



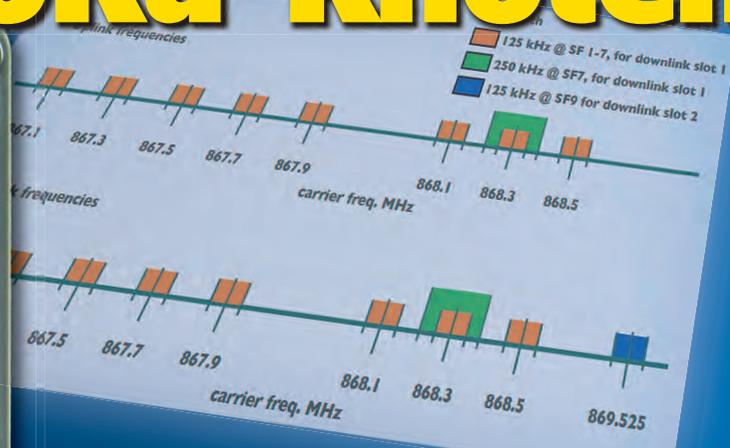
# elektor

DESIGN ✂ SHARE ✂ SELL **ELECTRONICS**

Nr. 572 - März/April 2020

## LoRa™

# Der Elektor-LoRa-Knoten



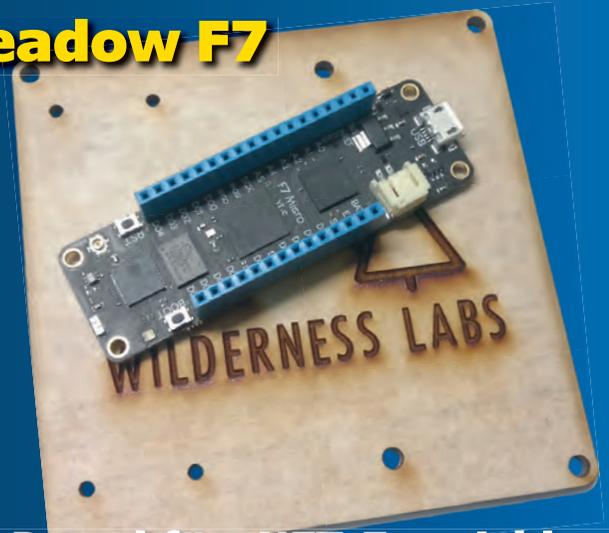
## Vielseitige 868-MHz-Fernbedienung mit großer Reichweite und Zustandsrückmeldung

## ESP32-Türklingel



### mit Telegram Messenger

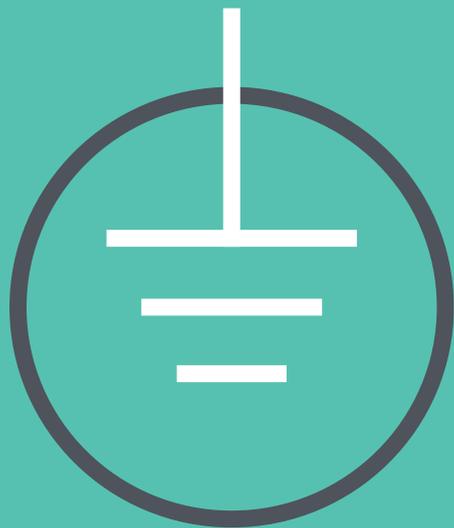
## Meadow F7



### Ein Board für .NET-Entwickler

Analoge Elektronik: Vorverstärker für MEMS-Mikrofone ✂ My IoT-Button: Der Knopf fürs Netz ✂ BASIC für ESP32/ESP8266 ✂ Kleine Schaltungen ✂ LoRaWAN - ein einfacher Einstieg ✂ Interview: Wienke Giezeman, The Things Network ✂ ESP32-Multitasking: Task-Prioritäten ✂ Sigfox: Erste Schritte im Netz ✂ Umweltdaten messen mit Raspberry Pi ✂ Review: Mikroskop Andonstar AD407 ✂ Optischer Tastkopf für Oszilloskope ✂ Und vieles mehr!





# UNSERE PREISE DER BESTE SCHUTZ VOR HOHEN KOSTEN

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

## Mit reichelt holen Sie mehr aus Ihrem Budget.

Dank effizienter, selbstentwickelter Logistik und IT und der Bündelung unsere Einkaufspower auf ausgesuchte Qualitätsprodukte, liefern wir Kleinmengen zu Top-Preisen.

### Labornetzgerät mit USB-Ausgängen

0 - 60 V, 0 - 6 A

Für Ausbildungs- und Entwicklungszwecke bietet dieses Labornetzgerät zwei schaltbare USB-Ausgänge mit eigener LCD-Anzeige für Spannungs- und Stromwerte.

- maximale Dauerleistung: 150 W
- C.C. (Konstantstrom) und C.V. (Konstantspannung) Modi
- Schutz gegen Kurzschluss, Überlast und Übertemperatur

**NEU**

Bestell-Nr.: **PEAKTECH 6227**

**109,00**  
(91,60)



**PeakTech®**



### Abisolierzange, Secura 2K

für Leiter und Litzen von 0,2 - 6 mm<sup>2</sup>

- für ein- und mehrdrätige Leiter
- automatische Einstellung auf den Kabelquerschnitt
- integrierter Seitenschneider bis 2,5 mm<sup>2</sup>
- verstellbarer Längenanschlag



Bestellnr.:  
**JOK 20 100 29,95** (25,17)

### Selbstnivellierender Kreuzlinienlaser

mit bis zu 30 m Reichweite

Projiziert exakt waagerechte und senkrechte Linien bei jedem Bau- oder Renovierungsprojekt.

- Genauigkeit  $\pm 3$  mm/10 m
- Batterie-Laufzeit bis 12 Stunden (inkl. 2x AA (Mignon) Batterien)



**NEU**

Bestellnr.:  
**UT LM570R-1 49,95** (41,97)

**UNI-T®**

- Top Preis-Leistungs-Verhältnis
- über 110.000 ausgesuchte Produkte

- Zuverlässige Lieferung – aus Deutschland in alle Welt.

Bestellservice: +49 (0)4422 955-333

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

**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: 3. 2. 2020

## Impressum

51. Jahrgang, Nr. 572  
März/April 2020

Erscheinungsweise: 9x jährlich  
(6x Elektor-Doppelheft + 3x Elektor Industry Magazin)

## Verlag

Elektor-Verlag GmbH  
Kackertstraße 10  
52072 Aachen  
Tel. 0241 95509190

Technische Fragen bitten wir per E-Mail an  
redaktion@elektor.de zu richten.

## Hauptsitz des Verlags

Elektor International Media  
Postbus 11, 6114 ZG Susteren  
Niederlande

## Anzeigen

Margriet Debeij (verantwortlich)  
Tel. 0241 95509174  
Mobil: +49 170 5505396  
E-Mail: margriet.debeij@elektor.com

Büsra Kas

Tel. 0241 95509178  
E-Mail: busra.kas@elektor.com

Es gilt die Anzeigenpreisliste ab 01.01.2020.

## Distribution

IPS Pressevertrieb GmbH  
Postfach 12 11, 53334 Meckenheim  
Tel. 02225 88010  
Fax 02225 8801199

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.

© 2020 elektor international media b.v.  
Druck: Pijper Media, Groningen (NL)  
ISSN 0932-5468



# Daten funken leicht gemacht

In der letzten November-Ausgabe hatte ich bereits erwähnt, dass unser Labor an Projekten arbeitet, die das Funk-Protokoll LoRa nutzen. Hiermit kann man Daten über weite Strecken besonders stromsparend übertragen. In diesem Heft finden Sie gleich zwei Einstiege in das spannende Thema. Der Elektor-LoRa-Knoten ist ein flexibel konfigurierbares Modul, mit dem Sie eigene Projekte „lora-fähig“ machen können. Als Beispiel-Anwendung – die natürlich auch 1:1 nachgebaut werden kann – haben wir uns eine Fernbedienung ausgedacht, mit der 230-V-Verbraucher über Entfernungen von mehreren Hundert Metern geschaltet werden können. Wie es sich für LoRa gehört, kann so ein Knoten autark im Feld (mit Li-Ion-Zellen) betrieben werden.

Wer Sensordaten ins Internet senden will, um sie überall auf der Welt abrufbar zu machen, sollte sich unseren LoRaWAN-Artikel anschauen. Hier beschreiben wir, wie man mit minimalem Aufwand ein Board mit dem offenen *The Things Network* verbindet. Man kann dabei entweder ein bereits vorhandenes Gateway in der Umgebung nutzen, oder ein eigenes, günstiges Gateway installieren, das die Sensordaten über LoRa entgegennimmt.

Natürlich gibt es in der Welt der Mikrocontroller noch viele weitere interessante Entwicklungen. Eine davon lautet Künstliche Intelligenz, und Sie dürfen sicher sein, dass meine Kollegen und ich auf der Messe *Embedded World* einige Neuheiten entdecken, über die wir dieses Jahr noch berichten werden. Schon jetzt in unserem Shop erhältlich ist das MAix BiT Kit, mit dem man (über die Arduino-IDE programmierbar!) eigene Experimente in Sachen Sprach- und Bilderkennung durchführen kann. Mehr darüber in den nächsten Heften und natürlich aktuell unter [www.elektormagazine.de](http://www.elektormagazine.de)

Jens Nickel

Chefredakteur Elektor

## Unser Team

Chefredakteur:	Jens Nickel (v.i.S.d.P.) (redaktion@elektor.de)
Redaktion:	Eric Bogers, Jan Buiting, Rolf Gerstendorf, Denis Meyer, Dr. Thomas Scherer, Clemens Valens
Leserservice:	Ralf Schmiedel
Elektor-Labor:	Mathias Claußen, Ton Giesberts, Hedwig Hennekens, Luc Lemmens, Clemens Valens, Jan Visser
Grafik & Layout:	Giel Dols



# Inhalt

51. Jahrgang – Nr. 572  
März/April 2020

# 6



## LoRa™



## Rubriken

- 3 **Impressum**
- 16 **Interaktiv**  
Korrekturen & Updates || Fragen & Antworten
- 59 **electronica Fast Forward 2020:**  
Die Startup-Plattform powered by Elektor
- 68 **Startup-Zone**  
Der vielleicht erfolgreichste Startup-Beschleuniger Europas?
- 74 **Aus dem Leben gegriffen**  
Bauteile bestellen in der Ukraine und Russland
- 78 **Mein Labor/Projekt**  
Ein Blick ins Allerheiligste aller Elektroniker
- 87 **Bemerkenswerte Bauteile**  
Die TMS1000-Mikrocontroller-Reihe
- 90 **Von Entwicklern für Entwickler**  
Tipps & Tricks, Best Practice und andere nützliche Infos
- 104 **Retronik**  
Intelektor: Der Elektor-Schachcomputer (1981)
- 110 **Jenseits der Elektronik**  
PCB Art – Über die Grenzen der industriellen Fertigung hinaus
- 114 **Hexadoku**  
Sudoku für Elektroniker



## Hintergrund

- 18 **Entwicklung analoger Elektronik**  
Fall Nr. 1 – Teil 2:  
Vorverstärker für MEMS-Mikrofone
- 21 **My IoT-Button: Der Knopf fürs Netz**  
Teil 1: IoT-Architektur
- 26 **BASIC für ESP32/ESP8266**  
Programmierung mit Annex WiFi RDS

## ESP32-Türklingel mit Telegram-Nachricht

29

### Wenn der Postbote nur einmal klingelt...

- 46 **Ein starker Verfechter des Offenen Internets**  
Interview mit Wienke Giezeman, Gründer von The Things Network
- 49 **Meadow F7**  
Ein Board für .NET-Entwickler
- 54 **Praktisches ESP32-Multitasking**  
Teil 2: Task-Prioritäten
- 67 **Raspberry Pi - Bash-Befehle in der Übersicht**
- 71 **RPi-HAT Enviro+**  
Umweltdaten messen mit Raspberry Pi und der HAT Enviro+

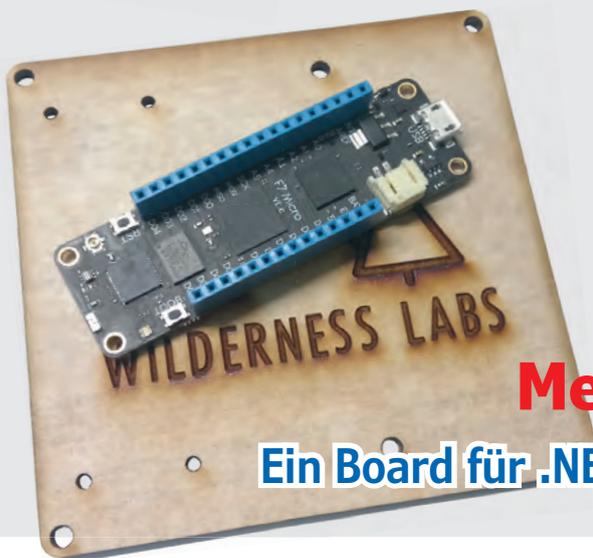


**Interview mit**  
**Wienke Giezeman,**  
**Gründer von**  
**The Things Network**

# Der Elektor-LoRa-Knoten



**Vielseitige 868-MHz-Fernbedienung mit großer Reichweite und Zustandsrückmeldung**



49

**Meadow F7**

**Ein Board für .NET-Entwickler**



## Projekte

- 6 Der Elektor-LoRa-Knoten**  
Vielseitige 868-MHz-Fernbedienung mit großer Reichweite, Zustandsrückmeldung und einem STM32 im Inneren
- 29 ESP32-Türklingel mit Telegram-Nachricht**  
Wenn der Postbote nur einmal klingelt...
- 34 Kleine Schaltungen**  
Neues aus der Elektor-Ideenkiste
- 36 LoRaWAN - ein einfacher Einstieg**  
Mit Blue Pill, LoRa-Breakout-Board und The Things Network
- 60 Mit dem Fuchs ins IoT (3)**  
Erste Schritte im Netz
- 80 Optischer Tastkopf für Oszilloskope**  
Helligkeitsschwankungen von Beleuchtungen messen
- 82 Das TABULA-Projekt - ein Update**  
Tangibles mit User-Feedback



## Vorschau

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

### Aus dem Inhalt:

- Wetterstation 2020
- Experimente mit KI
- Nixie Bargraph Thermometer 2.0
- Sigfox-Daten visualisieren
- E-Scooter aufgeschraubt
- Balancierender Roboter mit Arduino
- Basic für ESP32: Anwendung
- Triac und ATmega
- FreeRTOS und ESP32: Timer
- ARM A7 und M4 in einem Chip

### Und vieles mehr!

Änderungen vorbehalten.

Elektor Mai/Juni 2020 erscheint am 7. Mai 2020.

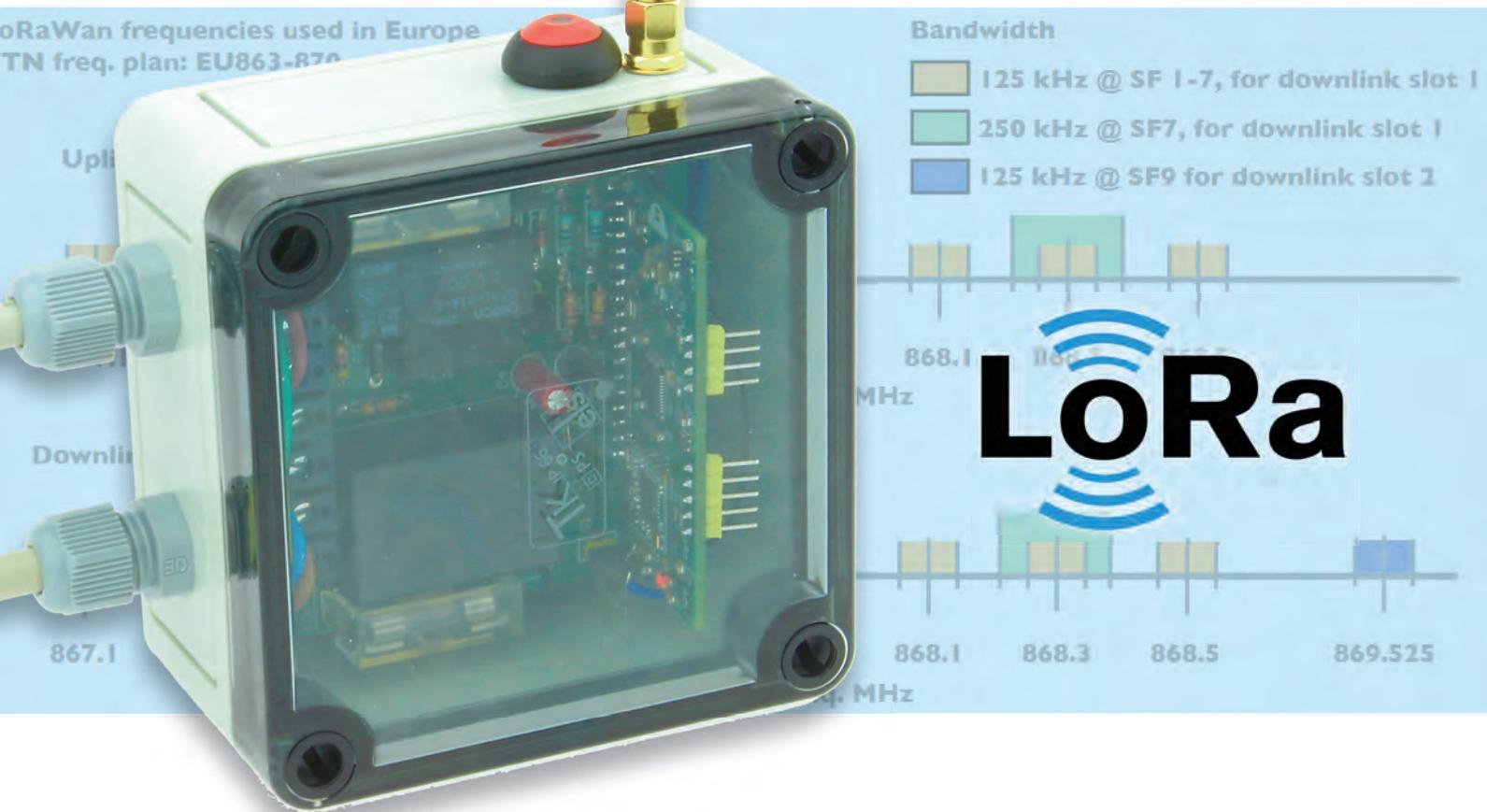


46

- 76 Mikroskop Andonstar AD407**
- 84 Gewusst wie**  
Berechnung des prospektiven Kurzschlussstroms
- 88 Aller Anfang...**  
Basiskurs für Einsteiger
- 93 Leitungen aufspüren**  
Über das Suchen und Finden verdeckt verlegter, schadhafter Leitungen
- 98 Portables 3-in-1-Oszilloskop Joy-IT DMS02D72**
- 100 Selbstbau-PC fürs Elektronik-Labor**  
Tipps für Komponentenauswahl und Bau

# Der Elektor-LoRa-Knoten

Vielseitige 868-MHz-Fernbedienung mit großer Reichweite, Zustandsrückmeldung und einem STM32 im Inneren



Von **Mathias Claußen** und **Luc Lemmens** (Elektor-Labor)

LoRa ist nur etwas für die Profis? Mitnichten! Das aus drei Bestandteilen aufgebaute LoRa-Projekt, das Sie zu einer zuverlässigen Ein/Aus-Steuerung mit Status-Feedback leicht aus- und nachbauen können, arbeitet im 433-MHz-ISM-LowPower-Funk über Entfernungen, die 10...20 Mal so groß sind wie die eines gewöhnlichen Consumer-WLANs.

Der Elektor-LoRa-Knoten wurde aus der Idee geboren, eine kompakte Schaltung mit Versorgung durch wiederaufladbare und nicht wiederaufladbare Batterien, einem handelsüblichen Mikrocontroller wie dem STM32 im LQFP48-Gehäuse und

einem LoRa-Modul vom Typ RFM95 zu entwickeln. Außerdem sollte die Platine in ein Gehäuse passen, das bei einem der größeren Distributoren gut erhältlich ist, und auch für den STM32-Compiler... **STOPP!**

Da das Gehäuse sowohl den zur Verfügung stehenden Platz als auch die Art der Energieversorgung mit bestimmt, müssen wir zunächst das Pflichtenheft diskutieren. Die ursprüngliche Idee war, das Gerät mit nicht wiederaufladbaren Batterien zu betreiben. Da wir aber bei Elektor den Beitrag zum Batteriemüllberg so niedrig wie möglich halten möchten, haben wir nach einer Lösung mit einer wiederaufladbaren Batterie gesucht, die durch den Benutzer leicht ausgetauscht werden kann. Dies würde zu einem kleinen, gebrauchsfähigen, spritzwassergeschützten Knoten führen, der nicht nur unter den komfortablen Bedingungen des Elektor-Labors, sondern auch

„draußen in der Wildnis“ arbeiten und reale Umweltdaten übermitteln könnte. Weitere Aspekte, die während der Entwicklungsphase des Projekts heftig diskutiert wurden, waren die Batterielebensdauer und die Sensoren, die man auf oder direkt am Board unterbringen können soll. Unser Ziel war natürlich eine flexible und nützliche Plattform zur Erforschung von LoRa und möglicherweise LoRaWAN, nicht für den „akademischen“ Gebrauch im Labor mit Klimaanlage und Teppichboden, sondern für den Einsatz an jedem (rauen) Ort dort draußen, um Geräte ferngesteuert ein- und auszuschalten. Mit anderen Worten, verlangt in Hinsicht auf Flexibilität, Stromverbrauch und Größe war nichts Geringeres als die Quadratur des Kreises.

Lesen Sie hier, wie wir das gemacht haben. Die folgende Geschichte ist chronologisch gehalten, da wir mit diesem Projekt nicht nur das Endprodukt und eine qualitativ hochwertige Abhandlung mit Ihnen teilen möchten, sondern auf Wunsch vieler Leser auch über all die realen Probleme, auf die wir gestoßen sind, und über die Werkzeuge zur Entwicklung des Projekts berichten.

### Drei Bestandteile, vier Platinen

In diesem Artikel geht es wie schon gesagt um drei Bestandteile:

- **LoRa-Knoten** (Platine 180516-1), die LoRaWAN-Knoten-Experimentierplattform für „lokales“ LoRa. Für dieses Projekt sind zwei Exemplare erforderlich.
- **LoRa-Button** (Platine 180666-2)
- **LoRa-Wechselstromschalter** (Platine 180666-1)

Zusammen bilden die drei Elemente nicht nur eine Fernsteuerung mit einer Reichweite, die 10- bis 20-mal größer ist als die einer Consumer-WLAN-Verbindung bei 2,4 GHz, sondern auch eine LoRa-Entwicklungsplattform mit Potenzial für das LoRaWAN. Beginnen wir mit dem Hauptelement, dem LoRa-Knoten.

### Die frühen Anfänge

Das Platinenlayout wurde mit KiCad entworfen, einem kostenlosen, quelloffenen Multipattform-Elektronikdesign-Programm, das nicht nur unter Hobbyanwendern täglich neue Nutzer findet. Wenn Sie KiCad kennenlernen und damit arbeiten möchten, haben wir ein schönes Buch in petto, das heißt, im Elektor Store [1].

Nach einem Fehlstart mit „einem anderen“ Prozessor und weiteren Stolpersteinen im Dunkeln wurde eine Platine für den STM32F072 im LQFP48-Gehäuse entworfen, der von seiner Kompatibilität mit der Arduino-IDE profitiert. Dies macht die Entwicklung ziemlich einfach, da wir und Sie die meisten der bekannten Arduino-Bibliotheken verwenden können. Da wir keinen Quarz auf der Platine vorgesehen hatten, griffen wir auf die Option des internen Oszillators zurück. Als Standardfrequenz für die MCU wurden 8 MHz voreingestellt, um den Stromverbrauch gering zu halten, aber wir können die Taktrate auf Wunsch bis zu 48 MHz steigern.

### Ein modularer Ansatz

Die in **Bild 1** dargestellte (frühe Version der) Elektor LoRa-Knoten-Platine ist nicht nur die Grundlage für das hier beschriebene Projekt, sondern wird auch in Spin-off-Projekten verwendet, die im Laufe dieses Jahres in Elektor veröffentlicht werden sollen. Begeben wir uns daher zunächst auf einen Rundgang durch die Bauteile auf der Platine und die Optionen, die zusätzlich auf dem Board realisiert werden können. Die Hardcore-Elektroniker unter Ihnen möchten vielleicht von dem Schaltplan in **Bild 2** auf diesem Rundgang begleitet werden. Um bestimmte Entwicklungsaspekte und Anforderungen zu erfüllen, haben wir uns daran gewöhnt, modular zu denken. Mit der vollständigen Bestückung aller

## INFOS ZUM PROJEKT

LoRa

Arduino

STM32

Einsteiger

→ Fortgeschrittene

Experte

circa 4 Stunden

USB-UART,  
Arduino-IDE,  
Labor-Werkzeugsatz

etwa 150 € ohne Gehäuse

Bauteile hätten Sie ein LoRaWAN-Gerät mit einem GPS-Modul, einem Krypto-Coprozessor, einem SPI-F-RAM oder einem Flash-Speicher, alles Funktionen „nice to have“, aber für Sie vielleicht verzichtbar und deshalb herausgeschmissenes Geld. Da die Pinbelegung der SPI F-RAM- und SPI Flash-Module standardisiert ist, können Sie Ihr Lieblingsmodul hinzufügen. Dasselbe gilt für den Krypto-Coprozessor, einen ATECC608a: Sie können auch nur ein I<sup>2</sup>C-EEPROM- oder einen F-RAM-Chip hinzufügen. Ihnen steht also

## Eigenschaften

### LoRa-Knoten

- LoRaWAN-Experimentierplattform-Platine mit minimaler Bestückung
- Li-Ion-Batterien, Zellen durch Benutzer austauschbar
- STM32F072C8T6TR ARM-Cortex-M0-MCU
- SPI F-RAM oder Flash (optional)
- Krypto-Coprozessor (optional)
- GPS-Modul (optional)
- USB-Schnittstelle (optional)

### LoRa AC-Schalter (Slave)

- Rückmeldung des Relaiszustands an den Zentralknoten
- Schaltkontakte bis 5 A (1000 W bei 230 VAC)

### LoRa-Button (Master)

- Low-energy-Entwurf
- Optionales OLED-Display
- Integrierte Spulenantenne
- Reichweite 100...500 m zum LoRa-Slave

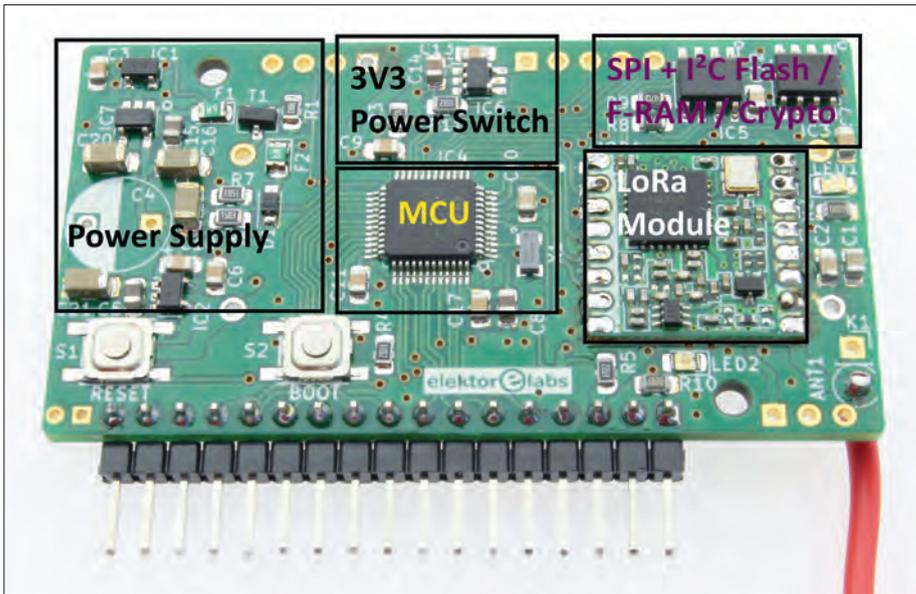


Bild 1. Die Platine des LoRa-Knotens mit ihren funktionalen Blöcken. Das Foto zeigt eine minimal bestückte LoRaWAN-Knoten-Experimentierplattform.

ein sehr flexibles und benutzerkonfigurierbares Gerät zur Verfügung.

Die MCU STM32F072 im LQFP48-Gehäuse auf der Platine ist ein Cortex-M0-basierter Controller mit 64 kB Flash und 16 kB RAM und ermöglicht Firmware-Updates (oder die Programmierung eigener Anwendungen) über USB.

Das Schöne an der STM32-Serie ist die absolute Pin-Kompatibilität. Sie können die MCU beliebig wechseln, ohne das Layout der Platine zu ändern. Zum Beispiel lässt sich der Spar-STM32L072 mit noch geringerer Leistung einsetzen oder ein STM32L151, falls Sie mehr Rechenleistung und Flash-Speicher benötigen. Der STM32F103, den Sie üblicherweise auf der Bluepill-Platine finden, kann ebenfalls auf der Platine montiert werden.

Der Kasten **Eigenschaften** ist nur ein Versuch, die Vielseitigkeit des Boards aufzuzeigen.

### Platinentwurf und Herausforderungen

Nach dem Bau der ersten Prototypen schienen einige Änderungen angebracht zu sein. Für das GPS-Modul brauchten wir für einen besseren Empfang einige Anpassungen. Sicherlich kann man eine miniaturisierte Antenne bauen, aber man muss auch die Massefläche berücksichtigen. Das Datenblatt des GPS-Moduls gibt eine Mindestgröße der Platine an, damit die Antenne funktioniert. Alles berücksichtigt, dennoch waren wir mit

dem Signalempfang nicht zufrieden.

Nach der ersten Platinenversion haben wir das GPS-Modul Quectel L96 in die Ecke gelegt und es durch das GPS-Modul aus dem Elektor-Store [2] ersetzt, da dieses sowohl billiger als auch einfacher als quasi-externes Bauteil an die Platine anzuschließen war. Dies führte zu weiteren Redesigns der verwendeten Steckverbinder, alle in einem 0,1-Zoll-Raster, so dass man das Modul auf ein Breadboard stecken kann.

Das LoRa-Modul RFM95W benötigt auf jeden Fall eine Antenne. Bei der Größe und dem gewählten Gehäuse wollten wir eine selbstgebaute interne Spulenantenne für 868 MHz einsetzen. Wenn Sie einen noch besseren Empfang wünschen, ist eine externe Antenne eine Option. Wir erkannten so schnell, dass das Design nur ein Kompromiss zwischen Platz, Funktionalität und HF-Leistung sein kann. Lektion gelernt!

Wir haben bereits erwähnt, dass Li-Ionen-Batterien im 10440-Format (AAA) zur Spannungsversorgung benutzt werden, Typen, wie sie auch in E-Zigaretten zu finden sind. Diese Batterien lassen sich vom Benutzer leicht wechseln und können außerhalb des Geräts mit einem geeigneten Li-Ion-Ladegerät aufgeladen werden. Um die Zellen vor Tiefentladung zu schützen, wurde ein Spannungsüberwachungs-IC eingebaut, das im Falle eines Falles den LDO für die 3,3-V-Versorgung abschaltet.

Außerdem wurde an dieser Stelle ein

2-mm-Loch hinzugefügt, um LiPo-Akkupacks, die üblicherweise für Drohnen verwendet werden, anbringen zu können. Diese kleinen und flachen Zellenpacks werden mit Molex-510005- oder JST-Steckverbinder geliefert und bieten ein tolles Kapazitäts-/Platzverhältnis. Dies macht das gesamte Design flexibler und ermöglicht die Verwendung von Standard-Akkus, die bei großen Versandhändlern wie Amazon bezogen werden können.

Zum Gegenstromschutz haben wir uns für eine „Ideale Diode“ MAX40200 entschieden, die die Versorgungsspannung abschaltet, wenn ein bestimmter Rückstrom durch sie fließt. Eine zweite Diode wurde in die Stromversorgung des GPS-Moduls (K4) aufgenommen, wodurch die MCU die Stromversorgung des Moduls unterbrechen kann, um die Standzeit der Batterie zu verlängern. Wenn Sie auf diese Weise das GPS-Modul verlieren, kann die MCU die 3,3-V-Schiene zur Stromversorgung anderer externer Schaltungen offen halten (oder bei Bedarf auch deaktivieren).

Die Stromversorgung des Geräts mit NiMH-Zellen wäre prinzipiell auch machbar, allerdings wäre ein zusätzlicher Schaltkreis wie ein DC/DC-Aufwärtswandler erforderlich, dessen Boost-Stufe die ganze Zeit Strom verbraucht. Dies verkürzt wiederum die Batteriestandzeit; und über einen Unterspannungsbegrenzer müsste man ebenfalls nachdenken. Da unsere selbstgefertigte Antenne gut sendete, schienen zu diesem Zeitpunkt die größten Herausforderungen in der Projektsoftware zu liegen. Frohgemut ließen wir die Punkt-zu-Punkt-Kommunikation im RAW-Modus der LMIC-Bibliothek laufen.

Doch dann traten auf der Hardware-Ebene einige Probleme auf, darunter eines mit einem 470-µF-Kondensator, der zuvor vor IC1 lag. Wenn der Eingang unter 2,97 V liegt, deaktiviert er den HS-Schalter IC7 und die Versorgungsspannung wird für eine Zeitspanne von 20 ms unterbrochen.

Ohne einen großen Pufferkondensator stören hohe Ströme beim Einschalten den UVLO und seine Eingangsspannung fällt unter 2,97 V, was zu einer permanenten Schleife aus Aktivieren und Deaktivieren des LDOs führt. Mit einem zusätzlichem „dicken“ Elko C4 (**Bild 3**) war das Problem beseitigt und der Strompfad funktionierte wie erwartet. Bringen Sie

diesen Kondensator nur dann an, wenn Ihnen die beiden 100-µF-MLCC-Kondensatoren zu teuer sind. Gemäß Datenblatt

haben wir auch keinen FET oder eine Diode vorgesehen, die die Batterien vor dem Aufladen über  $V_{USB}$  schützt. Diese

Aufgabe wird innerhalb des MAX4020 erledigt - wenn ein Rückwärtsstrom fließt, unterbricht der Chip die Stromzu-

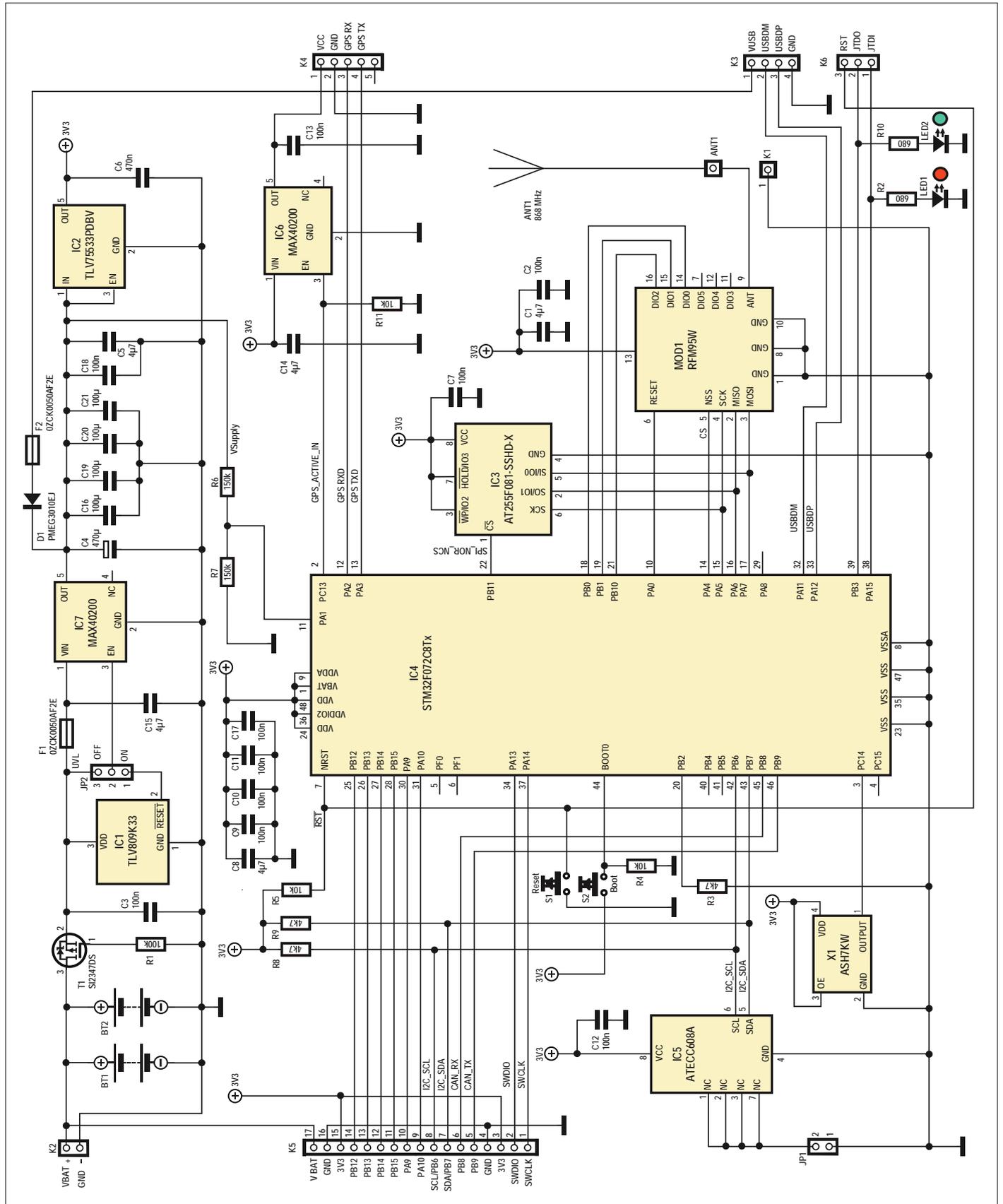


Bild 2. Schaltplan der LoRa-Knoten-Steuerplatine. Es handelt sich im Wesentlichen um die LoRaWAN-Knoten-Experimentierplattform mit einer spezifischen Bestückung von ICs.

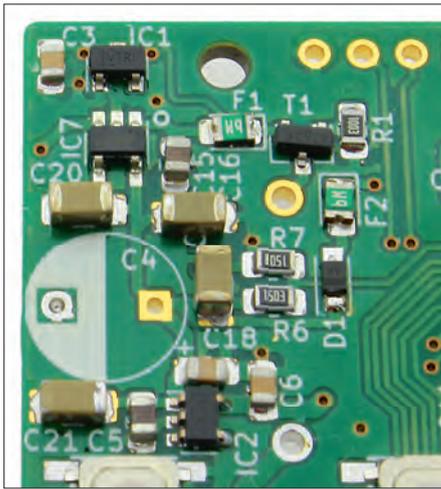


Bild 3. Der Footprint für den Elektrolytkondensator C4, des voluminösesten Bauteils auf der Platine.

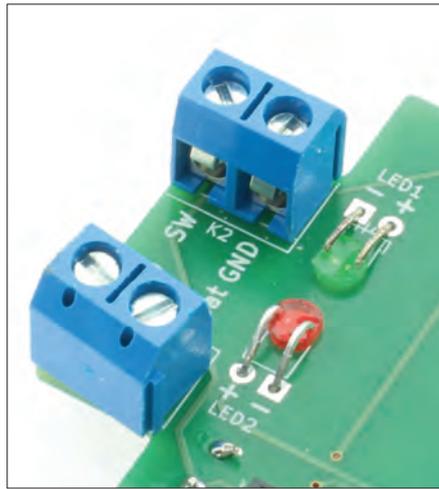


Bild 5. Die LEDs auf dem LoRa-Button schauen durch die Platine hindurch.

fuhr, auch, wenn die von  $V_{USB}$  kommende Spannung die von den Systembatterien bereitgestellte Spannung übersteigt. Die beiden LEDs auf der Platine zeigen den Board-Status an, wie er von der MCU auf den Portleitungen PB3 und PA15 gemeldet wird. Diese Informationen sind

auch als JTDO und JTDI auf der Stiftleiste K6 abrufbar.

### Software: Grundlagenarbeit

Obwohl Systemsoftware zunehmend die Funktionalität einer Hardware bestimmt, ist es für einige nur schwarze Magie, was

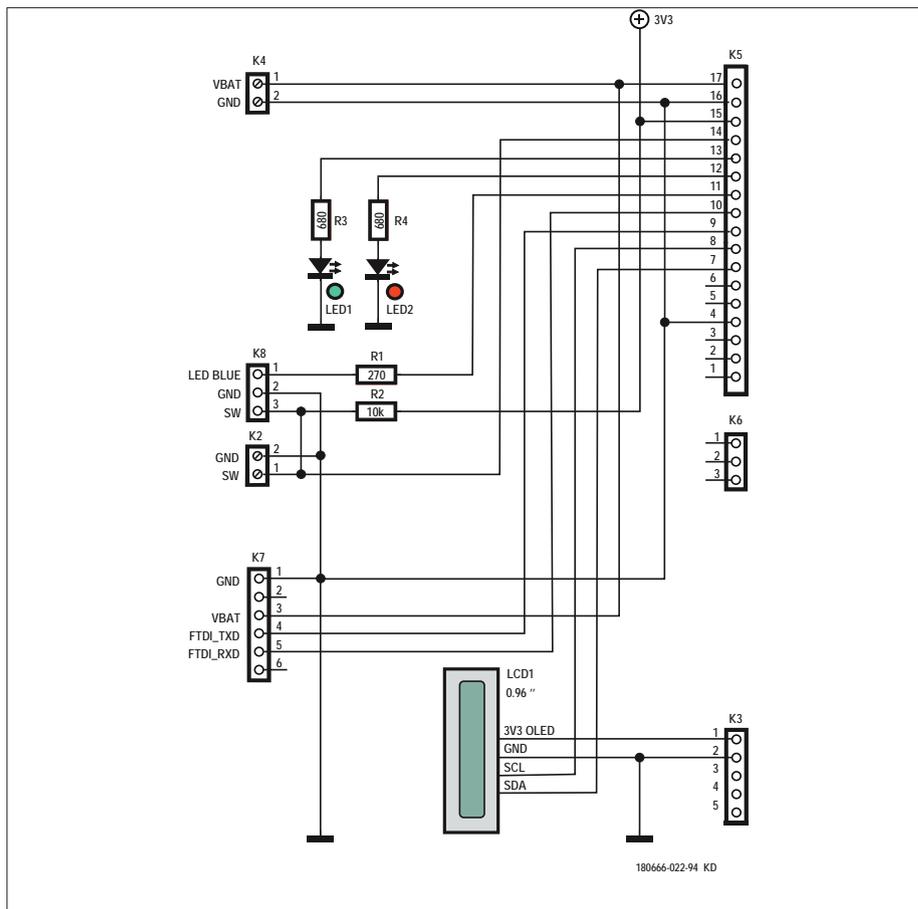


Bild 4. Schaltung des LoRa-Buttons. Dies ist der „sendende“ oder der „Master“-Teil des Elektor-LoRa-Knotens. Die OLED-Anzeige ist optional und sollte wegen ihres Energiebedarfs wohldurchdacht sein!

da in den Chip „geflasht“ werden muss. Ein Vergleich der Laborzeiten, die für Software und Hardware aufgewendet werden, zeigt, dass für die Software viel mehr Arbeitszeit aufgewendet werden muss. Wie kommt das?

Wie bereits erwähnt, können Sie das Board mit Ihrer Arduino-IDE programmieren. Dabei hat das *stm32duino*-Projekt [3] eine besonders breite und gute Unterstützung von STM32-MCUs und STM32-Boards zu bieten.

Das formelle Hinzufügen der Elektor-LoRa-Node-Experimentierplattform war die erste wirklich kritische Hürde, die es zu nehmen galt. Im Arduino-Core des STM32 mussten wir einige Dateien bearbeiten und die Board-Definition hinzufügen. Nach einigen Fehlern und unerwartetem Software-Verhalten brachten wir die Vorlage zum Laufen und konnten mit der Kodierung der ersten paar Zeilen beginnen.

Eine Sache, die normalerweise nicht viel ausmacht, ist die Art und Weise, wie die MCU den Takt erzeugt. Bei den älteren AVR-Chips hatte man einen Quarz und das war es dann auch schon, bei den neueren MCUs jedoch hat man mehr Möglichkeiten: Der Haupttakt der MCU kann aus internen PLL-/FLL-Quellen oder internen Hochgeschwindigkeits-RC-Oszillatoren stammen. Wir haben uns für die internen Taktquellen entschieden, die zwar nicht besonders frequenzstabil sind, aber gut genug, damit wir damit arbeiten konnten. Da wir die Taktgeschwindigkeit während der Laufzeit ändern können, spart „langsam“ Energie und „schnell“ führt zu einem höheren Durchsatz. Der Standardwert von 8 MHz für den Kern und die Peripherie hält den Stromverbrauch niedrig, der Takt ist aber doch schnell genug, um den LoRa-Stack zu verarbeiten.

Für die Genauigkeit im RTC-Bereich wurde ein 32,768-kHz-Uhrensoszillator hinzugefügt (X1). Er ist zwar optional, aber wenn man auf Kosten eines einzigen Mikro-Amperes UARTs erhält, die vom Quarz getaktet werden können, sowie eine RTC, die die MCU nach einer bestimmten Zeitspanne aufwecken kann, lässt man sich nicht zweimal bitten.

### Der LoRa-Button

Als ein einfaches Eingabegerät, das seinen Zustand über Funk übermittelt und über weitere Dinge informiert, kommt die zweite Komponente des Projekts zur Vorstellung, der LoRa-Button.



Bild 6. Mit einem scharfen Messer oder einem spitzen Seitenschneider wird dem Kunststoffkörper der Stiftleiste zu Leibe gerückt...

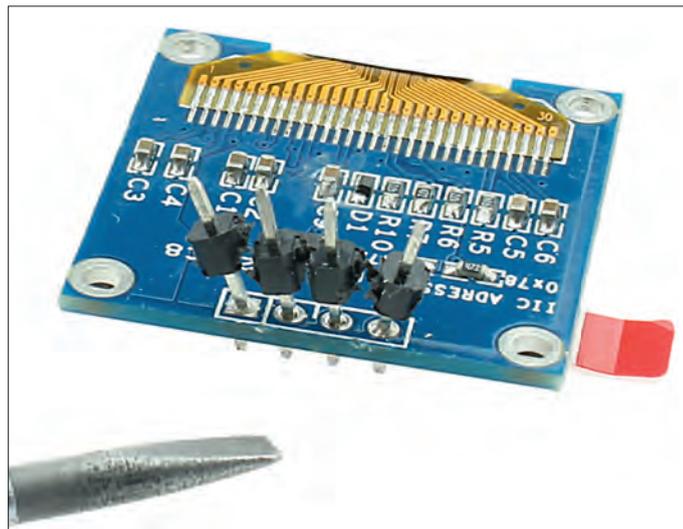


Bild 7. ...und die Einzelteile werden dann vorsichtig mit der flachen Schraubendreherklinge von den Metallstiften entfernt.

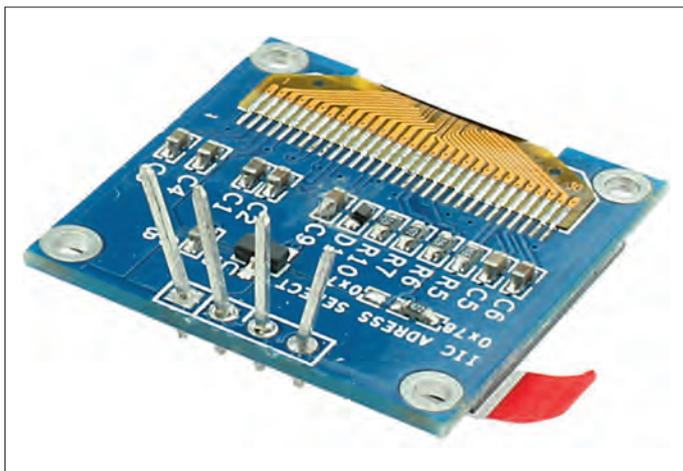


Bild 8. Nun sind die Metallstifte nackt und ein wenig geknickt.

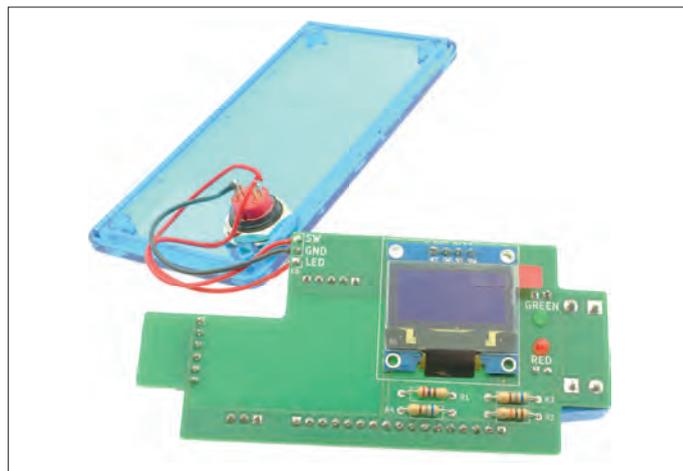


Bild 9. Die fertige Platine des LoRa-Buttons.

Die Schaltung in **Bild 4** zeigt nur einige wenige Widerstände, ein Paar LEDs, Platinenanschlussklemmen sowie Stift- und Buchsenleisten. Diese Teile sind für die Basisversion des Elektor-LoRa-Knotens unerlässlich. LCD1, die OLED-Anzeige aus dem Elektor-Store [3], ist dagegen optional.

Beim Zusammenbau der Platine sind einige Dinge zu beachten. Die LEDs zum Beispiel werden so umgebogen, dass sie durch die Löcher in der Platine auf die andere Seite schauen (**Bild 5**).

Für das OLED-Display ist ein wenig Feinmechanik angesagt: Man kann entweder die ganze Stiftleiste ablöten und bearbeiten oder „in situ“ die Kunst-

stoffummantelung um die Stifte der Leiste mit dem feinen Seitenschneider durchzwacken (**Bild 6**) und die einzelnen Kunststoffteile dann mit der Schraubendreherklinge langsam über die Stifte nach oben schieben (**Bild 7**). Wenn alles gut geht, sind die Metallstifte bald ganz freigelegt (**Bild 8**).

Ziel dieser Mühen ist es, dass das OLED-Display flach auf der vorgesehene Position liegt. Dann bringt man die Buchsenleisten, die gewinkelten Stiftleisten und die Platinen-Anschlussklemmen an. Damit ist die Konstruktion fast abgeschlossen, es fehlt nur noch die eigentliche Taste (**Bild 9**) und

### Zwischenspiel: eine DIY 868-MHz-Antenne

Der LoRa-Button benötigt eine Antenne zum Senden und Empfangen auf 868 MHz. Eine Viertelwellen-Antenne aus Kupferlackdraht mit 1 mm Durchmesser (AWG 18) ist eine gute und kostengünstige Option. Unter der Annahme, dass eine Massefläche vorhanden ist, sollte der Draht im Vakuum eine Länge von 8,635 cm haben, in der Praxis jedoch 8,2 cm! Das liegt an dem *Verkürzungsfaktor* von 0,95 [4]. Je nach dem verfügbaren Platz kann der Antennendraht wie bei einer ordentlichen Stabantenne gerade gelassen oder zu einer Spule gewickelt werden, dessen Wendel man

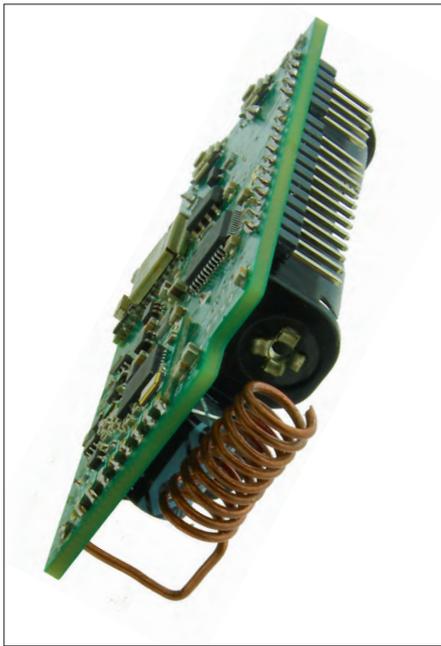


Bild 10. Der Fußpunkt der selbstgewickelten Viertelwellen-Spulenantenne für das 868-MHz-Band ist mit dem ANT-Pad auf der Platine des LoRa-Buttons verbunden (hier ein früher Prototyp).

auf 1 mm Abstand drückt. In beiden Fällen wird der Draht an den ANT-Anschluss gelötet (Bild 10).

### Der LoRa-Wechselstromschalter als Slave

Als Beispiel für ein slave-gesteuertes „Ausgabegerät“, das sowohl vom Elektor-LoRa-Knoten gesteuert wird

als auch auf den LoRa-Button „hört“, haben wir einen Leistungs-Netzschalter für Geräte mit einem Verbrauch von bis zu 1 kW entwickelt. Ein Optokoppler gibt den (Ein/Aus-)Zustand des Wechselstromschalter-Slaves an den LoRa-Knoten zurück. Auf der Platine befindet sich auch ein Spannungswandler, der die Netzwechselspannung unmittelbar auf 5 V<sub>DC</sub> reduzieren kann und es ermöglicht, dass LoRa-Knoten und Wechselstromschalter-Slave direkt aus der Steckdose versorgt werden.

Die eigentliche Intelligenz des LoRa-Wechselstromschalters steckt in Hardware und -Software des aufgesteckten LoRa-Knotens. Dennoch lässt sich über den Schalter-Slave einiges Interessantes mitteilen:

- 5-V<sub>DC</sub>-Versorgungsspannung
- Leistungsschaltung bis maximal 1 kW
- Rückmeldung des Schalterzustands
- lokaler Steuertaster (Ein/Aus).

Der Schaltplan des LoRa-Leistungsschalter-Slaves ist in Bild 11 dargestellt. Betrachten wir die wichtigsten Bauteile und den Aufbau.

### Stromversorgung

Der AC/DC-Wandler, der aus der Netzspannung 5 V erzeugt und 2 W liefern kann, stammt von der Firma MeanWell. Er erwies sich als eine kompakte, sichere und relativ kostengünstige, kurz gesagt, eine optimale Lösung für das Projekt.

Obwohl in der Dokumentation nichts von einer Sicherung auf der Wechselspannungsseite gesagt wird, haben wir doch F1 aus Sicherheitsgründen hinzugefügt. Der NTC-Widerstand R1 (negativer Temperaturkoeffizient) begrenzt den Einschaltstrom, der bei diesen Invertiern stark ansteigen kann. C1, C2 und Common-mode-Filter L1 am Ausgang des AC/DC-Wandlers mögen übertrieben erscheinen, aber wir haben uns nicht auf die Aussage von MeanWell verlassen, dass das Modul genügend interne EMI-Störunterdrückung enthält.

### Leistungsschalter

Für diese Anwendung wurde ein Relais aus der G5RL-K1-Reihe von Omron ausgewählt. Es handelt sich um ein bistabiles Relais mit einer Set- und einer Reset-Spule. Eine kurze Erregung einer der beiden Spulen reicht aus, um die Kontakte umzuschalten. Obwohl die Relaiskontakte für einen maximalen Strom von 16 A ausgelegt sind, schmilzt die Sicherung F2 schon bei etwa 5 A durch, um die Leiterbahnen auf der Platine vor dem Durchbrennen zu schützen.

### Rückmeldung

Theoretisch sollte der Mikrocontroller auf der LoRa-Knotenplatine in der Lage sein, sich an die letzte an das Slave-Gerät gesendete Aktion zu erinnern (ist der Schalter ein- oder ausgeschaltet?). Aber um wirklich sicher zu gehen, ist eine physikalische Rückmeldung des Ausgangsstatus kein überflüssiger Luxus, auch, weil die Position eines bistabilen Relais nicht definiert ist, wenn das ganze Projekt zurückgesetzt wird. Denken Sie daran: Die Schaltung um den Optokoppler IC1 zeigt nur an, dass RE1 geschlossen ist und dass die Netzspannung an den Klemmen von K2 anliegt. Ob die Last tatsächlich funktioniert (ob sie Strom verbraucht), wird durch die Zustandsrückmeldung nicht festgestellt und sollte daher real und vor Ort überprüft werden.

### Manueller Betrieb

Die Idee hinter diesem Projekt war es, eine Last aus einer (großen) Entfernung per Knopfdruck schalten zu können. Wenn man aber nahe dem Empfänger ist, wäre es toll, wenn man ihn auch manuell bedienen könnte, ganz ohne LoRa-Button und LoRa-Knoten. Und wenn die LoRa-Verbindung, aus welchem Grund auch immer, verloren geht, ist es zudem

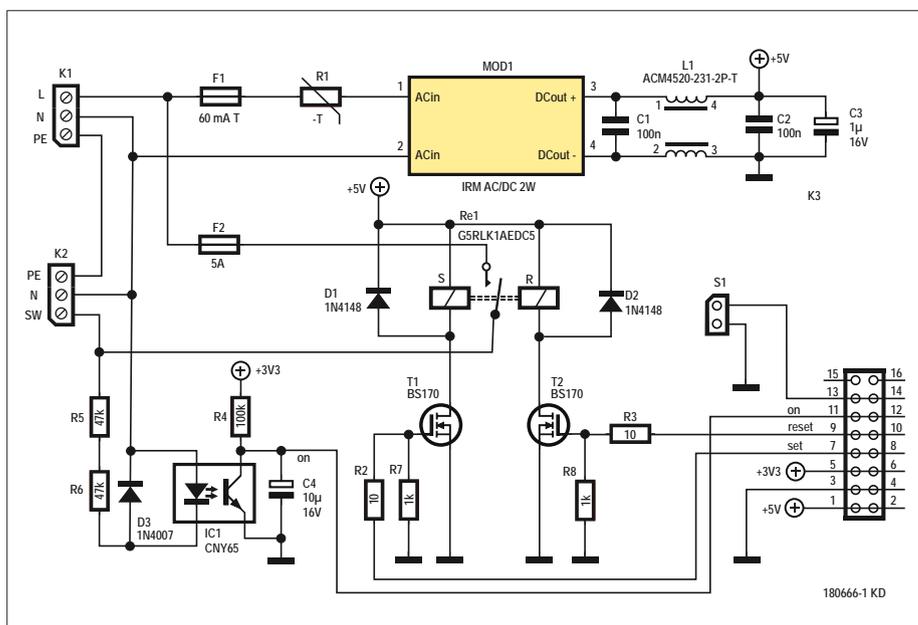


Bild 11. Schema des LoRa-Wechselstromschalters. Gar nicht so dumm, sendet der Schalter seinen tatsächlichen Ein-/Aus-Zustand an den LoRa-Knoten zurück.



## STÜCKLISTE

### Für die LoRaWAN-Experimentalplattform-Platine

Hinweis: 3 Stück, je für Schalter und Button, erforderlich

#### Widerstände:

(alle Dickschicht, 5%, 0,1 W, 150 V)  
R1 = 100 k  
R2, R10 = 680 Ω  
R3, R8, R9 = 4 k  
R4, R5, R11 = 10 k  
R6, R7 = 150 k

#### Kondensatoren:

C1, C5, C8, C14, C15 = 4 μF, 16 V, X7R, SMD 0805  
C2, C3, C7, C9, C10, C11, C12, C17, C18 = 100 n, 50 V, X7R, SMD 0805  
C4 = siehe Text  
C6 = 470 n, 25 V, X7R, SMD 0805  
C16 = nicht bestückt  
C20 = nicht bestückt  
C19, C21 = 100 μF, 10 V, SMD 1206 [3216 Metrisch], ±20%, X5R

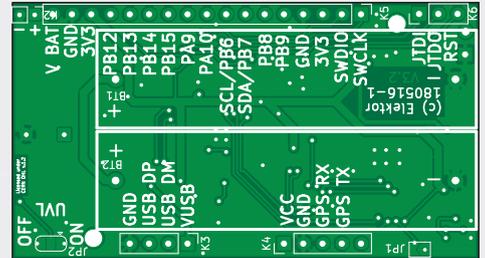
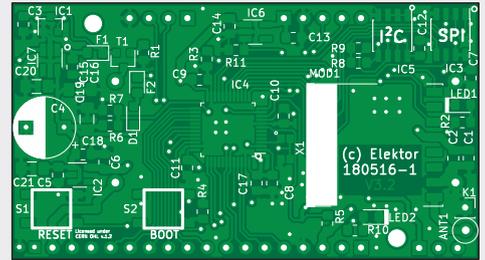
#### Halbleiter:

D1 = Diode PMEG3010EJ, 115, 30 V, 1 A  
LED1 = Low-current-LED, rot, SMD 0805  
LED2 = Low-current-LED, grün, SMD 0805  
T1 = SI2347DS, MOSFET, p-Kanal, 5 A, 30V, 0Q033

IC1 = TLV809K33DBVR, Spannungsüberwachung  
IC2 = TLV75533PDBVR, 3V3 LDO, 500 mA, niedriger I<sub>Q</sub>, SOT-23-5  
IC3 = Footprint für optionales AT25SF081-SSHD-T 8Mbit-Flash-IC  
IC4 = STM32F072C8T6TR ARM Cortex-M0 Mikrocontroller  
IC5 = Footprint für optionales ATECC608A-SSHDA-B, Krypto-Authentifizierung  
IC6, IC7 = MAX40200AUK+T „dealer Dioden“-Controller, 1-Kanal, 1 A  
MOD1 = RFM95W-868S2 LoRa-Sender/Empfänger, Elektor-Store SKU 18715

#### Außerdem:

X1 = Oszillatormodul 32,768 kHz, SMD, 3,2 mm x 1,5 mm (Abracon ASH7KW-32,768KHZ-L-T)  
S1, S2 = Tastschalter, 12V, 50 mA (Multicom TM-5531-Q-T/R)  
Bt1, Bt2 = Footprint für optionalen AAA-Batteriehalter mit Platinenpins (Multicom MP000341)  
F1, F2 = Rückstellbare PPTC-Sicherung 500 mA (Bel Fuse OZCK0050FF2E)  
K1 = Lötnagel



K2 = 2-polige Stiftleiste  
K3 = 1x4-polige SIL-Stiftleiste  
K4 = 5-polige Stiftleiste  
K5 = 17-polige SIL-Stiftleiste  
K6 = 3-polige Stiftleiste  
K5, K6, K3, K4 = 40-polige SIL-Stiftleiste  
ANT1 = Drahtantenne, 1 mm CuL, Länge 8,2 cm  
Platine 180516 V3.1 aus dem Elektor-Store



## STÜCKLISTE

### Für die LoRa-AC-Schaltplatine

#### Widerstände:

R1 = NTC, 120 Ω, Epcos Typ B57236S0121M000  
R2, R3 = 10 Ω, Kohlenstoff-Folie, 5%, 0,25 W, 250 V  
R4 = 100 k, Kohlenstoff-Folie, 5%, 0,25 W, 250 V  
R5, R6 = 47 k, Kohlenstoff-Film, 5%, 0,25 W, 250 V  
R7, R8 = 1 k, Kohlenstoff-Film, 5%, 0,25 W, 250 V

#### Induktivität:

L1 = ACM4520V-231-2P-T Common-mode-Filter (Farnell 2455201)

#### Kondensatoren:

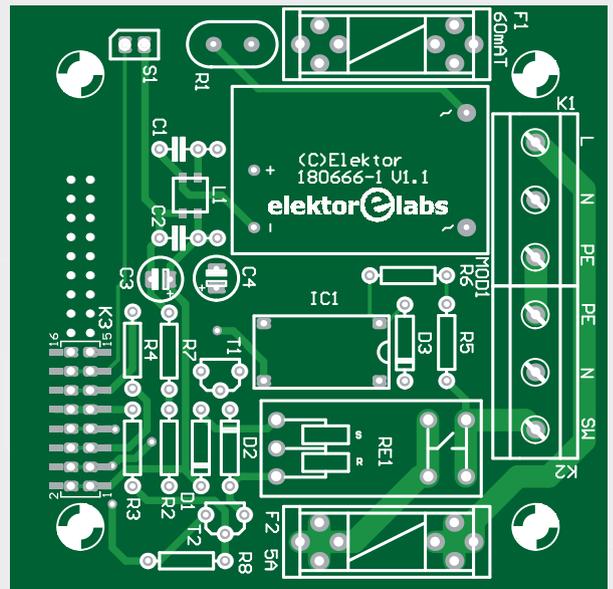
C1, C2 = 100 n, 50V, X7R, 0,2" (5,08 mm) Teilung  
C3 = 1 μF, 16 V, radial, 5 mm  
C4 = 10 μF, 16 V, radial, 5 mm

#### Halbleiter:

D1, D2 = 1N4148  
D3 = 1N4007  
T1, T2 = BS170  
IC1 = Optokoppler CNY65

#### Außerdem:

RE1 = G5RLK1AEDC5 Leistungsrelais, Zweifach-Spule mit Verriegelung, 5 V, 16 A, SPST (Omron)  
K1, K2 = 3-polige Platinenanschlussklemme, 0,3" (7,62 mm), 500V  
K3 = 2-reihiger Board-to-Board-Steckverbinder, 2,54 mm (0,1") Raster, 16 Kontakte, Buchse, Serie WR-PHD, SMD (Würth # 610316243021)  
S1 = Drucktasten, IP68-Klasse (Alcoswitch PB6B2FM3MICAL07)  
MOD1 = AC/DC-Stromversorgung für Platinenmontage 5 V, 200 mA (MeanWell IRM-02-5)



F1, F2 = Sicherungshalter, Platinenmontage, 5x20 mm  
F1 = 60-mAT-Sicherung, 20 mm  
F2 = 5-AT-Sicherung, 20 mm  
Platine 180666-1 V1.1 aus dem Elektor-Store  
Gehäuse: HAMMOND 1591-ATBU (Ice Blue)



## STÜCKLISTE

### Für den LoRa Button

#### Widerstände:

(alle Kohlenstoff-Folie, 5%, 0,25 W, 250 V)

R1 = 270  $\Omega$ , 5%, 0,25 W, 250 V

R2 = 10 k, 5%, 0,25 W, 250 V

R3,R4 = 680  $\Omega$ , 5%, 0,25 W, 250 V

#### Halbleiter:

LED2 = rot, 3 mm

LED1 = grün, 3 mm

#### Außerdem:

K2,K4 = 2-polige Platinenanschlussklemme, 0,2"-Raster (5,08 mm), 630 V

K3 = 1x5-polige Buchsenleiste, abbrechbar, vertikal

K5 = 1x17-polige Buchsenleiste, abbrechbar, vertikal (siehe Text)

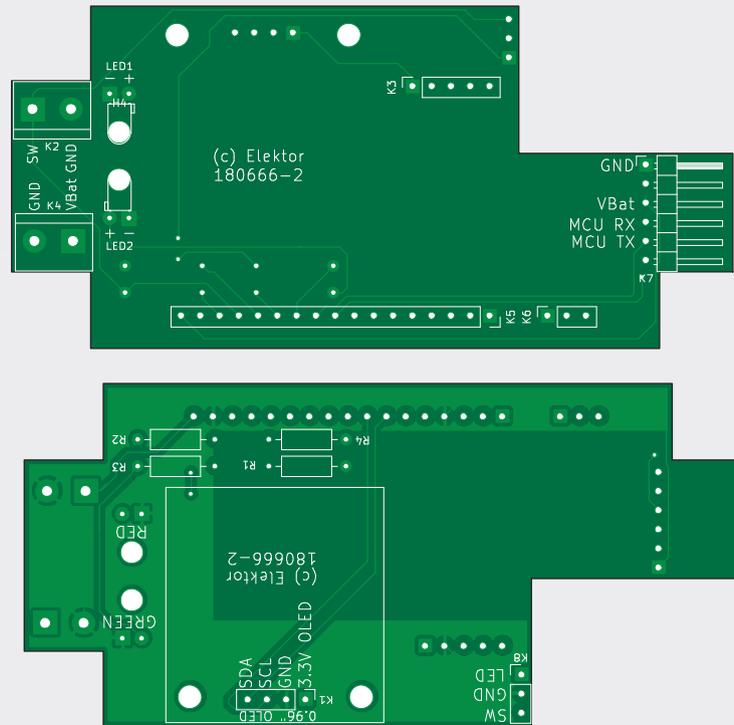
K6 = 1x3-polige Buchsenleiste, abbrechbar, vertikal

1 Schalter mit integrierter blauer LED

Blaue 0,96"-OLED-Anzeige, I<sup>2</sup>C, 4-polig (optional)

Platine 180666-2 V1.2 aus dem Elektor-Store

Gehäuse: Spelsberg Typ TK PS 94 x 94 x 57 mm, IP66



nützlich, wenn man auch dann die Last noch schalten kann.

### Montage und Installation

Die Lötarbeiten an der Platine sind wenig herausfordernd. Abgesehen vom Common-mode-Filter und dem (möglichen) Steckverbinder K3 haben wir es nur mit bedrahteten Bauteilen zu tun. Die Anschluss-Pads von K3 sind ziemlich groß, und auch L1 lässt sich problemlos anbringen.

K3 benötigt nur sieben der sechzehn Pins auf der Platine. Er gibt zwei Optionen, aber beide gehen davon aus, dass die Platine des LoRa-Knotens einen gewinkelten Standardsteckverbinder mit 0,1-Zoll-Rastermaß in der Position K5

besitzt. Die sieben Stifte in den Positionen 11 bis 17 (siehe Platine 180516-1) sind ausreichend, aber K3 kann auch eine 17-polige, auf Länge geschnittene Stiftleiste über die gesamte Länge von K5 sein.

Der in der Stückliste mit K3 gekennzeichnete Steckverbinder von Würth Elektronik ist sicher schwieriger zu beschaffen als eine passend abgeschnittene Stiftleiste. Dennoch haben wir uns für das Würth-Modell entschieden, weil es eine geringe Höhe hat, so dass die beiden rechtwinklig zueinander angeordneten Platinen in das empfohlene Gehäuse passen. Die zweite Kontaktreihe ist überflüssig, aber wir konnten einfach keine einreihige Steckdosenleiste mit

einer geeigneten Höhe finden.

Der Footprint für den Steckverbinder K3 wurde jedoch so angepasst, dass auch normale (through-hole) 0,1-Zoll-Buchsenleisten passen. Das ist recht praktisch, wenn Sie ein anderes, höheres Gehäuse wählen. Sie können K3 auch ganz weglassen und den rechtwinkligen Steckverbinder des LoRa-Knotens direkt anlöten, aber dann lässt sich das Gebilde nicht mehr so einfach trennen. Der Drucktaster S1 ist am Gehäuse montiert und mit zwei Drähten mit der Platine verbunden.

### Installation der Software

Das komplette Software-Paket für den Elektor-LoRa-Knoten ist unter [6] erhältlich.

### Weblinks

[1] KiCAD Like a Pro: [www.elektor.de/kicad-wie-ein-profi](http://www.elektor.de/kicad-wie-ein-profi)

[2] GPS-Modul aus dem Elektor-Store: [www.elektor.de/open-smart-gps-serial-gps-module-for-arduino-arduino-2-5-flight-control](http://www.elektor.de/open-smart-gps-serial-gps-module-for-arduino-arduino-2-5-flight-control)

[3] Projekt stm32duino: [https://github.com/stm32duino/Arduino\\_Core\\_STM32](https://github.com/stm32duino/Arduino_Core_STM32)

[4] Verkürzungsfaktor: <https://de.wikipedia.org/wiki/Verk%C3%BCrzungsfaktor>

[5] STM32CubeProgrammer: [www.st.com/en/development-tools/stm32cubeprog.html](http://www.st.com/en/development-tools/stm32cubeprog.html)

[6] Projektsoftware: [www.elektormagazine.de/180666-03](http://www.elektormagazine.de/180666-03)

[7] Hinzufügen der STM32-Boards: <https://github.com/stm32duino/wiki/wiki/Getting-Started>

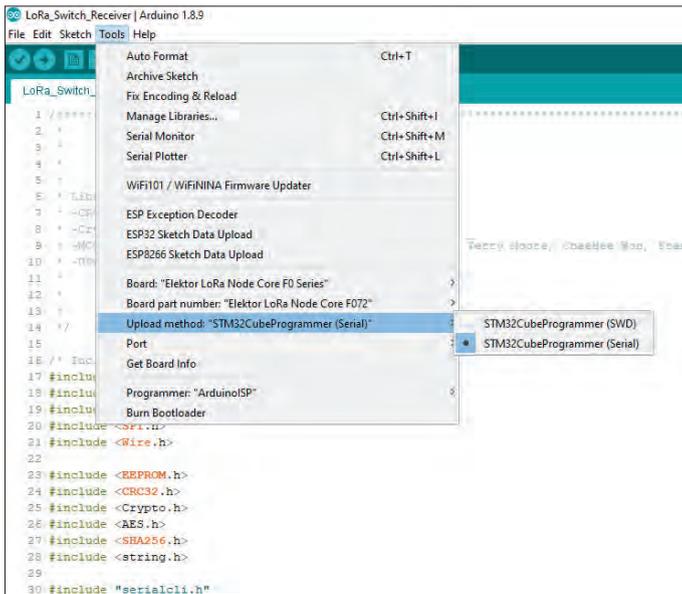


Bild 12. Verwenden Sie diese Board-Einstellungen.

