



*magazin für  
computer  
technik*

8.11.2024

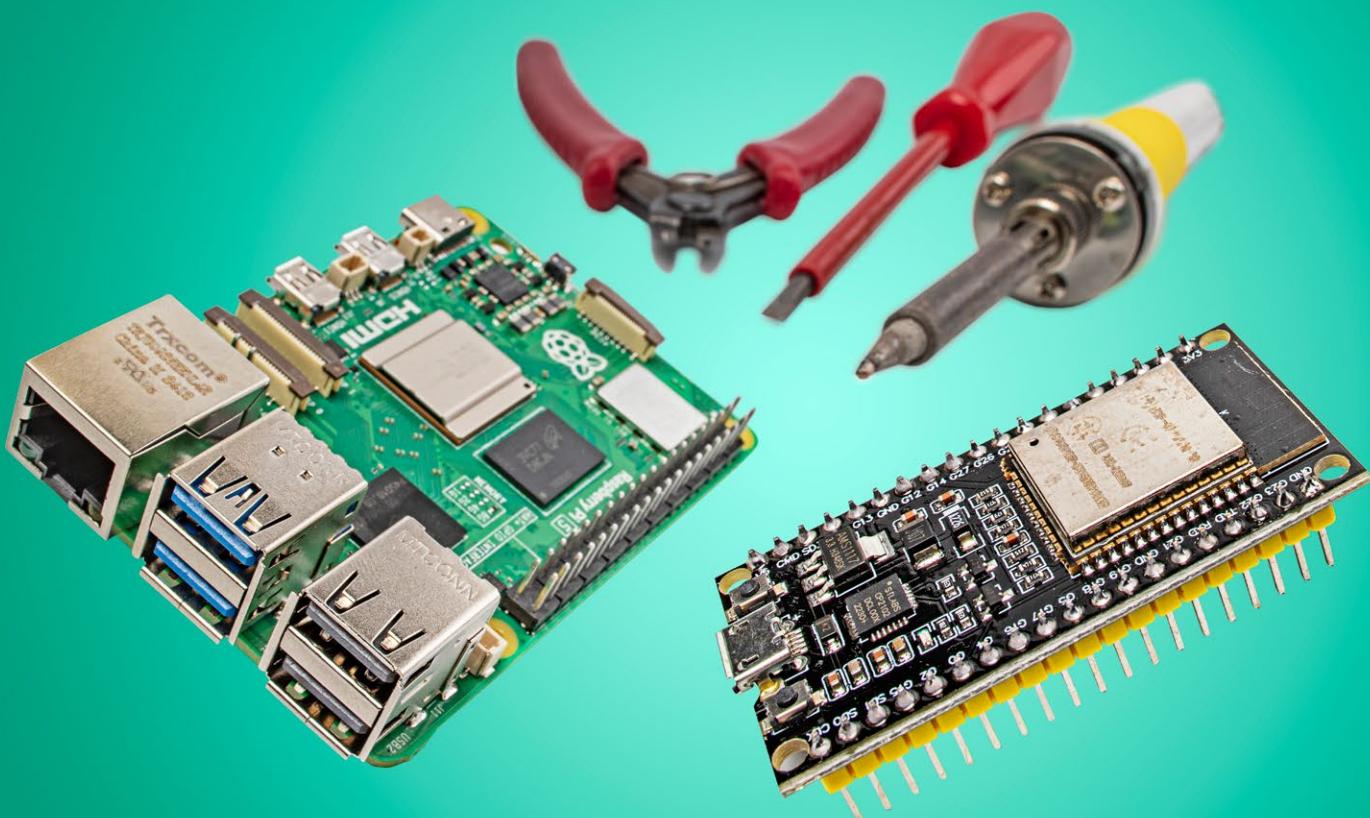
Europas größtes IT- und Tech-Magazin

Sonderausgabe

# Das c't-Bastel-Kompendium

Was Anfänger und Profis brauchen

Kaufberatung • Elektronikgrundlagen • Programmierpraxis



## Spaß mit Raspi, ESP & Co.

Projekte von LED-Armband bis Kubernetes-Cluster

Zubehör: Gehäuse, Sensoren, Kameras ...

© Copyright by Heise Medien.

E-BOOK  
ZUM  
HEFT



# Mission Serverstart: Ihr Einstieg in die IT-Welt!

Leistungsstarke & kostengünstige Server  
mit geringem Platzbedarf

Die Einstiegs-Server der Thomas-Krenn.AG bieten die **perfekte Kombination** aus Preis und Leistung für Unternehmen jeder Größe. Dank flexibler Größenoptionen passen sie sich optimal an individuelle Anforderungen an – egal ob kleines Büro oder wachsendes Unternehmen. Ideal um zuverlässig und kosteneffizient in die Serverwelt einzusteigen. Vertrauen Sie auf bewährte Qualität und Flexibilität – für eine zukunftssichere IT-Infrastruktur!



Jetzt sichern:  
Einstieger-Systeme  
[thomas-krenn.com/einstieg24](http://thomas-krenn.com/einstieg24)

**THOMAS**  
**KRENN**<sup>®</sup>  
IT's people business



## Basteln: Spaß mit Raspi, ESP & Co.

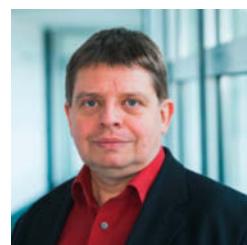
Winterzeit ist Bastelzeit: Diese c't-Sonderausgabe eröffnet offiziell die Programmier- und Lötaison. Wer schon immer mal ausprobieren wollte, wie man einen Arduino, Raspberry Pi oder ESP-Mikrocontroller zum Laufen kriegt, wie man dann mit Sensoren und anderen Erweiterungen ein nützliches Projekt aufsetzt und dazu die passende Software schreibt, der ist hier genau richtig.

Wir erleichtern Ihnen den Einstieg und ebnen den Weg zum Erfolg mit Ihrem ersten kleinen Projekt. Dazu gehört es, die geeignete Hardware zu finden und das passende Betriebssystem samt Entwicklungsumgebung dazu. Dann geht es los und bald lässt man die ersten LEDs blinken. Die Freude ist groß, doch schnell erkennt man: Da geht noch mehr. Kein Problem, denn dieses Heft spannt einen weiten Bogen.

Erfahrene Bastler finden neue Anregungen, eine Übersicht über die aktuellen Boards und Erweiterungen sowie über Entwicklungswerkzeuge. Und selbst Profi-Entwickler lernen garantiert noch dazu. Die Bandbreite beginnt bei Projektideen für Schüler und endet bei anspruchsvollen Lösungen, wie etwa der Steuerung von Nachspeicheröfen und dem Raspi als Überwachungskamera oder als Navigationshilfe auf dem Wasser.

Das Schönste daran: Die finanzielle Hürde ist zunächst überschaubar, der Spaßfaktor jedoch um Vieles höher. Wenn daraus allerdings ein ausgewachsenes Hobby wird, kann es etwas teurer werden. Etwa, weil passende Gehäuse für die selbst gebauten Smart-Home-Geräte hermüssen und die Anschaffung eines 3D-Druckers unumgänglich wird. Na ja, oder man beauftragt einen 3D-Druckservice, was dann wiederum erheblich günstiger ist.

Etliche c't-Redakteure haben das Basteln mit Mikrocontrollern und Kleincomputern zum Hobby erkoren. Beim Lesen dieses Heftes können Sie von all den gesammelten Erfahrungen profitieren und sich von der Begeisterung anstecken lassen. Viel Spaß dabei!

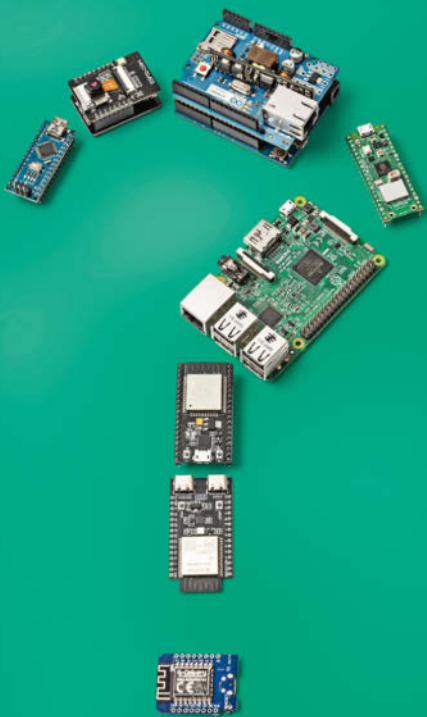


*Axel Kossel*

Axel Kossel

# Inhalt c't 25/2024

## 8 Einstieg ins Basteln



Ein ganzes Heft voll Bastelkram – sind die bei c't jetzt dem Makramee verfallen? Natürlich nicht! Stattdessen geht es darum, was man mit Kleinstcomputern und Mikrocontrollern so alles anfangen kann. Einsteiger richten mit unseren Tipps ihre Werkstatt ein und erleben erste Erfolge mit ESP, Raspi & Co., erfahrene Bastler finden neue Ideen für teils recht anspruchsvolle Projekte.

## 78 Die ESP-Familie

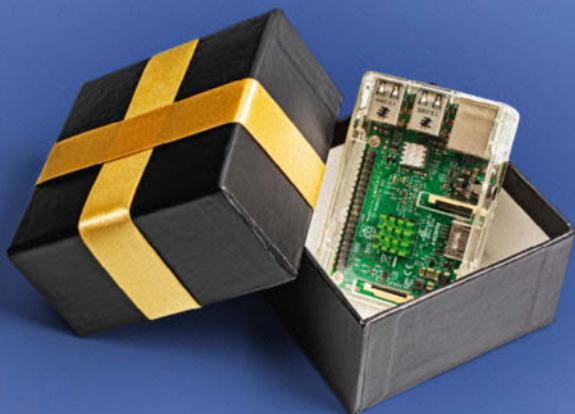


Hardware zum Taschengeldpreis und eine riesige Community machen die ESP-Chips der Firma Espressif zur idealen Spielwiese für alle, die sich an das Thema Mikrocontroller herantasten wollen. Wir erläutern, welches Platinchen Sie für welche Aufgabe brauchen, und stellen die wichtigsten Entwicklungsumgebungen vor. Drei fertige Projekte regen zum Nachmachen an.

- 8 Loslegen** mit Raspberry Pi, ESP32 & Co.
- 12 Wegbereiter** Wo Arduino und Raspberry Pi glänzen
- 18 Interview** Experimentelle Synthesizer von Soma
- 24 Glossar** Komponenten, Schnittstellen, Protokolle
- 32 Bastelwerkstatt** perfekt einrichten
- 36 Recht** Haftung und Gewährleistung beim Basteln
- 40 Gehäuse** Schicker Schutz aus dem 3D-Drucker
- 46 Fractal North Pi** Edles Raspberry-Gehäuse
- 50 Sensoren** für Ihre Bastelpunkte
- 58 Messdaten** sammeln und auswerten
- 62 Programmieren in C** Die Überall-Sprache
- 70 Versionsverwaltung** mit Git, GitHub und GitLab
- 74 Bücher** Raspberry-Pi-Anleitung, 255 Raspi-Tipps
- 75 Bücher** Python auf dem Pi, Raspberry-Handbuch
- 76 Web-Tipps** Entwerfen, bauen, reparieren

- 78 Mikrocontroller** Chips und Boards der ESP-Familie
- 84 Programmierwerkzeuge** für ESP & Co.
- 90 „On Air“-Lampe** mit ESP8266-Fernsteuerung
- 96 Konzertarmband** mit Arduino und Infrarot steuern
- 102 Speicherheizung** mit ESP und Sensoren optimieren

## 108 Loslegen mit dem Raspberry Pi



Hardware, die ein ausgewachsenes Linux mit grafischer Oberfläche wuptt, dazu 40 An-schlusspins für Elektronikbasteleien und das Ganze für nicht mal 'nen Fuffi: Das ist das Erfolgsrezept des Raspberry Pi. Wir erklären, welches Modell Sie wofür kaufen sollten und wie Sie es in Betrieb nehmen. Unsere Projekt-vorschläge reichen von der blinkenden LED bis zum Raspi-Cluster mit Cloudtechnik.

Frisch aus  
**c't Nerdistan**

- 46 Edles Raspi-Gehäuse** Fractal North PI nachgebaut
- 102 Heizungssteuerung** mit ESP & Co.
- 160 Bootsnavigation** Raspi warnt vor dicken Pötten

**c't Hardcore**

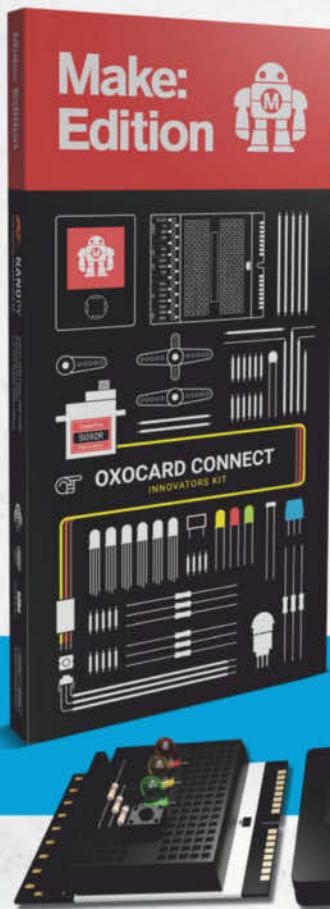
- 62 C-Tutorial** Die Überall-Progammiersprache
- 132 Raspberry Pi RP2350** Der Mikrocontroller des Pico
- 150 Raspi-Cluster** mit Kubernetes und Ansible

## Immer in c't

- 108 Raspberry Pi** Kaufberatung und erste Schritte
- 116 Raspi OS** Bootmedien erstellen und konfigurieren
- 118 GPIO-Anschlüsse** des Raspi programmieren
- 126 IoT-Fernwartung** Raspberry Pi per Cloud managen
- 130 MicroPython-Entwicklung** auf dem Raspi Pico
- 132 Raspi-Mikrocontroller** RP2350 mit ARM und RISC-V
- 138 Raspi im Unterricht** Projektideen für Schüler
- 144 Sofort loslegen** mit fertigen Spezial-Images
- 150 Kubernetes-Cluster** mit dem Raspberry Pi
- 154 Outdoor-IP-Kamera** im Eigenbau
- 160 Schiffs-Tracking** Raspi als Boots-Navigationshilfe

# Dein Einstieg in die Welt der Elektronik

## Einstecken, ausprobieren, experimentieren



Die Oxocard Connect ist ein kleiner Experimentiercomputer, in den man fertige oder selbst gelötete Platinen einstecken kann. Wie man es von Spielkonsolen kennt, können Sie sofort loslegen!

- Experimentierset mit 96 Elektronikbauteilen
- Einfach programmierbar mit Skriptsprache nanopy und browserbasierter Programmierplattform (IDE)
- Inkl. Spielen und Demoprogrammen
- Kompletter Elektronik-Kurs (online) mit 13 Beispielen

Zum GRATIS-Artikel  
der Make-Redaktion:



[make-magazin.de/  
oxocard](http://make-magazin.de/oxocard)

Oxocard Connect  
mit Breadboard-Cartridge

### Erweiterungen für deine Oxocard Connect



**Expansion Cartridge** für  
einfache elektronische  
Schaltungen



**Pixelmatrix Cartridge** mit 25 farbigen,  
einzeln adressierbaren LEDs

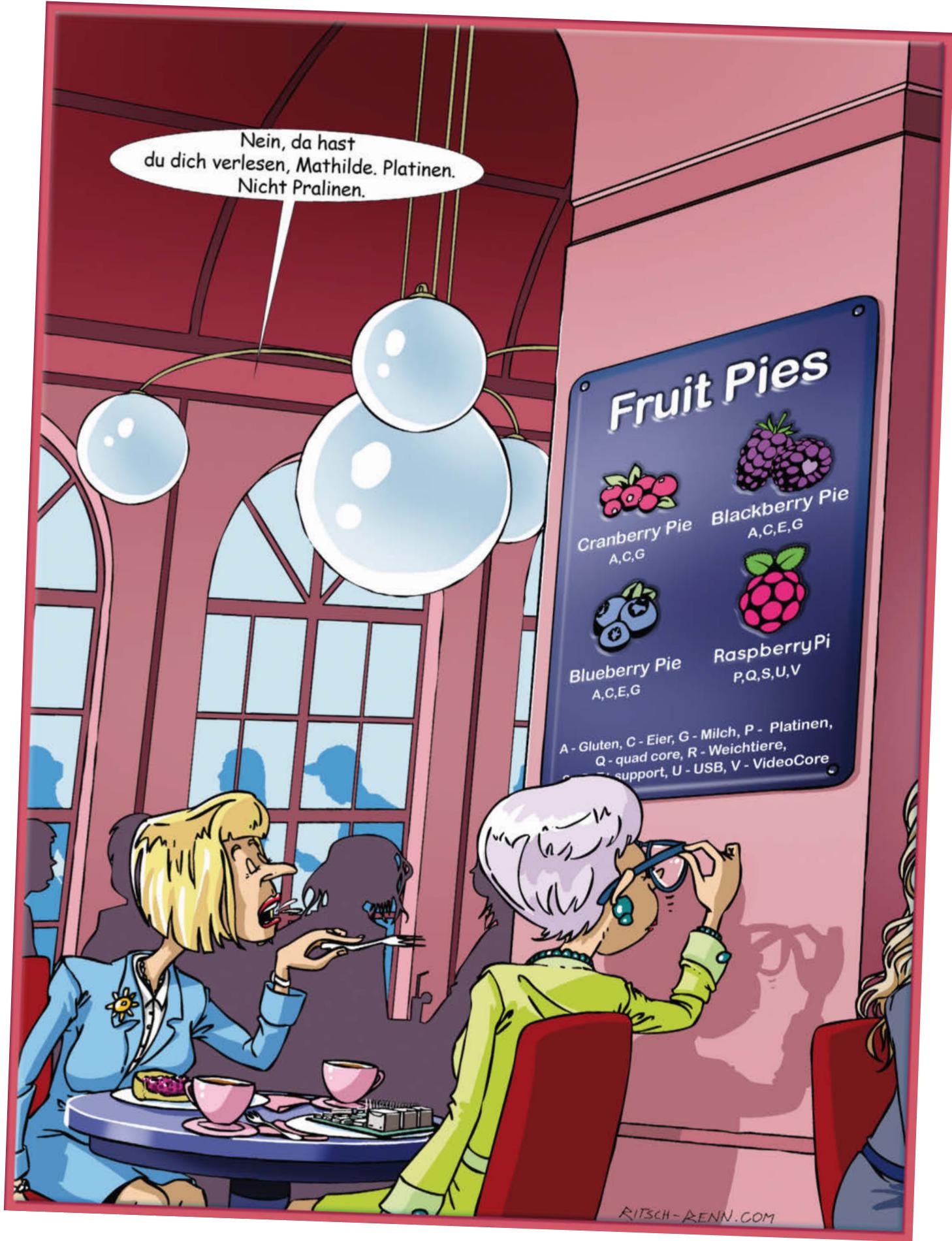


**Synthesizer Cartridge:**  
38-stimmig mit Drumkits,  
Reverb uvm.

Hier geht's zu unseren Oxocard Connect-Produkten:

[shop.heise.de/oxocard](http://shop.heise.de/oxocard)

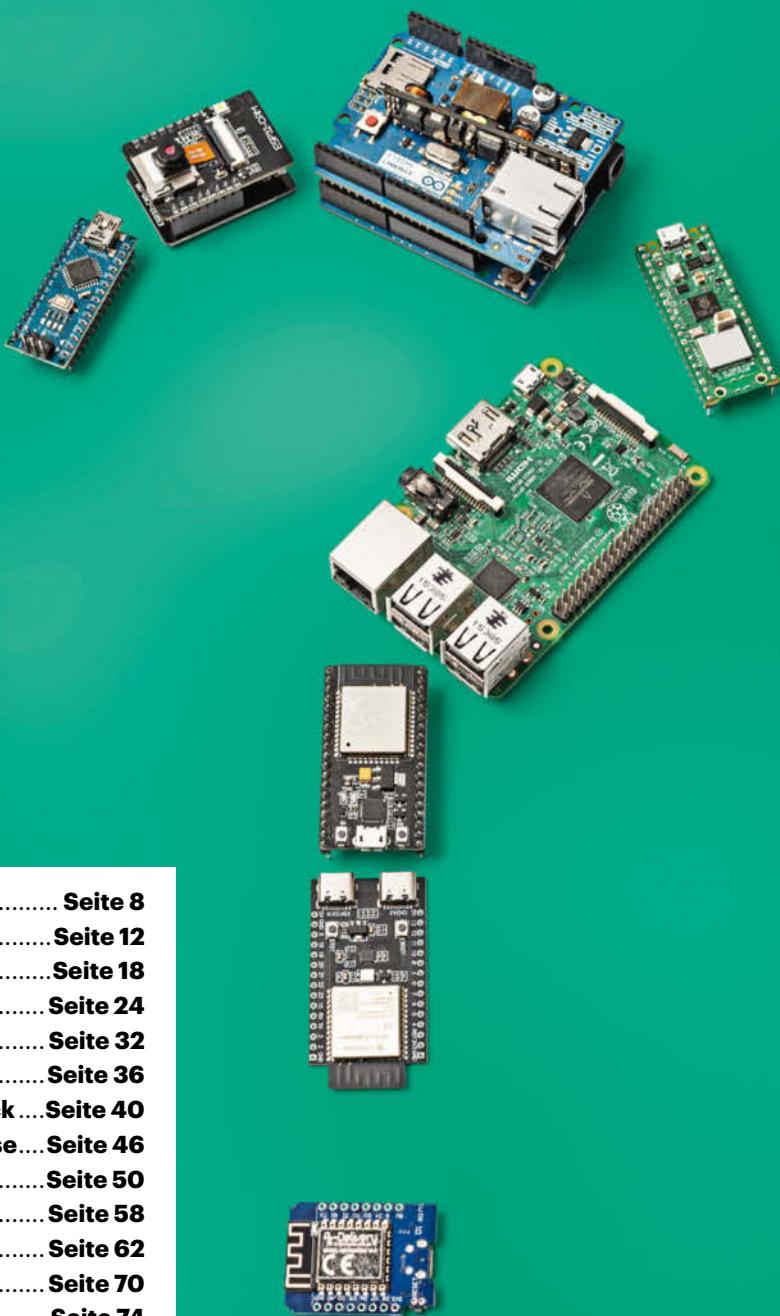




Weitere Schlagseiten auf [ct.de/schlagseite](http://ct.de/schlagseite)

# Aufbruch ins Bastelparadies

Einstieg ins Basteln: Wie Sie mit Raspi, ESP & Co. loslegen



<b>Einstiegstipps .....</b>	<b>Seite 8</b>
<b>Arduino, Raspi &amp; Co. ....</b>	<b>Seite 12</b>
<b>Instrumentenbau .....</b>	<b>Seite 18</b>
<b>Begriffe erklärt .....</b>	<b>Seite 24</b>
<b>Bastlerwerkstatt .....</b>	<b>Seite 32</b>
<b>Haftung und Recht .....</b>	<b>Seite 36</b>
<b>Gehäuse per 3D-Druck ....</b>	<b>Seite 40</b>
<b>Raspi-Design-Gehäuse....</b>	<b>Seite 46</b>
<b>Welt der Sensoren .....</b>	<b>Seite 50</b>
<b>Messdaten im Griff.....</b>	<b>Seite 58</b>
<b>Sprache C lernen .....</b>	<b>Seite 62</b>
<b>Git, GitHub, GitLab .....</b>	<b>Seite 70</b>
<b>Buchtipps zum Raspi .....</b>	<b>Seite 74</b>
<b>Web-Tipps .....</b>	<b>Seite 76</b>

# Nie war es einfacher, Ideen mit Digitaltechnik umzusetzen: Die Einzelteile sind spottbillig zu haben. Nötige Software gibt es kostenlos. Weltweit teilen ambitionierte Bastler frei-zügig ihre Kreationen. Folgen Sie uns in diese faszinierende Welt.

Von Peter Siering

**M**otivationen in die Welt der digitalen Bastelprojekte einzusteigen, gibt es viele: Sie sind generiert vom Cloud-Drall der Hersteller, die ihren Sensoren nur mit fragwürdigen Apps Leben einhauchen? Sie hoffen, tot geglaubter Technik einen zweiten Frühling zu verschaffen? Sie wollen Zusammenhänge verstehen oder begreifen Bastelei schlicht als spannenden Zeitvertreib? Sie möchten Ihre Kinder an moderne Technik heranführen? Willkommen!

Vorweggeschickt: Sie brauchen weder einen Lötkolben noch Erfahrung in analoger Schaltungstechnik und auch keinen 3D-Drucker; das alles hilft, aber für den Anfang genügen ein paar Bauteile, Jumper-Kabel und ein USB-Netzteil. Noch einfacher gelingt der Einstieg mit Geräten, die schon in Bastelprojekten verbreitete Technik verwenden, zum Beispiel WLAN-Steckdosen und -Schalter. Bei denen machen es sich Hersteller zunehmend einfach, indem sie Open-Source-Firmware verwenden.

Letzteres zeigt, Basteln ist ein dehnbarer Begriff. Der studierte oder trainierte E-Techniker wird den Begriff anders verstehen als jemand, der erste Eigenbau-Gehversuche unternimmt. Letzterer freut sich über die blinkende LED, die vermessene Heizung, das selbstgestrickte NAS auf dem Raspi oder die nachgerüstete Laderegelung für das E-Auto; ersterer ist erst zufrieden, wenn er oder sie ein Dutzend analoge Bauelemente in kreativer Verschaltung kombiniert hat. Basteln tun beide und Spaß haben sie vermutlich auch.

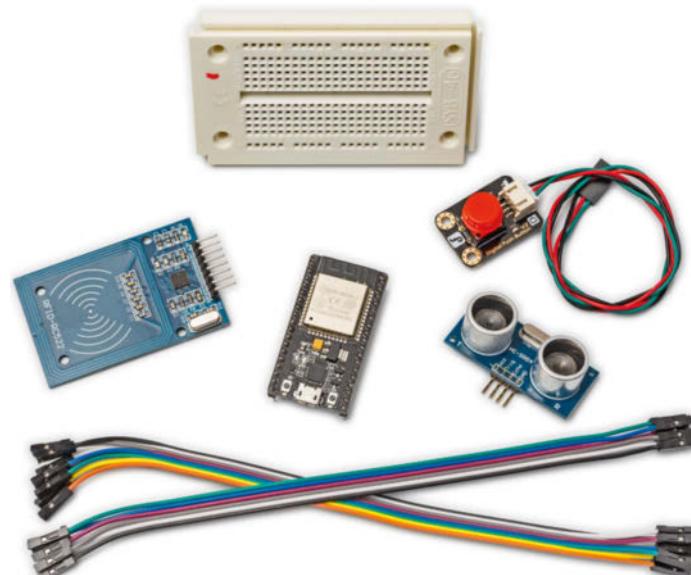
## Schwund gehört dazu

Die einfachste Möglichkeit zu starten besteht darin, dass Sie sich ein fertiges Projekt schnappen. Sei es ein Bausatz, den Sie bei einschlägigen Händlern als Komplettspaket erwerben, oder eine Anlei-

tung, die detailreich genug scheint, um die Einzelteile zu beschaffen. Rechnen Sie in beiden Fällen damit, dass beim Experimentieren ein Teil hopps geht oder womöglich fehlt. Den Schwund sollten Sie einplanen; da es dabei normalerweise nicht um teure Teile geht, ist er meist verschmerzbar.

Viele Projekte sind eine Mischung aus Hard- und Software. Außer die Komponenten zusammenzustecken, gilt es, die Software auf den Mikrocontroller oder Minicomputer zu expedieren. Wie einfach das geht, hängt vorwiegend davon ab, wie verbreitet ein Projekt ist: Oft genügen schon gängige Webbrowser wie Edge und Chrome. Die können fertige Firmware direkt von der Website auf einen Mikrocontroller überspielen.

Etablierte Projekte kennen oft alternative, benutzerfreundliche Methoden, zum Beispiel im laufenden Betrieb Updates über WLAN einzuspielen, oft als OTA abgekürzt (over-the-air). Es kann aber auch sein, dass Sie eine Entwicklungsumgebung bemühen müssen, um die Software eines Projekts in Betrieb zu nehmen.



Für den Bastelaufang genügen üblicherweise die Bauteile und ein paar Jumper-Kabel. Erst wenn diskrete Bauelemente wie Widerstände, LEDs und Transistoren ins Spiel kommen, empfiehlt sich ein Breadboard.

men. Diese Umgebungen gibt es von einfach bis kompliziert. Oft kommt man gerade durch solche Herausforderungen auf den Geschmack nach mehr.

Eine Besonderheit stellen reine Software-Projekte dar, wie sie im Umfeld des Raspberry Pi gängig sind: Das leuchtende Beispiel Pi-hole kennt inzwischen fast jeder, der netzwerkweit wirkende DNS-Filter hält allen Geräten Werbung und manchmal auch darin eingebettete Malware vom Hals. Auch hier greift das Open-Source-Prinzip: Aus von anderen geschaffenen Angeboten lässt sich Neues zusammenstellen, ein Software-Remix sozusagen. Hier liegt die Schwelle noch niedriger, etwas Linux-Know-how genügt oder man erwirbt es unterwegs.

## Widersprüche aushalten

Zugängigen Bastelprojekten finden Sie oft mehrere Anleitungen – das ist hilfreich, weil die unterschiedlichen Perspektiven der Autoren oft erhelltend sind. Rechnen Sie aber damit, dass sich Angaben widersprechen. Bug-Tracker und Foren aus dem Umfeld eines Projekts sind ebenfalls gute Informationsquellen. Auf den ersten Blick mag es mühsam sein, die Rosinen aufzulesen, aber man lernt viel und weitet den Blick.

Die Bastelei lässt sich als Spiel mit mehreren Stufen begreifen: Anfangs bauen Sie schlicht nach oder greifen gleich auf ein fertiges Gerät zurück. Eins höher variieren Sie fertige Projekte für die eigenen Bedürfnisse, ergänzen zum Beispiel Sensoren oder nutzen bisher nicht vorgesehene Schaltaktoren. Dann verändern Sie

die Software, sodass sie besser für Ihren Anwendungsfall passt. Schließlich fangen Sie an, aus bekannten Dingen Neues zu schaffen.

Diese c't-Ausgabe bietet vor allem Neueinsteigern in die Thematik Hilfe und praktische Tipps. Viele Artikel gehen aber auch tiefer und sollten so auch Leser mit aktualisierten Erkenntnissen versorgen, die alle Stufen bereits durchgespielt haben. Was solch ein Heft nie leisten kann,

selbst wenn es Buchform annimmt: Den kompletten Markt und alle Möglichkeiten abdecken – dazu gibt es zu viele Plattformen, Ideen und Permutationen.

## Der Weg ist das Ziel

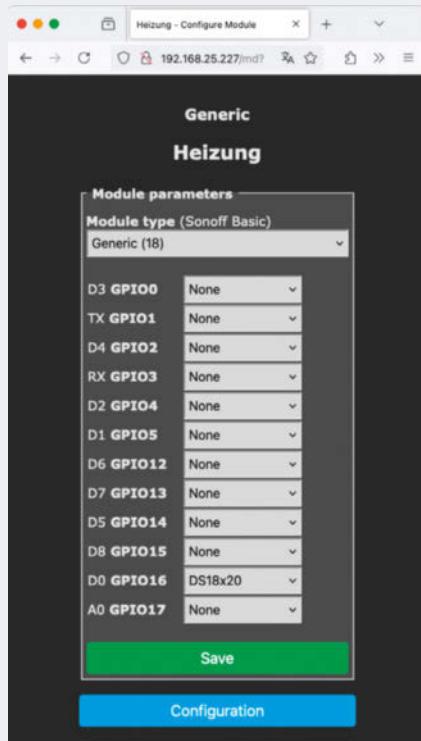
Die folgenden Artikel würdigen deshalb vor allem zwei Plattformen zum Basteln. Zum einen geht es um die ESP-Familie als Mikrocontroller (ab S. 78). Sie kosten so wenig, dass es sich kaum lohnt, für Projek-

te auf klassische Arduinos zurückzugreifen – viele Arduino-Boards sind inzwischen nämlich auch mit Espressif-Chips bestückt. Zum anderen drehen sich die Artikel um die Familie der meist mit Linux betriebenen Raspberry Pis (ab S. 108).

Wir wünschen viel Spaß beim Lesen, Stöbern, Experimentieren und Projektieren. Lassen Sie uns gern wissen, welche Themen wir in kommenden c't-Ausgaben vertiefen sollen. (ps@ct.de) **ct**

## Tausendsassa Tasmota – Firmware für viele Fälle

Besonders für die Familie der ESP-Mikrocontroller muss heute niemand mehr selbst Firmware schreiben. Geht es schlicht darum, mit einem solchen Platinchen Messdaten zu erfassen oder Geräte zu steuern, lohnt ein Blick auf fertige Firmware: Das Open-Source-Projekt Tasmota erfüllt nämlich viele Wünsche dieser Art. Den Grundstein für diese Firmware, die heute fast alle ESP-Modelle unterstützt, legte 2016 Theo Arends.



**Die Weboberfläche eines mit Tasmota versehenen ESP hilft bei der Konfiguration der angeschlossenen Sensoren. Weitere Konfigurationsoptionen erlauben unter anderem das Verbinden mit einem MQTT-Broker, um die Daten dort abzuliefern.**

Arends schrieb für das WLAN-Schaltmodul namens Sonoff Basic eine Cloudfreie Firmware namens Sonoff-MQTT-OTA. Die machte um einen ESP8266 herum gestrickte Geräte per MQTT zugänglich und war obendrein übers WLAN aktualisierbar (OTA). So ließen sich die Geräte in die Heimautomatisierung einbinden, ohne dass man dazu auf Clouddienste zurückgreifen musste.

Das Ersetzen der Herstellerfirmware durch Tasmota in frühen Tagen war mit einigem Aufwand verbunden: Man brauchte einen UART-TTL-Programmieradapter oder einen Raspi, musste die Geräte öffnen und Kontaktstifte einlöten oder geschickt einklemmen. Die Betriebsspannung lieferte währenddessen der Programmieradapter, man setzte sich also keinen allzu großen Gefahren aus.

Schnell fanden sich weitere um den ESP8266 herum aufgebaute Geräte, WLAN-Steckdosen, Schalter, die von freier Firmware profitieren würden. Vor allem betraf das zunächst Geräte, die von der chinesischen Firma Tuya stammten. Die gab es unter diversen Bezeichnungen, teils auch von namhaften Herstellern. So entstand das Projekt „tuya-convert“, das trickreich die Firmware gegen Tasmota ersetzt und ohne Eingriffe in die Hardware auskommt.

Heute ist es noch viel einfacher: Einige Hersteller liefern ihre Geräte gleich mit vorinstalliertem Tasmota aus. Bei vielen lässt sich die gelieferte Version auch mit den in der Firmware vorgesehenen Mechanismen auf aktuellem Stand halten. Wer gehäuft Tasmota einsetzt, kann dafür sogar auf Update- und Administrationshelfer wie „tasmoadmin“ zurückgreifen. Das Open-Source-Projekt liefert eine simple Weboberfläche dafür.



**Mit dem Web-Installer ist das Installieren von Tasmota auf ESPs mit USB-Anschluss ein Kinderspiel. Anschließend fragt der Installer die WLAN-Zugangsdaten ab, pflanzt sie ein und liefert Zugriff auf die Konsole.**

Für die gängigen Dev-Kits auf Basis von ESP8266 und ESP32 gibt es Tasmota als generische Firmware, die sich aus Browern heraus direkt von einer Website installieren lässt; der Browser braucht dazu Zugriff auf den seriellen USB-Port (Edge und Chrome beherrschen das). Man stellt nach dem Flashen dann nur noch ein, um welche Art Modul es sich handelt und welche Sensoren oder Aktoren an die Pins des Moduls angeschlossen sind.

Daraus ist ein Markt entstanden: Auf eBay und anderen Handelsplattformen kann man allerlei fertige Module kaufen, die zum Beispiel auf das Auslesen moderner Stromzähler oder das Zählen von Impulsen von Gasuhren eingestellt sind. Die Anbieter backen dafür oft eine eigene Firmware, die auf den Zweck ausgerichtet ist. Und ergänzen diese um Tasmota-Skripte, die zum Beispiel Zählimpulse in tatsächliche Verbrauchskosten umrechnen.

Τ

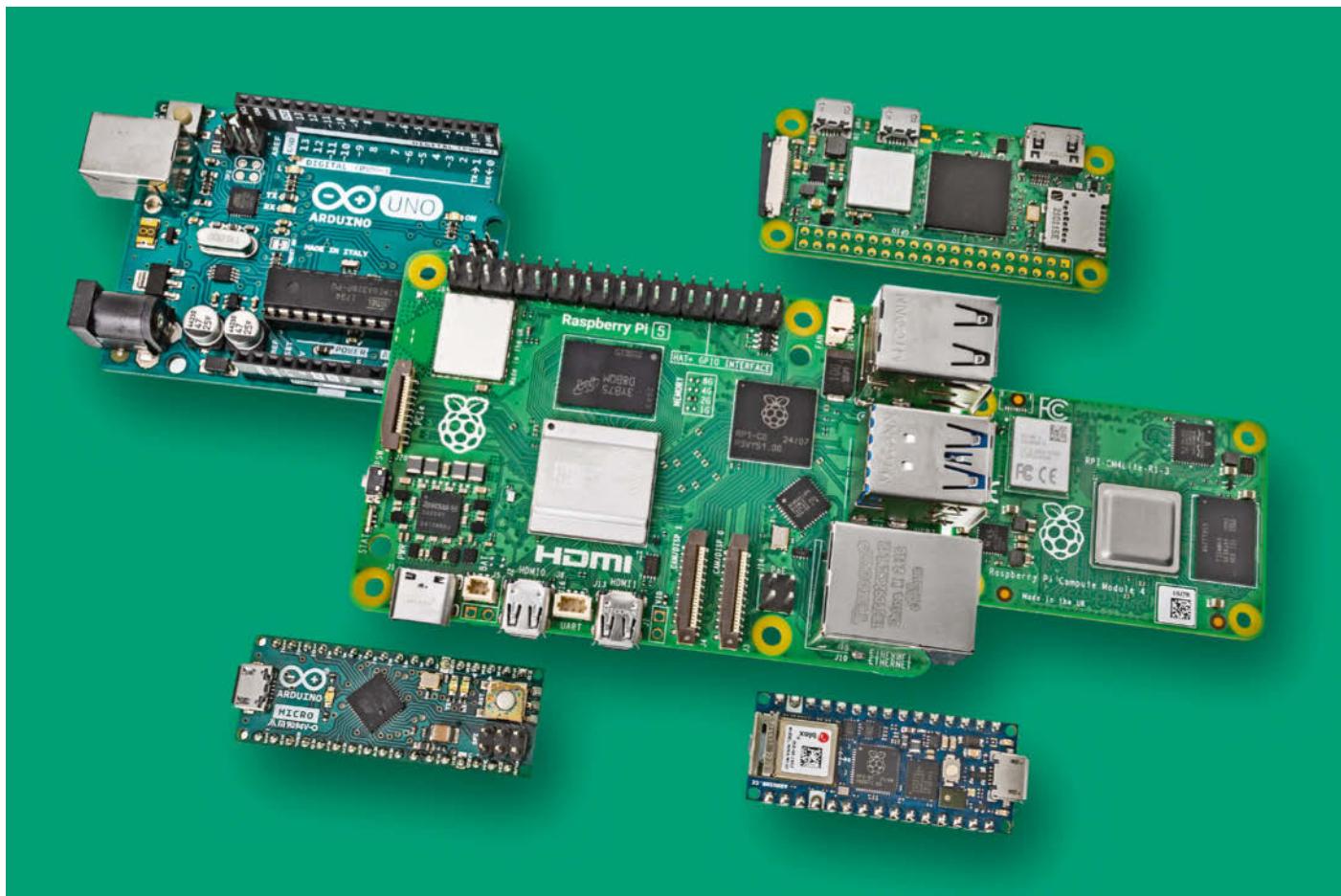
# Home stream home

Streame deine Party überall – nicht nur in der Küche.

45 Jahre Teufel Sound feiern wir am liebsten mit allen. Mit den WLAN-fähigen Speakern STEREO M 2, STEREO L 2 und MOTIV® HOME genießt du über AirPlay2 oder Google Cast synchron deine Musik in jedem Raum. Freue dich auf klanglich überragenden Stereo-Sound und zeitloses Design. **Mehr über die Teufel Home Serie und unsere Jubiläumsangebote erfährst du auf [teufel.de](http://teufel.de)**



45 years  
**Teufel**



# Bastelfreunde

## Welche Vorteile Arduino und Raspi attraktiv machen

**Der Arduino und später der Raspberry Pi veränderten die Welt der IT-Bastler, indem sie zuvor unzugängliche Technik leicht nutzbar machten. Ihre Konzepte ebneten anderen Bastelplattformen wie ESP32 & Co den Weg.**

Von Christof Windeck

**E**ntwicklerboards und Einplatinencomputer gibt es schon seit Jahrzehnten. Doch sie waren teure Technik für Experten, bevor Arduino und Raspberry Pi auf der Bildfläche erschienen. Die erschwinglichen Platinchen machten Mikrocontroller (Arduino) beziehungsweise Mikroprozessoren (Raspi) leicht zugänglich für Schulkinder, Studierende und Bastler. Durch ihren Erfolg und dank ihrer großen Entwicklergemeinden wurden Arduino und Raspi zu Blaupausen auch für andere Bastel-Boards wie die ESP-Familie. Schaut man genauer auf die Geschichte von Arduino und Raspi, zeigen sich viele Gemeinsamkeiten, aber auch wichtige Unterschiede. Wer sie kennt,

trifft bessere Entscheidungen für künftige Projekte.

Sowohl bei Arduino als auch beim Raspi geht es im Kern nicht um Hardware, sondern um Software, Dokumentation und Communities. Die Hardware war bei beiden Plattformen schon vor deren Start vorhanden. Der Clou war jeweils, vorhandene Chips geschickt zu kombinieren und für Hobbyentwickler nutzbar zu machen.

Bevor Massimo Banzi, David Cuartielles und ihre Mitstreiter 2005 den Arduino erfanden, konnten fast nur Spezialisten Mikrocontroller programmieren. Denn dazu brauchte man oft einerseits relativ teure Technik, nämlich etwa JTAG-Programmieradapter und eine kostenpflichtige

## c't kompakt

- Arduino und Raspberry Pi eröffneten Hobbyentwicklern ganz neue Möglichkeiten für den Einsatz von Mikrocontrollern und Linux-Einplatinencomputern.
- Viele Bastelplattformen und Profi-Entwicklerkits orientieren sich an Ideen von Arduino und Raspi.
- Wichtiger als die Hardware ist der gute Software-Support, der riesige Ökosysteme aus Code-Beispielen und Projektideen wachsen ließ.

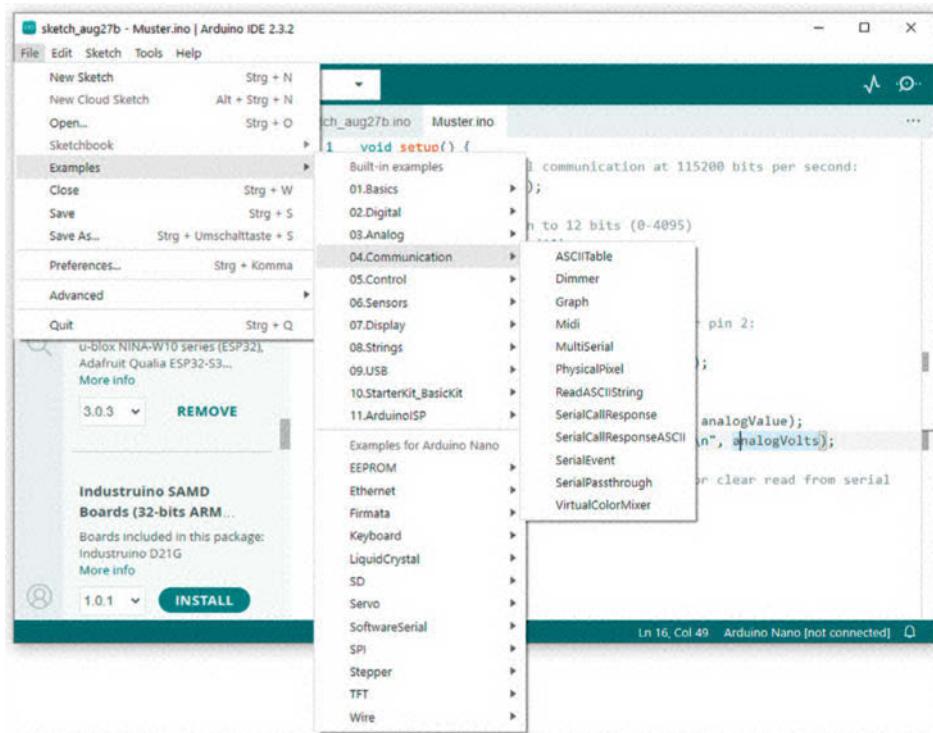
tige Programmierumgebung (Integrated Design Environment, IDE), die zum jeweiligen Chip passt. Andererseits war spezielles Vorwissen nötig, das nicht immer öffentlich zugänglich ist: Manche Hersteller von Mikrocontrollern versorgen nur Partnerfirmen mit detaillierten Datenblättern und Anleitungen für ihre Chips, manchmal sogar nur unter Geheimhaltungsvereinbarung.

Die Arduino-Erfinder rissen diese Mauern ein, indem sie dem preiswerten 8-Bit-Mikrocontroller ATmega8 eine serielle RS-232-Schnittstelle verpassten sowie einen dazu passenden Bootloader und die Arduino IDE schufen: eine verständliche, leicht bedienbare Programmierumgebung. Dazu kam gute Dokumentation. Wenige Jahre später wurde die serielle Schnittstelle durch eine flexible USB-Buchse ersetzt, die manche Arduino-Variante auch gleich mit Energie versorgt.

Was zuvor ein Buch mit sieben Siegeln war, konnte man sich nun mit Neugier und Geduld erarbeiten: das Programmieren eines Mikrocontrollers. Rasch bildete sich eine Gemeinschaft, die Projekte, Schaltpläne und Codebeispiele teilte. Weil der Arduino auch für billige Spaßprojekte wie die Ansteuerung bunter LED-Leuchtstreifen taugt, weckte er das Interesse vieler Privatleute, weit über die Kreise von Elektronikbastlern hinaus. Dadurch wiederum entstand ein neuer Markt, den spezialisierte Firmen wie Adafruit und Sparkfun bedienten und in dem sie rasch wuchsen.

### Arduino-Biotop

Günstige Umstände ließen das Arduino-Biotop rasch gedeihen. Anfang der 2000er-

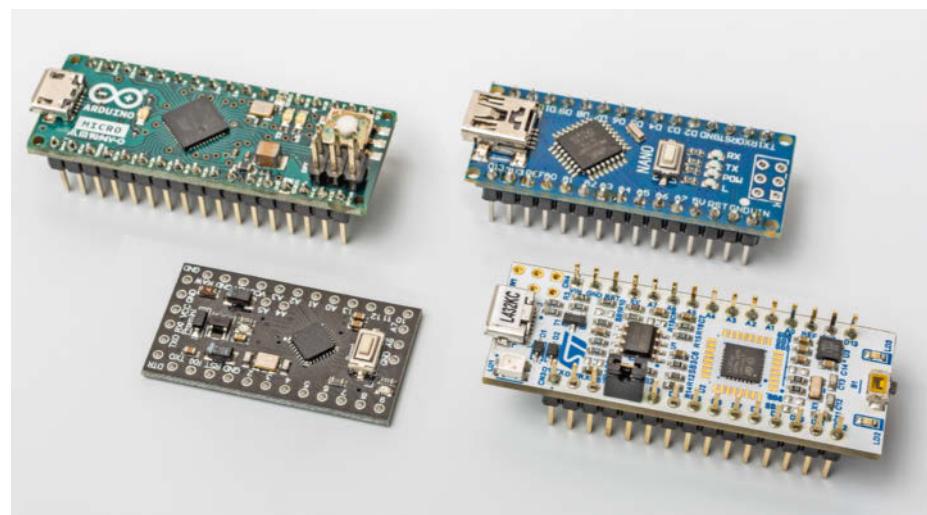


**Die Arduino IDE ist ein mächtiges, aber leicht erlernbares Code-Werkzeug, das zahllose Programmierbeispiele und Bibliotheken für eine gigantische Auswahl an Bastel-Boards erschließt.**

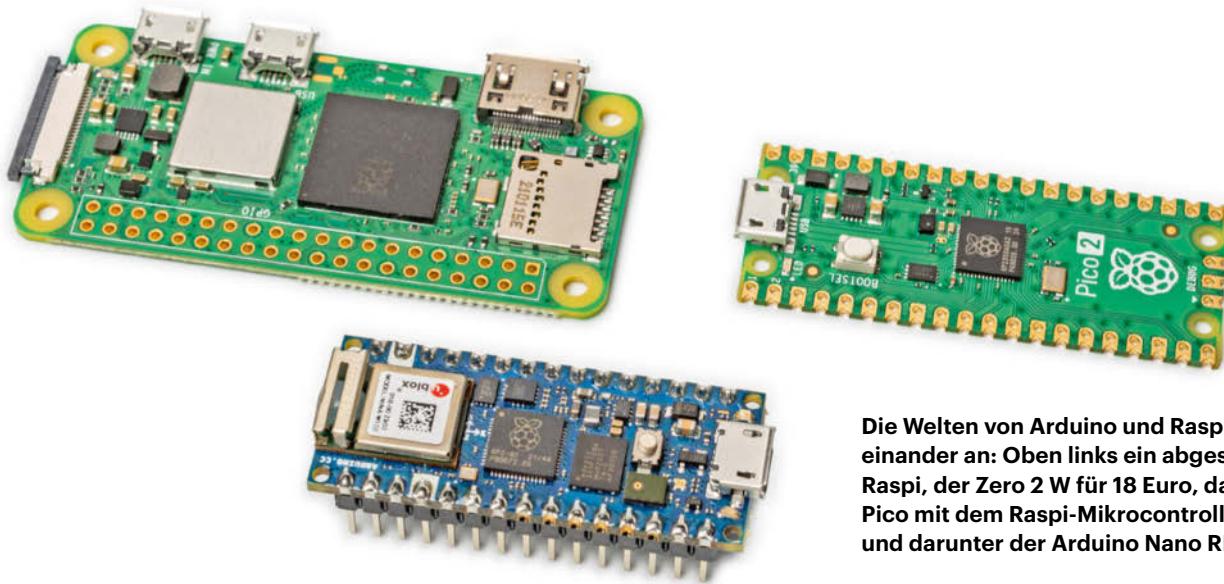
Jahre bekamen immer mehr Menschen schnellen Internetzugang und somit raschen Zugriff auf Software und Dokumentation. Außerdem setzte der Online-Handel zum Höhenflug an, auch mit elektronischen Bauteilen, die früher schwierig zu beschaffen oder teuer waren. Chinesische Shopping-Plattformen wie AliExpress

haben zwar ein kritikwürdiges Geschäftsmodell, eröffneten Elektronikbastlern aber völlig neue Möglichkeiten. Im Gegenzug starben die früher in vielen Städten ansässigen, klassischen Elektrobastlergeschäfte aus: Arlt, Conrad, Völkner ...

Die stetig fallenden Elektronikpreise machten Basteleien für jeden erschwing-



**Für viele Arduino-Projekte muss es nicht das Original (oben links) sein. Oft reicht ein 2-Euro-Nachbau des Arduino Nano (oben rechts) oder eine Micro-Version ohne USB-Port (unten links). Auch Mikrocontroller-Giganten wie STMicroelectronics nutzen das Arduino-Format, rechts unten ein „Nucleo“ mit STM32L432.**



**Die Welten von Arduino und Raspi nähern sich einander an: Oben links ein abgespeckter Raspi, der Zero 2 W für 18 Euro, daneben der Pico mit dem Raspi-Mikrocontroller RP2040 und darunter der Arduino Nano RP2040.**

lich. Großen Anteil hat daran die heute allgegenwärtige USB-Schnittstelle, nicht nur als Daten- und Programmierzugang, sondern auch, weil 5 Volt aus einem spottbilligen oder gebrauchten USB-Netzteil für viele Projekte genügen. Eine wichtige Rolle für den Erfolg digitaler Bastelplattformen spielen Online-Entwicklungstools wie GitHub, das 2008 startete. Hat man das Konzept erst einmal verstanden, gelingt es sehr leicht, offengelegten Code und Programmierbeispiele in eigene Projekte einzubauen.

Gleichzeitig erblühte neue Bastellust: Dale Dougherty gründete 2005 im Silicon Valley das Do-it-Yourself-Magazin Make, 2007 fand die erste Maker Faire statt. In Deutschland erschien ab 2012 das Magazin c't Hardware Hacks, heute Make: Magazin. Für die Löter-Zeitschrift Elrad, aus der 1983 die c't entstanden ist, kam diese Entwicklung 15 Jahre zu spät: sie wurde 1998 eingestellt, aber das nur am Rande.

### Abklatscher

Der Arduino ist Open-Source-Hardware, die Schaltpläne liegen offen. Zwar darf die Marke Arduino nur mit Lizenz verwendet werden. Aber Nachbauten sind erlaubt und erwünscht. Zwischenzeitlich gab es mal Streit zwischen den Arduino-Gründern und den Anbietern einer Marke Genuino, aber grundsätzlich bewährte sich das offene Konzept. Denn zu den unterschiedlichen Varianten des Arduino – Uno, Nano, Leonardo, Mega, Due und so weiter – kamen viele billigere Kopien auf den Markt, teils mit anderen Mikro-

controllern. Bei AliExpress bekommt man einen Arduino-Nano-Nachbau mit chinesischem USB-Adapterchip für unter 2 Euro, wenn man Zehnerpacks kauft. Allerdings braucht man dabei Glück, weil auch unzuverlässiger Ramsch umläuft.

Der Clou ist jedenfalls, dass auch die Billig-Boards mit der gewohnten Programmierumgebung Arduino IDE zusammen spielen. Das ist ein Exempel dafür, dass es oft stärker auf die Software ankommt als auf die Hardware. Arduino-Klone und zahllose Varianten mit Zusatzfunktionen wie WLAN, Akkuladeschaltung oder Anschlussklemmen erweiterten sowohl die Einsatzbereiche als auch die Entwicklergemeinde enorm und damit die Anzahl der Projekte und Bastelvorlagen.

Weil die Arduino-IDE so erfolgreich war, machen sie sich wiederum auch andere Mikrocontroller-Plattformen zu Nutze. Sie programmieren dazu ein kleines Stück Firmware für das jeweilige Board sowie eine digitale Beschreibung des Platinchens für den „Board Manager“ der Arduino-IDE. Das klappt auch mit Boards, die einen ganz anderen Funktionsumfang haben und auf andere Einsatzbereiche zielen. Denn das erschließt den Entwicklern die zahllosen und oft gut gepflegten Softwarebibliotheken (Libraries) etwa für digitale I/O-Schnittstellen, Protokolle und Sensoren.

Eine weitere geniale Idee waren die „Shields“ genannten Erweiterungen, die auf eine simple, billige Pfostensteckerleiste passen. Ohne löten zu müssen – was manche Hobbyisten abschreckt – kann man seinen Bastelcomputer leicht erwei-

tern, etwa um ein Display, mit Sensoren, Schalt- und Steuerplatinen.

Damit Shields funktionieren können, muss die Pinbelegung der Pfostenstecker (Pinout) ebenso festgelegt sein wie die anliegenden Betriebs- und Signalspannungen sowie die jeweilige Strombelastbarkeit. Die typischen Pinbelegungen der unterschiedlichen Arduino-Varianten entwickelten sich zu Quasistandards. Diese nutzen sogar mehrere Chiphersteller bei ihren jeweiligen Development Kits (DevKits) für Profis, etwa STMicroelectronics. Seit rund zehn Jahren gibt es etwa die Nucleo-Boards mit leistungsstarken STM32-Mikrocontrollern. Es zeigte sich rasch, dass auch Profi-Entwickler eher mal eine neue Plattform ausprobieren, wenn das einfach klappt und nicht viel Geld kostet. Man hat damit auch gleich beliebte Werbegeschenke sowie billige Hardware für Schulungen und Fortbildungen.

### Raspberry Pi

Der Raspberry-Pi-Erfinder Dr. Eben Upton griff auf viele Arduino-Ideen zurück, als er mit mehreren Mitstreitern den 2012 vorgestellten Bastelcomputer entwickelte. Er arbeitete seinerzeit als SoC Architect (System on Chip) in der Broadcom-Niederlassung in Cambridge, wo auch ARM ansässig ist, und kümmerte sich besonders um den Grafikprozessor VideoCore. 2009 gehörte er zu den Gründern der Raspberry Pi Foundation, mittlerweile ist er CEO der börsennotierten Raspberry Pi Plc, deren Anteile größtenteils die erwähnte Stiftung besitzt.

Upton wählte für den ersten Raspberry Pi das billige SoC BCM2835, das Broadcom eigentlich für Settop-Boxen verkaufte. Der eingebaute (einzelne) ARM-Prozessorkern war aber mächtig genug, um ein angepasstes Linux auf Debian-Basis auszuführen, genannt Raspbian, heute Raspberry Pi OS.

Der erste Raspi sollte vor allem billig und leicht in Betrieb zu nehmen sein. Erklärtes Vorbild war der BBC Micro aus den 1980er-Jahren, der als Lehrcomputer gedacht war – auch die Raspi Foundation fördert IT-Bildung. Den gigantischen Erfolg des Raspi auch bei Bastlern und sogar bei Firmen sah man seinerzeit nicht voraus.

Schon der erste Raspi hatte eine simple Pfostensteckerleiste mit per Software konfigurierbaren Ein-/Ausgabekontakten, allerdings mit 26 Pins. 2014 kam die 40-polige GPIO-Leiste mit definierter Belegung, die sich dadurch für Hardware Attached on Top (HAT) eignet – Arduino-Shields lassen grüßen.

Ebenso wie der Arduino glänzt der Raspi mit einer geschickten Kombination aus billiger, aber ausreichend starker Hardware mit hoch optimierter Software, Erweiterbarkeit und ausgezeichneter Dokumentation. Rasch bildete sich eine riesige Fangemeinde, die Ideen, Projekte, Schaltpläne und Code austauscht. Mittlerweile sind über 50 Millionen Raspis verkauft worden.

Viele andere Einplatinencomputer imitieren die erfolgreichen Raspis. Typische Kennzeichen sind eine Platine im



**Der Arduino fand seinen Weg sogar schon in professionelle Industriesteuerungen (SPS), beispielsweise in diese von Finder produzierte Arduino Pro Opta für Schaltschränke.**

Format einer Bezahlkarte, USB-Speisung, HDMI-Ausgänge, MicroSD als Massenspeicher und eine 40-polige GPIO-Leiste mit Raspi-kompatibler Beschaltung.

### Keine Raspi-Klone

Während es vom Arduino Hunderte Kopien und Abwandlungen gibt, existieren keine Raspi-Klone. Denn die Raspis sind zwar gut dokumentiert, aber keine Open Hardware; vielmehr behält der seit Juni 2024 an der Londoner Börse notierte Hersteller Raspberry Pi Plc die Fäden in der Hand. Die Raspi-Macher betonen zudem, dass sie die Hardware in Europa unter fairen Arbeitsbedingungen produzieren lassen, und zwar von einer Sony-Fertigungssparte in Wales.

Allerdings werden alte Raspi-Modelle noch jahrelang weiter produziert und dann oft billiger verkauft. Manchmal erscheinen später auch modifizierte Varianten. Daher gibt es mehr als 15 verschiedene Raspi-Typen, siehe auch Seite 108, plus viel Zubehör und Erweiterungen. Zurzeit kann man mit dem Raspberry Pi Zero 2 W für weniger als 20 Euro, zuzüglich Netzteil und MicroSD-Karte fürs Betriebssystem, in die Raspi-Welt eintauchen. Damit klafft weiterhin eine große Preislücke zu chinesischen Arduino-Nano-Nachbauten. Aber ein vollwertiges Linux mit grafischem Desktop braucht eben sehr viel mehr Hardwareressourcen als ein Mikrocontroller, außer stärkeren CPU-Kernen und einem Grafikprozessor auch reichlich RAM und Flash-Speicher.

Die Raspi-Entwickler geben sich große Mühe, Software-Kompatibilität zu wahren. Die 32-Bit-Version von Raspi OS läuft weiterhin auf allen bisher erschienenen Raspi-Varianten. Man steht also nicht unter Druck, fertige Software oder Hardware-Erweiterungen für neue Raspi-Typen zu modifizieren, und kann kompatible Ersatzteile nachkaufen. Das funktionierte nur während der Coronapandemie nicht, als der Chipnachschub einbrach.

Die Raspi-OS-Entwickler bemühen sich zudem, Anpassungen für den Raspi an den allgemeinen Linux-Kernel weiterzutragen (Upstreaming). Ein Raspi läuft deshalb auch mit vielen anderen Linux-Distributionen. Bei dieser breiten Kompatibilität liegt auch der wichtigste Unterschied zu vielen anderen Einplatinencomputern mit ARM-SoCs etwa von Allwinner oder Rockchip. Viele davon benötigen besondere Anpassungen oder Treiber, von denen einige nach wenigen Jahren nicht


**ABECO**

Competence in Hardware-Design

**SMARTE AUTOMATISIERUNG**



**Modular Embedded Box PC**



256 - 370 mm      76 - 209 mm

**EmKITT-3046-W12**

- Modular skalierbarer Box PC**
- Bis zu 5 Erweiterungskarten**
- Dual GPU Option**



- EmKITT 1005 HEL**
- Sehr kompakt**
  - 3 M.2 Slots**
  - Hutschienenmontage**



- Jetson Orin Nano / NX**
- 4 GB - 16 GB**
- 20 - 100 TOPS**

Industrie PCs made in Germany 


**ABECO**

Competence in Hardware-Design

Designstraße 2  
47807 Krefeld  
+49 2151 38712 00  
N51,29100° E6,56565°





zent aller Raspis und Compute Modules (CMs) an Gewerbekunden.

Arduino und Raspi haben für kleinere Firmen und Projekte mit kleinen Stückzahlen erhebliche Vorteile: Die Hardware ist sehr gut dokumentiert und die Codeauswahl gigantisch, dadurch spart man viel Zeit bei der Entwicklung.

Arduino bestückt die Arduino-Pro-Geräte mit STM32-Mikrocontrollern für den industriellen Einsatz. Unter der Marke Arduino Pro Portenta gibt es etwa ein Modul zur Maschinensteuerung. Die italienische Firma Finder, die seit Jahrzehnten Relais fertigt, baut die speicherprogrammierbare Steuerung (SPS, PLC) Arduino Pro Opta im Hutschienengehäuse, die sich in Schaltschränke einbauen lässt. Sie ist außer etwa mit der Arduino IDE auch mit klassischen SPS-Methoden wie Kontaktplan oder Funktionsplan programmierbar.

Raspberry Pi verkauft CMs, die zum Einbau in andere Geräte gedacht sind. Diese Module sowie die Mikrocontroller RP2040/RP2350 haben die Raspi-Macher dafür nach einschlägigen Industrienormen von Prüfinstituten zertifizieren lassen. Außerdem sichern sie lange Verfügbarkeit zu, manche Chips sollen bis nach 2035 erhältlich sein. Mehrere Firmen verkaufen für den industriellen Einsatz ausgelegte Raspis etwa in Hutschienengehäusen, beispielsweise Revolution Pi.

## Erste Wahl

Wer für ein Projekt oder ein neues Gerät einen Mikrocontroller oder Einplatinencomputer benötigt, sollte es zuerst einmal mit Arduino oder Raspi probieren. Denn damit gelingt der Einstieg besonders leicht, die riesige Auswahl an Projekten beschert eine gigantische Auswahl an Codevorlagen, die Dokumentation ist besonders gut und die großen Communities helfen bei Problemen mit Lösungsvorschlägen.

Für Bastelprojekte, die viele Jahre lang funktionieren sollen, etwa im Smart Home, stehen dank der großen Auswahl an kompatiblen Geräten Ersatzteile bereit. Gewichtigster Vorteil ist nach unserer Erfahrung die gute Softwarepflege – zwar nicht perfekt, aber sehr viel besser als bei zahlreichen anderen Plattformen. Die sind nur dann attraktiv, wenn sie essenzielle Vorteile bringen für das jeweils geplante Projekt. Ansonsten geht die Tüftelei mit Arduino oder Raspi meistens deutlich leichter von der Hand. (ciw@ct.de)

**Die Firma Revolution Pi baut Raspberry Pi Compute Modules in Gehäuse für die Hutschiene: Der Raspi arbeitet als Profi-Steuerungscomputer.**

mehr gepflegt werden. Damit eignen sich solche Plattformen eher für erfahrene Linuxer und sind sonst nur attraktiv, wenn größere Distributionen wie OpenWrt sie unter ihre Fittiche nehmen.

Mit den selbst entwickelten Mikrocontrollern RP2040 und RP2350 runden die Raspi-Macher ihr Angebot nach unten ab. Hier sind Nachbauten beziehungsweise besser gesagt Varianten ausdrücklich erwünscht, aber nur mit denselben Chips. Außer den Boards Pico und Pico 2 gibt es mittlerweile mehr als 50 andere von Firmen wie Adafruit, Sparkfun, Pimoroni und Seeed.

## Software ist der Schlüssel

Nicht zufällig lassen sich Mikrocontrollerboards mit RP2040 oder RP2350 auch mit der Arduino-IDE programmieren, vielmehr ist diese Kompatibilität essenziell. Wer sich nicht in die Mikrocontrollerprogrammierung einarbeiten möchte, sondern einfach nur eine Anleitung nachbastelt, hat mit der Arduino-IDE gute Chancen, das zu schaffen. Hat man dann die wesentlichen Schritte verstanden, gelingt auch der Wechsel zu mächtigeren Programmierumgebungen leichter. Die Raspi-Macher empfehlen etwa Thonny, das unter Raspi OS auf einem Raspi läuft, um Python-Code für Pico oder Pico 2 zu schreiben. Viele Mikrocontroller lassen sich via PlatformIO auch mit dem kosten-

losen Microsoft Visual Studio Code programmieren. Und es geht noch einfacher: Einige ESP-Projekte lassen sich direkt von einer Webseite aus in den Speicher eines per USB angeschlossenen Boards schreiben, siehe Seite 84.

Viele Chip- und Hardwarehersteller haben verstanden, wie wichtig ein gut geschlossenes Software-Ökosystem für den Erfolg ihrer Produkte ist. Ein prominentes Beispiel ist die chinesische Firma Espressif, die mit ihrem besonders billigen Mikrocontroller ESP8266 mit eingebautem WLAN-Adapter zunächst nur auf billige Smart-Home-Produkte wie WLAN-Schaltsteckdosen zielte. 2014 erkannten Hacker das Potenzial dieser Chips für Bastelprojekte. Innerhalb kurzer Zeit schossen Projekte wie die alternative Tasmota-Firmware zur Ent-Cloudung von ESP8266-Funkschaltern aus dem Boden. Das wiederum regte Espressif dazu an, viel mehr Dokumentation sowie Codebeispiele bereitzustellen, etwa auch via GitHub.

## Profi-Hardware

Sowohl Arduino als auch Raspberry Pi Plc verkaufen ihre Chips und Plattformen längst auch an Firmen, also an gewerbliche Nutzer. Das soll langfristig planbare Einnahmen bringen, um die teure Entwicklung neuer Plattformen zu finanzieren. Laut Börsenprospekt von Raspberry Pi Plc. verkaufte man 2023 bereits 72 Pro-

**DIE ERSTE EXTRA  
BLACK WEEK  
VOM 25. - 29.11.**

**Jetzt vormerken  
und als Erster  
dabei sein!**

**... wenn  
der Nebel  
sich lichtet**



[extracomputer.de/black-week](http://extracomputer.de/black-week)



# Wilder Minimalismus

## Interview: Vlad Kreimer über den experimentellen Instrumentenbau von Soma Laboratory

**Klaviertasten? Langweilig! Soma baut seine Synthesizer mit Metallsensoren in Baumscheiben und verdrahtet Drumcomputer mit Krokodilklemmen. Im Interview erzählt Soma-Gründer Vlad Kreimer, wie er auf seine bizarren Kreationen kommt und was Bastler von seinem russischen Erfindergeist lernen können.**

Von Hartmut Gieselmann

**U**ngewöhnliche Musikinstrumente haben in Russland eine lange Tradition. Bereits vor mehr als hundert Jahren stellte Leon Theremin den nach ihm benannten Synthesizer der staunenden Öffentlichkeit vor. Und noch heute fasziert das Theremin, wenn Virtuosen die Töne mit ihren Händen in der Luft dirigieren, ohne das Instrument auch nur zu berühren.

Der Forscherdrang ist ungebrochen. Gerade weil die Versorgung mit Konsumgütern und Bauteilen schwieriger ist als in vielen westlichen Industrienationen, müssen Musiker und Instrumentenbauer improvisieren und die vorhandenen Ressourcen besonders effizient nutzen. Das gilt insbesondere für elektronische Hardware und Synthesizer: Wer Bauteile aus alten Küchengeräten oder lärmendem Kinderspielzeug ausschlachtet, muss sich

schon sehr gut mit Schaltungen auskennen oder sich etwas Neues einfallen lassen, wenn für Bob Moogs berühmtes Laddern-Filter ein paar wichtige Transistoren fehlen.

Die meisten dieser skurrilen Kreationen sind handgefertigte Einzelstücke, die in kleinen subkulturellen Kreisen in Russland zirkulieren. International hat Soma Laboratories in den vergangenen Jahren mit seinen ungewöhnlichen und oft bizarren Instrumenten Kultstatus erlangt. Gründer Vlad Kreimer hat es in nur wenigen Jahren vom Hobbybastler zum weltweit renommierten Instrumentenbauer gebracht. Seine Synthesizer sind – wenn auch in kleinen Stückzahlen – in allen Winkeln der Welt zu hören. Im Interview erklärt er seinen Designansatz, was ihn an westlichen Synthesizern stört und warum weniger oft mehr ist.

**c't: Herr Kreimer, vor acht Jahren haben Sie Soma Laboratories gegründet. Wie kam es dazu?**

**Vlad Kreimer:** Das kam unerwartet. Früher habe ich als Musiker und Live-Performer Instrumente für mich selbst gebaut. Davon habe ich Videos gemacht – nur so zum Spaß. Die Resonanz wurde immer größer. Die Leute fingen an, mir immer mehr Fragen zu stellen. Irgendwann beschloss ich, ein gutes Demovideo für meinen Synthesizer Lyra-4 zu machen: den Vorgänger des Lyra-8, der später der erste bekannte Soma-Synthesizer werden sollte.

Ich hatte nur eine einfache, persönliche Seite in den russischen sozialen Medien. Deshalb war ich so überrascht, als ich Anfragen aus dem Ausland bekam, sogar aus den USA. Das hat mich so beeindruckt, dass ich später ein aufwendiges Demovideo des ersten Lyra-8 mit englischen Untertiteln für YouTube gemacht habe. Die Resonanz war enorm, vor allem wenn man bedenkt, dass es keine Werbung gab, sondern nur meine persönliche Seite mit ein paar hundert Followern.

Ich bekam Dutzende von Anfragen von Zuschauern, ob ich das Instrument nicht auch für sie bauen könnte. Nachdem ich etwa 50 Vorbestellungen hatte, wurde mir klar: Das ist ein Geschäft und eine gute Tätigkeit für mich.

Also hörte ich auf, Musiker und Produzent zu sein, und begann, Synthesizer zu bauen. Normalerweise fangen die Leute mit einer Finanzierung oder Forschung an und kommen dann langsam zum Erfolg. Bei mir war es umgekehrt. Ich



**Vlad Kreimer hat über ein Dutzend elektronische Instrumente und Effekte entwickelt, die höchst ungewöhnliche Klänge produzieren und sich eigenwillig spielen lassen. Bei unserem Videointerview hielt er einen frühen Prototyp von Terra im Arm.**

hatte Erfolg und fing an, eine Firma zu gründen, Geld aufzutreiben und Produktionsanlagen zu bauen.

**c't: Es gibt viele Entwickler in Russland, die elektronische Instrumente bauen. Der Dokumentarfilm „Electro Moskva“ zeigt zum Beispiel äußerst kreative Elektroniker, die aus Müll und billigem Elektronikspielzeug aus China neue Synthesizer basteln. Doch kaum einer von ihnen ist über die Landesgrenzen hinaus bekannt. Warum ist das internationale Geschäft so schwierig?**

**Kreimer:** Weil es viel mehr Arbeit macht, ein internationales Unternehmen zu sein als ein lokales. Das gilt nicht nur für russische Unternehmen, sondern für alle. Innerhalb der Europäischen Union zum Beispiel ist es viel einfacher, weil man sich im gleichen Finanzraum, im gleichen Zollraum befindet. Wenn man also ein Geschäft außerhalb Russlands betreiben will, braucht man dort entweder sehr gute Unterstützung oder man muss eine Niederlassung in einem anderen Land haben.

Für Soma war das ein ziemlich schneller Prozess. Am Anfang war Soma ein rein russisches Unternehmen. Als ich die Lyra-8 ein halbes Jahr lang verkauft hatte, fand ich einen sehr netten Partner in Europa, in Polen, und wir eröffneten die europäische Niederlassung von Soma. Heute stellen

wir den größten Teil unserer Instrumente in Europa her.

Die große Mehrheit unserer Kunden lebt außerhalb Russlands. Wir haben ein sehr großes Netz von etwa 200 Geschäften. Das reicht von großen Häusern wie Thomann oder Schneidersläden in Deutschland bis hin zu winzigen Boutiquen in Australien oder Italien, die vielleicht nur ein paar Instrumente pro Jahr verkaufen.

## Ingenieurskunst und Schamanismus

**c't: Was unterscheidet Ihren Ansatz von dem anderer Synthesizer-Hersteller?**

**Kreimer:** Ich arbeite interdisziplinär. Einerseits bin ich ein logisch denkender Ingenieur. Auf der anderen Seite bin ich aber auch ein wilder Mensch, der sich mit Schamanismus beschäftigt, viel meditiert, spirituell lebt und Entscheidungen aus dem Unterbewusstsein heraus trifft. Ich sehe keinen Widerspruch, sondern eine Harmonie zwischen diesen beiden Polen.

Als Musiker weiß ich auch, was ein Musiker braucht, und verstehst seine Perspektive. Ich kenne viele Toningenieure, die sich nur um technische Aspekte wie die Latenz eines Audio-Interfaces kümmern. Aber Musiker interessiert dieser ganze technische Kram nicht besonders. Sie brauchen Inspiration, sie brauchen Gefühle.

Musik machen ist keine rationale Tätigkeit, es ist irrational. Und deshalb brauchen Musiker irrationale Werkzeuge. Die



**Lyra-8 war der erste bekannte Synthesizer von Soma. Er besteht aus acht Oszillatoren, die sich gegenseitig beeinflussen und über Feedbackschleifen sowie ein frei einstellbares Delay laufen. Der Spieler kann die Tonhöhe jedes Oszillators frei stimmen und die Töne über Metallkontakte mit den Fingern spielen.**

## c't kompakt

- Soma ist der bekannteste Hersteller von experimentellen Synthesizern aus Russland.
- Entwickler Vlad Kreimer startete als Hobbybastler und ließ sich ungewöhnliche Schaltungen einfallen, die sich am Minimalismus primitiver Organismen orientieren.
- Statt möglichst viele Funktionen in digitale Instrumente einzubauen, konzentriert sich Soma auf das Notwendigste und verknüpft einfache Schaltungselemente zu komplexen Systemen.



**Terra ist der erste Soma-Synthesizer mit einer digitalen Tonerzeugung. Dank der 32 Synthesizer-Engines klingt das Instrument mal wie ein Streichinstrument, mal wie ein Chor. Die rechte Hand spielt die Töne über sensible Metallkontakte. Je mehr Haut diese berührt, desto lauter der Ton. Die linke Hand steuert Effekte und ändert die Klangfarbe. Über ein eingebautes Gyroskop lässt sich die Tonhöhe beim Spielen fein nuanciert ändern.**

beliebtesten Geräte auf dem Musikmarkt sind zum Beispiel Gitarreneffekte. Und der König der Gitarreneffekte ist das Verzerrerpedal – das ist nichts anderes als ein kaputter Verstärker.

#### c't: Und wie irrational ist der Lyra-Synthesizer?

**Kreimer:** In der Natur werden viele komplexe Systeme aus relativ einfachen Bausteinen zusammengesetzt. Das Gehirn einer kleinen Fruchtfliege hat nur etwa 500 Neuronen. Mich hat schon immer fasziniert, wie ein so kleiner Biocomputer ein so komplexes Verhalten steuern kann: fliegen, navigieren, Nahrung finden, sich fortbewegen und so weiter.

Der Lyra-Synthesizer ist aus einfachen Blöcken mit vielen Rückkopplungsschleifen aufgebaut. Sie kennen vielleicht das Doppelpendel: Ein Pendel schwingt sehr präzise und vorhersagbar. Setzt man ein weiteres Pendel an das untere Ende des ersten Pendels und lässt es gemeinsam schwingen, entsteht eine sehr komplexe Schwingung, die sich nicht mit einfachen linearen Gleichungen beschreiben lässt.

Den gleichen Ansatz verfolgte ich bei den organismischen Strukturen der Lyra. Sie verhält sich unvorhersehbar und lädt den Musiker ein, sie zu erforschen.

Meiner Meinung nach ist die Unvollkommenheit der alten analogen Synthesizer ihre größte Stärke. Deshalb mag ich, ehrlich gesagt, moderne Synthesizer nicht. Wenn man sie hundertprozentig digital steuert, tötet man ihre natürlichen, wilden Eigenschaften.

#### c't: Für Musiker ist es aber schwierig, bestimmte Klänge mit der Lyra zu reproduzieren. Stört das nicht bei Live-Auftritten?

**Kreimer:** Das gehört zur Philosophie. Ein deutscher Musiker hat einmal gesagt, Lyra

mache ihn unsicher. Er kann sich nicht hinter Presets verstecken, sondern muss auf der Bühne eine Beziehung zum Instrument aufbauen. Das macht den Auftritt umso spannender. Viele Leute vermissen das. Unsere Welt ist heute zu logisch, zu berechenbar.

#### Teure Tasten

**c't: Ihre Instrumente wie die Lyra oder auch das neue Terra haben elektrische Kontakte, die man mit den Fingern spielt. Warum verwenden Sie keine gewöhnliche Klaviatur, mit der Musiker bereits vertraut sind?**

**Kreimer:** Zum einen ist eine Klaviatur ziemlich teuer, und wir produzieren unsere Instrumente nicht billig in China. Vor allem aber brauchen experimentelle Musiker eine neue Art des Musizierens. Sie versuchen zum Beispiel, einer Geige ungewöhnliche Töne zu entlocken, indem sie an bestimmten Stellen zupfen oder klopf-

fen, die ursprünglich nicht dafür vorgesehen waren. Solche Möglichkeiten baue ich auch in meine Instrumente ein.

Vor etwa 15 Jahren war ich mit der chromatischen Tonleiter unzufrieden. Denn sie hat als einziges Intervall den Halbtontschritt, der sich zwölffach wiederholt. Ich hörte ihn überall, in jeder Art von Musik, und er fing an, mich zu stören.

Die Lyra dagegen hat nur acht Oszillatoren, die man während des Spielens frei stimmen kann, sodass man jedes Intervall spielen kann. Es fördert die natürlichen musikalischen Fähigkeiten und das Verständnis für Musik. Meine Instrumente sind sehr eingängig. Viele Kinder machen von der ersten Minute an etwas Interessantes damit.

#### c't: Wie sieht bei Ihnen der Prozess vom Prototyp bis zum fertigen neuen Instrument wie dem Terra-Synthesizer aus?

**Kreimer:** Terra ist im Unterschied zu Lyra ein digitaler Synthesizer. Die sind viel schwieriger zu entwickeln als analoge, weil sie fast unendlich viele Möglichkeiten bieten. Vor allem, wenn sie ein Display haben, mit dem man so tief in die Menüs eintauchen kann, wie man will, und alles einbauen kann, was man weiß.

Das größte Problem in der heutigen Welt der Technik: Wie kann man das Unnötige weglassen und nur das wirklich Notwendige behalten? Denn die Technik ist viel weiter als unsere Fähigkeit, sie zu nutzen. Niemand kennt alle Möglichkeiten seiner digitalen Audio-Workstation

Dieser Prototyp von Terra bestand im Innern noch aus vielen handgelöteten Kabeln, weil sie sich schneller variieren lassen als eine fertig layouerte Platine.



Bild: Vlad Kreimer

oder einer Software wie Massive von Native Instruments mit 500 Reglern. Selbst die besten Profis kennen kaum 20 Prozent der Funktionen. Zum Vergleich: Gitarristen kennen ihr Instrument in- und auswendig und wissen sogar, was passiert, wenn sie mit dem Plektrum über die Saiten jenseits des Stegs streichen.

Das Schwierige bei der Entwicklung eines neuen Instruments ist, eine gute Verbindung zwischen dem Musiker und dem Instrument herzustellen. Deswegen sind meine Instrumente so minimalistisch.

Wenn ich mich als Musiker mit komplizierten Menüs und der Eingabe von Zahlen über Tasten beschäftigen muss, verliere ich meine Inspiration. Deshalb war es für Terra so wichtig, zunächst das Interface ohne Display so minimalistisch zu gestalten, dass man sofort losspielen kann, ohne programmieren zu müssen.

## Inspiration aus dem Wald

**c't:** Wie sind Sie auf die Idee gekommen, Terra in eine Baumscheibe einzubauen?

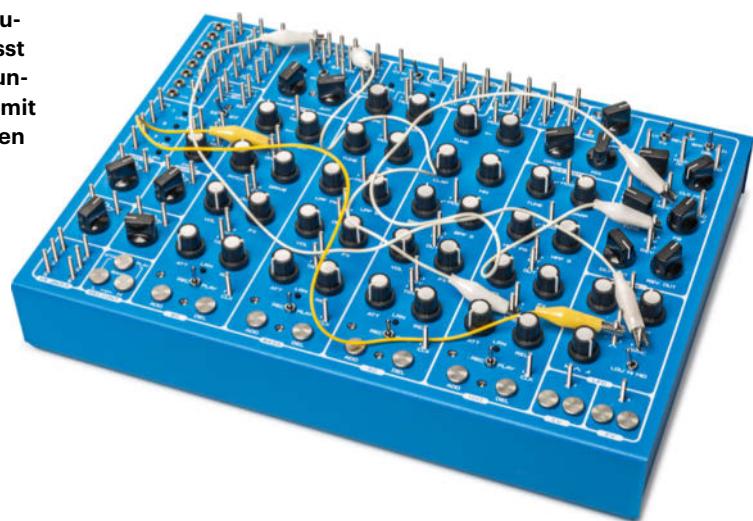
**Kreimer:** Ich war mit meiner Frau Runa, der Designerin von Terra, im Wald spazieren. Wir fanden ein sehr schönes Stück Holz und träumten davon, wie ein Synthesizer die Form eines wilden Holzstücks aus dem Wald haben könnte. Ich stellte mir vor, dass das Instrument nicht mit den üblichen Reglern, sondern mit Sensoren gesteuert werden sollte.

Die erste Idee war, es mit analogen Schaltungen zu bauen. Aber ich habe schnell gemerkt, dass das aus Platzgründen nur für eine Krachmaschine funktionieren würde. Viel besser wäre ein digitaler Kern mit einem minimalistischen Interface.

Das erste Gehäuse für Terra ist auf dem Foto zu sehen. Dann kam mir die Idee, ein Interface zu entwickeln, das der natürlichen Form der Hände entspricht. Ich mache nur kleine Bewegungen mit den Fingern und kann trotzdem fünf oder sechs Oktaven spielen. Das ist ziemlich viel, wenn man bedenkt, wie einfach das Interface ist.

Es war viel Forschungsarbeit nötig, um diese Sensoren zu entwickeln, die sehr sensibel und einfach zu spielen sind. Ich mag keine tauben Instrumente, die man stark drücken muss. Terra reagiert fast wie ein akustisches Instrument auf kleinste Bewegungen. Man kann hören, wenn die Hand zittert und man ein bisschen nervös ist.

**Der Drumcomputer Pulsar 23 lässt sich auf über hundert Kontakten mit Krokodilklemmen verdrahten.**



Ich habe zum Beispiel 30 Soundchips wie Analog-Digital-Codecs und Wandler ausprobiert, um den besten für meine Zwecke zu finden. Obwohl wir nur ein kleines Team sind, hat die Entwicklung nicht lange gedauert. Vom Waldspaziergang mit meiner Frau bis zum Verkauf des ersten Terra-Synthesizers vergingen vielleicht eineinhalb Jahre.

**c't:** Mit welcher Software haben Sie die Platinen entworfen und den Mikrocontroller programmiert?

**Kreimer:** Für den Terra haben mein Mitentwickler Maxim Manakov und ich die EWARM-Entwicklungsumgebung (Embedded Workbench for ARM, Anm. d. Red.) für den ARM-Mikrocontroller von IAR verwendet. In den ersten Entwürfen für Terra wurde noch viel Draht verwendet, da Platinen keine schnellen Änderungen zulassen. Als die Schaltungen standen, haben wir eine ältere Version der EAGLE-Software für das Platinenlayout verwendet. Außerdem viele andere Standardprogramme für die Erstellung von 3D-Prototypen aus Holz, für Beschriftungen und Vektorzeichnungen.

Ich benutzte hauptsächlich meine eigene Spielerfahrung. Später haben uns Freunde, Betatester und Reviewer wie Loopop Feedback gegeben. Aufgrund des Feedbacks haben wir zum Beispiel einen zusätzlichen Hold-Sensor für die mit der rechten Hand gespielten Töne eingebaut.

**c't:** Setzen Sie auch bei Bastlern beliebte Plattformen wie Arduino, ESP oder Raspi ein?

**Kreimer:** Nein. Wir nutzen zu hundert Prozent nur unseren eigenen, handgeschrie-

benen Code, um die volle Kontrolle über jeden Befehl zu haben. Nur so können wir die Qualität und Zuverlässigkeit unserer Instrumente garantieren. Plattformen wie Arduino & Co. setzen ausgiebig auf riesige Bibliotheken, die von einer Community gepflegt werden. Das ist gut für Amateurprojekte, aber nicht für Profis.

## Wie klingt ein Stück Käse?

**c't:** Viele internationale Hersteller modularer Synthesizer verwenden das von Dieter Döpfer entwickelte Eurorack-Format. Dabei werden kleine Klinkenkabel in die Module gesteckt, um Steuerspannungen und Musiksignale zu übertragen. Ihre Instrumente wie der Drumcomputer Pulsar 23 oder der Sequenzer Ornament 8 verwenden dagegen Kabel mit Krokodilklemmen. Warum?

**Kreimer:** Ganz einfach: Der Pulsar 23 hat 118 Anschlusspunkte. Wenn ich im Eurorack-Format arbeiten will, brauche ich 118 hochwertige Miniklinkenbuchsen. Denn die Mechanik ist das schwächste Glied in jedem Synthesizer, sie knistert schnell oder gibt das Signal nicht richtig weiter.

Im Vergleich dazu ist ein Pin nur ein Stück Metall, viel billiger als eine gute Buchse. Beim Pulsar 23 konnten wir die Kosten sogar um rund 500 Euro senken, indem wir die speziellen Pins der ersten Generation durch einfache Schrauben ersetzten. Mir ist es lieber, etwas herzustellen, das auch für junge Leute in weniger reichen Ländern erschwinglich ist. Wir haben zum Beispiel Kunden in Afrika, China und Korea, die nicht so viel Geld haben.

Außerdem braucht so ein Pin viel weniger Platz, nur etwa ein Fünftel bis ein



Bild: Soma

**Der Lyra-8-Synthesizer ist ausschließlich aus analogen Bauteilen aufgebaut. Wer mag, kann sich ein Kit bestellen und ihn selbst zusammenlöten.**

Viertel einer Buchse. An einem Pin kann ich vier bis fünf Klemmen anbringen, was bei Eurorack-Steckern viel schwieriger ist. Nicht zuletzt kann ich mit Circuit Bending und Body Patching alle möglichen Haushaltsgegenstände anschließen: Gabeln, Löffel, ein Stück Käse oder ein altes Radio. Die Leitfähigkeit des Körpers reicht aus, um verschiedene Punkte zu verbinden. Natürlich sind auch Eurorack-Verbindungen über die eingebauten Adapter von Pulsar 23 und Ornament 8 möglich. Die Spannungen liegen in einem ähnlichen Bereich von 0 bis 10 Volt.

**c't: Das klingt nach viel Bastelspaß. Kann man Ihre Synthesizer auch mit Bausätzen selbst zusammenbauen und modifizieren?**

**Kreimer:** Pulsar 23 und Terra sind zu komplex. Es ist zum Beispiel unmöglich, einen aktuellen SMD-Chip von Hand mit einem einfachen Lötkolben zu löten. Dafür braucht man eine professionelle Lötzstation und besondere Fähigkeiten.

Aber die Lyra-8 ist ideal und wir verkaufen sie als Selbstbausatz mit Anleitung

für 100 Dollar. Es macht wirklich Spaß zu sehen, was die Leute daraus machen: von Holzgehäusen im Steampunk-Stil bis hin zu Metallgehäusen aus computergesteuerten CNC-Fräsen.

### Universelle Sprache

**c't: Sie arbeiten auch als Autor und Philosoph und haben vor vier Jahren das Buch „Redefining Conscience in an Era of Rapid Change“ kostenlos online veröffentlicht, in dem Sie sich kritisch mit Politik, Medien und Gesellschaft auseinandersetzen. Sie schreiben, ein großer Teil der Gesellschaft sei egozentrisch. Sie streben dagegen eine weltzentrische Denk- und Lebensweise an. Wie können Ihre Musikinstrumente dazu beitragen?**

**Kreimer:** Die schlimmste Krankheit unserer Zivilisation ist, dass wir ständig gegeneinander kämpfen, ob innerhalb eines Landes oder zwischen verschiedenen Ländern. Wenn ich als Ingenieur die Menschheit als einen Mechanismus betrachte, sagen wir als ein Getriebe, dann ist es nicht

gut, wenn es mit sich selbst kämpft und viel Energie darauf verwendet, verschiedene Teile seiner eigenen Struktur zu zerstören.

Deshalb ist es wichtig, eine gemeinsame universelle Sprache zu finden, die für jede Kultur und jede Weltanschauung geeignet ist. Musik ist vielleicht die beste Sprache für einen interkulturellen Austausch, weil sie universell ist. Ich kann zum Beispiel afrikanische Musik hören, ohne eine spezielle Ausbildung zu haben, die ich bräuchte, um die Sprache zu verstehen.

Es gibt keine isolierte Musikkultur. Wir lernen alle voneinander. Es ist ein ständiger Austausch von Ideen und wir lassen uns von Kulturen aus verschiedenen Ländern inspirieren. Soma hat zum Beispiel eine buddhistische Herangehensweise an Bewusstseinsbildung und Meditation, eine europäische Herangehensweise an die postmoderne Kultur, eine postmoderne Herangehensweise an die Erzeugung von Klangfarben, eine muslimische Herangehensweise an die Erzeugung von Mikroklangen, eine russische Herangehensweise an den Aufbau emotionaler Beziehungen und so weiter. In unserem Unternehmen kommen also viele Kulturen zusammen.

### c't: Was können andere von Soma lernen?

**Kreimer:** Wir sollten alle versuchen, weniger technisch und weniger geschäftsortientiert zu sein, sondern mehr humanistisch. Wenn man die junge Generation, die mit Social Media aufgewachsen ist, mit der älteren vergleicht, dann liest sie viel weniger Bücher. Sie hat die Fähigkeit verloren, komplexe, tiefgründige und lange Informationen aufzunehmen, die viel Konzentration und Zeit erfordern. Wenn ich eine Art Internet-Gott wäre, würde ich es mir zweimal überlegen, ob ich soziale Medien in der Form, wie wir sie heute haben, freigeben würde.

Die westliche Gesellschaft mag keine Selbstbeschränkung. Aber um etwas Wertvolleres zu erreichen, muss man sich selbst beschränken – und zwar von Herzen, nicht weil jemand von außen Druck ausübt. Vergessen Sie nicht, dass es eine sehr alte, aber immer noch sehr wahre Aussage ist, dass alles, was funktioniert, Disziplin erfordert.

(hag@ct.de) **ct**

**Bauanleitung Lyra-8:** [ct.de/yegx](http://ct.de/yegx)

# DENN SIE WISSEN NICHT, WOHIN DU GEHST.

Neue Wege in neue Arbeitswelten.  
Sicherer. Einfacher. Besser.

## Cordaware bestzero

Die unkomplizierte VPN Alternative  
für remote work in unsicheren Zeiten

Erhebliche Reduzierung von Cyber-Bedrohungen.  
Dedizierter Zugang für einzelne Anwendungen.  
Zeitsparend, kosteneffizient, flexibel.

Remote Work  
Ganz einfach und  
ohne VPN!

- ✓ Zeitgesteuerter und 2FA bedingter Appzugriff
- ✓ Keine offenen eingehenden Ports erforderlich
- ✓ Remote Zugriff auf Ressourcen schnell und einfach bereitstellen



Cordaware GmbH Informationslogistik +++ Fon +49 8441 8593200 +++ info@cordaware.com +++ www.cordaware.com

Verfügbar für Windows, macOS, Linux, iOS und Android

© Copyright by Heise Medien.



IT Made in Germany



# Akü-Fighter

## Ein Glossar zu den wichtigsten Schnittstellen und Protokollen rund um Mikrocontroller

**Der ESP32 hat nicht nur Thread und BT, sondern oft auch noch I2C, I2S, MCPWM; einige Exemplare können sogar mit PSRAM umgehen! Abgehängt mit nur einem Satz – das muss nicht sein: Unser Glossar erklärt, was sich hinter den Abkürzungen verbirgt und wofür die Schnittstellen und Protokolle oft genutzt werden.**

Von Georg Schnurer

Mikrocontroller gibt es von einfach bis hochkomplex. Vor allem die beliebten SoCs – sorry, Systems-on-Chip – der chinesischen Firma Espressif aus der ESP32-Reihe warten mit sehr vielen Schnittstellen auf und beherrschen zahlreiche Protokolle. Im Datenblatt wie auch in unserer Tabelle auf Seite 83 stehen deshalb sehr viele Abkürzungen, die längst nicht alle selbsterklärend sind. Hier ein Versuch, etwas Ordnung und Klarheit in den Aküfi zu bringen. Wir haben lange überlegt, wie so ein Glossar sinnvoll und nachvollziehbar sortiert werden sollte. Nach Funktionsgruppen, nach unterstütz-

ten Komponenten, nach Protokollen – letztlich entschieden wir uns für eine simple alphabetische Sortierung.

Den Anfang macht – ganz ohne konkreten Mikrocontroller- und Bastelbezug – der **Aküfi**: Beim Abkürzungsfimmel reden Sprachwissenschaftler von einem Determinativkompositum aus den Substantiven Abkürzung und Fimmel mit dem Fugen-s. Das Wort umschreibt häufig abwertend die mitunter übersteigerte Vorliebe mancher Menschen und Branchen für Abkürzungen.

Ein **ADC** ist ein Eingang eines Mikrocontrollers, der ein analoges Signal – übli-

cherweise eine Spannung – in ein digitales wandelt. Die Abkürzung kommt aus dem englischen und steht für „analog-to-digital converter“. Ebenfalls gebräuchliche Bezeichnungen sind Analog-Digital-Umsetzer (**ADU**) oder **A/D-Wandler**. Die wichtigsten Kenngrößen eines ADC sind die Abtastrate, die Auflösung, der Eingangsspannungsbereich und die Latenz.

Die Abtastrate beschreibt die Frequenz, mit der ADC das Eingangssignal abtastet. Damit die Form des Eingangssignals digital sicher abgebildet werden kann, muss die Rate mindestens doppelt so hoch sein wie die höchste im Eingangssignal vorkommende Frequenz (Nyquist-Shannon-Abtasttheorem). Andernfalls entstehen im digitalen Abbild Frequenzen, die im Original nicht vorhanden sind (Aliasing). Um das zu verhindern, wird am ADC-Eingang üblicherweise ein Tiefpassfilter vorgesehen. Letztere bringen einige A/D-Wandler bereits mit, oft muss er aber aus diskreten Bauteilen hinzugefügt werden (bei passiven Filtern üblicherweise ein Kondensator und ein Widerstand, Tiefpass 1. Ordnung).

Die Auflösung gibt an, wie viele digitale Stufen der Wandler darstellen kann. Ein 8-Bit-Wandler sieht das Eingangssignal also in Form von 256 diskreten Spannungswerten. Je höher die Auflösung ist, desto genauer bildet das digitale Signal das Eingangssignal ab. Eingangsspannungen, die zwischen zwei Stufen fallen, werden je nach Umwandlungsverfahren der nächst niedrigeren oder höheren Stufe zugeordnet.

Der Eingangsspannungsbereich gibt an, in welchem Bereich sich die an den ADC-Eingang angelegte Spannung bewegen darf. Sehr viele ADC erlauben einen Bereich von nominell 0 bis 1 Volt. Bei vielen Mikrocontrollern ist der Eingangsbereich per programmierbarem Spannungsteiler einstellbar. Das Ergebnis vergleichen sie mit einer internen Referenzspannungsquelle. Allerdings lohnt sich hier ein genauerer Blick auf das Datenblatt, denn in der Praxis reicht der Digitalisierungsbereich selten bis zu diesen Grenzen. Der ESP32 beispielsweise misst im Spannungsbereich zwischen nominell 0 und 3,3 Volt unterhalb von 0,15 Volt nichts und zeigt ab 3,1 Volt stets den Maximaldigitalwert von 4096 an. Zudem ist die Wandlung nicht zwingend linear, weshalb man in der Regel auf der Softwareseite mit Kalibrierungskurven arbeitet. Eine weitere wichtige Kenngröße ist die Latenz. Sie beschreibt, wie lange ein ADC benötigt, um für ein

anstehendes analoges Signal das digitale Äquivalent zu liefern.

Für die Umwandlung eines analogen Signals in ein digitales sind sehr unterschiedliche Verfahren nutzbar. In der Praxis haben sich je nach Zielsetzung vier Methoden durchgesetzt: Kommt es auf möglichst hohe Abtastraten an, haben sogenannte Pipeline-Umsetzer die Nase vorn. Sie ermöglichen Geschwindigkeiten von 40 Megasamples pro Sekunde (MSPS) bis hin zu 5 GSPS. Übliche Auflösungen sind 8 und 12 Bit (bis 4 GSPS) oder 16 Bit (bis 1 GSPS).

Steht hingegen hohe Genauigkeit bei gemäßigten Abtastraten auf der Wunschliste und spielt Latenz keine große Rolle, kommen Delta-Sigma-Umsetzer zum Einsatz. Übliche Geschwindigkeiten sind hier wenige SPS bis hin zu 2,5 MSPS bei einer Auflösung von 16 bis 24 Bit.

Ist die Latenz wesentlich oder stört das vergleichsweise steile Tiefpassverhalten anderer Wandlungsverfahren, sind sukzessiv approximierende Umsetzer in ihrem Element. Typische Abtastgeschwindigkeiten liegen hier bei 0,1 MSPS bis 10 MSPS.

Vor allem einfache Digitalmessgeräte wie etwa Multimeter setzen auf langsame, Störungen dämpfende Zählverfahren wie Dualslope-Umsetzer.

ADCs in Mikrocontrollern haben oft mehrere Analogeingänge, die sie nacheinander abtasten und in Digitalwerte umwandeln. Die maximal nutzbare Geschwindigkeit des ADC teilt sich so auf die genutzten Kanäle auf. Mitunter wartet ein Microcontroller auch mit mehreren eigenständigen ADCs auf.

Generell arbeiten in Microcontroller integrierte ADCs weniger genau und linear als diskrete ADC-Bausteine. Das

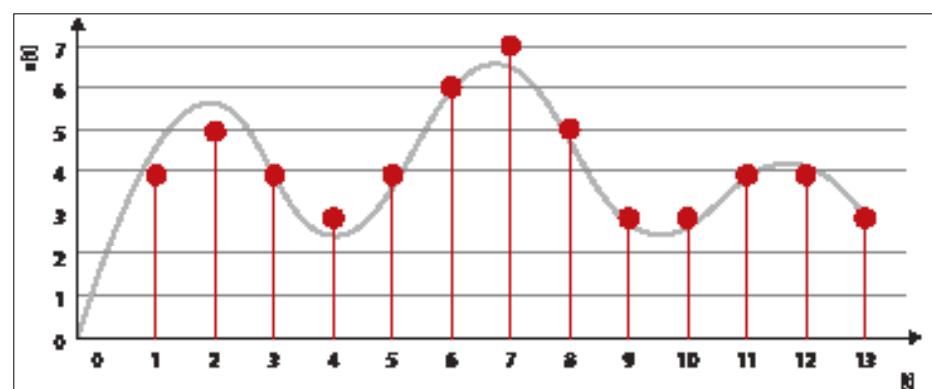
## c't kompakt

- Moderne Mikrocontroller können viel und bieten Dutzende Schnittstellen.
- Wir erklären die wichtigsten Abkürzungen rund um ESP, Raspberry Pi und Co.
- Nebenbei fällt nützliches Grundwissen rund um die Controllertechnik ab.

liegt unter anderem an der hohen Integrationsdichte der Microcontroller. Störungen durch andere Komponenten lassen sich nicht so leicht unterbinden wie bei separaten Bausteinen. Bevor man also zu viel Aufwand in die Beseitigung der systembedingten Macken eines integrierten ADC steckt, lohnt es sich bei höheren Anforderungen, ein externes Bauteil etwa mit SPI-Schnittstelle zu nutzen. Andererseits bieten integrierte ADCs bei mäßigen Anforderungen einen schnellen und oft unkomplizierten Zugang zu einem Wandler.

Die Abkürzung **AHB** (Advanced High-performance Bus) taucht bei Mikrocontrollern in der Regel im Zusammenhang mit ARM-Prozessoren (Advanced RISC Machines Ltd.) auf und beschreibt einen Teil der Advanced Microcontroller Bus Architecture (**AMBA**) moderner ARM-CPUs. Der AHB ist nicht von außen zugänglich und spielt deshalb nur dann eine Rolle, wenn es um das Verständnis der inneren Abläufe in ARM-Chips geht.

**BT** steht in Datenblättern in der Regel für **Bluetooth**. Das ist ein Industriestan-



Ein ADC tastet ein analoges Signal in festen Zeitabständen ab und weist jedem Analogwert einen digitalen Wert innerhalb eines vorgegebenen Rasters zu.

dard für die Datenübertragung zwischen Geräten über kurze Distanz per Funk. Der Name „Bluetooth“ leitet sich vom dänischen König Harald Blauzahn ab (dänisch Harald Blåtand, englisch Harold Bluetooth). Der Funkstandard arbeitet im lizenzzfrei nutzbaren Frequenzband um 2,45 GHz, das zu den sogenannten ISM-Bereichen zählt (ISM für Industrial, Scientific, Medical). War Bluetooth ursprünglich eher für langsame Datenübertragungen im Nahbereich konzipiert, bieten aktuelle Versionen auch die Möglichkeit zu flotterem Transfer über Distanzen bis etwa 200 Meter. Wie gut und wie schnell eine Bluetooth-Verbindung arbeitet, hängt zum einen von der BT-Version, zum anderen aber auch von der Umgebung ab. In der ersten, 1999 festgezurten BT-Version 1.0A/B betrug die maximale Datenrate gerade einmal 732,2 kbit/s bei einer Reichweite von einem (Indoor) bis 10 Meter (Outdoor). Die aktuelle Bluetooth-Version 5.4 (seit Februar 2023 final spezifiziert) ermöglicht bis zu 50 Mbit/s und überbrückt in Gebäuden

bis zu 40 Meter, im Freien sind sogar 200 Meter möglich.

Moderne Mikrocontroller warten mitunter mit Bluetooth LE v5.3 (**BLE** v5.3) auf, wobei das LE für Low Energy steht, was niedrigen Energieverbrauch verspricht, die Reichweite aber auf etwa 10 Meter reduziert. Ältere Mikrocontroller bieten oft BLE v4.2 mit ähnlicher Reichweite und einer maximalen Datenrate von 26 Mbit/s.

Wer eigene Projekte mit Mikrocontrollern und integriertem BT plant, sollte sich frühzeitig nicht nur über die unterstützten BT-Versionen informieren, sondern auch einen Blick auf die von den zugehörigen Programmierbibliotheken unterstützten BT-Profile werfen. Bluetooth ist inzwischen in sehr vielen Bereichen zu einem Quasi-Standard geworden. Ob zwei Komponenten aber miteinander Daten austauschen, hängt auch davon ab, ob beide mit den gleichen Datenübertragungsprofilen umgehen können. Bringt die bevorzugte Programmierumgebung nicht für beide passende BT-Profile mit, kann das erheblichen Mehraufwand bei der Programmierung nach sich ziehen.

Unter einer **CPU** (Central Processing Unit) oder auch Micro Processing Unit (**MPU**) können sich die meisten etwas vorstellen, schließlich sitzt solch ein Prozessor in jedem Notebook, Desktop-Rechner und Server. Eine CPU führt typischerweise ein funktionsreiches Betriebssystem aus, das die Installation und Ausführung vieler verschiedener Programme – mitunter auch gleichzeitig (Multitasking) ermöglicht. Die CPU verarbeitet Tausende laufende Prozesse und Threads sehr schnell, aber es kommt selten darauf an, dass diese auf wenige Millisekunden genau vorhersagbar ablaufen.

Im Gegensatz dazu steuert ein **Mikrocontroller** (**μC**) meistens nur wenige immer gleiche Funktionen, verarbeitet also eher eine Art Firmware. Als Herz einer Steuerung muss ein μC in vorhersagbarer Zeit – quasi in Echtzeit (Real Time) – Berechnungen erledigen. Das Steuergerät eines Verbrennungsmotors etwa muss Messwerte von Kurbelwellensensor, Luftmassenmesser und Gaspedal innerhalb von Millisekunden zu einem Signal für die Einspritzdüsen verarbeiten.

**Cores** oder auch die Prozessorkerne bilden den zentralen Teil eines Mikroprozessors. In aktuellen CPUs wie etwa dem Ryzen 9 von AMD stecken bis zu 16 Cores. Die einzelnen Kerne können auf besondere Aufgaben spezialisiert sein, etwa auf

höchste Rechenleistung (Performance-Core, P-Core) oder möglichst geringen Energieverbrauch (Efficient-Core, E-Core).

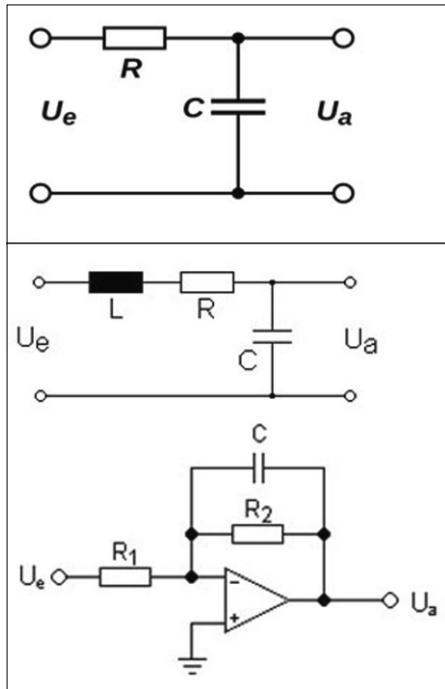
Viele Mikrocontroller haben demgegenüber nur einen Prozessorkern. In der Regel reicht das auch völlig aus, da ein μC meist nur ein sehr eingeschränktes Aufgabenspektrum zu erledigen hat. Doch dieser Unterschied weicht so langsam auf: Moderne, als SoC (System-on-Chip, siehe unten) aufgebaute Mikrocontroller warten inzwischen wie eine CPU mit mehreren Cores auf.

## Noch ein Wandler

Soll ein digitales Signal in ein analoges verwandelt werden, kommt ein digital analog converter (**DAC**) zum Einsatz. Die deutschsprachige Abkürzung **DAU** – Digital/Analog-Umsetzer – findet man seltener in Datenblättern. Im einfachsten Fall besteht ein DAC aus einem Spannungsteiler mit so vielen gleichgroßen Widerständen, wie es Wandlungsstufen gibt. Hinzu kommt noch ein Multiplexer. Dieses diskrete Wandlungsverfahren ist schnell und garantiert monoton, wird aber mit zunehmender Auflösung sehr aufwendig. Für einen 8-Bit-DAC braucht man immerhin 256 Widerstände und 272 Schalter.

Weniger aufwendig ist das sogenannte parallele Umsetzungsverfahren. Hier nutzt man zumeist ein sogenanntes R2R-Netzwerk, an dem eine Referenzspannung anliegt. In dieser Kette von Widerständen und Schaltern (einer pro Bit) ergibt sich eine dem Digitalwert proportionale Ausgangsspannung; ein nachgeschalteter Verstärker puffert sie, damit niederohmige Senken das Signal nicht verfälschen. Parallelumsetzer sind weniger aufwendig als diskrete Wandler, arbeiten aber ähnlich schnell und linear wie diese.

Kommt es bei der Digital-Analog-Umsetzung nicht so sehr auf die Geschwindigkeit an, bietet sich das Zählverfahren an. Hier wird quasi eine Pulsdauermodulation verwendet, die so viele Abstufungen hat, wie es Wandlungsstufen gibt. Bei einem 8-Bit-DAC wären das also 256 Stufen. Das Ausgangssignal wird über einen Tiefpassfilter geführt, der einen Mittelwert der Spannung ausgibt. Dieser garantiert monoton arbeitende DAU lässt sich gut als integrierte Schaltung realisieren und ist bei Mikroprozessoren verbreitet. Werden höhere Umsetzungsgeschwindigkeiten benötigt, kommt oft das Delta-Sigma-Verfahren zum Einsatz, das aber höheren Rechenaufwand verursacht.



**Um Fehler durch zu hohe Frequenzanteile beim analogen Eingangssignal zu vermeiden, setzt man Tiefpassfilter ein. Sie schneiden hohe Frequenzanteile ab. Je nach gewünschtem Filterverhalten sind das RC- (passiver Tiefpass 1. Ordnung) oder LRC-Kombinationen (passiver Tiefpass 2. Ordnung). Aufwendiger sind aktive Tiefpassfilter mit Operationsverstärkern.**

**Flash** oder besser Flash-EEPROM (Electrically Erasable Programmable Read-Only Memory) ist ein nichtflüchtiger, aber trotz des Read-Only blockweise wiederbeschreibbarer Speicher. Die dort hinterlegten Daten bleiben beim Abschalten der Versorgungsspannung erhalten. Typischerweise werden Flash-Speicher bei Mikrocontrollern als Programmspeicher genutzt. Um den Speicherinhalt zu ändern, muss stets ein kompletter Block zunächst gelöscht und anschließend neu beschrieben werden. Dadurch sind Flash-Speicher langsamer als flüchtige Speicher (RAM). Zudem kann ein Flash-Speicherblock nur eine bestimmte Anzahl mal neu beschrieben werden.

Da ein Mikrocontroller meist sehr viele Schnittstellen mitbringt, steht längst nicht für jede Funktion ein eigener externer Kontakt zur Verfügung. Hier kommt General Purpose Input/Output (**GPIO**) ins Spiel. Je nach Programmierung dient ein GPIO-Kontakt als digitaler oder analoger Ein- oder Ausgang oder auch als Teil einer Schnittstelle, die mehrere Kontakte benötigt.

Ein **Hall-Sensor** erfasst Magnetfelder und ermittelt deren Stärke. In der einfachsten Form arbeitet er dabei mit einem meist programmierbaren Schwellwert. Liegt die Feldstärke darüber, liefert der Sensor einen Impuls, liegt die Feldstärke darunter, bleibt der Sensor stumm. Es gibt aber auch Hall-Sensoren, die eine zur ma-

**Groß und mächtig neben klein und fein:**  
**Eine CPU kann eine Vielzahl von Aufgaben mit Hilfe eines oft sehr umfangreichen Befehlssatzes erledigen. Im Gegensatz dazu erledigt ein Microcontroller in der Regel die immer gleiche, oft sehr spezialisierte Aufgabe.**



gnetischen Feldstärke proportionale Spannung ausgeben.

Geht es darum, Daten möglichst flott aus einem Mikrocontroller heraus zu bekommen, wird das High Speed Serial Transmit (**HSTX**) Interface interessant. Über diese Schnittstelle kann ein Mikrocontroller bis zu acht Bit breite Daten seriell – also nacheinander – ausgeben. Die Taktrate für HSTX beträgt maximal 150 MHz und lässt sich unabhängig vom Mikrocontroller-Takt einstellen. Da HSTX sowohl bei der steigenden als auch bei der fallenden Taktflanke Daten überträgt (DDR, Double Data Rate), sind Datenraten von bis zu 300 Mbit/s pro Pin oder kombiniert über alle acht unterstützenden Pins 2400 Mbit/s realisierbar. Aber Achtung: HSTX ist ein unidirektonaler Bus. Das schnelle Einlesen von

Daten in den Mikrocontroller ist darüber nicht möglich.

**I2C** (Inter-Integrated Circuit) wird in der Regel zur langsamten Kommunikation zwischen einem Prozessor und Peripheriekomponenten wie etwa Sensoren oder kleineren Displays auf einer Platine genutzt. Die Kurzstreckenverbindung nutzt zwei Leitungen (SDA, Datenleitung und SCL, Takteitung) für die Kommunikation. Einen Datentransfer stößt stets der I2C-Master an. Er beginnt immer mit dem Senden einer Startbedingung, gefolgt von der Adresse des Slave-Geräts. Sobald der Slave seine Adresse erkennt, nimmt er das nächste Bit als Lese-/Schreib-Flag. Sobald der Slave dem Master seine Aufmerksamkeit gemeldet hat, fließen die Daten. Zum Schluss sendet der Master eine Stop-Bedingung und kann die Kommunikation mit



## CodeMeter – Ein endloser Erfolgskreislauf für Ihr Unternehmenswachstum

### SCHÜTZEN SIE IHRE SOFTWARE

mit besten Verschlüsselungs- und Obfuscations-Technologien

### ERFÜLLEN SIE WÜNSCHE

Ihrer Kunden mit vielseitiger und skalierbarer Lizenzierung

### PROFITIEREN SIE

von Ihrer Arbeit auf globaler Ebene wieder und wieder



Treffen Sie uns!



**SPS**

Halle 6  
Stand 428



**formnext**

Halle 12  
Stand B01C

+49 721 931720  
[sales@wibu.com](mailto:sales@wibu.com)

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



**SECURITY  
LICENSING**

**PERFECTION IN PROTECTION**

anderen Slave-Geräten beginnen. Ein Slave kann von sich aus keine Kommunikation anstoßen. Der Master muss also alle angeschlossenen Slaves regelmäßig abfragen.

I2C arbeitet mit einem eingebauten Prüfsystem: Jedes Datenbyte wird vom Empfänger entweder mit ACK oder NAK bestätigt, um dem Sender zu signalisieren, ob die Daten empfangen wurden oder nicht. Der Master bestimmt bei einem NAK, ob die Daten erneut übertragen oder ob die Kommunikation abgebrochen wird.

Je nach Adressmode (7 oder 10 Bit) können zwischen 128 und 1024 Slaves an einem I2C-Bus hängen. Die Taktrate kann je nach vereinbartem Übertragungsmodus zwischen 100 kHz und 5 MHz betragen. Slaves können die Datenübertragung über Clock Stretching bremsen. Wenn erforderlich, kann es auf einem I2C-Bus auch mehrere Master (Multi-Master-Protokoll)

geben. Diese Betriebsart wird aber nur von sehr wenigen I2C-Programmierbibliotheken unterstützt.

