

# Erste Schritte ins IoT – mit dem ESP32-C3

WLAN-Taster  
und -Relais

S. 6



IM FOKUS  
**Internet  
of Things**

S. 54 **Herausforderungen bei der  
Markteinführung von IoT-Lösungen**

Sorgen um Sicherheit, Skalierbarkeit  
und Wettbewerb

S. 12 **IoT-Cloud à la Arduino**

Praktische Lösung für  
Applikationsentwickler

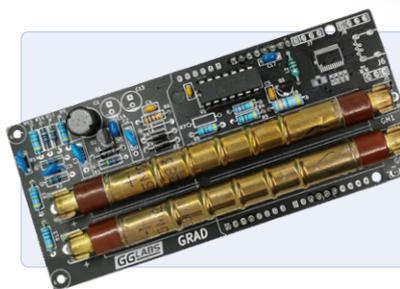
S. 74 **Dragino LPS8  
Indoor Gateway**

Schnell zum eigenen LoRaWAN-Gateway



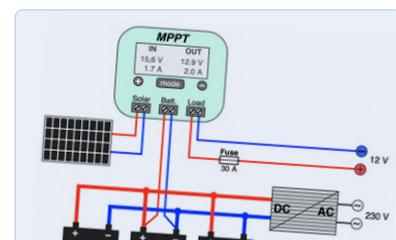
**CO<sub>2</sub>-Wächter**  
Ein DIY-Ansatz zur  
Überwachung der Luftqualität

S. 24



**Arduino-Shield für zwei  
Geiger-Müller-Zählrohre**  
Hochempfindlicher,  
stromsparender Strahlungsmesser

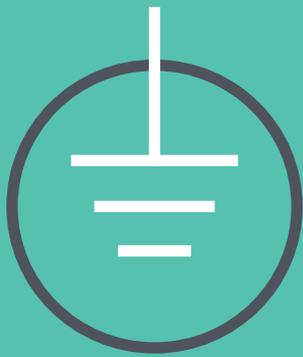
S. 18



**Insel-Solaranlagen**  
Elektrische Energie  
unabhängig vom Netz

S. 104





# UNSERE PREISE DER BESTE SCHUTZ VOR HOHEN KOSTEN

The best part of your project:  
[www.reichelt.de](http://www.reichelt.de)

## Mit reichelt holen Sie mehr aus Ihrem Budget

Dank effizienter, selbstentwickelter Logistik und IT liefern wir Kleinmengen zuverlässig zu Top-Preisen. Als offizieller Reseller von ARDUINO™ sind wir daher bei Entwicklungsprojekten, Instandhaltung und Kleinserienproduktion trotz Versandkosten die bessere Wahl.



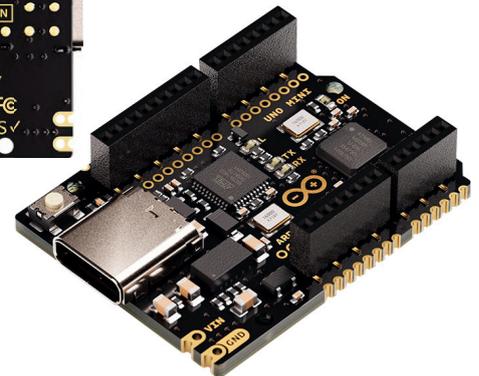
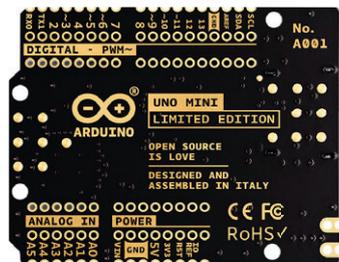
### Arduino UNO Mini

ATmega328P, USB-C

Arduino UNO Mini ist ein 27 x 34 mm Mikrocontroller-Board mit einem ATmega328P Mikrocontroller.

Limited Edition – alles an dieser Version des Arduino UNO ist einzigartig: Schwarz und Gold, elegantes Design, auf der Platine nummeriert und mit einem handschriftlich signierten Brief von den Gründern.

- 14 digitale I/O - Schnittstellen (6 davon als PWM-Ausgang nutzbar)
- 6 analoge Eingänge
- 16 MHz - Quarzoszillator
- USB-C-Anschluss



**LIMITED EDITION**

Bestell-Nr.:  
ARDUINO UNO MINI

**39,50**

Arduino – Leistungsfähige Mikrocontroller für Schalt- und Steueraufgaben

Jetzt entdecken ► <https://rch.it/ard22>



**reichelt**  
elektronik **MAGAZIN**

Arduino im reichelt Magazin:  
Arduino Projekte, Ratgeber und  
How-tos - legen Sie los!

Jetzt entdecken ►  
<https://rch.it/m-arduino>



NEU: KATALOG 06 | 2022

**ONLINE & GEDRUCKT**

- mehr als 120.000 Artikel aus Elektronik & IT
- mehr als 10.000 Neuheiten
- über 2.000 Seiten

Jetzt anfordern ►  
[www.reichelt.de/katalog](http://www.reichelt.de/katalog)



■ Top Preis-Leistungs-Verhältnis

■ über 120.000 ausgesuchte Produkte

■ zuverlässige Lieferung – aus Deutschland in alle Welt

[www.reichelt.de](http://www.reichelt.de)

Bestellhotline: +49 (0)4422 955-333

**reichelt**  
elektronik – The best part of your project

Es gelten die gesetzlichen Widerrufsregelungen. Alle angegebenen Preise in € inklusive der gesetzlichen MwSt., zzgl. Versandkosten für den gesamten Warenkorb. Es gelten ausschließlich unsere AGB (unter [www.reichelt.de/agb](http://www.reichelt.de/agb), im Katalog oder auf Anforderung). Abbildungen ähnlich. Druckfehler, Irrtümer und Preisänderungen vorbehalten. reichelt elektronik GmbH & Co. KG, Elektronikring 1, 26452 Sande, Tel.: +49 (0)4422 955-333

TAGESPREISE! Preisstand: 7.4. 2022

### Verlag

Elektor Verlag GmbH  
 Kackertstraße 10  
 52072 Aachen  
 Tel. 0241 95509190

Technische Fragen bitten wir per E-Mail an [redaktion@elektor.de](mailto:redaktion@elektor.de) zu richten.

### Hauptsitz des Verlags

Elektor International Media  
 Postbus 11, 6114 ZG Susteren  
 Niederlande

### Anzeigen

Raoul Morreau (Leitung)  
 Mobil: +31 6 440 399 07  
 E-Mail: [raoul.morreau@elektor.com](mailto:raoul.morreau@elektor.com)

Büsra Kas

Tel. 0241 95509178  
 E-Mail: [busra.kas@elektor.com](mailto:busra.kas@elektor.com)

Es gilt die Anzeigenpreisliste ab 01.01.2022.

### Distribution

IPS Pressevertrieb GmbH  
 Postfach 12 11, 53334 Meckenheim  
 Tel. 02225 88010  
 Fax 02225 880199

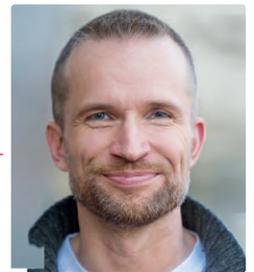
### Druck

Senefelder Misset, Doetinchem (NL)

Der Herausgeber ist nicht verpflichtet, unverlangt eingesandte Manuskripte oder Geräte zurückzusenden. Auch wird für diese Gegenstände keine Haftung übernommen. Nimmt der Herausgeber einen Beitrag zur Veröffentlichung an, so erwirbt er gleichzeitig das Nachdruckrecht für alle ausländischen Ausgaben inklusive Lizenzen. 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.

## von Jens Nickel

Chefredakteur *ElektorMag*



# Aus Fehlern lernen

Ein Editorial zu schreiben heißt, seiner Zeit immer ein paar Wochen voraus zu sein. Wenn Sie diese Zeilen lesen, gibt es - so hoffe ich sehr - schon die ersten Anzeichen von Frieden in der Ukraine. Und vielleicht sind auch die Diskussionen abgeebbt, wie viele Corona-Maßnahmen wir noch brauchen, um den hohen Inzidenzen Herr zu werden.

Was haben die (meist schlechten) Nachrichten aus aller Welt mit Elektronik zu tun? Wir alle mussten in den letzten Monaten erkennen, dass wir unsere Welt hochvernetzt und globalisiert haben, und zwar in gutem Glauben, dass stetig Frieden herrscht und große Katastrophen ausbleiben. Viele Jahre ging das ja auch gut! Selbst Großkonzerne wie die internationalen Autobauer hatten sich schon so an die komfortable Lage gewöhnt, dass für den Fall der Fälle kein Plan B in der Schublade war.

Wenn die Krisen überwunden sind, werden viele Unternehmen die gleichen Fehler sicher nicht noch einmal machen. Hohe Verfügbarkeit und kürzere Transportwege, so etwas bekommt einen höheren Stellenwert gegenüber immer nur dem günstigsten Preis. Dies wird auch der Elektronikbranche guttun.

In vielen Bereichen der Elektronik ist der stetige Aufwind abgeschwächt worden, doch es geht weiter vorwärts, so zum Beispiel beim Internet of Things - dem Schwerpunktthema dieser Ausgabe. Dazu hat mein Kollege Robert van der Zwan interessante Infografiken zusammengestellt (Seite 60). Natürlich haben wir auch wieder Hintergrundartikel und Projekte zu bieten! Wir bleiben dran.



### Abschied, kein Vergessen

Dass unser langjähriger Mitarbeiter Ralf Schmiedel im März verstorben ist, war für mich und meine Kollegen ein Schock. Seit 2014 betreute Ralf den Leserservice von Elektor. Der Ingenieur kümmerte sich mit großer Geduld und vielen eigenen Ideen um die unterschiedlichsten Anliegen von Lesern und Autoren, stellte Korrekturen von Projekten zusammen und half unserer Redaktion bei weiteren Aufgaben wie den Jahres-DVDs. Viele Dankesmails von Lesern zeigten, wie wertvoll seine Unterstützung war. Ralf, wir alle werden dich in guter Erinnerung behalten!

## — Unser Team —



Chefredakteur:	Jens Nickel (v.i.S.d.P.) ( <a href="mailto:redaktion@elektor.de">redaktion@elektor.de</a> )
Redaktion:	Eric Bogers, Jan Buiting, Stuart Cording, Rolf Gerstendorf, Dr. Thomas Scherer, Clemens Valens
Leserservice:	Ralf Schmiedel
Elektor-Labor:	Mathias Claußen, Ton Giesberts, Luc Lemmens, Clemens Valens
Grafik & Layout:	Giel Dols, Harmen Heida
Herausgeber:	Erik Jansen



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



Elektor ist Mitglied von FIPP, einer Organisation, die „über fast 100 Jahre gewachsen ist und Medienbesitzer und Content-Ersteller aus der ganzen Welt umfasst“.

## Erste Schritte ins IoT - mit dem ESP32-C3

WLAN-Taster und -Relais



## Rubriken

- 3 **Impressum**
- 36 **Aller Anfang ...**  
Willkommen bei der Diode!
- 40 **Von Entwicklern für Entwickler**  
Tipps & Tricks beim Bauteiltest
- 68 **Bemerkenswerte Bauteile**  
Wanderfeldröhren
- 82 **Projekt 2.0**  
Korrekturen, Updates und Leserbriefe
- 111 **Ethics**  
Das 10-Jahres-Handy
- 114 **Hexadoku**  
Sudoku für Elektroniker

## Hintergrund

- FOCUS**  
12 **IoT-Cloud à la Arduino**  
Erster Kontakt mit der Arduino-Cloud
- 32 **MonkMakes:**  
**Luftqualitäts-Messgerätekit für den Raspberry Pi**  
Misst Temperatur und CO<sub>2</sub>e

- FOCUS**  
69 **NB-IoT - ein Überblick**  
Standards, Abdeckung, Verträge und Module
- FOCUS**  
74 **Dragino LPS8 Indoor Gateway**  
Schnell zum eigenen LoRaWAN-Gateway
- 86 **Schaltungssimulation  
mit TINA Design Suite und TINACloud**  
Beispielkapitel: Sinusförmige Oszillatoren
- 92 **WinUI 3: Neues Grafik-Framework für Windows-Apps**  
Eine erste App für Elektroniker
- 100 **GUIs mit Python**  
Das schlechteste GUI der Welt
- 104 **Insel-Solaranlagen**  
Elektrische Energie unabhängig vom Netz

## Industry

- FOCUS**  
54 **Herausforderungen bei  
der Markteinführung von IoT-Lösungen**  
Sorgen um Sicherheit, Skalierbarkeit und Wettbewerb
- FOCUS**  
60 **Infografik**  
Fakten und Zahlen zum Internet of Things

**Arduino-Shield für zwei  
Geiger-Müller-Zählrohre**  
Hochempfindlicher,  
stromsparender Strahlungsmesser





## Tipps & Tricks beim Bauteiltest

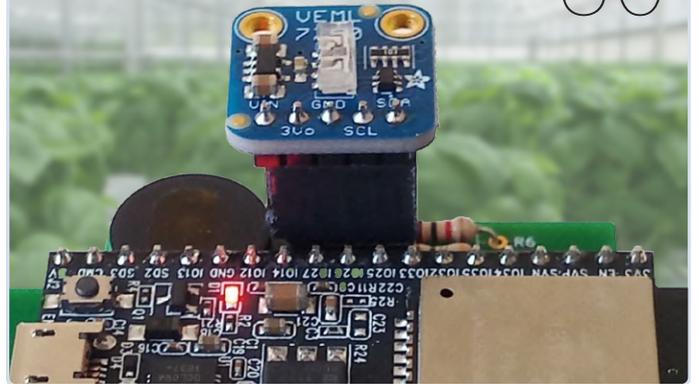
Recycling ohne teures Messequipment

40

## Lichtschalter DeLux

Lösung für hochpräzises  
lichtgesteuertes Schalten

50



### 62 Lieber doch verkabelt

Tipps zur Entwicklung einer 1-Gbit/s-Schnittstelle  
im Industrie-Umfeld

### 66 Edge Impulse

## Vorschau

### Elektor Juli/August 2022

Das nächste Heft ist wie immer randvoll gefüllt mit Schaltungsprojekten, Grundlagen sowie Tipps und Tricks für Elektroniker.

## Projekte

#### FOCUS

### 6 Erste Schritte ins IoT - mit dem ESP32-C3

WLAN-Taster und -Relais

### 18 Arduino-Shield für zwei Geiger-Müller-Zählrohre

Hochempfindlicher, stromsparender Strahlungsmesser

#### FOCUS

### 24 CO<sub>2</sub>-Wächter

DIY-Ansatz zur Überwachung der Luftqualität

### 46 Stromsparende Maulwurfscheuche

ATtiny13 ersetzt den 555

### 50 Lichtschalter DeLux

Lösung für hochpräzises lichtgesteuertes Schalten

### 77 ATtiny-Mikrocontroller mit C und Assembler erforschen

Beispielkapitel: I/O-Ports des ATtiny

#### FOCUS

### 83 LoRa-GPS-Tracker - ein Update

Daten abholen und visualisieren mit dem Raspberry Pi

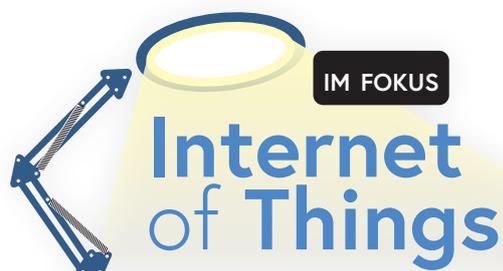
#### Aus dem Inhalt:

- › Autonomes Induktivitäts-Messgerät
- › CO<sub>2</sub>-Meter mit Sigfox
- › Design-Tools für analoge Filter
- › Smarte Steckdosen: Ein neugieriger Blick ins Innere
- › Einfaches analoges ESR-Meter
- › Richtig messen mit dem Oszilloskop
- › Raspberry Pi Pico als MSF-SDR
- › Netzspannungs-Frequenzmeter

#### Und vieles mehr!

Elektor Juli/August 2022 erscheint am 7. Juli 2022.

Änderungen vorbehalten.



# Erste Schritte ins **IoT** - mit dem **ESP32-C3**

## WLAN-Taster und -Relais



Von Mathias Claußen (Elektor)

Das IoT ist kein Buch mit sieben Siegeln. Clevere Controller wie der neue ESP32-C3 und die einsteigerfreundliche Arduino-IDE machen das Entwickeln von kleinen Projekten einfach.

Wer vom Internet of Things (IoT) redet, meint damit, dass immer mehr Dinge unseres täglichen Lebens an das Internet angeschlossen werden. Das beginnt mit Leuchten, Heizungen und Sensoren in der Wohnung und setzt sich fort über Autos, Verkehrsampeln, Seecontainer und vieles mehr. In jedem der angeschlossenen Dinge sind kleine netzwerkfähige Komponenten verbaut, die den Austausch von Daten ermöglichen. Wie man eigene Elektronik mit dem IoT verbinden kann, lernt man am besten anhand eines praktischen Beispiels. In diesem Artikel soll ein WLAN-fähiger Taster und ein WLAN-fähiges Relais entstehen; das Relais lässt sich mit dem Taster fernschalten und meldet seinen Status zurück.

### Zutatenauswahl

Wie bei allen Projekten müssen zuerst passende Zutaten ausgewählt werden. Hier kommt das ESP-C3-12F-Kit (**Bild 1**) aus dem Elektor-Shop zum Einsatz. Auf diesem Board sitzt ein WLAN-fähiger Mikrocontroller ESP32-C3 von Espressif. Der ESP32-C3 ist als Ersatz für den bewährten ESP8266 angetreten, neben einem modernen CPU-Kern bietet der Chip eine gute Mischung aus integrierter Peripherie, die sowohl einsteigerfreundlich als auch leistungsfähig ist (siehe unser Review des ESP32-C3 [1]). Eine Übersicht der integrierten Hardwareblöcke ist in **Bild 2** zu sehen. Neben dem ESP32-C3 sind auch eine RGB-LED und ein USB-Seriell-Wandler auf dem Board integriert. Für unser Projekt



Bild 1. ESP32-C3-12F Kit.

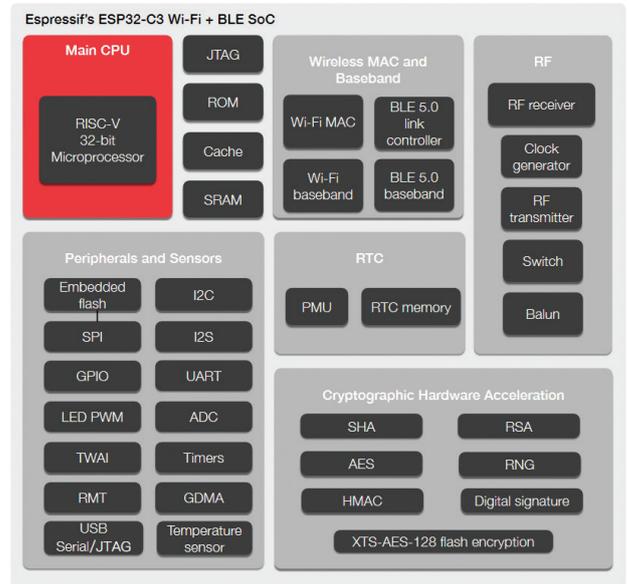


Bild 2. ESP32-C3 Funktionsblöcke (Quelle: ESP32-C3 Datenblatt).

brauchen wir natürlich zwei ESP-C3-12F-Kits. Neben den ESP-C3-12F-Kits werden auch noch ein Sensor und ein Aktor benötigt. Hier kommt das 37-in-1-Sensor-Kit von Elektor zum Einsatz, in dem 35 Sensoren enthalten sind (von den einst 37 Sensoren wurden zwei quecksilberhaltige Typen inzwischen aus Sicherheitsgründen aus dem Kit entfernt). Eine Übersicht der Sensoren des Kits (**Bild 3**) kann man **Bild 4** und dem Infodokument [2] entnehmen. Wir benötigen zuerst einmal den Jostick, der hier als Taster verwendet wird, und das Relais, das als Aktor zum Einsatz kommt. Für das Verbinden der Module werden noch ein paar Steckbrücken (weiblich/weiblich) benötigt. Diese sind unter anderem im Maker-Kit „Mini Breadboards & Jumper“ von Pimoroni (siehe Kasten **Passende Produkte**) enthalten. Hinzu kommt noch ein Rechner, zum Beispiel ein Raspberry Pi, der später als Server agiert, über den die IoT-Geräte ihre Daten austauschen können. Ein ausrangierter Raspberry Pi 1 ist hierfür ausreichend, mindestens ein Raspberry Pi 2 wird aber empfohlen. Wer auf der Suche nach einem kleinen, aber leistungsfähigen Raspberry Pi ist, sollte sich einmal das *Raspberry Pi Zero 2 W Bundle* (Kasten) ansehen. Es muss

aber kein Raspberry Pi sein, ein PC mit einer Linux-Distribution wie Ubuntu [3] ist auch vollkommen ausreichend. Bevor wir mit dem eigentlichen Projekt starten, wollen wir uns noch ein klein wenig damit beschäftigen, wie wir Daten übertragen können.

### MQTT

IoT-Geräte, ob Sensor oder Aktor, müssen Daten austauschen. Hierzu können entweder proprietäre Protokolle zum Einsatz kommen, bei dem jeder Entwickler das Rad komplett neu erfindet, oder es werden standardisierte Protokolle verwendet. Ein standardisiertes Verfahren, das große Verbreitung gefunden hat, ist MQTT. Ursprünglich war das die Abkürzung von *Message Queuing Telemetry Transport*, doch 2013 wurde offiziell beschlossen, dass MQTT nur noch für sich steht [4]. MQTT ist ein Protokoll, das für den Nachrichtenaustausch durch einen Broker (Server) sorgt, ohne dabei festzulegen, wie die Nachrichten aussehen müssen. Man kann das mit dem Versand eines Briefes vergleichen: Die Logistik und das Format des Umschlages wird durch das Postunternehmen vorgegeben, was aber in dem Brief steht und in welcher Sprache dieser geschrieben ist, kann jeder selbst entscheiden.



Bild 3. Elektor 37-in-1 Sensorkit.

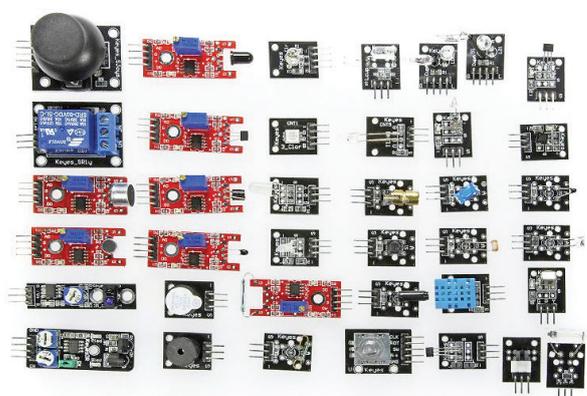


Bild 4. Übersicht der enthaltenen Sensoren.

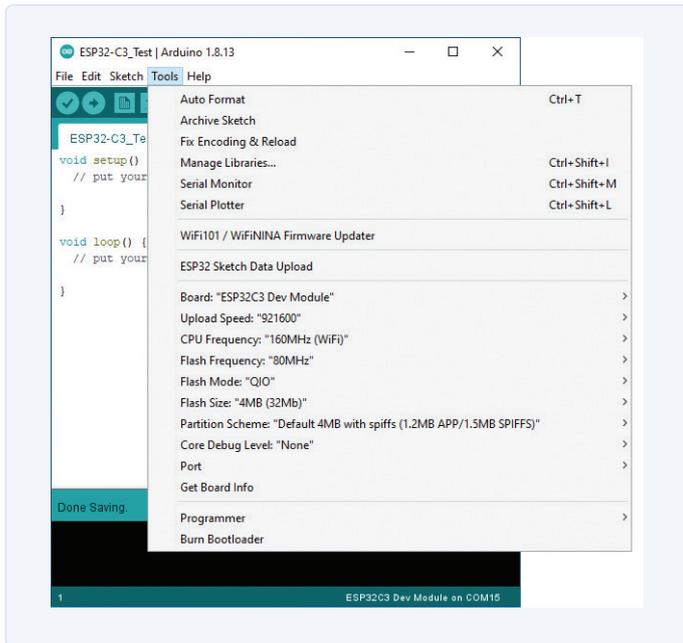


Bild 5. Einstellungen für den ESP32-C3.

Doch welche „Sprache“ soll nun für unsere Nachrichten verwendet werden? Auch hier gibt es mehrere Optionen. Eine, die nicht nur für MQTT genutzt wird, ist JSON.

## JSON – Java Script Object Notation

JSON ist ein leichtgewichtiges Format für den Datenaustausch, auch kleinere Mikrocontroller können solche Nachrichten leicht generieren und aufschlüsseln. Hinzu kommt, dass JSON auch für Menschen nicht nur einfach zu lesen, sondern auch einfach zu schreiben ist. Eine Übersicht über die JSON-Spezifikation findet man auf der Webseite des JSON-Standards [5]. JSON wird nicht nur in Verbindung mit MQTT verwendet, sondern auch in vielen anderen Bereichen. JavaScript, die Programmiersprache, von der JSON abgeleitet

wurde, ist eine der Kerntechnologien, auf denen heute das World Wide Web basiert. Eine gute Einführung in JSON mit praktischen Beispielen findet man auf der Site von Mozilla [6].

## Vorbereitung der IoT-Umgebung: Der MQTT-Broker

Wie bei allen Projekten sorgt eine passende Vorbereitung später für weniger Überraschungen. MQTT benötigt einen Broker, der auf einem alten PC oder einem Raspberry Pi installiert werden kann, um Nachrichten auszutauschen. Mit Node-RED steht eine kompletter Werkzeugkoffer für das Entwickeln von Netzwerk-Anwendungen bereit, der nicht nur auf das Verarbeiten von MQTT-Nachrichten beschränkt ist. Node-RED [7] wurde bei Elektor schon häufig eingesetzt, um MQTT-Daten zu verarbeiten, und lässt sich dank der Anleitungen auf dem Raspberry Pi [8] oder dem PC [9] zügig installieren.

## Die Arduino-IDE

Als Entwicklungsumgebung kommt die Arduino-IDE zum Einsatz. Der Editor der IDE ist nicht der beste in seiner Klasse, bietet aber momentan die stabilste Unterstützung für den ESP32-C3. Die Arduino-IDE [10] kann kostenfrei von der Arduino-Homepage heruntergeladen und installiert werden. Anschließend muss die Arduino-ESP32-Unterstützung (so wie in der Espressif-Dokumentation [11] beschrieben) installiert werden. Die Einstellungen für das Board sind wie in **Bild 5** vorzunehmen.

Neben der Arduino-ESP32-Unterstützung werden für die ersten Schritte auch noch ein paar Bibliotheken benötigt. Damit der ESP32-C3 Daten per MQTT/JSON versenden kann, wird der *PubSubClient* von Nick O’Leary benötigt und die *ArduinoJson*-Bibliothek von Benoit Blanchon. Diese lassen sich über den Bibliotheks-Manager der Arduino-IDE installieren (**Bild 6** und **Bild 7**).

## Zusammenbau der Hardware

Die beiden ESP32-C3-basierten Module werden gemäß den Schaltplänen in **Bild 8** und **Bild 9** zusammengesetzt. Für den Taster und das Relais sind jeweils nur drei Verbindungen herzustellen, wobei zu beachten ist, dass das Relais aus 5 V gespeist wird. Ist alles zusammengesetzt, sieht der Aufbau wie in **Bild 10** aus.

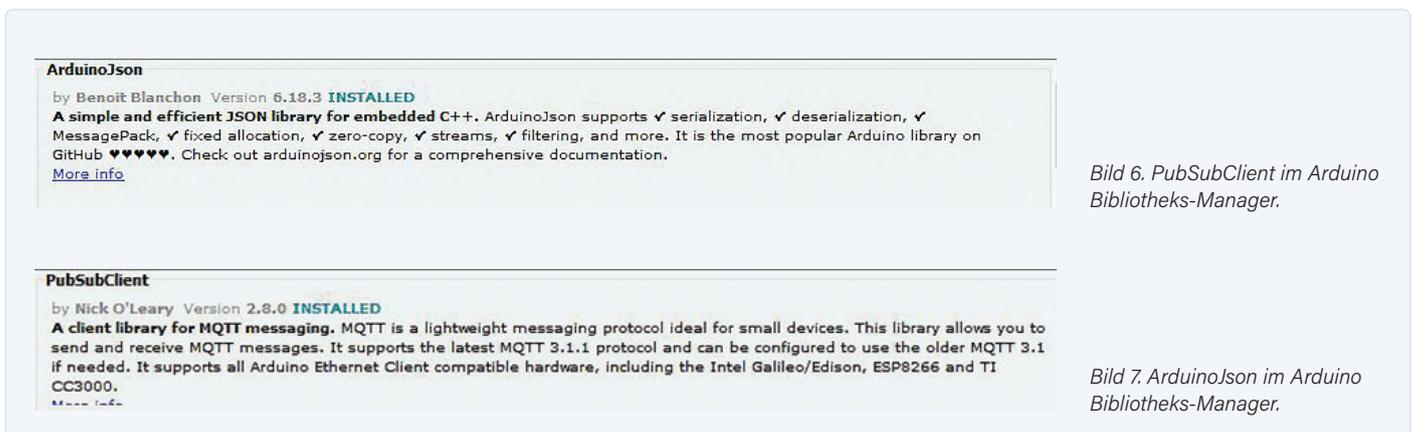


Bild 6. PubSubClient im Arduino Bibliotheks-Manager.

Bild 7. ArduinoJson im Arduino Bibliotheks-Manager.

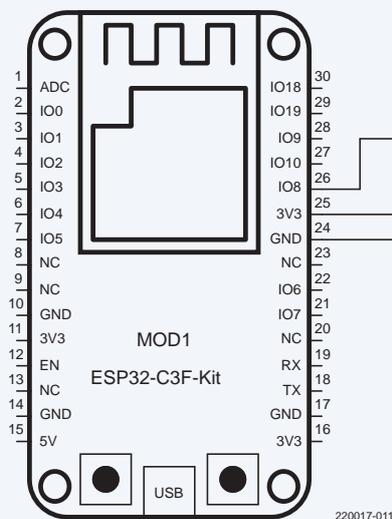


Bild 8. Schaltplan ESP32-C3 und Joystick.

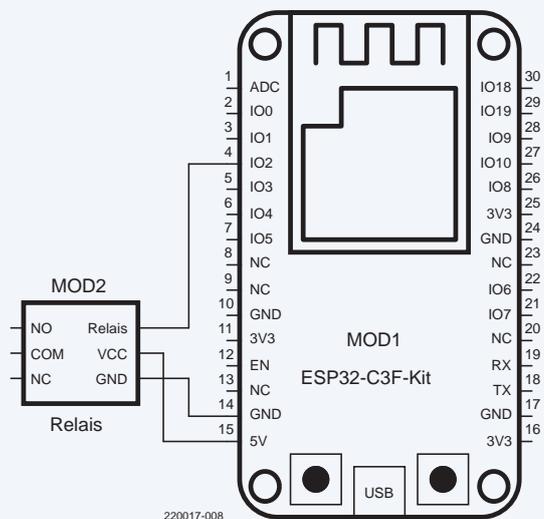


Bild 9. Schaltplan ESP32-C3 und Relais.

## Einrichtung der Software

Auch bei diesem Projekt steht der Quelltext wieder auf GitHub [12] bereit. Von dort können die Sketche für die beiden ESP32-C3-Controller heruntergeladen werden. Bevor diese jedoch einfach auf die ESP32s hochgeladen werden können, bedarf es noch etwas Konfiguration. Damit die beiden Controller Daten mit einem MQTT-Broker austauschen können, muss alles passend eingestellt werden. Dazu sind in den beiden Arduino-Sketches am Anfang `#defines` vorhanden.

```
#define WIFI_SSID "changeme"
#define WIFI_PASS "changeme"
#define MQTT_SERVER "test.mosquitto.org"
```

Diese drei `#defines` am Anfang des Arduino-Sketches müssen für das eigene Netzwerk angepasst werden. Man beginnt mit der WLAN-SSID

und dem Passwort. Für den MQTT-Server wird die IP-Adresse des Node-RED-Computers im eigenen Netzwerk angegeben. Sind beide Sketche (sowohl für das Relais als auch für den Taster) vorbereitet, dann können diese auf den jeweiligen ESP32-C3 hochgeladen werden. Beide ESP32s sollten danach starten und die große LED auf den Boards jeweils anfangen, weiß zu blinken. Dies zeigt an, dass der ESP32-C3 versucht, sich mit dem WLAN zu verbinden. Ist die Verbindung hergestellt, dann leuchtet die LED dauerhaft. Die Farbe hängt von der Funktion des Boards ab: Weiß leuchtet die LED auf der Platine, an die das Relais angeschlossen ist. Bei der Platine, an die der Taster angeschlossen ist, leuchtet die LED rot (Relais aus) oder grün (Relais an), also je nach Zustand des Relais.

Damit sind beide ESP32-C3 erfolgreich in Betrieb genommen. Ein Druck auf den Taster wird nun für eine Zustandsänderung des Relais sorgen; die Farbe der LED wird von Rot nach Grün oder Grün nach

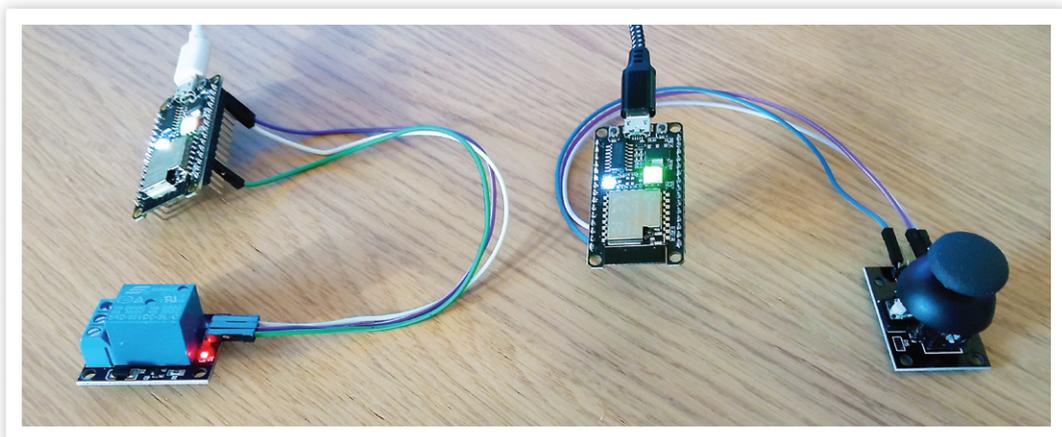


Bild 10. Fertiger Hardwareaufbau.

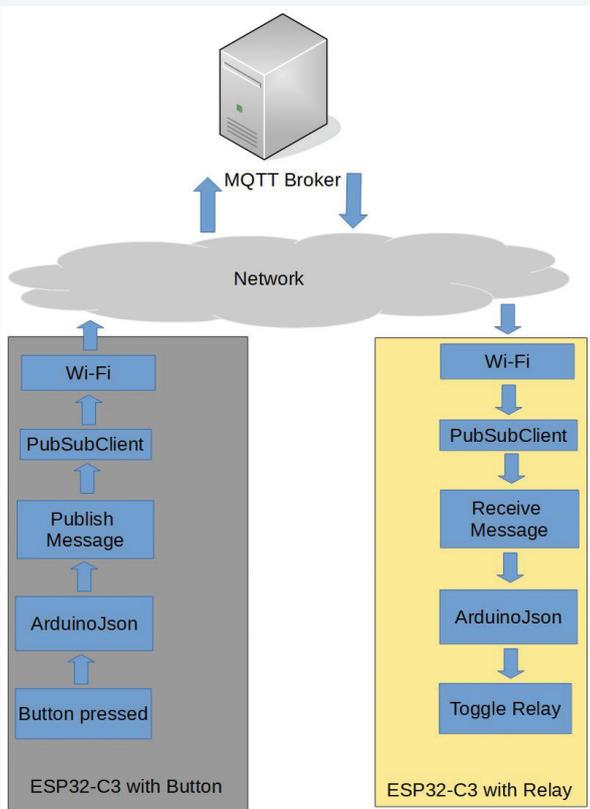


Bild 11. Datentransport zum Broker und zurück.

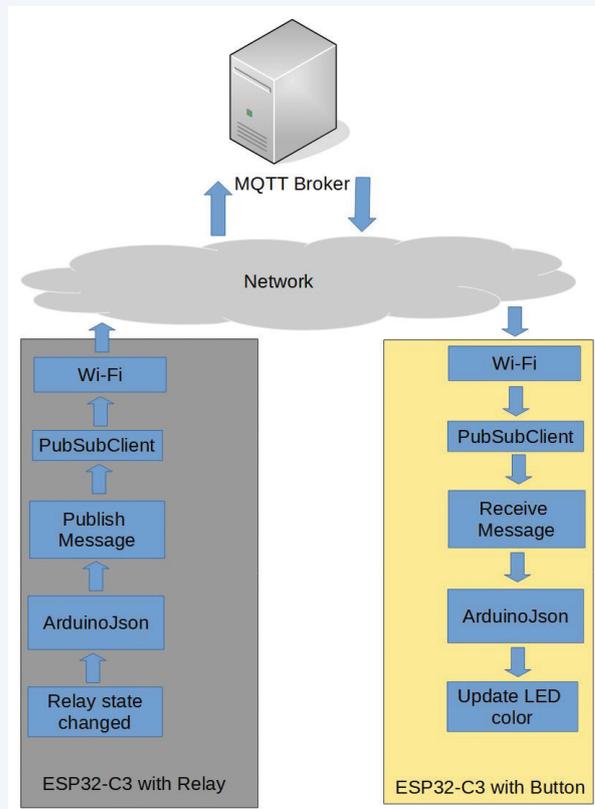


Bild 12. Feedback vom Relais.

Rot wechseln. Der Taster des einen ESP32-C3 kann so erfolgreich das Relais des anderen ESP32-C3 steuern und bekommt den Zustand des Relais auch zurückgemeldet. Die erste eigene IoT-Anwendung läuft! Doch wie funktioniert der Datenaustausch?

### Einmal Relais und zurück

Zuerst wollen wir uns den Weg ansehen, den ein Tastendruck bis zum Relais nimmt. In **Bild 11** ist zu sehen, wie die Nachricht Schicht für Schicht verpackt und anschließend per WLAN an den Broker gesendet wird. Im Quelltext geschieht dies durch `client.publish(MQTT_TOPIC_OUT, (const uint8_t*)buffer, n, true);`.

Doch warum heißt die Funktion `publish` und nicht einfach `send`? Dies liegt an der Art, wie bei MQTT die Daten später verteilt werden. Auf einem MQTT-Broker werden Nachrichten anhand eines Topics (Thema) verteilt; in diesem Fall ist das Topic (`MQTT_TOPIC_OUT`) „*BUTTON*“. Beim Verbinden mit dem MQTT-Broker kann der Client, also der ESP32-C3 unseres Relais, mitteilen, welche Themen für ihn interessant sind, diesen Nachrichtenkanal also „abonnieren“.

Jeder Teilnehmer, der dem MQTT-Broker mitgeteilt hat, dass er sich für ein bestimmtes Thema interessiert, bekommt die unter diesem Thema gesendeten Nachrichten zugestellt. Der Sender andererseits muss sich nicht um die Verteilung kümmern, sondern seine Nachrichten

nur an den Broker senden (sie „veröffentlichen“).

Der ESP32-C3 mit dem Relais abonniert das Thema *BUTTON* in seinem Code mit `client.subscribe(MQTT_TOPIC_IN);`, wobei hier `MQTT_TOPIC_IN` „*BUTTON*“ ist. Jedes Mal, wenn nun durch den Taster eine Nachricht ausgelöst wird, geht diese zum MQTT-Broker. Sie wird dann dem ESP32-C3 mit dem Relais zugestellt, was das Relais zum Umschalten bringt.

Wenn das Relais seinen Zustand ändert, wird auch eine Nachricht an den MQTT-Broker unter dem Topic „*RELAIS*“ gesendet, in dem der nun aktuelle Zustand (ob an oder aus) festgehalten ist. Der ESP32-C3 mit Taster wiederum hat beim MQTT-Broker das Topic „*RELAIS*“ abonniert und bekommt damit, wie in **Bild 12** zu sehen, die Nachricht mit dem neuen Zustand zugestellt. Dies bewirkt, dass die Farbe der LED auf Rot oder Grün gesetzt wird.

Das Schöne an diesem Aufbau ist, dass sich so auch ein zweiter Schalter integrieren lässt, in dem dieser wie der erste Schalter Nachrichten unter dem Topic „*BUTTON*“ sendet. Der ESP32-C3 mit dem Relais wird dann die Daten von beiden Tastern entgegennehmen und entsprechende Schalthandlungen durchführen. Sollten Sie Lust bekommen haben, weiter mit MQTT zu experimentieren, dann schauen Sie doch auch einmal die anderen Projekte von Elektor an, bei denen per MQTT Daten versendet werden. Beispiele sind die Wetterstation [13] oder

die Monster-LED-Uhr mit externem Temperatursensor [14]. Mehr zu MQTT und wie man damit zum Beispiel Daten an Cloud-Plattformen senden kann, findet man auch in der Serie „Mein Weg in das IoT“ [15].

## Zusammenfassung

IoT muss nicht kompliziert sein: Mit einfachen Mitteln und dem Wissen um die passenden Zutaten lassen sich schnell IoT-Geräte entwickeln. Diese können komplexer sein als nur ein Taster oder ein Relais. Von der Heizungssteuerung bis zur Türklingel: Der Fantasie sind keine Grenzen gesetzt. Ein ESP32-C3 ist in jedem Fall eine preiswerte Basis für eigene Versuche! ◀

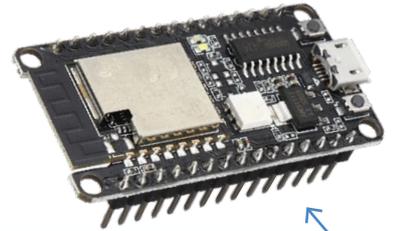
220017-02

### Ein Beitrag von

Entwurf und Text: **Mathias Claußen**  
Redaktion: **Jens Nickel**  
Layout: **Giel Dols**

### Sie haben Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an den Autor unter [mathias.claussen@elektor.com](mailto:mathias.claussen@elektor.com) oder kontaktieren Sie Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).



## PASSENDE PRODUKTE

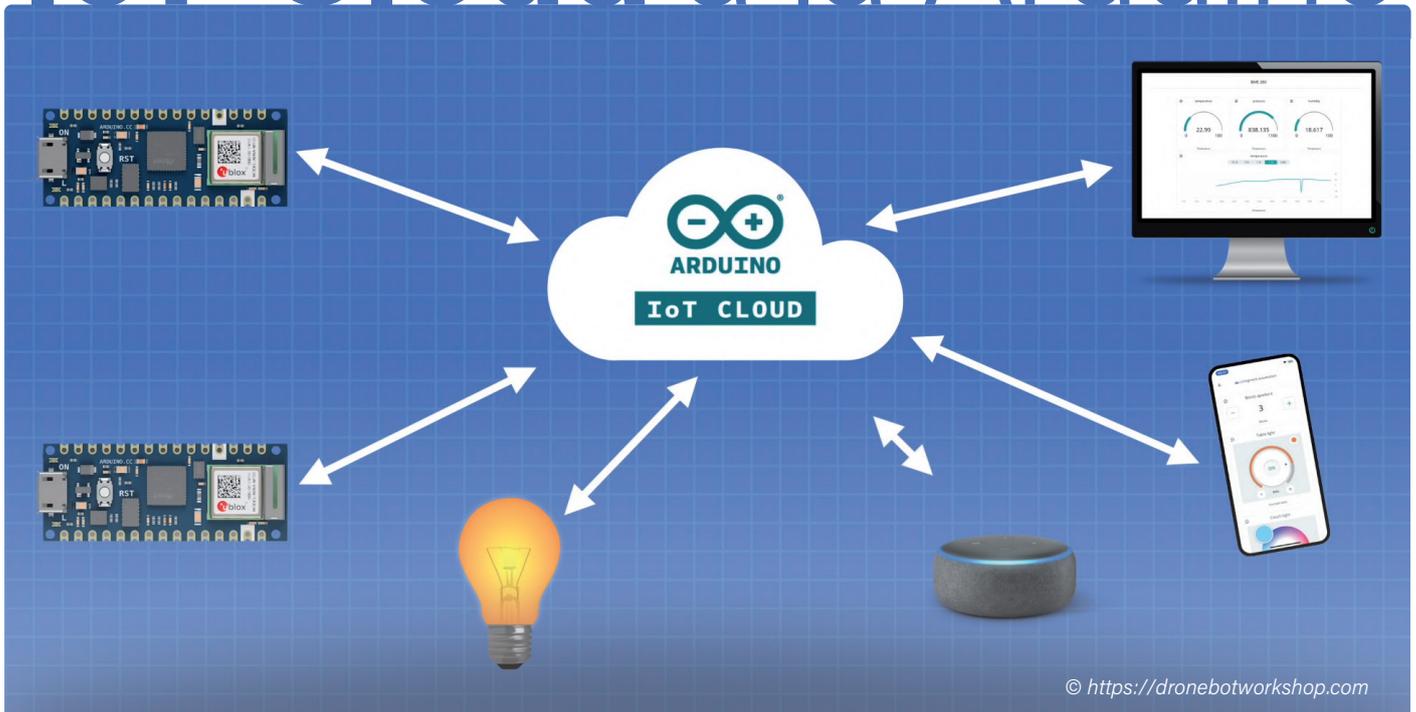
- **ESP-C3-12F-Kit Development-Board mit 4 MB Flash**  
SKU 19855  
[www.elektor.de/19855](http://www.elektor.de/19855)
- **Elektor 37-in-1 Sensorkit** SKU 16843  
[www.elektor.de/16843](http://www.elektor.de/16843)
- **Raspberry Pi Zero 2 W Bundle** SKU 19952  
[www.elektor.de/19952](http://www.elektor.de/19952)
- **Pimoroni Maker Essentials - Mini-Breadboards und Jumperkabel** SKU 18430  
[www.elektor.de/18430](http://www.elektor.de/18430)



## WEBLINKS

- [1] Mathias Claußen, „ESP32-C3: 32-Bit-RISC-Einkerner“, Elektormagazine.de: [www.elektormagazine.de/news/esp32c3-32bitrisc-einkerner](http://www.elektormagazine.de/news/esp32c3-32bitrisc-einkerner)
- [2] Elektor 37-in-1 Sensor-Kit Dokumentation: [www.elektor.com/amfile/file/download/file/1170/product/6171/](http://www.elektor.com/amfile/file/download/file/1170/product/6171/)
- [3] Ubuntu Linux Distribution: <https://ubuntu.com/>
- [4] OASIS MQTT TC Minutes vom 25.04.2013:  
[www.oasis-open.org/committees/download.php/49028/OASIS\\_MQTT\\_TC\\_minutes\\_25042013.pdf](http://www.oasis-open.org/committees/download.php/49028/OASIS_MQTT_TC_minutes_25042013.pdf)
- [5] JSON.org: [www.json.org/json-de.html](http://www.json.org/json-de.html)
- [6] Mozilla Web Docs: Working with JSON: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>
- [7] Node-RED: <https://nodered.org/>
- [8] Node-RED Installation auf dem Raspberry Pi: <https://nodered.org/docs/getting-started/raspberrypi>
- [9] Node-RED Installation auf dem PC: <https://nodered.org/docs/getting-started/local>
- [10] Arduino-IDE Download: [www.arduino.cc/en/software](http://www.arduino.cc/en/software)
- [11] Espressif Arduino-ESP32 Installationsanleitung: <https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html>
- [12] Elektor GitHub Repository: [https://github.com/ElektorLabs/220017-ESP32-C3-and-IoT\\_First-steps](https://github.com/ElektorLabs/220017-ESP32-C3-and-IoT_First-steps)
- [13] Roy Aarts, „Wetterstation mit ESP32“, Elektor 1-2/2019: [www.elektormagazine.de/magazine/elektor-69/42263](http://www.elektormagazine.de/magazine/elektor-69/42263)
- [14] Mathias Claußen, „Monster-LED-Uhr mit WLAN und Temperatur-Anzeige“, Elektor 5-6/2019:  
[www.elektormagazine.de/magazine/elektor-95/42563](http://www.elektormagazine.de/magazine/elektor-95/42563)
- [15] Jens Nickel, „Mein Weg in das IoT“, Elektormagazine.de: [www.elektormagazine.de/tags/mein-weg-ins-iot](http://www.elektormagazine.de/tags/mein-weg-ins-iot)

# IoT-Cloud à la Arduino



Von Tam Hanna (Slowakei)

Wer einen IoT-Clouddienst realisiert, verwendet einen MQTT-Broker. Diese Binsenweisheit gilt zumindest für alle Cloudsysteme der „großen“ Anbieter. Mit der Arduino-Cloud schickt das Arduino-Entwicklerteam nun eine Cloud-Variante ins Rennen, die durch eine andere Herangehensweise beeindruckt. Mit einer kleinen Anwendung wollen wir einen ersten Kontakt mit der Wolke aufnehmen: Wir senden einen Variablenwert an die Cloud und lassen eine LED blinken.

Die Basis der Arduino-Cloud ist das *Thing*. Darunter verstehen Massimo Banzis Leute ein Objekt der realen Welt wie einen Server, ein Controllerboard oder eine ähnlich intelligente Einrichtung. Analog zum in der „großen Cloud-Welt“ verbreiteten mentalen Konstrukt des Digital Twin [1] gilt auch im Fall des Things, dass dieses eine Reihe von Eigenschaften bereitstellt, die in der Arduino-Cloud *Variables* genannt werden.

## Wer profitiert von der Arduino-Cloud?

Schon an dieser Stelle sei angemerkt, dass die Arduino-Cloud explizit kein Ersatz für Amazon AWS IoT Core, Microsoft IoT Hub oder Yandex IoT Core ist. Sofern Sie große Mengen von Geräten zu verwalten haben, ist die Nutzung der alteingesessenen IoT-Cloud-Dienste anzuraten.

Das Alleinstellungsmerkmal der Arduino-Cloud besteht darin, dass die in den diversen Things angelegten Variablen nicht nur für dieses Thing sichtbar sind - ein Webdienst erlaubt je nach kommerzieller Ausbaustufe mehr oder weniger Interoperabilität. Zudem gibt es einen nach Ansicht des Autors sehr leistungsstarken Web-Editor zur Erzeugung von Dashboards. Die Arduino-Cloud kann ihre Stärken immer dann ausspielen, wenn es um einfache Entwicklungen mit hohen Ansprüchen sowohl an einfache Programmierbarkeit als auch an die Attraktivität der Dashboards geht. Dagegen ist eine bequeme Provisionierung der Endgeräte kein Ziel der Arduino-Cloud, die hier immer „von Hand“ durch den Entwickler zu erfolgen hat.

## Einrichtung der Hardware

Arduino-Cloud ist per se kein kostenloser Dienst. Die Preisstaffel in **Bild 1** informiert darüber, dass ein Großteil der interessanten Funktionen, zum Beispiel die Interaktion mit Fremd-Diensten, bezahlt werden muss. Logischerweise ist ein derartiger Dienst immer dann besonders profitabel, wenn er für möglichst viele Kunden benutzbar ist. Im Fall der Arduino-Cloud wirkt sich dies unter anderem dadurch aus, dass der Service nicht nur für hausesigene Boards (**Tabelle 2**) verfügbar ist, sondern auch viele andere Plattformen unterstützt, wie ein Blick auf die Kompatibilitäts-Liste [2] beweist. Neben Boards auf Basis des ESP8266 und des ESP32 gibt es Treiberbibliotheken, die verschiedenen Linux-basierten Systemen das Hoch- und Herunterladen von Informationen in die Arduino-Cloud erlauben.

Der Autor dieser Zeilen wird - schon aus reiner Bequemlichkeit - in den folgenden Schritten auf einen Nano RP2040 Connect setzen: Dieses auf dem Pico-Mikrocontroller der Raspberry Pi Foundation

FREE	ENTRY	MAKER <span style="background-color: yellow; border-radius: 50%; padding: 2px;">BEST VALUE</span>	MAKER PLUS
<p>Everything you need to learn Arduino, build your first IoT project and control it from your phone.</p> <ul style="list-style-type: none"> <li>✔ 2 Things</li> <li>✔ Unlimited dashboards</li> <li>✔ 100 Mb to store sketches</li> <li>✔ 1 day data retention</li> <li>✔ 200s/day of compilation time</li> <li>✔ LoRaWAN connectivity *</li> </ul> <p style="text-align: center; font-weight: bold; font-size: 1.2em;">Free</p> <p style="text-align: center; background-color: #008080; color: white; padding: 5px; border-radius: 10px;">GET STARTED</p>	<p>Get unlimited storage, scale up your IoT projects and get access to advanced features.</p> <ul style="list-style-type: none"> <li>✔ 10 Things</li> <li>✔ Unlimited dashboards</li> <li>✔ Unlimited storage for sketches</li> <li>✔ 15 days data retention</li> <li>✔ Unlimited compilation time</li> <li>✔ LoRaWAN connectivity</li> <li>✔ APIs</li> <li>✔ Over the Air Updates</li> </ul> <p style="text-align: center; font-weight: bold; font-size: 1.2em;">\$ 1.99/month</p> <p style="text-align: center; font-size: 0.8em;">Plus applicable taxes. Not available in Brazil.</p> <p style="text-align: center; background-color: #008080; color: white; padding: 5px; border-radius: 10px;">GET STARTED</p>	<p>For makers that are getting serious and need a reliable and sophisticated IoT platform to run their projects.</p> <ul style="list-style-type: none"> <li>✔ 25 Things</li> <li>✔ Unlimited dashboards</li> <li>✔ Unlimited storage for sketches</li> <li>✔ 3 months data retention</li> <li>✔ Unlimited compilation time</li> <li>✔ LoRaWAN connectivity</li> <li>✔ APIs</li> <li>✔ Over the Air Updates</li> <li>✔ Dashboard sharing</li> </ul> <p style="text-align: center; font-weight: bold; font-size: 1.2em;">\$ 5.99/month</p> <p style="text-align: center; font-size: 0.8em;">Plus applicable taxes. Not available in Brazil.</p> <p style="text-align: center; background-color: #008080; color: white; padding: 5px; border-radius: 10px;">GET STARTED</p>	<p>The option for makers with ambitions, that need to manage a small fleet of connected devices.</p> <ul style="list-style-type: none"> <li>✔ 100 Things</li> <li>✔ Unlimited dashboards</li> <li>✔ Unlimited storage for sketches</li> <li>✔ 1 year data retention</li> <li>✔ Unlimited compilation time</li> <li>✔ LoRaWAN connectivity</li> <li>✔ APIs</li> <li>✔ Over the Air Updates</li> <li>✔ Dashboard sharing</li> </ul> <p style="text-align: center; font-weight: bold; font-size: 1.2em;">\$ 19.99/month</p> <p style="text-align: center; font-size: 0.8em;">Plus applicable taxes. Not available in Brazil.</p> <p style="text-align: center; background-color: #008080; color: white; padding: 5px; border-radius: 10px;">GET STARTED</p>

Bild 1. Mit der Arduino-Cloud unternimmt Arduino erste Schritte in Richtung „Recurring Revenue“ (Stand 201.2022).

basierende Board bringt ein WLAN- Modul aus dem Hause u-blox mit, was die Inbetriebnahme erleichtert. Bevor es mit der Einrichtung der Arduino-Cloud losgeht, wird das Board per USB-Kabel am Computer angeschlossen.

Der erste Schritt besteht darin, die Webseite [3] zu besuchen und ein neues Arduino-Konto zu beantragen. Für unsere kleinen Experimente reicht das kostenlose Konto aus. Die Arduino-Cloud ist explizit an Entwickler gerichtet, die in dieser Beziehung über keine große Erfahrung verfügen. Zunächst klicken wir auf den Knopf *Create Thing*, um ein neues Thing anzulegen.

Lohn ist die Übersicht in **Bild 2**, deren dreigeteilte Konfiguration die fehlenden Elemente sichtbar macht. Da der Autor für dieses Experiment sein Arduino-Konto von allen Konfigurationsparametern befreit hat, können wir mit der „frischen“ Einrichtung eines Geräts beginnen. Die folgenden Schritte erfolgen unter Windows 10, wobei die Arbeit unter Ubuntu Linux identisch ist; unter Unix funktioniert die Hardwareerkennung aber meist besser.

Zunächst klicken wir auf das Verknüpfungs-Symbol in der Rubrik *Device* und entscheiden uns danach für die Rubrik *Set Up an Arduino Device*. Einige Sekunden nach dem Anklicken dieser Option weist das Backend darauf hin, dass die als *Arduino Create Agent* bezeichnete Komponente fehlt - klicken Sie auf den Download-Knopf, um die Software herunterzuladen und wie gewohnt zu installieren.

Beachten Sie, dass der *Create Agent* browserspezifisch ist: Wenn Sie die Installation unter Chrome durchführen, müssen sie eine abermalige Installation vornehmen. Eventuell aufpoppende Firewall-Warnungen werden abgenickt. Achten Sie dabei darauf, den Zugriff auf private und auf öffentliche Netzwerke gleichermaßen zu erlauben. In der Folge nistet sich der *Arduino Create Agent* in ihrer Task-Leiste ein - in manchen Fällen muss er danach nochmals aus dem Startmenü heraus aufgerufen werden, was die

**Tabelle 1. Arduino-Boards für die Arduino Cloud.**

**WLAN**

- MKR 1000 WiFi
- MKR WiFi 1010
- Nano RP2040 Connect
- Nano 33 IoT
- Portenta H7

**LoRaWAN**

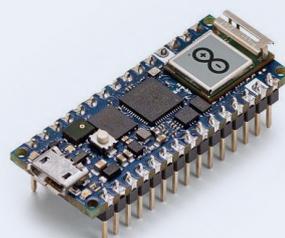
- MKR WAN 1300
- MKR WAN 1310

**GSM/NB-IoT**

- MKR GSM 1400
- MKR NB 1500

**ESP32/ESP8266**

- große Anzahl von Boards



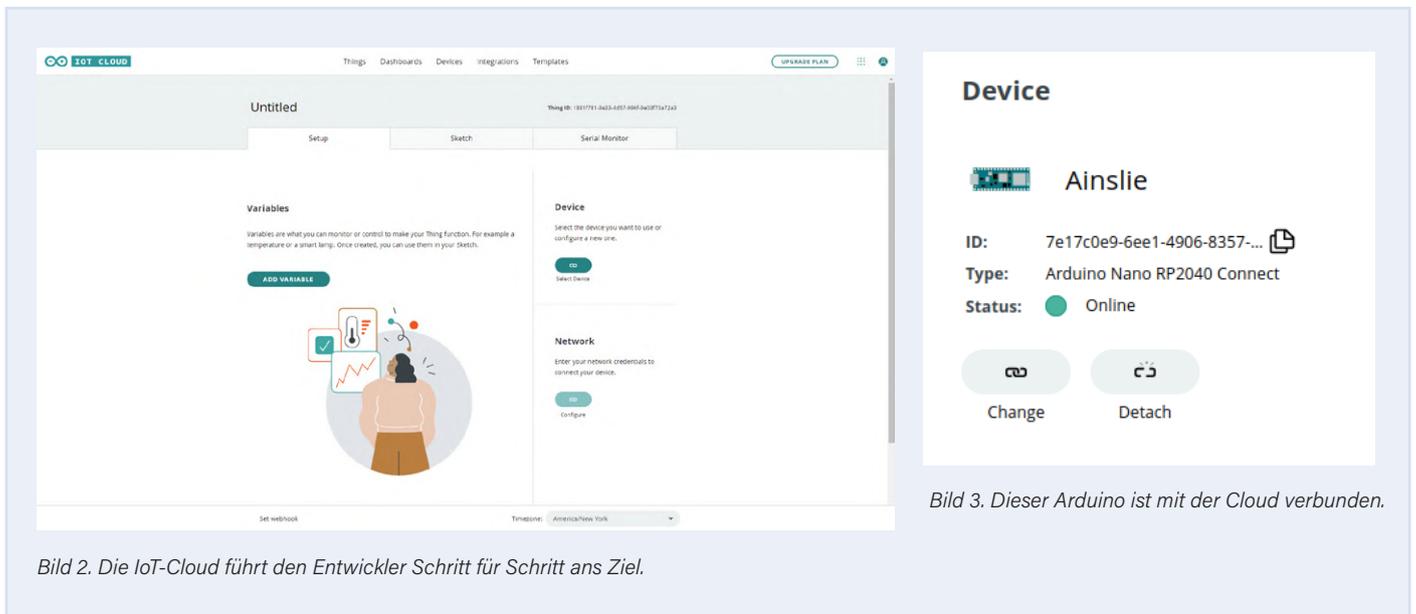


Bild 2. Die IoT-Cloud führt den Entwickler Schritt für Schritt ans Ziel.

Bild 3. Dieser Arduino ist mit der Cloud verbunden.

Treiber-Installation abschließt.

Im nächsten Schritt aktualisieren wir die Ansicht, bis uns die Arduino-Cloud darüber informiert, dass unser Nano RP2040 Connect erkannt ist. Klicken Sie danach auf den *Configure*-Knopf, um den Konfigurationsassistenten zu starten - er fragt nach einem „freundlichen“ Namen und initialisiert danach das Secure-Element des Zielsystems mit der grundlegenden Kommunikations-Software. Im Rahmen der Provisionierung kommt es unter Windows manchmal zu Problemen. Die zuverlässigere und erfolgreiche Methode ist es, den Provisionierungs-Schritt stattdessen unter Linux durchzuführen. Die Rubrik *Network* wird dabei von der Arduino-Cloud übrigens nicht automatisch freigegeben. Sie steht erst dann zur Verfügung, wenn Sie eine der für den Datenaustausch vorgesehenen Variablen erzeugen.

Deshalb klicken wir zunächst auf *Variables*, was den Dialog zum Hinzufügen einer neuen Variable öffnet. Als erstes vergeben wir den Namen `ledIntenBool` und entscheiden uns im Datentyp-Feld für den Wert *Boolean*. Die Arduino-Cloud unterstützt lustigerweise nicht nur C-Programmireinheiten, sondern realisiert auch „Wrapper“ um Größen der realen Welt.

Möchten Sie sich aber ausschließlich auf C beschränken, so empfiehlt sich die Auswahl der Option *Basic Types*. In den Feldern *Variable Permission* und *Variable Update Policy* könnten wir dann theoretisch Einstellungen vornehmen, aber die Voreinstellungen reichen für uns aus, weshalb wir den Dialog schließen. Im nächsten Schritt legen wir dann noch ein ein Feld vom Typ *Integer Number* an, dem wir den Namen `ledIntenInt` verleihen.

### Bearbeitung des Codes

Nach dem Anlegen der Variablen sehen Sie im *Sketch*-Tab rote Bubbles, die sie über Änderungen an der Programmstruktur informieren. Jetzt ist es möglich, auf das Verknüpfungs-Symbol in der Rubrik *Network* zu klicken, um die Einstellungen für das WLAN festzulegen. Der Autor empfiehlt Ihnen hier dringlich, die Werte auf einem Linux-Laptop mit einem Kommandozeilen-Tool wie *iwlist* zu ermitteln und über die Zwischenablage zu übernehmen. Im nächsten Schritt wechseln wir auf den Reiter *Sketch* und klicken auf das *Verify and Upload*-Symbol. Die Arduino-Cloud beginnt daraufhin mit der Kompilation des Codes und schickt ihn danach unter Nutzung des *Arduino Cloud Agent* auf den angeschlossenen RP2040. Bei erfolgreicher Auslieferung des Kompilats wird in der Cloud jedenfalls die Meldung „*Untitled\_dec25a uploaded successfully*“

on board *Arduino Nano RP2040 Connect (/dev/ttyACM0)*“ angezeigt. Spätestens nach dem obligatorischem Reset wird der RP2040 damit beginnen, über seinen WLAN-Transmitter nach Hause zu telefonieren. Mehrfaches Drücken von *F5* führt nach einiger Zeit dazu, dass das Gerät wie in **Bild 3** gezeigt mit „*Status: online*“ erscheint. Besitzer eines kostenpflichtigen Abonnements können Software-Updates auch direkt per WLAN ausliefern; für unsere kleinen Experimente mit dem kostenlosen Account ist eine Kabelverbindung erforderlich.

### Analyse des Codes, zum Zweiten

Der im *Sketch*-Tab eingblendete Editor ist nicht wirklich komfortabel. Der Button *Open full editor* führt zu einer vollwertigen cloudbasierten IDE, mit der sich kleinere Projekte komfortabel bearbeiten lassen. Als Erstes wollen wir uns den Inhalt der Datei *thingProperties.h* ansehen, die strukturelle Elemente enthält. Zuerst sehen wir die folgenden Deklarationen, die die für den WLAN-Zugriff erforderlichen Elemente bereitstellen:

```
const char SSID[] = SECRET_SSID; // Network SSID (name)
const char PASS[] = SECRET_PASS; // Network password
// (use for WPA, or use as key for WEP)
```

Den Eintrag von Namen und Passwort erledigt die Arduino-Cloud über die schon in der Rubrik *Network* eingepflegten Einstellungen. Der nächste Teil betrifft die beiden folgenden Variablen, deren Namen von der Einrichtung aus der *Variable*-Rubrik bekannt erscheinen sollten:

```
int ledIntenInt;
bool ledIntenBool;
```

Die Arduino-Cloud realisiert die Variablen „im Backend“ als gewöhnliche C-Variablen, die mit zusätzlicher Intelligenz ausgerüstet werden. Die Intelligenz findet sich unter anderem in der Methode *initProperties*, die sich nach folgendem Schema um die Einrichtung der für die Cloud-Kommunikation notwendigen Primitiva und Strukturen kümmert:

```
void initProperties() {
  ArduinoCloud.setThingId(THING_ID);
  ArduinoCloud.addProperty(ledIntenInt, READWRITE,
    ON_CHANGE, onLedIntenIntChange);
}
```



```
ArduinoCloud.addProperty(ledIntenBool, READWRITE,  
ON_CHANGE, onLedIntenBoolChange);  
}
```

Besonders interessant ist hier die Methode `addProperty`, die sich um die „Anmeldung“ der Attribute kümmert. Beachten Sie insbesondere das Übergeben der Funktionspointer `onLedIntenIntChange` und `onLedIntenBoolChange` - sie werden später noch eine wichtige Rolle spielen.

Die eigentliche Logik der Applikation findet sich dann in dem Sketch, der mit der (hier nicht abgedruckten) Inklusion der Header beginnt. Im nächsten Schritt findet sich die Initialisierung, die nach folgendem Schema abläuft:

```
void setup() {  
  Serial.begin(9600);  
  delay(1500);  
  
  initProperties();  
  
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);  
  setDebugMessageLevel(2);  
  ArduinoCloud.printDebugInfo();  
}
```

Aus Sicht der Arduino-Programmierungsumgebung ist die Arduino-Cloud ein Hardwaretreiber wie jeder andere. Das globale Objekt `ArduinoCloud` exponiert dabei eine Gruppe von Funktionen, über die Ihr Code mit dem Cloud-Treiber kommuniziert. Von besonderer Wichtigkeit ist hier der Aufruf von `setDebugMessageLevel`, der die „Gesprächigkeit“ des Treibers festlegt - desto höher der numerische Wert, desto mehr Debugger-Informationen gibt der Cloud-Treiber über die serielle Schnittstelle des Boards aus.

Bei „komplizierten“ Treibern stellt sich immer auch die Frage, wie die Bereitstellung der Rechenleistung erfolgt. Das von der Arduino-Cloud für uns angelegte Projektskelett beantwortet diese Frage in der Methode `loop`, die sich nach folgendem Schema um die Zuweisung von Rechenleistung kümmert:

```
void loop() {  
  ArduinoCloud.update();  
}
```

Von Haus aus spendiert uns die Arduino-Cloud außerdem noch die drei folgenden Listener-Methoden:

```
void onTestScheduleChange() {  
}  
void onLedIntenBoolChange() {  
}  
void onLedIntenIntChange() {  
}
```

`onLedIntenBoolChange` und `onLedIntenIntChange` sind dabei für die weiter oben im Backend angelegten Variablen verantwortlich, während `onTestScheduleChange` „interne“ Funktionen der Arduino-Cloud realisieren hilft.

Im nächsten Schritt wollen wir uns darum kümmern und der Cloud-Lösung Eigenintelligenz einzuschreiben. Der verwendete Arduino hilft uns dabei insofern, als dass er neben der gewöhnlichen (hier roten) LED an Pin 13 auch eine RGB-LED mitbringt, die per Pulsbreitenmodulation ansprechbar ist (Kanäle LEDR, LEDG und LEDB).

Als Nächstes kehren wir deshalb in die Funktion `setup` zurück, in der wir die verschiedenen Pins initialisieren:

```
void setup() {  
  ...  
  ArduinoCloud.printDebugInfo();  
  
  pinMode(LED_BUILTIN, OUTPUT);  
  pinMode(LEDB, OUTPUT);  
}
```

Von der Cloud ausgelöste Änderungen führen logischerweise zur Aktivierung der Listener, in denen wir die angelieferten Werte in Richtung der Hardware weiterschreiben:

```
void onLedIntenBoolChange() {  
  digitalWrite(LED_BUILTIN, ledIntenBool);  
}  
void onLedIntenIntChange() {  
  analogWrite(LEDB, ledIntenInt);  
}
```

An dieser Stelle können Sie den Sketch abermals auf die Platine übertragen. Die RGB-LED spielt insofern eine Sonderrolle, als dass sie durch eine Ausgabe einer Null aktiviert wird. Da unsere frisch initialisierten Variablen den in **Bild 4** gezeigten Zustand aufweisen, leuchtet nach der erfolgreichen Initialisierung die blaue Diode der RGB-LED auf.

## Modifizieren des Variableninhalts

Böse Zungen behaupten, dass der Grad beziehungsweise die Geschwindigkeit der Management-Adoption eines IoT-Systems zu einem wesentlichen Teil von der grafischen „Qualität“ von Dashboards und Co. beeinflusst wird. Der Autor möchte sich

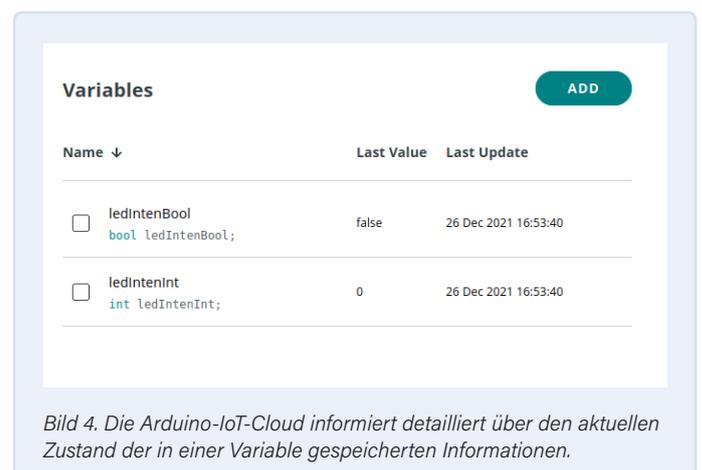


Bild 4. Die Arduino-IoT-Cloud informiert detailliert über den aktuellen Zustand der in einer Variable gespeicherten Informationen.

