

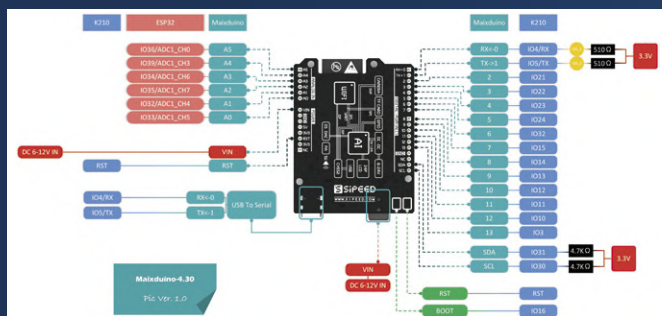
## Open-Network- Wetterstation Mark 2

**Messen, Loggen,  
Streamen  
von Wetterdaten  
zu ThingSpeak und  
openSenseMap**



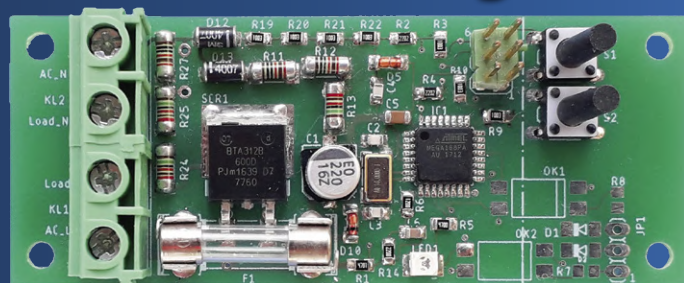
**Neue Serie!**

## KI-Praxis für Einsteiger



**Objekterkennung mit dem Maixduino-Board**

## Universelles Ansteuermodul für Triacs mit ATmega



**Schalten und Dimmen  
unterschiedlicher Lasten**

50 Jahre Elektor – die Projekt-Highlights ✂ Neue Nixie-Uhr mit GPS ✂ CAN-Bus plus Arduino: Überwachung von Solarmodulen ✂ Selbstbalancierender Roboter ✂ ESP32-Multitasking: Software-Timer ✂ Manipulationssicheres Datenpaket ✂ GitHub für Dummies ✂ Entwicklung analoger Elektronik ✂ Das Elektor-Investitionsprogramm ✂ Sigfox: Einrichtung des Dashboards ✂ Kleine Schaltungen ✂ Und vieles mehr!



# 50

## Jahre Elektor

Seit **Mai 1970** gibt Elektor eine deutsche Ausgabe heraus. In diesem Jahr feiern wir **fünf Jahrzehnte** innovativer Elektronik-Projekte, informativer Ingenieur-Tutorials und Fachartikel zu einer Vielzahl von technischen Themen. Schließen Sie sich uns an, um weitere **50 Jahre Elektronik** zu entwerfen, zu teilen und zu verkaufen!

**Elektor-Archiv 1970-2019** (USB-Stick)

**BESTELLEN SIE NOCH HEUTE!**

[www.elektor.de/1970-2019](http://www.elektor.de/1970-2019)



**elektor50.**  
design > share > sell

## Impressum

51. Jahrgang, Nr. 573

Mai/Juni 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](mailto: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](mailto:margriet.debeij@elektor.com)

Büsa Kas

Tel. 0241 95509178

E-Mail: [busra.kas@elektor.com](mailto: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



## Ein gutes Netzwerk – wichtiger denn je!

Noch vor wenigen Wochen hatte ich den Plan, in diesem Editorial von der Embedded World zu berichten – wie immer habe ich dort interessante Produkte entdeckt. Doch die Messe, die Ende Februar stattfand, scheint nun zu einer lange vergangenen Zeit zu gehören. Inzwischen haben wir uns daran gewöhnt, dass wir die Kollegen nur auf Videobildern sehen und vor Super- und Baumärkten anstehen müssen. Dagegen mutet es mir fast schon surreal an, wenn ich im Internet Videos von letztem Jahr sehe – Elektronik-Vorträge vor dicht an dicht sitzendem Publikum oder Musikfestivals mit jubelnden Menschenmengen.

Jedoch – das Leben geht weiter. Mit etwas Kreativität kann man sich auch in diesen Zeiten vernetzen. Einer meiner Kollegen hat kürzlich seinen runden Geburtstag über die Video-Plattform Zoom gefeiert, hier waren auch viele Elektroniker dabei, die sich teilweise aus ihrem Labor zugeschaltet hatten. Ich selbst bin mit einigen Freunden dabei, einen regelmäßigen DJ-Livestream auszusenden, aus Wohnzimmern, aber ordentlich technisch ausgestattet. Und vielleicht haben auch Sie eine Idee, wie Sie sich weiter mit Gleichgesinnten austauschen können, um ein gemeinsames Projekt voranzubringen. Eine Plattform für Elektronik-Projekte können wir unter [www.elektor-labs.com](http://www.elektor-labs.com) bieten – seit diesen Tagen im neuen Gewand, einer optimierten Bedienung und erweiterten Funktionen.

Die deutsche Ausgabe unserer Zeitschrift wird in diesem Monat 50 Jahre alt (siehe Seite 6). In der Branche ist das ein einzigartiger Erfolg, gerade weil viele unserer Leser uns schon seit Jahrzehnten treu geblieben sind. Doch speziell in diesen Zeiten darf das kein Grund sein, uns auf unseren Lorbeeren auszuruhen. Das Gegenteil ist der Fall, bei uns glühen die Drähte und Gehirnzellen. So viel kann ich schon verraten: Wir arbeiten gerade an einigen neuen Angeboten und Medienformaten, um alle Elektroniker noch zielgerichteter informieren zu können.

Bleiben Sie dran und vor allem, bleiben Sie gesund!

Jens Nickel

Chefredakteur Elektor

## Unser Team

Chefredakteur:

Jens Nickel (v.i.S.d.P.) ([redaktion@elektor.de](mailto: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

## Rubriken

- 3 Impressum**
- 17 Startup-Zone**  
Das Elektor-Investitionsprogramm
- 47 Bemerkenswerte Bauteile**  
Sprachsynthesizer TMS0280
- 48 Interaktiv**  
Korrekturen & Updates || Fragen & Antworten
- 56 Von Entwicklern für Entwickler**  
Tipps & Tricks, Best Practice und andere nützliche Infos
- 95 Mein Labor/Projekt**  
Ein Blick ins Allerheiligste aller Elektroniker
- 112 Elektor Ethics**  
Sichere Produkte im IoT-Zeitalter
- 114 Hexadoku**  
Sudoku für Elektroniker

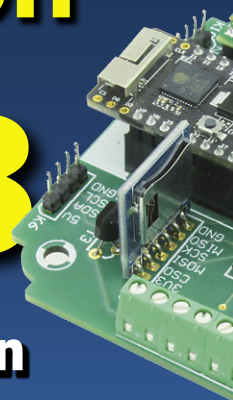
## Hintergrund

- 6 50 Jahre Elektor**  
Projekt-Highlights aus einem halben Jahrhundert Technik-Geschichte
- 12 Elektor wird 50**  
Stimmen zum Jubiläum
- 24 KI für Einsteiger (1)**  
Objekterkennung mit dem Maixduino-Board
- 38 Gewusst wie**  
GitHub für Dummies
- 40 Mein IoT-Button: Der Knopf fürs Netz**  
Teil 2: Prototyping mit Board und Cloud
- 54 Automatische Identifizierung**  
mithilfe der RFID-Lesegeräte von Elatec

# Open-Network- Wetterstation Mark 2

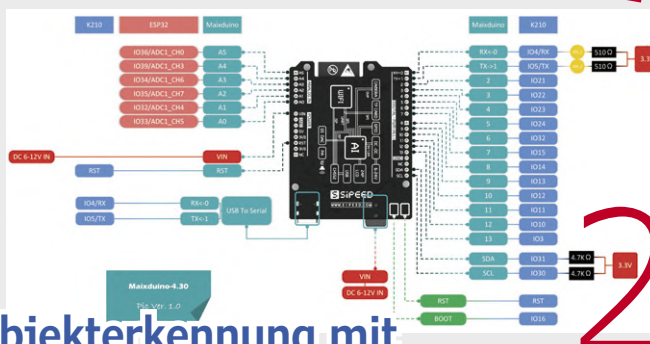
# 18

**Messen, Loggen, Streamen  
von Wetterdaten zu  
ThingSpeak und openSenseMap**



## KI-Praxis für Einsteiger

**Neue  
Serie!**



**Objekterkennung mit  
dem Maixduino-Board**

# 24

- 64 Aller Anfang...**  
Basiskurs für Einsteiger
- 82 Praktisches ESP32-Multitasking**  
Teil 3: Software-Timer
- 96 Review:**  
Elektronische USB-Last Joy-iT HD35
- 104 Entwicklung analoger Elektronik**  
Fall Nr. 1 – Teil 3:  
Optimierung des Vorverstärkerverhaltens  
und Kompromisse
- 107 Review: Labornetzteil PeakTech 6080 A**

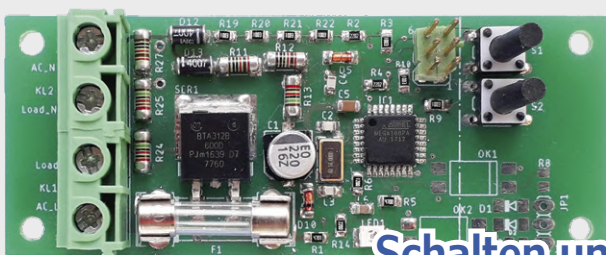
**50 Jahre  
Elektor**

Projekt-Highlights  
aus einem halben  
Jahrhundert Technik-  
Geschichte





## Universelles Ansteuermodul für Triacs mit ATmega



30

Schalten und Dimmen unterschiedlicher Lasten

- 59 BalBot: Ein selbstbalancierender Roboter**  
Moderne Beschleunigungssensoren erleichtern den Bau
- 66 Kleine Schaltungen**  
Neues aus der Elektor-Ideenkiste
- 68 Manipulationssicheres Datenpaket**  
Daten sicher mit der Post verschicken
- 76 Reloaded: Neue Nixie-Uhr mit GPS**  
Ein Engineering-Bulletin für begeisterte Nutzer
- 86 Mit dem Fuchs ins IoT (4)**  
Einrichtung des Dashboards
- 100 CAN-Bus plus Arduino: Überwachung von Solarmodulen**  
Erkennen und Lokalisieren von problematischen Panels

## Vorschau

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

### Aus dem Inhalt:

- Nixie-Bargraph-Thermometer Update
- HV-Stromversorgung mit Kennlinienschreiber
- Home-Info-Display mit Raspberry Pi
- Hausautomatisierung mit Home Assistant
- KI für Einsteiger: Neuronales Netz mit Python
- LED-Booster
- Open-Netzwerk-Wetterstation: Software
- FFT auf dem Maixduino
- PIC programmieren mit Assembler
- NeoPixel Jewel mit ESP8266 steuern
- Erste Schritte mit Node-RED

### Und vieles mehr!

Änderungen vorbehalten.

EElektor Juli/August 2020 erscheint am 2. Juli 2020.



6

## Projekte

- 18 Open-Netzwerk-Wetterstation Mark 2**  
Teil 1: Einführung und Hardware
- 30 Universelles Ansteuermodul für Triacs mit ATmega**  
Schalten und Dimmen unterschiedlicher Lasten
- 50 BASIC für ESP32/ESP8266 (2)**  
Sanduhr mit ESP8266 und Annex WiFi RDS

# 50 Jahre Elektor – die Projekt-Highlights

Markantes und Außergewöhnliches  
aus einem halben Jahrhundert Technik-Geschichte



Bild 1. Titelseite der allerersten in Deutschland erschienenen Elektor-Ausgabe.

Von **Dr. Thomas Scherer**

50 Jahre – eine lange Zeit! Erinnern Sie sich mit uns daran, wie alles anfing und was seitdem alles passiert ist. Schmunzeln und staunen Sie über Schaltungskonzepte, Herausragendes und Besonderes aus der fünf Jahrzehnte langen Historie der immer etwas anderen, eigenwilligen Zeitschrift Elektor!

Als ich gebeten wurde, mich für dieses **goldene Jubiläum** zu erinnern, tauchte geradezu eine Flut von Schaltplänen, Bildern und Ereignissen vor meinem geistigen Auge auf. Aber 50 Jahre? Echt jetzt, sooo lange schon?

Den Anfang macht die Elektor-Ausgabe 1 für Mai/Juni 1970 (siehe **Bild 1**). Und der Knüller in dieser Ausgabe war? Das wird noch nicht verraten. Im **Kasten Leserbrief** finden Sie aber eine kleine Hilfestellung. Nach welchem Verstärker hatte unser Leser erst kürzlich (fast 50 Jahre nach seiner Veröffentlichung) gefragt? Na, klingelt's? Bei mir sofort: Es war der Edwin-Verstärker!

## Leserbrief (vom 14.01.2020)

*Sehr geehrte Damen und Herren,*

*ich bitte um die 1970 in der Zeitschrift Elektor veröffentlichten Unterlagen zum Bau des Verstärkers, da ich eine Reparatur durchführen möchte.*

*Mit freundlichen Grüßen  
Martin Oswald*

## Edwin

Seine 20 Watt Ausgangsleistung waren für diese Zeit enorm. Man bedenke: Die damals noch neue Audio-Norm DIN 4550 verlangte für das Prädikat „HiFi“ nur mindestens 6 W. Man konnte durch Selbstbau also ordentlich sparen. Ein Problem beim Selbstbau war, dass eine „normale“ Audio-Endstufe der Klasse AB einen einzustellenden Ruhestrom hat, der gerne driftete. Das Edwin-Prinzip löste das Problem, wie die Prinzipschaltung von **Bild 2** zeigt. Echt nachbaufreundlich. Aus diesem Grund war der Edwin-Verstärker für Jahre ein Verkaufsschlager; einige in Elektor inserierende Elektronik-Händler boten Bausätze samt Elektor-Platine an. Übrigens



# Goldenes Jubiläum 50

war das auch meine erste Begegnung mit Elektor. Gerade pubertierend wollte ich die 40-W-Version des Verstärkers unbedingt bauen. Unter „technische Kompetenz“ konnte ich damals die mehrfache Lektüre des Buchs „Radiobasteln für Jungen“ [1] samt erfolgreichem Aufbau eines Detektorempfängers (was war ich stolz!) und einer Lichtorgel verbuchen, welche erst der örtliche Radio/Fernseh-Techniker zum Laufen brachte. Mein Wissen reichte nicht und ich brauchte Wochen, bis Edwin endlich verstärkte – doch das Ganze lehrte mich viel Disziplin. Übrigens: Bis dahin waren in der Elektronik die Verstärker-Klassen A, B und AB bekannt. Wie kam es nun zu diesem seltsamen Namen? Gerüchte besagen, dass der Elektor-Entwickler Djerd Veeneman gerade einen Sohn namens Edwin bekam, als er die Idee zu diesem Verstärker hatte...

## Weitere Highlights

Im ersten Elektor-Jahrzehnt erschienen etliche Projekte und Artikel in Elektor, von denen ich schwer beeindruckt war. Wie wäre es zum Beispiel mit einem echten Computer zum Selberbauen?

## Ein Moog-Clone

Ein ganz besonderes Großprojekt war der *Formant*, der dem jedem Musiker bekannten, monophonen Synthesizer von Moog nachempfunden war. Letzterer war ein echter Meilenstein der elektronischen Musik, und jetzt konnten sich ambitionierte Musiker mit technischem Background so etwas wirklich selbst bauen!

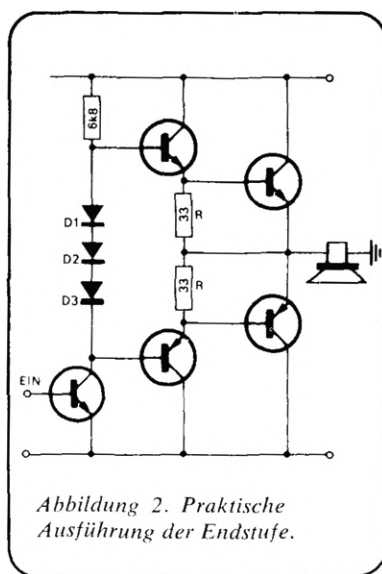


Bild 2. Das Edwin-Prinzip. Nur die beiden „mittleren“ Transistoren haben einen Ruhestrom. Die Spannung an den beiden 33-Ω-Widerständen beträgt recht konstante 0,7 V und der Ruhestrom folglich etwa 10 mA. Bei größeren Strömen übernehmen die beiden Endtransistoren rechts.

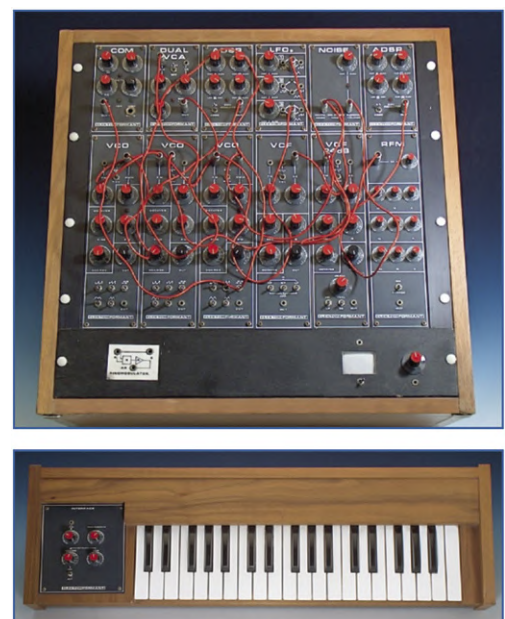


Bild 3. Der *Formant* (hier komplett aufgebaut im schönen Edelholzgehäuse) war ein dem Exemplar von Moog ähnlicher, analoger und monophoner Synthesizer.

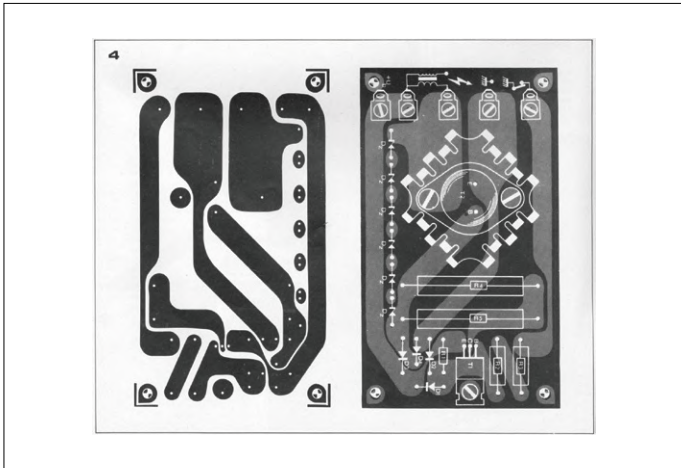


Bild 4. Aus dem Themenheft *Paranormale Elektronik*: Die Netzteil-Platine für die Elektronik zur Kirlian-Fotografie, die elektrische Felder erfasst.

Los ging es mit der September-Ausgabe 1976 und fertig war die Serie erst viele Hefte und 100 Seiten später. Zum Bau war damals unerhörterweise unbedingt ein Digitalvoltmeter erforderlich, wegen der 10 M $\Omega$  Eingangsimpedanz. **Bild 3** zeigt das imposante Musikinstrument im schönen Holzgehäuse. Etliche

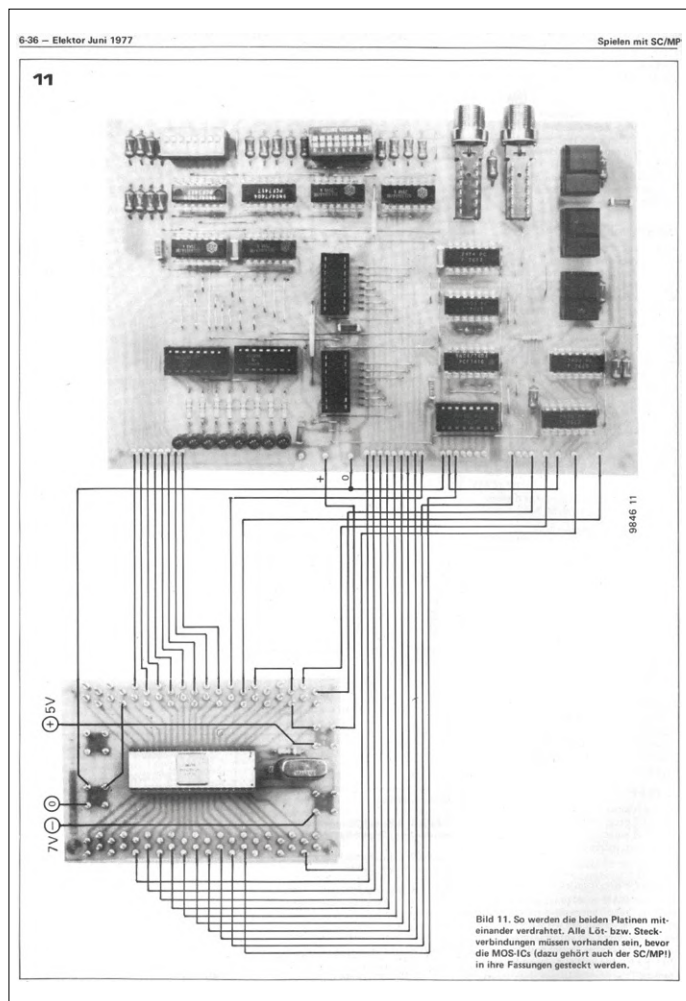


Bild 5. Ein erstes Mikrocomputer-System zum Selbstbau von Elektor Juni 1977, bestehend aus mehreren Platinen. Die untere Platine barg den wertvollen Mikroprozessor SC/MP mit immerhin 1 MHz Takt.

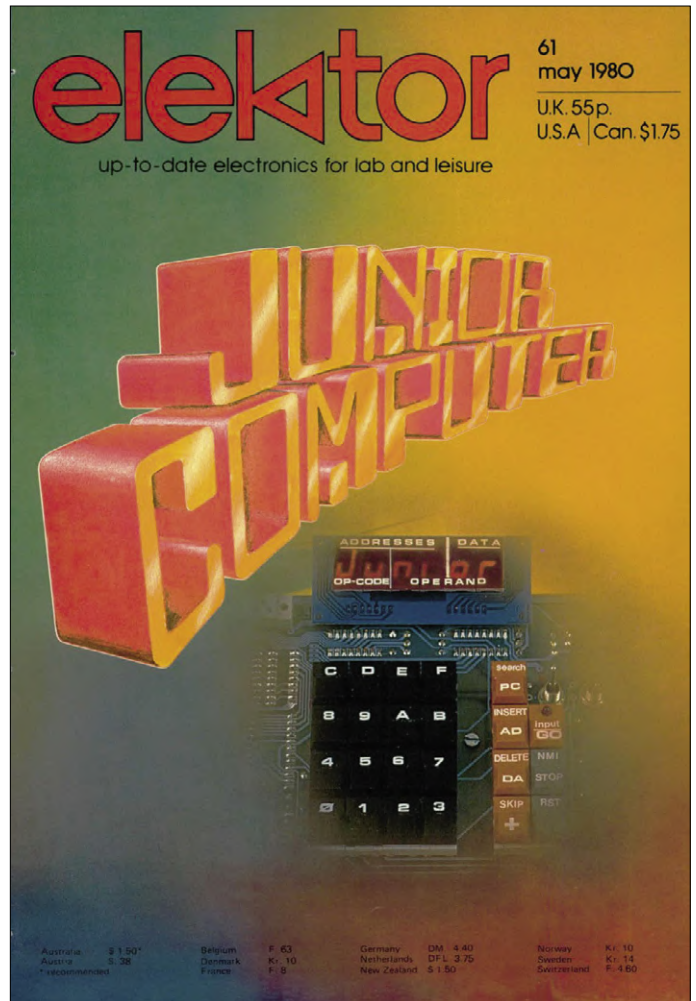


Bild 6. Der Junior-Computer prangt auf dem Titelblatt der englischen Mai-Ausgabe von Elektor.

Webseiten beschäftigen sich heute noch damit [2]. Auf Youtube ist ein interessantes Video mit Klangbeispielen zu finden [3].

### Paranormale Elektronik

Dass in der Elektronik nicht alles mit rechten Dingen zugeht, der Gedanke ist doch jedem spätestens beim Thema HF schon einmal gekommen, Hand aufs Herz. Elektroniker mit Hang zum Außersinnlichen kamen in Elektor Mai 1977 voll auf ihre Kosten. Das Heft-Thema „Paranormale Elektronik“ schlug ein wie eine Bombe. Hier gab es eine schöne kleine Schaltung namens „Klopfgeist“ – ein elektronisches Hilfsmittel zum „Faken“ einer Séance. Außerdem den vieltausendfach nachgebauten und später auch in diversen kommerziellen Erscheinungen auftretenden „Magnetiseur“, eine Art elektronisches Placebo, dessen magnetische Schwingungen gesund machen sollten.

Auch eine Bauanleitung zur Kirlian-Fotografie war enthalten, mit der sich angeblich die menschliche Aura fotografieren lässt. An der Platine des passenden Netzteils in **Bild 4** kann man wunderschön sehen, dass damals die Leiterbahnen mit Letra-set und von Hand gemalt wurden.



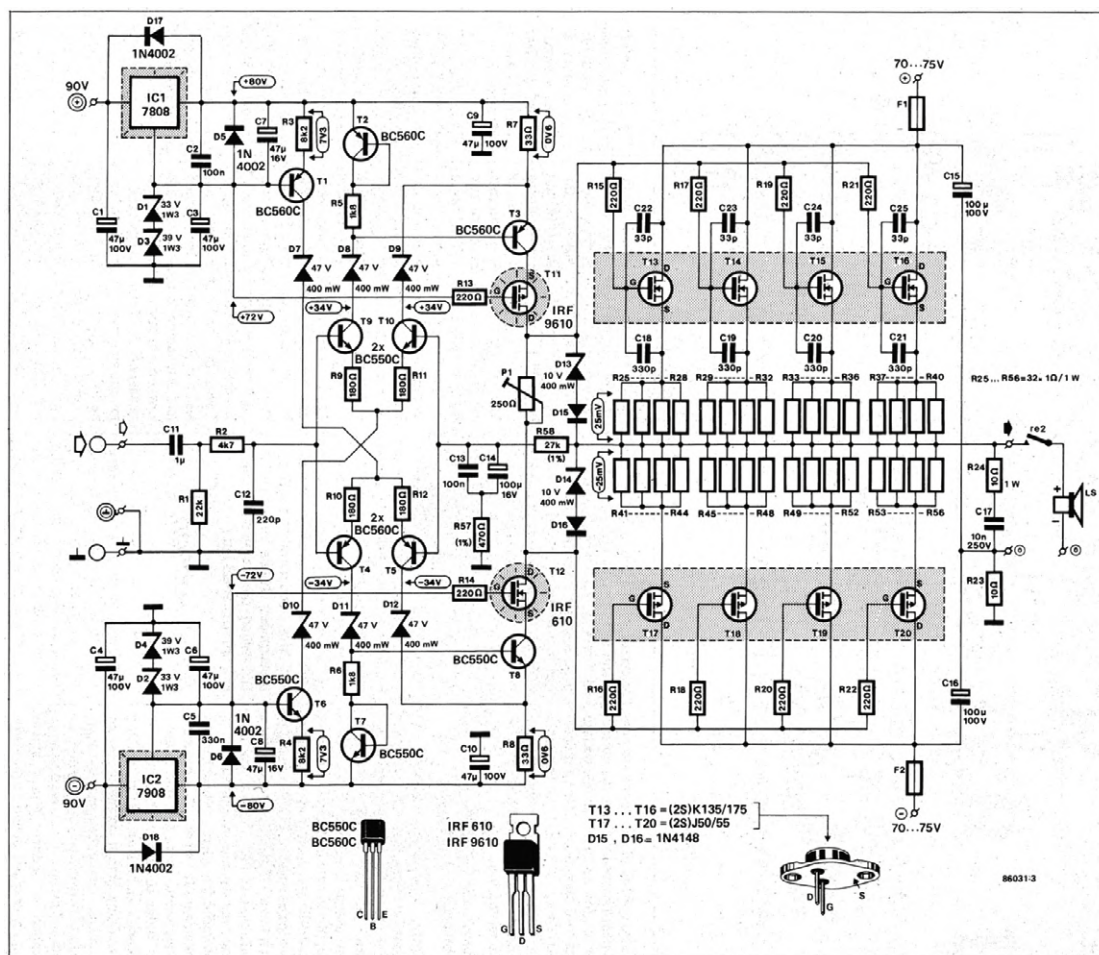


Bild 7. Komplette Schaltung einer Halbbrücke des 1-kW-Verstärkers *Gigant*. So ästhetisch können Schaltpläne sein.

## SC/MP

Einen DIY-Computer gab es schon 1977, rund um den 8-Bit-Mikroprozessor SC/MP (gesprochen „Skemp“) von National Semiconductor – ein Halbleiterhersteller, der damals schon 18 Jahre existierte! Elektor brachte eine mehrteilige Artikel-Serie unter dem Titel „Spielen mit SC/MP“ mitsamt Platinen-Layouts, Stücklisten und Beispiel-Code in Assembler. Sie haben richtig gelesen: Dieses kleine Computersystem mit dem sagenhaften Speicher von 256 Byte RAM bestand aus gleich mehreren Platinen, die miteinander verkabelt werden mussten, wie **Bild 5** demonstriert. Statt Tastatur hatte man Dip-Switches und die Anzeige bestand aus LEDs. Echt Hammer!

## Junior-Computer

Ein Jahrzehnt nach dem Start gehörte ich auch zur Elektor-Redaktion. Genau zu der Zeit tobte sich Kollege Alois Nachtmann mit dem Design eines Selbstbau-Computers auf Basis der CPU 6052 von MOS Technology aus, die auch von Commodores Millionenseller C64 bekannt ist (für den in Elektor etliche Erweiterungs-Projekte veröffentlicht wurden). In der ersten Ausbaustufe verfügte der „Junior-Computer“ genannte Rechner

über 1 KB RAM und für das „OS“ über 1 KB ROM. Welch ein Fortschritt!

**Bild 6** zeigt das Titelblatt der Elektor-Ausgabe vom Mai 1980. Sie haben richtig gelesen: Elektor erschien mittlerweile auch auf Englisch. Und auf Französisch. Der Bau eines eigenen Junior-Computers war mit 450 DM für damalige Verhältnisse unverschämte günstig. Es erschienen natürlich viele Erweiterungen in etlichen Artikeln. Es gab sogar vier Bücher zu diesem Stück Hightech! Und in der letzten Ausbaustufe konnte er sogar BASIC. Auf Messen war der Junior-Computer immer schwer umlagert. Es gibt sogar einen (rudimentären) Wikipedia-Eintrag zu diesem Projekt [4].

## 1 kW HiFi

In Elektor erschienen aber weiterhin auch interessante analoge Schaltungen. Im Mai 1986 wurde der Audio-Verstärker *Gigant* vorgestellt, ein HiFi-Verstärker mit einer für die damalige Zeit gigantischen Leistung von 1.000 W „sinus“ und trotzdem richtig guten Daten. **Bild 7** zeigt eine Hälfte der Brückenschaltung. Für das Kilowatt HiFi wurden insgesamt 16 Power-MOSFETs benötigt – und zwei schwere Trafos mit je 800 VA. Es gab

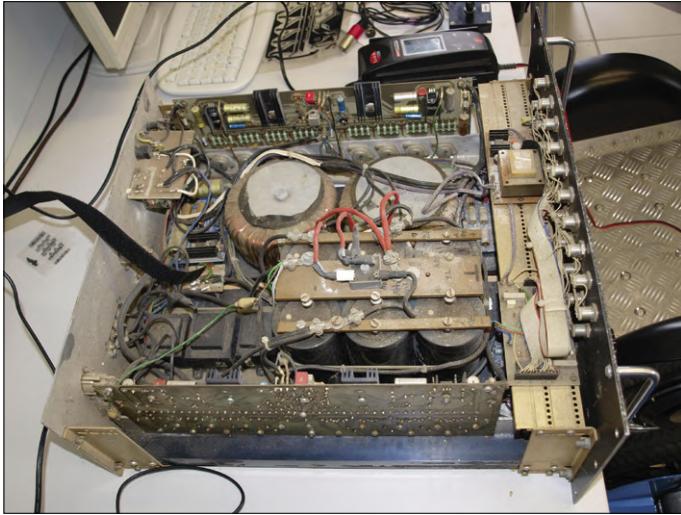


Bild 8. Was für eine Überraschung: Mein Freund Günter Gerold präsentierte „seinen“ Gigant, den er aus einer Ecke seiner Werkstatt ausgegraben hatte. Extrem schwer!

auch schöne Konstruktionszeichnungen für die Mechanik. Mein Freund Günter Gerold überraschte mich übrigens vor einigen Jahren mit seinem Gigant-Nachbau. **Bild 8** zeigt ihn mit dem Staub von Jahrzehnten.

Soviel zu den Anfängen. Und was gab es sonst noch? Bei abertausenden Schaltungen, Hintergrundberichten und Projekten zu jedem möglichen und unmöglichen Thema ist es wirklich schwer, den Überblick zu behalten. Es gibt einfach sooo viel!

### Eine Digitaluhr? Ganz viele...

Wie sieht es aus mit Uhren? Wenn Sie jetzt schmunzeln, weiß ich warum. Schon die Suche auf [www.elektor.de](http://www.elektor.de) bringt 124 Einträge zu Artikeln, die alle irgendwie mit einer Uhr zu tun haben. Besonders imposant ist die *Flipper-Uhr* in **Bild 9**, auch wenn sie moderner ist, als sie aussieht, denn sie entstammt Elektor 3-4/2019.

Aber nur 124 (Digital-)Uhren? Das kann nicht sein! Das wären ja nur etwa 2,5 Uhren pro Jahr. Gefühlt waren es wesentlich mehr!

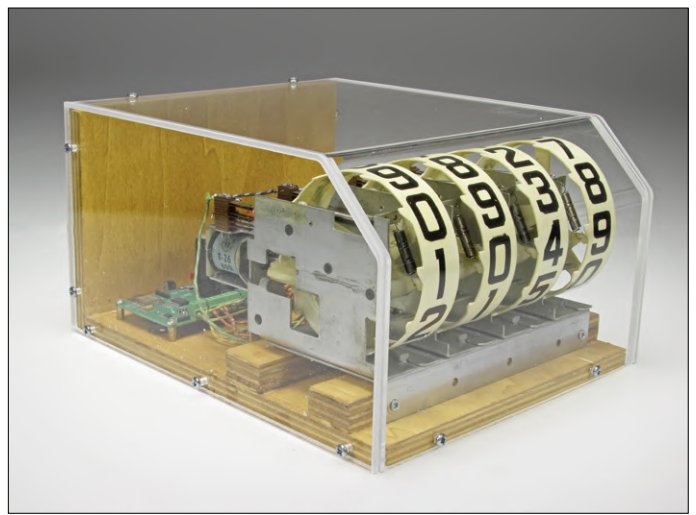


Bild 9. Modernste Retro-Elektronik: Die *Flipper-Uhr* nutzt die elektromechanischen Anzeigen eines alten Flippers.

### Netzteile?

Nach demselben Verfahren erzielt man hier immerhin 199 Treffer = 4 Netzteile pro Jahr. Klingt besser, überzeugt mich aber auch nicht richtig. Denn schließlich muss ja jedes nicht batteriebetriebene Projekt irgendwie irgendwo ein Netzteil haben. Es waren also eher mehr als nur ein explizites Netzteil alle zwei bis drei Ausgaben. Ein besonders interessantes Exemplar ist das *UniLab* in **Bild 10**, ein geschaltetes Labornetzteil mit einer von 0 bis 30 V einstellbaren Spannung bei 3 A (April 2010). Zwei Hefte später gab es praktische Aufbauhinweise und ein passendes Digital-Display.

### Und Radios?

Nach meinen Recherchen wurden über 100 verschiedene Radios und Empfangskonzepte veröffentlicht. Die meiner Ansicht nach interessanteste Entwicklung ist das Projekt *SDR mit AVR* aus Elektor 3/2012. Die Hardware dazu ist in **Bild 11** zu sehen. Es ist geradezu unglaublich, was der langjährige Elektor-Au-

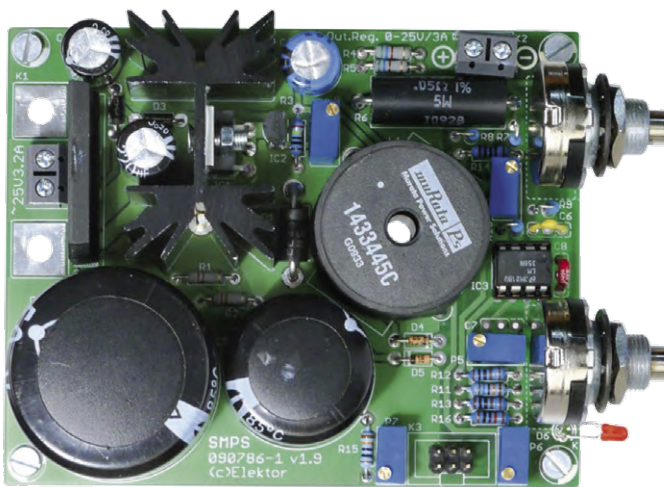


Bild 10. Das geschaltete Labornetzteil *UniLab* mit 0...30 V bei 3 A.

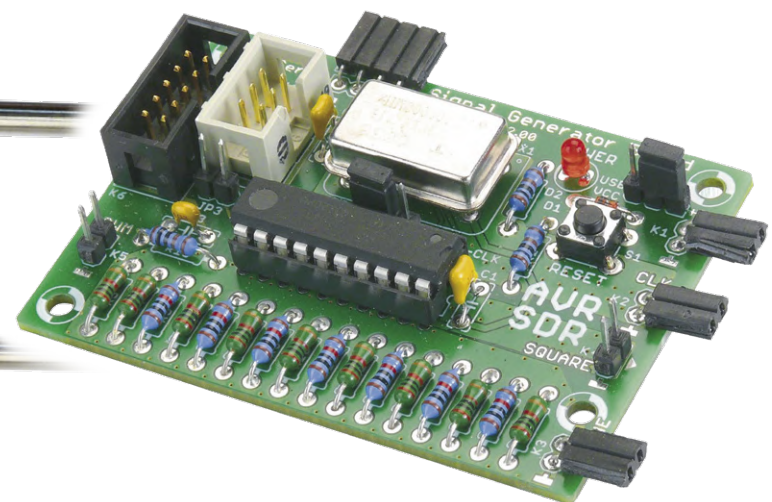


Bild 11. Die Hardware des *SDR mit AVR* besteht im Wesentlichen aus einer kleinen MCU samt Quarz und einem per R2R-Netzwerk gestrickten 8-Bit-D/A-Wandler.



tor Martin Ossmann mit cleveren Algorithmen aus so einem kleinen Mikrocontroller wie dem ATtiny2313 herausholt. Die Serie aus sechs Artikeln macht nicht nur fit in der  $\mu$ C-Programmierung, sondern erlaubt vertiefte Einblicke in die Radio- und Empfangstechnik.

### Last not least: Mikrocontroller!

Ich traue mich und gebe „Mikrocontroller“ in die Suchmaske ein. Die Elektor-Webseite liefert erschlagende 692 Treffer. Solche Chips gibt es seit 1980, also über 17 Treffer pro Jahr und gut 1,5 pro Heft. Also mal mit „Arduino“ versuchen: 132 Treffer. Da es Arduino erst seit 15 Jahren gibt, müsste es wohl jedes Heft mindestens eine Erwähnung geben. Hmm...

War da nicht einmal eine richtig spektakuläre Aktion? Doch, Stichwort: R8C.

Auf der Elektor-Ausgabe 12/2005 (**Bild 12**), also pünktlich zu Weihnachten, gab es in Kooperation mit dem Distributor Glyn ein Hardware-Geschenk an alle Leser, denn es war tatsächlich ein komplettes kleines Entwicklungsboard auf die Titelseite geklebt. **Bild 13** zeigt, wie der kleine 16-Bit-Mikrocontroller R8C/13 auf diesem Platinchen aussah. Sensationell: Die Auflage war bei fast allen Zeitschriftenhändlern schnell vergriffen. In manchen Kiosken soll es sogar zu kleinen Diebstählen gekommen sein, die Mini-Boards wurden von den verbliebenen Heften gerupft. Und das Ganze war kein One-Hit-Wonder: Neben einem Programmierwettbewerb gab es noch 16 weitere Folgeartikel!

### 50+ Jahre

Als Elektor-Mitglied können Sie alle hier angesprochenen Artikel kostenlos herunterladen [5]. Soviel zu den letzten 50 Jahren. Nun schauen Sie sich um: Auch in dieser Jubiläumsausgabe präsentiert Elektor in gewohnter Tradition neue Entwicklungen, interessante Projekte und viel Hintergrundwissen - wie immer als Mischung aus Theorie und Praxis.

Wohin wird sich die Welt der Elektronik in den nächsten 50 Jahren entwickeln? Wir sind gespannt! ◀

200092-01



Bild 12. Auf dieser Titelseite von Elektor 12/2005 war ein leibhaftiges Entwicklungsboard mit 16-Bit-MCU R8C aufgeklebt.

### IM ELEKTOR-STORE

→ Elektor-Archiv 1970-2019 auf USB-Stick  
[www.elektor.de/1970-2019](http://www.elektor.de/1970-2019)

→ Elektor-Mitgliedschaften  
[www.elektormagazine.de/abo](http://www.elektormagazine.de/abo)

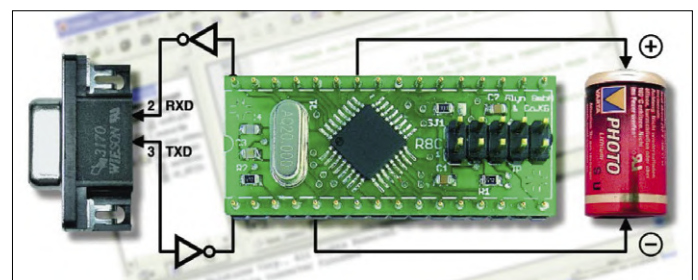


Bild 13. So sah das digitale Weihnachtsgeschenk alias R8C-Platine aus.

### Weblinks

- [1] Heinz Richter, Radiobasteln für Jungen: [www.amazon.de/r/s?k=radiobasteln+für+jungen](http://www.amazon.de/r/s?k=radiobasteln+für+jungen)
- [2] Formant im Synthesizer-Museum: [synthemuseum.com/elektor/eleform01.html](http://synthemuseum.com/elektor/eleform01.html)
- [3] Formant-Demo auf Youtube: [www.youtube.com/watch?v=3XnxSrIjHng](http://www.youtube.com/watch?v=3XnxSrIjHng)
- [4] Wikipedia zum Junior-Computer: <https://de.wikipedia.org/wiki/Junior-Computer>
- [5] Alle Elektor-Ausgaben als PDF: [www.elektormagazine.de/magazine](http://www.elektormagazine.de/magazine)

# Elektor wird 50

Vor 50 Jahren - im Mai 1970 - erschien die erste deutschsprachige Ausgabe von Elektor. In den folgenden fünf Jahrzehnten hat unser Magazin die Entwicklung der Elektronik begleitet und dabei unzählige Elektroniker inspiriert. Viele junge Menschen kamen durch unsere Zeitschrift zu ihrem schönen Hobby und wurden dann Profis: In der Entwicklung, der Ausbildung, dem Marketing oder anderen Bereichen der Branche. Wir haben ein paar Stimmen zum Jubiläum gesammelt!

## Elektor in den Anfangstagen

Das erste Mal gesehen habe ich Elektor in meiner ersten Woche als Technischer Redakteur bei dem einzigen Hersteller von Großcomputern in Deutschland. Ich war sofort begeistert: Ein ganzes Heft voller Schaltungen - einfach genial! Das war im August 1970, und eine derartige Zeitschrift gab es bis dahin einfach noch nicht. Ein Kollege hat mir die ersten beiden Elektor-Hefte geliehen - allerdings nur kurz zum Kopieren. Das Heft Nr. 3 bekam ich bereits als Abonnent und ließ es dann andere kopieren. So hat wohl fast jeder Elektor-Leser damals zur raschen Verbreitung der neuen Zeitschrift beigetragen.

Nach meinem Eintritt in die Elektor-Redaktion im Januar 1974 habe ich dann auch erfahren, wie Elektor nach Deutschland kam. In den Niederlanden gab es schon in den 60er Jahren die von Bob W. van der Horst gegründete Elektronik-Zeitschrift *Elektuur*, mit der er die Elektronik „entmystifizieren“ und populär machen wollte. Mit so großem Erfolg, dass er 1970 den Sprung nach Deutschland wagte. Der Titel *Elektuur* wurde nicht ganz wörtlich mit „Elektor“ übersetzt (sonst hätte es „Elektüre“ heißen müssen). Die Übersetzung der für Heft 1 ausgesuchten *Elektuur*-Artikel übernahmen der erst vor kurzem verstorbene Wolfgang Back, damals noch Student und später beim WDR („Hobbythek“, „Computerclub“) tätig; sowie der erste fest angestellte Elektor-Redakteur Manfred Kalsbach.

Die Initialzündung war dann im Mai 1970 der Versand der ersten Elektor-Ausgabe an 2000 Adressen aus dem Bestand eines Elektronik-Händlers - mit (fast) 2000 Abonnenten als Ergebnis. Danach ging es mit dem Abonnentenbestand unaufhaltsam nach oben. Wolfgang Back nannte einmal die schier unglaubliche Zahl von 90.000 Abonnenten nach eineinhalb Jahren. Von meinem ersten Messeinsatz am Elektor-Stand auf der Hannover Messe 1974 habe ich jedenfalls in Erinnerung, dass so ungefähr jeder dritte Standbesucher Elektor abonnieren wollte.



Mit dem Start von Elektor einher ging die Gründung der Elektor-Verlag GmbH als Tochterunternehmen des *Elektuur*-Verlags, der zuvor schon von Amsterdam nach Beek bei Maastricht verlegt worden war, um die Wege nach Deutschland kurz und die Kosten durch Anmietung einer geräumigen alten Villa in der Provinz (Limburg) niedrig zu halten. Van der Horst war zu diesem Zeitpunkt Eigentümer, Verleger, Herausgeber und Chefredakteur von *Elektuur* und Elektor in Personalunion.

In der deutschen Redaktion waren wir 1974 zu dritt: Manfred Kalsbach als stellvertretender Chefredakteur, Raimund Krings (seit 1973) und ich. Für den Leserservice war die Redak-



tion zuständig. Dafür gab es eine telefonische Fragestunde und trotzdem Unmengen an Leserbriefen. Beides war nicht nur ziemlich anstrengend, sondern auch enorm wichtig für den Kontakt mit unseren Lesern. Ideen gab es immer im Überfluss, und nach den ersten drei Elektor-Jahren auch schon sehr enge Kontakte zu externen Entwicklern, Autoren und Überset-

zern. Extern deshalb, weil wir all das ja auch selbst (intern) gemacht haben.

Bei *Elektuur*/Elektor gab es eine unglaublich kreative Atmosphäre mit viel Freiraum für Individualisten. Wir waren fast alle zwischen zwanzig und dreißig, arbeiteten (gefühlte) am Puls des Fortschritts und hatten zwar noch kein Internet, aber fast alles, was man an Datenbüchern, Datenblättern, Applikationsberichten und Elektronikzeitschriften auf



# Stimmen zum Jubiläum

# 50

Englisch, Deutsch und Niederländisch bekommen konnte. Fast alle Redakteure waren auch Entwickler. Jeder machte mit unterschiedlichen Schwerpunkten beides, und groß geplant wurde auch nicht. Wir hatten unsere (Schaltungs-)Ideen, van der Horst hatte seine Inspirationen und Visionen, und zum Glück hat er lieber stimuliert als organisiert und reglementiert. Wir waren eine Art Chaos Elektronik Club mit kontrolliertem Wildwuchs auf zwei Etagen der Villa. Kontrolliert insofern, als einmal im Monat geerntet werden musste, um das nächste Heft zu füllen. Irgendwie ist immer alles rechtzeitig genug fertig geworden.

Die Villa war schon 1974 stark überbelegt und ein einziges Provisorium, in dem zum Beispiel Platinen in einem ehemaligen Badezimmer geätzt wurden. Ein Jahr später konnte der Betrieb in einen großzügigen Neubau auf einem Nachbargrundstück umziehen, die Familie van der Horst hatte die Villa für sich, und die einzelnen Abteilungen ließen sich neu strukturiert in dem ziemlich schnörkellosen Ambiente unterbringen. Administration und Herstellung (Satz, Grafik etc.) unten, Redaktion und Labor mit Bibliothek in zwei Großraumbüros oben. Es wurde viel investiert, besonders in die Laborausstattung und in neue Kollegen, die jetzt getrennt für Labor und Redaktion eingestellt wurden. War es vorher die Regel, dass man zuerst eine Schaltung entwickelt oder zumindest aufgebaut und getestet hat, um dann selbst einen Artikel darüber zu schreiben, wurde das Entwickeln und Schreiben jetzt zwischen Labor und Redaktion aufgeteilt.

Damit ging aber auch die Ära zu Ende, die „Elektor vor 50 Jahren“ ausgemacht hat. Für mich selbst konnte ich diese Phase noch um etliche Jahre verlängern. Ich war inzwischen Nachfolger von Manfred Kalsbach geworden, der sich selbständig gemacht hatte und einige Jahre später Chefredakteur der „Elrad“ in Hannover wurde. Bei einem Team von mittlerweile vier deutschen Elektor-Redakteuren war es mir zeitlich möglich (und von der Geschäftsführung auch gewollt), große Projekte wie den Formant-Synthesizer weiterhin selbst im Labor zu bearbeiten (redaktionell sowieso) und zwischendurch immer mal wieder eigene Projektideen selbst zu realisieren.

Das alles ist inzwischen sehr lange her, aber beim Schreiben dieses Rückblicks auch wieder erstaunlich präsent. Acht Jahre nach meinem Abschied von Elektor warte ich immer noch gespannt auf das neue Elektor-Heft im Briefkasten. Ich habe mitverfolgt, dass das Labor und die Redaktion aller Elektor-Ausgaben mittlerweile komplett in Aachen angesiedelt sind, und zwar wie vor 50 Jahren ganz nah beieinander. Ich denke, Elektor ist damit auf einem guten Weg in das sechste Elektor-Jahrzehnt.

**Ernst Krempelsauer**

Elektor-Redakteur ab 1974, Chefredakteur von 1985 bis 2011

## Von Elektor inspiriert

Irgendwann Ende des Jahres 1971 bekam ich im Alter von 12 Jahren von einem Bekannten die deutsche Elektor-Erstaussage geschenkt. Endlich gab es eine Zeitschrift voll mit Elektronik. Schon in diesem Heft beeindruckten mich der Gyrator und der Edwin-Verstärker. Ab Januar 1972 hatte ich Elektor abonniert, und wartete jeden Monat voller Spannung auf das neue Heft. Ich denke ein beträchtlicher Teil meiner praktischen „Basteleien“ sind von Elektor inspiriert worden. Der 1974 veröffentlichte Computer-74 inspirierte mich drei Jahre später



zum Eigenbau eines TTL-Computers. Interessant waren auch immer die Grundlagenartikel. Ich denke, damit war ich aufs Elektrotechnik-Studium gut vorbereitet, währenddessen ich das Basteln leider zurückfahren musste. Nach meinem Studium kehrte ich 1989 als Autor zu Elektor zurück. In meinem ersten Artikel ging es um die Dekodierung von RDS-Daten, ein stark mathematisch angehauchtes Thema aus dem Bereich der Fehlerkorrektur. Später habe ich einen 8051-Assemblerkursus geschrieben. Dieses Projekt musste der deutsche Chefredakteur Ernst Krempelsauer gegen den Willen der internationalen Redaktion durchsetzen - der Erfolg des Kurses (mehrere tausend verkaufte Disketten) gab ihm aber recht. Dann haben wir später die Basic-Matchbox herausgebracht, das war sozusagen der Arduino seiner Zeit. Auch eine Reihe Aprilscherz-Artikel habe ich mir mit viel Spaß ausgedacht. Das waren übrigens Zeiten, in denen es in Aachen mindestens fünf gut sortierte Elektronikläden gab... Inzwischen schreibe ich vornehmlich Mikrocontroller- und Software-Artikel. Ich bin gespannt, was die Zukunft für Elektor bringt. Ich bin auf jeden Fall dabei!

**Martin Oßmann**

Professor an der FH Aachen und langjähriger Elektor-Autor

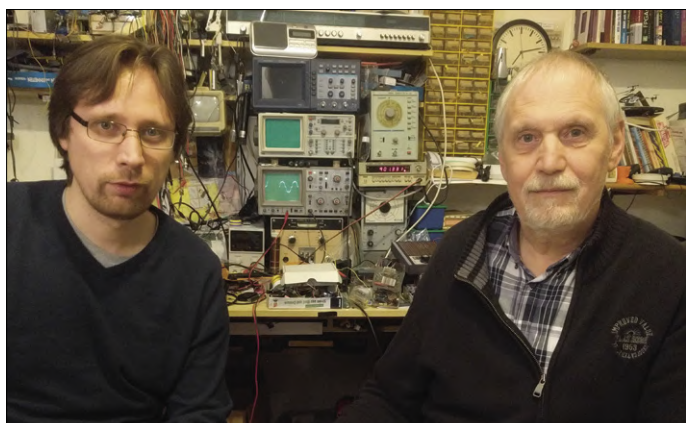
## Zwei Generationen Elektroniker

Meinen ersten Kontakt mit Elektor hatte ich 1973. Da machte mich mein bester Freund auf den Schaltungs-Wettbewerb „Aktion für Aktion“ aufmerksam. Man sollte eine sinnvolle Schaltung mit möglichst geringem Bauteileaufwand entwickeln. Und es sollte damit die Aktion Sorgenkind unterstützt werden. Gerade der minimalistische Ansatz hat mich gereizt. Damals habe ich die Ablenkspulen eines alten Schwarzweiß-Fernsehers abgeklemmt, um sie anders anzusteuern. Das Ergebnis war ein ganz einfaches Oszilloskop mit kleinem Bild auf großem Schirm. Dieses Projekt habe ich dann bei Elektor eingereicht. Das war meine erste Veröffentlichung, und ich war mächtig stolz. Elektor hat mich seitdem begleitet: Der Edwin-Verstärker, die Halbleiter-Schaltungshefte und vieles mehr sind unvergessene Bausteine auf dem Weg in die Elektronik.

Nach längerer Pause war ich dann 1993 wieder mit der Redaktion in Kontakt, weil ich ein Buch veröffentlichen wollte: „PC-Schnittstellen angewandt“. Der damalige Chefredakteur Ernst Krempelsauer hat sich an mein Fernseh-Oszilloskop erinnert. Seine Einladung, weitere Artikel für Elektor zu schreiben, habe ich gerne angenommen. So hat Elektor mit zu meinem späteren Entschluss beigetragen, meinen ersten Beruf als Lehrer an den Nagel zu hängen und mich ganz der Elektronik zu widmen.

**Burkhard Kainka**

Elektor-Autor



In jungen Jahren habe ich meinem Vater gerne beim Basteln über die Schulter geschaut. Natürlich wollte ich schon selbst an den Lötkolben, noch bevor ich wusste, an welchem Ende man ihn halten muss. Doch mein Vater hat mir stets mit Rat und Tat zur Seite gestanden. Als Jugendlicher verlagerte sich mein Interesse in Richtung Informatik und speziell Mikrocontroller. Zusammen haben wir oft mit AVR-Controllern gespielt und Projekte wie einen Zeitzünder für Silvesterknaller, eine Alarmanlage oder eine per IR-Fernbedienung gesteuerte Steckdose realisiert. Elektor hat dabei immer für Hintergrundwissen und Inspiration gesorgt.

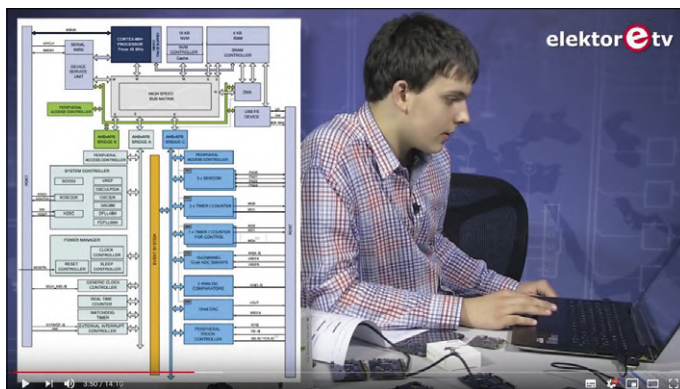
Inzwischen durfte ich auch schon für Elektor schreiben, zuletzt über NodeMCU, ein ESP8266-Entwicklungsboard. Wie zu erwarten war, geht mein Interesse mehr in Richtung Informatik, während das Interesse meines Vaters mehr in der analogen Elektronik liegt. Oft können wir uns ergänzen und gegenseitig helfen. Und wir beide lesen weiterhin Elektor, weil wir dort interessante Artikel zu beiden Themenbereichen finden.

**Fabian Kainka**

Elektor-Autor

## Elektor live! mit 14

Alles fing in der Bibliothek einer badischen Kreisstadt an. Kurz nachdem ich im Alter von zehn Jahren nach Deutschland kam, habe ich mich dort regelmäßig auf dem obersten Stockwerk – Abteilung „Fachzeitschriften“ – aufgehalten. Einerseits gab es in einem der Kästen die Elektor-Ausgaben zur Ausleihe, andererseits auch die Elektor-Schaltungssammlungen als Bücher. Zur „Elektor live!“-Veranstaltung in Darmstadt habe ich mich mit 14 Jahren angemeldet und dort Jens Nickel, den Elektor-Chefredakteur, bei einem Gespräch über eine Sourcecode-Veröffentlichung kennengelernt. Kurz danach tauschten wir schon Mails über unterschiedliche, gemeinsame Projektideen aus.



Als Erstes durfte ich einen zweiseitigen Report über die Embedded World 2014 schreiben. Dieser Artikel sollte in meiner bisherigen Laufbahn auch gleich der unter höchstem Druck geschriebene bleiben! Das Heft war fertig gesetzt und kurz vor der Druckmaschine, es ging gefühlt um Stunden. Eine zu späte Abgabe und die zwei Seiten wären leergeblieben...

Es folgten viele Artikel mit Bauteil-Tipps, Marktübersichten, Programmierkursen und mehr. Gut erinnern kann ich mich noch an meinen Besuch im Elektor-Schloss in Limbricht (Niederlande). Innerhalb von wenigen Stunden mussten im Elektor-Studio Videos über den SAM D20 und WINC1500 produziert werden. Ich hatte das mit meinen jungen Jahren noch nie gemacht und der erste Lauf musste gleich stimmen. Patrick Wielders von Elektor und ich gönnten uns keine Pause! Das ungeplant stundenlange Material wurde dann aufgrund der Länge aufgeteilt und dank des guten Teamworks zu einem YouTube-Erfolg.

Ich habe auch dank Elektor damals gelernt, unter Zeitdruck zu arbeiten - für meine berufliche Laufbahn in der Elektronikbranche ein Gewinn. Deadlines sind manchmal schmerzhaft, aber überlebensnotwendig. Das Durchhaltevermögen samt schlaflosen Arbeitsnächten wird dadurch nur bestärkt, genauso wie der Perfektionsdrang durch die Vielzahl an Korrekturrunden. Und es ist immer wieder schön, tolle Rückmeldungen von Lesern zu bekommen und ihnen bei den weiterführenden Fragen und Projekten zu helfen.

Besonders gefreut hat es mich, dass Elektor immer mehr online veröffentlicht. Darüber hinaus steht Elektor als Unternehmen mit internationalen (Business-)Ausgaben sowie Kooperationen mit großen Partnern





und Veranstaltungen (z.B. Startup-Wettbewerb) prima da, so dass man sich um die Zukunft keine Sorgen machen muss. Aktuell liegt die neueste Elektor-Ausgabe übrigens immer noch im gleichen Holzkasten der Bibliothek, aber zwischen ganz anderen Nachbarzeitschriften. Auf weitere 50 Jahre Elektor als Fels in der umstürzten Brandung!

## Viacheslav Gromov

Elektor-Autor und Startup-Gründer

## 50 Jahre Elektor – Eurocircuits gratuliert

Seit 50 Jahren auf dem deutschen Markt präsent – das darf gefeiert werden!



Mit meinen 57 Lenzen war ich 1969 noch etwas zu jung, um die erste Ausgabe von Elektor zu lesen. Doch in meiner Gymnasialzeit in den 1970er Jahren und später während meiner Ingenieurausbildung lag das Elektromagazin Elektor immer irgendwo herum – in der Schule, der Uni oder bei mir zu Hause. Elektor hatte seinerzeit fast ein Monopol in den Schulen und Universi-

täten in Belgien und Holland.

Wir von Eurocircuits haben 1999 zu Elektor gefunden. Für uns war es ein großer Vertrauensbeweis und eine Ehre, die Platinen für die Baupakete fertigen zu dürfen und später die komplett bestückten Baugruppen. Seit 2009 organisieren wir für Elektor den [www.elektorPCBservice.com](http://www.elektorPCBservice.com) als Leiterplatten- und Baugruppenangebot für professionelle Elektor-Kunden und [www.elektorpcb4makers.com](http://www.elektorpcb4makers.com) für die Maker unter den Lesern. Man kann sagen, dass wir mehrfach verbandelt sind, was uns sehr gefällt.

Zum Jubiläum wollen wir aber nicht nur gratulieren, sondern unseren Respekt zollen. In den letzten 10 Jahren hat sich Elektor besonders erfolgreich entwickelt und sich eine wichtige Position unter den Fachmedien in Europa erarbeitet. Zu diesem Wachstum beigetragen haben sicher auch die neuen Unternehmungen wie „Fast Forward 2020: The Start-up Platform Powered by Elektor“. Projekte wie dieses unterstreichen die Innovationskraft von Elektor und bilden die Basis für den Erfolg und die Zukunft dieses Unternehmens. Eurocircuits und ich wünschen noch 50 weitere Jahre Erfolg!

## Dirk Stans

Geschäftsführer Eurocircuits

## reichelt elektronik gratuliert zu 50 erfolgreichen Jahren

Elektor ist sowohl für viele Mitarbeiter als auch Kunden von reichelt elektronik eine Pflichtlektüre – damals wie heute, für die Inspiration und Informationen rund um Elektronikprojekte. Da wir selbst schon das eine oder andere Projekt privat umgesetzt haben, wissen wir, was Elektor-Leser lieben, und welche Komponenten und Werkzeuge sie dafür brauchen.



Auch reichelt elektronik hatte im vergangenen Jahr sein 50-jähriges Jubiläum – daher ist es nicht verwunderlich, dass reichelt und Elektor schon lange miteinander verbunden sind. Denn die spannenden Themen und Projekte, die Elektor in seinem Magazin behandelt, lassen sich ideal mit dem breiten Spektrum an preisgünstigen Komponenten von reichelt realisieren.

Inhaltlich schätzen wir an Elektor besonders die interessante Themenauswahl und die vielen Projekte, die dort detailliert vorgestellt werden. Diese zeichnen Elektor als eines der wenigen deutschsprachigen Fachmagazine für Maker besonders aus. Dank seiner europaweiten Verbreitung und der aktiven Community ist die Arbeit mit Elektor facettenreich und spannend.

Wir gratulieren zu der 50-jährigen Unternehmensgeschichte, auf die Elektor nun schon zurückblicken kann, und hoffen, dass sich diese in den kommenden Jahren ebenso erfolgreich fortsetzen wird. Auf viele weitere spannende Kooperationen!

## Ulf Timmermann

Geschäftsführer reichelt elektronik GmbH & Co. KG

## Herzlichen Glückwunsch von Arrow Electronics

Sicherlich haben die meisten Bastler, Techniker und Ingenieure bereits einmal ein Projekt von Elektor realisiert. Auch ich habe als Teenager erste Schaltungen nachgebaut, etwa einen MP3-Player, als diese als Fertiggerät noch vergleichsweise teuer waren.

Somit hat Elektor sicher viele berufliche Werdegänge beeinflusst und den Weg in ein technisches Studium begleitet. Auch heutzutage freut sich jeder Entwickler in unserem Büro, sobald die neue Elektor-Ausgabe erscheint, und mit großer Neugier schauen alle Kolle-



ginnen und Kollegen nach den neuesten Projekten. Denn Elektor bietet immer wieder Projekte auf Basis der neuesten Technologien, die für Hobbyisten und Profi-Entwickler gleichermaßen interessant sind. Wir von Arrow freuen uns auf weitere 50 Jahre mit vielen interessanten und spannenden Projekten!

## Christian Reuter

Training Officer,  
Arrow Electronics



## 50 Jahre erstklassiger Journalismus

Würth Elektronik gratuliert Elektor auf das Allerherzlichste zum fünfzig-jährigen Bestehen. Gleichzeitig bedanke ich mich für eine langjährige, professionelle Zusammenarbeit. In der dynamischen Elektronikbranche sind 50 Jahre eine Ewigkeit. Wer es so lange schafft, ganz oben zu bleiben, versteht sein Geschäft

wirklich. Elektor ist über all die Jahre mit seinen Themen und der Berichterstattung am Puls der Zeit geblieben, hat die Branche und ihre Entwicklung begleitet und so auch mitgeprägt.

Elektor berichtet aus der Perspektive von Elektronikentwicklern und Hobby-Elektronikern. Beiden Gruppen dabei stets informative, lehrreiche und auch unterhaltsame Neuigkeiten, Artikel und Reviews auf einem gleichbleibend hohen Niveau zu bieten, ist eine bemerkenswerte Leistung. Auch in unserem Unternehmen hat Elektor eine ansehnliche Fangemeinde – besonders in den Entwicklerteams. Vergesse ich es, mein Exemplar weiterzugeben, dauert es nicht sehr lange, bis der Ruf von Kollegen nach der aktuellen Ausgabe laut wird. Meine Bitte an Verlag und Redaktionsteam: Macht weiter so! Würth Elektronik wünscht Euch alles Gute und weiterhin viel Erfolg.

**Mathias Kalmbach**

Marketing Executive der Würth Elektronik eiSos Group



## Elektor und der R8C

Auf der sehr dünn besuchten embedded world 2020 trafen Jens Nickel und ich uns wie üblich, für ein kurzes Gespräch über die neuesten Entwicklungen. Das Gespräch kam neben dem neuen Corona-Virus und seinen Auswirkungen auch auf das Thema 50 Jahre Elektor und die Frage „seit wann bist Du eigentlich Elektor-Leser?“

Den Grundstein legte 1993 ein Schülerpraktikum in einem Elektroniklabor. Hier standen die Elektor-Ausgaben von 1970 bis 1993, fein säuberlich aufgereiht. Aus diesen durfte ich ein Projekt auswählen. So habe ich dann den Schaltplan eines Soundgenerators abgepaust, belichtet und meine erste Platine geätzt. Ab da war ich vom Elektronikvirus erfasst, und schließlich stimmten meine Eltern einem Elektor-Abo zu. Im Dezember 2005 hatte ich als E-Technik-Ingenieur bei der Firma GLYN die einmalige Chance, mit Elektor für Chaos im Zeitschriftenhandel zu sorgen. Heft Nr. 420 mit dem von uns entwickelten, aufgeklebten R8C/13-Board wurde der Renner; die gesamte Auflage war ausverkauft! Heute stehen in meinem privaten Labor die Jahrgänge 1993 bis 2020, lückenlos. Die Zeiten ändern sich, und so hat sich Elektor über all die Jahre auch gewandelt. Ausgehend von reinen Hardwareschaltungen ist das inhaltliche Spektrum nun viel breiter geworden; Themen wie WiFi, LoRa, Bussysteme sowie Embedded-Software sind hinzugekommen. Ich hoffe, dass Elektor auch in den nächsten Jahrzehnten ein fester Anker in der schnelllebigen Elektrotechnik bleibt!

**Alexander Pokorny**

Applikations-Ingenieur Glyn GmbH & Co. KG

## Leserbrief: Der Edwin-Verstärker

Seit Elektor Heft 2 bin ich Leser von Elektor. In Heft 1 wurde der große Edwin-Verstärker veröffentlicht, und in Heft 2 kam dann der kleine Nachfolger mit 20 W. Ich habe die Platinen bestellt und die Bauteile in Berlin beim legendären Atzert-Radio gekauft (mit der U-Bahn unter der DDR durch nur mit einem Halt am Alexanderplatz - im Intershop Zigaretten kaufen!). Hinzu kamen noch das Netzteil und ein Chassis. Die notwendigen Alubleche gab es gleich gegenüber im Modellbau-Laden. In den 70ern waren Schieberegler für Lautstärke, Balance und Klang gerade modern. Den schicken Netzschalter habe ich als Ersatzteil von einem Schaub-Lorenz-Receiver gekauft (das Gerät war jenseits meines verfügbaren Budgets - ich war ja noch Schüler bzw. anschließend Studienanfänger). Leider gab es beim ersten Aufbau Probleme. In den folgenden Elektor-Heften folgten dann aber noch einige Updates bzw. Verbesserungen. Als das alles nichts half, habe ich zwei komplett bestückte Baugruppen bestellt, einen richtigen Kabelbaum geknotet (hatte ich gerade im Praktikum bei der AEG gelernt) und alles sehr ordentlich in das Chassis eingebaut. Doch auch diese Version funktionierte nicht richtig. Um den Fehler zu analysieren hat mir mein Vater ein Hameg-Oszilloskop mit 20 MHz Bandbreite und richtigen 10/1-Testköpfen gekauft. Der sehr einfache Röhren-Sinusgenerator musste nach kurzer Zeit durch einen modernen HP-Funktionsgenerator 3311A für 1000 DM ersetzt werden (immerhin funktioniert er noch!). Damit konnte ich dann doch wenigstens das Problem analysieren - der Verstärker schwingt! Ich habe an dem Projekt drei Jahre in den Schulferien gebastelt und einige Fachartikel gelesen (Internet gab es ja noch nicht) - der „Tietze/Schenk“ wurde mein bester Freund. Ich habe in der Zeit sehr viel über Verstärker, Netzteile und Elektronik gelernt. Funktioniert hat der Edwin nie. Nach 3 Jahren und sehr viel (Lehr-)Geld habe ich es dann aufgegeben und eine Endstufe mit zwei STK35-Modulen gebaut. Sehr lange war dieser Verstärker aber auch nicht im Einsatz, weil ihn dann - endlich - ein Kenwood-Verstärker ersetzt hat (der ab und zu noch heute in Betrieb ist).

Fazit: Ich habe sehr viel Zeit und Geld für mein Elektronikwissen investiert und so ganz nebenbei gelernt, das Ausdauer zwar eine Tugend ist, aber eine Trennung sehr befreiend sein kann. Elektor hat mir auf diese Weise lebenswichtige Erkenntnisse vermittelt, die über die reine Elektronik weit hinaus gehen. Meine nächsten wesentlichen Erfahrungen machte ich dann während meiner ersten beruflichen Anstellung bei Createc - aber das ist dann eine andere Geschichte. Ist es nicht schade, dass nur wenige der genannten Firmen noch existieren?

**Florian Jankowsky**



### IM ELEKTOR-STORE

**Markantes und Außergewöhnliches aus einem halben Jahrhundert Technik-Geschichte**

→ Elektor-Archiv 1970-2019 auf USB-Stick  
[www.elektor.de/1970-2019](http://www.elektor.de/1970-2019)

200116-01





# Start-Up Update: Das Elektor Investitionsprogramm

Vom **Elektor Start-up Team**

Elektor fördert und stärkt Elektronik-orientierte Start-ups (Hardware, Software und technische Dienstleistungen) durch Veranstaltungen, Wettbewerbe und Geschäftslösungen. Das Elektor-Investitionsprogramm [1] ist ein neuer Service sowohl für junge Unternehmen als auch für Investoren, die sich in der Elektronikindustrie erfolgreich engagieren möchten.



Für Start-ups oder expandierende Jungunternehmen kommt eine Zeit, in der Bootstrapping keine Lösung mehr ist. Wenn Sie Glück haben, haben sie bereits einige zahlende Kunden, aber sie brauchen Geld, um das Unternehmen auf das nächste Level zu bringen. Vielleicht müssen das Team erweitert, neue Märkte erschlossen oder die Entwicklungsarbeit ausgeweitet werden. Der am nächsten liegende Schritt ist das Gespräch mit Investoren. Als Gründer könnten Sie sich fragen: Bin ich bereit, Investoren anzusprechen? Sind mein Businessplan und mein Pitch Deck dafür bereit? Werden mich die Investoren ernst nehmen? Mitglieder des Elektor-Investitionsprogramm-Teams werden Sie

nicht gut genug. Deshalb glauben wir, dass wir Investoren Zeit sparen können, indem wir sie mit vielversprechenden Technologie-Startups, die dem jeweils möglichen Investitionsumfang entsprechen, aus unserem globalen Netzwerk verbinden.

### Vorteile für Start-ups

Sind Sie Mitglied eines Start-ups oder sogar eines Scale-ups, das eine Finanzierung benötigt? Durch die Teilnahme am Elektor-Investitionsprogramm können Sie sinnvolle Maßnahmen ergreifen, um Ihre Chancen auf eine Finanzierung zu verbessern. Klingt nach einem guten Geschäft, oder? Hier ist unser Angebot:

- Persönliche Hilfe zur Verbesserung Ihres Pitch Decks und Ihres Geschäftsplans
- Geschäftsanalyse zur Erstellung eines Profils Ihres Unternehmens
- Die Möglichkeit, mit zu Ihnen passenden Investoren aus unserem Netzwerk in Verbindung zu treten.

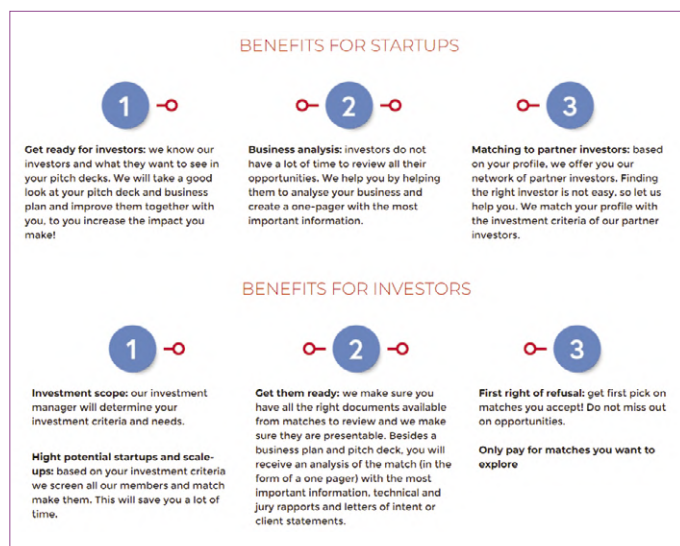
### Vorteile für Investoren

Das Elektor-Investitionsprogramm kann Investoren helfen, innovative Unternehmen zu finden. Neben der Durchführung erfolgreicher Start-up-Wettbewerbe sucht unser professionelles Start-up-Team weltweit nach vielversprechenden Unternehmen. Unternehmen, die dem Programm beitreten, werden von unseren Experten geprüft und beurteilt, so dass wir Investoren mit vielversprechenden Leads unterstützen. Wir bieten:

- Zugang zu technischen Start-ups und Scale-ups mit hohem Potenzial, die Ihrem Investitionsvolumen entsprechen
- Korrekte Dokumentation, professionelle Geschäftsbeziehungen und eine Geschäftsanalyse der Übereinstimmungen mit den Start-ups

durch dieses schwierige Vorhaben führen. Wir können Ihnen bei Ihrem Businessplan helfen, Sie bei der Vorbereitung Ihres Pitch Decks unterstützen und Sie mit Investoren aus unserem Netzwerk zusammenbringen.