### Audio-Spezialist

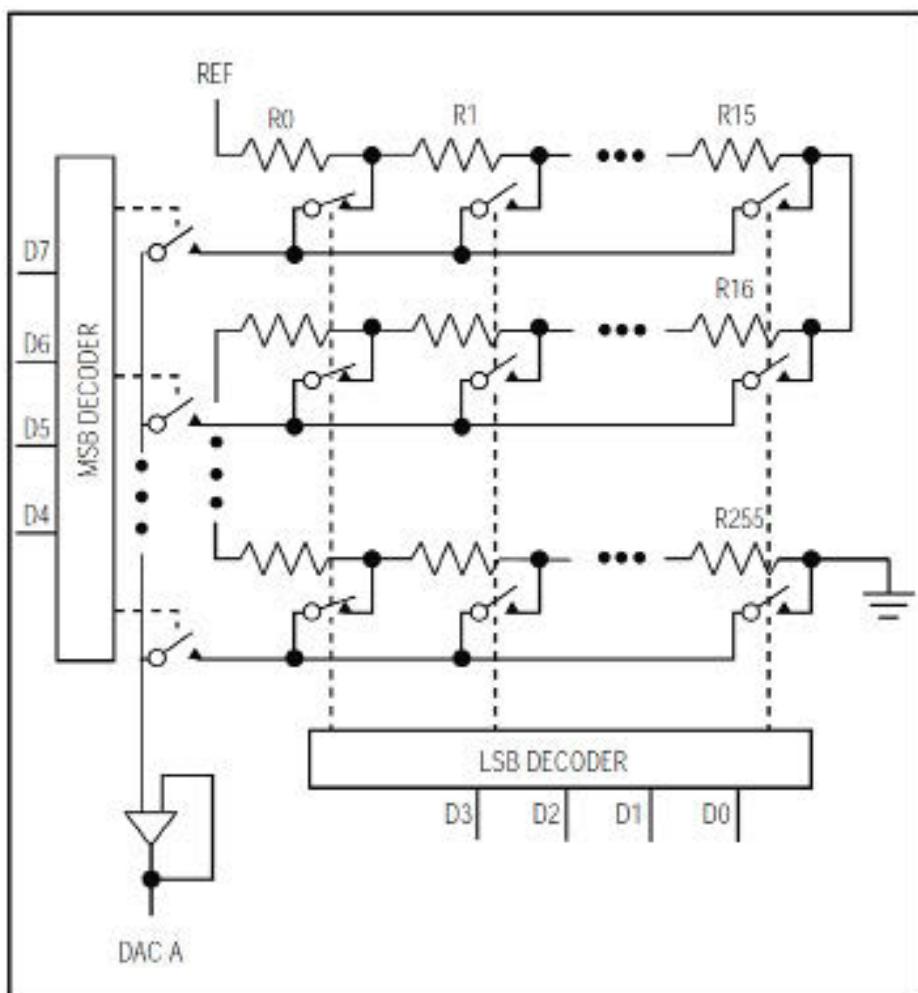
Speziell für den Transport von Audiodaten innerhalb einer Baugruppe wurde **I2S** (Integrated Circuit Sound) entwickelt. Diese Verbindung nutzt mindestens drei Leitungen (Serial Clock, SCK; Word Select, WS und Serial Data, SD), in vielen Schaltungen wird zusätzlich noch „Master Clock“ verwendet. Das Signal lässt sich bei einigen Bausteinen aber auch intern generieren, ein Blick ins Datenblatt hilft weiter. Als unidirektonaler Master-Slave-Bus wurde I2S zunächst zur Übertragung von Puls-Code-Modulierten Signalen (PCM) entwickelt. Die WS-Leitung signalisierte hier, ob die gerade übertragene Bitfolge dem linken oder dem rechten Stereokanal zugeordnet ist. Inzwischen gibt es viele Audio-ICs (DSPs, Audio DACs, Audio

ADCs), die auch andere Datenformate wie etwa Dolby E verarbeiten können. Manche Bausteine ermöglichen auch die Weiterleitung verarbeiteter Signale an weitere I2S-Chips.

Mit **JTAG** wird eine Testschnittstelle für integrierte Schaltungen bezeichnet. Chips, die das ursprünglich von der **Joint Test Action Group** spezifizierte Verfahren beherrschen, haben einen Test Access Point (TAP), der aus mindestens vier Leitungen besteht. Dabei liefert Test Clock (TCK) die Taktrate, Test Mode Select (TMS) bestimmt den durchzuführenden Testzyklus, Test Data Input (TDI) dient als serieller Eingang und über Test Data Output (TDO) lassen sich mehrere Chips mit JTAG-Interface in Reihe schalten. Optional gibt es noch einen Reset-Pin (Test Reset, TRST), der aber auch wegfallen kann. Den Testprozess steuert ein TAP-Controller. Viele Mikrocontroller lassen sich über ihren JTAG-Anschluss auch programmieren. Dazu wiederum verbindet man den Mikrocontroller über einen JTAG-Adapter mit einem PC.

**MIPI** steht für Mobile Industry Processor Interface. Das ist eine Sammlung von Schnittstellen, die innerhalb von Mobilgeräten das Zusammenspiel von Bausteinen unterschiedlicher Hersteller ermöglichen. Die verbreitetsten MIPI-Schnittstellen sind **MIPI CSI** (Camera Serial Interface) und **MIPI DSI** (Display Serial Interface). Beide Schnittstellen sind für Point-to-Point-Verbindungen ausgelegt und nutzen oft Flachbandkabel. Wer Kameras mit MIPI CSI nutzen will, muss darauf achten, dass das Kameramodul und der Mikrocontroller die gleiche Schnittstellenversion unterstützen. Neben der ursprünglichen Variante gibt es inzwischen auch **MIPI CSI-2** für höher auflösende Kameras, **CSI-3** steht bereits in den Startlöchern; Hardware, die diese Version unterstützt, ist uns aber noch nicht untergekommen.

Eine Phasenregelschleife (**PLL**, phase-locked loop) ist ein Regelkreis mit einem gesteuerten Oszillator, dessen Phase der eines äußeren Signals nachgeführt wird. Bei Phasenregelschleifen ist die Abhängigkeit der Stellgröße von der Regelabweichung – der Phasenverschiebung – periodisch. Die Regelung kann also auf verschiedene relative Phasenlagen „einrasten“, die sich um ganze Vielfache von  $2\pi$  (360 °) unterscheiden. Im eingerasteten Zustand ist die Frequenz des Oszillators die des Referenzsignals, mit Teilen auch



Ein Digital-Analog-Converter mit Widerstandsnetzwerk ist schnell und liefert ein gleichbleibend abgestuftes analoges Ausgangssignal, wird aber mit zunehmender Auflösung schnell komplex.

ganzzahlige Vielfache davon. Anwendungen findet die PLL in der Nachrichten-, Regel- und Messtechnik, etwa für Filtern, zur Modulation und Demodulation, in digitalen Kommunikationssystemen zur Taktrückgewinnung und zur Takt-Synchronisation.

Viele Mikrocontroller enthalten einen **PMIC** (Power Management Integrated Circuit), oft auch als **PMU** (Power Management Unit) bezeichnet. Die Einheit übernimmt je nach Chip die unterschiedlichsten Aufgaben. Dazu gehören Spannungswandlung, Batteriespannungsüberwachung, Ladekontrolle, Eingangsspannungswahl und das Festlegen und Überwachen der Einschaltreihenfolge für die verschiedenen, innerhalb des ICs genutzten Spannungspiegel. Kann der Mikrocontroller die Versorgungsspannung intern absenken, etwa um im Idle-Betrieb weniger Energie zu verbrauchen, regelt auch das die PMU.

**PSRAM** (Pseudostatisches RAM) ist eine Speichervariante, die neben dem üb-

lichen flüchtigen DRAM (dynamischer Speicher) auch noch eine Steuerschaltung enthält, die das zyklisch erforderliche Aufladen der DRAM-Elemente übernimmt. Der Baustein verhält sich damit äußerlich wie ein statischer Speicher (SRAM), arbeitet aber intern mit günstigeren DRAM-Zellen. PSRAM für Mikrocontroller hat oft ein SPI (siehe unten), ist also nicht besonders schnell.

### Blink-Dimmer

Pulsweitenmodulation (**PWM**) wird häufig verwendet, um etwa die Helligkeit von LEDs zu regulieren (**LED PWM**), ohne die Versorgungsspannung zu verringern. Bei einem Rechteck-Signal mit konstanter Frequenz variiert dabei die Einschaltdauer. Soll die LED maximal hell leuchten, bleibt das Signal permanent auf eins (100 Prozent), bei 50 Prozent ist es hingegen eine halbe Periode lang. Ist die Signalfrequenz hoch genug (über 100 Hz), nimmt das Auge das Flimmern der LED nicht wahr, sondern registriert stattdessen nur

eine geringere Helligkeit. Viele Mikrocontroller bringen mehrere integrierte LED-Controller (**LEDC**) mit, die die Helligkeitssteuerung von LEDs sehr komfortabel machen.

Manche Mikrocontroller wie etwa der ESP32 haben noch einen oder mehrere Einheiten zur Motorkontrolle (Motor Control Pulse Width Modulator, **MCPWM**). Mit diesen lassen sich Motordrehzahlen ohne allzu großen Aufwand regulieren. Man braucht dazu meistens noch Leistungstransistoren, weil viele Motoren deutlich stärkere Ströme benötigen, als ein Mikrocontroller schalten kann.

**OTP** (One Time Programmable) beschreibt bei Mikrocontrollern einen nicht-flüchtigen Speicher (**PROM**, Programmable ROM), der sich nur einmal beschreiben lässt.

**RISC-V** (Reduced Instruction Set Computers Five) beschreibt eine quelloffene Befehlssatzarchitektur. Jeder kann auf dieser Basis einen eigenen Prozessorkern entwickeln und auch verkaufen. Li-

Brüssel is calling:

**Sie brauchen bald einen digitalen Produktpass!**

Bereiten Sie sich mit uns auf die europäische Ökodesign-Verordnung vor.



zenzgebühren fallen nicht an. Die RISC-V-Technik konkurriert unter anderem mit Rechenkernen von ARM für Embedded-Systeme.

Zufallszahlen spielen in vielen Bereichen der Technik eine wichtige Rolle. Besonders dann, wenn es um sichere Verschlüsselung geht, ist ein zuverlässiger Zufallszahlengenerator (**RNG**, Random Number Generator) nützlich, weshalb manche Mikrocontroller einen solchen RNG gleich mitbringen.

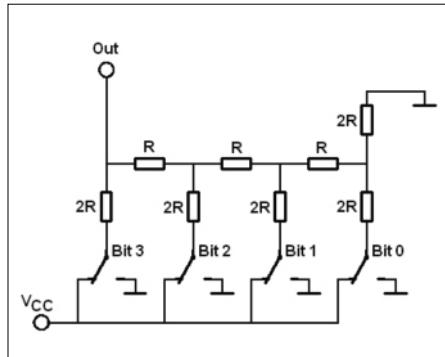
Ein in einen Mikrocontroller integrierter Remote Control Transceiver (**RMT**, ReMoTe Control Transceiver) erleichtert vor allem die Nutzung von Funk- und Infrarot-Fernbedienungen. Er übernimmt die Trägerfrequenzmodulation und hilft bei der Signalkodierung und -dekodierung.

In einem **ROM** (Read Only Memory) kann der Chiphersteller bestimmte Elemente des Maschinencodes fest und unveränderbar hinterlegen. Das passierte in der Vergangenheit oft schon bei der Chipherstellung über Maskenprogrammierung. Allerdings ist dieses Verfahren unflexibel, weshalb heute veränderbare Speicher wie etwa Flash-ROMs oder OTP gängig sind.

Eine Echtzeituhr (**RTC**, Real Time Clock) ermöglicht viele zeitgesteuerte Anwendungen, weshalb sie in kaum einem Mikrocontroller fehlt.

Will ein Mikrocontroller SD-Speicher-karten schreiben oder lesen, muss er als **SDIO Host** (SD Input/Output Host) arbeiten. Soll er sich hingegen gegenüber anderen Geräten wie einer SD-Karte verhalten, ist er ein **SDIO Slave**.

**SPI** (Serial Peripheral Interface) ist eine serielle Master-Slave-Schnittstelle mit einer vom Master bereitgestellten Taktleitung (SCLK) und separaten Eingangs-(MISO, Master Input, Slave Output) und Ausgangsdatenleitungen (MOSI, Master Output, Slave Input). Über eine Slave-Select-Leitung (SS) legt der Master fest, welchen Slave er ansprechen will. Je nach Aufbau des SPI-Masters kann dieser Slaves über mehrere SS-Leitungen (meist SS1 bis SS4) oder in einer kaskadierten Konfiguration über hintereinandergeschaltete Slaves ansprechen (MISO-MOSI-Kette). Letzteres unterstützen allerdings nicht alle SPI-Bibliotheken. Die Durchsatzraten für einzelne SPI liegen zwischen 10 und theoretisch 24 Mbit/s. Welche Datenrate tatsächlich erreicht werden kann, hängt von den verbauten Chips ab. Die meisten Bausteine schaffen 10 Mbit/s.



Bei Parallelumsetzern erzeugt ein Netz aus Widerständen und Schaltern eine dem Digitalwert proportionale Ausgangsspannung. Dahinter folgt ein Pufferverstärker, damit nichtlineare Senken die Wandlung nicht verfälschen.

Sind höhere Datenübertragungsraten erforderlich, kommen **DSPI** (Dual Serial Peripheral Interface) oder **QSPI** (Dual Serial Peripheral Interface) zum Einsatz. Hier gibt es dann keine separaten Input- und Output-Dateneleitungen, sondern pro Kanal eine in beide Richtungen nutzbare Leitung (IO0 bis IO3). Mit DSPI lassen sich Datenraten von etwa 20 Mbit/s erreichen, bei QSPI sind es bis zu 40 Mbit/s.

Statisches RAM (**SRAM**, Static Random-Access Memory) ist ein flüchtiger Speicher, der beim Abschalten der Versorgungsspannung seine Daten verliert. SRAM ist in der Regel schneller, aber auch teurer als DRAM, braucht dafür aber keinen speziellen Mechanismus (Refresh), um die gespeicherten Daten zu erhalten.

Nahezu alle Mikrocontroller und Prozessoren kann man heute als System-on-Chip (**SoC**) bezeichnen, weil sie neben dem eigentlichen Prozessorkern in der Regel weitere Einheiten wie etwa einen Speichercontroller, Powermanagement-Units, Schnittstellen, Flash-Speicher, SRAM und vieles mehr enthalten.

**Thread** ist eine eigens für Smart-Home-Anwendungen entwickelte Funktechnik, um Geräte zu vernetzen. Komponenten, die Thread beherrschen, bauen ein Mesh-Netzwerk auf. Lichtschalter, Thermostate, Steckdosen, Sensoren und so weiter kommunizieren so auch untereinander. Solch ein Thread-Netzwerk ist damit nicht von einem zentralen Knotenpunkt abhängig. Fällt ein Gerät aus, suchen sich die Datenpakete den Weg über ein anderes. Theoretisch wird ein auf Thread aufsetzendes Smart-Home-System also umso stabiler, je mehr Geräte im Netz hängen.

Time-of-Flight (**ToF**) ist eine Methode zur Distanzmessung zwischen einem Sensor und einem Objekt, basierend auf der Zeitdifferenz zwischen der Aussendung eines Signals und seiner Rückkehr zum Sensor, nachdem es von einem Objekt reflektiert wurde. Chips wie der ESP32-S3 bieten Unterstützung für die Entfernungsmessung über ihren WLAN-Port. Vieles ist hier aber aktuell noch experimentell.

Ein **Touch**-Sensor reagiert auf Berührungen und arbeitet bei den meisten Mikroprozessoren zumeist kapazitiv. Mehrere GPIO-Pins können als Touch-Pins festgelegt werden. Problematisch ist mitunter die Adaption der Touch-Flächen am Gehäuse.

Aus dem Automobilbereich stammt das Two-Wire Automotive Interface (**TWAI**). Diese echtzeitfähige serielle Schnittstelle ist multimeterfähig und wird deshalb auch im industriellen Bereich für Steuer- und Regelungsaufgaben eingesetzt.

**UART** steht für Universal Asynchronous Receiver/Transmitter und definiert ein Protokoll beziehungsweise einen Regelsatz für den Austausch von seriellen Daten zwischen zwei Geräten. UART ist sehr einfach und nutzt lediglich drei Drähte zwischen Sender und Empfänger, um zu senden (Transmit, Tx) und zu empfangen (Receive, Rx); die dritte Leitung trägt das gemeinsame Bezugspotenzial (Masse). Die uralte und weit verbreitete serielle Schnittstelle (RS-232, COM-Port) nutzt etwa einen UART.

USB On-The-Go (**USB OTG**) ist eine Spezifikation, die es USB-Geräten erlaubt, mal als USB-Master und mal als USB-Slave zu agieren. So kann etwa ein Smartphone über seine USB-Schnittstelle eine externe Festplatte ansprechen (Master), aber auch angeschlossen an einen Computer als Datenquelle fungieren (Slave).

## So viele Begriffe

War es das nun mit typischen Abkürzungen in der Mikrocontrollerwelt? Leider nein, denn je tiefer man in die verschiedenen Modelle hineinschaut, desto mehr vor allem systemspezifische, mehrbuchstabige Kürzel begegnen einem. Wir haben versucht, die gebräuchlichsten Begriffe zusammenzutragen. Wenn Ihnen eine Akü begegnet, die hier nicht erklärt, aber dennoch von allgemeiner Bedeutung ist, schreiben Sie uns an ct@ct.de.

(gs@ct.de) ct

**Spezifikationen und Normen:** [ct.de/yf67](http://ct.de/yf67)

# 206 Milliarden Gründe, heute noch in die Sicherheit Ihrer IT zu investieren



Laut einer Studie des Branchenverbands Bitkom ist der deutschen Wirtschaft im vergangenen Jahr durch Diebstahl von IT-Ausrüstung und Daten sowie durch Wirtschaftsspionage und Sabotage ein Schaden von 206 Milliarden Euro entstanden, davon allein 148 Milliarden Euro durch Cyber-Angriffe. Die fortschreitende Digitalisierung und zunehmende Vernetzung vergrößern die Angriffsflächen – und diese werden genutzt.

Die Zeiten, in denen Cybersicherheit ein Randthema der IT-Abteilungen war, sind daher längst vorbei. Heute steht sie im Zentrum des strategischen Managements eines jeden Unternehmens – oder sollte es zumindest.

#### **Warum können Unternehmen das Thema nicht ignorieren?**

Die Abhängigkeit von IT-Systemen ist immens und nimmt mit jeder digitalen Innovation zu. Ein Ausfall dieser Systeme durch Cyberangriffe kann zu erheblichen Verlusten führen, die weit über finanzielle Schäden hinausgehen. Reputationsschäden, der Verlust von Kundenvertrauen und rechtliche Konsequenzen sind nur einige der schwerwiegenden Folgen.

Business Security ist im Übrigen nicht nur eine Frage der Risikominimierung, sondern eine Grundvoraussetzung für den Geschäftserfolg im 21. Jahrhundert. Kunden, Partner und Behörden erwarten einen verantwortungsvollen Umgang mit Daten und eine sichere IT-Infrastruktur. Regulierungen wie die DSGVO in Europa erhöhen den Druck auf Unternehmen, ihre Sicherheitsstandards zu verbessern.

#### **Partners in Crime:**

#### **M-net hilft Ihnen, Ihre optimale Sicherheitsstrategie zu finden**

Sie sind nicht allein! Unkompliziert, sicher und maßgeschneidert unterstützen wir Sie partnerschaftlich im Kampf gegen Cyberbedrohungen:

#### **DDoS-Schutz**

Distributed Denial of Service (DDoS)-Angriffe überlasten Server oder Netzwerke und legen Dienste lahm – mit oft katastrophalen Folgen. Um Ihr Unternehmen vor DDoS-Attacken zu schützen, müssen diese erkannt, gefiltert und blockiert werden. Gleichzeitig sollen reguläre Nutzeranfragen jedoch ungehindert weiterfließen können. M-net bietet effektive DDoS-Schutzpakete ganz nach den Bedürfnissen Ihres Unternehmens.

#### **Sichere Business-Telefonie**

Die zunehmende Verbreitung von VoIP-Lösungen stellt Unternehmen vor neue Herausforderungen. Als KRITIS-Unternehmen bietet M-net seinen Kunden mit einem BSI-konformen SIP-Trunk Sicherheit auf höchstem Niveau. M-net unterstützt Sie bei der Zertifizierung Ihrer Telekommunikationslösung nach BSI-Konformität und entwickelt gemeinsam mit Ihnen eine optimale Sicherheitsstrategie für Ihre VoIP-Kommunikation.

#### **Maximale Ausfallsicherheit**

Ausfälle des Internets führen im digitalen Zeitalter schnell zum kompletten Stillstand und zu finanziellen Verlusten. Durch eine Kombination aus maßgeschneiderten Backup-Verbindungen und der Implementierung redundanter Zugangs- und Systemlösungen bleibt Ihr Unternehmen auch in Krisenzeiten funktionsfähig und gut geschützt.

Mehr zu den M-net Business Security-Lösungen erfahren Sie unter [www.m-net.de/business-security](http://www.m-net.de/business-security)

Holen Sie sich auch das kostenlose Whitepaper „7 Strategien für eine proaktive Business Security“:

[www.m-net.de/whitepaper-security](http://www.m-net.de/whitepaper-security)

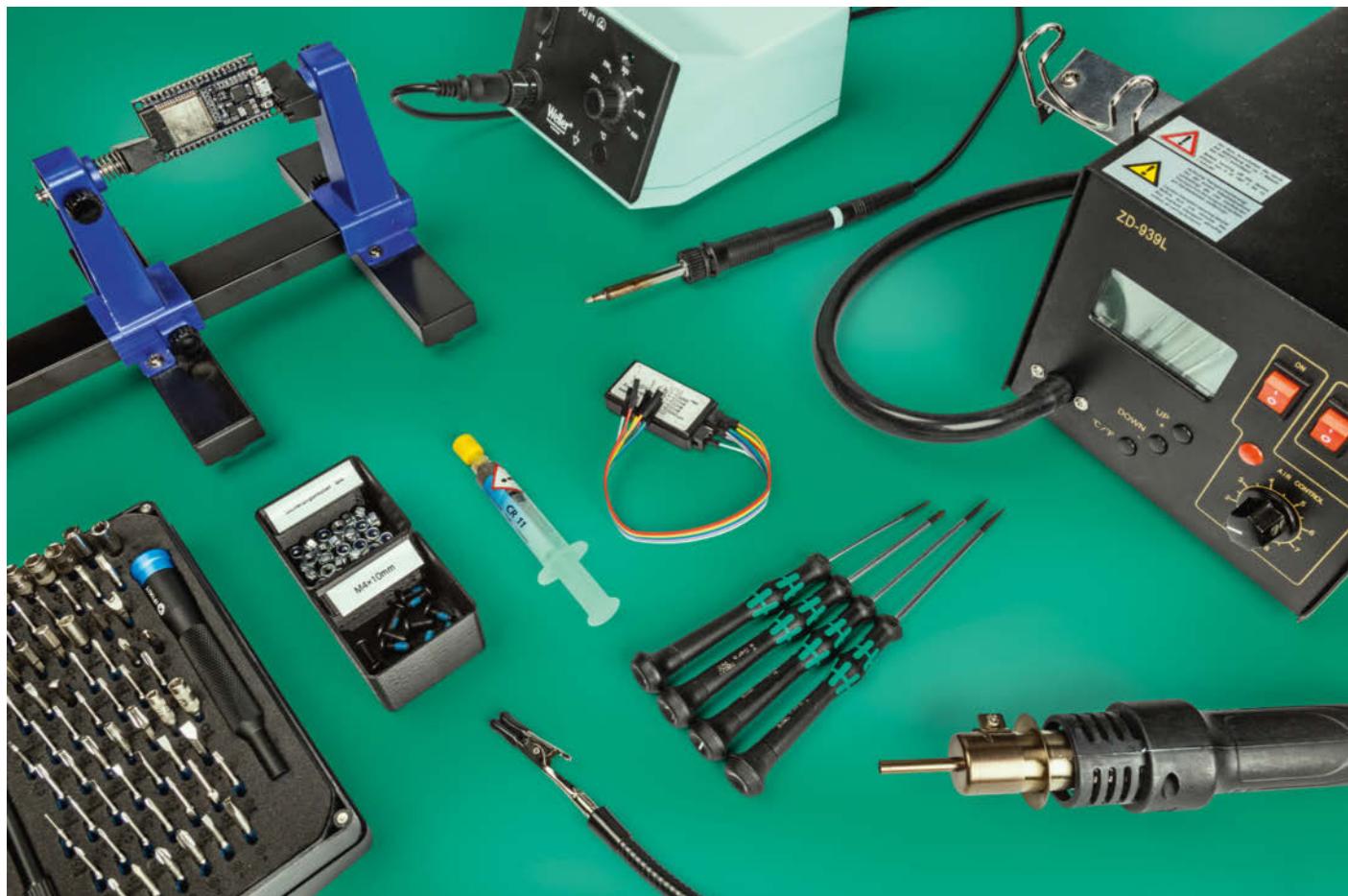
Sind Sie dabei? Kontaktieren Sie uns für eine unverbindliche Beratung.



M-net Telekommunikations GmbH  
Frankfurter Ring 158 | 80807 München  
Tel.: 0800 723 98 48

M-net ist klimaneutral

\*Bereits erreicht und noch viel vor: [m-net.de/klimaneutral](http://m-net.de/klimaneutral)



# Schluss mit Pfusch

## Die fast perfekte Bastel- und Reparaturwerkstatt

**Basteln mit Elektronik macht Spaß, ist lehrreich und die Hardware kostet heute kein Vermögen, sodass fast jeder loslegen kann. Doch ganz ohne Werkzeug und Ausstattung geht es nicht – und das kann ins Geld gehen, wenn man gleich die Vollausstattung bestellt. Ein paar Investitionen müssen sein, aber an vielen Stellen kann man Geld sparen.**

Von Jan Mahn

**D**as Bastelfieber ist ausgebrochen. Experten sind sich einig: Da hilft keine Medizin, da hilft nur Basteln. Doch solange dem Bastelfieber die medizinische Anerkennung als Krankheit verwehrt bleibt, weigern sich selbst private Krankenkassen, die Kosten für dringend benötigte Hilfsmittel zu bezahlen. Wenn Sie nicht gerade die luxuriöse Situation vorfinden, dass Ihr Arbeitgeber, Ihre Schule, eine Universität oder ein gut ausgestatteter Maker-Space Ihnen eine Elektronik- und Bastelwerkstatt bereitstellt, müssen Sie wohl oder übel selbst in die Tasche greifen. Damit dieser Griff nicht unnötig tief ausfällt, haben wir diesen Ratgeber zusammengestellt. Darin tragen wir Werkzeuge zusammen, die sich in der Redaktion bewährt haben und geben Tipps, in

welcher Reihenfolge sich die Anschaffung lohnt.

### Ohne geht nichts

Ohne vernünftige Schraubendreher ist eine Werkstatt keine und eine gut überlegte Investition kann eine ganze Bastlerkarriere halten. Der ärgste Feind von gutem Werkzeug ist ohnehin nicht der Verschleiß, sondern der Verlust, vor allem verursacht durch unvorsichtigen Verleih. Tipp: Die Investition in eine Wandhalterung oder eine Tasche lohnt sich. So hat jeder Schraubendreher einen festen Platz und es geht weniger verloren.

Kauft man seine Schraubendreher bei einem der Markenhersteller, bekommt man auch nach Jahren noch einen zum System passenden Ersatz – bei den 10-Euro-

Sets, die im Baumarkt an der Kasse liegen, ist das nahezu ausgeschlossen. Die Gattung der Schraubendreher teilt sich in zwei Untergattungen, der Übergang zwischen beiden ist fließend: Präzisionsschraubendreher (auch Uhrmacher- oder Feinmechanikerschraubendreher) haben einen dünnen Griff, den man gut zwischen zwei Fingern drehen kann – ideal für gefühlvolle Arbeiten und kleine Antriebe. Als Antrieb bezeichnet die Fachsprache die Passformen im Schraubenkopf. Daneben gibt es die Schraubendreher mit einem großen Griff.

Schlitz ist der einfachste, älteste und gleichzeitig nervigste Antrieb. Der Schlitz ist so alt, dass seine deutsche Norm noch eine zweistellige Nummer hat: DIN 84. Ein Schlitz zentriert aber nicht, weshalb der Schraubendreher ständig verrutscht. Ein paar Schlitzschraubendreher gehören dennoch in jedes Sortiment. Auch wenn es verlockend ist: Sie sind zum Schrauben gemacht, nicht zum Hebeln oder Meißeln.

Dann gehören kleine Dreher mit Kreuzantrieb ins Sortiment, die in Fachkreisen Phillips heißen (benannt nach einer US-Werkzeugfirma, nicht nach dem Elektronikkonzern Philips), abgekürzt PH. In Beschreibungstexten steht hinter PH eine Nummer, die die Größe angibt. PH00 ist der kleinste, den man regelmäßig braucht, dann folgen PH0, PH1, PH2 und PH3 (der ist schon 5 mm breit).

Bei Schrauben, die mit mehr Drehmoment versenkt werden, zum Beispiel Spanplattenschrauben im Holzbau, ist ein anderer Antrieb mehr verbreitet, der auf den ersten Blick auch wie ein Kreuz aussieht: Pozidriv (PZ). Der hat zusätzlich zum großen Kreuz noch ein um 45 Grad gedrehtes kleineres Kreuz – das Werkzeug hat dazu passend ebenfalls ein weiteres nach hinten versetztes Kreuz. Ein Pozidriv-Dreher in einem Pozidriv-Kopf rutscht seltener ab und kann mehr Kraft übertragen. Pozidriv-Schrauben kann man auch mit Phillips-Werkzeug drehen, jedoch mit weniger Drehmoment. Andersherum rutscht der Dreher aber ständig ab und ruiniert den Kopf, weil das zweite Kreuz im Weg ist. Wenn Sie sich mal über eine vermeintlich ausgenudelte Schraube ärgern: Vielleicht passt das Werkzeug einfach nicht zur Schraube und weitere Versuche mit diesem verbessern die Lage nicht.

In den vergangenen Jahrzehnten hat eine weitere Form den Schlitten und Kreuzen den Rang abgelaufen – und zwar sowohl im Holzbau als auch bei Elektrogeräten: Der Torx, ein Stern mit sechs Spitzen

und eine geschützte Marke. Firmen, die diese Markenrechte nicht gekauft haben, schreiben TX. Weil Torx so verbreitet ist, lohnt sich ein Feinmechanik-Satz davon in jedem Fall. Ein Satz Innensechskantschlüssel (unter der Markenbezeichnung Inbus bekannt) rundet die Grundausrüstung ab.

Abeis der gängigen Antriebe tummeln sich leider viele Exoten, besonders in Heimelektronik, wo sie Reparateure abschrecken sollen. Für diese Varianten braucht man nicht immer eigene Schraubendreher, ein Set aus Bithalter und exotischen Bits reicht aus. Zu den häufigsten Ärgernissen bei der Reparatur von Geräten gehören Schrauben vom Typ Torx-TR. Die haben außen die Stern-Form von Torx, in der Mitte aber einen runden Pin, sodass normale Torx-Schraubendreher nicht passen. Das TR steht für „tamper resistant“, also manipulations sicher. Die fragwürdige Sicherheit besteht darin, dass man schlicht bereit sein muss, passendes Werkzeug zu kaufen.

Verschiedene Größen von Torx-TR braucht man unbedingt in der Sammlung. Wer auch mal Geräte von Apple aufschrauben will, muss zusätzlich Werkzeug vom Typ Pentalobe haben, ein etwas abgerundeter Stern mit nur fünf Spitzen. Weitere gängige Schikanen: Spanner (zwei Löcher nebeneinander) und (sehr selten) ein Dreieck. Eine recht vollständige Bitsammlung für typische Reparaturen hat die Firma iFixit für rund 40 Euro im Sortiment. Inspiriert ist die Auswahl durch das, was sich in realen Notebooks, Tablets und Handys so an Schrauben ansammelt.

## Ordnungssysteme

Wenn Sie bisher dachten, den sogenannten Vendor-Lock-In hätte allein Apple perfektioniert, um die Kunden im eigenen Ökosystem gefangen zu halten, dann irren Sie. Schauen Sie sich einmal im Vorratsraum, in der Garage oder in der Küche um:



**Druckt man seine Sortierschachteln selbst, ist der Nachschub gesichert. Die Druckvorlagen folgen einem Rastersystem, man kann also auch längere, breitere und höhere Schachteln anfertigen.**

## c't kompakt

- Wer sich eine eigene Werkstatt ausstattet, muss sich entscheiden. Die Erfahrung lehrt: Gutes Werkzeug hält lange und erhöht den Spaß bei der Arbeit.
- Eine Lötstation statt eines Lötkolbens bedeutet nicht nur Komfort, mit Temperaturregelung ist das Löten auch weniger ungesund.
- Wer Kleinteile sortieren will, kann die Schachteln auch selbst drucken. Das schützt davor, eines Tages keine passenden Gefäße mehr nachkaufen zu können.

Wie viele verschiedene, nicht kompatible Kisten- und Schachtelsysteme zählen Sie?

Unsere Tipps: Für alle größeren Teile sind die Eurobehälter (auch Euronormbehälter genannt) unschlagbar. Schlichte Kunststoffkisten mit standardisierten Größen, die ähnlich wie das DIN-A-System bei Papier funktionieren. Die großen Kisten haben Außenmaße von 40 × 30 mm. Auf sie passen zwei Stück der nächstkleineren Sorte nebeneinander. Acht der großen Kisten passen wiederum auf eine Europalette, davon wiederum drei nebeneinander in einen Eurocontainer – höchst befriedigend, wenn man nicht gern Platz verschenkt. Untereinander kompatible Eurobehälter in allen Größen gibt es von verschiedenen (No-Name-)Herstellern und mittlerweile in jedem Baumarkt, das Ökosystem ist also sehr offen und Nachschub auch in Jahrzehnten noch sehr wahrscheinlich.

Für Kleinteile wie Schrauben, Muttern und Elektronikteile, aber auch USB-Adapter und -Kabel ist das Euroboxsystem aber zu groß. Eine hervorragende Ergänzung können 3D-gedruckte Schachteln sein, die einem Raster folgen – und zwar in Länge, Breite und Höhe. Groß gemacht hat die Idee der Bastel-YouTuber Alexandre Chappel, der Druckvorlagen für ein solches Rastersystem auf der Teileplattform Thingiverse hochgeladen hat. Die Boxen können quadratisch sein oder schmal und lang. Anstatt die mitgelieferten Größen eines gekauften Sortiments irgendwie zu füllen, schauen Sie erst mal, welche Teile Sie herumliegen haben und geben die passenden Schachteln auf dem 3D-Drucker in Auftrag.

Klar, das Gedrucke kostet Zeit, aber erfahrungsgemäß hat man pro Tag auch nur begrenzt Lust, Schrauben und Elektronikbauteile zu sortieren. Gleichzeitig ist das ein Projekt, um endlich altes Filament oder Reste aufzubrauchen und den 3D-Drucker auszulasten. Auf Seite 33 sehen Sie Schachteln, die vom Chappel-Raster inspiriert sind, aber minimal andere Maße haben, damit sie in Werkzeugkisten vom Typ Systainer passen. Oben haben sie jeweils eine angeschrägte Fläche für Beschriftungsbänder. Alle Druckvorlagen finden Sie über ct.de/y8kk.

## Löten

Keine Elektronikwerkstatt ohne Ausstattung zum Löten. Spätestens wenn der Raspi- oder ESP-Aufbau mit Sensoren, LEDs und anderen Bauteilen auf dem Steckbrett (Breadboard) halbwegs funktionsfähig erscheint, ist es Zeit, den Löt Kolben aufzuheizen und den Aufbau zum Beispiel mit einer Lochrasterplatine in einen robusteren Prototyp zu verwandeln.

Die wichtigste Frage lautet dann: Löt Kolben oder Lötstation? Die billigsten Löt Kolben bekommt man schon ab 10 Euro und viel Magie steckt in solchen Kolben auch nicht. Man steckt sie in die Steckdose und sie erzeugen mithilfe der Netzspannung Wärme, ähnlich wie ein Tauchsieder. Typischerweise wird ihre Leistung mit 60 oder 80 Watt angegeben. Für absolute Gelegenheitslöter, die wirklich nur mal fix mit einer mittelmäßig breiten Spitz e eine

LED auf eine Platine löten wollen, reicht ein solcher Kolben aus. Die Probleme der Technik: Auf der einen Seite ist das Kabel meist recht dick und starr, was bei filigranen Arbeiten wirklich nervt, weil das Kabel immer etwas zu sehr am Kolben zieht. Onlineangeboten sieht man leider nicht an, wie flexibel das Kabel ist, doch es gibt durchaus Unterschiede.

Das größte Problem der schlichten Technik ist aber die fehlende Temperaturregelung. Irgendwie bekommt man mit den 60 Watt Dauerleistung das Lötzinn, das je nach Typ (dazu später mehr) bei etwa 200 Grad schmilzt, in der Regel flüssig. Aber eben nur irgendwie und nicht immer elegant.

Ein harter Gegner für schlichte Löt Kolben sind solche Werkstücke, die viel Hitze abführen können, weil sie etwa große Kontaktflächen haben. Versucht man an solchen zu löten, sinkt die Temperatur an der Kolbenspitze schnell ab. Das Zinn wird nicht richtig flüssig, der Bediener wird unruhig und stochert hektisch herum. Am Ende hat man länger als nötig die Platine mit Hitze belastet und schlimmstenfalls noch andere Teile beschädigt, nur weil der Kolben nicht lange genug heiß genug war. Man könnte jetzt der Idee verfallen, einfach gleich einen leistungsstärkeren Löt Kolben zu kaufen, der von Anfang an eine höhere Temperatur hat. Gut gedacht, aber auch nicht ideal: Zu hohe Temperaturen beschädigen direkt Bauteile und Platinen, außerdem verbrennen sie das Flussmittel, das im Inneren des Lötdrahts steckt.

Deshalb kommen temperaturgesteuerte Lötstationen ins Spiel. Sinkt die Temperatur an der Spitze ab, regelt die Station dagegen und erhöht die Heizleistung, um weiter die eingestellte Temperatur zu halten. Netzteil und Steuerung stecken in der Station, die auf dem Tisch steht, sodass der Kolben selbst viel leichter ist, wodurch filigranes Löten einfacher von der Hand geht. Das Kabel zur Station ist dünner, es handelt sich meist um ein flexibles Silikonkabel.

Braucht man eine Lötstation? Nicht unbedingt: Ein guter und erfahrener Löter wird auch mit einem 10-Euro-Kolben gute Verbindungen hinbekommen, aber hin und wieder leise fluchen. Und eine teure Station macht aus einem Anfänger noch keinen guten Löter. Die Lernphase macht aber unbestritten mehr Spaß.

Unser Tipp: Eine hochwertige Lötstation wie die von Weller, die Sie auf dieser Seite unten links sehen, kostet über 300 Euro. Weil die Geräte aber bekannt dafür sind, dass sie Jahrzehnte problemlos funktionieren, schauen Sie sich einmal in Kleinanzeigen um. Der Markt für gebrauchte Stationen ist recht groß. Egal ob Station oder Kolben: Sie brauchen in jedem Fall eine sehr feine punktförmige Spitz e für präzise Arbeiten und eine flache, um mehr Hitze auf große Lötstellen übertragen zu können.

## Schutz vor Schmutz

Ebenso wichtig wie das Lötgerät selbst ist der Schutz des Anwenders: Denn auch wenn man sprichwörtlich von Fehlern nicht dümmer wird, gilt das ausdrücklich nicht für den Fehler, beim Löten die Dämpfe einzutreten! Lot, das im privaten Bereich eingesetzt wird, enthält meistens neben Zinn auch Blei (zu erkennen an den Buchstaben Pb im Namen), weil das eine schön niedrige Schmelztemperatur hat. Händewaschen nach der Arbeit ist schon mal Pflicht. Und Bleidämpfe haben nachweislich negativen Einfluss auf das Gehirn, Lötdämpfe machen also langfristig dumm. Selbst wenn man bleifreies Lot verwendet, wie es bis auf wenige Ausnahmen in der Industrie heute Pflicht ist, bleiben genug andere giftige Substanzen (Formaldehyd, Phenol, Kohlenmonoxid) im Rauch übrig, die nicht in Nase und Lunge gehören. Das Berufsgenossenschaftliche Institut für Arbeitsschutz hat schon 2004 eine Studie zur Lötrauchemission durchgeführt und einen lesenswerten Bericht veröffentlicht (siehe ct.de/



**Harter Angriff auf den Geldbeutel: Eine Lötstation aus dem Hause Weller ersteht man am besten gebraucht, dann ist der Preis erträglich. Rechts daneben eine simple Lötrauchabsaugung, die im gewerblichen Bereich nicht ausreicht, zu Hause aber die Lunge schont.**

y8kk). Eine Erkenntnis: Begrenzt man die Temperatur (etwa auf 350 Grad), verringert das die Menge an Rauch signifikant – die Gesundheit ist ein weiteres Argument für eine Lötstation.

Untersucht wurde außerdem die Wirksamkeit von Schutzmaßnahmen. Auf Seite 34 neben der Lötstation sehen Sie das Einstiegsmodell unter den Lötrauchfiltern: Ein Tischgerät mit Ventilator und Aktivkohlefilter, für das Sie um 50 Euro bezahlen müssen. Die Berufsgenossenschaften kommen in ihren Informations-schriften (zu finden über ct.de/y8kk) auf Grundlage der Studie zu dem Schluss, dass ein solcher Tischfilter nicht ausreicht, wenn man als Arbeitnehmer jeden Tag lötet. In solchen Umgebungen muss es eine weitaus teurere Absaugung mit einem flexiblen Schlauch und Trichter sein, den man direkt ans Werkstück führt. Für den Hobbybereich ist ein einfacher Lötrauch-filter auf dem Tisch aber definitiv besser als gar kein Schutz und die Anschaffung ist dringend empfohlen – eine sinnvolle Investition auch und gerade wenn man mit Kindern löten übt.

## Heiße Luft

Zu den vielseitigsten Geräten gehört die Heißluft-Lötstation wie jene mit dem Namen ZD-939L, die Sie auf Seite 32 rechts im Bild sehen. Eine solche gibt es ab 55 Euro und kaum ein Gerät für diesen Preis erweitert so sehr das Spektrum der Arbeiten, die man mit etwas Übung selbst durchführen kann. Primäre Aufgabe sind Löt- und Entlötarbeiten an sogenannten oberflächenmontierbaren Bauteilen, auf Englisch „Surface Mountable Devices“ (SMD). Darunter fallen alle Bauteile auf Platinen, die nicht in Durchsteckmontage auf einer Lochrasterplatine installiert sind. In der industriellen Fertigung wird zuerst über eine Maske eine zinnhaltige Lotpaste aufgestrichen, dann platziert ein Roboter die Teile auf der Paste und anschließend werden die Platinen in einem Ofen erhitzt, sodass die Paste schmilzt und nach dem Abkühlen eine Verbindung entsteht. Selbst SMD-Platinen zu fertigen und zu bestücken ist eine Disziplin für sich, um die es hier nicht gehen soll.

Mit einer Heißluftstation sind Sie aber in der Lage, selbst defekte SMD-Bauteile auszutauschen – zum Beispiel eine defekte USB-Buchse an der Powerbank oder einen durchgebrannten Kondensator. Mit dem Lötkolben sind Sie da in der Regel chancenlos, weil alle Kontakte gleichzeitig

erhitzt werden müssen, um ein Teil zu entfernen. Bei integrierten Schaltungen (IC) sind die Kontakte auch mal auf der Unterseite, da kommt kein Kolben hin.

Tipp zum Üben: Schnappen Sie sich ein altes Mainboard oder andere defekte Elektronik, die schon für den Elektroschrott vorgesehen war und nehmen Sie sich zunächst vor, gezielt einzelne Teile auszubauen, ohne mit der heißen Luft benachbarte Komponenten mitzubewegen. Danach besorgen Sie sich selbst eine Tube Lotpaste und bauen Sie Teile unterschiedlicher Größe wieder ein.

Eine Heißluftstation kann aber noch mehr: Zum Beispiel Schrumpfschlüsse schrumpfen und Klebstoffe erweichen, die anders nicht nachgeben wollen. Auch die Klebebänder, mit denen die Displays von Smartphones verklebt sind, geben mit warmer Luft irgendwann nach, sodass man das Display entnehmen kann.

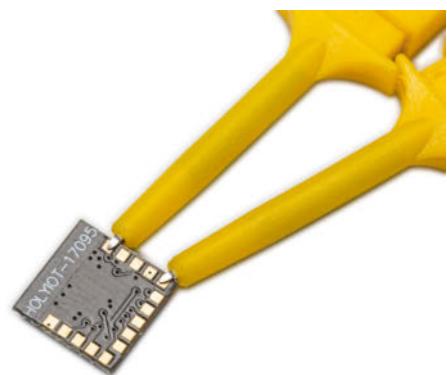
## Viele helfende Hände

Zu den Dingen, die man in der Werkstatt nicht genug haben kann, gehören Hände. Weil die meisten Bastler nur mit bis zu zweien ausgestattet sind, müssen sie sich mit mechanischem Ersatz behelfen. Platinen hält ein „verstellbarer Leiterplattenhalter“ sehr gut, wie die Konstruktion im Onlinehandel heißt, die Sie auf 32 oben links im Bild sehen. Gerade bei Durchsteckmontage muss man die Platine ja immer wieder umdrehen, was damit sehr gut klappt. Kostenpunkt: 15 Euro. Für viele Arbeiten haben sich Schwanenhälse mit Krokodilklemmen bewährt (ab 10 Euro), die es in verschiedenen Varianten gibt.

Eine sinnvolle Anschaffung, wenn man die Kontakte an filigranen Chips abgreifen muss: Messklemmen wie die auf auf dieser Seite oben rechts abgebildeten. Kostenpunkt für ein Set: 10 Euro.

## Teure Elektronik

Von den erschwinglichen Kleinteilen kommen wir nun zu den großen Anschaffungen und teils jahrelangen Träumen vieler Bastler. Ein Oszilloskop steht oft auf der Wunschliste und ein solches ist auch längst nicht mehr so unerschwinglich wie noch vor 20 Jahren. Schon für rund 350 Euro bekommt man ein anständiges digitales Vier-Kanal-Oszilloskop [1]. Wem es aber vor allem darum geht, hin und wieder einen vertrackten Fehler an einem I2C-Sensor (mehr dazu ab S. 50) im Zusammenspiel mit dem ESP32 aufzuspüren, der muss nicht so viel ausgeben: USB Logic



**Kleine Helfer:** Messklemmen kosten nur ein paar Euro und machen das Leben leichter, wenn man Chips programmieren oder ein Signal abgreifen will.

Analyzer messen anders als ein Oszilloskop nicht die Spannungen an den Kanälen, sondern erfassen nur High und Low und stellen diese Werte in einem Diagramm über die Zeit dar – komfortabel in einer Software am PC. Das reicht für Protokolluntersuchungen völlig aus. Im Wimmelbild auf Seite 32 sehen Sie einen solchen Logic Analyzer, der acht Kanäle aufzeichnet und knapp 15 Euro gekostet hat. Er arbeitet zwischen 0 und 5 Volt, alles über 2 Volt wertet er als High.

Ein weiteres Gerät drängt auf den Wunschzettel, wenn man einmal wieder frustriert in der Sammelkiste mit alten Netzteilen kramt, in der Hoffnung, zufällig eines herauszuziehen, dessen Spannung und Leistung ungefähr zu einer Bastelei passen: ein Labornetzteil, also eine regelbare Spannungsquelle. Für die günstigsten Modelle kann man um 80 Euro einplanen, für Raspi- und ESP-Bastuprojekte reicht ein solches absolut aus.

## Fazit

Ganz ohne Werkzeug kein Bastelspaß. Vielleicht haben Sie ja das ein oder andere Werkzeug entdeckt, das einen Platz auf Ihrem Weihnachtswunschzettel findet. Oder Sie haben endlich die eine sinnvolle Anwendung für einen 3D-Drucker gefunden und freuen sich schon darauf, ein Ordnungssystem selbst zu drucken. Über ct.de/y8kk finden Sie Links zu vielen der angesprochenen Produkten sowie die Druckvorlagen.

(jam@ct.de)

## Literatur

[1] Jan Mahn, Wellengucker, Einstieg in die Arbeit mit dem Oszilloskop, c't 16/2021, S. 162



# Durch Feuer und Wasser

## Wer haftet, wenn Marke Eigenbau Schäden verursacht

**Basteln soll vor allem Spaß machen und idealerweise ein praktisches Ergebnis liefern. Doch wer haftet, wenn durch einen Fehler im System die Hütte brennt? Einige zivilrechtliche Vorschriften sollten Sie im Hinterkopf behalten.**

Von Till Jasper Dölle

**W**er arbeitet oder bastelt, macht nun mal auch Fehler. Zudem ist die Trial-and-Error-Methode im Bastelkeller ein bewährtes Konzept. Kompliziert wird es aber, wenn durch Eigenbauten Schäden entstehen. Dann stellt sich schnell die Frage: „Und wer zahlt das jetzt?“

Selbst wenn man fertige Komponenten wie einen Raspi, ESP32 oder Arduino verwendet, sind Malheure denkbar. Im Gerät selbst kann ein Fehler auftreten, man kann es falsch in ein Netzwerk integrieren oder überlasten. Fehler können auch in der verwendeten Software stecken, die dann Systemausfälle verursacht.

Im Extremfall sind teure Folgen denkbar: Vielleicht fällt das sonst automatisch

schließende Garagentor plötzlich herunter und beschädigt dabei ein Auto. Das selbst programmierte Pumpensystem für den Garten könnte versagen und auf einmal steht der Keller unter Wasser. Oder es brennt einfach das Gerät durch und verursacht ein größeres Feuer. Dieser Artikel beleuchtet die Frage, wer, wann und auf welcher Grundlage in der zivilrechtlichen Haftung steht und wie sich Freundschaftsdienste von Hilfe gegen Geld unterscheiden.

### Rechtliche Grundlagen

In der Regel haftet finanziell derjenige, der die Verantwortung dafür trägt, dass ein Schaden entstanden ist. Wem diese Verantwortung zufällt, richtet sich nach den gesetzlichen und vertraglichen Regelungen. Die wesentlichen Gesetze zu Haftungsfragen findet man im Bürgerlichen Gesetzbuch (BGB). Relevant sind insbesondere die Vorschriften aus dem Schuldrecht. Neben den Gewährleistungsrechten aus dem gesetzlichen Vertragsrecht hat auch das im BGB enthaltene Deliktsrecht eine große Bedeutung. Die Vorschriften beider Gebiete schließen sich nicht gegenseitig aus, sondern kommen grundsätzlich nebeneinander zur Anwendung. Zusätzlich zu den Regelungen des BGB wird im Bereich der Softwarehaftung zukünftig auch das sogenannte Produkthaftungsgesetz eine noch größere Rolle spielen.

Vereinfacht betrachtet kann eine Haftung aus vertraglichen Gewährleistungsrechten nur im Verhältnis zwischen Personen greifen, die miteinander einen Vertrag geschlossen haben. Ein solcher Antrag begründet gegenseitige Leistungs- oder Rücksichtnahmepflichten. Verletzt eine Seite ihre Pflichten mindestens fahrlässig, können Schadensersatzansprüche entstehen.

Die Haftung nach dem Deliktsrecht hingegen greift unabhängig davon, ob beide Seiten einen Vertrag geschlossen haben. Das Deliktsrecht schützt alle Personen davor, dass jemand seine Rechte schuldhafte und widerrechtlich beeinträchtigt. Insbesondere bei sogenannten schuldhaften Verletzungen des Lebens, der Gesundheit, der Freiheit und des Eigentums begründet das Deliktsrecht Schutz- und Ersatzansprüche.

Kommen bei einem Schadensfall sowohl vertragliche als auch deliktische Ansprüche in Frage, kann sich der Geschädigte aussuchen, ob er seinen Anspruch auf das eine oder das andere stützt. Oft dürfte er sich dabei für vertragliche Schadensersatzansprüche entscheiden, weil sie für ihn meist günstiger ausfallen. Das vertragliche Gewährleistungsrecht enthält an einigen Stellen besondere Vermutungs- und Beweislastregeln. Sie machen es vor allem Verbrauchern erheblich leichter, ihre Rechte durchzusetzen.

Für den Bastler, der möglicherweise haftet, hat es ebenfalls Folgen, ob das Delikts- oder das vertragliche Gewährleistungsrecht angewandt wird. Für ihn ist es unter Umständen sogar von Vorteil, wenn kein Vertrag besteht. Damit scheidet nämlich eine umfassendere vertragliche Haftung aus. Hinzu kommt, dass die Klägerseite dem Bastler im Streitfall oft schwieriger nachweisen kann, dass ihr nach dem Deliktsrecht ein Anspruch zusteht. Und nach diesem Recht obliegt es dem vermeintlich Geschädigten, einen Schaden zu beweisen.

Eine verschuldensunabhängige Haftung für digitale Produkte findet sich zudem im Produkthaftungsgesetz. Nach diesem haftet der „Hersteller eines fehlerhaften Produktes verschuldensunabhängig“ für bestimmte Schäden, die aus den Produktfehlern entstanden sind. Voraussetzung hierfür ist insbesondere, dass der Hersteller ein fehlerhaftes Produkt in Verkehr gebracht hat und er sich einer Haftung nicht entziehen kann. Nicht haften würde er zum Beispiel, wenn eine von ihm

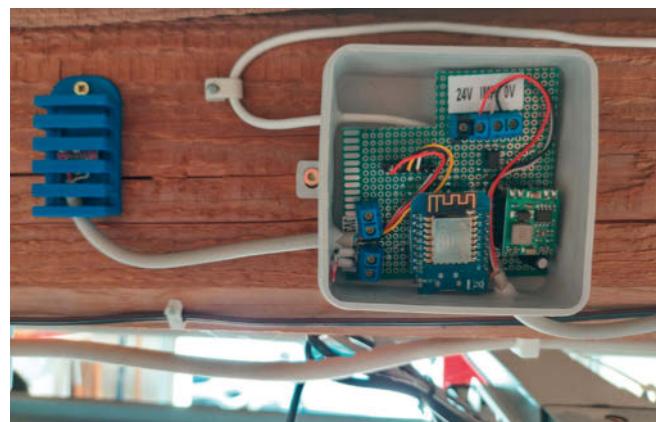
## c't kompakt

- Trotz aller Sorgfalt können bei Eigenbauten Fehler passieren, die im schlimmsten Fall zu großen Schäden führen. Meistens regelt das Bürgerliche Gesetzbuch die Haftung.
- Hat man im privaten Rahmen selbst gebaut, muss normalerweise der Geschädigte dem Bastler eine Schuld nachweisen.
- Sobald man etwas für andere tut und dafür Geld fließt, greifen verschärzte Regelungen, die den Abnehmer begünstigen.

gefertigte Platine aufgrund eines insgesamt fehlerhaften Eigenbaus überhitzt und ein Feuer verursacht.

Ein gewerblicher Hersteller muss nach dem Produkthaftungsgesetz Schäden an Leib und Leben, an der Gesundheit und an bestimmtem Privateigentum ersetzen. Nicht ersetzen muss er allerdings die Schäden an dem erworbenen Produkt selbst. Grundsätzlich sieht das Produkthaftungsgesetz eine Selbstbeteiligung des Geschädigten in Höhe von 500 Euro vor. Gleichzeitig begrenzt es das verbleibende Haftungsrisiko des Herstellers auf maximal 85 Millionen Euro.

Weil zumindest zukünftig auch Software ein Produkt im Sinne dieser Vorschriften sein wird, könnte das Produkthaftungsgesetz auch bei Eigenbauten an Bedeutung gewinnen. Zwar dient das Gesetz in erster Linie dem Verbraucherschutz gegenüber Unternehmen. Dennoch könnten auch Hobbybastler, die ihre Entwicklungen an andere verkaufen, unter diese Vorschriften fallen.



## Praktische Anwendungsfälle

Nun kommen die oben vereinfacht dargestellten Grundsätze zwar in vielen Fällen zum Tragen. Dennoch unterscheidet sich der Sachverhalt beispielsweise je nachdem, wie jemand eine Raspi-Steuereinheit oder generell einen Eigenbau konzipiert und betreibt. Am häufigsten dürften Schäden vorkommen, die dem Bastler selbst entstehen. Verwendet dieser zum Beispiel einen Raspi als Steuerelement und es kommt zu einem Systemausfall, könnten Schäden am Gerät selbst und an anderen Sachen entstehen.

Für die Haftung kommen dann drei Akteure in Betracht. Entstehen Schäden zweifelsfrei aus Versäumnissen des Bastlers selbst, wird dieser auf seinen Kosten sitzen bleiben. Haben jedoch der verwendete Raspi oder eine andere Komponente den Schaden verursacht, könnten dem Bastler Ansprüche gegenüber dem Verkäufer oder Hersteller des Geräts zustehen.

Der Verkäufer des Geräts kommt dann für eine Haftung in Frage, wenn er dieses bereits in einem mangelhaften Zustand an den Käufer übergeben hat. Das wird der Bastler im Nachhinein zwar nur schwer beweisen können. Aber an dieser Stelle hilft, zumindest einem Hobbybastler, die verbraucherfreundliche Gesetzgebung der Europäischen Union.

Hat er beispielsweise einen Raspi von einem Unternehmen erworben und geht das Gerät im ersten Jahr nach der Übergabe kaputt, greift eine gesetzliche Vermutung: Der Bastler darf dann davon ausgehen, dass der Raspi bereits zum Zeitpunkt der Übergabe mangelhaft gewesen ist. Das hilft ihm nicht nur bei einer Reklamation. Dieselbe Annahme erhöht auch seine Chancen, dass ihm der Verkäufer etwaige Begleitschäden ersetzen muss. Zwar hat der Verkäufer die Möglichkeit, die Vermutung zu widerlegen und zu be-

**Manch einer baut die Fernsteuerung für sein Garagentor erfolgreich selbst. Sollte solch ein Tor aber durch einen Systemfehler unkontrolliert herunterraschen, könnte ein hoher Schaden entstehen.**

weisen, dass der Raspi keine Mängel aufwies. Das wird in der Praxis oft aber kaum möglich sein.

Neben dem Verkäufer haftet möglicherweise auch der Hersteller. Zwar haben der Hersteller und der Bastler keinen Vertrag geschlossen. Eventuell muss der Hersteller aber nach dem Produkthaftungsgesetz für den Schaden oder Teile davon aufkommen. Greift dieses Gesetz, trägt der Bastler zwar den Eigenanteil von 500 Euro. Darüber hinaus trägt die Kosten jedoch der Hersteller. Damit das Produkthaftungsgesetz greift, muss der Raspi bereits fehlerhaft gewesen sein, als der Hersteller ihn in Verkehr gebracht hat. Er muss sich also quasi von vorneherein nicht für den normalen Gebrauch geeignet haben. Zusätzlich darf es für den Hersteller nicht unmöglich gewesen sein, dass ihm der Fehler auffällt.

Nimmt der Bastler den Verkäufer oder den Hersteller auf Schadensersatz in Anspruch, muss er sich seine eigenen Versäumnisse aber grundsätzlich weiter anrechnen lassen. Wer am Ende welchen Anteil an welchen Schäden bezahlen muss, bleibt damit von den Umständen des Einzelfalls abhängig.

## Projekte für Nachbarn und gewerblicher Vertrieb

Vielleicht kommt das Bastelprojekt aber auch bei anderen Personen als dem Bastler selbst zum Einsatz. So könnte er seinem Nachbarn eine automatisch gesteuerte Pumpenanlage für die Balkonbewässerung eingerichtet haben, wie er sie auch selbst seit Jahr und Tag nutzt. Plötzlich spielt das System verrückt und dem Nachbarn entstehen Wasserschäden. Der könnte seinen bastelnden Bekannten nun auffordern, ihm die Kosten für die Beseitigung des Wasserschadens zu ersetzen. Auch in diesem Fall kommen wieder die Ansprüche aus den eingangs erläuterten Rechtsgebieten in Betracht.

Nach dem vertraglichen Gewährleistungsrecht muss der Bastler nur haften, wenn er mit seinem Nachbarn auch einen Vertrag geschlossen hat. Wenn keine schriftliche Vereinbarung existiert, heißt das aber noch lange nicht, dass ein Vertrag fehlt. Verträge kann man grundsätzlich auch mündlich schließen. Ob ein Vertrag vorliegt, würde ein Richter daher aus einer Würdigung der Gesamtsituation schließen.

Ob Geld geflossen ist oder nicht, kann dabei entscheidende Auswirkungen



**Wer eine Hydroponik auf dem Balkon wie hier gezeigt sorgfältig installiert, dürfte von Wasserschäden verschont bleiben. Wer schlamppt, muss bei Schäden unter Umständen haften.**

haben. Auch die Interessen der Beteiligten spielen eine Rolle. Will der Bastler seinem Nachbarn nur einen freundschaftlichen Gefallen tun, spricht das gegen einen Vertragsschluss. Lässt er sich die Arbeit vergüten, spricht das eher dafür, dass beide einen Vertrag geschlossen haben.

Eine Haftung nach dem Deliktsrecht kommt hingegen immer in Betracht. Das gilt selbst dann, wenn beide Seiten ganz bewusst keinen Vertrag geschlossen haben. Hat der Bastler den Schaden am Haus des Nachbarn durch einen eigenen Fehler fahrlässig oder vorsätzlich verursacht, muss er diesem die Kosten grundsätzlich ersetzen. Zwar vertreten einige Juristen in diesem Kontext auch die Ansicht, dass der Bastler erst bei grober Fahrlässigkeit haftet. Die Gerichte folgen dieser Auffassung in der Regel aber nicht.

Zur groben Orientierung: Fahrlässig handelt man, wenn man mit seiner Erfahrung und seinem Wissen einen Fehler hätte erkennen können oder seine Fähigkeiten überschätzt, grob fahrlässig, wenn man beispielsweise eine offensichtliche Schwachstelle wie ein durchgescheuertes Kabel ignoriert. Diese Frage spielt übrigens auch gegenüber einer Haftpflichtversicherung eine Rolle, wenn diese einspringen soll.

Dennoch verbleiben zwei gute Nachrichten für jeden Hobbybastler: Zum einen haftet man als solcher in aller Regel nicht verschuldensunabhängig. Zwar hat man

gegebenenfalls ein fehlerhaftes Produkt hergestellt. Das Produkthaftungsgesetz bleibt aber außen vor, solange die Apparatur nicht im Rahmen der beruflichen Tätigkeit entstanden oder für einen Verkauf beziehungsweise einen anderen Vertrieb mit wirtschaftlichem Zweck hergestellt worden ist. Zum anderen bleibt die Möglichkeit, dass Ansprüche des Bastlers gegen den Hersteller oder den Verkäufer einer fehlerhaften Komponente bestehen – also zum Beispiel eines Raspi oder einer Pumpe. In solch einem Fall kann sich eine rechtliche Beratung doppelt lohnen.

Kommt ein Hobbybastler mit zunehmender Erfahrung allerdings auf die Idee, aus seinem Hobby eine fortlaufende Einnahmequelle zu machen und zum Beispiel eine selbst konstruierte Bewässerungssteuerung auf eBay anzubieten, ändern sich die Vorzeichen komplett: Er agiert dann gewerblich. Dabei steigen insbesondere seine Haftungsrisiken.

Hat er in der Entwicklung Fehler gemacht, haftet er gegenüber Kunden grundsätzlich als Vertragspartner und gegebenenfalls sogar als Unternehmer. Wie in den bereits beschriebenen Fällen kommt außerdem eine Haftung nach dem Deliktsrecht in Betracht. Nicht zuletzt muss unser Bastler sich darauf einstellen, dass das Produkthaftungsgesetz samt seiner verschuldensunabhängigen Haftung für seine eigenen Produkte greift.

## Fazit

Basteln ist rechtlich immer auch mit Haftungsrisiken verbunden. Das gilt unabhängig davon, ob ein Bastler für sich selbst oder andere aktiv wird. Verursacht er durch eigene Fehler Schäden, liegt es oft an ihm, für deren Ersatz aufzukommen. Dennoch gibt es oftmals nicht nur den einen Verantwortlichen für alle Schäden. Je nach Situation können für den Bastler und den Geschädigten auch Ansprüche gegen den Verkäufer oder den Hersteller einer Komponente bestehen. Es kann sich also lohnen, solche Ansprüche anwaltlich prüfen zu lassen.

Gleichzeitig können wir Entwarnung geben: Selbst unseren Kollegen vom Make Magazin sind keine größeren Katastrophen durch Bastelprojekte bekannt, und kleinere Malheure bleiben ebenfalls die Ausnahme. Kommt es doch zum Schaden, wird in vielen Fällen die private Haftpflichtversicherung weiterhelfen. Es lohnt sich daher, deren Bedingungen zu kennen.

(mon@ct.de)



DIE KONFERENZ FÜR  
SOFTWARE-ARCHITEKTUR

# BRIDGING THE GAP

3.-7. FEBRUAR 2025

ICM MÜNCHEN

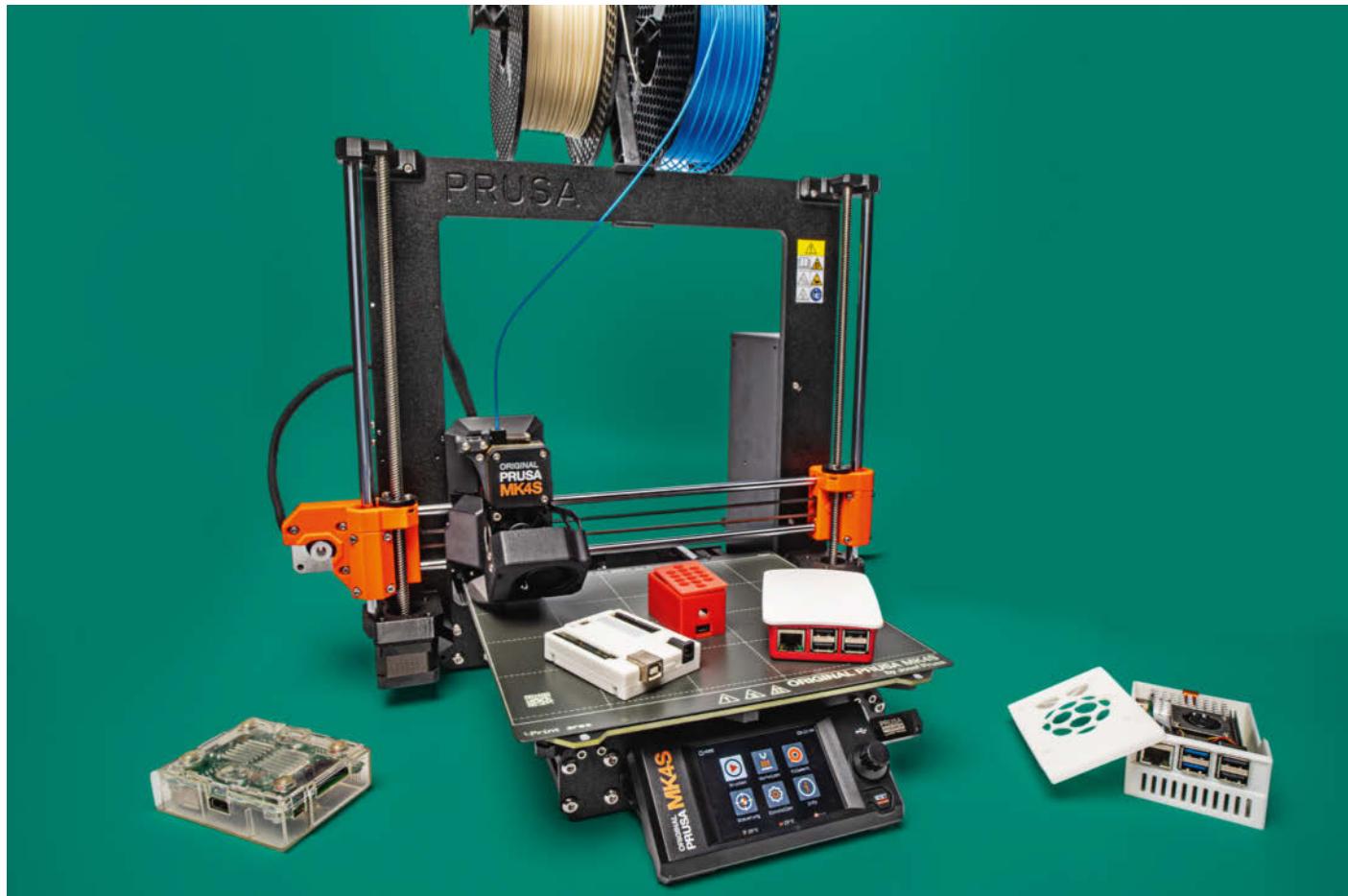
DER WISSENS-HUB  
FÜR SOFTWAREENTWICKLUNG

Bis 20.12.24  
Preisvorteile sichern



[WWW.OOP-KONFERENZ.DE](http://WWW.OOP-KONFERENZ.DE)

© Copyright by Heise Medien.



# Reichlich Rüstungen für Ritter Raspi

## Gehäuse fürs Bastelprojekt

**Nicht alles, was gebastelt ist, muss auch nach Bastelei aussehen. Mit einem maßgeschneiderten Gehäuse aus eigener Produktion verhüllen Sie die Technik und schützen sie obendrein.**

Von Michael Link

Bastelprojekte – und seien sie noch so genial – finden im Familienrat erst dann Gnade, wenn sie nicht nach Bastelei aussehen, sondern die im Innern schlummernde Technik möglichst unsichtbar eingepackt ist. In spezialgelaerten Sonderfällen will der Selbstbau auch besonders effektvoll präsentiert sein, was man beispielsweise durch reichhaltige Beigabe von Lichteffekten und durch gewagte Formgebung des Gehäuses für Gerät und Peripherie wie Sensoren oder Aktoren erreichen kann.

Gleich, wie man es gern hätte, man sollte alles, was elektronisch ist, sowieso stets einpacken. Die Platinen der Bastelrechner von Raspberry, Arduino und anderen erzeugen zwar keine für Menschen

gefährlichen Spannungen, jedoch besteht andersrum für die kleinen Rechner Gefahr. Man kann nämlich der empfindlichen Elektronik beim Berühren eine elektrostatische Entladung verpassen, die sie unvermittelt in den Elektronenhimmel schicken kann.

### Wer suchet, der fluchet

Der einfachste Weg, zu einem Gehäuse zu kommen, besteht darin, ein offizielles bei dem Hersteller zu bestellen, von dem man seinen Mikrorechner ordert. Die Kosten halten sich in Grenzen: Für einen Fünfer hat man sein Schutzkästlein. Dem Ästheten sträuben sich bei einem solchen Einheitskleid aber die Haare selbst auf der poliertesten Glatze.

Wer sich in den einschlägigen Shops für Kleinrechner umtut, entdeckt dort diverse Gehäuse anderer Firmen, die auch sehr besondere Anwendungsfälle berücksichtigen. Am häufigsten finden sich Umhüllungen beispielsweise für den Raspberry Pi 4; solche für den neueren Pi 5 sind noch weniger verbreitet. Besonders für Rechner im Dauerbetrieb empfiehlt sich eine großzügig bemessene Wärmeabfuhr, also viel Platz im Gehäuse für Kühlkörper beziehungsweise Lüfter. Beides hilft, den schädlichen Betrieb der CPU oberhalb ihrer Temperaturgrenzen zu vermeiden. Wem Kühlkörpertürme zu stark aufdringen, der greift eher zu Lüftern, die allerdings stets Geräusche machen. Es gibt sogar Fertiggehäuse, die bereits einen Lüfter integrieren. Gehäuse mit Kühlrippen oder Lüfter sind ab 11 Euro zu haben.

Fertiggehäuse gibt es auch für eine Kombination aus Rechnerplatine und Display. Das KKS-B Gehäuse für 33 Euro beispielsweise fasst einen Raspi 4 oder 5 mit samt des von der Raspi-Foundation empfohlenen Touchscreens im 7-Zoll-Format. Ebenfalls nützlich können Gehäuse für spezielle Befestigungsorte sein, etwa Hutschienengehäuse oder solche für VESA-Mounts.

Schwieriger wird die Gehäusesuche, wenn Sie keinen der weitverbreiteten Raspi- oder Arduino-Kleinrechner haben, sondern einen exotischeren Typ. Das Gleiche gilt, wenn Sie einen externen Sensor schützend einpacken wollen. Für den Kleinrechner Nvidia Jetson Nano beispielsweise ist außer O8/15-Schuhschachteln nicht viel zu bekommen.

Nun kann man sich fragen, was denn so schlecht an einfachen Gehäusen sein soll. Im Grunde genommen nichts, jedoch passen sie nicht immer zum Anwendungsfall. Gerade Kleinrechner wie der Raspi oder Arduino werden mit bestückten Anschlüssen und Steckplätzen so moppelig, dass alles zusammen nicht mehr in ein Standardgehäuse passt.

Vielfach werden Platinchen huckepack (HAT) auf Kleinrechner wie den Raspi oder Arduino gepackt. Auch Anbauteile wie Sensoren, Displays oder Kamera-module brauchen entweder ein großes Gehäuse, das alle Komponenten aufnimmt, oder man benötigt mehrere Gehäuse für die Einzelkomponenten. Für viele dieser spezialgelagerten Sonderfälle gibt es nichts von der Stange. Das gilt ganz besonders für Peripheriebausteine. Woher also nehmen?

## ct kompakt

- Gehäuse schützen die Elektronik und sorgen für eine bessere Akzeptanz von Bastelprojekten.
- 3D-Druckdesigns kann man auch benutzen, wenn man keinen eigenen Drucker hat.
- Weil es schon viele 3D-Modelle gibt, muss man sie häufig nur etwas anpassen – eine totale Neukonstruktion ist selten nötig.

## Selbst drucken muss nicht sein

Wer gerade Zigarrenkisten oder ausgediente Gefrierdosen zur Hand hat, kann die natürlich zweckentfremden. Aber, mit Verweis auf den erwähnten eher mäßigen Akzeptanzfaktor, wäre ein genau zum Zweck passendes Fertiggehäuse sicher besser. Das gibt es vielleicht nicht materiell zu kaufen, aber man kann sich aus dem Netz Baupläne dafür besorgen und die Teile mit einem 3D-Drucker selbst anfertigen. Wenn man keinen hat, lohnt der Besuch eines Maker Space. Hier trifft man in der Regel auch auf kundige und hilfsbereite Menschen, die beim Drucken helfen. Alternativ kann man sein Gehäuse auch bei einem 3D-Druckservice in Auftrag geben. Ein Raspi-Gehäuse auszudrucken, das auch den HAT-Aufsatz integriert, kostet beispielsweise 13 Euro. Kos-

tenrechner dafür finden sich im Netz, auch Slicer-Programme, welche die 3D-Modelle in detaillierte Druckanweisungen umsetzen, schätzen die Kosten zumindest grob, aber nur für den Material- und Zeitaufwand. Druckdienstleister schlagen diesbezüglich noch einiges auf, damit sich ihr Maschinenpark rechnet.

Ob Sie für den Druck einen Dienstleister, Freunde oder Bekannte einspannen oder mit dem eigenen 3D-Drucker fertigen: Sie brauchen eine Datei mit einem druckfähigen 3D-Modell dafür. Fertige Modelle gibt es auf vielen Konstruktionsplattformen, auf denen Nutzer ihre Eigenkonstruktionen kostenfrei oder gegen eine kleine Zahlung anbieten.

Am bekanntesten dürfte die Plattform Thingiverse sein (Links zu allen erwähnten Plattformen unter ct.de/ytz6). Hier gibt es eine große Auswahl an fertigen 3D-Modellen, die heruntergeladen und gedruckt werden können. Auf GrabCAD finden sich technische und industrielle Designs, aber auch viele Gehäuse für Kleinrechner. Bei der Plattform MyMini-Factory kosten viele der tendenziell eher hochwertigen und geprüften Druckdateien Geld. Allerdings ist wenig für Rechnerbastler im Katalog. Anders sieht es bei Printables aus. Auch wenn diese Plattform so aussieht wie eine Quelle für Drucker des Herstellers Prusa, sind die 3D-Modelle auch für Drucker anderer Hersteller nutzbar.

Auf Cults3D finden sich Modelle für sehr viele Themenbereiche. Tippt man in

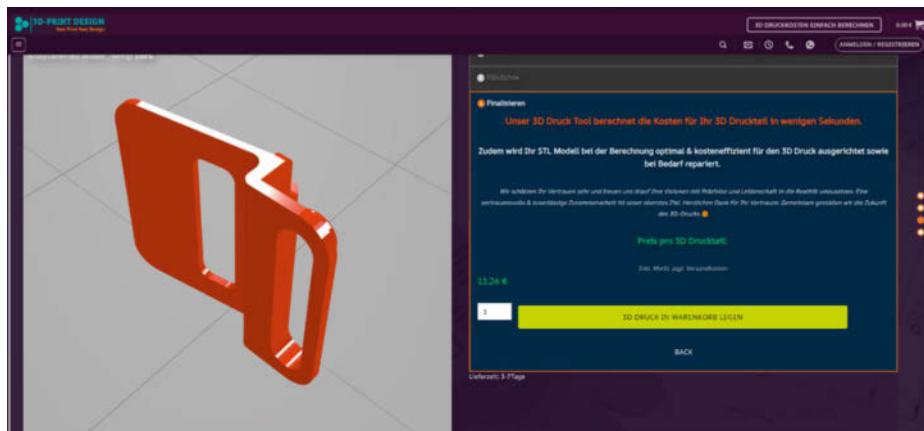
**Argon NEO 5 BRED**  
Gehäuse für  
Raspberry Pi 5 mit  
eingebautem Lüfter

16,99 €  
inkl. 19% USt., zzgl. Versand (Standard-Versand)

Hersteller: Argon 40 Limited  
Kategorie: Gehäuse  
Artikelnummer: ebt20900  
Mehr anzeigen

Für dieses Produkt erhältst du **16 Bonuspunkte** im Wert von 0,16 €.

Wer einen Raspi mit einem Lüfter anstelle eines Kühlkörpers vor dem Hitzetod schützen will, bekommt in etlichen Onlineshops fertige Gehäuse samt Lüfter.



**Hat ein 3D-Druckdienstleister einen Kostenrechner auf seiner Seite, sieht man nach dem Upload seines 3D-Modells, was man jeweils für unterschiedliche Ausführungen berappen muss.**

der Suchleiste „Raspberry Pi“ ein, erhält man 3900 Treffer. Die dünnen sich allerdings bis zum Redaktionsschluss zu neun Designs aus, wenn man nur kostenlose, gut bewertete und nachgedruckte Modelle filterte. Tinkercad wiederum ist kostenlos und richtet sich in seiner Einfachheit vornehmlich an Jugendliche. Man findet dort viele Modelle, die man auch gleich weiterbearbeiten und verändern kann. Die Werkzeuge dazu sind limitiert, man bekommt das Konstruieren und Verändern von Modellen aber auch als Laie schnell in den Griff.

Es gibt viele weitere Plattformen. Wer sie nicht alle einzeln durchsuchen will, kann auch auf eine Meta-Suchmaschine für 3D-Modelle zurückgreifen. Der Katalog von Yeggi umfasste zum Redaktionszeitpunkt etwa 5,6 Millionen Objekte.

## Alternativen

Auch das Durchforsten der Websites von Kleincomputer- und Zubehörherstellern fördert im Support-Bereich gelegentlich Druckdaten zutage. Idealerweise sind das passende Gehäuse, aber zuweilen gibt es immerhin ein 3D-Modell des Computers oder des Zubehörs. Um diese digitalen Dummys herum kann man dann eigene Gehäuse konstruieren. So ein Dummy eignet sich aber auch zum Prüfen der Passgenauigkeit von Gehäusemodellen aus fremden Quellen.

Findet man gar nichts, was genau zum Zweck passt, lohnt es sich, auf Plattformen nach etwas Ähnlichem zu suchen und es – je nach Können und Anspruch – mit einer 3D-Software zu modifizieren. Dazu lädt man das Modell in einem Format herunter, mit dem die Modelliersoftware etwas an-

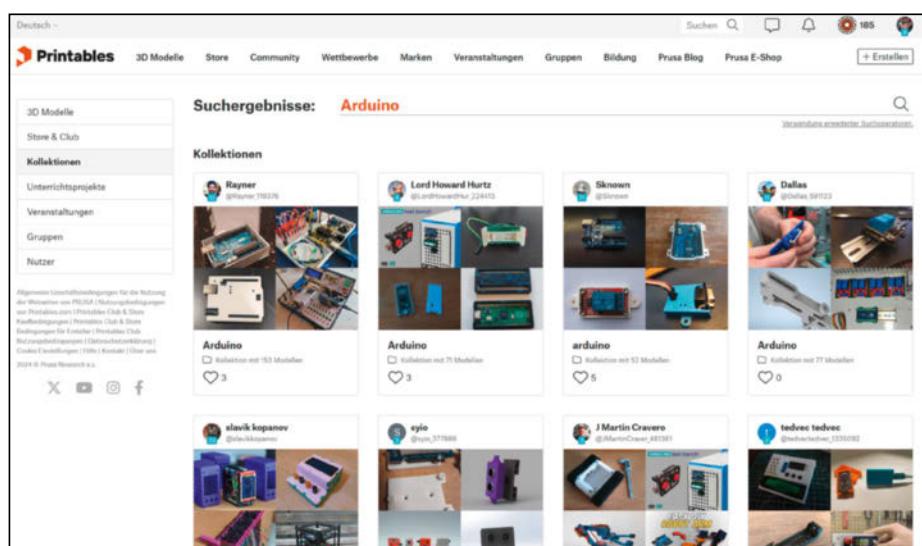
fangen kann. Bei Tinkercad beispielsweise sind die Formate OBJ, STL sowie 3MF und SVG importierbar. OBJ- und 3MF-Dateien sind flexibler als STL-Dateien und unterstützen Farben und Texturen, was sie für detaillierte Modelle und Farb-3D-Drucke geeigneter macht.

Durch Addition oder Subtraktion von Grundformen verändern Sie die Geometrie nach Wunsch. Das ist aber zuweilen eine elende Frickelei, denn die Dateien enthalten oft lediglich ein Gesamtobjekt, sodass die beim Konstruieren genutzten Gruppierungen von Elementen nicht mehr einzeln manipulierbar sind. Wünschenswert wäre das beispielsweise, wenn Sie zwar das Gehäuse nutzen wollen, aber beispielsweise die Öffnungen für Buchsen verlegen müssen, weil sich die Grundpla-

tine geändert hat. So bleibt für Tinkercad-Nutzer der Weg, manuell Material im Modell zu ergänzen oder wegzunehmen. Also: falsch gesetzte Bohrungen mit Objekten digital zuzuspachteln und Bohrungen an anderer Stelle hinzuzufügen. Wie weiter oben angesprochen, kann man sich ein digitales Modell dessen, was eingepackt werden soll, herunterladen und schauen, ob es perfekt ins Gehäusemodell passt. Das kann manchen Fehldruck ersparen.

Für anspruchsvollere Aufgaben wie das Vergrößern eines Gehäuses bei gleichbleibender Wandstärke einiger Teile sind Modelle besser, die gruppierte Bestandteile enthalten. Die können Sie mit potenteren Werkzeugen weitgehend umdeneln, sogenannten parametrisch arbeitenden 3D-Konstruktionsprogrammen. Sie lassen Änderungen ausgewählter Parameter zu, während andere unbeeinflusst bleiben. Für aufwendigere Vorhaben werden Fusion360, Onshape sowie Solidworks, OpenSCAD und FreeCAD immer wieder genannt. Eine Einführung in die kostenlose CAD-Anwendung Onshape, die komplett im Browser läuft, haben wir bereits veröffentlicht [1]. Daneben gewinnt Plasticity Nutzer, die eher künstlerische Ambitionen haben [2]. Welche Software für Sie am besten ist, kann man nicht pauschal sagen. Es hängt unter anderem davon ab, welche Erfahrungen Sie im Bereich der 3D-Konstruktion schon haben, welcher Workflow Ihnen am besten passt und schlicht vom Preis.

Der Workflow beim Konstruieren orientiert sich vereinfacht an diesem Sche-



**3D-Druckplattformen enthalten zig Gehäusemodelle für verbreitete Kleinstrechner, aber nur wenige für Exoten und besondere Anwendungsfälle.**

Thingiverse

[Non-functional] Raspberry Pi 2/3 For case development

OculusQuest April 27, 2020 0 6 1

Download all files

Post a make Post a remix Collect Watch Report thing

Ein Dummy, etwa der unter anderem auf der Plattform Thingiverse zu findende für den Raspi, eignet sich hervorragend, um damit zu prüfen, ob ein dafür konstruiertes Gehäuse passt.

ma: Messen Sie Ihren Kleinstrechner und notieren Sie die Abmessungen sowie Positionen von Anschlüssen und Befestigungspunkten. Starten Sie dann mit einem einfachen Design, also: Erstellen Sie in der Software Ihrer Wahl einen rechteckigen Kasten als Grundform. Berücksichtigen Sie dabei auch Toleranzen, sodass Sie nicht am Ende zur Feile greifen müssen, damit Ihre Geräteplatine hineinpasst. Fügen Sie Löcher und Aussparungen für Anschlüsse, Lüftungsschlitzte und Befestigungspunkte hinzu. Schon zur Kür gehören dann weitere Details wie Beschriftungen, Logos oder dekorative Elemente. Es ist eine gute Idee, jeweils Zwischenentwürfe zu speichern, damit man diese für weitere Entwürfe etwa für nachfolgende Geräte aufgreifen kann. Wählen Sie zum Export eine Datei, die ein Slicer für den 3D-Druck benutzen kann, also STL oder OBJ. Das STEP-Format ist ebenfalls sinnvoll, etwa wenn Sie das Modell in anderen Programmen weiterverarbeiten wollen.

### Modell drucken

Hat man die Datei, stellt sich die Frage: Wer druckt sie? Wer nur alle Jubeljahre mal

einen Druckjob hat, geht damit lieber zu einem Maker Space in der Nähe oder schickt den Auftrag an einen Fertiger. Im Netz gibt es etliche Adressen dafür, aber nur wenige fertigen Einzelstücke für Privatleute. Die Druckkosten und Lieferzeiten fallen je nach Anbieter und Material sehr unterschiedlich aus, sodass wir Seltendruckern raten, Anbieter zu bevorzugen, die auf ihren Websites Druckkostenrechner vorhalten. Idealerweise kann man darin seine 3D-Modelle hochladen, das Material sowie die Fülldichte für den Druck auswählen und sieht sofort, was das kostet. Einige Anbieter berechnen die Kosten auch, wenn Sie ein paar Parameter angeben können, etwa wie viel Material sie benötigen und wie lange der Druck auf einer bestimmten Maschine dauern würde. Diese Angaben spucken Ihnen Slicer-Programme für 3D-Drucker aus, die also auch dann sinnvoll sein können, wenn Sie gar keinen 3D-Drucker besitzen.

Aber selbst wenn Sie einen 3D-Drucker haben, kann es sinnvoll sein, einen Druckdienstleister zu beauftragen. Zum Beispiel, wenn das Gehäuse zu groß für Ihr Druckbett und nicht modulierbar ist. Oder

120 cm breit, HDMI eARC und Bluetooth, Schieflack weiß oder schwarz, 1.969,- € (inkl. 19% MwSt., ggf. zzgl. Versand) Nutbert electronic GmbH, Nutbertstr.1, 73529 Schwäbisch Gmünd, 30 Tage Rückgaberecht · Webshop: nutbert.de · Vorführstudios in Schwäb. Gmünd, Duisburg · Expertenberatung 0711 8712-0

Direkt + günstig vom Hersteller

### Klang und Raum neu definiert

- Für ultimatives Heimkino: Integrierte Subwoofer und 820 W Musikleistung
- Volle Konnektivität für Top-Musikgenuss aus digitalen und analogen Quellen
- Spitzenklang aus 120 cm Breite – präziser Bass mit X-Room Calibration
- Individuelle Soundmodi, beste Sprachverständlichkeit mit Voice+
- Steuerzentrale für echtes DTS:X / Dolby Atmos: Wireless zum 7.1.4-System erweiterbar
- die erste Soundbar mit „Qualitätsurteil Sehr Gut“

**Mit dem Testsieger in neue Klang-Galaxien**

nupro XS-8500 RC



**TESTSIEGER**

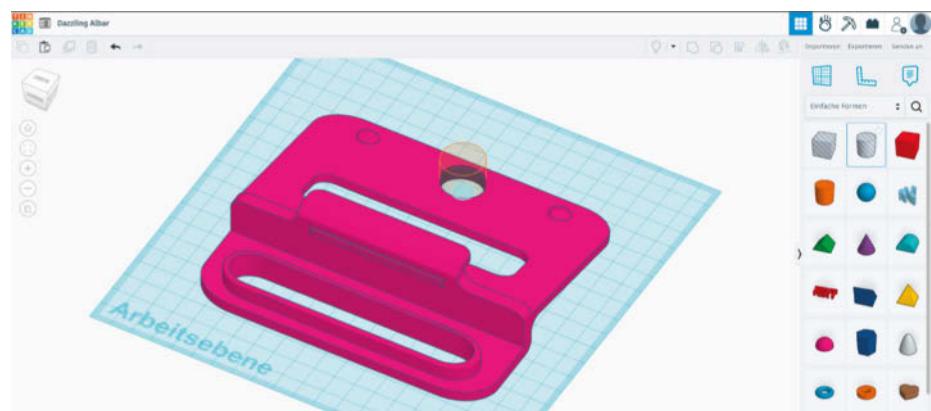


wenn es aus schlagfestem ABS-Kunststoff sein soll. Denn das anspruchsvolle Material verströmt beim Drucken daheim nicht gerade die Wohlgerüche Arabiens, und wegen der dabei entstehenden gesundheitsschädlichen Gase braucht man eine sehr gute sachgemäße Belüftung.

Für den Selbstdruck unproblematisch ist ein 3D-Druck mit einem verbreiteten FDM-Drucker, also einem, der das auf einer Spule aufgewickelte Druckfilament nach und nach erhitzt und das geschmolzene Material durch eine winzige Düse an einem seitwärts durch Schrittmotoren bewegten Druckkopf auf ein sich vorwärts und rückwärts bewegendes Druckbett fließen lässt. Im Resultat sieht man daher in der Regel Schichtlinien. Die gibt es nicht bei Druckern, die mit Photopolymerharzen arbeiten. Allerdings sind diese Drucker teuer und das Hantieren mit den Harzen macht nicht jedem Spaß.

Die FDM-Drucktechnik hat in den letzten Jahren Fortschritte gemacht, die man mit der Evolution des Autos aus den Oldtimern vergleichen kann. War in den frühen Jahren noch viel Schraubergeschick nötig, um beides am Laufen zu halten und diverse Mücken zu reparieren, so kann man heutzutage die meisten benutzen, ohne sich laufend mit technischen Problemen herumzuschlagen. Für den 3D-Druck heißt das: Druckjob hinschicken, schauen, dass die erste Schicht gut haftet, und nach der im Slicer oder auf dem Drucker selbst angegebenen Druckzeit wiederkommen und den Druck entnehmen.

Als Filamente bieten sich vornehmlich PLA und PETG an. PLA ist etwas weniger



**Mit dem gratis nutzbaren Tinkercad kann man mit einfachen Werkzeugen Gehäuse erstellen und Modelle aus anderen Quellen rudimentär bearbeiten.**

elastisch und ist bei Temperaturen ab rund 75 Grad nicht mehr formstabil, dafür beim Druck etwas gutmütiger. Vertiefendes zum 3D-Druck finden Sie in [3].

### Nachbearbeitung

Bei Gehäusen aus dem FDM-Drucker sieht man die einzelnen Druckschichten. Beim Entfernen von Stützstrukturen für Überhänge und von Rändern (Brims) verbleiben gelegentlich Reste am Objekt, die sich schlecht lösen. Solche Mängel sowie Grate kann man mit Schleifpapier (400 fürs Grobe, bis zu 4000 fürs Feine) sowie Allzweckmessern loswerden. Eleganter geht es mit Entgratwerkzeugen.

Oberflächen kann man auch mit einem Heißluftgerät glätten. Dazu muss der Druck aber sehr gleichmäßig gedreht werden und Sie müssen sehr vorsichtig sein, nicht zu viel Hitze aufzubringen. Mit

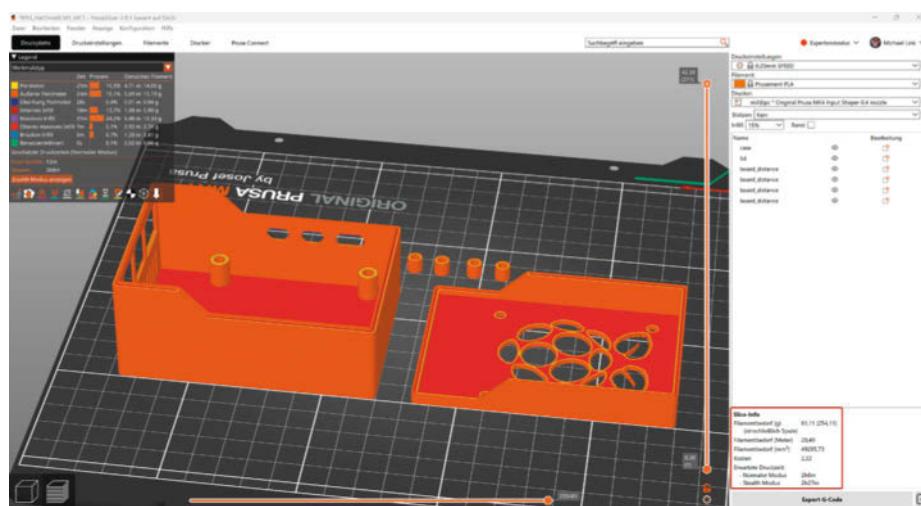
einem Föhn geht es nicht, denn der erreicht nicht die nötige Temperatur.

Statt Material abzutragen, kann man auch welches aufbringen und Farbe oder Epoxidharz aufpinseln. Damit kann man auch die von Natur aus eher nicht komplett wasserdichten Drucke nachträglich vor Wassereinbruch schützen. Bei einigen Mäkern steht auch das Eintauchen in Substanzen wie Dichlormethan hoch im Kurs, das aber nur mit adäquaten Vorsichtsmaßnahmen praktikabel ist.

### Nur Mut

Der Druck von Gehäusen ist – wie gezeigt – in allen Geschmacksrichtungen zu haben: mit einem fertigen Digitalmodell, das jemand anderes druckt, bis hin zur völligen Neukonstruktion von Grund auf, weil es nichts gibt, auf dem man aufbauen könnte. So erschließen sich unter Umständen neue Kenntnisbereiche, die man umfassend auch für andere Zwecke nutzen kann.

Wer öfter seine Gehäuse selbst baut, hat ein Hobby im Hobby gefunden. Das eigene Gehäuse ist zudem ein auffälligeres Ergebnis als Codezeilen in einem Programmierprojekt. Das mag zumindest im Familienrat helfen, für die Frickelei auch mal ein Lob zu ernten. (mil@ct.de) ct

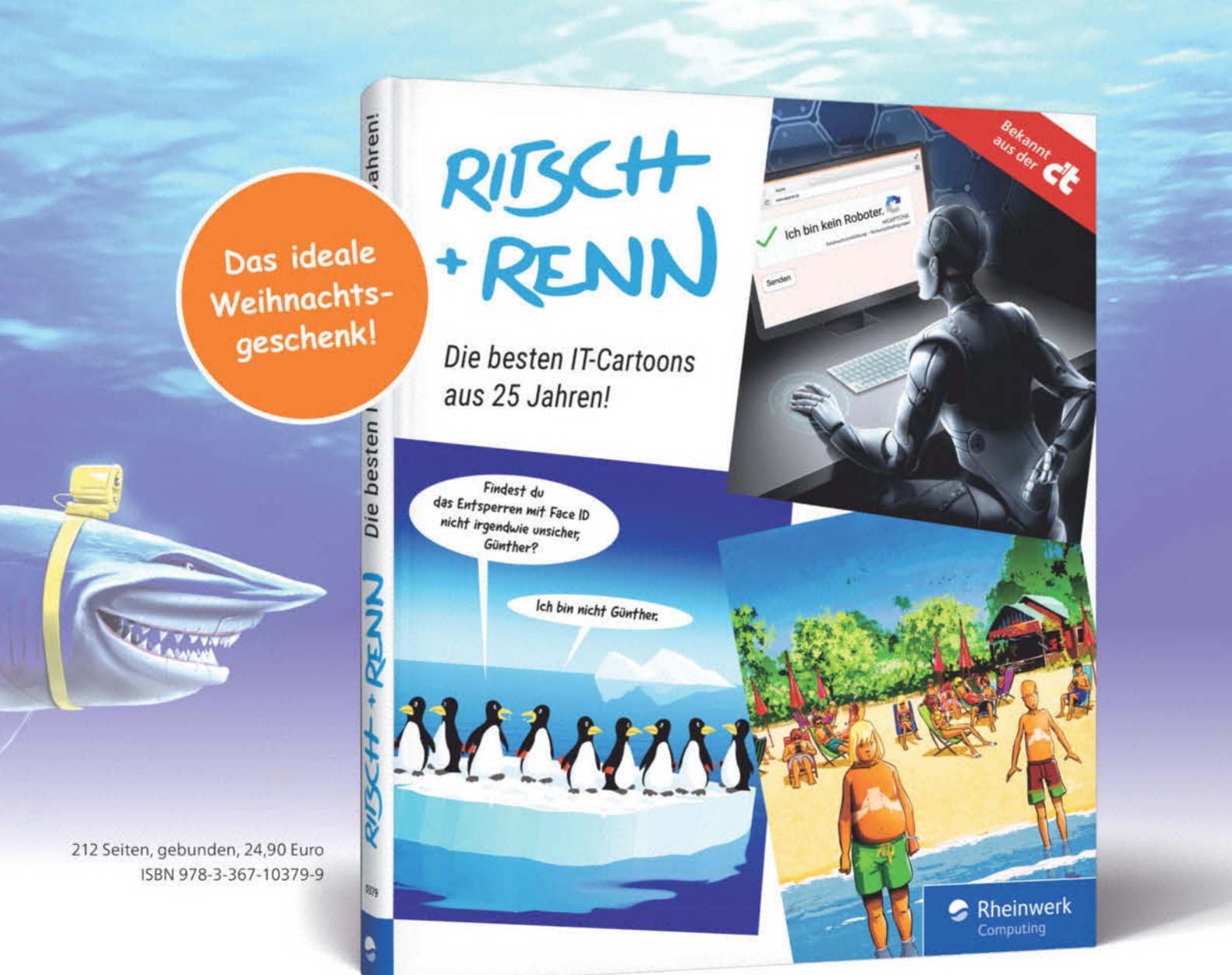


Slicer-Programme werfen nach dem Generieren der Druckdaten aus, wie lange der Druck braucht und wie viel Material dafür benötigt wird.

### Literatur

- [1] Pina Merkert, 3D-Modell selbst gezüchtet, Ersatzteile konstruiert mit Onshape, c't 4/2021, S. 146
- [2] Michael Link, Einfach mal ranklotzen, Plasticity: ein CAD-Programm für Künstler und Hobbyisten mit 3D-Drucker, c't 9/2024, S. 130
- [3] Pina Merkert, Christoph Schmith, Fabriziert nach Maß, 3D-Drucker produzieren individuelle Teile für jeden, c't 8/2023, S. 58

**3D-Modelle und -Programme:** [ct.de/ytz6](http://ct.de/ytz6)



212 Seiten, gebunden, 24,90 Euro  
ISBN 978-3-367-10379-9

## Willkommen in der Welt des gezeichneten Humors!

Aktuelle Computer-Themen und Trends, der knifflige Umgang mit Technik, der alltägliche IT-Wahnsinn: All das bilden Ritsch + Renn in ihren Cartoons humorvoll, ironisch und auf den Punkt illustriert ab. In diesem Best-of-Buch finden Sie eine Sammlung der witzigsten Cartoons aus 25 Jahren »Schlagseiten« mit exklusivem, unveröffentlichtem Material, Skizzen und Hintergrundinfos. Das perfekte Geschenk für alle, die sich für IT begeistern und das Thema mit Humor nehmen.

Weitere Geschenktipps finden Sie auf:  
[www.rheinwerk-verlag.de](http://www.rheinwerk-verlag.de)



# Raspi schön verpackt

## Fractal North Pi: Edles Raspberry-Gehäuse nachgebaut

**Der Gehäusehersteller Fractal hat das Design seines populären Gehäuses namens North geschrumpft, sodass ein Raspberry Pi hineinpasst. Damit man das Gehäuse nachdrucken kann, gab Fractal die Dateien für das Gehäuse frei. Doch unser erster Nachbauversuch lief nicht ganz glatt.**

Von Wilhelm Drehling

Auf der IT-Messe Computex im Juni dieses Jahres zeigte der schwedische Computergehäusehersteller Fractal Design ein Gehäuse für Einplatinencomputer im Stil seiner Gaming-PC-Hülle North. Das als Marketing-Gag gedachte Raspberry-Pi-Gehäuse stieß auf so viel Anklang, dass bereits kurze Zeit später erste Projekte aus der Community zum Nachbau auf 3D-Druck-Seiten wie Printables auftauchten.

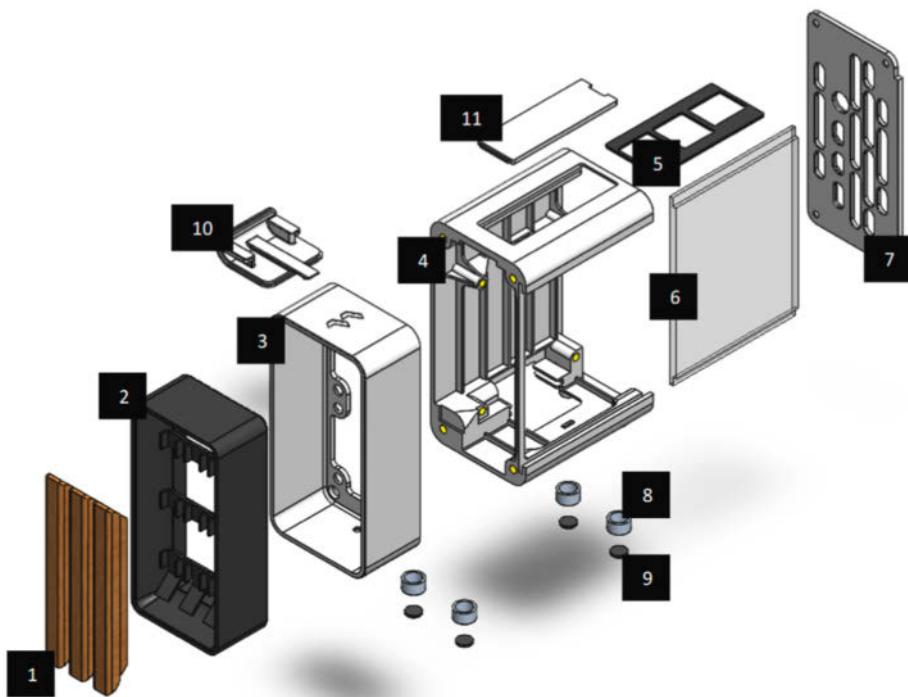
Auch wenn das Mini-Gehäuse nicht in Serienfertigung geht, so veröffentlichte Fractal alle STL-Dateien sowie eine Anleitung, um das Gehäuse selbst drucken

Frisch aus  
**ct Nerdistan**

und zusammenschrauben zu können (siehe [ct.de/y3k4](https://ct.de/y3k4)). Uns hat das sogenannte North-Pi-Case zugesagt, deswegen haben wir es kurzerhand für unseren Raspberry Pi 4 mit 8 GByte RAM nachgedruckt. Das Gehäuse soll aber nach Angaben von Fractal auch

mit dem Raspi 5 und dem Einplatinencomputer Radaxa Rock 4C Plus kompatibel sein. Bei unserem ersten Druck funktionierte nicht alles reibungslos. Lesen Sie deshalb unseren Erfahrungsbericht mit Ratschlägen und kleinen Änderungen, damit Ihr Nachbau beim ersten Versuch glückt.

Bild: Fractal



**Das Gehäuse besteht aus dem „Main-Body“ (4) und dem Vorderteil (3), in dem die Streben (1) in der Strebenhalterung (2) befestigt werden. Das I/O-Shield (5) wird genau so wie die Scheibe (6) seitlich ins Gehäuse geschoben. Die Abdeckungen (10, 11) sind für den SD-Karten-Slot und die Anschlüsse. Die Rückwand (7) liefert Fractal gleich dreimal als Datei mit: für den Raspi 4, 5 und eine DIY-Version.**

## Vorbereitungen

Sie benötigen einen 3D-Drucker und einen temperaturgesteuerten Lötkolben, um Schmelzgewinde ins Gehäuse zu drücken. Die braucht man, um das Gehäuse zusammenzuschrauben und den Raspi darin befestigen zu können. Insgesamt brauchen Sie 13 M2.5-Gewinde mit einem Außen-durchmesser von 3,5 Millimeter und genau so viele M2.5 × 6 (mm) Schrauben. Sie sollten auch etwas Sekundenkleber zur Hand haben, um ein paar der Teile zu befestigen.

Fractal liefert 13 STL-Dateien, manche, wie die Datei für die Rückseite des Gehäuses („Rear-Panel“), liegen mehrfach vor, weil sich die Anschlüsse bei Raspi 4 und 5 voneinander unterscheiden. Zusätzlich gibt es ein „DIY-Rear-Panel“, das Sie für die Anschlüsse anderer Einplatinencomputer anpassen können. Eine Übersicht aller Teile finden Sie oben in der Explosionszeichnung.

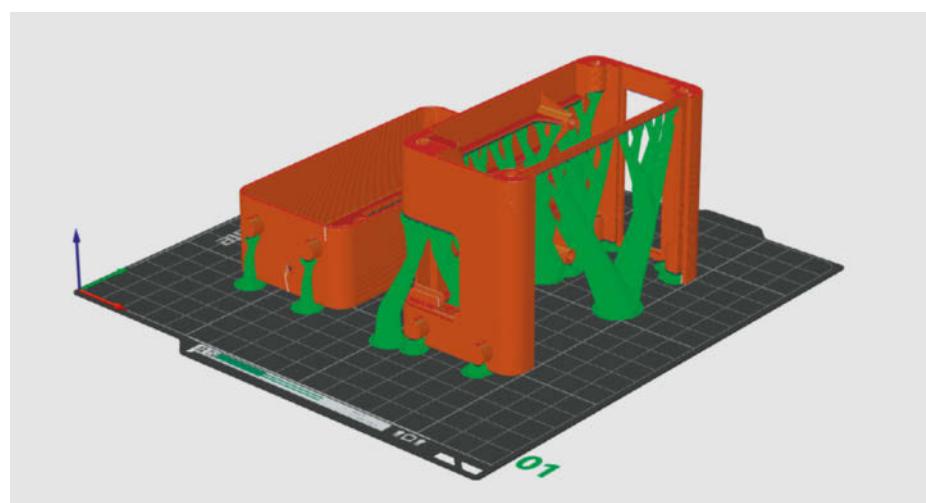
Fractal empfiehlt, das North Pi aus PLA zu drucken, die Streben aus Holz zu fertigen und die Scheibe aus Acrylglas zuzuschneiden. Der Empfehlung sind wir größtenteils gefolgt. Wir haben weißes PLA für das Gehäuse gewählt und ein graues PLA (Space Grau) als farblichen

Akzent für die Fußverkleidung und für die Strebenhalterung vorn im Gehäuse. Sie können natürlich auch andere Farben aussuchen und zum Beispiel ein dunkles Gehäuse als Grundlage verwenden. Bei den Holzstreben haben wir es uns einfach gemacht und sie aus holzfarbenem PLA gedruckt, da die Form der Streben schwierig aus Holz zu fertigen sind.

## Projektinfo

- 🎯 Edles Gehäuse für den Raspberry 4/5 selbst drucken
- 🖨️ Grundkenntnisse 3D-Druck
- ⌚ Druck: 1 Tag, Zusammenbau: 1 bis 2 Stunden
- 💰 ca. 20 Euro
- 🛒 PLA in unterschiedlichen Farben  
13 M2.5 Schmelzgewinde  
13 M2.5 × 6 (mm) Schrauben  
Acrylglasscheibe
- 🔧 3D-Drucker  
Lötkolben  
Sekundenkleber

Die Acrylglasscheibe soll laut Anleitung 3 Millimeter dick sein und an den kürzeren Außenkanten 1,5 Millimeter tief und 4 Millimeter breite Aussparungen haben, damit die Scheibe bündig ins Gehäuse geschoben werden kann. Die Form ist ohne passendes Werkzeug aber nicht so leicht zu bewerkstelligen, also haben wir uns dazu entschieden, eine 1,5 Millimeter dicke Acrylglas-Scheibe für wenige Euro bei einem Online-Zuschnittservice zu bestellen. Mit den geringfügig abgeänderten Maßen von 64 × 85 Millimeter passt die Scheibe immer noch ruckelfrei ins Gehäuse. Weil sie etwas schmäler als



**Bambu Studio zeigt in der Vorschau, welche Überhänge von den Support-Bäumen gestützt werden.**



**Vorsicht beim Einstellen der Drucktemperaturen: Wenn das Bett zu warm wird, können sich Bauteile böse verformen.**

die Vorlage ausfällt, sieht es nun so aus, als sei die Scheibe eingelassen und nicht auf gleicher Höhe wie die Seitenwand. Das fällt aber nicht störend auf.

Wenn Sie ausnahmslos alle Komponenten inklusive der Scheibe drucken wollen, können Sie es gerne mit transparentem PLA ausprobieren. Das sieht bei einem klassischen FDM-Drucker mit Düse aber eher milchig und nicht wirklich durchsichtig aus; Sie könnten allerdings mit einem Resin-Drucker Erfolg haben. Alternativ können Sie auch einfach Mesh-Gewebe verwenden oder die Scheibe in Weiß drucken und sie als eine normale Seitenwand benutzen.

## Druck und Zusammenbau

Sämtliche Teile für das Gehäuse haben wir mit einem Bambu Lab A1 Mini gedruckt. Das funktionierte problemlos, solange wir uns beim Einstellen der Temperaturen für Bett und Düse an die Empfehlungen der PLA-Hersteller hielten. Es stellte sich heraus, dass unser Space Grau ein fünf Grad kälteres Bett als das weiße PLA von einem anderen Hersteller bevorzugt, weswegen wir mit Warping zu kämpfen hatten (siehe Bild oben). Das Bett war also so warm, dass das aufliegende Plastik sich verzog. In den meisten Fällen können Sie das verhindern, wenn Sie die Temperaturen an dem 3D-Drucker oder in einem zugehörigen Programm wie Bambu Studio für Bambu-Lab-Geräte einstellen.

Das Gehäuse (gedruckt mit den Dateien „Front“ und „Main-Body“) sowie die Streben und ihre Halterungen („Front-Bar“ und „Front-Inner-Frame“) haben einen Überhang und benötigen daher Support-Material. Bis auf die Streben, für die wir den klassischen Support benutzt haben, haben wir sämtliche Stützstrukturen mit Bäumen realisiert. Diese lassen sich leichter ablösen und hinterlassen tendenziell weniger Spuren am gedruckten Objekt zurück. In Bambu Studio aktivieren Sie Support-Bäume unter dem Menüpunkt

Stützen, ähnlich klappt es bei Prusa Slicer und Software anderer Hersteller.

Nachdem alle Teile gedruckt, Stützmaterial entfernt und hervorstehende Kanten mit einem Messer sorgfältig abgetrennt sind, müssen Sie die 13 Gewinde ins Gehäuse einschmelzen. Dazu benutzen Sie am besten einen temperaturgesteuerten Lötkolben, der sich auf rund 250 Grad einstellen lässt. Das ist wichtig, denn Lötkolben erreichen üblicherweise eine viel höhere Betriebstemperatur und würden damit beim Eindrücken der Gewinde das Plastik verformen. Der „Main-Body“ bekommt zwölf Gewinde, das letzte landet unten bei der Strebenhalterung, damit die an das vordere Gehäuseteil angeschraubt werden kann.

Als Nächstes befestigen Sie die Beinverkleidungen mit Sekundenkleber, drücken die Streben vorne in die Halterung und sichern auch sie mit etwas Kleber. Achten Sie darauf, dass Sie das I/O-Shield oben im Gehäuse korrekt eingeschoben haben, sodass die Anschlüsse des Raspi bündig ins Shield passen.

Damit sich der Raspi nicht aufheizt und dabei womöglich das Gehäuse verformt, haben wir ihm noch einen Aluminium-Kühlkörper mit Wärmepads und zwei kleinen RGB-Lüftern spendiert – für

das besondere Gaming-Gefühl. Anschließend können Sie die Scheibe hineinschieben und die Rückwand und die Front mit je vier Schrauben an den Main-Body schrauben. Schließlich befestigen Sie die Strebenhalterung mit einer Schraube. Das Endresultat sehen Sie im Bild unten.

## Fazit

Mit dem Kühlkörper und den kleinen Lüftern bleibt unser Raspi 4 im Test kühl genug, um das Gehäuse nicht zu verformen. Achten Sie unbedingt auf ausreichend Luftdurchlass, denn PLA ist ein hitzeempfindliches Material. Vorn hinter den Streben ist Platz für zwei 40-Millimeter-Lüfter der Breite 10 Millimeter, die die vom Raspi erwärme Luft rauspusten können. Wer auf Nummer sicher gehen will, kann das Gehäuse aus dem beständigeren Material PETG drucken.

Fractal zielt mit seinem edel ausschenden Mini-Gehäuse auf Ästhetikliebhaber. Der Zusammenbau ist spielend leicht gewesen und die Anleitung lässt keine Fragen offen. Dank einer Vielzahl an unterschiedlichen Filamenten haben Sie freie Wahl beim Gestalten Ihres Gehäuses.

(wid@ct.de) **ct**

**Anleitung plus STL-Dateien: [ct.de/y3k4](http://ct.de/y3k4)**



**Mit Kühlkörper und RGB-Lüfter ausgestattet wird der North Pi auf dem Schreibtisch zum Blickfang.**



 heise academy

# Für erfolgreiche IT-Teams von morgen

## Weiterbildung als Erfolgsstrategie

Professionelle IT-Weiterbildung für Unternehmen – das bietet die heise academy. Als Tochter der heise group haben wir es uns zur Aufgabe gemacht, Unternehmen und ihre IT-Professionals mit digitaler Weiterbildung voranzubringen, Qualifikationslücken zu schließen und internes Lernen zu fördern.

## Ihre Zukunft, unser Plan

- Sofort einsetzbare Lernumgebung
- Verschiedene Lernformate – digital und vor Ort
- Große Auswahl an aktuellen IT-Themen
- Individuelle Lizizenzen für Ihr Team
- Onboarding & persönliche Betreuung durch unser Sales Team



**Interesse geweckt? Hier mehr erfahren:**  
[heise-academy.de/academy-pass](http://heise-academy.de/academy-pass)



# Sensorium

## Sensoren für Bastelprojekte und ihre Schnittstellen

Viele Bastelideen drehen sich um die Überwachung von Temperatur, Helligkeit, Druck, Feuchtigkeit, Gaskonzentration oder auch um Kameras und Schalter. Passende Sensoren gibt es in riesiger Auswahl, auch wenn man nicht löten möchte: Tipps zur Auswahl.

Von Christof Windeck

Wie kalt ist es gerade im Keller? Diese Frage beantwortet ein vernetzter Temperaturfühler, der sich mit einem WLAN-Mikrocontroller wie dem ESP32 in wenigen Stunden selbst basteln lässt. Aber auch für viele andere physikalische Größen wie Luftfeuchtigkeit, Luftdruck, Helligkeit, ultraviolette und infrarote Strahlung, Vibration, Beschleunigung, Drehung oder Kraft gibt es bezahlbare Sensoren. Viele lassen sich sehr einfach und sogar ohne zu löten an eines der gängigen Mikrocontroller-Platinchen oder an Einplatinencomputern wie den Raspberry Pi anschließen. Allerdings muss man zueinander passende Komponenten finden – darum geht es in diesem Artikel.

Um einen kompatiblen Sensor zu finden, muss man verstehen, wie die Kommunikation zwischen Sensor und Mikrocontroller funktioniert. Dafür gibt es nämlich einige besonders gängige Schnittstel-

len und Protokolle und für diese wiederum gute Dokumentationen, fertige Software sowie Programmierbeispiele. Wir fokussieren auf Sensoren für Mikrocontroller-Bastelboards, also etwa für Arduino und Nachbauten, ESP8266, ESP32 und Raspberry Pi Pico. Grundsätzlich funktionieren Sensoren jedoch auch an Einplatinencomputern wie dem Raspi ganz ähnlich.

### Riesige Bandbreite

Schon ein simpler Taster, ein mechanischer Neigungsschalter oder ein magnetisch betätigter Reed-Kontakt können als Sensoren dienen, beispielsweise wenn sie in einem mechanischen Gerät so eingebaut sind, dass sie die Position eines beweglichen Teils an den steuernden Chip zurückmelden. Das ist etwa auch das Funktionsprinzip bei Tür- und Fensterkontakten für Alarmanlagen und bei Schwimmerschaltern, die in Flüssigkeitsbehältern einen Pegelgrenzwert melden.

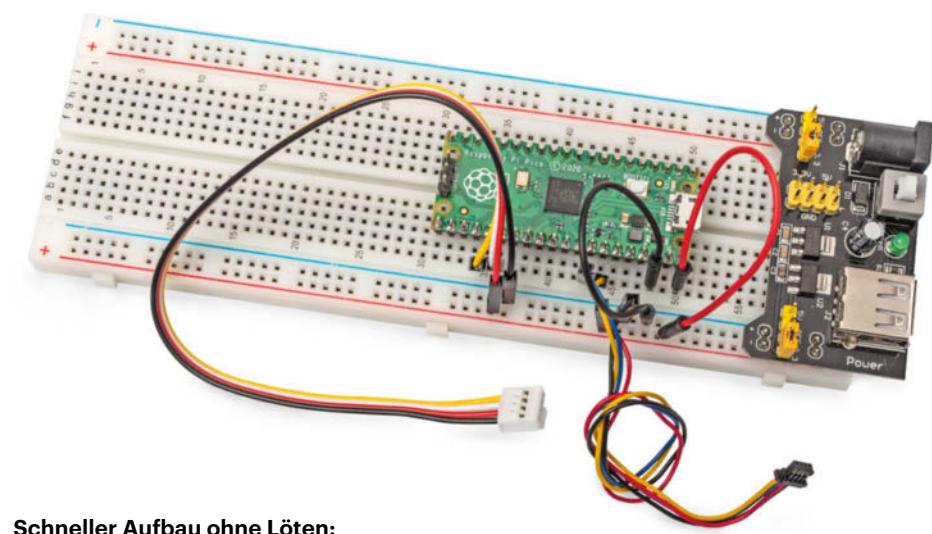
## c't kompakt

- Es gibt eine gigantische Auswahl an bezahlbaren Sensoren für Bastelprojekte.
- Bastler benötigen Code-Libraries für die jeweilige Kombination aus Sensor, Schnittstelle und Bastelboard.
- Ein Sensor mit Steckanschluss vermeidet nicht nur Lötrei, sondern meistens gibt es dafür auch Programmierbeispiele und weitergehenden Support.

