adapterplatine 2 mm (1/10") sind drei SMD-Widerstände verlötet. Einer hat 0 Ohm und dient als Brücke, zwei weitere haben 10 KOhm.



Leider ist im Schaltplan nur einer davon vorhanden. Alle Nachbauer, die die Adapterplatine verwenden, werden den Fehler nicht merken, die anderen leider schon. Hier der korrigierte Schaltplan, den wir auch auf Github im Projekt-Repository hinterlegt haben.

Toleranzen einhalten

Programmieren mit Sming, Make 4/24, S. 94

Vielen Dank für den interessanten Artikel "Programmieren mit Sming"! Ein Thema waren darin auch Time-of-Flight-Sensoren. Ich würde solche Sensoren gern zur Überwachung eines Sähgeräts verwenden. Dazu brauche ich eine stabile Distanzmessung mit Toleranz von +/-1 mm. Leider liefert der ToF-Sensor VL6180X selbst nach Kalibrierung, bei unterschiedlichen Messmethoden (Single-Shot/Continuous) und unabhängig von der Sensor-Charge bei real 6 mm stabilem Abstand immer Zufallszahlen zwischen 4 und 10 mm. Gibt es irgendeine Möglichkeit, den Sensor so zu tunen, dass die Toleranz eingehalten wird?

Uwe Beierlein

Wenn ich mir das Datenblatt ansehe, ist schon das Rauschen bei 2 mm. Der Sensor ist wohl nicht für so kleine Entfernungen und Genauigkeiten gemacht. Ist halt ein bezahlbarer Sensor. Für ein Sähgerät vielleicht nicht das Richtige; ich denke, er arbeitet auch eher an glatten Oberflächen. Mit einem Mittelwert zwischen 10 und 4 mm ist er ja mit 7 mm schon mal nah dran. Weiter habe ich mich bisher auch nicht mit dem Sensor beschäftigt. Viele Grüße Carsten Wartmann

Stolzer Papa

Löt-Makey, Maker Faire und Magazin

Liebe Make-Redaktion, vor ungefähr einem Dreivierteljahr habe ich zum ersten Mal mit meinem Sohn (nun sechs Jahre alt) einen

Preisrahmen (Tibber)





Makey-Bausatz gelötet. Heute ist mein Geburtstag und er hat mir ein selbst gemaltes Bild geschenkt. Darauf zu sehen: zwei Personen, die auf Bürostühlen an einem Schreibtisch sitzen. Auf dem Monitor vor ihnen ist die Makey-Bauanleitung zu sehen. Daneben wurde der Makey in echter Größe abgemalt. Ich glaube, er hat diesen Tag in sehr schöner Erinnerung. Vielen Dank für diesen Stolzer-Papa-Moment!

Stolzer Papa

Gern geschehen. Wir freuen uns immer wieder, auch die Jüngeren fürs Selbermachen begeistern zu können!

Zu teuer

Ikea-Hack-Sandmaltisch, Make 4/24, S. 16

Vor ein paar Tagen lief ich Gefahr, den Frühstückskaffee wieder auszuspucken, als ich die Kosten für den IKEA-Hack-Sandmaltisch im Heft 4/24 gesehen habe. 500 €! Für ein Spielzeug, welches Kinder am Strand oder im Sandkasten mit den Fingern reinmalen. Never ever würde ich für so etwas so viel Geld ausgeben, das keinen Mehrwert hat. Einfach eine Spielerei. Haben die Leute zu viel Geld?

Da ist der batteriebetriebene Rahmen ein paar Seiten vorher viel "sinnvoller", da es mir doch etwaige Mehrkosten ersparen könnte. Ich bin langjähriger Leser der Make-Zeitschrift, von Beginn an, habe mir fast alle Hefte durchweg gekauft, bis auf die letzten zwei oder drei. Mich wundert es nicht, dass sich alte Elektroniker-Hasen öfters mokieren.

Alexander Kominik

Ein Mondrian an der Wand hat sicher auch keinen weiteren Mehrwert, als seinen Besitzer zu beglücken. Und um damit angeben zu können ;-) Ein teures Projekt pro Jahr können wir schon mal ins Heft bringen, oder?





Der Schimpfroboter

Wieder eine sachliche Debatte verloren, weil kein Schimpfwort parat war? Oder blamiert, weil "Du Doof...ding" oder "Sie Ääähmwurst...fisch" einfach nicht zünden wollte? Ja, das ist schlimm. Zeit, unbesiegbar zu werden und mit Schimpfolino immer eine von über zwölf Millionen möglichen Beleidigungskombinationen parat zu haben!

von Nikolai Radke



Miefender Nacktfladen Schimpfolino besteht aus einem Display, einem Knopf und ist sofort schimpfbereit. Kein Schnickschnack, handlich, klein und energiesparend. Und kindgerecht sind seine Beleidigungen auch, obwohl er so ein Halunke ist.

Wer oder was ist Schimpfolino?

Kurz: ein ATtiny85, der auf Knopfdruck zufallsbasiert Schimpfwörter aus einem EEPROM liest, diese zusammensetzt und auf dem Display ausgibt, um anschließend nach acht Sekunden in den Tiefschlaf zu fallen. Anders gesagt ist Schimpfolino ein unflätig schimpfendes Monster, hier in einer Nokolino-Hülle, zum Selberbauen. Ganz schön profan, aber lustig.

Nokolino? Vielleicht erinnert sich die Leserschaft an Noko, das Monster aus den Heften 4/16 bis 1/17 und dessen kleinen Bruder Nokolino aus Heft 1/18, der auch auf Workshops bei einigen Maker Faires gebaut werden konnte und als Bausatz eine Weile im Shop erhältlich war. Was hat Schimpfolino nun damit gemein? Der Schimpfwortgenerator, allerdings erheblich verbessert und erweitert, entstammt Noko, und die Gestaltung des Gehäuses dem kleinen Nokolino.

Display, ATtiny, EEPROM und ein Knopf – das ist technisch in der Tat nichts Besonderes. Das Projekt lebt aber von der witzigen Idee. Der wahre Schatz liegt im EEPROM: 8 KB seltsame Wörter, das sind 800-mal zehn Zeichen, die allesamt witzig und dennoch jugendfrei und somit stilvoll sind. Das konzentrierte Böse.

Was wir vorhaben

In diesem ersten Artikel der dreiteiligen Reihe (siehe Kasten "Schimpfolino – vom Steckbrett zur handlichen Wunderwaffe") soll ein Schimpfolino programmiert und auf dem Steckbrett gebaut werden. In der Make 2/24 ging der Artikel "ATtiny statt Arduino" umfangreich auf die Programmierung neuerer ATtiny ein und richtete sich dabei an erfahrenere Nutzer, die die Kinderstube (also die Arduino-IDE) verlassen und weiterführend entwickeln wollen. Der



Bild 1: Schimpfolino-Erfinder Nikolai Radke

Kurzinfo

- » Hard- und Softwareaufbau eines handlichen Schimpfwortgenerators
- » Aus fünf begrenzten Wortlisten zwölf Millionen mögliche Kombinationen erzeugen
- » Aufspielen eines Sketches auf den ATtiny85



Schimpfolino-Artikel geht einen Schritt zurück und wendet sich an Nutzer, die sich vor SMD-Lötarbeiten fürchten, so wie es beim armen Autor der Fall ist, oder die noch nie in Berührung mit den Winzlingen gekommen sind, aber gerne platz- und stromsparend entwickeln möchten. Und einfach an jene, die Spaß am Beleidigen haben.

Schimpfolino – vom Steckbrett zur handlichen Wunderwaffe

In diesem und den folgenden beiden Ausgaben wird erklärt, wie aus einer Idee ein fertiges Gerät auf einer Platine in einem Gehäuse entwickelt wird. Wer, aus welchen Gründen auch immer, keinen Bedarf für einen Schimpfolino hat, kann anhand dieser Reihe dennoch Techniken und Tricks für eigene Projekte lernen.

Teil 1: Hard- und Software

In diesem Artikel wird die Funktionsweise des Schimpfolinos beschrieben. Welche Hardware verwendet wird, wie die Software funktioniert und vor allem, wie man zwölf Millionen Kombinationsmöglichkeiten programmiert. Auf einem Steckbrett wird gezeigt, wie ein ATtiny85 programmiert und ein erster Schimpfolino gebaut wird.

Teil 2: EEPROM und Strom sparen

Ein ATtiny85 braucht nur wenig Strom. Wie aber die Lebensdauer der Batterie von einigen Tagen auf viele Jahrzehnte erweitert werden kann, wird im zweiten Teil erklärt. Ebenso wie ein EEPROM extern beschrieben und intern gelesen werden kann und dann zu Schimpfolino hinzugefügt wird.

Teil 3: Vom Steckbrett zur Platine

Eine gedruckte Schaltung zu entwerfen und zu bestellen, ist viel einfacher, als viele denken. Die Vorteile liegen auf der Hand: einfaches Bauen, Platz sparen, weniger Einstreuungen und, und, und. Der letzte Artikel zeigt Schritt für Schritt, wie mit KiCad eine Platine entworfen und bestellt werden kann. Anschließend geht es um den Einbau in ein Gehäuse und fertig ist der Schimpfolino.



Bild 2: Mit einem einzigen Taster wird der Schimpfolino bedient.

Bild 3: Eine CR2023-Batterie dient als Stromversorgung.

Schimpfolinos Innenleben

Die Programmierung steckt in einem ATtiny85 (siehe Kasten "ATtiny85"), der Wortschatz in einem 24AA64. Die Anzeige erledigt ein 0,96-Zoll-SDD1306-Display (ca. 24 Millimeter Bilddiagonale). Bedient wird der Schimpfolino mit einem Taster (Bild 2). Die beiden ICs wurden gesockelt, um sie jederzeit neu beschreiben zu können. Gemäß Datenblatt des ATtiny85 und des EEPROMs wird jeweils ein Kondensator von 100 nF zur Stabilisierung der Spannung empfohlen. Nach ausführlichen Messungen haben sich die Stützkondensatoren als überflüssig erwiesen und werden deswegen hier weggelassen.

Als Stromversorgung dient eine handelsübliche CR2023-Batterie in einem Batteriefach (Bild 3). Das Display wurde mit einem doppelseitigen Klebepad auf der Platine befestigt, um es stabil zu halten und die empfindliche Rückseite zu schützen. Die Platine selbst ist eine Eigenkreation und wird im dritten Artikel behandelt.



Das Display habe ich wie auch das EEPROM über den I²C-Bus mit dem ATtiny verbunden. Der I²C-Bus wird im folgenden Artikel vorgestellt.

ATtiny85

Der Winzling aus dem Hause Microchip, ehemals Atmel, wird seit 2005 gefertigt und ist gerade bei Arduino-Fans sehr beliebt. Er kann leicht über die bekannte IDE programmiert werden und ist dank der DIP8/DIL8-Bauweise in Steckbrettern einzusetzen. DIP steht für Dual Inline Package und DIL für Dual in Line. Beide Schreibweisen können synonym verwendet werden. Mit 8-KByte-Flashspeicher lassen sich auch komplexere Projekte realisieren. Wer mehr Pins benötigt, kann auf den ATtiny84 (DIP14) zurückgreifen.

Die Software des Schimpfolino kommt mit weniger als 4-KByte-Flashspeicher aus und würde daher auch mit dem kleinen Bruder des ATtiny85, dem ATtiny45, auskommen. Der Wortschatz benötigt 8 KByte zusätzlichen Speicher, der sich mit einem externen EEPROM erweitern lässt. Da das EEPROM mit dem Wortschatz aber erst im nächsten Artikel beschrieben wird, verwenden wir bis dahin einen abgespeckten Wortschatz aus 450 Wörtern, der noch gerade so in den Flashspeicher eines ATtiny85 passt. Im Folgenden wird beschrieben, wie die Software aufgebaut ist. Erschaffen wir ein Monster!

Schimpfolinos Software

Der Aufbau ist simpel. Wird die Batterie eingelegt, geht Schimpfolino sofort in den Tiefschlaf. Drückt man einen Knopf, wird für acht Sekunden eine Beschimpfung angezeigt. Diese erstellt der Schimpfwortgenerator, wie das Flussdiagramm (Bild 4) verbildlicht. Danach legt sich der Schimpfolino wieder schlafen, bis der Knopf wieder gedrückt wird.

Insgesamt verwendet Schimpfolino fünf Wortlisten. Jede Beschimpfung besteht aus drei Teilen und wird hier am Beispiel "Nervender Mistbeutel" erklärt. Damit immer wieder neue Schimpfwörter entstehen, ist auch hier ein Zufallsgenerator entscheidend. Dieser wird im Kasten "Zufallszahlen" näher beschrieben.

Wird der Knopf gedrückt, wird ein zufälliges Wort aus Liste 1 eingelesen und in einem Buffer gespeichert, dabei werden die Buchstaben gezählt. Die direkte Ausgabe funktioniert nicht, da der Text zentriert wird. Schließlich soll stilvoll geschimpft werden, dazu muss die Wortlänge bekannt sein. Nun wird das Wort zentriert auf dem Bildschirm ausgegeben.

ATtiny85

Technische Daten

- » 8-Bit-Risc-Mikrocontroller
- » 2,7 bis 5,5 V Versorgungsspannung
- » Bis 20 MHz Taktfrequenz
- » 6 I/O-Pins
- » 8-KByte Flash
- » 5 PWM-fähige Pins
- » 512-Byte-SRAM
- » I²C, SPI, 4 ADC (10 Bit)
- » 512-Byte-EEPROM
- » DIP8/DIL8-Bauform

Projekt



© Copyright by Maker Media GmDH. 5/2024 |

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.



Zufallszahlen

Schimpfolino verwendet die random(a, b)-Funktion, bei der eine Zahl zwischen a und b – 1 ausgewählt wird. Allerdings liefert diese Funktion lediglich Pseudo-Zufallszahlen aus einer sehr, SEHR langen Kette, vielleicht vergleichbar mit den Nachkommastellen von Pi, nur endlich. Es würden also bei jedem Mal Batterieeinlegen dieselben Wörter erscheinen, bei allen Schimpfolinos, was ein Wortduell ziemlich witzlos machen würde, weil beide dasselbe sagen würden.

Die Funktion randomSeed() schafft hier Abhilfe: Mit einem Parameter vom Typ long wird der Startpunkt, also der Seed, in der Kette ausgewählt, was angesichts der vielen möglichen Zahlenwerte einen guten Eindruck von Zufall vermittelt.

Wie wird der Seed gewählt, da er ja selbst zufällig erzeugt werden muss? Eine oft

verwendete Möglichkeit ist das Auslesen des Rauschens an einem unbenutzten Port des Mikrocontrollers: random-Seed(analogRead(A0)).

Schimpfolino geht aber einen anderen Weg: Nach dem Einlegen der Batterie geht er in den Schlafmodus. Beim ersten Knopfdruck wird die Zeit zwischen Aufwachen und dem Loslassen des Knopfes in Millisekunden gezählt: random-Seed(millis()). Die Funktion millis() zählt die Millisekunden nach dem Start oder Aufwachen des ATtiny aus dem Tiefschlaf, die als Seed verwendet werden. Das funktioniert wunderbar. Wer es noch feiner haben will, kann auch Mikrosekunden mit micros() zählen lassen. Fazit: Für die Erzeugung von wirklich echten Zufallszahlen ist nichts besser als das irrationale Verhalten des Menschen.





Zunächst wird das Genus (Variable genus im Sketch) per Zufall ermittelt: Maskulinum, Femininum oder Neutrum. Bei feminin bleibt das Wort, wie es ist, bei maskulin wird ein "r", bei Neutrum ein "s" angehängt. In diesem Beispiel ist das Wort maskulin, also wird ein "r" angehängt. Der erste Teil des zweiten Wortes wird aus Liste 2 zufällig ermittelt und gespeichert. Hier ist es "Mist".

Für den zweiten Teil des zweiten Wortes ist wiederum das bereits ermittelte Genus entscheidend: Je nach Geschlecht wird auf eine entsprechende Wortliste zugegriffen. Da das zuvor ermittelte Geschlecht maskulin ist, wird auf Liste 4 zugegriffen und der wieder zufällig ermittelte zweite Teil, hier "beutel", an den ersten Teil des Wortes im Speicher angehängt, sodass daraus nun "Mistbeutel" wird.

Das zusammengesetzte zweite Wort wird zum Schluss in einer neuen Zeile zentriert ausgegeben. Nun steht "Nervender Mistbeutel" auf dem Display. Fertig.

In jeder der fünf Listen des EEPROMs befinden sich 160 Wörter, zusammen also 800. Die Anzahl der Kombinationen beträgt demnach 160 \cdot 160 \cdot (160 + 160 + 160) = 12.288.000. Das sollte eine Weile reichen.

Den ATtiny flashen

Im Gegensatz zu einem Arduino-Board hat ein ATtiny-Chip keine USB-Schnittstelle, um mithilfe des bordeigenen USB-zu-seriell-Interface einfach bespielt werden zu können. Abhilfe schaffen sogenannte Programmer, die diese Aufgabe für den ATtiny übernehmen. Die sind schon für weniger als zehn Euro zu bekommen. Einfach geht es aber auch mit einem Arduino Uno/Nano als ISP (In System Programmer).

Zunächst muss die ATtiny-Unterstützung in der Arduino IDE installiert werden (siehe Kasten "Arduino-IDE vorbereiten"). Anschließend wird der Arduino selber vorbereitet, indem der entsprechende ISP-Sketch aufgespielt wird. Dazu muss er mit dem PC verbunden werden. Unter "Werkzeuge/Board" prüft man, ob das richtige Board mit richtigem Port eingestellt ist. Nun wird unter "Datei/Beispiele/ ArduinoISP/ArduinoISP" der benötigte Sketch geöffnet und das mit beherztem Druck auf den Pfeil hochgeladen.

Arduino-IDE vorbereiten (2.X)

- » Unter "Datei/Einstellungen" im Feld "Zusätzliche Boardverwalter URLs" wird folgende URL eingetragen: http://drazzy.com/package_drazzy.com_index.json
- » Mit "OK" bestätigen
- » IDE neu starten
- » Im Menü "Werkzeuge/Board/Board Verwaltung..." den Boardverwalter starten
- » "ATTinyCore von Spence Konde" suchen und auf "Installieren" klicken

Jetzt ist Schimpfolinos Software an der Reihe. Der angegebene Link in der Kurzinfo führt zum GitHub-Repository. Dort kann man die Dateien beim blauen Button "Code" als Zip herunterladen, um sie in den Sketchordner zu entpacken. Darin sind unter anderem zwei verschiedene Schimpfolino-Sketche samt Libraries enthalten. Dann muss man die IDE neu starten.

Der ATtiny wird mithilfe eines Steckbretts, sechs Dupont-Kabeln und einem Elko wie in Bild 5 dargestellt verbunden. Die kleine Kerbe des ATtiny muss nach links zeigen. Vorher bitte den Arduino vom PC trennen.

Bevor der ATtiny bespielt werden kann, muss noch der Bootloader gebrannt werden. Das ist eigentlich Quatsch, da gar kein Bootloader verwendet wird. Es werden lediglich Fuses, also Sicherungen gesetzt, um den ATtiny zu konfigurieren, aber die IDE nennt auch diesen Vorgang so (siehe Kasten "Fuses setzen"). Dabei wird auch die Taktfrequenz festgelegt. Schimpfolinos ATtiny läuft mit 8 MHz und dem internen Taktgeber.

Ein "echter" Bootloader wird benötigt, um Software über einen anderen Weg auf den ATtiny zu spielen. Braucht Schimpfolino aber nicht, nimmt nur Platz weg.

Verbindung und ATtiny sind nun vorbereitet. Im Sketchbook sollte der Ordner "Schimpfolino" mit seinem Unterordner "src" liegen. Dort befinden sich zwei Einträge: "Schimpfolino" und "Schimpfolino_noeeprom". Da das EEPROM erst im nächsten Artikel behandelt wird, muss, Schimpfolino_noeeprom" verwendet werden, die Variante mit eigenem, abgespeckten Wortschatz. Der Sketch wird geöffnet, angepasst werden muss nichts, alles ist böse genug. Das Aufspielen erfolgt jedoch nicht, wie womöglich gewohnt, über den Button mit dem Pfeil, sondern über das Menü: "Sketch/Mit Programmer hochladen" und der ATtiny wird bespielt. Dabei sollten die LEDs des Arduinos hektisch blinken. Sofern keine Fehlerausgabe in der IDE erfolgt, ist der ATtiny nun bespielt.

Schimpfolino auf dem Steckbrett

Es ist an der Zeit, Schimpfolino zum Leben zu erwecken. Mithilfe des Steckbretts und der Dupont-Steckverbindungen ist das auch ganz leicht, wie die Bilder 6 und 7 zeigen. Je nach Hersteller des Displays kann die Anordnung der Pins auf dem Displaymodul von der Fritzing-Grafik in Bild 7 abweichen. SCL wird manchmal synonym mit SCK verwendet, der Unterschied spielt in diesem Zusammenhang keine Rolle.

Fuses setzen

- » Den Arduino mit dem PC verbinden. Unter "Werkzeuge/Port" ggf. den verbundenen Port anpassen
- » Unter "Werkzeuge/Programmer/Arduino als ISP" einstellen
- » Unter, Werkzeuge/Board/ATTinyCore/ ATtiny25/45/85 (No bootloader)" einstellen
- » Unter "Werkzeuge/B.O.D. Level [...] B.O.D. Disabled (saves power)" auswählen
- » Unter "Werkzeuge/Chip" nun ATtiny85 auswählen
- » Unter "Werkzeuge/Bootloader brennen" starten
- » Unter "Clock Source [...]" 8 MHz (internal) auswählen

"Dabei ist zu beachten, dass die Pins genau so angeschlossen werden, wie in den Bildern.

Das geübte Maker-Auge sieht natürlich sofort, dass die Stromversorgung des Displays an einem der Pins des ATtiny hängt. Wie die Schaltung im Detail funktioniert, wird aber erst im nächsten Artikel beschrieben.

Zum Abschluss wird die Batterie eingesetzt. Jetzt läuft Schimpfolino und kann mit einem Knopfdruck geweckt werden, um die geneigte Schöpferin oder den geneigten Schöpfer angemessen zu begrüßen.

So geht es weiter

Im nächsten Heft bekommt Schimpfolino das externe EEPROM und es wird erklärt, wie mit Tiefschlaf Strom gespart und er mit Interrupts geweckt werden kann. Schimpfolino freut sich auf ein Wiedersehen. Faules Flötengejodel. — mch



Bild 6: Die Pinbelegung des ATtiny85. Die Farben entsprechen den Dupont-Kabeln in Bild 7.



Bild 7: Schimpfolino auf dem Steckbrett

fritzing

Verkaufsautomat mit ESP

Eigene Waren zu vertreiben, kann ganz schön aufwendig sein. Mit einem selbst gebauten Automaten geht das ganz einfach. Einmal eingestellt, übernimmt er die Transaktionen und meldet Verkäufe und Fehler ans Smartphone. Man muss sich nur gut überlegen, wie Hard- und Software zusammenspielen sollen. Dann lässt sich sogar im Schlaf Geld verdienen.

von Tobias Vogel



A Is unser damals 14-jähriger Sohn erklärte, dass er imkern wolle, nahmen wir ihn zunächst nicht ernst. Zwei Jahre und einige Imkerkurse später stand dann aber der erste Eimer mit 14 kg frisch geschleudertem Honig auf dem Tisch – und wir fragten uns plötzlich, was wir mit so viel Honig machen sollten. Als Servicetechniker unter anderem für Transaktionsautomaten hatte ich die Idee, einen kleinen Verkaufsautomaten zu bauen. Also überlegten wir uns gemeinsam ein Design und nannten die Erfindung, frei nach Dr. Heinz Doofenshmirtz aus Disneys Phineas und Ferb, "Vendinator".

Da das Projekt insgesamt recht komplex ist, habe ich jeden Funktionsbereich des Automaten einzeln entwickelt und schließlich alles miteinander verbunden. In diesem Artikel gehe ich auf die wesentlichen Komponenten ein. Um das Projekt nachzubauen, benötigt ihr zusätzlich den vollständigen Schaltplan und die Projektdateien aus dem GitHub-Repository des Projekts (siehe Link in der Kurzinfo).

Der Aufbau des Vendinators

Der Automat besitzt sechs Fächer für die Honigware. Diese können mithilfe elektronischer Schrankschlösser geöffnet werden (Bild 1). Ihr jeweiliger Zustand lässt sich aufgrund eingebauter Schalter erkennen. Als Benutzerschnittstelle nutze ich eine 4×4-Matrixtastatur (Keypad) und ein 4-Zeilen-LC-Display, mit dem man durch den Verkaufsprozess oder die Wartung des Automaten navigieren kann.

Natürlich benötigt ein Automat auch einen Münzprüfer. Ein kostengünstiges Modell, das bis zu sechs verschiedene Münzen akzeptiert und auf alle Münzen trainierbar ist, findet man unter dem Suchbegriff "Multi Coin Acceptor". Dieser wird von verschiedenen Herstellern vertrieben. Das Gerät im Vendinator hat seitlich sechs LEDs, zwei Siebensegment-Anzeigen und drei Tasten. Eine Bezugsquelle findet ihr über den Link in der Kurzinfo.

Damit man als Kunde den Verkaufsvorgang auch abbrechen kann, ohne sein Geld zu verlieren, besitzt der Automat eine Zwischenkasse (auch Escrow genannt), in der das eingezahlte Geld vorübergehend gesammelt wird. Dieser Behälter ist 3D-gedruckt und kann mithilfe eines Servos (SG90) in zwei Richtungen kippen, sodass bei einem erfolgreichen Verkauf (Warenausgabe) das Geld in die Kasse und bei einem Abbruch in ein Ausgabefach gelangt.

Hinzu kommen verschiedene kleinere Bauteile: ein Taster, der zu Beginn jeder Transaktion gedrückt werden muss, um den Vendinator einzuschalten, sowie drei Relais und acht MOSFETs. Diese schalten über einen Programmierbefehl die Türschlösser, regeln die Stromzufuhr für den Münzprüfer und kappen nach einer bestimmten Zeit im Leerlauf die Strom-

Kurzinfo

» Einen eigenen Verkaufsautomaten bauen
 » GPIOs mit Port-Expandern erweitern
 » Münzprüfer programmieren und trainieren

Checkliste	Material
Checkliste Seitaufwand: mindestens ein Wochenende mindestens ein Wochenende Sosten: To bis 200 Euro (für die Elektronik) Ducker 3D-Drucker 3D-	 Material NodeMCU-Devkit mit ESP8266 2 Port-Expander PCF8574P, DIP 16 Inverter-IC 74HCT540N 8 MOSFETs IRL 540N 3 Dioden 1N4001 13 Widerstände 10 kΩ 8 Widerstände 1 kΩ Drucktaster 12 mm Durchmesser, 12 V 6 elektromagnetische Schlösser 12 V 4×4-Matrixtastatur Münzprüfer für 6 Münzarten SG90 9g Servo LC-Display-Modul 3,3 V, I^cC Step-down-Wandler LM2595S Step-up/-down-Wandler XL6019 Blei-Akku mit 12 V 3 5-V-Relais HK19F 30mA-LEDs 6 grün, 1 gelb, 1 rot Leiterplatten-Anschlussklemmen 2-polig Platinenmaterial oder vom Anbieter gefertigtes PCB Schrauben, Kabel und Material für ein Gehäuse
 » Gustav Wostrack, Der Weg zur Platine, Teil 1, Make 6/21, S. 104 Alles zum Artikel im Web unter make-magazin.de/xkjn 	

versorgung des Vendinators. Der Strom, den ein 12-Volt-Akku liefert, wird von einem Stepup/-down-Wandler als 12 V und durch einen Step-down-Wandler als 5 V an die jeweiligen Elektronikkomponenten weitergeleitet.

Um bei der Programmierung die Zustände der digitalen Ausgänge besser überprüfen zu können, habe ich dem Gerät noch ein paar LEDs spendiert: sechs grüne für die Türschlösser, eine gelbe für das Münzprüfer-Relais und eine rote für das Abschalt-Relais.

Als Steuereinheit verwende ich ein Node MCU-Devkit, das ich noch in der Bastelkiste herumliegen hatte. Der ESP8266 auf dem Board verfügt über 4 MB Flash-Speicher, was für den Vendinator völlig ausreicht. Mithilfe von WLAN kann er zudem über einen Telegram-Chatbot mit der Außenwelt kommunizieren und Verkäufe melden. Zum Programmieren habe ich die Arduino IDE 1.8.19 verwendet.

Nicht genug Ports

Damit wir alle elektronischen Komponenten des Vendinators mit dem Mikrocontroller

ansteuern können, benötigen wir insgesamt 22 GPIO-Ports. Das NodeMCU-Devkit hat aber nur 17 GPIOs, von denen einige zudem für spezielle Funktionen reserviert sind wie SPI oder das Aufwecken aus dem Deep-Sleep.

Um die Anzahl der verfügbaren Ports zu erweitern, bietet sich ein Port-Expander an. Da das Display und das Keypad über I²C angesteuert werden, lag es nahe, einen I²C-Port-Expander zu verwenden. Ich habe mich schon immer gerne für den PCF8574 entschieden, der praktischerweise noch im DIP-Gehäuse erhältlich ist, das sich leicht auf einer Lochrasterplatine verlöten lässt.

Man kann insgesamt bis zu acht dieser Bausteine parallel verwenden und über die I²C-Pins (SDA, SCL) mit dem Mikrocontroller verbinden. Je Expander erhält man dann 8 weitere I/O-Pins (insgesamt maximal 64), die mit P0 bis P7 gekennzeichnet sind (Bild 2).

Damit man die Expander und ihre Ports unabhängig voneinander ansprechen kann, hat jeder Expander-Chip drei Adresspins (A0, A1 und A2), mit deren Hilfe man seine I²C-Adresse festlegen kann. Dafür verbindet man diese



jeweils mit VCC (3,3 V) oder GND. Da es verschiedene Versionen des Expanders gibt, variieren die Adressbereiche je nach Chip zwischen 0x00 und 0x7F (z. B. 0x20 bis 0x27 oder 0x40 bis 0x4E).

Wie die l²C-Adresse eines Gerätes genau lautet, erfährt man entweder im Datenblatt des Herstellers oder ermittelt sie mithilfe eines l²C-Scanner-Sketches (siehe Link in der Kurz-



Bild 2: Pin-out des Port-Expanders PCF8574

info). Diesen lädt man dazu auf den ESP8266 und öffnet den seriellen Monitor der Arduino IDE. Dort gibt der Sketch alle fünf Sekunden die Adressen der gefundenen I²C-Geräte aus. Damit man weiß, welche Adresse zu welchem Gerät gehört, sollte man diese zunächst einzeln an den Mikrocontroller anschließen.

Expander und Display

Der Vendinator verwendet zwei der beschriebenen Port-Expander: einen für das Keypad und einen für die Steuerung der Schlösser. Die Adresspins des ersten PCF8574 habe ich alle mit GND verbunden, während bei dem zweiten A0 und A1 zu GND führen und A2 zum 3,3-V-Port des NodeMCU. Danach verbindet man noch SDA mit D1 und SCL mit D2 sowie VCC mit 3,3 V (Bild 3).

Der Bildschirm mit 4 Zeilen und 20 Zeichen wird ebenfalls vom Mikrocontroller-Board mit 3,3 V Spannung versorgt. Die Hintergrundbeleuchtung ist über einen Jumper auf der Rückseite des Display-Moduls aktiviert. Dort befindet sich auch ein Poti, über das man den Kontrast des Bildschirms anpassen muss, wenn man ihn mit 3,3 V betreibt. Die beiden I²C-Leitungen SDA und SCL sind wie bei den Expandern mit D1 und D2 am NodeMCU verbunden. Um sie zu stabilisieren, habe ich sie zusätzlich mit zwei Pull-up-Widerständen von je 10 k Ω auf HIGH gelegt. In meinem Aufbau hat das Display die Adresse 0x27, der Expander mit allen Adressleitungen auf GND die Adresse 0x38 und der andere 0x3C. Wie bereits erwähnt, können diese Adressen bei euch anders lauten.

Um das Display mit dem Mikrocontroller anzusteuern, nutze ich die Bibliothek "Liquid-Crystal I2C" von Frank de Brabander in der Version 1.1.2, die man über den Bibliotheksverwalter in der Arduino IDE findet. Sie wird zwar für den ESP8266 als inkompatibel angezeigt und gibt beim Kompilieren eine Warnung aus, funktionierte bei mir aber trotzdem einwandfrei.

Um mit den Expandern zu kommunizieren und die Schlösser zu schalten, reicht die Wire-Bibliothek, die mit der Arduino IDE mitgeliefert wird. Nur für das Keypad benötigt man zusätzliche Bibliotheken (dazu gleich mehr).

Keypad für die Eingabe

Das Eingabefeld mit der bestehenden Schaltung zu verbinden, ist nicht weiter aufwendig, da man es der Reihe nach an den Ports (P0 bis P7) des PCF8574 anschließt (Bild 4). Den eigentlichen Zauber übernehmen zwei Bibliotheken. Die eine findet man über den Bibliotheksverwalter als "Keypad" von Alexander Brevig. Sie dekodiert den Tastendruck in eine verarbeitbare char-Variable. Damit dies auch zusammen mit dem Expander funktioniert, bedarf es einer weiteren Bibliothek: Keypad_I2C



Bild 3: Display und Port-Expander sind über I²C am Mikrocontroller angeschlossen.

(siehe Link in Kurzinfo). Nachdem man diese heruntergeladen hat, kopiert man den Ordner "Keypad_l2C" in den Ordner "Library" der Arduino IDE und startet diese neu.

Um den Aufbau zu testen, eignet sich ein Beispiel aus der Keypad-Bibliothek, das man im Menü unter, Beispiele/Keypad_12C/Custom-Keypad_12C" findet. Bei #define I2ADDR 0x38 geben wir statt 0x38 die ermittelte Adresse unseres Expanders ein. Da mein verbautes Keypad vier Reihen und vier Spalten hat, passen die hinterlegten Variablen für ROWS und COLS. Das Layout der Tastatur kann man im Array hexaKeys einstellen:

```
char hexaKeys[ROWS][COLS] = {
  { 'D', '#', '0', '*' },
  { 'C', '9', '8', '7' },
  { 'B', '6', '5', '4' },
  { 'A', '3', '2', '1' }
};
```

Der Anschluss Reihe/Spalte sieht in meinem Aufbau so aus:

byte rowPins[ROWS] = {0, 1, 2, 3}; byte colPins[COLS] = {4, 5, 6, 7};

Was mit Reihe und Spalte gemeint ist, wird klar, wenn man sich den Aufbau einer typischen Matrixtastatur anschaut (Bild 5) – die Taster sind in Reihen und Spalten zusammengeschlossen.

Der Mikrocontroller wählt nacheinander jede Zeile der Tastatur an. Wenn er eine Zeile aktiviert, überprüft er die Spalten, um zu sehen, ob eine Taste gedrückt wurde. Sobald er eine gedrückte Taste findet, speichert der Mikrocontroller die Zeile und die Spalte, um die gedrückte Taste zu identifizieren. Dieser Vorgang wird wiederholt, bis alle Zeilen überprüft wurden.

Man kann die eingegebene Taste auch auf dem LCD anzeigen (siehe Listing "Keypad-LCD"). Neben der Display-Bibliothek, der Konfigurationszeile und Initialisierung im void setup() habe ich schon die Zeilen lcd.set-Cursor(0,0) und lcd.print("Taste: ") eingefügt. So wird gleich zu Beginn der Text "Taste: " in der ersten Zeile ausgegeben. Im void loop() wird mit lcd.setCursor(9,0) zuerst der Cursor auf das neunte Zeichen in der ersten Zeile gesetzt und dann mit lcd.print(customKey) angezeigt.

Inverter für die Schlösser

Der zweite Expander steuert die sechs 12-V-Türöffner, die beiden verbleibenden Ports benötigt man für das Aktivieren des Münzprüfers und das Abschalten des Vendinators. Acht LEDs zeigen zusätzlich den aktuellen Zustand der Ports an, um die Programmierung zu erleichtern.

Der PCF8574 ist dafür allerdings nur bedingt geeignet, denn erstens sind seine Ports Open-Collector-Ausgänge und können nur sehr wenig Strom bei einem HIGH-Pegel liefern (in der Größenordnung von ca. 100 μ A). Sie können jedoch bis zu 25 mA aufnehmen, was ausreicht, um etwa eine LED leuchten zu lassen. Zweitens stehen alle Ports des Expanders standardmäßig auf HIGH, wenn man ihn mit Spannung versorgt. Dies würde bedeuten, dass sich bei jedem Neustart des Vendinators alle Türen gleichzeitig öffnen würden – was



Bild 4: Acht Kabel verbinden das Keypad mit dem ersten Port-Expander.

für den Betreiber eines Verkaufsautomaten natürlich inakzeptabel ist.

Um das zu lösen, habe ich hinter die Ports des PCF8574 einen Inverter-IC (74HC540N) geschaltet. Nun sind nach dem Einschalten hinter dem Inverter alle Ports auf LOW. Und da der Inverter mehr Strom liefern kann als der Expander, schafft er es auch, bei HIGH die LEDs zu betreiben.



Bild 5: Schaltschema eines Keypads mit 4×4 Tasten

Türen öffnen und schließen

Von einer Lichtshow alleine gehen die 12-V-Schlösser aber noch nicht auf. Um die Spule eines Schlosses zu bewegen, wird circa 1 A benötigt. Das ist mehr als der Inverter liefern kann. Auf der Suche nach einer Lösung kam ich auf den MOSFET IRL540, der ab einer Spannung von 2V leitend wird und bereits bei 3,3 V ausreichend durchschaltet, um die Schlösser auszulösen (Bild 6). Die zulässige Höchstspannung von 100 V und der maximale Dauerstrom von 18 A werden vom Vendinator nicht ausgereizt.

Um ein Schloss auszulösen, setzt man den betreffenden Ausgang am Expander auf LOW. Der Inverter kehrt das Signal dann auf HIGH um. Dadurch wird der MOSFET leitend und GND wird zum Schloss durchgeschaltet, wodurch es geöffnet wird.

Keypad-LCD

```
#include <Keypad_I2C.h>
#include <Keypad.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#define I2CADDR 0x38
const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
{'D','#',
{'C','9',
{'B','6',
          '0','*'},
'8','7'},
'5','4'},
{ 'A '
      '3',
           '2',
               '1'}
};
byte rowPins[ROWS] = {0, 1, 2, 3};
byte colPins[COLS] = {4, 5, 6, 7};
Keypad_I2C customKeypad(
                  makeKeymap(hexaKeys),
                  rowPins, colPins
                  ROWS, COLS, I2CADDR);
LiquidCrystal_I2C lcd(0x27,20,4);
void setup(){
  Wire.begin();
  customKeypad.begin();
  Serial begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Taste:
                        "):
}
void loop(){
  char customKey =
    customKeypad.getKey();
  if (customKey != NO_KEY){
    Serial.println(customKey);
     lcd.setCursor(9,0);
     lcd.print(customKey);
  }
}
```

Beim Verkauf versucht der Vendinator zehnmal (maxRetries) mithilfe der Wire-Bibliothek, die jeweilige Tür (slot) über den Port-Expander zu öffnen, indem der verbundene MOSFET durchgeschaltet wird (siehe Listing "openslot (gekürzt)"). Dabei gibt der Automat absichtlich nur einen kurzen Impuls auf den Ausgang, um zu verhindern, dass die Spule im Schloss durchbrennt. Wird nach einem Impuls die Tür als offen gemeldet (HIGH an D7), gibt der Vendinator eine Warnung auf dem Display aus: "Bitte Tür wieder schließen." Auch das prüft er dann zehnmal. Durch ein Delay zwischen den Überprüfungen ist die Zeit lang genug, um beguem die Ware zu entnehmen. Sollte die Ware nicht rechtzeitig entnommen und die Tür nicht schnell genug wieder verschlossen werden, schaltet der Vendinator mit einer Fehlermeldung ab. Bleibt die Tür nach dem Neustart weiterhin offen, sendet das System eine Fehlermeldung per Telegram (dazu später mehr).

Münzprüfer trainieren

Der Münzprüfer (Bild 7) kann verschiedene Münzen erkennen und sendet über einen Pin je nach erkannter Münze eine vorher festgelegte Anzahl von Impulsen. Mit dem Node MCU verbunden, bringen diese den Port D5 auf LOW. Damit der Mikrocontroller durch die 12 V, mit denen der Münzprüfer versorgt wird, keinen Schaden nimmt, muss man unbedingt darauf achten, dass alle GNDs miteinander verbunden sind. Mir ist es einmal passiert, dass ich versehentlich die GND-Verbindung zwischen 12 V und 5 V entfernt habe. Dadurch kamen über den Coin-Ausgang des Münzprüfers 12 V an den Port des Mikrocontrollers und haben diesen gebraten.

Damit der Prüfer alle Münzen von 5 Cent bis 2 Euro akzeptiert, muss er auf 6 Münzsorten konfiguriert werden: Bei einer 2-Euro-Münze soll ein Impuls ausgegeben werden, bei einer 1-Euro-Münze sollen es zwei sein und so weiter. Die Sensitivität habe ich auf 8 festgelegt, da ich mit dieser Einstellung gute Erfahrungen gemacht habe. Mit dem Schalter "Fast/Medium/Slow" am Münzprüfer lässt sich die Geschwindigkeit der Impulse angeben (Bild 8). Hier wählen wir "Fast". Den Schalter "NO/NC" stellen wir auf "NC" (Normally Closed).

Den Ablauf der Programmierung zeigt die Tabelle "Münzprüfer konfigurieren". Die Einstellungen für die Anzahl der Trainingsmünzen (H), die Impulse (P) und die Sensitivität (F) wiederholen sich für die jeweiligen Münzen. Sie müssen nur darauf achten, die Anzahl der



Bild 6: Ein Inverter kehrt die Signale für die elektronischen Schlösser um.

Impulse nach jeder Münze um Eins zu erhöhen (P1 = 1 Impuls ... P6 = 6 Impulse).

Ist das eingestellt, muss der Münzprüfer noch angelernt werden. Dafür hält man die SET-Taste so lange gedrückt, bis A1 erscheint. Anschließend wirft man 20 2-Euro-Münzen ein. Das Gerät zählt mit und wechselt automatisch zur nächsten Münzsorte, bis man alle sechs trainiert hat. Um sicherzustellen, dass der Münzprüfer die Münzen später zuverlässig erkennt, sollte man jeweils 20 verschiedene Münzen jeder Sorte einwerfen.

Nach dem Training gibt der Münzprüfer beim Einwurf einer Münze die passende Münzsorte auf seinem Display aus: Bei 2 Euro eine 1, bei 1 Euro eine 2 und so weiter.

Impulse erfassen

Schließt man ein Oszilloskop am Coin-Pin des Münzprüfers an, kann man beim Einwurf einer 5-Cent-Münze das Oszillogramm wie in Bild 9 messen. Man erkennt, dass die Spannung am Ausgang sechsmal auf Masse gezogen. Um diese Impulse mit dem Mikrocontroller auswerten zu können, müssen wir erkennen, wann der erste Impuls erfolgt. Ab diesem Zeitpunkt zählt man für eine bestimmte Dauer die Impulse (siehe Listing "Münzimpulse").

Damit der Mikrocontroller die Impulse des Münzprüfers unmittelbar verarbeitet, verwende ich den Interrupt isr(), den ich im void setup() mit dem Pin coinpin verknüpft habe. Der Wert falling gibt an, dass eine fallende Flanke (von HIGH nach LOW) den Interrupt auslöst. Registriert der Mikrocontroller auf coinpin also ein LOW-Signal vom Münzprüfer, unterbricht er den Main-Loop und regelt mit zwei if-Bedingungen den weiteren Ablauf.

Dafür speichert der Interrupt mit dem Befehl millis() zunächst einen Zeitstempel in der Variable timenow – genauer gesagt die vergangenen Millisekunden, die seit dem Einschalten des Mikrocontrollers vergangen sind. Danach prüft die erste if-Bedingung mit coinnew == false, ob es sich um eine neue Münze handelt. Falls ja, wird der Wert der Variable auf true gesetzt, und es wird mit coinstart ein neuer Zeitstempel erfasst. Zusammen mit der Variable timeout, die den Wert 710 (Millisekunden) hat, ergibt sich anschließend der Timer cointimeout. Dieser setzt ein Zeitfenster, in dem weitere eingeworfene Münzen ignoriert werden.

Danach prüft eine zweite if-Bedingung erneut mit coinnew, ob sich eine Münze im System befindet und noch Zeit für die Erfassung der Impulse bleibt – also ob timenow kleiner als cointimeout ist. Falls ja, wird timenow aktualisiert und coinimpulse hochgezählt. Das passiert so lange, bis der Timer cointimeout abgelaufen ist und entsprechend alle Impulse vom Münzprüfer erfasst wurden.

Danach kehrt das Programm wieder zurück zum Main-Loop, wo eine weitere if-Bedingung abfragt, ob coinnew nach wie vor true, der Timer aber abgelaufen ist. Falls beides zutrifft, wird zum Guthaben credit ein Wert addiert, der von der Anzahl der Impulse (coinimpulse) abhängig ist. Schließlich werden coinnew und coinimpulse zurückgesetzt und das System ist bereit für die nächste Münze.

Escrow – eine Münzweiche

Um die Münzverarbeitung vollständig zu machen, fehlt noch die eingangs besprochene Zwischenkasse. Dazu habe ich mit Tinkercad eine dreieckige Schale konstruiert, die durch einen Servo in zwei Richtungen gekippt werden kann. Die Schale ist in einem Gehäuse untergebracht (Bild 10). Beim Schwenk in die eine oder andere Richtung wird der Inhalt entweder durch die rechte oder linke Öffnung des Gehäuses geleitet. Alle Teile sind zusammengesteckt, sodass man (als Wartungstechniker) die Zwischenkasse leicht zerlegen kann, falls doch mal eine Münze stecken bleibt.

Im Gegensatz zu den anderen Bauteilen ist der Servo an die 5-V-Betriebsspannung angeschlossen. Das PWM-Signal vom Mikrocon-

openslot (gekürzt) void openslot (int slot) { int retries = 0; while (retries < maxRetries) {</pre> Wire.beginTransmission(IOexpander); switch (slot) { case 1: Wire.write(B11111110); break; case 2: Wire.write(B11111101); break: Wire.write(B11111011); case 3: break: Wire.write(B11110111); case 4: break; Wire.write(B11101111); case 5: break; Wire.write(B11011111); case 6: break; default: Wire.write(B11111111); break: Wire.endTransmission();

troller funktioniert mit 3,3 V dann trotzdem problemlos. Im Code heißt der Servo escrow. Mit escrow.write(85) ist er in Grundstellung, escrow.write(0) und escrow.write(180) schwenken die Zwischenkasse zur Kasse oder zum Kunden.

Abschalten und Strom sparen

Die Stromversorgung des Vendinators übernimmt ein 12-V-Akku, da am vorgesehenen Standort keine Steckdose vorhanden ist. Wir benötigen 12 V für die Türschalter und den Münzprüfer und 5 V für den Servo und den Mikrocontroller.

Da die Spannung des 12-V-Akkus nie genau 12 V beträgt, verwende ich einen Stepup/-down-Wandler, um die Eingangsspannung gerade zu ziehen. Das bedeutet, dass der Wandler die Spannung erhöht (Step-up) oder verringert (Step-down), wenn die Eingangsspannung nicht genau 12 V entspricht.



Generell portofreie Lieferung für Heise Medien- oder Maker Media Zeitschriften-Abooreasiges gelensgowaren waren warent von 20 € (innerhalb Deutschlands). Nur solange der Vorrat reicht. Preisänderungen vorbehalten.

In heise shop

Münzimpulse

```
int timeout = 710;
int timenow:
float credit = 0;
int coinimpulse = 0;
bool coinnew = false;
unsigned long coinstart = 0;
unsigned long cointimeout = 0;
static const uint8_t coinpin = D5;
void ICACHE_RAM_ATTR isr() {
  timenow = millis();
  if (coinnew == false) {
    coinnew = true;
    coinstart = millis();
    cointimeout = coinstart+timeout;
  if
     (coinnew == true &&
      timenow < cointimeout) {</pre>
    timenow = millis();
    coinimpulse++:
  1
}
void setup() {
  Serial.begin(74880);
  pinMode(coinpin, INPUT);
  attachInterrupt(coinpin,
                           isr.
                   FALLING);
}
void loop() {
  timenow = millis();
  if
     (coinnew == true &&
      cointimeout < timenow){</pre>
    Serial.printf("Gezählte Impulse
                 %u \n", coinimpulse);
    switch (coinimpulse) {
      case 1: credit = credit + 200;
               break;
      case 2: credit = credit + 100;
               break
      case 3: credit = credit + 50;
               break:
      case 4: credit = credit + 20:
               break:
      case 5: credit = credit + 10:
               break
      case 6: credit = credit + 5:
               break:
    Serial.printf("credit %u \n",
                   credit);
    coinnew = false;
    coinimpulse = 0
  3
}
```



Bild 9: Bei einer eingeworfenen Münze gibt der Coin-Pin am Münzprüfer Impulse aus.



Bild 7: Diese Münzprüfer sind günstig und leicht zu programmieren.

Diese geregelten 12 V gehen dann in ein Stepdown-Modul, das auf 5 V eingestellt ist.