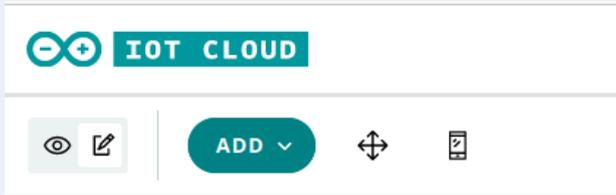


Bild 5. Die oben links eingeblendeten Bearbeitungs-Steuerelemente legen den Betriebsmodus fest.

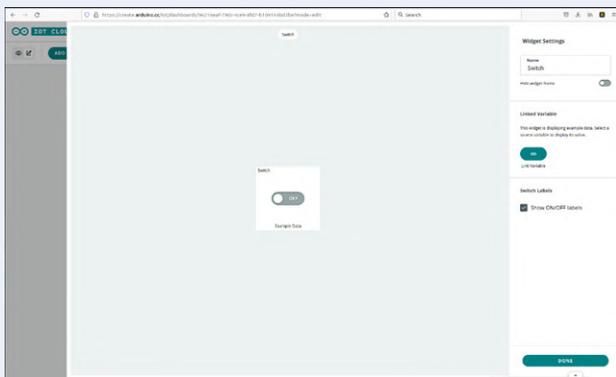


Bild 6. Die Bearbeitungs-Oberfläche für Steuerelemente ist „modal“.

Add variable

Name  
TamsSchedule

Sync with other Things

Schedule eg. Every MON at 8:00 AM

Declaration  
CloudSchedule tamsSchedule ;

Variable Permission

Read & Write  Read Only

Variable Update Policy

On change  Periodically

ADD VARIABLE CANCEL

Bild 7. Der Datentyp Cloud Scheduler ist eine „normale“ Variable wie jede andere.

an dieser Stelle zu diesem häufig geäußerten Vorurteil nicht weiter äußern und empfiehlt stattdessen, in das Haupt-Backend der Arduino-Cloud zurückzukehren. Klicken Sie dort auf das *Dashboards*-Tab, das sie im Fall eines jungfräulichen Accounts auffordert, ein neues Dashboard anzulegen. Klicken Sie den *Build Dashboard*-Knopf an, um den Editor zu starten, was auch bei schneller Internetverbindung die eine oder andere Gedenksekunde in Anspruch nimmt.

Am wichtigsten ist dabei der in **Bild 5** gezeigte und oben links eingeblendete Moduswechsler – er erlaubt die Festlegung der Funktion der Weboberfläche. Zum Hinzufügen neuer Widgets müssen Sie logischerweise im Bearbeitungsmodus sein, der durch das Bleistift-Piktogramm dargestellt ist. Im nächsten Schritt können Sie auf das *Add*-Button klicken, um sich für verschiedene Steuerelemente zu entscheiden. Als Erstes klicken wir einen *Switch* an, der daraufhin in der in **Bild 6** gezeigten Bearbeitungsoberfläche erscheint.

Wichtig ist vor allem die Rubrik *Linked Variable*, in der sich abermals das vom Einrichten der Netzwerk-Zugangskonfiguration bekannte Verbinden-Symbol verbirgt. Klicken Sie es an, um eine Liste aller im Cloud-Konto enthaltenen Things und der in ihnen angelegten Variablen zu aktivieren. Daraus folgt hier, dass wir uns für die Variable `ledIntenBool` entscheiden und sie durch die *Link Variable*-Option verbinden. Danach fehlt noch ein Klick auf *DONE*, um die multimodale Bearbeitungsoberfläche zu schließen und in das Dashboard zurückzukehren. Danach klicken wir auf das Augen-Symbol, um den Schalter zur Aktivierung freizugeben. Das Ein- und Ausschalten des Switch sorgt dafür, dass die rote LED neben der MicroUSB-Buchse aufleuchtet beziehungsweise erlischt.

Um die Helligkeit der blauen Leuchtdiode einstellen zu können, müssen wir den Dashboard-Editor wieder in den Bearbeitungsmodus zurückversetzen und über *Add* -> *Widgets* abermals ein neues Steuerelement hinzufügen. Der Autor entscheidet sich dieses Mal für den Typ *Slider*. In seiner Bearbeitungsoberfläche legen wir als *Value Rang* den Wert von 0 - 255 fest. Die Verknüpfung erfolgt durch die Variable `letIntenInt`, die ja für die „Helligkeitssteuerung“ von RGB-LEDs verantwortlich ist. Zu guter Letzt wechseln wir auch hier in den Aktivierungsmodus und stellen erfreut fest, dass Änderungen des Sliders die Helligkeit der blauen Leuchtdiode beeinflussen.

## Eile mit Weile

Zu der Zeit, als der Artikel verfasst wurde, war die Funktion zur Einrichtung von Web-Cronjobs brandneu: Dahinter versteht Arduino die Möglichkeit, „Aufgaben“ am IoT-Endgerät programmatisch auszulösen. Zur Vorführung dieser Möglichkeiten wollen wir eine neue Variable vom Typ *Schedule* einrichten. **Bild 7** zeigt die gewünschte Konfiguration. Vorrangiger Lohn unserer Arbeit ist, dass wir im Code nun die folgende neue Variable vorfinden:

```
CloudSchedule tamsSchedule;
```

Interessant ist, dass die Einstellung der Arbeitsdauer über das Dashboard erfolgt - die in den **Bild 8** gezeigte Einstellungsseite verrät mehr über das dazu vorgesehene Steuerelement.

Im folgenden Schritt müssen wir uns darum kümmern, die „lokale“ Verarbeitung der in `tamsSchedule` enthaltenen Werte durchzuführen:

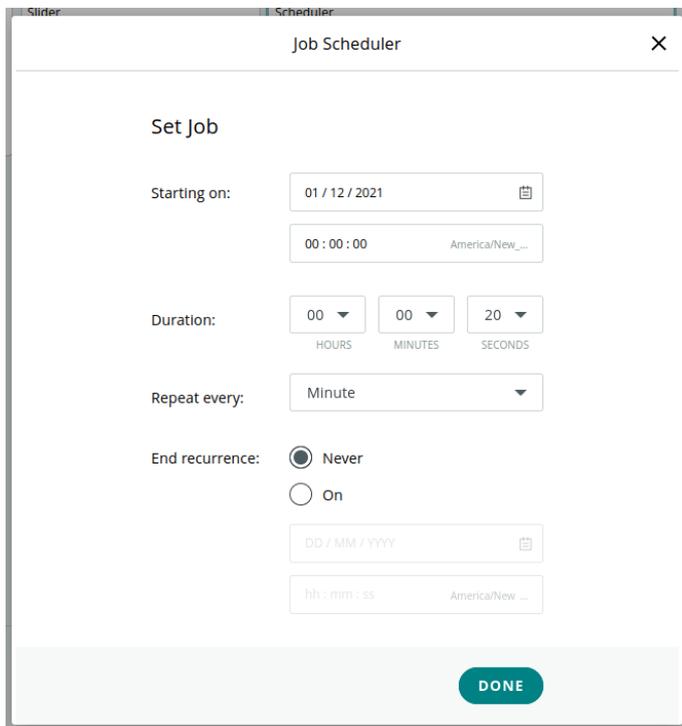


Bild 8. Die Konfiguration des Schedulers erfolgt per Dashboard.

```
void setup() {
  ...

  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(LED_B, OUTPUT);
  pinMode(LED_R, OUTPUT);
}

void loop() {
  ArduinoCloud.update();
  // Your code here
  if(tamsSchedule.isActive()){
    digitalWrite(LED_R, HIGH);
  }
  else{
    digitalWrite(LED_R, LOW);
  }
}
```

Wichtig ist hier, dass das „Herausschreiben“ der in `tamsSchedule` angelieferten Informationen die ausschließliche Aufgabe des Entwicklers ist. Die Cloud beschränkt sich darauf, den in `tamsSchedule` enthaltenen Wert periodisch anzupassen. Die hier gezeigte permanente Polling-Prozedur in der Methode `loop` mag aus ressourcentechnischer Sicht nicht optimal sein, funktioniert aber problemlos. Schicken Sie das Programm anschließend zum Arduino und erfreuen Sie sich des periodischen Ein- und Ausschaltens der roten Komponente der RGB-LED.

An dieser Stelle konnte sich der Autor einen weiteren kleinen Versuch nicht verkneifen. Er schaltete das „Labor-WLAN“ ab und überprüfte das Systemverhalten. Der Arduino reagierte darauf zuerst mit „chaotischem“ Ausschalten der blauen RGB-Komponente und der LED an Pin 13, um einige Sekunden später komplett neu zu starten.

Zum Zeitpunkt der Drucklegung dieses Artikels war nicht wirklich klar, wie die Arduino-Cloud auf den Verlust der Funkverbindung

vom Endgerät zum Server reagiert - nach Ansicht des Autors wäre ein Nachbessern an dieser Stelle wünschenswert.

## Fazit

Sofern die zu realisierende Aufgabe nicht extrem sicherheitskritisch ist (Stichwort: Keine Turbinen- oder Heizungssteuerungen!), bietet die Arduino IoT-Cloud schon in der vorliegenden Version eine bequeme und niederschwellige Möglichkeit, Cloud-Backends zu realisieren, ohne sich auf den Kampf mit MQTT und Co. einlassen zu müssen. Nach Ansicht des Autors ein durchaus empfehlenswertes Produkt! ◀

210550-02

## Ein Beitrag von

Idee, Illustrationen und Text: **Tam Hanna**

Redaktion: **Rolf Gerstendorf**

Layout: **Harmen Heida**

## Sie haben Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an den Autor unter [tamhan@tamoggemon.com](mailto:tamhan@tamoggemon.com) oder an die Redaktion unter [redaktion@elektor.de](mailto:redaktion@elektor.de).

## WEBLINKS

- [1] Digitaler Zwilling: [https://de.wikipedia.org/wiki/Digitaler\\_Zwilling](https://de.wikipedia.org/wiki/Digitaler_Zwilling)
- [2] Liste kompatibler Boards: <https://bit.ly/3t8Vl3W>
- [3] Arduino Things: <https://create.arduino.cc/iot/things>



## PASSENDE PRODUKTE

- > **Arduino MKR WiFi 1010 (SKU 19935)**  
[www.elektor.de/19935](http://www.elektor.de/19935)
- > **Arduino Nano RP2040 connect (SKU 19754)**  
[www.elektor.de/19754](http://www.elektor.de/19754)
- > **Arduino Nano 33 IoT (SKU 19937)**  
[www.elektor.de/19937](http://www.elektor.de/19937)

# Arduino Shield für zwei Geiger-Müller-Zählrohre

Ein hochempfindlicher, sehr stromsparender Strahlungsmesser



Gabriele Gorla (Italien/USA)

Das Geiger-Müller-Zählrohr-Shield verwandelt Ihren Arduino Uno in ein Instrument zur Messung und Aufzeichnung von radioaktiver Strahlung. Es bietet sogar Platz für zwei Rohre, um eine besonders hohe Empfindlichkeit zu erzielen. Das Shield kann mit einem Dragino-Shield für LoRa-Konnektivität kombiniert werden. Sammeln Sie Strahlungsdaten „im Feld“ - und greifen Sie von überall auf der Welt darauf zu!

Wie viele andere Hobbyelektroniker war ich schon immer von Radioaktivität und den Sensoren, mit denen sie nachgewiesen werden kann, fasziniert. Geiger-Müller-Rohre [1] sind eine gängige und relativ kostengünstige Methode zur Messung von Strahlung. Mein Projekt habe ich auf den Namen GRAD getauft. Es ist eine Komplettlösung für die Strahlungsmessung im Formfaktor eines Arduino-Shields. Seine Hauptmerkmale sind eine hohe Empfindlichkeit durch die Verwendung zweier Zählrohre und ein sehr geringer Strombedarf.

## Was ist ein Geiger-Müller-Zähler?

Jeder Geiger-Müller-Zähler benötigt vier wesentliche Funktionsblöcke.

- **Geiger-Müller-Rohr:** Das Rohr hat zwei Anschlüsse und ist mit einem Niederdruck-Gasgemisch gefüllt. Wenn es mit der entsprechenden Spannung vorgespannt wird, ionisiert das Gas und leitet kurzzeitig Strom, sobald es von Strahlung getroffen wird. Je nach Rohrtyp ist es möglich, Alpha- und Betaeichen sowie Gammastrahlung nachzuweisen. Das im GRAD verwendete SBM-20 ist besonders empfindlich für Gamma- und hochenergetische Betastrahlung.
- **Hochspannungsnetzteil:** Das verwendete Rohr wird im so genannten Geiger-Müller-Bereich betrieben. Dies ist ein Vorspannungsbereich, in dem die Impulszahl nahezu unabhän-

gig von der Vorspannung ist. Bei herkömmlichen Röhren liegt die Vorspannung zwischen 400 V und 500 V. Die optimale Spannung für die SBM-20 beträgt etwa 400 V.

- **Impulsdetektor:** Die aus dem Rohr kommenden Impulse sind sehr kurz und von unterschiedlicher Spannungsamplitude. Der Impulsdetektor bereitet das Signal so auf, dass es von der folgenden digitalen Stufe leicht gezählt werden kann.
- **Impulszählung:** Die Impulse werden über ein festes Zeitintervall gezählt, um einen CPS- (counts per second) oder CPM-Wert (counts per minute) zu berechnen. Dieser Wert kann mit den im Datenblatt des Rohrs angegebenen Parametern grob in eine Dosisleistung umgerechnet werden. Für die Impulszählung und deren Visualisierung und/oder Protokollierung wird ein Arduino-Board verwendet.

## Hochspannung

Im Internet gibt es viele Schaltungen, um die hohe Spannung für den Betrieb von Geiger-Müller-Röhren zu erzeugen. Viele davon sind Aufwärtswandler, die mit einem 555-Timer aufgebaut sind, einige mit offenem Regelkreis, andere mit Rückkopplung. Die Entwürfe mit offenem Regelkreis wurden nicht in Betracht gezogen, da sie eine individuelle Abstimmung jedes Geräts erfordern und keine stabile Spannung bei hohen Impulszahlen liefern. Ausführungen mit geschlossenem Regelkreis sind wegen ihrer Stabilität besser geeignet. Um die Gesamtstromaufnahme niedrig zu halten, sollte jedoch der Rückkopplungsschleife besondere Aufmerksamkeit gewidmet werden (12 µA Strom bei 400 V sind etwa 5 mW). Um die Leistungseinbußen bei der Hochspannungsrückkopplung zu vermeiden, verwenden die elegantesten Lösungen einen nicht isolierten Aufwärtsregler wie den LT3420 von Analog Devices, der die Spannung auf der Primärseite erfasst.

Wir haben stattdessen einen einfachen Aufwärtswandler mit einem sehr geringen Strom-Feedback implementiert. Unser Entwurf in **Bild 1** basiert stark auf den Theremino-Geiger-Adaptoren in [2]. Wir haben die dortigen drei Vorschläge (SMD, DIY und „Flintstones“) zu einer Schaltung mit Z-Dioden-Rückkopplung kombiniert, die ausschließliche

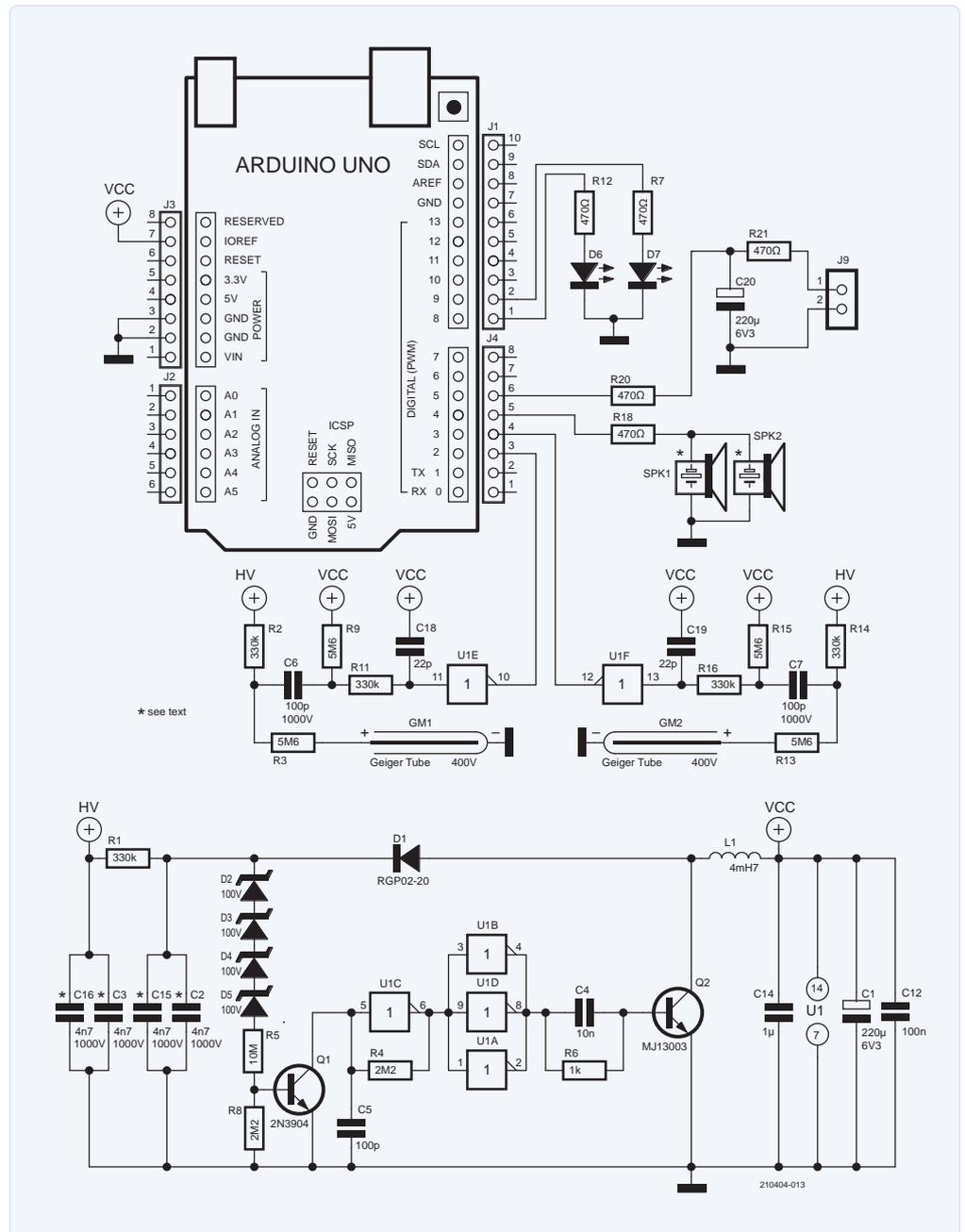


Bild 1. Hochspannend, aber einfach: Die Schaltung des GRAD03-Projekts [3].

bedachtete Bauteile verwendet.

Der Schmitt-Trigger-Inverter U1C bildet mit R4 und C5 einen Oszillator, der die Impulse zur Ansteuerung des Schalttransistors Q2 erzeugt. U1A, U1B und U1D sind parallel geschaltet, um den Basisstrom von Q2 zu erhöhen.

Die Rückkopplung wird mit einer Reihe von Z-Dioden mit sehr geringem Leckstrom (D2...D5) realisiert. Wenn die Ausgangsspannung die resultierende Z-Spannung übersteigt, schaltet Q1 durch und stoppt den Oszillator. Wenn die Ausgangsspannung wieder abfällt, sperrt Q1 wieder und gibt den Oszillator frei. R1 und C3 reduzieren die Welligkeit der Ausgangsspannung. Zusätzliche Vorwiderstände versorgen jede Röhre separat (R2 und R3 für GM1, R13 und R14 für GM2).

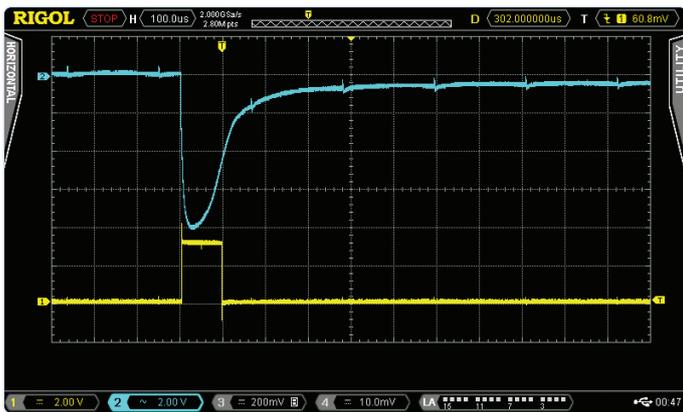


Bild 2. Oszillogramm eines ionisierenden Teilchens, das auf ein Rohr trifft, und der resultierende Impuls.

## Zählung

Die Zählpulse werden auf der Hochspannungsseite der Rohre über Koppelkondensatoren (C6 und C7) gleichspannungsfrei abgegriffen und von den beiden noch zur Verfügung stehenden Schmitt-Trigger-Invertern „digitaltauglich“ geformt (Bild 2).

J1 bis J4 sind die Standard-Arduino-Anschlüsse, von denen das Shield nur wenige Pins verwendet. Die digitalen Eingänge D2 und D3, an denen die beiden Rohre angeschlossen sind, sind die Interrupt-Pins des Arduino Uno. Über sie kann der Arduino im Hintergrund Impulse zählen, während er andere Aktionen durchführt.

Die digitalen Ausgänge D8 und D9 des Arduino sind mit einem

LED-Paar verbunden, um anzuzeigen, wenn ionisierende Strahlung auf das jeweilige Rohr trifft. Pin D4 ist mit dem Piezo-Lautsprecher SPK1 verbunden, um eine akustische Rückmeldung zu erhalten. Schließlich gibt es noch einen digitalen Ausgang D5/J9 mit einem Tiefpassfilter (R20, R21 und C20), an den ein einfaches analoges 10-mA-Panelmeter angeschlossen werden kann.

## Platinentwurf

Der Schaltplan und die Platine (Bild 3) wurden in KiCad entworfen. Die Designdateien stehen auf der Elektor-Labs-Seite [3] dieses Projekts im Bereich *Elements* zum Download bereit. Die Gerber- und Bohrdateien der Leiterplatte sind ebenfalls dort zu finden. Sie können diese Dateien verwenden, um eine Platine bei Ihrem bevorzugten PCB-Lieferanten zu bestellen. Zu guter Letzt finden Sie dort eine Excel-Tabelle mit einer sehr detaillierten Stückliste, die vor allem die kritischeren (Hochspannungs-!) Bauteile dokumentiert, die für dieses Projekt benötigt werden, einschließlich Angaben zu Herstellern, Typennummern und sogar Bestellcodes.

Bild 4 zeigt die bestückte Leiterplatte mit eingebauten SBM-20-Rohren. Die Bauteilpaare SPK1/SPK2, C3/C16 und C2/C15 sind jeweils nur alternative Fußabdrücke auf der Platine; es wird also nur SPK1 oder SPK, C3 oder C16 und C2 oder C15 installiert. J5 und J6 auf der Platine bezeichnen zwei zusätzliche Anschlussflächen für die Minuspole kürzerer Geiger-Müller-Rohre.

## Zählrohroptionen

Die Platine ist für den Betrieb mit den russischen 105 mm langen Zählrohren SBM-20, STS-5, der chinesischen J305 oder der 90-mm-Rohre J305 und M4011 ausgelegt. Jedes andere 400-V-Rohr funktioniert ebenfalls, wenn es mechanisch anschließbar ist. In allen



## STÜCKLISTE

### Widerstände:

R1,R2,R11,R14,R16 = 330 k  
 R3,R9,R13,R15 = 5M6  
 R4,R8 = 2M2  
 R5 = 10 M  
 R6 = 1 k  
 R7,R12,R18,R20,R21 = 470 Ω

### Kondensatoren:

C1,C20 = 220 µ, 6,3 V  
 C2,C3 = 4700 p, 1000 V  
 C4 = 10 n, 6,3 V  
 C5 = 100 p, 50 V  
 C6,C7 = 100 p, 1000 V  
 C12 = 100 n, 50 V  
 C14 = 1 µ, 6,3 V  
 C15,C16 = 4700 p, 1000 V  
 C18,C19 = 22 p, 50 V

### Induktivität:

L1 = 4m7

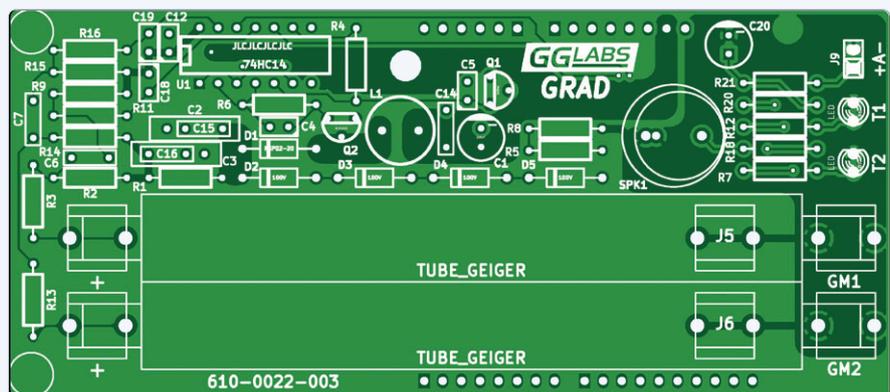


Bild 3. Das Platinenlayout des GRAD03-Shields.

### Halbleiter:

D1 = Diode 1 A, 800 V  
 D2,D3,D4,D5 = Z-Diode 100 V, 1,5 W  
 D6, D7 = rote LED, 3 mm (T1, T2 auf der Platine)  
 Q1 = 2N3904  
 Q2 = MJ13003  
 U1 = 74HC14

### Außerdem:

GM1,GM2 = Geiger-Müller-Zählrohr, z.B. SBM-20  
 J1,J2,J3,J4 = Anschlussleisten-Satz für Arduino-Uno-Shield  
 SPK1 = AC-Piezo-Summer, z.B. AC-1205G-N1LF

Fällen muss der Anodenanschluss so kurz wie möglich gehalten werden, um die parasitäre Kapazität zu minimieren. Bei Zählrohren, die andere Spannungen benötigen, sollte(n) die Zenerdiode(n) ausgetauscht werden, um die erforderliche Spannung zu erhalten. Jede Diode mit einem Leckstrom von 0,5  $\mu$ A oder weniger sollte funktionieren.

### Leistung

Die Stromversorgung ist auf 65  $\mu$ A bei 5 V, also 325  $\mu$ W ausgelegt. Bei batteriebetriebenen Geräten kann die Schaltung auch mit 3,3 V versorgt werden, so dass die Leistungsaufnahme auf nur 150  $\mu$ W (45  $\mu$ A) sinkt.

### Arduino Beispielcode

Ein einfacher Arduino-Sketch zur Ansteuerung des Zählers ist unter *Project Elements* auf der Elektor Labs-Seite verfügbar. Alle 60 Sekunden gibt die Software über die serielle Schnittstelle eine kommagetrennte Zeile von Werten (CSV) aus mit: einer fortlaufenden Nummer, dem Rohwert für jedes Rohr, einem gleitenden Mittelwert von `cnt1 + cnt2` und einer auf Grundlage dieses Mittelwerts berechneten Dosisleistung in  $\mu$ Sv/h. Das folgende Datenfragment zeigt den Beginn einer Messung der Hintergrundstrahlung in Santa Clara mit zwei SBM-20-Rohren auf dem GRAD-Shield. In Kalifornien strahlt also nicht nur die Sonne!

```
Seq , cnt1 , cnt2 , avg10 ,  $\mu$ Sv/h
1 , 14 , 19 , 33.0 , 0.075
2 , 10 , 17 , 32.3 , 0.073
3 , 16 , 12 , 31.8 , 0.072
4 , 16 , 24 , 32.5 , 0.074
5 , 13 , 11 , 31.6 , 0.072
```

Die Grafik in **Bild 5** zeigt die gesamte Messung über einen längeren Zeitraum.

Am Anfang des Sketches gibt es einige Definitionen, die entsprechend der Boardkonfiguration und/oder den Benutzerwünschen eingestellt werden können. Der erste Satz konfiguriert die Parameter der Geiger-Müller-Rohre und die Größe des Fensters für die Messwertausgabe.

```
#define CPM2USV 220 // tube CPM to  $\mu$ Sv/h conversion
// factor
#define TUBES 2 // number of tubes installed
#define WSIZE 10 // moving average window (in
// minutes)
```

Der Parameter `CPM2USV` ist die Zählung pro Minute für 1  $\mu$ Sv/h für das eingesetzte Zählrohr. Leider ist in den veröffentlichten Datenblättern keine „richtige“ Zahl für diesen Parameter angegeben. Begeisterte Geiger-Müller-Bastler im Netz verwenden für die SBM-20-Rohre Werte zwischen 130 und 220 (Umrechnungsfaktor von 0,0075...0,0045).

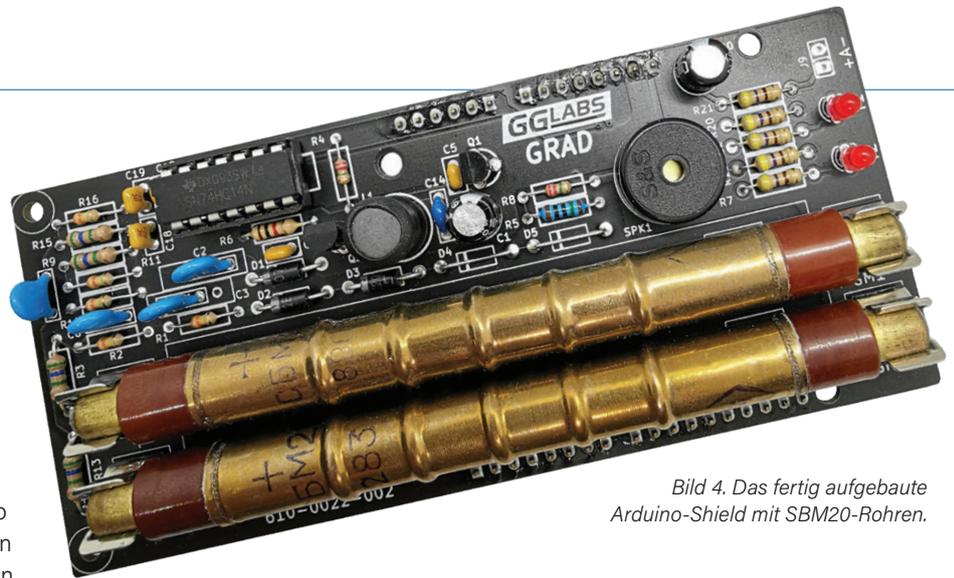


Bild 4. Das fertig aufgebaute Arduino-Shield mit SBM20-Rohren.

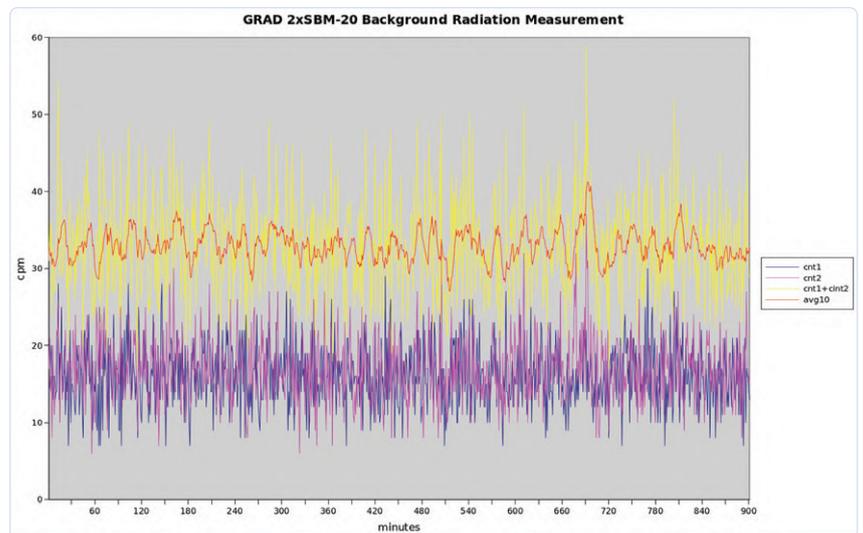


Bild 5. Hintergrundstrahlung im kalifornischen Santa Clara.

Der Parameter `TUBES` gibt, wie der Name schon sagt, die Anzahl der installierten Zählrohre an. Gültige Werte sind 1 und 2. Der Parameter `WSIZE` schließlich bestimmt die Größe des Fensters, in dem der gleitende Mittelwert der Zählungen dargestellt wird. Der Standardwert von 10 bedeutet ein 10-Minuten-Fenster.

Der zweite Satz von Parametern legt fest, wie die optionalen Hardware-Funktionen angeschlossen werden:

```
#define LED1_PIN 8 // pin for TUBE1 LED
#define LED2_PIN 9 // pin for TUBE2 LED
#define SPKR_PIN 4 // pin for speaker connection
#define LED_BLINK_MS 20 // duration of LED blink for
// each count
```

Alle drei `_PIN`-Definitionen wählen den Arduino-Pin aus, an dem die LED oder der Lautsprecher angeschlossen ist. `LED_BLINK_MS` definiert die Zeit in Millisekunden, die die LED für jeden Zählimpuls aufleuchtet.

### LoRaWAN-Vernetzung

Der grundlegende Sketch erfordert, dass das Gerät über USB mit einem Computer verbunden ist. Oft aber wünscht man sich, den Sensor

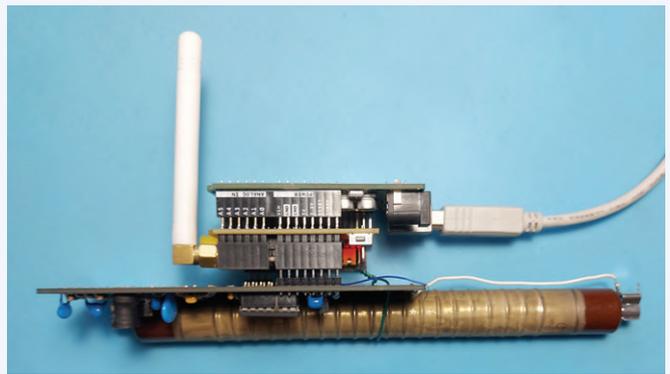
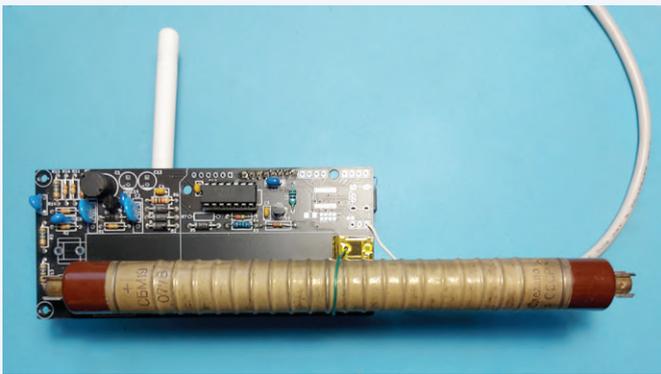


Bild 6. GRAD03-Platine, Arduino Uno und das Dragino-LoRa-Shield kombiniert.

weit ab vom Computer zu platzieren, so dass keine USB-Verbindung möglich ist.

Auf den Arduino Uno und das GRAD-Shield kann ein Dragino-LoRa-Shield aufgesteckt werden, um eine Verbindung zum LoRaWAN zu erreichen. Leider verwendet das Dragino-Schild auch den Arduino-Pin D2, so dass es zwei Optionen gibt:

**Option mit einem Zählrohr:** In diesem Fall kann nur das Rohr GM2 verwendet werden. Pin 10 des Schmitt-Triggers U1 muss aufgetrennt werden, um Störungen der LoRa-Kommunikation zu vermeiden. Die Fotos in **Bild 6** zeigen den vollständigen Stapel mit einer einzigen russischen SBM-19.

**Option mit zwei Zählrohren:** In diesem Fall müssen sowohl das Dragino-Shield als auch die GRAD-Platine modifiziert werden. Beim

Dragino-Shield müssen R5 und J\_DIO0 entfernt und die Verbindung zwischen Funk-Pin DIO0 und Arduino-Pin D7 mit einem Draht realisiert werden.

Bei der GRAD-Platine müssen R7, R12, C20 und die beiden LEDs entfernt werden. R21 sollte durch einen Draht ersetzt und eine einzelne LED an J9 (ersetzt den analogen Zählerausgang) angebracht werden.

Ein Sketch, der mit *The Things Network* (TTN) kommuniziert und alle 60 Sekunden die Zählwerte der Rohre anzeigt, steht ebenfalls zum Download bereit. Es verwendet die Arduino-Bibliothek LMIC, um das Dragino-LoRa-Shield zu steuern und verbindet sich mit dem TTN mittels *Over The Air Activation* (OTAA). Zur Anzeige der Daten genügt ein einfacher Node-RED-Flow, wie in **Bild 7** zu sehen.

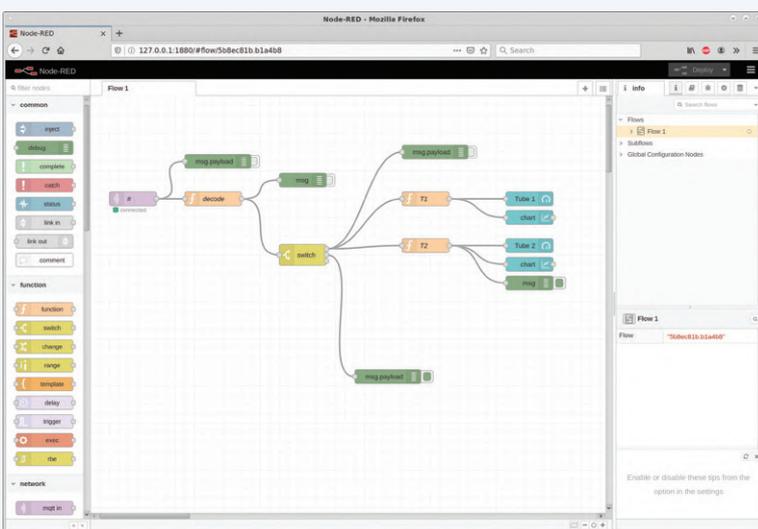
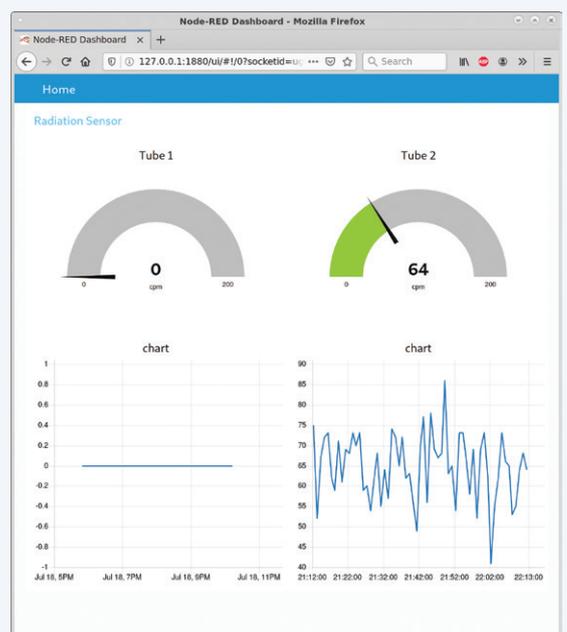


Bild 7. Node-Red-Flow und Dashboard.



## Messung der Strahlung

Um die korrekte Funktion des Geigerzählers zu überprüfen, wird eine Strahlungsquelle benötigt. So etwas kann man online bei speziellen Anbietern erwerben, alternativ aber bestimmte alte Schätzchen in einem Secondhandladen oder bei Ebay erstehen, die kleine Mengen radioaktiver Materialien enthalten: Uranglas aus Böhmen, einen alten (und mittlerweile in der EU verbotenen) Thorium-Glühstrumpf, bestimmtes farbig glasiertes Keramikgeschirr aus China oder den USA.

In Ermangelung solcher Gegenstände kann man auch einfach die Zerfallsprodukte von Radon untersuchen, die vom Luftfilter einer Klimaanlage oder eines Luftreinigers aufgefangen werden (**Bild 8**). Diese haben eine relativ kurze Halbwertszeit (in der Größenordnung von einigen zehn Minuten). Lassen Sie die Klimaanlage ein paar Stunden lang laufen und messen dann sofort am Filter. Wenn das Filter fein genug ist, dürfte er Strahlung mit einer Rate emittieren, die um ein Vielfaches höher ist als die lokale Hintergrundstrahlung. Die Grafik zeigt, dass die Strahlung fast das Zehnfache der Hintergrundstrahlung erreicht und dann der für den radioaktiven Zerfall typischen Exponentialkurve folgt. ◀

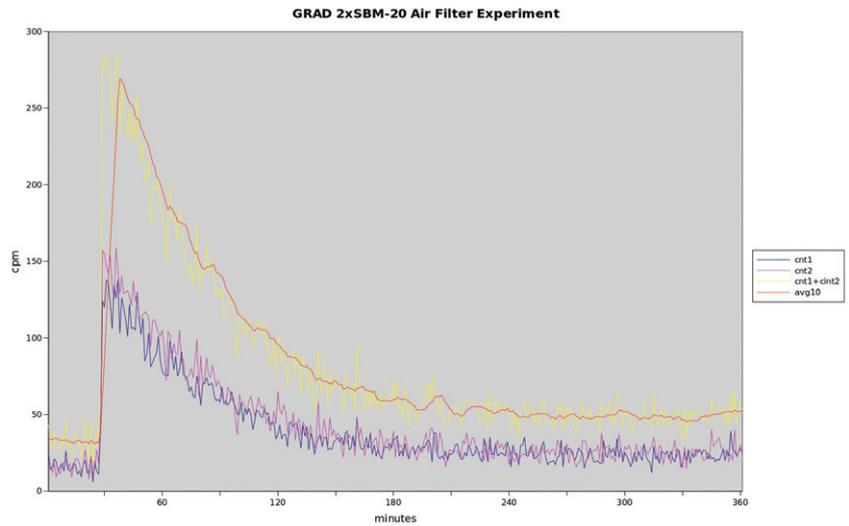


Bild 8. Durch Zerfall von Radon verursachte Strahlung aus einem Klimaanlagefilter.

210404-02

Dieser Artikel basiert auf dem Material, das auf der Elektor Labs-Seite dieses Projekts [3] vorgestellt wird. Dort finden Sie alle Downloads für GRAD03 sowie Diskussionen und Anmerkungen zu diesem Thema.

### Ein Beitrag von

Entwurf: Gabriele Gorla

Text: Gabriele Gorla, Luc Lemmens

Illustrationen: Gabriele Gorla, Patrick Wielders

Redaktion: Jens Nickel, C. J. Abate

Übersetzung: Rolf Gerstendorf

Layout: Giel Dols

### Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an den Autor unter [gorlik@yahoo.com](mailto:gorlik@yahoo.com) oder an Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).



### PASSENDE PRODUKTE

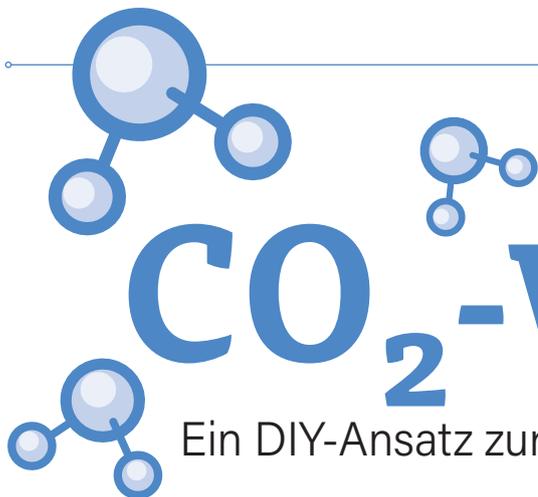
- > **Arduino Uno SMD Rev3 SKU 19938**  
[www.elektor.de/19938](http://www.elektor.de/19938)
- > **MightyOhm Geigerzähler-Kit (inkl. Gehäuse) SKU 18509**  
[www.elektor.de/18509](http://www.elektor.de/18509)

### WEBLINKS

[1] Geiger-Müller-Zählrohr: <https://de.wikipedia.org/wiki/Z%C3%A4hlrohr>

[2] Theremino Geiger-Adapter (siehe die drei letzten Entwürfe auf dieser Seite): [www.theremino.com/de/technical/schematics](http://www.theremino.com/de/technical/schematics)

[3] Dieses Projekt auf Elektor Labs: <https://bit.ly/3uZlqmU>



# CO<sub>2</sub>-Wächter

Ein DIY-Ansatz zur Überwachung der Luftqualität



Von Mathias Claußen, Ton Giesberts und  
Luc Lemmens (Elektor)  
Entwurf von Florian Schäffer

Hohe CO<sub>2</sub>-Konzentrationen können in Räumen mit schlechter Belüftung Probleme verursachen. Dieser CO<sub>2</sub>-Wächter bietet eine Lösung: Das Gerät warnt den Benutzer mit drei LEDs und einem akustischen Alarm, wenn es Zeit zum Lüften ist. Der Entwurf ist ausgesprochen nachbaufreundlich, besteht er doch durchweg aus bedrahteten Bauteilen; Gerberdateien für die Platine und die Gehäusefrontplatte können kostenlos heruntergeladen werden.

Dieser CO<sub>2</sub>-Wächter ist ein Selbstbauprojekt zur Überwachung der Raumluft und zur Bewertung der Luftqualität anhand der Kohlendioxidkonzentration (Indoor Air Quality, IAQ). Das Gerät verwendet einen nichtdispersiven Infrarotsensor (NDIR) und zeigt die Luftqualität in fünf Bereichen auf einer dreistelligen LED-Skala an. Darüber hinaus ertönt ein Alarm, wenn die CO<sub>2</sub>-Konzentration unzulässig hoch wird. Es besteht die Möglichkeit, die Messwerte (CO<sub>2</sub>-Wert und Temperatur) über WLAN an die IoT-Analyseplattform ThingSpeak zu übertragen und dort die Daten grafisch auszuwerten.

Es gibt viele Faktoren, die die Luftqualität beeinflussen, aber die CO<sub>2</sub>-Konzentration in einem Raum ist einer, den man fast sofort spürt. Eine zu hohe Konzentration dieses Gases führt zu Müdigkeit, Konzentrationsschwäche oder sogar Übelkeit, ganz zu schweigen von den dauerhaften negativen Auswirkungen bei permanenter und viel zu hoher Exposition. Letzteres kommt (hoffentlich!) in einer Wohnung oder einem Büro nur selten vor, aber dieser CO<sub>2</sub>-Wächter kann Ihnen im Fall der Fälle rechtzeitig anzeigen, dass Sie den Raum lüften sollten.



## Die Schaltung

Das Herzstück des Schaltplans in **Bild 1** ist MOD2, ein Wemos D1 Mini, ein ESP8266EX-basiertes WLAN-Modul mit 4 MB Flash-Programmspeicher. Es sammelt die Daten des CO<sub>2</sub>-Sensors MH-Z19C (MOD1) über einen Software-UART, steuert drei LEDs zur Anzeige des CO<sub>2</sub>-Gehalts (LED1...3) und dazu den akustischen Alarm in Form von Summer BZ1, wenn der CO<sub>2</sub>-Gehalt inakzeptabel hoch ist [15]. Der CO<sub>2</sub>-Wächter wird über den Micro-USB-Anschluss des Wemos-Moduls von einem gewöhnlichen 5-VDC-Steckernetzteil versorgt. Diese Versorgungsspannung wird über eine interne Diode (mit Sicherung) zu Pin 9 des MOD2 und an einen integrierten 3,3-V-Spannungsregler geleitet. Die Ausgangsspannung dieses LDO wird für die interne Stromversorgung des Wemos-Moduls verwendet und ist zusätzlich mit Pin 8 des Moduls verbunden. So können die 3,3 V auch für die Versorgung eines an K1 angeschlossenen Miniaturlüfters verwendet werden, wie wir gleich sehen werden. Der 5-V-Ausgang von MOD2 wird für die Stromversorgung des CO<sub>2</sub>-Sensors und für eine zusätzliche LED im Schalter S1 verwendet, die anzeigt, dass die WLAN-Verbindung aktiv ist. Auch diese Spannung kann über R6 an K1 als alternative Energiequelle für den Lüfter verwendet werden. Mit dem Taster S1 kann der CO<sub>2</sub>-Sensor veranlasst werden, um die automatische Kalibrierung zu starten.



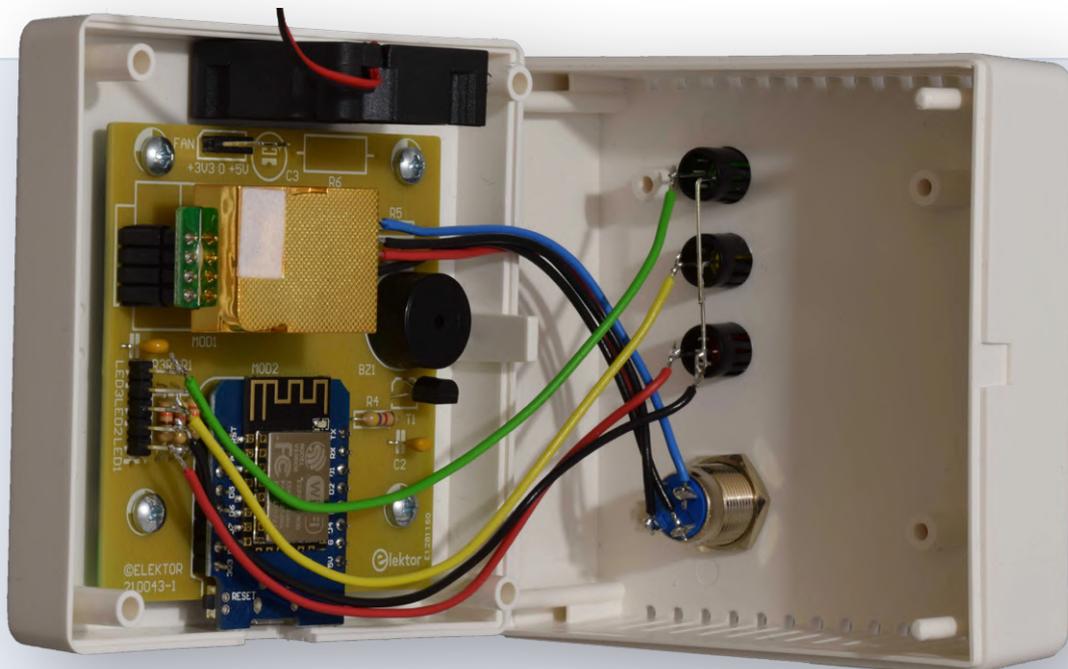


Bild 3. Der CO<sub>2</sub>-Wächter mit eingebautem Lüfter.

und die Stiftleisten (Buchsenleisten für den CO<sub>2</sub>-Sensor MOD1, optionale Stiftleisten für den Schalter und die LEDs). Um das D1-Mini-Modul (MOD2) zu montieren, löten Sie zunächst die beiden achtpoligen Stiftleisten an die Seite des Moduls, auf der sich auch die USB-Buchse und die meisten anderen Bauteile befinden (siehe **Bild 3**). Wir empfehlen, für MOD2 keine Buchsenleisten auf der Platine zu verwenden, da beim Einstecken und Herausziehen eine gewisse und unnötige Kraft auf das Modul einwirkt. Am besten ist es, die Stiftleisten des D1 Mini direkt in die Platine zu löten. Außerdem könnte, wenn man Buchsenleisten verwendet, der Luftstrom durch die Lüftungslöcher des Gehäuses behindert werden.

Wenn die Platine mit vier Schrauben an der unteren Hälfte des Gehäuses befestigt und der Lüfter eingeklebt ist (siehe unten), kann das Sensormodul in die entsprechenden Buchsen gesteckt werden. Berühren Sie niemals die Oberseite des Sensors, wovon auch im Datenblatt [2] gewarnt wird: *„Please avoid the pressure of its gilded plastic chamber from any direction, during welding, installation, and use. Especially never touch the air intake (white dust filter).“*

Die vier- und fünfpoligen Buchsen für den Sensor werden wahrscheinlich als eine 1x10-polige Buchsenleiste geliefert. Schneiden Sie diese so ab, dass Sie eine vier- und eine fünfpolige Buchsenleiste erhalten.

### Vorbereiten des Gehäuses

Verwenden Sie eine Fotokopie des Frontplattenentwurfs als Schablone für die korrekten Positionen der vier Löcher (drei LEDs und der Schalter) und benutzen Sie eine Reißnadel oder einen ähnlich spitzen Gegenstand, um die vier Bohrlöcher exakt zu markieren. Auf der Projektseite auf Elektor Labs [3] gibt es einige Tipps und Tricks zu lesen, um das Gehäuse beim Bohren größerer Löcher in die Kunststoff-Frontplatte nicht zu beschädigen. Achten Sie bei diesem Gehäusertyp besonders auf die vier langen Passer-Stifte, die bei unsachgemäßer Handhabung gerne brechen!

An einer Seite der unteren Gehäusenhälfte muss eine Aussparung für den Micro-USB-Anschluss des D1-Mini-Moduls angebracht werden, der als Stromversorgungseingang für den ganzen CO<sub>2</sub>-Wächter fungiert. Achten Sie darauf, dass die Aussparung groß genug für den USB-Ste-

cker Ihres Netzteils ist. Schrauben Sie die Platine provisorisch auf den Schraubdomen fest und stecken Sie das D1-Mini-Modul auf die Platine, um die richtige Stelle für die Aussparung zu markieren. Sie befindet sich leicht neben der Gehäusemitte.

Für den Lüfter müssen keine Lüftungsschlitze gebohrt oder gefräst werden; sie befinden sich schon im Gehäuse. Der Lüfter wird praktischerweise mit Sekunden- oder Epoxidkleber in die Unterschale des Gehäuses geklebt (Bild 3). Damit nichts klappert oder brummt, sollte der Lüfter die Oberseite des Gehäuses nicht berühren. Der Lüfter sollte die Luft *aus* dem Gehäuse blasen. Überprüfen Sie vor der Montage die Richtung des Luftstroms! Meist ist auf dem Lüfterrahmen ein kleiner Pfeil zu sehen, der die Lüftungsrichtung anzeigt, wenn nicht, legen Sie probeweise eine Spannung von 3,3...5 V an die Anschlussdrähte. Achten Sie dabei aber auf die richtige Polarität, denn moderne Lüfter besitzen oft eine interne Schaltung, die bei verpolter Spannung Schaden nehmen kann.

### Anschließen der Frontplatten-Bauteile

Für den Anschluss der LEDs werden nur vier Drähte benötigt, da die Masseverbindung von der Platine zu den drei LEDs durchgeschleift wird. Verbinden Sie, wie im Foto zu sehen, einfach die drei LED-Kathoden und verwenden Sie ein einziges Kabel.

Achten Sie darauf, dass keine Drähte auf der Antenne von MOD2 liegen, wenn der Gehäusedeckel schließlich auf der unteren Schale befestigt wird. Halten Sie die Kabel auch so weit wie möglich vom Wemos-Modul entfernt, aber machen Sie sie nicht länger als nötig. Die Drähte zum Taster können bis zu 8,5 cm lang sein und die Drähte zu den LEDs etwa 12 cm. Mit so bemessenen Kabellängen können Sie noch bequem den oberen Gehäuseteil abnehmen und ihn auf die Seite neben den Boden legen. Zum Verschließen des Gehäuses werden keine Schrauben benötigt, weil die vier langen Kunststoff-Passerstifte des Deckels gut in den langen Montagebuchsen feststecken.

Natürlich können Sie die Frontplatte auch nach Ihren eigenen Vorstellungen gestalten. Wenn Sie einen Taster und eine separate (blaue) LED verwenden, müssen Sie darauf achten, dass LEDs und Taster den Modulen nicht zu nahe kommen.

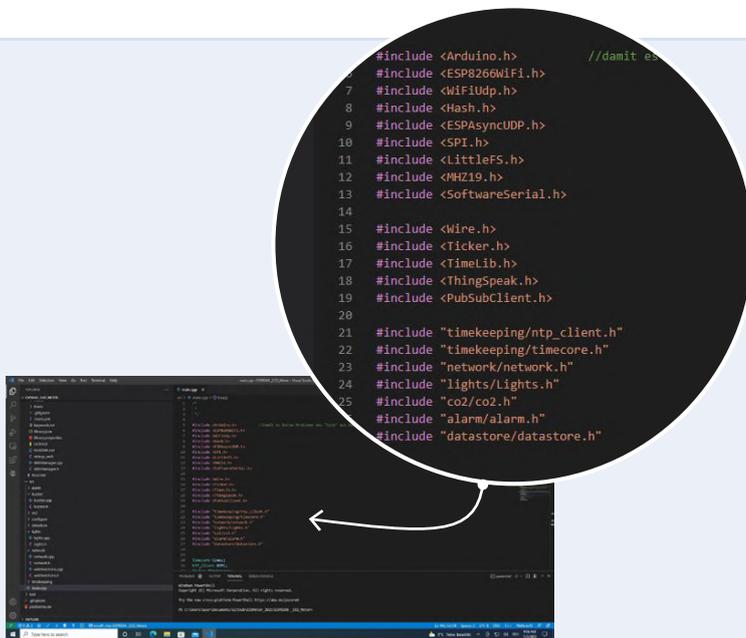


Bild 4. Der Code in der PlatformIO-IDE.

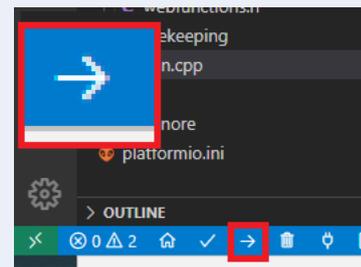


Bild 5. Der Knopf für den Firmware-Upload.

## Firmware auf den Wemos laden

Die Firmware für den ESP8266 auf dem Wemos-Modul wird mit dem Arduino-Framework geschrieben. Sie benötigen den Sketch selbst und einige Bibliotheken. Für die Entwicklung des Programms für den CO<sub>2</sub>-Wächter kamen und kommen PlatformIO und Visual Studio Code zum Einsatz. Bibliotheken, Projekt und Upload-Einstellungen sind kombiniert, so dass das Kompilieren des Codes kein großes Problem darstellen sollte. Um Visual Studio Code und PlatformIO zu installieren, sollten Sie der Anleitung [4] von PlatformIO folgen. Nach der Installation müssen Sie sich den Quellcode aus dem GitHub-Repository von Elektor holen. Wenn Sie sich nicht sicher sind, wie Sie den Code bekommen, schauen Sie sich das Video an, das unser Kollege Clemens Valens [5] erstellt hat. Wenn Sie mehr über GitHub und die Verwendung von Repositories wissen möchten, schauen Sie sich das Elektor-Webinar auf GitHub [6] an. Nachdem Sie den Code mit der PlatformIO-IDE geöffnet haben, sollte er so aussehen wie in **Bild 4**. Schließen Sie das Wemos-Modul an Ihren Computer an und klicken Sie auf die Upload-Schaltfläche (**Bild 5**). PlatformIO lädt alle erforderlichen Bibliotheken, Tools und Board-Support-Pakete herunter und beginnt mit dem Kompilieren und anschließenden Hochladen.

PlatformIO hat zwar nun die Firmware erstellt und geflasht, doch wie bei den meisten ESP8266-Projekten gibt es auch hier einen zweiten Teil. Da die Firmware die Konfiguration über einen Webserver anbietet, müssen wir in diesem zweiten Schritt auch die Webseiten auf den ESP8266 schreiben. Klicken Sie in Visual Studio Code auf den kleinen Ameisenkopf, sollte eine Liste von Tasks erscheinen (**Bild 6**). Klicken Sie zunächst auf *Build Filesystem Image* und anschließend auf *Upload Filesystem Image*. Dadurch wird der Inhalt für den Webserver auf den ESP8266 geschrieben. Wenn Sie Fehlermeldungen wie „can't access COM port“ erhalten, sollten Sie überprüfen, ob nicht ein serielles Terminal geöffnet ist, das den Port blockiert. Danach kann die Firmware konfiguriert und verwendet werden.

## WLAN- und Cloud-Konnektivität

Wenn Ihr ESP8266 auf dem Wemos-Modul keine Verbindung zu einem WLAN herstellen kann, startet er einen Access-Point namens

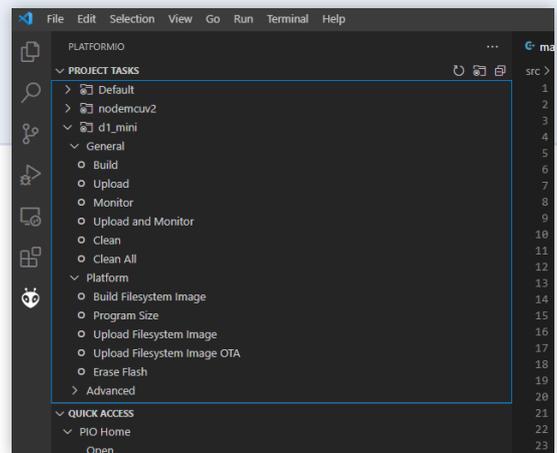


Bild 6. Der Ameisenkopf links legt weitere Optionen frei.



Bild 7. Hier werden die WLAN-Einstellungen vorgenommen.

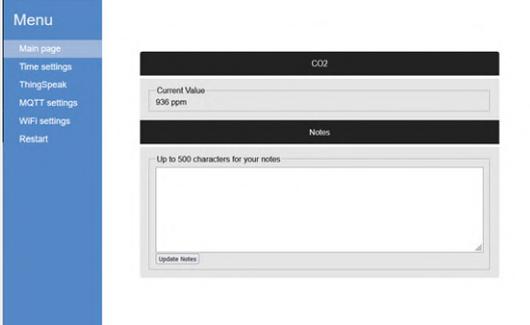


Bild 8. Das Web-Interface des CO<sub>2</sub>-Wächters.

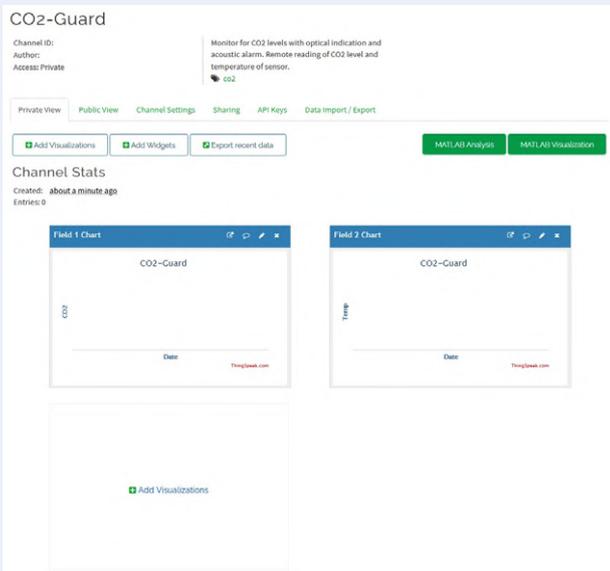


Bild 10. Ein neu erstellter ThingSpeak-Kanal.

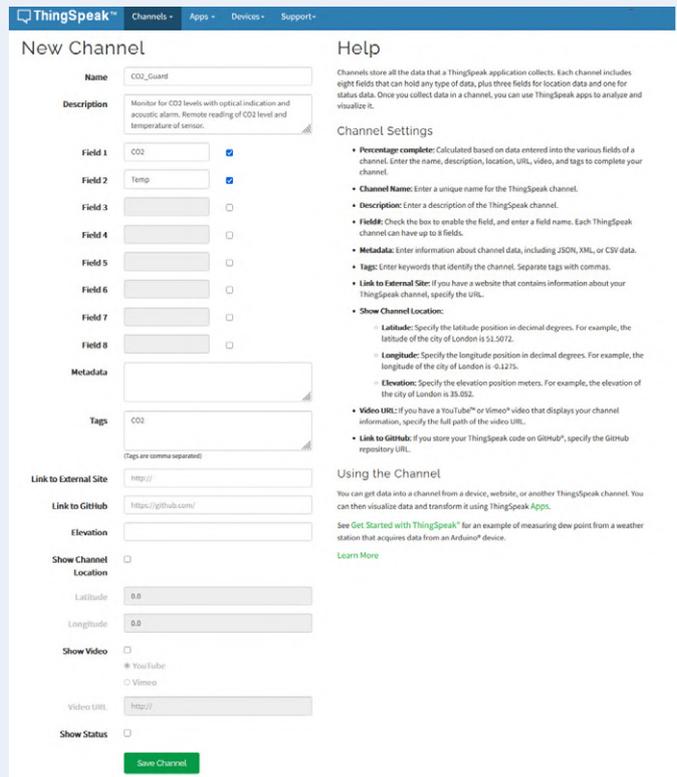


Bild 9. Hier werden die ThingSpeak-Einstellungen vorgenommen.

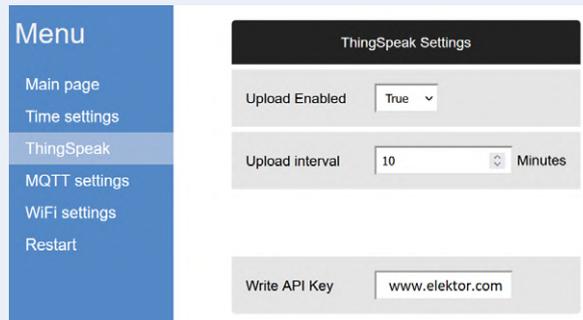


Bild 11. Weitere ThingSpeak-Einstellungen.

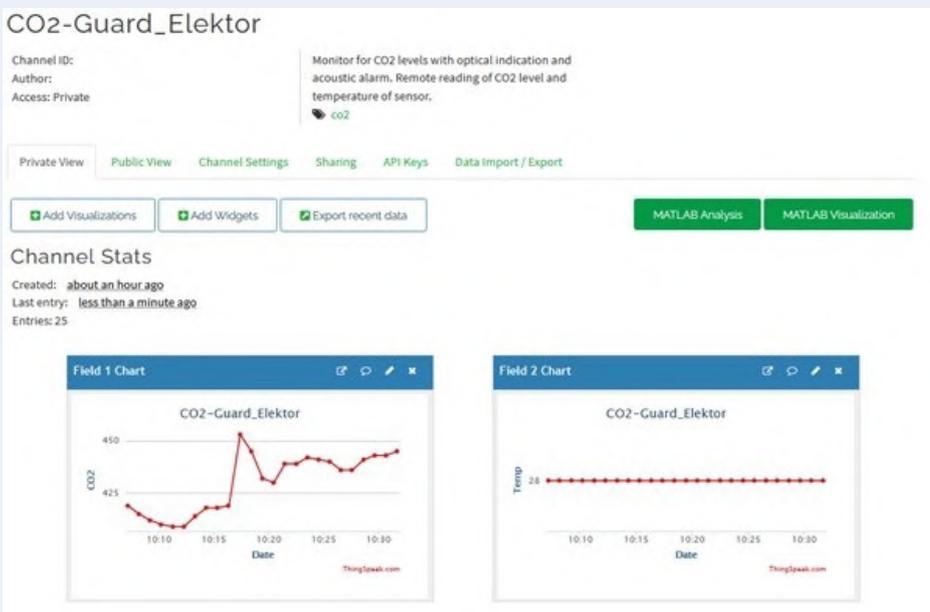


Bild 12. Die Daten treffen ein!





„CO<sub>2</sub> Meter“. Verbinden Sie sich mit diesem und öffnen Sie in einem Browser die Adresse <http://192.168.4.1>. Sie sollten die Konfigurationsoberfläche des WLAN-Managers sehen (**Bild 7**). Klicken Sie auf **Configure WiFi**, wählen Sie Ihr Netzwerk aus und klicken auf **Save**. Der ESP8266 wird nun versuchen, sich mit Ihrem Netzwerk zu verbinden. Wenn er eine Verbindung herstellen kann, können Sie innerhalb Ihres Netzwerks <http://co2guard.local> verwenden, um auf die Weboberfläche zuzugreifen (**Bild 8**). Von hier aus beginnt die Konfiguration des CO<sub>2</sub>-Wächters.

Wenn Sie möchten, können Sie den CO<sub>2</sub>-Wächter mit einem webbasierten Datenspeicher wie ThingSpeak oder einem MQTT-Broker, etwa dem aus Node-RED, verbinden. Mit einem MQTT-Broker können Sie den Sensor noch weiter automatisieren, indem Sie zum Beispiel eine Nachricht an Ihr Mobiltelefon senden, wenn der CO<sub>2</sub>-Gehalt einen bestimmten Wert überschreitet. Um mit ThingSpeak loszulegen, gibt es nun eine kurze Tour durch die Einrichtung des Kontos.

### ThingSpeak: Ein MathWorks-Konto erstellen

Wenn Sie nur die aktuelle CO<sub>2</sub>-Konzentration in dem Raum, in dem Sie sich gerade aufhalten, überwachen wollen, brauchen Sie keine WLAN-Verbindung und schon gar kein ThingSpeak-Konto, denn Sie müssen nur auf die LEDs schauen und im Extremfall auf den Summer hören, um einen klaren Hinweis auf die Luftqualität zu erhalten, zumindest was CO<sub>2</sub> betrifft. Wenn Sie aber beispielsweise den Verlauf dieser Messungen protokollieren und die Daten über die Zeit analysieren wollen oder aus größerer Entfernung ablesen möchten, wie hoch die aktuelle Konzentration ist, dann bietet sich eine Verbindung über ein WLAN zu ThingSpeak an. Sie können dann sogar die Temperatur verfolgen, die zwar ebenfalls vom Sensormodul gemessen wird, aber nicht auf dem CO<sub>2</sub>-Wächter angezeigt wird.

Um einen Online-Kanal für Ihre Messungen zu erstellen, gehen Sie auf die ThingSpeak-Website [7] und wählen Sie „Get Started For Free“. Auf der nächsten Seite können Sie ein MathWorks-Konto erstellen (falls Sie noch keines haben). Danach erstellen Sie einen neuen Kanal und geben die erforderlichen Daten ein, wie in **Bild 9** vorgeschlagen. Klicken Sie auf **Save Channel**. Eine Übersicht über diesen damit erstellten Kanal ist in **Bild 10** zu sehen.

