DEUTSCHLANDS GEFÄHRLICHSTES DIY-MAGAZIN

Make

ćŁ

Jake:

3 Servos, 6 Beine: BASIC-Roboter aus dem Drucker

WLAN-LED-Controller

- Fertige Firmware, fertige App
- Dutzende Effekte und Animationen vorkonfiguriert
- Ohne Programmieren
- Lichtorgel-Funktion



Projekte

ChatGPT auf der Schreibmaschine
H0-Modellauto fernsteuern mit ESP
Programmierbarer Rasensprenger

ESP32-Radio

- Mit Internet- und UKW-Empfang
- Drehgeber mit Push-Funktion zur Senderwahl



Workshops

Grafik auf Mini-Displays
Fluoreszenzdisplays ansteuern
KI-Bildgenerator entwirft Möbel

Laborkoffer

- Für mobile
 Elektronik Experimente
- Modular und anpassbar



2/23 6.4.2023 CH CHF 26.50 AT 14,90 Benelux 15,90

£ 13 5(



Deutschlands

7 BIE

 \bigcirc

Hannover Maker Faire[®]

Maker-Treffen

Melde dich mit deinem Projekt **bis zum 11. Juni kostenfrei** als Aussteller an.

19.–20. Aug.

Hannover Congress Centrum

maker-faire.de



Multimedial

Früher war der Begriff Multimedia bei mir leider eher negativ konnotiert, ich musste dann immer an den Multimediaraum meines Gymnasiums denken: VHS-Rekorder, Röhrenfernseher und Over-head-Projektor. Mittlerweile beschreibt dieser Begriff jedoch am besten die positive Weiterentwicklung des Make-Magazins. Vor knapp einem Jahr haben wir begonnen, ergänzende Videos zu vielen unserer Artikel zu machen. Eine Einführung etwa in eine Bedienoberfläche funktioniert in einem Video viel besser als eine geschriebene Klickanleitung (siehe FreeCAD-Artikel auf Seite 106). Wie der WLAN-LED-Controller (Seite 94) vom Titel in Aktion funktioniert und wie die umwerfenden, teils soundgesteuerten Animationen aussehen, kann man nur schwer in statischen Bildern vermitteln.

Eine Schritt-für-Schritt-Anleitung zum Aufbau eines Projekts, nebst Schaltplan, Materialliste, Listings und Konfigurationstabellen ist aber wiederum besser in einem Artikel im Magazin aufgehoben. Wer will schon zum Lesen eines Listings in einem Video vor- und zurückspulen? Und zum Schmökern, Durchblättern und schnell mal Nachschlagen eignet sich ein gedrucktes Heft sowieso viel besser. Aber die Kombination aus Print (bzw. online lesen) und Video ist aus unserer Sicht ideal. Sofern es zu einem Artikel ein zusätzliches Video gibt, ist dies in der Kurzinfo zu sehen. Alle Video-Links führen zu YouTube, wo die Videos allerdings nicht gelistet sind. Das heißt: Nur Sie als Leser sehen dieses Zusatzmaterial, da Sie den Link kennen. Darüber hinaus produziert seit Ende letzten Jahres unser Videoredakteur Johannes Börnsen zusätzliche Videos für YouTube, die nicht immer an Inhalte des Magazins gekoppelt, aber frei verfügbar sind. Wir hoffen damit, eine größere Zielgruppe für das Thema Making neugierig zu machen. Wir würden uns freuen, wenn Sie in unsere Videos reinschauen und diese teilen würden. Haben Sie Themenwünsche, Lob oder Kritik zur Umsetzung der Videos? Schreiben Sie uns.

make-magazin.de/x656

Daniel Bachfeld

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



WLAN-LED-Controller

Von der stimmungsvollen Ambient-Beleuchtung zur psychedelischen Lightshow: Mit einem ESP8266 oder 32, LED-Streifen von der Rolle und der kostenlosen WLED-Software gelingen atemberaubende Effekte, egal ob für den Fernsehabend oder im Lichtorgel-Modus für die Party. Der Aufbau gelingt ganz ohne Programmierkenntnisse und mit sehr überschaubarem Löt-Aufwand.

8 WLAN-LED-Controller

Inhalt

Projekte

Der Sommer kommt bestimmt, und um gezielt Ihr gepflegtes Grün und nicht den Gehweg zu bewässern, empfehlen wir unseren programmierbaren Vektor-Rasensprenger. Was sonst noch die Flora freut, könnten Sie ja unsere schlaue Schreibmaschine fragen, die auf eingetippte Probleme probate Lösungen liefert - per ChatGPT.

- 28 KI auf der Schreibmaschine
- 44 Programmierbarer Rasensprenger mit Joypad-Steuerung
- 52 H0-Modellautos mit ESP fernsteuern



- **3** Editorial
- 6 Leserforum, Comic
- 8 Projekt: Disco-Lichtshow mit WLED
- 16 Projekt: Internet-UKW-Radio mit ESP32 und Akku
- 24 Maker Faire: Ausblick auf Hannover und Wien
- **28** Projekt: KI auf der Schreibmaschine
- 34 Know-how: Nachttisch aus KI-Entwurf
- **38** Community-Projekt: Laser-Scanning-Mikroskop aus Blu-Ray-Player
- 40 Community-Projekt: WLED mittels Herzfrequenz steuern
- 42 Community-Projekt: Atomic-Chess
- 44 Projekt: DIY-Rasensprenger mit Spielecontroller steuern
- 52 Projekt: H0-Modellautos mit ESP fernsteuern
- 58 Projekt: BASIC-Roboter als Luftmessstation
- 64 Workshop: Adafruit GFX-Library Malen mit Zahlen

Workshops

Von Retro bis Avantgarde: In unseren Workshops lernen Sie, wie man kultige Fluoreszenzdisplays zum Leben erweckt, mit der GFX-Library ansehnliche Daten-Visualisierungen auf ein Mini-Display zaubert oder die Midjourney-KI dazu bringt, Möbel-Unikate passend zu den Möglichkeiten Ihrer Holzwerkstatt zu entwerfen.

- 34 KI-Bildgenerator entwirft Möbel
- **64** Grafik auf Mini-Displays
- 72 Fluoreszenz-Displays ansteuern

ESP32-Radio

Keine Angst, es gibt keine Spulen zu wickeln und Platinen zu bestücken: Für unseren leistungsfähigen Kleinempfänger für UKW und WLAN-Streaming reichen dank vorgefertigter Module auch durchschnittliche Lötkünste. Mit Lilon-Akku und Stereo-Empfangsmodul funktioniert er auch abseits von Internet und Steckdose.

16 Internet+UKW-Radio mit ESP32 und Akku



- 72 Know-how: Fluoreszenzdisplays
- 80 Reingeschaut: Mini Game Machine
- 82 Workshop: Experimentierkoffer Make-Breadboard++
- 90 Tipps & Tricks: Mehr Messwerte bei eq3-Thermostaten, Pi4-Bluetooth-Probleme, OpenAl in Home Assistant
- 94 Werkstatt: 3D-Drucker tunen, Teil 2
- 102 Test: RISC-V-Einplatinencomputer StarFive VisionFive 2
- 106 Workshop: Technische Zeichnungen mit FreeCAD, Teil 3
- 110 Kurzvorstellungen: Mikrocontroller-Boards, 3D-Drucker, Raspi-Kamera, Debugging-Adapter für Picos, innovative Akkus, G-Gode-Add-on für Blender
- 114 Impressum, Nachgefragt



Laborkoffer

Gestatten: MoaB – Mother of all Breadboards, das Unterwegs-Labor für verregnete Urlaubstage, Maker-Treffen oder lange Reisen. Der Experimentierkoffer vereint Netzteil, Verdrahtungsfeld und diverse Bedienelemente zu einem luxuriösen Kompaktlabor, das im netzunabhängigen Akkubetrieb den Erfindergeist überallhin begleitet.



82 Laborkoffer Make-Breadboard++

Leserforum

BASIC seit Jahren resident

Raspberry Pi Pico: Back to BASIC, Make 1/23, S. 72

Zum Artikel wollte ich noch bemerken, dass von Olimex seit Jahren ein Board mit residentem MMBASIC vertrieben wird (Duinomite). Dieses Board habe ich 2013 entdeckt und für eine Steuerung/Regelung meiner Fußbodenheizung eingesetzt. Die Ports auf diesem Board sind kompatibel zum Arduino und entsprechend programmierbar. Auch einige A/D und D/A lassen sich konfigurieren. Letztlich habe ich mich für den 1-Wire-Bus entschieden, da die DS18-Sensoren die zuverlässigsten Messwerte bei langen Leitungswegen ergeben. Mit einer USB-Zehnertastatur und einer 7"-VGA-Anzeige steht eine komfortable Heizungsregelung zur Verfügung – übrigens seit 2014 störungsfrei und 24/7 im Einsatz.

Jürgen Biel

Eindimensional fokussiert

Heizungslüfter, Make 1/23, S. 24

Sicher sind wir alle auf der Suche nach Möglichkeiten, unsere Heizkosten zu senken. Allerdings hat das Problem ein paar mehr Facetten, als nur den Vorlauf zu senken.

Richtig ist, dass die Flachheizkörper (wie einer im Bild im Artikel zu sehen ist), den Raum nur dann warm bekommen, wenn sie richtig dimensioniert sind und der Vorlauf stimmt. Nimmt man die Vorlauftemperatur herunter, dann treten mehrere Effekte ein, die die Wärmeabgabe reduzieren. Diese Effekte versuchen Sie durch den Lüfter zu kompensieren. Sie pusten kühle Luft vom Boden in den Heizkörper, wodurch der Wärmeübertrag an die Luft verbessert wird. Gleichzeitig verteilt sich die Luft im Raum schneller, weil diese explizit angetrieben wird. Soweit hört sich das plausibel an. Den womans-acceptance-factor lassen wir mal außer Acht.

Meiner Auffassung nach holen Sie sich aber mehr Probleme durch die Absenkung des Vorlaufs ins Haus, als nur die verminderte Abgabe der Wärme und die Verteilung der Warmluft: Zum einen müssen Sie eine bestimmte Leistung (gemessen in Watt) bereitstellen, die der Raum durch die Temperaturdifferenz nach außen wieder abgibt. Damit halten Sie in einem Fließgleichgewicht die Raumtemperatur. Diese Leistung bekommen Sie bei gegebener Vorlauftemperatur aber nur, wenn Sie durch das Luftpusten den Rücklauf weiter runterbringen (die Spreizung muss die gleiche sein), oder Sie müssen den Mengenfluss im Heizkörper hochbringen, was Ihr hydraulischer Abgleich nicht zwingend hergibt.

Allerdings möchten Sie eigentlich gar nicht die Temperatur im Raum einstellen, sondern Sie möchten sich wohlfühlen. Der Energieberater lernt dazu in seiner Ausbildung, dass man die Raumtemperatur niedriger halten kann, wenn die wärmeabgebende Fläche groß ist.

instagram.com/

MakeMagazinDE

pinterest.com/

youtube.com/

MakeMagazinDE

MakeMagazinDE

Dadurch wird die Fußbodenheizung besonders wirksam. Sie fühlen sich bereits wohl, wenn die Temperatur noch gar nicht so hoch ist.

Sie haben nun einen kleinflächigen Heizkörper, der mit Ihrer Maßnahme noch kälter wird. Dadurch müssen Sie die Raumtemperatur höher halten, um sich wohl zu fühlen. Da die Luft insgesamt nicht so warm aus dem Heizkörper ausströmen kann, muss sie auch schneller bewegt werden, um überall wärmend anzukommen. Dabei werden Sie Einbußen im Wohlfühlklima haben, da der erhöhte Luftstrom natürlich auch Ihre Hautoberfläche besser kühlt. Sie können dann also nicht mehr in einem warmen "stehenden" Luftpolster am Arbeitsplatz sitzen. Je nach persönlichem Empfinden müssen Sie dann eine höhere Raumtemperatur haben, um sich wieder wohl zu fühlen.

Wenn Sie z. B. bei einer Wärmepumpe den Vorlauf von 55 auf 50°C reduzieren, steigern Sie den SCOP (*Seasonal Coefficient of Performance*) um ca. 6,5%. Wenn Sie jetzt aber den Raum z. B. 1°C höher heizen müssen, dann sind die 6% wieder weg.

Fazit: Es wurde zu eindimensional auf das Problem "Vorlauf" fokussiert und eine Enttäuschung ist vorprogrammiert. Ich würde mir wünschen, dass Sie immer den Rat eines zweiten Experten hinzuziehen, damit Sie die Euphorie nicht in eine völlig falsche Richtung laufen lässt.

Sensorenposition

entscheidend

Dr. Frank Lechtenberg

Kontakt zur Redaktion

Leserbriefe bitte an:

Korrekturen

heise.de/make/kontakt/

Wir behalten uns vor, Zuschriften unter Umständen ohne weitere Nachfrage zu veröffentlichen; wenn Sie das nicht möchten, weisen Sie uns bitte in Ihrer Mail darauf hin. Sie haben auch die Möglichkeit, in unseren Foren online über Themen und Artikel zu diskutieren:

O

- www.make-magazin/forum
- www.facebook.com/ MakeMagazinDE
- Www.twitter.com/ MakeMagazinDE

Manchmal unterläuft uns ein Fehler, der dringend korrigiert gehört. Solche Informatio-

Sie alle Ergänzungen und Berichtigungen zu einzelnen Heft-Artikeln auch zusätzlich

über den Link in der Kurzinfo am Anfang des jeweiligen Artikels.

nen drucken wir weiterhin auf den Leserbriefseiten im Heft, aber seit Ausgabe 1/17 finden

n online Heizungslogger, Make 1/23, S. 18

Wegen einer Fehlfunktion meiner Heizung bin ich diesen Winter ein ähnliches Projekt angegangen, allerdings mit zehn Temperatursensoren und auf Basis eines ESP8266, *ESPHome* und *Home Assistant*. Gerade am Speicher ist mir dabei aufgefallen, dass die Positionierung der Sensoren entscheidend ist. Mein Heizkessel zeigte im Display völlig andere Temperatur-Werte als meine Sensoren an und letztere wurden wegen schlechter Positionierung stark von der internen Rohrführung von Vor- und Rücklauf des Kessels und des Heizkreises beeinflusst. Resultat: wild schwankende Werte und Kurven, die mir wenig brachten.

In Ihrer Grafik aus dem Artikel sieht man hervorragend, wann Verbraucher ein- und ausschalten. Das ist bei mir leider überhaupt nicht der Fall. Der Temperaturunterschied zwischen Vor- und Rücklauf sollte allerdings (grob) Aufschluss über die entnommene Wärmemenge geben und kann sicher zur Optimierung der Heizkurve verwendet werden.

Sich der Heizung als größtem Energieverbraucher in Häusern zu widmen, ist eine fantastische Idee! An einer Weiterführung des Themas wäre ich sehr interessiert, gerade wenn man auch auf die ein oder andere Besonderheit von Altbauten eingehen könnte, wie fehlendem hydraulischem Abgleich oder einer falschen Einstellung/Auslegung der Pumpenleistung. Das Problem an meiner Heizung lag übrigens an einem Übergangswiderstand in einer Steckverbindung des Kessel-Temperatursensors (widerstandsbasierte Messung), sodass der Kessel etwa 70°C gemessen hat (~Solltemperatur), effektiv aber nur 30–40°C vorlagen. Digitale Sensoren (wie der DS18B20) hätten mich vor der aufwendigen Fehlersuche wahrscheinlich bewahrt.

Roy Meissner

Alternative AskSin

Ich habe seit Jahren eine Steuerung der Umlaufpumpe für Warmwasser mit zwei Sensoren im Einsatz. Öffne ich kurz den Wasserhahn, wird dies erkannt und die Pumpe läuft, bis das warme Wasser am Rücklauf angekommen ist – das ist natürlich nichts Neues. Das funktioniert im Prinzip ausreichend. Manchmal schaltet sie aber ohne Anlass, weil nur Vor- und Rücklauf gemessen werden und nicht die Speichertemperatur. Ist die Speichertemperatur niedrig, schaltet sie auch mal gar nicht. Deshalb denke ich schon seit einiger Zeit über eine Neugestaltung nach. Interessant nun, auch die Heizung mit einzubeziehen. Dabei stellt sich mir die Frage, ob ich dann nicht vier Sensoren brauche, weil die Vorlauftemperatur für die Fußbodenheizung eine andere als für das Warmwasser ist. Damit fällt der *Shelly* für dieses Vorhaben wohl aus.

Nun bin ich stark *Homematic*-lastig. In dem Bereich bietet sich vielleicht eine Lösung mit *AskSin* – eine Alternative für Homematic-Aktoren und -Sensoren auf Basis z. B. des *Atmel328p* – an. Ich habe im Laufe der Jahre verschiedene – nicht-Homematic-Komponenten – über mehr oder weniger aufwendige Lösungen (ESP32, CGI, XML-API, MQTT) in die Homematic eingebunden, bis ich vor einem halben Jahr auf AskSin gestoßen bin. Ein Paradies für DIY's, die Homematic bevorzugen. Ich habe schon verschiedene Lösungen nachgebaut.

Klaus Mühlbach



Disco is back – Lichtshow mit WLED

Mit der kostenlosen Firmware WLED auf ESP-Basis lassen sich Lichteffekte mit adressierbaren RGB-LEDs in wenigen Minuten aufbauen und über WLAN per App oder Browser steuern. Die Effekte reagieren auch auf Musik oder Geräusche und Installationen in verschiedenen Räumen lassen sich über das Netzwerk synchronisieren.

von Carsten Wartmann



ichtstimmungen in Räumen oder ambiente Beleuchtung z. B. hinter Fernsehgeräten sind spektakulär, verändern den Raum optisch und können die Stimmung der Bewohner positiv beeinflussen. Schnelle Effekte beeindrucken Besucher, die Kinder feiern eine Disco-Party im Kinderzimmer. Im Alltag werden eher ruhige und wenig wechselnde Szenarien eingesetzt. Und das beste: Für alles zusammen gibt es mit WLED eine einfach nutzbare, kostenlose Lösung zur Ansteuerung.

WLED ist eine Open-Source-Firmware für die bei Makern zu Recht so beliebten und günstigen ESP-Mikrocontroller-Boards. Durch die lange Entwicklungszeit und die aktive Community hat das Projekt einen Reifegrad erreicht, der so manches kommerzielle Produkt alt aussehen lässt. Tatsächlich nutzen auch einige der als fertige Produkte käuflichen LED-Controller diese Open-Source-Software. WLED funktioniert ohne Cloud, per App oder Browser, hat eine *HTTP-API* und beherrscht *MQTT* für Smarthome-Steuerung.

Die Hardware für die ersten Experimente ist schnell zusammengebaut ①, die Software blitzschnell per Browser installiert und die Konfiguration bequem per App oder Browser erledigt. All das macht das Projekt sehr einsteigerfreundlich. Hat man erst einmal Blut geleckt, kann es immer weiter ausgebaut werden, bis auch der letzte Winkel der Wohnung beleuchtet ist und das Smarthome alles steuert.

In diesem Sinne wird hier ein einfacher Aufbau gezeigt, der zum Einstieg für ca. 3 bis 5 Meter adressierbare LED-Streifen mit 5V geeignet ist. Ob und wie man diesen Streifen verlegt, z. B. als langen Streifen an der Wand, als umlaufende Unterbeleuchtung für die Couch oder ob man damit eine 2D-Matrix aufbaut, ist zunächst egal. Im weiteren Verlauf werden wir einzelne Räume mit separaten WLED-Installationen vernetzen und die sehr interessante Audiosteuerung vorstellen.

Hardware

Die Hardware ist so einfach, dass sie problemlos auf einer Lochrasterplatine **2**, **3** oder direkt am Board gelötet aufgebaut werden kann. Es gibt auch fertige Platinen mit einem Controller oder Platinen, auf die ein ESP-Board aufgesteckt werden kann.

Für die ersten Experimente reicht ein ESP-Board und ein USB-Netzteil: Damit kann man aber nur etwa 30 5V-LEDs mit 100 Prozent Dauerhelligkeit auf allen Farbkanälen versorgen. Auf keinen Fall sollte mehr als ein Ampere durch den ESP fließen. Um auf Nummer sicher zu gehen, kann die maximale Stromstärke im WLED-Interface begrenzt werden. Man muss nur daran denken, diese Option wieder auszuschalten, wenn das richtige Netzteil angeschlossen ist, sonst regelt WLED auch damit die Helligkeit herunter. Für Umgebungsbe-

Kurzinfo

» Einfache Hardware und Software mit ESP (8266 und 32)
 » Kopplung mehrerer WLED-Geräte per Netzwerk

» ESP32 als Lichtorgel reagiert auf Frequenz und Amplitude



ichtshow mit WLED





Aufbau mit ESP32 und digitalem Mikrofon





3 Hier meine fertige Platine, auf der ich anders als im Schaltungsvorschlag ein Netzteil aus der Bastelkiste fest verlötet habe. Die Sicherung sitzt in der Zuleitung.



4 Das sehr praktische Planungstool für die WLED-Installation

leuchtung und Effekte kann man damit aber schon recht weit kommen, da hierbei in den seltensten Fällen alle LEDs voll aufgedreht sind.

Mit externen Netzteilen, die den ESP und direkt den LED-Streifen versorgen, können dann mehr LEDs betrieben werden. Die Anzahl der LEDs ist übrigens die wichtigste Größe für die Dimensionierung, denn je nach Art des Streifens kommen hier 30, 60 oder sogar 144 LEDs pro Meter zum Einsatz! Will man mehr, braucht man Stromstärken jenseits der einstelligen Amperezahlen, Einspeisepunkte alle paar Meter in den LED-Streifen, Sicherungen gegen Beschädigung und Brandgefahr und sollte über den Einsatz von 12V-Streifen nachdenken. Einen guten Start bietet hier online der *WLED-Calculator* (Links in Kurzinfo).

Unsere Empfehlung: Kaufen Sie zunächst 5m 5V-LED-Streifen und ESP-Boards für ein bis zwei Installationen und schauen Sie, ob Ihnen die Effekte gefallen. Das ist schon für unter 50 Euro zu bekommen. Eine größere Installation kann schnell mehrere hundert Euro kosten und einiges an Arbeit für die Montage und eventuelle Umbauten in der Wohnung.

Testweise kann man das Projekt auch auf einem Breadboard aufbauen. Auch hier sollte ein Ampere nicht überschritten werden, um das Breadboard zu schonen. Auf einer Lochrasterplatine kann man dann dickere Kabel verwenden. Alle Komponenten, durch die der Strom zu den LEDs fließt, müssen für den maximalen Strom ausgelegt sein, auch hier hilft der oben erwähnte WLED-Rechner 4. Neben der Stromberechnung, mit der auch das passende Netzteil dimensioniert werden kann, werden auch Kabellängen, der Spannungsabfall darüber, die Sicherung und vieles mehr berechnet. Hilfreich ist das auch bei der Planung größerer Lichtinstallationen, denn zu groß dimensionierte Komponenten gehen schnell ins Geld, zu klein dimensionierte können eine Brandgefahr darstellen.

Mit einem ESP8266 kann man fast das gesamte Effektprogramm von WLED abrufen, aber eines kann der kleine Prozessor auf dem Board nicht: Die Steuerung per Audio, was besonders bei Kindern gut ankommt und auch als effektvolle Beleuchtung in Musikräumen sehr schön aussieht. Die Berechnung von Frequenzen und Amplituden erfordert ein ESP32-Board. Diese Funktion wird bei WLED *Audio-Reactive* genannt (dazu später mehr). Damit kann man Musik und Sound über Mikrofon (digital/analog), Line-In oder auch über Netzwerk in den ESP32 streamen, viele Effekte reagieren dann im Takt der Musik (§).

WLED unterstützt fast alle Arten von LED-Streifen: Mit zusätzlichen Schalttransistoren können auch nicht einzeln adressierbare RGB-LEDs verwendet werden, wenn es eher um eine allgemeine Lichtsteuerung geht. Wir haben allerdings nur die beliebten und preiswerten Streifen WS2812B (mit 5V) und SK6812 (ebenfalls 5V) ausprobiert und empfehlen diese für erste Versuche. Die SK6812-LEDs haben gegenüber den WS2812B noch eine extra weiße LED und sind daher besser geeignet, wenn der Streifen zwischendrin auch als möglichst helle Beleuchtung dienen soll. Reine RGB-Streifen erzeugen ein unruhiges und nicht reines Weiß, was besonders an den Schattenkanten auffällt, und verbrauchen viel mehr Strom, wenn alle drei Farbkanäle leuchten, um Weiß zu erzeugen.

Software aufspielen

Es spielt keine Rolle, ob Sie WLED nur testen oder gleich eine komplette Installation aufbauen wollen: Zuerst muss die Firmware auf das Board. Wer jetzt mit Schrecken an Software-IDEs, Kompilieren und Upload-Orgien von der Kommandozeile denkt: Keine Sorge, die WLED-Installation ist per Browser in wenigen Minuten erledigt, die Konfiguration erfolgt dann per WLAN und Browser oder über die App.

Das ESP-Board wird per USB an den Computer angeschlossen. Dies funktioniert unter *Windows, Linux* und *macOS* mit *Chrome* und *Edge*. Für die einfachste Installation besucht man anschließend die Webseite *install. wled.me* **6**.

Zuerst wird man vom Browser nach der Erlaubnis gefragt **?**, einen USB-Port benutzen zu dürfen. Dort sind etliche Ports aufgelistet und wenn man sich nicht sicher ist, welcher der richtige ist, kann man das ESP-Board kurz ausstecken. Der Port, der dann verschwindet, ist der richtige. Auch im Hardware-Manager von Windows kann man den Port finden.

Je nach gewählter Version startet ein Klick auf den *Install*-Button die Installation. Für Updates muss man das Board übrigens nicht aus dem Schrank holen, hier wird das Update einfach per WLAN (*Over The Air*, OTA) aufgespielt!



Die Effektbeleuchtung an Keyboardrack, die Rhythmen und Frequenzen visualisiert.

Die Installation dauert ein paar Minuten, dann ist man per USB mit dem WLED-Controller verbunden und kann direkt das gewünschte Netzwerk mit *SSID* und Passwort eintragen. Nach einem automatischen Neustart verbindet sich die WLED dann mit dem Router und erhält eine IP-Adresse, über die sie in Zukunft erreichbar ist. Diese muss man sich aber nicht unbedingt merken: Aus der WLED- App heraus kann man auch WLED-Controller im eigenen Netz entdecken (*discover*) und sich direkt verbinden.

Konfiguration

Nun geht es an die Konfiguration. Die WLED-GUI mit dem Farbwähler (8) sollte sich nach der Installation schon im Browser geöffnet



6 Online-Website für die Installation der Firmware. Sie können sich ruhig trauen, die Beta-Entwicklungsversion zu benutzen.

Projekt



haben. Die Einrichtung ist prinzipiell auch per App möglich, allerdings ist die Oberfläche im Browser für die Erstkonfiguration etwas übersichtlicher und man kann auch mehrere Browserfenster nebeneinander geöffnet haben.

Über das Zahnrad-Icon im Menüpunkt *Config* geht es dann mit der Konfiguration der Hardware **9** weiter.

Der wichtigste Punkt ist zunächst einmal LED Preferences (1). Hier muss der Typ der LEDs, die Anzahl der LEDs und der GPIO-Pin (Datenpin, GPIO2 bzw. auf Wemos-D1 mit D4 beschriftet, in unserem ESP-32 Schaltungsvorschlag GPIO32) eingetragen werden. Hier kann man nichts kaputt machen, evtl. leuchten die LEDs alle nicht (falscher GPIO oder Streifentyp) oder manche nicht (falsche Anzahl eingegeben) oder in falschen Farben (dann stimmt die *Color-Order* nicht).

Nach dem Einstellen der Parameter nicht vergessen oben auf der Seite auf *Save* zu klicken. Dann *mit Back* oben auf der Seite und nochmals mit *Back* im Einstellungsmenü zurück zum WLED-Interface.

Übrigens kann man unter dem Menüpunkt User Interface im Abschnitt Web Setup seine WLED-Instanz benennen, das hilft spätestens bei der Installation der dritten Instanz.

Interface

Die LEDs sollten nun in der Standardfarbe (orange) leuchten (falls nicht, zurück zum Hardware Setup). Wenn Sie nun den Farbwähler im ersten Reiter (*Color*) durch Anklicken oder Ziehen mit gedrückter Maustaste ändern und durch Loslassen der Maustaste bestätigen, wechselt die Farbe sanft von der aktuellen zur gewählten Farbe: Glückwunsch! Wie auch in ① zu sehen, können im PC-Modus alle Reiter gleichzeitig angezeigt werden.

Bei Streifen mit einer zusätzlichen weißen LED kann diese mit dem Regler *White channel* eingestellt werden. Unterhalb des Farbrades befinden sich einige Standardfarben und ein Knopf *R*, mit dem man eine Zufallsfarbe einstellen kann. Unter den Standardfarben befinden sich je nach Effekt noch ein paar runde, beschriftete Knöpfe: Diese werden für die Effekte verwendet.

Im zweiten Reiter *Effects* (bzw. in der zweiten Spalte im *PC Mode*) werden dem Streifen bzw. den Segmenten Effekte zugewiesen. Diese sind zum Teil spektakulär, aber es gibt auch ruhige Farbüberblendungen und ähnliches. Eine ganze Reihe von Effekten sind auch *Audio-Reactive*, zu erkennen an dem Noten-Icon, dazu später mehr. Wenn Sie keine Audio-Hardware angeschlossen haben, sind diese Effekte allerdings wenig aufregend.

Experimentieren Sie mit den Effekten, probieren Sie auch andere Paletten als die Standardpalette aus: Eine Kerzenanimation kann mit einem blauen Farbverlauf durchaus interessant sein. Auch die Parameter der Effekte laden zum Experimentieren ein, hier hilft eine ausführliche Webseite (Link in Kurzinfo), welcher Parameter bei welchem Effekt was bewirkt.

Für die verschiedenen Effekte erscheinen noch beschriftete runde Knöpfe unter den Standardfarben. Diese sind je nach Effekt mit 1,2,3 beschriftet, was einzelne Farben oder die

Configure Wi-Fi							
Enter the credentials of the Wi-Fi network that you want your device to connect to.							
Network Name UnserFritzi							
Password							
SKIP C	CONNECT						

7 So klickt man sich durch die Installation

Paletten 1 bis 3 meint. Weiterhin gibt es je nach Effekt noch *Fx, Bg oder Cs.* Ihnen kann man mit den Farbwählern Farben zuweisen, aus denen dann Paletten generiert werden. Diese sind in den Paletten darunter mit einem Stern (*) gekennzeichnet. Die *-Palette *Random Cycle* generiert einen zufälligen Farbverlauf.

Fx ist die Hauptfarbe des Effektes **1**2 oder einer Palette. Wenn man also die Farbe des Effektes auf eine einzelne Farbe einstellen will, muss man auf den Knopf Fx klicken und dann auf Color 1 aus der Palette. Das ist anfangs nicht ganz intuitiv: Man sollte bei diesen Schaltflächen immer in Paletten denken, auch wenn diese nur eine Farbe enthalten soll. Bg stellt die Hintergrundfarbe für den Effekt ein. Je nach Effekt werden die Paletten auch noch einmal anders interpretiert. So können sie zu einem Farbverlauf über den Streifen werden. zu einem Farbwechsel einer oder mehrerer LEDs oder es werden zufällig zwei Farben aus der eingestellten Palette gewählt. Hier hilft nur Experimentieren und die Peek-Funktion aus der Menüleiste, die eine Vorschau der LEDs in der GUI anzeigt und die Hilfeseite zu den Effekten.

Das alles ist schwerer in Worte zu fassen, je komplexer die Effekte werden: Daher experimentieren Sie! Das Ausprobieren und Anpassen der Effekte an den eigenen Geschmack ist sicher das, was am längsten dauert, aber auch am meisten Spaß macht.

Synchronisation

Wenn die WLED-Installationen wachsen und sich über mehrere Räume oder Bereiche erstrecken, ist es praktisch, durch eine Hauptinstanz (*Controller*) die anderen Instanzen steuern zu lassen, und genau dafür gibt es den *Sync*-Button. Jede Einstellung, ob manuell oder automatisch, die am Controller vorgenommen wird und Farben, Helligkeit, Effekte oder Presets betrifft, wird dann auf alle Instanzen übertragen.

In den Einstellungen findet man den Menüpunkt *Sync Interfaces* ^(B). Mit den *Sync Groups* kann man fein steuern, welche Geräte synchronisiert werden. Installationen, die verschiedenen Personen im Haushalt gehören, können so vom Besitzer ausgeschlossen werden oder bei Partys gemeinsam für Stimmung sorgen. Standardmäßig steuert der Sync-Button sowohl das Empfangen als auch das Senden von Sync-Nachrichten. Dies kann im Web-Setup geändert werden, übersteuert aber nicht die feineren Einstellungen in den Gruppen. Ist die Option deaktiviert, muss der Button nicht mehr auf allen zu synchronisierenden Instanzen gedrückt werden, was aber die Freiheit des Besitzers dieser Instanz nimmt.

Polizeieffekt zum Mittagessen zu rufen. Die Synchronisation funktioniert gut, wenn die Installationen keine zu unterschiedlichen Streifenlängen und vor allem keine komplizierten Segmenteinstellungen haben. Es wird immer erst synchronisiert, wenn man einen Parameter ändert! Für größere und komplizierte Installationen sollte man dann seinem Smarthome die Steuerung überlassen. Mittels der REST-API kann man WLED bis ins kleinste Detail steuern

nicht von außen gesteuert zu werden. So ist

es z. B. möglich, den Teenager durch einen

Ein kleiner Wermutstropfen: Sync funktioniert nicht, um Audioeffekte auf andere Installationen zu übertragen. Jede Instanz benötigt eine eigene Audio-Hardware und damit auch einen ESP32. Im nächsten Abschnitt zeigen wir aber, wie es mit virtuellen LEDs trotzdem funktioniert.

Audio Reactive

Mit dem Usermod Audio Reactive (früher auch abweichend Sound Reactive, SR) kann ein Mikrofon oder ein Line-Eingang die LEDs steuern. Das Video gibt einen Eindruck wie das aussieht. Live ist es noch beeindruckender. Leider hat ein ESP8266 nicht genug Rechenleistung, um das zu tun. Daher wird hier ein ESP32 benötigt, einen Bauvorschlag gibt

hat. Hier finden Sie auch die automatische Strombremse.

es in 🚯, das fertige Gerät kennen Sie bereits aus 🚺.

Wenn Sie ein Mikrofon verwenden möchten, sollten Sie eine digitale Variante wählen, d.h. ein Mikrofon, das die Daten über l²S an den Mikrocontroller sendet. Diese haben eine bessere Leistung und sind rauschärmer als analoge Mikrofone. Ich benutze ein INMP401 (von Dollatek) und bin sehr zufrieden. Es soll aber auch hier Nachbauten mit schlechter Qualität geben. Weitere digitale Mikrofone, z. B. mit ICS-43434, sind zurzeit schwer zu bekommen. Als analoge Alternative wird von den Machern der WLED besonders das MAX9814 empfohlen, das aber oft auch nicht viel billiger als ein digitales ist.

Die Konfigurationen und Einstellungen werden im Einstellungsmenü Usermods vorgenommen. Im Unterpunkt AudioReactive 🚯 wird zuerst Enabled angehakt. Für analoge Mikrofone muss unter Analogmic/Pin der GPIO eingetragen werden, -1 deaktiviert diese Funktion. Unter Digitalmic sollte für gängige Digitalmikrofone Generic I2S gewählt werden und dann die Pins SD, WS und SKC entsprechend der Schaltung gewählt werden. Der Pin MCLK kann für Line-In zu I²S Chips erforderlich sein.

Wenn Sie die Type ändern mussten, kann ein Neustart erforderlich sein, den Sie entweder durch den Reset-Taster auf dem Board, Ziehen des Steckers oder im Menü Info mit Reboot-WLED auslösen können. Der Neustart dauert nur wenige Sekunden.

Wenn Sie jetzt einen audioreaktiven Effekt auswählen, sollten Sie bereits durch lautes Klatschen oder Pfeifen eine Reaktion der LEDs auslösen können.

Nun gilt es, einen Effekt für die Konfiguration zu finden: In der Suche kann man das Unicode-Zeichen für eine Note eingeben, dann wird die Effektliste nach audio-reaktiven Effekten gefiltert. Einfacher ist es aber, das kleine Filtermenü zu benutzen. Hier kann man ein Icon auswählen: Neben Noten gibt es auch



8 Mit einem WS2812b-Streifen und Standard-GPIOs sollte die Farbwahl bereits funktionieren.

Filter für Effekte, die Paletten benutzen oder 2D-Matrix-Effekte.

Die Einzelnote bedeutet, dass die Effekte nur auf die Lautstärke (Amplitude) reagieren. Die gebundene Doppelnote filtert nach frequenzabhängigen Effekten. Dies ist sicherlich der Grund, warum diese Erweiterung nur auf ESP32 läuft: Die dahinterliegende Fast Fourier Transformation (FFT) benötigt einige Rechenleistung, um in Echtzeit ein Spektrum der detektierten Frequenzen zu berechnen. Damit kann der Effekt dann z.B. die Farbe je nach Frequenz anpassen. Mein Lieblingseffekt Freqmatrix 16 sendet damit frequenzabhängig farbige Lichter entlang des LED-Streifens,



🕕 Die Kernparameter für die Hardware: LED-Typ, Farbreihenfolge, Anzahl der LEDs, GPIO.



Wenn Ihr Browserfenster groß genug ist, können Sie die Tabs im PC-Modus nebeneinander in einem Fenster anzeigen.

je nach Länge des Tons, wie bei einem Lauflicht (siehe auch Video). Freqmatrix ist eine der besten Visualisierungen die ich kenne; man kann die Musik praktisch *sehen. Freqmatrix* eignet sich daher auch sehr gut zum Einstellen des Mikrofons.



Die Effekt-Paletten: Hier jagt ein blauer Lichtfleck (Cs) einen rosa Fleck (Fx) auf gelbem Hintergrund (Bg). Mit der Palette Color Gradient werden die Übergänge der Farben weich. Mit Gain 1 wird die Verstärkung des Mikrofons eingestellt. Experimentieren Sie hier, bis Sie eine gute Balance zwischen leisen und lauten Tönen gefunden haben.

Danach sollte die Störgeräuschunterdrückung (*Squelch*) so weit erhöht werden, dass normale Nebengeräusche im Raum oder Rauschen des Mikrofons keinen Effekt mehr haben, bei digitalen Mikrofonen ist dieser Wert eher niedrig.

AGC (Automatic Gain Control, automatische Aussteuerung) ist für ein Mikrofon sehr nützlich, AGC sorgt für einen ähnlichen Effekt, egal, wie laut die Musik ist. Ich habe es auf Normal eingestellt, Vivid regelt schneller, Lazy langsamer.

Unter Frequency/Scale kann man einstellen, wie auf den Frequenzgang reagiert wird, für mich passt Square Root (Energy) am besten.

Audioeffekte per Netzwerk

Möchte man von dem ESP32 mit Mikrofon weitere WLED-Geräte ansteuern, benutzt man virtuelle LEDs. Hierbei werden die Echtzeitdaten der FFT über das Netzwerk (UDP) gesendet. Die Einstellungen finden Sie unter LED Preferences. Mit dem Plus-Icon unter Hardware Setup einen neuen Strip () hinzufügen und als Typ DDP RGB (network) wählen und die Anzahl der zu verwendenden LEDs eingeben. Der Wert Start: entspricht der Anzahl physikalischer LEDs auf der steuernden Instanz, sodass der Streifen praktisch virtuell verlängert wird. Wenn das nicht gewünscht ist, kann man Custom bus start indices ankreuzen und einen Start von 0 wählen, dann wer-



Sync-Einstellungen meiner Hauptinstanz – sie sendet auf den Gruppen 1 und 2 und empfängt nie.

den die Streifen parallel geschaltet – das sieht besser aus, wenn die LEDs räumlich getrennt sind.

Auf der empfangenden WLED-Instanz muss dann in *Sync Interfaces* unter *Realtime* noch der Haken *Receive UDP realtime* gesetzt werden. Auf den Netzwerkinstanzen erhält man in der WLED-GUI einen Hinweis, wie die Steuerung umgangen oder ausgesetzt werden kann ().

Gib mir mehr!

Wie schon erwähnt, ist WLED auch über die Rest-API bis ins kleinste Detail steuerbar, was die Integration in ein Smarthome sehr einfach macht. Die Software ist aber schon recht komplex, daher hier einige interessante Parameter, die man sonst nicht so leicht findet. In 20 wird gezeigt, wo man den gewünschten Effekt nach Neustart einstellt.

Eine klassische Einschlaffunktion kann mit dem *Timer* (Mond-Icon) in der Oberfläche aktiviert werden. Die Dauer ist standardmäßig auf eine Stunde eingestellt, in der die Helligkeit der LEDs auf null reduziert wird. Da es sich um eine globale Funktion handelt, unabhängig vom geladenen Preset/Effekt, finden Sie die Einstellungen für die Einschlafzeit in den LED-Präferenzen unter dem Punkt *Timed light*.

Der Countdown-Modus ist eine lustige Funktion, mit der man anhand der Anzahl der leuchtenden LEDs sehen kann, wie lange es noch bis zu einem Ereignis dauert, sei es Weihnachten, die nächste Make-Ausgabe oder der Jahresurlaub. Die Funktion versteckt sich unter Clock im Menü Time & Macros.





AudioReactive Enabled Analogmic Pin -1 Digitalmic Type Generic I2S requires reboot! Pin I2S SD 13 sd/data/dout ws/clk/lrck Pin I2S WS 12 sck/bclk Pin I2S SCK 14 Pin I2S MCLK -1 only use -1, 0, 1 or 3 Pin I2C SDA -1 Pin I2C SCL -1

(6) Der Effekt Freqmatrix mit den von mir im Video benutzten Einstellungen

WLED kann die aktuelle Uhrzeit von einem ntp-Server im Internet holen, die Konfiguration findet sich unter *Time & Macros*. Damit sind tageszeitabhängige Lichtszenen möglich. Neben dem Laden von Presets nach Stunde, Minute, Datum und Wochentag (Kalender-Icon) findet man hier auch das Laden von Presets bei Sonnenaufgang und Sonnenuntergang. Die exakte Berechnung des Sonnen-

B Kontiguration fur die Audio-Steuert	ung per
Mikrofon	

standes wird durch die Einstellung der eigenen Geolokation ermöglicht.

Letztlich bleiben nicht viele Wünsche übrig. Ich würde mir z. B. noch einen Modus wünschen, bei dem etwa beim Fehlen von Audio-Input auf einen anderen Preset geschaltet wird, aber vielleicht gibt es diese Funktion schon als *Usermod* und ich habe sie nur noch nicht gefunden. —*caw*



Einstellungen f
ür Dynamik und Frequenzgang



Welcher Effekt beim Booten/Einschalten des Boards aktiviert wird, ist etwas in den LED Preferences versteckt.

2:	DDP R	GB (netw	rork) ~	
Start:	100	Lengt	th: 128	
P address:	192	168	0	120

(B) Erweiterung des LED-Streifens per Netzwerk. Für Streifen mit extra weißer LED wählt man DDP RGBW (network).

W	LED is receiving live D	☆ DP data from 192.168.0.116
T u can reti	o use built-in effects, us urn to realtime mode by	se an override button below. pressing the star in the top left corner.
	Override once	Querride until reheat

Wer macht hier Party? Der Besitzer kann die Netzwerksteuerung umgehen.

Internet+UKW-Radio mit ESP32 und Akku

Niedlich und nützlich: Dieses kleine Selbstbau-Radio hat ein hübsches Gehäuse und funktioniert auch, wenn mal kein Stromnetz oder WLAN verfügbar ist. Und falls Sie auf Design-Wechsel stehen, können Sie dank der Chassis-Bauweise das Gehäuse auch schnell wechseln, falls mal das kleine Schwarze besser passt ...

von Heinz Behling



iele, insbesondere Jüngere, hören heutzutage kaum noch Radio. Ich meine: wirkliches Radio, also nicht via Internet gestreamte Radiostationen, sondern drahtlos vom nächsten Sendemast/Fernsehturm aus ausgestrahlte UKW-Sender. Dabei ist das doch eine praktische Einrichtung, weil sie fast überall funktioniert. Das lernt man aber wohl erst wirklich zu schätzen, wenn einmal keine Datenverbindung zur Verfügung steht, sei es auf Wanderungen fernab vom Funknetz, weil der Smartphone-Akku gerade leer ist oder weil wegen sonstiger widriger Umstände (Unwetter) lokal das Internet-/ Handynetz ausgefallen ist.

Wie es auch sei: Mit diesem kleinen Radio sind Sie allen Fällen gewappnet: Es kann nämlich sowohl aus dem Internet via WLAN Radiostationen streamen als auch per Ultrakurzwelle direkt empfangen. Durch den eingebauten Akku funktioniert das auch, falls kein Stromanschluss zur Verfügung steht.

Die Elektronik-Hersteller machen uns das auch recht einfach: Statt wie früher beim Radio-Selbstbau noch mühsam Spulen zu wickeln, Schwingkreise mechanisch möglichst stabil zusammenzubauen und alles hinterher auch noch mit einem nicht unerheblichen Messgeräteaufwand genau abzugleichen, bekommt man heute einen kompletten UKW-Tuner auf einem kleinen Platinchen 1, das kaum Briefmarkengröße erreicht.

Allerdings muss man dem IC (SI4703) darauf noch irgendwie mitteilen, welche Frequenz er empfangen soll. Dafür muss ein Mikrocontroller herhalten, der mit dem Tuner per I²C-Bus kommuniziert. Unser Radio verwendet einen ESP32, der mit der Steuerung des UKW-Tuners allein aber gewaltig unterbeschäftigt wäre. Allerdings wollen wir ja auch Internet-Stationen per WLAN empfangen. Und dafür bringt

Kurzinfo

» Stereo-Radioschaltung mit ESP32 und UKW-Empfänger » Steuerung per Rotary-Encoder

» Firmware mit erweiterbarer Senderliste



- » Christian Schweigert, Anleitung: Senioren-taugliches Internet-Radio mit Raspberry Pi bauen, Make 1/22, S. 14
- » Christoph Goebel, Internet-Radio mit Pi Zero, Make 1/19, S. 38

Alles zum Artikel im Web unter make-maaazin.de/xeir

Material

» ESP32-Board » LC-Display 2004 mit I²C-Adapter » PC-Pegelwandler 3V-5V » Step-up-Wandler MT3608 » Laderegler HW107 » LiPo-Akku Eremit 104050 » 2 Lautsprecher Visaton K50/8 Ohm » Stereo-Verstärker-Modul 5V, 3W » Rotary-Encoder KY-040 » UKW-Empfänger Sparkfun SI4703 » 2 Widerstände 22 Kiloohm » 2 Widerstände 4,7 Kiloohm » 1 Widerstand 10 Kiloohm » 4 Schrauben M3 × 8 Linsen- oder Inbus-Kopf » 7 Schrauben M3 × 10 Senkkopf » Kabel in verschiedenen Farben zur Verdrahtung der Bauteile » Antenne aus Schaltdraht oder Schaltlitze (Wurfantenne)

Werkzeug

- » Lötkolben
- » Heißklebepistole
- » Seitenschneider, Schraubendreher
- » Multimeter



 So klein ist heutzutage ein UKW-Empfänger. Der einzelne Lötanschluss oben links ist für die Antenne.