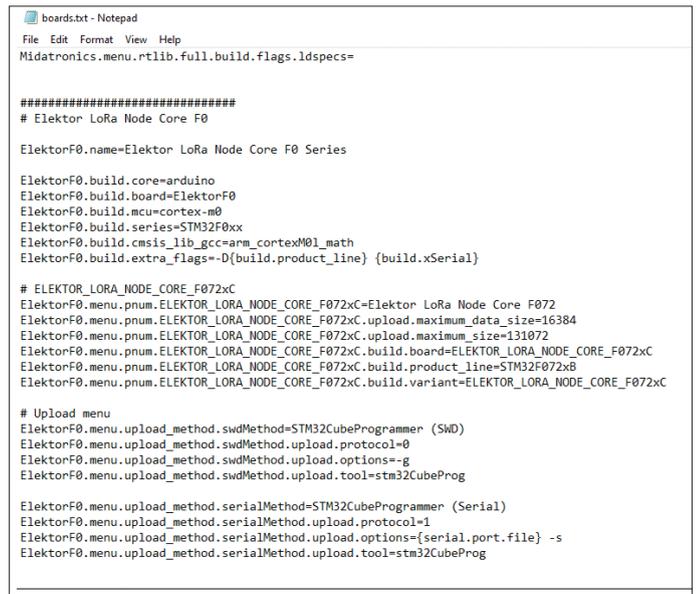


Bild 13. Dies war nur einer der Schritte, die das Elektor-Labor gehen musste, um den LoRa-Knoten (eine Inkarnation der LoRaWAN-Knoten-Experimentierplattform) als STM32-Arduino-Gerät genehmigen und registrieren zu lassen. Elektor-Lesern bleibt dieser administrative Prozess erspart; die Projekt-Software ist bereit zum Start.

lich. Die Archivdatei enthält auch eine detaillierte Installationsanleitung, die hier aus Platzgründen nicht abgedruckt werden kann.

Wenn es um die Firmware-Entwicklung für den LoRa-Knoten geht, kann man entweder den harten C-Weg mit der STM32CubeIDE oder den einfachen Weg mit der Arduino-IDE wählen. Für letzteren müssen Sie neben der Arduino-IDE auch den STM32 Arduino-Core herunterladen. In der Arduino-IDE fügen Sie dazu einfach folgenden URL zum *Additional Boards Manager* (unter Voreinstellungen) hinzu:

[https://github.com/stm32duino/BoardManagerFiles/raw/master/STM32/package\\_stm\\_index.json](https://github.com/stm32duino/BoardManagerFiles/raw/master/STM32/package_stm_index.json)

und installieren Sie die STM32-Boards gemäß den Anweisungen unter [7].

Nun haben Sie den STM32-Kern, aber für den Elektor-LoRa-Knoten müssen Sie noch ein paar Dinge in Ihrem lokalen Dateisystem patchen. Unter Windows gehen Sie zu:

`%localappdata%\Arduino15\Pakete\STM32\Hardware\stm32\1.7.0`

und fügen Sie am Ende von *Boards.txt* den Inhalt der Datei *Add\_To\_Board.txt* hinzu, die wir für Sie geschrieben haben.

Kopieren Sie auch den Ordner:

`ELEKTOR_LORA_NODE_CORE_F072xC`

in den Ordner *Versions*. Nun sollte alles bereit sein, um die Software für unser

Projekt zu kompilieren. Am Ende müssen Sie die in **Bild 12** gezeigten Board-Einstellungen definieren.

An diesem Punkt können Sie die Software für das Board kompilieren. Um einen Upload durchzuführen, müssen Sie den STM32CubeProgrammer [5] installieren. Wenn Sie den Code hochladen möchten, bringen Sie das Board in den Bootloader-Modus, indem Sie die Boot-Taste gedrückt halten und kurz auf Reset drücken. Dann können Sie die neue Firmware auf das Board hochladen.

### Zu Ihrem Vorteil

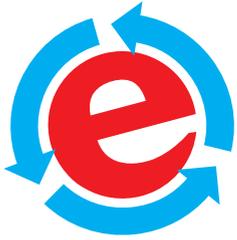
Zu Beginn des Artikels haben wir behauptet, dass diese Geschichte teilweise

chronologisch sein würde. Mitte Dezember 2019, als dieser Artikel schon längst in Bearbeitung war, wurden die Definitionen der Elektor-LoRaWAN-Knoten-Experimentierplattform-Platine in das offizielle STM32Arduino-Git-Depot eingefügt, nachdem die ziemlich strengen Bedingungen des STM32duino-Gremiums erfüllt wurden (**Bild 13** – thanks guys). Als glückliche Konsequenz müssen Sie nichts mehr wie oben beschrieben patchen, sondern können einfach das Elektor-LoRa-Knoten-Board aus der Liste der genehmigten Elemente auswählen. ◀

180666-03

### IM ELEKTOR-STORE

- **Elektor-LoRa-Knoten, 180516 V3.1**  
[www.elektor.de/180516-1](http://www.elektor.de/180516-1)
- **LoRa-Button, 180666-2 V1.2**  
[www.elektor.de/180666-2](http://www.elektor.de/180666-2)
- **LoRa AC-Schalter, 180666-1 V1.1**  
[www.elektor.de/180666-1](http://www.elektor.de/180666-1)
- **RFM95W-868S2 LoRa-Sender/Empfänger**  
[www.elektor.de/18715](http://www.elektor.de/18715)
- **KiCAD wie ein Profi-Buch**  
[www.elektor.de/kicad-wie-ein-profi](http://www.elektor.de/kicad-wie-ein-profi)
- **GPS-Modul (optional)**  
[www.elektor.de/open-smart-gps-serial-gps-module-for-arduino-apm2-5-flight-control](http://www.elektor.de/open-smart-gps-serial-gps-module-for-arduino-apm2-5-flight-control)



# Interaktiv

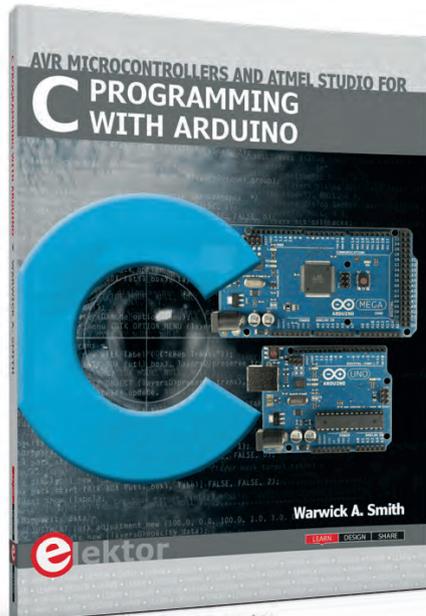
## Korrekturen & Updates || Fragen & Antworten

Von **Clemens Valens** (Elektor-Labor)

Aktualisierungen und Ergänzungen von Projekten, die in Elektor veröffentlicht wurden, ergänzt mit Tipps und Tricks, technischen Ratschlägen und Antworten auf Leserfragen.

### Probleme mit Arduino Uno

**F:** Ich habe vor kurzem das Elektor-Buch C-Programmierung mit Arduino erworben und mit Begeisterung damit gearbeitet. Dazu habe ich auch einen Atmel-ICE-Programmierer von Microchip gekauft, der sehr gut mit der im Buch verwendeten IDE Atmel



Studio 7.0 (AS7) zusammenarbeitet. Nun habe ich aber festgestellt, dass der Arduino Uno nicht mehr mit der Arduino-IDE programmiert werden kann. Ein weiterer, anderer Uno lässt sich dagegen immer noch gut mit der Arduino-IDE programmieren. Ich habe versucht, das Board wieder anzuschließen, den Computer oder die Software neu zu starten und

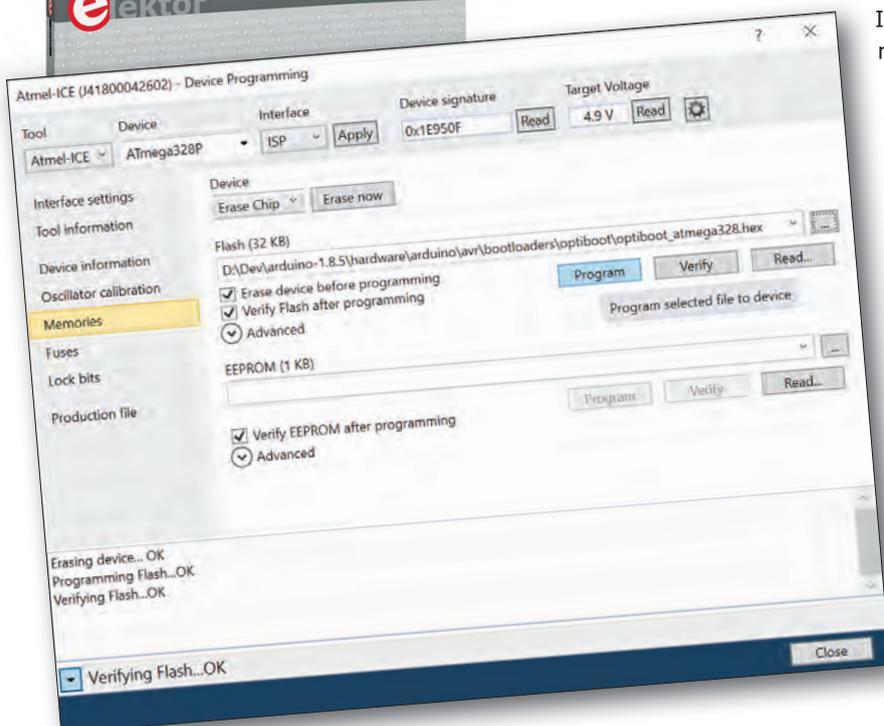
ähnliche Maßnahmen ergriffen, alles ohne Erfolg. Natürlich habe ich auch mehrere Foren im Internet durchstöbert, aber auch dort konnte ich keine Lösung finden. Haben Sie einen Vorschlag, was ich tun kann, oder noch besser, eine Lösung?

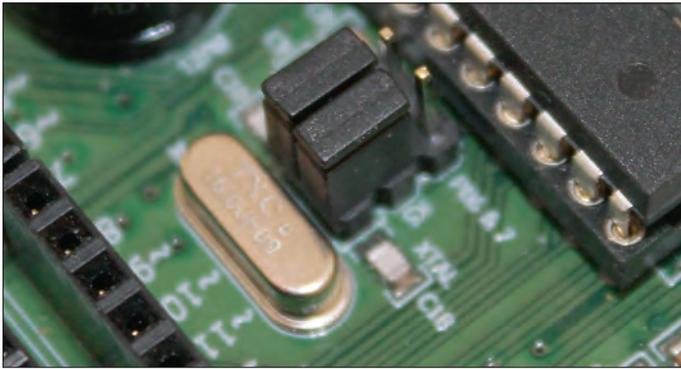
**A:** Die Sache, die einen Mikrocontroller in einen Arduino-kompatiblen Mikrocontroller verwandelt, ist der Arduino-Bootloader. Das ist ein kleines Stück Software im Speicher der MCU, das in der Lage ist, ein Programm, das es irgendwie (meist über eine serielle Verbindung) empfangen hat, in den Speicher der MCU zu schreiben, ohne sich selbst zu überschreiben. Die Arduino-IDE kennt den Bootloader und verwendet den seriellen Port (über USB), um das Arduino-Board zu programmieren. Atmel Studio kennt weder Arduino noch den Arduino-Bootloader, daher wird ein spezieller Programmieradapter benötigt, um ein Programm in den Speicher der MCU zu brennen. Der Programmieradapter Atmel-ICE (und andere) verbindet sich mit der In-System-Programming-Schnittstelle (ISP) der MCU, um direkten Zugriff auf den Speicher der MCU zu erhalten. Trifft man keine speziellen Maßnahmen dagegen, überschreibt er den gesamten Speicherinhalt einschließlich des Bootloaders. Dies erklärt, warum der Arduino Uno nicht mehr mit der Arduino-IDE zusammenarbeitet: Der Bootloader ist weg!

In AS7 kann mit Atmel ICE der Arduino-Bootloader neu programmiert werden. Die benötigte Datei ist `[Arduino]\hardware\arduino\avr\bootloaders\optiboot\optiboot_atmega328.hex` wobei `[Arduino]` der Ordner ist, der die Datei `arduino.exe` enthält. Einmal in AS7, öffnen Sie *Device Programming* (Umschalt-Strg-P). Klicken Sie dann auf der Registerkarte *Memories* in der Sektion *Flash* auf den Schalter *Browse for file* und navigieren Sie zur HEX-Datei des Bootloaders. Klicken Sie auf *Program*, um den Bootloader in die MCU zu flashen.

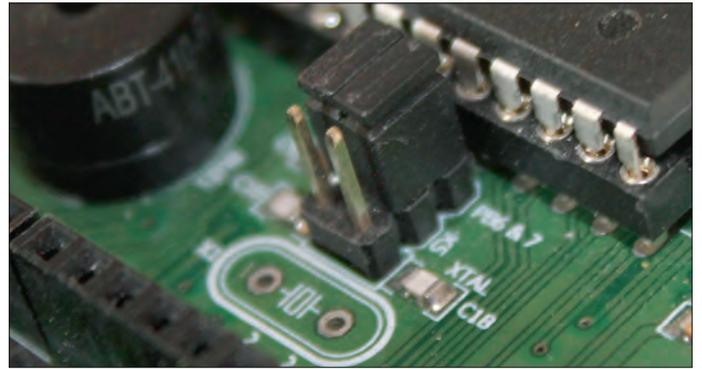
Es gibt viele Möglichkeiten, AS7 so zu konfigurieren, dass es den Arduino-Bootloader standardmäßig verschont, wenn Sie auf *Program* klicken. Sie reichen von der Installation von AS7-Erweiterungen bis zur Anpassung der Linker-Einstellungen. Bei einer Internet-Recherche werden Sie sicher die passende Methode finden!

**F:** Ist es möglich, den ATmega328 eines Arduino-Uno-Boards auf dem AVR-Playground zu verwenden und umgekehrt?





Der AVR-Playground läuft mit einem 16-MHz-Quarz.



Der AVR-Playground läuft mit seinem internen 8-MHz-RC-Oszillator.

**A:** Obwohl der Arduino Uno und der AVR Playground auf dem gleichen Mikrocontroller ATmega328 basieren, kann man sie nicht einfach austauschen. Das liegt an der Art und Weise, wie der Taktoszillator auf den Boards verwendet wird. Der Arduino Uno läuft mit einem 16-MHz-Takt, der von einem 16-MHz-Quarz erzeugt wird. Im Gegensatz dazu verwendet der AVR-Playground standardmäßig den internen 8-MHz-RC-Oszillator der MCU. Sie können die MCU eines Arduino Uno durch eine MCU ersetzen, die auf einem AVR-Playground verwendet wird. Der Uno arbeitet dann genau wie ein AVR-Playground, der mit 8 MHz läuft. Der umgekehrte Weg, die MCU auf einem AVR-Playground durch eine vom Arduino Uno zu ersetzen, ist jedoch nur möglich, wenn der AVR-Playground mit einem 16-MHz-Quarz (X1) ausgestattet ist und die Jumper an K5 auf die Pins 1-3 und 2-4 gesteckt sind. Wenn diese Bedingungen erfüllt sind, wird der AVR-Playground genau wie ein Arduino Uno funktionieren. Doch es gibt noch mehr dazu zu sagen. Der Bootloader, der in der MCU eines Arduino-kompatiblen Boards steckt, ist auf die Taktrate der MCU abgestimmt, um die Geschwindigkeit der seriellen Schnittstelle, die zum Hochladen von Sketches verwendet wird, zu konfigurieren. Wenn sich die Taktfrequenz ändert,

ändert sich auch die Geschwindigkeit der seriellen Schnittstelle. Leider erlaubt die Arduino-IDE dem Benutzer nicht, die Geschwindigkeit der seriellen Schnittstelle für das Hochladen von Sketches festzulegen. Wenn Sie ein Arduino-Uno-Board in der Arduino-IDE auswählen, geht die IDE davon aus, dass es mit 16 MHz getaktet wird und eine Upload-Geschwindigkeit von 115.200 Baud benötigt. Wenn ein Uno nicht mit 16 MHz läuft, schlägt der Upload eines Sketches daher fehl. Ähnlich wird von der IDE ein AVR-Playground erwartet, der mit 8 MHz und einer Upload-Geschwindigkeit von 57.600 Baud läuft.

Die Einfachheit der Arduino-IDE wird von den meisten seiner Benutzer hoch geschätzt, aber manchmal kann sie im Weg stehen. Wer Abenteuer liebt, kann versuchen, in der IDE die Datei `boards.txt` zu modifizieren, die alle Informationen über die Boards enthält. Allerdings kann es sein, dass es mehr als eine dieser Dateien in Ihrer Arduino-IDE-Installation gibt. Die Keys, die geändert werden müssen, sind `[Board].upload.speed` und `[Board].build.f_cpu` (wobei `[Board]` der Name des Boards ist). Die IDE muss neu gestartet werden, um alle Änderungen in dieser Datei zu berücksichtigen.

[www.elektormagazine.de/labs/avr-playground-129009-2](http://www.elektormagazine.de/labs/avr-playground-129009-2)

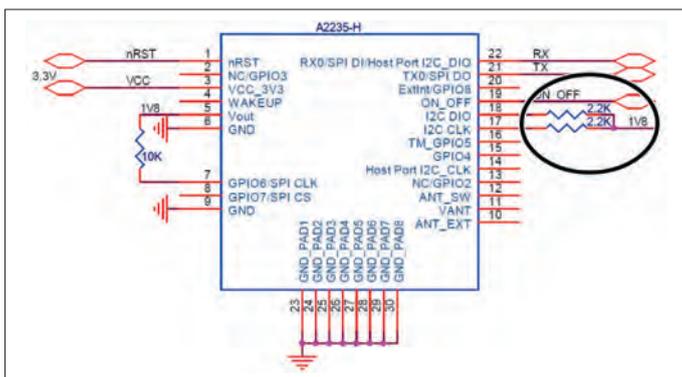
## Neues GPS für die Neue Nixie-Uhr

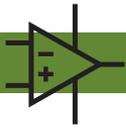
Es scheint, dass das in diesem Projekt verwendete GPS-Modul vom Typ Maestro A2035-H nicht mehr produziert wird. Lantronix, das im Sommer 2019 Maestro übernommen hat, bietet jedoch mit dem A2235-H ein neueres, wenn auch kleineres GPS-Modul an, das auf unsere Platine passt. Es benötigt zwei zusätzliche Pull-Up-Widerstände von 2,2 kΩ von den I2C-Pins zum 1,8-V-Ausgang des Moduls. Möglicherweise benötigen Sie

auch eine (Polyimid-)Isolierung zwischen der Platine und dem Modul, um einen Kurzschluss zwischen der Abschirmung des Moduls und einer darunter liegenden Durchkontaktierung zu vermeiden. ◀

[www.elektormagazine.de/labs/150189-6-digit-nixie-clock](http://www.elektormagazine.de/labs/150189-6-digit-nixie-clock)

190379-B-03





# Entwicklung analoger Elektronik

## Fall Nr. 1 – Teil 2: Vorverstärker für MEMS-Mikrofone

Von **Ton Giesberts** (Elektor-Labor)

In dieser Artikelserie beleuchten Experten spezielle Aspekte der analogen Elektronik-Entwicklung, die einem zunehmend „rein digitalen“ Publikum zugute kommen. Sie zeigen, dass „analog“ keine schwarze Magie ist. In der vorhergehenden Folge wurde das MEMS-Mikrofon diskutiert, das zum ersten Mal in einem Elektor-Projekt namens Fledermaus-Detektor<sup>PLUS</sup> verwendet wurde. In dieser Folge geht es um Design-Aspekte des Mikrofonverstärkers.

Diese Folge aus der Serie „Entwicklung analoger Elektronik“ enthält eine detaillierte Betrachtung des im Projekt Fledermaus-Detektor<sup>PLUS</sup> [1] verwendeten Mikrofonvorverstärkers und diskutiert mögliche Optimierungen für die Verwendung mit dem in Teil 1 dieser Fallstudie untersuchten MEMS-Mikrofon [2]. Für eine wirkliche Beurteilung der Elektronik ist es unerlässlich, den Schaltplan des Mikrofon-Vorverstärkers zu inspizieren. Er ist Teil der Schaltung von **Bild 1**. Es wurde absichtlich der gesamte Schaltplan des Fledermaus-Detektors nachgedruckt, um folgende Aspekte explizit zu zeigen:

- Ansteuerung: durch das angeschlossene Mikrofon an K1
- Verstärkung: durch die Opamps von IC2 und IC3
- Versorgung: mit  $\pm 4,5$  V, symmetriert durch IC1

Jetzt haben Sie den Überblick. Zunächst zur ursprünglichen Schaltung: Auch viele Jahrzehnte seit seinem Erscheinen ist der Low-Power-Opamp TL062 mit seinen 200  $\mu$ A Strombedarf pro Verstärker nicht schlecht im Vergleich zu vielen aktuellen Low-Power-Versionen, die fast durchweg in immer kleineren SMD-Versionen zu finden sind. Offensichtlich gibt es jetzt (viel) bessere. Das Verstärkungs-Bandbreiten-Produkt (GBW) des TL061 beträgt nur 1 MHz – ein Hindernis bei der Entwicklung guter Elektronik, weil eine erhebliche Verstärkung von etwa 2.000 benötigt wird. Die Verwendung nur eines Operationsverstärkers würde zu einer unbrauchbaren Bandbreite von etwa 500 Hz führen. Für mehr Bandbreite wurden drei Opamps mit jeweils geringerer Verstärkung hintereinander geschaltet und so die Gesamtverstärkung zugunsten von mehr Bandbreite aufgeteilt. Die Frage ist, wie viele Opamps einen guten Kompromiss ergeben. Benötigt ein einzelner aber viel schnellerer Opamp wirklich mehr Strom als diese Dreierkette? Heute gibt es „verdammte schnelle“ Exemplare wie den Dual-Opamp OPA2889 mit einer GBW von 75 MHz. Der Strombedarf liegt bei nur 0,92 mA für beide Opamps und damit kaum über der Lösung mit zwei Dual-Opamps des Typs TL062.

Nochmals zur Anzahl benötigter Opamps: Die ist schwierig zu berechnen, da der Bandbreitenverlauf nicht einem simplen Tiefpass erster Ordnung entspricht. Außerdem: Die Bandbreite

von mehreren in Reihe geschalteten Opamps ist kleiner als bei nur einem Verstärker. Als Näherung kann man so eine Opamp-Kaskade aber wie ein RC-Netzwerk erster Ordnung betrachten. Dies gilt sicherlich für die untere Grenzfrequenz bei der Kaskadierung gleich dimensionierter Verstärkerstufen. Jede zusätzliche Stufe erhöht dann die untere Grenzfrequenz. Die Übertragungsfunktion eines RC-Hochpasses erster Ordnung entspricht:

$$j\omega RC / (1 + j\omega RC) \quad (\text{Gl. 1})$$

Aufgrund der Phasenverschiebung ist die tatsächliche Spannungsverteilung:

$$\omega RC / \sqrt{1 + (\omega RC)^2} \quad (\text{Gl. 2})$$

Bei der Grenzfrequenz entspricht dies  $1/2\sqrt{2}$ , dem bekannten „-3-dB-Punkt“. Ganz genau ist das:

$$20 \log 1/2\sqrt{2} = -3,0103$$

Für die tatsächliche Bandbreite kaskadierter Verstärker nimmt man statt  $1/2\sqrt{2}$  den Faktor  $x$ . Daraus ergibt sich:

$$\omega RC = x / \sqrt{1 - x^2} \quad (\text{Gl. 3})$$

Bei einem -3-dB-Punkt setzt man den Wert  $1/2\sqrt{2}$  für  $x$  ein. Das ergibt genau 1, und damit hat man die bekannte Standardformel für die -3-dB-Grenzfrequenz:

$$f = 1 / (2\pi RC) \quad (\text{Gl. 4})$$

Nun ist die Grenzfrequenz für einen Verstärker bzw. ein RC-Glied bekannt. Bei einer Kaskadierung von drei Verstärkern wie in der Schaltung sucht man die Frequenz, bei der die Dämpfung 1 dB beträgt. Der Einfachheit halber nehmen wir nicht  $-3,0103 / 3$ , sondern glatt  $-1$ . Aus  $20 \log x = -1$  folgt:

$$x = 10^{(-1/20)} \quad (\text{Gl. 5})$$

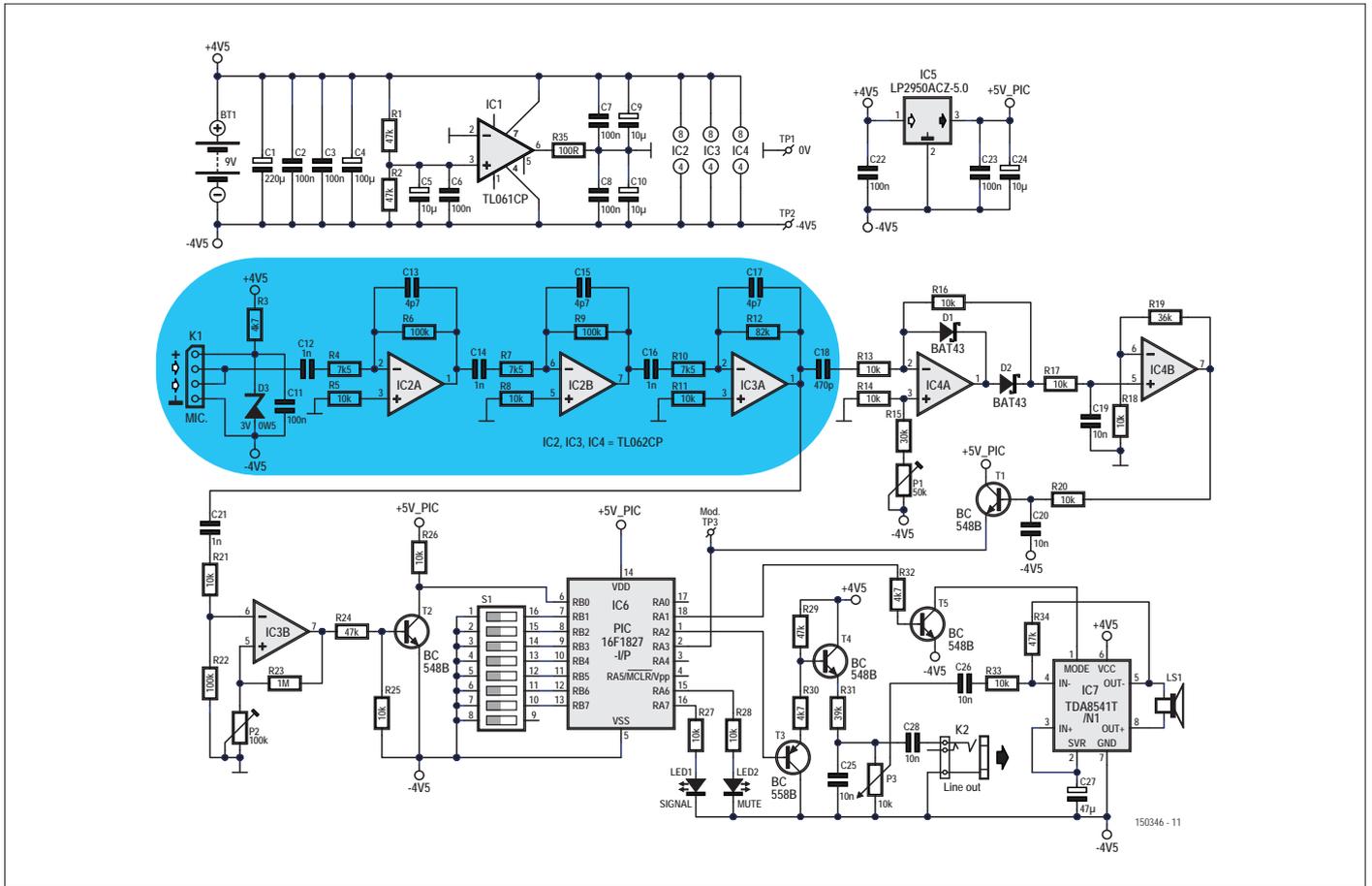


Bild 1. Fledermaus-Detektor<sup>PLUS</sup>. Vollständige Schaltung mit hervorgehobenem dreistufigen Mikrofonvorverstärker.

Die Dämpfung beim -1-dB-Punkt beträgt dann 0,891.251. Gibt man das in Gleichung 3 ein, so ergibt  $\omega RC$  den Wert 1,965.23. Daraus resultiert, dass die neue Grenzfrequenz für ein Trio aus gleichen Stufen fast doppelt so hoch ist. Bei einem Tiefpass wird die Grenzfrequenz entsprechend fast halbiert. Für  $n$  Stufen gilt allgemein:

$$x = 10^{((-3/n)/20)} \quad (\text{Gl. 6})$$

Beim TL062 ergibt sich für IC2A und IC2B eine Verstärkung von ca. 13,3 (100 kΩ / 7,5 kΩ, wenn man C12 und C14 ignoriert). Folglich ergibt sich die obere Grenzfrequenz bei einer GBW von 1 MHz zu 75 kHz. Die zusätzlichen 4,7-pF-Kondensatoren parallel zu R6 und R9 senken die Grenzfrequenz weiter auf etwa 60 kHz pro Opamp und „versteilern“ den Frequenzgang in Richtung höherer Ordnung.

**Zunächst simulieren...**

...und dann erst eine Platine entwerfen oder einen Prototyp auf einem Steckbrett bzw. einer Lochrasterplatine bauen. Dies hilft herauszufinden, ob weitere Stufen benötigt werden oder ob etwas anderes angepasst werden muss. Oft ist die Simulation einer Schaltung schneller als die sonst nötige Rechnerei. Die Simula-

tion des Vorverstärkers ergab einen Frequenzbereich von 23 kHz bis 58 kHz - zu niedrig! Die obere Grenzfrequenz des Prototyps erreichte sogar nur knapp 44 kHz. Nun gut, es gibt Toleranzen. Die GBW von 1 MHz beim TL062 ist ein **typischer** Wert - Maximum und Minimum sind nicht angegeben, und die Opamps werden hier mit niedrigeren Spannungen versorgt als im Datenblatt für die angegebene GBW spezifiziert. Außerdem entspricht der Frequenzgang der drei Tiefpässe nicht der ersten Ordnung.

**Wie viele Stufen dann?**

Falls man den guten, alten TL062 einsetzen will: Es wird empfohlen, die gewünschte Bandbreite nicht durch den Operationsverstärker selbst begrenzen zu lassen, da dessen GBW einer beträchtlichen Streuung unterliegt. Die eigentlich negativen Auswirkungen bezüglich Verzerrung und Intermodulation wären hier kein großes Problem. Doch wäre es besser, die GBW so groß zu halten, dass sich eine Grenzfrequenz vom Doppelten des eigentlich gewünschten Tiefpasses ergibt. Ein dreifacher Wert wäre noch besser. Eine höhere nutzbare Bandbreite erfordert eine niedrigere Verstärkung. Bei der deutlich besseren Bandbreite von 360 kHz ergäbe sich folgende Verstärkung:

$$A = 1 \text{ MHz} / 360 \text{ kHz} = 2,78.$$

So wenig Verstärkung pro Stufe ergibt viele Opamp-Stufen! Bei der errechneten Verstärkung  $A$  pro Stufe und einer Gesamtverstärkung von 2.000 ergibt sich:

$$A^n = 2.000 \quad (\text{Gl. 7})$$

Umgestellt nach der Anzahl nötiger Stufen  $n$  ergibt sich:

$$n = (\log 2.000) / (\log A) \quad (\text{Gl. 8})$$

Bei  $A = 2,78$  erhält man  $n = 7,43$ . Nötig wären dann acht Stufen bzw. zwei Vierfach-Opamps (z.B. zwei TL064, Gesamtstromaufnahme = 1,6 mA). Die Verwendung eines schnelleren (Dual-) Opamps scheint die deutlich bessere Wahl zu sein, denn diese Lösung kommt mit weniger Bauteilen aus und braucht nicht so viel Platinenplatz. Der Vorteil mehrerer Opamps ist ein größerer physikalischer Abstand zwischen dem relativ großen Ausgangssignal und dem Verstärkereingang, was zu einer geringeren Schwingneigung aufgrund parasitärer Kopplung führt. Bei schnellen SMD-Opamps empfiehlt sich die SMD-Bauform auch bei den anderen Bauteilen. Parasitäre Kopplungen mit dem Mikrofon sind kaum zu befürchten, da es sich um ein abgeschirmtes Modell handelt. Nur in wirklich üblen Fällen kann so ein Effekt bei Steckern und Kabeln zu und von Platinen auftreten.

### Gesucht: Bandbreite!

Zur Sicherheit kann man für die theoretische Bandbreite eine Überschlagsrechnung anstellen. Je mehr Stufen, desto geringer die benötigte Verstärkung. Bei  $n$  Abschnitten ist die Verstärkung  $A$  pro Abschnitt:

$$A = \sqrt[n]{2.000} \quad (\text{Gl. 9})$$

Bei acht Stufen ist die erforderliche Stufenverstärkung  $A = 2,586$ . Gleichung 6 ergibt 0,95775 für  $x$ . Laut Gleichung 3 wird die Grenzfrequenz um den Faktor 3,33 verschoben. Daraus ergibt sich die neue Bandbreite (ohne die  $C_s$  in der Gegenkopplung) zu:

$$1 \text{ MHz} / (2,586 \times 3,33) = 116 \text{ kHz.}$$

Das ist immer noch knapp, obwohl der Verstärker bereits von drei auf acht Stufen erweitert wurde. Aus den vorgenannten Gründen ist damit aber noch keine akzeptable Bandbreite erreicht. In einer Simulation mit 7,5 k $\Omega$  am Eingang und 19,3 k $\Omega$  für die Rückkopplung (für  $A = 2,57$ ) ergeben acht Stufen in Reihe eine obere Grenzfrequenz von etwa 107 kHz (siehe **Bild 2**). Das ist noch nicht optimal. Außerdem sollten die 1-nF-Koppelkondensatoren angepasst werden. Will man eine untere Grenzfrequenz von etwa 10 kHz erreichen, bräuhete man  $C_s$  mit dem krummen Wert von 7 nF. Der nächstgrößere Wert der E6-Reihe liegt bei 10 nF. Hiermit ergibt sich eine Grenzfrequenz von 6,6 kHz. Ein anderes Konzept bestände in noch schnelleren Opamps und aktiven Filtern, z.B. mit einer Butterworth- oder Tschebyscheff-Charakteristik.

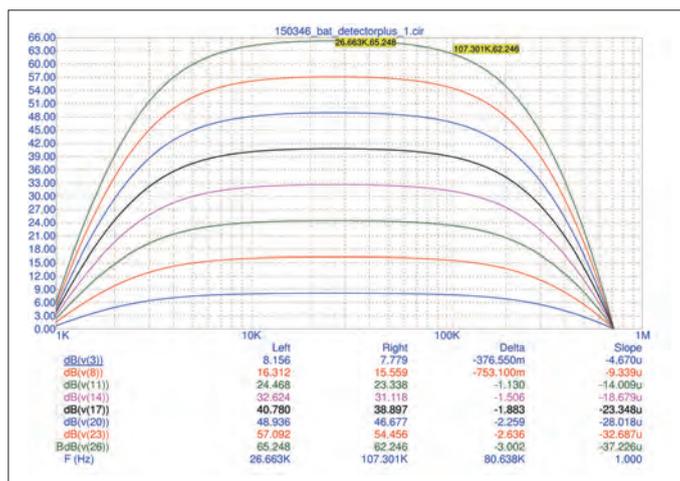


Bild 2. Die Grafik zeigt die Wirkung von acht identischen Verstärkern in Serie (Quelle = 0  $\Omega$ ).

### Signal/Rausch-Verhältnis

Auch für das Rauschen sind acht Stufen nicht so ideal, da die erste Stufe wesentlich über den Signal-Rausch-Abstand entscheidet. Am Eingang wäre ein nichtinvertierender Verstärker besser. Es läge dann (auch dank der niedrigen Impedanz des Mikrofons) kein mit 7,5 k $\Omega$  recht hochohmiger Widerstand mit eigenem Rauschbeitrag in Serie mit dem Eingangssignal und der Widerstand der Gegenkopplung der ersten Stufe könnte niederohmiger ausfallen. Beides käme dem Signal/Rausch-Verhältnis und dem Dynamikbereich zugute.

Das verwendete MEMS-Mikrofon hat eine maximale Ausgangsimpedanz von 400  $\Omega$ . Ein Elektret-Mikrofon hat noch höhere Werte. Der Frequenzgang der Schaltung ist nicht für dieses Mikrofon optimiert. Leider geben die meisten Hersteller den Frequenzgang über 20 kHz nicht an. Oft wird er gar nur bis 10 kHz angegeben – für einen Fledermaus-Detektor irrelevant! Es wäre schön, wenn man den Frequenzgang selbst messen und überprüfen könnte. Absolute Genauigkeit ist nicht erforderlich – ein Vergleich mit dem bekannten Frequenzgang einer Breitband-Ultraschallquelle würde genügen. Die Beschränkungen der Selbstbau-Kompatibilität wären dann ein zusätzliches Problem.

### Ausblick

Der abschließende Teil 3 dieser Fallstudie wird in der Mai/Juni-Ausgabe 2020 von Elektor erscheinen. Darin werden weitere analoge Herausforderungen und Lösungen erörtert, einschließlich des Dynamikbereichs von Vorverstärkern, des Eingangsrauschens, der Bias-Ströme von Opamps und des Anschlusses von zwei- und dreipoligen Mikrofonen. Auf Wiederlesen. ◀

191230-06

### Weblinks

- [1] Fledermaus-Detektor<sup>PLUS</sup>: [www.elektormagazine.de/magazine/elektor-201607/29063](http://www.elektormagazine.de/magazine/elektor-201607/29063)
- [2] Entwicklung analoger Elektronik (1): [www.elektormagazine.de/191143-03](http://www.elektormagazine.de/191143-03)

# My IoT-Button: Der Knopf fürs Netz

## Teil 1: IoT-Architektur

Von Dr. Veikko Krypczyk

Immer neue Anwendungen gibt es im Umfeld des Internet of Things (IoT). Eine davon ist der IoT-Button, ein Taster, der nach Belieben programmiert und verwendet werden kann. Er benötigt eine Netzwerkverbindung und holt seine Logik aus der Cloud. Im ersten Teil des Artikels geht es um die IoT-Architektur auf der Basis von Serverless Cloud Computing und das Zusammenspiel von Hard- und Software.

Immer wieder fallen uns neue Anwendungen des Internet of Things (IoT) ein. Viele Ideen sind experimentell und/oder werden wohl nicht so schnell den Weg in die reale IoT-Welt finden. Die Vision des sich selbst füllenden Kühlschranks scheitert wahrscheinlich bereits an der Tatsache, dass man Lebensmittel eben doch ganz gern nach dem aktuellen Geschmack im Markt seines Vertrauens aussucht. Eine interessante Idee, die an der Grenze zwischen ernsthafter Anwendung und spielerischer Beschäftigung angesiedelt ist, sind so genannte IoT-Buttons. Solche IoT-Buttons sollen beispielsweise

- etwas starten oder stoppen
- etwas bestellen
- jemanden anrufen
- etwas zählen.

Die Anwendungen sind vielfältig und können sowohl im privaten als auch im unternehmerischen Bereich liegen. Die Idee ist, mit einer Hardware in Form eines Tasters (Button) ein Signal zu versenden. Wohin das Signal gesendet wird, welche Aktionen damit verbunden und welche Systeme angebunden sind und ob es sich überhaupt nur um einen einzigen Taster handelt, bleibt zunächst völlig offen.

Zunächst nehmen wir das Thema zum Anlass, moderne IoT-Architektur zu beleuchten und zu verstehen. Wir beschäftigen uns im ersten Teil des Artikels mit dem Zusammenspiel der Komponenten und loten die Funktionen und Möglichkeiten von Cloud

Services aus. Im IoT lassen sich sehr einfache Experimente durchführen, was auch für die Nutzung von Cloud-Diensten gilt. Mit einer (minimalen) Hardware, etwas Software und einigen Konfigurationen werden wir im zweiten Teil des Artikels erste Experimente unternehmen und so die Voraussetzungen schaffen, um eigene Ideen rund um diese Spielart des IoT zu erproben.

### IoT-Architektur

Die Architektur einer modernen IoT-Lösung kann man sich – unabhängig vom konkreten Einsatzzweck und des letztendlich gewählten Dienstes – wie in **Bild 1** vorstellen. Die einzelnen Geräte (Devices) auf der linken Seite können



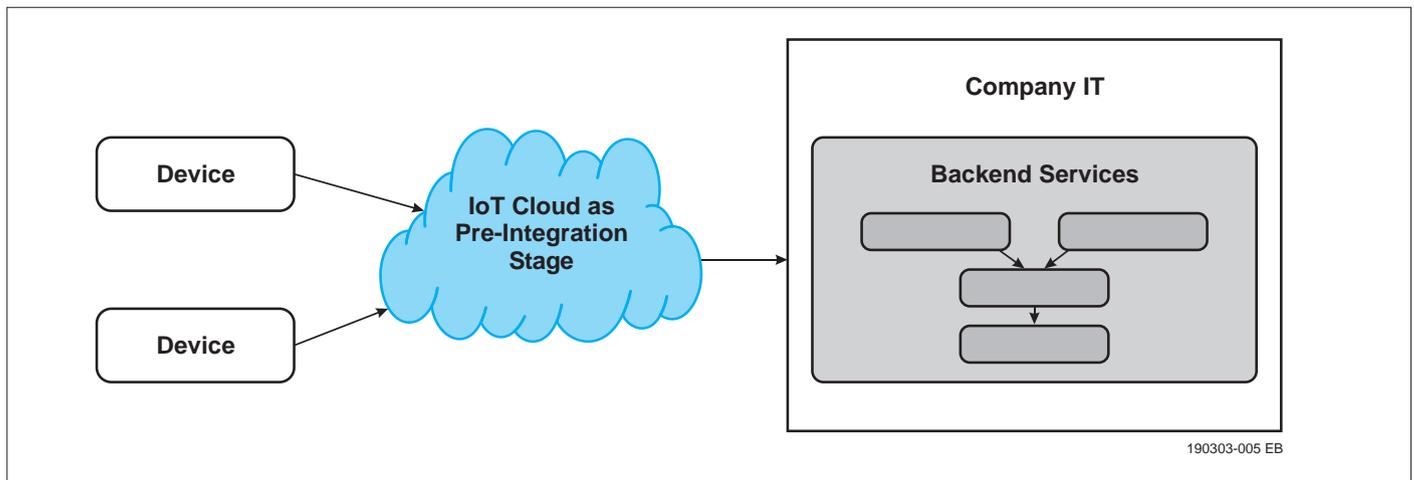


Bild 1. Prinzipielle Architektur einer IoT-Anwendung mit Cloud-Backend.

Sensoren oder Aktoren oder Kombinationen aus beidem sein. Sensoren messen zum Beispiel Umweltzustände wie die Temperatur, überwachen den Status von Anlagen oder senden ein Signal, wenn sich der Zustand eines überwachten Systems ändert. Aktoren führen dagegen Reaktionen aus, sie öffnen oder schließen beispielsweise das Ventil eines Heizkörpers. In einigen IoT-Anwendungen kommen sowohl Sensoren als auch Aktoren zum Einsatz, die gewissermaßen als Regelkreis agieren – gesteuert über ein entferntes Anwendungssystem. In anderen IoT-Szenarien geht es lediglich um die Erfassung oder Überwachung eines Zustandes. Bei unserem simplen IoT-Button wird nur das Betätigen eines Tasters überwacht.

Die Information, dass sich der Zustand des Buttons geändert hat, wird über ein Netzwerk (in der Regel das Internet) übertragen. Empfänger der Signale ist ein Dienst auf einem Server. Die speziellen, auf die Anforderungen von IoT-Anwendungen zugeschnittenen Dienste werden zur so genannten IoT-Cloud zusammengefasst. Diese IoT-Cloud nimmt eine zentrale Rolle ein, sie übernimmt die Rolle eines Servers und ist damit der Kommunikationspartner für die IoT-Devices. Da man hier nur ausgewählte Funktionen und keinen kompletten Server nutzt, spricht man vom Serverless Computing. Natürlich laufen die Dienste auf einem Server eines Diensteanbieters, man braucht sich dabei aber nicht um die Einrichtung, die Administration und ähnliches Beiwerk kümmern. Bei einer höheren Last, zum Beispiel einer größeren Anzahl an Calls an den Dienst,

werden zusätzliche Ressourcen automatisch bereitgestellt (horizontale Skalierung). Auch darum muss man sich als Nutzer normalerweise nicht kümmern. Zentrale Funktionen der IoT-Cloud sind:

- **Gerätebasierte Authentifizierung:** Ermöglicht eine vielfältige Anmeldung und Authentifizierung der Devices, zum Beispiel auf der Basis von Token oder Zertifikaten.
- **Sichere Kommunikation:** Sie bietet einen sicheren Kommunikationskanal, über den die Daten mit den IoT-Geräten ausgetauscht werden können.
- **Weiterleitung der Daten:** Einige Daten können direkt in der IoT-Cloud ausgewertet und verarbeitet werden, andere müssen an Dienste zur weiteren Verarbeitung weitergeleitet werden. In der Abbildung sind diese Dienste als Backend-Services einer bestehenden Unternehmens-IT ausgeführt. In diesem Fall würde es sich um eine industrielle IoT-Lösung handeln. Einfachere Anwendungen leiten die Daten lediglich an eine Software zur Auswertung und visuellen Anzeige weiter.

Im Falle des IoT-Buttons muss entschieden werden, welche Reaktionen beim Betätigen des Tasters ausgelöst werden. Zum Beispiel könnte ein weiteres IoT-Device in Form eines Aktors angesteuert werden.

Neben diesen Aufgaben sollte sich die IoT-Cloud möglichst „kommunikativ“ geben, das heißt, mit einer Vielzahl von Geräten funktionieren. Sie sollte so flexibel sein, dass sie Programmierschnitt-

stellen für unterschiedliche Systeme der IoT-Devices anbietet. Idealerweise werden diese Schnittstellen über leicht integrierbare SDKs (Bibliotheken) in verschiedenen Zielsprachen (zum Beispiel C, Python, Java) zur Verfügung gestellt. Ebenso sollte eine generische Kommunikation via REST-Interface (siehe **Kasten:** „Was ist REST?“) möglich sein. Die IoT-Cloud sollte auch unterschiedliche Protokolle, zum Beispiel HTTPS, AMQP oder MQTT zur Kommunikation mit den Devices unterstützen.

### IoT-Cloud-Services

Die großen Cloud-Anbieter stellen auch Funktionen zur Verfügung, die für die spezialisierten Anforderungen des IoT angepasst sind. Marktführer von IoT-Cloud-Lösungen sind die Plattformen von Windows Azure, Amazon Web Services, Qloud und Oracle Cloud. Die Wahl der richtigen Plattform ist nicht einfach und stark von den jeweiligen Einsatzziele abhängig [1]. Folgende Kriterien sind bei einer Auswahl einzubeziehen:

- **Plattform-Features:** Welche Funktionen werden genau angeboten, unter anderem Device-Management, Authentifizierung und Autorisierung sowie Datenweiterleitung.
- **Plattform-Eigenschaften:** Dazu gehören die Aspekte des Datenschutzes, der Verschlüsselung und der Bedienoberfläche (Dashboard).
- **Anbindung:** Unterstützte Programmiersprachen, SDKs und Schnittstellen.
- **Sonstiges:** Darunter fallen Kriterien wie Community und Dokumentation,

Support, Kosten und Preismodelle. Die meisten IoT-Cloud-Services können bis zu einem bestimmten Nutzungsgrad kostenfrei getestet werden. Auf den Internetseiten der Anbieter finden sich in der Regel dokumentierte Beispiele in Form von Quellcode und zur Einrichtung (Konfiguration).

### IoT-Device und Anbindung

Die Verbindung zwischen den IoT-Devices und dem Cloud-Dienst erfolgt über das öffentliche Internet. Dabei spielt es zunächst keine Rolle, auf welche Weise diese Verbindung hergestellt wird. Neben einer kabelgebundenen direkten Verbindung (LAN) stationärer Devices kann die Anbindung auch kabellos über das lokale Netzwerk (WLAN) oder über Funk (LTE) erfolgen; es ist eine Frage der Anwendung, der vorhandenen Infrastruktur, der ausgewählten Hardware und auch des Energieverbrauchs. Bei unserem IoT-Button gehen wir von einer möglichst einfachen Integration in ein lokal vorhandenes WLAN aus. Da der IoT-Button flexibel nutzbar und das Gehäuse kompakt (beispielsweise in Form eines Lichtschalters für eine Stehlampe) sein sollte, gibt es klare Vorgaben an die Hardware. In Frage kommen daher nur Hardwareplattformen (Boards) mit einer leichten WLAN-Konnektivität zum Netzwerk und einem möglichst geringen Energiehunger, besonders im Standby-Modus. Auf diese Aspekte werden wir im zweiten Teil noch genauer eingehen, wenn es um das Experimentieren und erste Prototypen geht.

Hat man diese Beschränkungen hinsichtlich Baugröße, Energieversorgung und kabelloser Anbindung an das Netzwerk nicht, dann ist die Auswahl der Plattform für das IoT-Device deutlich größer

und einfacher. Soll der IoT-Button zum Beispiel dauerhaft an einem Ort platziert werden, dann kann dieser mit Netzspannung versorgt werden, der Energiebedarf im Standby-Modus ist weniger kritisch und eine LAN-Anbindung ist auch möglich.

### Anbieter von IoT-Clouds

Jetzt muss noch ein Anbieter für IoT-Cloud-Services ausgewählt werden. Der Dienst *Azure IoT Hub* von Microsoft [2] ist für eine bidirektionale Kommunikation zwischen IoT-Geräten und dem Cloud-Portal Azure vorgesehen. Es ist ein in der Cloud gehostetes Lösungs-Back-End, an das man eine Vielzahl von Geräten binden kann. Die Integration der Devices erfolgt über SDKs, die für verschiedene Plattformen und Programmiersprachen zur Verfügung stehen. Ebenso bietet der IoT-Hub Funktionen zum Managen der angeschlossenen Geräte an. Dabei hat jedes Gerät seine eigene Identität mit Zugangsdaten oder Zertifikaten, jedes Gerät kann unabhängig von den anderen über das Dashboard direkt im Browser aktiviert oder deaktiviert werden. Diese Funktionalität wird auch über APIs angeboten, um Entwicklern die Möglichkeit zu geben, IoT-Devices in eigenen Applikationen zu verwalten. Die Online-Dokumentation bietet umfassende Schritt-für-Schritt-Anleitungen: Unter [3] wird zum Beispiel beschrieben, wie man einen Raspberry Pi an den IoT-Hub anbindet und Daten vom Device entgegennimmt. Die Nutzung des Azure-IoT-Hubs ist für das Prototyping sowie die kommerzielle oder private Nutzung kostenfrei, solange man mit einer Beschränkung der Anzahl von Calls leben kann. Der Tarif *Standard* verursacht keine Kosten, sofern man

nicht mehr als 8.000 Nachrichten pro Tag mit dem Device austauscht und die Größe einer Nachricht 0,5 KB nicht überschreitet.

In die gleiche Richtung geht das Angebot von Google mit dem Dienst *Cloud IoT Core* [4]. Es handelt sich um einen vollständig verwalteten Dienst, mit dem man Devices über das Internet verwalten und Daten austauschen kann. Cloud IoT Core kann dabei mit anderen Cloud-Services von Google interagieren, zum Beispiel mit Google Big Data Analytics und ML-Diensten wie Dataflow, BigQuery, BigTable, ML, Data Studio oder BI-Tools von Partnern. Das ermöglicht eine effiziente Auswertung, Verarbeitung und Visualisierung der IoT-Daten in Echtzeit. Der Dienst unterstützt die üblichen MQTT- und HTTP-Protokolle, wodurch eine große Kompatibilität mit vielen Devices gegeben ist. Eine Kernfunktion von Cloud IoT ist der integrierte Gerätemanager. Die verwalteten Devices können über eine Konsole angepasst werden, alternativ ist auch eine Steuerung über Programmcode möglich. Der Gerätemanager ermittelt die Identität eines Geräts und sorgt für eine eindeutige Authentifizierung. Zur sicheren Datenübertragung wird auf eine so genannte End-to-End-Verschlüsselung gesetzt. Cloud IoT Core ist technisch gesehen ebenfalls ein serverloser Dienst, der bei einem quantitativen Anwachsen der Anforderungen ohne merkliche Verzögerungen horizontal skaliert. Technisch wird auf ein REST-Interface gesetzt und damit eine weitgehende Unabhängigkeit von den Systemen der Devices erreicht. Die Kosten des Dienstes richten sich nach dem Verbrauch und sind gestaffelt. Bis zu einem monatlichen Datenvolumen von 250 MB ist Cloud

Anzeige

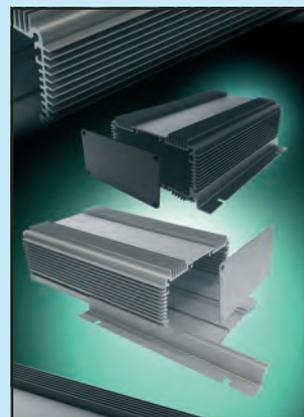
**M** HAMMOND  
MANUFACTURING®

**W**

## Profil - Gehäuse Standard und Kühlkörper

Mehr als 5000 verschiedene Gehäusedesigns.  
[hammfg.com/electronics/small-case](http://hammfg.com/electronics/small-case)

+ 44 1256 812812  
[sales@hammondmfg.eu](mailto:sales@hammondmfg.eu)

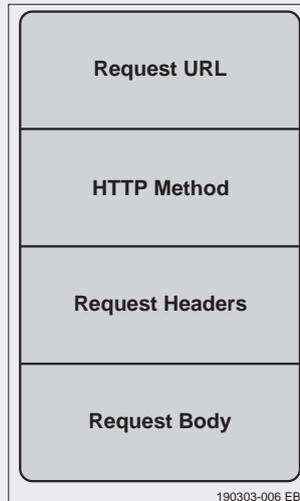


## Was ist REST?

Werden Client und Server über das Netzwerk (Internet) miteinander vermittelt, dann basiert eine solche API meist auf der Basis von REST [9][10]. REST steht als Abkürzung für *REpresentational State Transfer*. RESTful APIs basieren auf standardisierten Verfahren wie HTTP/S, URI, JSON oder XML. Folgende Prinzipien sind maßgebend:

- **Client-Server-Modell:** Die Kommunikation erfolgt auf der Basis eines Client-Server-Modells. Das Ziel ist eine flexible und generische Nutzung der Services über Plattformgrenzen hinweg.
- **Zustandslosigkeit:** Die Kommunikation ist stets zustandslos. Jede Anfrage vom Client an den Server muss vollständig sein. Der Server kann auf keine Daten vorheriger Anfragen zurückgreifen.
- **Caching:** Clients können Antworten des Servers speichern (cacheable). Damit können Netzausfälle überbrückt oder ein zeitweiliger Offline-Betrieb ermöglicht werden. Diese zwischengespeicherten Daten können dann bei einer erneuten Anfrage alternativ statt einer neuen Antwort vom Server verwendet werden.
- **Einheitlichkeit:** Die Services nutzen eine einheitliche Schnittstelle, welche vom bereitgestellten Dienst entkoppelt ist.

Ein REST-API wird in der Regel per http beziehungsweise https umgesetzt. Die Services werden per URL und den http-Methoden wie GET, POST, PUT genutzt, bei denen der Client eine Anfrage (*Request*) an den Server sendet und eine Antwort (*Response*) erhält. Eine Anfrage besteht aus



Aufbau einer REST-basierten Anfrage.

vier Bestandteilen: Endpunkt (*endpoint*), Methode (*method*), Header (*header*) und Daten (*data*). Der Endpunkt setzt sich aus dem *root-endpoint*, dem *path* und möglichen Parametern zusammen. Beispielsweise ist `https://api.github.com` der *root-endpoint* von GitHub und `/users/veikkoef/repos` der *path* zu den Repositories des Autors. Zusammen ergibt dieses `https://api.github.com/users/veikkoef/repos`. Es können noch Parameter folgen, zum Beispiel `?query1=value1&query2=value2`.

Bei den Methoden hat man die Wahl zwischen *GET*, *POST*, *PUT/PATCH* und *DELETE*. *GET* steht für Read-Operationen, also das Lesen von Daten. Da *GET*-Requests nur lesend auf den Server zugreifen, können sie beliebig oft abgeschickt werden. *GET* ist die standardmäßige Methode. *POST* wird für Create-Operationen, also das Erzeugen eines Datensatzes eingesetzt. *POST* ist nicht frei von Seiteneffekten. Es können durch

einen *POST*-Aufruf Felder in einer Datenbank verändert oder Prozesse auf dem Server gestartet werden. *PUT/PATCH* verwendet man für Update-Operationen und *DELETE* für das Löschen von Daten.

Das dritte Element einer Anfrage ist der *Header*. Der Header dient zur Bereitstellung von Informationen für den Client und den Server. Dieser kann für viele Zwecke verwendet werden, zum Beispiel zur Authentifizierung und Bereitstellung von Informationen über den nachfolgenden Inhalt.

Abschließend und als letztes Element einer Anfrage folgen die Daten (*Body*), welche man an den Server übermitteln möchte. Der Body-Bereich ist nur relevant für die Operationen *POST*, *PUT*, *PATCH* oder *DELETE*.

IoT Core von Google kostenfrei. Damit kann man diesen Dienst gut für Experimente, Prototyping, private oder kleinere kommerzielle Projekte einsetzen. Ähnli-

che Angebote – aber nahezu ausschließlich für industrielle Kunden - bieten die Cloud Services für IoT von Oracle [5], die AWS IoT Services von Amazon [6] und

die Plattform Q-loud IoT [7].

Nicht unerwähnt dürfen die Angebote von *ThingsBoard* [8] bleiben. Sie bieten eine Open-Source-Cloud-Plattform für die

### Weblinks

- [1] Plattformen im Vergleich: [www.informatik-aktuell.de/betrieb/virtualisierung/iot-in-der-cloud-erkenntnisse-und-erfahrungen-eines-plattformvergleichs.html](http://www.informatik-aktuell.de/betrieb/virtualisierung/iot-in-der-cloud-erkenntnisse-und-erfahrungen-eines-plattformvergleichs.html)
- [2] Microsoft Azure: <https://azure.microsoft.com/de-de/services/iot-hub/>
- [3] RPi in Azure: <https://docs.microsoft.com/de-de/azure/iot-hub/iot-hub-raspberry-pi-kit-c-get-started>
- [4] Google-Cloud: <https://cloud.google.com/iot-core/>
- [5] Oracle-IoT-Cloud: [www.oracle.com/internet-of-things/](http://www.oracle.com/internet-of-things/)
- [6] Amazon-IoT-Cloud: <https://aws.amazon.com/de/iot/>
- [7] Q-loud-Cloud: [www.q-loud.de/](http://www.q-loud.de/)
- [8] Thingsboard: <https://thingsboard.io/>
- [9] REST: [www.codecademy.com/articles/what-is-rest](http://www.codecademy.com/articles/what-is-rest)
- [10] REST API: <https://restfulapi.net/>

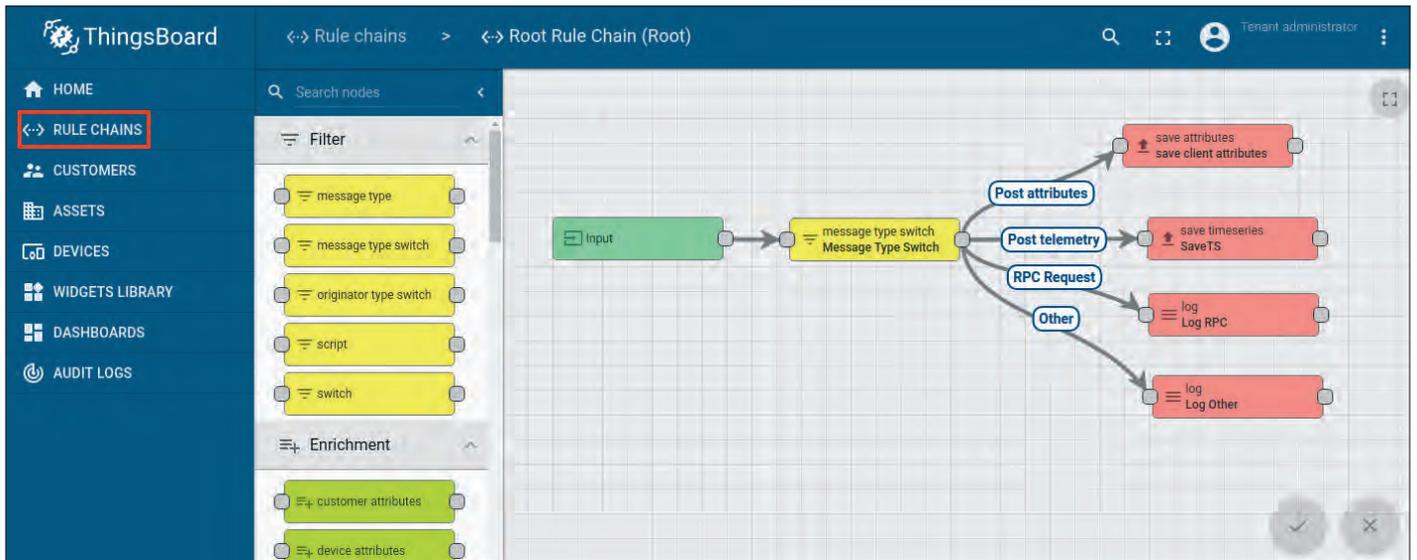


Bild 2. Die Verarbeitung von Ereignissen lässt sich grafisch definieren (Quelle: ThingsBoard).

Zwecke des IoT. Die Nutzung ist vollständig kostenfrei, wenn man die zugehörige Software auf dem eigenen Rechner installiert. Man erstellt gewissermaßen sein eigenes IoT-Back-End, was als „On Premise“ bezeichnet wird: Man ist für die Installation, Wartung und den Betrieb selbst verantwortlich. Interessant ist die Vielfalt der unterstützten Systeme, unter anderem läuft das Ganze auf diversen Linux-Distributionen (Ubuntu, CentOS, Red Hat), Windows oder alternativ auf einem Raspberry Pi 3. Letzteres dürfte gerade für Maker- und Community-Projekte und für das Prototyping interessant sein. Die Software von ThingsBoard wird auch als verwaltete Cloud-Lösung angeboten, deren Kosten abhängig vom Nutzungsvolumen gestaffelt sind, beginnend mit 10 US-\$ pro Monat. Zu den Funktionen dieses Server- oder Cloud-dienstes gehören die typischen Aufga-

**IM ELEKTOR-STORE**

→ Buch „Mein Weg in das IoT“: [www.elektor.de/mein-weg-in-das-iot](http://www.elektor.de/mein-weg-in-das-iot)

ben eines IoT-Backends, also Verwaltung von Geräten, Sammeln und Visualisieren der Daten, Ereignisverarbeitung und Remote-Prozeduraufrufe zur Steuerung der Devices. Die Funktionen werden auch über eine REST API angeboten und sind damit system- und geräte-neutral nutzbar. Über die so genannte Rule Engine kann komfortabel auf grafischer Ebene definiert werden, auf welche Weise eintretende Ereignisse verarbeitet werden (**Bild 2**).

**Fazit und Ausblick**

Sie haben jetzt einen Überblick über die Architektur einer IoT-Lösung erhalten,

deren zentraler Bestandteil ein Cloud-Dienst zur Verwaltung der IoT-Devices und zur Entgegennahme und Weiterleitung der Daten ist. Im kommenden zweiten Artikelteil wollen wir dieses Wissen praktisch erproben. Es wird um den Entwurf eines Prototypens für ein IoT-Device in Form eines IoT-Buttons gehen. Auch wenn es solche Lösungen schon fix und fertig im Handel zu kaufen gibt, lohnt sich das Experimentieren, denn man begreift die Zusammenhänge viel besser und kann eigene Lösungen leichter konzipieren. ◀

190303-01

Anzeige

**EMS PROTO**  
Rapid Prototyping & Electronics Manufacturing Services

BERECHNEN  
SIE IHREN  
PREIS  
**ONLINE**

**Leiterplatten online  
konfigurieren & bestellen**

[www.emsproto.com](http://www.emsproto.com)

LIEFERUNG  
**2 bis 12**  
TAGE

ANZAHL  
**1 bis 50**  
STÜCK

# BASIC für ESP32/ESP8266

## Programmierung mit Annex WiFi RDS



Von **Peter Neufeld**

Annex WiFi RDS (Rapid Development Suite) ist eine Entwicklungsumgebung, die ursprünglich für die Verwendung der Programmiersprache BASIC auf preiswerten ESP8266-Modulen entwickelt wurde. Inzwischen gibt es auch eine ESP32-Version davon. Wer bislang die 8-Bit-Controller von Atmel/Microchip mit AVR-BASIC programmiert hat, dem erschließt sich nun die 32-Bit-Leistungsklasse.

Da ein ESP32 praktisch ein ESP8266 mit mehr Power ist, wurde nach der Vorstellung der ersten Version für den ESP8266 auch eine funktionsgleiche Version für den ESP32 konzipiert. Während der Erstellung dieses Artikels ist sie zwar noch im Beta-Stadium, sie sollte aber im März final sein.

Wer gerne kleine Projekte mit Netzwerk-Anbindung entwickelt, muss sich nun nicht mehr zwingend mit der Arduino-IDE und C/C++ herumschlagen. BASIC ist einfach simpler. Außerdem ist die Verwendung dieser BASIC-Variante für die nicht kommerzielle, private Nutzung nicht mit Kosten verbunden. Vielleicht reizt es Sie, das einmal auszuprobieren?

### Annex-BASIC

Annex WiFi RDS [1] basiert auf dem ursprünglichen Konzept von ESPbasic [2], an dem der Entwickler Francesco Ceccarella alias chicciobc damals mitgearbeitet hat. Annex ist allerdings nicht einfach nur eine andere BASIC-Variante, sondern eine komplett überarbeitete Suite, die verbesserte Funktionen, eine höhere Zuverlässigkeit und eine optimierte Nutzung der Fähigkeiten der Espressif-Controller bietet. Außerdem wurde auch eine angemessene Dokumentation erstellt.

Der Funktionsumfang der IDE mitsamt BASIC-Interpreter ist erstaunlich (siehe **Kasten Features**). Die komplette Software steckt dabei im Mikrocontroller und läuft dann im Web-Browser,

## Features

### Annex WiFi RDS beinhaltet folgende Funktionen:

- Integrierte IDE via Webserver, Nutzung mit Web-Browser (optimiert für Chrome und Firefox).
- BASIC-Interpreter mit Gleitkomma- (doppelte Genauigkeit) und Stringvariablen, mehrdimensionalen Arrays (Float und String) sowie Subroutinen.
- Syntax-Highlighting mit kontextsensitiver Hilfe.
- Programmierbarer Web- und Datei-Server.
- Unterstützt OTA-Updates (Firmware-Updates via WLAN).
- Unterstützt asynchrone Ereignisse (Interrupts, Timer, Webzugriff, UDP etc.).
- Breakpoints, sofortige Ausführung von Befehlen, Anzeige von Variablen, Einzelschritt.
- Zugriff auf alle I/O-Pins, 1-Wire, SPI, I2C, PWM, Servo, NeoPixel, USART.
- Fehlerbehandlung, Watchdog.
- TCP (HTTP GET und POST).
- UDP-Kommunikation.
- Senden von E-Mails via SMTP-SSL-Server.
- AJAX-, ESP-NOW-, MQTT und FTP-Kommunikation (Client).
- IMU/AHRS-Fusionsalgorithmen 6 und 9 DOF (Madgwick und Mahony).
- PID-Regler (4 Kanäle).
- Windows-basierte Utility-Suite (Annex-Toolkit) mit: Flasher, Datei-Manager, Backup/Restore, HTML-Konverter, Serial-Port-Monitor, OTA-Update-Server, UDP-Konsole und IP-Scan-Tool.

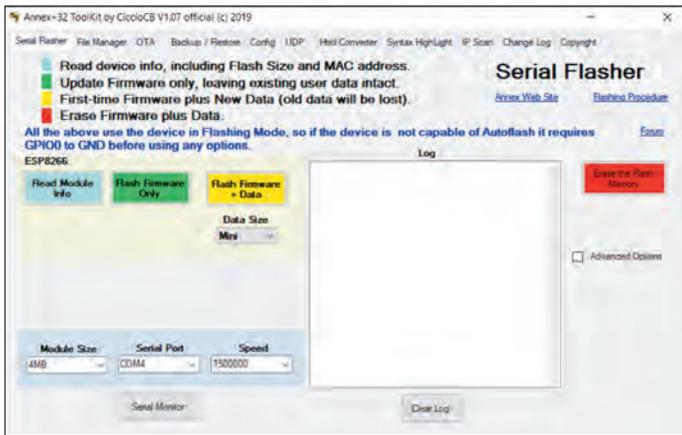


Bild 1. Annex-Tool Serial Flasher zum Übertragen der Firmware auf ein ESP-Modul.

der die integrierte Webseite anzeigt. Das ESP-Modul stellt dafür entweder einen eigenen WLAN-Access-Point und einen Webserver unter der IP-Adresse 192.168.4.1 bereit oder man bindet das Modul ins eigene WAN per (statischem) DHCP ein und notiert sich die vom Router erhaltene bzw. dort konfigurierte IP-Adresse. Bei Letzterem kann der ESP auch ohne zusätzliche Hardware-Uhr die Zeit und das Datum mit einem Internet-Time-Server synchronisieren.

## IDE

Im Gegensatz zu compiler-basierten Ansätzen für ESP-Controller wie etwa der Arduino-IDE enthält die Annex-Firmware einen festgelegten, nicht konfigurationsabhängigen, integrierten Satz von Funktionen und Treibern für gängige Peripherie (siehe **Kasten Hardware-Unterstützung**). Dies ist Segen und Fluch zugleich: Einerseits ist schon alles im Modul abgelegt, um alle gängige Hardware anzusprechen und sofort jeden dies nutzenden Skript-Code mit vollem Sprachumfang ablaufen lassen zu können – andererseits kosten diese Funktionen auch Speicherplatz und schränken daher den Umfang eigener Programme ein.

Allerdings sind die notwendigen Mittel vorhanden, um aus eigenen BASIC-Subroutinen auf der Protokollebene mit externen Geräten zu kommunizieren. So kann man z.B. auch via

## Installation

Die Installation und Inbetriebnahme im eigenen WLAN wird unter [4] sehr detailliert beschrieben. An dieser Stelle daher nur kurz das Wichtigste:

- Die Firmware und evtl. weitere Dateien werden via serielle Schnittstelle oder USB auf ein ESP-Modul übertragen und dessen Speicher evtl. zuvor gelöscht.
- Im Accesspoint-Modus (Auslieferungszustand) des ESP-Moduls einloggen oder besser das Modul als WLAN-Client in das eigene WLAN einbinden.
- Grundlegende Konfigurationen wie LAN-Parameter, Kennworte, Zeitzone/DST, Autostart-BASIC-Programm, OTA-Server etc. setzen.

Jetzt ist alles bereit für das Browser-basierte Laden, Editieren, Debuggen und Ausführen von BASIC-Programmen.

SPI oder I<sup>2</sup>C mit eigenem Code eine hier nicht aufgeführte Peripherie ansprechen.