Warum haben wir das Elektor-Investitionsprogramm ins Leben gerufen? Wir führen seit einigen Jahren Wettbewerbe für Start-ups durch [2] und wir haben in dieser Zeit mit vielen vielversprechenden Start-ups zusammengearbeitet. Die Entwicklung eines Programms, das Start-ups bei der Suche nach Finanzierung und Partnern unterstützt, war für uns ein logischer nächster Schritt. Wir wollen auch Investoren unterstützen. Wir wissen, dass Investoren jedes Jahr Dutzende von Pitch Decks erhalten. Es erfordert viel Zeit, sie alle zu lesen, und die meisten sind



Möchten Sie am Elektor-Investitionsprogramm teilnehmen oder mehr darüber erfahren? Besuchen Sie die Website des Investitionsprogramms [1] oder senden Sie noch heute eine E-Mail an Arthur Lampe (Elektor-Investitionsprogramm-Manager) unter [arthur.lampe@elektor.com](mailto:arthur.lampe@elektor.com). Viel Erfolg! ◀

200112-03

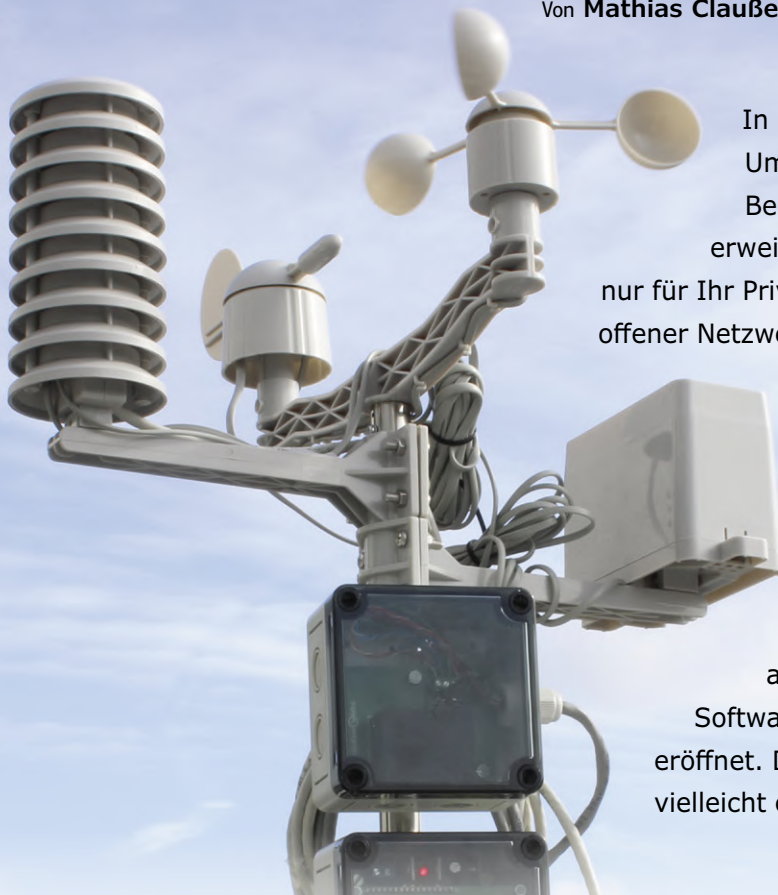
[1] [www.elektormagazine.com/widgets/investment-program](http://www.elektormagazine.com/widgets/investment-program)

[2] [www.elektormagazine.de/select/fast-forward-award](http://www.elektormagazine.de/select/fast-forward-award)

# Open-Network-Wetterstation Mark 2

## Teil 1: Einführung und Hardware

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



In diesen Zeiten des Klimawandels und der globalen Umweltsorgen können auch Sie einen positiven Beitrag leisten, indem Sie eine leistungsfähige und erweiterbare Wetterstation aufbauen. Und das nicht nur für Ihr Privatvergnügen, sondern auch als Teil globaler, offener Netzwerke wie ThingSpeak und openSenseMap. Die in diesem mehrteiligen Artikel beschriebene Station wurde mit Blick auf Flexibilität bei der Messung und Aufzeichnung von Wetterdaten entwickelt. Sie vereint einen minimalen Strombedarf mit hoher Zuverlässigkeit im Außenbereich. Kern des Projekts ist ein ESP32-Modul, an das handelsübliche Sensoren angeschlossen sind und das mit einer freien Software betrieben wird, die Ihnen viele Möglichkeiten eröffnet. Das Projekt lässt wenig zu wünschen übrig, außer vielleicht etwas Sonnenschein und eine leichte Brise!

### Eigenschaften

- 3-Platinen-System: Hauptplatine, Sensor-Anschlussplatine, Staub/Partikel-Platine
- Bedrahtete Bauteile und kommerzielle Module
- Sensoren (Basissystem): Temperatur, relative Feuchtigkeit, Luftdruck, Staub/Partikel
- ESP32-basiert (Pico-Kit)
- I<sup>2</sup>C-Bus
- 12-V-(Solar-)Batterie mit automatischer Abschaltung bei niedriger Spannung
- Platinenanschlussklemmen für die (meisten) Anschlusskabel
- Wasserdichte Gehäuse (IP66) für Haupt- und die Staub/Partikel-Sensorplatine
- Open-Source-Software
- Kompatibel mit ThingSpeak, openSenseMap
- Mit kundenspezifischen Sensoren erweiterbar
- RFM95-LoRa-Erweiterung (optional, in Entwicklung)
- ESPHome- und HomeAssistant-kompatibel
- zugreifbar über WLAN
- SD-Karte für lokale Steuerung

Obwohl der Titel auf „Mark 2“ endet, handelt es sich bei diesem Projekt um eine völlige Neugestaltung der ersten Version der ESP32-Wetterstation [1], die für den Neubau nicht mehr empfohlen wird.

Die Leiterplatten für das vorliegende Projekt wurden so konzipiert, dass sie zu den mechanischen Teilen des im Elektor-Store erhältlichen Wetterstations-Kits passen, das eine Halterung und Sensoren für Windgeschwindigkeit, Windrichtung, Regen, Temperatur und relative Luftfeuchtigkeit umfasst.

Die Elektronik ist auf drei Platinen aufgeteilt:

- die Hauptplatine 191148-1, die in einem wasserdichten Gehäuse für



die Installation im Freien geeignet ist, aber relativ nahe dem Haus montiert werden soll.

- die Sensor-Anschlussplatine 191148-2, die in das Gehäuse des Temperatur/Feuchte-Sensors passt. Sie bietet RJ11-Buchsen für den Anschluss der Module des Wind- und des Regensensors. Die Platinen -1 und -2 sind über sieben (der acht) Leiter eines gewöhnlichen Ethernet-Kabels verbunden.
- die Platine 191148-3 für den Feinstaub/Partikel-Sensor, die

nur ein Adapter ist, um den mechanischen Anschluss des Honeywell-Feinstaubsensoren HPM115 zu erleichtern. Diese Einheit ist im gleichen Gehäusotyp wie die Hauptplatine untergebracht und kann (ebenfalls über ein Ethernet-Kabel verbunden) räumlich abgesetzt von der Hauptplatine der Wetterstation verwendet werden.

## Die Hauptplatine

Die Hauptplatine enthält eine Reihe von Elementen, die von allen in der Wetter-

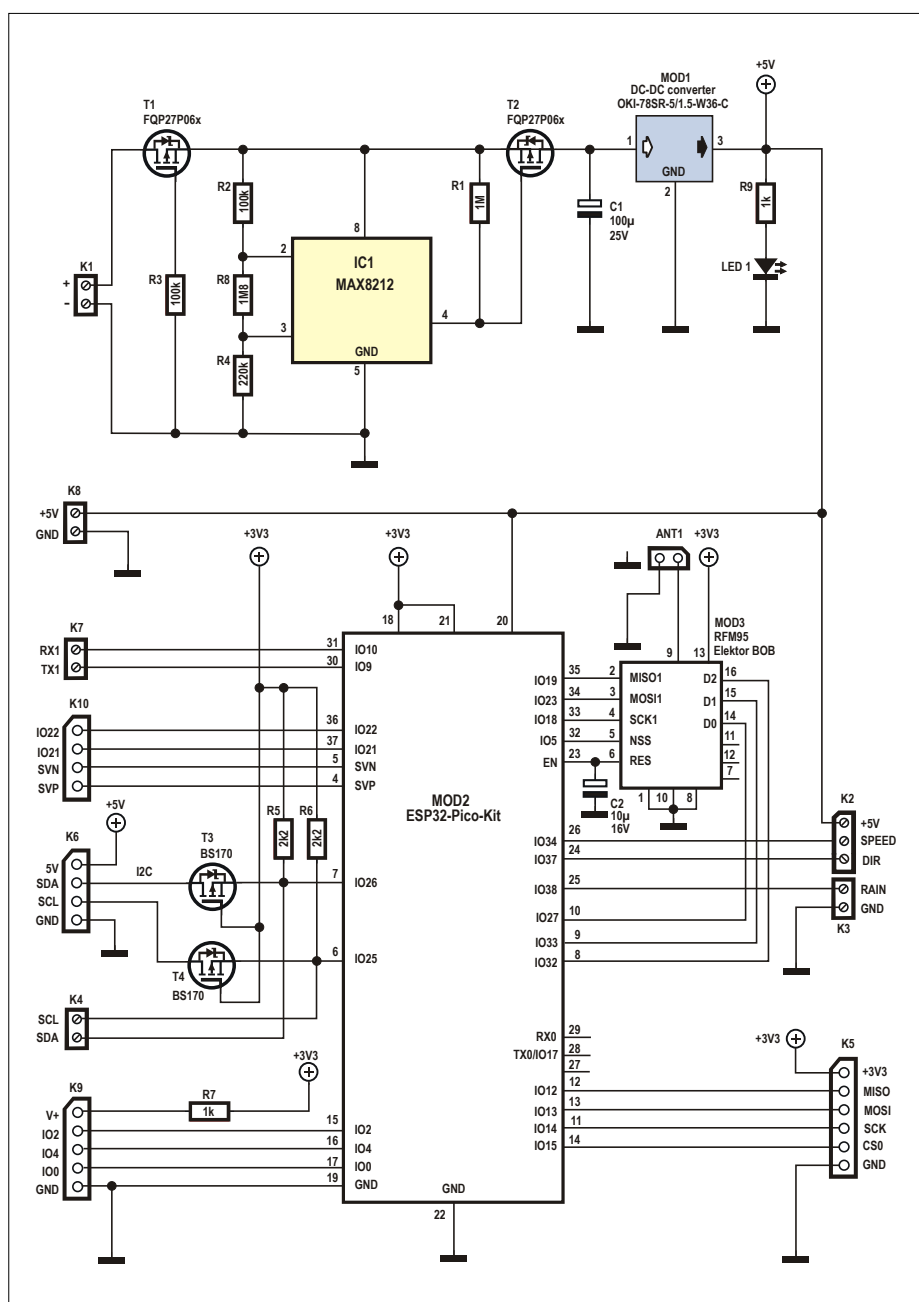


Bild 1. Kernschaltung der Open-Netzwerk-Wetterstation Mark 2 auf der Hauptplatine, die mit globalen Sensorik-Plattformen wie ThingSpeak und openSenseMap kompatibel ist. Man beachte die simplen, aber perfekt funktionierenden bidirektionalen 3,3-V/5-V-Pegelwandler am I<sup>2</sup>C-Bus, die mit FETs realisiert sind.

## INFOS ZUM PROJEKT



Sensoren Wetter  
ESP32 ThingSpeak  
openSenseMap



Einsteiger  
→ Fortgeschrittene  
Experte



ca. 3 Stunden  
(nur Platinenbestückung)



Laborwerkzeuge,  
kleiner Bohrer



60 € (Hauptplatine inkl. Gehäuse),  
40 € (Sensorplatine inkl. BME280),  
50 € (Feinstaubsensor inkl. Gehäuse)

station verwendeten Sensoren genutzt werden. Diese Elemente sind in der Schaltung in **Bild 1** leicht zu erkennen.

## Stromversorgung mit Verpolungs- und Tiefentladeschutz

Die Wetterstation wird normalerweise von einem 12-V-Akku versorgt, der seine Energie von einem Solarpanel in Kombination mit einem externen Solarlader bezieht. Die Wetterstation nimmt einen maximalen Strom von etwa 1,5 A auf, was weitgehend abhängig ist von der Zahl und der Art der verwendeten Sensoren. Der nutzbare Eingangsspannungsbereich liegt bei etwa 11...14 V<sub>DC</sub>. Der MOSFET T1 leitet nur, wenn die Batterie richtig herum angeschlossen ist. Der FET stellt mit dem Widerstand R3 also einen Verpolungsschutz dar. Wenn das Solarpanel versagt oder der Solarlader nicht genügend Energie liefert, muss der Verbraucher abgeschaltet werden, um eine Tiefentladung zu verhindern. Für diese Aufgabe haben wir den programmierbaren Spannungsdetektor ICL8212 auserwählt, dessen moderner Nachfolger die CMOS-Variante MAX8212 von Maxim Integrated ist. Mit nur drei externen Widerständen zur Einstellung der Schwellen/Hysteresis und dem externen p-Kanal-MOSFET T2 als Schal-

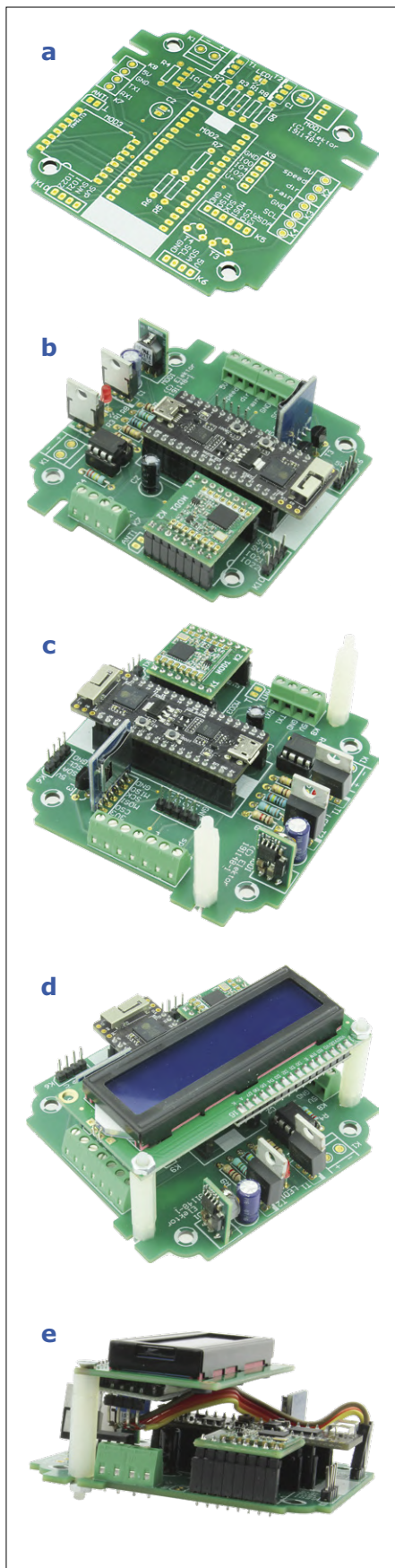


Bild 2. Die Hauptplatine in verschiedenen Stadien der Fertigstellung: (a) unbestückt, (b) mit bestückten Bauteilen und eingesetzten Modulen, (c) mit montierten Abstandshaltern, (d) mit angeschraubtem Display und (e) mit internen Kabeln.

ter macht der 8212 genau das, was wir wollen: im Bedarfsfall die Versorgungsspannung unterbrechen.

Mit den angegebenen Werten für R2, R8 und R4 ist eine untere Spannungsschwelle für eine Klemmenspannung der Batterie von 10,5 V eingestellt, unter der die Batterie vom Verbraucher getrennt wird. Die obere Spannungsschwelle, bei der die Verbindung nach erfolgreicher

Aufladung wieder geschlossen wird, entspricht einer Klemmenspannung von 11,1 V.

Ein Gleichspannungswandler (MOD1) setzt die 12-V-Batteriespannung auf eine geregelte 5-V-Versorgungsspannung für die „echte“ Elektronik herunter, wobei LED1 als Einschaltanzeige dient. Das ESP32-Pico-Kit-Modul besitzt einen internen 3,3-V-Regler, der nicht nur die



## STÜCKLISTE

### Hauptplatine 191148-1 v.1

#### Widerstände:

(alle Kohleschicht, 5%, 0,25 W, 250 V)

R1 = 1 M

R2,R3 = 100 k

R4 = 220 k

R5,R6 = 2k2

R7,R9 = 1k

R8 = 1M8

#### Kondensatoren:

C1 = 100 µ, 25 V, radial 2,5 mm

C2 = 10 µ, 16 V, radial 5 mm

#### Halbleiter:

IC1 = MAX8212EPA+

(Spannungsüberwachung)

LED1 = rot, 3 mm

T1,T2 = FQP27P06X, -60 V, -27 A, 120 W,  
V<sub>gs</sub> = -4 V, R<sub>dson</sub> = 70 mΩ

T3,T4 = BS170, 60 V, 500 mA, 830 mW,  
V<sub>gs</sub> = 2,1 V, R<sub>dson</sub> = 1Ω2

#### Außerdem:

K1 = 2-polige Platinenanschlussklemme,  
Raster 5,08 mm, 630 V

K2 = 3-polige Platinenanschlussklemme,  
Raster 3,81 mm, 300 V

K3,K4,K7,K8 = 2-polige  
Platinenanschlussklemme, Raster 3,81 mm,  
300 V

IC1 = DIL-8-Sockel

MOD2 = 17x1-Buchsenleiste, Raster 2,54 mm

K5 = BoB für Micro-SD-Karte,  
aus dem Elektor-Store

K6,K10 = 1x4-polige Stiftleiste, abbrechbar,  
vertikal

K9 = 1x5-polige Stiftleiste, abbrechbar,  
vertikal

LCD1 = LCD mit I<sup>2</sup>C-Erweiterung,  
2x16 Zeichen, hintergrundbeleuchtet,  
aus dem Elektor-Store \*

MOD1 = DC/DC-Wandler

OKI-78SR-5/1,5-W36-C, 5V 1,5A (Murata)

MOD2 = ESP32 Pico KIT V4,

aus dem Elektor-Store

MOD3 = Elektor-RFM95-BoB

(optional für zukünftige Erweiterungen)

S1 = Druckschalter, IP68,  
Alcoswitch PB6B2FM3M1CAL07

Gehäuse: TK PS 94×94×57 mm, Klasse IP66  
(Spelsberg)

2 Stück Sechskant-Abstandhalter M3×25mm

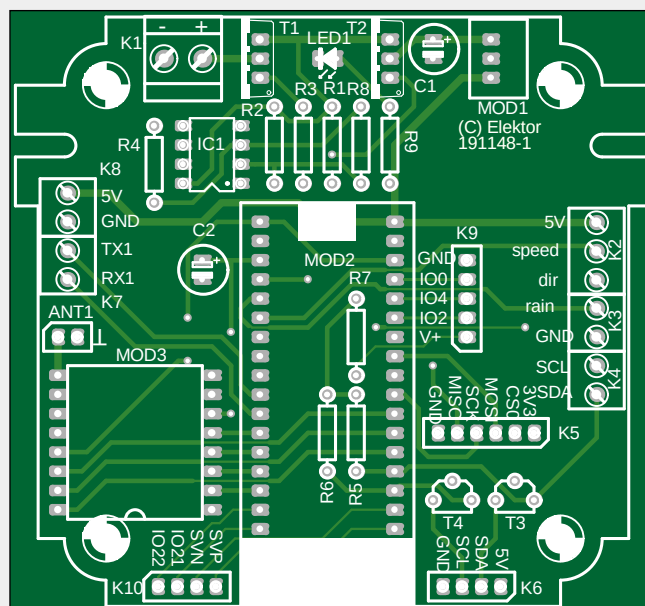
2 Stück M3-Mutter

2 Stück M3×10 mm Maschinenschraube

2 Stück PG7 Kabelverschraubung Nylon, IP68

6 Stück Steckbrücken, Buchse/Buchse

Platine 191148-1 V1.2, aus dem Elektor Store



\* aufgrund von  
Produktions- und  
Exportproblemen  
kann es zu Verzöge-  
rungen bei der  
Lieferung kommen





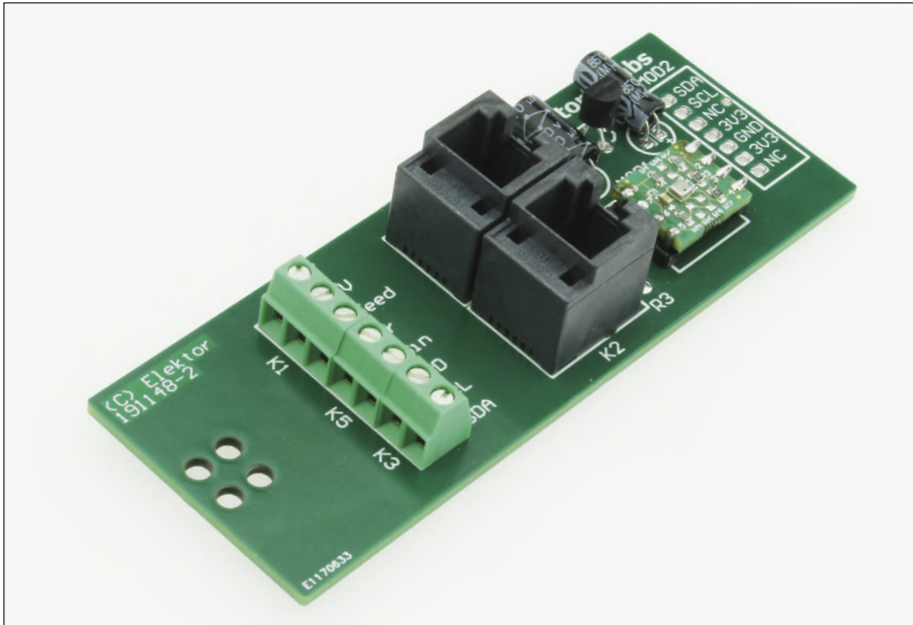


Bild 4. Fertige Anschlussplatine, von oben betrachtet. Hinter den RJ11-Buchsen befindet sich das kombinierte Druck/Feuchte/Temperatur-Sensormodul.

Anschlüssen nach innen), um im Gehäuse mehr Platz für den Anschluss der Kabel zu schaffen. Eine seltsame Lösung, aber durchaus praktikabel.

### Sensor-Anschlussplatine

Obwohl auf dieser Platine auch das Modul mit dem Feuchte/Temperatur/Luftdrucksensor untergebracht ist, besteht ihre Hauptfunktion darin, die Verbindung zwischen dem ESP32-Mikrocontroller und den anderen peripheren Sensoren herzustellen. Der Schaltplan in **Bild 3** macht dies deutlich. Diese Schaltung ersetzt die Platine im Inneren der Thermo/Hygro-Einheit, die mit dem WH-SP-WS02-Outdoor-Kit geliefert wird und passt perfekt in das Gehäuse.

Der 3,3-V-LDO-Regler IC1 (Low Dropout) ist das einzige aktive Bauteil auf dieser Platine. Abgesehen von den drei Pull-up-Widerständen R1, R2, R3 für die Schalter im Regen- und im Windsensor beschränkt sich der Rest der Bauteile auf den Anschluss aller in der Wetterstation verwendeten Sensormodule.

Die RJ-11-Buchsen werden mit dem Windsensor (K2) und dem Regensensor (K4) verbunden, wobei ihre Position mit den Ausbrüchen in dem evakuierten Gehäuse der Thermo/Hygro-Einheit (und auch der Beschriftung auf dem Gehäuse) übereinstimmt. An Position MOD2 können weitere Sensoren mit I<sup>2</sup>C-Bus angeschlossen werden.

Das beliebte BME280-Breakout-Board von Elektor kann als 3-in-1-Sensormodul verwendet werden, das die Temperatur, die relative Feuchtigkeit und den Luftdruck ermittelt. Die Pinnummerierung auf der Platine an der Position MOD1 stimmt mit der des BME280-BoBs überein. Sollten Sie ein I<sup>2</sup>C-kompatibles T/RH/P-Sensormodul einer anderen Marke vorziehen, so können Sie das BME280-BoB abziehen und Ihr Modul an Position MOD2 anschließen, aber beachten Sie dabei die Anschlussbelegung.

**Bild 4** zeigt die vom Elektor-Labor entworfene und auf elektrische



### IM ELEKTOR-STORE

→ Hauptplatine, unbestückt:  
[www.elektor.de/191148-1](http://www.elektor.de/191148-1)

→ Steckverbinder-Platine, unbestückt:  
[www.elektor.de/191148-2](http://www.elektor.de/191148-2)

→ Staub/Partikelsensorplatine, unbestückt:  
[www.elektor.de/191148-3](http://www.elektor.de/191148-3)

→ Wetterstation Mk. 2, Bausatz mit Teilen (Elektronik):  
[www.elektor.de/191148-71](http://www.elektor.de/191148-71)

→ 2×16-Zeichen-LCD mit I<sup>2</sup>C-Erweiterung \*:  
[www.elektor.de/two-wire-lcd-071035-93](http://www.elektor.de/two-wire-lcd-071035-93)

→ MicroSD-Karten Breakout-Board \*:  
[www.elektor.de/19251](http://www.elektor.de/19251)

→ BME280 BoB, I<sup>2</sup>C-Version (160109-91):  
[www.elektor.de/bme280-mouser-intel-i2c-version-160109-91](http://www.elektor.de/bme280-mouser-intel-i2c-version-160109-91)

→ ESP32-PICO-Kit V4:  
[www.elektor.de/esp32-pico-kit-v4](http://www.elektor.de/esp32-pico-kit-v4)

→ Wetterstation WH-SP-WS02 (Kit mit Windmessern, Regensensor, Thermo/Hygro-Sensor, Halterung):  
[www.elektor.de/professional-outdoor-weather-station-wh-sp-ws02](http://www.elektor.de/professional-outdoor-weather-station-wh-sp-ws02)

\* Lager/Versandverzögerungen aufgrund von Produktions- und Exportproblemen

### Weblinks

- [1] Wetterstation mit ESP32, Elektor Januar/Februar 2019: [www.elektormagazine.de/magazine/elektor-69/42263/](http://www.elektormagazine.de/magazine/elektor-69/42263/)
- [2] 2-Draht-LCD: [www.elektor.de/two-wire-lcd-071035-93](http://www.elektor.de/two-wire-lcd-071035-93)
- [3] Projekt-Webseite: [www.elektormagazine.de/191148-03](http://www.elektormagazine.de/191148-03)
- [4] Elektor-Labs-Webseite: [www.elektormagazine.com/labs/remake-elektor-weather-station](http://www.elektormagazine.com/labs/remake-elektor-weather-station)
- [5] Datenblatt Honeywell-Sensor:  
<https://sensing.honeywell.com/honeywell-sensing-particulate-hpm-series-datasheet-32322550>



und mechanische Eignung geprüfte Anschlussplatine. Das Verbindungskabel kann mit kleinen Kabelbindern an den vier Platinenbohrungen sicher befestigt werden.

### Platine für den Feinstaub/ Partikel-Sensor

Die Anschlüsse K7 und K8 können mit einem Honeywell-Feinstaub/Partikel-Sensor HPMA115S0-xxx verbunden werden. Da der Sensor bei Frost und Feuchtigkeit irreparabel beschädigt werden kann, haben wir ihn nicht in das Gehäuse des Thermo-/Hygro-Sensors integriert. Der Sensor muss nämlich „ventiliert“ eingebaut werden, siehe Datenblatt [5]!

Der Hauptzweck dieser Platine ist es, einen einfachen Anschluss und eine sichere Befestigung des Sensormoduls zu gewährleisten. Entsprechend simpel sieht auch der Schaltplan in **Bild 5** aus. Der Honeywell-Sensor besitzt einen achtpoligen Steckverbinder mit einem Rastermaß von 1,25 mm (0,05 Zoll), der sich ohne professionelles Werkzeug nur schwer anschließen lässt. Verwenden Sie deshalb die in der Stückliste angegebenen Steckverbinder von Würth Elektronik mit konfektionierten Kabeln. So müssen Sie einfach nur die vier Kabel an K3 (zwei für die 5-V-Stromversorgung und zwei für Rx/D und Tx/D) befestigen und den Steckverbinder im 1,25-mm-Raster vorsichtig in den Sensor stecken. Feinstaubsensor- und Hauptplatine können über die Platinenanschlussklemmen K1 und K2 und einem Ethernet-Kabel verbunden werden - ganz benutzerfreundlich. Das Foto in **Bild 6** zeigt die fertige Platine mit dem Sensor.

Verwenden Sie drei M2x10-mm-Maschinenschrauben, um den Sensor auf der Platine zu befestigen. Vor der Montage der Platine müssen Sie an geeigneter Stelle zwei Lüftungslöcher in das Gehäuse bohren, damit ein ausreichend starker Luftstrom für korrekte Messungen erzielt werden kann und gleichzeitig verhindert wird, dass Feuchtigkeit im Sensor verbleibt, wenn dieser später in der freien Natur, aber am besten vom Regen geschützt, installiert ist.

In der Juli/August-Ausgabe von Elektor geht es weiter mit der Beschreibung der Software. Sie können die Fortschritte auf unserer Labs-Website [4] aktiv verfolgen. ◀

191148-03

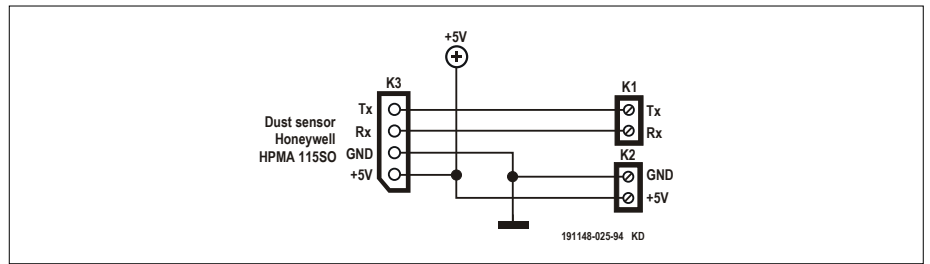


Bild 5. Minimale Schaltung der Feinstaub/Partikel-Sensorplatine für den Anschluss des Honeywell-Sensors HPMA115.



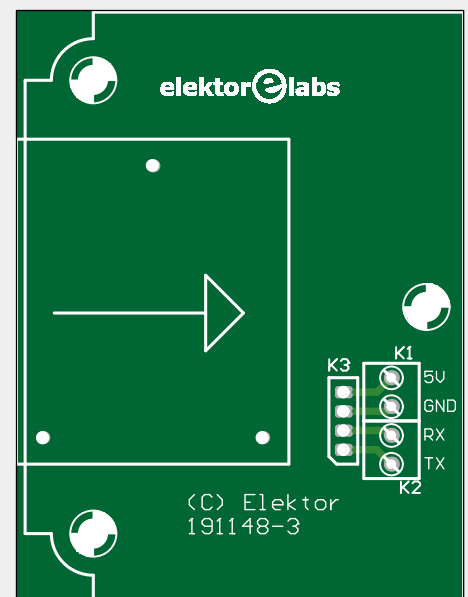
Bild 6. Die fertig bestückte Staub/Partikel-Sensorplatine, montiert in einem Gehäuse der Schutzklasse IP66.

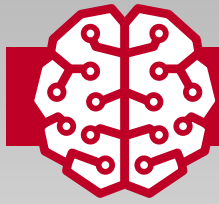


## STÜCKLISTE

### Feinstaub/Partikel-Sensorplatine 191148-3

- MOD1 = Feinstaub-Sensormodul  
HPMA115S0-xxx  
(Honeywell, bei Farnell 2770767)
- K1, K2 = 3-polige  
Platinenanschlussklemme,  
Raster 3,81 mm, 300 V
- 8-poliges WR-PHD-  
Steckverbindergehäuse,  
Raster 1,25 mm  
(Würth Elektronik 653008113322)
- 2 vorverpresste Leitungen für WR PHD  
(Würth-Elektronik 653153128130)
- 3 Stück M2x10 Maschinenschraube
- Gehäuse: TK PS 94x94x57 mm,  
Klasse IP66 (Spelsberg)
- Platine 191148-3 V1.0  
aus dem Elektor-Store



**AI**

# KI für Einsteiger (1)

## Objekterkennung mit dem Maixduino-Board

Von **Walter Trojan**

Kein anderes IT-Thema steht derzeit so im Vordergrund wie die Künstliche Intelligenz. Ein guter und preisgünstiger Einstieg in die Welt der Sprach- und Bilderkennung sowie weitere Facetten der KI ist das Maixduino-Board. Ausgestattet mit einer Kamera und einem kleinen LCD-Panel kostet es nur rund 30 € und ist darüber hinaus über die bekannte Arduino-IDE programmierbar. Im ersten Teil dieses Artikels stellen wir die reichhaltige Hardware des Maixduinos und einige Demo-Programme vor, darunter eines, das 1000 verschiedene Objekte erkennen kann.

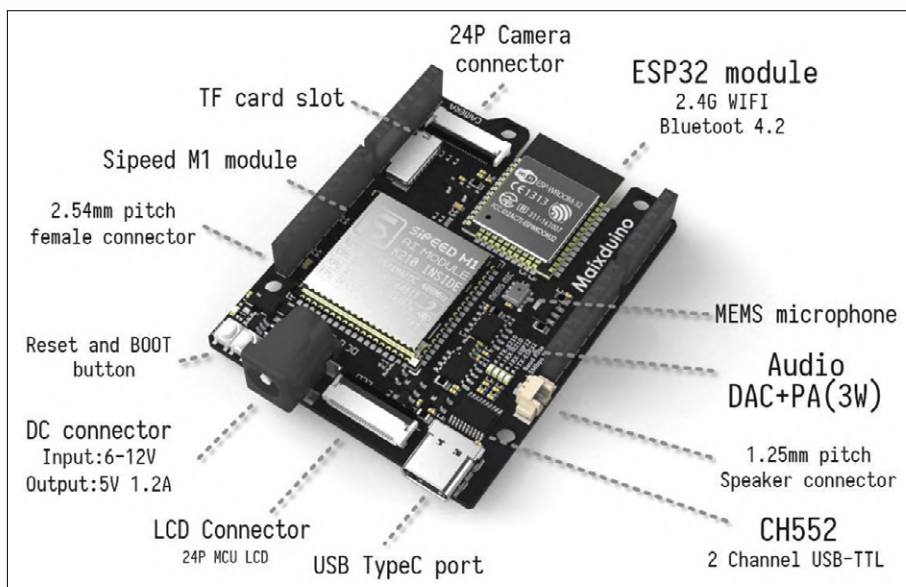


Bild 1. Ausstattung des Maixduino.

In vielen Bereichen ist die Künstliche Intelligenz auf dem Vormarsch und hält Einzug in viele Geräte und Anwendungen. Für das laufende Jahr werden die weltweiten Umsätze mit Unternehmensapplikationen im Bereich KI auf 4,8 Milliarden US-\$ prognostiziert, in 2025 sollen es schon über 30 Milliarden US-\$ sein. Wer sich in diesem Bereich auskennt, erhöht seine Jobchancen, denn KI-Fachleute werden dringend gesucht. Es erfordert zwar etwas Aufwand, sich mit dem Thema zu befassen, man erschließt sich damit aber geradezu neue Welten von möglichen Anwendungen. Nicht

zuletzt bereitet das Ganze auch großes Vergnügen.

Unter Künstlicher Intelligenz ist u.a. die Nachbildung eines intelligenten Verhaltens zu verstehen. Ein Teilgebiet davon ist Maschinelles Lernen, bei dem eine Applikation nicht mehr vom Entwickler in allen Einzelheiten durchdacht wird, sondern sich ein universell einsetzbares Programmgerüst in Form eines Neuronalen Netzes (NN) anhand zahlreicher Beispieldaten die erforderlichen Funktionen selbst beibringt. Eine Unterkategorie davon ist Deep Learning (vertieftes Lernen), bei dem sich mittels komplexer

und optimierter Programmstrukturen die gelieferten Ergebnisse weiter verbessern. Näheres zu den hier genannten Fachbegriffen finden Sie im weiteren Verlauf dieser kleinen Artikelserie.

Dabei ist KI kein brandneues Thema, sondern existiert bereits seit den 50er Jahren des letzten Jahrhunderts. Der Durchbruch gelang allerdings erst vor wenigen Jahren, als die Rechenleistung von Prozessoren erheblich gesteigert werden konnte. Grafikkarten mit hundert parallel arbeitenden Rechenwerken und hochspezialisierte KI-Chips ermöglichen nun das Training und die Anwendung von KI-Lösungen. Damit wurde der Weg frei für die in vielen persönlichen Assistenten eingesetzte Spracherkennung sowie die Bilderkennung von Objekten jeglicher Art. Auch beim autonomen Fahren ist der KI eine wesentliche Rolle zugeordnet. In Teilbereichen ist die KI bereits heute dem Menschen überlegen, sie schlägt die Weltmeister im Schach- und Go-Spiel und kann z.B. im Gesundheitswesen verlässlicher Tumore erkennen als spezialisierte Fachärzte. Aber keine Sorge, diese Überlegenheit bezieht sich lediglich auf die antrainierten Spezialfähigkeiten - das jeweilige Trainingsgebiet gibt auch in Zukunft der Mensch vor.

### Preisgünstiger Einstieg

Ein guter und preisgünstiger Einstieg in diese Welt ist der Maixduino, der ausgestattet mit einer Kamera und einem kleinen LCD-Panel gerade einmal rund



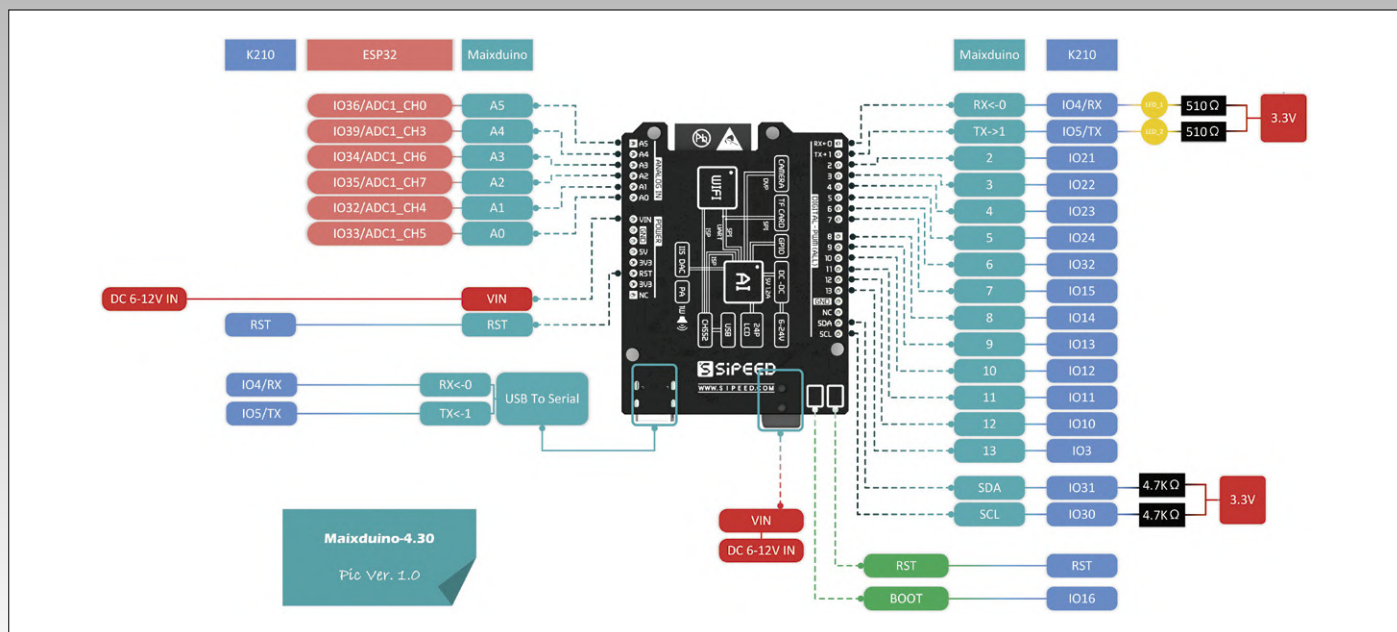


Bild 2. Pinbelegung des Maixduino.

30 € kostet (das Kit aus Board, Kamera und Display ist als *MAix BiT Kit* auch bei Elektor erhältlich, siehe Kasten). Die Platine hat die Bauform eines Arduino Uno, woher sich auch der Name ableitet, sie ist aber sehr viel reichhaltiger ausgestattet. Hergestellt wird das Board von der chinesischen Firma Sipeed. Alternativen zum Maixduino sind z.B. der Jetson Nano von Nvidia, der ROCK PI N10 Model A, der Intel Neural Compute Stick 2 und einige andere, die allerdings in der 100-€-Klasse angesiedelt sind.

Im ersten Teil dieses Artikels stelle ich Ihnen die reichhaltige Hardware des Maixduinos und seine Programmierung mittels der Arduino-IDE vor. Dabei zeige ich neben einigen Arduino-typischen Anwendungen den Einsatz von Kamera und Bildschirm. Als Highlight demonstriere ich abschließend, wie man das Board zur Objekterkennung nutzen kann. Im zweiten und dritten Teil steigen wir dann vertieft in das Thema KI ein, beschreiben den Aufbau neuronaler Netze, installieren MicroPython und die dazugehörige IDE und zeigen, wie Gesichtserkennung funktioniert. Darüber hinaus wird dargestellt, wie eigene KI-Anwendungen programmiert werden und die Kommunikation mit dem Internet erfolgt.

### Ein detaillierter Blick auf den Super-Arduino

Die Spezifikationen des Maixduinos lassen die Herzen von Hardware-

Enthusiasten höher schlagen. Größe und Bauform des Boards stimmen weitgehend mit dem Arduino Uno überein, aber man erkennt sofort die höhere Bauteildichte und zahlreiche Zusatzanschlüsse. Im Mittelpunkt stehen zwei große Chips. Zum einen der KI-Prozessor Sipeed M1 mit dem Kendryte K210, auf dessen Innereien ich später zu sprechen komme, sowie ein ESP32 zur Kommunikation über WLAN und Bluetooth und zur Erfassung analoger Signale. Letzterer steuert mit seinen beiden mit 240 MHz getakteten Prozessoren schon eine erhebliche Verarbeitungsleistung bei und entlastet die Hauptprozessoren von lästiger Kommunikationsarbeit. Der ESP32 ist in dieser Zeitschrift schon mehrfach vorgestellt worden, so dass ich hier nicht auf weitere Einzelheiten eingehe.

Die Buchsenleisten entsprechen in Polzahl und Anordnung dem Arduino-Original und selbst die Pinbelegung ist weitgehend gleich. Aber Achtung: Die Ein/Ausgänge sind entweder für 3,3 V oder sogar nur für 1,8 V ausgelegt, **5 V wären absolut zerstörerisch!**

Weitere Details sind in **Bild 1** dargestellt. Zum Anschluss von Kamera und LCD-Panel stehen jeweils 24-polige Buchsen zur Verfügung. Über einen Slot für eine Micro-SD-Karte können dem Maixduino auch große Datenmengen zugänglich gemacht werden. Der USB-Anschluss in zeitgemäßer Typ-C-Technologie dient zur Programmierung und zum Monitoring. Und zur Verarbeitung von Audio-Daten

stehen ein digitales Mikrofon sowie ein Audioverstärker mit 3 W Ausgangsleistung und vorgeschaltetem Digital-Analog-Konverter zur Verfügung. Mit dieser reichhaltigen Ausstattung sind sicher zahlreiche Projekte ohne Zusatz-Boards realisierbar.

Ein Blick auf die Pinbelegung (**Bild 2**) zeigt die starke Verwandtschaft mit dem Original. Die Spannungsversorgung kann über die DC-Buchse oder den Pin VIN mit einer Gleichspannung von 6...12 V erfolgen oder alternativ via USB mit 5 V. Vom ESP32 sind die GPIOs 32...36 und 39 herausgeführt, die alternativ als Analog-eingänge A0...A5 nutzbar sind. Auf der gegenüberliegenden Seite befinden sich die Ein/Ausgänge des K210-Moduls. Sie können mit den gewohnten Arduino-Befehlen angesteuert werden, wobei die Arduino-Pinnummer der Nummer des K210-Ein/Ausgangs entspricht:

```
#define LED 12
// K210-IO12, Maixduino-Pin 10
pinMode(LED, OUTPUT);
// Port als Ausgang
digitalWrite(LED, HIGH);
// Port auf High-Potential (3,3 V)
```

Der Pin RST ist ein 1,8-V-Typ und sollte bei externer Beschaltung nicht mit einer höheren Spannung beschickt werden. Die Ausgänge IO36...IO47 sind ebenfalls für den niedrigen Pegel ausgelegt, aber nicht auf Board-Pins herausgeführt, sondern werden intern, z.B. für die LCD-An-

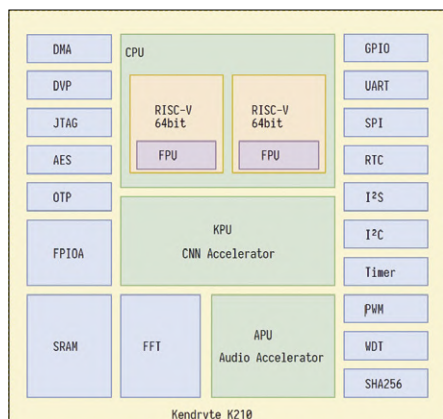


Bild 3. Blockschaltung des Kendryte K210.

steuerung, genutzt. Die Ports für RX/TX und I2C sind bereits auf dem Board mit passenden Pullup-Widerständen versehen.

Leider spezifiziert das K210-Datenblatt nicht den maximalen Ausgangsstrom der Ausgänge. Er dürfte zwar zur Ansteuerung einer LED mit ca. 10 mA

reichen, aber darüber hinaus würde ich ein Treiber-IC einsetzen.

### Herzstück: Der Kendryte K210

Das SoC (System-on-a-Chip) K210 der chinesischen Firma Kendryte ist das Herzstück des Maixduinos. Es wird in einer stromsparenden Technologie mit 28 nm Rastermaß gefertigt und ist seit September 2018 auf dem Markt. Zur Abwicklung der „normalen“ Arbeitslast stehen zwei 64-Bit-Prozessoren zur Verfügung, die mit 400 MHz getaktet sind, wobei eine Übertaktung bis 800 MHz möglich ist. Ihr Inneres basiert auf dem offenen RISC-V-Standard, was dem Hersteller die Lizenzgebühren an die Firma ARM erspart und etwas zu dem niedrigen Verkaufspreis beiträgt. Beide Prozessoren sind zur Unterstützung von Gleitkommaoperationen (einfach und doppelt genau) mit einer FPU (Floating Point Unit) ausgestattet. **Bild 3** zeigt die weiteren Innereien dieses Chips im Detail.

Das Besondere am K210 ist seine KPU (Knowledge Processing Unit) zur Abbil-

dung und Berechnung Neuronaler Netze. Er verfügt über eine in diesem Preissektor phänomenale Rechenleistung von 0,46 TOPs, das sind 460 Milliarden Instruktionen pro Sekunde. Das ließe sich durch Overclocking noch verdoppeln (womit zum Beispiel eine Erkennungsrate von 60 Objekten pro Sekunde möglich wäre). Diese hohe Verarbeitungsgeschwindigkeit wird von 64 parallel arbeitenden Arithmetik-Einheiten mit einer Bitbreite von 576 erzielt. Dabei ist der Energieverbrauch von 0,3 W im Vergleich zu anderen KI-Systemen sehr gering. So empfiehlt z.B. Nvidia für den Jetson Nano ein Netzteil mit 5 V/4 A = 20 W bei einer vergleichbaren Rechenleistung von 0,4 TOPs.

Die KPU kann fortschrittliche Architekturen von Neuronalen Netzen abbilden, z.B. Convolutional Netze, die sich durch ein besonders effizientes Lösungsverfahren auszeichnen und sich aufgrund spezieller Filterverfahren besonders für die Bildauswertung eignen. Dazu mehr im zweiten Artikel. Der im SoC integrierte schnelle Hauptspeicher mit einer Kapazität von 8 MB, aufgeteilt zwischen den Prozessoren und der KPU im Verhältnis 6 zu 2 MB, ermöglicht es, dass bis zu 5,9 MB für Neuronale Netze zur Verfügung stehen. Damit lassen sich durchaus mittelgroße Netze implementieren.

Aber der K210 hat noch weitere Hardware-Schätze zu bieten. Da wäre zum einen der Audio Prozessor (APU) zur Analyse von Audiodaten, besonders hilfreich bei der Vorverarbeitung der Sprachanalyse. Dieser verfügt über 8 Kanäle bzw. 4 Stereo-Kanäle. Eingangssignale lassen sich mit einer Abtastrate von bis zu 192 kHz erfassen und von einer FFT-Einheit (Fast Fourier Transformation) im Hinblick auf das enthaltene Frequenzspektrum zerlegen.

Außerdem sind die Beschleuniger für Verschlüsselungen noch erwähnenswert, der AES Accelerator und SHA256 Accelerator.

Natürlich ist auch die sonst übliche Peripherie an Bord, wie UART, I2C, SPI, I2S, Timer, RTC und PWM.

Sipeed hat das Maixduino-Board zusätzlich mit 16 MB Flash, einer passenden Spannungsversorgung, Mikrofon und 3-W-Verstärker und dem großartigen ESP32 ausgestattet. Hardware-Herz, was willst du mehr? Weitere Informationen über die Inbetriebnahme, die Schaltung und andere Einzelheiten entnehmen Sie bitte den Weblinks [1][2][3].

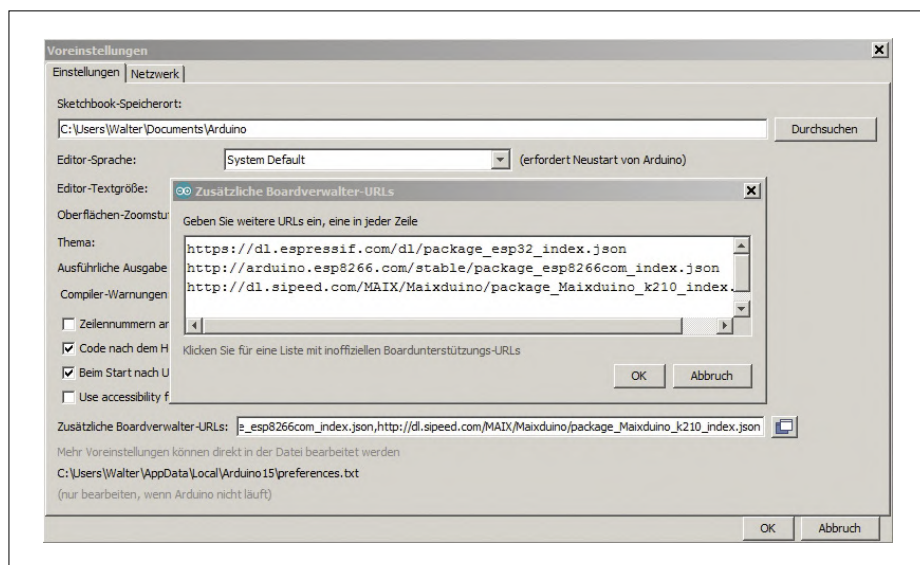


Bild 4. Eingabe Maixduino-Core.

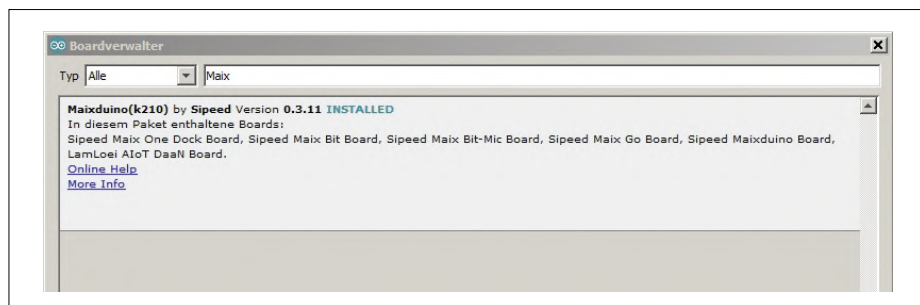


Bild 5. Installation des Maixduino-Core.



## Software-Entwicklung

Nicht nur die Bauform, sondern auch die Software-Entwicklung ist am Original-Arduino angelehnt, es wird dabei die Arduino-IDE eingesetzt. Zur Erweiterung dieser Entwicklungsumgebung wird wie beim ESP8266 oder ESP32 der Maixduino-Core in die IDE integriert. Dazu gibt man im Menü unter *Datei* -> *Voreinstellungen* eine zusätzliche Boardverwalter-URL ein, wobei ein Klick auf den Button rechts neben der Eingabezeile ein kleines Zusatzfenster zur bequemen Eingabe öffnet (**Bild 4**). Wenn Sie nur den Maixduino programmieren wollen, dann ist lediglich die „sipeed“-Zeile erforderlich, denn der On-Board-ESP32 wird auch darüber programmiert.

Zur Installation des Maixduino-Cores wird über das Menü *Werkzeuge* -> *Board* der Boardverwalter aufgerufen. Nach Eingabe des Suchbegriffs „Maix“ (**Bild 5**) kann die Installation erfolgen.

Und schon können wir mit der Programmierung beginnen. Auf das übliche „Hallo Welt“ verzichte ich hier, wir fangen sofort mit der Überprüfung von Kamera und Bildschirm an. Schließen Sie den Maixduino über USB an und stellen über den Menüpunkt *Werkzeuge* die Parameter für das Board ein:

Board: [Sipeed Maixduino](#)  
CPU Clock Frequency: 400 MHz  
Burn Tool Firmware: [open-ec](#)  
Burn Baud Rate: 1.5 Mbps  
Tool Install Location: [Standard](#)  
Port: [<Verwendeter COM-Port>](#)  
Programmer: [k-flash](#)

Ein Demo-Programm zur Bildaufnahme und Wiedergabe auf dem LCD steht bereits in der IDE zur Verfügung; es wird mittels *Datei* -> *Beispiele* -> *Sipeed\_OV2640* -> *selfie* aufgerufen und nach dem Hochladen (gegebenenfalls die Reset-Taste betätigen) ausgeführt. Den Code sieht man in **Bild 6**.

Am Programmanfang erfolgt die Definition der benötigten Sipeed-Bibliotheken für Kamera und LCD, die mittels SPI-Bus mit dem Board verbunden sind. Die gewählte Auflösung entspricht dem QVGA-Standard und beträgt 320 \* 240 Bildpunkte (farbig, RGB565). In der Setup-Routine erfolgt dann die Initialisierung der beiden Geräte. Und in der endlosen Loop-Schleife werden die aufgenommenen Bilder jeweils unmittelbar auf dem LCD dargestellt. Viel einfacher geht es nicht.

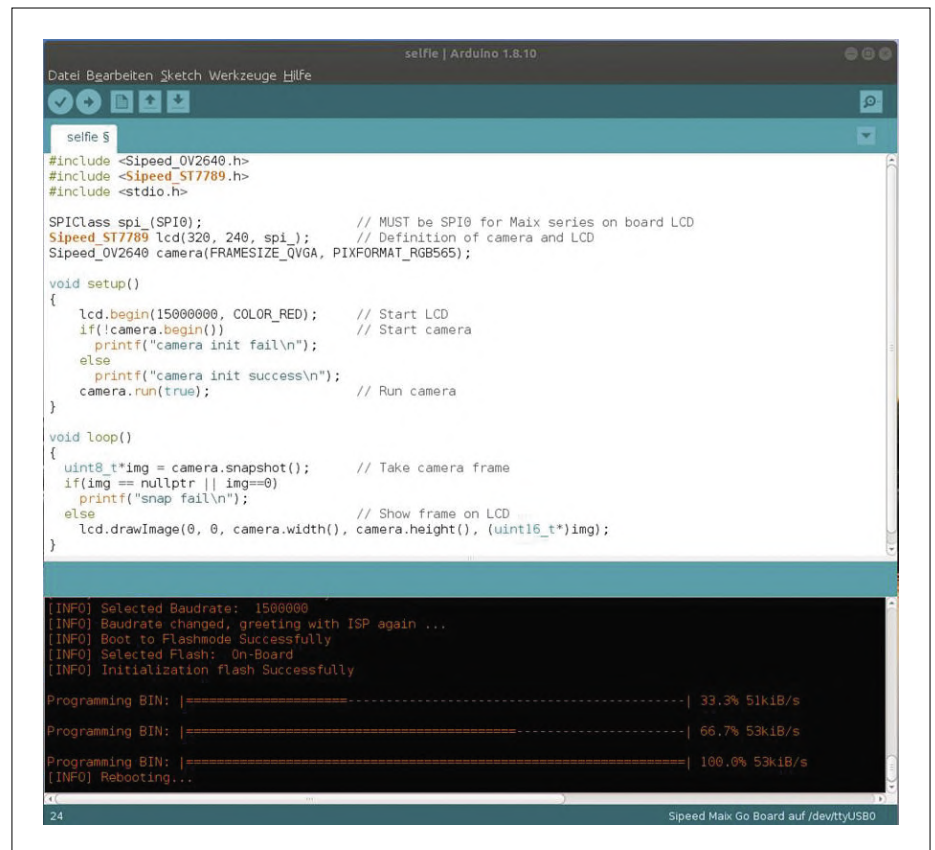


Bild 6. Arduino-IDE mit Selfie-Programm.

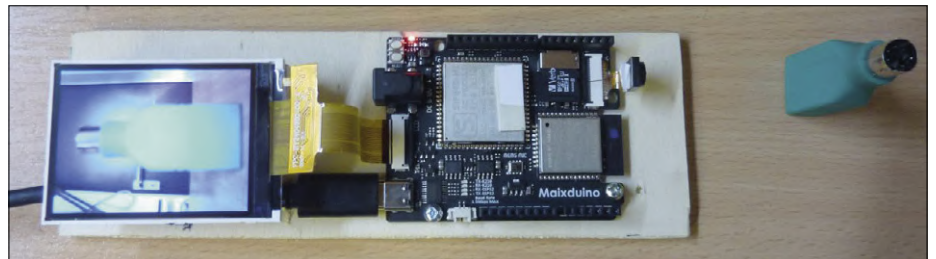


Bild 7. Aufnahme eines Adapters.

Das Bild ist nicht sonderlich kontrastreich, aber scharf, und die Wiedergabe erfolgt flüssig. In **Bild 7** sieht man das Resultat. Es zeigt sich, dass der Maixduino genauso zugänglich ist wie ein Uno, er bietet jedoch viel mehr Möglichkeiten und man kann sofort mit anspruchsvolleren Applikationen starten.

## Ein erstes KI-Modell

Obwohl die Künstliche Intelligenz mit Deep Learning vertieft im zweiten Artikelteil behandelt wird, soll hier bereits eine erste Applikation demonstriert werden. Es handelt sich dabei um die *MobileNet*-Demonstration, die nichts mit Mobilfunk zu tun hat, sondern 1000 Objekte

aus dem Alltag erkennen und klassifizieren kann. Dazu werden einem Neuronalen Netz, einer Software-Struktur bestehend aus in Schichten angeordneten Knoten, die Bilder der Objekte in einem aufwändigen Trainingsverfahren tausendfach vorgelegt.

Da die Datei mit den Bildern für das Training eine Größe von ca. 200 GB hat und der Anlernvorgang sehr aufwändig ist, wäre dieser auf dem Maixduino kaum durchführbar. Aber es gibt die Möglichkeit, ein bereits trainiertes Modell einzusetzen und unmittelbar mit der Objekterkennung zu beginnen. Genau dies möchte ich jetzt demonstrieren.

Die dazugehörige Software steht unter

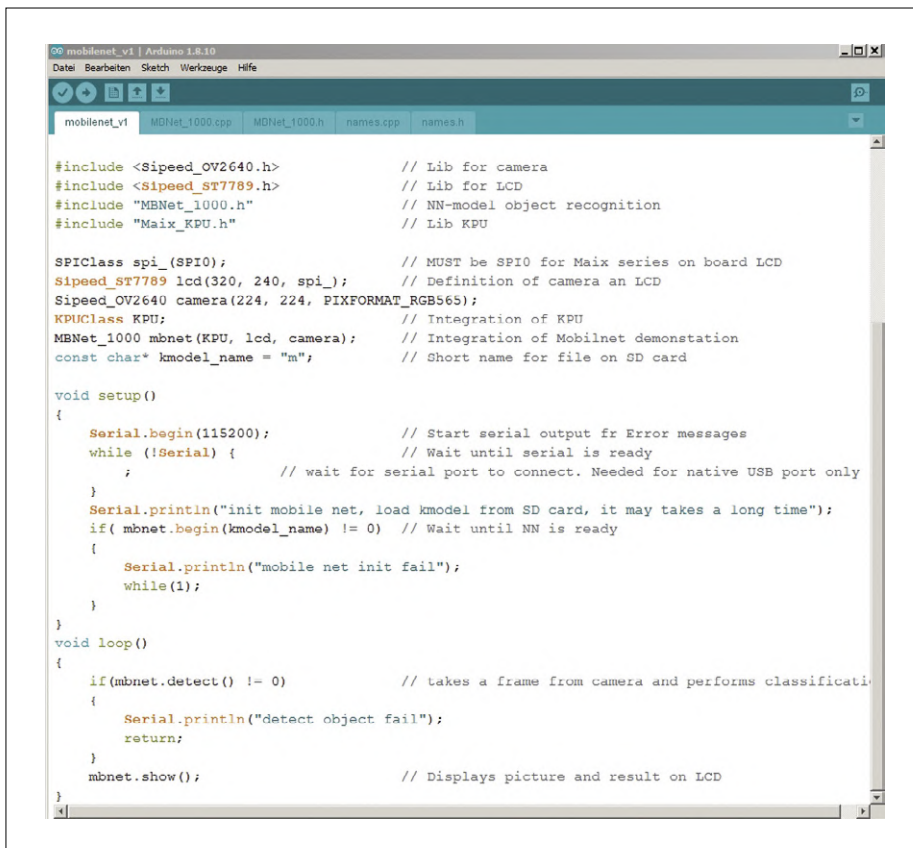


Bild 8. Programm Mobilenet-Demo.

dem Link [4] zur Verfügung und enthält folgende Komponenten:

- *mobilenet\_v1.ino*: C++-Hauptprogramm zur Abwicklung der Demo
- *MBNet\_1000.h*: Header-Datei für Demoroutinen
- *MBNet\_1000.cpp*: C++-Routinen für die Erfassung, Erkennung und Darstellung der Objekte
- *names.h*: Header-Datei für die Objektbezeichnungen
- *names.cpp*: C++-Routine mit den Bezeichnungen der erkennbaren Objekte

Diese Dateien laden Sie herunter und platzieren Sie alle in einen Ordner. Das bereits trainierte KI-Modell „mobile-net\_0x300000.kfpgk“ ist unter Link [5] zu finden, gepackt ist es einige MB groß. Nach dem Entpacken mit 7zip finden Sie einen Ordner „mobilenet\_0x300000“ in dem sich zwei Dateien befinden, darunter die Datei „m“ mit dem Modell. Diese kopieren Sie auf eine Micro-SD-Karte in das Root-Verzeichnis (das oberste Verzeichnis) und schieben die Karte in den Maixduino-Slot.

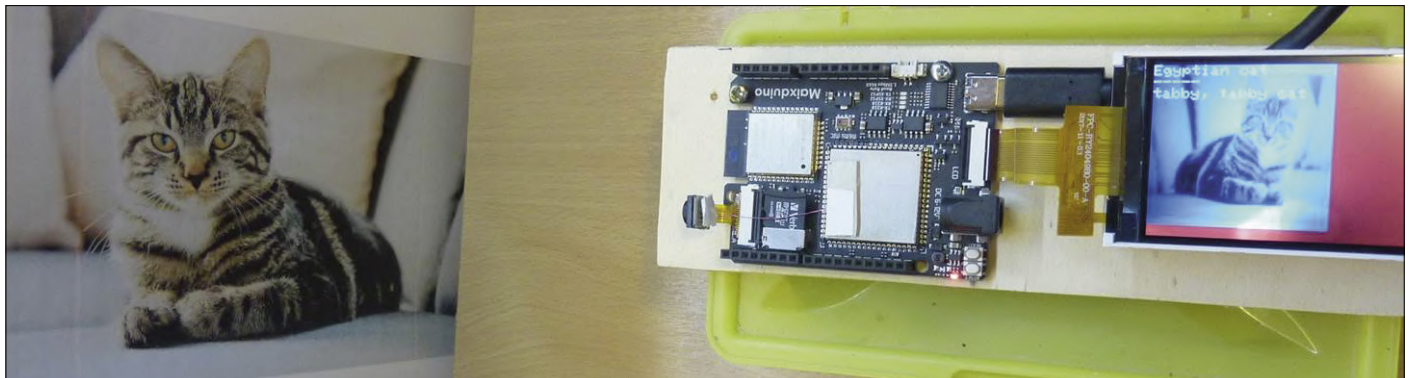


Bild 9. Aufnahme eines Objekts.

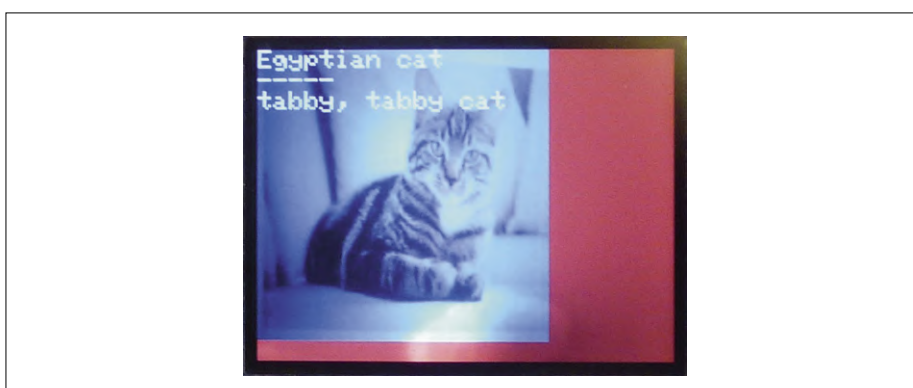


Bild 10. Objekt erkannt und klassifiziert.

Ein Blick auf den Programmcode in **Bild 8** zeigt, dass man mit geringem Aufwand auch sehr komplexe Applikationen programmieren kann. Das liegt natürlich an der hohen Leistungsfähigkeit der Bibliotheken, die im KI-Bereich zahlreich und hocheffizient verfügbar sind. Im ersten Programmteil erfolgen die Einbindung von Kamera, LCD, KPU und der Demoanwendung und danach deren Parametrierung. Dabei wird die Kamera auf das Format der Trainingsbilder eingestellt, nämlich auf 224\*224 Pixel. Schließlich erhält das Objekt *mbnet*



die Betriebsmittel KPU, Kamera und LCD zugeordnet.

In der Setup-Routine wird dann die Demonstration gestartet und in der Loop-Schleife erfolgt die Klassifizierung der von der Kamera erfassten Objekte, gefolgt von deren Darstellung auf dem LCD. Viel einfacher kann man eine Objekterkennung nicht programmieren.

Als Testobjekt habe ich dem Maixduino ein Katzenbild präsentiert (**Bild 9**), das er sofort als gemusterte bzw. ägyptische Katze identifizierte (**Bild 10**). Die Anwendung klassifiziert ca. 5 Bilder pro Sekunde und wäre damit in der Lage, auch aus der Bewegung heraus nutzbare Resultate zu erzeugen. Dabei sollte allerdings die Beleuchtung gut und der Hintergrund nicht zu verworren sein. In Bild 10 kann man erkennen, dass die Aufnahme zur Erzielung einer besseren Erkennungsrate nun eine blau-weiße Farbgebung aufweist. Das ist eine oft praktizierte Methode: Statt mit drei Farbenen eines Bildes arbeitet man einfarbig, um die Anzahl der Bildpunkte und damit den erforderlichen Rechenaufwand zu reduzieren. Der hier praktizierte Einsatz eines bereits trainierten NNs, also Training und Klassifikation auf unterschiedlichen Plattformen, ist in diesem Sektor durchaus üblich. Viele



Anwender nutzen zum Training die hohe von AWS, Microsoft Azure oder Google Cloud bereitgestellte Rechenleistung und führen das NN mit geringerem Ressourcenbedarf auf kleineren Plattformen aus. Die Halbleiterindustrie reagiert bereits darauf, so hat Intel kürzlich die Neuronal Network Processors Nervana™ NNP-T und NNP-I angekündigt, wobei die T-Version mit hoher Rechenleistung zum Training und die I-Version mit geringerer Performance für die Inference (Schlussfolgerung bzw. Klassifizierung) ausgelegt sind.

#### Fortsetzung folgt

Mit der Demonstration einer KI-Anwendung konnte das Thema in diesem Beitrag lediglich angerissen werden, aber ich habe hoffentlich Ihr Interesse für lernende Maschinen geweckt. Im nächsten Artikel werden Struktur und Funktionsweise eines Neuronalen Netzes näher beleuchtet. Dann kommt eine neue Entwicklungsumgebung auf Sie zu, denn ich möchte Ihnen die Befas-

sung mit Linux und der Programmiersprache Python schmackhaft machen. Denn dort stehen die leistungsfähigsten Bibliotheken und Frameworks zur Verfügung. Diese machen z.B. die Gestaltung eines Neuronalen Netzwerks fast so einfach wie das Zusammenstecken von Legobausteinen. Als zusätzliche KI-Applikation stelle ich die Gesichtserkennung und die Entwicklung eigener NNs vor. Für die Ungedulden und Neugierigen unter Ihnen empfehle ich die Lektüre „Neuronale Netze selbst programmieren“ von Tariq Rashid. ◀

200023-01

#### Weblinks

- [1] Maixduino Beschreibung: <https://wiki.sipeed.com/en/maix/board/maixduino.html>
- [2] Sipeed Forum: <https://bbs.sipeed.com/>
- [3] Maixduino-Schaltung: <http://dl.sipeed.com/MAIX/HDK/Maixduino/Maixduino-4.30/>
- [4] Mobilenet-Demo: [https://github.com/sipeed/Maixduino/tree/master/libraries/Maix\\_KPU/examples/mobilenet\\_v1](https://github.com/sipeed/Maixduino/tree/master/libraries/Maix_KPU/examples/mobilenet_v1)
- [5] Mobilenet-Modell: [http://dl.sipeed.com/MAIX/MaixPy/model/mobilenet\\_0x300000.kfpgk](http://dl.sipeed.com/MAIX/MaixPy/model/mobilenet_0x300000.kfpgk)

Anzeige



HAMMOND  
MANUFACTURING®

## Hand Held Gehäuse Standard und Wasserdicht

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

**+ 44 1256 812812**  
**sales@hammondmfg.eu**

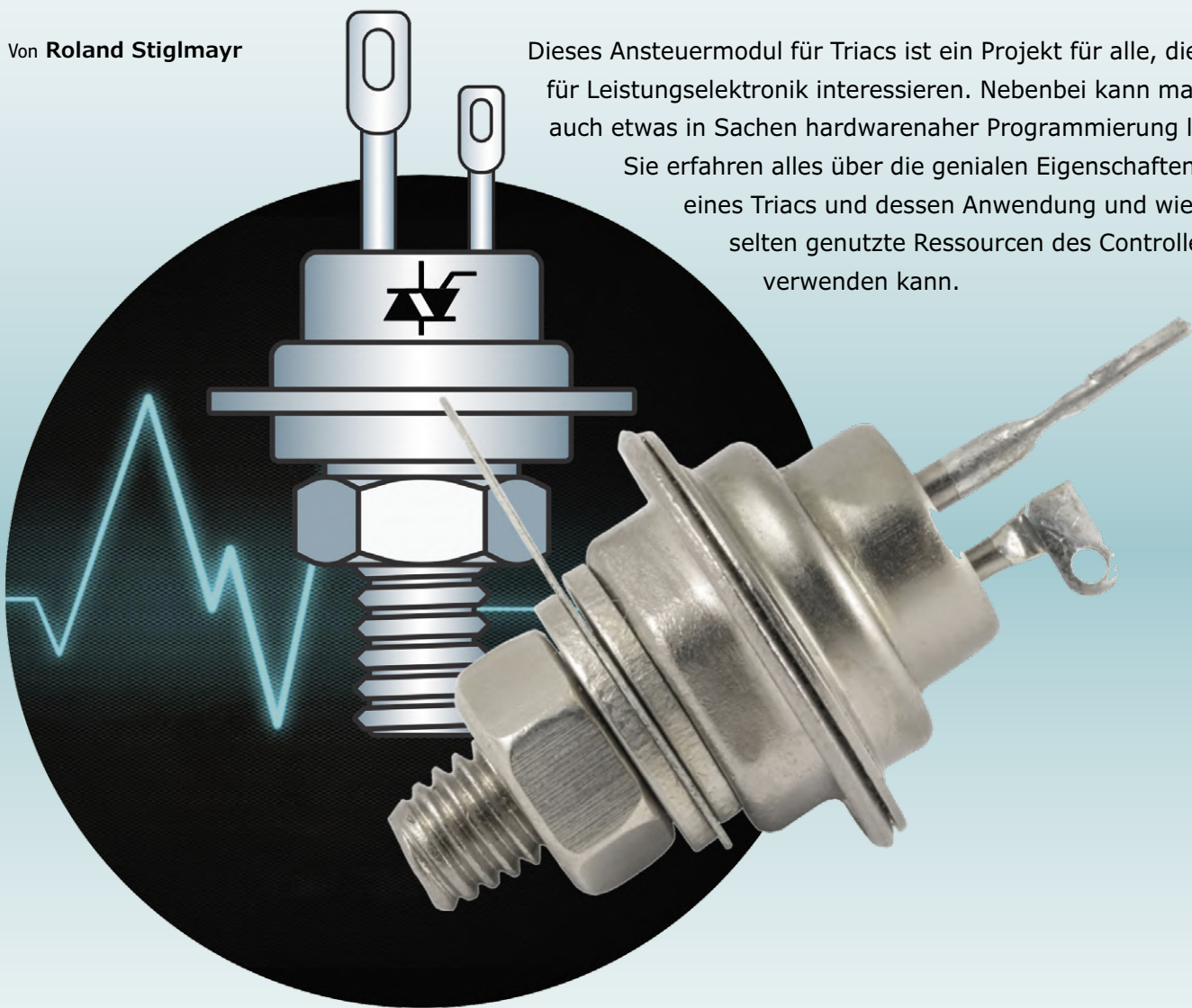


# Universelles Ansteuermodul für Triacs mit ATmega

## Schalten und Dimmen unterschiedlicher Lasten

Von Roland Stiglmayr

Dieses Ansteuermodul für Triacs ist ein Projekt für alle, die sich für Leistungselektronik interessieren. Nebenbei kann man auch etwas in Sachen hardwarenaher Programmierung lernen. Sie erfahren alles über die genialen Eigenschaften eines Triacs und dessen Anwendung und wie man selten genutzte Ressourcen des Controllers verwenden kann.



Jeder Elektroniker weiß halbwegs, was ein Thyristor und ein Triac ist, wie sie funktionieren und wo man sie einsetzt. Weil diese Bauteile aber enorme Möglichkeiten bieten, werden sie nur von Spezialisten richtig ausgereizt. Da Thyristoren heute meist nur in Spezialfällen bei sehr hohen Spannungen ( $>1$  kV) und Strömen (viele kA) eingesetzt werden, dreht sich dieser Beitrag ausschließlich um Triacs.

Zunächst geht es um die Grundfunktion eines Triacs und relevante Parameter. Anschließend wird dann die Elektronik des Ansteuermoduls beschrieben. **Bild 1** zeigt den fertigen

Prototypen. Die Kombination aus ATmega und Triac erlaubt viele Funktionen: Je nach Firmware kann das Modul als einfacher (Remote-)Schalter, Timer, Dimmer oder Softstart-Modul zum Schalten induktiver Lasten wie Trafos oder Motoren eingesetzt werden.

### Grundlagen

Ein Triac ist ein Halbleiter-Schalter, der den Strom in durchgeschaltetem Zustand in beiden Richtungen leitet – folglich ein echter Wechselstromschalter. Durch die beiden Elektroden T1



und T2 (von Terminal) fließt der (Last-)Strom. Diese Elektroden bezeichnet man auch als MT1/MT2 (von Main Terminal) bzw. früher als A1 und A2 (von Anode). Das Gate G als dritte Elektrode ist der Steueranschluss. **Bild 2** zeigt das Schaltsymbol eines Triacs in der Elektor-Variante (links) und eine einfache Ersatzschaltung (rechts).

Ein Triac sperrt, bis einmal ausreichend Steuerstrom durch G in Richtung T1 fließt. Ist der Triac erst einmal leitend, verbleibt er solange in diesem Zustand, bis der Strom durch T1/T2 unter einen typabhängigen Schwellwert gefallen ist. Dann sperrt er wieder. Im Betrieb an Wechselspannung wird er folglich bei jedem Nulldurchgang des Stroms (der nicht unbedingt in Phase mit der Spannung ist) abgeschaltet, wenn er nicht permanent an G durchgesteuert wird. Ein Triac lässt sich also über einen kurzen und damit energiearmen Impuls am Gate zünden, doch sein Abschalten kann man nicht direkt erzwingen.

T1 ist der Bezugspunkt bzw. das Referenzpotential für das Gate G. Anhand der Polarität des Potentials am Gate- und T2 jeweils relativ zu T1 ergeben sich vier unterschiedliche Stromflussrichtungen. Diese sind im Vier-Quadranten-Modell von **Bild 3** dargestellt.

Die X-Achse repräsentiert die Polarität der Spannung am Gate relativ zu T1. Die Y-Achse repräsentiert die Polarität der Spannung an T2 relativ zu T1. Die sich ergebenden vier Quadranten sind in Bild 3 mit römischen Zahlen bezeichnet. Positive Werte sind rot und negative blau dargestellt. Die Parameter eines Triacs werden in den Datenblättern für alle vier Quadranten angegeben, da sie sich oft unterscheiden.

