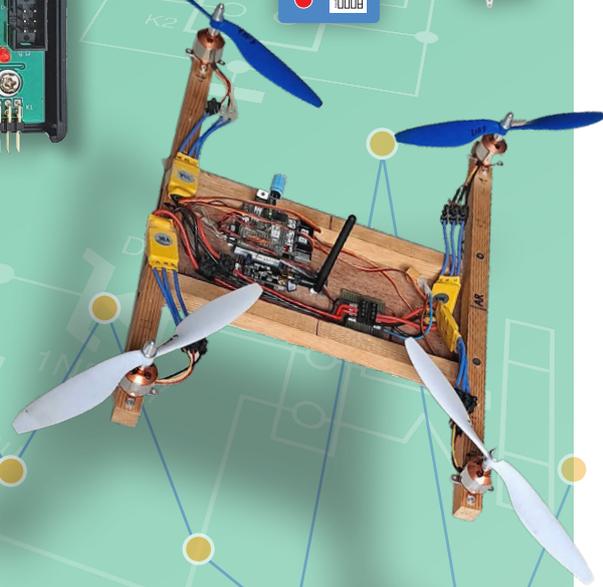
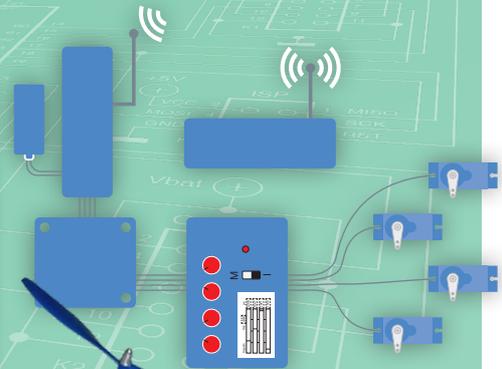
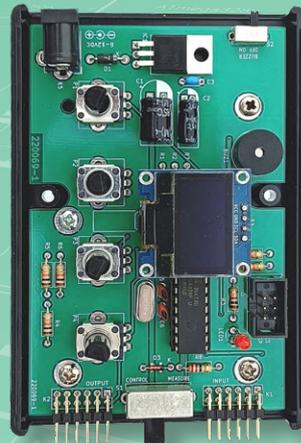


Testet bis zu vier Servos
einzeln oder eingebaut

SUPER servo- Tester



IM FOKUS Prüf- und Messtechnik

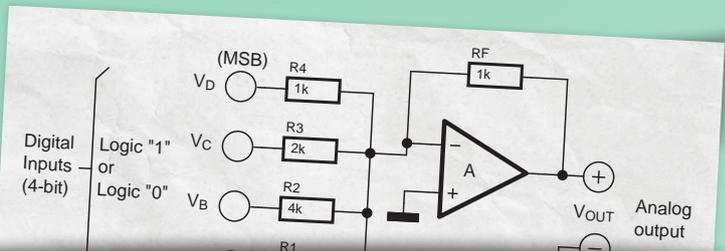
Automatisierung der Prüf- und Messtechnik

Tipps für die Programmierung
von Prüfgeräten



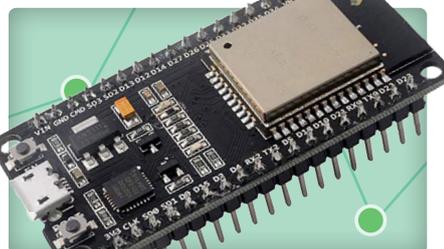
Analoge Signale und der Mikrocontroller

ADCs, DACs, Strommessung und mehr



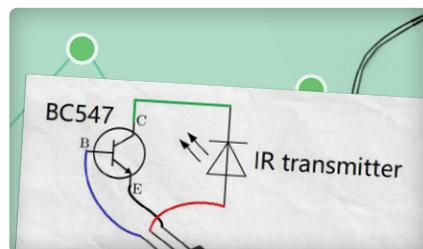
Energielogger
Energiebedarf messen
und aufzeichnen

S. 74



Hier Android-Smartphone,
dort ESP32?
Praxisprojekt mit der
Android-WiFi-API

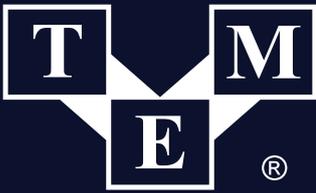
S. 26



IR-Fernsteuerung mit
Smartphone und ESP32
Anpassungsfähig und universell

S. 48



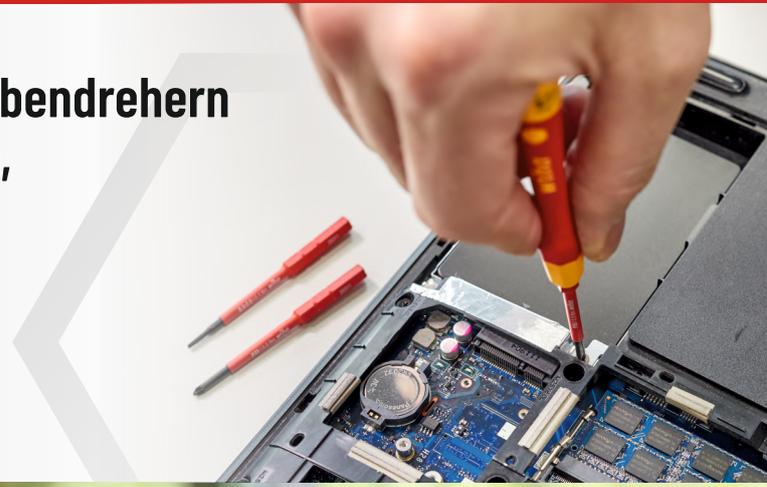


Electronic Components

TRANSFER MULTISORT ELEKTRONIK

■ GLOBAL DISTRIBUTOR OF ELECTRONIC COMPONENTS

Entdecken Sie das Angebot an Schraubendrehern
ElectricVario und überzeugen Sie sich,
dass die Qualität zählt



Mehr erfahren:



TME Germany GmbH
Dohnanyistraße 28-30, 04103 Leipzig, tme@tme-germany.de

tme.eu

YOU NEED IT, WE HAVE IT!

Schließen Sie sich uns an:

■ ■ ■ ■ tme.com ■

54. Jahrgang, Nr. 593
 Mai/Juni 2023
 ISSN 0932-5468

Das Elektor Magazin wird 8 mal im Jahr
 herausgegeben von

Elektor Verlag GmbH

Lukasstraße 1, 52070 Aachen (Deutschland)
 Tel. +49 (0)241 95509190

www.elektor.de | www.elektormagazine.de

Für alle Ihre Fragen

service@elektor.de

Mitglied werden

www.elektormagazine.de/abo

Anzeigen

Büdra Kas

Tel. +49 (0)241 95509178

busra.kas@elektor.com

www.elektormagazine.de/mediadaten

Urheberrecht

© Elektor International Media b.v. 2023

Die in dieser Zeitschrift veröffentlichten Beiträge, insbesondere alle Aufsätze und Artikel sowie alle Entwürfe, Pläne, Zeichnungen einschließlich Platinen sind urheberrechtlich geschützt. Ihre auch teilweise Vervielfältigung und Verbreitung ist grundsätzlich nur mit vorheriger schriftlicher Zustimmung des Herausgebers gestattet. Die veröffentlichten Schaltungen können unter Patent- oder Gebrauchsmusterschutz stehen. Herstellen, Feilhalten, Inverkehrbringen und gewerblicher Gebrauch der Beiträge sind nur mit Zustimmung des Verlages und ggf. des Schutzrechtsinhabers zulässig. Nur der private Gebrauch ist frei. Bei den benutzten Warenbezeichnungen kann es sich um geschützte Warenzeichen handeln, die nur mit Zustimmung ihrer Inhaber warenzeichengemäß benutzt werden dürfen. Die geltenden gesetzlichen Bestimmungen hinsichtlich Bau, Erwerb und Betrieb von Sende- und Empfangseinrichtungen und der elektrischen Sicherheit sind unbedingt zu beachten. Eine Haftung des Herausgebers für die Richtigkeit und Brauchbarkeit der veröffentlichten Schaltungen und sonstigen Anordnungen sowie für die Richtigkeit des technischen Inhalts der veröffentlichten Aufsätze und sonstigen Beiträge ist ausgeschlossen.

Druck

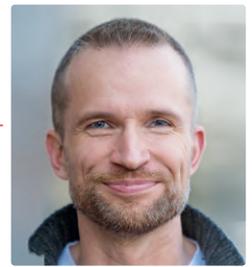
Senefelder Misset, Mercuriusstraat 35
 7006 RK Doetinchem (Niederlande)

Distribution

IPS Pressevertrieb GmbH, Carl-Zeiss-Straße 5
 53340 Meckenheim (Deutschland)
 Tel. +49 (0)2225 88010

Jens Nickel

Chefredakteur *ElektorMag*



Wissen für alle!

Das Programmieren ist eine Disziplin, die auch Einsteiger mit Erfolgserlebnissen belohnt. Bereits mit Grundkenntnissen einer Programmiersprache und ein paar guten Ideen lassen sich kleine Tools erstellen, die schwuppdiewupp ein paar Aufgaben erledigen, wo sonst mühevoller Handarbeit erforderlich wäre. Und seien wir ehrlich: Wer hat so etwas nicht schon stolz im Bekanntenkreis vorgeführt?

Das Programmieren einer kleinen Datenbank, das Aufsetzen eines Servers oder ähnlich Nerdiges wissen leider oft nicht alle Angehörigen zu würdigen. Der Coolness-Faktor steigt jedoch, wenn etwas gemessen, geregelt und gesteuert wird, was tatsächlich vorführbar ist und das Alltagsleben einfacher macht.

Gerade manche junge „Maker“ kommen dann jedoch schnell an ihre Grenzen. In die grundlegende Programmierung von Mikrocontrollern kann sich jeder schnell reinfuchsen, der schon ein paar Apps für den PC geschrieben hat - Sprachen wie Python sei Dank. Doch wie verbinde ich einen Mikrocontroller mit Sensoren, Aktoren und Anzeigeelementen, und was muss ich dabei beachten? Übrigens sind hier auch viele Studenten der Elektrotechnik nicht ganz sattelfest, wie uns Hochschullehrer immer wieder schreiben.

Bei Elektor haben wir uns zum Ziel gesetzt, auch solches Grundlagenwissen zu vermitteln - für Einsteiger, aber auch zur Auffrischung für alle, die schon Jahrzehnte den Lötcolben schwingen. Auf Seite 12 finden Sie einen Artikel von Mathias Claußen, der beschreibt, wie man mit Mikrocontrollern analoge Größen wie zum Beispiel Temperaturen erfassen kann, mit Sensoren, die diese Größen in Spannungen und Ströme umsetzen. Wie üblich kommt dabei auch die Praxis nicht zu kurz. Zu diesem Thema gibt es übrigens auch einige Perlen in unserem Archiv, zum Beispiel die Serie „Sensoren“ von Burkhard Kainka (Elektor ab 11/2016). Abonnenten können die PDFs wie immer kostenlos herunterladen; und natürlich sind die Artikel auch auf unserem USB-Stick-Archiv zu finden (www.elektor.de/20373).

Auch in Zeiten von steckbaren Fertigmodulen und automatischer Code-Generierung darf solides Grundlagenwissen nicht verloren gehen - sicher stimmen Sie mir hier zu. Falls Sie Wünsche oder Anregungen haben, schreiben Sie mir unter redaktion@elektor.de!



Veröffentlichen Sie bei Elektor!

Ihr Fachwissen über Elektronik ist willkommen: Schicken Sie uns Ihr Video, Ihren Artikelvorschlag oder Ihre Idee für ein Buch! Wir haben unseren Leitfaden für Autoren und Ersteller von Inhalten aktualisiert. Unter www.elektormagazine.com/submissions finden Sie alle Einzelheiten.

Unser Team

Chefredakteur: Jens Nickel (v.i.S.d.P.) | **Redaktion:** Asma Adhimi, Eric Bogers, Jan Buiting, Stuart Cording, Rolf Gerstendorf (RG, SG), Ton Giesberts, Hedwig Hennekens, Qiëlle Lont, Alina Neacsu, Dr. Thomas Scherer, Clemens Valens, Brian T. Williams | **Regelmäßige Mitarbeiter:** Roberto Armani, David Ashton, Tam Hanna, Priscilla Haring-Kuipers, Ilse Joostens, Prof. Dr. Martin Ossmann, Alfred Rosenkränzer | **Grafik & Layout:** Harmen Heida, Sylvia Sopamena, Patrick Wiolders | **Herausgeber:** Erik Jansen | **Technische Fragen:** redaktion@elektor.de

DEUTSCHE

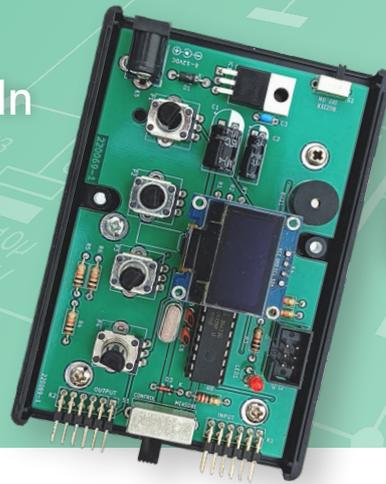
FACHPRESSE

Elektor ist Mitglied des 1929 gegründeten VDZ (Verband Deutscher Zeitschriftenverleger), der „die gemeinsamen Interessen von 500 deutschen Consumer- und B2B-Verlagen vertritt.“

Super Servo-Tester

Testet bis zu vier Servos einzeln oder eingebaut

6



Rubriken

3 Impressum

FOKUS

24 Von Entwicklern für Entwickler

Sub-Nyquist-Sampling in der Praxis

39 Aller Anfang...

Wir multivibrieren weiter!

42 Projekt 2.0

Korrekturen, Updates und Leserbriefe

72 Aus dem Leben gegriffen

Elektronik auf hohem Niveau

94 Das besondere Projekt

Es gibt noch viel zu tun!

110 Ethics in Action

Generative KI

114 Hexadoku

Sudoku für Elektroniker

44 Das neue I²C-Protokoll

Würdiger I²C-Nachfolger oder viel Rauch um Nichts?

52 Mikrocontroller-Dokumentationen verstehen

Teil 2: Register und Blockschaltbilder

79 Aufgebaut: Das 4tronix M.A.R.S. Rover Kit

92 Hinter den Kulissen von High-End-Audio im Eigenbau

Elektor-Ingenieur Ton Giesberts im Interview über die hohe Kunst des Analogdesigns

106 Roboterarm-Bausatz

Mit Raspberry Pi Pico und MicroPython

Industry

FOKUS

56 Automatisierung der Prüf- und Messtechnik

Programmieren von Prüfgeräten, damit sie tun, was Sie wollen

FOKUS

60 Infografik

Tools zum Testen und Messen

62 Überspannungsschutz für sicheren Betrieb

Transientenschutz für nicht-isolierte DC/DC-Leistungsmodule

FOKUS

66 Mess- und Prüfgeräte von Wiha

Installationstester und Messgeräte, auf die Sie sich verlassen können

FOKUS

68 Automatisierung von Tests und Zusammenarbeit bei Testergebnissen

Hintergrund

FOKUS

12 Analoge Signale und der Mikrocontroller

ADCs, DACs, Strommessung und mehr

20 Embedded World 2023

Interessante Technik auf der Messe

26 Hier Android-Smartphone, dort ESP32?

Praxisprojekt mit der Android-WiFi-API

Automatisierung der Prüf- und Messtechnik

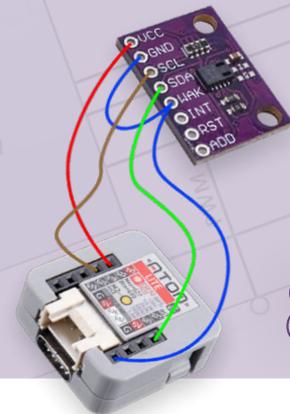
Programmieren von Prüfgeräten



56

eCO₂-Telegram-Bot

Luftqualitätsmessung mit Telegram-Anbindung



89

Projekte

FOKUS

6 Super-Servo-Tester

Bis zu vier Servos einzeln oder eingebaut testen

FOKUS

34 Aktives 1-kHz-Filter für Verzerrungsmessungen

Bessere Messungen durch Optimierung des Messsignals

48 BlueRC

IR-Fernsteuerung mit Smartphone und ESP32

FOKUS

74 Energielogger

Energiebedarf messen und aufzeichnen

82 Parkscheibe mit E-Paper Display

Ein innovativer digitaler Ersatz

FOKUS

89 eCO₂-Telegram-Bot

Luftqualitätsmessung mit Telegram-Anbindung

98 RFID-Tag lesen und RFID-Türschloss

Projektbeispiele aus dem Arduino-Experimentierpaket von Elektor

FOKUS

103 Oszilloskop-Stromsonde für HF

Hochfrequenz-Strommessung leichtgemacht

Vorschau

Elektor Juli/August 2023

Das nächste Heft ist wie immer randvoll gefüllt mit Schaltungsprojekten, Grundlagen sowie Tipps und Tricks für Elektroniker. Schwerpunkt der nächsten Ausgabe ist: Sensoren und das Internet der Dinge.

Aus dem Inhalt:

- > Energiesparende LoRa-Wetterstation
- > Impedanz-Analysator mit ESP32
- > GPS-basiertes Tachometer
- > Doppler-Sensor in der Praxis
- > Drehscheiben-Telefon als Fernsteuerung
- > Alles über das Matter-Protokoll
- > NTP-Uhr mit CircuitPython auf dem Raspberry Pi Pico W
- > Schritt für Schritt TinyML verstehen

Und vieles mehr!

Elektor **Juli/August 2023** erscheint am 12. Juli 2023.



Roboterarm-Bausatz
Mit Raspberry Pi Pico
und MicroPython

106



IM FOKUS

Prüf- und Messtechnik



Super-Servo-Tester

Bis zu vier Servos einzeln oder eingebaut testen



Bild 1. Der Quadcopter, für den der Super-Servo-Tester entwickelt wurde.

Von Olivier Croiset (Frankreich)

Ich habe den hier vorgestellten Super-Servo-Tester entwickelt, weil die Drohne, an der ich arbeite, bei Testflügen manchmal einen unangenehmen Geruch nach überhitzter Elektronik und verschmortem Kupferlack verströmte. Ich brauchte ein Werkzeug, um das Problem zu diagnostizieren und die Ursache zu verstehen.

Im Internet finden Sie viele Servotester zum Testen eines einzelnen Servos. Damit können Sie überprüfen, ob der Servo, das Sie gerade gekauft haben, richtig funktioniert. Der hier vorgestellte Tester geht jedoch einige Schritte weiter. Sie können damit nicht nur bis zu vier Servos gleichzeitig testen, sondern auch die elektronische Drehzahlregelung (ESC) oder den Flugregler zusammen mit Servos in verschiedenen Konfigurationen. Die Drohne, für die ich den Super-Servo-Tester benötigte, ist in **Bild 1** zu sehen. Sie basiert auf [1].

Einmaleins der Servosignale

Zur Erinnerung: Ein Servo wird durch ein PWM-Signal mit einer Frequenz (oder Pulswiederholungsrate, PRR) von 50 Hz gesteuert. Die Impulsbreite (die Zeit, in der das Signal high ist) sollte im Bereich von ein bis zwei Millisekunden liegen, für den Rest der Zeit ist das Signal low. Der Servo befindet sich in seiner Mittelstellung, wenn die Impulsbreite 1,5 ms beträgt. Wenn der Impuls kürzer

ist, dreht sich der Servo in eine Richtung, wenn er länger ist, dreht er sich in die andere Richtung.

Bei einem typischen Flugsystem sind fünf Parameter zu überprüfen: die Dauer der vier Servo-Befehlsimpulse (Schub, Rollen, Neigen und Gieren) und die Versorgungsspannung. Bei komplexen Systemen, etwa bei der Entwicklung eines Flugcontrollers (einer Drohne oder eines anderen RC-Modells) oder eines Pulsmischers, zeigt dieser Tester das Ergebnis des Mischers an.

Alles, was das Gerät an Servos testen kann, trifft auch auf ESC oder jedes andere Gerät zu, das Servosignale erzeugt und/oder erkennt.

Funktionen und Möglichkeiten

Der Super-Servo-Tester (SST) misst die Dauer der Steuerimpulse von bis zu vier Servosignalen und gibt Auskunft über die Qualität der Stromversorgung. Er kann zwischen den Fernsteuerungsempfänger und die Servos, zwischen den Empfänger und den Flugregler der Drohne oder zwischen den Flugregler und die Servos geschaltet werden. In **Bild 2** bis **Bild 6** sind die möglichen Konfigurationen zu sehen.

Betriebsarten

Der SST verfügt über zwei Betriebsmodi, die über einen Schiebeschalter ausgewählt werden können:

- In diesem Modus erzeugt der SST die Impulse für vier Servos oder für den Flugregler. Die Breite der

Impulse wird mit vier Potentiometern gesteuert. Hier versorgt der SST die Servos (oder den ESC) mit Strom, wobei die Stromversorgung des Modells mit keinem von ihnen verbunden sein darf. Die Versorgungsgleichspannung des SST muss zwischen 7,5 V und 12 V liegen.

- In diesem Modus werden die Längen der vom Empfänger oder Regler kommenden Impulse gemessen. Die Signale werden auch an die Ausgänge des SST weitergeleitet, um die Servos (oder den Flugregler) zu steuern. In diesem Modus werden der SST und die Servos (oder ESCs) von der Stromversorgung des Modells versorgt. Die Versorgungsspannung darf 7,49 V nicht überschreiten und der SST sollte nicht an seine eigene Spannungsversorgung angeschlossen werden. Außerdem müssen alle vier Eingangskanäle angeschlossen sein, sonst signalisieren die LED und der Summer des SST einen Fehler.

Anzeigemodi

Das Display zeigt die Dauer der Impulse grafisch in vier Balkendiagrammen zusammen mit ihrem numerischen Wert in Mikrosekunden an (von 1000 μ s bis 2000 μ s). Die Balkendiagramme sind auf den Bereich 1000 μ s bis 2000 μ s begrenzt, der numerische Wert jedoch nicht.

Wenn ein Wert außerhalb der Grenzen liegt, wird ein Rahmen um den numerischen Wert gezeichnet. In diesem Fall leuchtet die LED auf und der Summer piept, falls er eingeschaltet ist.

Das Display zeigt auch den Wert der Versorgungsspannung der Servos in beiden Betriebsarten an (*Manual* oder *Inputs*). In der Tat ist die Qualität der Stromversorgung des Modells wichtig für den Flugregler und für die Sicherheit des Piloten und der Menschen, die seine Flugkünste bewundern. Die gemessene Versorgungsspannung wird mit einem Rahmen versehen, wenn ihr Wert unter 4,5 V liegt. Die LED leuchtet ebenfalls auf und der Summer piept, sodass Sie den Tester nicht die ganze Zeit beobachten müssen.

Der SST bietet zwei Anzeigemodi, *Stack* und *Square* (siehe **Bild 7**). Für einen Quadrocopter ist der zweite Typ besser geeignet. Die Position des Potentiometers P4 beim Einschalten bestimmt den Anzeigemodus: Wenn P4 vor dem Einschalten ganz nach links gedreht wird, verwendet der SST die *Stacked*-Anzeige. Wenn Sie es nach rechts drehen, wird beim Einschalten die *Square*-Anzeige aktiviert.

Die Schaltung

Da wir nun wissen, wie man den Tester benutzt, lassen Sie uns einen Blick auf die Schaltung in **Bild 8** werfen. Sie ist nicht allzu kompliziert, da sie hauptsächlich aus Anschlüssen besteht.

Das Herzstück der Schaltung ist ein ATmega328P-Mikrocontroller, wie er auch auf einem Arduino UNO zu finden ist. Seine Taktfrequenz wird von X1 bestimmt,

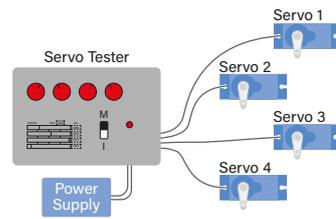


Bild 2. Konfiguration 1, einfacher Test von vier Servos (*Manual-Modus*), ermöglicht den gleichzeitigen Test von vier Servos. Die Servos werden über den Tester mit Strom versorgt und mit den Potentiometern gesteuert.

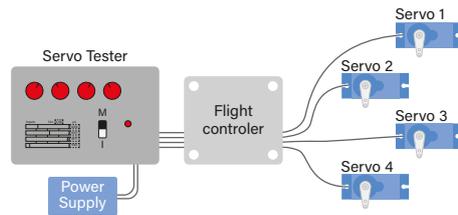


Bild 3. Konfiguration 2, Flugcontroller-Test (*Manual-Modus*), ermöglicht das Testen eines (Drohnen-) Flugcontrollers ohne Sender und Empfänger. Die Servos und der Flugregler werden vom Tester mit Strom versorgt und mit den Potentiometern gesteuert.

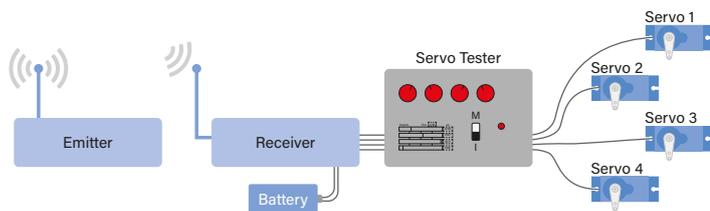


Bild 4. Konfiguration 3, Test von Sender und Empfänger (*Inputs-Modus*) zur Überprüfung von Sender und Empfänger. Der Tester, die Servos und der Flugregler werden über den Akku des Empfängers mit Strom versorgt.

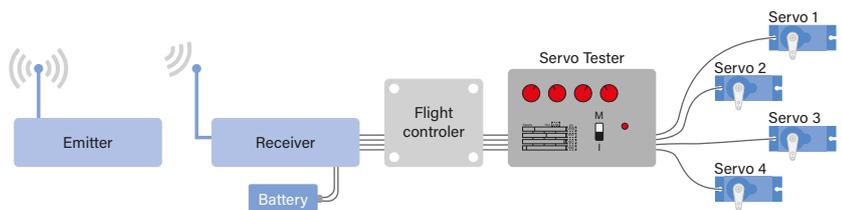


Bild 5. Konfiguration 4, Sender-, Empfänger- und Flugreglertest (*Inputs-Modus*), dient zur Überprüfung der korrekten Funktion von Sender und Empfänger mit dem Flugregler (im Fall einer Drohne). Das Testgerät, die Servos und der Flugregler werden über den Akku des Empfängers mit Strom versorgt.

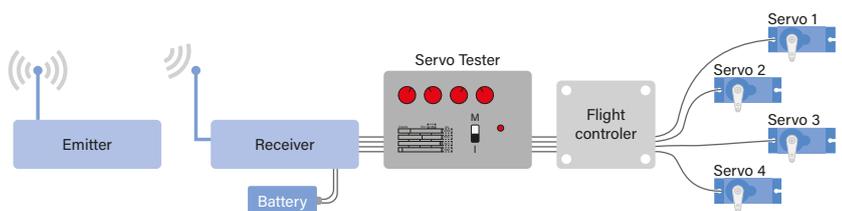


Bild 6. Konfiguration 5, Sender- und Empfängertest (*Inputs-Modus*), ermöglicht das Testen des Flugreglers mit dem Sender und dem Empfänger. Das Testgerät, die Servos und der Flugregler werden über den Akku des Empfängers mit Strom versorgt.



Analoge Eingänge

Die vier Potentiometer sind mit den analogen Eingängen der MCU an PC0...PC3 verbunden. Die Versorgungsspannung ist über einen Spannungsteiler (R4, R5, R6) mit dem analogen Eingang PC4 verbunden. Das Verhältnis zwischen (R4+R5) und R6 muss 2:1 sein, aber ihre absoluten Werte sind nicht entscheidend. Die Verwendung von drei Widerständen mit dem gleichen Wert vereinfacht die Auswahl nach Genauigkeit.

Um die Versorgungsspannung des Mikrocontrollers zu messen, wird für den Analog-Digital-Wandler (ADC) eine von der Versorgungsspannung unabhängige Referenzspannung benötigt. Der ATmega328P verfügt über eine eingebaute Referenz von 1,1 V, aber dieser Wert ist etwas niedrig. Ich habe daher D2, einen LM385-2.5, als externe Referenz von 2,5 V verwendet. Dieser Baustein ist genauer als eine einfache Zenerdiode mit zwei Anschlüssen, was die Messgenauigkeit der Servo-Versorgungsspannung verbessert: 1 % bis 2 % statt etwa 5 % bei einer üblichen Zenerdiode.

Beachten Sie, dass D2 ein recht hohes Gehäuse hat. Um die Schaltung flach zu halten, kann das Bauteil rechtwinklig umgebogen werden. Andererseits ist es auch möglich, es als Auflage für das Display zu verwenden. Die gemessene Versorgungsspannung darf nicht höher als 7,49 V sein, da die maximale Eingangsspannung der MCU 5 V beträgt. Daher darf die Stromversorgung des Empfängers und der Servos diesen Wert niemals überschreiten.

Anmerkungen zum Display

Das SSD1306-basierte I²C-OLED-Display wird auf den Anschluss K4 gesteckt. Der I²C-Port des Displays ist nicht mit dem I²C-Port der MCU verbunden, sondern mit PD0 und PD1. Der I²C-Bus wird von der Software emuliert. Das liegt daran, dass beim ATmega328 der I²C-Bus sich seinen Pin mit dem Analogeingang PC4 teilt, der bereits für die Messung der Versorgungsspannung verwendet wird. Daher kann die eingebaute I²C-Peripherie hier nicht verwendet werden.

R9 und R10 sind Pull-ups für den I²C-Bus. Offiziell sollten sie einen Wert von etwa 4,7 k Ω haben, aber 10 k Ω funktionieren auch und eine Zeile in der Stückliste ist auch gespart, da die meisten anderen Widerstände ebenfalls einen Wert von 10 k Ω aufweisen.

Beachten Sie, dass K4 als 2x4-poliger Anschluss gezeichnet ist, aber auf der Platine sollten Sie eine 1x4-polige Buchsenleiste entweder in Position „A“ oder „B“ löten, je nach Displaytyp. Bei diesen Displays befinden sich nämlich die Stifte GND und VCC an Pin 3 und Pin 4, aber nicht immer in der gleichen Reihenfolge. Mit K4A und K4B können Sie beide Typen verwenden. Zwar wird durch die beiden Reihen mehr Platz auf der Platine benötigt als nötig, letztendlich aber weniger, als wenn Drahtbrücken eingesetzt werden müssten. Dies erkaufte man sich mit dem Nachteil, dass die Position der Displayöffnung im Gehäuse vom Displaytyp abhängt. Außerdem können die

Abmessungen des Displays je nach Hersteller variieren. Daher müssen Sie Ihr Display *vor* der Bearbeitung des Gehäuses auswählen.

Auswahl der Betriebsart

Der zweipolige Umschalter (DPDT), Schiebeschalter S1, wählt den Betriebsmodus des Testers aus. Im *Manual*-Modus wird die 5-V-Versorgungsspannung mit den Servoanschlüssen verbunden. Im *Inputs*-Modus ist die Versorgungsspannung des SST U_{BATT}, sodass seine eigene 5-V-Versorgung abgeschaltet werden muss, um Konflikte zu vermeiden. Eigentlich sollten Sie in diesem Modus die Stromversorgung des SST abklemmen, aber um auf Nummer sicher zu gehen, unterbricht S1 die Verbindung für Sie. Die Zenerdiode D3 sorgt zusammen mit R8 dafür, dass die von U_{BATT} abgeleitete Stromversorgung für den Rest der Schaltung die für die anderen Bauteile erlaubten Maximalwerte nicht überschreitet. Die Position von S1 wird von PD4 gelesen, sodass die MCU den entsprechenden Betriebsmodus aktivieren kann. Streng genommen muss S1 gar kein DPDT-Typ sein, aber er muss in der Lage sein, die von bis zu vier Servos verbrauchte Leistung weiterzuleiten und sogar noch mehr, wenn an den Ausgängen auch ein ESC oder Flugregler angeschlossen sind. Geeignete DPDT-Schiebeschalter sind oft billiger als einpolige Typen, daher wird hier ein solcher verwendet.

Verschiedenes

Der GPIO-Port PD2 der MCU steuert die Alarm-LED und den Summer. Da sie parallel geschaltet sind, sorgt der MOSFET T1 für den zusätzlichen „Doppelwumms“, um beide anzusteuern, ohne die MCU zu überlasten. S2 ist ein zweiter Schiebeschalter, mit dem Sie den Summer stummschalten können, da er manchmal etwas störend sein kann. Dieser Schalter ist ein kleiner Typ für niedrige Leistung.

Die 5-V-Gleichspannungsversorgung für den SST im *Manual*-Modus wird von einem klassischen linearen 7805-Spannungsregler (IC1) geliefert. Achten Sie darauf, einen kräftigen TO220-Typ zu verwenden, da er bis zu vier Servos versorgen muss. Um die Baugruppe niedrig zu halten, können C1 und C2 liegend montiert werden. Schließlich gibt es noch den Anschluss K3, einen „Wannenstecker“, über den Sie den Mikrocontroller programmieren können, ohne ihn aus der Schaltung zu entfernen. Er ist auf die gleiche Weise verdrahtet wie der Arduino-ICSP-Anschluss (In-Circuit Serial Programming).

Ein paar Worte zur Software

Für dieses Projekt habe ich die Arduino-IDE verwendet, was beweist, dass diese auch für komplexe Projekte verwendet werden kann. Ich habe zunächst einen Prototyp mit einem Arduino UNO und einem Breadboard erstellt. Als alles wie gewünscht funktionierte, habe ich das Programm über den ICSP-Anschluss in das endgültige

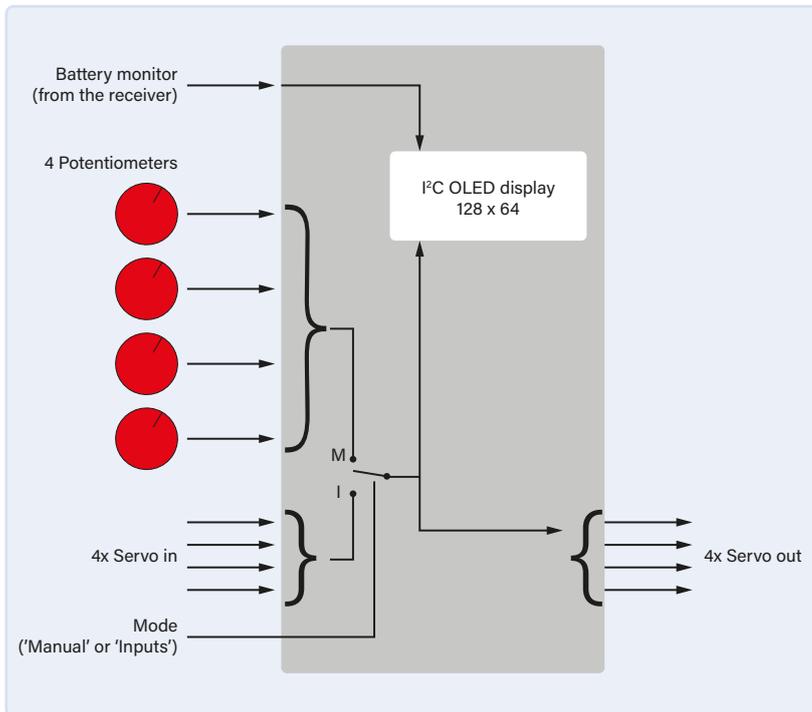


Bild 9. Schematische Übersicht der Signalflüsse im Inneren des Mikrocontrollers.

Gerät geladen. Beachten Sie, dass bei der Programmierung des Mikrocontrollers über den ICSP-Anschluss K3 nichts an die Anschlüsse K1 und K2 angeschlossen werden darf, da sich diese Anschlüsse jeweils ein Signal mit K3 teilen. Alle für das Projekt benötigten Bibliotheken sind im Download [3] enthalten, mit Ausnahme der Servo-Bibliothek, da diese ohnehin Teil der Arduino-IDE ist. Das Projekt ist in drei Dateien organisiert: *Display.ino*, *Inputs.ino* und die Sketch-Hauptdatei. Die Namen der Dateien erklären recht gut, was sie enthalten. Die Sketch-Hauptdatei enthält die Funktionen `setup()` und `loop()`, aber auch die Interrupt-Routinen, die zur Zeitsteuerung der Eingangsimpulse verwendet werden. Pegeländerungen in den Eingangssignalen erzeugen Pin-Änderungsinterrupts. Auf diese Weise kann die Funktion `micros()` zur Messung der Impulsdauer verwendet werden. Die gemessenen Pulsbreiten werden auf den Ausgang kopiert und an die Display-Funktionen für die Anzeige weitergegeben. **Bild 9** zeigt eine Übersicht der wesentlichen Signalflüsse innerhalb des Mikrocontrollers.

Verwendung der OLED_I2C-Bibliothek

Wie bereits erwähnt, wird das OLED-Display über einen I²C-Bus gesteuert. Beim ATmega328P wird dieser Bus mit dem ADC geteilt. PC4 kann entweder der SDA-Pin oder der Analogeingang A4 sein. Um dies zu umgehen, wird der I²C-Bus in Software emuliert. Die OLED_I2C-Bibliothek von Rinky-Dink Electronics [2] ermöglicht dies und erlaubt es uns, dem I²C-Bus einen beliebigen GPIO-Port zuzuweisen.

Ich musste die Datei *OLED_I2C.cpp* dieser Bibliothek leicht abändern, da eine falsche `include`-Anweisung die Kompilierung verhinderte. Ersetzen Sie Zeile 25:

```
#include "hardware/avr/HW_AVR.h"
```

durch: `#include "HW_AVR.h"`

Außerdem habe ich in der Datei *HW_AVR.h* in der Funktion `OLED::update()` die Zeile 24 wie folgt auskommentiert:

```
//noInterrupts();
```

Das ist notwendig, da Interrupts von der Servo-Bibliothek für die Erzeugung der 50-Hz-PWM-Signale an den Ausgängen des SST verwendet werden. Sie müssen auch für die Messung der eingehenden Impulse aktiviert bleiben.

Vor dem Einschalten

Wenn sich das Potentiometer P1 beim Einschalten in der Minimalposition (ganz nach links gedreht) befindet, zeigt das Display einen weißen Rahmen an. Dies hilft bei der Positionierung der Gehäuseöffnung für das Display und ermöglicht bei Bedarf deren Anpassung. Nicht jeder hat eine CNC-Maschine zu Hause! Drehen Sie P1 in die Mitte, um zum normalen Betrieb zurückzukehren.

Bild 10. Der Prototyp des Autors steuert vier Servos im Manual-Modus.

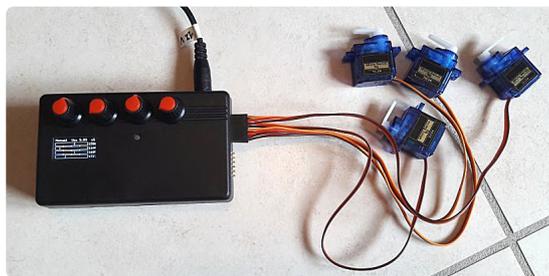
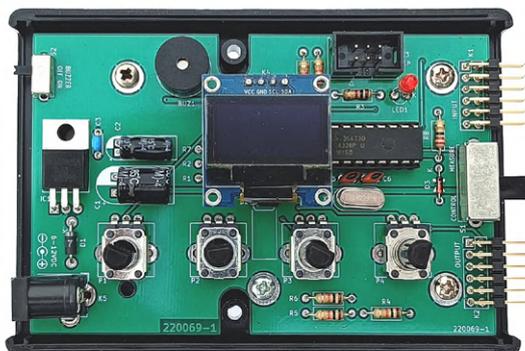
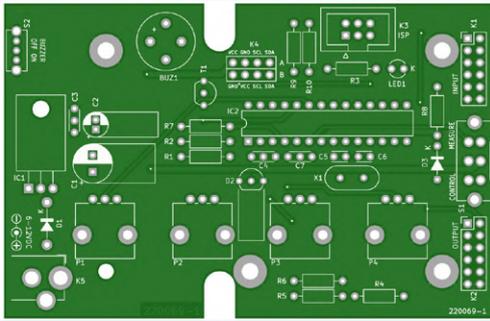


Bild 11. Der Prototyp von Elektor Labs in seinem Gehäuse.





STÜCKLISTE



(alle Bauteile bedrahtet)

Widerstände (alle 0,25 W, 5 %):

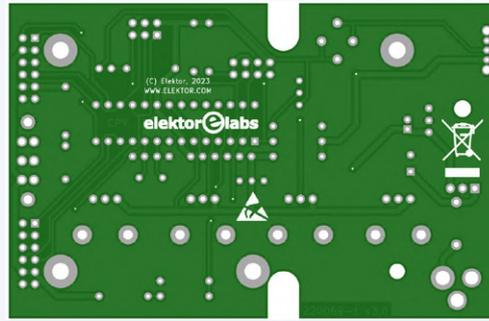
R1, R3 = 1 k
R2, R4, R5, R6, R7, R9, R10 = 10 k
R8 = 22 Ω
P1, P2, P3, P4 = 10 k, linear, vertikal

Kondensatoren:

C1 = 100 μ, 35 V, 8 mm Durchm., Raster 3,5 mm
C2 = 10 μ, 16 V, 5 mm Durchm., Raster 2 mm
C3, C4, C7 = 100 n, Raster 2,5 mm oder 5 mm
C5, C6 = 22 p, Raster 2,5 mm

Halbleiter:

D1 = 1N5817
D2 = LM385Z-2.5, TO-92
D3 = BZX79-C5V1
IC1 = 7805, TO220
IC2 = ATmega328-P
LED1 = LED, 3 mm, rot
T1 = 2N7000, TO-92



Außerdem:

BUZ1 = Piezo-Summer mit Oszillator, 12 mm Durchm.,
Raster 6,5 mm oder 7,6 mm
K1, K2 = 2×6-polige Stiftleiste, Raster 2,54 mm, gewinkelt
K3 = 2×3-poliger Wannenstecker, Raster 2,54 mm
K4* = 1×4-polige Buchsenleiste, Raster 2,54 mm
(Position A oder B je nach Display)
K5 = DC-Hohlbuchse

S1 = Schiebeschalter DPDT, 90° (z. B. MFS201N-16-Z)
S2 = Schiebeschalter SPDT, 90° (z. B. OS102011MA1)
X1 = Quarz, 16 MHz, HC-49S (Low-Profile)
OLED-Display, 0,96", 128 x 32 Pixel, 4-polig I2C
(z.B. www.elektor.de/18747)
Platine 220069-1
Empfohlenes Gehäuse: Hammond 1593N

* = Siehe Text

Wenn sich P4 beim Einschalten in der Minimalposition befindet, verwendet das Display den *Stacked*-Modus. Befindet sich P4 in der maximalen Position, wird der *Square*-Modus aktiviert. Im letzteren Fall bedeutet das, dass der an Ausgang 4 angeschlossene Motor beim Einschalten mit voller Geschwindigkeit läuft.

Zeit zum Abheben

Es hat Spaß gemacht, den Super-Servo-Tester zu entwerfen und zu entwickeln, und er ist ein nützliches Werkzeug für alle, die sich mit Drohnen und Quadrocoptern oder dem RC-Modellbau im Allgemeinen beschäftigen. Dieses Projekt hat mir nicht nur geholfen, meine Programmierkenntnisse zu verbessern, sondern ich kann jetzt auch endlich meine Drohne mit nach draußen nehmen, die schon ungeduldig darauf wartet, abzuheben! ◀

Jörg Starkmuth — 220069-02

Sie haben Fragen oder Kommentare?

Wenn Sie technische Fragen oder Anmerkungen zu diesem Artikel haben, können Sie sich gerne an die Elektor-Redaktion unter der E-Mail-Adresse redaktion@elektor.de wenden.



Passende Produkte

- > **Arduino Uno R3**
SKU 15877: www.elektor.de/15877
- > **OLED-Display 0,96" 128×64 I2C**
SKU 18747: www.elektor.de/18747

WEBLINKS

- [1] Drohnen-Website von Joop Brokking: http://www.brokking.net/ymfc-al_main.html
- [2] OLED_I2C-Bibliothek: <http://www.rinkydinkelectronics.com/index.php>
- [3] Projekt-Downloads bei Elektor Labs: <https://www.elektormagazine.com/labs/super-servo-tester>



Analoge Signale und der Mikrocontroller

ADCs, DACs, Strommessung und mehr

Von Mathias Claußen (Deutschland)

Während eine MCU intern mit Einsen und Nullen hantiert, ist die Welt um sie herum analog aufgebaut. So ist es manchmal nötig, analoge Werte zu lesen oder auszugeben. Mit nur wenigen Bauteilen lässt sich eine MCU mit der analogen Welt verbinden.

Egal, ob der Mikrocontroller (MCU) der Wahl Arduino Uno, STM32 oder sonst wie heißen mag, oft ist es nötig, mehr als nur digitale Signale zu verarbeiten. Die Welt um uns herum ist analog aufgebaut; es gibt nicht nur hell oder dunkel, sondern unendliche viele Schattierungen dazwischen. Daher soll dieser Artikel einen Einblick geben, wie analoge Werte von einem Mikrocontroller gelesen und wie auch wieder ausgegeben werden können. Für den erfahrenen Entwickler wird dieser Text sicherlich unter „leichte Lektüre“ fallen, doch jeder von uns musste in seinem Entwicklerleben irgendwann einmal anfangen. Und während das Setzen und Lesen von digitalen Pins sehr überschaubar ist, wird es bei der Ein- und Ausgabe von analogen Werten schon etwas kniffliger.

Die grundlegenden Fragen sind: Wie kommen die analogen Werte in eine digital arbeitende MCU und wie müssen diese dazu aufbereitet werden? Und wie kann eine MCU analoge Werte ausgeben? Dazu betrachten wir erst einmal zwei wichtige grundlegende Elemente, den Analog-Digital-Wandler (Analog-to-digital-Converter, ADC) und den Digital-Analog-Wandler (Digital-to-analog-Converter, DAC).

Analog-Digital-Wandler (ADC)

Der Analog-zu-Digital-Umsetzer oder -Wandler ist in der Lage, ein analoges Signal in eine digitale Repräsentation umzusetzen. Mit dieser digitalen Repräsentation kann dann später der Mikrocontroller die Daten weiterverarbeiten. Zwei häufig anzutreffende Kenngrößen eines ADC ist die Auflösung (*Resolution*) in Bits und die Abtastrate (*Sample Rate*) in Samples pro Sekunde. Bei Mikrocontrollern liegt die Auflösung meist im Bereich von 10...12 Bit.

Das bedeutet, dass der ADC seinen analogen Eingangsspannungsbereich in 1024 Werte (10 Bit) beziehungsweise 4096 Werte (12 Bit) einteilt und die anliegende Spannung einem dieser Werte zuweisen kann. Der Wertebereich ist nun endlich und das analoge Signal muss dem am besten passenden digitalen Wert zugewiesen werden, auch wenn es nicht exakt der rechnerisch festgelegten Stufe entspricht. Dieser Fehler wird Quantisierungsabweichung (*Quantization error*) genannt.

Die zweite Größe, die Abtastrate in Samples per Second, gibt an, wie oft pro Sekunde der ADC in der Lage ist, einen analogen Wert in einen digitalen Wert umzusetzen. Wenn das Signal nicht gerade eine Gleichspannung ist, bestimmt die Abtastfrequenz auch die maximale Bandbreite eines Signales, die verarbeitet werden kann. Der Kehrwert der Abtastfrequenz gibt an, wie lange es dauert, bis der ADC eine Analog-zu-Digitalwandlung durchgeführt hat, also wie lange die Software warten muss, bis nach dem Start einer Wandlung ein Wert verarbeitet werden kann.

Spannungsbereiche, Auflösung und Referenzen

Gehen wir von einem Eingangsspannungsbereich von 0 V bis 5 V bei 10 Bit Auflösung aus, so ergeben sich

$$5 \text{ V} / (2^{10} \text{ bit} - 1) = 4,89 \text{ mV/bit}$$

Eine Spannung von beispielsweise 452 mV würde damit intern den Wert 92 erhalten

$$452 \text{ mV} / 4,89 \text{ mV/bit} = 92$$

und damit als 449,88 mV weiterverarbeitet werden. Die Quantisierungsabweichung beträgt also 2,12 mV. Mehr als 0,5 LSB (Least Significant Bits), hier knapp 2,5 mV, kann die Abweichung aber nicht betragen.

Sollte der Spannungsbereich, der gemessen werden soll, deutlich kleiner sein als der erlaubte Eingangsspannungsbereich des ADC, so kann dieser bei den meisten ADCs modifiziert werden. Der ADC misst die Analogwerte immer gegenüber einer Referenzspannung. Viele MCUs stellen heutzutage eine interne Referenzspannung oder gleich mehrere zur Verfügung. Als Beispiel soll hier der AVR128DB von Microchip dienen. In **Bild 1** ist das Blockschaltbild der Spannungsreferenz dieses Controllers zu sehen.

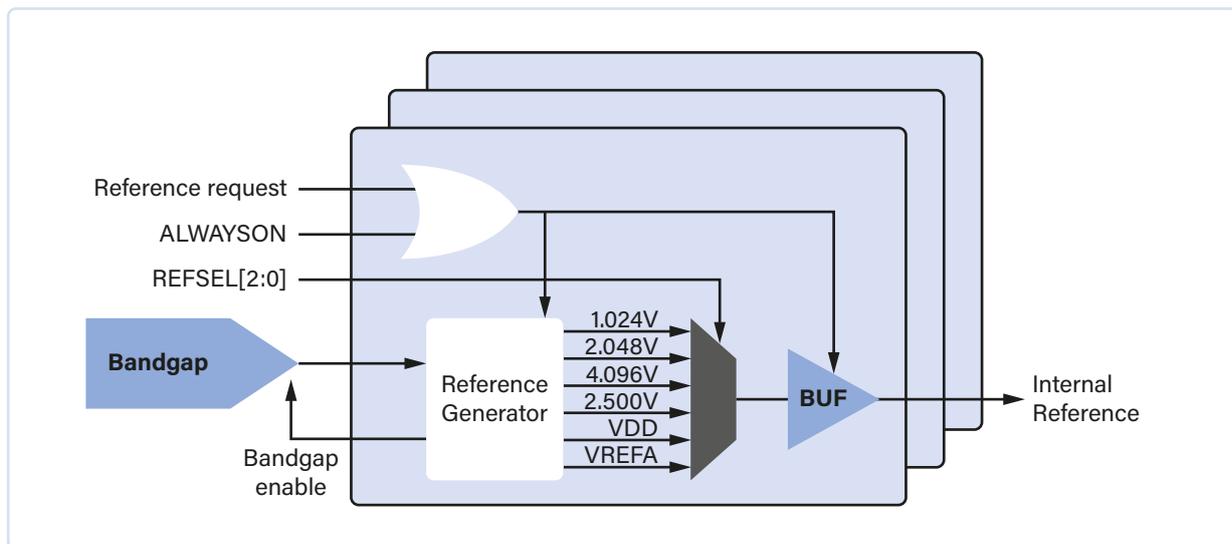


Bild 1. Spannungsreferenz AVR128DB (Quelle: Microchip, <https://bit.ly/3XFS5JH>).

Es stehen sechs Referenzspannungen von 1,024 V, 2,048 V, 4,096 V, 2,5 V, VDD und VREFA bereit. Die ersten drei Spannungen lassen sich als 2^N darstellen und erlauben es später, die digitale Werte des ADC sehr einfach wieder in Volt zurückzurechnen. Etwas, was bei fast allen MCUs zu finden ist, betrifft GND (Masse) und AGND (Masse für den analogen Teil der MCU). Da im digitalen Bereich hochfrequente Störungen vorhanden sein können, die sich auch durch den Massepfad bewegen, ist alles, was ADC oder DAC ist, mit einer eigenen Masse ausgestattet. So kann im Schaltungsdesign GND und AGND voneinander entkoppelt werden, um Störungen zu vermeiden. Wenn zu erwarten ist, dass die Messgröße nicht größer als 1 V wird, so lässt sich 1,024 V als Referenz wählen. Dies würde dazu führen, dass die Auflösung 1,00098 mV/Bit beträgt. 452 mV entsprechen dem digitalen Wert 452; bei der Weiterverarbeitung wäre der Quantisierungsfehler damit deutlich kleiner als im obigen Beispiel. Doch nicht nur die passende Wahl der Referenzspannung kann dafür sorgen, dass die Ergebnisse der Wandlungen näher am physikalischen (analogen) Wert liegen. Auch die Qualität der Referenzspannung hat darauf Einfluss.

Beim AVR128DB sind noch VDD und VREFA als Spannungsreferenz zu sehen. VDD ist in diesem Fall die Versorgungsspannung des Chips und VREFA ein externer Pin am AVR128DB, der es erlaubt, eine eigene externe Referenzspannungsquelle anzuschließen. VDD als Referenz kann eine unglückliche Wahl sein, da diese Spannung mit Störungen aus dem digitalen Teil der MCU belastet ist und diese auch nur schwer „weggefiltert“ werden können. Anders sieht es bei VREFA aus: Hier kann eine externe hochgenaue Spannungsquelle als Referenz genutzt werden. Natürlich ist auch diese mit der unsauberen Versorgungsspannung verbunden, aber extern kann sie entsprechend gut gefiltert werden. Eine stabile und störungsfreie Referenz ist die Grundlage einer guten Analog-zu-Digitalwandlung. Es gibt noch eine weitere Möglichkeit, die Auflösung eines ADCs zu erhöhen, ohne die Referenzspannung zu ändern, nämlich mit der sogenannten Überabtastung (Oversampling). Bei der Überabtastung nutzt man das niederwertigste Bit. Dieses wird zwischen Null und Eins hin und her wandern, wenn sich das Signal nicht

zu 100% mit der verfügbaren Auflösung des ADC darstellen lässt. Bildet man den Mittelwert dieses niederwertigen Bits, so ergibt sich ein Bruchteil einer Eins, was eine höhere Auflösung bedeutet. Allerdings wird durch diese Maßnahme die effektive Sample-Rate verringert. Für jedes Bit, das mittels Oversampling hinzugefügt wird, ergibt sich eine Viertelung der Samplerate. Schafft ein ADC zum Beispiel bei 12 Bit maximal 130 ksp/s (Samples pro Sekunde) so wären es bei vier Oversampling-Bits, also 16 Bit, nur noch 507 Samples pro Sekunde =

$$130 \text{ ksp/s} / 4^4$$

Hinzu kommt, dass je nach ADC der Prozessor die für das Oversampling nötige Mittelwertbildung übernehmen muss, was zusätzlichen Rechenaufwand bedeutet. Einige neuere ADC, unter anderem bei einigen STM32-Controllerfamilien, erledigen diese Aufgabe komplett autonom, ohne dass die CPU damit belastet wird. ADCs gibt es in unterschiedlichen Ausführungen. Die gängigsten ADCs in Controllern sind SAR-Typen (Sukzessive Approximation). Für hohe Auflösungen von 24 Bit und mehr bei jedoch relativ geringen Sampleraten (im Bereich ksp/s bis zu einigen Samples pro Sekunde) werden meist Delta-Sigma-ADCs eingesetzt. Die Details einer Delta-Sigma-Wandlung und deren Implementierung in einen CPLD/FPGA (Complex Programmable Logic Device / Field Programmable Gate Array) können in einem Elektor-Artikel [1] nachgelesen werden. Es gibt noch weitere ADC-Typen, von denen der Flash-ADC die schnellste Wandlungszeit verspricht, so dass er häufig in modernen Flashspeichern zum Einsatz kommt.

Digital-Analog-Wandler (DAC)

Mit dem DAC erzeugt man aus einem digitalen Wert wieder einen analogen. Nicht so häufig wie bei ADCs sind bei einigen MCUs DACs im Chip integriert. Beim AVR128DB ist es ein 10-Bit-DAC, der 1024 unterschiedliche Spannungen ausgeben kann. Sollte kein DAC in der MCU vorhanden sein, kann ein externer integrierter DAC wie der Microchip MCP4922 verwendet werden.

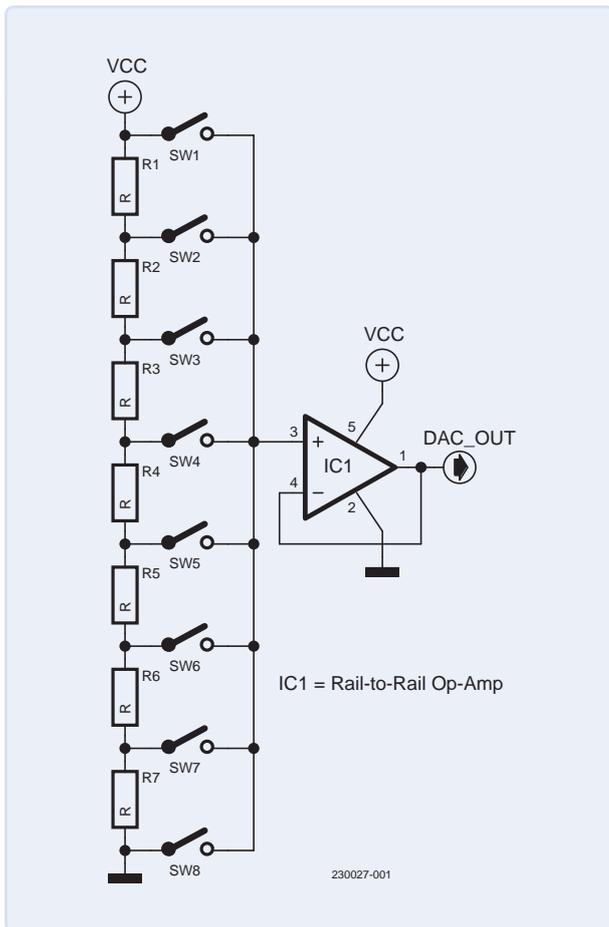


Bild 2. Aufbau eines String-DACs.

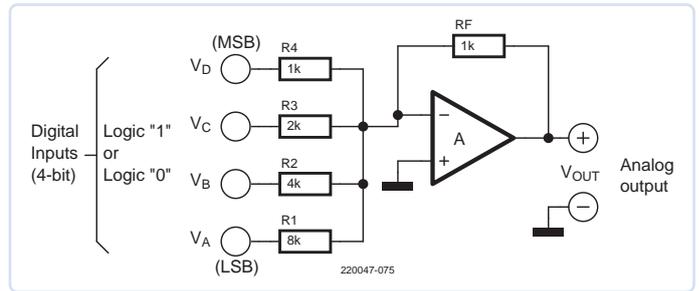


Bild 3. Schaltung eines binär gewichteten DACs.

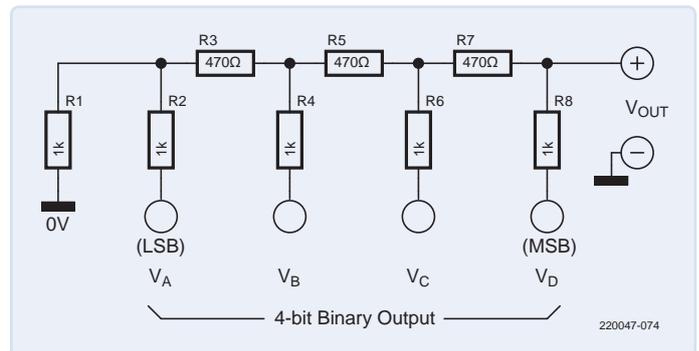


Bild 4. Prinzip eines R2R-DACs.

Beides sind Resistor-String-DACs, der MCP4922 ist ein 12-Bit DAC mit zwei Kanälen. Während sich ein Resistor-String-DAC in einem Chip gut integrieren lässt, ist er diskret mit Festwiderständen kaum realisierbar. Ein Blick auf den Resistor-String-DAC mit drei Bit in **Bild 2** lässt erahnen, warum dies so ist: Im DAC werden

$$2^N - 1 \quad (N = \text{Anzahl der Bits})$$

Widerstände und N Schalter benötigt. Für einen 16-Bit-DAC wären es 65535 Widerstände und 65536 Schalter!

Wie bei einem ADC benötigt auch ein DAC eine Referenzspannung, die den Bereich vorgibt, in dem sich die Ausgangsspannung bewegen kann. Der DAC selbst kann jedoch nur ein paar Milliampere treiben, selbst wenn der Ausgang durch einen Operationsverstärker gepuffert wird.

Wenn genug Pins an einer MCU noch frei sind und kein interner DAC vorhanden ist, gibt es aber andere Wege, einen externen DAC aufzubauen. Einer dieser Wege ist ein binär gewichteter DAC, wie in **Bild 3** zu sehen. Dabei werden für N Bit nur N+1 Widerstände und ein Operationsverstärker benötigt. Der Wert der Widerstände ergibt sich zu

$$R_N = R_{(N-1)} \cdot 2$$

wird also immer verdoppelt. Ein solcher DAC kann verwendet werden, um zum Beispiel analoge VGA-Signale zu erzeugen, wie im Datenblatt unter [2] zu sehen.

Eine weitere oft anzutreffende Variante ist ein R-2R-DAC. Im Gegensatz zu einem binär gewichteten DAC werden hier für N Bit zwar

N×2 Widerstände und ein Operationsverstärker benötigt, allerdings gibt es nur zwei Widerstandswerte, während bei einem binär gewichteten DAC die Werte für N meist „krumm“ sind. Der Schaltplan für einen solchen DAC ist in **Bild 4** zu sehen.

Wenn jedoch I/O-Pins ein knappes Gut bei der verwendeten MCU sind, kann auch die PWM-Einheit mit etwas externer Beschaltung einen DAC ergeben. Das `analog.Write()` auf einem Arduino Uno erzeugt auf diese Weise analoge Werte mit Hilfe der digitalen Pins. In **Bild 5** ist ein Beispiel aus dem Elektor-Artikel unter [3] für einen Arduino Uno zu sehen.

Ein einfacherer Aufbau ist die Schaltung in **Bild 6**, die nur aus einem RC-Tiefpass zum Glätten des Signals besteht. Ein solcher DAC bringt jedoch Nachteile mit sich. Da hier die PWM geglättet

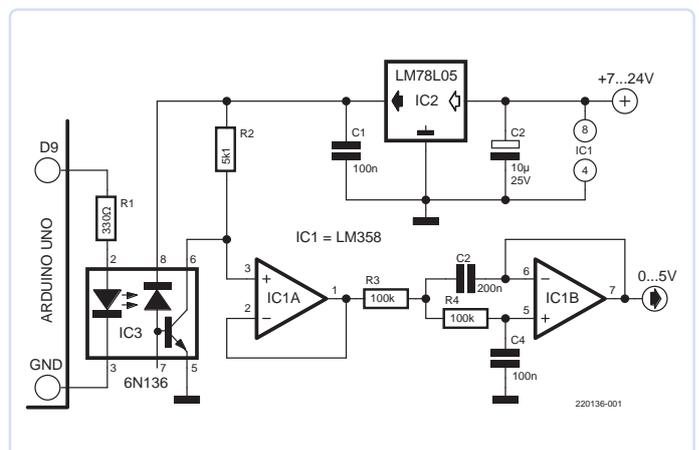


Bild 5. Isolierter Analogausgang für den Arduino Uno.

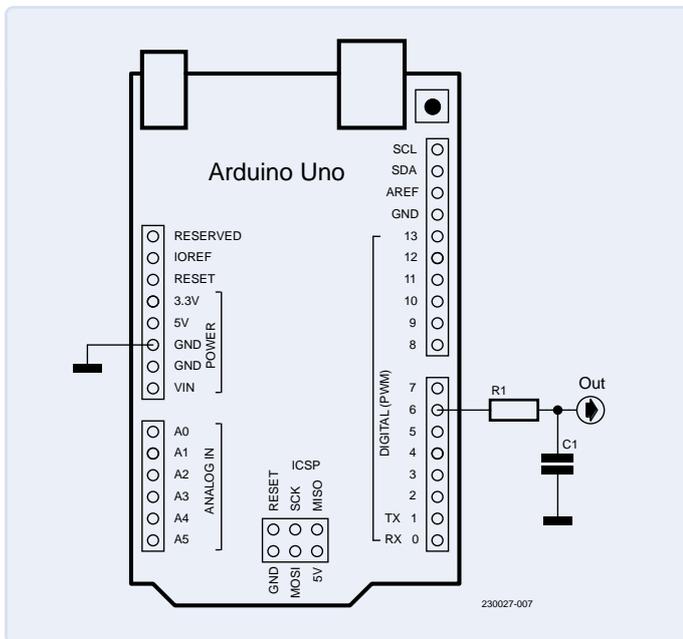


Bild 6. Arduino Uno mit RC-Tiefpass.

werden muss, sind die Werte für R und C für die jeweils gewünschte Anwendung zu berechnen. Es wird auch immer ein kleiner Rest der PWM im analogen Signal zu finden sein und sich als ungewolltes Rauschen bemerkbar machen.

Signalaufbereitung

Nicht immer sind die Signale, die gemessen werden sollen oder aber auch jene, die ausgegeben werden müssen, direkt mit der MCU der Wahl kompatibel. Im digitalen Bereich kennen viele sicherlich die Problematik, 5-V-Komponenten mit 3,3-V-System zu verbinden. Ähnliches ist auch in der analogen Welt und entsprechenden Schnittstellen zu finden. Audio, Video und auch 4-20-mA-Stromschleifen sind nur ein paar Beispiele, die eine Signalaufbereitung benötigen. Die wohl einfachste Form der Signalaufbereitung stellt der Spannungsteiler dar, der nicht nur für analoge Werte, sondern auch zur Anpassung von digitalen Signalen genutzt werden kann.

Spannungsteiler und der ADC

Hier wird der unbelastete Spannungsteiler (**Bild 7**) vorgestellt, eine idealisierte Form des Spannungsteilers. Mit der anliegenden Spannung U ergibt sich über R_1+R_2 ein Strom I . Über den Widerstand R_1 fällt nun eine Spannung von $R_1 \times I = U_1$ und über R_2 eine Spannung $R_2 \times I = U_2$ ab und es gilt $U = U_1 + U_2$. Will man nun zum Beispiel einen Bereich von null bis 10 V mit einem ADC messen, der nur null bis 5 V verarbeiten kann, so muss die Spannung halbiert werden. Es liegt auf der Hand, dass $R_1 = R_2$ sein muss.

Bei der Wahl der Widerstandswerte muss aber auch der Strom I beachtet werden. Zwar würden 10Ω für R_1 und R_2 mathematisch die gewünschten Spannungen für U_1 und U_2 ergeben, aber es würde ein Strom von 500 mA durch die Widerstände fließen. Dies könnte nicht nur die Spannungsquelle überfordern, sondern auch die Widerstände selbst. Denn eine Verlustleistung $P = R \times I^2 = 2,5 \text{ W}$, die jeder Widerstand in Form von Wärme abgeben muss, ist für die meisten kleineren Baugrößen viel zu viel.

Eine andere Option, die Verlustleistung zu reduzieren, wären 1-M Ω -Widerstände. Dann würde nur ein Strom von 5 μA fließen und der Verlust läge bei 25 μW , jedoch schafft dieses andere Extrem ein neues Problem. Ist in dem ADC nämlich eine Sample-and-Hold-Stufe verbaut wie beim Atmega328 in **Bild 8**, so kann es zu falschen

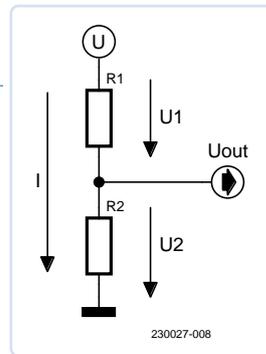


Bild 7. Unbelasteter Spannungsteiler.

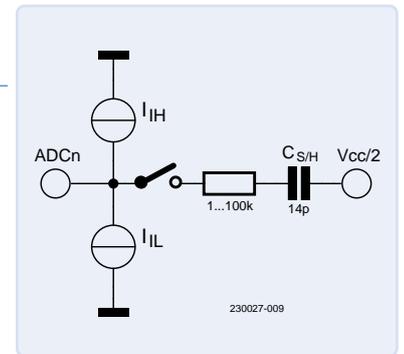


Bild 8. Innerer Aufbau des ADC-Eingangs eines Atmega328 (Quelle: Microchip, <https://bit.ly/405a9Pb>)

Messwerten kommen. Der ADC besitzt an seinem Eingang einen kleinen Kondensator, der geladen wird. Bei einer Umwandlung wird der Kondensator kurzzeitig (für die Dauer der eingestellten Abtastzeit) mit dem analogen Eingang verbunden und geladen. Ist der Strom zu klein, kann es sein, dass der Kondensator während der Abtastzeit nicht vollständig geladen wird. Wandelt der ADC später den Wert des unvollständig geladenen Kondensators, kommt es zu einem fehlerhaften Ergebnis.

Ein Blick in das Datenblatt der MCU der Wahl sollte helfen, hier die passenden Werte zu ermitteln. Für den AVR wäre in diesem Fall ein Wert von jeweils 4,7 k Ω für R_1 und R_2 und damit ein Strom von 1,06 mA angemessen.

Negative Spannungen

Positive Spannungen sind nicht zuletzt durch den Einsatz eines Spannungsteilers recht einfach zu messen. Doch was passiert, wenn die zu messende Spannung nicht positiv ist? Mit einem Opamp als invertierender Verstärker und einem Paar Widerständen wie in **Bild 9** kann die Spannung invertiert werden.

Solange die Spannung am Eingang dieser Schaltung unter 0 V bleibt, arbeitet der Operationsverstärker korrekt. Soll aber nun eine

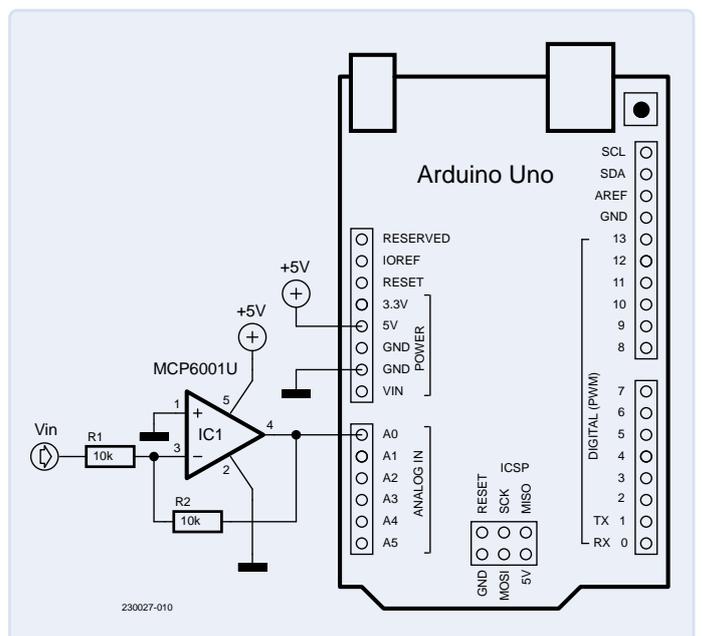


Bild 9. Invertierender Verstärker.

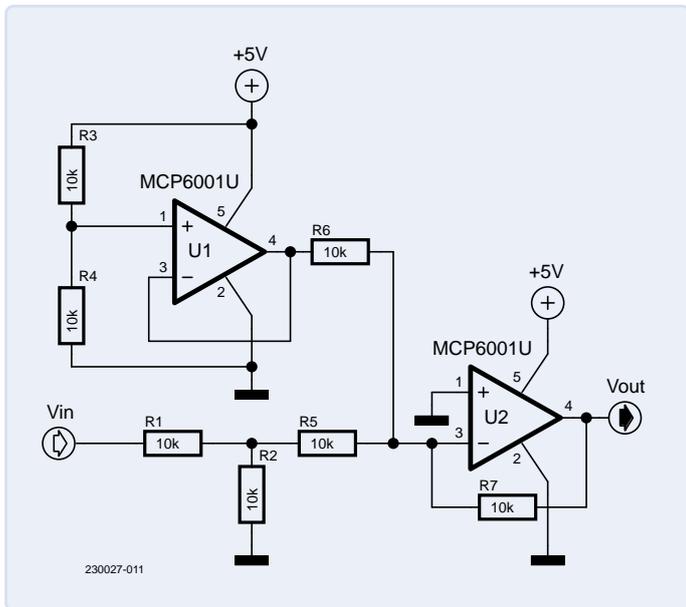


Bild 10. DC-Offset mit Operationsverstärkern.