Annex-WiFi-RDS ist nur in englischer Sprache verfügbar. Allerdings ist der Entwickler sehr aktiv und reagiert schnell auf Feedback im Forum auf der Projekt-Website [1]. Von der Projekt-Website [3] kann die IDE für Windows-Systeme als ZIP-Archiv herunter geladen und an beliebiger Stelle entpackt werden. Die darin enthaltene Applikation *ANNEX-toolkit.exe* ermöglicht dann die Installation des BASIC-Interpreters auf ESP-Modulen. Hinweise zur Einrichtung finden sich im **Kasten Installation**.

Ist die Firmware mitsamt Interpreter und IDE sowie optionalen Beispielen erst einmal auf dem ESP-Controller installiert und dieser per WLAN erreichbar, kann künftig alles komplett auf dem PC im Fenster des Browsers erledigt werden, der die IDE-Webseite im Controller geladen hat. Die Software ist speziell für Firefox und Chrome optimiert. Zusätzlich kann auch mittels Annex-Toolkit über die serielle Schnittstelle im laufenden Entwicklungs-Betrieb kommuniziert werden, was zur Fehlersuche sehr hilfreich ist. Parallel zur Ausgabe von Daten über die serielle Schnittstelle erfolgt auch eine Ausgabe in einen *Wlog* genannten Bereich der IDE, was die Entwicklung z.B. auch

## Hardware-Unterstützung

**Die folgenden Geräte, Aktoren und Sensoren werden direkt mit dedizierten Befehlen und Funktionen unterstützt:**

- DHT11, DHT21 und DHT22 Temperatur/Feuchtesensoren.
- DS18B20 Temperatursensor.
- BME280 Temperatur-, Feuchte- und Luftdrucksensor.
- APDS9960 Distanz-, Licht- (Intensität und Farbe) sowie Gestensensor.
- BNO055 absoluter Orientierungssensor.
- HC-SR04 Ultraschallsensor (Distanzmessung).
- DS1307 und DS3231 RTC-Uhr-Modul.
- PCA9685 PWM/SERVO Modul.
- LCD mit HD44780 via I2C; 1/2/4 Zeilen zu je 16/20 Zeichen.

- LCD mit ST7920 mit 128 x 64 Pixel monochrom.
- Grafisches OLED-Display mit SSD1306 oder SH1106; 128 x 64 Pixel monochrom.
- Grafisches LCD mit ILI9341; 320 x 240 Pixel in 16 Bit Farbe.
- 7-Segment-Display mit TM1637; 4 Stellen.
- 7-Segment-Display mit TM1638; 8 Stellen plus 8 LEDs und 8 Taster.
- 7-Segment-Display mit MAX7219; 8 Stellen.
- Dot-Matrix-Display mit MAX7219; 8 x 8 Dots.
- Farb-LEDs Neopixel WS2812.
- Dot-Matrix-Display mit Neopixel WS2812; 8 x 8 Dots.
- Infrarot-Interface; bidirektional (gängige RC-Protokolle).

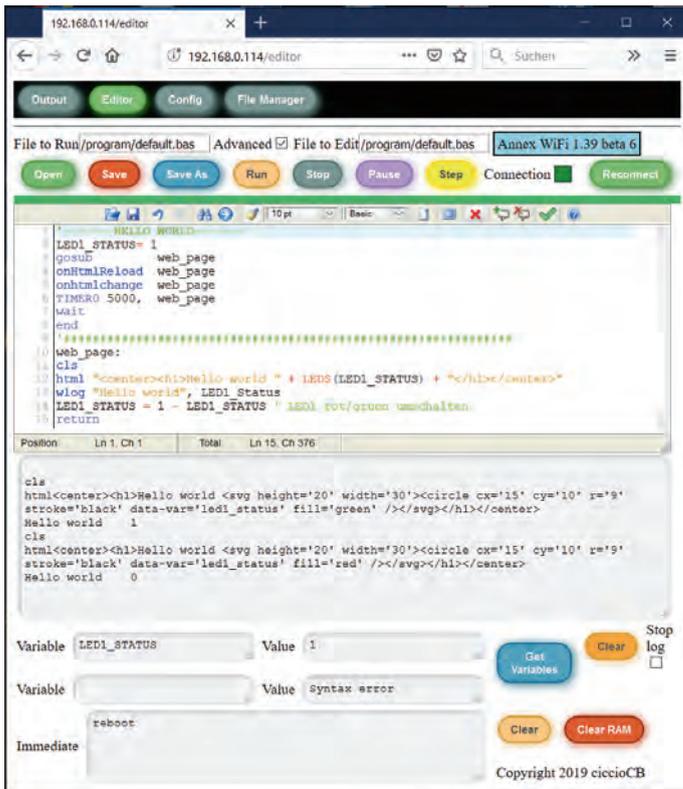


Bild 2. Annex-Editor im Browser-Fenster mit dem Beispiel „Hello World“.

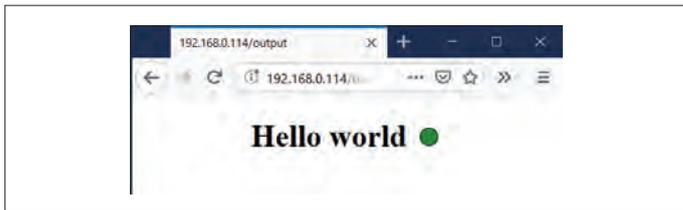


Bild 3. Ausgabe des Beispiels „Hello World“.

mit einem Notebook ohne Kabel-Verbindung zum ESP-Modul ermöglicht. Man kann so sogar mit mehreren Modulen parallel arbeiten.

### Weitere Aspekte

Damit der BASIC-Interpreter möglichst wenig RAM benötigt, wird das BASIC-Skript vom SPIFF-Speicher zur Ausführung in einen dafür vorgesehenen Bereich im Flash-Speicher kopiert, sodass nur die Liste der Programmzeilen, die Verzweigungsbezeichner

und die Liste der benutzerdefinierten Unterroutinen in den knappen Modul-Speicher geladen werden müssen. Dies ist etwas langsamer als die direkte Ausführung im RAM, schon aber das RAM zugunsten von Variablen und ermöglicht so relativ große Programme. Beim ESP32 mit seinem gegenüber dem ESP8266 vierfachen RAM ist das weniger problematisch.

Eine weitere Überlegung zur Leistung ist, dass ein ESP-Controller immer in der Lage sein muss, mehrere Aktivitäten im Hintergrund auszuführen (Webserver, Dateiserver etc.). Er benötigt daher ausreichend freies RAM für diese Aufgaben. Parallel ausgeführte Aktionen wirken sich zwangsweise auf die Ausführungsgeschwindigkeit von eigenem Code aus.

Laut dem Entwickler ist das Annex-BASIC weitgehend mit den bekannten PC-Varianten GW-BASIC und Visual Basic kompatibel und teilt viele Konzepte, Ideen und die Syntax mit diesen Vorbildern. Entsprechende Kompatibilität gibt es auch zum Projekt Micromite [5], einer BASIC-Variante für PIC32-Mikrocontroller. Die Grundlagen des Interpreters stammen vom dem Projekt MiniBasic [6] und der Texteditor aus dem Projekt EditArea [7].

Die Ausführungsgeschwindigkeit des Interpreters kann mit der von BASIC-Compilern natürlich nicht mithalten. Für die üblichen Aufgaben ist das interpretierte BASIC jedoch auf ESP-Controllern gut geeignet, ausreichend schnell, zuverlässig und dank seiner Browser-Basierung sehr einfach zu nutzen. Laut dem Entwickler ist das BASIC auf einem ESP8266 im „1980s BASIC Benchmark“ etwa zwei- bis viermal langsamer als unter Micromite bei 48 MHz Takt. Relativ zu dem, was diese IDE alles von Hause aus mitbringt, ist das keine schlechte Leistung.

Im Artikel „Sanduhr mit ESP8266 und Annex WiFi RDS“ in der nächsten Elektor-Ausgabe findet sich ein instruktives Praxis-Beispiel auf der Basis dieser ansprechenden IDE. Die Version für ESP32 wurde mit einigen für dieses SoC sinnvollen Erweiterungen ausgestattet. ◀

190400-01

**IM ELEKTOR-STORE**

→ ESP32 solo  
www.elektor.de/esp32-solo-1-wifi-bt-ble

→ ESP32 DevKitC  
www.elektor.de/esp32-devkitc

→ Elektor ESP32-Praxis-Bundle  
www.elektor.de/elektor-esp32-praxis-bundle

### Weblinks

- [1] Annex WiFi RDS: <https://sites.google.com/site/annexwifi/home>
- [2] ESPbasic: <http://www.esp8266basic.com>
- [3] Downloads: <https://sites.google.com/site/annexwifi/downloads>
- [4] Erste Schritte: <https://sites.google.com/site/annexwifi/home/first-steps>
- [5] Micromite: <http://geoffg.net/micromite.html>
- [6] MiniBasic: <https://malcolmmclean.github.io/minibasic/web/MiniBasicHome.html>
- [7] EditArea: [www.cdolivet.com/editarea/](http://www.cdolivet.com/editarea/)

# ESP32-Türklingel mit Telegram-Nachricht



Wenn der Postbote nur einmal klingelt...

Von **Luc Lemmens** (Elektor-Labor)

Sie sind zu Hause und ein Besucher steht vor der Tür, aber aus irgendeinem Grund hören Sie die Türklingel nicht? Das ist besonders ärgerlich, wenn Sie daheim auf ein Paket warten. Hier kommt die Lösung: Wenn es klingelt, sendet unsere Schaltung eine Nachricht an Ihr Smartphone oder Ihren Computer, auf dem eine App namens Telegram läuft.

Paketzusteller und auch einige Postboten haben in der Regel nicht die Geduld und die Zeit, vor ihrer Haustür eine Weile zu warten oder zweimal zu klingeln. Später am Tag finden Sie dann in Ihrem Briefkasten einen Zettel, der entweder den

Abholort des Pakets oder eine Alternative für einen zweiten Zustellversuch angibt.

### Telegram für Sie!

Sobald es an der Tür klingelt, sendet die Schaltung, die wir gleich beschreiben

werden, eine Nachricht an Ihr Smartphone oder Ihren Computer, auf dem eine App namens Telegram läuft. Der *Telegram Messenger*, so die Macher, „ist eine Cloud-basierte Mobil- und Desktop-Messaging-App mit dem Fokus

## INFOS ZUM PROJEKT

ESP32

Arduino IoT

M5Stack

Einsteiger

Fortgeschrittene

Experte

rund 3 Stunden

Arduino-IDE,  
normale Laborwerkzeuge

etwa 85 €

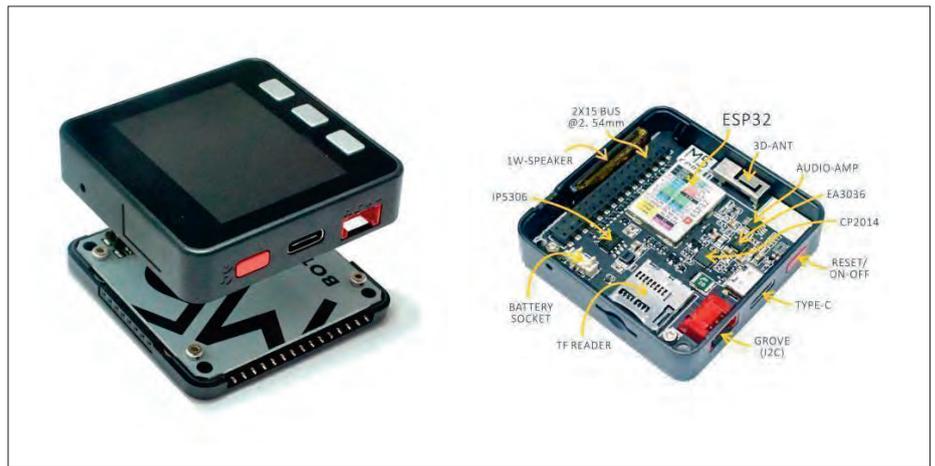


Bild 1. Das Modul M5Stack ESP32 Basic Core aus dem Elektor-Store.

auf Sicherheit und Geschwindigkeit“. Wie viele massentaugliche Kommunikationswerkzeuge im Internet gibt es Telegram in verschiedenen Formen, nämlich mobile Apps für

- iPhone/iPad,
- Android und
- Windows-Smartphones

sowie Desktop-Programme für

- PC/Mac/Linux,
- MacOS und
- Webbrowser.

Telegram ist ein Instant-Messaging-Dienst, der in seiner Funktionalität WhatsApp nicht unähnlich ist. Beide Dienste sind mit Ihrer Handynummer verknüpft. Darüber hinaus bietet Telegram auch die Registrierung Ihres Benutzernamens an, so dass Ihr Account mit Ihnen migrieren kann, wenn Sie eine neue Handynummer erhalten. Die Unterschiede zwischen den beiden Messaging-Diensten sind mehr oder weniger subtil, und wie zu erwarten, gibt es im Internet jede Menge Vergleiche, Bashing, Shitstorms, Glorifizierung, Fans, Hasser und Diskussionen. Wenn Sie jetzt neugierig geworden sind, hören Sie für einen Moment auf, den Artikel zu lesen und erleben Sie statt dessen die spärliche, wenn nicht gar spartanische, Telegram-Website mit ihren monochromen Cartoons [1].

Zurück zur Türklingel. Sofern Ihre Telegram-App aktiv und mit dem

Telegram-Server verbunden ist (per WLAN oder Mobilfunknetz), macht sie Sie auf einen Besucher vor Ihrer Haustür aufmerksam, egal in welcher Ecke der Welt Sie sich herumtreiben. Wenn Sie sicher sind, wer es ist und dieser Person vollkommen vertrauen, können Sie sogar eine „Nachricht“ zurückschicken, die einen elektrischen Türöffner betätigt, so dass Sie nicht einmal zur Haustür eilen müssen.

### Telegram trifft ESP32

Wir werden sehen, dass Telegram einfach einzurichten und mit einem „Bot“, einem Nutzer-Account für unsere automatisierte intelligente Anwendung, ausgestattet werden kann. In unserem Fall heißt das Fernkommunikation und -steuerung mit Hilfe eines ESP32, um eine Nachricht zu senden, wenn eine Taste gedrückt wird und um (möglicherweise) eine Antwort zu geben, die ein Relais aktiviert. Natürlich sind die Möglichkeiten solcher Bots nicht auf diese relativ einfache Anwendung beschränkt - das Konzept lässt sich leicht an die persönlichen Bedürfnisse anpassen und erweitern.

Der ESP32 (und sein Vorgänger namens ESP8266) wird in vielen Elektor-Projekten eingesetzt und in Elektor-Büchern ausführlich behandelt. Hier verwenden wir den ESP32 als ein intelligentes Modul, das über WLAN mit der Außenwelt interagiert. Zusammen mit der bekannten Arduino-Entwicklungsumgebung und der Bot-Bibliothek *UniversalTelegram* sollte die Entwicklung solch simpler Anwendungen wie unserer Türklingel ein Kinderspiel sein. In Wirklichkeit aber mussten wir uns während der Entwicklung dieses Projekts mit einem höchst ungewöhnlichen Phänomen auseinandersetzen, das

letztendlich viel mehr Entwicklungszeit in Anspruch nahm als gedacht (siehe Kasten **Unheimliche Begegnung**).

### Die Hardware: M5Stack ESP32 Basic Core

Dieses Projekt kann mit geringfügigen Modifikationen der Software mit jedem ESP32- oder ESP8266-Modul realisiert werden. Wir haben uns für das entzückende *M5Stack ESP32 Basic Core Module* entschieden, das im Elektor-Store erhältlich ist (siehe **Bild 1** und Kasten „Im Elektor-Store“). Das kompakte Modul ist in einem schicken kleinen Gehäuse eingebaut, zusammen mit

- einem Grafikdisplay,
- einem Satz Drucktasten,
- einem Lautsprecher und
- einer wiederaufladbaren Batterie.

Die I/O-Leitungen des *M5Stack ESP32 Basic Core* sind über Anschlüsse von außen zugänglich. Öffnet man jedoch die Rückseite des Gehäuses und entfernt die aufgesteckte Platine mit den Steckverbindern, so findet man einen 2x15-poligen Erweiterungsanschluss, auf den man eine selbstgestrickte Hardware stecken kann. Bei kleinen Erweiterungen wie dieser passt die zusätzliche Hardware auf ein Platinchen, das genau so klein ist wie das M5Stack-Modul selbst. Mit einem zusätzlichen Gehäuseteil, für das im Web zahlreiche 3D-Druckdateien zu finden sind [2], kann man ohne großen Mehraufwand ein Projekt von einem experimentellen Prototyp in ein auch im Wohnzimmer vorzeigbares Gerät verwandeln. Kurz gesagt: Dieses M5Stack-Modul ist eine großartige Grundlage für viele ESP32-Designs, die von Projektentwicklungsplattformen bis hin zu

- Startups aufgepasst - „geschlossenen“ oder „schlüsselfertigen“ Geräten reichen. Prinzipiell können Sie mit dem M5Stack und unserem Erweiterungsboard auch eine komplette Türklingelanlage aufbauen. Der eingebaute Lautsprecher kann in der Lautstärke durchaus mit einer richtigen Türklingel konkurrieren, und eine separate Klingeltaste kann parallel zum oder anstelle des Ausgangstransistors des Optokopplers geschaltet werden. Der Klingelknopf wird so angeschlossen, dass der Logikpegel an GPIO17 invertiert, wenn der Knopf gedrückt wird. Doch dies sind Pläne für übermorgen; konzentrieren wir uns erst einmal auf eine Klingel-Erweiterung, die an eine bestehende Installation angeschlossen wird!

### Der Schaltplan

**Bild 2** zeigt die Schaltung für die Telegram-Türklingel - einfacher geht es wirklich nicht. Wir gehen davon aus, dass das M5Stack-Modul extern über seinen USB-Anschluss mit Strom versorgt wird. Die Zusatzhardware hat nur zwei Aufgaben, nämlich den elektrischen Türöffner über T1 und RE1 anzusteuern sowie den Druck auf den Klingelknopf an K1 über den Optokoppler IC1 an den M5Stack weiterzuleiten. Traditionell wird eine elektrische Türklingel mit einer Wechselspannung von 6...24 V versorgt, eine Norm gab es aber nie und wahrscheinlich wird man auch seltene exotische Varianten außerhalb dieses Bereichs finden. Wie auch immer, diese Wechselspannung liegt, solange der Klingelknopf nicht gedrückt wird, an dessen Anschlussdrähten und somit an K1.

Der Optokoppler sorgt für eine galvanische Trennung der Türklingelelektrik von der unseres ESP32-Moduls. Der Optokoppler, der durch seine antiparallel geschalteten Infrarot-LEDs ideal für den Betrieb an Wechselspannungen oder Gleichspannungen mit unbekannter Polarität geeignet ist, „übersetzt“ die Wechselspannung in einen logischen Pegel auf der GPIO17-Leitung. Im Ruhezustand, wenn nicht geklingelt wird, leitet der Fototransistor und der Pegel ist low, wird der Klingelknopf gedrückt, ist der Klingelstromkreis geschlossen und die Klingel ertönt, aber die Spannung an K1 entfällt: Der Fototransistor sperrt und R2 zieht den GPIO17-Pegel auf high. Und dann gibt es noch den BT1-Anschluss, der dazu dient, die im M5Stack-Modul serienmäßig verbaute 3,7-V-Lithium-Batterie optional (wieder)

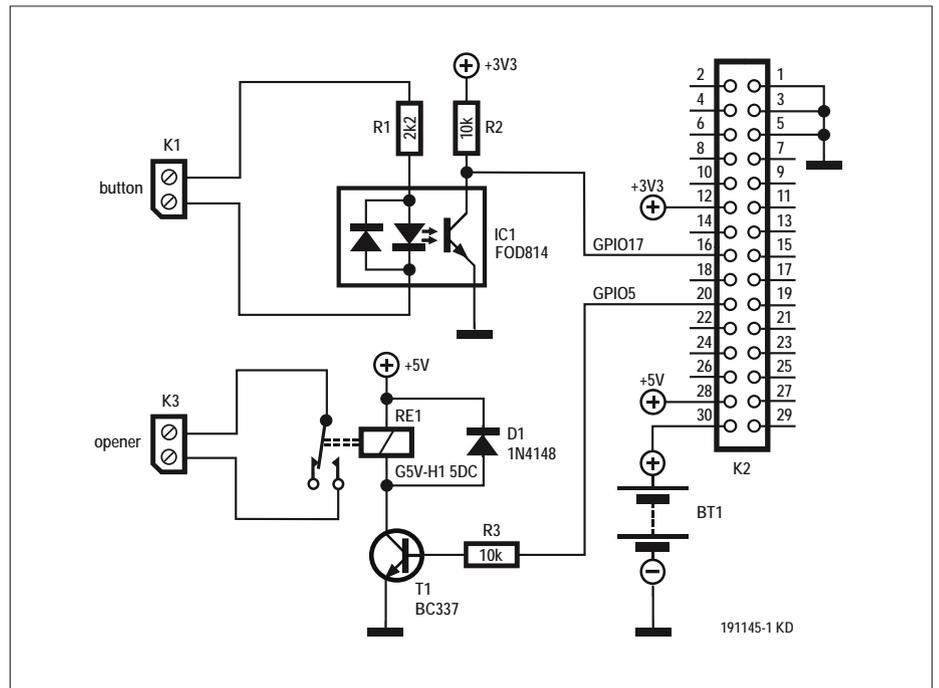


Bild 2. Schaltung der Türklingel-Erweiterung mit der leistungsstarken Kombination aus Arduino, ESP32 und dem kostenlosen Online-Messenger Telegram.

anzuschließen. Beachten Sie die Polarität: der rote Draht am Plus- und der schwarze Draht am Minuspol der Batterie. Die Erweiterungsplatine ist ausschließlich mit Durchsteckbauteilen bestückt und sollte einfach zu montieren sein. Achten Sie aber auf die Stiftleiste K2, die „falsch herum“ angebracht wird, wie man in **Bild 3** sieht. Die langen Pins werden von der Unterseite durch

die Platine gesteckt und auf der Bauteilseite verlötet. Die kurzen Stifte werden in den Erweiterungsanschluss des M5Stack Moduls gesteckt, wobei der Kunststoffkörper von K2 gleichzeitig als Abstandshalter zwischen beiden Platinen fungiert.

### Installation von Telegram

Ist die Hardware aufgebaut, muss ein Telegram-Konto auf Ihrem Handy erstellt

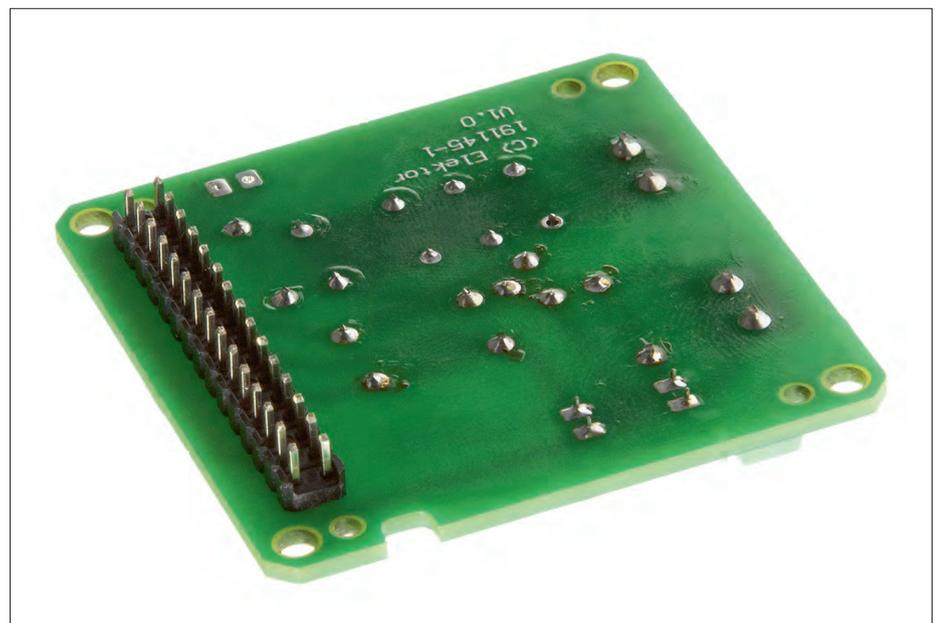


Bild 3. So sollte K2 montiert werden: Die langen Stifte werden von der Kupferseite (Unterseite der Platine) aus eingesteckt, nach oben geschoben und auf der Bestückungsseite verlötet.

## Unheimliche Begegnung der ESP32-Art

Stellen Sie sich vor: Ihr sorgfältig erstellter Quellcode wird ohne Fehlermeldungen kompiliert sowie problemlos auf den ESP32 hochgeladen und die Schaltung *scheint* zunächst wie erwartet zu funktionieren. Dann stellen Sie jedoch fest, dass der Mikrocontroller immer dann zurückgesetzt wird, wenn eine Telegram-Nachricht eintrifft! Alles geprüft, alles versucht - kein offensichtlicher Fehler zu finden. Also, Sie kompilieren und testen das Programm auf einem anderen Computer und zu Ihrem Erstaunen macht der ESP32 jetzt genau das, was man von ihm erwartet! Sie überprüfen noch einmal, ob auf beiden Computern die gleichen Versionen der Arduino-IDE und der Bibliotheken laufen - es gibt keine Unterschiede. Wir waren mit unserem Latein am Ende und haben auf dem ersten Computer „Arduino“ komplett entfernt. Keine gewöhnliche Deinstallation, die anscheinend Trümmer in versteckten Ordnern hinterlässt, sondern gründlich. Und dann installierten wir die Arduino-IDE neu und alle Probleme waren verschwunden.

Was war der Grund dafür? Wir haben immer noch keine Ahnung und wir werden uns nicht die Mühe machen, es herauszufinden. Wir haben dabei gelernt, dass sich die Arduino-IDE möglicherweise unberechenbar verhält. Auch wenn es das erste Mal überhaupt war, dass wir auf ein solches Problem gestoßen sind, war es für das Elektor-Labor auch das letzte Mal!

werden. Man lädt die Telegram-App aus dem App-Store oder Google Play herunter und installiert sie. Dabei wird das Konto mit der Handynummer verknüpft. Danach installieren Sie die für Ihr Betriebssystem passende Telegram-App auch auf Ihrem Computer, was etwas anders funktioniert, weil auch die Verbindung zwischen Telegram und dem ESP32-Sketch hergestellt werden muss.

Melden Sie sich mit der Handynummer bei der Telegram-App auf Ihrem Computer an. Die App auf dem Telefon

zeigt einen Verifizierungscode an, der wiederum in den Computer eingetippt werden muss, damit Telegram dort aktiv ist. Der Sketch auf dem ESP32 arbeitet mit einem „Bot“, ähnlich wie eine Art automatischer Anrufbeantworter. Er kann auf Befehle reagieren, die über eine Telegram-Chat-Nachricht empfangen werden, oder Nachrichten per Telegram senden, durch die Ereignisse im oder am ESP32 ausgelöst werden, zum Beispiel die Meldung eines Tastendrucks oder das Senden eines ADC-Wertes. In diesem

Projekt könnten die Befehle einen Türöffner ansteuern.

Dieser Bot muss zuerst in Telegram angelegt werden. Dazu sucht man in der Kontaktliste nach einem *BotFather*, der ebenfalls ein Bot ist, der über einen Chat-Dialog bei der Erstellung des Türklingel-Bots hilft. Der Dialog wird durch die Eingabe des Befehls `/newbot` gestartet. Man legt den Namen fest, der später in der Telegram-Kontaktliste angezeigt wird, und einen Benutzernamen für den Bot. Letzterer sollte immer mit den Buchstaben „bot“ enden. Beachten Sie den Schrägstrich ( / ) am Anfang jedes Bot-Befehls. Wenn dies erfolgreich erledigt ist, benachrichtigt der BotFather Sie und zeigt einen Code (Token) an, der benötigt wird, um den neuen Bot zu autorisieren, Anfragen an die Bot-API zu senden. Bewahren Sie dieses Token an einem sicheren Ort auf, da es jedem die Kontrolle des Türöffner-Bots gewährt. Damit Sie bei uns nicht so leicht einbrechen können, habe ich im Setup-Dialog unseres Bots (**Bild 4**) das Token unkenntlich gemacht.

Klicken Sie in dem Dialog mit BotFather auf den Link `t.me/...`, um ein Chat-Fenster für den neuen Bot zu öffnen, der ebenfalls in die Kontaktliste aufgenommen wird. Der Bot ist nun einsatzbereit, muss aber noch mit dem Arduino-Sketch der Türklingel-Erweiterung verknüpft werden.

### Arduino-Sketch

Im Software-Download dieses Projektes finden Sie den Sketch `DOORBELL.INO` mit dem Quellcode für die ESP32-Firmware [3]. Dieser Sketch kann modifiziert und muss nachgebessert werden. In seiner jetzigen Form soll er vor allem zeigen, wie Arduino/ESP32 und Telegram für ein einfaches Steuerungssystem zusammenwirken. Insbesondere die Bedienung des elektrischen Türöffners ist eine angreifbare Schwachstelle, über die sich finstere Zeitgenossen in den Bot hacken und die Tür öffnen könnten. **Elektor übernimmt keinerlei Haftung für Schäden, die aus der Anwendung dieses Projektes entstehen könnten!** Und was dem Sketch noch fehlt, ist das Klingelsignal. Dazu bindet man eine Bibliothek wie `pitches.h` ein und komponiert nach Herzenslust eine Melodie.

Im Sketch muss unser M5Stack-Modul im lokalen WLAN registriert werden und Zugriff auf den soeben erstellten Telegram-Bot erhalten. Die Zugangsda-

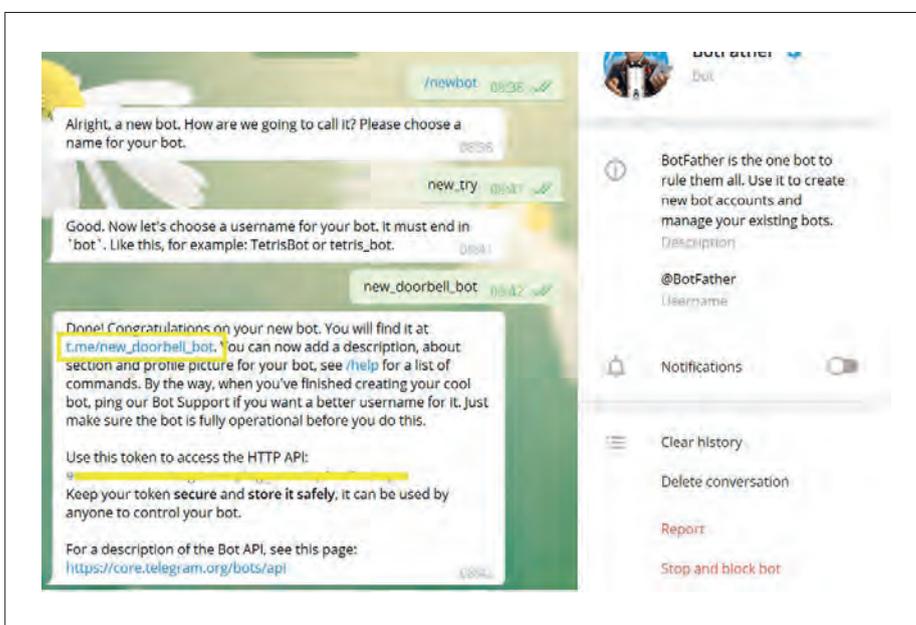


Bild 4. Der Screenshot zeigt unsere Diskussion mit dem BotFather. Das Token haben wir unkenntlich gemacht.



## STÜCKLISTE

### Widerstände:

R1 = 2k2, 5 %, 0,25 W, 250 V  
R2,R3 = 10 k, 5 %, 0,25 W, 250 V

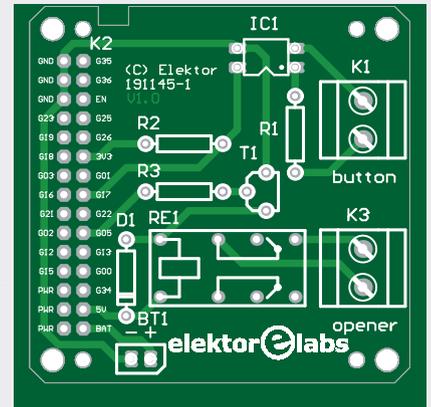
### Halbleiter:

D1 = 1N4148  
T1 = BC337  
IC1 = FOD814A, AC-Optokoppler mit Transistorausgang, 1-Kanal, DIP, 4-polig, 50 mA, 5 kV

### Außerdem:

RE1 = Relais G5V-2-H1 5DC, Spule 5 VDC  
Spule, DPDT, 1 A (Omron)

K1,K3 = 2-polige Platinenanschlussklemme,  
Raster 0,2" (5,08 mm), 630 V  
K2 = 2x15-polige Stiftleiste, vertikal (teilbar)  
M5Stack ESP32 Basic Core, aus dem  
Elektor-Store (oder gleichwertiges  
ESP32-Modul)  
Platine 191145-1 V1.0, aus dem Elektor-Store



ten Ihres WLANs werden in die Zeilen 17 und 18, das von BotFather festgelegte Token in die Zeile 21 eingetragen. Dies ist der Grund der Empfehlung, den Bot auf Ihrem Computer zu erstellen. Der umständlich lange Token kann nämlich einfach in den Sketch kopiert werden. Sobald die Installation der Arduino-IDE erfolgreich abgeschlossen ist, lässt sich der Sketch ohne Probleme kompilieren. Im Tools-Menü der IDE muss - wenn der M5Stack an einen USB-Port des Computers angeschlossen wird - das richtige Board (M5Stack-Core-ESP32) und der virtuelle COM-Port eingestellt werden, um den Sketch auf den ESP32 hochzuladen. Danach wird der Sketch gestartet und verbindet sich mit dem Netzwerk. Der Name des WLANs und die IP-Adresse der

Türklingel-Erweiterung erscheinen auf dem LCD des M5Stacks. Der Bot muss nach jedem Neustart des Sketches durch den `/start`-Befehl über das Chat-Fenster in Telegram aktiviert werden. Er gibt dann eine Rückmeldung inklusive einer Übersicht der von ihm unterstützten Befehle.

### Im täglichen Gebrauch

Wenn es an der Tür klingelt, wird eine Nachricht über die Telegram-App an Ihren Computer oder Ihr Smartphone gesendet. Sie können darauf reagieren, indem Sie die Tür manuell öffnen oder mit der Taste `/Open` einen elektrischen Türöffner (falls vorhanden) betätigen. Mit der `/Ignore`-Taste wird diese Option aufgehoben und die `/status`-Taste (überprüft

die Verbindung zum Türklingel-Extender) erscheint wieder am unteren Rand des Chat-Bildschirms. Diese Befehle können auch über das Chat-Fenster eingegeben und gesendet werden, achten Sie dabei auf Groß- und Kleinschreibung! Die Bot-Bibliothek UniversalTelegram erledigt eigentlich die meiste Arbeit in diesem Projekt. Die Bibliothek und ihre Dokumentation findet man unter [4]. Mit den Beispiel-Bots kann mit wenig Aufwand ein eigener Bot erstellt werden. Unser Türklingel-Sketch basiert auf dem Beispiel *Custom Keyboard*, das zeigt, wie man Tastensteuerungen an die Telegram-App bindet. Das ist sehr praktisch, denn man muss sich nicht die Mühe machen, selbst Steuerbefehle im Chat-Fenster einzugeben. Eine unverzichtbare Subroutine ist `handleNewMessages`, die die Verarbeitung von eingehenden Nachrichten übernimmt. In vier `if`-Anweisungen werden die Befehle und Messages abgehandelt, die unser Bot versteht:

- start,
- status,
- Ignore,
- Open.

Um Nachrichten versenden zu können, benötigt der Sketch die Chat-ID, die nach dem ersten `/start`-Befehl in der Variablen `ThisChat` gespeichert wird. Die Türklingel wird von einem Interrupt erfasst. Beachten Sie das Attribut `IRAM_ATTR`, das für die Deklaration einer ESP-Interrupt-Serviceroutine benötigt wird. ◀

191145-02



### IM ELEKTOR-STORE

→ M5Stack ESP32 Basic Core  
[www.elektor.de/m5stack-esp32-basic-core-development-kit](http://www.elektor.de/m5stack-esp32-basic-core-development-kit)

→ Leerplatine 191145-1 V1.0  
[www.elektor.de/191145-1](http://www.elektor.de/191145-1)

### Weblinks

- [1] Telegram-Messenger: <https://telegram.org/>
- [2] 3D-Druckdateien für Gehäuseteil:  
<https://github.com/m4k3r-net/M5Stack-3DPrintFiles>
- [3] Arduino-Sketch: [www.elektormagazine.de/191145-02](http://www.elektormagazine.de/191145-02)
- [4] Bot-Bibliothek UniversalTelegram:  
<https://Github.com/witnessmenow/Universal-Arduino-Telegram-Bot>



# Von der Pike auf gelernt

## Neues aus der Elektor-Ideenkiste

In der November/Dezember-Ausgabe 2019 von Elektor haben wir Ihnen eine extrem luxuriöse Softstart-Schaltung mit unzähligen Schnickschnack und ebenfalls unzähligen Teilen vorgestellt, die speziell für schwere Verstärker geeignet war. Dass es auch viel einfacher gehen kann, beweist einer der folgenden Beiträge...

Idee: Elektor-Labor

### Einfache Einschaltstrombegrenzung

Die Schaltung in **Bild 1** ist tatsächlich sehr einfach und die Funktionsweise bedarf eigentlich kaum einer Erläuterung. Nehmen wir an, Sie wollen einen schweren Verstärker einschalten, einen mit ein paar sehr dicken Elektrolytkondensatoren im Netzteil. Sie können sich vorstellen, was im Moment des Einschaltens passiert: Für kurze Zeit fließt ein sehr großer Strom, bis diese Elkos geladen sind. Und das Geräusch, das Sie in der Ferne hören, ist die Sicherung im Sicherungskasten, die rausfliegt... Wenn diese Schaltung zwischen dem Verstärker (oder einer anderen Last) eingeschleift wird, ist diese Gefahr beseitigt. Der Widerstand von 33 Ω begrenzt den Einschaltstrom auf maximal ungefähr 7 A. Durch diesen Strom fällt eine (beträchtliche) Spannung über dem Widerstand ab, so dass das 230-V-Relais nicht anziehen kann.

Nach kurzer Zeit (nur ein Sekundenbruchteil in der Größenordnung von 5...10 Perioden der Netzwechselspannung) sind die Elkos voll geladen und der Einschaltstrom reduziert sich auf den normalen Betriebsstrom der Last. Der Spannungsabfall über dem Widerstand nimmt entsprechend ab, die Spannung über der Relaispule steigt und das Relais zieht an. Dadurch wird die Reihenschaltung von Sicherung und Widerstand überbrückt und wir haben einen normalen, stabilen Betriebszustand erreicht. Der Widerstand ist ein „dicker Brummer“ mit einer Belastbarkeit von 50 W. Zwar wird er während der Einschaltspitze mit mehr als 50 W belastet, aber dies nur so kurz, dass er alles problemlos verkraftet. Ähnliches gilt für die Sicherung: Der Einschaltstromspitze ist so kurz und die Sicherung so träge, dass sie nicht auslöst.

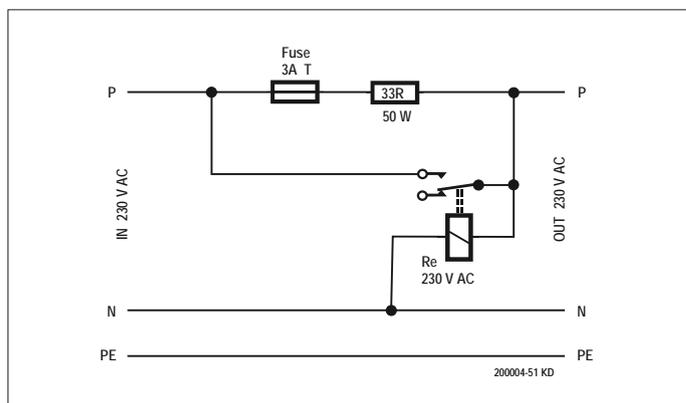


Bild 1. Diese Einschaltstrombegrenzung könnte kaum einfacher sein.

Nur wenn die Last defekt ist (beispielsweise ein Kurzschluss), bleibt der Einschaltstrom so hoch. Dann schmilzt die Sicherung, bevor der Widerstand abbrennt und eine wirklich gefährliche Situation eintreten kann.

Je nach Belastung können die Werte der Sicherung und des Widerstandes eingestellt werden.

Idee: Ton Giesberts (Elektor-Labor)

### Automatische Verstärkungsregelung

Diese Schaltung passt die Verstärkung von (kleinen) Eingangssignalen automatisch so an, dass das Ausgangssignal in einem weiten Bereich konstant bleibt. Die erforderliche Steuerspannung wird mit einer Kaskadenschaltung gewonnen.

Eine solche automatische Verstärkungsregelung (automatic gain control, AGC) kann bei der Detektion schwacher Signale sehr nützlich sein. Stellen Sie sich einen Fledermausdetektor vor: Die (Ultraschall-)Geräusche der Fledermäuse werden vom Mikrofon aufgenommen, verstärkt und in den für Menschen hörbaren Bereich verschoben. In erster Linie geht es darum, diese Signale überhaupt zu erfassen, eine „High-End-Verstärkung“ ist dabei von sekundärer Bedeutung. Für die eigentliche Verstärkung werden die beiden Hälften eines Doppel-Operationsverstärkers TL072 (IC1A und IC1B) eingesetzt. IC1A stellt einen nichtinvertierenden Verstärker dar, dessen Verstärkung

$$A1 = R4 / (R2 + R3 || T1) + 1$$

beträgt. IC1B ist dagegen als invertierender Verstärker geschaltet:

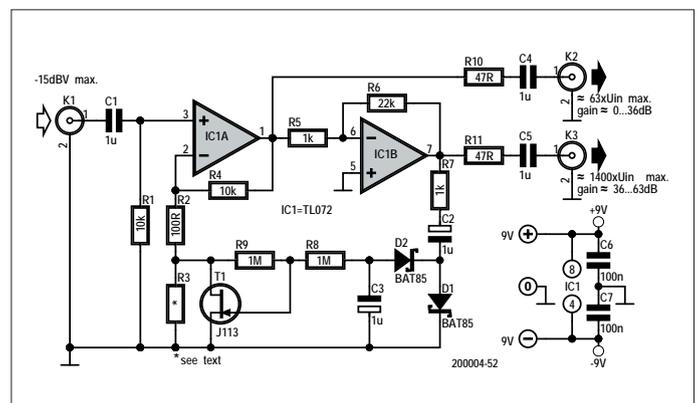


Bild 2. Eine automatische Verstärkungsregelung muss nicht extrem kompliziert sein.

$$A_2 = -R_6 / R_5$$

Der „Trick“ dieser Schaltung besteht in den beiden Dioden D1 und D2. Sie bilden zusammen mit den Kondensatoren C2 und C3 eine Kaskade (Spannungsverdoppler). Aus der Ausgangsspannung von IC1B wird eine Steuerspannung abgeleitet, mit der T1 (ein JFET) mehr oder weniger durchgesteuert wird.

Wie Sie wahrscheinlich wissen (und wenn nicht, dann wissen Sie es jetzt), kann ein JFET als variabler Widerstand verwendet werden kann. Sein Widerstandswert hängt von der an das Gate angelegten Spannung ab. In dieser Schaltung haben wir einen Widerstand R3 parallel zu JFET T1 geschaltet, um einen minimalen Verstärkungsfaktor für IC1A zu gewährleisten. An dieser Stelle kann man experimentieren - R3 entweder ganz weglassen oder mit seinem Wert spielen.

In der Schaltung (**Bild 2**) wird ein J113 als JFET verwendet. Beide Dioden sind Schottky-Dioden. Mit anderen FETs funktioniert es auch und die beiden Dioden müssen nicht unbedingt Schottkys sein; auch mit Standarddioden wie 1N4148 funktioniert es ganz gut.

Die Schaltung ist nicht sehr kritisch. Zum Experimentieren kann man sie auch auf einem Steckbrett aufbauen, für die endgültige Version darf eine Lochraster/punkt-Platine verwendet werden. Für Interessenten noch ein paar Details der Schaltung. Die Eingangsimpedanz der AGC-Schaltung wird mit R1 auf 10 kΩ eingestellt, die untere Grenzfrequenz liegt bei 16 Hz (C1). Der Wert von R2 plus der minimale Widerstand des JFETs von T1 (bei  $U_{GS} = 0$  V gemäß Datenblatt 100 Ω) bestimmt die maximale Verstärkung der Schaltung. Bei unserem Prototyp war dieser Mindestwiderstand von T1 etwas geringer (circa 60 Ω), so dass für die maximale Verstärkung der Schaltung bei  $U_{GS} = 0$  gilt:

$$A = (1 + R_4 / (R_2 + R_{GS})) \times (R_6 / R_5) \approx 1400$$

Die Geschwindigkeit, mit der die AGC reagiert (attack), wird durch R7 bestimmt und liegt in der Größenordnung von einigen Millisekunden. Machen Sie R7 nicht kleiner als im Schaltplan angegeben, da dies den Klirrfaktor am Ausgang K3 unerträglich erhöht. Die Erholungszeit (decay) der Schaltung wird bestimmt durch R8+R9, C2 und C3 und beträgt einige Sekunden.

Die Schaltung ist für die Versorgung durch zwei 9-V-Blöcke ( $\pm 9$  V) ausgelegt, die Stromaufnahme beträgt 3,5 mA. Natürlich haben wir die Schaltung in unserem Labor ein wenig getestet. In **Bild 3** sehen Sie, dass der Ausgangspegel (blaue Linie) über einen weiten Eingangsbereich (horizontale Achse) einen nahezu konstanten Wert hat.

Idee: Elex-Team

### Geradeaus-Empfänger

Zum Schluss noch ein weiterer Leckerbissen aus guten alten Zeiten: ein Geradeaus-Empfänger. Ob Sie es glauben oder nicht, mit diesem Empfänger ohne ein einziges aktives Bauteil können Sie Radiosendungen empfangen!

Das Prinzip ist denkbar einfach: Der Drehkondensator bildet zusammen mit der Spule einen Schwingkreis. Das mit ihm aufgefangene (amplitudenmodulierte) Signal wird von der Diode plus Ausgangskondensator erfasst und kann dann mit einem hochohmigen Kristallohrhörer abgehört werden.

Dieser sehr elementare Funkempfänger (so sahen die allerersten Empfänger [1] tatsächlich aus, nur haben wir hier den ursprünglichen Kristalldetektor durch eine Diode ersetzt) lädt

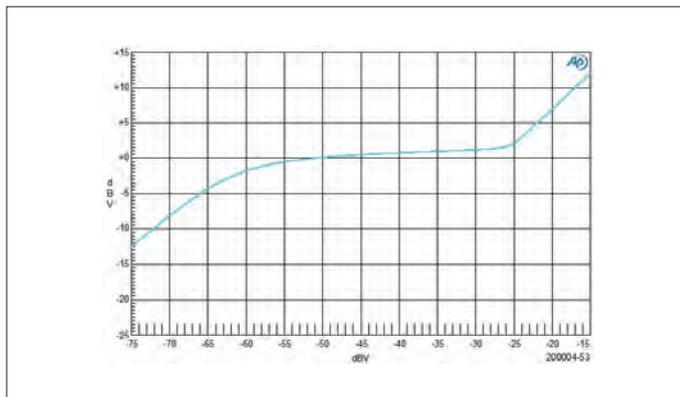


Bild 3. In der Praxis verhält sich die AGC-Schaltung überraschend gut.

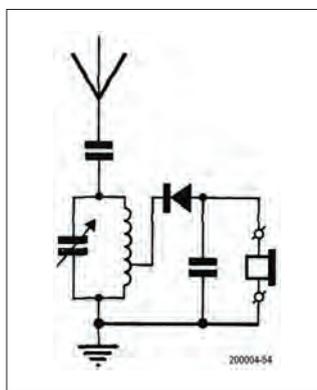


Bild 4. So begann die Geschichte des Radios...

uns zum Experimentieren ein! Für erste Versuche kann man etwa 85 Windungen emaillierten Kupferdraht (Kupferlackdraht) mit einer Stärke von 0,2 mm auf einen etwa 10 cm langen Ferritstab wickeln. Die Germaniumdiode (AA119 oder ähnlich) wird an einen Spulenabgriff angeschlossen. Der Kristallohrhörer sollte hochohmig sein (circa 2000 Ω). Der Abstimmkondensator besitzt eine Kapazität von etwa 200 pF, die Antenne ist über einen kleinen Kondensator (etwa 3,9 pF) angeschlossen.

Für den Gleichrichter Kondensator an der Anodenseite der Diode können Sie ein Exemplar von 10 nF verwenden.

Dieser Empfänger braucht definitiv eine „richtige“ Antenne - eine lange Drahtantenne von einigen Metern ist das Mindeste. Für eine einwandfreie Funktion benötigen Sie auch eine „echte“ Erde, die an ein Wasser- oder Zentralheizungsrohr angeschlossen wird!

Im nächsten Teil werden wir diesen primitiven Empfänger ein wenig mit aktiven Komponenten aufpeppen. ◀

(200004-01)

**IM ELEKTOR-STORE**

→ Buch (englische Sprache): **Electronic Circuits For All**

[www.elektor.de/electronic-circuits-for-all](http://www.elektor.de/electronic-circuits-for-all)

### Weblink

[1] [Detektorempfänger: https://de.wikipedia.org/wiki/Detektorempf%C3%A4nger](https://de.wikipedia.org/wiki/Detektorempf%C3%A4nger)

# LoRaWAN - ein einfacher Einstieg

## Mit Blue Pill, LoRa-Breakout-Board und The Things Network



Vom **Mathias Claußen** (Elektor-Labor)

Bei der Funktechnik LoRa, die von der Firma Semtech entwickelt wurde, handelt es sich um eine Datenübertragung mit hoher Reichweite bei niedrigem Energiebedarf. Dadurch eignet sich LoRa besonders für vernetzte Sensoren, die mit ihrer Energiequelle haushalten müssen. Ein beliebtes und offenes Netz, das Sensordaten empfangen und weltweit abrufbar machen kann, ist „The Things Network“. Für erste Experimente genügen ein paar kleine Boards für eine Handvoll Euro.

Ein LoRaWAN nutzt, wie der Name schon vermuten lässt, die LoRa-Funktechnik für den Transport von Daten in einem Weitverkehrsnetz (Wide Area Network, WAN). Es stellt die Gegenstelle für LoRa-Sensorknoten bereit, indem es die Daten mit sogenannten *Gateways*

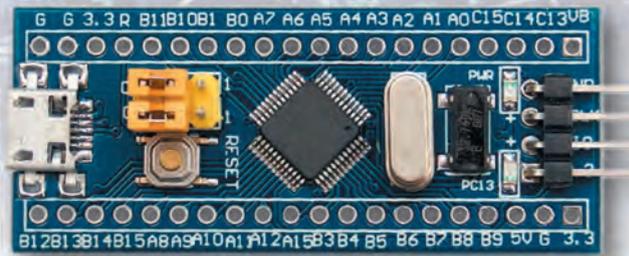
(Basisstationen) über LoRa empfängt und im Internet abrufbar macht. Ein populäres, sich stark ausbreitendes LoRaWAN ist „The Things Network“, ein freies, auf einer Community beruhendes Netzwerk mit guter Abdeckung. Um dieses Netzwerk aktiv zu nutzen, sind

nur eine Registrierung und natürlich eine passende Hardware erforderlich. Eine monatliche Grundgebühr oder eine Abrechnung pro Nachricht fallen nicht an; man muss sich lediglich an die Fair Use Policy im Netzwerk halten. Diese Policy soll allen Teilnehmern den Transport ihrer



# IN<sup>®</sup>

# THE THINGS NETWORK



Daten ermöglichen und eine Überlastung der Gateways verhindern.

### Mit kleinem Geld dabei

Um es vorwegzunehmen: Die hier vorgestellte Lösung ist nicht völlig konform zu den LoRaWAN-Spezifikationen und eignet sich deshalb nur zum Experimentieren. Die Kosten beschränken sich auf etwa 23 € für den Sensorknoten plus eventuell ein Steckbrett und ein paar Jumper-Drähte. Die Zutatenliste für den LoRa-Sensor ist recht kurz:

- Controllerboard STM32 „Blue Pill“ [1]
- LoRa-Funkmodul RFM95 [2]
- USB-Seriell-Wandler [3]
- Jumperwires
- Steckbrett

Für einen Einstieg verwenden wir ein arduinoides Blue-Pill-Board mit einem leistungsstarken STM32-Arm-Cortex-Controller darauf. Das Board ist inklusive Debugger schon für wenige Euro zu haben, bietet aber 64 kB Flash, 16 kB RAM sowie eine USB-Buchse. Letztere ist allerdings mit Vorsicht zu genießen, denn die Beschaltung auf diesen Boards ist sehr kostenoptimiert (um es positiv auszudrücken), so dass es leider immer wieder zu Problemen mit dem Port kommt. Deshalb wird die Buchse hier nur zur einfachen Spannungsversorgung herangezogen. Um den Chip zu programmieren, wird der integrierte Bootloader genutzt. Neue Software lässt sich zuverlässig über die serielle UART-Schnittstelle des Boards programmieren.

## INFOS ZUM PROJEKT



LoRa LoRaWAN  
BoB  
The Things Network



Einsteiger  
→ Fortgeschrittene  
Experte



30 Minuten



PC,  
Steckbrett und  
Jumper Wires



rund 30 Euro

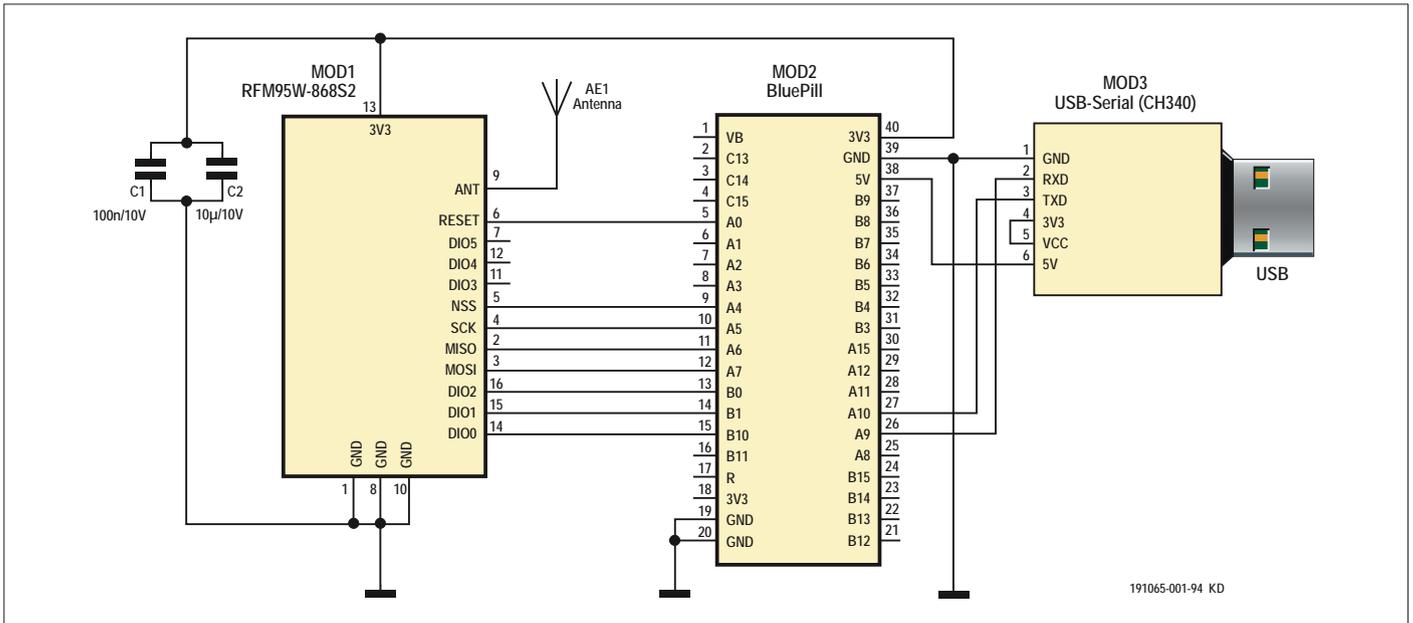


Bild 1. Schaltplan des Low-cost-LoRa-Nodes.

Die wenigen Bauteile, das LoRa-Modul, das Blue-Pill-Board und ein USB-Seriell-Wandler sind wie im Schaltplan (**Bild 1**) gezeigt miteinander verbunden. Da das LoRa-Modul mit seinem Rastermaß von 2,00 mm nicht gerade steckbrettfreundlich ist, hat das Elektor-Labor ein kleines Breakout-Board entwickelt (**Bild 2**), mit dem sich das Modul nicht nur bequem auf ein Steckbrett stecken lässt. Praktischerweise wird auch gleich die Montage der beiden Kondensatoren C1 und C2 ermöglicht. Was noch fehlt, ist eine Antenne. Ein Stück Kupferdraht mit 1 mm Durchmesser ist vollkommen ausreichend. Die

nötige Länge einer  $\lambda/4$ -Antenne für den Frequenzbereich von 868 MHz, in dem das LoRa-Modul arbeitet, ist:

$$\lambda/4 = (c_0/868 \text{ MHz})/4 = (299792458 \text{ m/s})/(868000000 * 4/s) = 0,08635 \text{ m} = 8,635 \text{ cm}$$

Diese Formel ist allerdings nur für die Ausbreitung im Vakuum gültig. Da sich das Signal im Kupferdraht ausbreitet und so langsamer ist als im Vakuum, muss ein Verkürzungsfaktor von 0,95 berücksichtigt werden. Damit erhält man als  $\lambda/4$ -Antennenlänge rund 82 mm. Wie in **Bild 3** zu erkennen ist, wird für

den Aufbau unseres LoRa-Sensors nicht viel an Hardware benötigt.

Bei den ersten Tests werden zudem keine externen Sensoren verwendet, vielmehr soll nur gezeigt werden, wie ein Sensor-knoten (auch Node genannt) eingerichtet und in Betrieb genommen wird.

### Gut vorbereitet...

Dazu ist erst einmal etwas Software nötig: die Arduino-IDE, die Bibliothek *MCCI LoRaWAN LMIC* und das Support-Paket *Arduino-Core-STM32 Board*. Die LMIC-Bibliothek lässt sich wie

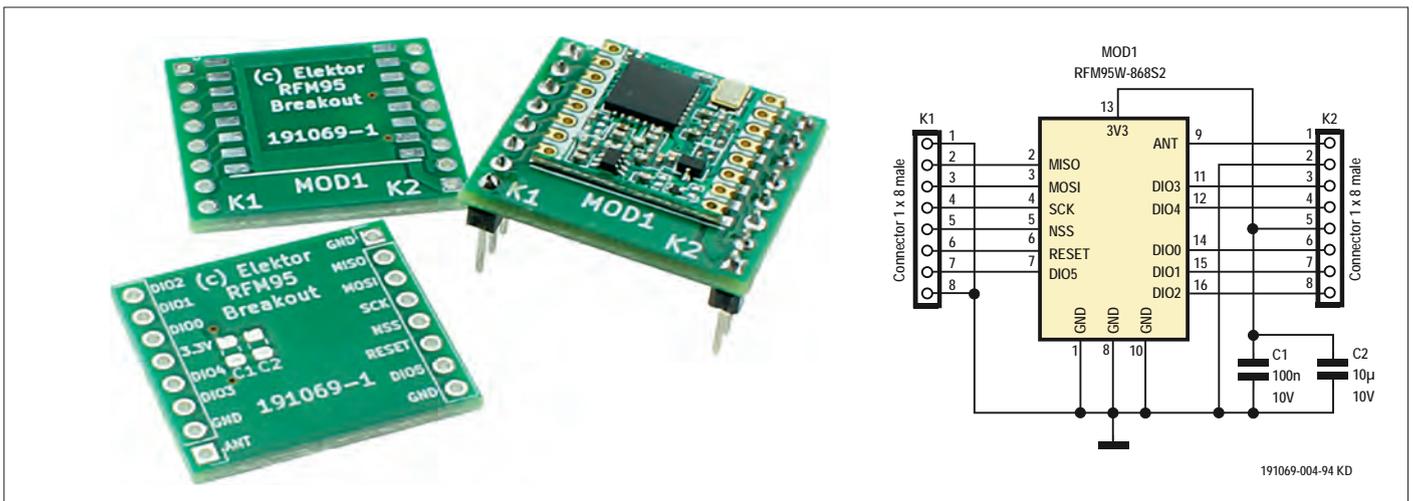


Bild 2. Breakout-Board für das LoRa-Modul.



## STÜCKLISTE

### Kondensatoren (0805):

C1 = 100 n (10%, X7R, 10 V)  
C2 = 10 µ (20%, X5R, 10 V)

### Außerdem:

K1, K2 = 8x1-Stiftleiste, RM 2,54 mm  
MOD1 = RFM95W LoRa-Transceiver-Modul, siehe Kasten „Im Elektor-Store“  
Platine 191069-1, siehe Kasten „Im Elektor-Store“

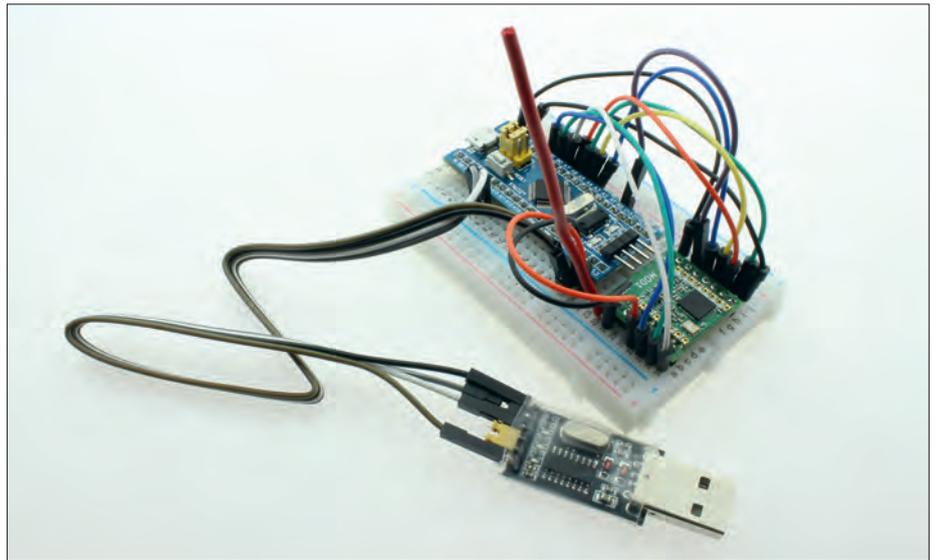
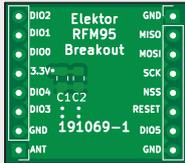
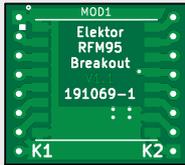


Bild 3. Die Hardware auf dem Steckbrett mit dem verdrahteten LoRa-Modul.

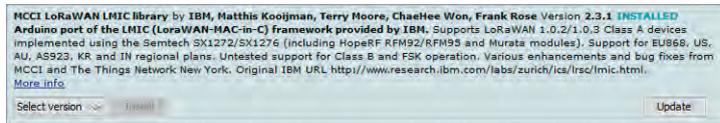


Bild 4. Installation der LMIC-Bibliothek.

gewohnt aus den Bibliotheken installieren (**Bild 4**). Im Config-File im Bibliotheksordner der LMIC-Bibliothek (unter Windows üblicherweise unter den Benutzer-Dokumenten) muss eingestellt werden, in welchem Teil der Erde man sich befindet und ob es sich um ein SX1276-LoRa-Modul handelt (was beim RFM95 der Fall ist). Dazu gibt man in der Datei `\project_config\lmic_project_config.h` für Europa folgendes an:

```
// project-specific definitions
#define CFG_eu868 1
//#define CFG_us915 1
//#define CFG_au921 1
//#define CFG_as923 1
//#define LMIC_COUNTRY_CODE LMIC_COUNTRY_CODE_JP /* for as923-JP */
//#define CFG_in866 1
#define CFG_sx1276_radio 1
//#define LMIC_USE_INTERRUPTS
```

Damit die Arduino-IDE das Blue-Pill-Board unterstützt, muss eine URL [4] unter *Datei* → *Voreinstellungen* → *Zusätz-*

*liche Boardverwalter-URLs* hinzugefügt werden. Danach können wir dann im Boardsmanager nach „STM32“ suchen und *STM32 Cores* installieren. Damit sind alle benötigten Bibliotheken eingebunden. Als letzter Schritt wird, wie in **Bild 5** gezeigt, das Board ausgewählt.

Bevor es an den ersten Code geht, muss noch kurz der Unterschied zwischen OTAA (*over the air activation*) und APB (*activation by personalization*) erwähnt werden. Bei OTAA tritt der LoRa-Node dem Netzwerk aktiv bei. Er erhält dabei vom Netzwerk eine Geräteadresse und tauscht mit dem Netzwerk Schlüssel aus. Bei The Things Network ist dies die bevorzugte Methode, um am Netzwerkverkehr teilzunehmen. Beim Modus APB dagegen werden die Geräteadresse und die Schlüssel für das Gerät fest im Code hinterlegt. Dies macht es zwar einfacher, den Node in Betrieb zu nehmen, schwächt aber die Sicherheit des Nodes. Für mehr Details zu diesem Thema lohnt sich ein Blick in die Dokumentation [5], die The Things Network bereitstellt.

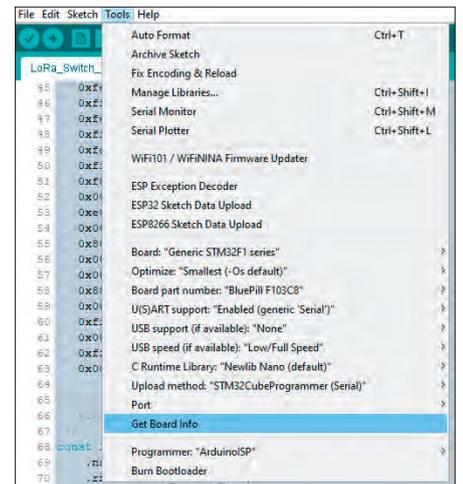


Bild 5. Die Boardinformationen werden ermittelt.

Als Ausgangspunkt wird das Beispielprogramm *tnn-abp* aus der LMIC-Bibliothek verwendet. Der Code sendet alle 60 Sekunden ein „Hello, world!“ und benötigt ein paar Anpassungen, die Hardware betreffend. Ab Zeile 86 des Arduino-Sketches wird das Pin-Mapping durch die folgenden Zeilen ersetzt:

```

/*
 * Pin-Mapping for the RFM95 LoRa Module
 */
const lmic_pinmap lmic_pins = {
  .nss = PA4,
  .rxtx = LMIC_UNUSED_PIN,
  .rst = PA0,
  .dio = {PB10, PB1, PB0},
};

```

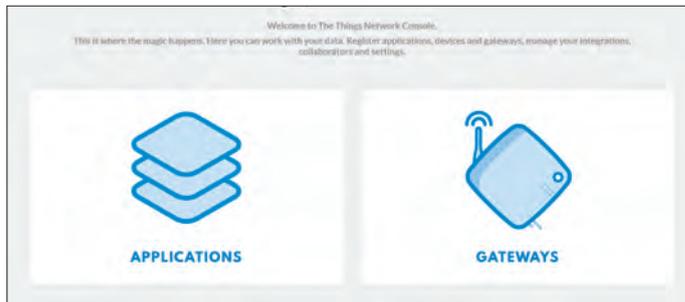


Bild 6. Willkommen auf der Konsole des The Things Network.



Bild 7. Add Application!

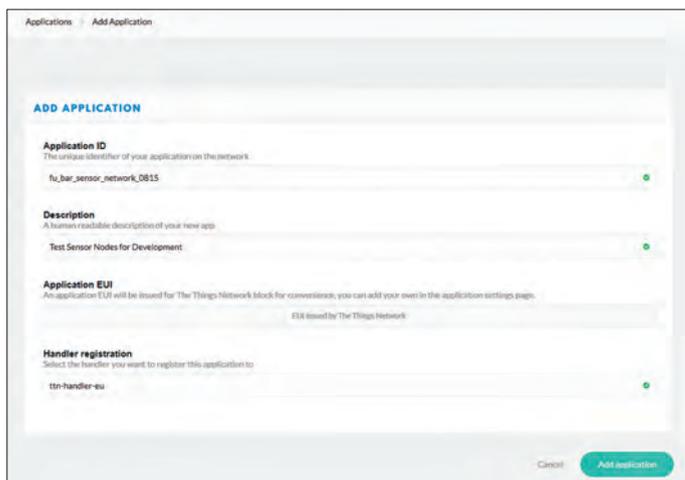


Bild 8. Anmeldung einer Applikation.

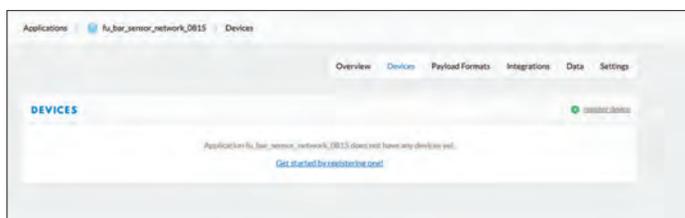


Bild 9. Aufforderung, Nodes in der Applikation zu registrieren.

## Anmeldung in The Things Network

Neben der Pins müssen auch die Netzwerkdaten für den Sensorknoten eingerichtet werden. Um diese zu erhalten, wird zunächst beim The Things Network ein neuer Node angelegt. Zunächst erstellt man unter [6] einen Benutzeraccount, in dem die Nodes angelegt und verwaltet werden. Damit ein Nutzer die Übersicht über alle seine Nodes behalten kann, lassen sie sich unter *Applications* je nach ihrem Einsatzzweck passend gruppieren. Nach dem Login können über die *Console* nun *Applications* und *Gateways* verwaltet werden, wie in **Bild 6** zu sehen.

Wenn kein nutzbares Gateway in der Nähe ist, so kann man auch sein eigenes betreiben und auch anderen Nutzern in der Nähe zur Verfügung stellen. Wie man dazu vorgeht, wird später erläutert.

Um für den Node passende Daten zu generieren, muss zuerst eine Applikation erstellt werden. Dazu klickt man das Icon

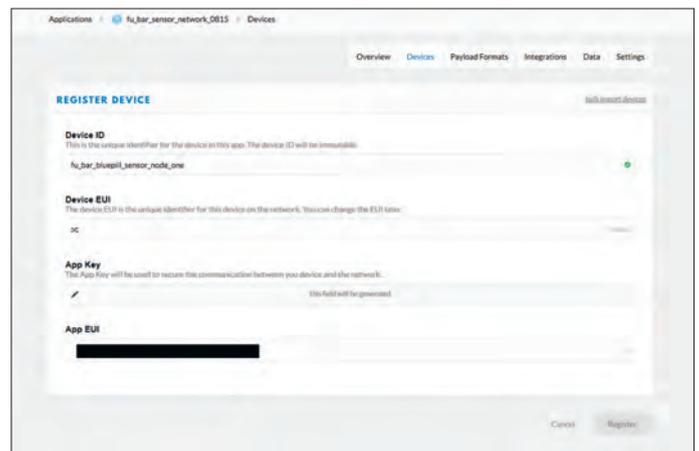


Bild 10. Hier werden die Daten des neuen Nodes angegeben.

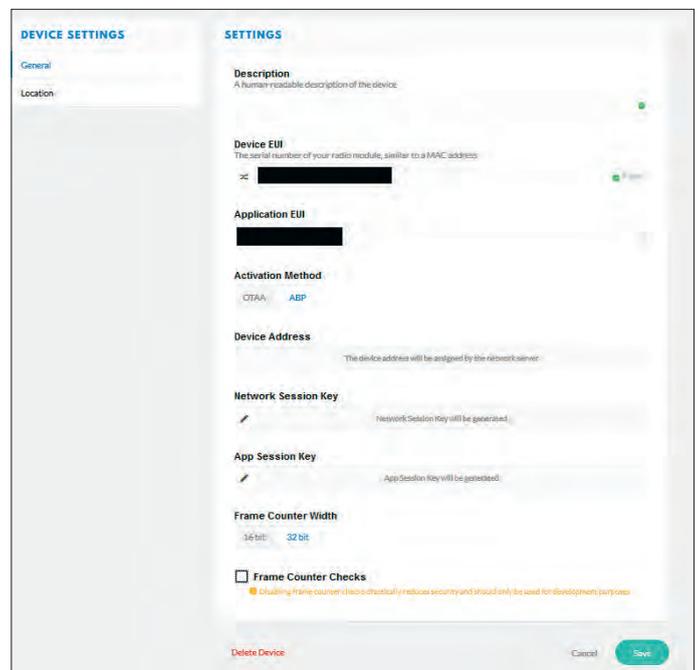


Bild 11. Aktivierungsmethode umstellen und (vorübergehend) die Frame Counter Checks abschalten.