Auf der Registerkarte **API Keys** müssen Sie den **Write API Key** für die spätere Verwendung notieren. Dieser Key ist „top secret“ und Sie sollten ihn entsprechend vertraulich behandeln. Rufen Sie nun das Webinterface des CO<sub>2</sub>-Wächters auf und wählen Sie die Option **ThingSpeak** im Seitenmenü. Es wird eine Seite mit den Einstellungen für ThingSpeak angezeigt. Geben Sie den API-Schlüssel ein, wählen Sie ein Upload-Intervall und aktivieren Sie ThingSpeak (**Bild 11**). Gehen Sie zurück auf die MathWorks-ThingSpeak-Seite. Unter dem Menüpunkt **Channels** wählen Sie **My Channels** und dann den Reiter **Private View**. Nach kurzer Zeit sollten wie von Geisterhand geschrieben die ersten Messwerte angezeigt werden (**Bild 12**).

Um Ihre Daten mit anderen Menschen zu teilen, können Sie einzelne Benutzer mit Anzeigeberechtigung hinzufügen, oder Sie können den Kanal öffentlich machen, so dass jeder Ihre Daten sehen kann. Einzelne Benutzer erhalten eine E-Mail an die Adressen, die Sie auf der Registerkarte **Sharing** eingeben können. Wenn Sie einen kostenlosen ThingSpeak-Account haben, ist dieses Sharing von Daten auf nur drei andere Nutzer beschränkt. Weitere Details zum Teilen von Daten finden Sie im MathWorks-Hilfe-Center [8].

### MQTT-Verbindung

**Message Queuing Telemetry Transport** (MQTT) ist eine vielseitige Möglichkeit, Ihre Daten an andere Systeme zu verteilen, die sie speichern oder verarbeiten sollen und können. Sie benötigen keinen externen Dienst, um Daten zu sammeln oder zu verarbeiten, und Sie können die Daten innerhalb Ihres eigenen Netzwerks halten. Eines der Tools, das MQTT-Konnektivität bietet, ist Node-RED, das besonders beliebt auf dem Raspberry Pi ist. Wir haben in der Vergangenheit viel über die Installation von Node-RED berichtet, und wenn Sie eine schnelle Installationsanleitung für Ihren Raspberry Pi suchen, schauen Sie auf der Node-RED-Website [9] nach. Für MQTT können Sie in der Weboberfläche des CO<sub>2</sub>-Wächters einen Server und ein Topic angeben, in dem die Daten veröffentlicht werden sollen (**Bild 13**). Die Daten selbst werden als JSON-String mit den Objekten **CO<sub>2</sub>Value** und **Temp** übertragen, die den CO<sub>2</sub>-Wert in ppm und die Temperatur in °C darstellen. Wenn Sie mehr über Node-RED mit einem Raspberry Pi lesen möchten, blättern Sie einmal im Buch von Dogan Ibrahim nach, *Programming with Node-RED* (Elektor 2020) [10].

### KiCad-Dateien für den Platinenentwurf

Die Platine wurde ursprünglich in Altium entworfen, und die Gerber-Dateien in [1] basieren auf diesem bewährten Design. Der Entwurf ist recht einfach, wie man am Schaltplan erkennen kann, so dass wir ihn kurzerhand mit KiCad neu erstellt haben. Neben dem Schaltplan sollte auch die Platine übertragen werden, aber als das Projekt in der letzten Stable-Version von KiCad 5 begonnen wurde, gab es keinen nativen (integrierten) Import für Altium-Platinendateien. Stattdessen haben wir die ursprünglichen Gerber-Dateien verwendet und sie in KiCad importiert. Der Gerber-Viewer in KiCad ist nämlich in der Lage, Platinen-Dateien zu erzeugen, die im PCB-Layout-Tool verwendet werden können. Damit erhält man nicht nur die Umriss der Leiter-

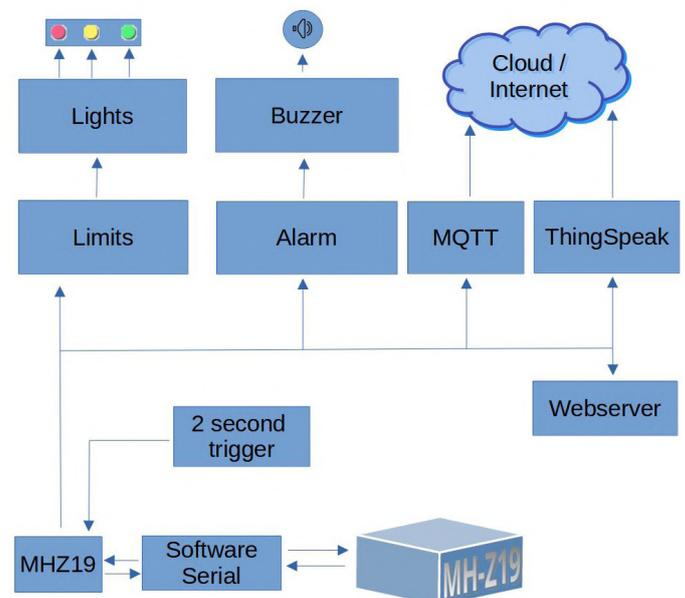


Bild 13. Einstellungen für den Message Queuing Telemetry Transport.



bahnen, sondern auch die Footprints und Befestigungslöcher, die man für die Platzierung der Bauteile verwenden kann. Es war zwar „zur Not“ möglich, Platinenlayouts auf diese Weise in KiCad zu importieren, aber es wäre vorzuziehen, mit der KiCad-Version 6 den nativen Altium-Board-Import zu verwenden.

Aber warum sollte man diese Konvertierung überhaupt durchführen? Erstens können so andere direkt mit den KiCad-Dateien arbeiten und sie auf Wunsch ändern. Und zweites ist es bei einigen PCB-Herstellern einfach möglich, eine KiCad-Platinendatei per Drag & Drop zu übertragen, um die Platine in Auftrag zu geben (zum Beispiel bei Aisler [11]). Sie müssen so nicht erst nach Gerber- und Bohrdateien exportieren, um eine ZIP-Datei zu erzeugen. Wenn Ihr PCB-Hersteller tatsächlich Gerber- und Bohrdateien benötigt, können Sie diese trotzdem in KiCad nach den von Ihrem Service geforderten Spezifikationen erstellen. Wenn Sie Ihr eigenes Gehäuse um die Leiterplatte herum entwerfen möchten, hat sich die Kombination von KiCad und FreeCad als sehr gut erwiesen, wie Sie an dem Weihnachtskamin-Projekt [12] sehen können. Der 3D-Druck eines Gehäuses kann eventuell bequemer sein als das Bohren und Feilen von Löchern in ein vorgefertigtes Gehäuse. Und wenn Sie zu Hause einen 3D-Drucker haben, der nur noch Staub ansetzt, warum ihn nicht nutzen? Außerdem bedeutet dies, dass man ein Bauteil weniger bestellen muss. Das 3D-druckbare Gehäuse ist zum Zeitpunkt der Erstellung dieses Artikels zwar noch nicht fertig, aber wir arbeiten eifrig daran.

## Bauen Sie Ihren eigenen CO<sub>2</sub>-Wächter

Der Bau eines CO<sub>2</sub>-Wächters für Ihr Zuhause oder Ihren Arbeitsplatz ist gar nicht so schwer, und wie Sie wissen, wurden bereits viele CO<sub>2</sub>-Messgeräte sowohl als Heimwerker- als auch als kommerzielle Projekte entwickelt. Mit dem MH-Z19-Sensor aus diesem Projekt haben Sie einen kostengünstigen Messaufnehmer. Der ESP8266 ist ein bewährter WLAN-Mikrocontroller, der nicht nur mit dem Arduino-Framework programmiert werden kann, sondern in der aktuellen

Kombination auch Unterstützung für den Home Assistant bietet. Möchten Sie Ihr eigenes Gerät bauen? Für den Sensor müssen Sie rund 30 € ausgeben (in Europa). Der Taster, der Wemod D1 mini, die Platine, die LEDs, das Gehäuse, der Lüfter und der Versand kommen noch hinzu, so dass Sie am Ende auf etwa 60 € für die Bauteile kommen. Wenn Sie die CO<sub>2</sub>-Konzentration in Ihren Räumen überwachen und kennen und entsprechend lüften, macht sich das Gerät schnell bezahlt! ◀

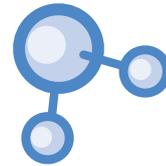
210043-02

### Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).

### Ein Beitrag von

Ursprünglicher Hardware-Entwurf: **Florian Schäffer**  
Entwurf und Text: **Mathias Claußen, Ton Giesberts, Luc Lemmens**  
Redaktion: **Jens Nickel, C. J. Abate**  
Übersetzung: **Rolf Gerstendorf**  
Layout: **Harmen Heida**



## Software

In der Software, die der Entwickler ursprünglich auf der Elektor-Labs-Plattform bereitstellte, gab es einige verbesserungswürdige Bereiche. So funktionierten die (fest kodierten) Zugangsdaten für das WLAN, mit dem das Gerät verbunden ist, nur, wenn sich Netzwerknamen oder Passwort nie ändern. Außerdem ist es nicht die sicherste Art, die Anmeldedaten irgendwo in irgendwelchen Programmdateien zu speichern. Das Gleiche gilt für den API-Schlüssel von ThingSpeak. Wenn Sie nämlich diesen Schlüssel ändern wollen, bedeutet das, dass Sie den gesamten Code neu kompilieren und auf das Gerät hochladen müssen.

Einiges wurde jedoch hartkodiert belassen (als `#defines` in `/configure/configure.h`), nämlich die Pin-Belegungen sowie die CO<sub>2</sub>-Schwellenwerte für die drei LEDs und den Summer. In der Datei `/configure/configure.h` haben Sie aber all diese hartkodierten Einstellungen an einem Ort und können sie bei Bedarf übersichtlich ändern. Wenn Sie irgendwelche Einstellungen in dieser Datei ändern, müssen Sie natürlich die Firmware neu kompilieren und auf den ESP8266 hochladen.

Innerhalb der Software ist der Datenfluss recht einfach. Alle zwei Sekunden werden die Sensorwerte abgefragt und an verschiedene Programmmodule geschickt. Für die lokale Überwachung des CO<sub>2</sub>-Gehalts haben wir die `limits`-Klasse, die letztlich die drei LEDs steuert, und die `alarm`-Klasse, die bei Überschreiten eines bestimmten CO<sub>2</sub>-Gehalts den Summer ertönen lässt. Diese Klassen sind generisch aufgebaut, so dass es einfach ist, die Anordnung der LEDs zu ändern, um eine andere Art der Wertanzeige zu erhalten. Das Gleiche gilt für den Summer. Wenn Sie einen aktiven Summer oder sogar ein Soundmodul verwenden möchten, ist diese Klasse flexibel genug, um dies zu ermöglichen.

Für die Cloud-Anbindung wurden, wo immer möglich, vorgefertigte Bibliotheken verwendet, insbesondere für ThingSpeak. Das macht den Datenaustausch mit diesem Dienst viel einfacher als bei der ESP32-Wetterstation [13], die wir vor einigen Jahren veröffentlicht haben [14].



## STÜCKLISTE

### Widerstände:

R1,R2,R3 = 330 Ω, 0,25 W, 5 %

R4 = 4k7, 0,25 W, 5 %.

R5 = 0 Ω (Draht), siehe Text

R6 = 0 Ω (Draht für 5 V oder offen lassen), siehe Text

R7 = 1k5, 0,25 W, 5 %

### Kondensatoren:

C1,C2 = 100 n, 50 V, 10 %, X7R, Raster 5 mm

C3 = 100 µ, 25 V, optional, nicht montiert, siehe Text

### Halbleiter:

D1 = BAT85, DO-35

T1 = BC327, PNP, TO-92

LED1 = LED rot, 5 mm

LED2 = LED gelb, 5 mm

LED3 = LED grün, 5 mm

### Außerdem:

MOD1 = 1x10-polige Buchsenleiste, aufgeteilt in 1x4 und 1x5 (siehe Text)

MOD1 = MH-Z19C, Version 400-5000 ppm, Pins auf der Unterseite

MOD2 = Wemos D1 mini mit zwei 1x8-poligen Stiftleisten

S1 = Taster mit eingebauter LED (1xan, „Klingeltaster“)

Gehäuse: Supertronic PP73BL

BZ1 = DC-Summer, 5 V, Raster 5 mm oder 7,62 mm, Ø<sub>a</sub> 14 mm max.

20 cm Schrumpfschlauch, Ø<sub>i</sub> 2,4 mm

3 Fassungen für 5-mm-LEDs (LED1..3), schwarz

Lüfter 5 VDC, 30x30x10,5 mm

1x12-polige Stiftleiste

Platine 210043-1 v2.0

### Dünne Litze (0,25 mm<sup>2</sup>)

#### zum Anschluss von LEDs und Schalter:

Litze, schwarz (2x8,5 cm und 12 cm)

Litze, rot (8,5cm und12 cm)

Litze, gelb (12 cm)

Litze, grün (12 cm)

Litze, blau (8,5 cm)



## PASSENDE PRODUKTE

- WeMos D1 mini Pro – ESP8266-basiertes WLAN-Modul (SKU 19185) [www.elektor.de/19185](http://www.elektor.de/19185)
- Bestückte Platine mit Bauteilen (ohne WeMos D1 und MH-Z19) [www.elektormagazine.de/aisler/co2guard](http://www.elektormagazine.de/aisler/co2guard)
- H. Henrik Skovgaard, IoT Home Hacks with ESP8266 Buch (SKU 19158) [www.elektor.de/19158](http://www.elektor.de/19158)  
E-Buch (SKU 19159) [www.elektor.de/19159](http://www.elektor.de/19159)
- D. Ibrahim, Programming with Node-RED E-Buch (SKU 19225) [www.elektor.de/19225](http://www.elektor.de/19225)



## WEBLINKS

- [1] Elektor GitHub-Repository: <https://github.com/ElektorLabs/210043-CO2-Guard>
- [2] Datenblatt des Winsen-Sensors MHZ-19: [www.winsen-sensor.com/d/files/infrared-gas-sensor/mh-z19b-co2-ver1\\_0.pdf](http://www.winsen-sensor.com/d/files/infrared-gas-sensor/mh-z19b-co2-ver1_0.pdf)
- [3] „CO2 Guard“, Elektor-Labs.com: [www.elektormagazine.de/labs/co2-guard](http://www.elektormagazine.de/labs/co2-guard)
- [4] PlatformIO-Installationsanleitung: <https://platformio.org/install/ide?install=vscod>
- [5] „How to Code (Download) or Clone Something From GitHub“, Elektor TV: [www.youtube.com/watch?v=X5e3xQBeqf8](http://www.youtube.com/watch?v=X5e3xQBeqf8)
- [6] „GitHub Webinar: My Way Into GitHub“, Elektor TV: [www.youtube.com/watch?v=j\\_LgvVhBdwQ](http://www.youtube.com/watch?v=j_LgvVhBdwQ)
- [7] MathWorks ThingSpeak: <https://thingspeak.com>
- [8] Hilfe für MathWorks ThingSpeak: <https://bit.ly/ChannelPropertiesMATLAB>
- [9] Installationsanleitung Node-RED: <https://nodered.org/docs/getting-started/raspberrypi>
- [10] D. Ibrahim, Programming with Node-RED, Elektor 2020: [www.elektor.de/programming-with-node-red-e-book](http://www.elektor.de/programming-with-node-red-e-book)
- [11] AISLER B.V: <https://aisler.net>
- [12] M. Claußen, „Der Weihnachtskamin - 3D-Puzzle aus Platintenteilen“, ElektorMag: <https://bit.ly/36mGlkj>
- [13] „ESP32 Weather Station“, Elektor-Labs.com: [www.elektormagazine.de/labs/esp32-weather-station-180468](http://www.elektormagazine.de/labs/esp32-weather-station-180468)
- [14] R. Aarts, „Wetterstation mit ESP32“, Elektorlabs 1/2019: [www.elektormagazine.de/magazine/elektor-69/42263](http://www.elektormagazine.de/magazine/elektor-69/42263)
- [15] „Gesundheitliche Bewertung von Kohlendioxid in der Innenraumluft“, Umweltbundesamt, 2008: <https://bit.ly/3v3bfXN>

# MonkMakes: Luftqualitäts-Messgerätekit für den Raspberry Pi

## Misst Temperatur und CO<sub>2</sub>e



Von Luc Lemmens (Elektor)

Weil die meisten von uns heutzutage viel in (privaten) Zimmern sitzen, werden Module zur kostengünstigen Messung der Luftqualität immer beliebter. Das *MonkMakes Air Quality Kit* misst den CO<sub>2</sub>e-Gehalt und die Temperatur. Es ist speziell für den Raspberry Pi 400 konzipiert, kann aber dank der mitgelieferten Jumper-Drähte und einer GPIO-Schablone auch an andere Raspberry-Pi-Modelle angeschlossen werden.

Bekanntermaßen misst man mit Thermometern die Raumtemperatur, aber in den letzten Jahren sind CO<sub>2</sub>(e)-Messgeräte zur Überwachung der Luftqualität immer beliebter geworden. Zu viel Kohlendioxid (CO<sub>2</sub>) wirkt sich negativ auf die Konzentrationsfähigkeit aus, und bei noch höheren Werten ist es sogar gesundheitsschädlich. Dieses Kit misst die Qualität der Raumluft sowie die Temperatur. Es ist vorrangig für den Raspberry Pi gedacht, kann aber auch als eigenständiges Gerät

eingesetzt werden. Das Board verfügt über einen Summer und eine Leiste mit sechs LEDs (zwei grüne, zwei orange und zwei rote), die die Luftqualität anzeigen. Die Messwerte für Temperatur und Luftqualität können von einem Raspberry Pi verarbeitet werden, der ebenfalls den Summer und die LED-Anzeige ansteuern kann.

Der Bausatz wird ohne gedruckte Dokumentation geliefert, aber er enthält einen Link zur MonkMakes-Website, wo das Datenblatt und die Anleitung heruntergeladen werden können [1]. Diese Dokumente enthalten alle relevanten Informationen und helfen dem Benutzer, das Board anzuschließen und zu verwenden. Beispielanwendungen in Python stehen auf Github [2] zum Download bereit.

### Die Hardware

Neben der Anzeige mit sechs LEDs und dem großen quadratischen Summer in der Mitte der Platine in **Bild 1** enthält die Platine eine Power-LED, einen Temperatursensor, einen eCO<sub>2</sub>-Sensor, einen Mikrocontroller und natürlich einen 40-poligen Steckverbinder, der direkt auf den Erweiterungsstecker eines Raspberry Pi 400 passt (**Bild 2**). Vielleicht überflüssig zu sagen: Andere Raspberry-Pi-Boards können nicht direkt angeschlossen werden, dafür sind Jumper-Drähte im Bausatz enthalten. Die vier benötigten Anschlüsse (zwei für die Stromversorgung, zwei für die serielle Verbindung) sind im Aufdruck auf der MonkMakes-Platine und auf der mitgelieferten Schablone so dargestellt, dass sie zu den entsprechenden Pins des GPIO-Anschlusses des Raspberry Pi

passen, wie in **Bild 3** dargestellt. Die Power-LED leuchtet auf, sobald die 3,3-V-Versorgungsspannung eingeschaltet wird, ebenso wie eine der CO<sub>2</sub>e-Pegel-LEDs.

Der Temperatursensor TMP235 stammt von Texas Instruments [3]. Seine Ausgangsspannung ist proportional zur Temperatur. Für die CO<sub>2</sub>-Messung verwendet das MonkMakes-Board einen TVOC-Sensor (Total Volatile Organic Compounds) des Typs CCS811 [4], der nicht direkt den CO<sub>2</sub>-Gehalt misst, sondern den Gehalt einer Gruppe von Gasen, die als flüchtige organische Verbindungen (VOCs) bezeichnet werden. In Innenräumen steigt der Gehalt dieser Gase in einer Kurve an, die mit der von CO<sub>2</sub> vergleichbar ist, so dass die Sensorwerte auch zur Abschätzung des CO<sub>2</sub>-Gehalts verwendet werden kann. Dies wird als CO<sub>2</sub>-Äquivalent oder CO<sub>2</sub>e bezeichnet.

Der eingebaute ATtiny1614-Mikrocontroller liest beide Sensoren aus und steuert die LED-Balkenanzeige und den Summer. Über ein serielles Protokoll kann ein Host-System wie der Raspberry Pi die Sensorwerte abfragen oder die LEDs und den Summer ein- und ausschalten. Das Datenblatt des Kits dokumentiert das einfache Protokoll, so dass es nicht allzu schwierig sein dürfte, eine eigene Software zur Unterstützung des Air Quality Kits zu schreiben. Wie der Name des Kits schon sagt, ist es für den Raspberry Pi konzipiert, aber es gibt keinen Grund, warum man es nicht mit anderen Boards oder Systemen mit einem 3,3-V-UART verwenden könnte.

Die Firmware des ATtiny bietet auch einen automatischen Modus, der standardmäßig eingeschaltet ist. In diesem Modus wird der CO<sub>2</sub>e-Pegel ohne eine externe Steuerung auf der LED-Leiste angezeigt. Alles, was in diesem Modus dann zusätzlich zum Kit benötigt wird, ist eine 3,3-V-Stromversorgung. So kann das Air Quality Kit auch ohne Host-System als CO<sub>2</sub>e-Monitor verwendet werden.

## Software

Wie bereits erwähnt, stehen auf der Github-Seite von MonkMakes einige Python-Beispielprogramme zur Verfügung, mit denen das Air Quality Kit gesteuert werden kann, um alle seine Funktionen zu testen und zu demonstrieren. Im Abschnitt *Getting Started* der Dokumentation zeigen die Anweisungen deutlich, wie die Software auf einem Raspberry-Pi-Board verwendet werden kann, um das Kit in ein CO<sub>2</sub>e-Messgerät, in ein CO<sub>2</sub>e-Messgerät mit akustischem Alarm (**Bild 4**) und in einen Datenlogger zu verwandeln. Wenn man sich die Python-Beispiele (**Bild 5**) ansieht, wird man feststellen, dass der ATtiny und die API das Abrufen und die Auswertung der Sensordaten vollständig übernehmen. Ein einfacher Befehl vom Host (Raspberry Pi) veranlasst das Air Quality Board, die aktuelle Umgebungstemperatur (in °C) beziehungsweise den CO<sub>2</sub>e-Gehalt (in ppm) auszugeben. Ähnliche Befehle werden verwendet, um den Summer ein- und auszuschalten und die LEDs der CO<sub>2</sub>e-Balkenanzeige zu steuern.

## Ein schönes Design

Um dieses Luftqualitäts-Kit in Betrieb zu nehmen, sind nur einige wenige Grundkenntnisse über den Raspberry Pi erforderlich. Was für die einen ein großer Vorteil ist, dürfte für die anderen weniger attraktiv sein: Kenntnisse über die Sensoren und die Steuerung von Buzzer und LEDs sind nicht erforderlich. Zudem ist der Quellcode der Firmware des ATtiny1614 nicht veröffentlicht, was heißt, dass wir nicht

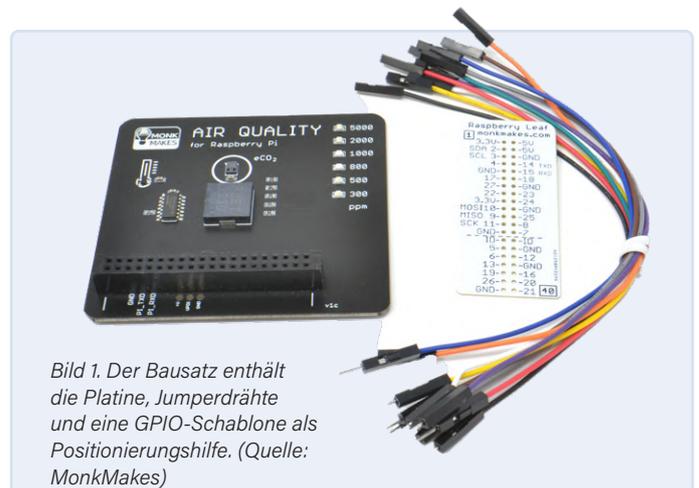


Bild 1. Der Bausatz enthält die Platine, Jumperdrähte und eine GPIO-Schablone zur Positionierungshilfe. (Quelle: MonkMakes)



Bild 2. Air Quality Kit, angeschlossen an einen Raspberry Pi 400. (Quelle: MonkMakes)

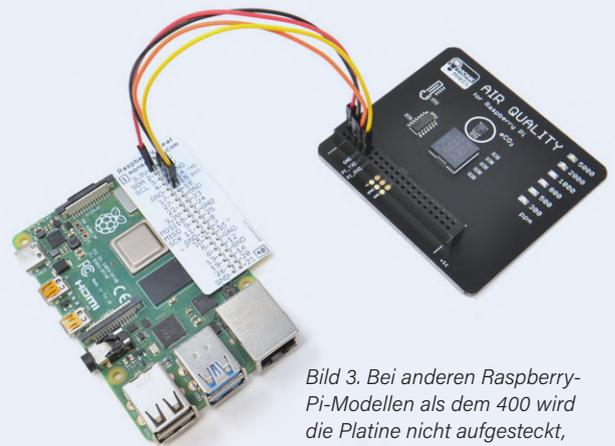


Bild 3. Bei anderen Raspberry-Pi-Modellen als dem 400 wird die Platine nicht aufgesteckt, sondern mit Jumper-Drähten angeschlossen. (Quelle: MonkMakes)



Bild 4. Raspberry-Pi-Displayausgabe für eines der Beispielprogramme. (Quelle: MonkMakes)

```

1 import threading
2 import time
3 from guizero import App, Text
4 from aq import AQ
5
6 aq = AQ()
7
8 app = App(title="Air Quality", width=550, height=300, layout="grid")
9
10 def update_readings(): # update fields with new temp and eCO2 readings
11     while True:
12         temp_c.field.value = str(aq.get_temp())
13         eco2.field.value = str(aq.get_eco2())
14         time.sleep(0.5)
15
16 t1 = threading.Thread(target=update_readings)
17 t1.start() # start the thread that updates the readings
18
19 aq.leds_automatic()
20
21 # define the user interface
22 Text(app, text="Temp (C)", grid=[0,0], size=20)

```

Bild 5. Der Python-Quelltext zeigt, dass nur einfache, kurze Anweisungen erforderlich sind, um mit dem Air Quality Kit zu kommunizieren.

wissen, was genau in diesem Mikrocontroller passiert. Das Protokoll zur Kommunikation mit dem Board ist jedoch einfach und gut dokumentiert und die Entwicklung eigener Anwendungen - auch für andere Zielsysteme als Raspberry-Pi-Boards - dürfte relativ einfach sein. Das Air Quality Kit für den Raspberry Pi von MonkMakes ist ein schön gestaltetes, gut dokumentiertes Board, das zusammen mit den Beispielen auch für Anfänger geeignet ist, die sich mit Luftqualitätsmessungen beschäftigen wollen.

210681-02

### Haben Sie Fragen oder Kommentare?

Haben Sie Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an den Autor unter [luc.lemmens@elektor.com](mailto:luc.lemmens@elektor.com) oder kontaktieren Sie Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).

### Ein Beitrag von

Text: Luc Lemmens  
 Illustrationen: MonkMakes, Luc Lemmens  
 Redaktion: Jens Nickel, C. J. Abate  
 Übersetzung: Rolf Gerstendorf  
 Layout: Harmen Heida



### PASSENDE PRODUKTE

- > **MonkMakes Air Quality Kit for Raspberry Pi (SKU 19913)**  
[www.elektor.de/19913](http://www.elektor.de/19913)
- > **Raspberry Pi 400 - Raspberry Pi 4-based PC (US) + FREE GPIO Header (SKU 19429)**  
[www.elektor.de/19429](http://www.elektor.de/19429)
- > **Raspberry Pi 4 B (1 GB RAM) (SKU 18966)**  
[www.elektor.de/18966](http://www.elektor.de/18966)

## Eigenschaften des Sensors

CO <sub>2</sub> e-Mindestwert	400	ppm
CO <sub>2</sub> e-Höchstwert	4095	ppm
CO <sub>2</sub> e-Auflösung	1	ppm
CO <sub>2</sub> e-Fehler	nicht angegeben	
Temperatur minimaler Messwert	-10	°C
Temperatur Höchstwert	100	°C
Temperatur Fehler	±2	°C

## Über CO<sub>2</sub>-Konzentrationen

Der CO<sub>2</sub>-Gehalt in der Luft, die wir einatmen, hat einen direkten Einfluss auf unser Wohlbefinden und ist aus Sicht der öffentlichen Gesundheit von besonderem Interesse. Einfach ausgedrückt ist er ein Maß dafür, wie stark wir die Luft anderer Menschen einatmen. Wir Menschen atmen CO<sub>2</sub> aus, und wenn sich mehrere Personen in einem schlecht belüfteten Raum aufhalten, steigt der CO<sub>2</sub>-Gehalt allmählich an, ebenso wie die Konzentration von Aerosolen, die Erkältungen, Grippe und Coronaviren verbreiten. Eine weitere wichtige Auswirkung des CO<sub>2</sub>-Gehalts ist die kognitive Funktion, das heißt, wie gut man sich konzentrieren und nachdenken kann.

Die nachstehende Tabelle zeigt, bei welchen Werten CO<sub>2</sub> ungesund werden kann. Die CO<sub>2</sub>-Werte sind in ppm (parts per million) angegeben.

### CO<sub>2</sub> -Gehalt

250...400	Normale Konzentration in der Umgebungsluft
400...1000	Konzentration, die für bewohnte Innenräume mit gutem Luftaustausch typisch ist
1000...2000	Beschwerden über Schläfrigkeit und schlechte Luft
2000...5000	Kopfschmerzen, Schläfrigkeit und das Gefühl von abgestandener und stickiger Luft, Konzentrationsschwäche, Konzentrationsverlust, erhöhte Herzfrequenz und leichte Übelkeit können ebenfalls auftreten
5000	In den meisten Ländern Grenzwert für die gemittelte Exposition am Arbeitsplatz
>40.000	Exposition kann zu ernsthaftem Sauerstoffmangel führen, der dauerhafte Hirnschäden, Koma und sogar Tod zur Folge haben kann

## WEBLINKS

- [1] MonkMakes-Webseite mit Anleitungen: [http://monkmakes.com/pi\\_aq](http://monkmakes.com/pi_aq)
- [2] Software auf Github: [https://github.com/monkmakes/pi\\_aq](https://github.com/monkmakes/pi_aq)
- [3] TMP235-Datenblatt: [www.ti.com/product/TMP235](http://www.ti.com/product/TMP235)
- [4] CCS811-Datenblatt: [www.sciosense.com/products/environmental-sensors/ccs811-gas-sensor-solution/](http://www.sciosense.com/products/environmental-sensors/ccs811-gas-sensor-solution/)

# e-lektor e-zine

Your dose of electronics



Jede Woche, in der Sie den Elektor e-zine Newsletter nicht abonnieren, ist eine Woche mit großartigen Artikeln und Projekten zum Thema Elektronik, die Sie verpassen!

Also, worauf warten Sie noch? Melden Sie sich heute für unseren Elektor e-zine Newsletter unter [www.elektor.de/ezine](http://www.elektor.de/ezine) an und erhalten Sie zusätzlich ein kostenloses Raspberry Pi Projektbuch!



## Was können Sie erwarten?

### Redaktioneller Elektor-Newsletter

Jeden Freitag erhalten Sie die wichtigsten Artikel und Projekte der Woche. Wir zeigen MCU-basierte Projekte, IoT, Programmierung, KI und mehr!

### Elektor-Newsletter mit exklusiven Angeboten

Verpassen Sie nicht unsere Shop-Angebote, jeden Dienstag und Donnerstag haben wir eine besondere Aktion für Sie.

### Mailing von externen Partnern

Sie wollen über die laufenden Aktivitäten in der Branche informiert bleiben? Dann gibt Ihnen diese E-Mail die besten Einblicke. Unregelmäßig, aber immer mittwochs.

# Aller Anfang ...

muss nicht schwer sein: Willkommen bei der Diode!

Von Eric Bogers (Elektor)

Es hat eine Weile gedauert, aber jetzt haben wir endlich unsere Betrachtungen „passiver“ Bauteile abgeschlossen. In dieser Folge geht es los mit einigen „aktiven“ Bauteilen - den Halbleitern, um genau zu sein. Jetzt wird es wirklich interessant!



Wir beginnen diese Folge mit dem einfachsten aller aktiven Bauelemente, der Diode und verwandten Bauteilen. Zugegeben: Über die Frage, ob eine Diode ein aktives oder ein passives Bauelement ist, kann man sich endlos streiten, denn aktive Bauelemente heißen so, weil sie ein Signal verstärken können, und das tut eine Diode nicht. Da es sich aber um Halbleiterbauelemente handelt (oder, in der Vergangenheit, um Elektronenröhren, aber das gehört nicht in den Rahmen dieser Artikelserie), werfen wir sie auf

einen Haufen mit Transistoren und anderen aktiven Bauelementen. Würden wir ein „richtiges“ Lehrbuch schreiben, müssten wir viele Seiten mit Informationen über Halbleitermaterialien (Silizium, Germanium, Selen und andere), Dotierung, PN-Übergänge und viele andere Dinge füllen; aber diese Artikelserie richtet sich an den angehenden Elektronikliebhaber und nicht an Physiker oder Halbleiterhersteller. Der angehende Elektroniker interessiert sich vor allem für die Frage, welche Funktion ein bestimmtes Bauelement erfüllt und wie er es in seinen Entwürfen verwenden kann - und viel weniger für die Frage, wie und warum ein bestimmtes Bauelement tut, was es tut.

In **Bild 1** sehen Sie ein paar gängige Arten von Dioden. Oben links befinden sich zwei „normale“ Dioden, eine 1N4148 und die etwas dickere 1N4001. Unten links sind drei LEDs (Leuchtdioden) mit Durchmessern von 3 mm, 5 mm und 10 mm zu sehen. Rechts davon sehen Sie vier verschiedene Brückengleichrichter - das sind Kombinationen von vier Dioden in einem einzigen Gehäuse. Solche Brückengleichrichter werden in großer Zahl in (Netz-) Stromversorgungen eingesetzt. Das Bauteil unten rechts beispielsweise besitzt ein Metallgehäuse und ist zum Anschrauben an einen Kühlkörper vorgesehen. Dieser Gleichrichter kann einen Strom von nicht weniger als 25 A verarbeiten!

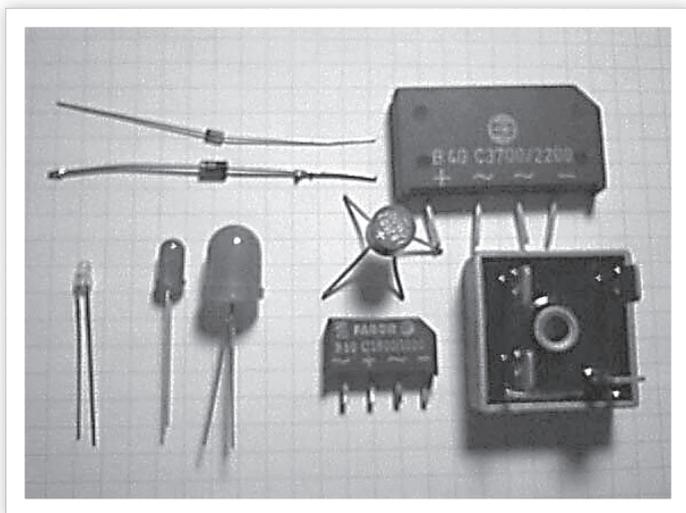


Bild 1. Mehrere Dioden und Brückengleichrichter.

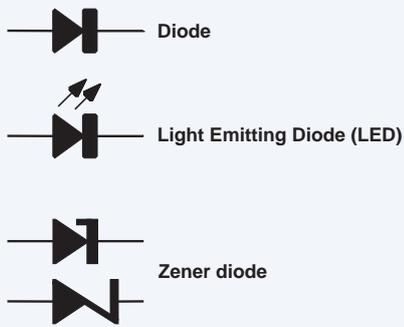


Bild 2. Schaltsymbole für einige gängige Dioden.

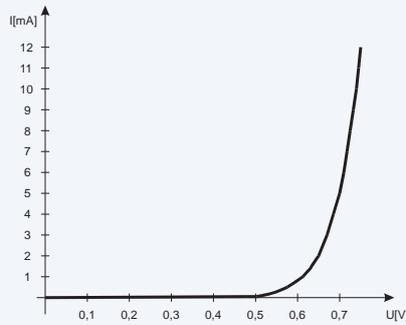


Bild 3. Die U/I-Kennlinie einer Diode.

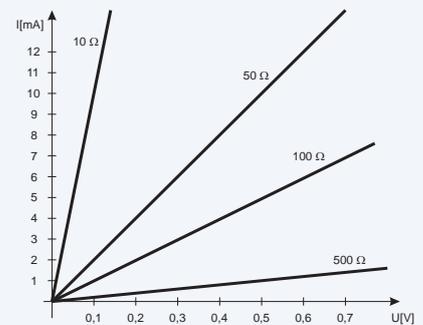


Bild 4. U/I-Kennlinien von vier Widerständen.

## Die Diode

Eine Diode wird oft als ein Ventil für elektrischen Strom betrachtet (was nicht ganz richtig ist, denn Dioden werden nicht „von außen“ betätigt wie zum Beispiel ein Heizungsventil). Das Bauteil leitet in eine Richtung, aber nicht in die andere. In **Bild 2** sehen Sie die Schaltsymbole einiger Dioden; von oben nach unten: eine gewöhnliche Diode, eine Leuchtdiode (LED) und darunter zwei Versionen des Symbols für eine Z-Diode (im unteren Symbol ist sogar ein „Z“ zu erkennen). Das Diodensymbol ähnelt einem Pfeil, der in die Richtung zeigt, in der der Strom fließen kann. Der Anschluss auf der linken Seite in Bild 2 wird als Anode und der auf der rechten Seite (der Balken) als Kathode bezeichnet.

Eine (Silizium-) Diode beginnt zu leiten, wenn das Potential an der Anode um etwa 0,6 V positiver ist als das an der Kathode. Das ist in etwa vergleichbar mit einem mechanischen Ventil, bei dem ein bestimmter Druckunterschied erforderlich ist, bevor sich das Ventil öffnet. Und bevor Sie fragen, woher diese 0,6 V kommen: Sie werden durch das Halbleitermaterial, in diesem Fall Silizium, bestimmt. Bei einer Germanium-Diode beträgt diese Mindestspannung etwa 0,3 V. Warum das so ist, kann Ihnen ein Physiker sicher im Detail erklären, aber das ist für uns hier nicht von Bedeutung. Wir erinnern uns nur an diese beiden Spannungen: 0,6 V für Silizium und 0,3 V für Germanium.

Interessanter wird es, wenn wir ein Diagramm des Stroms durch eine Diode in Abhängigkeit von der Spannung über der Diode erstellen. Im U/I-Diagramm in **Bild 3** zum Beispiel ist die Spannung auf der horizontalen Achse und der Strom auf der vertikalen Achse aufgetragen. Zum Vergleich zeigt **Bild 4** U/I-Kennlinien einiger gewöhnlicher Widerstände. Bei einem Widerstand ist die Kennlinie umso steiler, je kleiner der Widerstand ist und umso flacher, je größer der Widerstand ist.

Wenn wir uns Bild 3 genau ansehen, sehen wir, dass bis zu einer Spannung von 0,5 V kaum etwas passiert - es fließt praktisch kein Strom durch die Diode. Jenseits der 0,5 V wird die Kurve schnell immer steiler; die Steigung ab diesem Punkt wird hauptsächlich durch den Innenwiderstand der Diode bestimmt.

Der Vollständigkeit halber sei erwähnt, dass die Kurve in Bild 3 mit einer Siliziumdiode des Typs 1N4148 gemessen wurde, die U/I-Kennlinien anderer Si-Dioden zeigen jedoch einen sehr ähnlichen Verlauf.

## Die Spezifikationen einer Diode

Der Hersteller einer Diode gibt im Datenblatt mindestens den maximal erlaubten Strom in Durchlassrichtung und die maximale

Spannung in Sperrrichtung an (wobei das Kathodenpotential positiver ist als das Anodenpotential). Häufig wird auch die maximale Belastbarkeit (die maximale Verlustleistung) einer Diode angegeben.

Der maximale Strom durch die 1N4148 im Beispiel, einer typischen „Kleinsignaldiode“ beträgt 100 mA, während die maximale Verlustleistung bei 500 mW liegt.

Andererseits ist eine Diode auch nicht in der Lage, beliebig hohe Spannungen zu sperren. Wenn die maximale Sperrspannung überschritten wird, beginnt die Diode auch in umgekehrter Richtung zu leiten. Wird der Strom in diesem Fall nicht stark begrenzt, überhitzt sich die Diode und der Silizium-Halbleiterkristall schmilzt. Dann leitet die Diode in beide Richtungen, und zwar für immer... Diese höhere Spannung führt auch zu einer viel höheren Verlustleistung, so dass die Diode schon bei einem vergleichsweise kleinen Strom in Sperrrichtung zerstört wird.

## Regulierung von Spannungen

Mit gewöhnlichen Dioden lässt sich auf sehr einfache Weise eine einigermaßen stabile Spannung erzeugen (siehe **Bild 5**). Zunächst zur Schaltung links in Bild 5: Hier ist eine Diode über einen Vorwiderstand von 1 kΩ an eine Gleichspannungsquelle angeschlossen. Bei einer Spannung von 7,5 V fließt ein Strom von 6,68 mA, und der Spannungsabfall über der Diode beträgt (laut Datenblatt) 0,715 V. Wir verdoppeln nun die Versorgungsspannung auf 15 V. Der Strom erhöht sich natürlich auf 14,12 mA, aber die Spannung über der Diode steigt nur geringfügig auf 0,761 V. Fazit: Bei einer Verdopplung der Versorgungsspannung steigt die Spannung über der Diode nur um etwa 6 %.

Wenn wir eine Art Zweistufenschaltung wie rechts im Bild 5 verwenden, wird die Spannung noch besser stabilisiert. Mit den drei Dioden in Reihe wird eine Spannung von etwa 2 V erzeugt, und mit dieser bereits einigermaßen stabilen Spannung wird eine zweite Diode gespeist.

Dabei messen wir bei einer Eingangsspannung von 7,5 V eine Spannung von 0,617 V über der zweiten Diode und bei einer Eingangsspannung von 15 V eine Spannung von 0,662 V. Eine Verdoppelung der Eingangsspannung, also eine Erhöhung um 100 % führt in dieser Schaltung zu einer erhöhten Spannung über der zweiten Diode von nur 0,8 %.

Eine solche Stabilisierung von Spannungen mit gewöhnlichen Dioden wird nur für sehr niedrige Spannungen verwendet. Für höhere Spannungen werden Z-Dioden verwendet, die in der nächsten Folge vorgestellt werden.

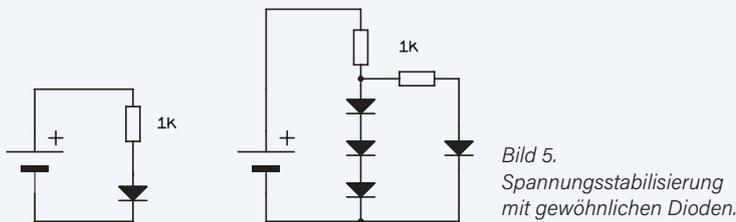


Bild 5. Spannungsstabilisierung mit gewöhnlichen Dioden.



Bild 6. Netzteil mit Einweg-Gleichrichter.

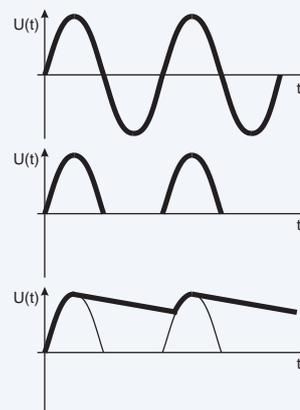


Bild 7. Spannungsverläufe an verschiedenen Stellen des Einweg-Gleichrichters.

## Stromversorgungsschaltungen

Eine der wichtigsten Anwendungen von Dioden ist die des Gleichrichters in Stromversorgungsschaltungen. Elektronische Schaltungen und Geräte benötigen für ihre Stromversorgung meist eine Gleichspannung, die je nach Gerät zwischen 1,5 V und 150 V liegen kann. Da das Stromnetz jedoch nur eine Wechselspannung von 230 V liefert, muss diese Spannung zunächst auf eine akzeptabel niedrigere Spannung transformiert, dann gleichgerichtet und geglättet und in den meisten Fällen auch stabilisiert oder gar geregelt werden.

## Einseitiger Gleichrichter

Der Begriff „einseitig“ (siehe **Bild 6**) hat nichts mit mangelnder Ausgewogenheit zu tun, sondern bedeutet, dass nur eine Hälfte der beiden Halbwellen der Eingangswchselspannung durchgelassen wird - und zwar nur die positive Halbwellen - und dies auch nur für die relativ kurze Zeit, in der die Ausgangsspannung des Transformators höher ist als die Spannung über dem Elektrolytkondensator (plus der Durchlassspannung der Diode von etwa 0,7 V). Diese Art von Schaltung wird passenderweise auch als Einweggleichrichter bezeichnet.

Wir haben dies in **Bild 7** skizziert. Die obere Grafik zeigt die Wechselspannung, wie sie vom Transformator geliefert wird. Wenn man den Puffer- oder Glättungs-Elektrolytkondensator für den Moment weglässt, ergibt sich über dem Widerstand (der Last) eine pulsierende Gleichspannung, wie sie im mittleren Diagramm dargestellt ist.

Das unterste Diagramm ist eigentlich das interessanteste. Es zeigt die Spannung über dem Pufferkondensator. Während der positiven Halbperioden der Wechselspannung wird dieser Kondensator in jedem Zyklus sehr kurz aufgeladen; in der restlichen Zeit des Zyklus wird der Kondensator von der Last entladen. In der Praxis handelt es sich bei dieser Last im Allgemeinen nicht um einen einfachen Widerstand, sondern um eine komplette elektronische Schaltung, die gerne mit einer stabilen Gleichspannung versorgt wird.

In der nächsten Folge werden wir damit beginnen, diese (und andere) Gleichrichter zu berechnen. ◀

220003-02

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

## Ein Beitrag von

Idee und Illustrationen: Michael Ebner

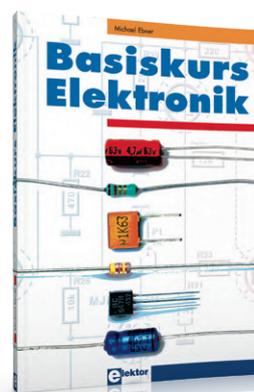
Text und Redaktion: Eric Bogers

Übersetzung: Rolf Gerstendorf

Layout: Giel Dols

## Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Senden Sie der Redaktion eine E-Mail an [redaktion@elektor.de](mailto:redaktion@elektor.de).



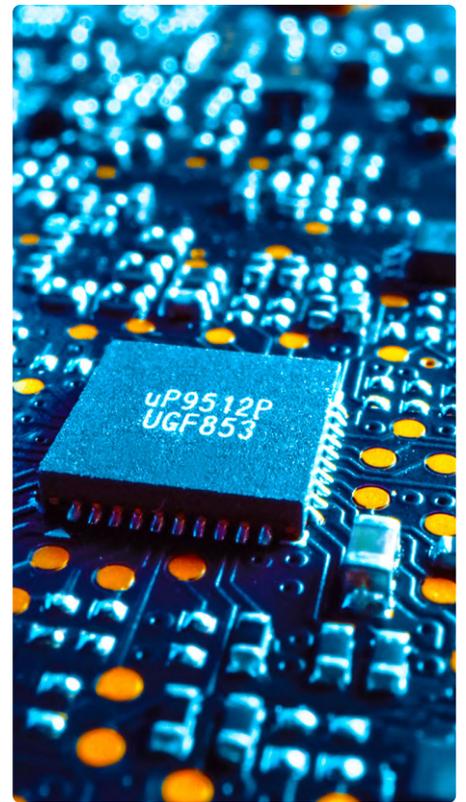
## PASSENDE PRODUKTE

- ▶ **Ebner, Michael, „Basiskurs Elektronik“**  
[www.elektor.de/basiskurs-elektronik-pdf](http://www.elektor.de/basiskurs-elektronik-pdf)
- ▶ **B. Kainka, Elektronik-Grundlagen und Einsteiger-Projekte (E-Book), Elektor 2020**  
[www.elektor.de/elektronik-grundlagen-und-einsteiger-projekte-pdf](http://www.elektor.de/elektronik-grundlagen-und-einsteiger-projekte-pdf)

# Beherrschen Sie die Fachsprache?

## Fünf praktische Tipps für Ingenieure

Ohne fließende Englischkenntnisse kann es schwierig sein, die von Autoren vermittelten Fachbegriffe zu verstehen. Außerdem erfordern viele Module in den Ingenieurwissenschaften das Lesen spezialisierter Bücher [1] und sogar das Verfassen von akademischen Berichten. Daher ist eine gute Beherrschung der englischen Sprache erforderlich.



Sie können dem Definitionsspiel andere Gegenstände hinzufügen, die sich auf einer Liste mit Fachbegriffen befinden. Der Lehrer muss den Lernenden die Mittel geben, mit denen sie diese Sachen beschreiben können, z.B. Formen, Abmessungen, Materialien, Adjektive, und Handlungen, die mit diesen Dingen ausgeführt werden. Er darf sie auch bitten, sich so lange wie möglich an nur eine dieser Kategorien zu halten, z.B. drei oder vier Adjektive zu verwenden, bevor sie zu anderen Beschreibungsarten übergehen.

### 4. Seien Sie konkret und genau

Im Englischen nutzt man kürzere Sätze und im technischen Englisch ist dies noch wichtiger. Ideen, die wir zu vermitteln versuchen sind oft komplex und daher muss die Sprache so klar wie möglich sein, um Verwirrung zu vermeiden. Um dies zu erreichen, verwenden Sie so einfache Bindewörter wie möglich.

### 5. Betrachten Sie Ihre Zuhörer

Möglicherweise verfügen die Personen, mit denen Sie sprechen (oder denen Sie schreiben), nicht über das gleiche technische Wissen wie Sie. Stellen Sie daher sicher, dass Sie ein angemessenes Maß an Hintergrundwissen zur Verfügung stellen. ◀

220188-02

Gründe, warum ein Ingenieur fließend Englisch sprechen sollte, gibt es genug. Hier sind fünf praktische Tipps.

### 1. Nehmen Sie an einem Online-Kurs teil

Es gibt kompetenzbasierte Kurse, die für praktizierende Ingenieure oder Studenten der Elektrotechnik entwickelt werden. Sie bieten sorgfältig abgestufte Übungen in den wichtigsten Bereichen, wie z.B. Hören und Sprechen. Darüber hinaus vermitteln sie den Lernern die Fachsprache der Elektrotechnik, die sie für eine erfolgreiche Arbeit im Betrieb oder an der Fakultät benötigen. Ausführliche Hörübungen stammen aus elektrotechnikbezogenen Vorlesungen und alle Lesetexte gehören zur gleichen Studienrichtung. Außerdem liegt der Schwerpunkt durchgehend auf dem Schlüsselvokabular in der Elektrotechnik, das die Lernenden benötigen. Solche Online-Englischkurse [2] sind bei Anbietern wie Preply verfügbar. Sie bieten folgende Vorteile:

- Zuhören: wie man längere Vorträge versteht und effektiv Notizen macht, einschließlich wie man der Argumen-

tation folgt und den Standpunkt des Redners erkennt;

- Sprechen: wie man sich in realistischen Situationen versteht;
- Vokabular: Hier gehört eine breite Palette von Aktivitäten zur Entwicklung des Wissens der Lernenden und der Verwendung von Schlüsselvokabular, sowohl im Bereich der Elektrotechnik als auch des akademischen Studiums im Allgemeinen.
- Audio für weiteres Selbststudium oder Hausaufgaben.

### 2. Konzentrieren, üben, sich herausfordern

Setzen Sie sich erreichbare Ziele. Versuchen Sie nicht, alles auf einmal zu lernen. Denken Sie über Ihre Bedürfnisse nach und setzen Sie dementsprechend Prioritäten: Verbringen Sie den größten Teil Ihres Arbeitstages damit, E-Mails auf Englisch zu verfassen und zu beantworten? Oder nehmen Sie an Online-Meetings mit internationalen Kunden teil? Priorisieren Sie das Lernen spezifischer englischer Ausdrücke, Vokabeln und Grammatik, die Ihnen bei diesen Aufgaben helfen. Lernen und entwickeln Sie Fähigkeiten, die der Art Ihrer Arbeit dienen. Sprache soll zu Ihnen passen, nicht umgekehrt!

### 3. Spielen Sie das Definitionsspiel

Das Definitionsspiel (bei dem die Lerner aufgefordert werden, ein Ding zu beschreiben, an das sie denken, ohne einen Teil des Begriffs zu erwähnen, bis die anderen erraten, was das Ding ist) ist am sprachintensivsten.



### WEB LINKS

- [1] Elektor-Bücher: <https://www.elektor.de/buecher>
- [2] Online-Kurse: <https://preply.com/de/kurse/englisch>



Widerstände, die so heiß geworden sind, dass sich das Gehäuse oder die Markierungen verfärbt haben, sollten normalerweise nicht aufbewahrt werden, aber manchmal möchte man sie aus nostalgischen Gründen vielleicht doch testen und behalten.



### Kondensatoren und Induktivitäten

Als ich vor vielen Jahren etwa 20 Jahre alt war, habe ich mir ein Kapazitätsmessgerät gebaut. Ein astabiler 555-Multivibrator triggerte eine monostabile Kippstufe mit einem 74121 und dem Testkondensator, um eine Impulsfolge mit einem zum Testkondensator proportionalen Tastverhältnis zu erzeugen. Für jeden Bereich verwendete ich ein voreingestelltes Potentiometer und kalibrierte es mit 1%-Kondensatoren. Das Gerät konnte Kondensatoren von etwa 1 pF bis 100 µF messen und war eines der nützlichsten Geräte, die ich je gebaut habe. Viele Digitalmultimeter verfügen heute über eine Kapazitätstestfunktion, und ich habe kürzlich in einer Elektronikzeitschrift ein Gerät für etwa 40 € gefunden, das Kapazität und Induktivität misst, wenn auch mit recht begrenzten Messbereichen. Ich hatte früher ein einfaches, aber angemessenes LCR-Messgerät etwa zum gleichen Preis, habe aber vor kurzem in ein vielseitigeres Gerät investiert. Bei dieser Art von Prüfgeräten bekommt man in der Regel das, wofür man bezahlt, und was man bekommt, hängt vom Budget und den Anforderungen ab. Es lohnt sich wirklich nicht mehr, eigene Testgeräte zu bauen. In manchen Fällen müssen Sie jedoch esoterischere Messungen vornehmen, zum Beispiel den äquivalenten Serienwiderstand (ESR) eines Kondensators oder die Güte einer Induktivität, und dafür benötigen Sie unter Umständen eine fortschrittlichere Testausrüstung. Elektor hat im Laufe der Jahre viele Entwürfe für Geräte zur Messung aller Aspekte einer Kapazität oder einer Induktivität veröffentlicht, und eine Suche auf der Elektor-Website kann Sie zu einem Projekt führen, das Ihre Bedürfnisse erfüllt. Im Elektor-Shop finden Sie auch einige preisgünstige kommerzielle Messgeräte (siehe Textkasten).

Superkondensatoren sind nur ein Sonderfall von Elektrolytkondensatoren, die jedoch die Möglichkeiten der meisten Tester überschreiten. Sie haben in der Regel eine Nennspannung von 5,5 V, die Sie nicht überschreiten sollten. Testen Sie sie mit einem Widerstand von 100 Ω an einer 5-V-Versorgung und zählen Sie die Sekunden, die ein vollständig entladener Kondensator benötigt, um auf 3,5 V zu kommen. Teilen Sie diesen Wert durch 100, dann haben Sie einen groben Schätzwert für die Kapazität in Farad. Bei dieser Abschätzung machen wir uns die RC-Zeitkonstante zunutze, also die Zeit, die der Kondensator benötigt, um 0,7 der vollen Ladung zu erreichen ( $5 \text{ V} \times 0,7 = 3,5 \text{ V}$ ). Ein 1-F-Kondensator benötigt  $100 \Omega \times 1 \text{ F} = 100 \text{ s}$ , um auf 3,5 V zu kommen.  $100/100 = 1$ , also 1 Farad.

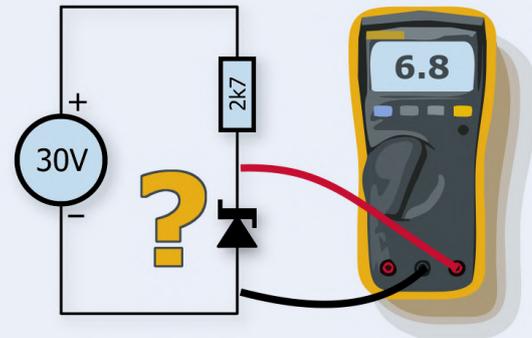
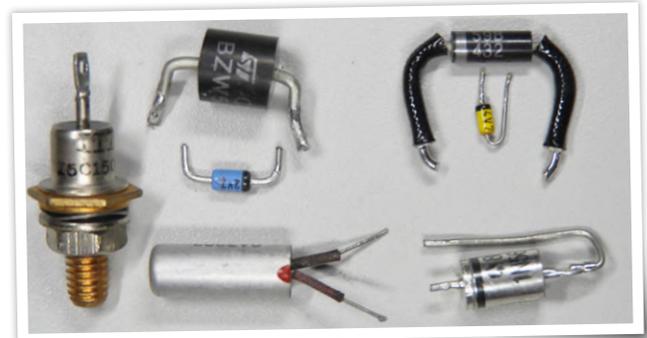


Bild 1. Eine Möglichkeit, Z-Dioden schnell zu testen.

### Transformatoren

Transformatoren sind nur besondere Spulen. Bestimmen Sie den Wicklungswiderstand mit einem Multimeter und ermitteln Sie dann die Induktivität. Bei einem kleinen Leistungstransformator dürfte eine Wicklung mit einem Widerstand über 50 Ω und einer Induktivität über 10 H eine Netzwicklung sein. Da sich die Induktivität proportional zum Quadrat der Anzahl der Windungen verhält, kann man das Übersetzungsverhältnis durch Ziehen der Quadratwurzel aus dem Verhältnis der Induktivitäten der Wicklungen ermitteln.

Ein Transformator mit Induktivitäten von 10 H und 50 mH besitzt zum Beispiel ein Windungsverhältnis von etwa  $\sqrt{(10/0,05)} = 14$ . Wenn es sich also um einen 240-VAC-Netztransformator handelt, sollte die Sekundärseite etwa 15 V liefern. Viele netzseitigen Wicklungen haben aber eine höhere Induktivität, als Ihr LCR-Meter messen kann..



### Dioden

Dioden sind mit jedem Multimeter leicht zu prüfen - sie leiten in eine Richtung und in die andere nicht. Digitalmultimeter haben üblicherweise einen Diodentestbereich, der Ihnen einen Hinweis auf die Durchflussspannung der Diode gibt. Etwa 0,6...0,7 bedeutet, dass es sich um eine Standard-Siliziumdiode handelt, 0,3...0,5, dass Sie eine Schottky-Diode vor sich haben und 0,2, dass Sie es mit einer Germanium-Diode zu tun haben. Die Position des Dezimalpunkts kann variieren, aber es ist die erste Ziffer hinter dem Dezimalpunkt, auf die Sie achten müssen. Z-Dioden sind etwas schwieriger. Sie sperren wie normale Dioden, wenn sie mit einem Multimeter geprüft werden, da ein Multimeter wahrscheinlich nicht genug Spannung anlegt, um sie in den leitenden Zustand zu versetzen. Zum Testen von Z-Dioden verwenden Sie am besten Ihr Labornetzgerät mit hoher oder maximaler Spannung und einer sehr niedrigen Stromgrenze (zum Beispiel 10 mA) oder einen Vorwiderstand, der bei voller Spannung etwa 10 mA durchlässt. Legen Sie diese Spannung an die Z-Diode und messen Sie die Spannung mit Ihrem Multimeter (Bild 1). Dies funktioniert natürlich nur bei Z-Dioden mit Durchbruchspannungen, die kleiner sind als die Ihres Netzteils, aber ein 30-V-Netzgerät reicht

## Tipps & Tricks, Best Practices und andere nützliche Infos

für die meisten in der Praxis vorkommenden Z-Dioden aus. LEDs können ähnlich wie Z-Dioden getestet werden, aber Sie müssen die Spannung des Netzgeräts auf etwa 5 V begrenzen, da LEDs eine recht niedrige Schwellspannung aufweisen. Testen Sie jede LED mit durchsichtigem Gehäuse in beide Richtungen, denn möglicherweise handelt es sich ja um einen zweifarbigem Typ. Begrenzen Sie den Strom auf 10 mA, was ausreicht, um jede LED zum Leuchten zu bringen, und niedrig genug auch für kleine LEDs ist.

### Transistoren

Als ich etwa 13 Jahre alt war, sah ich irgendwo einen Artikel über ein einfaches Transistortestgerät, das die Funktion prüft und einen  $h_{fe}$ -Wert (Gleichstromverstärkung) angibt. Bei Silizium-Kleinsignaltransistoren war das Gerät recht genau, aber bei Germaniumtransistoren ziemlich ungenau (was heutzutage kaum noch ein Problem ist). Ich habe es oft benutzt und benutze es auch heute noch. Aber auch hier gilt: Die meisten Multimeter besitzen einen eingebauten Transistortester, und im Allgemeinen leisten sie gute Arbeit.

Denken Sie auch daran, dass sich ein Transistor wie zwei Dioden verhält, so dass die Identifikation des Basisanschlusses recht einfach ist. Darlington- oder Leistungstransistoren sind schwieriger zu testen, da sie für einen korrekten Test höhere Basis-Emitter-Spannungen oder größere Kollektorströme benötigen. Hobby-Elektronikern genügt jedoch meist ein einfacher Funktionstest, nach dem man weiß, ob der Transistor funktioniert oder nicht.

### MOSFETs

MOSFETs sind inzwischen überall zu finden, lassen sich aber mit einem normalen Transistortester nicht testen. Die meisten von ihnen benötigen Gate-Source-Spannungen von 5 V oder mehr, um sie vollständig einzuschalten. Glücklicherweise gibt es eine einfache Möglichkeit, einen MOSFET mit einer strombegrenzten Spannungsversorgung (oder einer Spannungsversorgung und einem Widerstand/einer Glühlampe) und Ihren Fingern (oder einem Widerstand mit hohem Wert von einem Megohm oder so) zu testen. Wenn Sie über den Widerstand (oder Ihre Finger) das Gate an den Pluspol legen, wird der MOSFET eingeschaltet, beim Gate an die Source ausgeschaltet. Da MOSFETs eine beträchtliche Gate-Kapazität aufweisen, kann das Aus- und Einschalten eine Sekunde oder länger dauern. Verwenden Sie 12 V, um eine ausreichende Gate-Spannung anzulegen, und begrenzen Sie den Strom auf 100 mA oder weniger, sonst können Sie sich leicht die Finger verbrennen, wenn der MOSFET nicht auf einem Kühlkörper sitzt!

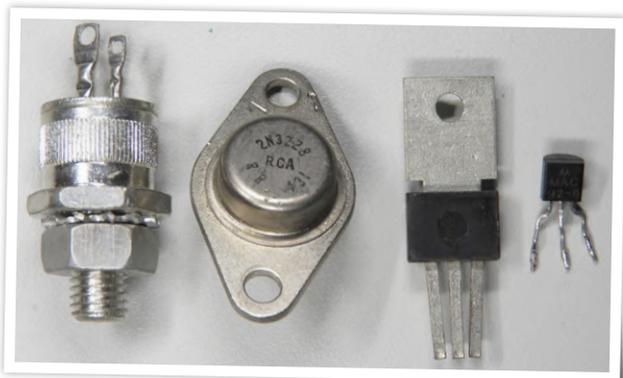


Bild 2. Ein einfacher Leistungshalbleitertester testet MOSFETs, Transistoren, Thyristoren und Triacs.

### Thyristoren und Triacs

Für solche Bauteile können Sie den gleichen Aufbau wie bei den MOSFETs verwenden, aber setzen einen kleineren Widerstand für das Gate ein - etwa 470  $\Omega$  bis 1 k $\Omega$  sollten ausreichen. Wenn Sie das Gate über den Widerstand mit der Anode oder der positiven Versorgungsspannung verbinden, sollte der Thyristor einschalten (Strom leiten) und dies auch dann noch tun, wenn Sie die Gate-Verbindung wieder entfernen. Um einen Thyristor auszuschalten, müssen Sie die Anodenverbindung unterbrechen. Bei Triacs verhält es sich ähnlich, nur leiten Sie den Strom in beide Richtungen.

Die Prüfanforderungen für MOSFETs, Transistoren und Thyristoren/Triacs sind ähnlich, und man kann ein einfaches Prüfgerät bauen, sie alle zu testen. Das von mir gebaute Gerät ist in **Bild 2** zu sehen.

### VDR und TVS

Spannungsabhängige Widerstände (VDR) und Suppressordioden (Transient-Voltage-Suppression-Diode, TVS) und andere vor Überspannungen schützende Bauteile wie Gasableiter lassen sich wegen der hohen Spannungen nur schwer überprüfen: Ein 460-V-VDR beginnt erst bei über 500 V zu leiten! Wenn Sie jedoch Zugang zu einem Megaohmmeter oder Isolationstester eines Elektrikers haben, können Sie diese Bauteile schnell überprüfen.

Die meisten Isolationsprüfgeräte haben drei Prüfspannungen: 250 V, 500 V und 1000 V. Eine Prüfspannung von 250 V führt bei einem 460-V-VDR zu keinem Ausschlag beziehungsweise Anzeige und bei 500 V nur ein wenig. Erst bei 1000 V sollte das Gerät einen geringen Widerstand des Prüflings anzeigen. Die Ergebnisse können je nach verwendetem Prüfgerät variieren, sollten aber für eine Prüfung ausreichen. Nicht funktionierende VDRs weisen entweder einen Kurzschluss auf oder sind „total“ offen, und dieser Test zeigt beide Zustände an. Übrigens: Halten Sie bei solchen Prüfspannungen Ihre Finger in Sicherheit!

## Polyfuses und PTC-Widerstände

Solche Bauteile weisen in der Regel einen geringen Widerstand von einigen Ohm auf, aber wenn mehr als der Nennstrom durch sie fließt, erwärmen sie sich, werden dabei hochohmig und bleiben dies in der Regel so lange, bis die Ursache des Überstroms beseitigt ist. Testen Sie die Bauteile an einem strombegrenzten Netzgerät und erhöhen Sie die Strombegrenzung langsam. Anfangs ist die Spannung über dem Bauteil gering, aber mit steigender Stromgrenze erhöht sich ihr Widerstand und auch die Spannung an ihnen steigt relativ schnell. Auf diese Weise können Sie sich ein Bild von ihrem Nennstrom machen (der unter dem Punkt liegt, an dem sie hochohmig werden). Und verbrennen Sie sich nicht die Finger, die Polyfuses erhöhen nicht nur ihren Widerstand, sondern werden dabei auch richtig heiß!

## DIP-Schalter

Bei gebrauchten DIP-Schaltern kommt es oft vor, dass einzelne Schalter ihren Dienst verweigern. Prüfen Sie DIP-Schalter deshalb vor ihrer Wiederverwendung. Natürlich kann man jeden einzelnen Schalter mit einem Multimeter durchpiepsen, aber bei einer größeren Anzahl von Prüflingen ist das sehr umständlich. Besser, Sie bauen schnell einen DIP-Schalter-Tester mit einer IC-Fassung, Widerständen und LEDs. Ein solcher Tester macht es viel bequemer zu erkennen, ob ein Schalter offen oder geschlossen ist.

Ein solches Prüfgerät lässt sich auch zum Test von LEDs einsetzen. Statt der Pole des Schalters wird die LED in die Fassung gesteckt. Sowohl diese als auch die LED des Testers sollten aufleuchten, wenn die LED in Ordnung ist. Eine von mir gebaute Schaltung ist in **Bild 3** zu sehen.

## Integrierte Schaltungen

ICs sind schwieriger zu testen, da es so viele verschiedene Arten gibt. Aber es gibt einfache Tests, die man durchführen kann. Da Operationsverstärker in der Regel die gleichen Pinbelegungen aufweisen, kann man einen einfachen Operationsverstärkertester bauen, indem man ihn so konfiguriert, dass er wie ein astabiler Multivibrator mit ein paar LEDs am Ausgang funktioniert. Das funktioniert auch bei vielen Komparatoren, aber bedenken Sie, dass manche Komparatoren nur einen nackten Transistor am Ausgang haben und keine Stromquelle sind.



Bild 3. Eine selbstgebaute Testeinrichtung für DIP-Schalter und LEDs im Einsatz.

## Das Programmiergerät TL866 als Digital-IC-Tester



Quelle: David Ashton

Der TL866 ist in erster Linie ein recht kompetentes Programmiergerät für ein sehr breites Spektrum von programmierbaren ICs, wie viele Testberichte, auch von Elektor [2] zeigen. Ich möchte mich jedoch auf seine Fähigkeit konzentrieren, Logik-ICs zu testen. Die aktuelle Version TL866-II präsentiert sich in einem Kunststoffgehäuse mit einer 40-poligen ZIF-Fassung guter Qualität und einigen LEDs sowie einem USB-Anschluss. Ich musste die Software von der Website des Herstellers herunterladen - etwas, das ich zwar nicht gerne tue, aber da es die Website des Herstellers ist, ist das Risiko nicht groß. Der TL866 hat ein gutes druckbares Handbuch (PDF). Die Software ist recht übersichtlich und man kann die meisten Bauteile und Funktionen schnell finden.

Auf dem Reiter Device im Hauptmenü befindet sich der Punkt Logic Test mit einer langen Auswahlliste von CMOS-4000- und TTL-74-IC-Typen und sogar einige Intel 8080-Peripheriebauteile. Ich habe eine gute halbe Stunde mit meinen Kisten voller CMOS- und TTL-ICs verbracht und viele von ihnen ausprobiert, und der TL866 hat die meisten von ihnen ohne Probleme gemeistert. Bei den 74LS21s trat jedoch immer wieder ein Fehler auf - ein wenig enttäuschend bei einem so einfachen IC.