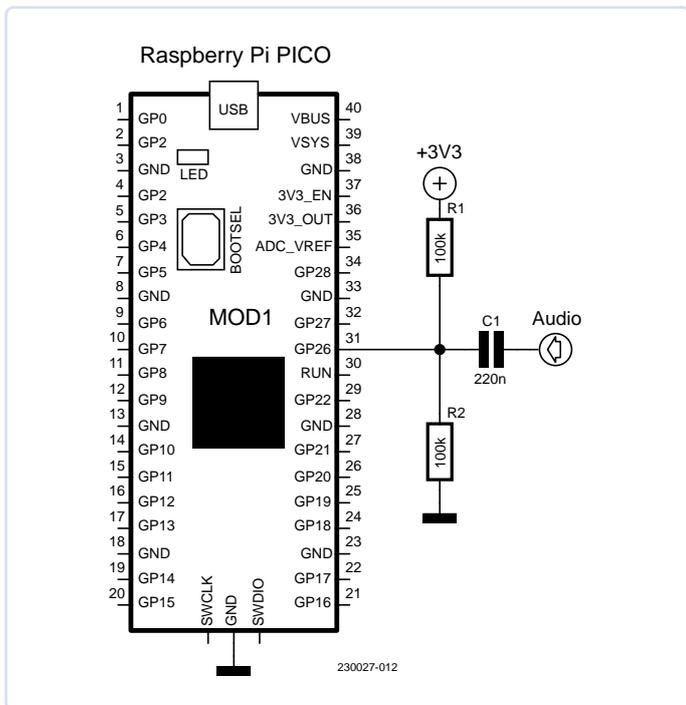


Bild 11. Audioeingang für den ADC eines Microchip AVR.

Spannung von -5 V bis $+5\text{ V}$ gemessen werden, so ist etwas mehr Aufwand erforderlich. Angenommen, der erlaubte Eingangsbereich des Controllers läge bei $0 \dots 5\text{ V}$, muss das Signal mit einem Hub von 10 V zunächst durch einen Spannungsteiler R_1/R_2 halbiert werden. Damit hätten wir es mit Werten von $-2,5\text{ V}$ bis $+2,5\text{ V}$ zu tun. Dieser Bereich wird nun durch den Spannungsteiler R_3/R_4 um $2,5\text{ V}$ angehoben. Der zweite, nicht-invertierende Operationsverstärker U_1 puffert diese Spannung. Der vollständige Schaltplan ist in **Bild 10** zu sehen.

Bei der späteren Auswertung der Messwerte ist zu beachten, dass nun der Nullpunkt der Eingangsspannung um die Hälfte der Versorgungsspannung angehoben ist, in diesem Beispiel also um $2,5\text{ V}$.

Audiosignale und der ADC

Etwas, das interessant sein kann, in einer MCU verarbeitet zu werden, sind Audiosignale. Audiosignale besitzen eine Spitze-Spitze-Spannung von $0,894 V_{SS}$ (-10 dBV) bis $3,472 V_{SS}$ ($+4\text{ dBV}$), je nachdem, welche Quelle das Signal bereitstellt. Die meisten Geräte im Heimbereich arbeiten mit -10 dBV , also $0,894 V_{SS}$. Dabei hat das Signal eine minimale Spannung von $-0,477\text{ V}$ und eine maximale von $+0,477\text{ V}$. Der ADC einer MCU wie dem Raspberry Pi RP2040 kann damit nicht arbeiten. Doch warum sollte man überhaupt Audio mit einem 10-Bit-ADC verarbeiten wollen? Sowohl der AVR als auch schnellere MCU sind durchaus in der Lage, beispielsweise das Spektrum eines Audiosignals zu berechnen und anzuzeigen. Bei dieser Anwendung kann man sich zu Nutze machen, dass der Gleichspannungsanteil nicht von Interesse ist; nur der Wechselspannungsanteil des Audiosignals wird benötigt (C_1) und für den ADC mit einem DC-Offset der halben Versorgungsspannung versehen (R_1/R_2), um das Audiosignal in den positiven Bereich zu heben. Ein Schaltplan hierfür ist in **Bild 11** zu sehen.

Bei dem Signal muss der (recht enge) Spannungsbereich berücksichtigt werden. Mit weniger als 1 V maximaler Spannung muss das Signal verstärkt oder die interne Referenzspannung des ADC passend gewählt werden, um unnötige große Quantisierungsfehler zu vermeiden.

Strommessungen

Der ADC selbst kann in der Regel nur Spannungen messen. Doch auch Ströme wollen erfasst werden! Die einfachste Methode dazu besteht in der Verwendung eines sehr niederohmigen Mess- oder Shunt-Widerstands. Soll die Stromaufnahme einer Schaltung gemessen werden, so gibt es zwei Optionen, nämlich die Low-Side- und die High-Side Messung. In **Bild 12** ist eine sehr einfache Schaltung für eine High-Side-Messung mit dem Arduino Nano zu sehen. Der Messwiderstand wird zwischen die positive Versorgungsspannung und den Verbraucher gesetzt. Um den Strom zu ermitteln,

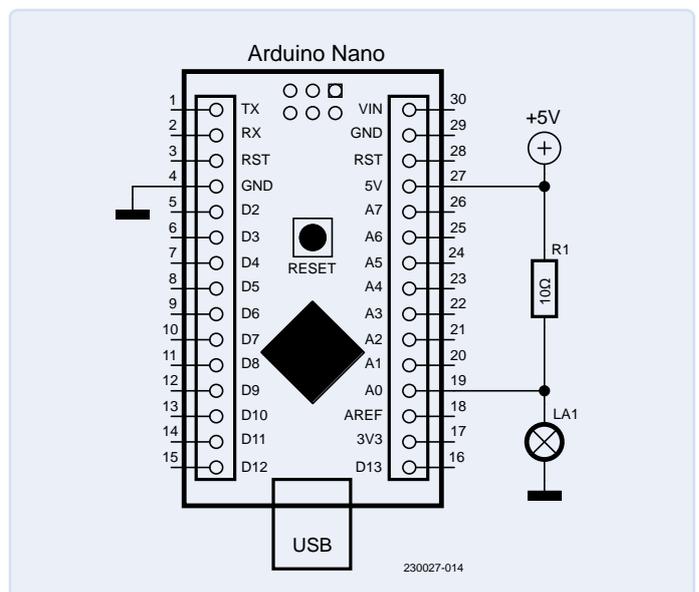


Bild 12. High-Side Messung.

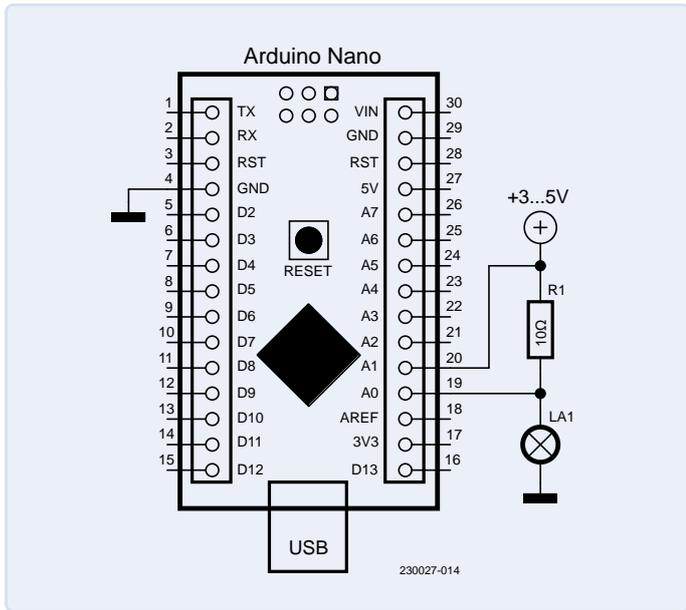


Bild 13. High-Side Messung – vor und hinter dem Messwiderstand.

der durch den Widerstand und damit auch durch die Last fließt, muss der Spannungsabfall über den Widerstand gemessen werden. Sollte die (konstante und bekannte) Spannung, mit der die Last versorgt wird, kleiner oder gleich der Referenzspannung des ADC sein, so lässt sich der Spannungsabfall am Messwiderstand mit überschaubarem Aufwand ermitteln. Ist die Versorgungsspannung der Last jedoch kleiner als die Referenzspannung des ADC, aber variabel, muss die Messspannung an beiden Seiten des Messwiderstandes herangezogen werden, wie in **Bild 13** zu sehen. Zugegeben, dieses Beispiel ist sehr idealisiert, denn die Versorgungsspannung der Last ist kleiner oder gleich der Referenzspannung des ADC. Ist die Versorgungsspannung nämlich größer als die Referenzspannung des ADC oder gar der MCU, so wird es schwieriger.

Überwachung einer Halogenlampe mit Strommessung

Nun soll die Stromaufnahme einer dimmbaren G4-Halogenlampe gemessen und überwacht werden, die mit einer maximalen Gleichspannung von 12 V versorgt wird. Das Leuchtmittel darf maximal 20 W „verbrauchen“ (1,67 A bei 12 V). Höhere Ströme als 2 A und geringere Ströme als 25 mA sollen von der MCU als fehlerhaftes Leuchtmittel klassifiziert werden.

Man könnte nun die Spannung vor und hinter dem Messwiderstand mit Spannungsteilern in einen Bereich bringen, der vom ADC verarbeitet werden kann. Das **Bild 14** zeigt eine entsprechend modifizierte Schaltung.

Die MCU wird mit 3,3 V versorgt und die Widerstände sind so gewählt, dass bei 12 V maximal 2,6 V an den Pins der ADC anliegen. Der Wert des Shunt-Widerstands beträgt hier 0,1 Ω. Bei 2 A sorgt er für einen Spannungsabfall von 0,2 V, gibt aber immer noch eine Verlustleistung von 0,4 W an Wärme ab. Das Verfahren funktioniert zwar prinzipiell, ist aber nicht besonders clever, wie wir gleich sehen werden.

Durch den Spannungsteiler wird der Messbereich noch weiter verkleinert, so dass der ADC der MCU bei 2 A einen maximalen Spannungsabfall von weniger als 0,05 V verarbeiten können muss. Auch die Tatsache, dass der Spannungsabfall durch zwei ADC-Kanäle gemessen wird, macht die weitere Verarbeitung nicht leichter.

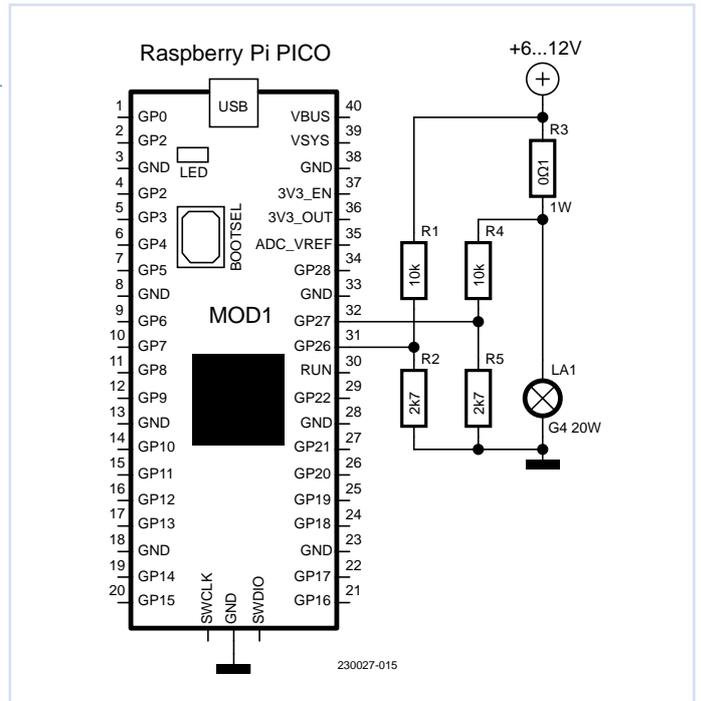


Bild 14. Modifizierte Schaltung mit Spannungsteilern.

In diesem Fall ist der Einsatz eines High-Side-Strommessverstärkers wie dem INA138 von Texas Instruments oder dem MCP6Co2 von Microchip die deutlich bessere Wahl. Hinzu kommt, dass in diesem Fall auch der Wert des Shunt-Widerstands erheblich kleiner werden darf. Der 10-mΩ-Shunt-Widerstand verursacht eine deutlich niedrigere Verlustwärme als sein zehn Mal größeres Pendant. **Bild 15** zeigt die Schaltung mit Strommessverstärker. Die Ausgangsspannung ist proportional zum Spannungsabfall über dem Widerstand und kann so leicht mit einem ADC-Kanal verarbeitet werden.

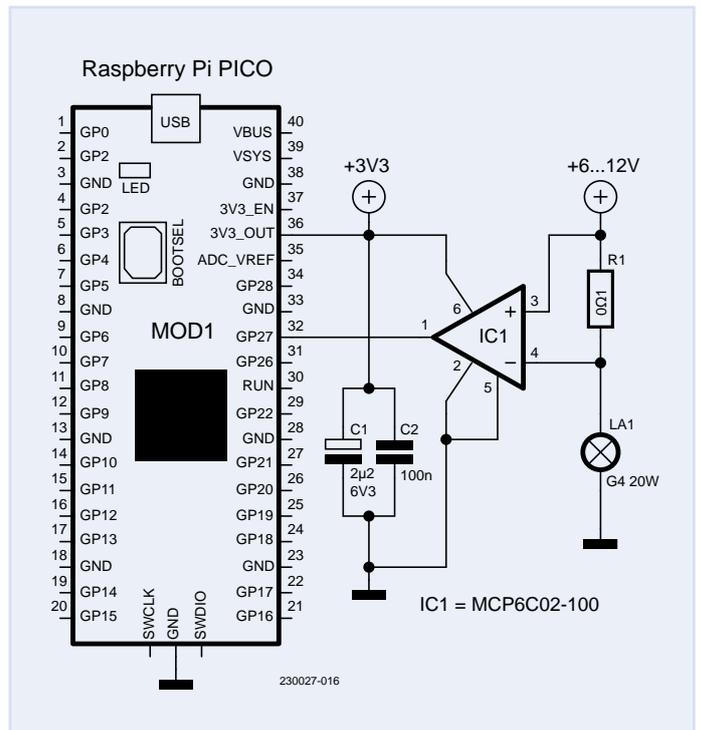


Bild 15. Schaltung mit Strommessverstärker.

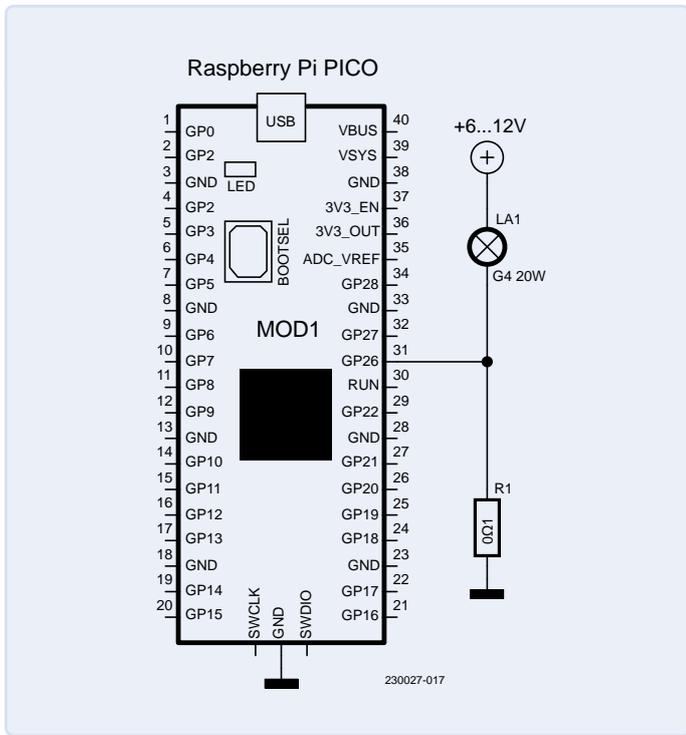


Bild 16. Low-Side Messung.

Low-Side-Messung

Anstatt einen Messwiderstand direkt vor den Verbraucher zu platzieren, kann dieser auch zwischen Verbraucher und Masse angeschlossen werden. Bleiben wir bei dem Beispiel der Halogenlampe. **Bild 16** zeigt eine solche Low-Side Messung. Hier wird ein 0,1-Ω-Widerstand verwendet, der bei 2 A Strom für einen Spannungsabfall von 0,2 V sorgt. Der Vorteil ist nun, dass die Potentiale der zu messenden Spannungen unter denen der Referenzspannung des ADC und der Versorgungsspannung der MCU liegen. Jedoch sind bei 25 mA Strom nur 2,5 mV zu messen (0,1 mV pro mA). Unter idealen Bedingungen würde der 12-Bit-ADC mit 3,3 V Referenzspannung immerhin 0,8 mV pro Bit auflösen. Jedoch sollte dabei ein Blick auf die effektive Anzahl der Bits (*effective number of bits*, ENOB) des ADCs geworfen werden, die die in der Praxis auftretenden Messfehler berücksichtigt, die aufgrund des Quantisierungsfehlers, von Verzerrungen, Rauschen und ähnlichem Ungemach auftreten. Für den RP2040 ist eine ENOB von etwa 9 spezifiziert, was so ungefähr 6,44 mA pro Bit ergibt. Für diesen Anwendungsfall wäre die Auflösung der Strommessung gerade noch ausreichend. Wenn jedoch eine filigranere Auflösung nötig ist, kann sich ein verstärkender Opamp als nützlich erweisen. Vor allem kann auch der Widerstandswert kleiner ausfallen.

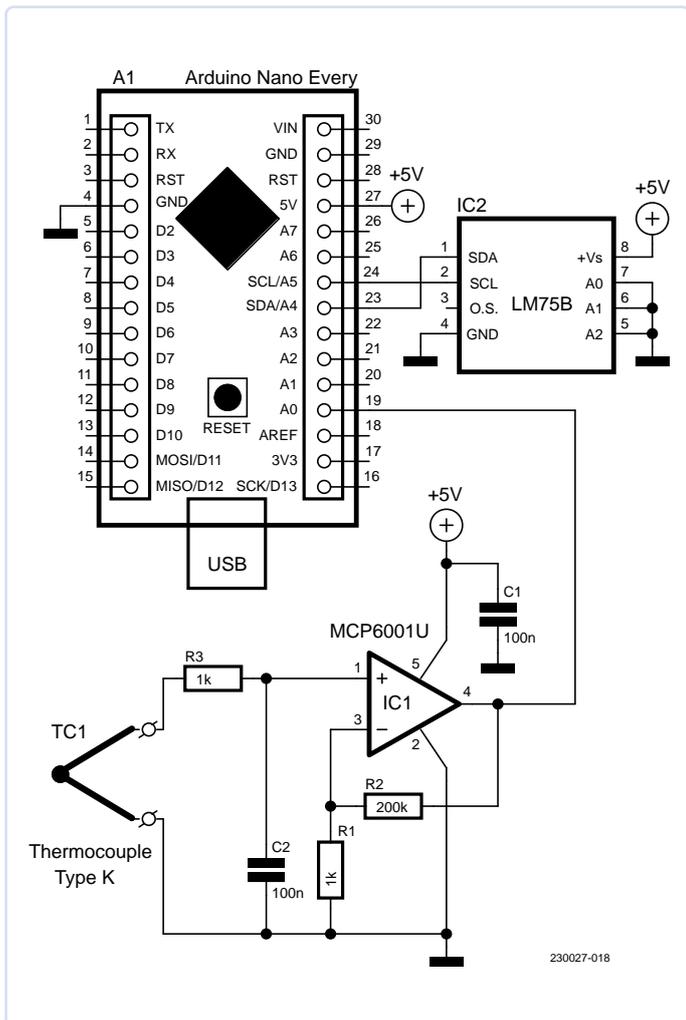


Bild 17. Auswertung eines Thermoelements mit dem Arduino Every Nano.

Operationsverstärker und Temperatursensoren

Über Operationsverstärker wurden schon Bibliotheken von Büchern geschrieben. Dieses so grundlegende und vielseitige Bauteil erlaubt bei passender Beschaltung eine Unzahl von Anwendungen, von denen schon ein paar in diesem Artikel beschrieben worden sind. Wenn es um das Messen von Spannungen geht, können es nicht nur Shunt-Widerstände sein, bei denen die Spannungen in einem nicht idealen Bereich für den ADC einer MCU liegen. Temperatursensoren sind auch solche Kandidaten, wenn es sich nicht um solche mit digitaler Schnittstelle wie einen DS18B20 oder einen DHT11 handelt, sondern um rein analoge resistive Typen wie einem PT100 oder um Thermoelemente, die aus zwei unterschiedlichen Metallen bestehen.

Thermoelemente

Diese aus zwei Metallen bestehenden Elemente verhalten sich als temperaturabhängige Spannungsquellen. Als Beispiel soll hier das weitverbreitete Typ-K-Element dienen. Bei 21 °C oder 69,8 °F gibt ein Element aufgrund des Seebeck-Effekts eine Spannung von 0,838 mV aus, bei 22 °C oder 71,6 °F sind es 0,879 mV. Die Differenz von nur etwa 40 μV pro Grad stellt den ADC einer MCU auf eine harte Probe. Hinzu kommt, dass bei dem Einsatz von Thermoelementen noch eine Kaltstellenkompensation (Cold Junction Compensation) [4] durchgeführt werden muss.

Mit einem Operationsverstärker lässt sich aber die Spannung eines Thermoelements verstärken. In **Bild 17** ist eine solche Schaltung beispielhaft für einen Arduino Nano Every realisiert, bei der eine Temperaturspanne von 0 °C bis 400 °C erwartet wird (0...16,396 mV Spannung vom Thermoelement).

Dabei wird der nicht invertierende Operationsverstärker mit einer Verstärkung von $A = 200$ betrieben. Um Störungen am Eingang des Operationsverstärkers zu minimieren, wurde ein RC-Filter hinzugefügt; um die Temperaturerfassung für die Kaltstellenkompensation kümmert sich ein einfacher digitaler Temperatursensor LM75.

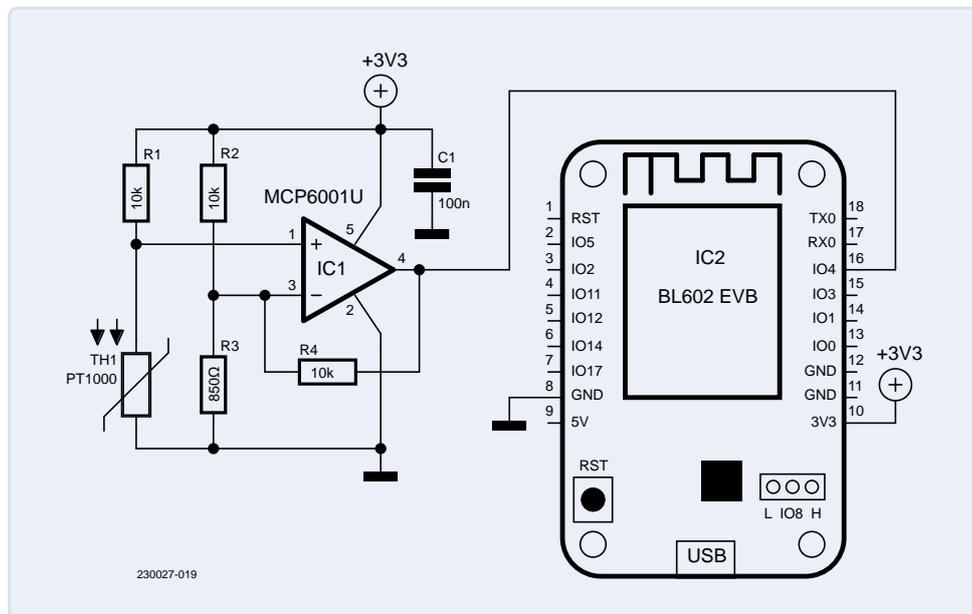


Bild 18. Messbrücke für einen PT1000.

PT100 und PT1000

PT100 und PT1000 sind temperaturabhängige Widerstände, die bei 0 °C einen Widerstand von 100 Ω (PT100) beziehungsweise 1000 Ω (PT1000) aufweisen. Hier muss zur Berechnung der Temperatur zunächst der Widerstandswert des Sensors gemessen werden. Eine gute Option, den Widerstandswert präzise zu ermitteln, stellt hier die Wheatstonesche Messbrücke dar. Wie auch bei den Thermo-Elementen sind die Spannungen auch bei PT100 und PT1000 ziemlich gering und es muss die geringe Differenz zwischen zwei Messpunkten ermittelt werden. Auch hier bietet ein Opamp, der als nicht invertierender Differenzverstärker beschaltet ist, die nötige Hilfestellung. Die Schaltung für ein Board mit 3,3 V Versorgungsspannung, zum Beispiel das Pinecone BL602 Evaluation Board, kann aus **Bild 18** entnommen werden.

Die Spannungsdifferenz zwischen dem positiven und negativen Eingang am Operationsverstärker wird verstärkt. Doch was ist, wenn die Temperatur kleiner als 0 °C wird? In diesem Fall müsste der Operationsverstärker eine negative Spannung ausgeben, die aber nicht zur Verfügung steht. Eine Option hier wäre, den 10-kΩ-Widerstand R3 durch einen 850-Ω-Widerstand zu ersetzen. Die Spannung zwischen R1 und R2 läge dann bei 0,3917 V, was bei -38 °C auch für den Punkt zwischen R3 und dem PT1000 gilt. Damit würde das Messfenster bei -38 °C entsprechend einer Spannungsdifferenz von 0 V beginnen. Bei 20 °C entstünde in der Schaltung eine Differenz von 0,09444 V.

Zusammenfassung

Die Verarbeitung und die Ausgabe analoger Werte lässt sich mit einem Mikrocontroller und überschaubarem zusätzlichen Aufwand bewerkstelligen. Auch wenn das Ganze nicht so simpel erscheint wie das Steuern eines einfachen digitalen I/O-Pins, ist der Umgang mit dem ADC etwas, das man zumindest einmal versucht haben sollte. Die hier gezeigten Beispiele sollen Anregungen für eigene Versuche sein, den Mikrocontroller mit der analogen Welt außerhalb seines Gehäuses interagieren zu lassen. ◀

RG – 230027-02

Sie haben Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an Elektor unter redaktion@elektor.de.



Passende Produkte

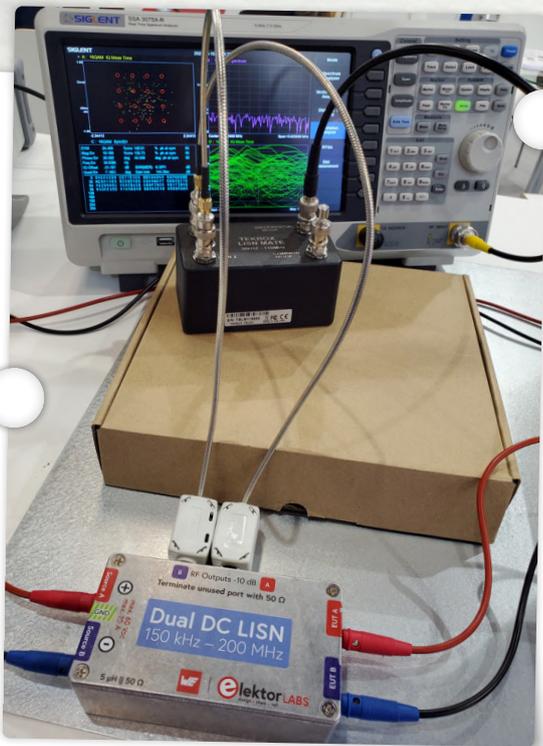
- > **Arduino Nano**
SKU 17002: www.elektor.de/17002
- > **Raspberry Pi Pico RP2040**
SKU 19562: www.elektor.de/19562
- > **Pinecone BL602 Evaluation Board**
SKU 19914: www.elektor.de/19914

WEBLINKS

- [1] Guido Nopper, „Einfacher A/D-Umsetzer mit PLD“, Elektor 9-10/2019:
<https://www.elektormagazine.de/magazine/elektor-109/51114>
- [2] Hardware design with RP2040: <https://datasheets.raspberrypi.com/rp2040/hardware-design-with-rp2040.pdf>
- [3] Giovanni Carrera, „Isolierter Analogausgang für Arduino Uno“, Elektor 11-12/2022:
<https://www.elektormagazine.de/magazine/elektor-281/61084>
- [4] Kaltstellenkompensation (Cold Junction Compensation) (Wikipedia):
https://de.wikipedia.org/wiki/Thermoelement#Einbeziehung_der_Vergleichsstellentemperatur

embedded world 2023

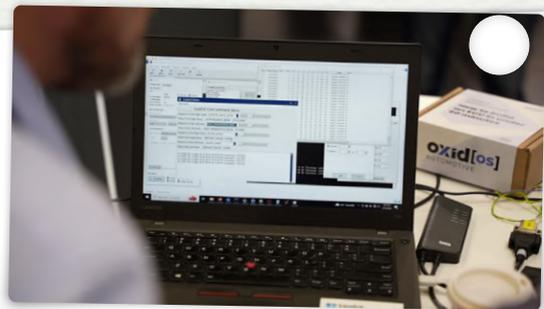
Die embedded world 2023 in Nürnberg - ein Muss für alle, die sich mit Mikrocontrollern und Tools beschäftigen - fand wieder zum gewohnten Zeitpunkt im Frühjahr statt. In diesem Jahr wurden 27.000 Besucher gezählt, was einer Steigerung von 50 % gegenüber der Messe 2022 entspricht, und es gab mehr als 950 Aussteller in sechs Hallen. Elektor-Redakteur Stuart Cording, Content Director C.J. Abate und Chefredakteur Jens Nickel hatten die Gelegenheit, sich umzuschauen und viel Interessantes zu finden, aber ihre persönliche Auswahl kann natürlich nur ein sehr kleiner Ausschnitt dessen sein, was sich auf dem Markt abspielt. Noch mehr finden Sie auf unserem YouTube-Kanal „Elektor TV – Industry“ unter <https://youtube.com/@ElektorIndustry>.



Siglent

Die Dual-DC-Netznachbildung (Line Impedance Stabilization Network, LISN) von Elektor wurde am Stand von Siglent in Betrieb genommen. Die duale 5- μ H-50- Ω -Netznachbildung unterstützt Spannungen bis zu 60 V und Ströme bis zu 10 A. Sie ermittelt HF-Störungen auf beiden Kanälen über 5- μ H-Sperrinduktivitäten. Das interne 10-dB-Dämpfungsnetzwerk (eines in jedem Kanal) enthält ein Hochpassfilter dritter Ordnung mit einer Grenzfrequenz von 9 kHz, um empfindliche Eingänge von Messgeräten wie einen Spektrumanalysator vor potenziell schädlichen Gleichspannungen oder niedrigen Frequenzen von den zu prüfenden Geräten zu schützen.

Sehen Sie sich die Demo an unter:
www.elektormagazine.com/ew-lisn



oxid[OS]

Die Programmiersprache C ist seit Jahrzehnten die Hauptstütze der Automobilindustrie. Aber wie lange noch? Wir sprachen mit Flavia Oprea, um zu erfahren, warum die aufstrebende Sprache Rust eine Möglichkeit darstellt, die traditionellen Fehler der eingebetteten C-Programmierung zu vermeiden. Das in Rumänien ansässige Startup-Unternehmen hat ein auf Rust basierendes Echtzeitbetriebssystem (RTOS) entwickelt, das auf der Messe auf einem STM32-Mikrocontroller vorgeführt wurde. Entwickler in der Automobilbranche zeigen bereits Interesse an diesem System, da es mehr Sicherheit verspricht und isolierten Legacy-C/C++-Code unterstützt.

<https://oxidos.io>

Sehen Sie sich das Interview an:
https://youtu.be/BNdvgJdN5_Q?t=1398

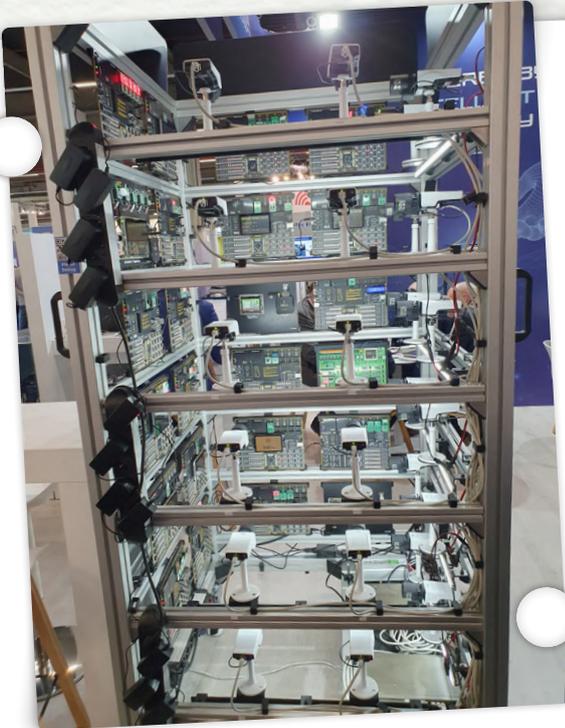


Axelera

Die niederländische Firma Axelera bietet Module an, die KI-Inferenzanwendungen wie Bilderkennung direkt at the Edge verarbeiten können. Die Boards gibt es in verschiedenen Formfaktoren wie M.2 (Bild) oder PCIe. Betrieben von einer einzelnen Metis-AIPU liefert das M.2-Modul bis zu 214 Tera-Operationen pro Sekunde (TOPs), während der Stromverbrauch minimiert und die Integration durch den herunterladbaren Software-Stack vereinfacht ist. Der Preis beträgt 149 USD/EUR.

<https://axelera.ai/ai-product-solutions>





Mikroelektronika

Auf der Messe zeigte Mikro einen mit einer Kamera ausgestatteten Rahmen für ihr Planet Debug System: Entwickler und Studenten können gemeinsam ein Mikrocontroller-Projekt aus der Ferne entwickeln und debuggen, ohne wirklich die gleichen Boards auf dem Schreibtisch haben zu müssen. Die Benutzer können nun ihre Hardware nach ihren eigenen Bedürfnissen konfigurieren, indem sie Motherboards, Prozessorboards und eine große Anzahl von Peripheriegeräten und Displays auswählen.

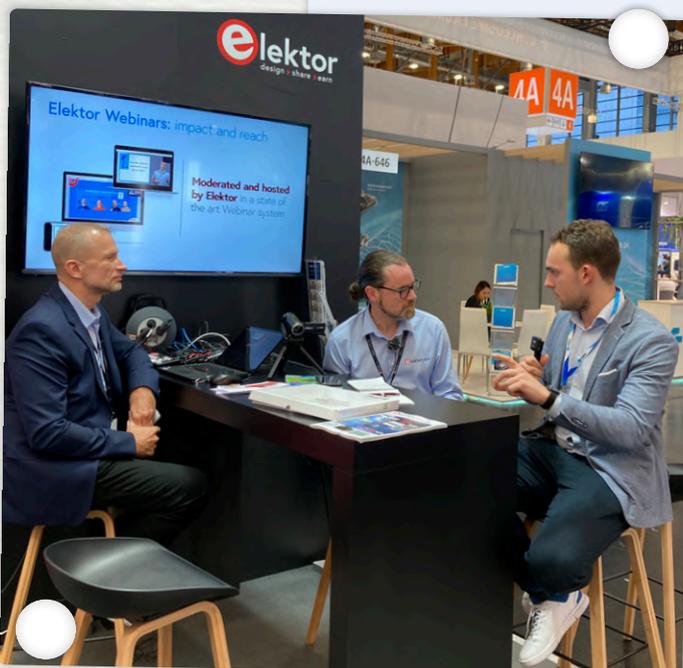
<https://mikroe.com/planet-debug>



Espressif

Der ESP32 ist bekannt für die Benutzerfreundlichkeit bei der Implementierung von drahtlosen Anwendungen. Deshalb waren ESP32 und andere Espressif-Lösungen auf der embedded world 2023 in Dutzenden von interessanten Demos zu sehen, zum Beispiel einem ESP-ZeroCode-Modul für intelligente Jalousien, ESP-RainMaker-Demos und Matter-WLAN-Geräte. Elektor sprach auf der embedded world mit Amey Inamdar von Espressif, um zu erfahren, was es Neues gibt. Zu den Highlights in der Pipeline gehören: RISC-V-SoCs sowie eine Dual-Core-Version, Unterstützung des Hausautomatisierungsstandards Matter, um die Bereitstellung von IoT-Geräten mit unterschiedlichen Protokollen zu vereinfachen; und die RainMaker-Plattform für die Implementierung des Cloud-Backends von IoT.

Sehen Sie sich das Interview von Stuart Cording an!
<https://youtu.be/EFnUtAJX2aA>

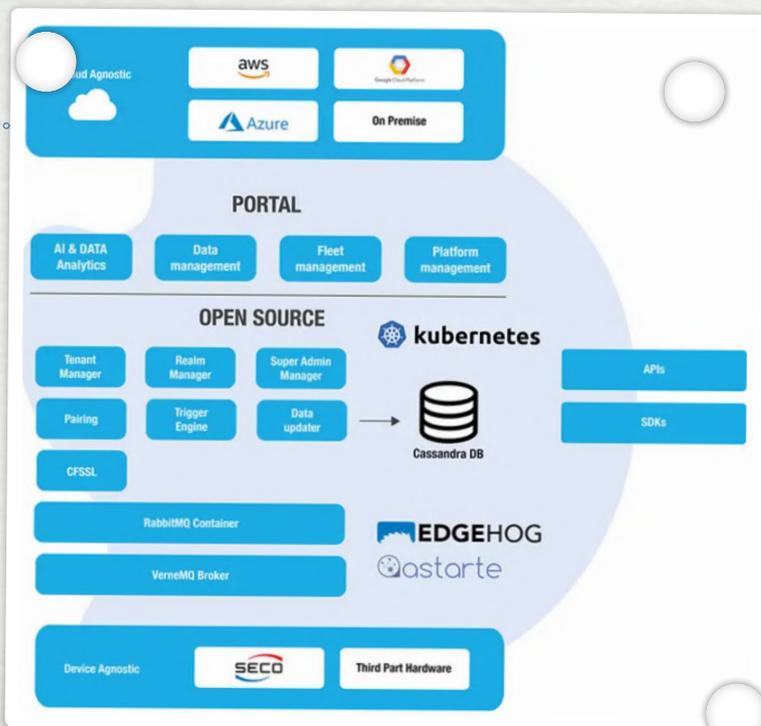


Live-Show-Splitter

Zum zweiten Mal haben sich Stuart Cording von Elektor Engineering Insights und Jens Nickel von Elektor Lab Talk zusammengetan, um live von einer Elektronikmesse zu berichten. In der einstündigen Sendung fachsimpeln Stuart und Jens mit Elektor-Content-Direktor C.J. Abate über die interessantesten Dinge, die auf der embedded world 2023 zu sehen sind (zum Beispiel ein bierzapfender, cloud-gesteuerter Roboter). Besonderer Gast: Elektor-Autor Viacheslav Gromov, der mit seiner Firma AITAD den „embedded award“ in der Kategorie KI gewonnen hat. ◀

elektor TV
Sehen Sie sich das vollständige Video hier an: www.youtube.com/watch?v=00d51VKZx7Q





Seco

Der italienische IPC- und Solution-Anbieter Seco präsentiert die modulare IoT-Plattform CLEA. Sie kann „at the Edge“ Ereignisdaten der Hardware selbst und der Kundenanwendungen überwachen. Dazu bietet CLEA offene APIs und Protokollkonverter sowie eine KI-basierte Edge-Logik zur Analyse von Rohdaten und Bildern. CLEA ist hardware- und cloud-übergreifend und basiert auf einer Open-Source-Basis, was eine langfristige Verfügbarkeit gewährleistet. Zu den bereits realisierten Anwendungen gehören intelligente Kaffeeautomaten, Ladestationen und medizinische Geräte.

<https://north.seco.com/en/products-and-services/clea-iot-platform>



Slint

Slint ist ein Toolkit zur effizienten Entwicklung grafischer Benutzeroberflächen für eingebettete und Desktop-Displays. Es werden mehrere Programmiersprachen unterstützt, Rust, C++ und JavaScript. Nach Angaben der Entwickler benötigt die Lösung nur ein paar hundert Kilobyte RAM und wenig Rechenleistung. Es gibt kommerzielle Lizenzen, aber auch eine Open-Source-Lizenz unter der GPLv3.

<https://slint-ui.com/#tryout>



Arm

Wie wir in unserer Elektor Engineering Insights Show gesehen haben, verändert sich die Welt des Automobils dramatisch. Heute wird die Software eines Steuergeräts nach ihrer Fertigstellung nicht mehr aktualisiert - es sei denn, es wird ein sicherheitsrelevantes Problem gefunden. Doch das ändert sich, wie Robert Day anhand der neuen SOAFEE-Initiative von Arm erläuterte. In Anlehnung an die Orchestrierung und die Verwendung von Containern in Cloud-Software und CI/CD-Entwicklungsprozessen werden solche Methoden für sicherheitskritische Automobilsoftware angepasst. Derzeit ist eine Reihe von Halbleiterherstellern, Automobil-OEMs und deren Zulieferer beteiligt.

Sehen Sie sich das Video-Interview an!
https://youtu.be/nr_6W4UgihM



Digi

Das 868-Modul aus der Familie Digi XBee XR ist eine kompakte und zuverlässige Lösung für den Einsatz von Verbindungsanwendungen mit großer Reichweite in der europäischen Region. Das vorzertifizierte Modul arbeitet zwischen 863 MHz und 870 MHz und unterstützt sowohl Punkt-zu-Punkt- als auch Mesh-Netzwerkprotokolle mit einer Sichtweite von über 14 Kilometern. Natürlich ist auch ein Dev-Kit erhältlich (Bild).

<https://digi.com/products/embedded-systems/digi-xbee/rf-modules/sub-1-ghz-rf-modules/digi-xbee-xr-868>

Treten Sie jetzt der Elektor Community bei!



Jetzt  Mitglied werden!



- ✓ Komplettes Webarchiv ab 1970
- ✓ 8x Elektor Doppelheft (Print)
- ✓ 8x Digital (PDF)
- ✓ 10% Rabatt im Online-Shop und exklusive Angebote
- ✓ Zugriff auf über 5.000 Gerber Dateien aus Elektor Labs



Auch erhältlich

Die digitale
Mitgliedschaft!



- ✓ Zugang zu unserem Webarchiv
- ✓ 10% Rabatt in unserem Online-Shop
- ✓ 8x Elektor Doppelheft (PDF)
- ✓ Zugriff auf über 5.000 Gerber Dateien aus Elektor Labs



www.elektormagazine.de/Abonnement

Sub-Nyquist-Sampling in der Praxis

Höhere Frequenzen mittels Unterabtastung sicher erfassen

Von Sebastian Westerhold (AI5GW) (Deutschland)

Wer sich mit abtastenden Systemen beschäftigt, wird sehr schnell die Grenzen des Nyquist-Shannon-Abtasttheorems kennenlernen. Hiernach sind Signale mit höherer Frequenz als die Hälfte der Abtastfrequenz nicht mehr zuverlässig erfassbar. Es handelt sich dabei allerdings nur um die halbe Wahrheit. Eine zuverlässige Methode zur Erfassung höherfrequenter Signale ist die sogenannte Unterabtastung, auch Sub-Nyquist-Sampling genannt. Dieser Artikel stellt die Methode praxisnah vor, wobei ein einfacher Arduino Uno verwendet wird.

Problematik

Das Nyquist-Shannon-Abtasttheorem ist ein äußerst wichtiges Theorem, wenn man sich mit abtastenden Systemen auseinandersetzt. Die stark verkürzte Variante des Theorems besagt, dass man als Abtastfrequenz mindestens das Doppelte der höchsten im abzutastenden Signal vorkommenden Frequenz benötigt. So oder so ähnlich hat man es sicher schon mal gehört oder gelesen.

Was passiert nun, wenn man ein rein sinusförmiges Signal mit einer Frequenz von 120 kHz mit einer Abtastrate von nur 100 kS/s abtastet? Dann entsteht ein sogenannter Alias bei 20 kHz. Die erfassten Abtastwerte würden also so aussehen, als hätte man ein sinusförmiges Signal mit einer Frequenz von 20 kHz (120 kHz - 100 kS/s) abgetastet. Und Aliasing-Fehler soll man ja auf jeden Fall vermeiden. Oder doch nicht?

Ob Aliasing-Fehler ein Störfaktor sind oder als nützliche Technik eingesetzt werden können, hängt von der konkreten Anwendung ab. Im Beispiel mit der Abtastrate von 100 kS/s wäre es unmöglich, zwischen einem 20-kHz-Signal und einem 120-kHz-Signal zu unterscheiden; beide Signale würden wie ein 20-kHz-Signal aussehen. Ein 80-kHz-Signal würde im Übrigen ebenso einen Alias bei 20 kHz (100 kS/s - 80 kHz) erzeugen (**Bild 1**). Kritisch ist dieser Umstand jedoch nur dann, wenn eine Unterscheidung zwischen den genannten Signalfrequenzen überhaupt notwendig ist. Fallen alle tatsächlich im Signal vorkommenden Signale in einen eindeutig identifizierbaren Bereich, ist die Abtastung auch dann entsprechend aussagekräftig, wenn absichtlich Alias-Effekte ausgenutzt werden. Die Eindeutigkeit ist dann gegeben, wenn alle im abzutastenden Signal vorkommenden Signale in dieselbe Nyquist-Zone fallen.

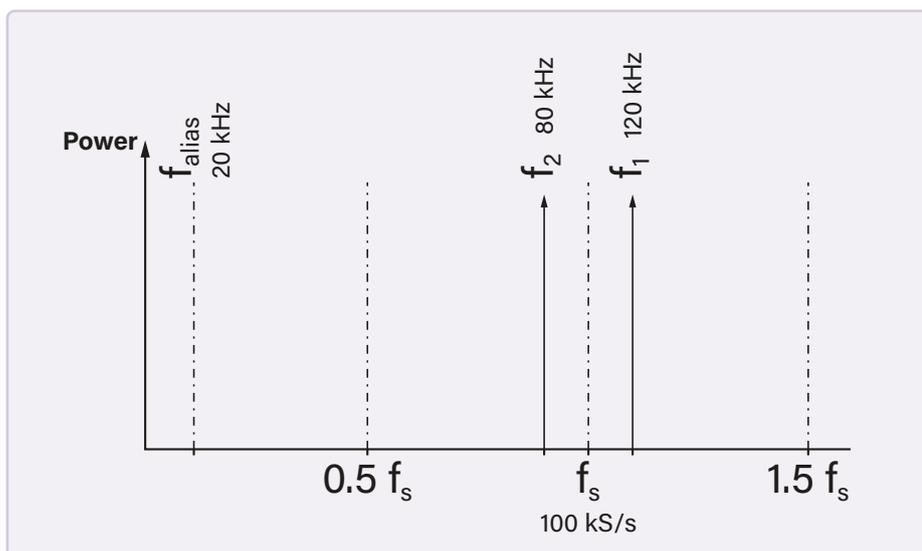


Bild 1. Bei einer Abtastrate von 100 kS/s erzeugen sowohl ein Signal bei 120 kHz (f_1) als auch ein Signal bei 80 kHz (f_2) einen Alias (f_{ALIAS}) bei 20 kHz.

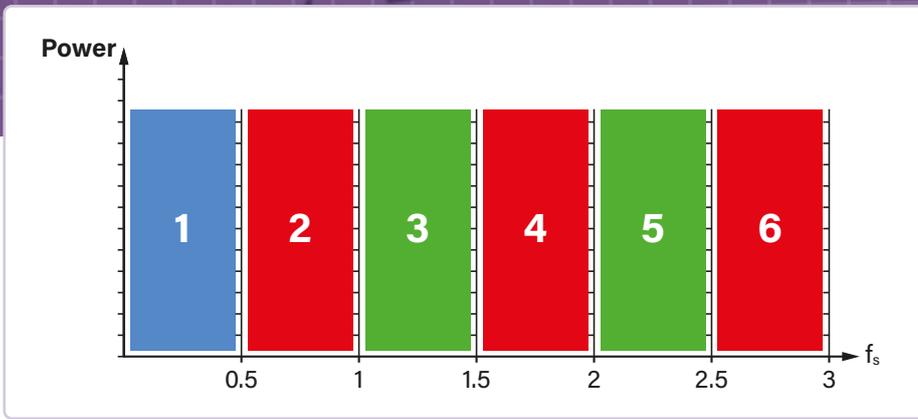


Bild 2. Nyquist-Zonen 1 - 6. Zone 1 = „normaler Bereich“, Zonen 2, 4, 6 = invertierende Bereiche, Zonen 3 und 5 = nicht-invertierende Zonen.

Den Frequenzbereich von 0 Hz (DC) bis zur Hälfte der Abtastfrequenz (f_s) bezeichnet man als Nyquist-Zone 1 (**Bild 2**). Diese Zone ist der „normale“ Arbeitsbereich für ein abtastendes System. Nyquist-Zone 2 erstreckt sich von $0,5 f_s$ bis f_s . Diese Zonen setzen sich dann im Abstand von $0,5 f_s$ theoretisch unendlich fort. Die Nyquist-Zonen mit geraden Zahlen haben eine besondere Eigenschaft: Innerhalb dieser Bereiche findet eine Invertierung des Frequenzspektrums statt.

Wenn man also die Bandbreite eines zu erfassenden Signals auf den Bereich einer dieser Nyquist-Zonen begrenzt, lassen sich mittels Unterabtastung / Sub-Nyquist-Sampling auch Signale erfassen, deren Frequenzbestandteile deutlich über die Abtastrate hinausgehen.

Praxisbeispiel

Für ein Projekt soll die Frequenz eines Signals im Bereich von rund 160 kHz bis 165 kHz zuverlässig mit einem Arduino Uno erfasst werden. Aus taktischen Gründen wurde die Abtastrate durch Setzen der entsprechenden Register (Prescaler 1:8) auf ungefähr 154 kS/s eingestellt (genaugenommen sind es 153.846 S/s, der Übersicht halber wird in

diesem Artikel die gerundete Zahl verwendet). Die Signalfrequenz ist also um einige Kilohertz höher als die Abtastrate selbst, aber kleiner als $1,5 f_s$ (entsprechend Nyquist-Zone 3). Die durch die Unterabtastung zu erwartenden Aliase liegen dementsprechend zwischen 6 kHz ($160 \text{ kHz} - f_s$) und 11 kHz ($165 \text{ kHz} - f_s$). Der Arduino-Code ist so aufgebaut, dass der A/D-Wandler des Arduino mittels einer Interrupt Service Routine (ISR) solange das Signal am Analogeingang 0 abtastet, bis ein Puffer mit 512 diskreten Abtastwerten gefüllt ist. Aus diesen Werten wird dann die (Alias-) Signalfrequenz berechnet, bevor die nächsten 512 Samples abgetastet werden und das Spiel von vorne losgeht.

Zur Frequenzbestimmung wurde der sogenannte Goertzel-Algorithmus verwendet, eine ressourcenschonende Sonderform der diskreten Fourier-Transformation [1]. Das Ergebnis wird dann zu Demonstrationszwecken über die serielle Schnittstelle ausgegeben, optisch aufbereitet für den seriellen Monitor (**Bild 4**).

Nach dem gleichen Prinzip könnte man beispielsweise eine 455-kHz-ZF aus einem Funkempfänger in Nyquist-Zone 6 ($3 f_s$,

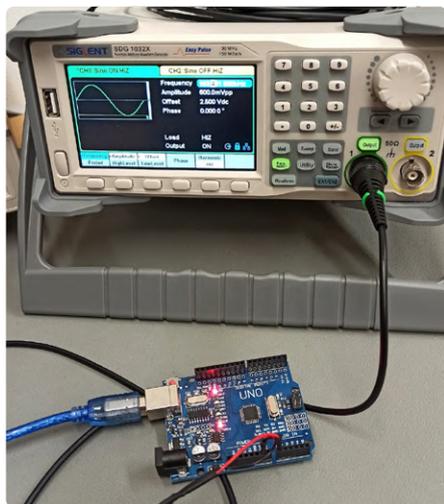


Bild 3. Arduino/Signalgenerator Setup.

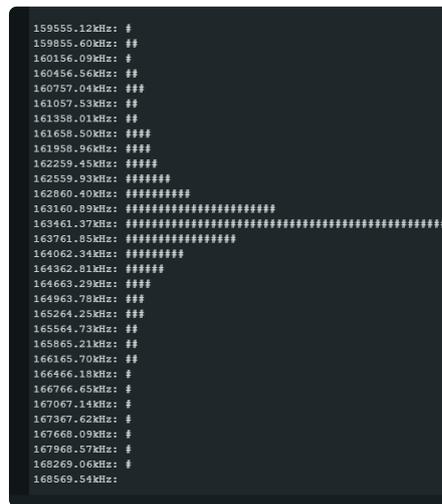


Bild 4. Ausgabe auf dem Seriellen Monitor.

entspricht circa 462 kHz) mittels Unterabtastung erfassen und FSK-Signale (NAVTEXT, RTTY, und so weiter) sogar mit dem relativ leistungsschwachen Arduino Uno demodulieren.

Anmerkungen und Hinweise

Damit die Unterabtastung wie gewünscht funktioniert, muss der verwendete ADC die notwendige Analogbandbreite aufweisen. Diese ist typischerweise um einiges höher als die Abtastrate. Im Datenblatt findet man diese Kennzahl typischerweise als „Full power bandwidth“. Beim ATmega328P scheint die Unterabtastung laut meiner eigenen Experimente sehr sauber bis in den Megahertz-Bereich zu funktionieren. Eigene Versuche mit dem Raspberry Pi Pico erzielten leider keine vielversprechenden Ergebnisse. ◀

220629-02

Sie haben Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an den Autor unter sebastian@baltic-lab.com oder an Elektor unter redaktion@elektor.de.



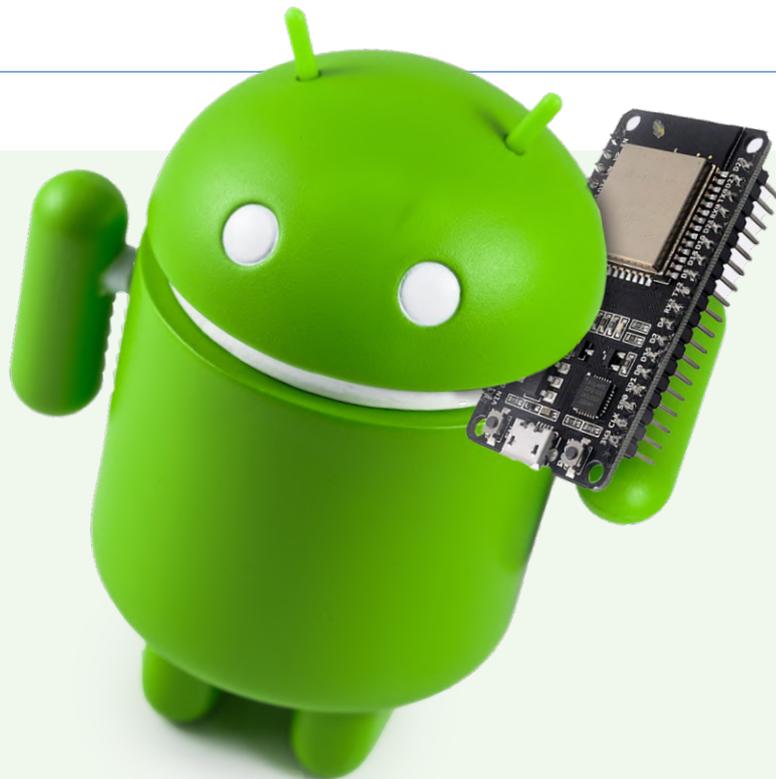
Passende Produkte

- > **Siglent SDG1032X 2-Kanal-Signalgenerator (30 MHz)**
SKU 20276: www.elektor.de/20276
- > **OWON XSA810 Spectrum Analyser (1 GHz)**
SKU 19714: www.elektor.de/19714
- > **HackRF One Software Defined Radio (1 MHz bis 6 GHz)**
SKU 18306: www.elektor.de/18306
- > **MonoDAQ-U-X Multifunktions-USB-Datenerfassungssystem**
SKU 18766: www.elektor.de/18766

WEBLINK

- [1] Sebastian Westerhold (2022): Goertzel-Bibliothek für Arduino: <https://github.com/AI5GW/Goertzel>

Hier Android- Smartphone, dort ESP32?



Praxisprojekt mit der
Android WiFi-API

Von Tam Hanna (Ungarn)

Für Android-Programmierer ist der Code für eine Verbindungsaufnahme zu einem WLAN nicht trivial, besonders wenn man den Usern etwas Komfort bieten möchte. Entwickler haben mit der Verschiedenheit der Android-Versionen genauso zu kämpfen wie mit Restriktionen aus dem Hause Google. Der Artikel soll helfen, einen Weg durch diesen Dschungel zu bahnen.

WLAN ist - unter Vernachlässigung des Aspekts des höheren Stromverbrauchs - insbesondere für kleinere Projekte der Funkstandard der Wahl: Die von der Bluetooth SIG erhobenen Lizenzgebühren fallen unter den Tisch, die Kommunikation erfolgt in den meisten Fällen per REST, außerdem gibt es WLAN-Transmitter so gut wie überall.

Schon aus diesem Grund ist es nur logisch, dass die in den diversen ESP32-Mikrocontrollern verbauten WLAN-Accesspoints seit jeher auch Begehrlichkeiten bei Android programmierenden Software-Nerds erwecken.

Während Android für Echtzeit-Aktivitäten nicht wirklich geeignet ist, ist das Betriebssystem mit umfangreichen GUI-Stacks ausgestattet, sogar für anspruchsvolle Aufgaben wie die Anzeige von Diagrammen gibt es per Gradle ladbare Bibliotheken.

Im Fall von Systemen mit Echtzeit- und UI-Ansprüchen bietet es sich an, einen verteilten Prozessrechner zu realisieren, wie in **Bild 1** gezeigt.

Störellement ist - wie so oft - der User, der am Ende mit dem System Werte schaffen muss. Weist man einen Benutzer dazu an, „von Hand“ Kontakt zu einem bestimmten WLAN aufzunehmen, so handelt man sich Unmengen von Kundensupport-Anfragen und frustriertes Gemecker ein.

Ein schönerer Weg wäre der automatische Verbindungsaufbau - problematisch nur, dass Google in diesem Bereich aus Datenschutz- und sonstigen Gründen Schindluder treibt.

Worum geht es hier?

Ziel des folgenden Artikels ist die Beschreibung des unter Android zum Verbindungsaufbau mit Ziel-WLANs erforderlichen Codes. Als Gegenstelle dient hier ein ESP32, der per SoftAP ein WLAN aufbaut - in der Praxis lässt sich dieser Code natürlich auch mit anderen WLAN-fähigen Controllern verwenden (Download unter [1]).

Der folgende Artikel basiert auf einem Projekt, das der Autor für einen Kunden entwickelt hat. Zum Nachvollziehen ist mehr oder weniger jedes ESP32-Board geeignet, das eine als Accesspoint konfigurierte Variante des Beispielprogramms *esp-idf/examples/wifi/getting_started/* ausführt (in der ESP-IDF [2] enthalten). Die Arbeit mit dem ESP32 setzt der Autor dabei als komplett bekannt voraus.

Auf Androidseite sollten Sie zwei physikalische Telefone besitzen: eines mit Android ≥ 10 , eines mit Android 9 oder älter. Als Entwicklungsumgebung dient Android Studio [3]; Grundkenntnisse der Arbeit mit Android müssen wir aus Platzgründen ebenfalls als gegeben annehmen.



Bild 1. Das Android-Smartphone kümmert sich um das GUI, während der ESP32 für die Hardware-Kommunikation zuständig ist.

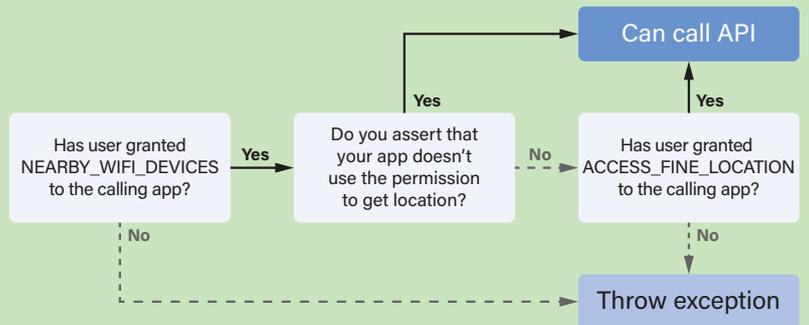


Bild 2. Für die Drahtlos-Kommunikation unter Android 13 sind mitunter zusätzliche Permissions erforderlich. (Bildquelle: [4]).

Konfiguration

Die Kompilations-Konfiguration eines Android-Projekts ist dabei ein durchaus komplizierter Vorgang. Normalerweise findet man eine zum Modul *app* gehörende Version der Datei *build.gradle*, in der sich der folgende stark gekürzte Code findet:

```
android {
    compileSdkVersion 29
    buildToolsVersion "30.0.1"
    defaultConfig {
        minSdkVersion 26
        targetSdkVersion 31
    }
}
```

Über den Wert *minSdkVersion* darf der Entwickler festlegen, welche Mindestversion auf dem Ziel- beziehungsweise Endgerät erforderlich sein muss, um die Applikation auszuführen. *compileSdkVersion* und *buildToolsVersion* zurren die Toolchain fest; dann gibt es das Attribut *targetSdkVersion*, das festlegt, welche Verhaltens-Variante der Entwickler bei der Erzeugung der Applikation im Hinterkopf hatte. Diese auf den ersten Blick kompliziert und verwirrend klingende Unterscheidung ist insofern erforderlich, als sich das „Verhalten“ vieler Android-APIs zwischen verschiedenen Versionen ändert. Ist *targetSdkVersion* auf einen gewissen Wert eingestellt, so geht das Betriebssystem davon aus, dass der Entwickler die in dieser Version eingeführten Änderungen beim Design seiner Applikation berücksichtigt hat - ist der Wert kleiner, so wird der Kompatibilitäts-Modus aktiviert. Leider macht Google im Playstore strenge Richtlinien in Bezug auf die erlaubten Werte von *targetSdkVersion* geltend. Ist eine Applikation zu alt, so weist Android Studio schon während der Kompilation durch eine nach dem Schema *Google Play requires that apps target API level 30 or higher* aufgebaute Meldung darauf hin, dass das Playstore-Backend das Hochladen der generierten APK verweigern wird. In nicht allzu ferner Zukunft wird Google „sehr alte“ Applikationen außerdem komplett aus dem Play Store entfernen.

Daraus folgt, dass man als Entwickler den eingeführten Änderungen schutzlos ausgeliefert ist und diese zeitnah nachvollziehen muss.

Permissions

Man kann die Welt von Android grob in die Zeit „vor“ und „nach“ dem neuem Permission-Managementsystem und/oder Storage Access Framework einteilen: In der guten alten Zeit konnten Entwickler im

Allgemeinen alles tun, während Google später immer mehr Nanny-state-Maßnahmen einführte.

Im Fall der WLAN-Konfiguration ist die Lage haarig, weil Informationen über in der Umgebung befindliche Drahtlosnetzwerke - denken Sie beispielsweise an Apples ersten iPod Touch - zur Ermittlung des Standorts genutzt werden können.

Es ist erforderlich, in der Manifestdatei einer WLAN-Applikation die folgenden Permission-Deklarationen festzulegen:

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Möchten Sie in ihrer Applikation die Version Android 13 unterstützen, so müssen Sie eine zusätzliche Modifikation durchführen. Die Permission-Logik von Android 13 funktioniert im Fall von WLAN nämlich wie in **Bild 2** gezeigt.

Bei der eigentlichen Deklaration der neuen Permission gibt es dabei Sonderfälle, die nach folgendem Schema aufgebaut sind:

```
<uses-permission android:name="android.permission.NEARBY_WIFI_DEVICES"
android:usesPermissionFlags="neverForLocation" />
```

Die *android.permission.NEARBY_WIFI_DEVICES*-Permission lässt sich in Theorie und Praxis beim Übergeben des Parameters *android:usesPermissionFlags="neverForLocation"* ein Sonderregime nutzen, in dem der Entwickler erklärt, die WLAN-Informationen ausschließlich zur Gerätekonfiguration und keinesfalls zur Standortermittlung zu benutzen. In diesem Fall erspart man sich unter Android 13 die Permission *android.permission.ACCESS_FINE_LOCATION*.

Da „ältere“ Versionen des Betriebssystems diese Permission voraussetzen, bietet sich in diesem Fall eine Konfiguration nach folgendem Schema an:

```
<uses-permission android:name=
"android.permission.ACCESS_FINE_LOCATION"
android:maxSdkVersion="32" />
```

Das Attribut `maxSdkVersion` sorgt in diesem Zusammenhang dafür, dass die Permission-Deklaration nur von älteren Android-Versionen erkannt wird.

Zweigleisige Entwicklung

Im Rahmen der Einrichtung der *Activity*, die für das „Scannen“ der Umgebung nach verwendbaren WLAN-Endstellen zuständig ist, müssen wir feststellen, ob der WLAN-Transmitter des Telefons aktiv ist:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    if (android.os.Build.VERSION.SDK_INT <
        Build.VERSION_CODES.Q) {
        wifiManager =
            (WifiManager) getApplicationContext()
                .getSystemService(Context.WIFI_SERVICE);
        if (!wifiManager.isWifiEnabled()) {
            wifiManager.setWifiEnabled(true);
        }
    }
}
```

Sofern wir es mit einer „alten“, also von der API-Änderung nicht tangierten Android-Variante zu tun haben, kann unser Programm den WLAN-Transmitter des Telefons aktivieren, ohne dabei eine Zustimmung vom Benutzer einholen zu müssen.

Leider gilt dies nicht für aktuellere Versionen von Android - in ihnen ist eine Zustimmung des Benutzers erforderlich.

Um beide Varianten zur Laufzeit auseinander zu halten, setzen wir auf die Konstante `android.os.Build.VERSION.SDK_INT`. Sie verweist auf die API-Version, die von der Firmware des vorliegenden Telefon unterstützt wird.

Das eigentliche Einschalten erfolgt durch Absenden eines *Intents* vom Typ `Settings.Panel.ACTION_WIFI`:

```
else {
    wifiManager =
        (WifiManager) getApplicationContext().
            getSystemService(Context.WIFI_SERVICE);
    if (!wifiManager.isWifiEnabled()) {
        Toast toast =
            Toast.makeText(getApplicationContext(),
                "PLEASE enable WiFi so that we can connect!",
                Toast.LENGTH_LONG);
        toast.show();
    }
}
```

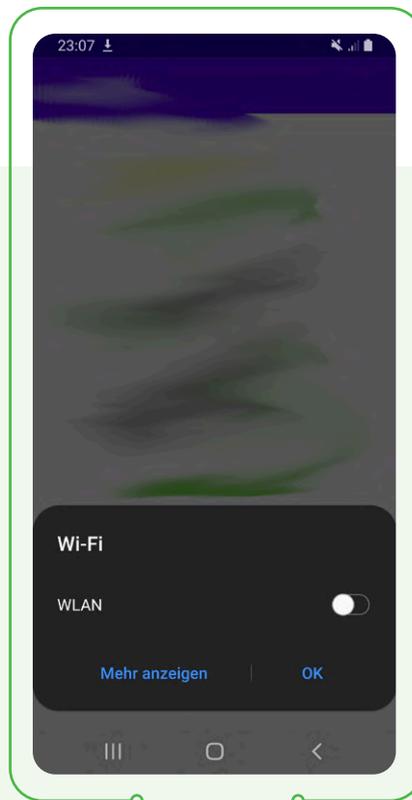


Bild 3. Google zwingt den Benutzer zur expliziten Zustimmung.

```
Intent panelIntent =
    new Intent(Settings.Panel.ACTION_WIFI);
this.startActivityForResult(panelIntent,
    ICY_WIFI_REQCODE);
}
```

Lohn der Mühen ist, dass zur Laufzeit die in **Bild 3** gezeigte Meldung aufscheint. Der Benutzer muss dort auf das Toggle klicken, um den Aktivierungsprozess zu autorisieren.

Unser hier abgedrucktes Snippet feuert vor dem Absenden des *Intents* einen *Toast* in Richtung des Benutzers, die über die Notwendigkeit zur Aktivierung des WLAN-Transmitters informiert.

Sofern Sie es mit technisch sehr herausgeforderten Benutzern zu tun haben, ist es empfehlenswert, „härtere Maßnahmen“ zu ergreifen - neben einer expliziten *MainActivity* nur für den Verbindungsaufbau kann es auch empfehlenswert sein, eine Schleife zu realisieren, die den Benutzer immer wieder mit der *Activity* konfrontiert.

Für Android-Quereinsteiger ist der an `startActivityForResult` übergebene Wert `ICY_WIFI_REQCODE` mitunter etwas seltsam. Es ist ein Korrelationscode, der der Einlaufstelle das Erkennen desjenigen Auslösers ermöglicht, der für die eingehende Anfrage verantwortlich war. Korrelationscodes sind dabei normale Integer-Werte, denen vom Betriebssystem kein signifikanter Wert zugeordnet wird. Der Autor deklariert sie in seinem Programm nach folgendem Schema - wichtig ist nur die Einzigartigkeit:

```
public class MainActivity extends
    AppCompatActivity implements
```



Von den Alten lernen!

So sie ihre erste Smartphone-Applikation realisieren, ist das Lesen des Klassikers *The Zen of Palm* eine geradezu unersetzbare Grundvoraussetzung - auch wenn PalmSource seit langer Zeit nicht mehr existiert, sind die Erkenntnisse nach wie vor wertvoll. Unter [5] finden Sie eine PDF-Version des nicht sonderlich umfangreichen Dokuments.