Um dieses Signal zu erfassen, braucht das Controller-Board einen digitalen Eingang (Input). Und schon diese scheinbar simple Funktion kann auf sehr unterschiedliche Art umgesetzt sein. Das beginnt mit dem zulässigen Spannungsbereich. Viele Mikrocontroller und der Raspberry Pi sind für Logikpegel von 3,3 Volt ausgelegt. Es gibt aber auch welche, die 5 Volt verlangen – und einige, die mit beiden Pegeln umgehen können. Passen die Spannungsniveaus nicht zueinander, helfen sogenannte Level Shifter aus wenigen elektronischen Bauteilen.

Die nächste Frage ist, wie der Taster oder Schalter den Eingang ansteuert. Schließt er den Stromkreis gegen Masse (0 Volt, Ground) oder gegen den Logikpegel (3,3 oder 5 Volt)? Je nachdem muss der Eingang dann passend mit Widerständen beschaltet sein, damit keine zu starken Ströme fließen. Manche Mikrocontroller haben per Softwarebefehl schaltbare, interne Widerstände an den Eingangspins: Ein Pull-up-Widerstand setzt den Kontakt auf den Logikpegel, ein Pull-down-Widerstand sorgt im Ruhezustand für 0 Volt.

Die meisten Bastelplattformen haben flexibel per Software konfigurierbare Ein-



**Schneller Aufbau ohne Löten:**  
Mit einem Steckbrett (Breadboard) geht die Bastelei flott von der Hand.

und Ausgabekontakte, genannt General Purpose I/O (GPIO). Die meisten kann man nicht nur von Ein- auf Ausgabe umschalten, sondern wie erwähnt oft auch Pull-up- oder Pull-down-Widerstände aktivieren. Außerdem stecken in den meisten Chips mit GPIO noch zahlreiche weitere Funktionen und Controller, die sich über per Software steuerbare, interne Umschalter (Multiplexer) mit den einzelnen externen Kontakten verbinden lassen. Ein GPIO-Pin hat also je nach laufender Software ganz unterschiedliche Funktionen.

Üblich ist mindestens ein Analog-Digital-Wandler (ADC), der eine anliegende Spannung in digitale Messwerte umsetzt. Manche Chips enthalten auch Digital-Analog-Wandler (DAC), um analoge Signale auszugeben. Üblich sind zudem spezialisierte Schaltungen zum Erzeugen eines pulsweitenmodulierten Logiksignals mit einstellbarem Tastverhältnis; eine solche PWM steuert beispielsweise Schrittmotoren an. Um PWM-Signale erzeugen und auswerten zu können, stecken in Mikrocontrollern auch Taktgeneratoren, Timer und Zähler. Die sind auch für Interfaces nötig, die mit einem Taktsignal (Clock) arbeiten.

Für den Anschluss digitaler Sensoren wichtig sind Standardschnittstellen wie I<sup>2</sup>C, I<sup>2</sup>S, SPI oder auch USB, auf die wir unten genauer eingehen. Auch dafür lassen sich GPIO-Pins konfigurieren, aber es müssen dann oft mehrere sein, falls das jeweilige Interface beispielsweise ein separates Taktsignal benutzt oder mehrere Datenleitungen.

Alle gängigen Programmierumgebungen für Bastler-Mikrocontroller stellen Funktionen für die GPIO-Einstellung bereit. Die möglichen Beschaltungsoptionen sind in den Konfigurationsdateien für das jeweilige Bastelboard hinterlegt. Daher ist es beispielsweise in der Arduino-IDE wichtig, im Board-Manager das genau passende Board auszuwählen. Denn je nach Board sind die extern nutzbaren GPIO-Pins mit unterschiedlichen Kontakten des Chips verbunden.

### Digital ist besser

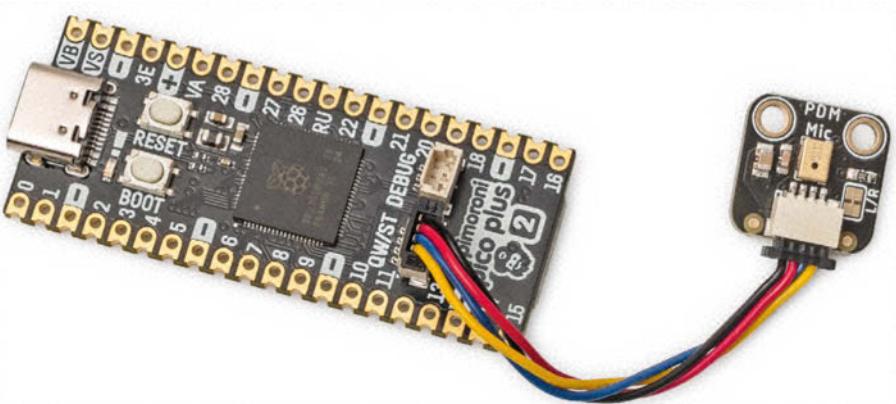
Alle gängigen Bastelboards können bereits ohne externe Sensoren einige physikalische Größen messen, denn ihr ADC digitalisiert anliegende Spannungswerte. Über einen einfachen Messwiderstand (Shunt) lässt sich im Prinzip dann auch

## Sensor-Steckanschlüsse für Bastelboards (Auswahl)

Name	Entwicklerfirma	Protokoll	Betriebsspannung	Pin 1	Pin 2	Pin 3	Pin 4
Grove	Seeed Studio	I <sup>2</sup> C, analog, weitere	(Vcc) 3,3 oder 5 Volt	gelb / bei I <sup>2</sup> C: SCL	weiß / bei I <sup>2</sup> C: SDA	rot / Vcc	schwarz / Masse
Qwiic Connect	Sparkfun	I <sup>2</sup> C	3,3 Volt	gelb / I <sup>2</sup> C SCL	blau / I <sup>2</sup> C SDA	rot / 3,3 Volt	schwarz / Masse
Stemma QT <sup>1</sup>	Adafruit	I <sup>2</sup> C	3,3 oder 5 Volt	gelb / I <sup>2</sup> C SCL	blau / I <sup>2</sup> C SDA	rot / Vcc	schwarz / Masse
Stemma	Adafruit	I <sup>2</sup> C	3,3 oder 5 Volt	grün / I <sup>2</sup> C SCL	weiß / I <sup>2</sup> C SDA	rot / Vcc	schwarz / Masse
Stemma 3-Pin	Adafruit	Analog, weitere	3,3 oder 5 Volt	–	weiß / Signal	rot / Vcc	schwarz / Masse

maximale Kabellänge bei I<sup>2</sup>C: bis zu 1 Meter; es gibt Umsetzer für differenzielle Übertragung bis 30 Meter

<sup>1</sup> Stecker von Stemma QT zu Qwiic kompatibel



### Mikrocontroller-Boards mit Sensor-Steckverbindern wie Grove, Qwiic oder Stemma erleichtern den Einstieg, weil man weniger Fehler machen kann.

die Stärke des durchfließenden Stroms messen.

Allerdings muss man dabei einige Probleme lösen. So ist etwa der zulässige Spannungsbereich der ADCs relativ klein, beim Raspi RP2040 reicht er etwa von 0 bis 1,4 Volt. Um höhere Spannungen zu messen, muss man diese also erst passend umsetzen. Zudem kann eine Potenzialtrennung nötig sein zwischen dem Mikrocontroller und der zu messenden Spannung. Schließlich misst der RP2040-ADC erst dann mit akzeptabler Genauigkeit, wenn man eine stabilisierte Referenzspannung einspeist. Um also den internen ADC für sinnvolle Messungen verwenden zu können, müsste man relativ komplizierte Zusatzschaltungen entwerfen.

Viel einfacher ist es daher, einen Sensor mit digitalem Ausgang an einen digitalen Eingang des Mikrocontrollers anzuschließen. Es gibt eine riesige Auswahl solcher Sensoren, viele kosten weniger als einen Euro. Einige Sensoren sind hochintegrierte Messsysteme, die mehrere physikalische Größen in bestimmten Wertebereichen messen.

Sensoren, die digitale Daten liefern, haben für Bastler große Vorteile. Man braucht etwa keine komplexe und fehlerträchtige Eingangsschaltung für die präzise Aufbereitung des Signals und seine Digitalisierung. Im Vergleich zu analogen Messdaten gibt es auch weniger Unsicherheiten bei der Interpretation der ausgelesenen Werte, und die Genauigkeit lässt sich gut einschätzen.

Bei vielen analogen Sensoren ändert sich der elektrische Wert am Ausgang nicht linear mit der Änderung der gemessenen Größe. Es gibt also eine Kennlinie, die man nach dem Auslesen der digitalen Rohdaten aus dem Analog-Digital-Wandler (ADC)

zunächst noch einrechnen muss. Außerdem kann die Signalspannung außer von der Messgröße noch von anderen schwankenden Faktoren abhängen, etwa der Versorgungsspannung des Sensors oder dessen Temperatur. Um digitalisierte analoge Sensordaten sinnvoll weiterverarbeiten zu können, muss man oft noch Kompensationsfaktoren einbeziehen. Das ist bei manchen digitalen Sensoren zwar genauso. Oft sind dann jedoch die nötigen Kompensationsfunktionen bereits in der Ansteuerungssoftware hinterlegt.

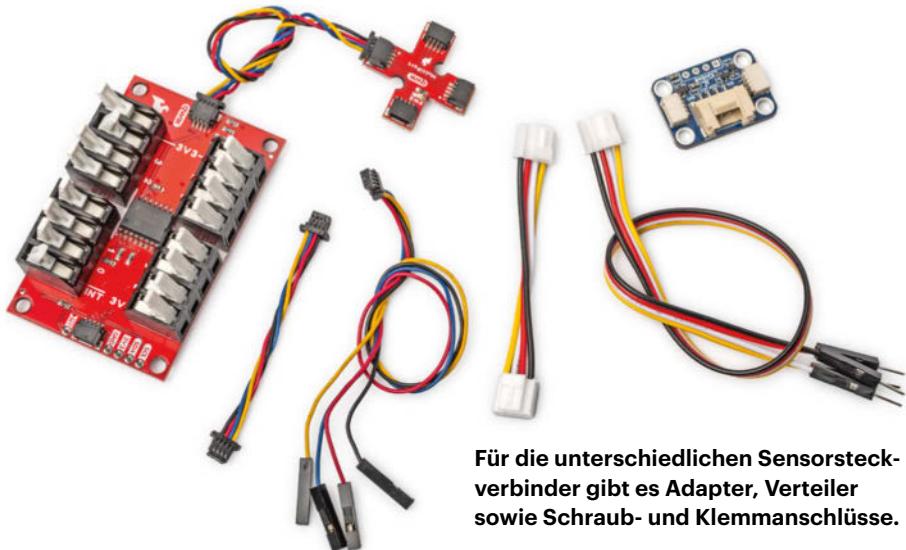
Allerdings muss der für die Weiterverarbeitung der Messdaten vorgesehene Mikrocontroller (oder Einplatinencomputer) die zum Sensor passende digitale Schnittstelle haben. Außerdem müssen die Versorgungs- und Signalspannungen von Bastelboard und Sensor harmonieren. Der Wertebereich des Sensors muss ausreichend groß sein, um die gewünschte Größe messen zu können, die Messung muss genau genug sein und bei dynamisch veränderlichen Größen auch schnell

genug. Wenn es um hohe Messfrequenzen geht, muss zudem die Schnittstelle zwischen Sensor und Board eine ausreichend hohe Datentransferrate besitzen.

Schließlich braucht man auch Software beziehungsweise Code, und zwar mehrere Komponenten. Die Programmierumgebung (Integrated Development Environment, IDE) muss zum gewünschten Bastelboard kompatibel sein. Gehen wir der Einfachheit halber mal von der beliebten Arduino-IDE aus: Dann muss das jeweilige Board im Board Manager auftauchen oder sich manuell hinzufügen lassen. Dann sind auch Codebibliotheken (Libraries) zu finden, um die GPIOs passend zur Sensorschnittstelle konfigurieren zu können. Zusätzlich braucht man eine Programmierbibliothek für das jeweils gewünschte Protokoll, als Beispiel sei hier der gängige I<sup>2</sup>C-Bus erwähnt. Dann ist aber auch eine Library für den jeweiligen Sensor nötig, in der beispielsweise eingearbeitet ist, in welchen Datenformaten er seine Messwerte anliefert oder ob und wie er sich konfigurieren lässt. Wie Sie diese Sensordaten weiterverarbeiten und beispielsweise anzeigen können, beschreibt der Artikel auf Seite 58.

### Steckfertig

Wenn Sie sich bei einem Bastelprojekt mit Sensoren nicht zu sehr mit Details belasten wollen, nehmen Sie am besten Sensoren mit Steckanschlüssen. Es gibt zwar keinen herstellerübergreifenden Standard dafür, aber eine große Auswahl von Firmen wie Adafruit (Stemma/QT), Seeed Studio (Grove) und SparkFun (Qwiic). Sie müssen dazu nicht einmal ein Board mit passender Steckbuchse kaufen, weil es zahlreiche Adapter gibt, unter anderem



Für die unterschiedlichen Sensorsteckverbinder gibt es Adapter, Verteiler sowie Schraub- und Klemmanschlüsse.

auch für Steckbretter (Breadboards). Wenn Sie ein Bastelboard mit aufgelöten Pfostensteckern nehmen, brauchen Sie nicht einmal das: Es gibt die Steckadapter auch mit einzelnen Kontakthülsen zum Aufstecken.

Ein Sensor mit Steckanschluss macht nicht bloß Lötarbeiten überflüssig: Für die meisten dieser Sensoren gibt es auch alle nötigen Libraries, Codebeispiele und Anleitungen. Denn das stellen die jeweiligen Hersteller der Stecksysteme typischerweise bereit.

Die Idee zu einem billigen, flexiblen Sensorstecksystem setzte wohl zuerst die chinesische Firma Seeed Technology aus Shenzhen um. Seeed brachte um 2014 Arduino-kompatible Boards mit vierpoligen „Grove“-Buchsen heraus sowie ein Grove Base Shield für den Arduino Uno. Die Grove-Buchsen mit Pins im 2-Millimeter-Raster sind relativ groß. Verwirrend ist, dass das Grove-Stecksystem kein digitales Protokoll festlegt, sondern etwa auch simple Ein-/Ausgabesignale oder analoge Signale führen kann.

Die Firma Adafruit folgte mit Stemma-Steckverbindern, die mit einfachen Kabeladapters zu Grove kompatibel sein können. Adafruit änderte aber zwei wesentliche Aspekte: Stemma verwendet billige Standardsteckverbinder der japanischen Marke JST und ist in der vierpoligen Version auf das Protokoll I<sup>2</sup>C festgelegt. Für analoge Signale oder Schalter gibt es eine dreipolige Version von Stemma.

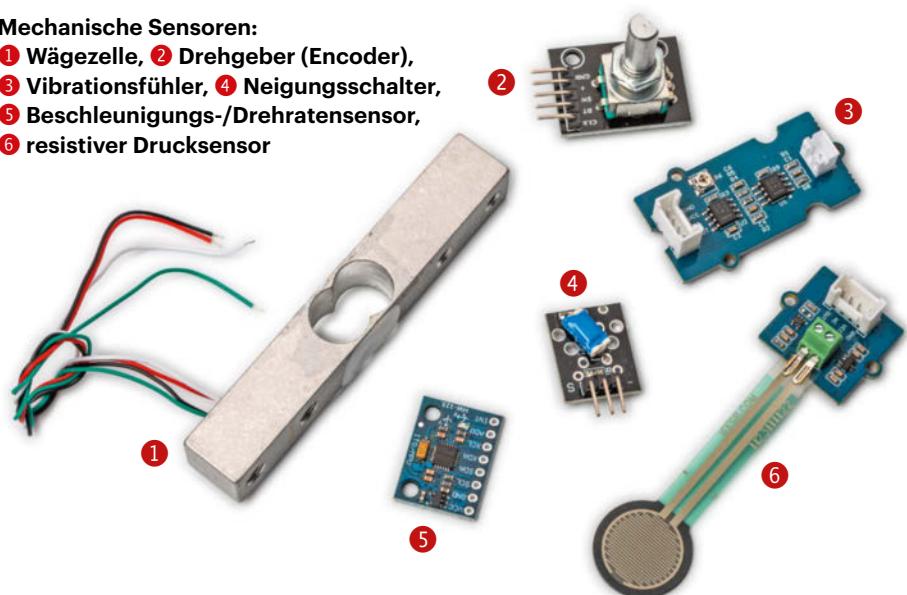
Stemma-QT ist eine kompakte Variante des vierpoligen Stemma, die Steckverbinder mit 1-Millimeter-Raster verwendet. Damit wiederum ist Stemma-QT nicht nur elektrisch, sondern auch mechanisch kompatibel zu Qwiic Connect von SparkFun. Die Buchstaben „QT“ sind eine Verballhornung des englischen Wortes „cute“, was so viel bedeutet wie „niedlich“. Die Eigenschaften der gängigsten Sensorstecksysteme sowie die verwendeten Kabelfarben zeigt die Tabelle auf Seite 51.

Sofern ein Sensor mit Grove, Stemma(-QT) oder Qwiic tatsächlich I<sup>2</sup>C nutzt, lässt er sich zumindest per Adapterkabel auch an ein anderes System anschließen. Eine Hürde kann dabei allerdings die Spannung der Versorgungs- beziehungsweise Signalleitungen sein, wie oben bereits erwähnt. Adafruit stattet viele Sensoren mit Level Shiftern aus.

Außer Adapterkabeln für die drei verbreiteten Steckersysteme gibt es auch wel-

#### Mechanische Sensoren:

- ① Wägezelle,
- ② Drehgeber (Encoder),
- ③ Vibrationsfühler,
- ④ Neigungsschalter,
- ⑤ Beschleunigungs-/Drehratensor,
- ⑥ resistiver Drucksensor



che, die am anderen Ende Schraub- oder Hebelklemmen für abisolierte Adern haben. Dabei müssen Sie aber wie beim Steckbrett ganz genau auf die richtige Belegung achten.

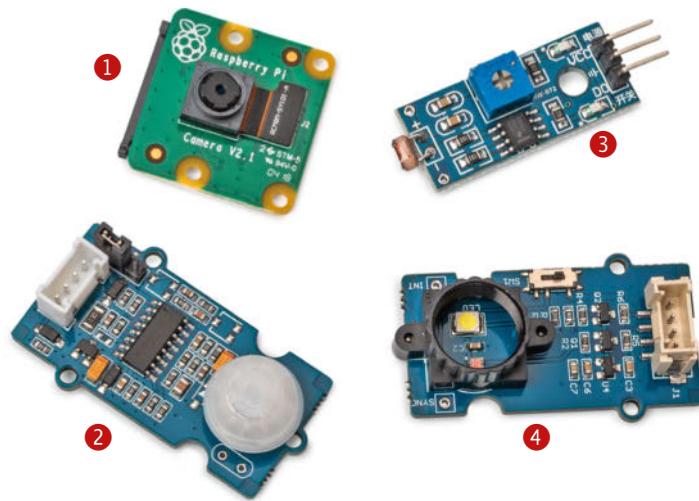
#### Sensor-Esperanto I<sup>2</sup>C

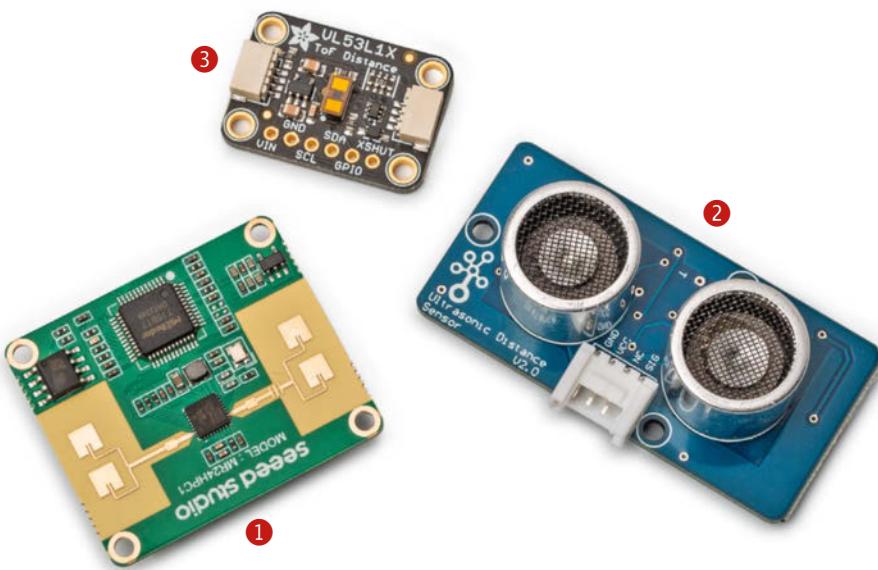
Der serielle Datenbus I<sup>2</sup>C – gesprochen „I Quadrat C“ oder auch „I Zwei C“ – spielt bei Mikrocontrollern und Sensoren eine wichtige Rolle. Der auch als I2C oder IIC abgekürzte Standard hat mehr als vierzig Jahre auf dem Buckel. Die Abkürzung I<sup>2</sup>C steht für Inter-Integrated Circuit, was andeutet, dass der Bus zur Kommunikation zwischen einzelnen Chips gedacht ist – also nicht als externe Schnittstelle. I<sup>2</sup>C verwendet nur zwei Leitungen, außer der für Daten (Serial Data, SDA) noch eine für das Taktsignal (Serial Clock, SCL). An einem I<sup>2</sup>C-Bus können über 100 Sensoren hängen, weshalb es auch Verteilerplatinchen für die oben erwähnten Stecksysteme gibt.

I<sup>2</sup>C stellt keine besonders hohen Datenträfferraten bereit, in der einfachsten Version sind es nur 100.000 Bits pro Sekunde (100 kbit/s, Standard Mode, „S“). Es gibt aber auch die schnelleren Transfermodi Fast (F) und High Speed (Hs) mit 0,4 respektive 3,4 Mbit/s – was geht, hängt von Mikrocontroller und Sensor ab. I<sup>2</sup>C ist zudem nicht für längere Kabelstrecken ausgelegt. Je nach Beschaltung und elektromagnetischen Störungen kommt I<sup>2</sup>C oft nicht weiter als 1 Meter. Es gibt allerdings Umsetzerbausteine für Differential I<sup>2</sup>C, das längere Kabelstrecken überwindet.

Noch billiger als mit I<sup>2</sup>C geht es mit nur einer einzigen Datenleitung, etwa „1-Wire“ (Single Wire, One Wire Bus/OWB), das ursprünglich die US-Firma Dallas Semiconductor entwickelte. Beispielsweise nutzen die kombinierten Temperatur- und Luftfeuchtesensoren DHT11 und DHT21 SingleWire, die Version DHT22 hingegen I<sup>2</sup>C.

- Die Datenmassen einer Kamera ① überfordern viele Mikrocontroller. Der PIR-Sensor ② erfasst infrarote Wärmestrahlung etwa von Menschen. Die Fotozelle ③ misst die Helligkeit sichtbaren Lichts, der Farbsensor ④ dessen Farbe.**





**Der Radarsensor ①** wertet seine Hochfrequenzsignale selbst aus und liefert die digitale Information „Person im Raum“. **② Ultraschall-** und **③ Time-of-Flight-Sensor** messen Distanzen.

Einige Sensoren verwenden wiederum das Serial Peripheral Interface (SPI). Es benötigt allerdings mindestens vier Leitungen, also doppelt so viele wie I<sup>2</sup>C, und eignet sich damit nicht für die erwähnten Steckverbindersysteme mit vier Pins, von denen zwei mit Versorgungsspannung und Masse belegt sind.

### Multimedia-Schnittstellen

Der Transport digitaler Ton- und Videosignale muss besondere Randbedingungen einhalten: Erstens müssen die Daten ohne Verzögerungen und Aussetzer fließen (isochron), zweitens können sehr hohe Datentransferraten nötig werden und drittens müssen Ton- und Bildspuren zueinander synchron bleiben. Daher gibt es spezielle Schnittstellen und -erweiterungen für Multimediadaten.

Auf Audiodaten zielt I<sup>S</sup> (Inter-IC Sound), das wie I<sup>2</sup>C seriell arbeitet, aber ein zusätzliches Taktsignal zur Synchronisation führt. Inklusive Spannungsversorgung sind folglich fünf Kontakte nötig.

Manche digitalen Mikrofone verwenden Pulsdichtemodulation (PDM). Oft handelt es sich um Mikromechanische Systeme (MEMS) mit direkt angeschlossener Logikschaltung. Letztere verwandelt die vom Mikrofon erfassten Luftschwingungen in eine Impulsfolge, und zwar synchron zu einem Taktsignal, welches der Mikrocontroller an das Mikrofon sendet.

Im Prinzip lassen sich die ADC- und DAC-Funktionen von Mikrocontrollern auch zur Ein- und Ausgabe von Audio nutzen.

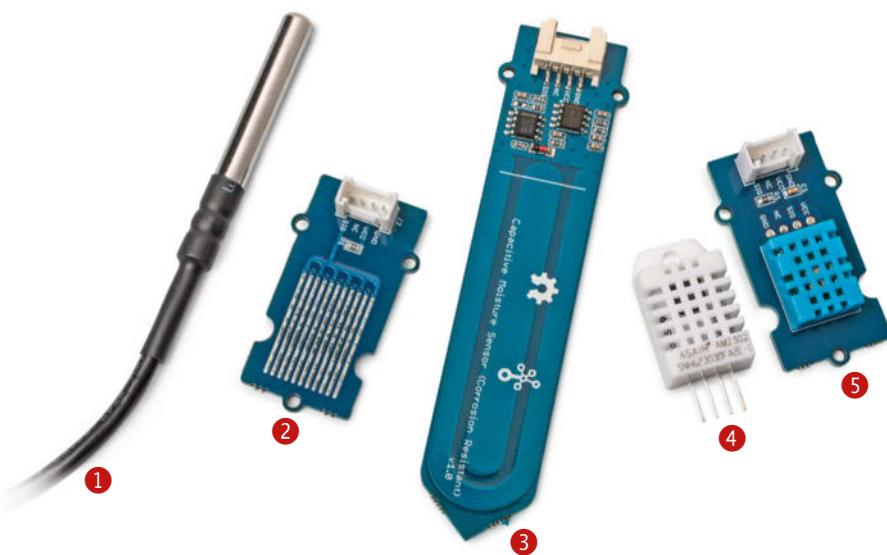
zen, aber die Qualität ist mäßig, und für Stereo braucht man zwei synchrone Kanäle. Besser ist ein I<sup>S</sup>-Soundchip, in dem für Audiosignale optimierte Wandler sitzen. Manche enthalten auch einen (Klasse-D-)Verstärker, um Lautsprecher anzusteuern.

Vergleichsweise gigantische Datenmassen erzeugen Bildsensoren, also Kameraschips. Ein Full-HD-Bild hat 1920 × 1080 Bildpunkte, also insgesamt 2 Millionen (2 Megapixel). Bei je 8 Bit (1 Byte) Auflösung für Rot, Grün und Blau (RGB), also 3 Byte pro Pixel, belegt ein unkomprimiertes 2-MPixel-Bild 6 MByte. Um pro Sekunde 30 solcher Bilder (30 fps) zu

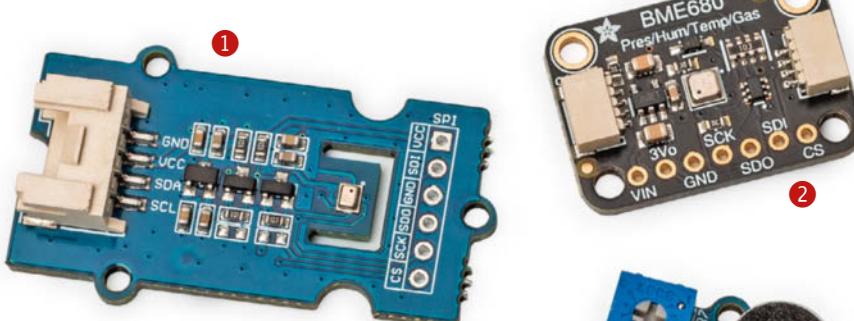
übertragen, sind 180 MByte/s nötig, also rund 1,5 Gigabit/s. Das übersteigt die Fähigkeiten von I<sup>2</sup>C und SPI bei Weitem. Aber auch der interne SRAM-Speicher typischer Mikrocontroller ist mit weniger als 550 Kilobyte (KByte) viel zu klein für auch nur ein einziges Bild. Die beliebte ESP32-Kamera [1] setzt daher ein ESP32-Board mit zusätzlichem PSRAM-Speicherchip voraus. Für Video nimmt man besser einen Einplatinencomputer wie den Raspi mit mindestens 512 MByte RAM. Zum Anschluss eines Kamera-Moduls hält der Raspberry Pi ein Camera Serial Interface (CSI) nach Spezifikationen der Mobile Industry Processor Interface Alliance (MIPI) bereit. MIPI CSI überträgt pro Lane mehr als 1 Gbit/s.

Einplatinencomputer enthalten auch einen USB-Host; Mikrocontroller-Boards mit USB-Buchse hingegen arbeiten meistens nur als USB-Device, melden sich also wie ein Peripheriegerät oder als USB-Massenspeicher an einem Host an. Außerdem beziehen sie Strom vom Host.

Einige Mikrocontroller wie Raspberry Pi RP2040 und RP2350 können auch einen USB-Host bereitstellen. Dann kann man im Prinzip auch USB-Sensoren anschließen. Doch Vorsicht: Man muss dann auch die 5-Volt-Versorgung für USB-Geräte bereitstellen. Und anders als bei einem Desktop-PC-Betriebssystem wie Linux oder Windows steht bei einem Mikrocontroller nicht automatisch die riesige USB-Treiberinfrastruktur bereit. Stattdessen muss man die USB-Unterstützung im Programmcode selbst zusammenbauen



Im Messfühler **①** steckt der 1-Wire-Temperatursensor DS18B20. Für Wasserstand **②** und Bodenfeuchtigkeit **③** gibt es simple analoge Platinenfühler. DHT22 **④** und DHT11 **⑤** erfassen Lufttemperatur und Luftfeuchtigkeit.



Der Bosch BMP280 ① misst Temperatur und Luftdruck, der BME680 ② zusätzlich Feuchtigkeit sowie die Konzentration bestimmter Kohlenwasserstoffe.

Das PDM-Mikrofon ③ verwandelt Schall in eine Pulsdichtemodulation, der Lärmföhler ④ schaltet bei Krach.

und auch für den jeweils gewünschten USB-Sensor alle nötigen Libraries finden oder eben selbst erstellen.

### Industrietechnik

Manchmal hat man nicht die Wahl, einen zum Board passenden Sensor zu beschaffen, sondern sucht eine Möglichkeit, einen vorhandenen Messfühler auszuwerten. Für manche etablierten Industriestandards wie RS-232 (COM-Port), RS-485, CAN-Bus, die gängige Analogschnittstelle 0 bis 10 Volt sowie die Stromschleife mit 4 bis 20 Milliampere finden sich Adapterboards.

Viele Mikrocontroller enthalten einen oder mehrere Universal Asynchronous Receiver-Transmitter (UART), deren RX- und TX-Leitungen sich auf GPIO-Pins schalten lassen. Dann ist für RS-232 oder RS-485 nur noch ein relativ einfach bestücktes Board mit einem Transceiverchip und einer DB9-Buchse oder Schraubklemmen nötig. Beim CAN-Bus ist es komplizierter, falls der Mikrocontroller keinen integrierten Controller dafür hat; bei

RP2040 und RP2350 lässt sich CAN-Bus über die PIO-Zustandsmaschinen anflanschen – plus Transceiverchip, ähnlich wie bei RS-232.

Für 4 ... 20 mA finden sich fertige Umsetzerplatinen, die das Signal digitalisieren und per I<sup>2</sup>C ausgeben. 0 ... 10 Volt lassen sich über einen Widerstandsteiler am GPIO per internem ADC erfassen. Ist das zu ungenau, nimmt man einen separaten ADC-Chip wie den TI ADS1113 mit eingebauter Referenzspannungsquelle und I<sup>2</sup>C.

### Gut genug?

Vor der Auswahl eines Sensors sollte man genau überlegen, welche Eigenschaften er für das jeweilige Projekt mitbringen muss. Das geht beim Messbereich los, wie sich am Beispiel Temperaturmessung zeigen lässt. Die bereits erwähnten Luftfeuchtigkeits- und Lufttemperatursensoren DHT11/12/21/22 wurden für die Heizungs- und Klimatisierungstechnik (Heating, Venting, Air Conditioning/HVAC) entwickelt. Einige Varianten sind sogar ab

Der optische Sensor links erfasst Staub in der Luft, rechts ein Durchflussmesser. In der Mitte ein Magnetfeldsensor.



# Rote Karte

für ungebetene Kommunikation



Die W&T-Microwall ist die whitelist-basierte Firewall für die Hutschiene.

Der neue Discover-Mode scannt jede einzelne Kommunikationsanfrage. Mit wenigen Klicks werden gewünschte Verbindungen auf die Whitelist übernommen - und alle anderen vom Platz geschickt.

Netzwerke sicherer machen:

[wut.de/discover](http://wut.de/discover)



**W&T**  
[www.WuT.de](http://www.WuT.de)

Werk kalibriert, manche aber nur für Innenräume gedacht. Sie messen keine negativen Temperaturen und maximal bis +50 Grad Celsius. Andere kommen auf höchstens 80 Grad und eignen sich folglich beispielsweise nicht als Saunathermometer. Typische Messintervalle bei solchen Sensoren dauern mehrere Sekunden – schneller verändern sich Temperatur und Luftfeuchtigkeit in Räumen selten. Und die Genauigkeit liegt auch nur bei  $\pm 0,5$  Grad, als Fieberthermometer sind sie also ungeeignet, ganz abgesehen von ihrer Bauform. Schließlich empfiehlt der Hersteller auch noch eine Anschlusskabellänge von höchstens rund einem Meter bei Versorgung mit 3,3 Volt; mit 5 Volt sollen bis zu 30 Meter möglich sein.

Vergleichbare Einschränkungen gelten bei fast allen Sensoren. Im Blick behalten muss man auch externe Störungen.

So schwankt die Messgenauigkeit bei manchen Sensoren je nach Temperatur. Auch Feuchtigkeit, Schmutz, Staub, Vibrationen, helles Sonnenlicht und elektromagnetische Einstreuungen von anderen Geräten können Messungen stören.

Nervtötend ist die Fehlersuche, wenn Schaltungsteile einander beeinflussen. So ist es zwar möglich, mehrere Shields an denselben Arduino anzukoppeln, aber manche stören sich gegenseitig. Empfindliche Sensoren kommen aus dem Tritt, wenn starke WLAN-Sender in der Nähe arbeiten. Umgekehrt zeigte sich, dass Radar-Bewegungssensoren, die mit 5,8 GHz arbeiten, das WLAN-Netz im 5-GHz-Band beeinträchtigen.

## Fazit

Mikrocontroller-Bastelboards und Einplatinencomputer ermöglichen es, clevere

Messtechnik für wenig Geld selbst zu bauen. Dafür steht eine riesige Fülle digitaler Sensoren zur Auswahl, die verlässliche Messdaten über gängige Schnittstellen anliefern. Man muss allerdings für das jeweilige Projekt einen bezahlbaren Sensor finden, für den es bereits alle nötigen Codebibliotheken und möglichst auch Beispielprojekte gibt. Denn viele Sensoren haben spezielle Eigenheiten, an denen man sich die Zähne ausbeißt, wenn man die Auswertesoftware im Alleingang programmieren möchte. Viel einfacher und schneller gelingt das eigene Projekt, wenn man sich dabei auf die Vorarbeit anderer stützt.

(ciw@ct.de) 

## Literatur

- [1] Peter Siering, Little Brother, Günstige WLAN-Kamera auf ESP32-Basis, c't 6/2023, S. 30

# Sensoren für Mikrocontroller-Bastelboards und Einplatinencomputer (Auswahl)

Typ/Chip	Messgröße/Funktion	Schnittstelle
<b>Schalter, Encoder, Lichtschranke</b>		
Taster, Schalter	schaltet bei Druckbetätigung	digitaler Ausgang
Neigungsschalter	schaltet bei bestimmtem Neigungswinkel	digitaler Ausgang
Reed-Kontakt (Fenster-/Türkontakt)	Magnetfeld schließt Kontakt	digitaler Ausgang
Dreh-Encoder	erzeugt Bitmuster je nach Drehrichtung	zwei digitale Ausgänge
Gabellichtschranke	Drehzahl (Impulse pro Minute)	digitaler Ausgang
<b>Beschleunigung, Drehrate, Vibration, Kraft</b>		
Beschleunigungssensor (Accelerometer)	Beschleunigung, meistens 3 Achsen	I <sup>2</sup> C
Drehratensor (Gyroskop)	Rotationsgeschwindigkeit, typ. 3 Achsen	I <sup>2</sup> C
Vibrationssensor	mechanische Vibration	I <sup>2</sup> C oder digital (Grenzwert)
Biegesensor (Wägezelle) mit HX711	Kraft, HX711 ist Wägezellen-ADC	I <sup>2</sup> C
resistiver Kraftsensor	verändert Widerstand je nach Druckkraft	analog
<b>Distanz/Anwesenheit</b>		
Ultraschall	Laufzeit hochfrequenter Schallimpulse	digitaler Ausgang
Passiv-Infrarot (PIR)	Wärmestrahlung, Grenzwert	digitaler Ausgang
Time of Flight (ToF)	Laufzeit von Lichtimpulsen	I <sup>2</sup> C
Radar (5,8 oder 24 GHz)	Personenerkennung	digitaler Ausgang, UART
<b>Temperatur, Luftfeuchte, Luftdruck</b>		
DHT11, DHT21 (Asair DHT11, Aosong AM2301)	Umgebungsluft: Temperatur, Feuchtigkeit	1-Wire
DHT20, DHT22 (Asair DHT20, Aosong AM2302)	Umgebungsluft: Temperatur, Feuchtigkeit	I <sup>2</sup> C
BMP280 (Bosch)	Umgebungsluft: Temperatur, Druck	I <sup>2</sup> C, SPI
BME680 (Bosch)	Umgebungsluft: Temperatur, Feuchtigkeit, Druck, VOC	I <sup>2</sup> C, SPI
DS18B20 (diverse Bauformen)	Temperatur	1-Wire
<b>Licht</b>		
Lichtschranke	Helligkeits-Grenzwert	digitaler Ausgang
Sonnenlicht	UV- und IR-Strahlung	I <sup>2</sup> C
Lichtfarbe	Spektrum des sichtbaren Lichts	I <sup>2</sup> C
<b>Analog-Digital-Wandler (ADC)</b>		
TI ADS1015	4-Kanal-ADC, 5,5 Volt, 12 Bit	I <sup>2</sup> C
HX711	24-Bit-ADC für Wägezellen	I <sup>2</sup> C
<b>diverse</b>		
Magnetfeldmesser (Kompass)	(Erd-)Magnetfeld	I <sup>2</sup> C
Wasserstand	Stromfluss durch benachbarte Leiterstreifen	digitaler Ausgang
Wasserstand	Schwimmerschalter	digitaler Ausgang
Wasserdurchfluss	Turbinenrad mit Hallsensor	digitaler Ausgang
Bodenfeuchtigkeit (kapazitiv)	Kapazität zwischen isolierten Leiterstreifen	analog



# IT-Sicherheitstag Gelsenkirchen

Cybersicherheit: Komplexität managen

21.11.2024  
Westfälische  
Hochschule

Unsere Partner



Bitdefender.

veeam

Der IT-Sicherheitstag der heise academy und der Westfälischen Hochschule fokussiert sich auf das Thema Cybersicherheit. Sie lernen, welche IT-Sicherheitsstrategien und -mechanismen Unternehmen helfen, sich angemessen gegen aktuelle Bedrohungen zu schützen.

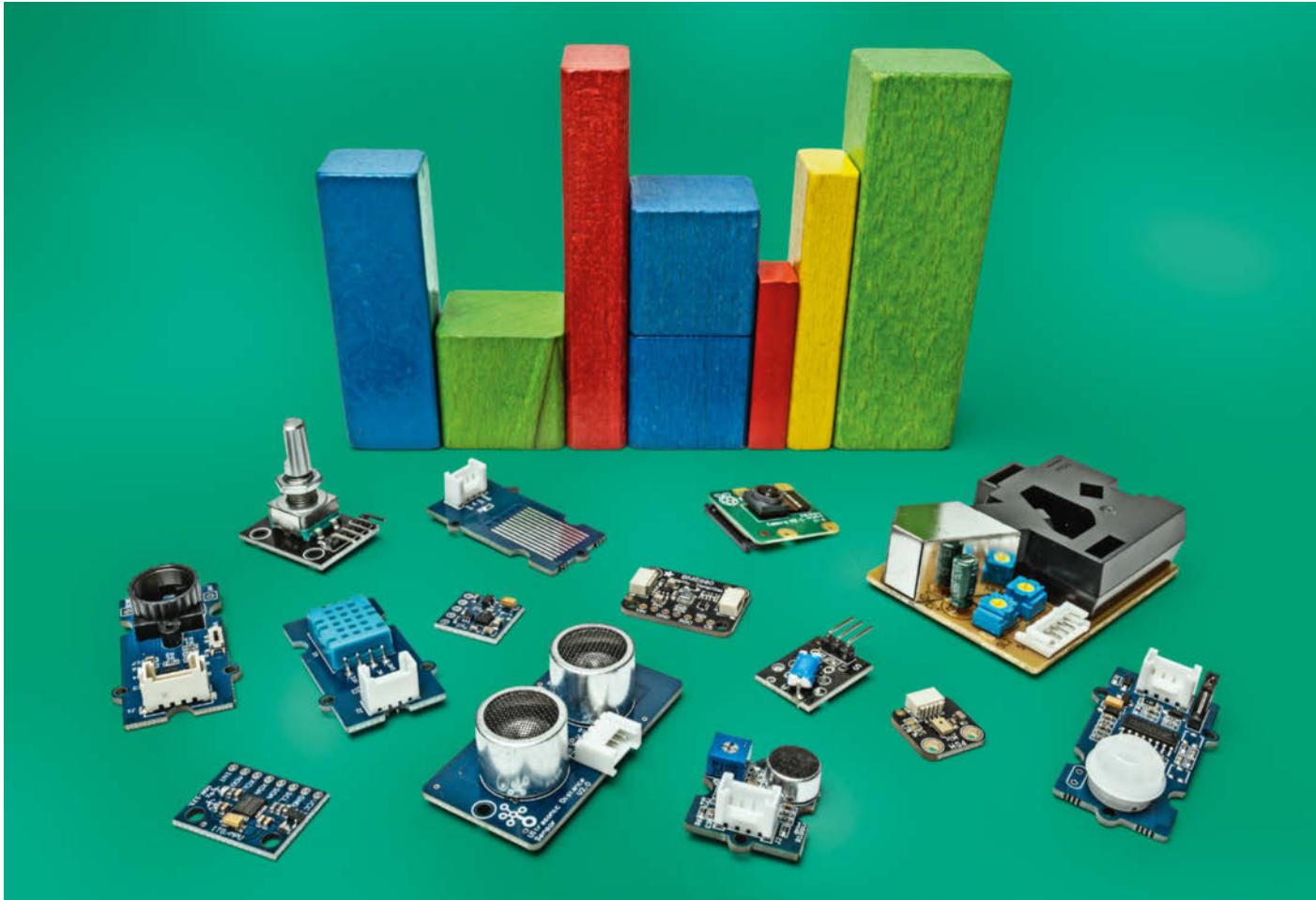
## Die Themen der Konferenz

- Welche IT-Sicherheit und Vertrauenswürdigkeit brauchen wir für unsere **komplexe digitale Zukunft**?
- Clientseitige **Verschlüsselung** für die Cloud
- Wie KI mehr IT-Sicherheit schafft
- **Digitale Signatur** zum Schutz vor Spear-Phishing
- Unternehmen oder Dienstleister: Wer managt die **IT-Sicherheit**?
- **Managed Endpoint Security**

**Jetzt Tickets sichern:**  
[konferenzen.heise.de/it-sicherheitstag](http://konferenzen.heise.de/it-sicherheitstag)

© Copyright by Heise Medien





# Daten greifen und begreifen

# Daten sammeln, verteilen und visualisieren: MQTT, Node-Red, Influx, Grafana

**Fertige oder selbstgebaute Kleingeräte mit Mikrocontroller darin, etwa Umweltsensoren, sind nicht dafür gemacht, Daten längerfristig aufzuheben. Wir zeigen, mit welchen Komponenten Sie deren Daten einsammeln, langfristig aufbewahren und auswerten.**

**Von Peter Siering**

W enn etwa in der Heimautomatisierung im Minutenrhythmus Messdaten anfallen und anderswo verarbeitet werden sollen, braucht es einen zuverlässigen Informationsvermittler. Der kann idealerweise über eine Standardschnittstelle Daten annehmen und abgeben. Mit dem „Message Queue Telemetry Transport“ (MQTT) gibt es genau das, ein Protokoll, mit dem Geräte nicht nur Daten austauschen, sondern auch Befehle empfangen können.

MQTT-fähige Geräte senden ihre Daten an einen MQTT-Server, der Broker heißt. Dort können andere MQTT-Geräte

die Daten abonnieren. Treffen neue Daten ein, reicht der MQTT-Server sie an alle Abonnenten weiter. MQTT ist keine Einbahnstraße: Jedes Gerät kann selbst auch Daten empfangen. In der Praxis erhalten sie häufig Konfigurationsdaten oder Steuerbefehle auf diesem Weg.

Auch Software kann als MQTT-Client die Datenquellen über ein Abonnement anzapfen. So lassen sich zum Beispiel Dienste realisieren, die Temperaturen regelmäßig in eine Datenbank schreiben, um sie später auszuwerten. Auch im Smart Home verwenden manche Steuerzentralen MQTT, um Lampen anzusteuern.

## MQTT entkoppelt

Unterm Strich hilft MQTT, Aufgaben zu trennen, etwa Sensoren und Steuerung, aber auch, Komponenten so zu isolieren, dass man sie austauschen kann, etwa die verwendete Steuerzentrale im Smart Home. Aber nicht nur das: Es ist auf Robustheit ausgelegt. Es kennt verschiedene Qualitätsanforderungen für die Zustellung von Nachrichten und funktioniert so auch über langsame und schlechte Verbindungen. Auf Wunsch speichert ein Broker stets die letzte Nachricht zwischen.

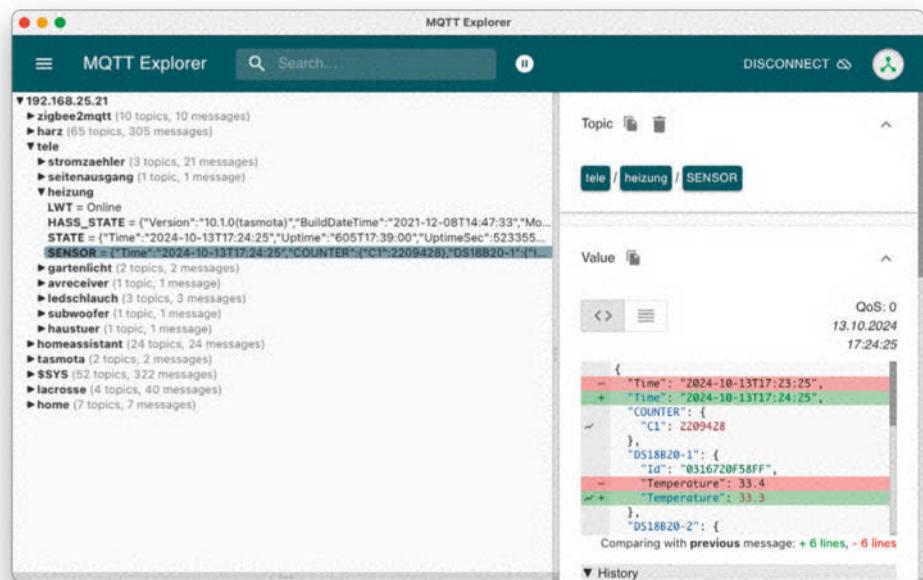
Nimmt sich einer der beteiligten Kommunikationspartner eine Auszeit, stört das die anderen nicht. Schlimmstens gehen Nachrichten verloren. Da die normalerweise ohnehin regelmäßig übertragen oder abgefragt werden, fällt das oft nicht auf. Clients können vorbeugend für einen Verbindungsabbruch einem Broker den Auftrag erteilen, Abonnenten eine besondere Nachricht zu übermitteln.

MQTT-Nachrichten bestehen aus aneinander gehängten Topics, wie zum Beispiel „Zuhause/Bad/Temperatur“. So ergibt sich auf dem Broker eine Hierarchie ähnlich dem Verzeichnisbaum eines Dateisystems. An einem solchen Topic hängt der Nachrichteninhalt. Das können Binär- oder Klartextdaten sein; oft ist es JSON. Vorgaben für die Struktur und wie Topics zu heißen haben, gibt es nicht.

MQTT-Clients können mit Wildcards Nachrichten für einen ganzen Teilbaum abonnieren: Ein angehängtes „#“ liefert alle Nachrichten an den darunter liegenden Teilbaum, etwa aus „Zuhause/Bad/#“ die Topics „Temperatur“ und „Feuchte“. Mit einem „+“ kann ein Client gezielt nur die in einem Teilbaum erfassten Werte für die Temperatur abfragen, etwa mit „Zuhause/+Temperatur“.

Nützlich für den Umgang mit MQTT ist Software, um die eingehenden Nachrichten zu beobachten. Hier empfiehlt sich der MQTT Explorer von Thomas Nordquist. Das Programm ist zwar etwas in die Jahre gekommen, aber nach wie vor sehr nützlich. Inzwischen hat der Entwickler angefangen, an der Version 0.4 zu arbeiten.

Für die Inbetriebnahme eines MQTT-Brokers lautet die Empfehlung nahezu einhellig, zu Eclipse Mosquitto zu greifen. Dieser Server-Dienst lässt sich am einfachsten per Docker einrichten. Wenn der Server auch aus dem Internet erreichbar sein soll, muss man akribisch darauf achten, die dafür eingerichteten Konten mit starken Passwörtern zu versehen und un-



**Um einen MQTT-Broker zu verstehen, hilft ein Programm, das ihn beobachtet. MQTT Explorer tut das und zeigt Änderungen an dorthin übermittelten Daten übersichtlich an, je nach Beschaffenheit sogar als Grafik.**

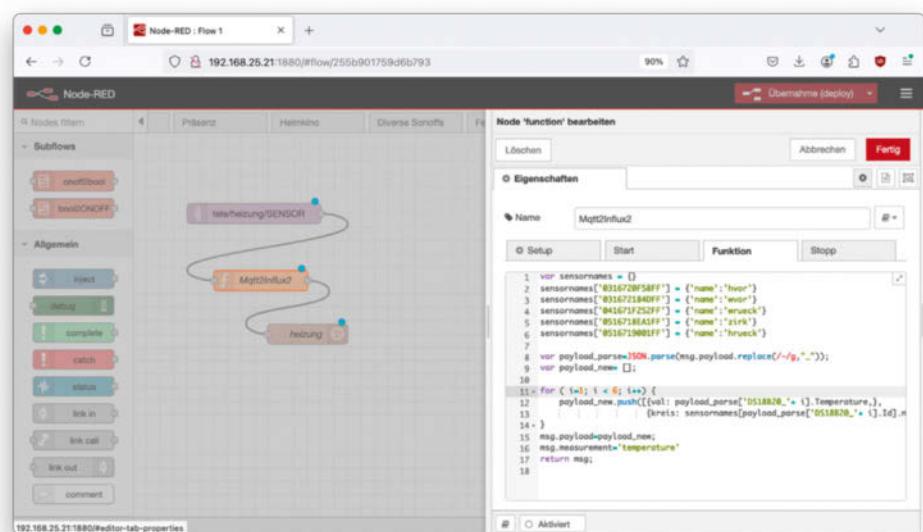
bedingt nur TLS-gesicherte Verbindungen zu verwenden.

## Node-Red verbindet

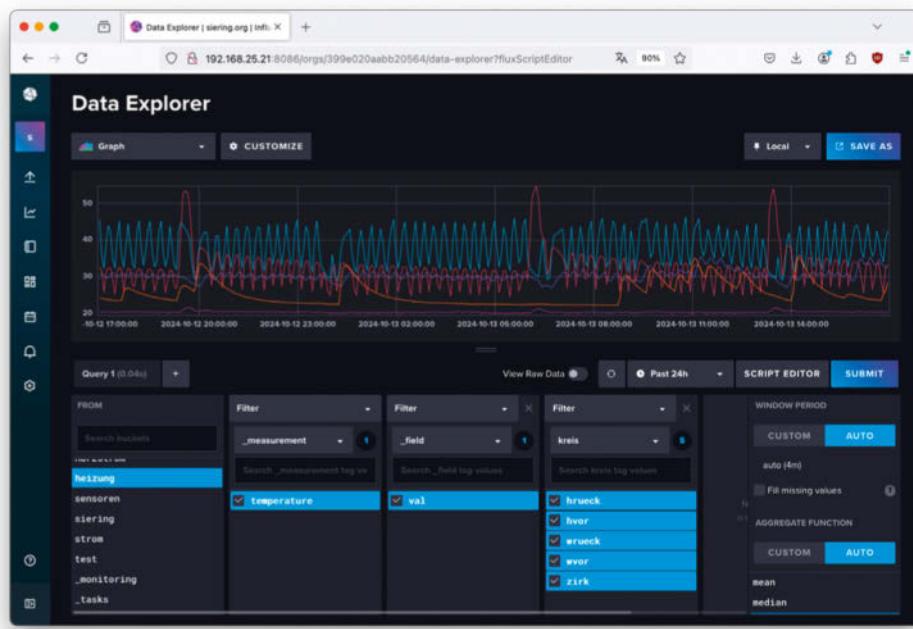
Wer selbst MQTT-Nachrichten generieren will, etwa um damit ein Gerät zu konfigurieren, kann Befehle auf der Kommandozeile einsetzen. Viel bequemer ist es aber, auf Software wie Node-Red zurückzugreifen: Dieses im Browser bediente und auf einem Server ausgeführte Werkzeug zur grafischen Programmierung spricht unter anderem MQTT.

Mit wenigen Mausklicks erstellen Sie in Node-Red eine Verbindung zu einem MQTT-Broker und abonnieren dort Topics. So können Sie Daten anzeigen oder weiterverarbeiten lassen. Nützlich ist Node-Red besonders dann, wenn Sie Daten von Geräten verarbeiten wollen, die selbst kein MQTT sprechen: Oft genügen einige Nodes oder ein paar Zeilen Code, um diese einzusammeln, passend aufzubereiten und dem Broker als MQTT-Topic zu übermitteln.

Für allerlei Funktionen und Erweiterungen stehen fertige Funktionen bereit,



**Das grafische Entwicklungswerkzeug Node-Red vermittelt bei Bedarf zwischen per MQTT bereitstehenden Sensordaten und Zeitreihendatenbanken. Dafür eingesetzter JavaScript-Code hilft dabei, Sensor-IDs in lesbare Namen zu übersetzen, die als Tags in InfluxDB landen.**



**Mit dem Data Explorer in der Weboberfläche von InfluxDB stöpselt man seit Version 2 recht schnell Auswertungen der eingelaufenen Daten zusammen und erzeugt daraus ansehnliche Dashboards.**

Knoten in der Node-Red-Diktion. Wenn die nicht genügen, um Daten von einem ins andere Format zu überführen, macht das nichts: Node-Red sieht vor, selbstgeschriebenes JavaScript in eigene Knoten zu verpacken.

Node-Red macht auch dann eine gute Figur, wenn es darum geht, per MQTT verfügbare Daten zum Beispiel an eine Zeitreihendatenbank weiterzugeben. Das ist so oft viel leichter zu realisieren als mit den umständlichen Werkzeugen, die solche Datenbanken selbst bereitstellen. Die naheliegende Lösung ist hier im Zweifel eine Ihnen vertraute Technik. Es finden sich im Netz Implementierungen, um mit Python Daten aus MQTT in eine Datenbank zu schieben.

Auf den ersten Blick klingt das alles nach einem sehr großen Aufwand für eine sehr kleine Aufgabe. Sensoren könnten doch schließlich auch direkt Daten in eine Datenbank schieben. Im Prinzip ja, aber: Dank MQTT verlassen die Daten ihr Silo und sind für so gut wie jeden gängigen Dienst zugänglich, etwa zur Heimautomatisierung mit Home Assistant, Homebridge, OpenHAB, ioBroker und so weiter – Sie lösen damit Ihre Daten und Sensoren aus der Abhängigkeit einzelner Dienste.

## InfluxDB speichert

Als Aufbewahrungsort für Sensordaten bietet sich die Zeitreihendatenbank InfluxDB an. Sie bringt eine eigene Web-

oberfläche mit, um die Grundkonfiguration einzurichten und eingegangene Daten auch grafisch darzustellen. In der Weboberfläche kann man die Abfragen mehr oder minder zusammenklicken, um die Daten zu selektieren. InfluxDB kennt obendrein zwei eigene Abfragesprachen.

InfluxDB verwendet etwas andere Konzepte und Namen als andere Datenbanken. Messreihen (measurements) entsprechen einer Tabelle in einer SQL-Datenbank. Automatisch erzeugte Zeitstempel (points) sind wie Zeilen einer SQL-Tabelle. Werte (fields), zum Beispiel eine Temperaturangabe, lassen sich mit nicht indizierten Spalten vergleichen. Zu-

sätzliche Merkmale für Werte (tags) sind verwandt mit indizierten Spalten einer SQL-Tabelle.

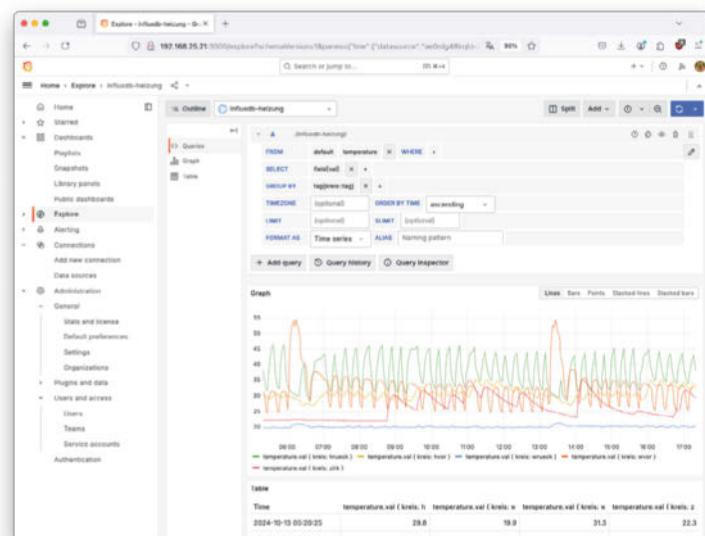
Außerdem begegnen dem Influx-Nutzer drei weitere Konzepte: Eine „Organisation“ stellt eine Verwaltungseinheit dar, innerhalb derer sich Rechte vergeben lassen. Hier genügt meist eine. Ein „Bucket“ könnte man als Container für Messreihen beschreiben. Wichtig ist die einem Bucket zugeordnete „Retention Policy“, also die Aufbewahrungsfrist; alte Daten löscht eine Influx-Installation auf Anforderung automatisch.

Für den interaktiven Zugriff auf die Weboberfläche setzt Influx auf herkömmliche Benutzernamen nebst Kennwort. Für den automatisierten Zugriff, also das Hinzufügen von Daten und deren Abruf, setzt Influx auf Tokens. Die generiert es als „API Tokens“ in seiner Weboberfläche mit unterschiedlichen Rechten. Achtung: Notieren Sie die generierten Tokens idealerweise in einem Passwortmanager, sie werden nur einmal angezeigt.

Viele Beispiele und Tutorials verwenden die an SQL orientierte Sprache „InfluxQL“, die mit InfluxDB in Version 1 eingeführt worden war. Version 2 ersetzt diese durch die im Vergleich dazu gewöhnungsbedürftige Sprache „Flux“, die völlig anders aufgebaut ist. Mit der nächsten Hauptversion wollen die Entwickler nach anhaltender Kritik zu InfluxQL zurückkehren. Kurzum: Ignorieren Sie Flux.

## Grafana garniert

Wenn Ihnen die Möglichkeiten zur grafischen Auswertung in InfluxDB nicht genügen oder Sie InfluxQL benutzen möchten, landen Sie früher oder später bei der



**Wer es hübscher mag, greift zu Grafana: Hier lassen sich die InfluxDB-Daten ebenfalls auswerten und vor dem Erstellen von dauerhaft erreichbaren Dashboards erforschen. Grafana verwendet auf Wunsch als Abfragesprache das wieder angesagte SQL-artige InfluxQL.**

Analyse- und Visualisierungslösung Grafana. Die kann nämlich auch mit der wiederbelebten Abfragesprache arbeiten. Etwas trickreich ist die Einrichtung einer Verbindung zu InfluxDB mit InfluxQL zur Datenabfrage. Unserer Erfahrung nach sind viele Anleitungen im Netz unvollständig oder führen nicht zum Ziel.

Bei uns hat das Einrichten von InfluxDB als Datenquelle in Grafana geklappt, indem wir zunächst in Influx einen Zugriffs-Token erzeugt haben. Den haben wir dann in Grafana unter Datasources als Custom HTTP-Header „Authorization“ in das Feld „Value“ eingetragen und ihm das Wort „Token“ mit einem trennenden Leerzeichen vorangestellt. Als weitere Felder sind nur noch die URL einzutragen, der Name der Datenbank (nicht die ID des Buckets) und als HTTP-Methode „GET“ zu wählen.

So mächtig InfluxDB und Grafana auch sind, so prächtig kann man sich darin verzetteln. Deshalb unser Tipp: Schauen Sie, ob nicht Influx schon genügt oder viel-

leicht die Funktionen der ohnehin schon einsatzbereiten Heimautomationslösung weit genug reichen. Manchmal ist weniger auch mehr.

Wo man Node-Red, MQTT-Broker, InfluxDB und Grafana laufen lässt, ist nicht nur eine Geschmacksfrage: Den beiden erstgenannten genügt ein Raspberry Pi 4 oder 5. Für die Zeitreihendatenbank eher nicht, weil wegen der sehr häufigen Schreibzugriffe keine SD-Karte als Speichermedium verwendet werden sollte. Und InfluxDB und Grafana brauchen etwas Dampf, wenn sie über Wochen oder Monate gesammelte Daten auswerten sollen. Ein NAS oder ein Heimserver sind dafür gut geeignet.

Wer die Dienste nicht selbst betreiben will, kann auch auf fertige Cloud-Angebote zurückgreifen. Doch das kann ganz schnell teuer werden und birgt mitunter Überraschungen: So musste InfluxData neulich einräumen, in seinen Rechenzentren vorschnell die Daten zahlender Kunden gelöscht zu haben. Die Vorwarnungen

hatten sie nicht erreicht und so wurden sie kalt erwischt.

## Strich drunter

Es klang schon an: Jedes der empfohlenen Werkzeuge ist für sich bereits ein heftiges Kaliber und braucht einige Einarbeitung. InfluxDB als Datensammelbecken ist außerdem in Bewegung. Wer die Komponenten mit Docker benutzt, sollte von der eigentlich sinnvollen Strategie Abstand nehmen, die Images automatisiert mit Updates zu versehen. Sonst sitzt man womöglich plötzlich vor einem grundrenovierten GUI, auf dem man sich kaum mehr zurechtfindet. Solange die Dienste nicht von außen erreichbar sind, scheint das für manchen der bessere Kompromiss. (ps@ct.de) ct

## Literatur

[1] Jan Mahn, Weltsprache, Das Protokoll MQTT für robusten Datenaustausch in Industrie und Hausautomation, c't 6/2018, S. 164

**Erwähnte Tools:** [ct.de/ywsf](http://ct.de/ywsf)

heise medien

**IT-Stipendium Wintersemester 2024/2025**

**heise medien fördert Studierende im Fach INFORMATIK**

Zusammen mit dem Karrierenetzwerk e-fellows.net schreiben die heise-Marken c't, iX und heise online ein **Stipendium für IT-Studierende** aus.

**Sichere Dir 600 Euro für das Wintersemester!**

Zusätzlich spendiert heise:

- Einen Schnuppertag in der jeweiligen Redaktion
- Ein Jahresabonnement von heise+
- Die Teilnahme an einem Kurs der heise academy
- Den Besuch der Sicherheitsmesse secIT

e-fellows.net



**BEWERBUNGSSCHLUSS:  
17. NOVEMBER 2024**

Jetzt in zehn Minuten bewerben:

**[www.e-fellows.net/IT-Stipendium](http://www.e-fellows.net/IT-Stipendium)**



# C-ABC

## Einführung in die Programmiersprache C

**C ist zwar nicht die modernste Hochsprache, aber eine der am verbreitetsten und flexibelsten. Man kann mit ihr Software entwickeln, die ebenso performant auf Supercomputern läuft wie auf Minirechnern oder auf noch kleineren Mikrocontrollern.**

Von Oliver Lau

**A**nfang der Siebziger haben die Informatiker Brian Kernighan und Dennis Ritchie eine Programmiersprache entworfen, die auch heute noch regelmäßig das Podest des monatlich aktualisierten TIOBE-Index erklimmt: C. Damit gehört sie in schöner Regelmäßigkeit zu den am weitesten verbreiteten Sprachen auf diesem Planeten.

Dieser Erfolg röhrt nicht von ungefähr: 1973 schrieb Dennis Ritchie zusammen mit dem Informatiker Ken Thompson das bis dato komplett in Maschinensprache programmierte Betriebssystem Unix in C um. Damit war Unix das erste in einer

Hochsprache entwickelte Betriebssystem und das Projekt setzte einen Meilenstein in der Geschichte der Informatik.

Daran erkennt man, dass sich C vortrefflich für die Entwicklung maschinennaher Software eignet, obwohl es eine

Hochsprache ist, also eine nicht maschinenspezifische. Wie es sich für eine Hochsprache gehört, ist C portabel.

Das heißt, man kann eine in C entwickelte Software ohne oder mit nur wenigen Änderungen von einer Plattform (z. B. Intel x86) auf eine andere (z. B. ARM) übertragen. Das gelingt, weil sie Details etwa der Speicherverwaltung oder der Pro-



## ct kompakt

- Die Programmiersprache C ist universell einsetzbar und die Basics sind vergleichsweise leicht zu erlernen.
- Ein Compiler übersetzt C-Code in unmittelbar von der Zielarchitektur ausführbaren Maschinencode.
- Bei der Übersetzung sorgen zahlreiche Optimierungsschritte dafür, dass der resultierende Code speicherfreundlich und außerordentlich effizient ablaufen kann.
- Das prädestiniert die Sprache für Performance-Enthusiasten und Leute, die jedes Bit zweimal umdrehen, bevor sie eine Software installieren.

zessorarchitektur abstrahiert. Damit geht einher, dass man nicht mehr mit kryptisch anmutenden Assembler-Mnemonics wie MOV, ADD oder JMP hantieren muss, sondern sich Befehlen bedienen kann, die an die englische Sprache und mathematische Formeln erinnern. C-Code ist dadurch viel besser schreib-, les- und wartbar als maschinenspezifischer Assemblercode.

Anders als in einer Assemblersprache, wo ein sogenannter Assembler die Mnemonics in direkt von der CPU ausführbare Maschinenanweisungen übersetzt, ist bei einer Hochsprache ein Compiler dafür verantwortlich, den Quellcode eines Programms in mehreren Schritten dorthin zu bringen. Im Unterschied etwa zu einem Python-Skript lässt sich C-Code also nicht direkt ausführen, sondern erst das fertige Kompilat, das zusammen mit ein paar Metainformationen zum Beispiel die Gestalt einer EXE-Datei annimmt. Das Kompilat enthält den ausführbaren Code, alle im Quellcode definierten Daten, zum Beispiel Strings (Zeichenketten), sowie ein bisschen Overhead, der zum sauberen Starten erforderlich ist und vom Compiler eingefügt wird.

Damit eine Sprache wie C maschinennah sein kann, muss sie direkten Zugriff auf den Speicher erlauben. Über sogenannte Pointer (Zeiger) können C-Programme einzelne Bytes oder Bytegruppen lesen und schreiben, darin lässt sich jedes einzelne Bit manipulieren. Mit Bytegruppen sind Ganzzahlen (Integers, mit und

ohne Vorzeichen) unterschiedlichen Wertebereichs gemeint, etwa mit 32 oder 64 Bit Breite, oder Gleitkommazahlen einfacher und doppelter Genauigkeit.

Die Passagen über Speicherzugriffe werden nicht leicht, aber vertiefen das Verständnis dafür, was in Prozessor und Speicher vor sich geht, während ein Programm läuft. Nebenher lernen Sie viele nützliche Details der Sprache C kennen. Wir setzen aber voraus, dass Ihnen fundamentale Begriffe wie Bit, Byte, Funktion, Parameter, Variable, Zuweisung, Schlüsselwort oder Typ geläufig sind. Grundkenntnisse in anderen Programmiersprachen sind von Vorteil, denn mit diesem Artikel werden Sie nicht das Programmieren im Allgemein erlernen, er soll lediglich dem Einstieg in C dienen.

## Brot und Spiele

Sie merken schon: In C zu programmieren, ist vermutlich nicht ganz einfach. Aber es lohnt sich, die Sprache zu lernen, weil die Compiler sogar aus eher schludrigem Code noch ganz viel Performance herauskitzeln können. Damit ist C viel besser als etwa die weit verbreitete Skriptsprache Python dazu geeignet, auf schwachbrüstiger Hardware wie Raspberry Pis, ESP8266-Computerchen oder noch klei-

neren Mikrocontrollern zu laufen. Auf der anderen Seite hilft sie Betreibern großer Rechenzentren, dass sich die Investition in jedes zusätzliche Gigahertz Prozessorkraft auszahlt. Und nicht zuletzt ist C ein prima Sprungbrett, um in den großen Ozean der Sprache C++ einzutauchen, mit der sich vieles von dem, was Sie im Folgenden lesen können, mit derselben Performance kompakter und besser wartbar realisieren lässt. Allerdings ist C++ noch ein paar Ecken schwieriger zu beherrschen als C.

Wir haben für diesen Artikel das beliebte Rätselspiel Wordle in C nachprogrammiert. Bei Wordle geht es darum, ein Wort aus fünf Buchstaben in maximal sechs Versuchen zu erraten. Unsere Version, die Sie unter dem Namen c't Nerd-Word auch im Web spielen können (Link via ct.de/yf4a), bedient sich nicht üblicher Wortlisten, sondern Begriffen aus dem weiten Feld von Computer und Technik, die wir aus 40 Jahren c't zusammengetragen haben [1]. Abkürzungen, Plurale und englische Wörter sind möglich – und fies! – auch Wörter mit Zahlen darin, etwa DOOM3. Für das Beispielprogramm haben wir den Wortschatz der Onlineversion ein bisschen erweitert. Wie Sie das Programm unter Windows, Linux und macOS kompilieren, steht in der README-Datei im Repository (siehe ct.de/yf4a).

Aus didaktischen Gründen haben wir für diesen Artikel „Nerd Word“ als Kommandozeilenanwendung programmiert und nicht mit einer hübschen grafischen Oberfläche, weil der Code an Übersichtlichkeit eingebüßt hätte. An diesem Spielchen wollen wir die wichtigsten Aspekte im Umgang mit C erläutern. Wir haben uns auf den 2018 veröffentlichten C-Standart C17 festgelegt, der alle wesentlichen Features umfasst, die seit der Urversion hinzugekommen sind.

## In medias res

Wie fängt man die Erklärung der Funktionsweise eines Programmes an? So blöd es klingt: am Anfang. Das ist anders als bei Skriptsprachen wie Python oder einer Shell wie Bash nicht die erste Zeile des Quellcodes, sondern eine bestimmte Funktion. Bei C heißt sie `main()`. Beim Starten des fertig kompilierten Programms wird sie als Erstes angesprungen.

Im Listing zur Mitte des Artikels, das die gesamte Spiellogik zeigt, finden Sie die Funktion ganz unten. Das Listing haben



**Rot markierte Buchstaben kommen nicht im gesuchten Wort vor, gelb markierte kommen vor, aber nicht an jener Stelle, und grüne stehen genau an der richtigen Position.**

wir aus Platzgründen um die meisten Kommentare bereinigt; im Repository bei GitHub finden Sie den kompletten Quellcode inklusive aller Anmerkungen (siehe ct.de/yf4a). Kommentare schreibt man in C-Programmen übrigens zwischen /\* und \*/; das geht auch über mehrere Zeilen hinweg. Es gibt auch einzeilige Kommentare, die // einleitet; alles von dort bis zum Zeilenende ignoriert der Compiler.

Die runden Klammern hinter dem Funktionsnamen nehmen die sogenannten Funktionsargumente auf. Das sind Parameter, die der Aufrufer der Funktion übergibt, damit sie etwas damit anstellt. Im Beispielcode ist die Funktion main() als

```
int main(int argc, char* argv[])
```

definiert (siehe die Datei wordle.c). Das bedeutet, sie erwartet zwei Parameter vom Aufrufer, in diesem Fall aus den Tiefen des Betriebssystems. Der erste ist die Anzahl der Kommandozeilenargumente (plus eins für den Programmnamen), die der User beim Aufruf des Programms eingegeben hat. Das zweite ist eine Liste von Zeigern auf jedes einzelne. Aus dieser kurzen Zeile kann man etliches über C lernen.

Im Einzelnen: Der erste Parameter hat den Namen argc und ist vom Typ int. Wann immer Sie in C einen Parameter oder eine Variable definieren, müssen Sie den Typ vor dem Namen angeben. Das werden Sie beim weiteren Durchgang durch den Code des Öfteren sehen.

int steht für Integer, eine vorzeichenbehaftete Ganzzahl mit typischerweise 32 Bit Breite, was einen Wertebereich von  $-2^{31}$  bis  $2^{31}-1$  abdeckt, also von -2.147.483.648 bis 2.147.483.647. Schreibt man stattdessen unsigned int, umfasst der Wertebereich die Zahlen 0 bis 4.294.967.296. Der C-Standard legt nicht fest, wie viel Bits ein int breit ist, nur dass es nicht weniger als 16 Bits haben darf (siehe die Tabelle „Datentypen in C“). Wenn man die genaue Größe in Byte eines Typs erfragen will, kann man sizeof verwenden; auf den meisten Plattformen wird sizeof(int) den Wert 4 liefern, also ist das int 32 Bit breit.

Das int vor dem Funktionsnamen ist der Datentyp des Wertes, den die Funktion zurückgibt. Gibt die Funktion keinen Wert zurück, muss void vor dem Funktionsnamen stehen.

Außer auf Ganzzahlen versteht sich C auch auf Gleitkommazahlen; der Beispielcode benötigt keine, sie sind aber erwähnenswert: Die meisten Programmierspra-

## Datentypen in C

Typ	min. Bits	typisch für 32-Bit-Plattform	typisch für 64-Bit-Plattform
<b>Ganzzahlen</b>			
char <sup>1</sup>	8	8	8
short <sup>1</sup>	16	16	16
int <sup>1</sup>	16	32	32
long <sup>1</sup>	32	32	64
long long <sup>1</sup>	64	64	64
<b>Gleitkommazahlen</b>			
float	—	32	32
double	—	32	64
long double	—	64 <sup>2</sup>	64 <sup>3</sup>

<sup>1</sup> immer vorzeichenbehaftet; ist kein Vorzeichen gewünscht,  
unsigned vorstellen   <sup>2</sup> variiert stark   <sup>3</sup> oder mehr

chen, auch C, richten sich nach dem Standard IEEE 754, wonach eine 32-Bit-Gleitkommazahl 1 Bit für das Vorzeichen vorsieht, 8 für den Exponenten und 23 für die Mantisse, bei 64-Bit-Gleitkommazahlen sind es 11 Bit für den Exponenten und 52 für die Mantisse. Damit lassen sich Zahlen von  $1,18 \cdot 10^{-38}$  bis  $3,4 \cdot 10^{38}$  respektive  $2,23 \cdot 10^{-308}$  bis  $1,8 \cdot 10^{308}$  darstellen, jeweils im positiven und negativen Bereich und mit 7 bis 8 beziehungsweise 15 bis 16 Dezimalstellen Genauigkeit, ebenso die Zahl 0. Details dazu in [2]. In C würde man die Zahlen übrigens 1.18e-38, 3.4e38, 2.23e-308 und 1.8e308 schreiben. Man nennt das Exponentialschreibweise und erkennt daran, dass in C wie auch in anderen Sprachen ein Punkt statt eines Kommas als Dezimalzeichen gesetzt wird.

Der Typ ist wichtig, weil C eine sogenannte statisch typisierte Sprache ist. Das heißt, dass sich der Typ einer Variablen während der Laufzeit nicht ändern kann, sondern beim Kompilieren festgelegt wird. Der Compiler stellt damit unter anderem sicher, dass Operationen nur mit kompatiblen Datentypen durchgeführt werden. Das verhindert unerwartete Fehler, die sich während der Laufzeit ergeben können, und führt zu robusterem und zuverlässigerem Code als bei dynamisch typisierten Sprachen wie Python oder JavaScript, bei denen der Typ einer Variable veränderlich ist.

Die Namen von Typen, Funktionen und Variablen heißen auch Bezeichner (engl. identifier). Sie müssen sich aus lateinischen Buchstaben, Ziffern und dem Unterstrich (\_) zusammensetzen, dürfen aber nicht mit einer Ziffer anfangen. C unterscheidet Groß- und Kleinschreibung. Würden Sie etwa Main statt main schreiben, beschwerte sich der Compiler, dass der Code keine main()-Funktion enthält.

## Zeiger und Arrays

Nun zum zweiten Parameter char\* argv[]. Der besagt erst einmal, dass er was mit dem Typ char zu tun hat. Variablen dieses Typs enthalten üblicherweise einzelne Zeichen aus dem ASCII-Vorrat oder einzelne Bytes von UTF-8-Zeichen [3]. argv ist aber kein einzelner char, sondern ein Array mit Zeigern darauf. Ein Zeiger repräsentiert eine Adresse im Hauptspeicher. Dass es ein Zeiger ist, erkennt man an dem Sternchen (\*), dass es ein Array ist, an den eckigen Klammern ([]). Als Array bezeichnet man in C eine Sammlung von Elementen gleichen Typs, die in einem zusammenhängenden Speicherbereich liegen (siehe das Bild „Array aus Zeigern (64 Bit)“), und zwar in diesem Falle so viele Zeiger, wie argc vorgibt.

Auf ein einzelnes Element innerhalb eines Arrays greift ein C-Programm über einen Index zu, den man in eckigen Klammern hinter den Variablennamen schreibt. Wie in den meisten Programmiersprachen üblich beginnt die Zählung bei 0. argv[0] referenziert also das erste Element in argv, argv[1] das zweite und so weiter. argv[0] zeigt stets auf den Namen der Datei, mit dem der User das Programm aufgerufen hat. Erst über die weiteren Indexe bekommt man die Zeiger auf die Kommandozeilenargumente, falls welche vorhanden sind. argc enthält mithin immer mindestens den Wert 1.

## Wahlweise

In den Zeilen direkt nach der Definition von main() können Sie sehen, was damit geschieht. Dort wird die Variable seed vom Typ unsigned int deklariert, also eine vorzeichenlose Ganzzahl. Dieser Wert soll in einem folgenden Schritt einen Pseudozufallszahlengenerator initialisieren. Das Gleichheitszeichen = gibt an, dass das Ergebnis des rechts davon stehenden Ausdrucks der Variable links davon zugewiesen werden soll. Nun kommt etwas, das sich Python-Programmierer in dieser Klarheit wünschen, und zwar ein sogenannter Ternäroperator, auch Bedingungsausdruck genannt:

```
unsigned int seed = (argc > 1)
? (unsigned int)atoi(argv[1])
: (unsigned int)time(NULL);
```

Der Sinn hinter diesen drei Zeilen ist Folgender: Wenn der User einen Kommandozeilenparameter übergibt, dann wird er in eine Ganzzahl gewandelt, sonst bestimmt

der aktuelle Zeitstempel den Startwert des Zufallszahlengenerators. Zufallszahlen braucht das Programm zur Auswahl des jeweils nächsten zu erratenden Wortes. Der Aufruf von `srand(seed)` initialisiert ihn in der nächsten Zeile mit dem Startwert. Auf diese Weise kann man zu Übungszwecken immer wieder dieselbe Folge aus Wörtern erraten oder eine zufällig vom Programm gewählte.

Zurück zum ternären Ausdruck: Die Bedingung `argc > 1` evaluiert zu `true` (wahr), wenn sie erfüllt ist, also `argc` einen Wert größer 1 hat, sonst zu `false` (falsch). `true` und `false` gibt es erst seit C99; davor war man gezwungen, `1` und `0` zu verwenden, was weniger aussagekräftig ist. Wenn die Bedingung `true` ergibt, landet der Wert des Ausdrucks hinter dem Fragezeichen in `seed`, sonst der hinter dem Doppelpunkt.

Im ersten Fall wandelt die Funktion `atoi()` einen String in ein `int` um. Dafür kennt C keinen eigenen Typ für Strings wie etwa `std::string` in C++, stattdessen muss man mit Zeigern auf `char` hantieren. `argv[1]` enthält einen solchen Zeiger, also ein `char*`. Er zeigt auf den Speicherbereich, der für das erste Kommandozeilenargument beim Programmaufruf angelegt wurde. Im Speicher liegen dessen einzelne Zeichen als Folge von Bytes direkt hintereinander, erfreulicherweise immer häufiger als UTF-8-Zeichen kodiert. Das Ende eines Strings kennzeichnet ein dahinter gespeichertes Byte mit dem Wert 0 (siehe Bild).

Weil `atoi()` ein `int` zurückliefert, aber `seed` vom Typ `unsigned int` ist, ist eine explizite Typkonvertierung (engl. type cast) erforderlich, damit auch der pedantischste Compiler nicht über inkompatible Typen bei der Zuweisung klagt. Das erledigt (`unsigned int`) vor dem Aufruf der Funktion.

Wenn `argc` nicht größer als 1 ist, kommt der Ausdruck hinter dem Doppelpunkt zum Zuge. Darin liefert der Aufruf von `time()` den aktuellen Unix-Timestamp zurück. Das ist die Anzahl der Sekunden, die zum Zeitpunkt des Aufrufs seit dem 1.1.1970 um 0 Uhr UTC verstrichen sind, Schaltsekunden inklusive.

## Typen definieren

`time()` gibt die Anzahl der Sekunden als Ganzzahl vom Typ `time_t` zurück. Das ist ein Alias für eine Ganzzahl, deren tatsächlicher Typ vom Betriebssystem abhängt. Unter macOS ist es zum Beispiel ein `long`. Die explizite Konvertierung in ein `unsigned`

`int` stellt sicher, dass der Wert garantiert zum Typ von `seed` passt.

Einen Typ-Alias kann man selbst definieren. Das ist etwa sinnvoll, wenn wie bei `time_t` der tatsächliche Typ vom Betriebssystem oder von anderen Bedingungen abhängt oder wenn man eine Bezeichnung für einen eher abstrakt klingenden Typ wie `int` oder `long` benötigt, die klarmacht, welche Art von Daten er beschreibt. Im Code schreiben Sie dafür einfach:

```
typedef long time_t;
```

Also vorhandener Typ zuerst, dann der Alias dafür.

`time()` erwartet einen Parameter vom Typ `time_t*`, also einen Zeiger. Wenn der `NULL` ist, ignoriert die Funktion ihn einfach. `NULL` steht typischerweise für `(void*)0`, was einen Zeiger unbestimmten Typs (`void`) beschreibt, der den Wert 0 hat. `NULL` ist ein sogenanntes Makro, das der Präprozessor des Compilers – das ist eine Vorstufe beim Kompilieren – durch einen konkreten Wert ersetzt. Ein Makro definiert man mit der Präprozessordirektive `#define`. Im Beispielcode sehen Sie das ganz oben in der Datei, wo zum Beispiel das Makro `MAX_TRIES` den Wert 6 bekommt.

Ein Programm kann `time()` auch mit einem Zeiger ungleich `NULL` aufrufen:

```
time_t t0;
time(&t0);
```

Das ignoriert den Rückgabewert. Stattdessen landet der Unix-Timestamp in der Variablen `t0`, für die beim Start des Programms automatisch irgendwo Platz reserviert wurde. Zugegebenermaßen ist „irgendwo“ reichlich unspezifisch, aber zur Speicherorganisation in C, C++ und anderen hardwarenahen Sprachen könnte man einen eigenen Artikel schreiben. Die Speicheradresse liefert das & vor dem Variablennamen. Der Funktion wird also ein Zeiger auf `t0` übergeben. `time()` schnappt sich diesen Zeiger und dereferenziert ihn, was bedeutet, dass aus dem `time_t*` ein `time_t` wird:

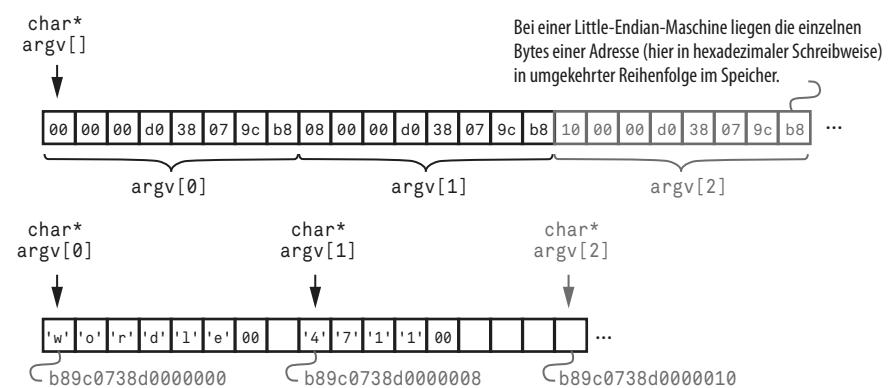
```
time_t time(time_t* tloc) {
    *tloc = 1721916279;
    return *tloc;
}
```

Damit landet die Zahl 1721916279 nicht im Zeiger `tloc`, sondern in der darüber referenzierten Variable `t0` vor dem Aufruf

## Array aus Zeigern (64 Bit)

Wenn der User ein C-Programm mit einem Kommandozeilenargument aufgerufen hat (hier „4711“), dann erhält der Parameter `argc` in der Funktion `main(int argc, char* argv[])` nicht den Wert 1, sondern 2, weil der Name des Programms mitgezählt wird. `argv[]` enthält Zeiger darauf. In C zeigt ein Null-Byte das Ende eines Strings an, deshalb muss man immer ein Byte mehr für einen String einplanen, als er tatsächlich lang ist.

Dass zwischen den Speicherorten der Strings Lücken sind, hängt mit der Computerarchitektur zusammen, bei der wir für das Beispiel von einer mit 64 Bit (8 Byte) ausgegangen sind. Die meisten CPUs können auf durch 8 teilbare Adressen schneller zugreifen als auf nicht durch 8 teilbare. Wenn ein Compiler Speicher für Variablen reserviert, berücksichtigt er in der Regel diesen Umstand.



```

#include <ctype.h> // tolower()
#include <stdbool.h> // bool, true und false
#include <stdio.h> // stdin, getchar(), fgets()
#include <stdlib.h> // atoi(), srand(), rand()
#include <string.h> // strncmp(), strchr()
#include <time.h> // time()
#include "words.h"

#define WORD_BUF_LEN (WORD_LENGTH + 1)
#define MAX_TRIES (6)

enum status {
    UNMARKED, // 0
    NOT_PRESENT, // 1
    PRESENT, // 2
    CORRECT // 3
};
typedef enum status state_t;

typedef struct {
    const char* word;
    char guess[WORD_BUF_LEN];
    state_t result[WORD_LENGTH];
    int n_tries;
} game_state;

bool word_is_allowed(const char* word) {
    for (int i = 0; i < NUM_WORDS; ++i) {
        if (strncmp(word, words[i], WORD_LENGTH) == 0)
            return true;
    }
    return false;
}

bool is_character_marked(game_state* state,
                        char c, int fromIdx) {
    for (int i = fromIdx; i < WORD_LENGTH; ++i) {
        if (state->guess[i] == c &&
            state->result[i] != UNMARKED)
            return true;
    }
    return false;
}

void update_state(game_state* state) {
    for (int i = 0; i < WORD_LENGTH; ++i)
        state->result[i] = UNMARKED;
    for (int i = 0; i < WORD_LENGTH; ++i) {
        if (state->guess[i] == state->word[i])
            state->result[i] = CORRECT;
    }
    for (int i = 0; i < WORD_LENGTH; ++i) {
        if (state->result[i] == CORRECT)
            continue;
        char c = state->guess[i];
        state->result[i] =
            (strchr(state->word, c) != NULL &&
             !is_character_marked(state, c, i))
            ? PRESENT
            : NOT_PRESENT;
    }
}

void get_input(game_state* state) {
    bool bad_word = true;
    do {
        printf("\n%d. Versuch:\n", state->n_tries);
        if (fgets(state->guess, WORD_BUF_LEN, stdin) == NULL)
            return;
        int ch;
        while ((ch = getchar()) != EOF && ch != '\n')
            /* lesen bis Eingabe- oder Zeilenende */;
        state->guess[WORD_LENGTH] = '\0';
        bad_word = !word_is_allowed(state->guess);
        if (bad_word)
    } while (bad_word);
}

printf("Das Wort ist nicht in der Liste "
      "erlaubter Wörter.\n");
} while (bad_word);

void print_result(const game_state* state) {
    printf("!");
    for (int i = 0; i < WORD_LENGTH; ++i) {
        switch (state->result[i]) {
        case CORRECT:
            printf("\033[37;42;1m");
            break;
        case PRESENT:
            printf("\x1b[37;43;1m");
            break;
        case NOT_PRESENT:
            // fall-through
        default:
            printf("\033[37;41;1m");
            break;
        }
        printf("%c", state->guess[i]);
    }
    printf("\033[0m\n");
}

bool another_round(void) {
    printf("Noch eine Runde? (J/n) ");
    char answer = (char)tolower(getchar());
    bool yes = answer == 'j' || answer == '\n';
    if (yes)
        printf("\nSuper! Dann mal los ... \n");
    return yes;
}

int main(int argc, char* argv[]) {
    unsigned int seed = (argc > 1)
        ? (unsigned int)atoi(argv[1])
        : (unsigned int)time(NULL);
    srand(seed);
    printf("\nNERD WORD\n\n"
           "Errate das Wort mit %d Buchstaben in"
           "maximal %d Versuchen.\n"
           "(Abbrechen mit Strg+C bzw. Ctrl+C)\n",
           WORD_LENGTH, MAX_TRIES);
    bool keepRunning = true;
    while (keepRunning) {
        game_state state;
        state.word = words[rand() % NUM_WORDS];
        bool doRestart = false;
        for (state.n_tries = 1;
             state.n_tries <= MAX_TRIES && !doRestart;
             ++state.n_tries) {
            get_input(&state);
            update_state(&state);
            print_result(&state);
            if (strncmp(state.guess, state.word, WORD_LENGTH)
                == 0) {
                printf("\nHurra, Du hast das Wort im "
                       "%d. Versuch gefunden!\n",
                       state.n_tries);
                doRestart = true;
                keepRunning = another_round();
            }
            else if (state.n_tries == MAX_TRIES) {
                printf("\nSchade, Du hast das Wort nicht "
                       "erraten.\nEs lautete: %s.\n",
                       state.word);
                keepRunning = another_round();
            }
        }
        printf("\nDanke, es war schön mit dir :-)\n\n");
    }
    return EXIT_SUCCESS;
}

```

Die gesamte Spiellogik unseres Wordle-Klons umfasst gerade einmal 146 Zeilen C-Code.

von `time()`. 1721916279 ist übrigens der Timestamp des Moments, kurz bevor die Zahl in den Artikeltext eingefügt wurde. In einer realen Implementierung von `time()` würde die Funktion den Wert durch den Aufruf einer Betriebssystemfunktion ermitteln. Zusammengefasst: & vor einer Variable liefert deren Adresse, \* kehrt den Vorgang um.

An dieser Funktion sehen Sie, was Ihnen sicher schon beim Betrachten des gesamten Listings aufgefallen ist: Ge-schweifte Klammern definieren einen Block. Das heißt, die eigentliche Funktion (der Funktionskörper) beginnt mit der öffnenden Klammer und endet mit der korrespondierenden schließenden. Blöcke werden später noch wichtig, wenn es um sogenannte Scopes geht. Und noch etwas sehen Sie: Mit dem Schlüsselwort `return` gibt eine Funktion einen Wert zurück; ein Semikolon beendet einen Befehl. Bei vergessenen Semikola stoppt der Compiler mit einer Fehlermeldung. Wer von JavaScript oder Python kommt, wird möglicherweise darüber fluchen ...

## Was ausgeben

Ein Programm wie ein Spiel tut gut daran, den Nutzer mit einer freundlichen Meldung zu empfangen. Für Ausgaben auf die Kommandozeile steht der Befehl `printf()` zur Verfügung, was die Abkürzung von „print formatted“ ist. Das bedeutet, ein Programm kann damit nicht nur statischen Text ausgeben, sondern darin dynamisch Werte etwa von Variablen einweben. Dafür integriert man Platzhalter in den String, die mit einem Prozentzeichen beginnen. Das `%d` steht für die dezimale Darstellung eines `int`. Die eingebetteten Platzhalter füllt `printf()` mit den auf den String folgenden Parametern, im Beispiel Konstanten für die Wortlänge (`WORD_LENGTH`) und die maximal erlaubte Zahl von Rateversuchen (`MAX_TRIES`).

Wichtig: Die Platzhalter müssen zum Typ der Werte passen, die damit befüllt werden sollen. Compiler geben in der Regel nur eine Warnung aus, wenn etwas damit im Argen liegt, also wenn Sie zum Beispiel mit `%ull` einen Platzhalter für einen `unsigned long long` festlegen, aber ein `int` als Wert angeben. Die Folge können zerhackte String-Ausgaben und sogar Programmabstürze sein.

Die `\ns` im String stehen übrigens für Zeilenumbrüche (Line Feed). Für Details zu `printf()` lohnt sich ein Blick in die Dokumentation des Befehls (siehe ct.de/

yf4a, in unixoiden Betriebssystemen auch zu erreichen über `man 3 printf` auf der Kommandozeile).

## Schleifen

Typisch für Spiele ist, dass sie in einer Endlosschleife so lange mit dem Nutzer interagieren, bis er keine Lust mehr hat und abbricht. Eine solche Schleife formt `while(keepRunning)`. Sie führt alles im von den geschweiften Klammern umfassten Block so lange aus, bis die Bedingung in den runden Klammern nicht mehr erfüllt ist. Man hätte auch `do { } while(keepRunning);` schreiben können.

Zu Beginn ist `keepRunning` gleich `true`, die Ausführung des Schleifenblocks kann also beginnen. Der Wert von `keepRunning` ändert sich, wenn der User später meint, dass er aufhören möchte. Doch erst einmal weiter mit den nächsten Zeilen im Code.

Im Verlauf des Spiels merkt sich das Wordle-Programm seinen Zustand. Dazu gehört neben anderen das geratene Wort und die Anzahl der Rateversuche. Solche logisch zusammenhängenden Werte legt man sinnvollerweise nicht einfach in separaten Variablen ab, sondern fasst sie zusammen. Eine solche Zusammenfassung repräsentiert der Typ `game_state`, von dem in der ersten Zeile des `while`-Blocks die Variable `state` angelegt wird.

`game_state` ist ein Alias für ein `struct`. Das ist ein Schlüsselwort, das die Definition eines Konservats von Variablen einleitet. Sie sehen die Definition weit oben im Code. Darin ist der erste Wert ein Zeiger auf das zu erratende Wort, der zweite ein String für das zuletzt geratene, der dritte ein Array mit der Auswertung eines jeden Buchstabens darin und der letzte die Anzahl bisheriger Rateversuche. Mit `typedef` bekommt das `struct` einen eigenen Namen. Das ist eine Kurzschreibweise für folgenden, gleichbedeutenden Code:

```
struct _game_state {
    const char* word;
    char guess[WORD_BUF_LEN];
    state_t result[WORD_LENGTH];
    int n_tries;
}

typedef struct _game_state game_state;
```

Zurück zur Schleife. Dort geht es weiter mit der Zuweisung eines Wortes aus dem Array der erlaubten Wörter. Die Funktion `rand()` bestimmt pseudozufällig einen Index in das Array und weist den so erhaltenen Zeiger dem String `state.word` zu.

Der Punkt bedeutet: Nimm die Variable `state` und wähle aus dem `struct` die Variable `word`. Das Prozent-Zeichen `%` ermittelt den Rest aus der Division zweier Ganzzahlen. Damit ist das zu erratende Wort gesetzt. Woher die Wörter in `words` kommen, erklären wir später.

## Schleifen, die Zweite

Nun lernen Sie den zweiten sehr häufig benutzten Schleifentyp kennen: die `for`-Schleife. Man benutzt sie vornehmlich zum Iterieren über einen Wertebereich, im Code über die Nummer des aktuellen Rateversuchs `state.n_tries`. Was zwischen den runden Klammern steht, teilt sich in drei durch Semikola getrennte Bereiche auf:

- Der erste bestimmt den Startzustand, indem er `state.n_tries` auf 1 setzt.
- Der zweite beschreibt die Abbruchbedingung; solange sie nicht erfüllt ist, läuft die Schleife weiter. Sie wird wie oben beim Beispiel mit `while` beim Eintritt in die Schleife geprüft. Zwei Bedingungen müssen erfüllt sein: Die Anzahl der Rateversuche muss kleiner oder gleich der maximal erlaubten Versuche `MAX_TRIES` sein *und* – das meint der boolesche Operator `&&` – `doRestart` muss `false` sein (! kehrt einen booleschen Ausdruck um).
- Der dritte Bereich legt fest, was am Ende der Schleife passieren soll. Im Beispiel zählt der Inkrement-Operator `++` den Wert von `state.n_tries` um eins hoch. Vom Inkrement-Operator gibt es zwei Varianten, Prä- und den Postinkrement-Operator, die sich im Verhalten unterscheiden. Beispiele:

```
int a = 1;
int b = ++a;
int c = a++;
```

Die erste Schreibweise `++a` enthält den Präinkrement-Operator, weil er *vor* die Variable geschrieben wird, die er hochzählen soll, und auch, weil er in der Priorität *vor* allen anderen Operatoren steht. Deshalb wird `a` zuerst um eins hochgezählt und dann das Ergebnis `b` zugewiesen. Danach sind sowohl `a` als auch `b` gleich 2. Beim Postinkrement-Operator ist es anders herum: Erst erhält `c` den Wert von `a`, dann wird `a` inkrementiert. Zum Abschluss dieses Codeschnipsels ist also `a` gleich 3, `b` gleich 2 und `c` gleich 2.

In der obigen `for`-Schleife ist die Variante `egal` und es ist eher eine Ge-

schmacksfrage, welche man wählt. Der Präinkrement-Operator stellt die Operation in den Vordergrund, nicht die Variable, was zum Verständnis beim Lesen eines Codes beitragen kann.

In der Schleife passieren drei Dinge, die wesentlich für Spiele sind: mit dem User interagieren, den Spielzustand dementsprechend aktualisieren und das Ergebnis präsentieren. Das erledigen die drei Funktionen `get_input()`, `update_state()` und `print_result()`.

Sie erwarten einen Zeiger auf das `struct` mit dem Spielzustand, nicht das `struct` selbst. Das trägt zur Performance bei, weil das Programm auf diese Weise nicht sämtliche einzelne Werte beim Aufruf übergeben und anschließend das komplette `struct` wieder zurückgeben muss, sondern nur einen Zeiger. Das sind also immer 4 oder 8 Byte in Abhängigkeit davon, ob das Programm auf einer 32- oder einer 64-Bit-Maschine läuft. Wenn die drei Funktionen Werte innerhalb des `struct` verändern, verändern sie damit die Werte der in `main()` definierten Variable `state`.

In `get_input()` und den weiteren Funktionen werden Sie viel von dem wiederentdecken, was wir bereits besprochen haben. Darum im Folgenden immer nur bislang Unerwähntes.

Weiter gehts mit dem Aufruf von `fgets()` (für „file get string“), der das Feld `guess` mit der Eingabe des geratenen Wortes füllt. Weil `state` in der Funktion ein Zeiger ist, muss man `->` statt `.` verwenden, um auf dessen Felder zugreifen zu können. `fgets()` liest die Eingabe des Users aus der Standardeingabe (`stdin`), und zwar maximal so viele Zeichen, wie `WORD_BUF_LEN` vorgibt. `WORD_BUF_LEN` ist um 1 größer als die erlaubte Wortlänge, weil ja noch ein 0-Byte als Endekennzeichen des Strings folgt.

Danach liest eine `while`-Schleife alle gegebenenfalls zu viel eingetippten Zeichen mit `getchar()` aus der Standardeingabe und verwirft sie. Weil `state->guess` als letztes Zeichen etwas anderes als ein 0-Byte haben könnte, wird es durch ein solches ersetzt.

In Wordle darf man nur Begriffe raten, die in der Liste der zu erratenden Wörter enthalten sind. Das prüft die Funktion `word_is_allowed()`. Die `do/while`-Schleife endet erst, wenn ein erlaubtes Wort eingetippt wurde, also `bad_word` gleich `false` ist.

In `word_is_allowed()` lernen Sie eine weitere, sehr nützliche Funktion kennen:

`strcmp()`. Sie vergleicht zwei Strings miteinander, aber niemals mehr Zeichen, als im dritten Parameter übergeben wird. Sie liefert einen Wert kleiner 0 zurück, wenn der erste String lexikografisch vor dem zweiten steht, einen Wert größer 0, wenn er in lexikografischer Sortierung auf den zweiten String folgt, und 0, wenn die Strings identisch sind. Sind sie identisch, gibt `word_is_allowed()` den Wert `true` zurück. Sind sie es nicht, geht die umrahmende `for`-Schleife zum nächsten Wort über, bis das letzte erreicht ist. Wenn sie das letzte erreicht hat und das gesuchte Wort nicht in der Liste enthalten ist, erhält der aufrufende Code den Wert `false`.

`char* word` ist in der Parameterliste von `word_is_allowed()` als `const` gekennzeichnet, was bedeutet, dass die über den Zeiger referenzierbaren Bytes im Speicher nicht modifiziert werden dürfen. Würde die Funktion also versuchen, etwa mit `word[0] = 'a'`; das erste Zeichen zu ändern, gäbe der Compiler eine Fehlermeldung wie „error: read-only variable is not assignable“ aus.

Damit ist das Feld `guess` aktualisiert, und zwar in der `state`-Variable, die in `main()` definiert wurde, weil `get_input()` ja mit einem Zeiger darauf hantiert hat.

Wenn `get_input()` zurückkehrt, schreitet die Ausführung des Programms bei `update_state()` fort. Bis auf die Funktion `strchr()`, die nach einem Zeichen in einem String sucht und einen Zeiger ungleich `NULL` zurückliefert, wenn es enthalten ist, haben Sie alle dort verwendeten C-Konstrukte schon kennengelernt.

## Enumerationen

Obwohl ... nicht ganz: Es tauchen darin Symbole wie `UNMARKED` und `CORRECT` auf, die einer Erwähnung bedürfen. Bei ihnen handelt es sich um Werte aus einer sogenannten Enumeration weiter oben im Beispielprogramm (suchen Sie nach `enum status`). Sie sind wichtig für die in `update_state()` stattfindende Auswertung des geratenen Wortes. `UNMARKED` meint, dass der betreffende Buchstabe noch nicht ausgewertet wurde, `NOT_PRESENT` steht für einen Buchstaben, der nicht im gesuchten Wort vorkommt, `PRESENT` für einen Buchstaben, der vorkommt, aber fehlplatziert ist, und `CORRECT` für einen richtig platzierten Buchstaben.

Enumerationen sind nützlich, um selbstdokumentierenden Code zu schreiben. Der Compiler weist den in einer Enumeration aufgeführten Bezeichnern

von 0 beginnend automatisch aufsteigende Werte zu. Das ist viel praktischer und weniger fehleranfällig, als dasselbe Ergebnis mit der Deklaration von Konstanten à la

```
const int UNMARKED = 0;
const int NOT_PRESENT = 1;
const int PRESENT = 2;
const int CORRECT = 3;
```

erreichen zu wollen.

Das Resultat der Auswertung in `update_state()` landet im Array `state->guess`. Die erste `for`-Schleife setzt alle Elemente des Arrays auf `UNMARKED`. Das ist wichtig, weil die Werte aller Elemente eines Arrays mehr oder weniger dem Zufall unterliegen. Das heißt, wenn man es deklariert, steht darin, was zum Zeitpunkt der Deklaration im dafür reservierten Speicher stand.

## Fallweise bunt machen

Wenn `update_state()` alle Buchstaben ausgewertet hat, kann `print_result()` das Ergebnis ausgeben. In der Funktion ist ein nettes Feature erwähnenswert, das die Ausgabe der einzelnen Buchstaben optisch aufpeppt. Die Rede ist von den sogenannten ANSI-Escape-Codes. Sie beginnen mit `\033[` und enden auf `m`. `\033` ist der Oktalwert für das Escape-Zeichen (dezimal 27), das man auch in hexadezimal `\x1b` hätte schreiben können. Darauf folgen mit Semikola getrennte Werte in dezimal: Sie stehen im Beispiel für die Vordergrundfarbe, also die Farbe des auszugebenden Zeichens, für die Hintergrundfarbe und für den Stil (1 steht zum Beispiel für Fettdruck). `\033[0m` stellt den Ursprungszustand des Terminals wieder her. Unter [ct.de/yf4a](http://ct.de/yf4a) finden Sie Links auf Webseiten, die die erlaubten Werte auf-listen.

Wenn ein Buchstabe `CORRECT` ist, erscheint er auf grünem Hintergrund, ein vorhandener, aber fehlplatziert auf gelbem, alle anderen auf grauem Hintergrund. Diese Fälle unterscheidet `switch/case`. Wichtig ist das `break` am Ende einer Fallunterscheidung. Fehlt es, setzt die Ausführung einfach an der nächsten Zeile fort. Es hat sich bewährt, ein bewusstes Auslassen eines `break` mit einem Kommentar wie `// fall-through` zu kennzeichnen. Ebenso hat es sich bewährt, immer alle sinnvollen Fälle explizit zu berücksichtigen und dann noch mit `default` alle anderen möglichen. Manch ein Compiler

gibt Warnungen aus, wenn man das nicht tut.

## Klammern und Geltungsbereiche

Anders als etwa in Python (`match`), Swift (`switch`) oder Kotlin (`when`) haben `cases` in C keinen Scope (Geltungsbereich). Das bedeutet, dass die darin definierten Variablen auch außerhalb des `case` sichtbar sind, also im Kontext des `switch` gelten. Stattdessen geben in C geschweifte Klammern den Scope einer Variable vor: Eine Variable ist nur innerhalb des dadurch definierten Kontextes sichtbar. Wenn man in C einen `case` „scopen“ will, muss man ihn mit geschweiften Klammern umrahmen, etwa wie folgt, damit man die Variable `scent` mehrfach verwenden kann:

```
switch (choice) {
    case 1: {
        char* scent = "4711";
        // ...
    }
    break;
    case 2: {
        char* scent = "Chanel 5";
        // ...
    }
    // ...
}
```

Auch eine Variable, die man innerhalb der runden Klammern einer `for`-Schleife definiert hat, gilt nur innerhalb der Schleife. Deshalb kann man wie zum Beispiel in `update_state()` die Zählvariable `i` mehrfach innerhalb der Funktion verwenden, muss sie dann aber auch mehrfach deklarieren. Es handelt sich dabei um unterschiedliche Variablen, auch wenn sie denselben Namen tragen.

Variablen innerhalb von geschweiften Klammern bilden einen sogenannten lokalen Kontext. Davon zu unterscheiden ist der globale Kontext. Ihn definiert man außerhalb jeglicher Klammern, also auf der Ebene, auf der alle Funktionen definiert sind. Alle im globalen Kontext definierten Bezeichner sind auch in allen lokalen Kontexten verfügbar.

## Header-Dateien

Die Wortliste in `words` befindet sich im globalen Kontext. Das Folgende klärt, wie die Wortliste eingebunden wird und woher Funktionen wie `getchar()`, `strcmp()` oder `fgets()` kommen. Die Wortliste finden Sie

in der separaten Datei `words.c`. Ein Auszug:

```
const
char words[NUM_WORDS][WORD_LENGTH + 1]
= {
    "abort",
    "acorn",
    "acryl",
    // ...
};
```

Die Konstante `words` steht nun aber nicht wie von Geisterhand in `wordle.c` zur Verfügung, sondern man muss sie über das Einbinden einer sogenannten Header-Datei bekannt machen. Das geht mit den `#include`-Anweisungen ganz oben im Code.

Die Header-Datei `words.h` enthält zum Beispiel Folgendes:

```
#ifndef __WORDS_H__
#define __WORDS_H__

#define WORD_LENGTH (5)
#define NUM_WORDS (627)
extern const char
    words[NUM_WORDS][WORD_LENGTH + 1];

#endif
```

`#ifndef` prüft, ob das Makro `__WORDS_H__` bereits definiert ist. Wenn nicht, wird der Teil bis zum korrespondierenden `#endif` importiert, sonst nicht. Das stellt sicher, dass man den Inhalt einer Header-Datei nicht aus Versehen zweimal importiert. Dann schimpft der Compiler gegebenenfalls über mehrfach definierte Symbole.

Das Schlüsselwort `extern` drückt aus, dass das Array `words` an anderer Stelle implementiert ist, eben in der Datei `words.c`.

Die Funktionen `getchar()`, `strcmp()` oder `fgets()` sind in diversen Header-Dateien definiert. Wo welche definiert sind, haben wir in Kommentaren hinter den `#includes` vermerkt. Ihre Implementierung finden Sie nicht im Quellcode des Spiels, weil es sich um Funktionen aus der C-Standardsbibliothek handelt. Solche Libraries sind bereits fertig kompilierte Dateien, die etwa unter Linux typischerweise im Verzeichnis `/usr/lib` zu finden sind.

## Modifikationsanreiz

Das wars auch schon mit unserem Schnell-durchlauf durch die Sprache C. Ein mahn-

nendes Wort noch zum Abschluss: Mit der Benutzung von Zeigern geht große Verantwortung einher, damit Ihr Code nicht versehentlich in falsche Speicherbereiche schreibt. Dazu ein Beispiel:

```
char *strtolower(char* s) {
    for (char* p = s; *p != '\0'; p++)
        *p = (char)tolower(*p);
    return s;
}
```

Diese Funktion wandelt mit der Funktion `tolower()` alle Großbuchstaben eines Strings in Kleinbuchstaben um. Weil Strings in C mit einem Null-Byte enden, durchläuft sie den String deshalb so lange, bis sie auf ein solches Zeichen stößt.

Doch Halt! Was ist, wenn ein String versehentlich nicht mit einem Null-Byte endet? Dann erreicht die Schleife vielleicht einen Speicherbereich, der mit anderen wichtigen Daten gefüllt oder nicht mehr fürs Programm reserviert ist. Es können dann schreckliche Dinge passieren: Das Programm zerstört Daten oder stürzt mit einem Speicherzugriffsfehler ab.

Als kleine Fingerübung modifizieren Sie die Funktion so, dass sie garantiert spätestens nach dem fünften Buchstaben abbricht (`WORD_LENGTH` verwenden). Dann platzieren Sie die Funktion in `wordle.c` und passen `get_input()` so an, dass sie das erwartete Wort in Kleinbuchstaben umwandelt. Denn in der aktuellen Implementierung unseres Nerd Word sind auch Großbuchstaben in der Eingabe erlaubt, aber die Wortliste umfasst nur Wörter in Kleinbuchstaben.

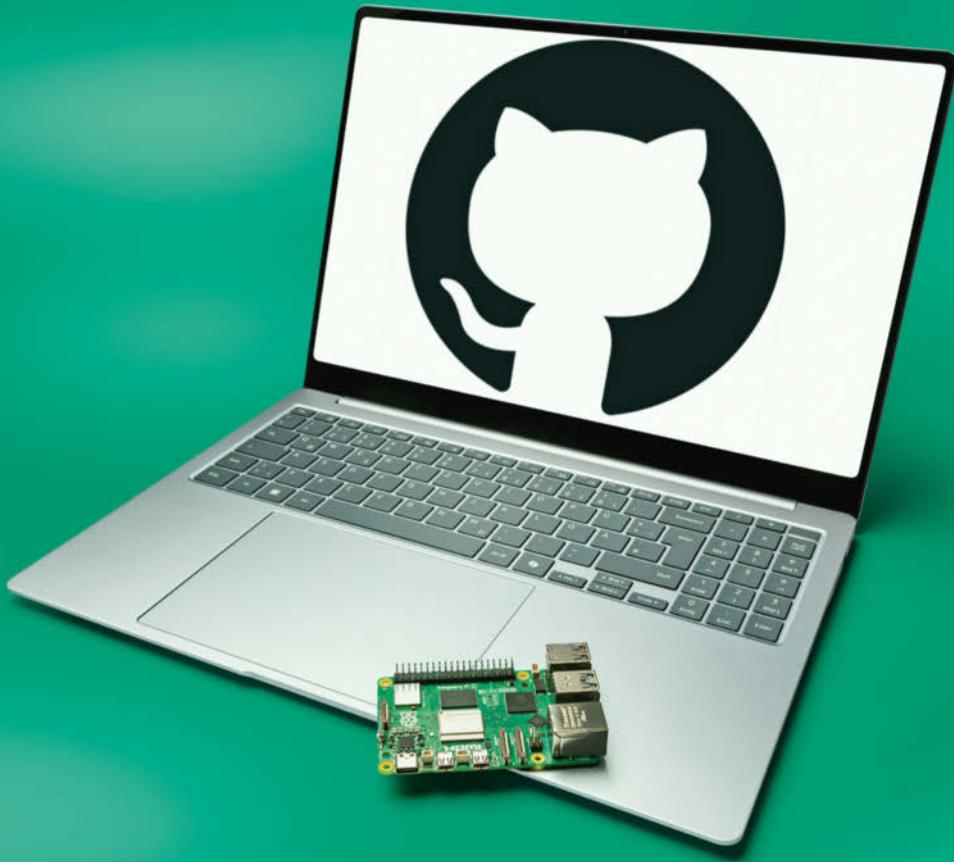
## Summa summarum

Sie wissen nun: C erfordert große Sorgfalt beim Entwickeln. Ihr Lohn: schneller und kompakter Code, bestens geeignet für alle Maschinen von David bis Goliath. Wenn Sie die Sprache C einmal verinnerlicht haben, steht Ihnen eine riesige Welt an Bibliotheken offen. Bon voyage! (ola@ct.de) ct

## Literatur

- [1] Niklas Dierking, c't-Textmining: Wir zählen Wörter aus 40 Jahren c't, c't 12/2023, S. 128
- [2] Hajo Schulz, Nummerngedächtnis, Wie Computer Zahlen speichern, c't 23/2014, S. 172
- [3] Hajo Schulz, Zeichentrickserien, Textkodierung von ASCII bis Unicode, c't 20/2017, S. 170

**Quellcode bei GitHub, C-Spezifikation:**  
[ct.de/yf4a](https://ct.de/yf4a)



# Git für alle

## Wegweiser durch Git, GitHub und GitLab

**Wenn mehrere Entwickler zusammenarbeiten, sind eine Versionsverwaltung und eine Code-Hosting-Plattform fast unausweichlich. Für Nicht-Entwickler, die dort einfach nur etwas herunterladen oder sich informieren wollen, wirken GitHub & Co. zunächst rätselhaft – warum zum Beispiel werden die Downloads so gut versteckt? Eine Einführung für alle, die (noch) nicht selbst programmieren und sich immer wieder auf der Plattform verirren.**

Von Jan Mahn

**D**as gesamte Projekt finden Sie bei GitHub – ein solcher Satz steht in vielen Anleitungen zu Software- und Hardwareprojekten, auch in dieser Ausgabe der c't. Eingeweihte, die selbst Software entwickeln, sehen dann schon vor ihrem geistigen Auge, was sie erwartet: Eine Online-Ablage mit Ordnern voller Code, weiteren Materialien, manchmal auch eine Dokumentation oder fertige Softwarepakete zum Download.

Für nicht Eingeweihte, also alle, die nicht regelmäßig oder beruflich Software im Team mit anderen entwickeln, ist das alles nicht so selbsterklärend und GitHub ein Buch mit sieben Siegeln. Die gleichen Probleme haben sie auch mit GitLab, Gitea und anderen Code-Ablageorte im Internet. Dieser Artikel erklärt die Grundlagen sogenannter Git-Hoster. GitHub steht im Fokus dieses Artikels, weil die Plattform, die mittlerweile Microsoft gehört, mit Ab-

stand den größten Marktanteil hat. Hat man einmal verstanden, warum Code-Hoster so aufgebaut sind, wie sie es sind, findet man sich künftig schnell in fremden Projekten zurecht, auch wenn man gar nicht selbst ins Programmiergeschäft einsteigen will.