Damit der Akku möglichst lange hält, schaltet sich der Vendinator nach jedem Verkaufsvorgang oder bei Nichtbenutzung ab. Dafür habe ich eine Relaisschaltung entwickelt (Bild 11), die wie folgt funktioniert:

Schaltet man den Automaten ein, um etwas zu kaufen oder ihn zu warten, erhält der erste Spannungswandler Strom. Dieser gibt 12 V an den zweiten Wandler weiter, wodurch 5 V über den Öffnerkontakt des Ausschaltrelais zur Spule des Einschaltrelais fließen. Das Einschaltrelais zieht daraufhin an und der Akku wird dadurch über den Schließer des Einschaltrelais mit dem 12-V-Spannungswandler verbunden, ohne dass der Taster weiter gedrückt werden muss. Das Einschaltrelais hält sich nun selbst.

Um den Vendinator wieder auszuschalten, muss das Ausschaltrelais betätigt werden. Dafür sorgt der Mikrocontroller mithilfe eines MOSFET, wie zuvor bei den Türschaltern beschrieben. Sobald das Ausschaltrelais anzieht, wird das Einschaltrelais nicht mehr mit Strom versorgt und fällt ab. Dadurch gelangt keine Akkuspannung mehr zu den Wandlern und der Automat schaltet sich ab. Da der Akku komplett von der Schaltung getrennt wird, verbraucht der Vendinator bis zum nächsten Betätigen des Einschaltknopfes keinen Strom.

Telegram-Bot meldet Status

Damit wir in Echtzeit über die Verkäufe des Vendinators benachrichtigt werden, habe ich den Automaten an Telegram angebunden – eine Funktion, die meinem Sohn besonders wichtig war. In der Telegram-App sucht man zuerst nach "BotFather". Diesem schickt man den Befehl /newbot und folgt den Anweisun-



Bild 8: Auf der Rückseite befinden sich Anschlüsse und Schalter zum Einstellen.

gen, bis man ein Token erhält, das dem Vendinator die Zugangsberechtigung zu Telegrams Bot-API gewährt – also gut abspeichern. Anschließend holt man sich von dem Bot, "IDBot" mit dem Befehl /getid noch eine eigene Chat-ID. So kann man im Programm festlegen, dass nur der Besitzer dieser Chat-ID Nachrichten vom Bot erhalten kann. Soll der Bot an eine Gruppe senden, muss man den IDBot der Gruppe hinzufügen und innerhalb der Gruppe den Befehl /getgroupid senden. Dann bekommt man die Gruppen-ID zurück.

Um den Bot in einen Arduino-Sketch einzubinden, benötigt man die Bibliothek "UniversalTelegramBot", die man über den Bibliotheksverwalter installieren kann. Der endgültige Vendinator-Sketch zeigt, wie die Implementierung des Chatbots aussehen kann (siehe Link in der Kurzinfo). Man muss lediglich die WLAN-Zugangsdaten, das Token und die Chat-ID eintragen und schon kann es losgehen. Im String statusMessage wird die Nachricht zusammengestellt und mithilfe des Befehls bot.sendMessage() verschickt. Damit sich der Sketch später nicht aufhängt, wenn keine WLAN-Verbindung besteht, prüft er den WLAN-Status im Vorfeld.

Inhalt und Kasse müssen stimmen

Um sicherzustellen, dass die Inhalte der Fächer, die Preise und der Kassenstand jederzeit korrekt sind, habe ich die EEPROM-Emulation des ESP8266 verwendet. Diese nutzt den Flash-Speicher und reserviert bis zu 4 Kilobytes zum Speichern dieser Daten. Der reservierte Bereich kann vom Programm gelesen und beschrieben werden, wobei die gespeicherten Daten auch nach einem Neustart erhalten bleiben. Es ist jedoch wichtig zu be-

Projekt

In heise academy

achten, dass ein Flash-Speicher maximal etwa 10.000-mal beschrieben werden kann. Für die Transaktionen eines Hobbyautomaten wie dem Vendinator reicht das lange aus – sofern man den Schreibbefehl nicht unbedacht im void loop() platziert.

Alles in einen Topf

Für die Entwicklung der finalen Firmware habe ich einen Versuchsaufbau erstellt, der alle benötigten Komponenten umfasst (Bild 12) – und habe dann wie üblich drauflos programmiert. Nach einigen Versuchen gelang es mir, einen funktionierenden Verkaufsvorgang mit Fachauswahl, Bezahlung und das Öffnen eines Fachs zu implementieren. Allerdings wurde der Code mit jeder zusätzlichen Funktion immer unüber-



Bild 10: Mit dieser Zwischenkasse kann man den Bezahlvorgang auch abbrechen.

Schritt	Taste Drücken	Anzeige	LEDs	Beschreibung		
1	ADD + MINUS	A	000000	Konfiguration starten		
2	SET	E	000000	Anzahl erkenn- barer Münzen einstellen		
3	ADD (bis 6 steht)	6	000000			
4	SET	H1	100000	Trainingsmün- zen für 2 Euro einstellen		
5	ADD (bis 20 steht)	20	100000			
6	SET	P1	100000	Impulse für 2 Euro einstellen		
7	ADD (bis 1 steht)	1	100000			
8	SET	F1	100000	Sensitivität für 2 Euro einstellen		
9	ADD (bis 8 steht)	8	100000			
10	SET	H2	010000	Trainingsmün- zen für 1 Euro einstellen		
11	ADD (bis 20 steht)	20	010000			
12	SET	P2	010000	Impulse für 1 Euro einstellen		
13	ADD (bis 2 steht)	2	010000			



© Copyright by Maker Media GmbH. 5/2024 | 21

Ihr Partner für IT-Weiterbildung

Videokurse für IT-Professionals



ChatGPT für den Berufsalltag ChatGPT kann natürliche Sprache analysieren und generieren. Dieses vielseitige Werkzeug eröffnet neue Möglichkeiten für Produktivität und Kreativität im Arbeitsalltag für verschiedenste Aufgaben.



Game-Engineering mit Unity und C#

In diesem Kurs geht es um die Grundlagen der Spieleentwicklung mit Unity. Sie lernen, virtuelle Welten aufzubauen und Programmlogik in C# zu formulieren.



Stable Diffusion: Bilder aus Texten generieren

Lernen Sie anhand des Stable-Diffusion-Modells, wie die KI-basierte Bildgenerierung funktioniert. KI-Experte Dr. Gerhard Heinzerling zeigt alle Inhalte Schritt für Schritt.

Jetzt alle Videokurse 7 Tage kostenlos testen: **heise-academy.de**



Bild 12: Ein erster Funktionstest mit allen benötigten Komponenten

sichtlicher und schwerer lesbar; jede neue Funktion verschlimmerte die Situation nur noch.

Also musste ich noch einmal genau überlegen, welche Anforderungen das Programm erfüllen sollte. Daraus entwickelten sich schließlich zwei klare Ablaufpläne (Bild 13), die ich nach und nach weiter verfeinerte: einer für den Verkaufsmodus und einer für den Adminmodus. Letzterer ermöglicht es unter anderem, Fächer zu befüllen, die Kasse zu leeren und Preise anzupassen. Diesen Modus erreicht man, indem man im Gehäuse des Vendinators den Admin-Switch umlegt und dann den Automaten einschaltet. Beide Pläne könnt ihr im GitHub-Repository des Projekts herunterladen.

Aus den Entwürfen entwickelte ich schließlich einen Zustandsautomaten (engl.: finite state machine), um das Programm wie in der Grafik als eine Reihe von Zuständen zu modellieren, die sich je nach Ereignis oder Bedingung ändern. Dieser verwendet zwei Zustandsbäume: Adminstate und Vendingstate, deren Zustände durchnummeriert und kommentiert sind, um sie leicht lesbar zu machen. Nachdem eine Aktion abgeschlossen ist, kehrt das System immer wieder zum Hauptmenü des jeweiligen Modus zurück.

Die Verwendung des Zustandsautomaten hat mir geholfen, den Sketch strukturierter und verständlicher zu machen. Jeder Zustand und seine Übergänge sind klar definiert und können leicht modifiziert werden. Zudem lässt sich der Code besser lesen, warten und skalieren.

Vom Steckbrett zur Platine

Normalerweise mache ich mir bei kleineren Projekten keinen Plan, sondern übertrage die Bauteile einfach vom Steckbrett auf die Lochrasterplatine und ziehe meine Leiterbahnen "frei Schnauze". Da dieses Projekt jedoch umfangreicher ist, habe ich mir in KiCad eine Platine konstruiert und bei JLCPCB fertigen lassen. Obwohl meine letzte Erfahrung mit einem Layoutprogramm ins Jahr 1997 zurückreicht, bin ich mit dem Programm gut zurechtgekommen, nicht zuletzt, weil es eine große Bauteilbibliothek enthält. Nur die Zuordnung der Bauteile zu den Footprints, also der tatsächlichen Form auf der späteren Platine, bereitete mir ein paar Probleme. So musste ich beispielsweise den Footprint des NodeMCU etwas in der Breite anpassen.

Nach einer zweiwöchigen Wartezeit hielt ich dann endlich meine Platinen in den Händen. Das Bestücken der Bauteile ging relativ schnell (Bild 14). Beim ersten Einschalten passierte allerdings nichts, da ich Plus und Minus an den Relais vertauscht hatte. Trotz mehrfacher Kontrolle kann es bei selbst entwickelten PCBs schon einmal zu Fehlern kommen. Mit einer improvisierten Aufsatz-Polwende-Platine für die Relais, die Plus und Minus vertauscht, konnte ich das Problem aber schnell lösen. Im GitHub-Repository findet ihr eine korrigierte Version des Platinenlayouts.

Einzug ins Gehäuse

Da mein heimischer Maschinenpark meine Lust an Holzarbeiten widerspiegelt – es finden sich gerade so eine Handkreissäge, eine Stichsäge und eine Vibrationssäge im Schrank – war meine Idee für das Automatengehäuse eher quadratisch, praktisch, einfach. Leider war das weit entfernt von den Vorstellungen meines Sohnes. Er wünschte sich wabenförmige Fächer, an denen sich Plexiglastüren befinden. Wenigstens der Teil mit dem Münzeinwurf und der Technik durfte ein normaler Kasten bleiben. Da traf es sich gut, dass der Vater der Freundin meines Sohnes Schreinermeister ist. Das Gehäuse des Vendinators ist ihm wirklich gelungen (siehe Titelbild).

Weiter geht's

Kaum ist das Projekt (nach etwa einem Jahr) beendet und der erste Honig verkauft, fangen die ersten Träumereien an. In diversen Shops aus Fernost lief mir vor Kurzem ein Banknoten-Akzeptor über den Weg, der preislich zum Experimentieren einlädt. Vielleicht werde ich den Vendinator irgendwann noch damit erweitern. Auch eine Wechselgeld-Erweiterung ist schon als sehr grober Plan vorhanden. Mir schweben da zwei Münzstapel vor, in deren Mitte ein Servo mit einem Hebel Münzen aus dem jeweiligen Stapel schiebt. Wer weiß, vielleicht läuft schon bald wieder der 3D-Drucker warm. — akf



Bild 13: Mit Ablaufplänen behält man beim Programmieren komplexer Abläufe den Überblick.



Bild 14: Die fertig bestückte Platine sieht ziemlich professionell aus und war gar nicht so schwer zu konstruieren.

WIR TEILEN KEIN HALBWISSEN. WIR SCHAFFEN FACHWISSEN.





WEBINAR

Wärmepumpentechnik für Einsteiger

Wir erklären die Arbeitsweise der verschiedenen Typen und liefern Anhaltspunkte für eine erste Machbarkeitsabschätzung in der eigenen Immobilie.



workshop Einführung Gitlab

Erfahren Sie, wie Sie GitLab einrichten, konfigurieren und anpassen. Außerdem lernen Sie, wie Sie eine eigene Instanz der Entwicklungsplattform betreiben.



WORKSHOP CI/CD mit GitLab

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



WORKSHOP

Einstieg in Microsoft Project

Der Workshop vermittelt ein praxisnahes Modell für die professionelle Erstellung eines Projektplans in Microsoft Project.



WORKSHOP Einführung in den Kea DHCP Server

Erfahren Sie alles über Kea-DHCP-Software auf Unix- und Linux-Systemen. Sie lernen mehr über die Installation, Konfiguration und Betrieb des Systems.



WORKSHOP

Docker und Container in der Praxis

Der Workshop richtet sich an Entwickler die neu in das Thema einsteigen. Der Fokus liegt vor allem auf den Herausforderungen im Alltag.

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

433-MHz-Fernsteuerung im Eigenbau

Die Fernsteuerung in diesem Projekt ist günstig und vielseitig einsetzbar für Fahrzeuge, Roboter und Ähnliches. Durch die günstigen ATtiny-Chips ist man in der Programmierung nicht eingeschränkt und kann die Funktionen für seinen Anwendungsfall optimieren.

von Prof. Kai Schauer



us Gewohnheit greifen wir in vielen Projekten immer wieder auf die gleichen Boards oder Mikroprozessoren zurück, die wir kennen, die aber oft deutlich überdimensioniert sind. Die winzigen Mikroprozessoren, wie der ATtiny von Atmel, können eine ganze Menge, sind sehr preiswert und robust. Ideal, besonders wenn nur wenige Funktionen wie PWM, AD-Wandler oder digitale Pins benötigt werden.

Mit dem hier vorgestellten Projekt möchte ich so einen Einsatz am Beispiel einer Fernsteuerung demonstrieren. In Kombination mit 433-MHz-Funkmodulen und Kleinteilen ist so etwas für unter 25 Euro im Eigenbau möglich. Die Fernsteuerung ist für verschiedenste Projekte wie Autos, Boote, Hausautomation etc. verwendbar. Der hier vorgestellte Demonstrator, das Standmodell Optimist (Abbildung 1), erhält mit dem 433-MHz-Funkmodul und zwei Servos für Ruder und Segelwinde ein Upgrade zum fahrtüchtigen RC-Modellboot. Der Bootstyp ist Programm!

ATtiny85

Mein Projekt hat zum Ziel, einen kleinen Mikrocontroller (µC) mit möglichst vielen Funktionen auszulasten. Darum habe ich mit dem ATtiny85 gearbeitet. Er bietet den größten Speicher (SRAM und FLASH) seiner Serie und kostet nicht viel mehr als die Varianten mit kleinerem Speicher. Die Prozessoren der ATtiny-Produktfamilie (25, 45 und 85, siehe Tabelle 1), zu der der ATtiny85 gehört, sind so bezeichnet, dass zuerst die Größe des Flashs und dann die Anzahl der nutzbaren GPIO-Pins benannt ist. Konkret heißt das also: Der ATtiny85 hat 8 kByte Flash. Die Ziffer 5 bedeutet sechs GPIOs (PB0 bis PB5) plus zwei Pins für GND und VCC. Er hat also acht Beinchen.

ATtiny sind durch ihre RISC-Architektur sehr flott. Sie arbeiten alle Registerbefehle in einem Takt ab. Der Takt ist wiederum wählbar und von der Versorgungsspannung abhängig (siehe Tabelle 2). Das ist für den Stromverbrauch wichtig zu wissen. Kleine Spannung und kleiner Takt heißen auch kleiner Stromverbrauch.

Reichweite der 433-MHz-Module

Funkmodule mit 433 MHz sind weit verbreitet und sehr preiswert. Fast jedes Garagentor oder elektrische Fensteröffner arbeiten damit. Meist werden sie genutzt, um nur eine Null oder eine Eins zu versenden, etwa für "Auf" und "Zu". Die Reichweite der meisten Produkte ist gering, weil einfach nicht mehr erforderlich ist. In meinem Projekt brauche ich die maximale Reichweite. Der Optimist soll schließlich sicher einen ganzen See erkunden können. Nachfolgend habe ich alle Parameter zusammengetragen, an denen gedreht werden kann, um die Reichweite zu vergrößern. Ich komme mit meinen

Kurzinfo

» Fernsteuerung mit 433-MHz-Funkverbindung bauen » Sender und Empfänger mit ATtiny85

» Servo zu Segelwinde mit Endanschlägen umbauen





Kosten: 25 Euro Material für Sender und Empfänger

Werkzeug

» Lötutensilien: Lötkolben, Lötzinn, Schwamm

- » Maker-Werkzeug: Zangen,
- Schraubendreher » ATtiny-Programmer oder Arduino
 - Uno
- » Multimeter
- » 3D-Drucker

Material

- » Elektronische Bauteile It. Liste im GitHub » Modellsegelboot Optimist von Krick » 2 × Micro-Servos » 2 × Lipo-Akku 1000 mAh Mehr zum Thema » Florian Schäffer, AVR-Programme debuggen,
 - Teil 1, Make 4/24, S. 104 » Florian Schäffer, AVR-Programme debuggen,
 - Teil 2, Make 5/24, S. 44
 - Ben Eadie, OpenTX Funkfernsteuer-Firmware, Make 4/23, S.30



Features auf ca. 30 m im Freien, ohne Hindernisse. Es geht bestimmt noch mehr. Mir hat es für die Demonstrationszwecke damit genügt.

Sender

Unsere 433-MHz-Sendermodule brauchen eine $\lambda/4$ -Antenne. Die Formel für die Wellenlänge " λ " lautet: $\lambda = c/f/4$, wobei "c" die Lichtgeschwindigkeit in m/s ist und "f" unsere Fre-

quenz in MHz, also 433. Eingesetzt in die Formel, liefert das ca. 173 mm. Ich habe einen mit Lack isolierten Kupferdraht auf Länge zugeschnitten und ihn am Sendermodul angelötet. Soll auf mehr Reichweite optimiert werden, so finden Sie hilfreiche Informationen unter dem Kurzlink des Artikels.

In Versuchen mit Sender und Empfänger auf dem Breadboard (Steckplatine, Steckbrett) gab es oft Wackelkontakte, grundsätzlich hat



Abbildung 1: Standmodell Optimist, Basis: Baukasten Fa. Krick

Tabelle 1: ATtiny Varianten

ATtiny	25	45	85
Programmspeicher (Flash) [byte]	2048	4096	8192
Variablen/Laufzeitspeicher (SRAM) [byte]	128	256	512
Preis ca. [Euro]	2,00	1,00	1,80

Tabelle 2: ATtiny Spannungen und Frequenzen

Spannung [V]	Taktfrequenz [MHz]
1,8 5,5	0 4
2,7 5,5	0 10

der Sender aber funktioniert. Die Fernbedienung sollte möglichst klein und zuverlässig werden. Also habe ich eine kleine Platine für den Sender entworfen, gefertigt und schnell gelötet.

Beim Empfänger kam anfangs nichts vom Sender an und die verzweifelte Fehlersuche blieb ergebnislos. Also wieder zurück auf das Breadboard und die Platine neu designt. Das gleiche Ergebnis. Ein paar Tage später habe ich einem Bekannten davon berichtet. Er sagte, ich solle es mit mehr Masse versuchen. Die Antenne in Dipol-Auslegung brauche nicht nur die Antenne, sondern auch ordentlich GND-Fläche. Daraufhin habe ich Bleche in meiner Schachtel verklebt und mit dem GND-Potenzial der Schaltung verbunden. Die Funkverbindung hat dann funktioniert und letztlich bin ich wieder bei der Lochrasterplatine für den Sender (Abbildung 2) geblieben.

Die FS1000A 433-MHz-RF-Transmitter sind sehr robust. Sie haben auch den einen oder anderen Kurzschluss schadlos überstanden. Praktischerweise können sie mit 3,5 bis 12 Volt versorgt werden. Höhere Spannung bedingt auch höhere Reichweite. Darum genügen bei kurzen Distanzen sogar Knopfzellen für den Betrieb. Soll die Reichweite höher sein, ist die maximal mögliche Spannung zu wählen.

Empfänger

Auf der Empfängerseite bin ich bei meinen Recherchen auf einen Beitrag von Wolfgang Ewald gestoßen (siehe Kurzlink). Er hat das Empfängermodul RXB8 als für gut befunden. Ich kann seine guten Erfahrungen bestätigen.

Oft wird behauptet, dass die Antenne auf der Empfängerseite keine Rolle spielt. Das will ich aber nicht so einfach stehen lassen. Es mag sein, dass es für manche Kurzstreckenprojekte genügt, einfach die Empfängermodule ohne zusätzliche Antenne zu verwenden. Ich habe das Modul RXB8 mit einer Helix-Antenne zum Auflöten (Helix-Ø 4,2 mm, Draht-Ø 0,6 mm) ausgestattet. Damit konnte ich die Reproduzierbarkeit und Stabilität der Verbindung deutlich verbessern. Ich habe die Antenne nicht einfach angelötet, das RXB8-Modul sollte ia austauschbar bleiben und die Antenne senkrecht aus dem Mastbrett im Bug ragen. Also erhielt die Platine einen abgewinkelten Pinheader. Die Antenne kann dann von oben eingesetzt und wieder entfernt werden.

Servo als Segelwinde

Der Schwenkwinkel handelsüblicher Modellbau-Servos von 60° genügt selbst mit langem Hebel nicht, um ein Segel zu verstellen. Spezielle Segelwinden-Servos sind für den "Optimisten" zu groß. Daher habe ich den gleichen Servo-Typ wie für das Ruder für die Segelwin-



Abbildung 2: Innenansicht des Senders



de zum Endlos-Servo umgebaut – ein altbekannter Hack bei Makern. Dazu musste ich ihn aufschrauben und das 5-k Ω -Poti ausbauen. Es wurde durch zwei Festwiderstände von je 2,4 k Ω in Reihe ersetzt, der Mittelpunkt dieses Spannungsteilers ist mit dem Kabel zum ursprünglichen Schleifer verbunden und die Enden mit den Anschlüssen für die Endpunkte der Poti-Widerstandsbahn.

Der Weg, mit dem die Schot den Großbaum bewegt, kann jeden Punkt zwischen ",ganz dicht" und "maximal gefiert" annehmen. Sind diese Endlagen aus der jeweiligen Richtung kommend erreicht, soll die Bewegung stoppen, bis ein Befehl für die entgegengesetzte Richtung ankommt. Wir haben also drei Zustände: Endlage 1, irgendetwas dazwischen und Endlage 2. Mit einem einfachen Spannungsteiler (beliebige Stufungen sind natürlich auch möglich) am Analog-Digital-Wandler (ADC) werden daraus etwa die Werte 0, 512 und 1023. Die Werte schwanken allerdings etwas, was dann in der Programmierung Berücksichtigung finden muss. Für die Endlagenschalter habe ich Mikrotaster mit Schaltarm so modifiziert (Abbildung 3 - Mitte), dass diese in den Seilumlauf eingreifen. Eine auf das Seil geklemmte Kugel löst den Schalter dann aus.

ADC3-Probleme

Die Steuerung des Ruders (Abbildung 4) soll im Sender stufenlos per Poti erfolgen: Mittelstellung, Backbord, Steuerbord. Der ATtiny85 hat vier Analog-Digital-Wandler (ADC0 bis ADC3). ADC2 ist ja bereits für die Schot vergeben. Darum habe ich ADC3 gewählt. Beim Funktionstest erschien mir auf der Empfängerseite merkwürdig, dass der Servo nur einen geringen Ausschlag von ca. 40° absolvierte. Es sollten doch ±60°, also 120° sein.

Beim Auslesen der Werte im seriellen Monitor endete der Wert für das Ruder schon bei halber Umdrehung des Potis bei 512 und der ATtiny85 reagierte nicht mehr. Es sollten aber 1023 sein und der Eingang war korrekt im Bereich von 0 bis 5V. Nach etlichen Versuchen musste ich feststellen, dass es einen Konflikt mit der "ManchesterRF"-Bibliothek gibt, die ja sonst für das Funken tadellos funktionierte.

Gelöst habe ich das pragmatisch mit einem Vorwiderstand zum Poti, der den Arbeitspunkt verschiebt. Als Vorwiderstand habe ich 62 k Ω durch Ausprobieren gewählt, damit der ATtiny85 ganz sicher nicht crasht. Nun sende ich den Wert vom ADC direkt an den Empfänger.

Schaltkreise für Sender und Empfänger

Alle Teilstromkreise (ATtiny, Potis, Funk, Servos und Endlagenschalter) wurden alleinstehend getestet, bis sie zusammen mit dem ATtiny85 liefen. Dann wurden sie, getrennt für Sender und Empfänger, in einer Schaltung zusammengefasst. Ich würde beim Nachbau auch immer empfehlen, so vorzugehen. Start mit dem ATtiny, dann erstes Poti dazu, testen, dann zweites Poti dazu, testen, dann FS1000A dazu, testen usw. Das ständige Testen ist zwar anstrengend, zahlt sich aber später aus. Wenn alles mit einem Mal zusammengesteckt wird und die Schaltung nicht funktioniert, vergehen manchmal Stunden der Suche.

Der Sender (Abbildung 5) hat einen geringen Stromverbrauch. Zwei Potis, ein μ C, ein Funkmodul und eine LED mit Vorwiderstand sind keine nennenswerten Verbraucher. Der Spannungsregler 7805 reduziert die Eingangsspannung auf 5 V. Daher ist die Energiequelle des Senders eine praktische und kleine 9-V-Blockbatterie. Bisher habe ich diese noch nicht tauschen müssen.

Bis die Schaltung des Empfängers reibungslos lief, hat es länger gedauert. Der ATtiny85 hat fünf benutzbare GPIOs. Diese können als Analogeingänge, PWM-Ausgänge und Interrupt-Pin, der aber auch gleichzeitig für den seriellen Monitor reserviert bleiben sollte, konfiguriert werden. Eigentlich ist alles, was vonnöten ist, vorhanden. Doch jetzt kommt das "Aber". Nicht jede beliebige Kombination der Pins für die geplanten Funktionen harmoniert miteinander. Der in Abbildung 6 vorgestellte Schaltplan des Empfängers ist das funktionale Ergebnis meiner Versuche mit verschiedenen Nutzungen der Pins für ADC und PWM. Schließlich blieb nur D5 frei als Reset-Pin (RST).

Am Ende galt es noch eine Herausforderung zu meistern. Der ATtiny setzte sich zurück, wenn einer der Servos bewegt wurde. Servos haben einen hohen Anlaufstrom. Alle Versuche mit Stützkondensatoren verliefen im Sande. Als einzige stabile Lösung funktionierte eine zweite, eigens für die Servos genutzte Spannungsquelle. Alle anderen Elemente und der Prozessor kommen mit sehr wenig Strom aus und werden von der ersten Spannungsquelle ausreichend versorgt.



Abbildung 4: Schaltung der Potis für Ruder und Schot



Abbildung 5: Breadbaord und Schaltung des Senders

Wenn die beiden LiPo-Akkus angeklemmt werden, ist auf die Reihenfolge zu achten. Zuerst muss der µC-Akku angeschlossen werden und dann der für die Servos. Andersherum machen die Servos, was sie wollen, und ackern unkontrolliert über die Endlagen hinaus. Nach so einer Aktion muss dann die Mechanik immer neu eingestellt werden.

Programmierung

Ich habe in der Arduino IDE programmiert. Nach einigen Fehlversuchen musste ich feststellen, dass die Programmierung meiner ATtiny leider nur mit einer älteren Version der Arduino IDE klappt. Die Version 1.8.19 läuft reibungslos und kann auch neben der neuen Version verwendet werden. Jetzt ist noch die Installation eines ATtiny-Bundles für die Unterstützung des ATtiny85 über den Board-Manager in der Arduino IDE zu erledigen (siehe Kurzinfo-Link).

ATtiny lassen sich nicht so komfortabel programmieren wie ein ATmega328P aus dem gleichen Hause Microchip (ehemals Atmel). Eine Möglichkeit ist es, einen Arduino Uno zu verwenden und die Programme via ISP-Verbindung auf den ATtiny zu flashen. Wer das nur ein- oder zweimal benötigt, dem kann diese Lösung genügen. Sie nimmt einigermaßen viel Platz auf dem Schreibtisch ein und ist anfällig gegen Wackelkontakte auf dem Breadbaord. Weil ich den ATtiny bei all den Versuchen häufig flashen musste, bewährte sich die Programmierung mit dem "Tiny AVR Programmer" von Sparkfun. Der ist schnell installiert, robust und schön klein. Er hat auch Buchsen unter den GPIOs, um ihn in die Testumgebung zu integrieren, ohne jedes Mal den ATtiny abziehen zu müssen.

In der Arduino-IDE habe ich die folgenden Einstellungen gewählt:

- Board: "ATtiny25/45/85"
- Prozessor: "ATtiny85"
- Clock: "Internal 8 MHz"
- Programmer: "USBtinyISP"

Doch aufgepasst beim Handling mit ATtiny und dem Programmer: Die kleinen Beinchen des ATtiny sind beim Aufstecken und Abziehen vom Programmer schnell verbogen. Ein Zahnstocher oder schmaler Spatel aus Holz schafft dabei Abhilfe, ebenso natürlich ein sogenannter "Chip Puller". Links und rechts vorsichtig gehebelt, lässt sich der kleine ATtiny leicht und ohne Schäden vom Programmer lösen. Bei den Bezeichnungen ist ebenfalls Obacht geboten. Die Nummer des Pins entspricht nicht dem Namen im Code. Im Programm ist ebenfalls darauf zu achten, wofür der Pin genutzt werden soll. Wird Pin 2 als digitaler IO genutzt, heißt er im Programm "3", als analoger IO-Pin heißt er "A3". Ich lege mir zu Projektbeginn ein eigenes Pinout an, um immer den Überblick zu behalten (siehe Abbildung 7).

Der klassische "Serial.print"-Befehl steht für den ATtiny aus Speicherplatzgründen nicht wie selbstverständlich zur Verfügung. Diese Funktion ist für das Debugging aber unerlässlich. Hier hilft die Bibliothek "ATtinySerialOut". Die Einbindung und Nutzung erfolgt wie üblich und ist exemplarisch dem Programmcode des Empfängers zu entnehmen. Als Sende-Pin (TX) ist PB2/D2/Pin7 zu verwenden, gesendet wird nichts, das spart einen RX-Pin.

Bei der Verwendung von Arduinos oder den ATtiny und 433-MHz-Funkmodulen werden gerne Bibliotheken wie "RCSwitch" oder "VirtualWire" verwendet. In meiner Anwendung habe ich mit einer Alternative hierzu sehr gute Erfahrungen gesammelt. Für mich ist die Manchester-Bibliothek die erste Wahl. Sie ist speziell für den ATtiny entwickelt. Meine drei zu versendenden Variablen nehmen maximal Werte bis 1023 an. Diese in einem Array vom Sender abgelegten Werte lassen sich via Manchester-Lib in einer sehr einfachen Zählschleife an den Empfänger senden.

Programmablauf Sender

Der Sender hat die Aufgabe, drei Werte zu versenden. Der erste Wert ist eine frei wählbare



Abbildung 6: Breadbaord und Schaltung des Empfängers

Zahl, die ich als Identifikation für den Empfänger verwende. Nur wenn die Zahl beim Empfänger ankommt, dürfen auch die danach folgenden zwei Werte weiterverarbeitet werden. Das ist eine sehr einfache Lösung zur Abgrenzung von anderen Geräten mit 433-MHz-Funk. Möglich wären auch Prüfsummen wie CRC oder Ähnliches. Die nächsten beiden Werte wurden mit den Potis generiert und stellen Ruder und Schot ein.

Das finale Programm braucht in der aktuellen Version nur 21 Zeilen Code (netto):

- Manchester-Lib aufrufen
- Variablen definieren
- den Kanal zum Senden öffnen
- die drei Werte (ID, Ruder, Schot) im Array ablegen
- Versenden der drei Werte in einer Zählschleife

Programmablauf Empfänger

Der Empfänger, also unsere Steuerung "an Bord", hat sehr viel mehr zu tun. Optimiert

habe ich die Komplexität durch die direkte Nutzung von PWM für die Servos. Schließlich hat der ATtiny genügend PWM-Ports. Im Netz existieren auch Lösungen mit bis zu vier PWM-Ausgängen am ATtiny. Das bedarf dann aber eines vertieften Wissens zur Ansprache der entsprechenden Register und Timer im Prozessor.

Das Programm für den Empfänger ist dem des Senders sehr ähnlich. Zu Beginn die Präambel, allerdings mit Nutzung der ATtiny-SerialOut-Lib, die zum Einstellen und Debug-

TECHNIKUNTERRICHT MACHT ENDLICH SPAB!



Make: Education

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



Jetzt kostenlos downloaden: make-magazin.de/education

Projekt

ATTiny 25/45/85



gen nötig ist. Dann folgt die Öffnung des Kanals für die Kommunikation und danach eine Schleife für den Empfang von Werten. In dieser Schleife steckt wiederum eine weitere, die die Validität der empfangenen Werte prüft und diese, wenn alles in Ordnung ist, dann zur Weiterverarbeitung freigibt. Der aufwendigste Teil des Programms steckt nun in der Weiterverarbeitung der Werte.

Die Schot lässt sich dichtnehmen oder fieren. Aus Platzgründen habe ich die Endlagenschalter der Seilwinde auf der Rückseite, also der gegenläufigen Seite des Endlosseils angeordnet. Das Prozedere ist ja schon weiter oben beschrieben worden. Die Logik der Bewegungen mündet im Programm des Empfängers in der Definition von Handlungsszenarien mittels "if, else if"-Abfragen, die mit einem Analog-Pin (A3, Anschlag-Wert im Code) am ATiny die Stellung von zwei Schaltern abfragen.

- Wenn die Variable Anschlag im Bereich 500
 ... 520 ist, wird Limit BUG ausgelöst, Schot:
 180 ... 189 ist noch erlaubt.
- Wenn Anschlag = 0, wurde Limit HECK ausgelöst, Schot: 193 ... 200 ist noch erlaubt.
- Wenn Anschlag = 1023, Normalbetrieb,
 Schot: 180 ... 200 (gesamter Weg) ist erlaubt.



Abbildung 8: Gehäuse des Senders aus einer alten Handy-Verpackung

Dieser Programmabschnitt und die verwendeten Werte sind konkret für meine Anwendung optimiert. Wer den gleichen Aufbau, aber andere Komponenten verwendet, muss hier Anpassungen vornehmen. Dafür sind die Nutzung der ATtinySerialOut-Lib und des seriellen Monitors unabdingbar.

Würde der umgebaute Servo für die Segelwinde ohne Geschwindigkeitsbeschränkung laufen, könnte der Antrieb Schaden nehmen. Das Endlosseil wird in ersten Tests ohne Öse und Endlagen-Anschläge betrieben, um die Ruhestellung zu finden, bei der der Servo weder nach rechts noch nach links dreht. Das ist leider nicht automatisch 127 als Hälfte von 254, dem maximalen PWM-Wert. Gleiches gilt für die Bestimmung der maximalen Geschwindigkeit, mit der die Schot dichtgenommen oder gefiert wird. Das muss jeder für sich herausfinden. Alternativ ginge das Trimmen mit zusätzlichen Potis am Sender, dieser hat noch unbenutzte Pin-Reserven.

Gehäuse

Für den Prototyp des Senders konnte ich das Gehäuse eines alten Radiobausatzes nutzen. Als dann die neue Schaltung fertig war, war nicht mehr so viel Raum vonnöten. Kurz umgeschaut, fand ich den Karton eines Handys (Abbildung 8) aus solider Pappe. Trotz weniger Bedienelemente muss der Sender sicher in der Hand liegen. Für die Antenne kam noch eine Holzleiste hinzu und für das Auge ein Label aus dem Laserdrucker auf die Vorderseite. Fertig!

Segelwinde

Die Segelwinde (Abbildung 9) entpuppte sich als die am schwersten zu knackende Nuss. Es gibt zwei maximale Stellungen, ganz dicht und maximal gefiert. Bei gefülltem Segel auf einem Kurs wäre es einfach, die Schot (das Seil) aufzuwickeln. In der Wende schlägt und flattert sie aber, die Schotspannung lässt nach. So ist ein sicheres Aufwickeln nicht möglich. Das Boot wird manövrierunfähig, wenn sich das Seil verheddert. Außerdem sollte der Antrieb nicht immerzu motorisch gegen die Belastung des Segels arbeiten. Der Antrieb muss also selbsthemmend sein. Das heißt, ist die gewünschte Segelstellung erreicht, schaltet der Motor ab und es fließt kein Strom mehr.

Für diese Aufgaben sind Schneckengetriebe prädestiniert. Fündig wurde ich in einem alten Kassettenrekorder. Die Maße für die Achsabstände von Schnecke zu Treibrad konnte ich leicht am Gehäuse abmessen und auf meine Konstruktion übertragen.

Die meisten Teile wurden im 3D-Drucker aus ABS hergestellt: zwei Rollen, ein Gehäuse und das Endstück der Umlenkung. Die Mikroschalter mit langem Hebel sind mit 2K-Epoxid-Kleber angeklebt. Dabei ist auf große Klebeflächen zu achten, sonst hält der Epoxid-Kleber nicht sicher. Die Klebeflächen sind auch penibel mit z. B. Isopropanol zu entfetten. Der Kleber darf nicht in die Mechanik gelangen. Mir ist einmal der Hebel des Mikroschalters verklebt, welcher dann komplett ersetzt werden musste.

Bei der Planung des Einbaus der Segelwinde habe ich gar nicht erst darüber nachgedacht, sie in der offenen Jolle "unsichtbar" werden zu lassen oder sie irgendwie zu verstecken. Als Veranschaulichung eines mecha-



Abbildung 9: Prinzipskizze und Foto der Segelwinde



Abbildung 10: Prinzipskizze und Foto der Ruderanlage



Abbildung 11: Einbau des Funkmoduls im Bug des Modells

tronischen Systems ist die Transparenz vorteilhaft und für Interessierte gut sichtbar.

Ruderanlage

Die Ruderanlage ist ein Koppelstangengetriebe. Drei Längen sind variabel: "I1" Drehpunkt des Servos bis zum Ende des Ruderhorns, "I2" die Koppelstange selbst und "I3" Drehpunkt des Ruders bis zum Gelenk mit der Koppelstange (Abbildung 10). Jede Änderung einer Länge verändert die maximalen Ausschläge des Ruders. Die Gelenke sind zwingend einzubauen, weil das System sonst statisch überbestimmt wird und dann große Kräfte bis hin zur Beschädigung auftreten.

Mein Steuermann Pitti erhält durch die Gelenke einen richtigen Ellenbogen, der die Ruderbewegung natürlich aussehen lässt. Bei solchen Aufgaben arbeite ich gerne mit Drähten. Die Geometrien kann ich so mittels Verbiegen schnell verändern und anpassen.

Funkmodul

Das Funkmodul findet seinen Platz im Bug unter dem Mastbrett hinter der kleinen magnetisch gehaltenen Abdeckung (in Abbildung 11 abgenommen), um leichter an das Funkmodul zu kommen oder es austauschen zu können. Da es vorne auch nass werden kann, schützen zwei Lagen Isolierlack vorn und hinten auf dem Funkmodul dieses vor einem Kurzschluss.

Pittiplatsch

Der Pittiplatsch aus dem Kinderfernsehen hat das Casting zum Steuermann aus zwei Gründen gewonnen. In Optimisten lernen Kinder das Segeln. Es liegt darum nahe, einem Steuermann mit Bezug zu Kindern das Boot anzuvertrauen. Der zweite Grund ergibt sich aus seiner Form. Zwei Kugeln bieten genügend Bauraum (Abbildung 12) für den Ruder-Servo, die Platine und zwei LiPo-Akkus. Ich musste nur noch darauf achten, dass mein Pitti klein genug ist, damit der Großbaum in der Wende nicht an seinem Kopf hängen bleibt.

Für den Körper habe ich 50-mm-Kunststoffkugeln zum Zusammenstecken aus dem Bastlerladen verwendet. Die beiden mittleren Kugelhälften wurden an den Berührungsstellen so lange geschliffen, bis zwei, etwa 15 mm gleich große Bohrungen entstanden sind. Diese wurden um die Öffnungen miteinander verklebt, sodass da später die Kabel durchgehen konnten. Die Kugelhälfte, in der der Ruder-Servo platziert wird, wurde im Bootsrumpf verklebt. So entstand ein belastbarer Grundkörper. Vor dem Einbau wurden die drei Teile lackiert.

Jetzt brauchten die Kabel von der Seilwinde noch eine Verkleidung. Das war mit Beinen und Schuhen schnell gelöst. Die Schuhe kamen wieder aus dem 3D-Drucker und erhielten einen Anstrich. Als Beine habe ich Isoliergeflechtschläuche verwendet, die gerne in Lampen zum Schutz der Kabel verbaut werden.

Zum Glück ist heute im Handel die Auswahl an LiPo-Akkus in Maß und Kapazität riesig. Die beiden benutzten Akkus haben jeweils 3,7 V (eine Zelle) mit 800 mAh. Das ist allemal ausreichend und passt gut in eine Kugelhälfte. Die Step-up-Converter machen aus den 3,7 V dann 5 V für alle elektronischen Komponenten. Die Platine wurde in der oberen Kugelhälfte des Halbkugelverbundes mit Epoxi eingeklebt. Für den zweiten Akku habe ich Tape zum Fixieren verwendet. Es ist wichtig, die LiPos nach Gebrauch auszubauen, aufzuladen und in einem feuerfesten Koffer zu lagern. Gelegentlich kommt es leider vor, dass sich LiPos von selbst entzünden. Dem ist vorzubeugen! -caw



Ruderservo und erster Akku

Leitungen verlegt

Platine und zweiter Akku

Abbildung 12: Pittiplatsch mit Elektronik, Servo und Akkus



13./14. November 2024 Mannheim



Die Konferenz für Developer Experience, Platform Engineering und mehr

Die CLC setzt 2024 rund um das Container-Ökosystem Themenschwerpunkte zu KI-gestütztem DevOps, Security und FinOps sowie Nachhaltigkeit.

Highlights aus dem Programm:

- Pipeline als Produkt denken: Modularisierung, Versionierung, Testen, ...
- Der KI-gestützte Entwickler Tools, Datenschutz, Mindset
- Sicherheitsrisiken von CI/CD-Systemen erkennen und vermeiden
- Nachhaltigkeit in der Cloud Herausforderungen meistern
- Praxisbericht: Ressourcen reduzieren und Cloud-Kosten senken

Jetzt Frühbucher-tickets sichern!

Workshops am 12. November

Veranstalter

Gold-Sponsoren



continuouslifecycle.de

Silber-Sponsoren









KI erobert die Maker Faire

Größer, bunter und vielfältiger denn je: Mit über 250 Ständen und 15.800 Besuchern hat die zehnte Maker Faire in Hannover neue Maßstäbe gesetzt. Zwischen den üblichen Robotern, Cosplayern und 3D-Druckern war eine Sache eindeutig zu erkennen: Künstliche Intelligenz spielt eine immer größere Rolle auf der Maker Faire. Aus vielen DIY-Projekten ist sie nicht mehr wegzudenken.

von Marcus Hansson

u 99 Prozent happy. So wird mein Gesichtsausdruck vom "Emotionizer" bewertet - einer interaktiven Installation in Form einer KI-gesteuerten Kamera, die darauf trainiert ist, Gefühle zu erkennen. Wie mein Gesicht von der KI interpretiert wird, kann ich live auf einem Monitor nachverfolgen (Bild 1). Je nach festgestellter Gemütslage toben sich zwei tanzende "Skydancer" (mit den Namen Sad und Happy) aus (Bild 2). Gebaut wurde der "Emotionizer" von Mitgliedern des Maker Space Experimenta aus Heilbronn. Vom Coding über das Training des Modells bis hin zum Schaltplan haben sie alles selbst gemacht und sogar die beiden tanzenden "Skydancer" zusammengenäht. Und dies ist bei Weitem nicht



Bild 1: Mithilfe von KI sagt dir der "Emotionizer", wie du dich gerade fühlst.

das einzige Beispiel für die wachsende Bedeutung von KI auf der Maker Faire.

Traumwelten mit KI erzeugen

Zweites Beispiel: Mein Kollege Ákos Fodor lässt sich am Stand der HAWK (Hochschule für Angewandte Wissenschaft und Kunst, Hildesheim) in Echtzeit in eine interaktive, KI-generierte alternative Realität versetzen (Bild 3). Auch hier werden Standbesucher von einer Kamera gefilmt. Ein benachbarter Bildschirm zeigt die Aufnahmen, nachdem sie durch eine KI in Echtzeit verarbeitet worden sind. Unter der Motorhaube dieser Installation stecken ein gut bestückter Rechner und die Software StreamDiffusion. Letzteres kann als interaktiver Echtzeit-Bildgenerator beschrieben werden. Je nach Prompt werden die von der Kamera aufgenommenen Inhalte neu interpretiert. An den genauen Prompt, mit dem die KI ihn in eine Frau mit Blume in der Hand verwandelt hat, kann sich Ákos nicht erinnern, nur daran, dass es irgendwas mit den Stichworten "french baguettes and roses" war.

Immer mehr Hochschulen auf der Maker Faire

Dass die Maker Faire ein Ort ist, an dem Menschen auf entspannte Weise mit KI experimentieren können, haben auch viele Hochschulen verstanden und nutzten die Chance, mit jungen Technikinteressierten ins Gespräch zu kommen. Etwa 30 Hochschulen und wissenschaftliche Institute waren dieses Jahr vertreten.

Auch öffentlich finanzierte Think-Tanks präsentierten spannende KI-Anwendungen. So zum Beispiel die KI-Ideenwerkstatt für Umweltschutz (Slogan: "Künstliche Intelligenz für alle – gemeinwohlorientiert und umweltgerecht"), die in dem Projekt BirdNET Pi ein neuronales Netz mit den Aufnahmen von Tausenden Vogelrufen trainiert hat. Das KI-Modell kann einfach auf einen Raspberry Pi mit USB-Mikrofon in den heimischen Garten oder auf den Balkon gestellt werden, wo es dann automatisch rund um die Uhr feststellen kann, welche Vögel sich in der Nähe aufhalten.

Geisterjäger erstmals auf der Maker Faire

Doch auch wer sich nicht für KI interessiert, kam auf der Maker Faire auf seine Kosten. Viel Aufmerksamkeit bekamen zum Beispiel die Ghostbusters-Cosplayer aus Hamburg (Bild 4). Die hanseatischen Geisterjäger sind Teil einer überraschend großen weltweiten Community aus Ghostbusters-Afficionados und waren zum ersten Mal auf der Maker Faire. Ihre extrem originalgetreuen Nachbauten erregten großes Aufsehen.

"Eine wirklich gute Geisterfalle zu bauen, kostet viel Zeit", erzählt Ollie, einer von den Ghostbusters, mit dem ich sprechen konnte. Ein halbes Jahr lang hat er seine gesamte Freizeit investiert, damit die Falle genau so wie in dem Film aus dem Jahr 1984 aussieht. Warum

Mehr über die Maker Faire erfahren

- » Weitere Infos zu allen hier erwähnten Projekten und Ausstellern über den Link
- » Wo und wann die nächste Maker Faire stattfindet



das so aufwendig war, versteht sich anhand der Geschichte von dem kleinen roten Lichthütchen, das unter den Knöcheln in Bild 5 erkennbar ist.

"Dieses Lichthütchen hier ist ein Original, was genau so an der echten Geisterfalle dran ist. Ist gar nicht mehr zu bekommen, weil es eigentlich aus Piloten-Cockpits von Flugzeugen stammt. Für dieses Lichthütchen habe ich jemanden, der in New York Urlaub gemacht hat, gebeten, für mich zu diesem Elektroladen zu gehen und diese Lichthütchen zu kaufen. Denn der Laden versendet nicht international", erklärt Ollie.

Meine sofortige Nachfrage: Wäre dann 3D-Druck keine Alternative?

"Ich habe keinen 3D-Drucker zur Verfügung und kenne mich mit der Technik nicht aus. Ich



Bild 2: Erkennt die KI eine fröhliche Stimmung, tanzt der "Skydancer" noch schneller.



Bild 3: Make-Redakteur Ákos Fodor (unten links) bei der Verwandlung durch Kl.



Bild 4: Ollie zeigt mir geduldig den ganzen Vorgang: wie man die Geisterfalle mit dem Pedal öffnet (wie im Film), dann mit dem zweiten Pedal auslöst (auch wie im Film) und dann den Geist einfängt und in die Falle zieht.

Bild 5: Schmuckstück einer gut nachgebauten Geisterfalle: das rote Lichthütchen, wie im Original aus einem Flugzeugcockpit.



Auf der Maker Faire sind auch Schmiedekünstler gut vertreten.

bin mehr so ein klassischer Modellbauer, Bastler, mir war es zu viel, mich auch noch in das Thema einzuarbeiten. Also habe ich mit dem gearbeitet, was mir so an Werkzeug zur Verfügung steht."

Aber um die Geisterfalle mit LEDs und Sounds auszustatten, führte kein Weg am Programmieren vorbei. Also hat sich Ollie etwas Programmieren beigebracht, damit zwei Arduino Nanos und ein DF-Miniplayer die Geisterfalle mit Leben erfüllen können.

"Das macht eigentlich das Hobby Cosplay auch so interessant, weil man sich sehr vielseitig aller möglichen Techniken bedienen kann, um sein Ziel zu erreichen", sagt Ollie.



Bild 7: Stephan Ruprecht hat die Espressomaschine von Grund auf gebaut.



Bild 6: Kaffee nach Maker-Art. Das gelbe Kästchen ist eine Waage, damit genau die richtige Menge Kaffee erzeugt wird.