## Parameter

Um den Triac zu triggern bzw. durchzusteuern, muss der Schwellwert von  $I_{GT}$  (Gate Trigger Current) überschritten werden. Der Triac bleibt nach Ende des Steuerimpulses nur leitend, wenn der einsetzende Laststrom größer als  $I_L$  (Latching Current) ist. Ist er kleiner, dann sperrt der Triac nach Ende des Steuerimpulses wieder. Trigger- und andere Ströme sind in den Datenblättern für jeden Quadranten einzeln spezifiziert. Unterschreitet der Laststrom durch den einmal gezündeten Triac den Haltestrom  $I_H$  (Holding Current), dann sperrt der Triac wieder. Bei induktiven Lasten ergibt sich zwischen Strom und Spannung eine vom induktiven Anteil abhängige Phasenverschiebung. Deshalb ist die Spannung an der Last im Nulldurchgang des Stroms nicht unbedingt auch null. Aus diesem Grund kann die Spannung an T2 beim Sperren schlagartig auf hohe Werte springen. Das kann Probleme geben: Ist die Steigung des Spannungsanstieges größer als der Grenzwert von  $dv_{COM}/dt$ , dann wird der Triac erneut leitend. Je nach Last kann das zerstörerisch wirken. Verhindern lässt sich ein zu schneller Spannungsanstieg durch ein parallel zu T2 und T1 geschaltetes RC-Glied (englisch: Snubber). Sogenannte Hi-Com-Triacs sind diesbezüglich wesentlich unempfindlicher.

Auch der Parameter  $dv_d/dt$  ist relevant. Wird sein Grenzwert am gesperrten Triac überschritten, wird er ebenfalls leitend. Wenn der Folgestrom dann gerade eine Halbwelle andauert, kann ein so eingeschalteter Trafo in die Sättigung geraten und schlagartig niederohmig werden. Der Triac und evtl. die Leiterbahnen der Platine überleben das oft nicht. Abhilfe bringt auch hier ein Snubber oder ein Hi-Com-Triac.

Kritisch ist auch der Parameter  $dI_T/dt$ , der Grenzwert der maximalen Stromanstiegsgeschwindigkeit nach dem Triggern des Triacs. Ein Überschreiten führt zur Zerstörung des Triacs

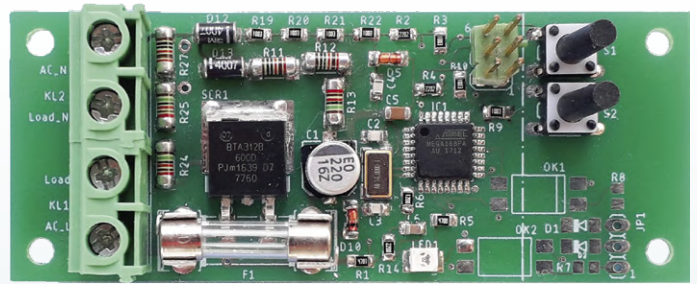


Bild 1. Prototyp des vom Autor realisierten Steuermoduls.

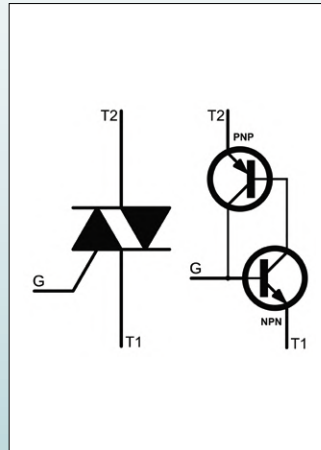


Bild 2. Elektorisiertes Schaltsymbol eines Triacs (links) und die Ersatzschaltung aus PNP- und NPN-Transistor für den ersten Quadranten (rechts).

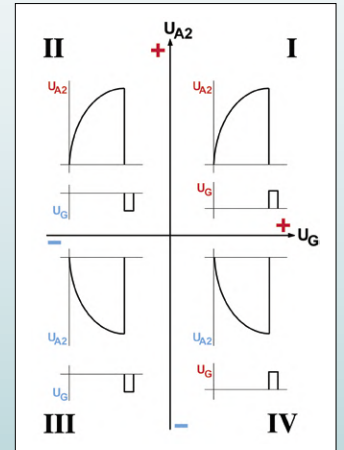


Bild 3. Die vier Quadranten ergeben sich aus der Polarität der Spannungen am Gate bzw. T2 gegenüber T1.

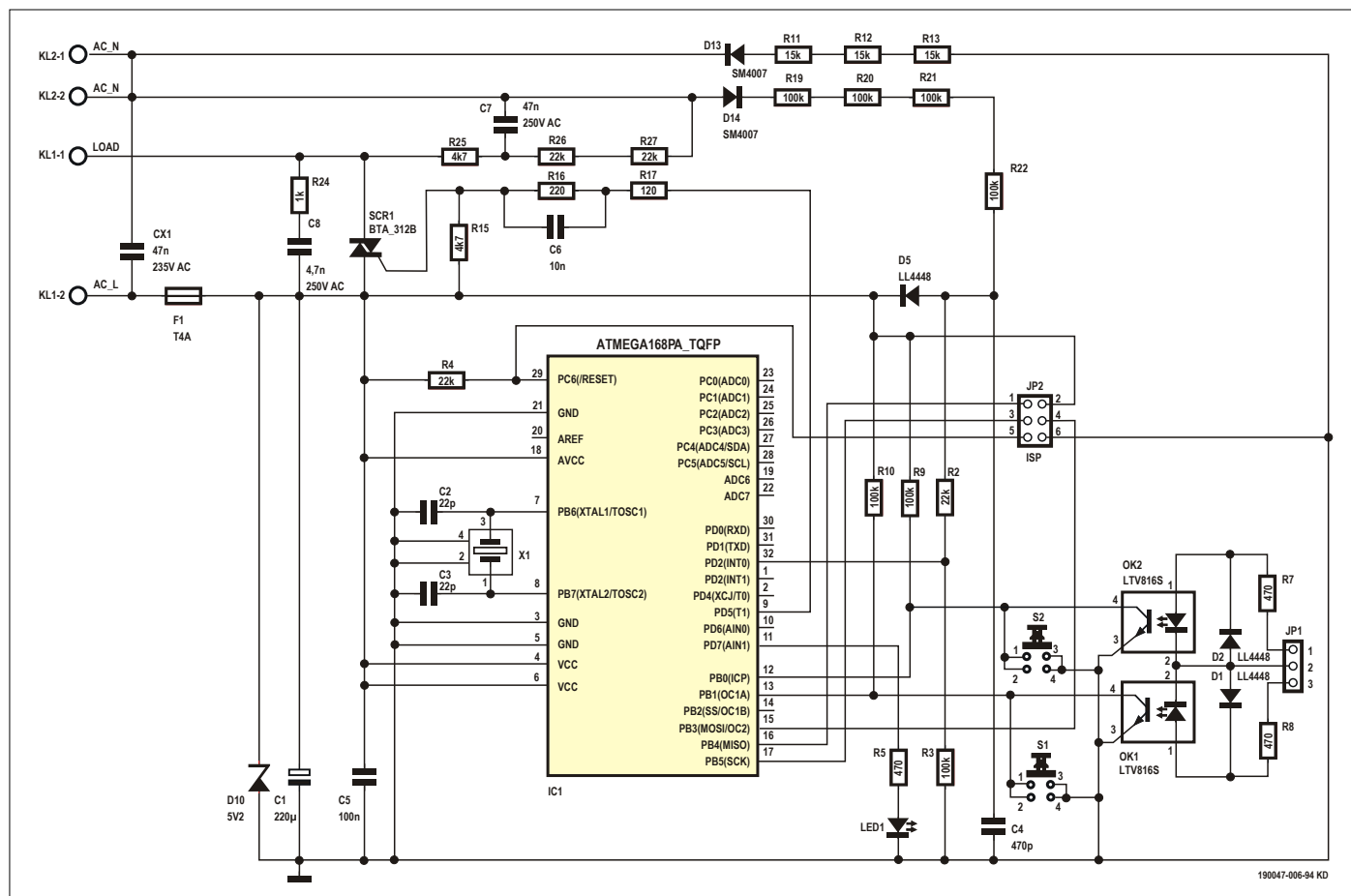
mit Kurzschluss zwischen T2 und T1. Der Grund hierfür ist die nicht homogene Stromdichte um die Gate-Elektrode, was zur lokalen Überhitzung führt. Da der Wert für  $dI_T/dt$  speziell im 4. Quadranten gering ist, sollte man diese Betriebsart vermeiden. Das  $dI_T/dt$ -Problem bleibt klein, wenn im Spannungs-Nulldurchgang geschaltet wird - allerdings nur für überwiegend resistive Lasten. Bei Bedarf kann die Anstiegsgeschwindigkeit des Stroms auch durch das Vorschalten einer kleinen Luft-Drossel begrenzt werden.

Der Parameter  $dI_{COM}/dt$  definiert den Grenzwert der Anstiegsgeschwindigkeit des Stromes im Stromnulldurchgang. Wird dieser Wert überschritten, wird der Triac nicht gelöscht, sondern bleibt leitend.

Fehlerhaft gezündet wird der Triac auch, wenn die periodische maximale Sperrspannung  $V_{DRM}$  (Repetitive Peak Off-state Voltage) überschritten wird. Abhilfe bringt ein Varistor parallel zu T2/T1, der die maximale Spannung begrenzt.

Der letzte erklärungsbedürftige Parameter ist das Grenzlasterintegral  $I^2t$  bzw. das Integral des quadrierten Stroms über 10 ms. Dieser Grenzwert entspricht der maximalen Energieabsorption der Triacs. Er sollte bei der Dimensionierung der Sicherung zwingend berücksichtigt werden.

Alle anderen im Triac-Datenblatt aufgeführten Größen sind weitgehend selbsterklärend. Wer selbst Triac-Schaltungen entwerfen möchte, sollte jetzt in der Lage sein, sich anhand der Parameter den idealen Triac für seine Anwendung auszuwählen.



## Triac-Ansteuermodul

Wichtig ist auch der minimal vorkommende Laststrom. Sowohl  $I_H$  als auch  $I_L$  müssen klar überschritten werden, sonst wird die Last nicht sauber eingeschaltet bzw. kommt kein kontinuierlicher Stromfluss zustande. Universell einsetzbare und für viele einfache Zwecke gut geeignete Exemplare sind die 600-V-Typen BT134-600E (4 A) und BTA312B (12 A).

mit der nötigen Kapazität und Spannungsfestigkeit sind nicht gerade preiswert. Noch simpler wäre ein ohmscher Vorwiderstand. Der Triac soll in den Quadranten II und III betrieben werden. Daher ist relativ zur direkt am Netz liegenden Elektrode T1 eine negative Gate-Triggerspannung erforderlich. Diese realisiert man am einfachsten per Einweggleichrichtung. Die Praktikabilität hängt aber direkt von der Stromaufnahme der Schaltung ab.

Der ungünstigste Fall ist eine hochfrequente Triggerung ohne Nulldurchgangsdetektion, bei der alle 278  $\mu\text{s}$  ein Triggersignal erzeugt wird. Bei  $V_{CC} = 4,7 \text{ V}$  braucht ein ATmega dann reale 1,9 mA. Bei einem Spitzenstrom von 11 mA für je 10  $\mu\text{s}$  bei einer Periode von 278  $\mu\text{s} = 3,6 \text{ kHz}$  ergibt sich ein mittlerer Gate-Strom von 400  $\mu\text{A}$ . Insgesamt fließen dann also ca. 2,3 mA. Dieser bestimmt den Wert des Vorwiderstandes und seine Verlustleistung. Bei der Halbwellengleichrichtung liegt der Effektivwert bei  $I_{pk} / 2$  und der Durchschnittswert bei  $I_{pk} / n$ . Daraus ergibt sich:



$$R = 325 \text{ V} / 7,3 \text{ mA} = 45 \text{ k}\Omega,$$

$$I_{\text{eff}} = I_{\text{pk}} / 2 = 3,7 \text{ mA}$$

und letztlich eine Verlustleistung am Widerstand R von:

$$P_{\text{vv}} = I_{\text{eff}}^2 \times R = (3,7 \text{ mA})^2 \times 45 \text{ k}\Omega = 0,62 \text{ W}.$$

Teilt man R auf drei 1-W-Einzelwiderstände auf, dann ist sowohl die Verlustleistung als auch die anliegende Spannung kein Problem.

Ein zusätzliches Problem ergibt sich beim Einschalten. Solange die MCU „im Reset“ ist, benötigt sie relativ viel Strom, und  $V_{\text{CC}}$  steigt nur sehr langsam. Durch Setzen des BOD-Levels auf 2,7 V und unmittelbares Aktivieren des Sleep Mode für 250 ms lässt sich dieses Problem lösen. Dies wird später noch erläutert.

## Sicherheit

Die Gefahren einer direkt am Stromnetz betriebenen Baugruppe dürfen niemals ignoriert werden. Bei Netzbetrieb darf kein Messgerät und schon gar kein PC angeschlossen sein. So etwas ist ein Fall für Elektrofachkräfte und Trenntrafos etc. sind obligatorisch. Sorgen Sie also für Berührungsschutz, wenn das Modul mit dem Netz verbunden ist. Es sollte unbedingt in ein isolierendes Gehäuse eingebaut sein!

Bei Tests oder Modifikationen muss das Modul vollständig vom Netz getrennt und über ein Netzteil mit 5 V versorgt werden. Ein 50-Hz-Signal mit einer Amplitude von 4 V aus einem Rechteckgenerator kann als Signal für die Nulldurchgangsdetektion dienen, das an C4 eingespeist wird.

## Hardware

In der Schaltung (**Bild 4**) wird die Versorgung der MCU samt Peripherie über die Vorwiderstände R11...R13 und die Einweggleichrichtung mit D13 sowie die Siebung mit C1 realisiert. MELF-Widerstände für R11...R13 bieten hohe Spannungsfestigkeit und Belastbarkeit. Dank der hochohmigen Stromversorgung ergibt sich annähernd eine Stromquelle, und die Z-Diode D10 kann daher problemlos für eine konstante Spannung sorgen. Der Quarz X1 taktet die MCU präzise und sorgt daher für Triggerimpulse mit exaktem Timing. D12 generiert zusammen mit dem Spannungsteiler aus R2, R3 und R19...R22 ein zur Detektion des Nulldurchgangs geeignetes Signal, das mit D5 auf  $V_{\text{CC}}$  „geklemmt“ wird.

R16 und R17 begrenzen den Gate-Strom. C6 parallel zu R16 sorgt für ein schnelles Triggern des Triacs. Parallel zu T2 und T1 des Triacs liegt der erwähnte Snubber aus R24 und C8, was fehlerhaftes Zünden verhindert. Da der eingesetzte Triac ein Hi-Com Typ ist, reicht eine kleine Kapazität und ein relativ hoher Widerstand.

Eine Besonderheit stellt die optionale Zusatzlast aus R25...R27 und C7 dar. Sie ist nur erforderlich, wenn die externe Last eine hohe Induktivität wie bei einem Transformator aufweist. Üblicherweise sind diese Bauteile nicht bestückt. Der X-Kondensator CX1 dient der Entstörung.

Über die Taster S1 und S2 wird das Modul bedient. Alternativ können auch die parallel geschalteten Optokoppler OK1 und OK2 genutzt werden. Für ausreichende Sicherheitsabstände nimmt man Taster mit langen Betätigungsstiften. LED1 dient als Betriebsanzeige. Programmiert wird IC1 über die

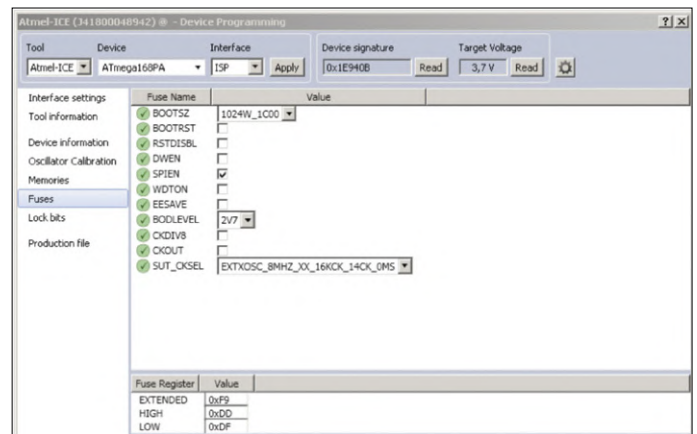


Bild 5. So setzt man die Fuses bei der ersten Programmierung des Atmega.

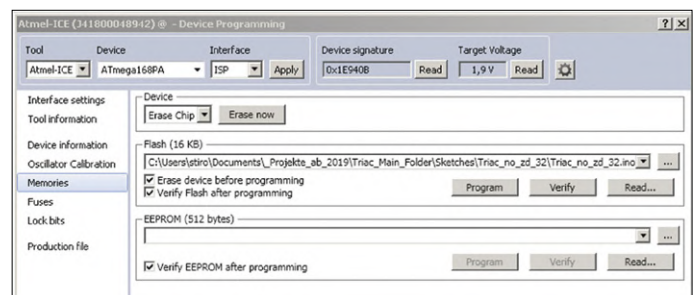


Bild 6. Einstellungen beim Brennen mit dem ISP-Programmer.

ISP-Schnittstelle an JP2. F1 am Netzeingang schützt im Falle eines Kurzschlusses.

## Software

Die Projekt-Software [1] wurde mit der Arduino-IDE erstellt. Wegen der Langsamkeit von Arduino-Funktionen hat der Autor allerdings auf diese verzichtet. Das selbst erstellte Programm ist sehr schnell und operiert deterministisch. Daher wurden die Sketche bewusst nicht nach Atmel Studio portiert.

Die Programmierung des Controllers kann leider nicht per Bootlader erfolgen. Er würde beim Einschalten so viel Strom benötigen, dass die Versorgungsspannung nicht hochlaufen würde. Folglich müssen die Fuses so gesetzt werden, dass der Bootlader inaktiv ist. Das Programmieren erfolgt direkt aus Atmel Studio durch einen geeigneten ISP-Programmer. Im Projektverzeichnis findet man für jede App neben dem dazugehörigen Sketch auch die HEX-Datei. Bei der ersten Programmierung müssen zwingend die Fuses wie in **Bild 5** programmiert werden. Dann wird das gewünschte HEX-File aus dem Projektverzeichnis geladen und auf die MCU gebrannt (siehe **Bild 6**). Nach der Modifikation eines Sketches muss dieser neu kompiliert werden (Überprüfen/Kompilieren). Das Statusfeld der IDE zeigt das Verzeichnis der erzeugten HEX-Datei an. Ein typischer Datei-Pfad wäre:

C:\Users\name\AppData\Local\Temp\buildxyz.tmp\TriaC\_...  
ino.hex

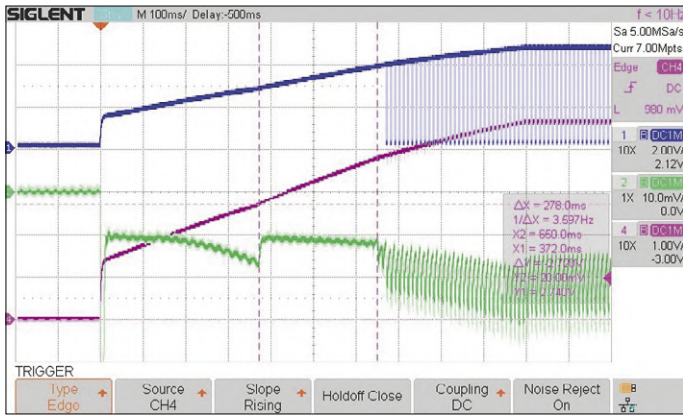


Bild 7. Verhältnisse beim Einschalten. CH1 (blau): Gate-Impulse; CH2 (grün): Strom; CH4 (rot):  $V_{CC}$ , X1, Y1: Start von WDT und Sleep, BOD-Level 2,7 V; X2: Ablauf WDT, Wake Up.

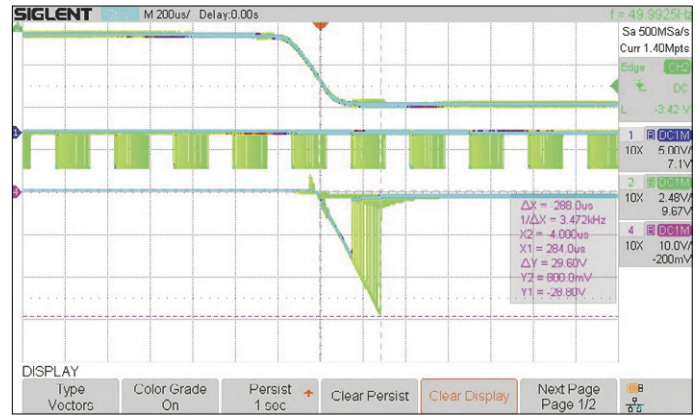


Bild 8. Spannung über Triac bei unkorrelierten Triggerimpulsen über 1 s. Obere Kurve (CH2): Netz Nulldurchgang; mittlere Kurve (CH1): Gate-Impulse; untere Kurve (CH4): Spannung über Triac.

Da das Verzeichnis und die Build-Nummer täglich neu vergeben werden, muss die Datei auch täglich im Atmel-Studio aktualisiert werden. Sollte das Verzeichnis *AppData* im Explorer/Dateimanager nicht sichtbar sein, muss im Explorer unter *Extras/Ordneroption/Ansicht „Alle Dateien und Ordner anzeigen“* eingestellt werden.

In Atmel-Studio lädt man dann das gewünschte HEX-File und programmiert damit den ATmega.

**Warnung:** Beim Programmieren muss das Modul völlig vom Netz getrennt und über eine sichere Kleinspannungsquelle versorgt werden.

## Anwendungen

Alle Apps haben die gleiche Startprozedur. Beim Einschalten bestimmt der BOD-Level von 2,7 V das Ende des Resets. Nach dem folgenden Programmstart wird der WatchDog-Timer aktiviert und das Programm in den Sleep-Mode versetzt. Dadurch kann  $V_{CC}$  sicher und schnell hochlaufen (**Bild 7**). Der Ablauf des WD-Timers löst den WDT-Interrupt aus und beendet den Sleep-Mode. Dann wird die Ausführung des eigentlichen Programms gestartet.

Der Sketch-Auszug von **Listing 1** zeigt, wie das Aufsetzen des WD-Timers und des Sleep-Modes realisiert wurde. Das

Listing enthält einen Code-Teil der Startprozedur. Zunächst wird der Watchdog gesetzt und aktiviert. Dann wird der Controller in den Sleep-Mode versetzt. Der Ablauf des WDT löst den WDT-Interrupt aus, der den Controller wieder aufweckt.

Für jedes Anwendungs-Beispiel ist ein eigener Sketch vorhanden. Jede App verfügt über unterschiedliche Funktionen, die über Taster oder Jumper selektiert werden. Weitere Informationen sind in den Kommentaren der jeweiligen Sketche enthalten.

## Einfache Schalter- und Timer-App

Da die Anwendung „Sketch Triac\_no\_zd\_32“ ohne Nulldurchgangsdetektion auskommt, zeigt sie die grundlegende Programmierung des Moduls und seine Besonderheiten. Die Schaltung wird zur Programmierung und für Tests an einem 5-V-Netzteil betrieben. Wenn die hardwarenahe Programmierung des WD-Timers, der unterschiedlichen Sleep Modes, der Interrupts, der Timer-Funktionen oder der I/O-Befehle interessiert, der vertieft sich in den Code, der wie alle Beispiele kostenlos von der Elektor-Webseite zu diesem Artikel [1] herunter geladen werden kann.

Per Taster wird die Funktion als Schalter oder als Timer ausgewählt. Mit der zusätzlichen Betriebsart „Autostart“ legt der Timer unmittel-

### Listing 1

```
void setup()
{
    //after reset, start with sleep mode to stabilize VCC.
    //prepare the watchdog for wakeup

    asm ("WDR");                //clear wd-timer
    WDTCSR= 0x1c;                //set WDCE, WDE to enable setting
    WDTCSR= 0x1c;                //wd-timer 0.25s
    WDTCSR= 0x14;                //clear WDE
    WDTCSR= (WDTCSR | 0x40);     //set WDIE, enable WD-interrupt
    interrupts();                //enable interrupts

    SMCR= 0x0f;                 //prepare xtd-standby
    asm ("SLEEP");               //enter sleep mode
    SMCR= 0x00;                 //finish, not necessary
    noInterrupts();              //disable interrupts, after wakeup

    MCUSR= (MCUSR & 0xf7);       //clear WDRF, wd-reset active
    WDTCSR= WDTCSR | 0x18;       //set WDCE, enable setting WDE
    WDTCSR= 00;                 //clear WDIE, disables wd-timer

    . . .
}
```



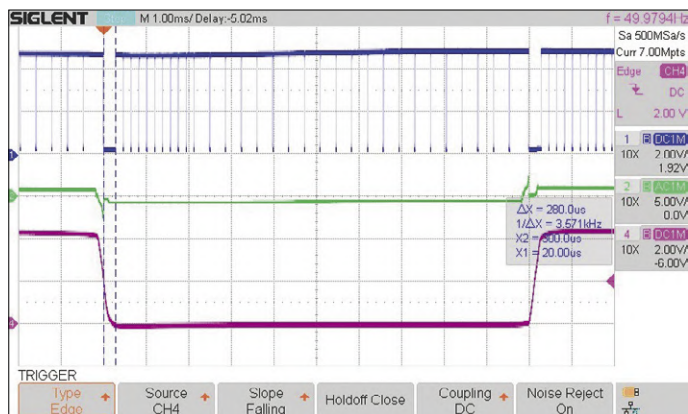


Bild 9. Einschalten einer LED-Lampe im Nulldurchgang der Netzspannung. CH1 (blau): Gate-Impulse; CH2 (grün): Spannung über Triac; CH4 (rot): Nulldurchgangs-Signal.

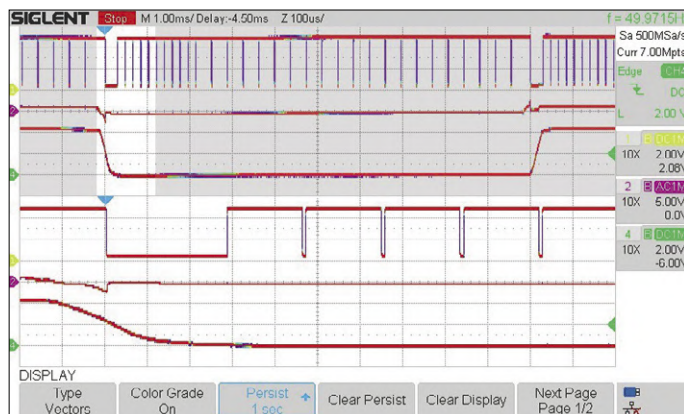


Bild 10. Einschalten einer LED-Lampe im Spannungsnulldurchgang. Verläufe über 1 s. CH1: Gate-Impulse; CH2: Spannung über Triac; CH4: Nulldurchgangs-Signal.

telbar nach dem Einschalten los und schaltet die Last nach dessen Ablauf wieder aus. Diese Funktion dient zur Begrenzung der Leuchtdauer einer Lampe. Die Abschaltzeit ist im Programm hinterlegt.

Das Programm erzeugt im Ein-Zustand zur Ansteuerung des Triacs 10- $\mu$ s-Impulse im Abstand von 278  $\mu$ s. Da die Steuerimpulse nicht auf die Netzfrequenz bezogen sind, sind die Schaltzeitpunkte asynchron zur Netzspannung. Im schlechtesten Fall wird die Last erst 278  $\mu$ s nach dem Nulldurchgang eingeschaltet. Zu diesem Zeitpunkt hat die Netzspannung bereits 29 V erreicht und Spannungs-Peaks entstehen, die EMV-technisch problematisch sind. Das Oszillogramm von **Bild 8** zeigt das Geschehen über 1 s und verdeutlicht diesen Zusammenhang. Die Triggerimpulse wandern unkorreliert gegenüber dem Nulldurchgang. Je nach Abstand der Triggerimpulse zum Nulldurchgang ändert sich die Spannung über dem Triac.

Abhilfe bringt die nächste praxistaugliche Anwendung, die den Nulldurchgang detektiert und so den Triac synchron zum Nulldurchgang triggert. Abweichend zu den folgenden Messungen ist hier  $V_{CC}$  das Bezugspotential. Da T1 des Triacs auf  $V_{CC}$  liegt, sieht man hier die negative Polarität der Steuerimpulse am Gate.

### Schalter und Timer mit Nulldurchgangs-Detektion

Gegenüber der vorherigen App wird hier ein aus der Netzspan-

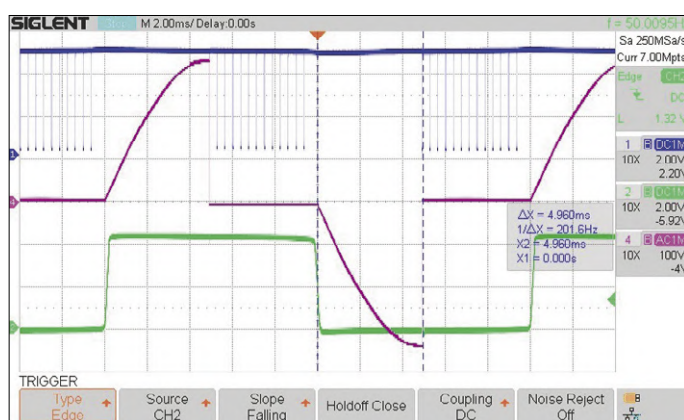


Bild 11. Dimmer mit Phasenanschnittsteuerung. CH1 (blau): Gate-Impulse; CH2 (grün): Nulldurchgangs-Signal; CH4 (rot): Spannung über Triac.

nung abgeleitetes Signal dazu benutzt, den Nulldurchgang der Netzspannung zu detektieren. Dieses Signal am Interrupt-Eingang INT0 löst mit der fallenden Flanke einen Interrupt aus und kennzeichnet so den Beginn der ersten Halbwelle. Das Ende dieser Halbwelle und der Beginn der folgenden Halbwelle wird über den Timer2 bestimmt. Dies spart aufwändigere Hardware zur präzisen Detektion beider Nulldurchgänge.

Anzeige



**Leiterplatten online**  
konfigurieren & bestellen

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



BERECHNEN  
SIE IHREN  
PREIS  
**ONLINE**



LIEFERUNG  
**2 bis 12**  
TAGE



ANZAHL  
**1 bis 50**  
STÜCK





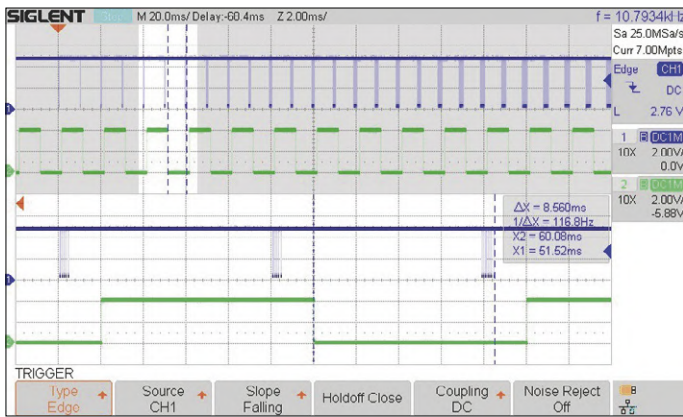


Bild 12. Ansteuerimpulse nach einem Neustart. CH1 (blau): Gate-Impulse; CH2 (grün): Nulldurchgangs-Signal. Der vergrößerte Bereich unten zeigt die Zunahme von fünf auf sieben Impulse.

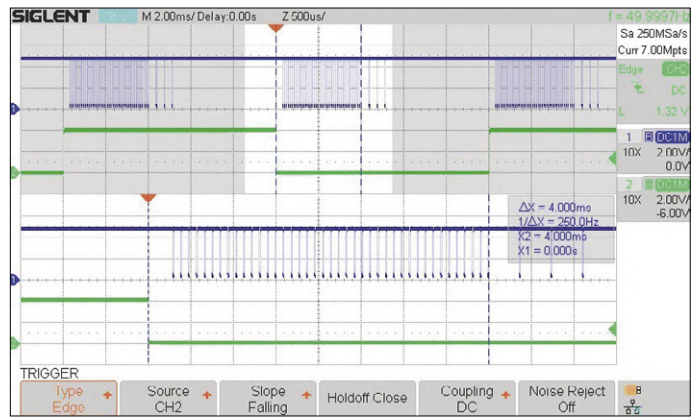


Bild 13. Ansteuerimpulse im eingeschwungenen Zustand mit vergrößertem Bereich unten. CH1 (blau): Gate-Impulse; CH2 (grün): Nulldurchgangs-Signal. Nach jedem Nulldurchgang werden 5 ms lang Füllimpulse erzeugt.

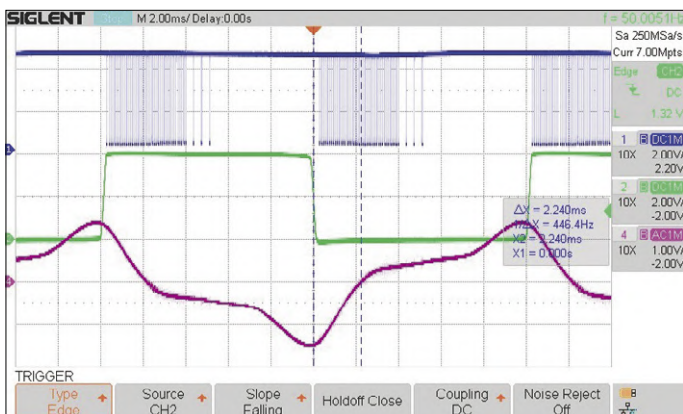


Bild 14. Trafostrom im Leerlauf. CH1 (blau): Gate-Impulse; CH2 (grün): Nulldurchgangs-Signal; CH4 (rot): Trafostrom.

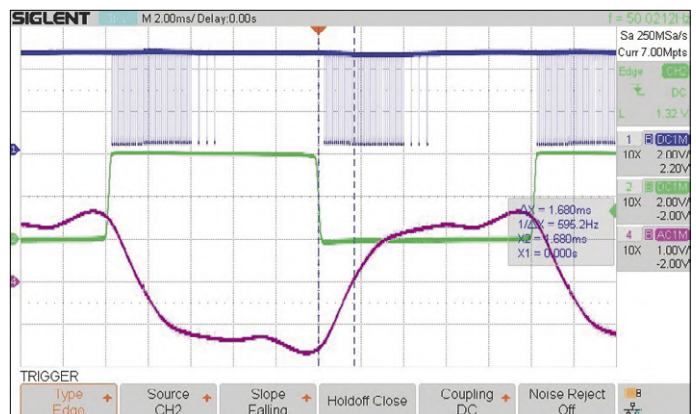


Bild 15. Trafostrom bei kleiner Last. CH1 (blau): Gate-Impulse; CH2 (grün): Nulldurchgangs-Signal; CH4 (rot): Trafostrom.

Für sicheres Triggern bei geringen Lastströmen ist der erste Triggerimpuls jeder Halbwelle auf 280  $\mu$ s verlängert. Es folgen dann noch weitere 10- $\mu$ s-Triggerimpulse. Dadurch wird auch bei hochohmigen Lasten  $I_L$  des Triacs überschritten. **Bild 9** demonstriert diesen Zusammenhang. Auffällig sind die kurzen Anstiege der Spannung über dem Triac auf ca. 3 V am Ende jeder Halbwelle. Der geringe Strom der angeschlossenen LED-Lampe fällt nämlich zum Ende der Halbwelle unter  $I_H$ . Dies hat aber nur wenig Einfluss auf entstehende Störpegel. **Bild 10** zeigt die Verläufe über 1 s übereinander. Die Ansteuerimpulse sind synchron zur Netzspannung.

## Dimmer

Eine typische Triac-Anwendung ist ein Dimmer zur Einstellung der Helligkeit von Lampen. Die Leistung der Lampe wird dabei per Phasenanschnittsteuerung eingestellt. Variiert wird hier der Einschaltzeitpunkt der Last in Bezug auf den Nulldurchgang jeder Netzhalbperiode ( $\approx$  Phasenwinkel). Dimmer können ziemliche EMV-Probleme verursachen, da je nach Phasenwinkel hohe Spannungsflanken entstehen. Im schlimmsten Fall wird beim Scheitelwert der Netzspannung (Phasenwinkel = 90°) geschaltet. Beim Zünden fällt die Spannung über dem

Triac schnell vom Scheitelwert auf null (**Bild 11**). Je nach Last entstehen dabei sehr hohe Ströme und damit hohe Störpegel. Ein vorgeschalteter EMV-Filter (z.B. aus einem ausgedienten Dimmer) ist daher ein Muss.

Viele LED-Lampen funktionieren aufgrund ihrer Art der Stromversorgung nicht mit herkömmlichen Dimmern. Auch dieses Modul hat mit manchen LED-Lampen Probleme. Die verwendete Lampe sollte daher dimmbar sein.

Interessant an der Triac-Ansteuerung ist hier, dass nach dem ersten Impuls weitere Impulse bis zum Ende der Halbwelle eingefügt werden. Dies ermöglicht die Verwendung von komplexeren Lasten. Bei Vollaussteuerung wird der erste Impuls nach dem Nulldurchgang auf 190  $\mu$ s verlängert. Dadurch wird  $I_H$  auch bei hochohmigen Lasten sicher überschritten.

Die Ausgangsleistung wird mit den beiden Tastern oder via Optokoppler eingestellt.

## Sanftes Einschalten induktiver Lasten

Induktive Lasten, speziell Trafos, können beim Einschalten sehr hohe Ströme ziehen. Der Spannungsnulldurchgang ist hier der problematischste Einschaltzeitpunkt. Auch der Restmagnetismus des Kerns kann Probleme bereiten, denn durch ihn ist der

## Über den Autor

Roland Stiglmayr ist Elektronik-Ingenieur mit Schwerpunkt Informationstechnik und hat 40 Jahre Berufserfahrung in Forschung und Entwicklung in den Bereichen Steuerungstechnik, HF-Technik, Mobilfunk, LWL-Technik und Datenübertragung. Heute ist er in beratender Funktion tätig.

Verlauf des Einschaltstroms nicht symmetrisch zur Nulllinie. Das B-Feld (siehe Hystereseschleife) des Kerns ist in eine Richtung verschoben und kommt somit leicht in den Bereich der Sättigung. Dann verliert die Wicklung schlagartig ihre Induktivität und der weiter steigende Strom wird fast nur noch vom Wicklungswiderstand etc. begrenzt. Ringkerntrafos sind davon besonders betroffen. Deshalb sollten sie deutlich höher als rein für den Laststrom erforderlich abgesichert werden.

Gelegentlich wird per Heißeiter oder Vorwiderstand mit Verzögerungsrelais der Einschaltstrom begrenzt, bis sich ein symmetrisches Magnetfeld und damit ein sauberer Wechselstrom einstellt. Diesem Problem kann man theoretisch auch durch langsames Hochfahren des Einschaltstromwinkels begegnen. Beim Einschalten des Trafos startet man mit einem Phasenwinkel von etwa  $180^\circ$  und reduziert ihn dann langsam bis auf  $0^\circ$ . Genau das ist in dieser Anwendung realisiert.

**Bild 12** zeigt die Impulsfolge nach einem Neustart. Der erste Trigger-Impuls wird ca. 8,6 ms nach dem Nulldurchgang erzeugt. Nach jeder Vollwelle wird ein weiterer Impuls davor hinzugefügt, bis die Zeit auf rund 0,3 ms reduziert ist. **Bild 13** zeigt den eingeschwungenen Zustand. Innerhalb der ersten 3 ms geht der Laststrom aufgrund der Phasenverschiebung durch Null. Um einen kontinuierlichen Stromfluss zu erhalten, ist es erforderlich, die Ansteuerimpulse in diesem Intervall eng zu setzen.

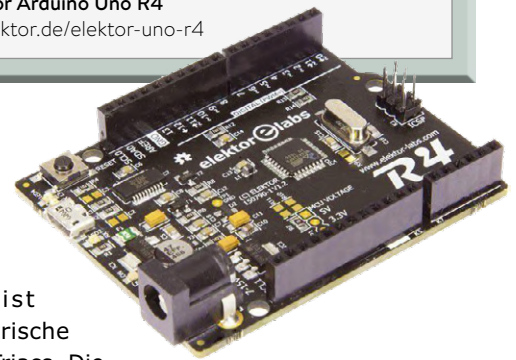
**Bild 14** stellt den Trafostrom im Leerlauf dar. Das Nulldurchgangs-Signal zeigt den Bezug zur Netzspannung. Zu sehen ist der Magnetisierungsstrom, der im Spannungsnulldurchgang sein Maximum erreicht. Der weitere Anteil resultiert aus den Eisen- und Kupferverlusten des Trafos. Dass der Strom nicht exakt  $90^\circ$  nacheilt, liegt an Verlusten und Nichtlinearitäten des Kerns. **Bild 15** zeigt den Trafostrom unter Last. Laststrom und Magnetisierungsstrom sind überlagert. Die Phasenverschiebung wird durch die Last kleiner.



## IM ELEKTOR-STORE

→ Elektor Arduino Uno R4

[www.elektor.de/elektor-uno-r4](http://www.elektor.de/elektor-uno-r4)



## Voraussetzung

für die fehlerfreie Funktion ist die völlig symmetrische Ansteuerung des Triacs. Die Ansteuerimpulse müssen bei beiden Halbwellen den gleichen Abstand zum Nulldurchgang haben, damit auch der Stromflusswinkel symmetrisch ist. Hier spielt  $I_{GT}$  und besonders  $I_H$  eine Rolle. Durch Verstärkung der vorderen Flanke des Gate-Stroms mithilfe des RC-Glieds wird sicheres Triggern gewährleistet. Damit der Triac leitend bleibt, muss der Trafostrom höher als  $I_H$  sein. Ringkerntrafos sind wegen ihrer hohen Induktivität problematisch. Doch eine kleine, resistive Zusatzlast (R25...R27; optional) löst das Problem. Manche Ringkern-Trafos erfordern sogar eine zusätzliche, externe Last.

Den Triac nur mit Impulsen anzusteuern war gerade bei Trafos eine Herausforderung bei der Programmierung der Firmware. Dank der ausführlichen Kommentare im Code sind die Tricks leicht verständlich.

Viel Spaß mit den beschriebenen Apps! Vielleicht fällt Ihnen ja eine neue Anwendung ein? Mit einer modifizierten Firmware ist das schnell gemacht. ◀

190047-01

## Weblink

[1] Elektor-Webseite zum Projekt:  
[www.elektormagazine.de/190047-01](http://www.elektormagazine.de/190047-01)

## Nachruf: Gerhard Haas

Mit Bestürzung haben wir vernommen, dass unser langjähriger Autor Gerhard Haas Ende Januar im Alter von 68 Jahren verstorben ist. In der Elektronik-Szene war Gerhard Haas als begeisterter Experte zum Thema Röhren(verstärker) bekannt. Er hat zahlreiche Fachbücher und viele Elektor-Artikel verfasst, die vor allem in den beliebten „Röhren“-Specials veröffentlicht wurden. Sein immenses Wissen hat der freiberufliche Entwickler sehr gerne auch in Seminaren und Vorträgen weitergegeben. Unvergessen ist sein Workshop auf der Elektor-Hausmesse Elektor live!, den er buchstäblich über Nacht zusammenstellte, um als Ersatz für einen erkrankten Kollegen einzuspringen. Nicht nur bei den Röhrenfans unter unseren Lesern hatte der Vollblut-Elektroniker deshalb einen hervorragenden Ruf.

Der langjährige Elektor-Redakteur Raimund Krings, der Leiter der Abteilung Bücher und Specials Ferdinand te Walvaart und die anderen Kollegen der Redaktion und des Verlags betrauern den Verlust des „Röhren-Papstes“ und Menschen Gerhard Haas sehr. Sabine Haas und den Freunden Michael Kaim und Norman Weber möchten wir auf diese Weise unser Beileid aussprechen.



# GEWUSST WIE Wie man etwas von GitHub herunterlädt

## GitHub für Dummies



Von **Clemens Valens** (Elektor Labs)

GitHub ist der Ort, an dem Menschen den Quellcode für alle Arten von Open-Source-Anwendungen teilen können, von einfachen Arduino-Sketches auf einer Seite bis hin zu riesigen Cloud-Anwendungen, die aus vielen Repositories mit Tausenden von Dateien bestehen. Darüber hinaus findet man bei Github auch andere Dinge wie CAD-Dateien oder einfache Textdokumente.

### Was ist GitHub?

GitHub ist eine Drehscheibe für Git, wobei Git ein Versionierungssystem ist, das die Änderungen an jedem ihm bekannten Dokument verfolgt. Der Hauptzweck von Git und GitHub ist es, Gruppen von Menschen die Möglichkeit zu geben, gemeinsam am selben Software-Projekt zu arbeiten.

Da GitHub ein Werkzeug für die professionelle und produktive Softwareentwicklung ist, kennt es viele Funktionen und Optionen, die seine Benutzung kompliziert machen, aber von den meisten Menschen nie gebraucht werden.

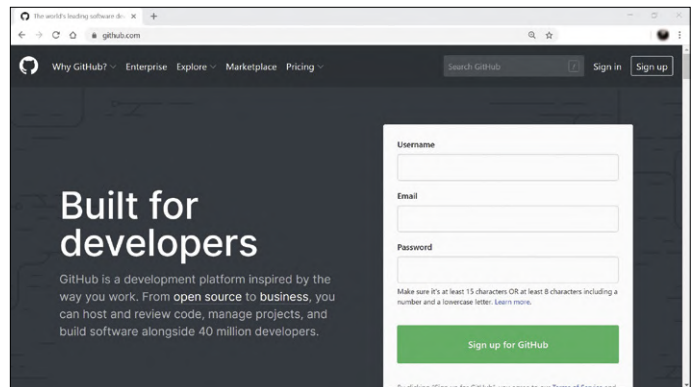
Auch bei Elektor veröffentlichen wir Code auf GitHub und ermöglichen unseren Lesern, ihn von dort herunterzuladen. Wir werden dann oft gefragt, wie man denn etwas von GitHub herunterladen könne. Erfahrenen Benutzern mag diese Frage dumm erscheinen, doch es stimmt, dass man die Dateien, die man in einem Repository aufgelistet sieht, nicht so einfach herunterladen kann. Es gibt keine Schaltfläche zum Speichern oder Herunterladen, und wenn Sie auf eine Datei klicken, wird sie in einem Browser geöffnet und nicht heruntergeladen. Wenn Sie mit der rechten Maustaste auf eine Datei klicken und sie dann speichern, erhalten Sie eine Datei, die HTML-Code und nicht den erwarteten Inhalt enthält.

### Raw-Modus

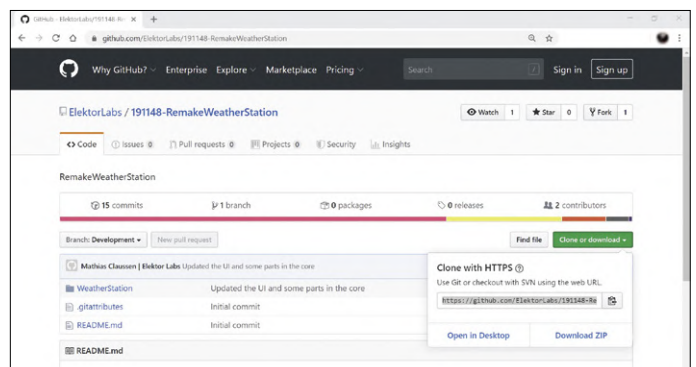
Eigentlich könnten Sie eine Datei ganz einfach herunterladen, wenn Sie sie zuerst im Raw-Modus öffnen und dann mit der rechten Maustaste klicken, um sie als Datei zu speichern. Bei ein paar Dateien ist das vielleicht praktikabel, aber sicher nicht, wenn es viele Dateien gibt.

### Der grüne Knopf *Clone or download*

Es gibt eine einfache Schaltfläche zum Herunterladen, die jedoch nur sichtbar ist, wenn Sie sich im Stammverzeichnis des Repositories befinden. Dann ist es ganz einfach: Klicken Sie einfach auf die grüne Schaltfläche mit der Aufschrift *Clone or download* und lassen Sie sich nicht durch die überraschenden Optionen *Clone with HTTPS*, *Open in Desktop* und *Download ZIP* verwirren. Klicken Sie auf diese letztgenannte Option *Download ZIP* und ignorieren Sie (zunächst) die anderen Optionen. Sie müssen sich dazu nirgendwo anmelden oder ein Konto einrichten.



Obwohl es jeder benutzen können sollte, ist Github für professionelle Software-Entwickler gedacht und besitzt daher viele Funktionen, die die meisten Menschen nie benötigen.



Der grüne Knopf *Clone or download* ist nur im Stammverzeichnis des Repositories sichtbar.

### Arduino und GitHub

Der Name der von Ihnen heruntergeladenen Datei kann mit „-master“ enden. Wenn Sie die Datei entpacken, erhalten Sie einen Ordner mit dem gleichen Namen. Meistens ist das nicht wirklich wichtig, aber wenn es sich bei dem Download um einen Arduino-Sketch handelt, spielt das doch eine Rolle. Für die Arduino-IDE muss nämlich der Name der ino-Datei mit dem Namen des Ordners übereinstimmen. Die Lösung ist einfach: Entfernen Sie den Teil „-master“, indem Sie den Ordner umbenennen. Normalerweise müssen Sie dies aber nicht tun, wenn sich der Sketch in einem Ordner befindet, der in der ZIP-Datei verschachtelt ist, da dieser Ordner dann denselben Namen wie die ino-Datei trägt.

Wenn es sich bei dem Download um eine Arduino-Bibliothek handelt, dann weiß die Arduino-IDE, wie sie damit umgehen muss. Sie können sie in der Arduino-IDE unter *Sketch->Bibliothek einbinden->ZIP-Bibliothek hinzufügen* importieren.



## Releases

Viele Projekte auf GitHub entwickeln sich ständig weiter. Wenn Sie die ZIP-Datei wie zuvor herunterladen, erhalten Sie einen sogenannten „development snapshot“, der möglicherweise nicht funktioniert oder nicht das tut, was Sie von ihm erwarten. Aus diesem Grund werden stabile Versionen, sogenannte Releases, veröffentlicht, die getestet wurden und (hoffentlich) einwandfrei funktionieren. Diese Versionen werden oft auf einer anderen Website gehostet, zum Beispiel auf der offiziellen Website des Projekts, aber auch GitHub bietet Platz dafür. Prüfen Sie auf der Registerkarte *Releases*, ob es solche stabile Versionen gibt und wenn ja, laden Sie die neueste oder, wenn Sie möchten, auch eine ältere stabile Version herunter. Sie sollten die Registerkarte *Releases* auch dann überprüfen, wenn Sie nicht am Quellcode des Projekts interessiert sind, sondern nur an vorkompilierten Binärdateien, die auf Ihrer Plattform laufen sollen. Solche Binärdateien sind wichtig, wenn die Kompilierung des Projekts für Sie zu kompliziert oder sogar unmöglich ist, weil Sie nicht über die entsprechenden Werkzeuge verfügen.

## Klonen eines Repositorys

Schließlich können Sie die offizielle GitHub-Methode verwenden, um ein Projekt herunterzuladen, und zwar durch Klonen. Damit dies funktioniert, müssen Sie zuerst Git auf Ihrem Computer installieren, das über die Kommandozeile bedient wird (<https://git-scm.com/downloads>). Es gibt aber auch Git-Clients mit schönen grafischen Benutzeroberflächen - ein Beispiel für Windows ist GitHub Desktop (<https://desktop.github.com>).

In jedem Fall benötigen Sie die vollständige URL des Repositorys, das Sie klonen wollen. Dazu drücken Sie wieder die Schaltfläche *Clone or download*, aber kopieren diesmal die URL im Feld *Clone with HTTPS*, indem Sie auf den kleinen Button rechts neben der URL klicken.

## Klonen über die Befehlszeile (CLI)

Navigieren Sie im Befehlszeilen-Interface zunächst zu dem Ordner, in dem Sie den Klon des Repositorys erstellen möchten, geben Sie dann den Befehl `git clone` ein, hängen Sie die URL des Repositorys an und drücken Sie die Eingabetaste. Wenn alles in Ordnung ist, wird der Klonvorgang gestartet. Zum Beispiel so:

```
git clone https://github.com/ElektorLabs/180468-ESP32-Weatherstation.git
```

## Klonen mit GitHub Desktop

Alternativ können Sie GitHub Desktop verwenden. Wählen Sie im Menü *File* die Option *Clone repository*, navigieren Sie zum Zielordner und fügen Sie die zuvor kopierte URL in das Feld *Repository-URL* ein. Klicken Sie dann auf *Clone*.

## Klonen ist einfacher als Herunterladen

Wenn Sie wissen, wie man das macht, ist das Klonen wahrscheinlich einfacher als das Herunterladen. Sie müssen nichts entpacken und können direkt in den Ordner klonen, in dem Sie das Ganze haben wollen. Und Sie bekommen auch keinen Anhang *-master*, der dem Namen des Repositorys hinzugefügt wird. ◀

200047-02

Anzeige



**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!

# Mein IoT-Button: Der Knopf fürs Netz

## Teil 2: Prototyping mit Board und Cloud

Von Dr. Veikko Krypczyk

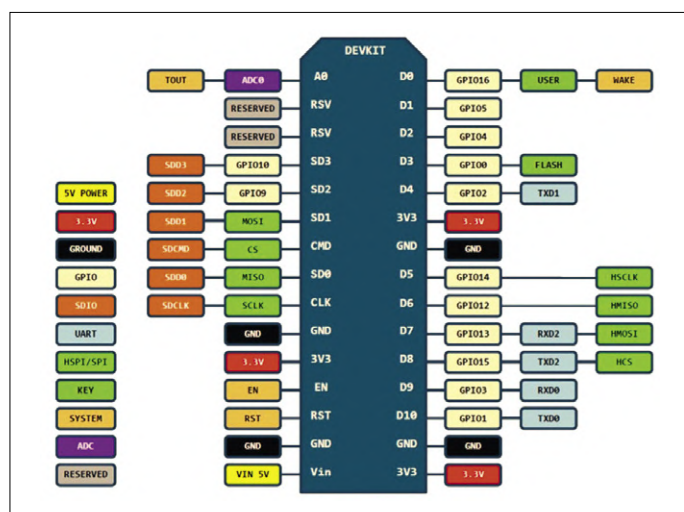
Die Idee und das Konzept eines IoT-Buttons basieren darauf, dass man über eine minimalistische Hardware mit einem entfernten Service in der Cloud kommuniziert. Dazu ist es notwendig, dass man eine Verbindung zum lokalen Netzwerk herstellt. Kleine und günstige Mikrocontrollerboards wie ein ESP8266 NodeMCU sind für diese Aufgabe prädestiniert. Wir beleuchten die relevanten Aspekte auf Seiten des IoT-Devices und stellen Ideen für eigene Projekte vor. Das Beste: Alles ist relativ einfach zu bewerkstelligen und Erfolgserlebnisse stellen sich rasch ein. Das liefert Spielräume für eigene Ansätze.

Das Internet of Things (IoT) macht alle erdenklichen Geräte „intelligent“ und verbindet sie über das Internet mit der Außenwelt. Im ersten Teil des Artikels haben wir die grundsätzliche Architektur einer IoT-Landschaft skizziert, bestehend aus Sensoren/Aktoren und einem Server-Backend. Wir haben gesehen, dass man die serverseitige Infrastruktur in der Regel nicht selbst aufbauen und programmieren muss, sondern sich aus dem vielfältigen Cloud-Angebot bedienen kann. Die Kommunikation erfolgt dabei über das Internet. Sehr einfach gestaltet sich die Angelegenheit, wenn man dabei auf etablierte Standards setzt. Für Abfragen hat sich die Kommunikation auf der Basis

von *REST* bewährt. Der Client, in diesem Fall das IoT-Gerät, kann Daten an Dienste (auf einem Server) übermitteln oder abfragen. In diesem zweiten Teil des Artikels soll es um die Hard- und Software auf Seiten des IoT-Devices gehen.

### Auswahl der Hardware

Welche Hardware wollen wir einsetzen? Grundsätzlich gibt es dazu einige Optionen, zum Beispiel einen Minicomputer wie den **Raspberry Pi**. Neben WLAN bietet der RPi auch genügend Rechenpower für alle Anwendungen. Wollen wir am gleichen Ort mehrere Buttons, Sensoren und Aktoren mit unterschied-



lichen Funktionen betreiben, so wären all diese Geräte mit den GPIO-Pins des Raspberry Pi verbunden. Der RPi würde über das Netzwerk die Verbindung zum Service in der Cloud herstellen. Für kleinere Aufgaben ist ein Raspberry Pi in seinen Abmessungen aber zu groß und würde auch eine dauerhafte Stromversorgung über ein Netzteil benötigen.

Schauen wir auf den **Arduino Uno**! Hierbei handelt es sich bekanntermaßen nicht um einen kompletten Rechner, sondern nur um ein Mikrocontrollerboard. Aber Hard- und Softwaremöglichkeiten sind umfassend und es gibt unzählige Bibliotheken und Anwendungsbeispiele. Die Programmierung kann komfortabel über die Arduino-IDE erfolgen. Von Haus aus kann ein Arduino jedoch mit einem WLAN nichts anfangen, so dass wir diese Funktion über ein Zusatzmodul nachrüsten müssten. Aber noch viel einfacher geht es mit einem Mikrocontrollerboard des Typs **ESP8266 NodeMCU** (siehe [1] und **Kasten**), das alle benötigten Funktionen und vor allem ein WLAN-Interface in sich vereint (**Bild 1**).

An den Anschluss D7 des Boards schließen wir lediglich eine Taste gegen GND an. Das Betätigen der Taste soll später über WLAN weitergeleitet werden. Eine Status-LED befindet sich bereits auf dem Board, so dass wir keine weiteren Bauteile benötigen. Die Stromversorgung erfolgt (zunächst) über den Micro-USB-Anschluss des Computers.

### Einrichtung der ESP8266-NodeMCU

Ebenso wie Arduino-Boards können ESP8266-Mikrocontroller über die Arduino-IDE programmiert werden. Zunächst muss man das Board mit einem Datenkabel über den USB-Anschluss mit dem PC verbinden und gegebenenfalls noch den Treiber für den Chipsatz CH340 [2] installieren. Wenn nicht schon längst geschehen (dann könnten Sie diesen Abschnitt überspringen), installieren Sie die Arduino-IDE auf Ihrem Rechner [3]. Die IDE muss um eine zusätzliche Komponente erweitert werden: Unter *Datei -> Voreinstellungen* trägt man bei *Zusätzliche Boardverwalter-URLs* den Link [4] ein (**Bild 2**). Schließen Sie das Dialogfeld mit *OK*.

Die eigentliche Installation erfolgt unter dem Menüpunkt *Werkzeuge*, wo der Eintrag *Generic ESP8266 Module* zu finden sein sollte. Man wählt ihn aus und klickt im aufpoppenden Menü oben auf den *Boardverwalter*, schränkt den Suchbereich mit dem Eintrag *NodeMCU* ein und nimmt die Installation vor (**Bild 3**).

Ob alles erfolgreich war, kann man mit einem einfachen „Hello World“-Programm testen (**Listing 1**). Kopieren Sie dazu den Code und laden Sie ihn mit *Sketch|Hochladen* in den Mikrocontroller. Die LED auf dem Board wird von der kleinen Schleife mit `digitalWrite(LED, HIGH)` für 3 s aus- und von `digitalWrite(LED, LOW)` für 1 s eingeschaltet.

Um einen Taster (an Pin D7) abzufragen, ist ebenfalls nur sehr wenig Quellcode notwendig. Zum Beispiel überprüft

```
int button=D7;
status=digitalRead(button);
if (status==LOW)
{ ..... }
```

ob der Taster geschlossen wurde, um nachfolgende Vorgänge auslösen.

Jetzt muss der ESP8266 nur noch Zugang zum Internet erhalten. Die Hardware dazu befindet sich in Form eines winzigen

### Listing 1. Ein „Hello World“ (blinkende LED) auf dem ESP8266.

```
#define LED D4

void setup()
{
    pinMode(LED, OUTPUT);
}

void loop()
{
    digitalWrite(LED, HIGH);
    delay(3000);
    digitalWrite(LED, LOW);
    delay(1000);
}
```



Bild 3. Installation der Bibliotheken für das Board ESP8266 NodeMCU.

## Die ESP8266 NodeMCU

Bei dem Mikrocontroller-Board ESP8266 NodeMCU handelt es sich um einen ESP8266-Chip mit einem Takt von 80 MHz und bis zu 4 MB Flash-Speicher. Auf der ESP8266 NodeMCU befinden sich ein WLAN-Modul und eine USB-Schnittstelle inklusive UART-/ USB-Wandler (CH340G). Das kleine Board hat die folgenden technischen Daten:

- unterstützt WPA/WPA2
- Chip: ESP8266
- Prozessor: Tensilica L106 mit einer Taktrate von 80 MHz
- Systemarchitektur: 32 Bit, Flash: bis zu 4 MB
- Unterstützte Protokolle: SPI, UART, I2C, I2S, IR Remote Control, PWM, GPIO
- Logik-Level: 3,3 V
- Stromaufnahme: < 10 uA im Sleep-Modus und bis zu 200 mA unter Last
- Spannungsversorgung und Programmieranschluss über Micro-USB: 5 V
- Programmierchip: CH340G
- Größe: 5,5 x 3,1 x 1,2 cm; Gewicht: 42 g

Mit diesen Spezifikationen ist das Board ESP8266 NodeMCU sehr gut geeignet, um kleine und vor allem energiesparende IoT-Geräte zu betreiben.



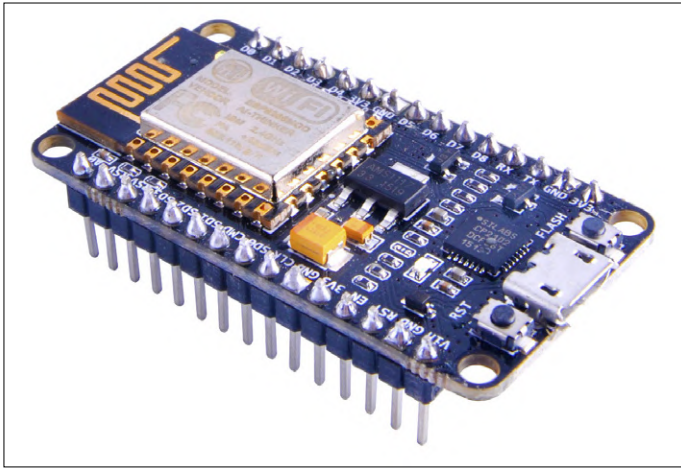


Bild 4. Die ESP8266 NodeMCU mit WLAN-Modul.

(graugrünen) WLAN-Moduls schon auf der Platine des ESP8266 (**Bild 4**).

Wenn man das WLAN des ESP8266 nutzen möchte, muss man in den Header des Quelltextes eine Bibliothek mit der Klausel `#include ESP8266WiFi.h` einbinden. Die Bibliothek erwartet die korrekten Daten für die Verbindungsaufnahme zum Internet, die SSID und das Passwort. Diese Daten werden (zunächst statisch) im Quelltext hinterlegt, später dann soll der Nutzer die Daten natürlich konfigurieren können. Das Herstellen der WLAN-Verbindung ist wunderbar unkompliziert (**Listing 2**).

Konzeptionell müssen wir jetzt so vorgehen, dass wir bei einem eintreffenden Ereignis eine Message über das Netzwerk senden. Als Empfänger einer solchen Netzwerkkommunikation dient ein REST-Endpoint, wie er im ersten Teil des Artikels beschrieben wurde. Dieser Endpoint nimmt die Daten beziehungsweise die Message entgegen und kann sie weiter verarbeiten und eine Reaktion einleiten. Bei einem simplen IoT-Button müssen keine weiteren Daten als das Ereignis „Tastendruck“ übertragen werden. Würde man die Anwendung entsprechend

#### Listing 2. Eine Verbindung zum Internet herstellen.

```
include <ESP8266WiFi.h>
void setup()
{
    Serial.begin(115200);

    WiFi.begin(<SSID>, <Password>);

    while (WiFi.status() != WL_CONNECTED);
    {
        delay(1000);
        Serial.print("Connecting ...");
    }
}
```

erweitern, könnte man zusätzliche Daten als Parameter über die REST-Anfrage übermitteln. Probieren wir das Ganze aus!

#### Daten in die Cloud

Um Daten in die Cloud, also zu einem entfernten Server zu senden, ist ein REST-Endpoint erforderlich. Das muss man nicht selbst programmieren, sondern man kann sich bestehender Services bedienen. Für Tests und private Anwendungen sind diese Cloud-Dienste meist kostenlos. Wir wollen experimentell und schrittweise vorgehen. Unter [5] können Sie für Testzwecke eine kostenfreie REST-API erstellen, die von unserem Mikrocontroller aus über das WLAN erreicht werden kann. Das funktioniert mit wenigen Mausklicks, sogar ohne vorherige Anmeldung. Schauen Sie einmal unter [6]; wir haben sogar schon einen Endpoint erstellt (**Bild 5**)! Um zu testen, ob und wie dieser Endpoint vom Mikrocontroller aus erreicht wird, senden wir, wenn der Taster betätigt wird, eine GET-Anfrage an den Dienst. Dies wird in die Programmschleife eingesetzt (**Listing 3**). Ob der Mikrocontroller mit dem Internet verbunden ist, überprüft die Quelltextzeile:

#### Listing 3. GET-Anfrage an REST-Endpoint senden.

```
void loop()
{
    if (WiFi.status() == WL_CONNECTED)
    {
        status=digitalRead(button);
        if (status==LOW)
        {
            digitalWrite(LED, LOW);

            BearSSL::WiFiClientSecure client;
            client.setInsecure();
            HTTPClient https;
            https.begin(client, "https://iotbutton.free.
                           beeceptor.com");
            int httpCode = https.GET();
            if (httpCode == 200)
            {
                String payload = https.getString();
                Serial.println(payload);
            }
            else
            {
                Serial.print("Error: ");
                Serial.println(httpCode);
            }
            https.end();
        }
        else
        {
            digitalWrite(LED, HIGH);
        }
    }
}
```

```
if (WiFi.status() == WL_CONNECTED)
{ ..... }
```

Alle eingeschlossenen Zeilen des Quellcodes werden also nur abgearbeitet, wenn eine Verbindung mit dem Internet besteht. Der nächste Schritt

```
status=digitalRead(button);
if (status==LOW)
```

prüft, ob der Button gedrückt wurde. Ist dies der Fall, wird ein http-Client initialisiert und damit die Option aktiviert, *https*-Kommandos zu übertragen. Dazu ist eine Bibliothek erforderlich, die im Header mit `#include` eingebunden werden muss. Um den URL für unseren REST-Endpoint anzugeben, wird der Wert `https://iotbutton.free.beeceptor.com` gesetzt. Es folgt eine einfache GET-Abfrage an den Endpunkt, deren Ergebnis wir durch die Zeile

```
String payload = https.getString();
```

als Zeichenkette erhalten.

Als direkte Reaktion auf das Betätigen des Tasters soll die LED auf dem Board ein- und nach dem Loslassen wieder ausgeschaltet werden (siehe oben). Ob die Daten auch in der Cloud angekommen sind, lässt sich online nachverfolgen. Gehen Sie zum Endpunkt auf der Beeceptor-Seite [6] und verfolgen Sie die Anfragen an den Endpunkt (**Bild 6**).

Beachten Sie, dass man über diesen Dienst nur 50 Anfragen pro Tag kostenfrei senden darf. Im Rahmen der Experimente haben wir diese Zahl überschritten, so dass wir auf die GET-Anfrage als Rückgabewert eine Fehlermeldung mit dem Code 429 erhielten. Sofern Sie aber die maximale Zahl der Anfragen nicht überschreiten, werden der Statuscode 200 (OK) und die Daten der REST-Anfrage an den Mikrocontroller zurückgegeben. Sie können in der Entwicklungsumgebung im Seriellen Monitor (*Werkzeuge*|*Serieller Monitor*) verfolgen, welche Rückgabewerte der Mikrocontroller empfängt (**Bild 7**).

Wir haben damit bei einem Tastendruck eine REST-Anfrage über das Internet übermittelt. Damit ist die technische Seite des IoT-Buttons konzipiert. **Listing 4** zeigt den gesamten Quellcode. Die individuell anzupassenden Stellen sind fett markiert.

## Stromgeschichten und Schlafmodus

Wir hatten uns für das ESP8266-Mikrocontroller-Board insbesondere wegen des geringen Stromverbrauchs entschieden. Man kann in der Praxis von den in [7] genannten Werten für den Stromverbrauch ausgehen. Mit einem einfachen Arduino-Programm sind es etwa 50...70 mA, im WLAN-Betrieb steigt der Strombedarf bis auf etwa 80...170 mA und für wenige Millisekunden kann das Board sogar bis zu 400 mA ziehen. Interessant ist deshalb die Möglichkeit, den Energiehunger im Deep-Sleep-Modus erheblich zu dämpfen. Je nach Datenblatt und Internetquelle ist von wenigen Mikroampere die Rede. Damit sollte eine autarke Batterie (Akku) für eine monatelange Versorgung ausreichen. Die Versorgungsspannung wird an die Anschlüsse +5V und GND angeschlossen, zum Beispiel in Form einer Power-Bank. Es gibt zwei Varianten des Deep-Sleep-Modus.

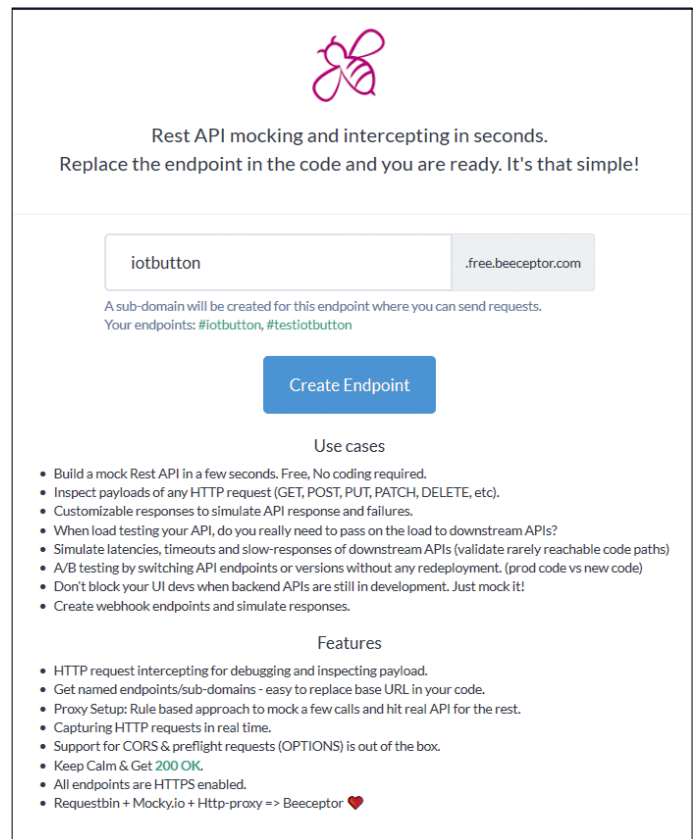


Bild 5. REST-Endpoint zum Testen auf der Beeceptor-Cloud.

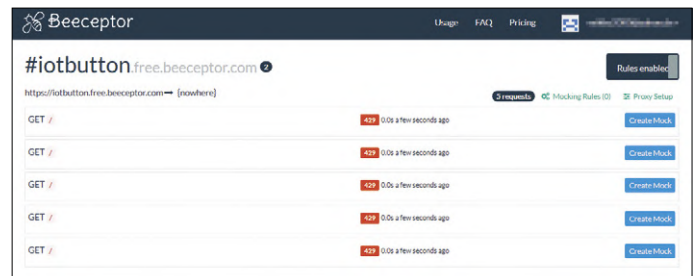


Bild 6. Protokollierte Anfragen an den REST-Endpoint auf der Beeceptor-Cloud.

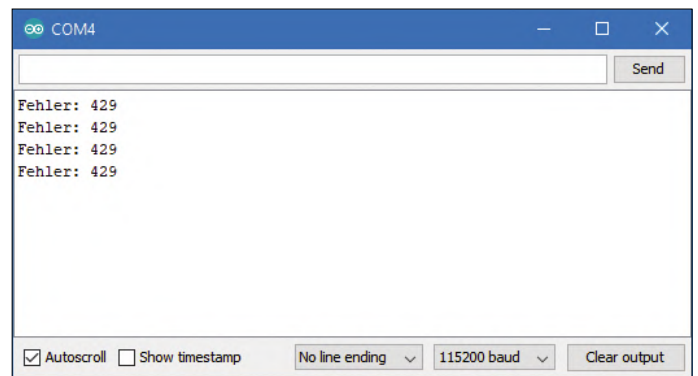


Bild 7. Im Seriellen Monitor werden die Antworten des REST-Endpunktes protokolliert. Hier erfolgt die Fehlermeldung 429 wegen Überschreitung der Anzahl der Anfragen.

**Listing 4. Senden einer GET-Anfrage via REST.**

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#define LED D4
int button=D7;
int status=1;
void setup()
{
    Serial.begin(115200);
    pinMode(LED, OUTPUT);
    digitalWrite(LED, HIGH);
    WiFi.begin(<SSID>, <Password>);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(1000);
        Serial.print("Connecting ...");
    }
}

void loop()
{
    if (WiFi.status() == WL_CONNECTED)
    {
        status=digitalRead(button);
        if (status==LOW)
        {
            digitalWrite(LED, LOW);
        }
    }
}
```

```
BearSSL::WiFiClientSecure client;
client.setInsecure();
HTTPClient https;
https.begin(client, "https://iotbutton.free.
                beeceptor.com");

int httpCode = https.GET();
if (httpCode == 200)
{
    String payload = https.getString();
    Serial.println(payload);
}
else
{
    Serial.print("Error: ");
    Serial.println(httpCode);
}
https.end();
}

else
{
    digitalWrite(LED, HIGH);
}
}
```

**Listing 5. Senden einer GET-Anfrage via REST und Deep-Sleep-Modus.**

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>

#define LED D4
void setup()
{
    Serial.begin(115200);
    pinMode(LED, OUTPUT);
    digitalWrite(LED, HIGH);
    WiFi.begin(<SSID>, <Password>);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(1000);
        Serial.print("Connecting ...");
    }
}

void loop()
{
    if (WiFi.status() == WL_CONNECTED)
    {
        digitalWrite(LED, LOW);
    }
}
```

```
BearSSL::WiFiClientSecure client;
client.setInsecure();
HTTPClient https;
https.begin(client, "https://iotbutton.free.
                beeceptor.com");

int httpCode = https.GET();
if (httpCode == 200)
{
    String payload = https.getString();
    Serial.println(payload);
    digitalWrite(LED, HIGH);
    delay(2000);
    ESP.deepSleep(0);
}
else
{
    Serial.print("Fehler: ");
    Serial.println(httpCode);
}
https.end();
}

}
```



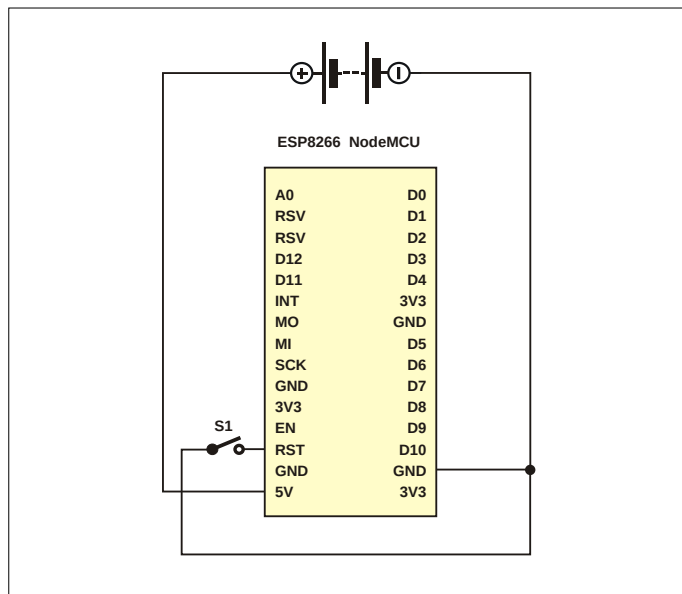


Bild 8. Der Schaltplan enthält neben dem Mikrocontroller nur einen Taster.