Aber jetzt kommt der spannendste Teil. Die TL866-Software zeigt die Testsequenz an, die sie verwendet, und man kann sie kopieren und bearbeiten (und unter einem neuen Namen speichern). Als ich mir das Datenblatt des 7421 und die programmierte Testsequenz ansah, fiel mir auf, dass Pin 3 und Pin 11 mit „H“ (High Level Expected) markiert waren. Diese Pins sind allerdings „not connected“ und sollten in der Testprozedur ignoriert werden. Als ich diese Pins auf „x“ (ignorieren) änderte, funktionierte der Testablauf, und ich testete alle meine 74LS21 erfolgreich und ohne Fehler. Ich bin nun zuversichtlich, dass ich so gut wie jedes digitale IC testen kann, auch wenn ich für einige von ihnen eigene Testsequenzen schreiben muss. Bei einigen anderen ICs traten Fehler auf, vor allem bei monostabilen Multivibratoren, die ohne Timingbauteile natürlich schwer zu prüfen sind.

Zum Testen von SMD-ICs gibt es auf eBay und anderen Webseiten Sets mit SMD-auf-DIL-Adaptoren. Tatsächlich kann man einen TL866-Programmer mit verschiedenen Adaptoren für weniger als 50 € kaufen - ein sehr handliches Prüfgerät zu einem guten Preis und eine gute Ergänzung für die Sammlung von Testgeräten jedes Elektroniklers.

# Von Entwicklern für Entwickler

Tipps & Tricks, Best Practices und andere nützliche Infos

Sie können auch Tester für Doppel- und Vierfach-Operationsverstärker herstellen. Verwenden Sie eine Fassung von guter Qualität, am besten einen Zero-Insertion-Force-Typ (ZIF), wenn Ihr Budget das zulässt. Das Gleiche können Sie für den beliebten 555-Timer tun - lassen Sie ihn als astabiler Multivibrator mit LED am Ausgang laufen.

Digitale ICs sind etwas schwieriger. Wenn Sie diese häufig testen müssen, kaufen Sie sich am besten ein geeignetes Testgerät. Es gibt ein Gerät namens TL866 [2] (siehe **Kasten**), das normalerweise für weniger als 50 € erhältlich ist und eigentlich ein EPROM-Programmiergerät ist, aber auch Logik-ICs testet. Es gibt einige verschiedene Modelle, daher sollten Sie sich vor dem Kauf die technischen Daten ansehen. Der TL866-II ist die aktuelle Version.

Alle Modelle sind mit einer ZIF-Fassung für die meisten Größen von bedrahteten ICs ausgestattet. Bei vielen Angeboten werden auch praktische Adapter zum Testen verschiedener SMD-Gehäuse mitgeliefert. Einige Modelle sind für die In-System-Programmierung (in-circuit serial programming, ICSP) geeignet, wenn das für Sie interessant ist, aber die neueren Modelle können leider keine sehr alten EPROMs programmieren, was für elektronische Oldtimer sehr wichtig wäre. Für das Testen von digitalen ICs ist aber beides nicht so wichtig. Sie können auch Ihre eigenen Testprogramme für ICs schreiben, die nicht in der Datenbank enthalten sind, oder für solche, bei denen die mitgelieferten Programme nicht funktionieren (es gibt einige).

## Quarze

Quarze und Oszillatoren gehören zu den häufigeren Bauteilen, die Sie beim Zerlegen kommerzieller Geräte finden. In der Regel sind diese Bauteile mit ihrer Frequenz gekennzeichnet, und es ist nur ein einfacher Test erforderlich, um herauszufinden, ob sie funktionieren oder nicht. Um die Frequenz zu überprüfen oder um nicht gekennzeichnete Quarze zu testen, wird aber ein Frequenzzähler benötigt. Man braucht nur zwei invertierende Gatter, um einen Quarz zum Schwingen zu bringen, und ein weiteres, um zu erkennen, dass er schwingt. Bei einem Vierfach-Gatter-IC können Sie das vierte Gatter als Puffer für einen Frequenzzähler verwenden.

Quarze für Mikrocontroller liegen typischerweise im MHz-Bereich, aber auch die so genannten Uhrenquarze mit 32,768 kHz werden häufig verwendet. Bei diesen Frequenzunterschieden benötigen Sie möglicherweise zwei Oszillatoren, um alle Quarze überprüfen zu können. Fügen Sie auch eine DIP-Fassung hinzu, um Oszillatoren zu überprüfen. Sie brauchen nur eine Stromversorgung und eine Verbindung zum Ausgang. SMD-Quarze und -Oszillatoren sind schwieriger zu messen, so dass Sie möglicherweise eine spezielle Prüffassung benötigen.

## Relais

Relais werden immer noch häufig verwendet. Wenn es um Isolation zweier Stromkreise und hohe Strombelastbarkeit geht, sind sie unschlagbar. Prüfen Sie die Kontakte mit einem Multimeter mit Durchgangssummer, der einen niedrigen Widerstand anzeigt, und schalten Sie das Relais mit der entsprechenden Spannung ein, um sicherzustellen, dass die Kontakte öffnen oder schließen. Wenn Sie viele Relais eines Typs haben, kann es sich lohnen, eine Prüfvorrichtung für sie zu bauen und eine Last von bis zu einigen Ampere (zum Beispiel eine alte Autoglühbirne) zu verwenden, um die Kontakte mit unzulässig hohem Widerstand auszusortieren.

Bild 4. Pinzetten-Messspitze für Multimeter. (Quelle: Sparkfun)



## Andere Messwerkzeuge

Wenn Sie sich nur ein einziges gutes Testgerät leisten können oder wollen, sollten Sie als eierlegende Wollmilchsau ein gutes Multimeter mit Transistortester, Kapazitätsmessung und „artfremden“ Messbereichen (Frequenz, Temperatur...) kaufen. Es wird sich wirklich sehr nützlich machen, bis Sie bereit sind, Geld für bessere Testgeräte in die Hand zu nehmen, seien es kommerzielle oder selbstgebaute.

Eines der nützlichsten Dinge, die Sie kaufen können, wenn Sie mit SMDs arbeiten, ist eine Pinzettenmessspitze (**Bild 4**). Sie lässt sich an Ihr Multimeter anschließen und ermöglicht es Ihnen, SMD-Widerstände, -Kondensatoren, -Induktivitäten und -Dioden schnell und einfach zu prüfen. Sie werden von Sparkfun hergestellt und sind überall erhältlich. Wenn Sie wirklich viele SMD-Tests durchführen, können Sie auch ein in eine Pinzettenmessspitze eingebautes Messgerät erwerben. So etwas gibt es, und mit dem DT71 sogar eines im Elektor-Store (siehe **Kasten**). Eine gute Übersicht solcher Messgeräte finden Sie unter [3]. Peak stellt eine Reihe von „handhabbaren“ Atlas-Bauteiltestern her, die gute Kritiken ernten und preisgünstig sind. Insbesondere der *DCA75 Pro Advanced Semiconductor Analyser* (siehe **Kasten**) identifiziert so gut wie jedes zwei- oder dreipolige Halbleiterbauelement und ist sehr empfehlenswert. Wenn Sie detailliertere Tests als die oben beschriebenen durchführen müssen, lohnt sich die Investition in ein solches Testgerät. Elektor bietet seit kurzem den Multifunktion-Bauteiltester Joy-iT LCR-T7 [5] an, der zwar nicht die Spezifikationen des Peak-Testers erreicht, aber auch nur etwa ein Viertel kostet und so ein hervorragendes Preis-Leistungs-Verhältnis bietet.

Eine Tisch- oder Handlupe mit LED-Beleuchtung und/oder ein USB-Mikroskop ist praktisch, um kleine Bauteile auf Wert, Polarität oder Ausrichtung zu prüfen. In vielen der Tests wird eine Stromversorgung benötigt. Als Minimum ist eine variable Spannungsversorgung von 0...30 V mit einer variablen Strombegrenzung von 10 mA bis mindestens 500 mA oder mehr ausreichend.

Die Anwendung dieser einfachen Techniken und einiger selbstgebauter Prüfgeräte wird Ihnen das Leben sehr erleichtern, wenn Sie „gerettete“ Bauteile für Ihre Projekte wiederverwenden wollen. ◀

210279-02

## Testbericht: Leistungsfähiger portabler Halbleiter-Tester PEAK DCA75



Vor einiger Zeit habe ich mir den Halbleiter-Tester PEAK DCA75 zugelegt. Ich muss zugeben, dass ich zuvor ein wenig gezögert habe, weil ich mich fragte, ob ein solch kleines Testgerät mit nur zwei Tasten seinen Preis wert sein kann.

Ich hätte jedoch nicht positiver überrascht werden können. Das Gerät bietet alles, was es verspricht, und ist zudem ganz einfach zu bedienen. Die Benutzeroberfläche ist einfach genial und lässt keine Fragen offen. Sie schließen einen beliebigen Halbleiter an und den DCA75 erkennt, was es ist, gibt einige grundlegende Parameter an und identifiziert zudem die Bauteilanschlüsse. So sehr ich mich auch bemühte, ich konnte den DCA75 nicht überlisten. Eines der ersten Bauteile, die ich ausprobierte, war der „0V8F-Transistor“, über den ich schon in meinem Artikel über die Identifizierung von Bauteilen geklagt hatte [1]. Ich konnte ihn trotz großer Bemühungen nicht identifizieren oder testen, so dass ich sogar die Möglichkeit nicht ausschloss, dass es sich um einen FET handeln könnte. Der DCA75 identifizierte das Bauteil jedoch sofort als Triac. Das allein schon hat mich überzeugt,

dass der DCA75 sein Gewicht in Gold wert ist!

Der DCA75 hat aber doch einige kleinere Einschränkungen: Er kann keine Z-Dioden und Spannungsregler mit mehr als 8...9 V testen und einige sehr exotische Halbleiter wie Unijunction-Transistoren nicht identifizieren, aber das sind keine ernsthaften Nachteile für den durchschnittlichen Anwender.

Mit einem PC und der mitgelieferten Software kann der am ebenfalls mitgelieferten USB-Kabel angeschlossene DCA75 auch Kennlinien von vielen Bauelementen erstellen. Alles in allem ein wirklich vielseitiges Testgerät, das Sie in Ihr Arsenal aufnehmen sollten. Es gibt einen kleinen Bruder, den DCA55, der etwa die Hälfte kostet, aber ich würde Ihnen raten, großzügig zu sein und sich den DCA75 zuzulegen.

### Ein Beitrag von

Text und Fotografien: David Ashton  
Redaktion: Clemens Valens  
Übersetzung: Rolf Gerstendorf  
Layout: Harmen Heida

### Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel?  
Kontaktieren Sie Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).



### PASSENDE PRODUKTE

- > **Peak Atlas DCA75 Pro Advanced Semiconductor Analyser (SKU 17567)**  
[www.elektor.de/17567](http://www.elektor.de/17567)
- > **Miniware DT71 Digital Tweezers (SKU 19422)**  
[www.elektor.de/19422](http://www.elektor.de/19422)
- > **Joy-IT LCR-T7 Multi-function Component Tester (SKU 19709)**  
[www.elektor.de/19709](http://www.elektor.de/19709)
- > **Elektor 2 MHz LCR Meter Kit (SKU 19883)**  
[www.elektor.de/19883](http://www.elektor.de/19883)

### WEBLINKS

- [1] D. Ashton, „Identifizierung von Bauteilen“, Elektor 3-4/2022: [www.elektormagazine.de/210024-02](http://www.elektormagazine.de/210024-02)
- [2] Review: MiniPro-Programmer TL866A: [www.elektormagazine.de/magazine/elektor-178/59754](http://www.elektormagazine.de/magazine/elektor-178/59754)
- [3] H. Baggen, „Digitale Pinzette Miniware DT71“, Elektor 7-8/2021: [www.elektormagazine.de/magazine/elektor-178/59754](http://www.elektormagazine.de/magazine/elektor-178/59754)
- [4] Leistungsfähiger portabler Halbleiter-Tester PEAK DCA75:  
[www.elektormagazine.de/news/leistungsfahiger-portabler-halbleiter-tester-peak-dca75](http://www.elektormagazine.de/news/leistungsfahiger-portabler-halbleiter-tester-peak-dca75)
- [6] Joy-iT LCR-T7 Multi-Funktions tester: [www.elektormagazine.de/articles/joy-it-lcr-t7-multi-funktions-tester](http://www.elektormagazine.de/articles/joy-it-lcr-t7-multi-funktions-tester)

# Stromsparende Maulwurfscheuche

Ein ATtiny13 ersetzt einen 555er



Von **Gerhard Dürr** (Deutschland) und **Luc Lemmens** (Elektor)

Maulwurfshügel sind lästig, kommerzielle elektronische Maulwurfscheuchen wegen des hohen Batterieverbrauchs umständlich bis teuer. Mit einem Hardware-Upgrade aber kann die Lebensdauer der Batterien um das Zehnfache erhöht werden!

Wenn Sie einen Garten mit Rasen haben, vor allem in einer ländlichen Gegend, ist die Gefahr von Maulwürfen unangenehm groß. Die Hügel, die diese possierlichen Tierchen bilden, können Ihren Rasen in Nullkommanichts ruinieren. Zwar deutet dies auf einen sauberen, fruchtbaren Boden hin, aber das ist wenig tröstlich. Sie werden die Tiere nie oder nur selten zu Gesicht bekommen, denn sie halten sich in der Regel unterirdisch in den von ihnen gegrabenen Tunneln auf, wo sie nach Würmern und Insekten suchen, die auf ihrem Speiseplan stehen. Die Erde, die sie beim Graben bewegen, muss natürlich irgendwo hin, und sie schieben sie nach oben und bilden so Maulwurfshügel. Es wurden zahlreiche Methoden erdacht, um der Maulwürfe Herr zu werden, vom brachialen Totschlagen und Ersäufen bis zu den so genannten elektronischen Maulwurfscheuchen; Geräten, die mit Hilfe von (Schall-) Vibrationen die unterirdischen Aktivitäten verhindern oder beenden. Es handelt sich dabei meist um ein Rohr mit einer Spitze, das in den Boden gesteckt wird, und in der Spitze befindet sich eine Schallquelle (meist in Form eines elektronischen Gleichstromsummers),

die Maulwürfe und andere Bestien wie Wühlmäuse und Ratten verjagen soll.

Einige Hersteller behaupten, dass die von ihnen hergestellten Scheuchen Ultraschallschwingungen erzeugen. Vielleicht gibt es so etwas tatsächlich, aber die meisten Geräte, die wir untersucht haben, erzeugen Töne mit Frequenzen im niedrigen Kilohertzbereich oder sogar darunter, also weit unterhalb des Ultraschallbereichs. In den meisten Fällen wäre also die Bezeichnung *Sonic Repeller* (ohne Ultra-) angemessener. Der Ton wird in Abständen von etwa 30 Sekunden für etwa eine halbe Sekunde eingeschaltet. Diese Impulse wären für einen Maulwurf so unangenehm, dass er seine Koffer packen und in höchster Eile das Weite suchen würde, behaupten die Hersteller, und es gibt sogar Leute, die von der Wirksamkeit dieser Geräte absolut überzeugt sind, während andere feststellen, dass es sich dabei nur um Betrug handeln und dass nur die Hersteller und Verkäufer davon profitieren würden. Wir werden uns zur Wirksamkeit kommerzieller Maulwurfscheuchen nicht äußern! Einige Gedanken zur Maulwurfabwehr im Allgemeinen

finden Sie im **Textkasten**. In diesem Artikel geht es nur um eine Möglichkeit, den Strombedarf einer batteriebetriebenen schallzeugenden Scheuche zu reduzieren, um die Kosten für die Batterien zu senken.

## Wie es funktioniert

Normalerweise enthält diese Art von Maulwurfsabwehr eine Timer-Schaltung und eine Art Schallgeber, in den meisten Fällen einen Piezo-Summer, aber wir haben auch schon Geräte mit Vibrationsmotoren wie im Handy gefunden. Jeder Elektroniker denkt beim Stichwort „Timerschaltung“ natürlich sofort an einen 555, das wahrscheinlich berühmteste aller ICs, das übrigens kürzlich seinen 50-jährigen Geburtstag feiern durfte.

Und tatsächlich hat der Entwickler der hier besprochenen Schaltung eine Maulwurfscheuche erworben, in der ein 555 als bistabiler Multivibrator verwendet wurde. Der größte Teil des Stroms fließt natürlich, wenn der Ton erzeugt wird, aber es ist auch erheblich Energie erforderlich, um den 555er am Laufen zu halten, wenn der Summer ausgeschaltet ist. Zwar könnte man die Schaltung optimieren und die Lebensdauer der Batterien etwas verlängern, indem man beispielsweise die CMOS-Version des 555 verwendet und/oder die Werte anderer Bauteile ändert, aber der Entwickler hat einen anderen Ansatz gewählt. Die komplette Timer-Schaltung wird durch einen kleinen Mikrocontroller ersetzt, der die meiste Zeit seines Daseins verschläft. Sein Watchdog-Timer weckt ihn nur etwa alle 30 s auf und lässt den Summer für eine Sekunde ertönen, bevor der Controller wieder in den Schlafmodus übergeht. Dadurch wird der Strombedarf auf ein gutes Zehntel reduziert! Statt 4 mA der ursprünglichen 555-Timer-Schaltung zehren mit dem Mikrocontroller an Timers statt nur noch durchschnittlich 430  $\mu$ A an der Kraft der Batterie. Die Batterielebensdauer wird so theoretisch von weniger als 200 Tagen auf etwa 1450 Tage verlängert! In der Praxis wird dies aufgrund von Faktoren wie der Selbstentladung natürlich weniger sein, aber dennoch ist dies eine große Verbesserung.

## Die neue Hardware

Der Schaltplan der Mikrocontrollerschaltung ist in **Bild 1** dargestellt. IC1, ein ATtiny13 von Microchip (früher Atmel), ist das Herz der Timerschaltung. Sein interner Programmspeicher kann über einen an K2 angeschlossenen ISP-Programmer geflasht werden. Alternativ kann dieser 6-polige Anschluss weggelassen werden, wenn der Mikrocontroller programmiert wird, bevor er in die Fassung für IC1 eingesetzt wird. Die Schaltung wird von den Batterien versorgt, in diesem Fall von vier D-Zellen in Reihe, also mit 6 V, die an K1 (plus) und K3 (minus) anliegen. Bei vier frischen 1,5-V-Batterien und ohne oder sehr geringer Last dürfte die Gesamtspannung etwas höher sein, etwa 6,5 V. Da der ATtiny13 eine maximale Versorgungsspannung von 5,5 V besitzt, sorgt D1, eine rote Standard-LED (vorzugsweise *kein* hocheffizienter Typ) mit einer Durchlassspannung von etwa 1,3 V für eine Absenkung der Versorgungsspannung auf einen für den Mikrocontroller sicheren Wert, selbst bei einem Durchlassstrom von nur 1  $\mu$ A. Alternativ kann D1 durch zwei oder drei Standarddioden 1N4148 in Reihe ersetzt werden. Ein linearer Spannungsregler sollte für diesen Zweck nicht verwendet werden, da er den Strombedarf des Maulwurfsabwehrsystems erhöhen würde. C1 und C2 sind Entkopplungskondensatoren für die Spannungsversorgung des Mikrocontrollers.

Der Summer wird aus der Originalhardware gerettet und an die Pins K4/K5 (plus) und K6 (minus) angeschlossen. Einige Summer sind

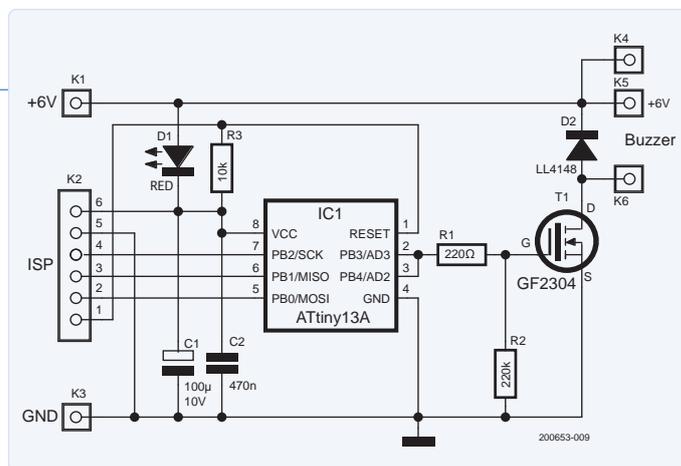


Bild 1. Der Schaltplan des Timer-Upgrades

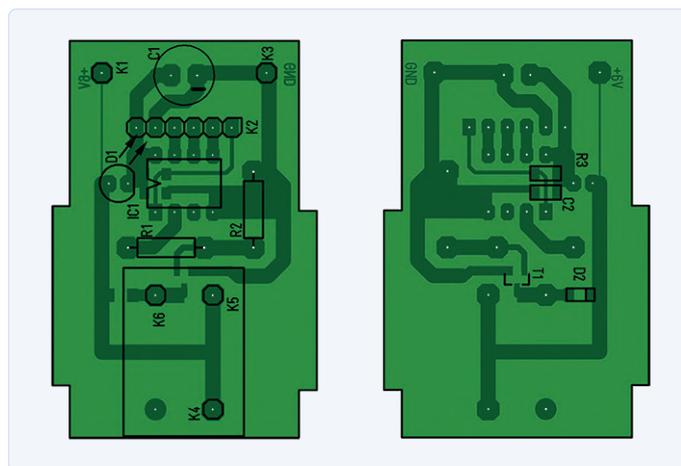


Bild 2. Das Layout der kleinen Platine.

induktiv und können Spannungsspitzen erzeugen, besonders beim Ausschalten. Die Freischaltdiode D2 verhindert, dass diese Spikes den Transistor T1 beschädigen. Die Wahl dieses Schalttransistors ist überhaupt nicht kritisch. Da der Strom durch den Summer weniger als 10 mA beträgt, kann auch ein Standard-BC547 verwendet werden, wenn der Basiswiderstand R1 auf 4,7 k $\Omega$  und R2 auf 10 k $\Omega$  geändert wird. Jeder Logikpegel-NMOS-FET ist ebenfalls geeignet, wenn man R1 durch eine Drahtbrücke ersetzt und R2 weglässt.

## Es gibt sogar eine Platine

Der Schaltplan wurde mit der kostenlosen Version von *Target3001!* entworfen, und der Autor hat sogar ein kleines Platinenlayout für das Mikrocontroller-Upgrade erstellt (siehe **Bild 2**). Die Designdateien stehen auf der Elektor-Labs-Seite dieses Projekts zum Download bereit [1]. In Anbetracht der vielen verschiedenen Marken von Maulwurfsabwehrgeräten, die auf dem Markt erhältlich sind, wäre es reiner Zufall, wenn die Platine genauso gut in Ihr Gerät passt wie auf den Bildern, die die Autoren gemacht haben (**Bild 3**). Dann sollte man sich die gute Gelegenheit nicht entgehen lassen, Target3001! mit diesem relativ einfachen Projekt auszuprobieren und zu versuchen, die Platine Ihren Bedürfnissen anzupassen. Es dürfte aber auch nicht allzu lange dauern, Schaltplan und Platine in einem anderen ECAD-Programm neu zu zeichnen. Auf der Kupferunterseite der Platine sind ein paar SMD-Bauteile aufgelötet, die auch ohne Änderung der Platine leicht durch bedrahtete Bauteile ersetzt werden können.



Bild 3a. Die Maulwurfscheuche im Überblick.

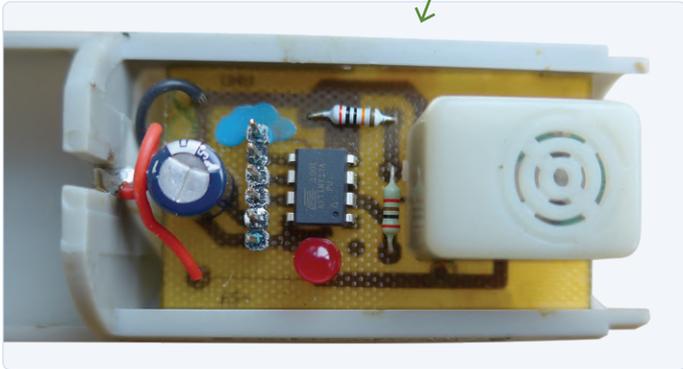


Bild 3b. Die neue Platine.

## Ohne Firmware läuft nichts

Der Quellcode (**Listing 1**) für den ATtiny13 wurde in AVR Studio 4.19 mit installiertem AVR-GCC-Compiler entwickelt. Obwohl diese IDE veraltet ist, steht sie immer noch zum Download zur Verfügung, aber es dürfte nicht schwierig sein, den Quellcode in modernere Entwicklungswerkzeuge zu portieren. Der Quellcode, das AVR-Studio-Projekt und die

## Maulwürfe in Ihrem Garten? Was sollten Sie (nicht) tun?

Wenn Sie im Internet recherchieren, was Sie gegen Maulwürfe im Garten tun können, oder wie Sie verhindern können, dass sie überhaupt auf Ihr Grundstück kommen, werden Sie überrascht sein, was im Laufe der Jahre alles erfunden wurde, um diesen Kreaturen das Leben schwer zu machen. Viele dieser Mittel werden von erfahrenen Experten in ihrer Wirksamkeit angezweifelt. Dass eine Methode nicht funktioniert, weiß man aber erst dann mit Sicherheit, wenn der schöne englische Rasen trotz allem plötzlich von Maulwurfshügeln übersät ist.

Wir sind zwar keine Maulwurfsexperten, aber nach den Informationen, die wir gefunden haben, scheint der beste Rat zu sein, gar nichts zu tun. Wenn es Maulwurfshügel gibt, hat ein Maulwurf bereits seinen Bau angelegt, der sich in der Regel nur dann ausdehnt, wenn es nicht genügend Nahrung gibt. Wenn Sie die Mini-Berge als störend empfinden, schaufeln Sie am besten vorsichtig die Erde weg, denn wenn Sie darauf klopfen oder hämmern, bricht der Maulwurfsgang zusammen und es besteht eine gute Chance, dass das Tierchen den Schaden ausbessert und dabei einen neuen Maulwurfshügel nebenan anlegt.

HEX-Datei können von der Elektor-Labs Seite dieses Projekts heruntergeladen werden [1]. Mit der HEX-Datei lässt sich der Programmspeicher des ATtiny13 ohne Kompilierung flashen. Die Konfigurations-Fuses dieses Mikrocontrollers bleiben auf den Werkseinstellungen, müssen also nicht programmiert werden.

Das C-Programm besteht größtenteils aus Assembler-Anweisungen. Nach dem Einschalten werden die GPIOs konfiguriert, die Timer, der ADC, die Pull-ups und der Analogkomparator abgeschaltet und der Schlafmodus auf Powerdown gesetzt. Der Code tritt dann in eine Endlosschleife ein (`while(1)`), die die folgenden Operationen durchführt:

- Der Summer wird eingeschaltet.
- Der Watchdog wird so eingestellt, dass er nach 128k Zyklen einen Interrupt erzeugt, was bei einem Takt von 128 kHz einem Interrupt-Intervall von (ungefähr) einer Sekunde entspricht.
- Der Interrupt wird aktiviert und der ATtiny schaltet mit dem Assembler-Befehl `sleep` in den *Powerdown*-Modus.
- Nach einer Sekunde weckt der Watchdog-Interrupt den Mikrocontroller aus dem Ruhezustand auf.
- Der Summer wird ausgeschaltet und der Watchdog-Interrupt auf sein maximales Intervall von 1024k Zyklen, also acht Sekunden, geändert. Da wir wollen, dass die Maulwurfscheuche etwa eine halbe Minute lang schweigt, muss der Watchdog drei Interrupts erzeugen, bevor `while(1)` wieder mit dem Einschalten des Summers beginnt.

## Maulwürfe weg, Nachbar sauer?

Mit diesem Upgrade der Maulwurfscheuche wird die Batterielebensdauer der vier D-Zellen verlängert, die Wirksamkeit des Gerätes selbst aber nicht verändert. Wie bereits erwähnt, gibt es viele Diskussionen zwischen Menschen, die diese Maulwurfabwehrgeräte für großartig halten, und anderen, die glauben oder sogar sicher wissen, dass sie überhaupt keine Wirkung haben. Die große Anzahl von Marken und Anbietern deutet darauf hin, dass es immerhin ein gutes Geschäft ist. Auf jeden Fall ist es ein Mittel, das keine allzu großen negativen Auswirkungen auf die Umwelt hat. Nur manche genauest zuhörenden Nachbarn könnten sich von den sich wiederholenden leisen Geräuschen belästigt fühlen. ◀

200653-02



## STÜCKLISTE

### Widerstände:

R1 = 220 Ω  
 R2 = 220 k  
 R3 = 10 k

### Kondensatoren:

C1 = 220 µ/25 V  
 C2 = 470 n

### Halbleiter:

D1 = rote LED  
 D2 = LL4148  
 T1 = GF2304 (siehe Text)  
 IC1 = ATtiny13A (DIL-8)

### Außerdem:

K2 = 1x6-polige Stiftleiste (Raster  
 2,54 mm)

## Ein Beitrag von

Entwurf: Gerhard Dürr

Text: Gerhard Dürr,

Luc Lemmens

Illustrationen: Gerhard Dürr,

Patrick Wiolders, Luc Lemmens

Redaktion: Jens Nickel,

C. J. Abate

Übersetzung: Rolf Gerstendorf

Layout: Giel Dols

## Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).



## Listing 1. Der vollständige Quellcode.

```
// Controlling the molechaser-buzzer with the watchdog-timer
// * MC ATtiny13A, all fuse bytes default
// * PB3 and PB4 HIGH: buzzer ON
// * Configurable with watchdog-timer-prescaler
// * Selected: 128K-cycles(~1s) ON, 3*1024K-cycles(~24s) OFF
// * Current consumption: powerdown ~6µA, buzzing ~7.2mA, average 0.43mA
#include <avr/io.h>
#ifndef F_CPU
#warning "Defining F_CPU 1.2MHz"
#define F_CPU 1200000UL
#endif
#include <avr/interrupt.h>
int main(void) {
    uint8_t i=0;
    DDRB = 1<<PB0|1<<PB1|1<<PB2|1<<PB3|1<<PB4;
    PRR = 1<<PRTIM0|1<<PRADC; // Power Reduction Timer0, ADC
    MCUCR = 1<<PUD|1<<SE|1<<SM1|0<<SM0; // Pullup Disable, Sleepmode: Powerdown
    ACSR = 1<<ACD; // Analog Comparator Disable
    while(1) {
        i = 0;
        PORTB |= 1<<PB3|1<<PB4; // buzzer ON
        // change Watchdog-Interrupt to 1s (measured: 1.14s @ 4.5V, 22°C)
        MCUSR = 0; // clear Reset-Flags
        WDTCSR = 1<<WDCE |0<<WDE; // Watchdog Change Enable
        // Watchdog Timer Prescaler: 128K-cycles(~1s), Watchdog Timer Interrupt Enable
        WDTCSR = 1<<WDTIE|0<<WDE|1<<WDP2|1<<WDP1;
        sei();
        asm volatile("sleep");
        asm volatile("wdr"); // Watchdog Reset, wait for WDT-ISR(~1s)
        PORTB &= ~(1<<PB3|1<<PB4); // buzzer OFF
        // change Watchdog-Interrupt to 3*8s: (measured: 27.9s @ 4.5V, 22°C)
        asm volatile("wdr"); // Watchdog Reset
        MCUSR = 0; // clear Reset-Flags
        WDTCSR = 1<<WDCE|0<<WDE; // Watchdog Change Enable
        // Watchdog Timer Prescaler: 1024K-cycles(~8s), Watchdog Timer Interrupt Enable
        WDTCSR = 1<<WDTIE|0<<WDE|1<<WDP3|1<<WDP0;
        while(i < 3) { // wait 24s
            i++;
            asm volatile("sleep");
            asm volatile("wdr"); // Watchdog Reset, wait for WDT-ISR(~8s)
        }
    }
}
EMPTY_INTERRUPT(WDT_vect); // awaking from powerdown-mode
```



**PASSENDES  
PRODUKT**

➤ **Diamex EXA-Prog-Paket  
 (SKU 19912)**  
[www.elektor.de/19912](http://www.elektor.de/19912)

## WEBLINK

[1] Dieses Projekt auf Elektor Labs: <https://bit.ly/34htTXN>



# Lichtschalter DeLux

Eine Lösung für hochpräzises lichtgesteuertes Schalten

Von Clemens Valens (Elektor)

Lichtgesteuerte Schalter gibt es in Hülle und Fülle und sie kosten im Einzelhandel um die 10 €, aber die meisten von ihnen ändern ihren Zustand nur „irgendwann“ in der Dämmerung. Doch manchmal benötigt man eine höhere Präzision und mehr Kontrolle, als diese billigen Schalter bieten können. Benötigen Sie auch solch eine lux-genaue Lichtsteuerung? Wenn ja, dann ist dieses Projekt genau das Richtige für Sie!

Es gibt viele, viele Entwürfe für lichtgesteuerte Schalter, und die meisten von ihnen funktionieren hervorragend für die Aufgaben, für die sie entworfen wurden. Diese bestehen in der Regel darin, eine Lampe einzuschalten, wenn die Umgebungshelligkeit unter einen bestimmten Schwellenwert fällt, und das Licht auszuschalten, wenn es wieder hell wird. Manchmal ist auch eine Zeitschaltuhr hinzugefügt.

Man könnte meinen, dass diese Konstruktionen alle möglichen Anwendungen abdecken, doch das stimmt nicht. Der Grund dafür ist die mangelnde Präzision, denn sie basieren auf einem LDR oder einem Fototransistor und schalten in der Regel irgendwann in der Abenddämmerung ein und in der Morgendämmerung aus. Die Umgebungshelligkeit variiert in der Realität jedoch viel stärker und viel unregelmäßiger.

## Helligkeit ist subjektiv

Für den Menschen wirkt die Helligkeit des Tageslichts oder, wie es fachlich korrekt

genannt wird, die Beleuchtungsstärke ziemlich konstant. Natürlich nehmen wir Schwankungen aufgrund von Wolken und Sonne wahr, aber wir sind nicht sehr empfindlich dafür. Der Grund dafür ist die logarithmische Reaktion des Auges auf Helligkeit. An einem bewölkten Tag kann die Helligkeit zwischen 5.000 lx und 10.000 Lux schwanken, und doch sieht sie für uns fast gleich aus. Direktes Sonnenlicht erreicht Werte von über 25.000 lx, was wir natürlich als besonders hell wahrnehmen, aber wir empfinden es nicht als dreimal oder mehr so hell.

Pflanzen hingegen reagieren viel empfindlicher auf die Beleuchtungsstärke als Menschen, ist doch Licht ihr Lebenselixier. Landwirte wissen das und bestrahlen besonders lichthungrige Pflanzen auch tagsüber mit künstlichem Licht, um den Ertrag zu steigern. An sonnigen Tagen ist dies in der Regel nicht nötig, aber an bewölkten Tagen kann es helfen. Wenn dies schon nicht besonders ökologisch ist, so benötigen

Landwirte exakte lichtgesteuerte Schalter, die Unterschiede in der Beleuchtungsstärke mit größerer Präzision erkennen können als ein LDR oder ein Fototransistor, damit wenigstens die Ökonomie stimmt.

Heutzutage gibt es Lichtsensoren, die die Beleuchtungsstärke direkt in einen Wert in Lux umwandeln, und dies mit einer Auflösung von bis zu 16 Bit. Einige dieser Sensoren messen nicht nur sichtbares Licht, sondern auch UV-A und UV-B. Mit einem solchen Sensor ist es recht einfach, einen hochpräzisen lichtgesteuerten Schalter zu bauen.

## Hochpräziser Umgebungslichtsensor

Ein beliebter Lichtsensor ist der VEML7700 von Vishay. Dabei handelt es sich um einen hochpräzisen Umgebungslichtsensor mit I<sup>2</sup>C-Schnittstelle, der auf einem kleinen Modul für ein paar Euro zu haben ist. Aus der gleichen Familie stammt auch der VEML6075, ein UV-A- und UV-B-Lichtsensor, ebenfalls mit I<sup>2</sup>C-Schnittstelle.

Dank seiner digitalen I<sup>2</sup>C-Schnittstelle benötigt der Sensor keinen externen Analog-Digital-Wandler und kann stattdessen direkt an die meisten Mikrocontroller angeschlossen werden. Seine Messdaten stehen in zwei 16-Bit-Registern zur Verfügung, eines für Umgebungslicht (ambient light, AL) und eines für Weißes Licht. Das Umgebungslicht ist das vom Menschen wahrgenommene Licht im spektralen Bereich von etwa 450...650 nm, Weißes Licht deckt ein breiteres Spektrum von 250...950 nm ab, umfasst also auch den



gesamten ultravioletten und den nahen Infrarot-Bereich. Pflanzen sind keine Menschen, daher hängt das zu verwendende Ausgangsformat von der jeweiligen Anwendung ab.

Auch die Sensorempfindlichkeit ist wichtig. Der VEML7700 besitzt eine Auflösung von 0,005 lx pro LSB und misst maximal 167.000 lx (Minimum ist 0,01 lx). Für einen solch großen Bereich bei gegebener Auflösung wären 25-bit-Werte erforderlich. Da aber der Sensorspeicher auf 16 Bit beschränkt ist, können Empfindlichkeitsbereiche (gain) angegeben werden, um die 16-bit-Werte in den richtigen Bereich zu bringen. Dank seiner hohen Empfindlichkeit kann der Sensor auch hinter Oberflächen mit geringer Lichtdurchlässigkeit (das heißt, dunklen Oberflächen) eingesetzt werden und dennoch brauchbare Ergebnisse liefern. Für schwache Lichtverhältnisse bietet der Sensor eine Integrationszeit von bis zu 800 ms. Schließlich können niedrige und hohe Schwellenwerte eingestellt werden, die Interrupts auslösen, was die Erstellung von Alarmen oder automatisches Schalten erleichtert.

### Einsatz am ESP32-Thermostat

Das für meine Experimente verwendete VEML7700-Modul habe ich bei Adafruit gekauft. Eine gute Plattform für das Modul ist das ESP32-verbundene Thermostat [1] (auch bekannt als Automator, siehe [2]). Das Modul hat fünf Anschlüsse, aber da es zwei Stromversorgungspins gibt, brauchen wir nur vier davon. Diese vier Pins haben die gleiche Reihenfolge wie der Anschluss des OLED-Displays am Thermostat, so dass wir das Modul auf K9 stecken können. Achten Sie darauf, dass der VIN-Pin ganz links nicht angeschlossen ist und das Modul nach oben zeigt (im Gegensatz zum OLED-Display, siehe **Bild 1**).

### Software mit ESPHome

Nachdem der Sensor mit dem ESP32 verbunden ist, müssen wir eine Software erstellen, damit alles funktioniert. Da das Ziel eine Art lichtgesteuerter Schalter ist, wäre es eine logische Wahl, eine Heimautomatisierungsplattform zu verwenden, und mein Favorit ist bekanntermaßen ESPHome. Elektor hat mehrere Projekte veröffentlicht, bei denen ESPHome zum Einsatz kam [3][4], so dass Sie vielleicht schon wissen, wie man es benutzt und konfiguriert, aber dieses Projekt führt ein Konzept ein, das ich bisher nicht behandelt

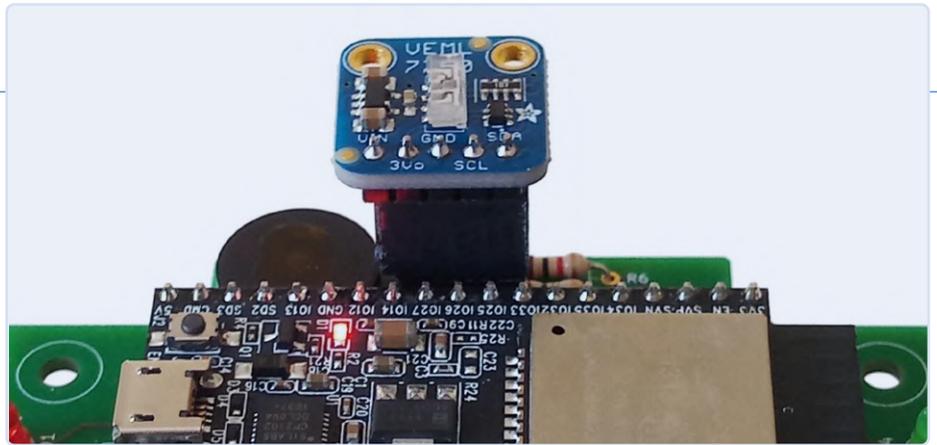


Bild 1. Das 5-polige Modul VEML7700 wird auf den 4-poligen Steckverbinder K9 gesteckt. Der VIN-Pin ist nicht angeschlossen, auch wenn es so aussehen mag, sondern hängt links in der Luft!

habe: die Erstellung eines benutzerdefinierter Sensors. Wenn Sie mit ESPHome noch nicht vertraut sind, empfehle ich Ihnen, zuerst [3] und [4] anzusehen.

### Ein benutzerdefinierter Sensor

ESPHome unterstützt viele Sensoren, aber (zum Zeitpunkt, als ich dies geschrieben habe) noch nicht den VEML7700. Es kennt zwar andere Sensoren für Beleuchtungsstärke, aber leider nicht diesen. Das ist jedoch kein Problem, denn ESPHome bietet eine Methode, einen eigenen Sensor hinzuzufügen. Dazu muss man allerdings etwas C++-Code schreiben, was die Angelegenheit ein wenig komplizierter macht.

Wie zuvor (siehe [3] und [4]) müssen wir in der ESPHome-Konfigurationsdatei eine *sensor*-Sektion für den VEML7700 deklarieren. Die *platform* des Sensors muss als *custom* definiert sein. Damit wird ESPHome mitgeteilt, dass Sie alle Details für den Sensor zur Verfügung stellen wollen.

Als nächstes folgt ein *lambda*-Abschnitt aus ein paar Zeilen C++-Code, um ESPHome anzuweisen, einen von uns entwickelten Sensor zu registrieren (den wir *veml7700* nennen). Wir müssen auch die Datenströme angeben, die unser Sensor erzeugt. Neben Registern für Weißes Licht und AL übernimmt die zum Adafruit-Modul gehörende Bibliothek die Berechnung des direkten Lux-Wertes. Die Reihenfolge der drei Ausgänge: ALS, Lux und Weißes Licht ist wichtig und muss beachtet werden, wenn an anderer Stelle in der YAML-Datei auf sie verwiesen wird.

Wir fahren mit normalen YAML-Anweisungen fort, um die Sensorausgänge weiter zu spezifizieren. Dies geschieht in einem Abschnitt *sensors* (Plural!), in dem wir für jeden Datenstrom den Namen, die Einheiten und die Anzahl der zu verwendenden Dezimalstellen angeben können. Die Reihenfolge ist dieselbe wie in der *return*-Anweisung im Lambda-Abschnitt.

Nur die Lux-Daten haben eine Einheit (lx), und es hat keinen Sinn, bei irgendeinem Wert Dezimalstellen anzugeben, so dass wir deren Anzahl auf Null setzen.

Als letztes muss angegeben werden, wo ESPHome den Treiber für den benutzerdefinierten Sensor finden kann. Dies geschieht in der Sektion *esphome* am Anfang der Konfigurationsdatei, wo wir eine Datei (*veml7700.h*) zum *includes*-Unterabschnitt hinzufügen. Auf dem System, auf dem ESPHome läuft, sollte sich diese Datei im selben Ordner befinden wie die entsprechende YAML-Datei.

### Der schwierige Teil

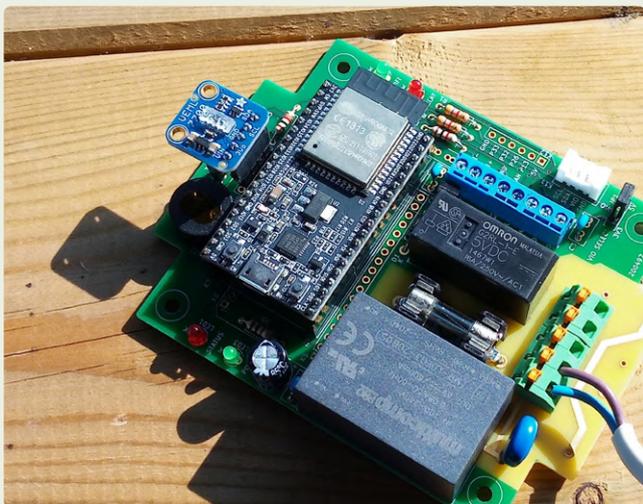
Geschafft? Nicht ganz. Den schwierigen Teil haben wir uns für das Ende aufgespart, nämlich die Erstellung des Treibers für den benutzerdefinierten Sensor. Dieser Treiber muss den ESPHome-Standards für *components* entsprechen (ein Sensor ist eine solche Komponente), das heißt, er muss bestimmte Funktionen bereitstellen, die ESPHome von seinen Komponenten erwartet, und er muss mit dem Sensor selbst kommunizieren.

Für die zweite Aufgabe können wir uns ganz auf Adafruit verlassen, die nicht nur das VEML7700-Modul herstellen und anbieten, sondern auch eine Arduino-Bibliothek dafür entwickelt haben [5]. Unser Treiber muss dagegen die Schnittstelle zwischen ESPHome und der Arduino-Bibliothek bereitstellen.

### Installation einer Bibliothek eines Drittanbieters

ESPHome bietet einen Mechanismus zum Hinzufügen von Open-Source-Community-Bibliotheken: Fügen Sie einfach den genauen (!) Namen der Bibliothek im *esphome*-Bereich in die Untersektion *libraries* ein. In unserem Fall ist dies „Adafruit VEML7700 Library“. Wenn ESPHome die Konfigurationsdatei verarbeitet, installiert es die Bibliothek, wenn noch nicht gesche-

## Einige Ergebnisse

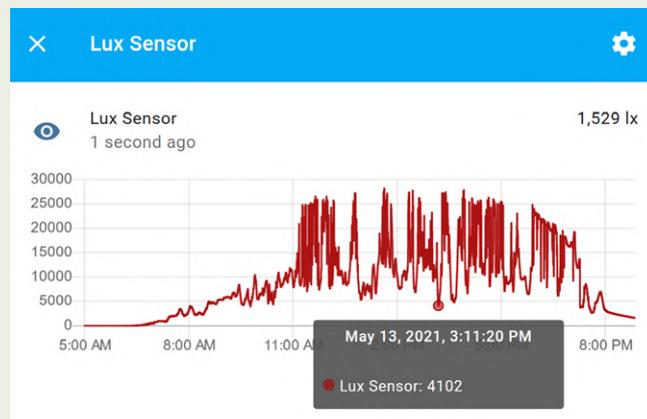


Zur Mittagszeit, mit einer Empfindlichkeit von 0,125 und einer Integrationszeit von 100 ms, meldete der Sensor Werte von etwa 48.000 für ALS und etwas mehr als 30.000 für weißes Licht. Diese Werte entsprechen einer Lichtintensität von etwa 22.000 lx. Um 10:30 Uhr habe ich unter den gleichen Bedingungen 16.800 lx gemessen, eine Differenz von 5.200 lx. Meine subjektiven Augen konnten so gut wie keinen Helligkeitsunterschied erkennen.



Entwicklung der Helligkeit an einem sonnigen Tag im Juli. Das „Prellen“ auf der linken Seite ist auf den Schatten eines Baumes zurückzuführen. Von Mittag bis 18 Uhr nimmt die Helligkeit langsam

ab, was man nicht wirklich merkt. Die kurzen Einbrüche werden von Wolken verursacht. Der starke Abfall ab 18 Uhr wird durch das Haus verursacht, der den Sensor für etwa eine Stunde abschattet. Danach nimmt die Lichtintensität bis zur Dunkelheit kontinuierlich ab. Menschen empfinden bereits 5.000 lx als ziemlich hell.



An einem teilweise bewölkten Tag im Mai schwankte die Beleuchtungsstärke zwischen etwa 4.000 lx und 27.000 lx.

Alle Diagramme wurden mit Hilfe von Home Assistant auf einem Breitengrad von lat = 47°N erstellt.



### PASSENDE PRODUKTE

- > ESP32 DevKitC SKU 18701  
[www.elektor.de/18701](http://www.elektor.de/18701)
- > I<sup>2</sup>C-OLED-Display 0,96" 128x64 SKU 18747  
[www.elektor.de/18747](http://www.elektor.de/18747)
- > Koen Vervloesem, Getting Started with ESPHome Buch, SKU 19738: [www.elektor.de/19738](http://www.elektor.de/19738)  
E-Buch, SKU 19739: [www.elektor.de/19739](http://www.elektor.de/19739)

hen (**Bild 2**), bevor es an die Kompilierung geht. In Ihrem eigenen Treiber können Sie die Bibliothek so einfach wie jede andere Bibliothek einbinden.

Ich habe mich mit dieser Funktion noch nicht tiefgreifend beschäftigt und kenne die Kriterien nicht, die eine Bibliothek eines Drittanbieters erfüllen muss, um auf diese Weise verwendet zu werden. Sie sollten die Dokumentation *platform.io* konsultieren, wenn Sie weitere Details erfahren möchten, da dies die von ESPHome verwendete Toolchain ist.

### Ein gekapselter Arduino-Sketch

Unser Treiber ist eine C++-Klasse mit mindestens einem Konstruktor und einer

Funktion namens *update*. Wir brauchen auch eine Funktion *setup*, damit wir den Adafruit-Treiber initialisieren können. Die Funktionen *setup* und *update* der Klasse tun das, was normalerweise in den Funktionen *setup* und *loop* eines typischen Arduino-Sketches für den Adafruit-Treiber geschehen würde. Im Grunde kapselt unsere Klasse einen Arduino-Sketch einschließlich globaler Variablen und fügt ihm einige ESPHome-spezifische Dinge hinzu. Für ESPHome müssen wir Calls zu *publish\_state* für jeden Datenstrom (ALS, Lux und Weißlicht) in die Funktion *update* einfügen. Dadurch werden die Daten für den Rest der Welt verfügbar gemacht. Die Reihenfolge der Calls muss dieselbe sein wie

in der Return-Anweisung in der *lambda*-Untersekktion der *sensor*-Sektion.

Da unsere Klasse von der *PollingComponent*-Klasse erbt, die ihrerseits von der *Component*-Klasse erbt, stehen weitere Funktionen zur Verfügung, die Sie vielleicht verwenden möchten. Eine Polling-Komponente wird periodisch mit festgelegten Intervallen aufgerufen, zum Beispiel in unserem Konstruktor (siehe [6] für weitere Details).

### Hinzufügen des I<sup>2</sup>C-Busses

Jetzt müssen wir nur noch den I<sup>2</sup>C-Bus des Sensors in der Software verbinden, um neben der physikalischen auch eine logische Verbindung herzustellen. Die Adafruit-Bibliothek verwendet zu diesem

## Install espiic.yaml

```

Library Manager: Installing esphome/AsyncTCP-esphome @ 1.2.2
Library Manager: AsyncTCP-esphome @ 1.2.2 has been installed!
Library Manager: Installing Adafruit_VEM_L7700 Library
Library Manager: Adafruit_VEM_L7700 Library @ 1.1.1 has been installed!
Library Manager: Installing dependencies...
Library Manager: Installing Adafruit_BusIO
Library Manager: Warning! More than one package has been found by Adafruit_BusIO requirements:
- adafruit/Adafruit_BusIO @ 1.9.1
- mertmechanic/Adafruit_BusIO @ 1.7.3
Library Manager: Please specify detailed REQUIREMENTS using package owner and version (showed above) to avoid name
conflicts
Library Manager: Adafruit_BusIO @ 1.9.1 has been installed!
Library Manager: Installing Hash
Library Manager: Already installed, built-in library
Dependency Graph
|-- <AsyncTCP-esphome> 1.2.2
|-- <Adafruit_VEM_L7700 Library> 1.1.1
| |-- <Adafruit_BusIO> 1.9.1
| | |-- <Wire> 1.0.1
| | |-- <SPI> 1.0
| |-- <Wire> 1.0.1
| |-- <SPI> 1.0
|-- <ESP8265> 1.0
| |-- <WiFi> 1.0

```

Bild 2. ESPHome installiert die Adafruit-Bibliothek automatisch.

Zweck die Wire-Bibliothek von Arduino. In der ESPHome-YAML-Datei fügen wir daher eine `i2c`-Sektion hinzu, damit ESPHome weiß, dass der I<sup>2</sup>C-Bus verwendet wird. Der ESP32 verfügt über zwei I<sup>2</sup>C-Busse mit SDA und SCL, die an fast jeden Pin des Chips angeschlossen werden können. ESPHome unterstützt daher mehrere I<sup>2</sup>C-Busse mit frei zuweisbaren Pins, so dass man nur noch angeben muss, wohin was gehört. Allerdings unterstützt die Arduino-Wire Bibliothek nur einen I<sup>2</sup>C-Bus. Wie kann man ihr mitteilen, dass sie den von Ihnen gewünschten Bus verwenden soll? Gar nicht! Die Bibliothek will immer den Standard-Bus verwenden. In ESPHome ist dieser Default-I<sup>2</sup>C-Bus der erste Bus, der in der `i2c`-Sektion angegeben ist. Es wäre schön gewesen, wenn man einen I<sup>2</sup>C-Bus über seine ID angeben könnte, und ESPHome wäre wahrscheinlich mehr als glücklich, wenn man das tun könnte, aber die zugrundeliegende Arduino-Wire Bibliothek erlaubt das nicht.

**Fertig!**

Hier endet dieser Artikel. Wenn Sie den Automator von ESPHome mit dem oben beschriebenen Code programmieren, erhalten Sie ein Gerät, das von einer Hausautomatisierungssteuerung wie Home Assistant

gesteuert werden kann. In der YAML-Datei von ESPHome sind keine Automatisierungsregeln enthalten, so dass das Gerät ohne eine Steuerung nur die Lichtintensität der Umgebung misst. Automatisierungsregeln können der YAML-Datei selbst hinzugefügt oder im Home Assistant erstellt werden. Eine Beschreibung des restlichen, nicht besonders aufregenden YAML-Codes finden Sie in [3] und [4]. Die YAML-Konfigurationsdatei und der C++-Code stehen auf der Webseite dieses Artikels zum Download bereit [7]. ◀

210190-02

**Ein Beitrag von**

Gestaltung, Text und Fotografien:

**Clemens Valens**

Redaktion: Jens Nickel, C. J. Abate

Übersetzung: Rolf Gerstendorf

Layout: Harmen Heida

**Haben Sie Fragen oder Kommentare?**

Haben Sie technische Fragen oder Kommentare zu diesem Artikel?

Schicken Sie eine E-Mail an den Autor unter [clemens.valens@elektor.com](mailto:clemens.valens@elektor.com) oder kontaktieren Sie Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).**WEBLINKS**

- [1] Y. Bourdon, „ESP32-verbundenes Thermostat“, ElektorMag 9/2021: [www.elektormagazine.de/magazine/elektor-182/59862](http://www.elektormagazine.de/magazine/elektor-182/59862)
- [2] Elektor Automator: [www.elektormagazine.de/labs/automator](http://www.elektormagazine.de/labs/automator)
- [3] ESPHome starts here: [www.elektormagazine.de/labs/how-to-home-assistant-esphome](http://www.elektormagazine.de/labs/how-to-home-assistant-esphome)
- [4] C. Valens, „Hausautomation leicht gemacht“, ElektorLab 9/2020: [www.elektormagazine.de/magazine/elektor-154/58936](http://www.elektormagazine.de/magazine/elektor-154/58936)
- [5] Bibliothek Adafruit VEM\_L7700: [https://github.com/adafruit/Adafruit\\_VEM\\_L7700](https://github.com/adafruit/Adafruit_VEM_L7700)
- [6] ESPHome auf GitHub: <https://github.com/esphome>
- [7] Downloads zu diesem Artikel: [www.elektormagazine.de/210190-02](http://www.elektormagazine.de/210190-02)

**Das Beste  
Preis-Leistungs-Verhältnis  
in der Messtechnik**

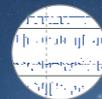
**SHS800X/SHS1000X  
Tragbares Oszilloskop**

**Fünf-in-Eins**

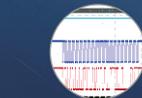
Oszilloskop



Data Logger



Multimeter

Serieller Bustrigger  
und Dekodung

Spektrum Analyse

Isolationsspannung (SHS1000X)  
CAT III 600 Vrms CAT II 1000 Vrms

**SIGLENT®**

[www.siglenteu.com](http://www.siglenteu.com)  
[Info-eu@siglent.com](mailto:Info-eu@siglent.com)

# Herausforderungen bei der Markteinführung von IoT-Lösungen

## Sorgen um Sicherheit, Skalierbarkeit und Wettbewerb



Bild 1. Die Europäische Kommission hat untersucht, ob die großen Akteure im Bereich der Sprachassistenten, der beliebtesten Benutzerschnittstelle für das intelligente Zuhause, den Wettbewerb behindern. (Quelle: Shutterstock/Gorodenkoff)

Von Stuart Cording (Elektor)

Das Internet der Dinge (IoT) gibt es seit mehr als 20 Jahren, und eine Vielzahl von drahtlosen Technologien ist entstanden, um seine Einführung zu unterstützen. Im privaten Bereich haben sich Sprachassistenten als wichtigste Benutzerschnittstelle für eine Reihe intelligenter Geräte etabliert. Trotz dieser Fortschritte schaffen es mehr als ein Drittel der IoT-Projekte nie über die Phase des Proof-of-Concept hinaus. Außerdem ist die Europäische Kommission besorgt, dass ein Mangel an Wettbewerb in einigen Anwendungsbereichen den Markteintritt von EU-Unternehmen behindern könnte. Was sind also die wirklichen Herausforderungen, und wie sieht es mit der Einführung einer IoT-Lösung in Europa aus?

Wenn Sie ein Gefühl für die Technologien bekommen wollen, die hinter dem Internet der Dinge (IoT) stehen, finden Sie schnell eine ganze Reihe von geeigneten Projekten. Suchen Sie einfach nach „IoT“ und der von Ihnen bevorzugten Prototyping-Plattform, und Sie werden mit Projekten, Cloud-Plattformen, Technologievergleichen und Listen von Ideen überschüttet. Auch Elektor ist eine hervorragende Informationsquelle: Seit der Begriff IoT vor mehr als 20 Jahren geprägt wurde, hat unsere Website mehr als 600 Artikel zum Thema veröffentlicht [1].



Zwischen einer Prototyp-Plattform, die das grundlegende Konzept einer IoT-Lösung veranschaulicht, und der tatsächlichen Realisation einer solchen Lösung klafft jedoch eine beträchtliche Lücke. Der Bericht *IoT Insights* von Microsoft [2] befragte im Jahr 2021 mehr als 3.000 IoT-Experten. Sie fanden heraus, dass 35 % der IoT-Projekte in der Test- oder der Proof-of-Concept-Phase scheitern, 5 % mehr als bei der gleichen Umfrage ein Jahr zuvor. Als Grund für das Scheitern wurden meist die hohen Kosten für die Skalierung genannt. Weitere Gründe sind die Unmengen zu testender Plattformen, zu prüfender Anwendungsfälle und auch ein Mangel an Ressourcen. Eine andere Studie von Cisco [3] berichtet, dass nur 26 % der befragten Unternehmen ihre IoT-Initiativen für erfolgreich hielten. Die Antworten in dieser Studie zeigten, dass die meisten IoT-Projekte zwar auf dem Papier gut aussehen, sich aber letztendlich als komplexer erweisen als ursprünglich angenommen. Trotz dieser düsteren Rückmeldungen zum IoT insgesamt gibt es Bereiche, in denen das IoT große Fortschritte macht und wachsende Umsätze bringt.

### EU prüft das IoT für Verbraucher

Viele EU-Bürger haben in den letzten Jahren das Angebot an IoT-Lösungen für Verbraucher begrüßt, und zwar so sehr, dass ein Bericht von Statista [4] über die Umsätze im Bereich *Smart Home* vorhersagt, dass sie sich von rund 17 Milliarden Euro im Jahr 2020 auf rund 38,1 Milliarden Euro im Jahr 2025 mehr als verdoppeln werden. Aus Sorge um den freien Wettbewerb in diesem Bereich hat die Europäische Kommission im Rahmen ihrer Digitalen Strategie eine Untersuchung durchgeführt [5]. Der Bericht dazu, der im Januar 2022 veröffentlicht wurde, enthält Beiträge von Herstellern portabler Geräte, von vernetzten Verbrauchergeräten, die im intelligenten Heim verwendet werden, und von Anbietern von Diensten über diese intelligente Geräte.

Außerdem bat die Kommission um Beiträge von Organisationen, die Normen und Standards festlegen. Es fällt auf, dass ein Großteil der Beiträge Sprachassistenten (Voice Assistants) als Benutzerschnittstelle für IoT-Produkte und -Dienste gewidmet sind (**Bild 1**). In der Analyse der Smart-Home-Landschaft macht der Bericht [6] deutlich, dass Googles *Google Assistant*, Amazons *Alexa* und Apples *Siri* die führenden Allzweck-Sprachassistenten in Europa sind. Es gibt weitere Sprachassistenten, die jedoch in ihrer Funktionalität eher beschränkt sind und sich auf die Unterstützung eines einzelnen Produkts oder einer App eines Dienstleisters konzentrieren. Laut ZDNet bietet unter den drei großen Anbietern Amazons *Alexa* den höchsten Grad an Kompatibilität und unterstützt rund 7.400 Marken [7]. Im Vergleich dazu unterstützt Google etwa 1.000 Marken, während Apple mit etwa 50 Marken die größte „Exklusivität“ aufweist.

### Wenig Chancen für Neueinsteiger

Bei so mächtigen und auf dem Markt etablierten Akteuren bleibt wenig Platz für Neueinsteiger, und der technologische Aufwand zur Entwicklung eines wettbewerbsfähigen Sprachassistenten

ist beträchtlich. Wenn Sie also eine Sprachsteuerung für Ihre IoT-Lösung nutzen wollen, müssen Sie nach der Musik der großen Drei tanzen. Ein alternativer Ansatz wäre die Lizenzierung eines Sprachassistenten, doch einige befragte Hersteller berichteten, dass die Lizenzbedingungen ihre Wahlmöglichkeiten stark einschränken. Diese reichten von Exklusivität oder Beschränkungen, die den gleichzeitigen Einsatz von mehr als einem Sprachassistenten verhinderten, bis hin zu einer Lizenzierung, die die Einbeziehung anderer Arten von Software oder Anwendungen erzwang, was bedeutete, dass die Sprachassistent-Technologie nicht eigenständig verwendet werden konnte.

Ein weiteres großes Problem ist der Zugang zu den Daten. Als „Third Party“, die auf einen Sprachassistenten aufbaut, haben Sie nur begrenzten Zugriff auf die erfassten Daten. Der Anbieter des Sprachassistenten selbst hat aber Zugang zu den Audioaufzeichnungen und weiß auch, wie viele fehlgeschlagene Versuche es gab, die für Ihr Gerät ausgewählten Befehle zu erteilen. Sie und

Ihr Team haben jedoch keinen Zugang zu den Audioaufzeichnungen, so dass Sie eher auf das Feedback der Endanwender warten müssen, um festzustellen, dass die von Ihnen gewählten Sprachbefehle in der rauen Praxis nicht so funktionieren wie in einem kleineren Test. Da der Sprachassistent-Anbieter außerdem alles analysieren kann, was in das Gerät gesprochen wird, könnte er diese Daten nutzen, um eine Lösung zu entwickeln, die mit der Ihren konkurriert, oder um die von Ihrer IoT-Lösung gelieferte Benutzererfahrung zur Verbesserung seiner Dienste zu nutzen.

Ein weiteres Problem ergibt sich, wenn der Sprachassistent-Anbieter auch Werbedienste anbietet. Theoretisch könnten ihm die von Ihren Nutzern bereitgestellten Spracheingaben dabei helfen, die Werbung genauer auf

die durch Ihren Kundenstamm repräsentierte gesellschaftliche Gruppe auszurichten.

Und schließlich geht auch der Wiedererkennungswert und die Erfahrung mit der Marke verloren. Ihre sorgfältig ausgearbeitete Lösung ist den Launen und der Gnade des Sprachassistenten ausgeliefert. Sollte sich der Anbieter dazu entschließen, eine wesentliche Änderung vorzunehmen, zum Beispiel die verwendete Stimme, das Weckwort oder sogar die Einführung von Funktionen zu ändern, die zu einem Rückgang der Nutzerzahlen führen, werden Sie unweigerlich auch diese Folgen zu spüren bekommen. In dem Bericht werden auch viele andere relevante Bereiche untersucht, darunter Anwendungsprogrammierschnittstellen (API), Standards, Interoperabilität, das Ungleichgewicht der „Macht“ von Drittentwicklern von IoT-Geräten und den großen Anbietern von Cloud-Plattformen sowie Vertragskündigungsklauseln.

Der Bericht enthält zwar keine Empfehlungen, doch in den Schlussfolgerungen heißt es, dass der Inhalt des Berichts zur Standardisierungsstrategie der Europäischen Kommission beiträgt und in die legislative Debatte über das Gesetz über digitale Märkte (DMA) einfließen wird.



*Die IoT-Landschaft zeigt zahlreiche Geschäftsmöglichkeiten, unabhängig davon, ob es Lösungen für Verbraucher oder die Industrie sind.*

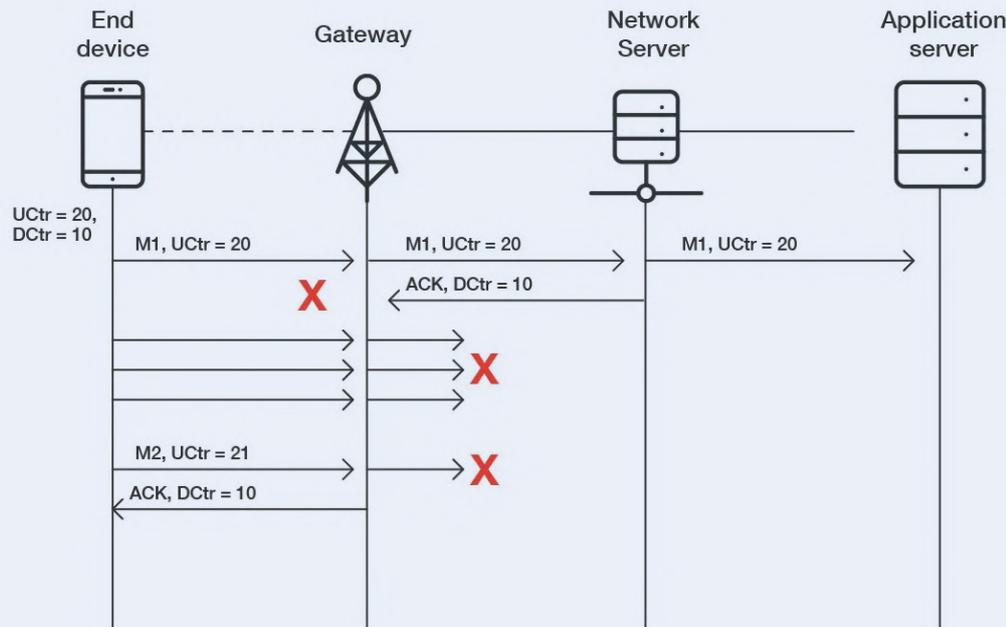


Bild 2. Forscher entdeckten einen Denial-of-Service-Angriff (DoS) in LoRa 1.0. Durch das erneute „Abspielen“ einer vorherigen erfolgreichen Datenübertragung wird ein LoRaWAN-Knoten daran gehindert, weitere Datenpakete zu senden. (Quelle: Trend Micro)

## Sicherheitsprobleme beunruhigen IoT-Implementierer

Bei den Daten im IoT-Insights-Bericht von Microsoft steht die IoT-Sicherheit ganz oben auf der Kummerkasten-Liste. 29 % der Befragten gaben an, dass die damit verbundenen Sicherheitsrisiken sie davon abhalten, das IoT verstärkt einzusetzen. Der Bericht zeigt auch, dass etwa ein Drittel der Unternehmen über die Sicherheitsrisiken des IoT besorgt ist, insbesondere über Verletzungen des Datenschutzes. Um dem entgegenzuwirken, wird Outsourcing als beste Möglichkeit beschrieben, den „Seelenfrieden“ zu verbessern. Zwar kennen viele Ingenieure den Satz „Sicherheit durch Unklarheit“ (beziehungsweise das Gegenteil), doch nur wenige sind in diesem Bereich erfahren genug, um eine durchgängige Lösung für den Schutz gegen Angreifer zu entwickeln. Und obwohl die Halbleiterhersteller eine Reihe von Single-Chip-Sicherheitslösungen anbieten, müssen die Entwickler lernen, diese richtig einzusetzen, um nicht versehentlich neue Sicherheitslücken zu eröffnen. In den letzten Jahrzehnten haben sich Low-Power-Wide-Area-Networks (LPWAN)-Lösungen wie LoRa und Sigfox als wichtige drahtlose IoT-Technologien etabliert, die eine Kommunikation mit großer Reichweite ermöglichen. Sie erreichen Reichweiten von mehreren zehn Kilometern und sind dank ihrer überlegenen Low-Power-Leistung bei geringen Datenmengen eine Alternative zu Mobilfunknetzen wie LTE Cat-M1 und NB-IoT [8]. Aber wie sicher sind sie?

## LoRaWAN auf dem Prüfstand

LoRa und LoRaWAN wurden von der Sicherheits- und Hackergemeinde besonders unter die Lupe genommen. Sébastien Dudek von Trend Micro, einem Unternehmen für IT-Sicherheitslösungen, ist einer von mehreren Forschern, die ausführlich über potenzielle Probleme veröffentlicht haben. In einer Reihe von drei technischen Kurzberichten [9][10][11] skizziert er Probleme bei der Implementierung und möglichen Angriffen, angefangen von Denial-of-Service- (DoS) (Bild 2) und Lauschangriffen, Bitflipping (Bild 3) bis

zu Spoofing von Bestätigungen (ACK) (Bild 4). Die Folgen solcher Angriffe reichen von der Unfähigkeit, mit einem Knoten zu kommunizieren, über die gewollte Verkürzung der Batterielebensdauer bis hin zur Veränderung von Anwendungsdaten.

Viele der aufgezeigten Schwachstellen wurden zwischen Version 1.0.2 und 1.1 des LoRaWAN-Standards behoben. Weitere Herausforderungen ergeben sich jedoch, wenn LoRaWAN-Knoten an Gateways betrieben werden, die unterschiedliche Versionen der Spezifikation verwenden. In solchen Fällen müssen Änderungen vorgenommen werden, um eine gesicherte Abwärtskompatibilität zwischen Endgeräten und Backend zu gewährleisten, wie in einem Papier von Tahsin C. M. Dönmez aus dem Jahr 2018 hervorgehoben wird [12].

Neben dem Hacken von drahtlosen Verbindungen gibt es auch das Problem, dass bösartige Akteure die Hardware stehlen und direkt angreifen können. Sébastien Dudek untersucht auch diesen Sicherheitsaspekt. Im Falle von LoRaWAN verwenden viele Entwickler einen Mikrocontroller (MCU) und ein drahtloses Modul von Semtech, die über SPI verbunden sind. Daten, die zwischen den beiden ausgetauscht werden, können leicht erfasst und analysiert werden.