```
AdapterView.OnItemClickListener {
    private final int MY_PERMISSIONS_
        ACCESS_COARSE_LOCATION = 1;
    int ICY_WIFI_REQCODE = 4242;
```

Die nächste Aufgabe ist das Auslösen des eigentlichen WLAN-Scanprozess. Da dieser Artikel nicht in eine Besprechung des GUI-Stacks von Android ausarten soll, gehen wir davon aus, dass Sie einen Weg zur Aktivierung der Methode `getWifi` finden werden.

Ihre erste Amtshandlung besteht dann darin, nach folgendem Schema zu überprüfen, ob wir schon eine Zugriffs-Berechtigung für die Permission `Manifest.permission.ACCESS_FINE_LOCATION` haben – unter Android 13 sollten Sie auch die neue Permission in den Analysecode einbeziehen:

```
private void getWifi() {
    if (ContextCompat.
        checkSelfPermission(MainActivity.this,
            Manifest.permission.ACCESS_FINE_LOCATION) !=
            PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.
            requestPermissions(MainActivity.this,
                new String[]{Manifest.permission.
                    ACCESS_FINE_LOCATION},
                    MY_PERMISSIONS_ACCESS_COARSE_LOCATION);
    }
```

Zur Erinnerung: Seit inklusive Android 6.0 werden „empfindliche“ Berechtigungen vom Betriebssystem nur noch dann freigeschaltet, wenn der Benutzer dem explizit zustimmt. Hierfür dienen Dialoge, die sich wie in **Bild 4** gezeigt präsentieren.

Sofern wir die Permission bereits zur Verfügung gestellt haben, folgt im nächsten Schritt eine weitere Versions-Weiche. Sofern wir es mit einem älteren Telefon zu tun haben, rufen wir die Methode `getWifiWorkerOld` direkt auf:

```
else {
    if (android.os.Build.VERSION.SDK_INT <
        Build.VERSION_CODES.Q) {
        getWifiWorkerOld();
    }
}
```

Im Fall eines neuartigen Telefons wird, um einen Absturz zu vermeiden, an dieser Stelle abermals der Zustand der Energieversorgung des WLAN-Transmitters überprüft, denn es ist Usern durchaus zuzumuten, dass sie den WLAN-Transmitter ausschalten, bevor sie die Methode aktivieren:

```
else {
    if (wifiManager.isWifiEnabled()) {
```

```
        getWifiWorkerOld();
    }
    else {
        Toast toast =
            Toast.makeText(getApplicationContext(),
                "PLEASE enable WiFi so that we can connect!",
                Toast.LENGTH_LONG);
        toast.show();
        Intent panelIntent =
            new Intent(Settings.Panel.ACTION_WIFI);
        this.startActivityForResult(panelIntent,
            ICY_WIFI_REQCODE);
    }
}
```

Sofern wir den Zustand der Energieversorgung des Transmitters erfolgreich überprüft haben, aktivieren wir im nächsten Schritt nach folgendem Schema den Scanprozess:

```
private void getWifiWorkerOld() {
    ...
    wifiManager.startScan();
}
```

`wifiManager` ist dabei eine System-Klasse, die wir im Rahmen der Aktivierung der Activity beschafft haben. Die Methode `startScan` kümmert sich dann darum, dass der eigentliche Scanprozess beginnt. Die Zurücklieferung der vom Scanlauf erbeuteten Informationen erfolgt durch `Broadcast`. Für seine Entgegennahme benötigen wir einen `Broadcast Receiver`. Da wir diesen nicht in der Manifestdatei anlegen wollen, schreiben wir ihn stattdessen im Rahmen der `onPostResume`-Methode der Activity ein:

```
@Override
protected void onPostResume() {
    super.onPostResume();
```

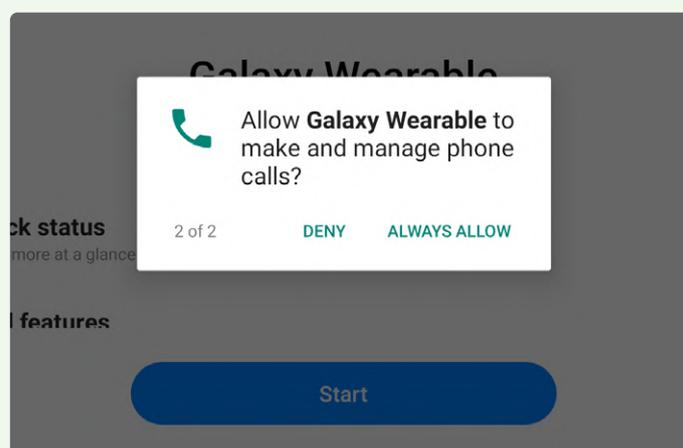


Bild 4. Das neue Permissionsystem bittet um Aufmerksamkeit.

```

receiverWifi =
new WifiReceiver(wifiManager, wifiList, this);
IntentFilter intentFilter =
new IntentFilter();
intentFilter.addAction
(WifiManager.SCAN_RESULTS_AVAILABLE_ACTION);
registerReceiver(receiverWifi, intentFilter);
}

```

Im Interesse der Play-Store-Konformität ist es wichtig, dass manuell angemeldete *BroadcastReceiver* auch wieder beim Betriebssystem abgemeldet werden. Dies lässt sich am bequemsten durch eine der Lifecycle-Methoden der Activity bewerkstelligen: Der Autor entschied sich hier für `onPause`:

```

@Override
protected void onPause() {
    super.onPause();
    unregisterReceiver(receiverWifi);
}

```

BroadcastReceiver zur Datenverarbeitung

Da die Entgegennahme der vom WLAN-Scanprozess zurückgegebenen Netzwerk-Informationen über das Broadcast-System von Android erfolgt, müssen wir die weiter oben beim Betriebssystem angemeldete *BroadcastReceiver*-Klasse realisieren.

BroadcastReceiver entstehen in Android in Form von Klassen, die von der Masterklasse *BroadcastReceiver* abzuleiten sind. Im Fall unseres Beispiels - ignorieren Sie die GUI-Variablen - sieht der „Header“ folgendermaßen aus (den zur Bevölkung der Felder notwendigen Konstruktor drucken wir nicht ab):

```

class WifiReceiver extends BroadcastReceiver {
    WifiManager wifiManager;
    ListView wifiDeviceList;
    MainActivity myParentAct;
}

```

Wichtig ist die Frage, wie die vom WLAN-Transmitter zurückgelieferten Netzwerk-Informationen zu verarbeiten sind. Die Antwort darauf findet sich in der Methode `onReceive`, die das Betriebssystem beim Eintreffen der übrigens in Form von Intents angelieferten Broadcast-Informationen aktivieren wird:

```

public void onReceive(Context context, Intent intent) {
    String action = intent.getAction();
    if (WifiManager.
SCAN_RESULTS_AVAILABLE_ACTION.equals(action)) {
        List<ScanResult> wifiList =
wifiManager.getScanResults();
        ArrayList<String> deviceList =
new ArrayList<>();
        for (ScanResult scanResult : wifiList) {
            if(scanResult.SSID.
contains("NAMEOFTHING"))

```

```

deviceList.add(scanResult.SSID );
}
myParentAct.myArrayAdapter=
new ArrayAdapter(context,
android.R.layout.simple_list_item_1,
deviceList.toArray());
wifiDeviceList.
setAdapter(myParentAct.myArrayAdapter);
}
}
}
}

```

Im ersten Schritt ruft der Code `intent.getAction()` auf, um den Namen des Intents zu überprüfen. *Broadcast Receiver* können prinzipiell mit beliebigen Intents beschossen werden - durch Prüfung gegen den String `WifiManager.SCAN_RESULTS_AVAILABLE_ACTION` stellen wir sicher, dass wir es mit einem „kompatiblen“ Intent zu tun haben. Da das Code-Beispiel, aus dem der Autor die folgenden Snippets entnommen hat, eine Liste zur Anzeige verwendet, ist im nächsten Schritt etwas Swizzling erforderlich. Sinn des Prozesses ist jedenfalls die Bevölkung der Klasse `deviceList`.

In einer realen Applikation würde an dieser Stelle eine Liste mit Informationen gefüllt werden - sofern Sie nur ein einzelnes Ziel-WLAN als gültig erachten, können Sie an dieser Stelle natürlich auch anders vorgehen.

Ein Beispiel für die andere Vorgehensweise wäre ein Gerät, das während der Konfiguration Kontakt zu seiner Basis-Applikation aufnimmt und dies durch einen bestimmten WLAN-Namen anzeigt. In diesem Fall könnten wir nach Erkennung des WLANs sofort weitermachen.

Geschickter Verbindungsaufbau

Benutzer mögen bei Smartphone- oder PDA-Applikationen keine längere Wartezeiten. Daher ist das Einblenden von Fortschrittsbalken oder ähnlichen Gadgets sinnvoll, die dem Benutzer versichern, dass die Applikation nicht abgestürzt ist.

In der Kunden-Applikation, die als Basis für diesen Artikel dient, wurde dies in Form einer dedizierten Activity realisiert: Sie zeigt das Firmenlogo und einen Fortschrittsbalken an und hält den Benutzer so bei Laune. Ihre Aktivierung erfolgt durch Senden eines Intents:

```

public void onItemClick(AdapterView<?>
adapterView, View view, int pos, long anID) {
    String y = (String) myArrayAdapter.getItem(pos);
    Intent intent =
new Intent(MainActivity.this,ConnActivity.class);
    intent.putExtra("WIFINAME", y);
    startActivity(intent);
}
}

```

Im „Code Behind“ beginnt sie mit der Deklaration eines *Async-Tasks*, der einen Präsenzttest durchführt. Es handelt sich dabei um eine (hier nicht zu besprechende) Verifikation, die überprüft, ob die Verbindungs-Gegenstelle einen Teil des Kunden-Ökosystems darstellt:



Achtung, googlesche Blockade

Permanentes Abfeuern von Permission-Anfragen wird in manchen Android-Versionen vom Betriebssystem mit dem Blockieren der Anfrage-Funktionalität bestraft. Behalten Sie dies bei der Realisierung ihres Benutzer-Belästigungsprozesses auf jeden Fall im Hinterkopf.

```
public class ConnActivity extends AppCompatActivity {
    PresenceTestAsynctask aT;
    ConnActivity myself;
```

In `onCreate` beginnt der Verbindungsaufbau ohne jedes weiteres Zutun des Benutzers. Die erste Amtshandlung ist das Konfigurieren einiger Member-Variablen und der Aufruf der Methode `getIntent().getExtras()`, um die weiter oben per Intent „paketierte“ Informationen mit dem WLAN-Namen und anderen Unterstützungsdaten zugänglich zu machen:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_conn);
    myself=this;
    Bundle b = getIntent().getExtras();
    String networkPass = "sdsds";
```

Im nächsten Schritt findet sich abermals eine Versionsweiche, die den „Versions-Stand“ der vorliegenden Ziel-Hardware überprüft.

Sofern wir es mit einem alten Telefon zu tun haben, folgt dann im nächsten Schritt der nach dem folgenden Schema aufgebaute Beginn des Verbindungsaufbaus:

```
if (android.os.Build.VERSION.SDK_INT <
Build.VERSION_CODES.Q) {
    WifiConfiguration conf =
    new WifiConfiguration();
    conf.SSID = "\"" +
    b.getString("WIFINAME") + "\"";
    conf.preSharedKey = "\"" + networkPass + "\"";
    WifiManager wifiManager =
    (WifiManager) getApplicationContext().
    getSystemService(Context.WIFI_SERVICE);
    wifiManager.addNetwork(conf);
```

Die Methode `addNetwork` übernimmt ein Objekt vom Typ `WifiConfiguration`. Seine Aufgabe ist das Anliefern eines kompletten WLAN-Informationssatzes, zu dem das Telefon danach Verbindung aufnimmt.

Im letzten Akt müssen wir dann nach folgendem Schema den eigentlichen Verbindungsaufbau befehlen, was durch den Methoden-Dreikampf `disconnect`, `enableNetwork` und `reconnect` erfolgt. Danach kommt die Aktivierung des AsyncTasks - wenn er die Gegenstelle erfolgreich authentifiziert, beginnt die Kunden-Applikation mit der eigentlichen Konfiguration der angeschlossenen Hardware:

```
//Can only come here from permission granted state
@SuppressLint("MissingPermission")
List<WifiConfiguration> list =
    wifiManager.getConfiguredNetworks();
for (WifiConfiguration i : list) {
    if (i.SSID != null && i.SSID.equals("\"" +
    b.getString("WIFINAME") + "\"")) {
        final ConnectivityManager connectivityManager =
            (ConnectivityManager)getApplicationContext().
            getSystemService(Context.CONNECTIVITY_SERVICE);
        ConnectivityManager.NetworkCallback
        networkCallback =
        new ConnectivityManager.NetworkCallback() {
            @Override
            public void onAvailable(@NonNull Network network) {
                super.onAvailable(network);
                //ONLY IF ANDROID 9
                if (android.os.Build.VERSION.SDK_INT ==
                    Build.VERSION_CODES.Q) {
                    connectivityManager.
                        bindProcessToNetwork(network);
                }
            }
        };
        connectivityManager.
            registerDefaultNetworkCallback(networkCallback);

        wifiManager.disconnect();
        wifiManager.enableNetwork(i.networkId, true);
        wifiManager.reconnect();
        aT = new PresenceTestAsynctask
        (getApplicationContext());
        aT.myAct=mySelf;
        aT.execute("...");
        break;
    }
}
```

Auf neuartigen Telefonen ist an dieser Stelle abermals ein Benutzer-Eingriff erforderlich. Die Arbeit beginnt damit, nach folgendem Schema ein `NetworkSpecifier`-Objekt anzulegen:

```
}else{
    final NetworkSpecifier specifier =
    new WifiNetworkSpecifier.Builder()
        .setSsidPattern(new PatternMatcher(
        b.getString("WIFINAME"),
        PatternMatcher.PATTERN_PREFIX))
        .setWpa2Passphrase(networkPass)
        .build();
```

`NetworkSpecifier`-Objekte sind eine Art Suchmaske, die der in Android enthaltenen GUI beim Identifizieren von für den Anwendungsfall relevanten Drahtlosnetzwerk-Gegenstellen hilft.

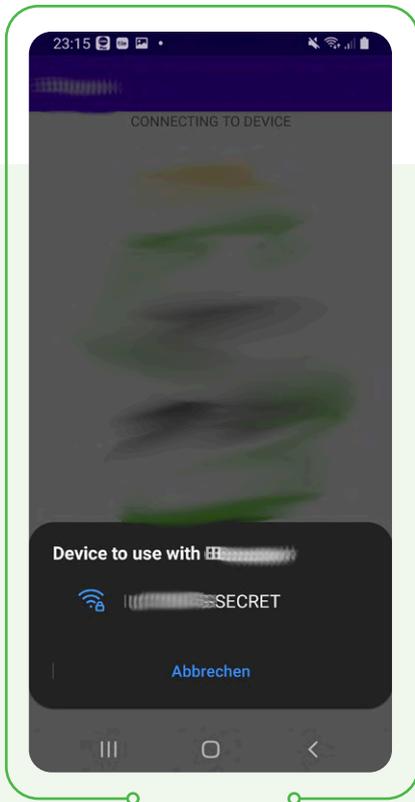


Bild 5. Android Q fragt den Benutzer vor dem Verbindungsaufbau.

Der nächste Schritt ist der Aufbau eines `NetworkRequest`-Elements, was nach folgendem Schema erfolgt:

```
final NetworkRequest request =
    new NetworkRequest.Builder()
        .addTransportType(NetworkCapabilities.
            TRANSPORT_WIFI)
        .removeCapability(NetworkCapabilities.
            NET_CAPABILITY_INTERNET)
        .setNetworkSpecifier(specifier)
        .build();
```

Der eigentliche Verbindungsaufbau erfolgt wie in **Bild 5** gezeigt. Der Benutzer muss dazu den WLAN-Namen antippen, den er als Konfigurations-Ziel verwenden möchte.

Nebeneffekt dieser Vorgehensweise ist, dass der WLAN-Verbindungsaufbau asynchron erfolgen muss. Die Rückmeldung über Erfolg und Misserfolg erfolgt über einen Callback, der in der für die Aktivierung verantwortlichen Applikation anzulegen ist:

```
final ConnectivityManager connectivityManager =
    (ConnectivityManager)getContext().
    getSystemService(Context.CONNECTIVITY_SERVICE);
final ConnectivityManager.NetworkCallback
    networkCallback =
    new ConnectivityManager.NetworkCallback() {
        @Override
        public void onAvailable(@NonNull Network network) {
            super.onAvailable(network);
        }
    };
```

```
connectivityManager.
    bindProcessToNetwork(network);
aT = new PresenceTestAsyncTask
    (getApplicationContext());
aT.myAct=mySelf;
aT.execute(" . . .");
}
@Override
public void onUnavailable() {
    super.onUnavailable();
}
};
```

Wenn die Methode `onAvailable` aufgerufen wird, verlief der Verbindungsaufbau erfolgreich. Die Applikation startet in diesem Fall abermals den `AsyncTask`, um die Gegenstelle zu authentifizieren und bei Bedarf weitere Aktionen vorzunehmen.

Die letzte Amtshandlung unserer Scanmethode ist dann der Aufruf von `requestNetwork`, der die weiter oben gezeigte Benutzerschnittstelle aktiviert:

```
connectivityManager.
    requestNetwork(request, networkCallback);
}
}
}
```

Lohnt es sich?

Über die Frage, ob sich der Aufwand für die Realisierung einer Android-Kombination lohnt, lässt sich hervorragend diskutieren. Dem Autor ist bekannt, dass Echtzeitbetriebssysteme - explizites Stichwort ist hier Azure RTOS - und Embedded-GUI-Stacks durchaus leistungsfähige Grafikschnittstellen realisieren, die fortgeschrittene Anwendungsfälle abdecken.

Gegen diese Vorgehensweise spricht allerdings, dass sich dadurch erstens die Kosten für die Hardware erhöhen - Touchscreen und die dazugehörige Logik (Stichwort: Framebuffer-Arbeitsspeicher) gibt es nicht ohne Einfluss auf die Bill of Materials. Dazu kommt der Entwicklungsaufwand: Wer in einem Echtzeitbetriebssystem beispielsweise das Versenden von Bildern und Ergebnissen per E-Mail realisieren möchte, darf einige Dutzend Mannstunden abschreiben. Außerdem - dies ist insbesondere im Consumerbereich relevant - sind mit Telefonen verwendbare beziehungsweise fernsteuerbare Accessories insbesondere bei Early Adopters „hip“.

Unterm Strich hofft der Autor, dass die hier durchgeführten Experimente und Exkurse einen Weg durch den Dschungel zeigen - falls Sie eine derartige Lösung realisieren, freuen wir uns naturgemäß über einen Leserbrief. ◀

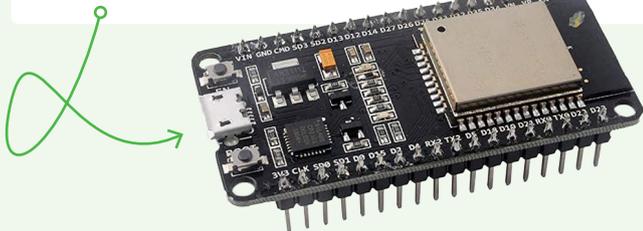
Sie haben Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare? Dann senden Sie eine E-Mail an den Autor unter tamhan@tamoggemon.com oder die Elektor-Redaktion unter redaktion@elektor.de.



Passende Produkte

- **ESP32-PICO-Kit V4 (mit Buchsenleisten)**
SKU 20323: www.elektor.de/20323
- **ESP32-DevKitC-32D**
SKU 18701: www.elektor.de/18701



WEBLINKS

- [1] Webseite zum Artikel:
<https://www.elektormagazine.de/210229-02>
- [2] ESP-IDF: <https://idf. espressif.com/>
- [3] Android Studio: <https://developer.android.com/studio>
- [4] Request permission to access nearby Wi-Fi devices, Android Developers: <https://developer.android.com/about/versions/13/features/nearby-wifi-devices-permission>
- [5] Zen of Palm (2003): <https://news.ycombinator.com/item?id=9282768>

**KONTAKT
CHEMIE**

By CRC Industries

KONTAKTPFLEGE IN DREI SCHRITTEN

1 OXIDE ENTFERNEN
MIT **KONTAKT 60**



2 REINIGEN/ENTFETTEN
MIT **KONTAKT WL**



3 SCHÜTZEN MIT
KONTAKT 61



crcind.info/KOC_Brochure_DE

hello@crcindinfo.com

Aktives 1-kHz-Filter für Verzerrungsmessungen

Bessere Messungen durch Optimierung des Messsignals

Von Alfred Rosenkränzer (Deutschland)

Die Erfassung selbst geringer Verzerrungen stellt nicht nur große Anforderungen an den Dynamikbereich und die Linearität des folgenden Messgeräts, sondern auch an das Messsignal, das der zu prüfenden Baugruppe zugeführt wird. Mit einem nachgeschalteten Filter können aber selbst durchschnittliche Signalgeneratoren hochqualitative Messsignale liefern.

Im Artikel „Fliege-Bandsperre für Audio-Messungen“ in Elektor September/Oktober 2022 [1] wurde gezeigt, wie man mit Hilfe eines Notch-Filters den Messbereich für Verzerrungsmessungen bei einem Digitizer erweitern kann. Dabei wird das Testsignal unterdrückt und so verhindert, dass der Digitizer selbst Verzerrungen erzeugt. Voraussetzung für die valide Erfassung niedriger Verzerrungspegel des Testobjekts ist selbstverständlich, dass der den Prüfling speisende Signalgenerator

ein hochwertiges Signal erzeugt. Ansonsten misst man neben den Verzerrungen der zu prüfenden Schaltung auch die des Generators und erhält unsaubere Messergebnisse.

Bei bezahlbaren Signalgeneratoren unterhalb der Profiklasse lässt die Qualität des erzeugten Signals allerdings oft zu wünschen übrig. Ein nachgeschaltetes Filter kann hier ein kleines Wunder bewirken und durch Dämpfung der Verzerrungen des Signalgenerators ein besonders „reines“ Messsignal generieren.

Anforderungen

Da die meisten Messungen im Audibereich bei der typischen Messfrequenz von 1 kHz erfolgen, reicht in der Regel ein Filter mit fester Durchlass- beziehungsweise Grenzfrequenz bei diesem Wert. Das Filter sollte die Testfrequenz ungedämpft passieren lassen, aber die Oberwellen (die Vielfachen der Testfrequenz) möglichst stark dämpfen. Eine weitere und nicht unwichtige Forderung ist: Das Filter darf selbst natürlich keine nennenswerten Verzerrungen produzieren!

Für die einfache Handhabung ist es sinnvoll, dass dieses Filter selbst nicht zu schmalbandig ausfällt, sondern eine gewisse Bandbreite besitzt. Dadurch kann man die Frequenz des Messsignals ohne Pegelschwankungen gut auf die exakte Frequenz des schmalbandigen Notch-Filters abstimmen, welches dann das Messsignal vor der Messung ausfiltert. **Bild 1** zeigt das Prinzip dieser Messung mit beiden Filtern vor und nach dem DUT (**D**evice **U**nder **T**est).

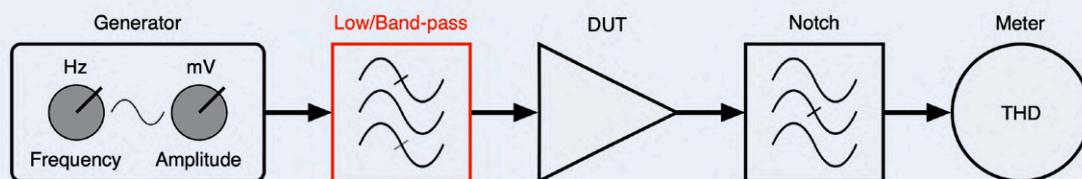


Bild 1. Das optimierte Messprinzip für Verzerrungen mit je einem Filter vor und hinter dem DUT. Der vorgeschaltete Tief- oder Bandpass (rot) entfernt Verzerrungen aus dem Signal des Generators.

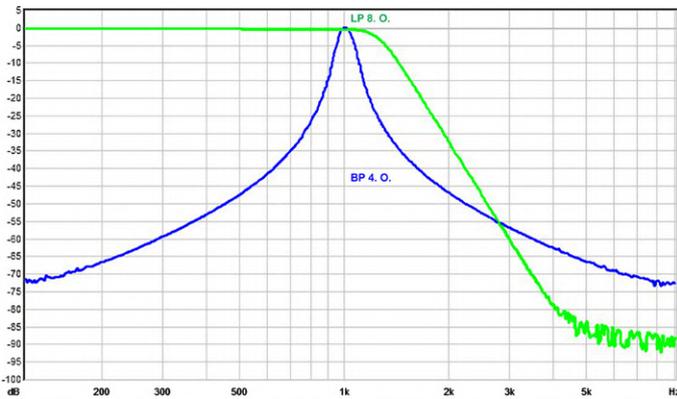


Bild 2. Die Frequenzgänge eines Butterworth-Tiefpasses 8. Ordnung (grün) und eines Bandpasses 4. Ordnung (blau) im Vergleich.

Filter

Prinzipiell kann man die Reduktion der Verzerrungen des Signalgenerators in Form der Dämpfung von Oberwellen ja nicht nur mit einem Bandpass, sondern auch mit einem passenden Tiefpass erreichen. Die Frage ist, welche Anordnung und welcher Filtertyp die beste Oberwellendämpfung relativ zum Bauteileaufwand hat. **Tabelle 1** zeigt die gemessenen Dämpfungen eines Bandpasses 4. Ordnung im Vergleich mit verschiedenen Tiefpässen für ein Testsignal von 1 kHz. Besonders relevant ist dabei die Dämpfung bei 2 kHz und 3 kHz, da diese Oberwellen bei den meisten Generatoren die höchsten Amplituden haben.

Ein Butterworth-Tiefpass 4. Ordnung dämpft bei 2 kHz nur um knapp 12 dB und ist damit für diesen Einsatzbereich nicht geeignet. Ein Tschebyscheff-Tiefpass gleicher Ordnung kommt hier immerhin schon auf

-26 dB. Noch deutlich besser ist ein Butterworth-Tiefpass 8. Ordnung oder ein Bandpass 4. Ordnung mit einer Bandbreite von 100 Hz. **Bild 2** zeigt die Frequenzgänge dieser beiden Filter im Vergleich. Da diese beiden Filter zu bevorzugen sind, wurden hierfür Platinen in derselben Größe wie das Notch-Filter von [1] entwickelt. Auch die damals verwendete Spannungsregler-Platine kann genutzt werden.

Tabelle 1. Filterdämpfungen

Frequenz (kHz)	BP 4. O. (dB)	TP Bw. 4. O. (dB)	TP Tsch. 4. O. (dB)	TP Bw. 8. O. (dB)
1,0	0,0	0,1	0,3	0,0
2,0	-45,0	-11,7	-26,0	-32,0
3,0	-56,0	-25,2	-42,0	-60,0
4,0	-62,0	-35,0	-52,0	-80,0
5,0	-66,0	-42,8	-60,0	-90,0
6,0	-69,0	-49,5	-67,0	-90,0
7,0	-72,0	-54,0	-72,0	-90,0
8,0	-74,0	-59,0	-77,0	-90,0
9,0	-75,0	-63,0	-81,0	-90,0
10,0	-76,0	-67,0	-85,0	-90,0

Schaltungen

Bild 3 zeigt die Schaltung des Butterworth-Tiefpasses 8. Ordnung. Damit im Bereich einiger zehn Hertz um 1 kHz keine nennenswerte Pegelabschwächung auftritt, ist seine Grenzfrequenz mit den

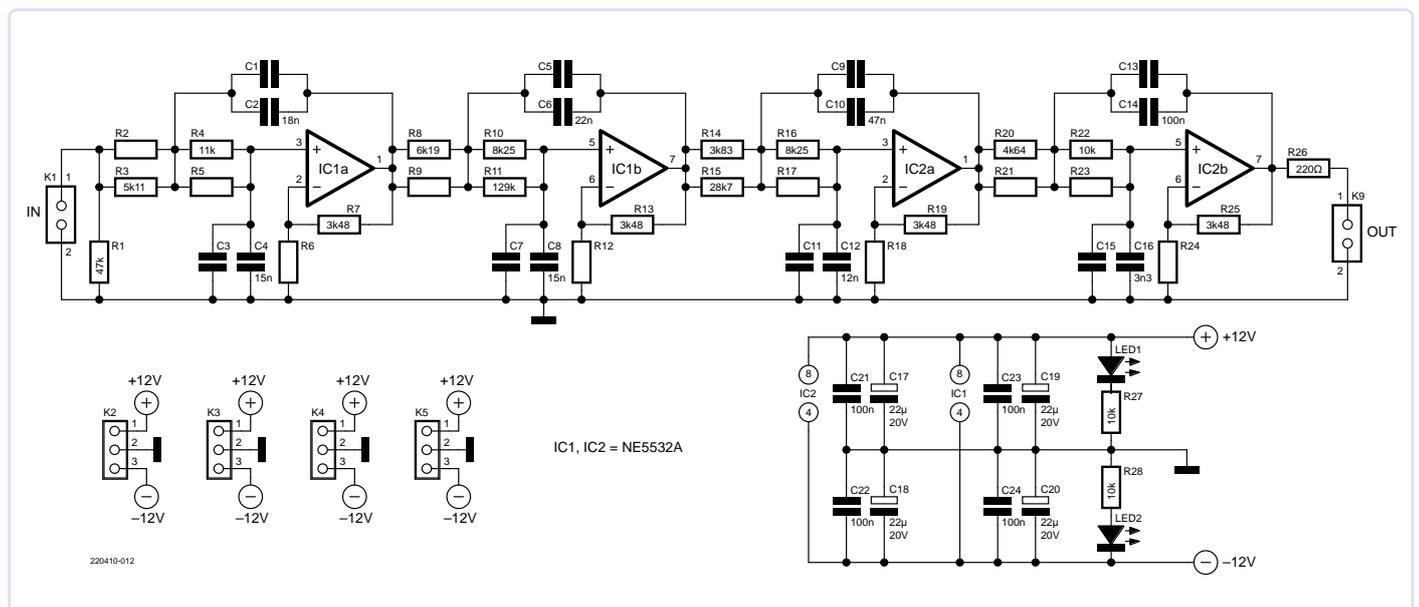


Bild 3. Die Schaltung des Butterworth-Tiefpasses 8. Ordnung.

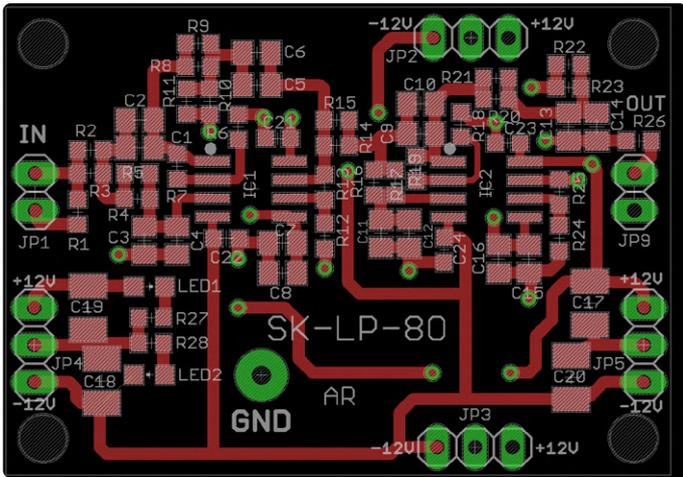


Bild 4. Das Platinen-Layout für den Butterworth-Tiefpass 8. Ordnung.



Bild 5. Fertig bestückter Tiefpass-Prototyp des Autors.

angegebenen Bauteilwerten auf 1,28 kHz festgelegt. **Bild 4** zeigt das Layout und **Bild 5** den Prototyp des Tiefpasses. Bei den die Frequenz bestimmenden Widerständen und Kondensatoren ist jeweils ein weiteres Bauteil parallelgeschaltet (in Bild 3 ohne Wertangabe), damit der errechnete Wert mit den handelsüblichen E-Reihen möglichst genau getroffen werden kann. Es empfiehlt sich besonders, die Kondensatoren vor der Bestückung auf Exaktheit auszuwählen, um nachträgliches Ersetzen einzelner Bauteile zu vermeiden. Es sind auf der Platine mit R6, R12, R18 und R24 auch zusätzliche Widerstände vorgesehen, die eine Filterverstärkung erlauben, welche aber hier nicht benötigt werden und daher entfallen.

Allgemein empfiehlt sich die Lektüre des Artikels zum Fliege-Bandpass [1] und gegebenenfalls auch des Artikels über Filter-Software [2], denn darin wird das Vorgehen beim genauen Abgleich der Filterfrequenzen

und die dafür eingesetzte Software zur Bauteilberechnung ausführlich erläutert. Ein Optimierungsbeispiel: Der gemessene Wert des 100-nF-Kondensators C14 beträgt 103 nF, was in der 5%-Toleranz des Bauteils liegt. Gibt man nun den Wert 103 nF in die Software ein, erhält man bei gegebener Frequenz die angepassten Werte für die anderen Bauteile dieser Stufe – und Widerstände bekommt man ja leicht und preiswert auch mit einer Toleranz von 1%.

Auch beim Bandpass (Schaltung **Bild 6**, Layout **Bild 7** und Prototyp **Bild 8**) ist der Autor so vorgegangen. Bei den frequenzbestimmenden Bauteilen wurde jeweils ein weiteres Bauteil für eine eventuelle Parallelschaltung vorgesehen und eine optionale Verstärkung mit R8, R9, R18 und R19 realisiert. Da die Werte der frequenzbestimmenden Kondensatoren hier gleich sind, sollte man möglichst Bauteile aus derselben Charge verwenden und die wahrscheinlich nur geringen

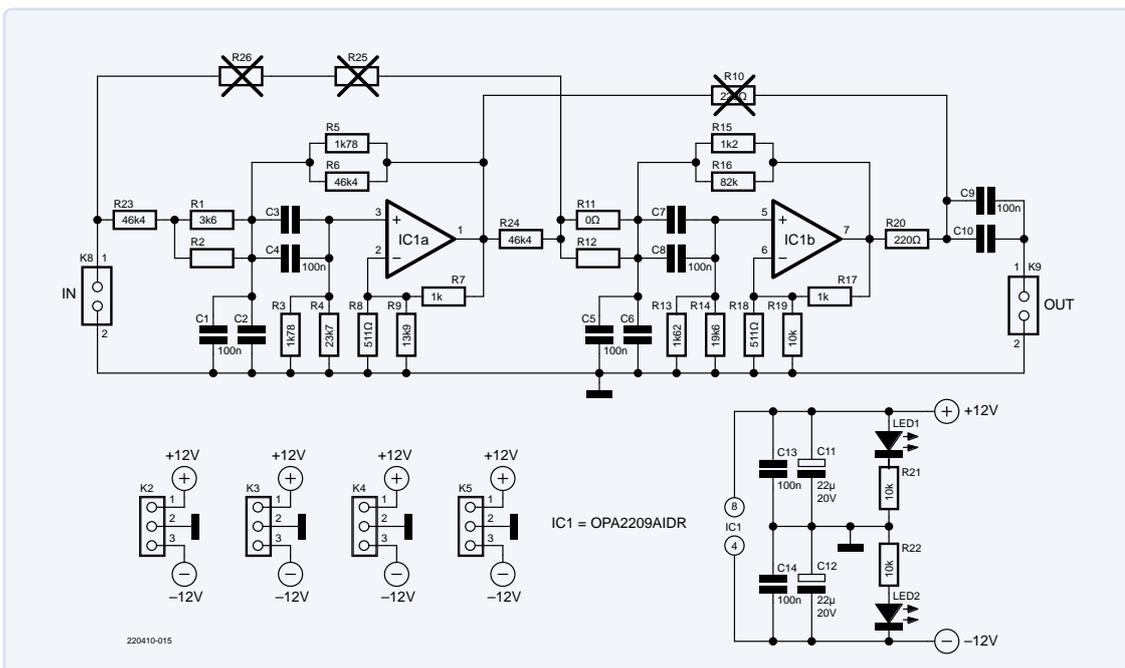


Bild 6. Die Schaltung des Bandpasses 4. Ordnung mit 100 Hz Bandbreite.

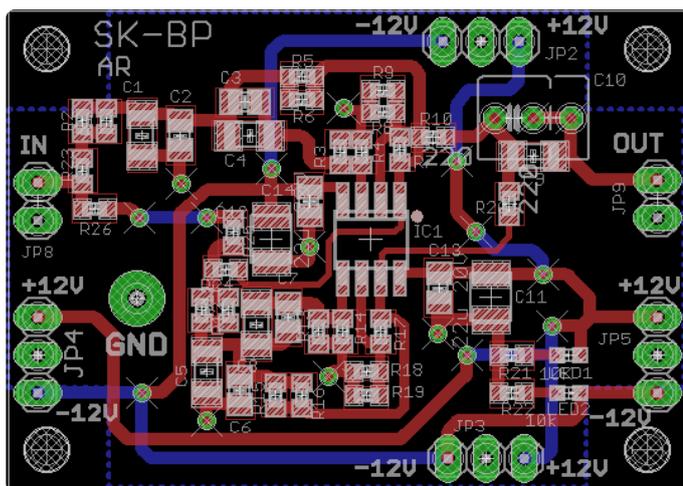


Bild 7. Das Platinen-Layout für den Bandpass 4. Ordnung.

Unterschiede mit parallelen Kondensatoren ausgleichen. Es empfiehlt sich hier ebenfalls, den resultierenden Wert in der Software einzugeben. Um die Widerstandswerte am Eingang jeder Stufe möglichst genau realisieren zu können, wurde jeweils eine Kombination von einem Widerstand in Reihe zu einer Parallelschaltung aus zwei Widerständen vorgesehen. Die Widerstände R10, R25 und R26 dienen wie in [1] beschrieben zum Messen der Charakteristik der einzelnen Stufen und entfallen somit. Zwecks Feinabstimmung empfiehlt sich eine Simulation der Schaltung mit den realen, gemessenen Werten.

Welches Filter?

Nachdem Sie jetzt die Wahl haben, entweder einen Tiefpass oder einen Bandpass einzusetzen, stellt sich natürlich die Frage: Welcher Filtertyp eignet sich besser für Ihre Zwecke?

Der Tiefpass braucht für eine brauchbare Dämpfung eine höhere Ordnung und benötigt deshalb vier Opamps. Damit steigen Eigenrauschen und natürlich auch die durch das Filter produzierten Verzerrungen. Auf der anderen Seite dämpft er Oberwellen höherer Ordnung ab etwa 3 kHz stärker als der Bandpass. Seine Eigenschaft, alle Frequenzen unterhalb der Grenzfrequenz durchzulassen, kann für andere Anwendungen günstig sein.

Der Bandpass 4. Ordnung kommt dagegen mit nur zwei Opamps aus und dürfte daher weniger Eigenrauschen und eigene Verzerrungen produzieren. Da er bei einer Bandbreite von 100 Hz nur Frequenzen mit geringer Abweichung von 1 kHz durchlässt, eignet er sich kaum für andere Anwendungen. Doch da er alle Frequenzen unterhalb von 1 kHz und somit auch unschöne Phänomene wie Netzbrummen dämpft, wird zudem auch niederfrequentes Eigenrauschen reduziert, was SNR-Messungen zugutekommen kann.

Sonstiges

Die Bestückung der Platinen ist natürlich aufgrund der Kleinheit der SMD-Bauelemente nichts für Leute, die zum ersten Mal von Hand löten. Geübte Elektroniker sollten damit aber kein Problem haben. Die Layoutdateien können im Eagle-Format kostenlos von der Webseite dieses Artikels [3] heruntergeladen werden. Aufgrund ihrer Eigenschaften sollte man bei allen Widerständen, die mit dem Signal zu tun haben, Dünnfilmausführungen guter Qualität verwenden.

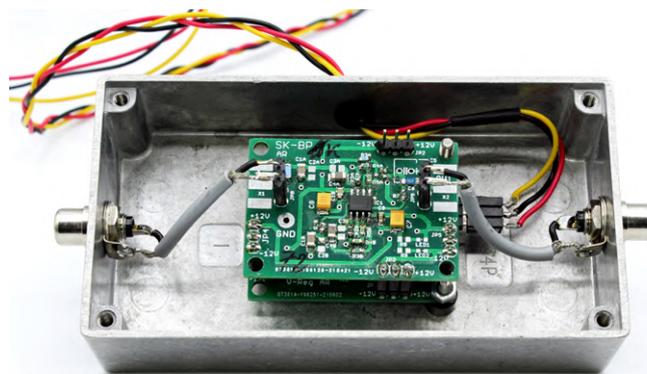


Bild 8. Fertig bestückter Bandpass-Prototyp, eingebaut in ein Alugehäuse.

Die signalführenden SMD-Keramik-Kondensatoren sollten zwingend Typen mit sogenanntem COG-Dielektrikum sein, um stabile Eigenschaften und geringe Eigenverzerrungen zu garantieren. Der Autor hat noch ein paar Leerplatinen übrig. Bei Bedarf können Sie sich per E-Mail (siehe „Über den Autor“) bei ihm melden.

Die beiden hier vorgestellten Filter sind natürlich nicht auf 1 kHz beschränkt. Man kann sich mit Hilfe der Designüberlegungen unter [1] leicht einen ganzen Satz an passenden Tief/Bandpässen und Notch-Filtern unterschiedlicher Frequenzen herstellen und damit das Verzerrungsverhalten von Audioschaltungen bei verschiedenen Frequenzen untersuchen. ◀

220410-02

Über den Autor

Alfred Rosenkränzer arbeitete viele Jahre als Entwicklungsingenieur, zu Anfang im Bereich der professionellen Fernsehtechnik. Seit Ende der 1990er Jahre entwickelt er digitale High-Speed- und Analogschaltungen für IC-Tester. Das Thema Audio ist sein privates Steckenpferd.

Sie haben Fragen oder Kommentare?

Haben Sie technische Fragen oder Anmerkungen zu diesem Artikel? Schicken Sie eine E-Mail an den Autor unter alfred_rosenkraenzer@gmx.de oder kontaktieren Sie Elektor unter redaktion@elektor.de.

WEBLINKS

- [1] Alfred Rosenkränzer, „Fliege-Bandsperre für Audio-Messungen“, Elektor 9-10/2022: <https://www.elektormagazine.de/magazine/elektor-273/60954>
- [2] Alfred Rosenkränzer, „Filter-Software“, Elektor 11-12/2022: <https://www.elektormagazine.de/magazine/elektor-281/61083>
- [3] Webseite zu diesem Artikel: <https://www.elektormagazine.de/220410-02>



STÜCKLISTE TIEFPASS

Widerstände:

(Dünnschicht, 1 %, SMD 0603)

R1 = 47 k

R2,R5,R9,R17,R21,R23 = Parallelschaltung *

R3 = 5k11

R4 = 11 k

R6,R12,R18,R24 = Verstärkung, entfällt *

R7,R13,R19,R25 = 3k48

R8 = 6k19

R10,R16 = 8k25

R11 = 129k

R14 = 3k83

R15 = 28k7

R20 = 4k64

R22,R27,R28 = 10 k

R26 = 220 Ω

Kondensatoren:

(wenn nicht anders angegeben 5 %, COG,

SMD0805)

C1,C3,C5,C7,C9,C11,C13,C15 = Parallelschaltung *

C2 = 18 n

C4,C8 = 15 n

C6 = 22 n

C10 = 47 n

C12 = 12 n

C14 = 100 n

C16 = 3n3

C17,C18,C19,C20 = Elko 22 μ / 20V, SMD-B

C21,C22,C23,C24 = 100 n, SMD0603

Halbleiter:

LED1,LED2 = LED, grün, SMD0805

IC1,IC2 = NE5532A, SO08

Außerdem:

K1,K9 = 2-pol. Stiftleiste, RM 1/10"

K2...K5 = 3-pol. Stiftleiste, RM 1/10"

* siehe Text



STÜCKLISTE BANDPASS

Widerstände:

(Dünnschicht, 1 %, SMD 0603)

R1 = 3k6

R2,R12 = Parallelschaltung *

R3,R5 = 1k78

R4 = 23k7

R6,R23,R24 = 46k4

R7,R17 = 1 k

R8,R18 = 511 Ω

R9 = 13k9

R10,R25,R26 = entfällt *

R11 = 0 Ω

R13 = 1k62

R14 = 19k6

R15 = 1k2

R16 = 82 k

R19,R21,R22 = 10 k

Kondensatoren:

(wenn nicht anders angegeben 5 %, COG,

SMD0805)

C1,C4,C5,C8,C9 = 100 n, SMD1206

C2,C3,C6,C7 = Parallelschaltung *

C10 = entfällt

Halbleiter:

LED1,LED2 = LED, grün, SMD0603

IC1 = OPA2209AIDR, SO08

Außerdem:

K8,K9 = 2-pol. Stiftleiste, RM 1/10"

K2...K5 = 3-pol. Stiftleiste, RM 1/10"

* siehe Text



Passende Produkte

> OWON AG051 Arbiträr-Funktionsgenerator (5 MHz)

SKU 18874: www.elektor.de/18874

> OWON SDS1102 2-Kanal-Oszilloskop (100 MHz)

SKU 18782: www.elektor.de/18782

> OWON XSA810 Spektrum-Analyser (1 GHz)

SKU 19714: www.elektor.de/19714



Aller Anfang...

muss nicht schwer sein: Wir multivibrieren weiter!

Von Eric Bogers (Elektor)

In der letzten Folge haben wir (endlich) den Transistor kennengelernt - ein Bauteil, das wir heute getrost als das „Herz unserer Gesellschaft“ bezeichnen können: Ohne Transistoren in der einen oder anderen Form wäre unsere Gesellschaft in ihrer heutigen Form nicht denkbar. Gehen wir also nun zu den Multivibratoren über und betrachten wir den Transistor als Verstärker!

seinem bistabilen Bruder abgeleitet, kennt aber, wie der Name schon sagt, nur einen stabilen Zustand. Das Prinzipschaltbild ist in **Bild 2** zu sehen. Auch wenn die Schaltung durch einen Impuls „von außen“ in einen anderen Zustand versetzt werden kann, kehrt sie nach einer bestimmten Zeit automatisch in den stabilen (Ausgangs-) Zustand zurück. Sie kennen diese Art von Schaltung vielleicht von den praktischen Treppenhauslichtern, die sich nach einer bestimmten Zeit automatisch wieder ausschalten.

In der Schaltung in Bild 2 ist der linke Transistor zunächst leitend. Wie bei der Kippschaltung aus Bild 1 können wir diesen Transistor sperren, indem wir seine Basis über den Taster mit Masse kurzschließen. Dadurch wird auch der Elektrolytkondensator sofort entladen.

Jetzt wird der rechte Transistor (vorübergehend) leitend. Allerdings wird dadurch auch der Kondensator über seinen Widerstand wieder aufgeladen. Sobald die

Zunächst ein kurzes Wort vorweg, vor allem für neue Elektor-Leser, die zum ersten Mal mit einer Folge der Reihe „Aller Anfang...“ konfrontiert werden. Diese Artikelserie erhebt in keiner Weise den Anspruch, ein gründlicher Theoriekurs in Elektronik zu sein. Wir wollen dem Elektronik-Neuling lediglich so viele Fakten an die Hand geben, dass er mit einiger Aussicht auf Erfolg selbst eine Schaltung entwerfen und bauen (oder eine bestehende Schaltung modifizieren) kann. Dabei nehmen wir es zugunsten leichter Berechnungen mit der Theorie hier und da nicht so genau. Uns ist klar, dass wir auf diese Weise den Perfektionisten und Theoretikern unter unseren Lesern manchmal vor den Kopf stoßen werden - das tut uns ein bisschen leid, aber nicht allzu sehr...

Letztes Mal haben wir den bistabilen Multivibrator oder Flip-Flop besprochen, dessen prinzipielle Schaltung wir in **Bild 1** noch einmal zeigen. Diese Schaltung verdankt ihren Namen der Tatsache, dass sie zwei stabile Zustände kennt: Entweder ist der

eine oder der andere Transistor in Betrieb. Das Umschalten ist mit Hilfe der Taster möglich. Welcher der beiden Transistoren unmittelbar nach dem Einschalten der Versorgungsspannung leitend ist, lässt sich nicht vorhersagen; sicher ist nur, dass einer der beiden leitet und der andere nicht.

Der monostabile Multivibrator

Ein monostabiler Multivibrator (auch unter dem Spitznamen *Monoflop* bekannt) ist von

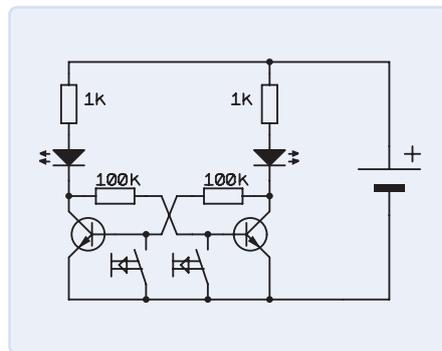


Bild 1. Der bistabile Multivibrator oder Flip-Flop.

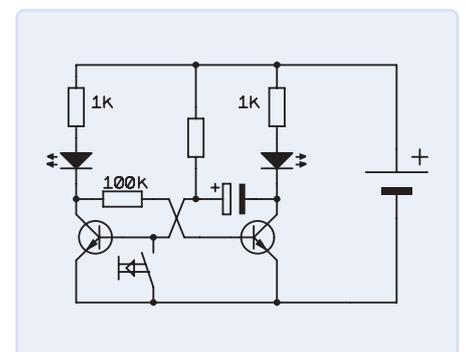


Bild 2. Der monostabile Multivibrator.

Spannung am Elektrolytkondensator etwa 0,7 V überschreitet, leitet der linke Transistor wieder, so dass seine CE-Spannung auf fast 0 V sinkt. Infolgedessen beginnt der rechte Transistor wieder zu sperren. Der stationäre Zustand ist wieder erreicht.

Die Zeit, die das Monoflop braucht, um in den stationären Zustand zurückzukehren, hängt (natürlich) vom Wert des (Lade-)Widerstands und des Kondensators ab, aber auch von der Höhe der Versorgungsspannung. Versuchen wir nun, diese „Monozeit“ über den Daumen zu berechnen. Wir nehmen die folgenden Werte an: $R = 100 \text{ k}\Omega$, $C = 100 \text{ }\mu\text{F}$ und $U_V = 12 \text{ V}$. Für die Spannung an einem Kondensator gilt:

$$U_C = \frac{Q}{C} = \frac{I \cdot t}{C}$$

Der Strom wird durch die Spannung über dem Widerstand und den Wert dieses Widerstands bestimmt. Diese Spannung nimmt kontinuierlich ab, je höher die Spannung des Kondensators steigt. Da die Schaltung bereits bei einer Kondensatorspannung von 0,7 V kippt, vernachlässigen wir das nichtlineare Verhalten und rechnen mit dem arithmetischen Mittel:

$$U_C = \frac{U_V \cdot t}{R \cdot C} \Rightarrow t = \frac{U_C \cdot R \cdot C}{U_V}$$

Die Schaltung kippt bei etwa 0,7 V, was bedeutet, dass die durchschnittliche Spannung am Widerstand auf diese Weise etwa 11,65 V beträgt.

$$t = \frac{0,7 \text{ V} \cdot 100 \text{ k}\Omega \cdot 100 \text{ }\mu\text{F}}{11,65 \text{ V}} = 0,6 \text{ s}$$

Der astabile Multivibrator

Ein astabiler Multivibrator hat keinen stabilen Zustand, sondern schaltet ständig zwischen den beiden möglichen Zuständen hin und her - daher sein Name. Die Schaltung in **Bild 3** sieht einem doppelten Monoflop ziemlich ähnlich, so dass man versucht ist, die vom Monoflop abgeleiteten Formeln für die Zeitmessung zu verwenden - doch damit läge man etwa um den

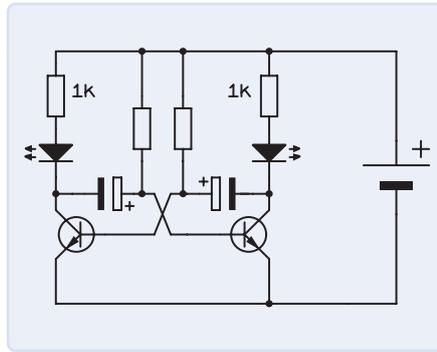


Bild 3. Der astabile Multivibrator.

Faktor 10 falsch. Das wirft natürlich die Frage auf, wie diese Schaltung eigentlich funktioniert.

Nehmen wir an, dass der linke Transistor leitet: Dann ist die U_{CE} des linken Transistors etwa gleich Null und die des rechten Transistors etwa gleich der Versorgungsspannung. Der linke Kondensator wird über seinen Lastwiderstand langsam aufgeladen. Zu einem bestimmten Zeitpunkt erreicht die Spannung am Kondensator 0,7 V und der rechte Transistor schaltet durch.

Wenn der rechte Transistor einschaltet (leitend) wird, ist die Änderung der U_{CE} ungefähr gleich der Höhe der Versorgungsspannung, und das bedeutet, dass das Kondensatorpotenzial um diesen Betrag abnimmt: Die Basis des linken Transistors führt also ein negatives Potenzial und der Transistor sperrt.

Das bedeutet aber, dass der Kondensator nicht von Null an aufgeladen wird, sondern von einer Spannung, die ungefähr der Versorgungsspannung entspricht, aber mit einem Minuszeichen davor. Und das braucht natürlich viel mehr Zeit.

Rechnen wir das mal durch. Der Kondensator muss also von -12 V auf +0,7 V aufgeladen werden. Wenn wir dabei von einer linearen Ladekurve ausgehen, ergibt sich ein Fehler von etwa 10 % (das entspricht in etwa der Toleranz eines Elektrolytkondensators vernünftiger Qualität). Wir wollen hier versuchen, etwas genauer zu sein. Für die Ladung eines Kondensators gilt:

$$U_C = U_0 \cdot \left(1 - e^{-\frac{t}{RC}} \right)$$

In diesem Fall ist U_0 gleich der doppelten Versorgungsspannung, und der Kondensator ist auf $U_V + 0,7 \text{ V}$ geladen:

$$U_C + U_V = 2 \cdot U_V \cdot \left(1 - e^{-\frac{t}{RC}} \right)$$

Diese Gleichung können wir nach t auflösen (wir werden Sie nicht mit der Herleitung belästigen) und erhalten als Ergebnis:

$$t = -R \cdot C \cdot \ln \left(1 - \frac{U_C + U_V}{2 \cdot U_V} \right)$$

Rechnet man dies mit den gleichen Bauteilwerten aus dem beim Monoflop ($R = 100 \text{ k}\Omega$ und $C = 100 \text{ }\mu\text{F}$) durch, so erhält man $t = 7,5 \text{ s}$, also eine zwölfmal längere Zeit als beim Monoflop.

Der Bipolartransistor als Verstärker

Ein Bipolartransistor ist eigentlich kein besonders „dankbares“ Bauteil, um einen Verstärker zu konstruieren: Alle Parameter hängen voneinander ab, keiner ist wirklich linear, und außerdem ist alles recht „eklig“ temperaturabhängig - wenn man nicht gegensteuert, fliegt einem der Arbeitspunkt von vornherein um die Ohren.

Aus diesen Gründen werden individuelle Transistoren nur noch selten für Kleinsignalanwendungen eingesetzt; stattdessen kommen vorzugsweise integrierte Operationsverstärker (Opamps) zum Einsatz. Opamps sind zwar auch mit Transistoren aufgebaut, aber die meisten der oben genannten Probleme wurden von den Entwicklern der Operationsverstärker bereits kompensiert, so dass wir mit ihnen auf sehr einfache Weise stabile Verstärker bauen können.

Operationsverstärker sind allerdings nur für den Kleinsignalbetrieb geeignet: Für Leistungsverstärker sind die maximale Ausgangsspannung und der maximale Strom meist zu gering. Opamps können in solchen Fällen mit diskreten (Leistungs-) Transistoren „geboostet“ werden - oder wir bauen die gesamte Schaltung von Grund auf mit einzelnen Transistoren auf.

Auf Opamps werden wir in einer späteren Folge eingehen. Um sie besser zu verstehen, befassen wir uns hier aber kurz mit dem Transistor als verstärkendes Bauelement. In den meisten Lehrbüchern findet man zu diesem Punkt viel Theorie und noch mehr Gleichungen, die aber mit der praktischen Entwicklung nicht viel zu tun haben. Die einzige wirklich wichtige Beziehung ist

die U/I-Kennlinie der Basis-Emitter-Diode, die wir schon bei der Diode kennengelernt haben - siehe **Bild 4**. Damals haben wir allerdings vergessen zu erwähnen, dass die U/I-Kennlinie temperaturabhängig ist. Wenn sich die Diode erwärmt, dann steigt (bei gleicher Spannung) der Strom an (und die Diode wird noch heißer). Dieses Verhalten ist bei einer gewöhnlichen Diode selten problematisch, aber bei einem Transistor ist es besonders schlimm, weil sich dadurch auch der Kollektorstrom ändert - und zwar um einen Faktor von etwa 100 mehr. Das liegt daran, dass der Kollektorstrom:

$$I_C = I_B \cdot h_{FE} = I_B \cdot \beta$$

ist. Mit h_{FE} oder β wird die Stromverstärkung des Transistors bezeichnet. Diese Stromverstärkung unterliegt jedoch nicht nur einer großen exemplarischen Streuung, sondern hängt (zu allem Übel) auch noch vom Kollektorstrom ab. Bei Kleinsignaltransistoren liegt der Wert von h_{FE} irgendwo zwischen 100 und 800, bei Leistungs- transistoren zwischen 10 und 100. Da die Stromverstärkung ein so unzuverlässiger Parameter ist, sollten wir versuchen, unsere Transistorschaltungen so zu gestalten, dass der genaue Wert von h_{FE} keine (große) Rolle spielt. Wir werden in der nächsten Folge sehen, wie das geht. **◀**

RG — 230033-02

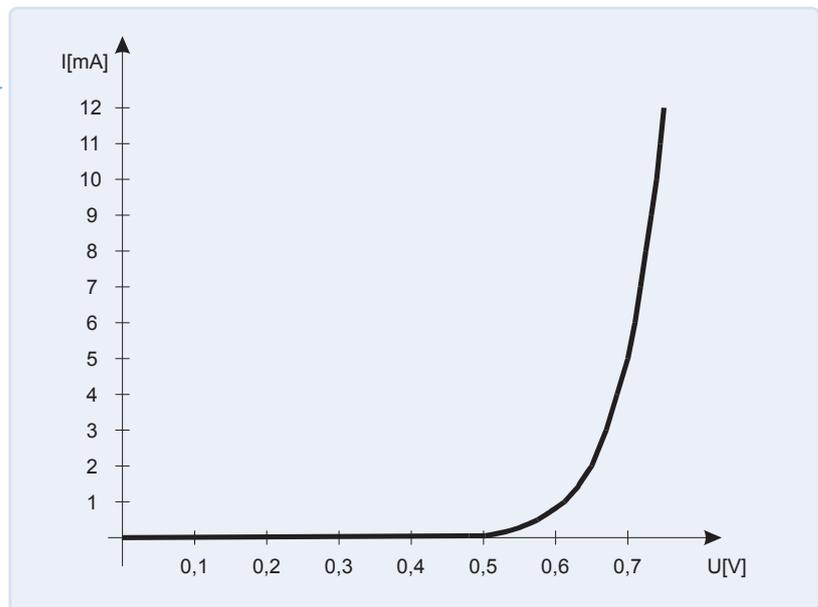


Bild 4. Die U/I-Kennlinie einer Diode.

Die Artikelreihe „Aller Anfang ...“ gründet auf dem Buch „Basiskurs Elektronik“ von Michael Ebner, erschienen im Elektor-Verlag.

Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Senden Sie eine E-Mail an die Elektor-Redaktion unter redaktion@elektor.de.



Passende Produkte

- > **Michael Ebner, Basiskurs Elektronik**
E-Buch, PDF, SKU 19155:
www.elektor.de/19155
- > **B. Kainka, Elektronik-Grundlagen und Einsteiger-Projekte (Elektor 2019)**
Buch, kartoniert, SKU 19035:
www.elektor.de/19035
E-Buch, PDF, SKU 19036:
www.elektor.de/19036

YOUR KEY TO CELLULAR TECHNOLOGY

© eiSmart



**WURTH
ELEKTRONIK**
MORE THAN
YOU EXPECT

**WE meet @
PCIM Europe**
Halle 6-217

Adrastea-I ist ein Cellular-Modul mit hoher Leistung, extrem niedrigem Stromverbrauch, Multi-Band LTE-M und NB-IoT-Modul.

Trotz seiner kompakten Größe verfügt das Modul über integriertes GNSS, integrierten ARM Cortex M4 und 1 MB Flash-Speicher für die Entwicklung von Benutzeranwendungen. Das Modul basiert auf dem leistungsstarken Sony Altair ALT1250 Chipsatz. Das von Deutsche Telekom zertifizierte Adrastea-I-Modul ermöglicht eine schnelle Integration in Endprodukte ohne zusätzliche branchenspezifische Zertifizierung (GCF oder Betreiberzulassung, sofern eine Deutsche Telekom IoT-Konnektivität (SIM-Karte) verwendet wird. Für alle anderen Betreiber bietet das Modul bereits die branchenspezifische Zertifizierung (GCF) an.

www.we-online.com/gocellular

- Kompakte Größe
- Lange Reichweite/weltweite Abdeckung
- Sicherheit und Verschlüsselung
- Multiband Unterstützung

Projekt 2.0

Korrekturen, Updates und Leserbriefe

Redaktion: Jens Nickel



Grow It Yourself

Elektor Arduino Ausgabe 2022, S. 54 (220414)

Ich habe Ihren Artikel in Elektor mit Interesse gelesen. Ich habe in der Vergangenheit Samen gepflanzt, war aber nicht sehr erfolgreich, weil ich es versäumt habe, die liebevolle Pflege zu geben, die Samen brauchen. Am Ende habe ich meine Tischgewächshäuser an meine Freundin verschenkt, die sich mehr für Pflanzen und deren Pflege interessiert. Jedenfalls hat mir Ihr Artikel sehr gut gefallen, und ich möchte für meine Freundin etwas Ähnliches bauen.

Sie erwähnen, dass Pflanzen 450 nm und 650 nm Licht mögen. RGB-LEDs erzeugen Wellenlängen von 460 nm, 525 nm und 625 nm (schnelle Google-Suche). Diese Werte stimmen nicht ganz mit den Anforderungen der Pflanzen überein. Sind sie nahe genug dran?

Elektor hat bereits 2019 zusammen mit Würth Elektronik einen Artikel über ein LED-beleuchtetes Gewächshaus veröffentlicht (<https://www.elektormagazine.de/magazine/elektor-109/51129>). Ihr Artikel hat mich inspiriert, dieses Thema weiter zu verfolgen. Einen Ultraschallnebelgenerator habe ich schon seit vielen Jahren hier herumliegen, vielleicht wird er ja endlich mal nützlich.

Michael Hompus

Vielen Dank für Ihre Frage! Vor einiger Zeit, nachdem ich das GIY-Projekt online veröffentlicht hatte, habe ich begonnen, das Thema LED-Licht für das Pflanzenwachstum zu erforschen, und Sie können meine Zusammenfassung dazu gerne hier lesen:

www.instructables.com/DIY-Grow-LED-Light-Designing-a-Better-Sun/?amp_page=true

Um Ihre Frage zu beantworten und Ihnen bei der praktischen Anwendung des neuesten Know-hows in der Branche zu helfen: Verwenden Sie weiße Farb-LEDs plus rote oder fernrote LEDs. Die Idee ist, ein komplettes Spektrum zu erzeugen, das dem der Sonne ähnelt, und dann die Effizienz zu erhöhen, indem man einige zusätzliche rote/fernrote/UV-LEDs hinzufügt.

Für Weiß würde ich die Samsung-Dioden LM301B oder LM301H empfehlen, für Rot Osrams hyper-rote LEDs. In der Praxis wird es etwa so aussehen: 6 bis 10 Stück weiße LEDs plus zwei rote LEDs auf einem Phyto-Lichtbalken. Und dann spielen Sie einfach mit der Farbtemperatur der weißen LEDs herum - wenn Sie kaltes Weiß wählen, erhalten Sie mehr Blau im Spektrum, wenn Sie warmes Weiß wählen, erhalten Sie mehr Rot im Spektrum, und so weiter. Ich hoffe, der Grundgedanke ist klar. Besuchen Sie www.cityfarm.md, um die endgültige Version des GIY-Projekts zu sehen.

Dmitrii Albot (Autor des Artikels)



Preiswerter Audio-Messplatz

Elektor 7-8/2022, S. 6 (200604)

In der DIY-Community ist die Software REW sehr weit verbreitet:

<https://www.roomeqwizard.com/>

REW ist grundsätzlich kostenlos, wobei man dem Autor etwas spenden kann/sollte. Die Software sollte alles können, was auch in dem Artikel genannt wurde. Für deutlich darüber hinaus gehende Ansprüche gibt es noch eine Pro-Version, die dann allerdings 100 € kostet.

Auf diyaudio.com gibt es einen langen Thread zu REW, auch in Verbindung mit verschiedenen Interfaces und deren Modding.

Frank von Zeppelin



The Tube

Elektor 1-2/2023, S. 70 (220089)

Ich bin schon seit vielen Jahren Abonnent. Mein erstes Projekt aus Elektor war ein UKW-Tuner mit gedruckten Spulen (1975), da war ich etwa 20 Jahre alt. Ich finde es gut, dass ihr euer Heft-Konzept und alle Angebote drum herum im Laufe der vielen Jahrzehnte angepasst habt an den tatsächlichen Bedarf. Macht weiter so! Im neuesten Heft hat mich allerdings einiges am Artikel „The Tube“ gewundert ... vielleicht sind es nur Tippfehler? Oder verstehe ich etwas falsch?

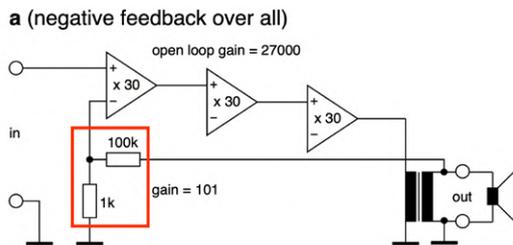


Bild 1a: Die Erklärung im Text suggeriert indirekt, dass in den einzelnen Stufen keine Gegenkopplung angewandt wird (z.B. durch Kathoden-Widerstände). Ich kann mir kaum vorstellen, dass es solche Ansätze tatsächlich gibt! Falls doch, dann hätte man auf diesen exotischen Sonderfall hinweisen müssen, oder?

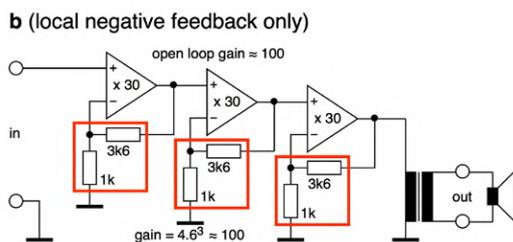


Bild 1b: Die Gesamtverstärkung (Gain) sollte doch mit den genannten Widerstandswerten eher $4,3^3 = 79,5$ sein? (Außerdem ist $3,3^3$ rund 36, aber niemals 100.) Und darf man hier überhaupt von einer *Open Loop Gain* sprechen? Besser wäre „Gesamtverstärkung des Verstärkerzuges“, denke ich.

Harald Sonnemann

Sie haben völlig Recht damit, dass es keinen einzigen praktikablen Audio-Verstärker geben kann, der wirklich ohne Gegenkopplung auskommt, da die verstärkenden Bauteile dafür nicht linear genug sind. Oft wird bei Röhrenfans der Begriff „ohne Gegenkopplung“ unsauber synonym für „ohne Gegenkopplung über alles“ verstanden.

Was ich gemacht habe, ist „einer Gegenkopplung über alles“ gerade das Prinzip der lokalen Gegenkopplung pro Verstärkungsstufe gegenüberzustellen, siehe Bild 1 im Artikel.

Im Eifer des Gefechts habe ich dabei tatsächlich falsch gerechnet. $4,3^3$ ist 79 und nicht 100; und definitiv ist 100 nicht $3,3^3$. Weder mir beim Korrekturlesen noch dem Autor bei der Abnahme ist das aufgefallen. Anbei ein korrigiertes Schaltbild.

Mit *Open Loop Gain* in Bild 1b ist dann natürlich eine Quasi-Open-Loop-Gain gemeint.

Thomas Scherer (Redakteur des Artikels)



Aller Anfang...

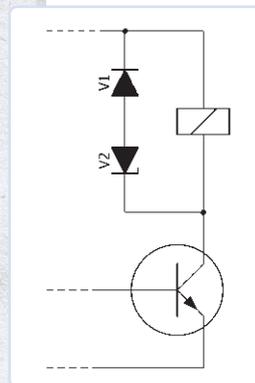
Elektor 9-10/2022, S. 34 (220256)

Projekt 2.0

Elektor 1-2/2023, S. 110 (220601)

Leider hat sich in meinem Vorschlag, der als Leserbrief im Januarheft 2023 veröffentlicht wurde, ein Fehler eingeschlichen. Ich hatte angeregt, beim Schalten eines Relais mit einem Transistor eine Z-Diode in Reihe mit der Freilaufdiode zu schalten. Das verkürzt nicht nur die Abschaltzeit des Relais signifikant, sondern trägt auch zur Lebensdauer seiner Kontakte bei, da die Kontakttrennung erheblich schneller vonstatten geht.

Leider wurde die Z-Diode in der dazugehörigen Zeichnung verkehrt herum dargestellt, sodass sich lediglich die beiden Vorwärtsspannungen der beteiligten Dioden überlagern, was nicht viel bringt. Im Bild ist die Z-Diode richtig herum geschaltet.



Wenn der Transistor abschaltet, addiert sich die Rückwärtsspannung der Z-Diode V2 zur Vorwärtsspannung der Gleichrichterdiode V1. Dadurch baut sich der Strom durch das Relais viel schneller ab, was man nicht nur messen, sondern auch deutlich hören kann.

Dabei muss beachtet werden, dass die Spannung zwischen Kollektor und Emitter des Transistors die Summe aus Betriebsspannung, Rückwärtsspannung der Z-Diode und Vorwärtsspannung der Gleichrichterdiode ist. Bei 12 V Betriebsspannung und einer 10-V-Z-Diode sieht der Transistor im Abschaltmoment also etwa 22,6 V, was bei der Auswahl des Transistors und der Z-Diode berücksichtigt werden muss.

Thomas Klingbeil

Das neue I³C-Protokoll

Würdiger I²C-Nachfolger oder viel Rauch um Nichts?

Von Tam Hanna (Slowakei)

Kaum ein Bus-Protokoll ist so weit verbreitet wie der einst in den 90ern von Philips Semiconductor auf die Menschheit losgelassene I²C-Bus. Im Laufe der letzten Jahre wurden die Schwächen dieses Bussystems allerdings offensichtlich. Mit I³C steht ein Nachfolge-System am Start, das wir Ihnen in diesem Artikel en détail vorstellen wollen.

Zum Verständnis der Möglichkeiten und Grenzen müssen wir uns daran erinnern, dass der I²C-Bus erst seit dem Auslaufen der von Philips gehaltenen Patente eine komplett freie Technologie ist. Zwar hat NXP als Nachfolger der Halbleitersparte von Philips noch geistige Eigentumsrechte am Logo und dem Namen, aber sonst kann die Technologie von jedermann frei verwendet werden.

Auch wenn die Debatte um I²C vs. TWI mittlerweile so gut wie verstummt ist, finden sich im Internet, beispielsweise unter [1], noch interessante, nach folgendem Schema aufgebaute Hinweise: „*The concept behind the license fee was to insure that people who claimed they were making I²C silicon (and selling it as I²C silicon) were supporting the standard and thus helping to grow the standard.*“

Diese sich mit der Erinnerung des Autors deckenden Ergebnisse zeigen, dass das Ziel von Philips immer war, durch die Erhebung der Lizenzgebühren die Lizenznehmer zu animieren, den I²C-Standard auch vollständig zu implementieren. Darunter verstand man explizit auch ein Glitch- beziehungsweise Spikefilter, das in der Vergangenheit ein häufiger Streitpunkt zwischen Philips Semiconductor und Drittherstellern war. Dieses Filter (wir kommen hierauf noch zurück) dürfte einer der Gründe sein, warum Philips einst überhaupt mit der Eintreibung von Lizenzgebühren begann. Mit dem Auslaufen der Philips-Patente war allerdings die freie Jagd eröffnet. Die Interoperabilität von I²C-Geräten in einem I³C-Bussystem kann deshalb nur als Best

Level Annual Turnover	1st Quarter Fees (Jan, Feb, Mar)	Annual Fee
Adopter > USD \$250m	\$8,000	\$8,000
Adopter ¹ < USD \$250m	\$4,000	\$4,000
Contributor > USD \$250m	\$32,000	\$32,000
Contributor ¹ < USD \$250m	\$16,000	\$16,000
Contributor ² < USD \$10m	\$8,000	\$8,000
Board (and Promoter)	\$40,000	\$40,000

Bild 1. Die Teilnahme an der MIPI ist durchaus teuer (Bildquelle: [3]).