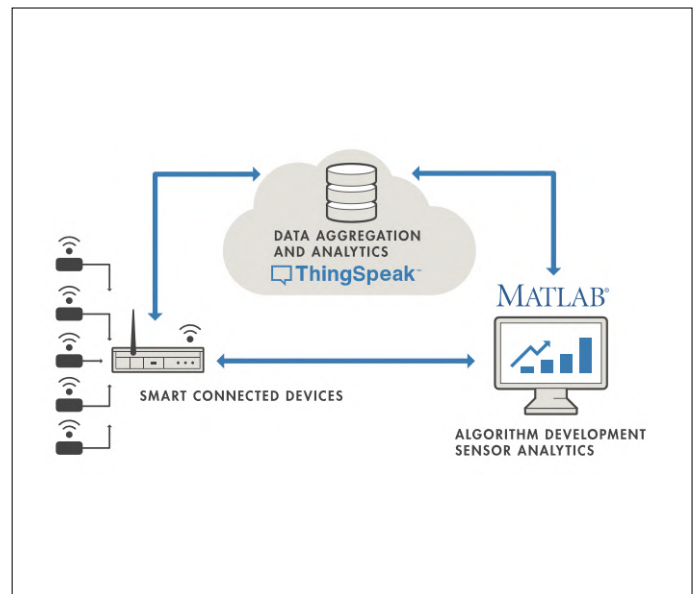


Bild 9. Zusammenarbeit von IoT-Device, IoT-Cloud und weiteren Diensten am Beispiel von ThingSpeak.

- **Software-Deep-Sleep:** Der ESP8266 wacht nach einer bestimmten Zeit eigenständig wieder auf. Dazu muss eine Verbindung von GPIO16 (D2) zum Anschluss RST hergestellt werden. Nach der eingestellten Zeit wird ein Signal gesendet, das den ESP eigenständig neu startet. Die maximale Zeitdauer ist hier jedoch auf circa 71 min begrenzt. In Szenarien, die zyklisch bestimmte Messwerte übermitteln, ist diese Option eine gute Wahl. Der zugehörige Befehl lautet: `ESP.deepSleep(SLEEP_TIME * 1000000)`
- **Interrupt-Deep-Sleep:** Der ESP8266 startet nicht eigenständig neu, sondern wird durch einen Interrupt aus dem Deep-Sleep-Modus geweckt. Diese Form des Deep-Sleep-Modus wird mit: `ESP.deepSleep(0)` gestartet. Dafür dürfen allerdings GPIO16 und RST nicht verbunden sein. Sobald das Ereignis ausgelöst wurde, sendet man ein LOW-Signal an den RST-Pin. Dadurch startet der ESP8266 neu und die Software wird ausgeführt.

Für den IoT-Button bedeutet dies, dass der Taster nun nicht mehr an einen GPIO-Pin geschaltet wird, sondern wie in **Bild 8** zu sehen an den Pin RST (und Masse). Im Code kann deshalb die Prüfung auf den Tastendruck entfallen. Stattdessen wird der Mikrocontroller nach dem erfolgreichen Senden der Informationen über die REST-Schnittstelle in den Schlafmodus versetzt (**Listing 5**). Kompilieren Sie den Quelltext (Sketch) und laden Sie ihn in den Controller. Nach dem Start verbindet sich dieser mit dem Netzwerk, setzt eine Message ab und geht in den Schlafmodus. Wird der Mikrocontroller über die Taste (Reset) geweckt, so wird der Vorgang von neuem angestoßen. Sie können online mitverfolgen, wie die REST-API erfolgreich via GET-Anforderung aufgerufen wurde.

### Noch einmal in die Cloud

Technisch haben wir den IoT-Button umsetzen können. Welche Aktionen man nunmehr erledigt, wenn die Message des Mikro-

### Weblinks

- [1] User-Manual ESP8266 NodeMCU: [www.handsontec.com/pdf\\_learn/esp8266-V10.pdf](http://www.handsontec.com/pdf_learn/esp8266-V10.pdf)
- [2] Treiber CH340: <https://sparks.gogo.co.nz/ch340.html>
- [3] Arduino-IDE: <http://www.arduino.cc/en/main/software>
- [4] Boardverwalter-URL: [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)
- [5] BEECEPTECTOR: <https://beeceptor.com/>
- [6] Console IoT-Button: <https://beeceptor.com/console/iotbutton>
- [7] Stromverbrauch ESP8266: <https://arduino-hannover.de/2018/07/25/die-tuecken-der-esp32-stromversorgung/>
- [8] Microsoft-IoT-Cloud: <https://azure.microsoft.com/en-en/services/iot-hub/>
- [9] ESP8266 in der Azure-Cloud: <https://sandervandeveld.wordpress.com/2019/05/07/connection-a-cheap-esp8266-to-azure-iot-central/>
- [10] NodeMCU in der Azure-Cloud: <https://www.thingforward.io/techblog/2018-05-22-connecting-nodemcu-to-microsoft-azure-iot-hub-part-1.html>
- [11] Thingspeak: <https://thingspeak.com/>

Bild 10. REST-Endpoint (Channel) bei ThingSpeak konfigurieren.

Bild 11. Apps bestimmen die Aktionen bei ThingSpeak für einen Channel.

controllers empfangen wird, lässt sich in einigen wenigen Schritten konfigurieren. Denn auch für diese Szenarien gibt es unterschiedliche Cloud-Dienste, zum Beispiel *Microsoft Azure IoT* [8][9][10] oder *ThingSpeak* [11]. Die Vorgehensweise läuft im Wesentlichen darauf hinaus, dass man online direkt im Cloud-Dienst einen Endpunkt konfiguriert und diesen über das Internet vom IoT-Device adressiert. Über den Dienst können dann die gewünschten Aktionen veranlasst werden (**Bild 9**). Erstellen Sie einen kostenfreien Account beim Dienst *ThingSpeak*, der nur über eine E-Mail-Adresse verifiziert werden muss, und dann einen neuen REST-Endpoint (*Channel*).

Füllen Sie das umfangreiche und rechts daneben erläuterte Formular (**Bild 10**) aus oder geben Sie dem Kind nun erst einmal einen Namen. Jeder Channel erhält einen eindeutigen URL und die Möglichkeit, über die Anfrage hinaus mittels URL-Parameter zusätzliche Daten zu senden. Für einen einfachen IoT-Button werden zwar keine weiteren Parameter benötigt, aber unter *My Channels/API Keys* sind die Lese- und Schreibschlüssel (und Anwendungsbeispiele, wie man die Keys in die GET-Anfragen übernimmt) des Channels zu finden. Ist ein Channel korrekt konfiguriert, können Sie über den Menüpunkt *Apps* festlegen, welche Aktionen ausgelöst werden sollen (**Bild 11**), zum Beispiel über einen Web-Dienst eine E-Mail zu versenden oder ein Tweet bei Twitter abzusetzen. Das IoT-Backend dient dabei gewissermaßen als Zwischenschicht.

**Gefahrenhinweis:** IoT-Devices, die mit Diensten im Netzwerk kommunizieren, benötigen einen Zugriff auf die Dienste. Dazu müssen sie sich natürlich identifizieren, beispielsweise mit Username und Passwort oder eindeutigen Merkmalen (Keys). Diese Werte müssen im Quellcode hinterlegt werden. Achten Sie darauf, dass Sie diese Informationen zum Beispiel beim Kopieren des Quellcodes oder bei Veröffentlichung im Internet (zur freien Verfügung...) schützen.

## Fazit und Ausblick

Experimente mit dem ESP8266-Mikrocontroller sind faszinierend. Keine oder nur sehr wenige zusätzliche Bauteile sind notwendig, um IoT-Szenarien umzusetzen. Um das Projekt des IoT-Buttons „produktionsfähig“ zu gestalten oder es sogar in eine kommerzielle Anwendung zu verwandeln, ist nicht mehr viel Arbeit notwendig. Meist geht es nur um mechanische Randbedingungen (Gehäuse) und die Energieversorgung (Akku dauerhaft anschließen). Auch Erweiterungen sind denkbar: Wird der Button beispielsweise durch einen Bewegungsmelder ausgelöst, so können Sie direkt „Alarm“ schlagen und per Internet kommunizieren, wenn sich jemand unberechtigterweise dem gesicherten Objekt nähert. Die eigentliche Arbeit erledigt man online über die Konfiguration in der Cloud. ◀

190303-B-01

### IM ELEKTOR-STORE

→ ESP8266 NodeMCU

www.elektor.de/17952

# Sprachsynthesizer TMS0280

## Bemerkenswerte Bauteile

Von **Neil Gruending** (Kanada)

Im Jahr 1976 wollte Texas Instruments (TI) ein Gerät entwerfen, das ein auf einer Tastatur eingegebenes Wort ausspricht, um Rechtschreibung zu lehren. Es sollte kostengünstig sein, also so viele Wörter wie möglich in dem etwa 256 kBit großen Speicher beherbergen. Außerdem musste es mobil sein, so dass auch die Abmessungen und der Stromverbrauch eine wichtige Rolle spielten. Schon im Jahr 1978 hatte TI das Vorhaben in die Tat umgesetzt! Dank des Sprachsynthese-ICs TMS0280 konnte (scheinbar) die Leistungsfähigkeit eines mehrere tausend Dollar teuren Computers in ein 50-Dollar-Gerät gepackt werden, das wir alle als *Speak&Spell* kennen.

TI hatte in dieser Zeit bereits den TMS1000-Mikrocontroller (siehe Elektor 03/2020) entwickelt, der aber nicht leistungsfähig genug war, um die Sprachsynthese allein durchzuführen. Deshalb setzte TI ein Team mit Larry Brantingham, Paul Breedlove, Richard Wiggins und Gene Frantz an diese Aufgabe. In vielen Recherchen stellten sie fest, dass die Verbraucher eine dumpfe, monotone Stimme mit computergenerierter Sprache in Verbindung brachten. Das war durchaus vorteilhaft, denn es vereinfachte die Anforderungen an die Synthese erheblich. Das Ergebnis der zweijährigen Entwicklungsarbeit war der TMS0280,



Bild 2. Speak&Spell von TI (Urheber: Bill Bertram 2006. Keine Änderungen vorgenommen. CC-BY-2.5 nach <https://creativecommons.org/licenses/by-sa/2.5/deed.en>).

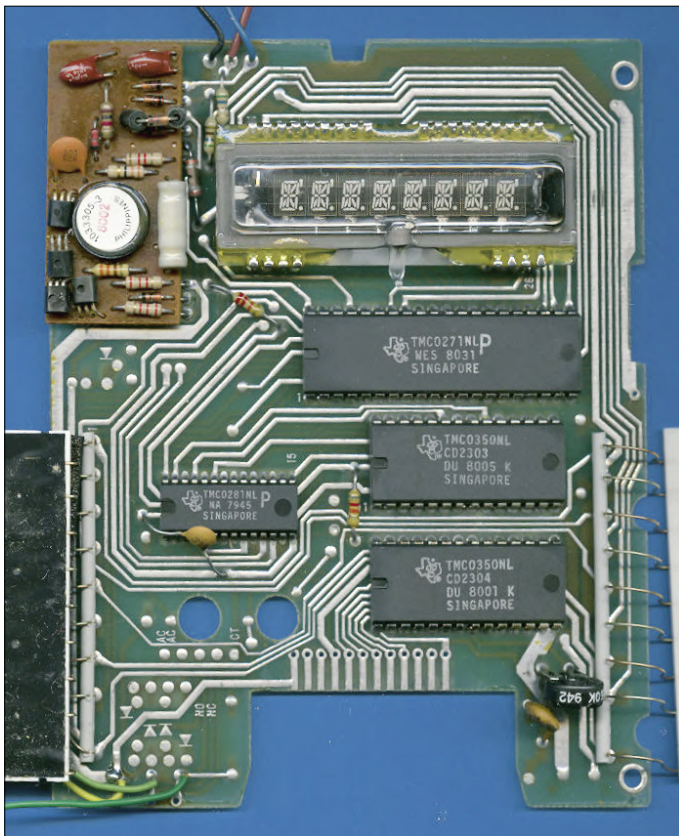


Bild 1. TMC0280 (links, mit der Bezeichnung TMC0281) in einem Speak&Spell (Quelle: Datamath).

der eine lineare prädiktive Kodierung (Vorselektion) verwendete, um aus Audiodaten, die mit 1200 Bit pro Sekunde kodiert wurden, Sprache mit einer Abtastrate von 10 kHz zu erzeugen. Die kodierten Daten wurden als Datenrahmen gespeichert, die Informationen wie Energie, Tonhöhe und Reflexionskoeffizienten enthielten, die für die Sprachalgorithmen benötigt wurden. Manchmal wurden diese Datenrahmen manuell im Labor erzeugt, aber normalerweise wurden die Kodierungsparameter von großen Computerprogrammen generiert, die echte Sprachdaten analysierten. Wie auch immer, beide Methoden hatten das Ziel, kodierte Daten zu erstellen, die Sprache so gut wie möglich abbilden sollten.

Der TMS0280 enthielt einen digitalen Signalprozessor mit einem Multiplizierer, eine Addierer-Implementierung eines Lattice-Filters mit zwei Multiplikationsgittern und ein maskenprogrammiertes ROM als Programmspeicher. Mit dem ROM konnten die Datenrate, die Erregung und die Filtereigenschaften je nach Bedarf der verschiedenen Produktvarianten geändert werden. Der Chip enthielt auch einen Oszillator, eine Schnittstellenlogik und einen 200-mW-D/A-Wandler für die Audioausgabe.

TI stellt *Speak&Spell* seit etwa 30 Jahren nicht mehr her, aber es gibt eine starke Gemeinschaft von Enthusiasten, die die Geräte modifizieren. Das wohl berühmteste war das *Speak&Spell*-Gerät, mit dem E.T. im Film *E.T. Der Außerirdische* nach Hause telefonierte. TI hat sogar spezielle Cartridges (Erweiterungsmodule) auf der Grundlage des Films hergestellt. Vielleicht ist es jetzt an der Zeit, das verstaubte, in einer Kiste vergessene *Speak&Spell*-Gerät ans Tageslicht zu holen und zu sehen, was man mit seinem TMC0280 so alles anstellen kann. ◀

190383-C-03





# Interaktiv

## Korrekturen & Updates || Fragen & Antworten

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

Aktualisierungen und Ergänzungen von Projekten, die in Elektor veröffentlicht wurden, gewürzt mit Tipps und Tricks, technischen Ratschlägen und Antworten auf Fragen der Leser.

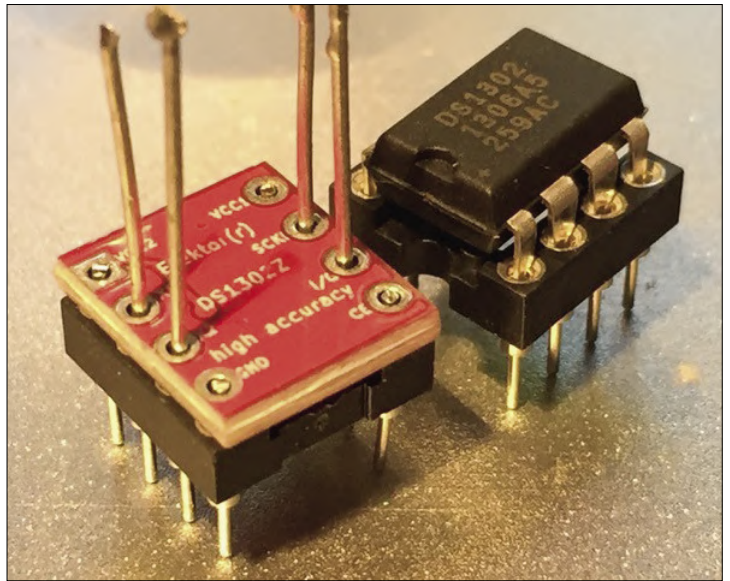
### Upgrade für den Wecker mit 3-fach-Anzeige

Elektor 5-6/2019, S. 56 (180620)

Ich melde mich erneut mit einem Lesertipp aus der Praxis. Im Artikel wird angedeutet, dass das RTC-BoB zu groß ist, um ohne Komplikationen den DS1302 (IC2) der Uhr zu ersetzen. Es werden drei Möglichkeiten beschrieben, die Uhr so zu modifizieren, dass das BoB passt. Ich habe eine vierte Möglichkeit gefunden, die ich persönlich für die beste halte, da sie keinerlei Änderungen an der Uhr erfordert. Der Trick besteht darin, nicht die abgebildeten Stiftleisten zu verwenden, sondern abgeschnittene Bauteildrähte, die auf dem Labortisch jedes Elektronikers reichlich vorhanden sein sollten.

Sobald die drei Bauteile des BoBs bestückt sind, legen Sie das Platinchen mit der Bauteilseite nach unten auf eine flache 2x4-polige IC-Fassung. Dann stecken Sie acht Drähte von oben durch die Löcher der Platine in die Kontakte der Fassung, wie es im Foto gezeigt wird. Verwendet man eine kleine Flachzange, pikt es nicht in den Fingerkuppen und man kann sicher sein, dass alle Pins bis weit in die Kontakte hineinreichen und fest sitzen. Schneiden Sie die Drähte bündig mit der kleinen BoB-Platine ab und löten Sie sie fest. Hebeln Sie dann die BoB-Platine vorsichtig aus der Fassung und setzen Sie sie in die Fassung für IC2 auf der Uhrenplatine ein. Wie Sie auch im Foto erkennen können, ist die RTC-Assemblage sogar noch niedriger als das ursprüngliche IC.

*Hans Schneider (Deutschland)*



### Flimmerfreier LED-Dimmer

Elektor 9-10/2019, S. 12 (190062)

Auch wenn mir die Lektüre des Artikels Spaß gemacht hat, war er meiner Meinung nach nicht vollständig, da er nicht erwähnt, dass den beiden Vorteilen der Methode, nämlich weniger Flimmern und bessere EMV, zwei Nachteile gegenüberstehen.

Erstens ist die Farbe einer LED vom Betriebsstrom abhängig. Das von J. Waagebaert vorgeschlagene Prinzip verändert den Betriebsstrom und damit die Farbe des erzeugten Lichts. Dies kann zum Beispiel in der professionellen Fotografie störend bis nicht akzeptabel sein. Beim Betrieb mit rechteckig gepulstem Strom (PWM) haben die einzelnen Stromimpulse immer den gleichen Wert. Die resultierende Lichtfarbe ändert sich daher in dieser Betriebsart nicht und bleibt über den gesamten Dimmbereich konstant.

Zweitens ist ein sanftes Dimmen im unteren Helligkeits-

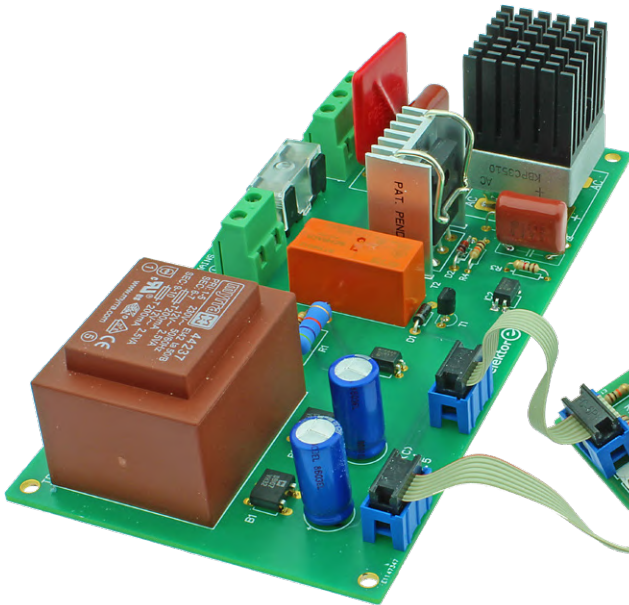
bereich nicht gewährleistet. Nicht alle Power-LEDs haben ein definiertes Verhalten im unteren Strombereich. Es gibt Hersteller, die einen Mindeststrom von beispielsweise 10 % des Nennstroms angeben. Mit einem gepulstem Strom ist es aber immer möglich, sanft bis zur Dunkelheit herunterzudimmen.

In der Praxis wird man für jede Anwendung einen geeigneten Kompromiss finden müssen.

*Daniel E. (Schweiz)*

## Einstellbarer 1-kW-AC-Motortreiber

**Elektor 1-2/2020, S. 62 (191142, 150199).**



Im Schaltplan in **Bild 1** müssen die Symbole für Plus und Minus im Brückengleichrichter B4 vertauscht werden.

Im Schaltplan in **Bild 2** ist nicht klar ersichtlich, dass die SDA- und SCL-Pins des OLED-Displays OLED1 mit Pin 23 und Pin 18 des Mikrocontrollers IC2 verbunden sind.

In der Stückliste ist für den Leistungstrafo TR1 fälschlicherweise Myrra 44327 angegeben, die korrekte Nummer ist 44237.

Schließlich ist die Leistungselektronik-Platine entgegen der Aussage im Artikeltext doppelseitig.

Keiner dieser oben genannten Berichtigungen hat aber irgendeine Auswirkung auf den Betrieb des Projekts oder die zugehörigen Platinen, die im Elektor-Store erhältlich sind.

[www.elektor-labs.com/3980](http://www.elektor-labs.com/3980)

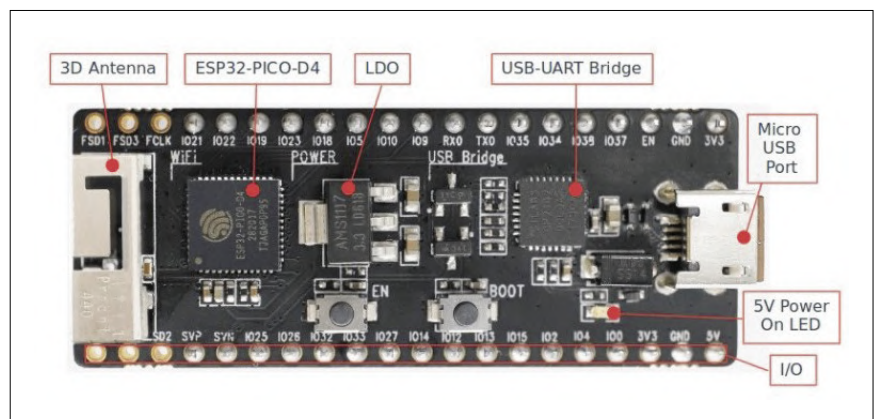
## ESP32-PICO-Kit v4.1 kann nicht mit der Arduino-IDE programmiert werden?

F: Die beiden ESP32-PICO-Kit-Module, die ich letzten Monat gekauft habe, haben die Versionsnummer 4.1, die früher gekauften die Versionsnummer 4. Die älteren Module konnten problemlos mit der Arduino-IDE programmiert werden, doch bei den neuen Modulen konnte ich nur einen einzigen Sketch hochladen, dann funktionierte das Hochladen nicht mehr. Der hochgeladene Sketch lief korrekt. Ich habe verschiedene Baudraten ausprobiert, aber keine Lösung gefunden. Was könnte das Problem sein?

*S. Kepper (Deutschland)*

A: Die beiden Module, die Sie uns geschickt haben, ließen sich problemlos auf unserem Testsystem programmieren. Wir haben jedoch festgestellt, dass es bei bestimmten Kombinationen von Windows 10 und USB-Controllern (AMD Ryzen-Plattform) ein Treiber- oder anderes Softwareproblem zu geben scheint. Leider haben wir keine Kontrolle über diese Softwarekomponenten. Für Abhilfe könnte aber gesorgt werden, wenn man einen alten USB-2.0-Hub dazwischen schaltet.

*Elektor-Labor*



Epilog: Vielen Dank für Ihre Antwort. Ihre Antwort hat mich dazu angeregt, die USB-Anschlüsse meines PCs einmal genauer zu betrachten. Er gibt sowohl eine USB-2.0- als auch eine USB-3.0-Schnittstelle. Ich habe festgestellt, dass ich die neueren ESP32-PICO-Kit-4.1-Module über die USB-3.0-Schnittstelle unter Windows 7 nicht programmieren sollte/kann. Vielleicht ist diese Entdeckung auch für andere Leser interessant. Warum die Programmierung des Moduls aber beim ersten Mal an einem USB-3.0-Port funktionierte, bleibt mir ein Rätsel.

*S. Kepper (Deutschland)*

190379-C-02



# BASIC für ESP32/ESP8266 (2)

## Sanduhr mit ESP8266 und Annex WiFi RDS

Von **Peter Neufeld**

In der letzten Elektor-Ausgabe wurde mit Annex WiFi RDS eine Plattform beschrieben, mit der man ESP8266/ESP32-Controller in BASIC programmieren kann. Jetzt geht es um eine praktische Anwendung: Die Firmware für eine Sanduhr – eben nicht in C, C# oder C++, sondern schnell, unkompliziert und einfach in BASIC.

Bild 1. Bei der fertigen Sanduhr leuchtet links eine vierstellige Digitaluhr und rechts wird eine echte und beleuchtete Sanduhr von einem verborgenen Servo zeitgenau umgedreht.



Zu der in Elektor März/April 2020 [1] beschriebenen BASIC-Entwicklungsumgebung Annex WiFi RDS [2] gibt es hiermit ein praktisches Einstiegsprojekt. Mit den für wenige Euros in diversen Bauformen erhältlichen ESP8266-Modulen eröffnen sich dem Elektroniker nicht nur dank WLAN-Fähigkeit unendliche Möglichkeiten für kleinere und größere Projekte, die sich einfach „per Funk“ an Smartphones, Tablets und PCs anbinden lassen. Da es für kleines Geld auch noch diverse, leicht an den Mikrocontroller anschließbare Sensorik und Aktorik gibt, könnte das Selbstbauen eigentlich wieder richtig Freude machen – wenn da nicht die Sache mit der recht komplizierten Programmierung moderner SoCs wäre. Man hat die Wahl zwischen verschiedenen Programmiersprachen, diversen IDEs, Bibliotheken, Treibern, Ressour-

cen und Compilern. Oder soll man sich daran vorbeimogeln und letztlich schwer verständlichen, fremden Code nehmen und irgendwie auf das Modul transferieren? Richtig kompliziert wird es, wenn man das eigene Projekt auch noch per Web-Interface mit einem Smartphone steuern möchte. Genau so ein Vorhaben wird durch die Entwicklungsumgebung Annex WiFi RDS mit der Programmiersprache BASIC deutlich simpler. Das wiederum führt zu schnellen Erfolgserlebnissen.

### Ein einfaches Projekt

Mit dem ESP8266 eine Digitaluhr zu realisieren, die nebenher auch noch den Wetterbericht anzeigt, ist eine Sache von wenigen BASIC-Zeilen, wie Beispielprojekte auf der Annex-Webseite [3] beweisen. Mir kam noch eine einfache Idee

mit besonderem Schick: Die Kombination einer digitalen mit einer analogen Anzeige. Beim Projekt „ESP8266-Sanduhr“ soll eine neben einer konventionellen vierstelligen Digital-Uhr befindliche kleine Ein-Minuten-Sanduhr zeitgenau umgedreht werden.

Das ist noch nicht alles: Der rieselnde Sand könnte beleuchtet werden. Hierzu dienen vier NeoPixel-LEDs als Hintergrundlicht. Natürlich soll die Helligkeit des Displays der Umgebungshelligkeit angepasst sein, wozu noch ein LDR als Lichtsensor addiert wird. Verbaut man noch einen 1-Wire-Temperatur-Sensor, dann kann man sogar die Raumtemperatur via Display anzeigen.

Selbstverständlich soll die Sanduhr ebenfalls per WLAN via integriertem Webinterface gesteuert werden können. Wenn man bei so einem Vorhaben an den



Code und die vielen potentiellen Bugs denkt, bleibt es manchmal nur bei einem heroischen Ansatz, der auf „mañana“ terminiert wird ;-).

Doch weil die Praxis dank BASIC viel einfacher ist, braucht die in **Bild 1** gezeigte, fertige Sanduhr lange nicht so viel Software-Arbeit, wie man denken könnte.

Mir hat das Schmökern in der umfangreichen, praxisorientierten Hilfedatei zu Annex WiFi RDS samt den Inhalten der zugehörigen Webseite einige Anstöße für neue Projekte gegeben. BASIC ist trotz seiner Mächtigkeit auch für Gelegenheitsprogrammierer gut beherrschbar. Die Softwareentwicklung wird dank der Annex **Rapid-Development-Suite** gegenüber anderen aktuellen Lösungen erheblich vereinfacht.

## Die Elektronik

Das verwendete Controller-Board D1-Mini bietet ein ESP8266-Modul und eignet sich dank seiner Pins für Experimente auf einem Steckbrett, denn alle relevanten Anschlüsse des SoCs sind herausgeführt. Über seine USB-Schnittstelle und die Auto-Flash-Fähigkeit lässt sich das Modul mit dem Annex-Toolkit [4] sehr einfach und schnell sowie ohne Hardwareeingriff mit Firmware bespielen. Hierbei sollte man nicht vergessen, auf der CONFIG-Page die richtige Zeitzone einzustellen und das BASIC-Skript in die dortige Autostartzeile einzutragen (der Pfad meines aktiven Skripts ist immer „/program/default.bas“).

Für meine Tests habe ich die Schaltung von **Bild 2** auf einem Steckbrett aufgebaut. Das Resultat kann man in **Bild 3** bewundern.

Wird die Betriebsspannung über die Micro-USB-Buchse zugeführt, so sind auch +5 V und +3,3 V nach außen geführt und beide Spannungen (begrenzt) zusätzlich belastbar, da eine Schottky-Diode und ein 3,3-V-Spannungsregler zwischengeschaltet sind. Die für die hier verwendete Peripherie geforderten Ströme reizen das nicht aus. Will man absolut sichergehen, kann man speziell den Servo separat mit +5 V versorgen. Die beiden Elkos zwischen +5 V und Masse puffern die Stromspitzen des Servos.

Zu beachten ist, dass die Eingänge eines ESP8266 nur maximal 3,3-V-Pegel vertragen. Daher sind auch fast alle externen Pull-Up-Widerstände an +3V3 gelegt. Der interne Spannungsteiler des Analog-

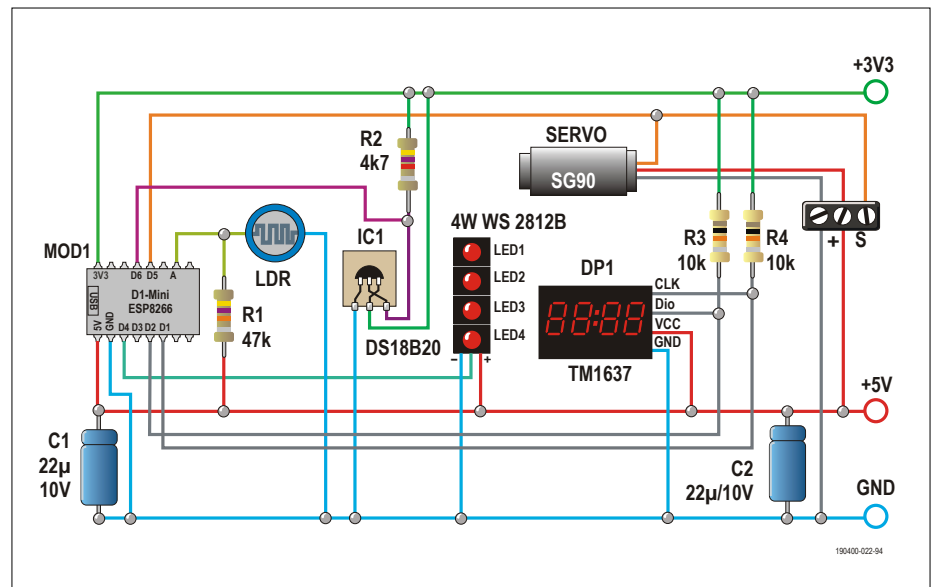


Bild 2. Die Schaltung der elektronischen Sanduhr ist wirklich so einfach: Außer dem ESP8266-Modul sind noch vier Widerstände, zwei Kondensatoren, vier LEDs, ein Temperatursensor, ein Display und ein Servo vonnöten. Eine Stückliste erübrigt sich.

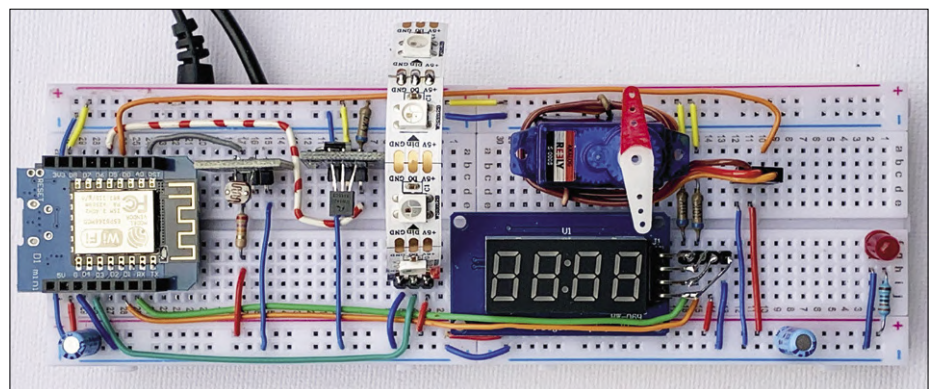


Bild 3. Der Testaufbau der Sanduhr auf einem Steckbrett nach der Schaltung von Bild 2.

gangs sorgt dafür, dass Spannungen von 0...3,3 V ausgewertet werden können und auch angelegte 5 V nicht zu Problemen führen. Der Spannungsteiler beim LDR sollte so dimensioniert sein, dass bei voller Beleuchtung nur maximal 500 mV am Eingang anliegen. Bleibt der Analogeingang unbeschaltet, liegen hier 0 V an und das Display wird folglich auf volle Helligkeit gesetzt.

Vorsicht beim Anschluss des Temperatursensors DS18B20! Das Exemplar mit Modulplatine hat nämlich eine andere Pin-Beschaltung als der Sensor im TO-Gehäuse.

## BASIC-Software

Nach einmaliger Installation der Firmware auf dem ESP8266 mittels des Annex-Toolkits [4] und Abarbeitung der Anleitung

unter [5] bewegt man sich nur noch mit dem Webbrowser auf dem Webserver des Moduls. Im Editor ist schnell neuer Code geschrieben oder per copy & paste in *default.bas* eingefügt.

In der Annex-Hilfedatei (F2-Taste im Editor) findet man Erläuterungen, Pin-Belegungen und Skript-Schnipsel zum TM1637-LED-Display, dem Analog-Servo, den NeoPixel-LEDs und verschiedenen handelsüblichen Aktoren und Sensoren sowie zu den verschiedenen ESP-Modulen. In der Onlinehilfe ist auch ein hilfreicher Abschnitt zu den Grundlagen der Sprache enthalten.

In **Listing 1** wird der Anfang des verwendeten BASIC-Skripts vorgestellt. **Listing 2** enthält den Code des bewusst einfach gehaltenen Web-Interfaces, das in **Bild 4** zu sehen ist.

### Listing 1. Anfang von „sanduhr.bas“.

```
##### HOURGLASS #####
' 4Digit digital clock combined with 10 minutes hourglass
VERSION$ = "V3.3"
' 10/2019 Peter.Neufeld@gmx.de min:ANNEX_1.39b6
' - ESP8266-Module: WEMOS-D1-Mini
' - Display: 4digit 7segment TM1637
' - Servo: analog servo (SG90)
' - Background light: 4 NeoPixel-LEDs
' - Temperature sensor: DS18B20
' - Light sensor: LDR @ analogue input
' GPIO-Mapping (PIN-Label) for WEMOS and NodeMCU boards
D0=16:D1= 5:D2= 4:D3= 0:D4=2
D5=14:D6=12:D7=13:D8=15:D9=3:D10=1
SERVO_PIN = D5 'Servo @ GPIO14=D5
TM_DATA = D2 'TM1637 Data @ GPIO4=D2
TM_CLOCK = D1 'TM1637 Clock @ GPIO5=D1
TM_BRIGHT = 7 'TM1637 brightness
TURN_TIME = 10 'Turn hourglass every xx minutes
SERVO_LEFT = 180 '1. Position of hourglass
SERVO_RIGHT = 0 '2. Position of hourglass
SERVO_POS = 0 'actual position of hourglass
BLINK = 0 '255= visible colon @ TM1637
ADC_DARK = 450 'max. ADC value for dark LDR,
' depends on LDR and Pull-up
LED1_STATUS = 0 'Background LEDs: 0=automatic, 1=manually
STATUS$ = "Modus: AUTOMATIK"
ADC_IN = 0:
```

Schnell kamen bei diesem Projekt immerhin rund 140 BASIC-Zeilen zusammen. Da sie mit ausreichenden Kommentaren versehen sind, versteht man so ein Skript auch dann, wenn man es nicht selbst geschrieben hat oder wenn man in ein paar Wochen eine Erweite-

rung oder Verbesserung vornehmen will. Der komplette Code steht auf der Elektor-Webseite zu diesem Artikel [6] zum Download zur Verfügung.

Im steten Wechsel von Öffnen der Datei *default.bas*, dem Editieren, Speichern (!), Starten, Fehlersuchen, Debuggen, Erwei-

tern, Speichern, Starten etc. werden übliche Fehler im Code vom Interpreter benannt und markiert. Sie sind dann zumeist schnell korrigiert. Der BASIC-Code ist bei Vergabe aussagekräftiger Variablenbezeichnungen und dem Einsatz von Unterprogrammen auch für Gelegenheitsprogrammierer gut zu durchschauen. BASIC ist einfach gut lesbar. Das automatische Code-Highlighting verbessert die Lesbarkeit und erleichtert die Syntax-Kontrolle. Bei Syntax-Problemen hilft die F2-Taste mit ihrer kontextsensitiven Suche in der Online-Hilfdatei. Außerdem gibt es noch den kleinen grünen Haken im Editorfenster, der zum Syntaxcheck führt.

Das BASIC-Skript ist zunächst recht linear. Nach Initialisierung einiger Variablen und einer kurzen Bewegung des Sanduhr-Servos wird die IP-Adresse des ANNEX-Webserver „portionsweise“ auf dem Display ausgegeben. Das hilft sehr beim Zugriff mit dem Webbrowser, gerade wenn der Router doch eine andere IP-Adresse zugeteilt haben sollte.

Die Hauptroutine wird mittels eines Timers einmal pro Sekunde abgearbeitet. Dabei werden Stunden und Minuten anzeigegerecht aufbereitet, und es werden sequenziell die Unterprogramme zur Lichtsteuerung und der Hintergrundbeleuchtung, der Darstellung von Uhrzeit und Temperatur sowie alle zehn Minuten auch die Servo-Steuerung aufgerufen.

Vom ursprünglichen Plan die Sanduhr einmal pro Minute zu drehen bin ich abgerückt, als ich merkte, wie laut ein Modellbau-Servo ist, wenn er die Sanduhr im Wohnzimmer dreht. Daher entschied ich mich für eine Zehn-Minuten-Sanduhr, die langsam über rund vier Sekunden und damit leiser und weniger häufig das Glas dreht. Wer das mag, kann mit einer schnelleren Sanduhr auch einen Ein- oder Fünf-Minuten-Rythmus über die Variable `TURN_TIME` realisieren.

### Webinterface

Wenn ein Webbrowser erstmalig oder erneut zugreift oder sich verwendete Variablenwerte ändern, wird der HTML-Inhalt der ESP8266-Webseite im Unterprogramm `WEB_PAGE` (siehe Listing 2) aufbereitet und aktualisiert. Das Webinterface zeigt die laufende Uhrzeit und die Temperatur. Die einfache LED-Hintergrundanimation hinter der Sanduhr wird per Taster durch eine

### Listing 2. Unterprogramm „WEB\_PAGE“.

```
WEB_PAGE:
cls
a$ = "<center><h1> - S A N D U H R - " + VERSION$ + " - </h1>"
a$ = a$ + "<br>" + textbox$(t$, "cssTB")
a$ = a$ + METER$(SS, 0, 60, "cssMET")
a$ = a$ + textbox$(TEMP$, "cssTB") + "<br><br>"
a$ = a$ + "<br>R: " + slider$(R, 0, 255) + textbox$(R, "cssTB")
a$ = a$ + "<br>G: " + slider$(G, 0, 255) + textbox$(G, "cssTB")
a$ = a$ + "<br>B: " + slider$(B, 0, 255) + textbox$(B, "cssTB")
a$ = a$ + "<br>" + LED$(LED1_STATUS)
a$ = a$ + "<br>" + textbox$(STATUS$)
a$ = a$ + "<br>" + BUTTON$("Umschaltung auto/manuell", MAN_AUTO)
a$ = a$ + cssid$("cssTB", " width:70px;text-align:center")
a$ = a$ + cssid$("cssMET", " transform:rotate(-90deg);")
html a$
return
```

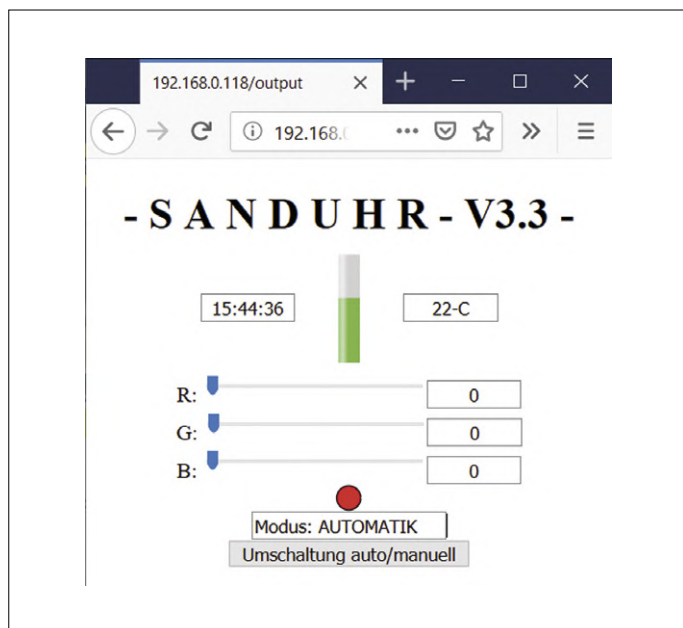


Bild 4. Mit wenig BASIC-Code kann man ein einfaches Webinterface realisieren.

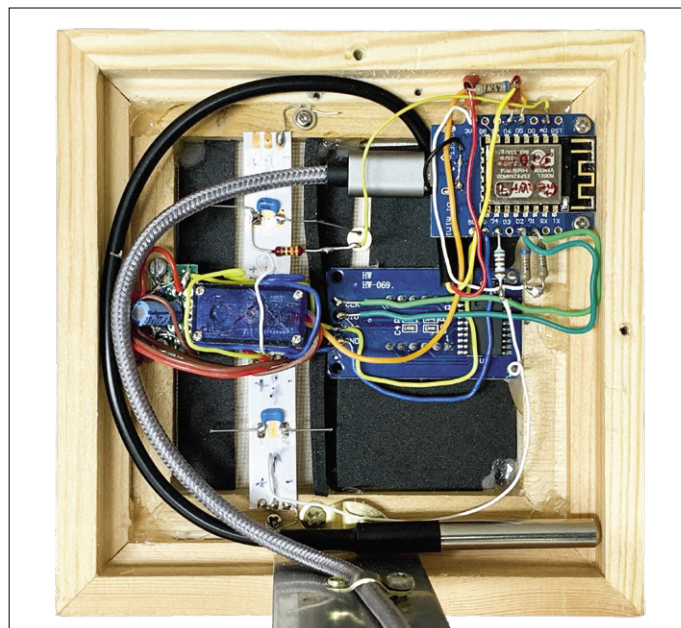


Bild 5. So sieht die elektronifizierte Sanduhr von hinten aus: Nicht schön, aber funktionierend fliegend verdrahtet.

manuelle RGB-Farbauswahl ersetzt. Die RGB-Regler und Anzeigefelder folgen im Automatikmodus live den laufend berechneten Werten der Animation.

Die Website soll hier lediglich beispielhaft demonstrieren, wie sich die Annex-Firmware im Hintergrund um die bidirektionale Kommunikation mit dem Webbrowser kümmert. Durch die Annex-unterstützte Nutzung von CSS (auch Javascript, AJAX und Websockets werden unterstützt) sind Experimenten zur Kommunikation mit (fast) beliebigen Webbrowsern kaum Grenzen gesetzt. Auch dazu gibt es viele Beispiele auf der Annex-Webseite – insbesondere auch in den Dateien, die aus dem Annex-Toolkit bereits bei der Firmware-Installation in das Verzeichnis „/program“ auf dem ESP-Modul kopiert werden können. Stöbern lohnt sich! Ein interessantes Skript ist mit einem Text-Editor (z.B. *GEANY.exe*) zum Testen per „ctrl+a“ (alles Markieren) sowie copy & paste schnell in *default.bas* im Annex-Editorfenster eingefügt, abgespeichert und gestartet! Komplette Bereiche des Skripts lassen sich auch aus dem Menü heraus schnell in Kommentarzeilen verwandeln und umgekehrt auch sehr schnell wieder aktivieren.

### Umzug ins finale Gehäuse

Die ausgetestete Schaltung wurde letztlich ganz ohne Platine oder Steckbrett einfach fliegend in einen kleinen Holzrah-

men eingebaut, der frontseitig mit einer lichtdurchlässigen Leinwand bespannt ist (**Bild 5**). Dadurch sind nur die aktiven, leuchtenden LED-Display-Segmente sichtbar und auch der LDR-Lichtsensord kann ausreichend Umgebungslicht aufnehmen. Die Servo-Achse für die Sanduhr wurde durch ein kleines Loch in der Leinwand geführt. Mit entsprechend zugeschnittenen Abschnitten aus Moosgummiplatten oder schwarzem Papier wird die Lichtdurchlässigkeit des Frontmaterials auf die notwendigen Zonen am Display, hinter der Sanduhr

und am LDR eingegrenzt.

Alternativ könnten auch andere Rahmen mit Plexiglasfronten (z.B. aus dem Sortiment eines großen Mitnahmemeubelhauses) verwendet werden. Die durchsichtige Front kann dann innen mit einem durchscheinenden Papier oder auch einem passenden Fotoausdruck hinterlegt werden. ◀

190400-B-01



### IM ELEKTOR-STORE

→ WeMos D1 mini Pro

[www.elektor.de/wemos-d1-mini-pro-esp8266-based-wifi-module](http://www.elektor.de/wemos-d1-mini-pro-esp8266-based-wifi-module)

→ Buch „IoT Home Hacks with ESP8266“

[www.elektor.de/iot-home-hacks-with-esp8266](http://www.elektor.de/iot-home-hacks-with-esp8266)

→ PDF „ESP32 & ESP8266 Kompilation“

[www.elektor.de/esp32-esp8266-kompilation-de](http://www.elektor.de/esp32-esp8266-kompilation-de)

### Weblinks

- [1] BASIC für ESP32/ESP8266: [www.elektormagazine.de/190400-01](http://www.elektormagazine.de/190400-01)
- [2] Annex WiFi RDS: <https://sites.google.com/site/annexwifi/home>
- [3] Beispielprojekte: <https://sites.google.com/site/annexwifi/projects>
- [4] Toolkit: <https://sites.google.com/site/annexwifi/downloads>
- [5] Anleitung: <https://sites.google.com/site/annexwifi/home/first-steps>
- [6] Webseite zum Artikel: [www.elektormagazine.de/190400-B-01](http://www.elektormagazine.de/190400-B-01)



# Automatische Identifizierung mithilfe der RFID-Lesegeräte von Elatec

Die auf der RFID-Technologie basierenden Geräte finden in der Lagerlogistik, Spedition und Verwaltung eine breite Anwendung. Eine große Auswahl an verfügbaren Lösungen und ein häufig auftretender Wunsch, alle Anwendungen in ein einheitliches, kohärentes System zu integrieren, können zusammen sehr problematisch und zeitintensiv werden. Die Lösungen von der Firma Elatec ermöglichen es, die Kosten zu senken und den Zeitaufwand zu reduzieren, der für die Integration der neuen Lösungen in die bereits existierenden Systeme erforderlich ist.

Die Märkte der RFID-Systeme und der kabellosen Datenübertragung entwickeln sich dynamisch und stellen sowohl die Hersteller der Geräte, als auch die Integratoren der Anwendungen vor immer neue Herausforderungen. Sie betreffen nicht nur die elektromagnetische Kompatibilität, sondern auch die Auswahl des Identifikationsstandards und den Typ der verwendeten Transponder-Systeme sowie der bedienenden Lese- und Programmiergeräte.

Unabhängig vom Anwendungsbereich ist die Auswahl der entsprechenden RFID-Technologie, die für die Identifizierung anzuwenden ist, der Schlüsselfaktor. Die Auswahl hängt von vielen weiteren Faktoren ab, wie z.B. die erwartete Reichweite, die Art der zu identifizierenden Objekte sowie die Frage, ob eine bilaterale

Kommunikation mit dem Transponder erwünscht ist. Einige von diesen Systemen sind nicht nur in der Lage, ein einmaliges Kennzeichen im Speicher aufzubewahren, sondern auch die Daten permanent zu speichern. Dabei kann die Kommunikation mit dem übergeordneten System verschlüsselt werden.

Die Auswahl der zur Identifizierung verwendeten Technologie ist umso wichtiger, dass viele verschiedene RFID-Standards gleichzeitig existieren. Sie unterscheiden sich voneinander mit den verwendeten Frequenzbereichen (daraus folgen unterschiedliche Größen von Antennen und Transpondersystemen), Reichweiten, Arten der Modulation, Datenübertragungsgeschwindigkeiten etc. Das kann dazu führen, dass die neu eingeführten und die bereits verwendeten Lösungen mit unterschiedlichen Kommunikationsstandards arbeiten werden. Ein Beispiel dafür ist ein Bürogebäude, in dem die RFID-Technik zur Arbeitszeiterfassung und Zutrittskontrolle in die Räume angewendet wird. In einem solchen Gebäude könnte es irgendwann notwendig sein, ein weiteres System dazu zu integrieren, das den ausgewählten Mitarbeitern Zugriff auf die Drucker oder Kopierer gewähren kann. Die Drucker und Kopierer werden auch mit RFID-Lesegeräten ausgestattet. Diese Lesegeräte arbeiten aber üblicherweise mit anderen Datenübertragungsstandards als diejenigen von der Zutrittskontrolle. So müssten die Mitarbeiter jeweils zwei oder mehr Karten haben: eine für den Zutritt in den Raum und eine oder mehrere für die Identifizierung vom Drucker oder Kopierer.

## Multisystem-Lese- und Programmiergeräte

Dieses Problem hat die Firma Elatec – ein bekannter Hersteller von RFID-Systemlösungen – bereits erkannt. Sie bietet universelle, fertige Lese- und Programmiergeräte, die mit einer seriellen Schnittstelle an das übergeordnete System angeschlossen werden. Den OEMs werden gehäuselose RFID-Module angeboten. Die fertigen Lese- und Programmiergeräte arbeiten meistens mit RS232 oder USB, die Module für OEMs bedienen auch viele andere Standards. Dabei hat man einen großen Spielraum und man kann das Lese- oder Programmiergerät an die verfügbaren Ressourcen einfach anpassen.



Bild 1. Externes Lesegerät TWN4 Multitech Legic 42.

Es ist immer gut, ein konkretes Beispiel zu nennen. Das im Angebot von TME verfügbare Lesegerät **TWN4 Multitech LEGIC 42** (siehe **Bild 1**) arbeitet in vielen sehr unterschiedlichen Frequenzbereichen: 125 kHz, 134,2 kHz und 13,56 MHz. Das Lesegerät hat eine RS232- und eine USB-Schnittstelle. Der Hersteller stellt Treiber für Windows und Linux zur Verfügung. Dieser Lesegerätstyp wird auch als OEM-Modul angeboten. In dieser Version kann er, außer den bereits Erwähnten, noch mithilfe von seriellen Schnittstellen (Logikpegel 3,3 V, toleriert 5 V CMOS, I<sup>2</sup>C) an das übergeordnete System angeschlossen werden. In speziellen Ausführungen kann er ebenfalls mit den Schnittstellen wie SPI, Wiegand, CAN und 1-Wire kommunizieren. Was die Funkschnittstelle angeht, ist eine spezielle Unterstützung für Transponder von vielen Herstellern verfügbar, u.a. Atmel (aktuell Microchip), EM, ST, NXP, Texas, TI, HID, LEGIC und andere, die mit folgenden Standards arbeiten: ISO14443A/B, ISO15693, ISO18092/ECMA-340 (NFC), HITAG, UNIQUE, ISO14443 A+B (Mifare DESFire EV1, Mifare Plus, Mifare SmartMX, my-d move, PayPass itp.), ISO15693 (EM4035, Tag-It, my-d vicinity, ICODE SLI), Mifare Classic, Mifare Ultralight, Sony FeliCa, NFC Forum Tag Type 2-4, PicoPass, HID iCLASS. Auf die Platte des Lesegeräts wurde eine Funkantenne integriert, die im Frequenzbereich 125 kHz / 134 kHz / 13,56 MHz arbeitet, den die o.g. Standards unterstützen. Außer der Schnittstelle zum übergeordneten System besitzt das Lesegerät noch acht GPIO-Ausgänge, über die Schösser, Signal-Dioden, Signaltöne o.ä. gesteuert werden können. Ihre Funktionsweise kann mithilfe einer Skript-Sprache programmiert werden.

### Identifizierung mit dem Smartphone

Es ist bemerkenswert, dass das Lese-/Programmiergerät, außer den meisten RFID-Technologien, auch NFC und Bluetooth Low Energy (BLE) unterstützt. Diese Standards werden in den Alltagsgeräten wie Smartphones oder Tablets verwendet. Man kann also z.B. ein Smartphone mit einer vorinstallierten App anstatt einer RFID-Karte verwenden. Das Lesegerät bedient die meist gängigen Modelle, wie iPhone und Smartphones mit Betriebssystemen Android und Windows. Zurück zum Beispiel mit dem Bürogebäude – die Anwendung der Lesegeräte von Elatec lässt die Probleme mit der Identifizierung vermeiden. Alternativ, anstatt die Mitarbeiter mit einer oder mehreren Karten auszurüsten, reicht es aus, eine entsprechende App auf ihre Smartphones zu installieren.

Für die Anwender der OEM-Module wird es bestimmt wichtig sein, dass die Firma Elatec hinter ihrer Herstellung steht. Dies stellt sicher, dass alle Normen für elektromagnetische Kompatibilität und Produktion (z.B. RoHS 2) eingehalten werden und somit das höchste Niveau an Zuverlässigkeit zugesagt werden kann. Mehr Informationen finden Sie beim Vertreiber – der Firma TME.

### Skriptsprache

Der Hersteller stellt ein Programmierpaket zur Verfügung, mit dem man Apps für die Lesegeräte erstellen kann. Die Funktionsweise der Lesegeräte kann durch die Skripts automatisiert werden, die direkt vom Prozessor des Lesegeräts bearbeitet werden. Außerdem kann man das Skript jederzeit modifizieren – z.B. nach dem Empfang des Befehls, den Arbeitsmodus zu ändern, kann das Lesegerät mit einem anderen RFID-Standard arbeiten. In dem oben beschriebenen Modul TWN4 ist dies mit



Bild 2. Das Modul des Lese-/Programmiergerätes für OEM, Typ TWN4 Multitech 2.

einer kabellosen Schnittstelle oder einer Konfigurationskarte möglich – die Karte wird von Elatec angeboten. Dies erleichtert die Arbeit der Service-Techniker, die beim Änderungsbedarf keine integrierten Geräte demontieren müssen, sondern alle Änderungen eingriffslos durchführen können.

### Erkennung der RFID-Technologie

In den Szenarien der Migration von einem System in ein anderes, ähnlich wie in unserem Beispiel eines Bürogebäudes, ist es erforderlich, die aktuell verwendete RFID-Technologie zu erkennen, um eine neue, optimale Lösung anbieten zu können. Im Normalfall müsste man dann an den Lieferanten oder Integrator eine RFID-Karte senden, die aktuell verwendet wird. Dieses langwierige Vorgehen ist jetzt nicht mehr notwendig – dank den RFID-Analysetools von Elatec, wie z.B. TechTracer Lite. Es kann sofort die RFID-Technologie erkennen, die in einer bereits funktionierenden Lösung angewendet wird. Dies reduziert nicht nur den Zeitaufwand, sondern auch die Kosten und das Fehlerrisiko auf beiden Seiten – sowohl beim Lieferanten, als auch beim Kunden. ◀

200140-01

### Weblink

Mehr Informationen über die RFID-Lese- und Programmiergeräte von Elatec unter: <http://tiny.cc/ylmemz>



# Von Entwicklern für Entwickler

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

Von Clemens Valens (Elektor-Labor)

# VON DER IDEE ZUM PRODUKT

## TEIL 5

Auch wenn es so aussieht, als ob unser neuer Produktentwurf eher im Schneckentempo vorankommt, dürfte es in der realen Welt wahrscheinlich ähnlich sein. Die Gründung eines Unternehmens braucht eben Zeit. In der letzten Folge dieser Rubrik haben wir eine vollständige und korrekte Stückliste (Bill of Materials, BoM) und eine Netzliste entwickelt, so dass wir nun mit der Arbeit an einer Platine (oder mehrerer, je nach Bedarf) für das Projekt beginnen können. Bitte beachten Sie, dass ein Platinenentwurf sehr komplex ist und in diesem kurzen Artikel nicht ausführlich behandelt werden kann. Weitere Informationen und Vorschläge für die Arbeit am Platinenlayout finden Sie unter [1].

### BEGINNE AM ENDE!

Eines der ersten Dinge, die wir in dieser Artikelserie getan haben, war die Erstellung einer „Attrappe“ unseres zukünftigen Produkts. Mit diesem Dummy im Hinterkopf entwarfen wir die elektronische Schaltung, die darin eingebaut werden sollte, und wählten folglich nur Bauteile und -gruppen aus, die in das Innere des Produkts passen und an die vorgesehene Arbeitsumgebung angepasst sind. Nun werden wir eine oder mehrere Leiterplatten in diese Attrappe einfügen, denn diese müssen natürlich ebenfalls passen. Bevor Sie sich also mit der Platzierung von Bauteilen und Leiterbahnen beschäftigen, sollten Sie sich Gedanken über den Umriss der Platine, die Montagelöcher

oder andere Möglichkeiten machen, mit deren Hilfe die Platine im Gehäuse an ihrem Platz fixiert werden kann.

Oft beginnt man einfach mit einem rechteckigen Umriss, platziert die Bauteile so nah aneinander, wie es das Routing erlaubt, und anschließend wird der Umriss soweit es geht verkleinert, um die kleinste Platine zu erhalten. Dann wird die Platine geätzt, bestückt, getestet und schließlich in das Gehäuse eingesetzt. Spätestens dann stellt sich aber heraus, dass entweder keine oder nur unzureichende Montagelöcher vorhanden sind, sie sich nicht an der richtigen Stelle befinden oder nicht den richtigen Durchmesser aufweisen. Außerdem befinden sich die Steckverbinder auf der falschen Seite und eine fachgerechte Wärmeabfuhr ist nicht möglich.

### GENIEßEN SIE IHRE FREIHEIT

Das Interessante am Platinenentwurf ist die Kombination von verschiedenen Disziplinen wie Elektronik und Maschinenbau. Es geht darum, ein mechanisches Teil mit bestimmten elektronischen Eigenschaften zu entwerfen, und Sie als Entwickler haben (fast) die volle Kontrolle über dieses Teil. Nutzen Sie also die Möglichkeiten, die Ihnen das PCB-Design bietet. Eine Platine kann Teil des Gehäuses sein, sie kann aus Aluminium gefertigt sein, sie kann flexibel sein, sie kann fast jede Form annehmen. Dass die Platinen, die Sie in Elektor sehen, so oft rechteckig sind, bedeutet nicht, dass Ihre auch so sein müssen.



Das Gehäuse des legendären Elektrosmog-Detektors TAPIR von Elektor wurde vollständig aus Platinen-Basismaterial hergestellt.

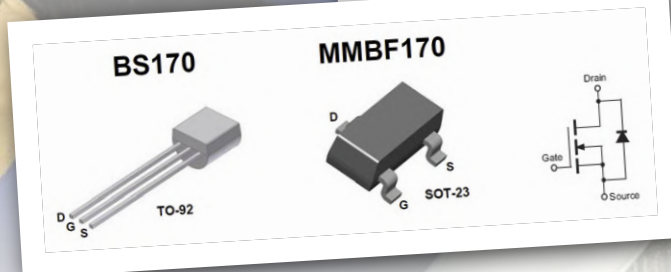
### ENTWURF FÜR DIE PRODUKTION

Es ist verlockend, bei der Gestaltung eines Footprints Sicherheitsmargen einzubauen, zum Beispiel ein Pad etwas größer als angegeben, ein Loch etwas breiter, so etwas. Das klingt nach einer guten Idee, aber es kann der automatisierten Bestückung sehr wohl Probleme bereiten. Übergroße SMD-Pads können dazu führen, dass Teile beim Reflow-Löten aus ihrer Position fließen oder sich drehen. Oder die überschüssige Lötpaste kann Kurzschlüsse verursachen. Löcher mit einem Durchmesser, der viel größer ist als der des durchlaufenden Leiters, können möglicherweise nicht richtig wellengelötet werden.



## KÖNNEN SIE IHREN BAUTEILEN VERTRAUEN?

Abgesehen von der Vernachlässigung der mechanischen Kompatibilität sind falsche Footprints ein weiteres sehr häufiges Problem beim Platinenentwurf. Wenn Sie sich an passive Bauteile in 0805-Gehäusen halten und nur SOT-23-Transistoren und -Dioden verwenden, werden Sie wahrscheinlich auf keine Schwierigkeiten stoßen, aber die Wahrscheinlichkeit ist groß, dass Sie für zumindest ein Bauteil einen Footprint selber definieren müssen, weil die Bauteilform nicht in den Bibliotheken Ihrer CAD-Software enthalten ist. Wieder stellt sich die Frage aus der vorherigen Ausgabe: „Eine Suche im Internet mag eine schnelle Lösung sein, aber können Sie „demo12345“ wirklich vertrauen? Sie könnten Bauteildateien von einem kommerziellen Schaltplan/Platinen-Bauteilhändler kaufen, was vielleicht besser ist, aber Geld kostet. 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-Bibliotheksbau-



Es ist überraschend, wie verwirrend Datenblätter sein können. Dabei geht es auch klar und deutlich.

teile, da Bibliotheken Fehler enthalten können.“ Neben dem Herunterladen des Datenblatts für jedes Bauteil Ihres Entwurfs ist es auch sehr empfehlenswert, dass vor Beginn des Platinenentwurfs Muster von jedem einzelnen Bauteil auf Ihrem Schreibtisch vorhanden sind. Es ist überraschend, wie unklar Datenblätter bezüglich Durchmesser (Pin oder Loch?), Sicht von oben oder unten, Zoll oder Millimeter, Pin-Nummerierung und so weiter sein können. Und sie können auch Fehler enthalten.

## SPRECHEN SIE MIT DER BESTÜCKUNGSFIRMA

Neben Lötproblemen aufgrund falscher Footprints kann auch die Platzierung von Bauteilen den Bestückungsprozess behindern. Ein großes Metallteil neben einem winzigen Kunststoffteil kann dazu führen, dass das erste schlecht gelötet und/oder das zweite aufgrund der ungleichmäßigen Wärmeanforderungen beschädigt wird. Die Bauteilpositionen sind möglicherweise nicht mit dem Bestückungsautomaten kompatibel, was zum Beispiel Snap-

off-Tabs (Stege zwischen Platinenabschnitten) auf der Platine erfordert. Es gibt auch Bauteile mit speziellen Lötanforderungen. Die meisten dieser Probleme können zwar auch von den Mitarbeitern der Bestückungsfirma gelöst werden, aber wenn Sie diese Mitarbeiter vor der Fertigstellung des Platinenlayouts konsultieren, dürfte die Platine billiger und schneller produziert werden und im Endeffekt auch zuverlässiger sein.

## LEITERPLATTEN-ITERATIONEN SIND TEUER

Mit hoher Wahrscheinlichkeit wird die erste Version Ihres Platinenlayouts ein paar Optimierungen benötigen. Damit Sie die verschiedenen Iterationen der Leiterplatte verfolgen können, markieren und beschriften Sie die Platinen deutlich. Vergessen Sie nicht, auf jeder neuen Platinenversion die Versionsnummer zu aktualisieren. Es ist auch empfehlenswert, die Änderungen und die Gründe dafür zu notieren. Je größer das Vertrauen in das Layout, desto mehr ist man geneigt, große Stückzahlen zu ordern. Das macht das Produkt zwar billiger, aber es macht Änderungen auch gefährlicher. Das Wegwerfen von zehn Prototypen ist wahrscheinlich kostengünstiger als der Austausch eines Widerstandes mit einem falschen Wert auf 1.000 Platinen.

### Weblink

[1] Die Kunst des Routens:  
[www.elektormagazine.de/160397](http://www.elektormagazine.de/160397)

## VORSICHT VOR GESPIEGELTEN FOOTPRINTS!

Der Fehler mit gespiegelten Footprints tritt so häufig auf, dass er einen eigenen Absatz verdient. Es ist nicht nötig, ihn zu erklären, Sie wissen, was ich meine. Hören Sie einfach auf, diesen Fehler zu machen. Punkt, Ende des Absatzes.

## WIE GEHT ES WEITER?

Unser Produkt ist fast fertig. Tatsächlich? Wir haben jetzt ein paar Prototypen zum Testen und zum Spielen. Testen, fragen Sie? Ja, genau. Tests für die Funktionalität, für die Zertifizierung, für die Qualität - es gibt eine Menge Tests, die erledigt werden müssen. Wie machen Sie das? Was sind die Auswirkungen? Wird denn diese Artikelserie niemals enden? ◀

190369-C-03

# Ein Grillthermometer und das Dolbearsche Gesetz



Auch auf die Gefahr hin, mich zu wiederholen: Die Temperatur ist vielleicht die weltweit am meisten untersuchte physikalische Größe. Es gibt mehrere Methoden zur Messung der Temperatur. In der Elektronik sind die gebräuchlichsten Methoden dazu:

- **Thermistor:** ein Bauelement, dessen Widerstand mit der Temperatur zunimmt (positiver Temperaturkoeffizient oder PTC) oder abnimmt (NTC)
- **Widerstands-Temperatur-Detektor (RTD):** ein RTD nutzt die genau bekannte Beziehung zwischen Temperatur und Widerstand eines Materials, typischerweise eines Metalls wie Platin
- **Thermoelement:** eine Verbindung von zwei verschiedenen Metallen, die durch den thermoelektrischen (Seebeck-) Effekt eine temperaturabhängige Spannung erzeugt
- **Sensoren auf Halbleiterbasis:** die Spannung über einer Diode oder einem PN-Übergang nimmt um etwa 1...2 mV pro Grad Celsius ab
- **Infrarot-Thermometer:** leitet die Temperatur aus der Wärmestrahlung ab.

Die letzte Methode wird oft zur Messung der menschlichen Körpertemperatur verwendet, indem man mit einer „Temperaturpistole“ auf die Innenseite eines Ohres zielt. Alles bekannt, nichts Neues, sagen Sie nun und unterdrücken diskret ein Gähnen. Aber wussten Sie, dass man die Temperatur mit dem Ohr selbst *hören* kann? Wie sich herausstellte, ist das von einer Grillenart (engl. cricket) erzeugte Zirpgeräusch temperaturabhängig. Ähnlich wie man die Entfernung eines Gewitters abschätzen kann, indem man die Sekunden zwischen einem Blitz und dem darauf folgenden Donner zählt, kann man die Temperatur der Luft abschätzen, indem man die Anzahl des von einer Grille in einer Minute erzeugten Zirpens zählt. Diese Korrelation wurde im Jahr 1881 von Margarette W. Brooks beobachtet, aber der Verdienst der Entdeckung ging an Amos Dolbear, nachdem er im Jahr 1897 einen Artikel mit dem Titel „*The Cricket as a Thermometer*“ veröffentlicht hatte. Die grüne Baumgrille, die Biologen als *Oecanthus fultoni* kennen, wird seitdem als Thermometergrille bezeichnet.

Die Berechnung der Temperatur  $T$  aus der Zirprate der Grille ist einfach, besonders in Fahrenheit:

$$T = N_{15} + 40 \text{ [}^{\circ}\text{F]}$$

Hier ist  $N_{15}$  die Anzahl der Zirp-Laute in 15 Sekunden. Umgerechnet (und gerundet) in Celsius ist das:

$$T = N_8 + 5 \text{ [}^{\circ}\text{C]}$$

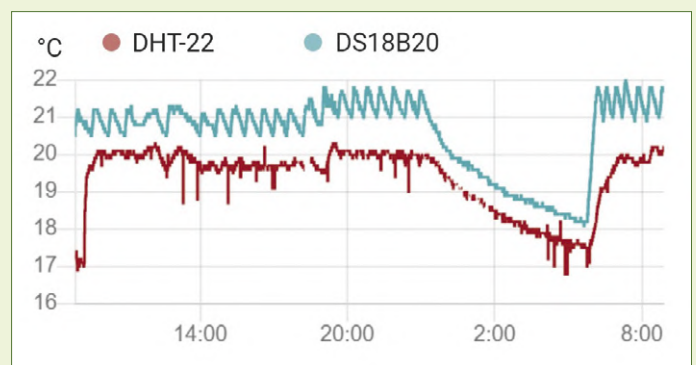
Hier ist  $N_8$  die Anzahl der Zirp-Laute in 8 Sekunden. Das Grillenthermometer ist im Bereich 5...30°C ziemlich genau.

## Grillenemulator

Nur so zum Spaß habe ich mit einem Arduino Uno, einem passiven Summer und einem One-Wire-Temperatursensor DS18B20 den Prozess umgekehrt, um auf Grundlage der Umgebungstemperatur eine Grille zu emulieren. Das Projekt erscheint, um ehrlich zu sein, selten dämlich, doch als ich es in Gang gesetzt hatte, bemerkte ich schnell, wie empfindlich das menschliche Ohr auf Veränderungen reagiert. Letztendlich denke ich, dass dieser Emulator ein nützlicher Temperaturindikator für Anwendungen sein könnte, bei denen es nicht praktisch ist, ein Display im Auge zu behalten.

Die vollständigen Projektdetails finden Sie unter:

[www.elektor-labs.com/3945](http://www.elektor-labs.com/3945)



Nicht alle Temperatursensoren sind gleich. Dieses 24-Stunden-Log zeigt die Ausgangsdaten eines DHT-22 und eines DS18B20-Sensors, die nebeneinander angeordnet sind.



# BalBot

## Ein selbstbalancierender Roboter

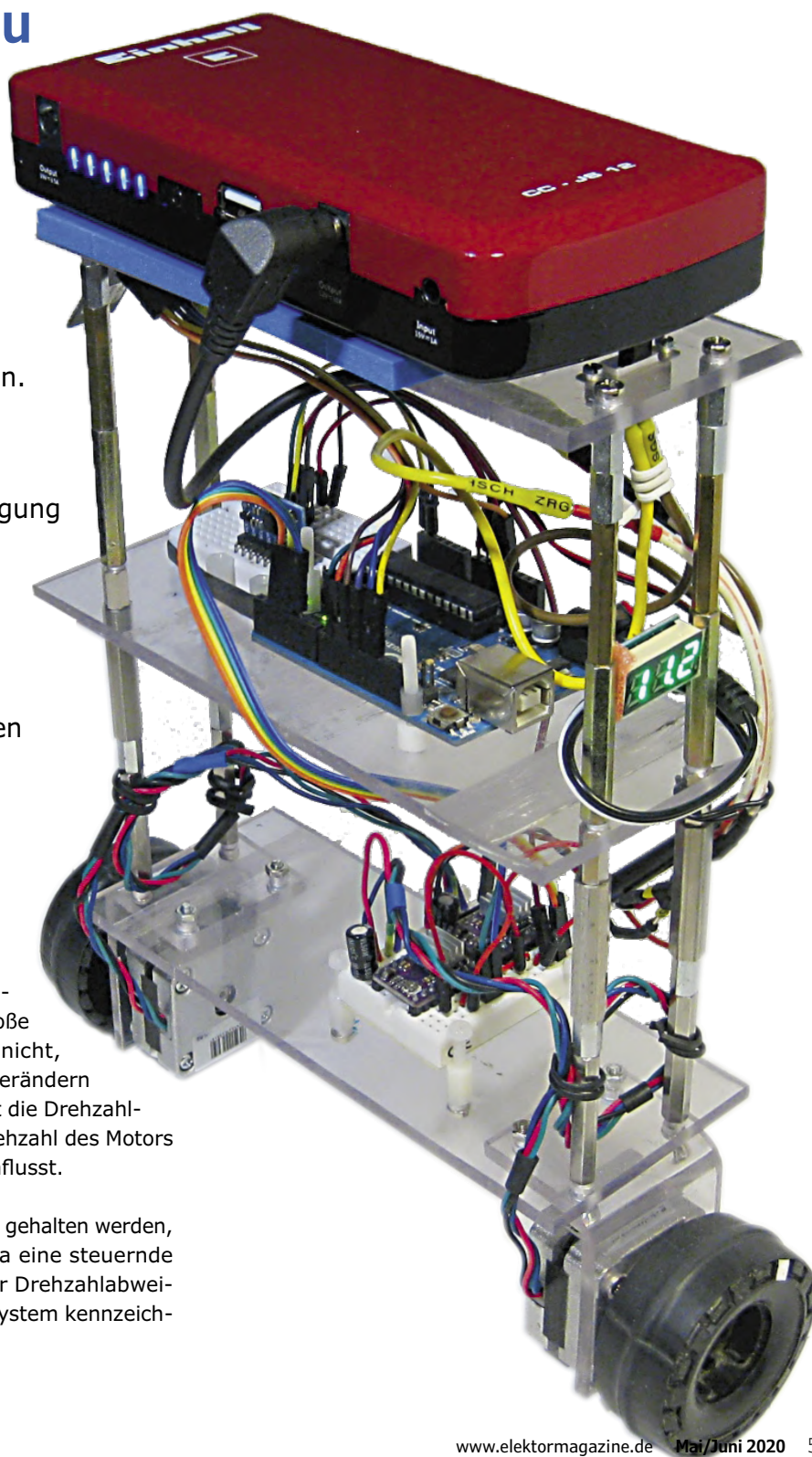
### Moderne Beschleunigungssensoren erleichtern den Bau

Von Dr. Günter Spanner

Regelungen sind nicht nur in der Technik weit verbreitet, sondern auch in der Natur häufig anzutreffen. Der Bau von selbstbalancierenden Robotern kann als Vorstufe von Androiden dienen, da die Fortbewegung auf zwei parallelen Rädern eine ähnliche Regelung erfordert wie der aufrechte Gang. Bei diesem Projekt werden ein Arduino Uno und ein IC eingesetzt, das einen Beschleunigungssensor und ein Gyroskop enthält.

Im Alltag wird im Gegensatz zu Technik und Wissenschaft häufig nicht zwischen Regelungen und Steuerungen unterschieden. Eine bloße Steuerung überwacht die Ausgangsgröße nicht, weshalb sie sich durch externe Störgrößen verändern kann. Ein einfaches und typisches Beispiel ist die Drehzahlsteuerung eines DC-Motors per PWM. Die Drehzahl des Motors wird hier auch von Lastschwankungen beeinflusst.

Nun zur Regelung: Soll eine Drehzahl konstant gehalten werden, ist eine Rückkopplung erforderlich, um etwa eine steuernde PWM bzw. Gleichspannung abhängig von der Drehzahlabweichung anzupassen. Ein so rückgekoppeltes System kennzeichnet einen Regelkreis.





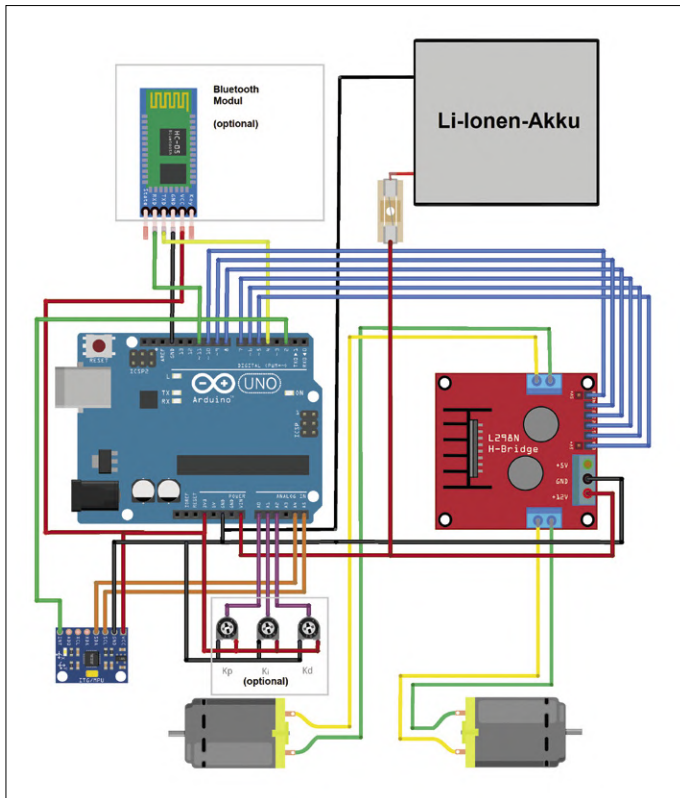


Bild 1. Verdrahtungsplan des BalBot. Bluetooth-Modul und Potis sind optional.