Darüber hinaus stellt sich auch die Frage nach der Sicherheit der MCU selbst. Bei einer Angriffsmethode wird einfach die Firmware aus dem Flash-Speicher extrahiert, so dass der Code analysiert werden kann. Wenn sich die Sicherheitsschlüssel auch in der Firmware befinden, kann ein Angreifer diese verwenden.

Darüber hinaus stellt sich auch die Frage nach der Sicherheit der MCU selbst. Ein Angreifer kann einfach die Firmware aus dem Flash-Speicher auslesen und den Code analysieren. Wenn sich Sicherheitsschlüssel in der Firmware befinden, kann ein Angreifer sie verwenden, um Knoten zu entwickeln, die authentische Endgeräte vortäuschen. Als Gegenmaßnahme wird die Verwendung von Secure Elements (SE) empfohlen, Einzelchip-Authentifizierungsbausteine, die Verschlüsselungsschlüssel sicher speichern.

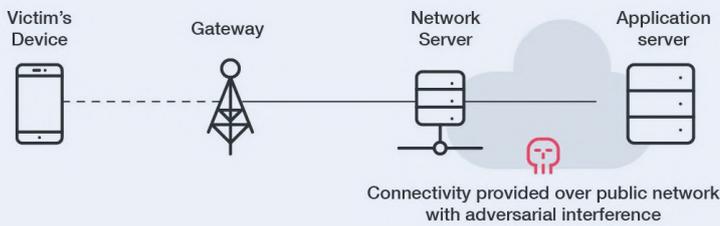


Bild 3. Aufgrund der bekannten festen Struktur von LoRaWAN-Datenpaketen ist es möglich, den Inhalt von Bits umzudrehen (Bit-Flipping-Angriff), ohne die Nachricht entschlüsseln zu müssen. Dazu ist der Zugriff auf den Netzwerkservers erforderlich, der die Daten empfängt. (Quelle: Trend Micro)

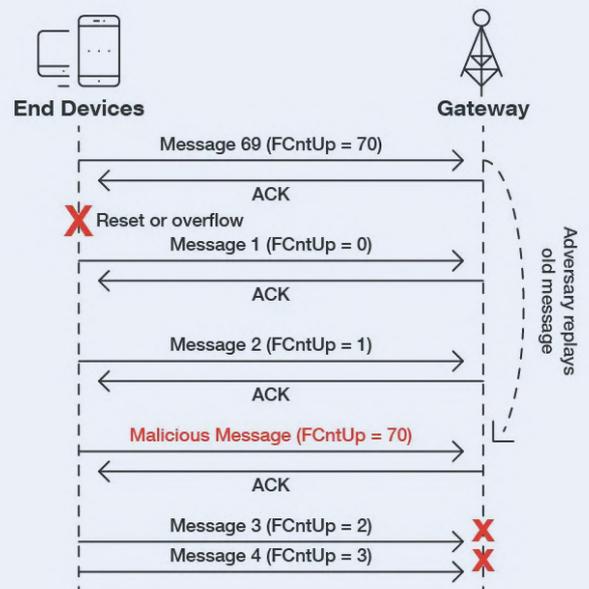


Bild 4. Das Stören der drahtlosen Verbindung führt dazu, dass der LoRaWAN-Knoten die Paketübertragung bis zu sieben Mal wiederholt, was die Batterielebensdauer verkürzt. Wenn der Angreifer auch Acknowledgment-Pakete (ACK) abfängt und wiedergibt, denkt der Knoten, dass die Verbindung noch funktioniert, da ACK-Pakete die Nachricht, die sie bestätigen, nicht deklarieren. (Quelle: Trend Micro)

Ein Ansatz mit dem ATECC608A von Microchip [13] ist einer von mehreren, für die Beispielcode verfügbar ist. Während aber die Beispielprojekte zeigen, wie die kryptografischen Schlüssel geschützt werden können, wird die sichere Boot-Funktion dieses Authentifizierungsbausteins jedoch nicht behandelt. Würde aber dieser Ansatz für ein Produkt verwendet, könnte der Authentifizierungsbaustein entfernt und als SE mit einer anderen MCU und neuer Firmware verwendet werden.

### Sicherheitsprobleme, wohin man schaut

LoRaWAN-Endgeräte bieten nur eine begrenzte Datenbandbreite und sind nicht adressierbar, da sie keine IP-Adresse wie ein WLAN-Modul haben. Daher stellen sie ein minimales Risiko für Unternehmensnetzwerke dar. Potenzielle Risiken bestehen jedoch bei Anwendungen, die auf solchen drahtlosen Technologien basieren. Diese können Auswirkungen auf Leben und Umwelt haben, wenn etwas schief geht. Beispielsweise können LoRa-basierte Sensoren als Teil einer intelligenten Überwachung des Wasserstands zuständig sein, um Überschwemmungen zu vermeiden. Wenn die Daten der Sensoren blockiert werden, können die Hochwasserschutzsysteme möglicherweise nicht reagieren. Umgekehrt könnten falsch eingespeiste Daten dazu führen, dass der Hochwasserschutz auf ein Ereignis reagiert, das gar nicht stattfindet, mit möglicherweise ebenso katastrophalen Folgen.

LoRa und LoRaWAN sind keine Einzelfälle. Viele Forscher untersuchten auch andere LPWAN-Technologien wie Sigfox und NB-IoT. In einem Papier von Florian Laurentiu Coman et al. [14] werden mehrere Proof-of-Concept-Angriffe auf andere drahtlose Netzwerke beschrieben. In einem technischen Brief der Deutschen Telekom [15] heißt es, dass die Implementierung von Sigfox und LoRaWAN „ohne Sicherheitselement selbst eine Ende-zu-Ende-Verschlüsselung wirkungslos sein kann.“ Im Gegensatz dazu profitiert NB-IoT von den seit langem bewährten LTE-Sicherheitsmerkmalen wie Authentifizierung und sichere Schlüsselgenerierung



Bild 5. Feinstaubsensoren von DEUS POLLUTRACK sammeln Daten von stationären und mobilen IoT-Geräten. Die Daten werden über 4G- und 5G-Mobilfunknetze an das Backend übermittelt. (Quelle: DEUS POLLUTRACK)

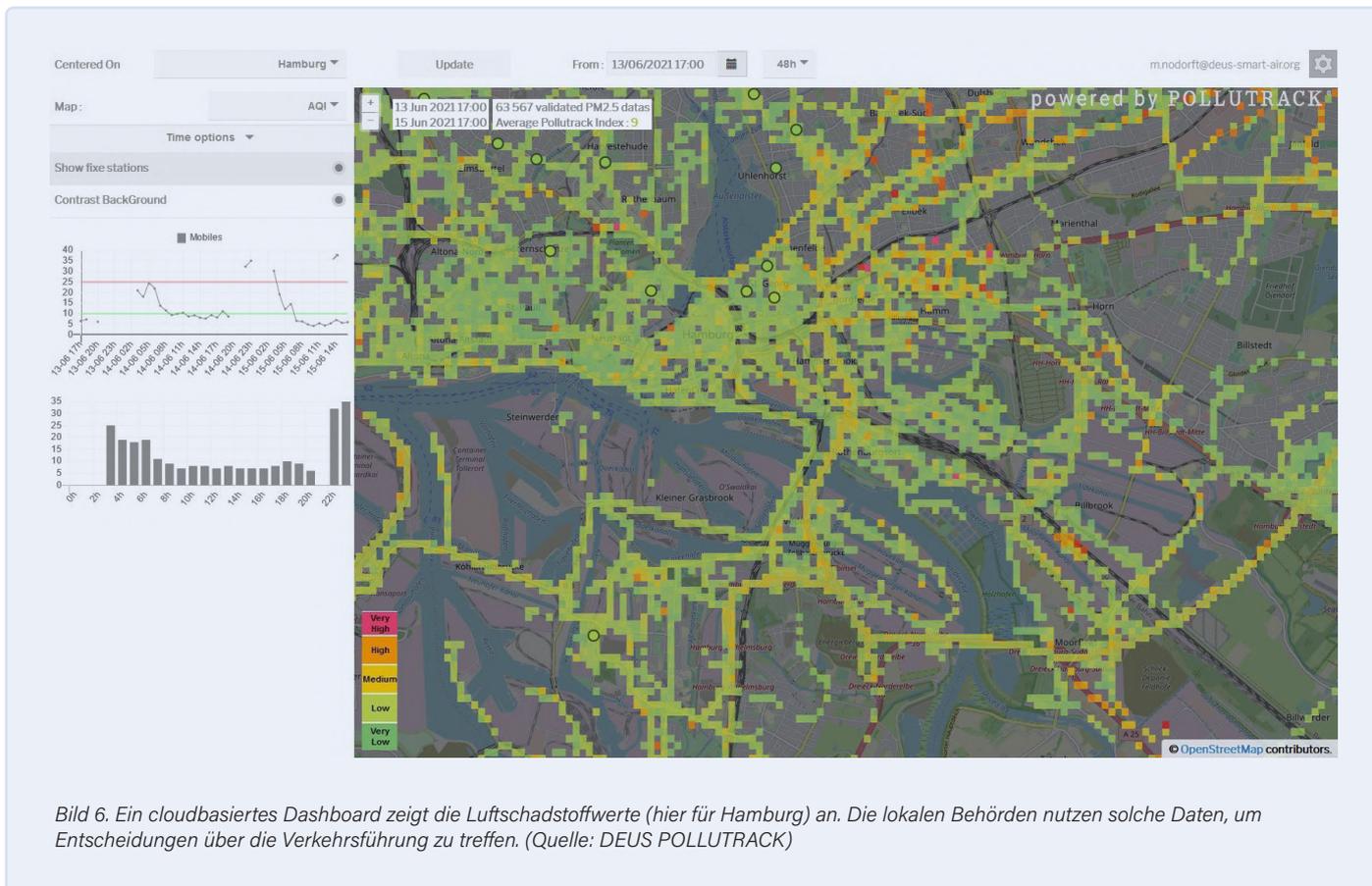


Bild 6. Ein cloudbasiertes Dashboard zeigt die Luftschadstoffwerte (hier für Hamburg) an. Die lokalen Behörden nutzen solche Daten, um Entscheidungen über die Verkehrsführung zu treffen. (Quelle: DEUS POLLUTRACK)

und -austausch. Es wird jedoch auch klargestellt, dass eine End-to-end-Verschlüsselung nicht zum Standard gehört und, falls sie als notwendig erachtet wird, mit dem Netzbetreiber besprochen werden muss.

### Bereitstellung von stadtweiten IoT-Lösungen

Bedenken hinsichtlich der Sicherheit in LPWAN-Netzen beeinflussen auch die Wahl der Technologie der DEUS POLLUTRACK Smart City für ihre IoT-Plattform [16]. Das DEUS-Team entwickelt seit mehr als einem Jahrzehnt IoT-Sensornetze zur Überwachung von Feinstaub in Städten. Die Technologie wird in mehr als 15 europäischen Städten eingesetzt und ermöglicht es den Verantwortlichen in den Kommunen, fundierte Entscheidungen zur Luftverschmutzung zu treffen. Die patentierten optischen Partikelzähler (OPC) sind in der Lage, bis hinunter zur Klassifizierung ultrafeiner Partikel (unter  $0,1 \mu\text{m}$ ) zu überwachen. Während größere Partikel wie  $\text{PM}_{10}$  als gefährlich für die Lunge gelten, können ultrafeine Partikel über die eingeatmete Luft in den Blutkreislauf und in andere Organe gelangen.

Die Sensortechnologie von DEUS (Bild 5) kombiniert stationäre und mobile Sensoren, die mit Back-End-Dashboards vernetzt sind, auf denen die erfassten Daten visualisiert werden. Städte wie Marseille und Paris verwenden 40 stationäre Sensoren, die durch 300 mobile Sensoren ergänzt werden [17]. Mobile Sensoren werden an Fahrzeugen von Partnern angebracht, etwa den Lieferwagen von DPD, die ohnehin häufig in der jeweiligen Stadt unterwegs sind. Diese Sensoren kalibrieren sich selbst anhand der bei der Vorbeifahrt an stationären Sensoren erfassten Daten, um die erforderliche Genauigkeit zu gewährleisten. All dies erfordert die

Wahl eines robusten, zuverlässigen und sicheren LPWAN.

Im Gespräch erklärt der DEUS-Mitbegründer Marc Nodorft, dass in den frühen Entwicklungsphasen sowohl Sigfox als auch LoRaWAN in Betracht gezogen wurden. Sigfox bot eine Konnektivitätsinfrastruktur, die die Systembereitstellung vereinfachte, aber keine der beiden LPWANs bot den erforderlichen Datendurchsatz. LoRaWAN war damals nicht sicher genug, und ohne Infrastrukturpartner in den Städten, in denen die Technologie eingesetzt werden sollte, mussten Gateways eingesetzt werden, die über Mobilfunknetze mit dem Backend verbunden waren. Infolgedessen wurden 4G- und später 5G-Mobilfunknetze gewählt, mit denen die Probleme der Abdeckung, Zuverlässigkeit und Sicherheit auf dem erforderlichen Niveau gelöst wurden.

Nodorft berichtete uns auch, dass es zwar viele billige elektronische Lösungen für das Internet der Dinge gibt, diese aber nicht robust genug für einen langfristigen Einsatz in den Umgebungen sind, in denen die DEUS-Geräte installiert werden. Daher entschied man sich für die Entwicklung nach Industriestandards, eine Möglichkeit für alle, die eigene IoT-Produkte planen.

Ein weiterer Aspekt ist der Back-End-Betrieb, der speziell für die Anforderungen der IoT-Implementierung entwickelt wurde (Bild 6). In Zukunft müssen Open-Source-Dashboards für die Berichte unterstützt werden, damit die Behörden, die das System nutzen, und die Bürger auf die Daten zugreifen können, wofür ein Anbieter von Cloud-Diensten erforderlich ist. Und obwohl es eine große Auswahl gibt, wird der Anbieter als ebenso wichtig angesehen wie die technische Lösung. Es wird also nach einem Anbieter gesucht, der persönliche Unterstützung bieten kann und nicht nur einen unpersönlichen Chatbot für den Kundendienst.

Mit so viel Erfahrung in bedeutenden IoT-Implementierungen und viel Wissen über die technischen Herausforderungen fragte ich mich, welche anderen Ratschläge Nodort denjenigen geben könnte, die IoT-Lösungen implementieren wollen. „Wir sind unserer Vision immer treu geblieben“, meint er, „was uns oft dazu gezwungen hat, unseren Ansatz zu ändern.“ Das bedeutete, verschiedene Technologien zu erkunden, mit verschiedenen Partnern zusammenzuarbeiten und die Vertriebsstrategie auf dem Weg zum Erfolg zu ändern.

### Expertenteams und Partnerschaften erforderlich

Betrachtet man die heutige IoT-Landschaft, so wird deutlich, dass es eine Fülle von Geschäftsmöglichkeiten gibt, unabhängig davon, ob man sich auf Lösungen für Verbraucher oder die Industrie konzentriert. Der Weg vom Konzept bis zur Markteinführung ist jedoch mit vielen Herausforderungen verbunden. Entwickler von eingebetteten Systemen mögen sich zwar mit der Entwicklung von Hardware und Firmware auskennen und sogar Erfahrung mit drahtlosen Technologien haben, doch das IoT und die damit verbundenen Herausforderungen in Bezug auf Sicherheit und Skalierbarkeit können eine Organisation rasch überfordern. Laut dem Bericht der Europäischen Kommission haben große Unternehmen auch bei den Geschäftsbeziehungen die Oberhand, wenn es um Dienste und Plattformen geht. Kleine Akteure und Startups dürften es als Einzelkämpfer schwer haben, in diesen asymmetrischen Beziehungen die nötige Unterstützung zu erhalten. Zweifelsohne ist Fachwissen, auch angemietet oder geliehen, unerlässlich, um über Beispielanwendungen, Demonstrations-Dashboards und Test-IoT-Dienste hinauszukommen. Und

schließlich ist es von entscheidender Bedeutung, dass Sie an Ihrer Vision festhalten und gleichzeitig in allen Bereichen der Umsetzung flexibel bleiben - von der Wahl der Technologie bis zum Zielmarkt, um sie zu verwirklichen. ◀

220053-02

### Ein Beitrag von

Text: **Stuart Cording**Redaktion: **Jens Nickel, C. J. Abate**Übersetzung: **Rolf Gerstendorf**Layout: **Harmen Heida**

### Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an den Autor unter [stuart.cording@elektor.com](mailto:stuart.cording@elektor.com) oder kontaktieren Sie Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).

## WEBLINKS

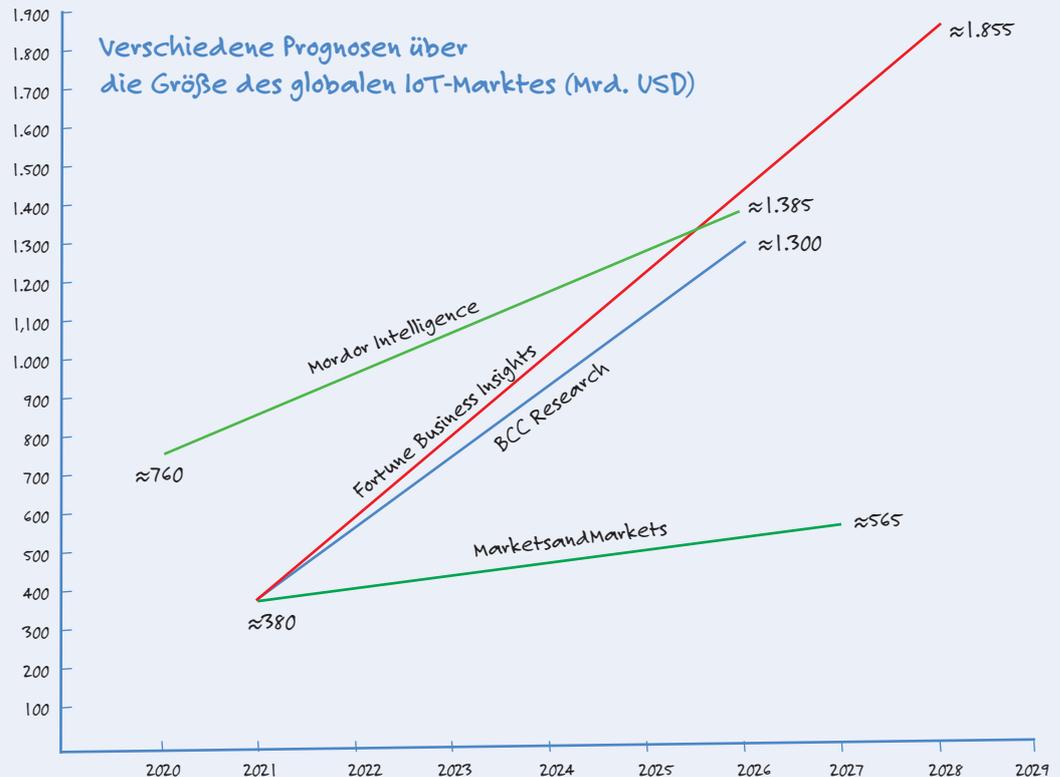
- [1] Elektor IoT-Artikel: [www.elektormagazine.de/select/internet-of-things](http://www.elektormagazine.de/select/internet-of-things)
- [2] „IoT Insights, Edition 3“, Microsoft/Hypothesis, Oktober 2021: <https://bit.ly/3rxMk3a>
- [3] „The Journey to IoT Value“, Cisco, Mai 2017 (Bezahlschranke): <https://bit.ly/3GzdJWS>
- [4] Dr. J. Lasquety-Reyes, „Smart Home - revenue forecast in Europe from 2017 to 2025“, Statista, Juni 2021 (Bezahlschranke): <https://bit.ly/3LIGiuG>
- [5] „Sector inquiry into the Consumer Internet of Things“, Europäische Kommission, Januar 2022: <https://bit.ly/3Lgw9iE>
- [6] „Final report - sector inquiry into consumer Internet of Things“, Europäische Kommission, Januar 2022: <https://bit.ly/3B2Htu9>
- [7] „The best voice assistant“, ZDNet, September 2021: <https://zd.net/3rxf6Rt>
- [8] L. Tan, „Comparison of LoRa and NB-IoT in Terms of Power Consumption“, KTH Royal Institute of Technology, Januar 2020: <https://bit.ly/3JafsUb>
- [9] S. Dudek, „Low Powered and High Risk: Possible Attacks on LoRaWAN Devices“, Trend Micro, Januar 2021: <https://bit.ly/3rA02Tg>
- [10] S. Dudek, „Gauging LoRaWAN Communication Security with LoraPWN“, Trend Micro, Februar 2021: <https://bit.ly/3LhV0T5>
- [11] S. Dudek, „Protecting LoRaWAN Hardware from Attacks in the Wild“, Trend Micro, März 2021: <https://bit.ly/3rxquge>
- [12] T.C.M. Dönmez, „Security of LoRaWAN v1.1 in Backward Compatibility Scenarios“, Elsevier 2018: <https://bit.ly/3GtzKq0>
- [13] Microchip-Produktseite - ATECC608A: <https://bit.ly/3B7zImS>
- [14] F.L. Coman et al. „Security issues in internet of things: Vulnerability analysis of LoRaWAN, sigfox and NB-IoT“, IEEE, Juni 2019: <https://bit.ly/3uwuUQX>
- [15] ] „NB-IoT, LoRaWAN, Sigfox: Ein aktueller Vergleich“, Deutsche Telekom AG, April 2021: <https://bit.ly/3uyUyjd>
- [16] Website DEUS-POLLUTRACK: <https://bit.ly/3sHL9O5>
- [17] DEUS-Sensornetze: <https://bit.ly/3Gzjbcc>

## IoT: Ist es der neue Heilige Gral?

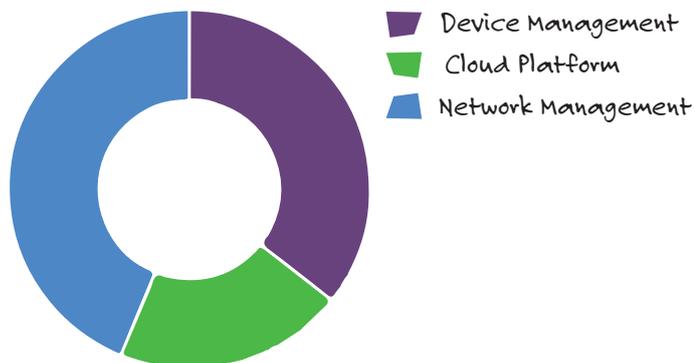


Was ist IoT? Wir alle wissen, wofür IoT steht. Die Frage sollte eher lauten: Was soll zum Internet der Dinge zählen? Umfasst der IoT-Markt auch *Data Warehouse as a Service* (Dwaas)? Umfasst das IoT auch die Cybersicherheit? Oder stehen diese Bereiche für sich allein? Abgesehen von der Debatte, was zum IoT gehört und was nicht, ist es erstaunlich, wie unterschiedlich die Prognosen über die Größe des globalen IoT-Marktes sein können. Die Prognosen (ja, Plural) der Marktforschungsunternehmen sind so unterschiedlich, dass das Glas nicht nur halb voll oder halb leer ist, sondern auch völlig voll, bis es überläuft. Tatsächlich, das riecht nach einem neuen Heiligen Gral!

(Quellen: BCC Research, Fortune Business Insights, MarketsandMarkets, Mordor Intelligence)

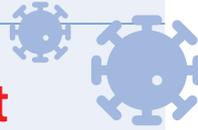


## Drei Zutaten? Nein, es sind vier



Globale IoT-Marktanteile, nach Komponenten, 2020

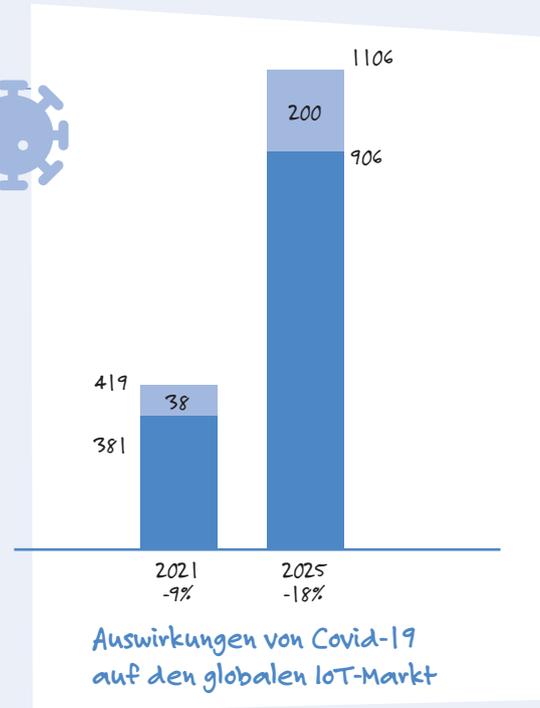
Wenn man an das IoT denkt, denkt man an Geräte und ihre Verbindungen zu einem zentralen Computer. Doch eine solch einfache Sichtweise entspricht nicht mehr der Realität. Neben einem Gerät, einer Verbindung und einem Computer gibt es noch eine weitere Zutat: eine Cloud-Plattform, die durch Rechenzentren ermöglicht wird. Die Geräte senden ihre Informationen an die Cloud, wo sie gespeichert und analysiert werden. Der Computer des Kunden zapft die Cloud an, um die endgültigen Ergebnisse zu erhalten. Dieses *Data Warehouse as a Service* (Dwaas) oder *Platform as a Service* (PaaS) macht schon etwa 20 % des weltweiten IoT-Umsatzes aus. Es wird erwartet, dass Cloud-Plattform-Dienste zwischen 2021 und 2026 um rund 22 % wachsen werden. (Quelle: BCC Publishing, Fortune Business Insights)



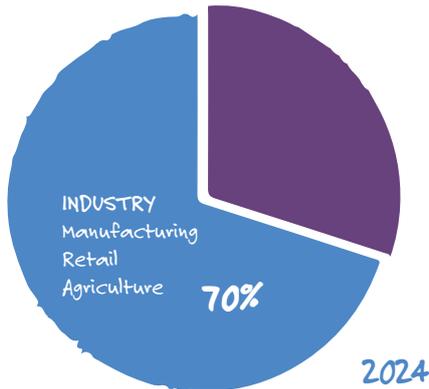
# Long Covid: Negativer Effekt auf den IoT-Markt hält an

Vodafone will zwar den IoT-Unternehmen, die ein rosiges Bild von ihren Markterwartungen zeichnen, nicht den Mut nehmen, doch hat das Telekommunikationsunternehmen die Auswirkungen von Covid-19 auf den globalen IoT-Markt untersucht. Das multinationale Unternehmen kam zu dem Schluss, dass die Auswirkungen von Covid-19 noch eine ganze Weile zu spüren sein werden. Während der weltweite IoT-Markt im vergangenen Jahr 9 % seines Umsatzes durch die Pandemie einbüßte, wird sich dieser Prozentsatz bis 2025 auf 18 % verdoppeln. Vodafone ist sich zwar des Schattens bewusst, den Covid-19 in den kommenden Jahren werfen wird, doch der Bericht über den IoT-Markt macht ebenso deutlich, dass nach dieser Zeit wieder mit besseren Aussichten zu rechnen ist.

(Quellen: Saft Batteries, Vodafone)



Anteil des Industriebereichs an der Gesamtzahl der IoT-Verbindungen



## Wohin führen all diese IoT-Verbindungen?



Welche Bereiche der Gesellschaft werden am meisten vom IoT profitieren? Hätte man diese Frage noch vor ein paar Jahren gestellt, wäre die Antwort gewesen: die Verbraucher. Intelligente Haushaltsgeräte sind zwar immer noch sehr wichtig, wenn man bedenkt, dass im Jahr 2022 der durchschnittliche Haushalt in Nordamerika nicht weniger als neun IoT-Geräte umfasst, von der Haustürkamera bis zur Temperaturregelung. Allerdings werden industrielle Anwendungen (einschließlich Einzelhandel und Landwirtschaft) im Jahr 2024 schon 70 % aller IoT-Verbindungen ausmachen. Die restlichen 30 % verteilen sich auf Unterhaltungselektronik und öffentliche Dienste wie intelligentes Parken, intelligente Abfallentsorgung, intelligente Straßenbeleuchtung und intelligentes Verkehrsmanagement.

(Quellen: Saft Batteries, Global Information, Inc.)

## Warum ist es am IoT so schön?



Eines ist klar: Das IoT wird immer mehr zu einem Teil unseres täglichen Lebens werden. Aber warum? Liegt es daran, dass IoT-Geräte immer billiger werden? Das mag stimmen, wirft aber natürlich die Frage auf, warum die Anbieter auf einen Massenmarkt setzen und ihre Preise so stark senken können. Die Antwort ist im Grunde eine doppelte: (1) das IoT spart seinen Nutzern in erheblichem Umfang Geld und (2) es schafft neue Einnahmequellen, ebenfalls in erheblichem Umfang. Bei den Einsparungen kann man an eine Steigerung der Mitarbeiterproduktivität und der Betriebszeit von Anlagen denken. Neue Einnahmequellen zu schaffen oder bestehende zu erweitern, ist eine weitere Möglichkeit, Geld zu verdienen.

(Quellen: Saft Batteries, Vodafone)

## Die wichtigsten Vorteile von IoT-Implementierungen

50%

Mitarbeiterproduktivität

42%

Anlagen-Betriebszeit (Konsistenz und Zuverlässigkeit)

34%

Das IoT generiert direkt neue Einnahmequellen

34%

Bestehende Einnahmequellen konnten um durchschnittlich 24 % gesteigert werden

# Lieber doch verkabelt

Tipps zur Entwicklung einer 1-Gbit/s-Schnittstelle im Industrieumfeld



Dr. Heinz Zenkner (freier Mitarbeiter bei Würth Elektronik)

Drahtlose Netzwerke erfreuen sich gerade in Industrieanwendungen großer Beliebtheit. Doch in vielen Fällen bleibt eine robuste Verkabelung per Ethernet die zuverlässigere und sicherere Alternative. Wie sich eine 1-Gbit/s-Schnittstelle einfach implementieren lässt, zeigt dieser Beitrag.

Intelligente Sensoren und Zähler, die effiziente Modulations- und Codierungsverfahren mit guten Ausbreitungseigenschaften und geringen Bandbreiten nutzen, gestatten die Einrichtung industrieller, drahtloser Sensornetzwerke. Die meisten der betrachteten Anwendungsfälle sind jedoch auf Anwendungen mit geringem Datendurchsatz beschränkt. Der tatsächliche Durchsatz beträgt in diesen Anwendungsfällen häufig nicht mehr als 1 Mbit/s.

Ein drahtloses Netz hat keine festen Grenzen. So können beispielsweise selbst kleine Anpassungen der Antennenposition des Zugangspunkts die Signalstärke an den anderen Stationen erheblich beeinflussen. Wände, Decken und Böden dämpfen das Signal, und metallische Gegenstände verursachen Reflexionen. Es kann vorkommen, dass eine Station das Signal eines Zugangspunkts empfangen kann, der Zugangspunkt jedoch das Signal der Station nicht. Darüber

hinaus könnte ein Zugriff von außen auf das Netz erfolgen oder die Funksignalübertragung gestört werden.

Daher ist die drahtlose Datenübertragung von Natur aus weniger zuverlässig als über ein kabelgebundenes Netz. So kann es, besonders im industriellen Umfeld zu Situationen kommen, in denen man um ein kabelgebundenes Ethernet-Netzwerk nicht herumkommt.

## Kabelgebundenes Ethernet-Netzwerk

Kabelgebundene Netzwerke ähneln den drahtlosen Netzwerken in dem Sinne, dass sie durch den Austausch von Ethernet-Rahmen zwischen Endpunkten funktionieren. Um Probleme beim Aufbau eines Netzwerks zu vermeiden, sind eine Vielzahl von Regeln zu beachten. Die häufigste Ursache für Netzwerkprobleme sind Regelverletzungen. So darf man im Ethernet z.B.

nicht beliebig lange Kabel verwenden. Bei der Kaskadierung, also der Hintereinanderschaltung von Hubs, dürfen nicht beliebig viele verwendet werden und auch eine ungünstig gewählte Netzwerkstruktur kann zu Fehlern im Netzwerk führen oder aber das Netzwerk unnötig belasten. Aber auch abhängig von der Qualität der Kabel und der Leistungsfähigkeit der Hardware lassen sich oftmals die gewünschten Datenraten nicht erreichen.

Gegenwärtig sind 100Base-TX (Fast Ethernet, 100 Mbit/s), Gigabit-Ethernet (1 Gbit/s), 10-Gigabit-Ethernet (10 Gbit/s) und 100-Gigabit-Ethernet (100 Gbit/s) verfügbar geworden. Für die meisten Zwecke funktioniert Gigabit-Ethernet gut mit einem regulären Ethernet-Kabel, speziell mit den Verkabelungsstandards CAT5e und CAT6. Diese Kabeltypen folgen dem 1000BASE-T-Verdrahtungsstandard, auch IEEE 802.3ab genannt.

Die 1-GB-Ethernet-Schnittstelle arbeitet nach dem Standard 802.3ab-1999 (CL40) und benötigt vier Adernpaare / Kanäle zur Signalübertragung. Somit ergibt sich eine Symbolrate von 125 Megabaud (MBd) mit einer Bandbreite von 62,5 MHz pro Kanal (2 bit pro Symbol). Die Signalspannung bei

1000BASE-T (GB-Ethernet) beträgt durchschnittlich 750 mV differentiell, für die Grenzen gilt  $820 \text{ mV} > U_{\text{Signal}} > 670 \text{ mV}$  bei einer Last von 100 W.

### 1-Gbit/s-Ethernet-Front-End

Ein typisches Front-End für Ethernet ist mit einer RJ45-Schnittstelle ausgestattet. Diese sind für Vollduplex-Übertragungen ausgelegt, d. h. für eine gleichzeitige Übertragung von Send- und Empfangsdaten. Dies ist möglich, weil der Steckverbinder zwei Adernpaare umfasst, wobei immer ein Paar pro Richtung benötigt wird (Differenzspannungsprinzip). Für jeden RJ45-Anschluss wird vom IEEE-Standard eine galvanische Trennung per Übertrager gefordert. Dieser Übertrager schützt die Geräte vor Beschädigung durch Hochspannung auf der Leitung und verhindert Spannungs-Offsets, die durch Potenzialunterschiede zwischen den Geräten auftreten können.

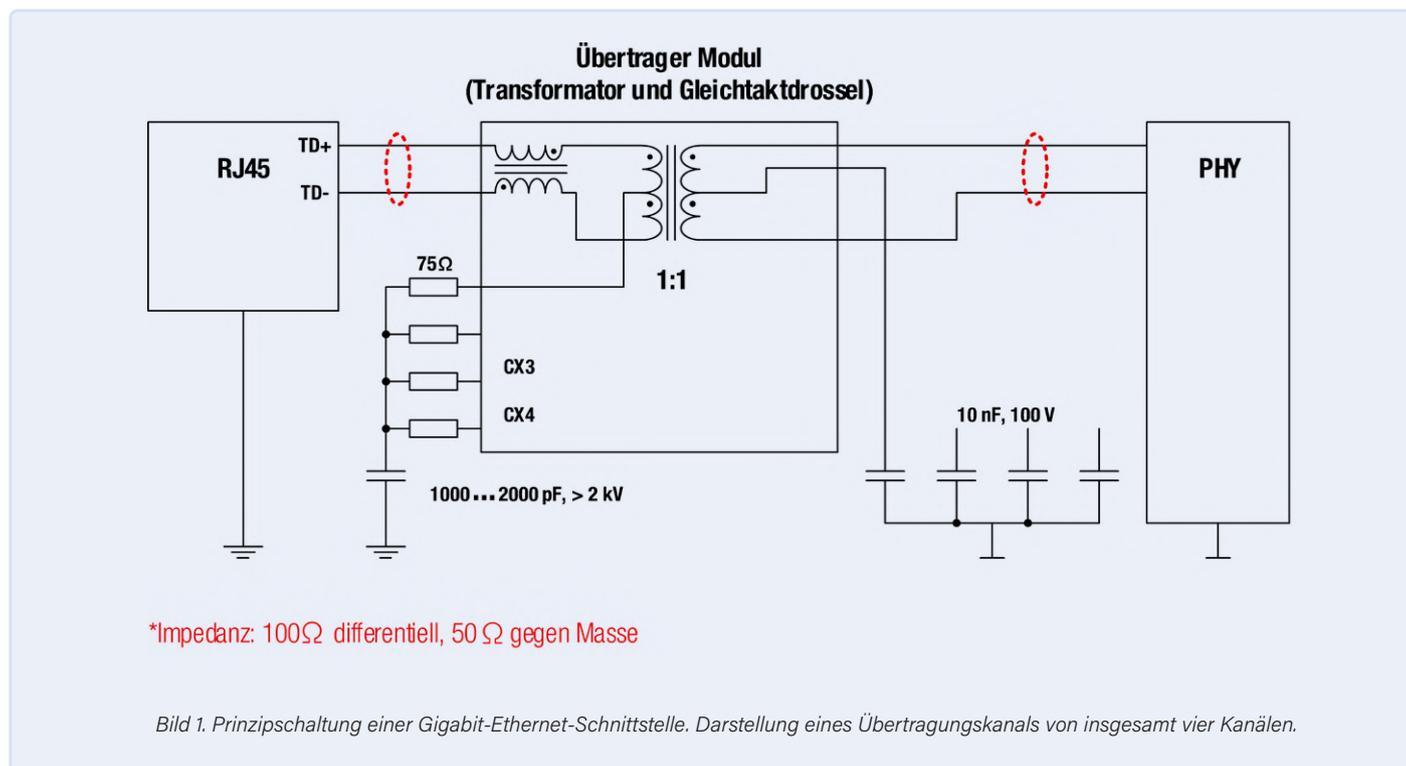
**Bild 1** zeigt die Prinzipschaltung einer Gigabit-Ethernet-Schnittstelle.

### Diskrete Beschaltung der Gigabit-Ethernet-Schnittstelle

Der Ethernet-Transformer (LAN-Übertrager) ist die Schnittstelle zwischen Gerät und

dem Ethernet-Kabel. Der Übertrager sorgt für die sicherheitsrelevante galvanische Trennung zwischen Gerät und Kabel und gleichzeitig für die Impedanzanpassung, einerseits zur internen Logik, andererseits an die symmetrischen Adernpaare. Weiterhin schützt der Übertrager das Gerät vor transienten Störungen, unterdrückt Gleichtaktsignale zwischen dem Transceiver-IC und dem Kabel, sowohl vom Gerät nach außen, als auch vom äußeren Kabel zur Elektronik im Gerät. Jedoch muss das Bauteil auch die Daten bis zu 1 Gbit/s breitbandig übertragen ohne das Send- und das Empfangssignal wesentlich zu dämpfen. Um die Anpassung und die EMV-Anforderungen zu erfüllen sind zusätzliche Komponenten erforderlich.

Eine Beschaltung der Gigabit-Ethernet-Schnittstelle mit diskreten Komponenten zeigt die Schaltung in **Bild 2**. Der LAN-Übertrager sorgt für eine DC-Trennung zwischen der Elektronik und dem Netzkabel. Der mittlere Abgriff der primärseitigen Wicklung zeigt den so genannten „Bob Smith“-Abschluss. Pro Adernpaar wird hier jeweils ein 75-Ω-Widerstand zu einem „Sternpunkt“ zusammengeschaltet, das Ganze wird dann



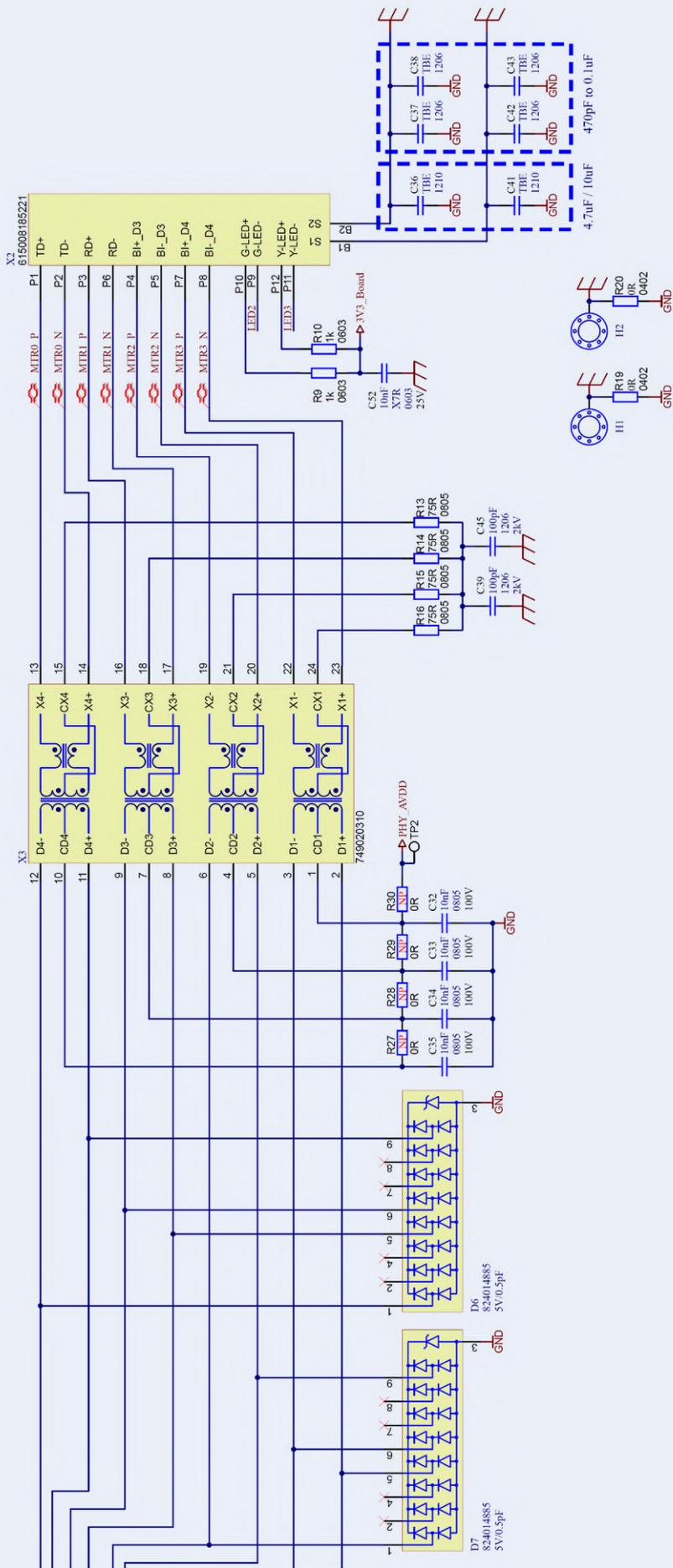


Bild 2. Diskret aufgebaute Schaltung einer Gigabit-Ethernet-Schnittstelle. Das Modul X3 umfasst die LAN-Übertrager sowie Gleichaktrosseln gegen unerwünschte Störungen.

galvanisch getrennt und mittels zwei parallel geschalteten 100-pF/2kV-Kondensatoren an die Gehäusemasse angeschlossen. Die zusätzlich im Modul X3 integrierten Gleichaktrosseln reduzieren Störungen, die über die langen Ethernet-Kabel sowohl kapazitiv, als auch induktiv eingekoppelt werden und so als Gleichaktstörungen die Ethernet-Datenkommunikation beeinträchtigen könnten.

R9, R10 und C52 in Bild 2 dienen der Stromversorgung der typischerweise in der Anschlussbuchse integrierten LEDs. Über die Kondensatoren C36 bis C38 und C41 bis C43 kann die Schirmung der Ethernet-Buchse mit der Platinenmasse (GND) verbunden werden. Bei Blechgehäusen ist es sinnvoll diese Kondensatoren nicht zu bestücken und die Masse (GND) der Elektronik direkt über Verschraubungen an das Gehäuse anzuschließen. Bei Gehäusen aus Kunststoff sollten die Kondensatoren bestückt werden, um den Schirm des Ethernet-Kabels an die Bezugsmasse anzuschließen. Den gleichen Zweck haben die 0-Ω-Widerstände R19 und R20. Hier erfolgt jedoch keine galvanische Trennung, wie sie mit den Kondensatoren realisiert wird. Die alternativen Bestückungen wurden hier zu „experimentellen“ Zwecken vorgesehen, um die Schirmqualität verschiedener Ethernet-Kabel zu vergleichen. Die Kondensatoren C32 bis C35 auf der Sekundärseite der Übertrager verbinden die Mittelabzapfungen dieser HF-technisch mit Masse (GND). Um DC-Ausgleichsströme vom PHY zu vermeiden ist eine galvanische Trennung über Kondensatoren erforderlich. Die Widerstände R27 bis R30 sind aufgrund von Anforderungen mancher PHY-Hersteller vorgesehen (Current Mode Line Driver – Option), werden aber in der Regel nicht benötigt, wenn der PHY im „Standard Voltage Mode“ arbeitet. Unverzichtbar sind hingegen die TVS-Dioden Arrays D6 und D7, die die schnittstellenseitig auftretenden transienten Störungen zum PHY gegen die Schaltungsmasse (GND) begrenzen. Sekundärseitig, d.h. nach den Übertragern des X3-Moduls, treten die transienten Störungen im Gleichtakt auf, deshalb muss an jeden Anschluss der Übertrager jeweils eine TVS-Diode gegen die Bezugsmasse geschaltet werden. Die Störpegel sind auf der Sekundärseite des Übertra-

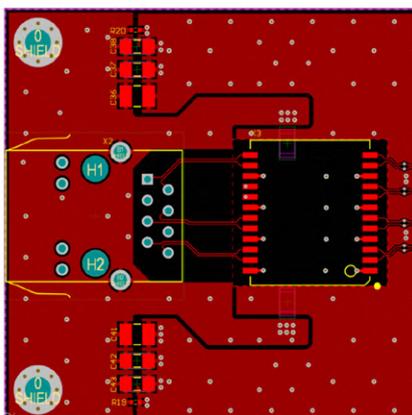
gers aber geringer, als auf der Primärseite. Wichtig für die Funktion der TVS-Dioden ist ein Anschluss der Dioden mit niedriger Impedanz, einerseits in die Signalleitungen eingeschleift und andererseits zur Masse. Das Layout aller vier Lagen der Platine vom Ethernet-Schnittstellenbereich ist in **Bild 3** gezeigt. Die Gehäuse-/Buchsenmasse zum Elektronik-GND ist in allen vier Lagen getrennt. Die Flächen der Gehäusemasse überlappen somit nicht mit anderen Lagen, um die kapazitive Kopplung so gering wie möglich zu halten. Die Masseflächen wurden im Raster ca. alle 4 mm durchkontaktiert. Die von der Ethernet-Buchse kommenden Signalleitungen sind symmetrisch, mit einer differentiellen Impedanz von  $100 \Omega$  gegen die Bezugsmasse geroutet. Die Leiterpärchen weisen eine Leiterbahnbreite von  $0,154 \text{ mm}$  auf und haben einen Abstand von  $0,125 \text{ mm}$  zueinander. Die Ethernet-Buchse ist an der Kante der Leiterplatte positioniert, damit ggf. eine Verbindung mit niedriger Impedanz zu einem Metallgehäuse gewährleistet werden kann.

Das Übertrager-Modul (X3) ist in unmittelbarer Nähe platziert, um die elektrischen Koppeleinflüsse, bzw. Beeinträchtigungen durch lange Leiterbahnen gering zu halten. Wie primärseitig, ist auch auf der Sekundärseite des Übertragermoduls bei den Leiterbahnen eine differentielle Impedanz von  $100 \Omega$  gegen die Bezugsmasse einzuhalten. Die TVS-Arrays müssen unmittelbar in den Signalpfad und gegen GND angeschlossen werden, um einen Spannungsabfall durch parasitäre Induktivitäten zu vermeiden.

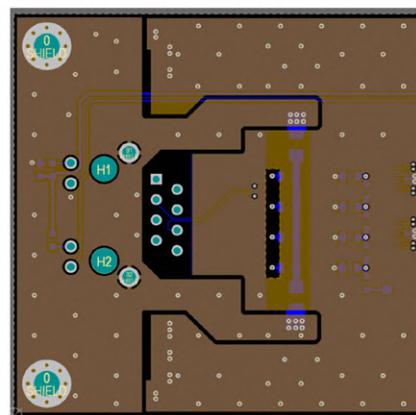
### EMV-Konformität

Das Board hält EMV-technisch die Störfestigkeit nach Industriestandard (EN61000-6-2) und die Funkstöreremissionswerte nach EN55032 Klasse B für Multimediageräte ein. Für ein erfolgreiches Design einer 1-Gbit/s-Ethernet-Schnittstelle sind zahlreiche Punkte zu beachten, diese sind ein HF-gerechtes Schaltungs- und Layout-Design, ein systemabhängiges Massekonzept

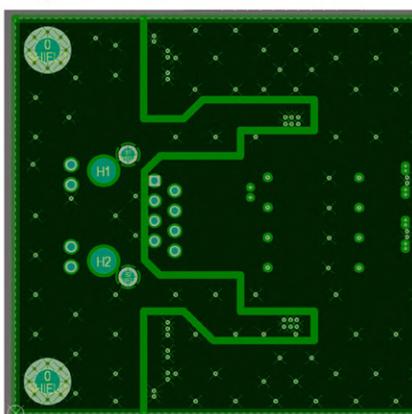
**Layout Top**



**Layout V<sub>CC</sub>**



**Layout GND**



**Layout Bottom**

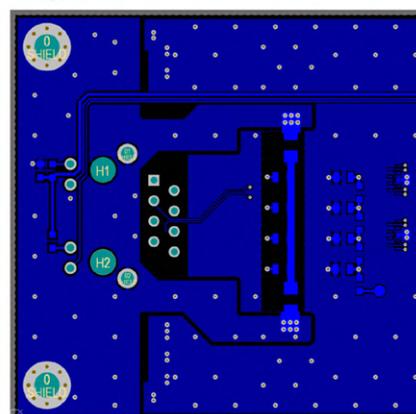


Bild 3. Layout aller vier Platinen-Lagen vom Ethernet-Schnittstellenbereich.

und die richtige Wahl der Bauelemente. Nur wenn alle diese Punkte gemeinsam beachtet werden, kann ein zuverlässig funktionierendes Produkt entwickelt werden, dass den hohen Anforderungen gerecht wird. Weitere Informationen zu

diesen Aspekten und auch für andere Schnittstellenstandards stehen in diversen App-Notes von Würth Elektronik unter [1] zur Verfügung. ◀

220182-02

### Autorenvorstellung



Dr.-Ing. **Heinz Zenkner** ist freier Mitarbeiter bei Würth Elektronik in den Bereichen Technical Marketing und Application Engineering und doziert im Bereich EMV an der technischen Akademie. Gleichzeitig ist Dr. Zenkner öffentlich bestellter und vereidigter Sachverständiger für EMV. Herr Zenkner ist langjähriger Autor verschiedener Fachzeitschriften und Bücher. Zusätzlich hat er an verschiedenen Universitäten, an der IHK und bei zahlreichen Seminaren als Dozent gearbeitet.

### WEBLINK

[1] Würth Elektronik Application Guide: [www.we-online.com/applicationguide/en](http://www.we-online.com/applicationguide/en)

# Objekterkennung in Echtzeit für MCUs mit Edge Impulse FOMO

Von Jan Jongboom, Edge Impulse

Wir Menschen verlassen uns bei vielen alltäglichen Aufgaben – von den einfachsten bis hin zu den komplexesten – in hohem Maße auf unser Sehvermögen. Mit einem Blick wissen wir, ob sich Menschen im Raum befinden, ob ein Elefant in der Nähe ist oder wie viele freie Parkplätze es gibt. Trotz der Bedeutung des Sehens können viele eingebettete Geräte Dinge nicht visuell wahrnehmen. Wäre es nicht großartig, wenn wir *allen unseren Geräten* beibringen könnten, die Welt so zu sehen wie wir?

In den letzten Jahren gab es einige erstaunliche Entwicklungen im Bereich des maschinellen Sehens, die zu Fortschritten bei Dingen wie selbstfahrenden Autos oder biometrischen Einreisekontrollen geführt haben (sehr nützlich, wenn man wie ich viel reist!). Aber diese Anwendungsfälle sind unglaublich rechenintensiv und erfordern teure Grafikprozessoren oder spezielle Beschleuniger.

Erfreulicherweise sind aber nicht alle Aufgaben für maschinellen Sehen so rechenintensiv. Die Beantwortung einer Ja/Nein-Frage wie „Sehe ich einen Elefanten?“ oder „Ist dieses Etikett richtig auf der Flasche angebracht?“ kann eingebetteten Geräten mit eingeschränkten Möglichkeiten einen enormen Mehrwert verleihen. Darüber hinaus lassen sich diese Probleme der Bildklassifizierung sogar von heutigen

Mikrocontrollern lösen.

Stellen Sie sich vor, wir könnten jedes eingebettete Gerät mit noch fortschrittlicheren Bildverarbeitungsfunktionen ausstatten!

## Classification result

Summary	
Name	Parking_data_244.png.2tgg40ai
CATEGORY	COUNT
F1 score	91.43%
car	16

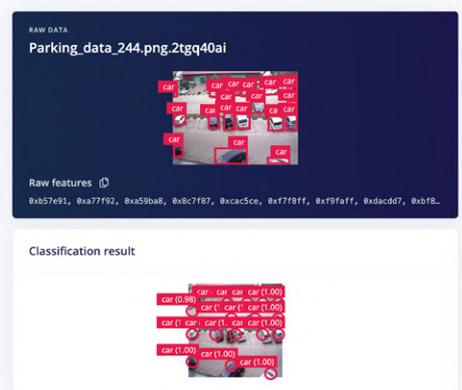
Bild 1. FOMO-Klassifizierung in Edge Impulse Studio.

## Hier kommt FOMO ins Spiel

Wir machen das wahr. Wir haben eine neuartige neuronale Netzwerkarchitektur für die Objekterkennung entwickelt, die wir „Faster Objects, More Objects“ oder FOMO nennen (Bild 1). Es wurde von Grund auf so konzipiert, dass es in Echtzeit auf Mikrocontrollern läuft, sodass Entwickler eingebetteter Systeme nicht befürchten müssen, etwas zu verpassen, wenn es um maschinelles Sehen geht.

## Schnell, schlank und flexibel

FOMO kann auf einer 32-bit-MCU, z. B. einem Arm Cortex-M7, mit einer Bildrate von 30 Bildern pro Sekunde laufen. Und wenn Sie sich das nächste Mal für ein Gerät der Raspberry-Pi-4-Klasse entscheiden, können Sie Objekterkennung mit etwa 60 Bildern pro Sekunde durchführen. Das ist etwa 30 Mal schneller als Mobile-



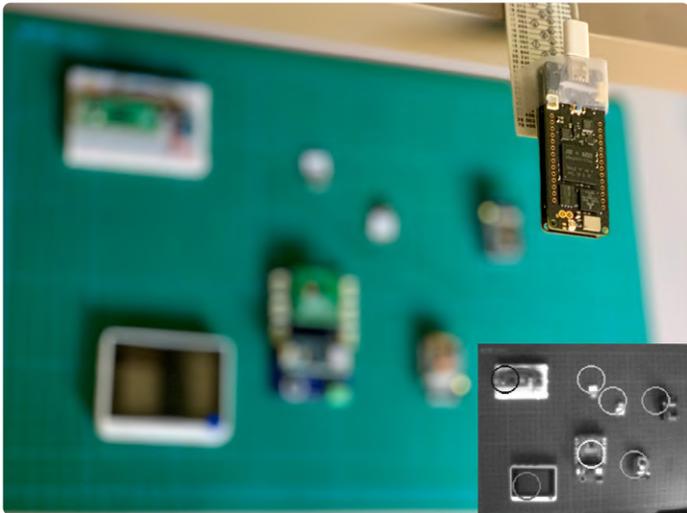
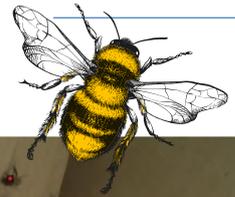


Bild 2. Objekterkennung auf einer Vielzahl von Entwicklungsplatinen, z. B. dem Arduino Portenta.



Bild 3. Eine frühere Version der FOMO-Architektur wurde zum Zählen einzelner Bienen verwendet.

Net SSD oder YOLOv5.

FOMO belegt nur etwa 100 KB RAM, was es möglich macht, die Objekterkennung in Echtzeit auf allen Systemen laufen zu lassen, von stark eingeschränkten Arm-Cortex-M4-Kernen bis hin zu leistungsfähigeren wie den Cortex-M7-Kernen auf dem Arduino Portenta H7 (Bild 2), dem neuen Arduino Nicla Vision (eine weitere 2-Kern-Arm-Cortex-M7/M4-CPU) oder sogar spezialisierten DSPs wie dem Himax WE-I.

FOMO kann von den kleinsten Mikrocontrollern bis hin zu vollständigen Gateways oder GPUs skaliert werden. Dank dieser hohen Flexibilität ist FOMO auch dann einsetzbar, wenn eine Fehlererkennung die Identifizierung von sehr, sehr kleinen Abweichungen innerhalb eines Bildes erfordert.

Bei einer MCU mit streng begrenzter Rechen- und Speicherkapazität ist es am besten, eine Bildgröße von etwa 96x96 Pixeln zu verwenden. Aber mit einem größeren Mikrocontroller sind 160x160 Pixel wahrscheinlich problemlos machbar. Wichtig ist, dass FOMO vollständig faltungsbasiert ist und daher mit jeder beliebigen Bildgröße funktioniert. Wenn Sie eine höhere Granularität, mehr Details oder mehr Objekte benötigen, können Sie einfach die Eingangsauflösung erhöhen.

### Es sieht auch kleine Dinge

Solange die Objekte im Bild eine ähnliche Größe haben und nicht überlappen, kann diese neue Architektur sogar viele sehr

kleine Objekte sehr effektiv erkennen und zählen (Bild 3). Das ist etwas, das MobileNet SSD und YOLOv5, obwohl sie größere und leistungsfähigere Modelle sind, nicht sehr gut können.

### Nichts mehr verpassen

FOMO ist bereits erhältlich, läuft auf einer Vielzahl von Computerplattformen und ist mit Linux-Systemen, Cortex-M-Mikrocontrollern und spezialisierten DSPs kompatibel. Sie brauchen nur noch eine Kamera und Edge Impulse, und schon sind Sie bereit.

Mit FOMO können Sie nahezu jedes kamerabasierte Gerät schnell mit einer Objekterkennung ausstatten und das Risiko vermeiden, etwas zu übersehen. Eine Gefahr, mit der sich Embedded-Entwickler bisher im

Bereich des maschinellen Sehens herum-schlagen mussten (Bild 4).

Um mehr über FOMO zu erfahren und mit Ihrem eigenen Algorithmus zu experimentieren, besuchen Sie [edgeimpulse.com/fomo](https://edgeimpulse.com/fomo). ◀

220207-02

### Über den Autor



Jan Jongboom ist Ingenieur für eingebettete Systeme und Förderer des maschinellen Lernens, immer auf der Suche nach Möglichkeiten, mehr Informationen aus der realen Welt zu gewinnen.

Er hat Geräte ausgeliefert, an den neuesten Netzwerktechnologien gearbeitet, Mikrocontroller simuliert, und es gibt sogar ein Denkmal in San Francisco, das seinen Namen trägt. Derzeit ist er als Mitgründer und CTO von Edge Impulse tätig, der führenden Entwicklungsplattform für eingebettetes maschinelles Lernen mit über 80.000 Projekten.

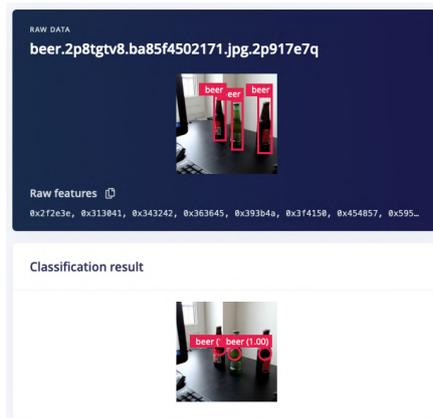


Bild 4. Training mit Centroids bei Bierflaschen: Oben die Quell-Label, unten das Ergebnis der Inferenz.

# Wanderfeldröhren

## Bemerkenswerte Bauteile

Von **Neil Gruending** (Kanada)

Die Welt der HF-Verstärker ist so faszinierend, weil es eine Vielzahl von Herangehensweisen gibt. Heutzutage werden viele HF-Verstärker mit Halbleitern gebaut, aber es gibt immer noch Situationen, in denen Vakuumröhren die einzig geeignete Wahl sind. Wir haben uns in der Vergangenheit mit Klystronen befasst, also wollen wir uns dieses Mal mit der Wanderfeldröhre beschäftigen, einem weiteren unbesungenen Helden der Elektronik.

Eines der faszinierendsten Dinge an Wanderfeldröhrenverstärkern ist ihre Funktionsweise. Sie bestehen aus einem Heizelement, einer Kathode und Beschleunigungselektroden, die ähnlich wie bei einer Kathodenstrahlröhre eine Elektronenkanone bilden, die einen Strom von Elektronen zum Kollektor schickt (Bild 1). Dieser Strom wird durch ein externes Magnetfeld fokussiert, das normalerweise von Permanentmagneten erzeugt wird. Durch Modulation der Geschwindigkeit des Elektronenstrom durch das Signal im Antennenfenster verstärkt die Röhre das

angelegte HF-Eingangssignal.

Da der Elektronenstrom viel langsamer ist als die „HF-Elektronen“, wird das HF-Signal durch einen spiralförmig gewickelten Draht, die sogenannte Helix, geleitet. Dadurch wird das HF-Signal auf die Geschwindigkeit des Elektronenstroms abgebremst.

Während sich die HF-Elektronen durch die Spirale bewegen, modulieren sie die Geschwindigkeit des Elektronenstroms, da die gleichphasigen Elektronen schneller und die gegenphasigen Elektronen langsamer werden. Diese modulierten Elektronen bündeln sich und induzieren ein verstärktes Signal zurück in die Helix, das dann am Ende der Helix mit einem Richtkoppler abgegriffen wird.

Im Vergleich zu Klystronen [1] haben Wanderfeldröhren den Vorteil einer größeren Bandbreite. Außerdem benötigen sie keine Resonanzkomponenten, was sie ideal für Mikrowellenanwendungen mit geringerer Leistung macht, wie bei Radar oder sogar in Raumfahrzeugen und Satelliten. Ein gutes Beispiel ist der S-Band-Verstärker von Collins Radio (Bild 2 und Bild 3), der im Apollo-Raumfahrtprogramm eingesetzt wurde [2]. Es handelte sich um einen kompakten 20-W-Verstärker mit einem Gewicht von knapp 15 kg, der für die gesamte Sprach-, Daten- und Fernsehübertragung an das NASA-Netz der 26 m hohen Parabolantennen auf der Erde verantwortlich war. Im Vergleich dazu verwendete die Bodenstation ein gebündeltes 10-kW-Signal, um mit dem Raumschiff zu kommunizieren.

Obwohl Wanderfeldröhren hauptsächlich

für kommerzielle Anwendungen genutzt wurden und werden, experimentiert eine kleine Gruppe von Enthusiasten immer noch gerne mit diesen wunderbaren kleinen Verstärkern in Amateur-Mikrowellensendern [3]. Die größte Herausforderung besteht jedoch darin, diese Röhren überhaupt zu finden! ◀

210418-02

### Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Senden Sie eine E-Mail an Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).



Bild 2. S-Band Wanderfeldröhrenverstärker „Collins Radio“, der während der Apollo-Mission für die Kommunikation mit der Erde verwendet wurde. (Quelle: Ken Shirriff)



Bild 3. Der Collins-Radio-Verstärker, der mit mehreren Volt arbeitet, war ein dicht gepacktes Gewirr von Koaxialkabeln. (Quelle: Ken Shirriff)

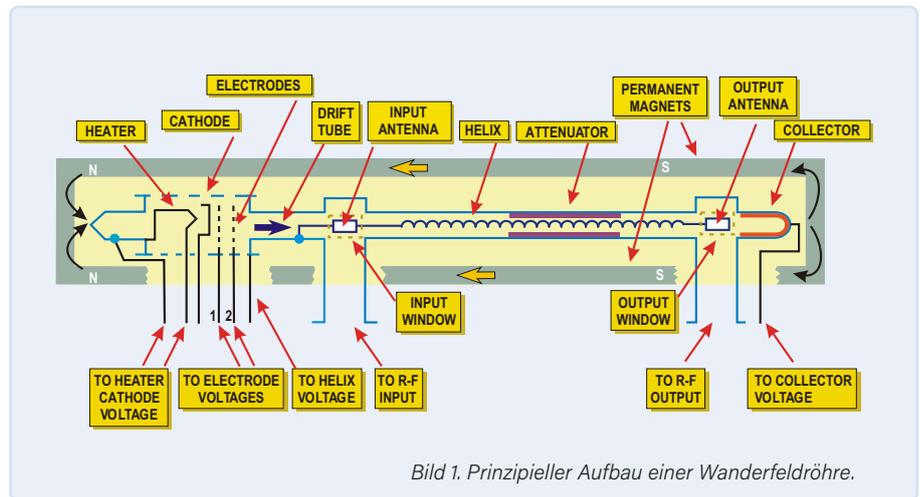


Bild 1. Prinzipieller Aufbau einer Wanderfeldröhre.

### WEBLINKS

[1] N. Gruending, „Klystrone, Bemerkenswerte Bauteile 13“, Elektor 3/2015: <https://bit.ly/3Cga4fF>

[2] K. Shirriff, „Inside a 20-Watt Traveling-Wave Tube Amplifier from Apollo“, Ken Shirriff's Blog, Juli 2021: <https://bit.ly/3ea8lOn>

[3] H. Griffiths, „Travelling Wave Tube Amplifiers“, The National Valve Museum, September 1980: <https://bit.ly/3wA8aCn>

# NB-IoT – ein Überblick



Standards,  
Abdeckung, Verträge  
und Module



Von **Tam Hanna** (Slowakei)

Neugierig auf das schmalbandige Internet der Dinge (NB-IoT)? Vielleicht ist das etwas für Sie. Finden Sie es heraus!

Quelle: shutterstock.com

Neben LoRa und Sigfox bieten sich natürlich auch die Mobilfunknetze an, IoT-Sensordaten zu übertragen. Der Umstieg von EDGE auf UMTS war hierbei von Vorteil: Die Verwendung eines schnelleren Sendesystems kann unterm Strich besser sein als die eines energiesparsameren, aber langsameren. Die immensen Bandbreiten- und Energiebedürfnisse von 4G/LTE aber sorgen dafür, dass diese alte Gleichung nicht mehr so wirklich gilt. Der Stromverbrauch der Transmitter ist wesentlich höher, dazu kommen höhere Preise der Module. Dennoch kann es sich lohnen.

Im Rahmen der allgemein und unwissenschaftlich als LTE bezeichneten Spezifikation 3GPP Release 13 spezifizierte die GSM Association gleich zwei Systeme für das Internet der Dinge – einerseits *Narrowband Internet of Things* (NB-IoT), andererseits auch *LTE-M*, das auch als *LTE Cat-M1* oder *eMTC* bezeichnet wird.

Während LTE-M im Prinzip nur ein „kleineres“ LTE (= 4G) mit einer Bandbreite von 1,4 MHz ist, handelt es sich bei NB-IoT um einen dedizierten Funkstandard für das Internet der Dinge. Der wichtigste Unterschied zeigt sich darin, dass LTE-M über VoLTE auch die Sprachübertragung unterstützt, während ein NB-IoT-System ausschließlich Datenmitteilungen überträgt.

NB-IoT nutzt in seinen nur 180 kHz breiten Kanälen eine Untermenge der in der LTE-Vollversion implementierten Verfahren. Im Uplink kommt eine einfache Version der *Frequency Division Multiple Access* (FDMA) zum Einsatz, für den Downlink setzt man auf *Orthogonal FDMA* (OFDMA). Das verwendete Modulationsverfahren *Quadrature Phase Shift Keying* (QPSK) benötigt aus rechen-technischer Sicht keinen großen Hardware-Aufwand.

Angemerkt sei noch, dass die Einführung von NB-IoT für den Carrier im Allgemeinen „nur“ die Kosten für die neue Hardware

**Tabelle 1. Frequenzbänder.**

Region	Bänder
Europea	3, 8, 20
(Ehemalige) GuS-Staaten	3, 8, 20
Nordamerika	2, 4, 5, 12, 66, 71, 26
Asien, Pazifik (APAC)	1, 3, 5, 8, 18, 20, 26, 28
Sub-Sahara-Afrika	3, 8
Naher Osten und Teile Nordamerikas	8,20
Lateinamerika	2, 3, 5, 29

verursacht. Aufgrund der extrem geringen Bandbreite passt NB-IoT komfortabel in das *Guard Band*, das die LTE-Frequenzpakete umfasst. Andererseits ist es natürlich auch erlaubt, NB-IoT im Standalone-Betrieb zu nutzen.

### Performance im Blick

Der technisch interessanteste Funkstandard hilft nichts, wenn die Übertragungsleistung nicht für die vorgesehene Aufgabe ausreicht. Im Fall von NB-IoT kommt es auf die Version an, denn es gibt Unterschiede zwischen LTE Cat NB1 (Release 13) und LTE Cat NB2 (Release 14). Während die ältere Version im Upstream nur 26 kbit/s schafft, ist Cat NB2 mit 127 kbit/s im Upstream und 159 kbit/s im Downstream deutlich schneller. Zum Vergleich: Klassisches, also Nicht-HSDPA-3G kam in der Anfangszeit auf 380 kbit/s. LTE Cat M1 arbeitet derweil im Up- and Downstream mit rund 1 Mbit/s, Release 14 erweitert auf 4 Mbit/s Upstream und 7 Mbit/s Downstream.

Massiv sind die Unterschiede im Bereich der Latenz: Während LTE-M meist mit 15 ms auskommt, wird bei NB-IoT als „Arbeitsband“ 1,6 s bis 10 s (!) avisiert. Der Modulhersteller Sierra Wireless, der vor allem in den USA beliebt ist, fasst die Lage folgendermaßen zusammen [1]:

„Another important fact to consider is that there are no NB-IoT use cases that LTE-M can't also support. In other words, LTE-M supports any LPWA application, whereas NB-IoT is designed for simpler static sensor type applications.“

Nur in der Version 2 des NB-IoT-Standards findet sich zudem eine Unterstützung für vom Netzbetreiber angebotene Positionierung. Wenn das Modul über kein GPS verfügt oder man eine externe Antenne einsparen möchte, lassen sich auf diese Weise grundlegende Positionsdaten erhalten. Release 14 beschleunigt zudem die Zellen-Neusuche, was besonders für in Bewegung befindliche Geräte von Vorteil ist. Trotz dieser neuen Vorteile von Cat NB2 bleibt LTE-M erste Wahl für Automotive und Co., da die Zellenübergabe hier „intelligenter“ gelöst ist. Die letzte Verbesserung betrifft die Sendeenergie: Nur in Release 14 sind Super-Low-Power-Sender erlaubt, die ihre Arbeit mit nur 14 dBm erledigen [2].