### Was ist Git?

GitHub, GitLab und Gitea, drei große Vertreter der Gattung der Code-Hoster, tragen die Silbe Git im Namen – das ist eigentlich wenig schmeichelhaft, denn Git bedeutet in der englischen Umgangssprache in etwa Depp, Dummkopf oder Dödel. In die IT-Welt kam der Name Git dank des Humors von Linux-Begründer Linus Torvalds. Dem wurde stets vorgeworfen, er sei selbstverliebt, weil er sein bekanntestes Projekt Linux nach sich benannt hatte, und so betonte er später ironisch, er habe auch sein zweites großes Projekt nach sich benannt. Git ist eine dezentrale Versionsverwaltung und Torvalds hatte das Projekt begonnen, weil er mit keiner existierenden Versionsverwaltung zufrieden war, da sie in seinen Augen alle nicht den Anforderungen der Linux-Kernelentwicklung genügten.

Git ist ein kleines Programm, das man vorwiegend über die Kommandozeile bedient, das sich aber auch bequem in Entwicklungsumgebungen einbinden lässt. In

vielen Linux-Distributionen ist es bereits vorinstalliert, unter macOS greift man am besten zum Paketmanager Homebrew und installiert Git flugs mit dem Befehl `brew install git`. Seit ein paar Jahren kommen auch Windows-Nutzer mit WinGet in den Genuss eines Paketmanagers und können Software bequem installieren. Der Befehl lautet:

```
winget install --id Git.Git -e ↵
↳--source winget
```

Wer sich nicht für Paketmanager begeistern kann, lädt Git über [git-scm.com/downloads](https://git-scm.com/downloads) herunter, installiert es per Hand und kümmert sich auch per Hand um Updates.

Die gute Nachricht für Einsteiger: Sie brauchen Git nicht zwangsläufig, um Inhalte von GitHub zu nutzen. Ein Grundverständnis, was die Versionsverwaltung Git tut, hilft aber beim Verständnis der Plattform ungemein. Daher vorab eine sehr knappe Git-Einführung.

## Abgezweigt

Der Befehl `git init` macht aus dem Ordner, in dem man sich auf der Kommandozeile gerade befindet, ein sogenanntes Git-Repository. Schaut man sich einen mit `git init` präparierten Ordner mit dem Dateiexplorer des Betriebssystems an, sieht man davon in der Regel: nichts. Git legt für Daten, die das Repository verwalten, einen Ordner namens `.git` an, den Dateiexplorer und auch der Befehl `ls` erstmal ausblenden (auf einer Unix-Kommandozeile verrät `ls -a` seine Existenz). Per Hand müssen und sollten Sie diesen Ordner auch nie bearbeiten.

Wenn Sie wissen wollen, ob ein Ordner ein Git-Repository ist, nutzen Sie lieber `git status`. Dieser Befehl verrät in einem leeren Repository zuerst, auf welchem Branch Sie sich befinden. „On branch main“ steht dort in einem neuen Ordner. Branches (zu Deutsch Zweige) sind der Kern der Git-Magie.

Die Grundidee der Verzweigungen: Bevor man an einer bestehenden Codebasis etwas ändert, legt man eine Verzweigung an und arbeitet dann in dieser. Anschließend analysiert Git, welche Dateien und welche Zeilen darin verändert werden. Ist man fertig mit der Entwicklung, vereinigt man die Änderungen wieder mit dem Main-Branch – das heißt in der Git-Terminologie Merge. Währenddessen können schon andere Entwickler eigene Verzwei-

gungen angelegt und an anderen Aufgaben gearbeitet haben. Im einfachen Fall haben sie ganz andere Dateien editiert, dann ist der Merge einfach. Haben sich im Ziel-Branch dieselben Dateien zwischendurch geändert, muss man beim Merge die Konflikte lösen und Git zeilenweise mitteilen, welche Zeilen man nutzen möchte. Das kann auch mal etwas mühsam sein und ist nur durch Absprachen zu verhindern.

Im Schnelldurchlauf läuft eine typische Entwicklung mit Branches wie folgt ab. Die folgenden Schritte können Sie mit einem leeren Repository (`git init`) leicht nachspielen.

Zuerst legen Sie einen Branch an, der im Beispiel „feature-test“ heißen soll:

```
git branch feature-test
```

Auf diesen Branch wechseln Sie mit:

```
git checkout feature-test
```

Ab jetzt geschehen alle Änderungen auf diesem Zweig und der Main-Branch wird nicht mehr berührt. Legen Sie zum Test eine Datei wie „test.txt“ an, zum Beispiel mit `touch test.txt` oder in einer IDE oder mit einem Texteditor.

Immer wenn Sie mit einer Änderung so zufrieden sind, dass Sie den Zwischenstand sichern oder für einen Merge vorbereiten wollen, führen Sie einen sogenannten Commit aus. Was einmal committed ist, kommt so schnell nicht weg. Alle Commits werden in der Git-Historie zusammen mit einer Commit-Message gespeichert, in der man beschreibt, was man da programmiert hat. Häufigere Commits erleichtern es später, Schritte nachzuvollziehen. Einen Commit für alle geänderten Dateien inklusive Commit-Message erzeugen Sie mit:

```
git commit -a -m "Created a text file"
```

Damit ist Ihre Arbeit auf dem Branch `feature-test` erledigt. Die Änderung (die neue Datei) ist durch den Commit dort jetzt verewigt. Wechseln Sie auf den Branch `main` zurück:

```
git checkout main
```

Wenn Sie jetzt in den Ordner schauen, werden Sie feststellen, dass die Datei `test.txt` verschollen ist – aber keine Angst, Git hat sie aufbewahrt. Sie sehen sie nicht, weil sie einen anderen Branch ausgecheckt

## c't kompakt

- GitHub verwaltet Repositories mit Code, organisiert die Zusammenarbeit von Entwicklungsteams und ist die Schnittstelle zu Nutzern und der Community für Open-Source-Plattformen.
- Nicht alles auf dieser Plattform ist selbsterklärend, Grundwissen über die zugrundeliegende Versionsverwaltung Git hilft ungemein.
- Der Funktionsumfang von GitHub und seinen Mitbewerbern wächst stetig, weshalb sich Einsteiger leicht verlaufen können.

haben. Git legt immer den Stand des aktuell ausgecheckten Branches in den Ordner. Um die Änderungen aus `feature-test` in `main` zu integrieren, führen Sie folgenden Befehl aus:

```
git merge feature-test
```

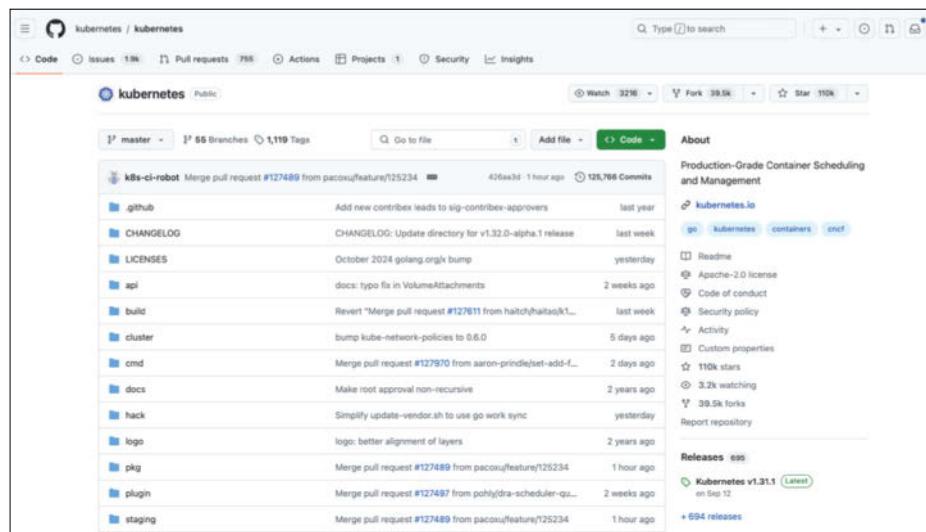
Tada – da ist die Datei `test.txt`, weil der andere Branch jetzt mit `main` verschmolzen wurde. Der Entwicklungszweig hat damit ausgedient, Sie können ihn direkt löschen:

```
git branch -d feature-test
```

## Zentrale fürs Dezentrale

Die Versionsverwaltung Git an sich ist schon praktisch und zunächst einmal extrem dezentral. Theoretisch könnten sich Entwickler den Ordner jetzt per Mail hin- und herschicken oder ihn auf einem Netzlaufwerk in der Firma lagern; das sind aber ausdrücklich keine Empfehlungen für die Praxis.

So richtig komfortabel wird Git erst, wenn man eine zentrale Ablage schafft, von der man vor der Entwicklungsarbeit den aktuellen Stand beziehen kann und auf der man später seinen Branch wieder per Merge einspielt. Genau das leisten Plattformen wie GitHub oder GitLab: Sie speichern einerseits das Repository, zeigen es in einer Weboberfläche als Ordneransicht an und machen auch Dinge auf der Website sichtbar, die in einem Repository so anfallen, also unter anderem Commits, Branches und Merges.



**Auf den ersten Blick unübersichtlich, für Entwickler gewohnt: In der Mitte zeigt GitHub die Dateien und Ordner, darunter die README-Datei. Wer fertige Binärdateien herunterladen will, findet sie in der Regel rechts unter „Releases“. Oben kann man „Issues“, „Pull Requests“ und in manchen Projekten „Discussions“ abrufen. Den „Branch“ wechselt das Aufklappmenü oben links.**

Als Anschauungsobjekt, in dem auch richtig was los ist, soll das zweitgrößte Open-Source-Projekt nach dem Linux-Kernel dienen: Am Container-Orchestrator Kubernetes haben über 3700 Menschen auf GitHub mitgearbeitet. Sie finden das Repository unter der Adresse [github.com/kubernetes/kubernetes](https://github.com/kubernetes/kubernetes).

Auf der Startseite des Repositories sehen Sie zentral die Ordneransicht mit allen Dateien darin, die zum Projekt gehören. Unter dieser Ordneransicht (bei dem großen Projekt muss man etwas scrollen) finden Sie eine Beschreibung, worum es in diesem Repository geht. Diesen Text erzeugt man als Besitzer eines Repos, indem man eine Datei „README.md“ erzeugt und die Inhalte im Format Markdown schreibt. GitHub (und auch GitLab) zeigen sie dann automatisch formatiert als Beschreibung an.

Oben links über der Ordnerübersicht finden Sie ein kleines Dropdown-Menü, in dem im Kubernetes-Repository „master“ vorausgewählt ist. Das entspricht dem, was bei neu angelegten Repos heute „main“ heißt. Sie sehen also aktuell den Hauptzweig des Repos. Wenn Sie im Dropdown einen anderen Branch auswählen, passiert das, was Sie auf der Kommandozeile als `git checkout <Branch>` kennengelernt haben.

## Auf den Rechner

In der Ordnerstruktur auf GitHub muss man in der Regel nur selten herumstöbern,

höchstens um sich mal einen Überblick zu verschaffen, was das Projekt so macht. Viele GitHub-Neulinge landen auf der Seite des Repositories und wollen die Inhalte lediglich herunterladen, zum Beispiel den Code für ein Bastelprojekt. Das geht auf zwei Arten: Oben rechts finden Sie eine grün hinterlegte Schaltfläche, die mit „Code“ beschriftet ist. Ein kleines Menü klappt aus, das mehrere Optionen enthält. Ganz unten finden Sie „Download ZIP“ – Sie bekommen den Inhalt des gerade ausgewählten Branches als ZIP-Datei und können damit lokal anstellen, was Sie mögen. Das kann manchmal schon reichen, wenn Sie nur schnell etwas ausprobieren wollen oder die Inhalte nur einmalig brauchen. Denken Sie aber daran: Wenn sich das Projekt weiterentwickelt, können Sie eine heruntergeladene ZIP-Datei nicht komfortabel auf den neuesten Stand bringen.

Das geht besser, indem Sie Git nutzen. Das Kommandozeilenprogramm kann nicht nur wie bereits gezeigt lokale Ordner verwalten, es lädt auch Repositories über HTTPS herunter. Die dafür nötige Adresse finden Sie ebenfalls im mit „Code“ beschrifteten Menü oben rechts. Navigieren Sie auf Ihrer Kommandozeile in einen Ordner, der selbst kein Repository ist. Am besten legen Sie einen Ordner als Sammelbecken für Ihre Codeprojekte an. Zum Herunterladen stellen Sie der Adresse, die GitHub angezeigt hat, den Befehl `git clone` voran.

Ausgerechnet den gesamten Kubernetes-Code wollen Sie nicht unbedingt herunterladen, weil das Projekt sehr groß ist – in dieser Ausgabe der c't finden Sie reichlich Beispiele für kleine Projekte, die Sie von GitHub herunterladen können. Zum Beispiel die Python-Bibliothek HT16K33, die im Artikel auf Seite 118 zur Anwendung kommt:

```
git clone https://github.com/jemcconville/HT16K33.git
```

Git legt den Ordner HT16K33 an und darin die Inhalte des Hauptzweigs ab, oder in Git-Sprache: Der Branch master wird direkt ausgecheckt. Mit `git checkout` können Sie jetzt lokal auf andere Zweige wechseln. Beim Herunterladen hat Git auch nicht vergessen, wo die Änderungen herkommen, es speichert vielmehr die Origin (Herkunft) ab. Führen Sie in einem ausgecheckten Repository `git remote show origin` aus, um zu sehen, aus welchem GitHub-Repository die Inhalte stammen. Weil Git die Verknüpfung kennt, können Sie später jederzeit den aktuellen Stand des Reps aus dem Internet herunterladen. Dafür reicht ein `git pull`.

## Rund um Git

Zurück von der Kommandozeile in die Weboberfläche, denn GitHub hat noch ein paar Funktionen um Git-Repositories herumgestrickt, die man kennen sollte. Die wichtigste für alle, die mit Git und dem Code so wenig wie möglich zu tun haben wollen und zum Beispiel nur eine komplizierte ausführbare Datei herunterladen wollen, sind die Releases.

Manche Projekte betreiben keine eigene Website und fordern die Nutzer auf, das Programm auf GitHub herunterzuladen. Leichter gesagt, als für Neulinge getan. Meistens werden sie dann fündig, indem sie rechts auf „Releases“ klicken. Als Inhaber eines Repositories kann man jederzeit den aktuellen Stand des Codes als Release kennzeichnen und daran weitere Dateien anhängen, zum Beispiel die kompilierte Version. Die Releases zeigt GitHub in absteigender Reihenfolge an, ganz oben findet man manchmal Alpha- und Beta-Versionen und die neueste stabile Version folgt darunter.

Zwei weitere Ecken sollten Sie unbedingt kennen: zunächst den Reiter „Issues“ (Probleme) über der Codeansicht. Dort treffen sich Entwickler und Nutzer eines Projekts, um Fehler zu besprechen.

Wenn Ihnen bei der Nutzung einer Software etwas komisch oder fehlerhaft vorkommt, schauen Sie einmal in den Issues vorbei und nutzen dort die Suche – entfernen Sie am besten den Filter auf ausschließlich offene Issues (indem Sie `is:open` aus der Suchleiste löschen), manchmal findet man den entscheidenden Hinweis auf eine Fehlbedienung in geschlossenen alten Issues.

Wenn Sie ein Problem finden, das noch kein anderer angesprochen hat, können Sie hier auch selbst ein Issue anlegen. Das ist die erste Aktion, für die Sie selbst einen Account bei GitHub brauchen (der ist in der Grundausstattung kostenlos). Vergleichsweise neu ist die Issues-Alternative „Discussions“, die Sie im Menü rechts daneben finden. GitHub hat damit auf die Tatsache reagiert, dass in Issues bisweilen sehr ausgiebig über neue Funktionen und Ideen diskutiert wird, auch wenn es sich gar nicht um Probleme im engeren Sinne handelt. Discussions sind ein Forum für allgemeinen Austausch rund um die Software. Noch nutzt aber längst nicht jedes Projekt dieses Angebots.

## Selber machen

Erfahrungsgemäß kommt irgendwann der Punkt, an dem man eine Software nicht nur nutzen, sondern auch etwas zu ihr beitragen will. Das muss nicht gleich eine große neue Funktion sein, die Karriere als sogenannter Contributor beginnt meist klein mit einem Tippfehler im Readme, den man gern beseitigen möchte. Dafür braucht man zwingend einen GitHub-Account, den man mithilfe der Schaltfläche „Sign Up“ oben rechts anlegt.

Mit dem bisher Gelernten (Branch, Commit und Merge) allein können Sie an einem fremden Repository nichts ausrichten, weil Sie keine Schreibrechte auf ein fremdes Repository haben. Open Source heißt nicht (wie so mancher annimmt), dass jeder in einem Repo schreiben kann.

GitHub handhabt das anders: Sie erzeugen zunächst einen sogenannten Fork eines Repos, kopieren es also komplett und nehmen dort die Änderungen vor. Sind Sie damit fertig, commiten Sie die Änderungen in Ihrer Kopie in einem eigenen Branch und stellen im Anschluss einen sogenannten Pull Request (GitLab nennt es sachlich korrekter Merge Request). Ein solcher Request sagt „Ich würde gern den Stand aus meinem geforkten Branch in den Main-Branch eures Projekts einbringen.“ Diesen Request bekommt der Inhaber des

Repositories zu sehen und muss ihn freigeben. Vorher wird er die Änderungen in der Regel ausprobieren oder mehr Informationen fordern. Die Höhe, was ins Repo kommt, liegt immer beim Besitzer oder bei weiteren Nutzern, die er mit Rechten ausgestattet hat. Wie solche Pull Requests in großen Projekten aussehen, können Sie sich leicht ansehen, indem Sie im Kubernetes-Repository oben das Menü „Pull Requests“ öffnen und sich einmal eine Diskussion ansehen.

Wenn Ihnen der Umweg über einen Fork, Commit und Pull Request für eine kleine Fehlerkorrektur doch arg umständlich vorkommt: Sofern Sie nur eine Datei ändern wollen, hat GitHub eine Abkürzung eingebaut und macht diese Schritte bequem im Hintergrund: Klicken Sie in einer Datei oben rechts auf die Schaltfläche mit dem Stift-Symbol, reparieren die Datei und folgen dem Dialog, der sie schnell zu einem fertigen Pull Request führt.

## Magie

Zum Schluss ein Punkt. Wortwörtlich, denn eine mächtige Funktion versteckt GitHub hinter dem Punkt auf der Tastatur. Wenn Sie mit einem GitHub-Account angemeldet sind und ein Repository im Browser geöffnet haben, öffnet sich mit diesem Tastendruck eine vollständige Entwicklungsumgebung (IDE), genauer eine Instanz von Visual Studio Code (VSC, mehr über diese IDE lesen Sie ab S. 84). Dabei wird nichts heruntergeladen, die gesamte IDE läuft im

Browser, weil sie ohnehin mit Webtechnik programmiert wurde.

In der IDE wird direkt das gerade geöffnete Repo ausgecheckt – das ist vor allem dann praktisch, wenn man mal schnell, unterwegs oder auf einem fremden Rechner eine Änderung vornehmen will und auf Funktionen wie Syntax-Highlighting nicht verzichten möchte. In dieser Oberfläche können Sie sehr komfortabel größere Änderungen, die aus mehreren Dateien bestehen, vorbereiten und zu einem Pull Request verschnüren. In der Menüleiste links sind die Git- und GitHub-Funktionen eingebunden, Sie können also direkt in der Browser-IDE Branches wechseln und Pull Requests verwalten.

## Mehr entdecken

Wenn Sie sich als beobachtender Nutzer bei GitHub zurechtfinden, können Sie schrittweise tiefer einsteigen. Probieren Sie zuerst einmal, ein eigenes Repository anzulegen (Account vorausgesetzt). Das gelingt über die Schaltfläche mit dem Plus oben rechts. In weiteren Schritten sollten Sie dann eine IDE (wie VSC) mit Git- und GitHub-Integration einrichten (dann müssen Sie auch nicht für jeden Arbeitsschritt zur Kommandozeile greifen). Und dann gibt es bei GitHub noch mehr zu entdecken: GitHub Actions zum Beispiel, die CI/CD-Umgebung von GitHub, mit der Sie Ihren Code zum Beispiel automatisiert kompilieren.

(jam@ct.de) ct

**Dokumentationen:** [ct.de/yuu](https://ct.de/yuu)

```

#!/bin/bash
# Copyright 2020 The Kubernetes Authors.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
# These tests enforce behavior of kube-addon-manager functions against a real
# cluster. A working Kubernetes cluster must be set up with kubectl configured.
# To run with the released version of kubectl, use make test.
set -o errexit
set -o pipefail
set -o nounset
#
# Default kubectl to the test users installation if needed.
KUBECONFIG="$HOME/.kube/config"
#
# Disabling pathcheck following files as the full path would be required.
# shellcheck disable=SC2003
source "kube-addons.sh"
TEST_MNG="kube-addon-manager-test"
function retry() {
    local tries=0
    while [ "$tries" -lt "$1" ]; do
        if [ "$2" -eq 0 ]; then
            sleep 1
        else
            sleep $2
        fi
        tries=$((tries+1))
    done
}

```

**Kein Download nötig. Mit dem Punkt auf der Tastatur öffnet GitHub eine Instanz von Visual Studio Code, die komplett im Browser läuft.**



Gareth Halfacree

**Das offizielle Raspberry Pi-Handbuch für Einsteiger**

Wie du deinen neuen Computer benutzt

Raspberry Pi Press,  
Cambridge (UK) 2023 (5. Aufl.),  
nichtkommerzielle Kopiererlaubnis  
(CC BY-NC-SA 3.0)  
ISBN 978-1912047369  
304 Seiten, 22-28 €  
(Epub-/Kindle-E-Book: 13-16 €,  
PDF per Bookshelf unter Raspberry Pi OS  
kostenlos, 4. Aufl. als Gratis-Download etwa  
bei CDN-Reichelt: <https://t1p.de/7imj5>)

## Was Offizielles

**Dem Raspberry Pi liegt klassischerweise kein Handbuch bei. Allerdings veröffentlicht die dahinter stehende Stiftung seit 2018 einen eigenen Einsteigerleitfaden, der inzwischen in fünfter Auflage vorliegt.**

Das von Gareth Halfacree verfasste Begleitwerk ist in fünf Sprachen erhältlich, darunter Deutsch. Wer auf die neueste Fassung Wert legt, muss sie entweder kaufen oder in PDF-Form über die Bookshelf-Anwendung von Raspberry Pi OS herunterladen. Ältere Auflagen stehen als PDF-Dateien kostenlos im Web zur Verfügung.

Der aktuelle Leitfaden passt grundsätzlich zu allen Raspi-Modellen, aber im Blickpunkt liegen die gegenwärtigen Baureihen. Der Autor stellt Pi 4 Model B, Pi 5, Pi 400 sowie Pi Zero 2 W ausführlich vor. Er beschreibt dem Raspi-Novizen, wie er das Betriebssystem einrichtet, und hilft ihm bei den ersten Schritten mit dem Dateimanager sowie dem Webbrowser Chromium. Außerdem zeigt er, wie man weitere Software installiert. Schnell sind Programmier- und Elektronikexperimente dran. Halfacree nutzt die grafische Programmiersprache Scratch 3, um drei kleine Spiele zu verwirklichen. Auch anschließend, wenn es um Python und das Thonny-IDE geht, halten spielerische Aspekte die Leserschaft bei der Stange: Der Autor zeichnet Schneeflocken mithilfe von Turtle-Grafik und verführt Raspi-Anwender zu einem digitalen Halloween-Streich. Beim Programmieren eines Textadventures kommen Schleifen, Variablen und Funktionen zum Zuge – ab diesem Abschnitt wird der Stoff spürbar komplexer.

Mittels Scratch und Python setzt Halfacree dann kleine Elektronikprojekte mit LEDs, Widerständen, Drucktastern und einem Summer um. Unter anderem entstehen dabei eine Ampelschaltung und ein Reaktionsspielchen für zwei Teilnehmer. Besitzer einer Sense-HAT-Zusatzplatine lernen, einen einfachen Tricorder als kleine Star-Trek-Hommage zu bauen. Wer eines der offiziellen Kamera-Module hat, kann damit nach der Lektüre unter anderem Zeitrafferaufnahmen anfertigen. Ein eigenes Kapitel behandelt die Programmierung des Raspberry Pi Pico und des Pico W mit MicroPython.

Mit detaillierten Schritt-für-Schritt-Anleitungen und vielen Screenshots empfiehlt sich das Buch nicht zuletzt für junge Leser. Die Übersetzung ist gut gelungen und hat sogar vor den Texten in Programmen nicht Halt gemacht. (Maik Schmidt/psz@ct.de)

## Tipp-Trommelfeuer

**Falls die Generation TikTok sich ein Raspi-Buch wünscht, sieht es bestimmt so aus wie Christian Immers „255 Anleitungen“. Die knallbunt gestaltete Tipp-Fundgrube eröffnet ein riesiges Themenspektrum und erfordert keine lange Aufmerksamkeitsspanne.**

Wer sich nicht auf tiefgehende Wissensvermittlung einlassen, sondern stöbern, entdecken und sich an kurzen Anregungen erfreuen will, ist bei Immers Buch genau richtig. Es bündelt 255 kurze illustrierte Anleitungen, in denen es darum geht, kleine Probleme rund um den Raspi zu lösen und kleine Projekte damit auszuprobieren. Im Mittelpunkt stehen Raspberry Pi 4 und 400, aber viele der Anregungen lassen sich ebenso gut mit anderen Modellen umsetzen.

Der Autor hat sein Material in die Kategorien Betriebssystem, Praxistipps, Software, Hardware, Programmierung und Elektronik eingeteilt. Die Bereiche sind farblich kodiert; überhaupt wirkt das Layout des Buchs mit seinen vielen Abbildungen einladend und übersichtlich. Die meisten Tipps sind sehr kurz gehalten und richten sich an Einsteiger. So erklärt der Autor etwa den Umgang mit USB-Sticks oder wie sich MP3-Dateien mittels des Dateimanagers abspielen lassen. Viele Ratschläge klingen banal, aber jeder kann irgendwie für einen Anwendungszweck und Anwendertyp nützlich sein. Viel Raum widmet Immmer der Multimedialwelt. Er behandelt etwa das Mediencenter Kodi und widmet sich verbreiteten Spielen. Freunde von Minecraft, von Point-and-Click-Adventures und von Spielesklassikern auf emulierten Konsolen kommen auf ihre Kosten.

Die Abschnitte zur Programmierung präsentieren häppchenweise etwa grundlegende Eigenschaften von Python und Scratch. Sie sind für Referenzzwecke hilfreich, können aber keinen Programmierkurs ersetzen. Die Kategorie Elektronik hält allerlei Projekte bereit, die dank der guten Anleitungen selbst ein Anfänger nachbauen und ausprobieren kann. Wer allerdings wissen will, wie und warum das Gebastelte eigentlich funktioniert, braucht ergänzende Grundlagenlektüre.

Manche der Tipps sind lediglich Denkanstöße. So kann die magere Seite über das Bildbearbeitungsprogramm GIMP nicht viel mehr tun, als lediglich darauf aufmerksam zu machen, dass es eine solche Software überhaupt gibt. Für Leser, die noch keine genaue Vorstellung davon haben, was sie mit ihrem Raspberry Pi so alles anstellen können, ist das Buch perfekt. Die vielen kleinen Informationshäppchen laden dazu ein, es immer wieder für ein paar Minuten zur Hand zu nehmen. (Maik Schmidt/psz@ct.de)



Christian Immmer

**255 Anleitungen für Raspberry Pi 4 und 400**

Für Einsteiger und Fortgeschrittene

Markt+Technik, Burgthann 2023  
ISBN 978-3988100092  
288 Seiten, 20 €  
(PDF-E-Book: 15 €)



Michael Weigend  
**Raspberry Pi programmieren mit Python**  
mitp, Frechen 2024 (6. Aufl.)  
ISBN 978-3747508299  
408 Seiten, 30 €  
(as Bundle mit E-Book beim Verlag:  
35 €;  
PDF-/Epub-/Kindle-E-Book allein: 30 €)

## Python auf dem Pi

**Michael Weigends didaktisch orientiertes Programmierbuch führt den Raspi gewissermaßen zu seinen Wurzeln. Anhand spannender Praxisprojekte lernen Leser, mit Python zu arbeiten.**

Der Raspberry Pi sollte ursprünglich vor allem eine Programmierplattform für Schüler sein. In diesem Konzept spielt die Sprache Python eine zentrale Rolle, die von Beginn an bestens ins Raspi-Ökosystem integriert war. Michael Weigend, selbst früher Lehrer für Naturwissenschaften, führt Python-Neulinge von einem praktischen Erfolgserlebnis zum nächsten. Sein Buch, für dessen Lektüre man keinerlei Programmierkenntnisse mitzubringen braucht, erscheint bereits in der sechsten Auflage.

Der Autor nutzt viele kleine Projekte, um zunächst die Grundlagen und dann immer weitere Python-Eigenschaften zuveranschaulichen. Während das Buch kein programmierbezogenes Vorwissen erfordert, sind einige Elektronikprojekte basteltechnisch durchaus ambitioniert. Zwar baut Weigend unter anderem einen pragmatischen Tastschalter aus zwei Reißzwecken und einer Wäscheklammer. Für andere Aufgaben braucht man aber anspruchsvollere Bauteile, die nicht in jeder Küchenschublade herumliegen, angefangen bei LED-Streifen, Transistoren und Relais. Der Leser muss auch zum Lötkolben greifen. Ein paar der Basteleien verwenden sogar kostspielige Teile wie einen Kohlendioxid-Sensor für 65 Euro.

Die Bastelprojekte sind durchweg lehrreich und haben gleichzeitig fast alle hohen Unterhaltungswert. So bildet der Autor das Spiel Pong nach und entwickelt einen Fußschalter zur Steuerung einer Kunstinstallation im Stil Gerhard Richters. Das Kamera-Modul nutzt er unter anderem, um Bewegungen zu erkennen. Auch die reinen Softwareprojekte sind motivierend. Zu Beginn liefert Weigend lediglich Codeschnipsel, aber schnell programmiert er eine umfangreiche Karteikartenanwendung und einen Staumelder, der auf Web Scraping beruht. Erfreulicherweise gehören auch einige grafische Anwendungen auf Grundlage von Tkinter zur Projektsammlung. Bei manchem Einsteiger dürfte dergleichen die Lust auf mehr wecken.

Den Sprachumfang von Python deckt das Buch sehr weitgehend ab. Außer Funktionen und den wichtigsten Collection-Klassen lernen Leser auch die objektorientierte Programmierung kennen. Selbst auf Feinheiten wie Callbacks und das Überladen von Operatoren geht Weigend ein. Die Übungsaufgaben am Ende eines jeden Kapitels helfen, die gelegten Fundamente zu festigen.  
(Maik Schmidt/psz@ct.de)

## Schwergewicht

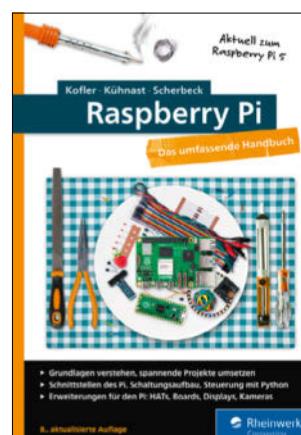
**In seiner achten Auflage wagt das Standardwerk von Kofler und Kollegen auf mittlerweile über 1000 Seiten das Kunststück, einen Raspi-Rundumschlag für Einsteiger und Fortgeschrittene zu vollziehen.**

Den Raspberry Pi gibt es seit zwölf Jahren. In dieser Zeit ist rund um den Zwergechner viel passiert – sowohl in puncto Software-pakete als auch bei den Hardware-Erweiterungen. Michael Kofler, Charly Kühnast und Christoph Scherbeck haben es sich zur Aufgabe gemacht, die wichtigsten Werkzeuge und Techniken abzudecken.

Die ersten 250 Seiten vermitteln alles, was ein Raspi-Neuling wissen sollte. Die Einsteigerhinweise spannen einen Bogen vom Kauf über die Inbetriebnahme bis zu den Grundlagen von Linux. Detailliert erklären die Autoren, wie man das Betriebssystem konfiguriert und den Raspi ins heimische Netzwerk integriert. Im Aufbauteil zeigen sie nicht nur, was mit dem Einplatinencomputer möglich ist, sondern geben ihren Lesern geeignetes Handwerkszeug mit auf den Weg, damit diese eigene Ideen umsetzen können. Beispielsweise enthält das Buch einen umfangreichen Elektronik-Crashkurs, der unter anderem erklärt, wie man viele externe Bauteile und Sensoren mit dem Pi verwendet.

Dem Programmieren widmen die Autoren ebenfalls viel Raum; ausführlich geht es um Python und das Shell-Scripting mit Bash. Bei C und PHP beschränken sich Kofler und Kollegen darauf, knapp zu beschreiben, wie man diese Sprachen auf dem Raspi verwendet. Auch Mathematica und die Wolfram Language kommen kurz zum Zuge. Ansonsten können Leser sich über eine Unmenge an Soft- und Hardwareprojekten sehr unterschiedlicher Art freuen. Einiges, wie der Wassermelder mit Smartphone-Alarm, fällt in den Bereich Heimautomatisierung. Andere Projekte zeigen, wie man den Raspi ohne viel Zubehör unter anderem in einen Audioplayer, einen Werbeblocker, ein NAS oder einen WLAN-Router verwandelt.

Bei allem fällt die Liebe der Autoren zum Detail auf. So ist es beim Audioplayer entscheidend, das richtige Raspi-Modell zu wählen. Geeignete Erweiterungen ermöglichen guten Klang und mehrere Playersoftware kommt unterschiedlichen Vorlieben entgegen. Ähnlich gründlich geht das Autorenteam auch bei den anderen Projekten vor. Dieses „umfassende Handbuch“ bietet tatsächlich für jeden Raspi-Anwender etwas und seine ansprechende Aufmachung mit vielen farbigen Abbildungen kann manchem die Scheu vor einem so dicken Wälzer nehmen. (Maik Schmidt/psz@ct.de)



Michael Kofler, Charly Kühnast,  
Christoph Scherbeck  
**Raspberry Pi**  
Das umfassende Handbuch  
Rheinwerk Computing, Bonn 2024  
(8. Aufl., der Buchverlag gehört wie c't zu heise medien)  
ISBN 978-3836296663  
1045 Seiten, 45 €  
(as Bundle mit E-Book beim Verlag:  
50 €;  
PDF-/Epub-/Kindle-E-Book allein: 45 €)

# Schaltungen zusammenklicken

[falstad.com/circuit/circuitjs.html](http://falstad.com/circuit/circuitjs.html)

[easyeda.com](http://easyeda.com)

Vom einfachen Schwingkreis bis zur Pong-Simulation: Schaltungen lassen sich mit dem kostenlosen **CircuitJS** bequem im Browser gestalten und ausprobieren. Um eine neue Schaltung zu entwerfen, wählen Sie über das Menü „Zeichnen“ oder einen Rechtsklick Bauteile aus und fügen sie auf der Arbeitsfläche ein. Der Simulator stellt Dutzende Basiskomponenten bereit, analogen wie Spulen oder Schalter und digitale, etwa Sieben-Segment-Anzeigen.

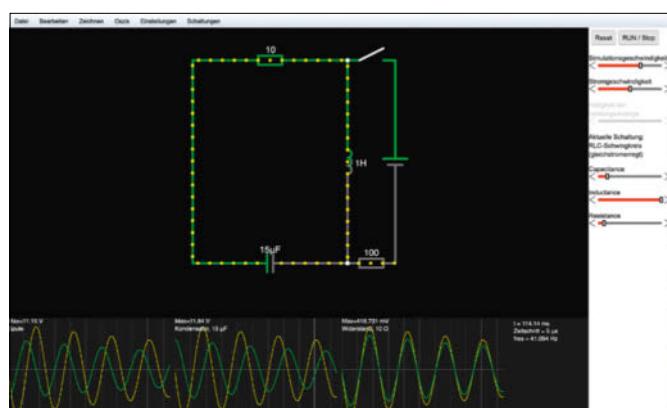
Wie Sie die Komponenten dabei miteinander verbinden und ihre Parameter editieren, erschließt sich nach ein wenig Geklicks intuitiv. Alternativ lassen sich viele Komponenten auch per Tastenkürzel einfügen.

CircuitJS eignet sich aus mehreren Gründen hervorragend für Lehrzwecke. So stellt die Anwendung eine Bibliothek mit vorgefertigten Basisschaltkreisen bereit, die sich individuell anpassen lassen – gewissermaßen das kleine Einmaleins der Elektronik.

Zudem zeigt CircuitJS auf verschiedene Weise an, was in der Schaltung vor sich geht. Der Simulator visualisiert die Ströme in den Verbindungen. Und ein Rechtsklick auf ein Bauteil zeigt wahlweise dessen Strom- oder Spannungsverlauf oder andere Werte in Echtzeit in einem Oszilloskop an.

Schaltungen lassen sich als PNG-Bild, SVG-Datei, in Textform oder als Link exportieren. Mit den letzten beiden Formaten können Sie eine Schaltung speichern und bei Bedarf im Simulator wiederherstellen. Der Quellcode von CircuitJS ist auf GitHub verfügbar. Wer mag, kann den Simulator auf seiner eigenen Homepage veröffentlichen oder eine lokale lauffähige Anwendung auf Basis von Electron herunterladen.

Geht es bei CircuitJS eher um das Lernen, eignet sich **EasyEDA** für die komplette Produktion: Elektronische Schaltungen lassen sich dort nicht nur entwerfen und testen. Die Plattform generiert auch Produktionsdaten für Leiterplatten. EasyEDA enthält wie CircuitJS eine riesige Sammlung an Bauteilen, die Benutzer in Schaltplänen verwenden können. Durch die Zusammenarbeit mit Lieferanten können Entwickler ihre Bauteile zudem einfach bestellen und ihre Leiterplatten fertigen lassen. Aufgrund der vielen Funktionen ist der Einstieg mit EasyEDA aber schwieriger als bei CircuitJS. Der Dienst ist in einer Basisversion kostenlos. (jo@ct.de)



# Reparieren statt wegwerfen

[repaircafe.org/de](http://repaircafe.org/de)

Der alte Ghettoblaster brummt, der Toaster toastet nur noch gelegentlich, der Kaffeevollautomat heizt, pumpt aber nicht mehr? Die alten Elektrogeräte landen jetzt entweder auf dem Müll – oder Sie besuchen damit mal ein **Repair Cafè**. Die Idee dieser Veranstaltungen: Man trifft sich in geselliger Runde, um gemeinsam die kaputten Dinge der Teilnehmer instand zu setzen. Wer nichts zu reparieren hat, trinkt einen Kaffee oder Tee, schaut zu und lernt ganz nebenbei ein paar Kniffe.



Repair Cafés haben also viele positive Effekte: Gegenstände leben länger und das Portemonnaie und die Natur sagen danke. Menschen mit Reparaturwissen erfahren Wertschätzung, Menschen ohne Reparatur-Know-how lernen etwas hinzu.

Das erste Repair Café fand vor 15 Jahren in Amsterdam statt, mittlerweile listet die Website Repair Café mehr als 3200 dieser Veranstaltungen weltweit, Hunderte davon in Deutschland. Die Website hilft aber nicht nur, Repair Cafés in der Nähe zu finden. Auch wer selber eines veranstalten will, findet dort Hilfestellungen. Zudem hat die Community mehr als 9000 Reparaturanleitungen gesammelt. (jo@ct.de)

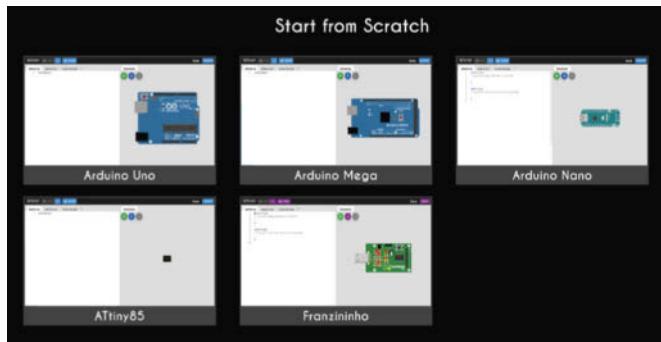
# Simulator für Arduino Uno & Co.

[wokwi.com](http://wokwi.com)

[heise.de/-8990252](http://heise.de/-8990252)

Sie haben die Idee für ein tolles Microcontroller-Projekt, aber gerade die passende Hardware nicht zur Hand? Kein Problem: Mit dem Simulator **Wokwi** planen und simulieren Sie Ihr Projekt im Browser. Der Dienst eignet sich auch, um Entwürfe zu testen, ohne durch eine Fehlbeschaltung teure Hardware zu schrotten.

Wokwi kennt etliche Mikrocontroller, darunter den Arduino Uno, Nano und Mega sowie diverse ESP32-Varianten, den Raspberry Pi Pico und Pico W. In ihren Projekten können Benutzer virtuelle Bauteile wie LEDs, Widerstände, Taster, Potenzimeter, Sensoren und viele andere elektronische Komponenten verwenden. Diese Bauteile zieht man per Drag-and-Drop in eine virtuelle Arbeitsfläche und verdrahtet sie miteinander.



Entsprechend klickt man sich die für die Ansteuerung des jeweiligen Controllers benötigten Bibliotheken zusammen. Mit einem Klick auf den Play-Button erwacht eine simulierte Schaltung schließlich zum Leben. Um ein Projekt ausgiebig zu testen, kann man Komponenten, etwa virtuelle Sensordaten, interaktiv steuern.  
(jo@ct.de)

## Vielseitige Informationsquelle

[allaboutcircuits.com](http://allaboutcircuits.com)

Wer sich für Elektronik und Elektrotechnik interessiert, der sollte die Homepage von **All About Circuits** bookmarken. Die englischsprachige Site spricht sowohl Einsteiger als auch erfahrene Elektronikbastler an. Sie enthält einen ausführlichen „Education“-Bereich mit einer Lernbibliothek. Diese umfasst eine Vielzahl an Lektionen und Kapiteln zu grundlegenden und fortgeschrittenen Konzepten in der Elektronik, einschließlich Themen wie Halbleiter, Schaltungsdesign, Digitaltechnik und vieles mehr.

Neue Bauteile, technologische Durchbrüche und Branchentrends behandelt All about Circuits in Form von News-Meldungen und tiefer gehenden Fachartikeln. Eine Fundgrube ist

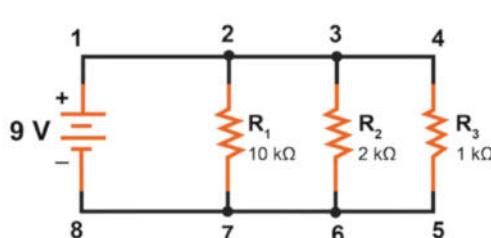


Figure 1. Parallel circuit with a battery and three resistors.

### Voltage in a Parallel Circuit

The first principle to understand about parallel circuits is that the voltage is equal across each parallel component. This is because there are only two sets of electrically common points in a parallel circuit, and the voltage measured between sets of common points must always be the same at any given time.

With that concept in mind, in the circuit of Figure 1, nodes 1, 2, 3, and 4 are the same electrical node. Likewise, nodes 5, 6, 7, and 8 are the same electrical node. Therefore, the voltage across  $R_1$  is equal to the voltage across  $R_2$ , which is equal to the voltage across  $R_3$ , and is then equal to the voltage across the battery (9 V).

auch das Forum, zum Beispiel für Tipps zu weiteren nützlichen Sites zum Thema Elektronik.  
(jo@ct.de)

## Die Maker

[heise.de/make](http://heise.de/make)

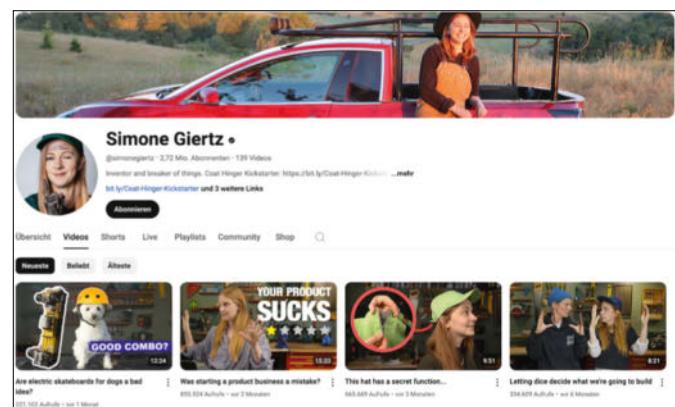
Die **Website des Make-Magazins** sollte eigentlich jeder c't-Leser kennen, der sich ein wenig für Basteleien mit und ohne Elektronik interessiert. „Die Make“ ist die Zeitschrift für die deutschsprachige Maker-Szene und eine Schwester der c't. Auf ihrer Homepage finden Sie aktuelle Testberichte, viel Know-how und detaillierte Schritt-für-Schritt-Bauanleitungen für Hardware-Projekte.  
(jo@ct.de)

## Video-Inspiration

[youtube.com/@simonegiertz](http://youtube.com/@simonegiertz)

[youtube.com/c/MarkRober](http://youtube.com/c/MarkRober)

Zugegeben: Viele der Projekte, die Maker-Stars wie **Simone Giertz** und **Mark Rober** in ihren Videokanälen vorstellen, wird Otto Freizeitbastler wohl schon aus Zeitgründen nie umsetzen können. Aber es macht Spaß, ihnen beim Maken zuzuschauen und die fertigen Schöpfungen im Einsatz zu bewundern.



Simone Giertz hat sich schon mal als „Queen of Shitty Robots“ bezeichnet und baut auch heute noch gelegentlich eingeschränkt funktionsfähige Roboter. Mittlerweile stellt sie aber auch Kleiderbügel, Stühle, Tische und vieles mehr her und berichtet sehr unterhaltsam darüber. Einen Tesla hat sie sich auch schon vorgenommen und ihn zu einem Pickup-Truck umgebaut.

Mark Rober, ein ehemaliger NASA-Ingenieur, führt in seinem Kanal Experimente durch, stellt eigene Erfindungen vor und bringt seinen Zuschauern wissenschaftliche Erkenntnisse näher. Legendar sind seine Videos der selbstentwickelten Glitterboxen, mit denen er Diebe hereinlegt, sowie seine Parcours, die den Eichhörnchen der Nachbarschaft alles abverlangen.  
(jo@ct.de)

Diese Seite mit klickbaren Links: [ct.de/ysug](http://ct.de/ysug)

# Vielseitige Großfamilie

Ein Blick auf die ESP-Familie von Espressif und darauf basierende Entwicklungsplatinen



Einführung in die Welt der ESP-Mikrocontroller .....	Seite 78
Programmier- und Entwicklungswerkzeuge für den ESP .....	Seite 84
Projekt: Eine „On-Air“-Lampe mit dem ESP .....	Seite 90
Projekt: Neues Leben für Konzertarmbänder .....	Seite 96
Projekt: Nachspeicherofen steuern .....	Seite 102

# Wer Gerätschaften entwickelt, die etwa im Smart Home Kontakt mit einem Funknetzwerk aufnehmen sollen, kommt kaum an den ESP-Mikrocontrollern der chinesischen Chipschmiede Espressif vorbei. Sie funken je nach Ausführung via WLAN, Bluetooth und Zigbee und bieten viele spannende Schnittstellen. Wir werfen einen Blick auf aktuelle ESP-Module.

Von Georg Schnurer

**S**o richtig ins Herz vieler Bastler und Entwickler spielte sich Espressif mit dem für wenige Euro erhältlichen ESP8266: Er bringt alles mit, was man benötigt, um eine Schaltung per WLAN ins Netz zu bringen. Den Kern des im August 2014 vorgestellten Mikrocontrollers bildet der vom amerikanischen Unternehmen Cadence zugelieferte CPU-Core namens Tensilica Xtensa. Dabei handelt es sich um einen sehr sparsamen, aber im Vergleich zu anderen Mikrocontrollern dennoch vergleichsweise leistungsfähigen Rechenkern.

Espressif nennt den beim ESP8266 verwendeten Prozessorkern „Xtensa L106“. Zu dem mit 160 MHz getakteten 32-Bit-Prozessor gesellt sich noch die von Espressif selbst entwickelte, in den Chip integrierte WLAN-Schnittstelle hinzu. Diese Kombination und die umfangreiche Software-Unterstützung macht den ESP zu einem kompakten und günstigen Baustein, um beinahe beliebige Schaltungen WLAN-fähig zu machen.

Besonders charmant: ESP-Chips kann man in einen „Deep Sleep“-Modus versetzen, in dem ihre Stromaufnahme im Idealfall nur noch zwischen 10 und 20 Mikroampere beträgt, kaum mehr als die Selbstentladung mancher Akkus. Abgesehen von der integrierten Echtzeituhr ist der Baustein dann tot. Das Aufwecken erfolgt über ein Reset-Signal (RST auf GND) oder aber nach einer vorgegebenen Zeit von maximal 71 Minuten durch die Echtzeituhr. Das ist ideal für batterie- oder akkubetriebene Schaltungen, die zum Beispiel in festgelegten Intervallen Messda-

ten ermitteln sollen. Aber Achtung: Längst nicht jedes ESP-Board ist im Deep-Sleep-Modus ein guter Energiesparer. Neben dem ESP-Chip sitzen ja auch noch andere Bausteine auf der Platine und manche davon sind echte Stromschlucker.

Ein ESP8266 arbeitet mit einer typischen Versorgungs- und Signalspannung von 3,3 Volt (2,5–3,6 V) und ist damit nicht kompatibel mit den meisten 5-Volt-Komponenten, etwa Sensoren. Gibt es für den vorgesehenen Einsatzzweck keine passenden 3,3-Volt-Chips, muss ein Pegelwandler her. Die typische zulässige Betriebstemperatur liegt zwischen –40 °C und +85 °C, es gibt auch Chipvarianten mit höherer Maximaltemperatur bis +125 °C. Für Module können andere Schwellen gelten, orientieren Sie sich dafür an deren Datenblättern. Die wichtigsten Eigenschaften der verschiedenen Controllertypen haben wir in der Tabelle auf Seite 83 zusammengefasst.

Neben dem WLAN-Interface, der 32-Bit-CPU und der bereits erwähnten Echtzeituhr (RTC, 128 Byte RTC-SRAM) stecken im ESP8266 noch weitere Schmankerl. Dazu gehören 160 KBit Speicher (SRAM), ein 10-Bit-Analog-Digital-Wandler (ADC), eine I2S- und 17 GPIO-Schnittstellen. Mit integriertem Flash-Speicher kann der Mikrocontroller nicht aufwarten, dieser wird üblicherweise über den ebenfalls vorhandenen I2C-Port angebunden. Hinzu kommen noch zwei Universal Asynchronous Receiver Transmitter (UART, also serielle Schnittstellen), von denen einer üblicherweise zum Programmieren des Flash-Bausteins auf den ESP-Modulen genutzt wird. Den Reigen komplettiert ein SDIO-Slave (SDI Input/Output), über den sich der Baustein wie eine SD-Karte ansprechen lässt.

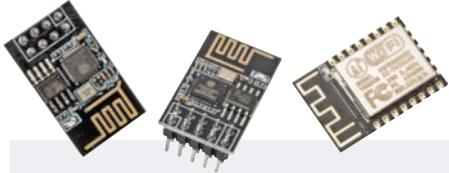
So viele Anschlüsse lassen sich nicht alle gleichzeitig aus einem Chip herausführen. Deshalb sind viele Pins des ESP8266 mehrfach belegt. Je nachdem, welche der integrierten Schnittstellen man beim Programmieren aktiviert, sind einige der GPIO-Pins bereits vorbelegt und damit nicht für andere Aufgaben nutzbar. Die aktuelle Version des ESP8266 trägt den Zusatz „EX“, ältere Chips ohne EX findet man kaum noch, auch wenn in vielen Baugruppenbeschreibungen immer noch vom ESP8266 und nicht vom ESP8266EX die Rede ist. Espressif empfiehlt im aktuellen Datenblatt des ESP8266EX, diesen Baustein nicht mehr zu verwenden, sondern auf den jüngeren ESP8684 auszuweichen, obwohl der ESP8266EX das Ende seiner Lebens- und Fertigungszeit erst 2029 erreicht. Gegen einen Wechsel spricht, dass es sich beim ESP8684 um einen Baustein mit komplett anderer Architektur handelt – doch dazu später mehr.

## ESP32

Wem die Rechenleistung des ESP8266EX nicht ausreicht, für den hält Espressif den ESP32 in sehr unterschiedlichen Varianten bereit. Der mit 240 MHz getaktete Baustein besitzt gleich zwei CPU-Kerne (Xtensa LX6) und ist deutlich leistungsfähiger als der ESP8266EX. Zudem bringt er außer WLAN (jetzt mit bis zu 150 Mbit/s) auch noch Bluetooth (BR/EDR und BT LE V4.2) mit. Auch bei den sonstigen Schnittstellen hat Espressif geklotzt und nicht gekleckert: Der ESP32 wartet mit beinahe allem auf, was man sich bei einem Mikrocontroller wünschen kann (siehe Tabelle).

## c't kompakt

- Die ESP-Familie von Espressif bietet eine reichhaltige Auswahl spannender Mikrocontroller für eigene Projekte.
- Je nach Chip-Modell haben die ESPs einen Xtensa- oder einen RISC-V-Prozessorkern.
- Neben Modulen und Entwicklungsbrettern gibt es auch viele interessante Fertig- und Halbfertig-Platinen, die den Bastelaufwand verringern.



### ESP8266EX

Schon für wenige Euro verschafft der ESP-01S (links) einer Schaltung eine WLAN-Verbindung. Das Modell ESP-01+ in der Mitte eignet sich besonders, wenn auch die I2C-Schnittstelle des ESP8266EX genutzt werden soll. Ganz rechts liegt ein ESP8266MOD. Dieses Modul bietet leichten Zugang zu allen Pins des SoC, muss aber über einen speziellen Programmieradapter mit Code bespielt werden.

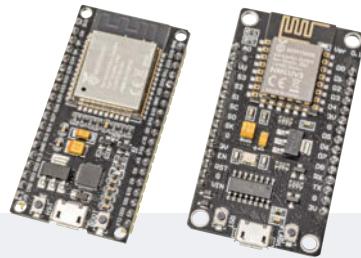
**Preise:** **ESP-01S ab 2 Euro,**  
**ESP-01+ ab 4,48 Euro,**  
**ESP8266MOD ab 2 Euro**



### ESP32-C3

Einen preisgünstigen Einstieg in die RISC-V-Welt von Espressif ermöglicht das Entwicklungsboard ESP32-C3-DevKitC-02. Es nutzt ein C3-WROOM-Modul mit 32-Bit-RISC-V-Prozessor (160 MHz), 400 KByte SRAM (davon 16 KByte Cache) sowie 384 KByte ROM. Wer mit dem ESP32-C3 entwickeln will, kann die Espressif-IDE nutzen. Für andere Entwicklungsumgebungen sind einige Kniffe erforderlich, die wir im nachfolgenden Artikel beschreiben.

**Preis:**  
**ESP32-C3-DevKitC-02 ab 10 Euro**



### NodeMCU

Module mit der Bezeichnung NodeMCU gibt es sowohl mit dem ESP8266EX als auch mit dem ESP32. Sie sind auf den ersten Blick nur schwer voneinander zu unterscheiden, bieten aber sehr unterschiedliche Performance und Schnittstellenausstattung. Zumeist haben diese Entwicklungsboards einen Mikro-USB-Anschluss, es gibt aber auch moderne Varianten mit USB-C.

**Preis:**  
**NodeMCU V3 (ESP8266) 3,90 Euro,**  
**ESP32 NodeMCU 6,80 Euro**



### ESP mit LAN-Port

Will man neben WLAN und Bluetooth auch noch einen klassischen LAN-Anschluss nutzen, bieten sich ESP32-Platinen mit ESP32-WROOM-Modul an. Es gibt sie in einfacher Ausführung etwa in Form des links abgebildeten ESP32-LAN, aber auch komplexer mit PoE-LAN-Port, über den dann gleich die Energieversorgung der Platine läuft.

**Preis:** **ESP32-LAN ab 8 Euro,**  
**Olimex ESP32-POE-ISO-EA ab 36 Euro**

und PSRAM (0 bis 2 MByte) mit, bei anderen Modellen wird der Flash-Speicher wie beim ESP8266EX via I2C angekoppelt.

Wer mit nur einem CPU-Kern sowie ohne Bluetooth auskommt, kann zum ESP32-S2 greifen. Er nutzt den moderneren Xtensa LX7-Core und bietet bis zu 43 GPIO-Pins. Der ESP32-S3 wiederum ist dessen Dual-Core-Variante und bringt wieder Bluetooth-Support (BT LE V5.0) sowie zwischen 39 und 45 GPIO-Pins mit.

### RISC-V

Den ESP32 gibt es zudem in den Modellvarianten ESP32-C2, ESP32-C6 und ESP32-H2. All diese Modelle nutzen je einen einzelnen Prozessorkern mit der quelloffenen RISC-V-Architektur. Damit schafft sich Espressif ein Stück Unabhängigkeit vom amerikanischen IP-Lieferanten Cadence. Wer die C- oder H-Modelle des ESP32 nutzen will, muss darauf achten, dass die ins Auge gefasste Entwicklungsumgebung damit umgehen kann.

Der ESP32-P4 ist Espressifs jüngstes Kind und derzeit nur zusammen mit einem sehr umfangreich ausgestatteten Entwicklerboard mit dem sperrigen Namen „ESP32-P4-Function-EV-Board“ für gut 65 Euro via AliExpress erhältlich. Er ist deutlich leistungsstärker als alle bisherigen ESP-Varianten, weil er gleich zwei RISC-V-Kerne mit bis zu 400 MHz enthält und zusätzlich noch einen besonders sparsamen mit 40 MHz. Der P4 bringt selbst

keine Funkanbindung mit, weshalb auf dem Entwicklerboard zusätzlich ein ESP32-C6 residiert, der WLAN, Bluetooth 5 (mit LE) sowie Zigbee, Thread und Matter beherrscht.

Das ESP32-P4-Function-EV-Board richtet sich vor allem an Entwickler von Smart-Home-Geräten. Es lässt sich auch per Fast Ethernet (100 Mbit/s) ins Heimnetz bringen und hat weitere Anschlüsse für USB 2.0, ein Display (MIPI DSI), eine Kamera (CSI), einen Lautsprecher sowie eine GPIO-Pfostensteckerleiste.

Mögliche Einsatzbereiche sind Haushaltsgeräte, Staubsaugerroboter, Türsprechstationen oder Bedienungseinheiten für E-Bikes und Roller. Mit seiner Kombination von Eigenschaften liegt das Espressif ESP32-P4-Function-EV-Board zwischen einfachen ESP32-Mikrocontrollern und rechenstärkeren ARM64-Einplatinencomputern, auf denen ein 64-Bit-Linux läuft, die aber auch mehr Energie benötigen.

### RISC-V-Kerne im ESP32-P4

Die beiden von Espressif selbst entwickelten „High Performance“-RISC-V-Kerne sind deutlich stärker als die bisherigen ESP32-Rechenkerne. Ihr Befehlssatz umfasst zudem Erweiterungen für Gleitkommaberechnungen (FPU), digitale Signalverarbeitung (DSP) und KI (AI), die Espressif bisher jedoch nicht detailliert dokumentiert. Auf den 32-Bit-Kernen (RV32) läuft

Je nach Ausführung bietet der ESP32 zwischen 29 und 34 GPIO-Pins; die Betriebsspannung beträgt typischerweise 3,3 Volt, kann aber auch niedriger liegen (2,3 – 3,6 V beziehungsweise 3,0 V–3,6 V). Ähnliches gilt für die zulässige Umgebungstemperatur. Einige ESP32-Varianten bringen integrierten Flash-Speicher (bis zu 8 MByte)



## D1 Mini

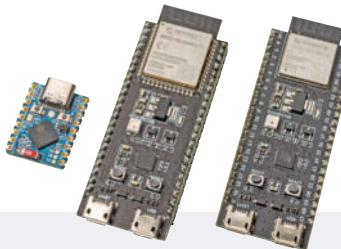
Das D1 Mini ist ein weiteres Entwicklungsboard auf Basis des ESP8266. Es bringt einen USB-Anschluss zum Programmieren mit und ermöglicht den leichten Zugriff auf alle Pins des ESP-Moduls. In der Standard-Version (D1-Mini, Mitte) hat es einen Mikro-USB-Port, die Version D1 Mini C (rechts) nutzt USB-C. Die Lite-Version (D1 Mini lite, links) gibt es in zwei Ausführungen, einmal wie abgebildet mit dem ESP8266 und einem separaten Flash-Baustein, und einmal mit dem ESP8285, eine Variante, die bereits den Flash-Speicher (1 MByte) enthält, sonst aber kompatibel zum ESP8266 ist. Eine weitere, hier nicht gezeigte Version ist der D1-Mini Pro. Er nutzt den ESP8266EX, bringt aber 16 MByte Flash und einen Antennenstecker mit.

**Preis:** D1 Mini 4,40 Euro,  
D1 Mini C 4,60 Euro,  
D1-Mini lite ab 3,90 Euro,  
D1 Mini Pro 6,80 Euro

allerdings kein RV64GC-Linux „von der Stange“. Es stehen zudem nur 768 KByte SRAM Arbeitsspeicher bereit sowie 32 Megabyte Pseudo-SRAM (PSRAM), die im Chipgehäuse integriert sind. Zusätzlich gibt es auch noch 768 KByte „L2MEM“. Der ESP32P4 hat eingebaute Decoder für H.264-Video und JPEG sowie einen „2D Pixel Processing Accelerator“.

Umfangreich ist die Ausstattung des ESP32-P4 mit Ein-/Ausgabeschnittstellen. Alle gängigen Mikrocontroller-Interfaces wie I2C, SPI, UART, GPIO, I2S sind ebenso vorhanden wie SDIO, MIPI CSI für eine Kamera, MIPI DSI für ein (mitgeliefertes) Display sowie Fast Ethernet. Außerdem gibt es USB 2.0 OTG, bei dem der Controller als USB-Gerät und nicht als USB-Host arbeitet.

Auf das ESP32-P4-Function-EV-Board lötet Espressif außer dem erwähnten ESP32-C6-Funkmodul noch 16 MByte SPI-Flash und einen MicroSD-Kartenleser. Zudem sind ein I<sup>2</sup>S-Soundchip samt



## ESP32-S3

Mit dem modernen Dual-Core-Prozessor Extensa LX7 schmücken sich verschiedene Entwicklungsboards, die derzeit direkt von Espressif gefertigt werden. Sie bringen zumeist zwei USB-C-Ports mit, unterscheiden sich aber bei der sonstigen Ausstattung. So nutzt das ESP32-S3-DevKitC-1 (Mitte) ein WROOM-2-Modul, wogegen auf der sehr ähnlich aussehenden Version ESP32-S3-DevKitM-1 (rechts) ein ESP32-S3-Mini-1-Modul sitzt. Letzteres bringt neben den üblichen Schnittstellen noch 8 MByte Flash mit, im WROOM-2 stecken hingegen 32 MByte Flash und 8 MByte PSRAM. Ganz links ist der ESP32-S3-Zero abgebildet. Die dort verlötzte S3-Variante bringt 4 MByte Flash, 2 MByte PSRAM und 512 KByte SRAM mit. Achtung: Der S3 wird aktuell nur von wenigen Entwicklungsumgebungen unterstützt.

**Preis:** ESP32-S3-DevKitC-1 129,90 Euro,  
ESP32-S3-DevKitM1 23,30 Euro,  
ESP32-S3-Zero 6,90 Euro



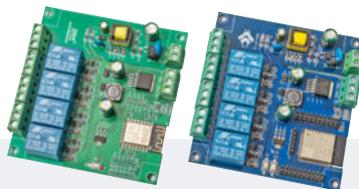
## ESP32 mit Display

ESP-Platinen mit Display gibt es zuhauf. Wir haben uns exemplarisch drei sehr unterschiedliche herausgepickt: Das „Lilygo T-Display“ mit 1,44-Zoll-Farbdisplay und ESP32, das „Heltec LoRa 32 V3“ und das Modell „APKLVSR ESP32 Display“ mit 2,8-Zoll-Bildschirm und Stiftbedienung. Letzteres nutzt einen ESP32-S2 und bringt einen Verstärker für einen kleinen Lautsprecher mit. Das Lilygo-Modell stützt sich auf den klassischen ESP32, enthält aber eine Ladeschaltung für 3,7-Volt-Lithiumzellen. Geht es um mobile Datenerfassung und deren Weitergabe per LoRaWAN, erspart das Platinchen von Heltec viel Lötarbeit, weil es sowohl LoRaWAN als auch einen Anschluss für eine externe WLAN-Antenne und eine Lithium-Akku-Ladeschaltung hat.

**Preis:** Lilygo T-Display ab 15 Euro,  
Heltec LoRa 32 V3 ab 26 Euro,  
APKLVSR ESP32 Display ab 22 Euro

## Eigene Projekte

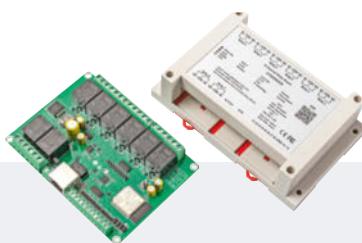
Für die ESP-Großfamilie gibt es sehr viele sehr unterschiedlich ausgestattete Platinen vor allem für den ESP8266, den ESP32 und den ESP32-S3. Dabei gilt es zwischen nackten Modulen und Entwicklungsboards zu unterscheiden. Letztere bieten die Möglichkeit, die ESP-Chips via USB



### ESP-Relais

Einfach nur Relais per WLAN schalten? Dafür gibt es Dutzende fertig bestückte Platinen mit ESP8266EX oder ESP32 und bis zu acht Relais. Die beiden abgebildeten Platinen bringen gleich noch die Stromversorgung für den ESP mit. Das grüne Modell links (ESP12F Relay X4) nutzt das ESP8266MOD-Modul, wohingegen die rechte blaue Platine (ESP32 Relay X4) mit einem ESP32-WROOM daher kommt.

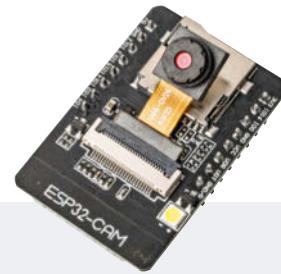
**Preis:** ESP12F Relay X4 ab 5,70 Euro,  
ESP32 Relay X4 ab 14 Euro



### DT Wonder

Eine gute Hardware-Plattform für eine Smart-Home-Zentrale etwa auf Basis von OpenHAB bietet das DT Wonder getaufte Entwicklungsbrett von Dingtian Tech. Es nutzt den ESP32-WROOM und hält neben acht Relais-Ausgängen auch noch acht Eingänge und einen RS485-Port vor. Auch ein Gehäuse für die Hutschienenmontage ist erhältlich

**Preis:** DT Wonder ab 38 Euro,  
mit Gehäuse ab 40 Euro



### ESP32-CAM

Das ESP32-CAM-Kit ist eine Entwicklungsumgebung mit ESP32-Modul, Kamerainterface, Kamera und SD-Kartenleser. Um es zu programmieren, wird eine spezielle USB-Aufsteckplatine benötigt, die sich auch verwenden lässt, um dem Gespann einen Mikro-USB-Port zu spendieren. Für das Kit gibt es reichlich Demo-Programme als Anregung für eigene Projekte.

**Preis:** ESP32-CAM-Kit 8,60 Euro,  
USB-Modul 1,90 Euro

mit Programmcode zu versorgen. Module hingegen sind kompakter und eignen sich eher für die (Klein-)Serienfertigung. Um diese mit Software zu betanken, gibt es verschiedene USB-Adapter-Platinen.

Den günstigsten Einstieg in die ESP-Welt ermöglicht der ESP-01. Das kleine Platinchen ist mit einem ESP8266EX bestückt, kostet in der Fünferpackung unter 2 Euro pro Stück und bringt je nach Ausführung zwischen 512 KByte und 1 MByte Flash-Speicher mit. Über einen achtpoligen Pfostenstecker nimmt es Kontakt zu einem Programmieradapter oder zur selbst entwickelten Schaltung auf. Aktuell ist das Modell ESP-01S, daneben gibt es noch die Variante ESP-01+, die bis zu 4 MByte Flash

mitbringt und einen zehnpoligen Pfostenstecker besitzt. Über diesen sind weitere GPIO-Pins herausgeführt, die eine komfortablere Nutzung von I2C ermöglichen. Wer den ESP-01+ nutzen will, muss allerdings einige GPIO-Vorbelegungen in der Software anpassen.

Ebenfalls sehr kompakt ist das ESP D1 Mini. Auf diesem Entwicklungsbrett ist bereits eine USB-Schnittstelle vorgesehen, die je nach Version eine Mikro-USB-Buchse (D1 Mini, D1 Mini lite) oder eine USB-C-Buchse (D1 Mini C) zur Programmierung des ESP8266EX mitbringt.

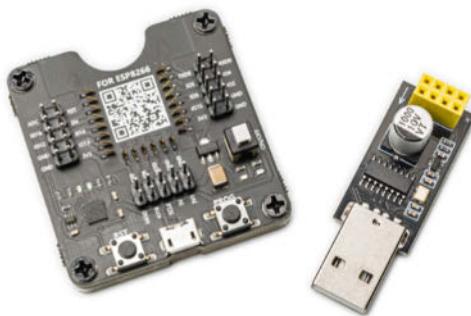
Einen einfachen Zugang zum leistungsstärkeren ESP32 bieten Entwicklungsplatinen vom Typ ESP32-NodeMCU mit einem ESP32-WROOM-Modul an Board. Auch sie bringen eine USB-Schnittstelle (je nach Typ Mikro-USB oder USB-C) für die Programmierung und die Stromversorgung mit. Für WLAN und Bluetooth ist zumeist eine auf der Platine mit Leiterbahnen realisierte Antenne vorgesehen. Es gibt aber auch Modelle, die stattdessen einen Steckkontakt für eine externe Antenne besitzen, was etwa die Montage in einem Metallgehäuse ermöglicht.

Noch mehr Performance versprechen Entwicklungsboards mit dem ESP32-S3, der unter anderem zwei USB-Schnittstellen betreiben kann. Meist nutzen sie ein Modul vom Typ ESP32-S3-WROOM-2 mit Leiterbahn-Antenne. Muss es ein ESP-S3 sein, der sehr kompakt ist, bietet sich die

Variante ESP32-S3 Zero an. Sie bringt zwar nur einen USB-Port mit, ist aber dank Chip-Antenne sehr klein. Das bringt allerdings auch Nachteile mit sich: Längst nicht alle I/O-Kontakte sind bequem per Pfostenstecker abgreifbar. Einige sind nur als kleine Lötpads herausgeführt, da braucht für die Verdrahtung sehr ruhige Hände und eine Lupe.

### Spezialitäten

Meist ist der ESP in Projekten kein Selbstzweck. In der Regel kommen noch weitere Komponenten hinzu, etwa Sensoren, Aktoren oder Displays, die man dann auf selbst entworfenen Leiterplatten unterbringt. Doch das ist mitunter gar nicht erforderlich, denn es gibt sehr viele vorbestückte ESP-Boards, die einem das Layouten und Löten weitgehend abnehmen. Einfachste Beispiele sind ESP32-Platinen mit integrierter LAN-Schnittstelle. Es gibt sie in den verschiedensten Varianten, an-



Um Programmcode auf die Module zu schreiben, gibt es spezielle USB-Adapter, die zum jeweiligen Modul passen müssen. Das quadratische Modell links nimmt ESP8266-Module auf, der kleine Stecker rechts passt an den ESP-01S.



**Das Entwicklungskit für den ESP32-P4 ist mit Touch-Display, Kamera, Mikrofon und Verstärker gut ausgestattet, kostet aber nur knapp 65 Euro.**

## Ausgewählte Mitglieder der ESP-Großfamilie

Serie	ESP8266EX	ESP8685	ESP32	ESP32-S2	ESP32-S3	ESP32-C2/ ESP8684	ESP32-C3	ESP32-C6	ESP32-H2	ESP32-P4
verfügbare Varianten	3	1	6	5	8	2	3	2	2	2
CPU-Architektur	Xtensa 32-Bit L106	32-Bit RISC-V	Xtensa 32-Bit LX6	Xtensa 32-Bit LX7	Xtensa 32-Bit LX7	32-Bit RISC-V	32-Bit RISC-V	32-Bit RISC-V	32-Bit RISC-V	32-Bit RISC-V
CPU-Kerne	1	1	2	1	2	1	1	1	1	2
Taktrate (MHz)	160	160	240	240	240	120	160	160	96	400
Betriebsspannung (V)	2,5-3,6 / 2,7-3,6	3,0-3,6	2,3-3,6 / 3,0-3,6	3,0-3,6	3,0-3,6	3,0-3,6	3,0-3,6	3,0-3,3	3,3-3,6	3,0-3,6
<b>Funkschnittstellen</b>										
WLAN (Wi-Fi 4, 2,4 GHz)	HT20, bis zu 72 Mbit/s	HT20/40, bis zu 144 Mbit/s	HT20/40, bis zu 144 Mbit/s	HT20/40, bis zu 144 Mbit/s	HT20/40, bis zu 144 Mbit/s	HT20, bis zu 72 Mbit/s	HT20/40, bis zu 144 Mbit/s	HT20/40, bis zu 144 Mbit/s <sup>1</sup>	—	—
Thread	—	—	—	—	—	—	—	✓	✓	—
Bluetooth	—	BT LE V5.0	BR/EDR + BT LE V4.2	—	BT LE V5.0	BT LE V5.3	BT LE V5.0	BT LE V5.3	BT LE V5.3	—
<b>Unterstützte Speichervarianten</b>										
SRAM (KByte)	160	400	520	320	512	272	400	512	320	768
ROM (KByte)	0	384	448	128	384	576	384	320	128	128
RTC SRAM (KByte)	1	8	16	16	16	—	8	16	4	32
Flash (MByte)	0	4	0-8	0-4	0-8	2-4	0-4	0-4	2-4	—
PSRAM (MByte)	0	0	0-2	0-2	0-8	—	—	—	—	16-32
Flash-Spannung (V)	3,3	3,3	3,3	3,3	1,8 / 3,3	3,3	3,3	3,3	3,3	3,3
<b>Peripherieschnittstellen</b>										
ADC	1 × 10-Bit, 1 Kanal	2 × 12-Bit, 6 Kanäle	2 × 12-Bit, 18 Kanäle	20 Kanäle, 2 × 12/2 × 13 Bit	2 × 12-Bit, 20 Kanäle	1 × 12-Bit, 5 Kanäle	1 × 12-Bit, 6 Kanäle	1 × 12-Bit, 7 Kanäle	1 × 12-Bit, Kanäle	2 × 12-Bit, 14 Kanäle
DAC	—	—	2 × 8-Bit	2 × 8-Bit	—	—	—	—	—	—
Touch	—	—	10	14	14	—	—	—	—	14
Temperatursensor	—	✓	—	✓	✓	✓	✓	✓	✓	✓
GPIO	17	15	29 / 31 / 33 / 34	43	39 / 45	14	22	22/30	19	55
UART	2	2	3	2	3	2	2	3	2	6
SPI	2	3	4	3	4	3	3	1	3	5
SDIO HOST	—	—	1	—	2	—	—	—	—	1
SDIO SLAVE	1	—	1	—	—	—	—	1	—	1
I2C	1	1	1	2	2	1	1	2	2	3
I2S	1	1	1	1	2	—	1	1	1	4
RMT	—	4 Kanäle	8 Kanäle	4 Kanäle	8 Kanäle	—	4 Kanäle	4 Kanäle	—	—
LED PWM	—	1 × 6 Kanäle	2 × 8 Kanäle	1 × 8 Kanäle	1 × 8 Kanäle	1 × 6 Kanäle	1 × 6 Kanäle	1 × 6 Kanäle	—	—
MCPWM	—	—	2	—	2	—	—	—	—	—
USB OTG	—	—	—	1	1	—	—	—	—	2
Hallsensor	—	—	✓	—	—	—	—	—	—	—
Ethernet	—	—	✓	—	—	—	—	—	—	✓
TWAI	—	1	1	1	1	—	1	2	1	3
JTAG	—	✓	✓	✓	✓	✓	✓	✓	—	—
Kamera	—	—	✓ <sup>2</sup>	✓ <sup>2</sup>	✓ <sup>2</sup>	—	—	—	—	1*DVP 8/16 Bit
TOF	—	—	—	✓	✓	—	—	—	—	—

✓ vorhanden — nicht vorhanden

<sup>1</sup> auch Wi-Fi 6 (HT20, max. 144 Mbit/s)<sup>2</sup> 1 × DVP 8/16 Bit

gefangen bei Modellen mit WLAN- und LAN-Port, bis hin zu galvanisch getrennten Platinen mit PoE-LAN für die Stromversorgung übers Ethernetkabel (Power-over-Ethernet, Olimex ESP32-POE-ISO).

Viel Auswahl gibt es auch für einfache Schaltaufgaben. ESP-Boards mit bis zu acht Relais sind in vielen Varianten im Handel. Einige integrieren sogar das Netzteil und lassen sich etwa zum Steuern der Beleuchtung nutzen. Andere Platinchen bringen ein kleines Display (ESP-T-Display) oder eine Kamera (ESP32-CAM) mit, andere haben größere Anzeigen mit Touch-Bedienung. Spannend ist auch ein Modul von Heltec, das neben

einem kleinen Farbdisplay auch noch LoRaWAN unterstützt und gleich die Ladeschaltung für einen Lithium-Ionen-Akku enthält.

Es geht aber auch ganz groß: Bei Aliexpress entdeckten wir DTWonder, ein Achtkanal-Relais-Board mit RS485-Schnittstelle, ESP32, LAN- und WLAN-Port sowie acht herausgeführten digitalen Eingängen. Es lässt sich in ein optional erhältliches Gehäuse für 35-mm-Hutschienen verpacken und bildet eine interessante Basis für Heimautomationsprojekte etwa mit OpenHAB. Passender Beispielcode liegt auf der Hersteller-Webseite.

## Losbasteln!

Das war nur ein kleiner Ausschnitt an aus unserer Sicht spannenden Entwicklungssystemen und „Halbfertig“-Produkten rund um die verschiedenen ESP-Varianten. Wer nun Lust hat, selbst ein kleines oder großes ESP-Projekt anzugehen, dem geben die folgenden Seiten einen Einblick in gängige Programmierumgebungen sowie einige ESP-basierte Projekte, die die Redaktion verwirklicht hat.

(gs@ct.de) ct

**ESP-Dokumentationen und Bezugsquellen:** [ct.de/y5y9](http://ct.de/y5y9)



# Programmierhelfer

## Entwicklungsgeräte für ESP & Co.

Egal, ob Sie selbst programmieren oder ein fertiges, nur im Quelltext erhältliches Projekt auf einen Mikrocontroller aufspielen möchten: Das geht nicht ohne passende Werkzeuge. Den Einstieg erleichtern grafische Entwicklungsumgebungen. Dieser Artikel stellt einige vor, beleuchtet aber auch deren Umfeld und wirft einen Blick auf Programmiermethoden.

Von Peter Siering

**V**orweg: Wenn Sie mit einem fertigen Projekt starten, das mit einer bestimmten Programmierumgebung geschaffen worden ist, experimentieren Sie nicht. Greifen Sie zu dem Werkzeug, das die Autoren seinerzeit benutzt haben – im Idealfall in identischer Version, denn solche Werkzeuge und die Bestandteile, die sie einbinden, entwickeln sich im Laufe der Zeit weiter. Wenn die Autoren ein Projekt nicht aktuell gehalten haben, steigert die ältere Version die Chance, es ohne Probleme zum Laufen zu bringen.

Andere veröffentlichte Projekte helfen sehr dabei, eine Entwicklungsumgebung und Soft- und Hardwareplattform

näher kennenzulernen. Alle hier vorgestellten Werkzeuge bringen eine Fülle von Beispielcode mit, den Sie studieren und als Inspiration nehmen können. Programmieren heute heißt auch, zu wissen, wo man „abschreiben“ kann. Das ist keine Schande. Rechnen Sie indes damit, dass manches nicht so funktioniert, wie es versprochen wurde.

Falls Sie noch nie einen Mikrocontroller wie Arduino oder ESP programmiert haben: Üblicherweise verwendet man als Hardware ein Dev-Kit, das per USB an den PC angeschlossen wird. Über den als serielle Schnittstelle eingerichteten USB-Port erhält es den Programmcode. Dort kann es auch Textdaten ausgeben, etwa

Meldungen zum Fortschritt oder zu Fehlern. Theoretisch ist es auch möglich, von ihr Texteingaben einzusammeln – davon machen aber nur wenige Projekte Gebrauch.

## Basisbetrachtungen

Die meisten Entwickler greifen zum Programmieren von Mikrocontrollern zur Sprache C/C++. Ob mit Objekten in C++ oder rein prozedural in C bleibt dabei dem Geschmack überlassen (mehr dazu ab Seite 62). Compiler übersetzen den Quelltext in Binärkode für die CPU des Mikrocontrollers. Ein Linker bindet fertige Bibliotheken ein. Das Ergebnis bringt ein Flasher dann auf den Controller. Die im Folgenden besprochenen grafischen Entwicklungsumgebungen gibt es für Windows, Linux und macOS. Sie bringen die nötigen Cross-Compiler und sonstigen Werkzeuge mit beziehungsweise richten diese automatisch ein.

Das klappt mit vielen hier vorgestellten Werkzeugen nicht nur für die ESP-Familie, sondern auch für Arduinos, Raspis aus der Pico-Familie und etliche weitere Mikrocontroller. In der Regel stellen deren Hersteller die Werkzeuge für das Übersetzen, Linken und Flashen bereit oder bedienen sich dazu der weit entwickelten GNU-Compiler, für die das Übersetzen von Code auf einer CPU für eine andere seit jeher selbstverständlich ist.

Die grafischen Entwicklungsumgebungen verstecken dieses Werkzeug hinter ihrer Bedienoberfläche. Sie als Nutzer drücken einen Knopf und die Prozesse laufen automatisch ab. Die Ausgaben der beteiligten Werkzeuge bekommen Sie aber üblicherweise zu sehen. Ganz ignorieren können Sie die nämlich nicht: Fehlermeldungen, meist farblich hervorgehoben, geben oft entscheidende und hilfreiche Hinweise.

Neben den Werkzeugen spielen auch Bibliotheken eine wichtige Rolle: Um Displays, Sensoren und Aktoren anzusteuern und auch für viele Grundfunktionen wie das dauerhafte Speichern von Daten oder das Einbinden von Funkprotokollen wie WLAN und Bluetooth greifen Programme auf fertige Funktionen zurück, anstatt das Rad ständig neu zu erfinden. Idealerweise hilft eine Programmierumgebung bei der Verwaltung dieser Bibliotheken.

Den Grundstock an Werkzeugen, Funktionen und Bibliotheken stellen üblicherweise die Hersteller der Mikro-

controller bereit. Im Fall der ESP-Familie veröffentlicht Espressif für den ESP8266 das Non-OS- und das FreeRTOS-SDK sowie für den ESP32 das Espressif IoT Development Framework (ESP-IDF). Sie bilden die Grundlage, um Programme zu schreiben, und bringen bereits einen umfangreichen Satz von Funktionen mit.

Die Programmierumgebungen ergänzen eigene Bibliotheken und Hilfsmittel oder schaffen mit weiteren Schichten mehr oder minder stark von den Ideen der Hersteller abweichende Möglichkeiten zur Programmierung. Krasse Beispiele dafür sind alternative Sprachen, siehe Kasten „Python, JavaScript, LUA und Rust“. Dabei variiert nicht nur die Sprache, sondern vor allem die Herangehensweise an die Programmierung.

## Superloop versus Tasks

Ein typisches Beispiel für die verbreitete C/C++-Programmierung ist die grafische Arduino-IDE (Integrated Development Environment). Ihre Grundlagen leben heute auf vielen weiteren, auch Nicht-Ar-

duino-Mikrocontrollern wie der ESP-Familie weiter. Man spricht auch vom Programmieren auf Basis des Arduino-Frameworks oder von Arduino-Core, das auf diverse Mikrocontroller-Plattformen portiert wurde.

Der Kern der Idee ist von einem vereinfachten Modell zur Programmierung namens Processing inspiriert: Ein Programm besteht aus zwei wesentlichen Funktionen. Die zuerst einmalig ausgeführte `setup()`-Funktion richtet den Controller für den Betrieb ein, verbindet etwa mit einem WLAN, liest Konfigurationsdaten et cetera. Alles Weitere passiert in der `loop()`-Funktion oder daraus aufgerufenen weiteren Funktionen und Bibliotheksaufrufern. Die `loop()`-Funktion wird, sobald sie endet, erneut aufgerufen. Wegen der Processing-Abstammung heißen Arduino-Programme auch Sketches.

Professionelle Entwickler belächeln das auch als Superloop bekannte Verfahren gern, es trägt aber erstaunlich weit. Das allerdings nur, wenn man es nicht überstrapaziert: Ein `loop()`-Durchlauf

## Python, JavaScript, LUA und Rust

Wachsender Beliebtheit erfreut sich Python auf dem ESP. Dabei landet auf dem Mikrocontroller kein von passenden Werkzeugen erzeugter Binärkode, sondern der Python-Quelltext. Den führt ein zuvor auf dem Mikrocontroller installierter Interpreter dann aus. Mit MicroPython und CircuitPython gibt es zwei Varianten, die viele Details unterschiedlich lösen. Nur MicroPython unterstützt noch den ESP8266, CircuitPython hat den Baustein aufgegeben und fordert einen ESP32.

Das Aufspielen von Python-Code auf einen Controller, auf dem bereits der Interpreter aktiv ist, fällt denkbar leicht: Steckt man ihn an den PC an, taucht er als Speichergerät auf, als wäre es ein USB-Stick; manchmal muss man zusätzlich eine Taste auf dem Controller dafür betätigen. Man legt dann das Python-Programm unter einem vorgesehenen Namen ab und nach einem Neustart führt der Controller den Code aus.

Um den Interpreter zu installieren und später bequemer mit mehreren

Skriptdateien und eventuellen Abhängigkeiten zu hantieren, empfiehlt sich ein eigenes Tool. Zu besonderer Beliebtheit hat es Thonny gebracht, eine grafische Python-Entwicklungsumgebung für Einsteiger. Die gibt es für Linux, macOS und Windows. Sie bietet viele Hilfen, die beim Schreiben von Python-Code nützlich sind. Wie das Entwickeln mit MicroPython in der Praxis funktioniert, lesen Sie ab Seite 130.

Außer Python gibt es Implementierungen für JavaScript, die ebenfalls als Interpreter auf dem Mikrocontroller laufen. Auch LUA als Skriptsprache ist erhältlich, eine Sprache, die der Hersteller Espressif einst favorisierte. Für Lua beziehungsweise das darunterliegende Lua RTOS gibt es Werkzeuge, die das grafische Programmieren per Drag & Drop erlauben. Doch das sind alles Ansätze, die eher auf dem absteigenden Ast sind. An Fahrt gewinnt hingegen Rust, das für den ESP Binärkode erzeugt – noch ist das aber eher etwas für Spezialisten ([ct.de/yudq](http://ct.de/yudq)).

sollte nicht ewig brauchen, keine Zeit mit aktivem Warten vertrödeln und, falls er etwas längere Rechenzeiten benötigt, darin regelmäßig die `yield()`-Funktion aufrufen. So stocken im Hintergrund zu erledigende Aufgaben nicht, etwa stirbt die WLAN-Verbindung nicht ab; solche Hintergrundaufgaben werden nach jedem `loop()`-Durchlauf automatisch aufgerufen.

Die reine Lehre der professionellen Entwickler, das sei hier noch erwähnt, sieht andere Methoden vor: Sie lassen das Betriebssystem, meist eines, das Reaktionszeiten garantiert (Echtzeit- oder Realtime-OS, RTOS), mehrere Aufgaben parallel mit vorgegebenen Prioritäten ausführen. So ist sichergestellt, dass nicht versehentlich etwas Wichtiges unter den Tisch fällt. Allerdings ist der Aufwand, solche Software zu schreiben, deutlich höher und überfordert Anfänger.

Wer die ESP32-Modelle mit einer Superloop beglückt, muss dabei auf gar nicht allzu viel verzichten. Das von den Profis geschätzte FreeRTOS bildet auf dem ESP32 auch die Basis für das Arduino-Framework. Es ist möglich, aus dem Arduino-Code heraus auf den Unterbau zuzugreifen, um zum Beispiel Interrupt-Routinen zu bauen oder auf interne Timer zuzugreifen, die solche Interrupts auslö-

sen. Auch hierfür finden sich zahlreiche Beispiele.

Espressif als Entwickler und Hersteller der ESP-Familie sieht sowohl eine Task-orientierte als auch eine Superloop-artige Methode für die Programmierung in seinem eigenen Entwicklungs-Framework für den ESP32 vor, dem Espressif IoT Development Framework. Das zu ESP-IDF abgekürzte Framework ist aus Herstellersicht die native Methode, für den ESP32 zu entwickeln. Es ist im Hobby-Umfeld aber weniger verbreitet.

### Arduino-IDE

Mit diesem Hintergrund fällt es leichter, die verschiedenen IDEs einzuordnen. Die Arduino-IDE hat dieser Artikel bereits erwähnt. Nach der Installation dieser Software muss man über den Board Manager zunächst auswählen, welche Art von Controller man programmieren möchte. Die Liste ist lang, man kann darin aber durch Teileingaben suchen und kann sich dabei mit zusätzlichen Produktbezeichnungen helfen.

Zum Übersetzen eigener und fremder Projekte werden oft zusätzliche Bibliotheken benötigt. Dafür bringt die IDE eine Bibliotheksverwaltung mit, die ebenfalls eine komfortable Suche enthält. So kramt man schnell alles Nötige zu-

sammen. Idealerweise hat der Autor einer Bibliothek Hinweise zu eventuell nötigen weiteren Bibliotheken gegeben, die man in der IDE via Zip-Archiv ergänzen kann; manchmal hat er die auch gleich in das Zip-Archiv mit dem Quellcode gepackt.

Die Arduino-IDE bedient sich mitunter zäh und bietet wenige Extras. Letzteres kommt Einsteigern entgegen, aber Funktionen zur Integration von Git zur Versionsverwaltung wären durchaus wünschenswert. So muss man dafür eine externe Software bemühen. Das ist auf Dauer unnötig mühsam. Andererseits hat die IDE auch alles an Bord, was nötig ist. Nach der Auswahl eines Controllers stellt sie auch eine umfangreiche Sammlung von Beispielen zur Verfügung, die man mit wenigen Klicks öffnen kann.

Darum, dass die nötigen Anpassungen für moderne ESP32-Modelle im Board Manager der IDE bereitstehen, kümmert sich Espressif selbst; für die älteren ESP8266 ist es komplizierter (siehe S. 90). Achten Sie beim Hinzufügen über den Board Manager darauf, dass Sie nicht versehentlich die Unterstützung von Arduino selbst für deren ESP-basierte Arduinos erwischen (Arduino ESP32 Board). Mit der Espressif-Variante (esp32 by Espressif Systems) stehen die Chancen besonders gut, mit der IDE auch Code für aktuelle Modelle schreiben zu können. Das ist, wie sich im Folgenden zeigt, leider nicht selbstverständlich.

### PlatformIO mit VS Code

Als heißer Kandidat für die ESP-Programmierung wird Visual Studio Code (VS Code oder VSC) mit der Erweiterung PlatformIO gehandelt, als Gespann auch PlatformIO IDE genannt. Diese Werkzeugsammlung beeindruckt aus mehreren Gründen: Sie glänzt mit vielen Erweiterungen und Selbstverständlichkeiten wie einer Git-Integration und bedient sich trotzdem flüssiger als die Arduino-IDE. Sie bearbeitet und erzeugt Software für mehrere Plattformen und Frameworks. Einsteiger werden anfangs mit den vielen Funktionen kämpfen.

PlatformIO ist in erster Linie ein Werkzeug, um plattform- und architekturübergreifend zu programmieren. Es klammert die Frameworks und Werkzeuge und versieht sie mit einer Art Paketverwaltung. Ein PlatformIO-Projekt beschreibt sehr exakt Abhängigkeiten, Werkzeuge und Bibliotheken. Wenn man ein vorhandenes

## ESP != ESP

Die ESP-Familie besteht momentan aus drei Hauptzweigen: dem alten ESP8266, dem nachfolgenden ESP32 mit Xtensa-CPU (an einem fehlenden Buchstaben oder angehängtem S zu erkennen) sowie dem jüngsten ESP32 mit RISC-V-CPU (erkennbar an angehängtem C, H oder P). Mehr zu den Unterschieden lesen Sie im Artikel ab Seite 78.

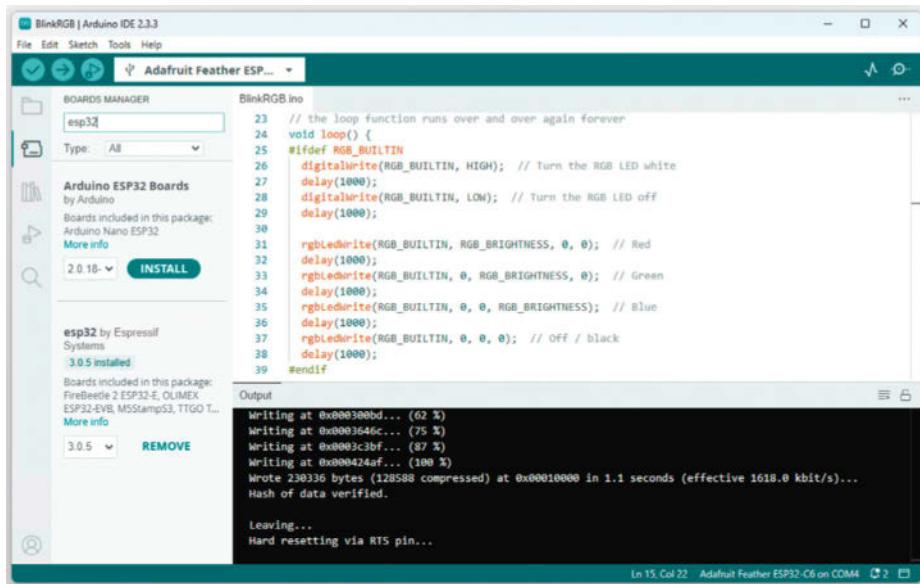
Die Grundlage für Entwicklungswerkzeuge bilden von Espressif veröffentlichte Software Development Kits (SDK): für den ESP8266 das NonOS- und später das RTOS-SDK, für den jüngeren Teil der ESP32-Familie das Espressif IoT Development Framework (ESP-IDF). Neue CPUs wie RISC-V deckt immer nur die aktuelle Version des ESP-IDF ab.

Die darauf zurückgreifenden Werkzeuge, etwa die Arduino-IDE, abstrahieren zwar von den Unterschieden, können entsprechend aber nur ESP-Modelle be-

dienen, auf deren SDK sie gründen. Was leider heißt: Werkzeug, das für den ESP32-S2 funktioniert, muss nicht zwangsläufig auch für den ESP32-C6 taugen.

Der ESP8266 ist bei dieser Betrachtungsweise aufgrund der anderen Basis mit dem ESP32 nur entfernt verwandt. Für den ESP8266 geschriebener Code läuft erst nach Anpassung auf ESP32-Modellen. Den Versuch, die ESP8266-Basis mit dem RTOS-SDK kompatibel zum ESP-IDF zu machen, hat Espressif still und leise aufgegeben.

Fazit des Exkurses: Wenn Sie auf fertige Projekte aufbauen wollen, greifen Sie zu Werkzeugen für die jeweiligen ESP-Modelle. Wenn Sie ein neues Projekt anfangen wollen, ist ein ESP32-Modell mit RISC-V-Prozessor eine gute Wahl. Achten Sie beim Werkzeug aber darauf, dass es auch diese Variante unterstützt.



**Einstiegerfreundliche Arduino-IDE:** Die Oberfläche besteht aus den typischen Elementen Quelltexte bearbeiten, Ausgaben beim Übersetzen und Flashen beobachten, optional dort auch die Ausgaben der selbstgebackenen Firmware sehen. Zusätzlich zeigt das Bild den im Tools-Menü zugänglichen Boards Manager, mit dem Sie die nötigen Dateien für einen Mikrocontrollertyp hinzufügen müssen.

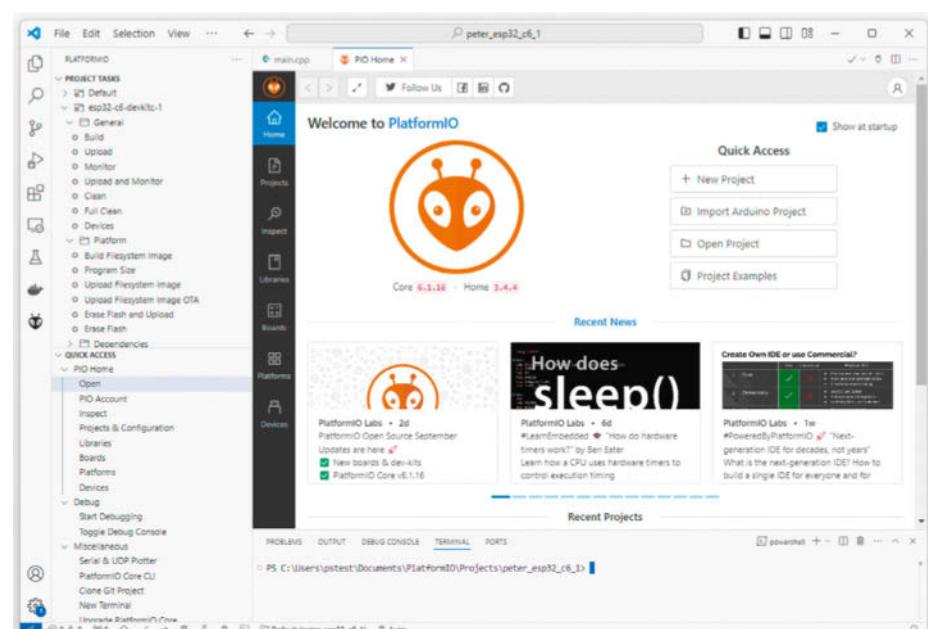
Projekt für PlatformIO findet, kann man sehr sicher sein, dass es sich übersetzen lässt. Man muss nicht manuell Bibliotheken beschaffen, das erledigt PlatformIO weitgehend automatisch.

PlatformIO arbeitet mit vielen Entwicklungsumgebungen und Editoren. Am besten spielt es aber zusammen mit VS Code auf. Nach der Installation von VS Code fügen Sie die Erweiterung „PlatformIO IDE“ hinzu. Daraufhin taucht in der Aktivitätsleiste, in der Regel am linken Fensterrand, das PlatformIO-Symbol auf. Klickt man darauf, zeigt die Side Bar die Funktionsvielfalt auf. Die angebotenen Funktionen variieren je nach Kontext. Ist ein Projekt offen, starten Sie zum Beispiel das Übersetzen und Bespielen des Mikrocontrollers per „Build“ und „Upload“.

In einer frischen Installation bietet die Umgebung an, einen lokalen Ordner zu öffnen oder ein Projekt zu erstellen. Beim Erstellen eines Projekts erfragt es die Zielplattform. Sollte die noch nicht in der Installation verwendet worden sein, beschafft PlatformIO alle nötigen Programme und Bibliotheken dafür. Außerdem legt es eine leere Projekthülle an. Obendrein ergänzt es für die jeweilige Plattform vorhandene Beispielprojekte zur späteren Auswahl.

Ein neues Projekt erstellen Sie auf einer in VS Code als Editor-Fenster inte-

grierten Seite namens „PIO Home“. Sie fügt Bibliotheken zur Installation und zu Projekten hinzu, zeigt eine Liste der unterstützten Boards an und hilft, Frameworks dafür zu installieren und Projekt-



**Microsofts Visual Studio Code mutiert über die Erweiterung PlatformIO in eine Entwicklungsumgebung für eine Vielzahl von Mikrocontrollern. Damit lässt sich im Arduino-Stil programmieren, aber auch aufbauend auf dem ESP-IDF. Viele erfolgreiche Open-Source-Projekte für ESP greifen auf PlatformIO zurück.**

details zu bearbeiten. Die projektspezifischen Informationen landen in einer Datei namens `platformio.ini`. Sie beschreibt ein Projekt so, dass PlatformIO auf einer anderen Instanz (erfüllbare) Abhängigkeiten automatisch auflösen kann.

Die Dokumentation der Abhängigkeiten in der zentralen Projektdatei `platformio.ini` geht so weit, dass dort sogar Versionsnummern notiert sind. So ist es in der Regel ein Leichtes, PlatformIO-Projekte in einer frisch installierten Umgebung zu bauen und auf einen Mikrocontroller zu überspielen. Im Vergleich zur Arduino-IDE und damit erstellten Projekten ist das ein Riesenfortschritt, aber wegen der insgesamt komplexeren Umgebung auch mühsamer zu erlernen.

Ein paar praktische Hilfen seien noch aufgezählt: Beim Erstellen eines Projekts bietet PlatformIO an, ein Arduino- in ein PlatformIO-Projekt zu überführen und unterstützt diesen Prozess unterwegs mit Hinweisen. In der mit einem Klick auf das PlatformIO-Symbol geöffneten Side Bar steckt unter „Miscellaneous“ die Funktion „Clone Git Project“. So bekommt man besonders schnell bestehende Projekte in die Umgebung.

Cool sind die Möglichkeiten, die ein PlatformIO-Konto eröffnet: Dann kann

ein weit entfernt aufgestellter und mit den PlatformIO-Werkzeugen für die Kommandozeile bestückter Raspi einen PlatformIO-Agent ausführen. Der bespielt an ihn angeschlossene ESPs mit frisch übersetzter Firmware und macht deren serielle Monitore zugänglich. Das Ganze beschickt eine lokal ausgeführte Instanz von VS Code.

Die Mächtigkeit und Komplexität von PlatformIO in VS Code sind Segen und Fluch zugleich: Manchmal kranken Details, etwa bei dem zuvor geschilderten Austausch mussten wir uns zweier Tricks aus bereits länger offenen Tickets zu PlatformIO auf GitHub bedienen (siehe ct.de/yudq).

### Eclipse: ESP-IDF-IDE

Wer dichter an das Espressif-Universum heranrücken will, kann statt des Arduino-Frameworks auch auf das ESP-IDF setzen. Das geht innerhalb von VS Code mit einer Erweiterung von Espressif oder mit PlatformIO, aber auch in der nativen Entwicklungsoberfläche von Espressif. Es handelt sich dabei nicht etwa um eine

eigenständig gepflegte Entwicklungsumgebung, sondern letztlich um ein Plug-in für Eclipse, einem Werkzeug, das im Java-Umfeld entstand, ihm aber längst entwachsen ist.

Espressif empfiehlt Eclipse und Plug-in als Bundle zu installieren und stellt passende Pakete zum Download bereit. Die sind gar nicht mal so groß, weil das eigentliche ESP-IDF-Framework erst nach der Installation hinzugefügt und heruntergeladen wird (via ESP-IDF-Manager im Espressif-Menü). Für Windows bietet Espressif zusätzlich einen Offline-Installer an, der die ansonsten wie bei Linux und macOS automatisiert heruntergeladenen Abhängigkeiten mit einem großen Download erschlägt. Geduld ist in beiden Fällen eine Tugend.

Das Ergebnis ist eine aufgeräumte IDE, die eng mit der Espressif-Welt verzahnt ist. Es gibt viel Beispielcode und Links in die ausführliche Online-Dokumentation. Wünsche, die eine Standard-installation nicht erfüllt, lassen sich als Eclipse-Plug-ins ergänzen, etwa Git. Die Quellen des Frameworks liegen ebenso

wie der Bug-Tracker auf GitHub – der Riesenfundus verrät Details zum Innenleben, selbst wenn man die Espressif-IDE gar nicht verwendet.

Mit dem ESP-IDF hat man als Entwickler die Wahl zwischen Superloop oder Task-orientierter Herangehensweise. Die als Task parallel laufenden Funktionen enthalten in der Regel eine Endlosschleife. Damit sie untereinander Daten austauschen können, stehen unter anderem Queues bereit. Das ESP-IDF stellt viele weitere Funktionen bereit, um den Datenzugriff zu serialisieren und um Tasks miteinander kommunizieren zu lassen, etwa Events.

Nur im ESP-IDF (in der IDE oder mit PlatformIO) bekommen Entwickler die Tiefe der Konfigurationsmöglichkeiten der ESP32-Plattform zu Gesicht. Teil eines solchen Projekts ist immer die Datei sdkconfig, die detaillierte Optionen zum Mikrocontroller und zur Software enthält, etwa der Größe des PSRAM. Die Datei wird üblicherweise automatisch generiert und lässt sich per menuconfig aus dem Tools-Menü auch interaktiv bearbeiten; sie erinnert entfernt an die Konfigurationsorgien beim Selbstübersetzen von Linux-Kernen.

## ESP32-C3 und -C6 mit PlatformIO und Arduino

Wer beim Kauf eines ESP32 nicht auf die Details achtet, bekommt womöglich einen ESP32-C6 serviert. Der hat einen RISC-V-CPU-Kern, für den PlatformIO momentan nicht gerüstet ist. Beim Anlegen eines neuen Projekts für diesen Mikrocontrollertyp bietet es nur ESP-IDF als Framework zur Auswahl an. Die Ursache dafür: Die Anpassung einer Mikrocontroller-Plattform an PlatformIO geschieht nicht automatisch, sondern ist eine umfangreiche Arbeitsleistung, die sich der gleichnamige Hersteller bezahlen lässt.

Üblicherweise bezahlt also der Mikrocontroller-Fabrikant die Firma hinter PlatformIO. Das ist das Geschäftsmodell, seit PlatformIO sich 2020 dazu entschlossen hat, seine Software kostenlos bereitzustellen. In der jüngsten Vergangenheit hat das aber weder bei den Pico-Modellen des Raspi noch bei den neuen RISC-V-CPUs geklappt. Weder die Raspberry Pi Foundation noch Espressif waren dazu bereit, nachlesen kann man das in diversen Tickets (siehe ct.de/yudq).

Da die neuen Mikrocontroller mit RISC-CPU attraktiv sind, haben von PlatformIO und Arduino-Framework abhängige Projekte selbst Hand angelegt. So gab es erste Fassungen des Arduino-Frameworks in Version 3, das längst in der Arduino-IDE angekommen und Voraussetzung für die ESP32-C-Serie ist, zunächst in Spezialversionen für das Tasmota-Projekt, mit einigen Optimierungen genau für diesen Einsatzzweck.

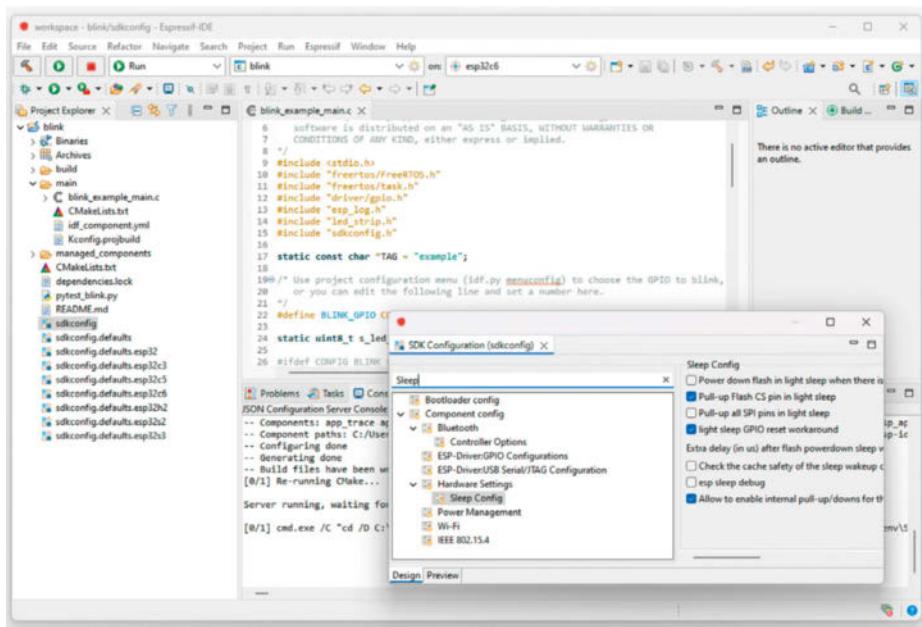
Inzwischen ist aber ein Community-Projekt aus der Integration des Arduino-Frameworks 3 in PlatformIO geworden. Es firmiert auf GitHub als „pioarduino“, was für „people initiated optimized arduino“ steht und sich selbst als „Fork of Platformio Espressif 32“ bezeichnet. Indem man die auf der GitHub-Seite genannte URL (siehe ct.de/yudq) in einer PlatformIO-Installation auf der „PIO Home“-Seite unter Plattform und „Advanced platform Installation“ einfügt, kann man die modernen ESPs mit Code im Arduino-Stil beschicken.

### Fehlerfreuden

Je nachdem, ob Sie mit Linux, macOS oder Windows arbeiten, müssen Sie eventuell mit seriellen USB-Schnittstellen und den Treibern dafür kämpfen. Einige IDEs installieren automatisch zum Betriebssystem passende Treiber. Manchmal muss man dafür sorgen, dass das Betriebssystem einen angesteckten Mikrocontroller nicht zum Beispiel als Braille-Ausgabe erkennt; ein c't-Hotline-Tipp für Linux-Mint zeigt, wie das per udev-Regel zu verhindern ist (siehe ct.de/yudq) – gern greifen Hersteller solcher Geräte nämlich zu den gleichen USB-Chips wie die Mikrocontroller.

Schwierigkeiten, wenn eine IDE den als USB-Gerät angesteckten Controller nicht findet, riesacken vor allem Windows-Nutzer. Unter Linux und macOS klappt das meist sogar ohne Nachdenken. Knifflig wird es bei allen Systemen, wenn mehrere serielle USB-Geräte ins Spiel kommen, wie das zum Debuggen nötig sein kann. Dann ist Geduld beim Recherchieren in Bugtrackern und Howtos gefragt, um die Probleme auszuräumen.

Hilfreich ist es auch, stets im Hinterkopf zu behalten, dass hinter allen Werk-



**Espressif als Hersteller der ESP-Familie hat für den ESP32 eine „eigene“ IDE im Angebot. Die baut auf der Entwicklungsumgebung Eclipse auf und öffnet die ganze ESP32-Welt im Detail, unter anderem auch die jedem Projekt zugrundeliegende SDK-Konfiguration, hier sogar mit grafischer Oberfläche.**

zeugen eine komplexe Kette von einzeln aktiv werdenden Programmen steckt. Zentral ist oft Python. Bei der Installation richten alle Umgebungen sämtliche nötigen Werkzeuge ein, berücksichtigen aber nicht immer alle Besonderheiten, etwa PlatformIO unter Windows: Dort muss man den eventuell aktivierte „Überwachten Ordnerzugriff“ deaktivieren oder sehr, sehr fleißig viele einzelne Programme davon ausnehmen.

Apropos schiefgehen: Was beim Entwickeln für und unter Linux, macOS und Windows selbstverständlich ist, nämlich die Fähigkeit der Entwicklungswerkzeuge, ein Programm Schritt für Schritt (Single-Step) im Debugger zu verfolgen, ist bei der Mikrocontroller-Programmierung die Ausnahme. Oft spicken Entwickler deswegen den Code mit print-Anweisungen, die auf der seriellen Konsole ausgeben, wo das Programm gerade steckt, welchen Wert eine Variable enthält und so weiter. Das ist hässlich und kann Seiteneffekte provozieren.

Mikrocontroller weisen üblicherweise JTAG-Anschlüsse auf. Daran lässt sich ein Adapter anschließen, der das Verfolgen des Programmablaufs mit einem Debugger erlaubt. So kann man Haltepunkte setzen, Variableninhalte ansehen und manipulieren. Die JTAG-Adapter sind teuer,

nicht für jeden Mikrocontroller zu bekommen und schwer zu handhaben.

Mit den ESP32-Dev-Kits der C-Serie mit RISC-V-CPU wird das einfacher: Deren USB-Anschlüsse machen die JTAG-Anschlüsse erreichbar. So kann man auch ohne JTAG-Adapter Fehler im eigenen Code mit einem Debugger suchen. Das vermag sogar schon die Arduino-IDE. Leider gelang uns das nicht reproduzierbar – oft stapelten sich bei Debugging-Versuchen obskure Fehlermeldungen der beteiligten Softwareschichten, denen wir noch nicht im Detail auf den Grund gegangen sind.

## Fazit

Die Arduino-IDE empfiehlt sich nur für Einsteiger: Sie ist, was die Unterstützung neuer Mikrocontroller angeht, durchaus vorn dabei. An Entwickler mit Erfahrung richtet sich VS Code zusammen mit PlatformIO. Das wäre auch unsere Empfehlung für das Veröffentlichen von eigenen Projekten für Dritte. Die Espressif-eigenen Werkzeuge auf Eclipse-Basis oder in PlatformIO sind eher etwas für Profis, denn diese helfen, auch das letzte Register eines ESP zu ziehen. (ps@ct.de) **ct**

**Downloads, Quellen und Tutorials:**  
[ct.de/yudq](http://ct.de/yudq)

**Es gibt 10 Arten von Menschen.  
iX-Leser und die anderen.**



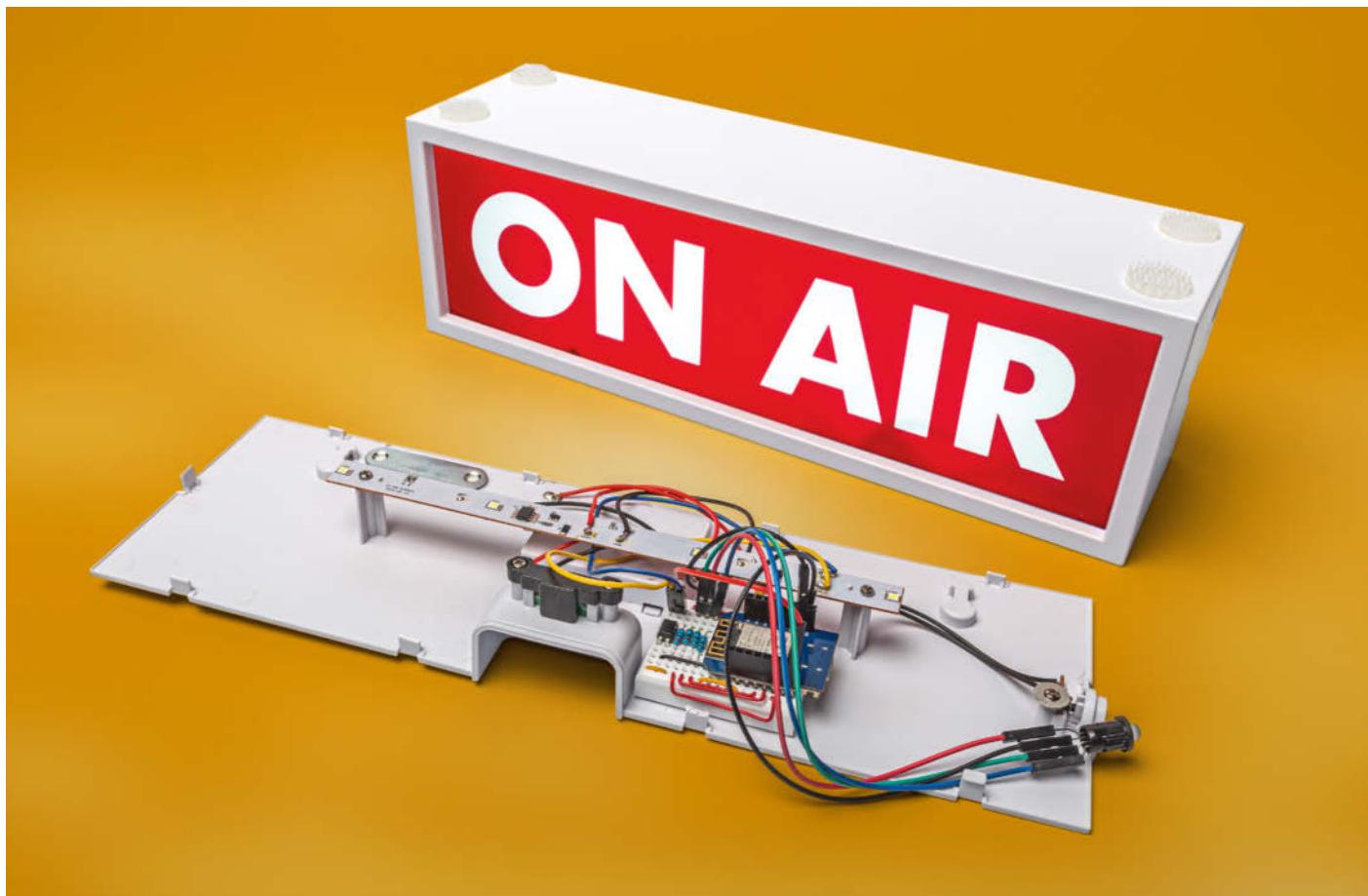
**Jetzt Mini-Abo testen:**

3 Hefte + Bluetooth-Tastatur  
nur 19,35 €

**[www.iX.de/testen](http://www.iX.de/testen)**



[www.iX.de/testen](http://www.iX.de/testen)  
[leserservice@heise.de](mailto:leserservice@heise.de)  
 49 (0)541 800 09 120



# Achtung, Aufnahme

## Bastuprojekt: „On Air“-Lampe mit WLAN

**Vor Radio- und Fernsehstudios warnt eine rote Lampe mit der Aufschrift „On Air“, wenn eine Aufnahme läuft und das Betreten des Studios unerwünscht ist. Mit ein bisschen Bastelei und Software bekommt man so etwas auch für Videokonferenzen aus dem Homeoffice hin und muss sich nicht einmal selbst darum kümmern, die Lampe ein- und auszuschalten.**

Von Hajo Schulz

**S**ein ich meine Arbeitszeit überwiegend im Homeoffice verbringe, höre ich mit schöner Regelmäßigkeit mehrmals am Tag die Frage: „Kann ich kurz reinkommen?“ Meine Frau hat mir irgendetwas mitzuteilen und möchte wissen, ob ich nicht gerade in einem Videoanruf oder einer Konferenz stecke.