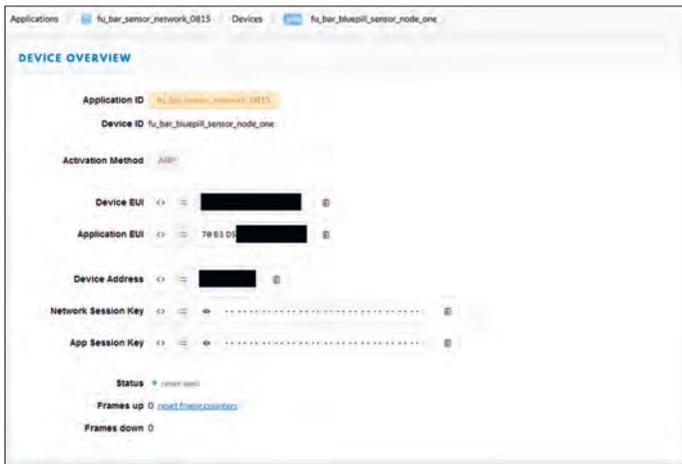


Bild 12. In der Device-Übersicht muss nur noch das Datenformat (LSB/MSB) richtig eingestellt werden.

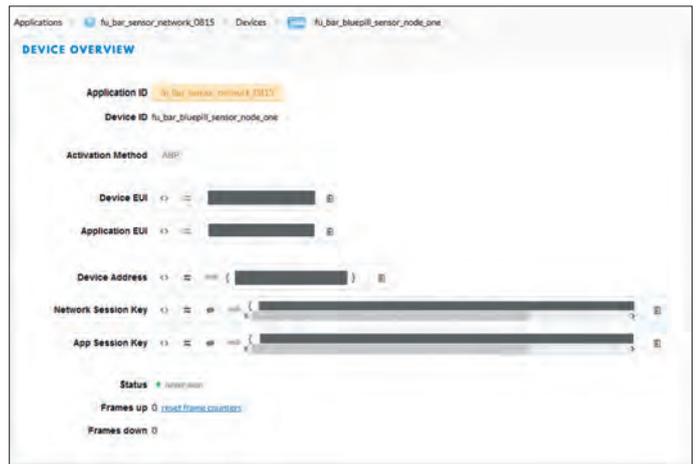


Bild 13. So ist das Device vollständig eingestellt.

*APPLICATIONS* und dann den Punkt *add application* an (**Bild 7**). Das Fenster wird wie in **Bild 8** ausgefüllt, danach wird der Vorgang mit *Add application* abgeschlossen. In dieser frisch erstellten Applikation ist noch kein Node angemeldet. Dies wird nachgeholt, indem man im sich öffnenden Fenster (**Bild 9**) den Punkt *register devices* anklickt. In dem Dialog, der sich nun öffnet (**Bild 10**), trägt man den Namen des Nodes ein und klickt unter *Device EUI* links auf das gekreuzte Pfeilsymbol, so dass der Text „*this field will be generated*“ erscheint.

Durch einen Druck auf *Register* schickt man diese Vorgaben an das Things-Network, das daraufhin den neuen Node registriert und die Zugangsdaten angibt. Unter *Settings* (**Bild 11**) wird die „*Activation Method*“ von *OTAA* auf *ABP* umgestellt und für die ersten Experimente das Häkchen bei *Frame Counter Checks* deaktiviert. Im „echten“ Produktivbetrieb ist diese Einstellung riskant; das Häkchen sollte bei der ersten besten Gelegenheit, also nach den ersten erfolgreichen Schritten, dringend wieder gesetzt werden!

Nachdem die Einstellungen gespeichert wurden, erhält man, zurück auf der *Device*-Karte, durch einen Klick auf das *Overview*-Tab alle für den Node nötigen Daten auf einen Blick (**Bild 12**). Die *Device EUI*, sie entspricht grob gesagt einer MAC-Adresse [5], interessiert uns hier nicht, genauso wenig wie die von TTN vergebene *Application EUI*, die man für das *OTAA*-Verfahren benötigt. Für uns ist hier lediglich der Inhalt der drei folgenden Felder relevant: *Device Address*, *Network Session Key* und *App Session Key*. Einen Stolperstein stellen die Datenformate dar: Man muss korrekt angeben, ob die Felder LSB oder MSB übernommen werden müssen. Für den Sketch werden *Device Address* und der *Network Session Key* als LSB benötigt und der *App Session Key* als MSB-Array. Die Einstellungen lassen sich durch einen Klick auf „< >“ umschalten; die Einstellungen werden links in den Fenstern eingeblendet (**Bild 13**).

Im Sketch wird dann in *NWKSKEY* der *Network Session Key* eingetragen, in *APPSKEY* der *App Session Key* und in *DEVADDR* die *Device Address*.

```
// LoRaWAN NwkSKey, network session key
static const PROGMEM u1_t NWKSKEY[16] = { Network
    Session Key };
```



Bild 14. Der Node sendet!

```
// LoRaWAN AppSKey, application session key
static const u1_t PROGMEM APPSKEY[16] = { App Session
    Key };

// LoRaWAN end-device address (DevAddr)
// See http://thethingsnetwork.org/wiki/AddressSpace
// The library converts the address to network byte
    order as needed.
static const u4_t DEVADDR = 0xDEVICE_ADDRESS;
// Change this address for every node!
```

Wenn ein LoRa-Gateway in der Nähe ist, kann der Sketch ohne weitere Änderungen kompiliert und hochgeladen werden. In der TTN-Konsole ist sogleich zu sehen, dass der Node das letzte Mal vor wenigen Sekunden Daten gesendet hat (**Bild 14**). Ein Klick auf *Data* zeigt die Nutzdaten des Nodes an.

### Eigenes Gateway einrichten

Wenn kein Gateway in Reichweite ist, kann man ein eigenes aufstellen. Es gibt unterschiedliche Möglichkeiten, von komplett fertigen kommerziellen Lösungen bis hin zu einem (nicht LoRaWAN-konformen) Single-Channel-Gateway, bestehend aus einem Raspberry Pi mit LoRaWAN-HAT. Ein vollständig kompatibles Gateway ist das LPS8 Indoor LoRaWAN Gateway



Bild 15. Ein Raspberry Pi mit Dragino-LoRa-HAT als Gateway.

der Firma Dragino [7], das im Elektor-Shop für etwa 120 € zu haben ist (siehe Kasten). Es kann auf allen von LoRaWAN genutzten Frequenzbändern im 868-MHz-Bereich gleichzeitig empfangen. Die Variante Dragino LG02 mit zwei LoRa-Modulen ist beim Empfang etwas eingeschränkt, dafür aber mit 80 € etwas günstiger. Die Gateways basieren auf der speziell auf Router ausgelegten Linux-Distribution OpenWRT mit ein paar Anpassungen für den Betrieb als LoRa-Gateway.

Wer einen Raspberry Pi besitzt, kann diesen mit einem LoRa-Modul ausstatten und als Gateway nutzen. Unter [8] wird die Software für ein Gateway bereitgestellt, die die Dragino-LoRa-HAT unterstützt [9]. Noch preisgünstiger ist es, das RFM95 Modul mit ein paar Jumperdrähten direkt an den RPi anzuschließen. Mit dem Breakout-Board ist das sehr einfach möglich, bei der Beschaltung und der Installation kann man sich an [8] orientieren und der Pin-Definition für die Dragino-LoRa-HAT folgen. Mit diesem RPi-Gateway kann man einen Node aber lediglich im ABP-Modus betreiben; ein Transport von Nachrichten aus dem LoRaWAN zum Node ist nicht möglich. Ebenso sind der Kanal und der Spreizfaktor (Spread Factor) fest im Node einzustellen. Wer schneller und mit weniger Stolperfallen loslegen möchte, sollte die paar Euro investieren und das LG02-Gateway von Dragino nutzen, das auch den OTAA-Modus und das Senden von Daten an Nodes aus dem LoRaWAN ermöglicht.

Aus Bequemlichkeit (und einigen schlechten Erfahrungen mit dem RFM95 und Jumper-Wires) haben wir den Elektor-Store geplündert und für den Testlauf eine Dragino-LoRa-HAT auf einem Raspberry Pi 3B+ mit frisch installiertem aktuellem Raspbian-Image verwendet (**Bild 15**). Das alles ist schnell

zusammengesetzt, die SD-Karte in den Slot und Monitor, Maus, Tastatur und Ethernetkabel für den Internetzugang eingesteckt, und schon kann der erste Boot erfolgen. Das System startet direkt in den Desktop, wo aber erst einmal das Terminal geöffnet und ein paar Hardwareeinstellungen angepasst werden müssen. Mit `sudo raspi-config` gelangt man in das Konfigurationsmenü für den RPi. Unter dem Punkt *Interface Options* wählt man *P4 SPI* aus und aktiviert diese Schnittstelle mit einem *Yes*. Nach dem Reboot ist die SPI-Schnittstelle aktiv, so dass mit der Konfiguration fortgefahren werden kann.

Wieder wird ein Terminal geöffnet, um einige Pakete und den Quelltext für den Packet-Forwarder herunterzuladen. Der Quelltext wird per GitHub bereit gestellt, so dass mit dem Befehl `git clone https://github.com/hallard/single_chan_pkt_fwd.git` die aktuelle Version heruntergeladen wird. Es müssen noch weitere Bibliotheken installiert werden. Der Befehl `sudo apt-get install wiringpi` installiert die WiringPi-Bibliothek, die wir zum Übersetzen benötigen. Mit `cd single_chan_pkt_fwd` wechseln wir in das Verzeichnis des heruntergeladenen Quelltextes, übersetzen mit `make` zuerst den Quelltext in ein Programm und installieren dieses anschließend mit `make install`.

Damit ist das Gateway grundlegend eingerichtet, es muss aber noch die Pin-Konfiguration für die LoRa-HAT in der Datei `global_conf.json` angepasst werden. Für die Dragino-HAT kann folgende Konfiguration verwendet werden:

```
{
  "SX127x_conf":
  {
    "freq": 868100000,
    "spread_factor": 7,
    "pin_nss": 6,
    "pin_dio0": 7,
    "pin_rst": 0
  },
  "gateway_conf":
  {
    "ref_latitude": 0.0,
    "ref_longitude": 0.0,
    "ref_altitude": 10,

    "name": "Enter your Gatewayname here",
    "email": "contact@whatever.com",
    "desc": "Dragino Single Channel Gateway on RPI",

    "servers":
    [
      {
        "address": "router.eu.staging.thethings.
network",
        "port": 1700,
        "enabled": true
      },
      {
        "address": "router.eu.thethings.network",
        "port": 1700,
        "enabled": false
      }
    ]
  }
}
```

Mit diesen Einstellungen lauscht das Gateway auf 868,1 MHz mit Spreizfaktor SF7. Damit die Daten an das TTN gesendet werden können, müssen noch ein paar kleinere Einstellungen vorgenommen werden. Zuerst muss das Gateway neu gestartet werden, um die gerade gemachten Änderungen zu aktivieren. Dazu gibt man im Terminal `systemctl stop single_chan_pkt_fwd` und anschließend `systemctl start single_chan_pkt_fwd` ein. Nun ist das Gateway als Service eingerichtet oder kann direkt unter dem Root-Benutzer ausgeführt werden. Mit `systemctl status single_chan_pkt_fwd` fragt man den Status des Service ab (**Bild 16**).

Das Gateway ist nun betriebsbereit. Es muss nur noch im TTN eingerichtet werden, damit auch die Pakete zugeordnet und das Gateway administriert werden können. Dazu wird in der Konsole von The Things Network das Icon **GATEWAYS** (**Bild 17**) und danach `register gateway` (**Bild 18**) angeklickt.

Auf der Karte `Register Gateway` (**Bild 19**) muss bei „I’m using the legacy Semtech packet forwarder“ das Häkchen gesetzt werden. Bei der `Gateway EUI` wird die MAC-Adresse des Raspberry Pi (sie wird mit `cat /sys/class/net/eth0/address` ermittelt) eingetragen und diese in der Mitte mit `FF FF` aufgefüllt. Wenn Sie zum Beispiel als MAC-Adresse `b8:27:eb:12:34:56` ermitteln, tragen Sie in der Konsole des The Things Network in das Feld `Gateway EUI` „`B8 27 EB FF FF 12 34 56`“ ein. Durch einen Klick auf `Register Gateway` ist dieses nun einsatzbereit. Damit LoRa-Node und Gateway zusammenarbeiten, muss im Quelltext eine Übertragungsfrequenz definiert und der Spreizfaktor festgelegt werden. Zusätzlich wird der Node auf die im Gateway definierte Frequenz beschränkt. Der Auszug aus dem Quelltext zeigt die Anpassungen für den LoRa-Node. Ein komplettes Beispiel kann von der Elektor-Labs-Seite [10] heruntergeladen werden.

```
...
// Disable link check validation
LMIC_setLinkCheckMode(0);

// TTN uses SF9 for its RX2 window.
LMIC.dn2Dr = DR_SF9;

// Set data rate and transmit power for uplink (note:
// txpow seems to be ignored by the library)
LMIC_setDrTxpow(DR_SF7,14);

//If we have a single channel one module gateway we
// need to add these lines
// Define the single channel and data rate (SF) to
// use
int channel = 0;
int dr = DR_SF7;

// Disable all channels, except for the one defined
// above.
// FOR TESTING ONLY!
for(int i=0; i<9; i++) { // For EU; for US use i<71
    if(i != channel) {
        LMIC_disableChannel(i);
    }
}
}
```



Bild 16. Screenshot des Terminals: Service okay!



Bild 17. Im TTN geht es hier zur Gateway-Einstellung.

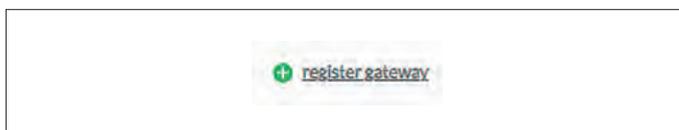


Bild 18. Zur Gateway-Registrierung im TTN.

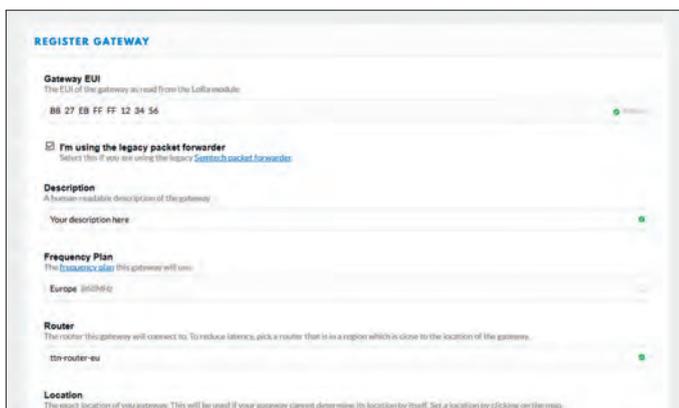


Bild 19. Hier werden die Gateway-Daten eingetragen.

```
// Set data rate (SF) and transmit power for uplink
LMIC_setDrTxpow(dr, 14);

// Start job
do_send(&sendjob);
...
```

In diesem Beispiel wird der Node auf 868,1 MHz und SF 7 festgelegt. Diese Werte nutzen wir auch in unserem Gateway.

Gateway und Node sind jetzt mit Software versorgt, so dass der Test starten kann. Nach kurzer Zeit sollten in der TTN-Konsole neue Nachrichten von unserem Knoten einlaufen (**Bild 20**).

### Fazit und Ausblick

Für die ersten Schritte mit LoRaWAN reichen also ein Node für etwa 25 € und ein Gateway auf Basis eines Raspberry Pi und eines RFFM95-Moduls aus. Diese Lösung ist allerdings stark eingeschränkt und nicht für den Produktivbetrieb geeignet. Wer nach dem Experimentieren darauf brennt, mehr mit LoRaWAN anzustellen, sollte das Gateway gegen ein vollständig LoRaWAN-taugliches, kommerzielles Gerät tauschen, das mittlerweile für weniger als 120 € [7] zu haben ist. Mit einem solchen Gateway lässt sich das volle Potential des LoRaWAN nutzen, inklusive Nachrichten vom Gateway zum Node.

Ein Punkt, den wir noch nicht betrachtet haben, ist die Abholung und Weiterverarbeitung der Daten aus dem The Things Network. Dies soll Stoff für einen weiteren Artikel in einer der nächsten Ausgaben sein. ◀

191065-01

Filters	uplink	downlink	activation	ack	error
	Time	counter	port		payload
	14:34:09	2	1		payload: 48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21
	14:35:56	0	1	very	payload: 48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21
	15:42:58	0	1	very	payload: 48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21

Bild 20. Die Daten des Nodes treffen ein!

### IM ELEKTOR-STORE

- **LoRa-Modul RFM95**  
[www.elektor.de/18715](http://www.elektor.de/18715)
- **LoRa-RFM95-Breakout-Board**  
[www.elektor.de/191069-1](http://www.elektor.de/191069-1)
- **USB/Seriell-Wandler CH340**  
[www.elektor.de/19151](http://www.elektor.de/19151)
- **Dragino LPS8**  
[www.elektor.de/dragino-lps8-indoor-lorawan-gateway](http://www.elektor.de/dragino-lps8-indoor-lorawan-gateway)
- **Dragino LG02**  
[www.elektor.de/dragino-lg02-dual-channels-lora-iot-gateway](http://www.elektor.de/dragino-lg02-dual-channels-lora-iot-gateway)
- **Dragino-HAT für RPi**  
[www.elektor.de/dragino-lora-gps-hat-for-raspberry-pi-868-mhz](http://www.elektor.de/dragino-lora-gps-hat-for-raspberry-pi-868-mhz)

### Weblinks

- [1] Blue Pill Controllerboard: [www.amazon.de/s?k=bluepill&\\_\\_mk\\_de\\_DE=%C3%85M%C3%85%C5%BD%C3%95%C3%91&ref=nb\\_sb\\_noss\\_1](http://www.amazon.de/s?k=bluepill&__mk_de_DE=%C3%85M%C3%85%C5%BD%C3%95%C3%91&ref=nb_sb_noss_1)
- [2] LoRa-Modul: [www.elektor.de/rfm95-ultra-lora-transceiver-module-868-915-mhz](http://www.elektor.de/rfm95-ultra-lora-transceiver-module-868-915-mhz)
- [3] USB-Seriell-Wandler CH340 (3,3 V): [www.elektor.de/19151](http://www.elektor.de/19151)
- [4] Boardunterstützung: [https://github.com/stm32duino/BoardManagerFiles/raw/master/STM32/package\\_stm\\_index.json](https://github.com/stm32duino/BoardManagerFiles/raw/master/STM32/package_stm_index.json)
- [5] LoRaWAN-Adressierung und -Aktivierung: [www.thethingsnetwork.org/docs/lorawan/addressing.html](http://www.thethingsnetwork.org/docs/lorawan/addressing.html)
- [6] TTN-account anlegen: <https://account.thethingsnetwork.org/register>
- [7] Gateway LPS8: [www.dragino.com/products/lora-lorawan-gateway/item/148-lps8.html](http://www.dragino.com/products/lora-lorawan-gateway/item/148-lps8.html)
- [8] Software Gateway-HAT: [https://github.com/hallard/single\\_chan\\_pkt\\_fwd](https://github.com/hallard/single_chan_pkt_fwd)
- [9] Gateway-HAT: [www.elektor.de/dragino-lora-gps-hat-for-raspberry-pi-868-mhz](http://www.elektor.de/dragino-lora-gps-hat-for-raspberry-pi-868-mhz)
- [10] Projektseite: [www.elektormagazine.de/191065-01](http://www.elektormagazine.de/191065-01)



Vom **Elektor-Team**

Ist Ihr elektronisches Start-up bereit für einen Wechsel? Möchten Sie Ihr junges Unternehmen bei potenziellen Kunden und Investoren bekannt machen? **electronica Fast Forward (e-ffwd)**, die Startup-Plattform powered by Elektor, bietet Ihnen eine einzigartige Möglichkeit, Ihr Unternehmen und Ihre Technologie der Welt zu präsentieren. Sie können ein Marketingpaket von Elektor im Wert von 75.000 € gewinnen und sich auf der **electronica 2020** internationale Aufmerksamkeit verschaffen! Registrieren Sie sich für den Wettbewerb **e-ffwd 2020** unter [www.elektormagazine.de/fastforward](http://www.elektormagazine.de/fastforward).

Gründung und erfolgreicher Betrieb eines auf Elektronik fokussierten Unternehmens sind keine leichte Aufgabe. Kluge Führungskräfte von Start-ups wissen, dass sie mehr brauchen als ein innovatives Produkt, eine neue Technologie oder super Software. Sie pflegen Beziehungen zu wichtigen Branchen Größen und Experten. Sie wissen um den Wert einer guten Präsentation von Eigenschaften, Funktionen und Vorteilen ihrer Produkte vor den relevanten Zielgruppen. Und sie wissen, dass zielgerichtete Marketingkampagnen wichtig sind, die verwertbare Leads liefern und den Umsatz ankurbeln.

Auch im Jahr 2020 arbeiten Elektor und die Messe München wieder zusammen, um Innovatoren die Möglichkeit zu geben, Start-ups voranzubringen und ihre Produkte einem internationalen Publikum vorzustellen. **electronica Fast Forward, die Startup-Plattform powered by Elektor**, ist Ihre Chance, eine globale Marketingkampagne von Elektor zu gewinnen und auf der **electronica 2020** – der Weltleitmesse für Elektronik – vom 10. bis 13. November 2020 in München international präsent zu sein. Die vielversprechendsten Start-ups haben dort die Chance, ihre Lösungen bei potenziellen Kunden, Branchenexperten und möglichen Investoren bekannt zu machen. Sind Sie bereit für den Wettbewerb?

- **Anmeldeschluss:** 1. Juli 2020 @ [www.elektormagazine.de/fastforward](http://www.elektormagazine.de/fastforward)
- **Das Finale:** auf der electronica 2020 in München vom 10. bis 13. November 2020
- **Preise:**
  - Erster Preis = Marketingpaket im Wert von 75.000 € von Elektor plus einen Stand auf der electronica 2022.
  - Zweiter Preis = Marketingpaket im Wert von 50.000 € von Elektor.
  - Dritter Preis = Marketingpaket im Wert von 25.000 € von Elektor.

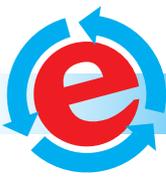


- Buchen Sie einen exklusiven, mit nur 1.905 € besonders preisgünstigen Stand von 5 m<sup>2</sup> auf der electronica 2020 (inkl. Grundmöblierung, Beleuchtung, Stromversorgung, täglicher Reinigung, Entsorgungsservice, etc.)
- Erhalten Sie einen Eintrag im Messekatalog (Print und Online).
- Präsentieren Sie Ihre innovativen Produkte und vernetzen Sie sich mit potentiellen Investoren, Partnern und Kunden vor Ort.
- Generieren Sie neue Leads und erhalten Sie große internationale Aufmerksamkeit vor, während und nach der electronica 2020.
- Erfahren Sie mehr über die Stärken und Schwächen Ihres Unternehmens.
- Und vieles andere mehr!



Verfügt Ihr Start-up über eine überlegene KI-Lösung, eine hochpotente IoT-Anwendung oder ein hochmodernes Robotik-Design? Vielleicht hat Ihr Team kürzlich ein neues Embedded-System oder ein nützliches Messinstrument entwickelt? Haben Sie eine besonders günstige Rapid-Prototyping-Plattform oder ein schickes elektronisches Gadget erfunden? Registrieren Sie sich für electronica Fast Forward, die Startup-Plattform von Elektor unter [www.elektormagazine.de/fastforward](http://www.elektormagazine.de/fastforward), und seien Sie bereit, vor der Welt zu stehen!

**Viel Erfolg!**



# Ein starker Verfechter des Offenen Internets

## Interview mit Wienke Giezeman, Gründer von The Things Network

Fragen von **Mathias Claußen** und **Jens Nickel**

Das IoT kann ohne energieeffiziente drahtlose Übertragung von Sensordaten nicht existieren. LoRaWAN ist ein Protokoll, das eine flexible Kommunikation zwischen Sensor-/Aktor-Knoten und Gateways ermöglicht, die als Basisstationen und Schnittstellen zum Internet fungieren. Allerdings müssen potentielle Nutzer eine eigene Netzwerkinfrastruktur aufbauen, was für Bildungseinrichtungen und private Nutzer eine Belastung darstellen kann. Bestehende Netze sind oft geschlossen oder anderweitig nicht zugänglich. Das *Things Network* ist eine Möglichkeit, diese Hindernisse zu überwinden. Überall auf der Welt installieren Menschen Gateways, die für andere Nutzer, die Daten übertragen möchten, offen sind, während kommerzielle Teilnehmer ihre Gateways ebenfalls für das Netzwerk öffnen können. The Things Network hostet auch LoRaWAN-Netzwerkserver und bietet kostengünstige Hardware und eine Menge Support.

### **Elektor: Wienke, wie kommt es, dass du TTN gegründet hast?**

**Wienke:** Johan Stokking und ich haben The Things Network (TTN) im Jahr 2015 gegründet, nachdem wir beide auf der Suche nach einem neuen Unternehmen waren. Johan ist seit seinem zwölften Lebensjahr IT-Unternehmer. Im Jahr 2015 hatte ich gerade meine bisherige Firma verkauft. Wir kamen bei einem Kaffee zusammen und dort ist dann die Idee entstanden. Ich war immer ein starker Befürworter von Open Source und Open Internet, sowohl persönlich als auch als Unternehmer. Ich habe die Kampagne *Startups for Net Neutrality* [2] gestartet, als die Netz-Neutralität in Europa bedroht war.

### **Elektor: Was sind deiner Meinung nach die Vorteile von LoRa?**

**Wienke:** Die Vorteile von LoRa als reine HF-Technologie sind einfach darzustellen. Die Geräte sind stromsparend, so dass man die Batterie nur alle zwei oder drei Jahre austauschen muss. Und die Reichweite kann kilometerweit sein. Das hängt natürlich von den Bedingungen ab.

Der andere Vorteil ist, dass es ein gut geregeltes Netzwerkprotokoll namens LoRaWAN gibt. Dieses Netzwerkprotokoll ermöglicht es Geräten, sich mit Netzwerken zu verbinden und ebenso LoRaWAN-Netzwerken, sich miteinander zu verbinden. Damit wird die Technologie von einem reinen Last-Mile-Tool zu einer Internet-Technologie. Da jeder ein LoRaWAN-Netzwerk aufbauen kann, wächst die Technologie mit den Netzwerken, die Unternehmen und Menschen aufbauen. LoRaWAN ermöglicht

es, diese Netzwerke einfach zu verbinden und so ein globales Internet der Dinge zu schaffen.

Bei *The Things Network* haben wir uns diese Vision von Anfang an zu eigen gemacht. Wir haben in Amsterdam begonnen, aber bald umfasste TTN Hunderte von Städten auf der ganzen Welt. Inzwischen betreiben wir mehr als 10.000 Gateways in über 700 Städten weltweit und fast 100.000 Entwickler sind auf der Plattform aktiv.

### **Elektor: Kannst du dir Anwendungen vorstellen, bei denen man LoRa definitiv nicht einsetzen sollte? Und warum nicht?**

**Wienke:** Viele. Die verschiedenen Technologien des Internet of Things besetzen immer ihre eigene Nischen oder ihre eigenen Segmente. Obwohl einige Marketing-Kampagnen glauben machen wollen, ihre Technologie könne jeden Anwendungsfall bedienen, ist das sicher nicht der Fall. Es sollte eine Übereinstimmung bestehen zwischen den geschäftlichen Anforderungen und dem, was die Technologie zu bieten hat.

Da die Stärke von LoRaWAN in der geringen Leistung und der großen Reichweite liegt, ist der Kompromiss, dass man nur sehr wenige Daten senden kann. Für die Stärke, dass es nur sehr preiswerte Gateways benötigt, muss man den Kompromiss eingehen, dass man einen Netzwerkserver in der Cloud braucht, der das Netzwerkprotokoll handhabt. Du kannst zwar ein Netzwerk mit wenig Herstellerbindung selbst aufbauen, bezahlst aber mit einer finanziellen Investition dafür.



Die TTN-Gründer Wienke Giezeman (links) und Johan Stokking (Mitte links) mit Mitarbeitern (Foto: Rebekka Mell / The Things Network).

**Elektor: Ist es nicht ein wenig gefährlich, wenn alles von einem einzigen Chiphersteller abhängig ist?**

**Wienke:** Das LoRa-HF-Protokoll ist Eigentum der Firma Semtech. Wenn man sich einen IoT-Stack oder Software-Stack anschaut, gibt es immer Abhängigkeiten und Risiken der Herstellerbindung. Unser Fall ist da ist keine Ausnahme: Wer eine IoT-Lösung erstellt, sollte dies in die Risikobewertung einbeziehen. Wenn man einen typischen LoRaWAN-Stack mit beispielsweise einem Sigfox-, NB-IoT- oder LTE-M-Stack vergleicht, wird man feststellen, dass die Erstellung eines LoRaWAN-Stacks das geringste Vendor-Lock-in-Risiko birgt. Bei unserer Software versuchen wir, das Risiko der Herstellerabhängigkeit zu reduzieren, indem wir offene APIs, Open-Source-Kernkomponenten und so viel wie möglich vom offenen LoRaWAN-Standard verwenden.

**Elektor: TTN baut auf einer Gemeinschaft von Freiwilligen auf, die Gateways installieren. Wie kann man mitmachen?**

**Wienke:** *The Things Network* wurde von Menschen und Unternehmen aufgebaut, die LoRaWAN-Anwendungen installieren. Das Gateway ist nur Mittel zum Zweck. Das Protokoll wurde so entworfen, dass die Netzwerke sich die Kapazität mit dem Ziel teilen, das knappe offene Spektrum optimal zu nutzen. LoRaWAN ermöglicht eine sehr effiziente Nutzung des Spektrums, da die Netzwerke gemeinsam genutzt werden und der LoRaWAN-Verkehr so schnell wie möglich in reichlich vorhandene IP-Netzwerke ausgelagert wird.

Die gemeinsame Nutzung eines Gateways oder eines Netzwerks hat also nicht viel mit der Freiwilligkeit zu tun. Es hat damit zu tun, die eigene Netzwerkkapazität und den Zugang durch den Beitritt zum Netzwerk zu optimieren.

Der Beitritt zum TTN ist sehr preiswert und einfach. Ein Gateway und ein Developer-Board kann man schon für unter 100 € bekommen und unter [www.thethingsnetwork.org](http://www.thethingsnetwork.org) vernetzen.

**Elektor: Wie erhalten Teilnehmer Unterstützung von eurer Seite?**

**Wienke:** Wir bieten Unterstützung durch unser Unternehmen *The Things Industries*. Derzeit helfen wir Hunderten von Unternehmen weltweit, von nationalen Eisenbahnorganisationen, globalen maritimen Unternehmen bis hin zu großen Immobiliengesellschaften beim Aufbau ihrer LoRaWAN-Netzwerke.



Auf der *The Things Conference* berichten Anwender über TTN-Projekte, hier die Deutsche Bahn (Foto: Rebekka Mell / The Things Network).

Dies geschieht durch ein bezahltes Angebot, bei dem Unternehmen Unterstützung und Tools zur Verwaltung ihres Netzwerks erhalten. Wir sind ziemlich stolz darauf, dass wir in der Lage sind, ein Geschäftsmodell auf einer offenen Plattform zu realisieren, ohne dabei den kommerziellen Erfolg und unseren Glauben an offene Technologien zu gefährden.

**Elektor: Wie verdient ihr euer Geld? Steht ein Geschäftsmodell hinter TTN?**

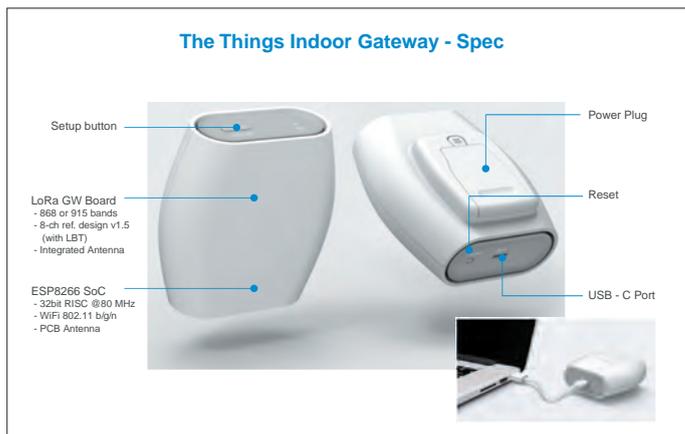
**Wienke:** Wir verdienen unser Geld, indem wir Dienstleistungen und Produkte über unser Unternehmen *The Things Industries* verkaufen. TTN kostet zurzeit nur Geld. Wir werden nach Möglichkeiten suchen, wie wir die Overhead-Kosten im Laufe der Zeit gerechter finanzieren können, aber der Saldo stimmt im Moment. Wir sind in der Lage, ihn durch unsere kommerziellen Aktivitäten zu finanzieren.

**Elektor: Können die Nutzer eigene Netzwerkserver betreiben und auch dem Netzwerk beitreten?**

**Wienke:** Ja, für Unternehmen haben wir unseren neuen LoRaWAN-Netzwerkserver namens *The Things Enterprise Stack*. Damit kann man einen professionellen LoRaWAN-Netzwerkserver einrichten, mit voller Kontrolle über die eigene Servicequalität und das Sicherheitsniveau. In der Zwischenzeit ist ein Netzwerk-Peering mit den anderen offenen Netzwerken möglich, um die gegenseitige Netzwerk- und Spektrumeffizienz zu erhöhen. Wir haben auch eine Open-Source-Version, mit der man das Netzwerk vollständig selbst einrichten kann.

**Elektor: Wie viele Gateways hat eine Stadt wie Amsterdam nötig?**

**Wienke:** Diese Frage ist schwer zu beantworten. Es hängt von der Bebauungsdichte und weiteren Faktoren ab. Das Gute ist, dass die Gateways sehr günstig sind, 500 € bis 700 € für eine Outdoor-Version und 60 € bis 100 € für eine Indoor-Version. Das ermöglicht einen schlanken Ansatz, bei dem man die Konnektivität nach Bedarf hinzufügt. Um zumindest eine grobe Antwort zu geben: Sagen wir, dass man für eine Stadt wie Amsterdam 100 Outdoor-Gateways benötigt und 300 Indoor-Gateways hinzufügen müsste, um die Lücken zu füllen. Die Gesamtkosten für die Gateways würden sich dann auf etwa 50.000 € belaufen,



Ein einfach zu bedienendes Indoor-Gateway.

eine sehr geringe Summe für diese Größenordnung. Wenn man bedenkt, wie viele Daten heute schon weltweit über Gateways und verbundene Netzwerke laufen, verteilen sich diese Kosten auf viele Anwendungen.

**Elektor:** Gibt es Gateways, deren Einsatz mit TTN du definitiv nicht empfehlen kannst?

**Wienke:** Alle Gateways, die die Semtech-Gateway-Chips und deren Referenzdesign verwenden, sind kompatibel. Im Moment sind mir keine besonders schlechten Gateways bekannt. Meistens hängen die Kosten von der Größe des Netzwerks ab. Ein einfaches Indoor-Gateway ist deshalb billiger, weil es ein kleineres Netzwerk schafft.

**Elektor:** Plant ihr auch die Unterstützung von Class-B-Geräten?

**Wienke:** Ja, das sieht unsere neueste Software vor.

**Elektor:** Gibt es auch ein Nachrichten-Roaming von und zu anderen Netzwerkservern?

**Wienke:** Ja, dafür haben wir das Konzept des Packet Brokers ([www.packetbroker.org](http://www.packetbroker.org)) entwickelt. Wie ich schon sagte, das Schöne an LoRaWAN ist, dass man ein echtes Internet der Dinge aus miteinander verbundenen Gateways und Netzwerken aufbauen kann. Mit dem Packet Broker versuchen wir, den Datenaustausch zwischen Netzwerken zu vereinfachen.

**Elektor:** Gibt es Partnerschaften mit Unternehmen?

**Wienke:** Schaut auf [www.thethingsnetwork.org](http://www.thethingsnetwork.org) und klickt auf *Market Place!* Dort stehen alle unsere Partner. Die Stärke des LoRaWAN-Ökosystems ist, dass es so viele Partner gibt.

**Elektor:** Kannst du etwas über die Partnerschaft mit der niederländischen KPN sagen?

**Wienke:** Die KPN ist ein etablierter Telekommunikationsanbieter (die ehemals staatliche Post, die Red.), der LoRaWAN-Gateways in den Niederlanden einrichtet und für den Anschluss etwa 30...40 € pro Jahr und Gerät verlangt.

**Elektor:** Zurzeit wird in Europa das 868-MHz-Band genutzt. Gibt es Pläne, auch das 433-MHz-Band einzubeziehen?

**Wienke:** Im Moment nicht.

**Elektor:** Wir haben gesehen, dass es seit neuestem Semtech-Chips für das 2,4-GHz-Band gibt. Wie könnte das in die bestehende Infrastruktur integriert werden?

**Wienke:** Erst einmal: Kein Kommentar.

**Elektor:** Kannst du etwas über LoRa im Weltraum sagen?

**Wienke:** Das ist super spannend! Wir sind große Fans von *Lacuna Space*, weil sie technisch so großartig fortschrittlich sind, dass sie LoRaWAN-Nachrichten von der Erde direkt zu einem Satelliten senden. Behaltet dieses Unternehmen im Auge, bald werden *The Things Network* und die Nutzer von *The Things Industries* dieses globale Netzwerk nutzen können!

**Elektor:** Was ist euer Beitrag zur LoRaWAN-Allianz?

**Wienke:** Wir sind ein sehr aktives Mitglied, das sowohl im technischen als auch im Marketingbereich tätig ist. Wir haben die größten globalen LoRaWAN-Konferenzen in Amsterdam, Haiderabad, Adelaide, Slowenien und in Großbritannien organisiert. Außerdem tragen wir zum technischen Komitee der LoRa-Allianz bei, indem wir beispielsweise die erste Lösung für ein Firmware-over-the-Air-Update entwickeln.

**Elektor:** Was sind die Pläne für die weitere Netzabdeckung? Was macht ihr, um sie zu verbessern?

**Wienke:** Wir starten unseren neuesten Netzwerkservers und werden ihn in [thethingsnetwork.org](http://thethingsnetwork.org) einbringen. Wir sehen ein solides jährliches Wachstum von 100 % und ein noch stärkeres Wachstum vernetzter Geräte.

**Elektor:** Wie siehst du die Zukunft von LoRaWAN?

**Wienke:** Die Zukunft von LoRaWAN hängt davon ab, wie sich der WAN-Teil entwickelt. Wie gesagt, es gibt viele Last-Mile-IoT-Lösungen, die Ihre Daten von A nach B bringen können. Es gibt aber nur eine, die es Unternehmen durch einen offenen Standard ermöglicht, Geräte, Netzwerke und Anwendungen aufzubauen, die alle interoperabel sind.

In kommerzieller Hinsicht bedeutet die Tatsache, dass LoRaWAN Problemstellungen in allen Branchen, in der Landwirtschaft, im Immobiliensektor, in der Messtechnik, in intelligenten Städten, in der Logistik, auf See und vielem mehr lösen kann, dass LoRaWAN eine dauerhafte Lösung ist. Die Stärke des Ökosystems wird bestimmen, wie schnell sich LoRaWAN verbreitet. Genau aus diesem Grund ist das Thema unserer kommenden LoRaWAN-Konferenz ([www.thethingsconference.com](http://www.thethingsconference.com)) die Interoperabilität. Wir zeigen, wie Partner, ja sogar Konkurrenten zusammenarbeiten können, um ein globales, interoperables LoRaWAN-Netzwerk voranzutreiben. ◀

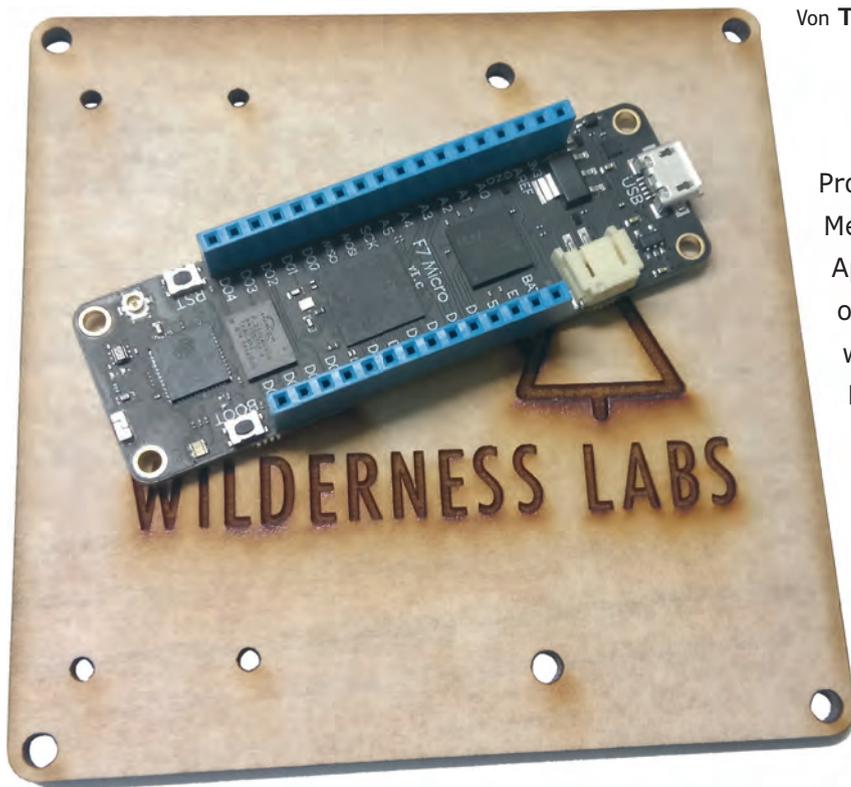
191193-02

#### Weblinks

- [1] Homepage The Things Network: [www.thethingsnetwork.org/](http://www.thethingsnetwork.org/)
- [2] Startups für Netzneutralität: [www.startupsfor.netneutrality.eu](http://www.startupsfor.netneutrality.eu)

# Meadow F7

## Ein Board für .NET-Entwickler



Von **Tam Hanna**

Prozessrechner gibt es wie Sand am Meer. Möchte man die Software seiner Applikationen allerdings mit Visual Basic oder C# entwickeln, so zerrinnt die Vielfalt wie Sand zwischen den Fingern. Beim Meadow F7 ist ein STM32F777 für die Ausführung der .NET-Runtime zuständig, während ein ESP32-Modul Kontakt zu WLAN-Netzwerken aufnimmt.

(Bildquelle: Wilderness Labs)

Bryan Costanich aus dem Umfeld von Xamarin, das für die Portierung der .NET-Umgebung auf Android und iOS verantwortlich ist, kaufte vor einiger Zeit das intellektuelle Eigentum des Unternehmens von Chris Walker. Das neue Produkt seiner neuen Firma Wilderness Labs ist der Prozessrechner Meadow, der im Titelbild im zusammengelöteten Zustand zu sehen ist. Meadow soll .NET-Entwicklern „erstklassige“ Zugangsmöglichkeiten zum IoT-Ökosystem verschaffen.

Architektonisch handelt es sich hierbei übrigens um ein Analogon dessen, was man bisher durch Kombination von Raspberry Pi oder Orange Pi und Echtzeit-Prozessorkern erreichte. Ein mit 216 MHz getakteter STM32F777 ist für die Ausführung der .NET-Runtime zuständig, während ein ESP32-Modul Kontakt zu WLAN-Netzwerken aufnimmt.

### Los geht's

Das Betriebssystem auf den Meadow-F7-Boards bei im Auslieferungszustand entweder veraltet oder im Interesse einer kostengünstigen Produktion erst gar nicht installiert. Das Deployment erfolgt über den SGS-Bootloader. Besuchen Sie [1] und laden Sie das aus zwei Dateien bestehende Betriebssystem herunter. Als Nächstes brauchen Sie die DFU-utils, die unter [2] warten und an Sie als erfahrenen Elektor-Leser sicher keine großen Herausforderungen stellen dürften.

Ein Blick auf die Schaltungsunterlagen zeigt, dass der Start des Bootloaders durch Drücken der Boot-Taste erfolgt. Halten Sie

den Taster gedrückt, während Sie die Platine per Micro-USB-Kabel mit ihrer Windows-Workstation verbinden. Zur Ermittlung der Seriennummer des Prozessrechners verwenden wir den List-Befehl:

```
C:\dfu-util-0.9-win64>dfu-util --list
dfu-util 0.9
```

```
...
Cannot open DFU device 0483:df11
```

Oops - was unter älteren Windows-Versionen problemlos funktioniert, scheitert unter Windows 10 wegen des unter [3] beschriebenen Fehlers. Er lässt sich aber ohne Schwierigkeiten beheben: Folgen Sie einfach den Anweisungen, um den Treiber zurückzusetzen. Nach getaner Arbeit lässt sich die Seriennummer nach folgendem Schema ermitteln:

```
C:\dfu-util-0.9-win64>dfu-util --list
dfu-util 0.9
...
Found DFU: [0483:df11] ver=2200, devnum=4,
cfg=1, intf=0, path="5-3", alt=3, name="@
Device Feature/0xFFFF0000/01*004 e",
serial="346B38733536"
```

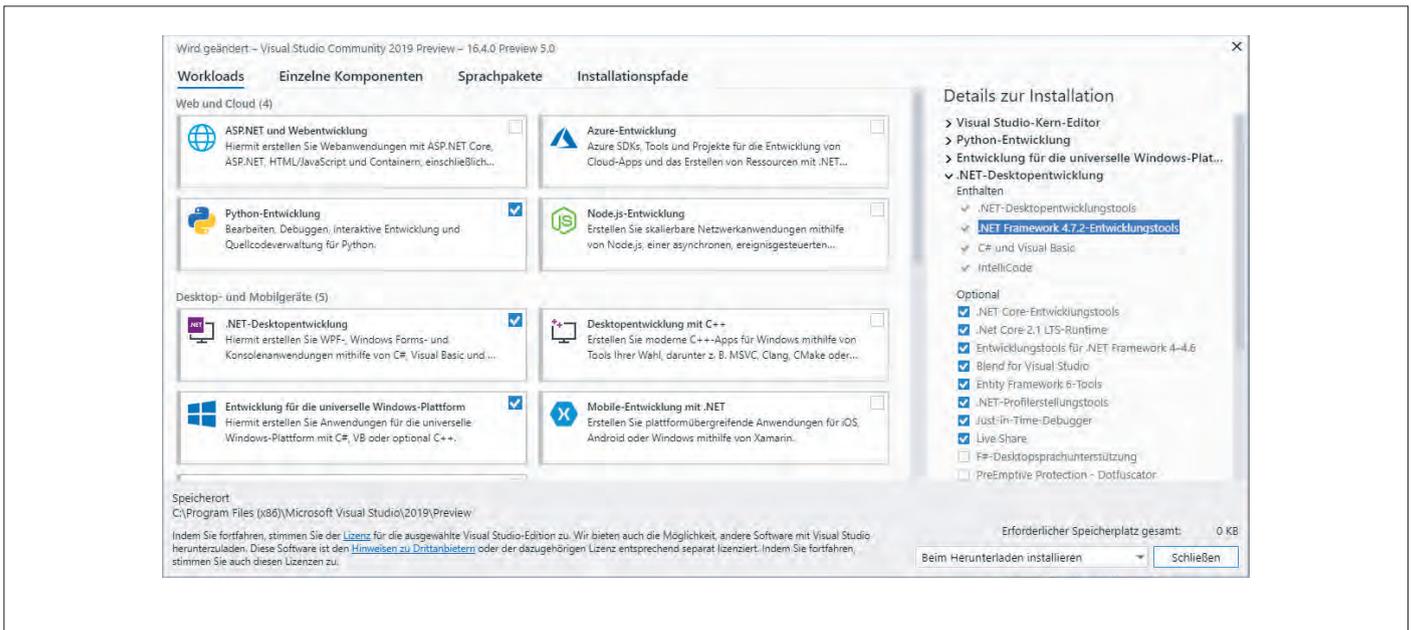


Bild 1. Ohne .NET Framework 4.7.2-Entwicklungstools macht der Meadow keinen Spaß.

Erfolgreich erkannte Meadow-Boards erscheinen nicht nur mit einer, sondern mit vier Endstellen in der Windows-Geräteliste. Für uns ist allerdings nur die Seriennummer wichtig, die wir im nächsten Schritt zum Aufspielen von Kernel und Runtime verwenden. Die hier gezeigten Befehle müssen Sie natürlich an Ihre Situation anpassen! Achten Sie darauf, die Hexadezimaladressen korrekt abzutippen und die beiden Dateien `Meadow.OS_Kernel.bin` und `Meadow.OS_Runtime.bin` im richtigen Ordner abzulegen:

```
C:\dfu-util-0.9-win64> dfu-util -a 0 -S
346B38733536 -D Meadow.OS_Kernel.bin -s 0x08000000
dfu-util 0.9
C:\dfu-util-0.9-win64> dfu-util -a 0 -S
346B38733536 -D Meadow.OS_Runtime.bin -s
0x08040000
dfu-util 0.9
```

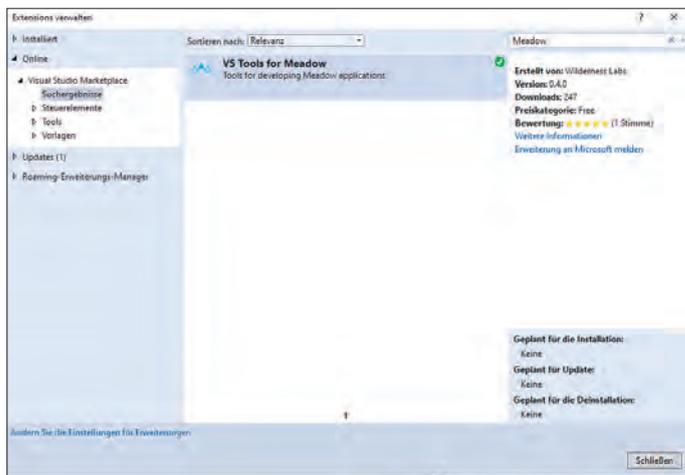


Bild 2. Visual Studio beschafft Erweiterungen auf Wunsch automatisch aus dem Internet.

Um sich des korrekten Starts zu versichern, drücken Sie die RST-Taste: Die RGB-LED beginnt daraufhin, ein wenig hektisch zu blinken. Als Entwicklungsumgebung auf dem Desktop habe ich Visual Studio in der kostenlosen Community-Version 2019.8 verwendet. Für die Installation der benötigten Komponenten werden im Installationsassistenten, der im Startmenü unter *Visual Studio Installer* bereitsteht, die in **Bild 1** gezeigten Komponenten ausgewählt und heruntergeladen.

Starten Sie Visual Studio dann wie gewohnt. Der neuartige Startassistent lässt sich durch Anklicken des Links *Ohne Code fortfahren* deaktivieren, wenn die IDE nicht sowieso mit einem schon geöffneten Projekt durchstartet.

Die Firma Wilderness Labs liefert das eigentliche SDK in Form eines Visual-Studio-Plugins aus. Klicken Sie auf *Erweiterungen* → *Erweiterungen verwalten*, um den Plugin-Assistenten zu laden. Dann wechseln wir, wie in **Bild 2** zu sehen, in die Rubrik *Online* und suchen nach dem String *Meadow*.

Nach dem obligaten Neustart gibt es dann eine neue Vorlage namens *Meadow Application*, die als Basis für eigene Experimente dient. Erzeugen Sie ein neues Programm mit der Bezeichnung „*ElektorSample*“, um sich am Code des im Rahmen des Bootloaders ausgelieferten LED-Blinkbeispiels zu erfreuen. Der in der Datei `MeadowApp.cs` befindliche Code dürfte selbsterklärend sein.

Von Arduino umsteigende Entwickler sollen nur darauf achten, dass das Framework selbst keine Schleifeninfrastruktur mitbringt. Das vorgegebene Beispiel initialisiert die beiden Methoden einfach aus dem Konstruktor; in `BlinkLeds` findet sich dann eine Endlosschleife:

```
public MeadowApp()
{
    ConfigurePorts();
    BlinkLeds();
}
```

## Start- und Pingtest

Muss ein Prozessrechner harte Echtzeit-Aufgaben bewältigen, so sollte das Betriebssystem „garantierte“ Reaktionszeiten bieten. Systeme, die auf verwaltete Sprachen wie Java oder C# aufgebaut sind, haben damit im Allgemeinen ihre liebe Not. Das liegt nicht nur an Ineffizienzen der Runtime - läuft der Garbage-Collector (automatische Speicherbereinigung), so ist keine andere Aufgabe ausführbar. Unser kleines Testprogramm kommt daher fast vollständig ohne dynamische Speicherallokation aus. Wenn aber die Steuerungsaufgabe viel Speicher alloziert und freigibt, besteht ein höheres Risiko von Garbage-Collector-Läufen.

Um das Verhalten des F7 zu testen, soll er (wie immer) eine charakteristische Wellenform ausgeben. Hierzu werden sowohl `ConfigurePorts` als auch `BlinkLeds` wie folgt angepasst:

```
public class MeadowApp : App<F7Micro, MeadowApp> {
    IDigitalOutputPort myOut;

    public MeadowApp()
    . . .
    public void ConfigurePorts() {
        Console.WriteLine("Creating Outputs...");
        myOut= Device.
        CreateDigitalOutputPort(Device.Pins.D05);
    }

    public void BlinkLeds() {
        var state = false;

        while (true) {
            myOut.State = true;
            myOut.State = false;
            myOut.State = true;
            myOut.State = false;
        }
    }
}
```

Die Meadow-Software verwendet für die Interaktion mit physikalischen Elementen Abstraktionsklassen. Unser digitaler Port entsteht beispielsweise durch ein Interface vom Typ `IDigitalOutputPort` - würde man eine Erweiterung anbieten, die ebenfalls GPIO-Pins exponiert, könnte man (bei passendem Treiber) Code zwischen dem „gewöhnlichen“ und dem neuen Peripheriegerät hin- und herschieben.

Zum Test wird ein digitales Speicheroszilloskop von Tektronix, ein TDS754D mit einer „gehackten“ Bandbreite von 1 GHz, zum Einsatz gebracht. Ein solches Gerät ist in/aus den USA gebraucht, aber frisch gewartet und sogar mit einem optionalen LCD versehen, recht preiswert erhältlich.

Da der Meadow F7 aus Sicht des Windows-Betriebssystems „nur“ ein COM-Port ist, muss man Visual Studio beim Deployment ein wenig auf die Sprünge helfen. Klicken Sie im ersten Schritt auf

*View Other Windows Meadow (oder Ctrl+Shift+M),*

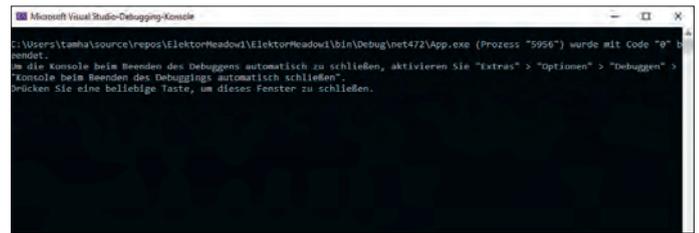


Bild 3. Dieses Fenster ist im Weg und muss weg.

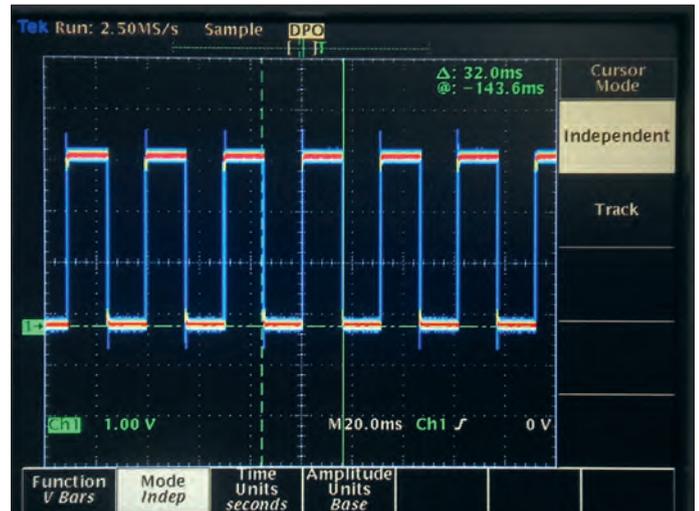


Bild 4. Eine Periode á 32 ms: Schnell sieht anders aus.

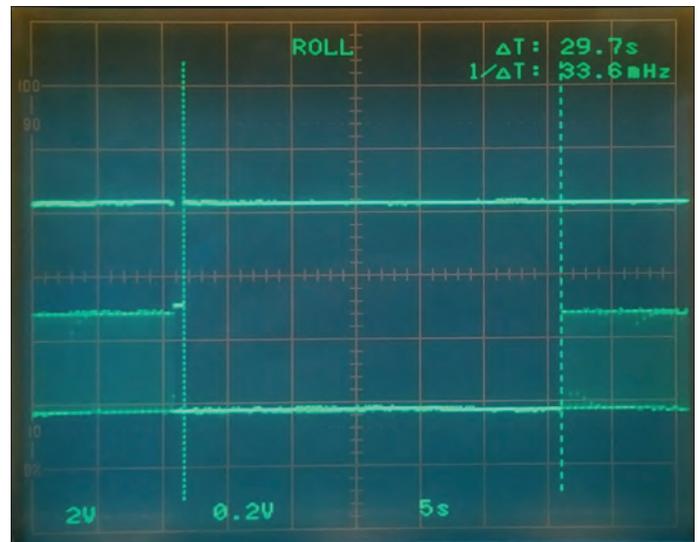


Bild 5. Gemütliche Startzeit von 30 s.

um das als „Meadow Device Explorer“ bezeichnete Geräteauswahlfenster zu aktivieren. Wählen Sie Ihren Meadow, um anschließend einen Debugging-Run zu befehlen. Im Rahmen jedes Programmstarts erscheint zudem das Fenster in **Bild 3**, das Sie durch ein beherztes Drücken der Enter-Taste schließen müssen.

Verbinden Sie das Oszilloskop nach dem erfolgreichen Deployment mit Pin D05 und erfreuen Sie sich an dem Oszillogramm in **Bild 4**. Dass die Runtime während der Programmausführung einige

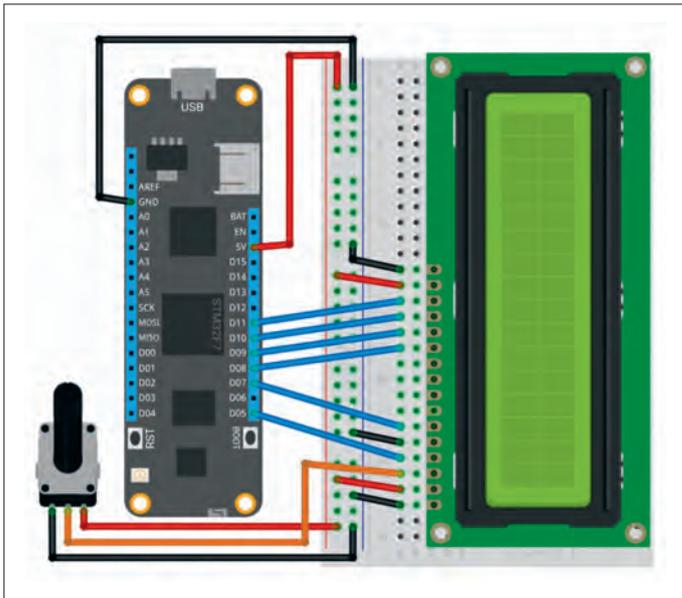


Bild 6. Hier wird es auf der Steckplatine etwas eng (Bildquelle: Wilderness Labs [5]).

Fehler in Bezug auf Datumsklassen erzeugt, ist dabei normal. Wer in der Schulstunde über Assembler-Programmierung aufgepasst hat, bekommt an dieser Stelle zwar Kopfschmerzen, allerdings ist der Unterschied der Wellentäler sehr gering. Das bedeutet, dass der Gutteil der Zeit auf das Umschalten der Pins entfällt - eine Situation, die man durch Optimierungen beheben könnte.

Der Reset-Anschluss ist nicht nur auf den Taster, sondern auch auf einen Pin der langen Buchsenleiste (erster Anschluss neben dem Spannungswandler) gelegt. Wer dort ein Oszilloskop mit Roll-Modus anschließt und Geduld mitbringt, kann eine Startzeit von fast 30 s beobachten (**Bild 5**)!

Ideal für erste Versuche ist es, wie in **Bild 6** ein Display an den F7 zu koppeln. Das oben beschriebene Designparadigma der Abstraktionsklassen wird auch beim `CharacterDisplay` fortgesetzt. Unsere erste Amtshandlung besteht darin, die `MeadowApp`-Klasse um eine weitere Instanz zu erweitern, die für die Kommunikation mit dem Display verantwortlich sein wird.

```
public class MeadowApp : App<F7Micro, MeadowApp> {
    CharacterDisplay myDisplay;
```

Der Aufbau von Hardwaretreibern folgt unter Meadow immer demselben Schema. Der Konstruktor nimmt zuerst einen Verweis auf ein `Device`-Objekt entgegen - wer das `Device` übergibt, weist den Treiber dazu an, die Ausgabe-Hardware des Prozessrechners zu benutzen. Analog zu Microsofts Kinect-SDK handelt es sich um eine Maßnahme zur Flexibilitätssteigerung (theoretisch könnte man auch einen GPIO-Extender realisieren).

Im nächsten Schritt folgt eine Gruppe benannter Parameter, die die verwendeten Ausgabepins beschreibt. An sie werden Werte aus der `Enum Device.Pins` übergeben - sie enthält für jedes Hardware-Peripheriegerät des STM32-Prozessors ein eigenes Bitfeld, was die Ansteuerung erleichtert:

```
public void ConfigurePorts()
{
    Console.WriteLine("Creating Outputs...");
    myDisplay = new CharacterDisplay(
        Device,
        pinRS: Device.Pins.D05,
        pinE: Device.Pins.D07,
        pinD4: Device.Pins.D08,
        pinD5: Device.Pins.D09,
        pinD6: Device.Pins.D10,
        pinD7: Device.Pins.D11,
        rows: 4, columns: 20
    );
}
```

An dieser Stelle bemerken wir, dass Visual Studio den Verweis auf die `CharacterDisplay`-Klasse nicht auflösen kann. Das liegt an der modularen Auslieferungsweise des Frameworks - klicken Sie das Projekt im Projektmappen-Explorer rechts an und entscheiden Sie sich für den NuGet-Paketverwalter. Suchen Sie danach in der Rubrik „Durchsuchen“ nach dem String `Meadow*Character` - das Sternchen ist eine Wildcard, die beliebigen Text erbeutet. Lohn der Mühen ist das Paket `Meadow.Foundation.Displays.LCD.CharacterDisplay`, das sie wie ein normales NuGet-Paket installieren.

Damit fehlt nur noch die eigentliche Ausgabefunktion, die Text zum Display schickt. Wilderness Labs nutzt hier die vom .NET-Framework bereitgestellte Infrastruktur; die Syntax der `writeLine`-Methode dürfte Ihnen bekannt vorkommen. Der zusätzliche numerische Parameter legt fest, in welcher Zeile des Displays der angelieferte String dargestellt werden soll. Wer den Wert „1“ übergibt, schreibt in die zweite Zeile von oben:

```
public void BlinkLeds()
{
    var state = false;

    while (true)
    {
        myDisplay.WriteLine("Hallo Elektor", 1);
        System.Threading.Thread.Sleep(1000);
    }
}
```

Laden Sie das Programm nach getaner Arbeit abermals auf den Meadow, um sich an der Anzeige zu erfreuen. Wer das im Hack Kit beigelegte Modul verwendet, muss auf die Kontrasteinstellung achten - sonst kann es passieren, dass die Zeichen zwar da, aber nicht zu sehen sind. Auffällig ist in diesem Zusammenhang zudem, dass der Aufbau des Bildschirms vergleichsweise langsam vonstatten geht - auch hier besteht zum Zeitpunkt der Drucklegung also noch Optimierungspotenzial.

### Erwartungsvoll...

Microsofts mittlerweile wieder abgekündigte Gadgeteer-Plattform wollte Hardware-unerfahrenen Entwicklern ermöglichen, schnell und problemlos Prototypen zu erstellen. Hierzu bot man eine Gruppe vorgefertigter Module an, die über „hauseigene“ Flachbandkabel Kontakt zu Basisboards aufnehmen.

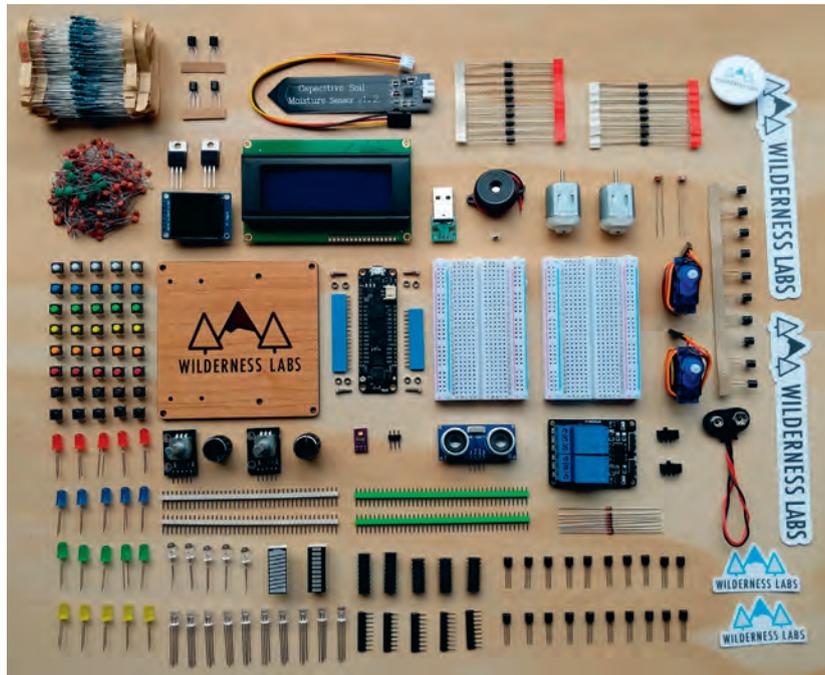


Bild 7. Viel Material zum Ausprobieren steckt im Development-Hack-Kit (Bildquelle: Wilderness Labs [6]).

Dieser Ingenieursgeist lebt bei Wilderness Labs weiter. Zum Lieferumfang des „Meadow F7 Micro Development Kit w/Hack Kit Pro“ (**Bild 7**), das auf der Webseite [6] vorbestellt werden kann, gehören neben Meadow F7 zwei Steckboards, ein durchaus hochwertiges 4x20-alphanumerisches LCD, ein Berg von aktiven und passiven Bauteilen, Aktoren und Sensoren sowie als Höhepunkt eine echt hölzerne Entwicklungsplattform.

Dazu gesellt sich eine sehr umfangreiche Treiberbibliothek, die das Team um Costanich zur Verfügung stellt. Zum Zeitpunkt der Drucklegung ist der Treiber für das (hochwertige) Farb-LCD zwar noch nicht einsatzbereit, aber auch das Kit selbst soll erst im März an den Verkaufsstart gehen. Schauen Sie sich stattdessen erwartungsvoll andere Sensoren an, die in der Hardwareliste [4] von Wilderness Labs aufgeführt sind.

Ein altes englisches Sprichwort besagt, dass Bettler nicht wählerisch sein dürfen. Wer .NET im Embedded-Bereich verwenden möchte, hat derzeit nur die Wahl zwischen dem veralteten

Netduino oder dem Meadow. Wer mit der gebotenen IO-Performance auskommt, findet allerdings eine sehr umfangreiche Treiberbibliothek vor, die das schnelle Realisieren von Prototypen ermöglicht – das Versprechen des Gadgeteers lebt also weiter! ◀

191190-01



#### IM ELEKTOR-STORE

→ Buch: Visual Basic für Elektroniksteuerungen und Entwicklung (PDF)

[www.elektor.de/17031](http://www.elektor.de/17031)

#### Weblinks

- [1] Meadow F7: [http://beta-developer.wildernesslabs.co/Meadow/Getting\\_Started/Deploying\\_Meadow/](http://beta-developer.wildernesslabs.co/Meadow/Getting_Started/Deploying_Meadow/)
- [2] DFU-utils: <http://dfu-util.sourceforge.net/releases/dfu-util-0.9-win64.zip>
- [3] Windows-Bug:  
[www.hanselman.com/blog/HowToFixDfuutilSTMWinUSBZadigBootloadersAndOtherFirmwareFlashingIssuesOnWindows.aspx](http://www.hanselman.com/blog/HowToFixDfuutilSTMWinUSBZadigBootloadersAndOtherFirmwareFlashingIssuesOnWindows.aspx)
- [4] Peripherie: <http://developer.wildernesslabs.co/Meadow/Meadow.Foundation/Peripherals/>
- [5] Character Display:  
<http://beta-developer.wildernesslabs.co/docs/api/Meadow.Foundation/Meadow.Foundation.Displays.Lcd.CharacterDisplay.html>
- [6] Meadow Kit /w Hack:  
<https://store.wildernesslabs.co/collections/frontpage/products/meadow-f7-micro-development-board-w-hack-kit-pro>

# Praktisches ESP32-Multitasking

## Teil 2: Task-Prioritäten

Von Warren Gay (Kanada)

In Mikrocontroller-Projekten stehen Entwickler oft vor dem Problem, dass viele Prozessoraufgaben gleichzeitig ausgeführt werden müssen. Der ESP32 und die Arduino-IDE machen die Task-Programmierung einfach, da das beliebte FreeRTOS bereits in die Core-Bibliotheken integriert ist [1]. In diesem zweiten Teil der Reihe befassen wir uns mit den Prioritäten der Tasks.



Bild 1. Wieder kommt der ESP32 von Lolin mit OLED-Display zum Einsatz.

Bei der ESP32-Implementierung des FreeRTOS-Schedulers werden die Tasks entsprechend ihrer Priorität ausgeführt, die bei der Erstellung des Tasks zugewiesen wurde (was aber später geändert werden kann). Tasks mit höherer Priorität werden von der ausgewählten CPU zuerst berücksichtigt, während Tasks mit Null-Priorität als letzte beachtet werden. Die Ausführung von Prioritäten mag ein bekanntes Konzept sein, aber der FreeRTOS-Echtzeit-Scheduler funktioniert anders als Sie es von Linux oder Windows gewohnt sind. In diesem Artikel wird der Unterschied anhand beispielhafter Versuche demonstriert. Die ESP32-Implementierung umfasst maximal 25 Prioritätsstufen, die von Null bis 24 reichen. Standardmäßig laufen die Arduino-Funktionen `setup()` und `loop()` (erinnern Sie sich, diese Funktionen stammen aus demselben Main-Task [1]) auf dem Prioritätslevel 1.

### Es lebe der Unterschied!

Wie unterschiedlich kann die Taskplanung sein? Auf einem Linux-System zum Beispiel beeinflusst die Priorität die relative Dringlichkeit des Prozesses oder Threads. Aber sogar ein Prozess mit niedrigster Priorität bekommt etwas CPU-Zeit ab: Normalerweise dauert es zwar länger, ihn auszuführen, aber der Prozess wird letztendlich doch abgehandelt. Und genau hier liegt der Unterschied.

In einem Echtzeitsystem wie FreeRTOS garantiert der Scheduler nicht, dass Aufgaben mit niedrigerer Priorität überhaupt jemals ausgeführt werden. Wenn Sie zum Beispiel Tasks mit der Priorität 9 haben, die immer *ready to execute* sind, dann wird kein Task mit Priorität 8 oder noch niedriger auf derselben CPU durchgeführt. Mit anderen Worten, die Tasks mit Priorität 9 lassen alle Tasks mit niedrigerer Priorität verhungern.