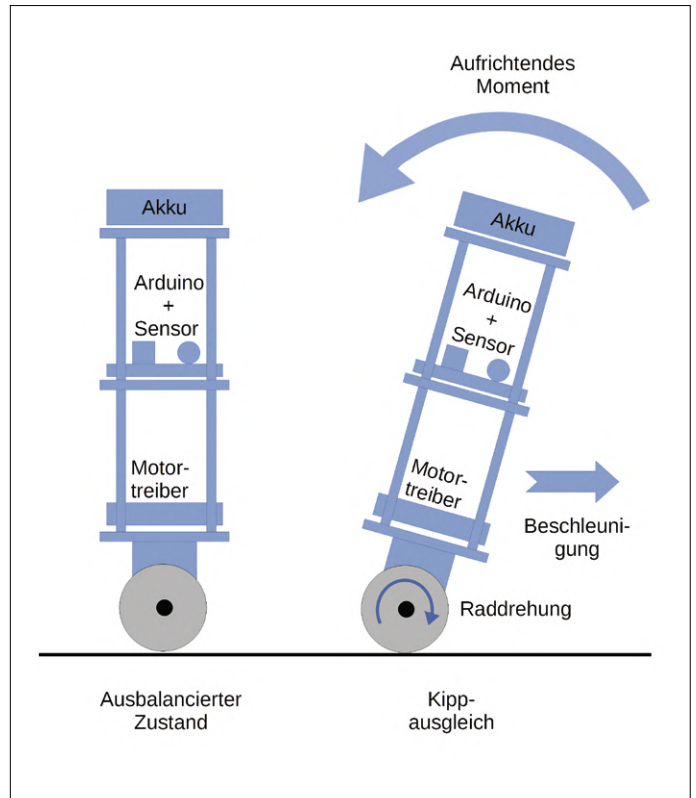


Bild 2. Prinzip der Selbstbalancierung.



## STÜCKLISTE

Arduino Uno  
Motortreiber(-Modul) L298  
MPU6050  
2 DC-Getriebemotoren, z.B. PGM37DC12 o.ä.  
Akkus (7 NiMH oder 2 LiPo)  
Sicherung 2 A  
Ggf. Schalter für Stromversorgung

Regeln bedeutet also, dass die Ausgangsgröße (z.B. Drehzahl) erfasst wird und bei einer Abweichung vom angepeilten Wert die Stellgröße (hier die PWM) passend verändert wird. Die zentralen Begriffe sind:

- Istwert:  $x$
- Sollwert:  $w$
- Regelabweichung:  $e = w - x$
- Stellgröße:  $y$
- Störgröße:  $z$

Diese Größen reichen zur Beschreibung eines Reglers aus. Weiter gibt es Reglertypen mit unterschiedlichen Regelungscharakteristiken. Sehr gebräuchlich und viele Fälle abdeckend ist der sogenannte PID-Regler. Er verfügt sowohl über einen proportionalen (P) als auch einen integralen (I) und einen differentiellen (D) Pfad. Durch Anpassung seiner Parameter können viele regelungstechnische Probleme mit hoher Präzision gelöst werden.

## Balance auf einer Achse

Ein interessantes Beispiel für die Anwendung solcher Regelungen ist ein selbstbalancierender Roboter. Prinzipiell handelt es sich hierbei um ein inverses Pendel auf Rädern. Das inverse Pendel ist aufgrund seiner Instabilität ein klassisches Beispiel für die Anwendung eines aktiven Regelkreises.

Der bekannte Segway-Roller ist ein einachsiges, zweirädriges, elektrisches Transportmittel. Die zu befördernde Person steht zwischen den Rädern quasi auf der Achse. Einachsige Fahrzeuge sind prinzipiell instabil. Also sorgt eine Regelung für die nötige Balance. Damit zu fahren macht Spaß. Die Popularität solcher Roller triggerte dann die Entwicklung selbstbalancierender Roboter.

Einen Roller nach dem „Segway-Prinzip“ nachzubauen ist nicht wirklich schwierig. Es gibt sogar komplette Bausätze. Zur Beschäftigung mit der zugrunde liegenden Regelungstechnik muss es aber nicht zwingend ein solcher Roller sein: Hierfür genügt der Bau eines kleineren, selbstbalancierenden Roboters, wie er nachfolgend beschrieben wird.

Damit ein selbstbalancierender Roboter das Gleichgewicht halten kann, müssen seine Räder permanent der Kippneigung entgegenwirken. Dies erfordert einen geeigneten Regelkreis, einen Sensor und Stellglieder. Als Sensorelement eignet sich das IC MPU6050 bestens, denn es enthält einen Beschleunigungssensor und ein Gyroskop. Damit kann man sehr präzise die Beschleunigung und die Drehung in den drei Raumachsen erfassen. Da solche Chips in vielen Geräten wie Kameras oder Smartphones eingesetzt werden, sind sie durch Massenproduktion sehr preisgünstig. Zur Verrechnung der Daten reicht schon ein Arduino Uno. Die mit den Rädern gekoppelten Motoren sind die Stellglieder.

Verbindet man den MPU6050 mit einem Arduino (siehe links unten in **Bild 1**), kann man die korrekte Funktion mit einem Testprogramm überprüfen. Im Download-Archiv zu diesem Artikel [1] ist hierfür „MPU6050\_test“ enthalten. Nach dem Laden des Testprogramms kann der serielle Monitor der Arduino-IDE gestartet werden. Sechs Linien zeigen die Reaktion der Sensorwerte auf Drehung bzw. Bewegung. Werden korrekte Bewegungsdaten angezeigt, kann man den Motortreiber mit dem Arduino verbinden. Als Treiber für kleine DC-Motoren eignet sich der Typ L298N (das IC oder ein damit bestücktes Modul). Als Energiequelle dienen entweder sieben 1,2-V-NiMH- (= 8,4 V) oder zwei 3,7-V-LiPo-Akkus (= 7,4 V). Bild 1 zeigt, wie die einzelnen Komponenten und Module (siehe Stückliste) miteinander verbunden werden.

## Balance-Algorithmen

Wie schon erwähnt, ist ein geeigneter Regelkreis erforderlich. Die Lage des Roboters entspricht der Istgröße. Sie soll möglichst wenig von der Senkrechten (Sollgröße) abweichen. Ein PID-Regler ist hierfür ideal. Unser Roboter namens BalBot erreicht mit optimal justierten Kp-, Ki- und Kd-Parametern ein erstaunlich stabiles dynamisches Gleichgewicht. Der Sollwert bzw. die Senkrechte wird zunächst fest in der Software eingestellt. **Bild 2** veranschaulicht, wie das in der Praxis funktioniert. Das MPU6050-Modul erfasst permanent die aktuelle Lage des Roboters. Seine Werte werden vom PID-Algorithmus im Arduino verarbeitet. Der Regelungs-Sketch führt die Berechnungen zur Steuerung der Motoren aus. Droht der Robot in eine Richtung umzukippen, wird das entstandene Kippmoment durch eine entsprechende Ausgleichsbewegung der Räder kompensiert. Der Roboter bleibt selbst dann senkrecht, wenn er leicht geschubst wird. Der zugehörige Sketch *BalBot.ino* ist ebenfalls im kostenlos erhältlichen Download-Archiv enthalten. Sein Code gliedert sich in sechs Teile:

1. Zunächst werden die Bibliotheken geladen. Soweit nicht standardmäßig in der Arduino-IDE enthalten, können sie via Internet [2][3] bezogen werden. Dann werden die Motoranschlüsse festgelegt. Über EN (enable) und IN (input) können die Motoren sowohl vorwärts als auch rückwärts betrieben werden. Die Motorleistung wird per PWM gesteuert.
2. Anschließend werden die für den Betrieb des MPU6050 erforderlichen Werte festgelegt. Hierbei handelt es sich im Wesentlichen um die Standardeinstellungen. Details dazu finden sich im Datenblatt des ICs [4].
3. Nun folgt die Festlegung der PID-Parameter. Sie sind von zentraler Bedeutung. Die folgenden drei Werte definieren das Regelverhalten: Kp (Bereich 0...1000; Standardwert 400), Kd (Bereich 0...100; Standardwert 30) und Ki (Bereich 0...500; Standardwert 200).
4. Diese Werte müssen an die realen Gegebenheiten der jeweiligen Hardware angepasst werden. Unterschiede beim mechanischen Aufbau (durch verschiedene Motoren, Radgrößen etc.) können hier ausgeglichen werden. Hinweise zur Justierung dieser Parameter finden sich im nächsten Abschnitt.
5. Im Setup werden jetzt die erforderlichen Schnittstellen aktiviert. Das serielle Interface dient lediglich zur Ausgabe von Debug-Infos. Dieser Code kann im „final release“ entfallen. Anschließend wird das MPU6050-Modul gestartet und auf Funktion geprüft - gegebenenfalls wird eine Fehlermel-

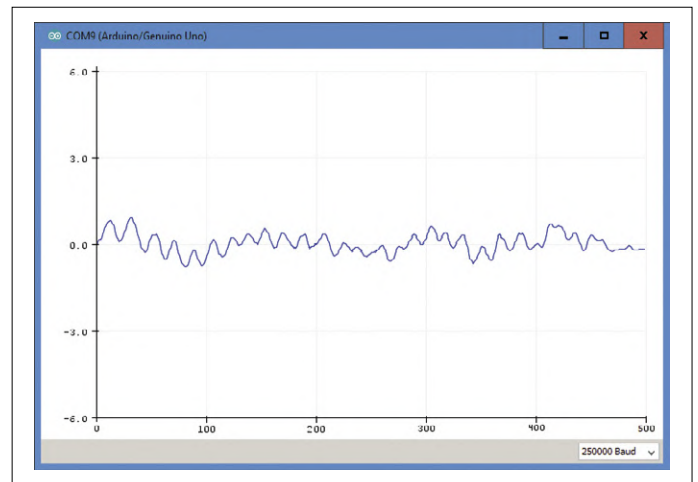


Bild 3. Kippwinkelverlauf im seriellen Plotter der Arduino-IDE.

- dung ausgegeben. Die angegebenen Offset-Werte können meist unverändert übernommen werden, da sich das Gyroskop automatisch kalibriert.
6. In der Hauptschleife wird lediglich das MPU-Signal ausgelesen sowie über den PID-Regler `pid.Compute()` das Steuersignal für die Motoren berechnet und an den Motortreiber mit folgendem Aufruf weitergeleitet:

```
motorController.move(PIDout, MIN_ABS_SPEED);
```

Darüber hinaus wird überprüft, ob der FIFO-Puffer des MPU6050 überläuft. Dies sollte im Normalbetrieb nicht passieren. Gegebenenfalls kann die Puffergröße angepasst werden.

Die Zeile:

```
Serial.println(ypr[1] * 180/M_PI); // actual tilt
```

gibt den Kippwinkel aus. Im seriellen Plotter der Arduino-IDE kann er kontinuierlich verfolgt werden. Falls regelmäßige, sinusförmige Werteverläufe auftauchen, ist dies ein klarer Hinweis für das Schwingen der Regelung. In diesem Fall müssen die Regelparameter angepasst werden. Meist ist es hilfreich, wenn der Kp-Wert reduziert wird. Im Idealfall sollte die Linie nur geringe, unregelmäßige Signalausschläge zeigen, wie sie durch zufällige Störeinflüsse (Luftbewegungen oder Bodenunebenheiten) entstehen. **Bild 3** zeigt die zufälligen Ausgleichsbewegungen um die Ruhelage.

## Parameteroptimierung und Justage

Die Optimierung der PID-Werte könnte über eine Simulation z.B. mit MATLAB erfolgen. Im professionellen Bereich ist dieses Vorgehen bei aufwändigen Reglern Standard. Beim BalBot kommt man jedoch auch mit dem Klassiker „trial and error“ zum Ziel. Dabei werden die einzelnen Parameter nach und nach angepasst, bis eine möglichst optimale Regelung erreicht ist:

1. Zunächst werden die drei Parameter Kp, Ki und Kd auf null gesetzt.
2. Dann wird Kp langsam schrittweise erhöht. Bei zu geringem Kp wird der Roboter umkippen, da die Korrekturen nicht schnell oder groß genug sind. Ist Kp zu groß, führt

der Roboter wilde Vor- und Zurückbewegungen aus, ohne dass sich ein Gleichgewichtszustand einstellt. Kp muss so eingestellt werden, dass sich der Roboter nur noch leicht hin und her bewegt.

3. Jetzt ist aufrechtes Stehen mit Ausgleichsbewegungen erreicht. Häufig tritt noch eine gewisse Schwingneigung auf: Der Roboter pendelt oder zittert rasch um die ideale Gleichgewichtslage.
4. Nun wird der Kd-Wert so eingestellt, dass sich die Schwingungen verringern. Im Idealfall verharrt der Roboter jetzt nahezu bewegungslos in aufrechter Position. Bei guter Einstellung von Kp und Kd bleibt er auch dann aufrecht, wenn er leicht angestoßen wird.
5. Abschließend wird der Ki-Parameter festgelegt. Auch bei optimalen Kp zeigt sich bei größeren Störungen noch eine

gewisse Schwingneigung. Er kann dabei sogar sein Gleichgewicht verlieren. Ki ist dann ideal eingestellt, wenn die Lagestabilisierung nach einer Störung schnell erfolgt.

Sind alle drei Parameter optimal eingestellt, rastet der Roboter nahezu in senkrechter Position ein, sobald der Fangbereich des Reglers erreicht wird. Wie von Geisterhand gehalten steht er dann aufrecht da und gleicht selbst größere Störeinflüsse souverän aus.

Falls kein Gleichgewicht erreicht werden kann, sind eventuell die Motoranschlüsse vertauscht. Der Roboter beschleunigt dann nicht in die Richtung des Kippmoments, sondern in die entgegengesetzte Richtung. **Bild 4** zeigt den fertigen Roboter in aufrechter, selbstbalancierender Position.

Da die Einstellung der Parameter per Software recht umständlich

ist, gibt es auch eine einfachere Variante.

Ein Arduino verfügt ja über mehrere analoge Eingänge. An diese können drei Potis angeschlossen werden, mit denen man die Werte für P, I und D direkt einstellen kann. Dazu muss die Software lediglich die Abfrage für die Poti-Werte enthalten (siehe **Listing Potis**).

So können die Parameter im laufenden Betrieb über die Potis optimiert werden (siehe *BalBot\_Pots\_BT.ino* im Download). Der Anschluss der optionalen Potis ist in Bild 1 schon berücksichtigt.

#### Listing Potis.

```
void readPIDTuningValues()
{
  int potKp = analogRead(A0);
  int potKi = analogRead(A1);
  int potKd = analogRead(A2);
  kp = map(potKp, 0, 1023, MIN_P, MAX_P) / 100.0;
  ki = map(potKi, 0, 1023, MIN_I, MAX_I) / 100.0;
  kd = map(potKd, 0, 1023, MIN_D, MAX_D) / 100.0;
}
```

#### Listing Bluetooth.

```
void readPIDTuningValues()
{
  int potKp = analogRead(A0);
  int potKi = analogRead(A1);
  int potKd = analogRead(A2);
  kp = map(potKp, 0, 1023, MIN_P, MAX_P) / 100.0;
  ki = map(potKi, 0, 1023, MIN_I, MAX_I) / 100.0;
  kd = map(potKd, 0, 1023, MIN_D, MAX_D) / 100.0;
}

content=blue.readString();

#if LOG_BT
  Serial.print("content: ");Serial.print(content);
  Serial.print(" - setpoint: "); Serial.print(setpoint);
  Serial.print(" - rotation: "); Serial.println(rotate);
#endif

if(content[0]=='F')
  // forward
else if(content[0]=='B')
  // backward
else if(content[0]=='L')
  // left
else if(content[0]=='R')
  // right
```

#### Fernsteuerung per Smartphone

Der Bau eines selbstbalancierenden Roboters ist eine interessante Herausforderung. Funktioniert alles, ist dieses zunächst erfreuliche Ergebnis auf Dauer nicht mehr spannend. Eine Fernsteuerung könnte das ändern. Damit ließe sich BalBot drahtlos in alle Richtungen bewegen. Da jeder (Elektroniker) ein Smartphone mit Bluetooth-Interface hat und viele davon unter Android laufen, dient ein solches Einfachheit halber zur Steuerkonsole des Roboters.

Hierzu muss lediglich ein Bluetooth-Modul an den Arduino angeschlossen werden. Gut geeignet ist ein HC05/06-Receiver, der auch schon in Bild 1 enthalten ist.

Zu seiner Verwendung muss dann noch die Standard-Bluetooth-Library der Arduino-IDE eingebunden und der Sketch entsprechend ergänzt werden (**Listing Bluetooth**).

Auch diese Erweiterung ist bereits im Sketch *BalBot\_Pots\_BT.ino* enthalten. Auf dem Smartphone wird dann lediglich die App „Arduino Bluetooth“ von Circuit Magic benötigt. Die App kann kostenlos aus dem Playstore geladen werden [5]. Nach Installation und Verbindung mit dem HC05/06-Modul sendet die App die notwendigen Steuerbefehle



an den Arduino. **Bild 5** zeigt die Bedienungsfläche der App. Die Steuerung erfolgt über die Pfeiltasten auf der linken Bildschirmhälfte.

### Fazit & Ausblick

Der hier vorgestellte Roboter lässt sich recht einfach aufbauen. Neben den Antriebsmotoren sind lediglich ein Arduino, ein Motortreiber und ein Beschleunigungs/Gyroskop-Modul erforderlich. Durch zusätzliche Potis wird die Justierung der Regelparame-ter deutlich vereinfacht. BalBot kann sogar per Bluetooth ferngesteuert durch die eigenen vier Wände bewegt werden.

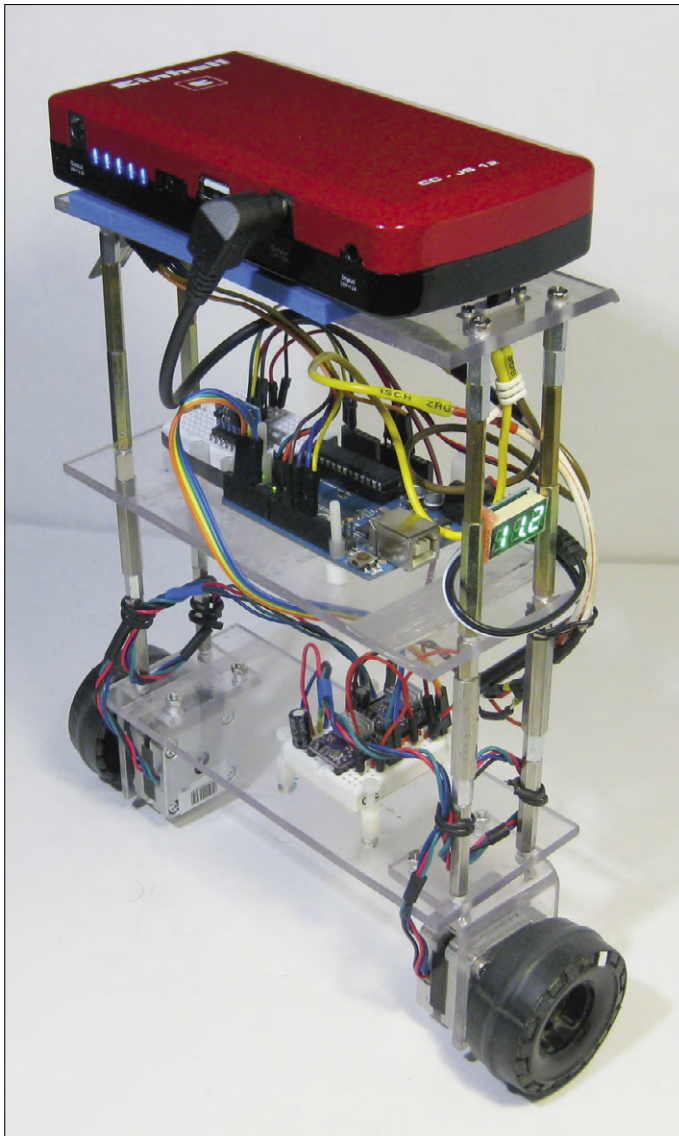


Bild 4. BalBot in Aktion.

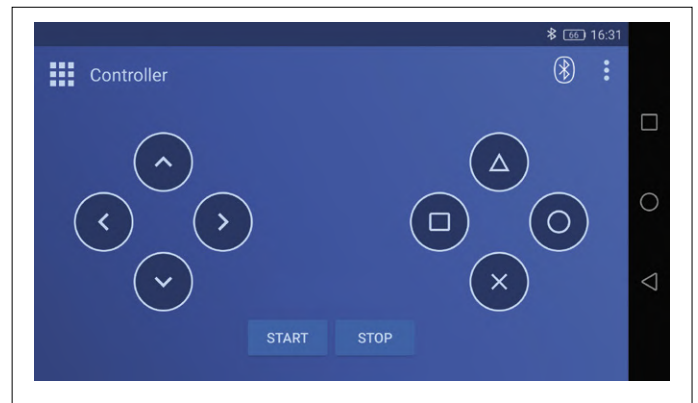


Bild 5. Android-App zur Steuerung von BalBot via Bluetooth.

Eine weitere Verbesserung von Balance und Fahrverhalten besonders bei ferngesteuertem Betrieb ließe sich durch sogenannte Rotary-Encoder erreichen. Damit kann die Umdrehungsgeschwindigkeit der Räder erfasst werden. Mit dieser Information lässt sich die Stabilität beim Vor- und Zurückfahren nochmals deutlich verbessern. In der Praxis verliert der ferngesteuerte BalBot gelegentlich das Gleichgewicht, wenn nur die Werte des MPU6050 zur Verfügung stehen. Dieser Sensor kann nämlich leider nur Beschleunigung und Rotation, nicht aber die Geschwindigkeit direkt erfassen.

Eine andere Alternative besteht in der Verwendung von Schrittmotoren. Durch das Zählen der Schritte erhält man eine ähnliche Information wie durch Encoder. Neben dem Tausch der Motoren muss dann natürlich auch der L298 durch geeignete Stepper-Treiber (wie etwa den DRV8825-Treiber) ersetzt werden.

Im Buch „Robotik und Künstliche Intelligenz“ des Autors findet man viele weitere Infos und Robotik-Projekte (siehe Kasten „Im Elektor-Store“). ◀

200074-01

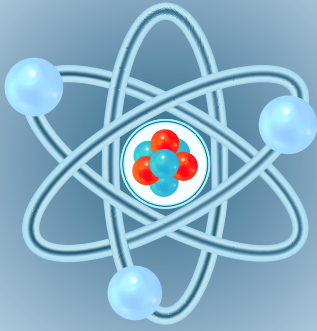
**IM ELEKTOR-STORE**

→ Buch „Robotik und Künstliche Intelligenz“

[www.elektor.de/robotik-und-kunstliche-intelligenz](http://www.elektor.de/robotik-und-kunstliche-intelligenz)

### Weblinks

- [1] Webseite zu diesem Artikel: [www.elektormagazine.de/200074-01](http://www.elektormagazine.de/200074-01)
- [2] PID\_v1.h: [https://github.com/br3ttb/Arduino-PID-Library/blob/master/PID\\_v1.h](https://github.com/br3ttb/Arduino-PID-Library/blob/master/PID_v1.h)
- [3] MPU6050\_6Axis\_MotionApps20.h: <https://bit.ly/37T7T1c>
- [4] MPU6050 Datenblatt: [www.invensense.com/products/motion-tracking/6-axis/mpu-6050/](http://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/)
- [5] App „Arduino Bluetooth“: <https://play.google.com/store/apps/details?id=com.circuitmagic.arduino bluetooth&hl=de>



# Aller Anfang... muss nicht schwer sein!

Von **Eric Bogers**

In der vorhergehenden Folge haben wir uns mit dem Ohmschen Gesetz beschäftigt und einige grundlegende Berechnungen durchgeführt. Jetzt wird es etwas komplizierter - aber keine Angst: Wir brauchen keine höhere Mathematik...

## Reihenschaltung von Widerständen

Was passiert, wenn wir wie in **Bild 1** zwei Widerstände in Reihe schalten? Betrachtet man einen Widerstand als ein Bauteil, das den Strom „behindert“, könnte man wie folgt argumentieren: Der Strom wird durch R1 behindert, und der Teil des Stroms, den dieser Widerstand durchlässt, wird wiederum von R2 behindert. Es ist also offensichtlich, dass der Gesamtwiderstand zunimmt, wenn man zwei (oder mehr) Widerstände in Reihe schaltet. Ja, wir wissen, dass diese „Argumentation“ an allen Ecken klemmt, aber sie versinnbildlicht doch das Geschehen.

Nun überprüfen wir, was genau passiert. Dank des Ohmschen Gesetzes wissen wir, dass die Spannung U über der Reihenschaltung von R1 und R2 (die wir R nennen) gleich

$$U = R \cdot I$$

ist. Nun sind Widerstände keine mystischen Bauteile - in ihnen geht kein Strom verloren. Der Strom I in die Reihenschaltung (manchmal auch Serienschaltung genannt) hinein ist also gleich dem Strom, der die Reihenschaltung verlässt. So fließt

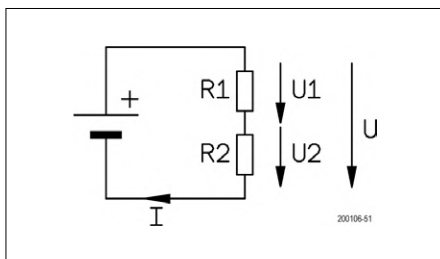


Bild 1. Reihenschaltung von zwei Widerständen.

durch beide Widerstände derselbe Strom. Über jeden dieser Widerstände fällt eine bestimmte Spannung ab (die sich mit dem Ohmschen Gesetz berechnen lässt), und da solche Widerstände rein *passive* Bauelemente sind, wird auf dem Weg keine Spannung und kein Strom hinzugefügt (und es verschwindet auch keine Spannung und kein Strom), so dass die Summe der beiden Teilspannungen U1 und U2 gleich der Gesamtspannung U sein muss. Mit diesem Wissen können wir schreiben:

$$U_1 = R_1 \cdot I$$

$$U_2 = R_2 \cdot I$$

Und weil  $U_1 + U_2 = U$  ist:

$$U = (R_1 \cdot I) + (R_2 \cdot I) = (R_1 + R_2) \cdot I$$

Daraus folgt, dass der Gesamtwiderstand R gleich der Summe der Widerstände R1 und R2 ist. Und seien Sie ehrlich: Sie haben das natürlich schon lange vermutet, und es ist auch offensichtlich, aber es ist natürlich schön, es beweisen zu können.

Wir überlassen es Ihnen, das Obige anhand eines Zahlenbeispiels zu überprüfen, zum Beispiel mit einer (Batterie-) Spannung von 15 V,  $R_1 = 10 \Omega$  und  $R_2 = 20 \Omega$ .

## Parallelschaltung von Widerständen

Widerstände können nicht nur in Reihe („hintereinander“), sondern auch parallel („nebeneinander“) geschaltet werden, wie in **Bild 2** skizziert ist. Denken Sie zunächst einmal einen Moment lang nur an R1, durch den, wie wir wissen, ein bestimmter Strom fließt. Jetzt fügen

wir R2 hinzu, so dürfte auch dort ein gewisser Strom fließen. Das bedeutet, dass bei einer Parallelschaltung der gesamte Strom ansteigt; und da sich die Spannung an den Widerständen nicht ändert, kann es nur bedeuten, dass der Gesamtwiderstand einer Parallelschaltung kleiner ist als die Einzelwiderstände! Wenn wir Bild 2 ansehen, fallen zwei Dinge auf: Über den beiden Widerständen R1 und R2 liegt die gleiche Spannung U. Es muss auch stimmen, dass die Summe beider Ströme durch die einzelnen Widerstände gleich dem Gesamtstrom I ist. Da wir es mit rein passiven Bauteilen zu tun haben, verschwindet kein Strom und es wird auch keiner hinzugefügt. Mit dem Gesamtwiderstand der Parallelschaltung R gilt wieder das Ohmsche Gesetz:

$$U = R \cdot I$$

Wenden wir nun das Ohmsche Gesetz auf die einzelnen Widerstände an, so sind:

$$U = R_1 \cdot I_1 \rightarrow I_1 = U / R_1$$

$$U = R_2 \cdot I_2 \rightarrow I_2 = U / R_2$$

Und weil  $I_1 + I_2 = I$ , schreiben wir

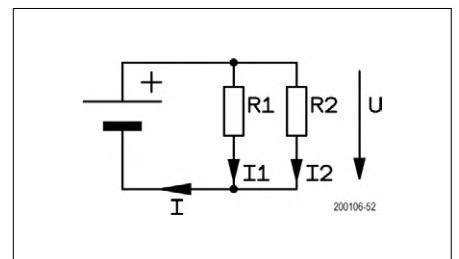


Bild 2. Parallelschaltung von zwei Widerständen.

$$I = (U / R_1) + (U / R_2) = U / R$$

Daraus folgt

$$1 / R = (1 / R_1) + (1 / R_2)$$

oder anders geschrieben:

$$R = 1 / \{ (1 / R_1) + (1 / R_2) \}$$

Wir überlassen Ihnen als aufmerksamen Leser, dieses Ergebnis anhand eines Zahlenbeispiels zu überprüfen.

### Gemischte Schaltung

In der täglichen Praxis der Elektronik haben wir es nicht nur mit „reinen“ Reihen- und Parallelschaltungen zu tun, sondern auch mit „gemischten Schaltungen“ wie in **Bild 3**. Dort sehen wir eine Reihenschaltung eines einzelnen Widerstandes ( $R_1$ ) mit einer Parallelschaltung von zwei Widerständen ( $R_2 || R_3$ ). Hinweis: Eine Parallelschaltung wird oft durch zwei parallele vertikale Striche  $||$  angedeutet.

Sehen wir uns hier ein Zahlenbeispiel an:  $R_1 = 10 \Omega$ ,  $R_2 = 20 \Omega$  und  $R_3 = 30 \Omega$ . Obwohl es möglich ist, eine einzige Formel für den Gesamtwiderstand aufzuschreiben, fördert ein solches (oft unübersichtliches) Vorgehen Schreib- und Rechenfehler. Es ist besser, den Gesamtwiderstand in zwei Schritten zu berechnen, zuerst der Parallelschaltung und dann der Serienschaltung. Für die Parallelschaltung von  $R_2$  und  $R_3$  ist:

$$1 / R_{\text{parallel}} = (1 / R_2) + (1 / R_3)$$

Geben Sie die Beispielwerte ein:

$$1 / R_{\text{parallel}} = (1 / 20 \Omega) + (1 / 30 \Omega) = (3 / 60 \Omega) + (2 / 60 \Omega) = 5 / 60 \Omega$$

Der Kehrwert wäre dann:

$$R_{\text{parallel}} = 60 \Omega / 5 = 12 \Omega$$

Dieser Widerstand liegt in Reihe mit  $R_1$ ; für den Gesamtwiderstand  $R$  dieser gemischten Schaltung ergibt sich

$$R = 10 \Omega + 12 \Omega = 22 \Omega.$$

### Einige Anmerkungen

Für schnelle Abschätzungen gelten folgende Faustregeln, wie Sie leicht verifizieren können:

- Bei einer Reihenschaltung von Widerständen ist der Gesamtwiderstand immer größer als der größte der in Reihe geschalteten Widerstände.

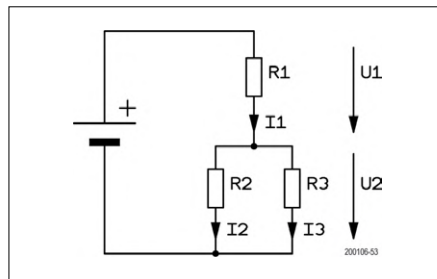


Bild 3. Einfache gemischte Schaltung.

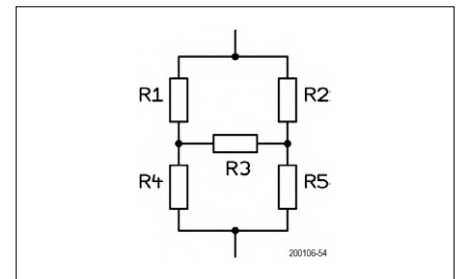


Bild 4. Die H-Schaltung, die besonders bei Prüfungen gefürchtet ist.

- Bei einer Parallelschaltung von Widerständen ist der Gesamtwiderstand immer kleiner als der kleinste der parallel geschalteten Widerstände.

Wenn wir zum Beispiel eine Reihenschaltung aus einem sehr großen und einem sehr kleinen Widerstand haben (sagen wir  $100 \text{ k}\Omega$  und  $10 \Omega$ ), dann berechnen wir in der alltäglichen Elektronikpraxis nicht genau, sondern sagen einfach, dass der Gesamtwiderstand gleich dem größten Widerstand ist, also  $100 \text{ k}\Omega$ . Und bei Parallelschaltung der gleichen Widerstände gälte: Wir ersparen uns die Mühe der Berechnung und sagen, dass der Gesamtwiderstand gleich dem kleinsten Widerstand ist, also  $10 \Omega$ . Der Unterschied zu den tatsächlichen Werten ist so gering, dass er in der Toleranz der Widerstände verschwindet und wir uns darüber keine Gedanken machen müssen.

Und noch eine Bemerkung fast zum Schluss: Wir haben uns bisher auf die Reihen- und Parallelschaltung von je zwei Widerständen beschränkt, aber sinngemäß gilt alles, was wir begründet, bewiesen und berechnet haben, auch für Reihen- und Parallelschaltungen von mehr als zwei Widerständen:

- Reihenschaltung von  $n$  Widerständen:  
$$R_{\text{Reihe}} = R_1 + R_2 + \dots + R_n$$
- Parallelschaltung von  $n$  Widerständen:  
$$1 / R_{\text{parallel}} = (1 / R_1) + (1 / R_2) + \dots (1 / R_n)$$

Kurz zurück zu Bild 2: Auf Grundlage dieses Bildes stellten wir fest, dass die Summe der Ströme  $I_1$  und  $I_2$  gleich dem Gesamtstrom  $I$  ist. Das scheint ziemlich offensichtlich, aber in der Physik (und Elektronik ist nichts anderes als angewandte Physik) ist etwas nur dann offensichtlich, wenn es auch zweifelsfrei bewiesen ist. Diesen Beweis hat der Physiker Gustav Kirchhoff bereits im Jahr 1845 geliefert. Das erste Gesetz von Kirchhoff besagt nämlich, dass die Summe der Ströme **in** einen Knoten gleich der Summe der Ströme **aus** diesem Knoten ist.

### Die H-Schaltung – ein kleiner Blick in die Zukunft

Mit dem jetzt erworbenen Wissen können wir einfache Mischschaltungen wie in Bild 3 ohne große Probleme berechnen. Aber schauen Sie sich einmal **Bild 4** an – eine gemischte Schaltung der kniffligen Art. In dieser so genannten H-Schaltung gibt es nämlich keine Kombinationen, die wir separat als Serien- oder Parallelschaltungen berechnen könnten. Wie können wir bei einer solchen Schaltung vorgehen?

Denken Sie darüber nach! in der nächsten Folge werden wir natürlich darauf zurückkommen (Stichwort: Stern-Dreieck-Transformation beziehungsweise Dreieck-Stern-Transformation). Und dann werden wir uns auch mit Wechselströmen und -spannungen befassen. ◀

200106-04

**IM ELEKTOR-STORE**

→ Basiskurs Elektronik (PDF): [www.elektor.de/basiskurs-elektronik-pdf](http://www.elektor.de/basiskurs-elektronik-pdf)





# Von der Pike auf gelernt

## Neues aus der Elektor-Ideenkiste

Von **Eric Bogers**

Kleine Schaltungen, die ohne SMDs und Spezialwerkzeuge auf einer Lochrasterplatine aufgebaut werden können und zum Experimentieren und Improvisieren einladen - sie sind oft das Sprungbrett für eine Karriere in der Elektronik. Hier präsentieren wir Ihnen eine weitere Auswahl aus dem Vorschläge-Kasten Ihrer Lieblingszeitschrift.

Idee: Michael A. Shustov (Russland) und Andrey M. Shustov (Deutschland)

### Energieeffizientes lichtempfindliches Relais

In der ersten Folge dieser Rubrik (Elektor Januar/Februar 2020) haben wir einige Variationen zum Thema „energieeffizientes Relais“ – eigentlich Varianten einer Treppenhausschaltung - von Vater und Sohn Shustov beschrieben. Seitdem waren die Shustovs nicht untätig: **Bild 1** zeigt eine mit einem lichtempfindlichen Transistor ausgestattete Schaltung, bei der die Lampe nicht nur durch das Drücken einer der Tasten SB1...SBn, sondern auch aus der Ferne, zum Beispiel mit einer Taschenlampe, eingeschaltet werden kann.

Wenn wir uns T3 für einen Moment wegdenken, funktioniert die Schaltung wie in der ersten Folge beschrieben. Im Ruhezustand sind die Drucktasten SB1...SBn (der Einfachheit halber ist nur ein Taster eingezeichnet) geschlossen. Ein kleiner Strom kann über R1 und LED1 durch die Lampe zur Masse fließen - die LED leuchtet auf, um anzuzeigen, dass die Schaltung „scharfgestellt“ ist. Der Kondensator C1 wird über die Diode D1 geladen. Übrigens ist der Strom durch die Lampe nicht hoch genug, als dass selbige aufleuchten würde. Und es wird klar, warum es sich bei der Lampe um eine Glühbirne handeln muss!

Sobald eine Taste gedrückt wird, fließt kein Strom mehr durch die Lampe und die Basis des Transistors T1 wird über R3, R2, R1 und LED1 auf Massenniveau gezogen. Der Transistor geht in den leitenden Zustand, so dass sich C1 über R4 entladen kann. Dadurch wird die Spannung über R4 hoch genug, um T2 in den leitenden Zustand zu versetzen. Durch diesen Transistor fließt nun ein ausreichend hoher Strom, damit die Lampe leuchten kann (sobald der Druckknopf losgelassen wird).

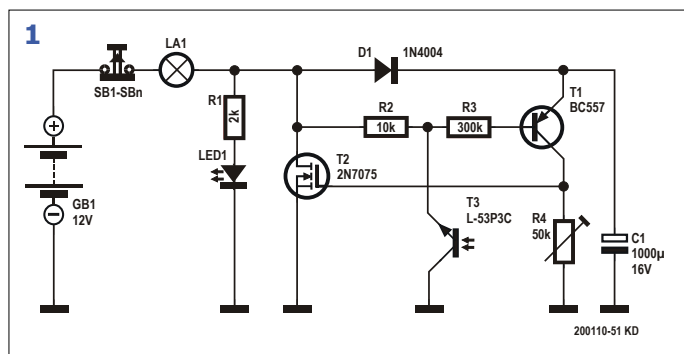
Diese Situation ist stabil, solange die Spannung an R4 hoch genug bleibt, um T2 im leitenden Zustand zu halten. Irgendwann aber ist der Kondensator C1 so weit entladen, dass T2 wieder sperrt und die Anfangssituation wieder hergestellt ist. Nach kurzer Zeit ist nämlich C1 (über D1) wieder aufgeladen und die Schaltung „scharf“. Setzen wir Transistor T3 wieder ein: Er macht eigentlich dasselbe wie die Drucktasten. Ist es dunkel, sperrt dieser lichtempfindliche Transistor und die Schaltung ist im Ruhezustand. Sobald Licht auf die Basis des Transistors fällt, zum Beispiel von einer Taschenlampe, schaltet er durch und zieht die Basis von T1 auf Masse. Dann verhält sich die Schaltung genau wie nach einem Druck auf einen der Taster. Auf diese Weise ist die Schaltung auch für den Einsatz im dunkelsten Keller geeignet. Man muss nicht einmal nach dem Lichtschalter herumtasten, sondern einfach beispielsweise das Handy zücken und die Taschenlampenfunktion betätigen.

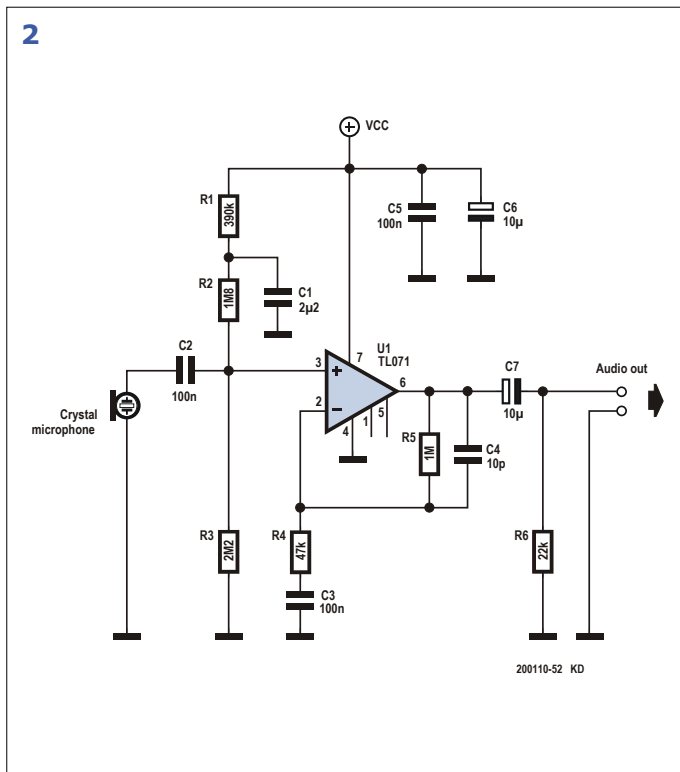


Idee: Joseph Kreutz (Frankreich)

### Vorverstärker für Kristallelemente

Erinnern Sie sich an diese so genannten „Tonmöbel“, die es bis in die 1970er Jahre gab? Diese großen Holzkisten mit einem Radio samt Verstärker (nichts war es mit Transistoren, alles Röhren) mit dicken mechanischen Wahltasten und den großen Lautstärke- und Abstimmknöpfen rechts und links neben einer





schönen „gestuften“ Abstimmkala und auf der anderen Seite des Möbels ein Plattenspieler für 33, 45 und 78 Umdrehungen pro Minute?

Dieses Grammophon... Antriebsriemen waren eine Rarität und der Direktantrieb musste erst noch erfunden werden. Der Plattenteller wurde mit einem Gummirad zwischen seinem Innenrand und der Motorwelle in Rotation versetzt. Und dieses Grammophon war oft mit einem piezoelektrischen Kristallelement ausgestattet.

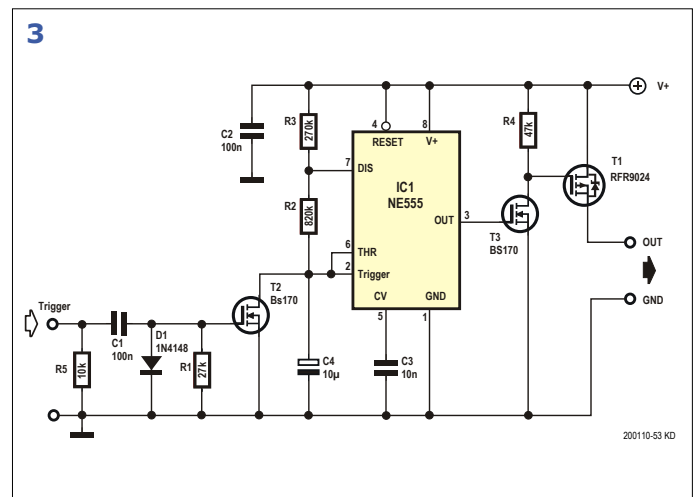
Ein solches Element lieferte eine relativ hohe Ausgangsspannung, durfte aber nicht niederohmig belastet werden. Letzteres war für die damalige Röhrenelektronik natürlich kein Problem, doch bei moderneren transistorisierten Verstärkern besitzt der Phono-Eingang (vorausgesetzt, er ist überhaupt vorhanden) eine normierte Eingangsimpedanz von 47 k $\Omega$ , und das ist viel zu niedrig. Doch nicht nur dies, moderne Phono-Verstärker sind für Signalpegel in der Größenordnung von nur einigen Millivolt ausgelegt. Das dürfte also garantiert nicht gut gehen. Um mit alten Kristall-Grammophonen experimentieren zu können, brauchen wir deshalb einen Vorverstärker/Impedanzwandler. Glücklicherweise ist so etwas leicht zu realisieren, wie **Bild 2** zeigt. Das Herzstück ist ein hochohmiger FET-Operationsverstärker namens TL071 (alt, bewährt und gut erhältlich). Mit den Widerständen R1...R3 wird die Eingangsimpedanz möglichst gut an das Element angepasst. Der Operationsverstärker wird nichtinvertierend betrieben, seine Bandbreite wird von den Kondensatoren C3 und C4 auf etwa 33 Hz bis knapp 16 kHz begrenzt. Der Verstärkungsfaktor liegt bei etwa 21, so dass der Vorverstärker direkt an den CD-Eingang eines modernen Verstärkers angeschlossen werden kann.

Über diese einfache Schaltung gibt es nicht viel mehr zu sagen, allenfalls, dass sie nicht nur für piezoelektrische Kristall-Abnehmer, sondern auch für piezoelektrische Mikrofone geeignet ist, wie sie damals mit vielen Tonbandgeräten geliefert wurden (und wie es auch im Schaltbild angegeben ist).

Idee: Wolfgang Borst (Deutschland)

### Externer Watchdog für Mikrocontroller-Systeme

Ein Mikrocontroller, der 24 Stunden am Tag und 7 Tage in der Woche in Betrieb ist, kann sich – auch wenn es glücklicherweise selten vorkommt – aus welchem Grund auch immer aufhängen. Und es kann auch passieren, dass ein normaler Reset, der durch den internen Watchdog des Controllers ausgelöst wird, nicht ausreicht, um aus dieser Situation zu entkommen. In diesen ganz seltenen Fällen hilft nur eines: die Stromversorgung vollständig zu unterbrechen und dann wieder einzuschalten, von Computernerds auch Hardreset genannt. Genau dies macht die kleine Schaltung in **Bild 3**: Es handelt sich um einen kleinen externen Watchdog, der, wenn er etwa 10 s lang keine Triggerimpulse vom Mikrocontroller erhält, die Stromversorgung für etwa 5 s ab- und dann wieder einschaltet. Die Schaltung ist mit einem „guten alten“ 555-Timer aufgebaut, eines der ältesten Arbeitspferde der Elektronik und ein immer noch ein beliebtes Bauteil, wenn es um Timing geht.



Die Funktionsweise der Schaltung ist nicht schwer zu ergründen: Die regelmäßigen Triggerimpulse des Mikrocontrollers (die auf diesem Controller laufende Software sollte sich natürlich darum kümmern, aber das dürfte für einen auch nur wenig erfahrenen Programmierer kein Problem sein) schalten immer den Transistor T2 durch, über den sich der Timing-Kondensator C4 des als astabiler Multivibrator geschalteten 555 entlädt. Auf diese Weise kann die Spannung an diesem Kondensator unter normalen Umständen niemals den Triggerpegel erreichen, der zur Aktivierung des 555 erforderlich ist. Der Ausgang (Pin 3) des 555 bleibt fortwährend high und die Stromversorgung für den Mikrocontroller über T3 und T1 eingeschaltet. Sobald die Triggerimpulse jedoch für mehr als 10 s ausfallen, wird die Spannung an C4 hoch genug, um den 555 zu triggern: Der Ausgang wird low und über T3 und T1 schaltet die Stromversorgung für den Controller ab. Nach etwa 5 s kippt der 555 wieder in seinen Ausgangszustand zurück und die Stromversorgung für den Controller wird wieder eingeschaltet. ◀

200110-03



# Manipulationssicheres Datenpaket

## Daten sicher mit der Post verschicken

Von **Luka Matić** (Kroatien)

In sicheren Kommunikationssystemen, die auf One-Time-Pads (OTPs) basieren, werden Nachrichten mit einem Zufallsschlüssel ver- und entschlüsselt. Damit dies funktioniert, müssen sowohl der Absender – normalerweise Alice genannt – als auch der Empfänger – besser bekannt als Bob – denselben Schlüssel verwenden. Alice erzeugt den Schlüssel, aber wie kann sie ihn auf sichere Weise an Bob senden? Es ist ein typisches Henne-Ei-Problem, das das in diesem Artikel beschriebene Projekt zu lösen versucht.



### Eigenschaften

- Open-Source-Design ohne versteckte Tricks
- Kapazitive Erkennung der Manipulation
- Umgebungslichterkennung der Manipulation
- Überprüfung der Lieferzeit
- Erkennung von Temperaturangriffen
- Infrarot-Kommunikation
- wiederverwendbar

Dieses Projekt ist eine logische Fortsetzung meines Zufallszahlen-Generators [1] und des OTP-Krypto-Shields [2] und versucht, das Problem des sicheren Austauschs von One-Time-Pad-Schlüsseln zu lösen, die zum Beispiel auf einer SD-Karte gespeichert sind. Ich begann mit der Idee, ein manipulationssicheres Gerät zu entwickeln, lernte dann aber, dass das sichere Löschen von Daten auf einem Gerät mit Flash-Speicher

wie einem Solid-State-Laufwerk (SSD) oder einer SD-Karte praktisch unmöglich ist [3]. Selbst SRAM-Speicher zeigen einige parasitäre Burn-in-Effekte, die eine Wiederherstellung der darin enthaltenen Daten ermöglichen [4].

Vor diesem Hintergrund habe ich beschlossen, keine manipulationssichere Schaltung zu bauen, sondern eine, die anzeigt, dass sie manipuliert wurde.



## Das Funktionsprinzip

Alice speichert einen frisch generierten OTP-Schlüssel auf einer microSD-Karte und legt ihn in das manipulationssichere Paket. Sie schließt die Box und macht sie scharf, indem sie einen Challenge-and-Response-Authentifizierungsstring über eine berührungslose (Infrarot-)Verbindung darin speichert. Dann schickt sie das „Datenpaket“ an Bob. Wenn er das Paket erhält, bittet er Alice um den Challenge-String. Dafür können die beiden jeden ungesicherten Kanal benutzen, zum Beispiel Telefon oder E-Mail, da nur Alice diese Zeichenfolge kennt und Bob das Paket hat. Über die optische Verbindung entsperrt Bob dann das Paket mit dem Challenge-String. Das Paket antwortet entweder mit dem Response-String oder mit einer Warnung über einen möglichen Manipulationsversuch. Nur Alice kann die Gültigkeit des Response-Strings bestätigen. Wenn der Response-String ungültig ist oder wenn das Paket vor einem Manipulationsversuch warnt, verwirft Bob einfach die SD-Karte und verwendet den potenziell unsicheren OTP-Schlüssel nicht für die sichere Kommunikation mit Alice.

Wenn das Paket geöffnet wird, ohne dass es zuvor mit dem Challenge-String entschlüsselt wurde, oder wenn das Paket auf andere Weise manipuliert wurde, löscht es die Challenge-and-Response-Strings aus dem Speicher. Die Lauscherin Eve oder die böswillige Fälscherin Mallory können das Paket abfangen, um die SD-Karte zu lesen, zu kopieren oder zu fälschen und das Paket an Bob weiterzuleiten, aber sie können das Paket nicht mit denselben Challenge-and-Response-Sequenzen wieder scharf machen, weil diese nur Alice kennt. So erkennt Bob zweifelsfrei jeglichen Manipulationsversuch!

## Beschreibung der Hardware

Die vollständige Schaltung des Manipulationssicheren Datenpakets ist in **Bild 1** zu sehen. Der Mikrocontroller IC1 ist das Herzstück des Geräts. Alle zwei Sekunden erwacht er für weniger als 50 ms aus seinem stromsparenden Schlafmodus, um verschiedene Parameter zu überprüfen. Wenn diese Parameter ein Manipulationsereignis anzeigen, löscht der Controller den Inhalt seines SRAMs auf eine sichere Weise (was bei einem Flash-Speicher oder EEPROM nicht

### INFOS ZUM PROJEKT

Sichere Kommunikation

Verschlüsselung

IR-Kommunikation

AVR

Einsteiger

Fortgeschrittene

Experte

ca. 3 Stunden

(SMD-)Lötkolben,

AVR-Programmer

ca. 50 €

möglich wäre).

Ein Öffnen des Pakets wird auf zwei Arten erkannt:

- Änderung der Kapazität des Pakets. C1x und C2x (**Bild 2**) sind offene

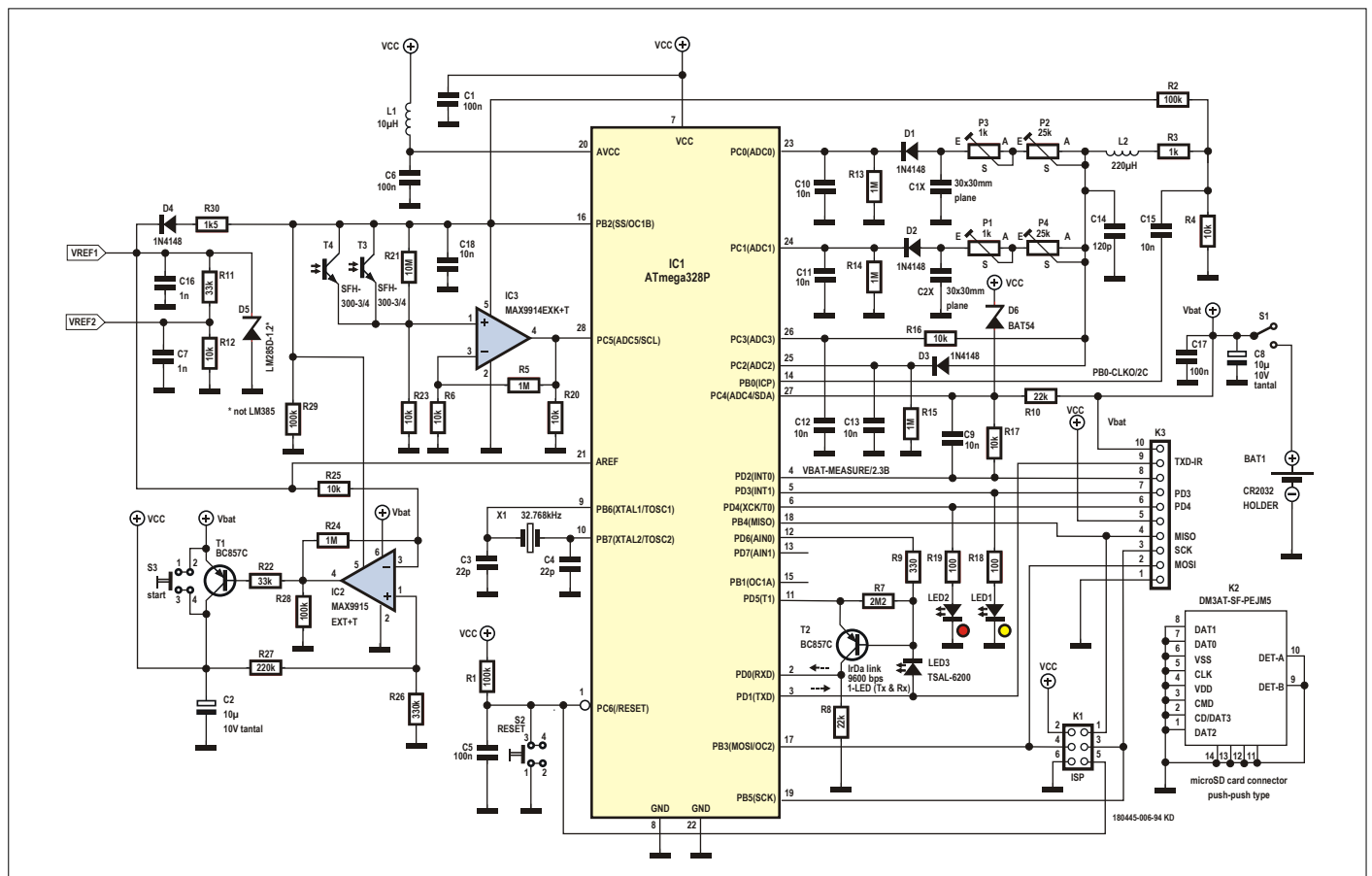


Bild 1. Das Manipulationssichere Paket verwendet Fototransistoren und offene Plattenkondensatoren, um eine Manipulation zu erkennen.

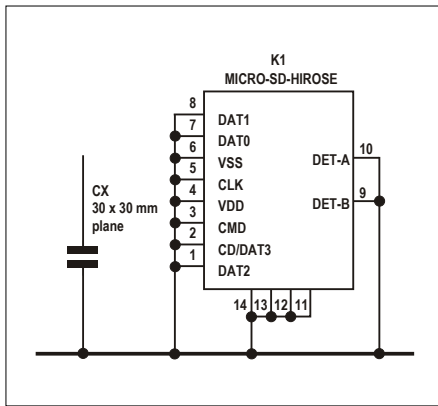


Bild 2. Die offenen Plattenkondensatoren C1x und C2x bestehen aus einseitig beschichteten, 30 x 30 mm<sup>2</sup> großen Platinenabschnitten. Der microSD-Card-Slot liegt innerhalb dieser Platten.

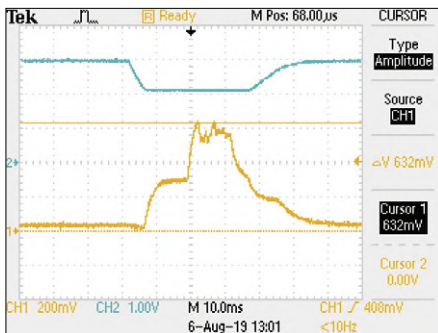


Bild 3. Um Strom zu sparen, wacht das Päckchen nur alle zwei Sekunden für etwa 50 ms auf. Der Spannungsregler wird aktiviert (obere Kurve), damit wichtige Manipulationsparameter gemessen werden können. Die untere Kurve zeigt die Stromaufnahme während der Wachzeit.

Plattenkondensatoren, die die dazwischen liegende SD-Karte „schützen“. Die jeweils andere Platte ist das Metallgehäuse (!) des Pakets. Eine Änderung der Kapazität deutet auf eine Manipulation hin.

- **Licht-Erkennung.** Das Paket ist geschlossen und hundertprozentig lichtundurchlässig. Im Inneren des Pakets ist es normalerweise völlig dunkel. Die Fototransistoren T3 und T4 erkennen infrarotes und sichtbares Licht, sobald die Schachtel geöffnet wird.

Für die Eingabe der Challenge-and-Response-Codes wird eine infrarote, IrDA-ähnliche Verbindung verwendet. Die bidirektionale Kommunikation findet über LED3 statt. Die maximale Reichweite beträgt zwar nur etwa einen Zentimeter, aber das ist mehr als genug. Die Schaltung wird von einer

CR2032-Batterie mit einer Nennspannung von 3 V gespeist. Sie hält länger als einen Monat (**Bild 3**), was selbst für die langsamste Schneckenpost ausreichen sollte. Eine frische Batterie weist eine Klemmenspannung von typisch 3,3 V auf, und die Schaltung arbeitet mit einer Eingangsspannung von nur 2,1 V. Dies ist ausreichend für den Low-Dropout-Spannungsregler (LDO) IC2, um für eine stabile Betriebsspannung ( $V_{CC}$ ) von 2,0 V zu sorgen.

D5 liefert eine präzise Referenzspannung für den Analog-Digital-Wandler (ADC) des Controllers. Der LM285 ist eine Variante des LM385 mit erweitertem Temperaturbereich, so dass die Schaltung sicher und mit ausreichender Präzision im Temperaturbereich von -20 °C bis +60 °C arbeiten kann, also bei allen Temperaturen, die während des Transports mit der Post auftreten können.

Der Controller wird von seinem internen 1-MHz-RC-Oszillator getaktet. Um den Energieverbrauch so gering wie möglich zu halten, wird dieser Taktgeber nur in Zwei-Sekunden-Intervallen für eine Dauer von etwa 50 ms eingeschaltet. Der auf dem Uhrenquarz X1 basierende 32,768-kHz-Taktgeber mit geringem Leistungsbedarf arbeitet kontinuierlich als Aufwecktimer. Er wird auch dazu verwendet, den UART-Takt des Controllers bei jedem Erwachen einzustellen und er liefert schließlich die Zeitbasis für die präzise Messung der Transportdauer des Pakets.

R3, L2 und C14 filtern das 1-MHz-Rechtecksignal, das vom Taktausgang PB0 des Controllers abgeleitet wird, um eine Sinuswelle zur genauen Messung der Kapazitäten von C1x und C2x zu erhalten. Es werden sowohl Spitzen- als auch Mittelwerte gemessen und die Verhältnisse der Wechsellspannung an C1x und C2x zur Wechsellspannung an C14 berechnet. Eine signifikante Änderung in einem dieser Verhältnisse deutet auf eine Manipulation hin.

LED1 und LED2 zeigen verschiedene Zustände des Pakets an. LED1 blinkt zum Beispiel, wenn ein korrekter Challenge-Code eingegeben wurde und LED2 blinkt, wenn er falsch war. Beide LEDs blinken nach dem Einschalten einmal. LED1 blinkt einmal bei jedem Aufwachen des Controllers, da sie auch hilft, den LDO-Ausgang schnell auf 2 V anzuheben. Wenn aber LED2 einmal bei jedem Aufwachen des Controllers blinkt, wurde ein Manipulationsversuch festgestellt.

Schalter S1 ist der Ein/Aus-Schalter des Pakets, S2 der Reset-Taster des Controllers. Der Drucktaster S3 ermöglicht es, die Schaltung auch dann noch zu booten, wenn die Batteriespannung etwas zu niedrig ist.

Beachten Sie, dass die Platine beidseitig bestückt und ein Teil der Hardware so den neugierigen Blicken der böswilligen Mallory verborgen ist, um es ihr schwerer zu machen, günstige Stellen für die Platzierung von Sonden oder „Störer“ zu finden, ohne die Platine auszubauen. Der Plattenkondensator auf der Unterseite würde dies sofort erkennen.

## Der IR-Kommunikationsadapter

Zur Kommunikation mit dem hermetisch verschlossenen Manipulationssicheren Datenpaket gibt es eine Infrarot-Verbindung, die zwar nicht besonders schnell ist (9.600 Baud) und eine eher geringe Reichweite besitzt (1 cm), aber nichtsdestotrotz funktioniert. So kann man mit dem Paket kommunizieren, ohne dass es geöffnet werden müsste. Die Adapter-schaltung (**Bild 4** und **Bild 5**), die dies ermöglicht, besteht aus einem kleinen Mikrocontroller, einem ATtiny45, der mit einem IR-Transceiver ausgestattet ist, der fast identisch mit der IR-Schaltung im Paket ist.

Da die Eingabe der Challenge-and-Response-Codes ein sicherheitskritisches Verfahren ist, stehen auf der Nutzerseite gleich mehrere serielle Anschlüsse zur Verfügung. So kann der IR-Adapter auf verschiedene Weise mit einem Computer verbunden werden, zum Beispiel über einen USB-Seriell-Konverter, über einen „echten“ RS232-Anschluss und sogar über Ethernet, wenn Sie einen Ethernet-Seriell-Adapter (WIZnet WIZ107) hinzufügen. Diese Optionen ermöglichen es, ein billiges Notebook, das niemals mit dem Internet in Verbindung steht, allein für die Eingabe der Codes bereitzuhalten.

## Einstellung der kapazitiven Detektoren

Für den korrekten Betrieb der kapazitiven Manipulationserkennung müssen die Trimpoties P1, P2, P3 und P4 eingestellt werden. Schalten Sie das Paket ein (LED2 blinkt einmal, gefolgt von LED1) und platzieren Sie in einer Entfernung von weniger als 1 cm die IR-LED des IR-Adapters vor LED3 des Pakets. Schließen Sie den IR-Adapter an einen Computer

mit einem seriellen Terminal (9600N81) an. Nachdem Sie `U` (0x55) am Terminal eingegeben haben, erhalten Sie alle zwei Sekunden einen Datenblock wie folgt:

```
Box unlocked. Status-1. Define
challenge string.
Capacitor Cx1 voltage ratio: 0.703
Capacitor Cx2 voltage ratio: 0.703
Phototransistor voltage [V]: 0.181
Box temperature [degC]: +22
Battery voltage [V]: 2.900
```

Für eine maximale Empfindlichkeit müssen die Trimpotis P1...P4 so eingestellt werden, dass der Widerstand P2+P3 gleich dem Blindwiderstand von C1x und der Widerstand P1+P4 gleich dem Blindwiderstand von C2x ist. Das klingt kompliziert, aber die Statusdaten im Terminal zeigen die Spannungsverhältnisse auf, die nahe bei  $1/\sqrt{2}$ , das heißt 0,707 liegen sollten. Da für die Plattenkondensatoren C1x und C2x das Gehäuse selbst als Kondensatorplatte dient, müssen die Spannungsverhältnisse bei aufgesetztem Gehäusedeckel korrekt sein. Verwenden Sie P2 und P4 für die Grobeinstellung und P1 und P3 für die Feineinstellung.

### Temperaturmessungen

Der Controller besitzt einen eingebauten Temperatursensor. Seine Präzision ist zwar gering – der Fehler liegt bei etwa  $\pm 5^\circ\text{C}$  – aber gut genug, dass der Controller abnormale Temperaturen erkennen und die ungefähren Minimal- und Maximaltemperaturen während des Pakettransports protokollieren kann.

Der interne Temperatursensor wird auf dem ADC-Kanal 8 ausgelesen. Obwohl der Sensor für den Betrieb mit der internen Spannungsreferenz des Controllers (1,1 V und nicht gerade stabil) ausgelegt ist, kann auch eine externe Spannungsreferenz (D5) verwendet werden, wenn wir etwas die Mathematik (eine affine Transformation) bemühen. Auf diese Weise kann eine viel bessere Genauigkeit als im Datenblatt des Controllers angegeben erreicht werden. Bei Bedarf kann die Steigung der affinen Abbildung im Programm in der Funktion `measure_signals()` angepasst werden.

Der Temperatursensor muss, um ihn wirklich nutzen zu können, kalibriert werden. Dies geschieht mit dem Befehl `TP`, der vom Terminal aus gesendet wird. Die Syntax ist `TPsxx`, wobei `s` das obligatorische (!) Vorzeichen `+` oder `-` und

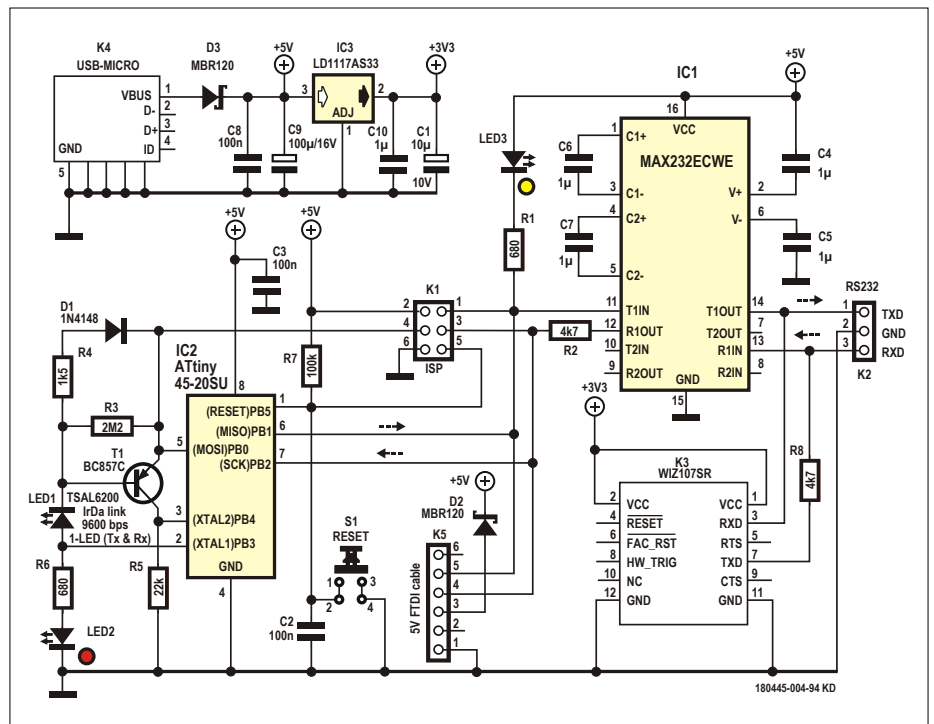


Bild 4. Die IR-Kommunikationsschaltung stellt eine Schnittstelle zwischen dem hermetisch verschlossenen Manipulationssicheren Datenpaket und einem Computer dar.

`xx` ein zweistelliger Offset zwischen `00` und `99` ist. Wenn zum Beispiel für die `Box temperature` der Wert `+40` angegeben wird und die Umgebungstemperatur  $+25^\circ\text{C}$  beträgt, dann sollten Sie am Terminal `TP-15` eingeben. Durch Senden von `TP+00` wird der Offset auf null zurückgesetzt. Der Offsetwert wird im EEPROM des Controllers gespeichert.

### Benutzerhandbuch für Spione

Alice erstellt mit ihrem Zufallszahlen-Generator eine Datei mit Zufallszahlen und kopiert diese auf eine noch nie benutzte

SD-Karte. Sie legt eine neue Batterie in das Datenpaket ein und schaltet es mit S1 ein. Da die Batterie frisch ist, ist es nicht notwendig, S3 zu drücken, um das Gerät zu starten. Dann steckt sie die SD-Karte in den Slot und schließt das Gehäuse. Mit dem IR-Adapter vor der IR-LED des Geräts sendet sie zunächst das Zeichen `U` (0x55) von ihrem Terminal. Dadurch kann der Controller des IR-Adapters seinen UART-Takt kalibrieren. Das Terminal empfängt nun Nachrichten vom Paket, was in etwa so aussieht:

```
Box unlocked. Status-1. Define
```

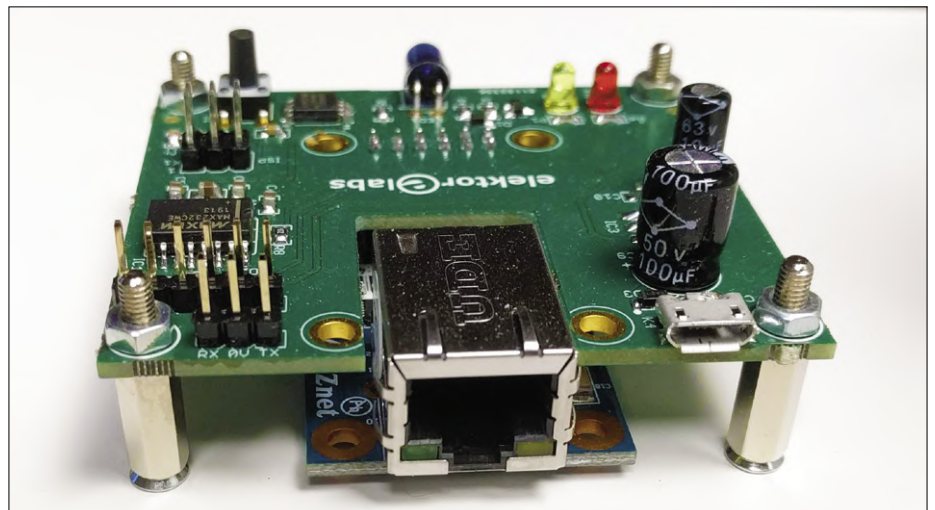


Bild 5. Der IR-Kommunikationsadapter mit dem optionalen Ethernet-Modul.



challenge string.

Capacitor Cx1 voltage ratio: 0.710  
Capacitor Cx2 voltage ratio: 0.704  
Phototransistor voltage [V]: 0.181  
Box temperature [degC]: +25  
Battery voltage [V]: 3.354

Das Paket gibt diese Datenblöcke im Zweisekundentakt über den Infrarot-Port aus, außer in Status 3 und Status 5, in denen die Datenübertragungen nach 60 s stoppen, um Strom zu sparen.

### Status-1

Status-1 ist der Ausgangszustand des Pakets nach dem Einschalten. Das Gerät wartet auf das Eintreffen eines Challenge-Strings an seinem Infrarotempfänger. Alice gibt diesen Challenge-String am Terminal ein (bis zu 65 Zeichen, längere Strings werden abgeschnitten). Die Box

geht danach in den Status-2 über.

*Wichtiger Hinweis: Alle Zeichenketten, einschließlich mehrzeiliger Befehle, müssen per Copy/Paste in das Terminal eingegeben werden und nicht Zeichen für Zeichen!*

Box unlocked. Status-2. Define response string.

Capacitor Cx1 voltage ratio: 0.710  
Capacitor Cx2 voltage ratio: 0.704  
Phototransistor voltage [V]: 0.181  
Box temperature [degC]: +25  
Battery voltage [V]: 3.354

### Status-2

Das Paket wartet auf die Eingabe eines Response-Strings. Die Spannungsverhältnisse von Cx1 und Cx2 sowie die Spannung am Fototransistor werden im SRAM des Controllers gespeichert. Wenn sich diese Spannungen während des Daten-

transports zu stark ändern, wird eine Manipulationswarnung ausgegeben. Alice gibt den Response-String (bis zu 65 Zeichen) ein, um das Paket scharf zu stellen, zu verschließen und es in den Status-3 zu versetzen. Die Uhr für die Transportzeit wird ebenfalls gestartet. Alice schreibt das Datum und die genaue Tageszeit zusammen mit den Challenge- und Response-Strings auf und versteckt alles sicher (wenn möglich in ihrem Kopf, praktikabler in einem Password-Safe). Diese Informationen werden später zur Überprüfung verwendet. Dann schickt sie das Paket zu Bob.

Box locked. Status-3. Enter

challenge string to unlock.

Capacitor Cx1 voltage ratio: 0.710  
Capacitor Cx2 voltage ratio: 0.704  
Phototransistor voltage [V]: 0.181  
Box temperature [degC]: +25



## STÜCKLISTE

### Manipulationssicheres Datenpaket

#### Widerstände:

(alle SMD 0805)

R18, R19 = 100  $\Omega$   
R9 = 330  $\Omega$   
R3 = 1 k  
R30 = 1 k5  
R4, R6, R12, R16, R17, R20, R23, R25 = 10 k  
R8, R10 = 22 k  
R11, R22 = 33 k  
R1, R2, R29, R28 = 100 k  
R27 = 220 k  
R26 = 330 k  
R5, R13, R14, R15, R24 = 1 M  
R7 = 2 M2  
R21 = 10 M  
P3, P1 = 1 k Trimpoti  
P2, P4 = 25 k Trimpoti

#### Kondensatoren:

(alle SMD 0805)

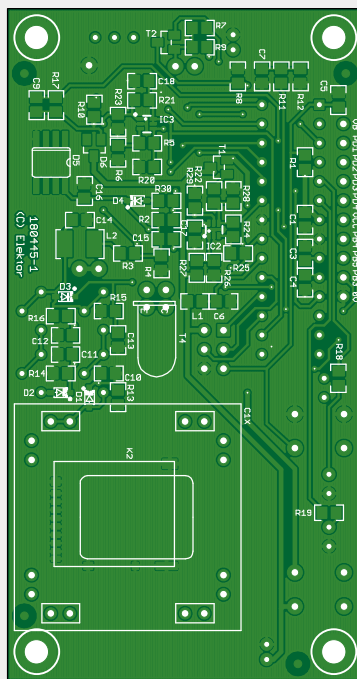
C3, C4 = 22 p  
C14 = 120 p  
C7, C16 = 1 n  
C9, C10, C11, C12, C13, C15, C18 = 10 n  
C1, C5, C6, C17 = 100 n  
C2, C8 = 10  $\mu$ , 10 V, Tantal, Raster 2 mm

#### Induktivitäten:

L1 = 10  $\mu$  (0805)  
L2 = 220  $\mu$  (1812)

#### Halbleiter:

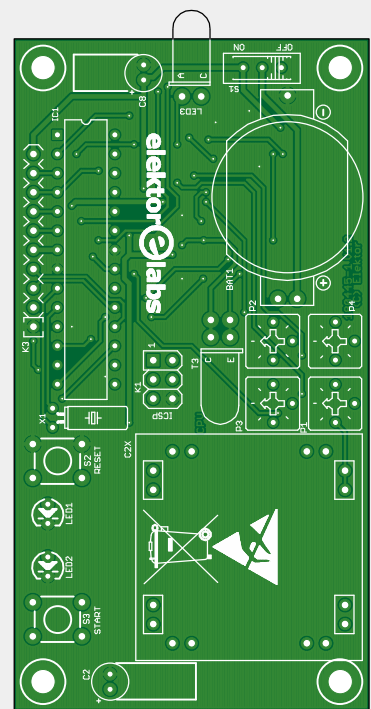
D1, D2, D3, D4 = 1N4148WS (SOD-323)  
D5 = LM285D-1.2V



D6 = BAT54  
IC1 = ATmega328P-PU, programmiert  
IC2 = MAX9915EXT+T  
IC3 = MAX9914EXK+T  
LED1 = 3mm, gelb  
LED2 = 3mm, rot  
LED3 = TSAL6200  
T1, T2 = BC857C  
T3, T4 = SFH-300-3/4

#### Außerdem:

BAT1 = Batteriehalter, 2032, z.B. Renata HU2032



C1x, C2x = 2-polige Stiftleiste + Fassung für C1x und C2x, Raster 0,1"  
K1 = 2x3-polige Stiftleiste, Raster 0,1"  
K2 = microSD-Card-Buchse, push-push  
K3 = 10-polige Stiftleiste, 0,1" Teilung  
S1 = Schiebeschalter, SPDT  
S2, S3 = Tastschalter, 6x6 mm<sup>2</sup>  
X1 = Quarz 32,768 kHz  
Geschlossener LED-Halter für LED3  
1 Platine 180445-1  
2 Platinen 180445-3

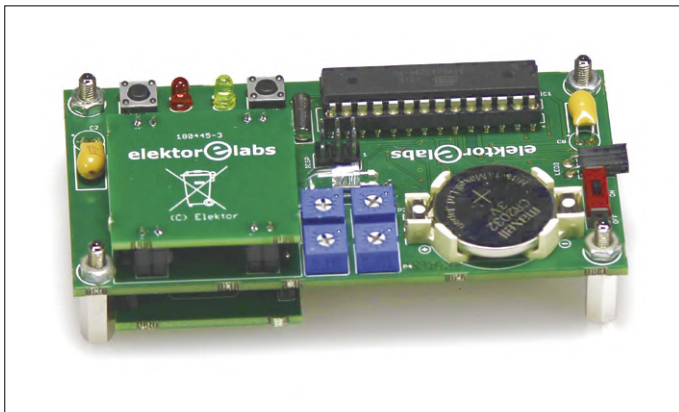


Bild 6. Die vollständig bestückte Platine mit beidseitig installierten Plattenkondensatoren.