Abhilfe könnte ein Schild oder ein anderes Signal an der Bürotür schaffen, das ich immer dann aktiviere, wenn ich das Bedürfnis habe, nicht gestört zu werden. Wir haben es probiert, aber ich schaffe es einfach nicht, bei jedem Videoanruf daran zu denken. Eine Automatik muss her.

In Ton- und Videostudios gibt es so etwas ja auch: Sobald eine Aufnahme oder eine Sendung läuft, verkündet eine Leuch-

te an der Tür: „On Air“. Für den Hausgebrauch wäre es ideal, wenn solch eine Lampe immer dann leuchtete, wenn ich am PC in einem Programm wie Teams oder Zoom per Video kommuniziere. Eine drahtlose Lösung wäre wünschenswert, um kein neues Kabel durch die Wohnung ziehen zu müssen. Damit ist das Pflichtenheft des in diesem Artikel beschriebenen Bastuprojekts auch schon komplett.

Es besteht aus zwei Teilen: Auf dem PC überwacht eine im Hintergrund laufende Software, ob die Webcam gerade aktiv ist. Wie wir weiter unten noch beschreiben, ist es zwar ein bisschen tricky, an diese Information zu kommen, dafür ist der Ansatz unabhängig von der eingesetzten Konferenzsoftware.

## Erleuchtung

Die eigentliche Bastelarbeit steckt in einer Lampe, die sich im heimischen WLAN anmeldet, auf Zurufe übers Netzwerk wartet und sich daraufhin ein- und ausschaltet. Als Basis für so ein Netzwerkgerät bietet sich ein Mikrocontroller vom Typ ESP8266 an, der eine WLAN-Schnittstelle an Bord hat und sich mithilfe des Arduino-Frameworks programmieren lässt. Die Rechenleistung reicht für den angepeilten Zweck völlig. Der Vorteil des Veteranen gegenüber dem moderneren und sehr populären ESP32 besteht vor allem darin, dass es mit

## Projektinfo



Eine Windows-Software kommuniziert per WLAN mit einem in eine „On Air“-Lampe eingebauten Mikrocontroller und bringt sie immer dann zum Leuchten, wenn auf dem Rechner die Webcam aktiv ist.



Grundlegende Elektronik-Bastelkenntnisse  
Arduino-Programmierung  
Windows-Programmierung (.NET, C#)



4–6 Stunden



ca. 40 Euro +  
USB-Stromversorgung



On Air Light (Paladone PP8597)  
ESP8266-Experimentierplatine  
„D1 Mini“  
Mini-Steckplatine  
Optokoppler  
(LTV 817, PC817C oder ähnlich)  
RGB-LED mit Fassung  
Elektronik-Kleinmaterial  
(Widerstände, Jumper-Kabel,  
Schaltdraht)  
auf Wunsch Magnetschnäpper  
und doppelseitiges Klebeband  
USB-Netzteil oder -Powerbank



spitzes Messer  
Seitenschneider  
Abisolierwerkzeug  
Spizzange  
Lötkolben  
Multimeter  
8-mm-Bohrer

ersterem bestückte Bastelboards in sehr kompakter Bauform gibt – in der Lampe, die wir als Gehäuse für das Projekt gefunden haben, geht es benötigt zu.

Bei der Lampe handelt es sich um das „On Air Light“ mit der Modellnummer PP8597 des britischen Herstellers Paladone, das es bei MediaMarkt/Saturn oder im Versandhandel für unter 20 Euro gibt. Strom bezieht es wahlweise von drei in das zugehörige Fach eingelegten AAA-Batterien oder über einen Micro-USB-Anschluss. Was es für unseren Zweck besonders geeignet macht: Es leuchtet nicht direkt, wenn man es an eine Spannung anschließt, sondern hat einen Taster, mit dem man es zwischen den drei Betriebsmodi Aus, Dauerleuchten und Blinken umschaltet. Der Plan: Eine um den Mikrocontroller herumgestrickte Elektronik überbrückt diesen Taster und simuliert somit ferngesteuert Tastendrücke. Das hat den Vorteil, dass die eigene Schaltung sich nicht darum kümmern muss, welche Spannung und wie viel Strom die LEDs benötigen, die die Lampe zum Leuchten bringen.

Damit die beiden Schaltungen – die in der Lampe vorhandene und unsere mit dem Controller – einander ansonsten nicht in die Quere kommen, sollten sie elektrisch voneinander getrennt sein. Das passende Bauelement, um aus einem Schaltkreis heraus etwas in einem anderen galvanisch getrennt zu steuern, ist ein Optokoppler (siehe Kasten). Das Exemplar, das wir verwendet haben, trägt die Typenbezeichnung LTV 817 und ist beim Elektronikversender ein Cent-Artikel. Als Vorwiderstand für seine LED sind 220 Ohm ein passender Wert.

Damit ist die zu bauende Schaltung eigentlich schon fertig beschrieben. Um Lötarbeiten auf ein Minimum zu beschränken, kann man sie auf einem Mini-Breadboard zusammenstecken. Das hat dann sogar noch Platz für ein bisschen Luxus: Eine zusätzliche RGB-LED soll den Betriebszustand der ferngesteuerten Lampe anzeigen. Deren Beschaltung ist kein großes Geheimnis: Gängige Ausführungen verhalten sich wie drei LEDs in einem gemeinsamen Glaskörper mit vier Anschlüssen: einer für jede Farbe und ein gemeinsamer Masse-Pin (Common Cathode); es gibt auch Modelle mit getrennten Masseanschlüssen und gemeinsamem Pluspol (Common Anode). In beiden Fällen braucht man pro Farbe einen eigenen Vorwiderstand zur Strombegrenzung. 220 Ohm würden auch hier passen,

aber dann leuchtet die LED sehr hell. Indem man den Wert auf 470 Ohm oder noch ein bisschen mehr erhöht, kann man sie auf eine für eine Kontrollanzeige angemessene Helligkeit dimmen.

## Schaltung

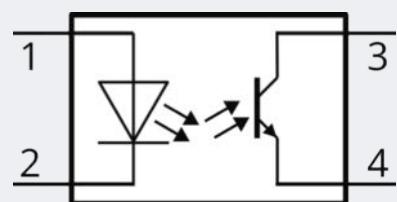
Eine Skizze der Schaltung zeigt das Bild auf der nächsten Seite. Entstanden ist sie mit der Webanwendung Cirkit Designer. Die Projektdatei ist im Downloadpaket zu diesem Artikel enthalten (alle Links und Downloads via ct.de/yqx5); im Cirkit Designer können Sie hineinsehen, indem Sie ein neues Projekt starten und die Datei dann über den Menübefehl „File/Import/Import Local Cirkit File“ hochladen.

Als Mikrocontroller sieht die Schaltung ein ESP8266-Bastelboard vom Typ D1 Mini vor. Das gibt es mit unterschiedlichen Markenaufdrucken wie Wemos, AZ-Delivery oder Berrybase. Deutsche Elektronikhändler verkaufen es zu Preisen ab etwa 4,50 Euro, im Chinaversand ist es teilweise schon für unter 1 Euro zu haben

## Optokoppler

Ein Optokoppler vereint in einem nach außen lichtundurchlässigen Gehäuse eine Leuchtdiode und einen Phototransistor. Letzterer wird leitend, wenn durch die LED ein Strom fließt und sie zum Leuchten bringt. Elektrisch sind die beiden Elemente voneinander isoliert, sodass man mit einer an die LED angelegten Spannung einen Strom in einer anderen Schaltung steuern kann, ohne dass unterschiedliche Potenziale stören könnten.

Gängige Optokoppler für Digitalschaltungen kommen in einem vierpoligen IC-Gehäuse; es gibt auch ICs mit mehreren Kopplern in einem Gehäuse und entsprechend mehr Pins. Wie jede LED etwa für eine Leuchtanzeige braucht auch die in einem Optokoppler einen Vorwiderstand zur Strombegrenzung.



(plus Versand, Steuern und Wartezeit, versteht sich). Geliefert wird es üblicherweise mit mehreren Sätzen Steckverbindern zum Anlöten; wir haben uns für die doppelte Ausführung mit einer Stiftleiste zum Einstechen in das Breadboard unten und einer Buchsenleiste für Jumper-Kabel oben entschieden.

Aus dem Schaltbild wird ersichtlich, dass der GPIO-Pin D1 des ESP-Moduls die LED im Optokoppler ansteuert. Indem die Software ihn auf HIGH setzt, simuliert sie eine gedrückte Taste an der Lampe. Die beiden Kontakte des Tasters sind dazu mit dem Ausgang des Optokopplers verbunden. Messen Sie beim Nachbau nach, an welchem Anschluss des Tasters Plus anliegt (bei unserem Exemplar war es die untere; siehe Foto auf S. 94), und verbinden Sie diesen mit Pin 3 des Optokopplers (Kontakte A17 bis D17 auf dem Steckbrett in der Skizze), den anderen mit Pin 4 (A16 bis D16).

Die RGB-LED bekommt ihren Strom über je einen Vorwiderstand direkt aus den Pins D5 bis D7 des ESP8266. Welcher genau für welche Farbe zuständig ist, entnehmen Sie dem Datenblatt Ihres Exemplars oder probieren es einfach aus; der

längste Anschlussdraht ist immer der für die gemeinsame Kathode beziehungsweise Anode.

Der Schaltplan zeigt noch ein paar zusätzliche Leitungen und einen Widerstand, die sich aber als unnötig erwiesen haben: Sie sollten ursprünglich dazu dienen, mithilfe eines Photowiderstands zu überwachen, ob die Lampe auch wirklich ein- beziehungsweise ausgeschaltet wurde. Im Testbetrieb hat sich herausgestellt, dass man auf die Überwachung verzichten kann. Wenn Sie es trotzdem probieren möchten, können Sie zwischen einem der Anschlüsse H11 bis J11 des Breadboards und Masse einen Photowiderstand mit einem Dunkel-Nennwert von 10 kOhm anschließen und die Software dahingehend erweitern, dass sie den ESP-Pin A0 als Analogeingang konfiguriert und ausliest.

Einbau

Das Gehäuse der Lampe zu öffnen, ist ein bisschen fummelig. Es gelingt am besten mit einem spitzen Messer oder ähnlichem Gerät, mit dem man vorsichtig in die Fuge um die Rückwand fährt und die Kunststoffnasen aushebt, die die Teile zusam-

menhalten. Wer sich die Prokeli nach dem ersten Öffnen sparen will, knipst die Nasen mit einem Seitenschneider ab und klebt stattdessen einen Magnetschnäpper und sein eisernes Gegenstück aus dem Baumarktregal für Möbelbeschläge in das Gehäuse beziehungsweise von innen auf die Rückwand.

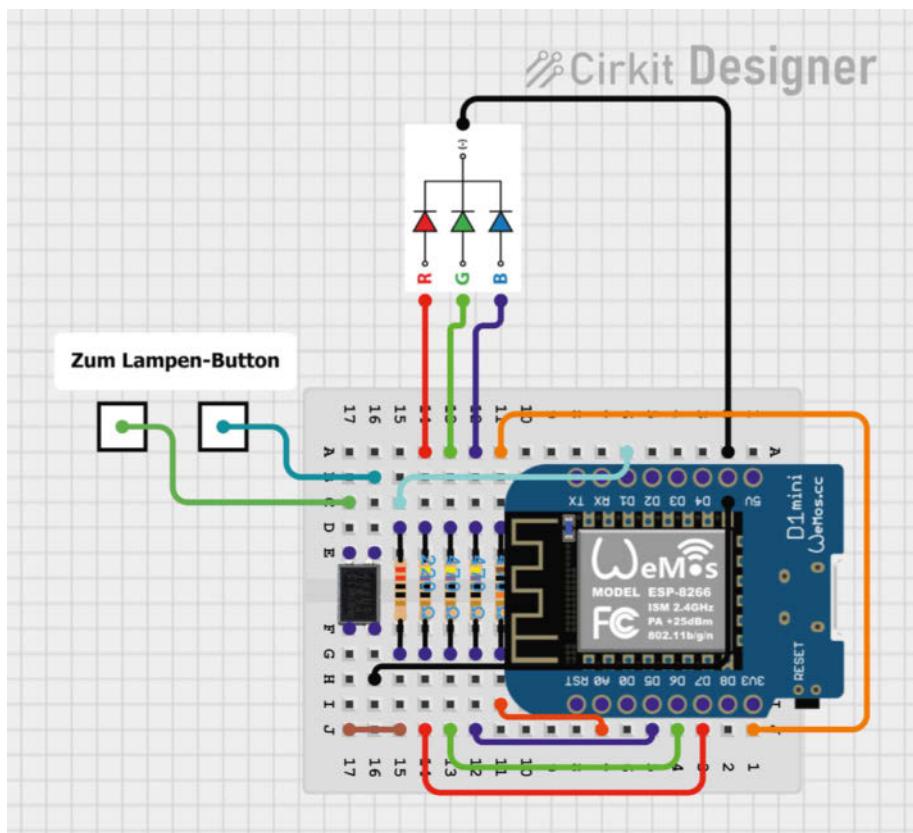
Für die Stromversorgung unserer Schaltung haben wir den USB-Anschluss der Originalelektronik in der Lampe an-gezapft. Der liefert 5 Volt und ist mit den Pins 5V und G (für „Ground“, also Masse) des ESP-Boards zu verbinden. Diese Anschlüsse sowie die Verbindung zum Lampentaster haben wir hergestellt, indem wir von Jumper-Kabeln jeweils an einem Ende den Steckpin abgeschnitten und das ab-isolierte Kabel an die entsprechende Löt-stelle auf der langen Platine in der Lampe angelötet haben. Achtung: Wenn Sie nach dem Einbau in die Lampe noch einmal die USB-Schnittstelle des ESP-Boards benut-zen, etwa zur Fehlersuche oder um eine neue Firmware aufzuspielen, trennen Sie zuvor unbedingt den 5-Volt-Anschluss – ein ESP8266 mag es nicht, aus zwei Quel-len gleichzeitig versorgt zu werden.

Die ferngesteuerte Lampe lässt sich nach dem Zusammenbau über die Mikro-USB-Buchse des Gehäuses aus einem gewöhnlichen USB-Netzteil mit Strom versorgen. Wo die baulichen Gegebenheiten einer Verkabelung im Wege stehen, kann man auch eine USB-Powerbank an das Lampengehäuse anflanschen.

## Firmware

Die Software, die auf dem ESP8266 läuft, birgt keine großen Geheimnisse. Sie ist in der Arduino-IDE entstanden und größtenteils aus Beispielen zusammenkopiert, die die Entwicklungsumgebung bereits mitbringt. Sie besteht im Wesentlichen aus einem per WLAN erreichbaren Webserver, der über bestimmte URLs die Lampe ein- und ausschaltet und deren Status zurückmeldet.

Um sie zu kompilieren und in den Mikrocontroller zu übertragen, benötigen Sie die Arduino-IDE, die es auf arduino.cc zum kostenlosen Download gibt. Von Haus aus kennt sie aber keine ESP8266-Boards. Um ihr deren Spezifika beizubringen, müssen Sie sie um die dazu nötigen Bibliotheken ergänzen. Dazu öffnen Sie die Preferences über den gleichnamigen Befehl im File-Menü und tragen „[https://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](https://arduino.esp8266.com/stable/package_esp8266com_index.json)“ (Kopiertvorlage



**Die Schaltung zum Fernsteuern der Lampe passt auf ein Mini-Steckbrett, das bequem im Gehäuse Platz findet.**



# .NET 9.0

**Das Online-Event zum neuen .NET-Release**

**19. November • Online**

Was sich im .NET-Ökosystem mit Version 9.0 geändert hat und was bei der Migration zu beachten ist, behandelt die betterCode() .NET 9.0. Dort präsentieren .NET-Experten den fertigen Stand des neuen Release anhand von Praxisbeispielen.

Die Konferenz bietet Expertenvorträge zu folgenden Themen:

- Die Neuerungen von .NET 9.0: SDK, Runtime und Basisklassen
- Einfacher lesbarer, stabilerer Code mit C# 13.0
- Alle Neuerungen von ASP.NET Core 9.0 und Blazor 9.0
- Neues beim OR-Mapping mit Entity Framework Core 9.0
- Das hat sich mit Windows Forms 9.0, WPF 9.0 und WinUI verändert
- Cross-Plattform-Entwicklung mit .NET MAUI 9.0



**Jetzt  
Tickets  
sichern!**

Sechs Workshops Ende November/Anfang Dezember

[net.bettercode.eu](http://net.bettercode.eu)

# PHP 2024

**21. November • Online**



**Mach deine PHP-Anwendung fit für 2025**

**Highlights aus dem Programm:**

- Release Manager Saki Takamachi über die Neuerungen in PHP 8.4
- PHPUnit 10-12: Was du wissen und was du vergessen solltest
- Sichere Software-Lieferketten für PHP-Entwickelnde
- Time Travel: Kann deine Software durch die Zeit reisen?



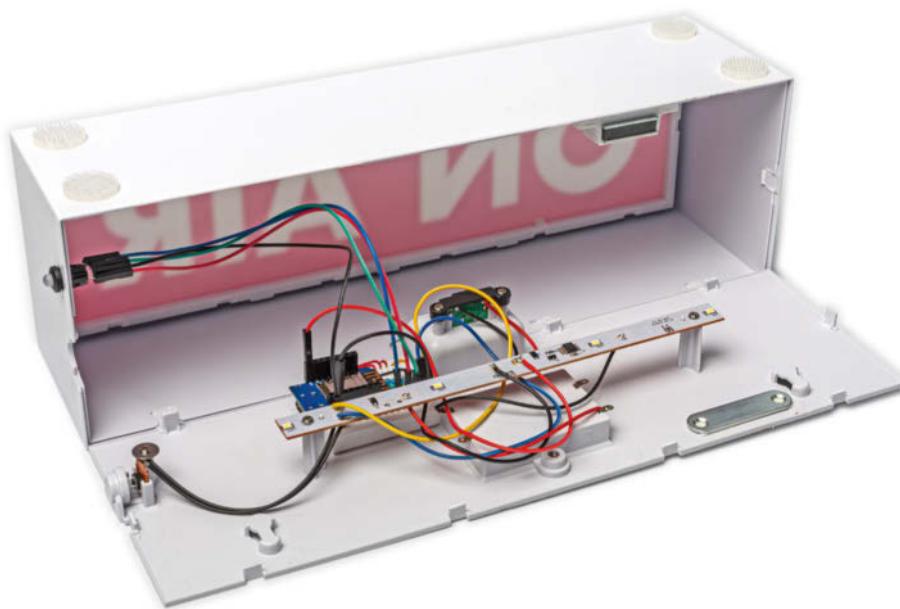
**Jetzt  
Tickets  
sichern!**

Kooperationspartner



**Workshop am 22. November:  
»Containerisierte Anwendungen sicher betreiben«**

[php.bettercode.eu](http://php.bettercode.eu)



**Das geöffnete Lampengehäuse, hier schon mit eingebauter Zusatzelektronik. Der Magnetschnäpper ersetzt die Plastiknasen, die die Rückwand ursprünglich festhalten, und erleichtert das wiederholte Öffnen.**

unter ct.de/yqx5) in das Feld „Additional boards manager URLs“ ein; sollte das Feld nicht leer sein, trennen Sie den neuen Eintrag mit einem Komma von den bestehenden. Anschließend öffnen Sie über das zweite Symbol in der linken Toolbar oder den Menübefehl „Tools/Boards/Boards Manager“ ebenen und tippen oben in dessen Suchfeld „ESP8266“ ein. In der Liste der Bibliotheken sollte genau ein Eintrag mit dem Titel „esp8266 by ESP8266 Community“ übrigbleiben – klicken Sie auf dessen „Install“-Button. Nachdem die Installation abgeschlossen ist, können Sie das D1-Mini-Board über den Menüeintrag „Tools/Board/esp8266/LOLIN(WEMOS) D1 mini (clone)“ auswählen.

Damit Windows mit dem Board kommunizieren kann, braucht es noch einen Treiber für dessen USB-Schnittstellenbaustein CH340. Laden Sie ihn via ct.de/yqx5 herunter, entpacken Sie die ZIP-Datei und führen die enthaltene SETUP.EXE aus. Wenn Sie das Board danach an Ihren PC anschließen, sollte die Arduino-IDE das mitbekommen und die Schnittstelle als COM3, COM4 oder ähnlich im Menü „Tools/Port“ anbieten – wählen Sie sie aus, um die serielle Verbindung herzustellen.

Unsere Lampen-Firmware liegt unter ct.de/yqx5 bereit. Kopieren Sie das Verzeichnis „OnAirLamp“ aus der ZIP-Datei am besten komplett in den Ordner Dokumente\Arduino auf Ihrem PC, dann können Sie den Quelltext in der Arduino-IDE bequem über das Menü „File/Sketch-

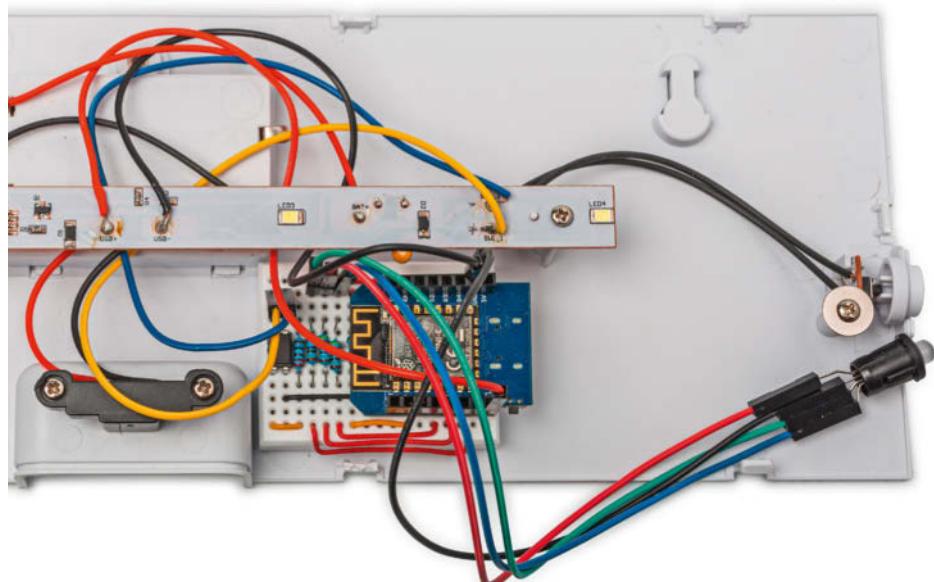
book“ laden. Bevor Sie die Software in Ihrem Zuhause benutzen können, müssen Sie noch die Zugangsdaten zu Ihrem WLAN in die Definition der Variablen WifiSsid und WifiPwd eintragen.

Gleich darunter finden Sie noch die mit „onairlamp“ vorbelegte Variable ServerName, die Sie ebenfalls anpassen können, wenn Sie mögen. Den Namen brauchen Sie gleich, um die Windows-Software zu konfigurieren, die mit der Lampe spricht. Unsere Firmware meldet sich

nämlich über das Protokoll mDNS (Multicast DNS) im Netzwerk an, sodass die Lampe unter <http://onairlamp.local> erreichbar ist – „onairlamp“ können Sie ändern, „.local“ ist von mDNS vorgegeben. Die Namensauflösung hat bei unseren Tests aber nicht in allen Windows-Installationen funktioniert. Ein frisches Windows 11 scheint keine Probleme zu haben, per Upgrade aus Vorversionen gewachsene Installationen haken manchmal. Dann müssen Sie die Lampe über ihre IP-Adresse ansprechen und können den Servernamen benutzen, um die in der Verwaltungsoberfläche Ihres WLAN-Routers nachzuschlagen.

Wenn die Lampen-Firmware startet, sollte die RGB-LED für ein paar Sekunden gelb blinken und dann in ein grünes Dauerleuchten übergehen – dann ist der Webserver bereit, Anfragen entgegenzunehmen. Rotes Blinken deutet auf Verbindungsprobleme hin, etwa falsche Zugangsdaten oder eine zu schwache Verbindung. Die Lampen-URL können Sie testweise auch in einen Browser eingeben, dann meldet die Lampe ihren Status zurück. Über die Adresse <http://onairlamp.local/on> können Sie sie ein- und über <http://onairlamp.local/off> ausschalten.

Der Quelltext sollte sich weitgehend selbst erschließen, wenn Sie schon mal unter Arduino programmiert haben. Einen Einstieg in die Programmiersprache C lesen Sie ab Seite 62. Die eigentliche Arbeit – das Drücken der Lampen-Taste – er-



**Die Verbindungsdrähte zwischen Originalelektronik und Controller-Schaltung haben wir lampenseitig an Kontakte der ursprünglichen Platine angelötet. Von den beiden Anschläßen zum Taster ist bei unserem Exemplar der untere Plus.**

ledigen die Funktionen `lampOn()` und `lampOff()`. Die Dauer der Tastendrücke und die Wartezeiten dazwischen haben sich in unseren Experimenten bewährt; Sie können sie ändern, indem Sie die Argumente der `delay()`-Aufrufe anpassen. Dass `lampOff()` die Taste zweimal drückt, kommt daher, dass die Lampe die drei Betriebsmodi An, Aus und Blinken kennt; ein Tastendruck wählt den jeweils nächsten Zustand aus. Wenn Sie möchten, dass die Lampe lieber pulsiert als dauerleuchtet, um eine Verbindung zu signalisieren, tauschen Sie einfach die beiden Funktionsnamen.

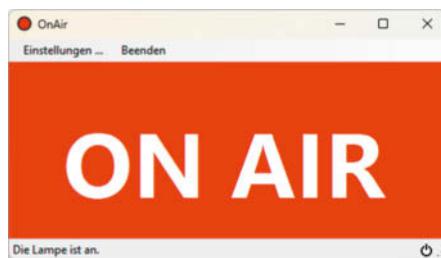
Falls Sie die HTML-Antwort der Lampe aufhübschen wollen (Funktion `prepareAnswer()`), achten Sie bitte darauf, dass sie auf jeden Fall einen `<span>`-Tag mit der ID `lampstate` enthalten muss. An dessen Inhalt erkennt die Windows-Software den Schaltzustand.

## Windows-Software

Der Windows-Teil des Projekts besteht aus einem gewöhnlichen Desktopprogramm namens „OnAir“, das in C# und .NET geschrieben ist. Installieren können Sie es, indem Sie den Ordner OnAir aus dem Downloadpaket komplett irgendwo auf den Systemdatenträger des Rechners kopieren. Das Programm benötigt eine .NET-Runtime; wenn die passende Version in Ihrem Windows fehlt, besorgt sich das Programm beim ersten Start nach Nachfrage die nötigen Dateien vom Microsoft-Server und installiert sie automatisch. Wie Sie das Programm ins Startmenü oder den Autostart bekommen, überlassen wir Ihnen; die Deinstallation ist mit dem Löschen des Programmordners erledigt.

Wenn Sie den Servernamen in der Lampen-Firmware unverändert gelassen haben und auf Ihrem Windows mDNS funktioniert, sollte das Programm auf Anhieb mit der Lampe kommunizieren und deren Zustand in der Statuszeile vermelden, anderenfalls erscheint dort eine Fehlermeldung. Die Lampen-URLs können Sie über den Einstellungen-Dialog anpassen; ohne mDNS funktioniert dort auch so etwas wie „<http://192.168.178.42>“, also eine IP-Adresse statt eines Host-Namens. Mit der kleinen On/Off-Schaltfläche ganz rechts unten im Fenster können Sie die Lampe per Mausklick auch manuell fernsteuern.

Im Quelltext (Ordner OnAirSource im Download) ist vor allem die Klasse `WebCamWatcher` interessant: Mit ihr findet



**Das zum Projekt gehörige Windows-Programm zeigt den Status der Webcam an und überträgt ihn an die Lampe. Wenn es Ihnen zu auffällig ist, minimieren Sie es in den System-Tray.**

das Programm heraus, ob eine an den Rechner angeschlossene Kamera in Betrieb ist. Eine offiziell dokumentierte Schnittstelle gibt es dafür in Windows nicht. Man kann sich aber mit einem Trick behelfen: Alle Anwendungen, die potenziell auf die Kamera zugreifen dürfen, sind in der Registry in Unterschlüsseln von „HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\CapabilityAccessManager\ConsentStore\webcam“ verzeichnet, die installierten Windows-Store-Apps direkt und gewöhnliche Win32-Anwendungen im Unterschlüssel „NonPackaged“.

Zu jeder dieser Anwendungen speichert Windows in Einträgen namens `LastUsedTimeStart` und `LastUsedTimeStop`, wann sie die Kamera zuletzt benutzt hat. Hat ein Programm die Kamera gerade in Benutzung, ist in ihrem Schlüssel eine Startzeit vorhanden und die Stop-Zeit 0. Um herauszufinden, ob die Kamera gerade an ist, kann man also den Registry-Ast nach solchen Null-Einträgen abklappern.

Für ein Programm, das den Kamerazustand dauerhaft überwachen soll, wäre es eine ziemliche Verschwendug, das regelmäßig alle paar Sekunden oder noch öfter zu tun. Für solche Zwecke hat Windows aber eine Lösung parat: die WMI-Events [1]. Über die „Windows Management Instrumentation“ kann man das System unter anderem beauftragen, sich zurückzumelden, wenn sich bestimmte Aspekte des Systemzustands geändert haben, zum Beispiel ein bestimmter Prozess gestartet oder ein Datenträger angelassen wurde. Auch Änderungen am Inhalt eines Bereichs der Registry lassen sich damit überwachen.

In .NET stehen solche WMI-Events über die Klasse `ManagementEventWatcher` aus dem Namespace `System.Management` zur Verfügung. Die richtige Ereignisklasse für

den angepeilten Zweck ist ein `RegistryTreeChangeEvent`. Der funktioniert allerdings nicht im Zweig `HKEY_CURRENT_USER`, weshalb sich das Programm zunächst die SID („Security Identifier“, so etwas wie eine interne Anmeldekennung) des aktuellen Benutzers besorgt und damit dessen Registry-Ast unter `HKEY_USERS` identifiziert.

Mit dem Wissen ist die Klasse `WebCamWatcher` gar nicht mehr so kompliziert zu implementieren. Ihre Programmierschnittstelle besteht aus den Methoden `Watch()`, die als Parameter einen Delegaten auf eine Handler-Funktion erwartet, und `EndWatch()`, die die Überwachung wieder beendet. Zusätzlich gibt es noch die statische Hilfsfunktion `GetAppsUsingCam()`, die eine Liste der Anwendungen liefert, in welchen gerade die Kamera aktiv ist.

Der Rest des OnAir-Programms ist Brot-und-Butter-Programmierung. Als Grundlage dient das WinForms-Framework und keine der neueren, XAML-basierten Bibliotheken, weil Plattformunabhängigkeit kein Thema ist und das Programm sich als Symbol im System-Tray neben der Uhr präsentieren soll, solange es läuft. Die dafür zuständige .NET-Klasse `NotifyIcon` gibt es nur in WinForms.

## Ans Werk!

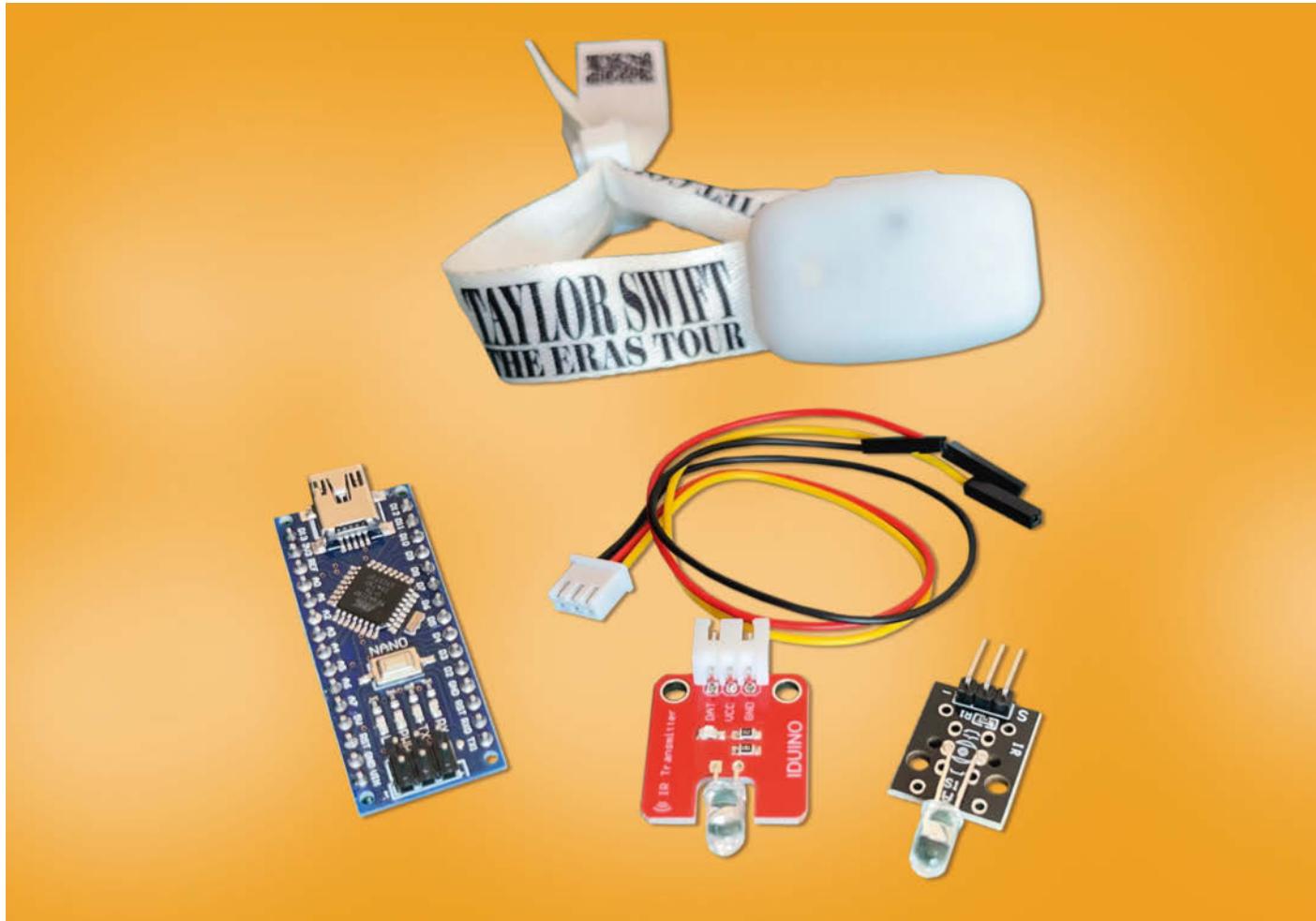
Software, die unseren Lampen-Umbau auch mit anderen Betriebssystemen als Windows nutzbar macht, gibt es noch nicht. Allzu schwer sollte sie aber nicht zu programmieren sein. Schließlich gibt sich die Lampe selbst als gewöhnlicher Webserver, und der Abruf von Webseiten sollte von so ziemlich jedem System aus und mit jeder ernst zu nehmenden Programmiersprache möglich sein. Wenn Sie dieser Artikel inspiriert hat, ein solches Programm für Ihr Leib-und-Magen-System zu schreiben, würden wir uns über eine Nachricht freuen.

Windows-Nutzer brauchen nicht viel mehr als ein bisschen Bastelgeschick, um das Projekt umzusetzen – die Software ist ja schon fertig. Viel Spaß beim Nachbau!

(hos@ct.de)

## Literatur

[1] Hajo Schulz, Nichts mehr verschlafen, Windows mit WMI-Events überwachen, c't 3/2019, S. 178



# Neues Leben für Konzertarmbänder

**Mit Arduino und Infrarot-LED leuchtet es wieder am Handgelenk**

**Blinkende LED-Armänder haben einen festen Platz bei vielen Superevents gefunden – funktionieren aber meistens nur vor Ort. Wer sein Konzertmitbringsel auch zu Hause ansteuern möchte, braucht dafür nur zwei Komponenten und wenige Minuten Zeit.**

**Von Jack Pop  
und Anne-Dorette Ziems**

Die Fan-Blöcke funkeln wie Sterne, dann erstrahlt die gesamte Menschenmenge rot: Unzählige Videos der Publikumslightshow von Taylor Swifts „The Eras Tour“ haben die sozialen Medien geflutet und neues Interesse für LED-Konzertarmbänder geweckt. Viele Besucherinnen und Besucher nehmen solche Gadgets als Erinnerungsstücke mit nach Hause. Das Problem: Die Steuersignale sind proprietär und die Armbänder scheinen außerhalb des Stadions nutzlos.

Mit nur zwei Elektronikkomponenten und wenigen Arbeitsschritten kann man sie

aber wieder zum Leuchten bringen. Denn seit zwei Jahren sammelt eine enthusiastische Maker-Community die nötigen Steuerbefehle im Netz zusammen (mehr dazu im Infokasten „Reverse Engineering“). Auf GitHub stellt sie ein äußerst umfangreiches Repository mit verschiedensten Anwendungsmöglichkeiten zur Verfügung [1] (alle erwähnten Links unter ct.de/yvqh). Hier finden sich zum Beispiel simple Python-skripte und Programmskripte für Arduino und ESP32, sogenannte Sketches.

Für Gelegenheitsbastler besteht die einfachste Variante aus einem Arduino,



Bild: Natasha Leake/PA Wire/dpa

## c't kompakt

- Die leuchtenden Armbänder von Taylor-Swift-Konzerten oder anderen Großevents werden mithilfe von Infrarotstrahlung angesteuert und kontrolliert.
- Ein Infrarot-Transmitter gesteuert von einem Arduino bringt die Armbänder auch zu Hause wieder zum Leuchten.
- Skripte und sogar Chatbots, die beim Programmieren der Armbandsteuerung helfen, erstellt das Pix-Mob-Reverse-Engineering-Projekt, eine globale Maker-Community.

**Taylor Swifts „The Eras Tour“ war die kommerziell erfolgreichste Tour überhaupt. Doch nicht nur sie nutzt Leuchtarmbänder, sondern auch Coldplay und die Veranstalter der Olympischen Spiele in Paris.**

einem Infrarot-LED-Modul und einem auf GitHub verlinkten Webinterface, das die Steuersignale integriert hat. Damit konnten wir nach nur zehn Minuten die Armbänder wieder aufleuchten lassen. Zu Beginn noch ein Hinweis: Das GitHub-Projekt ist auf die Armbänder von Marktführer PixMob spezialisiert, die nicht nur bei Taylor Swift, sondern auch bei der aktuellen Coldplay-Tournee oder den Olympischen Spielen in Paris zum Einsatz gekommen sind. Wir haben das Projekt allerdings nur mit dem Eras-Tour-Modell, X2, getestet.

### Zwei Hardwarekomponenten

Da die Übertragung der Farbsignale beim Konzert über Infrarotstrahler funktioniert (mehr dazu im Infokasten „Datenübertragung per Infrarot“), mussten wir einen solchen zunächst klein nachbauen. Dafür genügen zwei Komponenten: Als Steuerinheit haben wir uns für einen Arduino Nano entschieden (ähnliche Modelle anderer Anbieter funktionieren in der Regel jedoch auch) und als Signalübermittler für ein 5 V-Infrarot-Transmittermodul.

Grundsätzlich lässt sich das Transmittermodul auch durch eine einzelne Infrarot-LED ersetzen, die als Cent-Artikel sogar noch preiswerter ist. Allerdings kommt diese Variante mit einigen technischen Hürden: Die Bastlerin oder der Bastler muss den Vorwiderstand berechnen und in die Schaltung integrieren, damit die LED im Betrieb

nicht durchbrennt. Außerdem muss man die Stromstärke der LED berücksichtigen, um den Arduino nicht zu überlasten.

Wer sich das ersparen und den Griff zum Lötkolben vermeiden möchte, kommt mit dem Transmittermodul am schnellsten ans Ziel; dieses kann man direkt mit dem Arduino verbinden. Das dreipolige Verbindungskabel war bei unserem Modul im Lieferumfang enthalten, aber man

kann Module mit freiliegenden Anschlüssen auch mit einzelnen Jumper-Kabeln verbinden.

Hat man beide Komponenten beisammen, ist der Anschluss fast selbsterklärend: Das Verbindungskabel wird so an das Transmittermodul gesteckt, dass die Ader GND an einem der beiden GND-Pins des Arduino steckt. Die Ader VCC kann ignoriert werden. Diese regelt üblicherweise die Versorgungsspannung von Bauteilen mit größerer Last, der Stromfluss zur LED vom Arduino ist aber schon über die DAT-Ader gewährleistet.



Bild: Marius Becher/dpa

**Die vor der Show verteilten LED-Armbänder nehmen viele Fans als Erinnerung mit nach Hause.**

## Projektinfo

- Mit wenigen Handgriffen das LED-Armband vom letzten Taylor-Swift-Konzert wieder zum Leuchten bringen
- Programmieren Arduino-IDE
- 5 bis 15 Minuten
- 25 Euro
- Pixmob X2 Armband  
Arduino Nano (oder ähnliches Board)  
5 V-Infrarot-Transmittermodul
- Keins

Die DAT-Verbindung wird am Arduino auf den Pin D4 gesteckt. Prinzipiell können auch andere freie Output-Pins des Arduino genutzt werden, Pin 4 ist lediglich die Voreinstellung im Skript. Wer einen anderen Anschluss-Pin bevorzugt, muss später im Programmskript die Pin-Nummer im Code ändern.

### Der Arduino-Sketch

Ist alles zusammengesteckt, kann der Arduino per USB-Kabel mit dem Computer verbunden werden. Wer die frei verfügbare Arduino-IDE noch nicht auf dem Rechner hat, muss diese zunächst herunterladen und installieren (Link auf ct.de/yvqh). Für Arduino-Neulinge gibt es online kurze, grundlegende Erklärungen für die Verbindung zwischen IDE und dem Arduino [2].

Nun fehlt noch der Arduino-Sketch: ein Skript, das die serielle Kommunikation zwischen Computer und Arduino



Bild: Screenshot LinkedIn / PixMob

**Während der Abschlusszeremonie der Olympischen Spiele in Paris 2024 wurde das Publikum dank Infrarotarmbändern zur lebenden Leinwand.**

überwacht, eingehende Befehle in Sendesignale für das Infrarotmodul übersetzt und dieses ansteuert. Dieses Skript findet man im GitHub-Repository [1] im Ordner „arduino\_sender/PixMob\_Transmitter\_Arduino“.

Bevor man dieses auf den Arduino lädt, gibt es allerdings noch zwei Dinge zu prüfen: zum einen, ob in Zeile 39 tatsächlich der korrekte Daten-Pin hinterlegt ist. Die Voreinstellung sollte hier „4“ sein.

```
38 // SET THIS TO THE DATA PIN USED
FOR THE IR TRANSMITTER
39 const uint16_t kIrLed = 4;
```

Zum anderen fehlt noch eine Programmabibliothek, die innerhalb der Arduino-IDE für die Infrarotkommunikation verantwortlich ist. Dafür klickt man in der linken Seitenleiste auf das Bücher-Icon und gibt direkt oben in der Suchleiste „IRremote“ ein. Es sollte die Bibliothek „IRremote by shiriff, z3t0, ArminJo“ angezeigt werden, die nun installiert werden sollte. Danach kann man den Sketch per Upload-Button auf den Arduino laden.

### Steuerung über ein Webinterface

Wie angenehm zugänglich das PixMob-Reverse-Engineering-Projekt ist, zeigt sich im letzten Schritt: Online findet sich nämlich ein simples Webinterface, mit dem sich die Farbcodes für das Armband aus dem Browser heraus senden lassen [3]. Wichtig: Der verwendete Browser muss „Web Serial API“ unterstützen, das die serielle Kommunikation mit dem Arduino

ermöglicht. In Chrome, Opera und Edge läuft die Funktion schon seit 2021; Safari und Firefox lassen sich für dieses Projekt (noch) nicht nutzen. Den Implementationsstatus für gängige Browser findet man online (Link auf ct.de/yvqh).

Im Webinterface stellt der Button „Connect to board“ eine Verbindung zum Arduino her. Dann öffnet sich ein Pop-up, in dem man den Port „USB Serial“ auswählt und auf „Verbinden“ klickt – damit ist die Einrichtung komplett. Um das Armband anzusteuern, wählt man einen „Main Effect“ aus der oberen Dropdown-Liste aus und schickt den Befehl über den „Send“-Button los. Das Armband sollte nun aufblinken. Die zweite Dropdown-Liste lässt einen „Tail Code“ auswählen: einen Zusatzbefehl, der – wie es der Name schon sagt – ans Ende des Farbbefehls geschrieben wird. Zuverlässig funktionieren vor allem die Fade-Effekte, bei denen die Farbe sanft aufleuchtet und abblendet.

### Leuchtbefehle verbessern

An dieser Stelle wird ein kleiner Makel des PixMob-Reverse-Engineering-Projekts deutlich: Die aufgezeichneten, dokumentierten und im Webinterface implementierten Farbcodes stammen von unterschiedlichen Tourneen und Geräteversionen aus verschiedenen Jahren. Armbandhersteller PixMob aktualisiert und individualisiert seine Leucht-Gadgets ständig, sodass viele Befehle des Webinterface in unserem Test mit dem Taylor-Swift-Armband nicht (mehr) funktioniert haben. Außerdem erscheint die Bedienung – Liste aufmachen, durchscrollen,

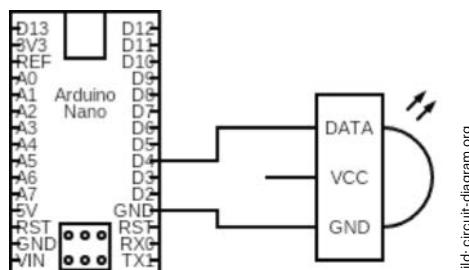
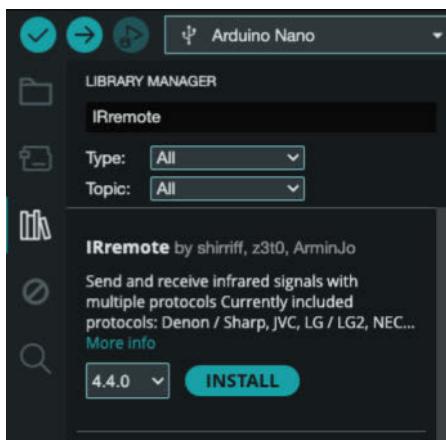


Bild: circuitdiagram.org

**Die Schaltung zwischen Arduino und Transmittermodul benötigt nur zwei Kabel.**



**Arduino-IDE ist die Arduino-App, mit der Programmksripte erstellt, bearbeitet und anschließend auf den Arduino geladen werden, wie zum Beispiel die Bibliothek „IRremote“.**

Effekt auswählen, senden – etwas umständlich.

Deshalb haben wir uns durch Dutzende Code-Listen, RGB-Tabellen, Demo-Versionen und Protokolle gearbeitet und eine übersichtlichere, erweiterte Web-Oberfläche zur Ansteuerung der Armbänder programmiert. Online haben wir 27 Eras-Tour-Farben und sechs Animations-Effekte hinterlegt [4]. Einer davon ist das sogenannte „letzte Signal“, das die Ver-

anstalter am Ende jedes Konzerts als eine Art Abschiedsgruß an die Armbänder schickten, um sie in einen dauerhaften Farbwechsel zu versetzen.

Wer selbst Hand anlegen möchte, sei hiermit ermutigt, unsere Version zu ergänzen und zu individualisieren: Unser Quellcode ist offen einsehbar, kommentiert und wie auch das PixMob-Reverse-Engineering-Projekt unter der MIT-Lizenz frei verfügbar. Wer lieber an den unbearbeiteten Originalcode möchte, findet diesen im GitHub-Repository [1] im Ordner „www“.

### Noch simpler mit Android-Handy

Mit etwas Glück geht es noch einfacher: Unter bestimmten Umständen lassen sich die LED-Armbänder nämlich auch mit dem Smartphone steuern. Dafür muss das Smartphone einen sogenannten Infrarot-Blaster besitzen. Das ist ebenfalls eine Infrarot-LED – nur eben im Smartphone verbaut. Auf Wikipedia findet sich eine Liste mit Mobiltelefonen, die damit ausgestattet sind (Link auf ct.de/yvqh). Dabei sind diverse Handys, die aus der Zeit vor der Smartphoneära stammen, aber auch einige aktuelle Android-Geräte.

Wer sein Handy in der Liste findet, kann sich im nächsten Schritt eine App he-

runterladen, um den Infrarot-Blaster zu steuern. Wir haben dazu die App „irplus Infrarot Fernbedienung“ genutzt. Analog zur Arduino-Variante haben wir online eine Datei hinterlegt, die man mit der App öffnet, um die Armbänder der Eras-Tour über farbige Buttons aufleuchten zu lassen [4]. Auch diese Datei basiert auf der Arbeit des PixMob-Reverse-Engineering-Projekts [1].

### Deep Dive

Für Power-Bastler enden die Möglichkeiten damit noch nicht. Es wäre zum

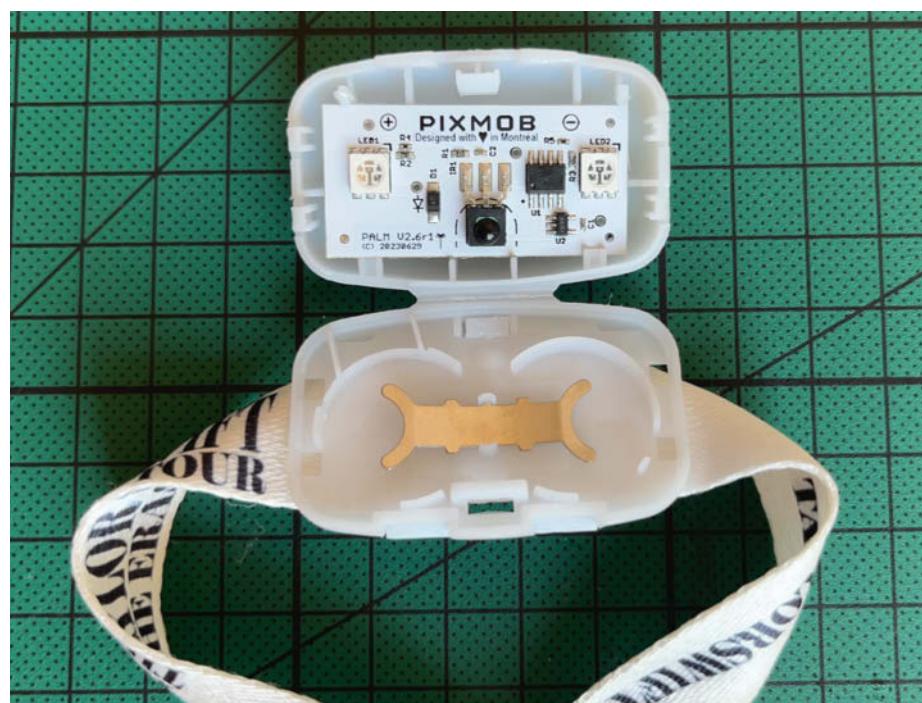
## Reverse Engineering

Um herauszufinden, welche Infrarotsignale die Armbänder zum Leuchten bringen, haben die Mitglieder des Pix-Mob-Reverse-Engineering-Projekts [1] die Signale, die während verschiedener Konzerte an die Armbänder gesendet wurden, mit speziellen Empfangsgeräten aufgezeichnet. Allerdings haben im Test von über 100 registrierten Signalen nur drei eine direkte Reaktion des Armbands hervorgerufen.

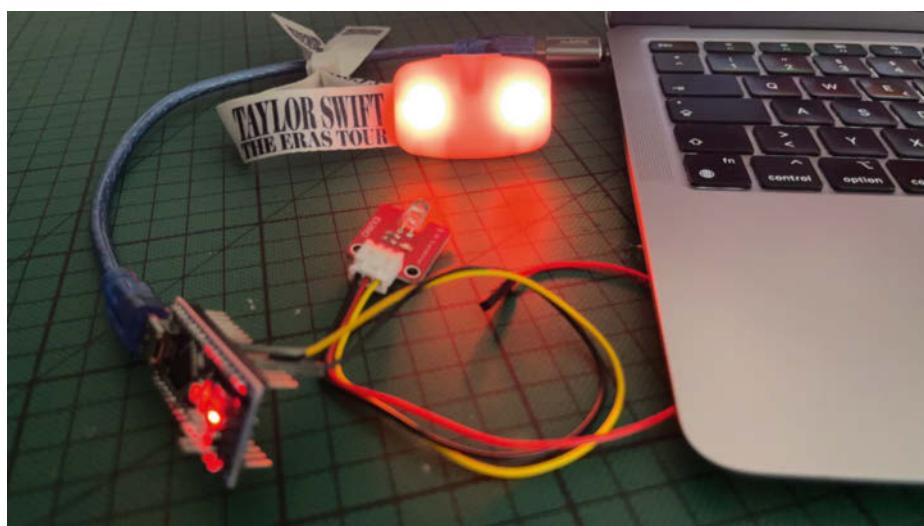
Die GitHub-Community vermutete, dass die meisten Signale, die während einer Show gesendet werden, keine direkten Befehle für bestimmte Farben sind, sondern Steuercodes. Diese signalisieren dem Armband, was zu tun ist, wenn es nur bruchstückhafte Farbbefehle erhält.

Immerhin konnte die Community so aber rekonstruieren, wie die Signale grundsätzlich aufgebaut sind. Mit einem Brute-Force-Ansatz – das heißt durch wildes Ausprobieren – hat die Gruppe dann weitere Befehle für verschiedene Farben und Modi herausgefunden.

Das PixMob-Reverse-Engineering-Projekt entwickelt sich ständig weiter, denn auch der Armbandhersteller Pix-Mob veröffentlicht fortlaufend neue Generationen der Armbänder und implementiert neue Effekte. Wenn Sie jetzt die Motivation gepackt hat, können Sie selbst am PixMob-Reverse-Engineering-Projekt teilnehmen, indem Sie zum Beispiel Fotos ihres eigenen restaurierten Leuchtarmbands auf den Discord-Server hochladen oder Abwandlungen des Web-Interfaces im GitHub-Repository anbieten.



**Die LED-Armbänder sind im ständigen Empfangsmodus und verbrauchen somit auch Strom, wenn sie nicht leuchten. Im Zweifelsfall kann man das Plastikgehäuse öffnen und die Batterien tauschen.**



die Lautstärke unterschiedliche Farbsignale triggern.

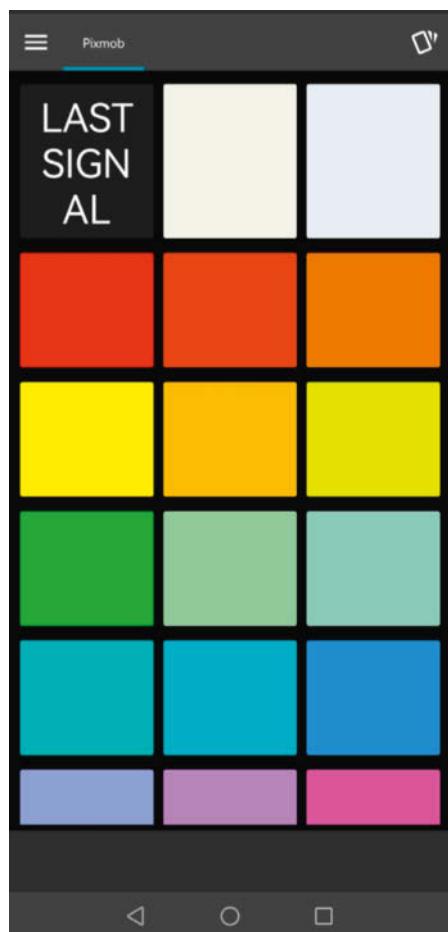
Weitere Inspiration gibt es auf dem Discord-Server „PIXMOD: Reverse Engineering PixMob!“. Er bietet als verlängerter Arm des Repository mehr dokumentierte Beispiele anderer Maker. Ein begeisterter User hat sogar einen Chatbot mit dem Namen PIXMOB-GPT entwickelt, der unter anderem Quellcode zum Steuern der Armbänder erstellt. All das zeigt, wie Einzelne getrieben von einer Leidenschaft zusammenfinden können – wie auch bei der Eras-Tour.

(spa@ct.de)

**Mit dem Webinterface können verschiedene Farben ausgewählt werden, in denen das Armband leuchtet.**

Beispiel möglich, das Armband per Musik anzusteuern. Dafür braucht der Arduino ein Schallerkennungsmodul, das genau wie das Infrarotmodul aus

einer kleinen Platine mit einem aufgelösten Bauteil besteht. In diesem Fall ist aber statt der LED ein hochempfindliches Mikrofon als Schallsensor integriert. Wird ein Lied abgespielt, kann bei entsprechender Programmierung dann



**Wenn das eigene Handy mit einem Infrarot-Blaster ausgestattet ist, kann man das LED-Armband auch mit einer App über das Smartphone ansteuern.**

## Datenübertragung per Infrarot

Infrarotlicht gehört zur elektromagnetischen Strahlung und hat weniger Energie als sichtbares rotes Licht. Infrarot-LEDs sind beispielsweise in Fernbedienungen für Fernseher und andere elektronische Geräte verbaut und funktionieren über kurze Distanzen. Wer also damit den Fernseher einschalten will, darf nicht allzu weit von ihm entfernt stehen. Gleches gilt für die Eras-Tour-Armbänder – vor allem in dieser Bastelvariante für zu Hause.

In den Stadien steuern große Scheinwerfer mit Dutzenden Infrarot-LEDs die Lichtershow im Publikum. Wir nutzen hingegen nur eine einzelne LED, deren Reichweite geringer ist. In unserem Setup können wir das Armband aus fünf Metern Entfernung noch zuverlässig steuern. Ideal ist dabei direkter Sichtkontakt zwischen Sender und Empfänger.

Ob mit den großen Scheinwerfern auf den Konzerten oder im kleinen Setup zu Hause: Die grundlegende Funktionsweise der Armbänder ist gleich und wird schon seit vielen Jahren von unterschiedlichen Anbietern eingesetzt. Die Infrarot-LEDs verschicken die Befehle für die ent-

sprechenden Farben; der Infratotsensor im Armband empfängt sie. Grundsätzlich gibt es verschiedene Möglichkeiten, das Signal zu modulieren. Bei den Eras-Tour-Armbändern funktioniert das mit einer Art Morsecode, für den die Infrarot-LED im Millisekundentakt strahlt. Jedes Farbsignal entspricht einer bestimmten Abfolge.

In den Stadien werden dafür in zwei Rotationsachsen bewegliche Scheinwerfer genutzt. Diese sogenannten Moving Heads können das Infratotsignal zielgerichtet in ganz bestimmte Bereiche im Publikum senden und so nicht nur alle Armbänder in der gleichen Farbe aufleuchten lassen, sondern räumliche Muster erzeugen.

Die ganze Bandbreite an Möglichkeiten hat sich zuletzt bei den Olympischen Spielen in Paris gezeigt. Dort haben neuartige Moving Heads mehrere Infratotsignale simultan übertragen und das Publikum dadurch wie ein Bildschirm mit „menschlichen Pixeln“ genutzt: Hunderte Armbänder haben riesige Sportsilhouetten erzeugt, zum Beispiel einen Pferdewagen.

## Literatur

- [1] GitHub-Repository des PixMob IR Reverse Engineering Project, [github.com/danielweidman/pixmob-ir-reverse-engineering](https://github.com/danielweidman/pixmob-ir-reverse-engineering)
- [2] Arduino-Einführung, [docs.arduino.cc/learn/start-guide/the-arduino-software-ide/](https://docs.arduino.cc/learn/start-guide/the-arduino-software-ide/)
- [3] Web-Interface für LED-Armbänder, [ivanr3d.com/tools/led-wristband/](https://ivanr3d.com/tools/led-wristband/)
- [4] Web-Interface für das Eras-Armband, [wissenhinterdenkulissen.de/armband/](https://wissenhinterdenkulissen.de/armband/)

**Links zu den erwähnten Webseiten:**  
[ct.de/yvqh](https://ct.de/yvqh)

Für KMUs gemacht!

# Online Marketing Day

by heise

21.  
Nov.  
24

Alles, was Sie über Suchmaschinenwerbung, SEO, Social Media, den richtigen Umgang mit KI, erfolgreiches Recruiting und Markenaufbau wissen müssen – mit uns sind Sie immer **eine Antwort weiter!**

- inspirierende Vorträge
- praxisnahe Impulse
- exklusive Einblicke

Kostenfrei. Gleich anmelden!



 **heise regioconcept**

© Copyright by Heise Medien.

Eine  
Antwort  
weiter



# Frischzellenkur für Nachtspeicher

**Mit ESP & Co. Speicherheizung erforscht und steuerbar gemacht**

Wie reagiert der c't-Redakteur, wenn er sich in einer Unterkunft wiederfindet, die keine fossile Heizung hat, sondern Nachtspeicheröfen, ein Anachronismus aus der Zeit, als AKWs noch eine strahlende Zukunft hatten? Mein Weg: in die Technik hineinrinden und die Heizdinos mit ESPs versmarten.

Von Peter Siering

Die in 1960er-Jahren erfundenen Nachtspeicheröfen verwandeln Strom in Wärme. Sie stammen aus einer Zeit, als Strom aus Kohle- und später Atomkraftwerken besonders in den Nachtstunden im Überfluss vorhanden war. Spezielle Niedertarifangebote (NT), die immer noch als Nachtstrom bekannt sind, erhöhten die Attraktivität, weil sie in der Anfangszeit subventioniert waren.

Zum Betrieb solcher Heizungen, die oft alte Gas- und Ölöfen ersetzten, genügt ein ausreichend dimensionierter Stromanschluss. Ein zweiter Zähler erfasst den Verbrauch der Heizgeräte. Günstigeren Strom lieferte der jeweilige Energieversorger nur zu speziellen Zeiten, oft eben in der Nacht; einige Anbieter aber auch tagsüber, um die Gesamtlast des Strom-

netzes zu regulieren. Noch heute führen einige Energieversorger Niedertarifangebote, die aber kaum günstiger als Normaltarife sind.

In der NT-Phase erhalten die Öfen üblicherweise Strom und erhitzen die hinter einer wärmeisolierenden Schicht liegenden Steine. Die Kunst beim Betrieb besteht darin, die Ladung so zu dosieren, dass die Wärme bis

zur nächsten Ladephase genügt. Gegen Aufpreis gab es auch Öfen, die tagsüber zum Normaltarif nachladen konnten. Wer das nicht hat, friert, wenn er sich beim Laden verschätzt.

**Frisch aus  
ct Nerdistan**

## Phasenfragen

Die Speicherfähigkeit gibt Stiebel Eltron als einer der Hersteller solcher Öfen als

sogenanntes Wärmerückhaltevermögen an: Nach Ablauf von 16 Stunden sollen noch 55 Prozent der gespeicherten Wärme im Gerät zur Verfügung stehen. Das heißt: Ein solcher Speicher gibt in dieser Zeit knapp die Hälfte der Wärme an seine Umgebung ab. Das lässt sich beschleunigen, indem in den Ofen integrierte Lüfter die Wärme aktiv herausziehen.

Weil die veralteten Geräte Energie speichern, passen sie mit ein bisschen Augen zudrücken auch ins Zeitalter der erneuerbaren Energien. Das Argument half jedenfalls, ein im Jahr 2009 beschlossenes Verbot von Nachspeicherheizungen 2013 aufzuheben. Es hätte deren Betrieb ab 2020 untersagt. Ein erneutes Verbot scheint nicht in Sicht.

So weit etwas Hintergrund zur Technik – all das wusste ich nicht, als ich das erste Mal in der neuen Wohnung vor den über 250 Kilogramm schweren weißen Kisten von Stiebel Eltron stand. Die erste Assoziation war: Die Öfen galten doch als asbesthaltig. Wie sich schnell herausstellte, betrifft das vor allem sehr alte Geräte. Ich hatte es zum Glück mit erst nach 2000 installierten, asbestfreien Exemplaren zu tun; eine Anfrage an den Hersteller sicherte die Erkenntnis ab.

An den Geräten gibt es einen Knopf mit drei Stufen, in jedem Raum an der Wand eine Art von Temperaturregler, in der Unterverteilung ein ominöses Kästchen von Stiebel Eltron mit einem Drehknopf und pro Ofen drei 16-Ampere-Lei-

tungsschutzschalter, im Volksmund auch Sicherungen. Mehr Steuerungsmöglichkeiten gab es auf den ersten Blick nicht. Niemand konnte erklären, wie all das wohl zusammenwirkt und bedient wird.

### Externe Hilfe

Zuerst suchte ich einen Elektriker, der die gesamte Elektroinstallation prüft und sich auch nicht scheut, die alte Heiztechnik zu warten. So konnte ich einen ersten Blick in die Elektroinstallation der Wohnung und die Innereien der Öfen werfen. Parallel dazu entdeckten wir auch eine Zentralsteuerung im Hausanschlussraum, wo die doppelt ausgeführten Zähler für zwölf Wohnungen hängen und in dem die Nase immer eine elektrisierende Atmosphäre wittert.

Die Aufladesteuerung misst die Außen temperatur und enthält eine Uhr. Mit diesen Informationen generiert sie ein Signal, das den Öfen mitteilt, wie stark sie sich in der kommenden NT-Phase aufladen sollen. Das kleine Kästchen mit dem Drehknopf in der Unterverteilung der Wohnung hilft, den Ladegrad der eigenen Öfen in Grenzen anzupassen.

Neben der Zentralsteuerung hängt ein weiteres spannendes Gerät: der Rundsteuerempfänger. Der Energieversorger sendet hieran ein Signal, das die Zentralsteuerung dazu bewegt, den Strom für die Aufladung freizuschalten – in Form von Leistungsrelais (Schützen), die drei Phasen für jede einzelne Wohnung anschalten. Zur elektrisierenden Atmosphäre des

## Projektinfo

 Drei Bauteile genügen, um ein 230-Volt-Steuersignal zu erzeugen, das in Altinstallationen mit Nachspeicheröfen üblicherweise von einer Zentralsteuerung kommt und grob die Aufladung regelt.

 Elektroinstallation  
Programmiererfahrung  
Lötkenntnisse  
Elektrofachkraft

 1 Stunde Installation

 20-30 Euro  
 ESP8266  
Solid-State-Relais-Board  
(USB-)Netzteil  
Sicherungshalter

 Lötkolben  
E-Installationswerkzeug

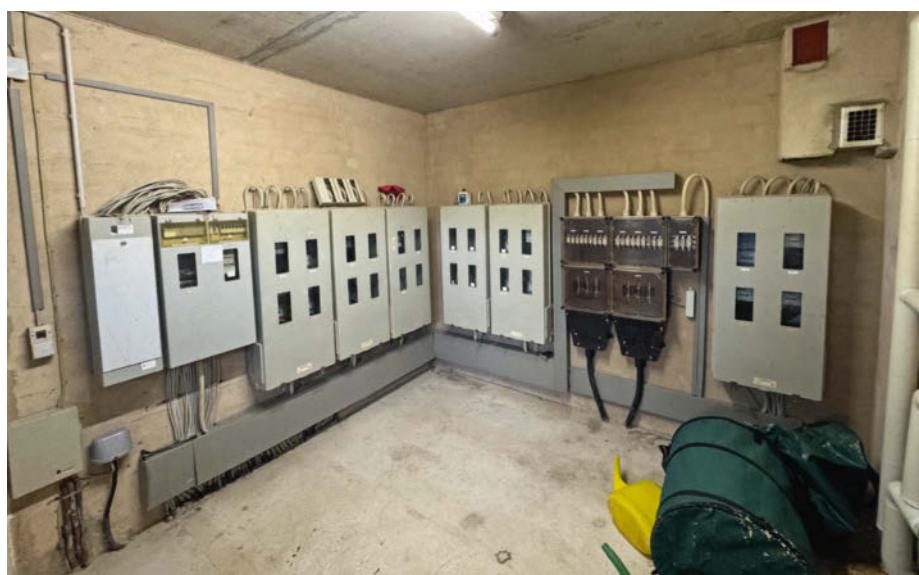
Raums gesellt sich dann noch ein sonores Brummen.

Die Uhrzeit hatte die Steuerung offenbar beim letzten Stromausfall aus den Augen verloren. Sie lief im „Notbetrieb“ – vermutlich seit Jahren, niemand hatte sich bis dato darüber gewundert. Die Fühler der Steuerung hatte die Hausgemeinschaft irgendwann mit der Wetterseite des Hauses hinter einer Isolierung versteckt, sodass deren Messwerte immer viele Grade neben den realen Verhältnissen liegen.

Während ich das Heizverhalten der gewarteten Öfen manuell regelte und ihre Außen temperatur mit DS1820-Sensoren an ESPs beobachtete, begann die Zentralsteuerung aufgrund eines Fehlers in einer anderen Wohnung zu spinnen: Auch tagsüber zogen die Schützen an und die Öfen luden nicht nur im typischen nächtlichen Zeitfenster, sondern auch am Tag auf. Der Spuk war nach einigen Wochen vorbei, machte mir aber klar, dass ich mich nicht von einer zentralen Steuerung abhängig machen wollte.

### Wenig Spielraum

Die Regelungsmöglichkeiten, mit denen ich mich an die nötige Aufladung der Öfen bei verschiedenen Temperaturen herantastete, fielen rudimentär aus: Per Le-



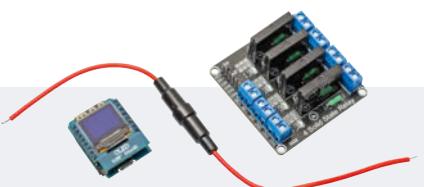
**Das Heizen mit Nachspeicheröfen mag an vielen Stellen Installationsaufwand sparen, für den Hausanschlussraum gilt das jedoch nicht. Dort sind für jede Wohnung gleich zwei Zähler installiert, die getrennt im Hoch- und Niedertarif bezogene Strommengen erfassen.**



## Raumtemperatur und Luftfeuchte erfassen

Zum Erfassen der Raumtemperaturen setze ich auf einen Klassiker der Heimautomation, einen JeeLink-LaCrosse-Stick, der die Signale von im Haushalt verteilten 868 MHz Funksendern empfängt und am USB-Port als Text ausgibt. Passende, auch für den Außeneinsatz geeignete Sender gibt es als Technoline TX29 DTH-IT für unter 20 Euro. Sie laufen mit zwei AA-Batterien rund ein Jahr.

JeeLink-Sticks kosten fertig rund 50 Euro. Mit einem Arduino Nano und einem Funkmodul RFM12B-868-S ist eine Open-Source-Alternative für unter zehn Euro aufgebaut. Der Nachteil der beiden Varianten: Zusätzlich braucht es Software, die aus den Textnachrichten verarbeitbare Daten etwa in einen MQTT-Server einspeist. Das lässt sich zum Beispiel mit Node-Red auf einem Raspi organisieren.



## Öfen steuern

Um die Steuersignale für die Öfen zu erzeugen, genügt ein ESP8266, dessen Digitalausgänge ein Solid-State-Relais-Board ansteuern. Halbleiterrelais arbeiten komplett geräuschlos und halten viel mehr Schaltzyklen aus als ihre elektromechanischen Pendants.

Das Steuersignal dient zwar zur Erwärmung eines Widerstands, aber dafür genügen wenige Watt. Eine Leistung, die das sicherheitshalber mit einer flinken 1-Ampere-Feinsicherung abgesicherte Board locker wuppt. Relaisboard, ESP und das vierzeilige OLED-Display sind für rund 20 Euro zu haben.

tungsschutzschalter lassen sich Öfen ganz stilllegen. Der Drehknopf in der Unterverteilung blieb in meinem Fall wirkungslos. Per Knopf am Ofen lässt sich dessen Ladegrad in drei Stufen anpassen, heißt: die Dauer der Ladung beeinflussen.

Grundsätzlich stellt der Energielieferant in der Zeit von 22 bis 6 Uhr Strom zum Aufladen der Öfen bereit. Auf den kleineren Stufen heizen sich die Geräte nicht bis zum Schluss auf, sondern stoppen schon deutlich vor 6 Uhr. Das hängt allerdings auch davon ab, wie stark der Ofen in den vorhergehenden Nächten bereits Wärme getankt hat.

Das Hochheizen bei knapp zweistelligen Temperaturen dauert die ganze Nacht, zieht vier Kilowatt und kostet pro Ofen dann schnell mehr als zehn Euro (mit 40 Cent pro kWh gerechnet). Die gemessene Temperatur am Ofengehäuse liegt dann je nach gewähltem Ladegrad bei 40 bis 60 Grad (innen laut Hersteller mehrere Hundert Grad) und fällt im Laufe des Tages um bis zu 20 Grad. Bis die Temperaturen der Wohnung die Komfortzone erreichen, dauert es rund zwei Tage.

Die vorgesehenen Steuermöglichkeiten genügen, wenn jemand vor Ort ist, ein Gefühl für das Verhalten entwickelt hat und vorausschauend herangeht. Aus der Ferne scheint eine Steuerung aussichtslos. Anfangs ging ich davon aus, dass es möglich sein sollte, den Nachstrom aus der Ferne zumindest ein- und ausschalten zu können, aber nach Experimenten mit verschiedenen Ladegraden war klar, dass eine feinere Steuerung nötig sein würde.

Obendrein schien der Installationsaufwand für simples Ein- und Ausschalten erheblich: Die Wohnungsunterverteilung war voll und übliche WLAN-Schalter wären den fließenden Strommengen ohnehin nicht gewachsen. Im Hausanschlussraum wäre es möglich, das Heizungsschütz für die Stromversorgung der Öfen zu beeinflussen, etwa mit einem WLAN-Schaltaktor von Shelly, aber dort hin hätte erst mal Internet gebracht werden müssen – mittels Power-over-Ethernet möglich, aber nicht wirklich schön.

Eine Alternative bietet die Vorrangschaltung für Warmwasser: Ein Lastabwurfrelais in der Wohnungsunterverteilung schaltet das Heizungsschütz im Keller ab, wenn während des Heizens der Durchlauferhitzer anspringt – das verhindert, dass die Last aller Verbraucher zusammen zu hoch wird. Ein WLAN-Schaltaktor könnte den Lastabwurf steuern und so das Laden der Öfen unterdrücken. Von einer

gezielten Regelung ist das aber weit entfernt.

## PWM mit 230 Volt

Ich recherchierte weiter und fand heraus, wie die eigentlich vorhandene Zentralsteuerung funktioniert: Sie bestromt mit 230 Volt einen Steuerwiderstand, dessen Erwärmung ein Thermorelais im Ofen betätigt, um die großen Ströme für die Heizstäbe zu schalten. Je stärker er beheizt wird, desto kürzer ist die Aufladezeit. Das funktionierte schon, bevor Digitaltechnik in die Steuerungen Einzug hielt.

Die Widerstandsheizung schluckt selbst Strom. Auch deswegen findet sich in der Unterverteilung der zusätzliche Regler. Es handelt sich um ein sogenanntes Gruppensteuergerät. In einer funktionierenden Installation zeigt es üblicherweise mit einem Blinken das Steuersignal an. Außerdem reduziert es die Last, indem es ein eigenes Steuersignal für die Öfen an der jeweiligen Unterverteilung generiert.

Als Steuersignal verbreitet ist ED80. Wofür diese konkrete Abkürzung steht, konnte ich nicht herausfinden. Im Fall von ED80 muss in zehn Sekunden mindestens acht Sekunden Strom anliegen (also 80 Prozent der Zeit), um eine Aufladung zu verhindern. Je kürzer Strom anliegt, desto höher fällt die Aufladung aus.

Es gibt Varianten, die eine andere Aufteilung vorsehen, etwa ED40, die sind aber weniger verbreitet. Moderne Steuergeräte können statt einer Wechselspannung von 230 Volt eine Gleichspannung zwischen 0,91 und 1,43 Volt verwenden. Beide Systeme sind so konzipiert, dass ein Ofen beim Ausfall des Steuersignals immer voll lädt, positives Störverhalten genannt.

Mit diesem Wissen ist der Weg zu einer eigenen Steuerung nicht mehr weit. In der Unterverteilung fand ich je eine Leitung für das Steuersignal zu den Öfen, die am Gruppensteuergerät hing. Ein Solid-State-Relais-Board und ein ESP 8266 schicken dieses Steuergerät in Rente. Sie erzeugen das ED80-Signal passend zu per MQTT übermittelten Aufladevorgaben. So lässt sich sogar jeder Ofen individuell steuern.

## Analoges Regeln

Ein paar Erkenntnisse am Rande: Die Knöpfe an den Öfen selbst deckeln die maximale Aufladung, egal, was das Steuersignal sagt. So kann man sicherstellen, dass bei Experimenten nicht übermäßig aufgeladen wird. Steht die Vorgabe am Ofen auf der kleinsten Stufe, muss das ED80-Signal



**Die beiden Steuergeräte links sollen mittels Temperatursensor Vorhersagen über den Heizbedarf treffen und per 230-Volt-Steuersignal in die Unterverteilung der Wohnungen übermitteln. Rechts hinter der Glasscheibe sitzt der Rundsteuerempfänger, über den der Versorger das Nachtstromsignal übermittelt.**

mindestens 30 Prozent Aufladung übermitteln, sonst bleibt der Ofen kalt.

Ich bin dazu übergegangen, in der Heizperiode maximale Aufladung einzustellen. Um zu verhindern, dass die Öfen beim Ausfall der Steuerung stets voll aufladen, habe ich das Störverhalten angepasst: statt positiv auf negativ, also keine Aufladung beim Ausfall des Signals. Viele Öfen bieten das an.

Je weiter ich mich in die Technik reingefuchst habe, desto mehr Merkwürdigkeiten fanden sich in der lokalen Installation: Zuerst stellte sich heraus, dass die Zentralsteuerung nicht lief. Dann fand ich Merkwürdigkeiten bei einzelnen Öfen: Bei einem hatte jemand das Steuersignal abgeklemmt und der Steuerung keinen Dauerstrom gegönnt; die lief erst an, wenn der Nachtstrom anlag.

In einem Raum stehen zwei Öfen, die kaskadiert angeschlossen sind. Bei dem zweiten Gerät in der Kette kommen zwar die Steuersignale an, aber es ignoriert sie und lädt immer voll auf. Ein Ofen hat bisher immer zum Heizen des Zimmers genügt, sodass der Zweitofen abgeschaltet bleibt.

Für eine weitere Überraschung ist der Lastabwurf gut: Das auf der Hutschiene in der Wohnungsunterverteilung montierte Gerät schaltet bei Warmwasserbezug per Durchlauferhitzer das Schütz für die Öfen im Keller vorübergehend ab. Eine Falle droht beim Tausch eines alten Durchlauf-erhitzers durch einen elektronischen. Denn dann belastet der alte Lastabwurf das Schütz im Hausanschlussraum stark,

weil es nun viel häufiger betätigt wird. Es klappert hektisch und seine Kontakte verschleissen schneller. Man muss also auch die Anlage anpassen.

Ein Wort noch zu den Raumtemperaturreglern: Die lassen sich per Schalter aktivieren und erlauben das Einstellen einer Wunschttemperatur. Sie steuern dabei nicht etwa, wie stark der angeschlossene Ofen aufgeladen wird, sondern nur, bis zu welcher Temperatur der Lüfter im Ofen Strom erhält.

Dieser Lüfter zieht kalte Luft durch die Steine im Ofen und beschleunigt so die Wärmeabgabe. Ich nutze diese Turboheizung allenfalls mal kurz nach dem Lüften. Normalerweise genügt die Strahlungswärme der Öfen und die so angeregte Konvektion. Deswegen habe ich mir über deren Fernsteuerbarkeit keine Gedanken gemacht.

### Steuern, nicht regeln

Am Ende meiner Forschung steht keine fertige Regelung, die Uhrzeiten, An- und Abwesenheit, Außentemperaturen und die Wettervorhersage berücksichtigt – alles Daten, die digital zugänglich wären. Aber ich habe immerhin eine aus der Ferne zugängliche Steuerung, kann also die Öfen vor Anreise vorheizen und bei einem Kälteinbruch die Grundwärme erhöhen. Das ist mehr, als mir anfangs habe träumen lassen.

Vergleichbare kommerzielle Lösungen kosten mehrere Hundert Euro und sind nur mit einem vollständigen Umbau der Unterverteilung realisierbar. Eine den Energiebedarf deutlich reduzierende Split-Klimaanlage mit Wärmepumpe braucht einen Aufenteil, den die Hausgemeinschaft tolerieren müsste, und würde eine fünfstellige Investition erfordern – für eine nicht dauerhaft genutzte Wohnung ist das eher keine Option.

Mit der manuellen Steuerung der Nachspeicher erreiche ich Grundwärme. Genügt die nicht, helfen per WLAN-Steckdose gesteuerte Billig-Infrarotpanels ohne eigene Regelung und Steuerung nach. Ob man eine weitergehende Regelung im ESP oder vielleicht per Home-Automation realisiert, dürfte Geschmackssache sein.

Spannend wäre, mehr über den ST62T65-Mikrocontroller herauszufinden, der in vielen Nachspeicherheizungen nicht nur von Stiebel Eltron zu finden ist. Mit etwas Hintergrund über den Aufbau und womöglich die dort laufende Software ließe sich mit Sicherheit mehr erreichen. Technische Dokumentation ist lei-



### Messdaten sammeln

Ein ohnehin als DNS-Filter laufender Raspi 3B sammelt dauerhaft Messdaten und überträgt Steuerinformationen. Auf ihm laufen mehrere Docker-Container mit MQTT-Server und Node-Red. Er vermittelt auch mittels Zigbee2mqtt und Homebridge zwischen MQTT und dem goldenen Käfig von Apples HomeKit. Als Zigbee-Gateway dient dabei ein USB-Dongle von Sonoff für 35 Euro.



### Oberflächentemperatur messen

Um mehr über das Aufladeverhalten der Nachspeicheröfen zu lernen, lagen einfache DS1820-Sensoren nahe. Die gibt es vorkonfektioniert bei einschlägigen Versendern. Mit einem Kabelbinder habe ich die auf die Oberseite der Öfen geschnallt und mit einem ESP8266-DevKit verbunden. Auf dem ESP läuft Tasmota, das von sich aus solche Sensoren mit und ohne Vorwiderstand auslesen und an einen MQTT-Server versorgen kann. Alte USB-Mobiltelefon-Netzteile oder naheliegende USB-Ports versorgen die ESPs mit Strom.

der nicht zugänglich. Vielleicht mag die ja jemand durchstechen? (ps@ct.de) **ct**

### Literatur

[1] Erste Inspiration, ein Posting im Homematic-Forum: [heise.de/s/Xe3Nm](http://heise.de/s/Xe3Nm)

**Erwähnte Projekte:** [ct.de/ypx7](http://ct.de/ypx7)

# secIT

by heise

HANNOVER 2025

18. - 20. MÄRZ 2025, HANNOVER

# Die Kongress- messe für Security-Profis

A1 Digital



Avast

baramundi

Bare.ID



Bitdefender  
Global Leader  
In Cybersecurity

BND



cybereason®

»CYBERSENSE

CYQUEO  
Cyber-Security Solutions

DATACORE

Delinea

Devolutions



ITISO

KnowBe4

KEYFACTOR



LANCOM  
SYSTEMS

LocateRisk

noSpam proxy®

oneconsult  
together against cyberattacks

opentext™  
Security Solutions

IPCO

semperis

SERASEC  
SERVICE AND SECURITY

SerNet

SONICWALL®

Offizieller  
Eventpartner

itsa EXPO CONGRESS  
HOME OF IT SECURITY

Kooperations-  
partner

© Copyright by Heise Medien.

CISO  
ALLIANCE

NIEDERSACHSEN  
DIGITAL

UVN  
UNTERNEHMERVERBAND  
NIEDERSACHSEN E.V.

# meet. learn. protect.



**Jetzt  
anmelden!**



chrome enterprise

cidaas

cirosec

COMPASS<sup>®</sup>

ConSecur  
[security and consulting]

**CONSIST**  
business information technology



enthus

**eset**  
Digital Security  
Progress. Protected.

EWE

exeon  
Smart Cyber Security

FB PRO GMBH  
System Hardening & Data Protection

**FORTINET**

**m** mITSM  
Wir verändern durch Wissen.

**Net at Work**  
Building IT-Excellence.

**NETCOW**

netskope

**nextron**  
systems

ninjaOne

**norton**

**SOPHOS** sosafe **SpaceNet**

**STORMSHIELD**

**TEAL**  
ALWAYS CHALLENGING IT

**tenfold**

**tresorit**

**veeam**

**W / T H**  
SECURE

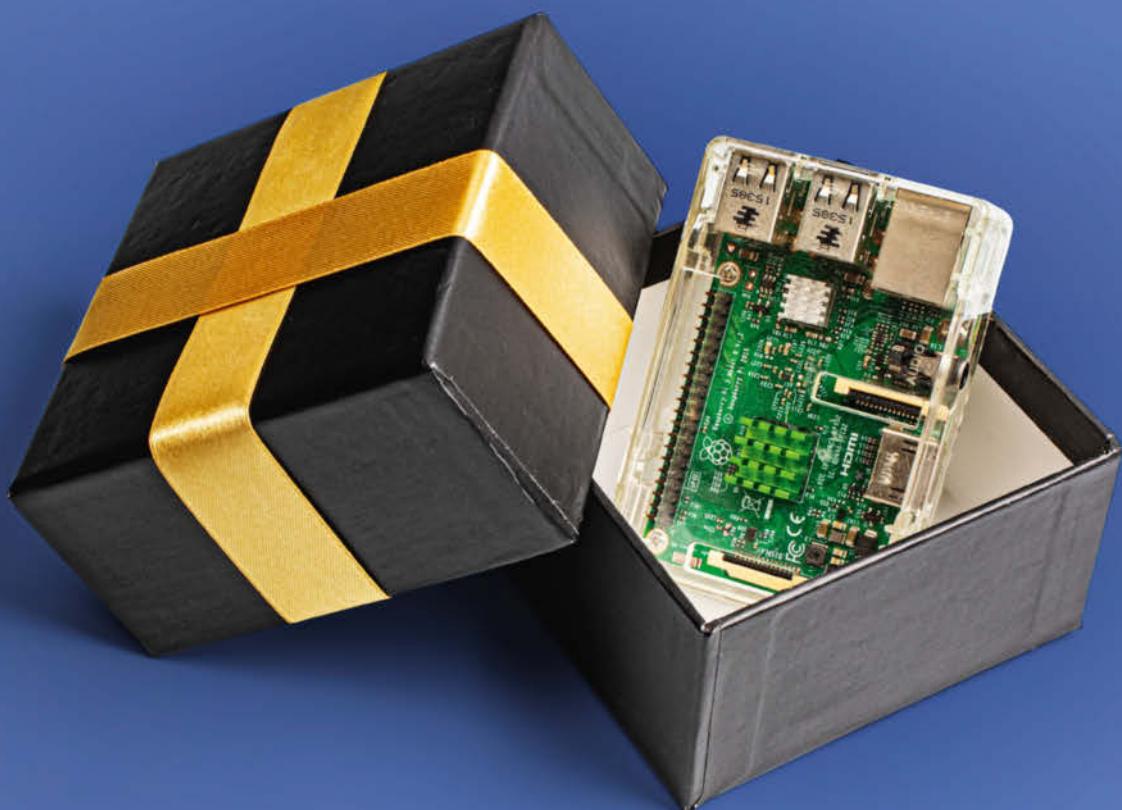
**xplicitTrust**

# secit-heise.de

© Copyright by Heise Medien

# Bastel-Mini

Kaufberatung und erste Schritte  
mit dem Raspberry Pi



<b>Modellauswahl und erste Schritte mit dem Raspi .....</b>	<b>Seite 108</b>
<b>Bootmedien für den Raspi bauen.....</b>	<b>Seite 116</b>
<b>Den GPIO-Anschluss des Raspi nutzen.....</b>	<b>Seite 118</b>
<b>Den Raspi aus der Ferne warten.....</b>	<b>Seite 126</b>
<b>MicroPython-Entwicklung auf dem Raspi Pico .....</b>	<b>Seite 130</b>
<b>Der Mikrocontroller Raspberry Pi RP2350.....</b>	<b>Seite 132</b>
<b>Raspi-Projektideen für Schüler.....</b>	<b>Seite 138</b>

# Eine LED blinken lassen, wenn eine neue E-Mail angekommen ist? Kein Problem für den Raspberry Pi und nur eines von Hunderten möglichen Projekten für den weltweit beliebten, kompakten und günstigen Bastelcomputer. Wir sagen Ihnen, welchen Raspi Sie wofür einsetzen können und wie Sie den Mini-Computer für Ihr Projekt vorbereiten.

Von Mirko Dölle

**E**in Mini-Computer für Basteleien, mit unter 40 Euro viel billiger als ein PC und weitaus besser geeignet für Steuerungsaufgaben. Bei dem es keine Katastrophe ist, wenn man mal einen fatalen Fehler macht und den Rechner grillt. Das ist der Raspberry Pi. Er ist nicht der erste Einplatinencomputer (Single Board Computer, SBC), der es unter die Lötkolben von Bastlern schaffte. Die lange Produktpflege, gute Verfügbarkeit, stetige Weiterentwicklung und vor allem der Preis haben ihm Millionen treue Fans beschert.

Von Anfang an achtete die federführende Raspberry Pi Foundation darauf, dass der Raspi möglichst günstig bleibt, denn man wollte ihn an Schulen für den Informatikunterricht etablieren. Gleichzeitig fand das seinerzeit konkurrenzlos günstige Bastel-Board den Weg in unzählige private Hobbykeller. So kostete nicht nur der Ur-Raspi um 35 Euro, auch spätere Modelle mit mehr Leistung und Funktionen gab es zum gleichen Preis, von gelegentlichen Preisschwankungen etwa bei schlechter Verfügbarkeit einmal abgesehen. Der Raspi 5 ist der erste, der mit über 50 Euro deutlich teurer ist als seine Vorgänger.

Dennoch ist der Raspberry Pi bis heute eine äußerst günstige Entwicklungsplattform, denn andere Einplatinencomputer wie der Radxa Rock 4 B+ oder Rock 5 kosten ab 75 Euro aufwärts, die Asus TinkerBoards sogar über 100 Euro. Die Boards sind damit nicht per se überteuert, sie bieten durchaus einiges für das Geld. Doch mit dem Preis steigen auch die Hemmungen, das Board bei Basteleien zu beschädigen.

Zur Verbreitung des Raspberry Pi hat auch beigetragen, dass die Raspi-Foundation die Software selbst für das älteste Modell weiterhin pflegt. Das ist eine Schwachstelle vieler Raspi-Alternativen: Kaum ist der Hype um ein neu erschienenes Board vorbei, ziehen die Entwickler zur nächsten Plattform. Daran krankten zum Beispiel auch die CubieBoards und CubieTrucks. Für den Ur-Raspi gibt es auch nach über zehn Jahren noch aktuelle Software.

Ganz klar abgrenzen muss man den Raspberry Pi und andere Single Board Computer gegenüber den Mikrocontrollern, zu denen neben dem ESP32 auch der Arduino und der Raspberry Pi Pico gehören. Das wichtigste Unterscheidungsmerkmal ist, dass Mikrocontroller mit einer Firmware bespielt werden und nicht über ein Betriebssystem verfügen. Näheres dazu finden Sie ab Seite 78 in diesem Heft.

## Das Raspi-ABC

Am Anfang der Entwicklung stand das Modell B des Raspberry Pi mit nur einem 32-Bit-CPU-Kern und 512 MByte RAM. Es gab weder WLAN noch Netzwerkkomponente, nur zwei USB-2.0-Ports. Auch der GPIO-Anschluss war mit 26 Pins weitaus kürzer als der nachfolgender Generationen. Das erst später produzierte Modell A war nochmal kleiner, hatte aber nur einen USB-Anschluss und spielte in der Praxis kaum eine Rolle.

Die Neuanschaffung eines solchen Ur-Raspis lohnt sich in der Regel nicht, denn er ist – sofern überhaupt lieferbar – meist 10 bis 20 Prozent teurer als ein Raspberry Pi 3B mit vier 64-Bit-CPU-Kernen, 1 GByte RAM, 100 MBit/s Ethernet, 2,4-GHz-WLAN und Bluetooth 4.0. Es gibt allerdings noch ein paar Projekte, deren Erweiterungsplatten auf den Ur-

Raspi zugeschnitten sind. Sollten Sie zufällig noch einen in der Bastelkiste finden, müssen Sie ihn auch nicht wegwerfen: Für einfache Schaltaufgaben und auch für die Mikrocontroller-Entwicklung, etwa um den ESP32 mit neuer Firmware zu versetzen, lässt sich ein alter Raspi immer noch einsetzen.

Mit knapp 40 Euro kostet der Raspberry Pi 4 so viel wie der Pi 3, hat aber vier mit 1,5 GHz getaktete CPU-Cores, 1 GByte RAM, Gigabit-Ethernet, 2,4- und 5-GHz-WLAN, Bluetooth LE 5, einen zweiten Monitoranschluss (Micro-HDMI) sowie zwei USB-3.0- und zwei USB-2.0-Anschlüsse. Dieses Modell empfiehlt sich aktuell für die meisten Projekte. Es gibt den Raspi 4 in mehreren Varianten, mit bis zu 8 GByte RAM und für rund 70 Euro sogar als „Brotkasten“-ähnliches Modell 400, vom Konzept her ähnlich dem legendären Commodore C64 oder C128, in einem Tastaturgehäuse mit Anschlüssen auf der Rückseite.

Das neueste Modell 5 der Familie ist nochmal deutlich leistungsfähiger als der Raspi 4: vier CPU-Kerne mit 2,4 GHz Taktfrequenz, 2 bis 8 GByte schnelleres RAM, eine PCIe-2.0-Lane und Monitorauflösungen bis zu 4K bei 60 FPS. Allerdings sprengt der Raspi 5 den bisherigen Preisrahmen: Los geht es bei über 50 Euro, auch die Modelle mit mehr Speicher kosten etwa 10 bis 15 Euro mehr als ein Raspberry Pi 4.

Besonders interessant ist die PCIe-Lane des Raspi 5. Hierüber können Sie besonders schnelle NVMe-SSDs anschließen. Das Ergebnis ist ein sehr leistungsfähiger Minicomputer, der knapp unterhalb von Kompaktsystemen wie dem Intel NUC rangiert. Allerdings auch preislich: Denn um den Raspi 5 ausreizen zu können,

## c't kompakt

- Klein, günstig und nach über zehn Jahren noch immer mit Software versorgt: Der Raspi hat es auf viele Basteltische geschafft.
- Die Raspi-Zero-Modelle für unter 20 Euro sind ideal für besonders kompakte Projekte.
- Eine robuste Paketverwaltung macht es leicht, weitere Software zu installieren.

benötigen Sie neben dem Raspi, der mit 8 GByte RAM knapp 90 Euro kostet, noch ein geeignetes USB-C-Netzteil und ein Kühler- und Lüfter-Set. Ungekühlter läuft der Raspi 5 heiß und bringt nicht die volle Leistung. Außerdem auf dem Einkaufszettel: ein Gehäuse und der Adapter für die NVMe-SSD. Das summiert sich auf weitere 40 bis 50 Euro – also gut 130 Euro für einen betriebsbereiten Raspi 5 mit 8 GByte RAM, zuzüglich der NVMe-SSD.

Der Vorteil von Kompaktrechnern wie dem Intel NUC ist, dass es sich um eine leistungsfähigere x86-CPU handelt, auf der Sie praktisch jede Linux-Distribution installieren können, und dass der Speicher über 8 GByte hinaus erweiterbar ist. Allerdings gibt es bei den NUCs keinen GPIO-Anschluss, über den Sie wie beim Raspi Steueraufgaben übernehmen oder zusätzliche Hardware wie Displays oder Servocontroller anschließen können. Bevor Sie sich in das Abenteuer eines Raspi 5 mit besonders viel Speicher stürzen, sollten Sie durchrechnen, was ein ähnlich ausgestatteter Kompakt-PC kosten würde, und sich dann für das für Ihr Projekt geeignete System entscheiden.

### Miniatur-Raspis

Für besonders kompakte Projekte eignet sich die Raspi-Zero-Familie. Ein Beispiel ist der Selbstbau eines portablen Positionsempfängers für Sportboote, den wir ab Seite 160 in dieser Ausgabe vorstellen. Die

Raspi Zero sind kaum länger als der 40-polige GPIO-Anschluss, nur 3 Zentimeter breit und kosten mit 10 bis 20 Euro gerade mal halb so viel wie die großen Brüder. Trotzdem sind sie vollwertige Raspis mit einem USB-Anschluss für Tastatur und Maus sowie einem Mini-HDMI-Anschluss für einen Monitor.

Es gibt mehrere Modelle des Raspi Zero: Der ursprüngliche Zero ist eine Mini-Version des Ur-Raspis, besitzt den gleichen, allerdings mit 1 GHz deutlich schneller getakteten SoC mit nur einem 32-Bit-CPU-Kern. Auch hat er mit 512 MByte genauso viel RAM. Allerdings gibt es beim Raspi Zero nur einen MicroUSB-Port – um hier herkömmliche USB-Geräte mit USB-A-Stecker anzuschließen, benötigen Sie ein sogenanntes USB-OTG-Kabel (On The Go) mit einem USB-Micro-Stecker auf der einen und einer USB-A-Buchse auf der anderen Seite. Hier können Sie etwa für die Installation auch einen USB-Hub für mehrere USB-Geräte wie Tastatur und Maus sowie einen USB-Ethernet-Adapter anschließen.

Beim Zero W ist zusätzlich das gleiche 2,4-GHz-WLAN- und Bluetooth-Modul aufgelöst, das auch im Raspberry Pi 3 steckt. Selbst wenn Sie später kein WLAN für Ihr Projekt benötigen, erleichtert das WLAN-Modul die Einrichtung ungemein: Wenn Sie das Betriebssystem gleich mit den Zugangsdaten Ihres heimischen WLANs konfigurieren und zudem den

Fernzugang aktivieren, benötigen Sie für die Einrichtung weder OTG-Kabel noch Tastatur, Maus oder Monitor: Sie können alle Arbeiten bequem vom PC aus per SSH erledigen. Wie Sie Raspberry Pi OS vorab konfigurieren, wird im nachfolgenden Artikel ab Seite 116 im Detail erklärt.

Der Raspi Zero 2 W ist das aktuelle leistungsfähigste Mitglied der Zero-Familie, kostet aber nur gut einen Euro mehr als der Zero W. Hier ist ein ähnlicher SoC wie beim Raspi 3 mit vier 64-Bit-CPU-Kernen aufgelöst. Mit 1 GHz sind die Kerne aber etwas langsamer getaktet; das 2,4-GHz-WLAN- und Bluetooth-Modul ist das gleiche wie beim Raspi 3. Aufgrund der um Generationen leistungsfähigeren 64-Bit-CPU-Kerne empfehlen wir den Raspi Zero 2 W für alle Projekte, bei denen Sie einen besonders kompakten Raspi benötigen.

Der Vollständigkeit halber seien noch die Raspi Compute Modules erwähnt: Damit lassen sich Raspi-Projekte in eine Serienfertigung überführen, wobei der Raspberry Pi nur noch eine Komponente auf einem selbst entwickelten Board mit allen notwendigen Anschlüssen ist. Das Compute Module (CM, entspricht dem Ur-Raspi) und das Compute Module 3 (CM3, Raspi 3) haben dabei das Format eines Speicherriegels (SoDIMM), das Compute Module 4 (CM4, Raspi 4) hingegen wird mit zwei Kontaktleisten auf ein Board gesteckt. Für Privatanwender spie-

## Raspberry Pi: gängige Modelle

	Raspberry Pi	Raspberry Pi 3 / 3+	Raspberry Pi 4	Raspberry Pi 5
				
SoC, Taktfrequenz	BCM2835, 700 MHz	BCM2837, 1,2/1,4 GHz	BCM2711, 1,5 GHz	BCM2712, 2,4 GHz
CPU-Cores, Architektur	1 × ARM 11, 32 Bit	4 × ARM Cortex-A7, 64 Bit	4 × ARM Cortex-A72, 64 Bit	4 × ARM Cortex-A76, 64 Bit
GPU, Video	VideoCore IV, 1 × HDMI-A	VideoCore IV, 1 × HDMI-A	VideoCore VI, 2 × Micro-HDMI	VideoCore VII, 2 × Micro-HDMI
RAM	512 MByte	1 GByte	1 bis 8 GByte	2 bis 8 GByte
Ethernet	10/100 MBit/s	10/100 MBit/s	1 GBit/s	1 GBit/s
WLAN	–	IEEE 802.11 b/g/n, 2,4 GHz	IEEE 802.11b/g/n/a, 2,4 und 5 GHz	IEEE 802.11ac, 2,4 und 5 GHz
Bluetooth	–	4.1, BLE	5.0, BLE	5.0, BLE
USB	2 × 2.0	4 × 2.0	2 × 2.0, 2 × 3.0	2 × 2.0, 2 × 3.0
GPIO	26 Pins	40 Pins	40 Pins	40 Pins
Sonstiges	Composite Video (Cinch)	PoE-Header	PoE-Header	PoE+-Header, PCIe 2.0
Stromversorgung	5V/2A, Micro-USB	5V/2,5A, Micro-USB	5V/3A, USB-C	5V/5A, USB-C
Preis	ca. 40 bis 50 €	ca. 40 €	ca. 40 bis 80 €	ca. 50 bis 90 €

len die Compute Modules keine Rolle, man findet sie vielmehr in kommerziell und industriell produzierten Geräten.

## Unter Strom

Für die Stromversorgung des Ur-Raspi und aller Raspi Zero genügt ein altes Handy-Netzteil mit USB-Micro-Anschluss, das 5 Volt und 2 Ampere liefern kann. Diese gibt es schon für unter 5 Euro. Ab dem Raspberry Pi 3 sollte es ein Netzteil mit 3 Ampere sein.

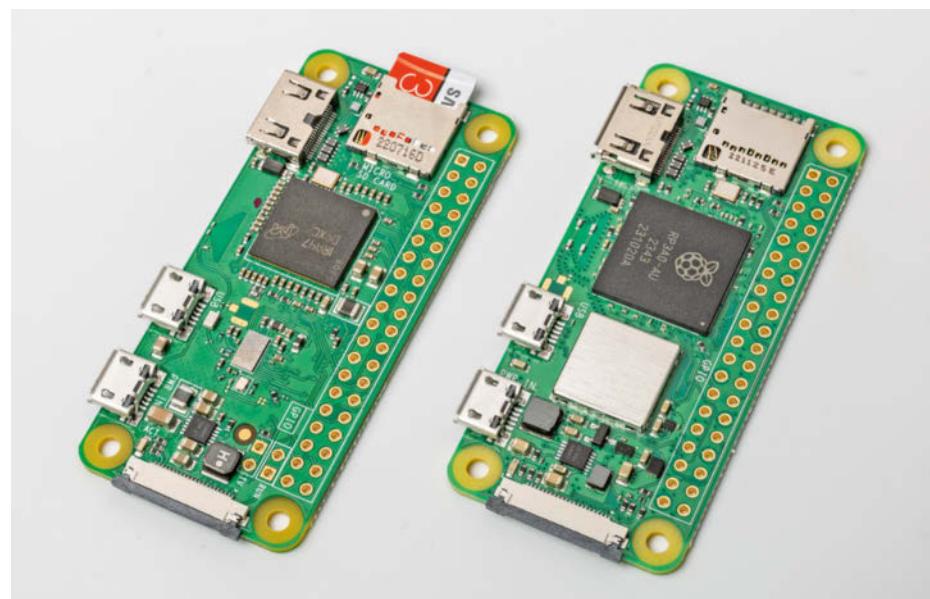
Beim Raspi 4 wurde der USB-Micro-Stromanschluss durch einen USB-C-Anschluss ersetzt. Doch die erste Fertigungsreihe hatte einen Konstruktionsfehler: Aufgrund eines fehlenden Widerstands war der Raspi nicht in der Lage, mit dem Netzteil 5 Volt und 3 Ampere zu vereinbaren. Dies betraf vor allem hochwertige USB-C-Netzteile und -Kabel. Deshalb produzierte die Raspi-Foundation spezielle Netzteile, die knapp 10 Euro kosten und auch ohne den Widerstand bis zu 3 Ampere Strom liefern. Außerdem wurden die Toleranzen der USB-Spezifikation ausgenutzt, weshalb das Raspi-4-Netzteil 5,1 Volt bereitstellt, sodass der Raspi geringfügig mehr Leistung abrufen kann.

Der Raspi 5 benötigt noch mehr Leistung, er braucht bis zu 27 Watt. Dafür ist ein Netzteil erforderlich, das bei maximal 5,1 Volt bis zu 5 Ampere Strom liefern kann. Viele auch hochwertige USB-C-Netzteile schaffen eine so hohe Leistung erst bei höheren Spannungen von 9 bis 20 Volt, die der Raspi 5 aber nicht nutzen kann. Deshalb hat auch hier die Raspi-Foundation ein spezielles Netzteil für den Raspi 5 entworfen, das Sie für gut 10 Euro bekommen.

## Welches Bootmedium?

Das traditionelle Bootmedium des Raspberry Pi sind SD- und MicroSD-Karten, von denen die Raspis der Generationen 1 und 2 ausschließlich booten. Die späteren Raspis 3 bis 5 besitzen ebenfalls einen MicroSD-Kartenschacht, booten aber alternativ auch von USB-Laufwerken oder ganz ohne Speichermedium übers Netzwerk. Für den Raspi 5 gibt es außerdem einen PCIe-SSD-Adapter, über den Sie sogar von NVMe-SSDs booten können.

Bei der Wahl des Bootmediums und etwaiger weiterer Speichermedien kommt es sehr auf den Einsatzzweck an. Die Zero-Modelle zum Beispiel werden meist für Steuer- und Regelungsaufgaben eingesetzt, etwa als Zentrale im Smart Home.



**Kaum größer als der 40-polige GPIO-Anschluss und nur 3 Zentimeter breit sind die Raspberry Pi Zero. Der Zero W (links) entspricht dem Ur-Raspi mit nur einem 32-Bit-CPU-Kern, hat aber das WLAN- und Bluetooth-Modul des Raspi 3 geerbt. Der Zero 2 W (rechts) bietet genau wie der Raspi 3 gleich vier 64-Bit-CPU-Cores plus WLAN und Bluetooth.**

Hier sind weder größere Datenmengen zu erwarten noch geht es darum, Dateien besonders schnell zu verarbeiten. Deshalb wird man beim Raspi Zero und Zero 2 in der Regel zu einer MicroSD-Karte mit 8 oder 16 GByte greifen. Das gilt auch für Raspi 1 und 2, die nur von MicroSD-Karte booten.

Ab dem Raspi 3 können Sie das Betriebssystem auch auf einem USB-Laufwerk installieren. Beim Raspi 3 ist das interessant, weil der Mini-Computer genau wie seine älteren Kollegen in dem Ruf steht, gelegentlich Speicherkarten „zu fressen“. Vor allem, wenn der Raspi nicht korrekt per `shutdown` heruntergefahren wurde, bevor Sie die Stromversorgung trennen, kann es dazu kommen, dass er nicht wieder bootet. Von USB-Laufwerken ist uns dieses Verhalten nicht bekannt, weshalb ein Marken-USB-Stick als Bootmedium eine geeignete Alternative ist.

Auch die Kühlung ist bei einem USB-Stick besser als im nicht gekühlten Kartenschacht auf der Unterseite des Raspi. Das spielt vor allem bei größeren Speicherkarten eine große Rolle, die mitunter den Hitzetod sterben. Wenn Sie den Raspi als Datei- oder Multimediaserver einsetzen wollen, also große Datenmengen bewegen, sollten Sie besser keine MicroSD als Halde verwenden: Den Karten fehlt ein leistungsfähiger Speicher-

controller, wie ihn handelsübliche SSDs besitzen und die etwa per Wear Level Management ein langes Leben der Flash-Chips garantieren. Außerdem ist die Kühlung von SSDs besser und die Laufwerke erreichen höhere Datentransferraten, weil sie auf mehrere Chips parallel zugreifen. Noch dazu sind SSDs mit 1 TByte oder mehr weitaus billiger als gleich große MicroSD-Karten.

Wenn es Ihnen nicht um das letzte Quäntchen Geschwindigkeit geht, sind externe USB-SSDs oder SATA-SSDs und ein USB-SATA-Adapter eine gute Wahl. Doch achten Sie unbedingt darauf, dass der USB-Controller des Laufwerks im besonders schnellen UAS-Modus (USB Attached SCSI) arbeitet. Bei USB-SATA-Adaptoren finden Sie die Abkürzung UAS oft in der Modellbezeichnung oder der Beschreibung. Schwieriger ist es bei externen USB-SSDs, hier finden Sie meist keine Angaben zum Controller. Nach unserer Erfahrung haben viele SSDs mit USB 3.1 Gen2 oder höher und USB-C-Anschluss einen UAS-Controller, etwa Samsungs Portable-SSD-Familie ab T5. Ob ein USB-Laufwerk einen UAS-Controller besitzt, finden Sie unter Raspberry Pi OS (etwa von MicroSD gestartet) mit folgendem Befehl im Terminal heraus:

```
dmesg -tW
```



**Die Compute Modules sind für Hardwarehersteller gedacht, die eigene Geräte auf der Basis des Raspberry Pi herstellen. Das erste Compute Module (links) in Form eines Speicherriegels ist ein fast vollständiger Ur-Raspi. Mit dem CM3-Adapter in der Mitte lässt sich das Compute Module 4, das einem Raspi 4 entspricht, auch in älteren Geräten nutzen, die noch für das Compute Module 3 (Raspi 3) entwickelt wurden.**

Danach schließen Sie das zu prüfende Laufwerk an und achten auf die Meldungen des Kernels. Handelt es sich um einen UAS-Controller, so finden Sie die Meldung

```
scsi host0: uas
```

Meldet der Kernel stattdessen „usb-storage“, handelt es sich um einen weitaus langsameren USB-Controller. Diese findet man noch häufig bei USB-Festplatten – deren hohe Stromaufnahme mitunter die Kapazität des USB-Ports übersteigt, weshalb es erforderlich werden kann, eine USB-Festplatte mit einem eigenen Netzteil zu versorgen.

## Betriebssystem-Wahl

Bei der Wahl des Betriebssystems und beim Erstellen des Bootmediums unterstützt Sie der Raspberry Pi Imager, kurz Raspi-Imager oder `rpi-imager` genannt. Der Artikel „Bootmedien schmieden“ auf Seite 116 beschreibt die Möglichkeiten des Programms im Detail. Das Standard-Betriebssystem für den Raspi heißt Raspberry Pi OS und es existiert in drei Varianten:

- Desktop (ca. 1 GByte) für den Betrieb mit Monitor, Tastatur und Maus
- Full (ca. 3 GByte) mit Desktop und mehr Anwendungen, um den Raspi als PC-Ersatz mit Monitor, Tastatur und Maus zu nutzen
- Lite (unter 500 MByte), schlank und ohne Desktop, wenn der Raspi aus der Ferne bedient wird, als Server dient oder (Netzwerk-)Dienste bereitstellt

Die Images für Full und Lite finden Sie auf der Download-Seite (siehe ct.de/y1xh) in der Rubrik „Raspberry Pi OS (other)“. Die Lite-Variante ist nicht nur erheblich kleiner als die Desktop-Variante, sondern erzeugt weniger CPU-Last und belegt auch weniger RAM – gerade bei älteren Raspis ein wertvolles Gut.

Die neuesten Raspi-OS-Abbilder basieren auf Debian Bookworm, dem aktuellen Stable Release der Linux-Distribution Debian. Bei manchen Raspis werden Ihnen noch ältere Abbilder mit Debian Bullseye als Basis angeboten. Diese sollten Sie nur dann wählen, wenn es dafür einen besonderen Grund gibt: etwa weil Sie einen Raspberry Pi Zero oder Zero W benutzen wollen, der bei Redaktionsschluss mit dem von Bookworm abgeleiteten Image nicht bootete. Auch wenn Sie eine ältere Anleitung für ein Raspi-Projekt finden, kann es sein, dass die einzelnen Schritte nicht mit dem aktuellen Bookworm-System funktionieren. Dann ist es durchaus gerechtfertigt, auf ein älteres System zurückzugreifen. Eine ältere Version von Raspberry Pi OS zu aktualisieren kann allerdings schwierig bis unmöglich sein, Details dazu finden Sie im Artikel auf Seite 116.

## Vom Stick zur SSD

Die Installation von Raspberry Pi OS auf USB-SSDs unterscheidet sich nicht von der auf andere USB-Speichermedien. Anders ist es beim Raspberry Pi 5, wenn Sie den SSD-Adapter verwenden und von einer

NVMe-SSD booten wollen. Die SSD können Sie nicht ohne spezielle USB-Adapter an den PC anschließen. Aber Sie können den Raspberry Pi benutzen, um das Betriebssystem auf der SSD zu installieren.

Dazu benötigen Sie zunächst einen 16 GByte großen USB-Stick oder eine Micro-SD-Karte. Dort installieren Sie mithilfe Ihres PCs wie im nachfolgenden Artikel beschrieben Raspberry Pi OS mit Desktop. Schließen Sie Monitor, Tastatur, Maus und die NVMe-SSD an, dann starten Sie den Raspi vom gerade erzeugten Bootmedium. Am Desktop angelangt starten Sie über das Anwendungsmenü links oben unter Zubehör den Imager auf dem Raspi – und wählen als Ziel die NVMe-SSD aus. Weitere Besonderheiten gibt es nicht zu beachten.

Allerdings bootet der Raspi nicht standardmäßig von der NVMe-SSD, sondern bevorzugt MicroSD-Karten oder den noch angeschlossenen USB-Stick, den Sie für die Installation benutzt haben. Mit dem Programm `raspi-config` können Sie aber später die Bootreihenfolge ändern, sodass der Raspi 5 als erstes von SSD bootet. Doch dazu später mehr.

## Start frei

Haben Sie beim Erstellen des Bootmediums keine Einstellungen vorgenommen, erwartet Sie beim ersten Start ein Dialog, der die wichtigsten Daten wie Sprache, Benutzername und Passwort abfragt und die WLAN-Verbindung herstellt. Dazu müssen Sie Monitor, Tastatur und Maus anschließen. Zum Abschluss können Sie, sofern eine Internetanbindung besteht, Raspberry Pi OS auch gleich aktualisieren. Das sollte auch auf einem vorab konfigurierten System Ihr erster Schritt sein, dazu führen Sie im Terminal oder per SSH folgende Befehle aus:

```
sudo apt-get update
sudo apt-get full-upgrade
```

Falls Sie WLAN nutzen und sich per SSH auf dem Raspi einloggen, sollten Sie noch die Stromsparfunktion des WLAN-Moduls deaktivieren. Dazu fügen Sie in die Datei `/etc/rc.local` als vorletzte Zeile folgendes ein:

```
/usr/sbin/iw wlan0 set power_save off
```

Danach ist ein Neustart des Raspi erforderlich.

Als Nächstes sollten Sie die Einstellungen des Raspi kontrollieren und ergän-

zen – vor allem dann, wenn Sie sie mit dem Raspi-Imager vorgenommen haben. Dazu gibt es das grafische Tool, das Sie im Menü unter „Einstellungen/Raspberry Pi Konfiguration“ finden. Wir empfehlen Ihnen aber, stattdessen `sudo raspi-config` im Terminal aufzurufen, denn es bietet mehr Einstellungsmöglichkeiten als die grafische Variante. Außerdem können Sie letztere Variante auch aus der Ferne per SSH nutzen.

Ein wichtiger Punkt, der sich nicht vorab mit dem Raspi-Imager konfigurieren lässt, verbirgt sich unter „Localisation Options“. Hier überprüfen Sie unter „Locale“ die aktuell verwendete Zeichentabelle. Achten Sie darauf, dass in der Liste „de\_DE.UTF-8“ ausgewählt und außerdem „en\_GB.UTF-8“ deaktiviert ist. Damit stellen Sie sicher, dass etwa deutsche Umlaute in Dateinamen kein Problem sind und Programme, sofern verfügbar, auf Deutsch erscheinen.

Unter „Interface Options“ finden Sie neben SSH auch verschiedene Schnittstellen des GPIO-Anschlusses, die wichtigsten sind SPI (Serial Peripheral Interface) und I2C (Inter-Integrated Circuit). Die SPI-Schnittstelle dient zum Beispiel zum Anschluss von Displays, während über den I2C-Bus Touchscreens oder auch Erweiterungen wie Schrittmotor-Controller angesteuert werden. Bevor Sie die Schnittstellen nutzen können, müssen Sie sie per `raspi-config` aktivieren.

Manche Erweiterungen wie die gerade erwähnte Schrittmotorsteuerung gibt es als sogenannte HATs (Hardware Attached on Top). Das sind kompakte Platinen, die Sie oft auf den 26- oder 40-poligen GPIO-Anschluss des Raspi aufstecken. Bei manchen HATs ist der GPIO durchgeschleift, sodass Sie dort weitere HATs aufstecken oder Pins für eigene Zwecke nutzen können. Beliebte HATs sind das PoE-HAT (Power over Ethernet) für Raspi 3+ und 4, das 3-Kanal-Relais-HAT, das Servo Driver HAT, das Stepper Motor Hat für Schrittmotoren, ein KI-Beschleuniger-HAT für den Raspi 5 und das HatDrive, mit dem Sie NVMe-SSDs an den Raspi 5 anschließen können. Ob und welche Protokolle Sie für den Betrieb dieser HATs benötigen, steht in den jeweiligen Beschreibungen oder Anleitungen.