Wer einmal ein 4G-Modul für Verizon entworfen hat, fragt an

dieser Stelle instinktiv nach den verwendeten Bändern. Das nur für Nordamerika wichtige Band 13 hat schon mehr als einem asiatischen oder europäischen Modulanbieter Probleme verursacht. Im unter [3] bereitstehenden *Deployment Guide* der GSM Association findet sich die **Tabelle 1**. Achten Sie darauf, dass Ihr auserwähltes Modul alle Bänder unterstützt, die ihr Wunsch-Carrier verwendet!

### Verfügbarkeit und Verträge

Funkstandards haben logischerweise nur dann Sinn, wenn sie auch praktisch verfügbar sind. Im Fall der beiden IoT-Funkstandards empfiehlt sich die interaktive Weltkarte der GSM Association in **Bild 1** [3] (Stand September 2021). Mexiko ist dabei das einzige Land, wo es (wohl wegen der höheren Reichweite) nur CAT-M gibt; insbesondere in den Flächenstaaten Asiens, witzigerweise aber auch in Osteuropa ist „NB-IoT Only“ weiter verbreitet. In den hochindustrialisierten Regionen Europas, Nordamerikas, Asiens und Australien und Ozeaniens stehen beide Varianten zur Verfügung.

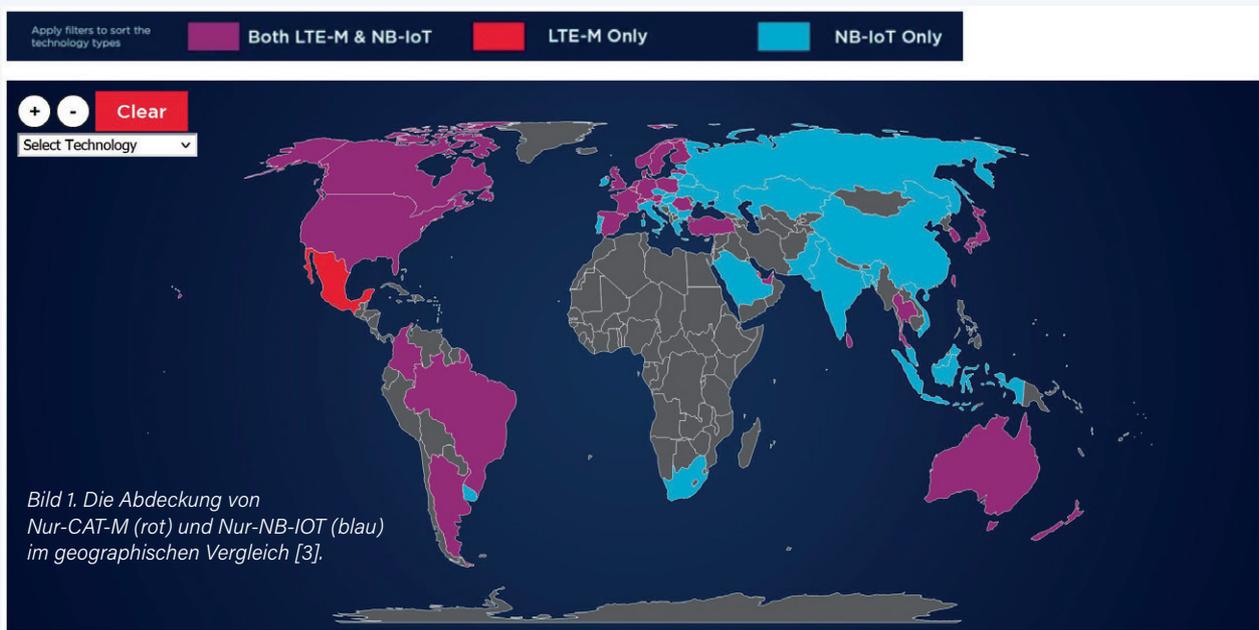
CAT-M-Verträge sind dabei im Allgemeinen gewöhnliche Verträge, bei denen der Volumenverbrauch und die Anzahl der SIM-Karten über die Gesamtkosten entscheiden. Der Vollständigkeit halber sei angemerkt, dass ein IoT-Provider wie *PodGroup* einer im Handel frei gekauften Prepaid-SIM in Kostenfragen oft überlegen ist.

Dass NB-IoT keinen Einschränkungen bezüglich des Duty Cycles unterliegt, bestätigt sich in der Beratungspraxis des Autors übrigens nicht. Spricht man mit dem Mobilfunkanbieter über IoT-Connectivity, ist und bleibt es Verhandlungssache und man bekommt nur allzu bald temporale Paket-Mengengrenzen auferlegt. Genaue Bestimmungen legen die Betreiber allerdings nur recht selten offen, weshalb das folgende Statement von T-Mobile USA [4] Seltenheitswert genießt:

„Join the first nationwide NB-IoT network to power asset tracking, connected cities, and more. Limited time offer; subject to change. Taxes and fees may be additional. Plan includes 10 single-packet transactions per hour at up to 64 Kbps, up to 12 MB. Full service payment due at activation.“

Interessanterweise handelt es sich dabei um eine Einzelmeinung – von Seiten der Hutchison Holding Ltd wurde bestätigt, dass man (bei im Rahmen bleibendem) Datenverkehr auch das gesamte Volumen an einem Tag verfeuern darf. Tom Tesch, der österreichische Pressesprecher von Hutchison, meint dazu:

„Die Datenrate von NB-IoT ist – entsprechend des Standards – sehr gering und vorrangig geeignet, um einzelne Messwerte oder Stati zu übertragen. Aus diesem Grund werden für NB-IoT Geräte sehr selten mehr als 5-10 MB pro Monat benötigt. Für bandbreitenintensive Anwendungen – zum Beispiel Übertragung von Bildern oder Videos – sind 3G/4G und natürlich 5G die geeignetere Technologie. Aktuell gibt es keine Begrenzungen, wann das Volumen genutzt werden darf/ kann – es kann daher auch alles an einem Tag verbraucht werden.“



## Wie beginnen?

Nach diesen grundlegenden Überlegungen ist es an der Zeit, darüber nachzudenken, wie man NB-IoT in praktische Systeme einbinden kann. Logischerweise ist die Entwicklung eines „hauseigenen“ Modems für das durchschnittliche Unternehmen nicht machbar - das „Design in“ von Funkmodulen haben wir Ihnen in der Vergangenheit, beispielsweise in Elektor 5-6/2021 [5], detailliert vorgestellt. Falls Sie nicht sofort mit der Entwicklung eines eigenen Boards beginnen möchten, bietet sich - die Verfügbarkeit von Qualcomm-Chips erweist sich hierbei als Problem - die Verwendung „schlüssel-fertiger“ Evaluation-Boards an.

Neben dem NBIOT-BG96-SHIELD von Avnet, das ein BG96-Modul von Quectel integriert, gibt es mit dem 5G NB IoT click ein Board von Mikro-Elektronika, das ein Cinterion-Modul zur Verfügung stellt. Arduino schickt mit dem MKR NB 1500 ebenfalls eine kleine Evaluationsplatte ins Rennen - beide Boards kosten allerdings mehr als 50 Dollar. Mag ein Evaluation-Board in vielen Fällen noch mit einer SIM ausgeliefert werden, so ist das massive Rollout von auf NB-IoT basierenden Lösungen alles andere als einfach. Ursache dafür ist, dass Netzbetreiber die Technologie noch nicht für Endkunden „paketiert“ haben. Dies wird von den Betreibern auch offen zugegeben, wie das Statement von Hutchison zeigt:

*„NB-IoT ist ein sehr junges und neuartiges Netzwerk. Da es kaum noch Geräte am Markt gibt, handelt es sich bei der Zielgruppe vorwiegend um Geschäftskunden aus der Hardware- (und Software-)Entwicklung. Das heißt, dass unser Angebot sich aktuell daher auch ausschließlich an Geschäftskunden richtet, für die wir in Zuge einer Beratung ein individuelles Angebot erstellen.“*

Bei der Arbeit mit „gewöhnlichen“ 2G/3G/4G-Systemen bietet die Nutzung von „virtuellen“ Mobilfunkanbietern wie der PodGroup einen Ausweg. Eine diesbezügliche Anfrage wurde dahingehend beantwortet, dass sich NB-IoT derzeit - insbesondere für „globale“ Lösungen, die mit einer einzelnen SIM-Karte auskommen müssen - noch nicht wirklich eignet.

Ursache dafür ist, dass erstens das NB-IoT-Rollout selbst nur vergleichsweise eingeschränkt ist und zweitens die Roamingverträge zwischen den verschiedenen Netzbetreibern im Allgemeinen (noch) nicht an den neuen NB-IoT-Funkstandard angepasst wurden. Analog zum Doppelbesteuerungsabkommen zwischen Staaten gilt auch hier, dass solche Anpassungen sehr viel Zeit in Anspruch nehmen. „Internationales“ NB-IoT-Roaming steckt also noch in den Kinderschuhen!

## Lohnt es sich?

Die Suche nach einem praktischen Modul, das nur NB-IoT anbietet, ist eine durchaus haarige Angelegenheit. Bei Quectel gibt es selbst bei der kleinsten BC660-Serie zwei Varianten: die eine nur mit NB-IoT, die andere mit eMTC und NB-IoT. Auch in größeren Systemen wie dem Bestseller BG95/BG96 sind beide Funkstandards vertreten. Gängige Marktpreise für die Module findet man nur bei SoS Electronic: das BC660K-GL kostet 7,63 € in Einzelstückzahlen, die Version mit LTE-M und NB-IoT ist dort nicht im Programm. Ein BG96 kostet dort 19 €.

Ergiebiger wird die Jagd bei u-blox [6]. Mit der SARA-N3-Modulfamilie steht ein ausschließlich für die NB-IoT-Protokollfamilie vorgesehenes Modul zur Verfügung, einen reinen CAT-M-Chip bieten die Schweizer indes nicht an (siehe **Bild 2**).

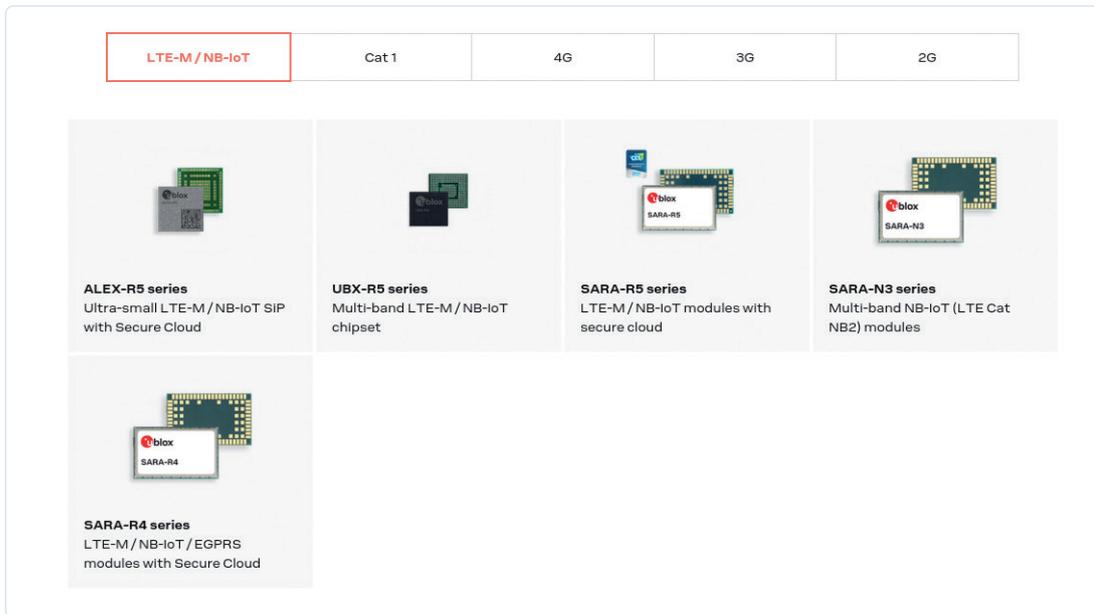


Bild 2. Die Suche nach einem reinen CAT-M-Modul ist unergiebig [6].



Bild 3. Traditionelles steirisches Windrad als Vogelscheuche (Quelle: Martin Geisler, CC BY-SA 4.0 [7]).

Bei Gemalto – seit der Übernahme durch Thales ist die Webseite verwirrender als je zuvor – gibt es mit dem EMS31 ein reines CAT-M-Modul, während der ENS22 ein im selben Formfaktor gehaltenes reines NB-IoT-Modul ist. Der tschechische Distributor Sectron ermöglicht einen Vergleich der Kosten: der EMS31 kostet 14 €, der ENS22 dagegen nur 8 €.

In den als *Hardware Interface Description* bezeichneten Datenblättern finden sich dann auch (verklauusulierte) Informationen zum Stromverbrauch. Der höchstmögliche Stromverbrauch des EMS31 tritt bei der Arbeit im Band 4 auf und liegt (bei 3,8 V Versorgungsspannung) bei 239 mA. Für den ENS22 werden im Band 28 als Höchststrom 404 mA aufgeführt, wobei allerdings zu beachten ist, dass Funkmodule diese Peaks oft nur für sehr kurze Zeit benötigen.

### Was springt für Sie heraus?

NB-IoT funktioniert aus technischer Sicht problemlos; hat man einen Carrier unter Vertrag genommen, so beschränkt sich der Aufwand für den Netzbetrieb (anders als bei einem hauseigenen

## WEBLINKS

- [1] LTE-M vs. NB-IoT: What are the Differences?: [www.sierrawireless.com/iot-blog/lte-m-vs-nb-iot/](http://www.sierrawireless.com/iot-blog/lte-m-vs-nb-iot/)
- [2] Wikipedia über Schmalband-IoT: [https://en.wikipedia.org/wiki/Narrowband\\_IoT](https://en.wikipedia.org/wiki/Narrowband_IoT)
- [3] GSMA: Weltkarte der IoT-Funkstandards: [www.gsma.com/iot/deployment-map/](http://www.gsma.com/iot/deployment-map/)
- [4] T-Mobile über Schmalband-IoT: <https://t-mo.co/3EC5Jo4>
- [5] Tam Hanna, Keine Angst vor dem Mobilfunkmodul, Elektormag 5-6/2021: [www.elektormagazine.de/200062-02](http://www.elektormagazine.de/200062-02)
- [6] Mobilfunkmodule von u-blox: [www.u-blox.com/en/cellular-modules](http://www.u-blox.com/en/cellular-modules)
- [7] Klapotetz: <https://bit.ly/3nOr0Fb>

LoraWAN) auf einen Anruf beim Anwalt. Der vergleichsweise geringe Spitzen- und Ruhestromverbrauch der Module trägt zudem zu geringen Kosten für die Energieversorgung bei.

Ob sich die Sache am Ende wirklich lohnt, ist – analog zu Parade-Steueroasen wie Dubai oder Monaco – vor allem eine Sache der Skalierung. Wer fünf Modems im Jahr kauft, fährt mit einem „vollen“ und vielleicht um ein paar Euro teureren 4G-Modul samt dickerem Schaltregler besser: Die traurige Lebenserfahrung des Autors zeigt nämlich, dass man in der Praxis immer wieder den „anderen“ Funkstandard braucht, schon deshalb, weil am Ende doch nicht jeder Funkturm jeden Funkstandard unterstützt.

Beim Kauf von 50.000 Modems, die alle an einen Kunden gehen, sieht die Sache natürlich anders aus. Braucht das Bürgermeisteramt von *Großdorf am Klapotetz* (**Bild 3**) NB-IoT, so wird der lokale Carrier wahrscheinlich aufrüsten – und die eingesparten Dollars helfen bei der großen Stückzahl. ❏

180021-02

### Sie haben Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an den Autor unter [tamhan@tamoggemon.com](mailto:tamhan@tamoggemon.com) oder an die Redaktion unter [redaktion@elektor.de](mailto:redaktion@elektor.de).

### Ein Beitrag von

Text: **Tam Hanna**

Redaktion: **Rolf Gerstendorf**

Layout: **Giel Dols**



### PASSENDE PRODUKTE

➤ **E-Book: IoT Home Hacks with ESP8266 (SKU 19159)**  
[www.elektor.de/19159](http://www.elektor.de/19159)

Anzeige

## Garantieren Sie Produkte, die zu 100 % authentisch sind

Mouser war der erste nach SAE AS6496 akkreditierte Distributor

Die größte Auswahl elektronischer Bauelemente auf Lager™

[mouser.de/authentic](http://mouser.de/authentic)



# Dragino LPS8 Indoor Gateway

Schnell zum eigenen LoRaWAN-Gateway



Bild 1. Dragino LPS8 Indoor Gateway (Quelle: Dragino [5]).

Von Mathias Claußen (Elektor)

Wie man eigene Elektronik mit einem LoRaWAN verbinden kann, haben wir schon des Öfteren beschrieben. Wer sich nicht in Reichweite eines LoRaWAN-Gateways befindet oder einfach noch tiefer ins Thema einsteigen will, kann ein eigenes Gateway betreiben. Wir haben das mit dem günstigen Dragino LPS8 Indoor Gateway ausprobiert.

LoRaWAN ist ein Thema, das schon öfter in Elektor behandelt wurde. Es ist recht einfach, einen LoRaWAN-Knoten wie zum Beispiel ein Sensorboard aufzubauen. Dazu stehen LoRa-Module zur Verfügung, die man an kompakte Controllerboards anschließen kann, etwa auf Basis eines STM32 oder Raspberry Pi Pico [1][2]. Doch damit die Daten, die ein solcher Knoten über LoRa sendet, auch weiter transportiert werden, braucht es eine Gegenstelle; in diesem Fall ein LoRaWAN-Gateway, das Daten über LoRa entgegennimmt und an eine Internetplattform wie *The Things Network* weiterleitet. Man kann hierzu ein vorhandenes Gateway nutzen (viele werden von Freiwilligen betrieben) oder ein eigenes Gateway einrichten. Ich habe seit mehr als einem Jahr ein *Dragino LPS8 Indoor Gateway* im Einsatz.

## Dragino LPS8

Das *Dragino LPS8 Indoor Gateway* (Bild 1) besitzt ein Kunststoffgehäuse. Die Elektronik im

Inneren wird von einem kleinen AR9331-Prozessor von Atheros (heute Qualcomm) mit 400 MHz befeuert. Mit 64 MB RAM und 16 MB Flash ist dieses Gehirn des Gateways sicher nicht mit einem Raspberry Pi Zero 2 W zu vergleichen - doch für die Funktionen, die ein Gateway ausführen muss, ist der Prozessor mehr als ausreichend. Der Atheros AR9331 integriert neben der CPU auch WLAN nach 802.11 b/g/n und einen 10/100-Mbit-LAN-Port. Für die Datenrate, die per LoRaWAN transportiert werden soll, sind diese Geschwindigkeiten mehr als ausreichend. Auch muss das Gateway selbst keine große Rechenleistung bereitstellen, da es sich nur um das Management des integrierten LoRa-Transceiver-Moduls kümmert und die Daten ins Internet weiterreicht. Ein Blockdiagramm ist in Bild 2 zu sehen.

Beim LoRa-Transceiver handelt es sich um eine Kombination aus einem LoRa-Baseband-Chip SX1308 von Semtech (Blockdiagramm in Bild 3) und zwei SX1257

## LPS8 System Overview:

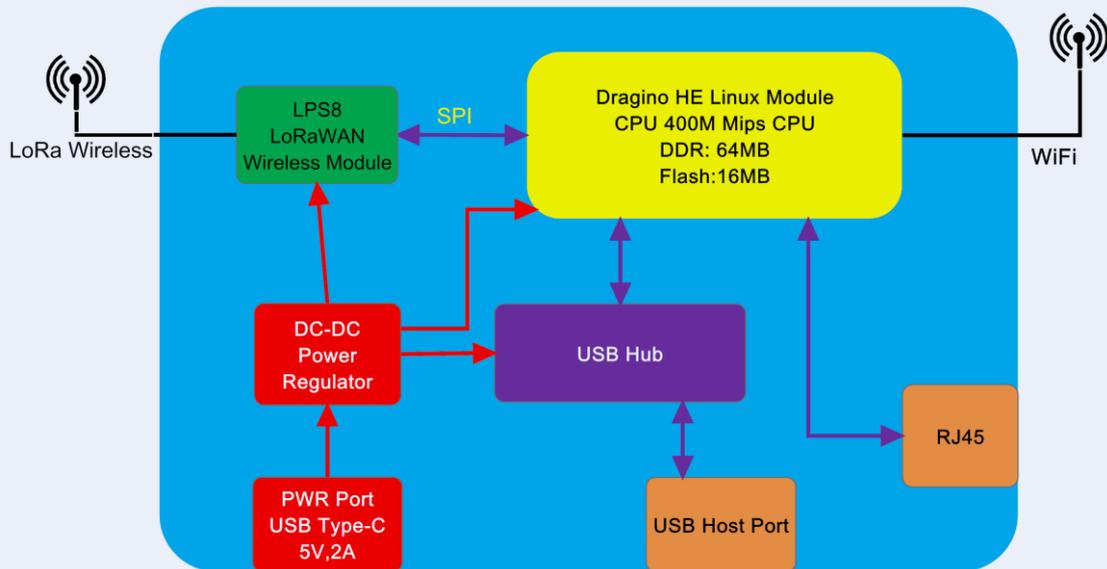


Bild 2. Blockdiagramm Dragino LPS8 (Quelle: Dragino [6]).

Front-End-Modulen (Blockdiagramm in **Bild 4**). Diese Kombination sorgt für die Umsetzung von der Funkschnittstelle zu Ethernet. Das Gateway ist sehr genügsam, was die Leistungsaufnahme angeht, 5 V und maximal 2 A (10 W) müssen von einem Netzteil bereitgestellt werden.

Wie der Name des Gateways schon verrät, ist das Gerät für den Einsatz innerhalb eines Gebäudes gedacht, die Umgebung sollte also trocken und staubarm sein. Das Gebiet, das mit dem Gateway abgedeckt werden kann, ist kleiner als eines, das ein Outdoor-Gateway abdecken kann. Letztere werden meist auf entsprechenden Sendemasten installiert.

### Handbuch, Firmware und Einrichtung

Zur Einrichtung des Gateways kann das Handbuch von Dragino in aktueller Version [3] heruntergeladen werden; es wurde seit dem Release des Produktes gepflegt und spiegelt die Funktionen und Updates der aktuellen Firmware wider. Löblich - es könnte einen traurig stimmen, dass sich nicht auch andere Firmen um eine solch gute Produkt-Dokumentation kümmern.

Auch die Firmware wird gut gepflegt. Das aktuelle Release ist vom 4.11.2021 (Stand 15.12.2021) [4]. Wenn möglich, empfiehlt sich vor der Inbetriebnahme ein Firmware-update auf die aktuelle Version, damit Bugs und Sicherheitslücken so weit wie möglich entgegengewirkt wird.

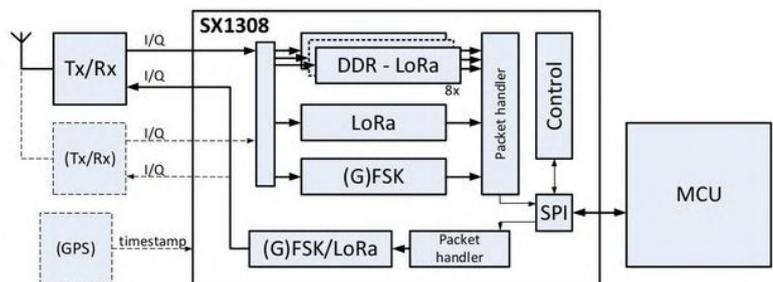


Bild 3. Blockdiagramm SX1308 (Quelle: Semtech [7]).

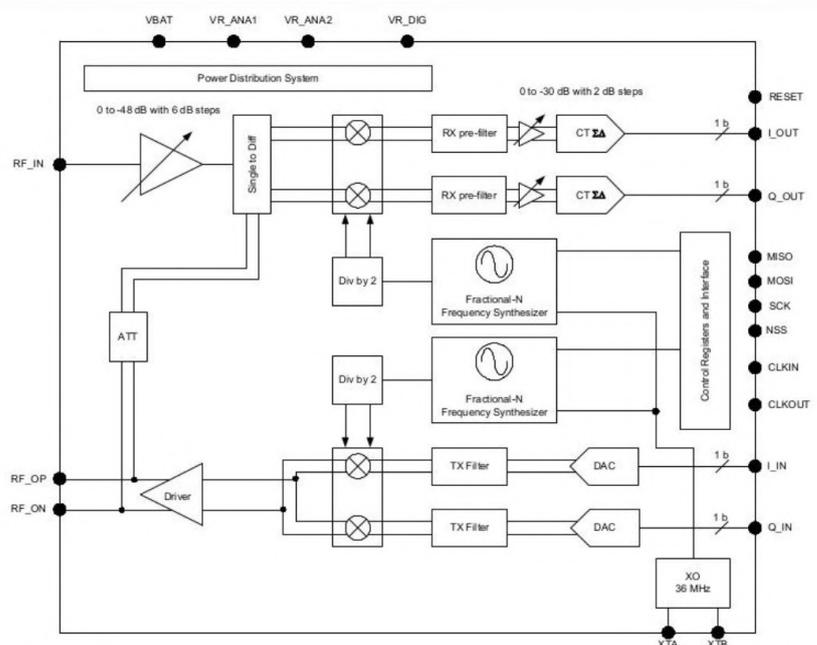


Bild 4. Blockdiagramm SX1257 (Quelle: Semtech [8]).

Das Handbuch führt einen durch die Einrichtung des Gerätes. Dabei müssen nur das Netzwerk passend konfiguriert und die Einstellungen für das passende LoRaWAN (zum Beispiel *The Things Network*) vorgenommen werden. Ab diesem Punkt ist das LoRaWAN-Gateway einsatzbereit (**Bild 5**).

### OpenWRT-Unterbau

Auch wenn die erste Seite des Webinterfaces das noch nicht vermuten lässt: Als Basis des *Dragino LPS8 Indoor Gateways* kommt ein OpenWRT zum Einsatz. Damit steht nicht nur ein LoRaWAN-Gateway zur Verfügung, es lassen sich auch etliche weitere Einstellungen vornehmen, die man von Routern her kennt (IP-Adressen, Weiterleitung, WLAN).

Dank des OpenWRT-Unterbaus lässt sich auch ein LTE- oder 5G-Modem an den USB-Port des Gateways anbinden, falls am Standort des Gerätes keine andere Verbindung zum Internet möglich sein sollte. Wer mag, kann auch per SSH auf die Kommandozeile des Linux zugreifen (auf eigene Gefahr!). So lassen sich per Webinterface oder Kommandozeile weitere Pakete installieren, um Funktionen hinzuzufügen.

### Abschließende Worte

Das *Dragino LPS8 Indoor Gateway* ist ein unauffälliger Vertreter eines LoRaWAN-

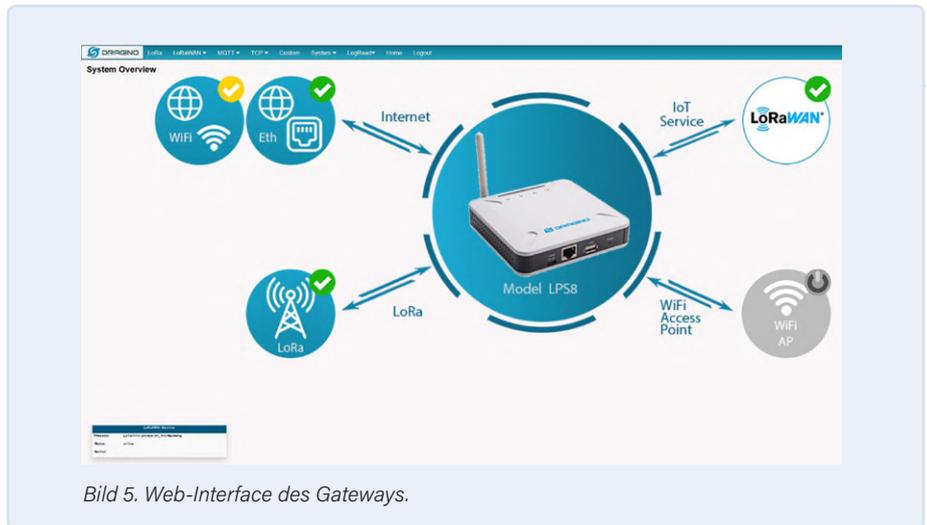


Bild 5. Web-Interface des Gateways.

Gateways, und in diesem Falle ist dies ein Pluspunkt. Während des über einjährigen Einsatzes hat das Gerät bei mir klaglos seinen Dienst verrichtet. Für den Betrieb meiner LoRaWAN-Knoten macht das Gateway einen guten Job und auch die Abdeckung des Gebäudes (und der umliegenden) ist mehr als erfreulich. Wer also auf der Suche nach einem preiswerten LoRaWAN-Gateway ist, sollte einen Blick auf das *Dragino LPS8 Indoor Gateway* im Elektor-Shop [5] wagen. ◀

210680-02

### Sie haben Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an den Autor unter [mathias.claussen@elektor.com](mailto:mathias.claussen@elektor.com) oder kontaktieren Sie Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).

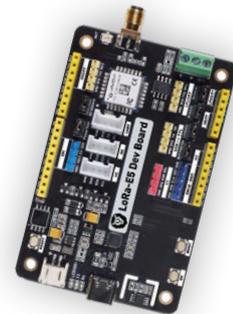
### Ein Beitrag von

Text: Mathias Claußen  
Redaktion: Jens Nickel  
Layout: Harmen Heida



### PASSENDE PRODUKTE

- **Dragino LPS8 Indoor LoRaWAN Gateway (868 MHz) SKU 19094**  
[www.elektor.de/19094](http://www.elektor.de/19094)
- **Entwicklungskit LoRa-E5 STM32WLE5JC von Seeed Studio (SKU 19956)**  
[www.elektor.de/19956](http://www.elektor.de/19956)



### WEBLINKS

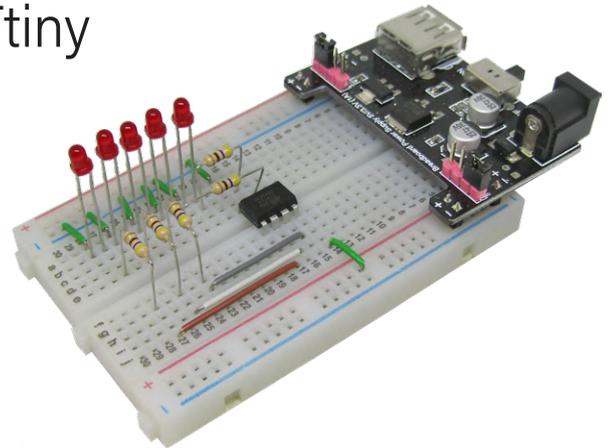
- [1] Mathias Claußen, „LoRaWAN - ein einfacher Einstieg“, *Elektormag* 3-4/2020 : [www.elektormagazine.de/magazine/elektor-140/57101](http://www.elektormagazine.de/magazine/elektor-140/57101)
- [2] Mathias Claußen, „LoRa mit dem Raspberry Pi Pico“, *Elektormag* 7-8/2021: [www.elektormagazine.de/magazine/elektor-178/59750](http://www.elektormagazine.de/magazine/elektor-178/59750)
- [3] Handbuch *Dragino LPS8 Indoor Gateway*: [www.dragino.com/downloads/index.php?dir=LoRa\\_Gateway/LPS8/](http://www.dragino.com/downloads/index.php?dir=LoRa_Gateway/LPS8/)
- [4] Download Firmware *Dragino LPS8 Indoor Gateway*:  
[www.dragino.com/downloads/index.php?dir=LoRa\\_Gateway/LPS8/Firmware/Release/](http://www.dragino.com/downloads/index.php?dir=LoRa_Gateway/LPS8/Firmware/Release/)
- [5] Bildquelle Gateway: [www.dragino.com/media/k2/galleries/148/LPS8-10.jpg](http://www.dragino.com/media/k2/galleries/148/LPS8-10.jpg)
- [6] User Manual: <https://bit.ly/LPS8-user-manual>
- [7] Datenblatt Front-End SX1257: <https://sforce.co/3fZmy1f>
- [8] Datenblatt LoRa-Transceiver SX1308: <https://sforce.co/32zxqAv>

# ATtiny-Mikrocontroller mit C und Assembler erforschen

## Beispiel-Kapitel: I/O-Ports des ATtiny

Von Warwick A. Smith (Südafrika)

I/O-Ports steuern die Pins eines Mikrocontrollers und können sie individuell als Eingänge oder Ausgänge konfigurieren. Diese weit gefasste Aussage sollte eigentlich in keinem Einsteigerkurs und in keiner Einführung in die Mikrocontroller-Programmierung fehlen. Um aber die I/O-Fähigkeiten eines Mikrocontrollers wirklich zu verstehen und zu nutzen, muss man tiefer in die Materie eindringen. In diesem Artikel demonstriert der Elektor-Buchautor Warwick Smith die Assembler-Programmierung an dem beliebten ATtiny-Mikrocontroller. Interessiert? Dann schauen wir es uns an!



**Anmerkung der Redaktion:** Dieser Artikel ist ein Auszug aus dem 376-seitigen Buch *Explore ATtiny Microcontrollers using C and Assembly Language* (W. Smith, Elektor 2021). Der Auszug wurde formatiert und leicht bearbeitet, um den redaktionellen Standards und dem Seitenlayout von Elektor Mag zu entsprechen. Da es sich um einen Auszug aus einer größeren Veröffentlichung handelt, können sich einige Begriffe in diesem Artikel auf Diskussionen an anderer Stelle im Buch beziehen. Der Autor und die Redaktion haben ihr Bestes getan, um solche Fälle auszuschließen, und sind gerne bereit, bei Fragen zu helfen. Kontaktinformationen finden Sie im Kasten Fragen oder Kommentare?

Die Konfiguration und Steuerung der I/O-Ports erfolgt beim ATtiny13(A) und ATtiny25/45/85 über vier Register. Wenn ein Pin als Eingang konfiguriert ist, kann ein auf dem AVR laufendes Programm den Logikpegel des Pins lesen. Wenn ein Schalter an den Pin angeschlossen ist, kann der Logikpegel am Pin gelesen werden, um festzustellen, ob der Schalter offen oder geschlossen ist. Wenn ein Pin als Ausgang konfiguriert ist, kann er verwendet werden, um den Logikpegel des Pins high (auf Logikpegel 1) oder low (auf Logikpegel 0) zu schalten. Ein Ausgangspin kann verwendet werden, um eine LED anzusteuern, wie es in dem an anderer Stelle in diesem Buch beschriebenen LED-Blinkprojekt geschehen ist.

### Konfigurieren von I/O-Pins als Ausgänge in Assembler

In diesem Abschnitt sehen wir, wie man mehr als einen Pin als Ausgang konfigurieren kann, indem man einen 8-poligen ATtiny mit fünf LEDs verwendet, die mit Vorwiderständen an den Pins angeschlossen sind. Der Programmcode konfiguriert die LEDs als 5-Bit-Binärzähler, der von Null an aufwärts zählt.

**Bild 1** zeigt einen Schaltplan eines ATtiny13(A) oder ATtiny25/45/85 AVR-Mikrocontrollers mit fünf LEDs, die an die I/O-Pins PBo...

PB4 angeschlossen sind. Der Pin PB5 des ATtiny-Mikrocontrollers wird in der Schaltung als debugWIRE-Pin für die Programmierung und das Debugging des Mikrocontrollers eingesetzt. Für

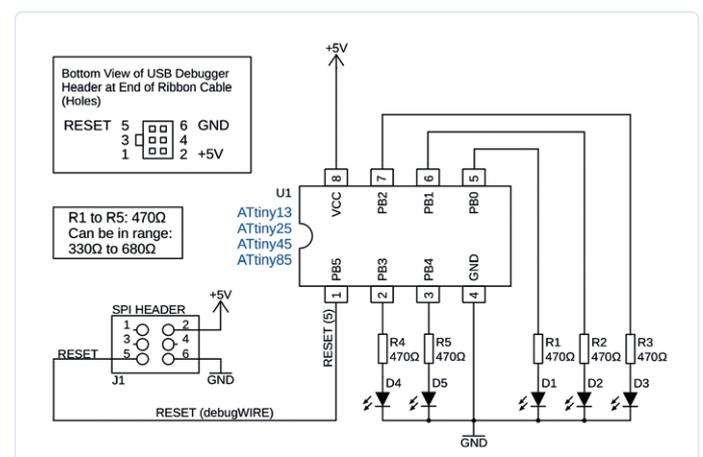


Bild 1. Schaltplan des Fünf-LED-Zählers.

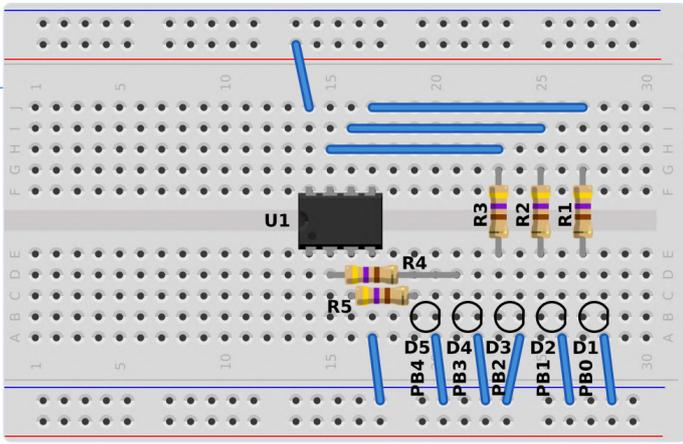


Bild 2. Breadboard-Layout des Fünf-LED-Zählers.

diese Konfiguration muss ein Programmierer/Debugger wie der Atmel-ICE oder AVR-Dragon verwendet werden.

Ein USB-Programmiergerät, das nur dem Programmieren dient, funktioniert nicht im debugWIRE-Modus und kann kein Debugging durchführen. **Setzen Sie deshalb die DWEN-Fuse nicht bei einem reinen USB-Programmiergerät, da dieses den AVR nicht aus dem DebugWIRE-Modus zurückholen kann.** Die Idee, den debugWIRE-Modus in dieser Beispielschaltung zu verwenden, ist es, die anderen Pins des AVR freizugeben, die normalerweise im ISP/SPI-Programmiermodus festgelegt sind.

### Reine Programmiergeräte

Um das folgende Beispielprogramm auf die Schaltung in Bild 1 zu laden und den Zählwert auf den LEDs anzuzeigen, kann ein USB-Programmiergerät mit reiner Programmierfunktion verwendet werden. Verdrahten Sie die LEDs wie in Bild 1 und Bild 2 gezeigt. Die originalen USBasp- und USBtinyISP-Designs besitzen Schutzwiderstände auf den Leitungen zum Programmiergerät, die sowohl das Programmiergerät wie auch den Ziel-AVR-Chip schützen. Der Arduino Uno, der als ArduinoISP eingesetzt wird, verfügt nicht über solche Schutzwiderstände, aber diese können der MOSI- und der SCK-Leitung des Arduino Uno zum Ziel-AVR hinzugefügt werden. Im originalen USBasp-Design werden 270-Ω-Schutzwiderstände auf den Leitungen MOSI, SCK und RESET verwendet, beim originalen USBtinyISP-Design sind 1,5-kΩ-Widerstände auf den Leitungen MOSI und RESET platziert.

Diese Schutzwiderstände verhindern einen Kurzschluss, falls einer der Ziel-AVR-Pins eine Ausgangsspannung liefert und das Programmiergerät eine andere.

### Peripherie, die die Programmierung stört

Obwohl die Schaltung in Bild 1 mit den angeschlossenen LEDs und Vorwiderständen über die ISP/SPI-Schnittstelle programmiert werden kann, ist es möglich, dass an anderen Schaltungen Hardware angeschlossen ist, die die Programmierung stört. Wenn periphere Hardware an den Pins eines zu programmierenden AVR die Programmierung stört, gibt es einige Lösungen für dieses Problem. Natürlich können diejenigen Leser, die einen debugWIRE-fähigen USB-Programmierer/Debugger wie den Atmel-ICE besitzen, den Ziel-AVR einfach in den debugWIRE-Modus versetzen und so nur einen Pin für die Programmierung verwenden. Eine andere Lösung besteht darin, den AVR auf einem anderen Breadboard über ISP/SPI zu programmieren und ihn anschließend an die Zielschaltung einzubauen.

Alternativ kann ein AVR mit mehr Pins verwendet werden, aber die freien Portpins stimmen nicht immer mit den gleichen Pins des gleichen Ports eines ATtiny im 8-Pin-PDIP-Gehäuse überein. Wenn man zum Beispiel die Portpins PBo...PB4 wie in der Schaltung in Bild 1 verwendet, stehen bei den 14-poligen ATtiny24/44/84-AVRs nicht fünf aufeinanderfolgende freie Pins zur Verfügung, da die ISP/SPI-Pins sowohl Pins von Port A als auch von Port B belegen. Beim 20-poligen ATtiny26/261/461/861 ist der gesamte Port A frei, wenn ein ISP/SPI-Programmierer angeschlossen ist, aber das bedeutet, dass die Software geändert werden muss, um Port A statt Port B zu verwenden. Glücklicherweise sind beim 20-poligen ATtiny2313/4313 die Pins PBo...PB4 frei, wenn ein ISP/SPI-Programmierer angeschlossen ist.

### Der AVR im debugWIRE-Modus

Damit der Mikrocontroller wie in Bild 1 über eine einzige debugWIRE-Leitung programmiert werden kann, müssen zunächst alle ISP-Header-Leitungen des USB-Programmierers/Debuggers an den Mikrocontroller angeschlossen und dann die DWEN-Fuse gesetzt werden, um den AVR in den debugWIRE-Modus zu versetzen. Dann sollte sich der AVR im debugWIRE-Modus befinden. Wenn nicht, schließen Sie Ihren Programmierer/Debugger wie den Atmel-ICE oder AVR-Dragon an. Überzeugen Sie sich, dass sie die DWEN-Fuse setzen können. Sobald dies geschehen ist, können alle ISP-Header-Anschlüsse aus der Schaltung entfernt werden, mit Ausnahme von RESET, +5 V (Vcc) und GND, wie in Bild 1 dargestellt.

### Aufbau der 5-LED-Zählschaltung auf dem Breadboard

Wenn Sie alle Bauteile zusammen haben, können Sie die Schaltung aus Bild 1 auf einem Breadboard aufbauen. Die fünf LEDs werden auf dem Breadboard in einer Reihe platziert, wobei die Positionen den Pins PBo...PB4 entsprechen, beginnend mit PBo (LED D1) auf der rechten und PB4 (LED D5) auf der linken Seite. Mit anderen Worten, wir wollen eine LED-Reihe, bei der PBo mit der LED rechts verbunden ist, PB1 mit der LED links daneben, PB2 mit der LED an dritter Stelle von rechts und so weiter, wie in Bild 2 zu sehen ist. Das Bild zeigt nur die Umrisse der LEDs, damit sie nicht die Sicht auf die Drahtverbindungen und Widerstände versperren. Wenn Sie nicht über die entsprechende Hardware verfügen, können Sie die Programme auch mit einem Simulator nachvollziehen.

### Assembler-Code für die 5-LED-Zählschaltung

Starten Sie ein neues Assembler-Projekt im Microchip-Studio-AVR mit dem Namen `led_count_asm`. Kopieren Sie den Code aus Listing 1 in die Datei `main.asm` des Projekts und ersetzen Sie dabei den Skeleton-Code.

Wenn Sie die Hardware aus Bild 1 verwenden, wählen Sie im Microchip-Studio das Hardware-Tool (Ihren Debugger) wie den Atmel-ICE, mit debugWIRE als Schnittstelle. Klicken Sie dazu auf das Hammersymbol in der zweiten oberen Symbolleiste. Wenn Sie den Simulator verwenden, wählen Sie stattdessen `Simulator` als Werkzeug. Wenn Sie ein „Hobby“-Programmiergerät verwenden, erfahren Sie im Folgenden, wie Sie das Programm auf den Ziel-ATtiny laden können.

Das Programm `led_count_asm` konfiguriert die Pins PBo...PB4 als



### Listing 1. led\_count\_asm : main.asm

```

; Set up pins PB0 to PB4 as output pins
    ldi    r16, 0b0001_1111
    out    DDRB, r16
    clr    r18           ; Clear count register
loop:
    out    PORTB, r18    ; Display count on LEDs
    rcall delay
    inc    r18           ; Increment count
    andi   r18, 0b0001_1111 ; Clear unused bits
    rjmp  loop
; Delay subroutine
delay:
    ldi    r16, 0xff
delloop1:
    ldi    r17, 0xff
delloop2:
    dec    r17
    brb    SREG_Z, delloop2
    dec    r16
    brbc   SREG_Z, delloop1
    ret

```

Ausgänge, damit die angeschlossenen LEDs durch das Programm ein- und ausgeschaltet werden können. Das Programm zeigt eine aufsteigende Binärzahl oder Zählung auf den LEDs an, beginnend bei 0 (was bedeutet, dass alle LEDs aus sind). Wenn der Zählwert seinen Maximalwert erreicht hat (alle LEDs an), kehrt er anschließend zu Null zurück und beginnt von neuem. Jede „Aus“-LED steht für eine binäre Null oder eine logische 0, und jede „An“-LED steht für eine binäre Eins oder eine logische 1. Es ist wichtig, die LEDs wie in Bild 2 anzuordnen, damit die Zählung korrekt angezeigt wird, wobei PBo/D1 das LSB und PB4/D5 das MSB des Zählwerts darstellt. Um das Programm zu erstellen und in den AVR in der physikalischen Hardware aus Bild 1 und Bild 2 zu laden, benutzt man beim Atmel-ICE oder AVR Dragon das Symbol *Start Without Debugging* in der oberen Symbolleiste des Microchip-Studios (oder die Tastenkombination *Ctrl + Alt + F5*), um das Programm in den AVR zu laden. Wenn die Debugger-Schnittstelle korrekt eingerichtet ist und der Chip sich im DebugWIRE-Modus befindet, wird das Programm hochgeladen und beginnt sofort zu laufen. Die aufsteigende Binärzahl wird von den LEDs angezeigt. Wenn Sie ein reines Hobby-USB-Programmiergerät verwenden, laden Sie das Programm mit der entsprechenden Funktion auf den Ziel-AVR. Ist der Code richtig eingegeben, die Schaltung richtig verdrahtet und das Programm nach der Eingabe gespeichert und kompiliert, sehen Sie den binären Zählwert auf den LEDs aufwärts zählen.

Wenn Sie nicht über die Hardware verfügen, können Sie den Zählwert dennoch mit dem Simulator im Microchip-Studio anzeigen. Mit den Icons *Start Debugging* und *Break* kann das Programm schrittweise durchlaufen werden. Wenn der Simulator gestartet ist, öffnen Sie mit *Debug Windows I/O* das I/O-Fenster im oberen Menü und klicken Sie dann auf *I/O Port (PORTB)*. Gehen Sie mit dem *Step Over*-Icon oder der *F10*-Tastaturtaste durch das Programm. Am unteren Rand des I/O-Fensters ist unter PORTB der Zählwert zu sehen, der auf den LEDs angezeigt würde, wenn sie denn angeschlossen wären. Der Zählerstand in PORTB wird jedes Mal aktualisiert, wenn die *OUT*-Anweisung in der Hauptschleife übersprungen wird.

### Funktionsweise des LED-Zähl-Assemblerprogramms

Die Hälfte des Programmcodes von *led\_count\_asm* besteht aus der Verzögerungs-Subroutine, die auch in dem an anderer Stelle in diesem Buch besprochenen LED-Blinkprogramm verwendet wurde. Diese Subroutine wird in der Hauptschleife des Programms einmal aufgerufen und verzögert die Zählung der LEDs so weit, dass sie für das Auge sichtbar ist, ansonsten würde die Zählung viel zu schnell ablaufen. Öffnen Sie das Projekt *led\_count\_asm* im Microchip-Studio und den Code von *main.asm*.

Die ersten beiden Anweisungen des Programms setzen die Pins PBo...PB4 als Ausgänge für die Ansteuerung der LEDs. Dazu werden Bits im DDRB-Register gesetzt (oben in **Bild 3**). Jedes Bit in diesem Register entspricht einem Pin des Mikrocontrollers. Das Bit DDB0 entspricht zum Beispiel dem Pin PBo, DDB1 dem Pin PB1 und so weiter.

Wenn ein Bit in DDRB auf logisch 1 gesetzt wird, wird der entsprechende Pin zu einem Ausgang. Wenn ein Bit in DDRB auf logisch 0 gesetzt wird, wird der entsprechende Pin zum Eingang (was auch die Voreinstellung aller Pins beim Einschalten oder Zurücksetzen

DDRB : Port B Data Direction Register. Configures pin direction: 0 = Input, 1 = Output.  
**Address:** 0x17 **Initial Value:** 0b0000\_0000 **Bits 5 to 0:** Read/Write (R/W)

7	6	5	4	3	2	1	0
-	-	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0

PORTB : Port B Data Register. Drives output pins. Enables input pin pull-ups with logic 1.  
**Address:** 0x18 **Initial Value:** 0b0000\_0000 **Bits 5 to 0:** Read/Write (R/W)

7	6	5	4	3	2	1	0
-	-	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0

PINB : Port B Input Pins Register. Read input pin state. Write logic 1 to toggle output pin.  
**Address:** 0x16 **Initial Value:** Depends on level on pin. **Bits 5 to 0:** Read/Write (R/W)

7	6	5	4	3	2	1	0
-	-	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0

Bild 3. I/O-Port-Register des ATtiny13(A) und des ATtiny25/45/85.

ist). Am Anfang des Programms wird zunächst 0b0001\_1111 in das Register R16 (unten in Bild 3) geschrieben und dann mit einem *OUT*-Befehl in das DDRB-Register übergeben. Es ist notwendig, diesen konstanten Wert zuerst in R16 zu laden, da es keinen Befehl gibt, um einen konstanten Wert direkt in ein I/O-Register wie DDRB zu schreiben. Das Schreiben von 0b0001\_1111 über R16 in DDRB setzt die Bits DDB0...DDB4 und macht die Pins PBo...PB4 zu Ausgängen.

Obwohl es vier Register zur Steuerung des I/O-Ports B gibt, sind in Bild 3 nur drei dargestellt. Das vierte Register MCUCR besitzt nur ein einziges Bit, das für den Anschluss B gilt. Dieses Bit ist ein globales Pull-up-Sperrbit, das in diesem Kapitel nicht verwendet wird. Dieses Register wird im Datenblatt für den ATtiny13, ATtiny13A oder ATtiny25/45/85 im Abschnitt I/O-Ports beschrieben. Das Register R18 in der Mitte von Bild 3 wird im Programm

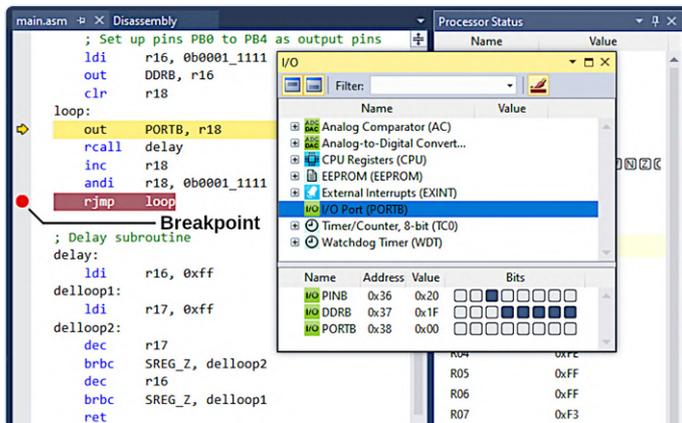


Bild 4. Einfügen eines Haltepunkts im Microchip-Studio-Debugger.

verwendet, um einen inkrementellen Zählwert zu erhalten, der auf den LEDs angezeigt wird. R18 wird vor der Hauptschleife mit `CLR` auf 0 zurückgesetzt, so dass die Zählung bei 0 beginnt.

In der Hauptschleife wird der Inhalt von R18 mit dem `OUT`-Befehl an das PORTB-Register gesendet. Auch hier entspricht jedes Bit dieses Registers einem Pin des Mikrocontrollers. Bei Pins, die mit DDRB als Ausgänge eingerichtet wurden, erscheinen die in das PORTB-Register geschriebenen Logikpegel an diesen Pins. Bei Pins, die als Eingänge definiert wurden, aktiviert eine logische 1 in PORTB einen internen Pull-up-Widerstand an diesen Pins. Eine logische 0 deaktiviert dagegen den Pull-up-Widerstand an dem zugehörigen Pin.

Da die Pins PBo...PB4 als Ausgänge konfiguriert sind, erscheint der in PORTB geschriebene Zählwert an diesen Pins als logische Pegel, die die LEDs ein- und ausschalten. Eine logische 1 im Zählwert schaltet die entsprechende LED ein und eine logische 0 schaltet die entsprechende LED aus.

Nachdem der Zählwert mit dem `OUT`-Befehl in PORTB geschrieben wurde, wird das Verzögerungsunterprogramm aufgerufen, um den Zählwert eine Zeit lang auf den LEDs zu belassen. Das Register R18 wird dann mit dem `INC`-Befehl um 1 erhöht, so dass der enthaltene Zählwert inkrementiert wird. Mit `ANDI` werden durch die Maske `0b0001_1111` die obersten drei Bits des Zählwerts in R18 gelöscht, so dass diese oberen Bits immer Null sind. Am Anfang der Schleife, wenn R18 wieder nach PORTB geschrieben wird, werden die obersten drei Bits immer mit 0 belegt, da sie mit `ANDI` gelöscht wurden. Wenn das Ende der Schleife mit der `RJMP`-Anweisung erreicht ist, beginnt die Programmausführung wieder am Anfang der Schleife, wobei der neue Zählwert in PORTB geschrieben und auf den LEDs angezeigt wird.

Bei diesem Programm sind einige Dinge zu beachten: Die I/O-Register DDRB und PORTB werden in beiden Programmen zum Einrichten und Steuern von Port B des Mikrocontrollers verwendet. Das LED-Blinkprogramm muss nur einen einzigen Pin steuern und verwendet daher die Befehle `SBI` und `CBI`, um ein einzelnes Bit in den I/O-Registern direkt zu setzen und zu löschen - keines der Arbeitsregister R0 bis R31 wird dafür benötigt. Das LED-Zählprogramm muss aber auf fünf Pins beziehungsweise LEDs gleichzeitig

zugreifen und verwendet daher den `OUT`-Befehl, um mehrere Bits in einem Rutsch in die Register zu schreiben.

In einem Assemblerprogramm ist es wichtig, die Verwendung von Registern im Auge zu behalten. Zum Beispiel wird zu Beginn des Programms ein augenblicklicher Wert in R16 geladen. R16 wird in diesem Fall nur vorübergehend gebraucht und kann daher später im Programm anderweitig verwendet werden, ohne dass sein Wert vorher gespeichert werden muss. So wird das Register im Unterprogramm `delay` wieder verwendet. Da R16 und R17 in der Verzögerungs-Subroutine gebraucht werden, wurde R18 für nichts anderes als die Speicherung des Zählwerts gewählt. Dadurch wird der Zählwert niemals überschrieben. Wenn ein Programm sehr groß wird und keine freien Register für spezielle Zwecke zur Verfügung stehen, können die Werte in den verwendeten Registern mit `PUSH`- und `POP`-Befehle am Anfang und Ende eines Unterprogramms erhalten werden.

## Breakpoints im Debugger verwenden

Mit dem Simulator oder der physischen Hardware und zum Beispiel dem Atmel-ICE können Sie den Debugger verwenden, um schrittweise durch das Programm zu gehen, indem Sie auf `Start Debugging` und `Break` klicken, wie wir es zuvor getan haben. Zeigen Sie die I/O-Port-Register an, indem Sie das I/O-Fenster in Microchip-Studio öffnen. Wählen Sie bei laufendem Debugger `Debug Windows I/O` und klicken Sie dann im I/O-Fenster auf `I/O Port (PORTB)`, wie in **Bild 4** gezeigt.

Wenn die Programmausführung in die Hauptschleife eintritt, ist es zweckmäßig, einen Haltepunkt (Breakpoint) an der `RJMP`-Anweisung zu setzen und die gesamte Schleife auszuführen, indem Sie auf die Schaltfläche `Continue` in der Symbolleiste klicken (Tastaturtaste F5). Ein Haltepunkt kann auf die `RJMP`-Anweisung gesetzt werden, indem man im Fenster des Microchip-Studios auf die graue Marge links neben dem Befehl klickt. Dadurch erscheint ein roter Punkt in diesem Bereich, der anzeigt, dass ein Breakpoint für diesen Befehl gesetzt wurde (siehe Bild 4). Alternativ können Sie auch auf die Anweisung klicken, die Sie mit einem Haltepunkt versehen möchten, so dass der Cursor darauf platziert wird, und dann `Debug Toggle Breakpoint` aus dem oberen Menü wählen oder die F9-Tastaturtaste drücken. Sobald der Haltepunkt gesetzt ist, können Sie die Schaltfläche `Continue` in der Symbolleiste oder F5 drücken, um die gesamte Schleife zu durchlaufen und nur an deren Ende zu unterbrechen. Auf diese Weise können Sie sehen, wie sich der Zählerstand im PORTB im I/O window und im Register R18 im `Processor Status window` um 1 erhöht, ohne dass Sie in der Schleife von Befehl zu Befehl springen müssten.

Um den Haltepunkt von der `RJMP`-Anweisung zu entfernen, klicken Sie auf den roten Punkt links der Anweisung oder verwenden Sie das Menü oder F9, wenn der Cursor steht auf der `RJMP`-Anweisungszeile. Setzen Sie einen Haltepunkt beim Befehl `INC R18`. Drücken Sie nun F5, um die Schleife zu durchlaufen. Der neue Haltepunkt sorgt dafür, dass das PORTB-Register und R18 denselben Wert enthalten, wenn die Programmausführung endet. Wenn die Programmausführung bei `RJMP` stoppt, wird R18 inkrementiert, bevor der neue Wert in PORTB geschrieben wird, so dass die beiden Werte in den Debugger-Fenstern von Microchip Studio nicht synchron ansteigen.

Die Software, die der Autor zur Unterstützung des Buches erstellt hat, kann kostenlos heruntergeladen werden. Gehen Sie zu [1], scrollen Sie nach unten zu den *Downloads* und klicken Sie auf den Dateinamen *Software\_Explore ATtiny Microcontrollers using C and Assembly Language*. Speichern Sie die ZIP-Archivdatei (etwa 29 kB) lokal und entpacken Sie sie. ◀

220045-02

### Ein Beitrag von

Text und Illustrationen: Warwick Smith  
 Redaktion: Jan Buiting  
 Übersetzung: Rolf Gerstendorf  
 Layout: Giel Dols

### Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Senden Sie eine E-Mail an den Autor unter [warwsmi@axxess.co.za](mailto:warwsmi@axxess.co.za) oder an Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).

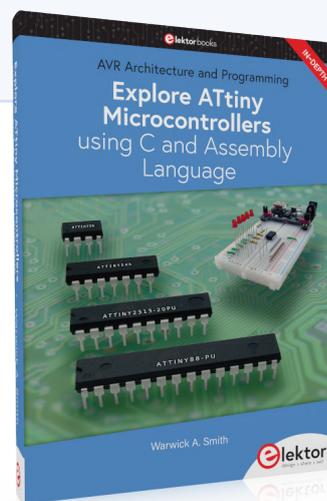
### WEBLINK

[1] Buchressourcen/Infoseite : [www.elektor.de/20007](http://www.elektor.de/20007)



### PASSENDE PRODUKTE

- Explore ATtiny Microcontrollers using C and Assembly Language  
 Buch SKU 20007: [www.elektor.de/20007](http://www.elektor.de/20007)  
 E-Buch SKU 20008: [www.elektor.de/20008](http://www.elektor.de/20008)



Anzeige

**PERFORMANCE.**  
**RELIABILITY.**  
**SERVICE.**

Optokoppler von Würth Elektronik



**WÜRTH  
 ELEKTRONIK**  
 MORE THAN  
 YOU EXPECT

**WE meet @ PCIM Europe**

Halle 6, Stand 402

#### Optokoppler von Würth Elektronik

Würth Elektronik präsentiert mit seinen Optokopplern die neueste Produkterweiterung im Bereich Optoelektronik. Das innovative Design sorgt durch seine koplanare Struktur und das hochwertige Silizium (Epoxy) für eine vollständige Reflexion im Inneren. Die koplanare Bauart stellt sicher, dass der Isolationspalt während des Produktionsprozesses fixiert bleibt und bietet perfekte Isolation und Schutz für Ihre Anwendung. Die innere Reflexion sorgt für einen stabilen CTR über den gesamten Temperaturbereich und ermöglicht einen hohen CTR bei geringem Strombetrieb.

Verfügbar in allen branchenüblichen Verpackungen. Alle Binnings ab Lager verfügbar. Muster kostenlos erhältlich: [www.we-online.com/optocoupler](http://www.we-online.com/optocoupler)

- Innovatives koplanares Design
- Hochwertige Silizium-Verkapselung
- Leadframe aus Kupfer für hohe Zuverlässigkeit
- Stabiler CTR über den gesamten Temperaturbereich
- Hoher CTR bei geringem Strom

# Projekt 2.0

Korrekturen, Updates und Leserbriefe

Redaktion: Ralf Schmiedel, Jens Nickel

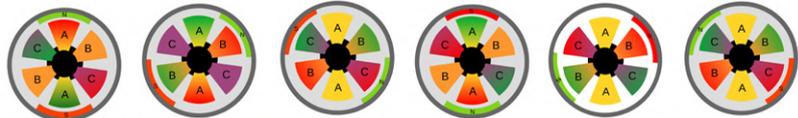
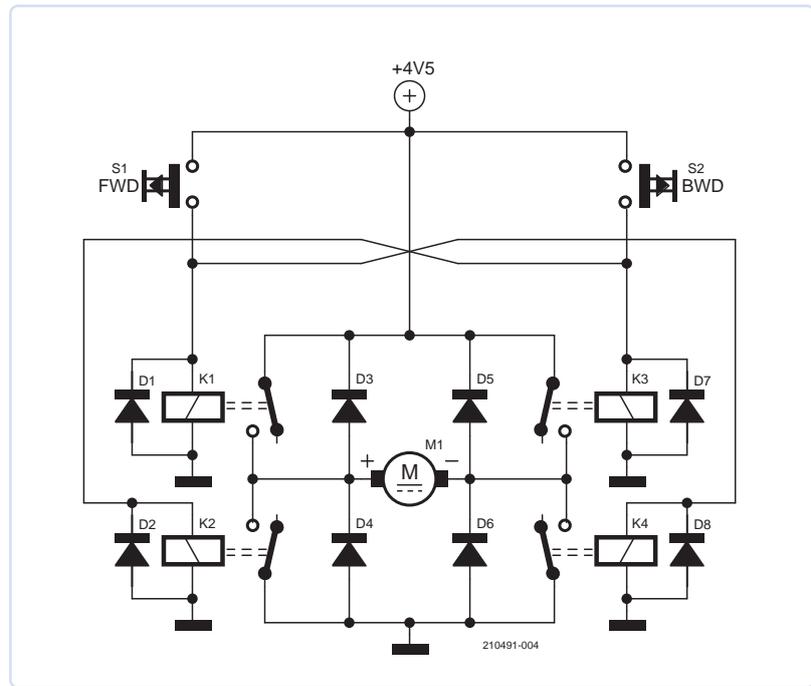
## Motorsteuerung mit H-Brücken

Heft 01-02/2022, S. 6 (210491)

Leider hat sich in einem der Schaltpläne ein Fehler eingeschlichen. In Bild 4 des Artikels (H-Brücke mit Relais) fehlt die Verbindung zwischen Motor M und den Relais K3 und K4. Daher wollen wir hier den vollständigen Schaltplan mit allen Verbindungen nachliefern. Danke an unsere aufmerksamen Leser, die uns diesen Fehler mitgeteilt haben.

Auch bei den Bildunterschriften ist ein Fehler passiert: In Bild 14 und Bild 20 muss es natürlich ein L298N sein und kein L294N.

In Bild 22 geben die Teilbilder die Rotation nicht korrekt wieder. Der letzte Schritt bei der Rotation ist eine Kopie des dritten Schrittes.



## Einfacher Erdschlussprüfer

Heft 09-10/2021, S. 106 (200576)

Im Hinweis „Vorsicht Hochspannung!“ am Ende des Artikels wird der Erdschlussprüfer Fehlerstrom-Schutzschalter genannt, dies führt zur Verwirrung.

Zum anderen wurde beim Beschreiben der Messmethode nicht noch einmal explizit auf das Abschalten der Netzspannung beim Messen hingewiesen. Das kann tödlich sein!

Der Erdschlussmesser kann zur Fehler-suche an ortsveränderlichen Geräten im spannungslosen Zustand sehr gut eingesetzt werden.

Jörg Stäudle

Unser Leser Jörg Stäudle hat Recht.  
Redaktion Elektor

## Kompassrose mit dem GY-271

Heft 09-10/2021, S. 78 (200597)

Ich finde den Artikel über den GY-271 sehr interessant, aber ich verstehe immer noch nicht ganz, warum er in einer Acht bewegt werden muss?

Jac Hetteema

Ich hatte das Problem „wie kann man am Handy die Kompassgenauigkeit verbessern“ recherchiert und war unter anderen auf diese Webseite gestoßen: <https://mediamag.mediamarkt.at/detail/smartphone-kompass-kalibrieren-in-5-schritten.html>. In einer Acht bewegen ist also ein Vorschlag von Handy-Experten zur Verbesserung des Kompassverhaltens. Sie stellt sicher, dass wirklich zwei Mal die 360 Grad durchlaufen werden.  
Rolf Hase (Autor des Artikels)

## Elektronische Last für DC und AC

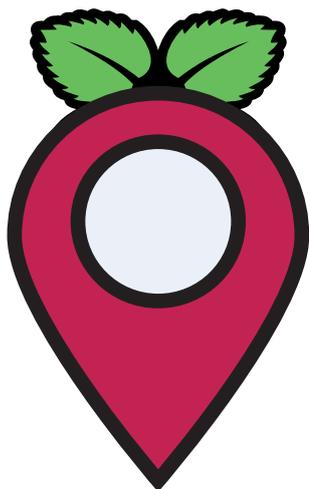
Heft 09-10/2021, S. 20 (191206)

Im Schaltplan auf Seite 23 liegt ein Fehler vor. D3 ist als 3V3-Zener angegeben, es sollte aber ein 10-V-Typ sein. ◀

220052-02

## Möchten Sie selbst etwas beitragen?

Gerne können Sie sich an die Elektor-Redaktion wenden unter der E-Mail-Adresse: [redaktion@elektor.de](mailto:redaktion@elektor.de).



# LoRa-GPS-Tracker - ein Update

## Daten abholen und visualisieren mit dem Raspberry Pi

Von Hans Schneider (Deutschland)

Im Artikel „LoRa-GPS-Tracker“ aus Elektor 11-12/2020 [1] wird beschrieben, wie man die Trackingdaten mit Node-RED vom TTN-Server abholt und auf einer Karte visualisiert. Prinzipiell sollte das nicht nur mit einem PC, sondern auch mit einem Raspberry Pi funktionieren, doch es gibt Stolpersteine. Einer unserer Leser hat die Lösung gefunden und den Node-RED-Flow auf den Einplatinencomputer portiert.

Der „LoRa-GPS-Tracker“ stellt Hard- und Software vor, mit der man bewegliche Objekte tracken kann. Ein kompaktes Tracker-Modul sendet die Daten eines GPS-Sensors über LoRa zu einem Gateway, das diese an einen Server von *The Things Network* weiterleitet. Dort sind die Rohdaten - die erforderliche Berechtigung vorausgesetzt - manuell über das Internet abrufbar. Komfortabler geht das natürlich automatisch über ein Framework wie Node-RED, das auch gleich die Darstellung auf einer (Welt-) Karte steuern kann. Der beim Artikel mitgelieferte Node-RED-Flow funktioniert auf dem PC auch sehr gut. Doch dies hat einen entscheidenden Nachteil: Der energiehungrige PC muss zu diesem Zweck laufen und der Node-RED-Server gestartet sein, sonst werden keine Daten empfangen.

In vielen Haushalten mit technikaffinen Mitgliedern gibt es aber mindestens einen Raspberry Pi, auf dem eine Software wie Pi-Hole [2], Homebridge [3] oder Flightradar [4] permanent darauf warten, Daten zu empfangen oder zu senden. Was also hindert uns daran, diesen Raspberry Pi noch mit einer weiteren Aufgabe zu betrauen? Allerdings ist bei einer Homebridge-Installation dringend von der Installation von Node-RED abzuraten, da die benötigten Ressourcen teilweise die Homebridge-Installation stören. Aber auch sonst geht es nicht ganz ohne Schwierigkeiten.

### Installation von Node-RED auf dem Raspberry Pi

Zunächst einmal muss Node-RED auf dem Raspberry Pi installiert

werden. Es wird vorausgesetzt, dass der Leser mit der Bedienung des Raspberry Pi über das Terminal vertraut ist. Zunächst ist das Einloggen im Terminal über SSH erforderlich, gefolgt von folgendem Befehl: `bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)`. Eventuell muss vorher noch `sudo apt install build-essential git` eingegeben werden, damit npm in der Lage ist, die binary-Module zu bauen, die installiert werden müssen. Das Script braucht einige Zeit, um durchzulaufen, man sollte bis zu einer halben Stunde einkalkulieren. Ist das Script fertig, sorgt die Anweisung `sudo systemctl enable nodered.service` im Terminal dafür, dass Node-RED automatisch startet, wenn der Pi eingeschaltet oder neu gebootet wird. Mit der Anweisung `sudo systemctl disable nodered.service` kann man das auch wieder abstellen.

Gestartet wird der Node-RED-Server mit `node-red-start`, gestoppt mit `node-red-stop`. `node-red-restart` stoppt und startet den Server anschließend wieder. Das Terminal kann nach dem Start von Node-RED verlassen werden, der Server läuft weiter. Wenn man das Terminal neu startet, kann man mit `node-red-log` den Output-Service des Servers starten und sieht im Terminal dessen Ausgabe. Das ist praktisch, wenn man nachsehen möchte, ob noch alles in Ordnung ist, ohne eine neue Instanz des Servers zu starten.

### Benötigte Module in Node-RED installieren

Module installiert man in Node-RED entweder über den Paletten-Manager im Browserfenster oder über npm im Terminal. Meine Empfehlung ist die Installation über npm, weil das nach meiner Erfahrung zuverlässiger funktioniert. Dann muss vorher allerdings Node-RED gestoppt werden.