2 Die vier Teile des Chassis werden mit M3-Schrauben und Muttern miteinander verbunden.



der ESP32 alles Notwendige mit, insbesondere WLAN und **D**igital-**A**nalog-**C**onverter (**DAC**), die den Datenstrom seines internen I²S-Busses in analoge Signale umwandeln und einen kleinen Niederfrequenz-Verstärker ansteuert. Der Verstärker wiederum speist dann zwei kleine Lautsprecher. Das bringt dann insgesamt zwar kein Hifi, aber für den Preis doch eine recht gute Klangqualität. Der Controller hat aber noch etwas mehr zu tun: Er kümmert sich auch um das Display, das Ihnen den jeweiligen Stationsnamen sowie zusätzliche Titelinfos präsentiert. Außerdem wertet er den Dreh-Encoder (auch Rotary-Encoder genannt) aus, mit dem man das Gerät bedient. Schließlich steuert er noch eine 2fach-Relaiskarte an, die den Eingang des Verstärkers entweder mit den Tonausgängen des UKW-Moduls oder mit denen des ESP32 verbindet. Nebenbei bemerkt: Hier wäre natürlich auch eine Umschaltung mit elektronischen Bausteinen möglich gewesen. Allerdings hätte man dann für diese ICs eine Lochrasterplatine verwenden und diese von Hand verdrahten müssen. Das erspart die Relaiskarte, die im Preis auch nicht teurer ist.



4 Der zentrale eckige Lötpunkt an der Unterseite des Verstärkers ist die Spannungsquelle für alle anderen Radio-Bestandteile.



S Mit sanfter Kraft in Form eines Schraubendrehers drücken Sie die Muttern in ihre sechseckigen Öffnungen.



6 Für die Leitungen zwischen Laderegler, Wandler und dem Verstärker sollten Sie Schaltlitze mit mindestens 0,25mm² Querschnitt verwenden.



Werden Display und I²C-Platine getrennt geliefert, ist noch etwas



8 So muss die Platine auf der Rückseite des Displays sitzen. Verlieren Sie beim Löten nicht den kleinen Jumperstecker an der linken Seite des Interfaces, sonst funktioniert hinterher die Hintergrundbeleuchtung nicht.

Und da das Auge ja schließlich auch mithört, sitzt das Ganze in einem Gehäuse, das sich an alten Röhrenradios orientiert. Falls Ihnen mein Entwurf nicht gefällt, kein Problem, denn die äußere Hülle lässt sich schnell wechseln.

Chassis-Bauweise

Das Radio ist nämlich, wieder dem Vorbild der Röhrenradios entsprechend, als eigenständiges Chassis aufgebaut. Schon früh benutzten die diversen Radiohersteller diese Bauweise. Solche Chassis wurden dann in unterschiedlichste Gehäuse eingebaut. Lediglich an der gleichen Anordnung der Bedienelemente zweier Radios konnte man erkennen, dass dieselbe Technik drinsteckt. Falls Sie Ihrem Radio-Selbstbau also mal ein besser passendes Kleid verpassen möchten, ist das nach Lösen weniger Schrauben schnell getan (Tipp: Das kleine Schwarze passt zu allen Gelegenheiten!).

Das Chassis besteht aus vier 3D-Druckteilen: Die Bodenplatte nimmt die meisten Teile auf. An den beiden Seitenelementen sitzen die Lautsprecher und in der Frontplatte erhält das Display, der Verstärker nebst Lautstärkeregler sowie der Dreh-Encoder seinen Platz 2.

Durch die Vierteilung kann man das Chassis ohne aufwändige Stützstrukturen drucken. Das spart eine Menge Zeit und Geld. Falls Sie keinen 3D-Drucker zur Verfügung haben, können Sie das Chassis mit etwas Geschick auch aus Sperrholz und kleinen Leisten bauen.

Damit Sie statt meines Gehäuses (Download über Kurzinfo-Link) sich selbst ein Radiogehäuse konstruieren können, sind die dazu notwendigen Dateien ebenfalls erhältlich. Insbesondere finden Sie dort in einer STL-Datei auch den Umriss des Chassis. Damit können Sie Ihre Konstruktionszeichnung vor

dem Druck darauf prüfen, ob das Radiochassis auch hineinpasst.

Schaltung

Die Schaltung sieht im Schaltplan 3 aufwändiger aus, als sie ist.

Zentrale ist der ESP32: Rechts von ihm sehen Sie den I²C-Bus-Teil mit dem LC-Display (4 Zeilen zu je 20 Zeichen) und dem Tunermodul. Da der Tuner und der ESP32 mit 3,3V an ihren Anschluss-Pins arbeiten und insbesondere das Radiomodul empfindlich auf höhere Spannungen reagiert, ist das LC-Display über einen bidirektionalen Pegelwandler angeschlossen, der seine 5V-Signale in 3,3V und umgekehrt konvertiert.

Unterhalb des Mikrocontrollers sitzt der Niederfrequenzteil, bestehend aus der Relais-Platine, dem Verstärker sowie den Lautsprechern. Das Verstärkermodul hat noch eine Sonderaufgabe: Da sein Potenziometer über einen Schalter verfügt, benutzen wir es, um darüber auch alle anderen Bestandteile einund auszuschalten. Deshalb wird die 5V-Versorgungsspannung für Controller, Display, Relaiskarte usw. an einem kleinen, eckigen Lötpunkt an der Unterseite der Verstärkerplatine abgenommen 4.

Unter dem Niederfrequenzteil sitzt die Stromversorgung: Sie besteht aus einem Li-Po-Akku mit 2500mAh, einem Laderegler, der den Akku per USB-Anschluss aufladen kann, und einem Step-up-Wandler, der aus den 4,2V des Akkus 5V für die Elektronik macht. Beim Akku ist auf seine Maße zu achten: Er sollte 10mm dick, 40mm breit und 50mm lang sein. Für diese Maße ist das Aufnahmefach im Chassis-Boden gebaut.

Schließlich steht im Schaltplan ganz links noch der Dreh-Encoder. Über zwei Leitungen erhält der ESP32 die Signale beim Drehen seiner Achse. Ein dritter Pin erhält die Information, ob der Knopf des Encoders gedrückt wurde. Achtung bei der Teile-Beschaffung: Es gibt auch Encoder ohne Schalter. Die haben dann nur vier Anschlüsse. Wer sicher gehen will, dass alle Teile passen, erhält über den Kurzinfo-Link die Bezugsadressen für alle Elektronik-Teile.

Aufbau

Nach dem Druck der Chassis-Teile sollten Sie als Erstes mithilfe einer M3-Schraube die Muttern in die beiden Aufnahmeöffnungen der Bodenplatte und die drei Öffnungen der Front einpressen. Dabei aber keine Gewalt anwenden: Die Muttern passen nur bis zur halben Höhe in die sechseckigen Öffnungen

Schrauben Sie das Chassis aber noch nicht zusammen. Erst bauen wir die Elektronik in die vier Druckteile ein. Beginnen wir mit der Stromversorgung 6.



9 Der Widerstand ist auf der Encoder-Platine am besten aufgehoben.



D ESP32 ist an Bord, ebenso die Relais-Platine.



🕕 Das Radio-Modul mit seiner "Highend-Antenne"



Beim Anschluss der Leitungen müssen an einige Pins des ESP-Boards zwei Litzen angelötet werden.



Die Lautsprecher werden eingeklebt.

Achten Sie dabei peinlich genau auf die Polaritätsangaben von Akku, Laderegler und Step-up-Wandler. Insbesondere beim Laderegler und dem Wandler stehen sich jeweils Minus- und Pluspole gegenüber. Die Verbindung muss daher gekreuzt werden.

Befestigt werden diese und die meisten anderen Teile der Elektronik mithilfe von Heißkleber. Machen Sie das aber bei der Stromversorgung erst, wenn Sie alle Kabel angelötet und den Step-up-Wandler eingestellt haben. Der besitzt nämlich ein kleines, per Schlitzschraubendreher bedienbares Potenziometer (blaues Teil auf der Platine), mit dem seine Ausgangsspannung auf 5,1 bis 5,2V eingestellt werden muss. Dazu muss der Laderegler per USB-Kabel mit einem passenden Netzteil oder der USB-Buchse eines Computers verbunden sein. Die Spannung messen Sie mit einem Multimeter an den beiden Punkten VOUT+ und VOUT-. Wundern Sie sich nicht, wenn Sie eine Weile am Potenziometer kurbeln müssen. Es ist ein sogenanntes 10-Gang-Poti und die richtige Einstellung findet sich meist erst kurz vor einem der beiden Endanschläge, erkennbar an einem leisen Klicken im Poti.

Als Nächstes widmen wir uns dem Display: Diese Displays werden oft mit nicht eingelötetem l²C-Interface geliefert **7**.

Achten Sie beim Einlöten der Platine auf der Rückseite des Displays auf die korrekte Richtung, sonst fabrizieren Sie später einen prächtigen Kurzschluss **8**!

Löten Sie danach an die vier Anschlusspins des Interfaces jeweils etwa 20cm Schaltlitze. Ich empfehle dringend, sich dabei an die Kabelfarben des Schaltplans zu halten. Das erleichtert die Verbindungsarbeiten erheblich. Im Plan habe ich grundsätzlich rot für 5Vund gelb für 3,3V-Versorgungsspannung verwendet. GND ist schwarz. Verwechslungen zwischen 5V und 3,3V führen fast immer zur Zerstörung diverser Teile. Mit vier M3-Schrauben befestigen Sie das Display an der Rückseite der Frontplatte. Die Bohrungen für



Or Pin, der im Bild nach rechts weist, führt 5V, der andere GND.

die Schrauben sind so eng, damit sich die Schrauben selbst ihr Gewinde hineinschneiden.

Auch den Dreh-Encoder versehen Sie mit Schaltlitzen. Den 10-Kiloohm-Widerstand zwischen den Anschlüssen SW und + löten Sie direkt auf die Unterseite der Encoder-Platine. Schrauben Sie den Encoder dann in die Frontplatte 9.

Die Relais-Platine sollten Sie ebenfalls bereits mit drei Schaltlitzen versehen, und zwar an den Anschlüssen VCC (rot), GND (schwarz) und IN1/IN2 (grau). Die beiden IN-Pins (Steuerpins für die beiden Relais) können zusammen mit einer Litze verbunden werden. Mit einem dicken Klecks Heißkleber unter der Platine kleben Sie sie in die Bodenplatte. Ebenso befestigen Sie das ESP32-Board (USB-Buchse an die Kante der Bodenplatte) 10.

Bereiten Sie anschließend die Tuner-Platine vor. Beachten Sie, dass im Schaltplan die Unterseite der Platine mit den entsprechenden Anschlussbezeichnungen gezeigt ist. Wichtig: Die Tuner-Platine besitzt an einer Ecke einen allein stehenden Lötpunkt für die Antenne. Ich habe dort gut 10cm Schaltdraht eingelötet 11. Das erbrachte in meiner Gegend einen ausreichenden Empfang. Sollten die Empfangsverhältnisse bei Ihnen schwieriger sein, löten Sie dort etwa einen halben Meter Schaltlitze an. Die führen Sie später aus dem Gehäuse als sogenannte Wurfantenne.

Jetzt kommt etwas mehr Lötarbeit: Schließen Sie den Pegelwandler und das Radiomodul sowie die Leitungen der Relais-Platine am ESP32 an. Außerdem verbinden Sie die beiden Tonausgänge des Tuners mit der Relais-Karte wie im Schaltbild angegeben. Zu lange Schaltlitzen sollten Sie kürzen, das erhöht die Übersicht **1**2.

28	String	ssid = "SSID1";
29	String	password = "PASSWORT1";
30	String	ssid2 = "SSID2";
31	String	<pre>password2 = "PASSWORT2";</pre>

Das Radio kann f
ür zwei Netzwerke konfiguriert werden.



🚺 Beim Hausbau wäre es jetzt Zeit fürs Richtfest: Der Rohbau ist fertig.

Weiter geht es mit den Lautsprechern: Sie werden in die Seitenteile mit 2-Komponenten-Kleber eingeklebt. Den Kleber geben Sie in die Nut der Seitenwand und setzen dann den Lautsprecher so ein, dass er eben liegt. Löten Sie nach Aushärten des Klebstoffs jeweils etwa 20cm lange Litze an deren Anschlüsse **B**.

Kommen wir nun zum Verstärker. An seiner Unterseite löten Sie zwei Kontaktpins ein, die künftig die per Poti-Schalter geschaltete 5V-Leitung und die GND-Leitung darstellen (4).

Jetzt wird es Zeit, die Boden- und die Frontplatte zusammenzuschrauben. Hier bitte M3-Schrauben mit Senkkopf verwenden. Entsprechende Vertiefungen sind in der Bodenplatte vorhanden. Danach beginnt die Fleißarbeit: Vervollständigen Sie die Verdrahtung zwischen Display und Pegelwandler, dem Verstärker, der Relaiskarte, dem Dreh-Encoder und dem ESP32 **(5**).

Schließlich löten Sie noch die Lautsprecherdrähte an den Verstärker und schrauben die Seitenteile ans Chassis. Damit ist der Rohbau des Radios fertig **16**.

Software

Ohne steuernde Software bleibt das Radio aber stumm. Die Software erhalten Sie über den Kurzinfo-Link. Laden Sie sie auf Ihren



Ib Halten Sie die Kabel möglichst kurz, sonst verlieren Sie die Übersicht.



B Ganz links ist der ESP32 mit seiner USB-Buchse. Rechts daneben und etwas tiefer liegt der Laderegler.

Die WLAN-Senderliste

Die Senderliste *WLANsendelist.ino* besitzt für jeden Internet-Radiosender zwei Zeilen. Hinter stationsname folgt zunächst die Stationsnummer (auf der Anzeige wird sie um 1 höher angezeigt, damit es keine Station 0 gibt). Danach folgt der Sendername. Die zweite Zeile enthält nach dem gleichen Schema die Internet-Adresse der Station.

Wichtig: Am Schluss der Liste (und nur dort) muss als Stationsname und -adresse UKW eingetragen sein.

Die Liste können Sie nach Belieben erweitern. Es ist noch ausreichend Speicherplatz auf dem ESP32 frei. Beachten Sie aber, dass Sie neue Einträge entsprechend weiter durchgehend nummerieren müssen. Haben Sie Ihre Liste komplett, muss im Hauptprogramm noch die Anzahl der Stationen in der Variablen STATIONS (Zeile 21) angepasst werden. Danach kompilieren Sie die Firmware erneut und überspielen sie auf den ESP. Nach dem nächsten Start stehen dann die neuen Sender bereit.

```
stationname[0] = "NDR Kultur";
  stationurl[0] = "http://icecast.ndr.de/ndr/ndrkultur/live/mp3/128/
stream.mp3":
  stationname[1] = "NDR2 Niedersachsen";
  stationurl[1] = "http://icecast.ndr.de/ndr/ndr2/niedersachsen/mp3/128/
stream.mp3";
  stationname[2] = "NDR1 Hannover";
  stationurl[2] = "http://icecast.ndr.de/ndr/ndr1niedersachsen/hannover/
mp3/128/stream.mp3";
  stationname[3] = "WDR1";
  stationurl[3] = "http://wdr-1live-live.icecast.wdr.de/wdr/1live/live/
mp3/128/stream.mp3";
  stationname[22] = "ABC Radio Australia";
  stationurl[22] = "http://live-radio01.mediahubaustralia.com/RAPW/mp3";
  stationname[23] = "104,6 RTL Berlin";
  stationurl[23] = "http://stream.104.6rtl.com/rtl-live/mp3-192";
  stationurl[24] = "UKW";
  stationname[24] = "UKW";
```

Computer und entpacken Sie die ZIP-Datei. Im Software-Verzeichnis finden Sie den Ordner *MakeRadio*. Er beinhaltet vier Dateien:

- MakeRadio.ino
- rotary.ino
- UKW.ino
- WLANsenderlist.ino

Das Hauptprogramm ist *MakeRadio.ino*. Laden Sie es in der Arduino-IDE (ich habe Version 1.8.19 verwendet). Die anderen drei Dateien werden ebenfalls mit geladen und erscheinen zunächst als Reiter oben im Programmcode-Fenster. In den Zeilen 28 und 29 des Hauptprogramms müssen Sie die Daten Ihres WLANs eintragen **1**. SSID1 und PASSWORT1 sind dazu einfach zu ersetzen. In den beiden folgenden Zeilen können Sie auch noch ein zweites WLAN einsetzen.

Falls das Radio beim späteren Start das erste WLAN nicht findet, versucht es, mit dem zweiten Kontakt aufzunehmen. Ganz praktisch fürs Wochenendhaus.

Unser, oder besser Ihr neues Radio braucht noch ein paar Software-Bibliotheken. Die *ESP32-audiol2S* sowie *radio* gibt es bei Github. Laden Sie beide als ZIP-Datei herunter und installieren Sie sie mit *Sketch/Bibliothek einbinden/.ZIP-Bibliothek hinzufügen*. Hinzu kommen noch einige Bibliotheken, die Bestandteil der IDE sind und daher mit *Sketch/Bibliothek einbinden/Bibliotheksverwalter* zu installieren sind. Im einzelnen:

- PU2CLR SI4703 (Version 1.0.4)
- LiquidCrystal_I2C (Version 1.1.2 von Marco Schwartz)
- AiEsp32RotaryEncoder (Version 1.4.0)

Wenn Sie alles beisammen haben, stellen Sie unter Werkzeuge noch als Board ESP32 Dev Module sowie die entsprechende Portnummer ein. Schließen Sie den ESP32 des Radios per USB an den Computer an ^(B). Schon kann es losgehen: Kompilieren Sie die Firmware und übertragen Sie sie auf den ESP. Das kann ein bis zwei Minuten dauern. Bei manchen ESP32-Boards müssen Sie, sobald die IDE die Connecting-Meldung anzeigt, den Boot-Taster für drei bis vier Sekunden drücken. Manche Boards erfordern das nur beim erstmaligen Aufspielen einer Firmware, andere bei jedem.

Danach kann es schon losgehen: Schalten Sie das Radio ein. Ist der Akku geladen? Andernfalls über die USB-Buchse des Ladereglers (Bild B rechts) nachholen. Das Radio sollte sich innerhalb von etwa 10 Sekunden mit einem der beiden WLANs verbinden und dies auch anzeigen. Der erste Sender aus der Senderliste *WLANsendelist* ertönt. Die Liste können Sie übrigens erweitern (siehe Kasten *Die WLAN-Senderliste*).

Danach können Sie mit dem Dreh-Encoder andere Sender aus der Senderliste auswählen. Die Stationsnummer und der Sendername wird beim Drehen angezeigt. Wenn Sie dann

Projekt



innerhalb von 10 Sekunden den Encoder drücken, schaltet das Radio zum entsprechenden Sender um. Falls sie nicht drücken, wechselt die Anzeige wieder zur ursprünglichen. Auf dem Display erscheinen zusätzlich Titelinfos () u. ä., je nachdem, was die jeweilige Station an Infos mitsendet.

Eine Sonderrolle bei der mitgelieferten Senderliste spielt die Station 25 mit dem treffenden Namen *UKW*. Wenn sie die wählen und den Encoder drücken, wird auf UKW-Empfang umgeschaltet. Der Dreh-Encoder funktioniert dann wie ein Frequenzwähler an einem herkömmlichen Radio. Haben Sie einen Sender gefunden und der strahlt RDS-Signale aus, zeigt das Radio auf dem Display nicht nur dessen Frequenz, sondern nach einigen Sekunden auch den Sendernamen an **21**. Um wieder zum Internet-Radio zurückzukommen, drücken Sie den Encoder und schon erscheint wieder Station 1 aus der Senderliste.

Gehäuse

Noch ist der kleine Radiologe aber ein Rohbau. Es fehlt ein passendes Kleid. Das gibt



es in Form von 3D-Druckdateien über den Kurzinfo-Link. Ich hab einige Versionen davon konstruiert: Mit Frontplatte und ohne, mit Lautsprecher-Gitter und ohne. Suchen Sie sich das passende aus und drucken Sie es. Vergessen Sie dabei aber nicht die Rückwand (so ganz im Retro-Style geschlitzt wie bei alten Röhrenradios). Passende Knöpfe hab ich auch beigelegt.

Aber Sie können natürlich auch selbst kreativ werden und ein eigenes Gehäuse konstruieren. Wandeln Sie einfach meine Entwürfe ab. Ich hab die Gehäuse in Tinkercad gebaut. Per Download können Sie auch eine Datei namens *Form.stl* bekommen (1). Importieren Sie diese in Tinkercad. Wenn Sie die Datei als Bohrung in einen Quader mit den Ihnen genehmen Außenmaßen einsetzen und beides miteinander kombinieren, bleibt ein Gehäuse übrig, in das das Radio passt.

Und bitte: Lassen Sie uns daran teilhaben. Senden Sie uns ein Foto oder auch die Druckdateien (gerne auch eine Beschreibung, wo Sie das Radio nutzen), damit andere ihrem Radio ebenfalls das passende Outfit verleihen können. — hgb



Das ist praktisch die Gussform, die Sie in Tinkercad für eigene Gehäuse benutzen können.

OXOCARD SCIENCE ENTDECKE UND ERFORSCHE DEINE UMWELT

Programmierbare Multisensorplatine mit hochauflösendem Farbscreen, fünf Tastern und vielen Sensoren.

Die sieben eingebauten Umweltsensoren liefern zwölf Sensorwerte, mit denen du experimentieren kannst.

- Temperatur
- Luftdruck

· VoC

- Feuchtigkeit
- Licht
 Infrarot

· Luftqualität (IAQ)

- · Dezibel
- Frequenzen
- CO₂ (berechnet) Fre • Ethanol (berechnet) • Bes
 - Beschleunigung





Jetzt im Heise-Shop bestellen

ORIGINALGRÖSSE – 55×85mm



- sie braucht dich und du sie

Einmal im Jahr findet in Hannover die zweitgrößte Maker Faire in Europa statt. Als Community-Event bringt sie Maker aus aller Welt zusammen. Wir nennen euch 10 Gründe für eine Ausstellerteilnahme.

von Leonard J. J. Ure



Wir können die Maker Faire Hannover am 19. und 20. August nicht abwarten! Deshalb haben wir mit einem gestandenen und einem ganz frischen Maker gesprochen, wie sie die Maker Faire Hannover sehen. Berthold und Max vom *ARTandTECH. space e.V.* haben wir gefragt, weshalb sie immer wieder zur Maker Faire Hannover kommen und warum sie als Aussteller ihre Projekte präsentieren. Berthold ist seit der ersten Maker Faire Hannover 2013 als Aussteller dabei. Max ist seit drei Jahren Aussteller. Gemeinsam haben sie schon mehrere Projekte auf der Maker Faire Hannover präsentiert und erlebt.

Make: Was bewegt euch immer wieder zur Maker Faire Hannover zu kommen?

Max: Die Maker Faire Hannover ist für uns letztendlich das Highlight des Jahres. **Berthold:** Auf jeden Fall ist sie das Maker-Highlight des Jahres. Sie ist was Spezielles und mittlerweile ein kleines Stück Familie geworden. Man trifft dort seine und neue Leute. Das andere ist Inspiration. Geh' nicht zur Maker Faire, wenn du Angst vor dem Kreativ-Flash hast!

Make: Habt ihr auf der Maker Faire Hannover etwas gelernt?

Max: Wenn ich hier (im Projektlabor im *ARTandTECH.space*) mit Leuten rede, sind sie bereits am Projekt beteiligt. Auf der Maker Faire Hannover redet man mit Leuten, die nichts mit dem Projekt zu tun haben. Das muss man lernen!

Berthold: Manchmal funktionieren Projekte nicht. Man lernt auf der Maker Faire Hannover, wie man es besser macht. Alle bringen ihre Projekte und man denkt: "Ja geil, das kann man auch so und so machen". Da tauchen Leute mit Lösungen auf, dank denen man später sagt: "Das hab' ich mir auf der Maker Faire Hannover geklaut".



Berthold und Max beim Interview im Projektlabor des ARTandTECH.space e.V.

Make: Ihr kommt zur Maker Faire Hannover bereits vorher vernetzt. Macht es Sinn, ohne ein Netzwerk zu kommen? Also als Maker, der alles alleine macht?

Max: Ich würde es fast noch eher den Einzelkämpfern empfehlen. Du triffst unglaublich viele Leute, die sagen "Krass, das hast du allein auf die Beine gestellt?" Dann kommen auch andere Maker oder man geht zu ihnen, gibt und bekommt Input. Das ist ein super Austausch! Aber auch Makerspaces können sich super präsentieren.

Berthold: Ich würde sagen, es macht mit und ohne Gruppe Sinn! Aber wir wären auch nicht das, was wir als Makerspace sind, wenn's die Maker Faire Hannover nicht gäbe. Wenn wir unsere Projekte mitbringen, geben wir immer was ins System. Natürlich, die Maker Faires funktionieren nicht ohne Maker. Letztendlich nehmen wir aber auch was mit: positive Energie und Inspiration. Deshalb sind wir letztes Jahr mit allen Autos, die wir finden konnten, gefahren. Alle wollten mit!



Süße Zahnbürstenroboter auf der Maker Faire Hannover 2022

Besuche uns auch auf der Intermodellbau

Vom 20. bis 23. April ist in der Messe Dortmund wieder *Intermodellbau*: Auf der nach Angaben des Veranstalters weltgrößten Messe für Modellbau und Modellsport kommen Fans von fliegenden, schwimmenden, fahrenden oder Schienen folgenden Modellen ebenso auf ihre Kosten wie die Freunde von Klemm- und Steckbausteinen. In diesem Jahr kooperiert die Maker Media mit der Intermodellbau. Sichert Euch mit dem Gutscheincode **155263** beim Ticketkauf 2,50 Euro Rabatt (über den Link unten) und besucht uns in Halle 6 am Stand 6.C28!

https://www.intermodellbau.de/ tickets-kaufen

Wenn du dein Projekt bei der Maker Faire Hannover präsentieren willst, dann melde dich jetzt über den folgenden Link bei unserem *Call for Makers*. Ob als privater Maker, Makerspace oder als Unternehmen könnt ihr euch noch bis zum 11. Juni 2023 anmelden. Wir freuen uns auf euch!

maker-faire.de/hannover/call-for-makers

auf der Maker Faire Hannover auszustellen

- 1. Präsentiere dein Projekt der Öffentlichkeit und erhalte direktes Feedback.
- 2. Dein Ausstellungsplatz ist vollkommen kostenlos. Die ganze Crew freut sich auf dich!
- **3.** Tausche dich aus. Begeistere andere von deiner Idee und erweitere dein eigenes Wissen.
- 4. Lass dich von faszinierenden Projekten inspirieren!
- **5.** Du stellst Produkte selber her, die du verkaufen möchtest? Bei uns gibt es die Option auf einen Direktverkauf.
- **6.** Unser Get-Together bietet dir die Möglichkeit, andere Maker kennenzulernen und dich mit ihnen zu vernetzen.
- **7.** Hol dir positive Energie mit nach Hause eine Teilnahme an der Maker Faire ist unkompliziert locker und macht Spaß.
- **8.** Die Maker Faire steht auch für Innovation. Gemeinsam gestalten wir die Zukunft. Teile deine Vision!
- **9.** Zeig der Welt, was du kannst. Wir bieten dir für jedes noch so verrückte Projekt den passenden Raum.
- **10.** Mach mit bei der größten Maker Faire in Deutschland, Österreich und der Schweiz und werde Teil einer weltweiten Community.



Maker Faire Vienna 2023

Nach drei Jahren pandemiebedingter Pause heißt es am 3. und 4. Juni 2023 endlich wieder: "Staunen, Mitmachen, Ausprobieren!" Die Maker Faire in Wien zählt neben gleichnamigen Veranstaltungen in Paris, Hannover und Rom zu den größten Maker Faires in Europa. 2023 werden 900 Selbermacher, Tüftlerinnen, Kreative und Start-Ups als Aussteller sowie 10.000 Besucherinnen und Besucher in der *METAStadt* Wien erwartet. Das 4800m² große ehemalige Fabrikgebäude, in dem Anfang des 20. Jahrhunderts noch Dampfturbinen und Dynamos produziert wurden, wird so wieder zur Kulisse der größten Erfindermesse Österreichs.

Makers for Future

Unter dem Motto "Makers for Future" widmet sich die Maker Faire Vienna 2023 neuen innovativen Ideen und Lösungen für die Gestaltung der Zukunft: Welchen Beitrag können Maker-Projekte zur Bewältigung der großen Herausforderungen unserer Zeit leisten? Ein besonderer Fokus liegt dabei auf dem Thema *Circular Economy* (Kreislaufwirtschaft).

In diesem Rahmen soll der Schwerpunkt, Let's make fashion last!" zeigen, welches Potenzial in der Textilindustrie steckt. Und wie viel hier auch global bewirkt werden kann, wenn bei diesem Thema angesetzt wird. Die gute Nachricht: Es findet bei sehr vielen Menschen ein Umdenken statt. Bei Vorträgen, Workshops und Ständen auf der Maker Faire Vienna präsentieren sich Projekte, die sich mit der nachhaltigen Verwendung von Kleidung beschäftigen dabei geht es um die Organisation von Kleidertausch, es werden neue Kleider aus alten Textilien genäht oder durch Visible Mending repariert und gleichzeitig zum Unikat aufgewertet. Ein zweiter Schwerpunkt ist "Circular ideas for everyone - Impule für Kreislaufwirt-



schaft" rund um Ideen und Strategien für Kreislaufwirtschaft im Alltag. So gibt es zu sehen, wie aus entsorgten Lattenrosten Pflanzenständer werden und wie ein innovativer Hocker zu Hause die Mülltrennung attraktiver macht und Platz spart.

Mitmachen!

Anfassen und Ausprobieren wird auf der Maker Faire großgeschrieben. Es gibt viele



In der Wiener METAStadt: Maker Faire mit altem Industrie-Flair

interessante Mitmachstationen für alle Altersgruppen, aber besonders auch für Kinder, ergänzt um spannende Vorträge und Workshops.

So haben die Teams von Vienna Open Lab und Happylab gemeinsam ein Laborprogramm zum Mitmachen erarbeitet, das die Themenbereiche Wissenschaft und Biohacking vereint. "Biohacking" oder Do-it-yourself-Biologie meint in diesem Fall selbstgebaute Laborgeräte, die Erwachsene wie Kinder beim familienfreundlichen Mitmachprogramm selbst ausprobieren und dabei knifflige wissenschaftliche Fragestellungen rund um das Thema pH-Wert lösen können. Das Technische Museum Wien schickt wiederum seinen rollenden Makerspace roadLAB zur Maker Faire, einen E-Bus, der mit 3D-Drucker, Lasercutter und Co. ausgestattet ist und mit dem das Museum normalerweise Schulen und andere Bildungseinrichtungen besucht. Auf der Maker Faire können Besucherinnen und Besucher dort kleine Projekte mit Lasercutter, Schneideplotter oder 3D-Drucker umsetzen. Auch in den Bereichen Robotik und Stop Motion kann man selber aktiv werden, bei Workshops von Vienna Hobby Lobby und RoboManiac. Außerdem wird man auf der Maker Faire in Wien an diversen Mitmachstationen Insektenhotels bauen, mit Kunststoff experimentieren, einfache Elektromotoren und Hologramme erschaffen und winzige Möbel aus Verpackungsresten basteln können.

Im mobilen Laser-Labyrinth von Jakob Schindler-Scholz, das der Maker während des



Maker Faire heißt mitmachen: Ein Wochenende lang wird gelötet, programmiert, gehämmert, geschraubt, gesägt, 3D-gedruckt und genäht.

ersten Lockdowns für seinen vierjährigen Sohn entwickelt hat, können Kinder und wagemutige Erwachsene ihre Geschicklichkeit testen und sich wie Agenten fühlen. Zudem wird es eine von *City Games Vienna* entworfene Rallye geben: An unterschiedlichen Stationen sollen Kinder und Jugendliche – ganz gemäß dem Motto "Makers for Future" – Einblick in neue Technologien der Zukunft bekommen. Mit einem vollen Stempelpass hat man dann die Chance auf den Hauptgewinn: einen 3D-Drucker.

Tickets im Vorverkauf

Im Online-Shop (siehe Link unten) kann man bis Ostern 2023 zu günstigen Vorverkaufspreisen Tickets erwerben: Tagestickets kosten jetzt 14,90 Euro (später 20 Euro). Ermäßigte Tickets (unter anderem für Kinder von 6 bis 14 Jahren, Schülerinnen und Schüler und alle ab 65 Jahren) gibt es für 9,90 Euro (später 15 Euro). Familientickets (max. 2 Erwachsene und max. 2 Kinder) sind vorab für 29,90 Euro (statt später 40 Euro) erhältlich. —pek

www.makerfairevienna.com/tickets



Visible Mending: Sichtbares Flicken verwandelt beschädigte Kleidung in einzigartige Stücke.



Geschicklichkeit ist Trumpf im mobilen Laser-Labyrinth.





2× Make testen und über 7€ sparen!

Ihre Vorteile:

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

* Für die Laufzeit des Angebotes.

Für nur 19,40 € statt 27 €



Jetzt bestellen: make-magazin.de/miniabo

Paris I

Copyrights by Maker Media Gabit 9 125

Kl auf der Schreibmaschine

Eine angestaubte DDR-Typenradschreibmaschine aus den 1980er Jahren bekommt einen ESP32 eingebaut, der ChatGPT anzapft – so entsteht ein Chatbot der neuesten Generation mit viel Retro-Charme.

von Dirk Wahl



Was ist die *erika 3004?* "Die erika 3004 ist eine elektrische Schreibmaschine, die in der DDR von der Firma *Robotron* hergestellt wurde. Das Modell wurde ab 1986 produziert und war in der DDR sehr verbreitet. Im Vergleich zu älteren Modellen war die erika 3004 mit modernen Funktionen wie einem Korrekturband ausgestattet. Außerdem verfügte sie über eine automatische Zeilenvorschubfunktion und eine automatische Korrekturfunktion, die das Korrigieren von Schreibfehlern erleichterte."

Dieser erste Abschnitt wurde durch *ChatGPT* erstellt und zeigt, wozu dieser Kl-Chatbot prinzipiell in der Lage ist: Konversationen in natürlicher Sprache mit dem Nutzer zu führen. Unter anderem, denn die Software kann auch Arduino-Code liefern, wenn man ihm die entsprechenden Anweisungen gibt und falls die Wünsche nicht zu kompliziert sind.

Viele von Ihnen dürften die Web-Oberfläche von ChatGPT kennen. Aber die Software kann auch über APIs (Programmierschnittstellen) mit anderen Anwendungen kommunizieren. Im Fall meines Projekts schreibt man seine Fragen an ChatGPT auf der erika 3004 und die tippt kurz darauf wie von Geisterhand die Antworten der KI darunter. Die ChatGPT-Plattform bietet eine REST-API, die es Entwicklern ermöglicht, die Funktionen des KI-Chats in eigenen Programmen und Projekten zu nutzen. Einige Beispiele 1 gibt es auf der Webseite von OpenAI, der Firma hinter ChatGPT – und auch hinter meinem Projekt steckt diese REST-API.

Grundsätzlich ist es nicht schwer, ChatGPT auf seinen eigenen Mikrocontroller zu bringen. Im Folgenden fangen wir erst mal auf der Seite von ChatGPT an und entfernen uns dann Schritt für Schritt und Schicht für Schicht von dort, bis wir auf dem ESP32 in meiner Schreibmaschine angekommen sind. Los geht's!

Erst mal anmelden

Um mit ChatGPT kommunizieren zu können, sei es über die Web-Oberfläche (siehe Link in der Kurzinfo) oder per API, ist eine Registrierung bei openai.com notwendig. Hier erkennt man schon den Informationshunger der Firma, denn ohne Eingabe der E-Mail-Adresse und einer Telefonnummer zur Verifizierung der Identität ist die Anmeldung leider nicht möglich. Von der Übersichtsseite aus erreicht man den *Playground* (2) und kann die ersten Versuche mit dem Chatbot unternehmen. Diese ersten Schritte sind hilfreich, um später die API-Anbindung nach eigenen Wünschen zu konfigurieren.

Rechts des Eingabefelds befinden sich Filter und Schieberegler für Parameter, mit denen sich die Berechnungen und die Ausgabe des Chatbots beeinflussen lassen.

Kurzinfo

 » Einstieg in den ChatGPT Playground: Parameter verstehen und setzen
 » ChatGPT-API mit Postman und ESP32 ansprechen
 » I/O-Schnittstelle der erika 3004 über den ESP steuern



Bet started ×	Playground	Load a preset.	× .	Save	View code	Share	***
Inter an instruction or select a preset, nd watch the API respond with a completion that attempts to match the ontext or pattern you provided. ou can control which model completes our request by changing the model. EEP IN MIND Use good judgment when sharing outputs, and attribute them to your name or company. Learn more. Requests submitted to our models may be used to train and improve future models. Learn more. Our default models' training data cuts of in 2021, so they may not have knowledge of current events.	Was ist das Make Das Make Magaz Menschen mit IT Wissenschaft ur an Anleitungen, von kreativen Id innovative Techn auch eine beglei einen Veranstalt	e Magazin? in ist eine monatli teresse an D1Y-Pi di Kunst richtet. Artikeln und Tut een helfen. Inhi ologien und Ma tende Websiz ungskalender ei	che Zeitschrift, die sich an ojekten, Technik, Centrols randomness: Lowering in less random completions. As temperature approaches zero, t model will become deterministi repetitive.	results the c and	Mode F Comple Model text-davinc Temperature Maximum len Stop sequence Enter sequence Top P	i-003 ogth ces and press	0.7 256 Tab
	Looking for	ChatGPT? Try it	now 🖻	×	Frequency pe	enalty	
3	-						

API keys

B

4

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically rotate any API key that we've found has leaked publicly.

SECRET KEY	CREATED	LAST USED	
skLpxb	7. Dez. 2022	14. Feb. 2023	8

+ Create new secret key

Usage

Below you'll find a summary of API usage for your organization. All dates and times are UTC-based, and data may be delayed up to 5 minutes.



Die neueste GPT-Version ist das Modell text-Davinci-003, das auch in meinem Projekt erika 3004 zum Einsatz kommt. Der Parameter Temperature beispielsweise steuert so etwas wie die Kreativität der Antworten, Maximum length gibt die maximale Länge der Antwort vor. Ein Hilfetext erscheint bei Mouseover und auf der Dokumentationsseite gibt es weitere detaillierte Beschreibungen.

Eine Konversation in deutscher Sprache ist möglich, wenn auch ChatGPT hauptsächlich mit englischen Texten gefüttert wurde. Interessant wird es, sobald der Chatbot kreativ werden soll, also komplett neue Zusammenhänge herstellt. Spannende Beispiele für *Prompts* gibt es im Netz etwa bei showgpt.co. Die Antworten sollten jedoch immer auch kritisch hinterfragt werden. So wurde das Modell nur bis Juni 2021 trainiert, und liefert deshalb speziell bei aktuellen Fragen oft nicht das korrekte Ergebnis, obwohl die Antwort meist überzeugend klingt.

API-Key erstellen

Um per REST-API mit ChatGPT kommunizieren zu können, wird ein API-Key zur Authentifizierung benötigt ③. Den erstellt man über das Menü *Personal/ View API Keys* rechts oben im Browser. Den Key sollte man an einer sicheren Stelle speichern, da er sich nicht nochmal komplett anzeigen lässt.

Es lohnt sich auch einen Blick in *Personal/ Manage Account* zu werfen – der Zugang ist nämlich nur für ein paar Monate und nur mit einem begrenzten Budget, das man als "Kredit" bekommt, kostenlos nutzbar, danach muss das Konto nachgeladen werden ④. Für die ersten Projektversuche reicht der Betrag aber allemal.

API-Aufruf mit Postman

Mit Hilfe der kostenlos nutzbaren API-Plattform *Postman* (siehe Links) oder einem ähnlichen Angebot kann man nun eine erste API-Anfrage an den Chatbot schicken **⑤**. Hierfür nutzt man deren Anfragetyp *POST Create completion*, indem man in Postman auf *New* klickt und dann einen neuen *HTTP Request* als *Building Block* wählt. In der Ausklappliste wechselt man dann vom standardmäßigen *GET* zu *POST* und trägt als *request URL* den Inhalt aus dem Kasten **⑥** ein.

Auf dem Karteireiter Authorization wählt man als Typ Bearer Token und trägt den eigenen API-Key ein. Auf dem Reiter Body wählt man als Typ raw und gibt beispielsweise die Anfrage aus Kasten 7 ein.

Hier tauchen nun als completions die vom Playground bekannten Filter und Parameter wieder auf. Gibt man nichts an, nimmt ChatGPT die Default-Werte. Wichtig für die Verarbeitung der Anfrage ist außerdem noch, als Sendeformat JSON auszuwählen (Karteireiter *Body*, Ausklappmenü ganz rechts, Standardeintrag ist *Text*). Ein Klick auf *Send* und nach wenigen Sekunden gibt uns der Chatbot im Panel unten nach "text": das Ergebnis.

Arduino-Code für den ESP32

Der API-Zugriff funktioniert also – und es ist nicht weiter kompliziert, sich mit einem ESP32 mit dem eigenen WLAN zu verbinden und einen HTTPS-Aufruf abzusetzen. Mein Code (gekürzt im Kasten ³, komplett als Download über den Link in der Kurzinfo) hat sich nach einigen Versuchen als recht stabil erwiesen, denn der Run auf ChatGPT ist momentan sehr groß und der Server öfter überlastet, was entsprechend berücksichtigt wird.

Notwendig sind die beiden Libraries WiFi.h und WiFiClientSecure.h, die – falls noch nicht vorhanden – in der Arduino IDE über Sketch/Bibliotheken einbinden/Bibliotheken verwalten installiert werden müssen (eine ausführliche Anleitung dazu gibt es über den Link in der Kurzinfo). Danach kann der ESP32 über den seriellen Monitor Eingaben an ChatGPT

6

https://api.openai.com/v1/engines/text-davinci-003/completions?prompt=<string>&max_tokens=<integer>&temperature=<number>



Q Search Postman 窗 0 🔞 tt https://api.openai.con 😐 8 🖹 Save https://api.openai.com/v1/engines/text-davinci-003/c 0 https://api.openai.com/v1/engines/text-davinci-003/completions?prompt=<string>&max_tokens=<integer>&te POST Body . Pre-request Script Tests Params Authorization • Headers (8) Settinos ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON > none ["prompt":"Was ist besser, Brot oder ein Apfel?", "temperature":0.7, "max_tokens":256] Body Cookies Headers (12) Test Results C 200 OK 4.14 s 977 B Save Re Pretty Raw Preview Visualize JSON 6 Q id": "cmn]-618b7m4wU3Pa1w "object": "text_completion", "created": 1678122453, del": "text-davinci-003", hoices": ["text": "\n\nDas hängt davon ab, w onach du gerade suchst, dann kann ein Apfel eine gute Wahl sein, da er nicht vorbereitet werde aber etwas mehr Sättigung suchst, dann ist Brot die bessere Wahl.", 'index": 0,
'logprobs": null, 10 11 12 finish_reason": "stop S Cookies ⊘ Auto-select agent ⊡ Runner 1 Trash æ 9 0

www.intermodellbau.de

DORTMUND



schicken und die Antworten auch wieder im Monitor ausgeben.

Mit der Antwort von ChatGPT in der Variablen Feedback lassen sich nun wunderbare eigene Projekte erstellen – und damit kommen wir zur Schreibmaschine.

Anschluss der erika 3004 an den ESP32

Für die Umsetzung wurde bei eBay für 20 Euro aus der erika-3004-Familie das Modell Sigma 8300i erstanden. Je nach Zustand findet man sogar noch kostengünstigere Modelle. In der erika 3004 ist eine serielle Schnittstelle verbaut, mit der eine Verbindung zu einem Heimcomputer hergestellt werden konnte, um einen damals teuren PC-Drucker kostengünstig zu ersetzen 9. Die 1200-Baud-Schnittstelle ist bidirektional, und es können sowohl Textzeichen als auch Steuerzeichen (beispielsweise für Fettdruck) ausgetauscht werden. Mit diesen Eigenschaften eignet sich das Interface sehr gut für den Datenfluss zwischen dem ESP32 und der erika 3004

Insgesamt braucht es nur sechs Kabel, um den ESP32 an die Pins A11-13 und B11-13 der erika 3004 anzuschließen **10 11**.

Für die serielle Kommunikation mit der erika 3004 wird die Library SoftwareSerial benötigt. Diese ist für den ESP32 leider nicht in der Arduino IDE enthalten und muss erst als .zip-Datei der Github-Seite von Peter Lerup heruntergeladen und dann in der Arduino IDE mit Sketch/Bibliotheken einbinden/.ZIP-Bibliotheken hinzufügen installiert werden (Download siehe Links in der Kurzinfo). Die beiden lokalen Libraries "ddr2ascii.h" und "ascii2ddr.h" zur Konvertierung der Steuerzeichen zu ASCII gibt es zum Download beim Chaostreff-Potsdam (siehe Links). Diese beiden Dateien kommen zusammen mit dem Arduino-Code ins gleiche Unterverzeichnis, und werden zu Beginn des Programms zusammen mit den RX- und TX-Pins deklariert. Dass es sich um lokale Bibliotheken handelt (die im selben Ordner wie der Sketch stehen), erkennt man im Code daran, dass sie in Anführungszeichen stehen und nicht, wie die üblichen Bibliotheken, in spitzen Klammern 12.

Zum Senden der Tastenanschläge an den ESP32 wird das der Taste entsprechende Byte über TxD geschickt, falls DTD *Low* ist. *High* auf DTD würde das Senden verhindern. Zur Vereinfachung ist der DTD-Pin dauerhaft mit GND verbunden. In der Gegenrichtung, also vom ESP32 an die erika 3004, werden die Leitungen RxD und RTS genutzt. Zuerst wird ein Datenbyte über RxD gesendet und dann der Zustand von RTS abgefragt. Ist RTS *Low*, dann wurde das Zeichen gedruckt und das nächste Zeichen kann gesendet werden. Den {"prompt":"Was ist besser, Brot oder ein Apfel?","temperature":0.7,"max_tokens":256}

kompletten Code für das Erika-Projekt gibt es über den Link in der Kurzinfo zum Download.

Die korrekte Verarbeitung des Strings Feedback von ChatGPT erfordert ein paar Befehle: Beim Senden an ChatGPT werden Umlaute in ae, oe, und ue geändert, um eine reibungslose Kommunikation zu ermöglichen. Und beim Empfang müssen Zeilenumbrüche '\n' an der richtigen Stelle eingefügt werden, um den Typenradwagen nach spätestens 70 Zeichen wieder nach links zu schicken. Sonderzeichen müssen ebenfalls gesondert verarbeitet werden, da der serielle Monitor erst ein Steuerzeichen und dann den ASCII-Code an den ESP32 schickt. Das alles erfordert die vorherige Konvertierung des Strings in ein Char-Array.

Fazit

Da der ESP32 vom Netzteil der erika 3004 versorgt wird, ist äußerlich nicht zu erkennen, dass das Innenleben nicht mehr dem Stand von 1987 entspricht. Der USB-Anschluss (B), normalerweise versteckt hinter einer Revisionsklappe, dient zum Aufspielen von neuen ChatGPT-Modellen. Gerade die Eloquenz von ChatGPT in Kombination mit der Ausgabe auf einer alten Schreibmaschine überrascht die Menschen vor der Tastatur und macht auch richtig Spaß. Wenn man mein Projekt für die nächste Maker Faire nachbauen will, sollte man sicherheitshalber genügend Papier und ein Ersatzfarbband einpacken, denn die Besucher werden die Maschine lieben ... Nach dem selben Prinzip lassen sich aber natürlich noch beliebige andere Maker-Projekte rund um ChatGPT stricken. Wir sind gespannt, was anderen Makern dazu einfällt - schicken Sie Ihre Ideen oder eigenen Projekte gerne an mail@make-magazin.de. —pek

#include <SoftwareSerial.h>
#include "ddr2ascii.h"
#include "ascii2ddr.h"
#define PC_BAUD 9600
#define RTS_PIN 23
//#define DTD_PIN -> GND
#define ERIKA_RX 18
#define ERIKA_TX 19
#define ERIKA_BAUD 1200
SoftwareSerial erika(ERIKA_RX,
ERIKA_TX); // RX, TX



			0	1
	A11-RxD	D19		
	A12-RTS	D23		
3004 5	A13-GND	GND		
4	B11-DTD	GND	ESP32	
2	B12-5V	5V		
1	B13-TxD	D18		





Diesen Nachttisch hat eine KI entworfen

Künstliche Intelligenz kann helfen, einer DIY-Idee die passende Form zu verleihen – wenn man weiß, wie die KIs zu bedienen sind und wo ihre Grenzen liegen. Hier erfahren Sie, wie Sie mithilfe der Bild-KIs Midjourney und Stable Diffusion Entwürfe für Möbel- und Produktdesign generieren.

von Johannes Börnsen



ie Sprach-KI ChatGPT hat in den letzten Wochen für viel Wirbel gesorgt. Neben Antworten auf Fragen nach dem Leben, dem Universum und dem ganzen Rest kann die Künstliche Intelligenz auch Programmcode schreiben oder ihr vorgelegten Code auf Fehler analysieren, beispielsweise für einen Arduino. Durchaus praktisch, wenn eine neue Programmiersprache im Maker-Labor zum Einsatz kommt.