### Ready to execute

Es ist wichtig zu verstehen, was *ready* für FreeRTOS bedeutet. Ein Task ist *ready*, wenn er nicht dadurch blockiert ist, dass er auf irgendetwas wartet, sei es ein Ereignis, ein (neuer) Eintrag in eine Queue oder ein *Mutex*, der entriegelt werden muss (was das ist, wird in einer späteren Artikelfolge behandelt). Ein ausführungsbereiter Task wird entsprechend seiner Priorität in die *Ready*-Liste des Schedulers aufgenommen und ausgeführt, wenn er an der Reihe ist. Da es sich um eine nach Priorität sortierte Liste handelt, werden die Aufgaben mit der höchsten Priorität zuerst abgehandelt.

Tasks mit gleicher Priorität werden nach einem *Round-Robin*-Verfahren [2] ausgeführt. Drei *Ready*-Tasks mit gleicher Priorität 9 (a, b und c) wechseln sich ab:

- task9a
- task9b
- task9c
- task9a
- task9b
- und so weiter

Solange diese Tasks nicht von Tasks höherer Priorität blockiert werden, geht das ewig so weiter. So kann sich ein ESP32-Task mit hoher Priorität namens *idc1* (für CPU 1) vor die Tasks mit Priorität 9 „drängeln“, um einige Aufgaben zu erledigen. Sobald der Task *idc1* fertig und wieder *not ready ist*, werden die Tasks mit Priorität 9 dort fortgesetzt, wo sie unterbrochen wurden. Hier sind einige Beispiele dafür, warum ein FreeRTOS-Task **nicht** *ready* sein kann:

- *sleep* oder *delay* für eine Weile (auf einen Timer warten)
- warten auf *Mutex* oder *Semaphore*

- warten auf den Empfang einer Nachricht aus einer leeren Queue
- warten auf das Einfügen einer Nachricht in eine volle Queue
- warten auf ein FreeRTOS-Event oder eine Gruppe von Events
- warten auf den Abschluss einer I/O-Aktion
- *suspended* = ausgesetzt (entweder durch den Task selbst oder durch einen anderen Task).

Eine der Möglichkeiten, *blocked* zu werden, ist das Warten auf den Empfang einer Nachricht aus einer leeren Queue, denn ist sie leer, gibt es für diesen Task nichts zu tun. Der Scheduler entfernt diesen Task aus der Ready-Liste und sucht nach anderen, die ausgeführt werden sollen. Dabei werden nur Tasks auf der Ready-Liste berücksichtigt. Ist dort kein Task zu finden, wird stattdessen die Leerlauf-Task des Systems ausgeführt. Beachten Sie, dass der Funktionsaufruf `taskYIELD()` nicht bei den Gründen in der Liste aufgeführt ist. Wenn ein Task die Kontrolle aufgibt, sei es, weil seine ihm zugestandene Zeit abgelaufen ist oder auch freiwillig durch den Aufruf von `taskYIELD()`, geht die Kontrolle wieder an den FreeRTOS-Scheduler über, so dass dieser einen anderen Task für den nächsten Zeitabschnitt auswählen kann. *Yielding* ist etwas anderes als *Blocking*, da diese Tasks weiterhin ausführungsbereit sind und bei der nächsten Gelegenheit wieder CPU-Zeit erhalten.

### SMP-Änderungen von FreeRTOS in der ESP-IDF

FreeRTOS wurde für Single-Core-Mikrocontroller entwickelt. Da alle ESP32-Varianten (außer dem ESP32-S2) eine Dual-CPU besitzen, hat Espressif die Scheduler-Komponente angepasst. Kurz überblickt gibt es die folgenden ESP32-CPU's:

- CPU 0, bekannt als PRO\_CPU (Protokoll-CPU) und
- CPU 1, bekannt als die APP\_CPU (Applikations-CPU).

Espressif gibt an, dass „die beiden Kerne in der Praxis identisch sind und sich den gleichen Speicher teilen.“

Zur Unterstützung von symmetrischem Multiprocessing (SMP) gibt Espressif an, dass „der Scheduler Tasks überspringt, wenn er eine Round-Robin-Planung zwischen mehreren Tasks gleicher Priorität im Ready-Zustand durchführt.“ Dies ergibt sich aus der Beschränkung, wenn man eine Ready-Liste, die für eine einzelne CPU entwickelt wurde, nun auf einer Plattform mit zwei CPUs [3] verwendet.

Das Problem, mit dem sich Espressif konfrontiert sah, war folgendes: Wenn eine CPU einen Task-Kontextwechsel benötigt, um den nächsten Ready-Task auszuführen, kann sie nur auf eine einzige Task-Ready-Liste zum Durchsuchen zurückgreifen. Wenn also der Index der aktuellen Liste auf bereite Tasks für die andere CPU zeigt, dann müssen diese Einträge übersprungen werden, bis ein Eintrag für die betreffende CPU gefunden werden kann. Unter diesen Bedingungen kann ein Round-Robin-Verfahren nicht immer optimal sein.

Das Fazit für den Entwickler ist also, dass das Round-Robin-Verfahren bei den zwei CPUs des ESP32 nicht ganz fair ist. Bei vielen Projekten dürfte dies nicht weiter auffallen, aber wenn es problematisch wird, muss man Möglichkeiten nutzen, dies zu umgehen. Seien Sie sich dessen bei der Task-Planung bewusst.

### Ein Versuch

Wir werden jetzt ein Demo-Programm für das ESP32-OLED-Dis-

playmodul (**Bild 1**) nutzen. Durch Änderung einiger Makros im Programm können Sie die Prioritäten von vier verschiedenen Tasks im Programm einstellen. Das Programm ist so konzipiert, dass es drei Spannerrauen (engl. Inchworm) zeigt, die im OLED horizontal hin und her kriechen. Jede der Raupen kriecht aber nur dann hin und her, wenn „ihr“ Task CPU-Zeit erhält. Wird einem Task nur wenig CPU-Zeit gegönnt, so sitzt die Raupe still und kriecht nur ganz langsam.

Jede Raupe wird von einem Task bewegt, der seine CPU-Zeit aufbraucht und dann eine Nachricht an den vierten Task sendet. Dieser vierte Task ist dann dafür verantwortlich, dass die Raupe sich bewegt und angezeigt wird.

Der Code zum Zeichnen und Verwalten des Zustands der Raupe wird in der `InchWorm`-Klasse definiert (hier nicht gezeigt). In diesem Artikel konzentrieren wir uns einfach auf die Wirkung der Methode `InchWorm::draw()` für jede Raupe. Jede Instanz der `InchWorm`-Klasse verwaltet ihren eigenen Status und Fortschritt. Die Anzeige- und Raupeninstanzen werden im Programm wie folgt deklariert:

```
static Display oled;
static InchWorm worm1(oled,1);
static InchWorm worm2(oled,2);
static InchWorm worm3(oled,3);
```

Jedes Raupen-Objekt verwendet eine C++-Referenz (wie ein C-Pointer) auf die Display-Klasse im ersten und die Nummer der Raupe als zweites Argument. Die Referenz auf das Display ermöglicht eine zukünftige Erweiterung, zum Beispiel die Unterstützung mehrerer Displays. Die Nummer der Raupe bestimmt, wo sie (vertikal) auf dem OLED angezeigt wird (1, 2 und 3 bezeichnen die obere, die mittlere und die untere Zeile).

Der Task hinter jeder Raupe ist einfach eine Schleife, die die CPU-Zeit vergehen lässt, und ein Aufruf zum Senden der Nachricht:

```
void worm_task(void *arg) {
    InchWorm *worm = (InchWorm*)arg;

    for (;;) {
        for ( int x=0; x<800000; ++x )
            __asm__ __volatile__(„nop“);
        xQueueSendToBack(qh,&worm,0);
        // vTaskDelay(10);
    }
}
```

Es ist wichtig, die Funktion `vTaskDelay()` zunächst auskommentiert zu lassen. Sie wird in einem späteren Experiment verwendet werden.

Für alle drei Inchworm-Tasks wird dieselbe Task-Funktion verwendet, wobei das Argument `arg` angibt, welche Instanz der Raupe wir kriechen lassen wollen. Die Adresse der Raupe wird aus einem Void-Pointer umgewandelt und in der lokalen Variablen `worm` gespeichert. Sie wird nur innerhalb dieses Tasks verwendet, um als Nachricht an den Haupt-Display-Task gesendet zu werden, um anzugeben, welche Raupe sich hin- und herbewegt. Beachten Sie, dass in diesem Versuch beim Aufruf von `xQueueSendToBack()` der Wartezeit-Parameter mit Null angegeben wurde (drittes Argument). Dies weist FreeRTOS an, sich

wenn möglich in die Queue einzureihen, schlägt aber sofort fehl, wenn die Queue voll ist. Dies ist auch beabsichtigt, denn wenn die Queue voll wird, wollen wir nicht, dass unser Rau-pen-Task deren Ausführung blockiert. Der Task darf die CPU in diesem Versuch nicht freigeben, damit er die CPU wirklich für sich beanspruchen kann.

Die äußere `for`-Schleife hat die Aufgabe, ihre Operationen für immer auszuführen. Die innere `for`-Schleife für die CPU-Wartezeit führt 800.000 Mal eine No-Operation-Operation (`nop`) aus. Das Schlüsselwort `__volatile__` verhindert, dass der Compiler dieses Loop-Statement wegoptimiert. Ungeachtet dessen, was der Compiler denken mag, ja, wir wollen die CPU-Zeit tatsächlich verschwenden!

### Listing 1. Die `Setup()`-Funktion.

```
void setup() {
    TaskHandle_t h = xTaskGetCurrentTaskHandle();

    app_cpu = xPortGetCoreID(); // Which CPU?
    oled.init();
    vTaskPrioritySet(h,MAIN_TASK_PRIORITY);
    qh = xQueueCreate(4,sizeof(InchWorm*));

    // Draw at least one worm each:
    worm1.draw();
    worm2.draw();
    worm3.draw();

    xTaskCreatePinnedToCore(
        worm_task, // Function
        "worm1", // Task name
        3000, // Stack size
        &worm1, // Argument
        WORM1_TASK_PRIORITY,
        nullptr, // No handle returned
        app_cpu);

    xTaskCreatePinnedToCore(
        worm_task, // Function
        "worm2", // Task name
        3000, // Stack size
        &worm2, // Argument
        WORM2_TASK_PRIORITY,
        nullptr, // No handle returned
        app_cpu);

    xTaskCreatePinnedToCore(
        worm_task, // Function
        "worm3", // Task name
        3000, // Stack size
        &worm3, // Argument
        WORM3_TASK_PRIORITY,
        nullptr, // No handle returned
        app_cpu);
}
```

Nach Abschluss der Wartezeit-Schleife senden wir die Adresse der zu kriechenden Raupe an die Queue, identifiziert mit dem Handle `qh`. Sobald die Nachricht vom Display-Task empfangen wird, veranlasst er, dass unsere Raupe in der Anzeige vorrückt. Die Arduino-Haupttask `loop()` wird als der Display-Task verwendet, um das Kriechen der Raupe durchzuführen:

```
void loop() {
    InchWorm *worm = nullptr;

    if ( xQueueReceive(qh,&worm,portMAX_DELAY) )
        worm->draw();
}
```

Diese Schleife blockiert die Ausführung, bis einer der Tasks die Adresse der zu zeichnenden Raupe sendet. Sobald dieser Class-Pointer empfangen wird, wird die Methode `InchWorm::draw()` aufgerufen, um die Raupe zu zeichnen und voran kriechen zu lassen.

Die `setup()`-Funktion in **Listing 1** zeigt, wie die drei Rau-pen-Tasks und die Queue erstellt werden.

### Ändern der Priorität

FreeRTOS erlaubt es einem Task, seine eigene oder die Priorität eines anderen Tasks mit Hilfe der Funktion `vTaskPrioritySet()` zu ändern. Standardmäßig läuft der Task, der `setup()` und `loop()` aufruft, auf der Prioritätsstufe 1 (diese Funktionen werden vom selben Haupttask aufgerufen). Für diese Demo muss diese Priorität höher sein als die der anderen drei Raupen-Tasks. Die `setup()`-Funktion ändert die Priorität ihres eigenen Tasks wie folgt:

```
static int app_cpu = 0; // Updated by setup()
...
void setup() {
    TaskHandle_t h = xTaskGetCurrentTaskHandle();

    app_cpu = xPortGetCoreID(); // Which CPU?
    ...
    vTaskPrioritySet(h,MAIN_TASK_PRIORITY);
}
```

Wie zu sehen, erhält die `setup()`-Funktion ihr eigenes Task-Handle durch den Aufruf von `xTaskGetCurrentTaskHandle()` und dessen Speicherung in `h`. Durch die Änderung der Haupt-task-Priorität im Aufruf von `vTaskPrioritySet()` wird auch die von `loop()` verwendete Task-Priorität beeinflusst. Dies ist ein Beispiel dafür, wie Task-Prioritäten angepasst werden können. Im ersten Experiment werden den Raupen-Tasks die Prioritäten 9, 8 und 7 zugewiesen. Dies setzt voraus, dass unser Display- oder Haupt-Task die Priorität 9 oder höher besitzt (wir setzen sie auf 10). Wenn dies nicht getan wird, verliert der Haupttask `loop()` die CPU und kann die Raupen mehr nicht animieren.

### Welche CPU?

Im `setup()`-Schnipsel wurde eine weitere ESP32-API-Funktion namens `xPortGetCoreID()` eingesetzt, um herauszufinden, auf welcher CPU die Applikation läuft. Dies wird im Programm in der statischen Variablen `app_cpu` gespeichert, so dass der Code

weiß, für welche CPU neue Tasks bestimmt sind. Für den Dual-Core-ESP32 ist der Wert von `app_cpu` 1, das heißt, die Applikation wird in einer Dual-Core-Konfiguration auf CPU 1 ausgeführt. Bei Single-CPU-Plattformen wird `app_cpu` auf null gesetzt. Durch diese Kodierung kann die Applikation portabel auf Plattformen mit Single- oder Dual-Core-CPU's ausgeführt werden.

Diese Demo dürfte jedoch aufgrund der Monopolisierung der CPU auf einer Einzel-CPU-Plattform nicht gut funktionieren. Der Watchdog-Timer würde auslösen und Resets verursachen. Sie haben nun aber gelernt, wie bei anderen Anwendungen grundsätzlich eine Portabilität erreicht werden kann.

### Beispielhafte Konfiguration

Der Quellcode zu diesem Versuch ist unter [4] verfügbar. Am Anfang des Demonstrationsprogramms befinden sich Makrodefinitionen, die jedes Experiment konfigurieren:

```
// Worm task priorities
#define WORM1_TASK_PRIORITY 9
#define WORM2_TASK_PRIORITY 8
#define WORM3_TASK_PRIORITY 7

// loop() must have highest priority
#define MAIN_TASK_PRIORITY 10
```

Belassen Sie diese Prioritäten zunächst wie beim ersten Experiment.

### Andere OLED-Anzeige

Falls Sie nicht das empfohlene ESP32 Lolin Board mit seinem integrierten OLED verwenden, können Sie die Displayeinstellungen selbst definieren:

```
Display(
  int width=128,
  int height=64,
  int addr=0x3C,
  int sda=5,
  int scl=4);
```

Wenn Ihre Einstellungen korrekt konfiguriert sind, sollte das OLED beim Programmstart sofort weiß werden. Wenn nicht, überprüfen Sie die Verbindungen und Einstellungen erneut.

### Versuch 1

Sie können den heruntergeladenen Code einfach kompilieren, flashen und die Anwendung ausführen. Ihr OLED sollte sofort weiß werden und drei schwarze Raupen zeichnen (siehe **Bild 2**). Die Konfiguration für dieses Experiment ist (wieder):

```
#define WORM1_TASK_PRIORITY 9
#define WORM2_TASK_PRIORITY 8
#define WORM3_TASK_PRIORITY 7
#define MAIN_TASK_PRIORITY 10
```

Diese Konfiguration bewirkt, dass sich die obere Raupe entlangschlängelt, während die beiden unteren still stehen. Die Frage ist: Warum bewegen sich der mittlere und der untere Wurm nicht?

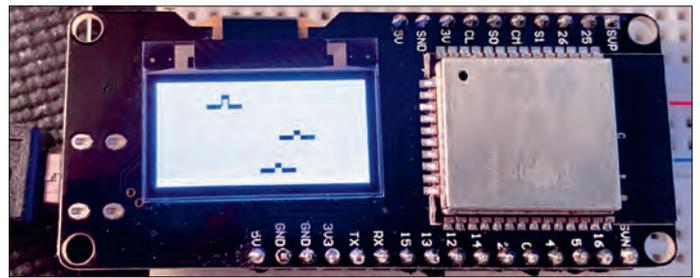


Bild 2. Wenn Tasks ausgeführt werden, bewegen sich die „Versuchstiere“.



Erinnern Sie sich, dass wir dem Haupt-Task die Priorität 10 belassen haben? Er genießt also die höchste Priorität aller Tasks in unserer Applikation. Die erste Raupe in der obersten Zeile des OLED konnte sich weiterbewegen, da er der einzige Task mit der Priorität 9 war, der durch die CPU ausgeführt werden konnte, weil der Display-Task mit Priorität 10 die Ausgabe zum OLED ausführt und dann darauf wartet, dass Nachrichten in der Nachrichten-Queue ankommen (und somit blockiert wird). Wenn aber der Display-Task blockiert ist, können andere Tasks mit niedrigerer Priorität eingeplant werden.

Die Tasks mit Priorität 8 und 7 für die mittlere und die untere Raupe aber werden nie ausgeführt, weil der Task mit Priorität 9 die CPU vollständig monopolisiert. Dies liegt in der Natur des Echtzeit-Scheduling in FreeRTOS. Im Gegensatz zu Linux oder Windows erhalten Tasks mit niedrigerer Priorität keinerlei Chance auf eine Ausführung.

### Versuch 2

Für das zweite Experiment ändern Sie die Konfiguration so, dass alle drei Raupen die gleiche Priorität erhalten und der Haupt-Display-Task bei Priorität 10 belassen wird. Stellen Sie alle drei Raupen-Tasks auf die gleiche Priorität 9, 8 oder 7 ein. Ich werde hier Priorität 9 verwenden:

```
#define WORM1_TASK_PRIORITY 9
#define WORM2_TASK_PRIORITY 9
#define WORM3_TASK_PRIORITY 9
#define MAIN_TASK_PRIORITY 10
```

Was sehen Sie, wenn Sie den Code nun neu kompilieren und wieder in den ESP32 flashen? Dies:



Die Raupen marschieren jetzt im gleichen Tempo (oder fast im gleichen Tempo) über den Bildschirm. Nur wenn Sie den Versuch lange genug laufen lassen, könnte ein Wurm den anderen ein wenig voraus sein.

### Versuch 3

Ändern Sie in diesem Experiment die Konfiguration, um wie im letzten Versuch den drei Raupen die gleiche Priorität zu geben,

aber setzen Sie auch den Haupt-Task auf die gleiche Priorität. Ich werde all diesen Tasks die Priorität 9 zuteilen.

```
#define WORM1_TASK_PRIORITY 9
#define WORM2_TASK_PRIORITY 9
#define WORM3_TASK_PRIORITY 9
#define MAIN_TASK_PRIORITY 9
```

Was beobachten Sie nun nach der Neukompilierung und dem erneuten Flashen, wenn Sie den Code ausführen? Sehen Sie den Unterschied? Warum kommen die Raupen mit unterschiedlicher Geschwindigkeit voran?

```
--
  --
    --
```

Wenn ich den Code ausführe, scheint die untere Raupe die meiste CPU-Zeit abzubekommen (sie kriecht am schnellsten). Die obere Raupe bewegt sich am langsamsten. Auch hier ist die von Espressif festgestellte Einschränkung der Round-Robin-Fairness daran schuld. Im Idealfall sollte der Displaytask nur ein wenig CPU-Zeit benötigen, um die Raupe zu zeichnen. In der restlichen Zeit sollten sich die drei übrigen Raupen-Tasks die verbleibende CPU-Zeit brüderlich zu gleichen Teilen teilen. Dennoch sehen wir, dass die Zeitplanung unausgewogen ist. Beide CPUs reagieren auf Timer und andere Interrupts. Der fehlerhafte Scheduler-Code ist dafür verantwortlich, dass die Fairness des Round-Robin-Verfahrens gestört wird.

#### Versuch 4

Bei jedem der bisherigen Versuche hat jede Raupe so viel CPU-Zeit verbraucht, wie sie bekommen konnte. Wie ändert sich das Verhalten, wenn wir eine kleine Verzögerung (zum Blockieren) innerhalb der Schleife einführen? Stellen Sie die Konfiguration so auf den Anfangszustand zurück, dass die Haupt-Display-Task die Priorität 10 und die Raupen-Tasks wieder die Prioritäten 9, 8 und 7 haben.

```
#define WORM1_TASK_PRIORITY 9
#define WORM2_TASK_PRIORITY 8
#define WORM3_TASK_PRIORITY 7
#define MAIN_TASK_PRIORITY 10
```

Dann heben Sie die Auskommentierung der Zeile auf, in der `vTaskDelay()` aufgerufen wird, so dass die Task-Schleife wie folgt aussieht:

```
void worm_task(void *arg) {
    InchWorm *worm = (InchWorm*)arg;
```

```
for (;;) {
    ...
    for ( int x=0; x<800000; ++x )
        __asm__ __volatile__(„nop“);
    xQueueSendToBack(qh,&worm,0);
    vTaskDelay(10); // Uncommented
}
}
```

Jetzt verbraucht jeden Raupe-Task CPU-Zeit, versucht, eine Raupe in die Warteschlange zu stellen und blockiert dann 10 Millisekunden lang. Kompilieren Sie dieses Beispiel, flashen Sie es und führen Sie es aus. Was beobachten Sie?

Die obere Raupe bewegt sich am schnellsten und die untere am langsamsten. Die obere Raupe bekommt aufgrund ihrer hohen Priorität 9 (während der Display-Task blockiert ist) den ersten Zuschlag in der CPU. Wenn der Raupen-Task aber durch den `vTaskDelay(10)`-Call blockiert ist, verbraucht der Task mit der nächstniedrigeren Priorität (die mittlere Raupe) etwas CPU-Zeit und ruft ihrerseits `vTaskDelay(10)` auf. Dies wiederum ermöglicht es dem noch niedrigeren Task mit der Priorität 7, einige CPU-Zyklen zu erhalten. Dies hat einen Pferdeäpfel-Effekt [5], der die CPU von der höchsten zur niedrigsten Ebene aufteilt. Beachten Sie jedoch, dass die Tasks mit Priorität 8 und Priorität 7 immer dann gestoppt werden, wenn der Task mit der höheren Priorität 9 wieder bereit ist. Deshalb bewegt sich der oberste Wurm am schnellsten. Die mittlere Raupe wiederum kann manchmal den Task mit Priorität 7 stoppen, so dass sie tendenziell schneller ist als die untere Raupe.

#### Weitere Experimente

Was passiert, wenn Sie die `vTaskDelay()`-Zeit viel länger als 10 Millisekunden machen? Versuchen Sie, sich die Antwort erst zu überlegen und führen Sie erst dann den Versuch aus. Warum haben Sie dieses Ergebnis erhalten? Was passiert eigentlich, wenn Sie die Verzögerungszeit auf 1 Millisekunde reduzieren? Diese Erkundungen bleiben Ihnen überlassen!

#### Konfiguration der Priorität

Obwohl wir Interrupts im ESP32 noch nicht behandelt haben, sollten Sie sich über die Header-Datei `FreeRTOSConfig.h` informieren, die die Prioritäten für die Plattform konfiguriert und

**IM ELEKTOR-STORE**  
→ Display-Modul ESP32 OLED Lolin  
[www.elektor.de/lolin-esp32-oled-module-with-wifi](http://www.elektor.de/lolin-esp32-oled-module-with-wifi)

#### Weblinks

- [1] Praktisches ESP32-Multitasking (1), Elektor 1/2020: [www.elektormagazine.de/190182-02](http://www.elektormagazine.de/190182-02)
- [2] Round-Robin-Verfahren: [https://de.wikipedia.org/wiki/Round\\_Robin\\_\(Informatik\)](https://de.wikipedia.org/wiki/Round_Robin_(Informatik))
- [3] Symmetric Multiprocessing: <https://thc420.xyz/esp-idf/file/docs/en/api-guides/freertos-smp.rst.html>
- [4] Projekt Source-Code: [https://github.com/ve3wwg/esp32\\_freertos/blob/master/priority-worms1/priority-worms1.ino](https://github.com/ve3wwg/esp32_freertos/blob/master/priority-worms1/priority-worms1.ino)
- [5] Pferdeäpfel-Theorie: <https://de.wikipedia.org/wiki/Trickle-down-Theorie>

hier zu finden ist:

`$IDF_PATH/components/freertos/include/freertos/  
FreeRTOSConfig.h`

Der Header definiert die folgenden Prioritäten-Makro-Werte. Die kompilierten Werte sind.

```
configMAX_PRIORITIES = 25  
configKERNEL_INTERRUPT_PRIORITY = 1  
configMAX_SYSCALL_INTERRUPT_PRIORITY = 3
```

Das erste Makro definiert die maximale Anzahl der verfügbaren Prioritäten. Das bedeutet, dass die gültigen Prioritätsnummern von 0 bis 24 reichen.

Das zweite Makro definiert die vom Kernel selbst verwendete Priorität für Interrupts. Damit verbunden ist das dritte Makro, das die höchste Priorität festlegt, die vom Kernel für Interrupts verwendet wird. Jeder FreeRTOS-API-Aufruf, der **innerhalb** einer Interrupt-Service-Routine (ISR) erfolgt, darf nur FreeRTOS-API-Funktionen mit Namen aufrufen, die auf `FromISR()` enden. Außerdem können diese Funktionen mit den gezeigten Werten nur von den Interrupt-Task-Prioritäten 1 bis einschließlich 3 aufgerufen werden. Wenn keine `FromISR()`-Calls getätigt werden, kann die ISR frei mit den Prioritäten 4 bis einschließlich 24 arbeiten.

## Zusammenfassung

Was können wir aus diesen Experimenten schließen? Sah das Konzept der Prioritäten zuerst scheinbar einfach aus, entpuppte es sich als voller Fallstricke. Die Konsequenz ist, dass es zu Überraschungen kommen kann, wenn Ihre Task-Prioritäten nicht gut geplant sind. Wir haben noch nicht über Watchdog-Timer gesprochen, aber das wirkt sich auch auf sie aus. Wenn beispielsweise der Watchdog-Timer in CPU 0 auslöst, wird Ihr ESP32 zurückgesetzt und neu gestartet.

Für den Dual-Core-ESP32 gibt es ein zusätzliches Problem, dass das Round-Robin-Verfahren auf derselben Prioritätsebene zu ungleichen Ausführungszeiten führen kann. Dies kann für einige Anwendungen problematisch sein, bei anderen spielt es keine Rolle. Das Problem hängt von der Art Ihrer Anwendung ab.

Für viele Anwendungen können Sie einfach Tasks erstellen, die mit Priorität 1 ausgeführt werden sollen. Dies ist die Priorität, die die Arduino-Tasks `setup()` und `loop()` besitzen. Tasks mit höherer Priorität können sicher verwendet werden, wenn sie durch eine Queue, einen Semaphore oder ein anderes Ereignis blockiert werden. Wenn ein Task blockiert oder angehalten wird, wird die CPU von anderen Tasks gleicher oder niedrigerer Priorität verwendet. Eine Anwendung mit ordnungsgemäß konfigurierten Taskprioritäten arbeitet wie eine gut geölte Maschine. ◀

191195-03

Anzeige

# Schaltungs-Sonderheft 2020



Dieses Schaltungs-Sonderheft enthält mehr als 90 kleine Schaltungen, Tipps und Tricks. Der Inhalt wurde aus veröffentlichten Elektor-Büchern und -Zeitschriften der letzten 10 Jahre ausgewählt. Bei der Auswahl der Artikel wurde darauf geachtet, dass die Schaltungen mit Standardkomponenten nachbaubar sind.

### Aus dem Inhalt:

- Drehgeber und Motordrehzahlanzeige mit RPi Zero W
- Eisenloser Kopfhörerverstärker mit 4x EL504
- 10-Volt-Referenzspannungsquelle
- Fotodiode misst Gammastrahlung
- Rechteckgenerator 125 Hz bis 4 MHz
- GPS-Außenantenne
- Diebstahlschutz über OBD
- 4-A-Solarlader
- Joule Robbin' Hood
- Motorregelung mit MCP3002 ADC und RPi

**FREI HAUS bestellen unter**  
[www.elektor.de/19140](http://www.elektor.de/19140)





# Mit dem Fuchs ins IoT (3)

## Erste Schritte im Netz

Von **Frank Schleking** und **Bernd vom Berg**

Das MKR FOX 1200-Board ist im Sigfox-Netz registriert, so dass dem ersten Kommunikationsversuch nichts mehr im Wege steht. Unser Sigfox-Device soll die weltberühmte Grußbotschaft „Hello World!“ in die Cloud übertragen.

Als Erstes schauen wir uns an, wie sich unsere Station in der Sigfox-Cloud präsentiert. Dazu melden Sie sich beim Sigfox-Backend [1] ganz normal mit E-Mail und Passwort an und gelangen so zur Startseite des Sigfox-Portals (**Bild 1**). Klicken Sie nun auf den Reiter *Device*. Sie erhalten eine Übersicht über die von Ihnen angemeldeten beziehungsweise aktivierten Sigfox-Geräte (**Bild 2**). Die Station MKR FOX 1200 sollte hier bereits (als einziges) eingetragen sein. In der kurzen Liste bedeuten:

- **Group:** Der Gruppenname wurde von Sigfox automatisch aufgrund Ihrer Angaben angelegt und kann von Ihnen nicht mehr geändert werden. In dieser Gruppe befinden sich nun alle Ihre Sigfox-Geräte. Durch Klicken auf den Gruppennamen erhalten Sie weitergehende Informationen zu dieser Gruppe (die ebenfalls nicht geändert werden können).

- **Device type:** Innerhalb der Gruppe kann man nun Stationen gleicher Art (gleich aufgebaut mit gleichen Funktionen, also identische Stationen) nach einzelnen Gerätetypen sortieren und einordnen, zum Beispiel Stationen zur Temperaturerfassung, Stationen zum Einlesen von

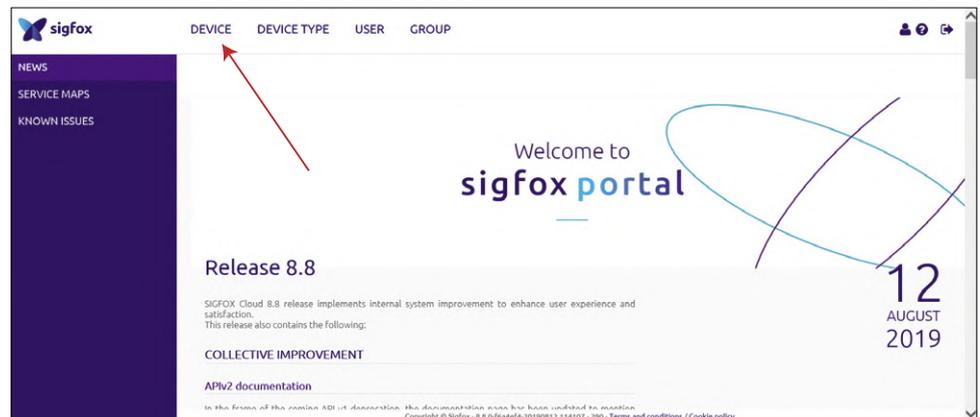


Bild 1. Das Sigfox-Portal.

analogen Sensorwerten, Stationen zur Erfassung von binären Signalen. In unserem Fall haben wir natürlich nur eine Stationsart, nur einen *Device type*, nämlich *Arduino\_DevKit\_1*. Später können Sie diesen Namen auch ändern und Ihren eigenen Bedürfnissen anpassen.

- **Id** (= Id-Nummer): Hier erscheint die eindeutige Id-Nummer der Station.
- **Last seen**: Hier trägt das Backend das Datum und die Uhrzeit ein, zu der sich die Station das letzte Mal gemeldet hat, also wann das letzte Mal ein Telegramm von der Station empfangen wurde. Die Angabe N/A bedeutet, dass das Backend noch kein einziges Telegramm von der Station empfangen hat.
- **Name**: Hier wird der Name der Station angezeigt.
- **Token state**: Ein Token ist ein Senderecht für eine Station im Sigfox-Netzwerk. Wenn man einen Vertrag mit Sigfox abschließt (und entsprechende Gebühren für die Netzwerknutzung bezahlt hat), erhält man eine bestimmte Anzahl von Tokens zur Verfügung. Soll nun eine Station im Sigfox-Netzwerk aktiv werden, so ordnet man dieser Station einen Token zu. Sie kann dann problemlos im Netzwerk Daten aussenden und empfangen. Beim Kauf eines Boards MKR FOX 1200 erhält man bei der Anmeldung im Sigfox-Backend von Sigfox kostenfrei einen Token für ein Jahr lang zur Verfügung gestellt. Das MKR FOX 1200-Board kann als einzelne Station ein Jahr lang im Netz arbeiten (jedem weiteren neuen MKR FOX 1200-Board erteilt Sigfox dann wiederum seinen eigenen einzigartigen Token). Der *Token state* zeigt nun an, ob die Station ihren Token schon einmal benutzt hat, ob die Station also schon einmal Daten in das Sigfox-Netzwerk gesendet hat. Da das bei uns noch nicht der Fall ist, steht hier ein Fragezeichen-Symbol; die Station hat ihr Senderecht noch nicht in Anspruch genommen. Sobald die Station aber ihr erstes Telegramm gesendet hat und damit ihr Senderecht genutzt hat, erscheint hier ein Häkchen. Erst dann ist der Token dieser Station endgültig zugeordnet.

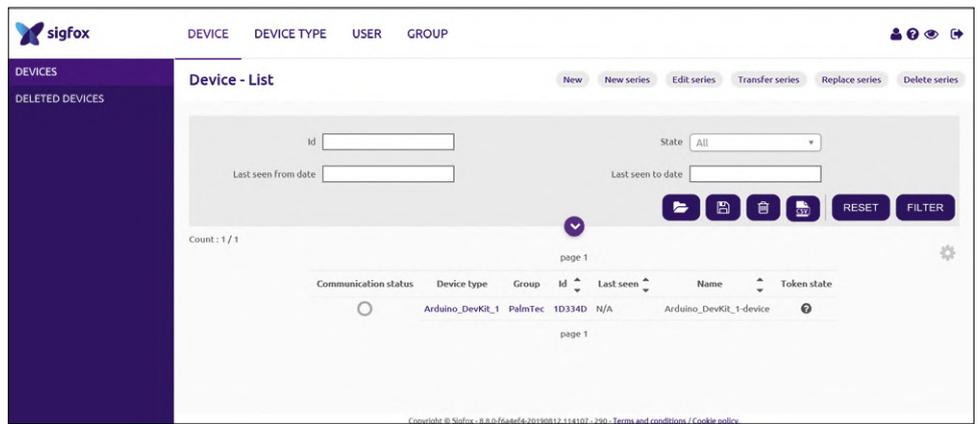


Bild 2. Die angemeldeten Sigfox-Devices.

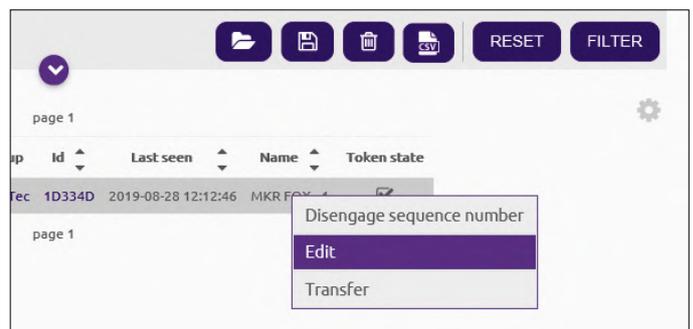


Bild 3. Das Pull-Down-Menü zum Name-Feld.

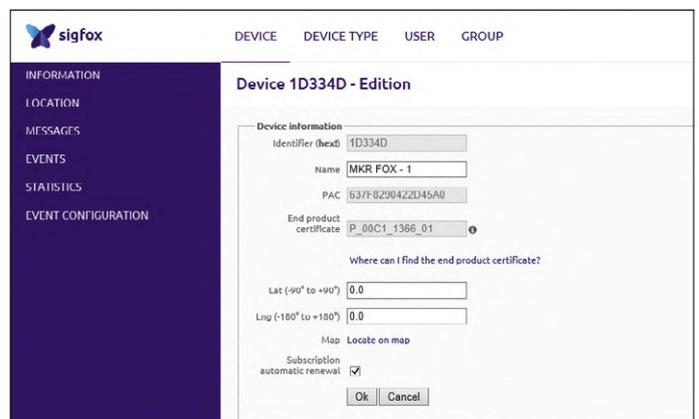


Bild 4. Das Editor-Fenster zum Device.

Klicken Sie nun mit der Maus auf das *Name*-Feld. Es erscheint ein kleines Pull-Down-Menü (**Bild 3**), in dem Sie den Punkt *Edit* auswählen. Im folgenden *Edition*-Fenster (**Bild 4**) werden einige Festlegungen zum *Device* gemacht oder geändert. Ändern Sie hier (bei Bedarf) den Namen der Sigfox-Station zum Beispiel in „MKR FOX – 1“. Alle anderen Felder bleiben unverändert. Klicken Sie abschließend auf *Ok*.

Sie gelangen nun zum Fenster *Device Information* (**Bild 5**), in dem zusammengefasst alle Informationen zum Sigfox-Modul angezeigt werden. Hier gilt: nur gucken, nicht anfassen (ändern). Wenn Sie oben links auf den *Device*-Reiter klicken, gelangen Sie wieder zurück zur *Device*-Übersicht, in der Sie die gemachte Namensänderung schon eingetragen sehen. Abschließend können Sie das Sigfox-Backend schließen.

## Hello World!

Damit die Aktivierung endgültig abgeschlossen wird und man mit der Station im Sigfox-Netz arbeiten kann, muss erst einmal eine gültige Nachricht gesendet werden. Von diesem Moment an beginnt die einjährige kostenfreie Sigfox-Mitgliedschaft (Subscription) mit 140 Nachrichten pro Tag im Upload und vier Nachrichten im Download.

Als erste Nachricht soll der weltbekannte Standard-Testtext „Hello World!“ an das Sigfox-Backend gesendet werden. Dieser String umfasst genau zwölf ASCII-Zeichen (= Bytes), er passt also exakt in einen Sigfox-Payload. Im letzten Teil des

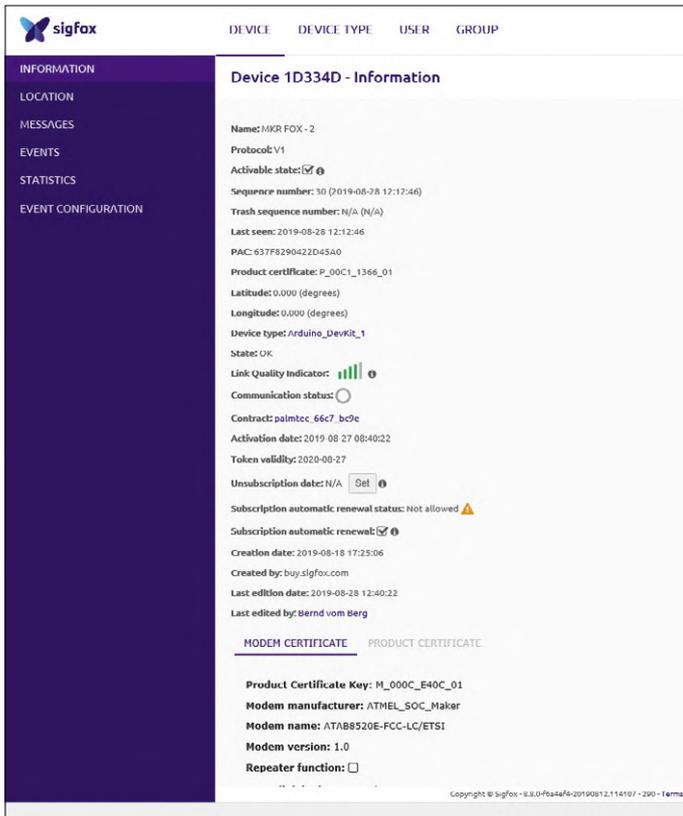


Bild 5. Das Informationsfenster zum jeweiligen Device.

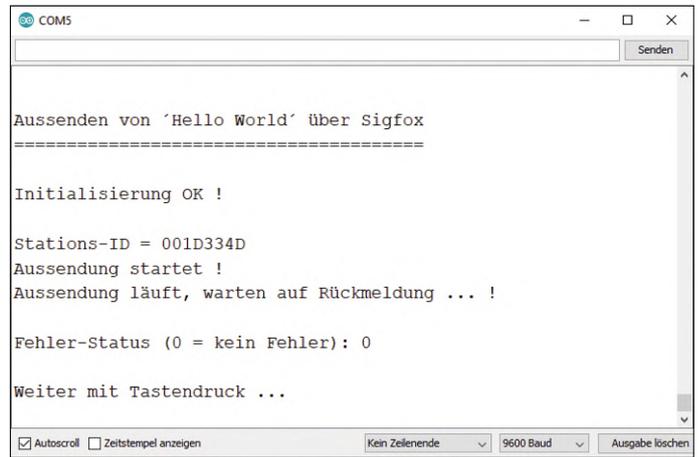


Bild 6. Das Aussenden des Test-Strings.

Artikels wurden schon die für den Sendebetrieb notwendigen Sigfox-Funktionen beschrieben, so dass der Kern der Senderoutine in **Listing 1** und die darin auftauchenden Funktionen leicht verständlich sind.

- **SigFox.begin**: Hiermit werden die Sigfox-Bibliothek und das Sigfox-Modem initialisiert und bei Nichterfolg ein Fehler zurückgegeben.
- **SigFox.debug**: Aktiviert den Debug-Modus. Die Energiesparfunktionen des Mikrocontrollers und des Modems werden deaktiviert, die gelbe LED des MKR FOX 1200-Board zeigt blinkend den Sende- und den Empfangsvorgang des Sigfox-Datentransfers an.
- **SigFox.beginPacket**: Diese Funktion leitet den Sendevorgang eines Sigfox-Paketes (= Sigfox-Payload) ein.
- **SigFox.print**: Der übergebene Wert oder String wird in den Payload geladen und gesendet. Zahlenwerte werden ebenfalls als ASCII-Strings gesendet. Bei Binärdaten („echte“ Zahlen, keine ASCII-Darstellung) wird die Funktion **SigFox.write** verwendet (dazu später mehr).
- **int ret = SigFox.endPacket**: Beendet die Sendung des Sigfox-Paketes (Sigfox-Payloads) und gibt den Fehlerstatus zurück: 0 (keine Fehler), 1 (Fehler).
- **SigFox.end**: Schließt die Sigfox-Bibliothek und das Sigfox-Modem.

So einfach kann eine komplexe Datenkommunikation abgewickelt werden, wenn dafür eine leistungsfähige Bibliothek zur Verfügung steht!

### Listing 1. Senden des Testtextes „Hello World!“.

```
// Sigfox-Modem aktivieren und Fehler abfragen
if (!SigFox.begin()) // Fehler aufgetreten
{
    Serial.println("Sigfox Modul Fehler!");

    while (1); // Im Fehlerfall: Endlosschleife
}

// Sigfox-Modem-Debug-Modus aktivieren
SigFox.debug();

// Nun soll der String ("Hello World!") übertragen werden

// Vorbereitung, um ein Paket zu senden
SigFox.beginPacket();

// Aussenden des Strings = Folge von Zeichen = Sigfox-Nachricht
SigFox.print("Hello World!");

// Beenden der Sigfox-Übertragung und Fehler abfragen:
// Status in der Variable ret
int ret = SigFox.endPacket();

Serial.print("\nFehler-Status (0 = kein Fehler): ");
Serial.println(ret);

// Deinitialisierung der Sigfox-Bibliothek und des Moduls
SigFox.end();
```

Laden Sie den Sigfox-Sketch auf das MKR FOX 1200-Board, öffnen Sie den seriellen Monitor und starten Sie den Sketch. Im Monitorfenster erscheint das Hauptmenü des Sketches. Wählen Sie nun den Menüpunkt *2 Aussenden von „Hello World!“*. Der beschriebene Ablauf zur Aussendung des Test-Strings läuft ab, wobei im Monitorfenster einige Zustandsinformationen ausgegeben werden. Gleichzeitig zeigt die blinkende gelbe LED auf dem MKR FOX 1200-Board den Sendevorgang an. Beendet werden sollte die komplette Aussendung ohne Fehlermeldung (**Bild 6**).

### Zurück im Backend

Wenn Sie innerhalb der Sigfox-Funkabdeckung liegen, erhält das Sigfox-Backend innerhalb kürzester Zeit die Daten. Das lässt sich leicht überprüfen, indem man das Sigfox-Backend startet, sich einloggt und im Startbildschirm auf den Reiter *Device* klickt. In der Liste der aktiven Sigfox-Devices klickt man (exakt) auf das *Id*-Feld der Station, so dass die Informationsseite zu dem Device erscheint, und dort auf der linken Seite auf *Messages*.

Im folgenden Fenster (**Bild 7**) sind alle Telegramme aufgeführt, die das Sigfox-Backend von der jeweiligen Station empfangen hat. Das Feld *Time* ist selbsterklärend, im Feld *LQI* (Link Quality Indicator) wird die Feldstärke angezeigt, mit der das Sigfox-Telegramm von den Basisstationen empfangen wurde. Fährt man mit der Maus auf dieses Feld, so erhält man eine Beschreibung zur Empfangsqualität und der lokale Betreiber des Sigfox-Netzwerkes, in dem sich die Sigfox-Station befindet, wird angezeigt. Der Betreiber muss nicht immer Sigfox heißen, es kann (beispielsweise im Ausland) auch ein Kooperationspartner sein. Beim Klick auf *Location* wird auf einer Karte (sehr grob) angezeigt, wo sich die Sigfox-Station ungefähr befindet.

Kommen wir nun zu den wichtigen Angaben im Feld *Data/Decoding*, wo der Inhalt des Payloads angezeigt wird, einmal als Rohdaten und einmal in dekodierter Form. Noch weiß das Backend nicht, wie die Rohdaten zu interpretieren sind, das muss ihm erst beigebracht werden. Schauen wir uns das Data-Feld etwas näher an:

```
48 65 6c 6c 6f 20 57 6f 72 6c 64 21
H e l l o   W o r l d !
```

Man erkennt, dass der String aus den ASCII-Codes der Zeichen

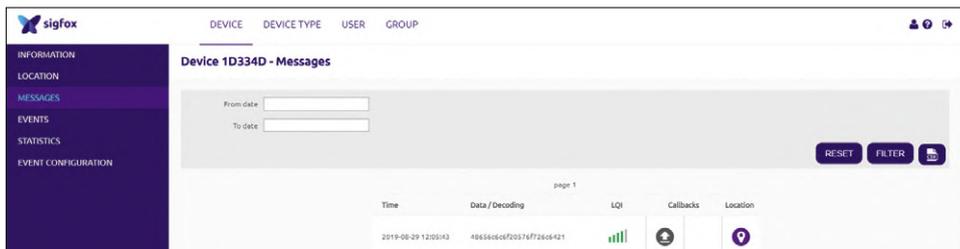


Bild 7. Das Message-Fenster des Devices.

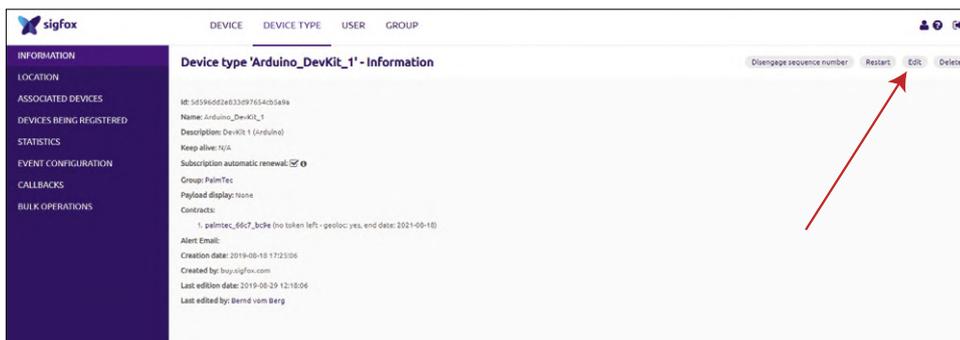


Bild 8. Das Informationsfeld zum Device type.

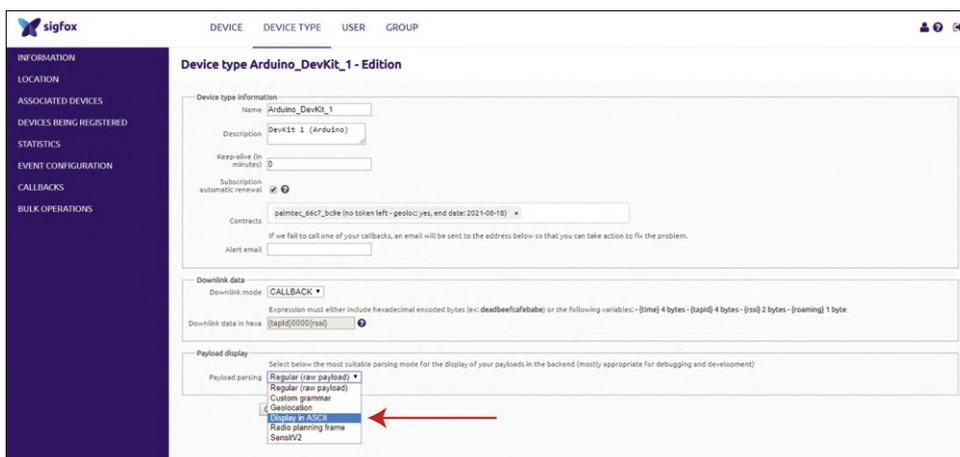


Bild 9. Das Editier-Fenster zum Device type.

des Testtextes „Hello World!“ (mit ASCII 20 für das Leerzeichen) besteht. Die Dekodierungsart lässt sich vom Fenster *Device-List* (Bild 2) aus einstellen. Klicken Sie genau auf den Namen *Arduino\_DevKit\_1* unter *Device type*, dann gelangen Sie zum *Information*-Feld zu diesem *Device type* (**Bild 8**), und nach einem Klick auf *Edit* zum dazugehörigen *Edition*-Fenster (**Bild 9**). Ändern Sie (nur) im Feld *Payload display* das *Payload parsing* auf *Display in ASCII* und bestätigen Sie die Änderung. Zurück im *Messages*-Fenster sehen Sie bei der Station eine zusätzliche Zeile mit der Darstellung des Payload-Inhaltes in ASCII-Zeichen (**Bild 10**).

Zur Übertragung von Zahlenwerten innerhalb des Payloads sind zwei grundlegende Schritte notwendig. Sie müssen zunächst festlegen, welche Art von Datentypen in dem 12 Byte großen Payload übertragen werden sollen, da jeder Datentyp aus

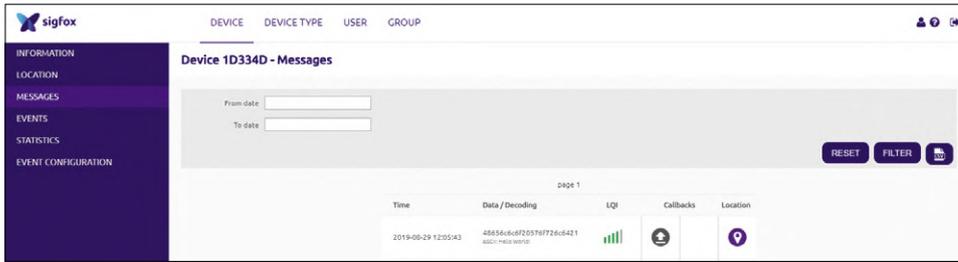


Bild 10. Die Dekodierung des Payload-Inhaltes in ASCII-Zeichen.

### Listing 2. Aufbau des benötigten Sigfox-Datentyps Sigi\_Dat\_1.

```
// Struktur/Datentyp 1 zu unserem Beispiel:
// Übertragung von festen Zahlenwerten
typedef struct __attribute__((packed)) sigfox_message {
    unsigned char wert_1; // Erster Wert im Payload: 1 Byte groß
    unsigned int wert_2; // Zweiter Wert im Payload: 4 Byte groß
    float wert_3; // Dritter Wert im Payload: 4 Byte groß
    unsigned char wert_4; // Vierter Wert im Payload: 1 Byte groß
} Sigi_Dat_1;
```

### Listing 3. Sendung mit festen Zahlenwerten.

```
// Definition von vier FESTEN Zahlenwerten
unsigned char AIN_1 = 0x1f; // 1. Wert, 1 Byte groß
unsigned int pressure = 0x12345678; // 2. Wert, 4 Byte groß
float temp = 1233.56; // 3. Wert, 4 Byte groß
unsigned char AIN_2 = 0x55; // 4. Wert, 1 Byte groß

// Sigfox-Modem aktivieren und Fehler abfragen
if (!SigFox.begin()) // Fehler aufgetreten
{
    Serial.println("Sigfox Modul Fehler! - Weiter mit RESET !");
    while (1); // Im Fehlerfall: Endlosschleife
}
else
{
    Serial.println("Sigfox-Modem OK !\n");
}

// Debug-LED aktivieren und Energiesparmodi deaktivieren
SigFox.debug();

// Löscht alle anstehenden Interrupts
SigFox.status();
delay(1);

// Nun werden ganz konkret die aktuell zu übertragenen Werte in die
// (Struktur)Variable 'SF_send' geschrieben,
// also: Zusammenbau des Inhaltes für das Payload
SF_send.wert_1 = AIN_1; // unsigned char - Wert: 1 Byte
SF_send.wert_2 = pressure; // unsigned int - Wert: 4 Byte
SF_send.wert_3 = temp; // float - Wert: 4 Byte
SF_send.wert_4 = AIN_2; // unsigned char - Wert: 1 Byte
```

Fortsetzung auf der nächsten Seite

Tabelle 1. Anzahl der Bytes verschiedener Datentypen.

Datentyp	Benötigte Bytes
char / unsigned char	1
int8_t / uint8_t	1
int16_t / uint16_t	2
int / unsigned int	4
int64_t / uint64_t	8
float	4

Tabelle 2. Mögliche Anordnung verschiedener Datentypen im Payload.

Wert im Payload	Datentyp	Bytes
1	unsigned char	1
2	unsigned int	4
3	float	4
4	unsigned char	1

unterschiedlich vielen Bytes besteht beziehungsweise unterschiedlich viele Bytes zur Speicherung im Payload benötigt. Die wichtigsten Datentypen zeigt **Tabelle 1**. Im Payload könnte man zu übertragene Datentypen wie in **Tabelle 2** anordnen. Von den zwölf im Payload zur Verfügung stehenden Bytes würden zehn Bytes benutzt. Als Zweites wird aus diesen Festlegungen ein eigener besonderer Datentyp definiert, in dem man sehr einfach die gewünschten Daten für das Payload zusammenhängend abspeichern kann. In unserem Beispiel sieht das aus wie in **Listing 2**. Um diese Konstruktion genau zu verstehen, sind tieferegehende C/C++-Kenntnisse erforderlich, aber man kann es vereinfacht erläutern.

Mit der Anweisung

```
struct ..... sigfox_message
```

und den nachfolgenden Festlegungen in den geschweiften Klammern stellen wir eine so genannte Struktur zusammen, die den Namen `sigfox_message` trägt. Solch eine Struktur ist nichts anderes als ein Satz zusammengehöriger Daten unter einem gemeinsamen (Haupt-)Namen. Sie kennen so etwas von einem Array, allerdings besteht ein Array stets aus Daten gleichen Typs. Bei einer Struk-

Fortsetzung von der vorherigen Seite:

```
// Und nun erfolgt das Sigfox-gemäße Aussenden der zuvor
// zusammengebauten Strukturvariablen 'SF_send'
// (= Inhalt des Payloads)

// Vorbereitung ein Packet zu senden
SigFox.beginPacket();

// Strukturvariable an das Sigfox-Backend senden
SigFox.write((char*)&SF_send, sizeof(SF_send));

// Fehlerprüfung: Wenn endPacket() eine 1 liefert, dann
// Fehlermeldung
int ret = SigFox.endPacket();
if (ret > 0)
{
    Serial.println("Fehler: Keine Übertragung ! - Weiter mit
        RESET !");
    while(1); // Endlosschleife
}
else
{
    Serial.println("Sigfox-Übertragung OK !");
}

// De-Initialisierung der Sigfox-Bibliothek und des
// Sigfox-Moduls
SigFox.end();
```

tur dagegen können Daten unterschiedlichen Typs zusammengestellt werden, hier aus zwei `unsigned char`-, einem `unsigned int`- und einem `float`-Wert. Die Datentypen der Elemente in der Struktur müssen exakt der Festlegung für das Payload entsprechen und auch in der richtigen Reihenfolge sein. Auf ein bestimmtes Element innerhalb der Struktur wird nun ganz einfach zugegriffen:

Name der Struktur.Name des Elements

Also wird zum Beispiel mit

```
sigfox_message.wert_3 =25.78;
```

das interne Float-Element `wert_3` auf den Wert `25.78` gesetzt.

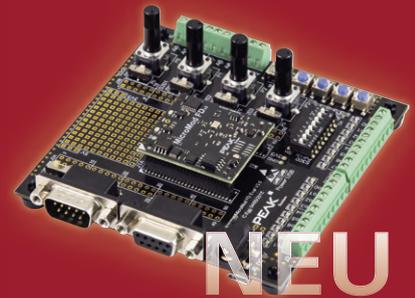
Und nun gehen wir noch einen Schritt weiter und generieren aus dieser Struktur einen ganz eigenen neuen Datentyp. Sie kennen natürlich die Grunddatentypen in C wie `unsigned char`, `int` oder `float`. Mit der zusätzlichen Anweisung `typedef` am Anfang der Strukturdefinition erzeugen wir einen weiteren Datentyp, der genau den Aufbau der nachfolgenden Struktur hat.

Dieser neue Datentyp benötigt natürlich auch einen neuen eigenen Namen, der am Ende der gesamten Definition angegeben wird, im Beispiel `Sigi_Dat_1`. Die Festlegung `__attribute__((packed))` bewirkt, dass überflüssige Füllbytes (Padding Bytes) aus der Struktur entfernt werden, damit wirklich nur die reinen Nutzdatenbytes in das Payload übertragen und gesendet werden. Diese gesamte Struktur wird nun direkt am Anfang des Sketches als globale Variable platziert, damit man von überall aus dem Sketch auf diesen Datentyp zugreifen kann. Will man nun mit diesem Datentyp arbeiten, so muss

Anzeige

You CAN get it...

Hardware und Software  
für CAN-Bus-Anwendungen



NEU

### PCAN-MicroMod FD

Universelles Einsteckmodul mit I/O-Funktionalität und CAN-FD-Interface. Erhältlich mit Evaluation-Board für die Entwicklung eigener Anwendungen.

ab 110 €



### PCAN-Router FD

Frei programmierbarer Router für CAN und CAN FD mit 2 Kanälen. Erhältlich mit D-Sub- oder Phoenix-Anschlusssteckern.

250 €



### PCAN-USB FD

CAN-FD-Interface für USB 2.0 inkl. Software, APIs und Treiber für Windows® und Linux.

245 €

Alle Preise verstehen sich zzgl. MwSt., Porto und Verpackung. Irrtümer und technische Änderungen vorbehalten.

www.peak-system.com

PEAK  
System

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

**Tabelle 3.**  
**Eintreffender Payload.**

1f	78563412	ec319a44	55
(A)	(B)	(C)	(D)

man natürlich Variablen zu diesen Datentyp definieren. Das geschieht ganz genau so, wie man es gewohnt ist, mit

```
Datentyp Variablenname;
```

Die Zeile `Sigi_Dat_1 SF_send;` legt eine Variable namens `SF_send` an, die genau die zuvor festgelegte Struktur hat (am besten auch als globale Variable festlegen). Auf die einzelnen Werte dieser Variablen kann dann zum Beispiel mit `SF_send.wert_1 = 12;` zugegriffen werden.

### Senden

Um zu erkennen, wie die Zahlenwerte im Sigfox-Payload verpackt sind, schreiben wir eine Funktion, die feste Zahlenwerte über das Sigfox-Netzwerk sendet (**Listing 3**). Die Funktion wird über den Menüpunkt 3 im Demo-Sketch aufgerufen. Viel erklären müssen wir dazu nicht mehr, nur dass zum Senden des Payloads nun die Funktion `SigFox.write` verwendet wird, da wir ja jetzt keine Zeichen, sondern Zahlenwerte aussenden. Wenn alles ordnungsgemäß funktioniert, erscheint im Backend der Payload wie in **Tabelle 3**.

Zum Abschluss entwickeln wir noch eine Funktion, mit der reale Messwerte übertragen werden (Menüpunkt 4 im Sigfox-Beispielsketch) und zwar den Temperaturmesswert des BMP280-Sensors (float, 4 Byte), den Luftdruckmesswert des BMP280-Sensors (float, 4 Byte) und den erfassten LDR-Messwert (dazu muss Jumper JP5 auf dem Motherboard in Position 1-2 gesteckt werden) vom Analogeingang A2 (uint16\_t, 2 Byte). Damit sind zehn der zwölf Bytes des Payloads aufgebraucht. Für diesen Datensatz definieren wir einen weiteren Sigfox-Datentyp (**Listing 4**) und eine neue Variable namens `SF_send_mw`:

```
// Variable zum Datentyp 'Sigi_Dat_2':
Sigi_Dat_2 SF_send_mw;
```

In einer neuen Funktion des Sketches werden die drei Messwerte erfasst, in die Strukturvariable geschrieben und gesendet (**Listing 5**). Kontrollausgaben über den seriellen Monitor und auf dem ePaper-Display zeigen sowohl die aktuellen Messwerte als auch die Abläufe bei der Sigfox-Kommunikation. Weitere Erläuterungen finden sich in dem gut dokumentierten Programm-Listing des Sketches.

### Ausblick

Damit sind wir am Ende unserer kleinen Einführung in das 0G-IoT-Netzwerk Sigfox, die wir mit Hilfe des Arduino Maker-Boards MKR FOX 1200 durchgeführt haben. Unsere erstellten Programme und Funktionen bieten Ihnen nun eine sehr gute

**Listing 4. Ein neuer Datentyp namens „Sigi\_Dat\_2“ zur Übertragung des Messwerte.**

```
// Struktur/Datentyp 2 zu unserem Beispiel: Übertragung von Messwerten
typedef struct __attribute__((packed)) sigfox_message_2 {
    float          wert_1; // 1. Wert im Payload: 4 Byte groß
    float          wert_2; // 2. Wert im Payload: 4 Byte groß
    uint16_t       wert_3; // 3. Wert im Payload: 2 Byte groß

    // Insgesamt werden hier also 10 Byte benötigt
} Sigi_Dat_2;
```

**Listing 5. Einschreiben der Messwerte in die Strukturvariable.**

```
// Nun werden ganz konkret die aktuell zu übertragene Werte in die
// (Struktur)Variable 'SF_send_mw' geschrieben,
// also: Zusammenbau des Inhaltes für das Payload
SF_send_mw.wert_1 = temp; // float - Wert: 4 Byte
SF_send_mw.wert_2 = pressure; // float - Wert: 4 Byte
SF_send_mw.wert_3 = LDR; // uint16_t - Wert: 2 Byte
```

Grundlage für die Realisierung eigener erfolgreicher Schritte im Sigfox-Netzwerk.

Was jetzt noch fehlt, ist die Visualisierung der Daten auf der Anwenderseite, also auf dem Anwender-PC/Laptop/Smartphone mit Hilfe eines frei konfigurierbaren Dashboards. Das soll das Thema des nächsten und letzten Teils der Serie sein. ◀

190281-C-01

**IM ELEKTOR-STORE**

→ Arduino MKR FOX 1200  
[www.elektor.de/19096](http://www.elektor.de/19096)

→ Arduino Antenne 868 MHz  
[www.elektor.de/19095](http://www.elektor.de/19095)

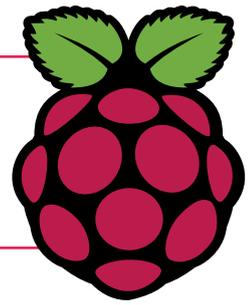


### Weblinks

[1] Login im Backend:  
<https://backend.sigfox.com/auth/login>

# Raspberry Pi

## Bash-Befehle in der Übersicht



Von **Clemens Valens** (Elektor-Labor)

Es ist praktisch unmöglich, mit dem Raspberry Pi (oder noch umfassender, mit Linux im Allgemeinen) zu arbeiten, ohne Befehle in einem Terminal eingeben zu müssen. Daher ist eine übersichtliche Liste häufig verwendeter Bash-Befehle sehr nützlich.

Ein Terminal ist das schwarze Fenster, in dem Sie nur Text eingeben können. Manchmal wird das Terminal auch als Befehlszeilen-Interface oder in Englisch als *Command-Line-Interface* (CLI) bezeichnet. Die Befehle, die Sie hier eingeben, werden vom Kommandointerpreter mit dem unheilvollen Namen *Bash* ausgeführt.

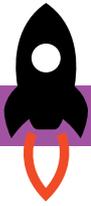
Es gibt viele Befehle und die meisten dieser Befehle akzeptieren alle möglichen Arten von Parametern und Argumenten. Um alles

über einen Befehl herauszufinden, können Sie diesem Befehl `--help` (mit zwei Bindestrichen) hinzufügen, also beispielsweise `rm --help`

Im Folgenden bezieht sich `[path]` auf einen relativen oder absoluten Pfad. Der absolute Pfad beginnt bei `„/“`, also zum Beispiel `„/home/pi“`. ◀

191250-03

<code>pwd</code>	Anzeige des Namens des aktuellen Arbeitsverzeichnisses
<code>ls</code>	Inhalt des aktuellen Verzeichnisses auflisten
<code>ls [path]</code>	Inhalt des angegebenen Verzeichnisses auflisten
<code>ls -l</code>	Listet den Inhalt des aktuellen Verzeichnisses mit zusätzlichen Informationen auf.
<code>ls -a</code>	Listet alle Dateien einschließlich versteckter Dateien auf, die mit <code>„.“</code> beginnen (d.h. versteckte Dotfiles)
<code>cd [path]</code>	Ändert das aktuelle Verzeichnis auf <code>[path]</code>
<code>cd ..</code>	Wechselt in das übergeordnete Verzeichnis (beachten Sie das Leerzeichen zwischen <code>cd</code> und <code>..</code> )
<code>cd /</code>	Wechselt in das Stammverzeichnis (beachten Sie die Leerzeichen zwischen <code>cd</code> und <code>/</code> )
<code>cd ~</code>	Wechselt in das Heimatverzeichnis (bestimmt durch die Umgebungsvariable <code>\$HOME</code> )
<code>mkdir [name]</code>	Erstellt das Verzeichnis <code>[name]</code> im aktuellen Arbeitsverzeichnis
<code>rmdir [name]</code>	Entfernt das leere Verzeichnis <code>[name]</code> aus dem aktuellen Arbeitsverzeichnis
<code>rm [name]</code>	Löscht die angegebene Datei
<code>rm *</code>	Löscht alle Dateien aus dem aktuellen Arbeitsverzeichnis
<code>rm -r *</code>	Löscht alle Dateien und Verzeichnisse aus dem aktuellen Arbeitsverzeichnis
<code>cp [from] [to]</code>	Kopiert eine Datei von der Quelle <code>[from]</code> zum Ziel <code>[to]</code>
<code>cp -r [from] [to]</code>	Kopiert alles einschließlich der Verzeichnisse von der Quelle <code>[from]</code> zum Ziel <code>[to]</code>
<code>mv [from] [to]</code>	Verschiebt eine Datei von der Quelle <code>[from]</code> zum Ziel <code>[to]</code>
<code>mv -r [from] [to]</code>	Verschiebt alles einschließlich der Verzeichnisse von der Quelle <code>[from]</code> zum Ziel <code>[to]</code>
<code>find</code>	Suche nach Dateien, die bestimmten Mustern entsprechen
<code>sudo [command]</code>	Superuser <code>do</code> . Ausführen von <code>[command]</code> mit erhöhten Rechten. Erlaubt es, Dinge zu tun, zu denen Sie normalerweise nicht berechtigt sind. Gängige Beispiele sind: <code>sudo raspi-config</code> Startet das Raspberry Pi Konfigurationstool <code>sudo reboot</code> Sicherer Neustart des Systems <code>sudo shutdown -h now</code> Fährt unmittelbar das System sicher herunter <code>sudo apt-get install [package]</code> Installiert ein (Software-)Paket <code>sudo apt-get update</code> Aktualisiert die Liste der Pakete, ohne etwas zu installieren <code>sudo apt-get upgrade</code> Aktualisiert die installierten Pakete auf die mit <code>apt-get update</code> erhaltenen Versionen <code>sudo chown pi:root [name]</code> Ändert den Besitzer von <code>[name]</code> auf <code>pi</code> und setzt die Gruppe auf <code>root</code> <code>sudo su</code> Werden Sie Superuser für mehr als einen Befehl <code>sudo ku</code> Nicht dokumentiert
<code>cat [name]</code>	Zeigt den Inhalt einer Datei an
<code>head [name]</code>	Zeigt den Anfang einer Datei an
<code>tail [name]</code>	Zeigt das Ende einer Datei an
<code>chmod [who][+,-,=][permissions] [name]</code>	Ändern der Berechtigungen für eine Datei
<code>chmod u+x [name]</code>	Ausführungsrechte für den Besitzer der Datei hinzufügen
<code>chmod 777 [name]</code>	Jedem Benutzer erlauben, die Datei <code>[name]</code> zu lesen, zu schreiben und auszuführen
<code>tar -cvzf [name] [path]</code>	Komprimierte Datei <code>[name]</code> aus dem Inhalt in <code>[path]</code> erstellen
<code>tar -xvzf [name]</code>	Extrahiert den Inhalt einer komprimierten Datei
<code>wget [uri]</code>	Lädt eine Datei aus dem Internet herunter
<code>man [command]</code>	Zeigt die Handbuchseite für einen Befehl an
<code>man man</code>	Zeigt die Handbuchseite des Befehls <code>man</code> an
<code>grep 'string' [name]</code>	Durchsucht eine oder mehrere Dateien nach <code>„string“</code>



# Der vielleicht erfolgreichste Start-up-Beschleuniger Europas?

## HighTechXL, Eindhoven, Niederlande

Von **Terry Boyd**, HighTechXL

HighTechXL ist ein High-Tech-Unternehmen, das von Philips, der Technischen Universität Eindhoven und anderen großen Unternehmen unterstützt wird. Um Teams für innovative Technologien des CERN und anderer Forschungseinrichtungen aufzubauen, veranstaltet HighTechXL jedes Jahr zwei „FasTrackathons“, die von talentierten Ingenieuren über Geschäftsführer bis zu Marketingexperten besucht werden. In nur fünf Jahren ist HighTechXL eine der profiliertesten Venture-Building-Plattformen Europas geworden.

HighTechXL ist in fast jeder Hinsicht nicht nur ein Erfolg, sondern ein wirtschaftlicher Entwicklungsmotor für Eindhoven und die niederländische Provinz Brabant: mindestens 2.000 neue Arbeitsplätze in den Niederlanden und anderen Ländern, mehr als 60 erfolgreiche Unternehmen in den Technologiebranchen der nächsten Generation, die viele neue Technologien auf den Markt gebracht haben. In nur fünf Jahren ist HighTechXL [1] eine der profiliertesten Venture-Building-Plattformen Europas geworden. Und der 2018 stattfindende Übergang zur Hochtechnologie macht HighTechXL noch erfolgreicher.

Im Jahr 2018 kündigte der Gründer und Geschäftsführer von HighTechXL, Guus Frericks, an, dass der High-Tech-Accelerator zum Deep-Tech-Venture-Building übergehen wird, das für das Ökosystem von Eindhoven von größerer Bedeutung ist.

Eindhoven hat grundlegende Halbleiterunternehmen wie ASML hervorgebracht, ein Halbleiterunternehmen „hinter den Kulissen“, das die wichtigsten Maschinen herstellt, die von Industriegiganten wie Intel benötigt werden.

Aber auch die ursprünglichen Bemühungen, die bis ins Jahr 2013 zurückreichen, sind eine beispiellose Erfolgsgeschichte. *Y Combinator* und *500 Startups* haben zwar größere Namen und Milliarden-Dollar-Abschlüsse hervorgebracht, aber beide gibt es schon länger ... und sie haben weitaus höhere Ausfallraten von etwa 90 Prozent. Dagegen haben in den sechs Jahren mehr als 60 Prozent der HighTechXL-Unternehmen überlebt. 17 verkaufen ihre Produkte weltweit und sind im *Star Portfolio* und auf dem Weg zur 10x-Rendite.