Kaffeemaschine nach Maker-Art

Mit ähnlich viel Aufwand, aber in eine völlig andere Richtung geht ein Projekt von Stephan Ruprecht, einem Mitglied der Gruppe, Wellenkino", die sich der Restauration von alten Oszilloskopen und anderen alten Messgeräten gewidmet hat: eine von Grund auf selbst gebaute Espressomaschine (Bild 6). Nur die Brühgruppe, also das massive Messingteil, das den Griff mit dem Siebträger hält, stammt aus einer alten Kaffeemaschine. Das Ergebnis: eine Kombination aus Messtechnik und moderner Regelungstechnik mit Mikrocontrollern in einem Oldschool-Look.

Meine Bewertung (allerdings ohne den Kaffee probiert zu haben): ein Meisterwerk. Durch Acrylglaswände können wir Pumpen und Kessel bei der Arbeit beobachten. Eine ganze Reihe alter analoger Instrumente vermitteln hier den Eindruck von wahnsinnig komplizierten Vorgängen. Man glaubt eher, ein Oszilloskop vor sich zu sehen als "nur" eine Espressomaschine. Hier wird nicht nur die Kesseltemperatur feinjustiert, eine Waage schaltet die Pumpe auch aus, sobald die genau richtige Menge Espresso in die Tasse gepresst worden ist.

Lot oder Leim hält uns zusammen

Für mich als neues Redaktionsmitglied war es übrigens das allererste Mal auf einer Maker Faire – eine ziemlich überwältigende Erfah-


Das Laserschwert dieser Lego-Figur hat umgerechnet ganze 1000 LEDs pro Meter. Gebaut hat es Martin von "makeTVee LED Projekte".

rung. Beim Schlendern durch die Hallen habe ich mir die Frage gestellt, was die ganze Maker-Community zusammenhält. Ob es einen gemeinsamen Nenner gibt, etwas, zu dem sich alle hier bekennen können? Ist es Leim oder doch Lötzinn - denn Löten verbindet, wie es am Stand von Blinkyparts hieß. Ich komme zu dem vorläufigen Schluss, dass es im Kern beim Maken darum geht, sich zum Herrn (oder zur Herrin) über die Technik zu machen, zu verstehen, wie etwas funktioniert, indem man es selbst baut. Und das in einer Welt, wo wir oft als stumpfe Konsumenten fertiger Produkte gesehen werden. Was kann schöner sein, als irgendwas in der Hand zu halten, das man selber erschaffen hat? Das ist für mich der rote Faden, der sich durch die Maker-Szene und die Maker Faire zieht - und übrigens auch über die Seiten dieses Magazins.

Wer dieses Jahr die Maker Faire in Hannover verpasst hat oder noch eine Maker Faire besuchen möchte, der hat noch zwei kleinere Veranstaltungen zur Auswahl: am 12. Oktober in Sindelfingen und am 9. November in Salzburg. Und am 23. und 24. August 2025 ist es wieder so weit: Deutschlands größte Maker Faire geht in die 11. Runde, wie in diesem Jahr im HCC in Hannover. Über den Link in der Kurzinfo gibt es mehr Informationen dazu. —mch



Bild 8: Die Make-Redaktion 2024 (von links): Rebecca Poweleit, Dunia Selman, Daniel Bachfeld, Louis Behrens, Daniel Schwabe, Johannes Börnsen, Ákos Fodor, Marcus Hansson und Carsten Wartmann



Zwei Welten treffen aufeinander.



Gordon Müllejans hat die Figur Grogu aus dem Star-Wars-Ableger "The Mandalorian" als animatronische Figur nachgebaut. Mit jeder Menge Servos unter der Silikonhaut wirkt der kleine Grogu hier fast noch lebendiger als in der Serie. Die Inspiration dazu war Daniel Springwald ("Basteln mit Daniel"), der auch selbst mit einer neuen Version seines Grogu auf der Maker Faire war.



Auch auf einer Maker Faire werden Verbrechen begangen. Zum Glück ist Batman, der dunkle Ritter, zur Stelle und ermittelt fleißig.

Innovators Kit – Make Edition

Mit unserem neuen Experimentierset steigt man superschnell und spielerisch in die Welt des Programmierens und der Elektronik ein. Wir zeigen, wie man mit den vorhandenen Beispielen im Handumdrehen einen eigenen Prototyp baut.

von Daniel Bachfeld



S chnell gemacht, einfach umgesetzt, einheitliche Hardware: Mit der Oxocard Connect, der Skriptsprache nanopy und einer vollwertigen, browserbasierten Programmierplattform (IDE) kann man leicht eigene Ideen ausprobieren, Prototypen für Projekte bauen oder einfach nur spielerisch Coden Iernen. Zusammen mit dem Innovators Kit macht man erste Schritte in die Welt der Elektronik und lernt, wie man die Bauteile mit einem Mikrocontroller kombiniert. In Make 2/24 haben wir die Oxocard Connect und die Oxocard Science zwar schon mal vorgestellt, hier zeigen wir aber im Folgenden ein konkretes Projekt.

Noch mal zur Erinnerung: Die Oxocard Connect ist im Grunde ein Minicomputer mit Display (240×240 Pixel), Joystick, USB-C-Anschluss, Netzwerkfunktionen und einem Erweiterungsport. Kern der Oxocard ist der populäre ESP32 mit 2 MByte PSRAM, 8 MByte Flash, WLAN und Bluetooth. Man programmiert sie mit einer an MicroPython angelehnten, superleicht erlernbaren Sprache.

Dank WLAN und Funktionen für HTTP und MQTT in nanopy kann man seine Projekte nicht nur lokal laufen lassen, sondern Daten versenden und abholen und so seine Oxocard etwa ins Smart Home oder seine IoT-Cloud integrieren.

Use

Keine Installation, kein Abtippen von Listings, keine Fehlersuche, stattdessen gleich mit den ersten Beispielen loslegen und mit dem Code herumexperimentieren. Use, Modify, Create: Dieser vom Hersteller Oxon hier umgesetzte didaktische Ansatz führt schnell zu ersten Erfolgserlebnissen und macht Lust auf mehr. Klar, Oxocard und nanopy ergeben gepaart ein Lernsystem. Aber keins, das sich als solches aufdrängt, sondern eher eins, das richtig Spaß macht und die Neugier unterstützt.

In der Programmierumgebung ist zu jedem Beispiel ein Tutorial verfügbar, in dem die Besonderheiten des nanopy-Skripts erklärt werden. Die Beispiele bauen aufeinander auf. Daneben gibt es eine Befehlsreferenz, in der man auch suchen kann. Nanopy lehnt sich an die Befehle und Struktur von Micro-Python an, orientiert sich bei den grafischen Funktionen aber an der bei Anfängern beliebten Sprache Processing. Damit lassen sich Grafiken mit wenigen Instruktionen auf das Display zaubern. Im allereinfachsten Fall genügen die zwei Befehle

drawRectangle(100, 100, 20 , 20)
update()

um ein Rechteck an der Position x=100, y=100 und der Kantenlänge 20 auf der Oxocard anzuzeigen.

Modify

Die Oxocard kommt mit einem vorinstallierten Betriebssystem, das diverse Aufgaben erledigt. Unter anderem koppelt es die Card per WLAN an die browserbasierte IDE auf nanopy.io und interpretiert die Skripte. Das WLAN ersetzt das Programmieren per USB-Verbindung, wie man es vom Arduino kennt. Wer will, kann seine Oxocard aber dennoch mithilfe eines Kabels programmieren. Das funktioniert jedoch nur mit Chrome und Edge, nicht mit Firefox oder darauf beruhenden Browsern. Der Hersteller bietet den Editor zudem auf GitHub zum Download an, sodass man ihn auch lokal installieren kann. Allerdings fehlen dann Komfortfunktionen wie Referenz, Tutorials, Dokumentation usw.

Auf Knopfdruck schickt die IDE eines der vielen Beispielprogramme an die Oxocard, die sie lokal auf dem ESP32 startet. Die Oxocard sendet wiederum Daten zur Laufzeit des Skripts per WLAN an die IDE zurück, etwa die Ausgabe des print-Befehls, die man sich mit der Terminal-Funktion anzeigen lassen kann – praktisch bei der Fehlersuche in eigenen Programmen oder in modifizierten Beispielen.

In der IDE lassen sich Werte im Skript respektive Programm während der Laufzeit durch grafische Schieberegler modifizieren und die Auswirkungen auf der Oxocard beobachten, ohne dass man wie bei Arduino erst langwierig erneut kompilieren und hochladen muss.

Daneben gibt es eine Debugging-Funktion, um seine Skripte Zeile für Zeile ablaufen zu lassen und derweil zu beobachten, welche Werte die Variablen annehmen.

Create

Durch das Herumprobieren bekommt man schnell ein Gefühl für die verschiedenen Befehle und deren Arbeitsweise. Dabei helfen auch die sich selbst erklärenden Funktionsnamen. Die vielen Beispiele lassen sich prima kombinieren. Für eigene Skripte gibt es eine separate Sektion im Editor mit einer Übersicht und der Option, sie auf der Oxocard zu speichern. Anschließend kann man sie dort im Startmenü unter "Eigene Skripte" aufrufen.

Kurzinfo

» Einführung in die Programmierung der Oxocard Connect
 » Web-IDE und nanopy in der Praxis
 » IoT-Thermostat mit Alarm schnell umgesetzt

Checkliste Zeitaufwand: 1 Stunde Kosten: 55 Euro

Material

 » Oxocard Connect Innovators Kit (Make Edition)
 » USB-Power-Bank

Praktisch ist auch der integrierte Sprite-

Editor, mit dem man pixelgenau bis zu 24×24

Sprites zeichnen und per Klick als Array in sein

Programm überträgt. Zusammen mit dem

Joystick kann man so Spiele und grafische An-

Cartridges

Über den Erweiterungsport kann man die

Oxocard Connect mit zusätzlicher Hardware

verbinden, womit sich ein riesiges Potenzial

eröffnet. Neben Cartridges für Luftmessun-

gen, Sound oder Pixelmatrizes gibt es die

Breadboard-Cartridge für Elektronikexperi-

mente. Im Bundle mit rund 30 elektronischen

Bauteilen, Jumperkabeln und natürlich der

Oxocard bietet das Make-Magazin das als

"Make Edition – Innovators Kit" im heise-Shop

abhängiger Widerstand (NTC) für Temperaturmessungen, ein lichtabhängiger Widerstand

(LDR) zum Messen der Helligkeit sowie ein PIR

als Bewegungsmelder enthalten. Ein Piezo-

Buzzer, ein Servomotor und diverse LEDs zur

Signalisierung oder Bewegung sind ebenso dabei wie Taster und Widerstände. Damit lässt

Als Beispiel exerzieren wir hier den Aufbau

eines Thermostats nach: Es soll die aktuelle

Temperatur auf dem Display anzeigen, dazu

korrespondierend einen Servomotor stellen,

bei Werten über 30 Grad Celsius Alarm geben

Bauelement und MQTT ein Tutorial mit Code, Schaltplan und Aufbauvorschlag vorhanden

ist, ist es ein Leichtes, das Projekt sukzessive

Da in der Online-IDE für jedes elektronische

und die Temperatur per MQTT verschicken.

Im Kit sind unter anderem ein temperatur-

an (siehe Link in der Kurzinfo).

sich allerhand bauen.

aufzubauen.

wendungen realisieren.



» Peter König, Daniel Schwabe, Lerncomputer für die Hosentasche im Test, Make 3/24, S. 34



Aufbau

Zuerst bauen wir den Temperaturmesser mit dem NTC. Der temperaturabhängige Widerstand ergibt zusammen mit einem Widerstand von 2,2 k Ω einen Spannungsteiler, dessen Wert wir mit dem Analog-Digital-Konverter an Pin IN06 der Cartridge messen können.

Daraus lässt sich mit den für den NTC charakteristischen Werten auf den Widerstand und damit auf die Temperatur schließen. Sieht man von den dafür erforderlichen Umrechnungsformeln im Beispiel mal ab, so reduziert sich das Programm auf:

```
while true:
   clear()
   adcValue = readADC(IN06, 100)
   T = berechneTausA(adcValue)
   drawText(10,90, "T = " + T + "°C")
   update()
   delay(1000)
```

Der Ablauf erklärt sich fast von selbst: In einer Endlosschleife löscht das Programm mit



Die Oxocard Connect (Make Edition) erkennt man am Makey auf der Rückseite und auf dem Bootscreen.



In der Online-IDE kann man komfortabel die Beispiele ausprobieren, die Tutorials und Aufbauanleitungen studieren und Befehle in der Doku nachschlagen.

clear() das Display und liest per ADC den Spannungswert ein. So weit, so gut, der erste Parameter der Funktion ist der Name des Pin, aber wofür ist der zweite? Rechts in der IDE gibt man in der Suche readADC ein und erhält: "Liest den analogen Wert… mehrere Male (nSamples) und bildet einen Durchschnittswert." Okay, damit kann man offenbar stark rauschende respektive schwankende Messungen filtern.

Die Temperatur T berechnen wir mit der Dummy-Funktion berechnetTausA(), die hier

>	«	۲	ଟ୍ତ	>_	ш
Var	iablen:				
Glo	bale Variablen:				
÷	uri				
+	username				
+	password				
+	publishTopic				
					0
ade	Value				567
				0.456	923
Vn				2.843	077
Rn	tc		10	136.001	953
				297.85	379
				24.703	3796
AN	IGLE			1.777	222
dut	tyCycle			302.338	3776

Mit dem Debugger untersucht man sein Programm zur Laufzeit und prüft die Inhalte der Variablen. stellvertretend für die Umrechnungen steht. Mit drawText() schreiben wir den Wert T an die x-y-Koordinaten 10 und 90 und hängen die Einheit als String an. Erst der Befehl update() zeichnet jedoch wirklich auf das Display. Alle Texte, Figuren und Pixel der draw-Befehle werden nämlich erst mal intern in einen Speicher gezeichnet. Für eine Verzögerung von 1000 ms sorgt delay(), bevor die Schleife erneut durchlaufen wird. Mit dem Button "Code ausführen" schicken wir unser Programm an die Oxocard Connect und auf dem Display erscheinen die aktuellen Werte in °C.

Stellglied

Temperaturen zu messen, ist neben LEDblinken-lassen der Favorit aller Maker für erste Projekte, aber ein bisschen langweilig. Erweitert man es um einen Servo, kann man auf das Gemessene physisch reagieren, etwa indem der Hebel ein Ventil stellt oder die Lamellen einer Jalousie steuert.

Zum Steuern des Servos reichen im Prinzip drei Zeilen:

setPWMFrequency(50)

dutyCycle = map(A, 90, -90, 102, 512)
writePWM(IO01, dutyCycle)

Der erste Befehl setzt die PWM-Frequenz auf die für Analog-Servos üblichen 50 Hz. Mit der Variablen dutyCycle steuert man den Winkel des Servos. Hier kommt eine sogenannte map()-Funktion zum Einsatz. Sie berechnet für den gewollten Winkel A das richtige Pulsbreitenverhältnis. Dazu gibt man in der map()-Funktion die bekannten Werte als Parameter an: bei 90 Grad 102 und bei 90 Gxrad 512. Die Werte leiten sich aus der internen Frequenzerzeugung und dem mit 12 Bit einstellbaren Verhältnis ab. Zum Glück ist alles schon vorgegeben. Wer eine genauere Erklärung benötigt, schaut einfach in die schnell erreichbare Online-Doku.

Bleibt zuletzt noch, dem Pin IOO1 als Ausgabe-Pin das Ergebnis mit writePWM() zu übergeben. Und schon stellt sich der Servo ein. Um allerdings linear zur Temperatur den Servo zu bewegen, benötigt man einen zusätzlichen Aufruf von map() vor dem eben erklärten map():

A = map(T, 40, 10, -90, 90) dutyCycle = map(A, 90, -90, 102, 512)

Damit mappen wir die zu erwartenden Temperaturen von maximal 40 °C auf den einen Vollausschlag des Servos und minimal 10 °C auf den anderen. Bei 25 °C sollte der Servo ungefähr in Mittelstellung stehen.

Alarm

Praktisch wäre ein akustischer Alarm, wenn es zu warm wird, beispielsweise ab 30 °C. Dann wird Pin IO02 mit einem Pulsbreitenverhältnis von 50 Prozent angesteuert. An dem Pin hängt der Piezo-Buzzer:

if T > 30

writePWM(I002,4096/2)

else writePWM(IO02, 0)

Die PWM-Frequenz haben wir oben schon beim Servo mit 50 Hz festgelegt und leider kann man nur eine Frequenz pro Card einstel-



Bei der Verkabelung der Bauteile hilft der in den Tutorials angezeigte Schaltplan.



len. Sobald die Temperatur unter 30 °C fällt, verstummt der Buzzer.

Praktisch wäre, die Temperatur aus der Ferne kontrollieren zu können, etwa indem die Oxocard die Werte per MQTT verschickt. Sie wissen nicht, was MQTT ist und wie es funktioniert? Auch hier hilft die Dokumentation mit einer kurzen Einführung weiter. Dort findet sich auch ein Beispiel, aus dem wir uns bedienen können. Auf die wichtigsten Zeilen reduziert, kann man die Temperatur so im eigenen WLAN veröffentlichen:

```
uri = "mqtt://ip-address"
username = ""
password = ""
connectMQTT(uri, username, password)
publishMQTT("Temperatur", T)
```

Die zwei Befehle connectMQTT() und publishMQTT() genügen im Prinzip, um sich mit einem MQTT-Server zu verbinden und über das Topic Temperatur eigene Werte zu versenden.

Zum Schluss speichern wir das gesamte Skript noch auf dem EEPROM der Breadboard-Cartridge und konfigurieren den Autostart. Damit führt die Oxocard unser Programm aus, sobald die Cartridge eingesteckt wird, ohne dass man die IDE benötigt.

Fazit

Use, Modifiy, Create: Genau mit diesem Ansatz haben wir ein eigenes Projekt mit den Teilen aus dem Innovators Kit gebaut und auf Grundlage der vorhandenen Beispiele innerhalb kurzer Zeit zum Laufen gebracht.

Ende des Jahres bringen wir ein Make Special als Ergänzung zum "Make Edition – Innovators Kit" heraus, das Einsteiger näher in die Programmierung und die Elektronik einführt. Daneben planen wir, die Oxocard künftig in Anleitungen in der Make einzusetzen. Schicken Sie uns Ihre Ideen für Projekte und Cartridges. — dab

> So sieht das fertige Skript für das Thermostat aus, wir bieten es auf unserem GitHub-Repo zum Download an.

Anders als in den Bauvorschlägen verkabelt, dennoch funktional: ein einfaches, digitales Thermostat







Daniel Bachfeld Chefredakteur, dab@make-magazin.de

Die 10. Maker Faire war ein echter Meilenstein, insbesondere was die Bedeutung des Maker-Movements im Bildungsbereich angeht. Die Zahl der Schulen und Hochschulen als Aussteller unterstreicht deutlich, dass das Thema ernst genommen wird und viel Innovationskraft mitbringt. Viel Energie haben mir auch unsere treuen Leser und Autoren mit ihrem positiven Feedback in persönlichen Gesprächen gegeben.



Marcus Hansson Redakteur, mch@make-magazin.de

"Lööööten!", riefen meine achtjährige Tochter und ihre Freundin begeistert, als ich vorschlug, den Lötworkshop von Blinkyparts zu besuchen. Dieser Moment auf der Maker Faire war für mich etwas ganz Besonderes. Bei all den Möglichkeiten auf diesem buntverrückten Jahrmarkt freuten sich die Kleinen ausgerechnet aufs Löten. So etwas erwärmt ein altes Maker-Herz.

Was war dein Highlight auf der Maker Faire 2024 und warum?



Carsten Wartmann Redakteur, caw@make-magazin.de

Mein Highlight war die Oxocard Connect mit ihren Aufsteckkarten, die ihre eigenen Programme speichern und beim Einstecken ausführen. Zusammen mit dem browserbasierten Nanopy-Editor mit Debugger hatte ich schon lange nicht mehr so viel Spaß beim Programmieren. Mein absoluter Favorit ist die Synthesizer-Karte, eine ganze Band in winzigem Formfaktor.



Ákos Fodor Redakteur, akf@make-magazin.de

Am besten fand ich, mich mit den Menschen zu unterhalten. Und damit meine ich nicht nur die Aussteller, sondern auch die vielen Besucher, die natürlich erst mal neugierig auf die Maker Faire waren. In lockeren Gesprächen erzählten sie mir dann aber auch, mit welchen tollen Projekten sie sich selbst beschäftigen. Das hat richtig Spaß gemacht.



© Copyright by Maker Media GmbH

Make-Team



Daniel Schwabe Technical Writer, das@make-magazin.de

Mich beeindrucken auf der Maker Faire am meisten die künstlerischen Projekte. Mit Elektronik befasse ich mich selbst viel, aber die handwerklichen und kreativen Fähigkeiten der KünstlerInnen dort finde ich schon außergewöhnlich und sehr bewundernswert.



Dunia Selman Social-Media-Managerin, dus@make-magazin.de

Mein Highlight auf der Maker Faire waren die vielen Mitmachaktionen. Ich habe mich selbst an einem Pflanzenbiotop versucht und konnte als Pflanzenliebhaberin meine Sammlung erweitern. Aber es gab auch viele andere technische und kreative Aktivitäten. Da war wirklich für jeden etwas dabei!



Nicole Wesche Mediengestalterin, niwe@heise.de

Mir hat die Mischung der Projekte gefallen. Ich habe mir unser Ohr erklären lassen, fasziniert zugesehen, wie aus nur einem Faden ein Bild wurde und wie ein Roboterarm softwaregesteuert eine OP unterstützt. Mein ganz persönliches Highlight war aber, dass sich meine zwei Söhne zusammengesetzt haben, um ein Spiel zu programmieren – ganz ohne Streit!



Daniel Rohlfing Leiter Events und Sales, dnr@maker-faire.de

Die Crew. Aussteller-Feedbacks wie "Es gab immer Ansprechpartner" belegen, dass wir als Einheit funktionierten. Alle packten mit an und waren mit Leidenschaft dabei. Viele der Freiwilligen sind seit Jahren dabei. Der Teamspirit war erneut überragend. Sonntagabend gab es für alle Pizza. Es kam die Frage auf, ob wir in der Zusammensetzung auch noch das 20. Jubiläum rocken. Eine schöne Vorstellung!



Nele Krautberger Werkstudentin Maker Faire, nekr@maker-faire.de

Mein Highlight war der Wissenshub in diesem Jahr. Da ich dafür verantwortlich war, konnte ich viele neue und interessante Leute kennenlernen und herausfinden, was ihre Passionen sind und wofür sie sich begeistern können. Durch die Vorträge habe ich viel Neues lernen können.



Kristina Fischer Projektleitung Maker Faire, krfi@maker-faire.de

Mein Highlight war es definitiv, die ganzen Maker wiederzusehen. Viele von ihnen kenne ich bereits seit meiner ersten Maker Faire 2017 und es fühlt sich jedes Jahr wieder so an, als ob man sich gestern erst getroffen hat. Der Austausch und die Herzlichkeit sind das Schönste für mich an diesem Event.



Johannes Börnsen Redakteur und YouTube-Host, jom@make-magazin.de

Mein persönliches Highlight war unser Make-Magazin-Stand. Den haben wir komplett umgestaltet, viel selber gebaut, optimiert und getüftelt. Es hat sich gelohnt: Wir können die Projekte aus der Make jetzt zugänglicher und interaktiver präsentieren. Das hat zu vielen spannenden Gesprächen und weiteren Ideen für unseren 2025er-Stand geführt.



Louis Behrens Werkstudent Social Media, loub@make-magazin.de

Meine Highlights auf der Maker Faire Hannover 2024 waren unser Maskottchen Makey, alle Cosplayer und der R2 Builders Club, die durch ihre Auftritte auf der Messe allen Besuchern und Ausstellern immer ein Lächeln aufs Gesicht zaubern, zum Staunen einladen und immer offen für ein cooles Foto sind.

Make

Carling Co

AVR-Programme debuggen, Teil 2

Um Ressourcen zu schonen, optimiert ein Compiler gern weg, was überflüssig erscheint. Wenn das aber zu einem Fehler führt, muss man manchmal ziemlich tief graben, um ihn zu finden. Hier ist ein Debugger Gold wert, denn er erlaubt, in Echtzeit einen Blick auf den Assembler-Code Ihres Programms zu werfen.

von Florian Schäffer



n der vorherigen Make-Ausgabe haben Sie gelernt, wie das Debugging eines ATtiny-Chips mit dem MPLAB SNAP über die UPDI-Schnittstelle funktioniert. Dieses Mal suchen wir einen Fehler auf einem Arduino UNO – genauer auf dem ATmega328. Als Debugging-Schnittstelle kommt debugWIRE zum Einsatz, das genau wie UPDI nur eine Datenleitung benötigt.

Das Beispiel, das wir in diesem Artikel debuggen, sieht auf den ersten Blick trivial aus, hat es aber, was die Fehlersuche angeht, in sich und erfordert einen sehr tiefen Blick ins System und sogar in die Assembler-Programmierung. Aber keine Sorge: Auch das lässt sich verstehen; wir begleiten Sie Schritt für Schritt.

Im Quellcode, den Sie über das GitHub-Repository herunterladen können, finden Sie ausführliche Kommentare, die Ihnen beim Verständnis helfen werden, wenn Sie parallel dazu das Datenblatt des Mikrocontrollers studieren. Für das Debugging-Beispiel ist es aber nicht notwendig, dass sie alle Anweisungen im Detail verstehen.

Einfacher Aufbau

Die Aufgabe des Programms soll sein, Tastendrücke zu erkennen und nach dem fünften Druck eine LED einzuschalten. Damit das Hauptprogramm nicht permanent in einer

Kurzinfo

» ATmega328 auf Arduino UNO debuggen
 » ISP und debugWIRE mit dem MPLAB Snap verwenden
 » Fehler im Code mit Assembler finden

de-	Checkliste	Material
us, in ins m- ich itt.	 Zeitaufwand: 2 Stunden Kosten: 20 Euro (zzgl. Kosten aus dem vorherigen Artikel) 	 » Arduino UNO mit ATmega328 » Breadboard mit Zubehör » Taster » LED, rot » Widerstand, 220 Ω
id- Sie im Ilel tu- ber en	 Mehr zum Thema » Florian Schäffer, AVR-Programme debuggen, Teil 1, Make 4/24, S. 104 » Florian Schäffer, ATtiny statt Arduino, Make 2/24, S. 72 » Daniel Bachfeld, Speicherverbrauch in Mikrocontrollern, Make 3/24, S. 92 	Werkzeug » MPLAB Snap In-Circuit Debugger/ Programmer » Lötkolben » Seitenschneider, Cutter, Dremel o. Ä.
en- en las	Alles zum Artikel im Web unter make-magazin.de/xxyf	

Automatischer Reset bei Arduino

Der USB-Seriell-Wandler auf Arduino-Boards führt automatisch einen Reset des Mikrocontrollers durch, sobald der serielle Monitor in der Arduino IDE geöffnet wird oder die Programmübertragung startet. Dazu wird die DTR-Leitung (**D**ata **T**erminal **R**eady) genutzt, die über einen Kondensator (typisch sind 100 nF) mit dem Reset-Eingang des ATmega328 verbunden ist (Bild 1). Die DTR-Ausgänge der typischen USB-Seriell-Wandler besitzen eine invertierte Logik zur üblichen Arbeitsweise von RS232, was im Datenblatt durch das Hash-Symbol hinter dem Namen angedeutet wird.

Sobald der PC Daten senden will, wird RTS (**R**equest **To S**end) HIGH und dem folgt automatisch DTR. Durch die invertierte Logik wird der Ausgang am USB-Wandler auf LOW gezogen und löst damit einen Reset am AVR aus. Das LOW hält an und der Kondensator lädt auf die Betriebsspannung VCC, was ihn (für Gleichstrom) sperren lässt, sodass die Reset-Leitung automatisch frei wird und über den Pull-up-Widerstand mit VCC verbunden ist. Endet die Datenübertragung, wird DTR



wieder HIGH, der Kondensator entlädt sich und wird für den nächsten Einsatz wieder leitend.

Grundsätzlich ist dieser Reset-Automatismus für die Arduino IDE nicht notwendig. Auch ohne Kondensator bzw. die Verbindung bei Reset-EN ist es möglich, ein Programm zu übertragen. Dafür müssen Sie nur im entscheidenden Moment den Reset-Taster drücken, um den Bootloader zu starten: Sobald der Kompiliervorgang abgeschlossen ist und die Übertragung starten soll (Meldung von avrdude in der Ausgabe). So komfortabel wie mit dem automatischen Reset ist das allerdings nicht.



Bild 2: Die Verbindung an der rot markierten Lötbrücke muss geöffnet werden, damit der Debugger auf die Reset-Leitung zugreifen kann.



Bild 3: Modifizierter Arduino UNO mit aufgelötetem Pinheader und Jumperstecker. Etwas RTV-Silikon rund um den Pinheader könnte den Umbau zusätzlich stabilisieren.

Schleife den Eingang abfragen muss, an dem der Taster angeschlossen ist, sondern (theoretisch) auch imstande sein soll, andere Dinge zu erledigen, wird ein Interrupt ausgelöst, sobald der Taster den Eingang auf LOW zieht (fallende Flanke). In der Interrupt Service Routine (ISR) wird ein Zähler (cnt) um Eins erhöht. Das Hauptprogramm prüft dann, ob der Zähler größer oder gleich fünf ist (sicherheitshalber größer oder gleich, da der Taster prellen und der Wert dadurch fünf übersteigen kann, bevor er geprüft wurde). So ist zumindest der Plan.

Arduino modifizieren

Damit Sie den AVR-Chip auf dem Arduino UNO debuggen können, müssen Sie das Board zuerst ein klein wenig modifizieren, denn Arduino nutzt einen Trick in Form eines automatischen Resets, damit die Boards so bequem über USB geflasht werden können (siehe Kasten). Dieser stört allerdings beim Debuggen, weil debugWIRE die Reset-Leitung benötigt, und muss daher deaktiviert werden.



Bild 4: Auf Arduino-UNO-Boards mit CH340 als USB-Seriell-Wandler muss der rot markierte Kondensator ausgelötet werden.

Bei originalen Arduino UNO (oder hochwertigen Nachbauten) gibt es eine kleine Leiterbahnbrücke mit Lötkontakten, die einen Kondensator mit dem Reset-Pin des ATmega328 verbindet (Bild 2). Diese ist mit Reset-EN oder -On beschriftet. Trennen Sie die Verbindung zwischen den beiden Pads mit einem spitzen Messer, Dremel oder anderem feinen Werkzeug, dann ist der UNO fürs Debuggen vorbereitet.

Um später über die Arduino IDE ein neues Programm aufzuspielen, müssen Sie die beiden Lötpunkte wieder verbinden. Mit etwas Geschick lässt sich auf die Lötflächen auch ein zweipoliger Pinheader auflöten, sodass Sie mit einer Steckbrücke schnell den Betriebsmodus wechseln können (Bild 3).

Günstigere Klone nutzen in den meisten Fällen den billigen CH340-Chip als USB-Bridge und ein anderes Board-Design ohne die Lötbrücke. Bild 4 zeigt ein solches Board, auf dem der Kondensator markiert ist. Hier bleibt Ihnen kaum eine andere Wahl, als den Kondensator auszulöten, um den Debugger nutzen zu können.

DWEN-Fuse-Bit setzen

Die Reset-Leitung des Mikrocontrollers muss für den Einsatz des debugWIRE (der den Reset-Pin nutzt) einmal umkonfiguriert werden. Bei den AVR-Chips sind dafür die Fuse-Bits zuständig und Sie müssen das Bit DWEN (**d**ebug **W**IRE E**n**able) setzen. In der etwas kruden Logik von AVR bedeutet dies: Das Bit muss auf 0 gesetzt werden, in den "programmiert"-Zustand. Weil der debugWIRE bis dahin aber noch gar nicht funktioniert, lässt sich das Fuse-Bit nur per ISP programmieren.

Microchip Studio kann sich semi-automatisch um dieses Fuse-Bit kümmern, sodass keine Gefahr besteht, etwas falsch zu machen. Sie müssen lediglich alle Verbindungen für ISP-Zugriffe zwischen dem Snap und dem Arduino herstellen, was über die ISP-Stiftleiste am einfachsten geht (Bild 7), aber auch über die Buchsenleisten am Rand möglich ist, wenn keine ISP-Stiftleiste auf dem Arduino-Board vorhanden oder eingelötet ist.

Solange debugWIRE aktiviert ist, bleibt die Reset-Leitung für ihre ursprüngliche Aufgabe ohne Funktion – Sie können es später ausprobieren und den Reset-Taster auf dem UNO drücken: Es wird nichts passieren. Das bedeutet aber auch, dass Sie den ATmega nicht mehr über die Arduino IDE bespielen können, da ein Reset (manuell oder über den beschriebenen Trick mit dem Kondensator) notwendig wäre, um in den Bootloader zu gelangen.

Doch keine Sorge: Natürlich lässt sich das DWEN-Fuse-Bit auch wieder zurücksetzen (AVR-Logik: auf 1 nach "nicht programmiert"). Dadurch wird die debugWIRE-Funktion deaktiviert und Reset arbeitet wieder wie gewohnt. Dafür brauchen Sie aber zwingend einen Debugger wie den Snap, der auch debugWIRE unterstützt, denn das Zurückstellen geschieht nicht per ISP, sondern mit debugWIRE.

debugWIRE aktivieren

Verbinden Sie den MPLAB Snap mit Ihrem PC und starten Sie Microchip Studio. Öffnen Sie dort das Projekt uno1.atsIn. Installieren Sie das Paket ATmega_DFP mit einem Klick auf "Download and Install", wenn Sie von Microchip Studio dazu aufgefordert werden. Starten Sie die IDE neu und öffnen Sie noch einmal das Projekt. Legen Sie danach zwei Breakpoints in main.c an, wie in Bild 8 zu sehen. Klicken Sie danach in der Menüleiste auf "Project/uno1 Properties …" und prüfen Sie, ob auf der Registerkarte "Tool" der MPLAB Snap als Debugger

In-System-Programmierung

Die In-System-Programmierung (ISP) stellte bei der Einführung einen enormen Zeitund Komfortgewinn dar. Anstatt den externen Speicher (EPROM) mit dem Programmcode für jede Änderung am Code aus der Schaltung herauszunehmen (wofür es spezielle Null-Kraft-Sockel gibt - siehe Bild 5), ihn in ein Programmiergerät zu stecken und dann wieder zurück in das Zielsystem zu setzen, konnte nun der Controller direkt im System beschrieben werden. So lässt sich damit z. B. der Bootloader für die Arduino IDE auf einen fabrikneuen ATmega328 flashen, obwohl der Chip bereits auf dem Board verlötet ist. Nur über die ISP-Schnittstelle ist zudem das Ändern von Fuse-Bits möglich oder die vom Programm unabhängige Vorbelegung des SRAM.



Bild 6: Anordnung der Kontakte für die In-System-Programmierung (ISP) auf dem Arduino UNO

Über ISP kann man aber nicht debuggen. Beim ATmega328 verwendet man dafür debugWIRE und bei anderen AVRs wie dem großen ATmega644 wird JTAG benutzt. Allerdings belegt debugWIRE drei der Leitungen, die auch das ISP verwendet, sodass es manchmal etwas zu Verwirrung kommen kann, da kein Umstecken von Kabeln erforderlich ist, wenn man zwischen den beiden Methoden wechselt. Zudem wird ISP benötigt, um den Mikrocontroller für den ersten debugWIRE-Zugriff freizuschalten.

ISP nutzt das Serial Peripheral Interface (SPI), das auch teilweise für die Kommunikation mit externen Modulen (z. B. Displays) verwendet wird. Für die drei Signalleitungen (MISO, MOSI und SCK), Reset und die Anschlüsse für die Versorgungsspannung hat sich für ISP ein sechspoliger (2×3) Stecker etabliert (Bild 6), der auch auf dem UNO und anderen Arduino-Boards vorhanden ist.

Im Verhältnis zu moderner Fertigungstechnik ist der ISP-Kontakt ein ziemlicher Klotz und nimmt viel Platz weg. Dennoch ist er auf vielen Platinen von Multimedia-Geräten über Haushaltsgeräte bis zu Baustellen-Blitzleuchten vertreten. Oft allerdings nicht als Stiftleiste, sondern als einfache Platinenkontakte, wie sie im Artikel "ATtiny statt Arduino" in Bild 1 zu sehen sind. So kann der Hersteller in der Fertigung oder später sehr einfach Firmware auf den verbauten Mikrocontroller aufspielen.

Bild 5: Ein professionelles Programmiergerät für PROMs (Programmable Read-Only Memory) mit Null-Kraft-Sockeln verschiedener Größen aus den 1980ern.



Bild 7: Verbindung vom Snap zum UNO für ISP und debugWIRE und die Schaltung für die Beispielanwendung mit Interrupt



Bild 8: Zwei Haltepunkte werden im Code benötigt, um den Fehler zu finden.

eingestellt ist und bei Interface "debugWIRE" steht. Wie immer sollte ebenfalls "Erase only program area" ausgewählt sein. Öffnen Sie auf der Registerkarte "Toolchain" und im Baum "AVR/GNU C Compiler" den Punkt "Optimization". Für dieses Projekt soll der übliche Optimierungsgrad "Optimize for Size (-Os)" benutzt werden.

Falls noch nicht geschehen: Öffnen Sie die Verbindung für die Reset-Leitung auf dem Arduino-Board. Versorgen Sie den Arduino UNO über eine separate Spannungsquelle (USB, Hohlbuchse oder VIN-Pin), weil der Snap dies, wie bereits angesprochen, nicht macht.

Starten Sie das Programm ohne Debugging (nicht gefülltes grünes Dreieck). Da debug-WIRE noch nicht aktiviert ist und der Arduino UNO sich im ISP-Modus befindet, erscheint kurz nach dem Start eine Fehlermeldung, dass die Verbindung mittels debugWIRE nicht aufgebaut werden konnte. Sie werden gefragt, ob Sie das Fuse-Bit DWEN aktivieren wollen (Bild 9). Klicken Sie auf "Yes".

Kurz danach erscheint die Erfolgsmeldung. Es ist wichtig, dass Sie jetzt die Spannungsversorgung des Arduino kurz trennen und erst danach auf "OK" klicken (Bild 10). Sämtliche Verbindungen zum Debugger können Sie aber bestehen lassen. Die ISP-Verbindungen zwischen Arduino und Debugger werden nun nicht mehr benötigt, sondern nur noch die drei für debugWIRE (VCC, GND und RESET). Es schadet aber auch nichts, alle Verbindungen beizubehalten, solange Sie experimentieren. Verbinden Sie den Arduino UNO wieder mit der Spannungsversorgung.

Ein Problem, wo keins sein sollte

Es wird Sie nicht wundern, aber das Programm (siehe Listing, main.c") macht mal wieder nicht das, was es soll. Ganz gleich, wie oft Sie (frustriert) den Taster drücken: Die LED wird nicht leuchten. Zeit, den Fehler aufzuspüren. Starten Sie dazu die Programmausführung mit Debugger (gefülltes grünes Dreieck).

Das Programm wird bis zum Breakpoint bei if ausgeführt und bleibt dann stehen. Klicken Sie rechts an einer beliebigen Stelle auf die Variable cnt und wählen Sie "Add Watch" aus, um ihren Inhalt in einem Extra-Fenster angezeigt zu bekommen. Wählen Sie in der Menüleiste "Debug/Step Over" aus oder drücken Sie F10. Der Breakpoint wird jetzt übersprungen und das Programm läuft weiter. Da es sich in einer Endlosschleife mit while befindet, passiert nichts weiter, aber das Programm ist durchaus noch aktiv (in der Statusleiste steht "Running").

Drücken Sie jetzt einmal auf den Taster in der Schaltung. Dadurch wird der Pin mit dem Taster, der an den Interrupt gekoppelt ist, ausgelöst. Der Programmablauf springt danach in die ISR und bleibt dort am Breakpoint cnt++ stehen. Klicken Sie im Menü auf "Debug/Step Out" oder benutzen Sie das Tastenkürzel Umschalt+F11. Damit wird die Routine abgearbeitet und das Programm wartet wieder am zweiten Breakpoint. Im Fenster "Watch 1", das Sie über das Menü mit "Debug/Windows/ Watch/Watch 1" erreichen, sehen Sie, dass in der Variablen cnt der Wert 1 steht: Die ISR hat den Variableninhalt also wie gewünscht inkrementiert.

Wiederholen Sie die vorherigen Schritte mit dem Drücken der Taste usw., bis cnt bei 6 steht. Eigentlich sollte längst die LED leuchten, denn die if-Bedingung ist augenscheinlich erfüllt.

Vom Compiler wegoptimiert

Halten Sie den Debugger mit der Stopp-Taste in der Symbolleiste oben links an (rotes gefülltes Quadrat). Ergänzen Sie im Programm die Variablen-Deklaration von cnt mit dem Zusatz volatile zu

volatile uint8_t cnt = 0;

Starten Sie das Programm dann erneut, aber ohne den Debugger. Jetzt klappt es: Nach ein paar Tastendrücken leuchtet die LED. Haben Sie sich sattgesehen, löschen Sie das Schlüsselwort volatile noch einmal aus dem Code, damit Sie auf Fehlersuche gehen können.

Die Entwickler von Mikrocontrollern und Compilern sind ziemlich clever und wollen alles aus einem System rausholen, was möglich ist. Das bedeutet auch, dass auf jede noch so kleine Anweisung verzichtet wird, wenn sie vermieden werden kann. Zudem gibt es Einschränkungen aufgrund der Systemarchitektur, die man handhaben muss. Eine solche ist, dass der Mikrocontroller gar nicht direkt mit Variablen und auch nicht mit dem Speicher in seinem RAM arbeiten kann, in dem der Wert liegt. Um cnt zu ändern, muss der Wert erst aus diesem Speicher in ein Register kopiert werden. Dort wird er geändert und dann wird der Registerinhalt wieder zurückgeschrieben.

Ein Arbeitsschritt beim Kompilieren des Programms sieht vor, unnötige Speicherzugriffe zu eliminieren. Dieser Funktion fällt nun etwas zum Opfer, was nicht verloren gehen dürfte: In der ISR wird der Inhalt aus dem Speicher in ein Register übertragen, erhöht und wieder zurückgeschrieben. So weit, so richtig.

In der Hauptroutine main() muss etwas Ähnliches erfolgen: Der Speicherinhalt der Variable cnt wird in ein Register kopiert und der Registerinhalt dann mit dem Vergleichswert (hier: 5) verglichen. Die Optimierung beim Kompilieren denkt aber, dass sie sich den ersten Schritt auf Dauer sparen kann, weil sie nicht erkennt, dass sich der Speicherinhalt außerhalb der Hauptroutine ändern könnte.

Also holt der Prozessor nur beim ersten Mal den Wert aus dem Speicher (0, weil die Variable mit dem Wert initialisiert wurde) und schreibt ihn in ein Register. Anschließend

Launch Fa	ailed		×
?	Failed to launch debug session with debugV circuitry or disabled debugWIRE interface. N continuing. Do you want to use SPI to enabl	VIRE. This could be caused fake sure that the reset line the DWEN fuse?	by reset line is free before
		Yes	No

Bild 9: Noch ist der Arduino nicht für Zugriffe per debugWIRE freigeschaltet. Die IDE kann das aber ändern.



Bild 10: Trennen Sie die Spannungsversorgung des Arduino, wenn Sie diese Meldung sehen, und verbinden Sie ihn wieder.

Flash-Speicher und debugWIRE

Während sich bei UPDI ein Extra-System in der CPU um das Debugging und Breakpoints kümmert (siehe Erklärung im vorherigen Artikel), werden bei debugWIRE für jeden Breakpoint Maschinenbefehle im Flash-Speicher des Mikrocontrollers manipuliert.

Um einen Breakpoint einzurichten, schreibt der Debugger auf dem Mikrocontroller an der gewünschten Halteposition einen Op-Code (Operation Code) über den eigentlichen Maschinencode, den der Compiler aus Ihrem C-Programm erzeugte. Dieser Op-Code veranlasst den Prozessor dann, an dieser Stelle mit der Ausführung anzuhalten. Soll sie fortgesetzt werden, wird der Op-Code durch die ursprünglichen Bytes ausgetauscht und es geht weiter. Davon merken Sie als Entwickler nichts und insofern hört sich das auch nicht nachteilig an.

Allerdings wird bei jedem Abarbeiten eines Haltepunkts der Flash-Speicher an der entsprechenden Speicherstelle zweimal beschrieben. Da kommt beim Experimentieren einiges zusammen - erst recht in Schleifen. Bei den meisten AVR-Chips sind für den Flash-Speicher nur 10.000 Schreib- und Löschvorgänge garantiert. Wird eine Speicherzelle öfter geändert, kann es zu Fehlern kommen, die sich dann so gut wie gar nicht auffinden lassen. Sie werden es möglicherweise nicht schaffen, 10.000 Zyklen beim Debugging zu erreichen, aber behalten Sie es dennoch im Hinterkopf oder schauen Sie sich die Tipps von Microchip an, die wir in der Kurzinfo verlinkt haben.

main.c

```
#define F_CPU 1600000UL
```

```
#include <avr/io.h>
#include <avr/interrupt.h>
uint8_t cnt = 0;
```

```
ISR (INT0_vect){
    cnt++:
```

}

```
int main(void){
    DDRB |= (1 << PINB4);
    DDRD &= ~(1 << PIND2);
    PORTD |= (1 << PIND2);
    EICRA |= (1 << ISC01);
    EICRA &= ~(1 << ISC00);
    EIMSK |= (1 << INT0);
    sei();
    while (1){
        if (cnt >= 5)
            PORTB |= (1 << PINB4);
    }
    while(1);
    return(0);
}</pre>
```

heise academy

Für erfolgreiche IT-Teams von morgen

Weiterbildung als Erfolgsstrategie

Professionelle IT-Weiterbildung für Unternehmen – das bietet die heise academy. Als Tochter der heise group haben wir es uns zur Aufgabe gemacht, Unternehmen und ihre IT-Professionals mit digitaler Weiterbildung voranzubringen, Qualifikationslücken zu schließen und internes Lernen zu fördern.



Interesse geweckt? Hier mehr erfahren:

heise-academy.de/Fuer-erfolgreiche-IT-Teams-von-morgen



Bild 11: Im Assembler-Code ist der Haltepunkt markiert und drei Befehle sind für den Vergleich notwendig.

benutzt er immer wieder dieses Register, ohne dafür zu sorgen, dass der aktuelle Wert aus dem Speicher ins Register übertragen wird. Die MCU sieht also gar nicht die neuen Werte, die von der Interrupt-Routine im Speicher abgelegt wurden, sondern prüft immer nur, ob 0 gleich oder größer 5 ist – was sich nie erfüllt und weshalb die LED auch nie leuchtet.

Erst das nachträglich ergänzte Schlüsselwort volatile macht dem Compiler klar, dass er nicht ganz so sparsam sein soll und bei jeder Nutzung einer Variable die aktuellen Daten aus dem Speicher in das Register kopieren und mit diesen arbeiten soll. Und so klappt es mit dieser Ergänzung dann auch mit dem Vergleich, der nach fünf Tastendrücken erfüllt ist.

EIMSK |= (1 << INT0);

if (cnt >= 5)

CPI R24,0x05

RJMP PC-0x0005

00000068 LDS R24,0x0100

00000068 BRCS PC-0x03

0000006C SBI 0x05,4

sei(); // Global Interrupt Enable

PORTB |= (1 << PINB4);</pre>

Bild 12: Leicht zu übersehen, aber dieser Wert für die Sprungstelle ist

00000066 SBI 0x1D,0

00000067 SEI

0000006A

0000006D

entscheidend.

Der Assembler-Code verrät's

Haben Sie das volatile entfernt, starten Sie den Code mit Debugger, damit Sie die Erklärungen mitverfolgen können und eine neue Funktion beim Debugging kennenlernen. Wählen Sie nach dem Start den Menüeintrag

"Debug/Windows/Disassembly" aus, wodurch sich ein neues Fenster in der Tableiste anheftet. Dieses zeigt den vom Compiler erzeugten Assembler-Code, der anschließend beim Kompilieren (vom Codegenerator) in den Binärcode übersetzt und auf den Mikrocontroller

// INTØ zulassen

Load direct from data space

// PA1 LED HIGH (ein)

Set bit in I/O register

Compare with immediate

Set bit in I/O register

Relative jump

Branch if carry set

Global Interrupt Enable

übertragen wird. Das Programm wird bis zum Breakpoint ausgeführt. Im Disassembly-Fenster sehen Sie, wo der aktuelle Haltepunkt liegt (Bild 11). Etwas weiter darüber steht der C-Befehl (als Kommentar zur Orientierung) und anschließend sehen

Arduino-Bootloader wiederherstellen

Falls Sie jemals beim Experimentieren den Bootloader des Arduino zerschießen, sodass er nicht mehr mit der Arduino IDE

kommunizieren kann, lässt sich das schnell mit dem SNAP beheben. Sie benötigen lediglich die HEX-Datei, die den Binärcode

MPLAB® Snap (BUR	204481828) - Device Pro	gramming			?	×
Tool MPLAB® Snap ~	Device ATmega328P	Interface	Device signature	Target Voltage		
Interface settings Tool information Device information Oscillator calibratio Memories Fuses Lock bits Production file	Device Erase Chip Flash (32 KB Z\optibool Flash (32 KB Z\optibool Frase de Verify FL Advance EEPROM (1	Erase now Erase no Erase no Era	ng F	Program Verify Program Verify	v Read Read	
rasing device OK rogramming Flash erifying Flash	ок					
 Verifying Flas 	h					E

Bild 13: In Microchip Studio können Sie den Bootloader eines Arduino schnell wieder herstellen.

des Bootloaders enthält. Diese finden Sie im Git von Arduino (siehe Link in der Kurzinfo). Laden Sie sich dort die zu Ihrem Board passende Datei herunter. Für den ATmega328 verwendet Arduino derzeit Optiboot (optiboot_atmega328.hex).