Schon vor der ChatGPT-Welle schwappte die Bild-Generier-KI-Welle durchs Netz. Und ich gestehe: Das Generieren dystopischer Bilder vom letzten Selfie auf der Erde oder Lamas auf dem Mond macht erstaunlich viel Spaß. Nachdem ich meinen Bedarf an KI-generierten Steampunk-Tieren und anderen Kuriositäten gestillt hatte, habe ich versucht, aus den Bild-KIs Midjourney und Stable Diffusion ähnlich nützliche Dinge herauszubekommen, wie die Programmier-Skripte aus ChatGPT, und die Generatoren auf Produkt- und Möbeldesign angesetzt. Tatsächlich lassen sich mit einigen Tricks und Kniffen und der richtigen Herangehensweise durchaus spannende Design-Mockups generieren, die bei so manchem Maker-Projekt eine Hilfe sein können. In diesem Artikel zeige ich am Beispiel eines Nachttisches, wie Sie Bilder-KIs zum Entwerfen für eigene Projekte nutzen können und was einen guten Eingabebefehl ausmacht.

Ich habe für meine Experimente die Bilder-KIs Stable Diffusion und Midjourney verwendet. Stable Diffusion lässt sich kostenlos lokal verwenden, wenn ein entsprechend leistungsstarker PC oder Mac vorhanden ist. Man kann Stable Diffusion aber auch im Browser kostenlos ausprobieren (Link siehe Kurzinfo), ein Account ist nicht notwendig. Geben Sie in das Prompt-Feld einfach auf englisch ein, was auf dem Bild zu sehen sein soll und klicken Sie auf Generate image. Nach etwa einer Minute erhalten sie vier zum Befehl passende Bilder. Eine Anleitung zur lokalen Installation von Stable Diffusion finden Sie ebenfalls in den Links, die Ergebnisse werden dann schneller und in höherer Auflösung errechnet.

Die Bilder-Kl Midjourney lässt sich nicht lokal installieren, sondern wird als Chatbot in Discord genutzt. Das spart Rechenpower und lässt sich sogar am Smartphone nutzen. Um Midjourney verwenden zu können, legen Sie unter discord.com zuerst einen Discord-Account an, laden die App herunter und loggen sich ein. Dann klicken Sie Join the beta auf midjourney.com an, woraufhin sich der Midjourney-Bot in Discord öffnet. Wenn Sie noch kein Abonnement abschließen möchten (ab 10 Euro pro Monat), haben Sie die Möglichkeit, im privaten Chat bis zu zehn Bilder kostenlos zu generieren. Alternativ können Sie sich in einem der Anfänger-Chats aufhalten, wo Sie zwar mit einer unübersichtlichen Menge an generierten Bildern anderer User konfrontiert

Kurzinfo

» KIs installieren und Produkte mit KI-Unterstützung entwerfen » So funktioniert eine Bilder-KI

» Wie das Trainingsmaterial das Ergebnis beeinflusst

Checkliste Zeitaufwand:



ein paar Stunden



Cloud-Lizenz ab 10 Euro

Computer:

bei lokaler Nutzung PC mit leistungsstarker Grafikkarte (mindestens 6GB VRAM, 10GB SSD, 12GB RAM), bei Cloud-Nutzung Discord-Account für Midjourney

Mehr zum Thema





werden, aber dafür unbegrenzt viele Bilder generieren lassen können.

Meiner Erfahrung nach führt mal Midjourney, mal Stable Diffusion zu passenderen Ergebnissen. Beide Generatoren sind englischsprachig und machen gleichermaßen Spaß. Die Beispielbilder für diesen Artikel stammen aus Midjourney.

Schöner befehlen

Sowohl Stable Diffusion als auch Midjourney generieren Bilder anhand eines als Text eingegebenen Befehls und auf Basis der ihr zum Training vorgesetzten Bilder. Dabei sind die generierten Ergebnisse jedoch nicht einfach aus den Trainingsbildern zusammengesetzt,

Midjourney nutzen

Midjourney wird als Chatbot innerhalb der Kommunikations-App Discord verwendet. Der Generier-Auftrag beginnt immer mit /imagine gefolgt vom gewünschten Motiv und zusätzlichen Attributen. Er wird einfach als Chat-Nachricht an den Bot gesendet, der nach einigen Sekunden mit vier Vorschlägen antwortet. Mit den Buttons U1, U2, U3 und U4 lassen sich einzelne Bilder in höherer Auflösung erneut generieren, V1, V2, V3 und V4 generieren Variationen des jeweiligen Bildes.





1 Etwas altbacken, aber mit Nachttischlampe: generiert mit Midjourney und dem Befehl /imagine night stand



2 Schon deutlich schicker: generiert mit Midjourney und dem Befehl / imagine night stand, danish design

sondern werden tatsächlich neu erfunden. Ein von Midjourney oder Stable Diffusion generiertes Bild werden Sie also weder im Netz noch im Möbelkatalog bereits finden. Trotzdem erinnern viele Ergebnisse an die Darstellungen einschlägiger Händler, was einfach an der Herkunft der Trainingsdaten liegt. Dennoch: Wenn man für eine DIY-Idee auf der Suche nach Inspiration ist, liefern die Bild-KIs zumindest Anhaltspunkte für weitere Gedanken zum eigenen Projekt. Je genauer man dabei den Befehl formuliert, desto ähnlicher werden die Ergebnisse pro Generier-Durchgang.

Auch wenn die Befehle theoretisch beliebig lang sein können, macht es Sinn, sich mit kurzen Befehlen zu einem Thema warmzugenerieren und zu testen, was Midjourney beispielsweise zum Thema Nachttisch aus dem Befehl /imagine night stand ohne weitere Attribute generiert 1. Im Falle von Midjourney sind das recht klassische Möbel, wie sie amerikanische Holzwerker vor einigen Jahrzehnten gebaut haben. Optisch eher schwer, vermutlich aus Mahagoni und für den heutigen, eher vom skandinavischen Design geprägten Geschmack, altbacken verziert. Die passende Lampe oder einen Wecker gibts direkt dazu.

Midjourney kann aber auch anders: Wenn man den Befehl um ein definierendes Attribut ergänzt (/imagine night stand, danish design),



6 Generiert mit Midjourney und dem Befehl /imagine night stand, nordic design, two drawers without handles, white oak



Generiert mit Midjourney und dem Befehl /imagine night stand, nordic design, two drawers without handles, white oak, hairpin legs
wandelt sich das generierte Bild und geht eher in Richtung dessen, was große Möbelhausketten aktuell im Programm haben 2. Man kann aber auch noch spezifischer werden. /imagine night stand, nordic design, two drawers without handles, white oak liefert zwar auch noch Schubladen mit Griffen, aber links oben auch einen sehr gut zum Befehl passenden Nachttisch 3. Fällt Ihnen auf, dass alle Entwürfe sehr ähnliche Beine haben, obwohl diese im Befehl gar nicht definiert sind?

Intelligent ja, kreativ nein

An dieser Stelle kommen wir auch bereits zum Problem der KI-Generatoren. Nicht ohne Grund werden diese Systeme Künstliche Intelligenz und nicht Künstliche Kreativität genannt. Das Erfinden genau passender Lösungen, zugeschnitten auf einen bestimmten Anwendungsfall, ist diesen Systemen allenfalls durch glücklichen Zufall möglich. Möchte man die hölzernen Beine durch Metallbeine ersetzen, muss man dies im Befehl definieren, beispielsweise indem der Befehl mit *Hairpin Legs* ergänzt wird 4. Die kreative Leistung, nämlich das Auswählen einer bestimmten Beinform, passiert im Befehl und liegt damit nach wie vor beim Anwender, andernfalls entscheidet der Zufall - beziehungsweise die Trainingsbilder.

Schubladendenken

Stable Diffusion und Midjourney basieren auf sogenannten Diffusionsmodellen. Beim Training werden Bilder schrittweise verfremdet, bis nur noch Rauschen übrig bleibt. Die KI lernt diesen Prozess umzukehren, um aus einem Rauschen ein Bild zu generieren.

Als ob das nicht schwer genug wäre, soll sie zusätzlich dafür sorgen, dass das Bild auch noch zur Bildbeschreibung passt. Die KI geht dabei so vor, dass sie schrittweise den Anteil vom Rauschen entfernt, der nicht wie die zum Befehl passenden Trainingsbilder aussieht. Das Ergebnis ist also vom Eingangsrauschen und den beim Training vorgelegten Bildern abhängig.

Bei dem gezeigten ersten Nachttisch-Beispiel versteht die KI nicht, dass auf den generierten Bildern auch eine Lampe auf dem Nachttisch steht. Für sie ist diese ein Teil des Ergebnisses, weil diese offenbar auch in den Trainingsdaten häufig gemeinsam abgebildet sind. Fordern wir im Befehl einen Nachttisch ohne Lampe, verändert die KI das Rauschen so, dass es wie Trainingsbilder mit der Verschlagwortung Nachttisch ohne Lampe aussieht, statt auf Bildern mit Lampen nachzuschauen, wie eine Lampe aussieht und diese gezielt wegzulassen.

Das Gleiche gilt auch für die Schublade des Nachttisches: Die KI hat keine Ahnung, wie eine Schublade funktioniert. Sie weiß nicht, dass sie sich aus dem Korpus herausziehen



6 Achten Sie auf die Maßlinien und den Text. Alle Klarheiten beseitig? Generiert mit Midjourney und dem Befehl /imagine detailed technical drawing for a modern wooden night stand

lassen muss. Sie weiß lediglich, wie Bildern von Schubladen aussehen und entfernt alles Rauschen, was nicht nach Schublade aussieht, ohne diese einer Prüfung auf Funktionalität und Praktikabilität zu unterziehen. Diesen Schritt müssen wir als Anwender machen, indem wir den Eingabesatz anpassen.

Daraus folgt auch, dass eine KI kein Verständnis von Perspektive und Maßstabseinheiten hat. Wollen wir den Nachttisch nachbauen, müssen wir den passenden Bauplan anhand der KI-Grafik also selber zeichnen. Zwar lassen sich mit passendem Befehl durchaus Grafiken generieren, die auf den ersten Blick nach einem Bauplan aussehen, auf den zweiten Blick wird hier die fehlende Logikprüfung aber überdeutlich **S**.

Auch das Generieren einer weiteren Ansicht vom selben Objekt ist nahezu unmöglich, da bei jedem Generiervorgang ein neues Ergebnis – und damit ein anderer Nachttisch – erzeugt wird und dem im ersten Durchgang erzeugten Nachttisch kein 3D-Modell zugrunde liegt, aus dem die zweite Ansicht generiert werden könnte.

Während die erzeugten Bilder als eine Art visuelle Recherche durchaus hilfreich sein können, ist die rechtliche Komponente der Ergebnisse noch diffus. Mehrere Künstlerinnen, die ihre Werke auf der Plattform *DeviantArt* gepostet haben, sehen in den Ergebnissen eine Urheberrechtsverletzung, weil ihre Werke ohne Zustimmung mutmaßlich zum Trainieren der Kls verwendet wurden (einen Hintergrundartikel dazu finden Sie über die Links in der Kurzinfo). Die Macher der Generatoren verweisen auf die in den USA gültige Fair-Use-Klausel, die es ihrer Meinung nach erlaubt, die Systeme mit urheberrechtlich geschützten Bildern zu trainieren, entschieden ist bisher nichts.

Mein persönliches Fazit

Dass KI-Systeme die Kreativbranche revolutionieren werden, steht wohl außer Frage. Dem oft zu hörenden Abgesang auf kreative Berufe und künstlerisches Arbeiten mag ich mich aber nicht anschließen. So viel Spaß es macht, mit Bildgeneratoren Grafiken und Produktdesign zu generieren, so wenig versteht die KI jedoch, was sie da macht. Ohne Anwender, der die KI in die richtige Richtung lenkt und ihre Ergebnisse weiterverarbeitet, ist sie reichlich nutzlos. Statt derlei Systeme zu verteufeln und zu versuchen, sie zu verbieten, ist es aus meiner Sicht besser, sich mit der Funktionsweise vertraut zu machen und sie als weiteres Werkzeug zu verstehen. Ich werde jetzt jedenfalls den Nachttischvorschlag aus dem Artikeltitelbild bauen, er gefällt mir nämlich tatsächlich ziemlich gut. Ein Video davon finden Sie auf unserem YouTube-Kanal. —iom

Laser-Scanning-Mikroskop aus altem Blu-Ray-Player

Optische Laufwerke sind so konstruiert, dass sie die winzigen Strukturen auf der Disc erkennen und lesen können, um Musik oder Filme abzuspielen. Mit einem Lasermodul aus einem alten Blu-Ray-Player, einer Handvoll elektronischer Bauteile und etwas Software lässt sich daher für wenig Geld ein Laserscan-Mikroskop bauen.

von Michael Linsenmeier



Das Herzstück eines optischen Laufwerks ist die Optical Pickup Unit (OPU). Sie fokussiert einen Laserstrahl (405nm, violett) mithilfe von Linsen auf die Oberfläche der Blu-ray-Disc. Das von den Pits und Lands reflektierte Licht wird über eine Laseroptik mit Spiegeln und Linsen auf einen Sensorchip, den Photo Diode Integrated Circuit, kurz PDIC, fokussiert. Dessen Ausgangssignale werden zur Fokussierung, zur Positionierung des Lasers auf der Spur und natürlich zur Gewinnung der Dateninhalte verwendet. Die OPU stellt diese Daten als analoge Signale zur Verfügung, sodass das Mikroskop sogar Graustufen darstellen kann.

Im Prinzip kann man jede OPU für dieses Projekt verwenden, man bekommt sie für wenig Geld z.B. bei *Ebay* oder *Aliexpress*. Ich habe eine OPU vom Typ *BDP-10G* aus einem defekten *Samsung* Blu-Ray-Player wiederverwertet. Natürlich geht das auch mit anderen OPUs, aber dann muss man die Pinbelegung herausfinden und gegebenenfalls die Elektronik anpassen. Besondere Vorsicht ist bei OPUs aus Blu-Ray-Brennern geboten, deren Laser mit deutlich höherer Leistung arbeiten und für die Augen gefährlich werden können. Generell sollte aber bei der Arbeit mit Lasern immer eine geeignete Schutzbrille getragen werden.

Und so funktioniert das Mikroskop: Die Linse der OPU ist von kleinen Spulen umgeben und zwischen Magneten beweglich aufgehängt. Dadurch kann die Probe in Querrichtung (Zeilen) abgetastet werden. Die Probe liegt dabei auf einem Schlitten, der von einem Schrittmotor mit angeflanschter M3-Gewindestange in Längsrichtung unter der OPU hindurchgezogen wird. Ein Mikrocontroller (ESP32) mit nachgeschalteter Elektronik steuert den Schrittmotor und die Spulen an. Die Elektronik verstärkt weiterhin die Ausgangssignale des Lichtsensors und liefert das Summensignal, das der Intensität des reflektierten Lichts entspricht. Dieses wird vom ESP32 digitalisiert und zur Graustufendarstellung des Bildes verwendet.

Mit dem derzeitigen Aufbau können Strukturen von etwa 1µm aufgelöst werden. Theoretisch sind 600 Nanometer möglich, was dem Durchmesser des Lasers im Brennpunkt entspricht. Da das ESP32 WLAN-fähig ist, kann das Mikroskop mit einem Webbrowser über das WLAN bedient werden. Die Software ist in *HTML* und JavaScript geschrieben, sodass außer einem Webbrowser keine weiteren Programme benötigt werden.

Videos, die das Mikroskop in Betrieb zeigen, sind auf meinem YouTube-Kanal @DoctorVolt zu finden. Der Quellcode, die Schaltpläne und alles, was man zum Nachbau braucht, sind unter dem untenstehenden Link zu finden. —caw

github.com/michalin/Blu-ray-Laser-Scanning-Microscope



Draufsicht einer OPU mit eingezeichneten Strahlengängen für Infrarot/Rot und Violett



Unterseite einer OPU. Man erkennt das Linsensystem mit den umgebenden Spulen für die Abtastung der Zeilen und zur Fokussierung.



Sie haben sich sicher schon über das Titelbild gewundert. Hier sehen Sie eine Aufnahme von Zellen einer Zwiebelhaut, die einzelnen Zellen sind etwa 200 Mikrometer groß.

WLED mittels Herzfrequenz steuern

Connected Fitness ermöglicht es modernen Ergometern, Trainingsdaten per Bluetooth zu übertragen. Das Rudergerät verbessert die eigene Fitness, die angeschlossenen WLEDs sorgen für das passende Ambiente. Spaß am Gerät: So verbindet sich Sport mit Technik.

von Ulf Mayer



S chon lange wollte ich mich mit den beliebten einzeln ansteuerbaren RGB-LEDs beschäftigen. Als mein Rudergerät nach einem Update in der dazugehörigen App einen *Bluetooth Broadcast* erhielt, war meine Neugier geweckt: Wäre es möglich, die Fitnessdaten zu nutzen, um diese smarten LEDs zu steuern? Das war der Anstoß, mich intensiv mit Bluetooth und *WLED* (siehe auch S. 8) zu beschäftigen, was eine super Alternative zu kommerziellen Systemen ist.

Da ich bisher noch keine Berührungspunkte mit Bluetooth hatte, nutzte ich dieses Projekt, um mehr über den *Fitness Machine Service* (FTMS) zu erfahren. Dieser Standard ermöglicht das Auslesen von Daten wie Herzfrequenz, Schlagfrequenz, Leistung in Watt und in diesem Fall wichtig: die Anzahl der Züge am Rudergerät.

Mit den üblichen Bluetooth-Paketen für NodeJS kam ich nicht weiter, fand aber schließlich mit @abandonware/noble ein NPM-Paket, das auf Anhieb funktionierte. Hier benutze ich die Eigenschaft 0x2AD1 - Rower Data, um die erhaltenen Daten an die WLED Rest API zu senden. Der WLED Controller ist mit dem lokalen Netzwerk verbunden. Leider war dies über den mDNS Service nicht schnell genug, aber eine direkte Adressierung der IP-Adresse funktionierte mit wenigen Millisekunden Latenz.

Rudern ist ein effektives Ganzkörpertraining, Ausdauer- und Krafttraining in einem. Es ist die eierlegende Wollmilchsau der persönlichen Fitness, um den eigenen Körper möglichst lange und schmerzfrei bis ins hohe Alter nutzen zu können. Ich rudere häufig *Steady State* mit einer Schlagzahl von 20 bis 22 SPM (**S**trokes **p**er **M**inute). Da die Daten ständig per Bluetooth übertragen werden, ist es sinnvoll, die WLED-API nicht mit Daten zu überfluten. Ich übermittle die Daten einfach bei jedem Zug am Rudergerät an die API.

Als Controller verwende ich den preiswerten ESP8266, mit dem ich 195 LED-Pixel an einem Netzteil mit 5V und 20A ansteuere. Mit Jumpern, Draht und etwas Löten habe ich die Bauteile verbunden. Ein spezielles doppelseitiges Klebeband wird an der Rudermaschine verwendet, um die LEDs zu fixieren. Das lässt sich rückstandslos wieder entfernen, allerdings verlieren die LEDs dadurch nach einigen Wochen ihren Halt. Eine stabilere Alternative ohne Bohren habe ich bisher noch nicht gefunden.

Meine Herzfrequenz messe ich mit einem normalen Brustgurt, auch einige Sportuhren unterstützen die Übertragung. Die WLED-Pixel werden analog zu den fünf Herzfrequenzzonen in verschiedenen Farben angesteuert, alternativ auf Basis der erzeugten Leistung. Einen Extra-Modus mit Durchlauf der LEDs habe ich ebenfalls umgesetzt. —*caw*

 get.harder-better-faster-stronger.de/ glow-row



Testaufbau während der Entwicklung



Funktionsdiagramm des Systems



Atomic-Chess

Dieser Schachtisch bewegt seine Figuren wie von Geisterhand, kann sich mit anderen Spielern über das Internet verbinden und besondere Partien wieder zum Leben erwecken.

von Marcel Ochsendorf



it seinem transparenten Spielfeld, dem Aluminium-Rahmen und der futuristischen LED-Beleuchtung wirkt *Atomic-Chess* wie eine Fusion aus Schachbrett und 3D-Drucker. Dass diese Verbindung kein Zufall ist, wird schlagartig klar, sobald der Tisch damit beginnt, gespielte Züge zu erfassen und gegnerische Figuren zu bewegen – stellvertretend für einen Gegner, der sich mit einem identischen Tisch an einem anderen Ort befindet oder nur virtuell existiert.

Mechanik und Elektronik sind vom 3D-Drucker inspiriert: Der Rahmen des Schachtischs besteht im Wesentlichen aus Aluminiumprofilen und 3D-gedruckten Komponenten. Als Spielfeld dient eine 5mm dicke und 62cm × 62cm große Acrylglas-Platte, auf deren Unterseite mithilfe eines Lasercutters ein Schachbrettmuster graviert wurde.

Unter dem Spielfeld bewegt sich ein Schlitten entlang der X- und Y-Achse (CoreXY). Auf ihm ist ein Elektromagnet befestigt, der die Schachfiguren über das Spielfeld ziehen kann. Dazu wurde in jede Figur ein Magnet eingelassen. Da der Bewegungsraum des Schlittens größer als das Schachfeld ist, werden geschlagene Figuren an den Rand und bei einem Neustart zurück auf das Spielfeld bewegt. Damit Atomic-Chess weiß, ob die Figuren vollzählig sind und um die eigenen Spielzüge zu digitalisieren, nutzt das Brett einen NFC-Reader, der ebenfalls auf dem Schlitten sitzt und den NFC-Tag der jeweiligen Spielfigur ausliest.

Schrittmotoren, Endschalter und der Elektromagnet steuert ein Marlin-2-kompatibles Board mithilfe von G-Code, der auch von 3D-Druckern verwendet wird. Diesen berechnet ein Raspberry Pi, der zudem auch als Schnittstelle zu dem zentralen Spielserver dient. Die angepasste Distribution des Atomic-Chess wurde mit dem Buildroot-Framework erstellt. Sie kann über ein Touch-Display an der Front des Schachbretts bedient werden.

Um Schachspiele zwischen zwei Schachtischen oder einem Schachtisch und einem Web-Spiel-Client zu realisieren, übernimmt ein zentraler Spielserver die Synchronisation von Spielesitzungen. Wenn sich kein menschlicher Mitspieler findet, wird automatisch ein Computer-Spieler erstellt. Weiterhin ist es über die Plattform möglich, andere Spiele und deren Züge live zu beobachten. Wer ältere Spielverläufe noch einmal erleben will, kann auf gespeicherte Partien zurückgreifen.

Auf der Website und im GitHub-Repository des Projekts gibt es noch mehr Informationen und auch die wissenschaftliche Arbeit, in deren Rahmen Atomic-Chess entstanden ist. Wer den Tisch nachbauen möchte, findet dort ebenfalls alle benötigten Dateien. —*akf*

make-magazin.de/x8j4



Den Rahmen bildet eine Konstruktion aus Aluminiumprofilen. Im Inneren bewegt ein CoreXY-System den Schlitten unter dem Spielfeld.



Damit der Schlitten genug Platz hat, sitzt die Elektronik am Rand.



Rasensprenger mit Spielecontroller steuern

Dieser Rasensprenger lässt sich kinderleicht bedienen und programmieren. Mit dem Controller einer Playstation 3 steuert man den 360-Grad-Bewässerungswinkel, die Höhe und den Wasserzulauf. Außerdem kann sich der ESP32, der den Rasensprenger steuert, ein Bewässerungsmuster merken und beliebig oft abspielen.

von Jens Hackel



er in seinem Garten keine fest installierte Beregnungsanlage hat, kennt die lästigen Tätigkeiten zu Beginn oder am Ende eines heißen Tages sicherlich auch: Gartensprenger aufstellen, Beregnungsweite und Wasserdruck einstellen, eine Zeit lang laufen lassen, dabei aufpassen, dass Gartenmöbel, Sonnenschirme oder das Klettergerüst nicht nass werden, ausschalten, Standort wechseln und je nach Größe des Gartens beliebig wiederholen. Wäre es nicht prima, den Gartensprenger nur einmal aufstellen zu müssen und den Beregnungsvorgang beguem aus dem Liegestuhl zu steuern? Oder auch komplett automatisch ablaufen zu lassen? Dafür habe ich einen fernsteuerbaren Gartensprenger konstruiert, den man programmieren kann. Wie man ihn nachbaut, zeige ich euch in diesem Artikel.

Das Herzstück des Gartensprengers ist ein ESP32-WROOM-32 NodeMCU-Mikrocontroller. Dieser steuert mittels dreier Modellbau-Servos die horizontale und vertikale Wasserstrahlrichtung sowie den Wasserdruck. Bedient und programmiert wird der Rasensprenger mit einem PlayStation 3-Controller, der per Bluetooth-Low-Energy (BLE) verbunden wird. Mit diesem kann man ihn komplett manuell betreiben, z.B. um die Nachbarskatze zu verscheuchen, oder einen Beregnungsbereich von bis zu 360 Grad in 2-Grad-Schritten programmieren.

Bauteile drucken

Alle benötigten 3D-Druckteile lassen sich auf einer Druckfläche von 20cm × 20cm ausdrucken. Sie stehen im Github-Repository des Projekts zum Download bereit (siehe Link in der Kurzinfo). Bewährt haben sich bei mir die Einstellungen 0,2mm Schichtdicke und 23% Infill für PLA als Druckwerkstoff. Lediglich die wasserführenden Teile (L-Adapter und Sprühkopf) müssen mit 100% Infill und ohne Stützmaterial gedruckt werden, damit nachher im Betrieb nichts undicht wird.

Die Elektronik

Während der 3D-Drucker beschäftigt ist, sollte parallel an der Elektronik gearbeitet werden, da diese vor dem Zusammenbau der Mechanik benötigt wird, um die Servos in die jeweils erforderliche Position für die Montage zu bringen.

Für die elektronischen Komponenten und die Verkabelung verwende ich eine 90mm × 50mm große Lochrasterplatine, die sich in einer Verteilerdose befinden wird 1. Damit sich die Platine leichter in der Verteilerdose befestigen lässt, habe ich die Ecken leicht abgeschrägt und M3-Nylon-Gewindestücke in die Verteilerdose geklebt.

Der ESP32 steckt mithilfe von Stiftleisten abnehmbar auf der Platine und ist so ausge-

Kurzinfo

- » Einen programmierbaren Rasensprenger bauen
- » SPIFFS-Dateisystem verwenden, um Konfigurationen auf dem ESP32 zu speichern
- » Servos mit einem PS3-Controller steuern

Checkliste

Zeitaufwand: 1 bis 2 Wochenenden Kosten:



etwa 90 – 100 Euro (ohne PS3-Controller)





Grundkenntnisse, Bestücken, Kabel- und Stecker konfektionieren

3D-Druck: Drucker mit 20cm × 20cm Druckfläche 3 D

Mehr zum Thema

- » Guido Burger und Klaus Uwe Gollmer, Der Solartisch und die grüne Steckdose, Make 4/22, S. 10
- » Ralf Stoffels, Der Rasenkabelfinder, Make 2/22, S. 26
- » Frank Siebert, Arduino züchtet Salat, Make 2/21, S. 32



Material

- » 3D-Druck-Bauteile
- 3 Servomotoren MG996R
- 180-Grad-Metallgetriebe
- Regulierstop-Ventil von Starnearby PlayStation-3-Controller
- ESP32-WROOM-32 NodeMCU
- DC-DC-Step-Down-Converter auf 3,3V
- Keramik-Kondensator 100nF, 10V
- Elektrolyt-Kondensator 1000µF, 10V
- **Batteriehalter und Batterien AA**
- oder 18650-Akku **Euro-Lochrasterplatine**
- Punktraster Rastermaß 2,54
- Abzweigdose 108mm × 108mm
- Silikonschlauch ca. 20cm lang, 7mm Innendurchmesser, 11mm Außendurchmesser
- Knickschutzfeder 8cm lang, 12mm Innendurchmesser
- Kugellager 625 ZZ
- Kleinmaterial M3-, M4- und M5-Schrauben samt Muttern, M4-Messing-Gewindeeinsätze, M8-Mutter und Gewindestange (optional), Schalter, Schlauchschelle, Kabelbinder, Gewebeschläuche

.....

Werkzeug

- **Raspberry Pi**
- mit Tastatur und installiertem RetroPie Lötkolben
- Schraubendreher
- Cutter



In der Abzweigdose ist die Elektronik vor Wasser geschützt.



richtet, dass sein USB-Port problemlos erreichbar ist, um den Mikrocontroller zu programmieren und die serielle Ausgabe im Testbetrieb zu nutzen. Dafür muss man nur den verdeckenden Blindstopfen an der Verteilerdose entfernen. Zwischen der Lochraster-Platine und dem ESP-Board ist ein *3,3V-Step-Down*- Spannungswandler zur Stromversorgung untergebracht, die Glättungskondensatoren links daneben 2.

Als Ein-/Ausschalter habe ich eine runde Variante gewählt, die perfekt in ein 20mm großes Loch in der Verteilerdose passt. Zur Stromversorgung habe ich für den Testbetrieb

U Update Retroit 201 contro 202 custor 207 ganeco 212 mkarca 217 powert 218 ps3cor 229 sixaxi 231 snesde 233 steano 237 xarcad 238 xboxdr 239 xpado	e all installed driver packages Pie - supplementary olblock mhidsony (Installed - via source) ondriver adejoystick block ntroller is (Installed - via source) ev controller de2jstick rv eo (Installed - via source)
--	--

3 Um den Controller koppeln zu können, muss zuerst der sixaxis-Treiber installiert werden.

eine Batteriehalterung mit 3 × 1,5V-AA-Zellen eingeklebt. Für die alltägliche Nutzung sollte hier sinnvollerweise entweder eine Variante mit höherer Kapazität oder eine externe Spannungsquelle verwendet werden, da speziell die Servos, je nach Belastung, einen deutlichen Strombedarf haben.

Die Anschlussleitungen der Servos habe ich gegen leichte Beschädigungen mit Gewebeschläuchen geschützt. Sie werden alle durch eine Schraubdurchführung in die Verteilerdose geführt. Eine Abdichtung mit Silikon oder Knetkleber schützt zusätzlich vor Feuchtigkeit.

Die Bluetooth-Adresse des Controllers

Um den PS3-Controller mit dem ESP32 verbinden zu können, muss man die *Bluetooth-MAC*-Adresse des Controllers ermitteln. Ich habe dafür einen Raspberry Pi mit der Distribution *RetroPie* verwendet, da diese über ein entsprechendes Plug-in verfügt.

Als Erstes verbindet man den PS3-Controller mit dem Raspberry Pi über ein USB-Kabel, und schaltet ihn ein. Sobald RetroPie gestartet ist, drückt man die A-Taste (Kreis auf dem PS3-Controller), um ins Hauptmenü zu kommen. Dort wählt man den Menüpunkt *RETROPIE SETUP* aus. In diesem navigiert man zum Menüpunkt *P: Manage packages* und wählt anschließend *driver* aus, was zu einer Liste der verfügbaren Treiber führt. Hinter der Nummer 229 verbirgt sich der *sixaxis*-Treiber (3), der zunächst installiert werden muss. Das geht mit dem Befehl *S: Install from source*.

Ist die Installation fertig, navigiert man wieder zurück ins RetroPie-Menü und geht dort auf Bluetooth. In den Bluetooth-Einstellungen wählt man danach P: Pair and Connect to Bluetooth Device aus. RetroPie gibt anschließend den Hinweis, dass man den Controller vom USB-Kabel trennen und die PlayStation-Taste gedrückt halten soll, bis der Controller blinkt, um ihn danach wieder anzuschließen. In der Liste, die nach ein paar Sekunden auftaucht, sollte der Controller jetzt erscheinen 4. Zum Verbinden wählt man ihn aus, trennt das USB-Kabel, hält die PlayStation-Taste ein erneutes Mal gedrückt, bis die vier LEDs am Controller blinken und geht anschließend im Bluetooth-Menü mithilfe der Tastatur auf den Punkt C: Connect to all paired devices. Bei erfolgreicher Verbindung leuchtet LED-1 am PS3-Controller dauerhaft. Danach kann man wieder in das RetroPie-Menü zurückwechseln.

Um schließlich die benötigte Bluetooth-Adresse auszulesen, drückt man abschließend die F4-Taste auf der Tastatur, gibt heiconfig im Terminal ein und bestätigt mit der Eingabetaste. Nun werden verschiedene Parameter aufgelistet 5. Unter BD-Address steht diejenige, die wir zum Verbinden mit dem Rasensprenger benötigen. Diese muss man sich für später notieren. RetroPie kann anschließend mit sudo shutdown heruntergefahren werden.

Firmware-Upload

Die Firmware wird mit der Arduino IDE (1.8.19) auf den ESP32 geladen. Ein Download-Link und die Anleitung, wie man einen ESP32 damit in Betrieb nimmt, befinden sich in der Kurzinfo. Dieses Projekt benötigt den Board-Treiber esp32 (1.0.4). Außerdem muss man die Bibliotheken PS3 Controller Host (1.0.0) und ESP32-Servo (0.12.1) über den Bibliotheksverwalter installieren. Es ist wichtig, genau die angegebenen Versionsnummern zu verwenden, damit die Verbindung mit dem Controller funktioniert. Zum Abspeichern der programmierten Positionsdaten nutzt der Rasensprenger drei Dateien, auf die er mithilfe von SPIFFS (SPI Flash File System) zugreifen kann. Um diese im Vorfeld auf den ESP32 laden zu können, muss zunächst das Plug-in ESP32FS zur Arduino IDE hinzugefügt werden (siehe Link in der Kurzinfo). Nachdem das Plug-in heruntergeladen ist, erstellt man im Arduino-Verzeichnis neben libraries den Ordner tools. falls dieser noch nicht existiert. Danach entpackt man das heruntergeladene Plug-in in folgenden Unterordner:

~/Arduino/tools/ESP32FS/tool/esp32fs. jar

Zum Abschluss muss man die Arduino IDE neu starten.

Als Nächstes öffnet man die Datei *Sprinkler_Firmware_V0-16.ino*, trägt die zuvor ermittelte MAC-Adresse des PS3-Controllers an der Stelle PS3.begin() in die Klammer ein und lädt die Firmware auf den ESP32. Danach muss man nur noch die drei Dateien, die sich im Sketch-Unterordner *data* befinden, auf den ESP32 übertragen. Das funktioniert durch einen Klick auf den Menüpunkt *ESP32 Sketch Data Upload*, den das Plug-in im Menü *Werkzeuge* hinzugefügt hat **6**.

Mechanik

Wenn die Programmierung des ESP32 erfolgreich war, und auch das Pairing des PS3-Controllers funktioniert hat, sollten nun die Servomotoren an den Mikrocontroller angeschlossen und ein Reset am ESP32 ausgelöst werden. Die Servos drehen sich daraufhin jeweils in die für die Montage erforderliche Position. Um ein Vertauschen beim Einbau zu vermeiden, sollte man die Servos markieren oder mit einem Aufkleber versehen. Anschließend kann der mechanische Zusammenbau beginnen.

Bodenplatte und Ventilhalter

In die Unterseite der Bodenplatte **A** werden zunächst zwei M3-Muttern in die dafür vor-

icuse cho	
	OC:FC:83:2E:8D:7A Sony PLAYSTATION(R)3 Controller
	Cance1>

4 Haben wir die Anweisungen richtig befolgt (und waren schnell genug), erscheint der PS3-Controller in der Liste der verfügbaren Bluetooth-Geräte.



Über den Befehl hciconfig erhalten wir die MAC-Adresse des PS3-Controllers.



🜀 Das ESP32FS-Plug-in ergänzt die Arduino IDE um die Option, Dateien auf den ESP32 zu übertragen.



Mechanischer Aufbau: Bodenplatte und Ventilhalter

gesehenen Löcher gepresst und anschließend auf der Oberseite eine M8-Mutter eingelegt 7. Hier kann man später eine M8-Gewindestange als Erdsporn einschrauben oder eine Schraube M8 × 20mm und den 3D-gedruckten Erdsporn nutzen 8. Auf die Bodenplatte wird nun mittig der Ventil-Adapterring **B** gelegt und darüber der Ventilhalter C. Die Teile A, B und C werden nun mit zwei Senkkopfschrauben M3 × 25mm verbunden. Danach wird das Ventil in den Ventilhalter eingelegt. Hier muss gegebenenfalls etwas nachgefeilt werden, damit das Ventil ohne Kraftaufwand hineinpasst. Wichtig: Das Ventil muss sich für die weiteren Montageschritte im geschlossen Zustand befinden.

Ventil-Servo und Seitenteile

In die sechs Löcher des Servohalters D sind nun M4-Messing-Gewindeeinsätze einzuschmelzen. Das gelingt sehr gut mit einer kurzen M4-Zylinderschraube und einer Lötstation. Danach schraubt man den Servo (in Grundstellung) mit vier M4 × 8mm-Schrauben am Servohalter **D** fest 9. Nun wird der scheibenförmige Servoadapter mit zwei Schrauben an der Rückseite der Stellzange Da befestigt. Die Stellzange wird dann so auf die Servoachse gesteckt, dass sie bei der Montage genau auf das Ventil passt, ohne den Servo zu verdrehen. Im Anschluss befestigt man den bestückten Servohalter D mit zwei M4-Schrauben am Ventilhalter C. In den Elektronikbox-Halter E und das Seitenteil F werden als nächstes M4-Gewindeeinsätze geschmolzen und beide Teile anschließend in die Bodenplatte A gesteckt.

Getriebe und Elektronik-Box

Zunächst müssen M4-Gewindeeinsätze in die Bauteile Getriebehalter **H** und in den zweiten Elektronikbox-Halter **E** sowie das Zentralrad **J** eingeschmolzen werden. Das Zentralrad **J** wird nun auf den Wellenzapfen am Getriebehalter gesteckt ¹² und mit dem Federhalter **K** arretiert. Die Drehung des Zentralrads sollte ohne großen Kraftaufwand möglich sein, eventuell ist etwas Nacharbeit mit einer Feile oder mit Schleifpapier erforderlich.

Im Anschluss wird der 7mm hohe Servo-Spacer **G** auf den Servo gesteckt und beides mit M4-Schrauben am Getriebehalter **H** festgeschraubt. Der scheibenförmige Servoadapter wird am Servozahnrad I mit zwei kleinen Schrauben befestigt und anschließend auf die Servowelle gesteckt. Dabei müssen sich das Servozahnrad und das Zentralrad in Grundposition (B) zueinander befinden.

Der zweite Elektronikbox-Halter **E** wird nun von unten in den Getriebehalter **H** gesteckt und mit zwei M4-Schrauben daran festgeschraubt. Ebenfalls mit zwei M4-Schrauben wird nun der Getriebehalter **H** am Seitenteil **F** befestigt und abschließend schraubt man den Elektronikbox-Halter **E** und das Seitenteil **F** mit vier M4-Befestigungsschrauben an der Bodenplatte **A** fest.

Jetzt kann die Verteilerdose an den beiden Elektronikbox-Haltern **E** mit passenden Schrauben befestigt werden. Je nach verwendetem Dosen-Typ ist hier beim Bohren der Befestigungslöcher eine individuelle Anpassung erforderlich. Die Bodenplatte **A** und der Getriebehalter **H** sollten nach der Montage der Verteilerdose parallel ausgerichtet sein.

Sprinklerarm und Sprühkopf

Auf dem Servo, der die Höhe (Elevation) steuert, wird zunächst der 5,5mm hohe Servo-Spacer **O** aufgesteckt (2), wobei die abgeschrägte Seite nach unten in Richtung Zahnrad zeigt. Anschließend wird beides mit vier M4-Schrauben am Zentralrad befestigt.

Am Sprinklerarm L wird zunächst mit leichtem Druck das Kugellager eingepresst und am anderen Schenkel mit zwei kleinen Schrauben ein scheibenförmiger Servoadapter montiert. Auf die Rückseite des Sprinklerkopfs **M** werden vier M4-Gewindeeinsätze eingeschmolzen. Danach wird er auf den Sprinklerarm L gesteckt und mit vier M4-Schrauben fixiert **(**). Ein Inbusdreher mit Kugelkopf erleichtert dabei das Festziehen. Wenn gewünscht, kann man nun noch einen andersfarbigen Deko-Ring **N** auf den Sprinklerkopf stecken. Er ist für die Funktion aber nicht erforderlich.



8 Mit diesem Erdsporn steht der Rasensprenger stabil.



Nun befestigt man den fertigen Sprinklerarm waagerecht am Zentralrad J. Dazu schraubt man auf der einen Seite den Servo an den zuvor montierten Servoadapter, indem man den Schraubendreher durch das Kugellager steckt (6). Die andere Seite befestigt man mit einer Schraube M5 × 16mm, die man durch das Kugellager in den ins Zentralrad eingeschmolzenen Gewindeeinsatz schraubt.

Jetzt kann man den Silikonschlauch auf den Anschluss am Sprinklerkopf schieben und mittels einer Schlauchschelle oder eines breiten Kabelbinders fixieren. Danach wird als Knickschutz eine Stahlfeder übergestülpt **1**7. Die Feder darf nicht zu lang sein, damit sie den Arm nicht nach oben drückt. Sie kann provisorisch mit schmalen Kabelbindern zusammengezogen werden, um so die optimale Länge vor dem Kürzen zu bestimmen. Das offene Ende des Schlauches wird nun durch den Federhalter in Richtung des L-Adapters **P** geschoben und am L-Adapter aufgesteckt. Die Fixierung erfolgt ebenfalls mit einer Schlauchschelle **1**8 oder einem breiten Kabelbinder.

Der mechanische Zusammenbau ist nun abgeschlossen und wir sind bereit für einen ersten Trockenlauf ohne Wasser.



12 Mechanischer Aufbau: Getriebe und Elektronik-Box



(B) So müssen die Zahnräder in der Grundposition zueinander stehen.

Make:markt

MIKROELEKTRONIK



AZ-Delivery

Ihr Experte für Mikroelektronik **20% sparen – Code: AZD-Make20** Die passenden MCU, Sensoren, Displays, etc. für Ihr nächstes Projekt! Projektideen im Blog Spezialangebote im Newsletter

www.az-delivery.de

EDUCATION



Das Designstudium für Maker an der HBK Essen!

Der Studiengang Digital Fabrication Design (B.A.) zieht die Kultur der Maker-Bewegung mit ein und verknüpft Ausprobieren, Tüfteln und Experimentieren mit Formgebung und Produktentwicklung. Jetzt bewerben und Produktdesign neu denken! **www.hbk-essen.de**

BÜCHER/ZEITSCHRIFTEN



Der Verlag für kreative Köpfe!

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

www.dpunkt.de



Was Maker schon alles geschaffen haben!

Die Antwort und viele Beispiele finden Leser in unseren Zeitschriften **"Space – das Weltraum Magazin"**, vom "Reißbrett" ins Weltall und dem "Urknall" vieler Computer- und Make-Enthusiasten – dem **"Retro Gamer".**

www.emedia.de

Gewindeeinsätze einschmelzen

Wer das Einschmelzen lieber zuvor einmal üben möchte, kann dies mit dem Probeteil tun (10), um nicht aus Versehen ein benötigtes Bauteil zu beschädigen. Die Gewindeeinsätze (11) lassen sich später mit einem Elektro-Seitenschneider wieder aus dem Probeteil herauslösen und wiederverwenden.



(0) Mit Lötkolben, Zange und Schraube kann man den Gewindeeinsatz langsam in den schmelzenden Kunststoff einlassen.



Eingeschmolzen sitzt der Gewindeeinsatz felsenfest.



Mechanischer Aufbau: Sprinklerarm und Sprühkopf



(b Sprinklerarm im Detail: Ein Inbusdreher mit Kugelkopf hilft beim Schrauben.

Bedienung

Wenn man den Rasensprenger einschaltet, fährt er kurz darauf in Grundposition. Um den PS3-Controller zu verbinden, muss man die PlayStation-Taste () drücken. Die vier schnell blinkenden LEDs zeigen daraufhin die Gerätesuche an. Hat sich die Fernbedienung mit dem Rasensprenger verbunden, leuchtet die erste der vier LEDs auf dem Controller auf und signalisiert, dass der Automatik-Modus aktiv ist (). Insgesamt gibt es drei Modi, zwischen denen man im Betrieb wählen kann. Die LEDs geben eine Rückmeldung darüber, welcher davon gerade aktiv ist:

Manueller Modus (LED-2): Um den Rasensprenger manuell zu bedienen, drückt man die Rechteck-Taste. Nun kann man mit dem linken Stick die horizontale und vertikale Position des Beregnungsarms steuern. Mit dem rechten Stick reguliert man die Wassermenge. Drückt man die Kreuz-Taste, wechselt man wieder in den Automatik-Modus.

Programmier-Modus (LED-4): Wenn man die Kreis-Taste drückt, kann man den Rasensprenger programmieren. Währenddessen bewegen sich die Servos mit, sodass man direkt sieht, was man einstellt. Mit dem linken Stick dreht man zuerst den Beregnungsarm in die gewünschte Richtung und bestätigt danach mit der Dreieck-Taste.

Nun leuchten LED-4 und -3. Mit dem rechten Stick kann man nun die Wassermenge einstellen und mit dem linken Stick die obere Grenze der Beregnungshöhe festlegen. Auch hier bestätigt man anschließend mit der Dreieck-Taste, wenn alles passt.

Jetzt leuchten die LEDs 4, 3 und 2. Mit dem linken Stick legt man nun die untere Beregnungshöhe fest und passt bei Bedarf mit dem rechten Stick nochmals den Wasserdruck an. Zum Abschluss bestätigt man wieder mit der Dreieck-Taste. Hiermit ist die Programmierung für diesen Winkel erfolgreich erstellt und der Rasensprenger geht wieder in den Automatik-Modus (auf Standby). Möchte man einen weiteren Winkel programmieren, drückt man einfach nochmals die Kreis-Taste und startet erneut den Programmiermodus.

Automatik-Modus (LED-1): Mit der START-Taste beginnt man den automatischen Bewässerungszyklus. Der Arm fährt zunächst in Grundstellung und danach die gespeicherten Werte ab. Winkel, in denen die Wassermenge auf null steht, werden übersprungen. Ein Abbruch ist jederzeit mit der *SELECT*-Taste möglich. Nach Ablauf der programmierten Positionen, spätestens aber nach einer vollendeten 360-Grad-Drehung, beendet der Rasensprenger die Bewässerung und kehrt in die Grundstellung zurück.