Wenn der Node-RED-Service dagegen auf dem Raspberry läuft, kann man sich über einen Browser auf dem PC in die Benutzeroberfläche einloggen. Dazu gibt man in der Adresszeile ein: `http://:1880`. Man landet im Flow-Editor. Hier werden zuerst die benötigten Module installiert, im Flow-Editor geht das über den besagten Paletten-Manager. Die benötigten Module sind folgende (teilweise wurden sie bereits bei der Installation von Node-RED installiert):

- > `node-red`
- > `node-red-contrib-worldmap`
- > `node-red-node-rbe`



Bild 1. MQTT-Node in Node-RED.

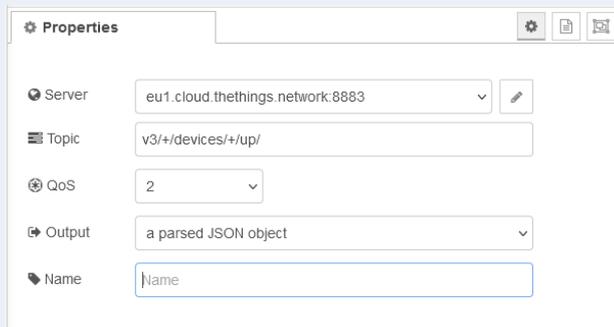


Bild 2. Einstellungen des MQTT-Nodes.

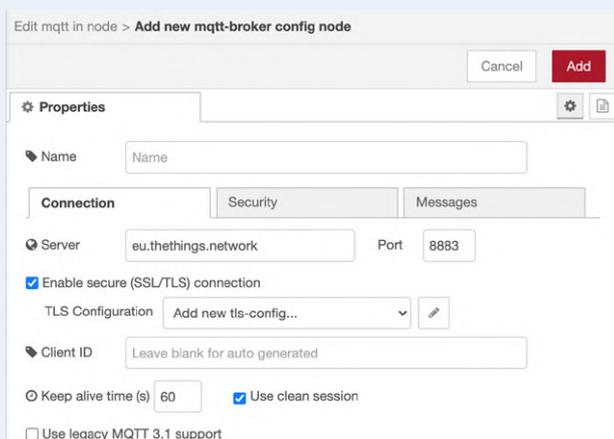


Bild 3. Hinzufügen eines neuen Servers.

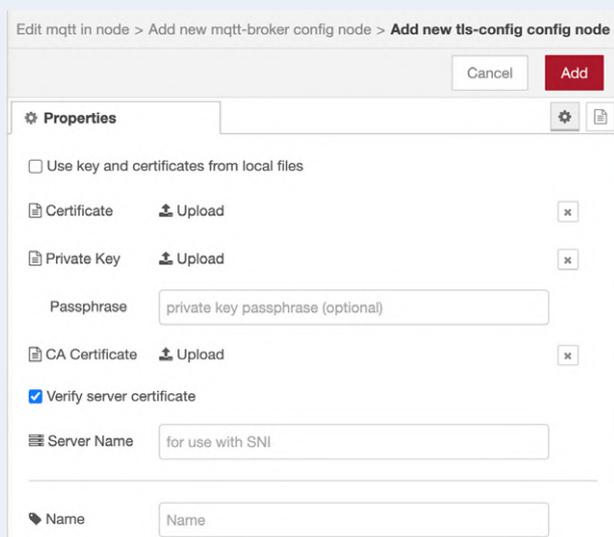


Bild 4. TLS-Konfiguration für den Server.

- > *node-red-node-sqlite*
- > *node-red-node-base64*
- > *node-red-tail*

Bei der Installation über den Paletten-Manager ist Geduld angesagt; wenn die Aktivitätsanzeige ausgeblendet wurde, dauert es immer noch etwas, bis am oberen Fensterrand die Erfolgsmeldung erscheint. Falls die Installation über den Paletten-Manager nicht klappen sollte, verlässt man den Editor wieder, stoppt im Terminal den Node-RED-Service auf dem Raspberry und macht es händisch. Die Anweisung `npm i --unsafe-perm node-red-node-sqlite` beispielsweise installiert das Modul *node-red-sqlite*, mit den anderen Modulen geht das entsprechend.

Wenn *node-red-node-sqlite* das Modul *node-red-sqlitedb* nicht mit installiert, muss es nachinstalliert werden. Das geht nur über npm, weil der Paletten-Manager in diesem Fall einen Konflikt mit dem Modul *node-red-node-sqlite* feststellt.

## Fehlende TTN-Module

So weit, so gut. An dieser Stelle wird vielleicht aufgefallen sein, dass die im Ursprungsartikel besprochenen TTN-Module (TheThingsNetwork) fehlen. Dies hat den einfachen Grund, dass diese die Umgebung des Raspberry nicht unterstützen. Wenn man zum Beispiel das Modul *node-red-contrib-ttn* geladen hat, ist es anschließend nicht mehr möglich, den Node-RED-Service auf dem Raspberry zu starten. Es hat ein wenig länger gedauert, bis ich herausgefunden habe, warum der Server seinen Dienst verweigerte - und gerade diese Fehlersuche möchte ich den Lesern ersparen.

Die Lösung ist das in Node-RED eingebaute Modul *mqtt*: Es wird in den Ursprungsflow anstelle des TTN-Uplink-Moduls aufgenommen. Sie können meinen speziell für den Raspberry angepassten Flow von der Elektor-Labs-Seite des LoRa GPS Trackers [5] downloaden und verwenden. Importiert wird der Flow, indem man die heruntergeladene JSON-Datei einfach in das Fenster des Flow-Editors zieht. Übrigens: Der für den Raspberry angepasste Flow läuft anstelle des Flows mit den speziellen TTN-Nodes auch auf dem PC.

Im Flow-Editor sollte man nun auf das *mqtt*-Symbol (**Bild 1**) doppelklicken und im sich öffnenden Fenster (**Bild 2**) die Einstellungen editieren. Unter *Topic* wird „v3/+ /devices/+ /up“ wie im Bild dargestellt eingetragen, um die Daten aller Devices in der TTN-Applikation zu erhalten. Als *Output* bleibt „a parsed JSON object“ als Default-Einstellung ausgewählt. Im Feld *Server* kann mit dem Stiftsymbol ein neuer Server eingetragen werden und es öffnet sich das Fenster aus **Bild 3**. Als *Name* wird die Application ID aus der TTN-Konsole eingesetzt, als *Server* *eu1.cloud.thethings.network*. Der vorgegebene Port 1833 ist der Port für eine Nicht-TLS-Verbindung; er muss auf 8883 geändert werden. Schließlich fehlt noch das Häkchen bei *Enable secure (SSL/TLS) connection*. Ein Klick auf das Stiftsymbol des Feldes *TLS Configuration* ruft das nächste Fenster (**Bild 4**) auf.

Hier sollten die Voreinstellungen durch einen Klick auf *Add* akzeptiert werden. Zurück im Fenster *mqtt-broker config node* wählt man die Registerkarte *Security* (**Bild 5**). Der *Username* ist die Application ID aus der TTN-Konsole plus einem *@ttn* am Ende. Als Passwort muss ein Access Key mit mindestens Messages-Rechten in der TTN-Konsole



Bild 5. Security Tab des MQTT-Nodes.

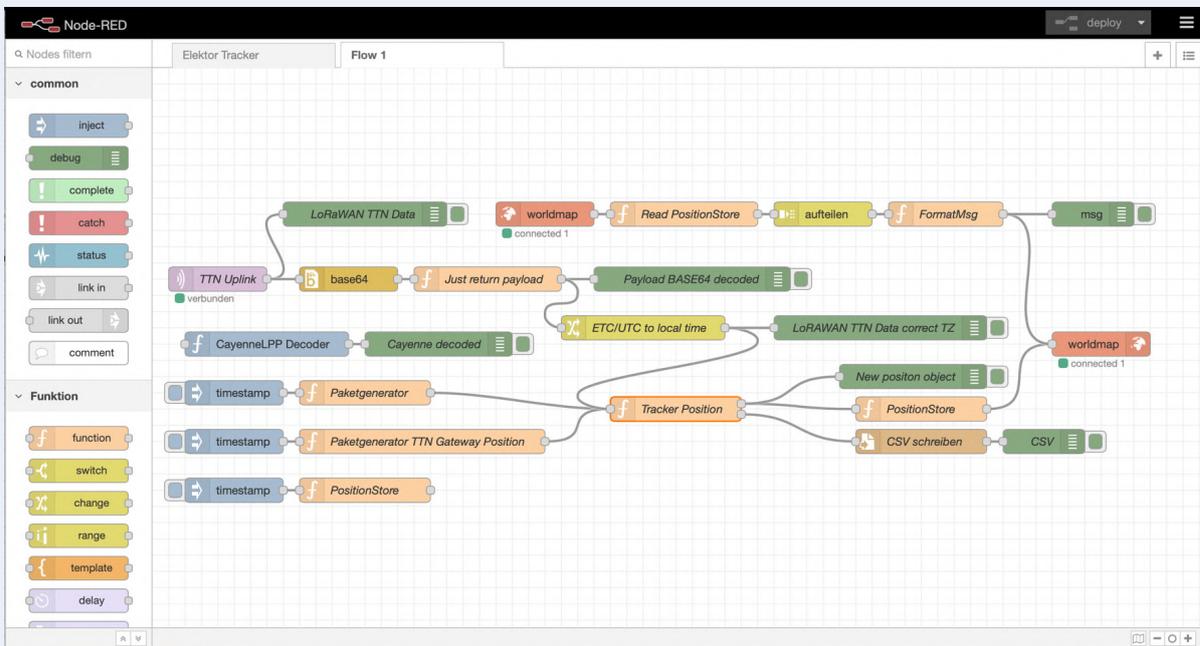


Bild 6. Fertiger Flow in Node-RED.

generiert werden. Der aus der TTN-Konsole kopierte Access Key wird dann unter *Password* eingesetzt und durch Klicken auf *Add* bestätigt. Zurück im Element *mqtt in* kann dem Element noch ein Name gegeben werden (etwa *TTN uplink*); ein Klick auf *Done* schließt die Prozedur ab. **Bild 6** zeigt den angepassten Flow-Chart in Node-RED. Der Cayenne-LPP-Decoder im Flow ist für eine spätere Erweiterung vorgesehen. ◀

210120-02

### Ein Beitrag von

Entwurf und Text: Hans Schneider  
 Redaktion: Mathias Claußen, Jens Nickel  
 Layout: Harmen Heida

### Sie haben Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an den Autor unter [hans.schneider@belga-com.net](mailto:hans.schneider@belga-com.net) oder kontaktieren Sie Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).



### PASSENDE PRODUKTE

- > Elektor LoRa Node – teilbestücktes Modul (SKU 19175)  
[www.elektor.de/19175](http://www.elektor.de/19175)
- > SeeedStudio RFM95 LoRa-Transceiver-Modul (868 MHz) (SKU 18715)  
[www.elektor.de/18715](http://www.elektor.de/18715)
- > Dragino PG1301 LoRaWAN GPS Concentrator für Raspberry Pi (868 MHz) (SKU 19367)  
[www.elektor.de/19367](http://www.elektor.de/19367)

### WEBLINKS

- [1] Mathias Claußen, „LoRa-GPS-Tracker“, ElektorMag 11-12/2020: [www.elektormagazine.de/180166-02](http://www.elektormagazine.de/180166-02)
- [2] Pi-hole: <https://pi-hole.net>
- [3] homebridge.io: <https://homebridge.io>
- [4] flightradar24: [www.flightradar24.com/](http://www.flightradar24.com/)
- [5] „LoRa GPS Tracker“, Elektor-Labs.com: [www.elektormagazine.de/labs/lora-gps-tracker](http://www.elektormagazine.de/labs/lora-gps-tracker)





# Schaltungssimulation mit TINA Design Suite und TINACloud

## Beispiel-Kapitel: Sinusförmige Oszillatoren

Von Dogan Ibrahim (Großbritannien)

Eine der besten Eigenschaften von TINA ist, dass eine simulierte Schaltung mit Auto-Placement- und Auto-Routing-Funktionen einfach auf einer Platine realisiert werden kann. Anwender können auch die Gerber-Plot- und CNC-Bohroptionen von TINA nutzen, um einen Prototyp eines Projekts zu entwerfen und zu implementieren. Das ist zwar großartig, aber bevor eine reale Platine ins Spiel kommen kann, muss die elementare Schaltungssimulation beherrscht werden. Dieses neue Elektor-Buch bietet eine rasante Einführung in TINA, die Sie sich nicht entgehen lassen sollten und die Sie auf Ihrem PC durchführen können.

Ein Sinusoszillator besteht aus einem Verstärker und einem Rückkopplungsnetzwerk (**Bild 1**). Für einen funktionierenden Oszillator müssen die folgenden zwei Bedingungen erfüllt sein:

- Die Schleifenverstärkung ( $A \times B$ ) in Bild 1 muss gleich oder größer eins sein
- Die gesamte Phasenverschiebung in der Schaltung muss 0 oder  $360^\circ$  betragen

In diesem Abschnitt werden einige Simulationen von Oszillatorschaltungen mit Operationsverstärkern vorgestellt.

### Simulation 1 - Oszillator mit Phasenverschiebung

Diese Schaltung wird auch als RC-Oszillator bezeichnet. Jedes RC-Paar führt zu einer Phasenverschiebung von  $60^\circ$ . Drei

Widerstände und drei Kondensatoren werden hier verwendet, um eine Phasenverschiebung von  $180^\circ$  in der Rückkopplungsschleife zu erzeugen.

Die gesamte Phasenverschiebung der Schleife beträgt also  $0^\circ$ , wie für eine Oszillation erforderlich.

### TINA-Schaltplan

**Bild 2** zeigt die Schaltung. Das RC-Netzwerk ist mit dem invertierenden Eingang des Operationsverstärkers verbunden. Unter der Annahme, dass die Widerstände und Kondensatoren gleich sind, muss die Spannungsverstärkung des Verstärkers größer oder gleich 29 sein, das heißt:

$$\text{Gain} = \frac{R_f}{R} \geq 29$$

**Anmerkung der Redaktion.** Dieser Artikel ist ein Auszug aus dem 440-seitigen Buch *Circuit Simulation with TINA Design Suite & TINACloud* (Elektor 2022). Der Auszug wurde so formatiert und leicht bearbeitet, dass er den redaktionellen Standards und dem Seitenlayout von *Elektor Mag* entspricht. Da es sich um einen Auszug aus einer größeren Publikation handelt, können sich einige Begriffe in diesem Artikel auf Diskussionen an anderer Stelle im Buch beziehen. Der Autor und der Herausgeber haben ihr Bestes getan, um solche Fälle auszuschließen, und sind gerne bereit, bei Fragen zu helfen. Kontaktinformationen finden Sie im Kasten Fragen oder Kommentare?

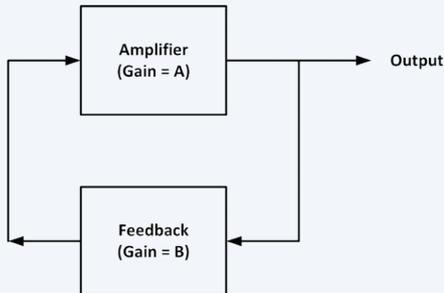


Bild 1. Prinzip eines Oszillators.

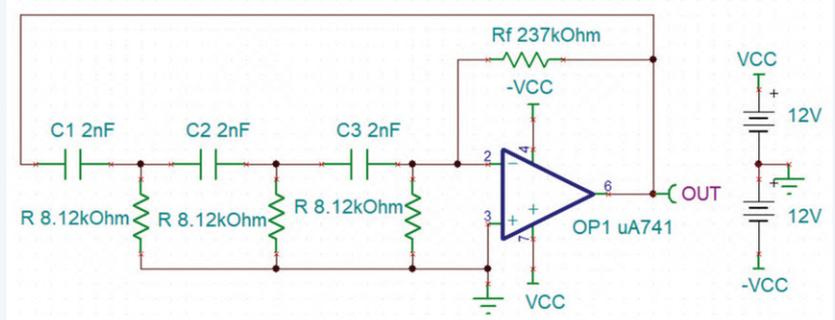


Bild 2. Schaltbild des RC-Oszillators.

Die Frequenz der Schwingung ist gegeben durch:

$$f = \frac{1}{2\pi RC\sqrt{6}}$$

In diesem Beispiel beträgt die gewünschte Frequenz 4 kHz. Wenn wir  $C = 2 \text{ nF}$  wählen, erhalten wir den für  $R$  erforderlichen Wert aus:

$$R = \frac{1}{2\pi f C \sqrt{6}} = \frac{1}{2\pi \times 4 \times 10^3 \times 2 \times 10^{-9} \times \sqrt{6}}$$

Das ergibt  $R = 8,12 \text{ k}\Omega$ . Dann ist

$$\frac{R_f}{R} \geq 29$$

Wir wählen  $R_f = 237 \text{ k}\Omega$ . In diesem Projekt wird ein Operationsverstärker des Typs UA741 verwendet. Die TINA-Schaltung ist als Datei *sim9* verfügbar (siehe Anmerkung am Artikelende).

### TINA-Simulation

Die Schritte zur Ausführung der Simulation sind:

- Klicken Sie auf T&M -> *Oscilloscope* und *Run*. Stellen Sie *Time/div* auf  $100 \text{ u}$  ein.
- Prüfen Sie die Ausgangswellenform (Bild 3). Die Periode beträgt  $250 \text{ }\mu\text{s}$ , was einer Frequenz von  $4 \text{ kHz}$  entspricht. Beachten Sie die Störung (Glitch) in der Ausgangswellenform auf dem Oszilloskop. Das Oszilloskop kann synchronisiert werden, um ein stabiles Bild zu erhalten. Wählen Sie dazu unter *Mode* die Option *Normal* und unter *Source* den Eintrag *synchronizing signal*. Es kann auch erforderlich sein, den (Trigger-) Pegel einzustellen.

### Simulation 2 - Der Wien-Brücken-Oszillator

Einer der einfachsten sinusförmigen Oszillatoren ist der Wien-Brücken-Oszillator, eine zweistufige RC-gekoppelte Schaltung, die bei ihrer Resonanzfrequenz eine gute Stabilität aufweist, wenig verzerrt und sehr einfach abzustimmen ist, was sie zu einer beliebigen Schaltung als Audiofrequenz-Oszillator macht. Die Schaltung verwendet ein in Reihe geschaltetes RC-Glied und

ein paralleles RC-Glied. Die Phasenverschiebung der Schaltung beträgt  $0^\circ$  in der Resonanzfrequenz und, da die Schaltung mit dem positiven Eingang des Operationsverstärkers verbunden ist, beträgt die Gesamtphasenverschiebung  $0^\circ$ . In der Regel wird eine nicht-invertierende Verstärkerkonfiguration verwendet.

Die Bedingung für die Oszillation ist, dass die Spannungsverstärkung gleich oder größer 3 sein muss. Unter der Annahme, dass die gleichen Widerstände und Kondensatoren verwendet werden, ist die Frequenz der Oszillation gegeben durch:

$$f = \frac{1}{2\pi RC}$$

In diesem Beispiel beträgt die erforderliche Frequenz  $5 \text{ kHz}$ . Wenn man  $C = 3 \text{ nF}$  wählt, kann man den erforderlichen Wert von  $R$  wie folgt ermitteln:

$$R = \frac{1}{2\pi f C} = \frac{1}{2\pi \times 5 \times 10^3 \times 3 \times 10^{-9}}$$

Das ergibt  $R = 10,6 \text{ k}\Omega$ .

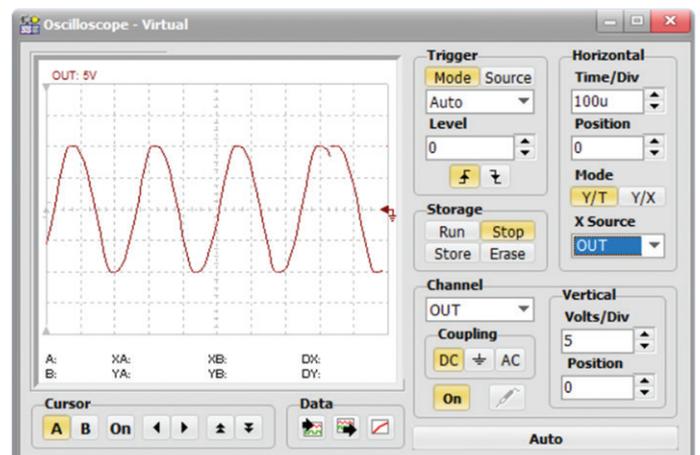


Bild 3. Ausgangswellenform des RC-Oszillators.

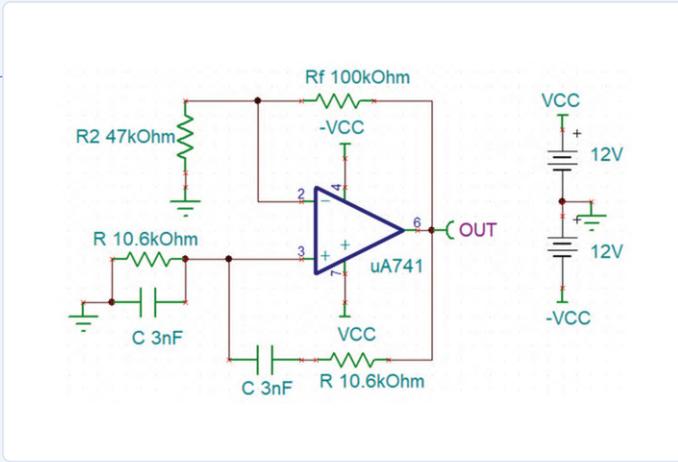


Bild 4. Schaltplan des Wien-Brücken-Oszillators.

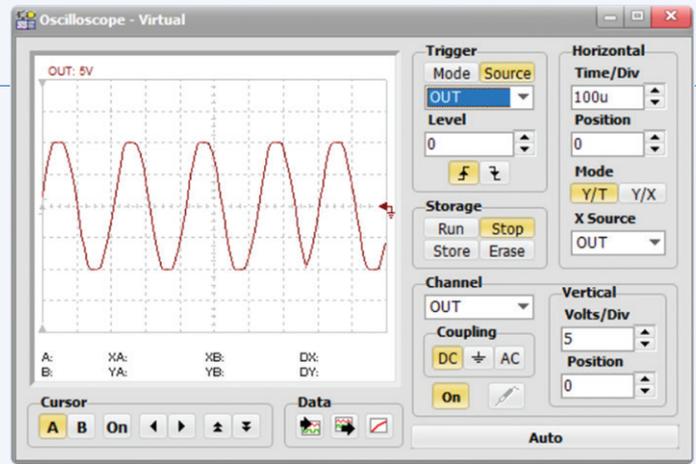


Bild 5. Ausgangssignalform des Wien-Brücken-Oszillators.

Um die Verstärkungsbedingung für einen nicht-invertierenden Verstärker zu erfüllen (siehe **Bild 4**):

$$Gain = 1 + \frac{R_f}{R_2}$$

Wählen Sie  $R_f = 100 \text{ k}\Omega$ :

$$1 + \frac{R_f}{R_2} \geq 3$$

Das ergibt:

$$R_2 = \frac{R_f}{3-1} \leq 50 \text{ k}\Omega$$

Wählen Sie  $R_2 = 47 \text{ k}\Omega$ .

### TINA-Schaltplan

Bild 4 zeigt die Schaltung des Wien-Brücken-Oszillators. Der Rückkopplungskreis ist mit dem nichtinvertierenden Eingang des Operationsverstärkers verbunden und die Verstärkung wird über  $R_f$  und  $R_2$  eingestellt. In diesem Projekt wird ein Operationsverstärker des Typs UA741 verwendet. Die Schaltung ist als Datei *sim10* verfügbar.

### TINA-Simulation

Die Schritte zur Durchführung der Simulation sind:

- Klicken Sie auf T&M -> Oscilloscope und dann auf Run. Stellen Sie Time/div auf 100 u ein.
- Prüfen Sie die Ausgangswellenform (**Bild 5**). Die Periode beträgt 200  $\mu\text{s}$ , was 5 kHz entspricht.

### Simulation 3 - Der Colpitts-Oszillator

Der Colpitts-Oszillator verwendet einen kapazitiven Spannungsteiler als Rückkopplungsquelle. Die beiden Kondensatoren  $C_1$  und  $C_2$  sind über eine gemeinsame Induktivität  $L$  geschaltet, wobei  $C_1$ ,  $C_2$  und  $L$  den abgestimmten Schwingkreis bilden. Der Rückkopplungskreis ist normalerweise mit dem negativen Eingang verbunden.  $C_1$ ,  $C_2$  und  $L$  sorgen für die zusätzliche Phasenverschiebung von  $180^\circ$ , die erforderlich ist, um die Gesamtphasenverschiebung auf  $0^\circ$  zu bringen. Die Frequenz der Schwingung ist gegeben durch:

$$f = \frac{1}{2\pi\sqrt{LC_T}}$$

Dabei ist  $C_T$  die Reihenschaltung von  $C_1$  und  $C_2$ , das heißt:

$$C_T = \frac{C_1 C_2}{C_1 + C_2}$$

Die Stärke der Rückkopplung hängt von den Werten von  $C_1$  und  $C_2$  ab. Durch Ändern der Werte dieser Kondensatoren können wir so die Höhe der Rückkopplungsspannung einstellen, die in den Schwingkreis zurückgeführt wird. Das Verhältnis von  $C_1$  zu  $C_2$  ist das Rückkopplungsverhältnis  $B$ :

$$B = C_1 / C_2$$

Für Schwingungen gilt  $AB \geq 1$ , wobei  $A$  die Verstärkung des Verstärkers ist. Oder

$$A \geq C_2 / C_1$$

In diesem Beispiel beträgt die erforderliche Frequenz 10 kHz. Wenn man  $L = 10 \text{ mH}$  wählt, kann man das Rückkopplungsverhältnis ermitteln:

$$C_T = \frac{1}{4\pi^2 f^2 L} = \frac{1}{4\pi^2 \times 10^8 \times 10 \times 10^{-3}}$$

Das ergibt 25,3 nF.

Wählt man  $C_2 = 250 \text{ nF}$ , so ergibt sich  $C_1 = 28,15 \text{ nF}$ , was einem Rückkopplungsverhältnis von  $B = C_2 / C_1 = 250 / 28,15 = 8,88$  entspricht. Wir können daher die Verstärkung des Verstärkers auf etwa 10 einstellen (siehe **Bild 6**). Dies ergibt:

$$Gain = \frac{R_f}{R_1} = 10$$

Wählt man  $R_1 = 5 \text{ k}\Omega$ , so ergibt sich  $R_f = 50 \text{ k}\Omega$ .

### TINA-Schaltplan

Bild 6 zeigt die Schaltung mit dem Rückkopplungskreis am invertierenden Eingang des Operationsverstärkers. Die Verstärkung wird über  $R_f$  und  $R_1$  eingestellt. In diesem Projekt wird ein Operationsverstärker des Typs UA741 verwendet. Die Schaltung ist als Datei *sim11* verfügbar.

### TINA-Simulation

Die Schritte zur Durchführung der Simulation sind:

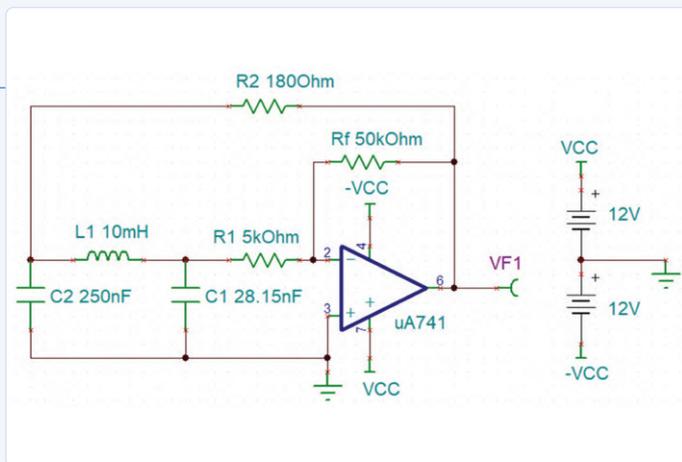


Bild 6. Schaltplan des Colpitts-Oszillators.

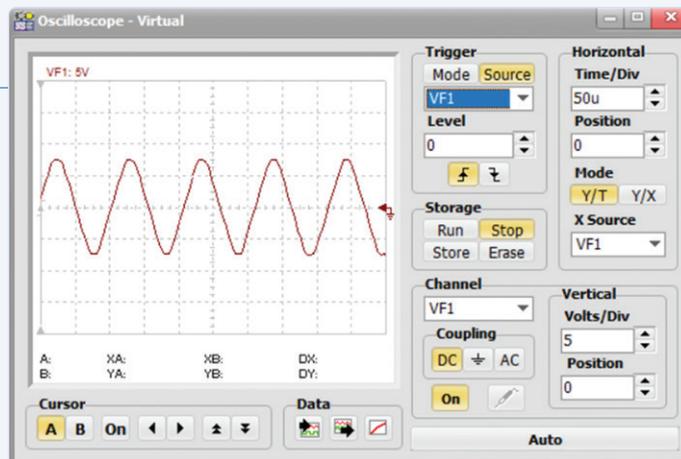


Bild 7. Ausgangssignalform des Colpitts-Oszillators.

- Klicken Sie auf T&M -> Oscilloscope und Run. Stellen Sie die Time/div auf 50 u ein.
- Prüfen Sie die Ausgangswellenform (Bild 7). Die Periode beträgt 100  $\mu$ s (100 u), was 10 kHz entspricht.

### Woher bekommen Sie die Dateien?

Die in diesem Artikel erwähnten TINA-Simulationsdateien sind in dem Softwarepaket enthalten, das der Autor und DesignSoft zur Unterstützung des Buches herausgegeben haben. Die Software steht zum kostenlosen Download zur Verfügung. Gehen Sie zu [1], scrollen Sie nach unten zu Downloads und klicken Sie auf den Dateinamen:

Contents\_Circuit Simulation with TINA Design Suite & TINACloud  
Speichern und entpacken Sie die ZIP-Archivdatei lokal (2,45 MB). Öffnen Sie TINA und laden Sie die Dateien sim9, sim10, sim11 wie in diesem Artikel beschrieben. Sie können die Dateien natürlich für Ihre eigenen Anwendungen abändern.

**Nur für kurze Zeit:** Die ersten 1000 Exemplare des TINA-Buches werden mit einer kostenlosen, einjährigen Lizenz der Basic-Edition der TINA Cloud geliefert. ◀

220025-02

### Ein Beitrag von

Text und Grafiken: Dogan Ibrahim  
Herausgeber: Jan Buiting  
Übersetzung: Rolf Gerstendorf  
Layout: Giel Dols

### 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](mailto:d.ibrahim@btinternet.com) oder an Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).

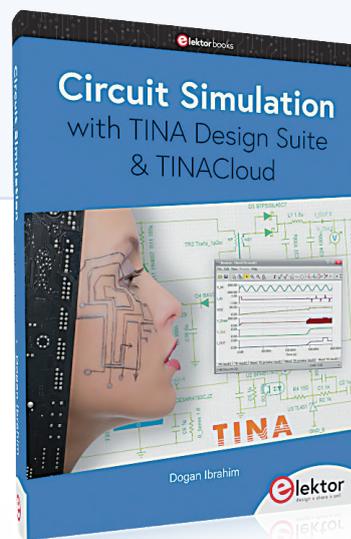


### PASSENDE PRODUKTE

- Dogan Ibrahim, „Circuit Simulation with TINA Design Suite & TINACloud“ inkl. Jahreslizenz der TINACloud (Basic Edition)

Buch: SKU 19977: [www.elektor.de/19977](http://www.elektor.de/19977)

E-Buch: SKU 19978: [www.elektor.de/19978](http://www.elektor.de/19978)



Das Buch enthält eine  
Einjahres-Nutzerlizenz  
für die Basic-Edition der  
TINACloud.

### WEBLINK

[1] Buch-Infoseite: [www.elektor.de/circuit-simulation-with-tina-design-suite-tinacloud](http://www.elektor.de/circuit-simulation-with-tina-design-suite-tinacloud)

# WinUI 3: Neues Grafik-Framework für Windows-Apps

## Eine erste App für Elektroniker

Von **Dr. Veikko Krypczyk**

Software zur Steuerung von unterschiedlichsten Elektronik-Anwendungen läuft häufig unter Windows. Über lokal installierte Desktop-Applikationen hat man einen direkten Zugriff auf alle Systemschnittstellen des PCs. Microsoft vollzieht hier aktuell einen technologischen Umbruch: Die neue Grafikschnittstelle WinUI 3 wird das Maß der Dinge. In diesem Artikel beleuchten wir die technischen Hintergründe und zeigen anhand einer Demo für Elektroniker, wie man das Ganze programmiert.

In vielen Elektronik-Projekten kommt ein Windows-PC zum Einsatz, zur Steuerung, zum Logging von Daten und weiteren Aufgaben. Meist muss dafür Windows-Software mit einer grafischen Benutzeroberfläche individuell programmiert werden.

Lokal ausgeführte Desktop-Applikationen sind hier oft das Mittel der Wahl, denn mit ihnen hat man einen vollständigen Zugriff auf die Systemumgebung und über entsprechende Treiber auch auf die angeschlossene Peripherie.

Microsoft, als Hersteller des Betriebssystems Windows, vollzieht hier gerade einen größeren technologischen Umbruch. Im Fokus stehen die Anbindung und Gestaltung der Benutzeroberflächen. Die Entwicklungen werden unter den Projektnamen *WinUI 3* und *Windows App SDK* koordiniert. In diesem Artikel zeigen wir, was es mit dieser neuen Grafikschnittstelle auf sich hat und wo man diese einsetzen kann. Als Erstes verschaffen wir uns einen Überblick über die Möglichkeiten, Windows-Applikationen mit grafischer Benutzerschnittstelle zu programmieren. Anhand unterschiedlicher Technologien und Anwendungstypen wird dann der Sinn der WinUI 3 deutlich. Natürlich soll es nicht nur bei der Theorie bleiben: Wir experimentieren mit dem neuen Typ einer Windows-Anwendung und erstellen eine erste Applikation.

### Technologien

Grundsätzlich kann man Anwendungen mit grafischer Oberfläche für das Betriebssystem Windows zum heutigen Zeitpunkt in zwei Typen einteilen. Zum einen haben wir *Desktop-Applikationen*. Diese beruhen im Kern auf der Verwendung der *Win32 API*. Für deren Entwicklung gibt es unterschiedliche Ansätze, Frameworks und Programmiersprachen. Aus dem Hause Microsoft stammen die Technologien Windows Forms (WinForms) und Windows Presentation Foundation (WPF). WinForms setzt auf die GDI-Schnittstelle von Windows, WPF basiert

intern auf DirectX und war ursprünglich als Ersatz für WinForms gedacht. Beide Grafikframeworks beruhen auf dem .NET-Framework, das für Windows-Programme vorgesehen war und dessen Weiterentwicklung mit Version 4.8 endet.

Version .NET 5 ist dagegen der technologische Nachfolger von .NET Core. Dieses Framework ist nicht auf Windows beschränkt, sondern kann auch auf anderen Betriebssystemen zum Einsatz kommen. Microsoft hat überraschenderweise sowohl WinForms als auch WPF nach .NET Core transformiert. Wenn Sie heute eine App für Windows erstellen und sich dabei für WinForms oder WPF entscheiden, dann haben Sie also die Wahl zwischen dem bisherigen .NET-Framework und .NET Core. Auch eine Migration von bestehenden Applikationen ist möglich, aber wie so oft bei solchen Vorhaben nicht selten mit einer Reihe von Problemen behaftet. Auch andere Hersteller von Entwicklungswerkzeugen für Windows-Applikationen haben meist auf die Grafikschnittstelle des Betriebssystems (GDI) aufgesetzt und diese in eigenen Frameworks zur Verwendung gekapselt.

Die zweite Kategorie von Windows-Applikationen sind Apps für die *Universelle Windows Plattform (UWP)*. Diese laufen in einem individuell abgeschotteten Bereich des Betriebssystems und haben auch nur einen eingeschränkten Systemzugriff. Anwender installieren diese Apps über den Store. In der Praxis hat sich dieser App-Typ jedoch nicht oder nur sehr zögerlich durchgesetzt. Das liegt unter anderem daran, dass der Systemzugriff aus diesen Apps sehr stark eingeschränkt ist. Die UWP hat jedoch den Vorteil, dass das genutzte Grafikframework WinUI 2 deutlich moderner als die Technologien von WinForms und WPF ist. Ein ansprechendes Design, neue visuelle Komponenten, der Einsatz von Materialien und die Orientierung an der Designsprache *Fluent Design* sind die herausragenden Eigenschaften. Mit anderen Worten: Apps für die UWP wirken modern, zeitgemäß und frisch in der

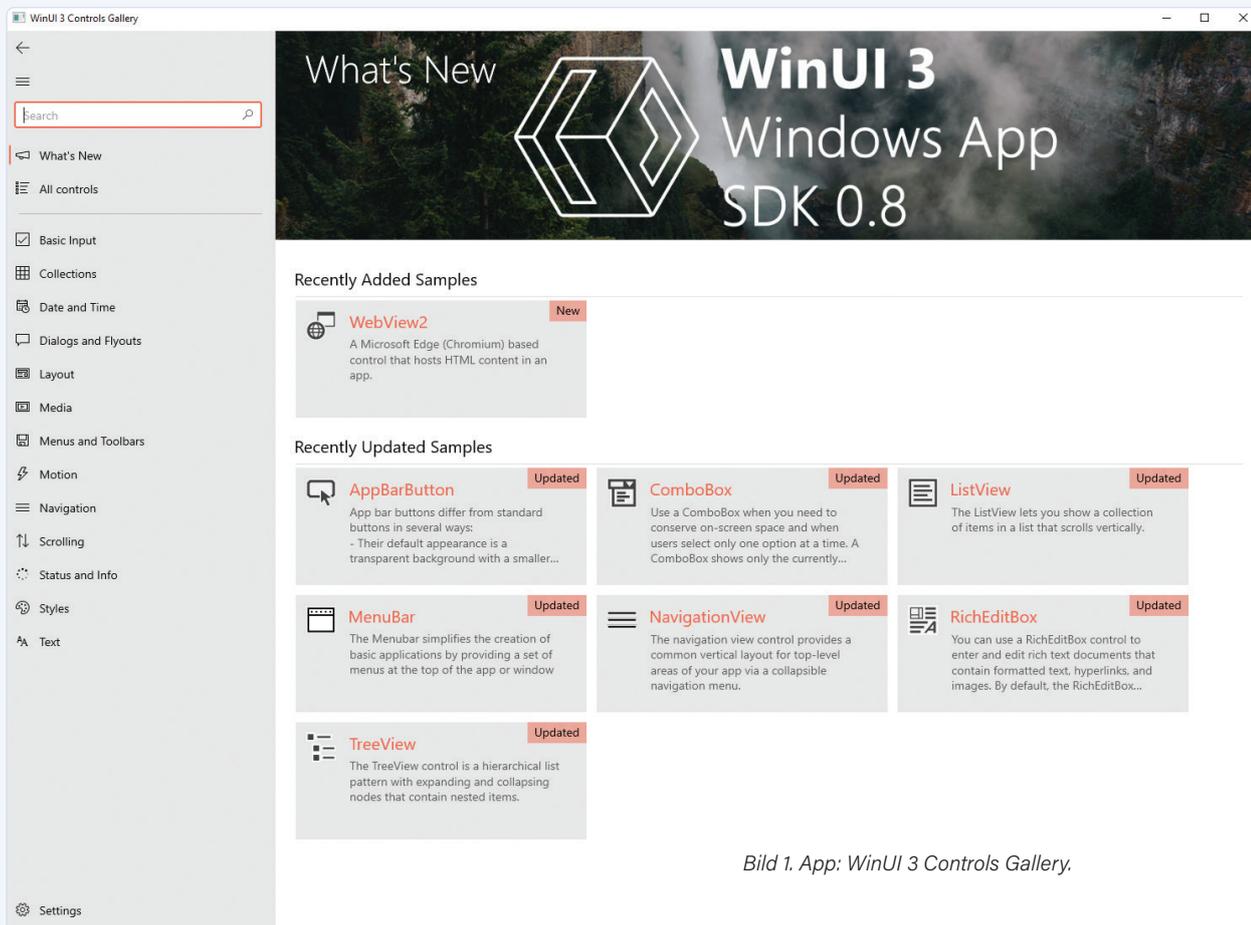


Bild 1. App: WinUI 3 Controls Gallery.

Ansicht, ihre Einsatzoptionen sind jedoch beschränkt. Um ähnliche Wirkungen mit den Technologien WinForms oder WPF zu erreichen, muss man einige Klammzüge machen, extensiv Komponenten von Drittanbietern verwenden oder die Technologien WinForms/WPF mit der UWP „mischen“. Das führt aber schnell zu komplexen App-Strukturen mit den typischen Nachteilen, einer höheren Fehleranfälligkeit und einer schlechteren Wartbarkeit.

Software für elektronische Steuerungen, Erweiterungen und so weiter sind fast ausnahmslos klassische Desktop-Anwendungen, die man auch mit anderen Werkzeugen und Frameworks erstellen kann. Für die Programmiersprache Java gibt es zum Beispiel das Grafikframework Swing, welches für Windows intern jedoch wiederum auf der Betriebssystemschnittstelle GDI basiert.

### Das Grafikframework WinUI 3

Mit der Einführung der Grafikschnittstelle WinUI 3 möchte Microsoft allen Anwendungen unter Windows die Verwendung einer zeitgemäßen Grafikschnittstelle ermöglichen. WinUI 3 ist der technologische Nachfolger von WinUI 2 [1]. Sie steht jedoch für alle Arten von Windows-Anwendungen zur Verfügung und ist damit nicht auf den Einsatz in Apps für die UWP beschränkt. WinUI 3 ist Bestandteil des neuen Windows App SDK, welches auch parallel mit der Einführung von Windows 11 bereitgestellt wird. Das Windows App SDK bündelt neue Features für die Entwicklung von Windows-Anwendungen. Es zielt dabei nicht nur auf Windows 11 ab, sondern kann auch unter den aktuellen Versionen von Windows 10 verwendet werden. Das Windows App SDK befindet sich noch in der laufenden Entwicklung. Doch es steht eine erste Version zur Verfügung, welche man bereits in neu erstellten Anwendungen einsetzen kann.

WinUI 3 basiert technisch und konzeptionell auf WinUI 2. Wenn Sie

schon mal eine App für die UWP entwickelt haben, dann werden Sie schnell zurecht kommen. Es wird auf die folgenden Prinzipien gesetzt:

- **Trennung von Code und Design:** Die Benutzeroberfläche wird deklarativ mit Hilfe der XML-basierten Sprache XAML in separaten Dateien erstellt.
- **Steuerelemente für die Gestaltung:** Es steht eine Palette von Steuerelementen zur Gestaltung der Oberfläche zur Verfügung. Darunter sind Basiselemente wie Buttons und Texteingabefelder, aber auch komplexere und modernere Elemente, wie ein Kalendersteuerelement, eine WebView oder ein Element zur Anzeige von Personendaten, wie wir es zum Beispiel bei einer Nutzerverwaltung benötigen. Sie möchten einen Überblick über die verfügbaren Steuerelemente bekommen? Dann wechseln Sie in den Microsoft Store. Laden Sie dort die App *WinUI 3 Controls Gallery* herunter. Bei dieser App handelt es sich um eine Vorschau der verfügbaren Steuerelemente für WinUI 3. Es wird deren Verwendung (Use Case) und die Einbindung im Quellcode (XAML) demonstriert. Ebenso findet man entsprechende Links zur Dokumentation (**Bild 1**).
- **Lose Kopplung durch DataBinding:** Die Eigenschaften und die Ereignisse der Steuerelemente werden mittels Datenbindung an den Quellcode gebunden. Auf diese Weise werden Daten zwischen dem Programmcode und den Steuerelementen der Benutzeroberfläche in beiden Richtungen ausgetauscht. Ebenso werden Ereignisse der Steuerelemente wie der Klick auf einen Button auf gleiche Art und Weise an den betreffenden Algorithmus weitergeleitet.
- **Modernes Design:** Die WinUI 3 setzt auf aktuelle Designtrends. Dazu gehören die Verwendung der Designsprache *Fluent* und

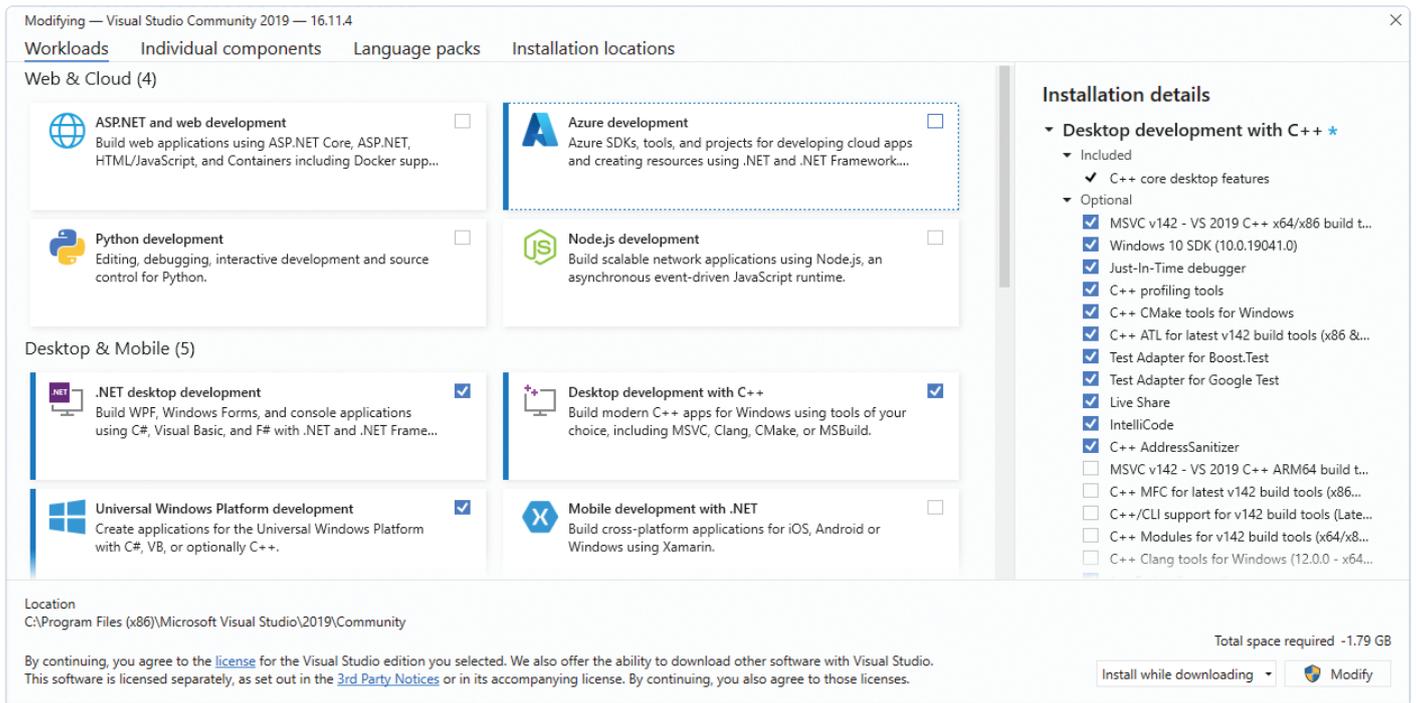


Bild 2. Auswahl der notwendigen Workloads für Visual Studio.

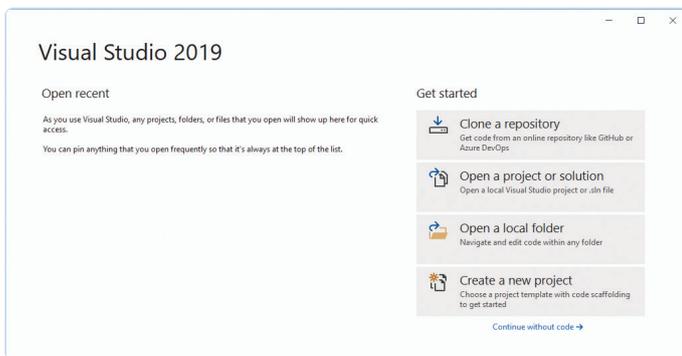


Bild 3. Erster Start von Visual Studio.

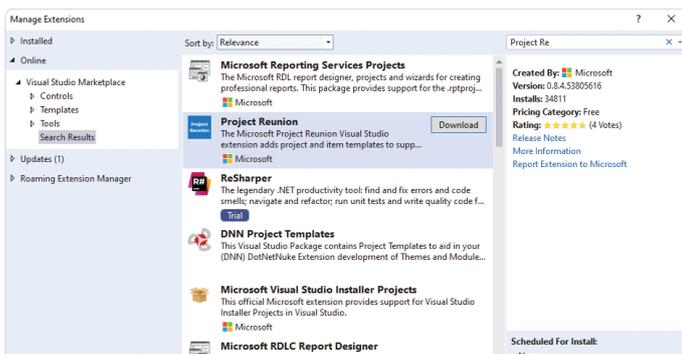


Bild 4. Installation des Windows App SDK (Projekt „Reunion“).

eine Nutzung des Materials *Mica*. Die Designsprache Fluent setzt auf die folgenden Elemente: bewusster Einsatz von Geometrie und Farbe, Überlappungen von Oberflächen, Nutzung ausgewählter Materialien und der Einsatz von spezifischer Ikonografie und Typographie zur visuellen Gestaltung mittels Bildern, Symbolen und Schriftarten. Ebenso werden Bewegungen zur Unterstützung des Benutzerfeedbacks verwendet.

Sehen wir uns jetzt die Vorgehensweise zur Anwendungsentwicklung mit der WinUI 3 an. Zuvor müssen wir jedoch die Entwicklungsumgebung einrichten.

## Entwicklungsumgebung und Setup

Als Entwicklungsumgebung kommt die aktuelle Version von Visual Studio 2019 [2] zum Einsatz. Es genügt die Community Edition. Wenn Sie diesen Artikel lesen, gibt es gegebenenfalls schon eine stabile Version von Visual Studio 2022, die Sie dann auch verwenden können. Zuvor empfiehlt es sich, aktuelle Updates für das Betriebssystem zu installieren. Während der Installation von Visual Studio werden Sie zur Auswahl der Installationspakete aufgefordert. Den *Visual Studio Installer* können Sie über das Startmenü auch zu einem späteren Zeitpunkt jederzeit aufrufen. Wählen Sie die folgenden Installationspakete (Workloads): *.NET desktop development*, *Desktop development with C++* und *Universal Windows Platform development* (Bild 2). Nach der Installation starten Sie Visual Studio und im Startbildschirm wählen Sie den Eintrag *Continue without code* -> (Bild 3).

Wir müssen nun zusätzlich das Template für die Entwicklung mit WinUI 3 installieren. In Visual Studio gehen Sie dazu zum Menüpunkt *Extensions | Manage Extensions*. Suchen Sie hier nach *Project Reunion*

(der Entwicklungsname des neuen Windows App SDK) und installieren Sie die aktuelle Version (**Bild 4**). Installieren Sie auf gleiche Weise auch die Erweiterung *Windows Template Studio*. Diese bietet erweiterte Vorlagen zum Erstellen einer neuen Anwendung. Visual Studio muss nach dem Download der Erweiterungen neu gestartet werden, dann erfolgt die Installation automatisch.

## Eine App für WinUI 3

Starten wir nun mit einem neuen Projekt. Als Vorlage (**Bild 5**) wählen wir *App WinUI 3 in Desktop (Windows Template Studio)*. Im Windows Template Studio (**Bild 6**) nehmen wir Voreinstellungen für das Projekt vor:

- **Projekttyp:** Festgelegt wird die Art der Navigation, zum Beispiel mit Menüleiste oder seitlicher Navigationsleiste (Hamburger-Menü).
- **Design-Pattern:** Direkte Installation und Konfiguration des MVVM-Toolkits. Das dient dazu, die Elemente des User Interfaces (definiert in XAML) an die Programmlogik (Programmiersprache C#) zu koppeln.
- **Seiten:** Wir können dem Projekt einige Seiten hinzufügen. Dabei haben wir die Auswahl aus unterschiedlichen Vorlagen, zum Beispiel eine Seite für Programmeinstellungen.
- **Ergänzende Features:** Hier lassen sich zum Beispiel Themes auswählen oder Programmeinstellungen speichern.

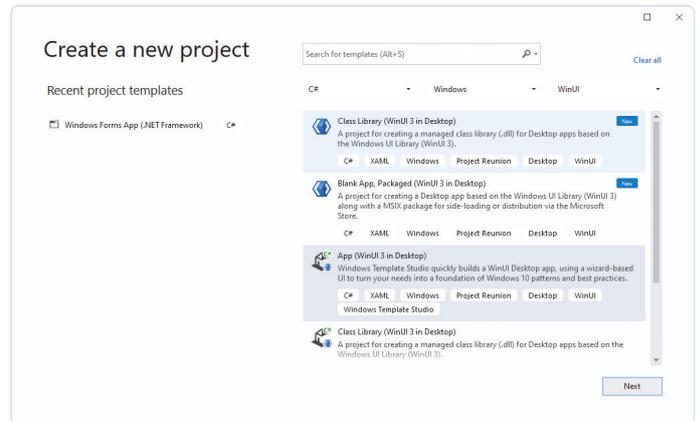


Bild 5. Projektvorlage WinUI 3 Desktop.

Mit einem Klick auf den Button *Create* erzeugen Sie die Desktop-Anwendung, welche WinUI 3 nutzt. Windows Template Studio generiert eine Projektmappe mit drei Projekten:

- **App:** Dieses enthält den Quellcode zur Desktop-Anwendung. Im Unterordner *View* finden Sie zum Beispiel die XAML-Dateien

Anzeige



By CRC Industries

# NEUES AUSSEHEN GLEICHE LEISTUNG



**KONTAKT  
REINIGER**

**ELEKTRONIK UND  
PRÄZISIONSREINIGER**

**DRUCKGASREINIGER**

**WEITERE  
REINIGER**

**BESCHICHTUNGEN  
UND SCHUTZLACKE**

**SCHMIERSTOFFE**

**SPEZIAL  
PRODUKTE**

**KÄLTESPRAYS**

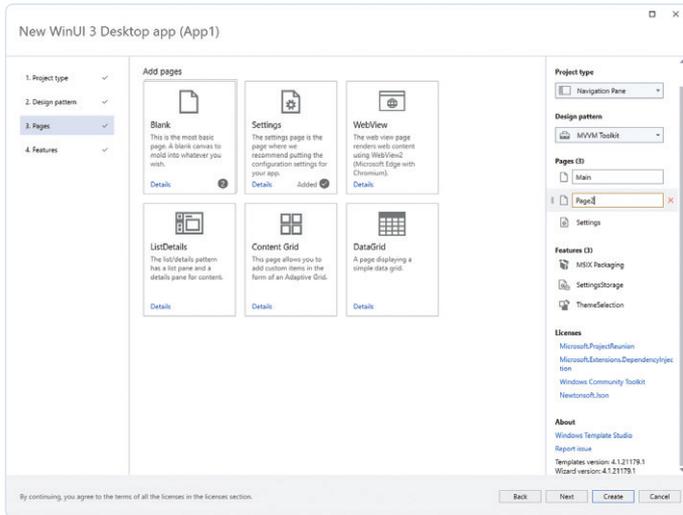


Bild 6. Windows Template Studio.

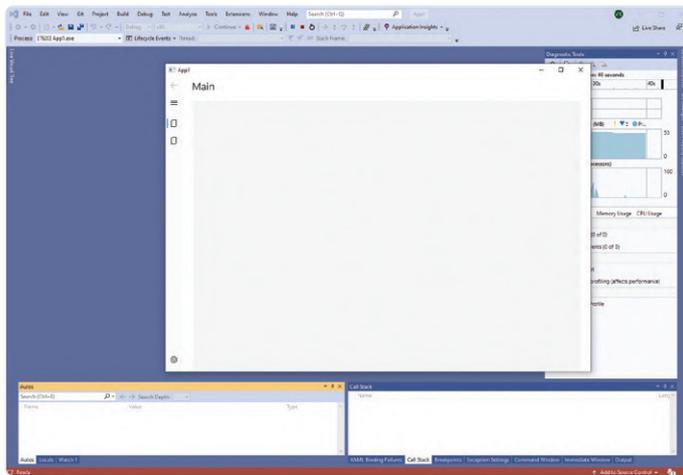


Bild 7. Eine erste Desktop-Anwendung mit WinUI 3.

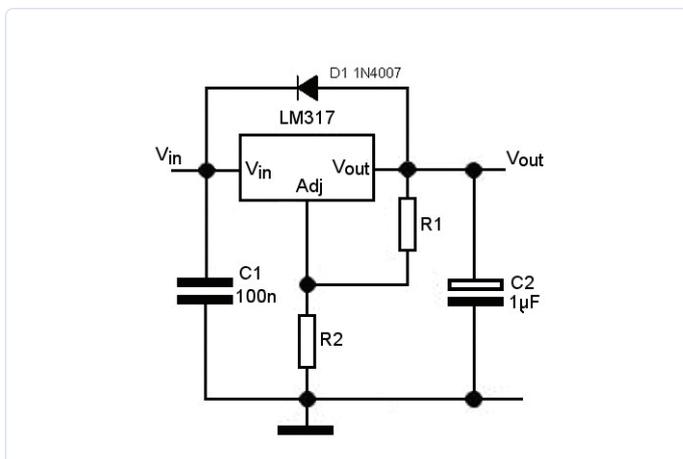


Bild 8. Schaltung mit Spannungsregler LM317.

für die Seiten, welche vom Windows Template Studio angelegt wurden. Die Programmlogik wird in Dateien des Programmordners *ViewModel* abgelegt.

- **Package:** Das Projekt ist für die Bereitstellung der Desktop-Applikation verantwortlich. Im Moment werden WinUI 3-Anwendungen über ein App-Package auf dem Zielrechner installiert. Diese Form wird bisher für Apps der UWP angewendet. Die generierten Packages können auch über den Store verteilt werden. Künftige Versionen des Windows App SDKs sollen auch eine Installation ohne App-Package ermöglichen.
- **Core:** Dieses Projekt enthält die Sammlung von Services und Klassen, welche Dienste für die App zur Verfügung stellen. Dieses Projekt ist nicht zwingend und kann auch weggelassen werden.

Starten Sie die Anwendung direkt aus Visual Studio über den grünen Pfeil in der Symbolleiste. Herzlichen Glückwunsch. Sie haben Ihre erste Anwendung mit WinUI 3 erstellt (**Bild 7**). Dabei weisen wir nochmals darauf hin: Es handelt sich um eine Desktop-Anwendung mit einem vollständigen Systemzugriff. Das ist wie gesagt wichtig für Software, die zum Beispiel externe Elektronik steuert. Eine seitliche Navigation, erste Seiten und die Möglichkeit, das Design der Anwendung anzupassen, sind bereits vorhanden. Sie haben alle Möglichkeiten für einen Zugriff auf das System, inklusive der Kommunikation mit den Systembibliotheken und Treibern. Jetzt wollen wir mit der Gestaltung des User Interfaces experimentieren.

## Demo-Applikation

Am besten wird man mit einem neuen System vertraut, wenn man damit arbeitet. Also gestalten wir eine Benutzeroberfläche einer ersten Anwendung (den Quellcode zum Beispiel finden Sie auf der Webseite zum Artikel [3]). Ausgangspunkt ist die XAML-Datei der betreffenden Seite. Als Experiment wollen wir ein Rechentool für den einstellbaren Spannungsregler LM317 erstellen (**Bild 8**). Die Berechnung der Ausgangsspannung folgt der bekannten Formel  $U_{\text{out}} = 1,25 (1 + R2 / R1)$ . Wir können diese Gleichung nach  $R2$  auflösen und damit den Wert für eine gewünschte Spannung errechnen. Mit Hilfe dieses Beispiels können wir die Vorgehensweise bei der Programmierung von Anwendungen mit WinUI 3 demonstrieren. Das umfasst folgende Schritte:

- Definition des User Interfaces in XAML.
- Kodieren der Programmlogik in C#.
- Bindung des User Interfaces an die Programmlogik.
- Weiterleiten der Nutzerinteraktion vom User Interface an die Programmlogik.
- Datenausgabe auf dem Formular.

Beginnen wir mit der Definition der Oberfläche. Wir benötigen zwei Textfelder für die Erfassung der Werte von  $R1$  und  $U_{\text{out}}$ . Ebenso benötigen wir ein Textfeld für den Wert von  $R2$ . Für das Auslösen der Berechnung wird ein *Button* eingesetzt. Für die Ein- und Ausgabe verwenden wir Steuerelemente des Typs *TextBox*. Alle Elemente sollen untereinander platziert werden, daher werden diese in einen Layout-Container des Typs `<StackPanel />` eingefügt. Ohne weitergehende Konfiguration werden alle Elemente innerhalb eines *StackPanel*s untereinander angeordnet. Die Steuerelemente werden über den XAML-Code konfiguriert, mit den Eigenschaften gemäß **Tabelle 1**.

**Tabelle 1. Steuerelemente.**

Steuerelement	Eigenschaft	Wert	Bedeutung
TextBoxR1	Width	200	Breite
	Margin	20, 20, 0, 0	Abstand nach links, oben, rechts, unten
	HorizontalAlignment	Left	Ausrichtung
	Header	R1:	Beschriftung
	Text	x:Bind ViewModel.R1	Bindung der Eigenschaft an die Variable R1 im C#-Code
TextBoxU (out)	Width	200	Breite
	Margin	20, 10	Abstand nach links und rechts, oben und unten
	HorizontalAlignment	Left	Ausrichtung
	Header	U (out):	Beschriftung
	Text	x:Bind ViewModel.UOut	Bindung der Eigenschaft an die Variable UOut im C#-Code
TextBoxR2	Width	200	Breite
	Background	LightGray	Hintergrundfarbe
	Margin	20, 10	Abstand nach links und rechts, oben und unten
	HorizontalAlignment	Left	Ausrichtung
	Header	R2:	Beschriftung
	IsReadOnly	True	Schreibschutz
	Text	x:Bind ViewModel.R2	Bindung der Eigenschaft an die Variable R2 im C#-Code
Button	Width	200	Breite
	Margin	20, 10	Abstand nach links und rechts, oben und unten
	Background	LightGreen	Hintergrundfarbe
	Command	x:Bind ViewModel.CalcCommand	Bindung an die Methode CalcCommand im C#-Code
	Content	Calc	Beschriftung
	FontWeight	Bold	Schriftstärke

Anzeige



Mit Schnelligkeit und Ausdauer zum optimalen Ergebnis!

Besuchen Sie uns auf der **SMTconnect** in Nürnberg  
10. - 12.05.2022  
Halle 4, Stand 100



Almit GmbH Unterer Hammer 3 64720 Michelstadt +49 (0) 6061 96925 0 www.almit.de info@almit.de



### Listing 1. Definition des User Interfaces.

```
<Page
  x:Class="App1.Views.MainPage"
  ...>

  <Grid x:Name="ContentArea" Margin="">
    <StackPanel Background="">
      <TextBox
        Width="200"
        Margin="20,20,0,0"
        HorizontalAlignment="Left"
        Header="R1:"
        Text="" />
      <TextBox
        Width="200"
        Margin="20,10"
        HorizontalAlignment="Left"
        Header="U (out):"
        Text="" />
      <Button
        Width="200"
        Margin="20,10"
        Background="LightGreen"
        Command=""
        Content="Calc"
        FontWeight="Bold" />
      <TextBox
        Width="200"
        Margin="20,10"
        HorizontalAlignment="Left"
        Background="LightGray"
        Header="R2:"
        IsReadOnly="True"
        Text="" />
    </StackPanel>
  </Grid>
</Page>
```

Den zugehörigen Quellcode sehen Sie in **Listing 1**. Sie können die Codierung der Oberfläche interaktiv vornehmen. Starten Sie die Anwendung und platzieren Sie die betreffende XAML-Datei in Visual Studio und die Anwendung nebeneinander auf dem Bildschirm (**Bild 9**). Änderungen im XAML-Code werden unmittelbar bei gestarteter Anwendung – ohne Speicherung – direkt übernommen und führen zu einer Aktualisierung der Anzeige. Das Feature heißt *Hot Reload* und ist Standard beim Erstellen von grafischen Benutzeroberflächen.

Interessant ist die Bindung der Eigenschaften mit der Bezeichnung *Text* der Steuerelemente des Typs *TextBox*. Hier finden Sie im XAML-Code einen Ausdruck nach dem Muster

```
Text="{x:Bind ViewModel.R1, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"
```

Das bedeutet, dass die Eigenschaft *Text* an die Variable *R1* gebunden wird. Diese wird im *ViewModel* der Seite definiert. Zugrunde liegt das MVVM-Konzept. Die Oberfläche ist die *View* und die Daten werden

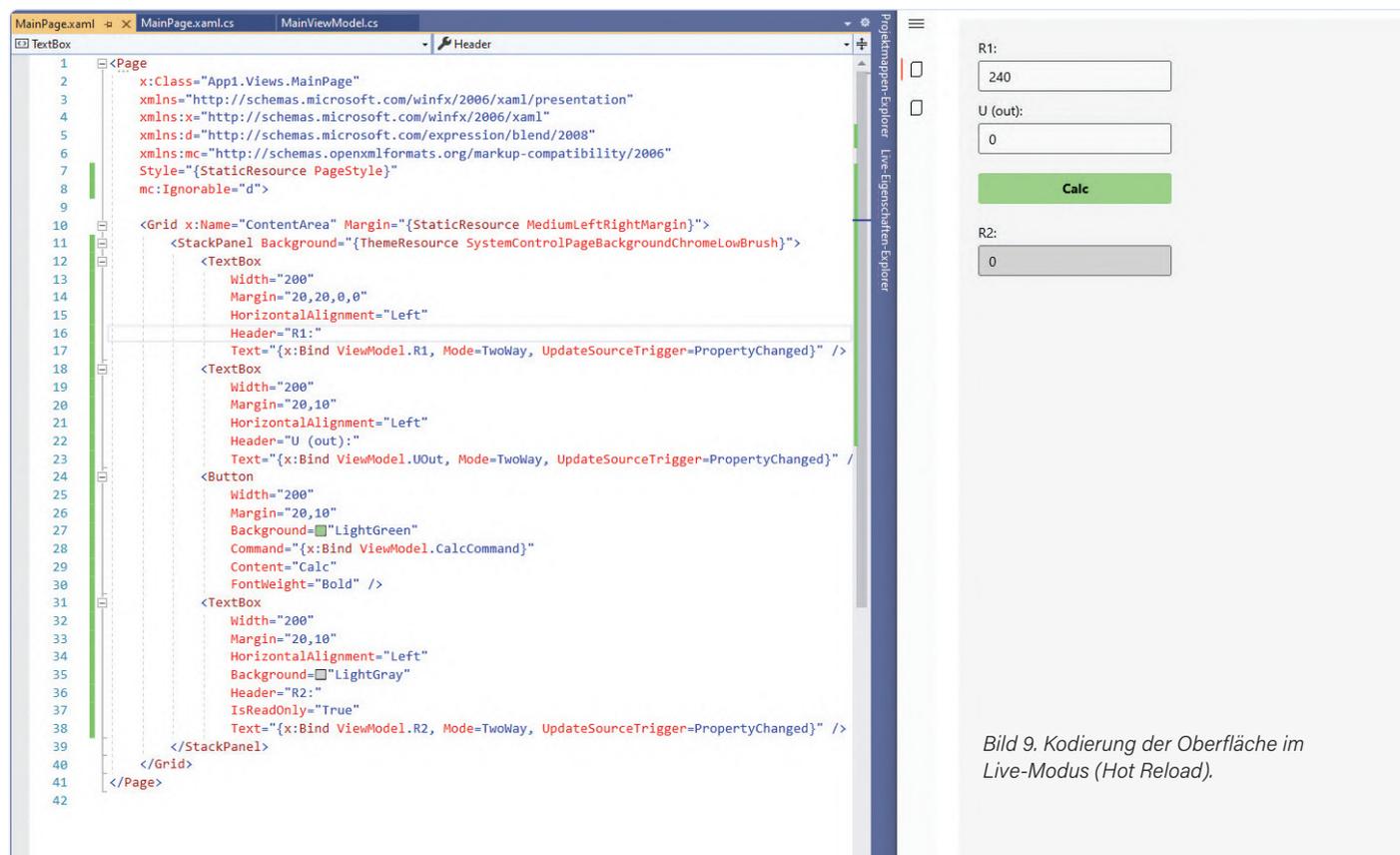


Bild 9. Kodierung der Oberfläche im Live-Modus (Hot Reload).



## Listing 2. Programmlogik für den Rechner in C#.

```
using CommunityToolkit.Mvvm.ComponentModel;
using CommunityToolkit.Mvvm.Input;

namespace Appl.ViewModels
{
    public class MainViewModel : ObservableRecipient
    {
        public double R1 { get; set; } = 240;
        public double UOut { get; set; }

        private double r2;
        public double R2
        {
            get
            {
                return r2;
            }
            set

```

```

        {
            SetProperty(ref r2, value);
        }
    }

    public RelayCommand CalcCommand;
    public MainViewModel()
    {
        CalcCommand = new RelayCommand(CalcCommandExecute);
    }

    private void CalcCommandExecute()
    {
        R2 = (UOut / 1.25 - 1) * R1;
    }
}
}

```

im *Model* verwaltet. Die Verbindung zwischen beiden Schichten stellt das *ViewModel* dar. Durch das MVVM-Konzept werden alle Schichten entkoppelt und lassen sich unabhängig voneinander entwickeln und warten. Informationen zum MVVM-Pattern finden Sie unter [4].