Practice oder frommer Wunsch angesehen werden. Praktische Tests solcher Mischsysteme sind in jedem Fall erforderlich. Doch schauen wir uns zunächst an, wer oder was hinter dem I³C-Standard steckt!

I³C und die MIPI Alliance

Hinter dem I³C-Standard steht ein als *MIPI Alliance* bezeichnetes Standardisierungsgremium [2], das diverse Standards für allgemeine Elektronik anbietet. Für die Mitgliedschaft fallen nicht unerhebliche Gebühren an, wie die gut versteckte Gebührenliste [3] zeigt (**Bild 1**). Im Interesse maximaler Adoption bietet das Standardisierungsgremium im Fall von I³C eine Basis-Variante der Spezifikation

komplett kostenlos an. Unter [4] findet sich eine Liste „fortgeschrittener“ Funktionen, deren Nutzung auf jeden Fall eine Mitgliedschaft in der MIPI voraussetzt. Auf dieser Seite erfahren wir auch, dass die Abkürzung I³C für *Improved Inter Integrated Circuits* steht. Nach Ansicht des Autors dürfte man sich bei der Nutzung von I³C in einer kommerziellen Schaltung nur dann „Ärger“ mit der MIPI einhandeln, wenn das Interface auf die eine oder andere Art selbst realisiert wird (Stichwort FPGA oder Bit Banging). Verwenden Sie hingegen ausschließlich gekaufte Bausteine (und nutzen das I³C-Logo und den Namen nicht in der Außenwerbung), so dürften keinerlei Probleme zu erwarten sein.

Abschied vom Open Drain ...

Der I²C-Bus erfreut den Entwickler seit jeher mit Ärgernissen: Erstens ist immer fraglich, wo die für das Aufladen der Bus-Kapazität notwendigen Pull-Up-Widerstände untergebracht werden. Zweitens ist es im Allgemeinen nicht möglich, vom nicht-taktgenerierenden Baustein aus Interrupts auszulösen. Möchte man einem „takt schnüffelnden“ Gerät (siehe **Kasten**) die Möglichkeit zum Absenden von Interrupts geben, so ist bei I²C in jedem Fall eine zusätzliche Verbindung erforderlich. Hat man einige Sensoren im Feld, so geht dies auf Kosten des GPIO-Budgets beim taktemittierenden Baustein.

Nettigkeit Numero drei ist die vergleichsweise geringe Geschwindigkeit: Philips nennt (bei einer Arbeitsfrequenz von 3,4 MHz) eine Wortrate von 3 Mbps. Daraus folgt, dass die gemächliche Vorgehensweise der Transceiver zu einem hohen Energieverbrauch führt, denn wenn der Datenaustausch zwischen zwei Geräten länger dauert, können diese auch erst später in einen energiesparenden Stand-by-Modus fallen. Das Diagramm (**Bild 2**) der MIPI ist nach Ansicht des Autors recht glaubhaft. Zu guter Letzt besitzen I²C-Geräte die fatale Möglichkeit, den Bus als Ganzes „auf ewig“ abzuschalten. Ein entsprechendes Schaltdetail wie in **Bild 3** findet sich in mehr als nur einem I²C-System!

... bringt Verbesserungen in der I3C-Welt

Die erste Änderung von I3C ist, dass sowohl SCL als auch SDA in den meisten Fällen im Push-Pull-Betrieb arbeiten. SCL ist dabei immer ein Push-Pull-Pin, während der SDA-Pin in manchen Zuständen auch in den Open-Drain-Betrieb wechselt. Zur Vermeidung diverser Probleme bei der Platzierung der Pull-Up-Widerstände gibt der Standard an dieser Stelle vor, dass diese stets im taktgenerierenden Gerät unterzubringen sind.

Wichtig ist dabei, dass SCL immer (Sonderfälle wollen wir an dieser Stelle auslassen) und ausschließlich vom taktgenerierenden Element gesteuert wird. Sofern sich mehrere potentielle Taktgeneratoren in einem Bus befinden, obliegt das Schalten des SCL-Pins dem Taktgenerator, der im aktuellen Zustand aktiv ist.

Das im klassischen I²C-Bus mögliche Clock Stretching durch den Slave wird in der Spezifikation von I3C nicht unterstützt. Deshalb ist der Betrieb solcher Bausteine in einem gemischten Bus nicht möglich. Andererseits sind dem Autor in der Praxis nur höchst wenige Anwendungsfälle des Clock Stretchings bekannt.

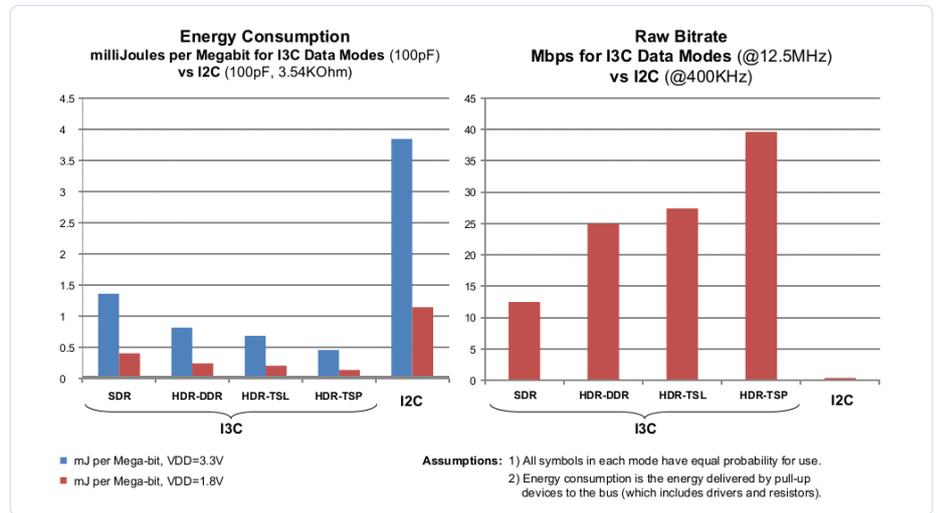


Bild 2. Analog zum Mobilprozessor: Geschwindigkeit spart Energie! (Bildquelle: [8])

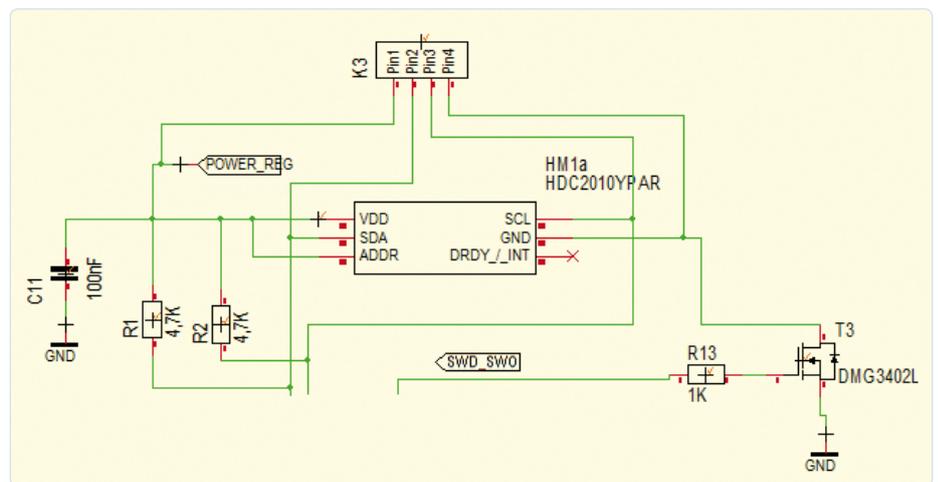


Bild 3. Der Not-Aus-Transistor T3 bewirkt eine „harte Abschaltung“ aller an den Bus angeschlossenen Sensoren, um ein „blockierendes“ Gerät in seinen Ausgangszustand zurückzusetzen.

Mit dieser Vereinfachung geht auch eine Vereinfachung der I/O-Pins auf Seiten der Mikrocontroller beziehungsweise der Taktgeneratoren einher. Während I²C wegen des in der Einleitung schon erwähnten 50-ns-Spike-Filters dedizierte GPIOs benötigt, setzt die MIPI-Spezifikation gewöhnliche GPIO-Pins voraus, die nur zum Treiben von 4 mA Strom befähigt sein müssen. In der Spezifikation ist explizit auch angeführt, dass die im Fall von I²C oft „haarigen“ Spikefilter nicht erforderlich sind.

Der Lohn der Schnelligkeit

Im nächsten Schritt wollen wir uns der eigentlichen Signalform zuwenden. Das in diesem Artikel schon mehrfach erwähnte Glitchfilter kommt auch bei I3C insofern zum Einsatz, als das SCL-Signal in einem I3C-Bus nicht mehr wie bei I²C ein Tastverhältnis von 50 % aufweist. Die High-Periode ist stattdessen mit kürzer als 45 ns deklariert (**Bild 4**), was logischerweise dazu führt, dass ein mit dem Standard konform implementierter I²C-Chip

mit diesen Wellenformen nichts anzufangen weiß.

Nach dem Eintreffen einer solchen „unsichtbaren“ SCL-Transaktion kann der Taktgenerator seinen SDA-Pin beziehungsweise die SDA-Leitung ebenfalls in den Push-Pull-Modus umschalten und die Arbeitsfrequenz des Busses auf bis zu 12,5 MHz erhöhen.

In einem derartigen Vollast-Betrieb wird laut MIPI-I3C-Spezifikation eine Übertragungsrate von bis zu 12,5 Mbps erreicht.

Höhere Übertragungsraten erreicht I3C in einem der drei HDR-Modi (High Data Rate), die allerdings nur in den zwei Sonderfällen verwendet werden dürfen, nämlich einerseits im als *Pure I3C Bus* bezeichneten Bus-System, das ausschließlich aus I3C-Bausteinen besteht, andererseits im so genannten *Mixed Fast Bus*, an den auch I²C-Bausteine angeschlossen sein dürfen, die dann allerdings - Sie hätten es fast erraten - eine spezifikationsgemäße Implementierung des Glitchfilters aufweisen müssen.

Die HDR-Modi werden durch eine spezielle

MIPI I3C Bus signal in SDR mode after dynamic address assignment

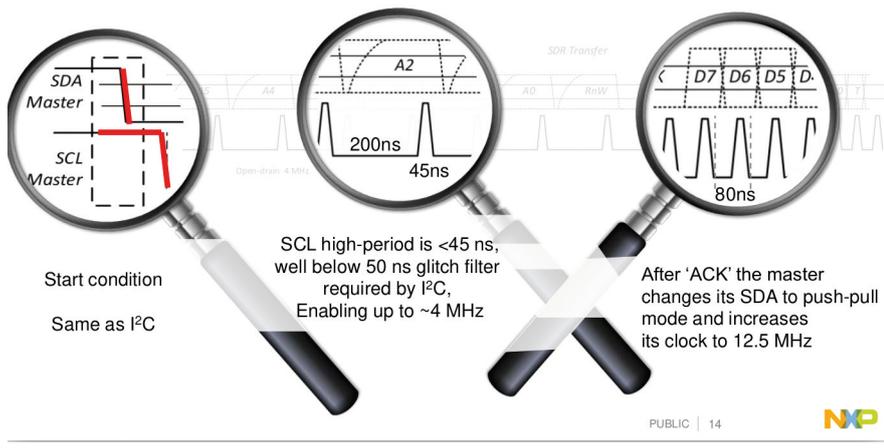


Bild 4. I²C-Bausteine betrachten diese Takt-Flanken als „runter“ (Bildquelle: NXP).

5 Pin Configuration and Functions

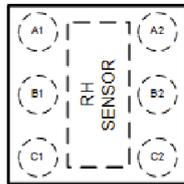


Figure 5-1. WLCSP (DSBGA) 6-Pin YPA Top View

Table 5-1. Pin Functions

PIN		I/O TYPE ⁽¹⁾	DESCRIPTION
NAME	NO.		
VDD	A1	P	Positive Supply Voltage
ADDR	B1	I	Address select pin – hardwired to VDD or GND. GND: slave address: 1000000 VDD: slave address: 1000001
GND	C1	G	Ground
SDA	A2	I/O	Serial data line for I ² C, open-drain; requires a pullup resistor to VDD
SCL	B2	I	Serial clock line for I ² C, open-drain; requires a pullup resistor to VDD
DRDY / INT	C2	O	Data ready/Interrupt. Push-pull output

(1) P=Power, G=Ground, I=Input, O=Output

Bild 5. Da der HDC2010 nur einen Adresspin hat, sind pro I²C-Bus nur zwei seiner Gattung erlaubt (Quelle: [9]).

Sequenz aus SCL-Pulsen bei in Ruhe befindlicher SDA-Leitung eingeleitet. Der einfachste Modus ist HDR-DDR, der bis zu 25 Mbps erreicht. Der Trick dabei ist, dass jede Flanke des SCL-Signals einen gültigen Trigger zur Entgegennahme von Informationen darstellt. Der Modus HDR-TSP (dahinter verbirgt sich *Ternary Symbol Pure*) erreicht gut 30 Mbps. Hier wird eine simultane Übertragung von Informationen unter Nutzung der Pins SCL und SDA erreicht. Mit HDR-TSL (kurz für *Ternary Symbol Legacy-inclusive-bus mode*) steht noch ein Sonderregime zur Verfügung, das TSP für Mixed-Busse bekömmlicher macht.

Automatische Adresszuweisung

Ein weiteres Ärgernis bei I²C-Systemen ist, dass die Zuweisung der Adressen der einzelnen Bauteile normalerweise „physisch“ erfolgen muss. Im Fall vieler Bauteile – der in Bild 5 gezeigte HDC2010 ist dafür ein gutes

Beispiel – führt dies dazu, dass der volle im Standard angebotene Adressraum nur teilweise nutzbar ist.

Statt des bei I²C möglichen 10-Bit-Adressmodus bietet I3C als Ersatz eine dynamische Selbstkonfiguration an. Zudem steht mit I3C-Hot-Join ein System zur Verfügung, das ein an USB erinnerndes „Hot Plugging“ von I3C-ICs an den Bus ermöglicht.

Im Hintergrund kommt dabei ein als *Address arbitration* bezeichnetes Verfahren zum Einsatz, das sich um die Zuweisung von Adressen an Slaves kümmert. Hierzu enthalten I3C-Endgeräte drei Werte: Neben zwei je acht Bit langen Feldern mit Statusinformationen gibt es eine 48 Bit lange *unique ID*. Die Spezifikation geht dabei davon aus, dass jede der als *Provisional ID* bezeichneten UUIDs in einem Bus einzigartig ist.

Während des Startens des I3C-Busses muss der Taktgenerator alle im Bus befindlichen

Geräte finden und ihnen die als *Dynamic Address* zugewiesenen Adressen zuweisen. Dabei kommt eine Vorgehensweise zum Einsatz, die an den von I²C bekannten Arbitrationsprozess erinnert: Derjenige unprovisionierte I3C-Baustein „gewinnt“, der den niedrigsten UUID-Wert aufweist. Der Taktgenerator führt mehrere Vergaberunden durch, bis alle am Bus befindlichen Geräte eine *Temporary ID* besitzen.

Adoption des Standards durch Hersteller

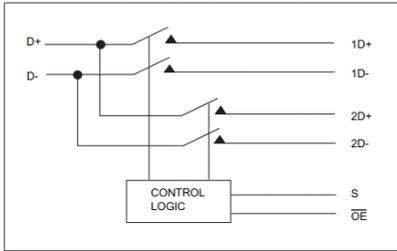
Besucht man die Liste von MIPI-Mitgliedern [5], so findet man ein Who is Who der Halbleiterhersteller; von GigaDevice über ST zu Solomon Systech: So gut wie alle einschlägigen Unternehmen sind vertreten. Vertieft man die Suche allerdings nach praktisch verfügbaren Bauteilen, so wird die Suppe schnell dünn. Bei Mouser finden sich derzeit beispielsweise nur Bauteile wie der von Dialog angebotene PI3CSW12. Bild 6 zeigt, dass sich diese Produkte durch die Bank um Multiplexing und Signalkonditionierung kümmern. Im Hause Renesas entschied man sich mit Chips wie dem IMX3102 zunächst für eine ähnliche Vorgehensweise. Zudem findet sich im Rahmen der Ankündigung folgende Aussage: „I3C's 12.5 MHz speeds significantly outpace current and legacy solutions, such as the I2C 1 MHz speed and analog passive quick switches.“

Sucht man nach Mikrocontrollern mit I3C-Support, so ist die Liste gar sehr kurz: NXP ist mit dem i.MX RT685 derzeit der einzige Spieler auf dem Markt. Unter der Nummer AN12796 findet sich eine Application Note, die die Software illustriert. Außerdem bietet NXP einen teilweise kostenlosen Softcore an, unter [6] gibt es sogar eine Sonderlizenz für Nicht-MIPI-Mitglieder. Ist man bereit, Geld für das Softcore-IP (Intellectual Property) auszugeben, so bieten Anbieter wie Silvaco oder Arasan Chip Systems schlüsselfertige I3C-VHDL-Module an.

Displays mit I3C – in der Spezifikation nennt man diesen Anwendungsfall immer wieder – sucht der geneigte Nutzer derzeit ebenfalls vergebens. Physische Hardware liegt im Moment nur in Form einiger Accelerometer vor: Bosch bietet den BMI263 an, während STM den LSM6DSO und den Drucksensor LPS22HH ins Rennen schickt.

Interessanterweise sieht es im Bereich Software erfreulicher aus: Unter [7] findet man eine vergleichsweise detailliert ausformulierte Beschreibung des im Linux-Kernel implementierten Interfacetreibers.

Block Diagram



Truth Table

S	OE	Function
X	H	Disconnect
L	L	D = 1D
H	L	D = 2D

Bild 6. I3C-Hardware beschränkt sich im Moment vor allem auf Signalkonditionierung (Quelle: [10]).

Wie lange wird es dauern?

Sowohl wegen der höheren Datenrate als auch der Möglichkeit zum Abfeuern von Interrupts ohne dedizierte GPIO-Eingänge dürfte I3C über kurz oder lang Freunde in der gesamten Industrie finden. Fraglich ist allerdings, wie viel Zeit dieser Prozess in Anspruch nimmt. Da es derzeit kaum Mikrocontroller mit passender I3C-Peripherie gibt, wird nach Ansicht des Autors noch einige Zeit ins Land gehen. ◀

RG — 210640-02

Wieso nicht Master und Slave?

Der politisch im Allgemeinen erzkonservative Autor bevorzugt die Bezeichnungen *Taktgenerator* und *Taktschnüffler*. Die didaktische Erfahrung mit Generationen von Kadetten lehrte, dass diese Denkweise zu einem tieferen Verständnis führt.

Wie viele Devices?

Was die Anzahl der Teilnehmer an einem Bus betrifft, ist I3C enger eingeschränkt als I²C, wohl wegen der höheren Geschwindigkeit und auch wegen der geringeren Fähigkeit der Pins, Strom zu liefern. In früheren Versionen des Standards sprach die MIPI davon, nicht mehr als elf Geräte pro Bus zu unterstützen. Mittlerweile findet sich diese (aus einer Simulation elektrischer Charakteristika stammende) Einschränkung nicht mehr im Standard. Von den theoretisch möglichen 2⁷ dynamischen Adressen stehen in Abhängigkeit von der Buskonfiguration rund 90 zur Verfügung.

Sie haben Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Dann kontaktieren Sie bitte den Autor unter tamhan@tamoggemon.com oder die Elektor-Redaktion unter redaktion@elektor.de.



Passende Produkte

- › Vincent Himpe, *Mastering the I2C Bus*
SKU 15385 (Buch): www.elektor.de/15385
SKU 16193 (E-Buch): www.elektor.de/16193

WEBLINKS

- [1] Diskussionen um I²C: www.verycomputer.com/31_b0b5a85949fccc82_1.htm
- [2] MIPI-Seite zu I3C: www.mipi.org/specifications/i3c-sensor-specification
- [3] Die Mitgliedschaft in der MIPI ist teuer: www.mipi.org/join-mipi
- [4] I3C-Basis-Spezifikationen:
<https://resources.mipi.org/blog/mipi-alliance-delivers-new-i3c-basic-specification>
- [5] Liste der MIPI-Mitglieder:
www.mipi.org/devcon//membership/all-member-directory
- [6] Softcore von NXP: <https://bit.ly/3qWAvD3>
- [7] API im Linux-Kernel: www.kernel.org/doc/html/latest/driver-api/i3c/protocol.html
- [8] MIPI I3C White Paper: <http://bit.ly/2gld6BL>
- [9] Texas Instruments Datenblatt HDC2010: www.ti.com/lit/gpn/hdc2010
- [10] Datenblatt Diodes Inc PI3CSW12:
www.diodes.com/assets/Datasheets/PI3CSW12.pdf

Detailgenaue Analyse dank 12-bit Auflösung



SDS2000X HD Digital Oszilloskop

- Mehr Dynamik
- Mehr Signaldetails
- Weniger Rauschen
- Höhere Genauigkeit



www.siglenteu.com
Info-eu@siglent.com

pcim
EUROPE

Nürnberg, 09. – 11.05.2023

Hall 6, Stand 6-449

BlueRC:

IR-Fernsteuerung mit Smartphone und ESP32

Anpassungsfähig und universell

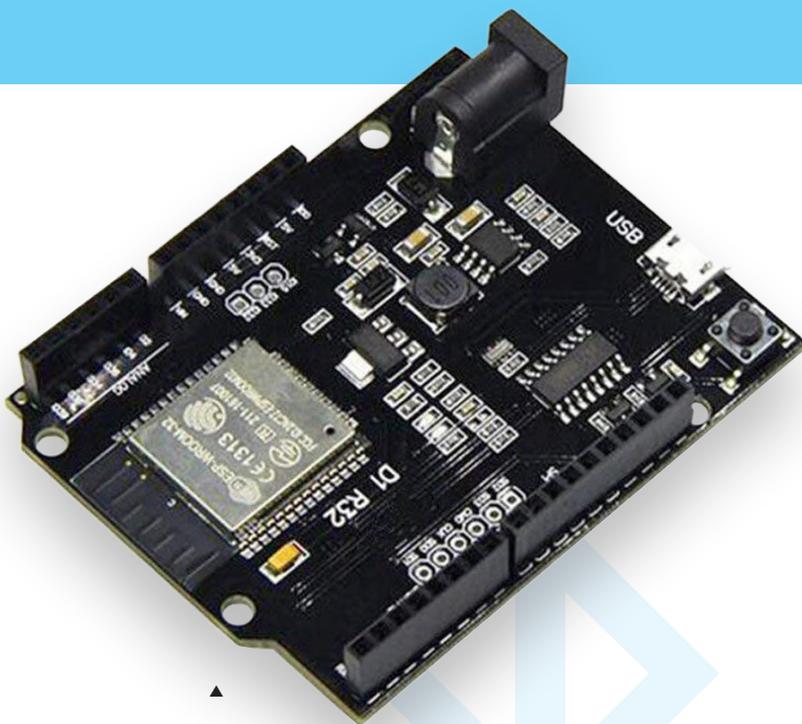


Bild 1. ESP32-WLAN/Bluetooth Entwicklungsboard D1 R32 von WeMos (HW-729 bei Aliexpress).

Von Xin Wang (Deutschland)

Es gab eine Zeit, in der Infrarot-Sender auf Mobiltelefonen üblich waren, aber das ist nicht mehr der Fall. Dieses Projekt bietet eine Lösung: Verwenden Sie Ihr Handy mit seiner Bluetooth-Konnektivität, um ein ESP32-Board zu steuern, das sowohl Infrarotbefehle senden als auch neue erlernen kann.

Irgendwann kam ich zu dem Entschluss, dass ich meine Infrarotgeräte mit meinem Handy steuern wollte. Ich habe einige Projekte im Internet miteinander verglichen: Die meisten von ihnen arbeiteten mit vordefinierten Codes, was bedeutet, dass die IR-Codes im Voraus bekannt sein und in einem Array gespeichert werden mussten. Ich wollte aber eine vollständig lernfähige Fernbedienung bauen, die keine Vorkenntnisse über proprietäre Fernbedienungs-codes erfordert. Die Codes müssen von der Schaltung in Echtzeit und im laufenden Betrieb aktualisiert werden.

BC547 IR transmitter

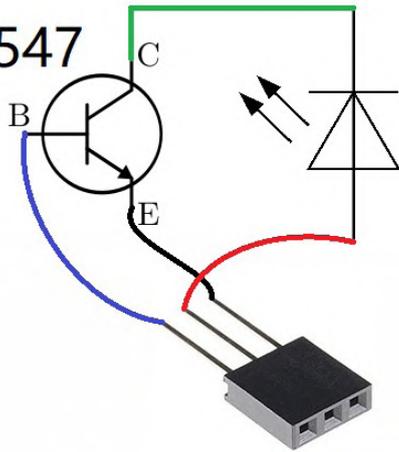
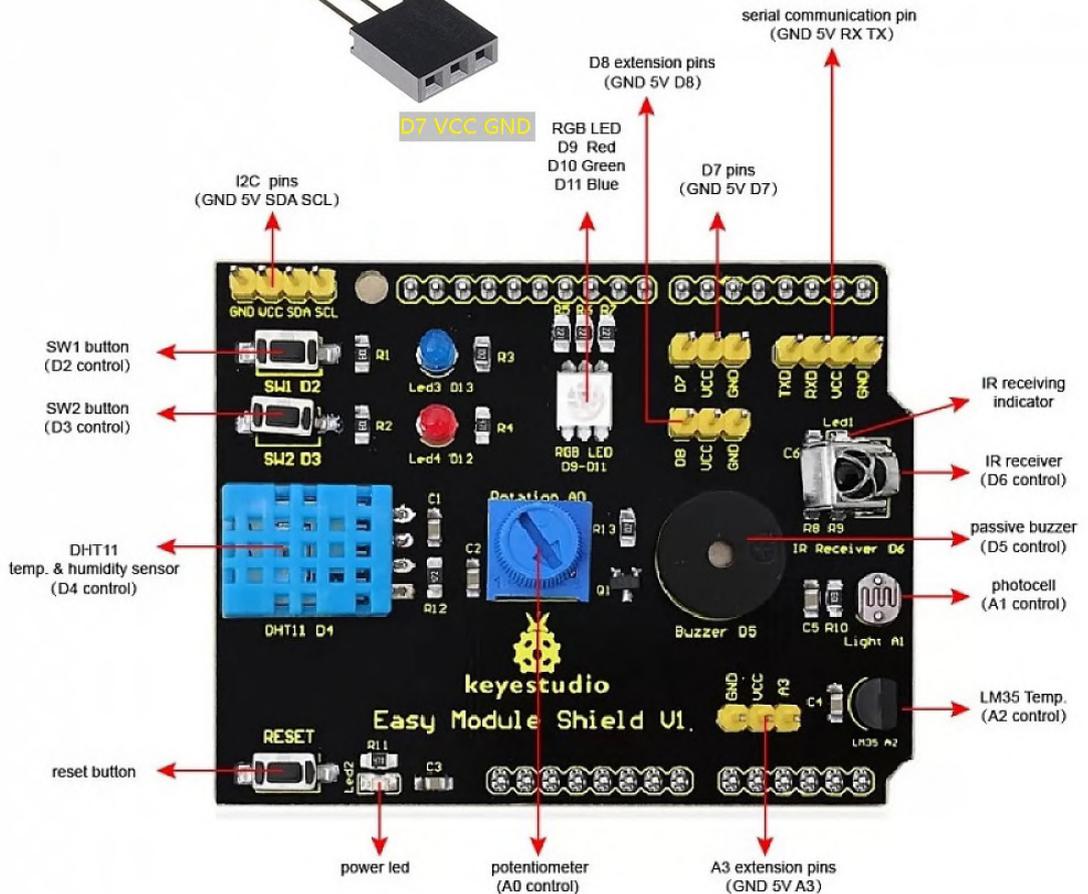


Bild 5. Das Shield und wie ich mein IR-Treiber-/ Sendemodul daran angeschlossen habe. (Quelle: flyrobo.in)



Port-Pin D14 beim Wemos-Board als Ausgang verwendet, um meinen IR-Sender zu steuern. Der IR-Empfänger der Abschirmung ist standardmäßig den Port-Pin D27 als Eingang verwendet. Wie ich das Ganze angeschlossen habe, können Sie in **Bild 5** sehen.

Die Software

Meine Software ist relativ einfach aufgebaut: Der ESP32 fungiert als Bluetooth-Server und eine Android-App als Client. Der ESP32 wartet einfach auf Bluetooth-Nachrichten von der Android-App. Es gibt nur eine minimale Bluetooth-Kommunikation zwischen den beiden Geräten, da die Tastendrucke der Fernbedienung - ebenso wie der Befehl zum Starten der Lernsequenz („N“) - von der Android-App als einzelne Zeichen an den ESP32 gesendet werden.

Zuerst habe ich die Standard-IR-Bibliothek für Arduino ausprobiert. Sie funktionierte gut, wird aber vom ESP32

nicht vollständig unterstützt, da der ESP32 einen proprietären Hardware-Fernbedienungs-Transceiver (RMT) mit einer Menge eingebauter Protokoll- und Modulationslogik für IR-Signale verwendet. Nach vielen Versuchen und fast ebenso vielen Irrtümern habe ich schließlich eine voll funktionsfähige IR-Bibliothek für den ESP32 namens *IRremoteESP8266* von Ken Shirriff et al. gefunden [1]. Vielen Dank, Ken.

Ursprünglich wollte ich die IR-Codes im EEPROM speichern, aber als ich meine Firmware schrieb, wurde mir bewusst, dass der ESP32 ja überhaupt kein EEPROM besitzt. Es gibt jedoch die Möglichkeit, mit der Bibliothek *Preferences.h* [5] Daten dauerhaft im Flash zu speichern. Da meine ursprüngliche Fernbedienung RC5-kodiert war, habe ich nur das RC5-Kodierungsschema implementiert. Möglicherweise müssen Sie den Arduino-Sketch ändern, wenn Ihre Fernbedienung ein anderes Protokoll verwendet.



Programmierung

Für das WeMos-Board können Sie den Arduino-Sketch *BlueRC* unter [2] herunterladen. Es ist ein einzelner *.ino*-Sketch, den Sie mit der Arduino-IDE hochladen können.

Um die Android-App einzurichten, gehen Sie folgendermaßen vor:

1. Laden Sie auf Ihrem Android-Gerät *BlueRC-app-V1.1.apk* von der GitHub-Seite [3] herunter. Sie müssen Ihr Telefon so einstellen, dass das Herunterladen von Apps aus externen Quellen (außer dem Google Play Store) erlaubt ist.
2. Schalten Sie das WeMos-Board ein, auf dem die *.ino*-Firmware BlueRC läuft.
3. Starten Sie die Android-App. Wenn Bluetooth nicht aktiv ist, fordert die App eine Erlaubnis dazu an: Tippen Sie auf *Accept*.
4. Tippen Sie auf das „Kebab-Menü“, die drei vertikal angeordneten Punkte oben rechts und dann *Connect BT Device*. Wenn Sie sich zum ersten Mal mit dem ESP32 verbinden, tippen Sie auf *Scan for devices* und warten Sie, bis das ESP32BlueRC-Shield unter *Other Available Devices* erscheint. Tippen Sie darauf und bestätigen Sie dann die Kopplung.
5. In der Statusleiste oben sollten Sie nun *connected: ESP32BlueRC-Shield* sehen.

Wenn Sie Ihre Hardware einmal gekoppelt haben, müssen Sie den Scan-Vorgang nicht erneut durchführen - tippen Sie in Zukunft einfach auf *ESP32BlueRC-Shield* unter *Paired Devices*.

Wenn die App installiert ist und funktioniert, können Sie nun den Lernprozess durchführen:

1. Tippen Sie lange auf die *LEARN*-Taste auf dem Display und die App antwortet mit „Learning begins, please press a button within 10 seconds...“ Die Android-App sendet dann das Sonderzeichen „^“ (Circumflex) an den ESP32, woraufhin die blaue LED auf dem Shield aufleuchtet.
2. Tippen Sie auf der Benutzeroberfläche auf die Taste, die Sie dem ESP32-Board beibringen möchten. Das Telefon sendet das entsprechende Zeichen an das ESP32-Board, das nun auf Ihre nächste Aktion wartet.
3. Diese besteht darin, die alte Infrarot-Fernbedienung auf das ESP32-Board zu richten und darauf die entspre-

chende Taste zu drücken. Der IR-Code für diese Taste wird automatisch im Speicher des ESP32-Boards abgelegt.

4. Wiederholen Sie Schritt 1 und Schritt 2 für alle Tasten, die Sie aufnehmen möchten.
5. Schließen Sie die App, indem Sie auf *Menu Exit program* tippen.

Das war's. Sie können nun Ihre Infrarotgeräte von Ihrem BlueRC-Gerät aus steuern, indem Sie die App auf Ihrem Android-Telefon verwenden. Für die Fehler-suche am ESP32 ist es am besten, ihn aus- und wieder einzuschalten!

Sie können ein Video des Systems „in action“ [4] auf meinem YouTube-Kanal sehen. ◀

RG — 220103-02



Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Dann wenden Sie sich bitte an die Elektor-Redaktion unter redaktion@elektor.de.



Passende Produkte

- **Dogan und Ahmet Ibrahim, Das offizielle ESP32-Handbuch**
Buch, kartoniert, SKU 18397:
<https://elektor.de/18397>
E-Buch, PDF, SKU 18399:
<https://elektor.de/18399>
- **Koen Vervloessem, Getting Started with ESPHome**
Buch, Paperback, SKU 19738:
<https://elektor.de/19738>
E-Buch, PDF, SKU 19739:
<https://elektor.de/19739>



WEBLINKS

- [1] Bibliothek IRremoteESP8266: <https://www.arduino.cc/reference/en/libraries/irremotesp8266/>
- [2] Arduino-Sketch BlueRC für ESP32: <https://elektor.link/BlueRCESP32>
- [3] Android-App BlueRC: <https://elektor.link/BlueRCApp>
- [4] Das BlueRC-System in Aktion: <https://youtu.be/Kb-TdF84Oz4>
- [5] ESP32 Daten dauerhaft speichern mit der Preferences-Bibliothek: <https://randomnerdtutorials.com/esp32-save-data-permanently-preferences/>

Mikrocontroller-Dokumentationen verstehen

Teil 2: Register und Blockschaltbilder

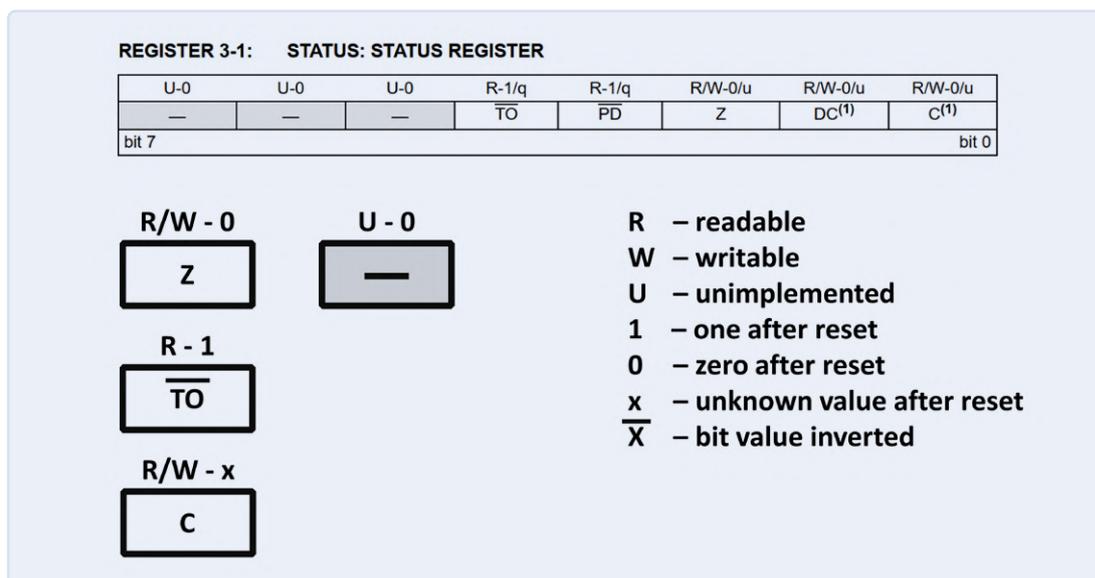


Bild 1. Bit-Beschreibungen von Registern sind manchmal ein wenig kryptisch. (Quelle: Microchip Technology)

Von Stuart Cording (Elektor)

Unabhängig davon, ob man sie liebt oder hasst, muss man sich mit der Dokumentation von Mikrocontrollern auseinandersetzen. Im ersten Teil dieser Artikelserie [1] haben wir uns angesehen, was in einem Mikrocontroller-Datenblatt zu finden ist. In diesem zweiten Teil gehen wir näher darauf ein, wie die Anwender über die Funktionen eines Controllers informiert werden, wobei wir alles von den Registern bis hin zu Peripherie-Blockschaltbildern behandeln.

Datenblatt: Die ersten Seiten

Nachdem Sie nun wissen, was ein Datenblatt enthält, können wir uns einige Details ansehen. Datenblätter scheinen eine recht merkwürdige Art zu haben, Dinge zu beschreiben, und verschiedene Anbieter erstellen ihre Beschreibungen und Zeichnungen auf unterschiedliche Weise. Wir konzentrieren uns hier auf den 8-Bit-Mikrocontroller PIC16F18877 von Microchip Technology [2], und Sie sollten das Datenblatt herunterladen, wenn Sie den Ausführungen folgen wollen.

Wenn Sie das heruntergeladene Datenblatt geöffnet haben, werden Sie mit den ersten beiden Seiten konfrontiert, die eine kurze Beschreibung dieses Mikrocontrollers zusammen mit einer Liste der Hauptfunktionen und der integrierten Peripherie enthalten. Bei diesen *Core Features* handelt es sich nicht um eigenständige Peripherie, sondern um Funktionen, die eng mit dem Prozessorkern verbunden sind. Dazu gehören Funktionen wie die Brown-Out-Schaltung, die erkennt, wenn die Versorgungsspannung unter den erforderlichen Wert fällt, und der Watchdog-Timer, der den

Mikrocontroller zurücksetzt und häufig in sicherheitskritischen Systemen verwendet wird.

Die dritte Seite ist etwas ausführlicher und erklärt, wie viele Peripheriefunktionen wir erhalten und wie viel RAM, Programm-Flash und EEPROM implementiert sind. Ergänzt wird dies durch die Seiten vier bis sieben, die grundlegende Informationen zu den Gehäusebauformen enthalten und erklären, welche Funktionalität welchem Pin zugewiesen ist.

Die Pin-Zuordnungstabellen (*Pin Allocation Tables*) sind ein entscheidendes Element eines jeden Mikrocontroller-Datenblatts. Da so viele Funktionen in die Chips gepackt werden, haben Mikrocontroller heute mehr Funktionen als Pins. Daher kann jedem Pin eine von mehreren Funktionen (und manchmal auch mehr als eine) zugewiesen werden. Bei diesem Controller können wir sehen, dass Pin RA2 als ADC-Eingang, analoge Spannungsreferenz, Komparatoreingang, Digital-Analog-Wandler-Ausgang (DAC) oder als Interrupt-on-Change-Pin verwendet werden kann. Um eine gewisse Flexibilität zu gewährleisten, kann eine bestimmte Funktion auch zwischen einem oder mehreren Pins umgeschaltet werden. Die Zuordnung von Funktionen zu Pins kann daher sehr komplex sein.

Verstehen von Registerbeschreibungen

Registerbeschreibungen bestehen aus zwei Teilen: dem Namen des Registers und den Namen der Bits (oder Gruppen von Bits)

innerhalb des Registers. Seite 38 des Datenblatts beschreibt die Funktionalität des STATUS-Registers als Teil des Prozessors. Dieses Register besteht aus Bits, die *implementiert* sind, und einigen, die *nicht implementiert* sind.

Die implementierten Bits sind mit einer Kombination aus *R* und/oder *W* gekennzeichnet, um anzugeben, ob sie gelesen oder geschrieben werden können. Nach dem Bindestrich folgt normalerweise eine Zahl: „0“ oder „1“. Diese geben den Wert dieses Bits nach einem Reset des Controllers an. Einige Bits können mit einem „x“ gekennzeichnet sein, was bedeutet, dass der Wert unbekannt ist. Der Wert einiger Bits kann davon abhängen, wodurch der Mikrocontroller aus dem Reset geholt wurde. Wurde der Mikrocontroller einfach eingeschaltet, geschieht dies durch einen Power-On-Reset (POR). Wenn die Stromversorgung einen Brown-Out-Reset (BOR) verursacht hat, kann dies auch zu einem anderen Wert in einem bestimmten Bit führen. In diesem Beispiel sind solche „reset-abhängigen“ Bits mit „q“ gekennzeichnet.

Nicht implementierte Bits können ein wenig gefährlich sein. Laut diesem Datenblatt sollte das Lesen der oberen drei Bits eine „0“ ergeben. Es ist jedoch unklar, was passiert, wenn man in diese Bits schreiben würde. Das Datenblatt kann eine allgemeine Empfehlung geben, aber wenn gar keine angegeben ist, sollten Sie vorzugsweise dafür sorgen, dass alle Schreibvorgänge in solche Registern nicht implementierte Bits auf Null setzen. In C/C++ lässt sich dies wie folgt bewerkstelligen:

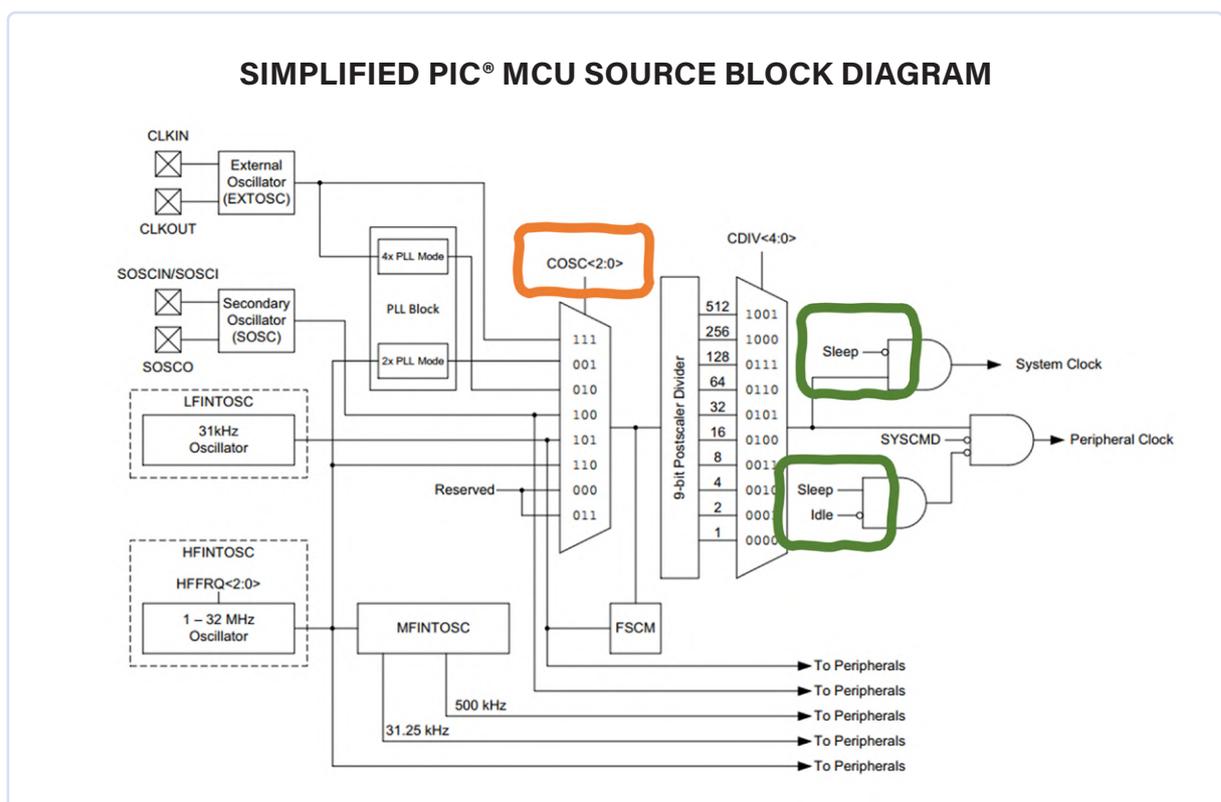


Bild 2. Blockschaltbild der Taktperipherie. Der Taktauswahl-Multiplexer und die Bedienelemente zum Ausschalten der Taktquelle sind hervorgehoben. (Quelle: Microchip Technology)



Unimplementierte Bits auf Null gesetzt

```
myValue &= 0b00011111;    // 3 meistsignifikante Bits zurücksetzen
                          // (Unimplementierte Bits)
STATUS = myValue;        // schreibe "myValue" ins STATUS-Register
```

Es ist sinnvoll, in einem Kommentar im Programmtext zu erläutern, warum dies getan wird, damit niemand auf die Idee kommt, bei einer späteren Revision Ihre Vorsichtsmaßnahme „wegzuoptimieren“!

Ein Strich über einem Bitnamen zeigt an, dass sein Wert invertiert ist, zum Beispiel beim „Time Out“-Bit \overline{TO} . Beim Auftreten einer Zeitüberschreitung könnte man erwarten, dass dieses Eintreten eines Ereignisses mit einer „1“ gekennzeichnet wird. Tatsächlich aber ist die Antwort, die man im Bit vorfindet, eine „0“. Solche negierten Bits haben für den Programmierer wenig Sinn; man muss einfach damit leben und sie in seinem Code angemessen behandeln. Manchmal ist eine Gruppe von Bits erforderlich, um etwas zu konfigurieren, zum Beispiel den Vorteiler für eine Taktquelle oder die Baudrate für eine serielle Schnittstelle. Solche Gruppen von Bits werden dann `BIT_GROUP<X:Y>` genannt, wobei `X` das höchstwertige und `Y` das niedrigstwertige Bit der Gruppe bezeichnet. Außerdem ist Vorsicht geboten: Manchmal kann es vorkommen, dass ein oder mehrere Bits, die zur selben Gruppe gehören, auf verschiedene Bereiche im selben Register oder sogar auf mehrere Register aufgeteilt sind!

Zeit, sich mit dem Timing zu beschäftigen

Der wichtigste Bestandteil des Mikrocontrollers ist seine Taktperipherie. Sie bestimmt, was als Taktquelle für den Prozessor und die On-Chip-Peripherie verwendet wird. Man sollte sich die Zeit nehmen, ihre Struktur zu studieren, wie sie auf Seite 110 beschrieben wird. Normalerweise wird diese Peripherie einmal zu Beginn des Anwendungscodes konfiguriert und nie geändert. Wenn sie nämlich geändert wird, kann sich das auf alle Peripheriebausteine auswirken, von der Abtastrate des ADC bis zur Baudrate der seriellen UART- und CAN-Schnittstellen.

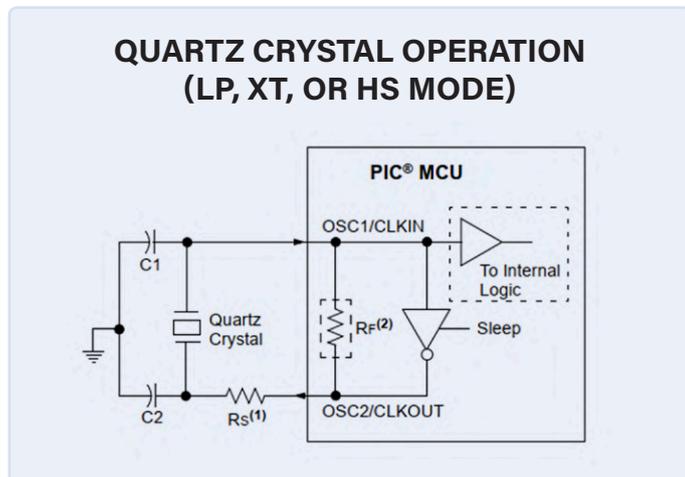


Bild 3. Empfohlene Beschaltung für den Quarzoszillator. (Quelle: Microchip Technology)

In diesem Beispiel sehen wir einige externe Quellen (Quarz- oder Keramikresonator-Oszillatoren) und einige interne Quellen. Die internen Quellen sind möglicherweise nicht genau genug (insbesondere bei Temperaturschwankungen), um als zuverlässige Taktgeber für einige Schnittstellen wie den UART zu fungieren. Das Datenblatt wird dies wahrscheinlich erwähnen. Andernfalls müssen Sie den erlaubten Fehler in dem Abschnitt über die *Electrical Characteristics* überprüfen.

Im Blockschaltbild ist orangefarben eine Gruppe von drei Bits (`COSC<2:0>`) in einem Register markiert, das wahrscheinlich mit dem Oszillator zusammenhängt und das über einen Multiplexer zur Auswahl unter den verfügbaren Taktquellen verwendet werden kann. Aus unbekanntem Grund wird in solchen Diagrammen nur die Gruppe von Bits genannt und nicht das Register, zu dem sie gehören. Der beste Weg, um herauszufinden, zu welchem Register die Gruppe gehört, ist die Suche im Datenblatt-PDF - es kann sein, dass sie sich in einem Register einer nicht verwandten Peripheriefunktion verstecken! In Blockdiagrammen werden Multiplexer-Diagramme häufig verwendet, um anzudeuten, dass Signale geschaltet werden können.

Der Ausgang dieses Blocks liefert die *System Clock* für den Prozessor, und die *Peripheral Clock* für die Peripheriefunktionen. Das Datenblatt kann im weiteren Verlauf auf diese beiden Begriffe verweisen, insbesondere bei der Erörterung von *Low-Power*- oder *Sleep*-Modi. Die grün markierten Blöcke lassen bereits vermuten, dass es einen *Idle*-Modus gibt, bei dem nur die Peripherie vom Takt getrennt wird, und einen *Sleep*-Modus, bei dem die Peripherie und auch der Prozessor vom Takt getrennt werden.

Wenn ein Quarz oder ein Keramikresonator verwendet werden soll, wird in der Regel eine Anleitung zum Aufbau der Schaltung und zum Layout der Platine mitgeliefert. Dies ist auch hier der Fall (Seite 112), wobei zwei Kondensatoren und optional ein Serienwiderstand erforderlich sind. Die Tatsache, dass diesem Blockschaltbild gleich vier Anwendungshinweise zum Oszillatorentwurf beige-fügt sind, sollte als Warnung dienen, dass die Gestaltung eines Quarzoszillators eine gewisse Sorgfalt erfordert.

Über Register und Blockschaltungen hinaus

Damit haben wir die Grundlagen der Beschreibung von Registern und des Blockschaltbilds der Clock-Peripherie behandelt. Wir haben sogar einen flüchtigen Blick auf einen der Oszillatoren geworfen und das Beispielschaltbild mit einem Quarz kennengelernt. Im nächsten Teil dieser Reihe werden wir uns einen weiteren wichtigen Peripherie-Block ansehen, den man gut verstehen muss: den Reset-Block. Wir werden auch nach all den Dingen suchen, die das Datenblatt nicht abdeckt. Wenn Sie auf der Suche nach Ressourcen sind, die Ihnen den Einstieg in die Welt der Mikrocontroller erleichtern, sollten Sie sich einige unserer Einführungsartikel [3] und Bücher ansehen. ◀

Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schreiben Sie dem Autor eine E-Mail an stuart.cording@elektor.com oder die Elektor-Redaktion unter redaktion@elektor.de.



Passende Produkte

- > T. Hanna, *Mikrocontroller-Basics mit PIC* (Elektor 2019)
Buch, kartoniert, SKU 18945: www.elektor.de/18945
E-Buch, PDF, SKU 18946: www.elektor.de/18946
- > A. Pratt, *Programming the Finite State Machine* (Elektor 2020)
Buch, Paperback, SKU 19327: www.elektor.de/19327
E-Buch, PDF, SKU 19328: www.elektor.de/19328

WEBLINKS

- [1] Stuart Cording, „Mikrocontroller-Dokumentationen verstehen (Teil 1)“, elektormagazine.de:
<https://www.elektormagazine.de/magazine/elektor-293/61554>
- [2] 8-Bit-Mikrocontroller PIC16F18877: <https://microchip.com/wwwproducts/en/PIC16F18877>
- [3] Tam Hanna, „PICs programmieren - von der Pike auf“ elektormagazine.de:
<https://www.elektormagazine.de/articles/pics-programmieren--von-der-pike-auf>

MagPi, das offizielle Raspberry Pi Magazin



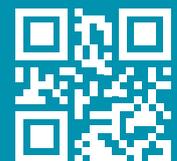
✓ 6 x MagPi als Printausgabe

✓ Zugang zum MagPi Online-Archiv

12 Monate
100+ Projekte
1 Preis

54,95 €

JETZT BESTELLEN AUF
WWW.MAGPI.DE/ABO



MagPi 
www.magpi.de Magazine

Automatisierung der Prüf- und Messtechnik

Programmieren von Prüfgeräten, damit sie tun, was Sie wollen

Von Stuart Cording (Elektor)

Testen kann eintönig und repetitiv sein, und die Fehlersuche ist eine besondere Herausforderung. Warum also nicht die Kommunikationsschnittstellen von Prüf- und Messgeräten nutzen, um die Arbeit zu erleichtern? Hier stellen wir Werkzeuge vor, die sich leicht mit Python steuern lassen, ohne dass man dafür Lizenzen oder teure Software kaufen muss.

Bei den meisten Entwicklern stehen Mess- und Prüfgeräte auf der Werkbank und warten darauf, eine Aufgabe zu erfüllen. Dreht man die meisten dieser Geräte um, findet man auf der Rückseite eine Kommunikationsschnittstelle. In Verbindung mit der entsprechenden Software können Sie damit Messungen steuern und Ergebnisse für die spätere Analyse sammeln. Dies kann bei der Suche nach sporadischen Ereignissen und Fehlern oder beim Testen einer Anwendung über verschiedene Systemparameter äußerst hilfreich sein. Auf diese Weise lassen sich auch End-of-Line-Tests und das Binning, das Einteilen von Komponenten automatisieren.

Fernsteuerung von Laborstromversorgungen

Ein guter Ausgangspunkt ist die Stromversorgung. Für die meisten Anwendungen ist es erforderlich, die Stromversorgung aus- und wieder einzuschalten, um einen Hard-Reset durchzuführen. Bei erweiterten Tests wird die Zielanwendung an den Grenzen der zulässigen Spezifikationen der Eingangsspannungsversorgung betrieben, und es kann auch erforderlich sein, Überspannungstests durchzuführen. Dies ist beispielsweise in der Automobilindu-

strie üblich, wo Impulse von bis zu 87 V für bis zu 400 ms ausgehalten werden müssen (ISO 7637-2). Eine weitere häufige Fehlerart in diesem Zusammenhang ist eine langsam ansteigende oder abfallende Spannung. Unter solchen Bedingungen geraten Schaltungen oft in einen Brown-out-Zustand, von dem sie sich nicht mehr erholen. Schließlich sollten automatisierte Prüfsysteme über Geräte verfügen, die sowohl für die Spannung als auch für den Maximalstrom korrekt konfiguriert werden können, insbesondere wenn dieselbe Testeinrichtung für mehrere verschiedene Produkte verwendet wird.

Die Preise für Labornetzgeräte sind in den letzten Jahren erheblich gesunken. Um dem Marktdruck billiger Geräte zu begegnen, erweitern bekanntere Hersteller in der Regel den Funktionsumfang ihrer Produkte, da sie nicht bereit oder in der Lage sind, allein über den Preis zu konkurrieren. Es ist nicht bekannt, ob dies bei *Aim and Thurlby Thandar Instruments* (Aim-TTI) im englischen Cambridge der Fall ist, aber sie halten ihr Versprechen eines „measurably better value“, wie immer man das interpretieren möchte. Aim-TTI bietet ein umfangreiches Sortiment an Gleichstromversorgungen an, von denen die Einstiegs-



Bild 1. Zwei der Gleichstromversorgungen der Serie EL-R für den Labortisch bieten serielle Schnittstellen (RS-232/virtual COM über USB) zur Unterstützung einer Testautomatisierung. (Quelle: Aim-TTI)

geräte der EL-R-Serie (**Bild 1**) einen näheren Blick wert sind, wenn man mit dem Gedanken spielt, seinen ersten automatisierten Testaufbau zu erstellen.

Diese Tischnetzgeräte verwenden eine rauscharme lineare Regelung und bieten Einzel-, Doppel- und Dreifachausgänge. Sie sind lüfterlos, das heißt konvektionsgekühlt, und haben eine Leistung von 30...130 W. Sie verfügen über eine oder zwei rote LED-Anzeigen und analoge Bedienelemente, einige Modelle auch über Fernbedienungsanschlüsse. Zwei interessante Modelle sind das EL302P mit einem Ausgang und RS-232 sowie das EL302P-USB mit USB, beides 60-W-Geräte mit einem Ausgang von 0...30 V und 0...2 A. Die Geräte werden mit Softwaretreibern für die Schnittstelle geliefert. Darüber hinaus wird auf der Website des Anbieters [1] das Hilfsprogramm „PSU Sequencer“ angeboten, mit dem

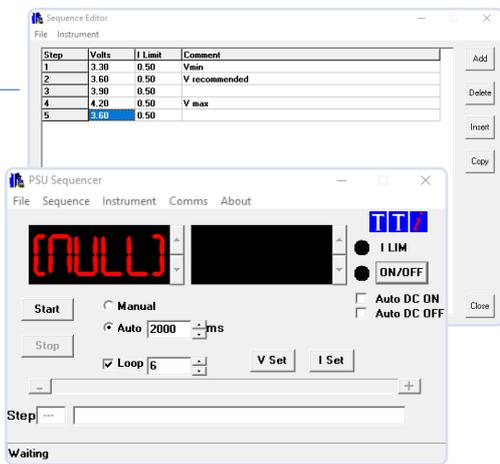


Bild 2. Die Software PSU Sequencer von Aim-TTi ermöglicht einfache, sich wiederholende Zyklen von Spannungs- und Stromereinstellungen. (Quelle: Aim-TTi)

benutzerkonfigurierte Spannungs-/Stromeinstellungen manuell oder automatisch durchlaufen werden können (Bild 2). Außerdem können vorgefertigte Sequenzen aus einer Tabellenkalkulationsdatei importiert werden.

Python für die Steuerung

Die Entwicklung einer eigenen, auf Ihre Prüfanforderungen abgestimmten Steuerungssoftware ist ebenfalls einfach. Die Hardware-RS-232-Schnittstelle arbeitet mit 600 Baud bis 9.600 Baud und die USB-Schnittstelle erscheint als virtueller COM-Port am PC. Alle Befehle sind in der Bedienungsanleitung [2] aufgeführt, und mit ein wenig Planung ist es einfach genug, eine Bibliothek zu entwickeln, die alle Befehle in klar benannte Funktionen oder Methoden umwandelt. Eine Möglichkeit besteht darin, mit einem Arduino und einem RS-232-Transceiver die Stromversorgung zu steuern. Dies hat den Vorteil, dass auch andere Testfunktionen integriert werden können, etwa das Schalten von Signalen mit Hilfe von Relais oder das Erfassen von analogen und digitalen Signalen. Alternativ bietet Python das Modul pySerial [3]. Mit einfachem Code (Beispiel 1) lässt sich die Befehlsschnittstelle als Python-Modul implementieren und eine automatisierte Steuerung realisieren. Mit etwas Lektüre der Dokumentation ist es auch möglich, eine TCP/IP-zu-Serial-Brücke zu implementieren [4]. Diese Funktionalität ist im experimentellen Memo RFC2217 [5] beschrieben und ermöglicht es einem entfernten PC, eine serielle Schnittstelle zu konfigurieren und die Kommunikation zu implementieren.

Signalen mit dem Oszilloskop folgen

Oszilloskope können auch aus der Ferne gesteuert werden. Mit ihrem breiten Spektrum an Fähigkeiten, von der Signalerfassung



Beispiel 1: Einfaches Python-Skript zur Erfassung des PSU-ID-Strings mit pySerial

```
import serial
ser = serial.Serial('/dev/ttyUSB0', 9600, timeout=0)

# open serial port
print(ser.name)
# confirm port was really used
ser.write(b'*IDN?')
# request PSU's ID strings
response = ser.readline()
# collect response
print(response)
# output PSU ID string
ser.close()
# close port
```

analoger und digitaler Signale bis hin zu FFTs, können sie für verschiedene automatisierte Tests verwendet werden. Manche Fehler sind beispielsweise schwer zu finden, weil sie durch eine komplexe Reihe von Ereignissen in einer bestimmten Reihenfolge ausgelöst werden. Sobald Ihr Team herausgefunden hat, wie der Fehler ausgelöst wird, besteht der nächste Schritt darin, das Oszilloskop so zu konfigurieren, dass relevante Signale erfasst werden, die zur Ermittlung der endgültigen Ursache beitragen.

Boards wie Arduino und Raspberry Pi eignen sich hervorragend für die schnelle Entwicklung eines Testumfelds, das den Fehler wiederholt auslösen kann, indem es analoge Ausgänge und digitale Signale verwendet, die bei Bedarf durch ein Relais oder einen MOSFET ergänzt werden. Sie können auch genau getimte Triggersignale für ein Oszilloskop liefern, damit der wesentliche Teil der relevanten Signale für die Analyse gespeichert wird.

Viele Oszilloskope bieten USB- und LAN-Schnittstellen, aber einige unterstützen nur proprietäre Software oder einen Zugang zu einem Webserver für die Konfiguration. Es gibt jedoch eine USB-Klassenspezifikation für Test- und Messzwecke namens USBTMC [6]. Ähnlich wie bei Speichergeräten, zum Beispiel USB-Laufwerken oder Human-Interface-Devices (HID) wie Maus und Tastatur, bietet diese Klasse vordefinierte Befehle, die sich für die Verbindung mit Prüf- und Messgeräten eignen.

Geräte wie das Mixed-Signal-Oszilloskop 2567B (200 MHz, 2 GSa/s) von B&K Precision [7] unterstützen diese Schnittstelle. Mit vier analogen Kanälen, einem 16-Kanal-Digitalport, einem eingebauten 50-MHz-Arbiträrwellenformgenerator und umfassenden Triggermöglichkeiten ist es über seinen großen, kapazitiven 10,1-Zoll-Touchscreen einfach zu konfigurieren (Bild 3). Genauso einfach geht es aber auch über USB.



Bild 3. Mit einer USBTMC-kompatiblen USB-Schnittstelle lassen sich die MSOs der Serie 2560B von B&K Precision leicht mit Python automatisieren. (Quelle: B&K Precision)

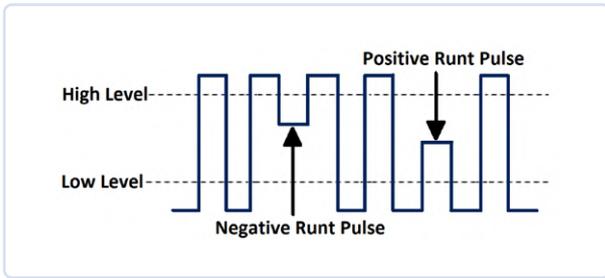


Bild 4. Der Runt-Trigger ist ein cleverer Mechanismus zum Auffinden von Impulsen, die nicht vollständig ansteigen oder abfallen. (Quelle: B&K Precision)



Bild 5. Das tellergroße Moku:Lab ist zwölf Messgeräte in einem und kann über eine Python-API konfiguriert werden. (Quelle: Liquid Instruments)

Fernkonfiguration

Auch hier ist Python die bevorzugte Programmiersprache, dank des USBTMC-Projekts [8] von Alex Forencich, das auf GitHub [9] gehostet wird. Das Modul funktioniert unter Linux, erfordert aber möglicherweise eine Änderung der Berechtigungen. Unter Windows müssen zunächst sowohl *PyUSB* [10] als auch *libusb* [11] installiert werden.

Die ersten Schritte sind relativ einfach. Sie benötigen erst einmal die Hersteller-ID (VID) und die Produkt-ID (PID), zwei USB-spezifische Referenzwerte, die ein USB-Gerät identifizieren. Unter Linux lassen sich diese Werte, sobald das Gerät angeschlossen ist, leicht über die Befehlszeile mit `lsusb` ermitteln. Die daraus resultierende Liste liefert die notwendigen Details. Unter Windows können diese Werte über den Geräte-Manager und die Eigenschaften des Geräts ermittelt werden. In beiden Fällen werden die Werte im 16-Bit-Hexadezimalformat angegeben.

Nun müssen Sie nur noch das *usbtmc*-Modul in Ihrem Python-Code importieren und die verfügbare Programmierschnittstelle (API) verwenden. Das ist so ähnlich wie das Drucken von Text auf dem Bildschirm und das Auswerten von Tastatureingaben. USBTMC ist eigentlich nicht mehr als ein Wrapper, um mit

leistungsfähigen Test- und Messgeräten zu kommunizieren. Die Steuerbefehle sind ASCII-Strings mit verschiedenen Optionen, wie im Programmierhandbuch des Oszilloskops [12] beschrieben. Falls verfügbar, können auch VISA-Resource-Zeichenketten [13] verwendet werden.

Die VID und die PID sind für ein Produkt eindeutig, nicht für das individuelle Gerät. Wenn also zwei oder mehr Geräte desselben Typs angeschlossen sind, können sie unabhängig voneinander über einen dritten Parameter, ihre Seriennummer, adressiert werden. Diese befindet sich normalerweise auf einem Etikett oder kann mit dem Befehl `*IDN?` abgefragt werden (**Beispiel 2**).

Abgesehen von Bandbreite, Abtastrate und Speichertiefe unterscheiden sich Oszilloskope heutzutage kaum noch voneinander. Aber gelegentlich taucht eine kleine, aber interessante Funktion auf: Diese B&K-Reihe verfügt über einen sogenannten Runt-Trigger (**Bild 4**), eine Funktion, die ausgelöst wird, wenn ein Signal einen Schwellenwert (zum Beispiel einen hohen Pegel) über-, aber nicht den anderen (zum Beispiel einen niedrigen Pegel) unterschreitet. Wenn es um Fehleranalyse geht, kann jede kleine Funktion hilfreich sein!

Headless-Testgeräte

Red Pitaya hat bei Testgeräten neue Wege beschritten und die Frage aufgeworfen, ob ein Gerät ein eigenes Display braucht. Dank eines leistungsstarken FPGAs und einer Ethernet-Schnittstelle stellt der PC oder der Laptop die Benutzeroberfläche dar. Liquid Instruments hat den gleichen Ansatz mit seinem Moku:Lab [14] verfolgt, einer eierlegenden Wollmilchsau von der Größe und Form eines Tellers. Das Multi-Testgerät bietet eine Reihe von analogen Ein- und Ausgängen und ist wie ein Labor „in a box“ konzipiert.

An der Vorderseite sind vier BNC-Anschlüsse zu sehen (**Bild 5**). Das rechte Paar stellt die analogen Ausgänge bereit und unterstützt eine Abtastrate von 1 GSa/s pro Kanal bei einer Auflösung von 16 Bit und einer -3-dB-Bandbreite von > 300 MHz. Das linke Paar verkörpert die analogen Eingänge mit einer -3-dB-Bandbreite von 200 MHz an 50 Ω und einer Abtastrate von 500 MSa/s pro Kanal bei einer Auflösung von 12 Bit. Die integrierte Zeitbasis bietet eine Messgeräteabweichung von weniger als 500 ppb. Ein Triggereingang ist ebenso vorhanden wie Anschlüsse, die die Synchronisation mehrerer Geräte ermöglichen. Kabelgebundene Konnektivität ist über einen Ethernet-Anschluss und eine USB-Schnittstelle gegeben, und ein zweiter USB-Stromanschluss ist zum Aufladen eines Tablets verfügbar. Schließlich gibt es noch einen SD-Kartensteckplatz und einen Gleichstromeingang zur Versorgung. Obwohl das Gerät über kabelgebundene Schnittstellen verfügt, ist es eigentlich für die Verwendung in Verbindung mit einem iPad über WLAN (802.11 b/g/n) und die passende App konzipiert. Über die Benutzeroberfläche können zwölf Geräte ausgewählt werden, von den bekannten, wie Oszilloskop und Spektrumanalysator, bis hin zu den eklektischen wie PID-Regler und Laser-Lock-Box.



Beispiel 2: Verwendung von USBTMC in Python zur Steuerung eines Oszilloskops von B&K Precision

```
import usbtmc
instr = usbtmc.Instrument(<VID>, <PID>, 'ABC123456')
# or, without serial number
instr = usbtmc.Instrument(<VID>, <PID>)
print(instr.ask("*IDN?"))
# returns 'BK Precision,2567B-MSO,ABC123456,5.0.1.3.9R3'
```



Beispiel 3: Erstellen einer beliebigen Wellenform auf dem Moku:Lab in Python

Das Tool fungiert auch als Datenlogger. Die Datenerfassung mit bis zu 100 kSa/s ist eigentlich nur durch die Speicherkapazität der SD-Karte begrenzt. Der interne Speicher erlaubt sogar Erfassungsraten bis 1 MSa/s. Natürlich bietet das Moku:Lab, wie alle modernen Messgeräte, eine Python-API sowie MATLAB- und LabVIEW-Unterstützung. Mit Hilfe zahlreicher Beispiele (**Beispiel 3**) kann das Gerät schnell in einen automatisierten Testablauf integriert werden. Es eignet sich gut für die Massenprüfung von HF-Geräten und das Binning von Bauteilen.

Zeit sparen, Genauigkeit verbessern

Mess- und Prüfgeräte sind die Augen des Ingenieurs, die zeigen, was in komplexen Systemen alles vor sich geht. Aber sie haben auch ihre Grenzen. Sporadisch auftretende Ereignisse lassen sich per Definition nur schwer aufspüren, und eine Lösung kann nur vorgenommen werden, wenn eine Ursache erkannt ist. API-gestützte Programmierschnittstellen für Prüf- und Messgeräte sind heute weit verbreitet und reichen von einfach bis raffiniert. Und immer häufiger wird das einfach zu erlernende Python zur Programmiersprache der Wahl. Wenn eine Messaufgabe für einen Fehler immer schwieriger auszulösen ist, aufeinanderfolgende Änderungen erfordert, um eine einzige Testrunde abzuschließen, oder einfach langweilig und repetitiv ist (was zu Bedienfehlern führt), ist im Vergleich dazu die Automatisierung gar nicht so komplex, wie Sie vielleicht befürchten! 

RG – 230046-02

```
"""pymoku example: Arbitrary waveform generator
(c) 2019 Liquid Instruments Pty. Ltd.
(shortened version for Elektor)
"""
from pymoku import Moku
from pymoku.instruments import ArbitraryWaveGen
import numpy as np
# generate a signal the the Arb Waveform Gen should generate on the output
t = np.linspace(0, 1, 100) # Evaluate our waveform at 100 points
# Simple square wave (can also use scipy.signal)
sq_wave = np.array([-1.0 if x < 0.5 else 1.0 for x in t])
# More interesting waveform. Note that we have to normalize this waveform
# to the range [-1, 1]
not_sq = np.zeros(len(t))
for h in np.arange(1, 15, 2):
    not_sq += (4 / (np.pi * h)) * np.cos(2 * np.pi * h * t)
not_sq = not_sq / max(not_sq)
# Connect to your Moku by its device name
m = Moku.get_by_name('Moku')
# Prepare the ArbitraryWaveGen instrument
i = m.deploy_or_connect(ArbitraryWaveGen)
try:
    # Load the waveforms to the device.
    i.write_lut(1, not_sq)
    i.write_lut(2, sq_wave)
    # Configure on-device linear interpolation
    i.gen_waveform(1, period=1e-6, amplitude=1, interpolation=True)
    i.gen_waveform(2, period=1e-6, amplitude=2, interpolation=False)
finally:
    m.close()
```

Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Senden Sie eine E-Mail an den Autor unter stuart.cording@elektor.com oder kontaktieren Sie Elektor unter redaktion@elektor.de.