Wasser marsch

Funktioniert alles nach Wunsch, kann man nun die Platine an ihrem Platz in der Verteilerdose festschrauben und die dauerhafte Verkabelung vornehmen. Um alles für den Einsatz im Garten vorzubereiten, sollte man zunächst den Gartenschlauch anschließen und das Wasser leicht aufdrehen. Wenn an den wasserführenden Bauteilen alles dicht ist und bleibt, kann man den Wasserhahn langsam ganz aufdrehen. Zwischendurch sollte man hin und wieder kontrollieren, ob immer noch alles dicht bleibt, dann den Controller anmelden und probieren, ob im manuellen Modus mit dem rechten Stick die Wassermenge geregelt werden kann.

Nun steht der kompletten Programmierung und dem anschließenden Einsatz im Garten nichts mehr im Wege! —*akf*

(6) Um den Sprinklerarm festzuschrauben, muss man den Schraubendreher durch das Kugellager stecken.





😢 An dem L-Adapter, der im Ventil steckt, wird der Silikonschlauch mit einer Schelle befestigt.





U Die Stahlfeder schützt den Silikonschlauch, damit dieser nicht knickt.



2 Die LEDs am PS3-Controller signalisieren, welchen Modus man gerade verwendet.

Programmierung im Detail

Das Getriebe ist so konstruiert, dass mit den 180 Grad des Servos eine volle Drehung des Sprühkopfs möglich ist. Daher ergibt sich eine Positionierungs-Genauigkeit von 2 Grad. In der Software wird die Richtung, in die der Sprühkopf zeigt, in der Variablen posAZ (Position Azimuth) abgelegt. Die drei Dateien im SPIFFS enthalten für jeden der 180 Winkelschritte einen gespeicherten Wert für:

» ELdn (unterster vertikaler Sprühwinkel) » ELup (oberster vertikaler Sprühwinkel)

» W (Öffnungswert des Wasserventils)

Die programmierten Winkel lassen sich einzeln überschreiben, sofern es einem gelingt, sie exakt anzusteuern. Möchte man komplett von vorne beginnen, muss man die ursprünglichen, leeren Dateien mit der Arduino IDE erneut auf den ESP32 kopieren.

HO-Modellautos mit ESP fernsteuern

Was sich im Maßstab H0 bewegt, ist nicht unbedingt eine Modelleisenbahn: Es gibt auch durchaus Straßenfahrzeuge in dieser Größe, die wie von Geisterhand gelenkt ihre Runden drehen können. In diesem Artikel lernen Sie, wie solche Fahrzeuge funktionieren – und können sogar einen Prototyp mit ESP32-Fernsteuerung aufbauen.

von Gustav Wostrack



er schon einmal das epische Miniatur Wunderland in Hamburg besucht hat, wird sie kennen: Selbstfahrende H0-Modellautos, die den Fahrbetrieb ungemein bereichern – sie halten selbstständig auf Parkplätzen und Betriebshöfen, blinken vorbildgetreu beim Abbiegen und suchen bei versiegender Akku-Leistung ihre Ladestation auf. Das zugrundeliegende Car System wurde ursprünglich von der Modellbau-Firma Faller entwickelt, und natürlich kann der solvente Modellbauer auch fertige, fahrbereite Modelle samt Steuerung kaufen. Wir wollen uns hier aber an eine Selbstbau-Konstruktion wagen, für die man das Car System nicht braucht. Das ist durch vorgefertigte Getriebe und Achsen gar nicht so schwer.

Ein in die Fahrbahn eingelassener Stahldraht (oder ein Magnetband) und ein Schleifer mit einem kleinen Magneten, der an der Vorderachse befestigt ist, hält die Fahrzeuge auf Spur. Der Schleifer schwebt über dem Stahldraht und bewegt sich je nach Fahrbahnverlauf entsprechend horizontal nach links oder rechts und lenkt damit die Vorderräder. Nun braucht man eigentlich nur noch einen kleinen Elektromotor, der die Hinterachse antreibt, einen ebenso kleinen Akku und fertig ist das kleine Car-System-Auto!

Fremdgesteuert

Die erste Generation dieser Autos war auch tatsächlich so einfach gestaltet; zum Anhalten ließ sich der Motor durch einen Reed-Kontakt und ein externes Magnetfeld abschalten. Damit wollen wir uns hier natürlich nicht zufriedengeben: Das Auto braucht Licht, eine Hupe muss her und der Motor muss in der Geschwindigkeit steuerbar sein. Um noch etwas obendrauf zu setzen, entschließen wir uns, ein Einsatzfahrzeug der Feuerwehr aus dem kleinen Dörfchen Plottenstein zu bauen. Deshalb muss das Gefährt auch mit Blaulicht ausgerüstet sein, und die Hupe kann auch das Martinshorn signalisieren. Und alles muss in ein Auto des Maßstabes 1:87 passen. Es hat also eine Größe von ungefähr 10cm Länge und 2,5cm Breite.

Um all das zu realisieren, benötigen wir einen Prozessor im Fahrzeug, der das Ansteuern der LEDs, der Hupe sowie des Motors übernimmt. Wie sich nun das Auto verhalten soll, muss von außen über ein Interface festgelegt werden. Also brauchen wir auch Rechnerleistung außerhalb des Fahrzeuges und eine drahtlose Verbindung mit der Steuerung.

Vielleicht ist dem einen oder anderen der Begriff *CANguru* bekannt; in der Make-Ausgabe 1/21 gab es bereits einen Artikel über diese Entwicklung (siehe Link im Kurzinfo-Kasten). Hierbei handelt es sich um ein serverbasiertes System zur Steuerung von

Kurzinfo

» Selbstfahrende Modellfahrzeuge für Maßstab H0
 » VNC-Server mit Raspberry Pi
 » Fahrzeugelektronik mit Mini-ESP32



Modelleisenbahnen, das gemäß einem Standard der Firma *Märklin* alle Informationen im CAN-Bus-Format versendet, daher der Name. Insofern war es nur naheliegend, das System für Fahrzeuge nun CARguru zu nennen.

Virtual Network Computing

Als Schnittstelle zwischen Bediener am PC oder Handheld-Device und dem Server, der die Fahrzeuge steuert, dient das Programm



Im Hamburger Miniatur Wunderland hat man die detail- und funktionsgetreue Nachbildung von selbstfahrenden Modellautos auf die Spitze getrieben. Hier die automatische Ladestation für die Fahrzeug-Akkus.



CARguru-Funktionsschema: Das Handheld-Device des Bedieners ist im Prinzip nur ein Sichtgerät für den (virtuellen) Server.

VNC, was kurz für *Virtual Network Computing* steht. Das ist eine Software, die in unserem Falle den (virtuellen) Bildschirminhalt des CARguru-Servers auf einem lokalen Rechner, einem PC oder einem sonstigen Handheld oder Smartphone spiegelt, auf dem VNC verfügbar ist, und gleichzeitig Tastatur- und Mausbewegungen von dort an den CARguru-Server sendet.

Wenn der Fahrer nun ein Kommando an das Fahrzeug schickt, beispielsweise die Geschwindigkeit zu erhöhen, wird dieser Befehl zunächst zum CARguru-Server geleitet. Somit bildet der CARguru-Server das Zentrum des Verbundes. Nun müssen die Befehle noch in das Fahrzeug kommen.

Es gibt ein spezielles Drahtlos-Protokoll mit dem Namen ESP-Now, das sich für diese Zwecke hervorragend eignet. Mit diesen Angaben ist schon klar, dass auf Sender- und Empfängerseite weitere kleine Rechner werkeln. Wegen seiner Brückenfunktion zwischen CARguru-Server und Fahrzeug ist der Name *CARguru-Bridge* zutreffend. Leider kann dieser Prozessor die andere Seite mit Ethernet und VNC nicht bewerkstelligen. Deshalb teilen sich ein ESP32 als Bridge und ein Raspberry PI 4B als VNC-Server die Arbeit. Diese beiden Komponenten sind über eine kleine Platine miteinander verbunden und tauschen über ihre seriellen Schnittstellen alle Informationen aus. So lassen sich auch mehrere Fahrzeuge parallel steuern.

Das Aufspielen des VNC-Servers auf den Raspberry 4B und die Konfiguration des ESP32 haben wir wegen des erheblichen Umfangs detailliert in einem zweiteiligen Online-Artikel für Abonnenten von *Make* und *heise*+ beschrieben, den Sie unter dem Link im Info-Kasten finden. Dieser Artikel geht auch auf unseren Quellcode näher ein.

Das Fahrzeug

Ursprünglich hatte ich geplant, das Chassis des Fahrzeuges mit Hilfe des 3D-Druckers

herzustellen. Nach etlichen Versuchen habe ich das Vorhaben aufgegeben: Zu instabil waren die entstandenen Gebilde und mitsamt Träger- und Tochterplatinen auch zu hoch.

Stattdessen habe ich nun die Grundplatine selbst als Chassis verwendet: Das war in der Entwicklung ein riesiger Schritt nach vorne. Auch die Auswahl des Prozessors hat eine längere Geschichte. Alle üblichen Bausteine auf Basis des ESP32 mit dem gängigen 38-Pin-Layout sprengen die Größenbeschränkungen.

In letzten Jahr kam auf Basis des ESP32-S3 ein neues Board der australischen Firma Unexpected Maker auf den Markt. Der Name TinyS3 ist in Anbetracht der Größe (35mm × 17,8mm) treffend gewählt, auch die etlichen Ausgänge erfüllen die Anforderungen. Zudem kann ein Akku direkt angeschlossen und auch geladen werden.

Noch besser geeignet war dann allerdings der gleich große *TinyPico* aus gleichem Hause. Auf beiden Modulen ist ein ESP32 verbaut, aber der Unterschied liegt im Detail: Der Tiny-Pico besitzt einen vollständigen ESP32 mit zwei DACs (Digital Analog Converter), der kleinere Bruder S3 eben nicht. Mit einem DAC lässt sich Sound für das Martinshorn über den Lautsprecher doch deutlich realitätsnäher wiedergeben – und so wurde es eben der TinyPico.

Schaltplan

Das nebenstehende Schaltbild zeigt, dass nur wenige weitere Bauelemente für die Ansteuerung aller Verbraucher notwendig sind. Zur Entlastung der Ausgänge des TinyPico sind für die Ansteuerung des Motors und des Lautsprechers Transistoren vom Typ BC 517 eingesetzt. Es handelt sich um Darlington-Typen, die diese Aufgabe ohne Anstrengung bewältigen.

Auf der linken Seite des Plans finden sich die acht LEDs, mit denen das kleine Fahrzeug ausgerüstet ist. Die Schaltung wird über einen



Zu wackelig: Ein Chassis aus dem 3D-Drucker bewährte sich nicht. Stattdessen bildet jetzt die Platine selbst den Unterbau.



Im Modellbau-Fachhandel findet man vorgefertigte Achsen und Motor-Baugruppen. Zum Zusammenbau müssen Sie kein gelernter Uhrmacher sein.



Die selbstentworfenen und -gefrästen Platinen bilden nun auch das Chassis des Fahrzeugs.

Lilon-Akku mit 3,7 Volt versorgt. Die Aufbereitung der 3,7 Volt für den Prozessor (3,3V) erledigt ein Spannungsregler-Chip auf der TinyPico-Platine.

Genauso ist Sorge dafür getragen, dass der Akku auch wieder aufgeladen werden kann. Dafür gibt es die beiden Anschlüsse, die in der Schaltung mit X1 für Masse (Minuspol) und X2 für 5 Volt (Pluspol) bezeichnet sind. Wir werden später sehen, dass sowohl der Schalter als auch diese Ladeanschlüsse von außen erreichbar sind.

Für das Fahrzeug werden insgesamt vier Platinen benötigt. Im Bild erkennt man in der Mitte die Unterseite der Fahrgestell-Platine, die Lötseite bildet später den "Unterboden" des Fahrzeugs. An dieser Platine gibt es einige Besonderheiten: Die Ausschnitte für die Räder sind mit dem Platinenumriss in einem Arbeitsgang ausgefräst. Im linken Teil der Platine sieht man den späteren Ausschnitt für den Motor, der hier noch von Stegen gehalten wird, und vier Bohrungen für die Befestigung des Motors.

Daneben liegen die Ausschnitte für die Räder der Hinterachse. Da die Vorderräder guasi in einem kurzen Kreisausschnitt beweglich sind, muss die dafür notwendige Aussparung auch entsprechend gestaltet sein. Die Vorderachse selbst wird in das rechteckige Feld dazwischen eingeklebt. Dieses Feld muss 0,3 mm tief eingefräst sein, damit die Vorderachse bündig eingesetzt werden kann und das Chassis waagerecht liegt. An den Fahrzeugenden gibt es jeweils 4 Lötpads für die kleinen LED-Platinen (unten). Sie unterscheiden sich später nur in der Bestückung: Wo vorne weiße LED für die Scheinwerfer eingebaut werden, kommen hinten erwartungsgemäß rote Leuchten für die Rücklichter bzw. das Bremssignal hin. Die orangen Blinker rechts und links gibt es natürlich auf beiden Platinen.

Die obere Platine wird in einem Abstand von ca. 2cm oberhalb der Fahrgestellplatine angebracht. Dazu lötet man auf diese Platine an festgelegten Stellen Stiftleisten-Abschnitte von ca. 1,5cm Länge ein, die in korrespondierende Buchsenleisten gesteckt werden. Indem dadurch eine elektrische Verbindung zwischen beiden Platinen geschaffen wurde, wird es möglich, den Akku, den Schalter sowie den Ladeanschluss oben unterzubringen. Letztere ragen dann gerade aus dem Dach des Fahrzeuges heraus und sind damit von außen zugänglich. Zum einen können die blauen LEDs für die Blitzerbeleuchtung dort platziert werden, genauso wie Verbindungen vom TinyPico zu den LED-Platinen, die ansonsten mit Kabeln hätten geführt werden müssen.

Die beiden kleinen LED-Platinen haben jeweils vier Anschlüsse zur Hauptplatine. Im Unterschied zu den übrigen Platinen sind hier nur die Leiterbahnen stehen geblieben, die restliche Kupferfläche wurde weggefräst. Das ist zwar zeitlich etwas aufwändiger, erleichtert aber das Löten der kleinen LEDs bzw. Widerstände in SMD-Format doch erheblich.

Die Platinen entstanden übrigens auf der Kleinfräse *Genmitsu 3018-PROVer*, diesen Vorgang haben wir bereits in einem früheren Artikel (siehe Kurzinfo) dargelegt. Die fertig gefrästen Platinen sind unbedingt mit einem Ohmmeter oder Durchgangspiepser auf Kurzschlüsse zu kontrollieren.

Beim Einlöten der Bauteile, die unter dem Chip liegen, ist darauf zu achten, dass sie die Oberkante der Buchsenleisten des TinyPico nicht überragen. Zum Anschluss des Lautsprechers und des Motors habe ich kleine Stifte eingelötet und etwas gekürzt.

Zusammenbau

Der Motor wird mit einem angeflanschten Getriebe und Hinterachse als fertige Einheit geliefert. Sie wird mit den abgebildeten Schrauben auf die Platine geschraubt. Links von den Schrauben erkennt man die beiden Zwillingsreifen, die auf die Achsen der Antriebseinheit geschoben werden.

Die Vorderachse wird wie abgebildet geliefert. Unterhalb der Achse erkennt man eine kleine rechteckige Platte, die später in die Ausfräsung der Fahrwerksplatine geklebt wird. Die beiden Felgenkappen können auf die Vorderräder gesteckt werden. Bleibt nur der Schleifer, der mit der winzigen Schraube auf die Vorderachse geschraubt wird, nachdem der Magnet auf den Schleifer geklebt wurde. Dabei ist die richtige Polarität zu beachten, wenn statt des Stahldrahtes ein Magnetband verwendet wird. Also den Magneten zunächst so auf das Magnetband legen, dass er davon angezogen wird, und ihn in dieser Lage einkleben.

Die Ränder der kleinen Platte an der Achse müssen genau parallel zu den Kanten der Platine verlaufen, wenn das Auto später gut geradeaus laufen soll. Das ist aber nicht die einzige Herausforderung: Es soll nicht verschwiegen werden, dass das Anbringen des Magnetschleifers unter der Vorderachse etwas kniffelig ist. Ich habe gute Erfahrung mit folgender Vorgehensweise: Zunächst verbreitert man die Bohrung im Querlenker der Achse mit einem spitzen, scharfen Messer ein wenig, sodass die winzige Schraube leichter ihren Weg in die Bohrung findet. Dann legt man den Schleifer in die endgültige Position auf die Achse und klemmt ihn mit einer kleinen Krokodilklemme fest. So fixiert lässt sich die



Das noch unbestückte Fahrgestell im Rohbau.



Lediglich zwei Transistoren werden für Motor und Lautsprecher benötigt, die LEDs können direkt an den Ausgängen des TinyPico-ESP32 betrieben werden.





Prototyp vor dem endgültigen Zusammenstecken: Stiftleisten dienen zur Verbindung von Dach- und Chassis-Platine.

Die CARguru-Bridge kann auch "fliegend" aufgebaut werden, zwischen ESP32 und Raspi4 sind nur TxD und RxD über Kreuz miteinander verbunden.



Das User-Interface für jedes Fahrzeug, wie es vom Server an den Bediener geliefert wird.

winzige Schraube besser einführen und festziehen (aber vorsichtig: Nach fest kommt lose!).

Die (zugegebenermaßen noch etwas rudimentäre) Karosserie der Feuerwehr habe ich mit *TinkerCAD* entworfen und in zwei Teilen 3D-gedruckt. Vor dem endgültigen Zusammenbau steht allerdings noch ein Test des fertig aufgebauten Chassis an – dafür benötigen wir allerdings zuerst die CARguru-Bridge.

CARguru-Bridge

Die Bridge ist ebenfalls mit einem ESP32 bestückt und hardwaremäßig mit dem CARguru-Server verbunden. Dazu wurde eine kleine Platine entworfen, auf der noch zwei Abgriffe eingeplant wurden. Beide stellen 5 Volt zur Verfügung. Einer ist für die Versorgung eines Lüfters und der andere für die Aufladung eines Fahrzeuges vorgesehen. Wie immer ist beim Anschluss auf die korrekte Polung zu achten. Die einfache Schaltung lässt sich aber ebenso gut "fliegend" mit ein paar Jumper-Kabeln verdrahten.

Für die Bestückung der Platine werden neben dem ESP32 nur einige Buchsen- und Stiftleisten für dieses Modul sowie für den Anschluss des Raspberry Pi benötigt. Bei der Beschaffung des ESP32 muss man etwas genauer hinschauen, es gibt eine Vielzahl unterschiedlicher Module, die mit diesem Prozessor bestückt sind. Wesentlich ist, dass der Baustein einen 38-poligen Anschluss, die korrekte Breite und auch die Pinbelegung eines ESP32-NodeMCU aufweist. Ansonsten muss eventuell das Layout der Platine etwas angepasst werden.

Die Schaltung ist erwartungsgemäß schlicht. Neben den Spannungsanschlüssen sind lediglich zwei Verbindungen nötig. Es handelt sich um die "kreuzweise" serielle Verbindung zwischen dem ESP32 und dem Raspberry Pi. Die Kontakte am oberen Rand sind für die Versorgung des bereits angesprochen Lüfters und der Fahrzeug-Ladebuchse vorgesehen.

Es wäre naheliegend, den ESP32 aus der Stiftleiste des Raspi-GPIO zu versorgen. Das funktioniert auch halbwegs. Wenn aber ein Lüfter und ein Fahrzeug zum Laden angeschlossen sind, geht nichts mehr; der Strombedarf kann so nicht mehr gedeckt werden. Ich empfehle, die Bausteine aus einem USB-Doppelnetzteil zu versorgen (siehe Bild).

User-Fontend

Die folgende Vorgehensweise setzt voraus, dass der VNC-Server und die ESP32-Module entsprechend der Online-Beschreibung vorbereitet sind. Das Frontend auf dem Raspi wird aus dem *LXTerminal* durch Eingabe von ./CARguru.sh gestartet. Nun sollte man die Fahrzeuge unter Spannung setzen. Im Frontend klickt man dann auf den *Start*-Knopf im unteren Bereich des Fensters. Zu den beiden Einträgen in der linken Spalte sind nun zwei weitere hinzugekommen.

Im oberen Bereich sieht man eine Liste der angeschlossenen Fahrzeuge, deren IP- und Moduladresse. Wenn Fahrzeuge frisch mit einer Firmware "bedampft" worden sind, verfügen sie immer über die gleiche Adresse 1. Der erste Schritt der Inbetriebnahme ist also, die Fahrzeuge mit unterschiedlichen Adressen zu versehen.

Dazu doppelklickt man auf den Eintrag des Fahrzeuges. Durch einen weiteren Klick auf die nun erschienene Adresse kann diese mit den Plus- und Minus-Buttons geändert werden. Mit *Speichern* wird dem Fahrzeug diese Adresse zugeordnet; sichtbar wird sie allerdings erst nach einem Neustart des Programms. Die angezeigte IP-Adresse wird für den *Over-the-Air*-Programmiervorgang (OTA) benötigt.

Kommen wir zu Play: Mit diesem Menü werden die Fahrzeuge gesteuert. Jedem angeschlossenen Fahrzeug ist ein solches Menü zugeordnet. Zur Steuerung der Geschwindigkeit gibt es einen Schieberegler. Unterhalb des Tastenfeldes wird die aktuelle Batteriespannung angezeigt, die das Fahrzeug alle 5 Sekunden aktualisiert. Die Bedeutung der weiteren Tasten ist von links oben: Licht ein/aus, Blinker links, Blinker rechts, Warnblinker, Lichthupe, Fernlicht, Hupe, Martinshorn und Blaulicht. Das offene Protokoll würde es natürlich ermöglichen, die Funktionen für einen automatisierten Fahrbetrieb auch programmgesteuert zu senden; der Autor arbeitet bereits an solchen Erweiterungen. —cm

building

Die Konferenz zu (l)loT

26. und 27. April 2023 in München

Software entwickeln für das (I)lot

Die Fachkonferenz building IoT ist seit 2016 der Treffpunkt für diejenigen, die Softwareanwendungen und digitale Produkte im Internet der Dinge und im Industrial Internet of Things entwickeln.

Das Programm bietet an zwei Tagen in drei Tracks 36 Vorträge unter anderem zu folgenden Themen:

- Datenanbindung und -analyse für das IIoT
- Edge-Computing mit Kubernetes
- Eclipse Sparkplug in Action
- Zeitreihendatenbanken f
 ür das IoT
- Maschinenbau trifft auf agile Softwareentwicklung

dpunkt.verlag

- Rust auf dem Mikrocontroller
- IoT Cybersecurity: EU-Normen-Update

www.buildingiot.de



Jetzt Tickets sichern!





Goldsponsoren







Bronzesponsor



heise Developer

BASIC-Roboter als Luftmessstation

Mithilfe von Umgebungssensoren für CO₂, Temperatur und Luftdruck spürt der sechsbeinige Laufroboter schlechte Luft auf und signalisiert dies seinem Besitzer eindringlich. Der 3D-gedruckte Roboter kommt mit drei Servos aus, die von einer modernen BASIC-Variante gesteuert werden.

von Thomas Euler



Die Idee zu diesem Projekt entstand beim Experimentieren mit dem kombinierten Kohlendioxid-Temperatur-Feuchtesensor *SCD41*, der als Breakoutboard von *Pimoroni* erhältlich ist. Mit einem Mikrocontroller als Gehirn und einem kleinen Farbdisplay lässt sich aus diesem Sensor leicht eine Messstation für das Raumklima aufbauen.

Um die Warnung vor ungesunden CO₂-Konzentrationen eindringlicher zu machen, sitzt die Messstation auf einem kleinen Roboter, der zum Lüften auffordert. Dieser sechsbeinige, handtellergroße Roboter wurde bereits in der Make 6/21 auf Seite 94 als *Robotling2* vorgestellt. Er verwendet als Mikrocontroller einen Raspberry Pi Pico und kann wahlweise mit C++, *MicroPython* oder dem *BASIC*-Dialekt *MMBasic* programmiert werden.

Um zu zeigen, dass ein modernes BASIC durchaus für Projekte wie dieses geeignet ist, habe ich hier MMBasic als Programmiersprache gewählt. Für eine Einführung in MMBasic, seine Möglichkeiten und Grenzen sowie Hinweise zur Installation der Firmware auf dem Pico-Mikrocontroller sei auf den Artikel ab Seite 73 in Make 1/23 verwiesen.

Der Roboter

Der Roboter ist der Nachfolger des *Robotling* (siehe *Mehr zum Thema*), der vor ein paar Jahren auf der Basis eines ferngesteuerten Spielzeugroboters *HEXBUG* entstand. Der *Robotling2* verfügt über eine ähnliche Funktionalität 1, benötigt aber keinen *HEXBUG* mehr: Er besteht hauptsächlich aus 3D-gedruckten Kunststoffteilen, drei Servomotoren für die Bewegung der Beine und einer einfachen Platine, die einen Raspberry Pi Pico aufnimmt.

Die hier beschriebene Version des Roboters verfügt über drei Abstandssensoren, die Objekte in bis zu 50cm Entfernung erfassen und die

Kurzinfo

» Autonomer, insektoider Roboter mit CO₂-Warnfunktion
 » Leicht zu drucken, kostengünstig, gut zu erweitern
 » Interaktiv programmiert in BASIC

Checkliste	Werkzeug
Zeitaufwand: 4 Stunden (ohne Druckzeit)	» 3D-Drucker » übliches Bastlerwerkzeug Schraubendreherset, Zangen, etc.
Kosten: 50 Euro (nur Laufroboter) bis 160 Euro (mit allen Sensoren und Display)	» Lötkolben und Zubehör
3D-Drucker: ab 100mm × 100mm Druckbett	Mehr zum Thema
Material » Raspberry Pi Pico Standard oder W-Modell » 3D-Druckteile PLA oder PETG » Spannungsregler Akkuspannung zu 5V » Drei Microservos	 » Thomas Euler, Raspberry Pi Pico: Back to BASIC, Make 1/23, S. 72 » Thomas Euler, Insektoider Roboter spricht Basic, Make 6/21, S. 94 » Hans-Martin Hilbig, Stromausfall-Monitor mit dem Pico selbst bauen, Make 4/22, S. 62 » Thomas Euler, Krabbeltier mit Python-Antrieb, Make 3/19, S. 88 » Video: Der Robotling stellt sich vor
Alles zum Artikel	Make: Robotling2

Distanz als gepulstes Signal zurückliefern. Sie sind schräg nach unten gerichtet und messen den Abstand zum Boden. Durch diese Anordnung können nicht nur Hindernisse, sondern auch Tischkanten erkannt werden. Alles, was zwischen 5 und 20cm gemessen wird, ist freier Raum; kleinere Abstände werden als Hindernisse, größere als Kanten samt Abgrund



Der Roboter in Übersicht als Foto (links) und CAD-Zeichnung (rechts)

Spannungswandler

Der Sockel J2 wurde für das 5V-Step-Up-Spannungswandler-Modul Pololu U1V10F5 entworfen. Das Modul ist aber aktuell nicht oder nur zu sehr hohem Preis verfügbar. Ein fast gleichwertiger Ersatz ist das Modul U3V16F5. Dort sind aber die Pins für VIn und VOut vertauscht, d. h. das Ersatzmodul muss kopfunter (Komponenten zum Board hin) eingesteckt werden, ansonsten können leicht Komponenten beschädigt werden!

dahinter interpretiert. Hinzu kommen ein ausklappbares Farbdisplay und die bereits erwähnten Sensoren für Kohlendioxid (CO₂), Temperatur und Luftfeuchtigkeit.

Um sich fortzubewegen, nutzt der Roboter ein einfacheres Prinzip als der erste Robotling, der eine komplizierte Kinematik und Getriebe braucht. Dafür benötigt er drei statt zwei Motoren: Ein Servo bewegt das mittlere Beinpaar und kippt den Roboter abwechselnd nach rechts oder links. Dadurch können die beiden anderen Servos, die jeweils das vordere und hintere Bein einer Seite hin und her schwingen, den Roboter bewegen. Je nach Reihenfolge und Richtung dieser Bewegungen läuft der Roboter vorwärts, rückwärts oder dreht sich auf der Stelle.

Um den Bau für den Einsteiger zu vereinfachen, habe ich eine Platine entworfen, die

den Aufbau erleichtert und eine sichere Basis auch für weitere Experimente bietet. Die Platine kommt mit nur wenigen Bauteilen aus. Das einzige SMD-Bauteil ist ein MOSFET, der verhindert, dass Strom vom USB-Anschluss unkontrolliert in den Akku fließen kann, da die Platine keine Ladeelektronik enthält. Die Spannung aus dem einzelligen LiPo-Akku wird von einem Spannungswandler auf 5V hochgewandelt, stabilisiert und versorgt das komplette Board inklusive Pico und Servomotoren. Auf der Platine befinden sich außerdem eine LED. ein Reset-Taster ein Fin/Aus-Schalter sowie ein STEMMA-Anschluss, an den man viele Sensor-Breakouts (z. B. von Adafruit) mit dem I²C-Bus des Pico verbinden kann. Dort wird auch der CO₃-Sensor angeschlossen. Der Pico selbst wird auf der Unterseite der Platine montiert, sodass die Pins des Mikrocontrollers auf der Oberseite zugänglich sind, z. B. um das Display anzuschließen 📿

Da der *Robotling2* als flexible Roboterbasis gedacht ist, besitzt er vorne an der Platine eine Anschlussleiste für Erweiterungen ③. Dort werden neben verschiedenen Spannungen (3,3V und 5V) auch neun GPIO-Pins des Pico zur Verfügung gestellt. Über diese können verschiedene serielle Schnittstellen (I²C, SPI, COM-Port), drei der Analog-Digital-Wandler (*A0* bis *A2*) und mehrere PWM-Kanäle (z. B. für Servomotoren) des Pico benutzt werden.

Die Funktion der meisten Pico-Pins ist per Software definierbar; so z. B. kann *GP26* als Analog-In (*A0*), SPI-Clock, *I²C-SDA* oder PWM-Kanal *5A* dienen. Diese Eigenschaft macht die Anschlussleiste sehr flexibel. Beim CO₂-Wächter-Roboter sind hier die drei Distanzsensoren sowie der Servo, der den Bildschirm hochklappt, angeschlossen. Die Leitungen zur Ansteuerung des Displays sind alle oben auf der Platine mit den Pins des Pico verbunden 2. Dabei belegt die Hintergrundbeleuchtung des Displays (BL für *backlight*) *GP22*, dessen Entsprechung an der Erweiterungsleiste deswegen nicht für anderes nutzbar ist. Weitere Details zum mechanischen Aufbau und der Elektronik des Roboters finden sich auf der Projektseite (siehe Link in der Kurzinfo).

Der Zusammenbau

Die Montage des Roboters wird in Übersicht d und ausführlich in einem Video gezeigt (siehe Link in Kurzinfo). Eine Liste der benötigten Elektronikteile sowie Hinweise zum Zusammenbau findet sich auf GitHub. Die Plastikteile können mit einem normalen 3D-Drucker aus PLA-Filament gefertigt werden. Neben einer Lötstation werden nur Werkzeuge gebraucht, die sich in jeder gut sortierten Maker-Werkstatt finden lassen.

Das Programm

Der Roboter kann in verschiedenen Sprachen programmiert werden; auf der Projektseite befindet sich auch Code in MicroPython, allerdings nicht für die CO₂-Wächterversion des *Robotling2*. Diese ist in *MMBasic* implementiert, einem BASIC-Dialekt, der etwa in der australischen Maker-Community weit verbreitet ist und dort zur Steuerung kleinerer Projekte eingesetzt wird. Spätestens seit dem modernen Homecomputer *Color Maximite 2* ist MMbasic weltweit beliebt.

Die Programmierung des Roboters erfolgt direkt auf dem Pico, mit dem man mittels des



2 Das Display ist auf der Oberseite der Platine direkt an die Pins des Pico über den SPI-Bus angeschlossen. Das CO₂-Sensor-Breakout hat wie die Platine eine 4-adrige STEMMA-Buchse; ein kurzes Verbindungskabel verbindet den Sensor mit dem l²C-Bus des Pico.

8

Terminalprogramms *TeraTerm* von einem PC aus Kontakt aufnimmt. Im Folgenden gehe ich davon aus, dass die aktuelle MMBasic-Firmware (5.07.xx) auf dem Pico frisch installiert wurde und die BASIC-Kommandozeile (>) in *TeraTerm* erscheint. Das aktuelle CO₂-Wächterprogramm (*rbl2_v1_13_co2.bas*) befindet sich auf GitHub im Verzeichnis *code/mmbasic* (siehe Link).

Zunächst muss MMBasic mit der Hardware bekannt gemacht werden: Dazu gibt man die OPTION-Befehle in den Kommentaren am Anfang des Programms in die Kommandozeile ein. Danach überträgt man das Programm vom PC auf den Pico: Nach der Eingabe von XMODEM R wartet der Pico darauf, dass man mit *File/Transfer/XMODEM/Send* die Übertragung in TeraTerm startet. Das Programm befindet sich nun im Flashspeicher des Pico und kann mit *F2* ausgeführt werden. Um es nicht versehentlich zu überschreiben, sollte man den Code auch in einen der fünf Programmspeicher kopieren (z. B. mit FLASH SAVE 1).

In der MMBasic-Version für den Pico ist es leider nicht möglich, Funktionen (wie z. B. die für den CO₂-Sensor) in getrennte Dateien auszulagern. Daher steckt der gesamte Code in einer einzigen Programmdatei. Um nicht immer durch jede Menge Code scrollen zu müssen, um ans Hauptprogramm zu kommen, befinden sich alle Funktionen zur Steuerung

Die Anschlussleiste für Erweiterungen

Die Pins (von vorn, Platine oben) können per Software konfiguriert werden, sodass sie verschiedene Funktionen annehmen können (rechts oben). Zum Beispiel kann Pin GPO als SDA- (I2C-Schnittstelle 1), TX- (COM-Port 1) oder PWM-Kanal 0A eingesetzt werden. An AO bis A2 liegen drei der Analog-Digital-Wandler des Pico. Die Belegung der Pins beim $\rm CO_2$ -Robotling sind unten gezeigt.



Grundlagen der Robotersteuerung

Alle Kommandos, Konstanten und Variablen, die die Steuerung des Roboters verkapseln, erkennt man daran, dass sie mit R. beginnen. So kann man beispielsweise durch Setzen der Konstanten R.SERVOS_ON festlegen, ob die Servos angesteuert werden - zum Testen kann es durchaus nützlich sein, wenn der Roboter nicht dauernd versucht, wegzulaufen (R.SERVOS_ON=0). Variable wie R. running zeigen den Zustand des Roboters an; hier, ob das Hauptprogramm läuft (1) oder nicht (0). Die wichtigsten Kommandos sind im Folgenden mit ihren Parametern erklärt. Eckige Klammern [] markieren optionale Parameter, bei denen Ø als Wert angenommen wird, wenn sie nicht angegeben sind. Durch einen Doppelpunkt getrennt lassen sich mehrere Anweisungen hintereinander in eine BASIC-Zeile schreiben.

R.Move *Richtung, Geschwindigkeit* – die Richtung wird von vier Konstanten bestimmt: vorwärts (FORWARD) oder rückwärts (BACKWARD), auf der Stelle drehend (TURN_LEFT, TURN_RIGHT). Der Parameter Geschwindigkeit nimmt Werte zwischen Ø (keine Bewegung) und 100 (maximale Geschwindigkeit) an. Der Befehl startet die Bewegung, kehrt aber sofort zum Programmcode zurück, sodass das Programm andere Dinge erledigen kann. Allerdings muss R.Spin (siehe unten) regelmäßig aufgerufen werden, damit die Bewegung weiter läuft.

R.Stop – bewegt die Beine in die Ruhestellung und hält den Roboter an.

R.Spin Dauer[,Bedingung] – pausiert das Programm für Dauer Millisekunden oder wartet, wenn Bedingung = 1 ist, darauf, dass der letzte Schritt des Roboters abgeschlossen ist. Alle Hintergrundprozesse laufen weiter, d. h. die Beine werden je nach dem vorherigen R.Move-Kommando bewegt und falls vorhanden werden Sensoren abgefragt und das Display aktualisiert.

Mit diesen Befehlen kann man den Roboter bereits im Quadrat laufen lassen. Hierzu bietet es sich an, das Programm *rbl2_v1_11_basic.bas* zu benutzen. Dies ist eine abgespeckte Version ohne Sensoren oder Displays, so lassen sich die Bewegungen des Roboters mit recht wenig Aufwand an Hardware erforschen.

In der Hauptschleife wird der Roboter abwechselnd für ein paar Sekunden gedreht (R.Move dir, 60: R.Spin 5000) oder vorwärts bewegt (R.Move FORWARD, 60: R.Spin 4000). Die Drehrichtung (dir) wird alle vier Drehungen geändert, sodass der Roboter erst ein Quadrat im und dann gegen den Urzeigersinn läuft. Sobald R.running=0 ist – z.B., wenn man die Taste Q drückt – endet die Hauptschleife und der Roboter stoppt mit der Sequenz R.Stop: R.Spin 2000, 1. Anschließend schaltet R.Shutdown die Servos und falls vorhanden das Display ab.

Weitere Befehle, Hilfsprogramme und Hinweise finden Sie im Wiki meines GitHubs (siehe Link in der Kurzinfo).



4 Zusammenbau des Roboters

des Roboters und des Displays sowie zum Auslesen des CO₂-Sensors hinter dem Hauptprogramm angeordnet. Alle Befehle, Variablen und Konstanten, die etwas mit der Steuerung des Roboters zu tun haben, beginnen mit R. (für *Robotling*, z. B. R. Move, siehe auch Kasten *Grundlagen der Robotersteuerung*). Der Punkt ist in MMBasic als Namensbestandteil erlaubt und kein Hinweis auf objektorientierte Programmierung, sondern dient nur der Strukturierung des Codes. Weil die meisten Funktionen des Roboters in Subroutinen und Funktionen verpackt sind, ist das eigentliche Hauptprogramm kurz.

Das Programm (*rbl2_v1_13_co2.bas* im Github-Repository) beginnt mit der Definition von Konstanten (Const), die die Pin-Belegung der Anschlussleiste festlegen und das Verhalten des Programms generell regeln. Wichtig ist die Konstante R.CO2. Ist sie gleich 1, benimmt sich der Roboter als CO₂-Wächter, bei 0 läuft er nur umher.

Es folgt ein Sprung in den Bereich des Programms, in dem der Roboter initialisiert wird (GoTo InitRobot) – auch dieser Code liegt der Übersicht halber hinter dem Hauptprogramm. Hier verwende ich den verpönten GoTo-Befehl anstatt ein Unterprogramm (Sub) aufzurufen, weil bei der Initialisierung eine Reihe interner globaler Konstanten definiert werden müssen, was in einem Unterprogramm nicht geht. Die Initialisierungsroutine springt dann zurück zum Label RobotMain, und das Hauptprogramm kann beginnen. Die Distanzsensoren werden initialisiert (R.CreateSensor) und nach einer Startnachricht zeigt das Display die GUI.

In der Hauptschleife arbeitet der Roboter periodisch zwei Aufgaben ab: Die erste ist die Abfrage der Sensoren – wie hoch ist die CO₂-Konzentration; erfassen die Distanzsensoren ein Hindernis? Alle Ereignisse werden in der Variablen ev als Bitmuster gesammelt. Zum Beispiel bedeutet Bit 3 (&B00000100), dass der rechte Sensor einen kürzeren Abstand misst, als zu erwarten ist, wenn nur der Boden erfasst wird, und sich daher rechts vorne wahrscheinlich ein Hindernis befindet (EV_OBJ_R). Die zweite Aufgabe ist es, auf die Ereignisse zu reagieren: z. B., wenn ev And EV_OBJ_R wahr ist, sich vom Hindernis wegzudrehen. Nach dem Befehl zum Drehen (R. Move TURN_LEFT, 60), bekommt der Roboter zwischen 1 und 5 Sekunden Zeit, sich zu bewegen (R.Spin 1000 +Int(Rnd()*4000), bevor die Schleife weiterläuft. Wenn die CO₂-Konzentration einen Grenzwert übersteigt (EV_C02_HI); dann erwacht der Roboter, klappt sein Display hoch und beginnt hin- und herzulaufen. Bei niedrigen CO₂-Konzentrationen ist der Roboter inaktiv.

Die Subroutine R.Spin ist wichtig, da sie den Roboter am Laufen hält: Sie aktualisiert Sensorwerte und GUI und sorgt dafür, dass die Positionen der Servos aktualisiert werden, wenn der Roboter sich bewegt. Daher muss R.Spin regelmäßig in der Hauptschleife aufgerufen werden. Da die Subroutine eine Wartezeit als Argument entgegennimmt, ersetzt sie außerdem den BASIC-Befehl Pause, bei dessen Verwendung der Roboter ins Stocken geraten würde. Neben R.Spin kümmert sich die Subroutine _cb_moveServos um die flüssige Bewegung des Roboters: Sie wird von einem Timer alle 15 Millisekunden aufgerufen (Set-Tick SERV_STEP_MS, _cb_moveServos, 1), um die Positionen der Servos zu aktualisieren. Daher kann man mit dem Befehl R. MoveServos einfach für jeden Servo eine Zielposition (als Winkel) sowie die Zeit angeben, in der die Bewegung abgeschlossen sein soll -_cb_move-Servos kümmert sich im Hintergrund um die Ausführung.

Den CO₂-Sensor auslesen

Der CO₂-Sensor *SCD41* von *Sensiron* misst die Kohlendioxid-Konzentration, Temperatur und Feuchtigkeit der Luft **5**. Draußen, an der frischen Luft, sollte er einen Wert von etwas mehr als 400ppm (*parts per million*) liefern, was einem CO₂-Anteil von 0,04% entspricht. Ist das nicht der Fall, sollte man ihn kalibrieren. Wie man kalibriert, wird im Datenblatt beschrieben (siehe Links).

Auch die Höhe des Standorts über dem Meeresspiegel (Meereshöhe) hilft dem Sensor bei der Berechnung des CO₂-Werts. Das Beispiel im Listing *Sensoren auslesen* zeigt, wie man die Daten des Sensors über die I²C-Schnittstelle ausliest und ausgibt.

Als erstes öffnen wir die l²C-Schnittstelle an Pins GP0 und GP1 (I2C Open) und definieren die Sensoradresse (I2C_ADDR). Der SCD41- Sensor erwartet zwei Byte lange Kommandos (siehe Datenblatt). Je nach Art des Kommandos ist es wichtig, kurz zu warten, bis man weitere Befehle sendet. Im Beispiel habe ich den entsprechenden Pause-Befehl an das jeweilige l²C-Kommando angehängt.

I2C Write schickt eine Anzahl von Bytes an den Sensor; I2C Read liest die Bytes, die der Sensor zurückgibt, in ein Array. Je ein I2C Write-Befehl initialisiert den Sensor und weist ihn an, die Messungen zu starten. Die Funktion isDataAvailable fragt beim Sensor nach, ob neue Messwerte vorliegen. Die Subroutine getData fordert die Messwerte an. Diese sind 16-Bit Integer und müssen aus jeweils zwei Bytes zusammengesetzt werden.

Die CO₂-Konzentration kommt direkt als ppm-Wert, Temperatur und Luftfeuchtigkeit müssen berechnet werden; die entsprechenden Formeln stehen im Datenblatt des SCD41. In der For-Next-Schleife wird isDataAvailable so lange aufgerufen, bis Messdaten vorliegen. Diese werden abgefragt und formatiert angegeben (Print). Nach zehn Durchläufen werden die Messungen mit einem I2C Write-Befehl gestoppt.

Ausblick

In der beschriebenen Version erfüllt der Roboter seine Aufgabe als CO₂-Wächter, aber es sind natürlich Verbesserungen denkbar: Ein Gehäuse würde ihm ein freundlicheres und Wohnzimmer tauglicheres Aussehen geben. Das kann man z. B. mit der Software *FreeCAD* machen, mit der der Roboter auch entworfen wurde. Alle Konstruktionsdateien liegen auf meinem GitHub.

Außerdem müsste der Schlafzustand optimiert werden; da verbraucht der Roboter noch ziemlich viel Strom. Ein erster Schritt könnte hier sein, dass sich der Roboter bemerkbar macht, wenn man ihn laden sollte – mit dem Code-Schnipsel VSYS=Pin(GP29)*3 lässt sich die System-Spannung des Pico mittels des Analog-Digitalwandlers an Pin 29 bestimmen, um auf einen Spannungsabfall reagieren zu können.

Der Roboter könnte seine Messwerte auch drahtlos an einen PC weitergeben, z. B. um das Raumklima aufzuzeichnen: Die zwei letzten freien Pins der Erweiterungsleiste sind GP4 und GP5; hier kann man ein Bluetooth-Modul mit dem zweiten COM-Port des Pico verbinden. Wie das geht, lesen Sie in dem Artikel ab Seite 72 in der Make 1/23 (siehe Mehr zum Thema in der Kurzinfo). Über diese Bluetoothverbindung kann der Roboter per Tera-Term auch programmiert werden, ohne das ein Kabel dabei stört. Ein WLAN-Modul (z. B. auf Basis eines ESP) wäre dann der nächste Schritt, um den Roboter auch über weitere Entfernungen steuern zu können. Es gibt inzwischen auch eine MMBasic-Betaversion mit WLAN-Unterstützung für den Pico W; wegen der etwas anderen Pin-Belegung müsste man das Robotling-Board allerdings dafür anpassen.

Weitere Sensoren und Aufgaben sind natürlich ebenso spannend – reicht ein Pico vielleicht schon aus, um einen autonomen Roboter zu bauen, der sich in seiner Umgebung zurechtfindet? Kann er etwa Überwachungsaufgaben übernehmen, indem er durch die Zimmer streift und offene Fenster oder Eindringlinge detektiert? Oder können mehrere Roboter miteinander interagieren? Welche Aufgaben jenseits des Spiel- und Lernfaktors kann so ein kleiner Roboter sonst noch übernehmen? Schreiben Sie uns gerne eine Mail mit Ihren Ideen und Erfahrungen an info@make-magazin.de. —*caw*



S Neben CO₂-Konzentration, Temperatur und Luftfeuchtigkeit werden auch die Messwerte der Distanzsensoren, der Akkuzustand und die eingestellte Meereshöhe (wichtig für die Konzentrationsberechnung) angezeigt.

```
Sensoren auslesen
Print "Initialize I2C ..."
SetPin GP0, GP1, I2C
I2C Open 100, 500
Const I2C_ADDR = 98
Print "Initialize sensor and start measuring
I2C Write I2C_ADDR, 0, 2, &H36, &H46 : Pause 50
I2C Write I2C_ADDR, 0, 2, &H21, &HB1 : Pause 2
Function isDataAvailable()
  Local Integer buf(2) = (0,0,0)
I2C Write I2C_ADDR, 0, 2, &HE4, &HB8 : Pause 2
I2C Read I2C_ADDR, 0, 3, buf()
isDataAvailable = ((buf(0) << 8) Or buf(1) And &HFFF) > 0
End Function
Sub getData CO2, Tmp, Hum
  Local Integer val, buf(8) = (0,0,0,0,0,0,0,0,0)

I2C Write I2C_ADDR, 0, 2, &HEC, &H05 : Pause 1

I2C Read I2C_ADDR, 0, 9, buf()

C02 = (buf(0) << 8) Or buf(1)

val = (buf(3) << 8) Or buf(4)

Tmp = ((21875 trail) >> 12) (5000) (1000)
   Tmp = (((21875 *val) >> 13) -45000) /1000
  val = (buf(6) << 8) Or buf(7)
Hum = ((12500 *val) >> 13) /1000
End Sub
Dim integer i
Dim float CO2, T, H
For i=1 To 10
   Print "Waiting for data ..
   Do : Pause 100 : Loop Until isDataAvailable()
  getData CO2, T, H
Print "CO2 = " +Str$(CO2,5,0) +" ppm"
Print " T = " +Str$(T,-3,1) +Chr$(18
                T = " +Str$(T, -3, 1) +Chr$(186) +"C"
   Print " H = " +Str$(H,5,0) +"%"
  Pause 1000
Next i
Print "Stop measuring ..."
I2C Write I2C_ADDR, 0, 2, &H3F, &H86 : Pause 600
```

Adafruit GFX Library: Malen mit Zahlen

Manchmal braucht man gar keine komplexen Grafikprogramme, um etwas auf einem Bildschirm zu malen. Mit ein paar simplen geometrischen Formen, Grundkenntnissen im Programmieren und ein bisschen Kreativität werden Dreiecke zu Kreisen und Balken, die sich biegen. Wir zeigen, wie das geht und warum das praktisch ist.

von Ákos Fodor

isplays sind eine praktische Ergänzung für viele Maker-Projekte, die mit uns kommunizieren und uns Feedback geben. Als Mensch-Maschine-Schnittstelle zeigen sie beispielsweise den Status eines Gerätes oder diverser Sensoren an und dank ihrer Fähigkeit, Grafiken abzubilden, lassen sich selbst komplexe Informationen wie Symbole, Diagramme oder Animationen darstellen. Was ein paar einfache LEDs manchmal nicht vollbringen, kann ein Display dank grafischer Bibliotheken mit Leichtigkeit umsetzen. Welche Werkzeuge die Adafruit GFX Library dafür bereithält und wie man selbst als Display-Neuling aus Dreiecken und Ouadraten erste Ladebalken erstellen kann, die sich steuern lassen, zeige ich in diesem Workshop – nur Bitmaps widmen wir uns erst beim nächsten Mal.