## Nachgelegt

Als ein von Debian abgeleitetes Betriebssystem verwendet Raspberry Pi OS das DEB-Paketformat, um Software zu ver-

```
mdoelle@raspberrypi: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
mdoelle@raspberrypi: $ dmesg -tW
usb 4-1: new SuperSpeed USB device number 2 using xhci-hcd
usb 4-1: New USB device found, idVendor=04e8, idProduct=4001, bcdDevice= 1.00
usb 4-1: New USB device strings: Mfr=2, Product=3, SerialNumber=1
usb 4-1: Product: PSSD T7
usb 4-1: Manufacturer: Samsung
usb 4-1: SerialNumber: S7MGN50X501859T
scsi 0:0:0:0: Direct-Access      Samsung PSSD T7          0   PQ: 0 ANSI: 6
sd 0:0:0:0: Attached scsi generic sg0 type 0
sd 0:0:0:0: [sda] 3907829168 512-byte logical blocks: (2.00 TB/1.82 TiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Mode Sense: 43 00 00 00
sd 0:0:0:0: [sda] Write cache: enabled, read cache: enabled, doesn't support DPO or FUA
sd 0:0:0:0: [sda] Preferred minimum I/O size 512 bytes
sd 0:0:0:0: [sda] Optimal transfer size 33553920 bytes
sd 0:0:0:0: [sda] Attached SCSI disk
EXT4-fs (sda): mounted filesystem 4e1bd94d-f7dd-4031-b68b-79060200c400 r/w with ordered data mode. Quota mode: none.
ieee80211 phy0: brcmf_p2p_send_action_frame: Unknown Frame: category 0xa, action 0x8
```

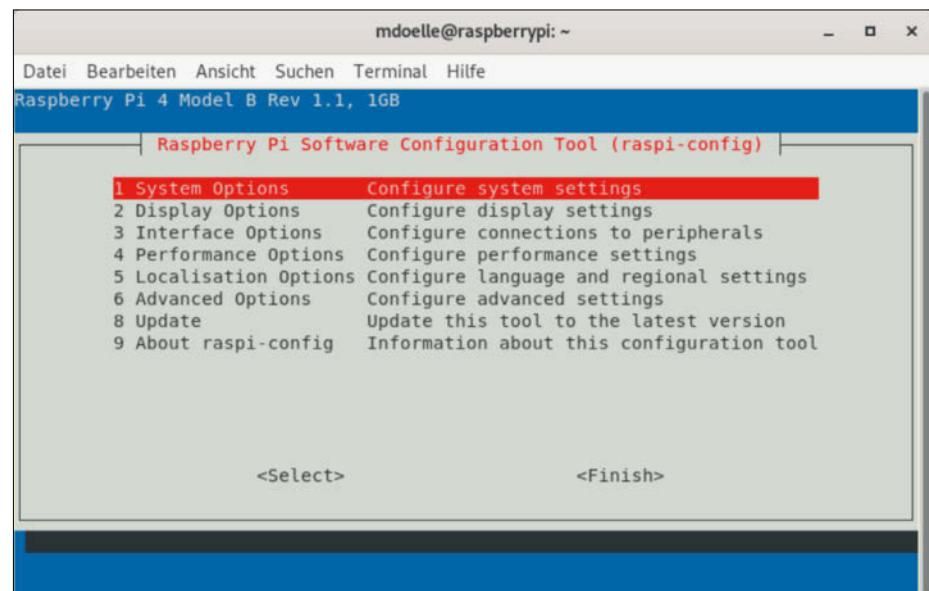
**USB-SSDs und USB-SATA-Adapter mit UAS-kompatiblen Controllern sind weitaus schneller als herkömmliche USB-Speichermedien. Ob ein Laufwerk einen UAS-Controller besitzt, finden Sie über die Kernel-Meldungen beim Anschließen des Geräts heraus.**

walten. Eine Stärke dieses Formats ist es, Abhängigkeiten zwischen Softwarepaketen festzustellen und automatisch aufzulösen. Das Spiel Dave Gnuke zum Beispiel, das dem bekannten Duke Nukem nachempfunden wurde, besteht aus zwei Teilen, also Paketen: dem eigentlichen Spiel mit dem Paketnamen „davegnukem“ und dem Paket „davegnukem-data“, das die Spieldaten wie den Spielplan, Grafiken und Sounds enthält. Ohne die Spieldaten funktioniert das Spiel nicht, weshalb das Paket „davegnukem“ eine Abhängigkeit zu „davegnukem-data“ besitzt.

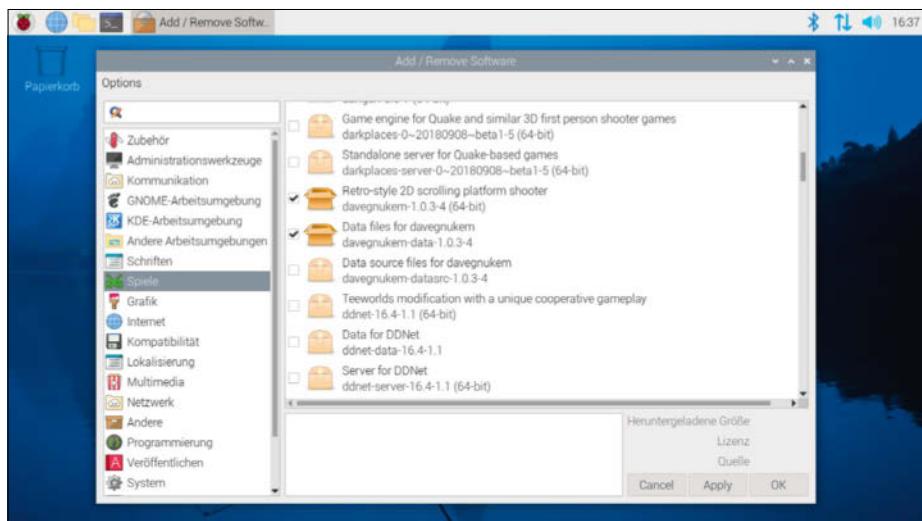
Die Paketmanager – den grafischen finden Sie im Menü unter „Einstellungen“, „Add/Remove Software“ – löst sol-

che Abhängigkeiten automatisch auf: Wenn Sie in der Rubrik „Spiele“ lediglich bei dem Paket „davegnukem“ einen Haken setzen und dann mit „Apply“ die Auswahl übernehmen, installiert die Paketverwaltung nach der obligatorischen Passwortabfrage automatisch beide Pakete (und falls erforderlich weitere Abhängigkeiten). Wenn Sie anschließend noch einmal im Paketmanager zu Dave Gnuke herunterscrollen, sehen Sie, dass nun beide Pakete angehakt, also installiert sind.

Die Aufteilung eines Programms in mehrere Pakete hat vor allem für Softwareentwickler Vorteile. So können sie Änderungen an der Spielewelt vorneh-



**Die Terminal-Variante des Raspi-Konfigurationsprogramms bietet Ihnen weitaus mehr Einstellungsmöglichkeiten als das GUI. Außerdem können Sie es auch per SSH aus der Ferne nutzen.**



**Raspberry Pi OS verwendet die leistungsfähige Paketverwaltung des Debian-Projekts. Damit können Sie komfortabel weitere Software installieren oder bereits installierte entfernen. Etwaige Abhängigkeiten zu anderen Paketen wie zum Beispiel Bibliotheken löst die Paketverwaltung automatisch auf.**

men, ohne das Paket mit dem Spiel aktualisieren zu müssen – oder einen Bug im Spiel fixen, ohne die Spielewelt antasten zu müssen. Außerdem können Programmierer sich viel Arbeit ersparen, indem Sie auf die Arbeit anderer zurückgreifen – in Form sogenannter Bibliotheken (DLL unter Windows). So muss nicht jedes Videoprogramm aufs Neue lernen, wie ein H.264-Video aufgebaut ist, sondern es bindet einfach eine H.264-Bibliothek ein und nutzt die dort enthaltenen Funktionen – womit automatisch das DEB-Paket der Bibliothek eine Abhängigkeit des Programms wird.

Im Terminal oder per SSH ruft man die Paketverwaltung üblicherweise direkt auf. Wollen Sie „Dave Gnuke“ über die Kommandozeile installieren, müssen Sie zunächst den korrekten Namen des Pakets ermitteln:

```
apt-cache search dave
```

Hier bekommen Sie alle Pakete aufgelistet, in deren Namen oder Beschreibung der Suchbegriff vorkommt. Eines der Ergebnisse ist:

```
davegnukem - Retro-style 2D ↵
scrolling platform shooter
```

Um das Paket (nebst Abhängigkeiten) zu installieren, benutzen Sie den Befehl:

```
sudo apt-get install davegnukem
```

Dabei können Sie Tippfehler vermeiden, indem Sie die Befehlsvervollständigung mit der Tabulatortaste verwenden. Diese funktioniert anders als unter Windows und ergänzt Befehle nur dann, wenn sie eindeutig sind. Haben Sie „sudo apt-get i“ eingetippt und drücken dann die Tab-Taste, so hängt das Terminal an das „i“ nur ein „n“ an, weil es mehrere passende Parameter gibt. Welche das sind, erfahren Sie, wenn Sie ein zweites Mal die Tab-Taste drücken – „in-dextargets“ und „install“. Indem Sie noch ein „s“ eintippen, der Parameter also „ins“ lautet, können Sie den Befehl mit Tab zu „sudo apt-get install“ vervollständigen. Nun fehlt noch der Name des Pakets oder der Pakete, die Sie installieren wollen. Für „davegnukem“ genügt es, wenn Sie „dave“ eintippen und erneut Tab drücken. Erfahrene Linux-Nutzer erkennt man typischerweise daran, dass die Tabulatortaste ihrer Tastatur völlig abgegriffen ist. Anders als bei der grafischen Paketverwaltung bekommen Sie auf der Kommandozeile angezeigt, welche weiteren Abhängigkeiten es gibt und welche Pakete nun installiert werden – bei Dave Gnuke sind das mindestens „davegnukem“ und „davegnukem-data“.

Der Befehl, um ein Paket wieder loszuwerden, lautet analog sudo apt-get remove gefolgt von dem Paketnamen – auch hier ist die Tabulatortaste Ihr bester Freund. Der Unterschied zu install ist, dass remove sich ausschließlich auf das angegebene Paket bezieht sowie auf Pakete, die von dem genannten abhängig sind. Bei

Dave Gnuke ist zwar das Programm „davegnukem“ von den Spieldaten in „davegnukem-data“ abhängig, weil das Spiel ohne die Daten nicht starten kann. Umgekehrt gilt das jedoch nicht, Sie können „davegnukem-data“ auch ohne das Spiel installieren. Folglich wird bei dem Befehl

```
sudo apt-get remove davegnukem
```

nur das Paket mit dem Spiel entfernt. Das Programm weist Sie aber darauf hin, dass das Paket „davegnukem-data“ als Abhängigkeit mit installiert wurde und nun nicht mehr benötigt wird. Sie könnten es also ebenfalls mit remove entfernen oder gleich zu Anfang das Paket „davegnukem-data“ löschen lassen, womit automatisch auch „davegnukem“ entfernt würde.

Bei Paketen mit vielen Abhängigkeiten wäre es sehr mühsam, nachträglich nun nicht mehr benötigte Pakete zu löschen, weshalb es den Aufruf

```
sudo apt-get autoremove
```

gibt, der ohne angehängten Paketnamen alle Pakete löscht, die früher mal als Abhängigkeiten installiert wurden, aber nun nicht mehr gebraucht werden.

Wir empfehlen Ihnen aber den Aufruf:

```
sudo apt-get autoremove dukegnukem
```

In dieser Form deinstalliert die Paketverwaltung in einem Rutsch das Paket „dukegnukem“ und alle anderen Pakete, die als Abhängigkeiten von „dukegnukem“ mit installiert wurden und deshalb nun nicht mehr gebraucht werden. So bleibt Ihr Raspberry Pi OS schlank.

Indem Sie sich mit der Paketverwaltung vertraut machen, haben Sie die Grundlagen geschaffen, Projekte mit dem Raspberry Pi umzusetzen – etwa indem Sie eine der Anleitungen aus dieser Ausgabe befolgen. Auch Installationsanleitungen aus dem Internet setzen häufig voraus, dass Sie mit der Paketverwaltung umgehen und etwaige fehlende Bibliotheken oder andere Pakete ohne Hilfe nachinstallieren können.

(mid@ct.de) ct

## Literatur

[1] Lutz Labs, Christof Windeck, Raspadapter, SSD- und PCIe-Adapter für den Raspberry Pi 5, c't 19/2024, S. 82



# Hören Sie von uns – Podcasts aus den heise-Redaktionen

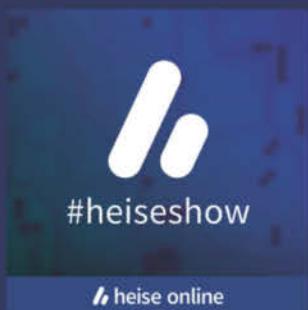
kurz informiert –  
der Tech-News-Podcast  
(zweimal werktäglich)



KI-Update – Infos zu  
den wichtigsten KI-  
Entwicklungen (werktäglich)



#heiseshow – die  
wöchentliche Dosis Technik  
und Netzpolitik (donnerstags)



Passwort – der Podcast  
von heise Security  
(14-täglich, mittwochs)



c't uplink – der Podcast  
aus Nerdistan  
(samstags)



Haken dran – das Social-  
Media-Update der c't  
(dreimal pro Woche)



Bit-Rauschen – der  
Prozessor-Podcast von c't  
(14-täglich, mittwochs)



Auslegungssache –  
der c't-Datenschutz-  
Podcast (14-täglich, freitags)



Vorsicht, Kunde! –  
der c't-Verbraucherschutz-  
Podcast (14-täglich, freitags)



Frauen und Technik –  
mit Eckert und Wolfangel  
(14-täglich, mittwochs)



Mac & i – der Apple-  
Podcast (14-täglich)



The Next (Big) Thing –  
alles rund um Apples Vision  
Pro (14-täglich, freitags)



Überall, wo es Podcasts gibt!

© Copyright by Heise Medien.

Hier hören:  
[heise.de/go/podcast](http://heise.de/go/podcast)



# Bootmedien schmieden

## Betriebssysteme für den Raspberry Pi auf MicroSD, USB-Stick oder SSD schreiben

**Der Raspberry Pi Imager bietet Orientierung im Dschungel der zahlreichen Versionen und Varianten von Raspberry Pi OS: Mit wenigen Klicks finden Sie das geeignete Betriebssystem für Ihren Raspi und können ihn sogar in Betrieb nehmen, ohne Monitor, Tastatur und Maus anzuschließen.**

Von Mirko Dölle

Ohne Betriebssystem läuft nichts: Genau wie PCs benötigt auch der Raspberry Pi ein Betriebssystem, um zu arbeiten. Das unterscheidet den Mini-Computer von den Mikrocontrollern wie dem Arduino, die nur mit einer Firmware bespielt werden. Die große Anzahl an Boot-Images macht es allerdings schwer, das richtige auszuwählen: Raspberry Pi OS gibt es in drei Varianten jeweils für 32 und 64 Bit, zudem gibt es Projekte wie LibreELEC, die ebenfalls Bootmedien bereitstellen.

Um ein Bootmedium für den Raspberry Pi herzustellen, egal ob MicroSD-Karte, USB-Stick oder SSD, empfehlen wir den Raspberry Pi Imager, den Sie sich für Windows, macOS und Linux auf [raspberrypi.com](https://raspberrypi.com) herunterladen können. Das Programm hilft Ihnen dabei, den Durchblick bei all den Boot-Images zu behalten, die es für die verschiedenen Raspi-Modelle gibt. Vor allem die Entscheidung, ob noch ein 32-Bit-Image benötigt wird oder ob Sie bereits das Standard-64-Bit-Image einsetzen können, fällt mit dem Raspi-Imager leicht. Das gewählte Image können Sie auf Ihrem gewohnten Rechner bequem konfigurie-

ren und anschließend betriebsbereit auf Ihr Raspi-Bootmedium übertragen.

Laden Sie dazu zunächst den Raspi-Imager für Ihr Betriebssystem herunter und installieren Sie ihn mit den Bordmitteln. Unter Windows und macOS rufen Sie einfach das Programm auf, unter Linux bemühen Sie die Paketverwaltung Ihrer Linux-Distribution. Anschließend starten Sie den Raspi-Imager – unter Linux zunächst ohne Root-Rechte (ohne sudo). Sobald das Programm zusätzliche Rechte benötigt, um das Bootmedium zu schreiben, wird es diese explizit anfordern.

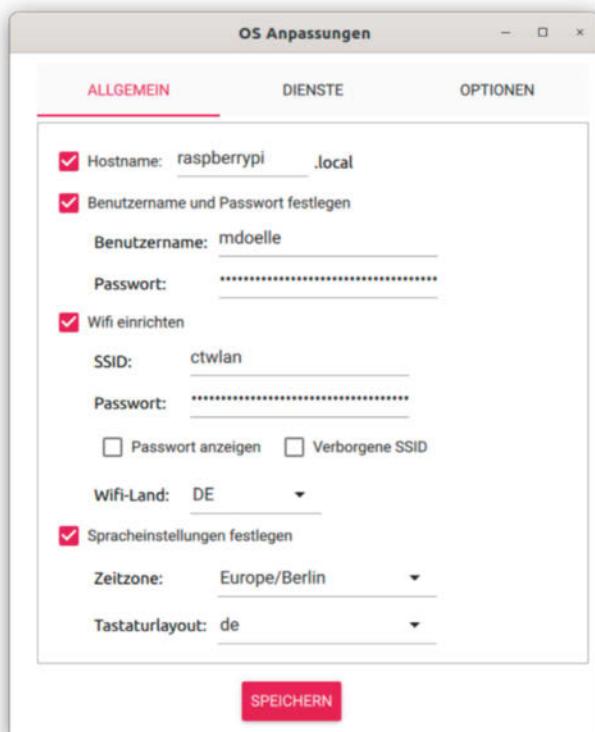
Beginnen Sie damit, Ihr Raspi-Modell auszuwählen. Achten Sie dabei auf die exakte Bezeichnung und die Modellangabe, sie ist bei allen Raspberry Pi auf der Platine aufgedruckt. So ist es ein großer Unterschied, ob Sie einen Raspberry Pi Zero W oder Raspberry Pi Zero 2 W auswählen – die Bootmedien unterscheiden sich fundamental.

### Qual der Wahl

Haben Sie das genaue Modell gefunden, bietet Ihnen der Raspi-Imager nur noch zur Hardware passende Abbilder an. Ganz oben in der Liste steht dabei die Desktopvariante des aktuellen Raspberry Pi OS, in der Unterrubrik „Raspberry Pi OS (other)“ finden Sie aber noch weitere Varianten, etwa mit zusätzlichen Anwendungen, ohne Desktop oder schlicht ältere Versionen des Betriebssystems. Es gibt noch weitere Unterrubriken, etwa „Media Player OS“ mit LibreELEC, das den Raspi in eine Medioplayer-Settop-Box verwandelt, oder RetroPie unter „Emulation and game OS“, womit Sie alte Spiele auf dem Raspi spielen können.

Die Rubrik „Misc Utility Images“ enthält außerdem ein paar Sonder-Bootmedien für den Raspi: Damit können Sie die Firmware Ihres Raspi aktualisieren und außerdem die Bootreihenfolge ändern, sodass der Raspi beispielsweise zunächst

**Auf Knopfdruck übernimmt der Raspi-Imager die WLAN-Konfiguration Ihres PCs und andere Einstellungen in das Bootmedium für den Raspberry Pi. So können Sie den Raspi gleich nach dem ersten Start ohne Tastatur, Maus und Monitor nutzen.**



## ct kompakt

- Der Raspberry Pi Imager weiß genau, welche Betriebssystem-Images für welche Raspis geeignet sind und hilft Ihnen bei der Auswahl.
- Benutzernamen und WLAN-Zugangsdaten liest der Imager direkt von Ihrem PC aus und trägt sie auf Wunsch im Betriebssystem des Raspi ein.
- Mit nur einem Knopfdruck erzeugen Sie sichere Schlüssel; so brauchen Sie keine Passwörter für den Fernzugang per SSH.

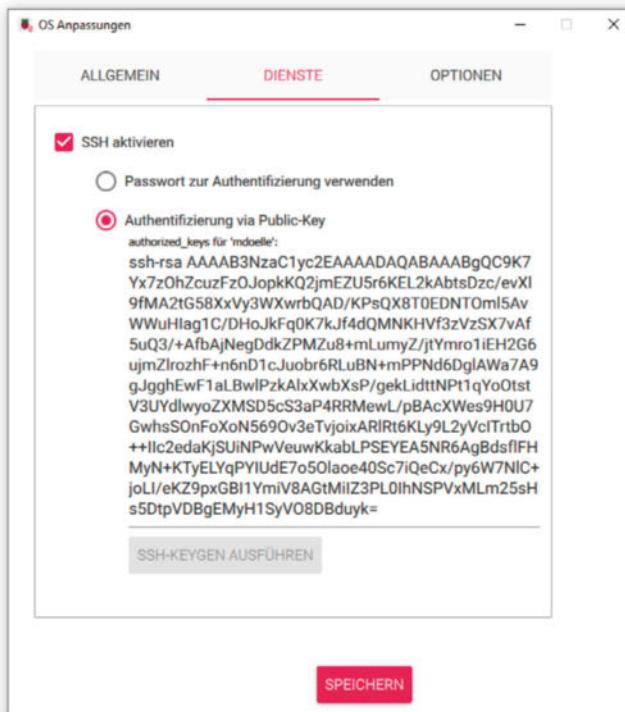
nach einer MicroSD-Karte sucht und dann von einem USB-Laufwerk bootet – oder auch anschließend über das Netzwerk. Für den Raspi 5 gibt es außerdem ein Image, mit dem der Raspi künftig zuerst von einer NVMe-SSD startet, die mit einem entsprechenden Adapter per PCIe angebunden ist.

### Vorkonfiguriert

Der Clou des Raspi-Imagers ist aber, dass Sie die Einstellungen anpassen können, unmittelbar bevor Sie es auf das Bootmedium übertragen. Das ist vor allem für Raspis interessant, die Sie ohne Monitor und Tastatur betreiben wollen, sich also nicht einfach bei Raspberry Pi OS anmelden können, um die Konfiguration abzuschließen.

Besonders praktisch ist die WLAN-Konfiguration: Wenn Ihr Rechner gerade per WLAN mit Ihrem Heimnetz verbunden ist, genügt ein Klick auf „Wifi einrichten“, und der Raspi-Imager übernimmt Ihre aktuelle WLAN-Konfiguration inklusive SSID und Passwort auch für den Raspi – leider jedoch nicht die Ländereinstellung. Dort müssen Sie unbedingt von Hand Deutschland auswählen, damit Ihr Raspi später nicht auf den falschen Frequenzen funkt. Auch die Spracheinstellungen übernimmt der Raspi-Imager aus Ihrem laufenden System, nicht jedoch das Tastatur-Layout – hier müssen Sie ebenfalls nacharbeiten.

Auch den Hostnamen des Raspi und den Namen des Standardbenutzers sowie dessen Passwort können Sie im Raspi-Imager festlegen. Über den Hostnamen können Sie in den meisten Heimnetzen



**Der Imager macht es besonders leicht, beim Fernzugang per SSH auf Passwörter zu verzichten: Ist in Ihrem System noch kein SSH-Key gespeichert, legt er auf Knopfdruck einen neuen auf Ihrem PC an und trägt ihn in Raspberry Pi OS ein.**

den Raspi ansprechen, ohne dessen IP-Adresse kennen zu müssen. Wenn Sie schon wissen, was Sie mit dem Raspi später einmal anfangen werden, sollten Sie den Projektnamen anstelle des Standardwerts „raspberrypi“ eintragen. So können Sie später mehrere Raspis in Ihrem Heimnetz leicht auseinanderhalten.

Beim Benutzernamen übernimmt der Raspi-Imager ebenfalls den, mit dem Sie gerade angemeldet sind – weshalb Sie das Programm unter Linux auch ohne sudo aufrufen sollten. Das funktioniert allerdings nicht, wenn Leerzeichen, Umlaute und andere Sonderzeichen enthalten sind. Wenn Sie Ihren PC-Benutzernamen für den Standard-Raspi-Benutzer übernehmen, müssen Sie ihn später nicht gesondert angeben, wenn Sie sich mittels SSH über das Netzwerk auf dem Raspi einloggen.

Sich aus der Ferne anzumelden, funktioniert allerdings nur, wenn Sie SSH unter „Dienste“ aktiviert haben. Standard ist, dass Sie dabei das Passwort eingeben müssen. Leichter und auch sicherer ist es jedoch, einen Schlüssel zu verwenden. Indem Sie auf „SSH-KEYGEN ausführen“ klicken, erzeugt der Raspi-Imager ein Schlüsselpaar auf Ihrem PC und trägt den öffentlichen Schlüssel unter Raspberry Pi OS ein. Keine Sorge, dabei wird kein schon existierender Schlüssel überschrieben – vielmehr trägt der Raspi-Imager diesen automatisch in der Maske ein. Und noch ein Hinweis: Heutzutage benötigt nie-

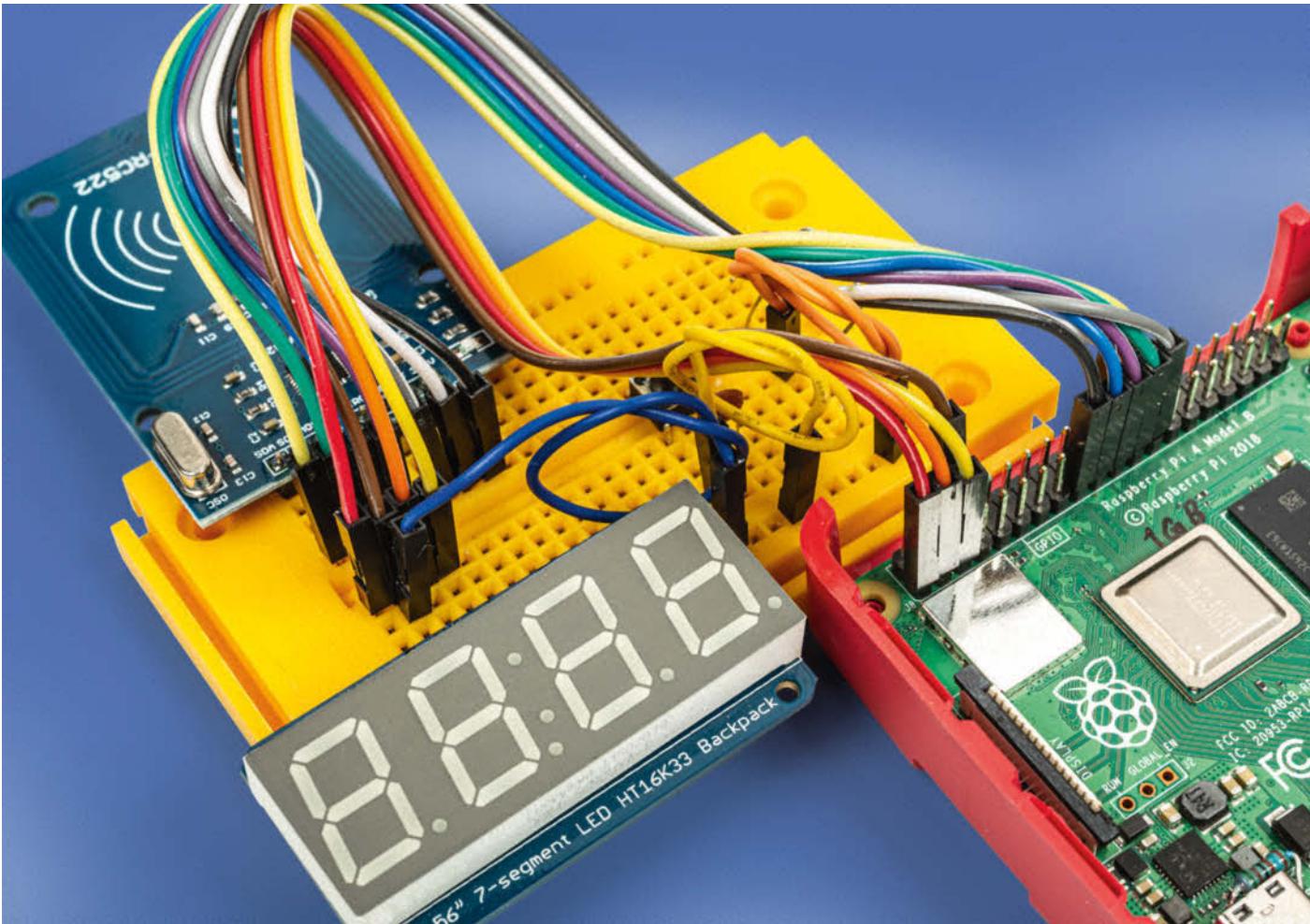
mand mehr „PuTTY“ oder andere SSH-Clients: OpenSSH gehört auch unter Windows schon seit Jahren zu den Standard-Tools in der PowerShell und der Eingabeaufforderung.

Speichern Sie Ihre Einstellungen und lassen Sie das fertig konfigurierte Betriebssystem auf das Bootmedium schreiben. Das kann je nach Geschwindigkeit des Speichermediums durchaus zehn Minuten dauern, zumal der Raspi-Imager das Bootmedium anschließend noch einmal überprüft. Mit dem Bootmedium ist Ihr Raspberry Pi unmittelbar einsatzbereit.

(mid@ct.de) ct

## Projektinfo

- Bootmedien für den Raspberry Pi herstellen und konfigurieren
- Grundkenntnisse Bedienung von Windows, macOS oder Linux
- 15 Minuten
- ab ca. 5 Euro
- MicroSD-Speicherkarte ab 8 GByte, USB-Stick oder USB-SSD
- MicroSD-USB-Adapter



# Schaltzentrale

## Den GPIO-Anschluss des Raspi programmieren

**Der GPIO-Anschluss mit zahlreichen Ein- und Ausgängen macht den Raspi für Elektronik-Basteleien besonders interessant. Wir erklären Ihnen, wie Sie LEDs und Taster direkt anschließen, Widerstände und Kondensatoren berechnen und so den Raspi in Ihre eigene Schaltung integrieren.**

Von Mirko Dölle

**E**in Einplatinencomputer, der Schulkindern Informatik und Elektronik näher bringen soll, das war das erklärte Entwicklungsziel der Raspberry Pi Foundation. Der zunächst 26- und später 40-polige GPIO-Anschluss (General Purpose Input/Output, universelle Ein-/Ausgabe) ermöglicht, Elektronikschaltungen zu entwickeln oder mit fertig aufgebauten Modulen den Funktionsumfang des Raspi zu erweitern. So können Sie von einer blinkenden LED bis zum Roboterarm alles steuern. Wir zeigen Ihnen, wie Sie von der LED bis zum RFID-Kartenleser unterschiedlichste Komponenten an den Raspi anschließen und die richtigen Bauteile und Bauteilewerthe für Ihre Schaltungen auswählen.

Von den 40 Pins des GPIO-Anschlusses können Sie 28 für unterschiedliche Schaltaufgaben verwenden, die restlichen

dienen der Stromversorgung oder sind Masse-Pins. Ganz wichtig: Der Raspi arbeitet grundsätzlich mit 3,3-Volt-Pegeln und verkraftet keine höheren Spannungen. Die Pins 2 und 4 führen die Betriebsspannung VIN des Raspi von 5 Volt und dienen dazu, den Raspi oder Ihre Zusatzschaltung mit Strom zu versorgen. In der Übersicht der GPIO-Pins sind sie lachsrosa gefärbt. Pin 1, zu erkennen am rechteckigen Lötanschluss auf der Rückseite des Raspi, und Pin 17 (rot) liefern 3,3 Volt (VCC). Diese Pins können Sie ebenfalls zur Spannungsversorgung verwenden, dürfen sie aber auch direkt mit Eingängen des Raspi verbinden – aber bitte stets mit einem Vorwiderstand von 1,8 bis 2,2 kΩ (Kilohm) zwischen Eingang und VCC.

Der Hintergrund ist, dass Sie alle 28 schaltbaren Pins sowohl als Eingänge als

## c't kompakt

- Vor dem Anschluss von Bauteilen und Modulen an den GPIO-Port des Raspi muss man die korrekten Vorwiderstände und Spannungsteiler berechnen.
- Für einfache Aufgaben wie eine LED ein- und auszuschalten gibt es das Kommandozeilen-Tool `pigs`.
- Python ist die eigentliche Programmiersprache des Raspi, sie eröffnet Ihnen den Zugriff auf unzählige Bibliotheken für Displays, RFID-Leser, Servo-Controller und Schrittmotorsteuerungen.

auch als Ausgänge konfigurieren können. Verbinden Sie VCC direkt mit einem Pin, der noch als Ausgang mit Massepegel (GND) geschaltet ist, gibt es einen Kurzschluss und der Raspi geht im Zweifel kaputt. Ein Vorwiderstand im Bereich von 1,8 bis 2,2 kΩ begrenzt den Strom auf unter 2 mA (Milliampere), was unkritisch ist. Ist der Pin als Eingang konfiguriert, fließt fast gar kein Strom; die Spannung am Eingang steigt auf 3,3 Volt an, so als gäbe es gar keinen Vorwiderstand.

### Programmierter Widerstand

Nach dem gleichen Prinzip funktionieren die internen Vorwiderstände an den Eingängen, die Sie als Pull-Up (vom Eingang zu 3,3 V) respektive Pull-Down (vom Eingang zu Masse) konfigurieren können. Indem Sie den Stromfluss auf unter 2 mA begrenzen, dürfen Sie den Eingang etwa über einen Taster gefahrlos direkt mit einem Masse-Pin oder VCC verbinden, ohne dass der Raspi Schaden nimmt.

Es gibt noch einen zweiten Grund für den Einsatz von Pull-Up- und Pull-Down-Widerständen: Angenommen, Sie würden einen Taster an Pin 5 anschließen, der den Pin auf Tastendruck mit Masse (GND, schwarz) an Pin 6 verbindet. Ohne einen Pull-Up-Widerstand gegen VCC befände sich Pin 5 in einem undefinierten Zustand, könnte auf Spannungen in der Umgebung reagieren und so ungewollt auslösen. Der Pull-Up-Widerstand stellt sicher, dass an Pin 5 immer 3,3 Volt anliegen, bis zu dem Moment, in dem Sie den Taster drücken und damit Pin 5 mit Masse verbinden – man sagt auch „auf Masse ziehen“. Da an dem

Pin im Ruhezustand High-Potenzial anliegt und erst bei Betätigung, also bei Aktivität, Low-Potenzial, spricht man hier von „Active Low“. Umgekehrt sollten Sie einen Pull-Down-Widerstand gegen Masse immer dann benutzen, wenn Sie einen Eingang etwa auf Tastendruck mit VCC verbinden wollen (Active High). Dann hat der Eingang im Ruhezustand Masse-Potenzial.

Bei den meisten Pins können Sie wählen, ob Sie mit Active-Low- oder Active-High-Logik arbeiten. Die Ausnahmen sind die Pins 3 und 5, bei denen ein 1,8-kΩ-Pull-Up-Widerstand fest verdrahtet ist, sodass diese Pins auch nach dem Herunterfahren des Raspi ein definiertes Potenzial haben. Sie können das ausnutzen, um an Pin 5 und 6 einen Power-Taster anzuschließen. Der schaltet Ihren Raspi wieder ein, wenn Sie ihn zuvor mit dem Befehl `poweroff` heruntergefahren hatten, ohne dass Sie die Stromzufuhr unterbrechen müssen.

### Im Zweifel Grün wählen

Ob und welche Sonderfunktion ein Pin hat, erkennen Sie an der Farbe des Anschlusses in der Tabelle: Grün sind alle Ein-/Ausgänge ohne Zusatzfunktionen, die blauen können als Takt- oder Modulationsausgang genutzt werden, orange sind serielle Schnittstellen, gelb die Pins der I2C-Busse (Inter-Integrated Circuit) und die violetten Pins gehören zu den SPI-Bussen (Serial Peripheral Interface). Wenn Sie auf der Suche nach einem Pin für eine Status-LED oder einen Taster sind, sollten Sie einen der grünen Anschlüsse bevorzugen.

Zwar können Sie eine solche Status-LED (mit dem zugehörigen Vorwiderstand) auch an Pin 3 und den Taster an Pin 5 anschließen, doch damit ist dann der I2C-Bus, für den die Pins 3 und 5 ebenfalls zuständig sind, belegt. Sollten Sie im Verlauf Ihres Projekts feststellen, dass Sie

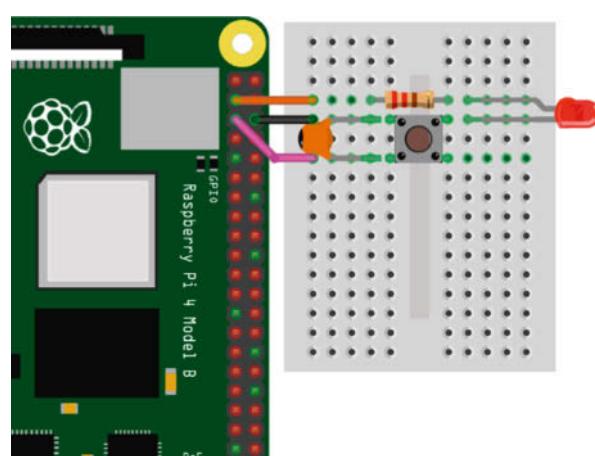
etwa für eine 7-Segment-Anzeige noch einen I2C-Bus benötigen, müssen Sie erst für LED und Taster neue Pins finden. Es gibt aber auch Ausnahmen, die es rechtfertigen, Pins mit Zusatzfunktionen für banale Schaltaufgaben zu verwenden: So können Sie nur Pin 5 zum Wiedereinschalten des Raspi verwenden, Sie müssen sich also zwischen einem Power-Taster und dem I2C-Bus entscheiden.

Wenn Sie Pin 5 für einen Power-Taster benutzen, ist der I2C-Bus damit ohnehin blockiert, können Sie auch Pin 3 für eine LED verwenden. Allerdings dürfen Sie sie nicht unmittelbar anschließen – als Halbleiter würde sie sofort durchbrennen. Gängige LEDs verkraften nur geringe Ströme von maximal 20 mA. Wie viel genau, müssen Sie im Katalog eines Teilelieferanten oder im Datenblatt der LED nachschlagen. Typischerweise reicht ein Betriebsstrom von 10 mA aus, damit sie hell leuchtet. Indem Sie gemäß dem Ohmschen Gesetz die Ausgangsspannung von 3,3 Volt durch den maximalen Strom von 20 mA teilen, finden Sie heraus, wie groß ein Vorwiderstand sein muss, damit die LED keinesfalls durchbrennen kann: 165 Ω. Mit weniger als 10 mA sollten Sie die LED aber auch nicht betreiben, sodass sich als oberer Widerstandswert 330 Ω ergeben.

Diese Überschlagsrechnung berücksichtigt nicht die Durchflussspannung der LED, die je nach Halbleitermaterial und Farbe 1,6 bis 2,2 Volt betragen kann – auch dieses Detail finden Sie im jeweiligen Datenblatt. Bei einer roten LED mit 10 mA Betriebsstrom und 1,6 V Durchflussspannung ergibt sich ein Spannungsabfall von 1,7 V am Vorwiderstand, der den Strom auf 10 mA begrenzen muss – was einen Widerstandswert von 170 Ω erfordert.

Ein Blick in die gängige Normreihe E12 zeigt, dass es keinen Widerstand mit

**Ohne den Vorwiderstand würde die LED an Pin 3 des Raspi durchbrennen. Wird Pin 5 als Eingang genutzt, zieht ihn der fest verdrahtete Pull-Up-Widerstand auf High – bis Sie die Taste drücken und den Eingang auf Masse ziehen.**



## Pin-Belegung des Raspi-GPIO-Ports

Sonderfunktion	Bezeichnung	Pin-Nr.	Bezeichnung	Sonderfunktion
	+3,3 V (VCC)	1	2	+5 V (VIN)
I2C 1, SDA (I <sup>2</sup> C 1, Daten)	GPIO2	3	4	+5 V (VIN)
I2C 1, SCL (I <sup>2</sup> C 1, Takt)	GPIO3 (Power on)	5	6	Masse (GND)
GPCLK 0 (Taktgenerator)	GPIO4	7	8	GPIO14 UART 0, TXD (seriell, senden)
	Masse (GND)	9	10	GPIO15 UART 0, RXD (seriell, empfangen)
	GPIO17	11	12	GPIO18 PWM 0 (Modulator)
	GPIO27	13	14	Masse (GND)
	GPIO22	15	16	GPIO23
	+3,3 V (VCC)	17	18	GPIO24
SPI 0, MOSI	GPIO10	19	20	Masse (GND)
SPI 0, MISO	GPIO9	21	22	GPIO25
SPI 0, SCLK (Takt)	GPIO11	23	24	GPIO8 SPI 0, CE0 (Chip Select 0)
	Masse (GND)	25	26	GPIO7 SPI 0, CE1 (Chip Select 1)
I2C 0, SDA (I <sup>2</sup> C 0, Daten)	GPIO0	27	28	GPIO1 I2C 0, SCL (I <sup>2</sup> C 0, Takt)
GPCLK 1 (Taktgenerator)	GPIO5	29	30	Masse (GND)
GPCLK 2 (Taktgenerator)	GPIO6	31	32	GPIO12
PWM 1 (Modulator)	GPIO13	33	34	Masse (GND)
SPI 1, MISO	GPIO19	35	36	GPIO16 SPI 1, CE2 (Chip Select 2)
	GPIO26	37	38	GPIO20 SPI 1, MOSI
Masse (GND)	39	40	GPIO21	SPI 1, SCLK (Takt)

165 oder 170  $\Omega$  gibt, sondern lediglich die Werte 150  $\Omega$  und 180  $\Omega$ . 150  $\Omega$  wären zu wenig, sodass Sie mindestens zu 180  $\Omega$  greifen müssen. Der obere Widerstandswert mit 330  $\Omega$  steht in der Tabelle. Dazwischen liegen 220  $\Omega$ , als Vorwiderstand für LEDs ein guter Kompromiss. Der im Raspi fest verdrahtete Pull-Up-Widerstand von 1,8 k $\Omega$  fällt nicht ins Gewicht.

### Es werde Licht

Ist die LED verdrahtet, müssen Sie sie ansteuern. Dafür gibt es verschiedene Wege. Möchten Sie etwa die LED einschalten, während ein Backup-Skript einer Datenbank läuft, empfehlen wir, den Pi-GPIO-Dienst nachzuinstallieren und zu aktivieren. Hier die Kommandos für das Terminal:

```
sudo apt-get install pigpio
sudo systemctl enable pigpiod
sudo systemctl start pigpiod
```

Nun können Sie den Pin mit folgenden Befehlen konfigurieren und ansteuern:

```
pigs modes 2 w
pigs write 2 1
pigs write 2 0
```

Der erste Befehl macht den Anschluss 2 zum Ausgang, der danach beschreibbar ist (w für writable). Die beiden anderen Befehle setzen den Wert 1 respektive 0 bei Anschluss 2 und schalten damit die LED an Pin 3 erst ein und dann wieder aus.

Wenn in diesem Artikel etwa von Pin 3 die Rede ist, dann ist damit der dritte Anschlusspin des GPIO-Ports auf der Platinne gemeint. Aufgrund der Leiterbahnführung ist die Pin-Nummer jedoch nicht die gleiche wie die Nummer des Ausgangs am Raspi-SoC (System on Chip). So ist Pin 3 dessen Ausgang GPIO2 und Pin 13 der

Anschluss GPIO27. Es gibt also zwei verschiedene Nomenklaturen beim Raspi, die Pin-Nummern auf dem Board und die Ausgänge des SoC. Das führt oft zu Verwechslungen, weil man feststellen muss, welche Benennung ein Projekt, ein Programm oder auch eine Bibliothek verwendet. Bei Pi-GPIO müssen Sie die Anschlussnummer des Broadcom-SoC angeben, also 2 für Pin 3. Welcher Pin welchem GPIO-Anschluss entspricht, finden Sie in der Pin-Belegung.

Soll die LED blinken, wird es kompliziert: Die denkbar schlechteste Lösung ist, die beiden pigs-Kommandos in einer Schleife immer wieder aufzurufen und dazwischen mittels sleep eine Pause einzulegen. Das führt zu einer enormen Auslastung des Raspi, der quasi nur noch damit beschäftigt ist, pigs aufzurufen und zu beenden, um die LED ein- und auszuschalten. Stattdessen können Sie den PWM (Pulsweitenmodulator) an Pin 12 (GPIO18) dafür benutzen, die LED blinken zu lassen:

```
pigs hp 18 1 500000
```

Der Parameter hp configuriert die Hardware-PWM des Raspi, dahinter folgen der GPIO-Anschluss (hier 18 für den Boardpin 12), die Frequenz in ganzzahligen Hertz, mit der die LED blinken soll, und der sogenannte Duty Cycle. Der Duty Cycle liegt zwischen Null und einer Million und legt fest, wie lang die LED pro Zyklus eingeschaltet ist. Ein Duty Cycle von 500.000 bedeutet also, dass die LED 50 Prozent der Zeit leuchtet. Je kürzer Sie den Duty Cycle wählen, desto mehr wird aus dem Blinken ein Blitzen der LED.

Alternativ zu Pin 12 können Sie auch den zweiten PWM-Ausgang (GPIO13) an Pin 33 benutzen. In der Pin-Belegungstabelle sind alle Taktgeneratoren und Modulatoren hellblau gekennzeichnet. Die beiden PWM können Sie auch dazu benutzen, die Helligkeit einer LED zu steuern: Ab einer Frequenz von etwa 30 Hertz erkennt das menschliche Auge kein Blinken mehr. Dann erscheint die LED um so heller, je höher der Duty Cycle eingestellt ist. Eine Frequenz von 0 Hz schaltet den PWM ab.

### Taster

Theoretisch könnten Sie auch pigs verwenden, um festzustellen, welche Stellung ein an den Raspi angeschlossener Schalter hat. So liefert Ihnen pigs read 3 den Wert 1,

wenn ein dort angeschlossener Schalter geöffnet ist, und den Wert 0, wenn Sie ihn betätigten haben. Für einen Taster eignet sich das Verfahren schon nicht mehr, denn Sie müssten `pigs` mehrfach pro Sekunde aufrufen, um mitzubekommen, wenn jemand kurz den Taster drückt. Die bei Pi-GPIO dafür vorgesehene Lösung ist, den Daemon mit der Überwachung des Pins zu beauftragen und ein Skript zu registrieren, das er bei einer Pegeländerung aufruft – ziemlich kompliziert.

Wir empfehlen, den GPIO-Anschluss nicht über Skripte, sondern über Python anzusprechen. Das ist nicht nur eine elegante Lösung, um einen Tastendruck zu erkennen, sondern eröffnet Ihnen den Zugriff auf unzählige Bibliotheken, mit denen Sie Erweiterungen und Sensoren direkt ansprechen und mit wenigen Befehlen nutzen können.

Python ist ein Interpreter, den Sie in Skripten auf die gleiche Weise einbinden wie eine herkömmliche Shell. Setzen Sie:

```
#!/usr/bin/python3
```

in die erste Zeile. Anschließend binden Sie die Bibliotheken ein, die Sie im Skript nutzen wollen. Um auf die GPIO-Pins zuzugreifen, benötigen Sie die Funktionen aus der Bibliothek RPi:

```
from RPi import GPIO
```

Um lediglich einen Taster anzusprechen, benötigen Sie keine weiteren Funktionen und können direkt loslegen. Der obligatorisch erste Befehl lautet:

```
GPIO.setmode(GPIO.BCM)
```

## Nummern-Wirrwarr

So legen Sie fest, nach welchem Namensschema Sie nachfolgend die GPIO-Anschlüsse ansprechen: Mit `GPIO.BCM` verwenden Sie die Nummerierung der GPIO-Ausgänge am Broadcom-SoC, genau wie bei `pigs` zuvor. Bei `GPIO.BOARD` hingegen können Sie direkt die Pin-Nummern des GPIO-Anschlusses angeben, was den Umweg über die Pin-Belegungstabelle erspart.

Um Pin 5 alias GPIO3 als Eingang zu konfigurieren, nutzen Sie folgenden Befehl:

```
GPIO.setup(3, GPIO.IN)
```

Haben Sie zuvor mit `pigs` gearbeitet, erhalten Sie nun eine Warnung, dass dieser

Pin bereits verwendet wird. Diese dürfen Sie ignorieren. Wie Sie sie abschalten können, verrät sie selbst:

```
GPIO.setwarnings(False)
```

GPIO3 an Pin 5 wird, wie zuvor erwähnt, bereits durch einen fest verdrahteten Pull-Up-Widerstand auf High-Potenzial gezogen. Deshalb liefert die Methode `GPIO.input(3)` im unbeschalteten Zustand den Wert 1:

```
print(GPIO.input(3))
```

Wenn Sie den Taster gedrückt halten und damit den Eingang auf Masse ziehen, stellt sich der Wert 0 ein. In einer Schleife immer wieder den Wert des Eingangs abzufragen, sogenanntes Polling, würde aber auch in Python den Raspi überlasten. Die GPIO-Bibliothek bietet dafür einen Event-Handler:

```
GPIO.add_event_detect(3, GPIO.BOTH, ↴
callback=buttonPress)
```

```
#!/usr/bin/python3
from RPi import GPIO
import time
from subprocess import call

def buttonPress(pin):
    if not GPIO.input(3):
        call(['sudo', 'shutdown', ↴
              '-h', 'now'])

GPIO.setmode(GPIO.BCM)
GPIO.setup(3, GPIO.IN)
GPIO.add_event_detect(3, GPIO.BOTH, ↴
callback=buttonPress)
while True:
    time.sleep(10)
```

**Mit PyShutdown fahren Sie Ihren Raspi auf Knopfdruck herunter und auch wieder hoch. Dazu müssen Sie lediglich einen Taster an die Pins 5 und 6 anschließen.**

Damit beauftragen Sie den Event-Handler, GPIO3 alias Pin 5 zu beobachten und bei jeder Veränderung des Zustands (`GPIO.BOTH`) die Funktion `buttonPress()` aufzuru-

## GPIO-Bibliotheken im Umbruch

Traditionell stand der GPIO-Anschluss jedem Benutzer des Raspi uneingeschränkt zur Verfügung. Bildlich gesprochen konnte niemand Pin 3 für sich beanspruchen, um allein die dort angeschlossene LED ein- oder auszuschalten. Für ein Mehrbenutzer-Betriebssystem ist es prinzipiell unerwünscht, dass Benutzer auf fremde Ressourcen zugreifen können, weshalb mit `gpiochip` eine Zugriffsverwaltung eingeführt wurde: Damit können einzelne Prozesse oder Programme einen Pin oder eine Schnittstelle zur exklusiven Nutzung anfragen und reservieren. Der Kernel achtet dann darauf, dass nicht zwei Prozesse auf dem gleichen Pin arbeiten, und räumt auch hinter den Programmen auf, falls sie enden, ohne den Pin vorher wieder freizugeben. Eine Emulationschicht gewährleistet, dass der bisherige ungeschützte Zugriff weiterhin funktioniert.

Mit Raspberry Pi OS vom März 2024 und durch Paket-Updates wurde die Bookworm-Variante auf Kernel 6.6 aktualisiert und die bisherige Kompatibilitätschicht entfernt. Das fiel oft nicht auf, weil die bisherige Python-Bibliothek `rpi-gpio`

größtenteils weiterhin funktionierte; Ausgänge ließen sich unverändert ansteuern und Eingänge auslesen. Anders ist es beim Event-Manager, der zum Beispiel einen Eingang auf einen Tastendruck hin überwacht. Bei Redaktionsschluss dieses Beitrags war die `rpi-gpio`-Bibliothek noch nicht vollständig aktualisiert, weshalb Sie mit folgendem Befehl ersetztweise die Bibliothek `rpi-lgpio` installieren müssen:

```
sudo apt install --auto-remove ↴
--purge python3-rpi-lgpio
```

Die Parameter `--autoremove` und `--purge` entfernen die konkurrierenden Pakete `python3-rpi.gpio` und `rpi.gpio-common` vollständig, damit es keinen Konflikt gibt. Dieser Schritt ist aber nur bei Raspberry Pi OS erforderlich, das auf Debian Bookworm basiert, und auch nur so lange, wie der Event-Handler Fehler meldet. Ältere Ausgaben, die noch auf Debian Bullseye beruhen, enthalten nach wie vor die alte Kompatibilitätschicht. Welches Debian Ihrem installierten System zugrunde liegt, enthüllt ein Blick in die Datei `/etc/os-release`.

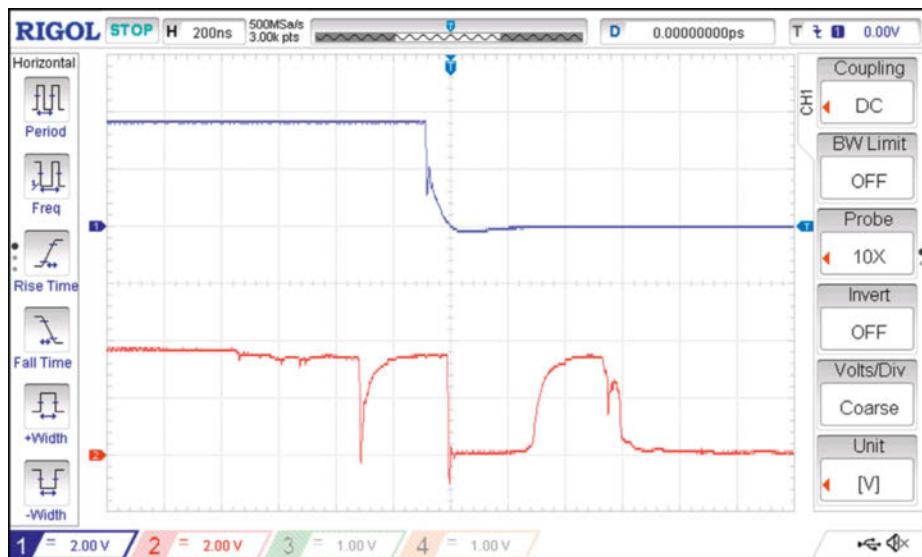
fen. Und die soll den Raspi herunterfahren:

```
def buttonPress(pin):
    if not GPIO.input(3):
        call(['sudo', 'shutdown', '-h', 'now'])
```

Die Funktion überprüft, dass der Taster an GPIO3 (Pin 5) gedrückt ist, bevor sie mit der Funktion `call()` aus der Bibliothek `subprocess` den Shell-Befehl `sudo shutdown -h now` ausführt. Damit wird der an Pin 5 angeschlossene Taster zum Ein-/Ausschalter des Raspi. Den vollständigen Python-Code finden Sie im Listing PyShutdown, das fertige Skript auf [ct.de/yk1f](#) zum Download.

Überprüfen Sie zunächst, ob das Programm läuft, indem Sie es zunächst mit dem Befehl `chmod a+x PyShutdown` ausführbar machen und dann mit `./PyShutdown` starten. Sollte dies zu einer Fehlermeldung `RuntimError: Failed to add edge detection` führen, müssen Sie Ihr Raspberry Pi OS auf eine andere GPIO-Bibliothek umstellen. Den Befehl dazu und mehr Informationen zum Hintergrund der Umstellung finden Sie im Kasten „GPIO-Bibliotheken im Umbruch“.

Läuft das Programm einwandfrei, können Sie den Aufruf in der vorletzten



Ein kleiner Kondensator, parallel zum Taster geschaltet, lässt das sonst unvermeidliche Tastenprellen (rot, unten) fast vollständig verschwinden (blau, oben).

Zeile der Datei `/etc/rc.local` eintragen, damit es künftig automatisch beim Booten des Raspi gestartet wird:

```
/home/pi/PyShutdown &
```

### Taster entprellen

Während es bei Pin 5 alias GPIO3 einen fest verdrahteten Pull-Up-Widerstand gibt, können Sie bei anderen Pins während der Initialisierung auswählen, ob Sie dort einen Pull-Up-, einen Pull-Down- oder gar keinen Widerstand haben wollen. Die nachfolgende Zeile wählt für GPIO17 alias Pin 11 einen Pull-Up-Widerstand:

```
GPIO.setup(17, GPIO.IN, pull_up_down=GPIO.PUD_UP)
```

Das vereinfacht die Schaltung, weil Sie nun lediglich noch einen Taster an den Eingang anschließen und ihn gegen Masse kurzschließen müssen.

Der Event-Handler verdient noch eine genauere Betrachtung: Bei PyShutdown reagiert GPIO3 durch die Angabe von `GPIO.BOTH` auf jegliche Veränderung. Die Funktion `buttonPress()` wird sowohl dann aufgerufen, wenn Sie die Taste drücken und dadurch die Spannung am Eingang von 3,3 V auf 0 V fällt (fallende Signallanke), als auch dann, wenn Sie die Taste wieder loslassen und die Spannung von 0 V wieder auf 3,3 V ansteigt (steigende Signallanke).

Das ist ein Schwachpunkt des Programms, denn durch sogenanntes Tasten-

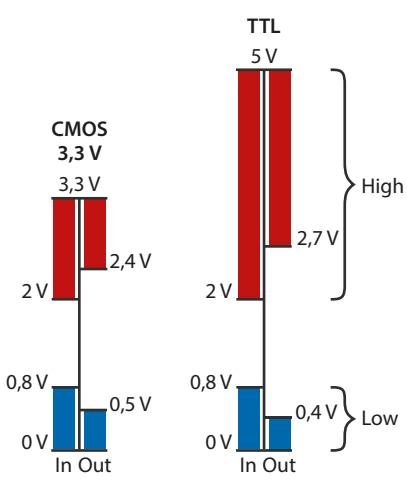
prellen kann es passieren, dass die Funktion `buttonPress()` mehrfach kurz hintereinander aufgerufen wird. Bei jedem mechanischen Taster und auch Schalter kommt der Kontakt nicht schlagartig zu stande, sondern die Kontaktflächen reiben zunächst übereinander. Verschleiß, Abbrand und Oxidation bewirken, dass in einem Moment ein elektrischer Kontakt da war, der im nächsten Augenblick wieder unterbrochen ist, bis schließlich die Kontakte vollflächig aufeinander liegen und eine stabile elektrische Verbindung herstellen.

Im Oszillosgramm oben sehen Sie auf dem zweiten Kanal (rot, unten), wie der Pegel am Eingang GPIO3 innerhalb von 1,5 Millisekunden mehrfach zwischen 3,3 V und 0 V hin und her springt, bevor sich der stabile Zustand 0 V einstellt. Der Event-Handler des Raspi ist schnell genug, um mehrere dieser Signallanken zu bemerken. Das Tastenprellen können Sie beseitigen, indem Sie parallel zum Taster einen kleinen Kondensator schalten. Der Spannungsverlauf auf Kanal 1 (blau, oben) zeigt den Spannungsverlauf desselben Tasters mit parallel geschaltetem 100-nF-Keramikkondensator.

Bei der Dimensionierung des Kondensators müssen Sie im Blick behalten, dass die Signallanke nicht zu stark abgerundet, also verschliffen werden darf. Idealerweise wäre das Schaltsignal ein perfektes Rechteck, doch der blaue Spannungsverlauf lässt bereits eine deutliche Verrundung nahe 0 V erkennen. Das ist ein Hinweis darauf, dass der Kondensator

## Logikpegel

Viele Komponenten und Module etwa aus Arduino-Projekten arbeiten mit TTL-Pegeln. Der Raspberry Pi verträgt allerdings nur CMOS-Logikpegel bis maximal 3,3 Volt, weshalb an Eingängen ein Spannungsteiler erforderlich ist. Die Spannung an den Ausgängen des Raspi hingegen genügt, um auch TTL-Komponenten direkt anzusteuern.



im Bezug auf den Pull-Up-Widerstand nicht noch größer gewählt werden sollte. Ein 47-nF-Kondensator würde bei einem Pull-Up-Widerstand von 1,8 kΩ eine steilere Signalflanke erzeugen. Je höher der Widerstandswert ist, desto kleiner müssen Sie den Kondensator wählen, um die Signalfanken nicht zu stark zu verschleifen. Andererseits darf der Kondensator auch nicht zu klein geraten, damit er genug Kapazität hat, um die Spannungswechsel beim Prellen abzufangen.

## In Software geflickt

Prinzipiell können Sie das Tastenprellen auch softwareseitig verhindern, indem Sie den zusätzlichen Parameter `bouncetime` nutzen:

```
GPIO.add_event_detect(3, GPIO.BOTH, ↴
    callback=buttonPress, bouncetime=200)
```

Dadurch ignoriert der Event-Handler mehrfache Pegelwechsel innerhalb von 200 Millisekunden. Für eine Tastatur wäre das ein zu hoher Wert, da man die Taste dann höchstens noch 5 Mal pro Sekunde betätigen kann. Für einen Auschalter ist das hingegen ausreichend. Wir empfehlen Ihnen aber, auf den Parameter `bouncetime` zu verzichten und das Problem in Hardware durch einen passenden Kondensator zu lösen, anstatt ein schlechtes Schaltungsdesign in Software zu reparieren und damit den Prozessor zu belasten.

Was Sie jedoch verbessern können ist, dass der Event-Handler auf fallende und steigende Signalflanken reagiert. Da an GPIO3 im Ruhezustand per Pull-Up-Widerstand 3,3 V anliegen, ist es besser, wenn der Event-Handler nur die fallende Flanke berücksichtigt:

```
GPIO.add_event_detect(3, GPIO.FALLING, ↴
    callback=buttonPress)
```

## LEDs in Python

In Python einen Pin als Ausgang zu konfigurieren, ist übrigens ähnlich einfach wie bei einem Eingang:

```
GPIO.setup(2, GPIO.OUT, ↴
    initial=GPIO.LOW)
```

Damit wird Pin 3 alias GPIO2 zum Ausgang für eine LED. Durch die Angabe von `initial=GPIO.LOW` schalten Sie den Ausgang nach der Initialisierung auf Massepotential

zial und damit die LED ab. Um die LED einzuschalten, müssen Sie den Pin auf High setzen:

```
GPIO.output(2, GPIO.HIGH)
```

Ähnlich wie mit pigs können Sie den Pulsweitenmodulator an Pin 12 alias GPIO18 nutzen, um die LED blinken zu lassen:

```
GPIO.setup(18, GPIO.OUT)
led = GPIO.PWM(18, 0.5)
led.start(10.0)
```

`GPIO.PWM()` initialisiert den PWM hier mit 0,5 Hertz. Anschließend benutzen Sie die Methode `start()` des zurückgelieferten Objekts `led`, um den PWM zu starten. Als Parameter übergeben Sie, wie lange die LED eingeschaltet sein soll (Duty Cycle). Der Wertebereich beginnt hier bei 0.0 und endet bei 100.0. Mit `stop()` schalten Sie den PWM später wieder aus. Wollen Sie nach der Initialisierung die Blinkfrequenz oder den Duty Cycle anpassen, erledigen das die Methoden `ChangeFrequency()` und `ChangeDutyCycle()`.

## Eingepegelt

Wenn Sie Elektronikkomponenten für Raspi-Basteleien kaufen, sollten Sie auf die Versorgungsspannung und vor allem die Signalpegel achten. Prinzipiell sind Komponenten wie zum Beispiel das Ultraschallmodul HC-SR04 zur Entfernungsmeßung unkritisch; so lange sie mit 3,3

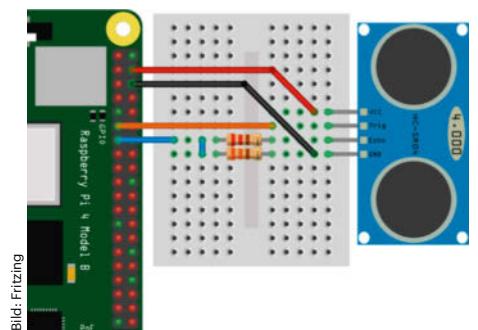
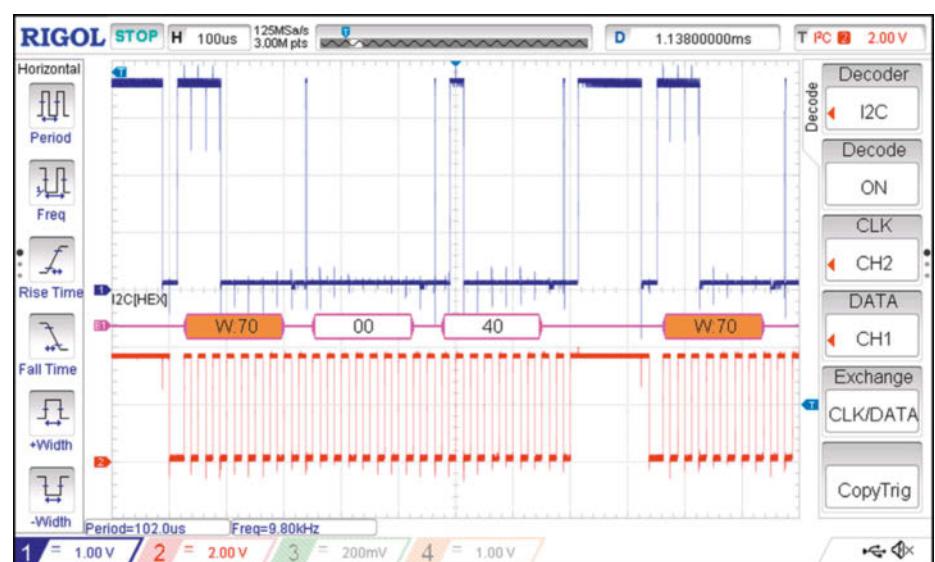


Bild: Fritzing  
Dank überlappender Signalbereiche von TTL- und CMOS-Pegeln läuft das Ultraschallmodul HC-SR04 am Raspi ohne Anpassungen, obwohl es für 5-Volt-Signale des Arduino gedacht ist. Nur seinen Ausgangspegel müssen Sie dem Raspi mittels Spannungsteiler verdaulich machen.

Volt betrieben werden, verwenden sie auch CMOS-kompatible Signalpegel. Viele Komponenten brauchen jedoch 5 Volt Betriebsspannung, vor allem solche für Arduino und andere Mikrocontroller. Hier müssen Sie unbedingt in den Datenblättern nachschlagen, wie hoch die Signalpegel von Ausgängen und Datenleitungen sind. Üblich sind hier 3,3 Volt CMOS, womit der Raspi problemlos zuretkommt, bei Arduino & Co. aber vor allem TTL (Transistor to Transistor Logic) mit 5 Volt.

TTL-Pegel an den Ausgängen einer Komponente müssen Sie unbedingt mit einem Spannungsteiler auf maximal 3,3



Der serielle I2C-Bus spricht alle Komponenten über verschiedene Bus-IDs (orange) an, die meist per Lötbrücke gesetzt werden können. So ist bei jedem Datenpaket klar, wer gemeint ist: Hier bekommt die 7-Segment-Anzeige mit der ID 70h zwei Datenbytes übertragen.

Volt anpassen, damit der Eingang des Raspi überlebt. Ideal wäre ein Widerstandsverhältnis von 1:2, doch das ist mit zwei Widerständen aus der gängigen Widerstandsreihe E12 schwierig. Geeignet wäre ein Spannungsteiler aus den Widerständen  $1,2\text{ k}\Omega$  und  $2,2\text{ k}\Omega$ , womit sich bei 5 Volt Signalpegel ein Strom von maximal 1,5 mA einstellt und die Spannung am Raspi-Eingang auf 3,25 Volt begrenzt wird.

Allerdings haben die Widerstände Toleranzen, bei den gängigsten im Handel üblichen sind bis zu 10 Prozent erlaubt, womit die Eingangsspannung schlimmstenfalls auf fast 3,5 Volt steigen kann – zu viel für den Raspi. Deshalb empfehlen wir, den Spannungsteiler mit den Widerständen  $2,2\text{ k}\Omega$  und  $3,3\text{ k}\Omega$  zu bilden. Damit reduziert sich auch die Belastung der Signalquelle auf unter 1 mA und es stellt sich eine Eingangsspannung von typisch 3 Volt und im Extremfall von maximal 3,2 Volt ein.

3 Volt genügen dem Raspi für ein High. Das liegt an den für CMOS und TTL genormten Toleranzbereichen, die sowohl Signalverluste über Kabel als auch etwaige Störspannungen berücksichtigen. Die Abbildung auf Seite 122 unten links zeigt, welche Spannungen bei CMOS und TTL für High- und Low-Signalpegel zulässig sind. Während sich die Spannungsgrenzen an den Ausgängen um wenige Millivolt unterscheiden, sind sie für die Eingänge bei CMOS und TTL identisch: Ein High-Signal hat mindestens einen Pegel von 2 Volt und ein Low-Signal maximal 0,8 Volt.

Der Clou daran: Da auch eine TTL-Komponente, die eigentlich für die 5-Volt-Pegel des Arduino vorgesehen ist, schon 2 Volt als High-Signal akzeptiert, benötigen

Sie an den Eingängen solcher Komponenten keine Verstärkerschaltungen oder Pegelwandler. Die Komponente ist mit den 3,3-Volt-Pegeln, die die Ausgänge des Raspi bereitstellen, vollauf zufrieden.

Für das Ultraschallmodul HC-SR04 bedeutet dies, dass Sie lediglich den Modulausgang mit einem Spannungsteiler an den Raspi anpassen müssen. Den Eingang können Sie direkt mit dessen GPIO-Anschluss verbinden; die Versorgungsspannung dürfen Sie an den Pins 1 oder 2 des GPIO-Anschlusses abgreifen. Die Schaltung auf Seite 123 oben zeigt die Verdrahtung des Ultraschallmoduls mit dem Raspi sowie den notwendigen Spannungsteiler mit den Widerständen  $2,2\text{ k}\Omega$  und  $3,3\text{ k}\Omega$ .

## Angebunden

Während das Ultraschallmodul mit zwei als Ein- und Ausgang konfigurierten Pins des Raspi angesprochen wird, verwenden sehr viele andere Erweiterungsplatinen den I2C-Bus (Inter-Integrated Circuit). Davon besitzt der Raspi gleich zwei: I2C-0 an den Pins 27 und 28, der jedoch dem Videoprozessor vorbehalten ist, sowie I2C-1 an den Pins 3 und 5.

Auf einem frisch installierten Raspberry Pi OS ist I2C standardmäßig abgeschaltet. Wenn Sie den Bus nutzen wollen, müssen Sie ihn zunächst aktivieren. Dazu rufen Sie im Terminal den Befehl `sudo raspi-config` auf und wählen unter „Interface Options“ den Punkt „I2C“. Sobald Sie das Konfigurationsprogramm mit „Finish“ beendet haben, können Sie den I2C-Bus nutzen. Ein Neustart ist nicht erforderlich.

Bei I2C handelt es sich um einen seriellen Datenbus mit Datenleitung (SDA, Se-

rial Data), Taktsignal (SCK, Serial Clock) und Masseverbindung (GND, Ground), an den Sie theoretisch über 100 Module parallel anschließen können. Dabei erhält jedes Modul eine eindeutige ID, die oft über Anschlusspins oder Lötbrücken festgelegt wird. Für die Adressierung stehen 7 Bit zur Verfügung, wobei allerdings einige IDs reserviert sind. Der Raspi kann an dem freien I2C-Bus an den Pins 3 und 5 eine Datenrate von bis zu 3,4 Mbit/s schaffen (High Speed Mode), voreingestellt sind aber lediglich 100 kbit/s (Standard Mode). Für OLED-Displays können Sie mit 400 kbit/s (Fast Mode) aber unter Umständen einen flüssigeren Refresh des Bildschirm Inhalts erreichen. Den Fast Mode unterstützen die meisten Komponenten problemlos.

Um den Fast Mode zu aktivieren, müssen Sie mit Root-Rechten den Eintrag in der Datei `/boot/firmware/config.txt` anpassen, der den I2C-Bus freischaltet. Dazu ändern Sie die Zeile

```
dtparam=i2c_arm=on
```

in:

```
dtparam=i2c_arm=on,i2c_arm_baudrate=400000
```

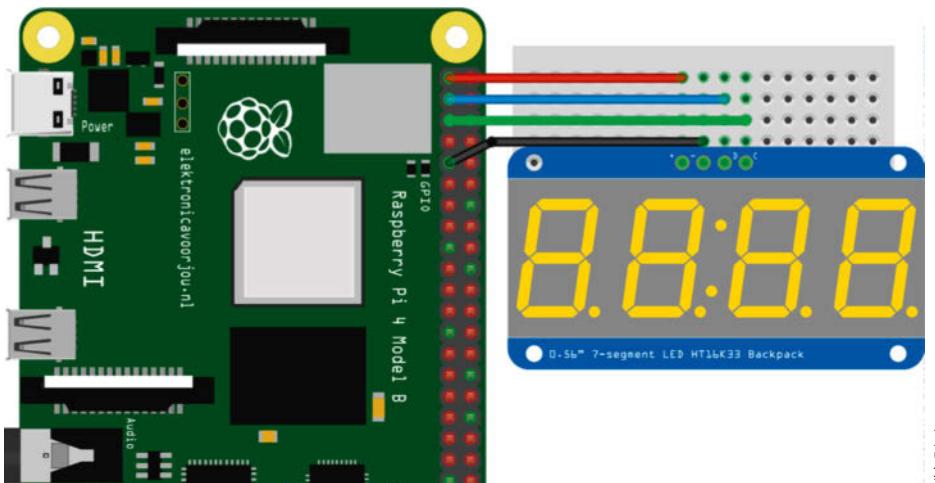
Anschließend will der Raspi einmal neu gestartet werden. Mit welcher Geschwindigkeit der I2C-Bus gerade arbeitet, deckt der Befehl

```
hexdump -C /sys/class/i2c-adapter/1_i2c-1/of_node/clock-frequency
```

auf. Das hexadezimale Ergebnis ist die Übertragungsgeschwindigkeit in bit/s.

## 7 Segmente in Python

Nicht nur Schrittmotor- und Servo-Controller nutzen den I2C-Bus, sondern auch Displays und LED-Anzeigen. Als Beispiel soll eine vierstellige 7-Segment-Anzeige mit dem verbreiteten LED-Controller Holtek HT16K33 dienen. Module mit diesem Controller sind üblicherweise ab Werk auf die I2C-ID 70h konfiguriert, durch Lötbrücken lassen sich die letzten drei Bits der Adresse ändern. Angeschlossen an den Raspi wird das Modul über vier Pins: VCC für die Betriebsspannung kommt an Pin 1 des Raspi, SDA und SCL werden mit den Pins 3 und 5 des I2C-Bus verbunden und GND mit einem der Masseanschlüsse, etwa Pin 6 oder 9. Die Abbildung links zeigt die Verschaltung.



Über den I2C-Bus des Raspis können Sie Schrittmotoren, Servo-Controller oder wie hier ein LED-Display anschließen. Die Programmierung in Python erfordert durch die zahlreichen Bibliotheken nur wenige Zeilen.

Ob der Raspi den Display-Controller korrekt erkennt, finden Sie heraus, indem Sie den Befehl `i2cdetect -y 1` auf dem Bus I2C-1 alle möglichen IDs abklopfen lassen. Ist die 7-Segment-Anzeige korrekt angeschlossen, taucht dort die ID 70 auf.

Für die Ansteuerung gibt es zwar eine Python-Bibliothek von Adafruit, diese ist aber sehr komplex und für viele Boards des Herstellers gedacht. Kompakter ist die Bibliothek von Eric McConville, die Sie bei GitHub herunterladen können:

```
git clone https://github.com/ericmcconville/HT16K33.git
```

Außerdem benötigen Sie die Python-Bibliothek SMBus aus dem Repository des Raspberry Pi OS:

```
sudo apt-get install python3-smbus
```

Damit können Sie die 7-Segment-LED-Anzeige initialisieren und darauf die Zahl 1234 ausgeben:

```
#!/usr/bin/python3
from HT16K33 import FourDigit
display = FourDigit(bus=1).setUp()
display.clear()
display.writeDigit(0, 1)
display.writeDigit(1, 2)
display.writeDigit(2, 3)
display.writeDigit(3, 4)
```

Weitere IC2-Komponenten verkabeln Sie einfach parallel zum LED-Modul.

## Ausgebaut

Ein zweites wichtiges Bussystem des Raspberry Pi ist der SPI-Bus (Serial Peripheral Interface), davor allem zum Anschluss von TFT-Displays mit höheren Auflösungen, aber auch für RFID- und NFC-Module gebräuchlich ist. Während bei I2C Raspi und I2C-Modul nur abwechselnd über dieselbe Leitung und alle Busteilnehmer mit derselben Geschwindigkeit Daten übertragen können, handelt es sich bei SPI um einen vollduplexfähigen Bus, der noch dazu jeden Busteilnehmer mit individueller Geschwindigkeit ansteuert. Dazu gibt es eine Datenleitung vom Raspi zum SPI-Modul, genannt MOSI (Master Output Slave Input) und eine zweite für den Rückweg, genannt MISO (Master Input Slave Output). Außerdem erzeugt der Bus Master, also der Raspi, das Taktsignal (SCK oder SCLK, Serial Clock) und vom Raspi kommt auch das Bezugspotenzial, die Masseleitung (GND).

Werden mehrere SPI-Module am selben Bus betrieben, dann sind alle vier bisher genannten Leitungen parallel mit allen Modulen verbunden – MOSI, MISO, SCK und GND. Unterschiedliche IDs wie bei I2C gibt es bei SPI jedoch nicht. Stattdessen gibt es für jedes Modul noch eine individuelle Leitung, CE (Chip Enable) oder CS (Chip Select) genannt, die nur dann aktiviert wird, wenn dieses SPI-Modul gerade angesprochen wird.

Als Beispieldiagramm beschreiben wir nachfolgend ein RFID-Kartenmodul mit dem verbreiteten Controller-Chip MFRC522. Die Verschaltung mit dem Raspi ist rechts beschrieben: Vier Leitungen benötigen Sie für MOSI, MISO, SCK und CE, außerdem natürlich noch Masse und 3,3 Volt Versorgungsspannung für das RFID-Modul. Eine Besonderheit: Das Modul besitzt einen zusätzlichen Reset-Anschluss, über den das Modul initialisiert wird. Diesen verbinden Sie mit Pin 22 (GPIO25) des Raspi.

## Python liest RFID

Zum Programmieren des RFID-Moduls empfehlen wir abermals Python, denn hier gibt es für den RFID-Controller eine Bibliothek, die Sie mit folgendem Befehl auf dem Raspi installieren:

```
git clone https://github.com/1AdityaX/mfrc522-python.git
ln -s mfrc522-python/src/mfrc522
```

Durch den symbolischen Link, den der zweite Befehl anlegt, können Sie die Bibliothek unmittelbar aus dem aktuellen Verzeichnis heraus und ohne den Python-Paketmanager pip verwenden:

```
from mfrc522 import MFRC522
```

Das ist insofern elegant, weil Sie ab Raspberry Pi OS mit Bookworm-Wurzeln prinzipiell eine virtuelle Python-Umgebung einrichten müssten. Ein ziemlicher Overkill für Raspis, die meist nur für ein konkretes Bastelprojekt eingesetzt werden und wo Konflikte zwischen System- und externen Bibliotheken kaum eine Rolle spielen.

Das nachfolgende Python-Programm wartet darauf, dass Sie ein Token oder eine Karte auf das RFID-Modul legen und zeigt dann dessen Seriennummer an:

```
#!/usr/bin/python3
from mfrc522 import SimpleMFRC522
```

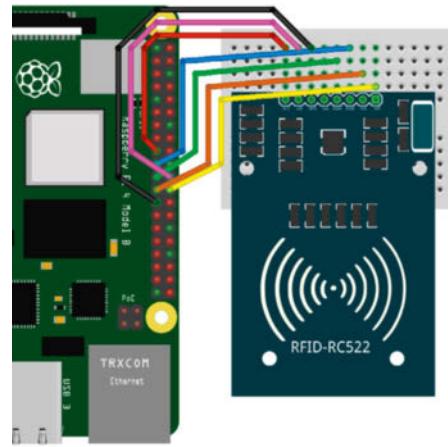


Bild: Fritzing

**Am SPI-Bus werden in erster Linie hochauflösende Displays, aber wie hier auch RFID- und NFC-Kartenleser angebunden. Eine Besonderheit des MFRC522 ist die pinkfarbene Reset-Leitung, über die der Controller initialisiert wird.**

```
reader = SimpleMFRC522()
id, text = reader.read()
print(id)
```

Zwar kann man die Seriennummer von Karten und Tags nicht ändern, aber es ist kein Sicherheitsmerkmal, das Sie etwa für die Anmeldung oder ein Schließsystem einsetzen dürfen: Sie lässt sich zum Beispiel mit dem Hacking-Werkzeug Flipper Zero spielend leicht kopieren. Die Python-Bibliothek besitzt aber auch Authentifizierungsfunktionen, mit denen Sie Daten, vor fremden Zugriffen durch einen Schlüssel geschützt, auf Karten und Tags speichern können. Nähere Informationen dazu finden Sie auf der Projektseite, die wir auf ct.de/yk1f verlinkt haben.

## Fazit

Mit etwas Wissen und Hardware lassen sich Sensoren und Aktoren wie Taster oder Leuchtdioden leicht an den Raspi anschließen und von ihm steuern. Mit komplexeren Modulen kommuniziert er über Busse wie I2C und SPI. Dafür ist Python die Programmiersprache der Wahl, denn dafür gibt es einen riesigen Fundus an Bibliotheken für praktisch alle auf dem Markt erhältlichen Erweiterungen. Dank derer reichen wenige Zeilen Python-Code, um Displays, Kartenleser oder Servos anzusteuern. Um einfach nur eine LED leuchten zu lassen, genügen schon Shell-Tools.

(mid@ct.de) ct

**Listings und Projekte:** [ct.de/yk1f](http://ct.de/yk1f)



# Raspi-Zentralverwalter

## Raspberry Pi aus der Ferne warten mit qbee

**Die Cloudplattform qbee überwacht und steuert IoT-Geräte. Wie gut sich deren Funktionsumfang für private Raspi-Bastler und gewerbliche IoT-Admins eignet, haben wir getestet.**

Von Jan Mahn

Die Bastelphase ist beendet, der Raspberry Pi hat eine verantwortungsvolle Aufgabe als Smart-Home-Zentrale, Überwachungskamera (siehe S. 154) oder als Boots-Navigationshilfe (siehe S. 160) bekommen. Es beginnt die Betriebsphase und ein mehr oder weniger wichtiger Teil des Alltags ist davon abhängig, dass es dem Minirechnerchen gut geht. Noch wichtiger ist das Wohlerge-

hen solcher IoT-Hardware für Firmen, die damit ihr täglich Brot verdienen. Hat man mehr als ein Maschinchen zu warten, reicht es schnell nicht mehr aus, ab und zu mal per SSH-Verbindung nach dem Rechten zu sehen oder sich nur dann um Updates und Logs zu kümmern, wenn sich schon Fehler bemerkbar machen.

An Geräteverwalter, die eine ganze Flotte Linux-betriebener IoT-Geräte betreuen, richtet sich das Angebot der Firma qbee aus Norwegen. Das Unternehmen verspricht eine cloudbasierte Flottenverwaltung, die über den Browser bedient wird. Auf der verwalteten Hardware läuft ein Agent, der Informationen zum Betrieb an die Cloud meldet und von dort auch Befehle entgegennimmt. Was im Kern für Unternehmen mit wirklich vielen Geräten gedacht ist, kann sich auch für Raspi-Bastler lohnen, denn bis zu zwei Geräte darf man mit einem kostenlosen Konto fernsteuern. Die Firma erhofft sich davon vermutlich, dass so manch privater Bastler auch beruflich mit IoT zu tun hat und die Software dem

Arbeitgeber vorschlägt. Wie gut die Fernwartung im Alltag funktioniert, haben wir über Monate mit einem Raspi getestet, der als Smart-Home-Zentrale arbeitet.

Die Einrichtung geht flott vonstatten, nachdem man auf [qbee.io](https://qbee.io) sein Konto eingerichtet hat. Unter dem Menüpunkt „Devices“ gibt es die Schaltfläche „Add device“. Hat man keinen Raspi oder andere IoT-Hardware zur Hand, kann man eine solche zum Experimentieren auch simulieren, indem man mit Docker einen Testcontainer hochfährt; den Befehl zeigt die Weboberfläche praktischerweise gleich an. Wer echte IoT-Hardware fernsteuerbar machen will, kopiert einen Code-Schnipsel, der per `wget` den Agent herunterlädt und diesem einen Schlüssel mitgibt, der ihn in der qbee-Cloud identifiziert. Wem ein Installationsskript, das Root-Rechte erfordert, nicht geheuer ist, für den ist qbee nicht das richtige Produkt. Man muss sich bewusst sein, dass der qbee-Agent weitreichende Rechte bekommt, damit er Befehle umsetzen kann.

## Schnell am Start

Im Test dauerte das Einrichten eines neuen Geräts wenige Sekunden. Die Hardware muss kein Raspi sein, der Agent läuft auf allerlei ARM-Prozessoren und auch auf x86-64-Hardware; selbst die steinalte Intel-386-Architektur wird noch unterstützt. Als weitere Zielgruppe spricht qbee Nutzer von OpenWrt an, die viele Router zu steuern haben. Der qbee-agent liegt auch als Paket im offiziellen OpenWrt-Paketrepository. Hat man viele kleine Helferlein oder Router auf der Welt verteilt, hinterlegt man für jedes den Standort und kann sich die eigene Flotte auf einer Karte ansehen.

Ist der Agent in Stellung gebracht, zeigt qbee viele Details zu dem, was im Linux vor sich geht. Dazu gehören Zusammenfassungen von Fehlern aus Logs nebst Diagrammen mit CPU- und RAM-Auslastungen sowie Speicherfüllständen und die Netzwerkkartenlast. Wie bei Monitoring-Lösungen üblich, kocht auch qbee nur mit Wasser und bezieht die Daten aus den Schnittstellen, die das Betriebssystem ohnehin bereitstellt. Es zeigt an, was man auch mit anderen Werkzeugen zu sehen bekommt, das aber zentralisiert und schematisiert.

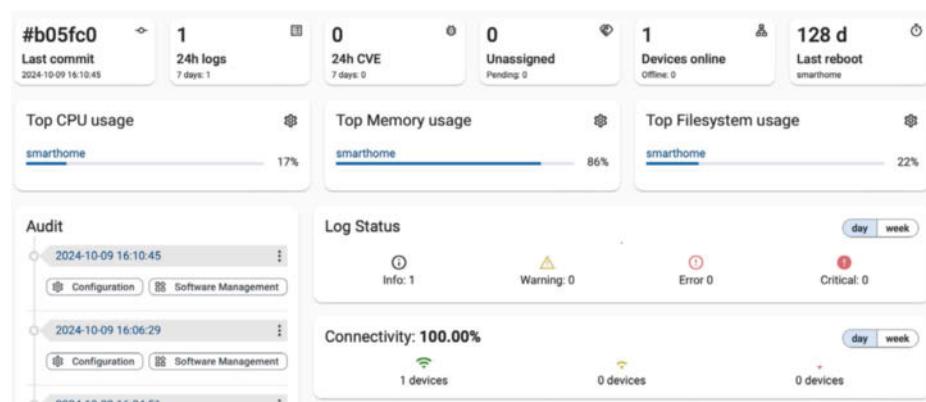
Ein Mehrwert von qbee: In den Einstellungen kann man Schwellwerte für Alarne hinterlegen. Dann schickt die Plattform zum Beispiel eine E-Mail, wenn ein Raspi unter zu hoher CPU-Last leidet oder die SD-Karte vollläuft. Schade: Andere Benachrichtigungsanäle wie die Chatsysteme Rocket.Chat, Slack oder auch Microsoft Teams, die bei Admins dafür vermehrt genutzt werden, beherrscht qbee noch nicht.

Prozesse, die in Docker- oder Podman-Containern laufen, bekommen in qbee eine eigene Übersicht. Die Oberfläche bietet aber nur lesenden Zugriff, neu starten oder löschen kann man Container über den Browser nicht. Wer solche Funktionen braucht, ist mit einer Containerverwaltungsplattform wie Portainer besser bedient.

## Fernkonfigurieren

qbee will aber mehr sein als ein aufgeblasenes Monitoringwerkzeug und kann Geräte auch steuern. Zum Funktionsumfang gehört unter anderem eine Dateiverwaltung. Dafür lädt man Dateien in die Plattform hoch und erstellt dann die Anweisung, auf welchen Geräten (oder Gruppen von Geräten) sie diese unter welchem Pfad ablegen soll.

Zusätzlich kann man je nach Betriebssystem auch Pakete über den jeweiligen



**Auf dem Dashboard erfährt man, wie es der eigenen Flotte geht und sieht, wo es möglicherweise klemmt. Für jedes Gerät gibt es auch detaillierte Diagramme und Statusberichte zu sehen.**

Paketmanager installieren lassen. Die Funktion heißt „Software Management“ und erwartet entweder eine Paketdatei (.deb für Debian-basierte Systeme, .rpm für Red Hat oder .ipk für OpenWrt) oder den Namen eines Pakets in den hinterlegten Paketquellen. Wie nützlich eine solche Funktion ist, wird klar, wenn man sich vorstellt, man müsse per Hand oder Skript zum Beispiel Hunderte Kassen-PCs mit einem Paket versorgen. Im Test war die Installation nach wenigen Minuten erledigt, den Arbeitsbericht des Agenten findet man in der Weboberfläche links unter Logs. Auf ähnlichem Weg kann man auch Container per Docker oder Podman konfigurieren und hochfahren lassen.

Jede Änderung, die man auf ein Gerät oder eine Gruppe loslässt, ist nicht nur ein Klick in einer Konfigurationsoberfläche – das wäre unpraktisch, wenn man Änderungen später auf andere Maschinen anwenden will oder eine Konfiguration rückgängigmachen muss. Stattdessen schnürt qbee immer sogenannte Konfigurationspakete, die aus mehreren aktivierten Einstellungen und Anweisungen bestehen und die man am Ende „committen“ muss, optional auch mit einem Bearbeitungsvermerk.

Jedes fertige Konfigurationspaket kann man als JSON-Datei exportieren und woanders einsetzen. Geht mal etwas schief, findet man links im Menü unter „Audit“ eine detaillierte Liste aller Ände-

The interface shows a sidebar with categories like Settings, System, Software, Security, and Container Management. The Software category is selected, showing the Software Management section with the following details:

- Software Management**: Install packages, monitor services and add key-value templating.
- Enabled**: Checked
- Extend**: Checked
- Packages**: A list box contains "jq". Below it are fields for "Software package\*" (jq), "Service name", and "Pre-condition (optional)".

**Statt sich auf jedem zu verwaltenden Raspi per SSH anzumelden, um über den Paketmanager ein Paket zu installieren, legt man über die Funktion „Software Management“ einen Befehl zur Installation an. Der Agent stopft es dann in ein oder mehrere IoT-Geräte und schreibt auch einen Bericht.**

### Configuration changes

Change	Node / Tag	Labels	Created at	Created by
9d3108	smarthome	Configuration Software Management	2024-10-09 16:10:42	Jan Mahn
<pre>{   "bundle_commit_id": "9d31083259fee67ec599d9729534e30cff8ff6bfff3f71ce4fdcc455dbfe04749",   "enabled": true,   "extend": true,   "items": [     {       "package": "jq"     }   ],   "version": "v1" }</pre>				

Eine Änderung ist nicht nur ein Klick in der Weboberfläche. qbee schnürt im Hintergrund Konfigurationspakete und zeigt sie als JSON-Objekt an, das man exportieren und anderswo nutzen kann.

rungen inklusive Verursacher sowie Datum und Uhrzeit. Ein weiterer Vorteil gegenüber vielen anderen Strategien zur Geräteverwaltung: Ist ein Gerät mal offline und qbee hat noch anzuwendende Konfigurationspakete, lädt der Agent diese unmittelbar herunter, wenn es wieder eine Internetverbindung gibt.

Wem die Verwaltung über die Weboberfläche nicht ausreicht, der kann qbee auch in bestehende Workflows integrieren. Das API ist dokumentiert und zum Benutzen freigegeben. Außerdem gibt es eine fertige Integration in die CI/CD-Umgebung GitHub Actions. Wer dort beispielsweise seine Software baut und möchte, dass eine Flotte aus Raspberrys jede neue Version nach dem Kompilieren automatisch herunterlädt und ausführt,

findet dafür in der „FAQ“ in der qbee-Dokumentation ([qbee.io/docs](https://qbee.io/docs)) eine Anleitung. Viele weitere Inspirationen für Projekte mit qbee findet man in der Dokumentation unter dem Punkt „Tutorials“.

Will man ein Gerät einmal über die Kommandozeile bedienen, muss man mit qbee weder eine Portweiterleitung für SSH einrichten, noch die öffentliche IP-Adresse des Routers kennen. Über den Agenten kann man aus der Cloudoberfläche heraus eine Kommandozeilensitzung in einem Browserfenster aufbauen.

### Fazit

Je größer die eigene IoT-Flotte und je schwerer erreichbar die einzelnen Geräte, desto eher lohnt sich qbee. Das heißt aber nicht, dass der kostenlose Account,

mit dem man zwei Geräte fernwarten kann, nutzlos wäre: Schon wer einen Smart-Home-Raspi im Ferienhaus deponiert hat und nicht gleich ein WireGuard-VPN oder eine Portweiterleitung für SSH basteln will, um ihn von zu Hause unter Kontrolle zu behalten, profitiert von der schnell eingerichteten Cloudverwaltung.

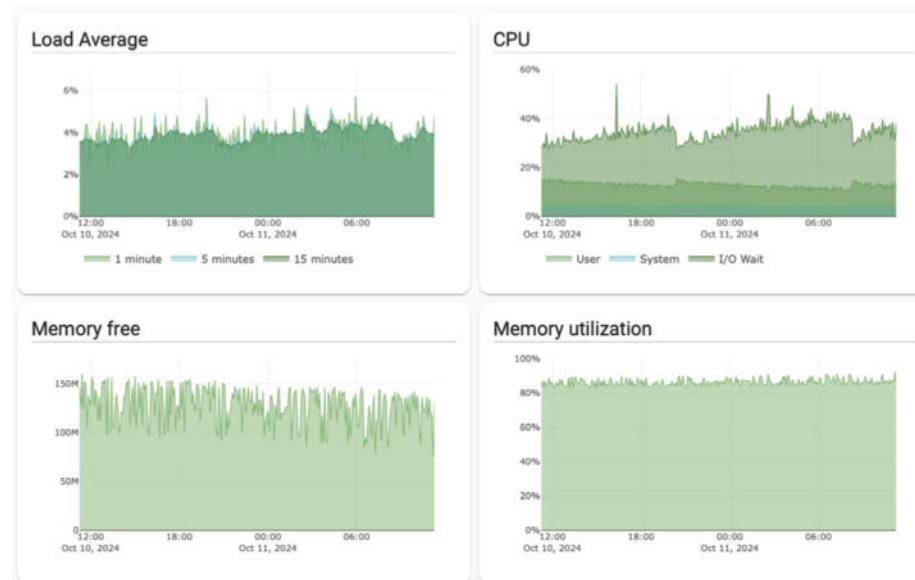
qbee ist daher ein Tipp für alle Heimbastler, die bereit sind, die Kontrolle an den Betreiber der Plattform abzugeben und dieser vertrauen. Die Firma aus Norwegen ist an die DSGVO gebunden, weil Norwegen Teil des Europäischen Wirtschaftsraums ist, wenn auch nicht der EU. Die Firma listet auf der Website unter anderem den Industriezulieferer ABB sowie Firmen aus der Automobil- und Eisenbahnbranche als Kunden.

Für IoT-Admins mit großen Flotten ist die durchdachte Funktionsweise interessant: Jede Änderung ist Teil eines Konfigurationspaketes, dessen Anwendung sauber dokumentiert wird. Das erleichtert die Arbeit im Team. Auch eine Rollen- und Berechtigungsverwaltung für Mehrnutzerbetrieb ist für solche Szenarien eingebaut. Integration per API und GitHub Actions ist das Sahnehäubchen.

Die Plattform hat bereits einen großen Funktionsumfang und die Verbindung zu den Agenten funktionierte im Test stets reibungslos, ganz ohne Löcher in der Firewall. Mit den qbee-Servern spricht der Agent per HTTPS auf Port 443. Läblich für eine Plattform, die so weitreichenden Zugriff auf die IoT-Geräte zulässt: Oben rechts unter „Company“ kann man für das eigene Team – das auch aus einer Person bestehen kann – die Zwei-Faktor-Authentifizierung per TOTP (Time-based one-time password) aktivieren. Dann kann sich jeder Nutzer eine TOTP-App einrichten und das Konto zusätzlich schützen.

(jam@ct.de) ct

Dokumentation: [ct.de/yw88](https://ct.de/yw88)



qbee speichert unter anderem Informationen zum Ressourcenverbrauch und zeigt sie in Diagrammen an. Mit einem kostenlosen Account bewahrt der Anbieter die Daten 7 Tage lang auf.

## qbee

### IoT-Verwaltungsplattform

Funktionen (Auswahl)	Logs, Metriken, Warnungen, Paketverwaltung, Fernzugriff, Dateiverteilung, Containerverwaltung, Konfigurationsverwaltung
Speicherdauer der Betriebsdaten	1 Woche (kostenloser Tarif), 4 Wochen (Startup-Tarif)
Preis	kostenlos (2 Geräte), Startup-Tarif: 89 US-Dollar/Monat (20 Geräte)

# WIR TEILEN KEIN HALBWISSEN. WIR SCHAFFEN FACHWISSEN.



20. - 21.11.



## WORKSHOP

### Einführung in den Kea DHCP Server

Erfahren Sie alles über Kea-DHCP-Software auf Unix- und Linux-Systemen. Sie lernen mehr über die Installation, Konfiguration und Betrieb des Systems.

21. + 28.11.



## WORKSHOP

### CI/CD mit GitLab

Der Workshop bietet eine praktische Einführung in die GitLab-CI-Tools und zeigt, wie man damit Softwareprojekte baut, testet und veröffentlicht.

26. - 27.11.



## WORKSHOP

### Docker und Container in der Praxis

Der Workshop richtet sich an Entwickler die neu in das Thema einsteigen. Der Fokus liegt vor allem auf den Herausforderungen im Alltag.

03.12.



## WORKSHOP

### Kluge Strukturen für Microsoft 365

Lernen Sie, wie Sie gemeinsam mit Ihren Mitarbeitern Leitlinien entwickeln, um das volle Potenzial für die Zusammenarbeit auszuschöpfen.

04.12.



## WORKSHOP

### Einführung ins Threat Modeling

Sie erfahren, was Threat Modeling ist und wie es genutzt werden kann, um IT-Produkte schon vor der Implementierung abzusichern

10.12.



## WEBINAR

### Digitaler Produktpass

Im Webinar erklärt der Referent, wie Sie Ihre Daten für den digitalen Produktpass aufbereiten und mit digitalen Zwillingen Ihre Effizienz steigern.

Sichern Sie sich Ihren Frühbucher-Rabatt:  
[heise.de/ct/Events](http://heise.de/ct/Events)

# Kleiner Bruder

## MicroPython-Entwicklung auf dem Raspberry Pi Pico

**Mit dem Raspi Pico können Sie viele Projekte, die Sie auf dem großen Bruder Raspberry Pi entwickelt haben, leicht übernehmen und viel kostengünstiger umsetzen. Denn nicht nur die Anschlüsse des Kleinen sind weitgehend kompatibel, mit dem MicroPython-Interpreter läuft auch Ihre Software mit nur wenigen Änderungen.**

Von Mirko Dölle

Für einfache Steueraufgaben ist der Raspberry Pi oft viel zu teuer und zu komplex, schließlich ist er ein Einplatinencomputer mit Mehrkern-Prozessor, Massenspeicher und Monitoranschluss, der ein vollständiges Betriebssystem wie Linux erfordert. Beim Raspi Pico respektive Pico W handelt es sich hingegen um vergleichsweise simple Mikrocontroller, auf denen lediglich ein einziges, im Funktionsumfang konkret umrissenes Programm läuft. Dafür kostet er mit 4 bis 7 Euro auch nur einen Bruchteil eines Raspberry Pi.

Eine Gemeinsamkeit des Raspi und des Pico ist der 40-polige GPIO-Anschluss – allerdings hat er beim Pico eine andere Pin-Belegung. Beide verwenden 3,3-Volt-CMOS-Pegel, sodass Sie praktisch alle Schaltungen vom Raspi für den Pico übernehmen können. Die Anschlussbezeichnung am Mikrocontroller hat aber nichts mit der Pin-Nummer an den Anschläßen der Platine zu tun: Pin 20 etwa heißt intern GP15.

Dass Sie den Raspi Pico in MicroPython programmieren können, erleicht-

tert die Portierung von Raspi-Projekten ungemein. MicroPython ist zwar nicht vollständig kompatibel zu Python, das Sie vom Raspi kennen, meist genügen aber wenige Anpassungen, um ein Programm vom Raspi auch auf dem Pico zum Laufen zu bekommen. Dabei hilft besonders die MicroPython-Dokumentation auf docs.micropython.org.

### Pin wechsle dich

Eine Besonderheit des Pico ist, dass Sie bei Zusatzfunktionen wie PWM (Pulsweitenmodulation) oder den I2C- oder SPI-Busen mehrere Pins zur Auswahl haben. Die Signale des ersten PWM-Generators PWM0 können Sie zum Beispiel entweder an den Pins 1 (GPO) und 2 (GP1) abgreifen oder aber an den Pins 21 (GP16) und 22 (GP17). Ebenso können Sie die beiden I2C-Busse (Inter-Integrated Circuit) I2C0 und I2C1 sowie für die SPI-Busse (Serial Peripheral Interface) SPI0 und SPI1 über verschiedene Pins herausführen. Außerdem hat der Pico drei analoge Eingänge mit 12 Bit Genauigkeit, die es beim Raspberry Pi nicht gibt. Welcher Anschluss welche Funktionen bietet, finden Sie im Kasten Pin-Belegung rechts oben.

Nach dem Auspacken des Raspi Pico können Sie jedoch nicht gleich mit MicroPython loslegen: Sie müssen den Interpreter erst in dessen Flash-Speicher schreiben. Der Download-Link ist in der offiziellen Dokumentation auf der Homepage der Raspi-Foundation versteckt. Damit Sie sich nicht durchwühlen müssen, haben wir die Seite auch auf ct.de/y6gq verlinkt.

Achten Sie darauf, dass Sie die richtige Firmware-Version herunterladen – je nachdem, ob Sie einen Pico, Pico W oder gar einen Pico 2 verwenden. Die Firmwareredatei trägt die Endung .uf2. Nun halten Sie die Taste unterhalb des USB-Ports gedrückt und schließen den Pico mit einem USB-Kabel an Ihren Rechner oder einen Raspberry Pi an. Dort meldet sich der Pico als USB-Datenspeicher und wird

üblicherweise automatisch eingebunden. Kopieren Sie die Firmwareredatei einfach auf das Flash-Laufwerk.

### Thonny als Entwicklungshelfer

Nach dem Neustart läuft auf dem Raspi Pico der MicroPython-Interpreter und Sie können damit beginnen, den Mikrocontroller zu programmieren. Dazu eignet sich die Entwicklungsumgebung Thonny am besten. Diese ist auf Raspberry Pi OS mit Desktop bereits vorinstalliert, Sie finden sie im Menü unter „Entwicklung“. Auf anderen Systemen installieren Sie Thonny von der Website thonny.org. Zunächst arbeitet Thonny mit dem lokal installierten Python-Interpreter als sogenanntes Backend. Um das zu ändern, müssen Sie in der Statusleiste rechts unten zunächst das Backend-Menü öffnen und MicroPython auf dem Raspi Pico als neues Backend auswählen. Dazu muss dieser weiterhin per USB angeschlossen sein.

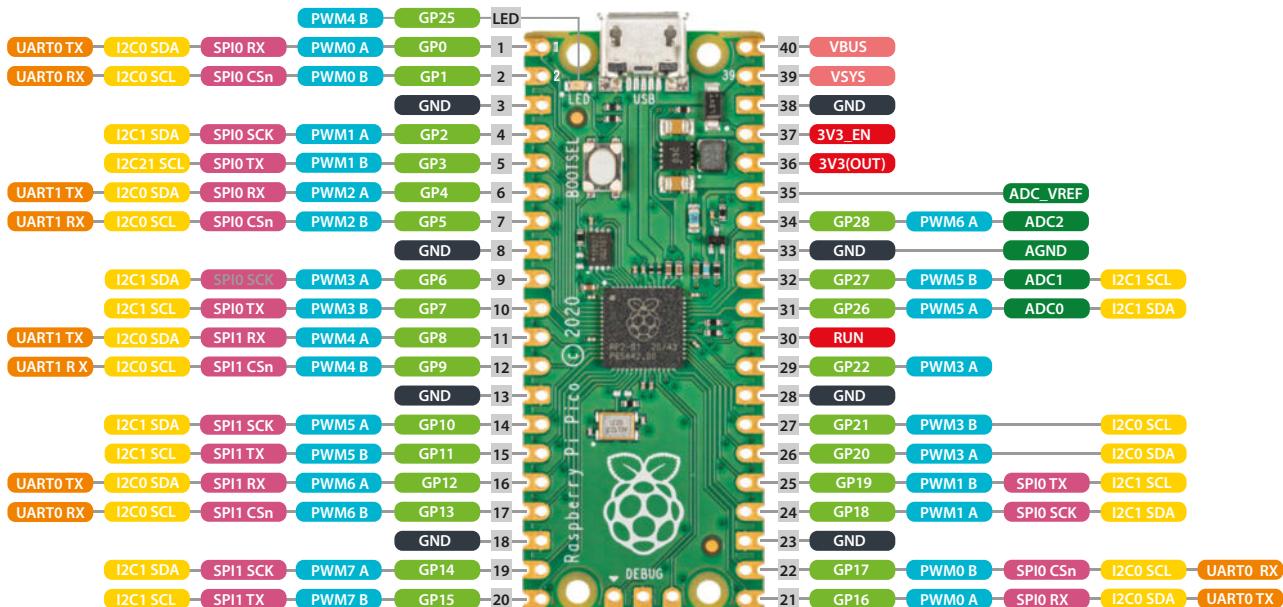
Mit Thonny programmieren und testen Sie direkt auf dem Raspi Pico. Haben Sie ein paar Zeilen Code eingegeben und wollen das Python-Programm speichern, so können Sie auswählen, ob es im Flash des Pico oder auf dem Computer gespeichert werden soll. Gleiches gilt, wenn Sie ein vorhandenes Programm öffnen. Um einen besseren Überblick zu bekommen, welche Dateien sich wo befinden, sollten Sie gleich zu Anfang im Menü „View“ „Files“ aktivieren. Dann erhalten Sie links in Thonny eine Navigationsleiste, in der Ihnen die lokalen Dateien und jene auf dem Pico angezeigt werden. Sie können darüber auch Dateien zwischen Computer und Pico hin- und herköpfieren.

Thonny startet mit einem leeren Code-Editor, in dem Sie unmittelbar die ersten Python-Befehle eingeben können – etwa um die LED des Pico einzuschalten:

```
from machine import Pin
led = Pin(25, Pin.OUT)
led.value(1)
```

Hier zeigen sich gleich die ersten Unterschiede: Während die Python-Bibliothek für den Raspberry Pi RPi heißt, binden Sie auf dem Pico die MicroPython-Bibliothek machine ein. Und während die GPIO-Anschlüsse des Raspi GPIO heißen, werden die Anschlüsse des Pico schlicht Pin genannt. Die zweite Zeile konfiguriert den Anschluss GP25, dort befindet sich die LED, also ein Ausgang. Die dritte Zeile schließlich schaltet die LED ein.

## Pin-Belegung des Raspi Pico (W)



Um das Programm auszuführen, speichern Sie es unter dem Namen „led.py“ auf dem Pico und klicken Sie auf den Play-Button in Thonny. Etwaige Fehlermeldungen oder Ausgaben Ihres Programms sehen Sie im „Shell“-Tab unterhalb des Code-Editors.

## **PWM statt Schleife**

Wenn Sie die LED nicht nur einschalten, sondern blinken lassen wollen, geht das so:

```
from machine import Pin
from utime import sleep
led = Pin(25, Pin.OUT)
while True:
    led.value(1)
    sleep(1)
    led.value(0)

    sleep(1)
```

Die while-Schleife ist nicht sehr elegant, schließlich blockiert sie den Pico für weitere Aufgaben. Um das Programm anzuhalten, müssen Sie den Stopp-Button in Thonny bemühen. Statt einer Schleife einen der PWM-Generatoren zu benutzen, um die LED blinken zu lassen, geht beim Pico so:

```
from machine import Pin, PWM  
led = PWM(Pin(25))  
led.freq(8)  
led.duty_u16(65536//10)
```

Eine Blinkfrequenz der LED von 8 Hertz ist die unterste Grenze des Pico – langsamer geht es nicht. Das Maximum liegt bei sportlichen 125 MHz. Der Duty Cycle muss als 16-Bit-Integerwert angegeben werden, 65536 // 10 liefert den entsprechenden ganzzahligen Teiler für zehn Prozent Einschaltdauer. Der Vorteil des PWM ist, dass die LED selbst dann blinkt, wenn das Programm längst beendet ist. Außerdem belastet das Blinken den Mikrocontroller nicht. In der Praxis eignet sich PWM vor allem, um bei hoher Frequenz die Helligkeit von LEDs über den Duty Cycle zu regeln oder um Servos anzusteuern.

## Interrupts statt Event-Handler

Ein weiterer wesentlicher Unterschied zwischen Python auf dem Raspi und MicroPython auf dem Pico: Ohne ein Betriebssystem gibt es auf dem Mikrocontroller kein Multitasking. Um auf Ereignisse wie zum Beispiel einen Tastendruck reagieren zu können, müssen Sie deshalb den Interrupt-Controller des Pico damit beauftragen, die gewünschte Funktion aufzurufen. Das folgende Beispiel schaltet die LED auf Tastendruck zwischen ein und aus um:

```
from machine import Pin  
led = Pin(25, Pin.OUT)  
button = Pin(0, Pin.IN, Pin.PULL_UP)  
  
def buttonPress(pin):
```

```
led.value((led.value() + 1) % 2)
```

```
button.irq(buttonPress, ¶Pin.IRQ_FALLING)
```

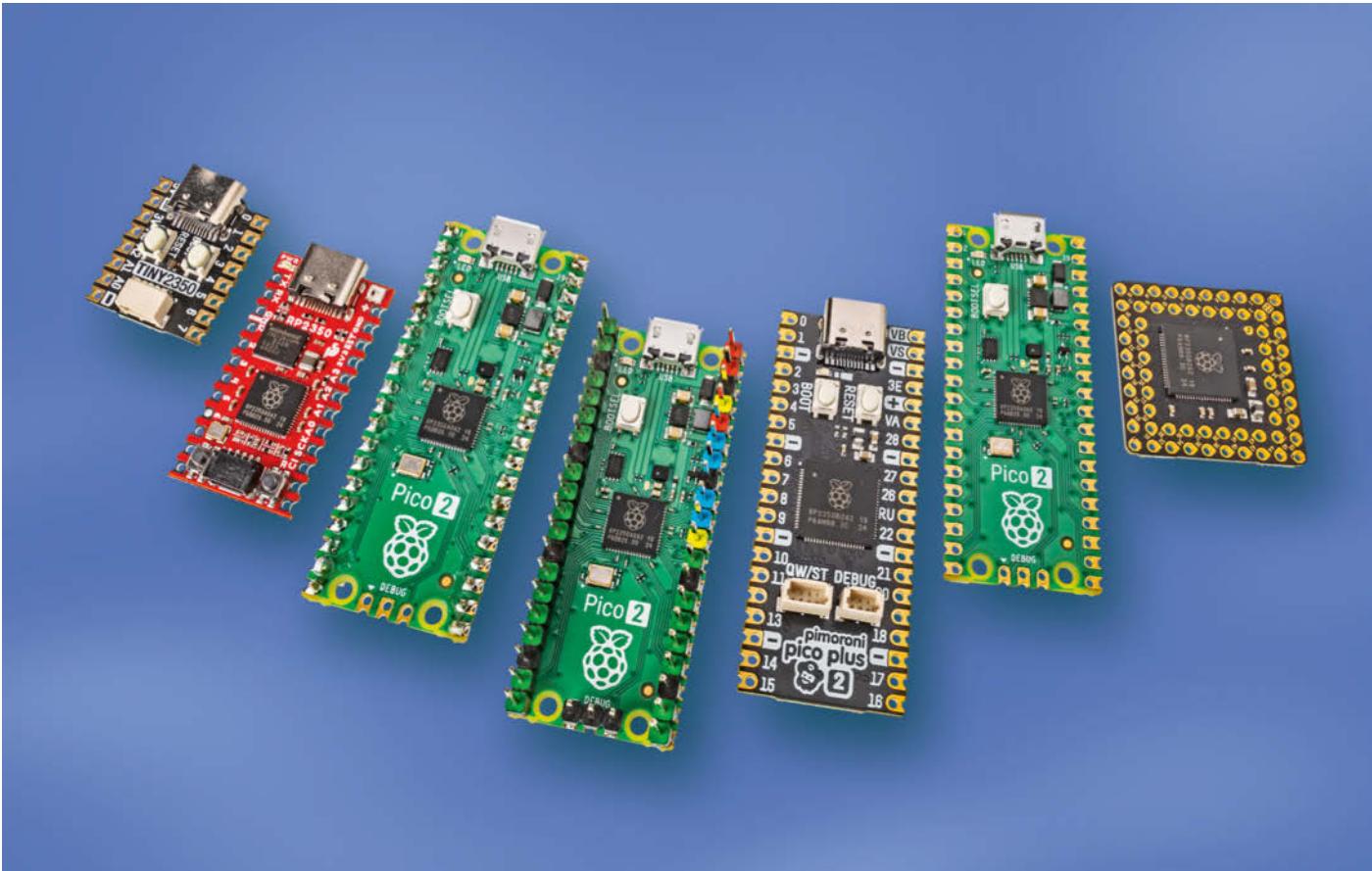
Um Tastenprellen (siehe S. 118) zu vermeiden, sollten Sie dem Taster zwischen den Pins 1 (GPO) und 3 (GND) des Pico unbedingt noch einen 100-nF-Kondensator parallel schalten – denn in MicroPython gibt es keinen Parameter, der das Tastenprellen softwareseitig herausfiltern würde.

Wenn Sie möchten, dass das Programm zum Ein- und Ausschalten der LED immer dann startet, wenn Sie den Pico mit Strom versorgen, so speichern Sie es unter dem Namen `main.py` auf dem Pico.

## Fazit

Der Raspberry Pi Pico ist eine interessante Alternative zu seinem großen Bruder, mit dem Sie viel kostengünstiger kleine Aufgaben umsetzen können, die keinen Einplatinencomputer mit einem ausgewachsenen Betriebssystem erfordern. Gleiche Signal- und Spannungspegel sowie Bussysteme erleichtern dabei die Portierung bestehender Projekte genauso wie der MicroPython-Interpreter des Pico – auch wenn dieser nicht ganz kompatibel zu Python auf dem Raspberry Pi ist. [mid@ct.de](mailto:mid@ct.de) 

*MicroPython und Dokumentation*: [ct.de/y6gg](https://ct.de/y6gg)



# Pi Harder

## Die Technik des Mikrocontrollers Raspberry Pi RP2350

**2021 brachten die Raspi-Macher den selbst entwickelten Chip RP2040 und das „Pico“-Board. Der stärkere Nachfolger RP2350 eröffnet jetzt völlig neue Möglichkeiten – auf dem Pico 2 und vielen anderen Boards.**

Von Dr. Maik Merten

**D**ie mittlerweile an der Börse notierte Firma Raspberry Pi Plc. hat sich ein weiteres Standbein geschaffen: Mikrocontroller-Chips. Die entwickelt das Raspi-Team selbst, anders als die Prozessoren der beliebten Linux-Einplatinen-

computer. Denn diese Mikroprozessoren beziehungsweise Systems-on-Chip kaufen die Raspi-Macher bisher alle von Broadcom zu.

Auf den 2021 eingeführten Mikrocontroller RP2040 folgt nun der deutlich stärkere und flexiblere RP2350, der trotzdem kaum mehr kostet, nämlich rund 90 Cent. Denn das Ziel sind besonders preisgünstige Schaltungen nicht nur für Bastler, sondern vor allem auch für gewerbliche Nutzer. Der RP2350 kommt außer auf dem rund 6 Euro billigen Raspberry Pi Pico 2 auch auf vielen anderen Platinchen von Firmen wie Adafruit, Melpoco, Pimoroni, Sparkfun oder Seeed Studio zum Einsatz.

Ebenso wie der RP2040 lässt sich auch der RP2350 vergleichsweise einfach programmieren, etwa mit der Arduino-

IDE, mit Python (Micropython, Circuitpython) oder Microsoft Visual Studio Code. Die Raspi-Macher stellen ein Software Development Kit (SDK) für C/C++ bereit. Unterstützung für Rust ist in Arbeit, ebenso wie für die Echtzeitbetriebssysteme FreeRTOS und Zephyr.

Der RP2350 kann mehr als ähnlich teure Mikrocontroller:

teurer: Er hat statt nur einem gleich zwei Prozessorkerne vom relativ neuen und starken Typ ARM

Cortex-M33. Außerdem hat er zwei alternativ nutzbare RISC-V-Kerne. Alle CPU-Kerne takten mit bis zu 150 MHz und können bis zu 520 KByte internes SRAM nutzen, mit externem Zusatzchip sogar mehr als 16 MByte Arbeitsspeicher. Das ist für Mikrocontroller geradezu gigantisch. Zudem lassen sich bis zu 16 MByte Flash-Speicher anschließen. Dazu kommen Be-

**ct Hardcore**

sonderheiten wie programmierbare I/O-Zustandsmaschinen (PIO), die leistungsfähige Schnittstellen wie CAN-Bus, USB und sogar Display-Interfaces wie DVI-D emulieren können. Dazu passend gibt es das schnelle Interface HSTX.