Der Arduino muss sich zum Flashen des Bootloaders im ISP-Modus befinden und mit einer externen Spannungsversorgung verbunden sein. Schließen Sie den SNAP mit den benötigten Verbindungen am Arduino an. Wählen sie in Microchip Studio im Menü "Tools/Device Programming" aus. Stellen Sie dann in dem erscheinenden Fenster als Tool den Snap und als Device den Mikrocontroller ein und klicken Sie auf "Apply". Geben Sie anschließend auf der Registrierkarte "Memories" bei "Flash" den Pfad zur HEX-Datei des Bootloaders an (Bild 13).

Mit einem Klick auf "Program" starten Sie den Brennvorgang und nach wenigen Sekunden ist alles erledigt. Danach können Sie das Fenster schließen, alle Verbindungen zum Snap trennen und den Arduino wie gewohnt mit der Arduino IDE nutzen.

Workshop

Sie die Speicheradressen, wie sie im Flash belegt werden und die Assembler-Befehle, mit denen ihre C-Anweisung umgesetzt wird. Die Kommentare dahinter helfen beim Verständnis.

Beachten Sie, dass einige Anweisungen zwei (oder mehr) Adressen belegen und die Speicherplatzangaben deswegen nicht fortlaufend sind: 0x0068 und 0x0069 werden vom LDS-Befehl belegt, sodass der nächste Befehl erst bei 0x006A abgelegt wird.

Auch wenn Sie kein Assembler beherrschen, sind die Anweisungen gar nicht so besonders unverständlich:

- LDS (Load Direct from SRAM) kopiert den Inhalt der Speicherzelle 0x0100 in das Register R24. In diesem Beispiel nutzen wir diese Speicherzelle für die Variable cnt. Das hat der Compiler so festgelegt, es könnte aber auch eine andere Speicherposition sein.
- CPI (Compare Immediately) vergleicht den Inhalt von R24 mit dem (konstanten) Wert 5 (0x05). Dazu wird intern vom Wert in R24 der Wert 5 subtrahiert. Das Ergebnis wird verworfen, aber gab es bei der Berechnung einen Unterlauf (negatives Ergebnis), wird automatisch das Carry-Flag (C-Flag) im Statusregister SREG (einem internen Register für Berechnungen etc.) gesetzt. Das passiert so lange, bis der Wert in R24 größer gleich 5 ist.
- BRCS (Branch if Carry is Set) führt einen Sprung (engl.: branch) aus, wenn das C-Flag bei der vorherigen Operation gesetzt wurde. Das Ziel des Sprungs lautet: PC-0x01.

Bei AVRs zeigt der Befehlszähler (PC: Program Counter) auf die Speicheradresse des derzeitigen Befehls. Die BRCS-Anweisung bedeutet also, dass der nächste Befehl an der Adresse steht, die eins kleiner ist als die aktuelle im Program Counter abgelegte. Derzeit steht der PC bei 0x006B und es wird zu 0x006A (zurück-) gesprungen, solange die if-Bedingung nicht erfüllt ist.

Der Sprung führt damit zu dem Befehl (CPI), der R24 mit dem Vorgabewert vergleicht. R24 wird aber auf diese Weise nicht mehr mithilfe von LDS aktualisiert. D. h. völlig unabhängig davon, ob cnt bei einem Druck auf den Taster in der Interrupt-Behandlung inkrementiert wird, vergleicht die Schleife jedes Mal mit dem ursprünglichen Wert 0, der am Anfang deklariert wurde. Für den Compiler ist nämlich die Verbindung zwischen der Hauptroutine und der Interrupt-Behandlung nicht ersichtlich. Daher geht er davon aus, dass cnt = 0 ist und auch immer Ø bleiben wird. Eine erneute Abfrage des Registers wäre also eine Ressourcenverschwendung und wird daher wegoptimiert. Glückwunsch, Sie sind dem Problem auf die Schliche gekommen.

Halten Sie den Debugger an, ergänzen Sie wieder volatile und starten Sie den Debugger erneut, um sich den Unterschied anzusehen. Öffnen Sie ggf. noch einmal das Disassembly-Fenster oder holen Sie den Tab nach vorne. Wenn Sie sich den Assembler-Code (Bild 12) jetzt am Breakpoint anschauen, werden Sie feststellen, dass es nur einen winzigen, aber alles entscheidenden Unterschied gibt: Bei BRCS wird nun an die Speicherstelle gesprungen, die drei Stellen vor dem aktuellen Wert des PC liegt: PC-0x03. Wenn Sie von der aktuellen Position 0x006B drei subtrahieren, ergibt sich als Sprungziel die Adresse 0x0068. Und genau hier wird mithilfe von LDS dafür gesorgt, dass sich R24 mit dem derzeitigen Inhalt der Speicherposition 0x0100 aktualisiert, an dem cnt gespeichert wird. Jetzt wird also bei jedem Vergleich mit dem korrekten Wert von cnt verglichen.

Genau betrachtet dauert dadurch alles etwas länger. In umfangreichen Programmen und bei zeitkritischen Anwendungen kann das in Summe durchaus relevant sein. Der LDS-Befehl benötigt auf dem ATmega328 zwei Taktzyklen zur Ausführung. Auf dem UNO mit seinem 16-MHz-Quarz bedeutet das für das Beispielprogramm eine Verzögerung von 125 ns – die aber in diesem Fall darüber entscheidet, ob Ihr Programm korrekt arbeitet.

Zurück zu ISP

Nach Ihrem Ausflug in die Welt des Debuggings sehnen Sie sich vielleicht wieder zurück zur Arduino IDE. Um wie gewohnt mit dem Arduino-Board über die serielle Schnittstelle kommunizieren und ihn per USB programmieren zu können, müssen Sie debugWIRE unbedingt wieder ausschalten und in den ISP-Modus wechseln.

Das ist ganz einfach: Starten Sie den Debugger mit einem beliebigen Programmcode, während Ihr Arduino mit dem Snap verbunden ist (mindestens die drei debugWIRE-Verbindungen). Wählen Sie dann, während der Debugger läuft, den Menüeintrag "Debug/ Disable debugWIRE and Close" aus. Nach einem kurzen Zugriff auf den Controller ist alles wieder beim Alten und Sie können den Debugger trennen.

Setzen Sie noch einen Jumper auf die eingelötete Steckbrücke oder schließen Sie die Kurzschlussbrücke, damit der Reset-Trick von Arduino wieder funktioniert und der Bootloader automatisch startet, wenn Sie ein Programm übertragen.

Mühe, die sich lohnt

Die in den beiden Debugging-Artikeln gezeigten Beispiel sind natürlich im Sinne eines Einstiegs einfach gehalten und wirken vielleicht etwas konstruiert. Sie zeigen zudem auch nur einen Ausschnitt der Möglichkeiten, die Ihnen der Snap und Microchip Studio bieten. Aber Sie haben ein paar wertvolle Techniken kennengelernt, die Sie bei Ihrer nächsten Fehlersuche anwenden können, wenn sich ein AVR-Programm mal merkwürdig verhält. — akf

Online-Shopping ohne Probleme: **c't hilft.**



Heft + PDF mit 29 % Rabatt

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

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

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

shop.heise.de/ ct-sicher-einkaufen23

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.



Wassertiefe messen und digital abfragen

Bei hohen Temperaturen können Teiche, Brunnen und Drainage-Schächte austrocknen. Um den Füllstand im Auge zu behalten, kann man mit einem hydrostatischen Drucksensor eine smarte Überwachung einrichten. So lässt sich der Wasserstand im Browser abrufen oder mit MQTT in Home Assistant einbinden.

von Uwe Rohne

M Markt werden verschiedene Sensoren für eine Füllstandsmessung von Wassertanks, Teichen und Zisternen angeboten. Viele haben mechanische Teile, die anfällig für Materialermüdung sind. Eine langlebigere Lösung bietet ein hydrostatischer Drucksensor. Seine Vorteile sind:

- keine mechanischen Teile
- hohe Genauigkeit
- einfache Installation: der Sensor wird z. B. im Teich versenkt
- kann f
 ür viele verschiedene Fl
 üssigkeiten angepasst werden (wird nicht in dieser Bauanleitung thematisiert)
- unabhängig von der Form des Behälters
- verschiedene Messbereiche, z. B. 0–1, 0–2 und 0–5 Meter Füllhöhe
- sehr geringer bis kein Wartungsaufwand
- Kompensation des sich ändernden Luftdrucks durch ein Kapillarröhrchen

Der Sensor wird an eine 24-Volt-Spannungsquelle angeschlossen. Je nach Füllstandshöhe fließt Strom, beispielsweise 4 mA bei 0 cm Füllstand (leer) bis 20 mA bei 100 cm (voll).

Messaufbau und Hardware

Damit man ein sauberes Messsignal erhält, bedarf es noch eines möglichst genauen Analog/Digital-Wandlers (A/D-Wandler) mit einer hohen und möglichst rauschfreien Auflösung – der Rest ist Software.

Da die Mikroprozessoren in der 5-V- und 3,3-V-Welt zu Hause sind, braucht man auch noch eine 24-V-Stromquelle für den Sensor. Das wird mit einem Step-up-Konverter gelöst.

Die Komponenten können auf einer einfachen Lochrasterplatine verdrahtet werden. Für die A/D-Wandlung wird der ADS115 genutzt. Dieser hat 4 Analogeingänge mit einer Auflösungsgenauigkeit von je 16 Bit (d. h. die zu messende Spannung von z. B. 0–3,3 Volt wird in 2¹⁶ = 65.536 Schritten gewandelt). Für den Prozessor fiel die Wahl auf den Wemos D1 Mini, der im Wesentlichen aus einem ESP8266 Prozessor und einem Seriell-USB-Wandler besteht.

Der Wemos D1 Mini besitzt eigentlich einen On-Board-A/D-Wandler. Allerdings bietet dieser nicht die gewünschte Auflösung und wird beim Messen durch das WiFi-Signal des Mikroprozessors durcheinandergebracht. Deshalb wird ein A/D-Wandler als zusätzliches Bauteil genutzt.

Der ADS1115 wird über den I²C-Bus mit dem Wemos verbunden. Die Stromversorgung erfolgt per USB-Netzteil für die gesamte Schaltung.

Vor dem ersten Anschließen des Sensors müssen über das Spindel-Potentiometer am Step-up-Konverter 24 Volt Ausgangsspannung eingestellt werden. Hierfür wird ein Spannungsmessgerät (z.B. ein Multimeter) benötigt. Es können 20 bis 30 Umdrehungen im

Kurzinfo

- » Messen der Füllhöhe von Wassertanks, Zisternen, Gartenteichen
- » Aufbau ohne mechanische Teile
- » Ausgabe über Browser, Telnet und MQTT-Schnittstelle



Material

- » Hydrostatischer Drucksensor
- passend für die Wassertiefe
- » Analog/Digital-Wandler Typ ADS1115
- » Wemos D1 Mini
- » Lochrasterplatine
- » Anschlussklemme
- » Feuchtraum-Aufputzdose » USB-Netzteil
- » DC-DC-Step-up-Konverter
- Тур МТ3608

Werkzeug

» Lötkolben » Multimeter

Mehr zum Thema

» Jan Moritz Behnken, Taupunkt-Lüftungssystem im Smart Home, Make 4/24, S. 36» Alexander Neubauer, Thermostate clever hacken, 2/24, S. 34



Uhrzeigersinn nötig sein, bis die 24 Volt erreicht sind. Die Schaltung bringt man am besten wasserdicht in einer Feuchtraum-Aufputzdose unter. Um das am besten schaffen zu können, muss man den A/D-Wandler unter dem Wemos-Board unterbringen. Hierfür verwendet man die den meisten Wemos-Boards beiliegenden Steckleisten mit den langen Anschlüssen und lötet die Fassungen etwas höher ein. Leider sind die Pins des DC/DC-Konverters nicht im 1/10-Zoll-Raster ausgelegt, die Fassungen kann man daher nur etwas schief einlöten. Für einen ersten Test der Schaltung empfiehlt es sich, statt des Sensors einen 2,2-k Ω -Widerstand zu verwenden. Die Ausgabe sollte dann eine Wassertiefe von knapp 400 mm anzeigen.

Die Software

Um den Wemos D1 mit der Arduino IDE nutzen zu können, muss das Board zuerst installiert werden. Wie das funktioniert, erklären wir in einem Artikel, der in der Kurzinfo verlinkt ist. Nach der Installation kann man über "Tools/ Board/esp8266/LOLIN(Wemos) D1 R2 & mini" das Board auswählen.

Vor dem Übersetzen der Software mit der Arduino IDE muss im Sourcecode in Zeile 17 (char ssid[18]="Name_des_WLAN", pass-



Es sind nur wenige Komponenten für die Füllstandsanzeige nötig.



word [18] = "Passwort_des_WLAN";) der Name des WLANs (SSID) und eine Zeile weiter das dazugehörige Passwort eingetragen werden. Außerdem muss man über den Bibliotheksverwalter in der Arduino IDE die Bibliotheken ADS1115-Driver.h für den A/D-Wandler und time.h für das Holen der Uhrzeit aus dem Internet installieren. Die vom Router vergebene dynamische IP-Adresse wird auf dem seriellen Port nach dem Rücksetzen mit ausgegeben. Nach der Initialisierung wird der A/D-Wandler alle 5 Sekunden abgefragt. Das Ergebnis wird auf dem seriellen Port der Arduino IDE und über Telnet (Port 23) ausgegeben. Die Telnet-Abfrage kann z. B. mit dem Windows Programm PUTTY oder Powershell erfolgen. Mit einem Internetbrowser kann man sich durch Eingabe der IP-Adresse ebenfalls die Werte anzeigen lassen.

Alle vollen 4 Stunden werden die Werte gespeichert. Es werden 8 Tage gesichert, somit

Sensor

Je nach Anwendungsfall braucht man unterschiedliche Messfühler, da sie für verschiedene Maximaltiefen ausgelegt sind. Um eine möglichst genaue Messung vornehmen zu können, sollte die verwendete Sonde möglichst genau für den maximalen Füllstand ausgelegt sein.

Das hier vorgestellte Projekt kann man auch nutzen, um den Füllstand anderer Flüssigkeiten zu messen. Auch hier muss man darauf achten, dass die Sonde auf die verwendete Flüssigkeit ausgelegt ist.

sind 8·6 = 48 Messwerte protokolliert. Nach 8 Tagen werden die ältesten Messwerte überschrieben.

Die Kalibrierung

Die Software gibt in der letzten Zeile den gemessenen Spannungswert aus, sodass hier kein Multimeter benötigt wird (siehe Ausgabe im seriellen Monitor der Arduino IDE). Da die Spannungsänderung zur jeweiligen Tiefe linear verläuft, werden lediglich zwei Messwerte benötigt.

Diese sind der Spannungswert der tiefsten Stelle, wo später auch der Sensor platziert wird, mit der dazugehörigen Tiefe in mm und



Für einen ersten Test verwendet man einen 2,2-kΩ-Widerstand.

Die Verdrahtung ist übersichtlich: Schwarz ist GND, Rot ist 3,3/5 Volt, der kurze weiße und der gelbe Draht führen das I²C-Signal und der lange weiße Draht führt zum A/D-Wandler.



Hydrostatischer Druck, was ist das eigentlich?

Die Füllhöhe h berechnet sich durch die Formel

 $h = P2 / (\rho \cdot g)$

wobei g der Erdbeschleunigung von 9,81 m/s² entspricht.



Variablen für die Berechnung



Trotz unterschiedlicher Geometrien und unterschiedlicher Wassermengen ist der hydrostatische Druck bei gleicher Wassertiefe in allen Gefäßformen identisch.

Die Faustformel für Wasser lautet: 100 kPa (= 1 bar) entsprechen der Füllhöhe von 10 m Wassersäule. Zum Beispiel beträgt der Druck bei Dieselkraftstoff bzw. Heizöl unter identischen Bedingungen 0,82 bar.

Auch der Umgebungsluftdruck (in der Grafik P1) hat einen Einfluss auf die Messung. Aus diesem Grund ist in der Zuleitung zu dem Druckmesser ein kleines Kapillarröhrchen integriert, um Luftdruckunterschiede zu kompensieren. Das bedeutet, wir brauchen uns um den sich ändernden Luftdruck nicht zu kümmern. Sich ändernde Temperaturen haben einen eher geringen Einfluss. Daher haben wir den Temperatureinfluss aus Vereinfachungsgründen weggelassen.

CE KI-PRAXIS

Mit 28 of * PDF

Hype oder Hilfe? Mit Künstlicher Intelligenz produktiv arbeiten

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

\overline shop.heise.de/ct-ki23

CE KI-PRAXIS Mit Künstlicher Intelligenz produktiv arbeiten

Grenzen der Sprachmodelle e Warum Sie Sprachmodellen nicht trauen dürfen Wie Unbeber im großen Still beklaut werden

KI-Programme anwenden Wo KI Assistenten tatsächlich helfen Ki-Stimmen - Schreibansistenten - Bildgeneratoren

Regeln für Schule und Arbei Wie Ni Schule und Arbeit verändert Was Unternehmen rechtlich beachten müssen

Die eigene Sprach-KI betreibe Mauraniseten Sprachmodellen Iskal oder under Geweinerten Sprachmodellen Iskal oder

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.

In heise shop

MQTT

Im GitHub-Repository zu diesem Projekt (Link in der Kurzinfo) befindet sich neben der hier beschriebenen Version des Programms noch eine, die mit einer MQTT-Schnittstelle erweitert ist.

Die Variable mqtt_broker speichert die IP-Adresse des MQTT-Servers (wenn ein Home-Assistant-Plug-in für MQTT verwendet wird, kann man hier homeassistant.local stehen lassen), topic1 ist der Name, unter dem die Daten übertragen werden sollen, und mqtt_port ist der Port des Servers (das ist in der Regel 1883).

Unter mqtt_user und mqtt_pass muss man die Zugangsdaten zu einem MQTT-Server eintragen.

Um das exemplarisch für Home Assistant einzurichten, installiert man auf der Web-Oberfläche von Home Assistant unter "Einstellungen/Add-Ons/Add-On Store" das Add-on "Mosquitto broker". Nach der Installation klickt man in der Leiste am oberen Bildschirmrand auf Konfiguration und trägt einen neuen User ein.

Danach klickt man unten rechts auf Speichern.

Über "Geräte & Dienste/Entitäten" kann man jetzt den MQTT-Broker zu Home Assistant hinzufügen. Jetzt braucht man das Add-on "File editor", das auch über Add-



Über den "File editor" kann die Konfiguration bearbeitet werden.

ons installiert wird. Nach der Installation muss hier der Schalter für "In der Seitenleiste anzeigen" umgestellt werden.

Jetzt erscheint an der linken Seite von Home Assistant die Option "File editor". Dort klickt man dann oben links auf den Ordner und lädt die configuration.yml. In dieser trägt man folgenden Code ein:

sensor:

```
- name: "Wasserstand"
   state_topic: "Wasserstand"
   unit_of_measurement: "mm"
```

Danach klickt man oben rechts auf "Speichern" und lädt dann in den Entwicklungswerkzeugen von Home Assistant die Option "Alle YAML-Konfigurationen" neu. Jetzt kann man in den Einstellungen unter "Geräte & Dienste/Entitäten" den Wasserstandsensor finden.

Um den jeweils aktuellen Wasserstand an einen MQTT-Server zu schicken, muss man die Server-Daten zwischen Zeile 5 und 9 ausfüllen.

5 const char *mqtt_broker = "homeassistant.local"; 6 const char *topic1 = "Wasserstand"; 7 const int mqtt_port = 1883; 8 const char *mqtt_user = "mq-serve"; 9 const char *mqtt_pass = "home";

der Spannungswert an der Oberfläche, was einer Tiefe von 0 mm entspricht. Für den Wert bei maximalem Füllstand muss der Sensor dann entweder in das zu messende Gefäß gelegt und dieses dann komplett gefüllt werden. Oder man verwendet ein senkrecht stehendes HT-Rohr (Hochtemperatur), das die gleiche Höhe wie der maximale Füllstand des späteren Einsatzortes für den Fühler hat. Unten wird das Rohr verschlossen, der Fühler eingelegt und das Rohr mit Wasser gefüllt. Das Wasservolumen spielt dabei keine Rolle. Es kommt nur auf die Höhe an.

Beispiel: In 900 mm Tiefe werden 1,4870 Volt und in 0 mm Tiefe 0,4300 Volt gemessen, dann muss Zeile 58 im Quellcode wie folgt aussehen: unsigned long tiefe1 = 900 *(value0-4300)/(14870-4300);

Diese Codezeile besagt in etwa Folgendes: Die gemessene Spannung kann zwischen 0,43 Volt und 1,487 Volt liegen. Die Werte dazwischen werden dann auf die maximale Füllhöhe von 900 mm aufgeteilt. Diese Funktion ändert sich für jede Flüssigkeit (siehe dazu Kasten "Hydrostatischer Druck, was ist das eigentlich?"). Nach dem Eintragen der Werte muss die Software neu übersetzt und auf den Wemos D1 Mini aufgespielt werden. Und wenn dann die Sonde professionell im Teich versenkt wurde und die Feuchtraumdose unter einem dekorativen Stein versteckt ist, hat man immer einen Überblick über den Wasserstand. Wir wünschen viel Erfolg beim Nachbau. —das

Projekt



Die Pin-Leiste des Wemos-Boards ist mit etwas Abstand zur Platine eingelötet worden, um den A/D-Wandler unter dem Board zu platzieren.

<	> @	公 0 19	2.168.178.	94	0	17.	۲ <u>⊒</u> 101	=
FU	ELLST	ANDSAN	ZEIG	E				
				-				
27.0	3.2024-17	:42:07; 0_Uhr	: 4_hr	8_Uhr	12_Uhr	16_Uhr	20_Uhr	
Tag	20.	702;	703;	686;	684;	679;	661	
Tag	21.	650;	646;	637;	626;	638;	622	
Tag	22.	629;	625;	613;	604;	633;	637	
Tag	23.	630;	652;	664;	709;	698;	694	
Tag	24.	700;	703;	707;	702;	703;	695	
Tag	25.	697;	698;	699;	695;	693;	679	
Tag	26.	689;	675;	667;	659;	662;	663	
Tag	27.	654;	655;	645;	633;	654;	0	
AK	TUELL	: Stunde=17;	1.2235V;	675mm;	-83;RSS	: 3033E	SDVCC	

Oben links wird die aktuelle Uhrzeit dargestellt. In der letzten Zeile die aktuelle Stunde, die vom A/D-Wandler via I²C gelieferte Spannung, dann die daraus errechnete Tiefe, der WLAN-Empfangspegel und die Betriebsspannung des ESP in mV.



Und das ist die ganze Installation. Die Messsonde liegt am Grund des Teiches und misst.

Es gibt 10 Arten von Menschen.

iX-Leser und die anderen.



Jetzt Mini-Abo testen:

3 digitale Ausgaben + Bluetooth-Tastatur nur **19,35 €**

www.iX.de/digital-testen

🗰 www.iX.de/digital-testen

leserservice@heise.de

o 49 (0)541 800 09 120

© Copyright by Maker Media GmbH. 5/2024 | 57



Automatisiert Strom sparen

Als Tibber-Nutzer sparen Sie mit diesem selbst gebauten Tibber-Relais Zeit und Energie. Es steuert große Geräte automatisch und findet den günstigsten Zeitpunkt dafür.

von Uwe Rohne

it einem dynamischen Stromvertrag kennt man bereits am Vortag die kWh-Preise der kommenden 24 Stunden. Da die Preise erheblich schwanken (im Juni/Juli waren es zwischen 5 und 40 Cent pro kWh), sollte man den Einsatz größerer Verbraucher wie E-Auto, Wärmepumpe oder Waschmaschine im Vorfeld planen. Dabei hilft der Tibber-Preisrahmen aus der Make 4/24 (Bild 1). Ein- und ausschalten muss man seine Geräte aber immer noch von Hand. Und wenn Sie Pech haben, am besten um 3 Uhr morgens. Mit dem hier gezeigten Tibber-Relais lässt sich das aber auch ganz leicht automatisieren.

Einfacher Aufbau

Die benötigte Hardware ist sehr simpel. Im Prinzip reichen ein WEMOS D1 mini (mit ESP8266-Prozessor), ein USB-Steckernetzteil sowie ein Shelly-Gerät. Um auch eine Wärmepumpe oder eine Wallbox steuern zu können, habe ich eine Schaltung entworfen, die mit einem Optokoppler und einem Relais funktioniert (Bild 2). Diese lässt sich auf einer Streifenraster-Platine aufbauen (Bild 3 und 4).

Da ein 3,3-V-Relais eine Stromaufnahme von deutlich mehr als 20 mA hat – in meinem Beispiel ca. 90 mA –, wird ein NPN-Universal-Transistor (BC107, BC108, BC109, BC547 usw.) zur Ansteuerung dazwischengeschaltet. Diese Transistoren haben mindestens eine hundertfache Stromverstärkung. Wir brauchen also einen Basisstrom von mindestens 1 mA, damit der Transistor voll durchsteuert. Mit einem 1-k Ω -Widerstand beträgt der Strom sogar ca. 2,5 mA.

Die Universal-Diode, antiparallel zum Relais, schließt die beim Abschalten entstehende Induktionsspannung kurz und schützt somit die Schaltung vor der Zerstörung durch zu hohe Spannungen. Es besteht auch die Möglichkeit, ein 5-V-Relais zu verwenden. In diesem Fall schließt man das Relais und die Diode einfach an den 5-Volt-Pin des WEMOS an. Theoretisch könnte man mit dem Kontakt des Relais einen 230-V-Verbraucher betreiben, wenn die Leistung des Relaiskontaktes mit



UNS ND

Bild 1: Der Tibber-Preisrahmen zeigt die Preise des kommenden Tages auf einem ePaper-Display.

Kurzinfo

- » Dynamische Stromtarife clever nutzen
- » Große Verbraucher automatisch zur besten Zeit ein- und ausschalten
- » Relais, Optokoppler und Shelly-Systeme mit WEMOS steuern



T1 BC547 X2-1 X2-2 1K ESP8266 R1 5U 303 GND D8 GND D7 D4 D3 D6 D2 D5 220 D1 DØ OX1-1 R2 AØ RX TX RESET OX1-2 WEMOS1 PC81

GND

Bild 2: Die Schaltung mit dem optionalen Relais (K1) und / oder einem Optokoppler (PC817). Das benötigte USB-Netzteil ist nicht dargestellt, wird aber direkt am WEMOS angeschlossen.



Bild 3: Das Tibber-Relais von der Lötseite. Um die senkrechten Bahnen zu unterbrechen, habe ich bis auf zwei Ausnahmen einen 3-mm-Bohrer benutzt.

dem zu schaltenden Verbraucher passt. Wer hier aber kein Sicherheitsrisiko mit Netzspannung eingehen will, sollte lieber zu einer Shelly-Steckdose greifen (dazu später mehr).

In dem Optokoppler ist eine LED verbaut, die einen Fototransistor ansteuert. Damit sind ähnlich wie bei dem Relais die Stromkreise zur Steuerung und des zu steuernden Gerätes galvanisch getrennt. Wenn durch die LED etwa 1,4 V und 10 mA Strom fließen sollen, müssen am Vorwiderstand 1,9 V abfallen. Dieser Widerstand lässt sich wie folgt ermitteln:

 $R = (3,3 V - 1,4 V)/10 mA = 190 \Omega$

Mit einem 220- Ω -Widerstand ist man auf der sicheren Seite, die LED im Optokoppler nicht



Bild 4: Auf der Bestückungsseite ist zu erkennen, dass der Optokoppler unter der WEMOS-Platine Platz gefunden hat.

zu überlasten und gleichzeitig den Fototransistor sicher zu schalten. Um den Aufbau zu schützen, bietet sich als preiswertes Gehäuse eine Aufputz-Verteilerdose aus dem Baumarkt an (siehe Titelbild des Artikels).

Firmware flashen

Das WEMOS D1 mini programmieren wir mit der Arduino IDE. Im GitHub-Repository des Projekts stehen der Sketch sowie die Dateien time.h, ota.h, internetzeit.h und uwesserver.h zum Download bereit (siehe Link in der Kurzinfo). Diese müssen sich nach dem Herunterladen und Entpacken alle in demselben Ordner befinden, weil der Sketch auf sie zugreift. Um das Programm für den Mikrocontroller kompilieren und überspielen zu können, muss der ESP8266-Core (3.1.2) in der Arduino IDE installiert sein (Anleitung siehe Link in der Kurzinfo). Neben den Board-Treibern liefert dieser Core die benötigten Bibliotheken ESP8266WiFi, ESP8266HTTPClient, WiFiClient SecureBearSSL und EEPROM. Eine weitere Bibliothek, ArduinoJson, muss man über den Bibliotheksverwalter installieren.

Danach kann man den Sketch tibber_relay. ino öffnen. Vor dem Übersetzen der Software gibt man die SSID seines WLANs, das zugehörige Passwort und seinen Tibber-Token im Sketch ein. Der Token lässt sich über die Tibber-Developer-Website beziehen (siehe Link in der Kurzinfo). Im Sketch befindet sich ein Demo-Token, der keine tagesaktuellen Werte und keine Netzentgelte generiert. Zum Testen reicht er aber völlig aus. Diesen Teil der Firmware habe ich 1:1 aus dem Code des Tibber-Preisrahmens (Make 4/24) übernommen. Die Web-Oberfläche habe ich von Hand mit entsprechenden HTML-Elementen zusammengefrickelt. Leider ist der Code etwas unübersichtlich geworden, aber er funktioniert.

Sobald man das WEMOS mit einem USB-Kabel an den Computer angeschlossen hat, wählt man in der Arduino IDE im Menü "Werkzeuge" das Board "LOLIN(WEMOS) D1 R2 & mini" und den seriellen Port aus, mit dem das WEMOS verbunden ist. Danach lässt sich die Firmware mit einem Klick auf "Hochladen" auf den Mikrocontroller flashen und das WEMOS resetten.

Zeitfenster einstellen

Verschiedene Verbraucher benötigen oft individuelle Zeitfenster: Zum Beispiel lädt man ein E-Auto in der Nacht oder steuert die Luftwärmepumpe eher tagsüber, da ihr Wirkungsgrad nachts durch die kälteren Außentemperaturen schlechter ist – auch wenn die Strompreise nachts in der Regel besser sein sollten.

In dem Web-Interface des Tibber-Relais, das man im Browser durch die Eingabe der IP-Adresse des WEMOS aufruft (Bild 5), lässt sich solch ein Zeitfenster einstellen. Die be-

Projekt

nötigte IP-Adresse kann man mithilfe des seriellen Monitors in der Arduino IDE ermitteln, sobald man den Mikrocontroller startet und er sich mit dem WLAN verbunden hat. In den Feldern "Zeitfenster" trägt man die Start- und Endstunde ein, die das Zeitfenster begrenzt. Strompreise außerhalb des gewählten Zeitfensters werden nicht berücksichtigt.



Bild 5: Über die Eingabefelder können anspruchsvolle Schaltanforderungen eingestellt werden.

void shelly() void shelly(boolean ein){ WiFiClient client; HTTPClient http; strcpy(puffer, "http:// strcat(puffer, html_i); "http://"); strcat(puffer, "/relay/0?turn="); if(ein==HIGH) strcat(puffer, "on"); else strcat(puffer, "off"); if (http.begin(client, puffer)){ int httpCode = http.GET(); if (httpCode > 0){ if (httpCode == HTTP_CODE_OK || httpCode == HTTP_CODE_MOVED_PERMANENTLY) { String payload = http.getString(); } http.end(); } }



Geballtes

Nerd-Wissen!

 Sicher oder nicht? Security-Mythen im Faktencheck

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



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.



© Copyright by Maker Media GmbH. 5/2024 | 61

Äußere for-Schleife

Innere for-Schleife



Bild 6: Prinzipdarstellung des Bubble Sort

2 3 4 6 9

nicht tauschen

Ist die Endstunde kleiner als die Startstunde, werden die Werte über Mitternacht bewertet. Für ein E-Auto könnte die Einstellung zum Beispiel von 20 bis 6 lauten. Hierbei beginnt die Startstunde mit 00 Minuten und die Endstunde endet mit 59 Minuten. Somit werden die Strompreise in diesem Beispiel von 20:00 bis 6:59 Uhr berücksichtigt.

Bubble Sort

Um die besten Strompreise zu ermitteln, müssen diese zuerst sortiert werden. Hierfür wird der Algorithmus "Bubble Sort" angewandt (siehe Bild 6 und das Listing). Dieser vergleicht als Funktion void bubblesort() jeweils zwei benachbarte Elemente des Arrays arr und tauscht sie mit void swap(), wenn der aktuelle Wert größer ist als der nachfolgende. Das geht so lange, bis alle Werte von klein nach groß geordnet sind.

Das Sortieren erledigen dabei zwei for-Schleifen: eine innere, die jeweils die benachbarten Elemente mit if (arr[j] > arr[j+1]) vergleicht,

und eine äußere, die prüft, ob nochmals verglichen werden muss. Diese Methode ist gut für kleine Datenmengen geeignet.

```
void bubbleSort(int arr[], int n)
 for (int i = 0; i < n-1; i++)
   for (int j = 0; j < n-i-1; j++)
     if (arr[j] > arr[j+1])
        swap(arr[j], arr[j+1]);
}
void swap(int &a, int &b) {
 int c = a;
```

a = b;

```
b = c;
```

```
}
```

4 C ▲ Nicht sicher 192.168.178.44/t \rightarrow

Tibber-Relais-Preise

		5	sa.				Ш			5	so.			
00h	•	3139	11	12h		2092	<u> </u>	00h	*	3229	11	12h		1918
01h		3097	11	13h		1993	THE	01h	*	3123	11	13h		1917
02h		3027	11	14h	*	1966	THE	02h	*	3070	11	14h		1892
03h		2977	11	15h	*	2065	1111	03h	*	3020	11	15h		1917
04h	•	2944	11	16h	*	2303	1111	04h		2991	11	16h		1929
05h		3046	11	17h	*	2793	THE	05h		2905	11	17h		2184
06h	•	3152	11	18h	*	3149	1111	06h		2842	11	18h		2788
07h		3054	11	19h	*	3226	1111	07h		2788	11	19h		3175
08h		2923	11	20h	х	3400	THE	08h		2678	11	20h		3344
09h	•	2703	11	21h	x	3543	1111	09h		2252	11	21h		3405
10h		2303	11	22h	x	3336	1111	10h		2013	11	22h		3261
11h	•	2159	11	23h	*	3123	1111	11h	•	1942	11	23h	•	3165

Bild 7: Mit dieser Übersicht kann man die automatische Planung bei Bedarf prüfen.

Da die Preise des Folgetages erst nach 13 Uhr vorliegen, darf bei einem Zeitfenster über Mitternacht die Startzeit nicht vor 14 Uhr liegen. Bei einer Fehleingabe wird dieser Fehler leider nicht innerhalb der HTML-Maske erkannt, sondern erst bei einem weiteren Aufruf signalisiert.

Bitte schalten, wenn...

Es gibt insgesamt vier Möglichkeiten, um zu bestimmen, bei welchen Strompreisen geschaltet werden soll:

1. Das Tibber-Relais schaltet, wenn der aktuelle Stundenpreis innerhalb des Zeitfensters besser ist als der Mittelwert + Minimum geteilt durch zwei.

- 2. Wie unter 1, es wird jedoch nur der Mittelwert innerhalb des Zeitfensters betrachtet.
- 3. Wie unter 1, jedoch rechnet das System mit dem Maximum statt mit dem Minimum.
- 4. Hier kann man die Zahl der eingeschalteten Stunden innerhalb des Zeitfensters frei wählen. Wenn man Verbraucher steuert, die eine bestimmte Dauer eingeschaltet werden (oder bleiben) müssen, ist das die optimale Einstellung. Ist die Zahl der Mindeststunden gleich oder größer als das Zeitfenster, gibt es bei einem erneuten Aufruf der Maske eine Fehlermeldung. Sollen die Stunden zusammenhängen, zum Beispiel beim Betrieb einer Waschmaschine, ist die Option "Stunden im Stück" zu wählen.

Die eingestellten Werte werden jeden Tag gemäß der aktuellen Preissituation neu bewertet und es wird entsprechend geschaltet.

Mit Shelly kommunizieren

In das letzte Feld der Eingabemaske kann man die IP-Adresse eines Shelly-Systems eingeben. Dorthin sendet das WEMOS eine mit der Funktion void shelly() zusammengesetzte HTTP-Zeile, um die Steckdose oder den Schaltaktor zu schalten, z. B.:

http://192.168.178.126/relay/0?turn=on

Wobei "on" einschaltet und "off" ausschaltet. Shelly nennt dies "Local Action". Man kann den Befehl auch testen, indem man ihn in die Adresszeile des Browsers eingibt. Die IP-Adresse ist beispielhaft und muss angepasst werden; und natürlich müssen sich beide Systeme im gleichen WLAN befinden. Nähere Informationen zu der Local-Action-Funktion liefert der Hersteller in seiner API-Dokumentation (siehe Link in der Kurzinfo).

Um das Absenden des Schaltbefehls zu bestätigen, wechselt die Eingabemaske die Hintergrundfarbe zwischen Hellblau und Hellbeige hin und her.

WEMOS übernimmt ab hier

Eine Minute, nachdem man die Werte in der Eingabemaske geändert hat, werden diese im EEPROM des WEMOS gespeichert, damit sie auch nach einem Neustart bestehen bleiben. Danach führt der Mikrocontroller das folgende dreistufige Programm aus:

- Mit dem Erreichen der Startzeit werden die aktuellen Tibber-Preise geladen.
- Danach werden in der Funktion ermittle_ schaltzeiten() das Minimum, Maximum und der Mittelwert für ein eingestelltes Zeitfenster ermittelt. Außerdem sortiert der Algorithmus "Bubble Sort" (siehe Kasten) die Preise und legt sie in der Tabelle preis_sort() ab. Die Funktion wird auch bei Änderungen in der Eingabemaske aufgerufen.
- Das eigentliche Schalten übernimmt die Funktion preisabhaengig_schalten(), die beim Wechsel einer jeden Stunde aufgerufen wird.

Plan im Blick

Wenn man die berechneten Schaltzeiten und die aktuellen Tibber-Preise sehen möchte,

muss man die IP-Adresse des WEMOS gefolgt von einem /t (t wie Tabelle) im Browser eingeben, z. B. 192.168.178.44/t. Daraufhin erscheint eine Übersicht, die auf der ursprünglichen Konfiguration in der Eingabemaske beruht (Bild 7).

Der Punkt bedeutet, dass die entsprechende Stunde sich außerhalb des Zeitfensters befindet. Bei einem "x" passiert nichts. Nur dort, wo ein Sternchen steht, schalten sich das Relais, der Optokoppler und / oder das Shelly-System ein. Bei einem Zeitfenster über Mitternacht liegen die Preise erst ab 14 Uhr vor, sodass die Tabelle vor 14 Uhr keine sinnvollen Werte anzeigen kann. Liegen die Zeiten innerhalb eines Tages (also nicht über Mitternacht), werden nur die Schaltungen des aktuellen Tages angezeigt und NICHT die des Folgetages – auch wenn der Aufruf nach 14 Uhr ist.

Die in der Tabelle angezeigten Strompreise verstehen sich pro kWh brutto, inklusive aller Netzentgelte und der Mehrwertsteuer, und werden in 1/100 Cent angezeigt. Die Ausgabe 3214 bedeutet demnach 32,14 Cent und somit 0,3214 Euro. Zum verbrauchten Strom kommt bei Tibber noch die monatliche Grundgebühr von zurzeit ca. 15 Euro dazu. —*akf*

Ihr Erste-Hilfe-Set: Das Notfall-System für den Ernstfall



c't Desinfec't 2024 komplett digital auf einem 32 GByte USB-Stick! Alle Anleitungen und Artikel zur Einrichtung des Rettungssystems finden Sie auf dem USB-Stick. Das ist neu an c't Desinfec't 2024:

- Gratis Signatur-Updates bis Juni 2025
- Detailverbesserungen beim Einbinden von Laufwerken
- Neue Expertentools zu Malwarejagd wie Capa und FLOSS
- Bei Update-Problemen: Signaturen von Scannern zurücksetzen
- Kernel 6.5 (optional 6.8 f
 ür neue Hardware)

Desinfec't-Stick 19,90 €

🐺 shop.heise.de/desinfect24

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.

In heise shop



Ein Maker-Paradies in der Wüste

Make zu Besuch bei Europas größtem Burning-Man-Ableger – Nowhere. Seit zwei Jahrzehnten versammeln sich jährlich 3000 Hippies, Maker und Künstler in einer spanischen Halbwüste, um eine wahre DIY-Orgie zu zelebrieren. In dieser Reportage stellen wir euch zwei Projekte vor, die auf ganz unterschiedliche Weise zum einzigartigen Flair des Festivals beigetragen haben, aber auch in anderen Kontexten nachgebaut werden können.

von Marcus Hansson



Kurzinfo

- » Bericht über das Maker-freundliche Nowhere-Festival in Spanien
- » Am Beispiel des Kunstwerks Firestarter werden die Grundlagen von Feuermaschinen erklärt
- » Präsentation einer Duschanlage, bei der der Wasserverbrauch mit Arduino und RFID-Tags gesteuert wird

Mehr zum Thema

c't Hacks 4/2014, S. 84

Alles zum Artikel im Web unter » Elke Schick, Nirgendwo in der Wüste, make-magazin.de/xvhg





Bild 1: Stundenlang kann man in die riesige Plasmakugel hineinschauen.

er ein ganzes Jahr ehrenamtlich arbeitet und Vorbereitungen trifft, um für eine einzige Woche eine ganze Stadt in der Wüste entstehen zu lassen, muss verrückt sein. Und ein bisschen verrückt sind die Leute vom Nowhere Festival schon, aber auf eine eindeutig positive Weise. Schon vor zehn Jahren war Make dort zu Besuch - den Link zur alten Reportage findet ihr in der Kurzinfo.

Es ist kein herkömmliches Festival, sondern es baut auf die zehn Prinzipien von Burning Man. Das bedeutet vor allem, dass kreative Zusammenarbeit großgeschrieben wird. Sämtliche Festivalbesucher sind nämlich auch Mitgestalter des Events. Die beeindruckenden Kunstinstallationen, die ohrenbetäubende Musik und das leckere Essen - alles wird in Eigenregie aufgestellt.

Vor allem nachts entfaltet Nowhere seine volle Magie. Kleine Gruppen von Menschen



Bild 2: Die mit viel Liebe gestalteten Kunstwerke schaffen eine ganz besondere Atmosphäre.



Bild 3: Das "Oculus Eye" ist eine drei Meter große, begehbare Kunststoffkugel, die einem Augapfel ähnelt. Auf einer Seite sind Tausende adressierbare LEDs kreisförmig angeordnet.



Bild 4: Mit den LEDs kann die Iris von den Besuchern nachgebildet werden. So befindet man sich im eigenen Auge und kann die Welt von dort aus betrachten.

schlendern über die sogenannte Playa, ein offenes Gelände, umrandet von Zelten. Das Gelände ist gesprenkelt mit bunten Lichtinstallationen und wirkt wie eine Art Kirmes der Nacht, wo überall etwas wummert, fährt oder leuchtet. Die Menschen werden wie Motten vom Licht angezogen (Bild 1 und 2).

In den oft sehr kruden und schnell zusammengeschraubten Objekten steckt jede Menge Elektronik. Zum Beispiel das "Oculus Eye" (Bild 3 und 4): Das ist eine drei Meter große Kunststoffkugel, die wie ein Augapfel aussieht. Durch eine runde Öffnung kann man in das Auge hineinklettern und es sich auf dem dort angebrachten Trampolin bequem machen. Auf einer Seite sind Tausende adressierbare LEDs kreisförmig angeordnet und bilden eine Art Bildschirm. Besucher können ihre Iris fotografieren lassen, die dann auf dem Bildschirm erscheint. Man befindet sich also in seinem eigenen Auge und kann die Welt von dort aus betrachten.

Maker-freundliche Atmosphäre

Nicht nur die Kunstinstallationen machen Nowhere zu einem Wallfahrtsort für Kreative und Maker. Überall kommen fantasievolle Lösungen zum Einsatz, etwa das aufwendige Vergraben von kilometerlangen Stromkabeln, die von Dieselgeneratoren ausgehend das bunte Lichtspektakel mit Strom versorgen. Selbst geschmiedete Pflüge werden hinter

Autos gespannt, um Rinnen in den Boden zu ziehen. Nachdem die Kabel verlegt worden sind, werden die Furchen wieder zugemacht. Da die meisten Künstler und Maker mit Elektronik arbeiten, gibt es auch ein kleines Maker-Space vor Ort. Dort sind Lötkolben und einiges anderes an Material vorhanden. Es wurde unter anderem von dem schwedischen Künstler Jonas Johansson (siehe weiter unten unter der Überschrift "Firestarter") aufgebaut, um mehr Elektronik- und Programmierprojekte in der Wüste zu ermöglichen. "Ich finde es sehr wichtig, dass Menschen, die hierherkommen, auch fühlen, dass sie hier mit Elektronik arbeiten können, auch wenn die Umgebung nicht für Feinarbeit angepasst ist. Ich würde mir wünschen, dass es mehr davon gibt und weniger Steampunk-Aktivitäten. Oder beides, am besten zusammen", sagt Jonas Johansson.

Hilfsbereitschaft statt Geld

Geld wird auf dem Festivalgelände nicht benutzt, einfach aus dem Grund, dass es nichts zu kaufen gibt. Stattdessen wird alles geteilt. Und auch wenn jeder letztendlich für sich alleine zuständig ist, gehört es dazu, dass alle darauf achten, dass es den anderen gut geht. Das schafft eine ganz besondere Atmosphäre von gegenseitigem Respekt und Wertschätzung. Wer beispielsweise mit der Spitzhacke in der prallen Sonne damit beschäftigt ist, schnell einen Graben zu ziehen, damit das Regenwasser von dem anbahnenden Sturm abgeleitet werden kann, bekommt ab und zu Besuch von "Fluffers". Diese bringen Wassermelone, Knabberzeugs oder Wasser zum Trinken vorbei. Oder sie kühlen die Arbeitenden mit Wassernebel aus der Sprühflasche ab.

Und das alles braucht es, um hier arbeiten zu können, denn die Wetterbedingungen sind brutal. Das offene Gelände, auf dem das Ganze stattfindet, wurde dieses Jahr abwechselnd von kräftigen Windböen, Gewittern und Schlagregen heimgesucht. Dazwischen sorgte eine unbarmherzige Sonne dafür, dass die Temperaturen auf über 35 Grad stiegen. Mit anderen Worten: perfekte Bedingungen für kreative Maker und deren selbst gebastelte Innovationen, die darauf abzielen, das Partyleben so angenehm wie möglich zu gestalten.

Und nach eine Woche Party ist das verrückte Wüstencamp dann kurz nach Ende des Festivals wie eine Fata Morgana spurlos verschwunden. Denn zu den Burning-Man-Prinzipien gehört auch "Leave no Trace" – keine Spuren hinterlassen.

Auf den folgenden Seiten findet ihr nun die Beschreibung zweier Projekte vom Nowhere-Festival, die auf ganz unterschiedliche Weise auch an anderen Orten sinnvoll umgesetzt werden konnten.



Bild 5: Steampunk-inspirierte Steuerkonsole für ein interaktives Kunstwerk



Bild 6: Unklar ist, ob es sich um einen Hardwareoder Softwarefehler handelt.



Bild 8: Alessandra und Jesse aus Kanada haben einen Waffelstand aus recyceltem Material gebaut. Von dort aus sollen mitten in der Nacht auf der Playa Waffeln serviert werden.



Bild 11: Die Fummelsäule. Eine Person steigt durch eine Öffnung in die Säule. Andere betatschen ihn durch mehrere kleine Löcher.



Bild 9: Jemand sagte, Nowhere sei ein "Karneval der Absurditäten", notdürftig zusammengehalten von Gaffa, Kabelbindern und Tackerklammern. Hier ist ein gutes Beispiel dafür.



Bild 12: Ständig geht etwas kaputt. Zum Glück gibt es auf dem Nowhere mehrere gut ausgestattete Werkstattcontainer.



Bild 7: Ein französisches Minitel-Terminal aus den 90er-Jahren hat in diesem Kunstwerk ein neues Leben als mehr oder weniger zuverlässiger Wahrsager erhalten.



Bild 10: In einer Art solarthermischem Rohr wird Baguette gebacken.



Bild 13: Der Glücksdrache Fuchur aus "Die Unendliche Geschichte", hier als herumfahrendes ArtCar



Bild 14: Öffentliches Wi-Fi über Starlink gehört mittlerweile auch zur Standardausrüstung auf dem Festival.