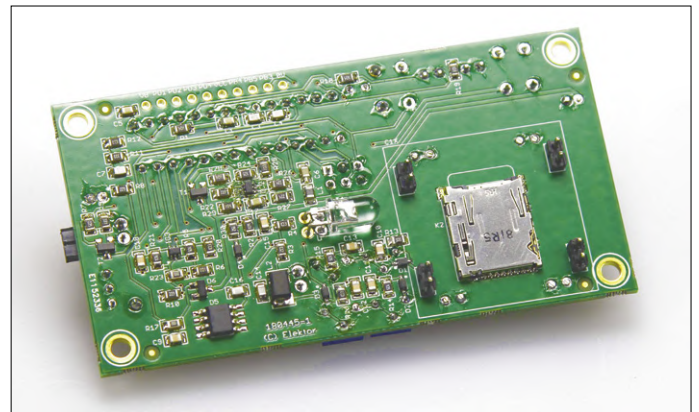


Bild 7. Die Unterseite der Platine ist fast vollständig mit SMD-Bauteilen bestückt. Nur Transistor T4, der wie eine LED aussieht, ist bedrahtet. Der Kondensator C1x wird über dem microSD-Card-Slot montiert.

Battery voltage [V]: 3.354

### Status-3

Challenge- und Response-String wurden eingegeben und das Paket ist scharf und verschlossen. Jede Manipulation oder die fünfmalige Eingabe eines falschen Challenge-Strings löst die Löschesequenz aus. Das Paket verbleibt während des Transports in Status 3.

Die SRAM-Puffer mit den Challenge- und Response-Strings werden alle zwei Sekunden komplementiert (bitweise invertiert). Auf diese Weise wird das parasitäre Einbrennen der SRAM-Zellen [4] verhindert, da jedes Bit gleich lange im Zustand „0“ und „1“ verbleibt.

### Das Datenpaket bei Bob

Nach einer langen und gefährvollen Reise kommt das Paket (hoffentlich) wohlbehalten bei Bob an. Er schließt seinen IR-Adapter an einen Computer an, stellt ihn vor die IR-LED des Pakets und drückt den Reset-Taster am Adapter. Dadurch wird ein 3 s wählender Impuls an der Infrarotschnittstelle erzeugt, um dem Paket mitzuteilen, dass das Gerät die Infrarotschnittstelle aktivieren und mit der Kommunikation beginnen soll. Nach drei Sekunden sendet Bob das einzelne Zeichen U, um den UART des IR-Adapters zu kalibrieren. Dann beginnt das Terminal, Daten anzuzeigen:

```
Box locked. Status-3. Enter
challenge string to unlock.
Capacitor Cx1 voltage ratio: 0.703
Capacitor Cx2 voltage ratio: 0.711
Phototransistor voltage [V]: 0.175
Box temperature [degC]: +23
```

Battery voltage [V]: 2.752

Wie zu sehen, ist die Batteriespannung gesunken, aber es gibt keine sichtbaren Anzeichen für Manipulationen, da sich die Cx-Spannungsverhältnisse und die Spannungswerte am Fototransistor nicht wesentlich geändert haben. Bob ruft Alice an und bittet sie um den Challenge-String. Er gibt ihn an einem Terminal ein, und wenn alles in Ordnung ist, erscheint:

```
Box unlocked. Status-4. Challenge
string correct!!
Capacitor Cx1 voltage
ratio: 0.703
Capacitor Cx2 voltage ratio: 0.711
Phototransistor voltage [V]: 0.178
Box temperature [degC]: +24
Battery voltage [V]: 2.752
Min temperature [degC]: -2
Max temperature [degC]: +31
Travel time [s]: 502,164
Response:
Trust no one 01a2g23w46e57f80g12r3
e34fv245hasdvfr4.
```

### Status-4

Bob schreibt sofort die aktuelle Uhrzeit und das Datum auf und ruft Alice erneut an, um den angezeigten Response-String zu verifizieren. Wenn Alice ihn akzeptiert, gibt sie Bob die genaue Uhrzeit und das genaue Datum an, die sie beim Abschließen des Pakets notiert hat. Mit dieser Angabe kann Bob die theoretische Reisezeit berechnen und mit der vom Paket angegebenen Zeit vergleichen. Wenn die beiden Zeiten übereinstimmen, darf Bob den auf der microSD-Card gespeicherten OTP-Schlüssel verwenden, um seine

sichere Kommunikation mit Alice zu ver- und entschlüsseln. Im Falle eines Manipulationsversuchs während des Transports löscht die Software den Teil des SRAMs, der die Codes und den Reisezeitgeber enthält. In diesem Fall gibt das Paket diese (oder eine ähnliche) Meldung aus:

```
Box zeroized. Status-5. Tampering,
wrong password, or low battery!
Capacitor Cx1 voltage ratio: 0.712
Capacitor Cx2 voltage ratio: 0.713
Phototransistor voltage [V]: 0.179
Box temperature [degC]: +12
Battery voltage [V]: 2.793
```

### Status-5

Bei der Erkennung eines Manipulationsversuchs oder nach fünfmaliger Eingabe eines ungültigen Challenge-Strings wird die Löschesequenz ausgelöst. Die Kontrollspannungen werden nicht mehr aktualisiert, so dass der Empfänger des Pakets den letzten aufgezeichneten Zustand vor der Erkennung einer Manipulation sehen kann.

Die Löschesequenz wird in einem der folgenden Fälle ausgeführt:

- nach fünfmaliger Eingabe eines falschen Challenge-Strings,
- das tatsächliche Spannungsverhältnis des Kondensators weicht zu stark vom aufgezeichneten Wert ab,
- die tatsächliche Spannung am Fototransistor weicht zu stark vom aufgezeichneten Wert ab,
- die Batteriespannung ist zu niedrig (unter 2,1 V) oder
- die Temperatur lag außerhalb des zulässigen Bereichs (unter -20 °C oder über +60 °C).

Um zu vermeiden, dass Streulicht den

dritten Fall auslöst, muss das Paket hermetisch/lichtdicht verschlossen sein. Tragen Sie etwas schwarze Dichtungsmasse (Butylpaste) um LED3 auf der Innenseite des Gehäuses auf und versiegeln Sie die Ränder des Gehäuses mit schwarzem Isolierband. Testen Sie die Lichtdichtheit, indem Sie das geschlossene Gehäuse (in Status-3) direkter starker Sonneneinstrahlung aussetzen. Jetzt kann Bob den **DE**-Befehl eingeben, um zu erfahren, was die Löschroutine ausgelöst hat:

```
Box zeroized. Status-6. Tampering
status display!
Capacitor Cx1 voltage ratio: 0.683
Capacitor Cx2 voltage ratio: 0.656
Phototransistor voltage [V]: 0.188
Box temperature [degC]: +10
Battery voltage [V]: 2.775
Min temperature [degC]: -5
Max temperature [degC]: +30
Cause of zeroization:
Capacitor Cx2. kCx2= 0.656
```

## Status-6

Nach Erhalt des Befehls **DE** übermittelt das Paket Details zur Ursache des Löschroutines.

Die in Status-5 oder Status-6 angezeigten Daten müssen nicht gültig sein, da es davon abhängt, was die bössartige Mallory mit dem Paket angestellt hat. Möglicherweise hat sie es zurückgesetzt und Challenge- und Response-Strings eingegeben, bevor sie das Paket im Status-3 an Bob weitergeschickt hat. Aber selbst wenn sie es irgendwie schafft, die richti-

gen Zeichenfolgen einzugeben, muss sie auch den Transporttimer manipulieren, und das dürfte noch viel schwieriger sein.

## Zurück zu Status-1

Mit dem Befehl **RE** kann das Paket auf Status-1 zurückgesetzt werden, ohne es zu öffnen. Ein Druck auf die Reset-Taste S2 im Inneren des Gehäuses hat die gleichen Folgen.

## Ein Wort über Hardware-Trojaner

Da sich die meisten Menschen normalerweise nur über Software-Malware und Trojaner Sorgen machen, die ihre Computer angreifen, sollen hier ein paar Worte über Hardware-Trojaner gesagt werden. Ein einfaches System mit einem einzigen Controller, das nie mit dem Internet verbunden war und ist, ist hundertprozentig sicher gegen typische Malware, die es auf Ihren PC oder Ihr Smartphone abgesehen hat. Der Artikel in [5] erläutert grundlegend die Probleme im Zusammenhang mit Solid-State-Speichern auf sehr hohem Integrationsniveau.

Ein Hardware-Backdoor-Trojaner besteht aus einer zusätzlichen Schaltung, die dem Siliziumchip des Controllers hinzugefügt wurde. Dies kann so viel Schaden anrichten wie ein Software-Trojaner oder sogar noch mehr. Ein solcher Trojaner stellt eine große Gefahr für die heutigen Mikrocontroller dar, deren CPU, SRAM, EEPROM und Flash-Speicher auf einem Siliziumchip integriert sind und die mit einer niedri-

gen Spannung (3...5 V) programmierbar sind. Grundsätzlich gibt es zwei Typen dieser Trojaner:

## 1. Hardware-Trojaner für allgemeine Zwecke

Mallory kann einen Hardware-Trojaner in den Controller einschleusen, um zum Beispiel regelmäßig das SRAM und die Controllerregister in einen geheimen Teil des Flash-Speichers zu kopieren. Unabhängig von der Anwendung dürfte ein „eingefrorener“ Schnappschuss des RAMs für sie und für Eve immer nützlich sein. Mit einem solchen Trojaner könnte Mallory die von Alice definierten Challenge- und Response-Strings leicht lesen. Das Spoofing des Reisezeit-Timers dürfte jedoch etwas schwieriger sein.

## 2. Anwendungsspezifische Hardware-Trojaner

Um diese Art von Hardware-Trojaner zu installieren, muss Mallory Kenntnisse über die Firmware des Controllers besitzen. Dann stattet sie den Controller mit einer Schaltung aus, die in der Lage ist, die Firmware zu erkennen, die Alice in ihren Controller geflasht hat. Beim Start des Controllers fungiert der Trojaner als Bootloader und ändert einige Zeilen im Programm (er füllt sie beispielsweise über den berechtigten SPM-Befehl des AVR mit NOPs), um einige wichtige Schutzmaßnahmen zu deaktivieren. Die Erfolgsaussichten eines Angriffs dieser Art sind allerdings viel geringer als bei einem Allzweck-Trojaner.



## STÜCKLISTE

### IR-Adapter

#### Widerstände:

(alle SMD 0805)  
R1,R6 = 680 Ω  
R4 = 1k5  
R2,R8 = 4k7  
R5 = 22 k  
R7 = 100 k  
R3 = 2M2

#### Kondensatoren:

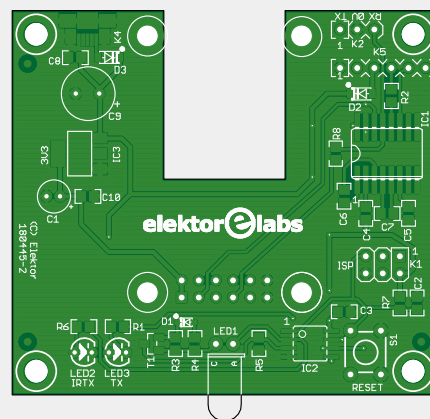
(alle SMD 0805)  
C2,C3,C8 = 100 n  
C4,C5,C6,C7,C10 = 1 µ, X7R, 50 V  
C1 = 10 µ, 10 V, Tantal, Raster 2 mm  
C9 = 100 µ, 16 V, Raster 3,5 mm

#### Halbleiter:

D1 = 1N4148WS (SOD-323)  
D2,D3 = MBR120  
IC1 = MAX232ECWE  
IC2 = ATtiny45-20SU, programmiert  
IC3 = LD1117AS33  
LED1 = TSAL6200  
LED3 = 3 mm, gelb  
LED2 = 3 mm, rot  
T1 = BC857C

#### Außerdem:

K1 = 2x3-polige Stiftleiste, Raster 0,1"  
K2 = 1x3-polige Stiftleiste, Raster 0,1"  
K3 = 2x6-polige Buchsenleiste, Raster 0,1"  
K4 = Micro-USB-Verbinder Typ B



K5 = 6-polige Stiftleiste, Raster 0,1"  
S1 = Tastschalter, 6x6 mm<sup>2</sup>  
WIZ107 Ethernet-Modul  
Platine 180445-2



## Gibt es denn gar keine Verteidigung?

Zum Glück für Alice und Bob gibt es viele Möglichkeiten, dieser Art von Hardware-Bedrohungen zu begegnen. Die einfachste ist, zur Technologie der 1980er Jahre zurückzukehren und ein einfaches Z80-System wie [6] mit SRAM und EPROM als getrennte ICs zu verwenden. Das EPROM wird mit UV-Licht gelöscht und benötigt zum Beschreiben eine Spannung von 13...14 V auf seinem  $V_{pp}$ -Pin, was es viel schwieriger macht, eine Hardware-Backdoor zu installieren. Darüber hinaus sind die internen Architekturen der Z80- und 27C-EPROMs, die seit über 20 Jahren in vielen Millionen Exemplaren hergestellt werden, sehr gut bekannt, und es wäre einfach, ein IC zu entkapseln und bei Bedarf unter einem Mikroskop zu inspizieren. Da es Millionen dieser Bauteile auf der ganzen Welt gibt, wäre es für Mallory schier unmöglich, sie alle zu manipulieren.

Die Portierung der Controller-Firmware des Zufallszahlen-Generators oder des OTP-Krypto-Shields auf einen Z80 kann, auch wenn die Verarbeitungsgeschwindigkeit sinken und ein solches Gerät physisch größer sein würde, gelingen und die Systeme würden gut funktionieren. Moderne Geräte in Small-, Mini-, Micro- und Nanoausführung müssen nicht immer besser sein!

Natürlich wäre ein Z80-basiertes System viel zu unhandlich und stromfressend, um auf den Postversand zu gehen. Aber dieses Problem kann auf andere Weise gelöst werden, mit einer kleinen Zusatzschaltung, die in das vorhandene Gehäuse eingebaut wird. Wenn Sie die Schaltpläne sorgfältig studiert haben, fragen Sie sich vielleicht nach der Funktion von K3. Dieser Anschluss ist für eine Zusatzschaltung vorgesehen, die das Paket vor Hardware-Trojanern schützt.

Mehr dazu in einem zukünftigen Artikel in Elektor, also bleiben Sie dran!

## Gute Aussichten!

Der auf den Fotos gezeigte Prototyp des Pakets hat mehrere (Land-) Transporte mit verschiedenen Post/Paketdiensten unternommen und sich dabei als gut funktionierend erwiesen. Einige dieser Paketdienste gehen wohl recht „rau“ mit der Post ihrer Kunden um, so dass es notwendig war, C1x und C2x besser zu befestigen und zu verhindern, dass sich die Platten lösen. Eine Möglichkeit ist das Aufkleben von (nichtleitendem) Schaumstoff auf den Platten oder auf der Innenseite des Gehäuses. Soll der Controller in einer Fassung montiert werden, sollte man ihn ebenfalls sichern, zum Beispiel mit einem kleinen Kabelbinde unter der Fassung. Und die Batterie sitzt zwar ziemlich fest in ihrer Halterung, aber man weiß ja nie...

Wenn Sie weitere Vorschläge zur Verbesserung des Manipulationssicheren Datenpakets oder des IR-Adapters haben, insbesondere bezüglich der Verteidigung gegen Hardware-Backdoors, kontaktieren Sie mich bitte über die Elektor-Labs-Projektseite [7]. Ich habe zwar schon ein paar Ideen, aber vielleicht haben Sie bessere, und der Vorschlag alternativer Methoden kann nie schaden. Um seine



Am Ende treffen sich Bob und Alice von Angesicht zu Angesicht.

Sicherheit zu bestätigen, muss nämlich jedes Kryptogerät von einer Reihe unabhängiger Experten überprüft werden. Also zögern Sie bitte nicht, sich in irgendeiner Weise daran zu beteiligen! Schaltpläne, Platinenlayouts und der Quellcode für dieses Projekt können von der Labs-Webseite des Projekts [7] heruntergeladen werden. ◀

180445-03



### IM ELEKTOR-STORE

→ Manipulationssicheres Datenpaket – alle Leerplatinen  
[www.elektor.de/180445-1](http://www.elektor.de/180445-1)

→ Manipulationssicheres Datenpaket - programmierter Mikrocontroller ATmega328P-PU  
[www.elektor.de/180445-51](http://www.elektor.de/180445-51)

→ IR-Adapter - programmierter Mikrocontroller ATtiny45-20SU  
[www.elektor.de/180445-42](http://www.elektor.de/180445-42)

## Weblinks

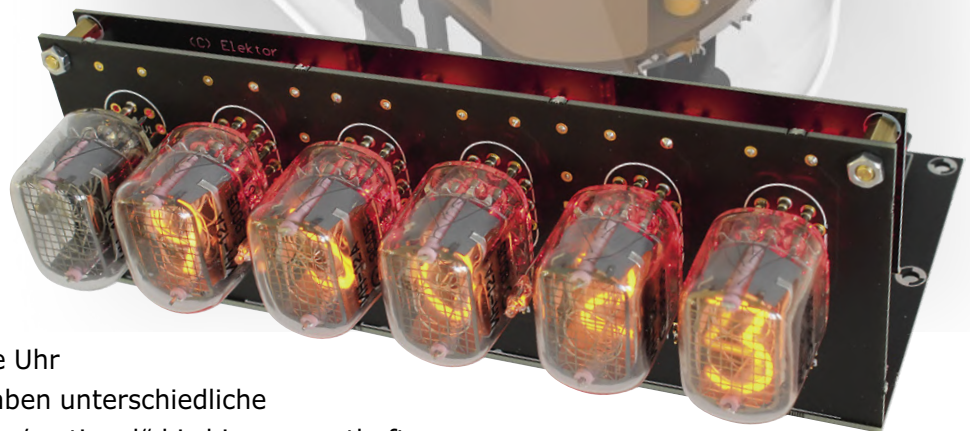
- [1] Zufallszahlen-Generator: [www.elektormagazine.de/magazine/elektor-201703/40169/](http://www.elektormagazine.de/magazine/elektor-201703/40169/)
- [2] OTP-Krypto-System: [www.elektormagazine.de/magazine/elektor-69/42261/](http://www.elektormagazine.de/magazine/elektor-69/42261/)
- [3] Löschen von Daten von Flash-Laufwerken: [www.schneier.com/blog/archives/2011/03/erasing\\_data\\_fr.html](http://www.schneier.com/blog/archives/2011/03/erasing_data_fr.html)
- [4] Daten-Remanenz in Halbleiterbauteilen: [www.usenix.org/legacy/events/sec01/full\\_papers/gutmann/gutmann.pdf](http://www.usenix.org/legacy/events/sec01/full_papers/gutmann/gutmann.pdf)
- [5] Secure Communications (Elektor-Video):  
[www.elektormagazine.com/news/secure-communications-an-interview-with-luka-matic](http://www.elektormagazine.com/news/secure-communications-an-interview-with-luka-matic)
- [6] Ein kleiner und einfacher Z80-Computer: [www.sunrise-ev.com/z80.htm](http://www.sunrise-ev.com/z80.htm)
- [7] Projektseite auf Elektor-Labs:  
[www.elektormagazine.com/labs/tamper-evident-paper-mail-box-for-secure-distribution-of-one-time-pads](http://www.elektormagazine.com/labs/tamper-evident-paper-mail-box-for-secure-distribution-of-one-time-pads)

# Reloaded: Neue Nixie-Uhr mit GPS

## Ein Engineering-Bulletin für begeisterte Nutzer

Von **Andy Leitch** (Großbritannien) und **Luc Lemmens** (Elektor-Labor)

Die Neue Nixie-Uhr mit GPS gehörte zu den beliebtesten Projekten, die in den letzten fünf Jahren veröffentlicht wurden, sowohl in Bezug auf die Resonanz der Leser als auch auf die Verkäufe der Bausätze. Wir wollten uns aber nicht auf unseren Lorbeeren ausruhen, sondern eine Reihe von Neuerungen und Verbesserungen für die Uhr zusammenstellen. Die Modifikationen haben unterschiedliche Tragweite und reichen von „nice to know / optional“ bis hin zu ernsthaften Ratschlägen zur Verlängerung der Lebensdauer Ihrer Nixie-Röhren. Obwohl die Modifikationen zu einer Aktualisierung der Platine und zu einer völlig neuen Hintergrundbeleuchtung geführt haben, ist es sehr gut möglich, Ihre „alte“ Platine anzupassen und auch die neue Software zu verwenden.



### INFOS ZUM PROJEKT



Nixie  
PIC  
Uhr Update



Einsteiger  
→ Fortgeschrittene  
Experte



ca. 1 Stunde (nur Upgrade/Modifikationen); 3 Stunden (Montage des Semi-Kits)



Laborwerkzeuge,  
PIC-Programmer



ca. 10 € (nur Upgrade/Modifikationen); ca. 140 € (Semi-Kit ohne Gehäuse)

Im Mai 2016 veröffentlichte Elektor die „Neue Nixie-Uhr mit GPS“ [1], und seitdem haben viele Leser entweder den kompletten Bausatz oder auch nur die nackten Platinen gekauft, um ihre eigenen Bauteile zu verwenden. Tatsächlich ist die Uhr, wenn man die Anzahl der Views und der technisch interessierten Follower auf der Elektor-Labs-Website [2] betrachtet, eine der beliebtesten Schaltungen der letzten Jahre.

Nach der Veröffentlichung im Mai 2016 erhielten wir viele Anfragen und Vorschläge zur Verbesserung und Erweiterung der Uhr, insbesondere der Software. Diese Aufgabe wurde schließlich von Andy Leitch, einem der treuesten Anhänger von Elektor Labs übernommen. Andy hat die ursprünglich in Assembler erstellte Software komplett in „C“ umgeschrieben und die Hardware so angepasst, dass die Uhr unter anderem

auch das Datum anzeigen und als Wecker fungieren kann. Um die Möglichkeiten der neuen Firmware [3] voll ausschöpfen zu können, sind kleinere Anpassungen der Platinen erforderlich. Wenn Sie jedoch damit zufrieden sind, nur die Datumsanzeige hinzuzufügen und das „Cathode Poisoning“ (dazu später mehr) zu verhindern/begrenzen, dann genügt es, den Mikrocontroller zu ersetzen beziehungsweise ihn neu zu programmieren.

Die augenfälligste Verbesserung, die Andy vorgenommen hat, ist die mehrfarbige Hintergrundbeleuchtung für die Nixie-Röhren, die die sechs blauen LEDs aus dem Bausatz ersetzt. Die neue Hintergrundbeleuchtung ist nicht nur dekorativ, sondern zeigt auch an, welche Informationen aktuell auf dem Display der Uhr angezeigt werden. Andy verwendete dafür einen Streifen Lochrasterplatine und bedrahtete WS2812-RGB-

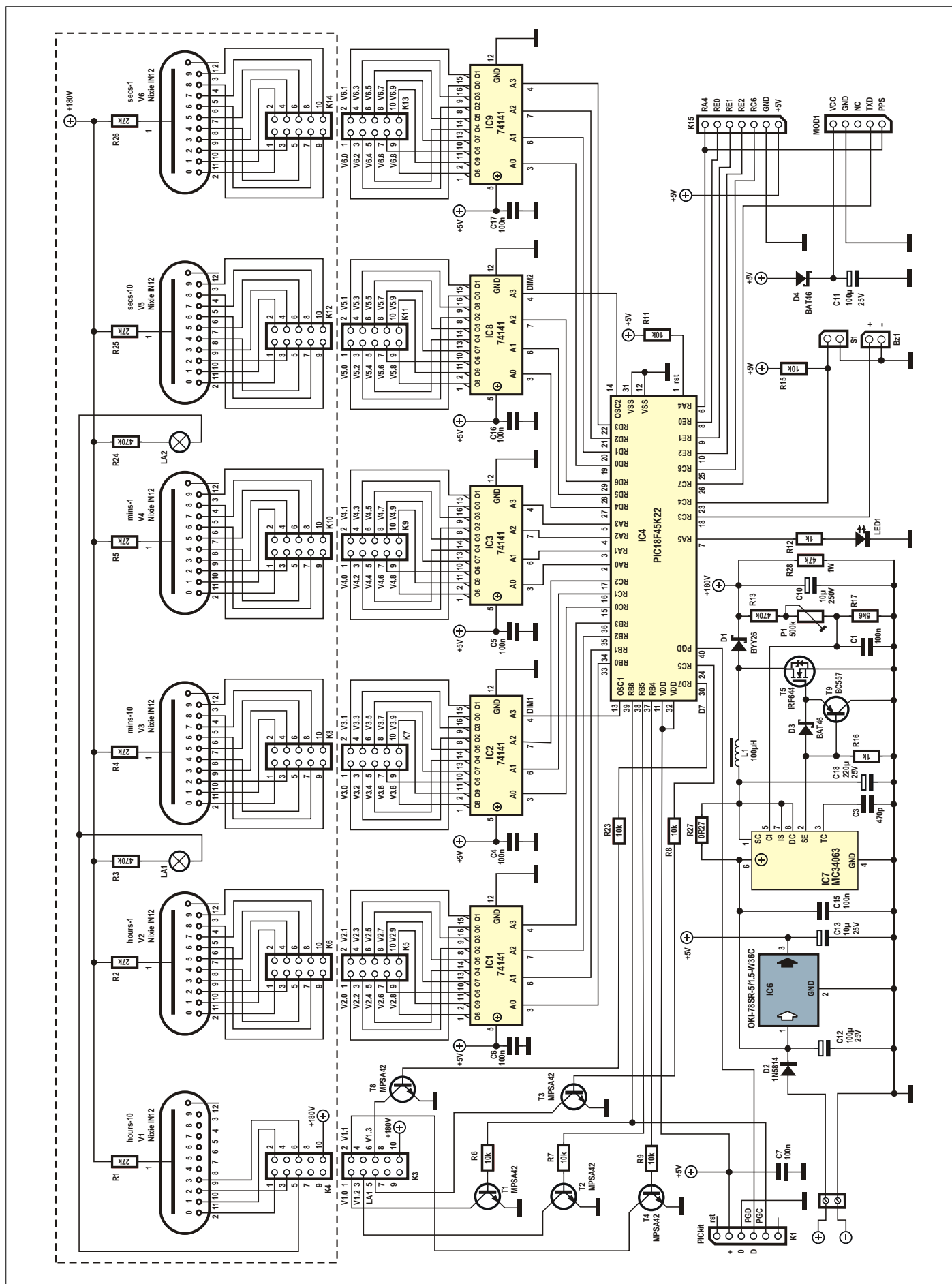


Bild 1. Schaltbild der überarbeiteten Neuen Nixie-Uhr mit GPS mit allen Änderungen gegenüber dem ursprünglichen Entwurf von 2016.



LEDs, aber der Ästhetik und einer leichten Montage willen hat das Elektor-Labor eine Platine für die gleichen LEDs in SMD-Gehäusen entworfen. Diese wie alle anderen Platinen des Projekts schwarze Platine ist im Elektor-Store erhältlich. Mehr dazu weiter unten.

### Neue Funktionen, neuer Schaltplan

Die Hardware und einige der Software-Modifikationen der Neuen Nixie-Uhr mit GPS haben zu einem neuen Schaltplan geführt, der in **Bild 1** zu sehen ist. Dieser Schaltplan entspricht der neuen Platine, die ab sofort in den Bausätzen der Neuen Nixie-Uhr aus dem Elektor-Store enthalten ist. Die folgende Anleitung ist für die alte Platine gedacht; die neue Version V2.1 benötigt keine Hardware-Modifikationen.

### Modifikation der Tasten

**Status: obligatorisch.** Der vorhandene Kippschalter S1 wird durch einen Taster ersetzt. Diese sehr einfache Modifikation erlaubt den Zugriff auf die meisten der Funktionen der neuen Software. Dazu wird einfach der vorhandene Schalter abgelötet und durch einen Drucktaster ersetzt. Möglicherweise müssen Sie das Befestigungsloch passend zum Ersatzschalter erweitern, aber wenn Sie einen Taster mit einem 4,95-mm-Schaft verwenden, passt er ohne Bohren.



Bild 4. Verbindung des „3er-Elements“ der Nixie-Röhre V1 mit dem mittleren Pin auf der Oberseite von K4 (nur bei bestehenden Platinen).



Bild 2. Die Leiterbahn muss (nur bei bestehenden Platinen) durchtrennt werden, um ein Neonlämpchen von seinem Treiber zu trennen.

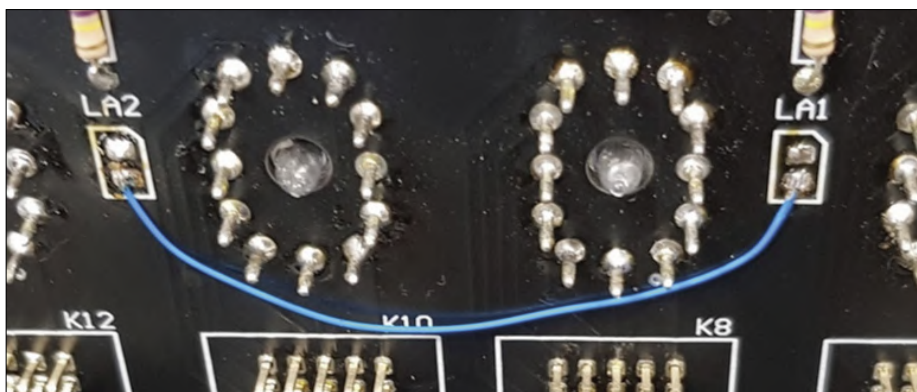


Bild 3. Verbindung der Neonlampen mit einem Kabel (nur bei bestehenden Platinen).

### Verhinderung des Cathode Poisoning

**Status: sehr empfohlen.** Wenn eine Nixie-Röhre betrieben wird, ohne dass regelmäßig alle Zahlen angesteuert werden, neigen die unbenutzten Kathoden dazu, von einem hochohmigen Material überzogen zu werden. Dieser Effekt wird allgemein als „Cathode Poisoning“ bezeichnet. Er führt zu dunklen Flecken in den Segmenten einer Ziffer, wenn diese vollständig beleuchtet werden soll. Bei Nixies einer Uhr ist ein Cathode Poisoning der Sekundenröhre zwar unwahrscheinlich, da in zehn Sekunden alle Ziffern durchlaufen werden, aber andere Röhren sind in Gefahr. Ein Cathode Poisoning kann weitgehend verhindert (oder zumindest begrenzt) werden, indem jede Röhre ab und zu alle Ziffern durchläuft, und genau dafür sorgt die neue Software. Die erste „Reinigung“ erfolgt beim Einschalten und wird alle 30 Minuten wiederholt (um Viertel vor und Viertel nach). Falls eingebaut, blinkt die neue WS2812-Hintergrundbeleuchtung und durchläuft das Farbspektrum, während

dieser Reinigungsvorgang durchgeführt wird. Diese Option kann in den Einstellungen der Uhr abgeschaltet werden.

### Änderung des Alarmsignals

**Status: sehr empfohlen.** Obwohl im Betrieb eigentlich nur bei der Weckerfunktion erforderlich, ist ein Signalton bei der Einstellung der Parameter unerlässlich. Es wird ein Summer mit niedriger Stromaufnahme benötigt, der leicht bei Bauteilhändlern im Netz erhältlich ist. Mit einer Suche nach 3–24 V *piezo buzzer* sollten Sie einen geeigneten Kandidaten aufstöbern können. Achten Sie auf einen Versorgungsstrom von 10 mA oder weniger, da der PIC nur 25 mA aus dem Port-Pin liefern kann. Der Summer ist gepolt und wird mit seinem schwarzen Draht an Pin 1 von JP1 am Platinenrand und mit dem roten Draht an den Pin auf der PIC-Seite angeschlossen. R14 kann ausgelötet oder von der Platine abgeknipst werden. Der Schall des eingeklebten Summers kann durch ein kleines Loch in der Gehäuserückseite in die Außenwelt entweichen.

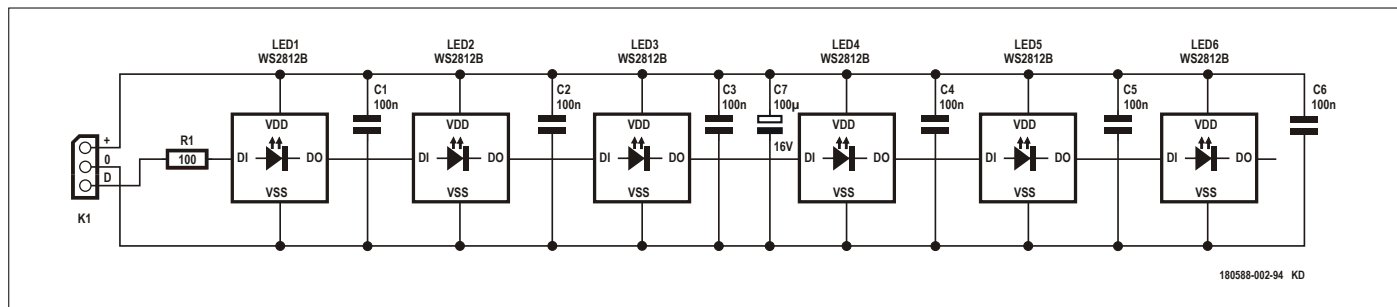


Bild 5. Schaltplan der dekorativen Hintergrundbeleuchtung, die auf der Platine mit SMD-LEDs realisiert wurde. Man beachte die Entkopplungskondensatoren, die das durch den relativ hohen Strom verursachte Rauschen unterdrücken.

## Änderung der Datumsanzeige

**Status: optional.** Diese Änderung ist erforderlich, um das Datum im EU-Format (tt/mm/jj) darstellen zu können, da eine „3“ in den äußersten linken Stelle der Tagesanzeige benötigt wird. Die alte Platine ist so ausgelegt, dass die beiden Neons separat angesteuert werden und die Ziffer „3“ der linken Nixie-Röhre nicht angeschlossen ist. Das Problem wird so gelöst, dass beide Neonlämpchen an einen Treiber angeschlossen werden und der nun freie Treiber für die „3“ verwendet wird.

**Schritt 1:** Trennen Sie ein Neonlämpchen von seinem Treiber. Die untere Leiterbahn auf der Röhrenseite der Platine wird mit einem Cutter im Abstand von 1 mm zweimal durchtrennt und der kleine Kupferstreifen dazwischen abgezogen (**Bild 2**).

**Schritt 2:** Verbinden Sie die Neonlampen miteinander, siehe **Bild 3**.

**Schritt 3:** Verbinden Sie die Ziffer „3“

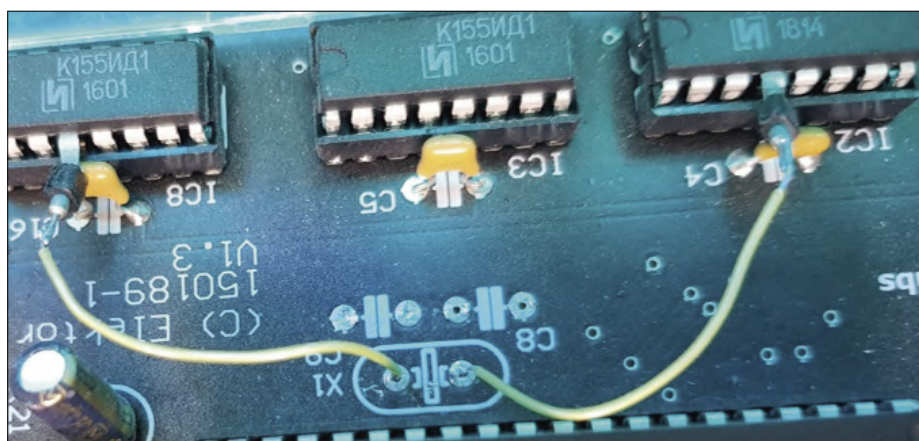


Bild 6. Die freien Quarzanschlüsse auf der Platine sind mit den seitlich abgebogenen Pins der Treiber-ICs verbunden (nur bei bestehenden Platinen).

zum mittleren Stift von K4 (obere Reihe) (**Bild 4**).

## Neopixel-LED-Hintergrundbeleuchtung

**Status: optional.** Vorsicht: Die LEDs benötigen zwar nur 5 V, aber ziemlich

viel Strom. Sie können einen zweiten 5-V-Regler verwenden, der von der 9-V-Eingangsspannung versorgt wird, oder Sie können Strom, Masse und Daten vom Programmier-Header beziehen. In diesem Fall benötigen Sie einen Kühlkörper auf IC6 oder noch besser, Sie erset-



## STÜCKLISTE

Hintergrundbeleuchtung (180588-1)

### Widerstand:

R1 = 100 Ω, Dickschicht, 5%, 0,1 W, 150 V, 0805

### Kondensatoren:

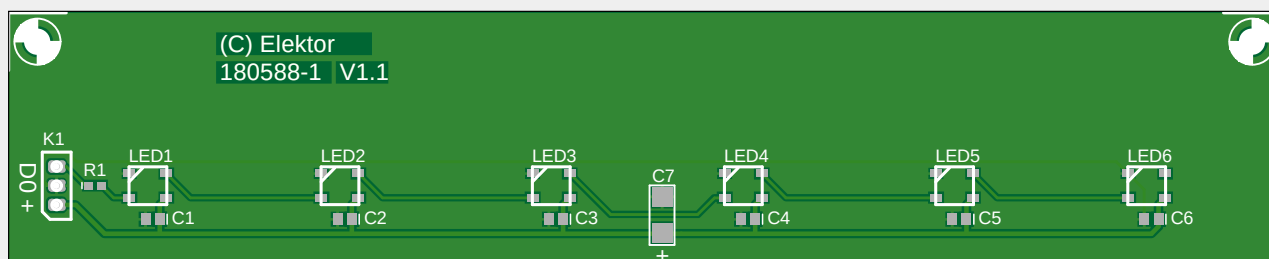
C1...C6 = 100 n, 50 V, X7R, 0805  
C7 = 100 µ, 16 V Tantal, 10%, 2312

### Halbleiter:

LED1...LED6 = WS2812B RGB-LED (SMD)

### Außerdem:

3-poliges Kabel, Länge ca. 10 cm  
3-poliger SIL-Verbinder  
Platine 180588-1 V1.1





## STÜCKLISTE

### Überarbeitete Nixie-Uhr (150189-1 V2.1)

#### Widerstände:

R1,R2,R4,R5,R25,R26 = 27 k, 5%,  
250 mW  
R6,R7,R8,R9,R11,R15,R23 = 10 k,  
5%, 250 mW  
R3,R13,R24 = 470 k, 5%, 250 mW  
R12,R16 = 1 k, 5%, 250 mW  
R17 = 5k6, 5%, 250 mW  
R27 = 0 Ω27, 5%, 2 W  
R28 = 47 k, 5%, 1 W  
P1 = Trimpoti 500 k

#### Induktivität:

L1 = 100 µH, 1,5 A, radial, Dxl  
8x8 mm

#### Kondensatoren:

C3 = 470 p, 50 V, Y5U  
C1,C4,C5,C6,C7,C15,C16,  
C17 = 100 n, 50 V, 20%  
C10 = 10 µ, 250V, 20%,  
radial, Raster 5 mm  
C11,C12 = 100 µ, 25 V, Raster  
3,5 mm, 8x11 mm  
C13 = 10 µ, 25 V, Raster 2 mm,,  
5x1 mm  
C18 = 220 µ, 25 V, Raster 5 mm,  
10x16 mm

#### Halbleiter:

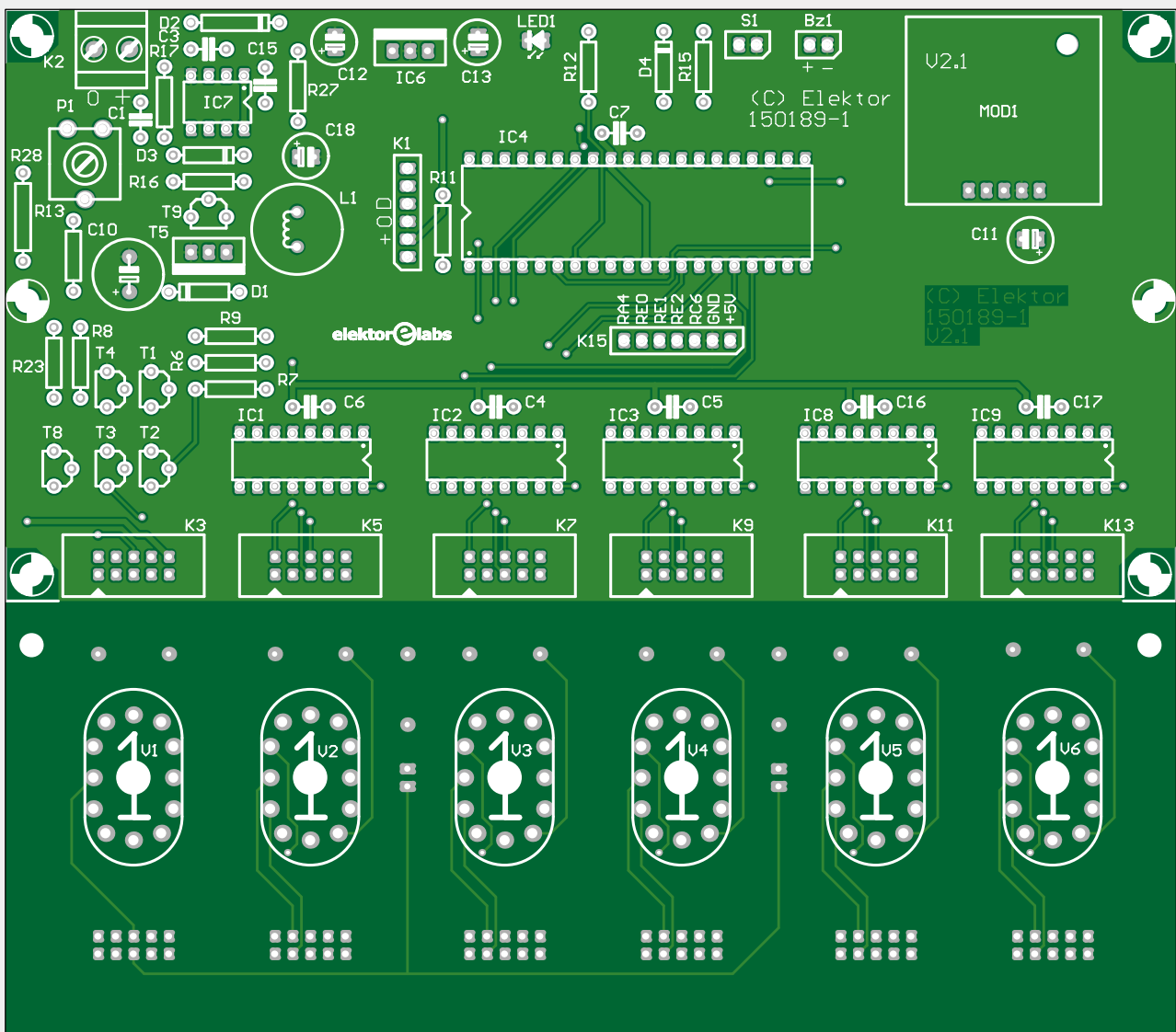
D1 = BYV26, ultraschnelle Diode,  
600 V, 1 A  
D2 = 1N5814, Schottky-Diode,  
40 V, 1 A  
D3,D4 = BAT46, Schottky-Diode  
100 V, 150 mA  
LED1 = rot, 3 mm  
T1,T2,T3,T4,T8 = MPSA42, NPN,  
300 V, 500 mA  
T5 = IRF644PBF, NMOS, 250 V,

14 A  
T9 = BC557C, -45 V, -100 mA,  
500 mW,  $h_{fe} = 400$   
IC1,IC2,IC3,IC8,IC9 = K155ID1  
(74141)  
IC4 = PIC18F45K22,  
programmiert,  
Elektor-Store 180588-41  
IC6 = OKI-78SR-5/1,5-W36-C,  
DC/DC-Wandler 5 V, 1,5 A,  
Murata  
IC7 = MC34063, DC/  
DC-Controller

#### Außerdem:

Bz1 = Summer 1,5...20 V<sub>DC</sub> z.B.  
Loudity LD-BZPG-2312  
S1 = SPST-Drucktaster, 20 V  
(Welle 4,95 mm)  
K1 = 1x6-polige Stiftleiste,  
Raster 0,1"  
K2 = 2-polige  
Platinenanschlussklemme,  
Raster 0,2", 630 V

K3,K5,K7,K9,K11,K13 = 2x5-polige  
Buchsenleiste, abbrechbar,  
vertikal  
K4,K6,K8,K10,K12,K14 = 2x5-  
polige Stiftleiste, abbrechbar,  
gewinkelt  
K1 = 1x6-polige Stiftleiste,  
Raster 0,1"  
LA1,LA2 = T1,1/4, Neonlampe mit  
Drahtanschluss  
MOD1 = GY-NEO6MV2, GPS-  
Modul mit Antenne  
V1,V2,V3,V4,V5,V6 = IN-12  
Nixie-Röhre  
Platine 150189-1 V2.1 aus dem  
Elektor-Store





zen den linearen durch einen Schaltregler wie den Murata OKI-78SR-5/1.5-W36-C. Die Schaltung der bunten Hintergrundbeleuchtung ist in **Bild 5** zu sehen. Pin RB7 des Controllers wird in dem Projekt außer für die Programmierung nicht verwendet. Daher wurde ein kleiner Codeabschnitt hinzugefügt, der sich um die Ansteuerung der Hintergrund-LEDs über diesen Pin kümmert. Dieses Signal ist auch an der Stiftleiste K1, Pin 4, zu finden.

Für die SMD-LEDs vom Typ WS2812 hat das Labor eine Platine entworfen, auf der ein 100- $\mu$ -Elko im SMD-Format sowie 100-nF-Kondensatoren an den Versorgungspins jeder LED geschaltet sind. Vor der ersten LED ist ein 100- $\Omega$ -Widerstand in die Datenverbindung von RB7 eingeschleift. Alternativ können Sie natürlich auch eine eigene Hintergrundbeleuchtung mit bedrahteten WS2812 auf einer Lochrasterplatine aufbauen. Wir empfehlen aber die SMD-Platine, da diese Art von LEDs leichter auszurichten sind und (weil die Bauteile flacher sind) die Platine ohne mechanische Probleme in das Acrylgehäuse passt.

## Dimmen

**Status: empfohlen.** Die 74141-kompatiblen Dekoder zeigen bei Eingabewerten über Neun nichts mehr an. Damit alle Ziffern blinken und gedimmt werden können, müssen bei zwei Chips die Bit-3-Pins (also Pin 4) mit dem Controller statt mit Masse verbunden werden.

Wenn das Dimmen in den Einstellungen aktiviert ist, ohne dass diese Modifikation vorgenommen wurde, zeigen die Digits die Ziffer „7“ an, anstatt abzuschalten. Der Quarz muss ausgelötet werden. Die Quarzanschlüsse werden dazu einer nach dem anderen abwechselnd erhitzt. Dann sollte man am Quarz wackeln, bis er vorsichtig herausgehoben werden kann. Die beiden Kondensatoren C8 und C9 werden ebenfalls entfernt.

Zwei der Dekoder-Chips (IC2 und IC8) werden aus den Fassungen gezogen, die Pins 4 vorsichtig waagerecht ausgerichtet und dann wieder eingesetzt. Achten Sie darauf, dass die Beine beim Wiedereinsetzen nicht umknicken! Verwenden Sie eine alte, gedrehte IC-Fassung oder Jumperkabel mit weiblichem Ende zum Anschluss. **Löten Sie nicht**, denn es könnten Leckströme auftreten, die beim Dimmen sichtbar würden! **Bild 6** zeigt, wie die IC-Pins mit den leeren Bohrungen für den Quarz verbunden werden.

## GPS-Modul

**Status: empfohlen.** In den vergangenen Monaten erhielt unser Labor Mitteilungen von Lesern, die die Nixie-Uhr bauen und die Bauteile separat kaufen wollten, aber keine Bezugsquelle für das GPS-Modul A2035H von Maestro Wireless Solutions gefunden haben. Es stellte sich heraus, dass dieses Teil nicht mehr produziert wird. Glücklicherweise hat Maestro jetzt das A2235H im Angebot, ein neueres GPS-Modul, das auf die alte Platine passt (obwohl das Modul kleiner ist als sein Vorgänger). Es benötigt jedoch zwei zusätzliche 2,2-k $\Omega$ -Pullup-Widerstände von den I<sup>2</sup>C-Pins zum 1,8-V-Ausgang des Moduls. Möglicherweise benötigen Sie auch eine (hitzebeständige) Kapton-Isolierung zwischen der Leiterplatte und dem Modul, um einen Kurzschluss der Abschirmung des Moduls mit einer darunter liegenden Durchkontaktierung zu vermeiden [2].

Beide Maestro-GPS-Module haben standardmäßig einen seriellen Ausgang von 4800 Baud. Es gibt auch viele andere Module, die aber normalerweise mit 9600 Baud kommunizieren. Glücklicherweise verfügt die neue Software der Uhr über eine automatische Baudratenerkennung für das GPS-Modul.

## Die Bedienungsanleitung

Die Uhr lässt sich mit der neuen Software durch einen einzigen Taster steuern. Kurzes Drücken ändert die Anzeige oder

den Parameter; langes Drücken ändert den Modus entsprechend der aktuellen Anzeige. Alle Optionen, Einstellungen und Parameter, die zur Konfiguration und Einstellung der überarbeiteten Neuen Nixie-Uhr zur Verfügung stehen, sind in der von Andy erstellten und unter [2] gespeicherten Bedienungsanleitung zu finden.

## Fazit

Obwohl die originale Neue Nixie-Uhr mit GPS mit der ebenso originalen Software über jeden Zweifel erhaben ist, bietet die hier vorgestellte noch neuere Version einige zusätzliche Funktionen, die ein Update wert sind. Wenn Sie alle Hardware-Änderungen vornehmen sollten, nun... das liegt natürlich an Ihnen. Wenn Sie nur einige auswählen, gut! Die Autoren glauben, dass der verbesserte GPS-Empfang, das Anti-Cathode-Poisoning und die mehrfarbige Hintergrundbeleuchtung absolute Verbesserungen für diese schöne und moderne nostalgische Uhr sind! ◀

180588-04



### IM ELEKTOR-STORE

- Neue Nixie-Uhr mit GPS, Platine (neue Version), unbestückt, 150189-1 V2.1  
[www.elektor.de/150189-1](http://www.elektor.de/150189-1)
- Neue Nixie-Uhr mit GPS, Platine Hintergrundbeleuchtung, unbestückt, 180588-1 V1.1  
[www.elektor.de/180588-1](http://www.elektor.de/180588-1)
- Programmierter Controller PIC18F45K22 (DIP-40), 180588-41  
[www.elektor.de/180588-41](http://www.elektor.de/180588-41)
- GPS-Modul  
[www.elektor.de/open-smart-gps](http://www.elektor.de/open-smart-gps)

## Weblinks

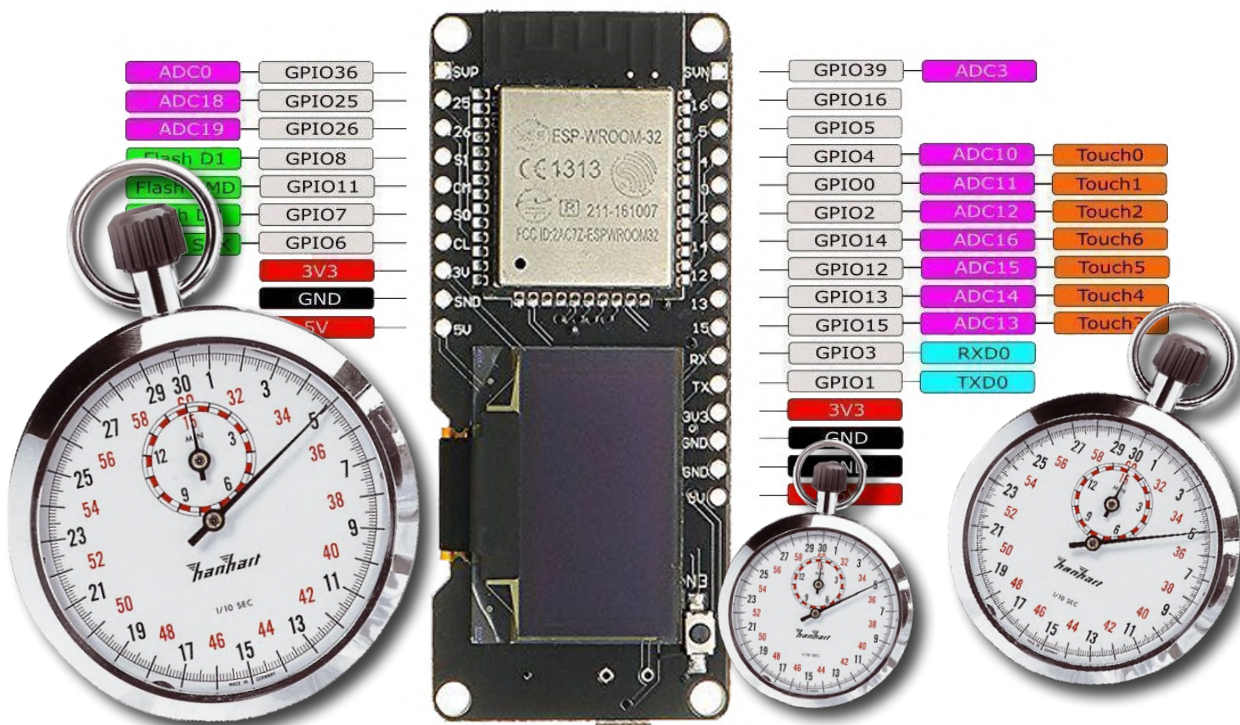
- [1] Neue Nixie-Uhr mit GPS, Elektor Mai 2016:  
[www.elektormagazine.de/magazine/elektor-201605/28919/](http://www.elektormagazine.de/magazine/elektor-201605/28919/)
- [2] Projektseite auf Elektor-Labs:  
[www.elektormagazine.com/labs/150189-6-digit-nixie-clock](http://www.elektormagazine.com/labs/150189-6-digit-nixie-clock)
- [3] Bedienungsanleitung und Firmware: [www.elektormagazine.de/180588-04](http://www.elektormagazine.de/180588-04)

# Praktisches ESP32-Multitasking

## Teil 3: Software-Timer

Von **Warren Gay** (Kanada)

Mikrocontroller-Entwickler benötigen oft Timer, um wiederholte Übertragungen von Daten und andere Verfahren zur Fehlerbehebung durchzuführen, um Tasten zu entprellen oder einfach eine LED blinken zu lassen. Das ESP32-Modul und die Arduino-IDE bieten Software-Timer, die die eingebundene FreeRTOS-Bibliothek verwenden. In diesem dritten Teil der Artikelreihe konzentrieren wir uns auf die Einrichtung dieser Timer.



Um zu veranschaulichen, wie FreeRTOS-Timer funktionieren, werden wir beispielhaft eine Alarm-LED-Anwendung konstruieren. Dabei soll es diesmal nicht einfach nur die permanent blinkende LED sein, sondern eine LED, die schnell fünf Mal blinkt und für den Rest der Periodenzeit dunkel bleibt, bis sich der Vorgang wiederholt. Ein solches Blinken erregt viel mehr Aufmerksamkeit.

Die `AlertLED`-Klasse verwendet intern einen FreeRTOS-Software-Timer, um eine LED zu steuern. Sie können so viele Instanzen dieser Klasse hinzufügen, wie Sie benötigen, ohne den Speicher des ESP32 bis an seine Grenzen zu belasten. **Listing 1** zeigt das vollständige Programm [2], wobei in Zeile 100 eine `AlertLED`-Instanz mit einer einfachen C++-Deklaration erstellt wird:

```
static AlertLED alert1(GPIO_LED,1000);
```

Der Name der Klasseninstanz ist hier `alert1`, wobei die LED (GPIO 12 in Zeile 5) und eine Blinkdauer von 1000 Millisekunden spezifiziert wird. Wir könnten leicht eine weitere LED an GPIO 13 blinken lassen, indem wir einfach die Zeile hinzufügen:

```
static AlertLED alert2(13,1000);
```

Unser kleines Demoprogramm beschränkt sich zwar auf nur eine LED, aber Sie sehen, wie leicht und bequem eine oder

mehrere C++-Klasseninstanzen hinzugefügt werden können.

### Die Klasse `AlertLED`

Die Klassendefinition ist der Bequemlichkeit vollständig im Artikel abgedruckt. Unter den Datenelementen (Data Members) der Klasse befindet sich Member `thandle` (Zeile 11), ein Handle für den FreeRTOS-Timer (Typ `TimerHandle_t`). Zunächst wird ihm der Wert `nullptr` zugewiesen (in C wäre dies `NULL`). Das Datenelement `state` verfolgt den Zustand der LED (Zeile 12), der aktiv high (`true` = on) ist. Das Datenelement `count` (Zeile 13) wird im Callback als weitere Zustandsvariable verwendet (dazu später mehr). Schließlich konfigurieren die Elemente `period_ms` und `gpio` die Blinkperiode in Millisekunden und den GPIO zur Ansteuerung der LED (Zeilen 14 und 15).

```
0010: class AlertLED {
0011:     TimerHandle_t    thandle = nullptr;
0012:     volatile bool     state;
0013:     volatile unsigned count;
0014:     unsigned          period_ms;
0015:     int               gpio;
```

**Listing 1. Das Programm alertled.ino [2].**

```
0001: // alertled.ino
0002: // MIT License (see file LICENSE)
0003:
0004: // LED is active high
0005: #define GPIO_LED      12
0006:
0007: //
0008: // AlertLED class to drive LED
0009: //
0010: class AlertLED {
0011:     TimerHandle_t      thandle = nullptr;
0012:     volatile bool      state;
0013:     volatile unsigned   count;
0014:     unsigned            period_ms;
0015:     int                 gpio;
0016:
0017:     void reset(bool s);
0018:
0019: public:
0020:     AlertLED(int gpio,unsigned period_ms=1000);
0021:     void alert();
0022:     void cancel();
0023:
0024:     static void callback(TimerHandle_t th);
0025: };
0026:
0027: //
0028: // Constructor:
0029: //  gpio      GPIO pin to drive LED on
0030: //  period_ms  Overall period in ms
0031: //
0032: AlertLED::AlertLED(int gpio,
0033:     unsigned period_ms) {
0034:     this->gpio = gpio;
0035:     this->period_ms = period_ms;
0036:     pinMode(this->gpio,OUTPUT);
0037:     digitalWrite(this->gpio,LOW);
0038: }
0039: //
0040: // Internal method to reset values
0041: //
0042: void AlertLED::reset(bool s) {
0043:     state = s;
0044:     count = 0;
0045:     digitalWrite(this->gpio,s?HIGH:LOW);
0046: }
0047:
0048: //
0049: // Method to start the alert:
0050: //
0051: void AlertLED::alert() {
0052:
0053:     if ( !thandle ) {
0054:         thandle = xTimerCreate(
0055:             "alert_tmr",
0056:             pdMS_TO_TICKS(period_ms/20),
0057:             pdTRUE,
0058:             this,
0059:             AlertLED::callback);
0060:         assert(thandle);
0061:     }
0062:     reset(true);
0063:     xTimerStart(thandle,portMAX_DELAY);
0064: }
0065:
0066: //
0067: // Method to stop an alert:
0068: //
0069: void AlertLED::cancel() {
0070:     if ( thandle ) {
0071:         xTimerStop(thandle,portMAX_DELAY);
0072:         digitalWrite(gpio,LOW);
0073:     }
0074: }
0075:
0076: // static method, acting as the
0077: // timer callback:
0078: //
0079: void AlertLED::callback(TimerHandle_t th) {
0080:     AlertLED *obj = (AlertLED*)
0081:         pvTimerGetTimerID(th);
0082:     assert(obj->thandle == th);
0083:     obj->state ^= true;
0084:     digitalWrite(obj->gpio,obj->state?HIGH:LOW);
0085:
0086:     if ( ++obj->count >= 5 * 2 ) {
0087:         obj->reset(true);
0088:         xTimerChangePeriod(th,pdMS_TO_TICKS
0089:             (obj->period_ms/20),portMAX_DELAY);
0090:     } else if ( obj->count == 5 * 2 - 1 ) {
0091:         xTimerChangePeriod(th,
0092:             pdMS_TO_TICKS(obj->period_ms/20+
0093:                 ms/2),
0094:             portMAX_DELAY);
0095:         assert(!obj->state);
0096:     }
0097: }
0098: // Global objects
0099: //
0100: static AlertLED alert1(GPIO_LED,1000);
0101: static unsigned loop_count = 0;
0102:
0103: //
0104: // Initialization:
0105: //
0106: void setup() {
0107:     // delay(2000); // Allow USB to connect
0108:     alert1.alert();
0109: }
0110:
0111: void loop() {
0112:     if ( loop_count >= 70 ) {
0113:         alert1.alert();
0114:         loop_count = 0;
0115:     }
0116:
0117:     delay(100);
0118:
0119:     if ( ++loop_count >= 50 )
0120:         alert1.cancel();
0121: }
```



```

0016:
0017: void reset(bool s);
0018:
0019: public:
0020: AlertLED(int gpio,unsigned period_ms=1000);
0021: void alert();
0022: void cancel();
0023:
0024: static void callback(TimerHandle_t th);
0025: };

```

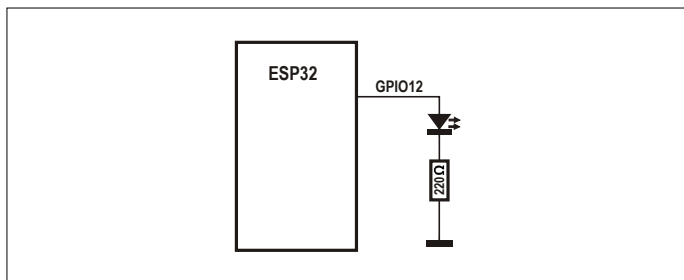


Bild 1. Die Schaltung für das Demonstrationsprogramm *alertled.ino*.

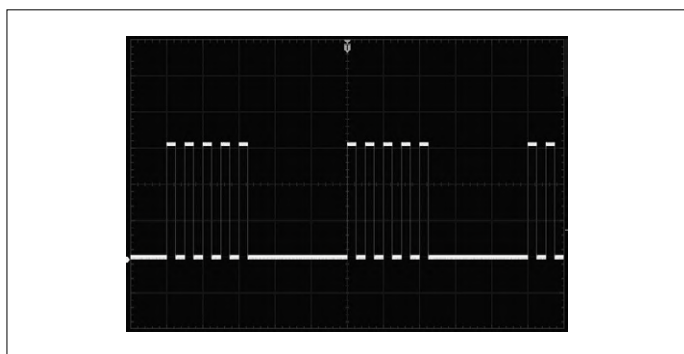


Bild 2. Das von *alertled.ino* erzeugte Steuersignal für die LED (horizontal 200 ms/div).

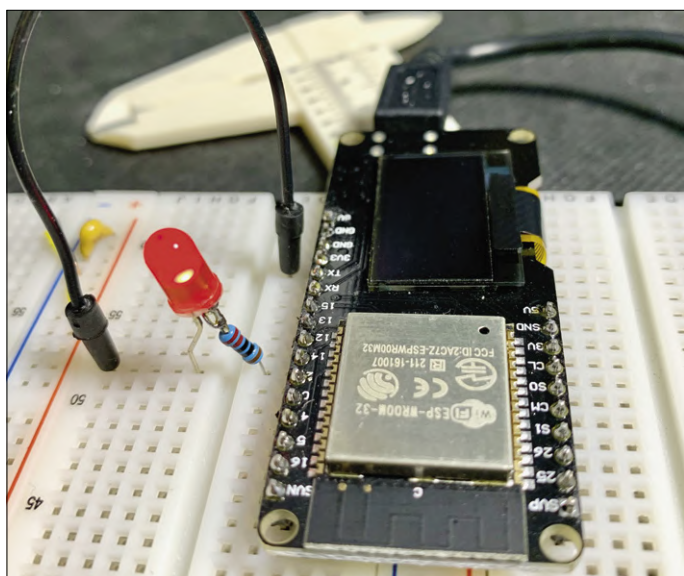


Bild 3. ESP32, LED und ein 220-Ω-Widerstand warten auf die Ausführung des Programms *alertled.ino*.

Die Klasse enthält auch einige öffentliche (public) Aufrufe von Methoden, darunter

- `alert()` - Aktivierung der LED-Anzeige (Zeile 21)
- `cancel()` - Deaktivieren der LED-Anzeige (Zeile 22)

Es gibt auch eine `static`-Methode namens `callback()` in Zeile 24.

### Methode `AlertLED::alert()`

Die Methoden `alert()` und `callback()` sind die interessantesten Teile der Klasse. Wir haben festgestellt, dass das Handle als `nullptr` initialisiert wurde. Wenn die Methode `alert()` zum ersten Mal aufgerufen wird, findet sie den Wert Null in Zeile 53 vor, so dass der Timer erstellt wird. Der Software-Timer wird durch `xTimerCreate()` in den Zeilen 54...59 erstellt, wobei das Handle des neu erstellten Timers zurückgegeben wird. Das `assert()`-Makro in Zeile 60 prüft, ob der Timer erfolgreich erstellt wurde. Die Argumente für `xTimerCreate()` sind:

1. Ein benutzerfreundlicher Stringname für unseren Timer (hier ist es einfach `alert_tmr`). Der Name selbst wird von FreeRTOS nicht verwendet, außer für einige statistische Auswertefunktionen. Er sollte normalerweise eindeutig sein, muss es aber nicht unbedingt, wenn Sie nicht nach dem Namen suchen.
2. Eine Timer-Periode in System-Ticks (vom Makro `pdMS_TO_TICKS()` in Zeile 56 konvertiert in Millisekunden).
3. Der nächste Parameter zeigt an, ob es sich um einen One-Shot- (Wert `pdFALSE`) oder einen Auto-reload-Timer (Wert `pdTRUE`) handelt. In unserem Beispiel ist der Timer für *auto-reload* konfiguriert (Zeile 57).
4. Dies ist eine „Timer ID“ in der FreeRTOS-Terminologie, weil sie den Timer eindeutig spezifiziert, falls wir mehrere verwenden. Es handelt sich um einen Void-Pointer (ein Pointer, der keinem bestimmten Variablentyp zugeordnet ist). In diesem Beispiel übergeben wir die Adresse des aufrufenden LED-Klassenobjekts, indem wir das in C++ reservierte Schlüsselwort `this` verwenden (Zeile 58).
5. Die Adresse der Callback-Funktion. In Zeile 59 geben wir den Namen der statischen Methode `AlertLED::callback()` an. Diese Funktion wird aufgerufen, wenn der Timer getriggert wird.

Einige Datenelemente werden durch den Aufruf der internen Methode `reset()` in Zeile 62 zurückgesetzt, gefolgt vom Timerstart in Zeile 63: `xTimerStart()`. Alle Timer werden auf „ruhend“ gestellt, bis sie (neu) gestartet werden. Sobald sie gestartet werden, gehen sie in den „laufenden“ Zustand über.

### Statische Methode `AlertLED::callback()`

Die Klasse `AlertLED` definiert einen *statischen* Methodenaufruf namens `callback()`. Für diejenigen, die mit C++ nicht vertraut sind: Statisch bedeutet, dass die Methode keinen Objekt-Pointer („this“-Pointer) besitzt - die Funktion ist der Klasse zugeordnet, nicht einer einzelnen Instanz davon. Sie ähnelt einer C-Funktion, mit Ausnahme des speziellen C++-Namens und der Tatsache, dass sie besonderen Zugriff auf die Interna des Klassenobjekts hat.

Eine Software-Timer-Callback-Funktion muss über ein Argument des Typs `TimerHandle_t` verfügen (Zeile 79). Dies gewährt uns Zugriff auf den Handle des Timers, der die Funktion ausgerufen hat. Aber wir brauchen mehr Informationen für jene Aktion, die nun beim Ablauf des Timers ausgeführt werden soll. In diesem

Fall benötigen wir die Instanz der LED, die wir nun ansteuern wollen. In Zeile 80 ermitteln wir zuerst die TimerID über das Handle des Timers, mit dem Aufruf `pvTimerGetTimerID()`. Und die TimerID enthält ja die Adresse der (LED-)Klasseninstanz (`alert1`). Über die Variable `obj` können wir auf das LED-Objekt und seine Member zugreifen.

In den Zeilen 83 und 84 wird der konfigurierte GPIO-Pin für die LED umgeschaltet. Dann wird der Wert von `count` inkrementiert (Zeile 86), und bis der Zähler 9 erreicht, kehren wir einfach vom Callback zurück. Damit wird das schnelle Blinken in der für die erste Hälfte der Blinkperiode durchgeführt.

Wenn der Zählwert 9 erreicht, werden die Zeilen 90...93 ausgeführt. Die Funktion `xTimerChangePeriod()` ändert den Zeitraum, den der Timer verwendet. Beachten Sie, dass `period_ms/20` zu `period_ms/2` addiert wird, um die letzte „dunkle“ Hälfte des schnellen Blinkens zu kompensieren.

Schließlich werden beim 10. Mal des Callbacks die Zeilen 87 bis 88 ausgeführt. Die Datenelemente werden durch den Aufruf der internen Methode `reset()` (Zeile 87) zurückgesetzt und die Timer-Zeit für das schnelle Blinken in Zeile 88 wiederhergestellt. Da das Datenelement `count` auf Null zurückgesetzt wird, wiederholt sich der gesamte Zyklus. Durch die Auto-reload-Eigenschaft des Timers wird das so lange wiederholt, bis `AlertLED::cancel()` aufgerufen wird.

### Methode `AlertLED::cancel()`

Um das Blinken des Alarms zu stoppen, wird die Methode `AlertLED::cancel()` verwendet. Dazu wird `xTimerStop()` aufgerufen (Zeile 71) und die LED ausgeschaltet, indem der LED-GPIO in Zeile 72 auf Low gesetzt wird.

### Die Notwendigkeit von `volatile`

In den Zeilen 12 und 13 der Klassendefinition haben wir das C-Schlüsselwort `volatile` verwendet. Dadurch wird der Compiler darüber informiert, dass immer dann, wenn er auf die Werte der Datenelemente `state` oder `count` zugreift, direkt in den Speicher der Klasseninstanz (`alert1` in unserer Demo) gehen sollte, anstatt einen Wert zu verwenden, der möglicherweise noch in einem Register existiert. Es gibt zwei Tasks, die mit Ihrer Klasse interagieren:

- die FreeRTOS-Dämon-Klasse (über den Callback)
- Ihr Task, der die Methoden `alert()` oder `cancel()` aufruft.

Wenn diese Werte nicht als `volatile` (flüchtig) markiert wären, könnten ein oder beide Tasks Werte verwenden, die sich noch in einem Register befinden, da dieses Vorgehen den Normalfall darstellt. Die Werte im Speicher können sich jedoch durch die andere Tasks geändert haben. Deshalb sorgt das Keyword `volatile` dafür, dass der Compiler eine solche mögliche Optimierung ignoriert und stattdessen Code erzeugt, der die Werte direkt aus dem Speicher holt.

### Die Demonstration

Um `alertled.ino` zu demonstrieren, kann wieder das ESP32-OLED-Displaymodul verwendet werden. Die „Verkabelung“ ist in **Bild 1** dargestellt.

Um zu beweisen, dass die Klasse `AlertLED` aktiviert und abgebrochen werden kann, enthält die Funktion `loop()` eine Zählvariable namens `loop_count` (Zeile 101). Die Funktion `setup()` aktiviert zunächst die LED, aber wenn der Schleifenzähler den Wert 50 erreicht hat, wird der Alarm in Zeile 120 wieder aufgehoben. Wenn die Schleifenzahl später 70 erreicht, wird der Alarm in Zeile 113 wieder aktiviert und der Zähler zurückgesetzt.

Wenn das Programm auf den ESP32 geflasht ist, sollte die Alarm-LED zunächst blinken, dann für eine Zeit verlöschen. Der Zyklus wiederholt sich, wie das Oszillogramm in **Bild 2** zeigt.

**Bild 3** zeigt den Aufbau auf einem Steckbrett.

### Einschränkungen von FreeRTOS

Bisher haben wir die Software-Timer-API angewendet, ohne allzu viel über einige wichtige Einschränkungen des Callbacks anzumerken. Die Einschränkungen sind:

- Führen Sie niemals einen blockierenden Aufruf innerhalb des Callbacks aus (wie zum Beispiel `delay()` oder einen blockierenden `queue::push`).
- Vermeiden Sie lange Ausführungszeiten (die API würde abstürzen).
- Vermeiden Sie eine zu starke Belegung des Stacks (ESP32 Arduino ist wahrscheinlich auf etwa 1500 Bytes begrenzt). Funktionen wie `printf()` und `snprintf()` benötigen oft einen erheblichen Platz im Stack und sollten daher im Callback vermieden werden.
- Einige der FreeRTOS-API-Funktionen arbeiten mit einer internen Queue. Aus diesem Grund verwendet unser Demoprogramm in den Zeilen 88 und 92 `portMAX_DELAY`. Die Tiefe der Timer-Queue für das ESP32-Arduino-API beträgt 10, was aber in den allermeisten Anwendungen nicht ausgeschöpft werden dürfte.

### Zusammenfassung

Warum können wir nicht einfach einen Task zur Verwaltung der einzelnen Alarm-LEDs verwenden? Der Hauptgrund dafür ist, dass jeder Task einen Stack und einen Task Control Block (TCB) benötigt. Wenn Sie 32 LEDs ansteuern wollten, würde dies 32 x (356 + 1500) Bytes erfordern, insgesamt also 59.392 Bytes! Im Gegensatz dazu benötigt der Ansatz des FreeRTOS-Timers nur 40 Bytes pro Timer, insgesamt also 1200 Bytes (der Objekttyp `StaticTimer_t` ist 40 Bytes groß). Der Stack wird mit dem FreeRTOS-Dämon-Task für alle Zeitgeber gemeinsam genutzt (früher bekannt als *Timer-Service-Task*). ◀

200071-04

**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: <http://www.elektormagazine.de/190182-02>
- [2] Projektsoftware: [https://github.com/ve3wwg/esp32\\_freertos/blob/master/alertled/alertled.ino](https://github.com/ve3wwg/esp32_freertos/blob/master/alertled/alertled.ino)



# Mit dem Fuchs ins IoT (4)

## Einrichtung des Dashboards

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

Mit der Entwicklung eines individuell gestalteten Dashboards zur grafischen und numerischen Visualisierung der erfassten und via Sigfox übertragenen Messwerte schließt diese kleine Artikelreihe ab.

Nachdem in den vorhergehenden Artikelteilen die Sigfox-Basishardware für die Messwerterfassung und -übertragung vorgestellt und das grundlegende Konzept der Datenübertragung von Sigfox erläutert wurde, liegen unsere Messwerte jetzt auf dem Sigfox-Backend zur Abholung und zur weiteren Verarbeitung bereit.