Aus den zahllosen Bildschirmen, die Adafruit unterstützt, habe ich zwei ausgewählt, die euch einen direkten Einstieg ermöglichen: ein monochromes OLED-Display (*SSD1306*) und ein TFT-Touch-Farbdisplay (*IL19341*). Beide unterscheiden sich in Größe, Auflösung, Darstellungsart und Farbtiefe sowie darin, wie sie mit einem Mikrocontroller kommunizieren. Während dem SSD1306 die Übertragungsrate der Schnittstelle *I*²C (Inter-Integrated Circuit) **1** völlig ausreicht, benötigt das IL19341 aufgrund der größeren Datenmengen das schnellere *SPI* (Serial Peripheral Interface) **2**, dafür aber auch doppelt so viele *GPIO*-Pins am *ESP32*.

Außerdem ist ein $10k\Omega$ -Potentiometer (kurz: Poti) Teil des Aufbaus. Während wir bei einem fertigen Projekt Daten von Sensoren oder aus anderen Quellen erhalten, hilft ein Poti beim Experimentieren und wir können schnell testen, ob unsere Programmierung wie gewünscht funktioniert.

In dem *GitHub*-Repository zum Projekt findet ihr die Sketches für beide Aufbauvarianten, sodass ihr die Displays nicht mehr manuell einbinden müsst. Für das Verständnis gehe ich dennoch darauf ein, was es mit den wenigen Codezeilen auf sich hat.

Kurzinfo

- » Ein SSD1306- oder ILI9341-Display mit einem ESP32 verwenden
- » Werkzeuge der Adafruit GFX Library kennenlernen
- » Dynamische Balkenanzeigen erstellen

Checkliste



Zeitaufwand: ab1Stunde



Programmieren: mit der Arduino IDE

Material

» ESP32-WROOM32 KS0413

- von keyestudio
- » SSD1306 OLED-Display
- mit 128 × 64 Pixeln » ILI9341 TFT-Farbdisplay
- » 10kΩ Potentiometer
- » Breadboard mit Kabeln

Arduino IDE und Bibliotheken

Doch zuerst benötigt ihr die Arduino IDE, mit der ihr den Mikrocontroller für die grafische Ausgabe programmieren könnt. Einen Download-Link und die Anleitung, wie man einen ESP32 damit in Betrieb nimmt, findet ihr in der Kurzinfo. Wählt anschließend in der Arduino IDE unter Werkzeuge das Board ESP32 Dev Module aus. Ist der Mikrocontroller einsatzbereit, ladet euch als Nächstes mithilfe des Bibliotheksverwalters folgende Bibliotheken herunter, damit der ESP32 weiß, wie er die Bildschirme anzusteuern hat:

- Mehr zum Thema
- » Peter Tschulik, LED-Laufschrift mit ESP32, Make 4/21, S. 8
- » Matthias Helneder, TFT-Displays im Stil von Nixie-Röhren, Make 5/20, S. 46
- » Carsten Romahn, Touchscreen GUI mit dem Raspberry, Make 6/19, S. 98
 » Maik Schmidt, Arduino am VGA-Monitor, Make 1/18, S. 54





– Adafruit SSD1306

- Adafruit ILI9341
- Adafruit GFX Library

Displays ansprechen

Wenn ihr das SSD1306-Display zum Experimentieren verwenden wollt, benötigt euer Sketch die folgenden Informationen, um auf den Bildschirm schreiben zu können:

#include <Wire.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_GFX.h>





2 Um ein ILI9341-Display anzuschließen, benötigt ihr doppelt so viele Kabel. Dafür habt ihr eine 16-Bit-Farbwiedergabe.

```
Adafruit_SSD1306 display =
Adafruit_SSD1306(128, 64, &Wire, -1);
```

ILI9341: Vordefinierte Farben

ILI9341_BLACK	ILI9341_GREEN
ILI9341_NAVY	ILI9341_CYAN
ILI9341_DARKGREEN	ILI9341_RED
ILI9341_DARKCYAN	ILI9341_MAGENTA
ILI9341_MAROON	ILI9341_YELLOW
ILI9341_PURPLE	ILI9341_WHITE
ILI9341_OLIVE	ILI9341_ORANGE
ILI9341_LIGHTGREY	ILI9341_green- Yellow
ILI9341_DARKGREY	ILI9341_PINK
ILI9341_BLUE	

display.dim(1); display.display(); }

Diese Zeilen laden die Wire-Bibliothek für die I²C-Kommunikation, die Bibliothek für das Display und die Adafruit GFX Library. Daraufhin definieren wir den Bildschirm als display-Objekt mit der Auflösung 128 × 64 Pixel. Anschließend wird der Bildschirm im void setup() mit dem Befehl display.begin() über die Standard-I²C-Adresse 0x03C initialisiert. Wenn ihr den Sketch auf den ESP32 übertragt, erscheint das Adafruit-Logo auf dem OLED-Bildschirm. Bleibt er jedoch wider Erwarten schwarz, ändert die I²C-Adresse von 0x03C auf 0x03D und überprüft noch einmal eure Kabel. Achtet in jedem Fall auf die Polung. Bei manchen Displays sind VCC und GND vertauscht.

Möchtet ihr lieber mit dem ILI9341-Farbdisplay starten, benötigt ihr einen ähnlichen Aufbau zum Start. Da bei diesem Display kein Adafruit-Logo zu Beginn angezeigt wird, habe ich die Funktion drawCircle() im void setup() eingefügt, damit man sieht, ob das Display richtig funktioniert.

```
#include <SPI.h>
#include <Adafruit_ILI9341.h>
#include <Adafruit_GFX.h>
```

Poti als Drehregler

Das eingangs erwähnte Poti verbinde ich über den mittleren Pin mit dem Analog-Eingangs-Pin 34 des ESP32. Damit der Mikrocontroller weiß, wo der Poti angeschlossen ist, muss der Sketch vor dem void start() noch um zwei Zeilen ergänzt werden. Außerdem speichere ich den Wert in der Variablen potValue.

#define potPin 34
int potValue;

}

Im void loop() liest die Funktion analog-Read() den Poti-Wert zwischen 0 und 4095 ein und überträgt ihn mit der map-Funktion auf eine Skala mit 100 Werten, wobei die Ø als Wert immer mitgezählt wird.

Auch wenn wir die Variable derzeit noch nicht verwenden, ist der Poti jetzt schon einsatzbereit und wir können die Variable potValue im Laufe unserer Versuche beliebig anpassen.

Adafruit GFX Library

Mit der Adafruit GFX Library kann man komfortabel einfache bis komplexe Grafiken 3



Binfache geometrische Formen lassen sich über ein paar Parameter schnell auf den Bildschirm zaubern.

auf allen Displays darstellen, die Adafruit in seinem Online-Shop anbietet. Aber auch Bildschirme von anderen Herstellern können funktionieren, sofern sie baugleich sind. Diese flächendeckende Kompatibilität vereinfacht den Wechsel zwischen verschiedenen Bildschirmtypen und Gelerntes lässt sich leicht übertragen.

Um mit dem Bildschirm zu interagieren, benötigen wir das display-Objekt, das wir am Anfang unseres Sketches definiert haben. Jeder Befehl beginnt mit display, gefolgt von einem Punkt und der anschließenden Funktion, z.B. display.drawPixel(). Für die vereinfachte Darstellung (außer bei Listings) habe ich in den folgenden Abschnitten das display-Objekt weggelassen, sodass ihr es euch vor jeden Befehl dazu denken müsst.

Die Arbeitsfläche

Das Koordinatensystem beginnt in der linken oberen Bildschirmecke und bewegt sich entlang der X-Achse nach rechts und auf der Y-Achse nach unten. Je nachdem, wie ihr das Display in ein Projekt einbaut, lässt sich das Koordinatensystem mithilfe von setRotation() in 90-Grad-Schritten drehen und ermöglicht beispielsweise auch Layouts mit einem Display im Hochformat. Setzt für eine Rotiation Werte von Ø (0 Grad) bis 3 (270 Grad) in die Klammer. Das Drehen ist sogar während eines laufenden Programms möglich.

Bevor ihr mit dem Zeichnen beginnt, solltet ihr, wie auf einer Tafel, zunächst das display-Objekt reinigen. Dafür verwendet das SSD1306 den Befehl clearDisplay() und beim ILI9341 lautet er fillScreen(c), wobei hier in der Klammer noch die Farbe (c) eingetragen werden muss, mit der ihr den Bildschirm füllen möchtet (mehr dazu im Abschnitt Farbe). Danach könnt ihr mit dem Zeichnen beginnen. Verwendet ihr das OLED-Display, werden die Grafiken erst auf dem Bildschirm angezeigt, wenn ihr den Befehl display() ausführt.

Da alle Grafiken nacheinander gezeichnet werden, können sie sich überlappen. Einmal gezeichnet, bleiben Grafik-Elemente nämlich so lange auf dem Display, bis sie von etwas anderem überlagert werden oder der Bildschirm gelöscht bzw. gefüllt wird. Um beispielsweise bei einem Zähler sicherzustellen, dass die Ziffern sich nicht überlagern, ist eine Ressourcen schonende Variante, die alte Zahl mit einer kleinen geometrischen Form zu überzeichnen, z.B. einem schwarzen Kreis oder Rechteck.

Farbe

Je nachdem, welches Display ihr verwendet, kann die Farbdarstellung von 1 bis 16 Bit reichen, von schwarz-weiß über Graustufen bis knallbunt. Mehr als 16 Bit Farbtiefe (65.536 Farben) geht aber nicht, denn Adafruit möchte möglichst viele Bildschirme unterstützen und nutzt dafür das RGB565-Format anstatt RGB888. Der Name RGB565 ergibt sich aus den 16 Bits, mit denen die Farben auf dem Display ausgegeben werden, bestehend aus $5 \times \text{Rot}$, $6 \times \text{Grün und } 5 \times \text{Blau}$ – im Vergleich dazu verwendet RGB888 für jede Farbe 8 Bits, also insgesamt 24 Bits, d.h. etwa 16 Millionen Farben. Hier muss also eine Umrechnung stattfinden, wenn wir eigene Farben verwenden wollen. Mit einem hilfreichen Online-Tool geht das ganz beguem von RGB nach RGB565 (siehe Link in der Kurzinfo). Wem das zu kompliziert ist, der kann auch Farben verwenden, die Adafruit freundlicherweise vordefiniert hat

Da ein SSD1306-Display die Pixel nur einund ausschalten kann, lassen sich hier 1 und 0 verwenden. Beim ILI9341 gibt es ein paar mehr Möglichkeiten: Insgesamt stehen 19 vordefinierte Farben zur Verfügung, wenn man einen Blick in die ILI9341-Bibliothek wirft (siehe Tabelle).

Habt ihr euch einen eigenen Farbton erstellt, gebt ihm am besten mit der folgenden Zeile einen eigenen Namen, damit ihr besser damit arbeiten könnt, z.B. so:

#define ILI9341_MEINE_FARBE 0x0339

Grafische Werkzeuge

Pixel: Das kleinste Werkzeug, mit dem wir die Display-Fläche bearbeiten können, ist der Befehl drawPixel(x,y,c). In der Klammer geben wir lediglich die X- und Y-Koordinaten sowie die Farbe (c) an.

Linien: Um Linien zu zeichnen, bietet Adafruit gleich drei Funktionen an, die auf unterschiedlichen Algorithmen basieren. Mit drawLine(x,y,x1,y1,c) könnt ihr Linien in alle Himmelsrichtungen zeichnen. Gebt dafür den Start- und Endpunkt sowie die Farbe an. Die schnelleren Funktionen drawFastHLine(x,y,1,c) und drawFastVLine(x,y,1,c) zeichnen nur horizontale oder vertikale Linien. Adafruit verwendet sie z.B. zum Füllen von Flächen. Anstatt eines Endpunktes muss hier eine Länge (1) angegeben werden.

Dreiecke: Wer Dreiecke auf dem Bildschirm benötigt, kann sie mit drawTriangle(x,y,x1,y1,x2,y2,c) als Kontur oder mit fillTriangle(x,y,x1,y1,x2,y2,c) gefüllt auf dem Bildschirm darstellen. Auf der Logik der Linien basierend ersetzt man die X- und Y-Werte mit beliebigen Koordinaten und trägt die gewünschte Farbe ein.

Zwei Dreiecke lassen sich schnell zu einem Viereck oder noch komplexeren Formen kombinieren, die flexibel im Raum positioniert werden können. Ein sich kreisförmig füllender Ladebalken lässt sich z.B. mit dieser Methode umsetzen, wie ich später noch zeigen werde. **Rechtecke:** Auch diese lassen sich konturiert oder gefüllt darstellen. Ihr Ausgangspunkt befindet sich in der linken oberen Ecke. Gezeichnet werden sie nach rechts unten. Daher benötigen die Funktionen drawRect(x,y, w,h,c) und fillRect(x,y,w,h,c) nur die Start-Koordinaten, die Breite (w), Höhe (h) und Farbe als Parameter. Einziger Nachteil: Sie lassen sich nur rechtwinklig abbilden.

Abgerundete Rechtecke kann man mit drawRoundRect(x,y,w,h,r,c) und fillRound-Rect(x,y,w,h,r,c) zeichnen. Der Parameter r steht für den Rundungs-Radius der vier Ecken. Richtig rund sehen diese allerdings erst ab einem Radius von 12 aus. Vorher sind sie eher im 45-Grad-Winkel abgefast.

Kreise: Um sie auf dem Display zu platzieren, muss man ein wenig umdenken, denn der Ursprungspunkt der Funktionen draw-Circle(x,y,r,c) und fillCircle(x,y,r,c) befindet sich in der Mitte des Kreises und nicht links oben. Der Radius gilt ab einem Pixel neben dem Mittelpunkt. Das hat zur Folge, dass jeder Kreis einen ungeraden Durchmesser besitzt. Buchstaben, die eine ungerade Breite und Höhe haben (z.B. 3×5 Pixel), lassen sich dadurch aber optimal in einem Kreis zentrieren.

Text

Möchtet ihr Text auf dem Display anzeigen, um beispielsweise exakte Messwerte auszugeben, könnt ihr das genauso mit print() und println() tun wie bei der Ausgabe am seriellen Monitor, nur dass ihr nicht Serial.print(), sondern display.print() dafür nutzen müsst. Die Parameter für Schriftgröße, -farbe und Ausrichtung sind nicht Teil des print-Befehls und müssen daher vorher über separate Funktionen eingestellt werden.

Adafruit hat in der GFX Library standardmäßig eine Schriftart integriert, die Zeichen mit 5 × 7 Pixeln erzeugt. Ihr Ursprungspunkt befindet sich wie gewohnt links oben und wird mit dem Befehl setCursor(x,y) festgelegt. Stellt danach die Schriftgröße mit setTextSize(m) ein und wählt eine Zahl als Multiplikator (m) aus, wobei 1 die Standardgröße ist. Mit setTextColor(c) bestimmt ihr schließlich noch die Textfarbe. Sollen die Zeichen einen Hintergrund (bgr) erhalten, könnt ihr dafür auch einen zweiten Wert in der Klammer hinzufügen, sodass der Befehl setTextColor(c,bgr) lautet – das funktioniert allerdings nur mit der Standard-Schrift.

In jedem Fall solltet ihr in Erwägung ziehen, den Textumbruch zu deaktivieren. Wenn ihr nämlich Texte frei platziert, kann es schnell passieren, dass diese zu nah an den rechten Rand geraten und umbrechen. Nutzt den Befehl setTextWrap(false) um das zu verhindern, aber bedenkt, dass Texte jetzt auch aus dem Bild laufen können.



Pixel-Schriften im Größen-Vergleich: (von oben) Standard-Schrift, Org_01, TomThumb und Picopixel

Andere Schriften

Weitere Schriften lassen sich genauso in euren Code einbinden, wie die Bibliotheken selbst. Adafruit liefert zum Glück auch schon ein paar Schriften mit der GFX Library, die ihr mit zwei Zeilen Code verwenden könnt. Ihr findet sie in eurem Arduino-Ordner unter libraries/ Adafruit GFX Library/Fonts. Auf den kleinen OLED-Displays funktionieren vor allem die Pixel-Schriften Picopixel, TomThumb und Org_014. Verwendet ihr das größere Display, reicht der Platz auch für die Schriftfamilien FreeMono (wie Courier), FreeSans (wie Arial) und FreeSerif (wie Times New Roman). Diese solltet ihr jedoch nicht skalieren, sondern eine der Schriftgrößen verwenden, die Adafruit anbietet - sieht schöner aus.

Um eine zusätzliche Schrift (hier Picopixel) in euer Projekt einzubinden, ergänzt folgende Zeile zu Beginn eures Programmcodes:

#include <Fonts/Picopixel.h>

und aktiviert die Schrift anschließend an einer Stelle eurer Wahl mit

setFont(&Picopixel);

bevor ihr den print-Befehl ausführt. Möchtet ihr danach an einer anderen Stelle etwas in der Standard-Schrift schreiben, könnt ihr mit setFont() dorthin zurückwechseln, indem ihr die Klammer leer lasst oder zu einer weiteren Schrift wechseln, die ihr im Vorfeld eingebunden habt. Denkt daran, dass auf dem OLED-Display Text erst mit dem display-Befehl auf dem Bildschirm erscheint.

Komplikationen mit Text

Wer das Platzieren von Objekten bis hierhin logisch fand, wird sich leider bei Texten mit selbstgewählten Schriftarten die Haare raufen. Plötzlich befindet sich der Ursprungspunkt nämlich nicht links oben, sondern links unten oder irgendwo dazwischen. Das fällt einem spätestens dann auf, wenn man setCursor(0,0) eingibt und anstatt des vollständigen Textes nur eine einzelne Pixelreihe erhält. Wollt ihr einen Text mit alternativer Schrift wie gewohnt ausrichten, müsst ihr ihn um seine Höhe nach unten versetzen. Um diese zu ermitteln, beinhaltet die Adafruit GFX Library die Funktion getTextBounds(). Mit ihr wird ein virtueller Rahmen um ein Textfeld gezeichnet, der die Breite, Höhe und die Koordinaten des Endpunktes zurückgibt. Vor allem bei Schriften, deren Größe im Dateinamen in Punkt (pt) angegeben wird, kann diese Funktion sehr hilfreich sein, da wir ihre Höhe in Pixel benötigen.

Verwendet folgende Zeilen, um getText-Bounds() zu auszuführen:

int16_t x1, y1; uint16_t w, h;

Setzt einen Text eurer Wahl als String ein (z.B. "Hello") **5** und bestimmt die X- und Y-Koordinaten des Ursprungspunktes. Mit x1 und y1 als End-Koordinaten, der Breite und Höhe könnt ihr euren Text anschließend besser platzieren oder ein Rechteck zeichnen, um z.B. einen alten Wert zu löschen, bevor der neue ausgegeben wird.

Text einfacher ausrichten

Um Texte leichter und vor allem flexibler platzieren zu können, habe ich zwei Funktionen geschrieben: standardFontAlign() und customFontAlign(). Mit ihnen könnt ihr Texte links-, rechtsbündig oder zentriert ausrichten, ohne das Verschieben der Koordinaten manuell berechnen zu müssen. Das geht sowohl mit der Standard-Schrift als auch mit anderen Schriften, wobei die Zeichen der FreeFonts teilweise sehr stark umhertanzen und sich nicht zuverlässig platzieren lassen. Dann bleibt euch nichts anderes übrig, als die X- und Y-Koordinaten ein wenig nachzujustieren.

Da standardFontAlign() und custom-FontAlign() auf Funktionen der Adafruit GFX Library zurückgreifen und in ihnen das display-Objekt angesprochen wird, müsst ihr display diesmal nicht vor die Funktion schreiben. Verwendet also nur den Befehl standardFontAlign() oder customFont-Align(). In die Klammer müsst ihr anschließend ein paar Parameter eingeben. Welche das sind, zeige ich euch exemplarisch anhand der Standard-Schrift, da beide Funktionen dieselben Parameter nutzen.

Beginnt mit dem String eurer Wahl. Der Parameter origin 6 bietet 9 verschiedene Ursprungspunkte, mit denen ihr euren Text platzieren könnt. Befindet er sich links, läuft der Text nach rechts und umgekehrt. Ist der origin mittig platziert, bewegt sich der Text in beide Richtungen nach außen. Bestimmt



5 Wenn ihr alternative Schriften verwendet, verschiebt sich der Ursprungspunkt (links: Standard-Schrift, rechts: Picopixel).



6 Dank standardFontAlign() und customFontAlign() lassen sich Texte besser platzieren. Als Ursprungskoordinaten stehen 8 Punkte zur Verfügung.

danach die X- und Y-Koordinaten, an die der Text mithilfe des origin auf dem Display platziert werden soll. Abschließend fehlen nur noch die Skalierung und die Farbe. Eine Hintergrundfarbe lässt sich hier nicht einstellen, ich habe aber eine weitere Variante der Funktionen geschrieben, in denen ein Rechteck den alten Wert löscht. Ihr findet den Code im GitHub-Repository.

Balkenanzeige programmieren

Wenn ihr visualisieren möchtet, wie sich ein Wert innerhalb eines festgelegten Rahmens verändert, bietet sich eine Balkenanzeige an. Zu den bekanntesten zählen vermutlich das Thermometer und die Windows-Statusbalken. Mit ihnen kann man schnell erkennen, ob man sich näher an einem Minimal- oder Maximalwert befindet.

Mit drei Komponenten lässt sich so ein Balken schnell programmieren 7. Als Erstes definiere ich den Rahmen (in diesem Fall ohne Skala). Als Höhe verwende ich 12 Pixel und als Breite nutze ich 60 Pixel. Damit der Rahmen mittig sitzt, muss ich die Bildschirmbreite und -höhe halbieren, davon die Hälfte der Balkenbreite und -höhe sowie einen Pixel abziehen, weil die 0 mitgezählt wird. Für das OLED-Display sieht die Berechnung dann wie folgt aus:

int x = 128/2 - 60/2 - 1; int y = 64/2 - 12/2 - 1; Im void loop() schreibe ich danach

display.drawRect(x, y, 60, 12, 1);

Wenn ihr das IL19341 verwendet, gebt als letzten Wert in der Klammer anstatt der 1 z.B. IL19341_WHITE an. Als Nächstes zeichne ich einen Löschbalken, der den Rahmen mit jedem Frame leert, bevor der eigentliche Balken gezeichnet wird. Da ich den Rahmen selbst nicht überschreiben will, muss ich den Löschbalken um 1 versetzen und um 2 auf beiden Achsen kürzen.

Danach platziere ich den eigentlichen Balken im Rahmen und lasse zu allen Seiten einen Pixel Freiraum. Außerdem verknüpfe ich die Länge des Balkens mit unserer Poti-Variable potValue.

In der map-Funktion ändere ich den Wert noch von 99 auf 56, weil unser Balken 56 Pixel breit ist, aber die Ø mitgezählt wird. Danach übertrage den Sketch auf den ESP32 und drehe an dem Poti.

Jetzt geht's rund

Wenn wir schon einen Drehregler angeschlossen haben, wäre es doch toll, dessen Position mithilfe einer kreisförmigen Balkenanzeige auszugeben. Der prinzipielle Aufbau ist ähnlich zu dem vorherigen Beispiel, nur gibt es für kreisförmige Balken keine vorgefertigte Funktion in der Adafruit GFX Library. Also müssen wir ein wenig in der Trigonometrie-Trickkiste kramen und die Werkzeuge kombinieren, die uns zur Verfügung stehen. Ladet euch am besten den Sketch aus dem GitHub-Repository, um die Erklärungen besser nachvollziehen zu können.

Von den verschiedenen Möglichkeiten, wie sich so ein Projekt umsetzen lässt, hat bei mir am besten funktioniert, den Kreis in Segmente zu unterteilen (8), die ihre Farbe wechseln



Oben die Balkenanzeige, darunter die Bestandteile einzeln: Rahmen, gefülltes Löschrechteck und der eigentliche Balken

können. Abhängig davon, welchen Wert das Poti an den Mikrocontroller übermittelt, werden die Segmente entweder weiß oder schwarz gefüllt und lassen den Kreis dadurch mehr oder weniger geschlossen erscheinen. Für die Darstellung der Segmente können wir zwei Dreiecke verwenden, die zusammengesetzt ein Trapez ergeben. Jetzt gilt es nur noch, die notwendigen Koordinaten zu ermitteln. Damit alles schön ordentlich bleibt, habe ich für diesen Zweck die Funktion bar() geschrieben, die für die Berechnungen ein paar Parameter benötigt.

Anders als im vorherigen Beispiel wird die Balkenlänge nicht durch die Bildschirmbreite begrenzt, sondern durch die maximalen

Studiengang



Vertiefungsrichtungen im Masterstudiengang:

- Elektrische Energietechnik
- Quantum Engineering
- Autonomes und vernetztes Fahren
- Künstliche Intelligenz
- Internet der Dinge
- Energieinformatik. Mai



TECHNISCHE UNIVERSITÄT

ILMENAU

Studienfachberatung: Univ.-Prof. Dr. rer. nat. Jochen Seitz jochen.seitz@tu-ilmenau.de

www.tu-ilmenau.de/eit-studieren

Segment-Koordinaten berechnen

```
void bar(int xCen, int yCen, int r, int color, int potValue){
   int segLength = r/5;
   int rInner = r - r/5;
   for (int i = 0; i < range; i += segLength){</pre>
      //Segment Start
      float sx = cos(radians(i - (range/2 + 90)));
float sy = sin(radians(i - (range/2 + 90)));
      //Segment Ende
      float ex = cos(radians(i - (range/2 + 90) + segLength));
float ey = sin(radians(i - (range/2 + 90) + segLength));
      //Segment Startkoordinaten
      int x = sx * rInner + xCen;
      int y = sy * rInner + yCen;
int x1 = sx * r + xCen;
      int y1 = sy * r + yCen;
      //Segment-Endkoordinaten
      int x2 = ex * rInner + xCen;
int y2 = ey * rInner + yCen;
int x3 = ex * r + xCen;
      int y3 = ey * r + yCen;
      if (i < potValue) {</pre>
        display.filTriangle(x, y, x1, y1, x2, y2, ILI9341_WHITE);
display.filTriangle(x1, y1, x2, y2, x3, y3, ILI9341_WHITE);
        else {
         display.fillTriangle(x, y, x1, y1, x2, y2, ILI9341_BLACK);
         display.fillTriangle(x1, y1, x2, y2, x3, y3, ILI9341_BLACK);
   }
}
```

360 Grad eines geschlossenen Kreises. Dafür habe ich die globale Variable range erstellt. Mit ihr kann man den Bewegungsrahmen der Kreisbalken bestimmen. Setzen wir den Wert unter 360 Grad, können wir den Kreis sogar auftrennen.

Für unsere Berechnungen fehlen noch ein paar weitere Parameter: der äußere Radius (r), ein innerer Radius (rInner), der die Segmentdicke bestimmt, und die Segmentlänge (segLength). Haben wir diese festgelegt, berechnen wir anschließend mithilfe einer for-Schleife die Koordinaten für die einzelnen Kreis-Segmente. Diese verwendet segLength als Schrittweite und i für den aktuellen Winkel innerhalb von range. Mithilfe der cos- und sin-Funktionen errechne ich anschließend die Positionen sx und sy auf einem Einheitskreis. Da die Arduino IDE einen Radiant (Bogenmaß) als Wert benötigt, rechne ich den Winkel i mithilfe der radians-



9 So sieht das fertige Ergebnis aus. In Funktionen verpackt lassen sich auch ganz einfach mehrere Kreise auf einem Bildschirm anzeigen. Fehlt nur noch eine Ziffer in der Mitte.



8 Die Kreissegmente entstehen aus zwei gefüllten Dreiecken. Dafür müssen wir zunächst vier Koordinaten auf dem Kreis herausfinden.

Funktion um. Damit die Öffnung des Kreissegments nach unten statt nach rechts zeigt, ziehe ich die Winkel range/2 und 90 von i ab.

Um die Segment-Startkoordinaten x, y, x1 und y1 zu erhalten, multipliziere ich sx und sy jeweils mit r bzw. rInner und versetze diese Werte mit xCen und yCen dorthin, wo ich den Mittelpunkt der kreisförmigen Anzeige haben möchte. Fehlen jetzt noch die Koordinaten, die eine Segmentlänge weiter vorne liegen. Dafür erstelle ich ex und ey, zu denen ich als Winkeloffset den Wert segLength addiere. Mithilfe von ex und ey errechne ich abschließend x2, y2, x3 und y3. Danach sind die Segmente bereit, gezeichnet zu werden. Das geschieht mithilfe einer if-Bedingung, die anhängig von potValue steuert, welche Segmente angezeigt (weiß) und welche unsichtbar (schwarz) sein sollen.

Damit optisch auch auf einem Monochrom-Display erkennbar ist, wie weit sich der Kreisbalken bewegen kann, zeichnet die Funktion border() im void start() einen dünnen Rahmen ober- und unterhalb des Balkens (9). Sie funktioniert genauso wie bar(), nur dass die Segmente viel flacher sind. Mithilfe der Variablen brHeight und brSpace könnt ihr die Dicke und den jeweiligen Abstand zum Balken einstellen. Nutzt ihr ein Farbdisplay, benötigt ihr die Funktion nicht unbedingt, denn ihr könnt in bar() das Schwarz der unsichtbaren Balken z.B. gegen ein dunkles Grau abändern.

Spielt mit den Funktionen und Werten. Mit den hier gezeigten Funktionen könnt ihr schon eine Menge schicker Anzeigen programmieren. Im kommenden Heft zeige ich euch, wie man die Anzeige auf den Displays mit eingebundenen Bitmaps zusätzlich aufpeppen kann. —*akf*

Für Maker und Nerds

shop.heise.de/shop-maker



ParkLite

ParkLite denkt mit. Die elektronische Parkscheibe stellt automatisch nach ca. 20 Sekunden die Parkzeit ein. Damit ist Schluss

mit Bußgeldern! Hitze- und kältebeständig, inklusive Reinigungstuch und Klebepads.

29,90 €

NEU





Aluminium-Case FLIRC

hochwertige Gehäuse aus stabilem Aluminium ist ideal, um den Raspberry

Pi 4 als Media Center zu verwenden. Das elegante Design integriert sich optimal in jede Wohnumgebung. Auch im Set mit Raspi 4 Model B 2GB erhältlich.

23,90 €

musegear® finder Version 2

Finden Sie Schlüssel, Handtasche oder Geldbeutel bequem wieder statt ziellos zu

suchen. Mit dem Finder können Sie z.B. das Smartphone klingeln lassen oder Wertgegenstände einfach tracken und noch mehr.

24.90 €



PokitMeter -Multimeter, Oszilloskop und Logger

PoKit misst, zeigt und protokolliert eine Vielzahl von Parametern wie Spannung, Strom, Widerstand und Temperatur mittels Verbindung via Bluetooth mit Ihrem Smartphone oder Tablet.



Raspberry Pi Pico Starter Kit

Mit Zubehör: Board, Micro-USB-Kabel, Stiftleisten, Breadboard, Jumperkabel.

Raspberry Pi Pico ist ein kostengünstiges, leistungsstarkes Mikrocontroller-Board mit flexiblen digitalen Schnittstellen.

19,90 €



27.70 € 19,90 €

Make Family + **Makey-Paket**

Das PDF-Magazin mit 21 Anleitungen zum kreativen Basteln mit Kids auf über 200 Seiten. Dazu: Makey-Plüschroboter und Makey-Lötbausatz mit LEDs und Batterie.

109,90 €



Joy-IT OR750i: Freifunk- & **OpenWrt-Dual-Band-Router**

Der Einstieg in die Freifunk- und OpenWrt-Welt kann oft schwierig sein. Deshalb hat Joy-IT in Zusammenarbeit mit Freifunk Hannover und c't den OR750i entwickelt.

Dank Webinterface kann man beliebige Firmwares einfach hochladen - ohne komplizierte Kommandos oder inkompatible Hardware-Revisionen; ideal für OpenWrt-Einsteiger und solche, die Freifunk einfach nur nutzen wollen.

39,90 €



REINER SCT Authenticator

Der REINER SCT Authenticator speichert die elektronischen Schlüssel für die Logins sicher in seiner Hardware und generiert die TOTP-Einmalpasswörter hochgenau alle 30 Sekunden. Er arbeitet ohne Internetverbindung und kann deshalb online nicht angegriffen werden. Zusätzlich kann seine Funktion noch mit einem PIN-Schutz abgesichert werden.

39,90 €



17,90 €

"No Signal" Smartphone-Hülle

Passend für Smartphones aller Größen bis 23cm Länge blockt diese zusammenrollbare Hülle alle Signale von GPS, WLAN, 3G, LTE, 5G und Bluetooth, sowie jegliche Handy-Strahlung.

Versilbertes Gewebe im Inneren der Tasche aus recycelter Fallschirmseide bildet nach dem Schließen einen faradayschen Käfig und blockiert so alle Signale.

39,90 €

shop.heise.de/shop-maker

PORTOFREI AB 20 € BESTELLWERT



SELLER

BEST-

Tasse "Kein Backup? Kein Mitleid!"

Wer kennt es nicht? Die lieben Kollegen haben wieder mal die Datensicherung vergessen und betteln bei Ihnen in der IT-Abteilung um Hilfe. Sie denken sich dabei nur: "Kein Backup? Kein Mitleid!" Platzieren Sie die schicke Keramiktasse, außen matt-

schwarz und innen rot, einfach demonstrativ auf Ihrem Schreibtisch. Der praktische Holzdeckel kann gleichzeitig als Untersetzer verwendet werden und verhindert somit nervige Kaffeeflecken auf dem Schreibtisch.

Fluoreszenzdisplays

Retro oder Mehrwert? Vakuum-Fluoreszenzdisplays (VFDs) sind hell, blickwinkelunabhängig und lassen sich hervorragend ablesen – man findet sie deshalb oft in hochwertigen Messgeräten, Musikelektronik und HiFi-Equipment. Wir zeigen an einer Beispielanwendung, wie man sie einsetzt und was bei der Ansteuerung zu beachten ist.

von Carsten Meyer


Die ersten Vakuum-Fluoreszenzdisplays wurden 1967 von der japanischen Firma Noritake Itron Corp. vorgestellt - als Ersatz für die voluminösen und (schon damals!) teuren Nixie-Ziffernanzeigeröhren, deren Hochspannungsversorgung in den zunehmend transistorisierten Geräten problematisch wurde. Während die Neon-gefüllten Nixies wie Kaltkathoden-Glimmlampen funktionierten, besann man sich bei den VFDs wieder auf das Prinzip der Elektronenröhre: Auch in einem VFD findet man wie in einer Triode eine beheizte Kathode, ein Steuergitter und eine Anode, hier natürlich als Leuchtschicht ausgeführt.

Die Leuchtschicht emittiert Photonen, wenn sie mit Elektronen bombardiert wird – vergleichbar mit dem "magischen Auge" in alten Röhrenradios oder auch dem Bildröhren-Prinzip. Die Elektronen stammen aus einer Wolke, die einen beschichteten, stromdurchflossenen Heizfaden umgibt; sie werden von einem positiven Potential auf der Anode angezogen und prallen dann lichtwandelnd auf die Leuchtschicht. Um einen nennenswerten Stromfluss zustande zu bringen, muss die Anode eine mit 12 bis 30V (für heutige Schaltungsverhältnisse) recht hohe Spannung führen.

Damit wäre das Display eigentlich schon komplett, wenn man die Anode auf einzelne Segmente (etwa in Form der bekannten Siebensegment-Anzeige) aufteilt. Nachteilig ist dann allerdings, dass man bei mehrstelligen Anzeigen für jedes Segment einer jeden Ziffer einen eigenen Schalttransistor braucht; ein bauteilsparendes Multiplexen ist aufgrund der thermischen Trägheit des Heizfadens nicht möglich.

In der praktischen Ausführung müssen die Elektronen vor dem Auftreffen auf der Anoden-Leuchtschicht deshalb noch ein feines

Kurzinfo

- » Ansteuerung von Fluoreszenzdisplays: Heizung, Gitter und Anoden
- » Erzeugen von Heiz- und Anodenspannung
- » Fluoreszenzdisplay als Verstärker (Röhren-Triode)
- » Beispielanwendung: Rentenuhr mit ATmega328



Gitter passieren. Liegt dieses auf gegenüber dem Heizfaden leicht negativem Potential (z.B. -1V), drehen die Elektronen unmotiviert wieder um, und es kann kein Strom fließen. Liegt dagegen eine gegenüber dem Heizfaden positive Spannung an, werden die Elektronen zur Anode beschleunigt und bringen sie zum Leuchten.

Gitter-Ansteuerung

Im Unterschied zur mit sehr viel höheren Anodenspannungen arbeitenden Röhren-Triode darf das Steuergitter von VFDs nämlich durchaus eine positive Spannung aufweisen; dies ist der niedrigen Anodenspannung geschuldet. Tatsächlich kommt hier der Elektronenfluss erst bei einigen Volt so richtig in Schwung. Ein VFD-Segment leuchtet also nur dann, wenn sowohl das Steuergitter als auch die zu leuchtenden Segmente mit Spannung versorgt werden.

Der Einfachheit halber steuert man das Gitter meist mit einer Spannung an, die gleich hoch wie die Anodenspannung ist. Mit 24V kann man hier nicht viel verkehrt machen,



Einfaches VFD von Futaba, das der Versender Pollin vor einiger Zeit für weniger als einen Euro im Programm hatte: Bei genauem Hinsehen erkennt man hier drei Heizfäden und die Gitter vor jeder Ziffer.



VFD-Prinzip: Man erkennt die typischen Bestandteile einer direkt geheizten Röhren-Triode wie Heizfaden (F) und Gitter (G), hier aber mit einer segmentierten Anode.







Die russischen IV4 und IV-17 (letztere ist etwas robuster) besitzen insgesamt 18 Segmente und können auch alphanumerische Zeichen darstellen. Die gleiche Ausführung mit 7 Segmenten heißt IV-11.



Das Multiplexen funktioniert mit VFDs ähnlich wie bei LED-Siebensegmentanzeigen: Jede Ziffer erhält ihre Daten über die angelegten Anodenspannungen genau dann, wenn sie mit einem positiven Gitter-Impuls eingeschaltet wurde.

Jahrelang führten VFD-Röhren wie diese niedliche IV-22 ein Schattendasein und wurden fast verschenkt - inzwischen haben sie aber zu den Nixies aufgeschlossen. Die benötigte Fassung kann man auch durch einzelne Kelchfeder-Kontakte substituieren.

auch wenn dann viele Elektronen auf dem Gitter landen und sich ein gewisser Gitterstrom einstellt, der wie der Segment-Anodenstrom im Bereich weniger Milliampere liegt. Die Gitter-Steuerung erfolgt also im Gegensatz zur Elektronenröhre nicht leistungslos.

Heizfaden

Der Heizfaden bildet wie bei einer direkt geheizten Elektronenröhre die Kathode; die zum Erwärmen nötige Leistung ist relativ bescheiden, meist kommt er mit 1 bis 2 Volt und einigen zehn Milliampere Strom aus. In der Regel versorgt man ihn über einen passend bemessenen Vorwiderstand aus der Logik-Betriebsspannung (5V).

Es ist vorteilhaft, wenn er gegenüber abgeschalteter Gitter (0-Pegel) leicht positiv vorgespannt wird: Dann "sieht" die Röhre gegenüber ihrer Kathode (Heizfaden) am Gitter eine negative Spannung und sperrt besser. Ansonsten wird man im dunklen Zimmer abgeschaltete Segmente doch schwach leuchten sehen – das kann bei einer Uhr nachts nerven.

In der Praxis geht man folgendermaßen vor: Den einen Anschluss des Heizfadens legt man an + 5V, der andere Anschluss geht über einen Vorwiderstand (Größenordnung einige zehn Ohm) an Masse. Dadurch ist sichergestellt, dass ein abgeschaltetes Gitter immer negativ gegenüber der Heizfaden-Kathode ist. Übrigens ist die Polarität der Heizspannung egal, in vielen Anwendungen (z.B. alte Taschenrechner) verwendet man die Wechselspannung aus einer Anzapfung des Übertragers für die Anodenspannungs-Erzeugung.

Um die bestmögliche Lebensdauer zu erreichen, sollte man sich an die Angaben im Datenblatt des VFDs halten; geringe Abwei-



Sehr retro, sehr cool: VFDs lassen sich nahezu durchsichtig fertigen.

chungen sind unkritisch. Ist ein Datenblatt nicht verfügbar, muss man die Heizspannung mit einem einstellbaren Labornetzteil ermitteln: Spannung an den Heizfaden anlegen und so lange erhöhen, bis er ganz schwach dunkelrot leuchtet, dazu die Zimmerbeleuchtung stark dämpfen. Notieren Sie dann die eingestellte Spannung und den aufgenommenen Strom, um den Vorwiderstand bemessen zu können.

Bei vielen VFDs kann man auch ohne Datenblatt die Anschlüsse durch das Glas hindurch ermitteln; im Zweifel hilft ein Multimeter im Ohm-Bereich: Zwischen den Heizspannungs-Anschlüssen ist ein Widerstand von einigen Ohm zu messen. Manchmal gibt es noch einen zusätzlichen Anschluss für eine Abschirm-Elektrode, die man über einen Vorwiderstand (Größenordnung 10kOhm) an die Anodenspannung legt. Das ist das Schöne an VFDs und Elektronenröhren: Man kann sehen, wie sie funktionieren!

Anodenspannung

Sicher wird man für ein VFD keinen eigenen Transformator vorsehen wollen, um die 24V für die Anoden- und Gitteransteuerung zu gewinnen. Ein einfacher Step-up-Schaltwandler tut es auch (siehe Schaltbild auf Seite 13). Wir haben hier den billigen MC34063 verwendet, der sich angesichts der relativ geringen Gesamt-Stromaufnahme des VFDs (selten mehr als 50mA) eher langweilt.

Über die Anodenspannung stellt man, falls etwa für eine Uhr benötigt, auch die Helligkeit des Displays ein. Sichtbares Leuchten erhält man ab etwa 9V, eine besonders helle Darstellung bei Spannungen von 24 bis 30V. Wie alle Elektronenröhren dürfen auch VFDs kurz überlastet werden, ohne Schaden zu nehmen; eine dauerhaft überhöhte Anodenspannung ist allerdings der Lebensdauer abträglich. In unserer Uhren-Schaltung (dazu später) sorgt übrigens ein (optionaler) LDR, also ein lichtempfindlicher



Alphanumerische Punktmatrix-VFDs sind in der Regel mit einem eingebauten Controller ausgerüstet. Das kleine Display vorn wird wie eines mit HD44780-Controller angesteuert, ist also kompatibel mit vielen LCDs. Das hintere stammt aus einem Kassensystem, es erwartet serielle Daten. Die Schnittstellenparameter lassen sich über Jumper einstellen.

Widerstand, für eine an die Umgebungshelligkeit angepasste Anodenspannung.

Multiplexen

Mit der Erkenntnis, dass sich eine einzelne Ziffer über das Steuergitter komplett einund ausschalten lässt, ist es zum leitungs- und transistorsparenden Multiplex-Betrieb nicht mehr weit. Man schaltet einfach "gleichnamige" Anoden-Segmente aller Ziffern parallel, inklusive Dezimalpunkt ergeben sich dann acht Leitungen. Bei einer sechsstelligen Anzeige sind dann nur 14 Treiberleitungen notwendig: Acht für die Anodensegmente und sechs für die Gitter.

Der steuernde Controller muss nun nur reihum die einzelnen Ziffern mit den zugehörigen (nach dem Siebensegment-Prinzip dekodierten) Daten versorgen, wobei nur das



Der VFD-Treiberschaltkreis PT6315 erledigt das Multiplexen der Anzeige selbst. Er kann sogar zusätzlich eine Tastenmatrix aus bis zu 32 Tastern abfragen, falls benötigt. Gitter der gerade angesteuerten Ziffer eingeschaltet wird. Die Trägheit des Auges in Verbindung mit der Nachleuchtdauer der Fluoreszenzschicht sorgt dann für eine flackerfreie Anzeige, dafür genügen 100 Durchläufe pro Sekunde.

Mehrstellige VFDs werden immer gemultiplext, sie kommen deshalb auch mit einer bescheidenen Anzahl Anschlusspins aus. Auch im Multiplex-Betrieb lassen sich einzelne Ziffern heller (hervorgehoben) oder dunkler darstellen, indem man einfach die Anzeigedauer der Ziffern gegenüber dem Gesamt-Zyklus verlängert oder verkürzt. Davon macht unsere Uhr zum Stellen der Stunden und Minuten Gebrauch.

Gesucht, gefunden

Mehrstellige Ziffern-VFDs bekommt man manchmal geradezu nachgeworfen; das auf

Seite 8 unten abgebildete vierstellige Display erhielten wir von Pollin zum Preis von nicht einmal einem Euro. Bisweilen sind zusätzliche Symbole enthalten, die kaum stören werden – man lässt sie einfach unbeschaltet. Wir fanden auch Displays mit 14 oder 16 Segmenten zum Anzeigen von Ziffern und Buchstaben oder Punktmatrix-Anzeigen mit 5 × 7 Punkten pro Stelle, hier kommen dann doch schon etliche Pins zusammen.

VFD als Verstärker

Vor einigen Jahren machte eine Entwicklung der Musikelektronik-Firma Korg Schlagzeilen, die in Zusammenarbeit mit Noritake Itron eine neuartige Elektronenröhre zum Einsatz in Musikinstrumenten vorstellte. In diesem Bereich sind Röhren immer noch äußerst beliebt, weil damit aufgebaute Verstärker vor allem bei Übersteuerung ein ganz besonderes Oberwellenspektrum erzeugen: Nichtlineare Verzerrungen, die vor allem die zweite Harmonische (k2) und nur wenig ungerade Obertöne (k3, k5, k7 usw.) enthalten, erzeugen einen "warmen", als angenehm empfundenen Klangeindruck.

Die Korg NuTube basiert auf der von Noritake entwickelten VFD-Technik und wird auch von dieser Firma hergestellt. Im Prinzip ist die NuTube ein VFD, das nichts anzeigt, sich aber den verstärkenden Effekt der Gittersteuerung zu Nutze macht. Wie auch die VFDs selbst kommt das Bauteil mit relativ bescheidener Anodenspannung (12 bis 24V) aus, die Wärmeentwicklung ist vernachlässigbar.