Vom Make-Leser zum Nowhere-Fan

Vor ziemlich genau zehn Jahren blätterte Oliver aus Deutschland in der damaligen c't Hacks, dem Vorgänger der Make. Dort stieß er auf eine Reportage über Nowhere, die seine Neugier so sehr weckte, dass er beschloss, dorthin zu reisen. Seitdem ist er Nowhere treu geblieben und hilft nun sogar, das Festival zu organisieren.

"Da war damals ein Artikel über ein buntes Völkchen an Menschen, die in irgendeiner spanischen Wüste eine Woche lang ihre eigene Stadt aufbauen und quasi aus den Abfällen der Zivilisation oder Elektroschrott Kunstwerke erschaffen. Und die sollten auch noch sehr nett sein", erinnert sich Oliver.

Schon bei seinem ersten Besuch war er überzeugt und ist heute sogar Teil des NOrg (Nowhere Organization), einer Gruppe von Freiwilligen, die Nowhere organisieren und koordinieren. Dieses Jahr war er für den NoInfo-Stand zuständig, eine Anlaufstelle für alle Fragen rund ums Festival. Dort werden aber auch jede Menge irreführender Informationen verteilt, was als typischer Nowhere-Humor gilt. Auch betreute er in technischer Hinsicht die Artomobile, also bewegliche Kunstwerke, die über die Playa fahren.

"Ein ArtCar oder Artomobile kann ein Auto sein mit irgendwas drauf, zum Beispiel einem Elefanten. Es kann aber auch ein Fahrrad sein, das optisch und technisch aufgewertet wurde. Da sind die Grenzen sehr fließend", erklärt Oliver.

Trotz der Selbstorganisation wird großer Wert auf Sicherheit gelegt. Bevor jemand mit einem mitgebrachten rollenden Kunstobjekt über die Playa fahren darf, muss sichergestellt werden, dass Bremsen und Licht funktionieren, und es wird entschieden, wie viele Leute ohne großes Risiko auf dem Rücken des fahrenden Elefanten mitfahren können.

Oliver, der im normalen Leben in einem technischen Beruf arbeitet und viel mit Elektronik gebastelt hat, findet, dass Nowhere der perfekte Ort für Maker ist.

"Ich kann hier sehr kreativ sein und quasi machen, was ich möchte. Es macht mir einfach Spaß, zu sehen, was die Menschheit so produziert, wenn man sie einfach machen lässt, ohne irgendwelche Zwänge."

Als europäischer Ableger des amerikanischen Burning Man wird Nowhere umgangssprachlich als "Burn" bezeichnet, also ein Festival oder eine Party im Stil von Burning Man. Das wichtigste Merkmal dabei ist für Oliver das Mitmachen aller Festivalbesucher.

"Dieses Volunteering, diese Partizipation, das ist ein ganz großes Ding. Alles, was dort passiert, passiert, weil es Freiwillige machen. Wenn du es nicht machst oder wenn du es nicht mitbringst, wird es wahrscheinlich auch nicht passieren. Das heißt, es ist ein riesiges Gemeinschaftsprojekt. Es gibt keinen Masterplan, der da abgearbeitet wird."

Für einen langjährigen Make-Abonnenten wie Oliver ist das Festival auch ein lebendiges Beispiel dafür, was mit dem Wissen aus der Make alles gemacht werden kann.



"Das, was man in der Make lernt, hält einem so ein bisschen den Spiegel vor, was manche Menschen mit diesem Wissen anfangen. Es kommen sehr schöne Ergebnisse dabei heraus. Das zeigt, welche Schönheit man mit Elektronik erzeugen kann, welche atemberaubenden Projekte entstehen, wenn man Transistoren, Kondensatoren und LEDs miteinander verbindet und natürlich auch viel Zeit investiert."





Der Firestarter – ein riesiges Feuerzeug gesteuert von einem normalen Feuerzeug

In der Mitte von Nowhere liegt wie gesagt die Playa, auch "Field of Dirt" genannt. Tagsüber beim Spaziergang werde ich von einem großen, metallglitzernden Objekt angelockt. Es ist eine vier Meter hohe Nachbildung des klassischen Zippo-Sturmfeuerzeugs. Der schwedische Lichtkünstler Jonas Johansson und seine Partnerin Rose Hallgren (Bild 17) sind gerade dabei, ihr Kunstwerk mit dem Namen Firestarter zu vollenden.



Bild 17: Jonas Johansson und Rose Hallgren aus Schweden haben den Firestarter entwickelt, ihr erstes Kunstwerk mit Feuer Der deutschspanische Künstler Erik Schmitz war ebenfalls beteiligt.

Die Funktionsweise ist simpel, aber faszinierend: An einem Pult, fünf Meter von dem gigantischen Feuerzeug entfernt, befindet sich eine mit Metall bedeckte Holzplatte mit einem kleinen Loch in der Mitte (Bild 19). Dort ist ein winziger IR-Sensor versteckt. Wer sein eigenes Feuerzeug aus der Tasche zieht und es kurz über das Loch hält, entfacht eine beeindruckende, meterhohe Flamme oben auf dem Riesenfeuerzeug (Bild 18). "Ich rauche nicht, aber ich sage scherzhaft, dass ich endlich eine Installation für Raucher gemacht habe. Jetzt sind sie diejenigen, die glänzen werden. Denn nur sie können es benutzen, da man es ohne Feuer nicht auslösen kann. Das finde ich sehr amüsant", erklärt Jonas Johansson.

Regen, Hitze und Wind

Gesteuert wird das Ganze über einen Arduino Uno mit DMX-Shield. DMX steht für Digital Multiplex und ist das gängige digitale Steuerprotokoll im Bereich Bühnen- und Veranstaltungstechnik. Nachdem der Sensor aktiviert wurde, wird ein DMX-Signal an drei DMX-Aufnahmevorrichtungen gesendet, mit dem Befehl, sowohl die Feuermaschine als auch mehrere LED-Leuchten innerhalb des Zippo-Feuerzeugs einzuschalten. Bild 20 zeigt eine schematische Darstellung des Aufbaus. Im



Bild 19: Unter dem kleinen Loch sitzt der IR-Sensor (Akozon IR Flame Sensor Module Detektor).

Bild 20: Der Arduino Uno mit DMX-Shield und IR-Sensor sendet über das DMX-Kabel das Signal zur Feuermaschine, eine Flamme zu erzeugen. Der blaue Stecker unten links ist ein 230-V-PowerCON-Anschluss für die Feuermaschine.





Bild 18: Das Riesenfeuerzeug dient keinem Zweck, außer schön auszusehen und Spaß zu machen.



Bild 21: Die Muster in den Wänden von dem Riesenfeuerzeug haben Jonas und Rose mit einem Blechknabber (auch Blechnibbler genannt) herausgeschnitten. Der Nibbler schneidet nicht, sondern stanzt winzige runde Löcher in das Blech, was schneller und leiser ist als eine Stichsäge. Grunde geht es darum, die Fantasie der Besucher anzuregen, indem sie selbst herausfinden müssen, wie die Installation gestartet wird. "Es ist so unglaublich spaßig, wenn Menschen unsicher sind, wie eine Interaktion funktioniert, die kein Knopf ist. Vielleicht müssen sie mit dem Körper arbeiten. Oder etwas einsetzen, das sie mit sich bringen. Jetzt wollen wir, dass sie zum Beispiel ihr Feuerzeug benutzen. Sonst hätten wir wie die meisten anderen einfach einen großen Druckknopf verwenden können", sagt Jonas Johansson.

Für ihn war die größte Hürde nicht die Elektronik, sondern die Wetterverhältnisse vor Ort. "Wie baue ich etwas auf, das Regen, Hitze und Windstärke 9 standhalten kann? Die ganze Konstruktion wurde in einer Industriehalle mit flachem Boden aufgebaut, aber jetzt sind wir in der Wüste, wo der Boden eine deutliche Neigung hat."

Das insgesamt vier Meter hohe Riesenfeuerzeug wurde aus einem Stahlrahmen gebaut und mit Zwei-Millimeter-Aluplatten verkleidet. Mit einer Grundfläche von zwei Quadratmetern





wird es ordentlich vom Wind belastet. Damit es nicht umkippt, wurden mehrere 30 Zentimeter lange Bolzen in den Boden geschraubt. Der sehr lehmige Boden bietet Erdschrauben einen sehr guten Halt. Sicherheitshalber wird die Konstruktion zusätzlich von Stahlseilen gehalten. Die an feurige Flammen erinnernden Muster in den Seiten des Feuerzeugs wurden freihändig mit einem Blechknabber hergestellt (Bild 21). An der Innenseite sind zwei größere Lampen angebracht sowie mehrere Streifen mit WS2815-LEDs, die ein pulsierendes Licht abgeben, wenn das große Feuer zündet. Die LED-Streifen wurden zudem in weiße Silikonschläuche gesteckt, damit das Licht besser diffundiert (Bild 22).

Vorsichtig mit Feuer spielen

Da es das erste Mal ist, dass Jonas Johansson mit Propan arbeitet, hat er sich gegen den ursprünglichen Plan, eine eigene Feuermaschine zu bauen, entschieden und stattdessen eine kommerzielle von der Firma Magix FX gemietet. "Stage Flame" wird als 2-in-1-Flammenwerfer beworben und kann aus einer Fünf-Liter-Propanflasche über 1000 Mal Feuer bis zu sechs Meter in die Höhe spucken. Zu Hause in Schweden hat er sich ein billigeres Gerät zugelegt, um damit experimentieren zu können. Online gibt es viele Angebote für Feuermaschinen schon ab etwa 100 Euro. Dazu benötigt man noch eine Propangasflasche, einen Druckminderer, Propanschläuche und auch Stromkabel für die Feuermaschine. Jonas Johansson empfiehlt robuste und verriegelbare Steckverbinder von PowerCON, um ärgerliche Kurzschlüsse wegen Wetterbeeinflussungen auszuschließen.

Feuer in Installationen kann sehr beeindruckend sein, birgt aber auch Risiken wegen der Brand- und Verletzungsgefahr. Jonas Johansson beschreibt, wie es sich für ihn anfühlte, das erste Mal mit Feuer zu spielen: "Pure Nervosität. Gas, Druck, Feuer – das ist nichts, was ich im Wohnzimmer, meiner Hauptwerkstatt, machen kann. Ich fühle mich auch nicht wohl, wenn ich vor meiner Wohnung stehe und es auf der Straße

Jonas Tipps für Feueranfänger

- » Besorge eine DMX-gesteuerte Feuermaschine und arbeite mit LPG-Propan.
- » Nutze einen einfachen DMX-Fader und finde einen ungestörten Ort zum Experimentieren.
- » Achte auf windstilles Wetter und wähle idealerweise einen Innenraum mit hoher Decke.
- » Viel Inspiration bietet das Buch "Make: Fire – The Art and Science of Working with Propane" von Tim Deagan (Link in der Kurzinfo).
- » Frag die Profis! Besonders beim Umgang mit Feuer und Gas lohnt es sich, erfahrene Personen um Rat zu bitten.
- » Kontaktiere ein Special-FX-Studio in deiner Nähe, das dir zeigt, wie sie mit Feuer arbeiten.
- » Schaut dir den Code des Firestarters auf GitHub an (Link in der Kurzinfo).
- » Auch eine Arduino-Library für DMX-Anwendungen findest du dort.



Inspiration und Wissen über den Umgang mit Feuer erhielt Jonas aus dem Buch "Make: Fire – The Art and Science of Working with Propane" von Tim Deagan.

mache." Beängstigend war es auch, als er bei einer Gelegenheit hören konnte, wie Gas zischte. Eine Verbindung war undicht und musste abgedichtet werden. Er hätte es leicht übersehen können, aber zum Glück hat Propan einen starken Geruch. Inzwischen fühlt sich alles weniger dramatisch an, auch weil alles über DMX gesteuert wird, ein Kommunikationsprotokoll, das stabil und vorhersehbar ist.

Ein kleines Rätsel gibt es aber noch. Als ich in der Nacht mit meinem Sturmfeuerzeug mit einer Jetflamme (Bild 23) versuchte, den Firestarter anzuzünden, passierte nichts. Mit einem herkömmlichen Feuerzeug ging es problemlos. Warum hat der IR-Sensor (Akozon IR Flame Sensor Module Detektor – Link zum Hersteller in der Kurzinfo) nichts registriert? Infrarot ist letztendlich Wärmestrahlung. Jonas Johansson hat dafür keine Erklärung: "Das ist mir völlig schleierhaft, warum es bei dieser Art von Feuerzeug nicht funktioniert. Ich vermute, dass der Schwellenwert des Sensors zu hoch ist oder dass ein so starkes Feuerzeug aus irgendeinem Grund weniger Infrarotstrahlung aussendet."

Die Redaktion freut sich, wenn Leser, die sich mit IR-Sensoren auskennen, eine plausible Erklärung liefern können!

Luxusdusche in der Wüste

Nach einem langen Tag in der Sonne gibt es nichts Besseres als eine heiße Dusche. Frederik aus Amsterdam, Mitglied des 70-köpfigen Curious-Creatures-Camps, hat die eleganteste und luxuriöseste Lösung dafür entwickelt. Er konzipierte eine Dusche, die RFID-Tags und Durchflussmesser nutzt, um den Wasserverbrauch zu kontrollieren und möglichst gering zu halten, während sie gleichzeitig ein wenig Luxusgefühl vermittelt. Hier folgt eine vereinfachte Beschreibung, die als Inspiration für alle Leser dient, die ihr Duscherlebnis kreativer gestalten möchten – sei es beim Camping, im Sportverein oder auf einem Festival.

"Die Leute sind schmutzig und wollen viel duschen. Aber wir wollen auch Energie und Wasser sparen. Also musste ich herausfinden, wie wir den Wasserdurchfluss minimieren können", erklärt Frederik, der über 200 Stunden und 3500 Euro in die Anlage investierte.

Zunächst zeigt er mir den "Maschinenraum" – einen etwa einen Meter hohen und zwei Meter langen Kasten (Bild 24). Darin stehen zwei Gasboiler, ein Sicherungskasten und ein Hydrophor. Der Hydrophor sorgt dafür, dass in allen Leitungen ein angenehmer Druck von drei Bar herrscht, wie zu Hause. Die Gasboiler sind an normale LPG-Propangasflaschen angeschlossen. Gas ist zwar nicht besonders umweltfreundlich, aber effizienter und zuverlässiger als Solarthermie, meint Frederik. Bei 70 Leuten im Camp muss die Versorgung zuverlässig sein. Im Sicherungskasten finden wir einen FI-Schutzschalter und einen Transformator für 12-Volt-Gleichstrom. Auch eine Entlüftungspumpe gehört zur Ausstat-



Eine kleine Oase in der Wüste. Hier stehen nicht nur zwei normale Duschen zur Verfügung, sondern auch ein Hamam-Tisch für Massagen oder eine erfrischende Abkühlung. Der Tisch aus Sperrholzplatten ist mit Epoxy beschichtet.

tung, denn der Hydrophor kann kein Wasser ansaugen. Daher muss das gesamte System bei der Inbetriebnahme mit der Entlüftungspumpe "geflutet" werden.

Aber woher kommt das Wasser?

Für die Trink- und Brauchwasserversorgung werden in allen größeren Camps auf Nowhere

"Water Cubes" eingesetzt – standardisierte 1000-Liter-IBC-Tanks, die je nach Bedarf von Tankwagen befüllt werden. Diese Gittertanks stehen auf Paletten und sind mit einem Metallgitter umgeben. Curious Creatures hat vier solcher Water Cubes, die miteinander verbunden sind (Bild 25). Von dort saugt das System Wasser in den Maschinenraum, von wo aus alle Bereiche des Camps mit "Leitungswasser" versorgt werden. Dazu zählen



Bild 24: Im "Maschinenraum" befinden sich zwei Gasboiler, ein Hydrophor, ein Sicherungskasten und eine Entlüftungspumpe.

die Küche, die Bar, der sogenannte Playspace (wo man sich zum Sex trifft und sich gerne vorher und nachher die Hände abwaschen möchte) und natürlich die Dusch-und-Spa-Anlage.

Warmduscher in der Wüste

Als ich mit Frederik in der kombinierten Dusch-und-Spa-Anlage stehe, wird sofort klar, wie viel Arbeit hier drin steckt. Der Raum, liebevoll mit Blumen und Schilfwänden gestaltet, bietet zwei großzügige Duschplätze und einen Hamam-Tisch. Sowohl die Duschwanne als auch der Hamam-Tisch sind bunt bemalt. Sie bestehen aus Sperrholzplatten und wurden mit mehreren Litern Epoxidharz wasserdicht gemacht. Die beiden Duschen und der Hamam-Tisch verfügen jeweils über eine eigene Steuereinheit, die in einer wasserdichten Acrylglasbox untergebracht ist (Bild 26). Diese besteht aus einem Arduino Nano und einem RFID-Reader (RC522).

Wie funktioniert die Dusche? Ganz einfach: Jedes Campmitglied erhält einen RFID-Tag in Form einer kleinen Scheibe, die als Anhänger getragen werden kann (Bild 27). Zum Duschen scannt man den Tag. Dabei leuchtet eine Reihe LEDs auf, wobei jede LED für eine bestimmte Menge Wasser steht (Bild 28).

"Wenn man mit dem Duschen beginnt, sinkt die Anzeige langsam, sodass man sieht, wie viel Duschwasser noch übrig ist. Es handelt sich also nicht um einen Timer, sondern um eine Durchflussanzeige. Man sieht, wie viele Liter man verbraucht." Pro Person und Duschvorgang stehen zwölf Liter Wasser zur Verfü-


Bild 25: Die Duschanlage erhält ihr Wasser aus den "Water Cubes".



Bild 26: Die Steuereinheit für die Dusche, ein Arduino Nano mit RFID-Reader, misst den Wasserverbrauch bei jedem Duschvorgang. Nach zwölf Litern endet der Vorgang.

gung. Es geht darum, Wasser zu sparen. Da nur der tatsächliche Verbrauch zählt, kann man sich entspannt einseifen und anschließend abspülen, ohne sich wie auf manchen Campingplätzen um die verbleibende Zeit sorgen zu müssen.

Ein Grund, warum Frederik sich für den Arduino Nano entschieden hat, ist seine kompakte Größe und der günstige Preis – er lässt sich leicht austauschen, wenn etwas kaputtgeht, denn der Arduino ist nicht gelötet, sondern auf einen Adapter gesteckt. "Ich habe vorprogrammierte Ersatzteile. Falls ein Arduino kaputtgeht, nehme ich ihn einfach raus und stecke einen neuen ein. Es muss nichts gelötet werden. So kann ich das gesamte System innerhalb von 10 Minuten neu aufsetzen."

Auf der Rückseite des Kastens mit dem Arduino Nano sind vier Gartenschläuche angeschlossen. Warum so viele? Sie führen kaltes und warmes Wasser aus dem Maschinenraum direkt zur Armatur. Der Ausgang läuft jedoch nicht wie üblich direkt zum Duschkopf, sondern in einen weiteren Technikschrank an der Rückseite der Spa-Anlage (Bild 29). Dort finden wir die Durchflusssensoren, die das fertig gemischte Warmwasser, zählen". Danach geht es über die vierte Leitung zurück zum Duschkopf. In diesem Technikschrank befinden sich auch die elektrisch gesteuerten Ventile, die den Wasserzufluss auf Signal der Arduinos stoppen (Bild 30).

Grauwasserentsorgung

Die letzte wichtige Komponente ist das Grauwassersystem. An einem warmen, schwitzigen Tag verbrauchen die Campbewohner mindestens 1500 Liter Wasser. Das bedeutet auch 1500 Liter Grauwasser, das entsorgt werden muss. Die Lösung ist eine große Kunststoffwanne, in die das Abwasser eingeleitet wird (Bild 31). Ein herausziehbares Gitter in der Mitte der Wanne dient als Grobfilter. Sobald die Wanne voll ist, startet eine Tauchpumpe und pumpt das grob gefilterte Grauwasser durch eine 150 Meter lange Leitung, die in einem



Bild 27: RFID-Tags schalten die Duschen frei.

Graben außerhalb des Geländes mündet. Da es direkt in der Natur versickert, achten die Campbewohner darauf, nur biologisch abbaubare Seifen und Shampoos zu verwenden.



Bild 28: Wenn die Dusche aktiviert wird, leuchtet eine Reihe von LEDs auf. Jede LED zeigt eine bestimmte Menge verfügbares Wasser an.

Pläne für die Zukunft

Wie viele Projekte hier auf Nowhere wird auch die Dusche nie ganz fertig. Es handelt sich um einen iterativen Prozess, bei dem Jahr für Jahr umgebaut und verbessert wird. Frederik arbeitet im Alltag als Ingenieur, aber ein wichtiger Grund für ihn, die Dusche zu bauen, war, Neues zu lernen. "Ich habe viele neue Fähigkeiten erlernt und musste einiges herausfinden, zum Beispiel, wie man ein bisschen programmiert. Dabei hat mir meine Freundin Anne auch geholfen."

Entstanden ist das Ganze im Maker-Space "Acta" in Amsterdam. "Es ist auch eine soziale Sache, dass man zusammenarbeitet, um Hilfe bitten kann und neue Dinge und Techniken lernt. Es war insgesamt eine sehr gute Erfahrung, es zu bauen", meint Frederik.

Für die Zukunft plant er, die Dusche interaktiver zu gestalten. Derzeit speichert der Arduino die Daten der Duschvorgänge nicht, denn nach jeder Dusche startet er neu. "Für das nächste Mal wollen wir alle Duschdaten in einem separaten Speicher ablegen und alles aufzeichnen. Dann wollen wir noch mehr Sensoren anschließen, um weitere Daten auszulesen, wie den Energieverbrauch und den Wasserverbrauch pro Dusche. So können wir positives Feedback geben: "Hey, du verbrauchst so und so viel. Oder ihr duscht zu viel. Ihr könnt aktiv dazu beitragen, dass wir weniger Wasser verbrauchen."" —mch



Bild 29: Im Verteilerschrank laufen alle Leitungen zusammen. Normale Gartenschläuche mit GEKA-Kupplungen dienen als Wasserleitungen. Der Schrank misst die verbrauchte Wassermenge.



Bild 30: Hier werden die jeweiligen Verbraucher per Magnetventil ein- oder ausgeschaltet.



Bild 31: Grauwasser fließt über die roten Schläuche hier ein, wird gefiltert und abgepumpt.

Schematische Funktionsweise

1. Die Dusche wird mit einem RFID-Tag aktiviert.

2. Der "Maschinenraum" leitet Kalt- und Warmwasser zur Duscharmatur, wo es gemischt wird.

3. Ein blauer LED-Balken zeigt an, wie viele Liter Wasser noch verfügbar sind.

4. Ein Durchflusssensor misst die Menge des verwendeten Warmwassers und übermittelt das Ergebnis an den Arduino Nano.

5. Nachdem zwölf Liter aufgebraucht sind, sendet der Arduino ein Signal an ein elektrisches Ventil, das den Wasserdurchfluss stoppt.

Eine ausführliche Beschreibung mit vielen Fotos des "Maschinenraums" findet ihr auf unserer Homepage (Link in der Kurzinfo).

Ihre IT-Feuerwehr im Ernstfall

Mit iX gut gerüstet bei Cyberangriffen



Dieses Special der iX-Redaktion gibt Ihnen Hilfestellung, um sich bestmöglich auf Cyberangriffe vorzubereiten. Je besser Sie sich technisch und organisatorisch für den Ernstfall wappnen, umso schneller und meist mit weniger gravierenden Auswirkungen können Sie in den Normalbetrieb zurückkehren.

iX hilft Ihnen bei diesen Themen:

- Wie Cyberkriminelle agieren: Die häufigsten Angriffsarten
- Incident Response: Technische und organisatorische Sofortmaßnahmen
- Angriffe erkennen mit Logs, EDR und SIEM
- Folgen bewältigen, Systeme und AD wieder aufsetzen
- Von Backup bis Notfallplan: Vorbereitet sein für den Ernstfall
- Meldepflichten und andere rechtliche Anforderungen

Jetzt bestellen

Heft für 19,90 € • PDF für 17,49 € • Heft + PDF 26,50 €

🚽 shop.heise.de/ix-notfallguide24



ESP32-Platine für Schrittmotoren

Das PD-Stepper-Board ist ein ESP32-Board mit eingebautem Motortreiber, mit dem Schrittmotoren bis zu 50 Watt über USB-C betrieben werden können.

von Daniel Schwabe



in neuseeländischer Maker mit Namen "joshr120" hat eine Platine entwickelt, um Smart-Home- und IoT-Projekte mit Schrittmotoren in einem möglichst kleinen Formfaktor umsetzen zu können. Die Platine hat die Außenmaße eines Nema-17-Steppers und soll auf dessen Rückseite montiert werden können.

Auf dem 42 × 42 mm großen Board befindet sich ein ESP32-S3 mit WLAN und Bluetooth. Über einen JST-PH-Stecker mit 4 Pins lässt sich ein bipolarer Schrittmotor an die Platine anschließen. Dieser kann vom ESP32 mithilfe eines Trinamic TMC2209-Motortreibers fast lautlos betrieben werden. Mithilfe eines AS5600-Dreh-Encoders auf der Rückseite der Platine lässt sich die Zahl der Umdrehungen des Motors genau bestimmen und vom ESP32 auslesen – vorausgesetzt, man versieht die Stepper-Welle mit einem passenden Magneten.

Die Stromversorgung von Mikrocontroller und Motor erfolgt über eine einzelne USB-C-Buchse. Das Board unterstützt USB-Power-Delivery und kann so mit kompatiblen Stromversorgungen aushandeln, welche Spannungen und maximalen Ströme fließen sollen. So lassen sich Motoren bis zu 50 Watt verwenden. Mit einem DC/DC-Konverter werden die 3,3 V für den ESP32 erzeugt.

Zusätzlich zum Motor- und Stromanschluss verfügt das Board über verschiedene Erweiterungsports. Dazu gehören ein Qwiic/Stemma-QT-Port, mit dem man die I²C-Schnittstelle des ESP nutzen kann, und ein AUX-Anschluss, mit dem man ebenfalls Zugriff auf I²C, aber auch auf UART und Serial hat.

Auf seinem YouTube-Kanal hat "joshr120" ein Video mit Eindrücken zum Projekt veröffentlicht und auf der GitHub-Seite des Projektes, wo sich Gerber-Dateien, Schaltpläne und STL-Dateien für Gehäuse befinden, hat "joshr120" auch einige Softwarebeispiele für Anwendungsfälle und generelle Funktionen bereitgestellt.

Dort zeigt er, wie sich beispielsweise in Kombination mit ESPHome eine einfach zu verkabelnde Lösung für die Elektrifizierung von Jalousien umsetzen lässt. Dadurch hat die Motorlösung einen kleinen Formfaktor, für den kein extra Gehäuse designt werden muss.

Weitere Beispiele zeigen, wie man die Lösung über eine Webseite bedient.

Man kann das Board mit Zubehör auf der Shopseite von "Joshr120" kaufen. Dort stehen zwei Kits zur Verfügung. Das eine enthält ein Gehäuse zusätzlich zum Board und zu den passenden Kabeln und Magneten für den Dreh-Encoder. Das andere Kit enthält kein Gehäuse. Ein Motor ist bei beiden Kits nicht enthalten. Mit Gehäuse kostet das Kit 62,95 Euro und ohne 48,95 Euro. Einen Link zum Shop und der Projektseite findet ihr hinter der Short-URL: —das

make-magazin.de/xdk7



Die einzelnen Bauteile der Platine



Beispiel für die Verwendung des Boards. Ein Kabel übernimmt die Stromversorgung.



Die Platine mit Gehäuse verbaut am Motor

Interaktives schwebendes Display

Eine Benutzeroberfläche, die man in der Luft bedienen kann, klingt nach Science-Fiction, lässt sich aber ziemlich leicht selbst bauen.

von Markus Mierse



S chwebende Displays (Aerial Displays), die Informationen in der Luft anzeigen, gibt es schon seit vielen Jahren und sie werden unter anderem zur Steuerung von Aufzügen, in Empfangsbereichen oder zur PIN-Eingabe eingesetzt. Sie nutzen eine inkohärente Lichtquelle (z. B. ein Display) und passive optische Komponenten, um die Lichtstrahlen auf eine bestimmte Fokusebene zu bündeln.

Da das Prinzip recht einfach ist, habe ich mir selbst so eine schwebende Anzeige gebaut und dazu drei optische Hauptkomponenten verwendet: ein helles Display, einen halbtransparenten Spiegel und eine retroreflektierende Folie. Die wichtigste Komponente ist der sogenannte Retroreflektor, den man vom Rückstrahler beim Fahrrad kennt. Er reflektiert – wie der Name vermuten lässt – das Licht zurück zur Quelle, unabhängig davon, aus welcher Richtung es kommt.

In meinem Aufbau wird das vom Display ausgestrahlte Licht vom halbtransparenten Spiegel (Strahlenleiter) nach unten zum Retroreflektor umgeleitet. Dieser wirft das Licht in dieselbe Richtung (!) wieder zurück, sodass es den Strahlenteiler passiert und dann auf einer Ebene außerhalb des Geräts (als schwebende Anzeige) gebündelt wird.

Um mit meinem schwebenden Display interagieren zu können, musste ich einen Weg finden, die Position eines Fingers in der Luft zu erkennen, der das Display "berührt". Für meine DIY-Version brauchte ich nur eine einfache Interaktion, d. h. ein paar definierte Bereiche, die bei "Kontakt" eine Aktion auslösen. Ich verwende dafür drei ToF-Sensoren (Time of Flight) vom Typ VL53LOX von STMicroelectronics. Sie verfügen über eine Laserdiode und einen Empfänger im selben Gehäuse.

Für mein System brauchte ich drei Sensoren, um das Erkennungsfeld abzudecken. Insgesamt können so neun Berührungsfelder (3×3) zuverlässig erkannt werden. Ein Arduino Nano liest die Sensordaten aus und sendet alle Entfernungsmessungen über die serielle Schnittstelle.

Um die Sensordaten zu visualisieren und die richtigen Bereiche zu finden, habe ich ein Testprogramm auf dem PC geschrieben. Es empfängt die Daten über UART (COM-Port) und zeigt die Entfernung jedes Sensors als farbigen Punkt an. So kann ich überprüfen, ob sich eine Fingerposition im gewünschten Bereich genau messen lässt.

Schließlich kommt zur Steuerung und Anzeige von Bitmaps und Animationen ein Lattepanda Delta als Hauptrechner zum Einsatz. Da es sich im Wesentlichen um einen kleinen Windows-Computer handelt, war die Entwicklung beguem als Windows-Applikation möglich.

Eine Anleitung und weitere Informationen zum Projekt findet ihr über den Link. —*akf*



Der Aufbau der schwebenden Anzeige besteht aus diesen drei Komponenten.



Damit man in der Luft mit der Anzeige interagieren kann, verwende ich drei ToF-Sensoren.



Gut für die Privatsphäre: Personen, die neben der Anzeige stehen, können nicht sehen, was man macht.

Elektromechanischer Handheld

Dieses Spielgerät lockt heute kaum noch Kids vom Sofa. Seine Erfinder hat es aber sicher mal um den Schlaf gebracht. Ein Blick ins Innere zeigt, wieso.

von Ákos Fodor



Die tragbare Spielekonsole Terra-Hit von TOMY aus dem Jahr 1979 könnte optisch direkt einem Alien-Film entstammen. Thematisch passend jagt man auf einer Spielfläche von 95 × 80 mm kleine Außerirdische in Space-Invaders-Manier (Bild 1). Allerdings nicht als Sprites auf einem elektronischen Display, sondern rein elektromechanisch. Zu den beweglichen Spielelementen gehören zwei Reihen fliegender Aliens, ein Raumschiff, das man selbst entlang der x-Achse lenken kann, und natürlich eine Rakete, mit der man auf die Aliens schießt. Diese fliegt leuchtend aufs Ziel zu und lässt es explodieren. Ein Timer und ein Punktezähler sorgen zusätzlich für Spannung.

Wer das Reingeschaut zu TOMYs Tischflipper gelesen hat (Make 4/23), kann sich vielleicht schon denken, wie viele Motoren die Entwickler in dem Spielgerät verbaut haben. Genau: einen einzigen (Bild 2). Mit ihm verbunden ist eine beeindruckend komplexe Mechanik, die auch das Lichtspiel und die Kollisionsabfrage clever löst. Einzig Musik fehlt. Dafür gibt es Krach und ... noch mehr Krach über eine Ratsche, sobald man einen Treffer erzielt hat.

Ein Film voller Aliens

Die Außerirdischen sind auf zwei transparente Bänder gedruckt, die sich mithilfe von Walzen in unterschiedliche Richtungen bewegen. Diese Bänder sind im Grunde die Programmierung, denn anhand von Perforationen "erkennt" das Spiel, ob man ein Alien getroffen hat oder nicht.

Rakete fliegt ...

Dafür muss man natürlich erst mal seine Rakete abfeuern, die TOMY (wie auch die restliche Mechanik) geschickt unter einer dunkelvioletten Scheibe versteckt hat. So erscheint sie erst, wenn sie leuchtend auf ein Alien zusteuert. Den Raketenflug übernehmen zwei



Bild 1: Spaß für unterwegs aus dem Jahr 1979

Schnecken, die sich entgegengesetzt drehen: Dabei zieht die erste die Rakete in Richtung Ziel und die zweite holt sie wieder zum Startpunkt zurück.

Der Richtungswechsel funktioniert mithilfe eines beweglichen Elements auf der Unterseite der Rakete, das bei Start und Ziel die Ausrichtung ändert (Bild 3) – so ähnlich, wie man es beim Drückmechanismus eines Kugelschreibers beobachten kann. Auf dem Hinflug berührt das bewegliche Element zudem eine Schiene und aktiviert so die Leuchte in der Rakete. Damit man sich das besser vorstellen kann, habe ich ein kurzes Video zusammengeschnitten und verlinkt.

... und explodiert

Sobald die Rakete die Aliens erreicht, trifft sie auf eine kleine bewegliche Box mit einem transparent-roten Deckel (ebenfalls unsichtbar unter der dunklen Scheibe) und schiebt diese vor sich her (Bild 4). Passiert diese Box die Aliens, schließt sich ein Teil des Kollisionsschaltkreises über einen Kontakt.

Der zweite Trigger für einen Treffer befindet sich auf der Rückseite der Bandkonstruktion (Bild 5). Dort berühren zwei Taster durch die



Bild 2: Diese Maschinerie steuert alles mit nur einem Motor, aber dafür vielen Getrieben.



Bild 3: Der Flügel unter der Rakete steuert, in welche Richtung sie sich bewegt.

Perforationen in den Bändern eine Metallplatte, sofern (von vorn gesehen) die Rakete ein Alien erwischt hat, und schließen so den vollständigen Schaltkreis.

Daraufhin gibt es eine kleine Lichtexplosion in der Box, die mithilfe eines beweglichen Hakens an Ort und Stelle verharrt und den Spieler so lange ablenkt, bis die Rakete wieder auf ihre Startposition zurückgekehrt ist. Erst dann schnappt die rote Box wieder an ihren Ursprung zurück. Ganz schön raffiniert. — akf

make-magazin.de/x678



Bild 4: Die rot-weiße Box schließt den ersten Teil des Kollisionsschaltkreises über die Kontakte rechts daneben, während sie von der Rakete nach oben geschoben wird.



Bild 5: Wenn gleichzeitig ein Taster durch ein längliches Loch die Platte auf der Rückseite berührt, schließt sich der Schaltkreis für die Kollisionsabfrage komplett.



Siebdruck mit Leim und Laser

Klassischer Siebdruck erzeugt professionell aussehende T-Shirt-Drucke, ist aber recht aufwendig. Die gleichen Ergebnisse lassen sich statt mit Chemikalien und Dunkelkammer aber auch mit wasserfestem Holzleim und einem Laser erzielen. Wir zeigen, wie es geht.

von Johannes Börnsen

al eben die spontane Motividee mit Siebdruck auf ein T-Shirt bannen? Das werden wohl die wenigsten machen, der Aufwand ist zu groß: Beim klassischen Siebdruck wird ein mit einem Sieb bespannter Rahmen unter Ausschluss von Tageslicht mit einer lichtempfindlichen Beschichtung versehen, auf die anschließend das Motiv belichtet wird. Dann muss der Rahmen ausgewaschen werden, wodurch sich unbelichtete Bereiche des Motivs lösen und ein druckfähiges Motiv auf dem Rahmen erscheint.

Wenn ihr einen Laser zu Hause habt, lässt sich dieser Prozess aber deutlich vereinfachen. Heraus kommen selbst gedruckte Shirts, die von "richtigem" Siebdruck nicht zu unterscheiden sind.

Rahmenbau

Handelsübliche Siebdruckrahmen sind mit einem Kunststoffgewebe bespannt. Das würde beim Lasern des Motivs Schaden nehmen, weswegen wir uns einen eigenen Siebdruckrahmen mit einem Edelstahlgewebe bauen (Bezugsquelle siehe Link in der Kurzinfo). Den Rahmen haben wir aus Multiplexleisten gefertigt, die wir an den Ecken jeweils zur Hälfte ausgeklinkt und verleimt haben. Natürlich kann man den Rahmen aber auch einfach verschrauben oder einen alten Bilderrahmen in passender Größe verwenden.

Das Edelstahlgewebe mit der Maschenweite 120 (mesh 120, entspricht einem klassischen Siebdruckrahmen mit einem T47-Sieb) wird mit leichter Spannung auf die Unterseite des Rahmens getackert und die offenen Siebkanten decken wir mit einem Klebeband ab. Tipp: Das Gewebe lässt sich problemlos mit einer alten Schere schneiden. Tragt dabei Handschuhe, an den Kanten des Gewebes kann man sich nämlich unangenehme Splitter holen.

Kurzinfo

» Siebdruckrahmen günstig selber bauen
 » Beschichten mit Holzleim statt Foto-Emulsion
 » Motiv freilasern und drucken



Beim Siebdrucken wird die Druckfarbe mit einem Rakel durch die Öffnungen des Siebes gedrückt. Damit dabei auch ein Motiv entsteht, müssen jene Teile des Siebes verschlossen sein, an denen keine Farbe auf das Shirt gelangen soll. Die feinen Öffnungen des Siebes werden also quasi teilweise verklebt. Dazu verschließen wir zunächst das komplette Sieb mit Weißleim. Wir haben wasserfesten Holzleim von Ponal verwendet. Er wird zunächst mit einem Kunststoff-Japan-Spachtel satt auf das Sieb aufgetragen und anschließend wieder abgekratzt. Der Leim bleibt dann nur in den Öffnungen des Siebes hängen und ergibt eine gleichmäßig geschlossene Fläche. Legt den Rahmen nun für einige Stunden in die Sonne, sodass der Leim gut durchtrocknet.

Die Suche nach dem Motiv

Nun erstellen wir am PC die zukünftige T-Shirt-Grafik in einer Software der Wahl oder suchen uns ein passendes Motiv aus dem Netz. Am besten eignen sich Motive mit klaren Grenzen von Schwarz und Weiß. Bilder mit Graustufen müssen zunächst in ein Halbtonraster umgewandelt werden. Eine Anleitung dazu findet ihr hinter dem Link in der Kurzinfo. Den werkelnden Maker aus unserem Beispiel haben wir übrigens mit Dall-E 2 ge-



Ein simpler Rahmen aus Multiplexstreifen, bespannt mit einem Edelstahlgewebe. Fertig fürs Beschichten!



Mit wasserfestem Holzleim werden die Öffnungen des Siebes verschlossen. Der Kunststoffspachtel schont das Gewebe und dient später auch als Rakel zum Drucken.



Dieser Siebdruckrahmen ist druckreif. Das Motiv hat der Laser freigelegt. In den dunklen Bereichen ist das Sieb also wieder für die Druckfarbe durchlässig.



Der fertige Siebdruck auf einem dunkelgrauen Shirt ist von anderen Siebdrucken nicht zu unterscheiden. Das Motiv ist übrigens KI-generiert. Falls ihr es nachdrucken wollt, folgt dem Link in der Kurzinfo.

neriert und etwas nachbearbeitet. Der Prompt lautete "linocut of a woodworking nerd".

Nun können wir das Motiv auf das Sieb lasern und dadurch das Sieb teilweise wieder öffnen. Dabei werden alle schwarzen Bereiche



Wenn die Farbe mit dem Spachtel über das Sieb verteilt wurde, kann sie im zweiten Durchgang mit moderatem Druck durch das Sieb hindurch auf den unter dem Rahmen liegenden Stoff gedrückt werden.

des Motivs so in den Leim graviert, dass dieser vollständig entfernt wird. Die Einstellungen für den Laser hängen vom jeweiligen Gerät und dessen Leistung ab. Wir haben einen 55-Watt-CO₂-Laser bei 30 Prozent Leistung und 250 mm/sec verwendet. Achtet auf eure Sicherheit, gute Belüftung und filtert die abgesaugte Luft.

Wenn ihr das Motiv von der Unterseite in die Leimschicht lasert, müsst ihr das Motiv spiegeln, damit es nachher richtig herum auf dem T-Shirt landet. Wenn alles geklappt hat, habt ihr nun einen druckbereiten Siebdruckrahmen.

Ordentlich Druck machen

Ab hier funktioniert alles genau wie beim klassischen Siebdrucken. Die Stofffarbe wird zunächst im oberen Bereich des Siebes aufgetragen. Dann wird das Sieb mit einem Rakel (wir haben wieder den Kunststoff-Japan-Spachtel verwendet) zunächst mit der Farbe geflutet, also ohne viel Druck über das Sieb gezogen, sodass überall eine dünne Schicht Farbe ist. Dann wird der Rahmen auf das Shirt gelegt und die Farbe mit dem Rakel mit kräftigem Druck durch das Sieb auf den Stoff gedrückt. Nehmt den Rahmen vorsichtig vom Shirt (Achtung, es klebt etwas fest, am besten gelingt dies zu zweit). Die noch nasse Farbe muss nun trocknen und anschließend nach Angaben des Herstellers fixiert werden, meist durch das Bügeln von der Rückseite.

Tipps und Tricks

Achtet auf die Hinweise des Siebdruckfarben-Herstellers, in der Regel müssen die Shirts vor dem Druck gewaschen werden. Wenn ihr mehrere Drucke vom selben Motiv machen möchtet, könnt ihr das Sieb direkt auf das nächste Shirt legen und erneut fluten und drucken. Ab etwa zehn Drucken am Stück ist es empfehlenswert, die Druckfarbe vorher mit einem Trocknungsverzögerer zu mischen, damit die Farbe auf dem Sieb nicht zu schnell eintrocknet.

Das Sieb könnt ihr nach dem Druck einfach unter fließendem Wasser auswaschen, am besten mit einem Brausekopf. Allzu stark und lange sollte man aber nicht waschen, auch der wasserfeste Leim kann sich sonst wieder anlösen.

Wenn ihr den Rahmen für ein neues Motiv nutzen möchtet, könnt ihr das Sieb mit dem Laser und einem großen, schwarzen Rechteck einmal komplett von Leim und Farbresten befreien und anschließend neu mit Leim beschichten. —jom



Beliebt auf heise +







Dynamische Stromtarife werden immer beliebter, müssen aber überwacht werden. Unser Preisrahmen stellt die Preise übersichtlich dar und sieht ansprechend aus.

Ikea-Sandmaltisch Bauanleitung, Teil 1: Material und Zusammenbau

Nach dem Erfahrungsbericht über den Bau des Sandmaltisches wollten viele Leser mehr darüber wissen und der Ruf nach einer kompletten Bauanleitung wurde laut.

26.07.2024 Q 9 | Make Magazin



Ein Infotext, ein paar animierte Grafiken oder eine Laufschrift auf einer LED-Matrix sind nicht nur praktisch, sondern machen auch optisch was her.

LED-Matrizes mit MicroPython steuern, Teil 2

29.07.2024 Make Magazin

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

Universeller Staubsauger-Adapter für die Werkstatt!

Wer mit Elektrowerkzeugen arbeitet, die eine Absaugung benötigen, kennt das Problem: Jedes Gerät hat einen anderen Anschluss für die Absaugung. In diesem YouTube-Video stellt Johannes drei Projekte von der Maker Faire Hannover vor, unter anderem die Lösung eines Absaugadapters osVAC neo. Neben diesem Projekt gibt es noch ein gemeinschaftsorientiertes Projekt: ein selbst gebauter Camper aus Holz von der gemeinnützigen VolksSchrauber-Akademie. Basierend auf nachhaltigen Materialien und einem Konzept, bei dem die Nutzung durch ein Chartermodell sichergestellt wird. Das dritte Projekt

bzw. der dritte Maker Daniel Springwald baut realitätsgetreue animatronische Figuren. Von diversen Figuren aus dem Star-Wars-Universum bis hin zu Handpuppen, die sich realistisch bewegen. —dus

www.youtube.com/@MakeMagazinDE

Weitere aktuelle Videos:







Bleib informiert:

www.make-magazin.de

@makemagazinde





X (Twitter): @MakeMagazinDE







Bluesky: @makemagazin.de



WhatsApp Channel: Make Magazin Deutschland



GitHub: MakeMagazinDE



Threads: @makemagazinde



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



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



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



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



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

Sandbilder erstellen Sandmaltisch, Teil 2

Einer rollenden Kugel im Sand zuzuschauen, ist mystisch und entspannend zugleich. Den dadurch entstehenden Sandbildern kann sich der Betrachter kaum entziehen. Unser Autor erklärt in diesem zweiten Teil unseres Sandmaltisch-Projekts, wie mit viel kreativer Inspiration spielend einfach neue Sandbilder entstehen.

von Benno Lottenbach



em Rollen einer Kugel folgen und dabei beobachten, wie deren Spur faszinierende Bilder in den Sand malt. Träumen, entspannen und dabei einen köstlichen Tee oder Kaffee genießen. Die in der vorigen Ausgabe der Make vorgestellte Bauanleitung für einen Sandmaltisch ermöglicht genau dies mitten im Wohnzimmer und lädt Jung und Alt zum Verweilen ein. Ich berichtete darüber, wie ein IKEA-Lack-Couchtisch mithilfe eines 3D-Druckers, ein paar gängigen Holzwerkzeugen und einiger weiterer Komponenten zu einem Sandmaltisch umgebaut werden kann. Der Nachbau dieses beeindruckenden Möbelstücks dürfte wohl einige Maker für eine Weile beschäftigt haben. Und wer das fertige Projekt bereits im Wohnzimmer stehen und die ersten Sandbilder betrachtet hat, wartet nun bestimmt auf Nachschub in Form von weiteren Sandbildern. Genau hier knüpft dieser Artikel an.

Im vorliegenden Artikel erkläre ich euch, wie ihr Sandbilder selber entwickeln könnt. Dabei verrate ich einige Tipps und Tricks, um lauernden Hürden und Stolpersteinen auszuweichen, und ich helfe euch dabei, mit viel kreativer Inspiration beeindruckende Kunstwerke zu gestalten. Quasi als Bonus zeige ich euch am Ende des Artikels, wie die entstandenen Bilder miteinander verkettet werden können, damit diese ähnlich einer Playlist automatisch nacheinander abgespielt werden. Damit überrascht der Sandmaltisch Familie und Gäste mit wechselnden Bildern und fasziniert immer wieder von Neuem.

Von Pfaden und Koordinaten

Um die Funktionsweise des Sandmaltisches und die Entstehung dieser Bilder besser verstehen zu können, stellt man sich am besten ein gängiges kariertes Notizpapier vor. Die Linien auf dem Papier bilden ein kartesisches Koordinatensystem, das wohl jedem Maker noch aus der Schulzeit bekannt sein dürfte. Dieses Koordinatensystem hat in unserem Fall zwei Achsen, die als X- und Y-Achse bezeichnet werden. Anhand dieses Koordinatensystems ist es möglich, jeden Punkt auf dem Notizpapier durch Angabe eines X- und eines Y-Abstands zu definieren. Dieses kartesischen Koordinatensystems bedienen sich nicht nur Computergrafik und Physik, sondern auch 3D-Drucker, Lasercutter und ähnliche CNC-Maschinen.

Die meisten Sandbilder bestehen aus einem mathematischen Muster, das auf dieses Koordinatensystem projiziert wird. Eine geometrische zweidimensionale Form, nehmen wir als Beispiel ein Quadrat, dient dabei als Grundlage für das Bild. Dieses Quadrat wird als Pfad beschrieben. Im Falle des Quadrats wären das also, vereinfacht ausgedrückt, vier Linien, die von Ecke zu Ecke des Quadrats führen, wobei jede Ecke als Punkt im Koordinatensystem abgebildet wird. Für ein Sandbild wird nun diese einfache Grundform mehrfach dupliziert. Dabei wird die Größe oder die Lage der Form mit jedem Duplikat ein klein wenig verändert. Wird dies oft genug durchgeführt, entsteht dadurch ein geometrisches Muster. Diese Muster lassen sich wiederum beliebig kombinieren und übereinanderschichten, wodurch schließlich unsere Sandbilder entstehen.

Sandmalerei für Tüftler

Die Sandmalerei, wie wir sie mit unserem Tisch hervorzaubern, hält einige Gemeinheiten für uns bereit, die die ganze Sache etwas herausfordernd, aber dadurch auch spannend für Tüftler machen. Eine dieser Gemeinheiten ist, dass die Kugel bei jeder Bewegung eine Spur hinterlässt, ob gewollt oder nicht.