Es soll nun ein freies webbasiertes Dashboard-Tool verwendet werden, mit dem der Anwender seine eigene Visualisierungs- und Auswerteoberfläche auf einem PC, Laptop, Tablet und/oder Smartphone entwerfen und betreiben kann. Nach einer entsprechenden Freigabe kann dieses Dashboard weltweit von jedermann beziehungsweise von jedem Berechtigten im Internet aufgerufen und angesehen werden.

Grundlage für die weiteren Ausführungen ist der Arduino-Sketch *Sigfox-3\_0.ino* für das Board MKR FOX 1200. Wählt man dort den Menüpunkt *4 Aussenden von drei Messwerten*, so werden fortlaufend drei Messwerte (Luftdruck, Lufttemperatur und Helligkeitswert des LDRs) erfasst und zunächst auf dem ePaper-Display des Motherboards und auf dem Seriellen Monitor der Arduino-IDE dargestellt. Auf Knopfdruck werden diese Werte dann automatisch über das Sigfox-Netz gesendet, vom Sigfox-Backend empfangen und entsprechend gespeichert. Und von dort geht es weiter zur Dashboard-Visualisierung.

### Die Dashboard-Plattform **thinger.io**

Bei *thinger.io* handelt es sich um eine Open-Source-IoT-Visualisierungsplattform, die es ermöglicht, schnell und einfach übersichtliche und anschauliche Visualisierungen von Daten zu erstellen. Sie ist für kleinere Projekte sogar kostenfrei benutzbar.

Mathematische Operationen mit den empfangenen Daten unterstützt das System jedoch derzeit (noch) nicht, was bedeutet, dass die Messwerte in schon fertig aufbereiteter Form angeliefert werden müssen. Das ist aber für unsere Arduino-Software auf dem MKR FOX1200 kein Problem. Die Visualisierung der Werte erfolgt auf sogenannten Dashboards (Armaturenbrett, Instrumententafel), die auch über einen Webbrowser öffentlich verfügbar gemacht werden können.

### Die notwendigen Schritte

Für die Entwicklung eines individuellen PC-Dashboards sind einige wenige einfache Schritte erforderlich:

- Anlegen eines kostenfreien Benutzerkontos (Account) bei *thinger.io*.
- Erstellen eines Datencontainers (Data Bucket) bei *thinger.io* zur Aufnahme der vom Sigfox-Backend gesendeten Messwerte.



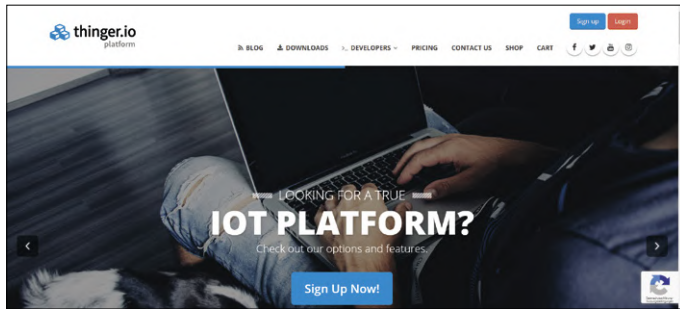


Bild 1. Homepage von thinger.io.

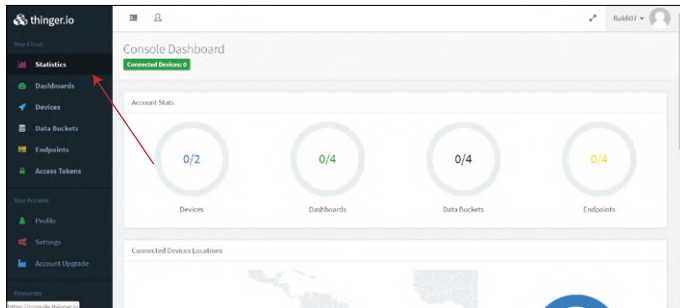


Bild 3. Hauptbildschirm des Accounts (Statistics).

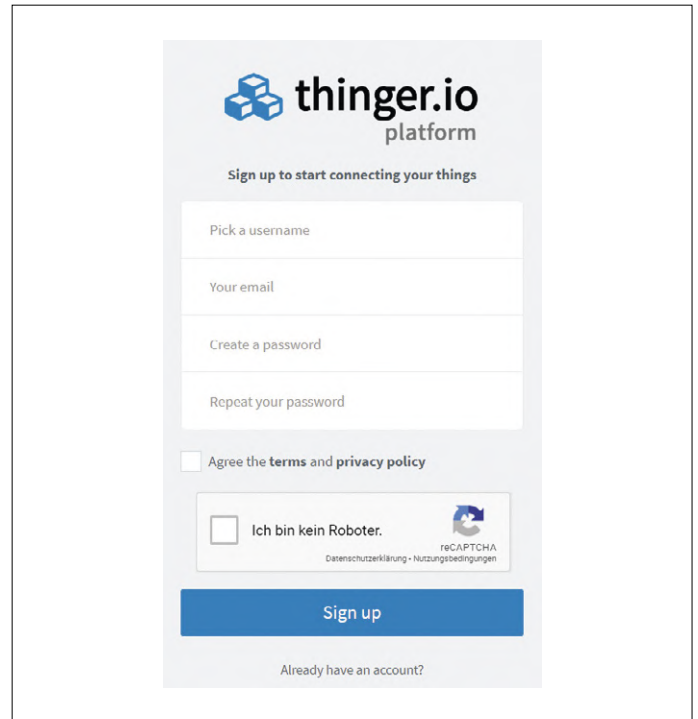


Bild 2. Eingabemaske zur Erstellung eines Erst-Accounts.

- Festlegung des Zugriffspunktes (Token) für das Sigfox-Backend und Erstellung einer Empfangsauthentifizierung (Access Token) bei thinger.io, damit das Sigfox-Backend die Genehmigung zur Übertragung von Daten an thinger.io erhält.
- Konfigurierung eines Callbacks auf dem Sigfox-Backend zur Internet-Übertragung der Daten vom Sigfox-Backend an den Data Bucket in thinger.io.
- Entwurf eines grafisch ansprechenden Dashboards auf der thinger.io-Seite zur Visualisierung der Daten.

### Anlegen eines Benutzerkontos bei thinger.io

Für die Nutzung von thinger.io ist zunächst die Einrichtung eines kostenfreien Benutzerkontos (Account) erforderlich. Dazu ruft man die Homepage von thinger.io [1] auf (**Bild 1**) und klickt oben rechts auf *Sign up*, so dass man zur Eingabemaske für Erstbenutzer (**Bild 2**) gelangt. Füllen Sie die Felder aus und klicken Sie auf *Sign up*.

Der Service ist bei einem kostenlosen Account zwar eingeschränkt, aber für die meisten kleineren Maker-Projekte mehr als ausreichend. Eine genaue Übersicht über die Möglichkeiten findet man unter dem Reiter *Pricing*.

Nachdem man seine E-Mail-Adresse bestätigt hat, gelangt man zur Login-Seite und kann sich anmelden. Es

erscheint der Hauptbildschirm *Statistics* Ihres Accounts, der der Ausgangspunkt für alle weiteren Aktionen ist (**Bild 3**). Auf der linken Seite befindet sich das Hauptmenü mit allen benötigten Funktionen zur Erstellung einer Visualisierung von IoT-Projekten.

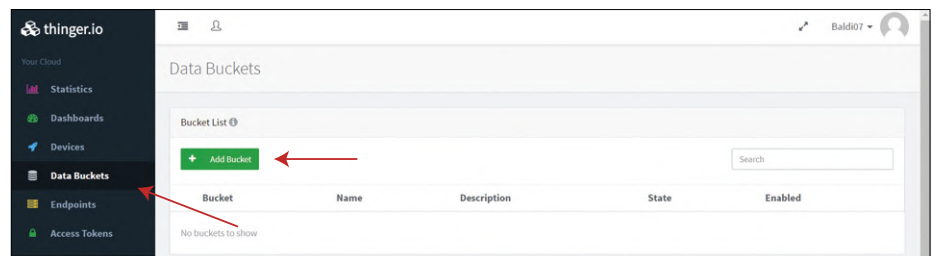


Bild 4. Anlegen eines neuen Data Buckets.

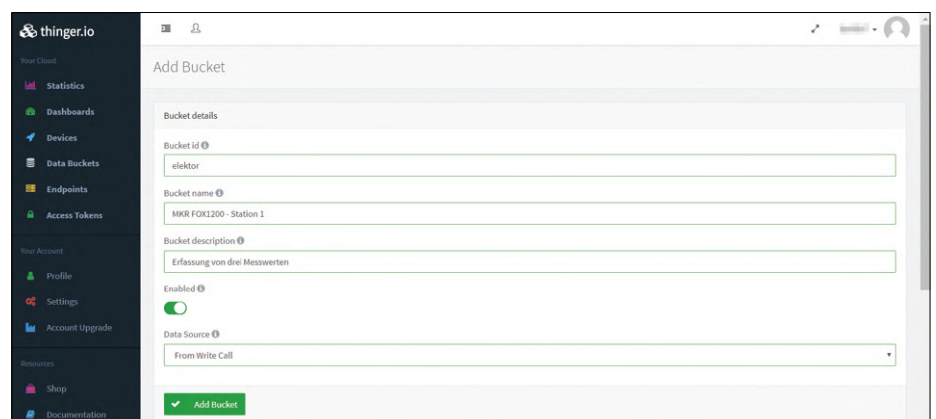


Bild 5. Festlegung der Eigenschaften des Data Buckets.

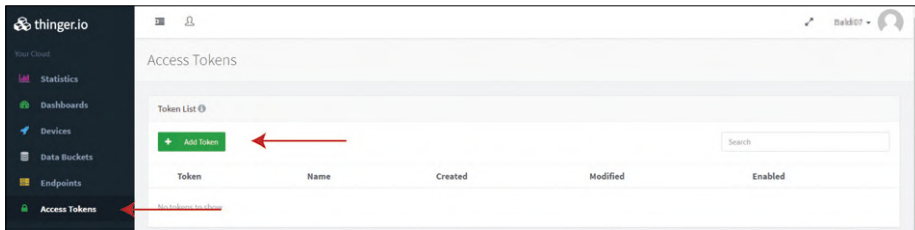


Bild 6. Anlegen eines neuen Tokens (Zugriffspunkt).

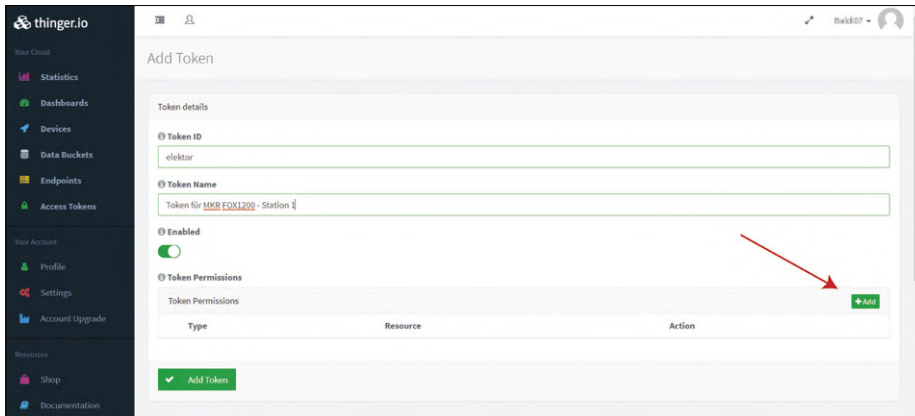


Bild 7. Festlegung der Token-Eigenschaften.

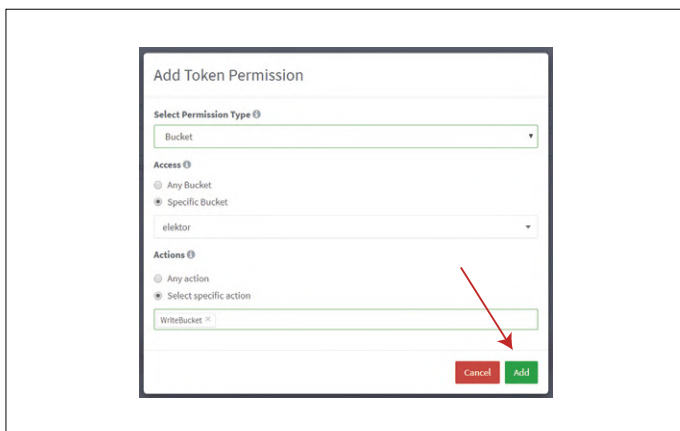


Bild 8. Konfiguration der Token Permission (Token-Zugriffserlaubnis).

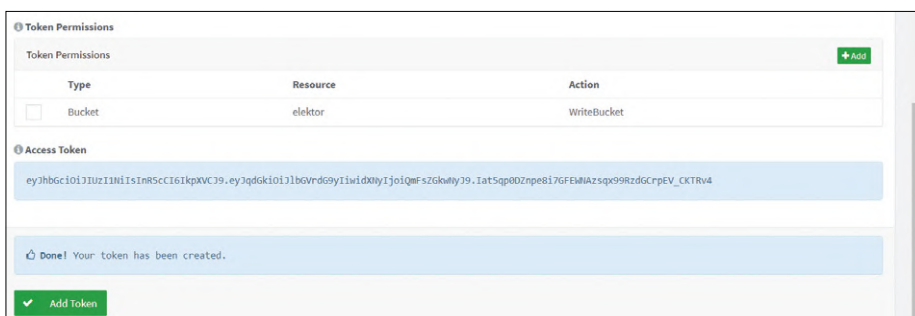


Bild 9. Der Token wurde erzeugt, aktiviert und die Zugriffsberechtigung vergeben.

## Die Bereitstellung eines Datencontainers bei thinger.io

Im ersten Schritt legt man einen *Data Bucket* (Bucket: Eimer, Kübel) an, eine Datenbank (Datencontainer), in der die Messwerte gespeichert und von dort wieder gelesen werden können. In diese Datenbank schreibt das Sigfox-Backend über den Sigfox-Callback automatisch die Messwerte.

Auf dem Hauptbildschirm klickt man dazu links auf *Data Buckets* und dann auf *Add Bucket* (Bild 4). Es erscheint das Fenster zur Eingabe der Data-Bucket-Eigenschaften (Bild 5). Die *Bucket id* ist ein eindeutiger Identifikator innerhalb des gesamten Accounts, der später für die Einrichtung mit Sigfox benötigt wird. Hier tragen wir zum Beispiel „elektor“ ein. In die Felder *Bucket name* und *Bucket description* kann man beliebige benutzerfreundlichere Informationen eingeben. Als Datenquelle wird im Auswahlfeld *Data Source* über das Pull-Down-Menü *From Write Call* ausgewählt. Dies bedeutet, dass die Daten durch einen Sigfox-Callback vom Sigfox-Backend via Internet zu thinger.io gesendet werden. Sobald also der Sigfox-Callback Daten sendet, werden die Daten hier hineingeschrieben.

Durch einen abschließenden Klick auf *Add Bucket* wird dieser nun erzeugt.

## Zugriffspunkt bei thinger.io für die Daten des Sigfox-Backends

Das Sigfox-Backend muss natürlich wissen, wohin die erfassten Messwerte via Callback gesendet werden müssen, mit anderen Worten, man muss dem Sigfox-Backend die Empfangsadresse bei thinger.io mitteilen. Dies wird im zweiten Schritt durch das Anlegen eines Tokens (Zugriffspunkt) mit der entsprechenden Berechtigung für den *Data Bucket* erledigt. Auf dem Hauptbildschirm klickt man dazu links auf *Access Tokens* und dann auf *Add Token* (Bild 6). Es erscheint ein Fenster zur Eingabe der Token-Eigenschaften (Bild 7). Hier wird zuerst eine beliebige eindeutige *Token ID* vergeben, zum Beispiel „elektor“. Unter *Token Name* kann wieder eine beliebige Beschreibung oder ein Name eingegeben werden.

Durch einen Klick auf *Add*, das kleine grüne Feld unten rechts außen, wird eine *Token Permission* (Token-Genehmigung beziehungsweise Zugriffserlaubnis) hinzugefügt, die jetzt ganz konkret konfiguriert werden muss (Bild 8).

Im Feld *Select Permission Type* wählen wir *Bucket* aus. Im Feld *Access* klicken Sie auf *Specific Bucket* und wählen dann im Dropdown-Menü den zuvor angelegten Bucket „elektor“ aus. Unter *Actions* wählen Sie *Select specific action* und fügen im Feld darunter (Drop-Down-Menü) die Auswahl *WriteBucket* hinzu.

Abschließend wird mit der Schaltfläche **Add** ein Token (Schreibberechtigung) für den gewünschten Ziel-Bucket „elektor“ vergeben und dieser Token zunächst in der Liste *Token Permissions* (Bild 7) aufgeführt. Klickt man nun auf den Button **Add Token** (Bild 7, unten links), so wird dieser Zugriffspunkt (Empfangsadresse für die Daten vom Sigfox-Backend) ganz konkret in *thinger.io* angelegt (Bild 9).

**Wichtig:** Im blauen Feld *Access Token* wird die automatisch von *thinger.io* generierte Zugriffsberechtigung (Authentifizierung) für das Sigfox-Backend angezeigt.

Das Sigfox-Backend muss später so konfiguriert werden, dass es diesen Access-Token-Code zusätzlich zur *thinger.io*-Zieladresse mitsendet, damit *thinger.io* die gesendeten Daten auch akzeptiert und im zugehörigen *Data Bucket* „elektor“ ablegt. Sie müssen diesen Access-Code mit copy & paste irgendwo sicher speichern. Damit sind die Einstellungen und Parametrierungen bei *thinger.io* zunächst beendet, so dass wir uns wieder dem Sigfox-Backend zuwenden können.

## Anlegen des Callbacks im Sigfox-Backend

Wenn der MKR FOX 1200 Daten an das Sigfox-Backend sendet, sollen diese automatisch zur weiteren Verarbeitung (Visualisierung) an *thinger.io* weitergeleitet werden. Hierzu bietet das Sigfox-Backend die schon mehrfach erwähnten *Callbacks* an. Ein solcher *Callback* ist nichts anderes als eine automatische Datenweiterleitung, die immer dann ausgeführt wird, wenn das Sigfox-Backend Daten von einer Sigfox-Station wie unserem MKR FOX 1200 empfangen hat. Eine solche Weiterleitung muss natürlich beim Backend richtig konfiguriert werden, damit die Daten auch beim gewünschten Empfänger ankommen.

Um nun die gesendeten Daten in unseren *thinger.io*-Data Bucket „elektor“ zu transferieren, bedienen wir uns des Tokens beziehungsweise des Access-Token-Codes, den wir (um genau zu sein, *thinger.io*) zuvor generiert haben. Mit anderen Worten: Das Backend greift über das Internet über eine http-Verbindung auf den Token zu und schreibt nach erfolgreicher Identifizierung mit dem Access-Token die Daten in den Data-Bucket.

Zum Anlegen des *Callback* loggen Sie sich auf dem Sigfox-Backend [2] ein. Auf der Hauptseite klicken Sie dann auf den Reiter *Device Type* und in der dann erscheinenden Liste (mit nur einem Eintrag) klicken Sie auf den Namen des Device-Typs *Arduino\_DevKit\_1* (Bild 10). Es erscheint die Übersichtsseite zu diesem Device. Dort klicken Sie in der Liste links auf *Callbacks*

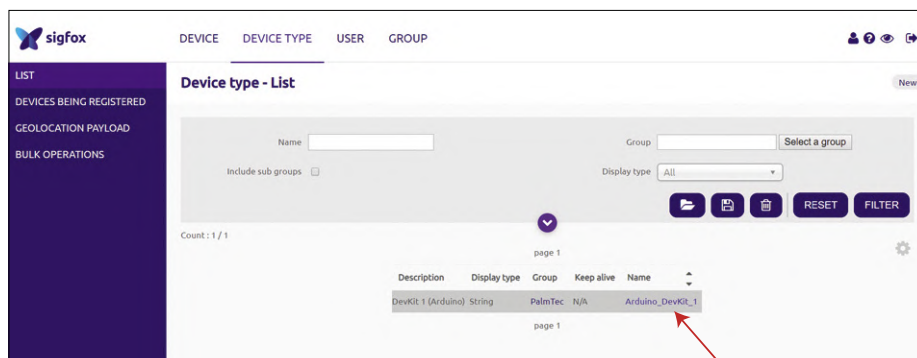


Bild 10. Auswahl des gewünschten Device Types.

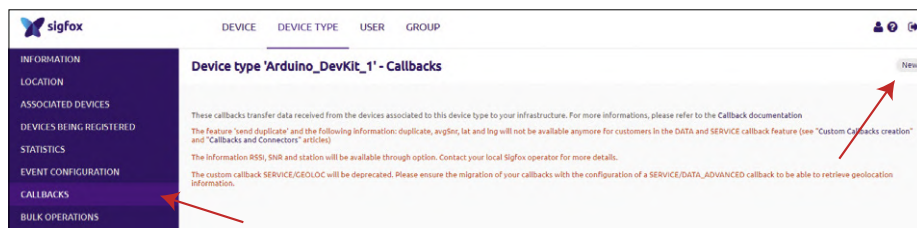


Bild 11. Das Callback-Fenster.

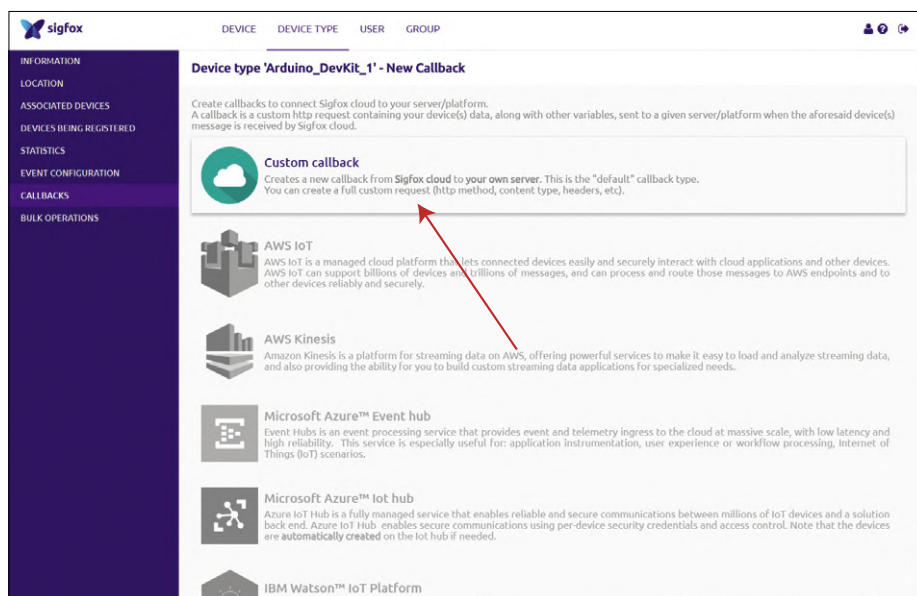


Bild 12. Die große Auswahl möglicher Callback-Typen.

(Bild 11). Wenn Sie in diesem Fenster rechts oben auf den kleinen Button **New** klicken, erscheint eine Liste mit allen möglichen Arten von Callbacks für die am weitest verbreiteten Arten von Dashboard/Cloud-Programmen (Bild 12). Hier wählen Sie *Custom Callback* aus. Im nun erscheinenden Fenster wird der Callback und das Aussenden der Daten vom Sigfox-Backend zu *thinger.io* konfiguriert (Bild 13).

**Achtung:** Hier müssen Sie sehr sorgfältig arbeiten und alle Eintragungen exakt so wie nachfolgend angegeben vornehmen, denn kleinste Abweichungen führen zu Fehlermeldungen beziehungsweise zum Nichtfunktionieren!

Eine genaue Beschreibung der einzelnen Festlegungen würde





det werden.

Unter *Url pattern* wird die Ziel-URL bei thinger.io angegeben, die folgendermaßen aufgebaut ist:

```
https://api.thinger.io/v1/users/{user_id}/buckets/  
{bucket_id}/data
```

In der Adresse sind *user\_id* der User-Name aus dem thinger.io-Account (Bild 2) und *bucket\_id* die Bucket-ID, also „elektor“ (Bild 5) (beides ohne geschweifte Klammern eintragen).

Unter *Use http Method* wird *POST* gewählt.

Als Nächstes muss für den Datentransfer zu thinger.io eine Authentifizierung angegeben werden. Dies geschieht in den beiden Feldern neben *Headers*. In das Feld *header* wird das Wort „Authorization“ eingegeben, daneben in Feld *value* das Wort „Bearer“, ein Leerzeichen und dahinter der erzeugte *Access Token* aus Bild 9:

```
Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLT...
```

Im Feld „*Content type*“ trägt man „application/json“ ein, da der nachfolgende Datenkörper (*Body*) im JSON-Format formatiert wird.

In diesem letzten Schritt wird festgelegt, was genau an thinger.io gesendet werden soll. Die Einträge (im JSON-Format) im Feld *Body* müssen **ganz genau** so übernommen werden:

```
{  
  "Device-ID" : "{device}",  
  "AtmPressure" : {customData#pressure},  
  "Temperature" : {customData#temp},  
  "Light-Sensor" : {customData#ldr}  
}
```

Der *Body* wird durch geschweifte Klammern {...} eingegrenzt. Was bedeuten nun diese Angaben?

Jede Zeile in den geschweiften Klammern beginnt mit einem Anführungszeichen „“, danach steht ein beliebiger, aber eindeutiger Variablen-Namen, der wiederum von einem Anführungszeichen abgeschlossen wird. Dann gibt man die Sequenz Leerzeichen-Doppelpunkt-Leerzeichen ein, woraufhin der Wert für die Variable (der Wert aus dem Sigfox-Backend) angegeben wird. Der Variablenname und der Variablenwert werden dann später so nach thinger.io gesendet und im entsprechenden Data Bucket gespeichert.

Es bedeuten nun:

- `"Device-ID" : "{device}"`  
ist die eindeutige Device-Nummer der Sigfox-Station (unseres MKR FOX 1200), die im Backend vorliegt. Diese Information senden wir zusätzlich zu den Messwerten, um die Station später eindeutig identifizieren zu können.
- `"AtmPressure" : {customData#pressure}`  
sendet eine Variable mit dem Namen *AtmPressure*. Der Wert dieser Variablen entspricht dem Wert der Backend-Variablen *pressure*, die zuvor beim *Custom payload config* festgelegt wurde. Das Schlüsselwort *customData#* weist darauf hin, dass nachfolgend der Wert einer Variablen aus dem *Custom payload config*-Feld eingesetzt werden soll. Mit anderen Worten: So werden dieser Variablen-Name und der zugehörige Wert nach thinger.io übermittelt.

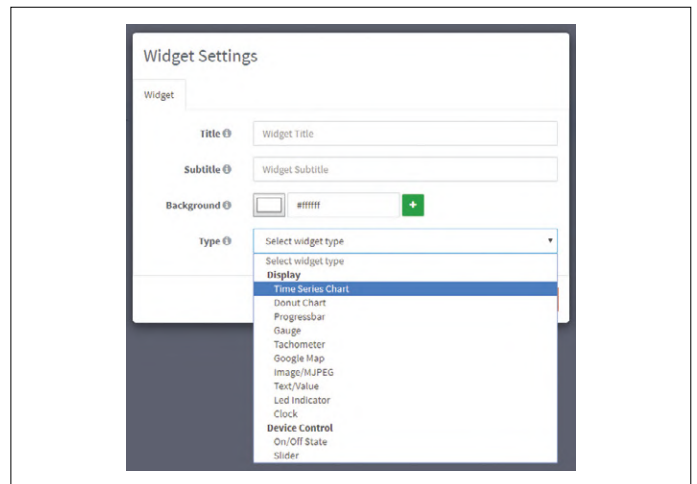


Bild 18. Basisfenster zur Erstellung eines Widgets.

- Die beiden letzten Zeilen erklären sich entsprechend. Jede Zeile (bis auf die letzte) muss mit einem Komma abgeschlossen werden.

Damit ist die Einrichtung des Callbacks auf dem Sigfox-Backend zur Übertragung der gewünschten Werte in den Data Bucket „elektor“ bei thinger.io abgeschlossen. Wenn Sie alles richtig gemacht haben, können Sie beruhigt auf *OK* klicken.

Wenn nun unser MKR FOX1200 seine Payload an das Sigfox-Backend sendet und diese dort fehlerfrei empfangen worden ist, so sendet das Sigfox-Backend **sofort und unmittelbar** diese Daten in Form des gerade eingerichteten Callbacks zur weiteren Verarbeitung an thinger.io.

## Das Dashboard bei thinger.io

Damit kommen wir jetzt zum letzten Schritt, den Feinschliff unserer Anwendung. Die Daten sollen in einer grafisch ansprechenden Form im Dashboard präsentiert werden. Um ein Dashboard anzulegen, klickt man in der Hauptkonsole von thinger.io (Bild 3) links auf den Punkt *Dashboards* und im folgenden Fenster auf den Button *Add Dashboard* (Bild 14). Im Fenster *Dashboard details* (Bild 15) füllt man die Felder aus: Die *Dashboard id* ist ein eindeutiger Identifikator innerhalb des gesamten Accounts. In die Felder *Dashboard name* und *Dashboard description* können Bezeichnungen eingetragen werden, die für den Benutzer leicht durchschaubar sind. Dann beenden Sie die Eingaben mit *AddDashboard*. Damit ist das Dashboard in der so genannten *Dashboard list* angelegt, die man sieht, wenn man in der Hauptkonsole wieder auf den Menüpunkt *Dashboards* klickt (Bild 16).

Klicken Sie auf den Namen des Dashboards, so öffnet sich die Entwurfsoberfläche für ein leeres Dashboard (Bild 17). Rechts oben befindet sich ein kleiner Schalter, den Sie auf „grün“ schalten, um in den Editier-Modus zu gelangen. Außerdem gibt es zwei Schaltflächen namens *Add Widget* und *Settings*. Dashboard-Anzeigeelemente werden bei thinger.io *Widgets* genannt (Komponente einer Benutzeroberfläche). Es steht eine Vielzahl davon zur Verfügung, zum Beispiel Zeit-Verlaufs-Diagramme, Donut-Diagramme, Balkenanzeigen, Analoganzeigen, Google Maps für Positionsdaten, Bilder, Textanzeigen, LEDs und eine Uhr. Für unsere Anwendung wollen

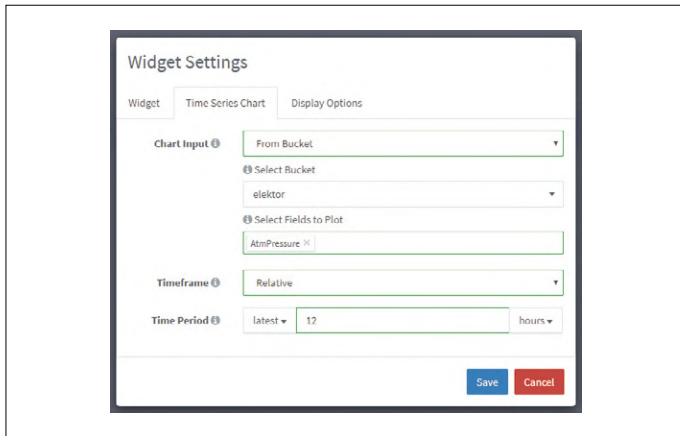


Bild 19. Konfiguration des Zeitverlauf-Diagramms.

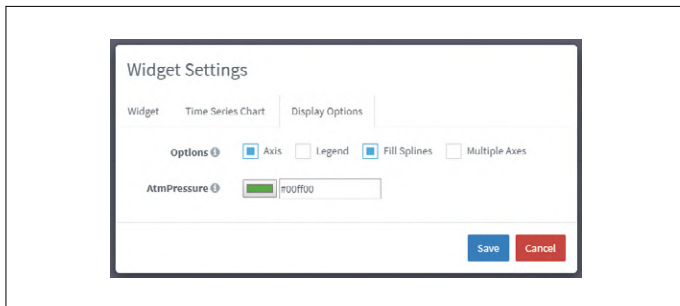


Bild 20. Festlegung der Display-Grundeigenschaften.

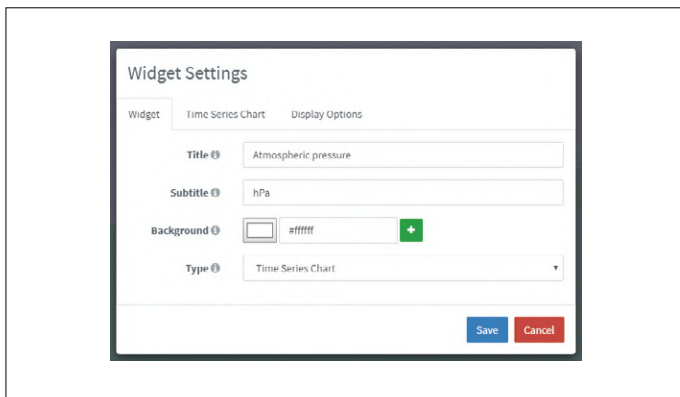


Bild 21. Festlegungen im Reiter *Widget*.

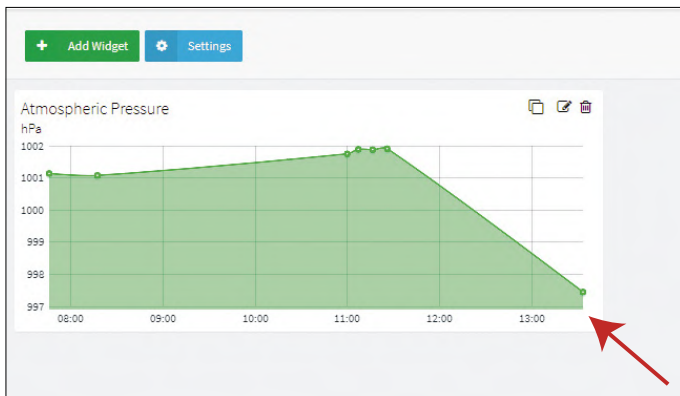


Bild 22. Das Zeitverlaufdiagramm auf dem Dashboard (mit einigen Messwerten).

wir für jeden der drei Messwerte ein Zeit-Verlaufs-Diagramm und eine Zahlenanzeige des jeweils letzten übermittelten Messwertes anlegen.

Klickt man auf *Add Widget*, poppt ein Auswahl/Einstellungsfenster namens *Widget Settings* (Bild 18) auf. Unten bei *Type* lässt sich angeben, welchen Widget-Typ man anlegen möchte. Hier wählen wir *Time-Series-Chart*. Es erscheinen zwei zusätzliche Tabs (Bild 19). Im Reiter *Time-Series-Chart* wählt man im Feld *Chart-Input* die Option *From Bucket* aus. Es erscheinen nun nach und nach weitere Felder, die wir entsprechend ausfüllen. Im Feld *Select Bucket* wird der gewünschte Bucket „elektor“ ausgewählt. Im nächsten Feld *Select Fields to Plot* wählen wir den darzustellenden Messwert aus.

Beginnen wir mit *AtmPressure*. Unter *Timeframe* wählen wir *Relative* und legen das darzustellende Zeitfenster auf *12 hours* ein. Damit werden die Werte der letzten 12 Stunden im Diagramm angezeigt, was natürlich beliebig variiert werden kann. Klicken Sie nun auf den Reiter *Display Options* (Bild 20) und schalten dort *Multiple Axes* aus, da für jeden Messwert ein eigenes Diagramm angelegt werden soll und deshalb nicht mehrere Y-Achsen benötigt werden. Durch die bereits aktivierte Option *Axis* werden X- und die Y-Achse im Diagramm automatisch skaliert. Im Feld bei *AtmPressure* kann die Farbe der Diagrammlinie festgelegt werden. Wir haben uns für ein zartes Grün entschieden. Wenn man jetzt noch *Fill Splines* aktiviert, wird der Bereich (die Fläche) zwischen der Diagrammkurve und der X-Achse mit einer Farbe ausgefüllt, die man im Feld *AtmPressure* festlegt. Mit der Option *Legend* kann man die einzelnen Kurven separat beschriften lassen. Dies ist natürlich nur sinnvoll, wenn es mehrere Kurven im Diagramm gibt.

Kommen wir nun zum letzten Reiter *Widget* (Bild 21), in dem man einen Titel und einen Untertitel für das Diagramm eingeben kann. Im Feld *Title* tragen wir *Atmospheric pressure* und unter *Subtitle* die Einheit *hPa* ein. Bei Bedarf kann man unter *Background* auch noch die Hintergrundfarbe für das gesamte Diagramm anpassen, wir wollen es aber nicht übertreiben und bleiben beim voreingestellten Weiß. Durch einen Klick auf *Save* wird die Parametrierung des Diagramms beendet und das Diagramm zum Dashboard hinzugefügt (Bild 22). An der unteren rechten Ecke kann das Diagramm größer oder kleiner gezogen und mit der Maus auf dem Dashboard-Bereich verschoben werden. Die Größeneinstellung ist nicht stufenlos, so dass das Diagramm beim Ziehen „hin und her“ springt. Der aktuelle Luftdruck-Messwert soll auch in Form einer Zahl dargestellt werden (Bild 23). Der Reiter *Add Widget* öffnet wieder das Menü zum Hinzufügen von Widget-Elementen; wir wählen im Feld *Type* die Option *Text/Value* aus.

Im Feld *Widget Value* tragen wir wieder *From Bucket* ein, wählen unseren Bucket „elektor“ aus und geben an, welchen Wert wir darstellen möchten (hier: „AtmPressure“). Bei *Bucket Value* stellen wir *Latest Value* ein. Im Reiter *Display Options* können verschiedene Einstellungen getätigt werden. Wir stellen hier beispielsweise zwei Nachkomma-Stellen ein (*Decimal Places*) und im Feld *Units* die Einheit „hPa“. Diese wird dann später im Widget hinter dem Wert angezeigt. Bei Bedarf können hier noch weitere Einstellungen vorgenommen werden, die wir hier aber nicht durchführen. Im Reiter *Widget* tragen wir im Feld *Title* wieder *Atmospheric pressure* ein. Das Feld *Subtitle* kann hier frei bleiben, da die Einheit schon hinter dem Wert angezeigt wird. Damit sind die beiden Dashboardanzeigen für den Luftdruck-



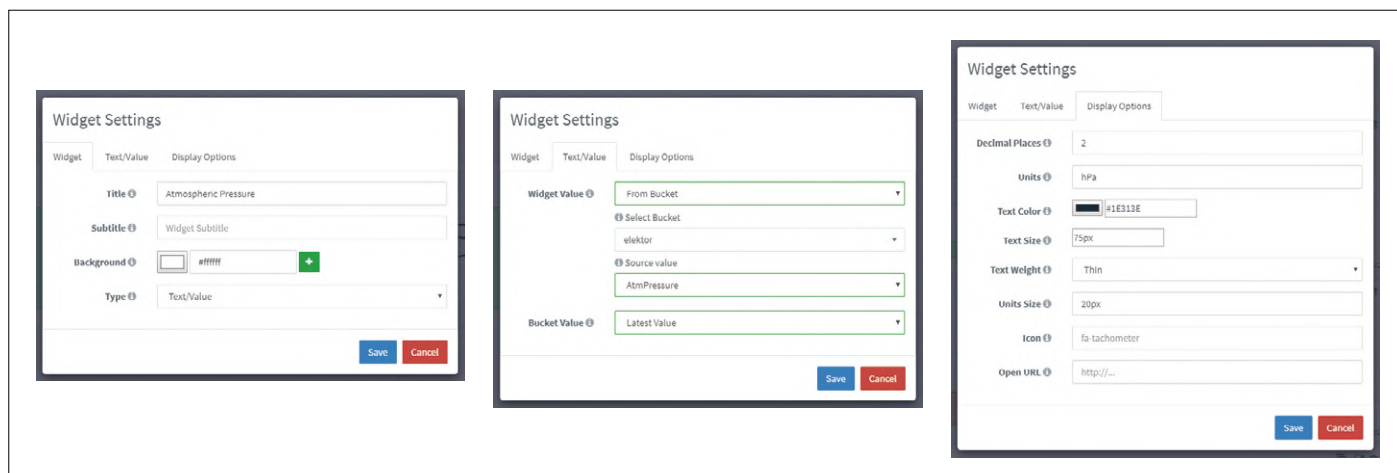


Bild 23. Festlegungen der numerischen Anzeige für den Luftdruck-Messwert.

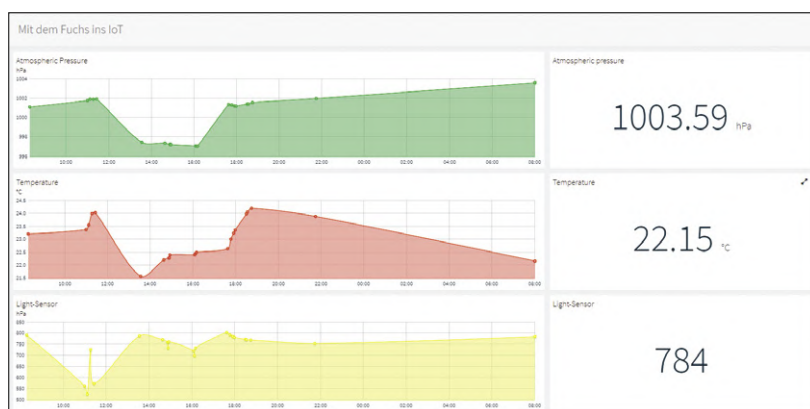


Bild 24. Das fertige Dashboard.

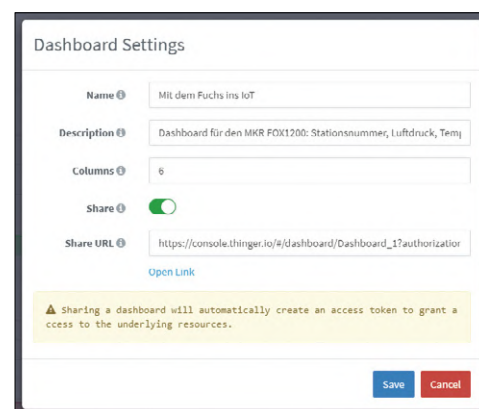


Bild 25. Die Dashboard-Settings.

Messwert fertig. Die Anzeigen für die beiden anderen Messwerte werden analog erstellt. Das Ergebnis könnte und sollte in etwa so aussehen wie **Bild 24**. **Wichtig:** Damit alle Eingaben und Änderungen in thinger.io gespeichert werden, müssen Sie zum Abschluss den Editier-Modus einmal aus- und dann wieder einschalten! Weitergehende Informationen zum Dashboard-Entwurf unter thinger.io findet man bei [7].

Was nun noch fehlt, ist die Freigabe des Dashboards für einen Zugriff „von außen“. Zurück im Dashboard-Menü gelangen Sie durch einen Klick auf die blaue Schaltfläche *Settings* zu den Dashboard-Einstellungen (**Bild 25**). Hier können ein Name und eine Beschreibung für das Dashboard eingetragen werden (ist aber nicht mehr notwendig, da wir dieses schon erledigt haben). Mit dem Schalter *Share* wird das Dashboard freige-

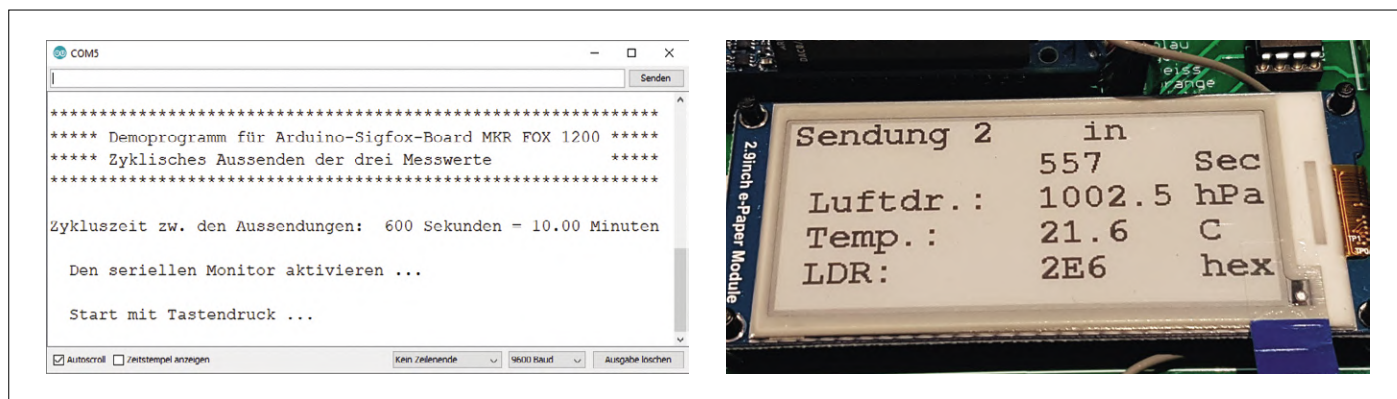


Bild 26. Die Ausgabe von *Sigfox-Demo.ino* auf dem Seriellen Monitor und auf dem ePaper-Display.



Bild 27. Macht auch auf dem Smartphone einen guten Eindruck!

geben. Der (extrem lange) Link zum Dashboard wird im Feld *Share URL* angezeigt. Über diesen Link kann man jetzt von überall auf dieser Welt auf das Dashboard zugreifen und die Werte begutachten. Da der URL sehr lang ist, empfiehlt es sich, sie mit Hilfe eines URL-Shorteners wie [bitly.com](https://bitly.com) zu kürzen. Diese gekürzte URL kann man in einen QR-Code verwandeln, der von Smartphones gescannt werden kann. So muss man beim Smartphone nichts umständlich eintippen.

Wichtig ist auch, dass dieses freigegebene Dashboard von [thinger.io](https://thinger.io) **automatisch** für die Öffentlichkeit **schreibgeschützt** (nur lesbar) ist. Es können also keine Einstellungen und Veränderungen durch Dritte vorgenommen werden. Dies ist weiterhin nur nach einem entsprechenden Login über das [thinger.io](https://thinger.io)-Portal möglich.

Um das Gesamtsystem umfassend auszutesten, ist es etwas umständlich, mit unserem Arduino-Sketch *Sigfox-3\_0.ino* immer wieder auf Knopfdruck ein Sigfox-Telegramm zu senden. Daher gibt es einen weiteren Sketch namens *Sigfox-Demo.ino*, der nun im Softwarepaket [8] enthalten ist. Er erfasst automatisch in periodischen 10-Minuten-Intervallen die drei Messwerte und sendet sie aus (**Bild 26**). Das Intervall wird im Programmtext des Sketches durch die globale Variable `w_zeit` festgelegt beziehungsweise nach Belieben geändert. Damit sind auch sehr einfach Langzeitmessungen beziehungsweise Langzeitaufzeichnungen der Messwerte möglich. Denken Sie aber stets daran, dass im Sigfox-Netzwerk nur maximal 140 Telegramme pro Tag gesendet werden dürfen, was etwa einem Telegramm pro 10 Minuten entspricht. Dann ist es auch nicht weiter kritisch, dass beim freien Account von [thinger.io](https://thinger.io)

nur einmal pro Minute in den Bucket geschrieben werden darf. **Bild 27** zeigt, wie schick unser gerade erstelltes Dashboard auf einem Smartphone-Bildschirm aussieht.

### Zum Abschluss

Nun sind wir nun am Ende unserer kleinen Reihe zum Einstieg in die Welt des 0G-Netzwerks Sigfox angekommen. Die sehr flexible und vielseitige Programmierung des Arduino MKR FOX 1200 ermöglicht es, eine Vielzahl verschiedenster Daten zu erfassen und über das Sigfox-Netzwerk weltweit zu übertragen. Die ebenfalls sehr einfachen und umfangreichen Gestaltungsmöglichkeiten von [thinger.io](https://thinger.io) lassen den schnellen und effektiven Entwurf individueller Dashboards zur Anwenderkommunikation zu.

Wir wünschen Ihnen jetzt noch viel Spaß und Erfolg mit Sigfox bei Ihren Streifzügen durch das Internet of Things! ◀

190281-D-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] Thinger: [www.thinger.io/](https://www.thinger.io/)
- [2] Sigfox-Backend: <https://backend.sigfox.com/>
- [3] Infos zur Sigfox-Callback-Konfiguration: <https://support.sigfox.com/docs/callbacks-documentation>
- [4] Infos zur Sigfox-Callback-Konfiguration: <https://support.sigfox.com/docs/custom-callback-creation>
- [5] Infos zur Sigfox-Callback-Konfiguration: <https://backend.sigfox.com/apidocs/callback>
- [6] Byte-Reihenfolge beim Speichern von Zahlenwerten: <https://de.wikipedia.org/wiki/Byte-Reihenfolge>
- [7] Infos zum Dashboard-Entwurf: <https://github.com/thinger-io/Docs/blob/gh-pages/console/dashboards.md>
- [8] Projektseite: [www.elektor.de/190281-D-01](http://www.elektor.de/190281-D-01)

# 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 dritten Episode der Rubrik präsentieren wir das Labor von Rainer Schuster, der vielen Elektor-Lesern nicht unbekannt sein dürfte.

Rainers Labor macht einen so organisierten und aufgeräumten Eindruck, dass man neidisch werden könnte. Er schreibt: „Mein Heimlabor ist schon fast so alt wie (die deutsche Ausgabe von) Elektor, zumindest was die Ausstattung betrifft. Alles begann im Jahr 1972 mit einem Multimeter und einem Netzgerät, und im Laufe der Jahre kamen all die anderen Geräte hinzu, die Sie in **Bild 1** sehen.“ Die meisten Geräte entstanden aus Bausätzen der (nicht mehr existierenden) Münchener Firma *RIM electronic* und der amerikanischen Firma *Heathkit*, die bis in die 1980er Jahre hervorragende und zu Recht berühmte Kits lieferte.

Auf dem Labortisch kann man auch einen Blick auf das Projekt erhaschen, an dem Rainer während der Aufnahme arbeitete. Es geht um einen Gitarrenverstärker für seinen Sohn, der auf der Guitarix-App basiert, die auf einem Raspberry Pi 4 läuft. Aber auch selbstentwickelte Geräte gehören zum Labor, wie der „Transistor-Kennlinienschreiber“, der in Elektor Februar 2009 veröffentlicht wurde (**Bild 2**) [1].

**Bild 3** zeigt einige weitere Eigenentwicklungen: den „HV Power Supply / Tube Curve Tracer“ [2] (oben) und die „Electronic Load for DC and AC“ [3] (unten), beide auf der Elektor-Labs-Webseite veröffentlicht.



Und schließlich konnte man Rainers Labor in dem Video [4] sehen, das er anlässlich der *Elektor-Videolympiade 2017* gedreht hat, wo er mit seiner Präsentation einiger der berühmten „Scherz-Schaltungen“, die Elektor im jährlichen Halbleiterheft veröffentlichte, einen ehrenvollen dritten Platz errang. ◀

200086-04

### Weblinks

- [1] Transistor-Kennlinienschreiber: [www.elektormagazine.de/080068](http://www.elektormagazine.de/080068)
- [2] HV Power Supply / Tube Curve Tracer: [www.elektormagazine.com/labs/hv-power-supply-with-tube-and-semiconductor-curve-tracer](http://www.elektormagazine.com/labs/hv-power-supply-with-tube-and-semiconductor-curve-tracer)
- [3] Electronic load for DC and AC: [www.elektormagazine.de/labs/electronic-load-for-dc-and-ac](http://www.elektormagazine.de/labs/electronic-load-for-dc-and-ac)
- [4] Video: [www.youtube.com/watch?v=2f75IK12Wjo&list=PL2a34OA-WuybkArjgAnXiIfqUc3KYf19A&index=4](https://www.youtube.com/watch?v=2f75IK12Wjo&list=PL2a34OA-WuybkArjgAnXiIfqUc3KYf19A&index=4)





REVIEW

# Elektronische USB-Last Joy-iT HD35

## Elektronische Last zum Test der Belastbarkeit von USB-Ports

Von Dr. Thomas Scherer

Das ist eine echte Spezialität: Eine elektronische Last zum Einstecken in oder Verbinden mit USB-Ports. Damit kann man testen, ob z.B. das superduper Ladegerät aus China mit drei USB-Ports und angeblich 4 A Ausgangsstrom und QC-Fähigkeiten tatsächlich hält, was es verspricht. Das Gerät ist natürlich auch für Entwickler gedacht, um die Belastbarkeit von USB-Ports von eigenen Projekten zu testen.



Als ich dieses kleine komische Teil namens HD35 von Joy-iT im Internet sah, war ich neugierig. Eine elektronische Last für den USB-Port? Auf was für Ideen man so kommt in Fernost! Aber als ich genauer las, was das Teil kann, war mir der Sinn schnell klar. Damit kann man die Belastbarkeit von USB-Ports aller Art testen. Nicht nur bei selbstgebaute n Geräten und bei

Produkten aus eigener Entwicklung ist das aufschlussreich, es lassen sich auch allerlei fertige Ladegeräte damit auf Herz und Nieren prüfen. Zwar hatte ich mit dem UM25C schon einmal ein Review eines USB-Testers [1] verfasst – aber diesmal handelt es sich um ein ganz anderes Gerät mit anderen Fähigkeiten.



Bild 1. Diese kleine Plastik-Box enthält die Elektronik des USB-Lastmoduls HD35.



Bild 2. Die Plastik-Box von unten. Hier sind die technischen Daten und der Hersteller-URL angegeben.

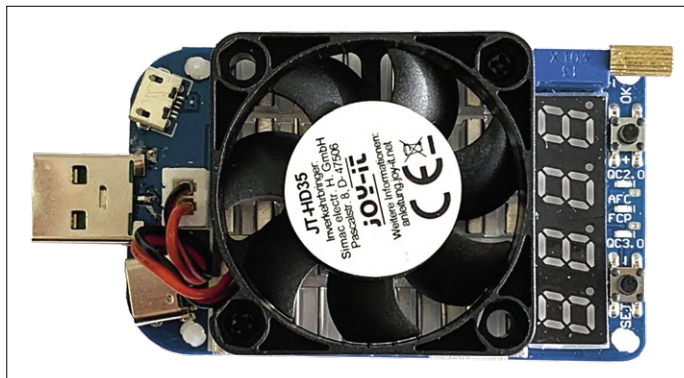


Bild 3. Mein USB-Lastmodul in aller Pracht. Alles ist dran: 3 USB-Eingänge, Lüfter, Display, 3 LEDs, 3 Taster (der dritte ist versteckt hinten unten) und ein Poti.

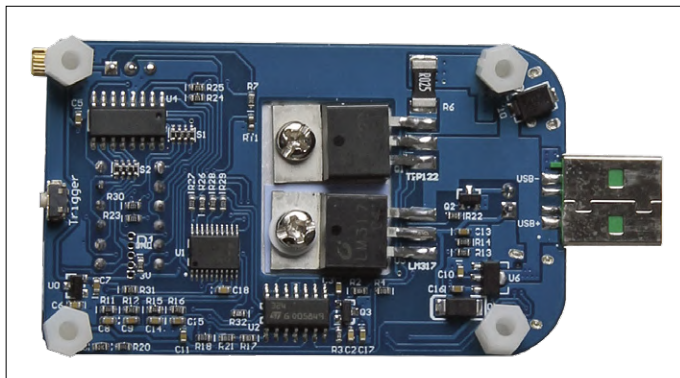


Bild 4. Unterseite des USB-Lastmoduls mit der eigentlichen Elektronik (Bild: Joy-iT).

## Features

Zunächst zu dem, was Joy-iT zu seinem Produkt sagt: „Mit dem JT-HD35 erhalten Sie einen kompakten, elektronischen USB-Lastwiderstand mit einer Leistungsfähigkeit von bis zu 35 Watt.“ Letzteres ist ein Wort, wenn es um USB geht. Weiter wird davon gesprochen, dass es mit diversen Modi für unterschiedliche Schnellladeverfahren ausgestattet ist, dass der Maximalstrom 5 A beträgt und dass man ihn einstellen kann (was es wohl „für fast jeden Anwendungszweck geeignet“ macht) sowie dass es einen Kühlkörper mit Lüfter „für eine effektive Kühlung des gesamten Gerätes“ besitzt. Und das stimmt. Die technischen Daten sind:

- Betriebsspannung: 4 bis 25 V (DC)
- Laststrom: 0,25 bis 5 A
- Leistung: max. 35 W (daher der Name ;-)
- Indikatoren: 3 Status-LEDs (grün, blau, rot)
- Display: vierstellige Siebensegmentanzeige LED rot
- Bedienung: 3 Taster, 1 Mehrgangpotentiometer (Strom)
- Anschlüsse: USB 2.0, Micro-USB, USB-C
- Lüfter: 40 mm, 0 bis 8.000 U/min, temperaturgesteuert
- Abmessungen: 84 x 41 x 28 mm
- Gewicht: 52 g

## Auspacken

Im Postpaket war eine unscheinbare kleine Plastik-Box (**Bild 1**) mit dem Foto des Moduls, das drin sein sollte. Meine nicht ganz unbegründete Vermutung war sofort: In dieser kleinen Box steckt zwar die Elektronik, aber nackt, d.h. ohne auch nur die geringste Spur einer gedruckten Betriebsanleitung. Nachdem oben kein Hinweis auf die Anleitung war und auch kein QR-Code mit einem Link dahin, drehte ich die Box um (**Bild 2**). Aber nix da, nada. Immerhin findet man unten und an der Stirnseite einen aufgedruckten URL von Joy-iT und gelangt so auf deren Webseite. Nach Eingabe des Produktnamens kommt man auf die Produktseite, und siehe da: hier findet man das Manual [2] als PDF.

Ehrlicherweise muss ich zugeben, dass ich gedruckte Anleitungen bei der Vielzahl an Geräten und Tools wirklich nicht vermisste. Ein PDF heruntergeladen und im Ordner „Anleitungen“ auf meinem PC verstaut spart mir mühsame Suche von Hand – und ein PDF ist auch so einfach wie bequem nach Stichworten durchsuchbar. Zunächst wollte ich aber einmal ohne Blick ins Handbuch schauen, wie weit ich komme und wie selbsterklärend dieses Produkt ist.

**Bild 3** zeigt den ausgepackten Inhalt der kleinen Plastik-Box. Witzigerweise ist der Lüfter mit nur zwei von vier möglichen



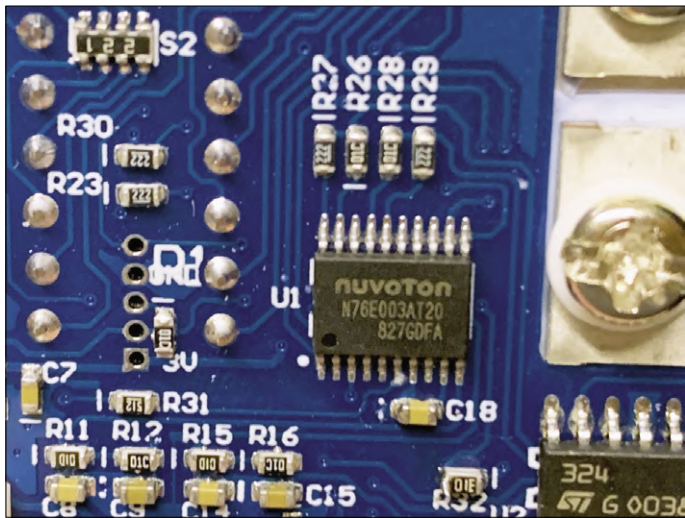


Bild 5. Die Vergrößerung zeigt die Typennummern der Chips klar und lesbar.



Bild 6. Zwei Netzteile mit drei Kabeln dienen als Versuchsobjekte..

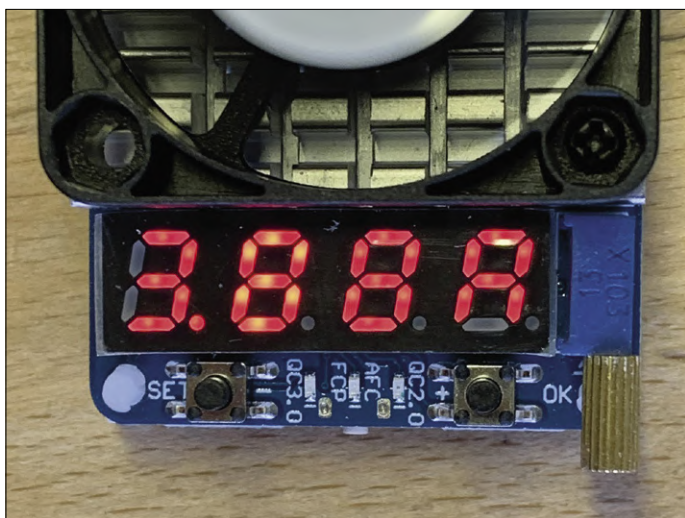


Bild 7. Beweis: Das QC-3.0-Ladegerät von Ravpower ließ sich via USB-C an 5 V immerhin mit 3,88 A belasten.

Schrauben am Kühlkörper befestigt. Da wurde also vermutlich ein ganzer Cent gespart, aber mechanisch reicht es aus. In **Bild 4** wird die Unterseite der Platine enthüllt. Man sieht mit dem TIP122 einen 65-W-NPN-Leistungsdarlington (vermutlich die eigentliche Last) und den einstellbaren Spannungsregler LM317. Außerdem entdeckt man mit dem Mikrocontroller U1 = N76E003AT20 ein kleines und mit 28 ¢ extrem preiswertes 8051-Derivat nebst Peripherie-Chips und passiven Bauteilen. Weil die Schrift dieser Winzlinge nicht gut zu lesen ist, fotografiere ich sie üblicherweise und vergrößere das Resultat (siehe **Bild 5**).

Nachdem das klar war, konnte die Probiererei losgehen...

### Ausprobieren

Als Testobjekte habe ich zwei Netzteile/Ladegeräte mit drei Kabeln (**Bild 6**) ausgewählt. Das kleine Schwarze rechts mit dem Micro-USB-auf-USB-A-Kabel ist ein billiger Standard-Lader mit 5,1 V und 1 A Belastbarkeit, wie er bei vielen (alten) Smartphones beilag. Es kann nichts außer diese fünf Watt liefern. In der Mitte umringt von einer USB-A-Verlängerung und einem USB-C-Kabel prangt das weiße Power-Ladegerät von Ravpower. Es bietet an seiner USB-A-Buchse 5 V mit 2,4 A und an der USB-C-Buchse nicht nur 5 V mit 3 A, sondern dank QC-3.0-Kompatibilität auch noch 9 V oder 15 V mit 2 A sowie 20 V mit 1,5 A. Die Gesamtbelastbarkeit ist mit 36 W angegeben, was es als prima Versuchskaninchen prädestiniert.

Nach dem Einschalten zeigt sich bei Anschluss an das schwarze Ladegerät zunächst die Spannung zweistellig auf dem Display, also etwa „5.1U“. Etwas wildes Rumdrücken auf den Tastern machte mir klar: Die Set-Taste schaltet bei jedem Druck zwischen der Anzeige von Spannung, Leistung (zunächst „0.0P“) und Strom (hier „1.00A“ blinkend) um. Drückt man jetzt einmal auf die OK-Taste, hört das Blinken auf und die Anzeige des Stroms bleibt stehen. Der nach 20 s einsetzende Lüfter machte mir klar, warum: Blinken bedeutet „Anzeige des eingestellten Stroms der Last, aber Last nicht angelegt“. Ist die Anzeige aber stabil, wird der eingestellte Strom auch entnommen. Noch ein Druck auf „Ok“ lässt die Stromanzeige wieder blinken.

Drehte ich mit dem Poti den Strom höher, so bricht das kleine schwarze Netzteil gleich ein und man konnte bei 1,1 A nur noch etwa 3,5 V messen. Es hat also erstaunlicherweise eine Strombegrenzung und fackelt nicht einfach bei Überlast ab! Interessanterweise zeigt sich hier auch, dass Micro-USB ein zwar kleiner, aber für Versorgungszwecke völlig ungeeigneter Stecker ist. Schon bei 1 A zeigt das Display nur 4,9 V. Bei Verwendung der USB-A-Verlängerung am gleichen Netzteil bleibt die Spannung auch da immerhin noch auf 5,0 V.

Nun zum Powernetzteil: Über die USB-A-Verlängerung an seiner USB-A-Buchse angeschlossen reagiert die Strombegrenzung erst bei etwa 3 A. Über das USB-C-Kabel angeschlossen konnte ich sogar 3,88 A einstellen, bis sich das Netzteil abschaltete. **Bild 7** zeigt das Beweisfoto. Ravpower hatte also nicht zu viel versprochen.

### Und jetzt?

Viel weiter kam ich nicht. Mir hat sich nicht von selbst erschlossen, wie ich die verschiedenen anderen Spannungen nach QC 2.0 oder 3.0 testen könnte. Also doch das Handbuch



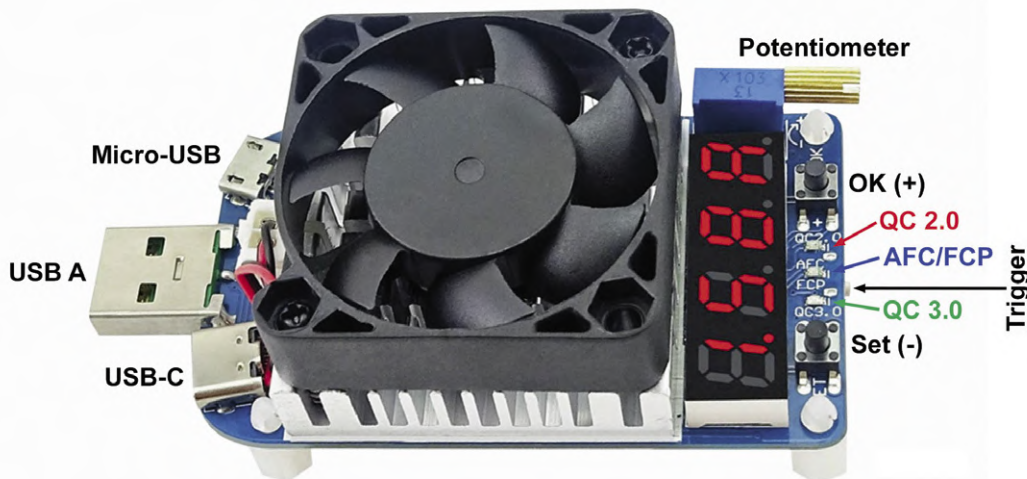


Bild 8. Anschlüsse, Anzeigen und Bedienungselemente des HD35 (modifiziertes Bild von Joy-iT).

herunterladen und studieren. Gleich fiel mir auf, dass ich den kleinen unscheinbaren Taster auf der Unterseite glatt vergessen hatte. Erst eine Grafik in der Anleitung hat mich draufgestoßen. Damit mir das nicht nochmal passiert, habe ich die vereinfacht beschriftete Abbildung von **Bild 8** gephotoshopt. Nun also geschaut, was man mit dieser unscheinbaren Trigger-Taste (unten in der Mitte rechts) machen kann. Ein Druck von 2 s wechselt vom Default-Mode (5 V „normal“) in die anderen Modi. Es sollte dann zunächst „-5.0“ zu lesen sein und man ist im Modus QC 2.0. Blinkt die rote LED, kann man mit den Tasten „+“ und „-“ die Spannung in den Schritten 5/9/12/15/20 V verstellen. Und mit Trigger gelangt man dann auch in die anderen Modi. Mangels passenden Ladegeräts konnte ich die Schnellladefunktionen der Modi AFC9 von Samsung und FCP9 von Huawei nicht testen.

Ohne Handbuch wäre ich nicht hinter die Modusumstellung gekommen. Die Bedienung dieser elektronischen USB-Last ist also (für mich) nicht gänzlich selbsterklärend. Wie ich mich kenne, nervt es mich, wenn ich in drei Monaten diese Last wieder verwenden will und gezwungen bin, die Anleitung zu bemühen, weil ich die Bedienung vergessen habe. Aber funktionieren tut es soweit, wenn man verstanden hat wie man das Gerät bedient.

### Fazit

Alles in allem ist diese preiswerte elektronische USB-Last funktional und tut genau das, was sie tun soll. Dass kein gedrucktes Handbuch beiliegt, ist eher positiv da umweltfreundlich und zeitgemäß. Dass das Handbuch trotz seiner Kürze von sieben Seiten vermeidbare Formulierungsschwächen und „Tippfehler“ aufweist, ist wohl dem Einsatz einer Übersetzungssoftware geschuldet. Gutes Deutsch wäre der Verständlichkeit förder-

lich. Wer gut Englisch spricht und jetzt denkt, er könne dann ja auf die englische Version des Handbuchs aufweichen, das ebenfalls auf der Joy-iT-Seite zur Verfügung steht, der wird enttäuscht, denn es ist nicht wirklich besser. Aber immerhin kann man verstehen, was der Autor wohl gemeint hat.

Verbesserungspotential gibt es auch bei der Anordnung vor allem des dritten Tasters „Trigger“. Ihn hinten unter der Platine zu verstecken mag dem Platinen-Layouter das Leben leichter gemacht haben und zu einer etwas kleineren Platine geführt haben, stört aber die Ergonomie. Um Trigger zu betätigen muss man das Modul mit der anderen Hand festhalten, sonst schiebt man es weg. Und wenn man wie ich gestrickt ist, muss man den Taster erst einmal finden ;-).

Alles in allem leistet diese USB-Last aber das, was sie verspricht. Sie ist preiswert und hält dank ausreichendem Lüfter auch längere Zeit eine relativ hohe Leistung aus. 30 W habe ich ausprobiert. Die maximale Leistung von 35 W ist für den Test von USB-Ausgängen aller Normen mehr als ausreichend. Wer eine solche elektronische Last also mehr als einmal im Leben benötigt, mag sie kaufen. Ein Selbstbau wäre mit Sicherheit nicht preiswerter. ◀

200049-01



**IM ELEKTOR-STORE**

→ USB-Last HD35 von Joy-iT

[www.elektor.de/joy-it-jt-hd35-usb-load-resistor-35-w](http://www.elektor.de/joy-it-jt-hd35-usb-load-resistor-35-w)

### Weblinks

[1] Review: USB-Tester UM25C von Joy-iT:

[www.elektormagazine.de/news/review-usb-tester-um25c-mit-lcd-farb-display-bluetooth](http://www.elektormagazine.de/news/review-usb-tester-um25c-mit-lcd-farb-display-bluetooth)

[2] Deutsche Betriebsanleitung zu HD35 von Joy-iT: [www.joy-it.net/files/files/Produkte/JT-HD35/JT-HD35-Anleitung.pdf](http://www.joy-it.net/files/files/Produkte/JT-HD35/JT-HD35-Anleitung.pdf)

# CAN-Bus plus Arduino: Überwachung von Solarmodulen

## Erkennen und Lokalisieren von problematischen Panels in großen Anordnungen

*Bild: Shutterstock*

Von **Somnath Bera** (Indien)

Warum den exquisiten CAN-Bus nur für die Sensorik im Fahrzeug verwenden? Das dachte sich der Autor, als Kunden, Installateure und Behörden sich über enttäuschende Ergebnisse eines kommerziell genutzten 25-kW-Solararrays stritten. So entwickelte er ein elektronisches System, das den Energieertrag jedes Solarpanels zusammen mit seiner individuellen Nummer ausliest, damit der Servicetechniker ein Problem schnell lokalisieren und beheben kann.

Bei der auf einem Dach montierten Photovoltaik-Anlage handelt es sich um ein Array von 55 einzelnen Paneelen, die jeweils 24 V und insgesamt bis zu 25 Kilowatt erzeugen können. Die Panels sind an einen großen einphasigen Wechselrichter mit 230-V-Ausgang angeschlossen, der seinerseits eine weiter entfernte große Batterieanordnung auflädt. „Konsument“ ist ein Konglomerat von Geschäften, das den Strom für die Beleuchtung in den Abendstunden und andere Aktivitäten benötigt. Die Geschäfte sollten sechs Stunden lang von 17:30 Uhr bis 23:30 Uhr beleuchtet werden.

### **Wasser ist zum Waschen da...**

Die Solaranlage, bestehend aus dem Photovoltaik-Array, dem Wechselrichter und dem Batteriesystem wurde mit großem Pomp und in Anwesenheit einiger hoher Tiere eingeweiht und funktionierte einige Zeit lang einwandfrei. Aber dann, langsam, über einige Monate hinweg, verschlechterte sich die Leistung so weit, dass selbst bei täglicher Sonnenscheindauer von zehn bis zwölf Stunden die Batteriefüllung am Abend nicht länger als für drei oder vier Stunden ausreichte.

Das allgemeine Hauen und Stechen begann: Die Geschäftsinhaber meckerten, die örtliche Gemeindeverwaltung machte die Installateure für die schlechte Qualität der Installation, billige PV-Produkte und so weiter verantwortlich. Natürlich verwahrten sich die erfahrenen Solartechniker gegen solcherlei Kritik, warfen einen kurzen Blick auf die auf drei Dächern montierten Solarpanels und enthüllten das Geheimnis der schwarzen Energie im Nu. Die 55 Paneele mit einer Nominalleistung von jeweils 500 Watt waren zur beliebten Spiel- und Ruhestätte sowie Start- und Landebahn einer großen Anzahl von Tauben und Krähen geworden, die die Oberflächen mit mehreren Lagen Vogelkot bedeckt hatten. Der Einkaufskomplex befindet sich

nämlich direkt neben dem lokalen Marktplatz, der ein magischer Anziehungspunkt für Vögel wie Krähen und Tauben zu sein scheint. Nach einer Grundreinigung waren innerhalb von etwa zwei Wochen die Moduloberflächen wieder vollständig von einer Lage Vogelkot abgeschattet.

Als Abhilfe schlug ich vor, kurzfristig alle zwei Tage abwechselnd die Hälfte der PV-Paneele mit Wasser zu waschen und als mittel- bis langfristige Maßnahme die Panels an einen Ort mit weniger Vögeln zu installieren. Durch die regelmäßige Reinigung wurde die Leistung wieder auf das ursprüngliche Niveau gebracht, und nach einer Weile stellte sich heraus, dass eine Reinigung alle zwei Tage nicht erforderlich war, sondern dass eine selektive Reinigung ausreichte, um die Leistung aufrechtzuerhalten. Dennoch musste die Person, die die Reinigung durchführte, zunächst eine Sichtkontrolle aller Paneele durchführen und dann erst mit der gezielten Reinigung fortfahren. Die Kontrolle nimmt aber viel Zeit in Anspruch, da der Arbeiter alle Paneele umrunden muss, um zu sehen, welche gereinigt werden müssen. Bei 55 beteiligten Solarpanels stellt das zwar noch kein großes Problem dar, aber für den nächsten Solarpark der Gemeindeverwaltung, der viermal so groß sein sollte, drohte der Reinigungsfirma ein Albtraum und das Personal mit Streik. Kurz gesagt wollte die Gemeindeverwaltung eine Einrichtung, die der Reinigungsfirma sofort mitteilen konnte, welche Panels der Anlage gereinigt werden müssen. So könnten die Reinigungskräfte sich direkt und ohne Zeitverlust die betroffenen Module unter den Wischmopp nehmen und die Oberflächenreinigung in kürzester Zeit durchführen.

### **Alles einsteigen: der CAN-Bus**

Es ist eine hervorragende Idee, mit dem preiswerten Arduino ein Control Area Network (CAN) aufzubauen, über das die aktuelle

Leistung der einzelnen PV-Paneele ermittelt werden kann. Der CAN-Bus wurde von der Firma Bosch für die Automobilindustrie erfunden und ermöglicht die Erfassung der Signale hunderter Sensoren, die in einem Fahrzeug verbaut und angeschlossen sind. Alle Sensoren sind mit nur zwei Leitungen in einer Busstruktur angeschlossen und mit dem Zentralrechner verbunden, der die Ausgangssignale der Sensoren überwacht und bei Bedarf durch die Ausgabe von Steuerbefehlen an relevante

Sensoren, wiederum über den Zweidraht-Bus, reagiert. Der CAN-Bus verläuft von Knoten zu Knoten, physikalisch als abgeschirmtes verdrehtes STP-Kabel (Shielded Twisted Pair) mit einer Impedanz von  $120\ \Omega$  (und für den Freilufteinsatz geeigneter guter Qualität und langer Lebensdauer). Für eine ordnungsgemäße Anpassung und zur Verringerung von Signalreflexionen sind an beiden Endknoten des CAN-Busses Abschlusswiderstände á  $120\ \Omega$  angeschlossen.