Die NuTube enthält zwei Trioden, die je nach Versorgungsspannung um 14 bis 17dB verstärken; damit lässt sich ein Stereo-Vorverstärker aufbauen, oder man schaltet die Trioden für eine höhere Verstärkung hintereinander. Für einen optimalen Arbeitspunkt benötigt sie im Gegensatz zu üblichen Trioden eine leicht positive Gitter-Vorspannung, die Ansteuerung ist deshalb nicht leistungslos und muss für Röhren-Verhältnisse relativ niederohmig (< 50kOhm) erfolgen. Auch am Ausgang benötigt sie einen Spannungsfolger (Impedanzwandler, FET oder OpAmp), da der Anodenwiderstand für folgende Transistorstufen zu hochohmig ist.

Als einzelnes Bauteil kostet die NuTube rund 50 Euro, ihr Einsatz will deshalb wohlüberlegt sein. Die von Korg angegebene Referenzschaltung kann aber auch als Grundlage für eigene Entwicklungen dienen, die ganz normale VFD-Röhren verwenden: Man schaltet einfach alle Anodensegmente zu einer gemeinsamen



Korg/Noritake Itron

Die Doppeltriode NuTube von Korg ist im Prinzip ein VFD, das nichts anzeigt – die Ähnlichkeit zu einem "richtigen" Fluoreszenzdisplay ist keineswegs zufällig.

Anode zusammen, die restliche Schaltung kann dann (bis auf die Anpassung der Heizspannung) übernommen werden. Mit dem Trimmer VR1 stellt man den Arbeitspunkt der Röhre so ein, dass sich an der Anode etwa 12V (bzw. die halbe Betriebsspannung) einstellen – oder eben die gewünschten Verzerrungen.



Besonders schöne Einzelziffer-VFD-Röhren (zum Teil aus russischen Militärbeständen) gibt es bei zahlreichen Online-Shops, die sich auf Nixie- und Elektronenröhren spezialisiert haben. Leider sind bestimmte VFDs zur Zeit nur aus der aktuellen Kriegsregion zu beziehen. Hier muss jeder selbst entscheiden, ob ihm die Bastelei das wert ist.

Die abgebildeten Röhren fanden wir bei einem Händler aus der Ukraine, angesichts des Krieges ist wegen der nötigen postalischen Umwege allerdings mit etwas Lieferzeit zu rechnen. Während VFD-Röhren noch vor wenigen Jahren für ein paar Cent zu bekommen waren, liegen sie inzwischen wie die Nixies im Retro-Trend, und man muss schon mal fünf Euro für ein schönes Exemplar anlegen.

Komplettpakete

Kurz erwähnen wollen wir noch die mit eigener Ansteuer-Elektronik ausgerüsteten alphanumerischen VFDs. Auch die gibt es aus laufender Fertigung, im Vergleich zu den bekannten Punktmatrix-LCDs sind sie aber sehr teuer. Zusätzlich zum (meist HD44780-kompatiblen) Controller sind hier alle Treiberstufen und meist auch ein Step-up-Spannungswandler enthalten, der die Anoden- und Heizspannung generiert. Sie verhalten sich wie ein gängiges Punktmatrix-LCD, besondere Anpassungen der Schaltung oder der Software sind (bis auf die nun überflüssige Kontrast-Spannungseinstellung) nicht erforderlich.

Man erhält die "intelligenten" VFDs sogar in den gleichen Abmessungen wie die alphanumerischen LCDs – allerdings zum zehnfachen Preis, was sie für den Maker uninteressant macht. Trotzdem: Auch hier gibt es bei Restposten-Händlern manchmal lohnende Schnäppchen, die aufgrund von zusätzlichen, nun obsoleten Symbolen von der Industrie verschmäht wurden. Achten Sie aber darauf, dass sich zum Display oder zumindest zum verbauten Controller auch ein Datenblatt auftreiben lässt.

Treiben lassen

Immer, wenn Spannungen außerhalb des Versorgungsspannungsbereichs eines Logik-Designs zu schalten sind, wird es etwas knifflig – so auch hier: Der Controller muss die 24V für Anoden und Gitter schalten können. Natürlich ließe sich das mit zwei Transistoren pro Segment oder Gitter bewerkstelligen: Einen, der den Logikpegel "übersetzt" und einen, der als Schalter fungiert. Damit wären wir bei einer 4-Stelligen Anzeige schon bei 22 Transistoren: 14 für die sieben Anoden-Segmente und acht für die vier Gitter.

Zum Glück gibt es bewährte integrierte Lösungen für sogenannte *High-Side-Schalter*, wie etwa den ULD2981: Der ähnelt dem bekannten ULN2803-Treiberbaustein, seine Ausgänge schalten aber nicht nach Masse



durch, sondern nach Plus (*High Side*), hier der Anodenspannung.

In einem IC sind acht Treiber enthalten, für eine einfache Uhr braucht man also zwei Stück (7 + 4 = 11 Signale). Die restlichen fünf Treiberausgänge können dann für eventuell vorhandene zusätzliche Symbole, Dezimaloder Doppelpunkte verwendet werden.

Doch selbst mit bescheidenen 11 freien Ports kann mancher Controller nicht dienen, etwa der beliebte ESP8266. Man kann sich behelfen, indem man zum Beispiel die sieben oder (mit Dezimalpunkt) acht Segment-Signale über ein Schieberegister (4094 oder 74HC595) ausgibt. Das benötigt nur noch drei Portleitungen, aber einen kleinen zusätzlichen Programmieraufwand, um das Datenbyte seriell in SPI-Manier herauszuschieben.

Spezial-ICs

Wenn man ohnehin Portleitungen sparen muss, kann man auch gleich einen speziellen VFD-Treiber verwenden, der Schieberegister und High-Side-Treiber in einem IC integriert und mit nur drei Portleitungen auskommt. Gern wird hier der MAX6921 von Maxim (jetzt unter dem Dach von Analog Devices) verwendet, der mit seinen 20 Ausgängen eine 12-stellige Anzeige treiben kann. Die Ausgänge lassen sich gleichwertig als Anoden- oder Gitter-Schalter benutzen, sie sind bis 76V spannungsfest. Mit 4 bis 5 Euro ist er nicht ganz preiswert, als industrielles Bauteil ist seine Lieferbarkeit aber sichergestellt.

Mit weniger als einem Euro deutlich billiger sind die VFD-Treiber des chinesischen Herstellers PTC, die man oft in Geräten der Unterhaltungselektronik findet. Der PT6312 etwa steuert bis zu 11 Segmente und 11 Digits an, alternativ 16 Segmente und 6 Digits. Die Digit-Ausgänge für die Steuergitter ermöglichen durch PWM-Betrieb eine Helligkeitsreduzierung einzelner Ziffern oder Ziffergruppen. Wie beim MAX6921 gibt es im Arduino-Universum passende Treiber, sodass sich der Programmieraufwand in engen Grenzen hält.

Für alphanumerische 16-Segment-Anzeigeröhren wie die russische IV-4 oder IV-17 empfiehlt sich der PT6315, der bis zu 19 Segmente und 9 Digits ansteuert. Vorteil der (größtenteils vom NEC µPD16312 abgekupferten) PTC-Bausteine ist der selbstständige Betrieb, das heißt, das Multiplexen und Dimmen der Ziffern übernimmt komplett das IC – man braucht nur einmalig den Display-Inhalt und die einzelnen Helligkeitswerte hineinzuschreiben. Beim MAX6921 dagegen muss der Controller das Multiplexen selbst übernehmen, was etwas Rechenleistung kostet.

Rentenuhr

Wie versprochen gibt es nun noch ein kleines Beispiel-Projekt in Form einer VFD-Uhr. Der Autor hat sie dereinst konstruiert, um täglich an die verbleibenden Tage bis zum Renteneintritt zu erinnern, aber natürlich kann sie auch ganz normal die Zeit anzeigen und bietet sogar eine Küchentimer-Funktion. Als "richtiges" Projekt eignet sich die rund um einem ATmega328 aufgebaute Schaltung leider nicht mehr, da die Liefersituation des verwendeten Sonderposten-Displays ungewiss ist. Sicher können Sie aber den grundsätzlichen Aufbau aber auch mit Einzel-Röhren oder anderen Display-Typen versuchen.



In der Schaltung finden sich alle bisher besprochenen Elemente wieder: Neben dem Display sind hier die zwei ULD2981-Treiber zu finden, die ihre Daten von zwei 4094-Schieberegistern erhalten. Die wiederum sind an die SPI-Schnittstelle des ATmega328 angebunden. Die Anodenspannung von 12 bis 27V erzeugt der Step-Up-Wandler MC34063 aus der Betriebsspannung von 5V; die Helligkeit lässt sich mit R4 einstellen. R4 kann durch einen lichtempfindlichen Widerstand (LDR) ersetzt werden, der die Anodenspannung bei hoher Umgebungshelligkeit erhöht. Die Uhr kann problemlos mit einem USB-Steckernetzteil betrieben werden. Datum und Uhrzeit liefert über I2C ein DS1307, der bei Stromausfall von einer 3V-Lithium-Zelle versorgt wird.

Die Alarm- und Tickgeräusche (abschaltbar!) liefert ein kleiner Lautsprecher oder Summer. Vier Taster dienen zum Stellen der Uhr und zum Starten der Küchentimer-Funktion. Zu erwähnen sind vielleicht noch die Pull-down-Widerstände RN1 und RN2. Sie sorgen dafür, dass die Gitter-Spannung auch schnell genug 0V erreicht, wenn eine Ziffer abgeschaltet wird. Hardwaremäßig war es das auch schon!

Softwareseitig gibt es einige Besonderheiten: Da wir hier auch die Gitter-Steuerung über die Schieberegister und nicht über einzelne Portleitungen erledigen, müssen die zwei Bytes für die Treiber immer gemeinsam übertragen werden. Ein Latch-Impuls an den G-Eingängen der 4094er gibt die Daten dann für alle Ausgänge frei. Der Multiplex-Zyklus umfasst hier fünf Schritte, da das verwendete Display für den mittigen Doppelpunkt ein eigenes Gitter (Grid 3) besitzt.

Wenn man wie hier die komplette Ansteuerung des VFD per Software erledigt, kann man sich auch einige nette Effekte erlauben: Beim Wechsel der Uhrzeit etwa laufen die Ziffern wie ein Klappdisplay von oben nach unten durch, während Datum und Resttage mit einem Wischeffekt eingeblendet werden. Das Hervorheben einzelner Ziffern (etwa zum Stellen der Eieruhr-Funktion oder der Resttage bis zur Rente) geschieht einfach durch Verlängern der Anzeigedauer innerhalb des Multiplex-Durchlaufs. Zur Konfiguration der Uhr (Effekte, Tick-Geräusch) mittels Klartext-Befehlen (ein simples Terminal-Programm reicht, Einstellung 57600Bd/8n1) dient die serielle Schnittstelle, die Anschlussbelegung ist mit den bekannten FTDI-Platinchen und -Kabeln kompatibel.

Die Schaltung eignet sich mit einer eigenen Firmware auch als universelles Display, etwa zur Anzeige eines Messwertes oder auch einer Synthesizer-Programmnummer. Sollten Sie ein solches Display auf Vorrat gekauft und noch herumliegen haben, empfehlen wir einen Abstecher in unser Github-Repository (siehe Link im Info-Kasten): Hier haben wir auch ein Platinenlayout und die Firmware abgelegt. —*cm*

Ein Veranstaltungsformat der Make:



Maker Faire

Das Format für Innovation & Macherkultur

Die nächsten Events





C

HANNOVER Maker Faire

... weitere folgen.

maker-faire.de



Mini Game Machine

Das rund 40 Jahre alte, taschenrechnerartige Gerät sieht zwar heutzutage wie ein billiges Spiel-Gadget aus, enthält aber für damalige Verhältnisse modernste Technik.

von Daniel Bachfeld

ch bekam den Taschenrechner mit drei integrierten Spielen (Autorennen, Zahlen raten, 17 +4 aka Black Jack) vor einigen Jahren bei einer Schrottwichtel-Runde in die Finger. Er war mir damals zum Wegschmeißen zu schade, ich hebe häufig "Dinge" für unsere Rubrik *Reingeschaut* auf.

Zunächst fand ich die miniaturisierten 7-Segment-Anzeigen besonders faszinierend und wollte sie mir ein wenig genauer anschauen. Sie erinnern auf den ersten Blick nämlich an Fluoreszenzdisplays, wie sie der Kollege Meyer auf Seite 72 beschreibt. Bei näherer Betrachtung erkennt man dann aber, dass sie aus sehr kleinen LEDs bestehen, deren Anzeige je eine Lupe pro Ziffer vergrößert. Nutzt man das Gerät als Taschenrechner oder für zwei der eingebauten Spiele, so werden die LEDs als Ziffern ange-



Die Transistoren treiben die LED-Segmente gemultiplext an.

steuert. Für das Autorennen hingegen leuchten nur die horizontal ausgerichteten Segmente (Querstriche) der Ziffern, die von links nach rechts durch die Anzeige scrollen, um ein sich bewegendes Hindernis zu simulieren. Mit den beiden Pfeil-Tasten steuert man einen rechts angezeigten Strich hoch und runter, um dem nahenden Hindernis auszuweichen.

Das Ganze ist noch langweiliger als Frogger auf einem alten Atari VCS2600. Und damit kommen wir zur eigentlichen Überraschung. Im Gerät steckt nämlich der erste Mikrocontroller, der den CPUs größerer Heimcomputer und Konsolen früherer Jahre durchaus Paroli bieten konnte: Ein MOSTEK 3870. Die Geschichte hinter dem IC ist spannend: Der µC geht auf das 1975 vom US-Hersteller Fairchild eingeführte 8-Bit-Mikroprozessorsystem F8 zurück, das aus einem 8-Bit-CPU-IC mit 2MHz sowie weiteren ICs für RAM, (EP)ROM sowie I/O-Ports bestand. Das 4-IC-System war so erfolgreich, dass die Gerätehersteller nach einer alternativen Lieferquelle verlangten, woraufhin Fairchild ein Lizenzabkommen mit dem Hersteller Mostek einging, der die Chips ebenfalls fertigen durfte.

Fairchild verfeinerte das System und brachte es 1977 als Single-Chip-Lösung im



Das Gerät wurde weltweit unter verschiedenen Marken und mit teilweise anderen Funktionen verkauft.

Modell 3859 auf den Markt, in der CPU, RAM, ROM und I/O in einem einzigen 40-poligen Gehäuse untergebracht waren. Die Mostek-Ingenieure setzten aber noch eins drauf und stellten den 3870 vor, der mit doppelter Taktrate von 4MHz trotzdem relativ sparsam mit nur einer Versorgungsspannung von 5V lief und keine zusätzlichen 12V mehr benötigte. Das vereinfachte nicht nur die Entwicklung, sondern ermöglichte tragbare Geräte mit Batteriebetrieb.

Der Verkauf des 3870 ging durch die Decke, Fairchild stellte die Produktion des eigenen 3859 ein und lizensierte das Design des 3870, um damit selbst ICs fertigen zu können. 1979 wurde Mostek von UTC übernommen und 1985 an Thomson verkauft, die später mit SGS zu SGS-Thomson fusionierten und seit 1998 STMicroelectronics heißen. Auf archive.org (siehe Link) findet man noch alte Datenbücher zu ICs von Mostek, mit denen der ein oder andere ältere Make-Leser womöglich früher seine Elektronik-Kenntnisse erweitert hat. — dab





Sogar mit Sound-Ausgabe: Der Piezo signalisiert Fahrzeug-Crashes beim Autorennen.



Make-Breadboard++ Experimentierset zum Selbstbau

Koffer auf, alles da: Steckbrett, Kabel, Netzteil, Taster, LEDs und was das Maker-Herz sonst noch beim Experimentieren begehrt. Die Module lassen sich schnell tauschen. Zugeklappt lässt sich alles zerstörungsfrei transportieren oder zwischenlagern. Gibt es nicht zu kaufen, kann man aber selber bauen – so geht's.



m Anfang stand die Idee, ein Breadboard in eine Art kleines Pult einzubauen, das auch eine Stromversorgung mit den üblicherweise benötigten Spannungen enthält. So etwas gibt es in verschiedenen Ausstattungen auch fertig zu kaufen, angefangen mit zwei Breadboards und 4mm-Buchsen auf einer stabilen und schweren Metallplatte bis hin zu Experimentierplätzen mit Ein- und Ausgabe, Funktionsgeneratoren und programmierbaren Spannungsquellen. Ersteres ist noch günstig und bietet eine gute Standfestigkeit und etwas Komfort, Vertreter der letzteren Klasse von Experimentiersets sind sehr teuer oder werden nur an Institutionen wie Schulen oder Universitäten verkauft. Im Preissegment dazwischen gibt es nicht nur wenig Auswahl das Hauptproblem ist in meinen Augen, dass die integrierten Komponenten fest installiert sind. Wenn man z. B. für seine Projekte eher keinen Signalgenerator und keine BNC-Buchsen braucht, sondern einen MIDI-Anschluss und Klinkenbuchsen, dann bleibt ein Großteil der Funktionen ungenutzt. Das hat mich dann auf die Idee gebracht, ein modulares System zu bauen, bei dem die Komponenten einfach ausgetauscht werden können.

Als Basis dient eine Sperrholzplatte, auf der die Breadboards liegen, umgeben von Ausschnitten, in die verschiedene austauschbare Module passen. Aber natürlich sind auch andere Materialien für die Grundplatte denkbar, bis hin zu aufwendig gestalteten Leiterplatten. Als Vorlage für die Modulgröße diente mir ein fertiges Netzteilmodul mit einer Ausschnittgröße von 72mm × 39mm. Alle Module sollten die gleiche Breite haben. Für mehr Flexibilität habe ich von Anfang an auch ein größeres, parallel verwendbares Format (72mm × 51mm) vorgesehen, um auch Module mit größerem Platzbedarf nutzen zu können. Für das kleinere Format gibt es diverse Module vom Netzteil über Funktionsgeneratoren bis hin zu Motortreibern fertig zu kaufen.

Bei der Suche nach einem günstigen Gehäuse stellte ich mir zunächst ein Pult vor, fand dann aber einen alten Kunststoffkoffer, der einmal ein Großmembranmikrofon *t.bone SC* 440 USB enthielt. Die Größe passte für meine Vorstellungen perfekt und der Koffer ist stabil. Für meine gewünschten Komponenten hatte ich mir schnell ein ungefähres Layout zurechtgelegt und das Basteln konnte beginnen. Ich habe dabei versucht, so wenig wie möglich neu zu kaufen und stattdessen Teile aus der Bastelkiste zu verwenden.

Kofferausbau

Ich habe mit verschiedenen Methoden und Schablonen versucht, die Innenform des Koffers abzupausen, brauchte aber dennoch zwei Anläufe, bis ich eine brauchbare Platte 1 angefertigt hatte. Vor allem am Griff ist der Wech-

Kurzinfo

- » Breadboard mit vielen nützlichen Extras für bequemes Experimentieren
- » Modularer und erweiterbarer Aufbau
- » Mobil oder stationär

Checkliste

Zeitaufwand: ab 10 Stunden



Je nach Inhalt der Bastelkiste und Wünschen zwischen 0 und 200 Euro

Werkzeug

- » Stichsäge mit schmalem Sägeblatt für Kurvenschnitte
- » Laubsäge für die Ausschnitte
- » 3D-Drucker
- » Lötkolben und Lötutensilien
- » Makerwerkzeug wie Schrauben-
- dreher, Zangen, Cutter, Bohrer etc.

Mehr zum Thema

- » Hans-Jürgen Pretzel, Erster Elektro-Experimentierkasten
- Make Sonderheft 2022, S. 10 » Jens Hackel, Biegelehre: Drahtbrücken und Bauteilbeinchen
- präzise abwinkeln, Make Sonderheft 2022, S. 24 » Carsten Wartmann, Elektronik
- steuern mit dem Pico für Anfänger, Make 5/22, S. 40
- » Video: Übersicht des Koffers



Material für den Beispielkoffer

- » Koffer vom Baumarkt für Werkzeug oder vom Flohmarkt
- » 5mm Sperrholz Maße je nach Koffer » Dekofolie etwa *d-c-fix*
- » Breadboards auf Qualität achten
- » ELV Mini-Steckplatine, 2 × 4 und 2 × 8
- » Buchsenleisten, 40-polig, RM 2,54, gerade
 - **Maschinenschrauben** M2 \times 10 und M3 \times 8
- Schalter, Taster, LEDs
- » Vorwiderstände für LEDs z.B. 4700hm
- » Potentiometer oder Trimmer 1kOhm, 10kOhm
- » Drehencoder mit Tastfunktion
- » Punktraster Platinen 30 × 70mm, 50 × 70mm
- » Labornetzteilmodul z.B. XT-SK35
- » USB-C Power-Trigger Power-Delivery Platine
- » 5V Spannungswandler Eingang 3-20V » Telefon-Buchsen 4 mm Farben nach Wahl
- » Wago Klemmen 2-5-fach
- » Kippschalter mit Schutzkappe und LED

Alles zum Artikel

make-magazin.de/x2o

im Web unter

- » Kabel Litze 0,5mm², für Stromversorgung
- » Kabel eindrähtig 0,05mm², zur Verdrahtung



zu erfassen. Wenigstens konnte ich mit der nicht verwendeten ersten Version noch üben, wie man Ausschnitte herstellt und die erste Anordnung der Module überprüfen **2**.

Der Einsatz wurde mit der Stichsäge ausgesägt, mangels schmalem Sägeblatt für Kurvenschnitte bei den Radien nur grob und dann per Raspel und Feile nachgearbeitet. Die Aus-

> Das Erstellen eines guten Umrisses zum Sägen war ein nicht triviales Problem und erforderte zwei Anläufe.



Achtung, Mitmachprojekt!

Der hier vorgestellte Koffer ist für meine speziellen Anforderungen gedacht, aber durch die Modulbauweise kann jeder Bastler, der mit Breadboards arbeitet, seine eigene Version bauen. Da der Entwurf eigener Module für Einsteiger zu kompliziert sein kann, möchten wir gerne gemeinsam mit Ihnen, unseren Lesern, ein umfassendes Modulsystem zum Nachbau entwickeln und dokumentieren.

Hierfür brauchen wir Ihren Input! Wir suchen ebenso Beschreibungen, Fotos, Pläne und Platinenlayouts von Modulen, die Sie selbst entwickelt und gebaut haben, wie Wünsche nach Modulen, die Ihre Fähigkeiten übersteigen und die wir für Sie und andere Leser entwickeln sollten. Dabei sind auch die von uns vorgeschlagenen Größen der Module nicht in Stein gemeißelt– wenn wir durch Ihre Vorschläge oder Wünsche ein noch praktischeres Format finden, umso besser.

Wie immer freuen wir uns über Anregungen, Fotos von Ihren Breadboards++ oder eigenen tollen Modulen per Mail an mail@make-magazin.de!

schnitte für die Module habe ich dann mit der Laubsäge per Hand (3) ausgesägt. Hier war mir die Verwendung der Stichsäge zu heikel und ich besitze auch keine passende kleine Handfräse, mit der man die Ausschnitte nach Schablonen herstellen könnte. So konnte ich ein paar Fähigkeiten meiner Kindheit wieder hervorkramen und mich etwas körperlich betätigen. Angezeichnet wurden die Ausschnitte mit einer 3D-gedruckten Schablone (mehr dazu siehe unter *CAD und 3D-Druck*).

Die Bauteile sind so angeordnet, dass ich als Rechtshänder beim Einstecken von Bauteilen und Jumperkabeln in das Breadboard möglichst nicht über höhere Module greifen muss und meine Hand flach auf dem Kofferrand aufliegen kann. Dies wird durch den bündigen Abschluss der Grundplatte zum Kofferrand noch angenehmer. So bleibt außerdem unter der Platte maximaler Platz für Einbauten wie Netzteile, Akkus oder auch Stauraum.

Meine Lackierkünste sind wohl noch schlechter als meine Sägekünste, und so habe ich wieder d-c-fix-Dekorfolie verwendet, um den Koffereinbau zu verkleiden, wie schon bei meinem Bartop Arcade (siehe Kurz-URL). Es hat sich herausgestellt, dass die Folie nicht nur gut aussieht, sondern auch super zu verarbeiten und sehr haltbar ist. Außerdem gibt es sie in verschiedenen Farben und Designs, glänzend oder matt.

2 Erste grobe Planung der Komponenten, einige Details wie das Display-Modul und das Strom-Modul wurden danach noch geändert.

Die zu folierenden Teile sollten *leicht* geschliffen werden – so glatt wie zum Lackieren muss es nicht sein. Bei gutem Sperrholz reicht eventuell auch eine kurze Reinigung: Die ist wichtig, weil die Oberfläche für das Folieren staubfrei sein muss, denn jeder noch so kleine Krümel auf der Klebefläche oder dem Holz ist später durch die Folie sichtbar. Außerdem kann der Kleber auf sauberen Flächen besser haften.

Ich habe die Folie 4 mit einem weichen Tuch von Hand angedrückt. An den Rändern wurde die Folie mit einem Schraubendrehergriff fest angedrückt.

Die Ausschnitte werden mit einem Cuttermesser ausgeschnitten, die Ränder nach innen gebogen und auf die Rückseite der Platte geklebt. An den Radien müssen einige Schnitte gemacht werden, um die Folie gut an die Rundung anpassen zu können. Da es sich nicht um eine sichtbare Kante handelt, ist es unkritisch, die Folie mit etwas Kraft zu ziehen, um sie überlappend zu verkleben. An Innenradien bleibt natürlich auf der Unterseite etwas Holz sichtbar, dies wird aber vom Kofferrand verdeckt.

Die Breadboards habe ich von unten durch die Einlegeplatte verschraubt und nicht das üblicherweise vorhandene Doppelklebeband benutzt, somit lässt sich das Breadboards auf Blechträgern is sind bereits Löcher für die Schrauben vorhanden. Ansonsten muss man die Löcher vorsichtig bohren, oft gibt es auch eine Nut unter der Platine an der Stelle, wo oben der Spalt zwischen den Platinenseiten ist, in die man die Schrauben eindrehen kann. Tabu sind natürlich die Stellen, an denen die Kontaktfedern sitzen, hier kann man zu leicht etwas beschädigen oder einen Kurzschluss verursachen.

Eine sehr schöne Alternative wäre es, das Breadboard magnetisch zu befestigen, sodass sie leicht gegen andere ausgetauscht werden können, was die parallele Arbeit an mehreren Projekten erleichtern würde. In Falle einer magnetischen Befestigung könnte man auch einen Ausschnitt unter dem Breadboard vornehmen, in dem sich ein herausnehmbares Kleinteilemagazin befindet.

Die Module

Zu meinen hier vorgestellten Modulen finden Sie die STL-Dateien für den 3D-Druck sowie die anpassbare *Blender*-Datei für eigene Modifikationen auf GitHub. Im selben Repository stellen wir auch die Schaltpläne und später auch Ihre Ideen und Module bereit.

Ich habe mir ein Buck-Up-Down-Netzteil angeschafft, das einen weiten Eingangsbereich hat, sodass unabhängig von der Versorgungsspannung des Koffers (siehe *Stromversorgung*) Spannungen bis 30V zur Verfügung stehen. Durch die Möglichkeit der Strom- und Spannungsbegrenzung können Schaltungen getestet werden, ohne dass bei Kurzschlüssen Bauteile durchbrennen.

Das erste selbst gebaute digitale Ein-/Ausgabemodul i enthält Taster und LEDs. Die LEDs besitzen bereits Vorwiderstände, die so dimensioniert sind, dass die LEDs direkt mit Spannungen von drei bis etwa neun Volt betrieben werden können. Aus Platzgründen wurden SMD-Widerstände direkt auf die Lochrasterplatine gelötet.

Das Schalter-Drehgeber-Modul 🕜 ist mit Wechsel-Schaltern, zwei Potentiometern und einem Drehgeber (*Encoder*) mit Tastenfunktion ausgestattet. Es dient als Test für das größere Modulformat mit 40% mehr Platz. Die Potis können als Spannungsteiler oder als einstellbare Widerstände verwendet werden. Eigentlich sind es Trimmer, die auf die Platine gelötet und dann mit einer Steckachse versehen werden.

Das Stromverteilungsmodul 🕜 stellt 5V, GND und die variable Spannung des Netzteils über 4mm-Buchsen (*Bananenstecker*) zur Verfügung. Zusätzlich bietet es noch +/- 12V für Schaltungen mit Operationsverstärkern auf Steckfeldern (siehe auch Abschnitt Stromversorgung).

Das Arduino-Modul mit OLED (im Titelbild oben in der Mitte) kann von Mikrocontrollern auf dem Breadboard oder vom integrierten Arduino Pro Mini über Buchsenleisten betrieben werden. Der Arduino fand sich in der Bastelkiste, muss aber über einen USB-Seriell-Wandler (UART) programmiert werden. Momentan läuft darauf ein Arduino-Sketch, der eine Logikpegelanzeige und ein einfaches Zeitdiagramm für vier Kanäle anzeigt. In Zukunft wird aber ein Raspberry Pico dort seinen Dienst tun, das OLED-Display wird wahrscheinlich durch ein kleines TFT wie das Pimoroni pico display (siehe Link) ersetzt werden. Es bietet nebenbei auch vier Taster, mit denen man z. B. das zu verwendete Programm auswählen kann und eine RGB-LED.

Links und rechts vom Griff befinden sich dann noch schmalere Module, die die Haupt-

8 Ausschnitte f
ür die Module, klassisch mit der Laubs
äge angefertigt

schalter und die Anschlüsse enthalten, in meinem Fall eine DIN-Buchse für *MIDI* und Miniklinke.

Mit Ausnahme des Netzteilmoduls, der Schalter und des Stromverteilermoduls sind alle Module nur von oben über Buchsenleisten mit Jumperkabeln anzuschließen. Die Kabel der Stromquelle, des Netzteilmoduls und des 5V-Moduls sind alle mit *Wago*-Klemmen verbunden. Diese lassen sich beim Austausch der Module schnell öffnen und sind wesentlich flexibler als Steckverbindungen.

Für die selbst gebauten Module habe ich Punkt-Lochraster-Platinen in Standardgrößen verwendet: Die normalen Module tragen eine 70mm × 30mm große Platine, die Platinen der großen Module messen 70mm × 50mm. Sollten aufwändigere Schaltungen benötigt werden, dann kann man bei Platznot auch eine zweite Platine stapeln.

Die Komponenten wurden dann im Falle von Schaltern und dergleichen verkabelt bzw.

auf den Platinen mit Lötzinnbrücken Kontakte zwischen den Lötaugen hergestellt. Das ist zwar einmalig viel Arbeit, aber für Einzelstücke praktischer, als eigens eine Platine zu entwerfen und die dann zu ätzen oder fertigen zu lassen.

Die Sicherheitsschalter sehen cool aus, sind aber wahrscheinlich der Knackpunkt, wenn man so einen Koffer mal durch die Sicherheitskontrolle am Flughafen bringen will. Im Ernstfall sind sie aber wirklich schnell ausgeschaltet (siehe Video) und die LEDs zeigen deutlich an, ob Strom anliegt.

CAD und 3D-Druck

Die Holzblende wurde zunächst nur auf ein paar 3D-gedruckte Abstandshalter in den Koffer gestellt. Erst als die endgültige Position feststand, habe ich die Füße von unten mit M3-Schrauben fixiert, die gleichzeitig die Gummifüße (9) halten, auf denen der Koffer sicher und rutschfest steht.

👍 Mit Dekorfolie erhält man in kurzer Zeit schöne und haltbare Oberflächen.

Sviele Breadboards bieten schon Schraubenlöcher, oft muss man aber unter dem Doppelklebeband suchen.

Oas Digital-IO-Modul, fünf Taster und fünf LEDs mit Widerständen. Der Pappstreifen sorgt für die bündige Ausrichtung der LEDs.

Schon während der allgemeinen Planung der Module habe ich angefangen, parallel in Blender (siehe Videolink in Kurzinfo) zu konstruieren, vor allem, um ein genaueres Gefühl für die Anordnung der einzelnen Komponenten auf den Modulen zu bekommen. Dabei wurden viele Modifier 🕕 in Blender wie Solidify, Mirror, Array und Boolean, sowie Linked-Meshes verwendet, um die einzelnen Module aufeinander aufbauen zu können. Damit vererben sich dann Änderungen und Variationen auf alle Module, die die gleiche Grundgeometrie verwenden, und die Modifier können unabhängig von dem Mesh geändert werden. Bei der Platzierung von Bohrungen und Durchbrüchen hat sich ein 2,54mm-Hilfsrasterobjekt als sehr nützlich erwiesen. Aus den Konstruktionsdaten der Module konnten dann auch einfach Schablonen zum Anzeichnen der Ausschnitte gewonnen werden.

Die Module werden mit je zwei M2-Maschinenschrauben gehalten, die nicht in eine Bohrung greifen, sondern nach unten wegspreizen und so das Modul festklemmen. Dies wird dadurch erreicht, dass die Schrauben leicht schräg in der Bohrung (1) geführt werden. Klemmnasen aus Kunststoff, wie bei dem gekauften Netzteilmodul, sind schwer zu drucken und brechen zu leicht ab, außerdem ist die Dicke meiner Sperrholzplatte etwas zu groß für diese Art von Haltenasen. Gedruckt wurde alles in PETG (*Prusament JetBlack*) auf meinem *Prusa i3 MK2*. In den Deckel wurden noch 3D-gedruckte Halterungen () für Jumper-Kabel und ein Kleinteilefach integriert. Momentan sind diese Einbauten noch mit *Patafix* (einer Knete-artigen Klebemasse) befestigt und können daher leicht anders angeordnet werden. Bisher hält das Patafix sehr gut und vielleicht erweist es sich sogar als dauerhafte und dennoch flexible Lösung. Durch die geringe Tiefe des Deckels ist allerdings nicht viel Platz, da die Module und Aufbauten auf dem Breadboard in den Deckel ragen und so beim Schließen beschädigt werden könnten.

Stromversorgung

Für die Stromversorgung B wurden von mir verschiedene Möglichkeiten in Betracht gezogen: Ein Netzteil mit Anschluss an 230V Haushaltsstrom, ausreichend vorhandene AA-Akkus, die in einem 10er Halter 12V liefern, ein Lithium-Akku oder direkt eine Powerbank mit USB-C und Power Delivery (PD, siehe unten). In der Bastelkiste fand sich ein Powerbrick, der unter das Grundbrett passte und praktischerweise 5V und 12V mit ausreichender Amperezahl liefern konnte. Wenn man statt eines Koffers ein stationäres Experimentierbrett bauen möchte, sind übrigens auch (nicht zu alte) ATX-Computernetzteile eine gute Wahl, da sie viele Spannungen und genügend Leistung bieten.

Letztlich habe ich mich dennoch für USB-C entschieden. Ich hatte mir mal USB-C-PD-(Power Delivery)-Platinen für ein anderes Projekt gekauft. USB-C kann nicht nur die üblichen 5V liefern, sondern auch in Stufen bis zu 20V, womit man bis zu 100W über ein USB-Kabel bekommt. Allerdings braucht man dazu einen Chip der diese Spannungen auch vom USB-Controller anfordert. USB ist überall und ein kleines USB-C-Netzteil mit PD-Fähigkeit passt

Das große Modul mit drei Schaltern, zwei Potentiometern und einem Encoder. Patafix-Knetkleber hält die Buchsen beim Löten.

8 Nicht nur in diesem Projekt haben sich Wago-Klemmen als äußerst flexibel und kontaktsicher erwiesen.

für den Transport auch noch in den Deckel, wenn es denn Netzstrom sein muss. Gleichzeitig ist man so in der Lage, auch mobil mit einer Powerbank zu arbeiten.

Mit USB-C (2) und Power Delivery funktioniert das sehr gut. Versorgt man den Koffer allerdings über ein USB-A-zu-USB-C Kabel, dann bekommt man am 5V-Anschluss nur ca. 4,6V, was empfindliche 5V-Chips stören kann. Das Buck-Boost-Netzteil hat damit aber kein Problem und so erhält man von dort die 5V und auch sonst alle Spannungen bis 30V – bei USB-2 natürlich nur mit geringer Stromstärke, da laut Spezifikation an USB-2-Ports nur 500mA, also 2,5 Watt geliefert werden. USB-2-Powerbanks oder Ladegeräte liefern hier schon mehr, trotzdem sollte man lieber ein PD-fähiges USB-C-Netzteil oder eine solche Powerbank verwenden.

Die 5V werden über ein separat abschaltbares Step-Down-Modul (roter Schalter) direkt vom USB-C-PD-Anschluss eingespeist. Das +/-12V-Modul zur Erzeugung dieser Spannungen wird von mir bereits in einem kleinen Eurorack-Synthesizer-System eingesetzt, in dem nur einige Operationsverstärker symmetrisch versorgt werden müssen. Es liefert zwar nur 300mA pro Kanal, was aber für die Anwendung auf dem Breadboard ausreicht, und arbeitet mit einem doppelten Buck-Boost-Wandler.

Alle 3D-Druckteile wurden in Blender konstruiert. Dabei wurden viele Modifier benutzt, um die Module leicht anpassen zu können.

Variationen und Erweiterungen

Die mechanische und optische Ausführung des Breadboard++ ist natürlich eine sehr persönliche Sache. Pult oder Koffer, Größe, Art und Anzahl der Module, Stromversorgung: All das hängt stark von den eigenen Bedürfnissen ab. Hier in der Redaktion kam auch die Idee auf, das Ganze mit Schubladenzügen unter dem Arbeitstisch zu montieren, eine platzsparende Idee, die das kreative Chaos zumindest auf dem Tisch in den Griff bekommt. Als Koffer werden diverse Werkzeugkoffer aus dem Baumarkt und Ikea-Behältnisse diskutiert.

9 Gummifüßchen sorgen für einen sicheren Stand des Koffers.

Für den **Maker-Neuling** (*Zero to Maker*) oder als **Lehrer** in Schule oder Ausbildung sowie in Makerspaces, können die klassischen Komponenten als Module gebaut werden: LEDs natürlich, Taster, Schalter, Joystick-Module, Tastenfelder, verschiedene Umweltsensoren, ein Servomodul oder eines mit Vibrationsmotor, RGB-LEDs und kleine Arrays aus programmierbaren RGB-LEDs (*Neopixel*). Die Module wachsen mit den Fähigkeiten, bleiben verfügbar und können im Makerspace untereinander getauscht werden.

Die Version für den **Synthesizer-Bastler** ist praktisch meine im Artikel vorgestellte Version. Ich entwerfe meine Eurorack-Module auf Breadboards und übertrage diese Entwürfe dann auf Leiterplatten (da es sich um Einzelstücke handelt, auf Lochraster- oder Streifenrasterplatinen). Da in Synthesizern und anderer Musikelektronik fast immer Operationsverstärker (OpAmps) verwendet werden, ist eine symmetrische Stromversorgung wichtig. Für die ersten Experimente und Tests wäre zusätzlich ein Modul mit Verstärker und Lautsprechern schön. Mehr Potis und Klinkenbuchsen wären noch besser, denn vor allem die Potis sind notorisch dafür bekannt, auf Bread-

boards Probleme zu machen. Andererseits kommen einige meiner hybriden (Mikroprozessor und analog) Eurorack-Module schon mit zwei Potis und vielleicht einem Drehgeber aus.

Der **Digitale Maker** experimentiert mit digitalen Schaltungen (*TTL/CMOS*) und benötigt eine perfekte Stromversorgung, vielleicht sogar mit Spannungen von 1,8, 2,5, 3,3 und 5 Volt gleichzeitig. Ein Mikroprozessormodul kann während der Entwicklung komplexere Aufgaben übernehmen oder Logikpegel anzeigen. Ein Modul, das Logikpegel (*Logic-Probe*, *HIGH/LOW/PULSE*) durch LEDs und Summer anzeigt, ist ebenfalls sehr nützlich. Im Titelbild befindet sich solch eine Schaltung im Versuchsstadium vorne links auf dem Breadboard. Impulsgeneratoren, Taktgeneratoren (*Clock*), Frequenzzähler und Frequenzgeneratoren dürfen auch nicht fehlen. Entprellte oder ge-

Es gibt verschiedene Möglichkeiten der internen und externen Stromversorgung, ich habe einige ausprobiert.

Die Schrauben funktionieren durch eine schräge Bohrung wie Klemmen, die sich in die Platte krallen.

pufferte Taster, Schalter und LEDs, I²C- oder Onewire-Scanner, Chip-Adapter und *Level-Shifter* sind ebenfalls nützlich.

Der Analoge Maker entwickelt analoge Schaltungen für Messtechnik oder Audioanwendungen. Er benötigt jedoch Netzteile, die weder brummen noch digitale Störungen auf den Versorgungsspannungen haben, oder er verwendet gleich eine Akkulösung. Die Breadboards begrenzen die Frequenzen der Schaltungen auf etwa 10MHz. Bei hochwertigen Breadboards (siehe Bezugsquellen in den Links) werden noch dünne Bleche mitgeliefert, die man unter die Breadboards montieren kann, die die Schaltungen noch etwas gegen Störungen von außen abschirmen. Als Module eignen sich für den Analog-Maker sicherlich Messgeräte, Komponententester, Frequenzgeneratoren und vielleicht sogar ein kleines Oszilloskop-Modul.

Denkbare Erweiterungen und Anwendungen für solch ein modulares Experimentierset sind so zahlreich wie die elektronischen Projekte der Maker. Passen Sie Ihr Breadboard++ genau Ihren Vorstellungen und Bedürfnissen an. Ich werde auf jeden Fall noch kleine Pfosten entwerfen, die die langen Jumper-Kabel von den Modulen zum Breadboard in geordnete Bahnen lenken und das Display- und Mikroprozessormodul neu gestalten. —*caw*

Much die USB-C-Anschlussplatte ist 3Dgedruckt. Ähnlich könnte man auch einen Kaltgeräte-Anschluss einbauen.

Generell portofreie Lieferung für Heise Medien- oder Maker Media Zeitschriften-Abonnenteg <u>eder ab einem Einkaufswert von 2</u>0 € (innerhalb Deutschlands). Nur solange der Vorrat reicht. Preisänderungen vorbehalten.

🕼 heise Shop

Tipps und Tricks

Auch ein Smarthome mit Home Assistant kann noch verbessert werden: Diese Tipps zeigen, wie man mehr Infos aus Heizkörperthermostaten herausholt, Bluetooth-Probleme beseitigt und Künstliche Intelligenz in Form von Chatgpt hinzufügt.

von Heinz Behling

Mehr Messwerte bei eq3-Heizkörper-Thermostaten

Die Heizungssteuerung mithilfe von elektronischen Heizkörperthermostaten der Marke *eq3* ist nun in *Home Assistant* deutlich einfacher und komfortabler geworden: Wer seine Thermostaten noch mittels Eintrag in der Konfigurationsdatei *configuration.yaml* in der Form

climate:

```
- platform: eq3btsmart
devices:
    room1:
    mac: "00:11:22:33:44:55"
```

eingebunden hat, sollte auf die unter HACS erreichbare neue Integration *dbuezas_eq3btsmart* wechseln. Dazu auf der Home-Assistant-Oberfläche auf HACS wechseln und dort auf die drei Punkte oben rechts klicken. Dann als benutzerdefiniertes Repository https://git-

https://github.com/dbuezas/ed	q3btsmart
Kategorie Integration	•

 Die Einträge f
ür die neue Integration der eq3-Thermostate

Hier müssen Sie auf den Integrationsnamen, nicht auf den roten Papierkorb klicken.

Anhand der MAC-Adresse und dem bisherigen Konfigurationseintrag können Sie feststellen, zu welchem Heizkörper dieses Thermostat gehört.

hub.com/dbuezas/eq3btsmart eingeben. Die Kategorie lautet Integration 1.

Nach einem Klick auf *Hinzufügen* erscheint oben in dem kleinen Fenster der Name der neuen Integration 2). Einmal auf den Integrationsnamen und anschließend noch zweimal auf *Herunterladen* geklickt, befindet sich die neue Integration auf dem Home-Assistant-Server. Der muss nun neu gestartet werden.

Unter Einstellungen/Geräte & Dienste dann mit Integration hinzufügen nach dbuezas_ eq3btsmart suchen und installieren. Danach braucht das System einige Minuten (bis zu einer Viertelstunde), um alle Heizkörperthermostate zu finden. Im Erfolgsfall wird ein kleines Fenster angezeigt, indem die MAC-Adresse des Thermostats aufgeführt ist. Die haben Sie bislang auch im Eintrag in der Konfigurationsdatei benutzt **3**.

Klicken Sie auf das Konfigurationsfeld und geben Sie dem Thermostaten einen Namen, zum Beispiel *Bad.* Sind alle Heizkörper eingerichtet, erscheint in Home Assistant eine Liste unter der Integration. Nach einem Klick auf einen der Thermostatnamen und auf *Geräte* sagt Ihnen die neue Integration alle Werte, die von nun an übermittelt werden. Neu ist zum Beispiel der Batteriestatus **4**.

Auch die Ventilöffnung (Valve) ist nun verfügbar: 0% bedeutet geschlossen, 100% entsprechend voll geöffnet **5**.

Bluetooth-Probleme beim Pi4 mindern

Bluetooth BLE ist eine sehr energiesparende Möglichkeit, Daten zum Beispiel von Thermometern und anderen Sensoren an einen Home-Assistant-Server zu übermitteln. Da die meist als Home-Assistant-Server benutzten Raspberrys Bluetooth bereits eingebaut haben, wird dies auch gern benutzt. Allerdings gibt es dabei häufig Probleme, denn wegen des geringen Energiebedarfs ist die Sendeleistung und damit die Reichweite einer BLE-Verbindung begrenzt. Erschwerend kommt hinzu, dass bei Verwendung eines Raspberry Pi 4 dessen USB-3-Interface Störungen im von BLE benutzten 2,4GHz-Bereich verursachen kann.

Mit einigen kleinen Maßnahmen kann man aber die Bluetooth-Verbindungsqualität verbessern. Am einfachsten und preiswertesten ist es, einen zusätzlichen preiswerten BLE-fähigen Bluetooth-USB-Stick als zweites Interface zu verwenden. Den sollte man aber nicht unmittelbar in die USB-Buchsen des Raspberrys stecken, sondern über ein USB-Verlängerungskabel (bis zu 5m Länge sind kein Problem) verbinden. So sitzt der Stick und damit dessen Antenne weit von den störenden USB-3-Buchen entfernt. Außerdem kann er auch näher zu den Sensoren platziert werden, die unter Verbindungsproblemen leiden. Die Bluetooth-Integration der aktuellen Version von Home Assistant verhindet Geräte automatisch mit dem Interface, das die beste Verbindung hat. Somit wären auch noch mehr BLE-Sticks möglich. Auch 10mlange USB-Kabel funktionieren eventuell, sodass man den Stick sogar in anderen Räumen platzieren könnte.

In schwierigeren Fällen hilft nur die Verwendung besserer BLE-Sticks, die dann aber auch deutlich teurer sind. Empfehlenswert sind Sticks mit CSR8510A10-Chipsatz. Bewährt hat sich der Sena UD100-G03 (a). Der kostet allerdings über 100 Euro (gebraucht oft billiger), bringt aber auch deutlich stabilere Verbindungen selbst durch mehrere Wände hindurch.

Neue KI-Integration OpenAI in Home Assistant

Künstliche Intelligenz ist im Moment und vermutlich auch in der Zukunft ein wichtiges Thema. Text aufgrund der im Internet vorhandenen Informationen verfassen ist mit ChatGPT bereits recht einfach. Seit kurzem gibt es auch für Home Assistant eine Integration zum Service von *OpenAI*, der Firma, die ChatGPT entwickelt hat und betreibt.

Die Installation ist recht einfach. Zuvor muss man sich aber bei OpenAl einen Key-Code besorgen: Nach der Registrierung auf https://platform.openai.com/signup, bei der man neben der Email-Adresse auch eine Tele-

Die Einführung von Arduino-Mitbegründer Massimo Banzi!