### Erfolgreiche Start-ups

- Accerion [2], mit Sitz in Venlo, Niederlande, stellt Positionierungstechnologie für mobile Roboter und autonom geführte Fahrzeuge her. Accerion verkauft seine Produkte auf verschiedenen Märkten weltweit und hat kürzlich eine bedeutende *A Round* [3] erhalten, die von Phoenix Contact Innovation Ventures mit Sitz in Deutschland in Zusammenarbeit mit der Wirtschaftsförderungsgesellschaft der niederländischen Provinz Limburg (LIOF) unterstützt wird.
- Amber Mobility [4] mit Sitz in Eindhoven hat ein Mitfahr-Netzwerk für Elektrofahrzeuge geschaffen, das in Zukunft auf autonome Fahrzeuge ausgedehnt werden soll. Amber hat nach den anfänglichen 500.000 € von befreundeten Personen und Institutionen Kapital in Millionenhöhe gesammelt, unter anderem von der Pala Group BV, und expandiert über Eindhoven hinaus mit Hubs in den ganzen Niederlanden.
- Das in Eindhoven ansässige Unternehmen Bambi Medical [5] hat ein drahtloses Gerät zur Überwachung der Vitalparameter von Frühgeborenen entwickelt. Bambi hat mindestens 4 Millionen Euro von privaten Investoren und 2,4 Millionen Euro von Horizon2020 erhalten.
- byFlow [6] stellt 3D-Lebensmitteldrucker für das Gaststätten-

### Über HighTechXL

HighTechXL ist ein High-Tech-Unternehmen, das von den Mitgliedern der Eindhovener Startup-Allianz unterstützt wird, darunter ASML, Philips, NTS-Group, die Technische Universität Eindhoven (TU/e), Brabantse Ontwikkelings Maatschappij (BOM), High Tech Campus, EY, HVG Law und ABN AMRO.

HighTechXL hat durch seine Partnerschaft mit dem niederländischen Forschungszentrum Nikhef (Nationales Institut für subatomare Physik) Zugang zu einigen der fortschrittlichsten Technologien der Welt, einschließlich Vereinbarungen zur Markteinführung. Um Teams rund um diese Technologien des CERN und anderer Forschungseinrichtungen aufzubauen, veranstaltet HighTechXL jedes Jahr zwei FasTrackathons, die talentierte Ingenieure, Physiker bis hin zu Geschäftsführern und Vermarktern anziehen. Bei den FasTrackathons werden die neuen Technologien den Teilnehmern vorgestellt, die sich dann in Teams zusammenschließen, die in ihren Fachgebieten und Interessengebieten tätig sind.



75 Personen nahmen am zweiten HighTechXL-FasTrackathon im Juni teil, bei dem sich Teams um fortschrittliche Technologien von CERN, TNO und Philips bildeten. Ganz links: Corne Rentrop, TNO-Projektleiter auf dem Gebiet der gedruckten und flexiblen Hybrid-Elektronik. Er war vor Ort, um die gedruckten Elektronik-Technologien, die die Teams auf den Markt bringen könnten, im Detail zu besprechen (Foto: HighTechXL).

gewerbe her. Über das in Eindhoven ansässige Unternehmen wurde international in den Medien berichtet, darunter BBC, CBS Morning sowie auf Industrie- und Technik-Websites.

- Die LifeSense Group [7] stellt Sensoren und gesundheitstechnische Wearables her, die zur Erkennung von Inkontinenz bei Frauen, Männern und Kindern eingesetzt werden. Das Unternehmen hat mehrere Millionen an Investitionskapital über seine Aktivitäten in Asien aufgebracht und plant, in den US-Markt zu expandieren.
- Manus VR [8] mit Sitz in Geldrop, Niederlande, hat Virtual-Reality-Handschuhe für VR-Schulungen entwickelt und eine Testumgebung aufgebaut. Zu den Kunden gehören die NASA und mehrere Spielefirmen. Manus mit Sitz in Eindhoven hat 2019 eine 2-Millionen-Euro *A Round* aus dem Eindhoven Venture Fund II aufgebracht.
- Das in Amsterdam ansässige Unternehmen Sustainable [9] entwickelte ein Verfahren zur Kombination von recycelten PET-Abfällen aus Kunststoffen mit Quarz, um den weltweit ersten kreisförmigen Verbundstein für Küchenoberflächen und andere Anwendungen herzustellen. Das Unternehmen hat mehrere Millionen Euro an Kapital für die spätere Phase eingeworben, darunter 2,2 Millionen Euro von Horizon2020. Das Unternehmen hat Pläne, in andere globale Märkte zu expandieren.
- ULU [10] entwickelt IoT-Technologie für Fahrzeuge, einschließlich Cartracker zur Verfolgung der Daten der Lieferflotte. Das Unternehmen hat mehr als 15.000 Abonnenten und Büros in Amsterdam, Ljubljana, London und Shenzhen.

### Allianz mit CERN

Im Juli 2018 verwandelte sich HighTechXL von einem erfolgreichen Hightech-Startup-Accelerator zu einer Deep-Tech-Venture-Maßnahme. Guus Frericks: „Wir sagten: ‚Was würde passieren, wenn wir nur mit Vorschlägen arbeiten würden, bei denen wir als Region etwas bewirken können?‘ Beginnen wir mit High-Tech-Start-ups oder gehen wir einen Schritt weiter

und nehmen bahnbrechende Technologie als Ausgangspunkt?“ Zu diesem Zeitpunkt ging HighTechXL eine historische Allianz mit CERN ein, um die neueste Technologie aus dem größten Teilchenphysiklabor der Welt auf den Markt zu bringen.

Frericks: „Das CERN antwortete: ‚Nun, wenn es eine Region in Europa gibt, in der man in bestimmten Bereichen etwas bewegen kann, dann ist es Eindhoven.‘ Es gab ein hohes Maß an Vertrauen, dass wir mit diesen CERN-Technologien etwas Spektakuläres tun könnten.“

Der Knackpunkt bestand darin, von der Beschleunigung der Teams, die Hightech-Produkte herstellen, zur Beschaffung von Technologie aus Forschungseinrichtungen überzugehen und dann die richtigen Leute für diese Teams zu finden, die die Technologie auf den Markt bringen. Der neue HighTechXL „wird durch das lokale Ökosystem angetrieben“, sagte Frericks. In den vergangenen fünf Jahren habe die Beschleunigung von Hightech-Start-ups dazu beigetragen, „sehr enge Beziehungen in der Region“ zu entwickeln. Das Vertrauen der Hauptakteure - Unternehmer, Unternehmensführer und technische Talente - bedeutete, dass es leichter war, die kollektive technische Macht der Provinz Brabant für den neuen HighTechXL-Ansatz zu mobilisieren. Und so funktioniert es:

### Eingeladen sind: Ingenieure, Physiker und andere Fachleute

Zunächst bezieht HighTechXL Technologien von Forschungseinrichtungen wie CERN [11], TNO und Philips. Dann wird ein *FasTrackathon* veranstaltet, ein umgekehrter Hackathon, bei dem Technologien und potenzielle Anwendungsbereiche präsentiert werden. HighTechXL fördert den FasTrackathon und lädt Ingenieure, Physiker und Fachleute wie Experten für Geschäftsentwicklung, Finanzmanager und Unternehmer dazu ein, sich den Teams anzuschließen.

Einzelpersonen und Teams nehmen an einer halbtägigen Ideensitzung teil, bauen eine Geschäftsmodell auf und stellen ihre Geschäftsideen am Ende des Tages vor.



Beim FasTrackathon war jeder Raum voll von Menschen und Post-It-Notizen, während alle ein Brainstorming über Anwendungsbereiche für fortschrittliche Technologien durchführten (Foto: HighTechXL).



Das Deep-Tech-Venture-Team Incooling macht das, was es am besten kann: Abkühlung während der niederländischen Dogdays im Sommer. Von links nach rechts: Karen Huaracha Magaña, Helena Samodurova, Hossein Beikii und Hejran Mehrtak (Foto: Incooling).

HighTechXL rekrutiert zusätzliche Teammitglieder, die daran interessiert sind, die Technologie auf den Markt zu bringen, und hält PreVProgram-Sitzungen ab, um die Teams weiter zu stärken. Die Teams durchlaufen einen Auswahlprozess, dann beginnen sie mit dem neunmonatigen Programm zur Entwicklung von Geschäftsmodellen.

### Die erste Deep-Tech-Generation

- Incooling [12], die eine neue Technologie zur Kühlung von CPUs und GPUs in Rechenzentren entwickeln, bereiten sich auf die CES 2020 [13] in Las Vegas vor. Die Mitglieder des Incooling-Teams reisten mehrmals nach Taiwan und auch quer durch Europa, um 2019 an Technologie- und Start-up-Veranstaltungen teilzunehmen. Mitglieder des Incooling-Teams, darunter Helena Samodurova, haben Pitch-Competitions in Amsterdam, Berlin und Taipeh gewonnen. Das Team hat mehr als 600.000 Euro von privaten Investoren eingeworben.
- Dynaxion-Teammitglieder [14] waren zweimal als Finalisten des Opioid Detection Challenge in den USA, einer Veranstaltung, die vom Department of Homeland Security, dem U.S. Postal Inspection Service und der U.S. Customs and Border Protection finanziert wird. Im Rahmen dieser Bemühungen

erhielten sie auch 100.000 Dollar. Dynaxion nutzt den kleinen Teilchenbeschleuniger des CERN, um Pakete und Fracht auf atomarer Ebene zu scannen und so den Transport von Drogen und anderen illegalen Materialien zu stoppen.

- Aircision [15] nutzt die Lasertechnologie, um die zuverlässigste Verbindung in der 5G-Kommunikationsmatrix aufzubauen, und ist in Gesprächen mit führenden Vertretern der Telekommunikationsindustrie, um Partnerschaften aufzubauen. Aircision wurde im März 2020 in Paris als Deep-Tech-Pionier auf der Hello Tomorrow ausgezeichnet.

Die nächste Generation von Deep-Tech-Unternehmen gibt es seit April 2019, wobei mehrere vielversprechende Start-ups auf Technologien von TNO und CERN fußen.

„Der Erfolg der ersten Kohorte beweist, dass Eindhoven sich um Deep-Tech dreht“, sagte Frericks. „Aber ausschlaggebend ist, dass das Netzwerk und die lokale Region durch diese Vorschläge in Bewegung gesetzt werden.“ Der Aufbau von Deep-Tech-Unternehmen ist die richtige Idee zur richtigen Zeit am richtigen Ort. „Und natürlich braucht es auch ein bisschen Glück.“

(191194-01)



Die Mitglieder des HighTechXL-Teams stellen sich der neuen Kohorte von Deep-Tech-Unternehmen vor, die das neunmonatige Venture-Building-Programm beginnen.

### Weblinks

[1] [www.hightechxl.com/](http://www.hightechxl.com/)

[2] <https://accerion.tech/>

[3] <https://accerion.tech/news/accerion-closes-series-a-investment-plans-major-expansion/>

[4] <https://driveamber.com/en/>

[5] [www.bambi-medical.com/](http://www.bambi-medical.com/)

[6] [www.3dbyflow.com/](http://www.3dbyflow.com/)

[7] [www.lifesense-group.com/](http://www.lifesense-group.com/)

[8] <https://manus-vr.com/>

[9] [www.sustainable.com/](http://www.sustainable.com/)

[10] <https://driveulu.com/en>

[11] <https://home.cern/>

[12] [www.incooling.com/](http://www.incooling.com/)

[13] <https://www.ces.tech/>

[14] <https://dynaxion.nl/>

[15] [www.aircision.com/](http://www.aircision.com/)



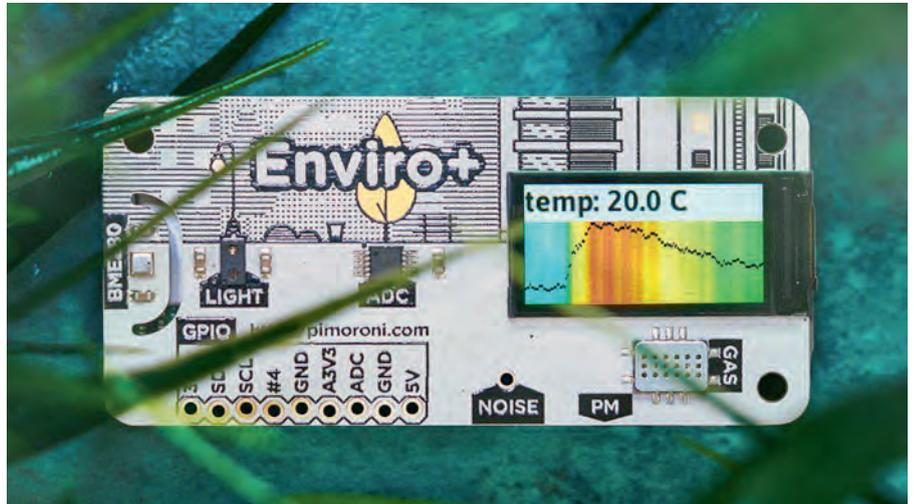
## REVIEW

# RPi-HAT Enviro+

## Umweltdaten messen mit Raspberry Pi und der HAT Enviro+

Von Dr. Thomas Scherer

Für den Selbstbau einer kleinen Messstation für Umweltdaten gibt es das klassische Rezept: Mikrocontroller-Platine plus entsprechende Sensoren. Ein RPi vereinfacht das Design deutlich. Noch schneller geht es mit einer passenden HAT: Enviro+ von Pimoroni.



Heutzutage muss man das Rad wirklich nicht permanent neu erfinden. Man muss keine eigene Mikrocontroller-Schaltung entwickeln, um die wichtigsten Umweltdaten zu erfassen. Man nehme schlicht ein passendes µC-Board samt fertiger Zusatzplatine mit Sensoren, stecke das zusammen und braucht dann „nur“ noch die Software zum Auslesen der Sensoren zu stricken.

mitbringen. Der besondere Vorteil neben der sehr reduzierten Hardware-Bastelei ist: Passende (Python-)Bibliotheken zum Ansprechen dieser Hardware samt Sensoren werden in aller Regel mitgeliefert.

### Enviro+

So ist das auch bei Enviro+ [1], einer kleinen HAT im Format des Raspberry Pi Zero W [2], die aber dank Kompatibilität zum 40-poligen Erweiterungs-Header auf fast alle RPi-Modelle passt. Geliefert bekommt man ein kleines Antistatik-Tütchen mit Aufkleber und nochmal Polsterfolie um die kleine HAT-Platine (Bild 1) – sonst nichts. Im Sinne der Papierersparnis ist nicht einmal ein kleines Zettelchen dabei. Alle Dokumentation steht im Netz. Aber da muss man erst einmal hinkommen.

Das kann man so machen, aber netter wäre es in Zeiten der ubiquitären Smartphones gewesen, der URL zur Dokumentation wäre auch als QR-Code o.ä. aufgedruckt worden – der Barcode enthält jedenfalls leider keinen Link. Schön wäre es außerdem gewesen, der abgedruckte URL wäre durch einen URL-Shortener gelaufen, damit man nicht so viel (fehlerträchtig) abtippen muss. Und am schönsten wäre es gewesen, der URL würde auch zu einer sinnvollen Webseite führen, statt die schöne Meldung „HTTP/1.0 404 Not Found“ auszugeben. Sarkasmus ist an dieser Stelle angebracht, denn so ein Fehler sollte nicht sein.

### Features

Enviro+ ist mit folgender Elektronik bestückt:

- BME280, Sensor für Temperatur, Luftdruck und Luftfeuchte [3]
- LTR-559, Licht- und Näherungssensor [4]
- MICS6814, Analoger Gassensor [5]



Bild 1. Der „Lieferumfang“: Platine in der Tüte.

Will man etwas Luxus wie etwa WLAN etc. haben, drängt sich der Einsatz des am meisten verbreiteten Single-Board-Computers auf: Ein Raspberry Pi bringt neben überflüssig viel Leistung mehr als alles mit, was man dazu brauchen könnte. Von besonderem Vorteil ist, dass es dafür fertige Aufsteck-Platinen in Form passender HATs (Hardware Attached on Top) gibt, die ein Sammelsurium unterschiedlicher Sensoren

- ADS1015, A/D-Konverter [6]
- SPH0645LM4H-B, MEMS-Mikrofon [7]
- 0,96" OLED-Farb-Display mit 160 × 80 Pixel
- Anschluss für optionalen Feinstaub-Luftsensor PMS5003 [8]

### Zusammenstecken

Wie dem auch sei: Zusammengebaut ist die Messstation schnell. Einfach Platinchen aus dem Tütchen holen und auf einen RPi gesteckt – fertig. Fast...

Hat man ein RPi Zero W, dann ergibt sich nach dem Zusammenstecken (mit passenden Abstandshaltern) ein schön kompaktes Modul (**Bild 2**), das sich prima in ein kleines Gehäuse einbauen lässt. Das vorgesehene Original-Gehäuse ist da weniger praktisch, denn es sollte viel Luft an das Sandwich, sonst stimmen die Messungen eher weniger. Und will man auch Helligkeit erfassen, sollte es zudem durchsichtig sein.

Da Enviro+ auf auf einen „normalen“ RPi passt, habe ich ihn auf einen RPi 3B+ gesteckt, der gerade nichts anderes zu tun hatte. Dabei zeigt sich schnell, dass der Deckel des Alugehäuses abgemacht werden muss (**Bild 3**), damit genug Luft und Licht an die Sensoren des Enviro+ kommt.

Doch was man da erblickt, das sind Kühlkörper auf den Chips. Besonders der Größere auf dem SoC macht bei vielen HATS Streß – so auch beim Enviro+. Man muss das kleine Platinchen daher sicherheitshalber mit einem kleinen Stück Tesa isolieren (**Bild 4**), damit es keine Kurzschlüsse gibt, wenn man sie auf den RPi steckt.

Dann endlich kann Enviro+ auf den 40-poligen RPi-Header gesteckt werden (**Bild 5**). Auf diesen Stecker ist man aber nicht beschränkt. Vorne an der Platinenkante sind beschriftete Pads, über die man an ein paar relevante Signale gelangt.

### Dokumentation und Beispiele

Auf der Webseite von Pimoroni findet man die Dokumentation



Bild 2. Zusammen mit einem RPi Zero W ergibt sich ein kompaktes Sandwich.



Bild 3. Bei einem RPi 3 oder 4 muss der Deckel ab, wenn das Board in ein blickdichtes oder gar ein Metallgehäuse eingebaut ist.

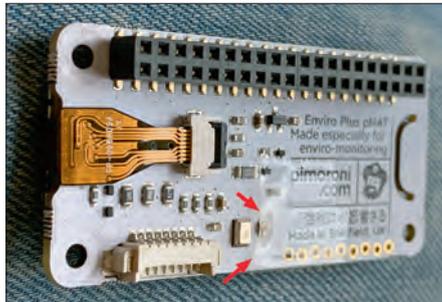


Bild 4. Die Unterseite des Enviro+ wurde mit einem Stückchen Klebestreifen gegen Kurzschlüsse gesichert.



Bild 5. Enviro+ mit Tesa gesichert auf einem RPi 3B+.

doch [9], auch wenn der fehlerhafte Link auf der Tüte das verhindern will. Dafür ist sie sehr gut gemacht und informiert auch grundlegend über die vorhandenen Sensoren sowie darüber, wie deren Werte zu interpretieren sind.

Besonders gut ist, dass dort auch auf Anleitungen verlinkt wird, wie man auf einfache Weise die grundlegende Installation eines RPi vornimmt, was für viele Anfänger sehr hilfreich ist. Anschließend wird gezeigt, wie man mit ein paar Terminal-Befehlen die Zusatz-Software für Enviro+ samt Python-Library und „Examples“ installiert. Das geht fix und schon kann man die Beispiele Ausprobieren. Die **Bilder 6 bis 9** zeigen die Bildschirmausgabe von vier Beispielen.

### Weblinks

- [1] Enviro+: [www.elektor.de/enviro-environmental-monitoring-station-for-rpi](http://www.elektor.de/enviro-environmental-monitoring-station-for-rpi)
- [2] RPi Zero W: [www.elektor.de/raspberry-pi-zero-wh-with-pre-mounted-header](http://www.elektor.de/raspberry-pi-zero-wh-with-pre-mounted-header)
- [3] BME280: [https://ae-bst.resource.bosch.com/media/\\_tech/media/datasheets/BST-BME280-DS002.pdf](https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BME280-DS002.pdf)
- [4] LTR-559: [http://optoelectronics.liteon.com/upload/download/ds86-2013-0003/ltr-559als-01\\_ds\\_v1.pdf](http://optoelectronics.liteon.com/upload/download/ds86-2013-0003/ltr-559als-01_ds_v1.pdf)
- [5] MICS6814: [www.sgxsensortech.com/content/uploads/2015/02/1143\\_Datasheet-MiCS-6814-rev-8.pdf](http://www.sgxsensortech.com/content/uploads/2015/02/1143_Datasheet-MiCS-6814-rev-8.pdf)
- [6] ADS1015: [www.ti.com/lit/ds/symlink/ads1015.pdf](http://www.ti.com/lit/ds/symlink/ads1015.pdf)
- [7] SPH0645LM4H-B: <https://media.digikey.com/pdf/Data%20Sheets/Knowles%20Acoustics%20PDFs/SPH0645LM4H-B.pdf>
- [8] PMS5003: [http://www.aqmd.gov/docs/default-source/aq-spec/resources-page/plantower-pms5003-manual\\_v2-3.pdf](http://www.aqmd.gov/docs/default-source/aq-spec/resources-page/plantower-pms5003-manual_v2-3.pdf)
- [9] Getting Started: <https://learn.pimoroni.com/tutorial/sandyj/getting-started-with-enviro-plus>
- [10] Luftdaten-Station: <https://learn.pimoroni.com/tutorial/sandyj/enviro-plus-and-luftdaten-air-quality-station>

```

pi@raspberrypi: ~/enviroplus-python/examples
Datei Bearbeiten Reiter Hilfe
Pressure: 982.37 hPa
Relative humidity: 15.56 %

2019-12-08 20:40:58.165 INFO    Temperature: 35.80 °C
Pressure: 982.38 hPa
Relative humidity: 15.58 %

2019-12-08 20:40:59.190 INFO    Temperature: 35.80 °C
Pressure: 982.39 hPa
Relative humidity: 15.59 %

2019-12-08 20:41:00.215 INFO    Temperature: 35.80 °C
Pressure: 982.40 hPa
Relative humidity: 15.57 %

2019-12-08 20:41:01.239 INFO    Temperature: 35.79 °C
Pressure: 982.40 hPa
Relative humidity: 15.57 %

2019-12-08 20:41:02.264 INFO    Temperature: 35.79 °C
Pressure: 982.40 hPa
Relative humidity: 15.56 %

```

Bild 6. Ausgabe von Temperatur, Luftdruck und Luftfeuchte, gemessen vom BME. Der SoC des RPI heizt dem Sensor ganz schön ein.

```

pi@raspberrypi: ~/enviroplus-python/examples
Datei Bearbeiten Reiter Hilfe
2019-12-08 20:42:26.028 INFO    Compensated temperature: 20.13 °C
2019-12-08 20:42:27.938 INFO    Compensated temperature: 20.27 °C
2019-12-08 20:42:28.948 INFO    Compensated temperature: 20.41 °C
2019-12-08 20:42:29.958 INFO    Compensated temperature: 20.41 °C
2019-12-08 20:42:30.968 INFO    Compensated temperature: 20.39 °C
2019-12-08 20:42:31.978 INFO    Compensated temperature: 20.39 °C
2019-12-08 20:42:32.988 INFO    Compensated temperature: 20.38 °C
2019-12-08 20:42:33.998 INFO    Compensated temperature: 20.52 °C
2019-12-08 20:42:35.008 INFO    Compensated temperature: 20.67 °C
2019-12-08 20:42:36.018 INFO    Compensated temperature: 20.69 °C
2019-12-08 20:42:37.028 INFO    Compensated temperature: 20.56 °C
2019-12-08 20:42:38.038 INFO    Compensated temperature: 20.55 °C
2019-12-08 20:42:39.048 INFO    Compensated temperature: 20.41 °C
2019-12-08 20:42:40.058 INFO    Compensated temperature: 20.23 °C
2019-12-08 20:42:41.068 INFO    Compensated temperature: 20.68 °C
2019-12-08 20:42:42.078 INFO    Compensated temperature: 20.74 °C
2019-12-08 20:42:43.088 INFO    Compensated temperature: 20.47 °C
2019-12-08 20:42:44.098 INFO    Compensated temperature: 20.46 °C
2019-12-08 20:42:45.108 INFO    Compensated temperature: 20.60 °C
2019-12-08 20:42:46.118 INFO    Compensated temperature: 20.87 °C
2019-12-08 20:42:47.128 INFO    Compensated temperature: 20.87 °C
2019-12-08 20:42:48.138 INFO    Compensated temperature: 20.86 °C
2019-12-08 20:42:49.148 INFO    Compensated temperature: 20.86 °C

```

Bild 7. Ein weiteres Beispiel kompensiert die Abwärme des RPI und gibt realistische Temperaturen aus.

```

pi@raspberrypi: ~/enviroplus-python/examples
Datei Bearbeiten Reiter Hilfe
Proximity: 00.00

2019-12-08 20:44:43.567 INFO    Light: 57.41 Lux
Proximity: 04.00

2019-12-08 20:44:44.589 INFO    Light: 57.41 Lux
Proximity: 00.00

2019-12-08 20:44:45.608 INFO    Light: 59.18 Lux
Proximity: 00.00

2019-12-08 20:44:46.627 INFO    Light: 58.30 Lux
Proximity: 00.00

2019-12-08 20:44:47.646 INFO    Light: 57.41 Lux
Proximity: 00.00

2019-12-08 20:44:48.665 INFO    Light: 58.30 Lux
Proximity: 00.00

2019-12-08 20:44:49.684 INFO    Light: 60.07 Lux
Proximity: 00.00

```

Bild 8. Das Beispiel mit dem Licht- und Näherungssensor gibt immerhin die Beleuchtung in Lux aus – auf Annäherung reagierte die Ausgabe nicht.

```

pi@raspberrypi: ~/enviroplus-python/examples
Datei Bearbeiten Reiter Hilfe
Reducing: 384000.00 Ohms
NH3: 65021.61 Ohms
2019-12-08 20:47:19.483 INFO    Oxidising: 1302.33 Ohms
Reducing: 387165.47 Ohms
NH3: 65980.26 Ohms
2019-12-08 20:47:20.555 INFO    Oxidising: 1302.33 Ohms
Reducing: 387165.47 Ohms
NH3: 67200.00 Ohms
2019-12-08 20:47:21.627 INFO    Oxidising: 1302.33 Ohms
Reducing: 387165.47 Ohms
NH3: 67943.66 Ohms
2019-12-08 20:47:22.699 INFO    Oxidising: 1302.33 Ohms
Reducing: 387165.47 Ohms
NH3: 68696.36 Ohms
2019-12-08 20:47:23.770 INFO    Oxidising: 1355.68 Ohms
Reducing: 390376.81 Ohms
NH3: 69458.25 Ohms
2019-12-08 20:47:24.842 INFO    Oxidising: 1355.68 Ohms
Reducing: 390376.81 Ohms
NH3: 70229.51 Ohms
2019-12-08 20:47:25.914 INFO    Oxidising: 1355.68 Ohms
Reducing: 390376.81 Ohms
NH3: 71272.73 Ohms

```

Bild 9. Der Gassensor besteht faktisch aus drei Sensoren, die unterschiedliche Luftbestandteile erfassen. Die Ausgabe in  $\Omega$  ist ungewöhnlich und müsste noch umgerechnet werden.

Das kleine Display (**Bild 10**) ist besonders dann hilfreich, wenn man die Elektronik an einer geeigneten Stelle platziert und vor Ort ihren Zustand oder die Messwerte prüfen will.

### Fazit

Mit dem Enviro+ kann man wirklich auf einfache Weise eine preiswerte Station zum Erfassen von Umweltdaten realisieren. Dank einer passenden Python-Library ist die Abfrage der Sensoren etc. ziemlich einfach. Die aus ein paar Code-Zeilen bestehenden Beispiele helfen dabei.

Doch das ist noch nicht alles. Auf der Pimoroni-Webseite findet sich auch noch eine komplette Bauanleitung für eine Luftmessstation [10] mit Anbindung an die Cloud „luftdaten.info“. Übrigens kein Wunder, dass die meisten privaten Luftvermesser in D beheimatet sind ;-)

Alles in Allem würde ich sagen: Gut gemacht! ◀

191248-01B



Bild 10. Das Display kann mit beliebigen Inhalten gefüttert werden. Auch hierfür stehen Python-Funktionen bereit.

**IM ELEKTOR-STORE**

→ Enviro+ (Environmental Monitoring Station for RPI)

[www.elektor.de/18975](http://www.elektor.de/18975)

# Aus dem Leben gegriffen

## Bauteile bestellen in der Ukraine und Russland

Von **Ilse Joostens** (Belgien)

Manchmal, wenn ich in nostalgischer Stimmung bin, denke ich wehmütig an die Zeit zurück, als ich auf dem Fahrrad durch die Gegend brauste, um Teile für ein Projekt im Elektronikgeschäft um die Ecke zu kaufen. Früher war alles besser, sagt man manchmal, aber die harte Wirklichkeit war weitaus weniger rosig. Denn wie oft habe ich Hundewetter und Gegenwind getrotzt, nur um doch bitter enttäuscht wieder nach Hause zu kommen? Leider stirbt der Elektronik-Einzelhandel und Online-Bestellungen sind mittlerweile die Regel. Heutzutage bestelle ich (notgedrungen) alles bei den bekannten Großhändlern.

Exotischere Bauteile bekommt man aber meist nicht mehr bei Händlern im Heimatland oder aus den „befeundeten“ Nachbarländern; man muss in der Regel auf das fernere Ausland zurückgreifen. Dasselbe gilt für die besseren „Glaswaren“, denn Röhren sind vor allem in Russland und der Ukraine noch leicht erhältlich. Der Kauf von Teilen im ehemaligen Ostblock hat zwar seinen Charme, aber es gibt auch Fallstricke... Der lateinische Ausdruck *caveat emptor* gilt hier mehr denn je (und ja, ich musste es nachschlagen).

### Russische Verpackungstechniken - eine Einführung

Um 2007 baute ich meine erste Nixie-Uhr mit IN-14-Röhren. Das war damals ein Hobby, aber aus irgendeinem Grund arbeite ich bis heute auch beruflich mit Nixieröhren und anderen Vintage-Bauteilen. Um ehrlich zu sein, die Verkäufe laufen ziemlich gut im Vergleich zu anderen Elektronik-Sachen. Im Laufe der Jahre habe ich durch eBay einige Verkäufer dieser Röhren kennengelernt. Die meisten von ihnen leben in der Ukraine und in Russland, aber ich habe auch Geschäfte mit Verkäufern in Bulgarien, Rumänien und Moldawien gemacht. Postpakete aus diesen Ländern haben in der Regel ein klares und wiedererkennbares Erscheinungsbild und jeder einzelne

Verkäufer hat auch seinen eigenen Stil. Eine kleine Aufzählung von gemeinsamen Merkmalen:

- ein graues und schmutziges allgemeines Aussehen
- die Verwendung von Paketschnur
- Pappschachtel mit grauem oder braunem Packpapier umwickelt oder verbeulte Verpackungen aus dünnem Weichkarton, die stabilisierend vollständig mit braunem Paketklebeband bedeckt sind
- mit einer übermäßigen Anzahl von Briefmarken versehen, die in der Regel auch von der örtlichen Post ordentlich abgestempelt werden (Ergotherapie)
- Schaumstoffabschnitte oder -folie, Styroporplatten, Zeitungspapier oder Plastiktüten als Füllmaterial
- enthält seltsame Notizen mit unverständlichen kyrillischen Buchstaben
- recycelte Kartons von Consumerprodukten, vor allem Damentoilettenartikel, aber auch Spielzeug, Kaffeepads und sogar Farbe.

Manche Verkäufer wickeln die Röhren sorgfältig in Schaumstofffolie ein, andere werfen alles zusammen in eine Plastiktüte. Gelegentlich sieht der Inhalt einer Verpackung eher wie ein



Sauber in Schaumfolie verpackt.



Epochemachende Wiederverwendung alter Kartons.

Abfallbehälter aus, in dem Sie dann nach Ihren Teilen suchen können. Trotz der manchmal verrückten Verpackung gehen neun von zehn Dingen einfach gut. Ab und zu habe ich Glasbruch, ab und zu so schlimm, dass das Paket voller Glasscherben war. Ich habe auch ein paar sehr „angeschmutzte“ Exemplare bekommen, als ob sie jahrelang in einem feuchten Keller gelegen hätten, und einmal komplett unbrauchbare Schrott-Röhren. In der Regel können Sie in einem solchen Fall nach langen Verhandlungen mit dem Verkäufer Ihr Geld zurückbekommen. Natürlich bleibt es ärgerlich, aber in all den Jahren bin ich nie wirklich betrogen worden.

Wenn Sie Röhren in großen Mengen kaufen, sollten Sie voraussetzen, dass wahrscheinlich zehn Prozent der Ware unbrauchbar sein dürfte. Doch das ist typisch für Vintage-Bauteile, die ja oft Jahrzehnte alt sind.

### Bezahlen

Sie bezahlen in der Regel über PayPal. Jetzt, besonders in der Ukraine, gibt es ziemlich viele Einschränkungen für PayPal-Konten, so dass die meisten Verkäufer dort Zwischenhändler wie Western Bid nutzen müssen, um Geld über PayPal zu erhalten. Andere nutzen ausländische Konten, ob von Freunden oder nicht. Seien Sie also nicht beunruhigt, wenn Ihnen ein PayPal-Konto von einem Fremden oder sogar aus Thailand oder Japan mitgeteilt wird.

### Der Zoll, dein Freund und Helfer

Wenn Sie Waren außerhalb der EU kaufen, müssen Sie in der Regel die Mehrwertsteuer bei der Einfuhr und manchmal auch Einfuhrzölle zahlen. Bei Sendungen durch Kurierdienste wie DHL, UPS und Fedex funktioniert die Abwicklung in der Regel ziemlich reibungslos. Kurierdienste sind aber in Osteuropa unerschwinglich und Pakete werden in der Regel per Luftpost verschickt. Das bedeutet, dass die Ware (in meinem Fall) von der belgischen Post zugestellt wird. Es scheint (nur mir?), als würden staatliche Stellen jedes Jahr ein wenig kreativer, wenn es darum geht, solchen Handel durch neue Vorschriften zu behindern. Nicht nur, dass das gesamte Abfertungsverfahren beim Zoll sehr lange dauert (man muss mit drei bis vier Wochen rechnen), sondern es fehlen in der Regel dann auch die notwendigen Unterlagen, die man in seiner Buchhaltung benötigt, um die Einfuhrumsatzsteuer wieder zurück zu erhal-



Scherben bringen Glück!

ten. Die Fehler sind zahlreich, vielfältig und meist zu Ihrem Nachteil. Es hat keinen Sinn, sich zu beschweren, da Ihr Paket schneller an den Absender retourniert als Ihre Angelegenheit bearbeitet wird. Es ist auch möglich, einen Fehler in der Abrechnung im Nachhinein zu korrigieren – allerdings sind dafür dann 85 € Gebühren zu entrichten! Ich will gar nicht darauf eingehen, was es heißt, mehr als 180 Rückstände zu haben, nur weil ein Paket mit Bauteilen hoffnungslos beim Zoll feststeckt. Anrufen und Betteln ist in einem solchen Fall nur kontraproduktiv, denn dann wird Ihr Paket einfach wieder unten in den Stapel gelegt. Merkwürdigerweise habe ich noch nie etwas für Pakete aus China, die per Post kommen, bezahlen müssen. Es scheint, als wolle man vor allem den Handel mit China, das ja bekanntlich kein EU-Land ist, erleichtern wollen.

Regierungen, Europäische Union, wenn ihr das hier lest, tut etwas dagegen! Ich möchte euch mitteilen, dass auch ich kreativ bin und meine Pakete ab sofort von Belgien in die Niederlande versenden möchte. Die holländische Post ist hoffentlich nicht so schwierig... ◀

191266-03



Umfassend gelabelt.



Ein gefundenes Fressen für Philatelisten.



## REVIEW:

# Mikroskop Andonstar AD407 Besser als sein Vorgänger?

Von **Luc Lemmens** (Elektor-Labor)

Schon seit Jahren sind digitale optoelektronische Geräte mit USB-Anschluss auf dem Markt, die als „Mikroskope“ gehandelt werden. Die ersten Modelle waren mechanisch zu instabil, um damit in einem Elektroniklabor vernünftig arbeiten zu können. Hinzu kam, dass der Abstand zwischen Objektiv und Objekt für Arbeiten mit dem heißen LötKolben zu kurz war. Bei solchen Arbeiten, zum Beispiel an Platinen mit SMDs, kam weiterhin eine Lupenleuchte oder Lesebrille zum Einsatz. Andere optische Hilfsmittel waren entweder für die Praxis untauglich, oder sie waren zu teuer.

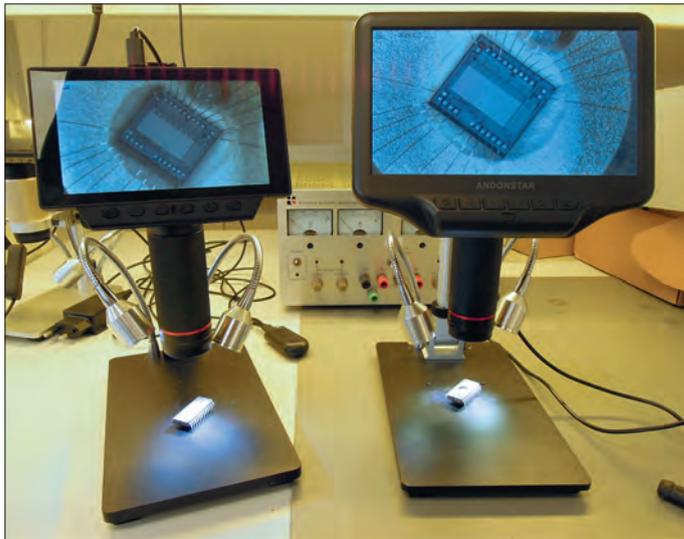


Bild 1. Das USB-Mikroskop Andonstar AD407 (rechts) und sein Vorgänger ADSM302.

Ein wesentlicher Fortschritt kam mit dem USB-Mikroskop Andonstar A1: Endlich ein erschwingliches und standfestes Gerät, das über USB mit einem PC verbunden wird. Das Objekt erscheint stark vergrößert auf dem PC-Bildschirm. Schon bald brachte der Hersteller ein Nachfolgemodell auf den Markt, bei dem das Objekt auf einem integrierten Farb-LC-Display dargestellt wird. Der vorher zum Betrachten notwendige PC wurde entbehrlich, was eine enorme Verbesserung bedeutet. Der Hersteller hat die Produktlinie weiter ausgebaut, so dass vom Elektor-Store bisher vier Versionen angeboten wurden. Das neu erschienene fünfte Modell ist das Andonstar AD407, das wir hier vorstellen. Wir wollen der Frage nachgehen, wie sich dieses Modell von seinem Vorgänger unterscheidet.

Was die Leistung und den Preis betrifft, kommt das neue Modell AD407 seinem Vorgänger ADSM302 am nächsten. In unserem Test haben wir die Eigenschaften und Merkmale des AD407 mit dem ADSM302 verglichen.

### Darstellung und Speicherung

Wenn die Modelle AD407 und ADSM302 wie in **Bild 1** nebeneinander auf dem Tisch stehen, fällt zuerst der Unterschied der Bildschirmformate auf. Beim neuen Modell ist die Diagonale um 5 cm (2 Zoll) größer als beim alten. Für die meisten alltäglichen Feinarbeiten reicht der 5-Zoll-Bildschirm des alten Modells zwar aus, doch im Elektor-Labor haben wir über HDMI einen externen 10-Zoll-Bildschirm angeschlossen. Die Bildschirmdiagonale wird dadurch verdoppelt.

Die zu den Geräten gehörenden Dokumentationen sind leider nicht sehr aufschlussreich, wenn es um technische Daten zur Bildqualität geht. Laut Hersteller ist im neuen AD407 ein 4-Megapixel-Sensor verbaut, während der Sensor des alten ADSM302 aus etwas weniger als 3 Megapixeln besteht.

Laut den technischen Daten bieten beide Mikroskope eine fotografische Auflösung von 12 Megapixeln (4032 · 3024). Diese Auflösung haben auch die Fotos, die vom betrachteten Objekt aufgenommen werden können (JPG-Dateien), um sie auf einem PC wiederzugeben. Die höhere Pixelanzahl des neuen Modells lässt erwarten, dass die Bildqualität höher als beim alten ist. Ein weiterer Unterschied betrifft den digitalen Zoom: Das alte Modell hat einen digitalen Zoom bis 4x, während das neue Modell seine Grenze bei 3x erreicht. In der Detailwiedergabe unterscheiden sich beide Modelle kaum voneinander, das Bild des neuen AD407 erscheint lediglich etwas schärfer. Das Video-Ausgangssignal am HDMI-Anschluss hat beim neuen Modell eine deutlich höhere Qualität als beim alten, sowohl hinsichtlich der Auflösung als auch der Bildrate. Das mag für den beabsichtigten Einsatzbereich nebensächlich erscheinen, doch diese Eigenschaften sind wichtig, wenn Präsentationen oder Demonstrationen auf großflächigen Bildschirmen durchgeführt werden sollen.

Die Bildübertragung über USB zum PC ist beim AD407 nicht mehr möglich, was eigentlich keinen Verlust darstellt. Mit dem großen Farb-LC-Display und dem HDMI-Ausgang ist eine USB-Verbindung für die Echtzeit-Bildwiedergabe verzichtbar geworden. Sie funktionierte bei den älteren Modellen des Andonstar sowieso nicht optimal. Wenn Aufnahmen dokumentiert werden sollen, sind Fotos oder Videos auf einer Micro-SD-

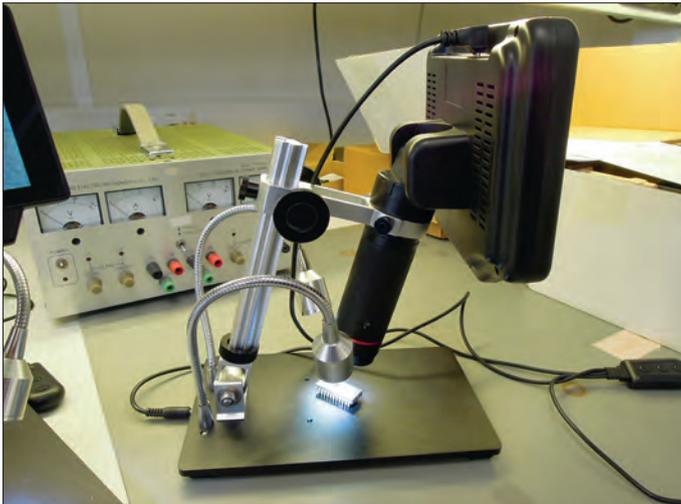


Bild 2. Durch den verstellbaren Stativwinkel können Objekte auch von der Seite betrachtet werden.



Bild 3. Der abgewinkelte USB-Stecker trägt zur Lebensdauer des Kabels bei.

Karte speicherbar, anschließend können die Aufnahmen auf einem PC bearbeitet werden.

### Stative

Beide Mikroskope sind mit robusten und standsicheren Stativen ausgerüstet, die nur wenige Wünsche offen lassen. Das Ausleuchten der Objekte übernehmen bei beiden Modellen zwei LED-Spots an flexiblen, biegsamen Armen. Die LED-Spots verbreiten zwar genügend Licht, doch erfahrungsgemäß sind sie oft im Weg, oder sie nehmen falsche Positionen ein. Wenn sich Hände oder Werkzeuge vor den Spots befinden, wird das Licht abgeschattet, der Betrachter sieht nur noch schemenhaft, was vor sich geht. Manchmal ist das Ausrichten der Spots etwas mühsam, die Arme können zurückspringen, wenn sie losgelassen werden. Eigentlich ist unverständlich, weshalb der Hersteller an dieser Art der Beleuchtung festhält. Ein LED-Ring, angeordnet um das Objektiv, dürfte in vieler Hinsicht effizienter sein.

Die Stative des AD407 und des ADSM302 sind nicht identisch, sie unterscheiden sich in wesentlichen mechanischen Eigenschaften. Das Stativ des Modells ADSM302 kann damit punkten, dass der Objektabstand (Abstand zwischen Objekt und Objektiv) bis zu etwa 12 cm betragen kann. Beim neuen Modell AD407 steht die Objektivunterkante nur bis zu 8 cm über dem Objektträger. Falls das zu einem Problem führt, lässt es sich meistens mit etwas Improvisation lösen. Beim Objektabstand ist das neue Modell dem alten unterlegen. Dafür kann das AD407 mit einer Neuerung aufwarten, die wirklich praktisch ist: Der Standwinkel der Stativsäule lässt sich verstellen (siehe **Bild 2**), so dass die Objekte auf dem Bildschirm auch von der Seite betrachtet werden können. Diese nützliche Option fehlt beim Vorgänger.

### Zubehör

Ebenso wie beim ADSM302 gehören zum Lieferumfang des AD407 ein Netzteil mit Kabel und Klinkenstecker für die Beleuchtung sowie einem Micro-USB-Stecker für die Elektronik. Auch hier wurde ein Detail verbessert: Das USB-Kabel hat jetzt einen rechtwinkligen Stecker, so dass dem Kabel scharfe Knicke erspart bleiben (**Bild 3**). Das Problem des Kabelbruchs trat bei

intensiver Nutzung gelegentlich bei den älteren Modellen auf. Beide Mikroskope werden mit Infrarot-Fernbedienungen geliefert, die nach unserer Erfahrung lediglich beim Aufnehmen von Fotos nützlich sind. Damit lässt sich vermeiden, dass sich das Bild während der Aufnahme bewegt oder verschiebt. Davon abgesehen halten wir eine Fernbedienung bei Mikroskopen dieser Art nicht für einen echten Mehrwert.

Zum AD407 gehört auch ein Inbusschlüssel, mit dem das Stativ montiert wird; laut Anleitung ist er auch zur Montage eines UV-Filters bestimmt. Das UV-Filter fehlte im Zubehör unseres Mikroskops, und um ehrlich zu sein: Auf Anheiß hätten wir nicht gewusst, welchem Zweck dieses Filter dient.

Offen ist noch die Frage, ob der Kauf des neuen Modells AD407 in Betracht gezogen werden sollte, wenn bereits ein früheres Modell zum Inventar gehört. Bei den Typen A1, V160 und ADSM201 ist unsere Antwort ein klares „Ja“, beim ADSM302 fällt unsere Antwort etwas zurückhaltender aus.

Ein Objekt aus der Winkelperspektive betrachten zu können, ist wirklich praktisch, doch dazu ist ein spezielles Stativ nicht unbedingt nötig. Auch das Objekt kann geneigt werden, um diesen Effekt zu erzielen. Mit dem großflächigen LC-Display wird das neue Modell AD407 zu einem eigenständigen, unabhängigen Mikroskop. Wie schon erwähnt, haben wir im Elektor-Labor unserem ADSM302 einen großen Bildschirm zur Seite gestellt. Beim AD407 ist ein externer Bildschirm überflüssig, der Platz auf dem Arbeitstisch wird frei. Das kann schon Grund genug sein, dem AD407 eine klare Präferenz zu geben, falls der Kauf eines neuen Geräts ansteht. Sofern das Budget es zulässt, würden wir ein vorhandenes USB-Mikroskop ADSM302 durch ein neues Modell AD407 ersetzen. ◀

191154-02



**IM ELEKTOR-STORE**

→ Andonstar AD407 HDMI-Digital-Mikroskop  
mit 7"-LCD

[www.elektor.de/19079](http://www.elektor.de/19079)

# Zutritt für Unbefugte verboten!

## Ein Blick ins Allerheiligste aller Elektroniker

Von **Eric Bogers**

Im Juli 2019 organisierte Elektor auf der Labs-Website einen Wettbewerb, um herauszufinden, wer das schönste/chaotischste/interessanteste/merkwürdigste Heimlabor vorweisen kann.

In dieser zweiten Episode der Rubrik präsentieren wir den Gewinner:  
Philippe Demerliac.



Bild 1. Das Siegerfoto.

Philippe (oder Cyrob, wie er von den Berufskollegen genannt wird) schreibt:

Mein Vater arbeitete für eine große Elektronikfirma und überließ mir einige gebrauchte Geräte. Das war der Beginn meiner Liebe zu Messgeräten - ich wollte wissen, wie sie funktionieren, ich wollte sie reparieren oder verbessern und ich wollte selbst neue Geräte entwerfen.

In den 1980er Jahren kaufte ich einen Bauernhof südlich von Nantes, nicht weit von der Atlantikküste, weil meine Frau Pferde züchtet und deshalb viel Platz benötigt.

Viele Jahre lang hatte ich eigentlich nur sehr wenig Ausrüstung - ein Oszilloskop 2225 von Tektronix, ein selbstgebautes Netzteil und einen ebenfalls selbst gebauten Generator sowie ein paar der unverzichtbaren Dinge, die die meisten von uns auch haben.



Bild 2. Am Anfang war mein Labor öde und leer...



Bild 3. Werkbank und Konsole.

Im Jahr 2004 beschloss ich, mein kleines Heimlabor etwas zu erweitern; ich entschied mich für den ungenutzten Schweinestall mit 80 cm dicken Wänden. Ich dämmte den Boden und verkleidete die Wände mit Kunststoffplatten.

Dann kamen die Werkbank mit einer Menge Steckdosen sowie eine Konsole für meine Server und das Kamera-Überwachungssystem hinzu.

Mein Labor hat drei unabhängige Stromkreise: einen nur für die Beleuchtung (damit ich nicht im Dunkeln sitze, wenn eine Sicherung durchbrennt), einen, den ich abschalte, wenn ich das Labor verlasse (damit die Geräte nicht unter Spannung bleiben), und einen, der immer eingeschaltet bleibt (für Alarmer, Server und ähnliches).

Mein Labor ist gut gedämmt und wird mit ein paar elektrischen Heizkörpern beheizt. Ein elektrischer Entfeuchter verhindert, dass die Bude feucht wird.

Nun besitze ich eine Menge an Ausrüstung - viel mehr als ich brauche, aber das liegt daran, dass ich auch ein Sammler bin. Aber es gibt nicht nur „altes Zeug“ in meinem Labor, ich habe auch moderne Geräte wie einen fantastischen Arbiträrge-

nerator mit der Bezeichnung AFG31000.

Es wird Sie nicht überraschen, dass meine größte Sorge wiederum der Platzmangel ist. Ich muss mich beherrschen (oder versuchen, diese dicken Mauern auszubeulen...), denn natürlich muss noch etwas Platz für Bauteile und dergleichen und für mich bleiben.

Weitere Informationen kann ich Ihnen geben, wenn Sie mich unter [info@cyrob.org](mailto:info@cyrob.org) kontaktieren. Und natürlich sind Sie auf meiner Website [1] und meinem YouTube-Kanal [2] herzlich willkommen. ◀

191265-02

#### Weblinks

- [1] Website von Philippe Demerliac: <http://www.cyrob.org>
- [2] YouTube-Kanal von Philippe Demerliac: [www.youtube.com/c/PhilippeDemerliac\\_Cyrob](http://www.youtube.com/c/PhilippeDemerliac_Cyrob)



Bild 4. Server und CCTV-System.



Bild 5. Ein sehr schöner Generator...

# Optischer Tastkopf für Oszilloskope

## Helligkeitsschwankungen von Beleuchtungen messen

Von Alfred Rosenkränzer

Wollen Sie nicht nur elektrische Signale messen, sondern auch optische? Dann brauchen Sie einen optischen Tastkopf, der eine Lichtintensität in für Oszilloskope auswertbare Spannungsschwankungen umsetzt. Dieser Artikel zeigt, wie Sie sich solch einen Tastkopf preiswert selbst bauen können.

Vielleicht erinnern Sie sich an diese Geschichte rund um die Problematik von LED-Lampen, die als Artikel „EMV-Störungen durch LED-Lampen“ in Elektor März/April 2018 erschienen ist? Während meiner Messungen der elektromagnetischen Verträglichkeit für diesen Artikel überprüfte ich auch die Helligkeits-

schwankungen der Lampen mit doppelter Netzfrequenz (100 Hz) und der Schaltfrequenz ihres Schaltnetzteils.

### Flackern

Hierzu baute ich mir damals einen Prototypen eines optischen Tastkopfes auf einer Lochrasterplatine. Einige Zeit

später wurden die Leuchtstofflampen in unserem Großraumbüro gegen passende Retrofit-LED-Lampen ausgetauscht. Die von ihnen abgestrahlten Störpegel waren zwar sehr gering, dafür flackerten sie aber merklich mit der Netzfrequenz oder dem Vielfachen davon.

Jeder konnte dies einfach selbst feststellen, da die LC-Displays unserer Büro-Telefone plötzlich zu flackern schienen. Grund war eine Interferenz des Lampenflackerns mit der Update-Rate der Displays. Die meisten Digitalkameras zeigen ebenfalls eine solche Interferenz zwischen einem Lampenflackern und der Bildwiederholrate des Sensors (und des Bildschirms). Da sich das Lampenflackern auf immer weitere Aufenthaltsbereiche ausbreitete, wollte ich diesem neuen Phänomen moderner Beleuchtung auch da auf den Grund gehen. Dies war Motivation genug, aus dem fliegenden Aufbau des optischen Tastkopfes eine bessere Variante mit Platine, Gehäuse und Kabelanschluss zu machen.

### Die Schaltung

Die Schaltung des Tastkopfes (**Bild 1**) ist supersimpel: Die Umwandlung des Lichts in ein elektrisches Signal übernimmt die Fotodiode D5 (bei mir eine BPW34, doch andere Fabrikate tun es auch) zusammen mit dem Opamp IC3a. Die so erzeugte Spannung wird von einem weiteren Opamp (IC3b) noch einmal zehnfach verstärkt. Der 50-Ω-Widerstand R7

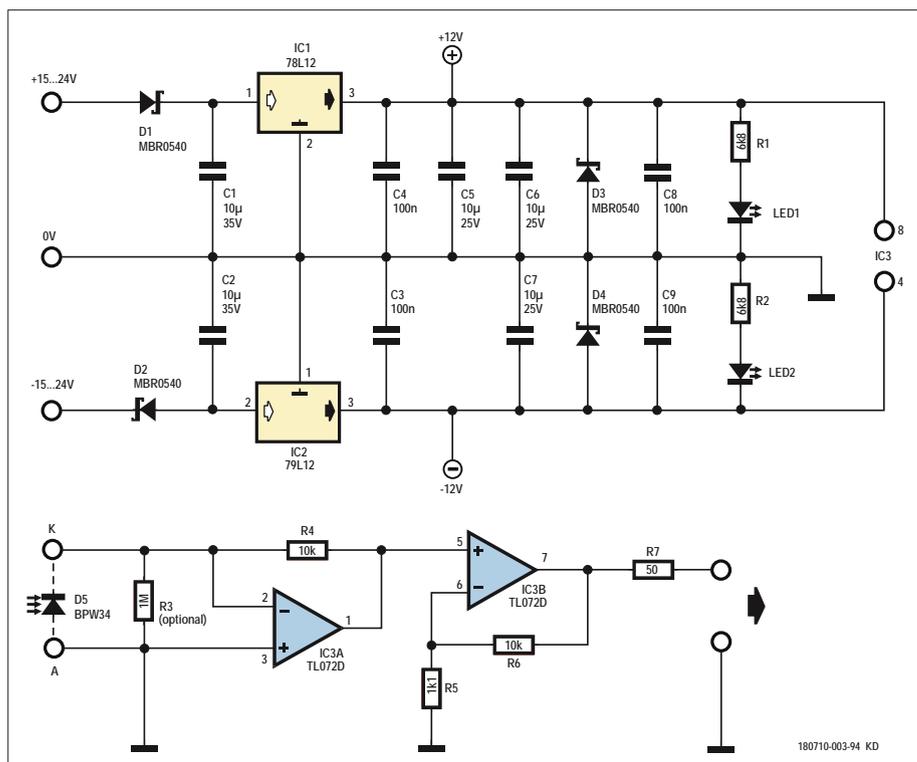
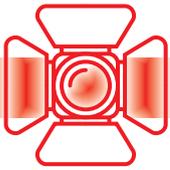


Bild 1. Schaltplan des optischen Tastkopfs.





# Das TABULA-Projekt - ein Update Tangibles mit User-Feedback

Von **Christian Cherek**, Lehrstuhl für Informatik 10 (Medieninformation und Mensch-Computer-Interaktion) an der RWTH Aachen

Das TABULA-Projekt der RWTH Aachen hat sich in den letzten zwei Jahren weiter entwickelt. Die Hardware der Tangibles scheint nun perfekt, um den vielfältigen Aufgaben verschiedener Apps gewachsen zu sein, unter anderem kam eine Rückmeldung per Summer und RGB-LEDs hinzu. Das Hauptaugenmerk liegt ab jetzt in der Entwicklung von (Lern-)Programmen für alle denkbaren Anwendungsbereiche.

In der Septemberausgabe 2017 hatten wir Ihnen ein interessantes, auch von Elektor unterstütztes Forschungsprojekt der Rheinisch-Westfälischen Technischen Hochschule (RWTH) in Aachen vorgestellt. Grundlage des TABULA-Projekts sind anfassbare Objekte, so genannte Tangibles, die auf einer großen kapazitiven Multitouch-Oberfläche abgesetzt und bewegt werden können [1]. Das System soll sich dabei jederzeit im Klaren sein, wo auf der Fläche sich jedes einzelne Tangible befindet, und diese Information an den angeschlossenen Rechner weiterleiten. Dass es sich bei dieser Nutzung der kapazitiven Berührungserkennung nicht um eine abgehobene akademische Spielerei handelt, beweist das lebhaftere Interesse potentieller Anwender: Die Spanne reicht hier von der Informatikausbildung über die Musikproduktion bis hin zur Lernsimulation.

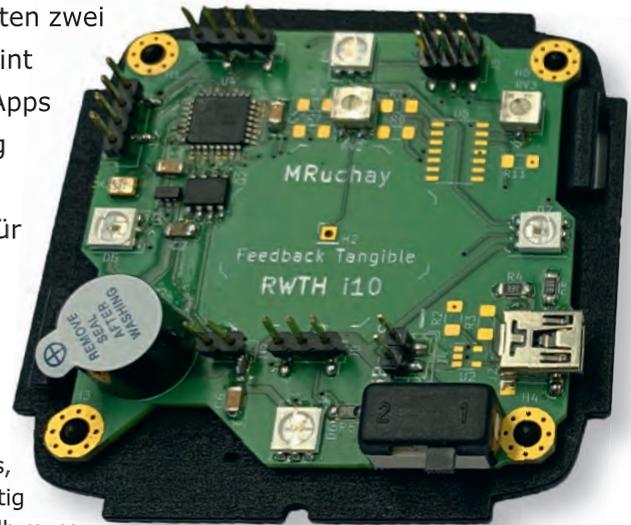
## Zur Erinnerung

In aller Kürze seien noch einmal die wichtigsten Funktionen und Eigenschaften der Tangibles rekapituliert. Die Tangibles besitzen elektrisch leitende, individuell angeordnete „Markerfüße“, die die Objekte auf diese Weise für den Touchtisch unterscheidbar machen. Eine sichere Erkennung ist dabei aber alles andere als trivial, wie im Elektor-Artikel von 2017 [2] ausführlich beschrieben wurde. In bestimmten Anordnungen auf dem

Tisch, oder wenn die Tangibles zu nahe beieinander stehen, reichen die Markerfüße nicht aus, um die Tangibles eindeutig zu unterscheiden. Deshalb muss der Software des Tisches mit Hardwaremaßnahmen unter die Arme gegriffen werden. Dazu gehört, dass ein Tangible ein regelmäßiges Anwesenheitssignal per Bluetooth an das System schickt, wenn es sich auf dem Tisch befindet. Außerdem sendet die Oberfläche des Touchtisches Lichtblitze an die erkannten Tangibles, die dieses Signal per Lichtsensor erfassen. Nur so ist eine individuelle Identifizierung sichergestellt.

## Neuerungen

In den letzten zwei Jahren ist weiter eifrig an der Hard- und der Software gestrickt worden. Eine wichtige Neuerung betrifft ein Feedback für die Nutzer. Dies zeigt ein Blick auf die noch nicht ganz fertig bestückte Platine des „neuen“ Tangibles im Titelbild. Wir haben im Projekt festgestellt, dass sich insbesondere introvertierte, unsichere Menschen schneller und länger aus den Lernspielen zurückziehen, wenn sie einen Fehler machen, als offenere, sichere Menschen [3]. Durch verschiedene Kanäle kann das Feedback individuell angepasst werden, sodass jemand, der einen Fehler macht, nicht befürchten muss, sich in aller Öffentlichkeit „lächerlich“ zu machen und sich



hoffentlich schneller wieder am Lernprozess beteiligt.

Um ein variables haptisches Feedback zu ermöglichen, haben wir auf der Platinenoberseite (**Bild 1**) einige LEDs, einen Summer und einen Vibrationsmotor hinzugefügt. Die an den Mitten der vier Platinenkanten auf der Oberseite verteilten vier NeoPixel-RGB-LEDs D5...D8 [4] dienen dem Feedback des Nutzers. Sie können durch Lichtkanäle aus fluoreszierendem Plexiglas, die das Licht streuen, an die Gehäuseoberseite verlängert werden. Wenn gleich vier LEDs verwendet werden, ist die Wahrscheinlichkeit gering, dass man die komplette Rückmeldung mit der Hand verdeckt.

Auch der Buzzer und der Vibrationsmotor dienen der Rückmeldung für den Nutzer. Die drei Feedback-Varianten (und sogar die vier LEDs) können einzeln angesprochen werden. Auf diese Weise lassen sich verschiedene Modi ausprobieren oder je nach Anwendung unterschiedliche Dinge ausgeben. Der Motor und der Buzzer können auch in der Intensität angepasst sowie einfache Muster/Melodien abgespielt werden. So können die Tangibles besser auf sich aufmerksam

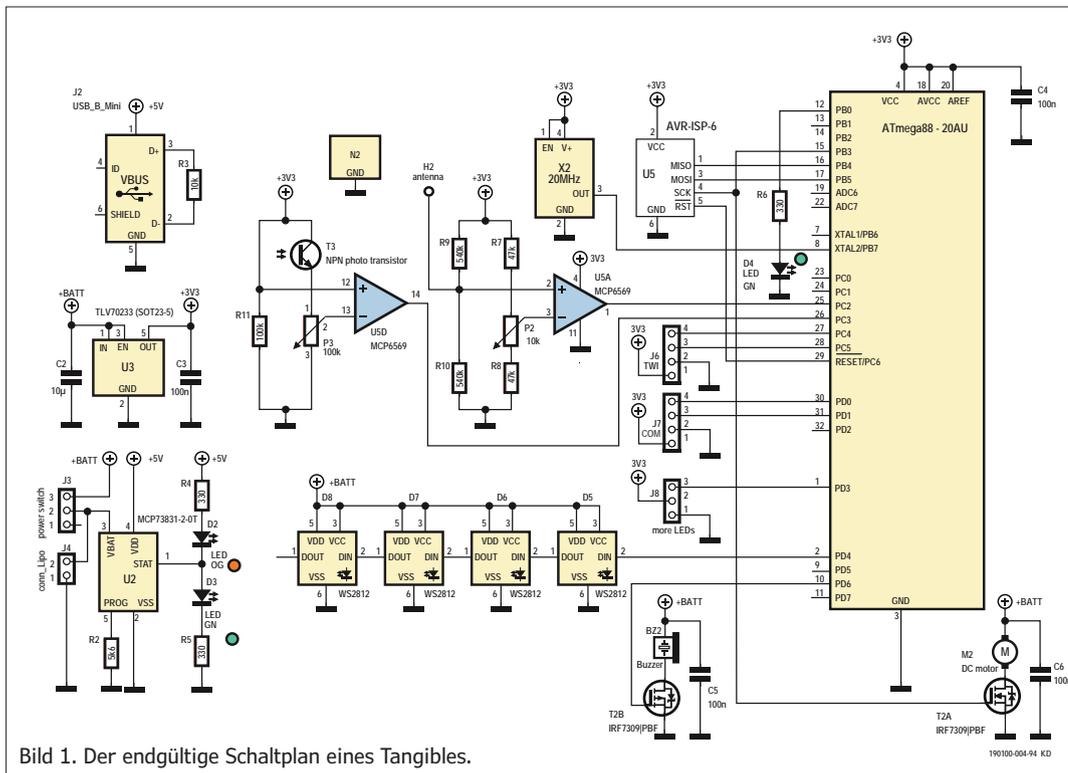


Bild 1. Der endgültige Schaltplan eines Tangibles.

machen, sowohl wenn ein Nutzer sie in der Hand hält, als auch wenn sie einfach auf dem Tisch platziert sind. Dadurch wird auch die Zusammenarbeit der Nutzer am Tisch unterstützt, da den Nutzern schneller auffällt, was gerade bei den Mitstreitern passiert [5].

Auch auf der Unterseite der Platine, zum Touchtisch hin, ist eine LED (D4) hinzugekommen. Sie wird direkt vom Controller gesteuert und leuchtet, wenn das Tangible eingeschaltet ist. Über die neue Micro-USB-Buchse kann das Tangible in dem neuen Gehäuse auch geladen werden. Es gibt eine Ladeelektronik für einen LiPo-Akku und einen Schiebeschalter, mit dem

man zwischen Laden und Betrieb umschalten kann. Zwei weitere LEDs (D2 orange und D3 rot) zeigen den Lade-/Betriebsmodus an. Der klobige Lichtsensor mit seinem „Platinenguckloch“ aus dem ersten Modell ist einer kleinen SMD-Variante auf der Platineunterseite gewichen.

Die neuen Feedback-Funktionen können in den ersten TABULA-Applikationen (Demonstrator 1 und 2) eine wichtige Rolle spielen.

### Erste Anwendungen mit Feedback

In *Dem 1* (eine Basisversion dieser App ist bereits fertig) sollen regulären Aus-

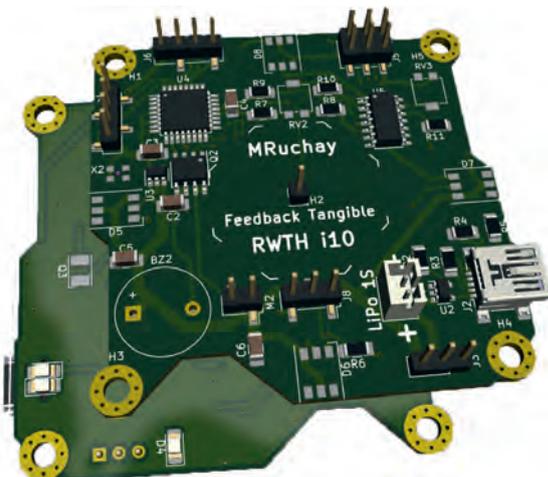
drücken passende Wörter zugeordnet werden. Dabei werden die Tangibles als Werkzeug und Feedback-tool benutzt, falsche oder korrekte Eingaben können jetzt direkt am Tangible angezeigt werden. Die App soll später modifiziert und für den Deutschunterricht in der Unterstufe (Zuordnung von Tierbezeichnungen) sowie in der Oberstufe im Chemieunterricht verwendet werden.

Mit *Dem 2* können aussagenlogische Schaltungen entworfen werden. Dabei werden die Tangibles als Gatter mit den logischen Funktionen NOT, AND, OR, NAND, NOR oder XOR programmiert und so zu digitalen Schaltungen zusammengesetzt. In dieser App zeigen die Tangibles über

die LEDs an, ob der Knoten schaltet oder nicht. Außerdem kann angezeigt werden, ob die Schaltung gerade im Bearbeitungs- oder im Simulationsmodus ist, indem die RGB-LEDs zum Beispiel rot/grün während der Simulation und weiß im Bearbeitungsmodus leuchten.

Auch wenn die Hardware somit fertig ist, gibt es nun noch viel zu tun. Als Nächstes wäre die Finanzierung einer Anschlussforschung wichtig, in der die Anwendungen dann auch einmal wirklich eingesetzt und ausgetestet werden können. ◀

190100-01



### Weblinks

- [1] RWTH Aachen: <https://hci.rwth-aachen.de/TABULA>
- [2] TABULA-Artikel : [www.elektormagazine.de/160123](http://www.elektormagazine.de/160123)
- [3] Ehlenz et al. The lone wolf dies, the pack survives?: Analyzing a Computer Science Learning Application on a Multitouch-Tabletop. Koli Calling ,18. ACM, New York, NY, USA, Article 4, 8 pages: <https://doi.org/10.1145/3279720.3279724>
- [4] NeoPixel-LEDs: [www.world-semi.com/DownloadFile/108](http://www.world-semi.com/DownloadFile/108)
- [5] Cherek et al.. 2018. Tangible Awareness: How Tangibles on Tabletops Influence Awareness of Each Other's Actions. CHI ,18. ACM, New York, NY, USA, Paper 298, 7 pages: <https://doi.org/10.1145/3173574.3173872>

# GEWUSST WIE Berechnung des prospektiven Kurzschlussstroms

## Wahl des richtigen Leitungsschutzschalters

Von Clemens Valens (Elektor-Labor)

Der unbeeinflusste oder prospektive Kurzschlussstrom ist der höchstmögliche Strom, der durch einen kurzgeschlossenen Stromkreis fließen kann. Der Wert wird für die Dimensionierung von Leitungsschutzschaltern und Schmelzsicherungen benötigt, um Schäden an Stromkreisen zu verhindern. Wie funktioniert die Berechnung?

Bevor wir loslegen, wollen wir klarstellen, dass es in diesem Artikel um Elektrotechnik und nicht um Elektronik geht. Wenn also von einem „Stromkreis“ die Rede ist, ist keine Schaltung, sondern ein elektrisches Leitungsnetz gemeint, das Lampen, Schalter, Steckdosen und Maschinen einschließlich aller Verkabelungen miteinander verbindet.

### Was ist der prospektive Kurzschlussstrom?

Der prospektive Kurzschlussstrom ist der maximale Strom, der durch einen kurzgeschlossenen Stromkreis fließen kann. Er folgt, wie jeder Strom, dem Ohmschen Gesetz. Daher bestimmen die Spannung des Stromnetzes und seine Impedanz zusammen den prospektiven Kurzschlussstrom.

### Warum sollte ich das wissen?

Sie benötigen den Wert des prospektiven Kurzschlussstroms, damit ein von Ihnen gewählter Leitungsschutzschalter die elektrische Installation wirksam schützt. Solche Schutzeinrichtungen müssen in der Lage sein, dem Kurzschlussstrom zu widerstehen. Ist nämlich das Schaltvermögen oder die Abschaltleistung des Schutzschalters zu gering, kann der hohe Strom einen Lichtbogen verursachen und/oder den Schutzschalter beschädigen (Bild 1). In beiden Fällen funktioniert die Schutzeinrichtung

nicht mehr richtig, so dass gefährliche Situationen entstehen können.

### Was ist mit dem Auslösestrom?

Das Schaltvermögen eines Leitungsschutzschalters ist nicht dasselbe wie sein Auslöse- oder Nennstrom. Letzterer ist der maximale Strom, den ein Leitungsschutzschalter als sicher erachtet. Das Schaltvermögen hingegen ist der Strom, den das Gerät aushalten kann, ohne beschädigt zu werden. So hat zum Beispiel der 1+N-polige Leitungsschutzschalter (LS, englisch Miniatur Circuit Breaker MCB) vom Typ SN201 L C32-L von ABB einen Nennauslösestrom ( $I_n$ ) von 32 A und einen Nennkurzschlussstrom ( $I_{kmax}$ ) von 4,5 kA (bei 230/400 V<sub>AC</sub>). Der Kurzschlussstrom hängt nämlich nur von der Kapazität des Stromnetzes ab und ist unabhängig von der Stromaufnahme der durch den Leitungsschutzschalter geschützten Last.