## Starke Kerne

Schon der Erstling RP2040 hat recht flinke 32-Bit-ARM-Kerne vom Typ Cortex-M0+, die mit bis zu 133 MHz takten. Deren nicht sehr umfangreicher Befehlsatz entspricht der Generation ARMv6-M, bei dem die meisten Befehle die platzsparende „Thumb“-Kodierung mit 16 Bit Länge haben. Es gibt nicht einmal einen Befehl zum Dividieren von Zahlen und auch keine für Gleitkomma-Arithmetik (Floating Point, FP). Daher hatte das Raspi-Team eine eigene Schaltung zum Dividieren in den RP2040 eingebaut. In dessen ROM liegen zudem einige liebevoll optimierte Fließkommaroutinen, die möglichst viel Rechenleistung aus den Kernen kitzeln.

Die 32-Bit-ARM-Kerne des RP2350 nutzen den mächtigeren Befehlssatz ARMv8-M, der unter anderem Divisionsbefehle umfasst. Für manche Einsatzzwecke noch wichtiger ist die eingebaute Gleitkommarecheneinheit (FPU) für 32-Bit-Daten (Single Precision). Weil sie keine doppelt genauen (FP64-)Werte verdaut, strickten die Raspi-Entwickler einen Koprozessor mit FP64-Basisoperationen an: Addieren, Subtrahieren, Multiplizieren, Dividieren und Radizieren.

Die M33-Kerne des RP2350 können aber noch mehr, denn sie haben auch Rechenwerke für digitale Signalverarbeitung (Digital Signal Processing, DSP). Die kann man grob mit den MMX-Einheiten alter x86-Prozessoren vergleichen: Sie umfassen SIMD-Befehle (Single Instruction, Multiple Data), um arithmetische Operationen gleichzeitig auf mehrere in Register gepackte Integer-Werte (vier 8-bittige oder zwei 16-bittige) anzuwenden. Das kann die Rechengeschwindigkeit erheblich steigern. Besonders für die Verarbeitung von Audio- und Bilddaten ist es nützlich, dass auch Sättigungsarithmetik unterstützt wird: Bei dieser bleiben die Ergebnisse innerhalb der Wertegrenzen des jeweiligen Datentyps, ohne dass es zu numerischen Über- oder Unterläufen kommt.

Schon gewöhnliche Mikrocontroller-anwendungen, bei denen es vor allem um Integer-Daten geht, dürfte der RP2350 pro

Takt um 50 bis 70 Prozent schneller verarbeiten als der RP2040; dazu kommt noch der um 13 Prozent höhere Takt (150 statt 133 MHz). Erste Messungen mit dem Mikrocontroller-Benchmark CoreMark zeigen, dass die ARM-Kerne des RP2350 etwa doppelt so schnell rechnen wie die des RP2040. Obendrein eröffnen FPU und DSP des M33 neue Einsatzbereiche, beispielsweise bei der Verarbeitung von Audio- und Videodaten: von der schnellen Fouriertransformation (FFT) über die Schätzung von Bewegungsvektoren bis hin zur Objektklassifikation mit kleinen neuronalen Netzen.

## RISC-V-Kerne

Die größte Überraschung im RP2350 sind aber die beiden RISC-V-Kerne, vor allem deshalb, weil die britische Firma ARM zu den größeren Aktionären von Raspberry Pi Plc. gehört. Die offene Befehlssatzarchitektur RISC-V steht nämlich in Konkurrenz zu ARM-Kernen, für deren Nutzung man Lizenzgebühren entrichten muss. Allerdings ist Raspberry Pi auch schon seit Jahren strategisches Mitglied der RISC-V-Foundation.

Die RISC-V-Kerne sind ebenfalls 32-Bitter. Entwickelt hat sie der Raspi-Mitarbeiter Luke Wren unter dem Namen „Hazard3“; ihr Design ist bei GitHub offen gelegt. Sie beherrschen wie auch die RISC-V-Kerne vieler anderer Mikrocontroller den Befehlssatz gemäß der Spezi-

## c't kompakt

- Der günstige Mikrocontroller Raspberry Pi RP2350 unterscheidet sich von anderen durch eine Kombination attraktiver Funktionen.
- Außer zwei ARM-Kernen hat der RP2350 auch welche mit RISC-V-Befehlssatzarchitektur.
- Der RP2350 ist sowohl für Bastler als auch für professionelle Entwickler interessant.

fikation RV32IMAC: 32-Bit-Integer (I), Multiplikation sowie Division mit Rest (M), atomare Operationen (A) und 16-bittige Befehlskodierung (C für Compressed). Hinzu kommen Befehle für Bitmanipulationen (B) sowie für kompakteren Code, insbesondere Push- und Pop-Operationen auf Registern (gemäß Zcmp). Auch unterstützt der Prozessor die unterschiedlichen Ausführungsprivilegien Debug, Machine und User. Was fehlt, sind Fließkommaeinheiten, weshalb die RISC-V-Kerne dabei deutlich schwächer sind als die Cortex-M33. Ganzzahlen verarbeiten sie allerdings fast ebenso schnell wie letztere – beeindruckend für einen selbst entwickelten Kern!



Den RP2350 gibt es in zwei Gehäusen: Der größere QFN80-Chip (rechts) hat 48 GPIO-Kontakte, die QFN60-Version bleibt bei 30 – wie der Vorgänger RP2040.

Die Siliziumfläche (das Die) des RP2350 ist mit 5,3 Quadratmillimetern mehr als doppelt so groß wie die des RP2040 (2 mm<sup>2</sup>). Laut Raspi-Chef Eben Upton liegt das nicht an den zusätzlichen RISC-V-Kernen, denn die sind winzig und deutlich kleiner als die Cortex-M33-Kerne. Mehr Chipfläche belegen vor allem der deutlich größere Speicher und die zusätzlichen Funktionen. Sowohl den RP2040 als auch den RP2350 lässt Raspberry Pi Plc. mit 40-Nanometer-Technik von TSMC fertigen. Auf eine Siliziumscheibe (Wafer) mit 30 Zentimetern Durchmesser passen mehr als 10.000 RP2350-Chips.

## Mehr Speicher

Mit 520 Kilobyte eingebautem SRAM hat der RP2350 fast doppelt so viel Arbeitsspeicher wie der RP2040 (264 KByte). Außerdem lässt sich das RAM nun via QSPI erweitern, und zwar auf bis zu 16 MByte Pseudo-SRAM (PSRAM). Pseudo-SRAM heißt so, weil es kein echtes SRAM

ist, sondern billigeres DRAM mit angebauter Refresh-Logik, das sich ähnlich wie SRAM ansteuern lässt. Ein QSPI-PSRAM-Chip mit 16 MByte kostet rund 1 Euro, wenn man große Stückzahlen kauft, und ist damit teurer als der RP2350 selbst.

QSPI bezeichnet ein klassisches Serial Peripheral Interface (SPI), aber mit vier (Quad) Datenleitungen statt nur einer einzigen. Wie sein Vorgänger hat der RP2350 keinen integrierten Flash-Speicher, weshalb der ebenfalls per QSPI angeschlossen werden muss; an den einzigen QSPI-Port kann man gleichzeitig Flash und PSRAM löten. Auch die Flash-Kapazität beträgt maximal 16 MByte; ein solcher Chip ist ab etwa 1,30 Euro erhältlich.

Aus dem Flash-Speicherchip kann der RP2350 wie sein Vorgänger Programmcode direkt ausführen, ohne ihn vorher ins SRAM zu kopieren: execute in Place, XIP. Damit das trotz der QSPI-Anbindung mit ordentlicher Performance läuft, gibt es einen XIP-Cache mit 16 KByte Kapazität.

Dieser Pufferspeicher lässt sich als interner Speicher umwidmen, was aber nur sinnvoll ist, wenn man auf XIP verzichtet und Code vor der Ausführung ins SRAM kopiert.

Auch das Boot-ROM ist beim RP2350 größer als beim RP2040, es fasst nun 32 statt 16 KByte. Ganz neu ist ein 8 KByte großer Speicher, der sich nur einmal beschreiben lässt: One-Time Programmable-(OTP)-Memory. Solchen Speicher nennt man auch E-Fuses; er dient beispielsweise dazu, kryptografische Schlüssel in den Chip zu „brennen“, um unbefugte Code-Manipulationen zu erschweren.

## Mehr (GPIO-)Pins

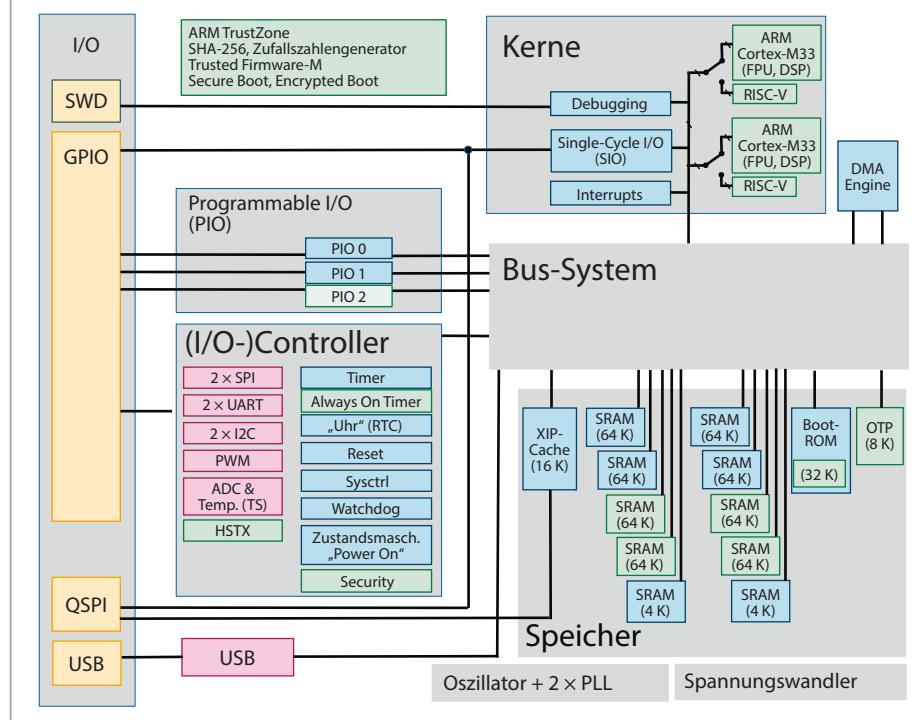
Der RP2350 steckt wie der RP2040 in einem quadratischen Gehäuse ohne Beinchen (Quad Flat No leads, QFN). Der RP2040 hat stets 56 Kontakte (QFN-56), der RP2350 aber 60 oder 80. Es gibt sogar vier unterschiedliche Versionen, nämlich mit 60 oder 80 Kontakten sowie ohne oder mit aufgestapeltem Flash-Speicherchip mit je 2 MByte Kapazität, siehe Tabelle auf Seite 135. Bei den Versionen RP2350B und RP2354B in QFN80-Bauform sind bis zu 48 GPIO-Pins nutzbar, bei den QFN60-Typen weiterhin bis zu 30. Davon sind auf dem Pico 2 höchstens 26 zugänglich, einer davon ist mit der Onboard-LED verbunden. Auf dem Pimoroni Pico Plus 2 sitzt hingegen der RP2350B mit 48 GPIOs.

Den aufgestapelten Flash-Speicherchip liefert die taiwanische Firma Winbond zu; die Stapeltechnik spart vor allem Platinenplatz. RP2354A/B eignen sich somit vor allem für besonders winzige Geräte. Auf dem Pico 2 sitzt wie beim RP2040 weiterhin ein separater Flash-Chip, nun aber mit 4 statt 2 MByte Kapazität.

Der RP2350 hat wie der RP2040 schnelle programmierbare Zustandsmaschinen für die Ein- und Ausgabe digitaler Signale (Programmable I/O, PIO), aber nun zwölf statt zuvor nur acht davon, nämlich drei statt zwei Blöcke mit je vier PIO-Einheiten. Damit lassen sich leistungsstarke Schnittstellen nachbilden, ohne mit dem üblichen „Bit-Banging“-Verfahren die CPU-Kerne zu belasten. Das geht vom simplen Ansteuern von RGB-LEDs (WS8212B) über CAN-Bus-Protokoll bis hin zu komplexem USB, VGA und DVI-D. Die CPUs können zudem nun direkt auf die PIO-FIFOs zugreifen und DMA-Requests haben einen Takt weniger Latenz. Die PIO-Idee ist nicht grundsätzlich neu, sondern findet sich auch in PSoC-Chips

## Blockschaltbild Raspberry Pi RP2350

Der von Raspberry Pi Plc. entwickelte Mikrocontroller RP2350 hat zwei 32-Bit-ARM-Kerne von Typ Cortex-M33 sowie zwei alternativ nutzbare RISC-V-Kerne (RV32IMAC). Die Änderungen zum Vorgänger RP2040 sind grün hinterlegt. Insgesamt sind 520 KByte SRAM und 8 KByte einmalprogrammierbarer Speicher (OTP) eingebaut, aber kein Flash-Speicher. Bis zu je 16 MByte Flash und Pseudo-SRAM (PSRAM) lassen sich per QSPI anschließen. Der RP2350 führt Daten aus dem Flash direkt aus (Execute-in-Place, XIP) und puffert die Zugriffe in 16 KByte XIP-Cache. Das Boot-ROM wuchs von 16 auf 32 KByte, auch wegen ARM TrustZone.



von Infineon (früher Cypress), einigen „Sitara“-Baureihen von Texas Instruments (TI) sowie im Parallax Propeller.

Auch andere I/O-Fähigkeiten wurden verbessert. Es gibt nun 24 (statt 16) PWM-Kanäle. Der Analog-Digital-Wandler (ADC) lässt sich bei den B-Varianten mit mehr Anschlüssen an bis zu acht GPIO-Pins nutzen, bisher waren es höchstens vier. Er liefert weiterhin 12 Bit, soll aber präziser arbeiten als beim RP2040. Dort schaffte er eine reale Auflösung (Effective Number of Bits, ENOB) von 8,7, beim RP2350 nennt das Datenblatt 9,5. Besonders genau misst aber auch der RP2350 erst, wenn man eine externe Referenz anschließt. Die Abtastrate beträgt maximal 500.000 Samples pro Sekunde; weiterhin ist ein interner Temperaturfühler vorhanden. Die USB-Schnittstelle ist unverändert: Es bleibt bei USB 1.1 mit bis zu 12 MBit/s, wobei der Mikrocontroller sowohl als USB-Host als auch als USB-Gerät agieren kann.

### HSTX-Speed

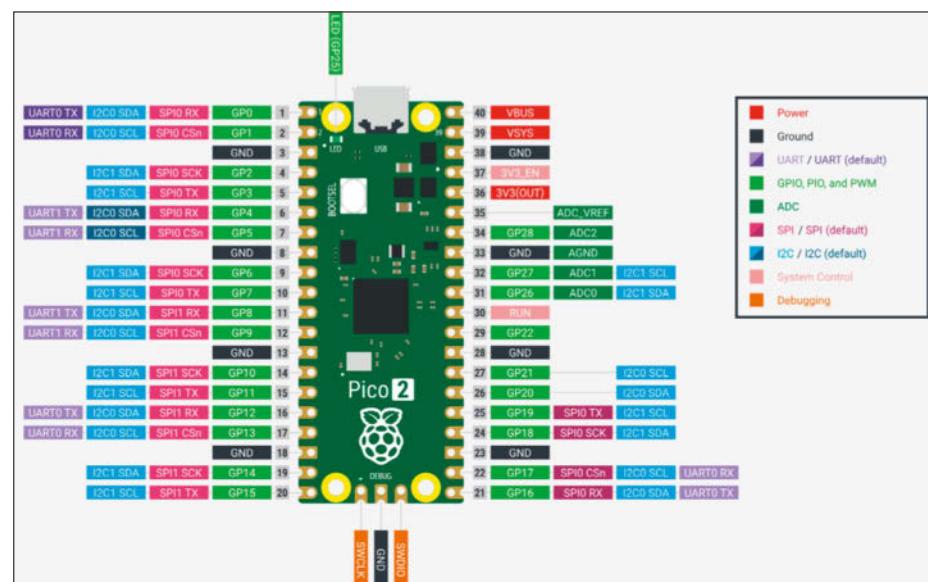
Eine Neuheit des RP2350 ist seine „High-Speed Serial Transmit“-Einheit (HSTX). Sie sendet pro Taktzyklus je zwei Bit (Double Data Rate, DDR) über bis zu acht Datenleitungen. Bei der maximalen Taktfrequenz von 150 MHz (also 300 Megabit pro Leitung) ergibt sich eine Datentransferrate von beachtlichen 300 Megabyte pro Sekunde.

Die HSTX-Übertragungsfrequenz liefert ein konfigurierbarer Generator unabhängig vom Systemtakt. HSTX gibt optional an je zwei benachbarten GPIO-Pins zueinander inverse Signale aus und erzeugt so ein Pseudo-Differential-Signal. Eine Anwendung hierfür ist die Ansteuerung von DVI- oder HDMI-Monitoren. Dafür will Adafruit bald eine Adapterplatine liefern, die mit einem passenden Transceiver-Chip bestückt ist. Der RP2350 eignet sich damit noch besser als Emulator alter Spieleplattformen. Ein Tüftler hat es mit dem RP2040 sogar schon geschafft, einen Apple Macintosh aus dem Jahr 1984 zu emulieren, inklusive VGA-Grafikausgabe.

### Effizienter & sparsamer

Trotz seiner weit höheren Rechenleistung zieht der RP2350 weniger elektrische Leistung als der 2040er. Dadurch arbeitet er insgesamt deutlich effizienter.

Zusätzlich beherrscht der RP2350 einige Stromspartricks, die der RP2040



**Auf dem Raspberry Pico 2 sitzt ein RP2350A, der 25 nutzbare GPIO-Kontakte bereitstellt. Das Pico 2 ist pinkompatibel zum Vorgänger Pico mit RP2040.**

nicht kennt. Teile des Systems lassen sich von der Versorgungsspannung trennen, wenn sie gerade nicht benötigt werden. Dazu ist der Chip in verschiedene Versorgungsbereiche (Domänen) unterteilt. Ein kleiner Logik-Teil bleibt immer mit Strom versorgt: Always On, AON-Domäne. Dafür gibt es auch einen separaten AON-Timer, der im Wesentlichen dazu dient, den restlichen Teil des Chips in bestimmten Abständen aufzuwecken. Die Prozessoren, die interne Bus-Logik sowie die Peripherie befinden sich in der Domäne SWCORE. Die XIP-Domäne umfasst im Wesentlichen den Cache für die XIP-Codeausführung über SPI. Jeweils rund die Hälfte des Arbeitsspeichers befindet sich in den Domänen SRAM0 und SRAM1.

Aus der Möglichkeit, verschiedene Domänen vom Saft abzuklemmen, ergeben sich zahlreiche unterschiedliche und unterschiedlich stromdurstige Betriebszustände. In den Zuständen P0.0 bis P0.4 werden Rechenkerne und XIP-Domäne versorgt, SRAM0 und SRAM1 aber nicht unbedingt. SRAM ohne Strom verliert nach wenigen Sekunden bis Minuten seine Daten.

Hinzu gesellen sich die Energiesparzustände P1.0 bis P1.7, in denen SWCORE abgeschaltet ist, also keine Rechenarbeit getätigter werden kann. Je nach Energiesparzustand bleiben unterschiedliche „SRAM-haltige“ Domänen versorgt. Wie viel Energie das spart, hängt im Wesentlichen davon ab, wie viel SRAM schlummert.

In den tiefsten Schlafzuständen kommt der RP2350 mit Bruchteilen eines Milliwatt aus, was insbesondere für Projekte attraktiv ist, die mit Akku- oder Batteriestrom laufen sollen. Braucht eine Anwendung nur den halben Arbeitsspeicher, lässt sich die andere Hälfte schlafen legen, um Energie zu sparen.

Auch der interne Spannungsregler für die digitale Logik des Chips wurde verbessert. Der RP2040 hat dafür einen mit 100 mA belastbaren Linearregler, der 1,1 Volt erzeugt. Der RP2350 hat einen effizienteren 200-mA-Schaltregler, der bei sehr geringer Last in einen linearen Regelmodus umschaltet. Dann liefert er höchstens 1 mA, aber das genügt für die P1.x-Sparzustände.

### Varianten des Mikrocontrollers Raspberry Pi RP2350

Modell	Flash	GPIOs	Kontakte	Gehäuse	Preis
RP2350A	–	30	60	QFN60	0,80 US-\$
RP2350B	–	48	80	QFN80	0,90 US-\$
RP2354A	2 MByte	30	60	QFN60	1,00 US-\$
RP2354B	2 MByte	48	80	QFN80	1,10 US-\$

Maße: QFN60 = 7 mm × 7 mm, QFN80 = 10 mm × 10 mm



**Das Adafruit Feather RP2350 hat einen 22-poligen HSTX-Anschluss; per Adapter mit Transceiverchip lässt sich darüber eine DVI-D- oder HDMI-Buchse anbinden.**

## Sicherheitsarchitektur

Die 8 KByte nichtflüchtigen OTP-Speichers im RP2350 lassen sich sogar gegen unbefugtes Auslesen schützen und sind unter anderem für kryptografische Schlüssel gedacht. Mit diesen kann man Codesignaturen vor dem Ausführen überprüfen oder verschlüsselten Code entschlüsseln, etwa für Funktionen wie Secure Boot und Encrypted Boot. Das funktioniert aber nur mit den ARM-Kernen, nicht mit RISC-V.

Secure Boot erschwert Malware-Angriffe, weil die Signaturprüfung veränderten Code enttarnt. Damit eignet sich der RP2350 auch für professionelle Projekte, bei denen ein Schutz gegen Firmware-Manipulationen nachgewiesen werden muss. Encrypted Boot ist vor allem für kommerzielle Geräte gedacht,

deren Entwickler verhindern wollen, dass ihr Code aus dem Flash-Speicher gestohlen wird.

Im RP2350 steckt zudem ein SHA256-Rechenwerk, das Hashes von Datenströmen mit einem Durchsatz von bis zu 79 MByte/s liefert. Außerdem ist ein Zufallszahlengenerator (True Random Number Generator/TRNG) eingebaut, der beispielsweise Saat für digitale Schlüssel bereitstellt.

Die ARM-Kerne haben das Funktionspaket TrustZone. Es ermöglicht es, sichere (Secure) von unsicheren (Non-Secure) Zugriffen und Operationen zu unterscheiden. Damit lassen sich geschützte Speicherbereiche einrichten, etwa für sensible Daten wie kryptografische Schlüssel und Passwort-Hashes. Das Raspi-Team kooperiert dabei mit dem Projekt Trusted Firm-

ware, genauer mit Trusted Firmware-M (TF-M).

Die RISC-V-Kerne unterscheiden zwischen Machine- und User-Mode (M-/U-Mode). Letzterer schränkt den Zugriff auf interne Statusregister des Prozessors ein und markiert Speicherzugriffe, wodurch sie sich auch filtern lassen.

## RP2350-Boards mit Funk

Wie sein Vorgänger besitzt der RP2350 keine eingebaute Funkschnittstelle, anders als etwa die ESP-Chips von Espressif. Es gibt aber bereits einige RP2350-Boards mit Funkmodulen und in den kommenden Monaten soll auch ein Raspberry Pi Pico 2 W für unter 10 Euro erscheinen, der WLAN und Bluetooth an Bord hat. Die Firma iLabs hat das 30 Euro teure Challenger+ RP2350 WiFi6/BLE5 angekündigt, auf dem auch der ESP32-C6 mit 4 MByte eigenem Flash sitzt; er funkelt per WLAN (2,4 GHz), Bluetooth, Zigbee und Thread. Außerdem spendiert iLabs dem RP2350 je 8 MByte Flash und PSRAM

## GPIO-Bug „E9“ im RP2350

Wer GPIO-Anschlüsse des RP2350 als Eingänge nutzen will, muss je nach Be- schaltung einen Bug umgehen. Der Fehler führt dazu, dass der RP2350 unter gewissen Bedingungen das Ein- gangssignal falsch interpretiert. Durchläuft der Signalpegel den undefinierten Bereich zwischen den Schwellen für logisch 0 und 1, stört ein geringer Leckstrom von ungefähr 120 Mikroampere. Die interne Pulldown-Schaltung ist dafür zu schwach, der Pegel bleibt im Bereich einer logischen 1 hängen. Deshalb muss man einen externen Widerstand von maximal 8,2 kΩ in die Schaltung einfügen. Dadurch eignet sich der Pico 2 nicht bei jedem Projekt als pin-kompatibler Ersatz für den Pico mit RP2040.

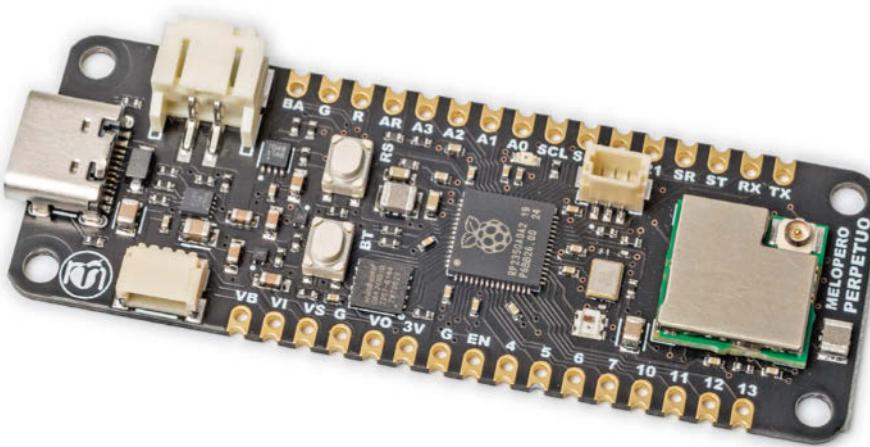
Der Fehler ist – wie einige weitere auch – im umfangreichen RP2350-Datenblatt vermerkt, und zwar mit der Bezeichnung „RP2350-E9: Increased leakage current on GPIO when pad input is enabled“. Betroffen sind alle aktuellen Chips der Revision A2 (Aufdruck: RP2350XXA2).

## Rechenkerne und Speicher in RP2040 und RP2350

Chip	RP2040	RP2350
CPU-Kerne	Cortex-M0+	Cortex-M33
Befehlssatz	ARMv6-M	ARMv8-M
Sicherheitsfunktionen	–	TrustZone-M
Coremark/MHz	2,46	4,1
max. Taktfrequenz <sup>2</sup>	133 MHz	150 MHz
Divisions-Rechenwerk	✓ <sup>3</sup>	✓
FPU + Integer-DSP	–	✓ <sup>4</sup>
SRAM	264 KByte (4 × 64 + 2 × 4)	520 KByte (8 × 64 + 2 × 4)
XIP-Cache	16 KByte	16 KByte
Boot-ROM	16 KByte	32 KByte
OTP-Speicher	–	8 KByte

<sup>1</sup> plus Bitmanipulation (B) und Push/Pop (Zcmp)  
ist möglich <sup>3</sup> nicht im ARM-Kern, sondern separat

<sup>2</sup> die Firmwares für die Arduino-IDE und Python setzen den Takt auf 125 MHz, Übertakten  
<sup>4</sup> 32-Bit-FPU und Integer-DSP im Cortex-M33, zusätzlich Double Precision Coprocessor



Der RP2350 kommt auch auf Boards mit Funkmodulen zum Einsatz, etwa auf dem Melopero Perpetuo mit LoRaWAN; ein Pico 2 W mit WLAN und Bluetooth ist angekündigt.

sowie einen Laderegler für einen Lithium-Polymer-(LiPo)-Akku. Das Melopero Perpetuo LoRa wiederum ist mit dem LoRaWAN-Modul Embit EMB-LR1276S (868 MHz) bestückt, in dem ein Microchip SAM R34 funkert.

Der RP2350 soll laut seinen Entwicklern bis mindestens 2045 gefertigt werden, also wie der RP2040 (bis 2041) noch rund 20 Jahre lang nach seiner Einführung. Das ist vor allem ein Versprechen an Industrie-kunden, die die Raspi-Mikrocontroller in großen Stückzahlen in eigene Geräte einbauen sollen. Die zusätzlichen Einnahmen durch den Verkauf an Firmenkunden sollen helfen, die Entwicklungskosten für kommende Raspi-Chips zu finanzieren.

## Fazit

Der Mikrocontroller RP2350 ist ein überraschend starker und vielseitiger Nachfolger des auch schon sehr flexibel nutzbaren RP2040. Die Raspi-Entwickler lassen dabei einerseits Baugruppen wie Flash-Speicher und Funkcontroller weg, um den Chip kompakt und die Herstellungskosten niedrig zu halten. Andererseits bauen sie Besonderheiten ein wie PIO-Controller, HSTX-Interface, RISC-V-Kerne, ARM TrustZone und OTP-Speicher. Offengelegte Firmware und lange Verfügbarkeit machen den RP2350 auch für Firmen attraktiv. Der Clou ist aber die leichte Programmierbarkeit, sowohl für Bastler als auch für Profis. (ciw@ct.de) **ct**

Es gibt **10** Arten von Menschen.

iX-Leser und die anderen.



## Bastelboards mit Raspberry Pi RP2350 (Auswahl)

Hersteller	Produkt	Flash / PSRAM	USB	Preis	Besonderheiten
Raspberry Pi	Pico 2	4 / – MByte	Micro-USB	5 €	LED
Raspberry Pi	Pico 2 W	4 / – MByte	Micro-USB	8 €	LED, WLAN/BT, angekündigt
Adafruit	Feather RP2350	8 / – MByte	USB-C	14 €	Stemma-QT, Feather, HSTX, LiPo
Adafruit	Metro RP2350	16 / – MByte	USB-C	offen	RGB-LED, Stemma-QT, HSTX, MicroSD, angekündigt
Cytron	Motion 2350 Pro	k.A.	USB-C	25 €	Motorsteuerung, Stemma, Qwiic, Grove, (RGB)-LEDs
ILabs	Challenger+ RP2350 WiFi6/BLE5	8 / 8 MByte	USB-C	30 €	WLAN/BT/Thread/Zigbee, Feather, LiPo, angekündigt
Melopero	Perpetuo LoRa	8 / – MByte	USB-C	25 €	LoRaWAN, Qwiic, RGB-LED, LED, LiPo
Pimoroni	Pico Plus 2	16 / 8 MByte	USB-C	14 €	Qwiic, SP/CE, LED, 48 GPIOs
Pimoroni	Tiny 2350	4 / – MByte	USB-C	9 €	Qwiic
Pimoroni	PGA2350	16 / 8 MByte	–	11 €	48 GPIOs, sehr klein, nur Lötanschlüsse
Seeed Studio	Xiao RP2350	2 / – MByte	USB-C	8 €	LED, sehr klein
SparkFun	Pro Micro RP2350	16 / 8 MByte	USB-C	15 €	RGB-LED

LiPo = Ladeschaltung für Lithium-Polymer-Akku; Grove, Feather, HSTX, Qwiic, Stemma(-QT) und SP/CE sind Steckanschlüsse unterschiedlicher Firmen.

Jetzt Mini-Abo testen:

3 digitale Ausgaben +  
Bluetooth-Tastatur  
nur **19,35 €**

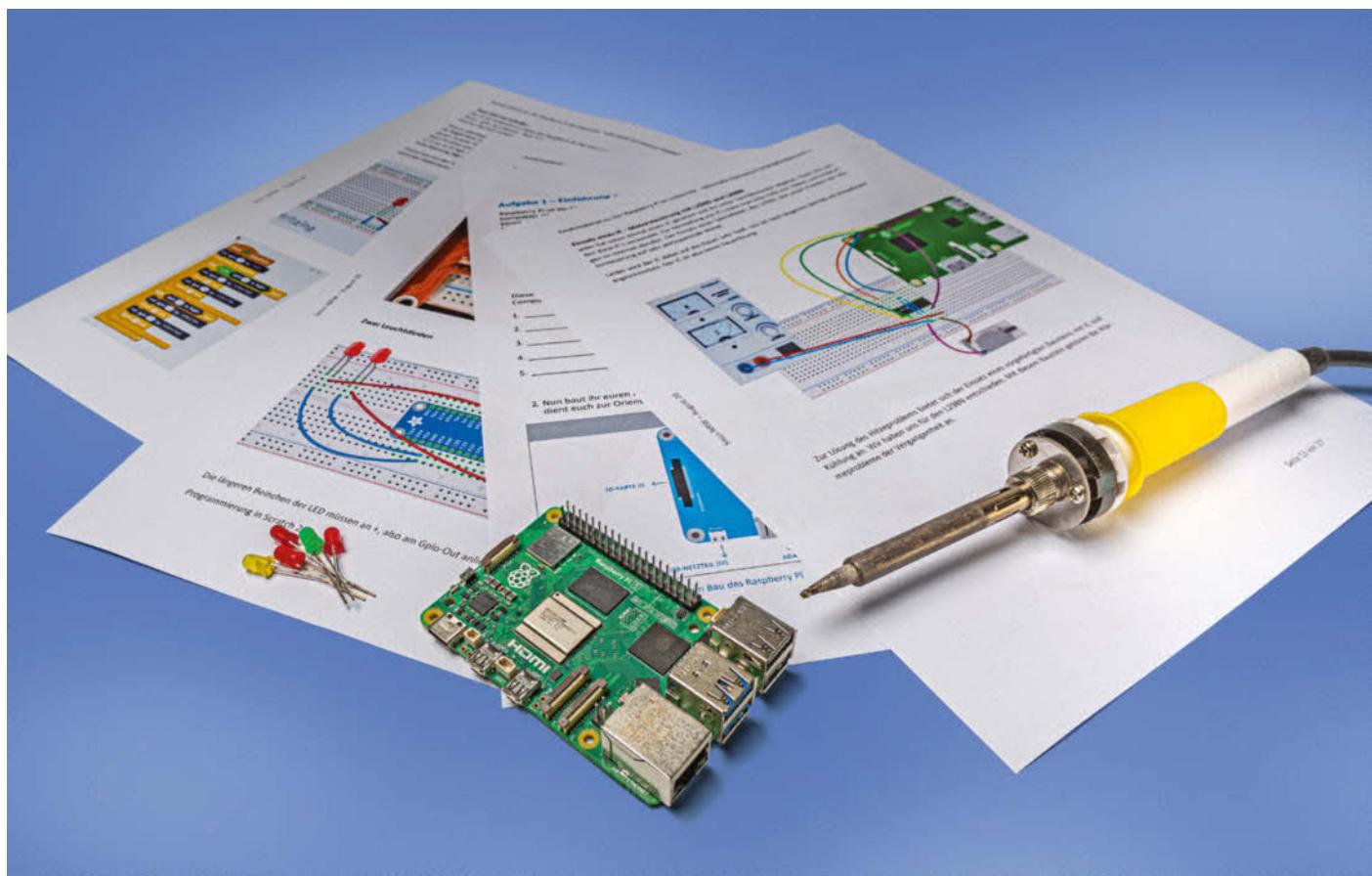
[www.iX.de/digital-testen](http://www.iX.de/digital-testen)



[www.iX.de/digital-testen](http://www.iX.de/digital-testen)

[leserservice@heise.de](mailto:leserservice@heise.de)

**49 (0)541 800 09 120**



# Genialer Lehrmeister

## Projektideen für junge Raspi-Programmierer

**Der Raspberry Pi ermöglicht Schülern einen kostengünstigen Einstieg ins Programmieren. Die Raspberry Pi Foundation stellt jede Menge Material zum Download bereit, doch deutschsprachiges Unterrichtsmaterial muss man länger suchen. Eine Übersicht über Blogs, Videokurse und Code-Editoren.**

Von Dorothee Wiegand

**D**er Brite Eben Upton ist Miterfinder des ersten Raspis und Mitbegründer der Raspberry Pi Foundation. Upton lernte als Schüler an einem BBC Micro programmieren und studierte nach einem Bachelor in Physik und Ingenieurwissenschaften an der Cambridge University Informatik. Als Director of Studies für das Fach Informatik am St. John's College in Cambridge war er zuständig für die Zulassung von Studenten. In dieser Zeit stellte er fest, dass sich immer weniger Schulabsolventen für ein Informatikstudium bewarben. Zudem brachten diese Bewerber häufig nur geringe Vorkenntnisse mit.

„Der Mangel an programmierbarer Hardware für Kinder – die Art von Hardware, die wir in den Achtzigerjahren hatten – wirkte sich negativ auf die Zahl der 18-Jährigen aus, die programmieren können“, erklärte Upton 2012 im Interview mit der Zeitschrift „Linux User & Developer“. Das sei ein Problem für die Universitäten und schließlich auch für die Industrie. 2008 gründet er mit Kollegen die Raspberry Pi Foundation. Ihre Idee war es, mit einem handlichen und vor allem erschwinglichen Minirechner für Schüler etwas für die Informatikkenntnisse von Kindern und Jugendlichen zu tun.

Upton hatte während seiner eigenen Schulzeit erlebt, dass längst nicht jeder Mitschüler Zugang zu einem Computer hatte. Chancengerechtigkeit war ihm wichtig: „Ich war in den Achtzigern enorm privilegiert. Ich hatte Zugang zu Bildung und Computern. Anderen blieb das verwehrt. Heute haben wir die Gelegenheit, es besser zu machen“, erklärte er 2016 in einem Interview mit dem Centre for Computing History in Cambridge.

Diesem Anliegen kam sicherlich zugute, dass parallel zur Einführung der ersten Raspi-Modelle die Schulen in Großbritannien große Anstrengungen unternahmen, um sowohl Medienkunde als auch Informatik zeitgemäß zu vermitteln. Das 2008 gegründete Projekt Computing at School (CAS) war maßgeblich an dieser Neugestaltung des Informatikunterrichts an britischen Schulen beteiligt. Der Raspberry Pi spielte dabei von Anfang an eine wichtige Rolle.

Im deutschsprachigen Bereich ist der Mini-Rechner längst nicht so verbreitet. Wir haben die Bildungsserver der deutschen Bundesländer und vergleichbare offizielle Websites in Österreich und der Schweiz nach dem Stichwort „Raspberry Pi“ durchsucht – leider mit mäßigem Erfolg. Nur ganz vereinzelt fanden sich dort Unterrichtsvorschläge für den Einsatz des Raspi in der Schule.

## Gesucht: Gute Ideen

Wer für Kinder, Enkel oder Patenkinder auf der Suche nach einem Einstieg in die Raspi-Welt ist, freut sich über spannende Projekte, die das Interesse der künftigen Coder wecken. Damit der Raspi den Unterricht bereichert, sollte das geplante Projekt zum Lehrplan passen. Lehrer, die selbst noch nicht allzu viel Erfahrung mit dem Raspi haben, profitieren von Projektvorschlägen, zu denen es eine ausführliche Dokumentation sowie vorbereitetes Unterrichtsmaterial gibt.

Tobias Hübner, Lehrer für Deutsch und Religionslehre, unterrichtet am Luisen-Gymnasium in Düsseldorf. Seit mittlerweile mehr als 15 Jahren setzt er sich dort auch für die Informatikausbildung ein. Dabei nutzt er gern günstige Hard- und Software wie den Raspberry Pi und Open-Source-Programme. Hübner möchte seine Schüler dazu befähigen, Software nicht nur anzuwenden, sondern selbst zu produzieren. Der Raspberry Pi ist für ihn ein Gegenentwurf zu dem in Schulen so

verbreiteten iPad: ein offenes System anstelle einer Black Box.

Auf YouTube hat Hübner unter dem Titel „**Raspberry Pi – Der Video-Kurs für Entdecker und Bastler**“ eine Videoreihe zum Raspi-Einstieg veröffentlicht, die keinerlei Vorkenntnisse voraussetzt. Darin geht es zunächst um Grundlagen wie Aufbau und kompatible Betriebssysteme, später um den Einsatz des Mini-Rechners als Foto- und Videokamera oder als Retro-Spielkonsole.

Auf seiner Website [medienistik.de](http://medienistik.de) veröffentlicht Hübner Lern- und Lehrmaterialien rund ums Unterrichten mit dem Raspberry Pi unter einer offenen Lizenz. Die Website bietet viel Inspiration und macht Lust aufs Experimentieren. Neben Unterrichtsentwürfen für das Fach Informatik stellt Hübner auch Raspi-Projekte für Deutsch, Englisch, Musik oder Ethik vor. Von einem Harry-Potter-Chatbot über ein selbstgebautes digitales Schlagzeug sind viele inspirierende Ideen dabei. Für Fortgeschrittene gibt es auch eine Anleitung, um das Computerspiel Minecraft mit einem Python-Programm auszubauen. Den Link zu Tobias Hübners Website und allen weiteren in diesem Artikel genannten Quellen finden Sie unter ct.de/yq4k.

## Einfach anfangen!

Eine weitere gute Anlaufstelle ist die Website **Tutorials for Raspberry Pi** des Informatikers Felix Stern aus Heidelberg. Er habe sein Blog vor gut zehn Jahren gestartet, weil es noch kaum öffentlich zugängliche Anleitungen auf Deutsch gab, berichtet Stern. „Daneben hat mir die Bastelei und das Coden am Raspberry Pi auch einfach unglaublich viel Spaß gemacht“, erklärt der Informatiker. Sein Rat an Einsteiger? „Einfach anfangen!“, am besten mit etwas Simplem, beispielsweise einer

Ampelschaltung. Der Code dafür sei überschaubar, aber der Lerneffekt vor allem am Anfang immens. „Und danach kitzelt es in den Fingern.“

Die Website von Reinhard Rahner namens **Unterrichts- und Lernmaterial für Mikrocontroller** ist stärker auf didaktische Aspekte ausgerichtet. Der pensionierte Schulleiter eines Gymnasiums in Schleswig-Holstein unterrichtete Mathematik, Physik und Informatik. Er habe 2015 angefangen, seine Website aufzubauen, berichtet Rahner. „Mir fiel auf, dass man im Internet viele englische, aber wenige deutsche Artikel über Mikrocontroller und Kleincomputer findet, die interessierte Menschen jeden Alters mit Hilfe einfacher Experimente in die Faszination dieser Technik einführen“, erklärt er gegenüber c't auf die Frage nach seiner Motivation.

Anfangs nutzte Rahner in seinem Unterricht Hardware des US-amerikanischen Anbieters Parallax. „Ergänzt wurde das Programm der Firma Parallax durch immer besser werdende pädagogisch-didaktisch aufbereitete Handbücher, die man als PDF kostenfrei herunterladen konnte“, berichtet Rahner. Dieses Material bildete später die Grundlage beim Aufbau seiner eigenen Website.

Auch bei kommerziellen Anbietern wird man mitunter fündig. Auf der **Website des Onlinehändlers BerryBase** gibt es unter anderem einen Blogbeitrag zum Thema „Lernen mit dem Raspberry Pi“. **Silicon Valley Kids** heißt die Website des Berliner Bildungsanbieters Mentorium. Neben vielen kostenpflichtigen Kursen, etwa zu den Programmiersprachen Scratch und Python oder zu Robotik, gibt es einige frei verfügbare Tutorials und FAQs. Die Kollegen von **t3n** haben auf ihrer Website eine Liste mit 30 Raspi-Projekten zusammengetragen, für deren



**Tobias Hübner, selbst Lehrer am Gymnasium, ist Raspi-Fan. Er teilt Grundlagenwissen und Projektanleitungen auf seiner Website und in YouTube-Videos.**

The screenshot shows the homepage of [TUTORIALS FOR RASPBERRY PI](https://tutorials-for-raspberry-pi.de/). It features a navigation bar with links to Einstieg, Hardware & GPIO, Software, Projekte, Sonstiges, Hausautomation, and social media icons. Below the navigation are three project cards: "Automatisches Raspberry Pi Gewächshaus selber bauen" (with an image of tomatoes), "Raspberry Pi: WS2801 RGB LED Streifen anschließen und steuern" (with an image of a glowing LED strip), and "Raspberry Pi Funksteckdosen (433MHz) steuern – Tutorial" (with an image of a power strip). A sidebar on the left contains a link to "Was ist der Raspberry Pi?" and a section titled "Entfernung messen mit Ultraschallsensor" with an icon of a sensor.

**Auf seinem Blog stellt der Heidelberger Felix Stern anspruchsvolle Smart-Home-Projekte mit dem Raspi vor. In der Kategorie „Erste Schritte“ finden Einsteiger hilfreiche Tipps und Tricks sowie erste Bastelideen.**

Nachbau Kinder aber sicher etwas Unterstützung benötigen.

## Fundgrube Raspberry Pi Foundation

Die Raspberry Pi Foundation stellte 2014 einige Entwickler und Lehrer aus der Raspi-Community ein, um Lernmaterial für Schulen zu entwickeln. Außerdem startete die Foundation Kurse für Lehrer, die den in Großbritannien gerade neu entwickelten Lehrplan für Informatik mit dem Raspberry Pi umsetzen wollten. Während Herstellung und Vertrieb der Mini-Rechner inzwischen bei der kommerziellen

Raspberry Pi Ltd liegen, ist die Raspberry Pi Foundation in Großbritannien als gemeinnützig anerkannt und engagiert sich weiterhin für die Informatik- und Programmierausbildung.

Manche Projekte der ersten Stunde sind zwischenzeitlich wieder eingeschlafen, andere Initiativen haben weltweit viele Fans gefunden. Auf der von der Raspberry Pi Foundation entwickelten Plattform **Code Club World** finden Eltern und Lehrer ein Kursprogramm mit aufeinander aufbauenden Übungen zu vielen Themen. Zu Beginn entwerfen Kinder ihren eigenen Avatar, lassen ihn tanzen und

spielen dazu Musik. In diesen ersten Übungseinheiten verwenden die kleinen Coder die speziell für Kinder entwickelte, blockbasierte Programmiersprache Scratch, später machen sie mit Python weiter.

Die Scratch Foundation rät dazu, dass Kinder mit acht Jahren ins Block-Coding einsteigen. Auch für die Teilnahme am Code Club World empfiehlt die Raspberry Pi Foundation ein Mindestalter von acht Jahren. An der Uni in Cambridge gibt es ein **Raspberry Pi Computing Education Research Centre**. Dessen Forscher begleiten Unterricht mit dem Raspi wissenschaftlich, erforschen beispielsweise das Debugging-Verhalten Jugendlicher, bieten auf ihrer Website aber auch eigene Unterrichtsmaterialien an.

Bei der **Astro Pi Challenge** dreht sich alles um Programme, die später auf Raspberry Pis an Bord der Internationalen Raumstation ISS laufen sollen. Die jährliche Challenge für Kinder und Jugendliche ist eine Kooperation zwischen der European Space Agency (ESA) und der Foundation. Der erste Wettbewerb dieser Art fand 2015/16 in Großbritannien statt. Die ISS hatte dafür zwei „Astro Pi“ an Bord – zwei zur damaligen Zeit aktuelle Modelle, die samt einer ganzen Reihe von Sensoren in einem speziellen Gehäuse steckten.

Weil die Idee gut ankam, weitete die ESA den Wettbewerb in den folgenden Jahren auf alle europäischen Länder aus, die an der ESA beteiligt sind. Die erste europäische Astro Pi Challenge lief im Schuljahr 2016/17. Interessierte Schüler-teams bewarben sich mit einer Idee für ein eigenes Forschungsprojekt. Das durfte ausschließlich den Astro Pi und seine Sensoren verwenden. Den für das Projekt nötigen Code sollten die Schüler später in Python schreiben und einreichen. Alle zur Teilnahme ausgewählten Teams erhielten ein ESA Astro Pi Kit, um ihre Experimente durchzuführen und ihren Code zu testen.

## Mein Code im Weltall

Inzwischen gibt es zwei Varianten des Wettbewerbs. **Mission Zero** richtet sich an Einsteiger. Teilnehmen können einzelne Kinder oder Gruppen von bis zu vier Schülern bis 19 Jahre. Die Aufgabe ist nicht allzu herausfordernd: Die Kinder schreiben ein Programm, das den Farb- und Helligkeitssensor eines Astro Pi ausliest. Diese Daten steuern die Hintergrundfarbe für selbst-

The screenshot shows a page from Reinhard Rahner's website. At the top is a large image of a breadboard with a microcontroller and the text "Unterrichts- und Lernmaterial für Mikrocontroller". Below this is a list of resources:

- + RasPi und Propeller C Code
- + RasPi und Propeller SPIN
- + RasPi mit Grove Base HAT
- + Blinklichter
- + Taster und Schalter
- + Der Sensor HC-SR04
- + Strom messen mit INA219

Below the list is a smaller image of a breadboard with a microcontroller and the text "Raspberry Pi - Python, C, SPIN Code". A note below it says "Das alles finden Sie in diesem Register und jeden Monat ein wenig mehr;-)".

**Die Website von Reinhard Rahner vermittelt Grundlagen rund um Mikrocontroller, insbesondere deren Programmierung. Seine Schritt-für-Schritt-Anleitungen eignen sich für ältere Schüler.**

gestaltete Pixelbildchen, die den Astronauten durch das Programm angezeigt werden. Laut Website schaffen Schüler diese Aufgabe in etwa einer Stunde. Der Code lässt sich auf jedem Computer mit Internetzugang im Browser schreiben und testen.

Nachwuchsprogrammierer, die die Teilnahmekriterien erfüllen und deren Einreichung den Vorgaben entspricht, dürfen ihr Programm bis zu 30 Sekunden lang im Weltraum laufen lassen. Sie erhalten ein personalisiertes Zertifikat, das die genaue Start- und Endzeit ihres Programms sowie die Position der ISS während der Ausführung angibt.

Die zweite Variante des Wettbewerbs namens **Mission Space Lab** ist anspruchsvoller: Jugendliche sind eingeladen, Programme zu schreiben, mit denen sich die Geschwindigkeit der ISS so genau wie möglich messen lässt. Mit Hilfe der Sensoren und der Kamera des Astro Pi sollen sie Daten über Ausrichtung und Bewegung der ISS sammeln, während diese die Erde umkreist. Teilnehmen können Teams aus zwei bis sechs Schülern wiederum bis 19 Jahre, die von einem Betreuer angeleitet werden. Der Startschuss zum aktuellen Wettbewerb fiel für beide Varianten der Challenge am 19. September 2024. Bis zum 24. Februar 2025 können Lehrer oder Betreuer nun Schülergruppen zur Teilnahme bei der Raspberry Pi Foundation anmelden.

Den Wettbewerb **Coolest Projects** richtet die Raspberry Pi Foundation in England, Irland, den USA und Indien als Vor-Ort-Veranstaltung aus. Als weltweites Online-Angebot an Kinder und Jugendliche führt die Foundation einen **Coolest Pro-**

The screenshot shows the homepage of the European Astro Pi Challenge. At the top, there are logos for Raspberry Pi Foundation and esa. The title "ASTRO PI" is prominently displayed with a small icon. Below the title, the text "The European Astro Pi Challenge 2024/25" is shown in large, bold letters. A subtitle "Giving young people the chance to run their computer programs in space" follows. A blue "Launch" button is visible. To the right of the text, there is a photograph of the Astro Pi hardware, which is a silver module with various ports and a small screen displaying a grid of colored squares.

**Teilnehmer der Astro Pi Challenge entwickeln Code, der später auf der internationalen Raumstation ISS auf einem Raspberry Pi ausgeführt wird.**

jects online showcase durch: Kinder und Jugendliche bis 18 Jahre präsentieren ihre digitalen Kreationen in einer Online-Galerie und können per Livestream bei der Abschlussfeier des Wettbewerbs dabei sein. Jeder Teilnehmer erhält eine Urkunde und digitale Fanartikel. Wer mitmachen möchte, kann sich für den Wettbewerb im kommenden Jahr ab dem 14. Januar und bis zum 28. Mai 2025 registrieren; für Kinder bis 13 Jahre sollte ein Erwachsener die Registrierung übernehmen.

### Wie sag ich's meinem Raspi?

Bevor es ans Coden geht, steht die Wahl der Programmiersprache an. Welche Sprache sich aus pädagogischer Sicht am ehesten für den Einstieg ins Programmieren eignet, ist Gegenstand vieler Diskussionen unter Lehrern und Didaktikern und wird

wissenschaftlich intensiv untersucht. Raspberry Pi OS, das offizielle Betriebssystem des Raspberry Pi, bringt einige vorinstallierte Programmierumgebungen mit, weitere lassen sich einrichten. In Schulen und Unterricht sind auf dem Raspi vermutlich **Scratch** und **Python** am häufigsten im Einsatz.

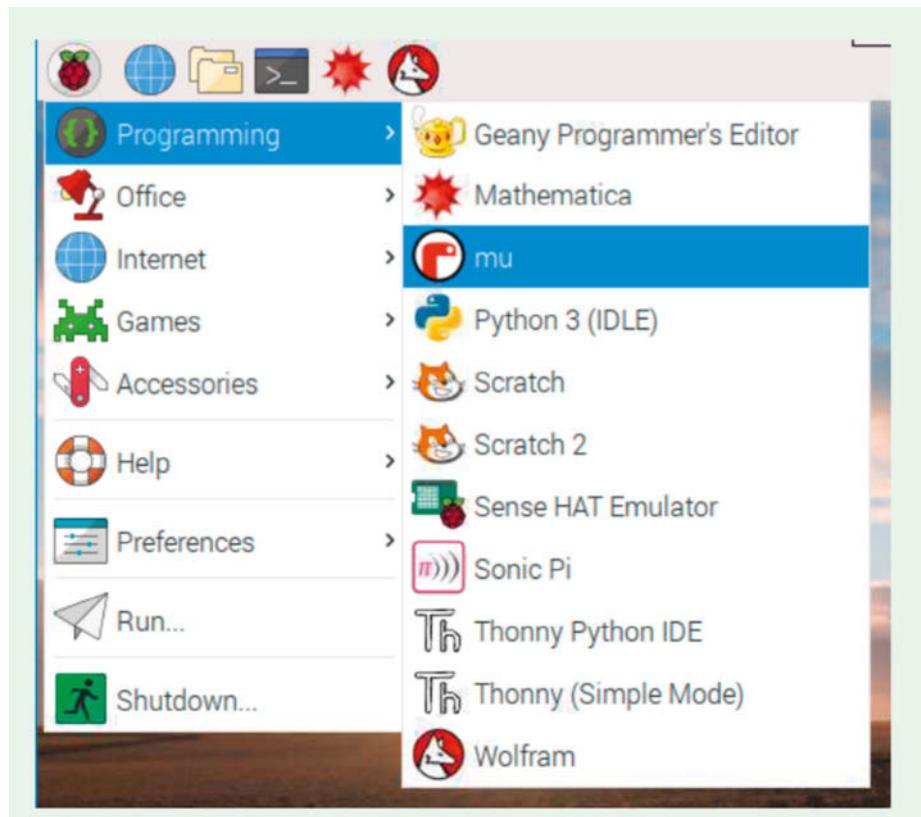
Scratch, das Forscher am Massachusetts Institute of Technology (MIT) entwickelten, gehört zu den sogenannten blockbasierten, visuellen Programmiersprachen. Anstelle von Codezeilen besteht ein Scratch-Programm aus untereinander angeordneten bunten Blöcken mit einer fest vorgegebenen Beschriftung. Für Werte oder Variablen haben die Blöcke Dropdown-Felder. Kleine Aus- und Einbuchtungen am Rand der Blöcke sollen garantieren, dass sich die Blöcke nur in korrekter Reihenfolge zusammenbauen lassen und am Ende die Syntax stimmt.

Der quietschbunte Scratch-Editor bringt Werkzeuge zum Malen und für Töne mit. Es gibt Comicfiguren, die sich durch Scratch-Befehle gesteuert auf einer Bühne bewegen. Kinder können damit kleine Simulationen erschaffen und interaktive Geschichten erzählen. Im Zusammenspiel mit dem Raspi bietet das Block-Coding durchaus auch fortgeschrittenere Möglichkeiten, zum Beispiel kann man auch die GPIO-Schnittstelle ansteuern. Derzeit ist Scratch 3 aktuell, für das man mindestens einen Raspberry Pi 4 benötigt.

Block-Programmiersprachen eignen sich frühestens ab einem Alter von sechs Jahren. Für die zum Lernen entwickelten

The screenshot shows the homepage of the Coolest Projects website. At the top, there are links for "Sign up" and "Log in". Below that, there is a navigation bar with "Take part", "Showcase", "Judges", "Guidance for mentors", "News", "Help", and a search icon. The main heading "THE GLOBAL DIGITAL SHOWCASE FOR KIDS" is displayed in large, bold letters. A subtext "Giving young people the chance to share the amazing things they make with digital technology." is present. At the bottom, there is a photograph of a young boy holding a small robot or model, standing in front of a booth with a "WELCOME" sign.

**Beim Wettbewerb namens Coolest Projects präsentieren junge Tüftler ihre Raspi-Projekte online vor anderen Raspi-Fans in der ganzen Welt.**



Nach Einrichtung des Betriebssystems stehen schon einige einsteigerfreundliche Programmierumgebungen zur Verfügung. Etliche weitere lassen sich nachträglich installieren.

Mikrocontroller wie BBC micro:bit und Calliope Mini gibt es eine ganze Reihe dieser Sprachen, beispielsweise MakeCode von Microsoft. Deren Block-Editoren lassen sich meist auch auf einem Raspberry Pi Zero nutzen. Eine weitere Block-Programmiersprache, die unter Raspberry Pi OS nutzbar ist: **EduBlocks**, ein kostenloses Angebot des britischen Softwarehauses Anaconda.

### Umstieg auf Python

Für die ersten Versuche in Python bietet sich der übersichtliche Code-Editor **mu** an, der sich speziell an Programmieranfänger richtet. Zwei weitere Editoren für Python, **Thonny** und **Geany**, sind ebenfalls in Raspberry Pi OS enthalten.

Ganz neu ist **Strype** vom Computing Education Research Centre am King's College in London. Strype ist ein Frame-basierter Editor für Python, der Schülern den Übergang vom Block-Coding erleichtern soll. Die Idee: Die Programmierumgebung kombiniert die Vorteile von block-basierten Editoren, insbesondere die Fehlervermeidung, mit der Flexibilität und Komplexität der textbasierten Program-

mierung. Strype verwendet dazu sogenannte Frames, die in ihren visuellen und interaktiven Eigenschaften den Blöcken ähneln, aber in bestimmten Slots die Eingabe von Text ermöglichen. Sie sollen einerseits Syntaxfehler vermeiden, andererseits aber auch fortgeschrittene Programmstrukturen erlauben.

Professor Michael Kölling leitet das Computing Education Research Centre am King's College. An der University of Kent entwickelte er bereits zwei speziell für Schule und Ausbildung konzipierte Programmierumgebungen für den Einstieg in Java: **Greenfoot** und **BlueJ**. Beide richten sich an Lerner ab etwa 14 Jahren. Greenfoot ist sowohl für den Einsatz im klassischen Unterricht als auch zum eigenständigen Lernen gedacht. Es soll vor allem den Spaß am Programmieren vermitteln und unterstützt den Anwender stärker, beispielsweise durch automatisch eingefügten Code. BlueJ soll den Lerner gründlich in Grundlagen und Prinzipien der objekt-orientierten Programmierung einführen, nimmt ihn weniger stark an die Hand und erlaubt vielfältigere Arten von Programmen.

Videospiele und Musik spielen im Leben der meisten Jugendlichen eine wichtige Rolle. Raspi-Projekte, die sich diesen Themen widmen, motivieren dazu, beim Programmieren dranzubleiben. **Sonic Pi** ist eine Programmierumgebung, mit der Schüler unter anderem auf dem Raspi ihre eigene elektronische Musik schreiben. Die Open-Source-Software haben Informatiker an der University of Cambridge entwickelt.

Die Raspberry Pi Foundation bietet einen Kurs zur Spieleentwicklung mit **Unity** an. Auf YouTube gibt es außerdem ein ausführliches Video, das zeigt, wie mit der Entwicklungsumgebung **GameMaker** auf einem Raspberry Pi 5 kleine Spiele entstehen.

### Nadeln im Heuhaufen

Obwohl oder vielleicht sogar weil es so viele unterschiedliche pädagogische Initiativen, Blogs und Videoanleitungen rund um den Raspberry Pi gibt, ist es gar nicht so leicht, das Passende für sich, seine Kinder oder seine Lerngruppe zu finden. Um Kinder für den Minirechner zu begeistern, braucht man ein Projekt, das sie inhaltlich anspricht, und eine Herangehensweise, die auf Alter und Vorwissen der Programmierer einsteiger abgestimmt ist. Ein zentraler Aspekt dabei ist die Frage nach der passenden Programmiersprache und -umgebung.

Leider finden sich im Web nicht viele deutschsprachige Anleitungen und Ideen rund ums Lernen mit dem Raspi. Insbesondere fehlt Material, das an die Informatiklehrpläne der Bundesländer angepasst ist und Lehrern konkret bei der Unterrichtsvorbereitung helfen könnte. Umso erfreulicher, dass einige engagierte Privatpersonen interessante Ideen auf ihren Blogs präsentieren und Material zum Download anbieten.

Für Lehrer, Eltern, Großeltern und alle, die mit Kindern und Jugendlichen den Raspberry Pi entdecken wollen, empfiehlt sich auch ein Blick auf das große Angebot der Raspberry Pi Foundation. Andere englischsprachige Quellen haben ebenfalls tolle kindgerechte Projektideen und didaktisches Begleitmaterial zu bieten. Die Sprache mag anfangs eine kleine Hürde darstellen, doch die weltweite englischsprachige Raspi-Community hat so viel zu bieten, dass sich der Blick über den Tellerrand auf jeden Fall lohnt. (dwi@ct.de) **ct**

**Blogs, Videos, Programmierumgebungen:** [ct.de/yq4k](http://ct.de/yq4k)

betterCode()

# GenAI 2025

Die Konferenz zu KI-gestützter Softwareentwicklung

21. Januar 2025 • Online

KI-Tools wie GitHub Copilot können mehr als Boilerplate-Code erstellen.  
Bring deine Projekt-Entwicklung aufs nächste Level:

- ✓ Status quo und Zukunft der KI-gestützten Entwicklung
- ✓ Die leistungsfähigsten KI-Tools im Überblick
- ✓ Schnelles Feedback durch Code Reviews mit GenAI
- ✓ Bessere Testergebnisse durch Einsatz von KI
- ✓ Altsysteme modernisieren: LLMs unterstützen Migration
- ✓ Security: Risiken und Chancen von KI

Jetzt  
Frühbucher-  
tickets  
sichern!

Workshop am 28. Januar: »Coding mit Co-Piloten: KI-Tools in der Softwareentwicklung«

[genai.bettercode.eu](http://genai.bettercode.eu)

Veranstalter



dpunkt.verlag

# M3 Minds Mastering Machines

## Die Konferenz für Machine Learning und KI

20.-21. Mai 2025 • Karlsruhe

Der Treffpunkt für Data Scientists, Data Engineers und Developer, die ML-Projekte in die Realität umsetzen

Call for  
Proposals  
bis 28.11.

Jetzt Vorschläge für Vorträge und Workshops einreichen

[m3-konferenz.de](http://m3-konferenz.de)

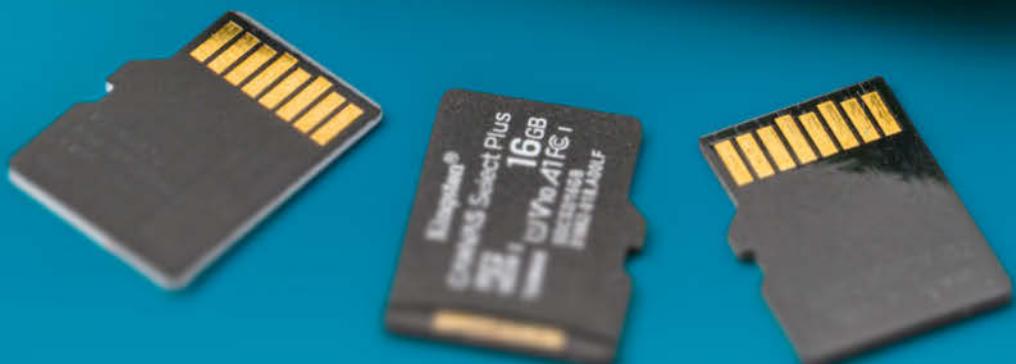
Veranstalter



dpunkt.verlag

# Plug'n'Pi

Fertige Projekte für den Raspberry Pi



<b>Raspi-Kubernetes-Cluster .....</b>	<b>Seite 149</b>
<b>Outdoor-PoE-Kamera mit Raspberry Pi .....</b>	<b>Seite 154</b>
<b>Raspi als Boots-Navigationshilfe .....</b>	<b>Seite 160</b>

# Der Raspi staubt ungenutzt in der Schublade vor sich hin, weil Zeit oder Geld für aufwendigere Projekte knapp ist? Mit dem Bastelcomputer lässt sich auch ohne ausgeprägten Basteltrieb Spaß haben. Wir stellen drei fertig einsatzbereite Images vor, die Ihnen (fast) alles abnehmen.

Von Niklas Dierking

**D**en Raspberry Pi mit Sensoren und Mikrocontrollern verdrahten, um dann eine einzige Zimmerpflanze zu gießen? Viele Projekte, über die man als Raspi-Neuling stolpert, können das Zeit- und Bankkonto strapazieren. Lesen Sie in diesem Artikel, wie sie eine Nummer kleiner anfangen. Komplexere Projekte finden Sie in den nachfolgenden Artikeln.

Man muss nicht zwingend Zubehör anschaffen oder in die Tiefen von Raspberry Pi OS, der geläufigsten Linux-Distribution für den Raspi, abtauchen und Konfigurationsdateien schrubbren. Längst gibt es eine ganze Reihe von ausgereiften Spezialdistributionen für den Raspi, die einem viel Arbeit abnehmen. Um mit den Projekten in diesem Artikel loszulegen, genügt eine microSD-Karte und ein Rechner, um sie zu beschreiben. Denn um nützliche oder spaßige Dinge mit dem Raspi anzustellen, braucht es kaum mehr als den Bastelcomputer selbst.

## Instant-Cloud für den Raspi: NextcloudPi

Nextcloud als persönlichen Cloudspeicher auf eigener Hardware zu betreiben, ist gar nicht so einfach. Der freien Cloud haben wir erst kürzlich in c't eine Artikelreihe mit einer Installationsanleitung gewidmet, die sich für die meisten Szenarien eignet [1]. Der Raspberry Pi eignet sich nicht unbedingt für alle Anwendungsfälle einer Nextcloud und Performance-Wunder sollte man nicht erwarten. Für die persönliche Dateiallage oder die eines kleinen Teams sollte es aber reichen. Ein Raspi der dritten

oder besser vierten Generation sollte es schon sein.

Wir empfehlen außerdem, das Datenverzeichnis auf einen flotten externen Datenträger wie eine USB-SSD auszulagern, aber dazu gleich mehr. Performance-König ist wenig überraschend der Raspberry Pi 5 mit einer NVMe-SSD, die auf einem Erweiterungsboard (HAT) sitzt. Wer so viel Rechen- und Speicherleistung braucht, sollte sich aber die Frage stellen, ob er mit einem Mini-PC, beispielsweise mit Intel-N100-CPU, nicht besser beraten ist. Der spielt in Sachen Stromverbrauch und Leistung in einer ähnlichen Liga, ist aber teilweise günstiger zu haben.

Wer sich für Nextcloud auf dem Raspi entschieden hat, darf sich aber über NextcloudPi freuen. Das von der Community betriebene Projekt bietet eine Nextcloud-Installation als fertiges Image an, das man nur auf ein Startmedium wie eine microSD-Karte schreiben muss, um loszulegen. NextcloudPi nimmt einigen einige Konfigurationsaufgaben komplett ab und vereinfacht andere durch eine weitere Weboberfläche, zusätzlich zu der eigentlichen Nextcloud-Bedienschnittstelle. Ausflüge auf die Kommandozeile sollten nur noch in Ausnahmefällen nötig sein.

Das fertige Image laden Sie aus dem Release-Bereich des GitHub-Repository von NextcloudPi herunter (siehe Seite 116). Entpacken Sie das Archiv und schreiben das Abbild auf eine microSD-Karte. Das geht besonders einfach mit dem Raspberry Pi Imager (siehe ct.de/y5ff) und seiner Option „Use custom“ im Menü „OS wählen“. Alternativ ist es auch möglich, NextcloudPi mittels Skript in ein bereits eingerichtetes Raspberry Pi OS zu installieren. Nachdem man den Raspi via Ether-

net mit dem lokalen Netzwerk verbunden und mit Strom versorgt hat, erreicht man die NextcloudPi-Instanz im Browser unter der IP-Adresse des Raspi (beispielsweise <https://192.168.178.100>) und muss beim ersten Aufruf eine Ausnahme für das selbst signierte Zertifikat hinzufügen.

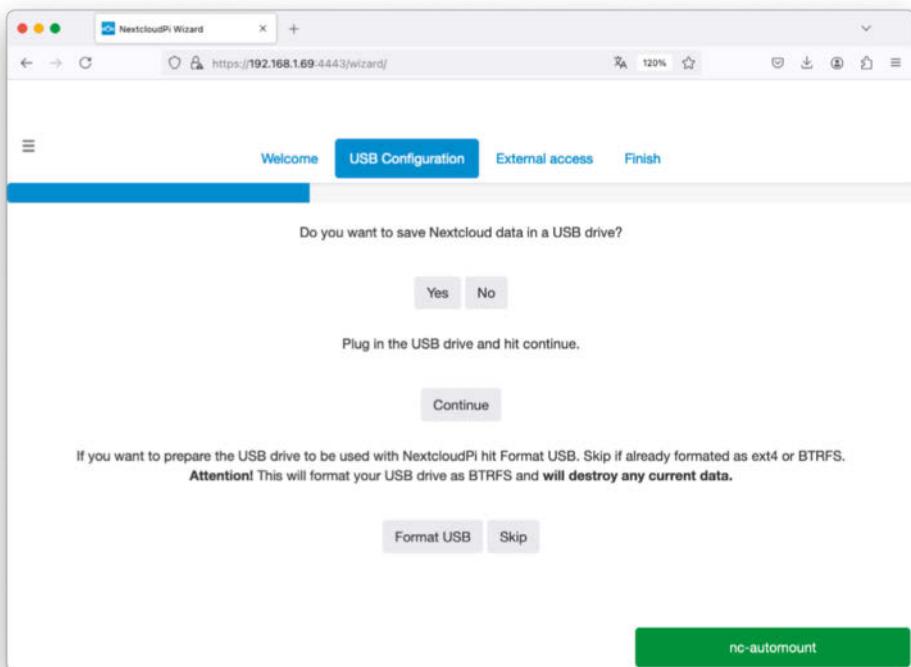
NextcloudPi zeigt jetzt zufällig generierte Zugangsdaten für Nextcloud und NextcloudPi, die Sie sicher speichern sollten. Wenn Sie auf die Schaltfläche „Activate“ klicken, werden Sie zur NextcloudPi-Weboberfläche weitergeleitet. Nachdem Sie die Zugangsdaten eingegeben haben, begrüßt Sie der Einrichtungsassistent. Über das Menü „USB Configuration“ können Sie mit wenigen Mausklicks ein USB-Laufwerk formatieren, das Nextcloud-Datenverzeichnis dorthin verschieben und konfigurieren, das Laufwerk beim Systemstart automatisch einzuhängen.

Gängige Methoden, um die Nextcloud-Instanz außerhalb des Heimnetzes erreichbar zu machen, sind eine Portweiterleitung am Router oder die Nextcloud hinter einem VPN wie WireGuard zu betreiben. NextcloudPi bietet dafür Integrationen von verbreiteten und kostenlosen DynDNS-Diensten wie DuckDNS. Wenn Ihr Internetprovider Ihnen eine neue öffentliche IP-Adresse zuweist, kümmert sich der Dienst darum, Ihre Domain, beispielsweise nextcloudpi.example.com automatisch der neuen Adresse zuzuweisen.

**Vorsicht:** Wenn Sie Ihre Nextcloud ins Internet hängen, sollten Sie unbedingt darauf achten, die Software stets auf dem aktuellen Stand zu halten. Das erledigen

## c't kompakt

- Es muss nicht immer Marke Eigenbau sein: Die hier vorgestellten Raspi-Projekte enthalten alles Nötige, sind besonders einsteigerfreundlich und führen schnell zum Ziel.
- Nachdem Sie das fertige Image auf eine microSD-Karte geschrieben haben, erledigen Sie alles Weitere über eine Weboberfläche oder textbasierte Menüs.
- Fertige Images sind eine gute Möglichkeit, den Raspi für verschiedene Aufgaben zu erproben, ohne viel Zeit in die Konfiguration zu stecken.



**Auf dem Raspi empfiehlt es sich, das Nextcloud-Datenverzeichnis von der microSD-Karte auf eine externe SSD zu verschieben. NextcloudPi führt mit einem Assistenten durch den Umzug.**

Sie bequem im Menü „Updates“ in der NextcloudPi-Oberfläche. Außerdem sollten Sie Ihren Admin-Account in den Nextcloud-Einstellungen im Bereich „Security“ mit einer Zwei-Faktor-Authentifizierung schützen. Richten Sie nur eine Portweiterleitung für die Nextcloud-Weboberfläche (TCP-Port 443) ein und nicht

für die NextcloudPi-Seiten (TCP-Port 4443).

Um ein TLS-Zertifikat von Let's Encrypt zu besorgen, muss zusätzlich TCP-Port 80 aus dem Internet erreichbar sein, um Ihre (DynDNS-)Domain mittels HTTP-Challenge zu verifizieren. Dann können Sie Let's Encrypt im Menü „Net-

working“ in der NextcloudPi-Oberfläche aktivieren. Tragen Sie Ihre (DynDNS-) Domain anschließend auch im Menü „nc-trusted domains“ ein, sonst verweigert Nextcloud den Zugriff. NextcloudPi kürzt die Installation von Nextcloud auf dem Raspi deutlich ab. Die Weboberfläche wirkt zwar etwas spartanisch, bietet aber viele nützliche Funktionen von Backups bis zum Fernzugriff, die dem Nextcloud-Admin das Leben leichter machen.

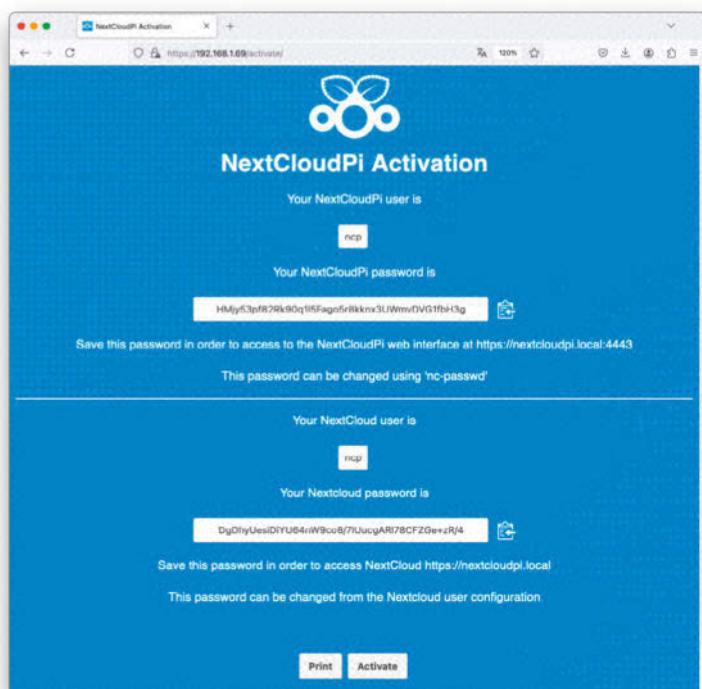
## Schmales Image, dicker Softwarekatalog: DietPi

Das DietPi-Projekt trägt eine grimmige Beere als Logo und verpasst Raspberry Pi OS eine Schlankheitskur. Es kommt wie Raspberry Pi OS Lite standardmäßig ohne grafische Oberfläche, geht aber einen Schritt weiter, indem es viele Pakete herauschmeißt, und röhmt sich damit, besonders wenig Arbeits- und Hauptspeicher zu beanspruchen. Auf einer frischen Installation laufen etwa gerade mal elf Prozesse, darunter Dropbear, eine abgespeckte Alternative zum OpenSSH-Server mit reduziertem Funktionsumfang. Logs schreibt DietPi in den Voreinstellungen nur in eine RAM-Disk, um die microSD-Karte zu schonen.

Durch den schmalen Fußabdruck eignet sich DietPi insbesondere für ältere Raspi-Modelle. Das Image für den Raspberry Pi 5 trägt noch das Label „testing“, für den produktiven Einsatz sollte man sich daher besser noch gedulden, auch wenn wir bei unserem Testlauf keine Probleme gefunden haben.

Die Passwörter der Benutzer `root` und `dietpi` sollten Sie beim ersten Login umgehend ändern, beispielsweise im Menü „Security“ von `dietpi-config`. Das ersetzt das bekannte `raspi-config`, enthält aber zusätzliche Tools, beispielsweise um angeschlossene USB-Laufwerke zu verwalten. Es gibt auch vorgefertigte Overclocking-Profile und Benchmarking-Werkzeuge, um sich der Stabilität des schwitzenden Raspi zu vergewissern, wenn Sie ihn übertaktet haben.

DietPi richtet sich an Nutzer, die Ihren Raspi als Mini-Heimserver einsetzen wollen, aber keine Lust haben, Dienste von Hand zu konfigurieren, oder die allergisch gegen Container sind. Das eigentliche Highlight von DietPi ist nämlich eine umfangreiche Sammlung an Bash-Skripten für die Installation von allen möglichen Anwendungen, die zum Self-Hosting-Portfolio gehören. Um sie alle zu listen,



**NextcloudPi bringt eine zusätzliche Weboberfläche, die es erleichtert, Nextcloud auf dem Raspi zu betreiben. Beim ersten Besuch generiert es zufällige Passwörter für den Nextcloud-Admin und für die Weboberfläche.**

fehlt hier der Platz, aber es ist von Medienspielern wie Plex oder Jellyfin über Torrent-Werkzeuge bis hin zu DNS-Werbeblockern wie Pi-hole oder Vaultwarden, einer Open-Source-Implementierung des Bitwarden-Servers, für alle etwas geboten. Den gesamten Softwarekatalog finden Sie in der DietPi-Dokumentation, die wir unter [ct.de/y5ff](http://ct.de/y5ff) verlinkt haben.

Besonders praktisch: Weil die Installationsskripte aufeinander abgestimmt sind, muss man sich beispielsweise nicht damit herumschlagen, dass Ports der Webserver kollidieren. In der Dokumentation finden Sie zu jeder Anwendung Antworten auf häufige Fragen: Wie erreiche ich die Anwendung mit dem Browser? In welchen Verzeichnissen liegen Daten und wichtige Konfigurationsdateien? Wie lese ich die Logs aus und wie bringe ich die Software auf den neuesten Stand?

Die eigentliche Installation erfolgt über textbasierte Menüs, die Sie mit dem Befehl `dietpi-software` aufrufen. Öffnen Sie zunächst den Katalog und setzen Sie Häkchen bei allen Anwendungen, die Sie installieren wollen. Danach rufen Sie den untersten Menüpunkt „Install“ auf und DietPi arbeitet die Liste der begehrten Anwendungen ab. Apps, die sie nicht mehr benötigen, sieben Sie über die Liste im Menü „Uninstall“ aus.

Bei der Installation von Software fragt DietPi nach, ob es Telemetriedaten erheben darf. Damit füttert das Projekt unter



**DietPi bringt viele gängige Anwendungen aus dem Homelab- und Heimserver-Bereich per Skript auf den Raspi.**

anderem die Rangliste der beliebtesten Software aus dem Katalog (siehe [ct.de/y5ff](http://ct.de/y5ff)). Die Daten werden nur erhoben, wenn Sie zustimmen. Wer plant, eine ganze Flotte von Raspis mit DietPi auszustatten, sollte einen Blick in den Abschnitt „automatisierte Installation“ der DietPi-Dokumentation werfen (siehe [ct.de/y5ff](http://ct.de/y5ff)). Obwohl DietPi zunächst für den Raspi entwickelt wurde, gibt es inzwischen auch Dutzende Images für Boards der Hersteller Odroid, Pine und Radxa, Installationsmedien für x86-PCs sowie Abbilder virtueller Maschinen für Hypervisor wie Proxmox.

## Retro-Gaming für den Raspi: Batocera

Retro-Gaming und die Emulation älterer Konsolen ist schon lange ein beliebter Einsatzzweck für den Raspi. Die Einrichtung geht so fix, dass zwischen dem Schreiben der microSD-Karte und dem nostalgischen Besuch im Pilzkönigreich und anderen Pixelwelten nur wenige Minuten liegen. Eines der ersten Projekte in diesem Bereich war RetroPie, dessen letztes Release allerdings schon zwei Jahre her ist. Ein Image für den Raspi 5 gibt es bislang nicht. Eine Umfrage unter den Raspi-Retro-Gamern in der Redaktion

## 2x Mac & i mit 35% Rabatt testen!

Mac & i – Das Magazin rund um Apple

- Tipps & Praxiswissen
- Hard- und Softwaretests
- Reports und Hintergründe

Für nur 16,80 € statt 25,80 € (Preis in Deutschland)



### + Geschenk nach Wahl

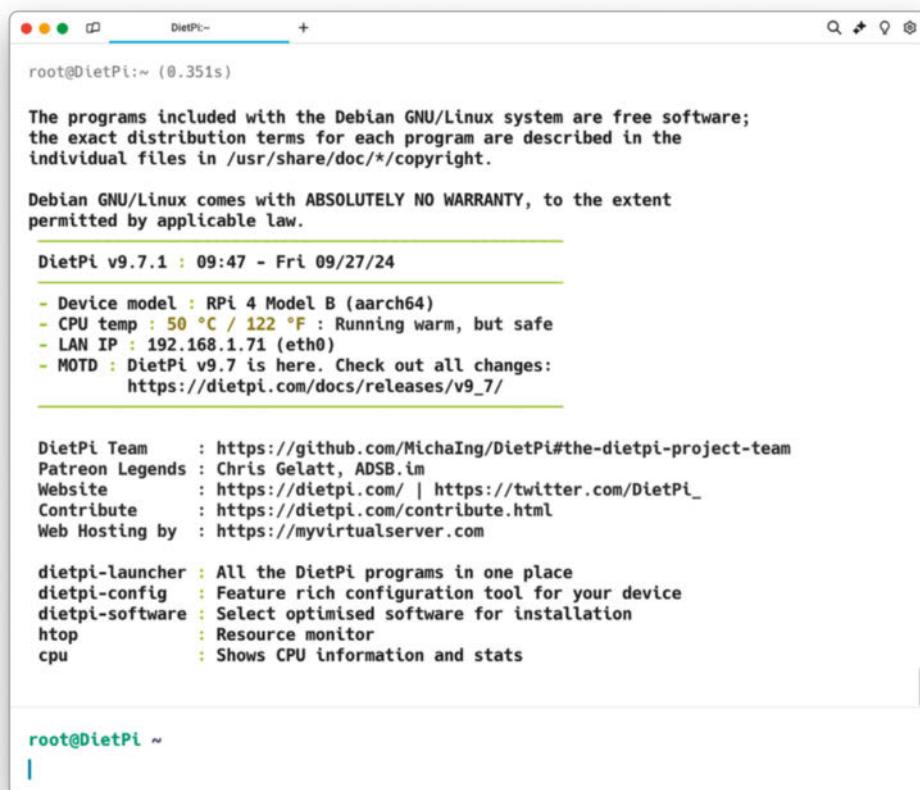
z. B. Kabelbox mit Adapters oder Buch Apple Junkies



Jetzt bestellen: [www.mac-and-i.de/vorteil](http://www.mac-and-i.de/vorteil)

Copyright by Heise Medien.

Mac & i. Das Apple-Magazin von c't.



```

root@DietPi:~ (0.351s)

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

DietPi v9.7.1 : 09:47 - Fri 09/27/24

- Device model : RPi 4 Model B (aarch64)
- CPU temp : 50 °C / 122 °F : Running warm, but safe
- LAN IP : 192.168.1.71 (eth0)
- MOTD : DietPi v9.7 is here. Check out all changes:
  https://dietpi.com/docs/releases/v9_7/

DietPi Team      : https://github.com/MichaIng/DietPi#the-dietpi-project-team
Patreon Legends : Chris Gelatt, ADSB.im
Website          : https://dietpi.com/ | https://twitter.com/DietPi_
Contribute       : https://dietpi.com/contribute.html
Web Hosting by   : https://myvirtualserver.com

dietpi-launcher : All the DietPi programs in one place
dietpi-config   : Feature rich configuration tool for your device
dietpi-software  : Select optimised software for installation
htop            : Resource monitor
cpu              : Shows CPU information and stats

root@DietPi ~
|
```

**Standardmäßig kommt DietPi ohne Desktop aus, erleichtert aber die Konfiguration mittels textbasierter Menüs.**

hat ergeben, dass Batocera inzwischen das Raspi-Retro-Gaming-Betriebssystem der Wahl ist.

Ähnlich wie bei RetroPie spannt Batocera für die Emulation von Heimkonolen und Handhelds unter anderem die Emulatoren des RetroArch-Projekts ein. Für die Organisation der ROMs und als Frontend kommt EmulationStation zum Einsatz. Wie man legal an ROMs kommt, erklärt c't-Kollege Jan-Keno Janssen in einem Video (siehe ct.de/y5ff). Das Image für Ihren Einplatinencomputer laden Sie im Downloadbereich der Batocera-Website herunter und schreiben es mit dem Raspberry Pi Imager auf ein Startmedium. Um den Raspi 4 oder 5 mit einem Monitor oder Fernseher mit HDMI-Anschluss zu verbinden, brauchen Sie ein Micro-HDMI-Kabel, das um die 3 Euro kostet und in vielen Raspberry-Pi-Sets enthalten ist.

Wenn Sie ROMs vor dem ersten Start hinzufügen möchten, kopieren Sie die in das Verzeichnis /share/roms auf der microSD-Karte. Damit Sie die Partition, die als ext4 formatiert ist, unter Windows und macOS lesen und beschreiben können, müssen Sie Software nachrüsten oder zum WSL (Windows Subsystem for Linux) greifen. In der ARM-Variante von

Batocera fehlt ein Dateimanager, deswegen dürfte es in den meisten Fällen leichter sein, die ROM-Dateien im laufenden Betrieb über das Netzwerk hinzuzufügen. Dafür betreibt Batocera eine Samba-Freigabe, die Sie auf einem Linux-System beispielsweise im Dateimanager Nautilus als smb://BATOCERA.local/share einbinden können. Unter Windows finden Sie das Netzlaufwerk unter \\BATOCERA\\share.



**Batocera ist ein Retro-Gaming-Komplettsystem. Mitbringen muss man nur Spiele, BIOS-Dateien und einen Controller.**

Welche Retro-Systeme Batocera flüssig emulieren kann, hängt von der Leistungsfähigkeit Ihres Einplatinencomputers ab. Auf einem Raspi 4 sind die 8- und 16-Bit Konsolengenerationen wie NES, Sega Master System, SNES und Sega Megadrive (Genesis) problemlos spielbar. Das gilt auch für viele 3D-Titel für die Plattformen PlayStation, Nintendo 64 und Sega Saturn. Der Raspi 5 packt sogar eine GameCube-Emulation und einige Wii-Titel. Einige Systeme, etwa die PlayStation, benötigen zusätzlich zu ROM-Dateien auch die korrekte BIOS-Datei im Verzeichnis /share/bios.

Batocera unterstützt eine große Anzahl von Controllern. Bluetooth-Controller fügen Sie im Menü „Controller“ und „Bluetooth Settings“ hinzu. Bei Controllern, die nicht Teil der Controller-Datenbank von Batocera sind, müssen Sie anschließend mit einem Assistenten die Tastenbelegung konfigurieren. In unserem Testlauf war es auch problemlos möglich, einen Xbox-One-Controller mit dem offiziellen 2,4-GHz-Adapter von Microsoft zu benutzen. Sollten Sie Probleme haben, ROMs hinzuzufügen oder Controller zu verbinden, empfiehlt sich ein Blick in das sehr ausführliche Batocera-Wiki, das wir unter ct.de/y5ff verlinkt haben. Viel Spaß beim Retro-Gaming und mit den folgenden Raspi-Projekten!

(ndi@ct.de) **ct**

## Literatur

- [1] Niklas Dierking, Die eigene Cloud wetterfest machen, Eine frische Nextcloud einrichten, c't 14/2024, S. 26

**Download der fertigen Images und Dokumentation der Projekte: [ct.de/y5ff](http://ct.de/y5ff)**

# Kubernetes Security

## Angriffe verhindern, Cluster schützen

Tauchen Sie in das Berechtigungssystem ein.  
Lernen Sie, wie Sie mit Network Policies  
Firewalls innerhalb der Cluster aufbauen.

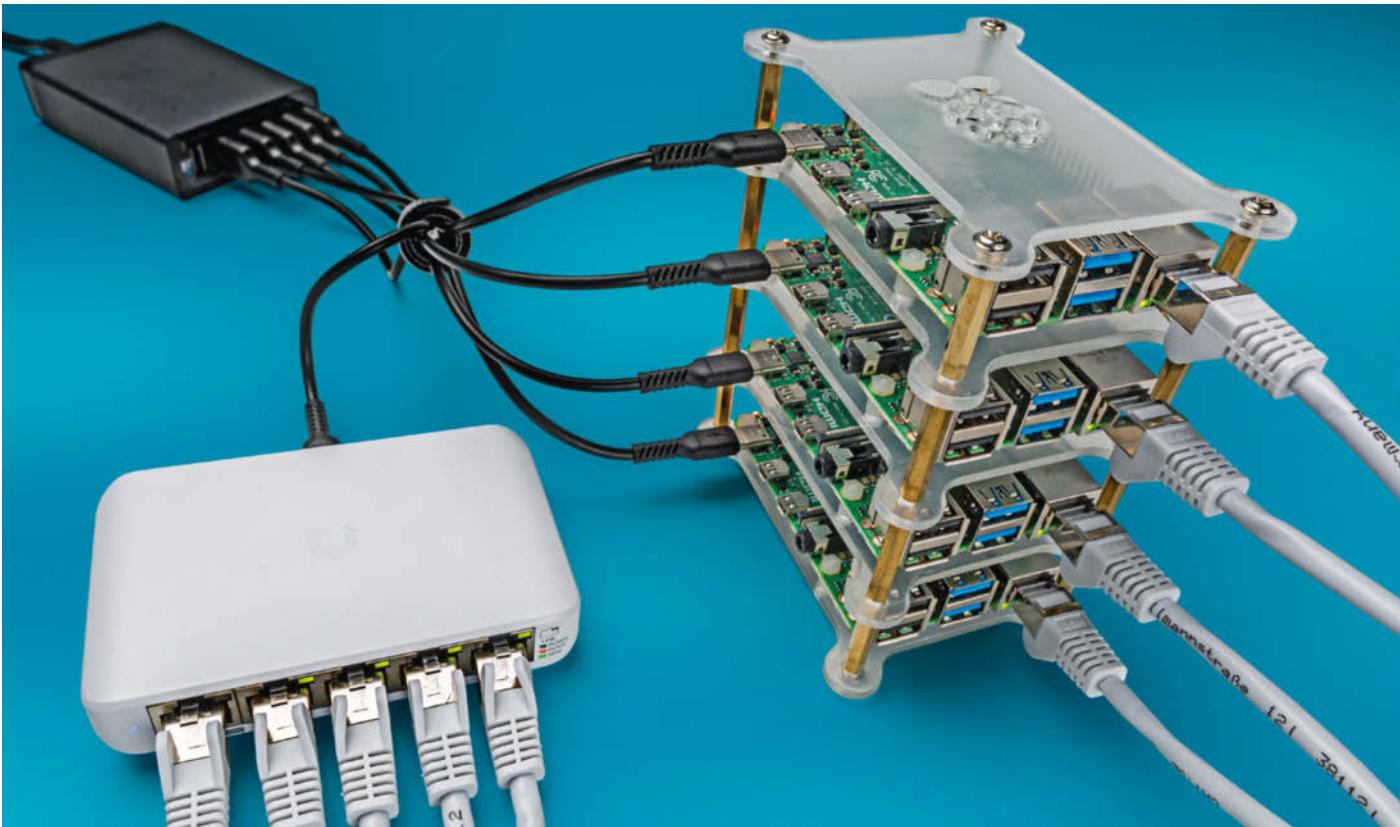
Start: 19. November

**CLASSROOM  
MIT FÜNF  
TERMINEN**



**Jetzt Ticket sichern:**

[heise-academy.de/classrooms/kubernetes-security-24](https://heise-academy.de/classrooms/kubernetes-security-24)



# Cluster-Beere

## Kubernetes-Cluster mit dem Raspberry Pi

**Der Einstieg in Kubernetes erscheint schwierig, aber es gibt Projekte als Starthilfe. Wir zeigen, wie Sie vier Raspis zu einem ausfallsicheren Cluster kombinieren. Der Automatisierungshelfer Ansible nimmt Ihnen dabei viel Arbeit ab.**

Von Niklas Dierking

**G**ründe für einen Kubernetes-Cluster im Homelab gibt es einige: Irgendwann wächst die Containerei vielleicht über Docker hinaus. Oder die Workloads sind so wichtig, dass sie nach Ausfallsicherheit verlangen. Oder man ist scharf auf einen Job, der Kubernetes-Fachwissen verlangt. Immerhin ist der Container-Orchestrator inzwischen zum Industriestandard geworden und schultert

die Workloads der großen Player, darunter Streamingdienste oder Telekommunikationsanbieter.

Warum also nicht den Raspi dafür einspannen? Anders als bei einer Flotte virtueller Maschinen auf einem einzelnen Host oder Hypervisor wie Proxmox, kann man so auch ausfallsichere Konfigurationen (High Availability) erproben. Mit echter Hardware kann man live erleben, wie der Cluster reagiert, wenn man einen Raspi ausknipst.

Beim Aufbau des Raspi-Clusters nehmen wir, Automatisierung und Vorarbeit der Open-Source-Community sei Dank, einige Abkürzungen, um das Projekt einfach zu halten. Dafür kommt auch die abgespeckte Kubernetes-Distribution k3s zum Einsatz. Das Projekt richtet sich an Leser, die nach Docker den nächsten Schritt gehen wollen und bereits Linux- und Container-Grundwissen mitbringen.

Eine Einführung in Kubernetes und sein Vokabular finden Sie in einer ausführlichen Artikelserie [1].

Der Cluster, den wir in diesem Artikel vorstellen, besteht aus vier Raspberry Pi 4 mit jeweils 4 GByte Arbeitsspeicher. Wir haben einen konfigurierbaren Unifi-Switch aus unserem Fundus verwendet, aber in diesem Aufbau genügt auch ein Exemplar aus der Restekiste, solange es genug Ports hat. Mit einem USB-C-Eingang für die Spannungsversorgung lässt der Switch sich gemeinsam mit den Raspis aus dem Netzteil speisen (Anker PowerPort 6).

Außerdem brauchen Sie einen weiteren Rechner im gleichen Netzwerk. Über den interagieren Sie mittels Ansible und SSH mit Raspberry Pi OS und der Kubernetes-Distribution k3s, später dann über das Kubernetes-API mit dem Cluster. Auf Ihrer lokalen Workstation müssen Sie Ansible, kubectl und git installieren. Ansible ist ein Werkzeug zur Automatisierung und zieht den Cluster hoch, über kubectl geben Sie Deployments, eine deklarative Beschreibung des gewünschten Zustands von Pods, in Auftrag. Bei einem Pod handelt es sich um einen oder mehrere Container, die sich Ressourcen wie Speicher und Netzwerk teilen.

**ct Hardcore**

## Software und Netzwerk vorbereiten

Unter macOS installieren Sie Ansible, kubectl und git am einfachsten mit dem Paketmanager homebrew. Dazu reicht der Befehl `brew install ansible kubectl git`. Auf einem Linux-Host installieren Sie die Werkzeuge mit dem Paketmanager ihrer Distribution, beispielsweise mit apt unter Ubuntu mit `sudo apt install ansible kubectl git`.

Damit ist die Software auf dem Kontroll-Computer schon vollständig. Als Betriebssystem für die Raspis haben wir uns für Raspberry Pi OS Lite in der 64-Bit-Variante entschieden. Sie schreiben es am einfachsten mit dem Raspberry Pi Imager auf ein Startmedium. Eine MicroSD-Karte genügt für diesen kurzlebigen Versuchsaufbau.

Dabei sollten Sie über die Imager-Einstellungen bereits einige Anpassungen vornehmen. Vergeben Sie Hostnamen, anhand derer Sie die Raspis identifizieren können. Wir haben uns für k3s-1 bis k3s-4 entschieden. Außerdem sollten Sie sich für einen einheitlichen Nutzernamen für alle Instanzen entscheiden und einen SSH-Schlüssel hinterlegen. Die WLAN-Einstellungen können Sie überspringen. Verbinden Sie die Raspis dann mit dem Netzwerk und der Stromversorgung.

In der Weboberfläche Ihres Routers sollten Sie für jeden Raspi eine feste IP-Adresse festlegen, damit die später nicht mit dem IP-Adressbereich kollidiert, den wir unter die Kontrolle von MetalLB stellen, aber dazu weiter unten mehr.

In unserem Aufbau haben wir für den Cluster ein eigenes Subnetz eingerichtet (192.168.3.0/24), damit dem Load Balancer, der später eingerichtet wird, möglichst viele IP-Adressen zur Verfügung stehen. Das ist für einen flüchtigen Versuchsaufbau aber nicht unbedingt nötig. Wir nutzen in diesem Testlauf nur IPv4, aber Kubernetes kann grundsätzlich auch IPv6. Die vier Raspi-Nodes tragen die folgenden Hostnamen und IP-Adressen:

k3s-1 = 192.168.3.101  
k3s-2 = 192.168.3.102  
k3s-3 = 192.168.3.103  
k3s-4 = 192.168.3.104

## Cluster-Architektur

Die eigentliche Installation und Konfiguration übernimmt das Automatisierungs-Werkzeug Ansible. Damit Sie trotzdem wissen, was im Cluster vor sich geht, wer-

fen wir einen Blick auf die Komponenten und die Architektur.

Die Wahl der Kubernetes-Distribution fällt auf k3s, weil es weniger Ressourcen als eine vollständige Kubernetes-Implementierung benötigt. Beim Raspi-Cluster handelt es sich um einen sogenannten Bare-Metal-Cluster, denn Kubernetes läuft ohne weitere Abstraktionsschichten direkt auf der Hardware. Die Architektur unterscheidet sich deswegen von Clustern im Rahmen eines „Managed-Kubernetes“-Angebots der großen Cloudprovider wie Amazon Web Services, Google Cloud und Microsoft Azure.

Die stellen Kubernetes einen Cloud-Load-Balancer zur Seite. Ein solcher weist Services mit dem Typ „Load Balancer“ eine IP-Adresse außerhalb des Cluster-Netzwerks zu, um Traffic zum gewünschten Pod zu routen. Kubernetes kennt mit NodePort, Ingress-Controller und Load Balancer mehrere Methoden, um Anfragen von außerhalb des Clusters an einen Pod durchzuleiten.

NodePort öffnet einen bestimmten Port auf allen Nodes und alle Anfragen an diesen Port werden zu einem der passenden Pods durchgereicht. Das funktioniert allerdings nur mit einem Service pro Port. Ein Service ist ein Objekt in Kubernetes, das einer Anwendung im Pod eine Netzwerkschnittstelle zur Verfügung stellt. Ein Ingress-Controller arbeitet als eine Art intelligenter Router, der vor den eigentlichen Services im Backend sitzt und eingehende Anfragen anhand konfigurierter Regeln weiterleitet. Ein beliebter Ingress-Controller für Kubernetes ist beispielsweise Traefik, dem wir bereits einen eigenen Artikel gewidmet haben [1]. Ingress-Controller sind sehr flexibel, erfordern aber

## c't kompakt

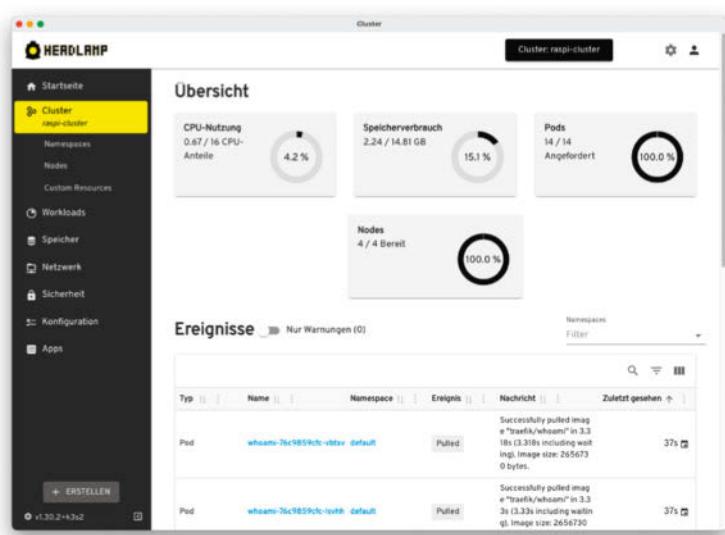
- Ein Raspi-Kubernetes-Cluster im Homelab ist eine gute Spielwiese, um sich mit dem mächtigen Container-Orchestrator vertraut zu machen.
- Der ausfallsichere Bare-Metal-Cluster kommt mit zwei Load Balancern, die seine Ressourcen außerhalb des Clusters verfügbar machen.
- Sind alle Raspis verkabelt, ist der Cluster dank Ansible in wenigen Minuten einsatzbereit und anschließend wieder abgeräumt.

weitere Konfiguration, was den Rahmen dieses Artikels sprengen würde.

Bei unserem Bare-Metal-Cluster kommt stattdessen ein Load Balancer zum Einsatz, der ähnlich wie die Load Balancer der Cloudprovider funktioniert und einige Ingress-Aufgaben mit abdeckt. Filter und Routen wie bei einem ausgewachsenen Ingress-Controller gibt es jedoch nicht. Das hat aber auch den Vorteil, dass sämtlicher Traffic bei den Pods ankommt.

Vorhang auf für MetalLB, einen Load Balancer speziell für Bare-Metal-Cluster: In seiner Konfiguration legen Sie später einen IP-Adressbereich fest, über den MetalLB verfügen darf, beispielsweise 192.168.3.20 bis 192.168.3.100. Er kümmert sich dann darum, neuen Services IP-Adressen zuzuordnen und sie wieder abzuräumen, wenn sie nicht mehr benötigt werden.

Im Homelab verfügen die wenigsten über Profi-Router, die BGP (Border Gate-



The screenshot shows the Headlamp web interface for managing a Kubernetes cluster. On the left is a sidebar with navigation links: Startseite, Cluster (highlighted), Namespaces, Nodes, Custom Resources, Workloads, Speicher, Netzwerk, Sicherheit, Konfiguration, and Apps. Below these are buttons for '+ ERSTELLEN' and a search bar with placeholder text 'Name / Namespace / Ereignis / Nachricht / Zuletzt gesehen'. The main area has three sections: 'Übersicht' (Overview) showing CPU-Nutzung (0.67 / 16 CPU-Anteile, 4.2%), Speicherverbrauch (2.24 / 14.81 GB, 15.1%), and Nodes (4 / 4 Bereit, 100.0%); 'Ereignisse' (Events) with a table showing two entries for 'Pod' objects named 'whoami-76c9859c7c-vbtx' and 'whoami-76c9859c7c-vbtx' in the 'default' namespace, both with status 'Pulled' and timestamp '37s ago'; and 'Namespaces' with a table showing two entries for 'whoami' namespaces, both with status 'SuccessFully pulled image "traefik/whoami" in 3.318s (3.308s including waitin...)' and timestamp '37s ago'.

way Protocol) sprechen, deswegen kommt MetalLB im Layer-2-Modus zum Einsatz. Die IPv4-Adressen, die MetalLB mit Services verdrahtet hat, teilt es dem lokalen Netzwerk über ARP (Address Resolution Protocol) mit. Grundsätzlich beherrscht es aber auch NDP (Neighbor Discovery Protocol) für IPv6-Adressen. MetalLB ist aber nur die erste von zwei Säulen eines resilienten Bare-Metal-Clusters, denn die IP-Adressen für die Services bleiben auch bestehen, wenn ein Raspi ausfällt.

## Noch ein Load Balancer?

Die Control-Plane-Komponenten wie den kube-apiserver und die verteilte Key-Value-Datenbank etcd halten wir auf drei Raspis vor, die als Server-Nodes fungieren. Die ungerade Anzahl ist eine Anforderung von etcd, damit es ein Quorum bilden kann. In einem Cluster mit  $n$  Mitgliedern braucht es  $(n/2)+1$  Mitglieder, die gemeinsam per Mehrheitswahl Entscheidungen treffen. Wie ein Entscheidungsprozess in einem Kubernetes-Cluster im Detail abläuft, lesen Sie detailliert in [2]. Die drei Server-Nodes bilden gemeinsam das sogenannte Control Plane.

Die Server-Nodes sind so konfiguriert, dass sie auch Workloads abseits des Cluster-Tagesgeschäfts ausführen. Dazu kommt ein Agent-Node ohne etcd, das nur Workloads ausführt. Das ist nicht unbedingt für den Produktivbetrieb zu empfehlen, denn eigentlich will man verhindern, dass ein fehlerhafter oder kompromittierter Pod ein Server-Node abräumt; im Homelab ist das aber tolerierbar.

Für das Load Balancing des Kubernetes-API kommt kube-vip zum Einsatz. Es nutzt die gleichen Werkzeuge (ARP, IP-Adresse), um Traffic in das Kubernetes-Kontrollzentrum zu leiten. Es ist die zweite Säule, um den Raspi-Cluster hochverfügbar zu machen.

## Ansible vorbereiten

Unter ct.de/ymyu finden Sie ein GitHub-Repository, das den Bauplan des eben beschriebenen Clusters in Form eines Ansible-Playbooks enthält. Das stammt nicht aus unserer Feder, sondern basiert auf dem Playbook des Entwicklers Timothy Stewart, der es unter der Apache-2-Lizenz zur Verfügung stellt und damit bereits einige tausend Sterne bei GitHub eingesammelt hat. Wir haben es etwas modifiziert, damit es leichter nachvollziehbar wird. Eine Einführung in Ansible lesen Sie in [3].

Klonen Sie zunächst das Repository und wechseln in das Verzeichnis:

```
git clone \
https://github.com \
ndi-ct/k3s-ansible.git
```

Anschließend müssen Sie einige Module herunterladen:

```
ansible-galaxy collection \
install -r \
/collections/requirements.yml
```

Das Playbook von Stewart ist eine ziemliche Allzweckwaffe und kann prinzipiell auch auf x86-Bare-Metal-Hosts oder Linux-Container in Proxmox angewendet werden. Außerdem richtet es optional auch komplexe Kübernetes-CNI (Container Network Interfaces) wie Cilium oder Calico ein. Es enthält deswegen eine Vielzahl von Rollen und Variablen, die meisten davon müssen Sie aber für dieses Beispiel nicht beachten. Ansible erkennt automatisch, dass es k3s auf einem Raspi-Verbund einrichten soll, lädt automatisch die korrekten Binärdateien herunter und spannt Flannel als CNI ein, das zur Serienausstattung von k3s gehört.

Sie müssen lediglich das Inventory, die Textdatei imINI-Format namens hosts.ini und ein paar Gruppenvariablen

in all.yml bearbeiten. Beide Dateien finden Sie im Unterverzeichnis /inventory/raspi-cluster im geklonten Repository. Öffnen Sie die Datei hosts.ini und tragen Sie die statischen IP-Adressen ein, die Sie für die Raspis festgelegt haben. Alle Hosts in der Gruppe master formt Ansible zu Server-Nodes, der Host in der Gruppe node dient als Agent-Node:

```
[master]
192.168.3.101
192.168.3.102
192.168.3.104
```

```
[node]
192.168.3.104
```

```
[k3s_cluster:children]
master
node
```

Bearbeiten Sie danach die Datei all.yml und tragen Sie bei ansible\_user den Benutzernamen ein, den Sie zuvor beim Schreiben des Images mit dem Raspberry Pi Imager hinterlegt haben. Geben Sie bei apiserver\_endpoint eine IP-Adresse ein, die kube-vip mittels ARP als Schnittstelle zum Kübernetes-API im Netz bekannt macht.

k3s-token dient als Authentifizierungs-Token und sichert die Kommunikation zwischen den Server-Nodes. Tragen Sie hier einen zufälligen String ein, den Sie beispielsweise auf der Kommandozeile mit diesem Befehl generieren können:

```
openssl rand -base64 48 | tr -dc \
'a-zA-Z0-9' | head -c 64
```

Zum Schluss braucht es noch einen IP-Addressbereich, den der MetalLB-Load-Balancer verwalten darf. Tragen Sie den bei metal\_lb\_ip\_range ein, beispielsweise 192.168.3.20-192.168.3.100. Damit ist der Raspi-Cluster startklar.

Mit folgendem Befehl ziehen Sie den Cluster hoch:

```
ansible-playbook site.yml \
-i inventory/raspi-cluster/hosts.ini \
--key-file "~/.ssh/cttest"
```

Ersetzen Sie dabei den Pfad hinter --key-file durch den Pfad zum SSH-Schlüssel, mit dem Sie auf die Raspis zugreifen können. Die vollständige Installation inklusive Reboot aller Raspis hat in unserem Versuch etwa fünf Minuten gedauert. Die Dauer hängt auch von der Geschwindigkeit des



**Die whoami-Anwendung zeigt bei wiederholten Aufrufen unterschiedliche Hostnamen an. Dann ist jeweils ein anderer Pod an die Reihe gekommen.**

Internetanschlusses ab, denn k3s lädt eine Reihe von Container-Images herunter. Besonders praktisch: Als letzte Amtshandlung fischt Ansible die kubeconfig-Datei von einem Server-Node und kopiert sie in das aktuelle Verzeichnis. Wenn Sie die Datei öffnen, finden Sie in der Zeile `server` die IP-Adresse, die kube-vip als Load Balancer für das Kubernetes-API angelegt hat.

Üblicherweise liegt die kubeconfig-Datei, der Schlüssel zum Cluster, als config im Verzeichnis `~/.kube`. Sie können die Datei umbenennen und verschieben, um mit `kubectl` auf dem ganzen System damit zu arbeiten oder Sie geben mit dem Parameter `--kubeconfig` stets den Pfad zur Datei an.

Mit folgendem Befehl (im Ansible-Verzeichnis) prüfen Sie, ob Sie Zugriff auf das Kubernetes-API Ihres Raspi-Clusters haben:

```
kubectl --kubeconfig kubeconfig get nodes
```

Als Ausgabe sollte `kubectl` die Server- und Agent-Nodes auflisten.

## It's alive!

Jetzt ist es an der Zeit, dem Cluster mit einer Beispielanwendung einen Workload zuzuweisen. Als Anschauungsmaterial dient der Container `traefik/whoami`, ein simpler Webserver, der Informationen über den Host und den Client bereitstellt. Beantragen Sie den Pod in vierfacher Ausführung (`replicas: 4`). Wichtig für MetalLB ist das Label `app: whoami`, das anschließend in der Beschreibung des Service referenziert wird.

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: whoami
spec:
  selector:
    matchLabels:
      app: whoami
  replicas: 4
  template:
    metadata:
      labels:
        app: whoami
    spec:
      containers:
        - name: whoami
          image: traefik/whoami
          ports:
            - containerPort: 80
```

**Der Load Balancer MetalLB reicht Anfragen an die IP-Adresse eines der Pods durch.**

```
~/Downloads/k3s-ansible-main * default (0.143s)
kubectl --kubeconfig kubeconfig describe service whoami
Name: whoami
Namespace: default
Labels: <none>
Annotations: metallb.universe.tf/ip-allocated-from-pool: first-pool
Selector:
Type: LoadBalancer
IP Family Policy: PreferDualStack
IP Families:
IPs: 10.43.141.139, 10.43.141.139
LoadBalancer Ingress: 192.168.3.150
Port: <unset> 80/TCP
TargetPort: 80
NodePort: <unset> 32266/TCP
Endpoints: 10.42.1.2:80,10.42.2.2:80,10.42.3.2:80
Session Affinity: None
External Traffic Policy: Cluster
Events: <none>
```

Befördern Sie das Deployment mit folgendem Befehl in den Cluster:

```
kubectl --kubeconfig kubeconfig \
apply -f example/whoami.yml
```

Noch wäre die Anwendung ohne Portweiterleitung oder NodePort nicht erreichbar. Das Problem löst MetalLB, indem es dem Deployment eine IP-Adresse zuweist. Der `selector`-Abschnitt gibt an, dass der Service nach Pods sucht, die das Label `app: whoami` tragen und den Netzwerkverkehr an sie durchreicht.