Arduino ist die Open-Source-Plattform für elektronische Prototypen, die die Maker-Bewegung im Sturm erobert hat. Diese gründliche Einführung, aktualisiert für die neueste Arduino-Version, hilft dir, sofort mit dem Prototyping zu beginnen. Von der Beschaffung der benötigten Komponenten bis zum letzten Schliff deines Projekts findest du alle Informationen, die du brauchst. Alle Projekte eignen sich für einen Arduino Uno oder Leonardo. Ohne Löten!

2023 · 282 Seiten · 29,90 € ISBN 978-3-86490-913-9

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

2. Auflage 2023 · 504 Seiten · 34,90 € ISBN 978-3-86490-918-4

3. Auflage 2022 · 366 Seiten · 36,90 € © Copyright by Maker Media GabH. ISBN 978-3-86490-867-5

×	Assistent	0
Wie	kann ich hehilflich sein?	
Wie	kann ich behilflich sein?	
Wie	e <mark>kann ich behilflich sein</mark> ? abe einer Anfrage	

8 Schmucklos fragt der Chatbot auf Deutsch nach Ihren Wünschen.

×	Assistent	0
Wie	e kann ich behilflich sein?	
	lst es im Badezimmer käl Wohr	ter als im nzimmer?
Yes terr terr low livir	s, according to the nperature/humidity sensor nperature in the bathroom er than the temperature in ng room.	rs, the is 1 the
Yes terr terr low livir	s, according to the aperature/humidity sensor aperature in the bathroom for than the temperature in ag room. gabe einer Anfrage	rs, the is in the

9 Frage auf Deutsch, die Antwort kommt jedoch englisch.

Machen Sie mit!

Kennen Sie auch einen raffinierten Trick? Wissen Sie, wie man etwas besonders einfach macht? Wie man ein bekanntes Werkzeug oder Material auf verblüffende Weise noch nutzen kann? Dann schicken Sie uns Ihren Tipp – gleichgültig aus welchem Bereich (zum Beispiel Raspberry, Arduino, 3D-Druck, Elektronik, Platinenherstellung, Lasercutting, Upcyling ...). fonnummer für die Bestätigung angeben muss, kann man den im User-Bereich unter API-Key generieren. Notieren Sie sich den Code, denn es gibt keine Möglichkeit, ihn nochmals anzeigen zu lassen.

Mit dieser Registrierung haben Sie als privater Anwender (bitte bei der Registrierung wählen), einen kostenlosen Testzugang erhalten, bei dem sich auf Ihrem OpenAl-Konto 5 Dollar Guthaben befinden (Stand Anfang März 2023). Jeder Zugriff auf den Service kostet dann ein paar Cent bis diese Summe verbraucht ist. Danach geht es nur noch mit einem kostenpflichtigen Service weiter. Aber zum Testen für ein paar Tage genügt das.

In Home Assistant fügen Sie unter *Einstellungen/Geräte & Dienste* die neue Integration *OpenAI Conversation* hinzu. Nach Eingabe des API-Keys ist die auch schon einsatzbereit. Sie befindet sich auf der Übersichtsseite hinter dem kleinen Sprechblasen-Symbol oben rechts **7**.

Das folgende Fenster ist der Zugang zum Chatbot 8.

Dort können Sie nun Fragen zu Ihrem Smarthome loswerden, beispielsweise nach Temperaturen (9), noch eingeschalteten Lampen usw.

Nach einer kurzen Bedenkzeit antwortet der Bot dann. Es gibt aber einige Einschränkungen: So kann der Bot keinerlei Aktionen im Smarthome auslösen, also keine Geräte ein- oder ausschalten, Einstellungen ändern oder Software installieren. OpenAl Conversation ist also nicht in der Lage, das Smarthome zu steuern, wie Alexa das beispielsweise kann. Allerdings kann man mit dem Chatbot auch mündlich kommunizieren, falls man auf einem Smartphone die Home-Assistant-App installiert und für die Spracheingabe freigegeben hat. Die Antworten liefert er aber auch dort nur schriftlich.

Und fragt man ihn beispielsweise um Hilfe bei Updates oder Erweiterungen der Software, dann verweist er lediglich allgemein auf die Supportseiten von Home Assistant. Spezifische Antworten zu einem in der Frage konkret angegebenen Problem gibt der Bot leider (noch?) nicht.

Wenn wir Ihren Tipp veröffentlichen, bekommen Sie das bei Make übliche Autorenhonorar. Schreiben Sie uns dazu einen Text, der ungefähr eine Heftseite füllt und legen Sie selbst angefertigte Bilder bei. Senden Sie Ihren Tipp mit der Betreffzeile *Lesertipp* an:

mail@make-magazin.de

Optionen			
Eing	abeaufforderung Vorlage *		
1234567	This smart home is controlled by Home Assistant. An overview of the areas and the devices in this smart (%-for area in areas %) (%-for device in area_devices(area.name) -%) (%- if not device atri(devices, ~disabled_by?) and (%- if not device atri(device, ~disabled_by?) and	t home	r: levice
9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	<pre>({ area.name }: (% - set area_info.printed = true %) (% - endif %) - {{ device_attr(device, "name") }}{% if device_attr((% - endif %) (% - endifor %) (% -</pre>	device /. Jest a	n, "mo ind su
24	Smart home: How can I assist?		
<			>
M te	innthussmodell* txt-davinci-003 laximale Anzahi von Tuken, die als Antwort zurückgegeben werden* 50		0
Тор	p*	1	
Теп	nperatur *	0,	5
		ABS	ENDEN

Das Optionen-Fenster von OpenAI

Außerdem verwendet er in seinen Antworten nicht die gewohnten Namen, die Sie Ihren Geräten zugeordnet haben, etwa Wohnzimmerlampe. Stattdessen erscheinen die eher unverständlichen Bezeichnungen, mit denen sich die Geräte bei der Installation im System angemeldet hatten. Im Beispiel meiner Wohnzimmerlampe war dies IKEA of Sweden TRADFRI bulb E27 WW 8061m.

Man kann allerdings noch einiges einstellen. Klickt man im Integrationsfenster auf *Konfigurieren*, erscheinen die *Optionen* (1).

Mit dem Schieberegler *Temperatur* stellen Sie nicht Ihre Heizung ein, sondern die Treffgenauigkeit des Chatbots: ein höherer Wert macht ihn kreativer, aber auch weniger exakt, sodass unerwartete Ergebnisse wahrscheinlicher werden. Ähnlich wirkt der Wert *Top P*. Leider findet sich bislang nirgend eine Möglichkeit, dem Chatbot Deutsch beizubringen. Obwohl man ihm Fragen in deutscher Sprache stellt, antwortet er stets englisch.

Insgesamt ist es eine Integration, die zu KI-Experimenten einlädt, im Moment aber noch keinen konkreten Nutzen hat. Aber das kann das nächste Update vielleicht schon ändern. —hgb

heise Security TOUR

01.06. KÖLN | 15.06. STUTTGART 29.06. ONLINE | 27.09. ONLINE

Wissen schützt

Security braucht Vertrauen – aber zu wem, wann und wie?

Jetzt auch mit Workshops

DIE THEMEN

- Lagebild IT-Security: Neues zu Cybercrime-Untergrund und Angriffstechniken
- Die Checkliste: Wem und was sollte ich weshalb in welchem Ausmaß vertrauen?
- Fallstricke und Lösungen beim Aufbau einer Zero-Trust-Umgebung
- Das Update zum IT-Recht inkl NIS2 und KI-Ratgeber
- Microsoft Exchange natürlich online?
- KI was macht das mit uns und der Security?

Jetzt Frühbucher-Rabatt sichern:

3D-Drucker tunen, Teil 2 Z-Achsen-Waagesensor

Diesmal geht es um den Sensor der dritten Achse des Creality Ender 3: Am genauesten lässt sich das Homing und Leveling mit der Druckdüsen-Spitze durchführen. Dazu benutzen wir eine Wägezelle ähnlich denen in der Küchenwaage als Sensor.

von Heinz Behling

eiter geht es auf der Tuning-Tour mit dem 3D-Drucker, wie im vorigen Teil am Beispiel des Creality Ender 3. In dieser Folge widmen wir uns dem Homina und Leveln der Z-Achse. Die kann im Gegensatz zu den X- und Y-Achsen nicht ohne Sensor gemessen werden. Grund dafür ist der Achsantrieb mit Gewindespindeln: Die Motortreiber-ICs auf dem Mainboard des Druckers bemerken wegen der großen Übersetzung eine Blockade durch Aufsetzen der Düsenspitze auf der Druckplatte erst viel zu spät. Dann aber ist die Kraftentfaltung schon so groß, dass im günstigsten Fall die Druckplattform in ihrer federnden Lagerung sehr weit nach unten gedrückt würde oder im schlimmsten Fall die Druckplatte oder das Hotend beschädigt würden.

Wägezelle – die Arbeitsweise

Dennoch ist es möglich, die Düsenspitze zur Bestimmung des Z-Achsen-Nullpunkts zu verwenden (genauer geht es schließlich nicht). In diesem Artikel bauen wir einen Sensor, bei dem das Hotend an einer Wägezelle (auch Wägebalken oder loadcell genannt) befestigt ist. Solche Zellen werden auch in elektronischen Waagen aller Art verwendet, ob Küchen-, Personen- oder LKW-Waage. Auch Ender-Hersteller Creality setzt etwas ähnliches bei seinen Druckern der CR6-Serie ein. Sie bestehen meist aus einem Metallbalken 1, der an einem Ende am Gehäuse der Waage (in unserem Fall am Druckkopf-Schlitten) befestigt ist, während das andere Ende eine Plattform o. ä. trägt, die das zu wiegende Gut aufnimmt. Bei einer Personenwaage ist das die Fläche, auf der man steht. In unserem Drucker wird das eine Halterung nebst Hotend sein.

Ändert sich das Gewicht (genauer gesagt die Kraft) an diesem Ende der Wägezelle, wird sie etwas gebogen. Das darf aber nur so weit gehen, dass die Wägezelle noch im Bereich der elastischen Verformung bleibt, also bei Entlastung wieder selbstständig die ursprüngliche Form annimmt. Wägezellen sind daher immer auf ein maximales Gewicht ausgelegt,

Kurzinfo

» Z-Achsen-Sensor zum Selbstbauen für Ender 3
 » Ausgleich der Temperaturdrift durch Software
 » Tipps für weitere mechanische Verbesserungen

das man nicht überschreiten sollte. Dann nämlich liefe man Gefahr, dass aus der elastischen eine plastische, also dauerhafte Verformung wird und die Wägezelle wäre irreparabel beschädigt.

Die Wägezelle für das Hotend ist für eine Belastung von 1kg ausgelegt. Das Hotend ist zwar wesentlich leichter, doch schwächere Zellen wären wegen der kleineren Abmessungen nicht stabil genug zu befestigen, um die während des Druckens auftretenden Kräfte auf Dauer auszuhalten.

im Web unter

make-maaazin.de/xe

Die Verformung muss aber noch in ein von der Drucker-Elektronik verwertbares

 Trotz der relativ massiven Bauweise reagieren solche Wägezellen auf sehr kleine Gewichte bzw. Kräfte.

elektrisches Signal umgewandelt werden. Dies geschieht mithilfe von Dehnungsmessstreifen, die im Verformungsbereich der Wägezelle kraftschlüssig aufgeklebt sind. Sie machen also die Verformung mit und ändern dabei ihren elektrischen Widerstand. Dabei wird eine sogenannte Wheatstone-Messbrückenschaltung ⁽²⁾ benutzt. Wer mehr darüber wissen möchte, findet Infos im Artikel *Kirmes-Kraftprotz-Ampel* aus Make 5/16 (im Volltext kostenlos online, siehe Kurzinfo).

Solch ein 3DTouch-Sensor unterliegt einem gewissen Verschleiß.

4 Die Teile des Druckkopfes wurden mit Tinkercad konstruiert.

S Die Muttern werden in den Hotend-Halter mit einer Schraube hineingezogen.

6 Die Wägezelle muss so in den großen Halter eingeschoben werden.

Die Messbrücke ist mit einem 24Bit-AD-Wandler (Typ HX711) verbunden. Der führt je nach Einstellung (siehe Kasten HX711 auf 80Hz umstellen) 10 oder 80 Messungen pro Sekunde durch und stellt das Ergebnis dann als 24-Bit-Zahlenwert zur weiteren Verarbeitung zur Verfügung.

Diese Kombination ist sehr empfindlich. Schon die Kraft, die das Gewicht eines einzelnen kleinen Jumper-Steckers ausübt, ist damit deutlich nachweisbar. Die Messdaten werden auf einen Arduino Pro Micro übertragen. Er nimmt die Auswertung vor. Bei einer entsprechenden Kraftausübung auf das Hotend teilt er dies der Drucker-Elektronik mit, wie das auch ein sonst üblicher Sensor (Mikroschalter, Touchsensor o. ä.) machen würde.

Wägezelle gegen Touch-Sensor

Welche Vorteile bietet nun eine Wägezelle gegenüber zum Beispiel einem Touchsensor, der die Druckplatte mithilfe einer ein- und ausfahrbaren Nadel abtastet? Der Hauptvorteil ist das Fehlen jeglicher beweglichen Teile. Gerade bei preiswerten Touchsensoren (*3D-Touch*)) verschleißt die Nadelführung recht schnell. Dadurch ergibt sich ein seitliches Spiel, sodass die Messnadel sich nicht mehr ganz senkrecht bewegt. Aus eigener Erfahrung weiß ich, dass so schon einmal Abweichungen von mehr als einem Zehntelmillimeter zustande kommen.

Das ist zu viel fürs Homing eines 3D-Druckers. Folge: Bei einem älteren, verschlissenen Sensor wird die Höhe des ersten Layers zu groß und das Filament haftet schlecht oder gar nicht auf der Druckplatte. Eine Korrektur des Z-Achsen-Versatzes in der Firmware bringt da auch kaum Besserung, da solche Sensoren eine recht schlecht Wiederkehrgenauigkeit besitzen, also bei aufeinanderfolgenden Messungen unterschiedliche Ergebnisse liefern. Etwas besser schneiden da hochwertigere Touchsensoren wie etwa der *CR-Touch* von Creality ab. Aber letztendlich haben auch sie eine bewegliche Nadel und ähnliche Probleme, wenn vielleicht auch erst etwas später.

Ein Wägebalken-Sensor hat keine beweglichen Bestandteile (wenn man von der

Während des Aushärtens sollte man die Kabel mit einer Klammer sichern.

elastischen Verformung absieht). Daher verschleißt da auch nichts. Doch so ganz problemlos sind auch sie nicht: Da ist zum einen die Temperaturabhängigkeit des Messergebnisses. Ändert sich die Temperatur, verändern auch die Dehnungsmessstreifen ihren elektrischen Widerstand, und zwar recht kräftig. Nun kann man bei einem 3D-Drucker wirklich nicht ausschließen, dass sich die Temperatur im Bereich des Hotends ändert. Da müssen Ausgleichsmethoden her. Und die finden in der Software statt. Anstelle des absoluten Messwerts wird die zeitliche Änderung des Wertes ausgewertet. Temperatureinflüsse finden langsam statt, das Aufsetzen der Düsenspitze auf die Druckplatte beim Leveln ist im Vergleich dazu um ein Vielfaches schneller. Ändert sich der Messwert also sehr rasch, ist dies das benötigte, von der Temperatur unabhängige Signal. Pate bei dieser Lösung stand das Projekt Strain-Gauge-Leveling-Probe vom Github-User sylim5442.

Zum anderen ist er sehr erschütterungsempfindlich, insbesondere, wenn die Kabel unmittelbar am Hotend sich beim Homing allzu heftig (im Millimeter-Bereich!) bewegen. Das läßt sich aber durch eine geschickte Kabelführung und -befestigung bewältigen.

Druckkopf selbstgedruckt

Um den Wägebalken am Hotend unterzubringen, wird ein komplett neuer Druckkopf benötigt, der am Wagen der X-Achse befestigt wird. Für Besitzer eines 3D-Druckers ist dessen Anfertigung ja kein Problem (sofern Sie das vor Beginn der Umbau-Arbeiten erledigen!). Über den Download-Link können Sie die Druckdateien für den Creality Ender erhalten. Falls Sie einen anderen Drucker besitzen: Rein elektronisch ist das kein Problem. Allerdings müssen Sie den Druckkopf dann an Ihren Drucker anpassen. Besonders nett wäre es, wenn Sie in diesem Fall Ihre Druckdateien dann auch anderen Make-Lesern zur Verfügung stellen würden (bitte per E-Mail an die Redaktion).

Der Druckkopf besteht aus drei Teilen 4: Dem großen Halter, der die Wägezelle, den HX711 und den Arduino aufnimmt (im Bild grün), der Hotend-Befestigung (blau) und dem Deckel für das Elektronikfach des großen Halters (gelb).

Der Druck kann mit PLA oder (meine Empfehlung) PETG erfolgen. Weitere Druckdaten:

- Halter (grün): Layerhöhe 0,2mm, Füllung 30%, Stützmaterial überall
- Deckel (gelb): Layerhöhe 0,2mm, Füllung 15%, Stützmaterial überall, Ausrichtung um 180° gedreht gegenüber dem Bild (flache Seite nach unten)
- Hotend-Befestigung: Layerhöhe 0,2mm, Füllung 100%, ohne Stützen

Nach dem Druck setzen Sie in die Hotend-Befestigung zwei M3-Muttern ein. Dazu gibt es passende sechseckige Öffnungen. Die Muttern bitte mit einer Schraube von der entgegengesetzten Seite in die Sechseck-Öffnung hineinziehen ⁽⁵⁾.

Entfernen Sie sehr gründlich sämtliches Stützmaterial aus den beiden anderen gedruckten Teilen. Einiges davon sitzt etwas versteckt im Inneren.

Als nächstes fädeln Sie die Wägezelle in den Halter ein. Bitte beachten Sie, dass die Gewindebohrungen der Zelle an beiden Seiten unterschiedlich sind. Dort, wo die vier Kabel befestigt sind, befinden sich 5mm-Gewinde, am anderen Ende sind es 4mm! Die 5mm-Gewinde dienen zur Befestigung der Zelle am Halter. Setzen Sie die Zelle mit den Drähten voran (Achtung, sehr empfindlich!) in den großen Halter wie im Bild **(**) ein.

Die Wägezelle schrauben Sie mit zwei M5×20-Schrauben mit Sechskantkopf fest.

Gut festziehen! Zur Sicherheit für die Drähtchen können Sie die mit etwas 2-Komponenten-Kleber sichern 7. Ein Tropfen Klebstoff genügt, mehr würde die Wägezelle mit dem Gehäuse verkleben.

Falls es Ihnen etwas pingelig vorkommt, dass ich die Befestigung der Kabel und Drähte hier und im weiteren Verlauf so ausführlich schildere: Die Wägezelle reagiert so empfindlich auf Erschütterungen, dass jede unnötige Kabelbewegung vermieden werden muss. Ich habe da lange dran getüftelt. Auch die überdimensionale Größe des Halters hat einen Grund: Der Schrittmotor für den Filamenttransport (wird gleich noch eingebaut) erzeugt relativ starke Magnetfelder bei einer Bewegung, die bei zu großer Nähe Störsignale in der Wägezelle und dem HX711 erzeugen. Ein paar Zentimeter Abstand helfen da.

Als Nächstes kürzen Sie die vier Drähte der Wägezelle auf etwa 6cm Länge nach der Klebestelle. Löten Sie sie an die Kontakte der

9 Die fertig zusammengelötete Elektronik: Länger sollten die Kabel nicht sein.

COM21		<u> </u>	
1			Senden
15:10:45.266 -> reading: -54445			^
15:10:45.266 -> Value: -24			
15:10:45.266 -> De: -1			
15:10:45.314 -> reading: -54345			
15:10:45.314 -> Value: -23			
15:10:45.314 -> De: 2			
15:10:45.366 -> reading: -54306			
15:10:45.366 -> Value: -23			
15:10:45.366 -> De: -1			
15:10:45.413 -> reading: -54107			
15:10:45.413 -> Value: -23			
15:10:45.413 -> De: 0			
15:10:45.467 -> reading: -54052			
15:10:45.467 -> Value: -24			
15:10:45.467 -> De: -1			
15:10:45.514 -> reading: -54295			~
Autoscroll Zeitstempel anzeigen	Neue Zeile 🗸 🗸	57600 Baud 🗸	Ausgabe löschen

🕕 Meldet der serielle Monitor solche Daten, funktioniert die Elektronik.

1 Alle Anbauteile entfernt: Gute Gelegenheit für ein Großreinemachen.

12 So sollte ein einwandfrei dichtes Hotend aussehen.

HX711-Platine entsprechend dem Schaltplan (3). Die Verbindungen zwischen der Platine und dem Arduino sollten ebenfalls kurz sein. So etwa 8cm reichen. Schließlich gehören an den Arduino noch drei weitere Schaltlitzen mit etwa 10cm Länge für die Versorgungsspannung, die Masse- und die Sensorleitung.

Die Sensorleitung führt über eine Diode (direkt in die Leitung eingelötet und mit Schrumpfschlauch isoliert), die mit ihrer Kathode zum Arduino weist. Das ist wegen der unterschiedlichen Spannungspegel von Arduino (5V) und Druckerboard (3,3V) notwendig. Der Sensor-Eingangspin des Druckerboards liegt über einen Pull-up-Widerstand an 3,3V. Gibt der Arduino an den Ausgangspin 4 High-Level (also 5V), dann ist die Kathode der Diode positiv gegenüber ihrer Anode: Sie sperrt. Legt der Arduino aber Low-Level (0V) an Pin 4, dann ist die Kathode der Diode negativ gegenüber der Anode: Sie leitet und zieht damit den Eingangs-Pin des Druckerboards ebenfalls auf 0V.

Die drei Leitungen vom Arduino zum Drucker-Board habe ich auf einen Dupont-Stecker geführt 9. Die weitere Verbindung zur Drucker-Elektronik übernimmt dann ein mindestens 1m langes Kabel, dass das entsprechende Gegenstück zum Dupont-Stecker besitzt.

Wer seinen Drucker bislang mit einem Touch-Sensor am Druckkopf benutzt hat, kann dieses Kabel weiterverwenden. Bei *BLTouch*bzw. *3D-Touch-*Sensoren entsprechen die Drahtfarben im Schaltplan (rot, schwarz, weiß) den Farben im Sensorkabel. **Achtung:** Beim *CR-Touch-*Sensor von Creality sind die Farben anders belegt: Auf der schwarzen Ader liegt dort 5V, auf weiß liegt GND und auf blau die Sensorleitung!

Bevor Sie die beiden Platinen in ihrem künftigen Kunststoff-Heim endgültig befestigen, sollten Sie sie testen. Dazu übertragen Sie mit der Arduino-IDE den Sketch z_achsen_sensor zum Arduino (Download-Adresse über Kurzinfo-Link erreichbar). Als Board wählen Sie Arduino Micro. Unter Sketch/Bibliothek einbinden/Bibliotheken verwalten suchen Sie nach HX711 und installieren die Bibliothek von Bogdan Necula (Version 0.7.5). Anschließend können Sie den Sketch kompilieren und auf den Arduino hochladen.

Ist das geschehen, starten Sie den seriellen Monitor. Sobald Sie das noch freie Ende der Wägezelle auch nur leicht berühren, muss er etwas anzeigen. Falls nicht, ist ein Fehler in der Verdrahtung. Kontrollieren und erneut checken bis alles klappt ⁽¹⁾.

Die HX-Platine und der Arduino werden im Halter mit Heißkleber befestigt. Wichtig: Beide Platinen dürfen nicht locker sein, damit es bei Kopfbewegungen keine Erschütterungen gibt. Für die USB-Buchse des Arduinos gibt es eine Öffnung im Halter. Wenn Sie beim Befestigen des Arduinos einen USB-Mikro-Stecker einstecken, wird das Einkleben einfacher und die Buchse liegt dann später auch mittig in der Öffnung.

Schließlich schrauben Sie noch den Hotend-Halter mit zwei M4×16-Schrauben ans freie Ende der Wägezelle. Damit sind die Vorarbeiten am Sensorgehäuse zunächst beendet. Jetzt geht es dem Drucker an den Kragen.

Sehr wichtig: Kabelführung

Falls noch Filament im Drucker eingelegt ist, entfernen Sie das zunächst. Ziehen Sie den Filamentschlauch aus dem Hotend (dazu den kleinen Kunststoffring am Schlaucheingang nach unten drücken). Lassen Sie danach den Drucker wieder völlig abkühlen. Fahren Sie dann die Z-Achse auf etwa 150mm Höhe. Schalten Sie den Drucker aus und ziehen Sie den Netzstecker.

Lösen Sie die beiden Schrauben (mit dem Inbus-Schlüssel aus dem Board-Werkzeug des Druckers), die das schwarze Lüftergehäuse am Hotend-Schlitten halten. Darunter befindet sich das ebenfalls mit zwei Schrauben befestigte Hotend (gleiches Werkzeug). Auch diese Schrauben lösen Sie. Achten Sie darauf, keine der zum Teil recht dünnen Kabel am Lüfter und Hotend zu beschädigen.

Anschließend bauen Sie auch den Extruder-Motor aus, der am linken Lager der X-Achse sitzt. Sein Kabel ist nur eingesteckt. Ziehen Sie den Stecker aus dem Motor. Den nun nackten Schlitten säubern Sie von anhaftendem Staub, Filamentresten und ähnlichem, was sich dort abgelagert hat 1.

Bei der Gelegenheit sollten Sie auch das Hotend unter die Lupe nehmen. Ziehen Sie die Silikon-Isolierung vom Heizblock. Ein einwandfreies Hotend ist darunter völlig sauber (2). Falls das Silikon daran klebt, ist das ein schlechtes Zeichen. Meist ist dann an der Düse oder an dem kleinen Verbindungsrohr zwischen Heizblock und Kühlkörper etwas undicht. Oft ist das die Folge, wenn die Düsenspitze mal einen heftigen Kontakt mit einem Gegenstand auf der Druckplatte hatte.

Falls auch noch die beiden Schrauben, die den Heizblock am Kühlkörper halten, schief sind, sollten Sie auf jeden Fall ein neues Hotend verwenden. Das bekommt man nämlich nicht mehr dicht. Aber auch bei geraden Schrauben würde ich das Hotend wechseln.

Den Extruder inklusive Schrittmotor schrauben Sie nun am Elektronik-Gehäuse an. Der Stecker des Motors an der Rückseite muss nach oben zeigen, das Anschlussstück für den Filamentschlauch an der Vorderseite nach unten in Richtung Hotend-Halter. Diese Einheit setzen Sie dann auf den Druckkopf-Schlitten ().

Befestigen Sie das Gehäuse zunächst provisorisch mit zwei M3×16-Schrauben. Als nächstes müssen Sie den Filamentschlauch passend zuschneiden. Bei meiner Hotend-/Extruder-Kombination sind 85mm Länge notwendig. Verwenden Sie den schwarzen Serien-Kunststoff-Extruder, kann eine andere Länge nötig sein. Schneiden Sie daher zunächst ein etwa 10cm langes Stück Schlauch zu. Wichtig: Die Schnittkanten müssen senkrecht zur Schlauchrichtung sein!

Stecken Sie das Schlauchstück ins Hotend und setzen Sie den Schlauch dann in das Gegenstück am Extruder ein. Der Schlauch hat dann die richtige Länge, wenn die beiden Schraubenlöcher des Hotends etwa 1 bis 2mm über den entsprechenden Löchern im Halter liegen. Der Schlauch muss etwas kürzer sein, damit er später bei einer Temperaturänderung keinen Druck auf den Wägebalken ausübt (2).

Ist die richtige Schlauchlänge gefunden, ziehen Sie den Schlauch wieder etwas aus dem Extruder, nicht aus dem Hotend! Liegen die Löcher im Halter und im Hotend übereinander, schrauben Sie das Hotend mit zwei M3×18-Schrauben am Halter fest.

Obwohl diese Art der Extruder-Hotend-Verbindung kein reinrassiger Direktantrieb

🚯 Das Elektronikgehäuse passt exakt auf die beiden Schraubenlöcher im Wagen der X-Achse.

ist, wie er zur Verarbeitung von elastischen Filamenten notwendig ist, erfüllt sie trotzdem denselben Zweck: Da das Filament um keinerlei Kurven innerhalb des Schlauches geführt wird, verklemmt es sich auch nicht auf dem kurzen Weg. Sie können damit künftig also auch gummiartiges Material verarbeiten. Prinzipiell könnte man den Halter auch so konstruieren, dass der Ausgang des Extruders direkt auf dem Eingang des Hotends säße. Dann käme man ohne Schlauch aus und hätte wirklich einen Direktantrieb. Allerdings sitzt der Schrittmotor dann sehr nahe an der Wägezelle, was wiederum stärke Einstreuungen von Störsignalen bewirkt. In meinen Versuchen ließ sich dann die geringe Kraft beim Aufsetzen der Düse auf die Druckplatte nicht mehr sicher registrieren. Folge: Die Z-Achse stoppte beim Homen zu spät und beschädigte die Druckplatte. Daher diese Bauweise mit mehr Abstand.

Jetzt können Sie die provisorische Verschraubung der Halterung am Wagen wieder lösen und dann Halterung und Lüftergehäuse zusammen befestigen. Achten Sie dabei darauf, dass die Kabel für das Heizelement, den Thermistor und den Lüfter innen im Lüftergehäuse so geführt werden, dass sie das Hotend nicht berühren (außer natürlich dort, wo Thermistor und Heizung sitzen) [].

U Zwischen rotem Kühlkörper und Kabeln muss ein Abstand bleiben.

Wenn die Schraubenlöcher so zueinander passen, hat der Schlauch die richtige Länge.

10 Diese Kabelführung hat sich in zahlreichen Versuchen als brauchbar erwiesen.

O An der Zahnriemenabdeckung sollte der Kabelbaum so befestigt werden.

Das Kabel des Extruder-Motors muss noch zum Motor hingeführt werden. Dafür ist es aber leider zu kurz. Entweder Sie verlängern es, indem Sie jede der vier Adern durchtrennen und etwa 20cm Schaltlitze dazwischen löten. Oder sie verwenden ein fertiges, längeres Kabel (siehe Bezugsquellen). Dazu müssen Sie das Elektronik-Gehäuse des Druckers öffnen, das alte Kabel entfernen und das neue einfädeln. Ich hab es auch gleich durch den Gewebeschlauch mit den anderen Leitungen hindurchgezogen. Bei der Gelegenheit sollten Sie auch das Thermistorkabel um etwa 15cm verlängern, sonst wird es knapp.

Am Druckkopf kommt es sehr auf die Kabelführung an. Das Kabel darf keine Kräfte auf den Hotend-Halter und damit auf die Wägezelle ausüben. Daher muss es zwischen dem großen Elektronikgehäuse und dem Hotend reichlich Luft haben. Andererseits dürfen sich aber auch bei Bewegung des Druckkopfwagens oder der X-Achse keine Erschütterungen vom Kabel auf die Wägezelle weiterleiten. Das Kabel muss daher am großen Elektronikgehäuse sehr gut befestigt sein. Dafür sind dort Schlitze für Kabelbinder eingearbeitet. Befestigen Sie das Kabel wie im Bild **(5**).

Außerdem muss das Kabel zwischen Druckkopf und dem Mainboard des Druckers so

Tipp: Doppelspindel-Antrieb für Z-Achse

Eine weitere gute Tuning-Maßnahme für den Ender 3 ist der Einbau einer zweiten Gewindespindel inklusive Schrittmotor für die Z-Achse. Serienmäßig hat er ja nur auf der linken Seite einen Antrieb. Das funktioniert auch recht gut, solange die Rollenlager auf beiden Seiten des X-Achsenbalkens spielfrei, aber leichtgängig eingestellt sind. Das wiederum gelingt nur, wenn der Rahmen absolut rechtwinklig aufgebaut ist. Die beiden senkrechten Säulen müssen also in allen Richtungen senkrecht auf ihren Träger stehen, parallel zueinander sein und die X-Achse muss auch senkrecht zu den Säulen angeordnet sein. Das sollten Sie einmal mit einem großen Dreieck prüfen und bei Abweichungen unbedingt justieren. Es wird nämlich spätestens im nächsten Teil der Tuning-Reihe wichtig: Dort wird mithilfe der Software Klipper die Druckgeschwindigkeit deutlich erhöht. Und dazu sollte die Mechanik einwandfrei sein.

Zur Lagereinstellung ist eine der jeweils drei Rollen jeder Seite exzentrisch gelagert. Mit einem Schraubenschlüssel kann man den Sechskant dieser Achsen drehen und somit die entsprechende Rolle etwas näher oder weiter zum Alu-Profil justieren. Ist auf der rechten Seite auch nur ein geringes Spiel vorhanden, hängt die X-Achse dort etwas herab. Das macht sich im Druckbild bemerkbar. Besonders der für die Haftung auf der Druckplatte so wichtige erste Layer wird rechts dünner. Mit der zweiten Gewindespindel rechts verhindern Sie das.

Die Erweiterung gibt es bei Creality für knapp 50 Euro. Im Netz findet man auch ab und zu günstigere Angebote. Achten Sie aber darauf, dass es sich um die Original-Erweiterung von Creality handelt. Sie enthält nämlich exakt passende Motorhalter, eine Halterung fürs Netzteil, einen neuen Rollenschlitten inklusive Spindelmutter für die rechte Seite und obere Gegenlager für beide Gewindespindel. Außerdem erhalten Sie damit ein Y-Kabel, mit dem Sie beide Schrittmotoren auch an ältere Mainboards mit nur einem Anschluss für die Z-Achse verbinden können. Der Einbau ist einfach, eine entsprechende kurze Bilderanleitung liegt bei.

Für die zweite Gewindespindel muss das Netzteil nach hinten versetzt werden.

geführt werden, dass es nirgends am Rahmen des Druckers schleift oder hängen bleiben kann. Klingt kompliziert, ist aber durch einfaches Befestigen des Kabels an der Zahnriemenabdeckung mittels Klebesockel und Kabelbinder der X-Achse links zu schaffen (). Falls sich die nicht immer zuverlässigen Selbstklebepads später lösen: Entfernen Sie sie und verwenden stattdessen 2-Komponenten-Kleber.

Drucker-Firmware anpassen

Die Drucker-Firmware muss in einigen Punkten angepasst werden. Im einzelnen betrifft dies diese Einstellungen:

- Geschwindigkeit der z-Achse beim Homing und Leveln
- Temperatur beim Homing
- Entfernen des Sondenversatzes in allen drei Richtungen
- Höheneinstellung zwischen den Messungen verringern

Die Geschwindigkeit, mit der beim Homing und Leveln die Achse nach unten geführt wird, muss deutlich geringer als bei herkömmlichen Sensoren sein. Dadurch braucht der Antrieb weniger Zeit (gleich Wegstrecke) zum Abbremsen und so wird die Messung deutlich genauer.

Außerdem muss mit aufgeheizter Druckdüse gemessen werden. Der Grund: An der Düsenspitze haftet immer ein kleiner Filamentrest. Der würde die Messung verfälschen, denn dadurch läge die Düsenspitze ja deutlich tiefer. Heizt man vor der Messung jedoch die Düse auf (200°C), ist dieser Rest weich und wird beim Homing einfach zusammengedrückt.

Da wir die Düsenspitze als Sonde verwenden, existiert natürlich auch kein Sondenversatz mehr, und zwar weder in X-, Y- noch in Z-Richtung. Auch dies muss in der Firmware eingestellt werden.

Schließlich dient die Verringerung der Höhenbewegung beim Ausmessen der Druckplatte der Beschleunigung der Messungen: Die Düsenspitze wird nur noch 2mm angehoben statt bislang 5mm.

Die Änderungen nehmen wir mit PlatformIO vor. Die Marlin-Firmware mit den entsprechend angepassten Dateien liegen auf Github bereit (siehe Kurzinfo-Link). Nach dem Download und Entpacken die Firmware als Projekt in PlatformIO öffnen und kompilieren. Die fertige Firmware firmware.bin dann auf eine Speicherkarte kopieren, in den Drucker einlegen und den einschalten. Danach noch im Druckermenü auf Konfiguration, erw. Konfiguration und Werkeinstellungen gehen, damit die neuen Vorgabewerte auch eingelesen werden. Wer mehr dazu wissen möchte, sollte sich das Video zu diesem Artikel ansehen. Dort zeige ich Ihnen, wie Sie die Firmware ändern und Sie können dort den Drucker auch in Aktion sehen. Solche zusätzlichen Firmware-

HX711 auf 80Hz umstellen

Platinen mit dem HX711 werden im Internet massenweise angeboten. Oftmals steht unter den Daten dann auch etwas wie "80Hz-Version". Ich habe Boards aus solchen Angeboten beschafft. Sie waren sämtlich auf 10Hz (also 10 Messungen pro Sekunde) eingestellt. Wenn Sie ein solches Board nachprüfen wollen: Liegt im Betrieb an Pin 15 des HX711 (das ist der zweite Pin von oben an der rechten Seite des ICs im Bild) 0 Volt an, dann arbeitet er mit 10Hz. Für 80Hz muss dort die Betriebsspannung liegen.

Auch das in den Bezugsquellen (siehe Kurzinfo-Link) genannte Board ist bei Lieferung auf 10Hz eingestellt. Es lässt sich aber im Gegensatz zu vielen anderen per Lötbrücke umschalten. Dazu muss zunächst der Sandkorn-große 0-Ohm-SMD-Widerstand oberhalb der Beschriftung 10Hz entfernt und daneben oberhalb von 80Hz die beiden Kontaktflächen durch einen Lötzinntropfen überbrückt werden. B++++ XFW-HX711

Änderungen können eventuell notwendig sein: So sollten Sie, falls Sie mit Hochtemperatur-Filamenten wie PETG oder Polyamid arbeiten, das Homing bei noch höherer Temperatur ausgeführen.

Erster Test

Jetzt ist der 3D-Drucker bereit für den ersten Test. Damit bei einem eventuellen Fehler die Druckdüse nicht rücksichtslos ins Druckbett gerammt wird, gehen wir so vor: Nach dem Einschalten des Druckers in dessen Menü die Z-Achse auf 150mm Höhe stellen. Danach Bewegen und Autohome wählen. Der Homing-Vorgang beginnt. Lassen Sie die Z-Achse einige Zentimeter nach unten fahren. Berühren Sie dann mit einem Werkzeug die Düsenspitze (nicht mit der Fingerspitze, die Düse ist jetzt heiß). Die Z-Achsen-Bewegung muss sofort stoppen und wieder einige mm nach oben fahren. Entfernen Sie das Werkzeug, noch bevor die Z-Achse wieder zur zweiten Messung nach unten fährt. Nach einigen Zentimetern erscheint eine Fehlermeldung auf dem Druckerdisplay und der Drucker geht in den Sicherheitsstopp. Schalten Sie den Drucker aus.

Wenn das nicht so abläuft, dann arbeitet der Sensor nicht korrekt: Meist ist dann an der Verkabelung etwas falsch angeschlossen oder die Kabelführung nicht korrekt. Dass der Sensor technisch funktioniert, haben Sie ja bereits zuvor gecheckt. Wenn alles okay ist, Drucker wieder einschalten und dann ein echtes Homing durchführen. Die Düsenspitze sollte ganz sanft auf der Druckplatte aufsetzen. Danach die Bettnivellierung aufrufen. Hier wird das Druckbett an 16 Punkten abgetastet. Falls der Druckkopf weniger Punkte anfährt, erhält er während der Bewegung ein fehlerhaftes Signal. Das rührt dann vom Kabelbaum zum Druckkopf her, das irgendwo auf dem Weg schleift oder hängenbleibt. Korrigieren und nochmals leveln und nivellieren, bis es klappt.

Ist der Test schließlich ordnungsgemäß verlaufen, ist der Drucker einsetzbar. Machen Sie einen kurzen Probedruck. Eventuell müssen Sie noch einen kleinen Z-Sondenversatz einstellen, falls der erste Layer zu dünn ausfällt. Der Versatz entspricht der Höhe, um die sich die Wägezelle beim Homing verformt. Bei meinem Drucker sind es 6/100 Millimeter.

Eins noch: Der Sensor ist sehr empfindlich gegenüber Erschütterungen. Vermeiden Sie es daher während des Homings/Nivellierens, das Gerät zu berühren oder an den Tisch/ Schrank zu stoßen. Während des Druckens ist das aber kein Problem mehr.

Und nun erstmal viel Spaß mit dem neuen Sensor. In der nächsten Folge geht es dann weiter wie beim Auto-Tuning: Ist die Mechanik erstmal in Ordnung, kann man an höheres Tempo rangehen. Mit der Software *Klipper* werden wir dann dem Drucker Beine machen. —hgb

Die HX711-Platine unten auf 10Hz, oben umgelötet auf 80Hz eingestellt.

StarFive VisionFive 2 RISC-V-Einplatinencomputer

Weltpremiere: Der VisionFive 2 aus China ist der erste bezahlbare Raspberry-Pi-Konkurrent mit Linux-tauglichem RISC-V-Prozessor. Der Test zeigt jedoch, dass die Software noch viel Arbeit braucht.

von Christof Windeck

ie offene Befehlssatzarchitektur RISC-V weckt Hoffnungen auf Alternativen zu den allgegenwärtigen ARM- und x86-Prozessoren. RISC-V-Technik ermöglicht im Prinzip die Entwicklung komplett offengelegter Chips und ist ohne einschränkende Lizenzbedingungen nutzbar. Doch erst jetzt, 13 Jahre nach der RISC-V-Vorstellung, kommt mit dem StarFive JH7110 ein System-on-Chip (SoC) mit RISC-V-Technik, das es halbwegs mit jenen ARM-SoCs aufnehmen kann, die auch den Raspberry Pi antreiben. Mit dem JH7110 bestückt die chinesische Firma StarFive ihren Einplatinencomputer VisionFive 2; er orientiert sich unübersehbar am Raspi, allerdings ist seine Platine rund 60 Prozent größer.

Seit Ende Dezember liefert der Hersteller die ersten Vorserienmuster aus, die man im August 2022 als "Early Bird"-Teilnehmer einer Kickstarter-Finanzierung bestellen konnte. Bald soll der VisionFive 2 auch im Onlinehandel erhältlich sein – etwa bei Amazon. Wir konnten eines der Vorserienmodelle testen. Der Unterschied zum Serienprodukt: Nur einer der beiden Ethernet-Ports ist gigabitfähig. Aber das ist nicht der einzige Haken des RISC-V-Vorboten.

Gemeinschaftsprojekt

StarFive steuert auch den wichtigsten Chip zum VisionFive 2 bei, den JH7110. Darin rechnen vier RISC-V-Kerne vom Typ U74-MC, die das US-Unternehmen *SiFive* entwickelt hat. StarFive wurde von SiFive mitgegründet, um die RISC-V-Technik in China voranzutreiben. Der Grafikkern im JH7110 stammt wiederum von der britischen Firma *Imagination Technologies*, deren PowerVR-GPUs in den ersten iPhones steckten. Imagination gehört seit einigen Jahren chinesischen Investoren. Die Fertigung des JH7110 übernimmt der taiwanische Gigant TSMC mit 28-Nanometer-Technik.

Im JH7110 sitzen noch zwei weitere RISC-V-Kerne, die interne Steuerungsaufgaben erledigen. Außerdem sind wie bei vielen modernen Systems-on-Chip (SoCs) mit ARM-Kernen zahlreiche weitere Controller eingebaut, etwa für Gigabit Ethernet, PCIe 2.0, USB 2.0, I²C, SPI und SDIO. Zudem gibt es Videode- und -encoder sowie einen Audio-DSP.

Der VisionFive 2 hat auch die Raspi-typische GPIO-Leiste mit 40 Pins, aber ihr fehlen einige Funktionen wie etwa I²S. Außerdem gibt es keinen WLAN-Chip auf dem Board. Anders als der Raspi hat der VisionFive 2 auf der Unterseite eine M.2-Fassung für eine SSD. Sie ist freilich nur mit einer einzigen PCIe-Lane angebunden, an der zweiten hängt der USB-3.0-Controller. Der JH7110 verwendet LPDDR4-2800-Speicher, der niedriger taktet als etwa beim Raspi 4. Unser Vorserienmodell war mit 4 GByte bestückt, Varianten mit 2 und 8 GByte sind geplant. Einen Monitor steuert der VisionFive 2 über eine normal große HDMI-Buchse an. Strom liefert man ihm per USB-C-Buchse. Ein 5-Volt-Netzteil genügt, aber wenn man eines mit USB Power Delivery (USB-PD) anschließt, nimmt das Board auch 9, 12 oder 15 Volt. Das klappte im Kurztest allerdings nicht mit jedem USB-PD-Netzteil. Die Platine des VisionFive 2 ist minimal größer als das PicoITX-Format (10cm \times 7,2cm).

Genau wie ein Raspi bootet der VisionFive 2 von einer MicroSD-Karte. Es ist aber auch ein Steckplatz für ein eMMC-Modul vorhanden. Booten von USB und M.2-SSD ist bisher nicht möglich.

Hakeleien

Die vier U74-Kerne sind zur RISC-V-Spezifikation RV64GC kompatibel, die sich als Basisanforderung für Linux etabliert hat. Einige Linux-Distributionen arbeiten bereits an angepassten Varianten für RISC-V-Boards, der VisionFive 2 dürfte eines Tages etwa unter Ubuntu laufen. Doch so weit ist es noch nicht.

Bisher muss man wie bei vielen Rockchip-Bastelboards ein vom Hersteller handgeklöppeltes Linux-Image herunterladen und auf eine MicroSD-Karte schreiben – etwa mit *Balena-Etcher*. StarFive liefert das Image auf Basis von Debian (mit angepasstem LTS-Kernel 5.15.0 und XFCE-Desktop) entweder per Torrent oder via Google Drive aus, in China per Baidu. Leider ist es über 6 GByte fett und kommt entpackt auf 16 GByte. Man braucht folglich mindestens eine 16-GByte-Speicherkarte.

Beim Vorserienmuster bootet das aktuelle Image Version 69 nicht einmal. Zuvor muss man in einer komplizierten Prozedur den Bootloader im SPI-Flash updaten. Das soll bei den Serienboards nicht mehr nötig sein.

Kurzinfo

- » RISC-V ist eine offene Systemarchitektur, die Flexibilität und Sicherheit verspricht
- » Bisher implementiert nur wenig Hardware RISC-V

Mehr zum Thema

- » Dieser Artikel erschien zuerst in c't 5/23 auf Seite 74
 » Christof Windeck, Flexiprozessor, Das Poten-
- zial von RISC-V-Prozessoren, c't 19/19, S. 134

Über die serielle Konsole des VisionFive 2 kann man die Bootloader U-Boot und OpenSBI steuern.

Auf seiner Unterseite hat der StarFive Vision-Five 2 (VF2) eine M.2-Fassung für eine NVMe-SSD im Format M.2 2280. Die Platine (links) hat ungefähr PicoITX-Format, ist also deutlich größer als die des Raspberry Pi (rechts) im Bezahlkartenformat.