## Programmlogik

Kommen wir zur Programmlogik, die in C# programmiert wird (Listing 2). Dazu wird zu jedem Fenster der Oberfläche (*View*) eine Programmdatei (*ViewModel*) zugeordnet. In unserem Beispiel ist es die Datei *MainViewModel*, welche der View *MainPage* zugeordnet

Anzeige

# JETZT ABONNIEREN UND SIE ERHALTEN



## Ihre Vorteile:

- 6 gedruckte Ausgaben der MagPi
- ein Willkommensgeschenk
- Zugang zu dem Online Archiv mit allen MagPi Ausgaben seit 2016
- Günstiger als Einzelkauf
- Jede Ausgabe direkt zu Ihnen nach Hause
- Alle Ausgaben auch als PDF
- Willkommensgeschenk mit tollen Inhalten

JETZT ABONNIEREN: [WWW.MAGPI.DE](http://WWW.MAGPI.DE)

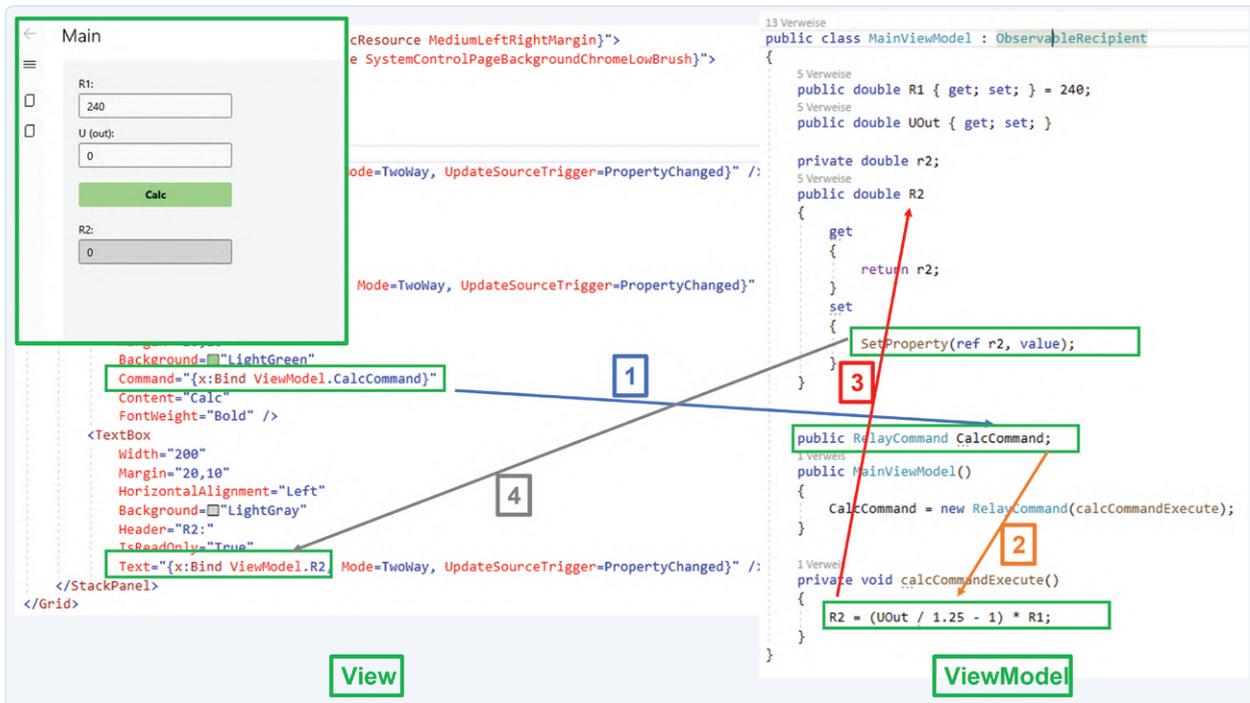


Bild 10. Zusammenhang zwischen View und ViewModel am Beispiel.

wird. Zum Programmcode sind ein paar Erläuterungen angebracht:

- **Import von Bibliotheken:** Das geschieht über die *uses*-Anweisung. In diesem Fall sind es zwei Bibliotheken für das MVVM-Pattern.
- **Definition von Eigenschaften:** Diese müssen öffentlich (*public*) sein, da wir von außen, in diesem Fall von der View, auf die Eigenschaften zugreifen.
- **Automatische Aktualisierung der Oberfläche:** Die Klasse *MainViewModel* leitet von der Basisklasse *ObservableRecipient* ab, welche durch den Projektassistenten beim Erstellen des Projektes angelegt wurde. Diese Klasse implementiert ihrerseits das so genannte *OnPropertyChanged*-Event, das dafür sorgt, dass bei Änderung eines Wertes einer Eigenschaft gebundene Elemente über die Änderung benachrichtigt werden. In unserem Fall ist die Eigenschaft *R2* von Interesse. Der Wert von *R2* wird im Programmcode errechnet. Dabei werden der so genannte *Setter* der Eigenschaft aufgerufen und über die Methode *SetProperty(...)* das eben beschriebene *OnPropertyChanged*-Ereignis ausgelöst. An die Eigenschaft *R2* (im ViewModel) ist wiederum die *Text*-Eigenschaft der *TextBox R2* gebunden. Bei Änderung von *R2* wird der dargestellte Wert in der zugehörigen *TextBox* automatisch aktualisiert. Das funktioniert dank Datenbindung.
- **Benutzeraktionen mittels Commands weiterleiten:** Drückt der Anwender auf den Button, dann wird ein *Command* ausgelöst. An dieses *Command* ist die Methode zur Berechnung gebunden. Auch hier sind Benutzeroberfläche und Programmcode nur über die Datenbindung miteinander gekoppelt.
- **Zuordnung von View und ViewModel:** Der Programmcode

(Datei: *MainViewModel.cs*) wird der Oberfläche (Datei: *MainPage.xaml*) zugeordnet. Das geschieht in der so genannten Code-behind-Datei der Seite (Datei: *MainPage.xaml.cs*). Sie können sich das im Quellcode ansehen.

- **Berechnung:** Die Berechnung des Wertes von *R2* erfolgt in der Methode *calcCommandExecute(...)* nach oben genannter Formel, welche nach *R2* umgestellt wird.

Die Oberfläche ist also mittels Datenbindung „lose“ an den Programmcode gebunden. Die eben beschriebenen Zusammenhänge zur Datenbindung sind in **Bild 10** am Beispiel visualisiert. Starten Sie die Anwendung und probieren Sie diese aus. Nach Eingabe von *R1* und *U<sub>Out</sub>* erfolgt die Berechnung des Wertes für den zweiten Widerstand *R2* (**Bild 11**).

Damit haben wir das grundsätzliche Entwicklungsmodell für Desktop-Anwendungen mit dem Grafikframework WinUI 3 beschrieben. Aus heutiger Sicht wird es unter Windows zu einem neuen Standard werden und auch durch andere Entwicklungsumgebungen und -sprachen nutzbar sein. Die Vielfalt der grafischen Optionen, um moderne Anwendungen zu erstellen, ist beeindruckend.

### Fazit und Ausblick

Sie haben die Möglichkeit, aus einer Vielzahl von Technologien zu wählen, um eine Anwendung für das Betriebssystem Windows zu erstellen. Der Trend – auch mit Blick auf Windows 11 – geht in Richtung des Einsatzes von WinUI 3. Damit können Desktop Anwendungen mit einem ansprechenden und modernen User Interface realisiert werden. Aus dieser Perspektive lohnt es sich, bei der Neuentwicklung einer Windows-Anwendung auf die WinUI 3 zu setzen, und für eine bestehende Anwendung eine Möglichkeit der Migration zu prüfen.

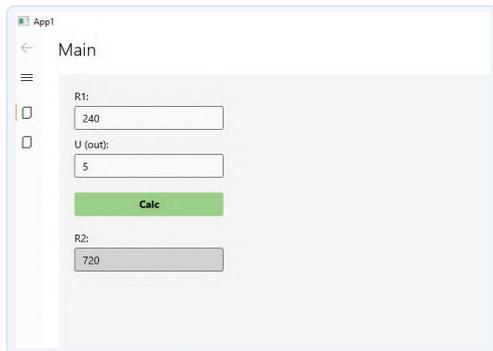


Bild 11. Die fertige Beispielanwendung.

Das Ergebnis sind Applikationen mit einer zeitgemäßen Oberfläche und einer guten User Experience. Einschränkungen beim Systemzugriff – wie bei der UWP – gibt es nicht. ◀

210407-02

### Ein Beitrag von

Text und Bilder: Dr. Veikko Krypczyk

Redaktion: Jens Nickel

Layout: Giel Dols

### Sie haben Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare? Bitte wenden Sie sich per E-Mail an die Elektor-Redaktion unter [redaktion@elektor.de](mailto:redaktion@elektor.de).

## WEBLINKS

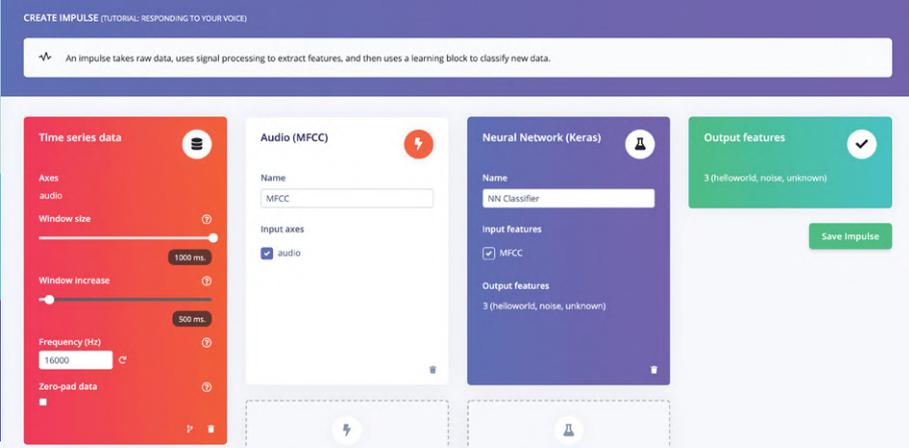
- [1] Infos zu WinUI 3: <https://docs.microsoft.com/en-us/windows/apps/winui/>
- [2] Visual Studio 2019: <https://visualstudio.microsoft.com/de/vs/>
- [3] Projektseite zu diesem Beitrag: <http://www.elektormagazine.de/210407-02>
- [4] Informationen zum MVVM-Pattern: [https://de.wikipedia.org/wiki/Model\\_View\\_ViewModel](https://de.wikipedia.org/wiki/Model_View_ViewModel)

Anzeige


EDGE IMPULSE

# Advanced ML for every solution

Try free at [edgeimpulse.com](https://edgeimpulse.com)



# GUIs mit Python:

## Die schlechteste GUI der Welt



**MAKER** Laura Sach

Laura leitet das A-Level-Team bei der Raspberry Pi Foundation und erstellt Ressourcen für Schüler, die etwas über Informatik lernen möchten.

@CodeBoom



**MAKER** Martin O'Hanlon

Martin arbeitet im Lernteam der Raspberry Pi Foundation, wo er Onlinekurse, Projekte und Lernressourcen erstellt.

@martinohanlon

Durch Fehler lernen.

**E**s ist nun an der Zeit, mit verschiedenen Widgets, Farben, Schriftarten und Funktionen zu experimentieren. Hier ein paar Beispiele dafür, wie Sie es auf keinen Fall machen sollten.

### Schlechte Lesbarkeit

Die richtige Wahl des Kontrastes zwischen Hintergrund und Schriftfarbe ist bei der Gestaltung einer grafischen Benutzeroberfläche sehr wichtig. Importieren Sie die Widgets am Anfang des Codes:

```
from guizero import App, Text
```

Erstellen Sie eine App mit einem Titel:

```
app = App("it's all gone wrong")
title = Text(app, text="Some hard to read text")
```

```
app.display()
```

Experimentieren Sie, indem Sie die Farben, die Schriftart und die Textgröße ändern (siehe Listing von **worst1.py**).

```
app = App("it's all gone wrong", bg="dark green")
title = Text(app, text="Some hard-to-read text", size="14", font="Comic Sans", color="green")
```

### Blinkender Text

Es ist wichtig, dass der Text auf einer GUI auch lange genug sichtbar bleibt, um gelesen zu werden. Er sollte auf keinen Fall verschwinden oder anfangen zu blinken.

Alle Widgets in guizero können mit den Funktionen `hide()` und `show()` unsichtbar (oder wieder sichtbar) gemacht werden. Wenn Sie die `repeat`-Funktion in guizero verwenden, um eine Funktion im Sekundentakt auszuführen, können Sie Ihren Text verstecken oder ihn blinken lassen. Ein Beispiel:

```
def flash_text():
    if title.visible:
        title.hide()
    else:
        title.show()
```

Sorgen Sie mit `repeat` dafür, dass die Funktion `flash_text` alle 1000 Millisekunden (1 Sekunde) aufgerufen wird.

```
app.repeat(1000, flash_text)
```

```
app.display()
```

Ihr Code sollte nun wie **worst2.py** aussehen. Der Titeltext sollte im Sekundentakt auftauchen und wieder verschwinden.

### Unpassende Eingabe-Elemente

Die Verwendung eines passenden Eingabe-Elementes, also Widgets, kann darüber entscheiden, ob eine GUI brauchbar oder unbrauchbar ist.

Welches Widget würden Sie zum Beispiel verwenden, um ein Datum einzugeben? Eine einzige `TextBox`? Mehrere `Combo`-Boxen? Eine `TextBox` wäre flexibler, aber komplizierter. Mehrere `Combo`-Boxen für Jahr, Monat und Tag wären einfacher zu dekodieren, aber umständlicher in der Anwendung. Die Verwendung eines Schiebereglers zum Einstellen von Datum und Uhrzeit (**Bild 2**), wie im



**Bild 1**

► **Bild 1** Umständlich, aber möglich: Texteingabe durch aneinandergereihte `Combo`-Boxen mit Einzelbuchstaben.

Codebeispiel von **worst3.py** ist allerdings keine gute Idee: Das Slider-Widget gibt eine Zahl zwischen 0 und 999.999.999 zurück. Dies ist die Anzahl der seit dem 1. Januar 1970 verstrichenen Sekunden. Die Funktion `ctime()` wird verwendet, um diese Zahl in ein Datum und eine Uhrzeit zu verwandeln.

Es gibt allerdings auch Anwender, die am liebsten nur die Maus benutzen möchten. Vielleicht wäre dann bei einigen Eingaben, die nur aus wenigen Buchstaben bestehen, eine Reihe von Comboboxen, die jeweils alle Buchstaben des Alphabets enthalten, besser (**Bild 1**). Beginnen Sie mit dem Importieren des `guizero`-Widgets und der `ascii_letters` Liste.

```
from guizero import App, Combo
from string import ascii_letters
```

`ascii_letters` ist eine Liste mit allen darstellbaren ASCII-Zeichen, die Sie für die Combo-Box verwenden können.

Erstellen Sie eine einzelne Combo-Box, die alle Buchstaben enthält und die App anzeigt.

```
a_letter = Combo(app, options=" " + ascii_letters, align="left")
```

```
app.display()
```

Ihr Programm sollte nun der Datei **worst4.py** ähneln. Sie sehen eine einzelne Combo-Box, die alle Buchstaben plus ein Leerzeichen enthält und am linken Rand des Fensters ausgerichtet ist.

Um eine Reihe von Buchstaben zusammen zu bekommen, könnten Sie kontinuierlich Combo-Widgets zu Ihrer App hinzufügen, zum Beispiel:

```
a_letter = Combo(app, options=" " + ascii_letters, align="left")
b_letter = Combo(app, options=" " + ascii_letters, align="left")
c_letter = Combo(app, options=" " + ascii_letters, align="left")
```

Alternativ könnten Sie auch eine `for`-Schleife verwenden, eine Liste von Buchstaben erstellen und jeden Buchstaben an die Liste anhängen, wie in **worst5.py** gezeigt.

Letztlich ist es Geschmacksache. Die `for`-Schleife ist flexibler, da Sie so viele Buchstaben erstellen können, wie Sie möchten.

## worst1.py

> Sprache: Python 3

LADEN SIE DEN  
KOMPLETTEN CODE  
HERUNTER:



magpi.cc/guizero

```
001. # Imports -----
002.
003. from guizero import App, Text
004.
005. # App -----
006.
007.
008. app = App("it's all gone wrong", bg="dark green")
009.
010. title = Text(app, text="Hard to read", size="14", font="Comic
011. Sans", color="green")
012.
013. app.display()
```

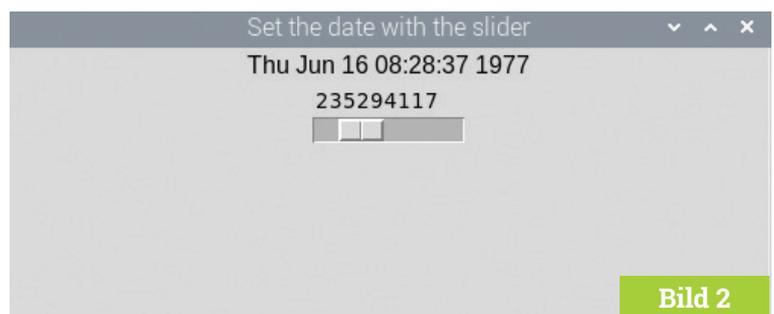


Bild 2

▲ Bild 2 Umständlich, aber möglich: Ein Schieberegler zum Einstellen von Datum und Uhrzeit.

## worst2.py

> Sprache: Python 3

```
001. # Imports -----
002.
003. from guizero import App, Text
004.
005. # Functions -----
006.
007.
008. def flash_text():
009.     if title.visible:
010.         title.hide()
011.     else:
012.         title.show()
013.
014. # App -----
015.
016.
017. app = App("it's all gone wrong", bg="dark green")
018.
019. title = Text(app, text="Hard to read", size="14", font="Comic
020. Sans", color="green")
021.
022. app.repeat(1000, flash_text)
023.
024. app.display()
```

## worst3.py

> Sprache: Python 3

```
001. # Imports -----
002.
003. from guizero import App, Slider, Text
004. from time import ctime
005.
006.
007. # Functions -----
008.
009. def update_date():
010.     the_date.value = ctime(date_slider.value)
011.
012.
013. # App -----
014.
015. app = App("Set the date with the slider")
016. the_date = Text(app)
017. date_slider = Slider(app, start=0, end=999999999,
018.     command=update_date)
019.
020. app.display()
```

## worst4.py

> Sprache: Python 3

```
001. # Imports -----
002. from guizero import App, Combo
003. from string import ascii_letters
004.
005.
006. # App -----
007.
008. app = App("Enter your name")
009.
010. a_letter = Combo(app, options=" " + ascii_letters, align="left")
011.
012. app.display()
```

► Bild 3  
Sinnlose Pop-up-  
Boxen

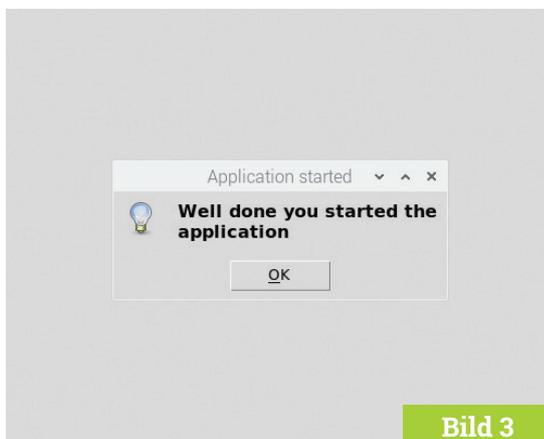


Bild 3

## Das Pop-up-Widget *Window*

Pop-up-Boxen können verwendet werden, um den Benutzer vor Entscheidungen zu stellen. Ihre Programmierung und Anwendung ist sehr einfach.

Wenn Sie zusätzliche Informationen anzeigen oder nach zusätzlichen Daten fragen möchten, können Sie das *Window*-Widget verwenden, um mehrere Fenster zu erstellen.

**Window** wird auf ähnliche Weise wie **App** verwendet und hat viele der gleichen Funktionen.

```
from guizero import App, Window

app = App("Main window")
window = Window(app, "2nd Window")

app.display()
```

Ob das Pop-up-Widget **Window** auf dem Bildschirm zu sehen ist oder nicht, können Sie mit den Methoden **show()** und **hide()** festlegen.

```
window.show()
window.hide()
```

Eine App kann dazu gebracht werden, auf das Schließen eines Fensters zu warten, nachdem es angezeigt wurde, indem man **True** an den **wait**-Parameter von **show** übergibt. Zum Beispiel:

```
window.show(wait=True)
```

Weitere Informationen über die Verwendung mehrerer Fenster finden Sie in der *guizero*-Dokumentation:

[lawsie.github.io/guizero/multiple\\_windows](https://lawsie.github.io/guizero/multiple_windows).

## Pop-ups

Keine als abschreckendes Beispiel dienende GUI wäre vollständig ohne eine Pop-up-Box. *guizero* enthält eine Reihe von Pop-up-Boxen, die dazu verwendet werden können, den Benutzer unter anderem auf etwas Wichtiges hinzuweisen – oder ihn zu irritieren und zu ärgern!

Erstellen Sie zunächst eine Anwendung, die beim Start ein sinnloses Kästchen einblendet, um mitzuteilen, dass die Anwendung gestartet wurde.

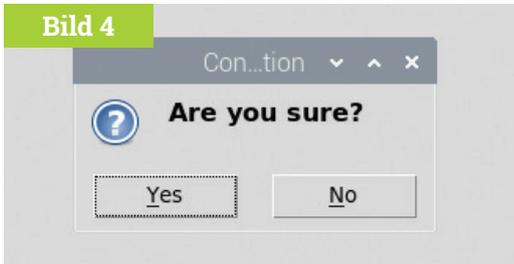
```
from guizero import App

app = App(title="pointless pop-ups")

app.info("Application started", "Well done
you started the application")

app.display()
```

Sie werden sehen, dass eine "info"-Box erscheint (Bild 3). Der erste Parameter, der an *info* übergeben



▲ Bild 4 Ja, wir sind uns sicher!

wird, ist der Titel des Fensters, der zweite Parameter ist die Meldung.

Sie können den Stil dieser einfachen Pop-up-Box ändern, indem Sie `warn` oder `error` anstelle von `info` verwenden.

Pop-up-Boxen können auch verwendet werden, um Informationen vom Benutzer zu erhalten. Das einfachste ist ein Ja-Nein-Feld, das dem Benutzer eine Frage stellt und auf Antwort wartet, zum Beispiel, ob er eine Datei wirklich löschen möchte. Aber natürlich nicht jedes Mal, wenn er irgend eine Taste drückt! Importieren Sie das `PushButton`-Widget in Ihre Anwendung:

```
from guizero import App, PushButton
```

Erstellen Sie eine Funktion, die das `yesno`-Pop-up benutzt, um nach einer Bestätigung zu fragen.

```
def are_you_sure():
    if app.yesno("Confirmation", "Are you
sure?"):
        app.info("Thanks", "Button
pressed")
    else:
        app.error("Ok", "Cancelling")
```

Fügen Sie die Schaltfläche in Ihre GUI ein, die die Funktion aufruft, wenn sie gedrückt wird.

```
button = PushButton(app, command=are_you_
sure)
```

Ihr Code sollte nun der Datei `05-worlds-worst-gui.py` ähneln. Wenn Sie die Anwendung ausführen und auf die Schaltfläche klicken, sehen Sie eine Pop-up-Box, die Sie auffordert, Ja oder Nein einzugeben (Bild 4).

Mehr über die Pop-up-Boxen in `guizero` erfahren Sie unter [lawsie.github.io/guizero/alerts](https://lawsie.github.io/guizero/alerts).

Wie wäre es, all diese "Funktionen" in einer ultimativen GUI zu kombinieren? [\[7\]](#)

## worst5.py

> Sprache: Python 3

```
001. # Imports -----
002.
003. from guizero import App, Combo
004. from string import ascii_letters
005.
006.
007. # App -----
008.
009. app = App("Enter your name")
010.
011. name_letters = []
012. for count in range(10):
013.     a_letter = Combo(app, options=" " + ascii_letters,
014.                     align="left")
015.     name_letters.append(a_letter)
016.
017. app.display()
```

## 05-worlds-worst-gui.py

> Sprache: Python 3

```
001. from guizero import App, PushButton
002.
003. def are_you_sure():
004.     if app.yesno("Confirmation", "Are you sure?"):
005.         app.info("Thanks", "Button pressed")
006.     else:
007.         app.error("Ok", "Cancelling")
008.
009. app = App(title="pointless pop-ups")
010.
011. button = PushButton(app, command=are_you_sure)
012.
013. app.info("Application started", "Well done you started the
014. application")
015.
016. app.display()
```

## Python 3 Programmierung und GUIs

Dies ist die zweite überarbeitete und aktualisierte Auflage eines Buches, das sich an Ingenieure, Wissenschaftler und Maker wendet, die PCs mittels grafischer Benutzeroberflächen mit Hardware-Projekten verbinden wollen, wobei sowohl Desktop- als auch webbasierte Anwendungen nicht zu kurz kommen. Als Programmiersprache wird Python 3 verwendet, eine schnelle – und eine der populärsten Sprachen überhaupt.

Mit diesem Buch lassen sich praktische Entwürfe leicht realisieren. Ein Texteditor ist alles, was zur Erstellung von Python-Programmen erforderlich ist. [www.elektor.de/python-3-programming-and-guis](http://www.elektor.de/python-3-programming-and-guis)





# Insel-Solaranlagen

## Elektrische Energie unabhängig vom Netz

Von **Dr. Thomas Scherer**

Was ist eine Insel-Solaranlage? Wo sind solche Lösungen notwendig und praktikabel? Was gibt es dabei zu beachten? Auf all diese Fragen und etwas mehr wird in diesem Artikel eingegangen.

Über Photovoltaik-Anlagen mit Netzanschluss haben wir bereits in Elektor 9-10/2021 berichtet [1]. In diesem Artikel geht es um vom öffentlichen Stromnetz isolierte, gewissermaßen autarke Solaranlagen. Damit generiert man auch dort elektrische Energie, wo ein Netzanschluss zu teuer wäre, wie etwa beim Gartenhaus eines Schrebergartens, oder gleich ganz unmöglich, wie etwa auf einem Motor- oder Segelboot. Es handelt es sich

in der Regel um Systeme kleinerer Leistung, die kurzfristig Spitzenleistungen im Bereich von einigen Watt bis zu wenigen Kilowatt abgeben können. Außerdem gibt es in Zeiten der sinkenden Einspeisevergütungen neuerdings sogar vereinfachte stationäre häusliche Solaranlagen mittlerer Leistung im Bereich einiger kWp (Kilowatt peak  $\approx$  Spitzenleistung der installierten Solarpaneele), welche nichts ins öffentliche Netz einspeisen, sondern die geerntete Solarenergie mit Hilfe eines Akkus für den kompletten Selbstverbrauch nutzbar machen. Doch zunächst zu den kleineren Anlagen.

### Prinzip und Details

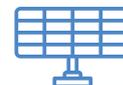
Für eine Insel-Solaranlage braucht es minimal drei Komponenten: Zunächst die Solarzellen, dann einen Energiespeicher in Form eines Akkus und schließlich einen Laderegler, der dafür sorgt, dass der Akku nicht überladen wird. Für kleine Insel-Lösungen auf der typischen 12-V-Basis reicht das im Prinzip schon aus. Will man zusätzlich auch noch 230-V-Wechselspannung mit 50 Hz oder

60 Hz, braucht es als vierten Teil noch einen passenden Wechselrichter. **Bild 1** zeigt so eine Vier-Komponenten-Lösung. Das sieht sehr einfach aus, aber wie so oft stecken die Details voller kleiner Teufel. Im Folgenden wird daher speziell auf die einzelnen Komponenten eingegangen.

Hier ein Praxisbeispiel: Aufgrund der einfachen Handhabung und des günstigeren Preises wegen hat sich mein guter Freund Klaus bei seinem Gartenhaus für eine 12-V-Anlage entschieden. Für die Planung und die Auslegung der Komponenten mussten zunächst zwei Fragen beantwortet werden.

### Energie & Leistung

Die erste Frage ist, wieviel elektrische Energie vorgehalten = gespeichert werden soll. Da sich dies direkt auf die Kapazität des geplanten Akkus auswirkt, muss man natürlich überschlagen, welche mittlere Last von der Inselanlage versorgt werden soll. Relevant ist außerdem, wieviel Tage die Anlage bei Bewölkung „überstehen“ soll. Klaus wollte in seinem Gartenhaus eine Bohrmaschine



betreiben oder sich eine Tasse Tee brauen, aber so etwas nur selten, weshalb man diesen Energieverbrauch fast vernachlässigen kann. Jedoch sollte durchgängig ein 12-V-Kühlschrank mit einer mittleren Leistungsaufnahme von etwa 20 W versorgt sein, damit immer kühles Bier zur Verfügung steht. Das Ganze müsste mindestens einen Tag ohne Sonnenschein funktionieren.

Die zweite Frage gilt der benötigten Spitzenleistung. Davon hängt nämlich die Strombelastbarkeit des eingesetzten Akkus ab und in der Folge sind davon auch der Laderegler und ein eventuell vorhandener Wechselrichter betroffen. In der Regel ist diese Frage recht einfach zu beantworten. Beim Schrebergartenhäuschen von Klaus waren das zum Beispiel 1 kW, denn er wollte einen Wasserkocher, eine normale Bohrmaschine und gelegentlich auch eine Wasserpumpe an 230 V betreiben.

### Solarpanel

In 24 Stunden dürfte der Kühlschrank im Beispiel von Klaus maximal 500 Wh verbrauchen. Klaus lebt im sonnenverwöhnten Freiburg, mochte aber das Solarpanel nicht aufs Dach montieren, weil es dort von einem Kirschbaum abgeschattet worden wäre. Stattdessen wurde das Panel senkrecht an eine recht gut nach Süden orientierte Wand des Gartenhauses geschraubt. Dies reduziert die mögliche Energieausbeute um bis zu 30 % gegenüber dem optimalen Winkel zur Sonne. Man muss das Solarpanel also um fast 40 % überdimensionieren, um diesen Verlust zu kompensieren. Kein großes Problem, da Platz genug da ist und Solarpanels in den letzten Jahren preiswert geworden sind. Der Vorteil der senkrechten Montage ist, dass im Winter kein Schnee auf dem Panel liegen bleibt, durch den dann günstigeren Einstrahlwinkel die Ausbeute steigt und somit im optimalen Fall sogar an sonnigen Tagen im Winter das Bier kühl bleibt.

Nun zur Berechnung der nötigen solaren Leistung: In Freiburg kann man mit über 1.200 kWh/m<sup>2</sup> im Jahr rechnen. Für den avisierten täglichen Verbrauch sollte eine Sicherheitsreserve von 100 % ausreichen. Bei den angenommenen 500 Wh/d (Frühjahr bis Herbst) sollte man also 1 kWh/d zu erntende Energie anpeilen. Kalkuliert man mit acht Sonnenstunden pro Tag, kommt man auf eine

notwendige Leistungsabgabe des Panels von etwa 125 Wp. Hinzu kommt die Kompensation wegen senkrechter Montage und man landet bei 175 Wp. Daher ist es ein 180-W-Panel (Bild 2) geworden – keine schlechte Wahl.

### Akku(s)

Der Energiespeicher sollte bei einem Tag Reserve  $\geq 500$  Wh fassen. Bei 12 V Nennspannung wäre also ein Akku mit einer Kapazität von mindestens 40 Ah erforderlich. Da ein

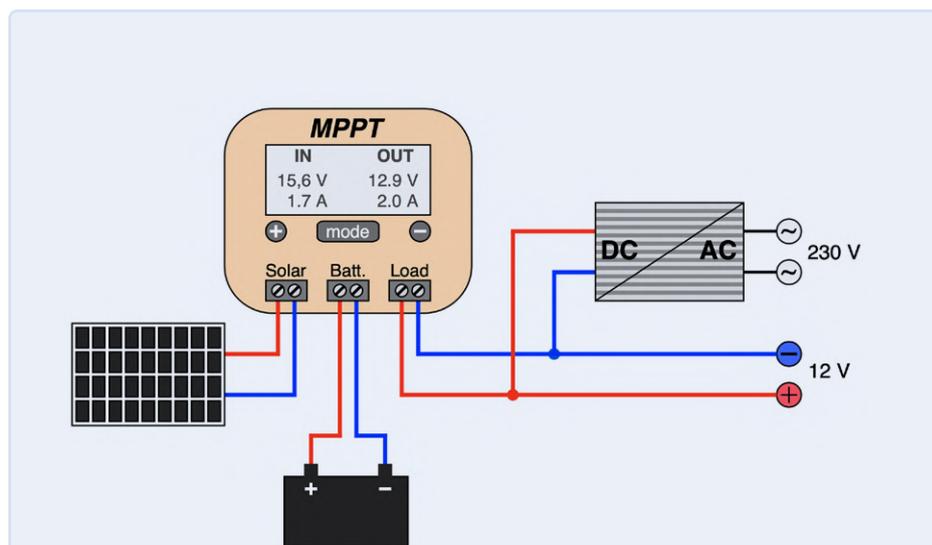


Bild 1. So werden die üblichen vier Komponenten einer solaren Inselanlage üblicherweise verschaltet. Der Inverter rechts ist nur erforderlich, wenn man für 230 V gedachte Geräte betreiben will.



Bild 2. Das 12-V-Solarpanel am Gartenhäuschen von Klaus ist senkrecht montiert und liefert 180 Wp.



Bild 3. Drei 12-V-Blei-Gel-Akkus mit je 36 Ah parallelgeschaltet als Energiespeicher im Gartenhäuschen von Klaus.



Bild 4. Von links nach rechts: Lichtschalter, elektromagnetische 30-A-Sicherung und MPPT-Laderegler.



Bild 5. Was so aussieht wie dieser Regler, hat sicher kein MPPT an Bord – selbst wenn diese vier Buchstaben aufgedruckt sein sollten (Bild: United States Department of Energy).

Inverter mit 1 kW Ausgangsleistung vorgesehen war, muss man im Hinterkopf behalten, dass dieser bei maximaler Last einen Eingangsstrom von gut 85 A benötigt. Dieser Aspekt ist wichtig bei der Akku-Auswahl. Zunächst musste der Akkutyp festgelegt werden. Ein Lithium-Akkupaket mit 40 Ah könnte aufgrund des niedrigen Innenwiderstands locker mit diesem Strom (etwa 2C = dem doppelten Strom, den der Akku 1 h lang liefern kann) fertig werden. Aber solche Akkus kosten schnell über 250 € und sind komplexer in der Handhabung. Daher wollte Klaus mit der simplen Blei-Akku-Lösung vorliebnehmen, die es für einen Bruchteil dieses Preises gibt. Naheliegender wäre eine Auto-Batterie, denn die packt auch hohe Spitzenströme. Ihre Nachteile (geringer Wirkungsgrad, geringe Zyklenbelastbarkeit und höhere Selbstentladung) führten zum Kompromiss Blei-Gel-Akku. Dieser Typ wiederum mag hohe Entladeströme nicht wirklich, weshalb sich Klaus letztlich für zwei parallelgeschaltete 12-V-Blei-Gel-Akkus mit je 36 Ah entschieden hatte. Diese Lösung bietet rechnerisch 864 Wh an gespeicherter Energie für knapp unter 150 €. Scheint die Sonne voll, liefert das gewählte Solarpanel so viel Energie, dass die Akkus an einem schönen Sommertag in Freiburg locker vollständig geladen werden. Der Energievorrat reicht gut für 1,5 Tage ohne Sonnenschein. Ich selbst hatte Bedenken wegen des hohen Stroms für den Inverter, aber Klaus wollte es zunächst darauf ankommen lassen und meinte, dass er immer noch einen Akku dazu kaufen könne, falls sie zu „weich“ sein sollten. Als die Anlage installiert und die Akkus voll geladen waren, zeigte ein erster Test mit dem 1-kW-Wasserkocher, dass während des Kochens die Nennspannung an den Akkupolen auf 11,7 V einbrach. Es war problemlos möglich, einen halben Liter Wasser zum Kochen zu bringen, aber der Wirkungsgrad der Akkus (Ladeenergie/Entladeenergie) bei diesen hohen Strömen lag geschätzt höchstens noch bei 50 % und den Akkus tat diese Extremlast auch nicht gut. Folglich wurde ein dritter 36-Ah-Akku nachgeordert und dazu geschaltet (**Bild 3**). Jetzt betrug die Entladespannung bei 85 A anfangs noch 12,6 V, was akzeptabel war. Außerdem werden nun fast 1,3 kWh gespeichert, was über zwei Tage Reserve garantiert.

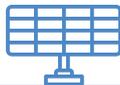


Bild 6. Der 1-kW-Inverter von Ective erwies sich bislang als sehr stabil und zuverlässig.

### Laderegler

Schaut man sich bei eBay oder spezialisierten Händlern um, wird man von der Vielfalt an Laderegler geradezu erschlagen. 10-A-Regler gibt es schon ab 15 €. Ein 180-Wp-Panel liefert an 12 V aber immerhin bis zu 15 A. Es muss also mindestens ein 20-A-Regler sein, den es für rund 20 € gibt. Wird der Laderegler aber so angeschlossen wie in Bild 1, sollte man besser eine 100-A-Variante wählen, die mit knapp unter 50 € zu Buche schlägt. So viel vorab.

Ein Laderegler hat die Aufgabe, den Akku mit der vom Panel kommenden Energie zu laden und die maximale Ladeschlussspannung einzuhalten, damit der angeschlossene Akku nicht überladen und somit beschädigt wird. Fast alle Laderegler steuern auch einen Lastanschluss, indem sie dafür sorgen, dass die Last bei Unterspannung abgetrennt und der Akku somit vor einer Tiefentladung geschützt wird. Da sie alle über einen Mikrocontroller verfügen, können fast alle Solar-Laderegler mit verschiedenen Akkutypen (normaler Blei-Akku, Blei-Gel-Akku und Lithium-Akku) und automatisch sowohl mit 12-V- als auch 24-V-Nennspannung umgehen. Außerdem lassen sich die Spannungen für Ladeschluss und Entladeschluss oft manuell festlegen.

Der nächste Aspekt ist die Ladetechnik, die ein Laderegler beherrscht. Dabei gilt: Alle preiswerten Exemplare arbeiten mit einer PWM-Regelung – auch wenn MPPT

draufsteht. Aufdrucke sind in China nämlich sehr preiswert – sie kosten lediglich die Druckerfarbe. „Echtes MPPT“ ist besser, aufwändiger und daher deutlich teurer!

Bei einer PWM-Regelung wird der Ladestrom so eingestellt, dass die Abgabespannung des Panels gerade noch über der aktuellen Akkuspannung liegt. Es wird also über einen weiten Regelbereich einfach mit dem maximal möglichen Strom geladen, der von der Beleuchtung des Panels, seiner „Größe“ und dem Ladezustand des Akkus abhängt. Außer einem simplen Mikrocontroller braucht es dazu nur einen Power-MOSFET – das ist preiswert, aber nicht optimal. Die Abgabeleistung eines Panels ergibt sich aus dem Produkt von Spannung und Strom. Dabei gibt es für jedes Panel und jede Beleuchtung einen optimalen Punkt maximaler Abgabeleistung, bei dem die Spannung des Panels fast immer über der aktuellen Akkuspannung liegt. Das MPPT-Verfahren (Maximum Power Point Tracking) ermittelt diesen „sweet spot“ permanent und steuert einen nachgeordneten Step-Down-Schaltregler über dessen Stromaufnahme so, dass sich immer die maximale Leistungsabgabe ergibt. Im optimalen Fall ist die Ausbeute mit MPPT um bis zu 30 % höher als bei PWM, aber das kostet eben. Die preiswertesten MPPT-Regler gibt es aktuell nicht unter 50 €. Für Markengeräte muss man mit mindestens 100 € rechnen. Der 30 A-Solarregler von Klaus in **Bild 4** ist ein preiswerter MPPT-Regler für etwa 60 €. Die Ausbeute war ihm also wichtig. Sollten Sie auch einen MPPT-Regler wollen, fallen Sie nicht auf Exemplare wie das von **Bild 5** herein. Diese Typen gibt es in allen Farben und mit unterschiedlichsten Bezeichnungen.

### Inverter

Falls man selbst eine Wechselspannung von 230 V erzeugen möchte, kommt man um einen Inverter nicht herum. Von preiswerten Exemplaren mit Fantasieangaben bezüglich Leistung und nicht annähernd sinusförmigem Spannungsverlauf sollte man gleich ganz die Finger lassen. Weiter wichtig: Die Angabe der maximalen Dauerleistung am Ausgang bezieht sich auf eine ohmsche Last. Die 1 kW des Inverters von Klaus harmonisieren mit seinem 1-kW-Wasserkocher mit 500 ml Fassungsvermögen 1:1. Völlig anders ist die Situation bei induktiven



**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.
- Schlißbildauswertung.
- 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

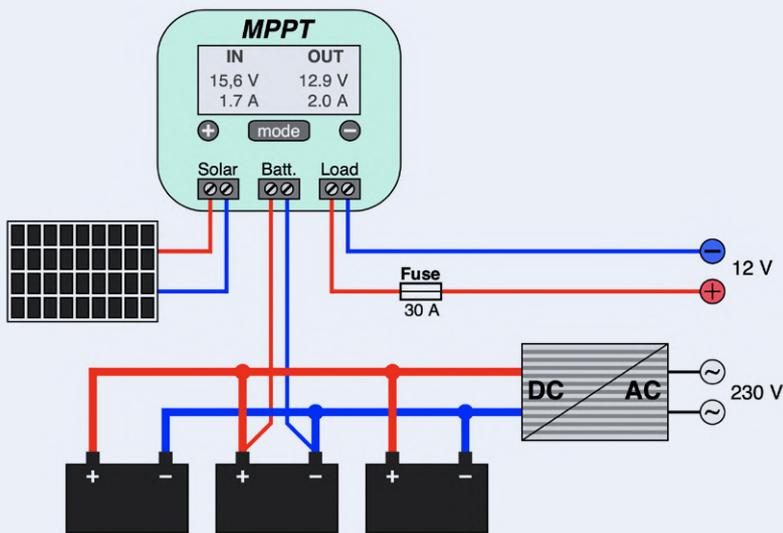


Bild 7. Bei der Lösung von Klaus ist der Inverter direkt mit den Akkus verbunden und der 12 V-Ausgang des Ladereglers extra abgesichert.

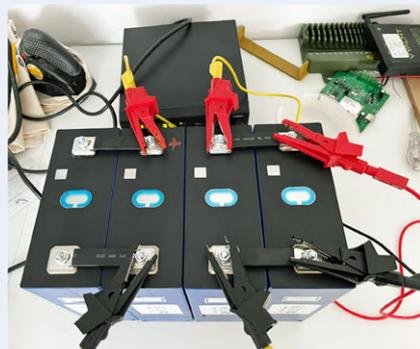


Bild 8. Die LiFePo-Akkus von Martin beim Test – vor dem Einbau in sein Boot (Bild: Martin Jepkens).



Bild 9. Das von Martin und Detlev verwendete faltbare Solarpanel liefert 120 Wp und wird bei Fahrt unter Deck verstaut (Bild: Martin Jepkens).

(häufiger) und kapazitiven Lasten (seltener). Hier muss man zusätzlich die Blindleistung berücksichtigen - die aufgenommene Leistung ist immer höher als die reine Wirkleistung. Am problematischsten sind Elektromotoren, denn sie haben zum Teil hohe Anlaufströme, welche bei zu knapp dimensionierten Invertern den integrierten Überlastschutz triggern. Eine Reserve von 100 % ist beim Betrieb von Motoren auch bei guten Invertern nicht übertrieben. Der 1-kW-Inverter von Klaus in **Bild 6** packt problemlos Bohrmaschinen und

eine Wasserpumpe mit 450 W Nennleistung. Er war nicht unter 200 € zu haben.

#### Verkabelung

Wie in den bisherigen Bildern zu erahnen wurde die Verbindung von Solarpanel mit Laderegler und Akku mit Laderegler durch feindrähtige Litze mit 6 mm<sup>2</sup> vorgenommen (nicht abgebildet sind die 12-V-Ausgänge via Auto-Zigarettenanzünder-Buchsen). Die Akkus selbst aber wurden durch Litzen mit 16 mm<sup>2</sup> parallelgeschaltet. Die Dicke der Drähte richtet

sich nach dem auftretenden Strömen – hier sollte man nicht am falschen Ort sparen. Außerdem wurde der Inverter direkt via 16-mm<sup>2</sup>-Leitungen an die Akkus geklemmt, um Verluste über den integrierten MOSFET des Ladereglers zu umgehen. Das kann man aber nur machen, wenn sich der Inverter (wie der hier eingesetzte) bei Unterspannung selbst abschaltet und so den Akku vor Tiefentladung schützt. Der Inverter wird nur eingeschaltet, wenn tatsächlich 230 V benötigt werden, denn sein Ruhestrom von einigen zig mA würde sonst unnötig Energie verbrauchen. Die resultierende Verschaltung ist in **Bild 7** zu sehen.

#### Andere Insel-Anlagen

Die Elektrifizierung des Gartenhäuschens von Klaus ist ein typisches Beispiel für die Installation einer solaren Insel-Anlage. Hierfür gibt es bei diversen Anbietern auch fertige Pakete aus Solarpanel, Regler und Inverter unterschiedlicher Leistung. Setzt man auf Wind- statt auf Solarenergie, gibt es auch hierfür passende Generatoren und Regler, doch das Prinzip bleibt im Wesentlichen gleich. Ich selbst hatte letztes Jahr meinen Rasenmäher-Roboter autark gemacht [1]. Hier war nur ein 50-W-Panel vorgesehen und ein einfacher PWM-Regler. Ein Inverter war überflüssig. Mittlerweile habe ich die Akkukapazität von 12 Ah auf nunmehr 30 Ah erhöht, da es dieses Jahr einige Regentage gab. Außerdem wurde der PWM-Regler dieser Tage durch einen besseren und auch teureren MPPT-Regler ersetzt, damit auch noch im Spätherbst genug Energie zum Mähen geerntet werden kann. Es gibt aber auch noch gänzlich andere Anwendungen für Insel-Anlagen: Zwei meiner Freunde haben Boote. Während Martin mit seinem größeren Stahlboot (Verdränger) durch die niederländischen Flusslandschaften navigiert, macht Detlev mit seinem schnellen Sportboot (Gleiter) das Mittelmeer unsicher. Beiden gemeinsam ist, dass sie oft Tage von Häfen und sonstigen Anlegestellen entfernt sind und sich daher mehr Autonomie bezüglich des Kühlschranks wünschen – dabei geht es nicht nur um Bier, sondern auch um essbare Lebensmittel. Damit sie nicht in unökologischer und ineffizienter Weise häufig den Motor laufen lassen müssen, um den Bord-Akku zu laden, haben beide eine Solaranlage nachgerüstet. Martin ist selbst ein ausgefuchster Elektroniker

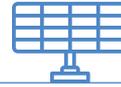


Bild 10. Anleitung zum Einbau des Ladereglers in das Boot von Detlev.

Kapazitätstests. Für LiFePo-Technik hat er sich vor allem aufgrund der vielen möglichen Ladezyklen entschieden. Außerdem brauchen LiFePos verglichen mit Blei-Akkus weniger Platz. Bord- und Starterbatterie sind in Martins Boot entkoppelt. Um die Lichtmaschine zu schonen, werden die Akkus über einen eigenen Laderegler geladen, wenn der Motor läuft. „Für unterwegs“ beziehungsweise ohne Netzanschluss in einem Hafen ist auch noch ein faltbares 120-Wp-Solarpanel samt Solar-Regler vorgesehen (Bild 9).

Beim Sportboot von Detlev wäre ein fest installiertes Solarpanel schon aus Platzmangel nicht denkbar. Deshalb hat er sich für das gleiche Panel wie Martin entschieden. Beide wussten nichts von der Wahl des jeweils Anderen. Detlev ist allerdings kein Elektroniker, und er wollte zunächst seinen extra 120-A-Bordakku weiter nutzen, weil er noch recht neu war. Also musste ich für ihn rechnen und ihm auch mitteilen, dass die Verwendung der Buchse des Zigarettenanzünders auf seiner „Brücke“ zum Anschluss des Solarpanels aus Zuverlässigkeitsgründen keine gute Idee ist. Ich hatte ihm die Verwendung von wasserdichten Neutrik-Steckverbindungen vorgeschlagen. Daher durfte ich die Stecker auch vorverdrahten und eine Anleitung (Bild 10) für seinen Bootsbauer erstellen, damit der das Ganze am Hafen in Istrien einbauen kann. Die Kombination aus faltbarem 120-Wp-Panel plus MPPT-Laderegler von

und nicht unbedingt auf meine Unterstützung angewiesen. Er diskutiert seine Überlegungen aber manchmal mit mir. So beschäftigte ihn die Frage, ob die Lichtmaschine seines Bootsmotors überlastet würde, wenn er gelegentlich

einen dicken, fetten 200-Ah-LiFePo-Akku dazuschaltet. Die Gefahren dieses Vorgehens werden zum Beispiel in einem YouTube-Video erläutert [2]. Bild 8 zeigt seine Akku-Anordnung bei den von ihm vorgenommenen

Anzeige

10. - 12. Mai  
**pcim** EUROPE  
Halle 6  
Stand 452

## THE NEW COOL BIS ZU 1200 WATT OHNE LÜFTER

**WARTUNGSFREIE STROMVERSORGUNG MIT LÜFTERLOSER KÜHLUNG  
FÜR MEDIZINISCHE UND INDUSTRIELLE ANWENDUNGEN**

- Kontakt- oder Konvektionsgekühlt
- Betriebstemperatur: von -40° bis +80°C
- Zertifiziert für medizinische, industrielle und ITE-Sicherheit
- Isolationssystem 2 MOPP, für BF-Anwendungen geeignet
- PMBus Version verfügbar\*
- Optional: Kundenspezifische Lösungen

\*RACM600-L/OF, RACM1200-V

101.6 mm  
50.8 mm

228.0 mm  
96.2 mm

127.0 mm  
76.0 mm

196.8 mm  
101.6 mm

**WE POWER YOUR PRODUCTS**  
recom-power.com/medical

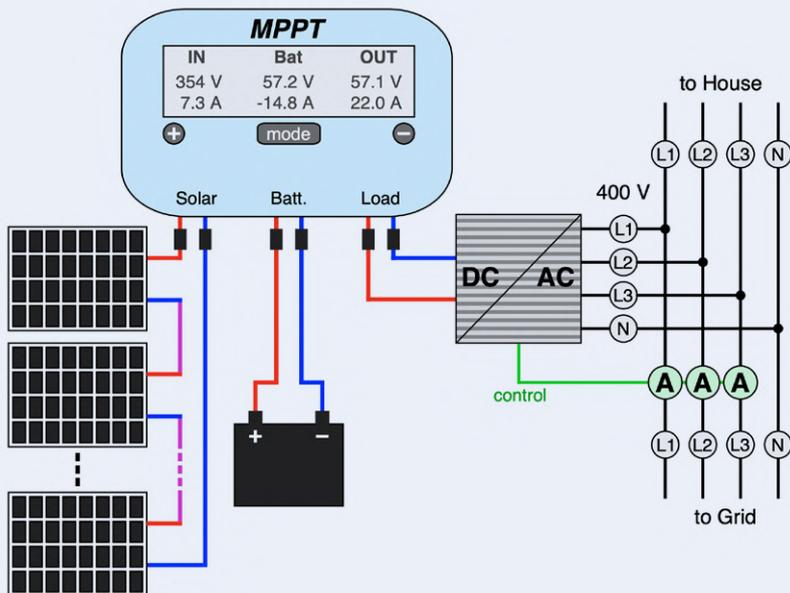
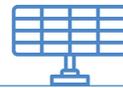


Bild 11. Solare Quasi-Inselanlage für ein Einfamilienhaus. Die Dreiphasen-Stromerfassung steuert den Wechselrichter so, dass keine elektrische Energie ins Netz abgegeben wird.

teuerste Element ist hier der Akku. Für einen anschlussfertigen 6,5-kWh-LiFePo-Akku muss man etwas über 3.000 € kalkulieren. Bei garantierten 6.000 Ladezyklen mit einem Hub von 90 % ergeben sich etwa 36 MWh, die über den Akku geflossen sind. Das ergibt Kosten von etwa 9 ¢/kWh. Da der Akku dann noch nicht tot ist, kann man mit noch geringeren Kosten pro kWh für den Akku rechnen. Über die Laufzeit spart man mit solch einer Lösung grob geschätzt Stromkosten von mehr als 13.000 € durch den damit möglichen hohen Selbstverbrauch. Lädt man mit solch einer Anlage auch noch ein Elektroauto (mit geringer Leistung), hat sich die Anlage in wenigen Jahren amortisiert. ◀

210644-02

### Ein Beitrag von

Text und Bilder (soweit nicht anders angegeben): **Dr. Thomas Scherer**  
 Redaktion: **Jens Nickel**  
 Layout: **Harmen Heida**

Victron Energy hat knapp unter 500 € gekostet. Der Laderegler verfügt über Bluetooth, so dass alle Parameter und Kurven per Smartphone mit einer App überwacht werden können.

### Das Haus als Halbinsel

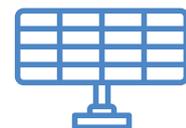
In Zeiten abnehmender Einspeisevergütungen beginnt eine Variante stationärer Solaranlagen interessant zu werden, die möglichst alle geerntete Solarenergie dem häuslichen Verbrauch zuordnet. Ein Array von zum Beispiel zehn Solarpanels liefert heute 3,75 kWp. Ein passender MPPT-Solarregler lädt einen LiFePo-Akku mit einer Energiekapazität von beispielsweise 6,5 kWh und ein dreiphasiger Wechselrichter wird von einer Stromüberwachungseinheit (die drei Amperemeter rechts unten in **Bild 11**) so gesteuert, dass unter keinen Umständen elektrische Energie ins Netz eingespeist wird. Aller „Strom“ wird so selbst verbraucht. Da für das Jahr 2022 in Deutschland mit kWh-Preisen von um die 36 ¢ gerechnet wird, ist das durchaus attraktiv, denn man spart sich einen

komplexeren und auch teureren einspeisefähigen Wechselrichter mit integrierter Ladeelektronik für den Akku sowie viel Bürokratie, was nicht zu verachten ist.

Bei der Anlage von Bild 11 summieren sich die Einsparungen für die vereinfachte Anlage auf 1.000 € bis 2.000 €. Es würde ein paar Jahre brauchen, bis man das mit der Einspeisevergütung wieder raus hätte. Das

### Sie haben Fragen oder Kommentare?

Bei technischen Fragen können Sie sich gern an die Elektor-Redaktion wenden unter der E-Mail-Adresse [redaktion@elektor.de](mailto:redaktion@elektor.de)



### PASSENDE PRODUKTE

- > PeakTech Stromzange 4350 (SKU 18161) [www.elektor.de/18161](http://www.elektor.de/18161)
- > Pokit Meter – Portables Multimeter, Oszilloskop und Logger (SKU 19854) [www.elektor.de/19854](http://www.elektor.de/19854)
- > PeakTech 3445 True RMS Digital Multimeter mit Bluetooth (SKU 18774) [www.elektor.de/18774](http://www.elektor.de/18774)

### WEBLINKS

- [1] Thomas Scherer, „Balkonkraftwerk“, ElektorMag 9-10/2021: <https://www.elektormagazine.de/210326-02>
- [2] Thomas Scherer, „Solaranlage für Mähroboter“, ElektorMag 7-8/2021: <https://www.elektormagazine.de/200553-02>
- [3] Victron Energy: „How to not blow up your Alternator when charging Lithium“: <https://www.youtube.com/watch?v=jgolocPgOug>

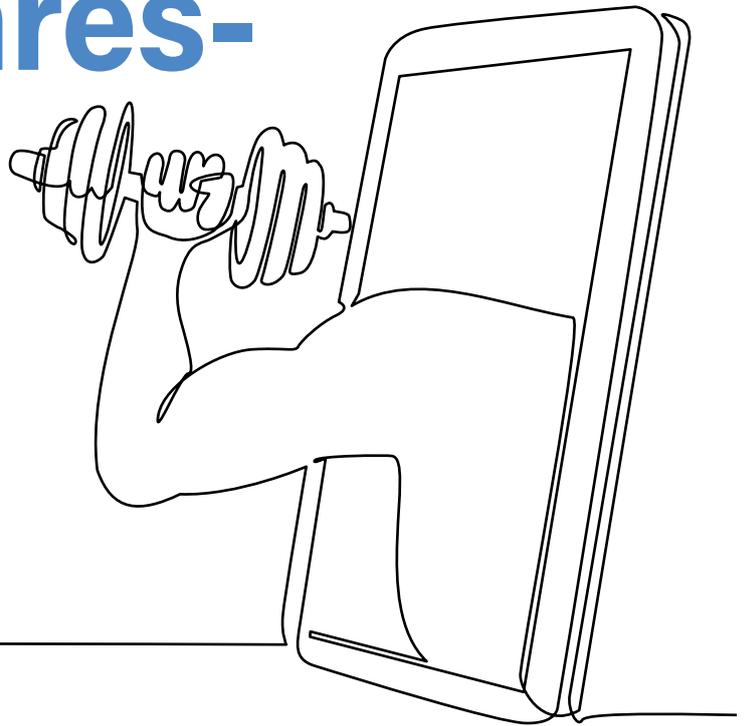


# Das 10-Jahres-Handy

Erneuern Sie Ihre Erwartungen!

Von Priscilla Haring-Kuipers (Niederlande)

Was wir von unseren Geräten erwarten, bestimmt, was wir von unseren Geräten akzeptieren. Wir erwarten zum Beispiel von unserem Smartphone keine lange Lebensdauer.



Vielleicht haben Sie im Internet schon einmal eine Anzeige für ein neues Smartphone gesehen, das zehn Jahre halten soll. Vielleicht haben auch Sie darauf geklickt, nur um dann enttäuscht festzustellen, dass es dieses Handy nicht gibt ... noch nicht. Die Initiative „10 years phone“ versucht, Ihre Aufmerksamkeit und die der EU-Gesetzgeber zu gewinnen, indem sie uns mit einer Zukunft lockt, in der es normal wird, dass ein Smartphone ein Jahrzehnt lang hält. Damit dies tatsächlich funktioniert, müssten Rechtsvorschriften über die Reparierbarkeit, die fortgesetzte Softwareunterstützung und die Verfügbarkeit von Ersatzteilen - insbesondere des Akkus - erlassen werden. Informationen darüber, wie man sein Telefon selbst reparieren kann, sowie eine offizielle Einstufung der Reparierbarkeit sollten ohne weiteres zugänglich sein [1].

## Zeitplan

Irgendwann auf dem Weg zu immer erstaunlicheren Mobiltelefonen scheinen wir uns damit abgefunden zu haben, dass diese teuren Helfer nur zwei bis vier Jahre mit voller Funktionstüchtigkeit halten. Ich schätze, wir machen uns nicht oft Gedanken darüber, wie viel Lebensdauer wir eigentlich von unseren Geräten erwarten. Wie lange sollte Ihr 4K-Flachbildschirm

halten? Was ist mit Ihrer Waschmaschine, Ihrem Wecklicht oder Ihrem Reflow-Ofen? Zwei bis vier Jahre scheinen für die meisten Geräte gar nicht so schlecht zu sein, denn in diesem Zeitraum wollen Sie wahrscheinlich sowieso etwas Neues. Das ist eine seltsame Auffassung. Woher kommt dieser Zwang, ständig aufzurüsten, und ist es nicht an der Zeit, dass wir ihn loswerden? In einer Zeit, in der unser Planet in unseren Produkten erstickt, der knappen Ressourcen und des Leids, die mit der Herstellung vieler unserer Geräte verbunden sind, schätzen wir selber das, was wir haben, nur gering ein. Wie wäre es aber, wenn wir unsere Geräte und unsere mentalen Modelle so anpassen, dass sie länger halten?

## Ein neuer Reflex

Mein Smartphone ist jetzt etwa fünf Jahre alt und ich muss dringend den Akku austauschen lassen. Der Ladestand sinkt im Handumdrehen von 42 % auf leer. Ich ertappte mich dabei, wie ich mich nach einem neuen Handy umsah, bis ich feststellte, dass ich mit meinem immer noch sehr zufrieden bin, abgesehen von der Todessehnsucht des Akkus. Mit diesem Reflex nach Neuem bin ich nicht allein: Nur 11 % der Menschen in der EU reparieren ihr Handy, wenn es nicht mehr funktioniert. Würde ich mich auch sofort nach

einer neuen Waschmaschine umsehen, wenn sie plötzlich den Schleudergang verweigern würde? Ich glaube nicht. Ich glaube, mein Reflex wäre es, das Problem zu googeln, ob das Internet vielleicht weiß, dass das ständig passiert und ich einfach die Dingsbums-Düse reinigen sollte. Ich würde die Waschmaschine selbst warten und reparieren. Und sollte das nicht klappen, würde ich wahrscheinlich einen Fachmenschen mit der Reparatur beauftragen, WENN die Waschmaschine nicht älter als zehn Jahre ist.

Waschmaschinen und Smartphones liegen in der gleichen Preisklasse. Aber warum erwarten wir nicht, dass unsere Mobiltelefone ein Jahrzehnt lang halten? Es ist an der Zeit, dass wir smarter werden!

*Das 10-Jahres-Smartphone ist eine Initiative der Europäischen Kampagne für das Recht auf Reparatur, einem Zusammenschluss von über 80 Organisationen aus ganz Europa, die sich für langlebigere und besser reparierbare Produkte einsetzen.*

210714-02

## WEBLINK

[1] 10-Jahres-Smartphone:  
<https://10yearphone.com/>

# 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.com](mailto:sale@elektor.com)). Unsere Bedingungen: **Nie teuer, immer überraschend!**

### puhui T-962 Infrarot Reflow-Lötöfen

Preis: 229,00 €

**Mitgliederpreis: 206,10 €**

 [www.elektor.de/20135](http://www.elektor.de/20135)

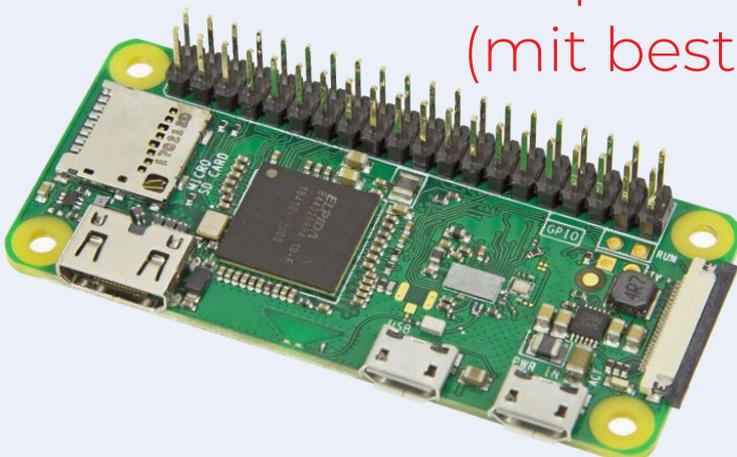


### Raspberry Pi Zero 2 WH (mit bestücktem GPIO-Header)

Preis: 34,95 €

**Sonderpreis: 29,95 €**

 [www.elektor.de/20157](http://www.elektor.de/20157)





## MakePython ESP32 Development Kit

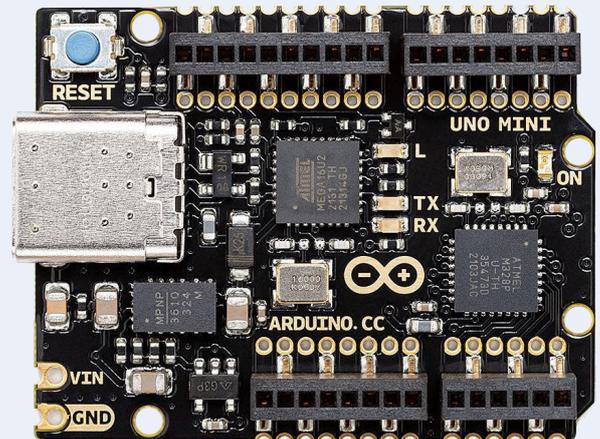


Preis: 74,95 €

**Sonderpreis: 64,95 €**

 [www.elektor.de/20137](http://www.elektor.de/20137)

## Arduino Uno Mini (Limited Edition)

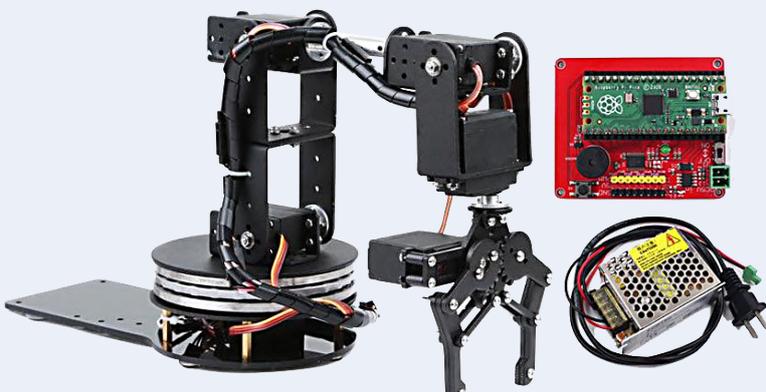


Preis: 54,95 €

**Mitgliederpreis: 49,46 €**

 [www.elektor.de/20098](http://www.elektor.de/20098)

## Makerfabs 6 DOF Roboterarm mit Raspberry Pi Pico

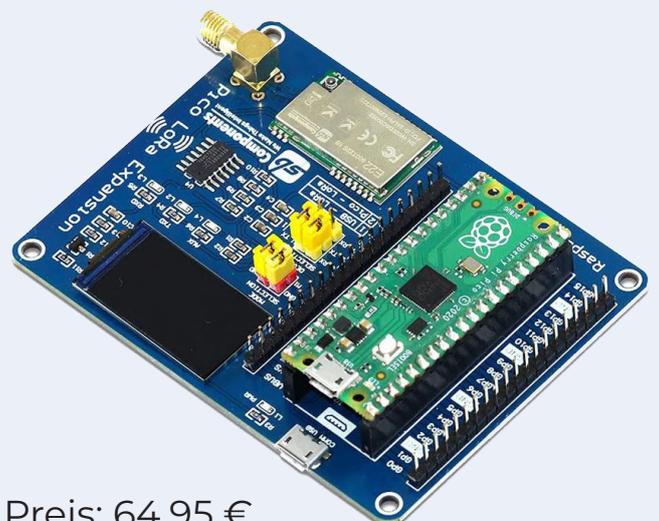


Preis: 139,95 €

**Mitgliederpreis: 125,96 €**

 [www.elektor.de/20130](http://www.elektor.de/20130)

## Raspberry Pi Pico LoRa Expansion + GRATIS Raspberry Pi Pico



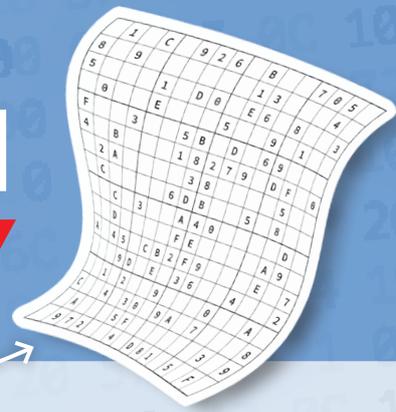
Preis: 64,95 €

**Mitgliederpreis: 58,46 €**

 [www.elektor.de/20096](http://www.elektor.de/20096)

# 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, Fax oder Post an:

**Elektor Redaktion**  
Kackertstr. 10  
52072 Aachen

Fax: 0241 / 955 09-013

E-Mail: [hexadoku@elektor.de](mailto:hexadoku@elektor.de)

Als Betreff bitte nur die Ziffern der Lösung angeben!

Einsendeschluss ist der  
15. Juni 2022.

## DIE GEWINNER DES HEXADOKUS AUS DER AUSGABE MÄRZ/APRIL STEHEN FEST!

Die richtige Lösung ist: **C73B8**

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](http://www.elektormagazine.de/hexadoku) bekannt gegeben.

Herzlichen Glückwunsch!

	D	7		1	6							B		8	0
4	6			D	5			0						C	F
2				F	4						B				E
			3	8		2		A	9		D				5
D	E	B	5	2			6				8			3	C
8	0	3				1			2	F	C	D			
			4		8	0			7		F				
				3	4			A	5						
	1		E								B		6		F
			C		D	8	7				E		9	B	
		2			E	0				F	5	3	7		
			7	B	F		3	E		1	4				
6					1	A			C	D		3	9		5
	A							1	5		4				8
9	C	5	D	7					4	2					1
B		4		6				9	E			C	7	A	

E	2	6	9	0	3	C	A	F	5	7	B	1	4	8	D
4	C	7	3	B	8	5	D	E	1	9	6	A	F	2	0
5	B	0	D	E	F	4	1	C	2	8	A	3	6	7	9
8	F	1	A	2	6	7	9	3	D	0	4	5	C	B	E
A	3	2	0	5	C	F	8	4	6	D	9	B	7	E	1
1	D	B	8	4	2	E	6	0	7	A	5	C	3	9	F
7	5	C	6	1	9	A	B	2	3	E	F	4	0	D	8
F	E	9	4	3	7	D	0	1	8	B	C	2	A	5	6
3	0	D	E	6	A	9	F	B	C	1	2	7	8	4	5
2	6	F	1	C	B	8	E	5	4	3	7	9	D	0	A
9	4	5	7	D	0	1	2	8	A	F	E	6	B	3	C
B	8	A	C	7	5	3	4	9	0	6	D	E	1	F	2
0	7	3	2	8	4	6	5	D	E	C	1	F	9	A	B
6	A	4	B	F	E	0	C	7	9	5	8	D	2	1	3
C	1	E	F	9	D	2	3	A	B	4	0	8	5	6	7
D	9	8	5	A	1	B	7	6	F	2	3	0	E	C	4

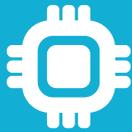
Der Rechtsweg ist ausgeschlossen. Mitarbeiter der in der Unternehmensgruppe Elektor International Media B.V. zusammengeschlossenen Verlage und deren Angehörige sind von der Teilnahme ausgeschlossen.



AISLER

# Dein Elektronik Projekt schnell und günstig gefertigt

Bestell deine erste Leiterplatte bei AISLER: erhalte 30 € Rabatt!



Beautiful  
Boards



Stellar  
Stencil



Amazing  
Assembly

## Bestell jetzt deine Platine

Und

# Gewinne

einen Voltera V-One  
Desktop PCB Printer  
(i.W.v. 4.150 €)



RABATT-CODE

**ELCCBBL**

Gültig bis 30. Juni 2022



**JETZT BESTELLEN!**

[www.elektormagazine.de/voltera-platinen-drucker-gewinnen](http://www.elektormagazine.de/voltera-platinen-drucker-gewinnen)

# Treten Sie jetzt der Elektor Community bei!

Jetzt Mitglied werden!



- ✓ Komplettes Webarchiv ab 1970
- ✓ 6x Elektor Doppelheft (Print)
- ✓ 9x Digital (PDF) inkl. Elektor Industry
- ✓ 10% Rabatt im Online-Shop und exklusive Angebote
- ✓ Elektor Jahrgangs-DVD

- ✓ Mit Tausenden von Mitgliedern des Online-Labors gemeinsam entwickeln mit Zugang zu über 1.000 Gerber-Dateien und direktem Kontakt zu unseren Experten!
- ✓ Veröffentlichen Sie Ihr eigenes Projekt oder verkaufen Sie direkt über unseren Shop!

## Auch erhältlich

Die digitale  
Mitgliedschaft!



- ✓ Zugang zu unserem Webarchiv
- ✓ 10% Rabatt in unserem Online-Shop
- ✓ 6x Elektor Doppelheft (PDF)
- ✓ Exklusive Angebote
- ✓ Zugang zu über 1.000 Gerber-Dateien



[www.elektor.de/mitglied](http://www.elektor.de/mitglied)