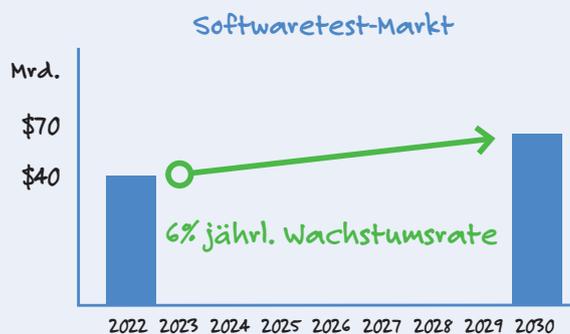
WEBLINKS

- [1] PSU Sequencer von Aim-TTi: <http://bit.ly/40jGiCQ>
- [2] DC-Stromversorgungen der Serie EL-R, Aim-TTi: <http://bit.ly/3kWZmXk>
- [3] pySerial-Modul: <http://bit.ly/3YgSJgZ>
- [4] Beispiel einer TCP/IP-Bridge mit pySerial: <http://bit.ly/3kTDKcm>
- [5] G. Clark, „RFC 2217 Telnet Com Port Control Option“, Cisco Systems Inc., Oktober 1997: <http://bit.ly/3jcQGM4>
- [6] „Universal Serial Bus Test and Measurement Class Specification (USBTMC)“, USB Implementers Forum Inc., April 2003: <http://bit.ly/3YbJY7L>
- [7] MSO-Modell 2567B von B&K Precision: <http://bit.ly/3kN3rNy>
- [8] A. Forencich, „Python USBTMC“, Juli 2014: <http://bit.ly/3HMLLeZN>
- [9] GitHub-Projekt python-usbtmc von A. Forencich: <http://bit.ly/3wK4i4r>
- [10] PyUSB-Software: <http://bit.ly/3WRFW3l>
- [11] libusb-Software: <http://bit.ly/3wLEvIY>
- [12] „Programming Manual 2560B Series“, B&K Precision, September 2022: <http://bit.ly/3WRDYjs>
- [13] „VISA Resource Syntax and Examples“, National Instruments Corp., Mai 2022: <http://bit.ly/3XK3okh>
- [14] Moku:Lab von Liquid Instruments: <http://bit.ly/3WT6bXb>

Test- und Messequipment ist für professionelle Ingenieure, Studenten und Entscheidungsträger wichtig und wird von ihnen häufig genutzt. Hier präsentieren wir Ihnen einige informative Daten zu test- und messtechnischen Themen, darunter Softwaretests, Systemsicherheit, der MEMS-Markt und der globale Markt für Industriesensoren.

Machen Sie weiter Softwaretests!

Arbeiten Sie in der Softwaretest-Branche? Wenn ja, dann herzlichen Glückwunsch: Sie sind in einem gesunden 40-Milliarden-Dollar-plus-Bereich beschäftigt. Gar kein schlechter Job! Einem kürzlich erschienenen Bericht von Global Market Insights zufolge wird der Markt für Softwaretests bis zum Jahr 2030 um 6 % wachsen, was bedeutet, dass Sie in den kommenden Jahren wahrscheinlich einige gute Beschäftigungsmöglichkeiten (und vielleicht auch einen sicheren Arbeitsplatz) haben werden [1].



Zu beachtende Unternehmen

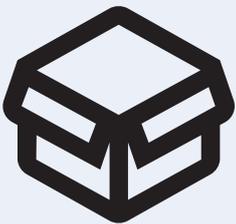
- > Accenture
- > Atos SE
- > Amdocs
- > Cognizant
- > IBM
- > Infosys
- > Keysight (Eggplant)

(Quelle: Global Market Insights)

Welcher Test für Sie?

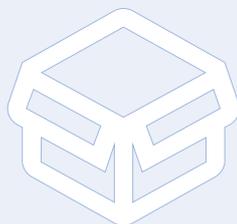
Sie müssen eine Software testen? Möchten Sie eine Anwendung debuggen? Ist Sicherheit für Sie wichtig? Wählen Sie Black Box oder White Box! Bevor Sie sich aber für schwarz oder weiß [2] entscheiden und dann mit dem Testen beginnen, sollten Sie sich die folgenden hilfreichen Ressourcen ansehen!

Black Box



- > Funktionstest: Testen von Funktionalität und Verhalten der Anwendung
- > Keine tiefgreifenden Programmierkenntnisse erforderlich
- > Von Testern verwaltet
- > Zugriff auf Programmcode ist nicht erforderlich

White Box



- > Struktureller Test: Test der Anwendungsinfrastruktur
- > Tiefgreifende Programmierkenntnisse erforderlich
- > Von Entwicklern verwaltet
- > Zugriff auf Programmcode erforderlich

HILFREICHE RESSOURCEN:

- > S. Cording, „Debugging Microcontrollers Without a Debugger“, 17. Januar 2023: <https://elektormagazine.com/no-debugger>
- > S. Cording, „The Full Gamut of Microcontroller Debugging Techniques“, 30. Juni 2020: <https://elektormagazine.com/mcu-debug-20>
- > M. Horkan, „Bewertung der Schwachstellen eingebetteter Systeme“, Elektor Business 8/2018: <https://www.elektormagazine.de/magazine/elektor-77/41808>
- > L. Labbe, „Tracing the Cause of Software Bugs Wirelessly: Circular Buffer and Webserver on the ESP32“, 29. September 2022: <https://elektormagazine.com/trace-bugs-wireless>
- > C. Valens, „Secure Communications – An Interview with Luka Matic“, 6. November 2019: <https://elektormagazine.com/secure-comm-19>

Der MEMS-Markt

Der Markt für mikroelektromechanische Systeme (MEMS) wächst. Verschiedene Forschungsunternehmen sind der Ansicht, dass der globale Markt bis 2031 mit einer jährlichen Wachstumsrate (Compound Annual Growth Rate, CAGR) zwischen 8 % und 11 % wachsen wird, so die Firma Verified Market Research [3][4]. Als Bauteile mit sowohl elektrischen als auch mechanischen Funktionen haben MEMS zahllose Anwendungen in einer Vielzahl von Branchen, darunter die industrielle Messtechnik, die Automobilindustrie, die Unterhaltungselektronik und die Luft- und Raumfahrt. Die superkleinen Bauteile werden zur Druckmessung, Signalverarbeitung, Datenübertragung und vielem mehr eingesetzt. „Die Unterhaltungselektronik hatte 2021 den größten Anteil am Markt für mikroelektromechanische Systeme, der voraussichtlich von 2022 bis 2031 mit einer signifikanten CAGR wachsen wird“, berichtete Allied Market Research im September 2022 [5]. Wenn Sie von MEMS-Technologien fasziniert sind, sollten Sie im Auge behalten, wie MEMS-Chips im medizinischen Bereich in mikrochirurgischen Werkzeugen, Implantaten und Mikronadeln für die Medikamentenverabreichung eingesetzt werden.

MEMS-Markt = **+11%** bis 2031

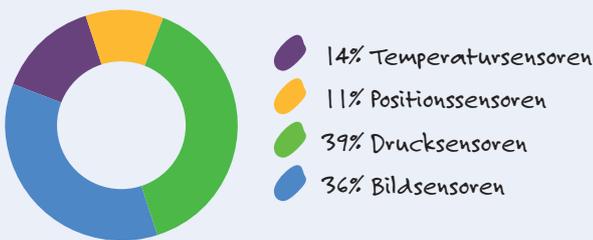


Die Aufgaben von Industriesensoren

Der weltweite Markt für Industriesensoren wird laut Fortune Business Insights bis Ende 2027 ein Volumen von über 33 Milliarden US-Dollar erreichen [6]. Diese Sensoren werden in einer Vielzahl von Anwendungen der Industrie 4.0 eingesetzt und sind für Mess-

und Prüfzwecke unverzichtbar. Zu den industriellen Sensortypen gehören Temperatursensoren, Durchflusssensoren, Gassensoren, Bewegungssensoren und Positionssensoren.

Sensortypen und Marktanteil 2021



Globaler Markt für industrielle Sensoren

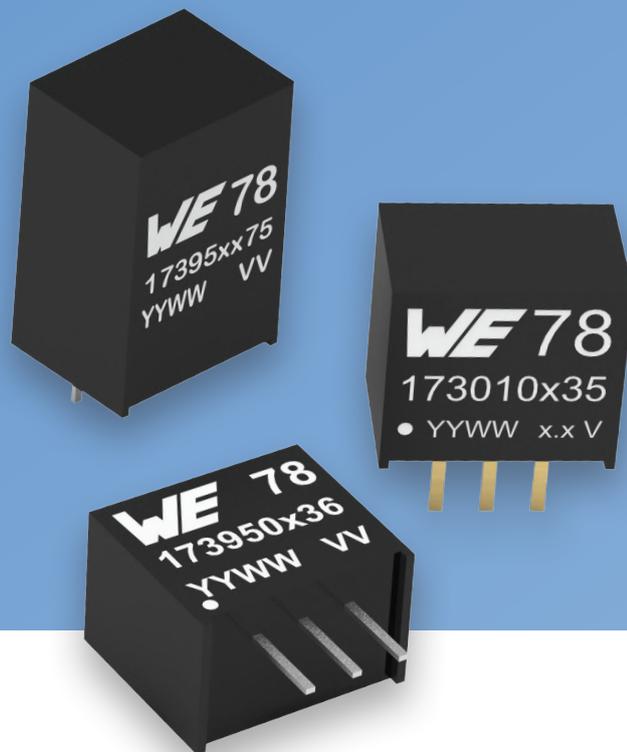


WEBLINKS

- [1] Global Market Insights, „Software Testing Market Report“, 2022: <https://www.elektormagazine.com/GMI-testing-software>
- [2] L. Nguyen, „Key Differences and Similarities Between Black Box and White Box Software Testing“, Orient Software, 2021: <https://www.orientsoftware.com/blog/black-box-and-white-box-software-testing/>
- [3] Verified Market Research, „Microelectromechanical System (MEMS) Market Size And Forecast“, 2023: <https://www.elektormagazine.com/vmr-mems-market>
- [4] Markets and Markets, „Micro-Electro-Mechanical System (MEMS) Market“, 2023: <https://www.elektormagazine.com/mandm-mems-market>
- [5] Allied Market Research, „Microelectromechanical System (MEMS) Market Size And Forecast“, 2022: <https://www.elektormagazine.com/amr-mems-2022>
- [6] Fortune Business Insights, „Industrial Sensors Market Size, Share & COVID-19 Impact Analysis“, 2019: <https://www.elektormagazine.com/fortune-sensors>
- [7] Markets and Markets, „Top 10 Sensors Market by Type (2021-2026)“, 2021: <https://www.elektormagazine.com/markets-top-10-sensors>

Überspannungsschutz für sicheren Betrieb

Transientenschutz für nicht-isolierte DC/DC-Power-Module



Von Timur Uludag (Würth Elektronik eiSos)

In industriellen Umgebungen kommt es aufgrund der umfangreichen elektrischen Infrastruktur häufig zu transienten Überspannungen. Um ein effizientes Filter zur Begrenzung transienter Überspannungen zu entwickeln, müssen viele Einflussparameter berücksichtigt werden.

Viele industrielle Anwendungen arbeiten heute mit logischen Eingangsspannungspegeln wie 5 V_{DC} oder niedriger, wie in **Bild 1** dargestellt. Das Stromverteilungssystem, das zur Versorgung solcher Anwendungen eingesetzt wird, arbeitet häufig mit einer DC-Busspannung von 24 V_{DC}. Für die Umwandlung der höheren Busspannungen auf den niedrigeren logischen Spannungspegel werden in der Regel schaltende DC/DC-Wandler eingesetzt.

Bild 1 zeigt die grundlegende elektrische Struktur einer Industrieanlage. Die einzelnen Teile der Anwendungen werden über einen DC-Bus versorgt. Vor Ort ist jeder einzelne elektrische Verbraucher über eine Unterverteilung mit 24 V angeschlossen. Um die Betriebsspannung für alle Teilsysteme bereitzustellen, werden nicht isolierte Power-Module eingesetzt.

Entstehung von Transienten

Transienten können als kurzzeitige Abweichungen von einem nominellen Spannungswert definiert

werden, die den zulässigen Toleranzbereich der Nennspannung in einem elektrischen System überschreiten. Die Auswirkungen der Transienten sind meist zerstörerisch.

Es gibt in der Regel nicht nur eine einzige mögliche Ursache für eine Transiente auf dem DC-Bus, die zu einem abrupten Anstieg führt. Der Ursprung der Transiente kann wie in Bild 1, Teil A ein Blitzschlag sein (in diesem Fall spricht man von einem Überspannungsstoß, „Surge“), oder sie kann im System selbst erzeugt werden (Bild 1, Teil B). Klassischerweise ist der 24-V-DC-Bus im industriellen Umfeld für 19,2...30 V spezifiziert. Bei der Betrachtung transienter Überspannungen müssen weitere Effekte berücksichtigt werden. Ist zum Beispiel die 24-V-Versorgungsleitung parallel zur Steuerleitung eines Frequenzumrichters verlegt, sind die Impulse kapazitiv gekoppelt und die 24 V schwingen im Pulsmuster des Frequenzumrichters.

Ein unsachgemäßer oder gar nicht vorhandener Überspannungs-/Transientenschutz führt zu Fehlfunktionen durch elektrische Beschädigung des DC/DC-Wandlers und damit zu höheren Anlagenausfallzeiten und Kosten. Für angemessene Annahmen und eine korrekte Berechnung müssen wir eine normalisierte Transiente wie den in der Norm IEC 61000-4-5 beschriebenen Überspannungsstoß verwenden.

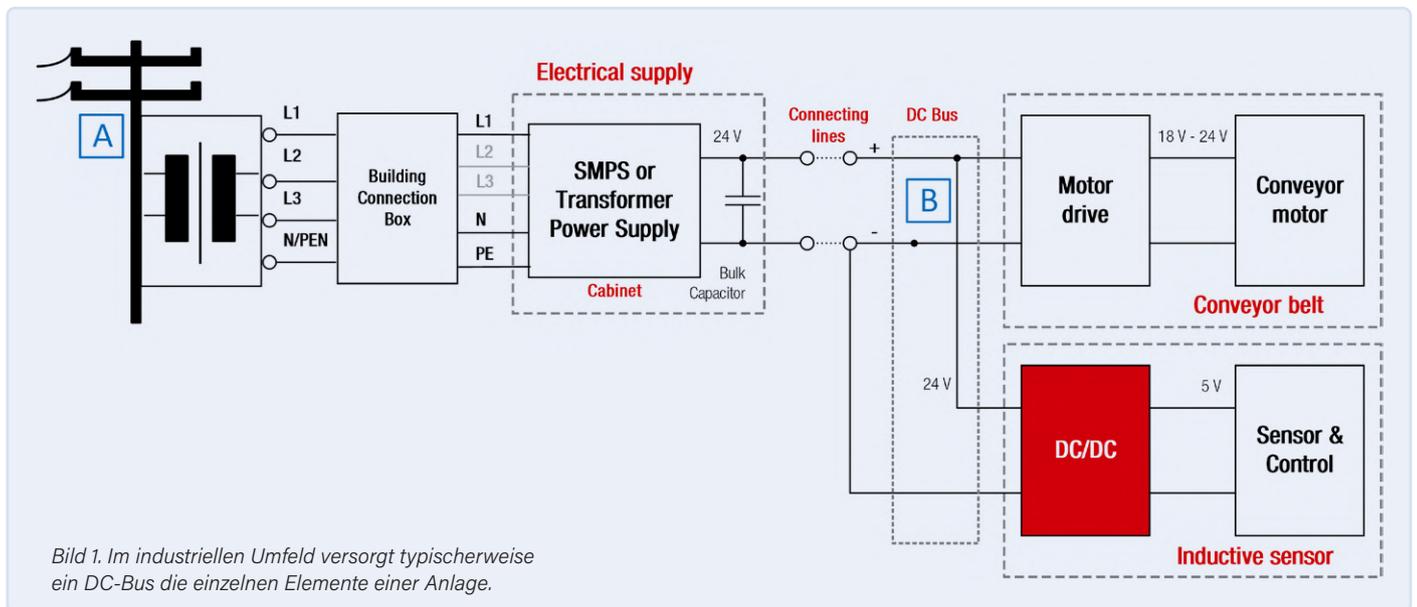


Bild 1. Im industriellen Umfeld versorgt typischerweise ein DC-Bus die einzelnen Elemente einer Anlage.

Filterkonzept

Bild 2 zeigt das gesamte Konzept eines Filters zur Erhöhung der Störfestigkeit (grün), das zwei Filterstufen umfasst. Eine Stufe klemmt die hohe transiente Überspannung während eines Überspannungsereignisses, was mit Bauteilen zur Transientenunterdrückung wie unidirektionalen TVS-Dioden erreicht werden kann. Für die zweite Stufe wird ein passives LC-Filter empfohlen, das die Spannungspiegel dämpft, die die maximale Betriebsspannung des DC/DC-Wandlers überschreiten.

Grenzwerte für die Auslegung von Störschutzfiltern

Der kritischste Parameter eines Schaltreglers, der für den Aufbau eines Transientenfilters benötigt wird, ist die Eingangsspannung. In vielen Fällen sind im Datenblatt zwei verschiedene Werte angegeben. Der eine ist die absolute maximale Eingangsspannung, deren Überschreitung zu einer dauerhaften Beschädigung des Schaltreglers führt. Der andere ist die maximale Betriebsspannung, die der maximal spezifizierten Eingangsspannung entspricht, die der Hersteller des Schaltreglers für den Betrieb erlaubt. Zum Schutz vor transienten Überspannungen wird empfohlen, den Immunitätsfilter so auszulegen, dass die Eingangsspannung des Moduls auch bei transienten Überspannungsereignissen niemals die maximale Betriebsspannung überschreitet. Für die weiteren Berechnungen wird hier als Beispiel der Abwärtsregler 173010535 mit einer absoluten maximalen Eingangsspannung von $U_{in} = 44 \text{ V}$ verwendet [1].

Entwurf des Störfestigkeitsfilters

In diesem Artikel wird eine unidirektionale Leistungs-Transienten-Spannungsunterdrückungsdiode (TVS) als Schutzelement für den Eingang des DC/DC-Moduls eingesetzt. Unidirektionales Betriebsverhalten bedeutet, dass die U-I-Charakteristik fast die gleiche ist wie die einer Zenerdiode. Daher wird die Diode

normalerweise in umgekehrter Richtung verwendet. Bei Überschreiten der spezifizierten Durchbruchspannung des Bauteils wird die TVS-Diode leitend. Die Höhe der Klemmspannung wird dann durch den Strom bestimmt, der durch das Bauelement fließt. Das folgende Rechenbeispiel bietet eine vereinfachte, praktische Anleitung zur Erstellung eines gut veranschlagten Filters. Diese Filterabschätzung ermöglicht eine schnelle Verbesserung bei der Durchführung realer Anwendungstests.

Um einen geeigneten Entwurf auf der Grundlage einer TVS-Diode für den Transientenschutz des Schaltregler-Moduls zu erstellen, sind folgende Parameter erforderlich:

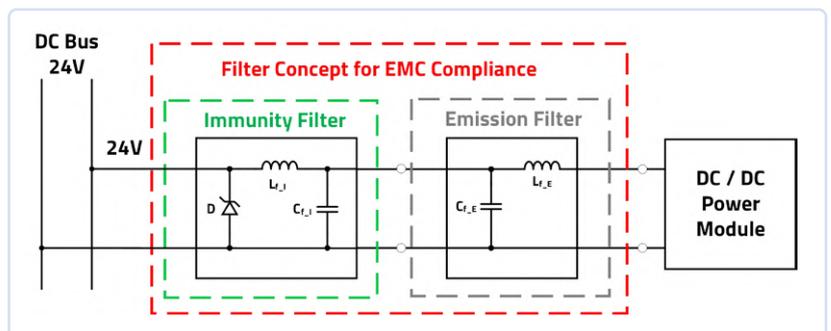
- U_{DC} : Versorgungsspannung für das Power-Modul
- U_{BR} : Spannung, bei der ein Strom von 1 mA durch die TVS-Diode fließt
- I_{Spitze} : maximaler Spitzenstrom, der durch die TVS-Diode bei $U_{Clamp\ max}$ fließt
- P_{Diss} : maximal zulässige Verlustleistung für die TVS-Diode
- $U_{Clamp\ max}$: Spannung, bei der der maximal spezifizierte Strom durch die Diode fließt

Bild 2. Filterkonzept für die EMV-Konformität von Mag³C-Power-Modulen mit Störfestigkeit und Emissionsfilter.

Erste Stufe des Störfestigkeitsfilters

Bestimmung von U_{DC}

Für U_{DC} ist der entscheidende Wert für die Störfestigkeit





die maximale DC-Busspannung, die vorhanden sein kann, nicht der Nennwert, der bei einem 24-V-Bus 24 V bedeutet. In der industriellen Umgebung ist ein 24-V_{DC}-Bus für 19,2...30 V spezifiziert. Daraus folgt, dass der Maximalwert $U_{DC} = 30 V_{DC}$ ist.

Die Auswahl der TVS-Diode für die nächsten Berechnungsschritte basiert auf den im Online-Katalog von Würth Elektronik verfügbaren Bauteilen. Es gibt zwei mögliche Kandidaten, die 824541301 und die 824551301 [2].

Bestimmung der U_{BR}

U_{BR} ist definiert als die Spannung, bei der ein Strom von 1 mA durch die TVS-Diode fließt. Dieser Wert, hier 35,05 V, ist in Wirklichkeit nicht exakt festgelegt, da es sich um einen PN-Übergang handelt und daher eine Toleranz besteht. Die Toleranz wird im Datenblatt mit $\pm 5\%$ angegeben. Daraus ergibt sich ein U_{BR} von 33,325 V bis 36,8025 V, in dem die Diode mit einem Strom von 1 mA zu leiten beginnt.

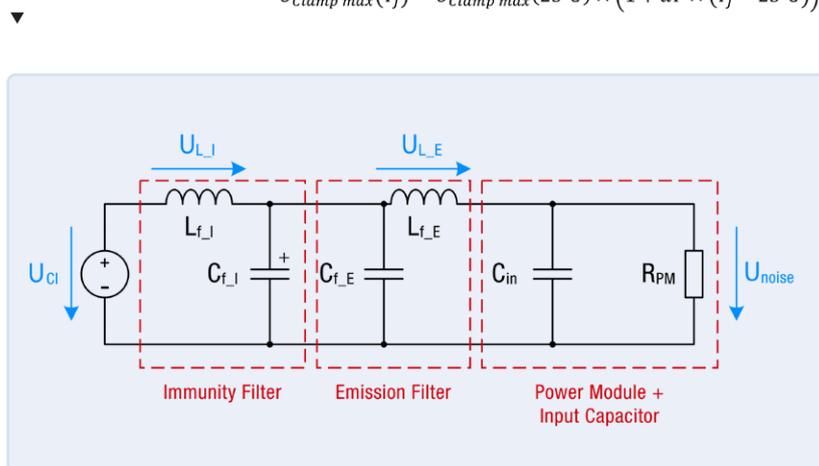
Es ist jedoch auch notwendig, den Wert zu kennen, bei dem die transiente Spannung geklemmt werden soll. Dies wird durch den Parameter $U_{Clamp\ max}$ dargestellt.

Bestimmung von $U_{Clamp\ max}$

Dieser Wert kann ebenfalls dem Datenblatt entnommen werden. Für die gewählte Diode beträgt die Spannung 48,4 V bei einem Spitzenstrom von I_{Spitze} von 31 A, was einem 10/1000- μ s-Impuls entspricht. Bislang wurde bei den Berechnungen von einer idealen Laborumgebung mit einer kontrollierten Umgebungstemperatur von 25 °C ausgegangen. Die Realität sieht jedoch anders aus. Umgebungstemperaturen von bis zu 55 °C sind für elektronische Bauteile wie einer TVS-Diode üblich. Daher muss die Berechnung um einen Temperaturfaktor ergänzt werden. Vor allem $U_{Clamp\ max}$ und die Impuls-Spitzenleistung sind stark von der Temperatur abhängig. **Gleichung 1** zeigt den Temperatureinfluss auf $U_{Clamp\ max}$:

$$U_{Clamp\ max}(T_j) = U_{Clamp\ max}(25^\circ C) \times (1 + \alpha T \times (T_j - 25^\circ C))$$

Bild 3. Ersatzschaltung für die Berechnung des Störfestigkeitsfilters der zweiten Stufe.



Für den „Standby-Fall“, in dem mit Ausnahme eines Leckstroms von 1 μ A fast kein Strom durch die TVS-Diode fließt, entspricht die Sperrschichttemperatur (T_j) fast der Umgebungstemperatur. Geht man von einem Temperaturkoeffizienten αT für diesen TVS-Typ von $9,9 \times 10^{-4}/^\circ C$ aus, so ergibt sich eine maximale Klemmspannung U_{Clamp} von 49,84 V bei 55 °C. Dieser Wert ist der Ausgangspunkt für die Dimensionierung der zweiten Stufe des Filters.

Zweite Stufe des Störfestigkeitsfilters

Die Frage ist nun, wie man die richtige Filterdämpfung abschätzen kann und wie man die besten Werte für die Filterbauteile erhält. Ausgehend von der Dämpfung kann die minimale Filterdämpfung mit **Gleichung 2** berechnet werden:

$$G = 20 \log \left(\frac{U_{PM\ Max}}{U_{Clamp\ max}} \right) \quad G = 20 \log \left(\frac{44 V}{49,84 V} \right) = -1,08 dB$$

(Anstelle des Symbols A für Abschwächung wird hier das Formelzeichen G (Verstärkung) verwendet. Eine negative Verstärkung bedeutet eine Abschwächung). Gleichung 2 berücksichtigt die resultierende Klemmspannung $U_{Clamp\ max}$ der TVS-Diode während des Überspannungseignisses und die maximale Betriebsspannung $U_{PM\ Max}$ des gewählten Wandlers. Die Aufgabe besteht darin, das Filter gemäß **Bild 3** zu entwerfen, wobei der TVS-Diode ein LC-Filter angefügt wird. Der Entwickler kann, um einen entsprechenden Filterkondensator zu berechnen, den Wert der Induktivität wählen. Der Grund dafür ist, dass die Filterinduktivität in Reihe mit der Anwendung liegt und ihr Widerstand (R_{DC}) so unerwünschte Verluste verursacht. Daher sollte die Induktivität mit dem kleinstmöglichen R_{DC} -Wert unter Berücksichtigung des maximalen nominellen Ausgangsstroms des DC/DC-Wandlers gewählt werden. Für diesen beispielhaften Filterentwurf wurde eine WE-PD2 (744776112) mit einer Induktivität von 12 μ H mit einem R_{DC} von 336 m Ω und einem Nennstrom von 2,72 A gewählt. Der DC-Eingangswiderstand des Power-Moduls kann mit der gegebenen Eingangs- und Ausgangsspannung, dem Ausgangsstrom sowie dem Wirkungsgrad im Betrieb bestimmt werden. Trägt man diese Parameter zusammen, lässt sich für den Eingangswiderstand des DC-Wandlers folgende **Gleichung 3** erstellen.

$$R_{PM} = \frac{U_{in}^2}{U_{Out} \times I_{Out} \times \eta} = \frac{U_{in}^2}{P_{in}} \quad R_{PM} = \frac{(24 V)^2}{5 V \times 1 A \times 0,88} = 101 \Omega$$

Bild 3 zeigt das Ersatzschaltbild, in dem die TVS-Diode vereinfacht als Spannungsquelle bei Überspannungsimpulsen dargestellt ist. Der verbleibende Teil

des Schaltbildes für das EMV-Modell besteht aus zwei LC-Filtern für Störfestigkeit (Überspannungsschutz) und Emission (EMV-Dämpfung), dem Eingangskondensator des DC/DC-Wandlers und dem Eingangswiderstand des Reglers.

Da es sich um eine Innenraumanwendung handelt und somit nur eine indirekte Überspannungseinkopplung vorliegt, basieren die folgenden Annahmen und Berechnungen auf einem 8/20-µs-Impuls. Zur weiteren Vereinfachung ist es möglich, C_{f_E} und L_{f_E} wegzulassen, da dieses Filter zur Unterdrückung von Störungen bei der Schaltfrequenz des DC/DC-Reglermoduls ausgelegt ist (**Bild 4**). Das Power-Modul schaltet typischerweise mit 520 kHz.

Um die Dämpfung G des Filters zu berechnen, ist es notwendig, die Ausgangsspannung mit der Eingangsspannung des Systems zu vergleichen (**Gleichung 4**).

$$\frac{U_{noise}}{U_{Cl}} = \frac{1}{Z_{Lf} + \left(\frac{1}{Y_{Cf} + Y_{Cin} + Y_{RPM}} \right)}$$

$$G = 20 \log \left(\left| \frac{U_{noise}}{U_{Cl}} \right| \right)$$

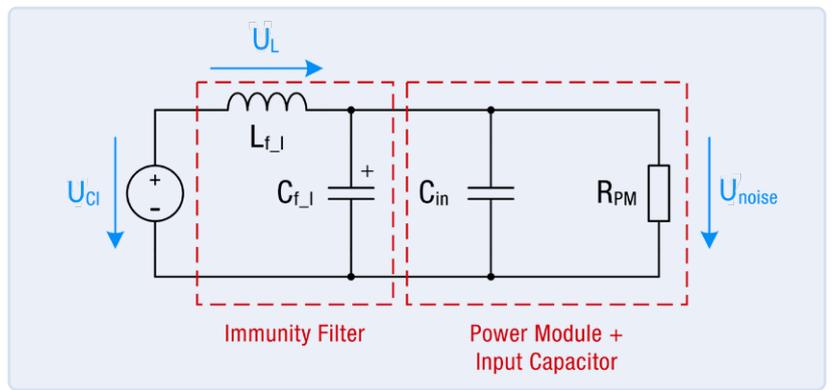
$$G_{immunity_{filter}} = 20 \log \left(\left| 1 - \omega^2 L_f (C_f + C_{in}) + j\omega \frac{L_f}{R_{PM}} \right| \right)$$

Mit dem zuvor berechneten Gleichstrom-Eingangswiderstand $R_{PM\ in}$ des DC/DC-Reglers kann mit **Gleichung 5** der erforderliche Kondensatorwert ermittelt werden:

$$C_f = \frac{1 - \left(10^{\frac{G}{20}} - \left(\omega \frac{L_f}{R_{PM}} \right)^2 \right)^{\frac{1}{2}}}{\omega^2 \cdot L_f} - (C_{in} + C_{f_emission})$$

Dabei steht L_f für die Entstör-Filterdrossel, R_{PM} für den Gleichstrom-Eingangswiderstand des Wandlers, C_{in} für den Eingangskondensator und $C_{f_emission}$ für den EMI-Filterkondensator einer Eingangs-Pi-Filterstruktur. Die Werte können dem Datenblatt des 173010535 entnommen werden. Daher wird dieser Frequenzwert für die Worst-Case-Berechnung verwendet. Unter der Annahme, dass eine Induktivität von 12 µH verwendet wird, wäre das Berechnungsergebnis für den Filterkondensator $C_f = 218 \mu F$. Für den Kondensator wurde der Standardwert von 220 µF (860010775018) gewählt, da dieser größer ist als der berechnete Wert. Ein Wert unter der berechneten Kapazität würde keine ausreichende Dämpfung des Filters gewährleisten. Die endgültigen Bauteile (von Würth Elektronik) des Filters sind also wie folgt:

TVS-Diode = 824541301, $L_{f_I} = 744776112$, $C_{f_I} = 860010775018$



Der Einfluss der Temperatur auf $U_{Clamp\ max}$ und damit auf den Wert des Filterkondensators ist in **Tabelle 1** dargestellt.

Die Kondensatorwerte sind zwar berechnet, doch der reale Kondensator unterliegt Toleranzen von bis zu $\pm 20\%$. Wenn die Temperaturabhängigkeit von $U_{Clamp\ max}$ nicht vollständig berücksichtigt wird, könnte es passieren, dass ein Kondensator mit einer zu niedrigen Kapazität eingesetzt wird.

Transientenschutz und EMI

Um ein effizientes Filter zur Begrenzung transienter Überspannungen zu entwickeln, müssen viele Einflussparameter berücksichtigt werden. Dies ist besonders wichtig in einer industriellen Umgebung, da hier aufgrund der umfangreichen elektrischen Infrastruktur häufig transiente Überspannungen auftreten. Das Filter mit Transientenschutz ermöglicht einen effizienten Schutz des DC/DC-Wandlermoduls und bewirkt gleichzeitig eine Dämpfung der hochfrequenten Emissionen. ◀

RG – 230180-02

Tabelle 1.

Umgebungstemperatur	$U_{Clamp\ max}$	Filterkondensator
25 °C	48,4 V	178 µF
55 °C	49,84 V	218 µF



Über den Autor

Timur Uludag erwarb seinen Dipl.-Ing. (FH) in Mechatronik an der Fachhochschule Regensburg. Anschließend arbeitete er mehrere Jahre als Hardware-Ingenieur im Bereich Schaltnetzteile und analoges Schaltungsdesign. Seit 2015 ist Timur Senior Technical Marketing Manager bei Würth Elektronik eiSos im Geschäftsbereich MagI³C Power Modules. Dort ist er auf die Roadmap-Planung und die Markteinführung neuer Power-Module spezialisiert.

WEBLINKS

[1] Datenblätter für SIP-3 MagI³C-FDSM Power Module: we-online.com/MAGIC-FDSM

[2] Datenblätter für WE-TVSS Dioden: we-online.com/WE-TVSP

Mess- und Prüfgeräte von Wiha



Installationstester und Messgeräte, auf die Sie sich verlassen können

Von Transfer Multisort Elektronik Sp. z o.o.

Elektrische Mess- und Prüfgeräte sind unverzichtbar für jeden Techniker, der sicher und effektiv arbeiten will. Die hier vorgestellten Wiha-Messgeräte sind nur der Anfang des beeindruckenden Sortiments von Transfer Multisort Elektronik.

Messgeräte sind Werkzeuge, die sich sowohl in den Wartungswerkstätten großer, moderner Fabriken als auch in der Garage eines Heimwerkers als nützlich erweisen können. Sie sind für die Messung bestimmter Eigenschaften von elektrischen Geräten und Anlagen geeignet, können aber auch zur Einschätzung der Umgebungsbedingungen verwendet werden. Dank solcher Werkzeuge ist es viel einfacher, einen Fehler in einem defekten Gerät zu entdecken oder zu überprüfen, ob ein neu zusammengestellter Verteiler richtig funktioniert. Auf der Suche nach den richtigen Messgeräten [1] lohnt es sich, einen Blick auf die Produktpalette von Wiha [2] zu werfen. Dieser deutsche Hersteller bietet Werkzeuge von einwandfreier Qualität, Präzision und Zuverlässigkeit, die weltweit anerkannt sind. Dies zeigt sich auch an den zahlreichen Auszeichnungen, die die Firma Wiha jedes Jahr für ihre innovativen Ideen, ihr Design und ihren Kundenservice erhält. Jetzt wurde die Produktpalette um Messgeräte erweitert.

Berührungslose Spannungsprüfer

Ein berührungsloser Spannungsdetektor (Bild 1) gilt als ein Muss für jeden Elektriker und Elektroingenieur. Mit einem solchen Produkt kann man überprüfen, ob bei einem Stromkreis eine elektrische Feldstärke vorhanden ist. Es

ist unglaublich sicher, mit solchen Geräten Tests durchzuführen, da sie die Spannung aus der Entfernung messen können. Mit anderen Worten: Sie müssen keine gefährlichen, unter Spannung stehende Teile berühren.

Wiha hat mittlerweile drei dieser *Volt-Detektoren* im Programm, die sich in Preis und Ausstattung unterscheiden. Das Basismodell ist mit einer LED ausgestattet, die den Benutzer über eine Wechselspannungserkennung im Bereich von 90 V bis 1000 V informiert. Die beiden anderen Geräte erkennen einen größeren Wechselspannungsbereich (zwischen 12 V und 1000 V) als das Basismodell und informieren den Anwender mit einer LED und einem akustischen Signal. Die High-End-Variante ist nicht nur mit einer Taschenlampe ausgestattet, sondern im Gegensatz zum Mid-Range-Gerät ATEX-zertifiziert. Sie darf daher auch in explosionsgefährdeten Bereichen eingesetzt werden.

Zweipolige Spannungs- und Durchgangsprüfer

Ein weiteres Gerät, das sich in einer Werkstatt als nützlich erweisen kann, ist der zweipolige Spannungs- und Durchgangsprüfer (Bild 2). Das Gerät kann zum Prüfen von elektrischen Anlagen und Schaltschränken verwendet werden.

Alle drei erhältlichen Tester besitzen einen Messbereich von 12...1000 V_{AC} oder

12...1500 V_{DC}, entsprechend den Normen DIN EN 612433 und VDE 0682-401. Sie können auch einen Durchgangswiderstand im Bereich von 0...500 kΩ messen. Die Mid-Range- und die High-End-Variante unterscheiden sich am stärksten. Das Mid-Range-Modell ist für die Prüfung von Photovoltaikanlagen geeignet und bietet die Möglichkeit der berührungslosen Spannungsmessung, der Frequenzmessung (von 1...950 Hz) und der Widerstandsmessung (zwischen 0...1999 Ω). Das High-End-Modell hingegen ist für den Einsatz in der Elektromobilität bestimmt und kann Frequenzen zwischen 40...400 Hz messen.

Digitalmultimeter

Digitalmultimeter (Bild 3) sind die am häufigsten verwendeten Messgeräte. Ihre Beliebtheit ist darauf zurückzuführen, dass sie verschiedene elektrische Eigenschaften messen können und einfach zu bedienen sind.

Zu dieser Produktkategorie gehören zwei leicht unterschiedliche Messgeräte. Das einfachere Modell WIHA 45218 kann den Durchgang in einem Bereich von bis zu 30 Ω feststellen und Widerstände von bis zu 40 MΩ messen. Darüber hinaus kann das Gerät Spannungen zwischen 0...600 V (sowohl Wechsel- als auch für Gleichspannung), Stromstärken zwischen 40 mA und 10 A (AC/DC) sowie Frequenzen bis 5 MHz messen.



Bild 2. Der zweipolige Spannungs- und Durchgangsprüfer kann den Durchgang in einem Bereich von 0...500 kΩ prüfen.



Bild 3. Das Digitalmultimeter WIHA 45218 misst Spannungen im Bereich von 0...600 V.



Bild 4. Mit einer Wiha-Strommesszange können Sie Ströme bis zu 400 A (AC/DC) messen.



Bild 5. Der Netzsteckdosentester ermöglicht eine einfache und schnelle Prüfung von Steckdosen.



Bild 6. Der Drehfeldrichtungsanzeiger ist eine perfekte Lösung für industrielle Wartungswerkstätten.

Darüber hinaus ist das Gerät mit einer HOLD-Funktion und einer automatischen Messbereichseinstellung ausgestattet.

Das andere Multimeter WIHA 45215 besitzt ein wenig weitere Messbereiche. Es kann ebenfalls einen Durchgang im Bereich von bis zu 30 Ω prüfen, aber höhere Widerstände bis zu 200 M Ω erfassen. Außerdem kann es Spannungen zwischen 0 V und 1000 V (AC/DC), Stromstärken von 40 mA bis 10 A (AC/DC) und Frequenzen bis zu 60 MHz messen. Darüber hinaus verfügt das Produkt über TrueRMS-, HOLD- und Min/Max-Funktionen. Es bietet auch die Möglichkeit der berührungslosen Spannungsmessung sowie eine automatische und manuelle Messbereichseinstellung.

Strommesszange

Ein Zangenmessgerät (Bild 4) kann nützlich sein, wenn Sie Strom messen müssen, ohne einen Leiter aufzutrennen. Um ein Kabel zu prüfen, brauchen Sie es nur zwischen die Backen zu legen, und das Messgerät zeigt Ihnen den Strom an. Aufgrund dieser Eigenschaft sind solche Geräte bei Elektroingenieuren und Technikern sehr beliebt.

Der Hersteller bietet ein professionelles Zangenmessgerät an, das WIHA 45219. Mit diesem Gerät können Sie nicht nur Ströme bis zu 400 A (AC/DC) messen, sondern Durchgangsprüfungen in einem Bereich bis zu 30 Ω vornehmen und Spannungen im Bereich von 0...1000 V bei Wechselstrom und von 0...1500 V

bei Gleichstrom messen, außerdem Frequenzen bis zu 5 MHz und Widerstände bis zu 40 M Ω . Das Produkt ist außerdem mit den Funktionen TrueRMS, HOLD und Min / Max / Avg ausgestattet. Darüber hinaus verfügt das Gerät über eine integrierte Taschenlampe und kann zum Beispiel bei der Prüfung von Photovoltaikanlagen und im Bereich der Elektromobilität eingesetzt werden.

Steckdosentester

Mit dem interessanten Netzsteckdosentester WIHA 45220 (Bild 5) können Sie verschiedene Arten von Steckdosen prüfen. Je nach Testergebnis leuchten die drei LEDs auf der Frontplatte in einer voreingestellten Konfiguration auf. Das Produkt kann zum Beispiel feststellen, ob die L/N-Kabel verkehrt herum angeschlossen wurden, aber auch eine fehlende Phasenleitung oder eine fehlende Erdung erkennen. Ein solches Gerät kann sich für Elektriker als äußerst nützlich erweisen.

Drehfeldrichtungsanzeiger (Phasenfolgeprüfer)

Die Phasenfolge ist in dreiphasigen Motorsystemen von entscheidender Bedeutung. Ein Fehler in der Phasenfolge kann zu Schäden am Motor und gar zu einem unerwünschten Stillstand der Produktion führen. Ein Gerät, mit dem Sie überprüfen können, ob die Phasen in der richtigen Reihenfolge angeschlossen wurden, ist der Drehfeldrichtungsanzeiger (Bild 6).



Bild 7. Das Ergebnis der Durchgangsprüfung wird durch eine LED und ein akustisches Signal angezeigt.

Wenn das Messgerät an den Motor angeschlossen ist, zeigen die LEDs die Drehrichtung an und informieren den Benutzer über eine falsche Phasenfolge. Ein solcher Drehfeldanzeiger ist ein Muss für industrielle Wartungswerkstätten.

Durchgangsprüfer

Auch dem besten Elektriker kann bei der Zusammenstellung einer Elektroinstallation ein Fehler unterlaufen. Um Fehler auszumachen und zu lokalisieren, können Sie einen Durchgangsprüfer verwenden (Bild 7). Das Gerät ist für Fachleute gedacht und bietet zwei Messbereiche (bis zu 10 Ω und bis zu 500 Ω). Der Test entspricht der Norm CAT II 400 V. Der Durchgang wird durch eine LED und ein lautes, hörbares Signal angezeigt. Der Hersteller gibt an, dass das Signal in einem anderen Raum oder sogar in einem anderen Stockwerk zu hören ist. ◀

RG – 230192-02

WEBLINKS

[1] Mess- und Prüfgeräte: https://www.tme.eu/de/katalog/messeinrichtungen_100164/p/wiha_248/

[2] WIHA-Produkte: https://www.tme.eu/de/linecard/p/wiha_248/

[3] Quelle (Originalartikel): <https://www.tme.eu/de/en/news/library-articles/page/51790/wiha-measuring-equipment/>



Bild 1. Das Moku:Lab hat zwei analoge Ein- und Ausgänge und verfügt über einen Schlitz an der Oberseite des Gehäuses, der als Tablet-Halter verwendet werden kann.



Automatisierung von Tests und Zusammenarbeit bei Testergebnissen

Von Stuart Cording für Mouser Electronics

Wenn es um das Testen komplexer Anwendungen geht, sind einige wenige Werkzeuge unerlässlich, zum Beispiel diese Testgeräte aus dem Sortiment von Mouser Electronics.

Moderne Applikationen werden immer komplexer, nicht nur in der Entwicklung, sondern auch beim Testen. Deshalb wird die Automatisierung von Tests auch in der Forschungs- und Entwicklungsphase immer wichtiger, um die kleinen, sporadisch auftretenden Probleme zu lösen, die man sonst nicht in den Griff bekommt. Doch für Testläufe über Nacht oder am Wochenende sind Prüf- und Messgeräte erforderlich, die flexibel sind und so programmiert werden können, dass sie eine Vielzahl von Testaufgaben ausführen. Die Messergebnisse müssen dann für eine Auswertung gespeichert werden, um diesen lästigen Problemen auf die Spur kommen zu können!

Das Angebot an Prüf- und Messgeräten und deren Funktionen für die Automatisierung und die gemeinsame Nutzung von Ergebnissen nimmt glücklicherweise zu. High-End-Anbieter erweitern die Zahl der angebotenen Features immer mehr und liefern durch ihre Zertifizierungsprogramme auch kalibrierte Ergebnisse. Zudem gibt es eine Reihe neuer Anbieter, die häufig eine beispiellose Messflexibilität in Kombination mit der Integration in individuelle Hardware-in-the-Loop-Testlösungen bieten. Solche Lösungen eignen sich hervorragend für Forschungs- und Entwicklungsteams, die ihre Entwürfe

schnell testen und iterieren müssen, aber nicht über fünf- oder sechsstelligen Budgets für Messgeräte oder ein professionelles HIL-Setup verfügen.

Maximale Messflexibilität und Headless-Testgeräte

Die Zahl der Headless-Messgeräte hat kürzlich stark zugenommen. Dabei handelt es sich um Geräte, bei denen die Messergebnisse mithilfe von Software auf Laptops, Computern oder Tablets angezeigt werden. Sie basieren in der Regel auf SoC-Bausteinen (System-on-Chip) mit FPGAs (Field-Programmable Gate Arrays), die mit High-Speed-Analog-Frontends und Digital-Analog-Wandlern (DAC) verbunden sind. Durch das FPGA lassen sich die Geräte im Handumdrehen umkonfigurieren, sodass sie mal als Oszilloskop, im nächsten Moment als Signalanalysator und anschließend als PID-Regler verwendet werden können. Dank der großen Kapazität des internen Speichers eignen sie sich außerdem perfekt für die Aufzeichnung von Messungen, ganz gleich, ob es sich um einen Burst von einer Millisekunde oder um eine kontinuierliche Aufzeichnung über Tage handelt.

Das Moku:Lab von Liquid Measurement [1] ist eine solche umkonfigurierbare Hardware-Plattform, die zwölf verschie-

dene Instrumente bietet. Die Software ist sowohl für Windows- und Mac-OS-Desktops/Laptops als auch für das iPad als App erhältlich. Das runde Gehäuse hat einen Durchmesser von 20 cm und eine Höhe von 4,4 cm und findet so auch auf vollen Labortischen seinen Platz. Ein Schlitz auf der Oberseite dient als Ständer für das iPad. Neben der Nutzung des internen Speichers können Daten, Einstellungen und Screenshots direkt in die Dropbox und zu anderen Cloud-Services übertragen oder per E-Mail verschickt werden.

An der Vorderseite befinden sich vier BNC-Anschlüsse. Das rechte Paar stellt die analogen Ausgänge bereit und unterstützt eine Abtastrate von 1 GSa / s pro Kanal bei einer Auflösung von 16 Bit und einer -3-dB-Bandbreite von > 300 MHz. Das linke Paar bietet analoge Eingänge mit einer -3-dB-Bandbreite von 200 MHz bei 50 Ω und einer Abtastrate von 500 MSa / s pro Kanal bei einer 12-Bit-Auflösung. Die integrierte Zeitmesseinheit bietet eine Messgeräteabweichung von weniger als 500 ppb (Bild 1).

An der Rückseite befinden sich weitere BNC-Anschlüsse für einen externen 10-MHz-Referenztakt Eingang und einen 10-MHz-Ausgang. Damit können die von mehreren Moku-Geräten erfassten Signale miteinander synchronisiert werden. Ein Triggereingang ist ebenfalls vorhanden. Daneben gibt es einen Ethernet-Anschluss, eine USB-Schnittstelle, einen USB-Power-Port (zum Aufladen eines Tablets), einen SD-Kartensteckplatz und den Gleichspannungsanschluss. Eine Verbindung über WLAN (802.11 b/g/n) wird ebenfalls unterstützt.

Die zwölf unterstützten Instrumente umfassen die gängigen Geräte, die man erwarten

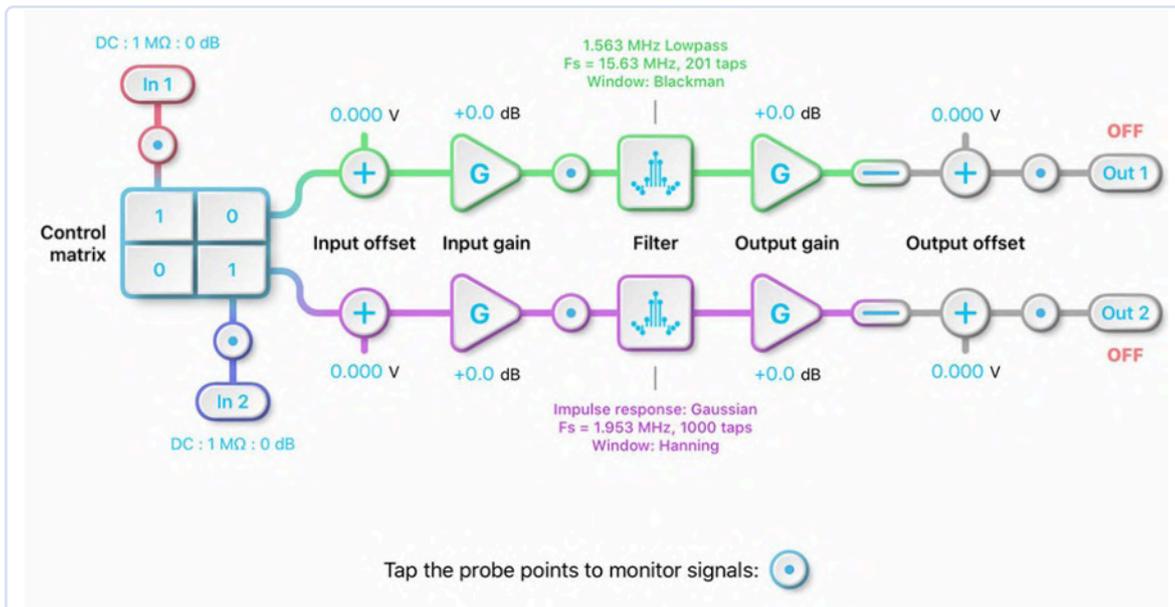


Bild 2. Die Benutzeroberfläche des FIR-Filter-Builders von Moku:Lab macht deutlich, was im Signalpfad passiert und vereinfacht die Konfiguration.

kann, sowie Tools, die gleichzeitig die Ein- und Ausgänge nutzen. Dazu zählen ein PID-Regler mit zwei vollständig konfigurierbaren Pfaden und ein FIR-Filter-Builder für Filter mit endlicher Impulsantwort und bis zu 14.819 Koeffizienten (Bild 2). Außerdem ist eine Laser-Lock-Box enthalten, die mithilfe des Pound-Drevel-Hall-Verfahrens [2] die Laserfrequenz stabilisiert, wie sie in Gravitationswellendetektoren, bei der Zeitmessung und in der Atomphysik verwendet wird.

Es gibt Programmierschnittstellen (API) für MATLAB, LabVIEW und Python, die viele Möglichkeiten zur Implementierung von

automatisierten Tests und Datenprotokollen bieten, die alle durch Beispiele auf GitHub [3] unterstützt werden. Für eine robuste und unkomplizierte Anwendung zur Zweikanal-Datenprotokollierung in Python sind nur wenige Zeilen Code erforderlich (Bild 3).

Langzeittests bei thermischen Problemen

Manchmal soll die langfristige Selbsterhitzung oder Wärmeabgabe des Gerätegehäuses untersucht werden. In solchen Fällen können Infrarotkameras eine große Hilfe sein, denn damit lassen sich heiße und kalte

Stellen aufspüren. Teledyne FLIR ist auf dem Gebiet der visuellen Temperaturerfassung sehr bekannt. Die Produkte des Unternehmens werden von vielen Softwareprogrammen unterstützt, die die Datenanalyse, -zusammenstellung und -zusammenarbeit vereinfachen. Geräte wie das MR265 [4] (Bild 4) bieten ein sogenanntes Multi-Spectral Dynamic Imaging Enhancement (MSX®) [5], bei dem die Bildschärfe im Vergleich zu herkömmlichen Infrarot-Bildgebungsgeräten verbessert ist. Erreicht wird dies durch die Kombination einer 160 × 120 Pixel (19.200 Pixel) großen Wärmebildkamera mit einer optischen 2-MP-Kamera. Wegen ihrer geringeren Auflösung liefern Kameras, die nur mit Wärmebildsensoren arbeiten, oft ein sehr unscharfes Bild.

Bilder, die mit solchen Geräten erstellt werden, können so schlecht sein, dass es unmöglich ist, den Inhalt des Bildes zu

```

from pymoku import Moku, StreamException
from pymoku.instruments import Datalogger
import time

m = Moku.get_by_name('Moku')

try:
    i = m.deploy_or_connect(Datalogger)

    # 100 samples per second
    i.set_samplerate(100)

    # Start data logger
    i.start_data_log(duration=10, use_sd=True, ch1=True, ch2=True,
                    filetype='bin')

    # Upload the log file to the local directory
    i.upload_data_log()

    # Clean up
    i.stop_data_log()

except StreamException as e:
    print("Error occurred: %s" % e)
finally:
    m.close()

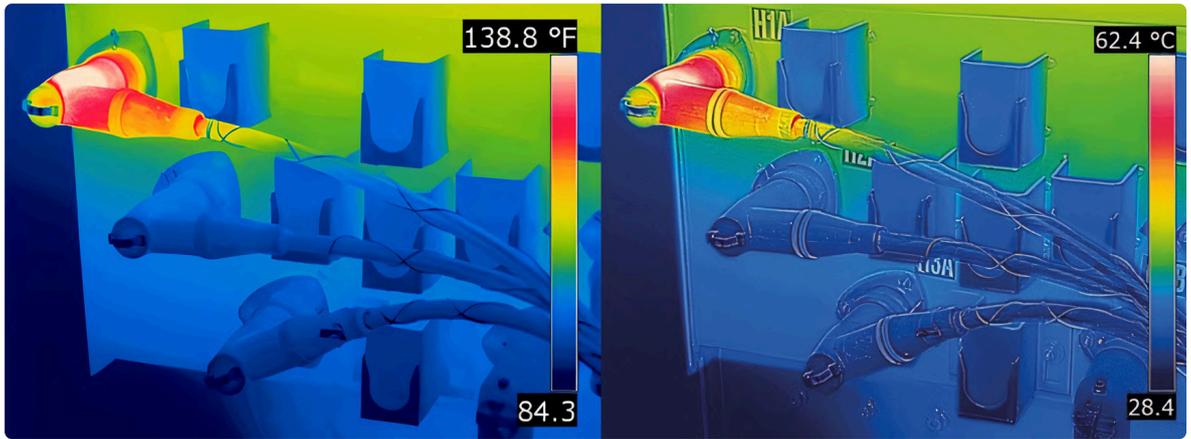
```

Bild 3. Beispiel für einen einfachen Datenlogger, der die Signale der beiden analogen Eingänge erfasst und der in Python für Moku:Lab geschrieben wurde.



Bild 4. Die Wärmebildkamera MR265 von Teledyne FLIR.

Bild 5. Ohne MSX® (links) fehlt die Beschriftung auf diesem Anschlussfeld. Mit MSX® (rechts) ist die Beschriftung klar und die Bildqualität insgesamt deutlich besser.



verstehen, ohne zu wissen, wo das Bild aufgenommen wurde. MSX® kombiniert die Wärmebilddaten mit dem kontrastreichen Bild der Kamera für sichtbares Licht (Bild 5). So werden die sichtbaren Details im Bild deutlicher und die Wärmemessung kohärenter. Diese Kameras werden auch von der FLIR Thermal Studio Suite [6] unterstützt, die sowohl Einzelbilder als auch Streaming-Videos erfassen kann. Für die Bildanalyse können Differenzen und Formeln eingesetzt werden; außerdem lassen sich bei Bedarf Messalarme definieren. Die Messergebnisse

können leicht mit Kollegen geteilt werden. Sofern unterstützt, kann auch Dual Streaming genutzt werden, sodass sowohl sichtbares Licht als auch radiometrische Videos (Wärmevideos) des gemessenen Objekts gespeichert werden können. Das MR265 bietet eine Messfrequenz von 9 Hz und misst Temperaturen von 0...100 °C (32 °F bis 212 °F) bei Objekten in einem Abstand von > 10 cm. Außerdem unterstützt das Gerät Feuchtigkeitsmessungen von 7 % bis 100 % über einen eingebauten Pinless-Sensor oder über separat erhältliche Kugel- und Pin-Sonden. Die Ergebnisse

werden auf dem 2,8-Zoll-Farb-TFT mit 320 × 240 Pixeln angezeigt.

Automatisierte Messung mit herkömmlichen Laborgeräten

Hochgradig konfigurierbare Headless-Messgeräte sind zwar in manchen Situationen attraktiv, aber die Notwendigkeit, ein zusätzliches Tablet oder einen Laptop zu verwenden, um die Ergebnisse anzuzeigen, kann zum Ausschlusskriterium für solche Geräte werden. Und manchmal ist das Altbekannte ganz einfach besser. Das Mixed-Signal-Oszilloskop 2560B von

Menu Bar
Access all feature and function settings by selecting a category from the menu bar.

10.1-inch Capacitive Touchscreen
Intuitive touchscreen gestures facilitate adjustment of waveform position and scale per division.

Advanced Triggering
Isolate the signal with advanced triggers including Edge, Slope, Pul Video (HDTV supported), Window, Interval, Dropout, Runt, Pattern and Serial.

Serial Bus Decoding
Edit serial bus decoding settings for I²C, SPI, UART, CAN and LIN. Additional protocols available with purchase of license key.

Arbitrary Waveform Generator Output
The 50 MHz waveform generator comes standard in all 2560B models. Output signal supports Bode plot function.

16-Channel Digital Port
Connect a logic analyzer probe to access 16 digital channels. Standard in MSO models and available as an upgrade for DSO models.

Two USB Host Ports
Connect USB flash drive to conveniently store and recall waveform data, setups and screenshots. Mouse and keyboard peripherals are supported as well.

Bild 6. Das MSO 2569B von B&K Precision bietet vier 350-MHz-Kanäle und einen digitalen 16-Bit-Logikanalysator.

B&K Precision [7] enttäuscht in dieser Hinsicht nicht: Es bietet einen großen kapazitiven 10,1-Zoll-Touchscreen mit Soft-Touch-Tasten und Drehknöpfen. Und mit einer Tiefe von nur 111 mm nimmt es nicht viel Platz auf dem Labortisch in Anspruch.

Das Spitzenmodell 2569B-MSO bietet vier Kanäle mit einer Bandbreite von 350 MHz und einen 16-Kanal-Digitalanschluss (**Bild 6**). Das Gerät verfügt außerdem über einen 50-MHz-Wellenform-Generator mit Bode-Plot-Funktion. Zudem bietet es erweiterte Triggeroptionen wie Flanken-, Flankensteilheits-, Impuls- und Videotrigger (einschließlich HDTV) sowie die Dekodierung serieller Busse für die gängigen Schnittstellen von Embedded-Systemen. Weitere serielle Schnittstellen wie CAN FD, FlexRay, I2S, MIL-STD-1553B und SENT können als Upgrade erworben werden. Fernsteuerung und Datenerfassung werden über die USB- und Ethernet-Schnittstellen unterstützt. Für USB wird ein USBTMC-Treiber der *Test & Measurement Class* benötigt. Hierfür wird das Produkt NI-VISA von National Instruments empfohlen [8], aber es gibt auch Alternativen. Auf GitHub beispielsweise findet sich eine Auswahl an Projekten, die USBTMC-Geräte in Python

unterstützen [9]. In Kombination mit dem Programmierhandbuch für die Serie 2560B [10] lassen sich Skripte entwickeln (**Bild 7**), mit denen die Datenerfassung und -protokollierung sowie die Dekodierung serieller Daten automatisiert, Messungen durchgeführt oder sogar Masken für Pass/Fail-Grenzprüfungen von Signalen definiert werden können.

Testen, testen, wieder testen

Für die meisten Entwickler ist die Arbeit am Prüfstand der Höhepunkt des Tages, obwohl die Suche nach Fehlern mit unklarer Ursache eine zeitraubende Aufgabe ist. Glücklicherweise können viele der heutigen Prüf- und Messgeräte (wie die hier vorgestellten) mit einer Software gekoppelt werden, um automatische Tests oder Datenaufzeichnungen durchzuführen. Aufgrund der Einfachheit und Flexibilität dieser Tools können innerhalb kurzer Zeit und oft mit Open-Source-Software einfache Skripte oder Prüfverfahren erstellt werden, um eine Vermutung zu überprüfen. Und wenn die Ursache des Fehlers gefunden ist, können diese Tests gespeichert und als Teil von *Continuous Integration Tests* eingesetzt werden, um sicherzustellen, dass der Fehler behoben bleibt. Die Ergeb-

nisse lassen sich auch leicht als Protokolldateien über Cloud-Services, E-Mail oder die Cloud-Plattformen des Anbieters weitergeben. So können die Teams qualifizierte Kollegen aus der ganzen Welt hinzuziehen, um die Ergebnisse gemeinsam zu überprüfen und zu bewerten. ◀

RG – 230189-02



Über den Autor

Stuart Cording ist ein freiberuflicher Journalist, der unter anderem für Mouser Electronics schreibt. Er ist auf Videoinhalte spezialisiert und konzentriert sich auf technische Hintergründe und Einblicke. Dabei interessiert er sich besonders für die Technologie selbst, ihre Einbindung in Endanwendungen und Vorhersagen über zukünftige Fortschritte. Mouser Electronics ist ein autorisierter Distributor für Halbleiter und elektronische Komponenten, der sich auf die Einführung neuer Produkte seiner führenden Herstellerpartner konzentriert.

```
import usbtmc
instr = usbtmc.Instrument(<VID>, <PID>, <SERIAL NUMBER>)
print(instr.ask("*IDN?"))
# returns 'BK Precision,2569B-MSO,XXXXXXXXXXXXXXXX,5.0.1.3.9R3'
```

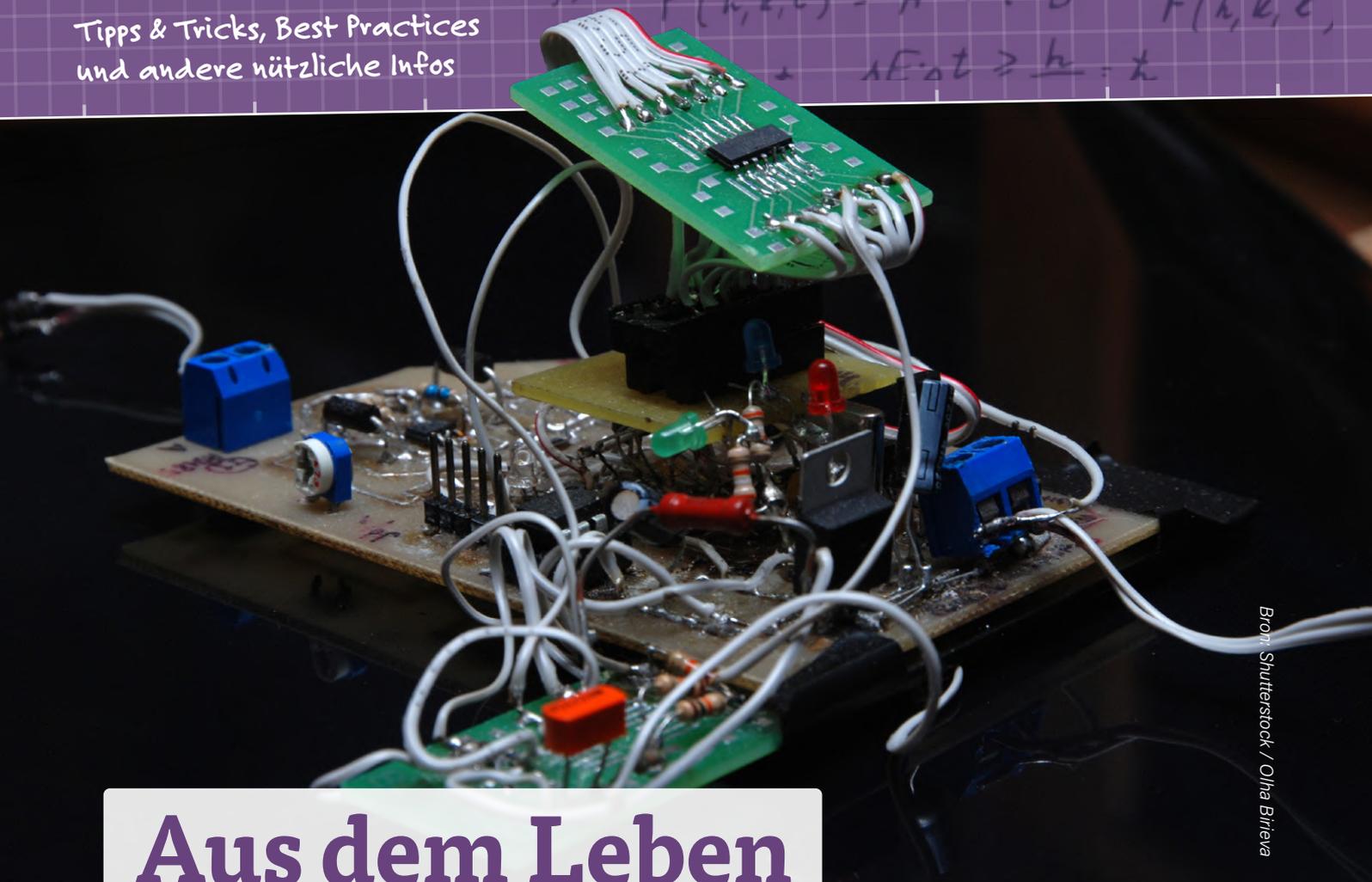
Bild 7. USBTMC-Unterstützung ist in Open-Source-Projekten zu finden. Dieser Python-Code greift auf das Oszilloskop über dessen VID und PID zu, um die Typenbezeichnung und die Seriennummer zu ermitteln.

WEBLINKS

- [1] Moku:Lab von Liquid Measurement: <https://bit.ly/3FI0BAW>
- [2] Pound-Drever-Hall-Verfahren: <https://de.wikipedia.org/wiki/Pound-Drever-Hall-Verfahren>
- [3] Liquid Instruments auf GitHub: <https://github.com/liquidinstruments>
- [4] MR265 : <http://bit.ly/3Zb4OUK>
- [5] Verbesserung der multispektralen dynamischen Bildgebung: <https://flir.com/globalassets/industrial/instruments/flir-msx-tech-note.pdf>
- [6] FLIR Thermal Studio Suite: <https://flir.eu/products/flir-thermal-studio-suite/>
- [7] Mixed-Signal-Oszilloskope der 2560-Reihe von B&K Precision: <http://bit.ly/40tFNW7>
- [8] National Instruments NI-VISA-Produkt: <https://ni.com/en-gb/shop/software/products/ni-visa.html>
- [9] USBTMC-Treiber in Python: <https://github.com/python-ivi/python-usbtmc>
- [10] Programmierhandbuch 2560B-Reihe: <https://bit.ly/40lx6gi>

Von Entwicklern für Entwickler

Tipps & Tricks, Best Practices
und andere nützliche Infos



Bron: Shutterstock / Olha Brieva

Aus dem Leben gegriffen

Elektronik auf hohem Niveau

Von Ilse Joostens (Belgien)

Vor etwa 15 Jahren bat mich mein damaliger Arbeitgeber, einen Aggressionsalarm zu bauen, weil sich einige Mitarbeiter nicht mehr sicher fühlten. Nach einer eher enttäuschenden Markterforschung hatte ich beschlossen, selbst etwas Drahtloses zusammenzuschustern. „Das schaffst du nie, kauf doch einfach etwas von der Stange“, sagte einer meiner technisch versierten Bekannten lächelnd, als das Thema aufkam.

Es hat mich Blut, Schweiß und Tränen gekostet, aber zum Leidwesen der besagten Person habe ich es geschafft. Mit Assembler und Manchester-Code, meine ich, denn damals konnte man sich noch keine Bibliotheken mit

Funkprotokollen aus dem Internet holen, und Arduino-Boards hatten noch nicht die Welt erobert. Dies veranschaulicht perfekt den aktuellen Trend, schnelle Ergebnisse ohne allzu viel Aufwand zu erzielen.

Bauklotzkisten

Vor einigen Jahren geriet ich in eine hitzige Diskussion mit einem frischgebackenen Master in angewandter Informatik. Der Mann hatte sicherlich seltsame Ansichten, und es schien, dass Arroganz und Herablassung Teil des Lehrplans seiner Ausbildung gewesen waren. Alles, was mit Firmware, Mikrocontrollern und anderen „Low-Level“-Zuständen zu tun hatte, gehörte seiner Meinung nach nicht in die Informatik, sondern in die Elektromechnik, ein Gebiet, auf das er verächtlich herabsah. Wahrscheinlich reagierte er so, um seine mangelnden Kenntnisse auf diesem Gebiet zu verbergen. Er war der Mann der hochentwickelten Betriebssysteme und Programmiersprachen [1]. Ich missgönne ihm seine Ideen nicht, aber es scheint mir keine angenehme Aussicht zu sein, beim Einschalten meiner Kaffeemaschine den Startvorgang eines ganzen Linux-Systems abwarten zu müssen, bevor das Ding anfängt, sich aufzuwärmen. Die Bequemlichkeit dient dem Menschen und es scheint, als ob sich der Drang, alles auf einer höheren Ebene machen zu wollen, auch in die Elektronik eingeschlichen hat. Ich gehöre noch zur alten Schule und entwerfe

meine Schaltungen normalerweise mit diskreten Bauteilen, aber im Internet und auch bei Elektor Labs [2] tauchen immer mehr Projekte auf, die nur aus einer einzigen Computerplatine und einer Reihe separater, vorgefertigter Module bestehen, wobei der Großteil der Funktionalität durch Software bereitgestellt wird. Oft besteht diese aus einem Mischmasch von Open-Source-Code und Bibliotheken, die wahllos aus dem Internet zusammengeklaut wurden. Wenn das Ganze dann so ziemlich das tut, was man sich vorgestellt hat, ist man scheinbar schon zufrieden - aber diese Arbeitsweise gleicht eher dem Spielen mit einer Kiste voller Bauklötze als dem Entwurf eines wirklich soliden Produkts. Vor kurzem habe ich mit ChatGPT [3][4] experimentiert, dem neuen Chatbot, der auf künstlicher Intelligenz basiert. Anscheinend kann dieses Programm auf Wunsch auch ganze Beschreibungen von elektronischen Schaltplänen und Quellcode ausspucken. Ob das alles korrekt ist, können wir nur vermuten, aber es ist ziemlich einfach und nett.

Nun bin ich überhaupt nicht gegen die Verwendung von Standardplatinen und Open-Source-Software. Ich bin der Erste, der ehrlich zugibt, dass ich während der Entwurfsphase manchmal auf vorhandene

Software und Breakout-Boards zurückgreife, aber das geschieht hauptsächlich, um etwas herauszufinden, etwas zu lernen, Dinge auszuprobieren oder um nicht gleich eine Platine entwerfen zu müssen, wenn es um superkleine SMD-Komponenten geht. Gleichzeitig können Module und Breakout-Boards von großem pädagogischen Wert sein, sowohl für fortgeschrittene Benutzer als auch für diejenigen, die in die wunderbare Welt der Elektronik eingeführt werden, selbst in jungen Jahren. Es gibt einige fortgeschrittene Sets, die manchmal buchstäblich wie ein Bauklötzkasten aussehen, sozusagen eine Hightech-Version des traditionellen Elektronik-Experimentierkastens.