Ein Abheben und Absetzen des "Stifts", wie man das bei einer Zeichnung auf Papier machen würde, kennt die Kugel nicht. In diesem Sinne muss man sich vor allem beim überlagernden Muster immer wieder überlegen, welchen Weg die Kugel vom Endpunkt des vorherigen Musters zum Anfangspunkt des neuen Musters nehmen soll. Schließlich will man das soeben gemalte Bild nicht durch einen unbedachten Fahrweg zerstören. Hierzu bieten sich zwei in der Praxis bewährte Lösungskonzepte an. Die einfachste Variante: Man führt die Kugel jeweils am Bildrand, dem sogenannten Perimeter, entlang zum neuen Startpunkt. Dies setzt allerdings voraus, dass die Kugel am Ende eines Musters in Randnähe zu liegen kommt. Sonst würde sie ja bereits auf dem Weg zum Rand das gemalte Bild wieder zerstören. Die etwas schwierigere Variante: Man führt die Kugel über einen Weg zum Startpunkt des nächsten Musters, der von diesem nächsten Muster möglichst bald wieder überzeichnet wird. Die Spur wird dadurch vom nächsten Muster überdeckt und ist somit nur kurze Zeit für den Betrachter sichtbar.

Eine weitere Eigenheit der Sandspuren ist mir bis heute irgendwie mysteriös geblieben:

Kurzinfo

» Sandmaltisch Teil 2: Software
 » G-Code f
ür den Tisch mit Sandify
 » Sandbilder verkettet abspielen



Obwohl die von uns eingesetzte Kugel mit einem Durchmesser von 20 mm eine relativ breite Sandspur hinterlässt, ist es ihr möglich, sehr detaillierte Sandbilder mit feinsten Strukturen zu erzeugen. Ausgezeichnet ist dies im Titelbild dieses Artikels sichtbar. Teilweise scheint es gar so, als wären die Spuren mit der Spitze eines Zahnstochers in den Sand gezeichnet worden. Der Sand wird nämlich von der Kugel, ähnlich einer Bugwelle eines Schiffes, vor sich hergeschoben und gleichzeitig zur Seite gedrückt. Mehrere solcher um wenige Millimeter versetzten Sandspuren ergeben am Ende feinste Strukturen. Einzig die zurückliegende letzte Spur der Kugel bleibt in ihrer ganzen Spurbreite liegen, sofern sie nicht von einer anderen Spur überrollt wird. Diese Eigenschaften der Bildentstehung sind letztlich immer wieder für Überraschungen gut. Wie ein in Sand gemaltes Bild am Ende tatsächlich aussehen wird, konnte mir bisher keine Software für G-Code-Simulationen realistisch aufzeigen.

Bühne frei für Sandify

Selbstverständlich wäre es viel zu aufwendig, jede einzelne Bewegung der Kugel im Koordinatensystem selbst programmieren zu müssen. Bei einem größeren, flächendeckenden Bild kommen nämlich schnell mal ein paar Zehntausend Fahrwege zustande. Diesen Aufwand überlassen wir deshalb lieber einer Software, die diese Arbeit perfekt für uns erledigt. Der Online-Dienst Sandify.org berechnet diese Pfade nämlich anhand von Parametern,

default machine				Rectangula	
BINGOBRICKS Sandmaltisch				Rectangula	
Name	BINGOBRICKS Sandmaltisch				
Туре	Rectangular	Rectangular			
Min x (mm)	0				
Max x (mm)	386				
Min y (mm)	0				
Max y (mm)	716				
Force origin	upper left	upper right	lower left	lower right	

Abbildung 1: Die Maschinendefinition des Sandmaltisches in Sandify



Abbildung 2: Ein Quadrat, das 90-mal wiederholt gezeichnet, stets etwas vergrößert und dabei leicht gedreht wird, ergibt unser erstes Sandbild.

die wir als Künstler und Maker bestimmen können. Das Resultat der Berechnungen gibt Sandify als G-Code-Datei aus, die die Bewegungen der Kugel im X-/Y-Koordinatensystem beschreibt. Bevor wir aber mit Sandify kreativ werden können, müssen wir als Erstes die Maschinenspezifikation unseres Sandmaltisches hinterlegen. Hierzu klicken wir in Sandify im Tab, Machines" auf das kleine Plus. Im erscheinenden Eingabeformular wählen wir den Typ "Rectangular" aus und geben unserem Sandmaltisch einen passenden Namen. Die Eingaben bestätigen wir mit "Create".

Besonders wichtig sind nun die Abmessungen der soeben erfassten Maschine, die wir gemäß Abbildung 1 hinterlegen. Diese Maße entsprechen der Zeichnungsfläche unseres Sandmaltisches, wobei ein paar Millimeter Randabstand als Reserve bereits einberechnet sind. Unter "Force origin" lässt sich einstellen, ob die Kugel am Ende eines Bildes jeweils in einer bestimmten Ecke der Bildfläche geparkt werden soll. Sandify berechnet dabei automatisch einen Weg über die Perimeter der Bildfläche zur angegebenen Ecke, um das gemalte Bild (möglichst) nicht zu zerstören. Ich empfehle, diesen Ursprungspunkt vorerst auf "lower left" zu stellen, da dort die Nullposition des Sandmaltisches liegt. Den Schalter "Minimize perimeter moves" lassen wir ausgeschaltet. Um Verwechslungen zu vermeiden, löschen wir abschließend die von Sandify bereitgestellte "default machine" durch Auswahl und anschließendem Klick auf den Papierkorb, sodass nur noch unser Sandmaltisch als Maschine in der Liste aufgeführt ist.

Kreative Geometrie

Jetzt können wir endlich unsere Kreativität walten lassen. Hierzu klicken wir (Abbildung 2) im Register "Layers" auf das kleine Plus und fügen damit einen neuen Layer zum Bild hinzu. Sandify fragt uns nun nach der Grundform ("Type"), auf welcher wir das Bild aufbauen möchten. Um beim Beispiel eines Quadrates zu bleiben, wählen wir hier "Polygon" aus, was einem Vieleck entspricht. Nach dem Klick auf "Create" wird der soeben erfasste Layer erstellt. Auf diese Weise lassen sich im Prinzip auch mehrere Layer erstellen. Wir bleiben aber der Einfachheit halber hier bei einem einzigen Layer und löschen deshalb bereits vorhandene andere Layer durch Selektierung und Klick auf den Papierkorb.

Jetzt ließe sich unser Polygon weiter spezifizieren, indem die Anzahl der Ecken, die Größe und die Lage der Grundfigur entsprechend verändert wird. Damit wird aus dem Quadrat beispielsweise auch ein Dreieck oder ein Oktagon. Die Ausgangsgröße und Lage der Grundform lässt sich übrigens auch mit der Maus direkt in der Bildvorschau verändern, statt über die numerische Eingabe. Das ist vielfach etwas komfortabler. Unter den Parametern finden wir die Option "Connect to next layer".

Sie beschreibt, welchen Weg die Kugel beim Übergang von diesem Layer, welchen wir gerade bearbeiten, zum nachfolgenden Layer nehmen soll. In der Regel ist es sicherer, den Weg über den Perimeter zu nehmen, als den Weg über die direkte Linie.

<u></u> <u></u> <u></u> <u></u> <u></u> <u></u>

G94

G17

G21

\$H

G54

Ist die Grundform des Musters erfasst, gilt es als Nächstes, einen oder mehrere Effekte einem Layer hinzuzufügen. Der wichtigste Effekt ist hierbei ohne Zweifel der Effekt namens "Loop". In den Einstellungen zu diesem Effekt wählt man, wie oft die Grundfigur dupliziert und welche Veränderung nach jedem Replikat vorgenommen werden soll. In der Regel lässt man die Grundfigur bei jedem Durchgang etwas größer zeichnen. Also wählen wir bei der Einstellung "Scale" einen beliebi-

gen positiven Wert ungefähr im Bereich von 20-80 und unter "Number of loops" eine Wiederholungszahl im Bereich von 50–100. Es gibt hier kein Richtig und kein Falsch. Man darf ruhig einfach mal drauflos probieren.

Schließlich aktivieren wir noch die Option "Spin", was dazu führt, dass die Ausrichtung der Grundform nach jedem Durchgang um die angegebene Anzahl Grad gedreht wird. Die Konsequenzen jeder Eingabe werden dabei von Sandify stets visualisiert, sodass sich erahnen lässt, wie das fertige Sandbild am Ende aussehen wird. Der Startpunkt des Bildes wird dabei grün, der Endpunkt rot markiert. Mit dem Schieberegler am unteren Bildschirmrand lässt sich zudem der Weg der Kugel in groben Schritten visualisieren.

Es würde nun zu weit führen, jeden einzelnen Lavertyp und jeden Effekt in diesem Artikel zu erklären. Am besten spielt man einfach ein wenig mit den verschiedenen Einstellungen rum und schaut, wie sich die Parameter auf das Bild auswirken. Auf einige spezielle Layertypen und Effekte möchte ich aber trotzdem kurz eingehen, da ich diese als hilfreich empfinde. Zum einen ist da der Layertyp "Point", mit welchem sich ein einzelner Punkt auf der Bildfläche anfahren lässt. Dies ist insbesondere nützlich, um die Kugel über einen Weg von definierten Kontrollpunkten an eine bestimmte Position zu fahren. Jeder Kontrollpunkt sollte dabei mit einem eigenen Layer des Typs "Point" erfasst werden. Damit dies funktioniert, muss die Option "Connect to next layer" (voreingestellt) jeweils auf "line" geschaltet sein. Die Reihenfolge der Layer in der Layerliste lässt sich übrigens durch Drag & Drop mit der Maus verändern.

Ein weiterer erwähnenswerter Layertyp ist "Wiper". Damit lässt sich die ganze Bildfläche durch das Abfahren regelmäßiger Linien aus-

G-Code Header ; File name: 'Bild1' File type: gcode BEGIN PRE (Modus: Absolute Positionierung) (Feedrate in units per minute, default) (Arbeitsbereich: Achsen auf xy, default) (Einheit: Millimeter) (Homing: Limit Schalter suchen.) (Das erste Koordinatensystem G54 anwenden.) G10 L20 P0 X0 Y0 (Offsets des ersten Koordinatensystems G54 setzen.) G1 F500 (Geschwindigkeit der G1-Bewegungen festlegen.) ; END PRE G1 X203.000 Y368.000 G1 X181.904 Y368.904 G1 X181.211 Y345.792 G1 X206.336 Y345.345

> wischen, wodurch ein Bild sozusagen gelöscht werden kann. Und schließlich gibt es noch den Effekt "Fine tuning". Mit diesem Effekt lässt sich, nebst anderen Möglichkeiten der Feinabstimmung, ein Pfad vorzeitig beenden. Das ist gut nutzbar, um ein Muster an einem geeigneten Punkt zu beenden, damit die Kugel auf möglichst kurzem Weg zum Perimeter gelangen kann. Idealerweise stoppt man die Bildentstehung genau auf dem Perimeter und kann dadurch auf sicherem Weg dem Perimeter entlang zum Startpunkt des nächsten Layers rollen.

Ist man mit der Vorschau des Bildes zufrieden, bleibt nur noch, den entsprechenden Pfad als G-Code generieren und exportieren zu lassen. Hierzu wählt man im Menü "File" den Punkt, Export as", wählt, GCode" als Dateiformat aus und gibt dem File noch einen Namen. Nach Klick auf "Download" findet man den generierten Code im Download-Ordner. Wer das erstellte Sandmalbild zu einem späteren Zeitpunkt weiterbearbeiten möchte, speichert die erstellten Layer und Effekte am besten zusätzlich als SDF-Datei über den Menüpunkt "Save as". Eine SDF-Datei lässt sich jederzeit wieder in Sandify öffnen.

Skripte nachbereiten

Schaut man sich das erstellte G-Code-File in einem Texteditor an, so erkennt man die von Sandify erstellten Bewegungskommandos, welche jeweils mit G1 beginnen. In dem in Kasten "G-Code Header" dargestellten Skript sind diese Kommandos ab Zeile 17 gelistet. Die Codes dieser Zeilen weisen den Controller an, die darauffolgend angegebene X- und Y-Koordinate anzufahren. Nun reicht das aber leider noch nicht ganz aus, um unser Sandbild in voller Pracht auf den Tisch zu kriegen. Es fehlen einige initiale Anweisungen, damit der Controller genau das macht, was wir von ihm verlangen. Daher fügen wir zu Beginn des Skripts die im Kasten ersichtlichen Befehlszeilen 5–14 ein. Alles, was auf einer Zeile in Klammern oder nach einem Strichpunkt steht, gilt als Kommentar und wird vom Controller ignoriert. Aber Achtung: Ausrufezeichen in Kommentaren werden in gewissen Fällen vom Controller dennoch interpretiert. Deshalb besser keine Ausrufezeichen in Kommentaren verwenden. Diese Anmerkungen in den Codezeilen müssen also nicht unbedingt abgetippt werden.

Zwei Zeilen im Anfangsbereich des abgebildeten G-Codes sind besonders interessant: Der Code \$H in Zeile 10 lässt die Kugel zur Home-Position fahren, also die Limit-Schalter suchen. Wenn diese Schalter einen Kontakt melden, weist die Firmware dieser Position die Nullposition zu. Die Lage der Kugel befindet sich damit in einem initialisierten Zustand, womit alle weiteren Punkte des Koordinatensystems zielgenau angesteuert werden können. Allerdings ist, wie wir ja bereits gelernt haben, auch dieser Weg der Kugel bedingungslos mit einer Sandspur verbunden. Ich habe mir trotzdem angewöhnt, im Header (fast) jeder G-Code-Datei ein solches sogenanntes Homing durchzuführen, da ich in aller Regel auch jedes Sandbild an der Home-Position beende. So kann ich beliebige Sandbilder in frei wählbarer Reihenfolge "aufeinanderschichten", und muss mich nicht um die Startund Endpositionen kümmern, da diese somit immer identisch sind. Ein Homing ist auf alle Fälle zwingend notwendig nach dem Einschalten oder Zurücksetzen des Sandmaltisches mittels Reset-Schalter. Ist das Homing also im Header jeder G-Code-Datei enthalten, braucht man sich vor dem Abspielen einer G-Code-Datei keine Gedanken dazu zu machen.

Die weitere wichtige Codezeile ist das erste G1-Kommando, welches einen F-Parameter

\$SD/Run=BINGOBRICKS/001_Wiper1.gcode ; PAUSE G4 P1 \$SD/Run=BINGOBRICKS/002_Welcome.gcode G4 P86400 ; PAUSE \$SD/Run=BINGOBRICKS/003_Bild2.gcode G4 P86400 : PAUSE \$SD/Run=BINGOBRICKS/004_Bild3.gcode \$SD/Run=BINGOBRICKS/000_Playlist.gcode

("Feed Rate") enthält, also Zeile 14. Da diese Codezeile keine X- und Y-Position enthält, bleibt die Kugel an Ort und Stelle stehen. Der F-Parameter aber bestimmt die Geschwindigkeit aller ab dieser Codezeile folgenden Bewegungen, die mit G1 ausgelöst werden. Somit lässt sich mit einer einzigen Codezeile



die Geschwindigkeit der Bewegungen für das ganze Sandmalbild festlegen. Ich persönlich mag eine langsame, achtsame Fortbewegung der Kugel und setze deshalb die Geschwindigkeit meistens auf F500. Eine höhere Geschwindigkeit, wie z. B. F3000, setze ich nur ein, wenn ich etwas ausprobieren und das Resultat möglichst bald betrachten möchte, oder bei nüchternen Löschbildern mit repetitiven Kugelfahrwegen. Die erhöhte Geschwindigkeit beeinflusst die Bildqualität übrigens in keinem oder zumindest nur sehr geringem Maß. Allerdings werden die Motorengeräusche der Stepper-Motoren bei höheren Geschwindigkeiten etwas lauter. Wer will das schon im Wohnzimmer? Die meisten Mitbewohner dürften einen annähernd lautlosen Betrieb mit bedächtigen Bewegungen bevorzugen.

Die jetzt nicht erläuterten Kommandos in den Zeilen 5-12 braucht man anfänglich nicht unbedingt zu verstehen. Wer sich allerdings intensiv mit der Thematik befassen möchte, dem empfehle ich, die Auswirkungen dieser Codezeilen nachzulesen. Das Wiki von FluidNC (Siehe Kurzlink in Kurzinfo) erweist sich dies-

bezüglich als wertvolle Wissensquelle. Weitere gute Stichwörter für die Google-Suche sind "GRBL" und "G-Code".

Selbstverständlich lässt sich der von Sandify generierte G-Code mit weiteren G1-Codes erweitern. So ließe sich zum Beispiel am Ende eines Skripts noch ein Fahrweg entlang des Perimeters einbauen, um einen schönen Abschluss in Form eines Rahmens rund um das gemalte Bild zu erzeugen.

> Da solche Codezeilen zu Beginn oder am Ende eines Skripts, also im Header respektive Footer des G-Codes, in der Regel immer ähnlich sind, gibt man diese am besten beim Exportieren des G-Codes in Sandify gleich mit. Hierzu ste-

Abbildung 3: Wie das in Sandify erstellte Sandbild tatsächlich aussieht, erkennt man erst, wenn das Kunstwerk gemalt wird. hen die Felder "Program start code" respektive "Program end code" in Sandify zur Verfügung. Alles, was im Header bzw. Footer (siehe Kasten "G-Code Header") zwischen; BEGIN PRE und ; END PRE steht, kann in Sandify eingetragen werden. Den Inhalt dieser Felder merkt sich Sandify beim Exportieren, sodass diese nicht immer neu befüllt werden müssen.

Aber Achtung: Sandify ignoriert unerklärlicherweise gewisse Zeilenschaltungen. Meldet die Firmware FluidNC bei der Ausführung des Skripts Fehler, so sollte man als Erstes auf korrekte Zeilenumbrüche im Header respektive Footer der G-Code Datei prüfen. Sind die anzuwendenden Header und Footer einmal in Sandify hinterleat, geht das Generieren von Bildern einfach von der Hand: G-Code exportieren, die Datei ohne weitere Nachbearbeitung direkt mit der Firmware FluidNC des Sandmaltisches auf die SD-Karte hochladen, G-Code-Datei abspielen - und los geht's mit der künstlerischen Sandmalerei! Wie das Sandbild (Abbildung 3) dann tatsächlich aussieht, bleibt spannend.

Verkettung von Sandbildern

Wer viel mit Sandify experimentiert, dürfte mit der Zeit ein ansehnliches Repertoire an Kunstwerken sein Eigen nennen. Schnell kommt dann der Wunsch auf, dass sich diese Sandbilder ähnlich einer Playlist nacheinander abspielen lassen. Andere Projekte ähnlicher Art setzen hierbei auf einen Raspberry Pi mit Octoprint, der die einzelnen Codezeilen einer G-Code-Datei über eine USB-Verbindung an die Firmware des Controllers übermittelt. Unser Sandmaltisch löst die Playlist-Funktionalität deutlich weniger aufwendig, nämlich ohne zusätzliche Hard- oder Software, was das ganze System wartungsfreundlicher und weniger fehleranfällig macht.

Möglich macht dies die brandneue Version 3.8.1 der Firmware FluidNC, welche erst vor Kurzem mit einer Job-Kontrolle ausgestattet wurde. Wer also noch die in der Bauanleitung und in Teil 1 des MAKE-Artikels genannte ältere Version auf dem Controller installiert hat, sollte für die Playlist-Funktionalität die Firmware aktualisieren. Die Aktualisierung ist simpel und innerhalb weniger Minuten erledigt. Es ist nicht einmal eine USB-Verbindung erforderlich, da die Installation über WiFi (OTA) erfolgen kann, sofern die Standardvariante von FluidNC wie empfohlen eingesetzt wird.

Als Windows-Nutzer lädt man zuerst die Datei fluidnc-v3.8.1-win64.zip von der FluidNC-Projektseite des GitHub herunter (Downloads siehe Kurzlink). In der ZIP-Datei findet man im Ordner wifi die Datei "firmware.bin". Anschließend ruft man mit einem Browser die Adresse "http://fluidnc.local" auf, womit man zur Benutzeroberfläche des Sandmaltisches gelangt. Dort klickt man im oberen Menübereich auf

FluidNC, um in die FluidNC-Einstellungen zu gelangen.

Durch Klick auf den mittleren, orangen Button mit dem Hinweistext "Update the firmware wirelessly (OTA)" lässt sich danach die heruntergeladene Datei "firmware.bin" auf den Controller hochladen. Nach einem Neustart des Controllers erfüllt damit unser Sandmaltisch bereits die Voraussetzungen für die Playlist-Funktionalität.

Eine Playlist (siehe Kasten "G-Code Verkettung") ist nichts anderes als eine G-Code-Datei, welche wiederum mehrere andere G-Code-Dateien in einer vorbestimmten Reihenfolge aufruft. Der Aufruf einer G-Code-Datei, also eines Sandbildes, geschieht durch das Kommando \$SD/Run= gefolgt vom Dateinamen des Bildes inklusive Pfadangabe. Bei der Pfadangabe sollte man unbedingt darauf achten, dass diese aus Sicht des Controller-Dateisystems angegeben wird und nicht etwa aus Sicht des lokalen PCs.

Da es meiner Meinung nach Sinn ergibt, zwischen dem Abspielen der einzelnen Dateien zu pausieren, um das fertige Bild länger betrachten zu können, füge ich jeweils zwischen den Dateiaufrufen eine Pause hinzu. Dies geschieht durch das Kommando G4 P86400, wobei der P-Parameter die Anzahl Sekunden festlegt, die pausiert werden soll. 86400 Sekunden bewirken umgerechnet eine Pause von 24 Stunden. Somit spielt der Sandmaltisch automatisiert jeden Tag ein neues Bild ab. Wenn die Pause unerwünscht ist, zum Beispiel wenn das Bild nur ein Löschbild ist, welches die Sandfläche für das darauffolgende Bild vorbereitet, kann man die Pause natürlich auch weglassen oder auf P1 (1 Sekunde) festlegen. Ein Header mit einer Home-Anweisung und weiteren Kommandos, wie er bei den Bilddateien verwendet wird, ist in der Playlist-GCODE-Datei übrigens nicht notwendig. Eine Playlist lässt sich auf diese Art und Weise mit jedem gängigen Texteditor programmieren. Beim Speichern der Textdatei sollte man darauf achten, dass die Datei in der UTF8-Codierung (ohne BOM) gespeichert wird und die Dateiendung "*.gcode" lautet.

Playlisten per Mausklick mit SATPG

Wer bereits einige Playlisten zusammengestellt hat, wird feststellen, dass die Erstellung einer ausgeklügelten Liste per Texteditor eine recht knifflige Sache ist. Die Schwierigkeit liegt darin, dass sich ein Sandbild nur schwer mit dem Dateinamen beschreiben lässt. Man weiß nie so genau, welches Bild sich hinter einem Dateinamen verbirgt. Außerdem werden, wie wir bereits wissen, die Sandbilder beim Abspielen der Playlist übereinander geschichtet. Wenn sich nun in der Playlist Bilder befinden, die sich nicht flächendeckend ausbreiten,



Abbildung 4: Mit SATPG lassen sich Playlisten per Mausklick zusammenstellen und visualisieren.

dann ist die Reihenfolge der Bilder durchaus entscheidend für das Gesamtergebnis. Und schließlich sehen wir uns noch der Herausforderung gegenüber, dass der Endpunkt des letzten Bildes, dort wo die Kugel zu stehen kommt, unweigerlich mit dem Startpunkt des neuen Bildes verbunden wird. Die so entstehende Sandspur wird bei der Zusammenstellung einer Playlist oft vergessen, denn diese Verbindung von zwei Punkten ist in einer einzelnen G-Code-Datei schlicht nicht enthalten.

Eine Lösung dafür bietet das Tool "Sand Art Table Playlist Generator" (SATPG, Abbildung 4), welches auf GitHub zum Download bereitsteht. Mit diesem Tool lassen sich Playlists per Mausklick zusammenstellen, indem die Bilddateien selektiert und in beliebiger Reihenfolge platziert werden. Eine grafische Voransicht zeigt jeweils den Inhalt der Bilddateien. Sie berücksichtigt dabei unter anderem auch die Homing-Position und den Endpunkt des vorgängigen Bildes. Eine so zusammengeklickte Playlist lässt sich schließlich als G-Code-Datei in einem frei wählbaren Verzeichnis ausgeben, wobei auch die Bilddateien der Abspielreihenfolge entsprechend nummeriert und ins Verzeichnis kopiert werden. Wie das genau vonstattengeht, wird euch in einem SATPG Tutorial (siehe Links zum Artikel) erklärt.

Replay-Funktionalität

Eine unendliche Wiederholung der Playlist (Replay) lässt sich mit dem derzeitigen Stand der FluidNC-Firmware nur über einen Hack realisieren. Dieser ist der letzten Zeile im Kasten "G-Code Verkettung" ersichtlich: Anstelle eines Sandmalbildes ruft sich die Playlist ganz einfach selbst noch einmal auf. Die so programmierte Endlosschleife funktioniert im Prinzip ganz gut. Allerdings sollte man darauf gefasst sein, dass irgendwann ein Speicherüberlauf stattfinden wird und der Controller neu gestartet werden muss. Wann dies geschehen wird, ist natürlich abhängig von der Größe und Anzahl der Dateien und der hinzugefügten Pausen. Die Risiken sind klein, man darf diesen Hack ruhig wagen.

Die Entwickler von FluidNC haben glücklicherweise angekündigt, demnächst eine Schleifen-Funktionalität (Do ...while ...) zu realisieren. Man darf also gespannt sein, ob in naher Zukunft die Anwendung dieses Hacks überflüssig wird und man auf "gesittete" Art und Weise Wiederholungen ohne Speicherüberlauf programmieren darf. Eventuell ist eine entsprechende Version mit dieser Funktionalität zum Zeitpunkt der Publikation dieses Artikels bereits verfügbar.

Im GitHub zum Artikel befindet sich im Ordner "2 - Sandbilder" ein Unterordner "BIN-GOBRICKS". Für einen Test der Playlist-Funktion kann dieser gesamte Ordner ins Root-Verzeichnis der SD-Karte des Sandmaltisch-Controllers kopiert und die Playlist-Datei daraus gestartet werden. Wenn man auf diese Weise vorgeht, sollten die Pfadangaben in dieser Beispiel-Playlist automatisch stimmen und sie kann somit ohne weitere Bearbeitung abgespielt werden. Ebenfalls im GitHub ist der Downloadlink (auch per Kurzlink zu erreichen) zu SATPG abgelegt.

Nun viel Spaß beim Erstellen und Betrachten von euren ganz eigenen Sandbildern. Mit einem besonders schönen Möbelstück wollte ich damals dem muffigen Bastelkeller entfliehen und an der Wächterin des Wohnzimmers vorbei zum Mittelpunkt unserer Wohnung gelangen. Diese Heldenreise ist mir gelungen und ich wurde mit einem dicken Kuss meiner Partnerin belohnt. So soll es euch auch geschehen. —*caw*

Die Soft-Matrix

In diesem Artikel zeigen wir, wie man berührungsempfindliche Panels mit Matrizen und leitfähigen Textilien herstellt. Diese berührungsempfindlichen Panels können mit einem Mikrocontroller verwendet werden, um Projekte mit User-Input zu steuern.

von Lee Wilkins



ine 2D-Matrix ist eine Reihe von Zeilen und Spalten – im Grunde eine Tabelle. Man kann eine Stelle in der Matrix bestimmen, indem man den Schnittpunkt von Zeile und Spalte überprüft. Zum Beispiel die Stelle 2b in Tabelle 1.

Dieses Konzept wird häufig bei Touchpads und -panels verwendet, weil man damit leicht herausfinden kann, wo der Nutzer physisch drückt. Heutzutage verwenden die meisten Touchscreens kapazitive Techniken oder andere Technologien, um feinere Unterschiede bei Berührung zu erkennen. Das funktioniert, indem die Berührung mit der Fingerspitze winzige Veränderungen des elektrischen Feldes im Touchpad erzeugt und diese dann gemessen werden.

Aber einige der ersten Touchscreens, wie der Palm Pilot, verwenden die resistive Touchtechnologie, um die Fingerspitze oder den Bedienstift durch Druck zu erkennen. Leitfähige Reihen und Spalten eines Gitters sind durch eine semileitende Barriere getrennt. Beim Drücken auf dieses Material ändert sich der elektrische Widerstand. Dadurch wird dann eine Reihe mit einer Spalte verbunden. Diesen Schnittpunkt kann man dann genau bestimmen (Abbildung 1).

In einem maschinell hergestellten resistiven Touchscreen wird ein transparentes, leitfähiges Material namens Indium-Zinn-Oxid in einer sehr dünnen Schicht in einem Raster auf die Bildschirmpanels aufgetragen, sodass das Bild des Displays noch sichtbar ist.

Man kann Matrizen auch als gängige Komponenten in der DIY-Elektronik wie einer LED-Matrix oder 7-Segment-Anzeige finden. Während ein Touchscreen funktioniert, indem Leiterbahnen zusammengedrückt werden, um Koordinaten zu erfassen, funktioniert eine LED-Matrix durch das Senden von Koordinaten aus einem Computerprogramm. Dadurch werden dann die positiven und negativen Enden einer LED durch Zeilen und Spalten verbunden, um ein bestimmtes Licht zum Leuchten zu bringen.

Um ein weiches Touchpanel zu schaffen, verwendet man resistive und leitfähige Materialien, um ein Sandwich aus Stoff, Leiterbahnen und semileitendem Material herzustellen.

Im Wesentlichen ordnet man die Leiterbahnen in Reihen und Spalten an, die durch ein semileitendes Material getrennt sind. Wenn der Schnittpunkt der Zeilen und Spalten gedrückt wird, ändert sich der Widerstand zwi-

Tabelle 1: Einfache Koordinatenmatrix

	1	2	3
а	1a	2a	3a
b	1b	2b	3b
с	1c	2c	3c

Kurzinfo

» Flexible Bedienelemente erstellen
 » Nutzung der Stoffmatrix mit Arduino
 » Herstellung aus einfachen Materialien



schen den Leiterbahnen, und man kann den physischen Ort der Berührung erkennen. Je nach der Dichte des Gitters erhält man mehr oder weniger genaue Daten bzw. mehr oder weniger Interaktionspunkte. Fangen wir ganz einfach an.

An der Herstellung eigener Touchmatrizen liebe ich, dass man sie in jeder beliebigen Form anlegen kann. Man kann es mit wackeligen Linien, neuen Textilien oder mit dem Vergrößern und Verkleinern des Rasters probieren. Auch wenn die meisten Widerstandsmatrizen ein gleichmäßiges Gitter haben, gibt es keinen Grund, dass man nicht auch ein wellenförmiges, sich überschneidendes Muster mit unterschiedlichen Dichtegraden erstellen sollte.

Vorbereitung

Man kann entweder eine vorgefertigte Druckmatrix verwenden wie die Mega Pressure Matrix von Loomia (Abbildung 2) oder eine eigene herstellen. Loomia fertigt eine Vielzahl von Soft Surface Interfaces für das Prototyping. Diese

8 input x/y multiplexed resistive touchscreen

8 input tri-state multiplexed touchscreen



Abbildung 1: Matrizen bestehen aus übereinandergelegten Kontaktbahnen.



Abbildung 2: Fertige Stoffmatrizen gibt es auch für DIY-Projekte.



Abbildung 3: Für alles gerüstet: große Kontakte fürs Prototyping und kleine zum festen Anlöten

Produkte bestehen aus einem Sandwich aus ähnlichen Materialien, wie sie später in diesem Artikel für eine DIY-Matrix genutzt werden. Die Rückseite des Panels hat außerdem eine sehr praktische Klebebeschichtung. Und es gibt große Anschluss-Pads für das Prototyping mit Krokodilklemmen. Dazu kommen noch passende Durchgangslöcher, die man fest verlöten kann (Abbildung 3).

In diesem Artikel wollen wir aber eine eigene Matrix erstellen. Dafür muss man zuerst ein Gitter entwerfen. Eine einfache Möglichkeit für den Anfang ist ein regelmäßiges Gitter. Wie bei den meisten elektronischen Heimwerkerprojekten, besonders bei Textilien, ist die Versuchung groß, beim ersten Versuch klein anzufangen, aber wenn man sich etwas Platz gönnt, wird der Prozess einfacher.

Man klebt die Reihen aus leitendem Klebeband auf ein Stück Stoff und die Spalten auf ein anderes Stück. Man muss darauf achten, dass die Reihen gerade sind und sich richtig überlappen, wenn man sie übereinander legt. Es ist hilfreich, zuerst eine Schablone mit einem Marker anzufertigen und die Leiterbahnen daraufzulegen (Abbildung 4).

Beginnen wir mit einer 3-mal-3-Matrix. Diese hat 9 Kreuzungen. Wichtig ist, sich zu vergewissern, dass die Zeilen und Spalten mindestens auf einer Seite bis zum Rand des Materials reichen. Dort werden später Kabel angeschlossen. Jetzt kommt das semileitfähige Material ins Spiel: Velostat. Das ist ein dünnes, leicht gummiartiges Material. Man legt ein Stück davon zwischen die beiden Gitter (Abbildung 5 und 6), sodass beide Gitterhälften vollständig voneinander getrennt sind, und näht sie dann zusammen. Der verwendete Faden sollte nicht leitfähig sein. Die Matrix selber ist jetzt fertig. An den freigelassenen Enden werden nun noch Krokodilklemmen angeschlossen

Das andere Ende der Krokodilklemmenkabel wird an den Mikrocontroller angeklemmt. Alle Zeilen sind analog und alle Spalten digital. Ich habe das Mikrocontroller-Board Adafruit Flora verwendet, da es für Krokodilklemmen geeignet ist (Abbildung 7). Aber dieses Konzept funktioniert mit jedem ähnlichen Mikrocontroller; man muss nur die Pinbelegung überprüfen, um die digitalen und analogen Pins zu finden (Abbildung 8).

Der Code

Den vollständigen, kommentierten Arduino-Code findet man auf GitHub. Ich habe für mein Projekt den Code von Kobakants ursprünglichem Resistiv-Touch-Konzept angepasst. Zu beidem findet man einen Link in der Kurzinfo.

Zuerst definiert man die Anzahl der Zeilen und Spalten.

#define numRows 3 #define numCols 3

Als Nächstes erstellt man die zwei Arrays rows[] und cols[] und fügt die Namen der verwendeten Pins hinzu – hier muss man auf das Pinbelegungsdiagramm achten. Danach erstellt man ein zweidimensionales Array, um die eingehenden Daten zu erfassen:

```
int rows[] = {A10, A9, A7};
int cols[] = {3, 2, 0};
int incomingValues[numRows][numCols] =
{};
```

In der Setup-Funktion werden alle Zeilen durchlaufen und die Pins auf INPUT_PULLUP gesetzt (siehe Kasten Code: Setup).

Die Pins für die Spalten stellt man ebenfalls auf INPUT und startet den seriellen Monitor:

An dieser Stelle wird der Code ein wenig knifflig. Jedes Element im zweiten Array wird überprüft, indem man zuerst durch Zeilen und dann durch die Spalten durchschaltet. Beim Durchlaufen der Matrix wird immer eine Spalte als OUTPUT gesetzt und dann werden alle Zeilen durchgegangen, ob es eine Verbindung zu





Abbildung 5: Sieht aus wie Papier, ist aber Velostat.

Abbildung 4: Das in der Kurzinfo verlinkte Kupferband klebt erstaunlich gut auf Stoff.

NIX VON DER Stange!



Wunsch-PC selber bauen oder aufrüsten

- Bauvorschläge für Gaming-PC/Allrounder
- Nachhaltig und günstig: Alten Rechner länger nutzen
- Praxisanleitung: Windows auf neue SSD umziehen
- Der große CPU-Wegweiser

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

🖙 shop.heise.de/ct-hardwaretipps24

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.





Abbildung 6: Kupferleiter – Velostat – Kupferleiter. Die Matrix ist fertig.

Abbildung 7: Der Adafruit Flora ist perfekt für das Prototyping mit Krokodilklemmen.



FLORA Wearable Electronics Platform adafruit.com/products/659 drawing 2012 by J. M. DeCristofaro -- CC-BY-SA 3.0

Abbildung 8: Dieses Pinout ist speziell für den Adafruit Flora.

Code: Setup
void setup() {
 for (int i = 0; i < numRows; i++) {
 pinMode(rows[i], INPUT_PULLUP);
 }
 for (int i = 0; i < numCols; i++) {
 pinMode(cols[i], INPUT);
 }
 Serial.begin(9600);
}</pre>

dieser Spalte gibt. Danach wird diese Spalte wieder auf INPUT gestellt und die nächste Spalte überprüft.

Wenn man diesen Code (Kasten Code: Loop) auf den Arduino hochlädt, kann man direkt loslegen. Im seriellen Monitor sind dann die Ausgaben zu sehen.

Testen der Touchmatrix

Am Ende des Codes befinden sich zwei Funktionen, die bei der Fehlersuche helfen. Die Funktion printGrid geht durch alle Zeilen und Spalten und schreibt sie als Zahl in den seriellen Monitor. Wenn man einen Kupferband-Schnittpunkt drückt, wird der Wert der gesamten Zeile niedriger, aber die genaue Position der Schaltfläche wird als niedrigster Wert angezeigt.

Um den niedrigsten Wert zu finden, kann man die Funktion findActiveButton() verwenden (Kasten Code: Debug). Diese Funktion durchläuft alle eingehenden Werte und stellt fest, welcher der niedrigste ist. Wenn einer von ihnen unter 100 liegt, ist es wahrscheinlich, dass auf eine Zeilen/Spalten-Kreuzung gedrückt wird. Bei einer eigenen Touchmatrix ist es möglich, dass diese Zahlen unterschiedlich sind. Hier muss man manuell prüfen, wie die Werte aussehen, wenn etwas gedrückt wird.

Wenn die Zahlen unregelmäßig sind, ist es möglich, dass das Widerstandsmaterial die leitenden Zeilen und Spalten nicht vollständig trennt oder dass eine Verbindung irgendwo nicht hergestellt wird. Überprüfen Sie Näharbeit und Verkabelung. Um der Touchmatrix eine andere Beschaffenheit zu geben, kann man mit der Platzierung und dem Material experimentieren.

Installieren

Jetzt kann man die Werte der Matrix auslesen und es wird Zeit, sie einzusetzen. Mit einem Arduino Pro Micro kann man z. B. ein super flaches, platzsparendes Macro-Pad bauen, das man ganz bequem unter einem Mauspad verstecken kann.

Der Arduino Pro kann sich nämlich als USB-Tastatur verkleiden und damit Befehle an einen

Code: Loop

```
void loop() {
  for (int col = 0; col < numCols; col++) {
    pinMode(cols[col], OUTPUT);
    digitalWrite(cols[col], LOW);
    for (int row = 0; row < numRows; row++) {
        incomingValues[row][col] = analogRead(rows[row]);
     }
    pinMode(cols[col], INPUT);
    }
    printGrid();
    findActiveButton();
    delay(10);
}</pre>
```

Code: Debug

```
void findActiveButton() {
    int smallest = 1024;
    int smallestRow = -1;
    int smallestCol = -1
    for
        (int col = 0; col < numCols; col++) {</pre>
         for (int row = 0; row < numRows; row++) {</pre>
             if (incomingValues[row][col] < smallest) {</pre>
                 smallest = incomingValues[row][col];
                 smallestCol = col;
                 smallestRow = row;
             }
        }
    if (smallest < 100){</pre>
        Serial.print("There is a button being pressed in row: ");
        Serial.println(smallestRow);
        Serial.print("and column: ");
        Serial.println(smallestCol);
    } else {
        Serial.println("Nothing is being pressed");
    }
}
```

Computer schicken. Dafür verwendet man die "keyboard Library" von Arduino, die man über den Bibliotheksmanager in der Arduino IDE installiert. Gleiches gilt für die HID-project-Bibliothek.

Jetzt reichen schon ein paar zusätzliche Zeilen im Code, um den verschiedenen Spalten/Zeilen-Kombinationen Befehle zuzuweisen (siehe Kasten Code: Macro). Über einen Switch-Case wird die aktivierte Zeile aufgerufen und in dieser die aktivierte Spalte - im Endeffekt also der gedrückte Schnittpunkt. Für jeden dieser Schnittpunkte ist dann eine Aktion auf dem Keyboard festgelegt: entweder das Tippen eines Textes, oder das Drücken einzelner Tasten. Wenn einzelne Tasten gedrückt werden, muss man diese auch wieder loslassen. Das geschieht mit Keyboard.release-All();. -das

Code: Macro void macros(int x, int y) { switch (x) { case 0: switch (y) { case (Keyboard.print('1'); break; case 1: Keyboard.press(KEY_RIGHT_CTRL); Keyboard.press('s'); delay(10); Keyboard.releaseAll(); break: } break; case 1: switch (y) { case 0: break: case 1: break; } break: 3 }

Apple II – Die Nostalgie-Zeitreise Mit Zeitzeugen-Interview mit John Romero

Als man mit 4K nicht die Auflösung, sondern die Größe des Arbeitsspeichers beschrieb, erschien mit dem Apple II der einflussreichste Computer aller Zeiten. Er sollte den Menschen zeigen, wozu sie einen "Personal Computer" in ihrem Leben überhaupt brauchen. Heute, fast ein halbes Jahrhundert später, kann der Apple II dank moderner Hardware und einer großen Fangemeinde zu neuem Leben erweckt werden.

von Tobias Hübner



enn es ein Äquivalent in der IT-Welt zum VW-Käfer gibt, dann ist es der Apple II. Für eine ganze Generation (vor allem US-amerikanischer) Kids in den 80ern war er der Inbegriff des Computers. Er war leicht zu verstehen und zu erweitern, sah gut aus und war erschwinglich.

Unglaubliche 16 Jahre lang – von 1977 bis 1993 – wurde der Apple II in verschiedenen Versionen verkauft. Insgesamt produzierte Apple vermutlich rund 6 Millionen Exemplare. Rückblickend schuf das Unternehmen so eines der langlebigsten IT-Ökosysteme, das in Form der Apple IIe Card für den Macintosh noch weit bis in die 90er-Jahre Bestand hatte. Diese Apple IIe Card ermöglichte es, Apple-II-Software auf neueren Macintosh-Computern zu nutzen.

Wie es zu diesem riesigen Erfolg kam (zum Vergleich: vom Apple-1 wurden gerade einmal 200 Exemplare verkauft), hat viele Gründe: etwa die acht Erweiterungsslots, das schicke Design, die Farbwiedergabe oder das effizient konstruierte Diskettenlaufwerk. Einen großen Anteil am Verkaufserfolg hatte aber auch Apples frühes Engagement im Bildungsbereich, denn dank eines staatlichen Förderprogramms konnte Apple im Jahr 1983 für wenig Geld jede Schule in Kalifornien mit einem Apple II ausstatten.

Dieser geschickte Schachzug, der Apple lediglich ca. 1 Million Dollar kostete, stellte sich im Nachhinein als eine überaus sinnvolle Investition heraus, denn Hersteller von Lernsoftware veröffentlichten ihre Produkte fortan in erster Linie für Apple-Rechner und Eltern kauften ihren Kindern den PC, den sie aus der Schule kannten. Rückblickend sagte Steve Jobs 1995 daher völlig zu Recht: "One of the things that built Apple II's was schools buying Apple II's."

Blauer Rauch zur Begrüßung

In Deutschland war der Apple II weitaus weniger verbreitet, aber auch hier waren es vor allem Schulen, die aufgrund einer starken Rabattierung den Apple II kauften. Es ist daher kein Wunder, dass ich meinen ersten verstaubten Apple IIe – das mit Abstand am meisten verkaufte Modell aus der Apple-II-Serie – vor ca. 10 Jahren im Keller einer Schule fand und gerade noch vor der Verschrottung retten konnte.

Die Freude darüber, dass der Rechner sich noch anschalten ließ, währte jedoch nur kurz, denn plötzlich hörte ich einen leisen Knall, Rauch stieg auf und übler Gestank verbreitete sich im Raum. Heute weiß ich, dass das eine Art Initiationsritus war, denn Gleiches geschieht bei ausnahmslos allen Apple-II-Rechnern nach kurzer Zeit. Zum Glück ließ sich der Fehler jedoch leicht wieder beheben. Im Netzteil befinden sich zwei Entstörkondensatoren,

Kurzinfo

» Rückblick auf die Geschichte des Apple II
 » Überblick über Erweiterungsmöglichkeiten
 » Interview mit John Romero

Mehr zum Thema

- » Thomas Ronzon und Michael Schröder, Per Anhalter durch die Amiga-Disk-Galaxie, Make 5/22, S. 28
- » Daniel Schwabe, Retrobright aus Gelb mach Weiß, Make 6/23, S. 16
- » Marcel Timm, Raspberry Pi als SD-Karten-Laufwerk für C64 & Co., Make 5/21, S. 62







Steve Jobs erster großer Marketing-Coup: Die Kampagne "Kids Can't Wait" brachte Apple-II-Computer in Kaliforniens Schulen.



Der Apple II stinkt und braune Flüssigkeit klebt auf der Platine des Netzteils? Willkommen im Club! Die Entstörkondensatoren müssen getauscht werden.



Glücklich ist derjenige, dessen RAM-Chips im Apple IIe nicht festgelötet (linkes Bild), sondern in Sockel gesteckt sind (rechtes Bild).

die man für wenige Euro nachbestellen und mithilfe eines Lötkolbens problemlos austauschen kann. Einen Artikel, wie man Schaltnetzteile repariert, findet man in der Make 06/2023 auf Seite 26 (siehe Link in der Kurzinfo).

Da der Apple II mit dem frisch reparierten Netzteil unregelmäßig falsche Zeichen anzeigte, war es Zeit für das interne Diagnoseprogramm. Es erscheint, wenn man die beiden Apple-Tasten links und rechts neben der Leertaste gedrückt hält und den Rechner anschaltet. Und tatsächlich half das Programm bei der Fehlersuche weiter. Die Ausgabe 1 0 0 0 0 0 auf dem Bildschirm zeigte an, dass ein RAM-Chip defekt war.

RAM-Chips für den Apple II sind für wenig Geld (etwa 2 bis 3 Euro) erhältlich. Wenn man Glück hat, befinden sich auf dem Motherboard bereits Sockets, sodass man den RAM-Chip einfach mit einem IC-Extractor entfernen kann. Sind die Chips verlötet, muss das Motherboard entfernt und der Chip aufgelötet werden. Wenn man sich schon diese Mühe macht, empfiehlt es sich, statt des Ersatzchips zunächst selbst einen Sockel einzubauen, damit ein weiterer Austausch in Zukunft weniger Mühe bereitet. Noch ein wichtiger Hinweis: Die oben genannte Programmausgabe gibt wenig intuitiv an, dass der Chip ganz rechts ausgetauscht werden muss.

Und was läuft nun auf diesem Gerät?

Bei meinem Apple IIe waren zwar zwei 51/4"-Laufwerke dabei, aber leider keine Disketten. Dennoch lassen sich problemlos ohne Zubehör Hunderte Programme auf den Rechner laden. Wie? Indem man die Kopfhörerbuchse eines Handys (oder PCs) mit dem Audio-Eingang des Apple II verbindet und die Seite asciiexpress.net (Link in der Kurzinfo) aufruft. Hier finden sich viele Programme, beispielsweise das hilfreiche Diagnosetool Apple-Cillin II, die sich auf den Rechner überspielen lassen. Computer einschalten, CTRL-RESET drücken, den Befehl LOAD eingeben und die Wiedergabe der entsprechenden Sounddatei auf dem angeschlossenen Gerät starten.

So lässt sich zwar viel Geld bei der Beschaffung passender Software sparen, vor allem bei alten Spielen entgeht einem jedoch einiges, wenn man das Original nicht besitzt. Im Gegensatz zu modernen Spielen, die meist nur noch digital vertrieben werden, boten die zunächst in verschließbaren Plastikbeuteln und später dann in großen Pappboxen verpackten Originale damals noch einen echten Mehrwert in Form aufwendig gestalteter Handbücher und origineller Feelies wie Münzen oder Stoffkarten. Drei dieser Spiele möchte ich einmal exemplarisch vorstellen: "The Hitchhiker's Guide to the Galaxy", das Textadventure zum gleichnamigen Roman von Douglas Adams, das unter anderem mit einem "Don't Panic"-Button und einer Microscopic Space Fleet (anders ausgedrückt: einer leeren Plastiktüte) ausgeliefert wurde.

Weiter geht es mit "Where in the World is Carmen Sandiego". Das auch heute noch gut spielbare Rätselspiel verlangte vom Spieler, dass er Fakten im mitgelieferten "World Almanac" von 1986 nachschlug.

Und zur guter Letzt, der König der Rollenspiele: Ultima. Das erste Spiel der Ultima-Serie wurde noch im Plastikbeutel vertrieben, der zweite Teil dann in einer "Big Box" ausgeliefert. Fortan wurden aufwendig gestaltete Packungsbeigaben wie Stoffkarten, Münzen und wunderschöne Handbücher das Markenzeichen der Rollenspiel-Serie.

Dass der Apple II bei Kindern und Jugendlichen zur beliebten Gaming-Plattform wurde, ist übrigens kein Zufall. Von Beginn an wollte Steve Wozniak mit dem Apple II einen Rechner konstruieren, mit dem man das Spiel Breakout spielen konnte. Daher besaß der Apple II von Anfang an Anschlüsse für Joysticks und Paddles, eine rudimentäre Tonausgabe und die im ROM integrierte Programmiersprache Apple BASIC.