### Messung des maximalen Kurzschlussstroms

Sie können den Kurzschlussstrom einer elektrischen Anlage mit einem so genannten PSC-Tester ermitteln. Dies ist ein einfach zu bedienendes Gerät, das den Kurzschlussstrom eines Stromkreises in Ampere (A) beziehungsweise Kiloampere (kA) anzeigt. Obwohl das Drücken der „Test“-Taste normalerweise

**Tabelle 1. Eigenschaften der Schleifenwiderstandsmessung des PeakTech 2715**

Bereich	Auflösung	Prüfdauer	max. Fehler bei Vollausschlag
20 Ω	0,01 Ω	25 A / 20 ms	±2% FS ±5 d
200 Ω	0,1 Ω	2,3 A / 40 ms	±2% FS ±5 d
2000 Ω	1 Ω	15 mA / 280 ms	±2% FS ±5 d

**Tabelle 2. Prospektiver Kurzschlussstrom, gemessen vom PeakTech 2715**

Bereich	Auflösung	Prüfdauer	max. Fehler bei Vollausschlag
200 A	0,1 A	2,3 A / 40 ms	±2% FS ±5 d
2 kA	1 A	25 A / 20 ms	±2% FS ±5 d
20 kA	10 A	25 A / 20 ms	±2% FS ±5 d

ausreicht, um einen Wert zu erhalten, erfordert der korrekte Anschluss des Geräts an den Stromkreis Fachwissen darüber, was Sie zu messen versuchen.

Im Schaltschrank oder an der Verteilerschalttafel wird der PSC-Tester normalerweise zwischen Außenleiter (L) und Neutralleiter (N) angeschlossen, bei einer Steckdose zwischen Außenleiter (L) und Schutzleiter (PE). Verwenden Sie auf jeden Fall das sichere vergossene Prüfkabel, das bei solchen Geräten zum Lieferumfang gehört. Vergewissern Sie sich stets, dass der Tester für die richtige Messaufgabe eingestellt ist, bevor Sie die „Test“-Taste drücken!

### Was macht ein PSC-Tester?

Ein PSC-Tester misst zunächst die Leerlaufspannung an den Klemmen ( $V_S$ , siehe **Bild 2**). Dann legt er kurzzeitig eine kleine Last an, um einen Strom mit bekanntem Wert durch den Stromkreis fließen zu lassen ( $I_T$ , **Bild 3**). Sobald dieser Prüfstrom fließt, misst das Gerät die Spannung  $V$  an den Klemmen. Aufgrund der Impedanz ( $Z_{LN}$ ) des Stromkreises wird die gemessene Spannung  $V$  etwas niedriger als  $V_S$  sein. Diese Impedanz wird berechnet aus:

$$Z_{LN} = (V_S - V) / I_T \text{ [}\Omega\text{]}$$

Unter der Annahme, dass  $Z_{LN}$  konstant ist, berechnet der Tester den prospektiven Kurzschlussstrom als:  $V_S/Z_{LN}$ .

Die Höhe des Prüfstroms hängt vom gewählten Messbereich ab und reicht zum Beispiel von 2 A bis zu 25 A oder mehr. Die Dauer der Messung variiert ebenfalls mit dem Bereich und liegt in der Regel in der Größenordnung von einigen zehn Millisekunden. **Tabelle 1** und **Tabelle 2** zeigen einige „echte“ Messparameter eines realen PSC-Testers.

### Was ist der Schleifen-test?

PSC-Prüfgeräte können auch die Erdschlusschleifenimpedanz  $Z_S$  oder  $Z_E$  oder den prospektiven Fehlerstrom oder Erdschlussstrom messen, der im Falle eines Kurzschlusses vom Außenleiter L zum Schutzleiter PE fließt. Dieser Strom ist von der Impedanz des Stromkreises abhängig (**Bild 4**). Eine niedrige Impedanz führt zu einem hohen Erdschlussstrom, so dass ein Leitungsschutzschalter schnell auslöst. Eine niedrige Impedanz

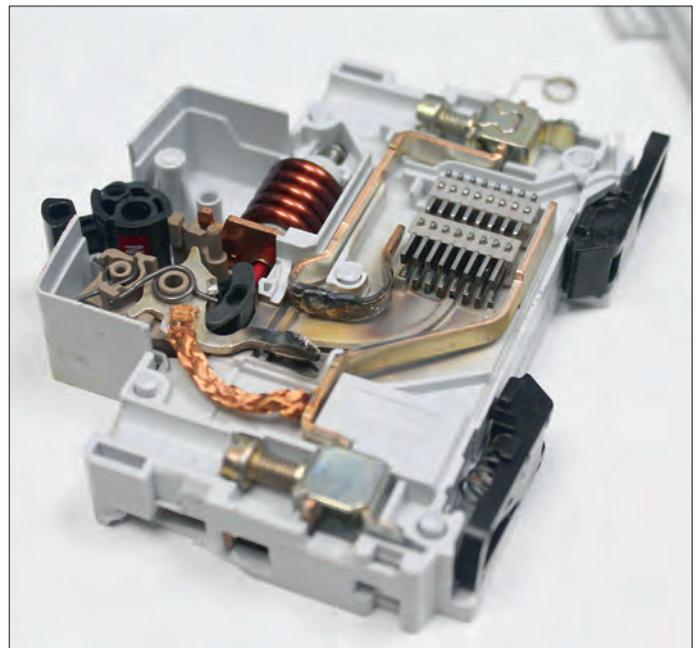


Bild 1. Das kann einem Leitungsschutzschalter (MCB) passieren, wenn er dem Kurzschlussstrom nicht gewachsen ist.

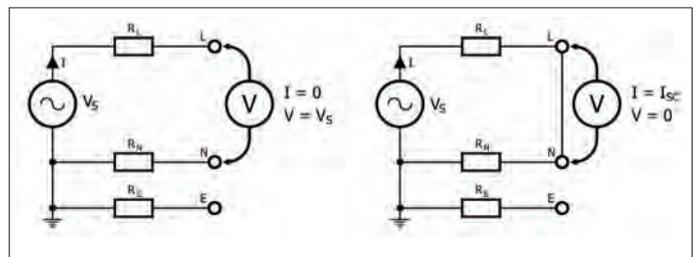


Bild 2. Messung der Leerlaufspannung (links) und des Stroms im kurzgeschlossenen Stromkreis (rechts).

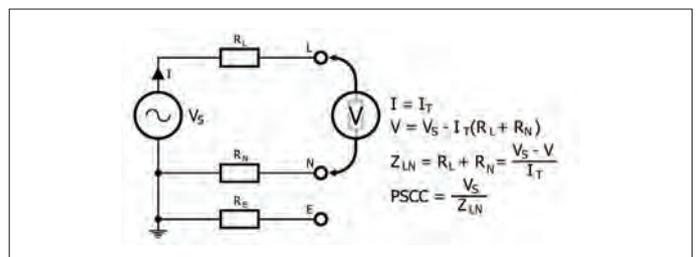


Bild 3. Der PSC-Tester sorgt für einen Prüfstrom in bekannter Höhe, um die Impedanz des Stromkreises zu berechnen.

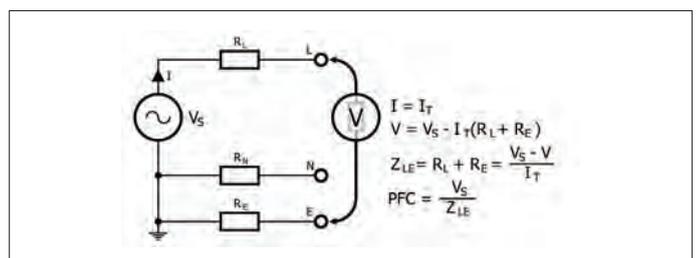


Bild 4. Berechnung der prospektiven Erdschlussimpedanz  $Z_S$  (oder  $Z_E$ ) beziehungsweise des Erdschlussstroms P(EF)C.

sorgt auch dafür, dass der Potentialunterschied zwischen dem PE-Leiter an der Steckdose und der Erde (auf PEN-Potential) besonders klein ist.

Bei der Messung an einer Steckdose wird die Erdschlusschleifenimpedanz als  $Z_s$  bezeichnet, bei der Messung an der Einspeisung der Versorgung wird sie  $Z_e$  genannt. So ist:

$$Z_s = Z_e + R_L + R_E [\Omega]$$

mit  $R_L$  und  $R_E$  als die jeweiligen Impedanzen der L- und PE-Leiter zwischen dem Einspeisepunkt und der Steckdose.

Beachten Sie, dass selbst bei einem geringen Prüfstrom ein Fehlerstromschutzschalter (FI-Schalter) auslösen kann, wenn er empfindlich ist oder wenn ein Leck im zu prüfenden Stromkreis vorhanden ist. Um dies zu vermeiden, können Sie den FI-Schutzschalter vorübergehend überbrücken (aber vergessen Sie nicht, seine Anschlüsse wieder herzustellen, wenn Ihre Messungen beendet sind!).

### Wie verläuft ein P(E)FC-Test?

Die Erdschleifenimpedanz wird auf die gleiche Weise wie die PSCC gemessen, aber der Prüfstrom ist, wiederum abhängig vom gewählten Bereich, viel kleiner, im zweistelligen Milliampere-Bereich. Außerdem wird die Messung zwischen dem Außenleiter L und dem Schutzleiter PE statt zwischen Außenleiter L und dem Neutralleiter N durchgeführt. Den PEFC erhält man durch Berechnung von  $V_s/Z_{LE}$ .



IM ELEKTOR-STORE

→ Peaktech 2715 Digital Loop / PSC Tester  
[www.elektor.de/peaktech-2715-digital-loop-psc-tester](http://www.elektor.de/peaktech-2715-digital-loop-psc-tester)

### Anmerkungen

Je nach Verdrahtung kann es vorkommen, dass sich die Messwerte von PSCC und P(E)FC unterscheiden. Ist dies der Fall, verwenden Sie den höheren Wert bei der Auswahl eines Leitungsschutzschalters.

**Hinweis: Das Arbeiten an elektrischen Anlagen, die unter Spannung stehen, ist (lebens-)gefährlich! Trennen Sie die elektrische Anlage stets vom Leitungsnetz, bevor Sie daran arbeiten!**

191160-03

### Weblink

[1] Video: Ausgepackt - PeakTech 2715 Loop & PSC Tester: <https://youtu.be/9dBhz3acowc>

## Netze und Leitungen

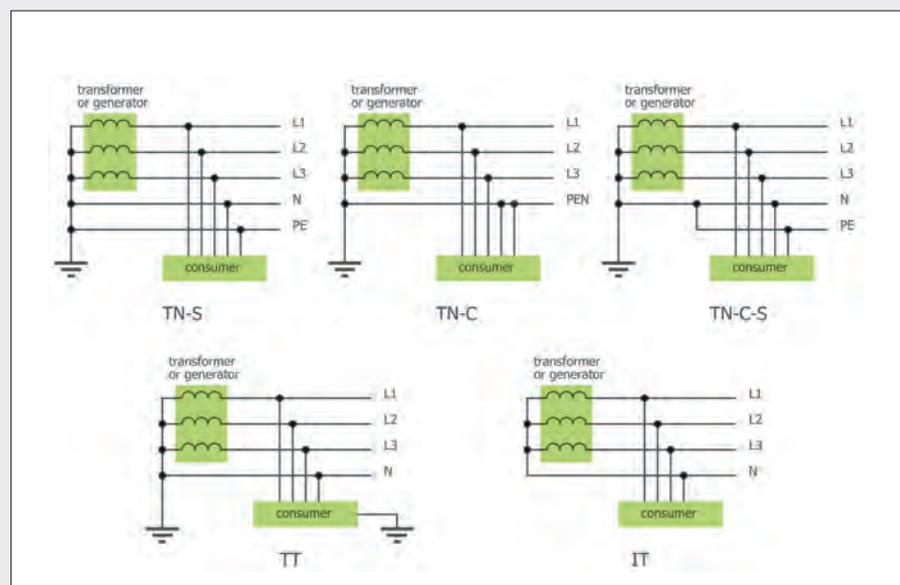
Bei der Bezeichnung elektrischer Leiter in Haushalt und Industriellen Anlagen hat man es mit einer verwirrenden Vielzahl von (aus dem Französischen und Englisch stammenden) Begriffen und Abkürzungen zu tun, die selbst vielen gestandenen Elektrikern nicht geläufig sind und dennoch korrekt angewendet werden sollten.

- TN terre neutre Erde neutral
- -S séparé getrennt
- -C combiné kombiniert
- IT isolé terre Erde isoliert
- TT terre terre Erde Erde
- L ligne Leitung, Linie
- N neutre neutral
- PE protective earth Schutzerde

In Europa sind in privaten und industriellen Niederspannungsanlagen TN-Netze üblich. Ein solches Netz besitzt immer drei Außenleiter (L1, L2, L3) und einen Neutralleiter N, der in Deutschland niederohmig mit der Erde verbunden sein muss. (Nur) in solchen Fällen kann das Netz als **TN-C-Netz** ausgeführt werden, bei dem Neutralleiter N und Schutzleiter PE in einem einzigen Leiter PEN kombiniert sind. Sicherer als diese „Billiglösung“ ist ein **TN-S-Netz** mit getrennten Neutralleiter

N und Schutzleiter PE. Die Verbindung der beiden Leiter erfolgt üblicherweise auf der Generatorseite des Sicherungskastens in Form eines speziellen Erdspießes, der für die Anbindung ans Erdreich sorgt. Zur „Verkomplizierung“ der Sachlage können TN-S- und TN-C-Netze zu einem **TN-C-S-Netz** zusammengefasst werden. Diese werden auch als PME- (Pro-

protective Multiple Grounding), MEN- (Multiple Grounded Neutral) oder MGN-Netze (Multi Grounded Neutral) bezeichnet. In einem **TT-Netz** verfügen sowohl der Generator als auch der Verbraucher über eigene Schutzleiteranschlüsse (Erdungsstäbe); ein PE-Leiter ist nicht vorgesehen. Schließlich hat ein **IT-Netz** überhaupt keinen Erdungsanschluss.



# Die TMS1000-Mikrocontroller-Reihe

## Bemerkenswerte Bauteile

Von **Neil Gruending** (Kanada)

Mikrocontroller werden in vielen elektronischen Geräten, die wir täglich benutzen, eingesetzt. Aber der Multimilliarden-Dollar-Markt für Mikrocontroller hatte einen bescheidenen Anfang, als mehrere LSI-Chips zu einem einzigen verdichtet wurden. Schauen wir uns ein paar der ersten kommerziell erhältlichen Mikrocontroller genauer an, nämlich die TMS1000-Serie von Texas Instruments.

Mikroprozessoren kamen in den 1970er Jahren auf den Markt, und zwar als eine neue Möglichkeit, komplexe Funktionen zu implementieren, ohne kundenspezifische Logik-Chips anfertigen zu müssen. Sie waren für die damalige Zeit sehr leistungsfähig, benötigten aber eine Gruppe von recht teuren Peripheriechips wie RAM, nichtflüchtiger Speicher und mehr. Texas Instruments brauchte eine viel kostengünstigere Lösung für einen neuen Rechner, so dass im Jahr 1971 Gary Boone und Michael Cochran den TMS1802 entwarfen. Es war ein Durchbruch zu dieser Zeit, da er die gesamte Logik, die für einen Vierfunktionsrechner notwendig war, in einem einzigen Chip enthielt, mit Ausnahme der Tastatur und des Displays. Der TMS1802 hat all das mit 3000 Bit Programmspeicher und 128 Bit Random-Access-Speicher erreicht.

Später, 1974, brachte Texas Instruments den ersten Mikrocontroller der TMS1000-Serie auf den Markt, der 1024×8 Bit ROM, 64×4 Bit RAM, eine CPU (den „Mikroprozessor“) und den Taktgeber in einem einzigen Chip integrierte. Das ROM war nicht modifizierbar, so dass ein ROM-loser TMS1000 verwendet wurde, um den Programmspeicher zu entwickeln, der dann als maskiertes ROM für die Produktion verwendet wurde. Dies hatte zwar hohe einmalige Anschaffungskosten zur Folge, aber die Produktionskosten waren recht niedrig. Alle Datenpfade in den TMS1000-ICs waren 4 Bit breit, mit einem separaten ROM- und RAM-Datenbus, wie er in den heutigen Harvard-Architektur-Mikrocontrollern üblich ist. Außerdem wurde ein 6-Bit-Programmzähler bereitgestellt,



Bild 1. Ein TMS1000-Rechner auf einem Chip (Foto: Christian Bassow, Wikimedia Commons, CC BY-SA 4.0, [https://upload.wikimedia.org/wikipedia/commons/d/d1/TI\\_TMS1000NP\\_1.jpg](https://upload.wikimedia.org/wikipedia/commons/d/d1/TI_TMS1000NP_1.jpg))

der ein ROM bis maximal 2048×8 Bit unterstützt. Die CPU unterstützte keine Interrupts, aber es wurde ein Register zur Speicherung des Programmzählers und des Carry-Flags für 1-Level tiefe Unterprogramme bereitgestellt.

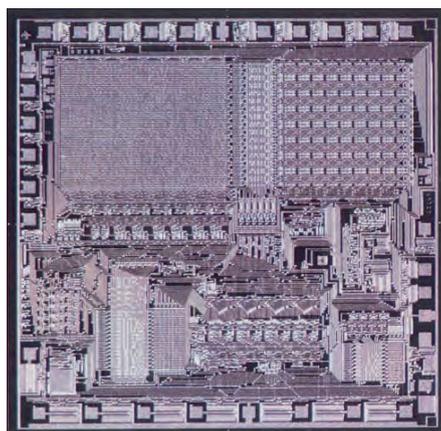


Bild 3. Das Innenleben des TMS1000 (Quelle: *State of the Art*, von Stan Augarten, <http://smithsonianchips.si.edu/augarten/i38.htm>).

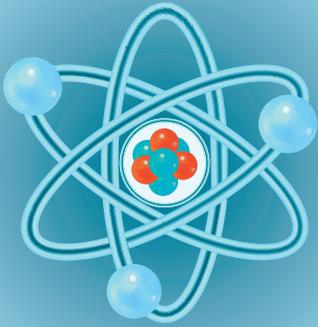


Bild 2. Der TMS1100 in einem Parker Brothers Merlin Game (Foto: Binarysequence - eigene Arbeit, Wikimedia Commons, CC BY-SA 4.0, <https://upload.wikimedia.org/wikipedia/commons/a/ad/Tms1100-merlin.png>).

Das klingt recht eingeschränkt, aber die TMS1000-Serie war immer als Ein-Chip-System ohne externe Schnittstellen (wie zum Beispiel ein UART) gedacht.

Es wurden etwa 100 Millionen Bauteile der TMS1000-Serie in ungefähr 40 verschiedenen Varianten produziert, die hauptsächlich in Produkten der Unterhaltungselektronik (wie Taschenrechner, Spiele und Geräte) eingesetzt wurden. Heute findet man ein Mitglied der TMS1000-Familie nur noch in museumsreifen klassischen Geräten. Aber doch hat diese TMS1000-Serie den Weg für die modernen Mikrocontroller geebnet, die wir alle kennen und lieben. ◀

190383-B-02



# Aller Anfang... muss nicht schwer sein!

Von **Eric Bogers**

Es gibt durchaus Hobbys, wo es kaum etwas zu berechnen gibt. Wer etwa mit Farbe und Pinsel operiert, muss nicht erst ausrechnen, wie grün das Gras ist (das übrigens beim Nachbarn immer grüner zu sein scheint ;-)). Anders in der Elektronik: Hier sind (genaue) Berechnungen ziemlich wichtig, um verkohlte Bauteile, durchgebrannte Sicherungen und Schlimmeres zu verhindern.

In der vorhergehenden Folge haben wir über Spannung und Strom gesprochen; nun geht es darum, was man damit alles anstellen kann.

## Anleitung

Wir alle wissen, dass manche Materialien (z.B. Metalle) elektrischen Strom gut leiten – andere hingegen leiten Strom kaum bis gar nicht. Im ersten Fall spricht man von Leitern, im Letzteren von Isolatoren. Glas ist ein guter Isolator, und – das mag Sie überraschen – auch Wasser isoliert prima (es muss dann aber destilliert oder besser noch doppelt destilliert, nämlich *aqua bidest* sein).

In der Physik gibt es eine Größe, die diese Materialeigenschaft angibt: die *elektrische Leitfähigkeit*. Materialien haben einen messbaren *Leitwert* – bestimmte sogar einen *spezifischen Leitwert*.

## Leitwert

Es besteht eine schön lineare Beziehung zwischen der Spannung an einem Leiter und dem Strom durch ihn, die sich als einfache Formel präsentiert:

$$G = I/U$$

Mit anderen Worten: Der Quotient aus Strom und Spannung ist der Leitwert. Daher gilt: Je mehr Strom (I) bei einer bestimmten Spannung (U) fließt, desto höher ist der Leitwert (G). Die Größe G hat die Einheit *Siemens* (S).

## Widerstand

In der Praxis ist die Spannung meistens numerisch größer als der Strom, so dass der Leitwert fast immer (viel) kleiner als eins ausfällt. Elektroniker sind auch nur Menschen: Sie rechnen nicht gerne mit winzigen Werten. Deshalb verwenden sie eher den Kehrwert des Leitwerts: den *Widerstand*. Je besser ein Leiter bei gegebener Spannung (U) den elektrischen Strom (I) leitet, desto geringer ist sein Widerstand (R). Die Größe R hat die Einheit *Ohm* ( $\Omega$ ):

$$R = U/I$$

Endlich ist es da, das berühmte *ohmsche Gesetz*, von dem selbst absolute Elektronik-Laien irgendwie, irgendwo und irgendwann schon einmal gehört haben.

## Intermezzo

Ein Widerstand von 1  $\Omega$  ist ein recht kleiner Wert. In der praktischen Elektronik hat man es meistens mit größeren Werten wie etwa 8.200  $\Omega$  etc. zu tun. Ein Farad (F) hingegen ist extrem groß (die Einheit F gehört zur Kapazität – davon in einer der nächsten Folgen mehr). Damit man nicht ständig viele Nullen vor oder nach dem Komma schreiben und aussprechen muss – *der Widerstand hat einen Wert von achttausendzweihundert Ohm* oder noch schlimmer: *der Kondensator hat einen Wert von zweiundzwanzig millionstel Farad* – verwendet man Dezimalpräfixe. Die Wichtigsten werden in der Tabelle vorgestellt.

Man spricht also von einem Widerstand von 8,2 k $\Omega$  und einem Kondensator von 22  $\mu$ F. Manchmal wird die Einheit einfach weggelassen und manchmal sogar lediglich das Dezimalpräfix

Vielfache		
Name	Präfix	Faktor
Tera	T	10 <sup>12</sup>
Giga	G	10 <sup>9</sup>
Mega	M	10 <sup>6</sup>
kilo	k	10 <sup>3</sup>
hekto	h	10 <sup>2</sup>
deka	d	10 <sup>1</sup>
Bruchteile		
Name	Präfix	Faktor
dezi	d	10 <sup>-1</sup>
centi	c	10 <sup>-2</sup>
milli	m	10 <sup>-3</sup>
mikro	$\mu$	10 <sup>-6</sup>
nano	n	10 <sup>-9</sup>
piko	p	10 <sup>-12</sup>
femto	f	10 <sup>-15</sup>

anstelle eines Dezimalkommas geschrieben. Aber Vorsicht: Das tut man nur, wenn keine Missverständnisse drohen! Also etwa bei Stücklisten in Elektor, da ein Widerstand nur die Einheit  $\Omega$  und ein Kondensator nur die Einheit F haben kann. Beispiele:

$$6.800 \Omega = 6,8 \text{ k}\Omega = 6\text{k}8$$

$$0,000.000.004.7 \text{ F} = 4,7 \text{ nF} = 4\text{n}7$$

### Stromkreis

Nun fließt ein elektrischer Strom nicht einfach nur so vor sich hin. Er kommt irgendwo her und geht irgendwo hin. Im einfachsten Fall kommt der Strom von einer Batterie (Pluspol) und fließt zum Minuspol derselben Batterie: ein geschlossener Stromkreis. In einem „offenen“ Stromkreis kann kein Strom fließen!

**Anmerkung:** Technisch sieht man die Sache so, dass der Strom von Plus nach Minus fließt. Doch Elektronen bewegen sich genau in die entgegengesetzte Richtung. Die Konvention stammt aus einer Zeit, da man noch nicht wusste, dass Elektronen bei Stromfluss fließen.

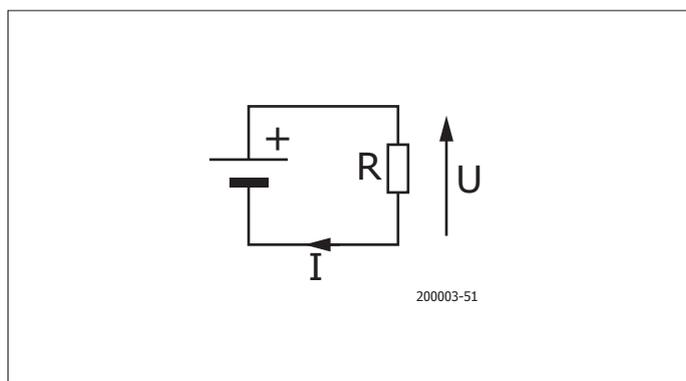


Bild 1. Der einfachste denkbare geschlossene Stromkreis.

Der elementarste, vorstellbare Stromkreis in **Bild 1** besteht aus einer Spannungsquelle wie einer Batterie (links) und einem Widerstand (rechts). Die Leitungen vom Pluspol der Batterie zum oberen Anschluss des Widerstands und vom unteren Anschluss des Widerstands zum Minuspol der Batterie werden als „ideal“ angenommen: Man tut so, als hätten sie keinen Widerstand. Das stimmt natürlich nicht, aber in den meisten Schaltungen ist diese Annahme unproblematisch. Über dem Widerstand liegt die Batteriespannung  $U$ , wie sie durch den Spannungspfeil angezeigt wird. In dieser Schaltung fließt ein Strom  $I$ , der ebenfalls durch einen Pfeil angezeigt wird (wie schon erwähnt fließt er von Plus nach Minus).

Jetzt (endlich!) darf man ein bisschen rechnen. Geht man von einer 9-V-Batterie (kleine Blockbatterie) und einem Widerstand von  $2.200 \Omega$  ( $2\text{k}2$ ) aus, stellt sich die Frage, welcher Strom in der Schaltung fließt.

$$R = U/I \text{ und umgestellt gilt: } I = U/R; \text{ daher rechnet man:}$$

$$I = 9 \text{ V} / 2.200 \Omega = 0,004.09 \text{ A} = 4,09 \text{ mA}$$

Diese „Schaltung“ macht natürlich nicht viel. Es fließt lediglich ein Strom durch einen Widerstand. Aber was macht das mit diesem Widerstand? Er heizt sich auf. So funktioniert ein

elektrischer Heizkörper. In der vorherigen Folge wurde gezeigt, wie man die Leistung berechnet, die im Widerstand in Wärme umgewandelt wird (Techniker sagen: „als Wärme abgeführt“):  $P = U \cdot I$  und umgestellt gilt:  $P = U^2/R$ ; daher rechnet man:  $P = 81 \text{ V}^2 / 2.200 \Omega = 0,036.8 \text{ W} = 36,8 \text{ mW}$

Das ist nicht viel. Der Widerstand heizt nur recht bescheiden. Hätte man einen Widerstand von  $220 \Omega$  verwendet ( $1/10$ ), dann ergäbe sich die zehnfache Verlustleistung von  $368 \text{ mW}$ . Doch in vielen elektronischen Schaltungen verwendet man möglichst kleine (und billige) Widerstände mit einer maximalen Leistung von  $250 \text{ mW}$  (siehe auch die Stücklisten dieser Zeitschrift). Der Widerstand würde also mit mehr belastet als erlaubt. Er würde sehr heiß werden und nach einiger Zeit wohl durchbrennen. Mit Folgen...

Noch ein Beispiel zum Abschluss. Ein 1-kW-Theaterscheinwerfer soll über ein Verlängerungskabel an einer normalen Steckdose betrieben werden. Welchen Strom muss dieses Kabel dauerhaft verkraften, ohne dass es gefährlich wird?



$$P = U \cdot I \text{ und umgestellt gilt: } I = P/U; \text{ daher rechnet man:}$$

$$I = 1000 \text{ W} / 230 \text{ V} = 4,35 \text{ A}$$

Anmerkung: Dass der „Leiter“ bei einer Kabeltrommel doch einen nicht vernachlässigbaren Widerstand hat, merkt man, wenn man hier die Trommel nicht komplett abwickelt. Wegen mangelnder Wärmeabfuhr kann eine aufgewickelte Kabeltrommel innen mit der Zeit sehr heiß werden...

In der nächsten Folge gibt es dann kompliziertere Schaltungen mit Widerständen. ◀

(200003-04)

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

**IM ELEKTOR-STORE**

→ Basiskurs Elektronik (PDF)

[www.elektor.de/basiskurs-elektronik-pdf](http://www.elektor.de/basiskurs-elektronik-pdf)

# Von Entwicklern für Entwickler

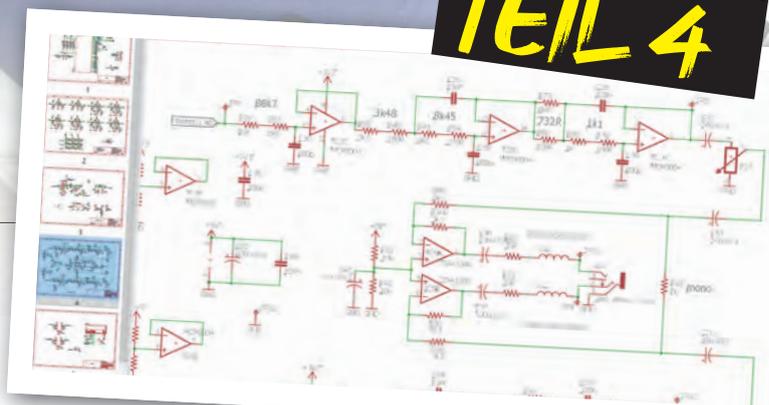
## Tipps & Tricks, Best Practice und andere nützliche Infos

Von Clemens Valens

# VON DER IDEE ZUM PRODUKT

Die Entwicklung unseres neuen Produktes schreitet langsam, aber sicher voran. Jetzt, wo wir einen funktionierenden Proof of Concept (PoC) haben, ist es an der Zeit, mit der Erstellung der Designunterlagen zu beginnen.

## TEIL 4



Es ist nie zu früh, um mit der Dokumentation Ihrer Arbeit zu beginnen. Wenn Sie jetzt etwas Zeit darauf verwenden, ersparen Sie sich später Ärger.

### DOKUMENTIEREN SIE IHRE ARBEIT

Der PoC, den wir im vorherigen Teil entwickelt haben, wurde auf einer oder mehreren Breadboards und/oder Prototyping-Platinen aufgebaut, die mit einer Reihe von Evaluierungs- und Entwicklungskits und -platinen verlötet und verkauft wurden. Es kann sogar sein, dass Bauteile einfach nur an dünnen Drähten hängen. Es sieht wahrscheinlich unordentlich aus, aber der PoC funktioniert, und das war schließlich der Sinn der Übung. Mit dem PoC weiter zu arbeiten, ist unpraktisch und führt zwangsläufig zu losen Kontakten und zum Abfallen von Bauteilen. Deshalb ist es unumgänglich, den PoC in eine stabile Form zu bringen und die Arbeit zu dokumentieren.

### DEN POC AM LEBEN ERHALTEN

Einen Schaltplan aus einem PoC abzuleiten ist eine gefährliche Übung, insbesondere wenn der PoC komplex ist und/oder Komponenten enthält, die nicht in den Bibliotheken des für die Aufgabe verwendeten Programms vorhanden sind, mit dem der Schaltplan erfasst werden soll. Es ist zwar wahrscheinlich, dass der PoC auf einem Prototyp-Schaltplan basiert, aber er kann sich seitdem weiterentwickelt haben. In diesem Fall müssen die Änderungen sorgfältig und fehlerfrei in den Schaltplan zurückportiert werden, ohne mit Murphys Gesetzen in Konflikt zu geraten. So ist es zum Beispiel schnell geschehen, sich bei einem Bauteilgehäuse mit feinem Rastermaß zu verzählen oder einen Tippfehler zu machen. Seien Sie vorsichtig, wenn Sie Verbindungen lösen oder Teile entfernen, um eine bessere Sicht auf das Geschehen zu erhalten. Achten Sie darauf, dass Sie den PoC genauso wiederherstellen, wie er war, denn der PoC ist Ihr Referenzdesign, bis Sie ein besseres gebaut haben. Bis dahin und wenn möglich auch danach muss er weiter funktionieren.

### KÖNNEN SIE IHREN BAUTEILEN VERTRAUEN?

Bibliotheken von Schaltplanerfassungstools enthalten zwar viele Bauteile, aber oft sind es nicht die Bauteile, die Sie benötigen! Folglich müssen Sie sie selbst erstellen. Eine Suche im Internet mag eine schnelle Lösung sein, aber können Sie „clemo12345“ wirklich vertrauen? Sie können Bauteildateien von einem kommerziellen Schaltplan/Platinenbauteil-Anbieter kaufen (ja, es gibt sie); das ist vielleicht besser, aber es kostet Geld. In jedem Fall sollten Sie jedes Bauteil sorgfältig prüfen, bevor Sie sicher sein können, dass es genau „passt“ und Sie sich darauf verlassen können. Dies gilt auch für Standard-Bibliotheksbauteile, da Bibliotheken Fehler enthalten können. Übrigens, wir haben an dieser Stelle (noch) nicht über Footprints für das Platinendesign gesprochen - dazu kommen wir später.

Sind die Bauteile, die Sie verwenden wollen, wirklich heute und auch morgen verfügbar? Wissen Sie genau, wie sie aussehen? Besorgen Sie sich Muster, damit Sie die Footprints und die Abmessungen überprüfen können.



## KEINE PLATZHALTER VERWENDEN

Ein Fehler, den viele Leute bei der Schaltplanerfassung machen, ist die Verwendung von Bauteilen der Bibliothek als Platzhalter, die die gleiche Funktion oder die gleichen Pins wie gewünschte Bauteil haben. Dies kann zwar den Zeichnungsvorgang beschleunigen, aber es werden sich zwangsläufig Fehler einschleichen. Ein klassischer Fehler ist es, wenn Sie vergessen, den Wert des Platzhalterbauteils nachher auf den richtigen Wert zu setzen. Dies ist bei Bauteilen wie Mikrocontrollern vielleicht offensicht-

lich, aber weniger zum Beispiel bei Steckverbindern. Subtilere Probleme können jedoch auftreten, wenn das Capture-Tool zum Export der BoM (Stückliste) verwendet wird. Platzhalterbauteile haben oft Attribute, die überhaupt nicht dem gewünschten Teil entsprechen. Oder doch? Der Zeitgewinn beim Zeichnen durch die Verwendung von Platzhaltern geht nun wieder verloren, da alle Attribute überprüft werden müssen. Denken Sie daran, dass wir uns später mit den Footprints beschäftigen.

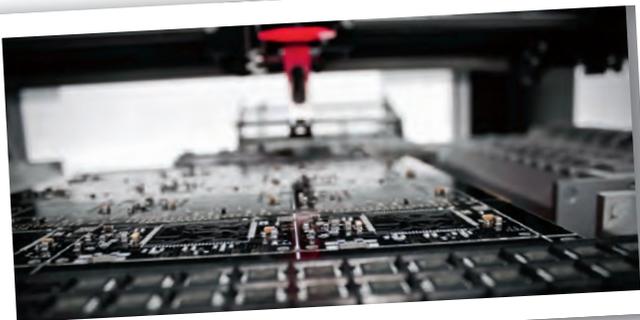
## VOLLSTÄNDIG UND KORREKT

An diesem Punkt des elektronischen Prozesses der Produktentwicklung ist der Schaltplan das wichtigste Dokument. Daher muss er so viele Informationen wie möglich enthalten und korrekt sein. Ein gutes Werkzeug zur Erfassung von Schaltplänen ermöglicht es dem Benutzer, alle möglichen Attribute für Bauteile und die Netze, die sie verbinden, zu spezifizieren. Verwenden

Sie diese Funktionen, damit der Export der BoM ein korrektes und vollständiges Dokument erzeugt. Dasselbe muss auch für die exportierte Netzliste gelten. Platzieren Sie gegebenenfalls Kommentare und Anmerkungen, beispielsweise wenn ein Transistor gekühlt werden muss oder ein Netz einen hohen Strom führt, damit diese Information nicht verloren gehen. Auch Anmerkungen zur Bauteilplatzierung können nützlich sein.

## ENTWURF FÜR DIE PRODUKTION

Neben den Kosten für die Bauteile, aus denen das Produkt besteht, lauert ein weiterer wichtiger Kostenfaktor,



nämlich die Produktionskosten. Kann das, was Ihnen vorschwebt, hergestellt werden? Ist dies zu vertretbaren Kosten der Fall, wie wird das Produkt getestet? Wie werden programmierbare Bauteile programmiert? Wie sieht es mit Hardware-Optionen aus? Eventuell sind Testpunkte oder ein spezieller Teststecker notwendig. Auch ein Testverfahren ist erforderlich. Sie möchten, dass das Testen so einfach und doch so umfassend wie möglich ist. Vielleicht muss der Entwurf dazu angepasst werden? Vielleicht kann die Software die Prüfung erledigen? Wird das Ganze den PoC modifizieren? Markieren Sie Testwerte und -abläufe im Schaltplan und vergessen Sie auch nicht die Bestückungsoptionen (board stuffing).

Da Ihr Entwurf wahrscheinlich eine ähnliche Maschine wie diese durchlaufen muss, stellen Sie besser sicher, dass sie kompatibel sind.

## ENTWERFEN FÜR DIE ZERTIFIZIERUNG

Je nach Anwendung muss Ihr Produkt bestimmte Zertifikate einhalten, bevor es in den Handel gebracht werden kann. CE und FCC sind die bekanntesten, aber es gibt noch viele andere. Der Entwurf muss von Anfang an berücksichtigen, dass Zerti-

fizierungsprüfungen bestanden werden müssen. Wahrscheinlich waren beispielsweise EMV-Maßnahmen wie Schirmung und Filterung nicht Teil des PoCs, müssen aber doch im Schaltplan aufgenommen werden (ohne den PoC zu zerstören).

## VERSIONSKONTROLLE

Es ist einfach nicht möglich, von Anfang an einen perfekten Schaltplan zu erstellen. Die Dinge werden sich ändern und entwickeln, wenn Ihr Wissen und Ihre Erfahrung wachsen. Mehrere Iterationen werden notwendig sein, um alles richtig zu machen. Behalten Sie diese Überarbeitungen im Auge, um zu vermeiden, dass Sie den gleichen Fehler noch einmal machen. Dies ist noch wichtiger, wenn sich ein Entwurf in mehrere Versionen gabelt. Verwenden Sie eindeutige und klare Identifikatoren für Versionen und Iterationen und seien Sie konsistent!

## WIE GEHT ES WEITER?

Die Ergebnisse der oben beschriebenen Entwurfsphase sind - neben einem schönen Dokument für die spätere Verwendung - eine Stückliste (BoM) und eine Netzliste, sowohl vollständig als auch korrekt. Mit diesen Dokumenten in der Hand können Sie mit dem Entwurf einer Platine (PCB) beginnen. ◀

190369-B-03

## Optimales Vorgehen: Das Hybrid-Relais

Triacs und Thyristoren sind großartige Bauteile, um AC-Lasten zu schalten. Sie sind schnell, klein und preiswert. Ein Nachteil ist jedoch ihr relativ hoher Einschaltwiderstand, der zu Leistungsverlusten führt und eine Kühlung notwendig macht, wenn die Last mehr als ein paar Ampere zieht. Auch Relais sind zum Schalten von AC-Lasten sehr gut geeignet. Sie haben einen niedrigen Einschaltwiderstand, sind einfach zu steuern und ihr Steuersignal ist von der Lastversorgung isoliert. Warum nicht ein Relais parallel zu einem Triac verwenden, um eine Last gemeinsam zu schalten? Auf diese Weise können wir das Beste aus beiden Welten haben.

### Das Einschalten

Es beginnt mit dem Einschalten des Triacs, um die Versorgungsspannung an die Last anzulegen. Da der Triac schnell ist, kann er die Last um einen Nulldurchgang der AC-Versorgungsspannung herum ein- und ausschalten und so die elektromagnetischen Störungen (EMI) auf ein Minimum reduzieren. Darüber hinaus kann ein Triac im Phasenanschnitt verwendet werden, was einen sanften Start der Last ermöglicht.

Sobald die Last gestartet wurde, wird auch das Relais eingeschaltet. Da der mechanische Schalter einen viel geringeren On-Widerstand als der Triac hat, fließt der meiste Strom durch diesen Schalter. Der Triac wird nicht mehr benötigt und kann abgeschaltet werden.

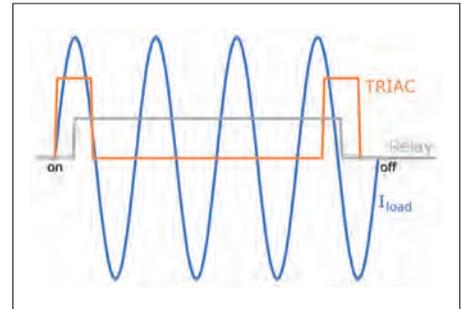
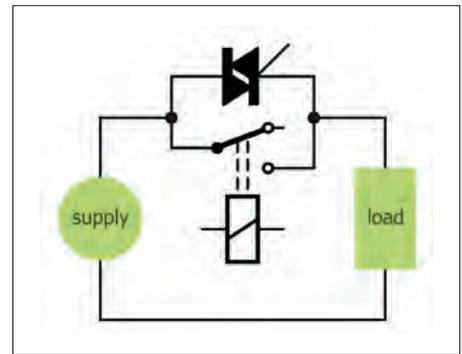
### Ausschalten

Das Ausschalten der Last beginnt mit dem Wiedereinschalten des Triacs, danach wird das Relais ausgeschaltet. Da der Triac in diesem Moment leitet, bleibt die Spannung über den mechanischen Schaltkontakten niedrig. Das Öffnen des Schalters verläuft nun sicher und erzeugt keine Funken und Lichtbögen, so dass die Schaltkontakte nicht abbrennen. Auch hier ist die EMV-Produktion gering, sobald das Relais geöffnet ist, wird der Triac „gelöscht“: Die Last wird abgeschaltet, sobald der Triac nicht mehr leitend ist.

### Vorteile

Ein so konstruiertes und betriebenes Hybridrelais hat viele Vorteile. Die Leistungsverluste sind dank des geringen Einschaltwiderstandes des Relais gering. Das spart auch Kosten und Platz, da der Triac keinen großen Kühlkörper benötigt. Der EMI-Rauschpegel bleibt niedrig, da der Triac kein Kontaktprellen kennt. Außerdem ist der Betrieb schnell und funkenfrei, was das Schalten sicherer und robuster macht.

Der größte Nachteil eines Hybridrelais ist die Anzahl der Teile, die zu seinem Bau benötigt werden. Auch ist die Steuerung komplexer und erfordert in den meisten Fällen einen Mikrocontroller.



**Sie haben eine Idee für ein Elektronikprojekt?  
Oder schon eine fertige Schaltung?**

**Posten Sie Ihr Projekt unter  
[www.elektor-labs.de](http://www.elektor-labs.de)**  
(oder senden Sie es an [redaktion@elektor.de](mailto:redaktion@elektor.de)).

**Auch kleine Projekte sind willkommen!**

**Elektor Labs: Von Lesern für Leser!**

**Elektor Labs: [www.elektor-labs.de](http://www.elektor-labs.de)**  
Design, Build, Share... Electronics!

# Leitungen aufspüren

## Über das Suchen und Finden verdeckt verlegter, schadhafter Leitungen

Von Dr. Thomas Scherer

Ja wo läuft sie denn, die Leitung? Diese Frage stellt sich nicht nur dem Elektriker bei älteren Gebäuden, sondern auch jedem Heimwerker in schöner Regelmäßigkeit – also jedem, der in Haus und Garten selbst gerne Hand anlegt. Manchmal ist das Auffinden verdeckt verlegter Leitungen echt tricky. Daher geht es hier um Methoden, diesen versteckten Kupferdrähten auch dann auf die Spur zu kommen, wenn sie unterbrochen sind.

Vor zwei Jahren fing alles an: Im Frühjahr begann das Gras auf meinem Rasen zu wachsen und der Rasenmäher sollte aus seinem Winterschlaf geweckt, leicht überholt, aufgetankt und dann auch bestimmungsgemäß eingesetzt werden. Doch schon im Jahr zuvor drängte sich angesichts des so schön gleichmäßig per Rasenroboter geschnittenen Nachbar-Rasens die Frage auf: Warum selber mähen, wenn das auch automatisch geht? Gedacht, gekauft, aber noch lange nicht erledigt...

### Leitungen & Rasen

Wenn man den Riesenkarton mit dem neuen Stück Technik in den Händen hält, kommen Probleme auf einen zu, die man leicht und dafür grob unterschätzt. Bei der Recherche, welcher Apparat sich für meinen U-förmigen Garten (mit mehreren Engstellen unter 1 m Breite) eignet, war ich natürlich auch auf die Eigenheiten dieser potentiellen Mähknechte gestoßen. Ich wusste, dass so ein Mähroboter eigentlich ziemlich „unintelligent“ ist und die Fahrten durchs Gras per Zufallssteuerung unternimmt. Wenn er irgendwo anstößt, fährt er zurück, dreht sich (zufällig) ein Stück und probiert es so lange weiter, bis die vorgegebene tägliche Mähzeit beendet



Bild 1. Mein Mähroboter (Bild: Gardena).

ist (oder aber der Akku zur Neige geht). Ein ziemlich dummer Roboter also, denn da ist nichts mit Kamera oder innerer kognitiver Landkarte. Um ihn am Verlassen des Grundstücks zu hindern und nicht zu mähende Flächen auszuschließen, muss man rundum ein Begrenzungskabel verlegen. Und damit er seine Ladestation findet, braucht der Roboter eine extra

Suchleitung - auf dem Rasen oder besser im Boden. Durch diese Leitungen fließt ein gepulster Strom und dank eingebauter Magnetfeldsensoren weiß der Roboter, wo er ist. Bei meinem Roboter (**Bild 1**) ist das Signal sogar individuell und per Passwort verschlüsselt, sodass ein gestohlener Robbi ein unbrauchbarer Robbi ist. Das schreckt immerhin Diebe



Bild 2. Zum Schutz von Ladestation und Mähroboter habe ich kurzerhand eine Hundehütte angepasst. Mein Hund hat schnell gelernt, dass das nicht sein Zuhause ist ;-).

ab, die noch auf drei zählen können. Der einzige zusätzliche Luxus bei meinem Exemplar ist eine Funkanbindung über ein Modem ans Internet. Ich kann also weltweit auf meinem Smartphone nachschauen, was der Roboter gerade macht und ihn etwa bei schwerem Regen in sein Häuschen (**Bild 2**) schicken.

Vor dem Lohn des Zuschauens beim automatischen Mähen haben die Götter allerdings die Verlegearbeiten gesetzt. Bei mir waren das gut 500 m grüne Litze mit ca. 1,5 mm<sup>2</sup> Querschnitt. Wenn man sich die Sache einfach machen will, verlegt man die Leitung auf dem Rasen und tackert sie alle 50 cm mit einer Art Plastik-Hering fest (siehe **Bild 3**). Mit der Zeit verschwindet die Leitung dann im Filz des Rasens. Sagte der Hersteller. Und generell stimmte das auch. Allerdings ist das nur da praktikabel, wo man seine Füße



Bild 4. Wasserdichte Klemme, gefüllt mit Fett. Ich habe mir eine Großpackung davon zugelegt...



Bild 3. Die Leitung wurde hier mit Plastikheringen **auf** dem Rasen verlegt.

sehr selten hinsetzt, sonst baut man sich Stolperfallen. Bei Wegen und Rabatten verlegt man die Leitung besser 10...20 cm tief unter Steinen oder (Beton-)Platten. Und/oder man zieht mühsam Kerben mit einem ungebogenen Spaten durch den Rasen und drückt die Leitung in die Erde. Oder aber man holt sich einen Profi mit einer Leitungsverlegemaschine, der diese schweißtreibende Tätigkeit in kurzer Zeit erledigt. Ich hatte mit allen vier Methoden zu tun, denn just nachdem ich fertig war, zeigten sich die Folgen einer fehlerhaften Beschreibung im Handbuch. Da stand, dass man den Abstand des Kabels zu Rändern und Hindernissen zwischen 15 und 50 cm konfigurieren könne. Im Vertrauen darauf habe ich die Leitungen in einem Abstand von 20 cm zur tatsächlichen Grenze verlegt. Beim anschließenden Konfigurieren meines Roboters merkte ich dann, dass der minimale einstellbare Abstand 25 cm beträgt. Mein Fluchen trug allerdings Früchte: Der Hersteller schickte mir netterweise einen Profi mit Verlegemaschine, der die schon von mir im Boden verlegten 250 m Leitung dann nochmals mit 30 cm Abstand versenkte. Alles war gut und der Mäher mähte hervorragend, wie von mir geplant und erhofft.

#### Kabelbrüche

Mein Glück währte allerdings nicht lange. Wenn man sich so intensiv mit seinem Rasen beschäftigt, dann kümmermt man sich auch. In meinem Fall hieß das „Kampf dem Löwenzahn!“.

Da ich ja wusste, wo die Kabel liegen, war ich der Einzige mit der Lizenz zum Töten von „Unkraut“. Zunächst klappte das ganz gut. Aber dann, irgendwann, passierte es: Beim Ausstechen eines prächtigen Löwenzahns hatte ich mich die entscheidenden Zentimeter vertan und der Roboter blieb einfach stehen. Ich hatte die Leitung unterbrochen. Kein großes Problem, denn ich wusste ja, wo genau ich in gefährlicher Nähe der Leitung gegärtnert hatte. Nach etwas Graben sah ich die durchtrennten Kabelenden. Wenn etwas Luft ist, steckt man die Enden in eine spezielle, mit Fett gefüllte, wasserdichte Klemme (**Bild 4**), quetscht die Klemme zu und alles ist wieder in Ordnung.

So ein Malheur ist mir allerdings nicht nur einmal passiert. Im zweiten Jahr des autonomen Rasenmähens wurde ich scheinbar unvorsichtiger, denn ich hatte am Rand des Gartens über etliche Meter verteilt einige Nichtgras-Pflanzen ausgestochen. Bis ich merkte, dass Robbi hilf- und orientierungslos stehengeblieben war. Ausgerechnet an dem Rand, wo die professionelle Verlegemaschine aktiv gewesen war und die Leitung besonders tief und nicht von mir verlegt im Boden lag. Was also tun? Alle potentiellen Stellen nacheinander ausgraben? Oder technische Hilfsmittel einsetzen?

Als Elektroniker tendiere ich nachvollziehbarerweise zu Technik. Also war zunächst Kaffeepause angesagt. Meine Stirn legte sich in angestrengte Nachdenkfalten. Und dann kam Google zum Einsatz.

## Trial & Error

Mein erster Gedanke war, alle Kabel von der Ladestation (**Bild 5**) im umfunktionierten Hundehäuschen abzumachen und sie über einen „Vorwiderstand“ alias Glühlampe direkt mit Netzspannung zu verbinden. Dann sollten an der defekten Stelle Geräusche der Kategorie „spratzel-zisch“ zu hören sein. Wenn denn der Boden feucht genug war. Es stellte sich raus: Er war es nicht! In aufrechter Position war nichts zu hören, egal welches Kabelende ich mit dem Netz verband. Und näher ran im Sinne von Hinknien und das Ohr Richtung Grashalme zu senken, das war mir dann doch etwas zu riskant, denn ein Ohr ist bekanntlich außen am Denkgehäuse festgemacht und Neuronen sind inkompatibel mit 230 V. [Anmerkung der Redaktion: Wir raten dringend davon ab, Netzspannung als Messmittel einzusetzen - **solche Versuche können tödlich enden!**]

Was dann? Hallensensoren hätte ich gehabt, aber bei einer Unterbrechung fließt kein Strom und wo kein Strom, da kann man auch kein Magnetfeld erfassen. Das ist auch dem Roboter schon aufgefallen ;-). Google meinte, dass solche Rasenroboter bzw. deren Ladestationen pulsierende Signale in die Leitungen injizieren, deren Oberwellen man mit einem Langwellenradio empfangen könne. UKW sollte nicht gut gehen, da feuchtes Erdreich hohe Frequenzen ganz gut abschirmt. Also Langwelle. Doch woher so eine Antiquität nehmen, wenn nicht stehlen? Bei mir kommt das Radio aus dem Internet bzw. der Netzwerkdose. Aber meine Mutter hatte noch solch ein analoges Kofferradio in der Küche. Also versuchte ich es mit dieser Leihgabe. Hierzu darf man nur immer ein Kabelende an die Ladestation anschließen, denn wenn beide Seiten des durchtrennten Kabels mit diesem Signal gespeist werden, findet man die Bruchstelle nie. So ultrapräzise ist eine Ortung per elektromagnetischer Abstrahlung schließlich nicht. Aber langer Rede kurzer Sinn: Es funktioniert!

Man kann deutlich eine Veränderung im Rauschen/Knistern feststellen, wenn man das Radio in die Nähe der Leitung mit Signal bringt. Man stellt dann die Lautstärke so ein, dass sich das Hörbare schon bei kleinen Abstandsänderungen verändert. Das klappt auch bei 20 cm tief vergrabenen Leitungen. Ich brauchte dann nur drei bis vier Testdurchläufe am Boden, um die Stelle zu finden, wo das Signal abrupt aufhört. Ich war mehr als

erleichtert und während des Flickens der schadhafte Stelle schwor ich dankbar, dass ich Google ein Bier ausbebe, wenn ich es mal treffe.

## Andere Fälle

Wer gelegentlich damit zu tun hat, Löcher in Wände zu bohren, um per Dübel und Schrauben ein Bild oder ein Regal aufzuhängen, der wird auch bei Beherzigung der Pi-mal-Daumen-Regel, dass eine Leitung unter Putz stets senkrecht oder waagrecht von einer Steckdose oder einem Schalter abgeht, immer ein mulmiges Gefühl haben. Auch wenn in Deutschland diese Regel seit den 1950er Jahren ziemlich gut befolgt wird, so gibt es a) Ausnahmen und b) auch ältere Gebäude mit „freier Verdrahtung“. Hinzu kommt, dass man in Bädern und anderen Nassräumen nicht nur auf die Strominstallation aufpassen muss. Letztlich muss man also wissen, ob irgendetwas Metallisches in der Wand versteckt ist, bevor man den Bohrhammer ansetzt.

Wenn es nur darum geht, ob in der Nähe der geplanten Bohrung ein Kabel oder ein Rohr verlegt ist, dann eignet sich für diesen Zweck sehr gut einer der in jedem Baumarkt erhältlichen „Metalldetektoren“. Diese Geräte entscheiden nach dem Motto „da oder vermutlich nichts da“, ob man das Bohren riskieren kann. Etwas bessere Geräte wie meines (**Bild 6**) haben verschiedene Betriebsmodi und können angeblich Holz (meiner Erfahrung nach eher schlecht bis gar nicht) bzw. dessen Feuchtigkeit, Metall (egal ob Eisen oder Nichteisen – das klappt halbwegs) sowie elektrische Wechselfelder (klappt auch halbwegs) detektieren. Für die normale Hausinstallation ist so ein Gerät ausreichend und immerhin besser als Nichts.

Neulich allerdings habe ich für Elektor einen „richtigen Metalldetektor“ getestet, den Typ CS-400 von Velleman (**Bild 7**) [1]. Dabei habe ich herausgefunden, dass sich so ein Gerät ganz besonders gut zum Aufspüren verdeckt verlegter Leitungen oder Rohren aus Metall eignet. Sogar für das präzise Lokalisieren von Holzbalken hinter Rigips-Platten ist so ein Schatzsuchgerät sehr gut geeignet. Selbst preiswerte Varianten richtiger Metalldetektoren sind also deutlich genauer und zuverlässiger als die einfachen Detektoren aus dem Baumarkt – allerdings auch teurer und etwas unhandlicher.

Metalldetektoren – egal ob super-empfindlich wie der von Velleman oder billig



Bild 5. Die Ladestation im (geöffneten) Hundehäuschen mit angeschlossenen Kabeln.

und einfach wie die aus dem Baumarkt – eignen sich also durchaus dazu, verstecktes Metall und mehr in Wänden oder in der Erde zu finden. Was damit aber nicht geht, ist Unterbrechungen zu lokalisieren, denn die Trennstellen sind typischer-



Bild 6. Einfacher Metalldetektor alias „Leitungssucher“ aus dem Baumarkt.



Bild 7. Display eines „richtigen“ Metalldetektors: Der Typ CS-400 von Velleman [1].



Bild 8. Dieses einfache Kofferradio mit LW-Bereich habe ich mir für einen lächerlichen Betrag bei eBay „geschossen“. Es dient als universeller Empfänger zum Aufspüren von Unterbrechungen bei versteckten Leitungen.

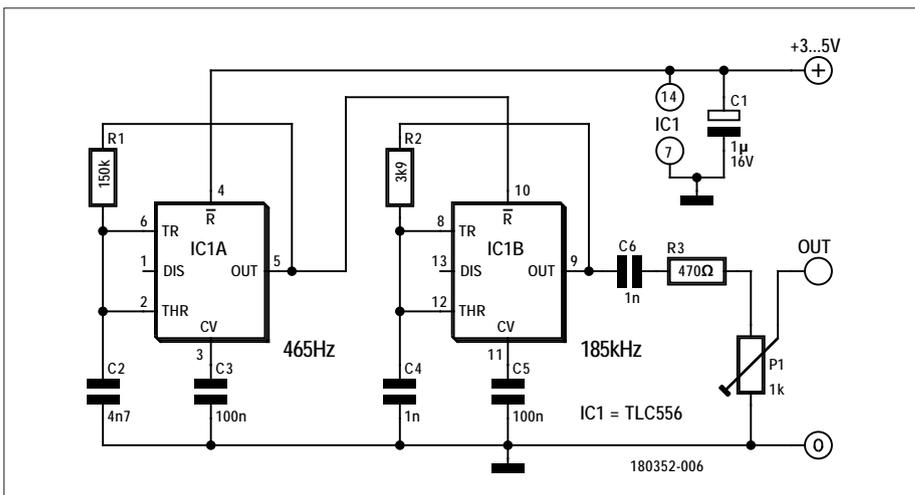


Bild 9. Ein Signalinjektor auf der Basis des CMOS-Doppeltimers 556. Er liefert ein mit 465 Hz amplitudenmoduliertes LW-Signal (185 kHz).

weise so schmal, dass die räumliche Auflösung nicht ausreicht. Also muss hier die Methode mit dem Radio angewendet werden. Doch was, wenn kein HF-Signal oder keines mit schnellen Flanken zur Verfügung steht?

Dann baut man sich einen Generator!

### Für alle Fälle

Durch meine Erfahrung mit den Leitungen im Rasen und die vorhergehenden Überlegungen wuchs der Wunsch nach einer Methode, die (fast) immer und überall funktioniert. Also auch in den Fällen, wo ein Rasenroboter keine steilflankigen Signale anbietet, die mit einem Radio erfassbar sind. Oder wo schlicht ein Stromkabel aus anderen Gründen (wie für eine Lampe) durchs Erdreich verläuft und durchaus einmal Bekanntschaft mit einem Spaten oder einer Motorhacke machen kann. Für alle Fälle, in denen eine Leitungsunterbrechung gefunden werden soll, braucht es also etwas Elektronik, die das erforderliche Signal zur Detektion durch einen passenden Empfänger generiert.

So etwas zu bauen, das ist nicht schwer. Zuvor sollte man sich aber über die Kriterien klar sein. Klar ist, dass höhere Frequenzen durch das Erdreich (oder Mauern) stärker bedämpft werden als niedrigere. Und da es Radios fertig gibt, spart man sich so schon mal den Bau eines Empfängers. Idealerweise sollte man sich also ein Radio mit Langwellenbereich zulegen. Dass in Deutschland wie in einigen anderen europäischen Ländern bald keine rein analogen Radios mehr verkauft werden dürfen [2], ist unproblematisch, denn der integrierte digitale Empfänger neuerer Radios stört ja nicht. Und außerdem gibt es immer noch eBay. Hier habe ich mir mein „eigenes“ Langwellen-Radio (**Bild 8**) für kleines Geld besorgt - und das können Sie auch, wenn sie nicht sowieso noch irgendwo so ein Gerät herumstehen haben. Übrigens muss es nicht zwingend ein Radio mit LW-Bereich sein. Mittelwelle geht in der Regel auch noch, sofern die Flanken des zu detektierenden Signals steil genug sind. Einfach ausprobieren...

### Signalinjektor

Ein amplitudenmoduliertes Signal zum Empfang mit einem LW-Radio ist schnell gebaut. Man könnte dazu einen handelsüblichen kleinen Mikrocontroller nehmen und ein paar Zeilen Code würden ausreichen. Aber wenn wir es schon mit analo-

ger Technik zu tun haben, dann sollte auch eine Analog-Schaltung zum Zuge kommen. Wie wäre es also mit dem IC aller ICs, dem Timer 555?

Geht prinzipiell. Doch wenn man ein amplitudenmoduliertes Signal will, dann braucht es schon zwei Timer-ICs – einen für die NF und einen für die HF. Sowa gibt es in einem Gehäuse: Der Typ 556 ist ein doppelter 555. Es wäre so schön einfach, aber ein normaler 555 hat wie auch sein dualer Kollege 556 den Nachteil, dass er 5 V zur Stromversorgung benötigt und einiges an Strom zieht. Nicht gut für eine Batterieversorgung. Aber glücklicherweise gibt es den 556 in einer CMOS-Variante. Das IC TLC556 funktioniert schon prima ab 2 V und kann demnach sehr gut durch zwei AA- oder AAA-Zellen oder eine Lithium-Knopfzelle des Typs CR3032 versorgt werden.

Aus diesem Grund wurde dieser IC-Typ für die Schaltung des universellen Signalinjektors in **Bild 9** verwendet. Mit R1 und C2 generiert Timer 1 (IC1A) ein symmetrisches Rechtecksignal von 465 Hz, also einen Halbton über dem Kammerton a. Damit wird über den Reset-Eingang von IC2A ein mit 465 Hz amplitudenmoduliertes und durch R2 und C4 bestimmtes 185-kHz-Rechtecksignal erzeugt. Mit P1 kann man die Amplitude des in die Leitung eingespeisten Signals so einstellen, dass dieses gerade noch gut im Radio zu empfangen ist. R3 dient lediglich als Kurzschlusschutz.

Bei den Bauteilwerten ist man nicht genau festgelegt. Die dritte oder fünfte Oberwelle von Timer 2 sollte noch genug Amplitude besitzen, um sie aus der Nähe mit einem Mittelwellenempfänger empfangen zu können. Wer das Pech hat, in der Nähe starker Sender im Bereich von 185 kHz zu wohnen, der kann natürlich mit R2 leicht andere Frequenzen realisieren. Dabei gilt  $f = 0,725 / (R2 \times C4)$ . ◀

180352-01

Anzeige

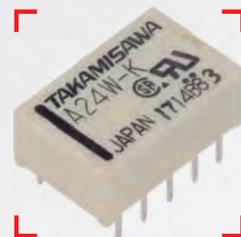
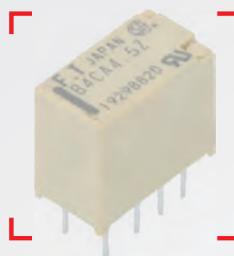
Das gesamte Angebot der Fujitsu Relais bei TME finden Sie hier:

FUJITSU



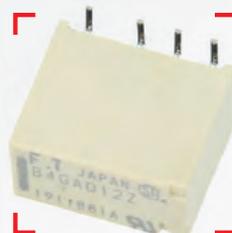
Bei Transfer Multisort Elektronik

Elektromagnetische  
Miniatur-Relais



Signal-Relais  
Serie: FTR-B4 sowie  
NA, RY, SY

Leistungs-Relais  
Serie FTR-F3 mit einer  
Spulenspannung  
von 3V, 5V, 12V und 24V



Automobil-Relais  
Serie: FBR51, FBR52  
FTR-P3 mit einer  
Spulenspannung  
von 10V und 12V



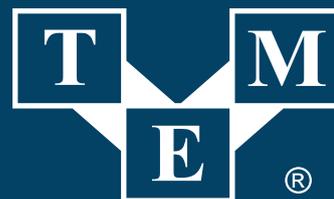
IM ELEKTOR-SHOP

→ Metalldetektor Velleman CS-400

[www.elektor.de/velleman-cs400-advanced-metal-detector-with-lcd](http://www.elektor.de/velleman-cs400-advanced-metal-detector-with-lcd)

### Weblinks

- [1] Review Velleman CS-400:  
[www.elektormagazine.de/news/review-velleman-cs-400-metall-detektor](http://www.elektormagazine.de/news/review-velleman-cs-400-metall-detektor)
- [2] Bundestag beschließt das Ende der UKW/MW/LW-Radios: <https://bit.ly/2IVUxVV>



Electronic Components

TRANSFER MULTISORT ELEKTRONIK

HUMBOLDTSTRASSE 2, 04105 LEIPZIG, DEUTSCHLAND  
TEL. +49 341 212 03 40, TME@TME-GERMANY.DE

tme.eu

facebook.com/TME.eu  
youtube.com/TMElectronicComponent  
instagram.com/tme.eu



## REVIEW

# Portables 3-in-1-Oszilloskop Joy-IT DMSO2D72

Von Harry Baggen

Für Messungen im Feld ist ein tragbares Messgerät besonders nützlich, besonders, wenn das Gerät drei Messfunktionen kombiniert. Dies ist beim Joy-IT DMSO2D72 der Fall, das ein Oszilloskop, ein Multimeter und einen programmierbaren Funktionsgenerator (Arbitrary Waveform Generator, AWG) in einem handlichen Gehäuse verbindet. Schauen wir uns dieses Gerät an!



Trautes Nebeneinander: der Bildschirm des Oszilloskops, AWG und Multimeter.

### Lieferumfang

Bei so vielen Funktionen in einem Gerät fragt man sich unwillkürlich, welches wohl die Hauptfunktion ist. Natürlich, es ist das in diesem Fall zweikanalige Oszilloskop mit einer Bandbreite von 70 MHz und einer Abtastrate von 250 MSamples/s (die Hälfte, wenn man beide Kanäle gleichzeitig verwendet). Das sind schon ordentliche Eigenschaften für ein Messgerät, das (für Elektor-Mitglieder) weniger als 200 € kostet. Hinzu kommt der AWG mit ebenfalls beachtlichen Eigenschaften (Abtastrate 250 MSamples/s) und das Autorange-Multimeter mit separaten Eingängen. Was braucht man mehr? All die Funktionen sind in einem stabilen Gehäuse mit Abmessungen von etwa 20x10x4 cm<sup>3</sup> untergebracht. Das DMSO2D72 sieht aus wie ein großes Multimeter. Das Gehäuse selbst ist durch einen „Stoßdämpfer“ aus Gummi geschützt und lässt sich mit der klappbaren Stütze an der Rückseite aufrecht stellen. Auf der Rückseite befindet sich

auch das Batteriefach für zwei gewöhnliche Li-Ionen-Zellen des Typs 18650, die sich leicht ersetzen lassen. Eine Abdeckung auf der rechten Seite schützt einen USB-C-Anschluss für eine drahtgebundene Stromversorgung und die PC-Kommunikation. An der Oberseite befinden sich drei versenkte BNC-Buchsen (Oszilloskop-Eingänge und AWG-Ausgang). Auf der Vorderseite sind ein 2,8 Zoll großes Display, die darunter liegenden Bedientasten und ganz unten die Multimetereingänge zu sehen. Das Messgerät wird in einem großzügigen Koffer mit viel Zubehör geliefert. Zum Lieferumfang gehören eine robuste Gerätetasche, ein Netzkabel und verschiedene Messkabel.

### Oszilloskop

Die grundsätzliche Bedienung des DMSO2D72 ist ohne vorherige Lektüre des Handbuchs möglich, jedoch lassen sich einige Dinge erst nach einigen Versuchen herausfinden. Mit drei blauen Tasten kann man zwischen den verschiedenen Funktionen wechseln, die vierte blaue Taste gestattet den Zugang zu einem vielfältigen Einstellungsmenü. Direkt unterhalb

des Displays befinden sich vier Funktionstasten, deren Bedeutungen auf dem Display angezeigt werden. Das Farbdisplay ist zwar gut ablesbar und besitzt mit 320x240 Pixeln Platz für reichlich Informationen, ich hätte es aber gerne etwas größer gehabt. Aber dann wäre das Gerät wahrscheinlich auch teurer geworden. Mit den vier Cursortasten können unter anderem die Zeitbasis, die Empfindlichkeit, der DC-Pegel und der Triggerpegel eingestellt werden. Das ist in der „Lernphase“ nicht immer einfach, da sich die Funktionen der Tasten nach dem Drücken der Channel- oder Time-Taste ändern. Leider wird auf dem Display nicht angezeigt, welche der beiden Tasten zuletzt gedrückt wurde. Das Oszilloskop verfügt über fast alle Funktionen eines „normalen“ Oszilloskops, zum Beispiel ein automatisch stillstehendes Oszilloskopbild, die Verwendung von Cursorlinien und die Anzeige verschiedener Messwerte. Die Trigger-Fähigkeiten sind zwar ziemlich begrenzt, aber für die meisten praktischen Situationen ausreichend. Das Oszilloskop verfügt sogar über einen Datenspeicher von 6 kSamples (1 Kanal) beziehungsweise 3 kSamples (2 Kanäle), dessen Inhalt am

oberen Rand des Displays angezeigt wird. Nach der Beendigung des Messvorgangs können Sie mit den Cursortasten durch den Speicher scrollen.

### AWG und Multimeter

Der eingebaute Arbiträr-Funktionsgenerator bietet eine Reihe von gebräuchlichen Kurvenformen und stellt zudem vier Speicher für (mit der entsprechenden PC-Software) selbst programmierbare Kurvenformen zur Verfügung. Die maximale Frequenz ist je nach Kurvenform unterschiedlich. So reicht der Sinusbereich bis 25 MHz, Rechteck bis 10 MHz und Dreieck nur bis 1 MHz. Für programmierte Kurvenformen ist der Bereich wieder mit 5 MHz angegeben. Der Frequenzbereich kann mit den Cursortasten durchlaufen werden. Nach zweimaligem Drücken der Frequenz-Funktions-taste erscheint eine Bildschirmtastatur, auf der Sie einen Wert eingeben können. Es ist möglich, den Generator gemeinsam mit dem Oszilloskop zu verwenden. Ist der Generator eingestellt und sein Ausgang aktiviert, können Sie auf das Oszilloskop umschalten (und der Generator bleibt aktiv). Dies ist ideal für die Messung in einer Schaltung; man legt ein Signal auf den Eingang und misst dann in der Schaltung oder an ihrem Ausgang, welches Signal dort vorhanden ist. Beachten Sie, dass der Wert der eingestellten Ausgangsspannung für eine Ausgangsimpedanz von 50  $\Omega$  gültig ist!

Die Stromversorgung erfolgt über zwei handelsübliche 18650-Li-Ionen-Zellen. Das Multimeter ist eine schöne Ergänzung des Ganzen. Es arbeitet, wie man es von einem durchschnittlichen Multimeter erwarten kann, aber die Genauigkeit ist nicht sehr hoch (Display bis 4000, Grundfehler 0,8%). Bei Messungen „im Feld“ ist dies aber in der Regel nicht so kritisch. Die automatische Bereichsumschaltung funktioniert tadellos und auf dem Display wird auch angegeben, welche Buchsen man für welche Messungen verwenden soll. Es gibt getrennte Eingänge für den Ampere- und Milliamperebereich. Achtung: Der 10-A-Eingang ist nicht mit einer Sicherung versehen! Das Multimeter misst auch Widerstände und Kapazitäten, besitzt einen Diodentester und einen Durchgangsprüfer.