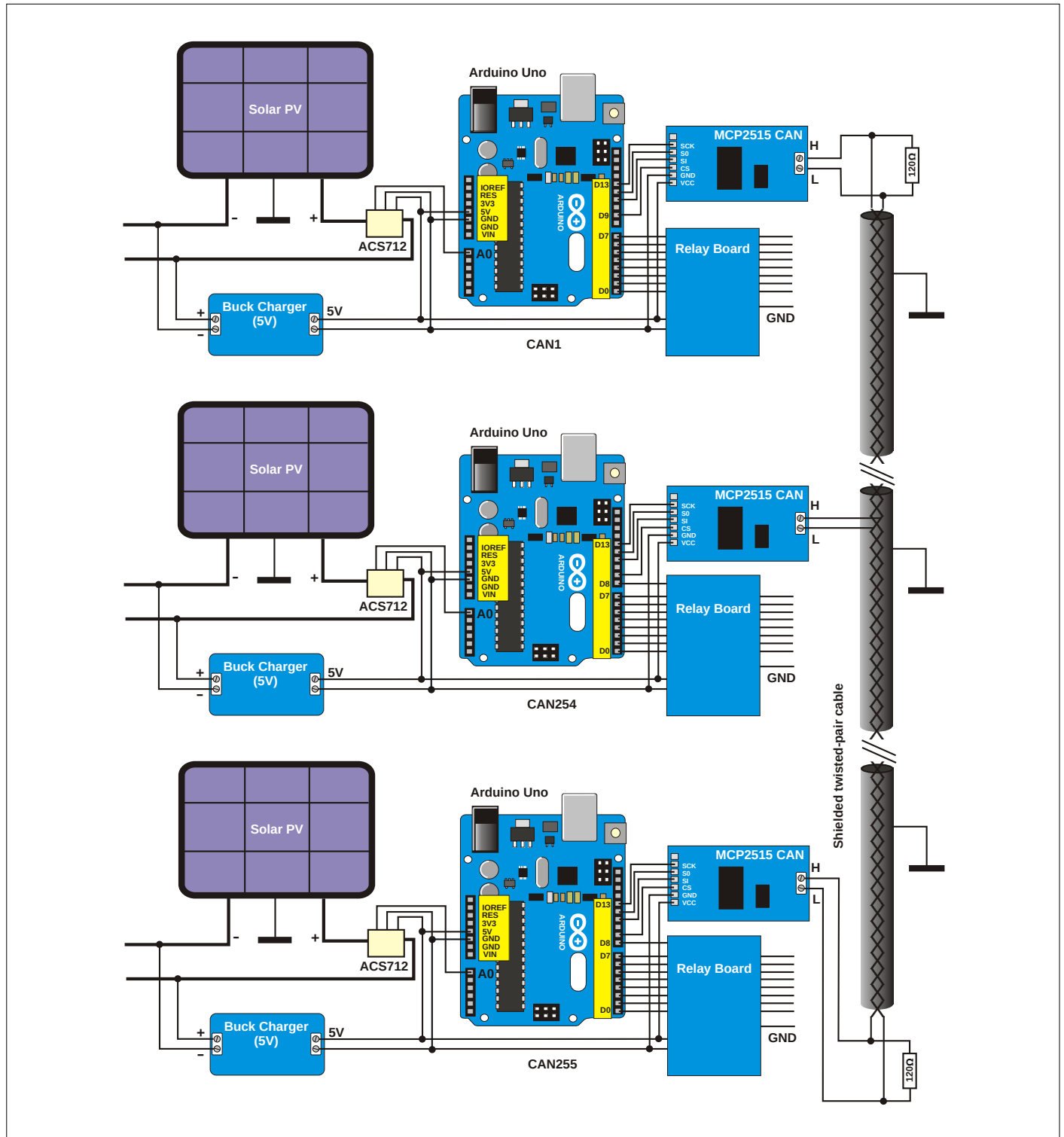


Bild 1. Blockdiagramm mit den Modulen, Bauteilen und Kabeln, die am CAN-Bus-basierten Überwachungssystem für ein Solarpaneel-Array mit bis zu 120 PV-Modulen (von denen hier nur drei dargestellt sind) beteiligt sind. Bei jedem Panel wird festgestellt, wenn etwas mit dem Ausgangsstrom nicht stimmt, unabhängig von der Ursache. Ein Arduino Uno ist der CAN-Bus-Master, die anderen sind Slaves.



Als CAN-Bus-Transceiver verwenden wir ein preiswertes Modul mit dem MCP2515, das leicht im Internet für zwei oder drei Euro erhältlich ist. Auf der Mikrocontroller-Seite besitzt es einen SPI-Anschluss, und auf der CAN-Bus-Seite eine Anschlussklemme für die Busleitungen CAN-H (high) und CAN-L (low). Als Mikrocontroller schlagen wir den ATmega328 auf einem Arduino Uno vor. Er hat die Aufgabe, das Sensorsignal (dazu kommen wir im nächsten Kapitel) an einem analogen Eingang zu empfangen, die Information auszuwerten/umzuwandeln und über seine digitalen I/O-Anschlüsse zum CAN-Bus-Transceiver und weiter auf den CAN-Bus zu übertragen. Zur Steuerung lokaler Aktoren (Relais, Servos und anderer Ausgabegeräte) stehen zudem acht weitere I/O-Leitungen zur Verfügung. Der Einfachheit halber verwenden wir nur CAN 2.0, das 0...8 Datenbytes gleichzeitig von einem Knoten zu einem anderen übertragen kann. Alle Knoten einschließlich des Masters erhalten einen Identifikator (ID 0 bis ID 255), der bei der Kommunikation gesendet und empfangen wird. Je niedriger der Identifikator, desto höher ist der Knoten in der Reihenfolge der sendenden und empfangenden Geräte. In der Regel erhält der Bus-Master den niedrigsten Identifikator und steht an höchster Stelle.

Das CAN-Bus-Protokoll verfügt über eine ausgezeichnete Fehlerprüfung. Die Anzahl der Knoten auf einem einzigen CAN-Bus kann zwischen 0 und 130 variieren, und auch die physikalische Länge variiert zwischen einigen Metern und etlichen Kilometern. Je größer die Entfernung zwischen den Endknoten, desto niedriger ist die erzielbare Geschwindigkeit der Datenübertragung. Für Buslängen von mehr als einigen hundert Metern sollte man CAN-Bus-Repeater verwenden, um Störfestigkeit und Signalintegrität zu gewährleisten. Die maximal zulässige Anzahl der Knoten auf einem typischen CAN-2.0-Bus liegt bei etwa 120, was immer noch weit über den 32 Knoten liegt, die eine Alternative wie der RS485-MODBUS bietet.

| Busgeschwindigkeit<br>[kbit/s] | Buslänge | Stichleitungs-<br>länge |
|--------------------------------|----------|-------------------------|
| 1000                           | 40 m     | 0,3 m                   |
| 500                            | 110 m    | 0,3 m                   |
| 250                            | 240 m    | 0,3 m                   |
| 125                            | 500 m    | 0,3 m                   |
| 100                            | 500 m    | 0,3 m                   |
| 50                             | 1300 m   | 0,3 m                   |
| 20                             | 3300 m   | 0,3 m                   |
| 10                             | 6600 m   | 0,3 m                   |
| 5                              | 30 km    | 0,3 m                   |

### Verschmutzungsmonitor für Solarpaneele

Mit Hilfe des CAN-Busses können also bis zu 120 Solarpaneele zusammengefasst und ihre individuellen Leistungen erfasst werden, also die Spannung und der Strom, die jedes einzelne Panel zu der in den Wechselrichter eingespeisten Gesamtenergie beiträgt. Ein Blockschaltbild für die Leistungsüberwachung in einem PV-Panelarray ist in **Bild 1** dargestellt.

Durch die Parallelschaltung weisen alle Solarpanels im Array die gleiche Klemmenspannung, aber unterschiedliche Ausgangsströme auf. Daher reicht es, einen einfachen Stromsensor in die Anschlussleitung jedes PV-Panels einzuschleifen. Dieser Stromsensor mit der Bezeichnung ACS712 basiert auf dem Hall-Effekt und kann (in der Version -30) einen Strom bis zu

30 A messen. Ähnlich wie der MCP2515 befindet sich der Sensor auf einem Breakout-Board (Preis etwa 2 €) und kann einfach über eine Platinenklemme in den Leistungsstromkreis eingeschleift werden. Auf der anderen Seite des Boards befindet sich ein dreipoliger Pfostenverbinder mit Spannungsversorgung und Analogausgang (anzuschließen an A0 des Arduino). Im Sketch wird der Sensor einfach mit

```
load_current = 0.075714*(analogRead(A0) - 512)
```

gelesen.

Die 5-V-Versorgung des Sensors und auch des Unos erfolgt über ein kleines Buck-Ladegerät, das an die 24-V-Leitungen angeschlossen ist. Der größte Vorteil eines Ladegeräts, das ebenfalls mit nicht mehr als 2 € pro Stück zu Buche schlagen sollte, liegt in seiner sehr geringen Verlustleistung.

Neben Vogelkot, Schmutz, Ruß und Pollen können weitere Faktoren die Leistung der Solarmodule beeinträchtigen: mangelnde Feuchtigkeitsbeständigkeit (kleine Blasenbildung auf der Oberfläche), so genannte Snail Trails (Verfärbung der Oberfläche), Mikrorisse und Hot Spots (Überhitzung einzelner Sektoren). All diese Faktoren können zu einer Minderung der Ausgangsleistung um bis zu 35% führen. Da wir nicht sehr an den Absolutwerten interessiert sind, können wir die Programmzeile vereinfachen in:

```
load_current = analogRead(A0) - 512
```

Dieser Wert `load_current` wird mit einem festgelegten Referenzwert sowie den Werten anderer Panels verglichen, um zu entscheiden, ob ein Alarm ausgelöst werden soll oder nicht. Nachdem die Reinigungsperson den Alarm gesehen und das betroffene Panel (die ID) identifiziert hat, geht sie dorthin, prüft es auf Staub, Schmutz und Pollen und reinigt es. Sollte das Problem weiter bestehen, muss das Panel auf Hot Spots oder andere der oben genannten Fehler überprüft und im schlimmsten Fall ersetzt werden.

### CAN-Bus 4x4

Das Projekt ermöglicht es, eine 4\*4-Tastatur an einen Arduino anzuschließen, um lokal Steuerbefehle am CAN-Bus zu erteilen.

**Bild 2** zeigt die Blockschaltung, **Bild 3** einen Versuchsaufbau auf der Werkbank des Autors. Die Tastatursteuerung ist (optional) in der Projektsoftware eingebunden.

### Die Projektsoftware

Die für das Projekt entwickelte Software ist ein Bündel von Arduino-Sketches, das von einigen Updates und Notizen des Autors begleitet von der Elektor-Labs-Website [1] heruntergeladen werden kann. Die Software soll als Ausgangspunkt verstanden werden, nicht als fertige, glatt laufende Anwendung. Sie ist im Großen und Ganzen in zwei Teile gegliedert: Master und Slave. Ein Slave nimmt den Messwert am A0-Port in Empfang und leitet ihn (digitalisiert) an den Master weiter. Der Master kennt zwei Betriebsarten: Anzeigemodus oder Steuermodus. Der CAN-Master erhält die ID 1, während den anderen Einheiten die ID 254 und ID 255 wie in Bild 1 oder eine andere ID zwischen 2 und 253 zugewiesen wird. Die maximale Anzahl liegt bei 120 Knoten.

**Master im Anzeigemodus:** Der Modus wird durch den Pegel

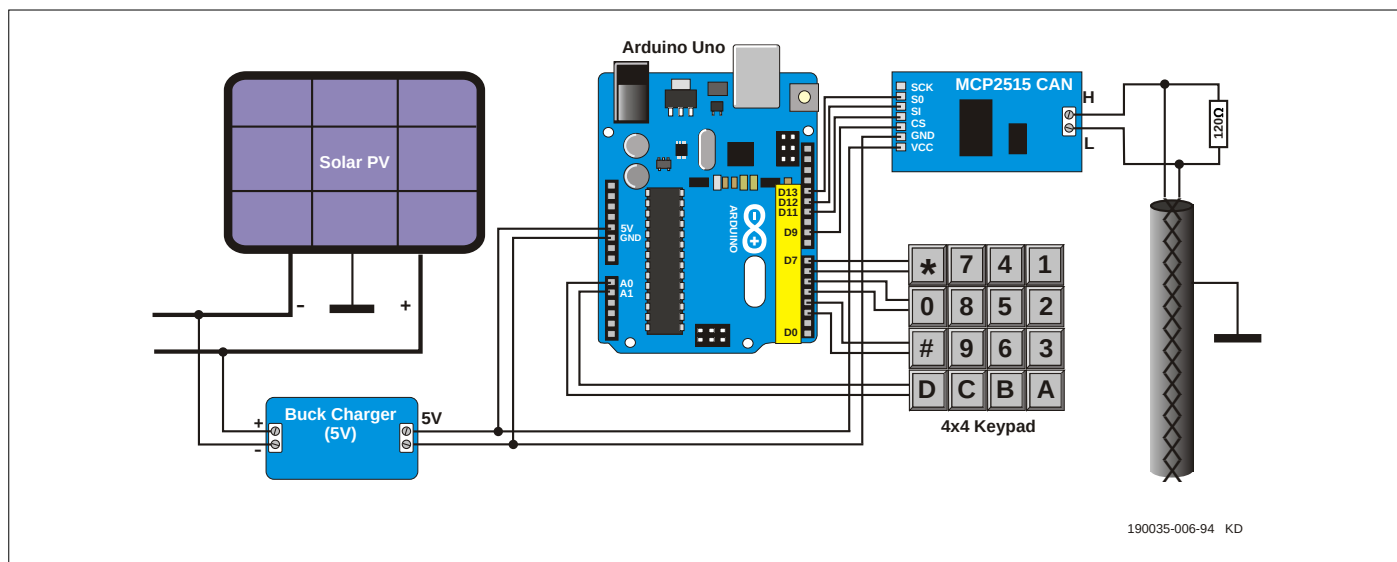


Bild 2. Blockschaltung des CAN-Bus-Masters mit einer 4x4-Tastatur für den Einsatz im Steuerungsmodus.

am I/O-Pin D10 eingestellt. Wenn D10 high ist, geht der Master in den Anzeigemodus, sammelt Daten von allen Slaves und zeigt sie an. Er stellt dann fest, ob ein Signal (das heißt, die gemessene Leistung eines Panels) unter dem gewünschten Wert liegt und gibt entsprechend einen Alarm aus. Dieser Modus ist optimal für die Überwachung von PV-Anlagen.

**Master im Steuermodus:** Wenn D10 offen oder low ist, geht der Master in den Steuermodus. Für diesen Fall sind an den Slave-Arduinos (D0 bis D8) Relais, am Master (ID 1) die besagte 4x4-Tastatur angeschlossen. Der Master gibt eine Befehlszeichenfolge aus wie

25581#,25580#,13011#

und die jeweiligen Knotenrelais werden ein- oder ausgeschaltet, um zum Beispiel eine Dusche und/oder einen an den PV-Panels applizierten Wischer zu steuern. Beispiele:

25580#: 255 ist die Knoten-ID, 8 ist die digitale I/O-Nummer, 0 ist aus; 1 ist an.

25471#: 254. Knoten, Relais an I/O 7, 1 = einschalten.

25130#: 251. Knoten, Relais an I/O 3, ausschalten.

Ein Doppelkreuz (#) schließt einen Befehl ab.

Der CAN-Bus-Master kann auch einen PV-Stromsensor besitzen, der am A0-Pin angeschlossen ist, oder kann „solo“ ohne Panel und damit auch ohne Stromsensor betrieben werden.

### Ihre Anwendung?

Eine alternative Anwendung dieser CAN-Bus-Konfiguration wäre, in einer kleinen Stadt oder einem Park zu überprüfen, ob Lichtmasten tatsächlich ein- oder ausgeschaltet sind. Ein

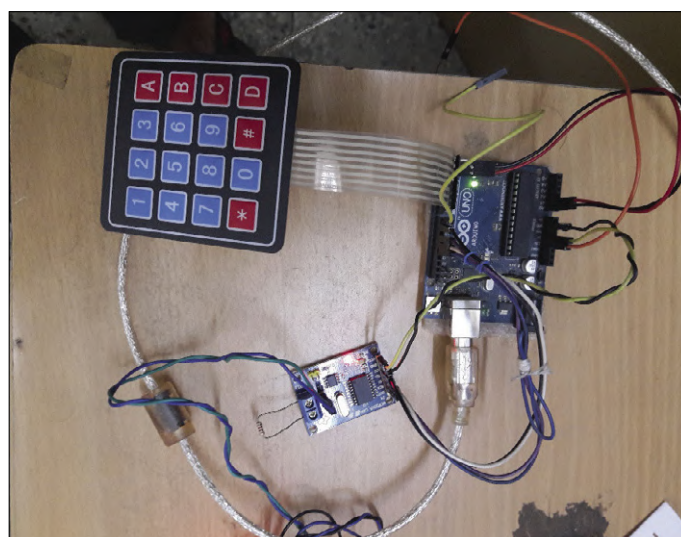


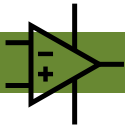
Bild 3. Experimentelle Version des CAN-Bus-Mastercontrollers, erweitert um ein 4x4-Tastenfeld.

System wie das hier beschriebene ermöglicht es auch, viele kleine Einheiten, die an den CAN-Bus angeschlossen sind, von einem zentralen Schreibtisch aus zu steuern. Weitere Anwendungen sind Hotelzimmer-Alarmknöpfe für Gäste und für die Beleuchtung von Hotels. Anstatt alle Arten von Kabeln innerhalb des Gebäudes zu verlegen, reicht ein einziger verdrehter CAN-Bus für eine umfassende, zuverlässige Steuerung und Prüfung aus. ◀

190035-03

### Weblink

[1] Projekt auf Elektor-Labs: <https://www.elektormagazine.com/labs/can-bus-for-solar-pv-cell-monitoring>



# Entwicklung analoger Elektronik

## Fall Nr. 1 – Teil 3: Optimierung des Vorverstärkerverhaltens und Kompromisse

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

In dieser Artikelserie beleuchten Experten spezielle Aspekte der analogen Elektronik-Entwicklung, die einem zunehmend „rein digitalen“ Publikum zugute kommen. In der vorhergehenden Folge wurde ein Vorverstärker für das MEMS-Mikrofon aus dem Fledermaus-Detektor<sup>PLUS</sup> entwickelt.

Jetzt konzentrieren wir uns auf die Merkmale, die ihn von einem üblichen, mit Operationsverstärkern aufgebauten Audio-Vorverstärker unterscheiden.

Wenn Sie neu in dieser Artikelreihe sind, sollten Sie zunächst die vorherigen Folgen [1][2] studieren. Die Schaltung des Mikrofonvorverstärkers, die wir gleich besprechen werden, wurde aus dem vollständigen Schaltplan des Fledermaus-Detektor<sup>PLUS</sup> extrahiert und ist in **Bild 1** separat dargestellt.

### Dynamischer Bereich

Aufgrund der sehr hohen festen Verstärkung und der niedrigen Versorgungsspannung des Vorverstärkers besteht die Möglichkeit, dass der Ausgang in die Sättigung gerät und der Verstärker „clippt“. Wenn die Signale zu klein und/oder frequenzabhängig sind, kann wiederum die Verstärkung nicht ausreichend sein. Der Dynamikbereich wird durch das maximale unverzerrte Ausgangssignal und das kleinste nutzbare Signal oberhalb des Rauschpegels bestimmt. Im Gegensatz zu ADCs und DACs bedeutet aber  $S/N$  nicht dasselbe wie Dynamikbereich. Das  $S/N$ -Verhältnis von Mikrofonen wird normalerweise bei 94 dB Schalldruck (1 Pascal) spezifiziert, aber bei den meisten Mikrofonen besteht ein linearer Zusammenhang zwischen Schalldruck und Ausgangsspannung, normalerweise bis zu 120 dB. Die Verzerrungen sind dann immer noch minimal, so dass der Dynamikbereich wesentlich größer als das spezifizierte  $S/N$ -Verhältnis ist. Bei einer Anwendung wie dem Fledermausdetektor ist es offensichtlich sinnvoll, eine schnelle, automatische Verstärkungsregelung, die an das Quellsignal angepasst (und vielleicht sogar begrenzt) ist, hinzuzufügen, um diesen Bereich weiter zu vergrößern.

Dies könnte ganz einfach zu realisieren sein, zum Beispiel über die klassische Schaltung mit einem JFET als variablem Widerstand, was auch die Detektion der Modulation verbessern würde. Es besteht kaum die Gefahr von Rückkopplungen (Heulen), wenn die hohen Frequenzen am Leistungsverstärker richtig herausgefiltert werden, so dass sie nicht im Lautsprecher zu hören sind (sollte dies überhaupt ein Problem darstellen). Leider kann es trotz der Frequenzfilterung sowohl durch Oberwellen im Ausgangssignal des Leistungsverstärkers als

auch bei der Wiedergabe durch den Lautsprecher zu Rückkopplungen vom Lautsprecher zum Mikrofon kommen. Dies ist ein mechanisches Problem, denn selbst der beste Lautsprecher ist nicht ideal und verzerrt den Klang. Im Fall des Fledermausdetektors gibt es natürlich keine Anforderungen an wirklich kleine Signale wie bei einem Audioverstärker. Ein Signal-Rausch-Abstand von 20 dB im Signal zum Komparator dürfte ausreichen.

### Eingangsruhestrome

Auf den ersten Blick mag es verwundern, dass an jedem nicht-invertierenden Eingang ein Widerstand von 10 k $\Omega$  nach Masse geschaltet ist. Dies wird normalerweise getan, um den zusätzlichen Offset zu kompensieren, der durch die Eingangsruhestrome der Operationsverstärker verursacht wird. Aber selbst im maximalen Temperaturbereich eines TL06x sind diese Ströme durch die JFET-Eingänge kleiner als 5 nA. Diese 5 nA über einen Widerstand von 100 k $\Omega$  verursachen nur winzige 0,5 mV. Dies ist fast vernachlässigbar, auch weil es keine Gleichstromkopplung zwischen den verschiedenen Verstärkerstufen gibt. Wenn aber bipolare Operationsverstärker eingesetzt würden, können diese Eingangsruhestrome einige 100 nA betragen oder gar im Mikroamperebereich liegen. Erst dann muss man sie berücksichtigen, natürlich abhängig von den Widerstandswerten. Außerdem machen die 10-k $\Omega$ -Widerstände die Verstärkerstufen empfindlicher gegen Störungen und unerwünschte parasitäre Rückkopplungen. Man sollte sie eventuell durch Drahtbrücken ersetzen.

### Operationsverstärker als Komparator

Die Schaltung verwendet auch einen halben TL062 als Komparator. Dies ist verlockend, wenn noch ein freier Operationsverstärker vorhanden ist oder ein Dual- anstatt eines Single-Opamps verwendet wird. Ein „echter“ Komparator würde dagegen ein zusätzliches IC erfordern. Echte Komparatoren besitzen aber üblicherweise einen Open-Collector-Ausgang, was nicht





immer von Vorteil ist. Zunächst muss man darauf achten, dass der maximale differentielle Eingangsspannungsbereich ausreicht, und das ist hier glücklicherweise der Fall. Dasselbe gilt für Operationsverstärker mit CMOS- und JFET-Eingängen. Bei bipolaren Operationsverstärkern, wie es Audio-Operationsverstärker üblicherweise sind, wird dieser Bereich oft durch ein zusätzliches Netzwerk zum Schutz der Eingangstransistoren begrenzt, zum Beispiel durch zwei antiparallele Dioden oder eine Reihenschaltung zweier Zenerdioden. Außerdem kann die Ausgangsspannung bei Überschreitung der Maximalwerte kippen. Ein Problem beim TL062 ist sein maximaler Eingangsoffset, der bis zu 15 mV beträgt, sogar bis zu 20 mV im gesamten Temperaturbereich. Ein angelegtes Signal muss diesen Offset übertreffen, bevor der Ausgang umgeschaltet wird. Im Falle des Detektors hängt die Empfindlichkeit vom verwendeten IC-Typ ab, und dies ist eine Abhängigkeit, die wir nicht wollen.

Ein weiteres Problem ist die differentielle Spannungsverstärkung, die laut Datenblatt minimal 3000 und typischerweise 6000 beträgt. Das ist eigentlich zu wenig, damit das IC als Komparator richtig funktionieren kann. Um korrekt zu arbeiten, ist mindestens das zehnfache, besser das 30-fache dieser Verstärkung erforderlich. Bei einem echten Komparator wie dem LM311 oder dem LM339 beträgt die Verstärkung typischerweise 200.000. Die Auswirkung der Versorgungsspannung auf die Differenzverstärkung ist in den Datenblättern nicht spezifiziert, aber es wäre interessant zu wissen, ob es angesichts der eher geringen  $\pm 4,5$  V der 9-V-Batterie eine gibt. Um ordentliche Rechtecksignale mit Frequenzen von bis zu 120 kHz zu erzeugen, ist mehr Bandbreite und sicherlich auch eine höhere Anstiegsrate erforderlich. Bei einem Komparator wie dem LM311 hängt die Anstiegsrate (Slew rate) vom Differenzsignal und faktisch von der Übersteuerung ab. Je größer dieses Eingangssignal ist, desto schneller ändert sich die Ausgangsspannung, aber selbst bei 2 mV lassen sich etwa 30 V/ $\mu$ s erreichen, im Wesentlichen wegen des Ausgangstransistors und nicht wegen eines Pull-Down- oder Pull-Up-Widerstandes. Im Gegensatz dazu ist die Anstiegsrate von 3,5 V/ $\mu$ s, die das TL062 in unserer Schaltung bietet, etwas mager.

### Elektret mit zwei oder drei Anschlüssen?

Vielen dürften Elektret-Mikrofonkapseln ein Begriff sein. Die meisten Typen besitzen nur zwei Anschlüsse. Einer, der Masseanschluss (GND), ist mit dem Metallgehäuse (Schirm) verbunden, der andere führt das Ausgangssignal, das vom Drain (d) eines internen JFETs stammt. Die Source (s) des JFETs ist mit Masse verbunden. Eine gepolte Membran wandelt den Schall in eine Spannung für das Gate (g) des JFETs um.

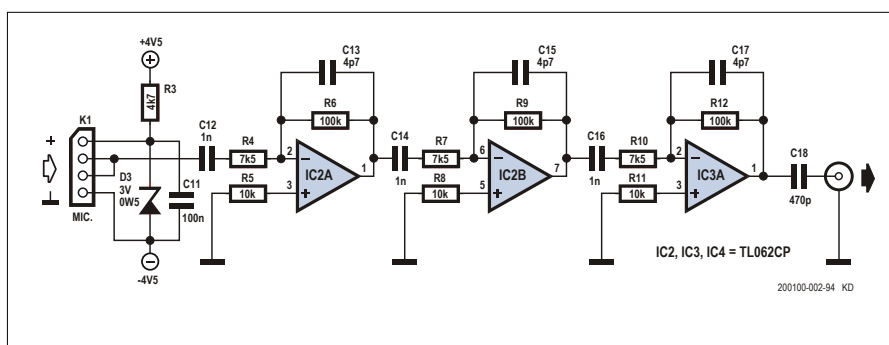


Bild 1. Der dreistufige Mikrofonvorverstärker im Fledermaus-Detektor<sup>PLUS</sup> zeigt einige bemerkenswerte Eigenschaften.

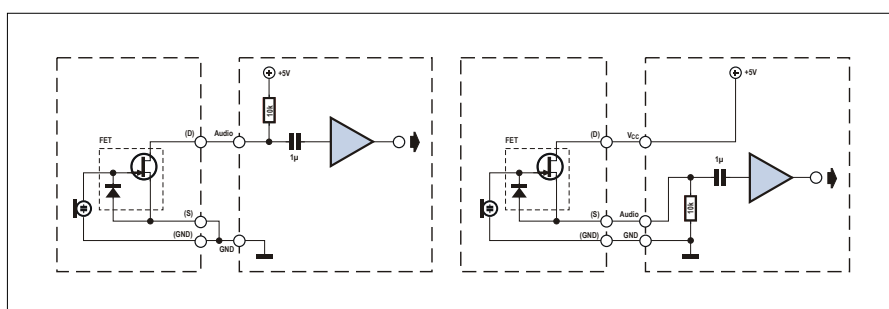


Bild 2. Anschlussschaltung eines zweipoligen Elektretmikrofons (links) und einer dreipoligen Version (rechts). Bei einem zweipoligen Elektretmikro sind FET-Source und Masse bereits intern verbunden.

Durch das Einfügen eines Pull-up-Widerstandes zwischen dem Eingang des Mikrofonverstärkers und der Versorgungsspannung (die normalerweise recht niedrig ist, zwischen 1,5 V und 5 V) wird der FET vorgespannt. Eine gewisse Verstärkung in der Größenordnung von 10 dB findet so bereits im Inneren des Elektretmikrofons statt. Der Nachteil dieser Verstärkung ist aber die nichtlineare Steilheit eines FETs. Die Steilheit oder Transkonduktanz  $g_m$  bezeichnet die Beziehung zwischen der Änderung des Drainstroms als Funktion der Änderung der Gate-Spannung. Da diese Beziehung nichtlinear ist, weicht das Ausgangssignal von der Spannung an der Membran ab, was zu einer leichten Verzerrung führt.

Bei kleinen Signalen ist dieser Effekt kaum wahrnehmbar, aber mit zunehmendem Schalldruck steigt die Verzerrung an und wird sogar auf einem Oszilloskop sichtbar. Auch kann die Phase des Ausgangssignals invertieren, aber das ist normalerweise kein Problem, weil es durch eine Umkehrung der Polarisierung der Membran gelöst werden kann.

Bei dreipoligen Elektretmikrofonen steht der Sourceanschluss des FETs am dritten Beinchen zur Verfügung, wie man in **Bild 2** sehen kann. Es handelt sich um Mikrofone ohne eingebauten Verstärker, der eine echte Versorgungsspannung benötigen würde; hier reicht eine Vorspannung aus. Mit Hilfe dieses dritten Anschlusses kann der FET als Source-Follower konfiguriert werden, wodurch die Verzerrung reduziert wird. Leider liegt das Ausgangssignal in dieser Konfiguration niedriger. Der Drain-Anschluss ist direkt mit der positiven Stromversorgung

verbunden und der Source-Anschluss ist auf Masse gelegt. Der Wert des Pull-up-Widerstandes in der zweipoligen und des Pull-down-Widerstandes in der dreipoligen Variante hängt natürlich vom Typ des Elektretmikrofons und der Versorgungsspannung ab. Um den Wert zu bestimmen, muss das Datenblatt konsultiert werden. MEMS-Mikrofone besitzen immer ein eingebautes IC, so dass eine echte Versorgungsspannung erforderlich ist. Es gibt Tricks, um eine Zweidraht-Verbindung zu einem MEMS-Mikrofon herzustellen, indem man eine Last am Ausgang platziert, um den Versorgungsstrom zu variieren, aber was dann mit den im Datenblatt angegebenen Eigenschaften passiert, ist reine Spekulation.

### Berechnung des Eingangsrauschens

Bei der Berechnung des Rauschens muss von einer Bandbreite von 120 kHz ausgegangen werden, statt von dem Audio-Standardbereich von 20 kHz. Schlimmer noch, das Rauschen wird oft nur A-bewertet im Datenblatt angegeben. Die bekannte Formel, die das Rauschen als Funktion der Temperatur, des Widerstands und der Bandbreite darstellt, besagt

$$U_{\text{noise}} = \sqrt{4 k T B R} \quad (\text{Gleichung 1})$$

$k$  = Boltzmann-Konstante ( $1,38 \times 10^{-23} \text{ J K}^{-1}$ )  
 $T$  = absolute Temperatur (in K, hier wird 270 K angenommen)  
 $B$  = anwendbare Bandbreite (in Hz)  
 $R$  = Widerstand (in  $\Omega$ )

Das Gesamtuschen am Eingang setzt sich aus dem bereits im Quellsignal vorhandenen Rauschen ( $U_{\text{noise1}}$ ), dem Widerstandsrauschen ( $U_{\text{noise2}}$ ) und dem vom verwendeten Operationsverstärkers ( $U_{\text{noise3}}$ ) erzeugten Rauschen zusammen.

$$U_{\text{noise(total)}} = \sqrt{(U_{\text{noise1}})^2 + (U_{\text{noise2}})^2 + (U_{\text{noise3}})^2} \quad (\text{Gleichung 2})$$

Laut Datenblatt wurde das Rauschen des von uns verwendeten Mikrofons mit einem A-gewichteten Filter gemessen. Dieses Filter ist leider nicht für Frequenzen oberhalb von 20 kHz spezifiziert. Gerade der Ultraschallbereich ist es jedoch, der uns bei einem Fledermaus-Detektor interessiert. Es wird sogar empfohlen, bei dem Schalldruckmessgerät ein steilflankiges 20-kHz-Filter mit dem A-bewerteten Filter zu kombinieren. Dies macht die Aussage im Datenblatt für uns erst recht unbrauchbar. Sicher ist nur, dass das Rauschen in unseren Berechnungen wesentlich größer sein dürfte. Wir können also nur eine Annahme treffen.

Wenn wir das A-bewertete Filter mit einem 10 kHz-Tiefpassfilter erster Ordnung vergleichen, dann ist das Rauschen für unsere Berechnung bei einer Bandbreite von 120 kHz grob geschätzt mindestens um das  $\sqrt{(120 / 10)} = 3,5$ -fache, also 11 dB höher als im Datenblatt angegeben. Daher wird für einen angegebenen Signal-Rausch-Abstand von 59 dB(A) bei einem Schalldruckpegel von 94 dB (1 Pascal, Pa) und einer Bandbreite von 120 kHz dann 48 dB geschätzt, wenn die Messung ungefiltert erfolgt. Bei diesem Schalldruck wird ein Ausgangspegel von -42 dBV für 1 kHz, also 8 mV, angegeben. Bei 1845-facher Verstärkung ergibt das 14,76 V, was den Verstärker gnadenlos übersteuern würde. Aber bei Ultraschall oberhalb von 40 kHz sinkt der Ausgangspegel unseres MEMS-Mikrofons, während er zwischen 10 kHz und 20 kHz sogar mehr als 10 dB höher ist, so dass der Grad der Übersteuerung stark frequenzabhängig

und damit fast unvorhersehbar ist. Eine Berechnung auf der Grundlage eines linearen Rauschverhaltens ist zwar eigentlich nicht genau das, was wir für den Betrieb des Detektors benötigen, aber wir kommen mit der Schätzung aus.

Natürlich wäre es besser, den Frequenzgang des Verstärkers an das Mikrofon anzupassen. Wie sollte die Berechnung für das Rauschen aber dann aussehen, besonders, wenn die Option bestehen sollte, Mikrofone auszutauschen? Dann würde nur eine digitale Lösung in Frage kommen. Aber zurück zu unserer Kalkulation. Das Rauschen des Mikrofons bei einem Schalldruck von 94 dB beträgt:

$$8 / (10^{(48/20)}) \times (100 / 7,5) = 0,425 \text{ mV}$$

Für den Anteil des resistiven Rauschens ist die Parallelschaltung der beiden verstärkungsbestimmenden Widerstände von 100 k $\Omega$  und 7,5 k $\Omega$  maßgeblich. Den 10-k $\Omega$ -Widerstand im nicht-invertierenden Eingangspfad lassen wir aus den Betrachtungen heraus und ersetzen ihn in durch eine Drahtbrücke:

$$\sqrt{(4 \times 1,38 \times 10^{-23} \times 270 \times 120000 \times 6977) \times (100 / 7,5)} = 0,047 \text{ mV}$$

Das Rauschen des Operationsverstärkers wird wie bei einem nicht-invertierenden Verstärker verstärkt. Das Rauschen eines TL06x beträgt typischerweise 42 nV/ $\sqrt{\text{Hz}}$ :

$$(42 \text{ nV} \times \sqrt{120000}) \times (1 + 100 / 7,5) = 0,21 \text{ mV}$$

Der Gesamtuschenpegel am Ausgang der ersten Verstärkerstufe im Fledermausdetektor beträgt

$$\sqrt{(0,425^2 + 0,047^2 + 0,21^2)} = 0,476 \text{ mV}$$

was ein Signal-Rausch-Verhältnis (S/N) von

$$(100/7,5) \times 8 \text{ mV} / 0,476 \text{ mV} = 224,1$$

ergibt oder solide 47 dB, wenn man es in Dezibel ausdrücken möchte. Mit anderen Worten, das Rauschen am Ausgang des Mikrofonverstärkers wird fast ausschließlich durch das MEMS-Mikrofon bestimmt. Der Beitrag der folgenden Stufen ist bei ausreichender Verstärkung vernachlässigbar. Wenn die Verstärkung pro Sektion abnimmt, haben auch die folgenden Verstärkerstufen einen Einfluss. Dies ist ein weiterer Grund, warum man mit acht Sektionen mit geringer Verstärkung schlecht beraten ist. Und auch hier gilt: Diese Berechnung ist nur unter der Annahme gültig, dass das Rauschen linear verteilt ist und nicht annähernd der Frequenzcharakteristik folgt.

Wie würde die praktische Anwendung eines MEMS-Mikrofons aussehen, wenn es direkt auf der Platine montiert wäre? Wie würden Sie Ihren eigenen Testaufbau erstellen, um die tatsächliche Frequenzcharakteristik mit beliebiger Genauigkeit zu bestimmen?

Mit dieser Herausforderung schließen wir den Fall 1 der Artikelreihe „Entwicklung analoger Elektronik“ ab. In der nächsten Elektor-Ausgabe werden wir ein weiteres Thema auf völlig analoge Weise durchleuchten. 

200100-03



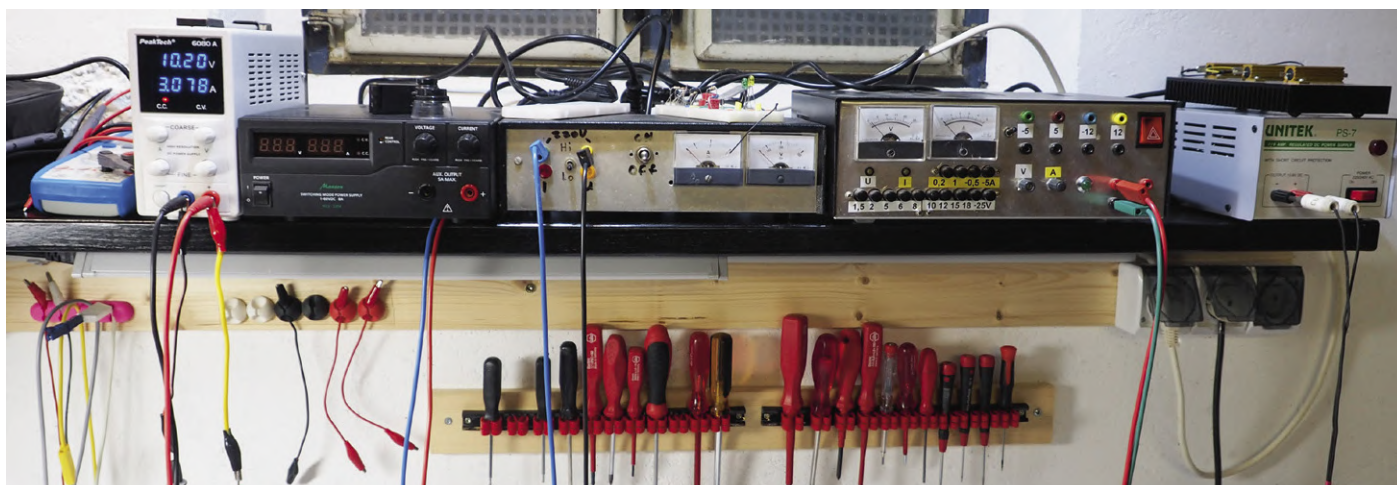
## REVIEW

# Labornetzteil PeakTech 6080 A

## Viel Netzteil für wenig Geld

Von Dr. Thomas Scherer

Netzteile kann der Elektroniker bekanntlich nie genug haben. Das gilt auch für mich, wie Sie am Foto aus meinem privaten Labor sehen können. Das kleine Labornetzteil PeakTech 6080 A ist – für heutige Zeiten erstaunlich – analog. Für kleines Geld bekommt man hier viel Genauigkeit.



Eigentlich ist mein kleines Elektronik-Labor ja schon ganz gut mit Netzteilen bestückt. Ein 12-V-Netzteil mit max. 9 A Belastbarkeit (in Laborbild ganz rechts) eignet sich gut für die Versorgung kleiner Bohr-/Schleifmaschinen von Dremel, Proxxon und Co. Daneben sieht man den Eigenbau des Elektor-Präzisions-Netzgeräts [1] von 1980! Es ist 40 Jahre alt und immer noch 0,1% genau. Dann folgt eine relativ hochohmige Netzspan-

nungsquelle mit Sicherheitsisolierung und etwa 60 W Belastbarkeit, die zwar nicht schön, aber praktisch und lebensrettend ist. Viel Power liefert das digitale Labornetzteil mit maximal 60 V und 8 A – ausnahmsweise nicht selbst gebaut. Nun hat sich das PeakTech 6080 A hinzugesellt, das zwar lediglich maximal 15 V und 3 A liefert, dafür aber mit anderen Features glänzt.

### Allgemeines

Das PT 6080 A erreichte mich nicht per Post, es wurde per Anruf angekündigt. Ein Nachbar einige Straßen weiter rief mich einen Tag vor Weihnachten an und meinte, seine Eltern hätten gesehen, wie aus einem Post-Transporter Pakete gefallen wären. Sie hätten angehalten und das mitgenommen, auf dem mein Name stand. Ich holte es gleich ab und der Postbote war happy, als er erfuhr, dass das Paket doch bei mir gelandet ist...

Wenn Sie sich nun fragen, was an dieser Anekdote relevant ist: Das Labornetzteil ist sehr gut verpackt und steckt im Karton in dicken Schaumstoffhaltern (**Bild 1**). Deshalb hat es die Fahrt zu mir ohne Blessuren und voll funktionsfähig überlebt.

Mit 80 x 160 x 225 mm (B x H x T) ist es vertikal statt wie üblich horizontal gebaut (**Bild 2**), was die nötige Stellfläche minimiert. Auffällig ist der dicke, fette Kühlkörper auf der Rückseite. Er ist das erste Indiz dafür, dass es sich hier um ein Netzteil mit analoger Regelung handelt, das bei kleinen Ausgangsspannungen



Bild 1. Das Labornetzteil ist durch dicke Schaumstoffhalter im Karton geschützt.





Bild 2. So sieht das PT 6080 A von vorne und von hinten aus (Bild: PeakTech).

relativ viel Leistung verheizen muss. Das zweite Indiz ist das Gewicht: Mit rund 2 kg ist es für 45 W Ausgangsleistung nicht gerade leicht. Da muss also ein ordinärer Trafo drin sein und kein kleiner HF-Übertrager, wie das bei Schaltnetzteilen der Fall ist. Das Labornetzteil stellt an seinen beiden Sicherheits-Bananenbuchsen eine von 0 bis 15 V einstellbare Ausgangsspannung zur Verfügung. Die Strombegrenzung ist von 0 bis 3 A einstellbar. Hierfür sind vier Drehknöpfe vorgesehen – je zwei sind für die Grob- und Feineinstellung von Spannung (rechts) und Strom (links) zuständig. Außerdem sind noch zwei vierstellige Anzeigen für Spannung und Strom und zwei LEDs vorhanden, die mit „C.C.“ und „C.V.“ bezeichnet sind. In Bild 2 leuchtet die CV-LED: Das Netzteil ist also gerade im Modus Konstantspannung (Constant Voltage). Wird der Strom begrenzt bzw. „geregelt“, leuchtet stattdessen die CC-LED (Constant Current).

### Genauigkeit

Was nicht nur Sie, sondern auch mich am meisten interessiert, ist die Genauigkeit der Anzeige. Bei meinem digitalen Power-Netzteil wird die Spannung und der Strom mit Hilfe zweier Drehgeber digital eingestellt. Diese kann man üblicherweise durch Druck auf den Knopf zwischen Grob und Feineinstellung im Verhältnis 1:10 umschalten. Leider zeigen digitale Netzteile genau das an, was eingestellt wurde, und leider nicht das, was an den Buchsen herauskommt. Folglich gibt es immer kleine Abweichungen im Bereich von 100 mV bzw. einigen 10 mA. Mit anderen Worten: Vor allem preiswertere digitale Netzteile sparen sich oft einen Digitalvoltmeter-Chip und zeigen daher weniger genau an, als es wünschenswert wäre.

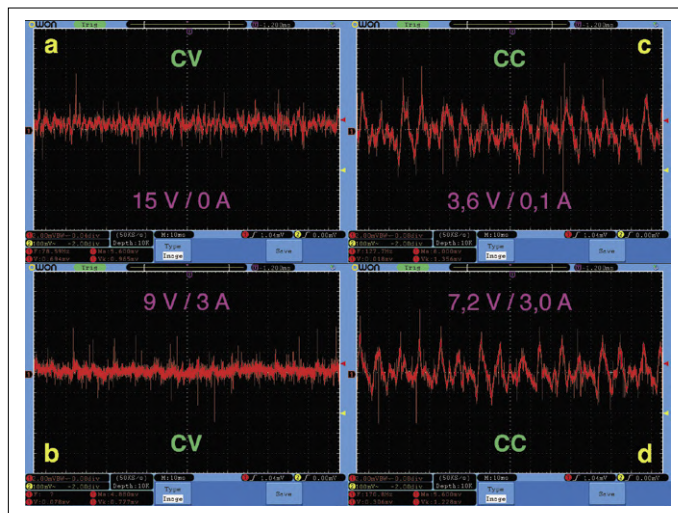


Bild 5. Oszilloskopierte Ausgangsspannung bei verschiedenen Bedingungen.

Beim PT 6080 A ist das anders. Statt Drehgeber sind hier normale Potis verbaut. Das „Übersetzungsverhältnis“ von Grob- zur Feineinstellung beträgt etwa 1:7. Für mich war diese analoge Einstellerei zunächst gewöhnungsbedürftig. Anders als bei digitalen Netzteilen sieht man nicht sofort, was man einstellt. Da die anliegende Spannung und der fließende Strom hier digital gemessen werden und die typischen A/D-Wandler-Chips etwa drei bis vier Messungen pro Sekunde hinbekommen, ergibt sich zwischen Potibewegung und der Anzeige der korrekten Werte immer eine kleine Verzögerung. Nicht schlimm, aber eben anders als bei digitalen Labornetzteilen. Dafür wird man aber entschädigt, wie sie in **Bild 3** sehen können: Die Anzeigen des Labornetzteils sind nahezu identisch mit denen meines Multimeters. Bei der Spannung ergibt sich eine Diskrepanz von 10 mV = 0,07 % und beim Strom von 0,3 mA = 0,08 % (bei 3 A ist die Abweichung ähnlich). Das ist insofern erstaunlich, als ich mein Multimeter vor der Messung mit einer 0,1 % genauen Referenzspannungsquelle kalibriert habe. Das PT 6080 A ist also viel genauer als die Herstellerangabe von 0,5 %  $\pm 5$  Digits und man kann sich auf seine Anzeigen verlassen. Das hätte ich so nicht erwartet!

**Bild 4** zeigt die maximale Spannung im CV-Modus (a) und den maximalen Strom im CC-Modus (b). Das PT 6080 A liefert also ein kleines bisschen mehr als die angegebenen 15 V und 3 A.

### Störpegel

Der nächste wichtige Aspekt ist die „Saubерkeit“ der Ausgangsspannung unter Last. Ein analoges Labornetzteil kleiner Leistung legt man sich ja nicht in erster Linie zu, um damit digitale



Bild 3. Vergleich von Multimeteranzeige und der Anzeige des Labornetzteils (links Spannung, rechts Strom).

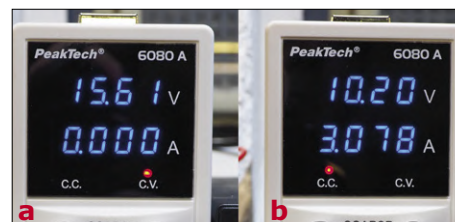


Bild 4. Maximale Spannung (a, CV) und maximaler Strom (b, CC).

Elektronik zu versorgen. Man will damit analoge Schaltungen wie Verstärker oder HF-Schaltungen mit möglichst geringen Störpegeln aus der Versorgung betreiben, damit Messungen ihrer Parameter aussagekräftig sind. Außer bei sehr teurem Labor-Equipment sind die Ausgangsspannungen bei digitalen Labornetzteilen immer mit Störsignalen der Schaltfrequenzen plus Oberwellen (mehr oder minder) verseucht. Linear geregelte, analoge Netzteile sollten da besser abschneiden. Doch ist das auch beim PT 6080 A der Fall?

Der Hersteller gibt an, dass die Restwelligkeit der Spannung  $0,5 \text{ mV}_{\text{eff}}$  beträgt. **Bild 5** zeigt vier Screenshots der oszilloskopierten Ausgangsspannung bei verschiedenen Bedingungen. Die Empfindlichkeit ( $2 \text{ mV/Teilstrich}$ ) und Zeitbasis ( $10 \text{ ms/Teilstrich}$ ) ist überall gleich. Rauschen und Brummen im CV-Betrieb liegen sowohl im Leerlauf (a) als auch unter Last (b) durchaus im angegebenen Bereich.

Nicht ganz so gut verhält sich die Ausgangsspannung, wenn sich das Netzteil im CC-Modus befindet. Hier liegt die Welligkeit unabhängig von der Last bei etwa  $2 \text{ mV}_{\text{eff}}$ . Beim gut sichtbaren niederfrequenten Anteil mit  $128 \text{ Hz}$  (c) bzw.  $171 \text{ Hz}$  (d) handelt es sich übrigens nicht um Netzbrummen, da sich ja sonst eine Frequenz von  $50$  oder  $100 \text{ Hz}$  zeigen müsste. Dieser Störpegel ist aber immer noch gut, und da man analoge Schaltungen ja eher im CV-Modus versorgt, ist der leicht erhöhte Störpegel im CC-Modus vernachlässigbar.

### Innenleben

Technisches Interesse zeigt sich ja von Kindesbeinen an darin, dass man wissen will, wie es innendrin aussieht. Also habe ich die sechs Schrauben des Blechdeckels entfernt und hineingeschaut (**Bild 6**).

Prominent zeigt sich der dicke Ringkerntrafo. Er hat eine umschaltbare Primärspannung für  $230\text{-}$  und  $115\text{-V}$ -Netze. Eine  $30\text{-V}$ -Wicklung mit Mittelanzapfung versorgt die vordere Steuer-/Display-Platine (c). Auf der Rückseite befindet sich die Platine mit Brückengleichrichter aus vier  $5\text{-A}$ -Dioden und dem mit  $4.700 \mu\text{F} / 50 \text{ V}$  gut dimensionierten Siebelko sowie der auf die Rückseite geschraubte Leistungstransistor. Das schwarze Teil rechts von den Dioden ist ein Relais, das man deutlich klicken hört, wenn man den Bereich um  $7,5 \text{ V}$  mit dem Poti überstreicht. Unter dieser Schwelle legt das Relais nämlich nur eine Sekundärwicklung von  $12,5 \text{ V}$  an den Gleichrichter. Darüber legt das Relais eine Wechselspannung von  $21,5 \text{ V}$  an. Diese Maßnahme reduziert die Verlustleistung, falls man niedrige Spannungen einstellt und viel Strom zieht. Ziemlich clever.

### Fazit

Man bekommt relativ viel Netzteil guter Qualität für die knapp  $60 \text{ €}$ , für die das PT 6080 A im Elektor-Store angeboten wird. Besonders überzeugend ist die hohe Genauigkeit der Anzeigen, die ich so nicht erwartet hätte. Zwar liegt ein Zettel mit Kalibrierungsangaben bei, aber eigenen Messungen vertraue ich mehr. Die Elektronik ist gut dimensioniert und Netzbrum-



Bild 6. Innenleben des PT 6080 A mit Netztrafo, Leistungsplatine und Steuerteil mit SMDs.

men zeigt sich nirgendwo. Die Spannungen werden sauber ausgeregelt und brechen auch unter Last nur extrem gering ein. Die Störpegel im CV-Modus sind gut und das Netzteil wird auch bei hoher Dauerlast dank des Tricks mit der Umschaltung per Relais nicht besonders heiß, was für analoge Netzteile nicht selbstverständlich ist. Kurzschlussfest ist es sowieso.

Alles in Allem verdient dieses kleine Labornetzteil das Prädikat „empfehlenswert“. ◀

200009-01

**IM ELEKTOR-STORE**

→ Netzgerät PeakTech 6080 A DC

[www.elektor.de/19111](http://www.elektor.de/19111)

### Weblinks

- [1] Elektor-Präzisions-Netzgerät (1980): [www.elektormagazine.de/index.php/magazine/elektor-198009/46150](http://www.elektormagazine.de/index.php/magazine/elektor-198009/46150)
- [2] PT 6080 A, Daten: [www.peaktech.de/produktdetails/kategorie/acdc-stabilisierte-labornetzgeraete/produkt/peaktech-6080-a.html](http://www.peaktech.de/produktdetails/kategorie/acdc-stabilisierte-labornetzgeraete/produkt/peaktech-6080-a.html)





# Willkommen in Ihrem E-SHOP

ELEKTOR EMPFIEHLT



## JOY-iT Joy-View 13 Touchscreen

Es ist nicht so einfach, einen kleinen, aber qualitativ hochwertigen Monitor zu finden, der auch noch portabel ist. Das Modell Joy-View 13 erfüllt diese Kriterien. Dank seiner HDMI- und USB-C-Ports kann er an fast alle modernen Geräte angeschlossen werden.

Der portable Touchscreen Joy-View 13 ist ein sauber verarbeiteter Bildschirm mit sehr guter Darstellung, eingebauter kapazitiver Touch-Fähigkeit und flexiblen

Anschlüssen sowie Stromversorgungsoptionen. Dadurch kann er für viele Anwendungen und Computer-Kombinationen verwendet

werden. Wenn Sie nach einem kompakten, stabilen und portablen Bildschirm suchen, ist er eine ausgezeichnete Wahl. Und wenn er ihnen zu klein sein sollte: Es gibt mit dem Joy-View 15 auch eine 15,6"-Version dieses Monitors.



**Harry Baggen**  
(Elektor-Autor)

[www.elektor.de/19162](http://www.elektor.de/19162) | [www.elektor.de/19163](http://www.elektor.de/19163)

## Elektor-Bestseller

1. Elektor-Archiv 1970-2019 (USB-Stick)

[www.elektor.de/19197](http://www.elektor.de/19197)



2. 45 Elektronik-Projekte für den Raspberry Pi

[www.elektor.de/19249](http://www.elektor.de/19249)

3. Raspberry Pi 4 B (2 GB)

[www.elektor.de/18965](http://www.elektor.de/18965)

4. Programming with Node-RED

[www.elektor.de/19224](http://www.elektor.de/19224)

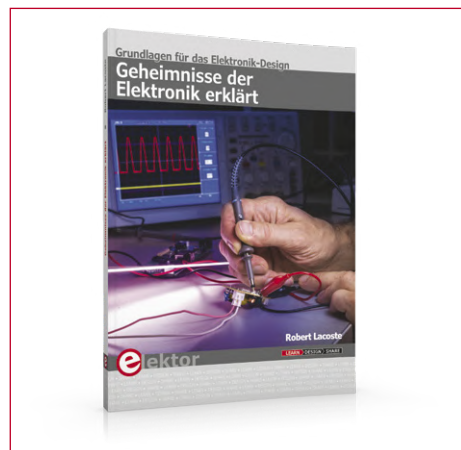
5. Mikrocontroller-Basics mit PIC

[www.elektor.de/18945](http://www.elektor.de/18945)

6. Raspberry Pi 4 B (4 GB)

[www.elektor.de/18964](http://www.elektor.de/18964)

## 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)

## JOY-iT JT-RD6006 Labornetzteil-Bundle



Dieses Bundle (exklusiv bei Elektor erhältlich) enthält das neue programmierbare Labornetzteil JT-RD6006, ein passendes Industriennetzteil (60 V, 6 A), ein großes Aluminiumgehäuse und ein plug-in WiFi-Modul, mit dem das JT-RD6006 über eine drahtlose Verbindung von anderen (mobilen) Geräten gesteuert werden kann. Das Bundle wird in einem praktischen Hartschalenkoffer geliefert.



Mitgliederpreis: 175,46 €

[www.elektor.de/19211](http://www.elektor.de/19211)

## IoT Home Hacks with ESP8266



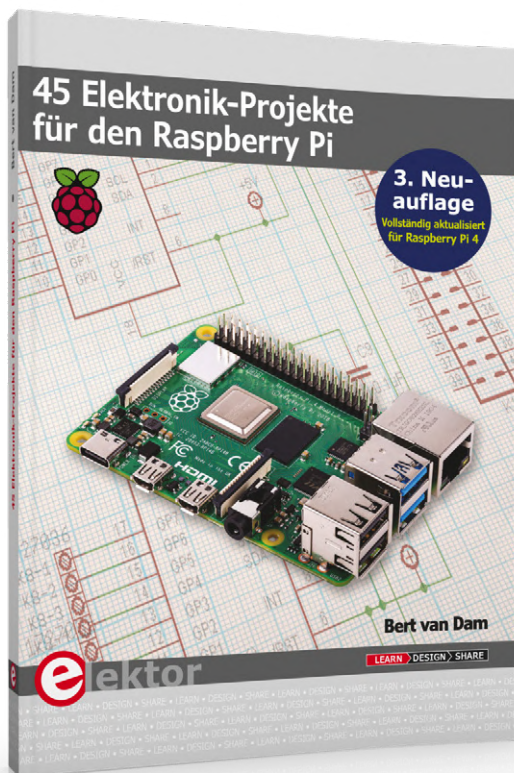
Es gibt viele sogenannte „Arduino-kompatible“ Plattformen auf dem Markt. Das ESP8266 – in Form des Wemos D1 Mini Pro – ist eines, das wirklich herausragt. Dieser Mikrocontroller verfügt über einen WiFi-Internetzugang und die Möglichkeit eines Flash-Dateisystems mit bis zu 16 MB externem Flash-Speicher. 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)





## 45 Elektronik-Projekte für den Raspberry Pi

Die komplett überarbeitete 3. Neuauflage des Bestsellers 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: 34,80 €

[www.elektor.de/19249](http://www.elektor.de/19249)

### Robotik und Künstliche Intelligenz



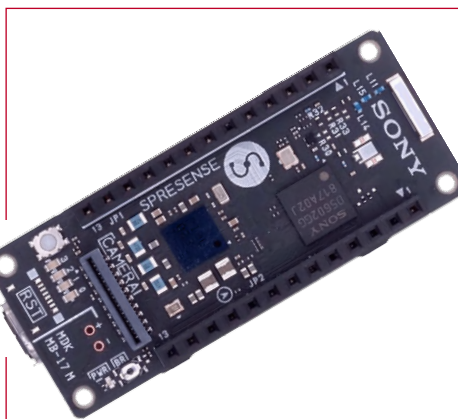
Dieses Buch ist eine Einführung in das hochaktuelle Gebiet der Robotik. Dabei stehen praktische Anwendungsbeispiele im Vordergrund. Neben den technischen und mechanischen Grundlagen werden die elektronischen Komponenten und Module erläutert. Eine zentrale Rolle spielt dabei der Mikrocontroller.



Mitgliederpreis: 32,80 €

[www.elektor.de/19120](http://www.elektor.de/19120)

### Sony Spresense Main Board



Spresense ist ein kompaktes Entwicklungsboard, das auf dem energieeffizienten Multicore-Mikrocontroller CXD5602 von Sony basiert. Es ermöglicht Entwicklern die Erstellung von Anwendungen in sehr kurzer Zeit und wird von der Arduino IDE sowie dem fortschrittlicheren NuttX-basierten SDK unterstützt. Die Hauptplatine verwendet einen von Sony entwickelten Prozessor für IoT- und Sensoranwendungen. Die Hauptplatine kann allein oder mit der Erweiterungsplatine betrieben werden.



Mitgliederpreis: 71,96 €

[www.elektor.de/19233](http://www.elektor.de/19233)

### Tactigon ONE – Gesten- und Bewegungscontroller mit KI



Tactigon ONE ist das ultimative Wearable-Board, programmierbar mit Arduino IDE. Es verfügt über GPIOs, UART, ein integriertes Batterieladesystem, einen schnellen Mikrocontroller und Bluetooth LE onboard! Messen Sie Linear- und Winkelbewegungen mit einem 3-Achsen-Gyroskop und einem 3-Achsen-Beschleunigungsmesser! Die Highspeed-Mikrocontroller-Einheit und das Arduino SDK ermöglichen die Ausführung von komplexen Sketches.



Mitgliederpreis: 71,96 €

[www.elektor.de/19218](http://www.elektor.de/19218)



**ELEKTOR ETHICS**

# Eine große Herausforderung

## Sichere Produkte im IoT-Zeitalter

Von **Tessel Renzenbrink** (Niederlande)

Sicherheitsgurte, Airbags, Knautschzonen, Geschwindigkeits- und Alkohollimits im Straßenverkehr. Jahrzehntelange technologische Innovation und Regulierung haben Autos immer sicherer gemacht. Dasselbe gilt für andere Konsumgüter. Wir kaufen sie ohne zu zögern im Laden und vertrauen darauf, dass sie sicher sind. Aber mit dem Aufkommen des Internets der Dinge wird die Büchse der Pandora geöffnet. Jetzt, wo immer mehr Produkte mit Elektronik, Software und Netzwerkverbindungen ausgestattet sind, entstehen auch neue Gefahren und Risiken. Es gibt zahlreiche Beispiele: von vernetztem Spielzeug, mit dem Kinder ausspioniert werden können, bis hin zu Autos, deren Steuerung aus der Ferne übernommen werden kann.

Professor Ross Anderson und seine Kollegen haben im Auftrag der Europäischen Kommission untersucht, welche Maßnahmen erforderlich sind, um die traditionelle Produktsicherheit mit der Ausbreitung des IoT zu vereinen. Die EU arbeitet an einem neuen Gesetz, um die Produktsicherheit auch in Zukunft zu gewährleisten. Sie baten Anderson, eine Übersicht der dafür erforderlichen Maßnahmen zu erarbeiten. Anderson, Professor für Security Engineering, tat dies zusammen mit seinen Kollegen Eirann Leverett, Risikoforscher, und Richard Clayton, Sicherheitsforscher, die alle mit der Universität Cambridge verbunden sind. Der Bericht mit den Ergebnissen wurde 2017 veröffentlicht [1].

### Trilemma

Wir haben noch einen langen Weg vor uns, schlussfolgern Anderson und seine Kollegen. Der Kern des Problems besteht darin, dass sich die Art und Weise, wie die Sicherheit von herkömmlichen Produkten erreicht wird, grundlegend von derjenigen für digitale Technologien unterscheidet. Produkte werden, bevor sie auf den Markt gebracht werden, getestet und geprüft und erhalten dann ein Sicherheitszertifikat. Wenn wesentliche Änderungen am Produkt vorgenommen werden, muss es neu zertifiziert werden. Die Sicherheit bei Software profitiert dagegen von Änderungen. Software wird kontinuierlich überwacht und regelmäßig mit Patches und Updates aktualisiert. Bei der Präsentation der Studie auf dem Chaos Communication Congress (CCC) im Dezember 2019 geht Anderson näher darauf ein [2]. Er nennt dies das Trilemma der IoT-Produkte. Wenn auf einer Zertifizierung vor der Markteinführung bestanden wird, kann die Software nicht mehr aktualisiert werden und das Produkt ist unsicher. Wenn die Software aktualisiert wird, ist die Zertifizierung futsch. Wenn aber Zertifizierung und Updates kombiniert werden - und damit nach jedem Software-Update der Zertifizierungsprozess erneut durchlaufen werden muss - dann steigen die Kosten ins Unermessliche.

### Lebensdauer und Komplexität

Traditionelle Produkte und digitale Dienstleistungen haben jeweils ihre eigenen Besonderheiten, die nun aufeinandertreffen. Und wir müssen aufpassen, dass wir nicht das Schlimmste aus beiden Welten übernehmen, sagt Anderson. Vergleichen Sie zum Beispiel Telefone und Autos. Die Firmware und das Betriebssystem der Telefone werden im Durchschnitt etwa drei Jahre lang unterstützt. Nachdem der Support abgelaufen ist, wird das Telefon unsicher, weil die Schwachstellen in der Software nicht mehr gepatcht werden. Die Menschen sind gezwungen, Telefone zu ersetzen, obwohl die Hardware oft noch in Ordnung ist. Wir müssen verhindern, dass diese Methode in Zukunft auch in der Autoindustrie verwendet wird, sagt Anderson.

Aber der Wunsch nach Langlebigkeit bringt neue Herausforderungen mit sich. Angenommen, ein Hersteller wird verpflichtet, ein bestimmtes Automodell mindestens 20 Jahre lang zu unterstützen. Ein Softwareentwickler, der heute ein Programm für ein Auto schreibt, das 2022 auf den Markt kommen wird, muss dann berücksichtigen, dass der Code bis 2042 funktionieren muss.

Eine weitere Herausforderung ist die Komplexität. Der Gedanke beim IoT ist, dass Geräte miteinander kommunizieren und selbstständig handeln können, selbstfahrende Autos, die Informationen mit einer intelligenten Straßeninfrastruktur austauschen, und Kühlschränke, die selbst Lebensmittel bestellen. Aus sicherheitstechnischer Sicht bedeutet dies, dass es nicht mehr ausreicht, Geräte einzeln zu testen. Die Risiken, die sich aus der Interaktion mit anderen Geräten ergeben, müssen ebenfalls berücksichtigt werden.

### Empfehlungen

Die Gewährleistung von Produktsicherheit im Zeitalter des IoT ist eine komplexe Aufgabe, die sich nicht von heute auf morgen realisieren lässt. In ihrem Bericht an die Europäische Kommission geben die drei Forscher eine Reihe von Empfehlungen. Die wichtigste ist die Einrichtung eines neuen europäischen Insti-



nien ändern, um neuen Entwicklungen Rechnung zu tragen. Beispielsweise sollten die Richtlinien für Produkthaftung auf digitale Dienste ausgedehnt werden. Nach der aktuellen Bestimmung kann ein Hersteller von Navigationsgeräten für Fehler haftbar gemacht werden, die Hersteller von Google Maps jedoch nicht.

### Die EU als Bollwerk gegen ein Internet of Trash

Es habe sich als schwierig erwiesen, die Empfehlungen umzusetzen, sagte Anderson während der Präsentation beim CCC. Es sind viele verschiedene Akteure beteiligt, die alle unterschiedliche Interessen haben. Beispielsweise haben Google und Facebook gegen den Vorschlag, digitale Dienste den bestehenden Haftungsregeln zu unterwerfen, Lobbyarbeit betrieben. Aber seit der Veröffentlichung des Berichts wurde auch einiges erreicht. So wurde beispielsweise die Richtlinie 2019/771 verabschiedet, die vorsieht, dass Verbraucher von Produkten mit digitalen Elementen Recht auf mindestens zwei Jahre Aktualisierungen haben, oder länger, wenn dies vernünftigerweise erwartet werden kann. Letzteres bedeutet, dass für Waren, die länger halten sollten (wie zum Beispiel Waschmaschinen),

auch länger Aktualisierungen zu erhalten sein müssen. Die immense Aufgabe, die Anderson, Leverett und Clayton beschrieben haben, um die Sicherheit von IoT-Produkten in Zukunft zu gewährleisten, steht in scharfem Kontrast zu den langsamen Entscheidungsprozessen der Europäischen Union. Dennoch setzen sie ihre Hoffnungen in die EU, denn, wie sie in einem entsprechenden wissenschaftlichen Artikel schreiben: „Die EU ist bereits die weltweit führende Regulierungsbehörde für den Datenschutz, da sich Washington nicht dafür interessiert und sonst niemand groß genug ist, um eine Rolle zu spielen. Sie sollte danach streben, auch der Hauptregulator für Sicherheit zu werden – oder sie riskiert, die aktuellen Sicherheitsziele zu gefährden“ [3].

200066-04

Foto: „Safety“ von Gabor Kiss, CC BY 2.0 <https://www.flickr.com/photos/-nevi-/5404057758/>.

tuts, der *European Safety and Security Engineering Agency* (ESSEA). Die ESSEA sollte über das technische Wissen verfügen, um unter anderem Standards zu entwickeln und die politischen Entscheidungsträger zu informieren. Darüber hinaus muss sie eine zentrale Datenbank beherbergen, in der Informationen über Software-Schwachstellen, Komponentenausfälle und Systemintegrationsfehler gesammelt werden. Derzeit werden solche Daten bei verschiedenen Instituten und Unternehmen gespeichert. Wenn die Daten zusammengeführt werden, können alle voneinander lernen, und es werden nicht immer die gleichen Fehler gemacht.

Eine weitere Empfehlung ist, dass Hersteller, die ihre Produkte um Netzwerkfähigkeiten erweitern, die Möglichkeit zur Durchführung von Software-Updates garantieren sollten.

Darüber hinaus sollte die EU eine Reihe bestehender Richtli-

### Links und Literatur

- [1] Eireann Leverett, Richard Clayton, Ross Anderson, Editor G.Baldini, *Standardisation and Certification of Safety, Security and Privacy in the 'Internet of Things'*, European Commission, Brussels, Belgium, 2017.
- [2] The sustainability of safety, security and privacy [Video]. Präsentation von Ros Anderson auf dem Chaos Communication Congress in Leipzig, 28. Dezember 2019  
[https://media.ccc.de/v/36c3-10924-the\\_sustainability\\_of\\_safety\\_security\\_and\\_privacy#t=1397](https://media.ccc.de/v/36c3-10924-the_sustainability_of_safety_security_and_privacy#t=1397)
- [3] Eireann Leverett, Richard Clayton, Ross Anderson, *Standardisation and Certification of the 'Internet of Things'*, 2017.



# 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 2. Juni 2020.**

### Die Gewinner des Hexadokus aus der Ausgabe März/April 2020 stehen fest!

Die richtige Lösung ist: **749DF**.

Einen Elektor-Wertgutschein über je 50 € haben gewonnen: Étienne Guyot, Ria Van Broeck,

Peter Tabatt, Tomas Bakke und Jose A. Gordo.

Herzlichen Glückwunsch!

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|   |   | C |   | 6 |   | 1 | 4 | 0 | 7 |   | 2 |   | A |   |   |
| 4 |   |   |   | A |   |   | 0 | 9 |   |   | 3 |   |   |   | 2 |
|   | A |   | 5 | F |   |   | D | 8 |   |   | 6 | 1 |   | E |   |
|   | 7 | 2 |   |   |   |   |   |   |   |   |   |   |   | F | 4 |
| 2 | 3 |   | C |   |   |   | F | 7 |   |   |   | 4 |   | B | E |
|   |   | 5 |   |   |   | 2 | E | 1 | 3 |   |   |   | 0 |   |   |
|   | 0 | E | 7 | B |   |   |   |   |   |   | F | 8 | 1 | C |   |
|   | 1 | 6 |   | 0 | C | 5 |   |   | D | B | E |   | 9 | 2 |   |
|   | 5 | 9 |   | 4 | 6 | B |   |   | E | 7 | D |   | 3 | F |   |
|   | 2 | D | 3 | 7 |   |   |   |   |   |   | 8 | B | E | 5 |   |
|   |   | 1 |   |   |   | A | 3 | B | F |   |   |   | 8 |   |   |
| 7 | 8 |   | B |   |   | 2 | C |   |   |   | D |   | A | 6 |   |
|   | 9 | 0 |   |   |   |   |   |   |   |   |   |   | 6 | 1 |   |
|   | C |   | A | E |   | 9 | F |   |   | 4 | 0 |   | 8 |   |   |
| 6 |   |   |   | 8 |   | 5 | E |   |   | 9 |   |   |   |   | 7 |
|   |   | B |   | 2 |   | 6 | 1 | 3 | 0 |   | 7 | 5 |   |   |   |

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 2 | C | D | 5 | 8 | E | 9 | 3 | 6 | 0 | 7 | A | B | F |
| 8 | 5 | 3 | 7 | 0 | F | B | C | E | A | D | 2 | 6 | 9 | 1 | 4 |
| 0 | D | 6 | E | 9 | 4 | 7 | A | B | 1 | C | F | 3 | 2 | 8 | 5 |
| A | 9 | F | B | 1 | 2 | 3 | 6 | 4 | 5 | 7 | 8 | C | D | E | 0 |
| 5 | 7 | 8 | 2 | A | B | F | D | 6 | 4 | 0 | C | E | 3 | 9 | 1 |
| 9 | 6 | 4 | F | C | E | 0 | 8 | 3 | 7 | 5 | 1 | 2 | B | D | A |
| B | 3 | D | 0 | 2 | 6 | 1 | 5 | 8 | 9 | E | A | F | C | 4 | 7 |
| C | A | E | 1 | 3 | 7 | 4 | 9 | D | F | 2 | B | 0 | 5 | 6 | 8 |
| D | 2 | 9 | 3 | 6 | 0 | A | F | 5 | 8 | B | E | 1 | 4 | 7 | C |
| E | 1 | 5 | A | B | 9 | 2 | 7 | C | D | 4 | 6 | 8 | F | 0 | 3 |
| 6 | B | 0 | 4 | 8 | C | D | 1 | 7 | 2 | F | 3 | 5 | E | A | 9 |
| 7 | F | C | 8 | E | 3 | 5 | 4 | A | 0 | 1 | 9 | B | 6 | 2 | D |
| 3 | 0 | A | D | 4 | 1 | E | 2 | F | 6 | 8 | 5 | 9 | 7 | C | B |
| F | E | 7 | 9 | 5 | 8 | 6 | B | 0 | C | A | D | 4 | 1 | 3 | 2 |
| 2 | 8 | B | 5 | 7 | D | C | 3 | 1 | E | 9 | 4 | A | 0 | F | 6 |
| 4 | C | 1 | 6 | F | A | 9 | 0 | 2 | B | 3 | 7 | D | 8 | 5 | E |

Der Rechtsweg ist ausgeschlossen. Mitarbeiter der in der Unternehmensgruppe Elektor International Media B.V. zusammengeschlossenen Verlage und deren Angehörige sind von der Teilnahme ausgeschlossen.

**188** Seiten mit den besten Artikeln über Raspberry Pi im Industrie-Einsatz

Der Raspberry Pi wurde ursprünglich für die Forschung und Lehre entwickelt. Der kleine Bastler-PC hat jedoch bewiesen, dass er auch in der Industrie vielseitig einsetzbar ist. In enger Zusammenarbeit mit ELEKTRONIKPRAXIS haben wir dieses einzigartige Buch über aktuelle Lösungsansätze und Anwendungsbeispiele vom RPi in der Industrie veröffentlicht.

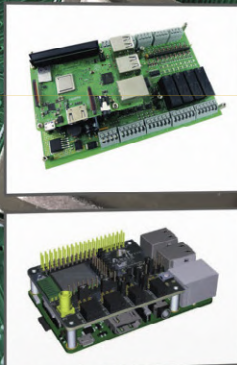
# Raspberry Pi goes Industry

**e**lektor

## Raspberry Pi goes Industry

Die besten Artikel über Raspberry Pi im Industrie-Einsatz

- Übersicht der Raspberry-Pi-Boards
- 40 Betriebssysteme für Raspberry Pi
- Raspberry Pi für die industrielle Steuerung
- Die besten Anwendungsbeispiele
- Und vieles mehr!



**JETZT**  
**24,90 €**  
im Elektor Shop

**e**lektor

ELEKTRONIK  
PRAXIS

Dieses Buch bietet eine anschauliche Übersicht über die Möglichkeiten der industriellen Anwendung des Raspberry Pi. Von Sensoren, Blockchain-Technologie, Industriecontroller über Automatisierung und KI, werden zahlreiche Anwendungsmöglichkeiten für den Raspberry Pi anschaulich erklärt. Außerdem werden unterschiedlichste Tools zur erweiterten Nutzung des Raspberry Pi sowie die verschiedenen Betriebssysteme vorgestellt. Die Möglichkeiten und Grenzen für den Raspberry Pi in der Industrie und konkrete Anwendungsbeispiele liefern ein realistisches Gesamtbild.

Folgende Themen werden beschrieben:

- Raspberry Pi 3 A+
- Raspberry Pi 4 B
- RPi CM3+
- 45 Betriebssysteme für RPi
- RPi und Blockchain Ethereum
- RPi im Industrieinsatz
- und viele weitere Themen

Der Raspberry Pi wurde ursprünglich für die Forschung und Lehre entwickelt. Der kleine Bastler-PC hat jedoch bewiesen, dass er auch in der Industrie vielseitig einsetzbar ist. Aufgrund der zahlreichen Anwendungen des Raspberry Pi in privaten Hobbykellern und Hochschullabors ist er etlichen angehenden Ingenieuren bereits bekannt, daher ist es nicht verwunderlich, dass er schon in unterschiedlichsten industriellen Betrieben zum Einsatz kommt.

ELEKTRONIK  
**PRAXIS**

Wissen.  
Impulse.  
Kontakte.

**Bestellen Sie jetzt:**

[www.elektor.de/raspberry-pi-goes-industry](http://www.elektor.de/raspberry-pi-goes-industry)





**electronica**  
**fast forward**  
the startup platform

powered by elektor

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 m²)** auf der electronica 2020
- Erhalten Sie **große Aufmerksamkeit** vor, während und nach  
der electronica 2020. (10.-13.11.2020)



Sign up  
here

[elektormagazine.de/fastforward](http://elektormagazine.de/fastforward)

electronica Fast Forward is brought to you by



**electronica**