StarFive VisionFive 2

Einplatinencomputer mit RISC-V-Prozessor					
Hardware-Ausstattung					
CPU / Kerne / Takt	StarFive JH7110 / 4 × SiFive U74 (RV64GC) / 1,5 GHz				
RAM	4 GByte LPDDR4-2800				
Grafik (-speicher)	Imagination IMG BXE-4-32 (vom Hauptspeicher)				
SSD-Steckplatz	1 × M.2 (PCle 2.0 x1)				
Sound-Chip	Audio-DSP im JH7110				
Netzwerk-Interface (Chip) ¹	1 × 1 Gbit/s + 1 × 100 Mbit/s (PHYs: Motorcomm YT8531C & YT8512C)				
USB-3.0-Host- controller	VIA Labs VL805 (4 × USB 3.2 Gen 1 (USB 3.0))				
Power Management IC (PMIC), Flash	X-Powers AXP15060, GigaDevice GD25LQ128ESIG (16 MByte QSPI)				
Abmessungen	8,0 cm \times 10,1 cm, (Platine 7,4 cm \times 10 cm), inklusive Buchsen und M2-Fassung 2,3 cm hoch				
Spannungsversor- gung	USB-C, 5 bis 20 Volt, 15 Watt (USB-PD 2.0)				
Anschlussbuchsen	1 × HDMI, 4 × USB-A 5 Gbit/s, 2 × LAN, 1 × Audio-Klinke, 1 × USB-C (Strom)				
Anschlüsse onboard	40-pol GPIO, 1 × MIPI-CSI, 2 × MIPI-DSI, 1 × PoE, 1 × Lüfter, Reset-Taster				
Steckfassungen / Anschlüsse unten	$1 \times$ MicroSD, $1 \times$ M.2, $1 \times$ eMMC (proprietär)				
Elektrische Leistungsaufnahme, Datentransfer-Messungen					
Soft-Off / Energie Sparen	4,5 W ² / -				
Leerlauf mit Full-HD- Monitor an HDMI	3,8 W (ohne Display: 3,3 W; mit Display & Ethernet: 4,3 W)				
Volllast: CPU / CPU und Grafik	6,6 / 7,7 W				
M.2-SSD lesen (schreiben)	192 (152) MByte/s				
USB 3.0 lesen (schrei- ben) ³	>150 (>50) MByte/s				
LAN RX (TX)	960 (940) Mbit/s				
Systemleistung					
openSSL AES-128 ohne / mit AES-NI	9,1 / 9,0 Mbit/s				
7-Zip komprimieren / dekomp.	2446 / 5543 MIPS				
Preis / Garantie	$90 \notin /k$ A				

¹ die Serienprodukte sollen 2 × Gigabit-Ethernet haben \checkmark funktioniert – funktioniert nicht ² Soft-off funktioniert anscheinend nicht richtig ³ keine Standardmessung möglich, USB-SSDs nicht erkannt

Das Image braucht 35 Sekunden zum Booten, dann meldet man sich als *root* mit dem vorgegebenen Passwort *starfive* an, die Lokalisierung ist auf Englisch voreingestellt. Das hängt weit hinter dem Komfort und den Sicherheitsvorgaben großer Distributionen zurück.

An einem 4K-Monitor lieferte das Board kein Bild, aber mit einem Full-HD-Bildschirm klappte es problemlos. Die XFCE-Oberfläche reagiert allerdings so träge, dass es schwierig ist, Symbole mit der Maus anzuklicken. Anscheinend beschleunigt der Grafikkern die 2D-Darstellung nicht. 3D-Beschleunigung per OpenGL-ES funktionierte hingegen und der Benchmark glmark2-es2 erkannte die GPU korrekt.

Der Browser *Firefox* baute Webseiten sehr langsam auf. YouTube-Videos liefen zwar, ruckelten aber selbst in 360p-Auflösung. Die Audio-Klinkenbuchse lieferte schlechte Signalqualität. Dem HDMI-Ausgang konnten wir kein Audiosignal entlocken, weil es im *PulseAudio*-Gebälk knirschte.

Die Netzwerkports arbeiten mit der erwarteten Geschwindigkeit. Ein USB-3.0-Stick übertrug Daten mit mehr als 150 MByte/s. Genauer konnten wir nicht messen, weil keine unserer USB-SSDs erkannt wurde. Eine Intel SSD 660p im M.2-Slot lieferte knapp 200 MByte/s; mit manchen anderen NVMe-SSDs gibt es wohl I/O-Probleme.

Laut SiFive sind die 2018 angekündigten U74-Kerne grob mit ARM-Kernen vom Typ Cortex-A55 vergleichbar. Allerdings haben die U74-Kerne wenig Gleitkommarechenleistung und keine mit ARM Neon vergleichbare SIMD-Rechenwerke. Dem Broadcom BCM2711 eines Raspi 4 mit vier ARM Cortex-A72 hinkt der JH7110 also weit hinterher, selbst der BCM2837 des Raspi 3 (4 × Cortex-A53) fühlt sich deutlich schneller an.

Benchmarks bestätigen den Eindruck: Beim Komprimieren und Dekomprimieren mit 7-Zip ist der Raspi 4 mehr als doppelt so schnell wie der JH7110 und bei AES-Verschlüsselung mit openSSL sogar neunmal so schnell.

Doch das ist eine Momentaufnahme. Es ist klar, dass die meiste Linux-Software noch nicht für RV64GC-Kerne im Allgemeinen und die Eigenheiten des JH7110 im Besonderen angepasst ist. StarFive unterstützt Entwickler mit vielen PDF-Dokumenten und direkter Hilfestellung über ein Forum auf dem RISC-V-Portal RVspace (siehe Link in der Kurzinfo).

Einige Entwickler verlinken im Forum eigene Linux-Images auf Basis von Ubuntu, OpenSuse und ArchLinux, siehe Link. Diese sind aber nur über eine serielle Konsole von einem anderen Rechner aus bedienbar, indem man einen USB-Seriell-Adapter an den UART-Ausgang anschließt (Pins 8 und 10 der GPIO-Leiste plus Masse an Pin 6). Darüber kann man auch den Bootloader U-Boot bedienen.

Closed-Source-Chip

RISC-V ist zwar eine offengelegte *Instruction Set Architecture* (ISA) und SiFive wirkt eifrig in der RISC-V Foundation mit. Doch weder die Schaltpläne der U74-Kerne, noch Details zum StarFive JH7110 sind offengelegt. Auch das Board ist keine Open-Source-Hardware und der GPU-Kern braucht proprietären Binärcode. Imagination Technologies verspricht zwar quelloffene Treiber, nennt dafür jedoch keinen konkreten Zeitplan.

Ein von Preis und Leistung zum chinesischen JH7110 vergleichbares RISC-V-SoC einer europäischen oder US-amerikanischen Firma ist bisher nicht in Sicht. Anscheinend arbeitet aber etwa die Firma NXP an RISC-V-Kernen. Doch der Traum eines offengelegten und somit besonders vertrauenswürdigen Chips, der Linux (oder BSD) ausführen kann, ist noch in weiter Ferne.

Fazit

Es funktioniert: Linux läuft auf dem RISC-V-SoC StarFive JH7110. Der VisionFive 2 taugt bisher aber noch nicht für Bastelprojekte oder gar als Desktop-PC-Ersatz. Es hakelt und ruckelt an allen Ecken. Dem Umfang und der Reife des Raspberry-Pi-Ökosystems hinkt der RISC-V-Rechner meilenweit hinterher. Jetzt müssen erfahrene Linux-Entwickler ran und fleißig Software optimieren. Erst, wenn eine größere Linux-Distribution wie Ubuntu, Fedora oder Debian kompatible Images liefert, wird der VisionFive 2 für Schrauber attraktiv. Trotzdem ist er ein Meilenstein der RISC-V-Geschichte: Endlich steht bezahlbare RISC-V-Hardware für Linux bereit. —ciw@ct.de

Die Heise-Konferenz zu Machine Learning und Künstlicher Intelligenz

9. – 11. Mai 2023 in Karlsruhe

Die Konferenz zu Machine Learning und KI

Die Minds Mastering Machines ist die Konferenz für Fachleute, die Machine-Learning-Projekte in die technische Realität umsetzen.

Das Programm bietet an zwei Tagen 36 Vorträge unter anderem zu folgenden Themen:

- Resilientes Machine Learning
- Komplexität in ML-Projekten reduzieren
- Kontinuierliches Training mit Active-Active-Architekturen
- Large Language Models auf eigene Daten anwenden
- Data-Science-Teams mit Kubeflow skalieren
- Föderiertes Lernen
- MLOps mit Argo und Kubernetes
- Erkennen von Bildmanipulationen

www.m3-konferenz.de

Workshops am 9. Mai: »Schritt für Schritt zur Erklärbaren KI« und »MLOps – wie bringt man ein ML-Modell in Produktion und hält es dort?«

Veranstalter

Goldsponsor

Jetzt Tickets sichern!

© Copyright by Maker Media GmbH.

Technische Zeichnungen mit FreeCAD, Teil 3 (und Schluss)

Auch wenn man Objekte in 3D auf dem PC konstruiert, benötigt man manchmal einen klassischen Plan auf Papier, etwa wenn man ein Teil in Auftrag geben will. Wie man den mit der kostenlosen Software FreeCAD erzeugt, zeigen wir in unserer kleinen Artikelserie. In dieser Folge werden Teile bemaßt.

von Matthias Mett

nser Beispielprojekt - ein universeller Flaschenverschluss – ist in 3D entworfen und gezeichnet, eine Zeichnungsvorlage angelegt und das Schriftfeld bereits ausgefüllt: das war der Stoff des ersten Teils unserer Artikelserie. Im zweiten Teil haben wir Ansichten und Schnitte erzeugt und aufs Blatt verteilt, jetzt sind die Bemaßungen dran. Denn die ersparen es beim Bauen nach einem gedruckten Plan, Distanzen, Durchmesser, Winkel und ähnliches mühsam vom Papier abmessen zu müssen. Wie immer zeigen wir in einem bealeitenden Video (zu erreichen über den Link in der Kurzinfo) das Vorgehen in FreeCAD Klick für Klick.

Maße nach DIN

Beim technischen Zeichnen gibt es sogenannte Bemaßungsregeln, welche die richtige Vorgehensweise beim Bemaßen beschreiben. Diese sind für die Industrie in der DIN-Norm 406 festgelegt. Für die Maker-Werkstatt ist es natürlich nicht entscheidend, dass die technische Zeichnung absolut normgerecht ist für uns geben diese Bemaßungsregeln nur Hinweise. Wichtiger ist, dass die Zeichnung übersichtlich ist und man in der Werkstatt selbst gut mit ihr zurechtkommt.

Aber ein Blick in die Profi-Ecke bringt durchaus Erkenntnisse, denn die Regeln sollen sicherstellen, dass die Maßinformationen möglichst klar erkennbar sind. Je nach Zweck unterscheidet man beim industriellen Zeichnen grundsätzliche Vorgehensweisen oder Systeme fürs Eintragen der Maße: die funktionsbezogene Bemaßung, die prüfbezogene Bemaßung und die fertigungsbezogene Bemaßung. Letztere ist für den Maker-Alltag in aller Regel passend, denn anschaulich gesagt liefert die fertigungsbezogene Bemaßung die wichtigen Informationen, um nach der technischen Zeichnung das Werkstück ganz praktisch anzufertigen: Auf welche Länge wird ein Werkstück abgesägt, mit welchem Radius wird eine Kante gerundet, mit welchem Bohrerdurchmesser ein Loch gebohrt und wie weit ist der Mittelpunkt der Bohrung von den Werkstückkanten entfernt? Ist die Zeichnung, nach der man arbeitet, gut bemaßt, kann man das alles direkt als Zahlen ablesen, ohne in der Zeichnung nochmal messen zu müssen.

Anatomie der Bemaßungen

Die Maßelemente bestehen aus einer Maßlinie mit dem Maßpfeil, einer Maßzahl und zwei Maßhilfslinien 1.

Maßhilfslinien verbinden den Körper mit der Maßlinie. Die Maßhilfslinien ragen jeweils 2mm über die Maßlinie hinaus. Die Maßlinie verbindet die Maßhilfslinien und hat als Endpunkte entweder Maßpfeile oder aber auch Punkte. Die Maßzahl steht in der Mitte auf der Maßlinie und man fügt ihr keine Einheiten an,

Kurzinfo

- » Technische Zeichnung mit Ansichten und Schnitten aus einem 3D-Modell erzeugen
- » Einfacher Flaschenverschluss als Beispielprojekt
- » Detaillierte Klick-Anleitung als Video zum Artikel

Mehr zum Thema

- » Matthias Mett, Technische Zeichnungen mit FreeCAD, Teil 2, Make 1/23, S. 106
- Matthias Mett, Technische Zeichnungen mit FreeCAD, Teil 1, Make 6/22, S. 100
- Eine verlinkte Liste aller weiteren FreeCAD-Artikel aus der Make gibt es
- online unter der Kurz-URL. Video: Technische Zeichnungen

sofern es sich um Millimeter handelt. Hingegen sind vorangestellte Zeichen wie Ø für Durchmesser oder R für Radius üblich, ebenso das Gradzeichen bei Winkeln. Maßpfeile sollen etwa dreimal so lang wie breit sein und sind immer ausgefüllt. Beim Zeichnen mit Bleistift verwendet man für die Maßpfeile gerne Schablonen, mit ein wenig Übung kann man die aber auch gut freihandzeichnen.

Bemaßungslinien und Maßhilfslinien zeichnet man immer dünn, damit man das Werkstück besser erkennen kann. Die erste Maßlinie soll einen Abstand zum Werkstück von 10mm haben und jeweils jede weitere darüber oder darunter 7mm zur vorherigen Maßlinie (2). Die Maßzahlen sollen von unten oder von rechts lesbar sein 3.

Staffelung zur Übersicht

Um mit einer Bemaßung zu beginnen, wählt man eine Maßbezugskante 4. Diese stellt idealerweise eine Seite des Werkstücks dar. Wenn man später in der Werkstatt alle Maße eines Werkstücks von derselben Bezugskante aus misst und anzeichnet, summieren sich die unvermeidlichen Messungenauigkeiten nicht auf.

Für Breitenmaße braucht man eine Maßbezugskante links oder rechts, für die Höhen-

maße oben oder unten. Dann startet man von der gewählten Maßbezugskante mit einer Maßhilfslinie und zeichnet die nächste Maßhilfslinie an der nächsten Kante und vervollständigt die Bemaßung mit Maßlinie, Maßpfeil und Maßzahl. Bei der nächsten Bemaßung geht

Maßbezugskanten von links und von oben

man wieder von der Maßbezugskante aus und gibt das Maß bis zur nächsten weiteren Kante an. So staffelt man die Bemaßungen übereinander, was die Übersichtlichkeit erhöht **3 4**.

Ganz systematisch betrachtet lassen sich noch verschiedene Maßarten unterscheiden: Das Grundmaß beschreibt die Größe des Werkstücks oder Grundkörpers. Das Lagemaß zeigt den Abstand einer Aussparung zur Körperkante und legt damit deren Lage fest. Das Formmaß hingegen gibt die Größe der Aussparung an (5). Man kann diese Systematik etwa benutzen, um zu prüfen, wann eine Zeichnung ausreichend bemaßt ist: Das ist der Fall, wenn die Grundmaße des Werkstücks sowie für alle Aussparungen Lage und Größe durch Maßzahlen direkt ablesbar sind.

Don'ts

Maßlinien sollten nicht als Maßhilfslinien zweitgenutzt werden und sie sollen auch nicht Kanten von Werkstücken oder Aussparungen verlängern. Beides würde zwar Maßhilfslinien sparen, wirkt aber schnell verwirrend. Bei der Bemaßung gilt zudem generell, dass Maßhilfslinien sich nicht schneiden sollen und Maßlinien sich niemals schneiden dürfen (6).

Was erst mal einigermaßen pedantisch klingt, hat allerdings durchaus seinen Sinn, denn es dient der Übersichtlichkeit. In deren Sinn sollte man zudem möglichst keine doppelten Bemaßungen anbringen. Außerdem sollte man – wenn möglich – alle Maße außerhalb des Werkstückes platzieren, damit dessen Form möglichst klar auf dem Plan zu erkennen ist.

Wenn der Abstand der Maßhilfslinien so klein ist, dass man nur noch Maßpfeile und keine Maßlinie mehr sehen würde, setzt man die Maßpfeile nach außen, wie bei den 5mm-Maßen auf den Bildern (2) bis (5) zu sehen.

CAD statt Stift

Die Maßelemente, die man beim klassischen technischen Zeichnen mit Lineal und Bleistift noch alle einzeln zeichnen musste, sind in *FreeCAD* natürlich schon fertig vorgegeben. Man muss die Bemaßung nur noch richtig positionieren. Dabei wählt man zwei Bezugspunkte aus und klickt den entsprechenden Button zur Bemaßung an. Die Maßzahl berechnet FreeCAD dann aus dem jeweiligen Maß im dreidimensionalen Werkstück automatisch. Man muss sie daher nicht mehr nachmessen und sie bleibt auch bei Veränderungen der Geometrie des Werkstücks stets aktuell.

Sobald man in FreeCAD eine Bemaßung erzeugt, kann man durch Klick auf die Maßzahl die Maßlinie mit der Maus verschieben und gleichzeitig die Maßzahl entlang der Maßlinie auf die gewünschte Position rücken. Die Eigenschaften der Bemaßung lässt sich ändern, indem man auf die Maßzahl doppelklickt. So kann man beispielsweise die Pfeilrichtung ändern, also die Pfeile von innen nach außen setzen, falls sonst keine Maßlinie mehr sichtbar ist. Auch um die Abstände der Maßlinien oder die genaue Ausrichtung der Maßzahl braucht man sich keine Gedanken zu machen: Der Reihe nach markiert lassen sich die Maßzahlen über die Buttons *Horizontale Maße anordnen* oder *Vertikale Maße anordnen* automatisch ausrichten. Schritt für Schritt zeigen wir das alles im begleitenden Video zu diesem Artikel.

Die Maßelemente sind in FreeCAD in der Leserichtung immer richtig ausgerichtet: Ein horizontales Maß ist von unten lesbar und ein vertikales Maß von rechts. Wenn man jedoch eine vertikale Ausrichtung mit einem Längenmaß belegt, wofür es ein eigenes Werkzeug gibt, erscheint die Maßzahl verkehrt herum. Daher sollte man immer das passende Maß-Symbol aus der Werkzeugleiste wählen – auch dazu mehr im Video.

Eine Reihe weiterer Buttons in FreeCAD ergänzen bereits vorhandene Bemaßungen. Beispielsweise lässt sich ein Durchmesserzeichen zu einer Maßzahl hinzufügen oder eine Kreismittellinie zu einer Bohrung. Auch Hilfslinien für Innengewinde gehören dazu, da man Gewinde damit kenntlich macht. Man kann auch ganze Maßketten auf einmal erstellen, wenn man Bemaßungspunkte nacheinander markiert und dann auf den entsprechenden Button klickt. Auch solche Maßketten kann man wieder automatisch anordnen lassen. Mit Maßketten gibt man das Prinzip der gemeinsamen Bezugskante allerdings auf und handelt sich eventuell beim Nachbau in der Werkstatt Ungenauigkeiten ein.

Bevor Sie sich jetzt aber zu sehr in den Finessen der Bemaßung verlieren, denken Sie sich lieber ein hübsches kleines Werkstück für Ihre erste eigene technische Zeichnung in FreeCAD aus – und dann ausdrucken und ab in die Werkstatt zum Fertigen nach Plan! — pek

Hier kommt Verstärkung



Das **Make-Sonderheft** bietet einen praxisorientierten Einstieg in Schaltungen mit Operationsverstärkern inkl. Experimentierset. Will man Sensorsignale verarbeiten oder verstärken, Spannungen überwachen oder Audiosignale filtern: Mit geringem Aufwand und ohne komplizierte Berechnungen setzt man Operationsverstärker ein. Das Heft erklärt, wie alle Schaltungen funktionieren.

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

Heft + Experimentierset für nur 49,95 €

🚽 shop.heise.de/make-opv



Raspberry Pi Debug Probe

Debugging- und USB-Seriell-Adapter für Picos



Debugging-Adapter dienen dazu, Fehler in Mikrocontroller-Software zu finden und zu analysieren. Die *Raspberry Pi Foundation* stellt mit dem *Raspberry Pi Debug Probe* einen eigenen Debugging-Adapter vor, mit dem man beispielsweise Code für den *Raspberry Pi Pico* mit dem Mikrocontroller RP2040 untersuchen kann.

Der Debug Probe dient dabei als Adapter zwischen USB und dem Pico-Anschluss *Serial Wire Debug* (SWD). Der Debug Probe kann aber auch als universeller USB-Seriell-Adapter arbeiten und ersetzt ähnliche Adapter mit Chips von *FTDI* (FT232) oder *Nanjing Qinheng* (CH340).

Der Mikrocontroller Raspberry Pi RP2040 des Pico hat zwei ARM-Kerne vom Typ Cortex-M0+, die jeweils einen Access Port (AP) für das Debuggingsystem ARM CoreSight haben. Um es per USB von einem anderen Rechner aus anzusprechen, dienen das CMSIS-DAP-Protokoll sowie eine Cortex-M-Fimware. Unter Linux gibt es das Debugging-Tool OpenOCD, das die Raspi-Macher auch für den RP2040 empfehlen.

Das SWD-Interface lässt sich zudem nutzen, um Firmware vom PC auf den Pico zu schreiben, wie die Raspberry Pi Foundation in der ausführlichen Online-Funktionsbeschreibung erklärt (siehe Link).

Der Raspberry Pi Debug Probe wird zusammen mit einem kleinen Gehäuse, einem USB-Kabel sowie zwei Kabeln für SWD- und UART-Kontakte geliefert. —ciw@ct.de

make-magazin.de/x2pu

Hersteller	Raspberry Pi Ltd
URL	make-magazin.de/x2pu
Preis	12,20 €

Raspberry Pi Global Shutter Cam

Raspi-Kamera für Spezialanwendungen

Mit der *Raspberry Pi Global Shutter Camera* erweitert die *Raspberry Pi Foundation* ihr bereits recht großes Angebot an Kamera-

modulen zum direkten Anschluss an Raspis. Die Besonderheit der **Global Shutter Camera** (GS Cam) ist der elektronische Verschluss des verwendeten Sony-Sensors IMX296LQR-C. Er arbeitet nicht zeilenweise (Rolling Shutter), sondern wirkt auf alle 1,58 Millionen Pixel gleichzeitig. Dadurch vermeidet die GS Cam jene Artefakte bei der Erfassung sich bewegender Objekte, die als Rolling-Shutter-Effekt bekannt sind.

Rolling-Shutter-Artefakte lassen sich in kontrollierter Umgebung mt künstlicher Beleuchtung auch vermeiden, indem man statt einer Dauerbeleuchtung ein Blitzlicht oder ein Stroboskop verwendet. Die Global Shutter Camera liefert aber ohne Hilfsmittel korrekte Bilder. Bei heller Beleuchtung soll sie mit einer minimalen Verschlusszeit von 30 Mikrosekunden arbeiten können. Die Raspberry Pi Global Shutter Camera hat wie die 2020 vorgestellte *Raspberry Pi HQ Ca*mera mit dem 12-Megapixel-Bildsensor Sony IMX477 einen Anschluss für C-/CS-Mount-Objektive. Je nach Anwendungsbereich lässt sich also ein optimal passendes Objektiv einsetzen, muss allerdings auch separat gekauft werden.

> Der Sony-Sensor IMX296-LQR-C der GS Cam hat nur eine relativ geringe Auflösung von 1456 × 1088 quadratischen RGB-Bildpunkten. Letztere haben aber jeweils 3,45 Mikrometer Kantenlänge und fangen mehr Licht ein als die kleineren Pixel höher auflösender Sensoren.

Einige Beispiele für Aufnahmen mit der Global Shutter Camera zeigt ein Blog-Beitrag von Eben Upton, CEO von Raspberry Pi Ltd (siehe Link). —*ciw@ct.de*

make-magazin.de/x2pu

Hersteller	Raspberry Pi Ltd
URL	make-magazin.de/x2pu
Preis	ca. 59 €

ePulse Feather

Sparsames Mikrocontroller-Board mit ESP32

Gegenüber anderen ESP32-Boards genehmigt sich das neue Mikrocontroller-Board der Schweizer Firma *Thingpulse* nur ein Zehntel der Ruhestromaufnahme: Mit nur 12 µA ist der *ePulse Feather* besonders für IoT-Geräte im mobilen Einsatz prädestiniert. Pinbelegung und Abmessungen entsprechen dem *Feather*-Formfaktor von Adafruit, wodurch eine Vielzahl von Erweiterungsplatinen (Wings) kompatibel sind.

Das Modul nutzt den relativ neuen ESP32-DOWDR2-V3-SoC von Espressif (zwei Tensilica Xtensa LX6-Kerne mit 240 MHz, 512 KByte RAM, 8Mbyte Flash-ROM). Es akzeptiert Versorgungsspannungen von 3,3V bis 6V und kann daher problemlos direkt aus einer einzelnen Lilon-Zelle versorgt werden. Für die Anbindung von Peripherie stehen 20 I/O-Pins zur Verfügung. —*cm*

Hersteller	Thingpulse
URL	make-magazin.de/x2pu
Preis	14,95 US-\$



© Copyright by Maker Media GmbH.

EsPiFF

ESP32-Board im Raspberry-Pi-Format

Zum Messen, Steuern und Regeln kann man einen Raspberry Pi verwenden – muss man aber nicht. Besonders dann nicht, wenn Grafikbildschirm und Linux als Betriebssystem gar nicht benötigt werden. Ein ESP32 reicht in diesen Fällen meist auch. Mit dem EsPiFF gibt es den demnächst sogar im Raspberry-Format, mit dem sich, so sein Entwickler, die meisten der Raspi-Aufsteckplatinen (Hats) verwenden lassen sollen.

Der zurzeit schwierig zu beschaffende Einplatinen-Computer ist bei vielen Aufgaben schlicht unterfordert oder auch aus anderen Gründen nicht wirklich die beste Wahl. So brauchen Raspis recht viel Strom, produzieren viel Wärme und müssen nach dem Einschalten erst booten, sodass sie ihre Arbeit nicht sofort aufnehmen. Der permanente Betrieb mit der SD-Karte gilt auch nicht als sonderlich stabil.

Der EsPiFF, entwickelt von der MDC-Service Wittenburg GmbH in Freiberg, ist auf den ersten Blick kaum vom Vorbild Raspi zu unterscheiden: Die Platine des Open-Source-Projekts hat genau die gleiche Größe, die 40polige Stiftleiste sitzt an der gleichen Stelle. Seitlich sind die POE-taugliche Ethernet- und eine USB-A-Buchse angeordnet. An der Längsseite sitzt eine USB-C-Buchse für die Programmierung und zur Stromversorgung (5V, 3A). Die hohe Stromaufnahme tritt aber nur auf, wenn entsprechend energiehungrige Hats aufgesteckt sind. Mit welchen im Einzelnen die Platine laufen soll, darüber gibt es noch keine näheren Infos.

Auf dem Board arbeitet ein *ESP32-Wrover*-Modul mit 8MB PSRAM und 16MB

Flash-Speicher. Für WLAN- und Bluetooth-Betrieb ist eine externe Antenne anzuschließen (wird mitgeliefert). Als Coprozessor gibt es einen RP2040, der sich vor allem um die Stiftleiste kümmert. Zusätzlich gibt es noch zwei Anschlüsse für UART-Schnittstellen und eine Supercap-gepufferte RTC.

Nach einer erfolgreichen Crowdfundig-Kampagne bei *Crowd Supply* kann das Board dort für 65 US-Dollar vorbestellt werden. Dazu



3ild: MDC-Service Wittenberg GmbH

kommen aber noch Frachtkosten und Mehrwertsteuer. Der Preis bei Bestellung aus Deutschland soll am Ende bei knapp 100 Euro liegen. —hgb

Hersteller	MDC-Service Wittenberg GmbH
URL	www.crowdsupply.com/mdc/espiff
Preis	65 US-\$ ohne Steuer, ca. 100 € mit Steue
	und Versand

Ausprobiert — von Make:—

Flatterer

G-Code für Lasercutter und Fräsen aus 3D-Daten erzeugen

Das Open-Source-3D-Paket *Blender* ist zwar kein parametrisches CAD-Programm, aber durchaus brauchbar, um damit technische Modelle und 3D-Druckvorlagen zu konstruieren. Bisher war es aber schwierig, seine 3D-Entwürfe etwa für den Lasercutter oder fürs 2,5D-Fräsen aufzubereiten. Dieses Problem hat der Blender-Entwickler und Maker Sybren Stüvel mit einem kostenlosen Add-On gelöst.

Flat mesh to SVG exporter for laser cutters				
3D Viewport > side-p	anel > Export			
CiUsers/Carsten War	manniAppD	atal.n3.2iscript	sladdonsli	latterer_initp
Sybren A. Stüvel				
Documentation			•	
ssumes that 1 Blender	init is 1 mm.			
gs:				
gs: Laser Beam Width				
gs: Laser Beam Width Material Width		0.16 m 300 m		
gs: Laser Beam Width Material Width Material Length		0.16 m 300 m 300 m		
gs: Laser Beam Width Material Width Material Length Material Thickness		0.16 m 300 m 300 m 3 mm		
gsi Laser Beam Width Material Width Material Length Material Thickness Margin		0.16 m 300 m 300 m 300 m 3 mn 5 mn		
gsi Laser Beam Width Material Width Material Length Material Thickness Margin Shape Padding		0.16 m 300 m 300 m 300 m 3 mm 5 mm 1 mm		
gs: Laser Beam Width Material Width Material Thickness Margin Shape Padding Pack Sorting	Descending	0.16 m 300 m 300 m 3 mm 3 mm 5 mm 1 mm 1 mm		
	3D Viewport > side-ps C:Users/Cansten Wart Sybren A. Stüvel 14.0 Documentation ssumes that 1 Blender v	3D Vewport > side-panel > Expor C-Ubers/Carsten WartmannAppD Sybren A. Stilvel 1-4.0 Documentation commentation commentation commentation	30 Viewport > side-panel > Export tab C:Ubers/Lianten WartmannAppDateln3.2tscript Sytren A: Silver 14.0 Documentation The Report a Bug ssumes that 1 Blender unt is 1 mm.	3D Vewport > side panel > Export tab CNDerryCanten WartmannAppDataL_A3 Ziscriptshaddons/ Sybren A. Slövet 14.0 Documentation @ Report a Bug sumes that 1 Biender unit is 1 mm.

Das *Flatterer*-Add-On ist inzwischen in der Version 1.4 verfügbar und ist hauptsächlich für die Bearbeitung von Gehäusen gedacht. Es lassen sich aber auch andere Teile oder Schablonen (z.B. für Handfräsen mit Kopierring) bearbeiten.

Das Add-On packt die

Linienzüge dabei möglichst platzsparend in den Arbeitsraum des Lasercutters oder der Fräse, erzeugt Schnitt- und Gravierlinienzüge, berechnet die Schnittbreite des Lasers (*Kerf*-Kompensation), hilft beim Einstellen der Szene in Blender auf Millimeter und kann *Solidify*-Modifier für das 3D-Modell hinzufügen.

Die Grenzen liegen dort, wo Objekte in Blender nicht für die 2D-CNC-Fertigung geeignet sind – ein automatischer "Slicer", wie ihn *Blender-CAM* bietet, ist nicht im Funktionsumfang enthalten. Man muss hier also



schon bei der Konstruktion an die gewünschte Fertigungstechnik denken und "flach" konstruieren, also in einer Hauptebene. Für den 3D-Effekt braucht es dann den *Solidify*-Modifier. Das hat aber auch den Vorteil, dass man auf diese Weise schnell merkt, ob das Objekt so überhaupt auf einem Lasercutter oder einer Fräse gefertigt werden kann. —*caw*

Entwickler	Sybren Stüvel
URL	gitlab.com/dr.sybren/flatterer
Preis	gratis

Loop Energy Akkus mit 1,5Volt



Die Maße von Lithium-Ionen-Akkuzellen passten bislang meist nicht in die Halterungen batteriebetriebener Geräte. Und spätestens bei der Ausgangsspannung von 3,7V war mit dem problemlosen Ersatz für herkömmliche Trockenbatterien ohnehin Schluss.

Die Loop-Energy-Akkus von Verico ändern das: Es gibt sie im Größenformat von AA- und AAA-Trockenzellen. Somit passen sie ohne weiteres in die Batteriefächer vieler Geräte. Eine in den Zellen integrierte Elektronik sorgt zudem während des gesamten Entladevorgangs für eine konstante Ausgangsspannung von 1,5V.

Die Triple-A-Zellen haben eine Kapazität von 900mWh, die AA-Typen sogar 2550mWh. Die Ladezeit beträgt ein bis zwei Stunden. Dazu besitzt jede Zelle einen USB-C-Anschluss, der zur Energiespende direkt mit einem entsprechenden Netzteil (nicht im Preis enthalten) verbunden werden kann.

Geliefert werden die Zellen im Set mit einem Vierfach-Ladekabel-Adapter, sodass ein Ladegerät mit nur einer Ausgangsbuchse genügt, um vier Zellen zu laden. Die Akkuzellen müssen zum Laden auch nicht denselben Entladestatus haben, da die Elektronik in den Zellen den Ladestrom individuell regelt. —hgb

eadapter)

45€

Hersteller	Verico
Vertrieb	z.B. Pollin
Preis	AA-Set (4 Zellen, Lac
	15,95 €, AAA-Set 15

Flsun V400

Schneller FDM-Delta-3D-Drucker

Der Flsun V400 ist die verbesserte Nachfolgeversion des Flsun Superracer, der bereits eine Druckgeschwindigkeit von 180 mm/s erreichte – der V400 arbeitet hingegen mit maximal 400 mm/s und ist damit der mit Abstand schnellste Drucker, den die Kollegen von TechStage bisher in ihrer Redaktion getestet haben. Das Druckbett hat einen Durchmesser von 30cm, die maximale Druckhöhe beträgt 41cm. Das Ganzmetall-Hotend erreicht 300° C. In Verbindung

mit der PEI-Druckplatte und zusätzlicher Einhausung können so auch Hochtemperatur-Filamente wie Nylon und PC verarbeitet werden.

Der Delta-3D-Drucker lässt sich in knapp einer Stunde aufbauen; die Antriebsmotorik ist in Streben und Druckkopf schon vorab fertig eingebaut und einsatzbereit. So bleibt dem Käufer nur noch, einzelne Teile zu verschrauben und von Steckverbindungen zu setzen. Den Aufbau fand man bei *TechStage* überraschend leicht – das Komplizierteste war das Einfädeln des Kabelbaums durch eine Strebe. Im Inneren des Druckers arbeitet die *Klipper*-Firmware, das Gerät ist mit der dafür nötigen Hardware (Mini-Computer mit Quad-Core-Chip und 1Gbyte RAM) ausgestattet. Bedient wird der V400 über einen 7-Zoll-Touchscreen und er verfügt über zahlreiche Schnittstellen wie drei USB-Ports, microSD-Karten-Slot und WLAN.

Im Test zeigte sich, dass 400 mm/s Druckgeschwindigkeit zwar durchaus drin sind, dabei aber die Druckqualität leidet, weshalb



die Standard-Einstellungen bei der äußeren Kontur, Oberund Unterseite auf 120 mm/s gedrosselt sind. Dennoch: Ein *Benchy* in guter Qualität ist nach den Erfahrungen von *TechStage* in einer guten halben Stunde fertig. —*pek*

Einen ausführlichen Testbericht sowie einen aktuellen Preisvergleich gibt es online:

make-magazin.de/x2pu

Hersteller F Bezugsquelle C Preis c

Flsun Online-Handel ca. 950 €

Metro M7

500-MHz-ARM-Board im Arduino-Format

Für besonders leistungshungrige Echtzeit-Anwendungen verspricht Adafruit mit dem neuen Board *Metro M7* einen erheblichen Geschwindigkeitsschub. Als Prozessor kommt hier ein *NXP iMX RT1011* mit 500 MHz zum Einsatz, der intern 128 KByte SRAM bietet und über QSPI auf 8MByte externen Flash-Speicher (Winbond 25Q64FV) zugreifen kann. Als Wireless-Coprozessor ist ihm ein ESP32 beigestellt, bei Adafruit als *AirLift* bezeichnet. Das Board ist vom Formfaktor her im *Arduino*-Design gehalten und nimmt zum *Uno* kompatible Shields auf.

Seine Programme kann man über eine USB-C-Schnittstelle oder das bei ARM-Controllern gängige SWD-Interface hochladen und debuggen; zum Einsprung in den stets verfügbaren Bootloader genügt ein "Doppelklick" auf den Reset-Taster. Ein Anschluss für Adafruits *STEMMA*-I2C-Peripherie ist ebenfalls vorhanden. Die Stromversorgung kann auch



über die vom Arduino Uno bekannte Koaxial-Steckverbindung erfolgen. Adafruit hat für das Metro M7 bereits ein *CircuitPython*-Paket freigegeben. Das Board war schon vor zwei Jahren angekündigt, hat sich aber wegen Lieferproblemen bei den Halbleitern verspätet. —*cm*

Hersteller	Adafruit
URL	www.adafruit.com/product/4950
Preis	29,95 US-\$

Arduino GIGA R1 WiFi

Extrastarkes Mikrocontroller-Board mit vielen Anschlüssen

Das Arduino GIGA R1 WiFi kostet zwar deutlich mehr als der billigste Raspberry Pi 4 (falls man einen ergattert). Doch vor allem dank des Mikrocontrollers STM32H747XI des europäischen Herstellers STMicroelectronics eignet sich das Board für anspruchsvolle Projekte. Denn es stellt vergleichsweise hohe Rechenleistung sowie eine Fülle von Anschlüssen bereit. Dazu gehören außer 76 GPIO-Pins auch WLAN, Bluetooth, ein USB-2.0-Host und eine Audio-Klinkenbuchse.

Der STM32H747XI enthält zwei 32-Bit-ARM-Rechenkerne, nämlich einen Cortex-M7 mit bis zu 480 MHz Taktfrequenz sowie einen Cortex-M4 mit 240 MHz. Die beiden Kerne teilen sich den Zugriff auf die I/O-Schnittstellen sowie 1MByte RAM und 2MByte Flash-Speicher. Jeder Kern kann aber eigene Software ausführen; laut Arduino-Blog etwa einen Arduino-Sketch und ein MicroPython-Programm parallel.

Die USB-Hostbuchse lässt sich nutzen, um Massenspeicher (einen USB-Stick) oder Eingabegeräte anzuschließen. Das WLAN-Modul stammt von *Murata* (1DX) und funkt im 2,4-GHz-Band. Es enthält auch einen Bluetooth-5.1-Teil. Eine externe Antenne liegt bei.



Das Arduino GIGA R1 WiFi hat eine USB-C-Buchse zur Stromversorgung und zum Programmieren. Alternativ lässt es sich mit 6 bis 24 Volt Gleichspannung speisen. Der STM32H747XI enthält auch eine Echtzeituhr (RTC) die sich über die VRTC-Pins puffern lässt. Das Board lässt sich per WLAN auch in die Arduino-Cloud einbinden, etwa um Sensordaten zu verarbeiten oder Steuerbefehle entgegenzunehmen. —*ciw@ct.de*

Hersteller	Arduino.cc
URL	store.arduino.cc/products/giga-r1-wifi
Preis	ca. 69 €

Creality CR-M4

FDM-3D-Drucker mit großem Bauraum

Stolze 450mm × 450mm × 470mm (Länge × Breite × Höhe) misst der Bauraum des neuen Modells *CR-4M* und ist damit vom Volumen her fast achtmal so groß wie der der bekannten Amateur-Geräte der *Ender*-Serie. Der Hersteller sieht das Gerät auch eher als Arbeitspferd ("workhorse") für die Kleinserienproduktion denn als Hobby-Printer.

Für den professionellen Einsatz spricht auch die massive Bauweise: Stolze 22,5kg Gewicht bringt der Drucker aufgrund des sehr stabil wirkenden Metallrahmens auf die Waage. Auch die statt der üblichen Wellen-/Gleitlagerkombination verwendeten Linearführungen für den Drucktisch machen Hoffnung auf exakte Mechanik und lange Lebensdauer. Die Motoren sollen ebenfalls verstärkt sein und laut *Creality* selbst 30 Tage ununterbrochenen Drucks ohne Überhitzungsprobleme durchhalten.

Die Stromversorgung erfolgt über ein Netzteil mit 800 Watt Leistung, was auf entsprechend hohe Heizleistung schließen lässt. Das Druckbett kann auf bis zu 100°C und das Hotend auf bis zu 300°C aufgeheizt werden. Somit sind auch Druckfilamente aus Nylon (Polyamid) oder ähnlichem Material mit hohem Schmelzpunkt verarbeitbar.

Der Direktantrieb-Extruder verarbeitet mit seinem Doppelgetriebe 1,75 mm-Filamente und ist auch für elastische Kunststoffe geeignet. Die Druckgeschwindigkeit wird mit 120 mm/s bei 0,05mm bis 0,35 mm Layerhöhe angegeben. Eine Filament- und Stromausfallüberwachung ist mit an Bord. Gesteuert wird das Gerät über einen 4,3-Zoll-Touchscreen, das auf neun Sprachen eingestellt werden kann.

Druckdaten erwartet der Printer über USB-C-Kabel, Speicherstick, LAN oder via Cloud und App des Herstellers. Damit soll auch ein komfortabler Betrieb einer Druckerfarm möglich sein: Einmal den Druckjob abgeschickt, soll er dann auf allen in der Farm verfügbaren CR-4M gedruckt werden – ohne weitere Bedienung. Für die Anfertigung



von Kleinserien könnte das sehr interessant werden. Auch der Preis ist für ein solch stabil gebautes Gerät eher günstig. Der Drucker kann derzeit im Online-Shop des Herstellers vorbestellt werden. —hgb

Hersteller	Creality
URL	make-magazin.de/x2pu
Preis	1359 € (Stand 23.03.23)

IMPRESSUM

Redaktion

Make: Magazin Postfach 61 04 07, 30604 Hannover Karl-Wiechert-Allee 10, 30625 Hannovei Telefon: 05 11/53 52-300 Telefax: 05 11/53 52-417 Internet: www.make-magazin.de

Leserbriefe und Fragen zum Heft: info@make-magazin.de

Die E-Mail-Adressen der Redakteure haben die Form xx@make-magazin.de oder xxx@make-magazin.de. Setzen Sie statt "xx" oder "xxx" bitte das Redakteurs-Kürzel ein. Die Kürzel finden Sie am Ende der Artikel und hier im Impressum.

Chefredakteur: Daniel Bachfeld (dab) (verantwortlich für den Textteil)

Stellv. Chefredakteur: Peter König (pek)

Redaktion: Heinz Behling (hgb), Johannes Börnsen (jom), Ákos Fodor (akf), Carsten Meyer (cm), Carsten Wartmann (caw)

Mitarbeiter dieser Ausgabe: Beetlebum (Comic), Thomas Euler, Jens Hackel, Michael Linsenmeier, Ulf Mayer, Matthias Mett, Leonard Ure, Dirk Wahl, Christof Windeck, Gustav Wostrack

Assistenz: Susanne Cölle (suc), Christopher Tränkmann (cht), Martin Triadan (mat)

Leiterin Produktion: Tine Kreve

DTP-Produktion: Martina Bruns, Martin Kreft (Korrektorat)

Art Direction: Martina Bruns (Junior Art Director)

Layout-Konzept: Martina Bruns

Layout: Nicole Wesche

Fotografie und Titelbild: Andreas Wodrich

Digitale Produktion: Kevin Harte, Thomas Kaltschmidt, Pascal Wissner

Hergestellt und produziert mit Xpublisher: www.xpublisher.com

Verlag

Maker Media GmbH Postfach 61 04 07, 30604 Hannover Karl-Wiechert-Allee 10, 30625 Hannover Telefon: 05 11/53 52-0 Telefax: 05 11/53 52-129 Internet: www.make-magazin.de

Herausgeber: Christian Heise, Ansgar Heise

Geschäftsführung: Ansgar Heise, Beate Gerold

Anzeigenleitung: Michael Hanke (-167) (verantwortlich für den Anzeigenteil), mediadaten.heise.de/produkte/print/ das-magazin-fuer-innovation

Leiter Vertrieb und Marketing: André Lux (-299)

Service Sonderdrucke: Julia Conrades (-156)

Druck: Dierichs Druck + Media GmbH & Co.KG. Frankfurter Str. 168, 34121 Kassel

Vertrieb Einzelverkauf:

DMV DER MEDIENVERTRIEB GmbH & Co. KG Meßberg 1 20086 Hamburg Telefon: +49 (0)40 3019 1800 Telefax: +49 (0)40 3019 1815 E-Mail: info@dermedienvertrieb.de Internet: dermedienvertrieb.de

Einzelpreis: 13,50 €; Österreich 14,90 €; Schweiz 26.50 CHF; Benelux 15,90 €

Abonnement-Preise: Das Jahresabo (7 Ausgaben) kostet inkl. Versandkosten: Inland 80,50 €; Österreich 88,90 €; Schweiz 123.90 CHF; Europa: 95,20 €; restl. Ausland 100,80 €

Das Make-Plus-Abonnement (inkl. Zugriff auf die App, Heise Magazine sowie das Make-Artikel-Archiv) kostet pro Jahr 6,30 € Aufpreis.



Abo-Service:

Bestellungen, Adressänderungen, Lieferprobleme usw.: Maker Media GmbH Leserservice Postfach 24 69 49014 Osnabrück E-Mail: leserservice@make-magazin.de Telefon: 0541/80009-125 Telefax: 0541/80009-122

Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Heraus-geber 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 2023 and published under license from Make Community LLC. All rights reserved.

Printed in Germany. Alle Rechte vorbehalten. Gedruckt auf Recyclingpapier

© Copyright 2023 by Maker Media GmbH

ISSN 2364-2548



Wir haben ein Internet-Radio im Heft: Welchen Podcast oder Radiosender hörst du damit?

Meike Weiß

Hannover, macht beim Heise-Marketing Make-Leser zu treuen Make-Abonnenten Wenn ich Auto fahre oder koche, schalte ich gerne mal das Radio an und summe fleißig mit. Am liebsten höre ich NDR 2

Niedersachsen oder

11 ive.

Leonard Ure

radio, um den Podcast Väter Freuden zu hören. Als junger Vater hilft es mir die Höhen und Tiefen des Elternseins zu verarbeiten. Außerdem die Presseschau und DLF.

Dunia Selman

Hannover, seit November Social Media Managerin bei der Maker Media

Ich höre zwar kein Radio, aber ich würde Energy Bremen einschalten, um Neuigkeiten aus der Heimat zu hören. Als Podcast würde ich einen Schlafmeditations-Podcast hören, den ich abends nebenherlaufen lasse.

Johannes Börnsen Ritterhude, seit Dezember Host und Produzent unserer Youtube-Videos In der Werkstatt höre

ich Podcasts, z.B. Making It mit Jimmy Diresta oder The Woodworking Talkshow. Wenn ich mich konzentrieren muss, läuft Chillhop Radio oder *Ibiza Global* Radio, repetitive Musik hilft mir beim Denken!

Inserentenverzeichnis

dpunkt.verlag GmbH, Heidelberg	. 9
OXON AG, Liebefeld	. 23
Technische Universität Ilmenau, Ilmenau	69

TUXEDO Computers GmbH, Augsburg	116
Westfalenhallen Unternehmensgruppe GmbH, Dortmund	31
Make:markt	49

Hannover, organisiert seit November als Event- und Community-Manager die Maker Faire Hannover mit

Ich nutze das Internet-

Ein Veranstaltungsformat der Make:

0

Hannover **Maker Faire**[®]

Das Format für Innovation und Macherkultur

Sichere dir noch bis deine Tickets Sichere dir noch bis deine Tickets Fritt noch bis deine Tickets Bird ter Bird Tant.

Früh Kaufen

19.–20. Aug. Hannover Congress Centrum

Generative.de







Linux Installation

Mit unserer WebFAI richten wir Ihre Linux-Installation vollautomatisiert und exakt so vollständig funktionsfähig ein, wie dies in unserer Fertigung bei TUXEDO Computers vorgenommen wird!

Linux Distribution

Zur Auswahl stehen alle Linux Distributionen, die wir auch in unserem Onlineshop-Konfigurator zur Auswahl anbieten. Je nach Distribution haben Sie die Wahl zwischen verschiedenen Desktopoberflächen.

immer aktuell

Durch unsere WebFAI erhalten Sie zudem von unseren Servern immer unsere aktuellsten von uns getesteten Linux Betriebssystemsversionen.

TUXEDO WebFAI Fully Automated Installation



Jahre Garantie







Datenschutz



vor Ort



🛱 tuxedocomputers.com

© Copyright by Maker Media GmbH.