### PC-Software

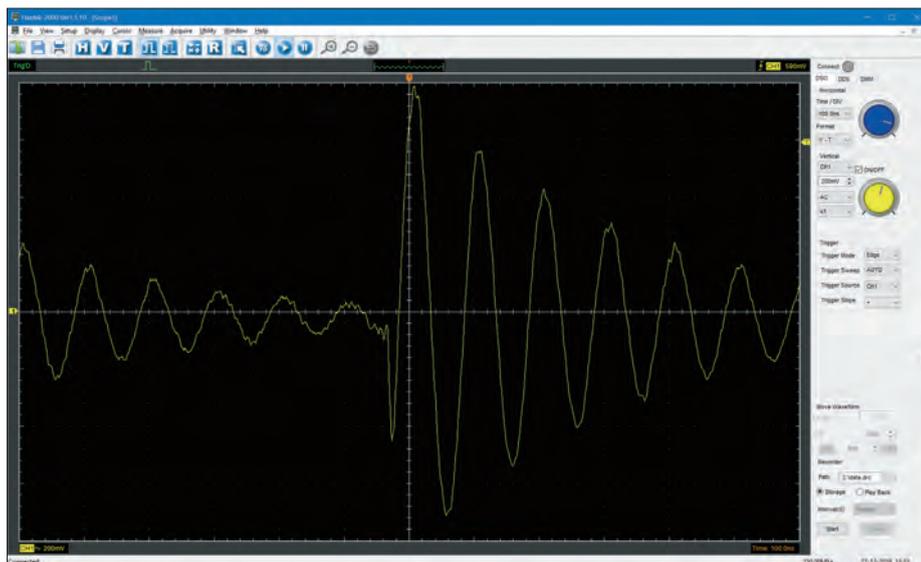
Auf der Website von Joy-IT steht eine Windows-Software (ab Win7) zur Verfügung, mit der Sie das Gerät von Ihrem Computer aus bedienen können. Die



Das DMS02D72 wird mit viel Zubehör geliefert.



Die BNC-Buchsen sind ziemlich tief in die Oberseite des Gehäuses versenkt.



Alle drei Funktionen des DMS02D72 können mit der PC-Software bedient werden.

Installation verlief bei mir reibungslos, aber nach dem Start erschien regelmäßig eine Fehlermeldung (*read allset failed*). Viele Einstellungen waren nicht korrekt und einige funktionierten überhaupt nicht. Eine Recherche im Internet ergab, dass dieser Fehler nicht unbekannt war, die Software konnte die Eigenschaften des angeschlossenen Gerätes nicht auslesen. Das Problem wurde durch die zusätzliche Installation einer älteren Softwareversion (1.1.10) gelöst, danach funktionierte alles einwandfrei und die aktuelle Version 1.1.11 konnte wieder hergestellt werden. Leider ist der Datentransfer im Oszilloskop-Modus recht langsam, aber ansonsten bietet das Programm viele Möglichkeiten, insbesondere das Programmieren (und Hochladen) von Wellenformen.

### Fazit

Für etwa 200 € erhalten Sie ein handliches und robustes Messgerät mit den drei Funktionen, die am häufigsten im allgemeinen Elektronikgebrauch benötigt werden. Die Messbereiche von Oszilloskop und Generator sind praxisgerecht, und wenn Sie erst einmal wissen, wie das Gerät zu bedienen ist, werden Sie gut damit arbeiten können. Es ist kein sehr genaues Messgerät, aber das kann man bei diesem Preis (und braucht man bei dem vorgesehenen Einsatzzweck) nicht erwarten. Die Kombination von drei Messfunktionen macht es zu einem sehr handlichen Gerät, das ich sofort kaufen würde, wenn ich eine solche portable All-in-one-Lösung benötigte. ◀

(191264-04)

**IM ELEKTOR-STORE**

→ Portables 3-in-1-Oszilloskop Joy-IT DMS02D72

[www.elektor.de/19157](http://www.elektor.de/19157)

# Selbstbau-PC fürs Elektronik-Labor

## Tipps für Komponentenauswahl und Bau

Von Dr. Thomas Scherer

Manchen kleinen Elektronik-Laboren geht es wie kleinen Brüdern: Sie müssen abgelegte Klamotten auftragen. Aber alte PCs sind nicht immer ideal, und sogar PCs von der Stange kneifen hier und da. Ein maßgeschneiderter PC ist einfach besser. Lesen Sie nachfolgend, was es beim Selbstbau zu beachten gilt.



Bild 1. Stapel der Innereien des Labor-PCs mit Motherboard, RAM, SSD, CPU und Wärmeleitpaste.

Alte PCs haben im Elektronik-Labor einen Vorteil: Es ist nicht so schlimm, wenn man sie aus Unachtsamkeit himmelt. Außerdem haben diese technischen Antiquitäten oft auch antiquierte Schnittstellen jenseits von USB und Ethernet, die für ältere Projekte und manch teures Messgerät erforderlich sind. Doch die Nachteile alter PCs überwiegen.

Zunächst wäre da die geringere Zuverlässigkeit. Auch wenn PCs gelegentlich zehn Jahre halten, beträgt die typische Abschreibungsfrist nicht ohne Grund drei Jahre. Nicht nur, dass jederzeit etwas ganz oder teilweise kaputt gehen kann, auf alten PCs läuft oft auch ein altes OS, das ein Sicherheitsrisiko darstellt. Hand aufs Herz: Werkelt bei Ihnen ein PC unter Windows XP? Neuere Betriebssysteme sind aufgrund schwächerer Ausstattung und fehlender Treiber oft nicht zu installieren oder würden alte PCs ausbremsen, womit wir schon beim dritten Nachteil wären: die magere Leistung.

Um das zu vermeiden baut man sich seinen Elektronik-PC selbst. Für relativ wenig Geld kann man sich problemlos etwas genau Passendes zusammenstellen.

### PC-Komponenten

Für einen Elektroniker ist der Selbstbau eines PCs eine Kleinigkeit. Wer weiß, dass rot  $\neq$  blau sowie plus  $\neq$  minus ist und was statische Elektrizität bei moderner Elektronik anrichten kann, der braucht kaum Nachhilfeunterricht beim Zusammenstöpseln der Komponenten. Also habe auch ich mir einen Labor-PC zusammengestellt (**Bild 1**) und an einem Abend zusammengebaut.

Gebraucht wird:

- Motherboard
- CPU
- CPU-Kühler
- RAM-Module
- SSD (Festplatte ist out)
- Netzteil
- Gehäuse

Optional sind noch Kleinteile wie Slotbleche mit serieller und paralleler Schnittstelle erforderlich, da diese bei Boards häufig nicht mitgeliefert werden. Bei Ansprüchen an die Grafik (3D-Renderings etc.) kann man statt einer in der CPU integrierten Grafik auch eine dezidierte Grafikkarte einplanen. Natürlich könnte man sich nahezu nach Belieben aus den Angeboten diverser Lieferanten bedienen und die Preissuchmaschine anwerfen. Doch es lohnt sich, vorab ein paar Überlegungen für jede der einzelnen Komponenten anzustellen.

### Motherboard

Zunächst die Grundsatzentscheidung: Intel oder AMD? AMDs neue Prozessoren liefern aktuell etwas mehr „bang for the buck“. Das setzte Intel mächtig unter Druck und sorgt für akzeptablere Preise. Wer nur Windows 10 einsetzt, der kommt mit AMD gut weg. Ich aber installiere auf mehreren Volumes nicht nur Windows 10, sondern auch Linux und MacOS. Folglich muss ich Intel wählen. Braucht man Platz für Steckkarten, geht die nächste Entscheidung für das Board-Format zugunsten von Standard-ATX

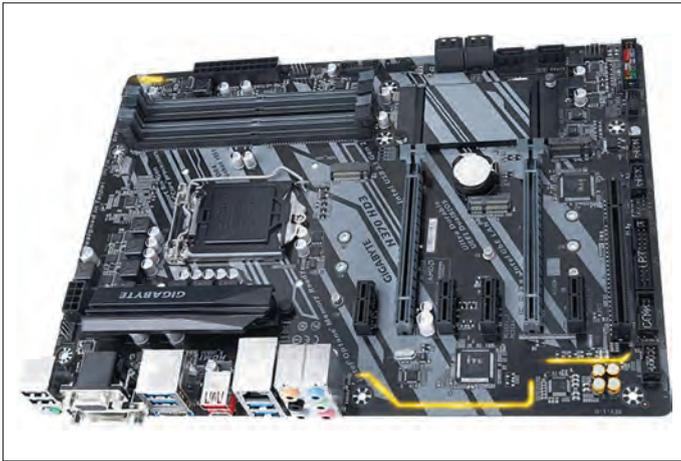


Bild 2. Motherboard mit Intels H370-Chipsatz für den Sockel 1151-2 (Bild: Gigabyte [1]).



Bild 3. Intels i3-9100F „boxed“ – eine schnelle Quad-Core-CPU ohne GPU.

aus. Alternativ kann man auch µATX wählen. Doch es spricht noch zweiter Punkt für ATX: Hierfür gibt es - vermutlich letztmalig – noch Boards mit einem PCI-Slot. Wichtig wenn man noch ältere PCI-Karten einsetzen will. Ansonsten ist man auf modernere PCIe-Karten beschränkt.

Entscheidet man sich für eine Intel-CPU, dann hat man die Auswahl aus Boards mit etlichen Varianten der X3XX-Chipsätze. Hier sollte man nicht versuchen ein paar Euro zu sparen, sondern die Chipsätze H310, B360 und B365 meiden. Diese für einfache Büro-PCs gedachten Varianten haben nämlich eine reduzierte Anzahl nutzbarer PCIe-Lanes, was insbesondere beim Einsatz von M2-NVME-SSDs und Grafikkarten den Durchsatz ausbremst. Es muss zwar auch kein teures Z390-Board sein. Die Varianten H370 oder Z370 tun es. Da das „Z“ am Anfang für einen freien CPU-Takt steht und ich meinen Labor-PC sowieso nicht übertakten will, reicht ein preiswertes H370-Exemplar. Meine Wahl fiel auf das Gigabyte H370 HD3 (**Bild 2**). Es hat den erwähnten PCI-Slot und ist ab 100 € zu haben.

## CPU

Nach der Entscheidung für Intel oder AMD wäre der nächste Aspekt die gewünschte Leistung, die heutzutage grob von der Anzahl der CPU-Cores abhängt. Zwar bietet Intel auch moderne CPUs für knapp 50 € unter der Bezeichnung „Celeron“ oder „Pentium“ an, die es für viele Zwecke tun, doch handelt es sich hier um schwachbrüstige Dual-Core-CPU. Man muss nicht viel drauflegen und bekommt dann schon leistungsfähige und schnelle CPUs in der 100-€-Klasse als Quad-Core-Ausführung. Bei AMD wären das die Typen Ryzen 3 oder 5 – ältere und langsamere oder gar andere als mit Sockel AM4 ausgestattete CPUs sollte man meiden. Bei Intel beginnt die Vernunft ab dem Typ i3-9100. Außerdem muss man hier darauf achten, dass die Typen mit „F“ am Ende mit deaktivierter integrierter Grafik etwas preiswerter verkauft werden. Da bei der Aufrüstung meines Haupt-PC eine moderne AMD-Grafikkarte übrig blieb, entschied ich mich für das rationale Minimum: einen „boxed“ i3-9100F für 80 € (**Bild 3**), der vier Kerne bei einem Basis-Takt von 3,6 GHz (Turbo = 4,2 GHz) bietet. Vor fünf Jahren hätte man für diese Leistung locker über 400 € gezahlt. CPUs von der Stange (als „tray“ gelabelt) sind kaum billiger – daher kaufe ich lieber originalverpackt.

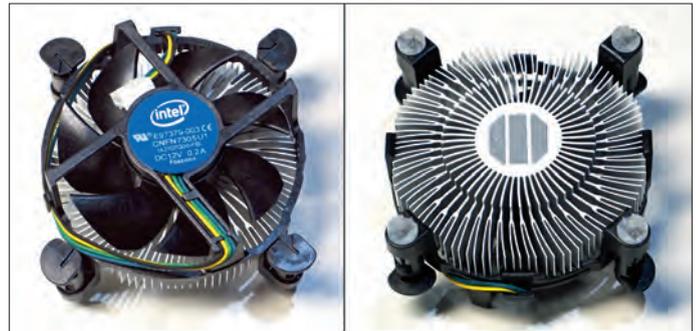


Bild 4. Mitgelieferter CPU-Kühler in der Version „boxed“.

Zum Zeitpunkt des Erscheinens dieser Elektor-Ausgabe könnte es schon Nachfolger geben. In diesem Fall würde ich den Typ i3-10100 vorziehen. Wer mehr Kerne will, der wird bei Intel mit den CPU-Typen i5, i7 und i9 überproportional zur Kasse gebeten.

## CPU-Kühler

Eine „boxed“ i3-CPU bringt auch schon einen ausreichenden Kühler (**Bild 4**) mit. Besonders leise ist er nicht, aber für mich okay.

Wer eine leistungshungrigere CPU einsetzt, der muss auch einen extra Kühler anschaffen. Zwar muss es im Labor nicht so leise sein wie im Büro, aber mindestens 20 € für einen Kühler mit Heatpipes und 120-mm-Lüfter (**Bild 5**) sollte man schon investieren. Die Auswahl ist hier riesig. Eine Wasserkühlung wäre überzogen, denn im Labor wird ja nicht gezockt, oder doch?

## RAM

Beim Arbeitsspeicher geht es erst einmal um die Frage: wieviel? Heute gängig und fast immer ausreichend sind 16 GB. Man ordert also als Kit aus zwei 8-GB-Riegeln, damit darauf beschleunigt = „interleaved“ zugegriffen wird. Bei 32 GB braucht ein 2x16-Kit weniger Energie als ein 4x8-Kit. CPUs nutzen schon lange DDR4-Speicher. Diese 260-Pin-Module benötigt man für Ryzen-CPU mit einem Maximaltakt ab 2.933 MHz. Bei den Coffee-Lake-CPU von Intel (iX-8XXX



Bild 5. Typischer Kühler: Arctic Freezer 34 mit vier Heatpipes (Bild: Arctic [2]).

und iX-9XXX) reichen schon 2.400 MHz. Mehr geht immer und lässt sich per BIOS (XMP = eXtreme Memory Profile) freischalten, es kostet aber auch mehr.

Neben dem Takt ist bei RAM-Modulen auch die Zugriffs-Latenz relevant, die als „CLXX-XX-XX-XX“ angegeben wird. Weniger ist mehr. Ich blieb bescheiden und habe mich für ein 2x8-GB-Kit mit 1,2 V von G.Skill aus der Value-Serie mit CL15-15-15-35 für nur 55 € entschieden.

## SSD

Hier gibt es zwei offensichtliche Kriterien: Kapazität und Geschwindigkeit. Wieviel Platz man braucht, muss jeder selbst wissen. Für Windows 10 samt diverser Entwicklungsumgebungen und CAD-Software plus Office-Paket reichen mir 250 GB bei weitem.

Entscheidender ist die Frage nach der Schnittstelle. Zwar hat mein Board sechs SATA-3-Schnittstellen, doch liegt deren Datendurchsatz bei „nur“ maximal 600 MB/s. Besser und nicht viel teurer sind SSDs mit M2-NVME-Schnittstelle. Dafür bietet



Bild 6. Schnelle M2-NVME-SSD ACE-A80 von Silicon Power (Bild: Silicon Power [3]).

mein Board gleich zwei Slots an. Hier ist dank vier PCIe-3-Lanes ein Datendurchsatz von bis zu 4 GB/s brutto möglich. Reale SSDs bieten locker 3 GB/s.

Ein nicht so offensichtliches Kriterium sind die möglichen I/O-Operationen pro Sekunde, denn diese entscheiden bei modernen Betriebssystemen mit ihren vielen Massenspeicherzugriffen über die Responsivität. Das von mir gewählte Exemplar ACE-A80 von Silicon Power (**Bild 6**) bietet mit 500/600 k/s (lesen/schreiben) gut das Fünffache üblicher SATA-3-Exemplare. Auch der Datendurchsatz ist mit 3.400/3.000 MB/s (lesen/schreiben) nicht zu verachten und kurz vor der theoretischen Grenze. Da so eine SSD gerademal 45 € kostet, ist jeder Gedanke an herkömmliche, elektromechanische Festplatten sinnlos.

Noch ein Hinweis: Vermeiden Sie QLC-SSDs! Diese „Quad Level Cells“ arbeiten tatsächlich mit 16 Spannungspegeln pro Zelle und können so 4 Bit pro Zelle speichern. Das macht sie preiswert, aber auch lahm, und darüber hinaus wenig haltbar. Ohne Tricks des Controllers (Stichwort „Wear Leveling“ und Reserveblöcke) wäre ein Sektor schon nach etwa 200 Schreibvorgängen instabil (gegenüber 2.000 bei TLC). Die Ersparnis lohnt nicht!

## Netzteil

Aus Umweltschutzgründen sollte man hier auf den Wirkungsgrad achten. Netzteile unter dem Label „80 plus“ sind in die Kategorien Bronze, Silber und Gold eingeteilt. Schlechter als Bronze sollte man nicht kaufen.

Weiter relevant ist die zur Verfügung gestellte Leistung. Hat man eine 65-W-CPU und nutzt die integrierte CPU-Grafik, dann dürfte es schwer fallen, das Gesamtsystem auf viel mehr als 100 W Leistungsbedarf unter Volllast zu bringen. Bei einer Grafikkarte käme noch deren Energiebedarf dazu. Für Labor-PCs irrelevant benötigen Spitzenmodelle für Zocker mehrere hundert Watt. Ein PC mit „normaler“ Grafik-Karte der unteren Mittelklasse sollte mit knapp über 200 W hinkommen. In der Regel dümpelt ein Labor-PC bei 20...40 W vor sich hin, wenn er nichts Besonderes zu tun hat.

Leider gibt es schon lange keine brauchbaren 250-W-ATX-Netzteile mehr. Bei höherer Leistung laufen diese Netzteile aber mit suboptimalem Wirkungsgrad. Für mich war ein 350-W-Exemplar von LC-Power des Typs LC420-12 (**Bild 7**) der ausreichende Kompromiss, der mich lediglich 25 € kostete. Seine Anschlüsse gehorchen dem ATX-Standard V2.31. Das Netzteil hat einen Wirkungsgrad von 88 % und ist besonders im Schlafmodus sehr effizient. Darüber hinaus ist der eingebaute Lüfter eine ausreichend leise 120-mm-Variante (je größer desto leiser).

## Gehäuse

Die Verpackung des PCs ist mehr oder minder Geschmackssache. Ich habe mich für das mit 20 € extrem preiswerte Modell CS-1103 von Aerocool entschieden, das an der Front oben zwei USB-2.0- und eine USB-3.0-Buchse bietet und zudem eine schwarze Alufontplatte hat (**Bild 8**). Was will man mehr?

## Außerdem

Die schon erwähnte Grafikkarte ist eine Radeon 560 von Sapphire mit 2 GB RAM. Das reicht für einen 4K-Monitor mit ruckelfreien 60 Hz. Hinzu kam noch ein simpler 90-mm-Lüfter mit niedriger Drehzahl für die Gehäuse-Rückseite. Wer viel Luft schaufeln muss, der wählt besser ein Gehäuse passend für mehrere 120- oder gar 140-mm-Lüfter. Schließlich kam

noch ein Slot-Blech mit Kabeln dazu, da ich die serielle und parallele Schnittstelle des Boards auch nutzen wollte. Eine per USB emulierte serielle Schnittstelle macht nämlich bei vielen schnellen Transfers weniger Bytes oft Probleme.

### Aufbau

Nicht umsonst sind die meisten Komponenten in Antistatik-Folien verpackt. Ein professionell ausgestatteter Arbeitsplatz mit Erdungsarmband wäre zwar gut, doch es reicht auch, wenn man weiß, was man tut.

Gesetzt den Fall, man hat keine schmutzigen Finger mit zu viel Hautfett, kann man alle Teile einfach anfassen – allerdings möglichst an Masseteilen und dabei nicht über Kunststoffböden oder Kunstfaserteppiche laufen.

Ein Board kann man einfach an dem Teil mit den Schnittstellen anfassen, da ist viel Metall mit Massepotential. Bevor man es ins Gehäuse einbaut, montiert man zunächst die CPU. Auch die kann man mit einer Hand packen und mit der anderen das Board an den Schnittstellen berühren, bevor man den Hebel mit der CPU-Fassung öffnet, die schwarze Plastikabdeckung entfernt und die CPU richtig platziert. Wegen entsprechender Nasen kann man das kaum verkehrt herum machen. Dann verriegelt man die CPU, platziert die Schnittstellenblende auf die Schnittstellen und setzt das Board ein. Meist sind sechs bis neun passende Schrauben im Lieferumfang des Gehäuses. Jetzt kommt der CPU-Kühler an die Reihe. Er ist normalerweise schon mit einer Schicht Wärmeleitpaste versehen und fast immer mit zu viel davon. Drückt man den Kühler auf die CPU, dreht ihn etwas und zieht ihn wieder ab, dann kann man die Paste auf der CPU mit einem Papiertaschentuch wegwischen. Jetzt müsste die Pastendicke besser passen. Ich habe sie gleich ganz entfernt und eine bessere, metallhaltige Paste sehr dünn mit einem Spatel aufgetragen.

Jetzt kommen die SSD an ihren Platz und die beiden RAM-Module. Das Handbuch klärt auf, welche Sockel zusammengehören, damit das Interleaving klappt. Nun braucht es etwas Sorgfalt, bis man all die Kabel des Gehäuses nach Handbuch eingesteckt hat.

Dann kommen die Stecker des Netzteils. Hier gilt es zu beachten, dass ein sechspoliger Stecker in der Regel für die Grafikkarte gedacht ist. Die Zusatz-12-V-Versorgung des Boards benötigt einen vier- oder achtpoligen Stecker. Das ist sehr wichtig!

Eine eventuelle Grafikkarte kommt in den PCIe-Slot mit 16 Lanes, der am nächsten an der CPU ist.

Jetzt kann man den PC einschalten. Am angeschlossenen Monitor sollte das BIOS zu sehen sein. Windows 10 kommt mit den Standard-Einstellungen zurecht. Andere Betriebssysteme benötigen etwas Feintuning.

Übrigens zeigte sich dieser sehr preiswerte Labor-PC subjektiv fast genau so fix wie mein sehr viel teurerer Hauptrechner. Jetzt muss ich erst einmal die ganze nötige Software installieren.



Bild 7. 350-W-Netzteil LC420-12 von LC-Power (Bild: LC-Power [4]).

Das dauert erfahrungsgemäß deutlich länger. Die Zeit für den Bau des PCs fällt also gar nicht ins Gewicht. ◀

191191-01



Bild 8. Äußeres und Inneres: der fertige Labor-PC.

### Weblinks

- [1] [www.gigabyte.com/de/Motherboard/H370-HD3-rev-10#kf](http://www.gigabyte.com/de/Motherboard/H370-HD3-rev-10#kf)
- [2] [www.arctic.ac/de\\_de/freezer-34.html](http://www.arctic.ac/de_de/freezer-34.html)
- [3] [www.silicon-power.com/web/product-P34A80](http://www.silicon-power.com/web/product-P34A80)
- [4] [www.lc-power.com/produkte/pc-netzteile/office-serie/lc420-12-v231/](http://www.lc-power.com/produkte/pc-netzteile/office-serie/lc420-12-v231/)

# Intelektor: Der Elektor-Schachcomputer (1981) Tiny Chess 86 auf Intel 8088 portiert

Von Jan Buiting

*„Wer Schach spielen kann, ist sicherlich oftmals auf der Suche nach einem Partner. Einem Partner, der immer Zeit hat, nicht ungeduldig wird, ziemlich stark spielt, und einen auch mal ein bisschen mogeln lässt, wenn man ihn nur nett darum bittet.“*



Das obige Zitat stammt wörtlich aus der Elektor-Ausgabe vom April 1981, und der einzige Grund für den Nachdruck hier ist der glückliche Zufall, dass der damalige Prototyp des Intelektor-Schachcomputers nicht im Juni 2006 weggeworfen wurde, als Elektor seine Büros im niederländischen Beek verließ, um ins Schloss Limbricht (ebenfalls in den Niederlanden) zu ziehen. Als eine der letzten Aktivitäten dieses Umzugs wurde das Elektor-Labor an einem heißen Juni-Nachmittag im großen Stil ausgeräumt. Dabei wurde eine Tonne Laborausrüstung, Demos, Datenbücher, Zeitschriften, Originale von Zeichnungen, Platinen und Prototypen – allesamt von der damaligen Geschäftsleitung als „untauglich für die zukünftige Nutzung“ eingestuft – aus zwei großen Fenstern geworfen, um so in einem großen Müllcontainer zu landen. Glücklicherweise hat das Labor-Team einige der vorzeigbaren Prototypen in verschiedenen Ecken und Winkeln gefunden und gerettet, um eine kleine Sammlung unter dem Titel „Retronik“ zusammenzustellen. Die hehren Stücke wurden auf dem Dachboden des Schlosses gelagert, wo sie etwa zehn Jahre bis zum nächsten Umzug – diesmal nach Aachen – sicher verwahrt wurden. Ich weiß nicht mehr, wer den Intelektor-Prototypen tatsächlich vor der Entsorgung gerettet hat, irgendwie muss es wohl am Gehäuse gelegen haben, in dem der Schach-Computer steckte. Auch nach 30 Jahren strahlt er wie neu mit seiner Acrylabdeckung, die speziell für die vielen Elektronik-Messen und Ausstellungen in ganz Europa hergestellt wurde, auf denen Elektor in den 1980er Jahren einen Stand hatte. Das Gehäuse stammt übrigens von Jan Visser, der noch immer im Elektor-Labor arbeitet. Jan war hoch erfreut, den Intelektor nach so vielen Jahren wiederzusehen.

### Vorzeigbar

Das Gehäuse des Intelektor verrät, dass er zum Vorzeigen gedacht war. Und wohl auch um zu überleben! Damals gab es auf vielen Hobby-orientierten Messen in D und NL immer einen gut frequentierten Elektor-Stand. Im Gegensatz zu den heutigen „professionellen“ Messen wie der *embedded world*, der *productronica* oder der *electronica* wurde er von den Lesern damals regelrecht belagert. Lange Schlangen und gar Schubereien waren keine Seltenheit, vor allem, wenn es Platinen und Bücher im Abverkauf gab, Demos verfügbar waren oder Autoren und Entwickler am Stand waren, mit denen man intensive Diskussionen führen und fachsimpeln konnte. Damals waren die Leser sehr an direktem Kontakt mit Elektor-Redakteuren, Labor- und Verlagsmitarbeitern interessiert – sei es auch nur, um die Menschen hinter dieser seltsamen „Zeitschrift aus Holland“ kennenzulernen - und an Schnäppchen, versteht sich. Zurück zum Intelektor-Gehäuse: Der versenkte Schalter dient der Interrupt-Auslösung. Die von Jan Visser in die Acrylabdeckung geschnittene, 12 × 17 mm große rechteckige Aussparung sollte verhindern, dass neugierige Besucher oder sonstige neckische Zeitgenossen auf der Messe eine Demo-Schachpartie stören. Der Schalter ist auf einem kleinen Stück Lochrasterplatine befestigt, das man nach Abschrauben der Abdeckung an den Drähten baumeln sehen kann (**Bild 1**). Die zweite bemerkenswerte Sache ist das Produktetikett mit der Aufschrift „intelekt“ (sic). Es besteht aus zwei rechteckigen Teilen einer Acrylplatte, die je nach der gewünschten Sprache umgedreht werden konnten.

**Bild 2** zeigt den Trick. Für die Niederlande fotokopierte Jan nur den niederländischen Titel des Artikels, fügte Monat und Jahr der Veröffentlichung sowie die Platinen-Nummer hinzu und klebte das Resultat auf eine der Platten. Bei der raffinierten

**EST<sup>20</sup> 2004**

www.elektor.tv



Retronik ist eine Rubrik, die antiker Elektronik und legendären Elektor-Schaltungen ihre Reverenz erweist. Beiträge, Vorschläge und Anfragen telegrafieren Sie bitte an Jan Buiting (editor@elektor.com).

aussehenden deutschen Version erinnerte sich Jan, er hatte damals die Titelseite der deutschen Ausgabe vom April 1981 abgerissen und dann die in Bild 2 sichtbare Collage verwendet. Der Gehäusedeckel besteht aus einer 3 mm dicken Plexiglas-Platte – auf Maß gebogen mit einer Maschine, die ich selbst noch 1985 in der alten Werkstatt gesehen habe. Der Deckel ist schlagfest und hat seinen Zweck erfüllt: die Elektronik zu schützen. Der Rest des Gehäuses ist schlichtes Metall mit Ein/Aus-Schalter, der Kaltgerätebuchse und serieller I/O per fünfpoliger DIN-Buchse (!) an der Rückseite (**Bild 3**).

### Unglaubliche 16 Bit

In der 1981er April-Ausgabe von Elektor war eine spezielle Beschreibung von 16-Bit-CPUs enthalten, die damals noch recht neu und nichts für Hobby-Elektroniker waren. Damals waren



Bild 1. Intelektor mit abgenommenem Deckel. Der Interrupt-Taster ist ein „Digitast“-Typ mit taktilem Feedback auf einer kleinen Lochrasterplatine, die mit Abstandshaltern am Deckel befestigt ist.



Bild 2. Die abschraubbare Intelektor-Produktbeschreibung kann man für niederländische oder deutsche Beschriftung wenden - passend zum Ausstellungs- oder Vorföhrrort.

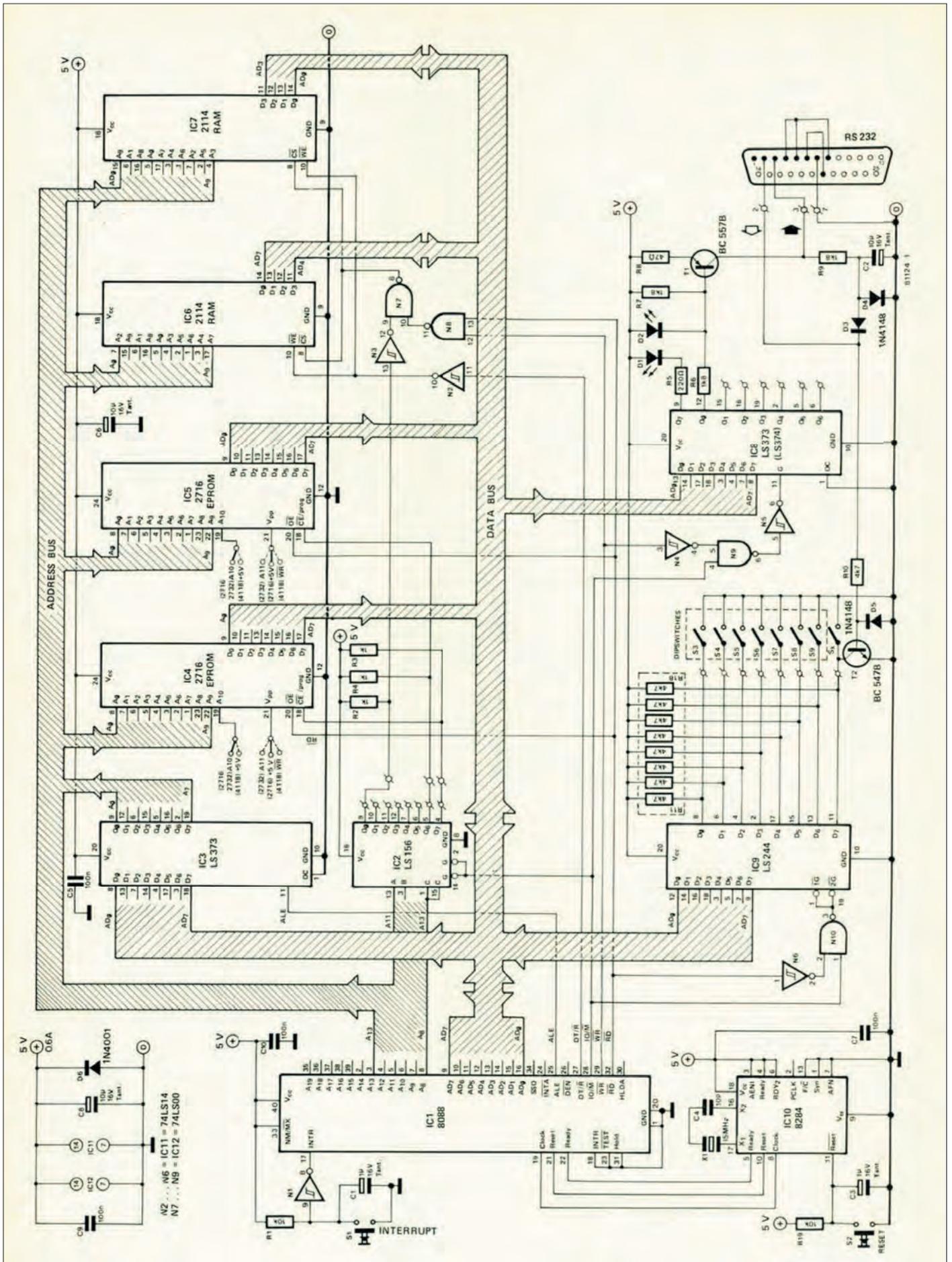


Bild 4. Scan der Intelkor-Schaltung aus Elektor April 1981.

noch 8-Bit-CPU's wie der 6502 oder der Z80 in sogenannten Heim-Computern der Hit, obwohl schon 16-Bit-Technik in professionellen und industriellen Rechenanlagen eingesetzt wurden. Diese 16-Bit-Sonderbeilage informierte vor allem über die Architektur dieser Monster (darunter der Intel 8088). Im Intelektor-Artikel selbst wird am Beispiel des 8088 (der CPU des originalen IBM-PCs) ausführlich für ein an 8 Bit gewohntes Publikum erklärt, wie der zweite 8-Bit-Block eines 16-Bit-Datenwort zur Peripherie übertragen wird. Wie lassen sich bloß 16 Bit parallel transportieren, wenn die Chips so wenige Pins aufweisen? Die Antwort: *Intelligent*, durch Multiplexen mit dem Adressbus unter Verwendung der vom Prozessor generierten Timing-Signale. Der Rest ist Geschichte, da die 8088-CPU im Gegensatz zum 8086 und dem achtbittigen 8085 nicht so recht ankam, zumindest nicht in Elektor.

Der Intelektor-Artikel gibt sich große Mühe zu begründen, dass 16-Bit-Datenverarbeitung für einen Schach-Computer unbedingt sein musste. Was das Computer-Schach betrifft verweist der Artikel auch auf frühere, allgemeinere Elektor-Beiträge, die Vorarbeit geleistet haben: „Wie ich gegen den Computer gewann“ David Levy und „Computer und Schach“, beide vom Januar 1979. Im April 1981 wurde es dann mit der Kombination der 16-Bit-Intel-CPU und passender Software richtig praktisch.

### Anfänge

Intelektor ist ziemlich berühmt, zumindest ist er im „Chess Programming Wiki“ [1] erwähnt. Dort steht, dass Intelektor auf einer an den 8088 angepassten Version von Tiny Chess für 8086 beruhte, die von dem Niederländer Jan Kuipers stammte. „TC86“ war in Assembler geschrieben! Der Schach-Computer trat erstmal öffentlich auf, als er beim offenen niederländischen Computer-Schachwettbewerb (DOCCC) 1981 antrat. Möglicherweise war das genau der Kasten, den ich hier vor mir hatte!

Die Spur zu Tiny Chess stammt aus der kleinen Erwähnung im Artikel von 1981, im Abschnitt über die Bedienung des Intelektor. Zitat:

„Jetzt sollte auf dem Bildschirm folgendes zu lesen sein:

```
TINY CHESS V1.0 (sic)
LEVEL IS 1 CHANGE TO _
```

Ohne diesen Hinweis hätte ich kaum gewusst, woher die in zwei EPROMs des Typs 2716 gebrannte Software stammen könnte. Intelektor gibt seine Schritte über eine serielle 3-Draht-Verbindung via dummem (passiven) Terminal aus und liest humane Züge ein.

### Obskure Software

Bis vor etwa zehn Jahren war die Firmware für Mikrocontroller- und Mikroprozessor-Projekte via Elektor-Software-Service (ESS) in Form von EPROMs erhältlich. Praktisch wurden sie früher oft auf einem Junior-Computer von - Sie ahnen es - Jan Visser für die Leser gebrannt. Weder im Intelektor-Artikel von 1981 noch in der Stückliste zur einseitigen (!) Platine gab es einen Hinweis auf Bezugsquellen für die EPROMs. Der Artikel ignoriert Software-Fragen hartnäckig und geht nur auf die Hardware bzw. Schaltung und Platine sowie das spielerische Können richtig ein. Stattdessen werden drei damit gespielte Schachpartien beschrieben.



Bild 3. Die Rückseite des Intelektor-Gehäuses ist normalerweise nicht für das Publikum zugänglich. Beachten Sie auch den Platz auf der Platine für den INT-Taster, der extern per Kabel Teil angeschlossen ist.

Heute in Zeiten von Open Source etc. wäre das ein Skandal gewesen. Bemerkenswerterweise ist „Intelektor“ eindeutig männlich, was mich vermuten lässt, dass es sich um eine schlechte Übersetzung der niederländischen Pronomen „hij“ bzw. „zijn“ handelt.

Zurück zu den EPROMs: Ich habe alle ESS/EPS-Listen des gesamten Elektor-Bandes von 1980-1989 abgeklappert und habe keine Software zum Intelektor-Projekt gefunden. Es gab nur eine Platine, die allerdings nach etwa drei Jahren verschwand. Das handgeschriebene „505“ auf den grünen Etiketten auf den beiden EPROMs im Prototyp half mir nicht weiter, da bis 1989 kein ESS-Artikel mit dieser Nummer aufgeführt oder in irgendeiner „Nachlese“ erwähnt wurde. Überraschenderweise annoncierten aber einige der damals bekannten Anbieter von Bausätzen in den späteren Monaten fröhlich den Intelektor-Bausatz „komplett mit Software“ in Elektor.

### Die Hardware

Der angebliche Autor des Intelektor-Artikels Jan Kuipers hatte bemerkt, dass ein kompliziertes Takt-Timing erforderlich sei, damit die 8088-CPU mit ihren zusätzlichen acht Datenbits effektiv ist. In typischer Intel-Manier wurde hierfür mit dem 8284 ein extra Taktoszillator-Chip eingesetzt. Man findet ihn

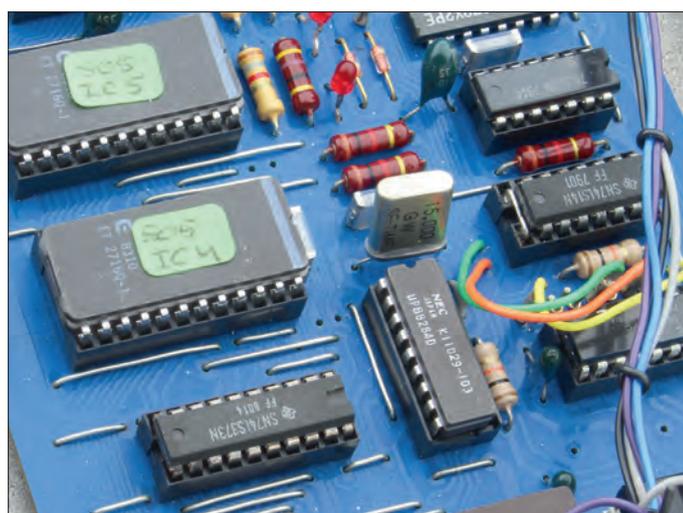


Bild 5. Ein 8088 hat keinen On-Chip-Taktgenerator, sondern erfordert mit dem 8284 hierfür ein spezielles IC.

## Digitast

Der Artikel aus dem Jahr 1981 beschreibt auch einen Drucktaster für die Platinenmontage, der „Digitast“ genannt wird. Als ich 1985 als Volontär für die UK-Ausgabe von Elektor anging, erhielt ich Briefe von Lesern, die mit diesem Digitast nichts anzufangen wussten. Meine Standardantwort war, dass es sich um ein deutsches Kunstwort aus „Digital-Taster“ handelte. Schwierig vorzustellen für Inselbewohner, wo ein „Schlüsselbrett“ (Keyboard) als Tastatur gelten muss ;-)  
Intelektor hat zwei Digitaster, je einen für Reset und Interrupt. Auf dem Prototyp ist nur der für „RES“ auf der Platine, während „INT“ per Kabel angeschlossen und durch eine



Art „Anti-Manipulationsloch“ gesichert ist. Problematisch an diesen Digitastern ist, dass sie alle den gleichen Platinen-Footprint haben, obwohl nur einige Modelle eine interne Drahtverbindung aufweisen. Wenn man also einen Digitast außerhalb der Platine montieren will, muss man die nun fehlende Verbindung auf der Platine durch eine Drahtbrücke ersetzen, damit das Ganze noch funktioniert.

Ohne diese will der 8088 möglicherweise trotz stundenlanger Fehlersuche bei 8 und auch bei 16 Bit nicht arbeiten! Glücklicherweise wird auf diese Eigenheit von Digitastern im Artikel eingegangen. Dann wäre also eine mehr als die eh notwendigen 43 Drahtbrücken auf der einseitigen (!) Leiterplatte zu biegen und zu löten.

in der Intelektor-Schaltung in **Bild 4** und auf der Platine von **Bild 5** in der Nähe des 15-MHz-Quarzes. Die Kosten dieses Chips waren im Vergleich zum 8088 im 40-poligen DIP-Gehäuse absurd hoch.

Obwohl die Schaltung einen 25-poligen Sub-D-Stecker für die serielle Verbindung zum Terminal vorsieht und sie protzig „RS232“ beschriftet, verdient sie diesen Namen sicher nicht, da sie effektiv nur zwei Leitungen hat: RxD und TxD. Es ist kein Handshaking implementiert, und der Pegelhub liegt bei 0...5 V statt den normgerechten  $\pm 10$  V. Diese Art von Pseudo-RS-232 mit BC54x sollte in späteren Jahren in vielen Elektor-Projekten auftauchen, was bei der Verwendung professioneller Terminals und anderer für die echte RS-232 konzipierte Geräte oft zu Kommunikationsproblemen und daher zu wildem Fluchen führte. Zur Entschuldigung: Das war lange vor der Einführung des MAX232 etc., wo integrierte Ladepumpen für negativen Pegel sorgten.

### Spielstärke

Intelektor ist kein vollgrafischer Schachcomputer. Nach dem Prompt zeigt er die Aufstellung an, allerdings auf sehr primitive Weise, wie es vom „nichtintelligenten“ „Elekterminal“ (20 Zeilen mit je 40 Zeichen) kaum anders zu erwarten war. Resultat ist eine äußerst primitive, hauptsächlich aus Kreuzen und Punkten bestehende Darstellung der Figuren auf dem Schachbrett. Vermutlich haben die meisten Spieler den Bildschirm ignoriert, die Züge einfach als Text gelesen und die relevanten Figuren auf einem echten Schachbrett nachgestellt. Für einen Zug gibt man ein:

- das Feld, das die zu ziehende Figur enthält;
- ein Leerzeichen;
- das Zielfeld;
- CR (Wagenrücklauf).

Auf eine Antwort von Intelektor bzw. seinen Zug kann man angefangen von „nur“ 25 s (auf der niedrigsten Stufe 1) bis zu, ähm, vielleicht einer Stunde auf „seiner höchsten“ Stufe 8 warten. Ein Beispiel für einen Mensch-Maschine-Dialog:

01W: e2 e5 — ein illegaler Zug, daher:

01W: e2 e4 (CR)

01B: c7 c5

02W: (warten auf Ihren nächsten Zug)

Die Software hat einige Befehle zur Spielkonfiguration eingebaut, wie z.B. Spielerwechsel (CTRL-X), Autoplay (CTRL-A) und die Einstellung des Levels (CTRL-N).

Der Original-Artikel hat einen langen Abschnitt am Ende, wo drei gegen Intelektor gespielte Schachpartien diskutiert und analysiert werden, komplett mit Zuglisten, spannenden Bretttaufstellungen und Kommentaren. Das Fazit des Artikels ist interessant, da es Hinweise gibt auf die Modifizierung der Software zwecks Verbesserung des Endspiels oder der Verwandlung von Bauern in andere Figuren als nur eine Dame. Allerdings wurde jedoch davor gewarnt, dass Software-Mods zusätzlichen Speicher brauchen könnten – amüsant, da die Intelektor-Platine schon für den Einbau größerer EPROMS vom Typ 2732 vorbereitet ist. Alles in allem ist der Artikel von 1981 sehr ausgewogen, da er sowohl die Stärken als auch die Schwächen von Elektors erstem eigenständigen und dedizierten Computer mit Tiny Chess diskutiert. Neben positiven Anmerkungen wie „...er spielt gut, und das ist doch das Wichtigste.“ und „Im Prinzip gebraucht Intelektor die geballte Kraft eines 16-Bit-Mikroprozessors...“ findet sich die ernüchternde Einschätzung „...beim Endspiel aber hat er deutliche Schwächen.“

Es reizt mich, unseren Intelektor einmal einzuschalten und zu schauen, ob er noch spielt. Eine Partie gegen einen modernen 32-Bit-Gegner wäre doch interessant, oder? Aber zuerst müssen diese blauen Tantal-Kondensatoren ersetzt werden, um kleinere Explosionen vor Spielbeginn zu verhindern. ◀

190382-B-03

### Weblink

- [1] [Chess Programming Wiki:  
www.chessprogramming.org](http://Chess Programming Wiki: www.chessprogramming.org)

# 30+

Dieses Kit enthält über 30 Komponenten, Bauelemente und Module

Wenn man wirklich innovative Projekte umsetzen möchte, muss man sich direkt mit elektronischen Bauelementen befassen. Dies stellt aber viele Einsteiger vor größere Probleme. Genau hier setzt dieses Bundle an, in dem es nicht nur um RFID geht. Es bietet eine Fülle an Praxisprojekten, die mit dem im Bundle enthaltenen RFID-Starterkit für Arduino aufgebaut werden können.

# „Elektor - Arduino Home Automation Bundle“

# elektor



## Description:

Neben den einfachen Elementen wie LEDs und Widerständen sind auch komplexe und hochmoderne Module enthalten, beispielsweise:

- ein Feuchtigkeitssensor
- eine Multicolor-LED
- eine LED-Matrix mit 64 Leuchtpunkten
- eine vierstellige 7-Segment-Anzeige
- eine Infrarot-Fernbedienung
- ein komplettes LCD-Display-Modul
- ein Servomotor
- ein Schrittmotor mit Steuermodul
- eine RFID-Platine mit Schlüsselkarte

Neben präzisen digitalen Thermometern, Hygrometern, Belichtungsmessern und verschiedenen Alarmanlagen entstehen auch praktisch einsetzbare Geräte und Anwendungen wie etwa ein vollautomatischer Regensensor, eine schallgesteuerte Fernbedienung, eine multifunktionale Klimamessstation und vieles mehr.

## Mehr Informationen:

[www.elektor.de/elektor-arduino-home-automation-bundle-de](http://www.elektor.de/elektor-arduino-home-automation-bundle-de)





## PCB Art

### Über die Grenzen der industriellen Fertigung hinaus

Von Tessel Renzenbrink

Saar Drimer ist PCB-Kunstschaffender. Drimer verbindet seine Ingenieurs- und Kunst-Fertigkeiten und stellt wunderschöne Leiterplatten her, die die Grenzen des traditionellen Elektronikdesigns sprengen. Für die Herstellung seiner Entwürfe arbeitet er mit dem Platinenhersteller Eurocircuits zusammen. Zuerst war die Zusammenarbeit holprig und Drimer war oft frustriert über die Beschränkungen, die ihm der industrielle Herstellungsprozess auferlegte. Bei Eurocircuits brachten ihm seine unkonventionellen Designs den Spitznamen „Troublemaker“ ein. Aber im Laufe der Jahre kamen sich Drimer und Eurocircuits näher, weil einer aus der Sicht des anderen lernte. Ich habe mit dem geschäftsführenden Gesellschafter von Eurocircuits, Dirk Stans, und Saar Drimer über diese Partnerschaft gesprochen. Aus diesen beiden Gesprächen ergibt sich ein gemeinsames Thema. Beide Männer wollen Elektronikdesign zugänglicher machen und mehr Farbe in die Branche bringen.



Skarabäus-Panels (Foto: Eurocircuits).

#### Freiformplatinen

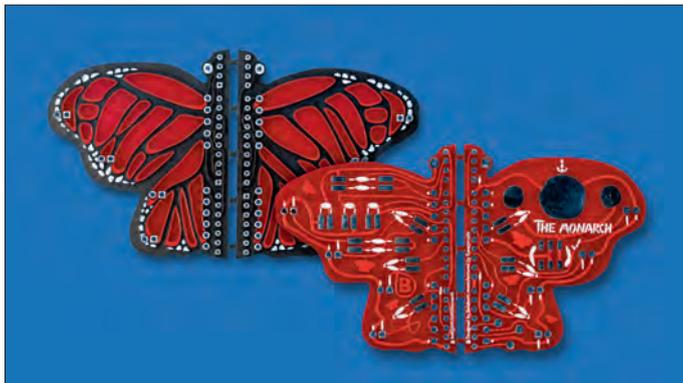
Um seine Ideen zu verwirklichen, schrieb Drimer seine eigene Platinendesign-Software PCBmode [1]. „Ich habe meine eigenen Tools entwickelt, weil die bestehenden Grenzen hatten“, sagt Drimer. „Als CAD-Werkzeuge (Computer Aided Design) in Mode kamen, waren die begrenzten Computerressourcen sehr einengend. Eine Beschränkung war, dass man nur gerade Linien und 45-Grad-Winkel, aber keine kurvigen Formen und Leiterbahnen herstellen konnte. Es ist sogar heute noch sehr schwierig, kreative Freiformplatinen mit traditionellen Werkzeugen zu erstellen. Ich wollte etwas, das es mir erlaubt, alles auf der Platine zu verwirklichen, egal wie komplex es sein mag. Also habe ich meine eigene Open-Source-Design-Software entwickelt. Und das Schöne daran ist: Die Leiterplatten-Maschine kann mit dem Platinenmaterial all diese Dinge ohne Modifikation tun, es ist nur nicht geschehen, weil die Entwicklungstools es nicht erlaubten. Aber die Maschinen können es, denn ihnen ist egal, ob etwas quadratisch ist oder wie eine Amöbe aussieht“.

Dirk Stans: „Mit der Design-Software von Saar kann man im wahrsten Sinne des Wortes wie ein Künstler zeichnen. Wenn man sich Saars Entwürfe aus rein industrieller Sicht anschaut, denkt man: Dieser Mann ist ein Spinner. Nichts, was er tut, entspricht den Normen. Aus industrieller Sicht hat es einfach keinen Sinn. Aber dann fängt man an, es aus einer anderen Perspektive zu betrachten: Es sind wirklich schöne Projekte und sie sind sehr innovativ. Die Zusammenarbeit mit Saar gibt uns die Möglichkeit, bei Eurocircuits neue Dinge auszuprobieren. Über die Jahre hinweg haben wir voneinander gelernt. Wir müssen verstehen, was er erreichen will, und er muss die Grenzen der industriellen Fertigung verstehen. Diese Aufgeschlossenheit hat es uns ermöglicht, gemeinsam schöne Dinge zu machen.“

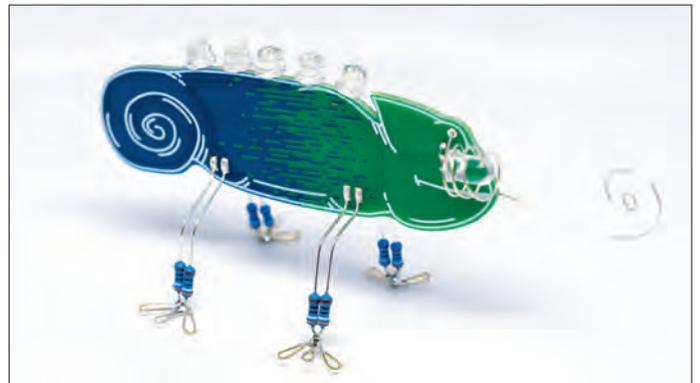
#### Der Marking-Editor

Der Umgang mit Produktnummern ist ein Beispiel für dieses wachsende Verständnis. Als Drimer eines der ersten bei Eurocircuits bestellte Boards erhielt, war eine Produktnummer scheinbar willkürlich auf seinen akribisch durchdachten Entwurf gedruckt. Er war damit nicht zufrieden, doch nachdem er das Thema angesprochen hatte, erklärte Stans, dass die Produktnummer für eine vollständige Rückverfolgbarkeit, also die Möglichkeit, jede Leiterplatte und jedes Bauteil bis zu ihrem Ursprung zurückzuverfolgen, unerlässlich sei. Nachdem er den Grund dafür erfahren hatte, akzeptierte Drimer bereitwillig, dass die Anforderungen der Industrie an die Rückverfolgbarkeit seinen Sinn für Ästhetik übertrumpften. Aber Drimers Bemerkungen mündeten wiederum in Stans Idee, den Entwicklern die Freiheit zuzugestehen, die Nummer dort zu platzieren, wo sie wollten.

„Eine der Herausforderungen bei der Platinenproduktion ist es, zu verstehen, was der Entwickler will“, sagt Stans. „Der wichtigste Schritt ist die Übersetzung der CAD-Daten in CAM-Daten (Computer Aided Manufacturing). Man will natürlich alle



Der Monarch mit schwarzer und roter Lötstoppmaske (Foto: Saar Drimer).



The Gent, ein blau-grünes Chamäleon, das im Dunkeln leuchtet (Foto: Saar Drimer).

möglichen Fehler beseitigen, bevor man etwas in die Produktion gibt. Um dies zu erreichen, bieten wir Entwicklern eine Reihe von kostenlosen Online-Tools an, mit denen sie das Design überprüfen können, bevor es in die Fertigung geht [2]. Wir haben uns entschieden, ein Tool namens *Marking Editor* hinzuzufügen, mit dem Entwickler die Produktnummer an beliebiger Stelle auf der Platine platzieren können. Die Gespräche mit Saar haben zu dieser Idee beigetragen. Aber zu sagen, dass wir das Tool nur für ihn gebaut hätten, wäre ein wenig übertrieben. Andere Leute hatten nämlich ähnliche Anfragen gestellt. Einige Entwickler wollen ihre eigene Nummer oder ihr Logo oder ein Foto auf der Platine anbringen. Dies lässt sich jetzt mit dem *Marking Editor* alles realisieren. Die Arbeit mit Saar hat uns dazu gebracht, den Produktionsprozess anders zu betrachten. Das Hinzufügen mehrerer Lötstoppmaskenfarben auf einer einzigen Platine zum Beispiel ist eine neue Funktionalität, die aus unserer Zusammenarbeit mit Saar hervorgegangen ist.“

### Lötstoppmaske in zwei Durchgängen

Drimers Firma Boldport bietet Elektronik-Kunsthandwerk als Dienstleistung an: Kunden können Entwürfe in Auftrag geben oder lizenzieren. Am bekanntesten ist sie jedoch für ihre exquisiten DIY-Kits [3]. Boldport bietet ein großes Angebot an Elektronikprojekten, zum Beispiel die Insektenbausätze, die einen Schmetterling oder einen Marienkäfer darstellen. Bei einem Kauf wird man eingeladen, dem Boldport Club beizutreten - einer aktiven Community auf einem von Drimer betreuten Discord-Server. Die neueste Kreation der Käfer-Reihe, an der Boldport und Eurocircuits arbeiten, ist der Skarabäus, eine käferförmige Leiterplatte mit einer mehrfarbigen Lötstoppmaske. Stans: „Man kann keine Lötstoppmaske an nur einigen wenigen Stellen auftragen, sondern immer nur auf der gesamten Oberfläche der Leiterplatte. Wenn man also mehrere Lötstoppmasken wünscht, muss man den Vorgang mehrmals durchführen. In einer industriellen Umgebung bedeutet das, die Anlage komplett zu reinigen und mit neuer ‚Tinte‘ zu füllen. Technisch gesehen ist es auch recht schwierig, die verschiedenen Farbbereiche sauber auszurichten und die Toleranzen gering zu halten. Aber es ist eine interessante Erfahrung für die Ingenieure in unseren Werken. Es ist eine Gelegenheit für sie, die industri-

elle Produktionsumgebung zu erkunden und neue Möglichkeiten zu entdecken.“

### Farbe in der Industrie

„Die Arbeit mit Saar bringt Farbe in die etwas graue Welt der Elektronik“, sagt Stans. „Und damit meine ich nicht nur mehrfarbige Lötstoppmasken, sondern ganz allgemein die Offenheit, die Dinge anders zu betrachten und originelle Lösungen zu finden. Die meisten Ingenieure konzentrieren sich ausschließlich auf die Funktionalität. Aber wenn funktionale Lösungen etwas Farbe abbekommen, macht es für alle mehr Spaß. Nehmen Sie zum Beispiel die auf großen Messen verschenkten Demo-Platinen - es sind die langweiligsten Platinen, die man sich vorstellen kann. Warum sollte man sie nicht mit einem künstlerischen Touch interessanter machen? Es wäre das kleine Extra, das Sie von Mitbewerbern unterscheidet. Man muss immer darauf achten, dass das Leben interessant bleibt.“

Drimer: „Dirk liegt viel daran, Menschen für Technik zu begeistern. Wenn man Menschen begeistern und motivieren will, ist es hilfreich, wenn das Objekt, das dabei herauskommt, interessant ist. Es gibt ein ganzes Spektrum von ‚interessant‘ - es muss nicht unbedingt ‚gut aussehend‘ sein. Ich glaube, die Industrie braucht bessere Mittel, wenn sie die Leute für die Elektronik begeistern will. Man muss dazu Werkzeuge zur Verfügung stellen, die nicht einschüchternd sind. Werkzeuge, die eher lenken als erzwingen. Man sollte nicht erwarten, dass die Leute sich gerne durch den Lernprozess dieser unglaublich harten Werkzeuge durchkämpfen. Sie müssen die Barriere beseitigen und den Leuten den Einstieg in die Welt des Elektronikdesigns erleichtern. Auf diese Weise ermöglichen Sie es ihnen, die Dinge zu entwerfen, die sie erschaffen wollen.“

191263-03

### Weblinks

- [1] <https://boldport.com/pcbmode>
- [2] [www.eurocircuits.com/online-smart-tools-services-products/](http://www.eurocircuits.com/online-smart-tools-services-products/)
- [3] <https://shop.pimoroni.com/collections/boldport>



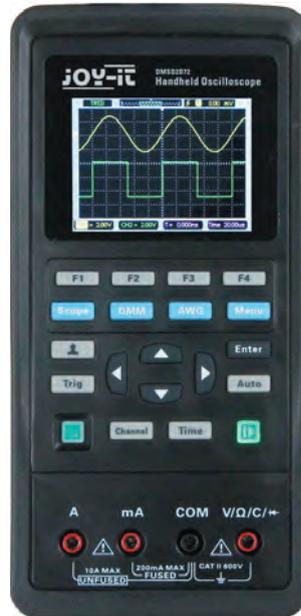
# Willkommen in Ihrem E-SHOP

ELEKTOR EMPFIEHLT



## JOY-iT 3-in-1-Gerät (Oszilloskop + Signalgenerator + Multimeter)

Bei so vielen Funktionen in einem Gerät fragt man sich unwillkürlich, welches wohl die Hauptfunktion ist. Natürlich, es ist das in diesem Fall zweikanalige Oszilloskop mit einer Bandbreite von 70 MHz und einer Abtastrate von 250 MSamples/s (die Hälfte, wenn man beide Kanäle gleichzeitig verwendet). Das sind schon ordentliche Eigenschaften für ein Messgerät, das (für Elektor-Mitglieder) nur 215 € kostet. Hinzu kommt der AWG mit ebenfalls beachtlichen Eigenschaften (Abtastrate 250 MSamples/s) und das Autorange-Multimeter mit separaten Eingängen. Was braucht man mehr?



Harry Baggen  
(Elektor-Autor)

[www.elektor.de/19157](http://www.elektor.de/19157)

## Elektor-Bestseller

1. Elektor  
Schaltungs-Sonderheft 2020  
[www.elektor.de/19140](http://www.elektor.de/19140)



2. Raspberry Pi 4 B (4 GB)  
[www.elektor.de/18964](http://www.elektor.de/18964)

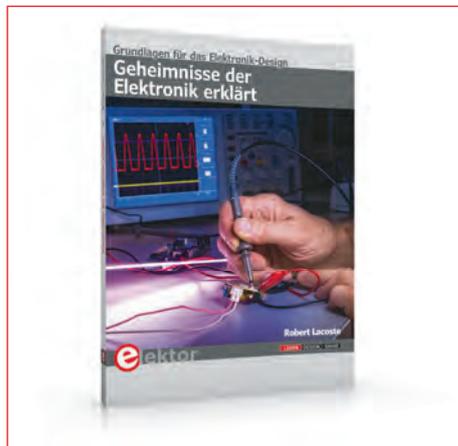
3. Geheimnisse der Elektronik erklärt  
[www.elektor.de/19138](http://www.elektor.de/19138)

4. Elektor SDR-Praxis-Bundle  
[www.elektor.de/19042](http://www.elektor.de/19042)

5. Robotik und Künstliche Intelligenz  
[www.elektor.de/19120](http://www.elektor.de/19120)

6. Elektor SDR Shield 2.0  
[www.elektor.de/18515](http://www.elektor.de/18515)

## Geheimnisse der Elektronik erklärt



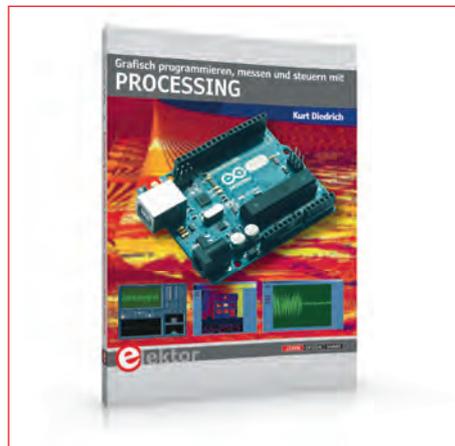
Autor Robert Lacoste, ein hochrangiger Elektronikingenieur, hat für professionelle Elektronik-Zeitschriften eine Serie von Grundlagenartikeln geschrieben, die hier in einem Buch zusammengefasst sind. Wichtige Themen wie Taktgeber, Filter, analoge Signalverarbeitung, digitale Kommunikation und viele weitere werden verständlich erklärt.



Mitgliederpreis: 34,80 €

[www.elektor.de/19138](http://www.elektor.de/19138)

## Grafisch programmieren, messen und steuern mit Processing



Dieses Buch führt den Leser in diese visuelle Programmiersprache ein. Das Buch richtet sich an Leser, die bereits allgemeine Erfahrungen im Umgang mit Programmiersprachen besitzen und wissen, worum es sich bei Strings, Arrays oder Schleifen handelt. Der Autor zeigt anhand vieler kurzer Programmbeispiele, wie einfach es ist, mit Processing auch leistungsfähige Software zu programmieren.



Mitgliederpreis: 34,80 €

[www.elektor.de/19037](http://www.elektor.de/19037)

## IoT Home Hacks with ESP8266



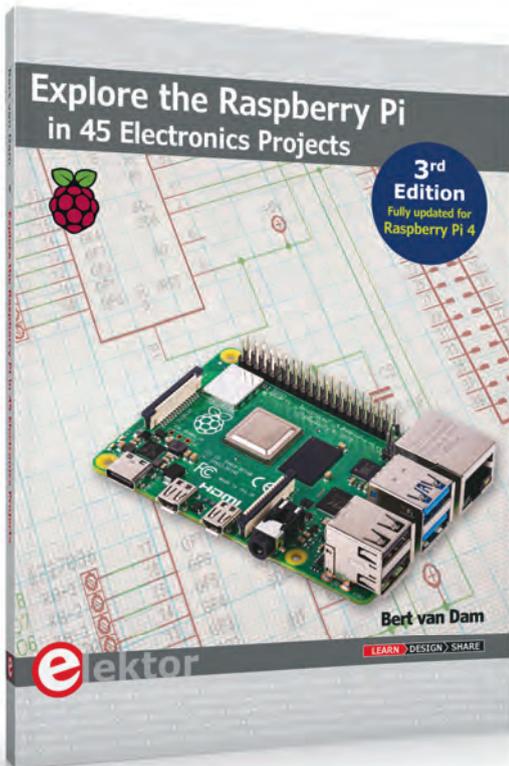
NEU

Es gibt viele sogenannte „Arduino-kompatible“ Plattformen auf dem Markt. Das ESP8266 – in Form des Wemos D1 Mini Pro – ist eines, das wirklich herausragt. Dieser Mikrocontroller verfügt über einen WiFi-Internetzugang und die Möglichkeit eines Flash-Datensystems mit bis zu 16 MB externem Flash-Speicher. Außerdem gibt es ausreichend Ein- und Ausgangspins, PWM, I<sup>2</sup>C und One-Wire. Natürlich können Sie problemlos viele kleine IoT-Geräte bauen!



Mitgliederpreis: 31,46 €

[www.elektor.de/19158](http://www.elektor.de/19158)



## Explore the Raspberry Pi in 45 Electronics Projects (3<sup>rd</sup> Edition)

Die komplett überarbeitete 3. Neuauflage beschreibt 45 spannende und interessante Projekte, wie zum Beispiel ein Wechselblinklicht, eine Motorregelung, Erzeugen und Verarbeiten analoger Signale, ein digitales Thermometer, ein Lichtmesser. Aber auch kompliziertere Projekte wie eine Motor-Geschwindigkeitsregelung, ein Webserver mit CGI (Common Gateway Interface) und Client-Server-Programme werden vorgestellt. Sie können dieses Buch als Projektbuch verwenden und die Projekte nachbauen, um sie dann in der Praxis einzusetzen. Durch die ausführliche Beschreibung mit Schaltplänen und Fotos gestaltet sich der Aufbau auf dem Steckbrett recht einfach.



Mitgliederpreis: 31,46 €  
[www.elektor.de/19190](http://www.elektor.de/19190)

### Robotik und Künstliche Intelligenz

### Elektor Ultimate Sensor Kit

### The State of Hollow State Audio



Dieses Buch ist eine Einführung in das hochaktuelle Gebiet der Robotik. Dabei stehen praktische Anwendungsbeispiele im Vordergrund. Neben den technischen und mechanischen Grundlagen werden die elektronischen Komponenten und Module erläutert. Eine zentrale Rolle spielt dabei der Mikrocontroller.



Mitgliederpreis: 32,80 €

[www.elektor.de/19120](http://www.elektor.de/19120)

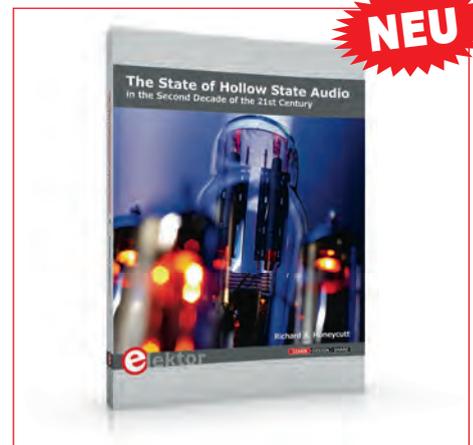


Dieses Bundle besteht aus einem hochwertigen Sensor-Kit und einem umfangreichen Projektbuch. Es wurde speziell entwickelt, um den Einsatz von Sensoren auf mehreren Plattformen wie Arduino, Raspberry Pi und ESP32 zu ermöglichen. Das Bundle enthält insgesamt 40 verschiedene Sensoren, die in mehreren Projekten eingesetzt werden. Alle beschriebenen Projekte werden durch Softwarebeispiele unterstützt.



Mitgliederpreis: 71,96 €

[www.elektor.de/19104](http://www.elektor.de/19104)



Audioröhren erleben derzeit ein Comeback, insbesondere in der Musik- und High-End-Audioindustrie. Dieses (englischsprachige) Buch gibt u.a. die Antwort auf folgende Fragen: Liefern Röhren einen besseren Klang und wie entwirft man Audio-Röhrenschaltungen? Kann man einige der klassischen Röhren-Audiogeräte für moderne Hörsäle und Aufnahmestudios nachbauen? Wie kann man Röhrenverstärker intelligent nach persönlichem Geschmack modifizieren?



Mitgliederpreis: 31,46 €

[www.elektor.de/19170](http://www.elektor.de/19170)

# 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 5. April 2020.**

## Die Gewinner des Hexadokus aus der Ausgabe Januar/Februar 2020 stehen fest!

Die richtige Lösung ist: **C7A13**.

Einen Elektor-Wertgutschein über je 50 € haben gewonnen: Francis Biette, Annie Tigchelaar, David Turnbull, Sabine Lamprecht und Gabi Kirchhof.

Herzlichen Glückwunsch!

1	4			5				6						B	F
8				0		C	E		2						4
		6	E	9		7		1	F	3	2				
		F				3	6	4	5					D	
	7	8			B	F			4	0				3	9
9				C	E	0			7	5	1				A
		D	0	2	6				E	A	F	C			
	A		1										0		6
		2		3									1		7
			5	A	B	9				4	6	8	F		
6					8	C	D			2	F	3			9
		F	C			3	5			0	1			6	2
			A			E	2	F	6					7	
			7	9	5		6			C		D	4	1	
2				7				3	1			4			6
4	C				A					3					5

F	4	D	9	8	C	7	A	1	3	E	5	B	2	0	6
E	1	2	A	B	3	6	9	7	0	8	C	F	4	5	D
B	5	3	C	0	4	D	E	6	9	F	2	7	8	A	1
8	0	6	7	F	1	5	2	4	A	B	D	9	C	E	3
9	E	A	1	6	0	F	3	C	8	5	7	4	B	D	2
2	B	7	F	9	D	8	C	A	1	3	4	0	E	6	5
C	6	4	0	1	A	B	5	9	2	D	E	3	F	7	8
5	3	8	D	E	2	4	7	B	F	0	6	1	9	C	A
3	F	1	E	7	5	C	6	8	B	9	0	A	D	2	4
6	7	0	5	2	8	1	B	D	E	4	A	C	3	F	9
4	2	C	B	D	9	A	F	5	7	1	3	6	0	8	E
A	D	9	8	3	E	0	4	2	6	C	F	5	1	B	7
7	9	F	4	A	6	E	D	3	C	2	B	8	5	1	0
D	8	B	3	C	7	9	0	E	5	A	1	2	6	4	F
0	A	5	2	4	B	3	1	F	D	6	8	E	7	9	C
1	C	E	6	5	F	2	8	0	4	7	9	D	A	3	B

Der Rechtsweg ist ausgeschlossen. Mitarbeiter der in der Unternehmensgruppe Elektor International Media B.V. zusammengeschlossenen Verlage und deren Angehörige sind von der Teilnahme ausgeschlossen.



# Schützen Sie Ihr IP, Ihre Marke und Ihren Umsatz

Sicherheitslösungen, die einfach hinzuzufügen und schwer zu knacken sind

Mit Microchip können Sie nicht nur Ihre Designs, sondern auch Ihre Marke und Ihren Umsatz sichern. Mit zwei Jahrzehnten Erfahrung im Bereich Sicherheit nehmen unsere Experten Ihnen die Angst, Sicherheit in Ihre Designs zu integrieren und machen teure interne Fachkenntnisse überflüssig. Vereinen Sie dieses Know-how mit unseren sicheren Fertigungs- und Versorgungsdienstleistungen, und Sie werden verstehen, warum viele Top-Unternehmen den Experten von Microchip vertrauen, wenn Hilfe während der Entwicklungsphase erforderlich ist.

Von der sicheren Verschlüsselung bis hin zu vertrauenswürdigen Ausführungsumgebungen finden Sie mit unserem umfassenden Angebot hardware- und softwarebasierter Lösungen die Sicherheitsimplementierungen, die Ihren individuellen Anforderungen entsprechen.



**Sichern Sie Ihr Design unter [www.microchip.com/Secure](http://www.microchip.com/Secure)**



Gewinnen Sie  
**75.000€**

Marketing-Paket von  
Elektor und einen Stand  
auf der **electronica**

COMPETE TO  
**LAUNCH**  
YOUR STARTUP AT  
**ELECTRONICA**  
**2020**

Ist Ihr Elektronik-Start-up bereit die  
Lösungen Ihres Unternehmens auf dem  
globalen Markt zu präsentieren?

➤ **DANN BEWERBEN SIE SICH JETZT!**

Besuchen Sie [elektormagazine.de/fastforward](http://elektormagazine.de/fastforward), um die  
Regeln, Bedingungen und Konditionen zu lesen und Ihre  
Bewerbung einzureichen. Viel Erfolg!

**VORTEILE DER FINALISTEN:**

- **Erster Preis = 75.000 €** Marketing-Paket von Elektor, plus  
einen Stand auf der electronica 2022
- **Zweiter Preis = 50.000 €** Marketing-Paket von Elektor
- **Dritter Preis = 25.000 €** Marketing-Paket von Elektor
- Erhalten Sie einen **exklusiven, besonders preisgünstigen**  
**(nur 1.905 €!) Stand (5 m<sup>2</sup>)** auf der electronica 2020
- Erhalten Sie **große Aufmerksamkeit** vor, während und nach  
der electronica 2020. (10.-13.11.2020)



Sign up  
here