Schmelzkleber

Ein befreundeter Gitarrist behauptet immer, dass das Leben eines Musikers zu drei Prozent aus Inspiration und zu siebenundneunzig Prozent aus Transpiration besteht, und meiner Meinung nach gilt das auch für die Elektronik. Sie können noch so gute Ideen haben, aber wenn Ihr Entwurf im Inneren wie ein Spinnennetz aus winzigen Platinen aussieht, das nur von einem Berg bunter Drähte zusammengehalten wird, wirkt er unprofessionell, ganz zu schweigen davon, ob die zusammengekratzte Software unter allen Umständen zuverlässig funktionieren wird. Es wird nicht das erste Gerät sein, dessen Benutzeroberfläche Ihnen nicht zusagt oder dessen Tasten „komisch“ reagieren. Auf diese Weise eignen Sie sich auch weniger Grundkenntnisse in Elektronik an, und es ist wahrscheinlich, dass Sie auch nicht (wirklich) verstehen, was diese Open-Source-Bibliothek tut oder wie sie genau funktioniert. Außerdem schaffen Sie auf diese Weise Abhängigkeiten, denn Ihr Projekt steht und fällt mit der Verfügbarkeit der verwendeten Module, und auch Software-Updates können unerwartete Hindernisse in den Weg legen. Der Trend zum modularen Design wird durch das massive Angebot an billigen Boards aus Fernost noch verstärkt. Jetzt, wo ich wegen der Bauteilknappheit selbst in der Klemme stecke,



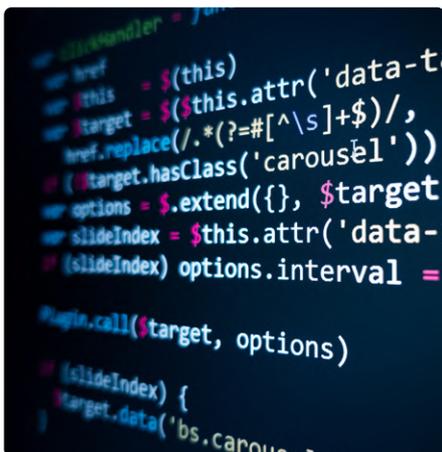
Bron: Shutterstock / Sinisa Botas

wurde mir auch schon mehrfach vorgeschlagen, ein Board von Onkel Ali einzubauen, aber ich bin noch nicht verzweifelt genug, um es wirklich in Erwägung zu ziehen.

Manchmal gibt es gute Gründe, ein vorgefertigtes Shield zu verwenden, aber richtig schlimm wird es, wenn kommerzielle Unternehmen sich in „Spinnwebkonstruktionen“ ergehen. Letztes Jahr wurde ich gebeten, mir ein medizinisches Gerät mit einem defekten USB-Anschluss anzuschauen. Als ich mich abmühte, das Gehäuse zu öffnen, stellte sich zu meiner Überraschung heraus, dass es einen billigen Arduino-Nano-Klon und ein ebenso billiges Board mit einem I²C-Drucksensor enthielt. Beide waren mit Schalllitze verbunden, die mit Heißkleber oder „heißem Rotz“ [5] zusammengebacken waren, wie es David Jones vom EEVBlog so plastisch ausdrückt. Auch die dazugehörige App war in Bezug auf Komplexität und Funktionalität nicht besonders spektakulär. Und das alles sollte ein paar hundert Euro kosten; reiner Profit oder mangelndes Wissen, oder beides? Ich überlasse es Ihrer Fantasie. ◀

SG – 230034-02

Anmerkung der Redaktion: Dieser Artikel wurde von einem Elektronikprofi geschrieben, und wir schätzen es, dass Ilse immer offen über ihre Perspektive spricht. Wir ermutigen jedoch alle unsere Leser, ihre Elektronikprojekte auf unserer Plattform Elektor Labs zu stellen, egal in welchem Stadium sie sich befinden. Auch Prototypen aus Bauklötzen, einfache Projektskizzen und auch bloße Ideen sind willkommen!



Bron: Shutterstock / jivacore

WEBLINKS

- [1] Programmiersprache bei Wikipedia: <https://de.wikipedia.org/wiki/Programmiersprache>
- [2] Elektor Labs: <https://www.elektormagazine.de/labs>
- [3] OpenAI, Forschungsbericht ChatGPT: <https://chat.openai.com>
- [4] OpenAI, Erste Schritte mit Codex: <https://help.openai.com/en/articles/6195637-getting-started-with-codex>
- [5] EEVblog: Sony Boombox REPAIR and Teardown: <https://youtu.be/8ToVCAhLg8A?t=487>

Energielogger

Energie messen und aufzeichnen

Von Georg Luber (Deutschland)

Es gibt viele potentielle Anwendungen, wo eine lückenlose Aufzeichnung der Erzeugung oder des Verbrauchs elektrischer Energie sinnvoll ist. Man denke nur an den Energieverbrauch bestimmter Stromkreise im Haus oder aber an die Energie, welche ein Balkonkraftwerk oder auch eine größere Solaranlage liefert. Zur Erfassung gibt es fertige Lösungen, doch man kann einen passenden Logger auch selbst bauen.



Bild 1. Der fertige Energielogger mit Zähler und Netzteil, eingebaut in einen kleinen Kunststoff-Verteiler.

Der hier vorgestellte Logger verwendet zur Erfassung elektrischer Energie einen Zähler mit So-Ausgang. Geeignete Zähler gibt es für weniger als 20 Euro zum Beispiel bei Amazon, eBay oder direkt aus Fernost. Damit kann der Energieverbrauch durchgeschleifter Stromkreise oder auch der Ertrag von Solaranlagen

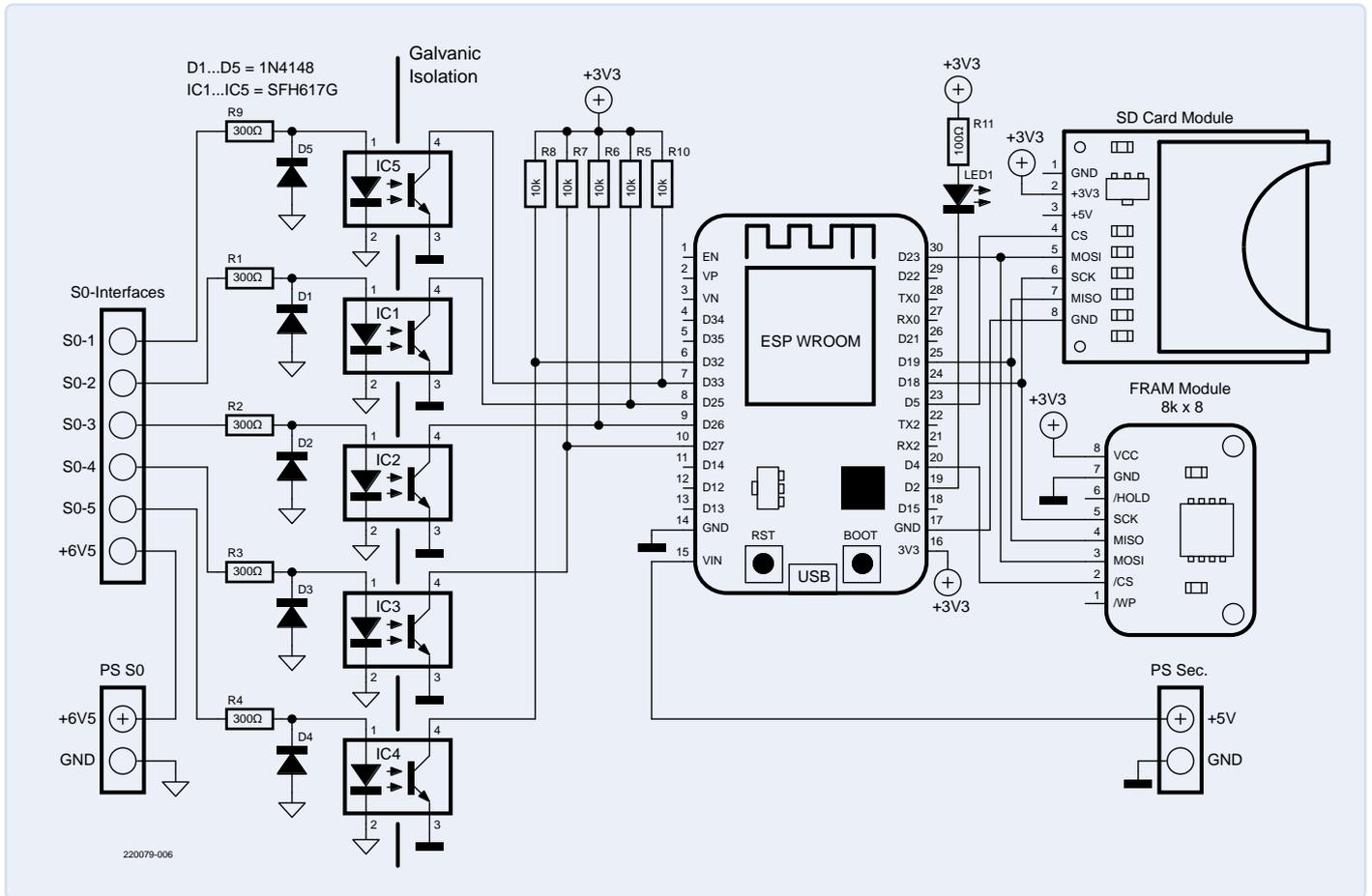
gemessen und mit der hier beschriebenen Elektronik „geloggt“ beziehungsweise aufgezeichnet werden. Für was hat der technische Fortschritt Mikrocontroller geschaffen? Ziemlich genau für solche Anwendungen. Vor dem computerisierten Zeitalter wäre die Aufzeichnung von Energieverbrauch oder -erzeugung über die Zeit nur schwer und für den Hausgebrauch praktisch unmöglich gewesen. Dank preiswerter μC mit all ihren tollen Fähigkeiten ist nicht nur die digitale Registrierung kein Problem, es werden auch schicke Dinge wie eine WLAN-Anbindung und anderes mehr möglich. Was der Energielogger alles kann, findet sich im **Kasten Eigenschaften**. Einen Eindruck von der fertigen Lösung vermittelt die in einen Kunststoff-Verteiler eingebaute Kombination von Zähler und Logger samt Netzteilen in **Bild 1**.

Schaltung

Dank Mikrocontroller ist die Schaltung des Energieloggers (**Bild 2**) ziemlich einfach: Links sitzen zwecks Potentialtrennung eine Reihe von Optokopplern, über die Daten der Zähler-So-Schnittstellen auf die entsprechenden Eingänge des Mikrocontrollers gelangen. Diese sind Interrupt-gesteuert, damit kein Impuls

Eigenschaften

- › Genau und sicher durch Zähler mit So-Schnittstelle und galvanischer Trennung.
- › Konfigurierbare Erfassung von bis zu fünf Zählern.
- › Zwischensicherung der Daten im FRAM zur Schonung der SD-Karte.
- › Aufzeichnung auf SD-Karte im CSV-Format.
- › Zeitliche Auflösung: 5 Minuten.
- › Pro Tag wird eine eigene Datei gesichert.
- › Anzeige einzelner und aggregierter Daten (aktuell und vom Vortag in kWh)
- › Anzeige von WLAN-Feldstärke und Anzahl der μC -Resets (Startzähler).
- › Download der Daten via FTP.
- › OTA: Update-Fähigkeit via WLAN.
- › Entwicklungsumgebung: VS-Code mit PlatformIO.



verloren geht. Ich habe auf ein preiswertes ESP32-Modul gesetzt, da es neben ausreichender Leistung auch noch ein WLAN-Interface mitbringt. Auf der rechten Seite befinden sich einmal ein Slot für eine SD-Karte als Massenspeicher und zusätzlich ein FRAM-Modul. Dieses dient als Zwischenpuffer für die Daten von fünf Minuten, um die Anzahl der Schreibzyklen für die SD-Karte zu begrenzen. Da die Schaltung so einfach ist, habe ich sie auf einer Lochrasterplatte aufgebaut. Wie zu sehen ist, sind aus Sicherheitsgründen zwei getrennte Netzteile vorgesehen. Für die SO-Schnittstellen links reichen 6,5 V aus; der Rest der Schaltung wird mit einem 5-V-Netzteil (rechts) betrieben. Eine Belastbarkeit von jeweils 0,5 A reicht aus. Zwecks Potentialtrennung dürfen die beiden GND-Leitungen der Netzteile natürlich keinesfalls verbunden werden. Außerdem muss unter den Optokopplern – zwischen deren Ein- und Ausgängen – alles Kupfer über eine Strecke von mindestens 4 mm entfernt werden. **Bild 3** zeigt, wie das gemeint ist. Die galvanische Trennung durch zwei separate Netzteile erlaubt, dass der ESP32 auch an die USB-Schnittstelle eines PC angeschlossen werden kann, um zum Beispiel nachträgliche Updates im eingebauten Zustand zu ermöglichen. Statt zweier Netzteile wäre auch eine Lösung mit nur einem Netzteil mit höherer Strombelastbarkeit plus einem isolierenden DC/DC-Konverter mit nachfolgendem Spannungsregler möglich gewesen, was die Sache

aber nicht wirklich einfacher und auch nicht preiswerter machen würde.

Speichern und Datenübertragung

Die doppelten Speicher sind, wie schon erwähnt, kein Luxus. Die SO-Schnittstellen können nämlich auch mehrere Impulse pro Sekunde liefern. Würden die auftretenden Messwerte nun in einem internen Speicher des Mikrocontrollers gesammelt, könnten bei einem Reset Daten verloren gehen. Diese aber sofort

Bild 2. Die Schaltung des Energieloggers ist ziemlich einfach.

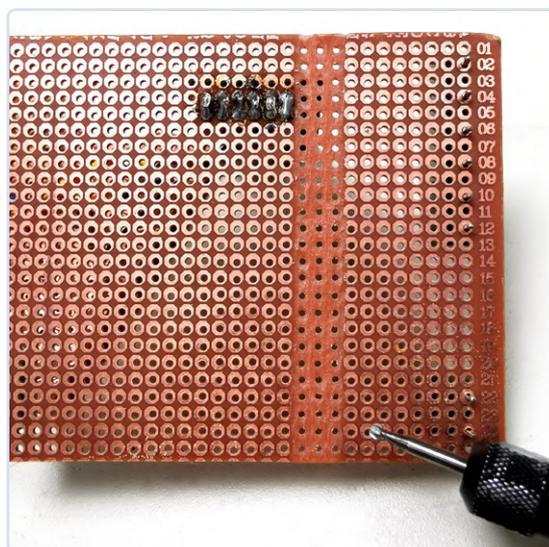


Bild 3. Das Kupfer auf der Unterseite der Lochrasterplatte zwischen den Ein- und Ausgängen der Optokoppler wurde mit einem kleinen Fräser entfernt.

	A	B	C	D	E	
1	/2021/07/31.csv - 00:00					
2		426572	415390	132309	129324	218375
3	05:55					
4	>					
5		426572	415390	132309	129324	218375
6		426574	415391	132309	129324	218375
7		426575	415393	132309	129324	218375
8		426575	415393	132309	129324	218375
9		426576	415394	132309	129325	218375
10		426577	415396	132310	129326	218377
11		426579	415398	132310	129326	218378
12		426583	415401	132310	129326	218380
13		426587	415405	132310	129326	218382

Bild 4. Die Datenstruktur wird sichtbar, wenn man eine CSV-Datei in eine Tabellenkalkulation importiert.

auf eine SD-Karte zu schreiben würde deren Lebensdauer massiv verkürzen. Bei nur einem Wert pro Sekunde kämen pro Jahr immerhin schon 31,5 Millionen Schreibzyklen zusammen. Speicherzellen einer SD-Karte kommen aber schon nach 1.000...3.000 Schreibzyklen an ihr Lebensende. Auch eine Karte mit großer Kapazität würde daher schon vor Ablauf eines Jahres mit ziemlicher Sicherheit Defekte aufweisen. Um das zu vermeiden, ist ein stabiler Zwischenspeicher vorgesehen. Das hier verwendete externe FRAM-Modul besitzt zwar gerade mal 8 kB Speicher, was aber locker ausreicht, um etliche Werte zu sammeln. Der größte Vorteil von FRAM ist, dass es je nach Hersteller mindestens 10^{10} mal beschreibbar ist – typischerweise trägt solch ein Speicher sogar Billiarden Schreibzyklen. Alle fünf Minuten werden

die im FRAM gesammelten Daten auf die SD-Karte übertragen. Das gibt nur noch knapp über 100.000 Schreibvorgänge pro Jahr, und da eine SD-Karte viele Speicherzellen hat, kann sie so etliche Jahre ohne Probleme betrieben werden.

Zusätzlich werden im FRAM die Anzahl der Neustarts des Loggers festgehalten. Damit kann man jederzeit sehen, wie oft ein Reset ausgelöst wurde. Weiter werden auf der SD jeden Tag eine neue Datei angelegt und die Daten dieses Tages darin gesichert. Um Speicherplatz zu sparen, wird die Aufzeichnung erst begonnen, wenn sich einer der Zählwerte seit Mitternacht geändert hat. Betreibt man den Energielogger zur Verbrauchserfassung, dann ist dieses Feature eigentlich überflüssig. Wird damit aber die Energieproduktion einer Solaranlage überwacht, ist es praktisch, wenn die Aufzeichnung erst nach Sonnenaufgang startet, also wenn der erste Strom erzeugt wird.

Datenstruktur und mehr

Bild 4 zeigt einen Ausschnitt einer Tabellenkalkulation, bei der die Daten im CSV-Format importiert wurden. Zelle A1 enthält Datum und Uhrzeit der Datei-erstellung. In Zeile 2 stehen dann die letzten Werte des Vortages als neuer Startpunkt. Die erste Änderung eines Zählwertes erfolgte um 05:55 Uhr (Zelle A3). Ab Zeile 5 folgen dann die neuen Daten dieses Tages. Die Spalten enthalten die Daten der einzelnen Zähler. Jede Zeile hat einen zeitlichen Abstand von fünf Minuten zur nächsten Zeile. Aufgezeichnet wird an jedem Tag bis jeweils 24:00 Uhr. Täglich wird wieder eine neue Datei angelegt. Aus der Differenz der gezählten Werte kann dann über die Impulswertigkeit der Zähler (zum Beispiel 0,5 Wh/Impuls) die Energie berechnet werden. Das CSV-Format ist sparsam und eignet sich zum Import in jede Tabellenkalkulation wie Excel, OpenOffice, Libre Calc oder Numbers beim Mac. Die Daten können dann nach eigenem Gusto weiterverarbeitet werden.

Auf der SD-Karte werden die Daten in einer temporalen Ordner-Struktur abgelegt. Auf oberster Ebene befinden sich die Ordner mit den Jahren, welche dann Ordner mit den Monaten und diese dann die Dateien der einzelnen Tage enthalten. **Bild 5** zeigt, wie das aussieht.

In diesem Beispiel werden die Werte von fünf Zählern ganzzahlig erfasst (siehe Bild 4). Dank des CSV-Formats wird hierfür relativ wenig Speicher benötigt. Pro Zähler kommt man so auf nur wenige Kilobyte pro Tag.

Name	Größe	Geändert
2021	9 Elemente	07.01.22 21:06
2022	11 Elemente	02.11.22 22:25
2023	1 Element	08.01.23 10:06
01	17 Elemente	17.01.23 17:21
01.CSV	7,8 KiB	08.01.23 10:07
02.CSV	7,8 KiB	08.01.23 10:07
03.CSV	7,7 KiB	08.01.23 10:07
04.CSV	7,8 KiB	08.01.23 10:07

Bild 5. Die Ordnerstruktur nach Jahr, Monat und Tag.

Die Übertragung der Daten vom Logger zu einem PC erfolgt via FTP. Hierzu eignen sich gängige FTP-Clients wie etwa FileZilla. Da der Logger per WLAN ins eigene Heimnetz integriert ist, stellt diese Methode wohl das einfachste Verfahren dar.

Den Source-Code für den ESP32 habe ich mit Hilfe der IDE PlatformIO für VSCode [1] erstellt. Er steht auf der Webseite zu diesem Artikel [2] kostenlos zum Download zur Verfügung.

Folgende Daten sind in der Datei `main.cpp` individuell zu ändern: Ab Zeile 30 muss die SSID und das Passwort des WLAN eingetragen werden. Der Benutzername und das Passwort für FTP sind in Zeile 813 angegeben. Standardmäßig ist hier als Benutzername und Passwort jeweils „esp32“ eingetragen. Natürlich braucht der FTP-Client noch die IP-Adresse des Energieloggers. Diese wird nach einem Reset auf der USB-Schnittstelle ausgegeben. Idealerweise legt man aber im Router fest, dass der Energie-Logger immer die gleiche IP-Adresse erhält.

Integrierte Webseite

Die aktuellen Daten sowie der Gesamtwert des Tages und des Vortages werden mit Hilfe des integrierten Web-Servers dargestellt. Der Server hostet zwei Seiten: Eine Seite zeigt die aktuelle Leistung pro Zähler sowie die Summe aller angeschlossenen Zähler. Außerdem wird die Gesamtenergie des Tages und des Vortages in kWh ausgegeben. Die andere Seite zeigt die Feldstärke des WLANs und die Anzahl an Starts des Loggers.

Bild 6 und **Bild 7** sind Screenshots von einem Smartphone. Wie Sie sicher sofort merken, setze ich den Energielogger nicht zur Verbrauchsmessung, sondern zur Überwachung meiner Solaranlage ein, die über fünf Wechselrichter verfügt. Von daher habe ich die Beschriftungen der einzelnen Werte meinem Einsatzzweck angepasst. Sie können das aber leicht in der Software nach Bedarf ändern. Beim Aufruf der IP-Adresse im Browser erhält man die Anzeige von Bild 6. Durch ein an die Adresse angehängtes „/about“ oder durch Klick auf die Schaltfläche *Info* erscheint Bild 7.

Noch ein Hinweis zur WLAN-Verbindung: Wenn Sie den Energielogger in ein Metallgehäuse einbauen, bekommen Sie ziemlich sicher Empfangsprobleme...

OTA

Wenn der Energielogger einmal in ein Gehäuse eingebaut und installiert ist, will man die Elektronik bei einer Softwareänderung nicht gerne wieder ausbauen.

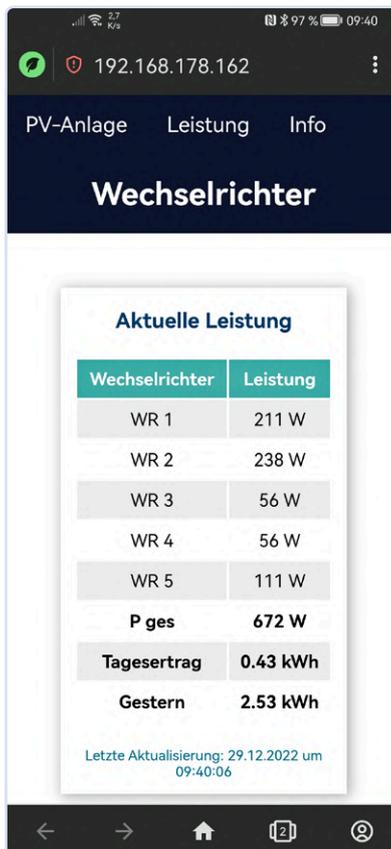


Bild 6. Die integrierte Webseite zeigt die Messergebnisse an.

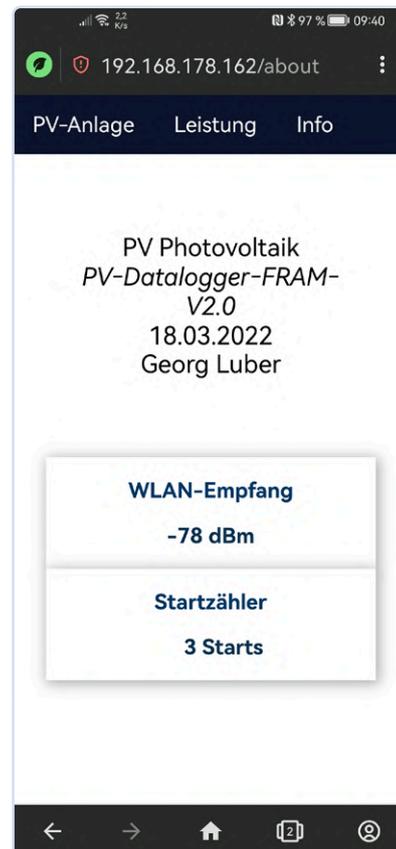


Bild 7. Bei an die IP-Adresse angehängtem „/about“ oder einem Klick auf Info erhält man diese Angaben.

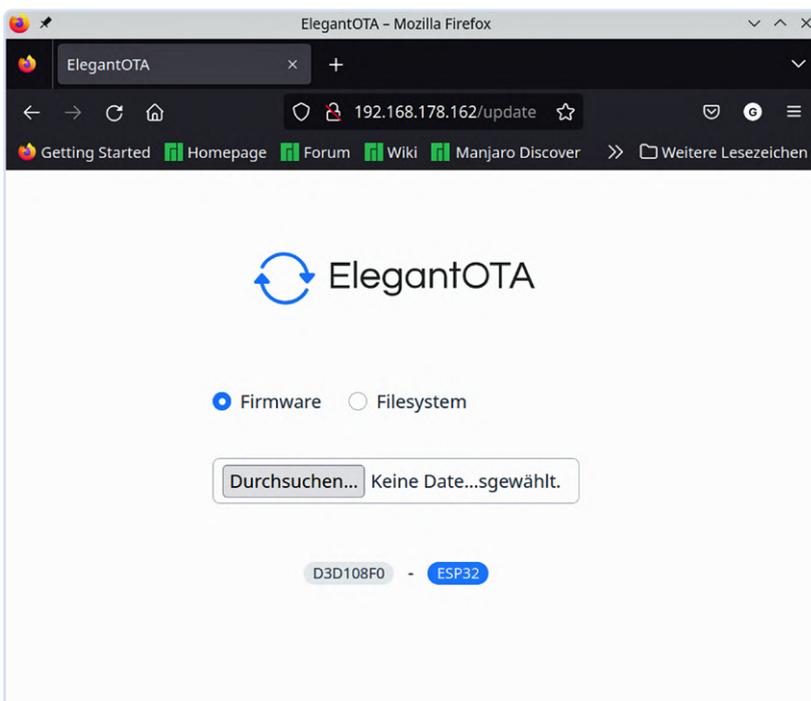


Bild 8. Updates sowohl der Firmware als auch der Datenstruktur sind per WLAN möglich.



Bild 9. Vier Zähler des Typs DDS5188 im Betrieb.

Deshalb kann der Datalogger „**Over The Air**“, das heißt via WLAN aktualisiert werden. Die komplette Software sowie das Dateisystem – die Webseiten mit Javascript und die CSS-Datei – können so bei Bedarf neu aufgespielt werden.

Will man das tun, ruft man die IP-Adresse mit angehängtem „/update“ auf. **Bild 8** zeigt, wie das bei mir aussieht. Die OTA-Fähigkeit habe ich mit Hilfe der Bibliothek von Ayush Sharma [3] realisiert.

Abschließend

Die Schaltung ist so einfach, dass sich eine explizite Stückliste erübrigt. Wichtige Bauteile sind neben dem überall erhältlichen ESP32-Entwicklungsboard ein SD-Karten-Slot, den es preiswert auf Breakout-Boards gibt, und das erwähnte FRAM-Modul – ebenfalls praktischerweise auf Breakout-Boards verfügbar. Da Breakout-Boards mit unterschiedlichen Pin-Belegungen erhältlich sind, sollte man beim Anschluss nicht auf die Pin-Nummern, sondern auf die Pin-Bezeichnungen achten. Beim SD-Karten-Modul kommt gegebenenfalls MOSI an DI oder SI und MISO an DO oder SO. Als Zähler eignen sich etliche preiswerte Modelle für Hutschienenmontage mit So-Schnittstelle. Der von mir verwendete Typ DDS5188 (**Bild 9**) ist leicht erhältlich und sehr preiswert.

Es sollte sich zwar von selbst verstehen, dass man bei Arbeiten am Netz die nötige Sorgfalt walten lassen muss, aber ein Hinweis sei erlaubt: Für Arbeiten am Sicherungskasten gelten unterschiedliche nationale Gesetze. In Deutschland müssen Sie hierzu als ausgebildeter Elektriker beim Netzbetreiber registriert sein.

Auch die Arbeit an externen, an den Sicherungskasten angeschlossenen Unterverteilern ist einer Elektrofachkraft mit entsprechender Ausbildung und Wissen vorbehalten. ◀

TS - 220079-02

Über den Autor

Georg Luber ist ausgebildeter Elektriker, hat dann Elektrotechnik studiert und war jahrelang in den Bereichen Elektrische Sicherheit und Elektroinstallation tätig. Er hat national und international in entsprechenden Normengremien (DKE/VDE, CENELEC, IEC und ISO) mitgearbeitet. Ein weiterer Schwerpunkt war die Gebäudeautomation und hier insbesondere KNX. Georg Luber beschäftigt sich in diesen Bereichen mit Software und der Entwicklung von Elektronikprojekten.

Sie haben Fragen oder Kommentare?

Haben Sie technische Fragen oder Anmerkungen zu diesem Artikel? Schicken Sie eine E-Mail an Elektor unter redaktion@elektor.de.



Passende Produkte

- ▶ **ESP32-DevKitC-32D**
SKU 18701: www.elektor.de/18701
- ▶ **Erik Bartmann, Das ESP32-Praxisbuch**
Buch, kartoniert, SKU 18572: www.elektor.de/18572
E-Buch, PDF, SKU 185733: www.elektor.de/18573
- ▶ **PeakTech Stromzange 4350**
SKU 18161: www.elektor.de/18161



WEBLINKS

- [1] PlatformIO für VScode: <https://platformio.org/platformio-ide>
- [2] Webseite zum Artikel: <https://www.elektormagazine.de/220079-02>
- [3] Bibliothek von Ayush Sharma: <https://github.com/ayushsharma82/ElegantOTA>
- [4] ESP32-Projekte: <https://randomnerdtutorials.com/projects-esp32>

Aufgebaut: Das 4tronix M.A.R.S. Rover Kit

Von Clemens Valens (Elektor)

Inspiriert von den Mars-Rovern Curiosity und Perseverance der NASA ist der M.A.R.S. Rover der Firma 4tronix ein autonomes Fahrzeug, das sich in unwegsamem Gelände auf der Erde statt auf dem Mars fortbewegen kann. Tatsächlich bezieht sich M.A.R.S. in diesem Fall nicht auf den Roten Planeten, sondern ist ein Akronym für **M**obile **A**utonomous **R**obotic **S**ystem. Welcher Zufall!

Der M.A.R.S.-Rover wird als Bausatz geliefert, muss aber nicht gelötet werden. Der Bausatz ist in sich geschlossen, inklusive Werkzeug, obwohl ich lieber meine eigenen Schraubendreher benutzt habe, da sie besser passen. Es gibt zwei Versionen des Bausatzes: mit Raspberry Pi Zero und mit BBC micro:bit. Ich habe die Raspberry-Pi-Version ausprobiert. Beachten Sie, dass der Raspberry Pi Zero oder BBC micro:bit nicht im Lieferumfang enthalten sind, ebenso wenig wie die vier AA-Batterien. Für den Raspberry Pi benötigen Sie außerdem eine microSD-Karte.

Auf der 4tronix-Website [1] wird ein zusätzliches Tastenfeld angeboten, aber ich habe es nicht ausprobiert. Die Raspberry Pi-Version des Rovers ermöglicht die Montage einer Raspberry-Pi-Kamera am „Mast“ des Ultraschallwandlers, aber auch das habe ich nicht ausprobiert.

Die Schachtel mit dem Bausatz ist erstaunlich kompakt (18 cm × 16 cm × 8 cm), aber ziemlich schwer (600 g) und bis zum Rand mit Plastiktüten gefüllt, die Teile enthalten. Ein gedrucktes Handbuch ist nicht enthalten, aber ausführliche Bau- und Programmieranleitungen sind auf der 4tronix-Website zu finden. Sie benötigen also einen Computer mit Internetanschluss, um den Rover zusammenzubauen.

Die Bauanleitung ist sehr klar und detailliert. Der Zusammenbau des Rovers ist nicht schwierig, aber es gibt einige knifflige Dinge wie das Anschrauben der Motorhalterungen, ohne dass die Muttern herunterfallen, oder das Befestigen einiger Sicherungsmuttern oder das Anschließen der Motoren. Achten Sie genau auf die Ausrichtung und Positionierung bestimmter Teile wie der Drehgestelle und der Servos, denn man kann sich leicht vertun und muss dann alles noch einmal machen. Ich erwähne dies, weil mir dies ein paar Mal passiert ist.

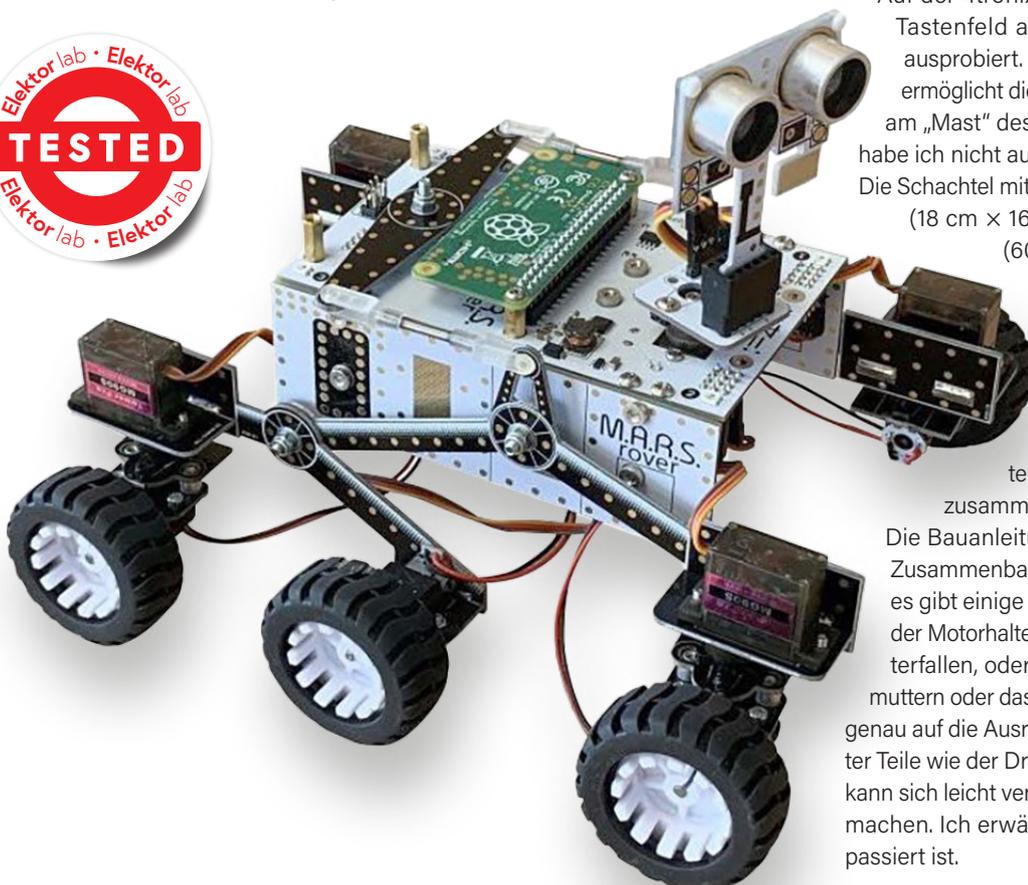


Bild 1. Bringen Sie Ihre eigenen Schraubendreher mit.

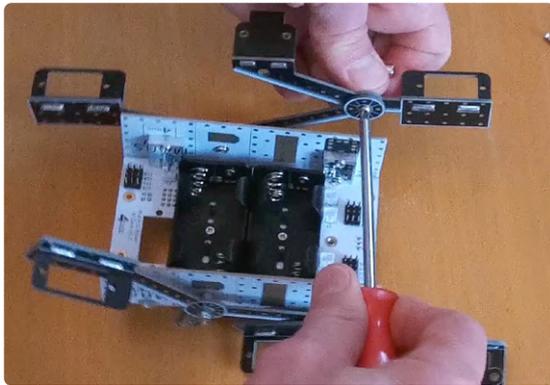


Bild 2. Alle Servoanschlüsse sind gleich ausgerichtet, eindeutig beschriftet und die Servodrähte farblich gekennzeichnet. Braun ist GND, gelb ist SIG (Signal).

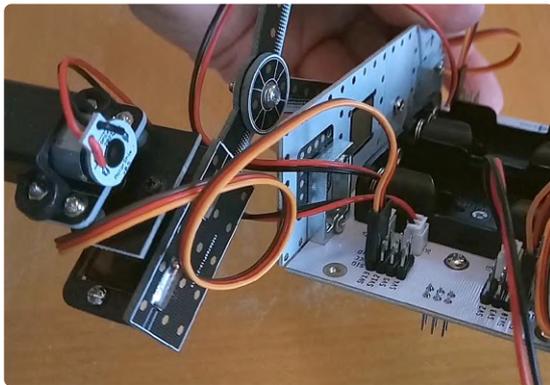


Bild 3. Das Drehgestell am Kipphebel sollte sich fast ohne Reibung drehen lassen.



Tipps zum Zusammenbau

Der Zusammenbau des Rovers dauerte etwa zwei Stunden, da ich einige Fehler gemacht habe. Hier sind ein paar Tipps:

- Besorgen Sie sich ein paar „richtige“ Schraubendreher (**Bild 1**)
- Es gibt drei 18 mm lange Schrauben, die durch die Platinen gehen. Achten Sie darauf, dass Sie sie gut anziehen, bevor Sie das Teil auf der anderen Seite mit einer Kontermutter befestigen, da es sich sonst lösen kann, weil Sie später die Mutter einstellen müssen, damit sich die befestigten Teile reibungslos drehen lassen.
- Alle Servoanschlüsse sind in der gleichen Weise ausgerichtet. Sie sind eindeutig beschriftet, und die Servodrähte sind farblich gekennzeichnet (**Bild 2**). Braun ist GND und gelb ist SIG (Signal). Notieren Sie sich die Nummer jedes Servos (SVx), wenn Sie das Programm *servoTest* ausführen.

- Der Mast des Ultraschallwandlers kann sich mit dem Servokabel verheddern, wenn dieses nicht ausreichend befestigt ist
- Nachdem Sie die Servos angeschlossen haben, können Sie ihre Drähte um die Servokörper wickeln, um die Sache etwas aufzuräumen. Lassen Sie dagegen die Kabel der Radmotoren frei, damit sie die Bewegungen der Kipphebel nicht blockieren (**Bild 3**).
- Es ist nicht erforderlich, die überstehenden Enden der Servoarme abzuschneiden, nur damit der Rover etwas „ordentlicher“ aussieht

Hinzufügen der „Gehirne“

Nachdem ich den Rover zusammengebaut hatte, habe ich darauf einen Raspberry Pi Zero 2 W montiert. Ich verwendete eine microSD-Karte mit dem installierten Raspberry Pi Buster OS. Diese Karte war bereits in einem anderen Projekt im Einsatz, sodass der Raspberry Pi direkt eine Verbindung zu meinem WLAN herstellen konnte und auch der SSH-Zugang schon aktiviert war. Bei einem neuen Pi müssen Sie dies natürlich zuvor einstellen. Um die Batterien zu schonen, habe ich den Raspberry Pi über USB mit Strom versorgt, um die Software herunterzuladen und zu installieren, damit der Rover läuft. Auch hier ist die Anleitung klar und detailliert, allerdings nur online verfügbar [2].

Ich habe auch ein Raspberry Pi Zero anstelle eines Zero 2 W ausprobiert und festgestellt, dass er genauso gut funktioniert. Der Vorteil eines Zero 2 ist natürlich, dass das Board viel schneller bootet.

Der Raspberry Pi versorgt den Rest des Rovers nicht mit Strom - dafür braucht man die Batterien. Wenn Sie also beim Ausprobieren der Beispiele eine Fehlermeldung bezüglich des I²C-Busses erhalten, aber sicher sind, dass Sie ihn auf dem Raspberry Pi aktiviert haben, dann überprüfen Sie den Ein/Aus-Schalter und Ihre Batterien. Eine blaue LED zeigt an, ob der Rover eingeschaltet ist oder nicht.

Ich habe eine Stromaufnahme von etwa 400...500 mA gemessen, wenn der Rover mit dem *motorTest*-Sketch fährt, und etwa 200 mA im Leerlauf.

Programmierung des Rovers

Wenn man eigene Programme für den Rover entwickelt [3], ist es wahrscheinlich am besten, so viel wie möglich den Raspberry Pi über USB zu versorgen, um die Batterien zu schonen, und den Rover nur einzuschalten, wenn man etwas ausprobieren möchte.

Ich habe alle von 4tronix [2] zur Verfügung gestellten Beispiele ausprobiert und fand, dass alles ohne Probleme funktionierte. Ich habe selbst nichts Neues programmiert. Beachten Sie, dass die Steuerung der RGB-LEDs des Rovers aus irgendeinem Grund Administratorrechte erfordert, so dass jedes Skript, das sie verwendet, mit `sudo` ausgeführt werden muss.

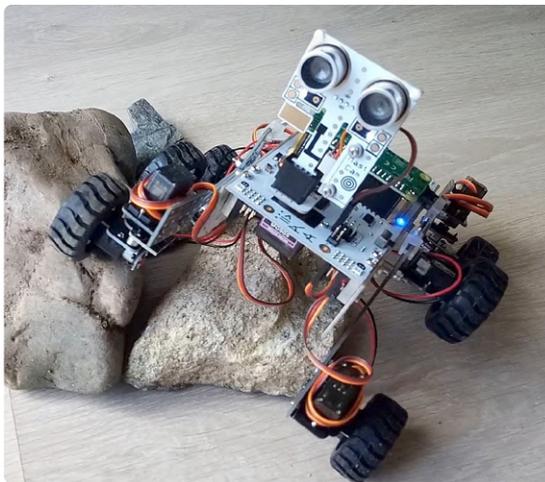


Bild 4. Einmal zusammengebaut, ist der M.A.R.S.-Rover fast nicht mehr zu stoppen.

Der Rover ist ziemlich beeindruckend, wenn er über Hindernisse fährt (**Bild 4**). Er fährt zwar nicht sehr schnell, aber die Kipphebel ermöglichen es dem Roboter, Hindernisse zu meistern, als wären sie gar nicht da.

Fast unaufhaltsam

Der M.A.R.S.-Rover-Bausatz von 4tronix ist ein hochwertiger Bausatz, der nach dem Zusammenbau dank seines dekorativen Drucks und des Platinendesigns ziemlich cool aussieht. Er misst 20 cm x 18 cm x 16 cm und ist damit etwas größer als der Karton, in dem er geliefert wird. Der Zusammenbau des Roboters ist einfach, da alles perfekt passt (was ziemlich selten ist) und die Anleitung klar und detailliert ist. Auch die Benutzung ist einfach, und die Beispiele und Testprogramme lassen sich problemlos installieren und ausführen.

Sobald der Rover zusammengebaut und die Software installiert ist, kann er losfahren und ist fast nicht mehr zu stoppen. Ein paar Programmierkenntnisse sind dazu natürlich erforderlich, aber dafür haben Sie ihn ja schließlich gekauft, oder?

Erweiterung des Bausatzes

Die Hauptplatine weist vier zusätzliche Befestigungslöcher auf, die zur Anbringung von Erweiterungen genutzt werden

können, und es gibt noch genügend Anschlüsse, um weitere Servos hinzuzufügen (im „Ersatzteillager“ meines Bausatzes war übrigens ein zusätzlicher Servo enthalten). Ich habe keinen Schaltplan gefunden, daher weiß ich nicht, ob die Servoanschlüsse für etwas anderes verwendet werden können. Es gibt auch zwei I²C-Erweiterungsstecker, an die man zum Beispiel Sensoren anschließen kann. **◀**

RG – 220107-02



Passende Produkte

- > 4tronix M.A.R.S. Rover Roboter-Kit für Raspberry Pi Zero
 SKU 19996: www.elektor.de/19996

WEBLINKS

- [1] Zusammenbau des M.A.R.S.-Rovers: <https://4tronix.co.uk/blog/?p=2112>
- [2] Programmierung des M.A.R.S.-Rovers auf dem Raspberry Pi Zero (inc v2): <https://4tronix.co.uk/blog/?p=2409>
- [3] MARS-Rover [GitHub]: <https://github.com/4tronix/MARS-Rover>
- [4] „Assembling the 4tronix M.A.R.S. Rover Kit“, dieses Review im Video: <https://youtu.be/Np8ZQQd85oc>

CTX THERMAL SOLUTIONS | 25 JAHRE

KEEP IT COOL



TALK TO US @ **pcim** EUROPE
 9.–11. Mai 2023, Nürnberg

Effiziente Kühllösungen für industrielle Elektronikanwendungen

www.ctx.eu

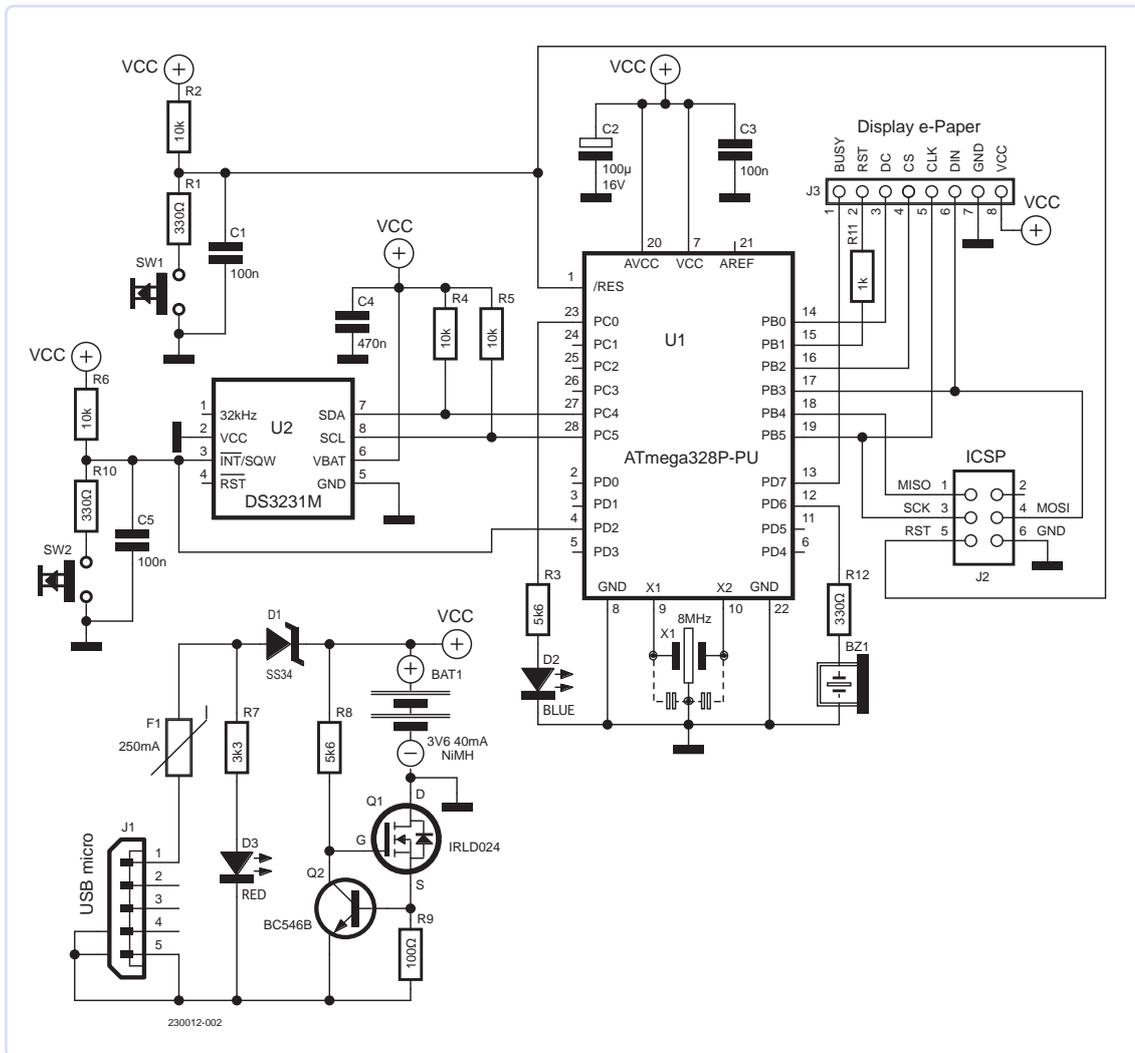


Bild 2. Schaltbild der Elektronik.

zu unterscheiden sind und die bei Bedarf auch aus der Ferne über Funk oder Satellit aktualisiert werden können. Um zu verstehen, wie die elektronische Tinte funktioniert, hilft uns **Bild 1**. Im einfachsten Schwarz-Weiß-Display sind in Mikrokugeln positiv (weiß) und negativ (schwarz) geladene Pigmente in einer Flüssigkeit suspendiert. Die Mikrokugeln stellen die Pixel des Displays dar. Wird ein geeignetes elektrisches Feld angelegt, positionieren sich die Pigmente, die von der Ladung mit entgegengesetztem Vorzeichen angezogen werden (Elektrophorese), so dass – von oben betrachtet – schwarze oder weiße Pixel entstehen, die das gewünschte Bild bilden. Auch wenn das elektrische Feld entfernt wird, bleiben die Pigmente in dieser Position, bis eine neue Ladung angelegt wird. Die Sichtbarkeit des besonders weiten Blickwinkels wird durch die Reflexion des Umgebungslichts erreicht. Wenn Umgebungslicht fehlt, ist eine spezielle Lichtquelle erforderlich. Interessanterweise erfolgt die Aktualisierung des Bildschirms bei direkter Sonneneinstrahlung nicht korrekt. Auf Grundlage dieses Prinzips wurden zahlreiche Arten von Bildschirmen realisiert, auch große und vollfarbige, die allerdings noch sehr teuer sind. Gleichzeitig hat das erschwingliche Angebot kleinerer, schwarz-weißer, graustufiger oder farblich begrenzter Displays im Fachhandel stark zugelegt. Allerdings konnte ich im Internet feststellen, dass das Interesse an diesen Komponenten von seiten der „Heimwerker“ nicht so stark gestiegen ist, wie es zu vermuten wäre. Der Grund dafür liegt meines Erachtens in verschiedenen kritischen Punkten, die auch bei der Entwicklung dieses Projekts deutlich wurden; zum Beispiel übergroße Anzahl von Modellen, Versionen, Größen, Treibern und Farbkombinationen auf dem Markt, das Fehlen gut

dokumentierter und einfach zu implementierender Bibliotheken für die verschiedenen Entwicklungsplattformen und die oft lückenhafte und unzureichende Information und Unterstützung durch die Hersteller selbst. Trotzdem war es möglich, durch die Auswahl der am besten unterstützten Produkte und durch die dem Maker innewohnende Hartnäckigkeit ein interessantes Ergebnis zu erzielen. Fahren wir also mit der Analyse des Schaltplans fort, der in **Bild 2** dargestellt ist.

Das Schaltbild

Für die Steuerung des Displays wurde ein Mikrocontroller ATmega328P [2] gewählt, derselbe, der auch im Arduino-UNO-Board verwendet wird. So können wir die reichhaltige Auswahl an Bibliotheken für diese Plattform verwenden. Die MCU arbeitet dank des externen Keramikresonators mit einer Taktfrequenz von 8 MHz und befindet sich normalerweise in einem stromsparenden Zustand (`SLEEP_MODE_PWR_DOWN`). Er wacht nur auf, wenn das Display aktualisiert werden muss. Lediglich die Real-Time Clock (RTC) in Form eines DS3231M [3] bleibt immer aktiv und kann, solange das IC mit Strom versorgt wird, Datum und Uhrzeit mit einem Fehler von ± 5 ppm ($\pm 0,432$ s/d) anzeigen. Weitere bemerkenswerte Merkmale sind ein Temperatursensor, eine doppelte Alarmfunktion und ein extrem niedriger Stromverbrauch in einer Größenordnung von einigen Mikroampere (μA) in der verwendeten *Single-Supply-Konfiguration* (V_{BAT} Only). Die RTC kommuniziert mit Hilfe einer speziellen Bibliothek mit dem Mikrocontroller über eine I²C-Schnittstelle. Die Taste SW2 (beim Prototyp blau) weckt die MCU, so dass diese Daten die aktuelle Uhrzeit empfangen und auf dem

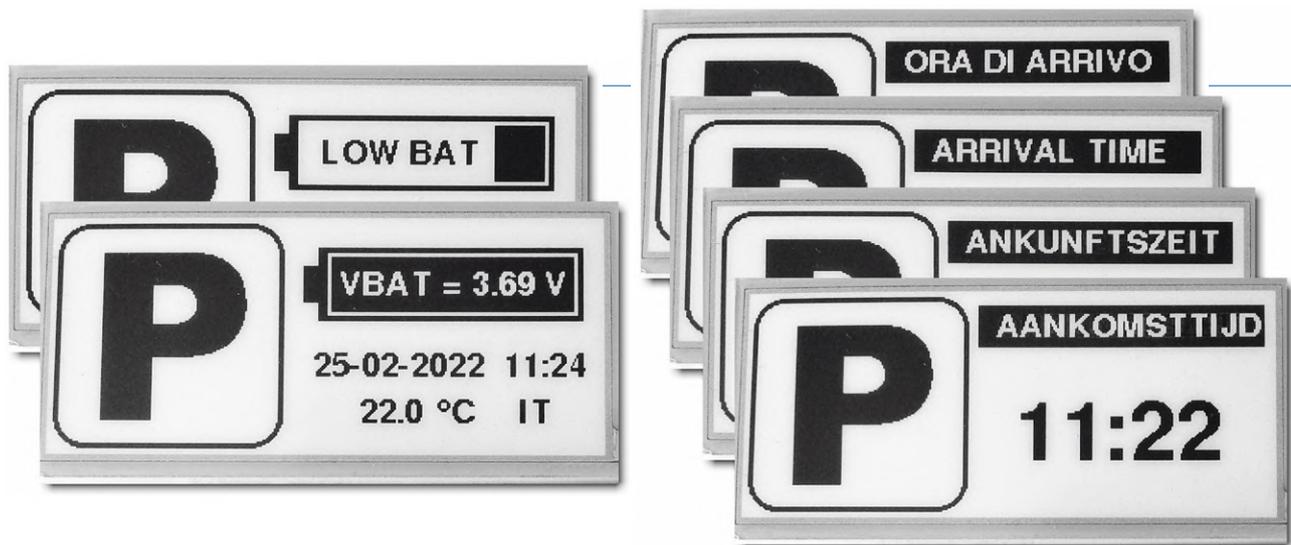


Bild 3. Beispiel für Info- und Parkbildschirm.

Display zusammen mit der Meldung „TIME OF ARRIVAL“ und dem „P“-Logo anzeigen kann, bevor sie in den Ruhezustand zurückkehrt. Ein kurzer Piepton des Summers bestätigt die Betätigung der Taste, während die blaue LED D2 für die Dauer des Ereignisses leuchtet. SW1 (beim Prototyp rot) bewirkt dagegen einen Hardware-Reset des ATmega328P, woraufhin sofort im Display das „P“-Logo, das Batteriesymbol mit der Batteriespannung, die Umgebungstemperatur, die aktuelle Uhrzeit und das Datum sowie die für die Meldung gewählte Sprache angezeigt wird (siehe Beispiel in **Bild 3**).

Derselbe Displayinhalt erscheint automatisch alle 24 Stunden (dank der RTC-Alarmfunktion zu einer gewünschten Zeit), um die vom Displayhersteller empfohlene vollständige Aktualisierung der Anzeige durchzuführen und den Batteriestatus sowie den korrekten Betrieb des gesamten Systems zu überprüfen. Daher können wir SW2 als „Parktaste“ und SW1 als „Infotaste“ bezeichnen. Die Widerstände R1 und R10 dienen zusammen mit den Kondensatoren C1 und C5 der Unterdrückung von Störungen, die durch das Prellen der Tastenkontakte entstehen, während R2, R4, R5 und R6 Pull-up-Widerstände sind. C2, C3 und C4 sind die üblichen Bypass-Kondensatoren der IC-Stromversorgungsanschlüsse. R11 begrenzt den Strom in den Reset-Anschluss des verwendeten Displays. JP2 ist der Anschluss für das Display, während JP1 das Hochladen von Firmware durch einen USBasp-Programmierer ermöglicht. Aufgrund der geringen Stromaufnahme ist ein kleiner 3,6-V-NiMH-Akku mit einer Kapazität von 40 mAh zur Energieversorgung des Prototyps völlig ausreichend. Q1, Q2, R8, R9 bilden einen Konstantstromregler mit geringem Spannungsabfall, der den Ladestrom auf etwa 6 mA begrenzt. Diesen Strom verträgt dieser Akkutyp auch bei Überschreitung der erwarteten 14...16 Stunden Ladezeit ohne Probleme. Eine Micro-USB-Buchse ermöglicht den Anschluss eines handelsüblichen 5-V-Steckernetzteils über eine selbstbrücksetzende Sicherung, während die Diode D1 den Rückfluss der Energie aus dem Akku verhindert. Die LED D3 zeigt an, dass das Gerät gerade geladen wird. Auf das Thema Stromversorgung werden wir aber später noch genauer eingehen.

Praktische Verwirklichung

Der Prototyp der Parkscheibe wurde wie üblich auf einer Prototyp-Platine erstellt, wie in **Bild 4** zu sehen ist. Natürlich ist dies nur eine von vielen Möglichkeiten; durch die Verwendung von SMD-Bauteilen und einer gerouteten Platine könnte man sicherlich eine kleinere Baugröße erreichen, nur ein wenig größer als das Display selbst. Die Batterie und der Summer befinden sich direkt unter dem E-Paper-Modul. Für den achtpoligen Anschluss

des Moduls habe ich eine Kombination aus Stift- und Buchsenleiste gebastelt. Die ICs auf der Prototyp-Platine sind mit Fassungen ausgestattet, während für die RTC DS3231M ein leicht erhältliches Breakout-Board in SO-8-Abmessungen verwendet wird. Bei der Wahl des Gehäuses genießt man maximale Freiheit, solange es transparent ist oder mit einem transparenten Fenster ausgestattet ist. Was das E-Paper-Modul betrifft, so ist es manchmal schwierig, aus der ganzen Typenvielfalt an scheinbar ähnlichen Modellen das passende herauszufinden. Das von mir verwendete schwarz-weiße in **Bild 5** stammt von Waveshare und besitzt eine 2,9-Zoll-Diagonale und eine Auflösung von 296 x 128 [4]. Das Modul verfügt über einen eingebauten Logikpegelwandler und unterstützt partielle Auffrischung, eine unverzichtbare Funktion in diesem Projekt. Andere Modelle garantieren möglicherweise nicht die gleichen Ergebnisse, die ich erzielt habe, oder erfordern Änderungen an der Firmware oder im Schaltplan. Zu beachten ist auch der zulässige Betriebstemperaturbereich, der in diesem Fall von 0...50°C reicht (ein Temperaturfenster, das im Auto oftmals zu eng ist).

Bild 6 zeigt die Lötseite des Prototyps, auf der sich auch die Diode D1 (SMD-Typ) und der Micro-USB-Ladeanschluss befinden.

Um auf das Thema Stromversorgung zurückzukommen: Die Wahl

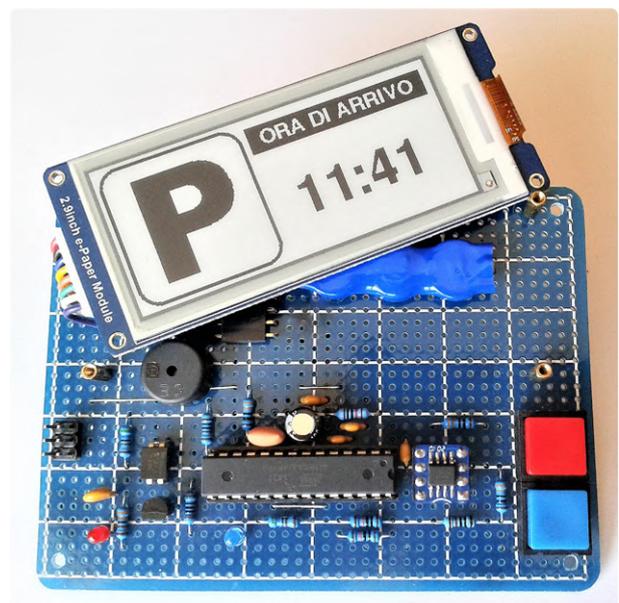


Bild 4. Ansicht des Prototyps von oben.



Bild 5. Das im Projekt verwendete e-Paper-Modul mit seinem „maßgeschneiderten“ Anschluss. (Quelle: Waveshare)

der nicht mehr ganz neuen Technologie eines NiMH-Akkus ist vor allem der höheren Sicherheit und Zuverlässigkeit unter den nicht immer optimalen Einsatzbedingungen in einem Fahrzeug geschuldet. Im Vergleich zu LiIon-Batterien beispielsweise besteht keine Explosionsgefahr, und auch die Leistung bei niedrigen Temperaturen ist besser. Der Ladevorgang ist zwar etwas langsamer, aber einfacher zu handhaben. In Anbetracht der minimalen Ströme und der Betriebsspannung der Schaltung, die leicht zwischen 3,3 V und 5,0 V variieren kann, sollte auch die Versorgung mit drei in Reihe geschalteten AAA-Alkalibatterien in Betracht gezogen werden, die die Ladeschaltung überflüssig macht und dennoch eine Laufzeit bietet, die die des Autos übersteigen dürfte! Die genaue Quantifizierung des „Verbrauchs“ der immer beliebter werdenden batteriebetriebenen Geräte, die sich den Großteil ihres Daseins im Energiesparmodus befinden (auch Schlafmodus, Power-down, Ruhezustand, Stromsparmodes genannt), ist nicht einfach. Es reicht nicht aus, wie früher ein Multimeter in Reihe mit dem Netzteil zu schalten und den Strom abzulesen. Die grundlegende Berechnung, die durchgeführt werden muss, um den Durchschnittswert des entnommenen Stroms und damit eine Abschätzung der Batterielebensdauer zu erhalten, kann mit der folgenden Pseudoformel dargestellt werden:

$$I_{AVG} = \frac{[(I_{ON} \cdot T_{ON}) + (I_{SBY} \cdot T_{SBY})]}{(T_{ON} + T_{SBY})}$$

wobei I_{AVG} den mittlere Strom, I_{ON} den Strom bei Aktivität, I_{SBY} den Stand-by-Strom, T_{ON} die Aktivitätszeit und T_{SBY} die Stand-by-Zeit bezeichnet.

Ich habe die Stromaufnahme während der Aktivität in **Bild 7** mit dem Oszilloskop anhand eines Spannungsabfalls an einem niederohmigen Präzisions-Shunt-Widerstand „eingefangen“ und

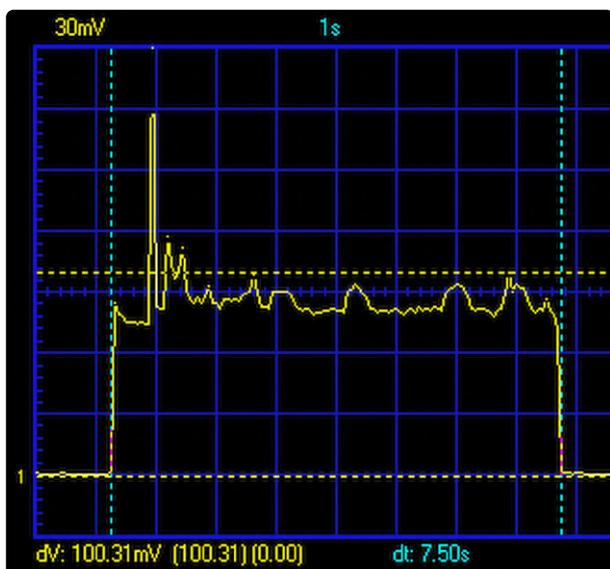


Bild 7. Verlauf der Stromaufnahme der Schaltung.

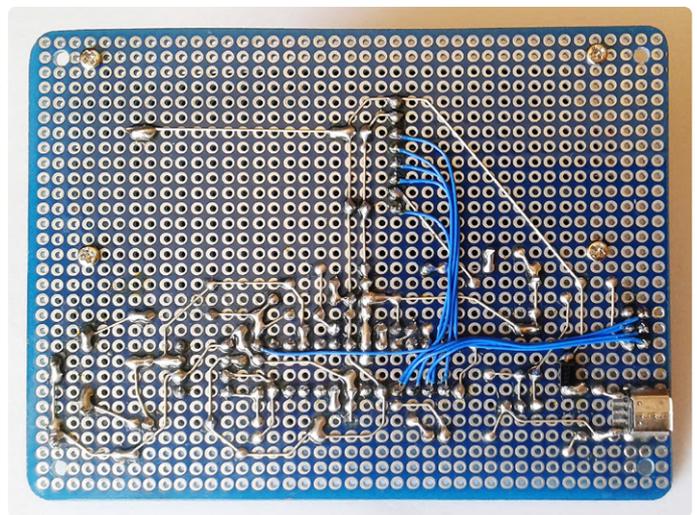


Bild 6. Lötseite des Prototyps.

mit akzeptabler Näherung einen Wert von 10 mA für eine Zeit von 7,5 s erhalten, was der Dauer eines Anzeigezyklus entspricht. Der Ruhestrom, mit einem guten Digitalmultimeter gemessen, beträgt dagegen nur 3 μ A. Geht man davon aus, dass die Parkscheibe viermal am Tag benutzt wird, zuzüglich des Auffrischungszyklus, ergibt sich eine Gesamtbetriebszeit von $7,5 \times 5 = 37,5$ s und eine Ruhezeit von $(24 \times 3600) - 37,5 = 86.362,5$ s über 24 Stunden. Wir erhalten also:

$$I_{AVG} \approx \frac{[(10 \text{ mA} \cdot 3,75 \text{ s}) + (0,003 \text{ mA} \cdot 86362,5 \text{ s})]}{(37,5 \text{ s} + 86362,5 \text{ s})}$$

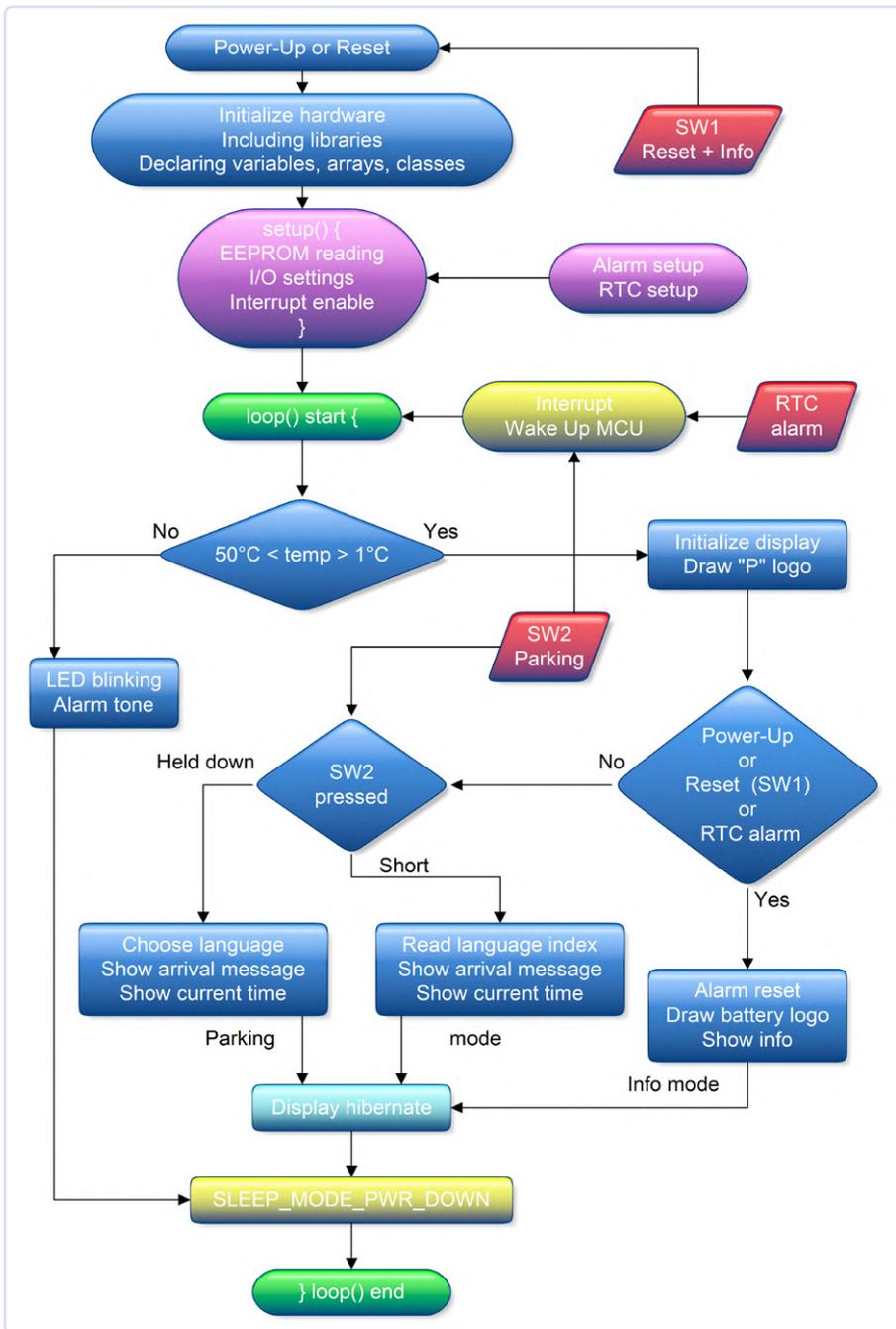
$$\approx 0,008 \text{ mA}$$

Eine 40-mAh-Batterie - wenn man der Einfachheit halber die Verringerung der Kapazität durch Selbstentladung, Absinken der Nennspannung, Temperaturschwankungen und andere Faktoren außer Acht lässt - könnte bei einem durchschnittlichen Strom von 0,008 mA den Stromkreis im Idealfall etwa $40/0,008 = 5.000$ Stunden lang versorgen (das heißt, mehr als 200 Tage oder mindestens sechs Monate)! Realistisch gesehen wäre sogar eine auf die Hälfte reduzierte Dauer ein gutes Ergebnis. Für diese Art der Berechnung gibt es inzwischen auch bequeme Online-Anwendungen. Ich gebe einen Link zu einer Anwendung an, die ich für eine der effektivsten und praktischsten halte [5].

Die Firmware und ihre Funktionsweise

In diesem Projekt wurde der Quellcode [9] in der Arduino-IDE 1.8.19 geschrieben, die mit dem Arduino-Kernel *MiniCore v2.1.3* und einigen spezifischen Bibliotheken ausgestattet wurde. Der verwendete Kernel ermöglicht eine effizientere und vielseitigere Verwaltung des ATmega328P-Mikrocontrollers und optimiert vor allem die Speichernutzung des kompilierten Codes, der immerhin 31.264 Bytes der insgesamt 32.768 Bytes des Programmspeichers (Flash) und 1.501 Bytes der 2.048 Bytes des dynamischen Speichers (SRAM) belegt, als fast komplett die Möglichkeiten dieser MCU ausnutzt. Es ist anzumerken, dass dieses Projekt nicht mit einem Arduino-Uno-Board durchführbar ist, auch nicht im Entwicklungsstadium, da ein Teil des Speichers des Arduino Uno vom Bootloader verwendet wird. Wir müssen einen Barebone-Mikrocontroller mit einem externen USBasp-Programmierer programmieren. Apropos Speicher: Das Schlüsselwort `PROGMEM` taucht mehrmals im Listing

Bild 8. Flussdiagramm der Firmware.