Der Apple II war die ideale Plattform zum Spielen.







Steve Wozniak wollte einen Computer, um Breakout zu spielen – so entstand der Apple II.



Mehr Speicher? Kein Problem. Mit diesem Board kann man Flash-Karten in den Apple II einsetzen.

Der Apple II und seine Erweiterungen

Aber der Apple II war viel mehr als eine Spielkonsole. Es ist von heute aus betrachtet fast schon unglaublich, was man mit einem 1 MHz "schnellen" Prozessor und einem 64 KB RAM "großen" Arbeitsspeicher alles anstellen konnte, z. B. Tabellenkalkulation mit der Software VisiCalc, einem echten System-Seller. Sogar einen grafischen Desktop mit Icons und Maussteuerung gab es auf dem Apple II und selbst die Programmierung von LEGO-Motoren und -Sensoren war dank einer speziellen Interface-Karte und der Programmiersprache Logo mit den LEGO-TC-Logo-Sets möglich.

Der Aftermarket für den Apple II

Auch heute hat der Apple II vor allem in den USA noch immer eine riesige Fangemeinde, die sich nicht nur online, sondern auch im echten Leben auf dem jährlich stattfindenden "Kansas Festival" trifft. Dort wird auch regelmäßig neue Software für den Apple II präsentiert, etwa "Nox Archaist", ein Rollenspiel im Stil der ersten Ultima-Titel, das 2020 erschien und bis heute das einzige Apple-II-Spiel auf Steam ist. Oder "Total Replay" – ein Frontend für ca. 100 beliebte Spiele auf dem Apple II, das moderne Features wie eine UI-Suche für Spiele und einen Cheat-Modus für die kniffligeren Passagen anbietet.

Auch die Hardware des Apple II wird von Enthusiasten stetig verbessert, vor allem in Form neuer Erweiterungskarten, die mangels eines deutschen Vertriebs leider meist aus den USA importiert werden müssen.

Die CFFA3000 steht auf der Wunschliste vieler Apple-II-Fans ganz weit oben. Sie ermöglicht es unter anderem, eine Festplatte zu emulieren und Daten per USB-Stick oder CompactFlash-Karte auf den Apple II zu übertragen.

Eine weitere Möglichkeit, die Floppy-Laufwerke zu ersetzen, ist der FloppyEMU, der nicht nur an den Apple II, sondern auch an den Macintosh und den LISA-Computer angeschlossen werden kann.

Wie bereits erwähnt, musste ich ja direkt am Netzteil meines Apple IIe meine Maker-Fähigkeiten beweisen. Das Herumfummeln an Netzteilen ist jedoch nicht ungefährlich. Auch hier steht einem die Community helfend zur Seite. Wer kein Risiko eingehen will, kann sich ein nagelneues Innenleben für sein Netzteil kaufen und ins alte Gehäuse einbauen.

Wem ein Anschluss für moderne Monitore am Apple II fehlt, der kann auf die Apple II VGA Card zurückgreifen, die mithilfe eines Raspberry Pi Pico ein gestochen scharfes Bild ausgibt. Die Karte kann man sich dank Open-Source-Lizenz selbst nachbauen (Link in der Kurzinfo). Andere Karten bieten auch die Möglichkeit einer HDMI-Ausgabe.

Wenig bekannt ist, dass es für den Apple II schon damals eine recht potente Soundkarte namens Mockingboard gab, die leider nur von wenigen Programmen unterstützt wurde.



Dank Steckkarte, Interface und passender Software konnte man mit dem Apple II LEGO-Motoren und -Sensoren ansteuern.



Floppy Emu – der freundliche Disketten-Emulator.

Nachbauten dieser Karte ermöglichen es heute, Spiele wie Ultima III, IV und V mit ansprechender Hintergrundmusik zu spielen. Im Bild zu sehen ist die Mega Audio von A2Heaven. Näher am Original ist der Nachbau der Mockingboard von der Firma ReActiveMicro, die zudem ein Set zum Selberlöten verkauft.

Wem die 1 MHz des 6502-Mikroprozessors nicht reichen, der kann die Leistung des



Better new than sorry – neue Netzteile gibt es auch heute noch für den Apple II.



Der Raspberry Pi Pico hilft bei der Ausgabe auf moderneren Bildschirmen.



Der Mega Audio von A2Heaven ist der Nachbau einer Apple II Soundkarte.

Apple II mit Accelerator-Karten wie der FASTchip //e auf bis zu 16 MHz erhöhen und nebenbei auch den Arbeitsspeicher erweitern.

Damit man all die schönen Erweiterungskarten auch sehen kann, verkauft die Firma MacCase transparente Gehäuse sowohl für den Apple IIe als auch für die Diskettenlaufwerke. Sogar eine transparente mechanische Tastatur ist erhältlich.

Dank der sehr aktiven Community entstehen noch immer neue faszinierende Erweiterungen für den Apple II. Jüngst erschienen ist beispielsweise die ESP32 SoftCard, die ein eigenes Composite-Video-Signal generiert und es so beispielsweise ermöglicht, Doom auf dem Apple II zu spielen oder einen Macintosh Classic zu emulieren.

Empfehlungen

In letzter Zeit sind einige herausragende Bücher zum Thema Apple II erschienen, allen voran "The Apple II Age" von Laine Nooney.



Jetzt aber schnell: Mit einer Beschleunigungskarte wird der Apple II pfeilschnell.

Wer sich mehr für die technischen Details interessiert, sollte einen Blick in "The New Apple II User's Guide" von David Finnigan werfen. Ausschließlich mit Computerspielen auf dem Apple II beschäftigt sich hingegen das Buch "Break Out – How the Apple II Launched the PC Gaming Revolution", das detailliert 14 wegweisende Spiele auf dem Apple II und ihre Macher porträtiert.

Wer nach dem Lesen dieses Artikels Interesse bekommen hat, den Spirit des Apple II einmal selbst zu erleben, aber keine Lust auf staubige und teure Hardware hat, kann einen der zahlreichen Emulatoren installieren, etwa microM8. Wenn man eine Sicherheitskopie der eigenen Software in Form einer Image-Datei auf dem Rechner hat, kann man sie einfach einem Laufwerk zuweisen. Dann entfaltet sich der Apple-II-Zauber auch auf dem modernen Flachbildschirm in 4k-Auflösung. Außerdem werden die knackigen Retrospiele mit modernen Features wie Speichern zu jeder Zeit etwas zahmer.



Mit einem schicken Gehäuse sieht man auch die Erweiterungen im Apple II.

Interview mit John Romero

John Romero, geboren 1967, wurde als Gründer von id Software und Programmierer von PC-Spielen wie DOOM und Quake zur Legende. Gerade ist seine Autobiografie "Doom Guy – Life in First Person" erschienen, die gleich doppelt verfilmt werden soll. Seine ersten Spiele programmierte und veröffentlichte Romero jedoch auf dem Apple II. Heute lebt er mit seiner Frau Brenda Romero, ebenfalls preisgekrönte Spieleentwicklerin, in Irland.

Make Magazin: Hallo John, vielen Dank, dass wir mit dir über den Apple II reden können. Hast du eigentlich noch einen bei dir rumstehen?

John Romero: Ja, gleich hier um die Ecke. Ich bin gerade im Büro und dort steht ein Platinum Ile. Zu Hause habe ich noch einen weiteren Platinum Ile auf einem separaten Schreibtisch in meinem Büro. Die Laufwerke liegen oben auf dem Amdek-Monitor – alles original. Ich habe insgesamt etwa elf Apple-II-Computer, vier davon funktionieren perfekt.

Make Magazin: Du hast gerade dein erstes Buch veröffentlicht – "Life in First Person". Darin beschreibst du, dass das wohl wichtigste Ereignis in deinem Leben 1979 im Computerraum des Sierra College stattfand. Was genau ist da passiert?

John Romero: Nun, ich hatte gehofft, dort ein paar wirklich tolle Arcade-Spiele spielen zu können. Ein Freund zeigte mir damals das Informatikgebäude am Sierra College, der örtlichen Volkshochschule, denn ich wollte unbedingt wissen, wo diese geheime Spielhalle ist. Er führte mich in einen Raum, in dem ein paar Menschen an Terminals saßen. Mein Freund loggte sich an einem davon ein und zeigte mir Games wie das Textadventure, Hunt the Wumpus". Es waren keine Arcade-Spiele – eher Brettspiele, bei denen man wirklich nachdenken musste. Ich dachte: "Wow, das ist wirklich neu und cool." Ich fragte dann noch jemand anderen im Raum, was er macht, und er antwortete: "Programmieren." "Programmieren?", fragte ich. "Ja, ich mache Programme wie diese Spiele hier." Und von da an wollte ich unbedingt wissen, wie man das macht.

Make Magazin: Dachtest du damals schon, dass du für den Rest deines Lebens Spiele programmieren möchtest oder hattest du noch einen Plan B?

John Romero: Das wusste ich tatsächlich schon damals. Wenn man erst einmal anfängt, Assembler zu lernen und Spiele wie



John Romero mit Steve Wozniak und Tom Hall (1998).

Subnodule zu entwickeln – mein erstes richtiges Spiel auf dem Apple II –, ist das eine Menge Arbeit, macht aber auch wirklich Spaß und es ist sehr kreativ. Man muss sich zunächst das Design ausdenken. Dann kommt der Sound dazu und das Zeichnen der Grafiken. All das musste man damals alleine machen, wenn man ein Spiel programmieren wollte. Aber dadurch habe ich auch sehr viel gelernt.

Make Magazin: Was würdest du heute tun, wenn du als Kind nie Zugang zu einem Computer gehabt hättest?

John Romero: Dann würde ich eben jetzt loslegen und alles darüber lernen wollen, was ich kann. Da ich mich für Spiele interessiere, würde ich auf jeden Fall lernen wollen, wie man sie programmiert. Heute würde ich erst einmal den Umgang mit der Unreal Engine lernen und wie das Blueprint-System funktioniert. Dann würde ich mich mit den Grundlagen von C++ beschäftigen und probieren, wie man C++ mit Blueprint verknüpfen kann.

Make Magazin: Glaubst du, dass man auch im 21. Jahrhundert noch wissen sollte, wie man auf einem Apple II programmiert oder wie z. B. Vektorgrafiken oder Old-School-Kachelgrafiken funktionieren?

John Romero: Oh ja, denn alle Grafiken sind eigentlich noch immer Kachelgrafiken. Wenn man beispielsweise Texturen für irgendetwas erstellt, sind diese quadratisch oder rechteckig. Wenn man die Oberflächen von 3D-Modellen gestalten will, muss man sie auseinanderklappen und dann hat man ein großes rechteckiges Bild, das man gestalten kann. Außerdem werden noch immer jede Menge Spiele im Retrolook entwickelt. Dieser Look ist einfach zu gut, als dass er jemals verschwinden wird.

Make Magazin: Wo ich das höre, erinnere ich mich daran, wie ich in den 90ern die Skins im Spiel Quake modifiziert habe.

John Romero: Oh ja, die Limitierungen von damals waren wirklich lustig. Es gab links und rechts am 3D-Modell der Hauptfigur kleine Streifen, die gar keine Texturen hatten. Quake war das erste Spiel überhaupt, das Texture-Skinning verwendet hat. Zuvor gab es lediglich einfarbig gefüllte Polygone. Dave Taylor hat damals den Code für die 3D-Modelle geschrieben und Adrian Carmack die Grafiken erstellt. Für Ouake mussten wir lernen, wie man Modelle in 3D erstellt und animiert. Wir mussten herausfinden, wie wir sie in die 3D-Umgebung integrieren können. Dazu kam dann auch noch die Beleuchtung. Es war einfach alles neu, da es komplett in 3D war, und trotzdem war das Spiel sehr Moddingfreundlich.

Make Magazin: In deinem Buch beschreibst du, dass du heute Kindern das Programmieren beibringst und sogar ein-



Das Team von id Software im Jahr 1992 (von I. nach r.): John Carmack, Kevin Cloud, Adrian Carmack, John Romero, Tom Hall und Jay Wilbur.

mal ein Online-Rollenspiel mit Lerninhalten für Kinder entwickelt hast.

John Romero: Ja, das war ein wirklich großes MMO (Massively Multiplayer Online Game). Es ist traurig, dass es nie erschienen ist. Wir hatten ein Beraterteam mit Mathematiklehrern aus der ganzen Welt, die uns dann vorgaben, welche universellen Fähigkeiten Kinder auf der ganzen Welt lernen sollten. Alle waren unglaublich begeistert von dem Spiel, weil es Möglichkeiten der Darstellung hatte, die es in Büchern nicht geben kann. Das Ziel des MMOs bestand übrigens nicht darin, Kindern etwas beizubringen, sondern sie mit speziellen Denkprozessen vertraut zu machen.

Make Magazin: Interessierst du dich auch für Dinge wie Physical Computing? Kinder sind ja immer sehr angetan, wenn sich etwas nicht nur auf dem Bildschirm, sondern auch in der realen Welt bewegt.

John Romero: Ja, hier gibt es einen Verein namens "Coder Dojo" und mein Sohn und ich halten dort viele Vorträge für Kinder. Meine Frau und ich veranstalten zudem interne Workshops mit dem Raspberry Pi für eine große Edge-Computing-Firma namens Fastly. Übrigens: Roger Wagner, der das berühmte Buch "Assembly Lines" verfasst hat, mit dem auch ich damals Assembler gelernt habe, ist ein totaler Raspberry-Pi- und Arduino-Freak. Er veranstaltet ständig Workshops für Kinder und bringt ihnen das Programmieren und Basteln mit Elektronik bei.

Make Magazin: In deinem Buch berichtest du auch von Plänen, ein eigenes Videospiel-Museum aufzumachen.

John Romero: Ja. Irgendwann werden wir das tun. Wir haben ein unglaubliches Archiv. Kennst du Nasir Gebelli, der damals unglaubliche Sachen programmierte? Ich habe seinen Apple II Plus, auf dem er Anfang der 80er-Jahre entwickelt hat, als er auch am ersten Teil von "Final Fantasy" arbeitete. Er hatte ein Nintendo Entertainment System, das er an den Apple II anschließen konnte. Er schrieb den Code auf dem Apple II und lud ihn dann auf das NES, um ihn zu testen. Neben Teil 1 bis 3 von "Final Fantasy" hat er auch "Secret of Mana" für das Super Nintendo geschrieben und ich habe auch seinen SNES-Hardware-Debugger, den er dafür verwendet hat. Und das ist noch lange nicht alles. Ich habe im Archiv auch die gesamte Hard- und Software von Bob Bishop, einem der ersten Apple-Mitarbeiter, und einen schwarzen Vader-Apple-II-Prototyp.

Make Magazin: Dieser schwarze Apple II wurde speziell für den Bildungsmarkt entwickelt, oder?

John Romero: Ja, er war nur für Schulen gedacht. Ich habe, wie gesagt, den Prototypen, von dem es weltweit nur ein Exemplar gibt. Er kommt auch von Bob. Daniel Kottke, der ebenfalls bei Apple an der Entwicklung des Apple II beteiligt war, sah einmal das Gerät bei mir zu Hause stehen und sagte: "Oh mein Gott, ich kann nicht glauben, dass ich das noch einmal sehe." Bob und er haben gemeinsam daran gearbeitet und jetzt habe ich es in meinem Museum.

Make Magazin: An welchem Projekt arbeitest du momentan?

John Romero: Ich erstelle gerade gemeinsam mit einem Team von ca. 60 Leuten einen First-Person-Shooter in Unreal 5. Er sieht fantastisch aus und macht super Spaß. Ich bin wirklich begeistert von dem Ergebnis und wir haben auch einen wirklich großen Publisher im Hintergrund. Daran arbeite ich also tagsüber. In der Nacht arbeite ich dann an Doom, erstelle Level oder Doom-Mods oder was auch immer. Und meine Frau Brenda hat mir gerade erzählt, dass, wenn sie einmal an den Punkt kommt, an dem sie ihr letztes Spiel macht, es ein Assembler-Apple-II-Spiel sein wird. Anfang und Ende auf derselben Plattform. -das



John Romero (hier 2014) programmiert immer noch Computerspiele und bastelt in seiner Freizeit an D00M-Leveln, die er über seine Website romero.com vertreibt (siehe Link in der Kurzinfo).



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.

In heise shop

Portraitobjektiv aus Holz

Back to the roots: Aus einfachsten Materialien wird ein Portraitobjektiv gebaut, das seinen eigenen Charakter und eine besondere Haptik hat. Zusätzlich werden die Grundprinzipien der Optik erforscht und erläutert.

von Erik Golz



A nfang des Jahres waren wir in geselliger Runde, als jemand leicht spöttisch sagte: "Drechsle doch mal ein Objektiv aus Holz!" Nach kurzem Nachdenken war meine Antwort: "Kein Problem, ich drechsle ein Portraitobjektiv aus Holz und du stehst Modell." Kurzes Gelächter am Tisch und die Herausforderung stand.

Einige Wochen ruhte die Idee, dann nahm ich mich der Sache an. Zwei Komponenten kann man nicht aus Holz herstellen, zum einen den Anschluss an die Kamera, dafür ist das Bajonett zu filigran. Hier musste ich auf ein bestehendes Bauteil aus einem alten Objektiv zurückgreifen. Zum anderen hat Holz auch keine optischen Eigenschaften, irgendeine Sammellinse geeigneter Brennweite musste also als lichtbrechendes Element dienen.

Das Projekt schien interessant. Wie minimalistisch kann ein optischer Aufbau sein? Reicht eine einfache Sammellinse für vorzeigbare Portraitaufnahmen? Optische Höchstleistung war nicht zu erwarten, ganz im Gegenteil. Trotzdem: Kommt hinterher nur Matsch heraus oder eine samtige Unschärfe wie beim legendären Imagon-Weichzeichner-Objektiv, das mit kontrollierter Dosis von optischen Abbildungsfehlern einen eigenen unvergleichbaren Look erzeugt?

Grundsätzliche Überlegungen

Das Auflagemaß (baulicher Abstand zwischen Aufnahmechip und Bajonettanschluss) einer spiegellosen Systemkamera beträgt je nach Hersteller ca. 20 mm, bei Spiegelreflexkameras etwa 40 mm bis 45 mm. Da ich aktuell mit Sony und Canon fotografiere, entschied ich mich, das Objektiv auf Canon EF (Auflagemaß 44 mm) auszulegen, um es später mittels Adapter ebenso für das spiegellose Sony E-Mount verwenden zu können.

Ein Portraitobjektiv hat etwa die doppelte Normalbrennweite, bei Vollformat also ca. 100 mm. Die Differenz zwischen Brennweite und Auflagemaß, hier also ca. 56 mm, würde ausreichend Platz für die mechanische Konstruktion bieten. 10 mm Auszug würden fürs Fokussieren reichen (Herleitung siehe Kasten "Optik und Auslegung"). Eine Sammellinse mit 100 mm Brennweite und der Möglichkeit, diese im Abstand von 100 bis 110 mm vor dem Chip zu verschieben, wäre daher die Minimalanforderung für das selbst gebaute Objektiv.

Es musste also nur noch die passende Sammellinse gefunden werden. Nahlinsen, wie sie in der Makrofotografie verwendet werden, sind allesamt Sammellinsen. Ihre Dioptrienzahl (dpt) ist der Kehrwert der Brennweite (in Meter gerechnet). Eine Brennweite von 100 mm, also 0,1 m, entspricht daher 10 Dioptrien. Eine solche Nahlinse mit +10 dpt hatte ich nicht in meinem Fundus, aber ich

Kurzinfo

- » Holz als Hauptmaterial
- » Verwendung einer einfachen Nahlinse als bildgebendes Element
- » Optische und mechanische Auslegung

Checkliste Material Zeitaufwand: » Nahlinse + 10 dpt » Skylightfilter aus der Grabbelkiste / Ein Wochenende Flohmarkt Kosten: » Defektes Objektiv oder Zwischenring » Pappe, Alufolie, schwarzer Filz Etwa 50 Euro ohne Holz » Holzrohling (kein Nadelholz), D = 90 mm, Gesamtlänge 200 mm Mehr zum Thema Werkzeug » Martin Spendiff, Photon: Open-Source-Belichtungsmesser, » Drechselbank mit Drehzahlregelung Make 1/24, S. 78 » 4-Backenfutter mit 100-mm-Standardbacken » Jan Peter Kuhtz, Mit dem Handhobel und 43-mm-Haibacken Holz bearbeiten, Make 6/22, S. 104 Bohrfutter oder Spannzangen » Daniel Springwald, Im Maker-Fotostudio, Make 4/22, S. 100 für den Reitstock Drechselwerkzeuge Schruppröhre, » Carsten Meyer, Günstiger Studioblitz, Spindelformröhre, Ovalmeißel, Spitzschaber, Make 4/21, S. 104 Flachschaber » Forstnerbohrer 45 mm, 50 mm, 65 mm



fand ein preiswertes Set mit 4 Nahlinsen mit

+2, +4, +8 und +10 Dioptrien für 30 Euro im

mit +8 Dioptrien hätte ich ebenso verwenden

Die im Set ebenfalls vorhandene Nahlinse

Versandhandel.

können, da die Brennweite von 125 mm recht nah an den geplanten 100 mm war.

» Persönliche Schutzausrüstung

Als Durchmesser der Nahlinsen wählte ich aus zwei Gründen 67 mm: Zum einen erlaubte der große Durchmesser eine robustere Kon-



Bild 1: Aufbau des Objektivs mit 10-dpt-Linse

Projekt



Bild 2: Werkstattzeichnung

struktion und ich hatte noch aus einem anderen Projekt einen 65-mm-Forstnerbohrer zur Hand, der mir an der Drechselbank viel Arbeit ersparen würde. Zum anderen passten die Nahlinsen mit 67 mm auf viele meiner Objektive, sodass die Ausgabe bei einem Fehlschlag des Bauprojekts nicht ganz umsonst gewesen wäre.

Nachteilig am großen Durchmesser war allerdings, dass größere Linsendurchmesser auch dickere Linsen bedingen und dickere Linsen mehr Linsenfehler als dünnere haben. Bei einem kleineren Durchmesser wäre also eine bessere Abbildungsqualität zu erwarten. Diesen Nachteil nahm ich jedoch in Kauf.

In der Schublade lag noch ein einfaches 80–200-mm-Zoomobjektiv mit defekter Blende bzw. Blendenelektronik. Da hier eine Reparatur nicht wirtschaftlich war, zerlegte ich es und nutze den Bajonettanschluss als Bauteil für mein geplantes Holzobjektiv. Alternativ hätte ich auf einen meiner Zwischenringe oder Adapter aus dem Canon-EF- oder FD-System zurückgegriffen, die aufgrund meiner Leidenschaft für Makrofotografie zahlreich vorhanden sind.

Als Konstruktion (Bild 1) schwebten mir drei Ringe bzw. Hülsen aus Holz vor. Der mittlere Ring (B), ausgeführt als doppelseitige durchgebohrte Dose, sollte im vorderen Teil die Nahlinse aufnehmen, die durch einen fest eingeklemmten Ring (C) gehalten werden würde. Der hintere Teil sollte als Schiebesitz den Anschlussring (A) aufnehmen, auf den der Bajonettanschluss aufgeklebt wäre. Über das Verschieben der beiden Ringe (A) und (B) gegeneinander könnte ich den Auszug einstellen und damit scharf stellen.

Meine größte Sorge war, eine Luftpumpe zu bauen, die zuverlässig den beim Verschie-



Bild 3: Holz aus dem Garten (Kirsche). Es wurde später auch Weide verwendet.

ben der Ringe auftretenden Holzstaub in die Kamera und damit auf den Chip pumpte. Um das zu verhindern, überlegte ich mir, in den hinteren Ring noch eine Klarglasscheibe einzukleben, die die Konstruktion kameraseitig abdichtet. Ein 48er-Skylight-Filter aus dem Fundus, der ohne Fassung einen Durchmesser von 45 mm hatte, erwies sich als perfekt. Dieser zusätzliche Filter würde zwar die Abbildungsleistung etwas verschlechtern, möglicherweise auch durch innere Reflexionen den Kontrast mindern, doch die Abdichtung nach hinten war hier zwingend erforderlich.

Nachdem die Bauteile festgelegt waren und die prinzipielle Konstruktion stand, erstellte ich eine Skizze (Bild 2), die die wichtigsten Abmessungen als theoretische Maße wiedergab. Als Wandstärke plante ich 5 bis 10 mm, die an meine Drechselfertigkeiten nicht die höchsten Anforderungen stellen würden. Außer den Funktionsabmessungen können alle anderen Dimensionen und Formen, also auch die Wandstärke, später beim Drechseln variiert werden.

Das notwendige Spiel des Schiebesitzes sowie die Klemmung des vorderen Rings würden sich bei der Herstellung ergeben oder später nach Gefühl eingeschliffen werden.

Ursprünglich wollte ich mehrere vordere Ringe (C) mit unterschiedlichen Öffnungsdurchmessern drechseln, um damit verschiedene Blenden zu erzielen. Es zeigte sich aber später, dass eine mit Pappe verstärkte eingelegte Alufolie die Funktion der Blende besser und flexibler erledigt.

Holzbearbeitung auf der Drechselbank

Nun ging es in den Garten, um geeignete und bereits länger gelagerte Holzstücke (Bild 3) auszusuchen, die ich anschließend auf eine passende Länge sägte. Danach ging es mit der Werkstattzeichnung und den Bauteilen in den Drechselkeller.
Projekt



Bild 4: Zwischen Spitzen runddrehen, beidseitig Zapfen andrehen für 100-mm-Backenfutter



Bild 5: Mittlerer Ring (B) in Standardbacken, bohren mit 65er-Forstnerbohrer



Bild 6: Hinterer Ring (A), direkt eingespannt in die 100er-Standardbacken

Schutz für Hackers Liebling



Forscher schätzen, dass in 90% der von ihnen untersuchten Fälle von Cyberangriffen, das Active Directory involviert ist! Mit dieser aktualisierten und erweiterten Neuauflage des **iX Kompakt zur AD-Sicherheit** können Sie sich dringend benötigtes Fachwissen zum Schutz vor Ransomware aneignen:

- Denken wie ein Hacker Angriffe verstehen und verhindern
- Forensische Analyse von Vorfällen und Angriffen
- Microsofts Schichtenmodell: Tiers festlegen und abschotten, privilegierte Zugriffe absichern
- Marktübersicht: Werkzeuge für die AD-Absicherung

Heft für 29,50 € • Digital für 27,99 € Heft + Digital 41,50 €



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 7: Hinterer Ring (A), Stufen gebohrt!

Bild 8: Hinterer Ring (A), umgedreht auf 43er-Haibacken, von innen gespannt

Aufgespannt zwischen 4-Zack-Mitnehmer und Mitlaufkörner (siehe Bild "Komponenten einer Drechselbank") wurden mit der Schruppröhre aus den Holzstücken einfache Zylinder gedrechselt. Jetzt zeigte sich, welches Stück brauchbar war und wie viel an nutzbarem Durchmesser und Länge jeweils zur Verfügung stand.

Als erstes Teil (Bild 4) drechselte ich den mittleren Ring (B), um ein Gefühl für die Dimension zu bekommen. Nachdem ein runder Körper mit dem ungefähren Durchmesser von 90 mm und einer Länge von etwa 120 mm vorlag, zeichnete ich die Lage des inneren Stegs sowie die vorgesehenen Bohrtiefen außen auf dem Umfang an. Die fertige äußere Kontur würde ich erst zum Schluss drechseln.

Zum Bohren ist eine sichere einseitige Aufspannung des Werkstücks auf die Drechselbank notwendig. Hierzu dienen 4-Backenfutter, die auf das M33-x-3,5-Gewinde der Antriebsspindel aufgeschraubt werden. 3-Backenfutter, wie sie auf Metalldrehbänken verwendet werden, sind fürs Drechseln ungeeignet und stellen auch ein Sicherheitsrisiko dar.

Für meine 100er-Standard-Backen, die Durchmesser ab ca. 80 mm von außen spannen können, drechselte ich an beide Enden des Holzzylinders mit dem Spitzschaber einen schwalbenschwanzförmigen Zapfen für einen





Bild 9: Aufgesetzter Bajonettring

sicheren formschlüssigen Halt und guten Rundlauf.

So vorbereitet konnte das Bohren der vorderen Aufnahme für die Nahlinse beginnen. Der 4-Zack-Mitnehmer wurde gegen das 4-Backenfutter getauscht und der eben vorbereitete Holzzylinder in die Backen gespannt. Dabei diente der Mitlaufkörner im Reitstock erst mal als Hilfe zum zentrischen Ausrichten.

Anschließend wurde die MK2-Pinole des Reitstocks gegen eine Bohrpinole mit Spannzange getauscht und in die Spannzange ein 65er-Forstnerbohrer eingesetzt (Bild 5). Es ist auch möglich, den Bohrer in ein Bohrfutter mit MK2-Schaft zu spannen und dieses in die MK2Reitstockpinole zu stecken. Bei einem Bohrdurchmesser von 65 mm treten jedoch große Kräfte auf! Daher ist hier die direkte Aufnahme des Bohrers mit Spannzange im Reitstock vorzuziehen.

Bei geringer Drehzahl von 400 bis 500 min⁻¹ bohrte ich dann bis zur auf der Außenseite markierten Solltiefe. Es ist immer wieder schön zu sehen, wie bei stehendem Bohrer und rotierendem Werkstück die Späne aus der Öffnung quellen. Dennoch sollte man nicht in einem Stück bohren, sondern in Etappen. Den Bohrer jeweils herausziehen und die sich ansammelnden Späne absaugen. Die Bohrpinole besitzt zwar eine Maßskala, trotzdem ist ge-



Bild 10: Die fertigen, eingeölten Holzringe (2 \times vorderer Ring aus Kirsche mit verschiedenen Öffnungen)

legentliches Nachmessen der Bohrtiefe mit dem Lineal empfehlenswert.

Nachdem die Solltiefe erreicht war, wechselte ich den 65er-Forstnerbohrer gegen einen mit 50 mm Durchmesser aus und bohrte für den Durchbruch im Steg weitere 20 mm tiefer.

Anschließend galt es, den gebohrten Durchmesser von 65 mm auf ca. 69 mm zu vergrößern. Hierzu nutzte ich Rezess-Schaber und gerade Flachschaber, die möglichst parallel zur Drehachse geführt wurden. Nachdem der gewünschte Innendurchmesser erreicht war, konnte die Nahlinse locker eingelegt werden. Aufpassen muss man, damit sich beim Testen nichts verklemmt!



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 11: Einkleben des Skylight-Filters in den hinteren Holzring



Objektiv mit EF-NEX-Adapter an

Nun muss die andere Seite zur Aufnahme des Anschlussrings (A) gebohrt werden. Hierzu wurde der Zylinder aus dem Backenfutter ausgespannt, um 180° gedreht und mit dem Zapfen der anderen Seite ins Backenfutter eingespannt. Jetzt erfolgte genau wie vorher erst das Bohren mit dem Forstnerbohrer und anschließend mit den Schabern das Erweitern der Bohrung auf 70 mm.

Für das Drechseln der Außenkontur und das Kürzen auf die korrekte Länge (Entfernen der Zapfen auf beiden Seiten) muss der Zylinder von innen aufgenommen werden. Normalerweise drechselt man sich dafür einen Hilfszapfen, auf den das Werkstück zur Bearbeitung geklemmt wird. Einfacher ist es jedoch, mit verlängerten Spannbacken (auch Haibacken genannt) direkt in die Bohrung zu greifen und von innen spreizend zu spannen. Die mit dem 4-Backenfutter mitgelieferten 43er-Haibacken waren dafür ideal geeignet. Beim Spreizen entstehen zwar Abdrücke (deswegen verwendet man normalerweise einen Hilfszapfen), aber in diesem Anwendungsfall stören diese nicht.

Aufgespannt auf die gespreizten Haibacken konnte ich jeweils das freie Ende bearbeiten, d. h. kürzen mit Spindelformröhre und Ovalmeißel, und dabei dem Holzwerkstück



Bild 13: Für die Verwendung an Canon-Kameras muss der Doppelpin 2/3 abgeklebt oder entfernt werden.



Bild 14: Portrait

eine schöne geschwungene Form geben. In gleicher Weise drechselte ich den vorderen Ring (C), der die eingelegte Nahlinse fixiert.

Für den hinteren Ring (A) verwendete ich ein quadratisches Reststück aus Weide (Bild 6), das ich der Einfachheit halber direkt in die 100er-Standardbacken spannte.

Zuerst drechselte ich mit der Schruppröhre grob vor, anschließend passte ich mit der Spindelformröhre und einem Meißel den Außendurchmesser und damit den Schiebesitz zur Bohrung des mittleren Ringes an. Zwischen beiden Ringen sollte so viel Spiel sein, dass eine eingeklebte dc-fix-Samtfilz-Folie ein sauberes Verschieben ermöglicht.

Mit Forstnerbohrern der Durchmesser 45 mm und 50 mm führte ich dann die stufigen Bohrungen (Bild 7) entsprechend der Werkstattzeichnung aus.

Anschließend wurde der Ring umgedreht und auf die von innen spreizenden Haibacken aufgesetzt (Bild 8). Das noch vorhandene quadratische Ende drechselte ich rund, setzte noch die rückseitige 40-mm-Bohrung und passte die Länge auf Maß an.

Jetzt konnte ich die hintere Kontur auf die Geometrie des Bajonetts aus dem zerlegten Canon-Objektiv anpassen. Nach einigen Versuchen saß der Bajonettring (Bild 9) stramm auf dem Holzwerkstück.

Damit waren die Holzringe fertig (Bild 10). Sie wurden mit Schleifpapier der Körnung 120 bis 400 geschliffen und anschließend mit Leinölfirnis geölt.

Der mittlere Ring (B) erhielt zusätzlich außen eine Schicht Carnaubawachs, aufgetragen auf das rotierende Werkstück.

Montage und Zusammenbau

Als letzter Fertigungsschritt wurde der Skylight-Filter (Bild 11) mit 2-Komponenten-Kleber in den hinteren Ring eingeklebt und der Bajonett-Anschlussring von außen auf die passend bearbeitete Form aufgeklebt. Damit war die Herstellung der Objektiveinzelteile abgeschlossen.

Test des Objektivs

Zusammengebaut und mit dem passenden EF-NEX-Adapter versehen, kann das neue Objektiv an einer spiegellosen Sony-Kamera (Bild 12) verwendet werden. Dazu muss die Kamera auf "Auslösen ohne Objektiv" eingestellt sein.

Für die Verwendung an Canon-Kameras muss zusätzlich am Objektiv-Bajonett der Doppelpin 2/3 (Bild 13), der der Kamera das Vorhandensein eines Canon-Objektivs meldet, abgeklebt oder besser ganz entfernt werden. Sonst interpretiert die Kamera das fehlende elektrische Innenleben des Holzobjektivs als "verschmutzte Objektivkontakte".

Die Bohrung des vorderen Halterings von 50 mm entspricht einer Blende von f2, eine fantastische Lichtstärke. Fantastisch waren aber auch die Linsenfehler, die das ganze Bild in Unschärfe und in eine nebelartige Kontrastarmut tauchten (siehe Kasten "Optik und Auslegung").

Erstaunlicherweise war zwar massiv sphärische Aberration vorhanden, also unterschiedliche Lichtbrechung von Zentrums- und Randstrahlen, die Unschärfe verursachte, jedoch wenig bis keine chromatische Aberration, also grüne oder violette Farbsäume an harten Kontrastkanten.

Gegen Ersteres half es, die Randstrahlen zu begrenzen, also mit einer Blende nur noch Zentrumsstrahlen auf den Chip fallen zu lassen. Mit kleiner werdender Blende stieg auch der Kontrast und als letztes Mittel zur Kontrasterhöhung gab es ja noch hartes Licht bei der Beleuchtung und die Tonwertkorrektur in Lightroom.

Ein direkt hinter die Nahlinse platzierter Karton mit einer Öffnung von ca. 15 mm (entspricht etwa Blende f6,7) führte zu brauchbaren Ergebnissen. Nicht scharf, aber auf dem richtigen Weg. Eine Alufolie, in die ich mit einem Bürolocher ein 5-mm-Loch stanzte, lieferte bei einer effektiven Blende im Bereich um f22 gute Ergebnisse, wie die Portraitaufnahmen zeigen.

Die Fotos haben keine Kernschärfe für "Pixelpeeper" in 100-Prozent-Ansicht, aber eine cremige Weichheit, teils bedingt durch die noch vorhandenen restlichen Linsenfehler, teils bedingt durch die bei f22 schon deutlich eintretende Diffraktion (Beugung an der Blende). Alles in allem zeigte sich in der Abbildung ein brauchbares Zentrum bei verschmierten Ecken. Und natürlich löste die Nahlinse qualitätsmäßig nicht die ganzen 24 Megapixel der Sony-Kamera auf: Aber Abzüge bis 20 cm×30 cm wären schon ansehnlich.



Bild 15: Portrait

Alles in allem ein Objektiv mit Charakter, bedingt durch die verwendeten Materialien und durch die verbliebenen Restfehler.

Praktischer Einsatz

Das jährliche Treffen der User der "de.rec.fotografie"-Newsgroup war der richtige Rahmen, das Objektiv auszutesten. Stark abgeblendet, mit einer leistungsfähigen Blitzanlage am Set, konnten stimmungsvolle Portraits (Bilder 14 und 15) erstellt werden.

Konsequenterweise in Schwarz-Weiß: Damit wurde die Dramatik noch verstärkt und es entstand fast ein Analog-Gefühl wie aus dem letzten Jahrhundert.



Bild 16: Industrieruine in Farbe

Aber auch in Farbe konnte ich gute Ergebnisse erzielen, z. B. von einer Industrieruine (Bild 16). Langsames Arbeiten, sorgfältiges Fokussieren, leichte Nachbearbeitung in Lightroom (Weißabgleich und Tonwertanpassung) – mehr brauchte es nicht.

Lohnt der Nachbau?

Für einen Maker auf jeden Fall. Back to the roots, ausprobieren, was geht. Die Grenzen ausloten und die Ergebnisse optimieren. Auch für Drechsler ist es mal ein anderes Projekt, denn Drechseln ist mehr, als nur Schalen und Pfeffermühlen herzustellen.

Als Ersatz für ein klassisches Objektiv war dieses Projekt nie gedacht. Zu groß sind die Einschränkungen im praktischen Einsatz. Kein Autofokus, nicht einmal ein Schneckengang zum Scharfstellen. Stattdessen ein Schiebesitz, mit dem man stark abgeblendet den optimalen Schärfepunkt finden muss. Der lichtverstärkende elektronische Sucher einer spiegellosen Systemkamera hilft dabei sehr, mit dem stark abgedunkelten Sucherbild im optischen Sucher einer DSLR ist das sehr schwierig. Blende 22 ist auch nicht die Blende, mit der man jeden Tag arbeiten möchte. Und über das Bokeh, also die Qualität des Unschärfebereichs, braucht man bei f22 eigentlich auch nicht lange zu reden.

Wer also nur ein passendes Objektiv in diesem Brennweitenbereich haben möchte, ist mit einem Vintageobjektiv aus dem Secondhandhandel besser bedient. Trotzdem gibt es Potenzial weiterzumachen. Mein Ansatz ist, mit höherwertigen Nahlinsen eine bessere Kernschärfe zu erreichen und die Blendeneinsätze zu variieren. So kann zum Beispiel statt einer kreisförmigen Blende eine Siebblende verwendet werden. Um dann vielleicht Ergebnisse wie mit dem legendären Imagon-Weichzeichner zu erzielen.

Wenn es mal etwas anderes sein soll und man die verblüfften Augen von Freunden und Fotografenkollegen genießen möchte, ist ein Holzobjektiv schon eine tolle Sache. Die drei Holzringe sind in der Herstellung nicht sehr komplex und stellen auch keine besonderen Anforderungen an die eigenen Drechselfähigkeiten. Vielleicht mal im Bekanntenkreis oder in Nachbarschaftsforen nachfragen, ob diese jemand herstellen kann, falls man keine eigene Drechselbank besitzt.

Alternativ sollten die Ringe und Hülsen auch mit dem 3D-Drucker gedruckt werden können (ich habe es nicht selbst probiert), wobei die Wandstärken entsprechend angepasst werden können. Hier wäre es interessant, den Bajonettanschluss direkt mitzudrucken, entsprechende 3D-Modelle des gewünschten Bajonetts gibt es auf Thingiverse (etwa nach "extension tubes" suchen). Auch eine Kombination mit dem Holzkorpus ist denkbar: Also Anschlussring (A) mit Bajonett gedruckt, der Rest aus Holz gefertigt. Welche Materialien Sie auch einsetzen, das Basteln und das spätere Fotografieren machen einfach Spaß. —*caw*

Optik und Auslegung

Ein unendlich weit entferntes Objekt wird von einer idealen Sammellinse in der Fokusebene abgebildet, d. h. im Abstand der Brennweite f. Ist das Motiv näher, liegt die Schärfeebene hinter der Fokusebene. Der Aufnahmechip oder der Film müssen also nach hinten verschoben oder die Linse um den Auszug e nach vorne verschoben werden, um ein scharfes Bild zu erhalten.

Aus der Linsengleichung

$$\frac{1}{f} = \frac{1}{b} + \frac{1}{g}$$

und dem daraus abgeleiteten Maßstab

 $M = \frac{B}{G} = \frac{b}{g} = \frac{b-f}{f} = \frac{e}{f}$

kann der notwendige Auszug

$$e = M * f$$

berechnet werden.

Bei einem Maßstab von 1:10 (M = 0,1), z. B. der Abbildung eines 24 cm hohen Kopfes auf 24 mm Bildhöhe im Querformat, beträgt bei einer Brennweite von 100 mm der notwendige Auszug 10 mm. Diese 10 mm Verschiebeweg ermöglichen die Fokussierung im Bereich von unendlich bis ca. 1 m Naheinstellgrenze.



Leider ist die verwendete Nahlinse optisch alles andere als ideal, sondern voller Linsenfehler. Am stärksten wirkt sich hier die sphärische Aberration aus, bei der Strahlen, die durch die Randbereiche der Linse verlaufen (in der Skizze rot eingezeichnet), anders gebrochen werden als solche, die durch das Zentrum gehen. Als Resultat überlagern sich unterschiedliche Schärfeebenen mit der Folge von genereller Unschärfe und geringem Kontrast.

Die Blende reguliert die auf den Chip fallende Lichtmenge und beeinflusst die Schärfentiefe. Aber sie schneidet auch Randstrahlen ab und verbessert damit die optische Leistung der Nahlinse.

Bei modernen Objektiven werden Linsenfehler in der optischen Konstruktion durch die Kombination mehrerer Einzellinsen weitestgehend kompensiert. Abblenden führt daher nur zu geringer Verbesserung der optischen Leistung. Bei der hier verwendeten Nahlinse aber gilt: Viel hilft viel, daher die kleine Blende. Eventuelle negative Effekte zu kleiner Blenden sind jedoch wahrscheinlich gegenüber den massiven vorhandenen Linsenfehlern vernachlässigbar.

WILLKOMMEN IM NEUEN Iot-ökosystem

Mit LoRaWAN und C-Programmierung über lange Distanzen messen und steuern



Im Make Special LoRaWAN führen Sie 15 Artikel Schritt für Schritt in die Hardware, LoRaWAN, The Things Network und ihre Programmierung ein. Es wird Schrittfür-Schritt erklärt, wie Sie aus den mitgelieferten LoRaWAN- und Sensormodulen einen Umweltsensorknoten entwickeln und noch vieles mehr.

DARUM GEHT'S:

- Einstieg in STM-Mikrocontroller
- Programmieren mit der STM32CubeIDE
- Spannungen, Temperatur und Luftfeuchte messen
- Mit LoRaWAN senden und empfangen
- Daten im The Things Networks verarbeiten
- Werte mit TagoIO visualisieren
- Refresher: Programmieren in C



Make Special LoRaWAN inkl. Experimentierset für 64,90 €

🚽 shop.heise.de/make-lorawan24



Funktechniken fürs Smart Home

Kein Smart Home funktioniert ohne die drahtlose Datenübertragung von Messwerten oder Befehlen. Hier erfahren Sie, welche Funktechniken dafür zur Verfügung stehen und welche Vor- und Nachteile sie jeweils haben.

von Heinz Behling



n einem Smart Home wird alles irgendwie automatisch gesteuert, geregelt und geschaltet. Das ist so ungefähr die grobe Vorstellung, die der Mann/die Frau auf der Straße von der Hausautomatisierung hat. Und, dass man sich dann um nichts mehr selbst kümmern muss.

Nur relativ wenige wissen, welch ein Verkehr in einem solchen automatischen Heim wirklich herrscht, genauer gesagt: Datenverkehr. Würden alle Informationen und Anweisungen im Smart Home per menschlicher Sprache von den zahlreichen Sensoren zum Server (meist ein Raspberry Pi) und umgekehrt übertragen, käme bei all dem Geschrei kaum noch iemand zur Ruhe. Da müssen beispielsweise Messwerte wie Temperatur und Luftfeuchtigkeit in regelmäßigen Abständen an den Smart-Home-Server gesendet werden. Bei Abweichungen von den Sollwerten muss der Server dann die Heizkörper entsprechend aktivieren oder drosseln. Und das Ganze jeweils für jeden Raum des Hauses. Soll die Raumbeleuchtung automatisch geschaltet werden, müssen Bewegungssensoren laufend registrieren, ob Personen im Leuchtbereich der jeweiligen Lampe sind und ein Helligkeitssensor muss ermitteln, ob es überhaupt dunkel ist. Aus diesen Daten wiederum generiert der Server bei Bedarf Steuerbefehle für die Leuchtkörper.

Dazu kommen dann noch Heimkinoanlagen, motorisierte Rollos, Rollläden, Alarmanlagen, Kameras, Garagentore und, und, und ... All diese Geräte kommunizieren ständig miteinander. Sogar, wenn es scheinbar gar nichts zu melden gibt, geben alle Komponenten in regelmäßigen Abstanden zumindest ein Lebenszeichen von sich, das dem Server und anderen Smart-Home-Kollegen meldet: Ich bin noch da.

Ein Smart Home braucht für diese Datenübertragung also Funktechnik. Und da gibt es einige Verfahren, die jeweils Vor- und Nachteile haben. Mal geht es um hohe Reichweite, bei anderen Anwendungen um möglichst geringen Energieverbrauch, damit die Batterie lange durchhält. Auch der Datenschutz spielt eine Rolle, und Argumente wie Zukunftssicherheit, Preis oder Unabhängigkeit von einem Hersteller oder einer Cloud spielen ebenfalls wichtige Rollen.

Die folgende Übersicht zeigt, welche Smart-Home-geeigneten Funktechniken es gibt und nennt deren Vorzüge und Nachteile.

WLAN

WLAN (**W**ireless **L**ocal **A**rea **N**etwork) ist sicher die bedeutendste Computerfunktechnik, nicht nur im Smart Home. Die verwendeten Frequenzbereiche 2,4 GHz/5 GHz führen zu Reichweiten unter günstigen Umständen bis zu mehreren Hundert Metern. In Gebäuden

Kurzinfo

- » Vor- und Nachteile von neun Smart-Home-Funktechniken
 » Stromsparende Funktechniken für Batteriebetrieb
- von Sensoren
- » Wie steht es um Matter?

Mehr zum Thema

- » Heinz Behling: Controllerboards fürs Smart Home, Make 4/23, S. 12
- » Heinz Behling: Solarstrom für Mikrocontroller, Make 4/22, S. 32
- » Heinz Behling: Smart Home: Heizung kontrollieren und Energiekosten senken, Make 2/21, S. 22





Bild 1: WLAN-fähige Mikrocontroller-Boards gibt es in Hülle und Fülle: hier zum Beispiel ein ESP01 (links oben), daneben ein Wemos D1 mini, beide mit einem ESP8266. Darunter ein Lolin 32 mini mit Akku-Anschluss und ESP32.

sind es aber meist deutlich weniger (20 bis 70 m), wobei die Reichweite bei 5-GHz-Netzen etwas höher ist.

Die Datenübertragung erfolgt relativ sicher verschlüsselt, wenn als Verschlüsselungsverfahren WPA2 oder 3 eingestellt ist und Benutzernamen und Passwort geheimgehalten werden. Die Datenraten sind hoch, sodass auch große Datenmengen im Smart Home (Videokameras) problemlos übertragen werden. Allerdings ist auch der Strombedarf von WLAN-Geräten recht hoch. Beim Senden kommen da schonmal 3 bis 5 W zusammen, weshalb sie meist am Stromnetz betrieben werden. Die zahlreichen Mikrocontroller-Boards für Selbstbaugeräte liegen in derselben Größenordnung. Daher sind sie für einen Batteriebetrieb kaum geeignet. Akkubetrieb ist möglich, wenn für eine ausreichende Akkukapazität und regelmäßiges Nachladen ge-

Reichweite von Bluetooth-Geräten				
Klasse	max. Sendeleistung	Reichweite im Freien	Reichweite in Gebäuden	
1	100 mW	> 100 m	50 m	
2	2,5 mW	50 m	10 m	
3	1 mW	10 m	1 m	

sorgt wird. Das kann beispielsweise durch eine kleine Photovoltaikanlage geschehen (siehe Artikel in Kurzinfo).