Modelle wie möglich auszuweiten. Ich habe daher versucht, die Bibliothek zu straffen und nur die Funktionen und Definitionen zu übernehmen, die für das im Projekt verwendete Display notwendig sind. Diese Zusammenfassung ist in der Datei *Waveshare_29_BW_avr.h* zu finden, während die Datei *ParkBitmap128x128.h* nur das mit Hilfe eines speziellen Konverters [6] gewonnene Array von Bytes für die Bitmap des Parklogos darstellt (das große P in einem Quadrat mit abgerundeten Ecken, Größe 128 x128 Pixel, schwarz/weiß). Diese Dateien werden von [9] heruntergeladen und in dem im Sketch-Ordner untergebracht, zusammen mit dem Hauptsketch *Disco_Orario_e-Paper.ino*, in dem sich auch Links zu den Core- und Bibliotheks-Sites, ausführliche Kommentare zum Code und andere nützliche Hinweise befinden. Wenn Sie an Details des Listings interessiert sind, sollten Sie den Sketch in einem Editor Ihrer Wahl öffnen. Hier möchte ich nur die Funktionsweise des Programms mit Hilfe eines Flussdiagramms in **Bild 8** veranschaulichen.

Mit der Tatsache im Hinterkopf, dass die Schaltung immer an der Batterie angeschlossen ist, werden beim ersten Einschalten (Power-Up) die Initialisierungsprozedur und die `setup()`-Funktion ausgeführt, wobei ein Interrupt aktiviert wird. Dann springt das Programm in die `loop()`. Zu Beginn der Schleife wird die Umgebungstemperatur geprüft. Liegt sie nicht im erwarteten Bereich, wird ein akustisches und optisches Alarmsignal ausgegeben, dann wird der Mikrocontroller in den maximalen Energiesparmodus (`SLEEP_MODE_PWR_DOWN`) versetzt. Andernfalls wird die `loop()`

auf und bezieht sich auf die Byte-Arrays der Bitmap und der Textzeichen-Strings, die schreibgeschützte Daten sind. Indem diese Arrays als `PROGMEM` deklariert werden, können Funktionen auf diese Daten zugreifen, sie direkt aus dem Flash-Speicher lesen, ohne sie vorher in das viel kleinere SRAM kopieren zu müssen. So bleibt das SRAM für die „dynamische“ Ausführung des Programms verfügbar. Für die Kommunikation mit der integrierten Echtzeituhr (RTC) wird die Bibliothek *DS3231M 1.0.6* verwendet, während die Bibliothek *GxEPD2 1.3.6*, die von der Grafikkbibliothek *GFX_Root 2.0.0* unterstützt wird, für die grundlegende Verwaltung des E-Paper-Displays gewählt wurde.

Die Grafikkbibliothek muss mit der Bibliothek dieses Projekts überschrieben werden, da diese geänderte Schriftarten enthält. Die *GxEPD2*-Bibliothek ist ein gewaltiges Stück Arbeit, aber leider fehlt eine gut strukturierte Dokumentation. Im Code sind zwar zahlreiche Beispiele zu finden, aber diese schrecken aufgrund ihrer scheinbaren Komplexität auf den ersten Blick ziemlich ab. Auf den zweiten Blick verwandelt sich die Komplexität in den Versuch, die Kompatibilität der Bibliothek auf so viele Display-

fortgesetzt: Das Display wird initialisiert und das „P“-Logo, der Batteriestatus, das aktuelle Datum, die Uhrzeit, die Temperatur und die Meldung in der gewählten Sprache angezeigt. Dieser *Info*-Modus wird verlassen und das Display in den Ruhezustand (*hibernate*) versetzt, was wiederum zur Energieeinsparung beiträgt, die MCU ausgeschaltet und die `loop()` abgebrochen. Aus diesem Zustand kann der Mikrocontroller durch einen Hardware-Reset (mit dem Taster SW1) oder durch einen Interrupt (die tägliche Alarmfunktion der RTC oder den Taster SW2) aufgeweckt werden (*Wake Up*). Wenn die Wiederaufnahme des Programms durch den Reset oder den Alarm ausgelöst wird, beginnt die `loop()` wieder von vorne und endet immer im *Info*-Modus. Wird der Mikrocontroller hingegen über SW2 reaktiviert, startet die `loop()` neu, initialisiert das Display, zeigt das Logo und, wenn die Taste gedrückt und sofort wieder losgelassen wurde, die Ankunfts meldung und die aktuelle Uhrzeit, was wir als *Parking*-Modus bezeichnen. Wenn Sie dagegen SW2 gedrückt halten, werden die Meldungen in den vier Sprachen nacheinander angezeigt. Lassen Sie einfach die Taste los, wenn Ihre gewünschte Sprache erscheint, dann wird die Einstellung bis zur



nächsten Änderung im EEPROM gespeichert. Danach erscheint die aktuelle Uhrzeit, und der Zyklus endet immer im *Parking*-Modus. Ein Beispiel für die Bildschirmdarstellung ist in Bild 3 zu sehen. Wir haben gesehen, wie das aktuelle Datum und die Uhrzeit von der RTC DS3231M bereitgestellt und beibehalten werden, aber die RTC muss beim allerersten Anlegen der Versorgungsspannung (und nach einem Ausfall der Versorgungsspannung) noch eingestellt werden. Um die Firmware und die Schaltung möglichst einfach zu halten, sind Datum und Uhrzeit in einer Zeile in der `setup()`-Routine „hardcoded“ vorgegeben, so dass sie gleichzeitig mit dem Laden des Sketches eingefügt wird:

```
DS3231M.adjust(DateTime(2022, 03, 02, 19, 10, 00));
```

Wie sie leicht erkennen können, ist das Format „JJJJ,MM,TT, hh, mm,ss“. Wenn Sie die richtigen Daten eingegeben haben, können Sie den Sketch laden, das Datum und die Uhrzeit überprüfen und dann die Zeile auskommentieren (fügen Sie den doppelten Schrägstrich // am Anfang hinzu). Wenn Sie den Sketch erneut laden, wird so vermieden, dass die Uhr bei jedem Zurücksetzen auf ihre ursprünglichen Einstellungen zurückfällt. Um eine ausreichende Genauigkeit zu erreichen, messen Sie einfach die Zeit, die für das Hochladen des Codes benötigt wird, zum Beispiel 30 s, und starten Sie dann das erste Hochladen 30 s im Voraus. Mit ein paar Versuchen kann eine sekundengenaue Synchronisation erreicht werden! Die tägliche Aktualisierungszeit wird ebenfalls im Code eingestellt, indem man einfach die gewünschte Zeit eingibt.

Zum Schluss noch ein Hinweis auf die Methode zur Messung der Batteriespannung. Man findet sie oft im Internet, mit einigen Variationen, und Microchip selbst dokumentiert sie in einer eigenen Notiz [7]. Der „Trick“ besteht darin, über die entsprechenden Register die interne Referenzspannung (1,1 V) als Eingangswert des ADCs und die zu messende Spannung (V_{CC}) als Referenz einzustellen. Jede Änderung von V_{CC} verändert den ADC-Messwert, so dass der Spannungswert mit ausreichender Genauigkeit berechnet werden kann.

Abschließende Überlegungen

Obwohl die E-Paper-Technologie zweifellos interessante Merkmale aufweist, die sie für das vorgestellte Projekt besonders geeignet machen, hat sie auch gewisse Einschränkungen, von denen die

Knöllchen droht!

Die Schaltung demonstriert zwar anschaulich die Ansteuerung und Verwendung eines E-Paper-Displays, dürfte Sie aber im Falle eines Falles, nämlich dem Einsatz als Parkscheibe kaum vor einem Knöllchen schützen. In Deutschland muss eine StVO-konforme Parkscheibe nämlich zwingend das bekannte blaue Design in den Maßen 110x150 mm aufweisen [9], da es sich um das offizielle Verkehrszeichen 318 handelt. Schlimmer noch: Ist für die kontrollierende Ordnungskraft nicht eindeutig erkennbar, dass ein eventuelles „Mitlaufen“ der Uhrzeit deaktiviert ist, kann man sich nach §§ 22, 23 StGB auch eine Anzeige wegen Betrugs einhandeln.



Bildquelle: <https://tinyurl.com/3y93x4ab>, gemeinfrei

You CAN get it...

Hardware und Software
für CAN-Bus-Anwendungen...



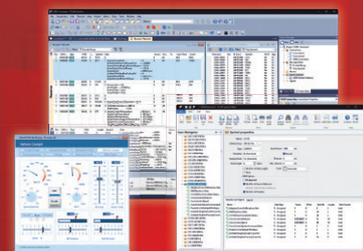
PCAN-USB FD

CAN-FD-Interface für den USB-Port inkl. Software, APIs und Treiber für Windows und Linux.



PCAN-MicroMod FD

Konfigurierbares Einsteckmodul mit CAN-FD-Interface und I/O-Funktionalität. Erhältlich mit Evaluation-Board oder einsatzbereiten Grundplatten.



PCAN-Explorer 6

Professionelle Windows-Software zur Steuerung und Überwachung von CAN-FD- und CAN-Bussen.

Irrtümer und technische Änderungen vorbehalten.

www.peak-system.com

PEAK
System

Otto-Röhm-Str. 69
64293 Darmstadt
Germany
Tel.: +49 6151 8173-20
Fax: +49 6151 8173-29
info@peak-system.com



offensichtlichste die niedrige Aktualisierungsrate ist. Das hier verwendete Modell führt den kompletten Auffrischungszyklus in zwei Sekunden und die teilweise Auffrischung in 0,3 Sekunden durch. Diese Werte können daher bei der Anzeige sich schnell ändernder Bilder, Grafiken und Texten nicht mit anderen Displaytypen mithalten. Dennoch hoffe ich, dass dieser Artikel Ihre Lust und Neugierde geweckt hat, mit E-Paper-Displays zu experimentieren und über den tatsächlichen Nutzen der vorgeschlagenen Schaltung hinaus einige interessante Ideen gefunden haben, die Sie mit der Einstellung eines echten Makers weiterentwickeln können. ◀

RG – 230012-02

Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Dann wenden Sie sich bitte an den Autor unter a.dellapia@elettronicaemaker.it oder an die Elektor-Redaktion unter redaktion@elektor.de.



Über den Autor

Antonello Della Pia interessiert sich schon seit seiner Kindheit für Elektrizität und elektronische Geräte. Er verfügt über ein Diplom als Elektrotechniker. Antonello hat seine Leidenschaft für analoge und digitale Elektronik stets gepflegt und weiterentwickelt. Zurzeit beschäftigt er sich mit Mikrocontrollern und Programmierung und versucht, seine Computerkenntnisse zu verbessern. Antonello liebt es, Projekte zu entwickeln und vorzuschlagen, die so originell wie möglich und - wie er hofft - auch interessant sind.



Related Products

- **Waveshare 2,9" 3-farbiges E-Ink/E-Paper-Displaymodul**
SKU 18776: www.elektor.de/18776
- **Ynvisible Segment E-Paper-Display-Bundle**
SKU 20143: www.elektor.de/20143



Stückliste

Widerstände:

(alle 0,25 W, 1%)
R1, R10, R12 = 330 Ω
R2, R4, R5, R6 = 10 k
R3, R8 = 5k6
R7 = 3k3
R9 = 100 Ω
R11 = 1 k

Kondensatoren:

C1, C3, C5 = 100 n, 50 V Polyester- oder Keramik-Vielschichtkondensator
C2 = 100 μ, 10 V Elko
C4 = 470 n, 50 V Polyester- oder Keramik-Vielschichtkondensator

Halbleiter:

Q1 = IRLD024, logic level gate drive MOSFET
Q2 = BC546B
D1 = SS34, Low-Drop-Schottky-Diode
D2 = Blaue LED 3 mm, Hi-Bright
D3 = Rote LED 3 mm, Hi-Bright
U1 = ATmega328P-PU
U2 = RTC DS3231M, SO-8-Gehäuse

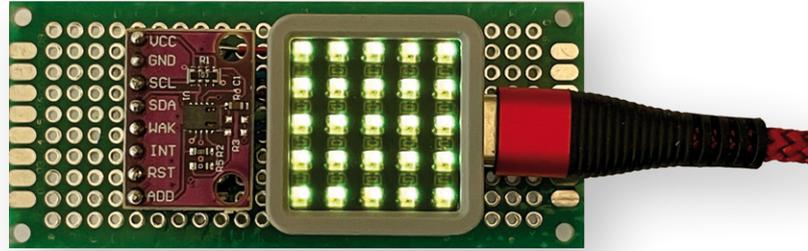
Sonstiges:

Y1 = 8-MHz-Keramik-Resonator
BUZZER = Piezosummer
SW1, SW2 = Drucktaster 1xan
F1 = Rückstellende Sicherung (Polyfuse) 250 mA
X1 = Micro-USB-Buchse Typ B
JP1 = 6-poliger ICSP-Platinenverbinder, male
JP2 = 8-poliger Platinenverbinder, male/female
DISPLAY = e-Paper-Display 2,9 Zoll S/W, (siehe Text)
BAT1 = NiMH-Batterie 3,6 V, 40 mAh (3 x V40H)
2x14-polige DIP-Fassung
2x4-polige DIP-Fassung
SMD - DIP8 für SO-8-Adapter
Prototyping-Platine

WEBLINKS

- [1] Elektronisches Papier (Wikipedia): https://de.wikipedia.org/wiki/Elektronisches_Papier
- [2] Datenblatt ATmega328P: <https://www.elektormagazine.com/atmega328p-datasheet>
- [3] Datenblatt DS3231M: <https://datasheets.maximintegrated.com/en/ds/DS3231M.pdf>
- [4] E-Ink Display Modul 296x128, 2,9 Zoll (Waveshare): <https://www.elektormagazine.com/waveshareeink>
- [5] Batteriebensdauer-Online-rechner: <https://www.omnicalculator.com/other/battery-life>
- [6] BitmapToByteArrayConverter: <https://www.briandorey.com/post/bitmap-byte-converter-for-e-ink-display>
- [7] Messung der VCC/Batteriespannung ohne I/O-Pin (Microchip): <https://www.elektormagazine.com/vcc/battery>
- [8] Grundlegende Funktion des E-Paper-Displays: <https://www.elektormagazine.com/bwcapsules>
- [9] Quellcode: <https://www.elektormagazine.de/230012-02>
- [10] Auf die richtige Scheibe kommt es an!: <https://www.stern.de/politik/flensburg--parkscheiben-posse-sorgt-fuer-kopfschuetteln-33209910.html>

eCO₂-Telegram-Bot



Luftqualitätsmessung mit Telegram-Anbindung

Von Peter Neufeld (Deutschland)

Nicht immer sind hochgenaue und teure Messgeräte zur Bestimmung der Raumluftqualität notwendig. Eine Groablesung und Grenzwertüberwachung kann auch sehr hilfreich sein, wenn man einfach nur wissen möchte, wann es angebracht ist, einen intensiv genutzten Raum zu lüften. Noch nützlicher wird ein solches Gerät, wenn seine einfache lokale Anzeige um eine Alarmausgabe an Telegram-Accounts auf Smartphones ergänzt wird.

Eine hochintegrierte Sensorik, ein moderner Mikrocontroller und ein wenig BASIC-Programmierung machen es bei sehr geringem technischem Aufwand möglich, mit diesem eCO₂-Monitor und seinen verschiedenen Anzeigemodi auf vier Arten auf schlechte Luftqualität hinzuweisen:

- lokale NeoPixel-LED(s) als Luftqualitätsampel
- Webinterface für lokale Geräte mit einem Webbrowser
- manuelle Abfrage per Telegram Messenger
- Telegram-Warnmeldung, die an einen dedizierten Telegram-Benutzer gesendet wird

Die Hardware

Die einfache Schaltung in **Bild 1** basiert auf einem eCO₂-Sensor CCS811 und einem ESP32-SoC-Modul. Die Luftqualität wird durch eine oder mehrere NeoPixel-LEDs dargestellt. Ein einzelner Tastschalter ist mit einem Eingangspin verbunden. Der Sensor und der SoC kommunizieren über einen zweiadrigen I²C-Bus. Die NeoPixel-Anzeige verwendet nur eine Datenleitung am GPIO27, ob nun ein einzelner LED-Punkt oder eine Matrix angesprochen wird.

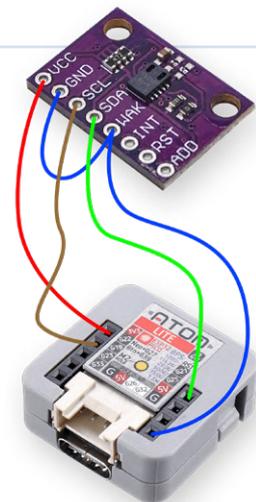
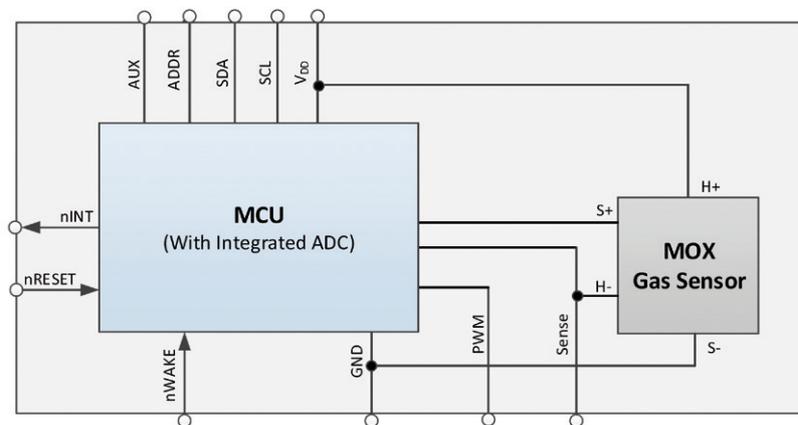


Bild 1. eCO₂-Telegram-Bot mit M5Stack ATOM-Matrix und CCS811 samt Innenschaltung. (Bildquelle: AMS-Datenblatt)

Obwohl fast jedes ESP32-Modul mit exponierten I²C-Pins für diese Arbeit geeignet ist, empfehle ich einen M5Stack „ATOM matrix“ oder „ATOM lite“. Der Grund dafür ist, dass diese praktischen Geräte einen ESP32-Pico-D4 mit Antenne, eine NeoPixel-Matrix oder einen einzelnen NeoPixel-Punkt, einen Taster und einiges mehr in einem kleinen und formschönen Schutzgehäuse vereinen. So sind der Sensor und das ESP32-Modul fast alles, was man an Hardware braucht, um das Gerät auf einer gewöhnlichen Lochrasterplatine aufzubauen. Dazu steckt man den CCS811 und den ATOM auf entsprechende Buchsen- beziehungsweise Stifteleisten und legt die wenigen notwendigen Verbindungen mit Schweißdraht, wie in **Bild 2** gezeigt. Die Unterseite der durchkontaktierten Rasterplatine wird mit Klebeband isoliert. Die Schaltung wird über das ESP32-Modul von einem USB-Netzteil (5 V bei maximal 500 mA) versorgt, wahlweise über den USB-C-, den HY2.0-Grove-Anschluss oder die unteren Buchsenleisten.

Der eCO₂-Sensor CCS811

Der CCS811-Sensor kann den CO₂-Gehalt nicht direkt messen! Er berechnet den äquivalenten CO₂-Gehalt (eCO₂) durch Messung der tVOC (Total Volatile Organic Compound), wobei die Hauptquelle dieser flüchtigen organischen Verbindungen die vom Menschen ausgeatmete Luft ist.

Dieser preiswerte Sensortyp namens *Metal Oxide Semiconductor (MOS)* sucht eine relative Basislinie für „gute Luft“, indem er die beste Luftbedingung über einen längeren Zeitraum bestimmt und dann annimmt, dass sich der Sensor zu diesem Zeitpunkt in frischer, unverschmutzter Luft mit 400 ppm CO₂ befindet. Aber: Der Sensor speichert diesen Wert nicht von alleine.

Außerdem kann sich die Sensorempfindlichkeit im Laufe der Zeit und unter verschiedenen Umgebungsbedingungen wie Temperatur und Luftfeuchtigkeit ändern. Um also zuverlässige eCO₂-Messwerte zu liefern, benötigt der Sensor:

- › eine einmalige Burn-In-Zeit von mehr als 48 Stunden
- › eine Mindest-Einlaufphase von circa 20 Minuten nach jedem Kaltstart.

Das Datenblatt [1] enthält nähere Informationen zum Gassensor.

Die Software

Das Programm wurde mit ANNEX32 entwickelt, einem BASIC-Interpreter für ESP32 [2]. Nachdem man über ein Installationsprogramm und eine serielle USB-Schnittstelle den Interpreter in das Flash-RAM des ESP32-Moduls gebrannt hat, laufen Interpreter samt seiner Entwicklungsumgebung vollständig im ESP32. Es wird nur ein Chrome- oder Firefox-Browser benötigt, um das BASIC-Skript zu laden, zu bearbeiten, zu testen und (automatisch) auszuführen. Die mindestens erforderliche Annex32-Version ist V1.435, da diese CCS811- und Telegram-Messenger-Unterstützung enthält. Die Online-Hilfe [3] für Annex32 ist eine sehr nützliche Einführung in diesen BASIC-Interpreter. Die Hauptaufgaben des BASIC-Codes sind:

- › Der CCS811 wird initialisiert und einmal pro Sekunde nach dem eCO₂-Wert gefragt

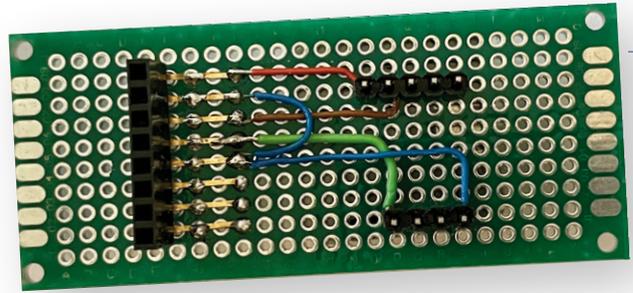


Bild 2. Einfache Verdrahtung der beiden Module auf einer Lochrasterplatine.

- › Der eCO₂-Status der Umgebungsluft wird als **GRÜN**, **GELB** oder **ROT** eingestuft
- › Die eingebaute NeoPixel-Matrix oder die einzelne NeoPixel-LED zeigt diese Kategorie lokal durch ihre Farbe an
- › Ein Webinterface zeigt den eCO₂-Wert und die Kategorie über einen Browser im (W)LAN an
- › Die Kategorie und der eCO₂-Wert können manuell per Telegram-Messenger abgefragt werden, da das Programm mit einem Telegram-Bot arbeitet und eingehende Benutzerbefehle vom Telegram-Server abholt
- › Eine Telegram-Warmmeldung wird automatisch an die letzte Telegram-Chat_id gesendet, wenn sich die Luftqualität im roten Bereich befindet (**Bild 3**)
- › Die Basislinie der Luftqualität kann manuell durch Drücken der Fronttaste des ATOM-Moduls oder durch einen eingehenden Telegrammbefehl gespeichert werden

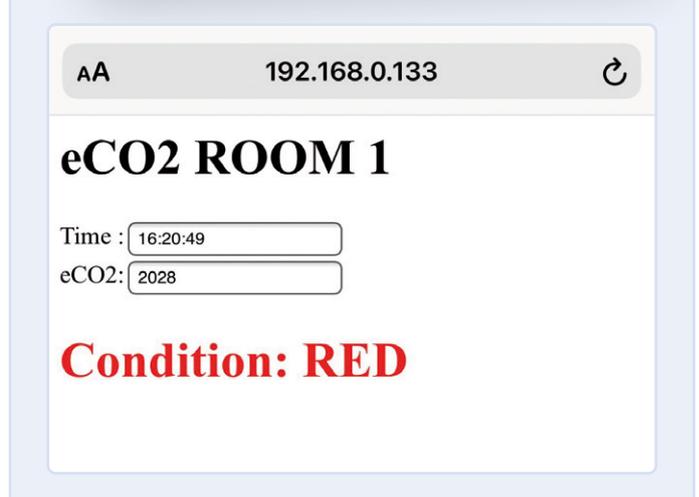
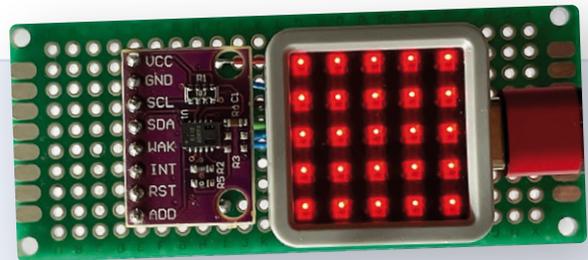


Bild 3. Warmmeldung: Die Luft ist so richtig mies!



Programmauszug: Ausgabe der Webseite

```
'#####
MAKE_WEBPAGE:
'-----
'create the autorefreshing html page with dynamic
'display values and matching colors
A$ = ""
A$ = A$ + "<H1>eCO2 " + LOCATION$
A$ = A$ + "</H1>"
A$ = A$ + "Time :" + textbox$(T$) + "<br>"
A$ = A$ + "eCO2: " + textbox$(STR_eCO2$) + "<br>"
A$ = A$ + |<span style="color:| + COND_COL$ + |">|
A$ = A$ + "<H1>Condition: "+ CONDITION$+ "</H1>"
A$ = A$ + "</span>"
cls
autorefresh 1000
html A$
return
'#####
```

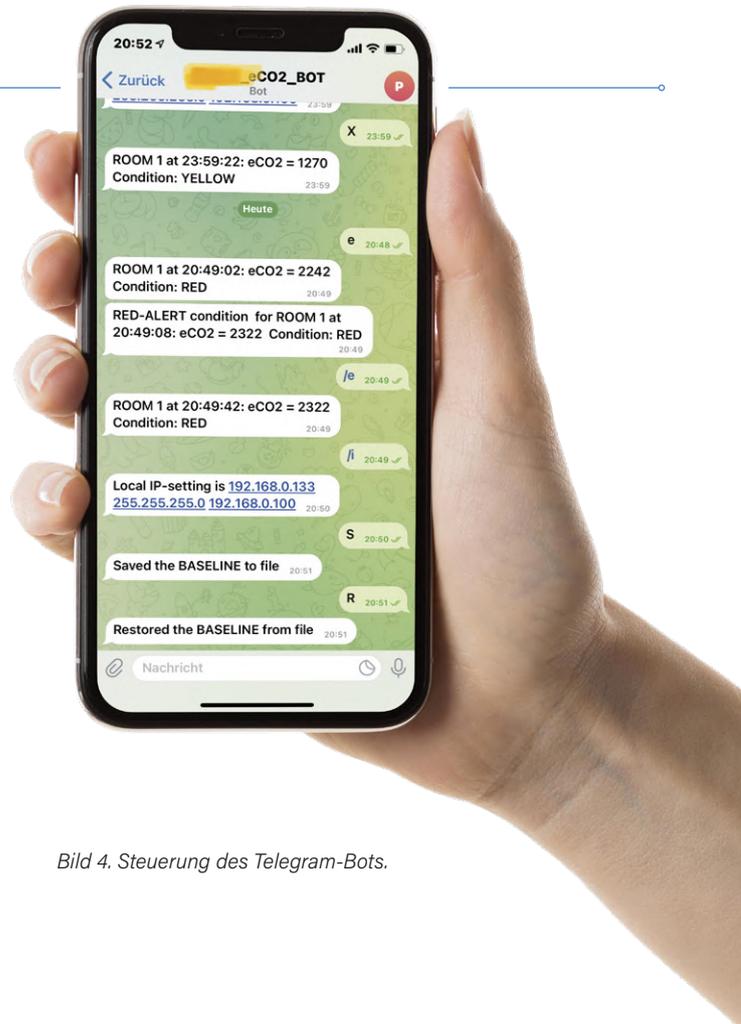


Bild 4. Steuerung des Telegram-Bots.

Die Telegram-Bot-Routine fragt die Telegram-Server regelmäßig nach eingehenden Befehlen ab. Sie reagiert auf diese Befehle, wie das **Bild 4** beweist.

- > /e gibt den eCO₂-Wert und die Kategorie zurück [GRÜN|GELB|ROT]
- > /s speichert den Basiswert in /baseline.txt
- > /r stellt den Basiswert aus /baseline.txt wieder her
- > /i gibt die lokalen IP-Einstellungen des Moduls zurück
- > [jedes andere Zeichen] bewirkt dasselbe wie /e

Ihr eigener Telegram-Token

Um die Telegram-Funktionen im BASIC-Programm nutzen zu können, müssen Sie zunächst Ihren eigenen Telegram-Bot erstellen, indem Sie den Anweisungen von BotFather unter [4] in Ihrer Telegram-App folgen. Dadurch erhalten Sie Ihr persönliches Telegram-Token und einen Bot-Namen. Wichtig: Diese beiden Angaben müssen Sie in das BASIC-Programm einfügen, um die entsprechenden Variablen zu setzen.

Die Verwendung von Annex32-BASIC, dessen aktuelle Version Sie stets im Annex-RDS-Forum [5] abholen können, hat sicherlich zumindest den Vorteil, auch für unerfahrene Programmierer gut lesbar und auch an die eigenen Bedürfnisse anpassbar zu sein. Zum besseren Verständnis der Funktionen ist der Code [6] dennoch reichlich mit Kommentaren versehen. Dieses Skript wird dann in den webbasierten Editor von Annex32 eingefügt und im ESP32-Modul als Autorun-Datei (/default.bas) gespeichert. ◀

RG – 210566-02

Sie haben Fragen oder Kommentare?

Haben Sie technische Fragen oder Anmerkungen zu diesem Artikel? Dann wenden Sie sich bitte per E-Mail an den Autor unter peter.neufeld@gmx.de oder die Elektor-Redaktion unter redaktion@elektor.de.



Passende Produkte

- > **M5Stack AtomU ESP32 Development Kit**
SKU 20184: <https://elektor.de/20184>
- > **ESP32-PICO-Kit V4**
Mit Buchsenleisten: www.elektor.de/18423
Ohne Buchsenleisten: www.elektor.de/20323
- > **Dogan and Ahmet Ibrahim:**
Das offizielle ESP32-Handbuch
Buch, kartoniert, SKU 18397: www.elektor.de/18397
E-Book, PDF, SKU 18399: www.elektor.de/18399

WEBLINKS

- [1] Datenblatt CCS811: <https://bit.ly/2qQKqKu>
- [2] Annex32: <https://bit.ly/3Gtf8RS>
- [3] Annex32-Hilfe: <https://bit.ly/3Gwhdgd9>
- [4] Telegram Bot-Father: <https://t.me/BotFather>
- [5] Aktuelle Annex32-Version und Forum: <https://bit.ly/3VBTRK2>
- [6] Projekt auf Elektor Labs: <https://www.elektormagazine.com/labs/eco2-telegram>



Hinter den Kulissen von High-End-Audio im Eigenbau

Elektor-Ingenieur Ton Giesberts im Interview über die hohe Kunst des Analogdesigns

Von Jan Buiting (Elektor)

Lernen Sie den Mann kennen, der hinter der Fortissimo-100-Endstufe und vielen anderen hochkarätigen Audioprojekten in Elektor steckt: Elektor-Ingenieur Ton Giesberts.

Jan: Erzähl uns kurz von deinem technischen Hintergrund, deiner Erfahrung und wie und wann du zu Elektor gekommen bist.

Ton: Nach meinem Studium wurde ich fast sofort von einer Stellenvermittlung angesprochen, ob ich nicht zu einem Gespräch vorbeikommen möchte. Sie boten mir mehrere Stellen an, darunter auch eine an einer Schule. Das gefiel mir nicht, aber eine Stelle bei einem Verlag für eine technische Monatszeitschrift im südniederländischen Beek gefiel mir. Als Abonnent wusste ich, dass dies nur Elektuur [so der frühere niederländische Name von Elektor] sein konnte. Elektuur suchte gerade jemanden, der sich mit analoger Technik beschäftigen wollte, und das passte gut zu mir.

Jan: Wie haben sich das Handwerk „Elektronik“ und seine Praxis im Laufe der Jahre verändert?

Ton: Mit der Zeit hat die Verwendung von verdrahteten Bauteilen abgenommen. Ich denke, dass das Aufkommen und die standardmäßige Verwendung von SMDs Selbstbauprojekte immer unattraktiver machten. Die Elektronik hat sich in rasantem Tempo auf die Softwareentwicklung verlagert. Mikroprozessorplatinen sind seit langem beliebt, aber sie selbst zu bauen

ist weniger attraktiv als fertige Module zu kaufen, besonders in den letzten Jahren. Die Software bestimmt dann die Anwendung, obwohl natürlich auch einige externe (analoge) Komponenten wie Sensoren und so weiter zum Einsatz kommen.

Jan: Wie ist deine Arbeit heutzutage organisiert, und mit welchen Laborgeräten und Werkzeugen arbeitest du normalerweise? Benutzt du ein Simulationsprogramm für deine Entwürfe?

Ton: Jetzt, wo ich für eine andere Elektor-Abteilung arbeite (Bücher, Bausätze und Säulenprodukte), gibt es andere Aufgaben, für die ich Zeit aufwenden muss. Bei diesen Aufgaben geht es häufig nicht darum, neue Schaltungen zu entwickeln oder geliefertes Rohmaterial zu verarbeiten, sondern um die Überprüfung und Beschaffung von Bauteilen, die Unterstützung von Kollegen und vieles mehr. Wie die meisten Elektronik-Ingenieure benutze ich am häufigsten ein Multimeter und ein Oszilloskop. Natürlich sind auch ein paar Labornetzgeräte und ein Funktionsgenerator unverzichtbar, nicht zu vergessen ein Audioanalysator. Zusätzlich zu einem einfachen LötKolben verwende ich einen Reflow-Ofen und Heißluft zum Löten von SMDs. Um gut belichtete Fotos von Prototypen zu machen, habe ich auf dem Dachboden ein kleines Studio eingerichtet. Für die Simulation verwende ich seit den seligen DOS-Tagen am liebsten Micro-Cap.

Jan: Wir haben gehört, dass du einen Audio-Precision-Analyzer besitzt. Wie gefällt dir die Arbeit mit einem Gerät, von dem viele Audio-Enthusiasten nur träumen können?

Ton: Ohne einen guten Analyzer ist es fast unmöglich, High-End-Audioschaltungen zu entwickeln. Natürlich und zum Glück sind Messdaten nicht alles, aber sie sind sehr gut geeignet, um Unzulänglichkeiten

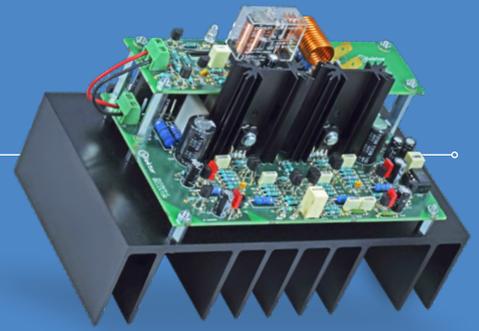
aufzudecken. Mit einem Oszilloskop und einem Multimeter kommt man nur bedingt weiter. Ich bin wirklich zufrieden, dass ich den AP-Analyzer viele Jahre lang benutzen konnte. Entscheidend ist jedoch, dass man weiß, wie und was man misst. Ohne gute Erfahrung und die richtigen Schlussfolgerungen kann man auch mit dem AP nicht viel erreichen. Das lernt man nicht an einem Tag.

Jan: Was sind deine Hauptinspirationsquellen für neue Entwürfe sowie den Einsatz neuer Bauteile und Techniken?

Ton: Für neue Entwürfe sind die Oldies eine sehr gute Inspirationsquelle. Im Internet findet man viele davon. Die Entwicklung von bipolaren Leistungstransistoren zum Beispiel ist nicht stehengeblieben. Was spricht also dagegen, einen alten Entwurf wieder aufzugreifen, ihn hier und da zu überarbeiten und mit modernen Bauteilen zu neuem Leben zu erwecken?

Jan: Deine Arbeit scheint sich in erster Linie auf analoge Elektronik zu konzentrieren, möglicherweise in Kombination mit digitalem Material, aber beschränkt auf „Hilfsfunktionen“. Wie wägst du die relative Bedeutung von analog und digital innerhalb deiner Entwürfe ab?

Ton: Ich mag nach wie vor die Standard-Logik-Serien, auch als Treiber oder in Kombination mit analogen Schaltungen. Aber ich habe gelernt, dass Mikrocontroller die Hardware billiger und kleiner machen, während Software ein Projekt vielseitiger macht. Programmieren hat mich noch nie gereizt; analoge Technik ist für mich „greifbarer“, aber das mag eine Illusion sein. Wenn ich einen Mikrocontroller verwende, dann ist es einer aus der AVR-Familie, den ich mit BASCOM programmieren kann. Gelegentlich ist so etwas sogar eine willkommene Abwechslung.



Jan: Du hast zu der Legion von High-End-Audioverstärkern beigetragen, die Elektor berühmt gemacht haben. Welche waren die renommiertesten und welche magst du am liebsten?

Ton: Einer der ersten Entwürfe, an denen ich gearbeitet habe, war im Jahr 1988 der LFA-150 [1] (allerdings nicht mein Design) und sein Derivat, der LFA-50-OA [2], dann der Medium-Power-Amp [3] und später der High-End-Poweramp, ein Verstärker mit bescheidenerer Leistung. Der HEXFET-Medium-Power-Amp [4] mit seinem Nachfolger, dem IGBT-Leistungsverstärker [5] (auf der gleichen Platine) mit speziell für den Audibereich entwickelten IGBTs, sowie Leistungsverstärker für aktive Subwoofer und verschiedene aktive Lautsprecher-systeme. Später kam dann der Verstärker Crescendo Millennium Edition [6] und der Klasse-T-Verstärker ClariTy 2x300W [7]. In jüngerer Zeit wurde ein 200-W-Audio-Leistungsverstärker der Klasse D [8] und die Kompakte Audio-Endstufe [9] entwickelt. Und dann natürlich der Fortissimo-100 [10]. Der Medium-Power-Amp wurde seinerzeit von einer deutschen Audiozeitschrift getestet und sehr positiv bewertet, und vielleicht ist das der Grund, warum ich ihn bevorzuge, aber ich möchte mich nicht wirklich auf einen Favoriten festlegen.

Jan: Dein Verstärker Fortissimo-100 wurde in bescheidenem Rahmen über die Website Elektor Labs entwickelt, hatte einen großartigen Start und verkaufte sich nach seiner „offiziellen“ Veröffentlichung im Elektor-Magazin noch besser. Erzählst du etwas über den Entwicklungs- und Veröffentlichungsprozess? Wo lagen die Schwierigkeiten?

Ton: Ich wurde gebeten, eine weitere neue Endstufe zu entwerfen. Ich dachte dabei hauptsächlich an den Medium-Power-

Amp und folgte weitestgehend diesem Konzept weit, abgesehen von der Endstufe. Das Problem war (und ist), dass bedrahtete Doppeltransistoren wie der MAT02 und der MAT03 nicht mehr hergestellt werden, und die kleinen Versionen in SMD-Bauweise waren nicht leistungsfähig genug. Also wandte ich mich den guten alten BC546B und BC556B zu, mit denen man einen sehr schnellen Verstärker bauen kann, der mit einer höheren Versorgungsspannung betrieben wird. Natürlich bevorzugt man heute die Miller-Kompensation, aber ich wollte das ursprüngliche Konzept so wenig wie möglich ändern. Ich habe nur ThermalTrack-Leistungstransistoren für die Ausgangsstufe verwendet, was es mir ermöglichte, zunächst in einem Proof-of-Concept einen symmetrischen Bootstrap zu versuchen. Dies funktionierte überraschend gut und brachte mir einen größeren Ausgangshub. Zur besseren Auswahl der BC-Transistoren im Bausatz wurden sie an so vielen Stellen wie möglich eingesetzt, einschließlich der Stromquellen der Differenzverstärker. Um die Leistung im Zaum zu halten, werden Kollektorwiderstände benötigt, die unter anderem wiederum eine stabilisierte Versorgungsspannung erzwingen.

Jan: Wir haben auf Elektor Labs gelesen, dass ein lineares Netzteil speziell für den Fortissimo-100 in Arbeit ist. Was ist der Grund dafür? Das SMPS800RE-Schaltnetzteil ist doch völlig in Ordnung, oder?

Ton: Für manche ist die Verwendung eines Schaltnetzteils ein Schimpfwort und ein absolutes No Go. Aber es gibt heutzutage gute Ausführungen, die speziell für den Audibereich entwickelt wurden. Eine davon ist halt das Netzteil SMPS800RE. Es macht die gesamte Stromversorgung schön kompakt. Natürlich ist der Rausch-

pegel bei einem Schaltnetzteil höher und es gibt eine gewisse HF-Welligkeit. Diese liegt jedoch weit außerhalb des Hörbereichs und wird zudem vom Verstärker wirkungsvoll unterdrückt. Um den Bedürfnissen derjenigen gerecht zu werden, die sich für den Verstärker interessieren, aber etwas gegen Schaltnetzteile haben, habe ich überlegt, einen analogen Regler zu bauen. Aber er sollte bescheiden bleiben und auch zu Hause leicht nachzubauen sein. Leider ist die Konstruktion heutzutage aufgrund der schlechten, verzögerten oder gar nicht vorhandenen Verfügbarkeit von Bauteilen eine ziemliche Herausforderung. Da ist man schnell gezwungen, Kompromisse einzugehen, und deshalb verwendet das Design Oldtimer wie TIP35C und TIP36C - die immer noch von mehreren Herstellern produziert werden und die bei verschiedenen Händlern ab Lager erhältlich sind.

Jan: Wie und wo siehst du dich in der Elektronik in einem Jahr arbeiten?

Ton: Ich gehe davon aus, dass ich mehr oder weniger das Gleiche machen werde wie jetzt, aber die Welt ist in Aufruhr, und es ist reine Spekulation, jetzt vorherzusagen, was die Zukunft bringt. ◀

RG – 220603-02



Passendes Produkt

➤ **Bausatz High-End-Verstärker Fortissimo-100**
SKU 20273: <https://elektor.de/20273>

WEBLINKS

- [1] „The Amp“, Elektor, 11/1988: <https://www.elektormagazine.de/magazine/elektor-198811/48741>
- [2] „Klasse-A-LFA“, Elektor, 10/1991: <https://www.elektormagazine.de/magazine/elektor-199110/29724>
- [3] „Medium Power Amp“, Elektor, 10-11/1990: <https://www.elektormagazine.de/magazine/elektor-199010/29400>
- [4] „HEXFET-Medium-Power-Amp“, Elektor, 12/1993: <https://www.elektormagazine.de/magazine/elektor-199312/30357>
- [5] „IGBT-Leistungsverstärker“, Elektor, 9/1995: <https://www.elektormagazine.de/magazine/elektor-199509/30840>
- [6] „Crescendo Millennium Edition“, Elektor, 4/2001: <https://www.elektormagazine.de/magazine/elektor-200104/1124>
- [7] „ClariTy 2 x 300 W Klasse-T-Verstärker“, Elektor, 6/2004: <https://www.elektormagazine.de/magazine/elektor-200406/1860>
- [8] „200W Class-D Audio Power Amplifier“, Elektor Labs, 10. August 2016:
<https://elektormagazine.de/labs/200w-class-d-audio-power-amplifier-150511>
- [9] „Kompakte Audio-Endstufe“ Elektor, 9/2013: <https://www.elektormagazine.de/magazine/elektor-201309/23316>
- [10] „High-End-Verstärker Fortissimo-100“, Elektor, 11/2022: <https://www.elektormagazine.de/magazine/elektor-281/61094>

Das besondere Projekt

Es gibt noch viel zu tun!

Bild 1. Dieser Tischtennisball schwimmt auf einem Wasserstrahl.

Von Ilse Joostens (Belgien) und Eric Bogers (Elektor)

Ilse Joostens ist treuen Elektor-Lesern wohlbekannt - nicht nur als Autorin der Kolumne „Aus dem Leben gegriffen“, sondern auch als Meisterin alter Nixie- und VFD-Röhren. Die Corona-Pandemie mit ihren Lockdowns und anderen Leiden ist auch an ihr nicht spurlos vorbeigegangen, wie die folgende Geschichte beschreibt.

Die Ursprünge des Geräts, das wir diesmal als besonderes Projekt vorstellen wollen, liegen viele Jahre zurück. Wir überlassen Ilse das Wort:

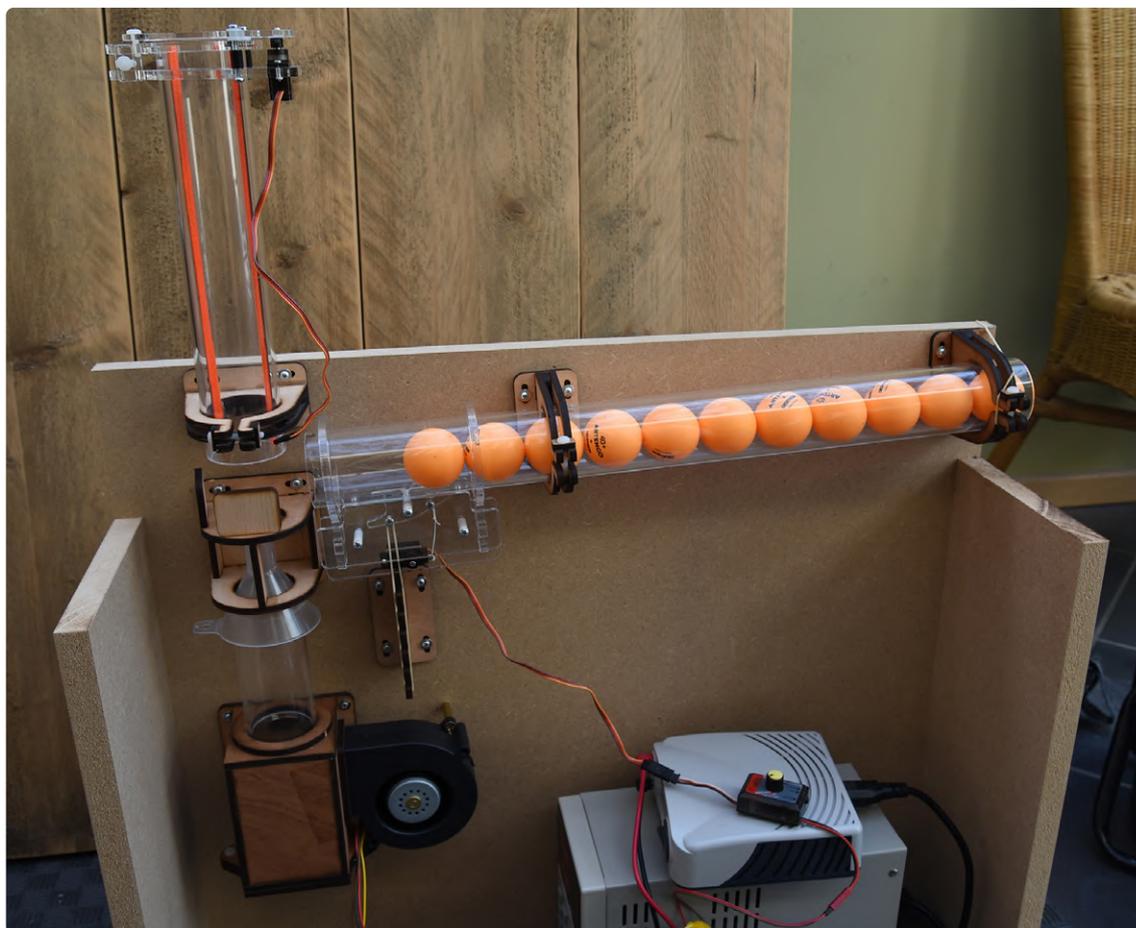
„Vor etwa 20 Jahren lebte ich als Single im flämischen Dendermonde und suchte nach einem Verein, in dem ich etwas unternehmen konnte. Zufällig gab es in der Nähe meiner Wohnung einen Schützenverein; ich wurde Mitglied. Oft wurde beim sogenannten ‚Spaßschießen‘ auf Tischtennisbälle geschossen, die auf Pressluft schwebten. Das Ärgerliche daran war, dass jedes Mal, wenn

die Tischtennisbälle getroffen wurden, (natürlich) alle Waffen entladen werden mussten und jemand neue Tischtennisbälle in den Luftstrom legen musste. Da wurde die Idee geboren, die Tischtennisbälle auf Knopfdruck durch ein Rohr zu befördern, aber niemand hatte eine Idee, wie das mechanisch funktionieren sollte, denn der Sinn war natürlich, dass die Bälle einzeln herauskommen sollten. Ich habe dann einen Entwurf auf einem Blatt Papier skizziert, aber letztendlich nichts aus der Idee gemacht.“

In der Zwischenzeit hatte sich Ilse als unabhängige Elektronikentwicklerin etabliert und sich auf alte Elektronikbauteile wie Vakuumröhren, Nixies und Vakuum-Fluoreszenz-Anzeigen (VFDs) spezialisiert. Dafür gab es einen ganz praktischen Grund - vieles in der altertümlichen Elektronik war schon entworfen, und die Konkurrenz in Sachen aktueller Elektronik aus China war schon damals stark. Die Entscheidung für einen Nischenmarkt war also naheliegend.*

„Dank COVID-19 wurde mir ziemlich langweilig; auch mein Schützenverein wurde vorläufig geschlossen. Um zu Hause ein wenig üben zu können, beschloss ich, mir ein paar Luftdruck- und Airsoft-Waffen zu

**Während der COVID-19-Krise erwarben die Chinesen praktisch alle noch vorhandenen Bestände an russischen Nixie-Röhren. Das ist einer der Gründe, warum die sechsstellige Nixie-Uhr von Elektor nicht mehr erhältlich ist.*



◀ Bild 2. Der Tischtennisball-Behälter.

kaufen. Das war auch ein willkommener Zeitvertreib während der Lockdowns.“

In Ermangelung von Nixie-Röhren und Plexiglas (auch eine Folge von COVID-19) wollte Ilse eine andere Richtung in der Elektronikwelt einschlagen, aber es sollte etwas Besonderes sein. Und da tauchte die Idee mit den Tischtennisbällen wieder auf, mit der Idee, mit Luftgewehren auf sie zu schießen. Etwas Ähnliches sieht man auf Jahrmärkten: an Schießständen, wo man auf Tischtennisbälle schießen kann, die auf Wasserstrahlen treiben (Bild 1).

„Derzeit habe ich nur einen funktionierenden mechanischen Prototyp gebaut. Was den Luftstrom betrifft, so habe ich ein Gebläse 9BMC24P2G001 von Sanyo Denki verwendet. Es ist leider nicht ganz billig, aber es liefert einen soliden Luftstrom und lässt sich leicht über ein PWM-Signal in der Blaskraft steuern. Das ist praktisch, denn durch die Modulation des Luftstroms kann man den Ping-Pong-Ball unregelmäßig auf und ab bewegen, um ihn zu einem etwas schwierigeren Ziel zu machen. Mithilfe eines demontierbaren Kabinetts, das auch als Pufferbehälter dient, und eines Pulvertrichters aus dem Chemiegroßhandel wird ein starker, konstanter Luftstrom erzeugt. Auf diese Weise kann kein Schmutz (Projektilstücke) in das Gebläse gelangen, und man kann den Boden des Gehäuses öffnen, um es zu reinigen.“

Die Tischtennisbälle werden in einem schräg angebrachten Plexiglasrohr (Durchmesser 50 mm, Innendurchmesser 44 mm, **Bild 2**) gelagert. Die Schwerkraft bewirkt, dass sie in Richtung des Luftstroms rollen. Es gibt keine Beschränkungen bezüglich der Länge des Rohrs; der Prototyp kann bis zu elf Tischtennisbälle aufnehmen. Auch eine Zuführung durch einen Trichter im Behälter ist möglich. Eine rotierende Kunststoffnocke lässt die Tischtennisbälle einzeln aus dem Behälter herausrollen.

▼ Bild 3. Dieser Mechanismus sorgt dafür, dass die Bälle nacheinander in den Luftstrom gelangen.



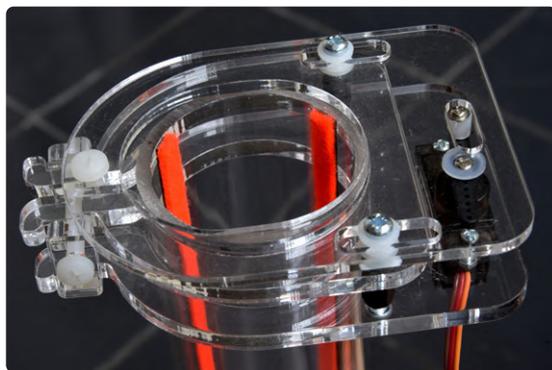
Bei jeder Hin- und Herbewegung der Nocke wird ein Tischtennisball freigegeben, während die übrigen Bälle automatisch blockiert werden. Die Nocke wird durch eine Zugfeder in einer Grundstellung gehalten und von einem Modellbau-Servo mithilfe einer Schnur bewegt (siehe **Bild 3**).

Ein solcher Mechanismus könnte auch die Basis für eine Kaffeekapselmaschine sein. Ein Konstrukt mit mehreren Röhren nebeneinander. Man wählt dann aus, welche Kapsel man möchte, und sie fällt in ein Fach an der Unterseite, wo man sie herausnehmen kann. Ich denke schon seit einiger Zeit über diese Idee nach, aber ich habe noch nichts Konkretes gemacht.“

Die Kugel, die aus dem Behälter [1] rollt, tritt dann in den Luftstrom über der Öffnung des Pulvertrichters ein und wird dann in das vertikale Plexiglasrohr katapultiert. Zwei Filzstreifen werden in das Rohr geklebt, um zu verhindern, dass die Kugel im Rohr „klappert“. Dies kann passieren, wenn sich die Kugel in der Röhre schnell bewegt, während die Luft zwischen der Kugel und der Wand der Röhre strömt.

„Wenn das vertikale Rohr oben offen wäre, würden die Tischtennisbälle nach oben geschleudert, ohne sich im Luftstrom zu „verfangen“. Vielleicht ist das eine Idee für eine Tischtennisball-Wurfmaschine? Um das unkontrolliertes Wegschießen der Tischtennisbälle zu verhindern, befindet sich am oberen Ende des Rohrs ein Schieber, der ebenfalls von einem Modellbau-Servo gesteuert wird (**Bild 4**). Die Tischtennisbälle fliegen nun bis zum Schieber oben in der Röhre hoch. Wenn sich der Schieber dann langsam öffnet, bewegt sich der Ball ein wenig nach oben und tanzt dann eine Weile auf und ab, bis er stabil auf dem Luftstrom schwebt [2]. Von da an kann das Gebläse mit einem PWM-Signal moduliert werden, um das Verhalten des Balls etwas unvorhersehbarer zu machen. In der Praxis schwebt

Bild 4. Dieser Schieber verhindert, dass die Bälle abgeschossen werden.



der Ball etwa 15...20 cm über dem Ende des Rohres. Ich habe auch Versuche ohne vertikales Rohr gemacht, aber dann war der Ball weniger stabil und fiel regelmäßig aus dem Luftstrom.“

Im Grunde kann man jede Art von Schusswaffe verwenden, um auf den Tischtennisball zu zielen - sogar Pfeil und Bogen. Wichtig ist allerdings, dass man möglichst billige Tischtennisbälle verwendet, da sie kein langes Leben erwartet. Außerdem sollten sie möglichst eine kontrastierende Farbe haben.

„Unmittelbar nach dem Bau der Mechanik begannen die Probleme mit dem Mangel an Bauteilen. Das Gebläse ist derzeit nicht mehr erhältlich (zumindest nicht kurzfristig), sodass bereits dafür eine Alternative gefunden werden sollte. Der Rest der Elektronik ist nicht so kritisch, da im Grunde nur zwei Modellbau-Servos angesteuert werden müssen und ein weiteres PWM-Signal für das Gebläse erzeugt werden muss. Im Moment ist das Projekt auf Eis gelegt, unter anderem wegen des Krieges in der Ukraine. Ich finde es nicht angebracht, ein Schießspiel auf den Markt zu bringen, wenn so viel Elend im Fernsehen und in den Medien zu sehen ist.

Im Moment bin ich mir nicht sicher, wie es weitergehen soll und ob dieses Projekt überhaupt Potenzial hat. Zumindest eine Fernbedienung wäre erforderlich. Mit jedem Knopfdruck bekäme man einen neuen Tischtennisball, eventuell mit einer Anzeige für den verbleibenden Vorrat an Bällen. Für letzteres könnte man IR-Reflexionssensoren am Behälterrohr anbringen. Andere zusätzliche Optionen wären die Wahl zwischen Schweben oder „Abschuss“ des Balls [2] oder die Wahl, ob das Gebläse moduliert werden soll oder nicht.

Ich würde gerne wissen, ob Interesse an dem Projekt besteht, und ob es Vorschläge und Wünsche für zusätzliche (oder andere) Funktionen gibt. Im Prinzip sollte es möglich sein, einen Bausatz mit mechanischen Teilen anzubieten, da alles einfach per Laser geschnitten und zusammengeklebt werden kann. Nur die Elektronik ist und bleibt ein Thema, zumindest bis Mitte 2023.“

(210591-02)SG

Haben Sie Fragen oder Kommentare?

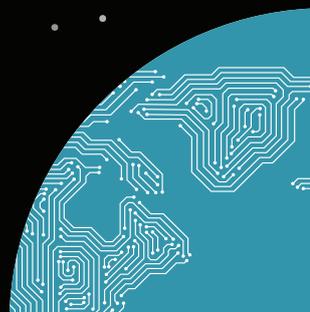
Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an die Elektor-Redaktion unter redaktion@elektor.de.

WEBLINKS

[1] Video 1 – Pingpongball-Behälter: <https://youtu.be/gTmytGPtHfQ>

[2] Video 2 – Schwebender Pingpongball: https://youtu.be/iH6_5-zcddw

EiC-???



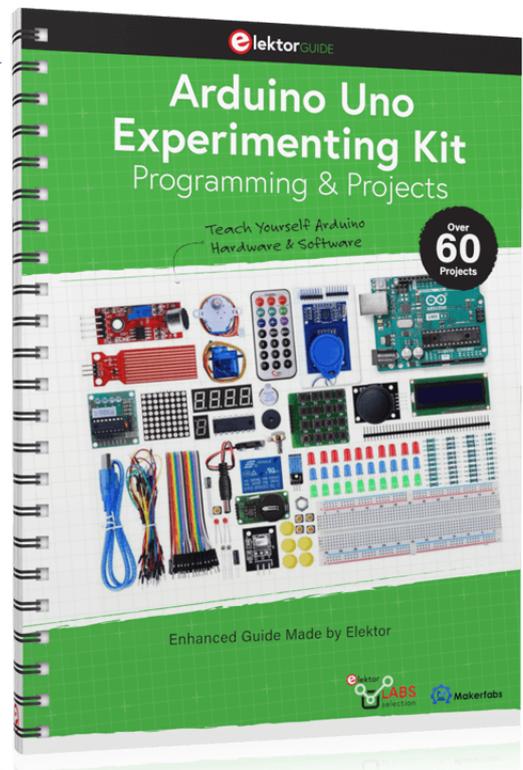
RFID-Tag lesen und RFID- Türschloss

Projektbeispiele aus dem Arduino-Experimentierpaket von Elektor

Von Dogan Ibrahim (Großbritannien)

RFID steht für *Radio-Frequency Identification* - eine drahtlose Technologie, die Geräte für Sicherheit, Zugangskontrolle und Warenverfolgung umfasst. Ein RFID-System besteht im Wesentlichen aus einem RFID-Lesegerät und einem oder mehreren Tags. Beide Hardwaregeräte sind in einem vielseitigen Bausatz für Arduino-Experimente enthalten, der von einer eigens verfassten Anleitung unterstützt wird. In diesem Artikel erkundet der Autor dieses Leitfadens, Dogan Ibrahim, als Beispiel für die Entwicklung eines Embedded-Projekts die RFID-Technologie und erhält dabei Hilfe von „unserem gemeinsamen Freund“, dem Arduino UNO.

Anmerkung der Redaktion. Dieser Artikel ist ein Auszug aus dem 238-seitigen Elektor-Buch *Arduino Uno Experimenting Kit - Programming & Projects*. Das Handbuch ist Bestandteil des Elektor Arduino Experimenting Bundle. Der hier präsentierte Auszug aus dem Buch wurde formatiert und leicht bearbeitet, um den Standards und dem Seitenlayout der Zeitschrift Elektor zu entsprechen. Da es sich um einen Auszug aus einer größeren Publikation handelt, können sich einige Begriffe in diesem Artikel auf Diskussionen an anderer Stelle des Buchs beziehen. Der Autor und die Redaktion haben jedoch ihr Bestes getan, um solche Fälle zu vermeiden, und helfen bei Rückfragen gerne weiter. Kontaktinformationen finden Sie im Kasten **Fragen oder Kommentare?**



Die RFID-Technologie nutzt elektromagnetische Felder, um Daten über kurze Entfernungen zu übertragen. RFID-Systeme werden hauptsächlich in Sicherheitsanwendungen eingesetzt. So kann beispielsweise eine Tür nur von einer Person geöffnet werden, die den richtigen RFID-Tag besitzt.

Das RFID-Lesegerät, das in dem Elektor-Bundle (bestehend aus einem Buch und einem Teilesatz) enthalten ist, wird als *RC522-Modul* bezeichnet (**Bild 1**) und hat die folgenden Eigenschaften:

- Betriebsfrequenz: 13,56 MHz
- Betriebsspannung: +3,3 V
- Betrieb sowohl mit SPI-Bus als auch mit I²C-Bus



Bild 1. RFID-Lesegerät und Tag für den Schlüsselbund.



Das RFID-Lesegerät wird mit Stiftleisten geliefert, die zunächst an die Buchsen am Rand des Lesegeräts gelötet werden müssen, bevor es verwendet werden kann.

Projekt 1: Ermitteln der Tag-ID

In diesem Projekt werden Sie lernen, die Tag-ID des mitgelieferten Tags im Seriellen Monitor der Arduino-IDE anzuzeigen.

Blockschaltung: Bild 2 zeigt das eher bildhafte als abstrakt/funktionale Blockdiagramm des Projekts.

Schaltbild: Die Verbindungen zwischen den Anschlüssen des UNO-Entwicklungsboards und dem RFID-Lesemodul sind wie folgt:

RFID-Reader-Pin	Entwicklungsboard-Port
SDA	10
SCK	13
MOSI	11
MISO	12
IRQ	nicht verwendet
GND	GND
RST	9
3.3 V	3.3 V

Bild 3 zeigt den Schaltplan des Projekts. **Achten Sie darauf, dass der Versorgungsspannungs-Pin nicht an +5 V angeschlossen wird!**

Programmlisting: Bevor Sie das RFID-Lesegerät verwenden können, müssen Sie die RFID-Bibliothek Ihrer IDE hinzufügen. Der Name der Bibliothek ist *MFRC522*, und die Schritte dazu sind wie folgt:

- › Laden Sie die Datei *rfid-master.zip* in einen Ordner herunter. Sie finden die Datei in der Support-Archivdatei für das Buch auf der Elektor-Store-Website [1].
- › Starten Sie die IDE.
- › Klicken Sie auf *Sketch* → *Bibliothek einbinden* → *.ZIP-Bibliothek hinzufügen*.
- › Navigieren Sie zu der gespeicherten *.zip*-Datei und klicken Sie auf *Öffnen*.
- › Nun können Sie die Bibliothek verwenden.

Die RFID-Reader-Bibliothek bietet viele Funktionen, die Sie durch Entpacken der Bibliotheksdatei sehen können. Zu den wichtigsten Bibliotheksfunktionen gehören:

<code>mfc522.PCD_Init()</code>	Initialisiert das RFID-Lesegerät
<code>mfc522.PICC_IsNewCardPresent()</code>	Suche nach einem RFID-Lesemodul
<code>mfc522.PICC_ReadCardSerial()</code>	Auswahl des zu verwendenden RFID-Lesegeräts
<code>mfc522.uid.uidByte[]</code>	Tag-ID in einem Array zurückgeben
<code>mfc522.PICC_HaltA()</code>	Lesevorgang beenden (Halt PICC)

Mit dem Programm *DumpInfo* aus der Arduino-IDE können Sie die Tag-ID Ihrer Karte ermitteln. Die Schritte dazu sind:

- › Starten Sie die IDE.
- › Klicken Sie auf *Datei* → *Beispiele* → *MFRC522* → *DumpInfo*.

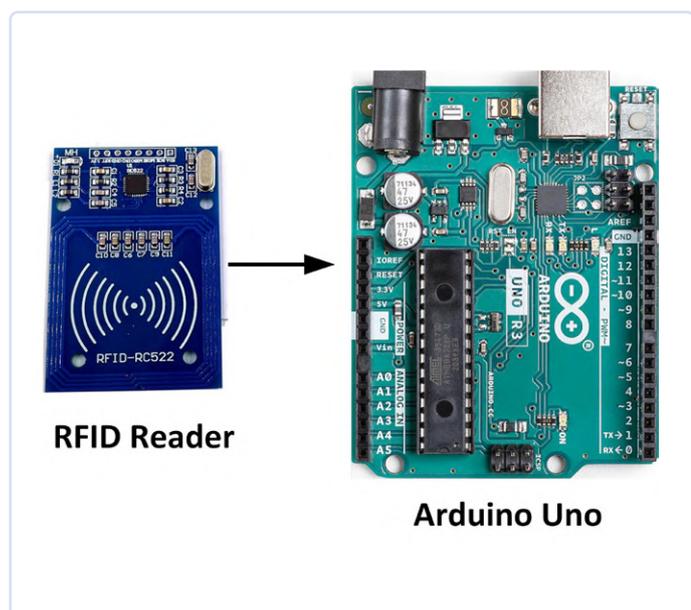


Bild 2. Bestandteile des Tag-Reader-Projekts.

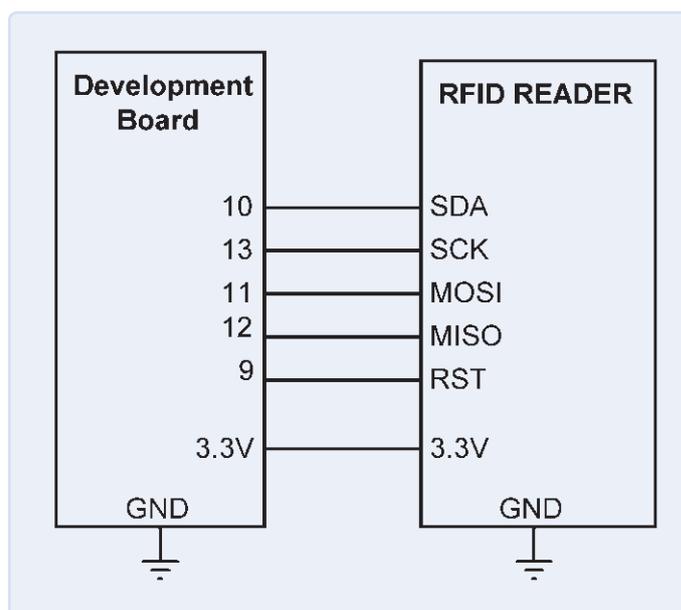


Bild 3. Schaltplan für das Tag-Reader-Projekt.

```

Firmware Version: 0x92 = v2.0
Scan PICC to see UID, SAR, type, and data blocks...
Card UID: 23 F0 58 A7
Card SAR: 08
PICC type: MIFARE 1KB
Sector Block 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 AccessBits
15 63 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
62 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
61 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
14 59 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
58 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
57 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
56 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
13 55 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
54 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
53 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
52 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
12 51 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
49 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
48 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
Autoscroll Show timestamp Both NL & CR 9600 baud Clear output

```

Bild 4. Daten im Tag-Speicher.

- › Kompilieren Sie das Programm und laden Sie es auf das Entwicklungsboard.
- › Starten Sie den *Seriellen Monitor*.
- › Legen Sie den weißen Tag auf das Lesegerät und lassen Sie ihn dort liegen, bis die Datenanzeige im Seriellen Monitor stoppt.

Die Daten sollten jetzt ähnlich wie in **Bild 4** aussehen. Es handelt sich um den 1 KB großen Speicher der Karte und deren Tag-ID. Der 1-KB-Speicher ist in 16 Sektoren (0..15) unterteilt, wobei sich jeder Sektor wiederum in vier Blöcke (0, 1, 2, 3) aufgliedert. Jeder Block kann 16 Bytes an Daten speichern (0..15). Daraus folgt:

$$16 \text{ Sektoren} \times 4 \text{ Blöcke} \times 16 \text{ Bytes} = 1.024 \text{ Byte Daten auf der Karte (also 1 KB)}$$

Der dritte Block eines jeden Sektors (das heißt, jeweils der oberste Block) wird als *Sector Trailer* bezeichnet und enthält die *Access Bits*. Diese steuern den Lese-/Schreibzugriff auf die übrigen Blöcke des

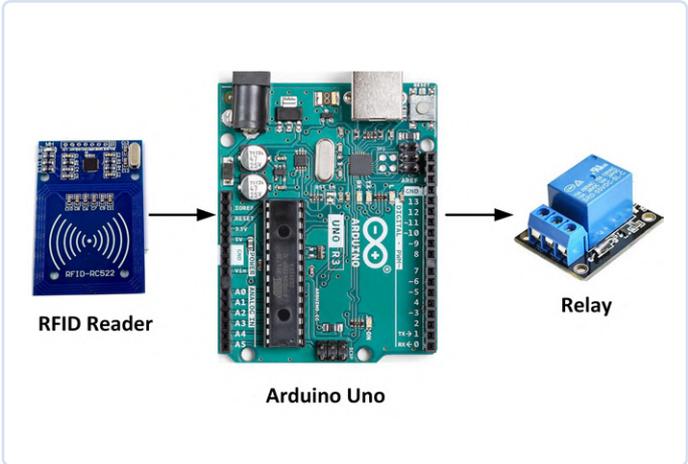


Bild 5. Blockschaltbild des Zugangskontrollprojekts.

Sektors. Daher sind nur die unteren drei Blöcke (0, 1 und 2) jedes Sektors für die Speicherung von Nutzdaten verfügbar. Das bedeutet, dass Ihnen $3 \times 16 \text{ Byte} = 48 \text{ Byte}$ pro 64-Byte-Sektor zur Verfügung stehen.

Block 0 des Sektors 0 wird als *Manufacturer Block/Manufacturer Data* bezeichnet und enthält die Herstellerdaten und die ID der Karte. Die Tag-ID wird auch unter *Card UID* (dritte Zeile in **Bild 4**) angezeigt. In diesem Projekt lautet die Tag-ID: **23 F0 58 A7**.

Projekt 2: RFID-Türschloss-Zugangskontrolle mit Relais

In diesem Projekt werden sowohl das RFID-Lesegerät als auch das Relais aus dem Experimentier-Kit an das Entwicklungsboard angeschlossen. Es wird angenommen, dass ein gesichertes Türöffnen mit Hilfe eines Relais unter dem Schutz eines RFID-Systems vorgenommen wird. Die Tür kann geöffnet werden, indem eine autorisierte Tag-Karte in die Nähe des RFID-Lesegeräts gehalten wird. Das Relais wird für 15 s aktiviert und fällt dann ab, so dass die Tür geschlossen ist.

Blockschaltbild: **Bild 5** zeigt das Blockdiagramm des Projekts. Auch hier werden Bilder der realen Platinen verwendet und nicht eine theoretische oder abstrakte Darstellung mit Funktionsblöcken.

Schaltplan: Der Schaltplan ist in **Bild 6** dargestellt. Im Gegensatz zu anderen „Schalt“-Projekten im Buch (die hier nicht besprochen werden), wird ein Relais zu Port 2 hinzugefügt.

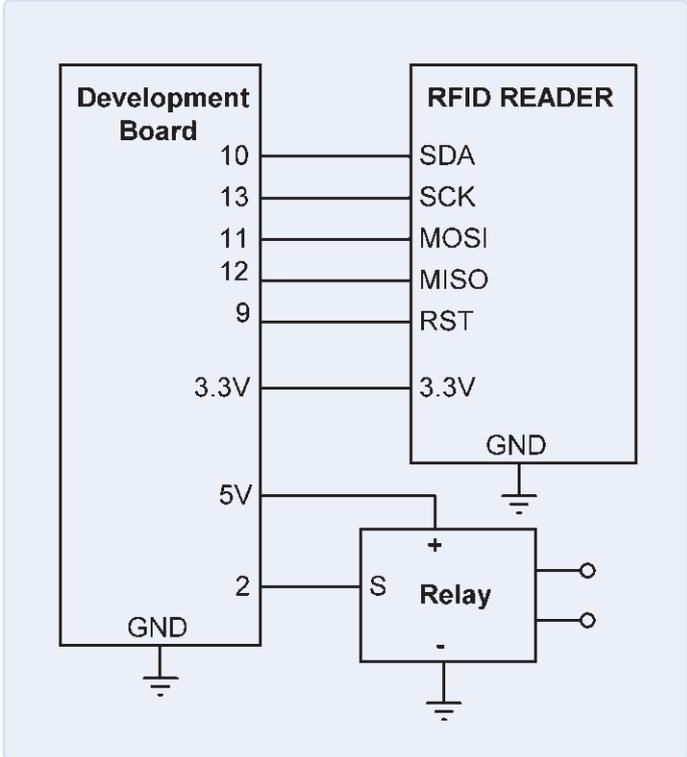


Bild 6. Bestandteile des Zugangskontrollprojekts.



Listing 1. Für das Projekt entwickeltes RFIDLock-Programm.

```
//-----  
//          RFID LOCK SYSTEM  
//          =====  
//  
// In this program, the RFID card reader is used with a relay. The relay  
// is only activated if an authorized tag is placed near the reader. The  
// relay stays ON for 15 seconds and then turns OFF. The Tag ID of the  
// authorized valid card in this example is: 23 F0 58 A7  
//  
// Author: Dogan Ibrahim  
// File  : RFIDLock  
// Date  : May, 2022  
//-----  
#include <SPI.h>  
#include <MFRC522.h>  
  
#define SS_PIN 10  
#define RST_PIN 9  
MFRC522 mfrc522(SS_PIN, RST_PIN);    // Create MFRC522 instance  
String ValidCard = "23F058A7";      // Valid Tag ID  
String TagID = "";  
int RELAY = 2;                       // RELAY at port 2  
byte i;  
  
void setup()  
{  
  pinMode(RELAY, OUTPUT);            // RELAY is output  
  digitalWrite(RELAY, LOW);         // Deactivate RELAY  
  SPI.begin();                      // Initiate SPI bus  
  mfrc522.PCD_Init();              // Initiate MFRC522  
}  
void loop()  
{  
  if (!mfrc522.PICC_IsNewCardPresent()) // Look for card  
  {  
    return;  
  }  
  
  if (!mfrc522.PICC_ReadCardSerial()) // Select the card  
  {  
    return;  
  }  
  TagID = "";  
  
  for (i = 0; i < 4; i++)            // Read 4 byte Tag ID  
  {  
    TagID.concat(String(mfrc522.uid.uidByte[i], HEX));  
  }  
  
  TagID.toUpperCase();              // Convert to upper case  
  mfrc522.PICC_HaltA();             // Stop reading  
  if(TagID == ValidCard)            // Valid card?  
  {  
    digitalWrite(RELAY, HIGH);      // RELAY ON  
    delay(15000);                   // Wait 15 seconds  
    digitalWrite(RELAY, LOW);       // RELAY OFF  
  }  
  else  
    digitalWrite(RELAY, LOW);       // RELAY OFF  
}
```



Programmlisting: Listing 1 zeigt das Programm, das für das RFID-gesteuerte Türschloss mit dem Namen *RFIDLock* entwickelt wurde. Zu Beginn des Programms werden die *SPI*- und *MFRC522*-Bibliotheken eingebunden. Die *Valid Card ID* wird in der Zeichenkette *validCard* gespeichert und *RELAY* Port 2 zugewiesen. Innerhalb der Funktion *setup()* wird *RELAY* als Ausgang konfiguriert und deaktiviert. Der *SPI*-Bus und der *MFRC522* werden ebenfalls initialisiert. Der Rest des Programms läuft innerhalb der Hauptprogrammenschleife ab. Hier wartet das Programm, bis eine Karte in die Nähe des Lesegeräts gebracht wird. Dann wird sie ausgewählt und eine *for*-Schleife gebildet, um die 4-Byte-Tag-ID in die Variable *TagID* einzulesen. Anschließend wird *TagID* mit der autorisierten Tag-ID in *validCard* verglichen. Bei Übereinstimmung ist der Benutzer autorisiert, und das Relais wird für 15 s aktiviert. Wenn die Karte nicht gültig ist, bleibt das Relais deaktiviert. Dieser Vorgang wird ständig wiederholt. **Bild 7** schließlich zeigt das auf dem Breadboard aufgebaute Projekt. ◀

RG — 230066-02

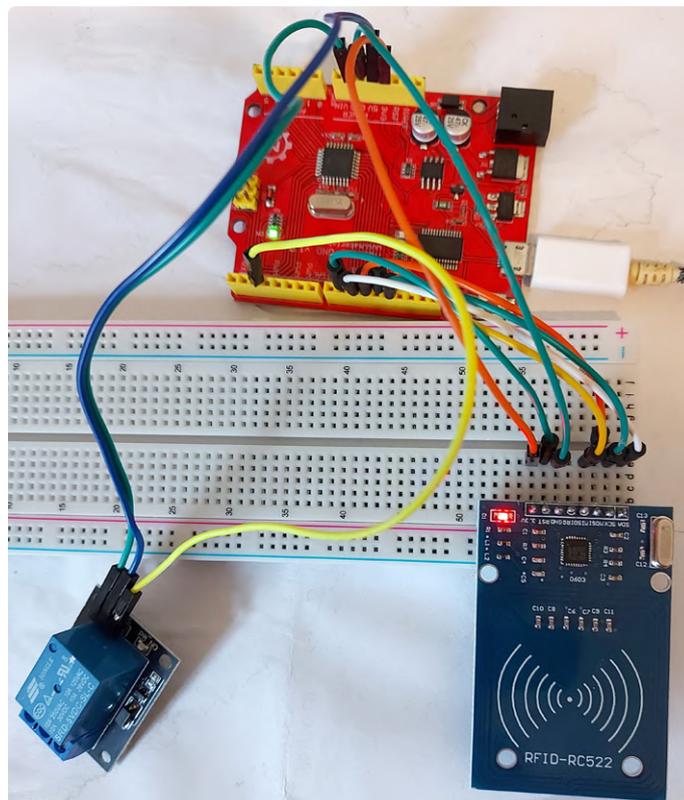


Bild 7. Das Zugangskontrollprojekt, auf einem Breadboard aufgebaut.

Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Senden Sie eine E-Mail an den Autor unter d.ibrahim@btinternet.com oder an Elektor unter redaktion@elektor.de.



Passendes Produkt

> Arduino Uno Experimentier-Bundle

Lieferumfang:

- Elektor-Buch mit über 60 Projekten (238 Seiten)
- Arduino-Experimentierkit inkl. Arduino Uno

SKU 20340:

www.elektor.de/arduino-uno-experimenting-bundle



WEBLINK

[1] Software-Archiv für das Bundle: <https://elektor.de/arduino-uno-experimenting-bundle>

Oszilloskop- Stromsonde für HF

Hochfrequenz-Strommessung leichtgemacht

Von Roberto Visentin (Italien)

Möchten Sie den Antennenstrom Ihres HF-Senders messen? Oder, wie ich, den Primärstrom einer Teslaspule? Diese Projekte erfordern einige Vorüberlegungen. Hier finden Sie Konstruktionsrichtlinien und ein Beispiel aus der Praxis.



Bild 1. Nahaufnahme meiner HF-Stromsonde.

In vielen Fällen sind Strommessungen ohne Gleichstromkomponente nützlich. Der häufigste Fall sind Stromwandler (CTs) für Wechselstromnetze. In diesem Artikel geht es um die Konstruktion von Stromwandlern für mittlere bis hohe Frequenzen, die wirklich einfach zu bauen sind. Die vorgestellten Formeln gelten auch für AC-Netzgeräte.

Sondenprinzip

Die Sonde in **Bild 1** ist für die Messung von bis zu $50 A_S$ (Spitze) in einem Frequenzbereich von 7 kHz bis zu mehreren zehn Megahertz ausgelegt. Das Schaltbild in **Bild 2** recht einfach: Die Leitung, deren Strom gemessen werden soll, wird durch den Ringkern geführt, bei dem es sich um einen gewöhnlichen FT 82-43 von Amidon handelt, der bis mindestens 50 MHz gut funktioniert.

Die Sekundärwicklung besteht aus zehn Windungen Draht, die gleichmäßig über den Kern verteilt sind. Falls verfügbar, verwenden Sie eine Litze mittlerer Stärke, aber das ist nicht zwingend erforderlich. Aufgrund des Verhältnisses von 1:10 Windungen beträgt der maximale Strom in der Sekundärwicklung $5 A_S$. Die Sekundärseite ist mit $0,2 \Omega$ belastet, was durch eine Parallelschaltung von fünf $1\text{-}\Omega$ -Widerständen realisiert wurde. Bei einem Spitzenstrom von $5 A_S$ beträgt die Spitzenspannung über diesen Widerständen $1 V_S$, was für Messungen mit einem Oszilloskop sehr praktisch ist.

Bei einem sinusförmigen Strom beträgt die durchschnittliche Verlustleistung an den Widerständen $R \times I^2 = R \times I_S^2 / 2 = 2,5 W$

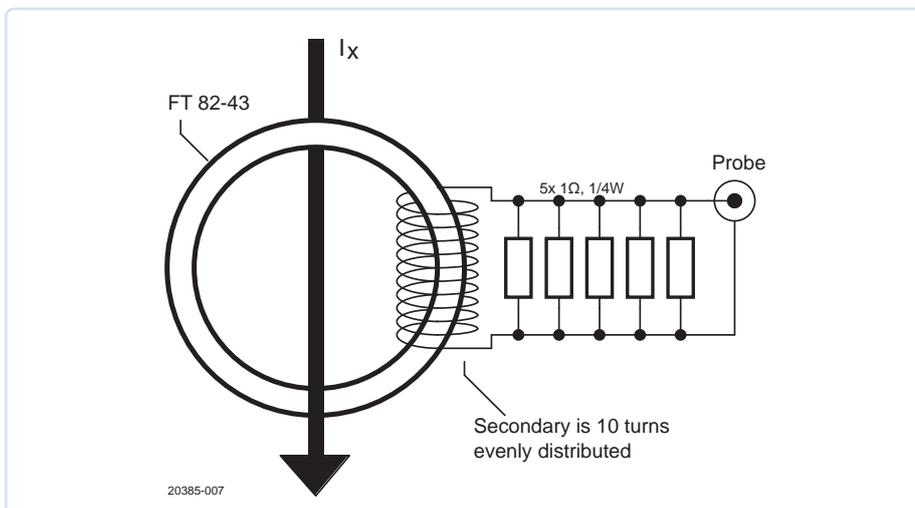


Bild 2. Die einfache Schaltung meiner HF-Stromsonde.

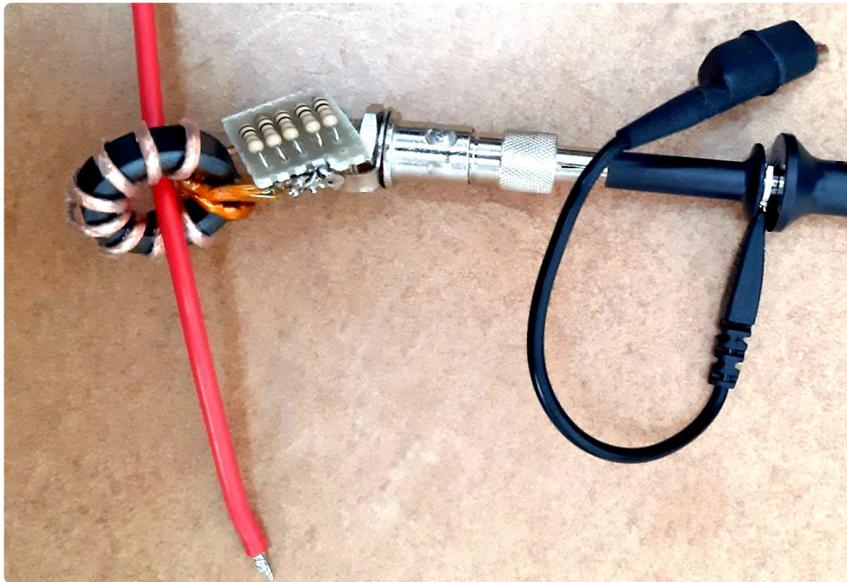


Bild 3. Die HF-Stromsonde im praktischen Einsatz.

oder 0,5 W pro Widerstand. Ein kontinuierlicher sinusförmiger Strom von 50 A_p kann nur mit Widerständen von 0,5 W oder größer gemessen werden. Aber wenn die Wellenformen gepulst sind oder sehr kurze Messungen durchgeführt werden, reichen auch Widerstände mit 0,25 W aus. Das war meine Wahl, weil ich das Design kompakt halten wollte, um eine bessere HF-Leistung zu erreichen. Okay, ich muss auch zugeben, dass dies die Widerstände waren, die ich gerade zur Hand hatte ...

Anwendung

Bild 3 zeigt die typische Anwendung der Stromsonde, die an einem BNC-Adapter für Oszilloskope angeschlossen ist. Das Gerät kann aber auch mit einer direkten Koaxialkabelverbindung zum Oszilloskop-Eingang verwendet werden, denn 1 V_S ist ideal für den Oszilloskop- $\times 1$ -Betrieb: In diesem Fall vermeidet ein kurzes Kabel Reflexionen im interessierenden Band, die entstehen, wenn das Koaxialkabel auf beiden Seiten fehlangepasst ist. Noch besser ist es, wenn das Koaxialkabel auf der Oszilloskopseite mit seiner charakteristischen Impedanz abgeschlossen wird: Viele moderne Oszilloskope bieten die einfache Möglichkeit, die Eingangsimpedanz auf 50 Ω einzustellen. In diesem speziellen Fall muss man aber bedenken, dass die Messung aufgrund der Parallelschaltung der 50- Ω -Last mit den 0,2 Ω der Sonde leicht abweicht (der Gesamtwiderstand wird zu 0,1992 Ω , was einen Skalierungsfaktor von 50,2 A/V ergibt).

Die Oszilloskopsonde darf aber keinesfalls ohne BNC-Adapter direkt mit Klemmen an die Widerstände angeschlossen werden, da bei der Messung hoher HF-Ströme selbst die kleinste ungeschirmte Schleife in den Sonden Artefakte in den Messungen erzeugt.

Berechnungen

Die Konstruktion des Stromwandlers ist nicht kompliziert, aber es müssen einige elektromagnetische Formeln beachtet werden. Zunächst zum Lastwiderstand R_L , der so klein wie praktisch möglich sein sollte, um die Verlustleistung zu minimieren, denn die zu messende Schaltung „sieht“ mindestens $R \times n^2$, wobei 1:n das Windungsverhältnis von 1:10 und R die Summe aus R_L (0,2 Ω) und dem sekundären Drahtwiderstand (ein paar Milliohm) ist. Wie bereits erwähnt, ist es sehr wichtig, dass die Sekundärseite gleichmäßig gewickelt ist, da die zu testende Schaltung sonst eine gewisse Streuinduktivität in Reihe aufweist. Wenn wir andererseits einen zu niedrigen Wert für R_L wählen, haben wir auch nur eine sehr kleine Spannung zum Messen, was zu Störsignalen auf den Leiterbahnen führt. Und schließlich sollte R_L größer sein als der Widerstand der Sekundärleitung. In meinem Fall habe ich 0,2 Ω gewählt, damit ich 1 V bei 5 A (50 A auf der Primärseite) erhalte, was dem zu testenden Stromkreis 2 m Ω hinzufügt.

Die Anzahl der Sekundärwindungen n bestimmt das Stromverhältnis. Bei einem Hochfrequenz-Stromwandler muss diese Zahl niedrig gehalten werden, um Eigenresonanz zu vermeiden, die durch Streukapazitäten zusammen mit einer hohen Induktivität verursacht wird. Bei Netzstromwandlern mit der recht niedrigen Frequenz von 50 Hz ist dagegen $n = 1000$ ein üblicher Wert. Normalerweise sind für n Zehnerpotenzen üblich, sodass das Stromwandlerverhältnis einfach ist, aber natürlich sind auch andere Werte möglich. Die höchste nutzbare Frequenz für einen Ringkern-Ferrit-Stromwandler hängt ab von:

- der Leistungsfähigkeit des Ferritmaterials
- der Eigenresonanz der Sekundärwicklung

- der Widerstandsinduktivität (Widerstand selbst und Anschlüsse)

Ein Design wie meines kann problemlos bis zu mehreren Dutzend Megahertz funktionieren, wenn ein geeigneter Ferrit wie das Material 43 von Amidon/Fair-Rite verwendet wird. Kerne mit hoher Permeabilität, die zur EMI-Unterdrückung verwendet werden, können ebenfalls eingesetzt werden, allerdings nur bis zu wesentlich niedrigeren Frequenzen. Kerne mit niedriger Permeabilität, die für Leistungsdrosseln und Induktivitäten mit hoher Güte verwendet werden, sind nicht zu empfehlen, da ihre Induktivität pro Windung zu gering ist, was sich auf den folgenden Punkt auswirkt. Die Wahl des Ferritkerns hat einen Einfluss auf die niedrigste nutzbare Frequenz, und zwar aus zwei Gründen:

- Um gute Messungen zu erhalten, muss die Reaktanz der Sekundärseite viel größer als R sein, da sie auch als Last wirkt. Da die Reaktanz $X = 2\pi f L$ ist, wirkt sich dies auf den unteren Frequenzbereich aus. In der Praxis sollte in diesem Bereich $X > 10 R_L$ sein.
- Wir müssen eine Sättigung des Kerns vermeiden, die bei hohem Strom und niedriger Frequenz auftritt. Die Sättigung von Ferriten tritt bei einem Feld von $B = 0,25 \dots 0,3$ T auf, aber um Nichtlinearitäten zu vermeiden, müssen wir bei maximalem Strom und minimaler Frequenz unter 0,2 T bleiben.

Aus den beiden genannten Gründen erfordern niedrige nutzbare Frequenzen mit einer geringen Anzahl von Windungen und einem hohen Strom tendenziell große Kerne. Die wichtigsten Kernparameter sind A_L , die Induktivität pro Windung, die bei HF-Kernen im Allgemeinen in nH/n² ausgedrückt wird, und der Kernquerschnitt S, der sich leicht aus den Kernabmessungen berechnen lässt. Beim FT 82-43 von Amidon ist $A_L = 470$ nH/n² und der Querschnitt beträgt 24,6 mm² (einfach zu berechnen aus Außendurchmesser, Innendurchmesser und Höhe).

Die Induktivität von zehn Windungen ist somit $470 \text{ nH} \times 10^2 = 47 \text{ } \mu\text{H}$ (denken Sie daran, dass die Induktivität von n^2 abhängt). Wenn wir akzeptieren, dass $X = 10 R \approx 10 R_L = 2 \Omega$, erhalten wir durch Umkehrung der Reaktanzformel $f > 6,8$ kHz als die untere Grenzfrequenz. Dies kann je nach Anwendung akzeptabel sein oder auch nicht, und ein

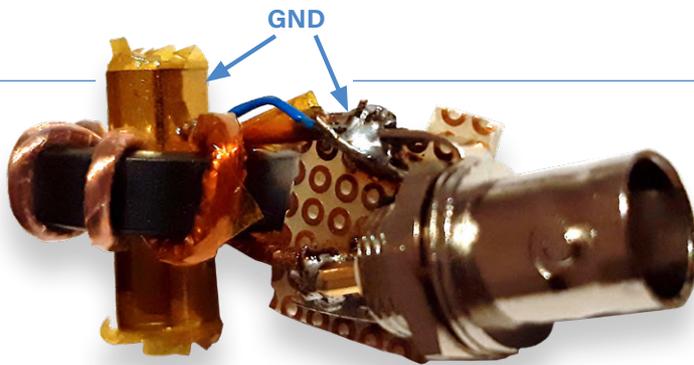


Bild 4. Verbesserte Sonde mit elektrostatischer Abschirmung.

größerer Kern mit einer höheren A_L hat eine niedrigere Grenzfrequenz. In meinem Fall – für die Messung einer „musikalischen“ Teslapule – war ich an Frequenzen über 500 kHz interessiert, daher war dieser Kern gut geeignet. Was die Sättigung angeht, so beträgt im sinusförmigen Bereich die Größe des Feldes B (magnetische Flussdichte) $|B| = U_S / (n \times 2\pi \times f \times S)$, wobei U_S die Spitzenspannung an der Wicklung ist. Dieser Ausdruck lässt sich durch Gleichsetzung der beiden Ausdrücke für den magnetischen Fluss ableiten: $\Phi_c = L \times I = n \times B \times S$, wobei wir I durch den absoluten Wert des AC-Magnetisierungsstroms $|I| = U / (2\pi \times f \times L)$ ersetzen. Wenn U und n in den vorangegangenen Entwurfsphasen festgelegt wurden, besteht die einzige Möglichkeit, ein niedrigeres f zu erreichen, darin, ein größeres S zu wählen, was wiederum einen größeren Toroid erfordert. In meinem Fall ist $U_p = 1\text{ V}$, $n = 10$, $S = 24,6\text{ mm}^2$ (achten Sie auf die korrekten Einheiten), also haben wir $B < 0,2\text{ T}$ für $f > 3,2\text{ kHz}$, was wiederum in Ordnung ist, zumindest für meine Anwendung. Es mag verwirrend sein, dass der Strom in der Berechnung nicht auftaucht: Er ist tatsächlich in U versteckt, denn wir wissen, dass $1\text{ V } 50\text{ A}$ entspricht.

Verbesserungen

Die kapazitive Kopplung zwischen Primär- und Sekundärwicklungen kann die Messungen bei den höchsten Nutzfrequenzen stören, oder sogar bei mittleren Frequenzen, wenn der Primärleiter einer hohen HF-Spannung ausgesetzt ist.

Das Design kann durch eine elektrostatische Abschirmung verbessert werden, die diese kapazitive Kopplung verhindert: In der Praxis wird hierzu der Primärdraht durch ein kleines Stück Metallrohr (typischerweise Kupfer oder Messing) geführt, das mit dem GND-Ausgang des Sekundärdrahtes verbunden ist, wie in **Bild 4** gezeigt. Dies ändert nichts an der magnetischen Kopplung, blockiert aber das elektrische Feld.

Fazit

Der Bau und die wichtigsten Entwurfskriterien eines HF-Stromwandlers zeigen, dass diese Materie weniger komplex ist, als es zunächst scheint. Ich hoffe, dass die hier vorgestellten Überlegungen und Formeln nützlich sind, um den Umgang mit Ringkernen zu vereinfachen und als Grundlage für Ihre eigenen Entwicklungen zu dienen. **◀**

Jörg Starkmuth — 200385-02

Über den Autor

Roberto Visentin ist ein kürzlich in den Ruhestand getretener Elektronikingenieur, der an Elektronik und Steuerungssystemen für Schiffe und Unterwasserroboter gearbeitet hat. Er arbeitet immer noch als freiberuflicher Berater und genießt es, mehr Zeit für die Entwicklung von Hobbyprojekten in seinem heimischen Elektroniklabor zu haben.

Sie haben Fragen oder Kommentare?

Wenn Sie technische Fragen oder Anmerkungen zu diesem Artikel haben, können Sie sich gerne an die Elektor-Redaktion unter der E-Mail-Adresse redaktion@elektor.de wenden.



Passende Produkte

- **OWON SDS1102**
2-Kanal-Oszilloskop (100 MHz)
(SKU 18782): <https://elektor.de/18782>
- **OWON XSA810**
Spektrum-Analyser (1 GHz)
(SKU 19714): <https://elektor.de/19714>



LPN liefert Leiterplatten aus Deutschland, vom Weltmarkt, aus NATO-Partnerländern oder mit anderen Restriktionen.

LPN ist nach ISO 9001:2015 Zertifiziert und das Personal beim Fraunhoferinstitut geschult.

LPN liefert jedes Basismaterial und jede in Deutschland oder am Weltmarkt verfügbare Technik.

- Multilayer bis 56 Lagen.
- Starrflex, Flex, Semiflex.
- Aluminium, auch Bergquist, Kupferkern, Messingkern, Stahlkern.
- Teflon, auch Rogers.
- Montagehilfen Kaptonband, Abziehlack und Weiteres.

LPN Qualitätsprüfungen

- 100% Kontrolle
- Kupferstärkenmessung mit Magnetfeld Messgeräten.
- Nachmessen gedruckter Induktivitäten.
- Schlibbildauswertung.
- Lot-Benetzungs-Test.
- Delaminations-Test.
- alle Fertigungsstätten halten ISO 14001 ein.

LPN Dienstleistungen

- Datenaufbereitung incl. Nutzaufbau,
- Machbarkeitsprüfung,
- EMPB.
- geklebte Vorlagen digitalisieren.
- Filme digitalisieren.
- Leiterplatten klonen.
- Leiterplatten nachlayouts.
- Terminaufträge.
- Abruflager für Jahreslose.

Profitieren Sie von den LPN Qualitätsstandards und den weltweiten Kontakten.

LPN Leiterplatten Nord GmbH

Hermann-Bössow-Straße 13-15
23843 Bad Oldesloe
leiterplatten-nord.de

Anfragen/Bestellungen:

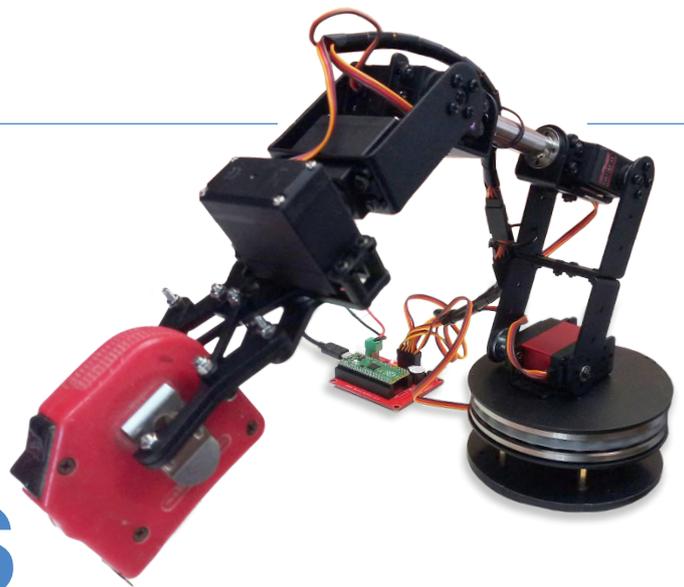
lpn@lp-nord.de
Telefon 04531 1708 0



Nichts für Anfänger

Makerfabs 6 DOF Robot Arm

Mit Raspberry Pi Pico und MicroPython



Von Clemens Valens (Elektor)

Ein Roboterarm, der sich dank sechs Gelenken in viele Richtungen bewegen und Gegenstände aufnehmen kann, ist ein lustiges Spielzeug, das noch mehr Spaß macht, wenn es sich mit MicroPython auf einem Raspberry Pi Pico steuern lässt. Mit diesem Bausatz können Sie selbst einen leistungsstarken Roboterarm ganz aus Metall bauen.

Worum handelt es sich?

Der 6-DOF-Roboterarm mit Raspberry Pi Pico ist ein Roboterarm, der von einem Raspberry Pi Pico Mikrocontroller-Board gesteuert wird. „DOF“ ist die Abkürzung für „Degrees of Freedom“ (Freiheitsgrade), was sich bei einem Roboterarm auf die Gelenke bezieht. Der hier vorgestellte Arm hat sechs, sodass er sich um sechs unabhängige Achsen bewegen kann. Jeder DOF erfordert einen Servo. Der Arm wird von einem Raspberry Pi Pico gesteuert – dieser steckt auf einer kleinen Erweiterungsplatine, die die Anschlüsse für die Servos, die Stromversorgung, einen Summer und einen Drucktaster bereitstellt (**Bild 1**). Die RP2040-MCU des Pico soll in MicroPython programmiert werden, aber das ist natürlich nicht obligatorisch. Der Treiber und die Beispiele sind jedoch in MicroPython geschrieben.

(Fast) alles inklusive

Der Roboterarm wird als vollständiger Bausatz mit allen Teilen geliefert (**Bild 2**). Ein Netzteil mit 5 VDC/6 A wird mitgeliefert, ebenso ein Netzkabel – dieses hat allerdings einen amerikanischen Stecker und muss daher ersetzt oder mit einem Adapter versehen werden. Ansonsten wird außer einem Computer für die Programmierung und einem Micro-USB-Kabel nichts weiter benötigt.

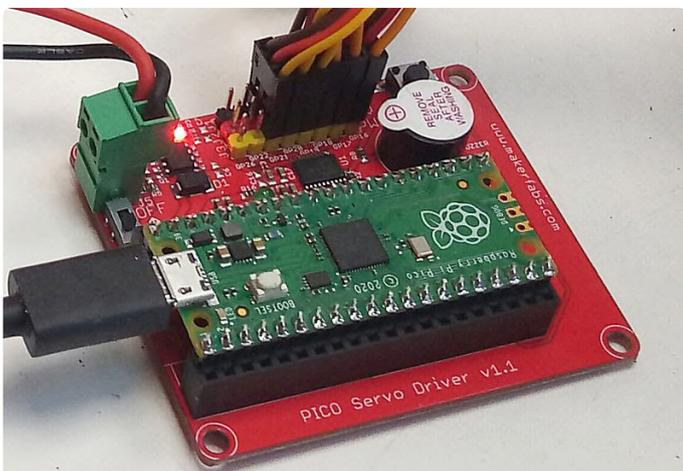


Bild 1. Ein Raspberry Pi Pico steuert die sechs Servos des Roboterarms.



Bild 2. Der 6-DOF-Roboterarm wird als Bausatz geliefert.



Es gibt sogar einen kleinen Schraubendreher, aber ich habe lieber meine eigenen Werkzeuge verwendet.

Es gibt eine ganze Reihe von Teilen und noch mehr Schrauben und Muttern, von denen einige aber Ersatzteile sind. Bis auf die Grundplatte aus Acryl und den „Armknöchel“ aus Stahl sind alle Teile aus Aluminium. Wenn man sie richtig zusammensetzt, erhält man kein wackeliges Plastikspielzeug, sondern einen starken, soliden Metallarm. Obwohl fast alles aus Aluminium ist, wiegt der zusammengebaute Arm 1,3 kg (ohne Netzteil).

Als Lernwerkzeug gedacht

Der 6-DOF-Roboterarm ist vor allem ein Lehrmittel, um die Programmierung in MicroPython zu lernen. Er ist ein spaßiges Gerät, das sich auf viele Arten bewegen kann, und er kann kleine Gegenstände wie sein eigenes Netzteil (circa 6 cm × 4 cm × 8,5 cm) aufheben. Wie ein Demo-Video zeigt, kann man damit zum Beispiel das Spiel „Türme von Hanoi“ spielen. Für industrielle Anwendungen oder hochpräzise Aufgaben ist er nicht geeignet. Er hat zwar einen Greifer, aber ich würde ihm mit einem Werkzeug darin nicht trauen (**Bild 3**).

Was Sie vorab wissen sollten

Der Zusammenbau ist nichts für Halbherzige. Die Anleitung (nicht mitgeliefert, man muss selbst danach suchen) ist nur auf Chinesisch und obendrein unklar und unvollständig. Es gibt ein kurzes Video zum Zusammenbau, aber auch das ist unvollständig. Mit etwas Geduld und gesundem Menschenverstand ist der Zusammenbau jedoch machbar, da die Konstruktion nicht sehr kompliziert ist. Das einzige komplexe Teil, der Greifer, ist bereits vormontiert (**Bild 4**). Allerdings gibt es ein paar Dinge, die man im Voraus wissen sollte (und die ich auf die harte Tour lernen musste):

- › Stellen Sie sich darauf ein, einiges wieder zerlegen zu müssen, um es beim zweiten Mal richtig zusammensetzen zu können.
- › Beginnen Sie unten, das heißt an der Grundplatte mit der drehbaren Plattform, und arbeiten Sie sich nach oben vor.
- › Das große Lager der rotierenden Basis besteht aus drei Teilen. Ich habe einen leichten Unterschied in der Drehqualität zwischen den beiden möglichen Montagevarianten festgestellt, aber das scheint letztlich keine Rolle zu spielen, da es nur als Totlast fungiert.
- › Verwenden Sie zuerst die beiden roten YF-6125MG-Servos (**Bild 5**). Mein Bausatz enthielt drei davon (einen als Ersatz), aber in der Anleitung und im Video werden sie überall verwendet. Diese Servos sind die stärksten und können das meiste Gewicht tragen – sie sollten daher am unteren Teil des Arms angebracht werden.
- › Vergewissern Sie sich, dass sich alle Servos in ihrer Mittelstellung befinden, bevor Sie sie montieren. Bei den schwarzen MG-996R-Servos können Sie dies manuell/visuell mit einer Art Servozeiger/-arm (im Bausatz enthalten) tun. Die roten YF-6125MG-Servos haben jedoch keine Endanschläge, sodass Sie nicht sehen können, wann sie sich in der Mittelstellung befinden. Daher habe ich den Pico für die Zentrierung der Servos während des Zusammenbaus verwendet. Das hört sich kompliziert an, ist aber eigentlich recht einfach. Ich würde sogar vorschlagen, schon vor der Montage die MicroPython-Entwicklungsumgebung einzurichten und die Demos mit

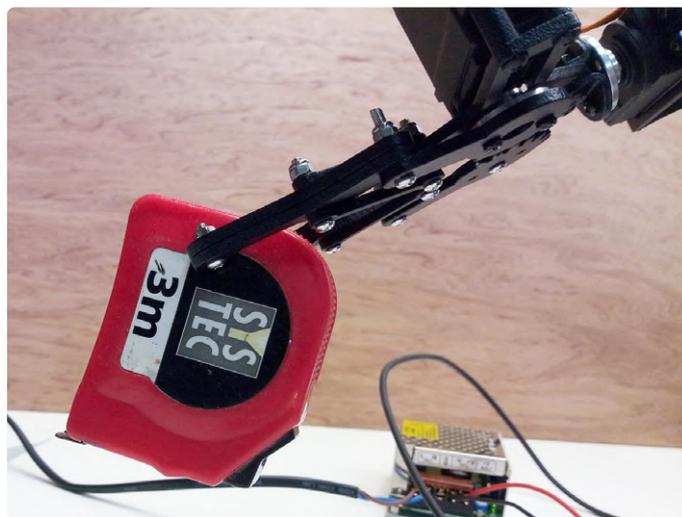


Bild 3. Der Arm kann ein Werkzeug halten, aber ich würde ihm damit nicht trauen.

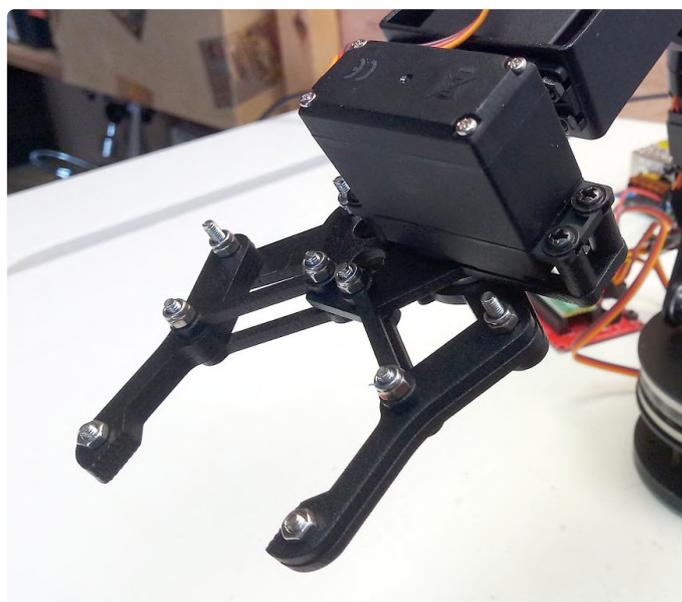


Bild 4. Der Greifer ist bereits vormontiert, Sie müssen nur noch den Servo anbringen.

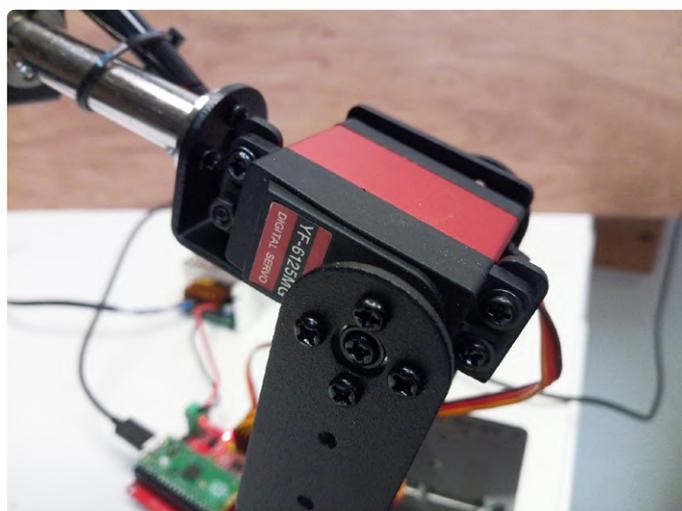


Bild 5. Die YF-6125MG-Servos sind leistungsstark und sollten den Großteil des Gewichts des Arms tragen.

unmontierten Servos auszuführen. Sobald Sie verstehen, wie man die Servos von MicroPython aus steuert, können Sie sie zentrieren und in den Arm einbauen.

- Bei allen Baugruppen wird der Servo zuletzt eingebaut. Er wird im Allgemeinen mit den Schrauben mit der Dichtung (Typ Mexican Hat) befestigt.
- Achten Sie darauf, alle Schrauben fest anzuziehen, denn das ist schwierig, wenn der Arm einmal montiert ist.
- Der vormontierte Greifer wirkte wackelig, daher habe ich die selbstsichernden Muttern etwas angezogen. Das Ergebnis war, dass der Servo, der den Greifer steuert, durchbrannte, da der Greifer offenbar zu schwergängig geworden war (obwohl ich ihn manuell bewegen konnte). Glücklicherweise liegen dem Bausatz zwei Ersatzservos bei.
- Es gibt keinen Schaltplan, es gibt sechs Servos, und die Servoanschlussleiste hat acht Positionen. Glücklicherweise ist sie farblich gekennzeichnet, sodass die richtige Polarität leicht zu finden ist. Schließen Sie die Servos von unten nach oben an. Der unterste Servo (die rotierende Basis, „0“ oder „A“ in der Software) wird an GPIO16 angeschlossen, der oberste (der Greifer, „5“ oder „F“ in der Software) an GPIO21. Die Software erlaubt eine Neuordnung der GPIO-Pins und die Drehrichtung jedes Servos kann geändert werden, sodass Sie hier gegebenenfalls etwas korrigieren können.
- Die Grundplatte hat viele Löcher, aber keine zur Befestigung der Pico-Platine oder des Netzteils.
- Stellen Sie sicher, dass das Netzteil 5 V liefert, bevor Sie es an den Arm anschließen. Die Ausgangsspannung kann von etwa 4,5 V bis 6,5 V eingestellt werden.

Vorsicht!

Bauen Sie den Roboterarm nicht einfach zusammen und starten Sie die Demo, ohne vorher alle Servos getestet zu haben. Der Arm könnte wild rotieren und um sich schlagen und dabei Gegenstände vom Tisch fegen oder Ihnen ins Gesicht schlagen. Diese Servos sind stark! Ich habe es geschafft, dass der Arm seine eigene 4 mm dicke Grundplatte zerstört hat, weil ein Servo nach unten dagegen schlug, statt nach oben zu gehen. Insbesondere die roten YF-6125MG-Servos bewegen sich ziemlich schnell, also Vorsicht. Ich empfehle dringend, den Arm auf der Werkbank (**Bild 6**) oder einer schweren Platte zu befestigen und den Platz um ihn herum freizuhalten. Beachten Sie auch, dass die roten YF-6125MG-Servos keine Endanschläge haben und möglicherweise in die falsche Richtung laufen, um ihre Zielposition zu erreichen. Dies kann passieren, wenn der Arm heruntergeklappt ist und dann eingeschaltet wird. Ich empfehle daher dringend, den Arm vor dem Einschalten in die senkrechte Position zu bringen, siehe (**Bild 7**).

Nichts für Anfänger

Der 6 DOF Robot Arm mit Raspberry Pi Pico von Makerfabs ist kein Bausatz für Anfänger. Der richtige Zusammenbau erfordert Geduld und Ausdauer. Obwohl der Arm dank seines in MicroPython geschriebenen Treibers und der Demo recht einfach zu bedienen ist, sollte er Schritt für Schritt gezähmt werden, da sonst Unfälle passieren können. Seine Servos sind schnell und stark, und der vollständig ausgefahrene Arm ist 44 cm lang (50 cm inklusive der Basis).



Bild 6. Festklemmen des Roboterarms an der Werkbank sorgt für Sicherheit beim Experimentieren.

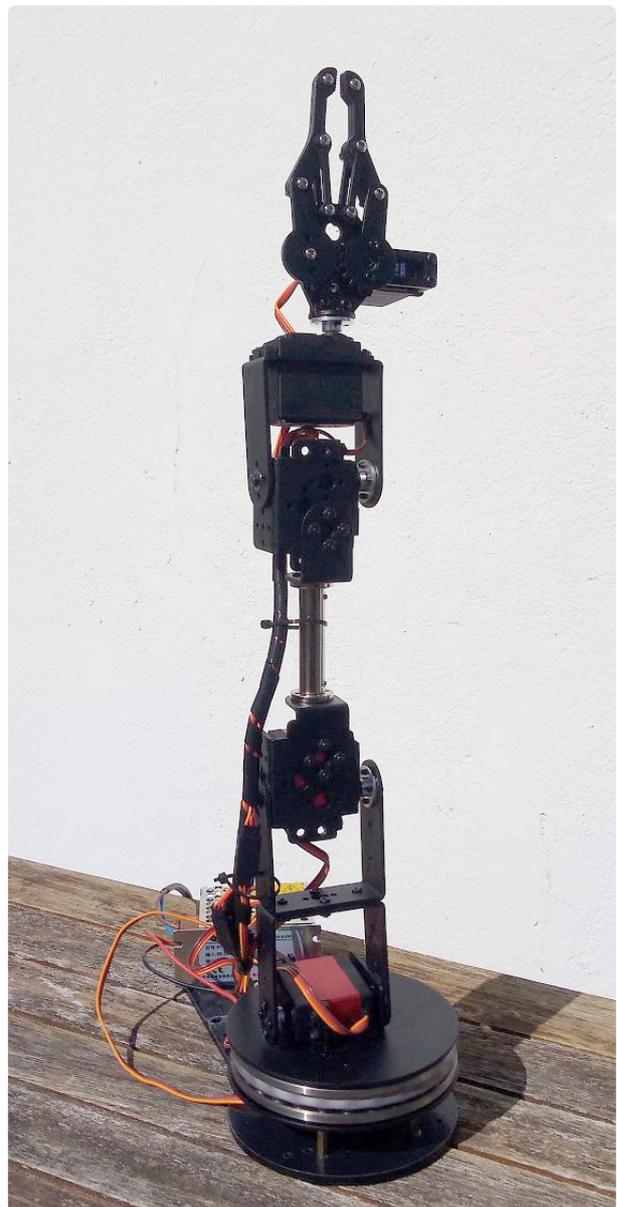


Bild 7. Die sicherste Ausgangsposition ist die senkrechte.



Es wird dringend empfohlen, den Arm in der Mitte einer freige-räumten Fläche von 1 m² am Boden zu befestigen. Sie wollen doch nicht, dass er Ihren Laptop zerschlägt oder Ihre Kaffeetasse vom Tisch fegt, oder? ◀

Jörg Starkmuth — 220232-02



Haben Sie Fragen oder Kommentare

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Senden Sie eine E-Mail an den Autor unter clemens.valens@elektor.com oder kontaktieren Sie Elektor unter redaktion@elektor.de.



Passende Produkte

- **Makerfabs 6 DOF Roboter-Arm mit Raspberry Pi Pico**
SKU 20130: www.elektor.de/20130
- **Elektor Raspberry Pi Pico Experimentier-Bundle**
SKU 19834: www.elektor.de/19834

WEBLINKS

- [1] Makerfabs 6 DOF Roboterarm bei Elektor: <https://www.elektor.de/makerfabs-6-dof-robot-arm-with-raspberry-pi-pico>
- [2] Demo-Video: <https://youtu.be/8dB1W6Jd07g>
- [3] Zusammenbau-Anleitung (Zip-Datei): <https://www.makerfabs.com/desfile/files/Raspberry-Pi-PICO-6-DOF-Robot-Arm.zip>
- [4] Zusammenbau im Video: <https://youtu.be/NqL9n3avBbY>
- [5] MicroPython-Treiber und Beispiele: https://github.com/Makerfabs/PICO_Merchanical_Hand_Driver
- [6] Webseite Makerfabs: <https://makerfabs.com/raspberry-pi-pico-6-dof-robot-arm.html>



wheel_me

Genius

Das erste und einzige autonome Wheel der Welt. Unsere innovative Technologie verwandelt fast alles in einen mobilen autonomen Roboter.

 Vermeidet statische und dynamische Hindernisse	 Omnidirektionale Fahrtrichtung
 State of art Sensor-Technologie	 Benutzerfreundliche Bedienung via App
 Flexible Anpassung der Nutzlast	 Einfaches Aufladen im Prozess

mehr details @ www.wheel.me

Quelle: P. Haring-Kuipers, erstellt mit DALL-E mit der Zielvorgabe: „AI working on a painting“.

Generative KI

Wer hat das eigentlich gemacht?

Von Priscilla Haring-Kuipers (Niederlande)

Es dreht sich alles um Künstliche Intelligenz. DALL-E und ChatGPT - beide von OpenAI - haben viele in ihren Bann gezogen und generative KI in der Gesellschaft etabliert.

K steht für Kunst

Die größte Debatte um visuelle generative KI wie DALL-E [1] dreht sich um die Kunst, mit der die KI trainiert wurde, ohne die ursprünglichen Künstler zu fragen. Für uns Menschen ist es in Ordnung, wenn wir uns von Kunst inspirieren lassen. Wir sind sogar einverstanden, wenn angehende Künstler bestehende Kunst als Ausgangspunkt kopieren, aber die Idee dahinter ist, dass man dann dazu übergeht, eigene Werke zu schaffen. Warum kann ein System Online-Kunstwerke nicht auf diese Weise nutzen? Ein Grund dafür ist, dass dies nicht in Betracht gezogen wurde, als der Künstler seine Kunst online stellte. Sie haben ihre Kunst nicht für das Training einer KI zur Verfügung gestellt, und niemand hat sie gefragt, ob sie ihre Kunst auf diese Weise genutzt wissen wollen. Die Regeln des Spiels wurden plötzlich geändert, und jetzt müssen wir alle gemeinsam herausfinden, welche neuen Regeln wir akzeptieren und wie sie angewendet werden sollen. Ein weiterer Grund ist, dass sich die KI nicht von der Kunst inspirieren lässt, sondern einen Datensatz aufbaut, mit dem sie ein Modell trainiert. Wir verstehen Intelligenz meist durch die Brille unserer eigenen Erfahrungen und erwarten, dass KI ähnlich funktioniert wie die menschliche Intelligenz. Das ist aber nicht der Fall. Wir sind multimodale Wesen mit direkter und indirekter Verarbeitung, assoziativem Lernen, emotionalen Filtern, Neurochemie und einer

ganzen Reihe anderer Faktoren, die unser Denken beeinflussen. Die künstliche Intelligenz kann mehr Daten viel schneller verarbeiten als wir, aber sie tut dies nicht auf dieselbe Weise. Menschliche Intelligenz braucht Jahre, um sich ihrer selbst bewusst zu werden, emotionales Verständnis zu entwickeln und die kreativen Sprünge zu machen, die zu etwas Künstlerischem führen können. Die KI wird sich nicht auf dieselbe Weise entwickeln, wenn wir ihr nur mehr Zeit oder mehr Rechenleistung geben. Wir formen die KI.

Es kommt ganz auf Sie an

Um bessere Modelle zu erstellen, die sich vielleicht so verhalten wie wir, wird Data Mining eingesetzt. Bei ChatGPT [2] ist es vielleicht weniger offensichtlich, aber diese KI wird auch an Ihnen trainiert. An jedem von uns. Massive Textbestände und unsere schriftlichen Interaktionen aus dem gesamten Internet werden verwendet, um das linguistische Modell zu trainieren, das den Kern von ChatGPT bildet. Wenn Sie mehr über diese zugrundeliegenden linguistischen Modelle erfahren möchten, empfehle ich Ihnen, sich „Speaking of Intelligence“ von Hannah Fry [3] anzuhören - eine Folge ihrer *DeepMind*-Podcast-Serie. Eines der Probleme des Modells besteht darin, dass es ungewollt rassistisch, aggressiv und einfach grauenvoll ist. Es ist durch die Inhalte des Internets trainiert und hat kein eingebautes Wertesystem, nach dem es diese Inhalte beurteilen könnte. Wie erwirbt es solche Werte? Durch Menschen. Schlecht bezahlte Bildschirmarbeiter, die ihren Tag damit verbringen, sich unmoralische Texte und Bilder anzusehen und sie entsprechend zu kennzeichnen. So könnte das Modell lernen, schlechte Inhalte zu erkennen und sie nicht mehr zu produzieren [4]. Auch bei der Moderation von Social-Media-Inhalten kommt dieser Prozess zum Einsatz. Dass das Modell immer besser wird, bedeutet, dass Menschen stundenlang Inhalte korrekt als „Vergewaltigung“, „rassistische Inhalte“, „Kindesmissbrauch“, „Gewalt“ und dergleichen kennzeichnen. Ein Großteil der Intelligenz der KI ist also unsere eigene. ChatGPT liegt auch oft sachlich falsch. Stack Overflow hat die Verwendung von ChatGPT vorerst gesperrt, weil „das Hauptproblem darin besteht, dass die Antworten, die ChatGPT produziert, zwar häufig falsch sind, aber typischerweise so aussehen, als könnten sie richtig sein, und dass die Antworten sehr leicht zu produzieren sind.“ [5] Da die KI Informationen in einem „lockeren“ Gesprächsstil gibt, neigen wir dazu, die Informationen leichter zu verarbeiten und sie schneller als zutreffend zu beurteilen, wenn wir sie überhaupt beurteilen. Ich möchte Sie bitten, bei Ihrer nächsten Interaktion mit ChatGPT daran zu denken, dass Sie lediglich mit dem Median des Internets sprechen und den Antworten entsprechend vertrauen und nicht vertrauen.

KI braucht BGE

Wir könnten der fortschreitenden Automatisierung der Arbeit einen Riegel vorschieben, indem wir jedem ein

Bedingungsloses Grundeinkommen (BGE) zur Verfügung stellen. Einfach jedem genug Geld geben, um davon zu leben. Ohne Bedingungen und ohne Fragen. Generative KI bedroht nicht die Kreativität eines Künstlers, sondern die Art und Weise, wie er sein Einkommen erwirtschaftet. Ich könnte eine KI verwenden, um diesen Artikel mit Bildmaterial zu unterlegen, anstelle meiner üblichen lizenzfreien Stockfotos oder gemeinfreien Bilder, aber ich würde ihn niemals von einer KI schreiben lassen. Ich möchte selbst schreiben. Mein Verleger könnte jedoch beschließen, dass es wirtschaftlicher ist, eine KI mit dem Schreiben ethischer Artikel zu beauftragen, statt mich weiter dafür zu bezahlen. Das ist von geringerer Bedeutung, weil mein Zuhause und meine Gesundheit nicht davon abhängen.

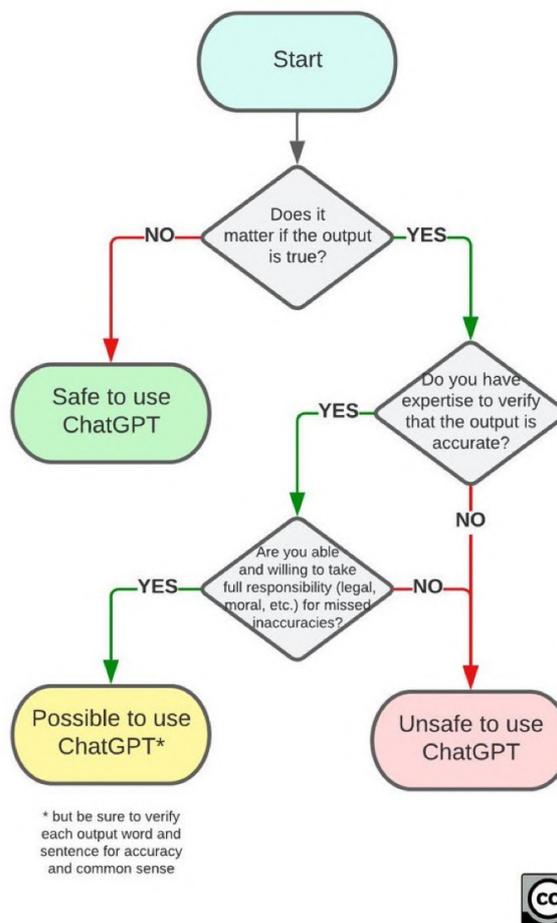
Täuschen Sie sich nicht: Es gibt keinen Beruf, der vor KI sicher ist. Ein Teil der Arbeit bildender Künstler, Schriftsteller, Anwälte, Programmierer und Ingenieure wird in naher Zukunft durch generative KI ersetzt werden. Wir könnten, wenn unser Lebensunterhalt gesichert ist, KI als Werkzeug nutzen und uns frohgemut auf das konzentrieren, wozu wir intrinsisch motiviert sind. Ein Bedingungsloses Grundeinkommen würde uns die Sicherheit geben, die Arbeit zwischen dem, was Menschen am besten können, und dem, was KI besser kann, aufzuteilen.

Wenn wir KI als ein hilfreiches Werkzeug betrachten, wird klar, dass der Moment der Entscheidung uns Menschen gehört. Die Entscheidung, das anzuwenden, was die KI produziert (oder halt nicht), vielleicht die Zielvorgaben weiterzuentwickeln oder die Parameter zu ändern und die KI es erneut versuchen zu lassen. Die Entscheidung darüber, was mit den Ergebnissen der KI geschehen soll, macht sie auch zu unserer Verantwortung. In einem Auto, in einem medizinischen Umfeld und beim Austausch von Informationen mit anderen. ◀

SG — 230048-02

Is it safe to use ChatGPT for your task?

Aleksandr Tiulkanov | January 19, 2023



ChatGPT-Flussdiagramm. Aleksandr Tiulkanov ist ein KI-Forscher.

Über die Autorin

Priscilla Haring-Kuipers schreibt über Technologie aus einer sozialwissenschaftlichen Perspektive. Sie interessiert sich besonders dafür, wie Technologie das Gute im Menschen unterstützen kann und glaubt fest an die Wirkungsforschung. Sie hat einen MSc in Medienpsychologie und schreibt This Is Not Rocket Science.

Das World Ethical Electronics Forum

Das World Ethical Electronics Forum (WEEF) inspiriert globale Innovatoren mit offenen Diskussionen und Veröffentlichungen über Ethik und nachhaltige Entwicklungsziele. Besuchen Sie worldethicalelectronicsforum.com, um sich inspirieren zu lassen und sich zu beteiligen.



WEBLINKS

- [1] OpenAI, DALL-E 2: <https://openai.com/dall-e-2/>
- [2] OpenAI, „ChatGPT: Optimizing Language Models for Dialogue“, 30. November 2022: <https://openai.com/blog/chatgpt/>
- [3] H. Fry, „Apropos Intelligenz“, DeepMind: The Podcast (S2, Ep2), 25. Januar 2022: <https://youtu.be/21JSKHR7KWw>
- [4] B. Perrigo, „OpenAI Used Kenyan Workers on Less Than \$2 Per Hour to Make ChatGPT Less Toxic“, 18. Januar 2023: <https://time.com/6247678/openai-chatgpt-kenya-workers/>
- [5] Stack Overflow, „Temporary Policy: ChatGPT Is Banned“, 5. Dezember 2022: <https://meta.stackoverflow.com/questions/421831/temporary-policy-chatgpt-is-banned>

Der Elektor Store

Nie teuer, immer überraschend!

Der Elektor Store hat sich vom Community-Store für Elektor-eigene Produkte wie Bücher, Zeitschriften, Bausätze und Module zu einem umfassenden Webshop entwickelt, der einen großen Wert auf überraschende Elektronik legt.

Wir bieten die Produkte an, von denen wir selbst begeistert sind oder die wir einfach ausprobieren wollen. Wenn Sie einen Produktvorschlag haben, sind wir hier erreichbar (sale@elektor.de).

Elektor Archiv 1970-2022 (USB-Stick)

5 Elektor-Jahrzehnte (70er, 80er, 90er, 2000er und 2010er Jahre) auf USB-Stick

Dieser USB-Stick (32 GB, USB 3.0) enthält alle Elektor-Ausgaben der Jahrgänge 1970 bis 2022 im PDF-Format. Die Fachzeitschrift Elektor vermittelt ihren Lesern moderne Elektronik und Computertechnik durch die Veröffentlichung nachbausicherer, professionell konzipierter Schaltungen zu allen Bereichen der Elektronik.

Preis: ~~199,95 €~~

Sonderpreis: 99,95 €

 www.elektor.de/20373



Mikrocontroller-Praxiskurs für Arduino-Einsteiger (Bundle)

Dieser Mikrocontroller-Praxiskurs für Einsteiger zeigt, wie man auch ohne große Erfahrung in Elektronik und Programmiersprachen eigene Projekte mit einem Mikrocontroller realisieren kann.

Preis: ~~109,95 €~~

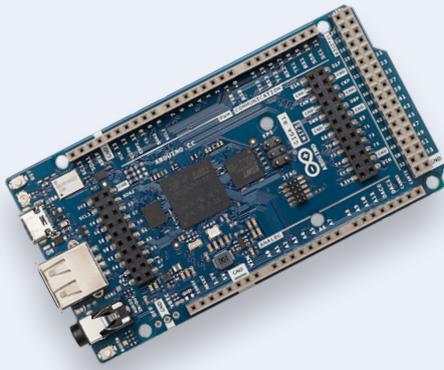
Sonderpreis: 94,95 €



 www.elektor.de/20412



Arduino Giga R1 WiFi



Preis: 89,95 €

Mitgliederpreis: 80,96 €

www.elektor.de/20436

FreeCAD für Elektroniker



Preis: 39,80 €

www.elektor.de/20430

SDRplay RSP1A – 14-Bit SDR-Empfänger (1 kHz bis 2 GHz)



Preis: 139,95 €

Mitgliederpreis: 125,96 €

www.elektor.de/20421

Whadda WTS100 Löt-Einsteigerset



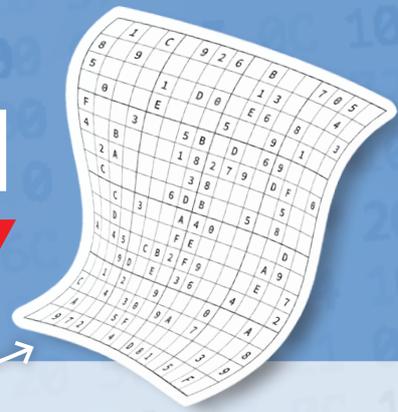
Preis: 49,95 €

Mitgliederpreis: 44,96 €

www.elektor.de/20426

Hexadoku

Sudoku für Elektroniker



Wie in jeder Ausgabe finden Sie auch in diesem Heft unser ganz spezielles Sudoku. PC, Oszilloskop und Lötkolben können sich erholen, während Ihre kleinen grauen Zellen auf Hochtouren arbeiten. Wenn Sie alle Hex-Ziffern in den grauen Kästchen herausgefunden haben, sollten Sie uns diese gleich zumailen – denn hier warten fünf Elektor-Gutscheine!

Die Regeln dieses Rätsels sind ganz einfach zu verstehen: Bei einem Hexadoku werden die Hexadezimalzahlen 0 bis F verwendet, was für Elektroniker und Programmierer ja durchaus passend ist. Füllen Sie das Diagramm mit seinen 16 x 16 Kästchen so aus, dass alle Hexadezimalzahlen von 0 bis F (also 0 bis 9 und A bis F) in jeder Reihe, jeder Spalte und in jedem Fach mit 4 x 4 Kästchen (markiert durch die dickeren schwarzen Linien) **genau einmal** vorkommen. Einige Zahlen sind bereits eingetragen, was die Ausgangssituation des Rätsels bestimmt. Wer das Rätsel löst – sprich die Zahlen in den grauen Kästchen herausfindet – kann einen von fünf Gutscheinen im Wert von 50 Euro gewinnen!



EINSENDEN

Schicken Sie die Lösung (die Zahlen in den grauen Kästchen) per E-Mail oder Post an:

Elektor Redaktion
Lukasstraße 1
52070 Aachen

E-Mail: hexadoku@elektor.de

Als Betreff bitte nur die Ziffern der Lösung angeben!

Einsendeschluss ist der 15. Juni 2023.

DIE GEWINNER DES HEXADOKUS AUS DER AUSGABE MÄRZ/APRIL STEHEN FEST!

Die richtige Lösung ist: **F13AB**.

Aus allen Einsendungen mit der richtigen Lösung haben wir die fünf Gewinner eines Elektor-Wertgutscheins über je 50 € gezogen.

Die Namen der Gewinner werden unter www.elektormagazine.de/hexadoku bekannt gegeben.

Herzlichen Glückwunsch!

E	2		A	0			B		F		C	6	4	9	
	C		F	1	5	8					0		B	A	7
		1				4							8		2
8	3	4				C	D		7	B	E		0	1	
		F	8								3			2	9
7		2	3			1				9	B	D			A
	6				D	8		A	E			5			
	1			5			6		8		D				
	7			A			3		E		6				
	D				6	2		B	3			8			
3		9	6			5				7	F	1			0
		0	5								9			7	6
0	9	E				2	A		3	F	5		C	6	
		7				6								5	3
	8		C	F	7	3					1		D	0	B
5	F		4	8			0		6		2	A	7	E	

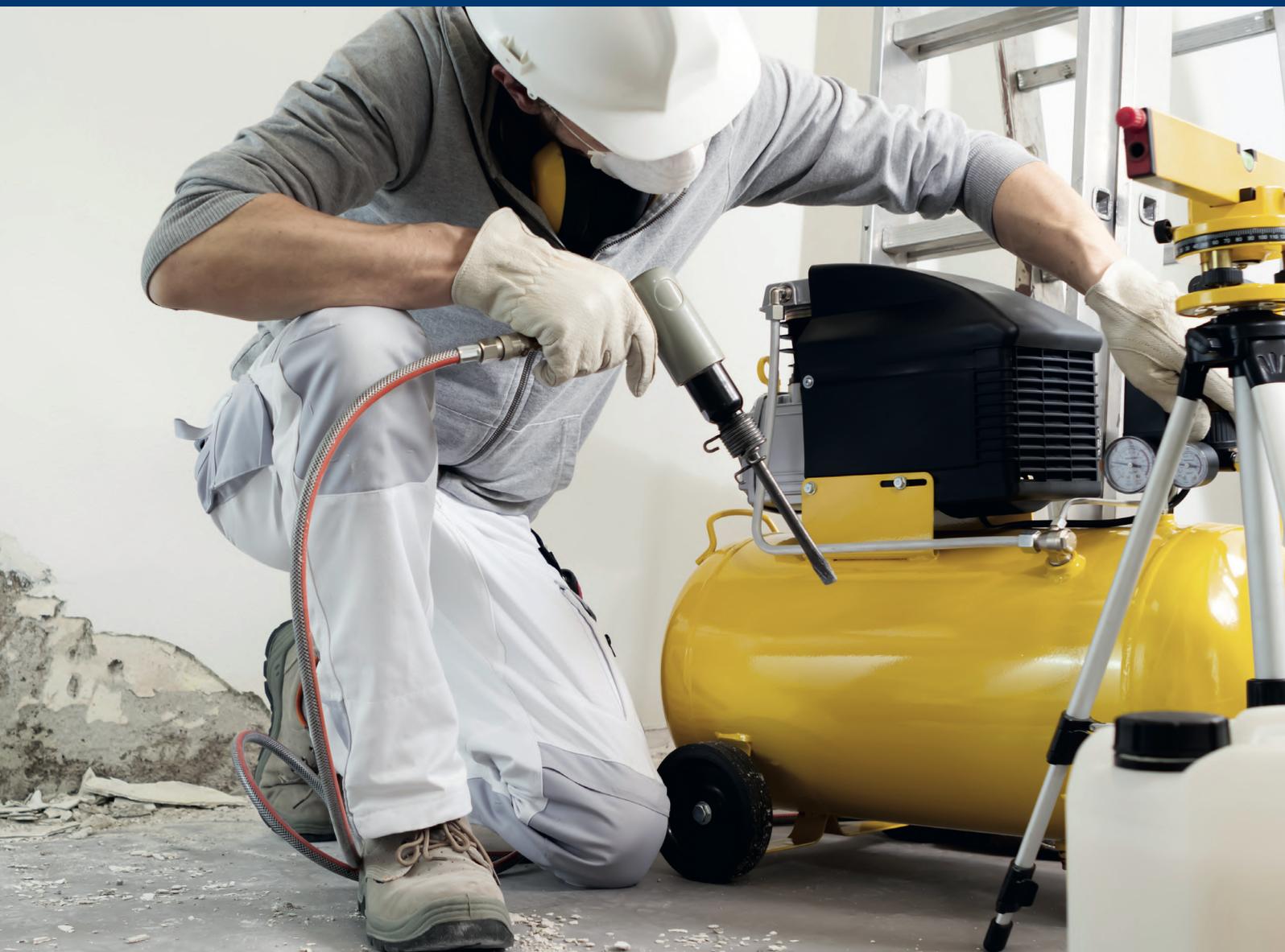
A	1	3	F	B	0	C	D	6	4	8	5	E	2	7	9
0	6	5	2	9	E	F	1	3	A	B	7	4	8	C	D
4	8	C	7	5	2	A	3	F	9	E	D	0	1	6	B
9	B	D	E	4	6	7	8	1	C	0	2	F	3	A	5
F	7	E	8	0	3	D	9	2	B	4	C	A	6	5	1
2	9	0	A	6	1	8	7	5	E	3	F	D	4	B	C
B	D	4	5	C	F	E	A	0	1	9	6	2	7	8	3
1	C	6	3	2	4	5	B	D	7	A	8	9	E	0	F
3	4	1	6	F	8	0	C	7	D	2	B	5	9	E	A
C	E	A	B	7	D	9	2	8	5	F	3	6	0	1	4
5	F	7	D	1	A	6	E	4	0	C	9	3	B	2	8
8	0	2	9	3	5	B	4	A	6	1	E	C	D	F	7
D	2	F	4	A	B	1	6	9	8	5	0	7	C	3	E
E	3	9	C	8	7	2	0	B	F	D	A	1	5	4	6
6	A	B	1	E	9	3	5	C	2	7	4	8	F	D	0
7	5	8	0	D	C	4	F	E	3	6	1	B	A	9	2

Der Rechtsweg ist ausgeschlossen. Mitarbeiter der in der Unternehmensgruppe Elektor International Media B.V. zusammengeschlossenen Verlage und deren Angehörige sind von der Teilnahme ausgeschlossen.

Handwerk und Technik

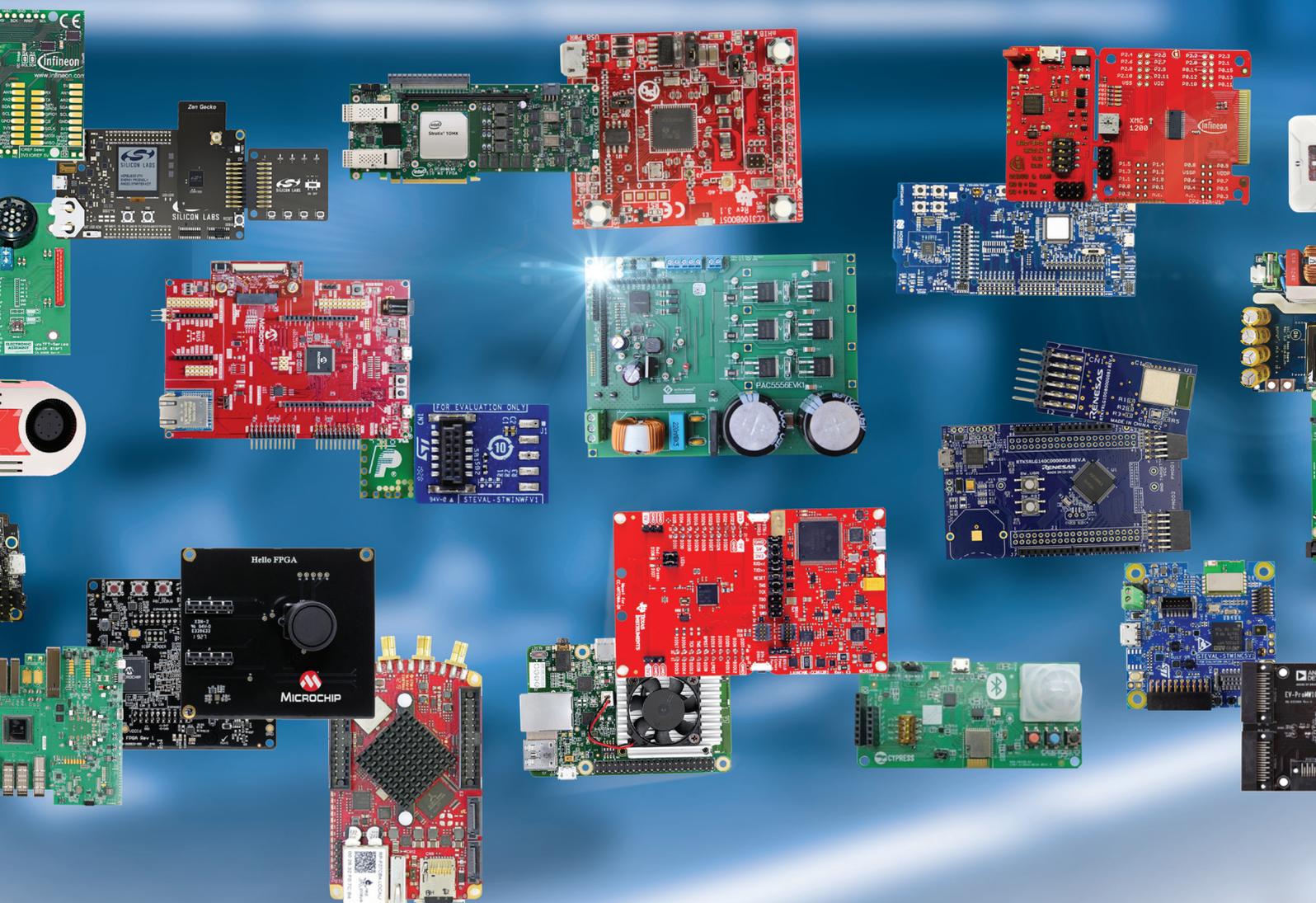
Wir haben alles für Ihren Erfolg:
Produkte, passende Services und persönliche Beratung.
Erfahren Sie mehr online.

conrad.de/handwerk



Development Tools alle an einem Ort

Tausende Tools von hunderten
zuverlässigen Herstellern



Wählen Sie Ihr Produkt aus
unserer breiten Palette auf
[mouser.de/dev-tools](https://www.mouser.de/dev-tools)