WLAN-Netze haben eine Sternstruktur, d. h., alle Netz-Clients haben direkt Verbindung zum Accesspoint, in der Regel ist das der Router im Smart Home. Die Reichweite eines Accesspoints kann aber durch Repeater erhöht werden. Diese haben zum einen mit dem Router Kontakt, spannen aber auch als Accesspoint wiederum ein eigenes WLAN auf, in dem sich dann wiederum Clients einloggen können. Bei 5-GHz-Netzen gibt es eine Besonderheit: Teile dieses Frequenzbereichs (Kanäle 52 bis 140) werden auch von bevorrechtigten Usern wie Flugsicherung und Militär benutzt. Der Accesspoint, beispielsweise eine Fritzbox, muss daher laufend diesen Frequenzbereich überwachen. Beim Erkennen eines Radarbetriebs in der Nähe (Radar Detection) muss(!) der Router die betreffenden Frequenzen sofort freigeben und darf sie vor Ablauf einer Stunde auch nicht wieder benutzen. Der Router wechselt dann auf einen anderen Kanal (36 bis 48) und teilt dies als letzte Amtshand-



Bild 2: Falls es mit der Bluetooth-Verbindung Probleme gibt, kann der Einsatz eines externen Bluetooth-USB-Sticks (hier ein UD-100 G03) zusätzlich zum im Server eingebauten Chip deutliche Verbesserung bringen.



Bild 3: Heizkörperthermostate von eq3 und Raumthermometer von Xiaomi arbeiten mit Bluetooth BLE, da damit die Batterien recht lange halten. lung auf der bislang genutzten Frequenz allen Clients mit, damit die ebenfalls wechseln. Repeater machen dies jedoch nicht. Daher kann eine WLAN-Verbindung in diesem Frequenzbereich ohne Vorwarnung unterbrochen werden, was fürs Smart Home äußerst ungünstig ist. Abhilfe bringt es, wenn fürs Repeater-WLAN ein Kanal im 5-GHz-Bereich fest eingestellt wird, der außerhalb des Radarbereichs liegt.

Bluetooth/BLE/Mesh

Auch Bluetooth arbeitet im 2,4-GHz-Bereich. Allerdings ist die Reichweite meist erheblich geringer. In Gebäuden kommt man bei Verwendung eines Raspberry Pi als Server oft nur etwa 10 m weit. Die Reichweite hängt von der Geräteklasse ab (siehe Tabelle). Sie lässt sich durch gute externe Class-1-Bluetooth-Sticks (z. B. den UD-100-G03 von Parani in Bild 2) am Server verbessern, da die Class 1 mit bis zu 100 mW Sendeleistung arbeitet und die Empfänger dieser Sticks erfahrungsgemäß auch deutlich empfindlicher sind.

Eigentlich zunächst unabhängig entwickelt wurde BLE (Bluetooth Low Energy), eine Funktechnik, die deutlich weniger Strom verbraucht. Seit Bluetooth 4 ist BLE Bestandteil der Bluetooth-Spezifikation. Die Sendeleistung ist auf maximal 10 mW beschränkt, oft wird aber weniger verwendet, um Strom zu sparen (Batteriebetrieb). Daher ist die Reichweite niedrig (max. 10 m). Das reicht für kleinere Wohnungen meist aus. Längere Verbindungen können in Home Assistant beispielsweise erreicht werden, indem ein Mikrocontroller als Hub eingesetzt wird. Der führt auf der einen Seite die Kommunikation mit Bluetooth-Geräten durch, leitet den Datenverkehr aber per WLAN an den für Bluetooth zu weit entfernten Server weiter.

BLE wird gerne in Raumthermometern, Heizkörperthermostaten und ähnlichen Smart-Home-Komponenten eingesetzt, da diese Geräte nur wenige Daten übertragen müssen, aber möglichst lang ohne Batteriewechsel auskommen sollen.

Die Datenraten liegen bei Bluetooth bei knapp 1 MBit/s, bei BLE sind es 0,27 MBit/s. Daher ist Bluetooth für Videoverbindungen und ähnlich hohes Datenaufkommen nicht geeignet.

Zigbee

Zigbee wird im Smart-Home-Bereich besonders bei kommerziellen Geräten eingesetzt. So benutzen beispielsweise Ikea mit seiner Tradfri-Reihe und Philips Hue diesen Standard. Viele Hersteller bieten Zigbee-Gateways an, die der Server zur Kommunikation mit den Zigbee-Geräten braucht. Allerdings schränken die Hersteller ihre Gateways oft so ein, dass sie nur mit den Komponenten des gleichen Herstellers zusammenarbeiten. Diese Einschränkung gibt es nicht bei einem Zigbee-USB-Stick wie dem in Bild 4 gezeigten.

Obwohl die Sendeleistung mit 6 mW und damit der Stromverbrauch sehr gering sind, kann man mit Zigbee enorme Reichweiten von deutlich über 100 m erreichen. Das liegt daran, dass Zigbee Mesh-fähig ist. Erklärung: Das Zigbee-Gateway des Servers in Smart Homes kann sich mit maximal 32 weiteren Geräten verbinden. Jedes dieser Geräte kann sich aber wiederum ebenfalls mit 32 anderen Zigbee-Komponenten verbinden usw. Diese Verbindungen bauen die einzelnen Geräte automatisch untereinander auf, sobald sie in Betrieb gehen. Sie weben also bildlich gesprochen ein Netz (engl. mesh), das man sich in Home Assistant sogar anzeigen lassen kann (Bild 5). Fällt im laufenden Betrieb ein Zigbee-Gerät aus (z. B. wegen leerer Batterie), bauen die anderen das Netzwerk automatisch um. Zigbee-Verbindungen sind daher sehr zuverlässig.

Über dieses Netz kann jedes Gerät mit dem Server und umgekehrt kommunizieren. Zigbee-Netze können daher recht groß werden, die Reichweite ist dann zwar von Gerät zu Gerät klein (< 10 m), aber durch die Vernetzung können nahezu beliebige Strecken überwunden werden.

Die Datenübertragung ist mit 250 KBit/s gering, daher ist dieses Verfahren nur für die Messwert- und Befehlsübermittlung geeignet. Da der Strombedarf aber ebenfalls sehr niedrig ist, halten batteriebetriebene Zigbee-Geräte lange durch. Bewegungsmelder aus dem Ikea-Programm kommen beispielsweise mit zwei 2032-Knopfzellen etwa ein Jahr lang aus.

Inzwischen gibt es auch Mikrocontroller-Boards mit Zigbee für Selbstbauprojekte (Bild 6).

Thread/Matter

Matter ist ein Netzwerkprotokoll, das auf dem Funkstandard Thread basiert. Thread arbeitet im selben Frequenzbereich wie Zigbee und weist auch beim Energiebedarf ähnliche Werte auf.

Matter wurde vor einigen Jahren als das künftig einzig seligmachende Funknetzsystem für Smart Homes angekündigt, denn es sollte alle bisherigen Standards und herstellerspezifischen Eigenentwicklungen ersetzen. Schließlich sollten so dann alle Smart-Home-Geräte, von welchem Hersteller sie auch stammen, direkt zusammenarbeiten können.

Darauf folgten auch zahlreiche Ankündigungen über neue, Matter-fähige Geräte oder Updates, um bereits vorhandene Hardware Matter-fähig machen zu können. Geschehen ist aber erheblich weniger. So wartet beispielsweise das von Ikea bereits seit mehr als zwei Jahren erhältliche und als Matter-fähiges Gerät angepriesene Hub Dirigera bis heute auf die



 Date Breedwater, Aprick Queue, Lessendern Egnin, Brit, Brit, Brit, Brit, Brit, Brit, Statu, Brit, Br







Bild 6: Wie bei WLAN gibt es auch für Zigbee Controllerboards (CC2530) für Selbstbauprojekte.



Bild 7: Die zurzeit erhältlichen Matter-fähigen Controllerboards haben einen ESP32-Chip, hier ein ESP32-H2. Er ruht auf einem Dirigera von Ikea, einem Hub, das Matter-fähig sein soll, sobald die Firmware fertig ist.

endgültige Matter-Firmware. Vor einigen Wochen gab es lediglich eine Beta-Version, die aber nur Ikea-Lampen steuern kann. Es gibt aber auch Hersteller wie Apple, die die Sache ernst nehmen: So sind aktuelle iPhones Matterfähig und für einige ältere gibt es Updates. Matter-Geräte lassen sich auch in Apples eigenem Homekit einbinden. Meiner Meinung nach wird sich aber an der zögerlichen Unterstützung vieler Hersteller nicht viel ändern, denn warum sollten sie sich freiwillig einer erheblich größeren und oft billigeren Konkurrenz aussetzen? Schließlich möchten sie ja berechtigterweise mit ihren Produkten Geld verdienen. Wenn sie aber durch den Einheitsstandard mit sämtlichen



Billigprodukten und sogar Selbstbaulösungen konkurrieren müssen, ist das für die Firmen nicht mehr lukrativ. So stiegen auch bereits erste Firmen aus der Matter-Entwicklung wieder aus.

Am meisten tut sich da noch bei den für den Selbstbau geeigneten Mikrocontroller-Boards. Es gibt bereits einige auf ESP32-H2/3-Basis (Bild 7), für die man sich mithilfe von Home Assistant und ESPHome problemlos eine Firmware programmieren kann.

Zur Technik: Matter benutzt in Europa zwei Frequenzbereiche (868 und 2400 MHz) mit unterschiedlichen Übertragungsraten (20 KBit/s bzw. 258 KBit/s). Daher eignet es sich nur für die Messwertübermittlung ähnlich wie Zigbee. Die Reichweite, aber auch der Stromverbrauch sind gering, was bei einer Sendeleistung von 1 mW nicht verwundert. Aber Matter ist Mesh-fähig, kann daher die Reichweite wie Zigbee durch Vernetzung vergrößern.

DECT

Viele kennen DECT (**D**igital **E**nhanced **C**ordless **T**elecommunication) nur vom drahtlosen Telefon her. Aber über DECT, genauer DECT ULE (**U**Itra Low **E**nergy) können auch Smart-Home-Komponenten kommunizieren. In Deutschland macht das beispielsweise AVM, die Firma, die auch die beliebten Fritzboxen herstellt. Diese Router haben übrigens schon alles an Board, um direkt auf deren Web-Oberfläche DECT-Geräte zu steuern. Die Geräte lassen sich aber auch über Home Assistant völlig ins Smart Home integrieren.

Das Angebot an Komponenten ist überschaubar: Erhältlich sind Schaltsteckdosen (Bild 8) und Heizkörperthermostate. Außerdem gibt es von Becker ein Rollladen-Antriebssystem. Für Selbstbauprojekte ist DECT ULE kaum geeignet, da die erforderlichen Kontroller für Privatpersonen nicht frei erhältlich sind.

Zur Technik: Da DECT ULE auf dem Telefonstandard beruht, ist es, was Reichweite (bis 40 m), Datenrate und -sicherheit angeht, ähnlich.

EnOcean

EnOcean ist eine batterielose Funktechnik für Sensoren. Ja, tatsächlich kommen die Sensoren der gleichnamigen Firma ohne Stromversorgung aus. Keine Angst, hier wird es nicht paranormal, da läuft auch kein Hamster in der Rolle, um Strom zu produzieren: EnOcean hat zum eine Elektronik mit extrem geringem Strombedarf für seine Sensoren (z. B. Schalter wie in Bild 9) entwickelt, zum anderen verwenden sie Energy Harvesting, um den benötigten Strom gratis aus der Umgebung zu beziehen.

So benutzen sie beispielsweise die Bewegungsenergie des Tastendrucks, um mithilfe von Piezo-Elementen genug Strom für den Funkschalter-Sender zu gewinnen. Da der ja nur ein sehr kurzes Signal senden muss (1 ms), reicht das aus. Auch Temperaturdifferenzen, wie sie beispielsweise an Heizkörpern auftreten, lassen sich nutzen. Die Daten werden mit ständig wechselndem Code verschlüsselt übertragen, sodass die Verbindung sicher ist.

Selbstbauprojekte sind nicht möglich, da EnOcean ausschließlich Fertigkomponenten anbietet.

Z-Wave

Z-Wave ist laut Wikipedia mit mehr als 2100 zertifizierten Produkten die weltweit größte Systemplattform fürs Smart Home. Es gibt kaum eine Gerätegruppe an Sensoren, Aktuatoren etc. (Bild 10), die es nicht mit ZWave gibt. Die verwendete Frequenz macht die Verbindung auch sehr zuverlässig und durch Mesh-Fähigkeit ist auch die Reichweite groß, weshalb dieser Funkstandard vermutlich gern von Firmen eingesetzt wird, die Smart Homes installieren. Allerdings ist die Übertragungsgeschwindigkeit nur für Messwert- und Befehlsübertragung sinnvoll.

Da Privatpersonen in den einschlägigen Märkten aber kaum ZWave-, sondern stattdessen meist Zigbee-Komponenten kaufen



können, setzt sich der Standard in diesem Bereich nicht so recht durch. So sind auch die Preise bei Zigbee meist günstiger.

LoRaWAN

LoRaWAN (Long Range Wide Area Network) beruht auf dem patentierten Übertragungs-

Bild 9: EnOcean stellt zahlreiche batterielose Schalter und Sensoren her, die aber einen USB-Controller für den Server verlangen.

verfahren LoRa, das auf eine Einrichtungsdatenübertragung (Uplink) bei hohen Reichweiten (<10 km) und geringem Energieverbrauch ausgelegt ist. Es handelt sich um eine Funktechnik zur Übermittlung kleiner Datenmengen über große Entfernungen – und groß meint hier wirklich groß, sogar weltweit!

Für alles gerüstet! Tests, Tipps und Tools



Das Sonderheft richtet sich vor allem an Privatnutzer, Freelancer und kleinere Unternehmen und enthält Kaufberatungen, Tests und Praxisanleitungen zu typischen Büroprogrammen, auch abseits von Microsoft Office.



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

🛃 shop.heise.de/ct-homeoffice24





Bild 10: Der dänische Hersteller Danfoss war eine der ersten Firmen, die ZWave-Komponenten wie Heizkörperthermostate anboten.

Das Ganze funktioniert so: Ein LoRaWAN ist ähnlich wie ein WLAN sternförmig aufgebaut. Der LoRaWAN-Client sendet also seine Daten an ein Gateway, das sie dann übers Internet an einen LoRaWAN-Server weiterleitet. Das Gateway kann dabei einige Kilometer vom Client entfernt sein. Der Rekord für eine Verbindung vom Client zum Gateway liegt übrigens bei über 800 km. Aber damit sollten Sie in unseren dicht bebauten und mit Funkwellen geschwängerten Bereichen nicht rechnen.

Die Gateways gibt es weltweit verteilt. Sie werden meist von Privatpersonen, Vereinen, Universitäten u. ä. Einrichtungen betrieben. Vom Server wiederum kann man übers Inter-



Bild 11: Das LoRaWAN-fähige Board LoRa V3 Meshtastic besitzt außer einem Display auch einen ESP32.

Smart-Home-Funktechniken

net die Daten abrufen und sie dann im Smart Home verwenden.

Damit die Daten nur den befugten Empfänger erreichen, ist die Datenübertragung vom Client zum Gateway verschlüsselt, und vom Server erhalten nur die dort für den jeweiligen Client registrierten User die Daten.

Wozu kann man nun LoRaWAN im Smart Home einsetzen? Beispiele gibt es dafür genug: So kann man damit den in weiter Entfernung dauercampenden Wohnwagen überwachen oder sich das Legen einer Leitung zum Grundstücksende ersparen, nur weil man dort kontrollieren möchte, ob ein Vogelnistkasten Mieter gefunden hat oder nicht.

Für LoRaWAN gibt es eine ganze Reihe an Controllerboards, die neben der eigentlichen Funktechnik oft auch noch eine Akku-Ladeelektronik oder Displays enthalten (Bild 11). Meist beruhen die auf einem ESP-Chip, denn der hat den Vorteil, dass man ihn recht einfach in den Tiefschlaf (Deepsleep) versetzen kann, in dem er so gut wie keinen Strom verbraucht. Nur zur eigentlichen Messwerterfassung wacht er dann auf und sendet die Daten. Danach legt er sich wieder zur Ruhe. So hält ein Akku oder eine Batterie lange durch.

RF433

Wie die Bezeichnung RF433 schon vermuten lässt, arbeitet diese Funktechnik mit einer Frequenz von 433 MHz. Wahrscheinlich ist sie Ihnen auch schon einmal begegnet, denn das ist die Technik, die die in Bau- und Discountermärkten zuhauf angebotenen Funksteckdosen verwenden.

Diese gibt es meist im Set mit drei Steckdosen und einem Fernsteuersender. Was aber kaum jemand weiß: Die Steckdosen lassen sich

Funktechnik	WLAN	Bluetooth/BLE	Zigbee	Thread/Matter
Frequenzbereich (in Europa)	2400 MHz – 2483,5 MHz 5150 MHz – 5350 MHz 5470 MHz – 5725 MHz	2402 MHz – 2480 MHz	2405 MHz – 2480 MHz	868 MHz – 868,6 MHz 902 MHz – 928 MHz (nur USA) 2400 MHz – 2483,5 MHz
uni-/bidirektional	bidirektional	bidirektional	bidirektional	bidirektional
Mesh-fähig	nein	nein	ja	ja
maximale Sendeleistung	100 mW (2,4 GHz) bis 200 mW (5 GHz)	100 mW/ 10 mW (BLE)	6 mW	1 mW
Reichweite	30 m	100 m (Class 1)/10 m (BLE)	> 100 m, da Mesh-Bildung	> 100 m, da Mesh-Bildung
Übertragungsgeschwindig- keit geeignet für	alle Anwendungen	> 1MBit/s	250 KBit/s	20 KBit/s - 258 KBit/s
geeignet für Selbstbaupro- jekte	ja	ja	ja	ja
Stromaufnahme typischer Controllerboards	hoch (700 mA)	gering (Bluetooth)/ sehr gering (BLE)	sehr gering	sehr gering
geeignet für Akkubetrieb	+	++	++	++
geeignet für Batteriebetrieb	-	++	++	++
Datenverschlüsselung	ja	ja	ja	ja

auch von einem Smart-Home-Server wie Home Assistant steuern. Dazu braucht der entweder ein Set aus Sender- und Empfänger-Platinen (Bild 12) oder ein kleines Gateway wie in Bild 13. Da die Teile allesamt sehr preiswert sind, ist dies die billigste Möglichkeit einer Smart-Home-Steuerung.

Doch die Sache hat einen Haken: So ist die Funktechnik völlig offen, da keinerlei Verschlüsselung benutzt wird. Das heißt, jeder, der das Funksignal kennt (kann man leicht mitschneiden), kann die Geräte ebenfalls steuern.

Außerdem gibt es keinen Rückkanal. Man kann daher nicht kontrollieren, ob etwa eine Schaltsteckdose nun gerade ein- oder ausgeschaltet ist. Daher sollte man diese billige Technik nur dann einsetzen, wenn der Schaltzustand unerheblich ist, beispielsweise für eine Dekobeleuchtung per LED-Streifen, aber nicht für einen elektrischen Heizofen o. Ä.

Zusammengefasst

Damit bin ich durch mit dem Streifzug durch die Smart-Home-Funktechnik. In der Tabelle am Schluss habe ich das Wichtigste noch einmal zusammengefasst. Ich hoffe, dass Ihnen die Infos helfen können, sich Ihr vollelektrisches Haus optimal aufzubauen und es nicht im Chaos endet wie im gleichnamigen Stummfilm mit Buster Keaton, der offenbar die technische Entwicklung schon vor 100 Jahren erahnte. —hgb

Bild 13: Für den Server gibt es RF433-Interfaces, die über USB angeschlossen werden und sowohl einen Sender als auch einen Empfänger enthalten.



Bild 12: Links ein 433-MHz-Sender, daneben ein einfacher Empfänger (unten) und ein deutlich empfangsstärkerer Superhet-Receiver.



DECT ULE	LoRaWAN	Z-Wave	EnOcean	RF433
1880 MHz – 1900 MHz	433,05 MHZ – 434,79 MHz 863 MHz bis 870 MHz	850 MHz – 950 MHz	868 MHz	433,05 MHz – 434,79 MHz
bidirektional	unidirektional	bidirektional	unidirektional	unidirektional
nein	nein	nein	nein	nein
250 mW	25 mW	25 mW (meist aber nur we- nige mW)	keine Angabe	10 mW
40 m	2 bis 40 km	40 m	10 m	30 m
1152 KBit/s	292 bit/s bis 50 KBit/s	bis 100 KBit/s	125 KBit/s (max für 1 ms)	nicht festgelegt
nein	ja	ja	nein	ja
gering	gering, nur während des Sendens	bei Netzstrom-betriebenen Geräten hoch, bei akku- und batteriebetriebenen gering, da nur sporadisch aktiv	keine Stromaufnahme	gering (10 mA)
++	++	+	nicht notwendig	++
++	++	+	nicht notwendig	+
ja	ja	ja	ja	nein



Pi Pico 2

Schnellere ARM-Kerne, mehr Flash und RAM: Das neue Board ist flinker als sein Vorgänger. Viel interessanter sind jedoch die zwei zusätzlichen RISC-V-Kerne. Wir haben uns das mal näher angeschaut und durchgemessen.

von Daniel Bachfeld

Die Raspberry Pi Foundation hat den bereits zum Börsengang im Juni angekündigten Pi Pico 2 vorgestellt – und passenderweise auch schon auf der Maker Faire in Hannover gezeigt. Auf dem im Vergleich zum Vorgänger gleich großen Board ist der neue SoC RP2350 verbaut, der erheblich leistungsfähiger als der RP2040 des Pi Pico 1 ist. Der Pico 2 kostet 5 US-Dollar, bei deutschen Versendern ab 5,50 Euro.

Im RP2350 befinden sich zwei ARM-Kerne Cortex-M33 (ARMv8), die an sich schon leistungsfähiger als die Cortex-M0+ des Vorgängers sind. Zudem werden sie standardmäßig mit 150 MHz statt mit 133 MHz getaktet. Zusätzlich enthalten sie Erweiterungen für Gleitkommaberechnungen mit einfacher Genauigkeit (SP-FPU) sowie zur digitalen Signalverarbeitung.

Doch neben der CPU hat sich noch mehr getan auf dem Pico 2. Sein RP2350A bietet satte 520 KByte statt 264 KByte SRAM, die auch schneller angebunden sind. Insgesamt laufen Programme auf dem Pico 2 im Schnitt doppelt so schnell wie auf dem Pico 1.

GPIO

Statt acht programmierbarer PIO-Einheiten sind es nun zwölf. Der RP2350 hat wie sein Vorgänger 16 PWM-Einheiten und vier der 26 GPIO-Pins des Pico 2 arbeiten als ADC. Daneben gibt es die B-Variante des SoC: Der RP2350B im QFN80-Gehäuse hat 48 GPIO-Pins, von denen sich acht für ADC konfigurieren lassen und 24 als PWM-Einheiten.

Der Pico 2 bringt auf der Platine einen 4 MByte großen QSPI-Flashspeicher mit. Die Varianten RP2354A und RP2354B enthalten je 2 MByte Flash-Speicher gleich im Gehäuse. Hersteller wie Adafruit, Pimoroni und Sparkfun bieten bereits zahlreiche eigene Boards mit den verschiedenen SoCs an, unter anderem auch mit externem WLAN. Das bringt der Pico 2 bislang noch nicht mit, bis zum Jahresende will man aber auch das nachschieben.

Gänzlich neu ist ein "Highspeed Serial Transmit"-Port (HSTX). Er fungiert als serieller Datenausgang, der mit Double Data Rate (DDR) und bis zu 150 MHz Takt arbeitet. Auf diese Weise sollen sich bis zu 300 Megabit/s pro Pin ausgeben lassen, wobei man bis zu acht GPIO-Pins parallel schalten kann. Erste Software-Beispiele demonstrieren bereits, wie sich darüber DVI und HDMI realisieren lassen. Zur Eingabe eignet sich der neue Port jedoch nicht.

Da auch Firmenkunden immer wichtiger für Raspberry Pi werden und es dort höhere Sicherheitsanforderungen an die Hardware und Software gibt, hat man spezielle Funktionen dafür eingebaut: ARM-TrustZone-Erweiterungen, ein geschütztes Secure-Boot-ROM, einen Zufallszahlengenerator (RNG), einmalig programmierbaren Speicher (OTP), etwa zum Ablegen von Fingerprints, sowie ein SHA256-Rechenwerk.

RISC-V

Der eigentliche Clou im RP2350: Er enthält zwei weitere RISC-V-Kerne namens Hazard3, die man alternativ zu den ARM-Kernen nutzen kann. Man kann sogar je einen ARM- und einen RISC-V-Kern laufen lassen. Zwei ARM- und zwei RISC-V-Kerne parallel laufen zu lassen, funktioniert jedoch nicht. Laut Eben Upton kann man nur zwei Kerne über den internen Multiplexer mit dem AHB-Bus verbinden. Wie man das genau konfiguriert, haben wir in der Redaktion aber noch nicht rausgefunden.

RISC-V steht als freie Architektur in direkter Konkurrenz zu ARM. Jeder kann sich auf Grundlage der offenen Spezifikationen einen eigenen Prozessor konstruieren und beispielsweise mit FPGAs realisieren oder in Silizium ätzen, ohne Lizenzkosten zahlen zu müssen.

Wer will, kann sich den offengelegten RISC-V-Kern des Hazard3 auf GitHub anschauen. Er implementiert einen 32-Bit-Kern RV32I-MACZb*, also mit den Erweiterungen für Integer, Multiplikation, Atomare Instruktionen und Bitmanipulationen Zba, Zbb, Zbc und Zbs.

Der Entwickler Luke Wren ist zwar ASIC-Ingenieur bei Raspberry Pi, hat aber den Prozessor alleine in seiner Freizeit entwickelt. Der Kern steht unter einer Apache-Lizenz und kann auch für Weiterentwicklungen oder eigene Implementierungen genutzt werden.

Bislang hinkten RISC-V-Implementierungen der ARM-Konkurrenz meist hinterher, sowohl bei CPUs für Single Board Computer als auch bei Mikrocontrollern. Unsere Benchmarks zeigen aber, dass der Hazard3 zumin-



Das Layout des Hazard3 als ASIC von 2022. Für eigene Experimente ist die Konfiguration auf Github frei verfügbar.





*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 24,00 € statt 31,75 €*

** im Vergleich zum Standard-Abo

Jetzt bestellen:





Test

spberry Pi Pico			Basic Setting	s Debugge
Basic Settings				
Name				
blink				
Board type				
Pico 2				
Location				
/home/dab				Change
Select Pico-SDK version		Use RISC-V		
v2.0.0				
Debugger				
DebugProbe (CMSIS-DAP) [Default]	SWD (Pi host)			
		Show Advanced Opti	ons Cancel	Create

Für das Pico-SDK gibt es ein Plugin für VSCode, in dem sich Einstellungen übersichtlicher konfigurieren lassen. Für RISC-V gibt es einen eigene Checkbox.

dest einem Cortex-M33 ebenbürtig ist, solange keine Gleitkommaberechnungen ins Spiel kommen.

Wir haben mal gemessen: Der offizielle CoreMark von EEMBC lässt sich zwar für die Picos übersetzen und auch starten, Ergebnisse zeigte er jedoch nicht an. Wir nutzten deshalb eine Portierung (nickfox-taterli, siehe Link). Beim CoreMark erreichten der RISC-V 285,4 und der ARM 284,1. Weit abgeschlagen kommt der Pico 1 nur auf 159. Ähnlich sieht es beim Berechnen von Fibonacci-Zahlen (Integer) und Dhrystone aus. RISC-V überholt mit 321879 Dhrystone/s den ARM mit 301192. Der Pico 1 erreicht gerade einmal 189105.

Gleitkomma

Das Blatt wendet sich, wenn Gleitkommaberechnungen ins Spiel kommen. Im Whetstone-Benchmark schlägt der ARM mit 2094290 µs (1000 Loops, 1 Iteration) den RISC-V,der mit 52050377 µs 25mal länger rechnet. Selbst der Pico 1 ist mit 10504362 µs fünfmal schneller. In einem reinen Gleitkom-

Benchmarks

ma-Benchmark übertrumpft ARM den RISC-V sogar um den Faktor 64 (siehe Tabelle).

Wer selber messen will, muss beachten, dass die neue Pico-SDK 2.0 bei Aufruf der in manchen Benchmarks verwendeten Zeitfunktion clock() für ARM Millisekunden zurückliefert, unter RISC-V indes Mikrosekunden. Wir haben dort der Einheitlichkeit halber zur Zeitmessung die Funktion time_us_64() genutzt.

Dass sogar der Cortex-M0+ im Pico 1 den RISC-V überflügelt, liegt vermutlich an den seit nunmehr einigen Jahren per Hand optimierten Software-Bibliotheken für Gleitkommaoperationen auf ARM-CPUs. Hier dürfte es bei RISC-V noch Nachholbedarf geben.

Programmieren

Die aktualisierte Pico-SDK 2.0 unterstützt bereits den RP2350. Dabei wählt man über das Tool cmake das Ziel aus, beispielsweise -DPICO_PLATFORM=rp2350-riscv für RISC-V, wenn man im Terminal arbeitet. Für diesen Artikel haben wir lieber mit VSCode programmiert und das "Raspberry Pi Pico Visual Studio

	Pico 2 ARM	Pico 2 RISC-V	Pico 1
CoreMark	284,1	285,4	159
Dhrystone	301192	321879	189105
nqueens	19661702 µs	22803151 µs	40081730 µs
Fibonacci (24)	20007 µs	19008 µs	27009 μs
Whetstone (1000 Loops)	2094290 µs	52050377 µs	10504362 µs
Float-ARM	5430 µs	348636 µs	46598 µs
CircuitPython (float, sin)	9,4 s	n/a	20,6 s
CircuitPython (int, mul)	17,7 s	n/a	31,9 s
circuiti ython (int, mui)	1,13	n/u	51,23

Code"-Plugin verwendet. Damit lässt sich schnell zwischen verschiedenen Boards umschalten. Zwischen ARM oder RISC-V als Ziel muss man trotzdem noch manuell im CMakefile.txt angeben. Immerhin kann man beim Anlegen neuer Projekte komfortabel verschiedene Optionen anklicken. Für die Zielplattform werden automatisch die passenden Cross-Compiler heruntergeladen.

Der Arduino-Core für Pico wurde ebenfalls aktualisiert, sodass man mit einem Update des Board-Support-Package "Raspberry Pi Pico/ RP2040" auf Version 4.0.1 Sketche für den Pico 2 übersetzen kann. Leider nur für ARM, eine Option zur Auswahl der RISC-V-Kerne haben wir nicht gefunden.

Zum Redaktionsschluss gab es ein Alpha-Image für CircuitPython und ein Preview-Image für MicroPython, jeweils für den Betrieb auf ARM und RISC-V. Sogar eine Portierung für Linux für die RISC-V-Kerne liegt schon vor. Das läuft allerdings im Moment nur auf einem Board von Sparkfun mit 16 MByte Flash und 8 MByte PSRAM.

GPIO-Fehler

Kurz nach den ersten Verkäufen des Pico 2 wurde ein Fehler in den GPIOs bekannt: Ist ein Pin als Eingang mit einem internen Pull-Down-Widerstand konfiguriert, bleibt der Eingangspegel auch nach Wegfall einer externen Eingangsspannung bei 2,2 V hängen, statt wieder auf 0 V zu sinken. Erst wenn man den Eingang (niederohmig) kurz auf 0 V legt, bleibt der Eingang anschließend auf 0 V.

Fazit

Mehr Leistung ist toll, aber den meisten Makern wird der Sprung nach vorne nicht unbedingt viel bringen, es sei denn, sie entwickeln hauptsächlich zeitkritische oder rechenintensive Anwendungen. Viele haben sich eher eine USB-C-Schnittstelle gewünscht. Der RP2350 wird aber vermutlich in der Industrie auf sehr fruchtbaren Boden fallen.

Besonders spannend ist die Möglichkeit, mit RISC-V und dessen relativ überschaubarem Befehlssatz experimentieren zu können. Espressif rüstet schon seit Längerem seine neuen ESP-Boards mit RISC-V aus und für den Programmierer ist es von Vorteil, sich auf eine CPU-Architektur konzentrieren zu können. Da bekommt der Autor dieses Tests sogar wieder Lust auf Assembler. —dab

Das Pico-2-Board hat uns Berrybase für den Test zur Verfügung gestellt.



© Weise academy WORKSHOPS 2024



9. - 10. Oktober

IEC 62443: Industrial Security für Produkthersteller

Erfahren Sie, wie Sie Ihre Produktentwicklung während des gesamten Lebenszyklus sicher gestalten.



14. – 15. Oktober Terraform: Infrastructure as Code

Anhand praktischer Beispiele und Szenarien wird das automatisierte Erstellen von IT-Infrastrukturen, das Modifizieren und Abbauen behandelt.



10. – 11. Oktober Microsoft 365 Copilot – Analyse und Evaluierung

Die Schulung vermittelt Methodenkompetenz und zeigt klassische Stolpersteine von Penetrationstests auf.



05. November Einführung in die Softwarearchitektur

Dieser Workshop führt praxisnah in die Inhalte und Aufgaben der Softwarearchitektur ein und gibt einen Überblick über die Rolle des Softwarearchitekten.



05. - 06. November

Penetrationstests: Methodik verstehen, richtig ausschreiben und Ergebnisse auswerten

Die Schulung vermittelt Methodenkompetenz und zeigt klassische Stolpersteine von Penetrationstests auf.



07. November

M365: Baselines für Tenant Settings, Informationssicherheit & Datenschutz

Sie erfahren, wie Sie den Einsatz von M365 sicher gestalten und lernen notwendige Maßnahmen von Datenschutz kennen.

Weitere Infos unter heise-academy.de/marken/ix

Ausprobiert — von Make:—

Fischertechnik Bausätze für erfahrene Maker

Dass Fischertechnik einen festen Platz in der Maker-Community hat, lässt sich anhand zahlreicher Projekte im Internet und auf Veranstaltungen wie der Maker Faire erkennen. Kits, die auf Maker abzielen, gab es vom Hersteller aber streng genommen noch nicht – bis jetzt. Mit den Maker Kits Car, Omniwheel und Bionic wagt sich Fischertechnik offiziell in die Maker-Szene und will damit erfahrene Maker, Maker-Spaces und Fablabs ansprechen; also Menschen, die sich bereits mit Mikrocontrollern, Elektronik und optional mit 3D-Druck auskennen.

Man kann sich zwischen einem Fahrzeug mit Differenzialgetriebe und Servolenkung, einem Gefährt mit Mecanum-Rädern und einem vierbeinigen Laufroboter entscheiden. Die Fahrzeuge verwenden Encoder-Motoren, mit denen sich die Zahl der Umdrehungen und darüber die Position genau bestimmen lässt, und der Laufroboter nutzt acht Digital-Servos (DS3012MG) mit einem Drehmoment von jeweils 3,5 kg/cm (bei 6 V). Ansonsten gibt es jede Menge Bausteine in gewohnter Fischertechnik-Qualität, ein Datenblatt, einen Link zur Online-Bauanleitung sowie ein GitHub-Repository mit vereinzelten Vorschlägen für die Programmierung. Abgesehen davon müssen Maker selbst entscheiden, welche Mikrocontroller und Programmiersprache sie verwenden.

Dadurch ist man, bis auf das hauseigene Steckersystem, nicht auf die proprietären Lösungen des Herstellers angewiesen. Damit man seine Elektronik trotzdem auf den Rücken der Modelle montieren kann, bietet Fischertechnik für den Arduino UNO (Rev 3, SMD-Version), MEGA, Raspberry Pi oder Pi Zero zusätzlich 3D-druckbare Gehäuse an, in deren Unterseite sich Verbindungssteine stecken lassen. Die Gehäuse





sind stellenweise etwas filigran konstruiert, aber zum Glück kann man sie zusätzlich zum STL-Format auch als STEP-Dateien herunterladen. Diese lassen sich in einem gängigen CAD-Programm an die eigenen Bedürfnisse anpassen. So kann man sich etwa die Verbindungselemente aus dem 3D-Modell herauskopieren und in ein eigenes Gehäuse integrieren, in das dann z. B. ein Raspberry Pi Pico, Arduino Nano oder ESP32-Board passt. Alternativ muss man sich mit weiteren Fischertechnik-Bausteinen, Kabelbindern o. Ä. behelfen.

Ergänzend zum Mikrocontroller benötigt man je nach Kit noch einen Servo-Controller (z. B. PCA9685), Motor-Treiber und was man sonst noch anbringen möchte. Mit der passenden Bibliothek (z. B. von Adafruit) lassen sich die Encoder-Motoren der Fahrzeuge schnell in Betrieb nehmen und die Servos des Laufroboters kalibrieren. Danach kann man sich ausgiebig damit beschäftigen, wie man vier Beinen das Laufen beibringt oder mit dem Omniwheel am besten von einer Parklücke zur nächsten gleitet.

Der Preis der Sets richtet sich nach dem Komplexitätsgrad des Modells und den mitgelieferten Motoren. Mit ca. 90 Euro ist das Car-Set am günstigsten, das Omniwheel-Fahrzeug liegt bei etwa 175 Euro und der Bionic-Laufroboter bei ungefähr 277 Euro (inklusive der Servos im Wert von knapp 150 Euro).

Die Maker Kits machen durchweg einen robusten Eindruck und bieten eine solide Basis für Fortgeschrittene, um in die Programmierung eines Roboters oder Fahrzeugs einzusteigen. Durch das Stecksystem eignen sich die Bausätze auch für Maker ohne CAD-Kenntnisse und 3D-Drucker. Und entscheidet man sich irgendwann doch, das Set für etwas völlig anderes zu verwenden, kann man alle Teile mit dem restlichen Fischertechnik-Sortiment kombinieren. Das ist in jedem Fall nachhaltiger, als ein Set zu kaufen, das nur für einen Zweck gefertigt wurde. —*akf*

Die Baukästen wurden uns von Fischertechnik zum Test zur Verfügung gestellt.



 Hersteller
 fischertechnik GmbH & Co. KG

 URL
 fischertechnik.de

 Preis
 90 bis 277 Euro (siehe Text)

© Copyright by Maker Media GmbH.

Ausprobiert von Make:

Color Kit Grande Enclosure

Gehäuse für ESP32-Feather und TFT-Display

In der Make 5/23 haben wir das ThingPulse ESP32 WiFi Color Display Kit Grande vorgestellt. Der kompakte Formfaktor dieses Touchscreens mit ESP-32-Board hat geradezu nach einem Gehäuse gerufen. Dies liefert die kleine Hardwareschmiede nun nach. Entwickelt wird in der Schweiz, produziert und versandt in bzw. aus China, daher kommen zum Preis noch Steuern, Versand und ggf. Zoll hinzu.

Dieses Produkt zeigt, wozu 3D-Druck heutzutage fähig ist und wie man auch für kleine Stückzahlen von Geräten Zusatzteile und Gehäuse herstellen lassen kann, die kaum von Spritzgussteilen zu unterscheiden sind. Das Gehäuse kommt in zwei Teilen, die mit vier kleinen Schrauben (selbst schneidend, Philips-Kopf) zusammengehalten werden. Die Passform und die Spaltmaße sind einwandfrei. Mit guten Augen erkennt man noch leichte Schichtenbildung vom SLA-Druck (Stereolithografie). Die anthrazitgraue, samtige Oberfläche fasst sich anders als ABS oder Ähnliches an, ist aber angenehm zu greifen.

Im Gehäuse ist noch einiges an Luft, um einen Akku oder zusätzliche Elektronik unterzubringen. Einen passenden 3,7-V-Akku mit 900 mAh (6 US-Dollar) bietet ThingPulse auch an, aber man kann jeden Akku, der einen JST-Stecker hat und ins Gehäuse passt, einsetzen. Das im "Color Kit Grande" verbaute ePulse-Feather-Board hat eine eingebaute Ladeelektronik.

ThingPulse stellt die 3D-CAD-Daten auf GitHub zur Verfügung. Die STEP-Dateien lassen sich in praktisch iedem CAD-System öffnen und bearbeiten. Damit können dann Varianten mit weiteren Durchbrüchen oder einem tieferen Gehäuse für eigene Projekte konstruiert und gedruckt werden. Wer keinen Resin- oder SLA-Drucker hat, kann die Teile auch auf FDM-Druckern herstellen; die Wandstärken sind ausreichend. Moderne Slicer wie der Prusa-Slicer importieren auch direkt die STEP-Dateien.

ThingPulse war inzwischen auch nicht untätig und so gibt es jetzt mehr Demos für das Color Kit Grande inklusive einer ausführlichen



Demo mit der LVGL-Bibliothek. Damit lassen sich unter anderem aufwendige Grafiken und eine Touchscreen-Bedienung realisieren.

Verpassen Sie Ihrem Projekt ein Gehäuse, das sich auch im Wohnzimmer gut macht, als Wetteranzeige, Solaranlagen-Monitor oder Ähnliches. -caw

Hersteller	ThingPulse Ltd.
URL	thingpulse.com
Preis	15 US-\$ (zzgl. Versand, MwSt. und Zoll)

Geheimakte Computer

Everything they don't teach you about tech

GEHEIM

COMPUT

Hast du die letzten 100 Jahre unter einem Stein verbracht oder kommst du gar von einem anderen Planeten und möchtest jetzt wissen, was dieser Computer ist, von dem alle auf der Erde ununterbrochen reden? Dann ist dieses Buch genau das Richtige für dich! Mit Geheimakte Computer hat der Autor Tobias Hübner (Lehrer, Autor, Dozent, IT-Trainer) auf gut 300 Seiten kurz und knapp alles zusammengefasst, was auch nur annähernd mit Computern zu tun hat.

Hier ist wirklich alles drin: die bunte Entstehungsgeschichte des Computerzeitalters,

Tipps und Tricks für die eigene Hard- und Softwareentwicklung, jede Menge Retro-Game-Nostalgie, eine Erklärung, wie ein Quantencomputer funktioniert, Maker-Challenges und (natürlich) eine Diskussion über die Gefährlichkeit von Kl. Die Sprache ist informell, direkt und sehr enthusiastisch. Manchmal fast zu viel. Zahlreiche Farbbilder und Screenshots laden zum spontanen Blättern ein.

Zielgruppe sind wohl vor allem Schüler und ihre Erziehungsberechtigten, die womöglich noch mehr Aufklärung brauchen als ihre Zöglinge. Aber auch Eltern und Menschen, die sich schnell einen breiten Überblick darüber verschaffen wollen, wie Computer unsere Welt beeinflussen, wie sie funktionieren und wie sie überhaupt entstanden sind, könnten sich für das Buch interessieren. In einer Reihe von Porträts lernen wir die Koryphäen der digitalen Geschichte kennen. Menschen wie

den Computerpionier Konrad Zuse, Apple-Mitgründer Steve Wozniak oder KI-Vordenker Joseph Weizenbaum.

Das Buch beschreibt, warum der Computer einerseits "das kreativste Werkzeug der Welt" ist, andererseits aber durch KI zunehmend zur "Bedrohung" wird. Der Leser wird über Risiken und die großen Fragen der Zukunft aufgeklärt, aber auch ermutigt und inspiriert, eigene Projekte mit Computern zu realisieren. Nachdem auf zwei Seiten erklärt wird, wie Transistoren und logische Operatoren funktionieren, folgt die Maker-Aufgabe, mit ein paar Schrauben und Büroklammern ein Logikgatter zu bauen.

Für eingefleischte Maker ist hier wenig zu holen. Dafür bietet das Buch einen faszinierenden und kompakten Überblick über die digitale Welt und ist damit das perfekte Geschenk für alle, die nicht verstehen, was in einer Maker-Werkstatt passiert. —mch

Hinweis: Dieses Buch erscheint in einem Verlag, der wie die Maker Media GmbH zu heise medien gehört.

Geheimakte Computer
Tobias Hübner
Rheinwerk Verlag GmbH
318 Seiten
978-3-8362-9369-3
$\textbf{24,90} \in (\textbf{29,90} \in \textbf{Buch} + \textbf{E-Book})$

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, xxr 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)

Redaktion: Heinz Behling (hgb), Johannes Börnsen (jom), Ákos Fodor (akf), Marcus Hansson (mch), Daniel Schwabe (das), Dunia Selman (dus, Social Media), Carsten Wartmann (caw)

Mitarbeiter dieser Ausgabe: Beetlebum, Erik Golz, Tobias Hübner, Benno Lottenbach, Markus Mierse, Nikolai Radke, Uwe Rohne, Kai Schauer, Florian Schäffer, Tobias Vogel, Lee Wilkins

Assistenz: Susanne Cölle (suc), Martin Triadan (mat)

Layout und Satz: Steffi Martens, Lisa Reich, Nicole Wesche, Heise Medienwerk GmbH & Co. KG

Korrektorat: Dörte Bluhm, Lara Bögner, Marei Stade, Christiane Tümmler, Heise Medienwerk GmbH & Co. KG

Titel: Nicole Wesche

Fotografie und Titelbild: Andreas Wodrich

Digitale Produktion: Melanie Becker, Thomas Kaltschmidt, Pascal Wissner

Hergestellt und produziert mit Xpublisher: www.xpublisher.com

Nachgefragt

Wir bauen im Heft einen

nutzen können?

Schimpfwortgenerator.

Wann hättest Du ihn zuletzt

Verlag

Maker Media GmbH Postfach 61 04 07, 30604 Hannover Karl-Wiechert-Allee 10, 30625 Hannover Telefon: 05 11/53 52-0 Telefax: 05 11/53 52-129 Internet: www.make-magazin.de

Herausgeber: Christian Heise, Ansgar Heise

Geschäftsführung: Ansgar Heise, Beate Gerold

Anzeigenleitung: Daniel Rohlfing (-844) (verantwortlich für den Anzeigenteil), mediadaten.heise.de/produkte/print/ das-magazin-fuer-innovation

Leiter Vertrieb und Marketing: André Lux (-299)

Service Sonderdrucke: Julia Conrades (-156)

Druck: Dierichs Druck + Media GmbH & Co.KG, Frankfurter Str. 168, 34121 Kassel

Vertrieb Einzelverkauf: DMV DER MEDIENVERTRIEB GmbH & Co. KG Meßberg 1 20086 Hamburg Telefon: +49 (0)40 3019 1800 Telefax: +49 (0)40 3019 1815 E-Mail: info@dermedienvertrieb.de Internet: dermedienvertrieb.de

Einzelpreis: 13,50 €; Österreich 14,90 €; Schweiz 26.50 CHF; Benelux 15,90 €

Abonnement-Preise: Das Jahresabo (7 Ausgaben) kostet inkl. Versandkosten: Inland 80,50 €; Österreich 88,90 €; Schweiz 123.90 CHF; Europa 95,20 €; restl. Ausland 100,80 €

Das Make-Plus-Abonnement (inkl. Zugriff auf die App, Heise Magazine sowie das Make-Artikel-Archiv) kostet pro Jahr 6,30 \in Aufpreis.



Nikolai Radke Berlin, ist der Erfinder des Schimpfolinos.

Mein Favorit wäre: bei der Papstwahl auf die Frage "Quo nomine vis vocari?". Sachlicher wäre: als mir mal wieder kein neues Passwort eingefallen ist. Tobias Hübner Düsseldorf, interviewt auf S. 98 id-Software-Legende John

Romero.

Als ich bei meinem Apple lle im laufenden Betrieb eine Steckkarte entfernt und dadurch den Computer geschrottet habe, hätte ich ein paar originelle Schimpfwörter gut gebrauchen können. Markus Mierse München, baut auf S. 78 ein besonderes Display mit Bedienung.

Als ich ein Gerät aus Fernost in Betrieb nahm und las: "Der Lösung häufig gefallener Fragen: Prüfen Sie, ob mit der Aus-Taste gearbeitet wird oder nicht. Verbunden so mit dem rot-weißen Kabel im Bild ohne Transformator."



Abo-Service:

Bestellungen, Adressänderungen, Lieferprobleme usw.: Maker Media GmbH Leserservice Postfach 24 69 49014 Osnabrück E-Mail: leserservice@make-magazin.de Telefon: 0541/80009-125 Telefax: 0541/80009-122

Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden. Kein Teil dieser Publikation darf ohne ausdrückliche schriftliche Genehmigung des Verlags in irgendeiner Form reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Alle beschriebenen Projekte sind ausschließlich für den privaten, nicht kommerziellen Gebrauch. Maker Media GmbH behält sich alle Nutzungsrechte vor, sofern keine andere Lizenz für Software und Hardware explizit genannt ist.

Für unverlangt eingesandte Manuskripte kann keine Haftung übernommen werden. Mit Übergabe der Manuskripte und Bilder an die Redaktion erteilt der Verfasser dem Verlag das Exklusivrecht zur Veröffentlichung. Honorierte Arbeiten gehen in das Verfügungsrecht des Verlages über. Sämtliche Veröffentlichungen in Make erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes.

Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.

Published and distributed by Maker Media GmbH under license from Make Community LLC, United States of America. The 'Make' trademark is owned by Make Community LLC Content originally partly published in Make: Magazine and/or on www.makezine.com, ©Make Community LLC 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



Erik Golz

Mühlheim, berechnet und fräst auf S. 106 ein eigenes Portraitobjektiv.

Wenn mein Smart Home mal wieder smarter ist als ich!

Ein Veranstaltungsformat der Make:

C



Maker Faire

Das Format für Innovation & Macherkultur

Die nächsten Events





Unendlich flexibel TUXEDO InfinityFlex 14 - Gen1