```
---
apiVersion: v1
kind: Service
metadata:
  name: whoami-lb
spec:
  ipFamilyPolicy: PreferDualStack
  selector:
    app: whoami
  ports:
    - port: 80
      targetPort: 80
    type: LoadBalancer
```

Erstellen Sie jetzt den Service:

```
kubectl --kubeconfig kubeconfig \
apply -f example/whoami-lb.yml
```

Um herauszufinden, welche IP-Adresse MetalLB `whoami` außerhalb des Clusters zugewiesen hat, lassen Sie sich Details zum Service anzeigen:

```
kubectl --kubeconfig kubeconfig \
describe service whoami
```

In der Zeile `LoadBalancer Ingress` steht die IP-Adresse, die Sie jetzt beispielsweise im

Browser aufrufen können. Die `whoami`-Anwendung zeigt bei jedem Aufruf den Hostnamen und die IP-Adresse innerhalb des Clusters an. Diese Werte ändern sich beim wiederholten Aufrufen der Seite, weil der Load Balancer Ihre Anfrage an einen anderen Pod durchreicht. Mit der IP-Adresse von MetalLB haben Sie eine praktische Schnittstelle, die Sie nutzen können, um Ressourcen auch außerhalb Ihres Netzwerks verfügbar zu machen, beispielsweise mittels VPN oder durch einen Tunnel [4].

Der Cluster verkraftet den Ausfall einzelner Nodes und gibt die Pods dann woanders in Auftrag, das funktioniert bislang aber nur mit Pods, die nicht auf persistenten Speicher angewiesen sind. Wenn Ihnen das nicht ausreicht, sollten Sie sich den redundanten Container-Speicher Longhorn ansehen [5], der prima zu dem bisherigen k3s-Setup passt. Falls Sie genug vom Raspi-Cluster haben oder von vorn anfangen möchten, versetzt das Playbook `reset.yml` alle Nodes wieder in den Ausgangszustand.

(ndi@ct.de) ct

## Literatur

- [1] Jan Mahn, Containerkompetenzoffensive, Auf dem Lernpfad zum Kubernetes-Kenner, Teil 1 bis 5, c't 22/2022, S. 164
- [2] Jan Mahn, Gemeinsame Wahrheit, Wie verteilte Systeme dank Raft-Algorithmus zusammenarbeiten, c't 22/2021, S. 146
- [3] Pina Merkert, Telerec't, Server im Rechenzentrum oder daheim mit Ansible automatisieren, c't 1/2024, S. 150
- [4] Niklas Dierking, Bye bye, localhost, Tunnelbau mit Cloudanbietern, c't 20/2022, S. 150
- [5] Jan Mahn, Speicherherde, Redundanter Container-Speicher mit Longhorn, c't 7/2023, S. 164

**GitHub-Repository zum Projekt:** [ct.de/ymyu](https://ct.de/ymyu)



# Himbeeriger IP-Ausguck

## Eigenbau: Outdoor-PoE-Kamera mit Raspberry Pi

**Wetterfeste Netzwerkkameras gibts zuhauf. Doch Open Source sind die wenigsten davon.**

**Mit einem Raspberry Pi, einer Kamera sowie überschaubarem Bastel- und Konfigurationsaufwand kann man sich aber eine quelloffene Alternative mit guter Videoqualität zusammenbauen. Wir zeigen, wie's geht.**

Von Andrijan Möcker

Wer Haus oder Hof mit IP-Kameras im Blick behalten möchte und Wert auf quelloffene Software legt, hat nicht sonderlich viel Auswahl. Tatsächlich quelloffene, betriebsfertige IP-Kameras fanden wir bei Recherchen für diesen Artikel nicht. Lediglich die alternative Firmware OpenIPC existiert, die aber nur einige Kameramodelle und SoCs unterstützt.

Scheuen Sie sich nicht vor etwas Bastlei, kommen Sie ganz leicht mit einem Raspberry Pi und einer passenden Kamera zu einer Überwachungskamera mit quelloffener Software. Per USB verbindet man die heute nicht mehr; man bedient sich der Kameras für das Camera Serial

Interface (CSI) des Raspberry Pi. Die Raspberry-Pi-Foundation bietet derzeit sieben Kameras für unterschiedliche Anwendungsfälle; weitere kompatible Kameras gibt es von anderen Herstellern.

Wir haben aus der großen Kameraauswahl und den unterschiedlichen Möglichkeiten, die Konstruktion wittertauglich zu machen, einen Bauvorschlag zusammengestellt, der sich auch für Basteleinsteiger eignet und wenig Werkzeug erfordert. Er sitzt in einem Außengehäuse und wird per Power-over-Ethernet versorgt, sodass die Spannungsversorgung keinen Elektriker erfordert. Auf Nachtbetrieb ist das Set indes nicht optimiert; entsprechende Hardware verlinken wir aber.

Im Verlauf des Artikels zeigen wir, wie Sie Ihre Raspberry-Pi-Kamera zusammenbauen und konfigurieren. Wichtig vorab: Gerade für Raspi-Einsteiger, die sonst nur App-gesteuerte IP-Kameras gewohnt sind, kann dieses Projekt schnell anspruchsvoll werden. Viele Konfigurationsschritte müssen auf der Kommandozeile erledigt werden und ein Netzwerktechnikgrundwissen hilft dabei enorm. Außerdem sollten Sie Schulenglisch mitbringen und sich nicht vor englischen Dokumentationen scheuen, denn dieser Artikel deckt nur die Grundkonfiguration der Software ab, nicht ihren gesamten Funktionsumfang.

## Einkauf

Wir haben uns für diesen Artikel für die „High Quality Camera“ der Raspberry-Pi-Foundation entschieden. Die 12-Megapixel-Kamera liefert nicht nur Full-HD-Videoauflösung mit bis zu 30 Bildern pro Sekunde, sie hat auch wahlweise einen M12- oder CS-Objektivanschluss. Dafür bekommt man in den Shops Objektive mit allerhand Brennweiten und Blenden. Gegenüber den Kameras mit Festbrennweiten hat das den Vorteil, dass man beim Installieren nicht so sehr darauf achten muss, dass der Ausschnitt stimmt. Derzeit kostet die Kamera ohne Objektiv rund 60 Euro.

Als Objektivwählten wir ein 8-50-Millimeter-Zoomobjektiv, das wir bei Berrybase für 37 Euro erstanden. Außerdem kauften wir bei AliExpress ein Zoomobjektiv mit 2,8 bis 12 Millimeter Brennweite.

Das Herzstück der ganzen Angelegenheit ist ebenso flexibel: Theoretisch funktioniert die HQ-Camera bereits mit einem Raspberry Pi 2. Allerdings ist für unseren Bauvorschlag mindestens ein 3B+ nötig. Ausdrücklich 3B+, nicht 3B, denn der 3B+ hat nicht nur mehr Rechenleistung als ein Pi 2, sondern auch die Pins für Power-over-Ethernet, kurz PoE.

Die IEEE-Spezifikation 802.3af/at/bt erlaubt Spannungsversorgung von Geräten über die Netzwerkleitung und ist heute bei IP-Kameras, WLAN-Access-Points oder IP-Telefonen weit verbreitet; ein lokales Netzteil entfällt dadurch. Pis sind aber nicht von Haus aus PoE-fähig, sie benötigen einen passenden Hat (so heißen Erweiterungsplatinen für den Rasp), der die bis zu 56 Volt aus der Netzwerkleitung für den Minicomputer verdaulich macht; aktuelle Modelle haben dafür vier Pins in direkter Nachbarschaft zur Netzwerk-

## ct kompakt

- Mithilfe der Tools MediaMTX und MotionEye ist der Bau einer Raspi-IP-Kamera leicht.
- Verwendet man ein fertiges Außengehäuse, hält sich der Bastelaufwand in Grenzen.
- Kameras und Objektive für unterschiedliche Anwendungsfälle findet man allerhand.

buchse. Hats für den Pi 3B+ sind auch mit dem Pi 4 kompatibel. Der Pi 5 ist hingegen anders bestückt und benötigt angepasste PoE-Hats.

Wir setzen in diesem Projekt den Raspberry Pi 4 mit 2 GByte RAM und dem PoE-Hat (E) von Waveshare ein (12,5 Watt Ausgangsleistung). Das genügt reichlich für einfache Bewegungserkennung, Videoaufzeichnung und Benachrichtigungsversand.

Wer gleich richtig einsteigen und rechenleistungsintensivere Anwendungen wie solche mit KI-Unterstützung laufen lassen möchte, sollte zu einem Raspberry Pi 5 greifen. Wir gehen darauf in diesem Artikel zwar nicht ein, Sie können die Schritte aber genauso befolgen und sich später in KI-Anwendungen einführen.

Waveshare bietet für den Pi 5 den PoE Hat (F) mit bis zu 22 Watt Ausgangs-

leistung. Das genügt auch, um das Raspberry Pi AI Kit zu versorgen, das per PCIe angeschlossen wird. Doch Obacht: Sitzt der PoE-Hat auf dem Pi, genügt das PCIe-Flachbandkabel (3 cm) des AI-Kits nicht mehr und muss durch ein sechs Zentimeter langes Exemplar ersetzt werden.

Zusätzlich zum Raspberry Pi benötigen Sie eine MicroSD-Karte, die mindestens 16 GByte, besser 32 GByte, Speicherkapazität hat und die Geschwindigkeitsklasse A1 erfüllt. Um die Karte am PC zu beschreiben, benötigen Sie zudem einen Micro-SD-Leser; ob integriert oder per USB angeschlossen, spielt keine Rolle. Ein Raspberry-Pi-Gehäuse ist nicht nötig.

Damit der Pi auch Energie bekommt, braucht eine Power-over-Ethernet-Spannungsquelle. Für unser Vorhaben genügen 15 Watt (802.3af); für einen Pi 5 plus KI-Modul sollten es 30 Watt sein (802.3at). Besitzen Sie keinen PoE-Switch, können Sie einen Injektor benutzen. Dabei lohnt es sich, gleich einen mit Luft nach oben zu nehmen – etwa den TL-POE160S (30 Watt) von TP-Link für rund 20 Euro. Er wird zwischen die Netzwerkverbindung (Router, Switch) und die Leitung zum Pi gesteckt.

Links zum Kauf aller bereits erwähnten und noch folgenden Komponenten finden Sie unter [ct.de/yp3g](http://ct.de/yp3g).

## Outdoor-Gehäuse

Ein wasserfestes Gehäuse zu entwerfen und beispielsweise mit dem 3D-Drucker oder einem Lasercutter selbst herzustellen



**Die „Raspberry Pi High Quality Camera“ gibt es wahlweise mit M12- oder CCS-Objektivanschluss. Für beide Anschlüsse gibt es allerhand Objektive und Objektivadapter. Praktisch ist außerdem, dass beide Varianten ein Stativgewinde haben, was bei unserem Projekt extrem einfache Montage erlaubt.**

## Projektinfo

- Raspberry Pi als PoE-gespeiste IP-Kamera im Außengehäuse**
- Netzwerk- und Linuxgrundkenntnis**
- 2-4 Stunden**
- ab 250 Euro**
- Raspberry Pi 3B+, 4 oder 5 microSD-Karte (ab 16 GByte)**  
Raspberry Pi High Quality Camera  
Raspi-Kühler  
Objektiv  
Außengehäuse mit Halterung  
PoE-Hat  
PoE-Injektor  
Netzwerkkabel  
RJ45-Stecker  
Abstandshalter  
Mast-Montageplatte
- Schraubendreherset**  
Bohrerset  
Akkuschrauber  
Bohrmaschine (Wandmontage)  
Seitenschneider  
(micro)SD-Leser

Gehäuse platzieren kann und auf dem genug Platz für Raspberry Pi und Kamera ist. Doch nicht nur das: Im Lieferumfang sind auch drei ¼-Zoll-Schrauben enthalten, von denen eine als Reserve gedacht ist und somit die Kamera auf dem Schlitten halten kann, der dafür sogar eine passende Nut besitzt. Außerdem hat das Gehäuse zwei Öffnungen auf der Rückseite, die der Hersteller bereits mit einer PG11- und einer PG9-Kabelverschraubung bestückt.

Da durch keine der beiden Öffnungen noch ein RJ45-Stecker passt, muss man das Netzwerkkabel im Gehäuse selbst mit einem Stecker verbinden. Ein Crimpset ist dafür nicht unbedingt notwendig, denn im Netz gibt es günstig sogenannte feldkonfektionierbare Stecker, die nahezu werkzeugfreies Auflegen erlauben. Meist reichen ein scharfes Messer zum Entmanteln des Kabels und ein Seitenschneider (notfalls auch eine Schere). Weitere Informationen zu Kabeltypen zum Auflegen von Netzwerkkabeln finden Sie in c't 26/2020 und c't 14/2023 [1, 2].

Um den Raspberry Pi zu montieren, haben wir noch ein Set mit M2.5-Abstandsbolzen sowie passenden Schrauben und Muttern bestellt ([ct.de/yp3g](http://ct.de/yp3g)). Ohne geht es nicht, denn die Schrauben des Schlittens sind zum Teil knapp unter dem Raspi und wir nutzen den Platz darunter zum Umbiegen des Flachbandkabels der Kamera; hätte der Anschluss direkt den Raspi im Rücken, würde das Kabel aufgrund der Enge wahrscheinlich brechen.

Außerdem haben wir Mast-Montageplatten für den mitgelieferten Arm der Kamera besorgt. Andernfalls taugt der Lieferumfang der Kamera nur zur Wandmontage.

## Raspi OS vorbereiten

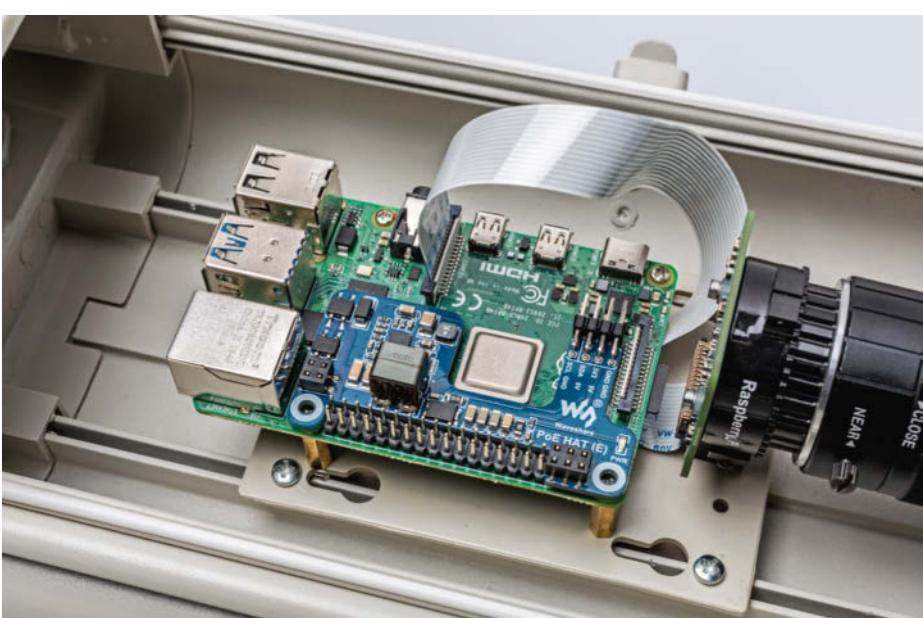
Dank des Raspberry Pi Imagers, einem Tool der Raspberry-Pi-Foundation für alle gängigen Betriebssysteme, ist das Vorbereiten der MicroSD-Karte ein Kinderspiel. Einen Bildschirm am Raspberry Pi anzuschließen ist nicht notwendig, weil der Rest per SSH-Konsolenzugriff erledigt wird.

Laden Sie zunächst den Raspberry Pi Imager für Ihr Betriebssystem herunter; Sie finden ihn über [raspberrypi.com/software/](https://raspberrypi.com/software/). Nach der Installation setzen Sie Ihre SD-Karte in den Leser ein, öffnen das Tool und wählen zuerst Ihr Raspberry-Pi-Modell aus. Im Menü „Betriebssystem (OS)“ klicken Sie auf die Option „Raspberry Pi OS (other)“ und dort auf „Raspberry Pi OS Lite (64-bit)“ (32 Bit beim Pi 3). Diese Version hat keine Desktopumgebung an Bord, was Speicherplatz spart.

Wählen Sie anschließend Ihre SD-Karte aus, klicken Sie dann auf „Weiter“ und im folgenden Pop-up auf „Einstellungen bearbeiten“. Das sich öffnende Menü erlaubt, das Raspberry Pi OS noch vor dem ersten Start mit Einstellungen zu versehen: Im Reiter „Allgemein“ setzen Sie zunächst Hostname, Benutzername und

erfordert einiges an Werkzeug und zeitlichen Aufwand. Wir haben uns deshalb für ein leeres Außen-Kameragehäuse entschieden, in das man beliebige Kameras einsetzen kann. Solche Gehäuse werden normalerweise von Herstellern professioneller Überwachungstechnik eingesetzt und sind teuer, doch auf der Fernost-Händelsplattform AliExpress kommt man günstiger zum Ziel. Da die Europäische Union außereuropäische Onlinehändler wie AliExpress dazu verdonnert hat, die Mehrwertsteuer selbst abzuführen und bis 150 Euro kein Zoll anfällt, wird für so einen Kauf auch kein Besuch beim Zollamt nötig.

Wir zahlten 38 Euro für das Gehäuse mit dem eingängigen Namen „10 Inch External Street Waterproof Aluminum CCTV Surveillance Camera Housing Case Shield with Clear Windshield“ vom „OwlCat Official Store“. Das Gehäuse hat eine leicht zu öffnende Haube und bietet rund 20 cm × 10 cm × 7 cm Platz im Innenraum. Mitgeliefert wird ein Schlitten (siehe Foto rechts), den man flexibel im



**Im Gehäuse ist genug Luft für den Raspberry Pi, die Kamera und mehr. Der mitgelieferte Schlitten eignet sich gut, um die beiden Komponenten zu montieren. Gummidichtungen stellen sicher, dass keine Feuchtigkeit eintritt.**

Kennwort sowie die Zeitzone und das Tastaturlayout; WLAN-Einstellungen sind nicht notwendig, Sie verbinden den Pi schließlich per Kabel.

Im Reiter Dienste aktivieren Sie SSH, damit später die Konsolenverbindung klappt. Haben Sie bereits Erfahrungen mit SSH gemacht und ein Schlüsselpaar zur Authentifizierung, können Sie dieses dort eintragen. Andernfalls wählen Sie die Option „Passwort zur Authentifizierung verwenden“. Damit sind dann die Zugangsdaten gültig, die Sie zuvor im Reiter „Allgemein“ gesetzt haben.

Bestätigen Sie anschließend die Einstellungen und starten Sie den Schreibvorgang. Sobald der Imager Erfolg meldet, können Sie die Karte herausziehen und in den Raspberry Pi stecken.

## Zusammenbau

Für den Zusammenbau legen Sie sich das im Projektkasten aufgeführte Werkzeug bereit. Entfernen Sie zunächst mit einem Kreuzschraubendreher den Schlitten aus dem Gehäuse. Dann nehmen Sie eine der  $\frac{1}{4}$ -Zoll-Schrauben und montieren die Kamera inklusive Schutzabdeckungen am Ende der Nut auf dem Schlitten, um ein Gefühl für deren Platzbedarf zu haben und gleich die Bohrlöcher für den Raspberry Pi anzulegen.

Danach legen Sie den Schlitten wieder ins Gehäuse, schrauben ihn mit zwei Schrauben fest und nehmen sich den Raspberry Pi zur Hand, um die Bohrlöcher anzulegen. Im Bild auf Seite 156 ist gut zu sehen, wie man ihn nicht montieren sollte: Wir haben den Pi etwas zu nah an der einen Nut positioniert, sodass die Muttern gleichzeitig als Abstandshalter fungiert. In Konsequenz ist das Kamerabild leicht schief. Wir haben das Problem umschifft, indem wir die Muttern auf der betroffenen Seite von den Abstandsbolzen entfernt haben, sodass diese nur noch als Stütze und nicht mehr als Halter fungieren. Der Pi sitzt damit noch immer mehr als ausreichend fest.

Allerdings ist es in dieser Position auch auf der anderen Seite relativ knapp, sodass man mit einem groben Seitenschneider ein Teil aus der Plastikführung entfernen muss, um die Muttern der Abstandsbolzen festziehen zu können. Wollen Sie das vermeiden, positionieren Sie die Löcher des Raspberry Pis auf der einen Seite oberhalb der Gehäusenut; dort können die Abstandsbolzen zwar auch nur als Stützen agieren, dafür ist aber auf der an-



**Unser Bauvorschlag sitzt mittlerweile auf einem Dach und hat dort schon einige Regenschauer überstanden.**

deren Seite mehr als genug Luft für die Muttern und der Schlitten sitzt waagerecht im Gehäuse. Zeichnen Sie die Löcher entsprechend an, entfernen Sie den Schlitten aus dem Gehäuse und die Kamera vom Schlitten. Wir haben die Löcher mit einem 3-Millimeter-Holzbohrer gebohrt.

Montieren Sie die Abstandsbolzen auf der Seite, wo diese Muttern bekommen, bereits auf dem Schlitten. Die für die andere Seite, die nur Stützen sind, fixieren Sie mit Schrauben am Raspberry Pi. Danach schrauben Sie auch die Kamera wieder am Schlitten fest und führen das Flachbandkabel nach hinten über den Schlitten. Entfernen Sie die Schutzkappen und drehen Sie das Objektiv auf die Kamera; warten Sie nicht zu lange nach dem Abziehen der Schutzkappen, da sich sonst Staub in Kamera und Objektiv sammelt.

Wollen Sie die Konstruktion für die Einrichtung zunächst ohne Gehäuse in Betrieb nehmen, geht das auch, andernfalls setzen sie den Schlitten ins Gehäuse, bewegen ihn nach vorne, bis das Objektiv an der Scheibe sitzt, und schrauben ihn fest.

Ob draußen oder eingebaut, jetzt müssen die Kamera angeschlossen und der Pi festgeschraubt werden: Greifen Sie sich den Raspberry Pi und suchen Sie den Flachbandkabelanschluss für die Kamera auf der Platine; er liegt zwischen dem Micro-HDMI-Anschluss und der Klinken-

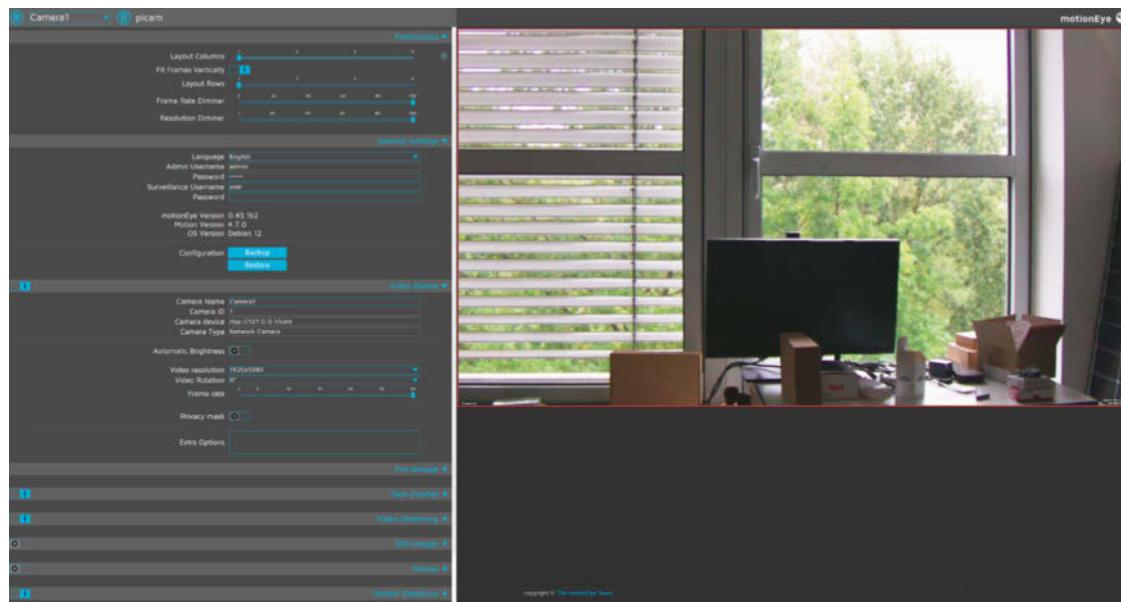
buchse. Ziehen Sie behutsam die schwarze Klemme hoch und führen Sie das Flachbandkabel der Kamera mit der blauen Seite zur Klemme in den Anschluss ein. Sobald Sie Widerstand spüren, drücken Sie die schwarze Klemme herunter, um das Kabel zu fixieren.

Damit das Flachbandkabel beim Einbau nicht abgeknickt wird, ziehen Sie es beim Aufsetzen auf den Schlitten mit einem Schraubendreher unter den Pi. Stecken Sie die „Stützen“ auf der einen Seite im Schlitten und schrauben Sie den Pi auf der anderen Seite an den Haltern fest.

## Energie!

Um den Raspberry Pi per Power-over-Ethernet zum Leben zu erwecken, montieren Sie jetzt den PoE-Hat. Wichtig dabei ist, dass die Bauteile nach oben zeigen müssen; die schwarzen Buchsenleisten werden von unten durchgesteckt, nicht von oben. Agieren Sie dabei vorsichtig und drücken Sie nicht weiter, wenn größerer Widerstand zu spüren ist. Wir haben bei unserem Zusammenbau etwas zu viel Druck ausgeübt, sodass sich die Buchse für die PoE-Pins von der Platine verabschiedet hat und wir sie wieder anlöten durften.

Sitzt der Hat, kleben Sie die Kühlkörper auf die CPU. Dann fädeln Sie das Netzwerkkabel durch eine der Kabelverschraubun-



**Das Webinterface von MotionEye ist schlicht gehalten, erlaubt aber Bewegungserkennung inklusive Fotos und Videos ohne zusätzlichen Netzwerkvideorecorder, direkt auf dem Raspi.**

gen des Gehäuses. Wir haben die dünnere der beiden genommen, weil wir die größere für spätere Projekte frei halten wollen. Anschließend crimpen beziehungsweise legen Sie die RJ45-Stecker auf beiden Seiten des Kabels auf.

Stecken Sie Pi, Zuleitung, PoE-Injektor und Ihre Netzwerkinstallation zusammen, sollte der PoE-Hat anfangen, rot zu leuchten und der Raspi erste Lebenszeichen zeigen. Kurze Zeit später leuchtet auch die Netzwerkbuchse. Wenn nicht, kontrollieren Sie alle Steckverbindungen und das Netzwerkkabel auf korrekte Belegung.

## Einrichtung

Ist Ihr Kamera-Pi online, begeben Sie sich auf das Webinterface Ihres Routers und prüfen, welche IP-Adresse er bekommen hat. In der Regel zeigen die Router in der Liste auch den Hostnamen des Gerätes an und somit auch den, den Sie im Imager für Ihren Pi vergeben haben. Details finden Sie in der Anleitung Ihres Routermodells.

Mit der IP-Adresse im Gepäck öffnen Sie auf Ihrem PC eine Kommandozeile und stellen eine SSH-Verbindung her: `ssh <IhrPiBenutzername>@<IP-Adresse>`. Die meisten Linux-Distributionen und macOS-Versionen bringen einen SSH-Client mit; unter Windows müssen Sie in den Einstellungen unter „System/Optionale Features“ den OpenSSH-Client installieren.

In der Raspberry-Pi-Konsole führen Sie zunächst `sudo apt update && sudo apt upgrade -y` aus, um alle Paketquellen und Pakete zu aktualisieren, dann geht es an die IP-Kamera-Software.

Dafür haben Sie drei Optionen: MediaMTX, MediaMTX mit MotionEye oder nur MotionEye. MediaMTX ist ein „Streaming Router“, kann also Videodaten unterschiedlicher Streams transkodieren, weiterleiten und so weiter. Er unterstützt auch Raspberry-Pi-Kameras und kann deren Videodaten als H.264-enkodierten Stream bereitstellen. MotionEye ist eine IP-Kamerasoftware mit Webinterface, die unter anderem Bewegungen erkennen, aufzeichnen und benachrichtigen kann.

MotionEye kann sowohl Streams anderer Kameras abrufen als auch Raspberry-Pi-Kameras direkt einbinden, hatte mit den neueren Modellen aber zuletzt Kompatibilitätsprobleme. Wir verwenden in diesem Artikel deshalb eine Kombilösung und binden den Stream von MediaMTX in MotionEye ein. Das stellt nicht nur sicher, dass alles funktioniert, sondern sorgt auch dafür, dass Ihr Kamera-Pi Ihnen komprimiertes Videomaterial statt bandbreitenhungriges MotionJPEG liefert. MotionEye kann nämlich nur das.

Besitzen Sie einen eigenen Netzwerkvideorecorder (NVR) oder etwa ein NAS mit NVR-Software, können Sie auf MotionEye verzichten, nur MediaMTX einrichten und die RTSP-URL in Ihrem NVR eintragen.

## MediaMTX

Um mit der MediaMTX-Installation loszulegen, müssen Sie zunächst zwei Abhängigkeiten installieren: `sudo apt install libfreetype6 libcamera0.3`. Danach versuchen Sie sich in der Konsolensitzung mit Root-Rechten, damit Sie nicht für jeden

Befehl `sudo` eingeben müssen: `sudo -i`. Wechseln Sie in das Verzeichnis für optionale Softwareinstallationen `cd /opt/`, erstellen Sie mit `mkdir medimtx` einen Ordner für die Installation und wechseln Sie mit `cd medimtx` hinein.

Die notwendigen Dateien stellt der Entwickler auf seiner GitHub-Seite bereit (Link via ct.de/yp3g). Der Pi 3 benötigt das armv7-Paket, die Pis 4 und 7 arm64v8; kopieren Sie den Link des passenden Pakets per Rechtsklick, wechseln Sie ins Terminal und laden Sie die Datei mit `wget <url>` herunter. Der Befehl `tar xzvf <Dateiname>` entpackt die Dateien. Tipp: Drücken Sie nach `tar xzvf` die Tabulator-Taste, macht Ihnen die Autovervollständigung das Leben einfacher. Anschließend löschen Sie die `.tar.gz` mit `rm <Dateiname>` und wechseln mit `cd medimtx` in das ausgepackte Verzeichnis.

Jetzt gehts an die Konfigurationsdatei: Öffnen Sie sie mit `nano medimtx.yml` und springen Sie mit Strg+W, Strg+V ans Ende der Datei. Entfernen Sie die Kommentare und Beispiele unterhalb von `paths:` und fügen Sie die Konfiguration für die Raspberry-Pi-Kamera hinzu:

```
paths:
  cam:
    source: rpiCamera
```

Den Pfad `cam` können Sie beliebig umbenennen, solange Sie keine Leerzeichen einsetzen. Speichern Sie die Konfiguration mit Strg+O, dann suchen Sie mit Strg+W nach `rpicamera`, bis Sie beim Parameter `rpiCameraCamID` angelangt sind. Tipp: Sie

können nach dem ersten Suchen Strg+W erneut drücken und dann mit Enter das zuvor gesuchte Wort erneut suchen, ohne es wieder eingeben zu müssen.

Ab rpiCameraCamID kommen alle Parameter, die für die Raspberry-Pi-Kamera relevant sind. Die voreingestellten Einstellungen taugen nahezu alle, wir haben jedoch den rpiCameraCodec auf hardwareH264 gestellt, das rpiCameraProfile auf High und die rpiCameraBitrate von 1 Mbit/s auf 15 Mbit/s (15000000 Bit/s).

Speichern und schließen Sie die Datei mit Strg+O, Strg+X, dann ist MediaMTX für einen ersten Test bereit: ./mediamtx. Sobald der Dienst läuft, öffnen Sie einen beliebigen RTSP-fähigen Videoplayer – zum Beispiel VLC – auf Ihrem Rechner und öffnen die RTSP-URL: rtsp://<IP-Adresse>:8554/cam (bzw. Ihr selbst vergebener Pfadname).

Klappt etwas nicht, werfen Sie einen kritischen Blick in die Log-Ausgaben und kontrollieren die Konfigurationsdatei auf Tippfehler.

Läuft MediaMTX, stoppen Sie ihn mit Strg+C und fügen den Server als Dienst hinzu. Systemd kümmert sich dann darum, dass MediaMTX beim Hochfahren startet, und hilft ihm wieder auf die Beine, sollte er mal abstürzen. Dazu erstellen Sie mit nano /etc/systemd/system/mediamtx.service eine neue Service Unit und füllen sie mit folgendem Inhalt:

```
[Unit]
Wants=network.target
[Service]
ExecStart=/opt/mediamtx/mediamtx /opt/
mediamtx/mediamtx.yml
[Install]
WantedBy=multi-user.target
```

Danach speichern und schließen Sie sie (Strg+O, Strg+X).

Mit systemctl daemon-reload fängt systemd unter anderem an, nach neuen Service-Dateien zu suchen. Ist der Befehl durchgelaufen, können Sie mit systemctl enable --now mediamtx den Dienst aktivieren und starten. Um das zu kontrollieren, tippen Sie service mediamtx status. Läuft alles, können Sie auch den Videostream wieder aufrufen.

## MotionEye

Wollen Sie jetzt noch MotionEye installieren, geht das in ebenso wenigen Schritten. Sie benötigen zunächst den Python-Paketmanager pip, den Sie wie folgt installieren:

```
curl -sSfO >
https://bootstrap.pypa.io/get-pip.py'
python3 get-pip.py
rm get-pip.py
```

Anschließend installieren Sie mit pip MotionEye auf dem Raspi: pip install --pre motioneye --break. Der Befehl motioneye\_init sorgt dafür, dass MotionEye als Service angelegt und gestartet wird. Verlassen Sie danach den Superuser-Modus wieder mit exit.

Unter http://<IP-Adresse>:8765 erreichen Sie das MotionEye-Webinterface. Klappt das nicht, schauen Sie mit service motioneye status nach dem Zustand und eventuellen Fehlermeldungen; sudo service motioneye restart löst einen Neustart aus.

Im Webinterface loggen Sie sich mit dem User admin ohne Passwort ein und nehmen das blau geschriebene Angebot an, eine neue Kamera hinzuzufügen. Als „Camera Type“ wählen Sie „Network Camera“, tragen Ihre zuvor getestete RTSP-URL ein und bestätigen mit „OK“.

In der Leiste links sollten Sie nun unter „Allgemeine Einstellungen“ ein Administrator- und ein User-Passwort setzen. Unter „Videogerät“ vergeben Sie optional einen anderen Namen für die Kamera, stellen die Videoauflösung auf 1920 × 1080 Pixel und die Bildrate auf 30. Mit „Apply“ speichern Sie die Einstellungen.

Weitere Einstellungen von MotionEye dürfen Sie selbst erkunden; vieles ist für den erfahrenen (IP-Kamera-)Administrator auch selbsterklärend. Zusätzlich helfen die verlinkten Dokuseiten, die Sie über ct.de/yp3g finden.

Haben Sie das Wichtigste konfiguriert, ist Ihre Kamera bereit für die Montage. Fahren Sie den Raspberry Pi mit dem Kommando sudo poweroff herunter; sobald die Netzwerkbuchse und die grüne LED dunkel sind, können Sie ihn stromlos machen und zum Installieren an seinen Bestimmungsort bringen.

(amo@ct.de) 

## Literatur

- [1] Andrijan Möcker, Alles Kabel!, Grundwissen Heimnetzverkabelung: Von Notlösung bis professionell, c't 26/2020, S. 132
- [2] Andrijan Möcker, Sparverdrahtung, Günstige, schnelle Netzwerkverkabelung für Haus und Wohnung, c't 14/2023, S. 136

**Hardware & Software:** [ct.de/yp3g](http://ct.de/yp3g)

Für echte  
Maker und  
**Make: -Fans!**



**NEU  
im heise  
shop**

**Jetzt bestellen**

 [shop.heise.de/2025](http://shop.heise.de/2025)

**Mit Humor  
durchs neue  
Jahr!**



**Auch als Bundle  
mit Buch**

Generell portofreie Lieferung für Heise Medien- oder Maker Media Zeitschriften-Abonnenten oder ab einem Einkaufswert von 20 € (innerhalb Deutschlands). Nur solange der Vorrat reicht. Preisänderungen vorbehalten.





# Leichtmatrose

## Raspi als Boots-Navigationshilfe

**Ein Raspi sorgt beim Bootstörn für mehr Sicherheit: Klein genug für die Hosentasche, nutzt er ein Software Defined Radio, um Positionsdaten von Schiffen zu empfangen und für Navigationsgeräte bereitzustellen. So erkennen Freizeitkapitäne frühzeitig andere Schiffe auf Kollisionskurs.**

Von Falk Nedwal und Mirko Dölle

Auf großen Seen, Flüssen und Kanälen kommen auch Freizeitkapitäne mit ihren Sportbooten und Yachten mit der Berufsschifffahrt und auch großen Binnenschiffen in Berührung – beziehungsweise vermeiden eine solche möglichst. Dabei hilft das automatische Identifikationssystem (AIS), denn darüber können

Sie die Position, Kurs, Geschwindigkeit, Typ und mitunter noch weitere Informationen von Schiffen aus der Umgebung auf den digitalen Navigationskarten Ihres Tablets anzeigen lassen. So erkennen Sie schon früh, ob Sie jemandem in die Quere kommen.

Wer kein eigenes Boot besitzt, muss auf dieses Sicherheits-Feature aber meist verzichten, denn nur wenige Charter-Jachten und Mietboote haben ein Navigationssystem mit AIS-Empfänger oder gar einen AIS-Transponder eingebaut. Dazu gibt es auch keine Verpflichtung. Üblich ist, dass Sie dann Ihr eigenes Tablet mit einer maritimen Navigations-

App wie zum Beispiel Navionics Boating von Garmin verwenden.

Frisch aus  
**c't Nerdistan**

**Groß wie klein**  
Auf vielen Schiffen aus der beruflichen See- und Binnenschifffahrt sind fest installierte AIS-Transponder vorgeschrieben;

sie senden laufend ihre Positions- sowie etliche Schiffsdaten und empfangen die Signale der anderen. Informationsseiten wie MarineTraffic oder Vesseltracker zeigen diese Daten für jedermann zugänglich an, sodass man den Kurs großer Containerschiffe am heimischen PC verfolgen kann. AIS wird ebenfalls genutzt, um etwa Fischerboote zu überwachen und illegale Fischerei aufzudecken. Aber auch kleinere Boote dürfen mit einem AIS-Transponder ausgestattet werden, der allerdings über eine geringere Sendeleistung und damit geringere Reichweite verfügt. Wer in einer Gegend mit (Binnen-)Hafen wie zum Beispiel Hannover wohnt, kann mit einem eigenen AIS-Empfänger die Schiffsbewegungen im Hafen und auf den umliegenden Flüssen und Kanälen verfolgen.

Mit einem Raspberry Pi und einem TV-Empfänger wie einem DVB-T-USB-Stick können Sie die AIS-Daten großer und kleiner Schiffe empfangen und an Ihr Tablet weiterreichen, sodass Sie andere Verkehrsteilnehmer frühzeitig angezeigt bekommen.

AIS zählt zu den aktiven kooperativen Systemen: Es setzt voraus, dass ein Schiff den Transponder betreibt, die korrekten Schiffsinformationen eingestellt sind und auch die Daten des GPS-Empfängers unverfälscht eingespeist werden. Die Datenübertragung erfolgt knapp oberhalb des 2m-UKW-Bands bei rund 160 MHz. Auf großen Schiffen installierte AIS-Transponder der Klasse A senden mit bis zu 12,5 Watt Leistung und übertreffen damit üblicherweise die Reichweite von Radargeräten. Kleine Anlagen der Klasse B für den Fischerei- oder Freizeitbereich senden mit nur 2 Watt.

Der Nachteil von AIS ist, dass man nur Schiffe und Boote mit eingeschaltetem AIS-Transponder im Umkreis einiger Kilometer sehen kann – anders als Radar, das alle größeren Wasserfahrzeuge in unmittelbarer Umgebung erfasst, jedoch nur bis zur nächsten Flussbiegung oder Landzunge reicht. Portable AIS-Transponder und -Empfänger für den Einsatz auf Charter- und Sportbooten kosten gleich mehrere hundert Euro: Für einfache Empfänger zahlt man bereits um 200 Euro, für die günstigsten Transponder muss man bereits mehr als 350 Euro ausgeben.

## Selbstbau für 50 Euro

Selbst gebaut liegt das erforderliche Budget bei gerade einmal 50 Euro: Außer einem Raspberry Pi Zero W oder dessen

Nachfolger Zero 2 W benötigen Sie lediglich einen USB-Stick für terrestrischen Fernsehempfang, die passende Antenne gehört meist schon zum Lieferumfang. Die Stromversorgung an Bord übernimmt entweder eine Powerbank oder ein USB-Ladeadapter für 12 Volt, wie er auch für Autos üblich ist.

Für den Nachbau sollten Sie zum geringfügig teureren Raspi Zero 2 W greifen, der Preisunterschied zum älteren Zero W beträgt kaum zwei Euro. Dafür bietet der Zero 2 mit seiner 64-Bit-Quad-Core-CPU weitaus mehr Rechenleistung als sein älterer Bruder, der nur eine 32-Bit-Single-Core-CPU besitzt. Zwar genügt prinzipiell auch der ältere Zero W, seine Rechenleistung reicht aber nur gerade eben so, um die AIS-Daten zu dekodieren. Es kommt mitunter zu Pufferüberläufen, bei denen Meldungen verloren gehen. Beim Zero 2 W beobachteten wir dieses Problem nicht.

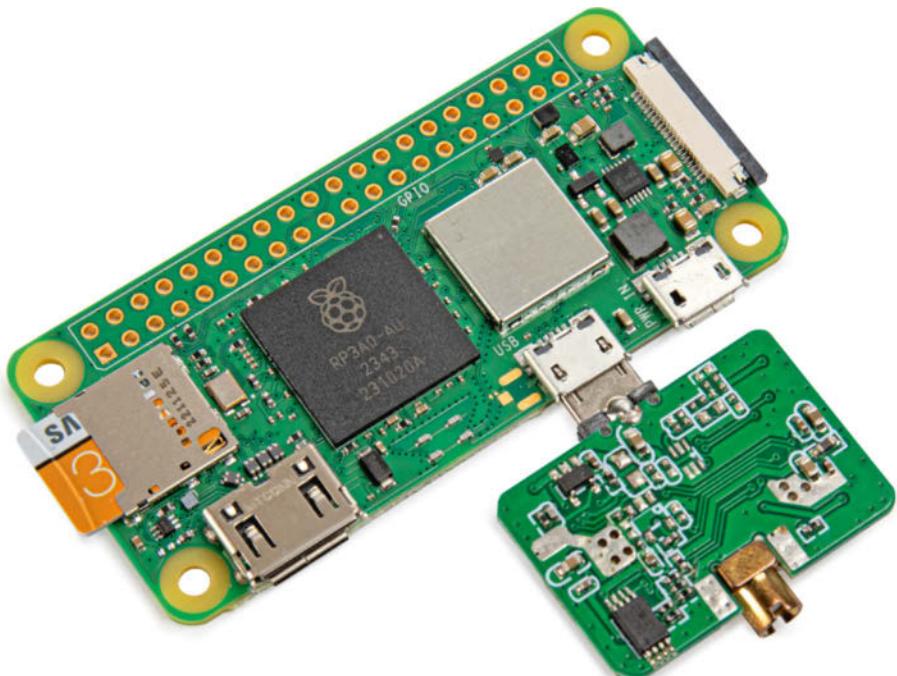
Als Empfänger wird ein TV-Stick mit einem bestimmten Realtek-Chipsatz zweckentfremdet: Einige Open-Source-Entwickler haben herausgefunden, dass sich der in einigen DVB-T-Sticks verbauten Demodulator-Chip RTL2832U direkt ansprechen und darüber der vorgesetzte Empfänger (Tuner) frei programmieren lässt. Mit der richtigen Software wird so aus dem TV-Empfänger ein generischer Breitband-Funkempfänger – ein soge-

## c't kompakt

- Per DVB-T-USB-Stick als Software Defined Radio empfängt und dekodiert der Raspi Zero die Daten der AIS-Transponder großer Schiffe.
- Ein SSH-Zugang via Bluetooth ist praktisch für den Notfall, wenn der Raspi einmal nicht per WLAN erreichbar sein sollte.
- Die integrierte Kartendarstellung zeigt übersichtlich, welche Schiffe auf Seen und Flüssen in der Nähe unterwegs sind.

nanntes Software Defined Radio (SDR). Welche Tuner die Bibliothek `librtlsdr` unterstützt, finden Sie im Wiki des Osmocom-Projekts (Open Source Mobile Communications, siehe [ct.de/ydxs](http://ct.de/ydxs)). Bei Redaktionsschluss waren das die Tuner-Bausteine von Elnecos, Rafael Micro und Fitipower. Sie unterscheiden sich vor allem im unterstützten Frequenzbereich.

Weit verbreitet sind die Tuner Rafael Micro R820T respektive R820T2, die den Frequenzbereich 24 bis 1766 MHz abdecken, sowie Fitipower FC0012 und FC0013 für den Bereich 22 MHz bis 948



**Mit einem TV-Empfänger für Android-Smartphones als Software Defined Radio ist der Selbstbau-AIS-Empfänger besonders kompakt.**

## Projektinfo

-  Raspberry Pi als AIS-Empfänger für Sport- und Charterboote
-  Grundkenntnisse Raspberry Pi OS/Linux
-  2 Stunden
-  ca. 50 Euro
-  Raspberry Pi Zero 2 W  
DVB-T-USB-Empfänger  
Kfz-Ladekabel oder Powerbank
-  Schraubendreher  
Bohrer

MHz und 22 MHz bis 1100 MHz. Den von AIS verwendeten 2m-UKW-Frequenzbereich decken alle vier Tuner problemlos ab. Sie finden sich auch in vielen No-Name-Empfangssticks, die man für Preise von 15 bis 25 Euro auf eBay bekommt – per Direktversand aus China. Achten Sie darauf, dass in der Beschreibung sowohl der Realtek-Chip als auch der Tuner genannt werden. Alternativen sind der Nooelec NESDR SMArt SDR für rund 35 Euro oder der RTL-SDR Blog V3 R860 für gut 50 Euro, der mit besonders präzisen Oszillatoren und guter Wärmeableitung beworben wird.

Da der Raspi Zero nur einen USB-Micro-Anschluss hat, benötigen Sie entweder ein OTG-Adapterkabel auf USB-A für herkömmliche DVB-T-Sticks oder Sie greifen direkt zu einem TV-Empfänger für Android-Smartphones. Diese gibt es für 20 bis 25 Euro inklusive Antenne. Damit können Sie einen besonders kompakten AIS-Empfänger bauen.

### Kühlen Kopf bewahren

Auch wenn sich die Gesamtleistungsaufnahme mit 2,5 Watt in Grenzen hält, werden der Raspi Zero und auch der TV-Stick punktuell ziemlich warm, vor allem das SoC des Raspi und der Tuner-Chip. Ein Metallgehäuse begünstigt die Wärmeableitung, vor allem dann, wenn Sie aus dem PC-Tuning ein 1,5 bis 3 Millimeter dickes Wärmeleitpad für RAMs benutzen, um den Raspi direkt auf das Gehäuse zu kleben. Wenn Sie außerdem das Gehäuse des TV-Sticks entfernen und ebenfalls per Wärmeleitpad eine Unterlegscheibe oder ein an-

gepasstes Aluminiumblech befestigen, verteilt sich die Wärme des Tuner-Chips. Noch besser ist es, wenn Sie den Tuner die Wärme direkt ans Gehäuse abgeben lassen.

Aus einem geschlossenen Metallgehäuse heraus sind WLAN und Bluetooth naturgemäß nicht mehr erreichbar, weshalb der Autor den Deckel seines Aluminium-Druckgussgehäuses durch eine Holzplatte ersetzt hat. Bei dem besonders kleinen TV-Empfänger mit USB-Micro-Anschluss könnten Sie auch zu einem verzинnten Stahlblechgehäuse greifen und den Raspi Zero dort einkapseln. Für WLAN und Bluetooth müssten Sie dann allerdings eine U.FL-Buchse auflöten und ein sogenanntes Pigtail-Kabel mit RP-SMA-Stecker für eine externe WLAN-Antenne durch das Gehäuse nach außen führen.

### Software

Ausgangsbasis für die Inbetriebnahme des Raspi Zero als AIS-Empfänger ist das aktuelle Raspberry Pi OS Lite – die Desktopvariante ist unnötig groß und verursacht durch den Desktop und viele weitere Prozesse nur unnötige Last und Speicherverbrauch. Für den Raspi Zero 2 W verwenden Sie die 64-Bit-Version, die Debian Bookworm entstammt. Beim älteren Zero W hingegen bootet die neueste, von Debian Bookworm abgeleitete 32-Bit-Version von Raspberry Pi OS nicht. Deshalb müssen Sie auf dem Zero W die ältere Version von Raspberry Pi OS Lite installieren, die sich auf Debian Bullseye stützt.

Für die Installation auf einer Micro-SD-Karte mit 8 GByte oder mehr empfehlen wir den Raspberry Pi Imager. Denn mit diesem können Sie gleich die wichtigsten Einstellungen am PC vornehmen: einen Benutzer mit Passwort oder SSH-Zertifikat anlegen und die Daten Ihres heimischen WLAN eintragen, damit Sie sich gleich nach dem ersten Booten auf dem Raspi Zero einloggen und alle weiteren Schritte unternehmen können. Ohne diese Vorarbeiten müssten Sie den Raspi Zero zunächst an Monitor und Tastatur anschließen.

Nach dem ersten Login auf dem Raspi sollten Sie zunächst alle verfügbaren Updates einspielen. Außerdem müssen Sie das Paket `librtlsdr-dev` installieren. Es dient in erster Linie dazu, den TV-Stick als Software Defined Radio (SDR) zu betreiben; das Development-Paket enthält darüber hinaus die Header-Dateien, mit denen Sie die Empfangssoftware selbst übersetzen können. Wichtig

ist, dass Sie den Raspi anschließend neu starten, damit auch etwaige Firmware-Änderungen aus den Updates und die Udev-Regeln für den TV-Empfänger wirksam werden.

Wieder in der Shell, beginnen Sie damit, die Empfangssoftware AIS-Catcher aus den Quellen von GitHub zu übersetzen. Dazu müssen Sie zunächst noch die Pakete `git` und `cmake` nachinstallieren, anschließend laden Sie die Quellen herunter:

```
git clone https://github.com/jvde-github/AIS-catcher.git
git clone https://github.com/jvde-github/webassets.git
```

Zur besseren Übersicht erzeugen Sie im Verzeichnis `AIS-catcher` noch das Unterverzeichnis `build`, bevor Sie mit `cmake` und `make` die Quellen übersetzen:

```
cd AIS-catcher
mkdir build
cd build
cmake ..
make
```

Der Vorgang dauert auf dem Raspi Zero W gut eine halbe Stunde, der Raspi Zero 2 Whingegen sollte schon nach rund 10 Minuten fertig sein. Haben Sie den TV-Empfänger angeschlossen, können Sie einen ersten Test wagen:

```
./AIS-catcher
```

Achten Sie darauf, dass das Programm unmittelbar nach dem Aufruf den Typ des Tuners Ihres TV-Sticks anzeigt. Klappt das nicht, haben Sie entweder vergessen, nach der Installation der Bibliothek neu zu booten oder aber Ihr TV-Stick ist mit einem nicht unterstützten Tuner-Chip ausgestattet.

Erhalten Sie wiederholt die Fehlermeldung, dass es zu einem Pufferüberlauf gekommen ist, dann genügt die aktuelle Rechenleistung nicht, damit AIS-catcher die Rohdaten dekodieren kann. Das betrifft aber eigentlich nur den Raspi Zero W, dessen Rechenleistung äußerst knapp bemessen ist. Schauen Sie dann in der Prozessliste nach, ob dort vielleicht unnötige Prozesse laufen – etwa ein Desktop, weil Sie nicht die Lite-Version von Raspberry Pi OS verwenden.

Funktioniert alles korrekt, sehen Sie etwaige eingehende AIS-Meldungen auf

der Konsole. Beenden Sie dann das Programm mit der Tastenkombination Strg+C und installieren Sie es im System:

```
sudo make install
```

Damit der AIS-Catcher fortan automatisch gestartet wird, kopieren Sie den Systemd-Dienst ais-catcher.service von ct.de/ydxs ins Verzeichnis /etc/systemd/system und aktivieren ihn:

```
sudo systemctl enable ais-catcher
```

Ab dem nächsten Neustart lauscht das Web-Frontend des AIS-Catchers dann auf Port 8100, außerdem bietet er Navigations-Apps auf Port 10110 seine Dienste an.

### Festgemacht

Der Raspi soll über WLAN und Bluetooth ein Netzwerk für Ihr Tablet aufspannen, auf dem die Navigations-App läuft oder auf dem Sie sich die Position der Schiffe im Web-Frontend anzeigen lassen. Um das heimische WLAN und den SSH-Zugang möglichst lange nutzen zu können, beginnen Sie am besten mit der Bluetooth-Einrichtung. Dazu installieren Sie zunächst die Bluetooth-Tools:

```
sudo apt-get install bluez-tools
```

Außerdem entfernen Sie den Avahi-Daemon, der auf Clients ein Ad-Hoc-Netzwerk für Bluetooth bereitstellt, und den Modem-Manager, der normalerweise Telefonmodems und LTE-Sticks initialisiert. Den Daemon Trigger Happy, der Eingabegeräte überwacht, benötigen Sie ebenfalls nicht:

```
sudo apt-get purge avahi-daemon \
    triggerhappy modemmanager
```

Der Raspi soll via Bluetooth eine Netzwerkverbindung bereitstellen, auch Personal Area Network (PAN) genannt. Dazu müssen Sie zunächst in der Datei /etc/bluetooth/main.conf die Gerätekasse des Bluetooth-Adapters anpassen. Außerdem soll er ständig sichtbar sein, sich mit neuen Geräten verbinden können und auch neue Kopplungsversuche von bereits bekannten Geräten akzeptieren. Ändern Sie in der Datei deshalb folgende Einträge:

```
Class = 0x020300
DiscoverableTimeout = 0
PairableTimeout = 0
JustWorksRepairing = always
```

Das zugehörige Netzwerkgerät heißt pan0 und wird über die Datei /etc/systemd/network/pan0.netdev definiert:

```
[NetDev]
Name=pan0
Kind=bridge
```

Damit Sie diese und alle weiteren Dateien nicht abtippen müssen, haben wir sie Ihnen auf ct.de/ydxs zum Download bereitgestellt.

In der Datei /etc/systemd/network/pan0.network legen Sie fest, welche Netzwerkadressen das PAN-Device nutzt:

```
[Match]
Name=pan0
```

```
[Network]
Address=10.201.201.1/29
DHCPServer=true
IPMasquerade=true
IPForward=true
IPv6SendRA=false
```

```
[DHCPServer]
PoolOffset=1
PoolSize=5
EmitRouter=false
EmitDNS=false
```

Wir haben uns für die private IP-Adresse 10.201.201.1 entschieden. Auch wenn es sich grundsätzlich um ein Class-A-Netzwerk mit 8-Bit-Subnetz handelt, nutzt pan0 ein nur 8 Adressen großes Subnetz mit 29 Bit, das von 10.201.201.0 bis 10.201.201.7 reicht. Die erste dient als Netzadresse, die letzte als Broadcast-Adresse für das Subnetz. Somit stehen insgesamt nur sechs IPv4-Adressen von 10.201.201.1 bis 10.201.201.6 für Netzwerkgeräte zur Verfügung – wovon der Raspi gleich die erste für sich selbst beansprucht. Sie können also maximal fünf weitere Geräte gleichzeitig Verbindungen zum Raspi herstellen lassen, die ihre IP-Adresse vom DHCP-Server des Raspi erhalten.

IPv6 wird nicht gebraucht, es würde die Netzwerkkonfiguration nur unnötig verkomplizieren. Deshalb schalten Sie es durch folgende zusätzlichen Einträge in der Datei /etc/sysctl.conf gänzlich ab:

```
net.ipv6.conf.all.disable_ipv6=1
net.ipv6.conf.default.disable_ipv6=1
net.ipv6.conf.lo.disable_ipv6=1
net.ipv6.conf.eth0.disable_ipv6=1
```

Nun fehlen noch zwei Systemd-Dienste, und zwar einer für das Netzwerkgerät selbst, wozu Sie die Datei /etc/systemd/system/bt-network.service anlegen:



**Ein Metallgehäuse schützt nicht nur gegen Spritzwasser, es verbessert auch die Wärmeableitung insbesondere des TV-Empfängers (rechts unterhalb des Raspi). Damit der Raspi Zero weiterhin per WLAN und Bluetooth erreichbar ist, haben wir den Alu-Gehäusedeckel durch einen aus Holz ersetzt.**

```
[Unit]
Description=Bluetooth NEP PAN
```

```
[Service]
ExecStart=/usr/bin/↳
bt-network -s nap pan0
Type=simple
```

```
[Install]
WantedBy=multi-user.target
```

Damit sich neue Geräte verbinden können, muss außerdem der Bluetooth Auth Agent gestartet werden, wofür Sie die Datei /etc/systemd/system/bt-agent.service anlegen:

```
[Unit]
Description=Bluetooth Auth Agent

[Service]
ExecStart=/bin/sh -c 'yes | ↳
/usr/bin/bt-agent -c NoInputNoOutput'
Type=simple

[Install]
WantedBy=multi-user.target
```

Wie sich der Raspi über Bluetooth zu erkennen gibt, können Sie bei Bedarf in der Datei /etc/machine-info explizit festlegen:

```
PRETTY_HOSTNAME="ZeroAIS"
```

Andernfalls meldet sich der Raspi mit seinem Hostnamen. Damit ist die Bluetooth-Konfiguration erledigt und Sie können die Dienste aktivieren:

```
sudo systemctl enable systemd-networkd
sudo systemctl enable bt-network
sudo systemctl enable bt-agent
```

Danach sollten Sie den Raspi neu starten, damit die Konfiguration wirksam wird. Zum Abschluss müssen Sie sich noch einmal über WLAN einloggen und den Bluetooth-Adapter sichtbar schalten:

```
sudo bt-adapter --set Discoverable 1
```

Sie können nun via Bluetooth alle Netzwerkdienste nutzen, die der Raspi anbietet – also sowohl den AIS-Catcher als auch SSH. Das ist auch für andere Raspi-Pro-

jekte interessant: So können Sie den Raspi auch dann noch per Bluetooth erreichen und umkonfigurieren, wenn die WLAN-Verbindung aus irgendwelchen Gründen nicht klappt, etwa weil Ihnen bei der nun folgenden Konfiguration als WLAN-Access-Point ein Fehler unterläuft.

## WLAN-Zugang

Die Verbindung zu Ihrem heimischen WLAN, deren Daten Sie bei der Installation auf der MicroSD-Karte angegeben haben, verwaltet Raspberry Pi OS im Network Manager. Ohne grafischen Desktop weichen Sie auf das Text-UI nmtui aus und finden dort alle Möglichkeiten vor, die Ihnen auch die grafische Oberfläche bietet. Wichtig: Rufen Sie das Programm mit vorangestelltem sudo auf – normale Anwender dürfen keine systemweiten Netzwerkeinstellungen vornehmen.

Das Menü von nmtui ist etwas verwirrend aufgebaut: Unter dem Menüpunkt „Activate a connection“ können Sie, solange der Raspi mit einem WLAN verbunden ist, lediglich eine zuvor schon konfigurierte Netzwerkverbindung ein- oder ausschalten. Die ist anfangs nur preconfigured, also die WLAN-Verbindung zu Ihrem Heimnetz, deren Daten Sie beim Erstellen der MicroSD-Karte angegeben haben. Hinter „Edit a connection“ verbirgt sich die gesuchte Verwaltung, mit der Sie manuell eine neue Netzwerkverbindung anlegen oder eine bestehende bearbeiten können.

Legen Sie mit „Add“ eine neue „Wi-Fi“-Verbindung an. Beim Namen haben wir uns für „AP“ entschieden, das spart später Tipparbeit, wenn Sie an der Kommandozeile zwischen verschiedenen Verbindungen umschalten. Bei „Device“ geben Sie „wlan0“ an und als SSID „ZeroAIS“, wie schon bei der Bluetooth-Konfiguration. Unter Modus stellen Sie „Access Point“ ein, anschließend können Sie unter „Security“ das WLAN absichern.

Als „IPv4 Configuration“ wählen Sie zunächst „Shared“ und klappen mittels „Show“ das Untermenü auf. Anschließend fügen Sie mittels „Add“ eine Zeile zu „Addresses“ hinzu und tragen dort „10.201.201.33/27“ ein. Damit erhält das WLAN ein Subnetz von 32 IPv4-Adressen, beginnend mit der Subnetz-Adresse 10.201.201.32 bis hin zur Broadcast-Adresse 10.201.201.64, wobei der Raspi Zero die erste verfügbare Adresse 10.201.201.33 unter anderem für das Web-Frontend des AIS-Catchers nutzt.



**Rush-Hour auf dem Mittellandkanal südlich der Schleuse Anderen bei Hannover: Übertragen Schiffe per AIS ihre Länge und Breite, zeichnet das Web-Frontend des AIS-Catchers einen maßstabsgetreuen Umriss um die grünen Positionsmarken – Pfeile für fahrende, Punkte für liegende Schiffe.**

IPv6 wird wiederum nicht gebraucht, weshalb Sie es per „Disabled“ abschalten. Deaktivieren Sie außerdem „Automatically connect“, sodass der Access Point zunächst inaktiv bleibt – andernfalls wird die WLAN-Verbindung zu Ihrem Heimnetz unterbrochen. Verlassen Sie die Eingabemaske über „OK“ rechts unterhalb der Eingabemaske, womit Ihre Einstellungen gespeichert werden. Danach können Sie den Network Manager verlassen.

Indem Sie Autoconnect ein- und ausschalten, können Sie von nun an bestimmen, ob sich der Raspi als Client in Ihrem Heimnetz anmeldet oder selbst ein WLAN aufspannt. Um auf den Hotspot umzuschalten, benutzen Sie die beiden folgenden Befehle:

```
sudo nmcli connection modify <connection> connection.autoconnect yes
sudo nmcli connection modify preconfigured connection.autoconnect no
```

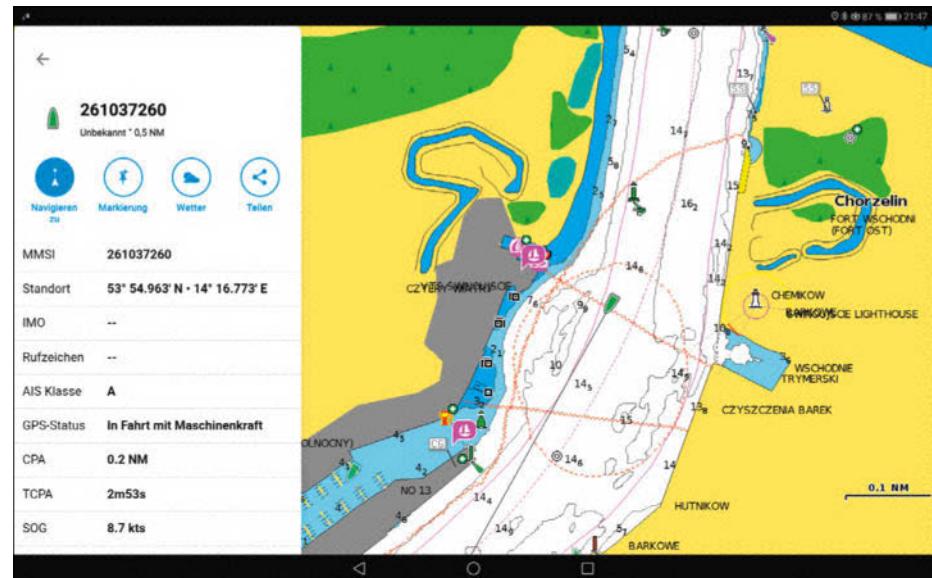
Haben Sie das Verbindungsprofil des Access-Points nicht „AP“ genannt, so müssen Sie den Namen entsprechend anpassen. Nach einem Neustart bootet der Raspi als Access-Point.

## Leinen los

Um den Raspi Zero als externen AIS-Empfänger in eine Navigations-App auf Ihrem Tablet einzubinden, stellen Sie zunächst eine WLAN- oder Bluetooth-Verbindung zum Raspi her. Anschließend öffnen Sie in Navionics Boating von Garmin das Menü und wählen „Gekoppelte Geräte“. Dort fügen Sie den Raspi als neues Gerät hinzu. Falls Sie den Raspi per Bluetooth angebunden haben, geben Sie als IP-Adresse 10.201.201.1 und Port 10110 an, bei WLAN lautet die Adresse 10.201.201.33. Das Protokoll ist UDP.

Alternativ können Sie den Verkehr auf Ihrer Wasserstraße aber auch über die Kartendarstellung des Web-Frontends von AIS-Catcher nutzen. Als Karte steht Ihnen unter anderem OpenStreetMap zur Verfügung – allerdings nur dann, wenn Ihr Tablet mit dem Internet verbunden ist. Deshalb sollten Sie sich per Bluetooth mit dem Raspi verbinden und parallel dazu WLAN oder Mobilfunk für die Internetverbindung nutzen. Sie erreichen das Web-Frontend dann über die URL <http://10.201.201.1:8100>.

Es gibt noch einen anderen Weg: Loggen Sie sich per SSH auf dem Raspi ein und



**Über Port 10110 können Sie den Raspi-AIS-Empfänger als externes Empfangsgerät zum Beispiel mit der Navigations-App Navionics Boating auf Ihrem Tablet verbinden.**

rufen Sie erneut den Network Manager mit sudo nmtui auf. Nun aktivieren Sie auf Ihrem Smartphone Bluetooth-Tethering und wählen im Network Manager „Activate a connection“. Dort gehen Sie unter „Bluetooth“ auf die Verbindung zu Ihrem Smartphone und wählen „Activate“ – woraufhin der Raspi die Verbindung unmittelbar aufbaut.

Verlassen Sie den Network Manager wieder und rufen Sie mit nmcli connection die Liste aller Verbindungen ab. Dort wird auch die Tethering-Verbindung aufgelistet. Notieren Sie den Namen der Verbindung aus der ersten Spalte, inklusive etwaiger Leerzeichen. Die Verbindung wird standardmäßig beim nächsten Neustart nicht wieder aufgebaut – dies ändern Sie mit folgendem Befehl:

```
sudo nmcli connection modify "iPhone <Network>" connection.autoconnect yes
```

Den Namen „iPhone Network“ müssen Sie an das anpassen, das Ihnen nmcli angezeigt hat. Da die WLAN-Verbindung des Raspi als „Shared“ konfiguriert ist, können WLAN-Geräte die Bluetooth-Tethering-Verbindung zum Internet mitbenutzen – sodass nicht nur der AIS-Catcher über den Access Point des Raspi erreichbar ist, sondern auch OpenStreetMap. Außerdem steht der Raspi weiterhin über Bluetooth als AIS-Empfänger zur Verfügung, trotz Tethering.

Benutzen Sie ohnehin einen mobilen WLAN-Hotspot, um unterwegs Ihren Mo-

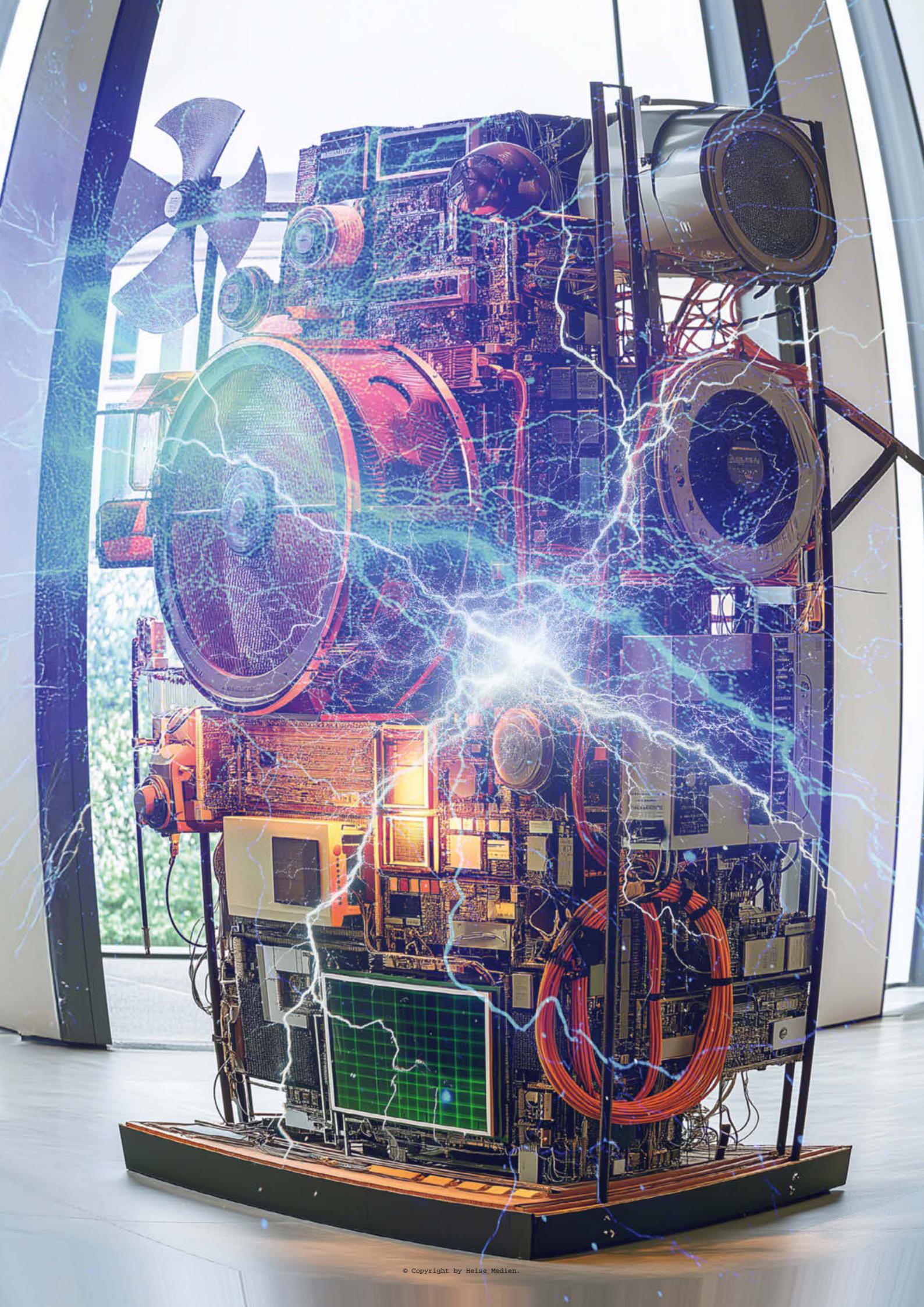
bilgeräten Internetzugang zu verschaffen, dann sollten Sie auch den Raspi Zero damit verbinden. Dazu verbinden Sie sich zunächst über Bluetooth mit dem Raspi und loggen Sie sich dann per SSH auf dem Raspi ein. Rufen Sie sudo nmtui auf und deaktivieren Sie unter „Activate a connection“ die WLAN-Verbindung „preconfigured“. Erst jetzt erfolgt ein Netzwerk-Scan und Sie können aus der Liste der WLANs das Ihres mobilen Hotspots auswählen. Mit „Activate“ fragt der Network Manager nach den notwendigen Zugangsdaten und verbindet den Raspi mit dem neuen WLAN. Sollte die URL <http://zeroais:8100> nicht über den Hotspot aufgelöst werden, müssen Sie dann noch die IP-Adresse des Raspi in Erfahrung bringen.

## Mit Sicherheit an Bord

Ob über die interne Kartendarstellung des Web-Frontends oder eingespeist in Ihre Navigations-App: Mit dem mobilen AIS-Catcher haben Sie das Geschehen auf dem Wasser stets im Blick. Aufgrund der geringen Kosten und des einfachen Nachbaus eignet sich der Raspi-AIS-Empfänger auch als Backup für ein fest installiertes AIS oder einfach nur, um den Schiffsverkehr auf einem nahegelegenen Fluss zu beobachten.

(mid@ct.de) ct

**Konfigurationsdateien und Links:**  
[ct.de/ydxs](http://ct.de/ydxs)



# SELBST IST DER MANN!

VON THOMAS HEITLINGER

**D**as Mahlwerk der Kaffeemaschine unterbrach kreischend die beschauliche Morgenruhe. Das chromglänzende Technikwunder in der Küche mahlte derzeit nur Bohnen für eine Tasse statt für zwei, wie es ein gemeinsames ehemaliges Frühstück erfordert hätte. Kaum hatte der Mahlprozess geendet, transportierte die Mechanik des Vollautomaten das frische Pulver in zwei lauten Stößen in die Brüheinheit. Gleich darauf rann der duftende Kaffee in die bereitgestellte Tasse. „Ha!“, rief Udo stolz. Seit nunmehr zehn Tagen war er Strohwitwer. Seine Frau Ulrike hatte ihm zuvor wiederholt attestiert, dass er unfähig sein würde, sich allein einen Kaffee zu kochen.

Unbeholfen öffnete er das Milchdöschen und gab den Inhalt in die Tasse. Deutliche weiße Schlieren durchzogen die schwarze Flüssigkeit. Er nahm einen Stift vom Sideboard, blickte zum Wandkalender und brachte dort einen weiteren Strich in der grün markierten Reihe an, die für den Reha-Aufenthalt von Ulrike stand. Noch rund zwei Wochen, dann würde sie zurückkommen. Insgesamt – und das versuchte er vor allem vor sich selbst zu verheimlichen – freute er sich darauf.

Die Uhr zeigte bereits neun. Seit einiger Zeit arbeitete Udo im Homeoffice, aber ohne Ulrike war selbst der Umstand, dass er sich seine Zeit frei einteilen konnte, kein Vergnügen, sondern eher eine schwere Last. Dabei hatte er am Tag zuvor eine geradezu sensationell gute Nachricht erhalten: Die monatliche Lotterie „Unverschämtes Glück zum Frühstück“, an der er seit Jahrzehnten dauerhaft teilnahm, bei der er zuvor aber noch niemals etwas gewonnen hatte, machte ihrem Namen endlich einmal alle Ehre. Eine Summe im niedrigen fünfstelligen Bereich hatte ein Geldbote ihm nebst einem Gratulationsschreiben der Lotteriebetreiber in die Hand gedrückt. Wenn Ulrike wieder da wäre, könnten sie damit beginnen, einen Traumurlaub zu zweit zu planen, der sie ganz komfortabel einmal rund um die Welt bringen würde. Die Vorfreude darauf kämpfte nun erfolglos mit der gedrückten Stimmung, die den einsamen Udo in seiner Strohwitwerschaft ergriffen hatte.

Missmutig schlurfte er aus der Küche. Als er an der Garderobe vorbeiging, blickte ihn im Spiegel ein unrasiertes Gesicht mit ungekämmten Haaren an. Udo öffnete die Haustür und kontrollierte den Briefkasten draußen, in den sich an diesem Tag lediglich eine Postwurfsendung verirrt hatte.

**Längst überholte Männerklischees sprechen von nie erwachsen gewordenen Kindern, für die der Umgang mit Technik in erster Linie Spielvergnügen bedeutet. Damit gilt es aufzuräumen! Der wahre Techniklüftler und Selbermacher lässt sich von rein rationalen Erwägungen leiten: Er greift vor allem zu Lötkolben, Schraubendreher und Hammer, um zu sparen.**

„Besser als gar nichts zum Lesen“, murmelte er. Beim Frühstück studierte er aufmerksam die Drucksache, die mit einem alarmierend roten Schriftzug „Strompreise steigen!“ aufgemacht war und für Balkonkraftwerke warb. „Verbraucher geschockt! In den nächsten Jahren wird Strom zum Luxus!“, las er da und sah sich förmlich schon am Bettelstab, getrieben durch die Unbilden der Energiepolitik und die Gier der Stromkonzerne. In seinem Kopf

poppten Visionen auf – er sah Legionen von Bettlern, die mit kleinen Akkus auf den Straßen saßen und handgeschriebene Täfelchen mit der Aufschrift „Bitte nur ein paar Wattstunden für einen mittellosen Strombedürftigen! Kann alle USB-Stromquellen verarbeiten!“ vor sich hatten.

„Sie müssen was tun!“, forderte die Drucksache ihn auf und wies mit einem schreiend roten Pfeil auf einen Bestellcoupon, der ihm gewissermaßen zum Sonderpreis die Aussicht auf eigenständige Stromgewinnung eröffnete. Vor seinem geistigen Auge sah er bereits den Stromzähler im Keller rückwärts laufen. Dieser Gedanke gefiel ihm.

**„SIE MÜSSEN WAS TUN!“, FORDERTE DIE DRUCKSACHE IHN AUF.**

Ja, warum eigentlich nicht? Er würde es allen zeigen und schon bald energetischer Selbstversorger sein. Entschlossen füllte Udo den Bestellcoupon aus und unterschrieb den dort vorgesehenen Lastschriftauftrag für die Lieferung – vom Lotteriegewinn würde allemal noch genug für einen Urlaub übrigbleiben.

\* \* \*

Das Panel samt Steuergerät und Spannungswandler zu montieren, war weniger mühsam, als Udo befürchtet hatte. Di-

## Über den Autor

Für erfahrene c't-Leser ist **Thomas Heitlinger** ein guter Bekannter. Mit „Selbst ist der Mann!“ legt er bereits seine 14. c't-Story vor. Der aus dem Kraichgau stammende Autor arbeitet seit über 30 Jahren in der Welt der großen IT. Einige Beispiele aus seinem Schaffen präsentiert er auf seiner Website [heitlinger.de](http://heitlinger.de). Unter anderem macht er sich auf mancherlei Weise um die badische Mundart verdient, etwa im Blog [badische-gutsele.de](http://badische-gutsele.de), außerdem mit dem Kabarettprogramm „Stutze'bock“ und im Badischen Landesmuseum in Karlsruhe, wo er Besucher „von unne nach owe“ mit viel Anekdotischem durch 900 Jahre Historie führt. In Heitlingers Science-Fiction-Kurzgeschichten spielt die Hohe Schule des Bastelns eher selten eine Rolle. Anlässlich des vorliegenden Themenhefts hat er in dieser Hinsicht augenzwinkernd eine Ausnahme gemacht und sich auf ein ganz und gar jetztzeitiges Do-it-yourself-Szenario eingelassen.



Bild: Thomas Heitlinger

auf die schnell anwachsende Liste der Kollateralschäden setzen müssen.

Einen kleinen Dämpfer versetzte ihm der Blick aufs Display des Steuergeräts. Seine kleine Stromfabrik arbeitete zwar, lieferte aber im Grunde genommen bloß eine ziemlich jämmerliche Leistung. Was hatte er falsch gemacht? Sorgfältig überprüfte er alle Kabel, wischte das Panel nochmals sauber. Der Blick zum bedeckten Himmel brachte die Erklärung. „Klar!“, rief er so laut aus, dass die Nachbarn auf den Balkonen zur Rechten und zur Linken besorgt in seine Richtung blickten. „Die Sonne! Die Sonne muss scheinen!“

\* \* \*

Inzwischen sammelte Udos Telefon unbeantwortete Anrufe. Sein Firmenchat und die Teamworking-Software auf dem Notebook verzeichneten immer mehr eingegangene Anfragen. „Nur einen Moment noch, wir haben es gleich!“, murmelte er. Nervös suchte er nach Wetter-Apps. „Rain-Today“ klang ungünstig, „SunToday“ wäre ihm lieber gewesen, aber eine solche App war nicht zu finden. Während er noch suchte, tat sich der Himmel auf und die ersehnte Sonne strahlte endlich auf Udos Solarpanel herab, in dessen Oberfläche sich das ungewaschene Gesicht seines Besitzers mit dem ungepflegten Fünftagebart spiegelte. Die Leistungsanzeige des Steuergeräts kletterte im Laufe des Tages auf etwa 75 Prozent des versprochenen Peak-Werts; sehr viel mehr würde das Panel wohl auch unter günstigen Umständen nicht liefern.

Ein kurzes Überschlagen ließ Udos Euphorie jäh zerstieben. Mit der abgegebenen Leistung ließ sich noch nicht mal ein Küchenherd betreiben. Es war klar: Er musste größer denken. Mehr Panels mussten her. Und ein Stromspeicher – denn selbst das freundlichste Sonnenwetter würde nachts Pause machen. Mit den lächerlich wenigen Kilowattstunden, die Udos Balkonkraftwerk an diesem Tag produziert hatte, war kein Blumentopf zu gewinnen. Apropos Blumentopf: Wenn er beginnen würde, auch die kleine Gartenfläche, die Ulrike und er nutzen durften, stromträchtig zu nutzen, würde es sinnvoll, ja sogar notwendig sein, die unproduktiven Pflanzenkübel wegzuschaffen, die bislang dort standen, und auch die kleinen Kräuterbeete zu opfern. Küchenkräuter gab es schließlich billig im Supermarkt. Ohnehin war von den Pflänzchen wenig zu retten. Udos Aktivitäten hatten das von Ulrike dringend angeordnete Gießen in den Hintergrund geraten lassen.

Nach dem täglichen Telefonat mit ihr, bei dem er wohlweislich weiterhin seine Aktivitäten verschwieg, investierte er an jenem Abend notgedrungen noch eine Stunde ins Homeoffice. Es gelang ihm, die wichtigsten aufgelaufenen Anfragen noch vor Mitternacht zu beantworten.

\* \* \*

Am folgenden Tag beschloss er, in Sachen Stromerzeugung Nägel mit Köpfen zu machen. Während der Pausen seiner nächtlichen Homeoffice-Sitzungen hatte er sich in diverse Tutorial-Videos vertieft und dabei gelernt, dass dem Do-it-yourself-Kraftwerksbetreiber kaum Grenzen gesetzt waren,

**HÄMMERN, SCHRAUBEN UND NAGELN  
HATTEN TIEF LIEGENDER INSTINKTE  
IN IHM AKTIVIERT.**

Das frisch montierte Panel hing am Balkongeländer, exakt nach Südwest ausgerichtet. Er hatte bei seiner Arbeit ohne Rücksicht auf die diversen Blumenkästen mit den bis dato sorgfältig gepflegten Geranien darin agiert. Auch die zuvor am Geländer residierenden neckischen Deko-Drahtvögel sowie anderen hinderlichen Zierrat hatte er ohne Reue

wenn er denn nur genug Mut aufbrachte. Praktischerweise hatte er in seinem Maileingang einige wirklich attraktive Angebote von Komponentenvertreibern gefunden. Einer bot stationäre Stromspeichermodule feil, die man fast grenzenlos übereinanderstapeln und so miteinander kombinieren konnte. Gesehen, bestellt! Einen kleinen Schreck bekam Udo, als er den Preis der Module las. Aber wer A wie Anfangen sagt, muss auch B wie Beharren sagen.

## WER A WIE ANFANGEN SAGT, MUSS AUCH B WIE BEHARREN SAGEN.

Der Baumarktprospekt, der am Tag zuvor ins Haus geblättert war, versprach reiche Beute in Form von Kabeln, Gestänge und allerlei weiterem Montagematerial. Seine erste große Baumarkt-Einkaufstour entließ nach mehreren Stunden einen schwer beladenen, aber glücklich lächelnden Udo. Unter anderem hatte er sinnreiche Smart-Power-Lösungen entdeckt, die aus Dauerstromverbrauchern im Handumdrehen sparsame Kurzzeitzücher zu machen vermochten. Lichtschalter überall in der Wohnung gaben nun kein verschwenderisches Dauerlicht mehr frei. Vielmehr aktivierten sich, indem er in die Hände klatschte, die Leuchten im betreffenden Raum stets nur noch für einen kurzen Zeitraum – so lange es eben notwendig war, um sich zurechtzufinden.



Dann kam der Tag, an dem Udo das Internet-Auktionsportal entdeckte. Ein Paradies, in dem Leute mit einem Herzen für Selbermacher nicht bloß alle denkbaren neuen und gebrauchten Teile zum Basteln feilboten, sondern auch geeignetes Werkzeug. Gutes Werkzeug war nicht billig, wie er feststellte. Aber man brauchte es ja nur einmal zu kaufen und konnte dann lebenslang davon profitieren. Was professionelles Werkzeug kostete, ließ sich – da war sich Udo sicher – letztlich garantieren in Form von Stromersparnis wieder herausholen.

Besonders hatten es ihm Windräder angetan. Auf der Plattform offerierte jemand Material dafür. Ein solches Ding konnte Strom liefern, auch wenn die launische Sonne mal nicht schien. In ein Modell, das man auf einem fast schon dezenten Gestell mit zwei Meter mal zwei Meter großem Grundriss installieren konnte, verliebte er sich geradezu. Einen leichtgängigen Dynamo, der sich ohne großen Aufwand heimwerkenderweise damit koppeln ließ, fand er auch schnell. Gedacht, bestellt, bezahlt, geliefert, montiert. Ein Problem

war bei seiner Bautätigkeit allerdings immer drängender geworden: Wie sollte er die inzwischen ungemein komplexe Maschinerie in Wohnung und Garten managen? Mit dem Steuergerät des Balkonkraftwerks war es ja längst nicht mehr getan. Da sah Udo zum Glück auf seinem Lieblings-Videoplattform einen Mann, der mit einer Serie von zwanzig Tutorial-Filmen auf wirklich inspirierende Weise die Vorzüge einer erweiterbaren digitalen Steuerzentrale für alle Arten von Stromerzeugern und -verbrauchern zeigte. Das Tutorial ließ den Aufbau und die Konfiguration einer solchen Zentrale, die auf Grundlage eines Einplatinencomputersystems arbeitete, ganz einfach erscheinen. Gesehen, geplant, alles bestellt, bezahlt, geliefert, montiert, programmiert – für alles fanden sich im Netz geeignete Schritt-für-Schritt-Beschreibungen sowie die nötigen Softwaredownloads.

Am Telefon teilte ihm Ulrike mit, ihr Reha-Aufenthalt sei um etliche Tage verlängert worden. Eigenartigerweise empfand Udo bei dieser Nachricht weder Enttäuschung noch nagende Sehnsucht. Tatendrang und Erfindungsgeist dominierten sein Gefühlsleben. „Sag mal, ist bei dir eigentlich alles in Ordnung?“, fragte Ulrike misstrauisch. „Mir gehts prima!“, gab Udo zurück. Was hieß schon Ordnung? Ja, schön, etliche Kabel führten inzwischen offen über den Balkon durchs Wohnzimmer in die Küche. Ob man so etwas als Ordnung empfand oder nicht, war doch eine Sache des Blickwinkels.

Als er allerdings auf sein Arbeitsnotebook schaute, musste Udo doch unwillkürlich schlucken. Es galt jetzt, am Ball zu bleiben, Prioritäten zu setzen und nicht nachzulassen. Kurz entschlossen meldete er sich für ein paar Tage krank.



Der unbarmherzige Stromzähler konfrontierte Udo wenig später mit der Tatsache, dass er trotz aller Maßnahmen doch noch nicht in allen strombezogenen Lebenslagen autark sein würde. Diese Erkenntnis löste bei ihm, der sich schon gewissermaßen, also beinahe so gut wie unabhängig von allen gierigen Versorgungsunternehmen geglaubt hatte, einen heftigen Wutanfall aus. Dieses Erlebnis jedoch spornte ihn nur noch mehr an: Das Gebot der Stunde hieß, unbedingt weitere nachhaltige Energiequellen zu erschließen. Gespart musste werden – und zwar um jeden Preis! Ein Besuch beim Flohmarkt weckte in ihm den genialsten aller bisherigen Geistesblitze: Dort standen ein bisschen abseits zwei noch fast neuwertig aussehende Fahrradergometer. Nur zwei Stände weiter bot jemand Dynamos an. Gedacht, gekauft, mitgenommen! Wie gut, dass er gleich zwei Muskelkraftwerke bauen konnte: Ulrike würde ja bald zurück sein und jeder musste schließlich seinen Beitrag leisten.

Auf demselben Flohmarkt fand er schließlich noch ein paar kleine chinesische Taschenventilatoren. Sein inzwischen geschultes Bastlerhirn sagte ihm sofort, dass er die, ordentlich versiegelt, zu kleinen Turbinen umfunktionieren und in die Fallrohre der Abwasserleitung sowie der Dachrinne integrieren konnte. So würde jeder Wasserfluss dort wiederum zur Stromerzeugung beitragen. Wieder daheim stellte Udo bei der Turbinenmontage fest, dass die Verbindungen am Fallrohr verrostet waren. Kein Problem: Er

erinnerte sich daran, dass WD-40 für alles gut war – vom Reinigen übers Entrostern bis zum Würzen von Grillgut. Das Universalmittel leistete auch tatsächlich die notwendige Überzeugungsarbeit gegenüber den Rohrverbindungen.

Nach intensivster Bastelarbeit schließt Udo am Abend völlig erschöpft, aber glücklich ein. So verpasste er den Anruf seiner äußerst besorgten Frau.

\* \* \*

Am nächsten Tag erwachte er mit Glücksgefühlen. Beim Frühstück bemerkte er, dass das launische Wetter diesmal eine neue Überraschung für ihn parat hatte: Ein Sturm kündigte sich an. Dunkle Gewitterwolken zogen auf; das selbst gezimmerte Windrad im Garten rotierte immer schneller. Dann ging alles sehr schnell. Untermalt von dumpfem Donnerpauken erhellt Blitz den Himmel. Bald tobten heftige Böen. Viel schneller, als Udo hätte reagieren können, verschob eine davon die Verankerung des rasenden Windrads, das nunmehr mit gewaltiger Wucht Ulrikes Rosenhecke rasierte, ihren ganzen Stolz. Dabei lösten sich die Verschraubungen der Rotorblätter. Das mächtige Windrad zerstob in seine Einzelteile. Mit ungeheurer Wucht flogen Holz- und Metallteile gegen die Panels der Balkonkraftwerke. Die brachen aus ihren wohl doch nicht professionell genug befestigten Halterungen. Tausende von Glasscherben wirbelten herum. Mittlerweile hatte sich ein handfester Orkan entwickelt, begleitet von Starkregen. Die Fallrohre der Dachrinne konnten der Wassermassen nicht mehr Herr werden. Die im Rohrinneren montierten filigranen Turbinchen verkeilten sich, was dazu führte, dass nunmehr Sturzbäche biblischen Ausmaßes vom Dach direkt auf den Balkon stürzten und sich von dort aus ihren Weg zur weitgehend ungeschützten provisorischen Elektroniklandschaft auf dem Fußboden der Wohnung suchten. Überall blitzte es, von unten und von oben.

## EIN GERÄT HATTE NOCH STROM: DAS MOBILTELEFON IN SEINER HOSENTASCHE.

So schnell der Sturm aufgekommen war, so schnell legte er sich auch wieder. Ein unglücklicher Udo schlurfte durch die zerstörten Reste seines Stromsparwerkes. Momentan war der Verbrauch in der Wohnung tatsächlich gleich null, denn sämtliche Sicherungen im Keller mussten herausgesprungen sein. Aber ein Gerät hatte noch Strom: das Mobiltelefon in seiner Hosentasche. Und dies war der Moment, in dem es klingelte. Er brauchte nicht aufs Display zu schauen, um zu wissen, wer ihn anrief.

## Die c't-Stories als Hörversion

Unter [heise.de/-4491527](http://heise.de/-4491527) können Sie einige c't-Stories als Audiofassung kostenlos herunterladen oder streamen. Die c't-Stories zum Zuhören gibt es auch als RSS-Feed und auf den bekannten Plattformen wie Spotify, Player FM und Apple podcasts ([ct.de/yz13](http://ct.de/yz13)).

„Udo“, fragte Ulrike mit belegter Stimme. „Was ist mit dir? Ich habe einen Kollegen von dir erreichen können, der mir erzählte, dass du dich krankgemeldet hast. Ich mache mir Sorgen. Morgen Abend komme ich zurück, früher kann ich nicht – offenbar gibt es irgendwelche Sturmschäden an der Bahnstrecke und viele Züge fallen aus!“

\* \* \*

Udo fand ein batteriebetriebenes Radio und weil er die destruktive Stille um ihn herum nicht ertrug, schaltete er es ein. Herbert Grönemeyers Stimme drang heraus: „Männer baggern wie blöde, Männer stehen ständig unter Strom.“ Niedergeschlagen schaltete Udo das Gerät wieder ab.

Es war schon später Nachmittag, als Udo kapitulierte. Er rief bei einem unverschämt teuren „Alles-weg-Service“ an, den er mit der „Wer macht was?“-App auf seinem Smartphone gefunden hatte. Entrümplung, Renovierung, Schadensbeseitigung in Wohnung und Garten, Elektrikdienste – alles aus einer Hand. Udo argumentierte, bat, flehte. Alles musste ganz, ganz schnell geschehen. Noch am selben Abend kamen ein Muldenkipper und ein Doppelkabinentransporter mit einem Handwerkerteam. Es wurde eine arbeitsreiche Nacht. Als die Sonne am nächsten Morgen aufging, waren alle Trümmer auf den Lkw verladen und Udos Wohnung hatte wieder Strom. Er zahlte die Handwerker aus. Nach Abzug aller Kosten waren ihm vom Lotteriegewinn noch genau 24 Euro und 25 Cent geblieben.

Dann fiel er wie tot ins Bett.

Was ihn schließlich weckte, war das vertraute Geräusch der sich öffnenden Haustür. „Schatz“, rief Ulrikes Stimme, „ich bin wieder da!“ Udo fuhr wie von der Tarantel gestochen im Bett hoch. Die Geschehnisse der letzten Tage zogen an ihm vorbei. Im Geiste sah er noch die Schlusslichter des Lkws vor sich.

„Als ich gehört habe, dass du krank bist ...“, rief es von der Diele, doch als Nächstes hörte er einen schrillen Schrei, der ihn schuldbewusst zusammenfahren ließ. Ulrike taumelte ins Schlafzimmer, die Hände über dem Kopf zusammengeklammert.

„Was ist denn mit unserem Garten passiert?“, fragte sie fassungslos.

Udo stammelte: „Eine Katastrophe. Das Gewitter gestern ...“ Und er umarmte herzlich seine Frau. „Du weißt schon. Es ist entsetzlich! Der Klimawandel! Aber schön, dass du wieder da bist. Und ... – ähm, darf ich dich heute Abend zu einer schönen Pizza einladen?“ (psz@ct.de)

**c't**

# ICH HACKE KEIN PROGRAMM. ICH PROGRAMMIERE AUF ERFOLG.

## Fernstudium IT-Security

Aus- und Weiterbildung zur Fachkraft für IT-Sicherheit. Vorbereitung auf das **SSCP**- und **CISSP**-Zertifikat. Ein Beruf mit Zukunft. Kostengünstiges und praxisgerechtes Studium ohne Vorkenntnisse. Beginn jederzeit.

**NEU:** Roboter-Techniker, Netzwerk-Techniker, Qualitätsbeauftragter / -manager TÜV, Linux-Administrator LPI, PC-Techniker

Teststudium ohne Risiko.  
GRATIS-Infomappe gleich anfordern!

FERN SCHULE WEBER - seit 1959  
Neerstedter Str. 8 - 26197 Großenkneten - Abt. C14  
Telefon 0 44 87 / 263 - Telefax 0 44 87 / 264

[www.fernenschule-weber.de](http://www.fernenschule-weber.de)



**c't**

# ICH WARTE NICHT AUF UPDATES. ICH PROGRAMMIERE SIE.

Webinar | 04. Dezember 2024

**heise security**

## Microsoft Cloud Security

Lücken und Fehler mit SCUBA selbst finden und beheben

Systematisches M365-Baseline-Auditing  
selbst gemacht

### Themenschwerpunkte:

- Secure Cloud Business Applications (SCuBA) Project und ScubaGear
- Microsoft Azure, Entra ID und M365
- Angriffe auf und Absicherung von Cloud-Umgebungen
- Selbstauditierung und Härtung



**Jetzt Ticket sichern:**

[heise-academy.de/webinare/microsoft\\_cloud\\_security](http://heise-academy.de/webinare/microsoft_cloud_security)





# AGILE LEADERSHIP CONFERENCE

So werden Teams besser

26. und 29. November • Online



Agile Arbeitsweisen stellen Führung und Management vor neue Herausforderungen.

Der **Leadership Day (26.11.)** und der **Self Leadership Day (29.11.)** geben fundierte Einblicke in Führung und Selbstführung.

- ✓ Wie führt man selbstorganisierte Teams im Gegensatz zu Einzelpersonen?
- ✓ Wie kann man Mitarbeitende beurteilen, wenn die Teamleistung im Fokus steht und die Individualleistung von außen kaum erkennbar ist?
- ✓ Braucht es überhaupt noch disziplinarische Führungskräfte?  
Wenn ja, in welcher Menge und mit welchen Aufgaben?

Jetzt  
Tickets mit  
**Frühbucher-**  
**rabatt**  
sichern!

Vier Workshops: November, Dezember, Januar 2025

[alc.inside-agile.de](http://alc.inside-agile.de)

Veranstalter



dpunkt.verlag

Kooperationspartner



[selbstfuehren.de](http://selbstfuehren.de)

## Die Konferenz für Enterprise-JavaScript

Mannheim • 7./8. Mai 2025

Call for  
Proposals  
bis 15.11.2024

Jetzt Vorschläge für Vorträge und Workshops einreichen

[enterjs.de](http://enterjs.de)

Veranstalter



dpunkt.verlag

**ORACLE Feuerwehr** www.oraservices.de**BITV-Tests – Digitale Barrierefreiheit**Tests nach BITV / WCAG, Entwicklerschulungen, Einführungsseminare, Gutachten, Leichte Sprache, Gebärdensprache-Videos. [www.twin-cubes.com](http://www.twin-cubes.com)nginx-Webhosting: **timmehosting.de**

IT-Nachtschicht.de -Wir übernehmen IT Aufgaben.

**www.patchkabel.de** - LWL und Netzwerk Kabel**xxs-kurze Daten- & Netzkabel:** **kurze-kabel.de****Rechnungsprogramm ohne Abo** [rechnungsbüro.de](http://rechnungsbüro.de)**www.embedded-specialists.de**

**Anzeigenschluss  
für die nächsten  
erreichbaren Ausgaben:**  
**27/2024: 11.11.2024**  
**28/2024: 25.11.2024**  
**01/2025: 04.12.2024**

**c't – Kleinanzeigen****Private Kleinanzeige:**

erste Druckzeile € 10,-; jede weitere Zeile € 8,-

**Gewerbliche Kleinanzeige:**

erste Druckzeile € 20,-; jede weitere Zeile € 16,-

**Chiffre-Anzeige:** € 5,- Gebühr**Hinweis:** Die Rechnungsstellung erfolgt nach Veröffentlichung der Anzeige!

Name/Vorname

Firma

Str./Nr.

PLZ/Ort

Bitte veröffentlichen Sie den Text in der nächsterreichbaren Ausgabe von c't.

Den Betrag habe ich auf Ihr Konto überwiesen.  
Sparkasse Hannover,  
IBAN DE98 2505 0180 0000 0199 68, BIC SPKH DE 2H

Bei Angeboten: Ich versichere, dass ich alle Rechte an den angebotenen Sachen besitze.

Bitte veröffentlichen Sie in der nächsterreichbaren Ausgabe (Vorlaufzeit mind. 3 Wochen) folgende Anzeige im Fließsatz  privat  gewerblich\* (werden in c't mit gekennzeichnet)  Chiffre

e 10,- (20,-)

e 18,- (36,-)

e 26,- (52,-)

e 34,- (68,-)

e 42,- (84,-)

e 50,- (100,-)

e 58,- (116,-)

e 66,- (132,-)

Pro Zeile bitte jeweils 45 Buchstaben einschließlich Satzzeichen und Wortzwischenräumen. Wörter, die **fettgedruckt** (nur in der ersten Zeile möglich) erscheinen sollen, unterstreichen Sie bitte. Den genauen Preis können Sie so selbst ablesen. \*Der Preis für gewerbliche Kleinanzeigen ist in Klammern angegeben. Soll die Anzeige unter einer Chiffre-Nummer erscheinen, so erhöht sich der Endpreis um € 5,- Chiffre-Gebühr.

**Ausfüllen und einsenden an:** Heise Medien GmbH & Co. KG  
c't-Magazin, Anzeigenabteilung  
Karl-Wiechert-Allee 10, 30625 Hannover

Datum    Unterschrift (unter 18, der Erziehungsberechtigte)

**Faxnummer: 05 11 / 53 52-200**  
**eMail: dispo@heise.de**

↳ Weiterlesen, wo andere aufhören.



MACH. WAS WIRKLICH ZÄHLT.

**INGENIEUR/IN MIT BACHELOR (M/W/D)**  
DEUTSCHLANDWEIT, U. A. MANNHEIM

**Die Versorgung der Streitkräfte sichern. In Zivil unterstützen.**  
Als Beamtin oder Beamter im gehobenen technischen Verwaltungsdienst können Sie u. a. im Bundesamt für Ausrüstung, Informations- und Nutzung der Bundeswehr sowie an den Wehrtechnischen Dienststellen eingesetzt werden.

#### IHRE AUFGABEN

- Sie erforschen und entwickeln Wehrmaterial in Zusammenarbeit mit der Industrie.
- Sie erwerben Kenntnisse zur Entwicklung von moderner Technik für die Bundeswehr.
- Sie lenken die Entwicklung von Wehrmaterial bei der Industrie und überwachen die Realisierung bis hin zur Abnahme des fertigen Produkts.

#### IHRE QUALIFIKATIONEN

- Sie können ein mit einem Bachelor oder einem (FH-)Diplom abgeschlossenes Hochschulstudium, das einem wehrtechnischen Fachgebiet zugeordnet werden kann, nachweisen.
- Sie haben zum Zeitpunkt der Verbeamung das 50. Lebensjahr noch nicht vollendet und besitzen die deutsche Staatsbürgerschaft.

#### IHRE VORTEILE

- Sie arbeiten bei einem anerkannten und familienfreundlichen Arbeitgeber in sicheren wirtschaftlichen Verhältnissen und erhalten ein attraktives Gehalt.
- Sie erhalten 30 Tage Urlaub pro Jahr, + 24.12. und 31.12. dienstfrei.
- Sie steigen als Beamtin bzw. Beamter auf Probe ein und werden nach einer erfolgreichen Probezeit in den Beamtenstatus auf Lebenszeit übernommen.

#### STARTEN SIE JETZT IHRE KARRIERE

Weitere Informationen erhalten Sie auf [bewerbung.bundeswehr-karriere.de](http://bewerbung.bundeswehr-karriere.de) (Job-ID: 216V\_0725-E). Bewerben Sie sich dort bis zum **31. Dezember 2024** für eine Einstellung ab dem 01. Juli 2025.



JETZT BEWERBEN!

#### HABEN SIE FRAGEN ZUR AUSGESCHRIEBENEN STELLE?

Frau Reuter (02203 105 2529)

E-Mail: [ac-bewerbung-direkteinsteig@bundeswehr.org](mailto:ac-bewerbung-direkteinsteig@bundeswehr.org)

Schwerbehinderte Menschen werden bei gleicher Qualifikation und Eignung bevorzugt eingestellt. Bewerbungen von Frauen sind ausdrücklich erwünscht. Die Bundeswehr begrüßt Bewerbungen von Menschen mit Migrationshintergrund.

**ZIVILISTIN.DE**

Karriere geht auch ohne Uniform.  
Jetzt informieren und bewerben!

oder 0800 9800880  
(bundesweit kostenfrei)



**heise medien**

## Manager (m/w/d) CRM & Analytics



#### Deine Aufgaben

- Du unterstützt die Fachbereiche bei der Ermittlung von Zielgruppen zur kundenspezifischen Ansprache sowie bei der Kampagnenplanung und -durchführung.
- Zudem erstellst du Reportings für diverse Stakeholder und unterstützt sie bei der Interpretation.
- Mit deinem Analytics Know-how koordinierst du die Weiterentwicklung von Modellen und Kennzahlen.
- Du unterstützt uns mit deiner Erfahrung im Umfeld unseres automatisierten Marketing-Ansatzes.

#### Deine Talente

- Du hast ein abgeschlossenes Studium mit analytischem oder informationstechnischem Schwerpunkt.
- Zudem bringst du Berufserfahrung in den Bereichen CRM, Analytik und Reporting sowie gute Kenntnisse in einem Newsletterversandtool mit.
- Du hast Kenntnisse im Umgang mit einer Customer-Data-Plattform oder einem Kampagnen-Managementsystem.
- Gute Kenntnisse in Excel, SQL oder ggf. einer Programmiersprache (bspw. Python) sind wünschenswert.

Gute Deutsch- und Englischkenntnisse runden dein Profil ab.

#### Deine Benefits

- Die Mitarbeit in einem etablierten und mitarbeiterorientierten Unternehmen mit familiär geprägter Arbeitsatmosphäre, flachen Hierarchien und modernen Arbeitsweisen mit digitalen Tools.
- Du arbeitest in einem engagierten und qualifizierten Team – die Entwicklung jedes Einzelnen ist uns dabei sehr wichtig.
- Du erhältst einen kostenlosen Zugang zu unseren Heise Produkten inklusive der heise Academy.
- Profitiere zusätzlich von flexiblen Arbeitszeiten, der Möglichkeit mobil arbeiten zu können und tollen Mitarbeiter-Events.
- Beim Digital Detox helfen dir unser Mitarbeiter-Fitnessprogramm, die Kaffee- und Wasser-Flat und unsere vielseitige Kantine mit täglich kostenlosem Mittagessen.

Bitte bewirb dich online: [karriere.heise.de](http://karriere.heise.de)



Bei uns ist jede Person, unabhängig des Geschlechts, der Nationalität oder der ethnischen Herkunft, der Religion oder der Weltanschauung, einer Behinderung, des Alters sowie der sexuellen Identität willkommen.

**Wir freuen uns auf deine Bewerbung!**

#### Deine Ansprechpartnerin

Tanja Eberstein, Personalreferentin  
Tel.: 0511 5352-5204



Mit über 15.000 Studierenden und 1.000 Lehrenden, Forschenden und Mitarbeitenden im Herzen von Frankfurt am Main sind wir eine der größten Hochschulen für angewandte Wissenschaften in Deutschland. Unser Selbstverständnis ist es, Studierende auf ihrem Weg zu verantwortungsvollen Zukunftsgestalter/-innen zu fördern. Wir richten unser Tun an unseren strategischen Leitplanken Praxisnähe, Interdisziplinarität, Internationalität, Nachhaltigkeit, Digitalisierung und gesellschaftlicher Verantwortung aus. Durch exzellente Lehre und Forschung bringen wir Wirtschaft, Politik und Gesellschaft in der Region und darüber hinaus entscheidend voran. Wir bilden zusammen mit unseren Partnerinnen in der UIREKA Hochschulallianz eine Europäische Hochschule.

An unserem **Fachbereich 1: Architektur • Bauingenieurwesen • Geomatik** ist zum **01.04.2025** folgende Stelle **unbefristet** zu besetzen:

## Professur

### „Smart Buildings und Gebäudeautomation“ (w/m/d) (Bes. Gr. W2 HBesG)

Kennziffer P43/2024

#### Ihre Aufgaben

- Sie bringen Ihre theoretischen und praktischen Erfahrungen interdisziplinär in die Lehrveranstaltungen unserer Bachelor- und Masterstudiengänge ein. Dabei liegt Ihr Schwerpunkt insbesondere bei den Bachelorstudiengängen „Real Estate“ und dem Masterstudiengang „Facility und Real Estate Management“.
- Sie betreuen unsere Studierenden bei Ihren Projekt- und Abschlussarbeiten.
- Sie engagieren sich in der angewandten und interdisziplinären Forschung sowie der Einwerbung von Drittmitteln und beteiligen sich an der Weiterentwicklung unseres Fachbereichs und unserer Hochschule und tragen zu deren Internationalisierung bei.

#### Ihr Profil

- Sie haben ein abgeschlossenes Hochschulstudium im Bereich Ingenieurwesen, Gebäudetechnik oder einer vergleichbaren Disziplin und können Ihre wissenschaftliche Qualifikation durch eine einschlägige Promotion im Fachgebiet nachweisen.
- Sie verfügen über mindestens 5 Jahre Berufserfahrung, davon mindestens 3 Jahre außerhalb des Hochschulbereichs oder können zusätzliche wissenschaftliche Leistungen, insbesondere in der Entwicklung, der Umsetzung sowie dem Betreiben von Smart Buildings nachweisen.
- Sie besitzen fundierte Kenntnisse in den Bereichen Mess-, Steuerungs- und Regelungstechnik sowie Gebäudeautomation.
- Sie sind engagiert, teamorientiert und haben Freude an international vernetztem Denken und Handeln sowie am interdisziplinären, vielfaltssensiblen Wirken in Lehre und Forschung.

Darüber hinaus wären folgende Punkte von Vorteil:

- Sie verfügen über Kenntnisse zur Nutzerzufriedenheit und zum Energiemanagement, sowie dem damit verbundenen Wissen zu den entsprechenden technischen Normen.
- Sie haben bereits Lehrerfahrungen sammeln können, z. B. durch Lehraufträge oder didaktische Weiterbildungen.
- Sie können Ihre wissenschaftliche Qualifikation durch Veröffentlichungen auf international führenden Konferenzen und/oder in Zeitschriften belegen.

Im Laufe des Verfahrens werden Sie gebeten, ein den Anforderungen der Stelle und des Fachbereichs entsprechendes Lehr- und Forschungs- ggf. Drittmittelkonzept vorzulegen. Weiterführende Informationen dazu und was eine Professur bei uns so spannend macht, finden Sie auf unserer Karriereseite-Professur. Nähere Information zu unserem Fachbereich finden Sie auf der Seite des Fachbereich 1.

Für Rückfragen steht Ihnen gerne der Vorsitzende der Berufungskommission Herr Prof. Dr. Volker Ritter per E-Mail [volker.ritter@fb1.fra-uas.de](mailto:volker.ritter@fb1.fra-uas.de) zur Verfügung.

#### Unser Angebot:

- wir fördern die Vereinbarkeit von Familie und Beruf durch selbstbestimmtes Arbeiten mit einem großen inhaltlichen und zeitlichen Gestaltungsspielraum, sowie unseren Angeboten im Rahmen des „audit familiengerechte hochschule“
- ein strukturiertes Onboarding durch das Serviceteam „Professur“ des Fachbereichs und unsere hochschuldidaktische Woche
- international ausgerichtetes Umfeld um gemeinsam Lehr- und Forschungsprojekte durchzu-führen
- Unterstützung bei der Erstellung Ihrer nationalen und internationalen Forschungsanträge durch unsere Abteilung „Forschung, Innovation und Transfer – FIT“
- ein lebendiger Campus im Herzen von Europa mit einem umfangreichen Sport- und Veran-staltungsprogramm (CampusSport, CampusKultur)
- kostenfreies Landesticket Hessen und eine gute Verkehrsanbindung (ÖPNV, Tiefgarage), Fahr-radfreundlichkeit (u. a. Fahrradstellplätze, Servicestationen und Dienstpedelecs)

Die Stelle steht unbefristet zur Verfügung. Bei der ersten Berufung auf eine Professur erfolgt die Beschäftigung zunächst in einem Beamtenverhältnis auf Probe bzw. in einem unbefristeten Beschäftigungsverhältnis mit einer dreijährigen Probezeit.

Einstellungsvoraussetzungen, weitere Dienstaufgaben und Voraussetzungen ergeben sich aus §§ 67 - 69, 75 und 76 des Hessischen Hochschulgesetzes (einsehen auf den Seiten des HMWK Hessen).

Wir wertschätzen Vielfalt und begrüßen daher alle Bewerbungen – unabhängig von Geschlecht, Nationalität, ethnischer und sozialer Herkunft, Religion/Weltanschauung, Behinderung, Alter sowie sexueller Orientierung und Identität. Menschen mit Behinderungen werden bei gleicher persönlicher und fachlicher Eignung bevorzugt. Als Trägerin des Zertifikats „Familiengerechte Hochschule“ berücksichtigt die Hochschule Ihre individuelle familiäre Situation bei der Gestaltung Ihrer Arbeitszeit.

Sofern Sie einen Hochschulabschluss im Ausland erworben haben, bitten wir Sie, Ihrer Bewerbung eine Zeugnisbewertung beizufügen. Hier finden Sie nähere Informationen zur Zeugnisbewertung.

**Wir haben Sie überzeugt?** Dann senden Sie Ihre Bewerbung bitte mit den üblichen aussagefähigen Unterlagen (inkl. Publikationsverzeichnis, Projekte, Darstellung bisheriger Lehr- und Forschungstätigkeit) bis zum **09.12.2024** unter Angabe der Kennziffer per E-Mail an [\(bitte alle Unterlagen in einer PDF-Datei zusenden\)](mailto:infoprof@hr.fra-uas.de):

#### An den Präsidenten der

#### Frankfurt University of Applied Sciences

Personal und Personalentwicklung  
Nibelungenplatz 1, 60318 Frankfurt  
[www.frankfurt-university.de](http://www.frankfurt-university.de)

#### Hinweis zum Datenschutz:

Die Hinweise zu den allgemeinen Datenschutzbestimmungen der Frankfurt UAS sind einzusehen unter: [www.frankfurt-university.de/?id=5184](http://www.frankfurt-university.de/?id=5184)

**Wissen durch Praxis stärkt**



## Fullstack-Developer (m/w/d) Webanwendungen



### Deine Aufgaben

- Du kümmert dich um die (Weiter-)Entwicklung bestehender und neuer SaaS-Webanwendungen, insbesondere um die Umsetzung von UI-Komponenten und die Implementierung neuer Funktionen.
- Darüber hinaus bist du für das Testing, die Optimierung, Wartung sowie Pflege bestehender Produkte und Angebote verantwortlich.

### Deine Talente

- Du hast Erfahrung in der Entwicklung von Webanwendungen unter Verwendung von Laravel (PHP), Vue.js, TypeScript und Tailwind CSS.

### Deine Ansprechpartnerin

Tanja Eberstein, Personalreferentin  
Tel.: 0511 5352-5204

Bitte bewirb dich online: [karriere.heise.de](http://karriere.heise.de)

Bei uns ist jede Person, unabhängig des Geschlechts, der Nationalität oder der ethnischen Herkunft, der Religion oder der Weltanschauung, einer Behinderung, des Alters sowie der sexuellen Identität willkommen.

**Wir freuen uns auf deine Bewerbung!**

- Erfahrung in Testing, Wartung und Fehlerbehebung bringst du bereits mit.
- Darüber hinaus verfügst du über Kenntnisse in der Performance-Analyse- und -Optimierung sowie Kenntnisse in der Anwendung von Accessibility-Standards sowie UI/UX-Best Practices.
- Mit Versionierungssystemen und CI/CD-Pipelines kennst du dich ebenfalls aus und hast Interesse an KI-Technologien und maschinellem Lernen.
- Idealerweise hast du Erfahrungen im Umgang mit WordPress, Plesk, Figma, Python und REST/FastAPI. Sehr gute Deutschkenntnisse runden dein Profil ab.

Tom (25) und Miriam (27),  
Cyberanalyst\*innen

Arbeite gemeinsam mit uns

# IM AUFTAG DER DEMOKRATIE!

Bewirb dich als Informatiker\*in, Techniker\*in oder Ingenieur\*in beim Verfassungsschutz.

Ob in der Cyberabwehr, der technischen Analyseunterstützung oder der IT-Infrastruktur – bei uns erwarten dich vielfältige Einsatzmöglichkeiten.



**WERDE VERFASSUNGSSCHÜTZER\*IN.**

Mehr Informationen unter  
[verfassungsschutz.de/karriere](http://verfassungsschutz.de/karriere)



# Inserenten\*

ABECO Industrie-Computer GmbH, Krefeld	15	<b>Stellenanzeigen</b>
combit GmbH, Konstanz	180	Bundesamt für Verfassungsschutz, Köln
Cordaware GmbH, Pfaffenhofen	23	Bundesamt für Personalmanagement der Bundeswehr, Köln
EXTRA Computer GmbH, Giengen-Sachsenhausen	17	Deep Content GmbH, Hannover
Fernschule Weber, Großkenneten	171	Frankfurt University of Applied Sciences, Frankfurt am Main
Lautsprecher Teufel GmbH, Berlin	11	Heise Medien GmbH & Co. KG, Hannover
M-net Telekommunikations GmbH, München	31	
Nubert electronic GmbH, Schwäbisch Gmünd	43	
Rheinwerk Verlag GmbH, Bonn	45	
SIGS-DATACOM GmbH, Troisdorf	39	
Thomas Krenn.com, Freyung	2	
WIBU-SYSTEMS AG, Karlsruhe	27	
Wiesemann & Theis GmbH, Wuppertal	55	
		* Die hier abgedruckten Seitenzahlen sind nicht verbindlich. Redaktionelle Gründe können Änderungen erforderlich machen.

**IT SUMMIT** by heise

Inspiration und Praxistipps  
für Ihre IT von morgen

**18. - 19.11.2024 • München**

**Expertenwissen trifft Praxiserfahrung**

Tickets und weitere Infos: [it-summit.heise.de](http://it-summit.heise.de)

**Premium-Partner**

**A<sup>1</sup> Digital**

**ninjaOne.**

**sosafe**

**Partner**

**CORDWARE**

**doubleSlash**

**DriveLock**

**HUAWEI**

**INNOQ**

**mITSM**  
Wir verändern durch Wissen.

**nomios**

**RATIO DATA**

**Kooperationspartner**

**CISO ALLIANCE**

**just@business**

**MBmedien Group GmbH**

**Veranstalter**

**heise medien**

**SIGS DATACOM**

© Copyright by Heise Medien.

# Impressum

## Redaktion

Heise Medien GmbH & Co. KG, Redaktion c't  
Postfach 61 04 07, 30604 Hannover  
Karl-Wiechert-Allee 10, 30625 Hannover  
Telefon: 05 11/53 52-300  
Telefax: 05 11/53 52-417  
Internet: www.ct.de, E-Mail: ct@ct.de

**Titelthemenkoordination in dieser Ausgabe:** Jan Mahn (*jam@ct.de*),  
Angela Meyer (*ann@ct.de*), Georg Schnurer (*gs@ct.de*)

**Chefredakteure:** Torsten Beeck (*tbe@ct.de*) (verantwortlich für den Textteil),  
Dr. Volker Zota (*vza@heise.de*)

**Stellv. Chefredakteure:** Martin Fischer (*mfi@heise.de*), Axel Kossel (*ad@ct.de*),  
Jan Mahn (*jam@ct.de*)

**Chef vom Dienst:** Georg Schnurer (*gs@ct.de*)

**Chefin vom Dienst New Media:** Hannah Monderkamp (*mond@heise.de*)

**Stellv. Chefin vom Dienst:** Angela Meyer (*ann@ct.de*)

**Koordination Leserkommunikation:** Martin Triadan (*mat@ct.de*)

**Leiter redaktionelle Entwicklung:** Jobst Kehrhahn (*keh@ct.de*)

**Ressort Internet, Datenschutz & Anwendungen**

**Leitende Redakteure:** Hartmut Gieselmann (*hag@ct.de*), Jo Bager (*jo@ct.de*)

**Redaktion:** Holger Bleich (*hob@ct.de*), Anke Brandt (*abr@ct.de*), Greta Friedrich (*grf@ct.de*),  
Tim Gerber (*tig@ct.de*), Arne Grävemeyer (*agr@ct.de*), Markus Montz (*mon@ct.de*),  
Dr. Sabrina Patsch (*spa@ct.de*), Peter Schmitz (*psz@ct.de*), Andrea Trinkwalder (*atr@ct.de*),  
Dorothee Wiegand (*dwi@ct.de*), Stefan Wischner (*swi@ct.de*)

**Ressort Systeme & Sicherheit**

**Leitende Redakteure:** Peter Siering (*ps@ct.de*), Sylvester Tremmel (*syt@ct.de*)

**Redaktion:** Niklas Dierking (*ndi@ct.de*), Mirko Dölle (*mid@ct.de*), Wilhelm Drehling (*wid@ct.de*),  
Liane M. Dubowy (*lmd@ct.de*), Ronald Eikenberg (*rei@ct.de*), Oliver Lau (*ola@ct.de*), Dennis Schirmacher (*des@ct.de*), Hajo Schulz (*hos@ct.de*), Jan Schüßler (*jss@ct.de*), Kathrin Stoll (*kst@ct.de*),  
Keywan Tonekaboni (*ktn@ct.de*), Axel Vahldiek (*avx@ct.de*)

**Ressort Hardware**

**Leitende Redakteure:** Christof Windeck (*cw@ct.de*), Ulrike Kuhlmann (*uk@ct.de*),  
Dušan Živadinović (*dz@ct.de*)

**Redaktion:** Ernst Ahlers (*ea@ct.de*), Christian Hirsch (*chh@ct.de*), Benjamin Kraft (*bkr@ct.de*),  
Lutz Labs (*ll@ct.de*), Andrijan Möcker (*amo@ct.de*), Florian Müssig (*mue@ct.de*), Rudolf Opitz  
(*rop@ct.de*), Carsten Spille (*csp@ct.de*)

**Ressort Mobiles, Entertainment & Gadgets**

**Leitende Redakteure:** Jörg Wirtgen (*jow@ct.de*), Christian Wölbert (*cwo@ct.de*)

**Redaktion:** Georgiy Belashov (*geb@ct.de*), Robin Brand (*rbr@ct.de*), Sven Hansen (*sha@ct.de*),  
Steffen Herget (*sh@ct.de*), Nico Jurran (*nij@ct.de*), André Kramer (*akr@ct.de*), Michael Link  
(*ml@ct.de*), Urs Mansmann (*uma@ct.de*), Stefan Porteck (*spo@ct.de*)

**Leiter c't 3003:** Jan-Keno Janssen (*jkj@ct.de*)

**Redaktion c't 3003:** Lukas Rumpler (*rum@ct.de*)

**c't Sonderhefte**

**Leitung:** Jobst Kehrhahn (*keh@ct.de*)

**Koordination:** Pia Groß (*pia@ct.de*)

**Redaktion:** Tom Leon Zacharek (*tz@ct.de*)

**c't online:** Sylvester Tremmel (*syt@ct.de*), Niklas Dierking (*ndi@ct.de*)

**Social Media:** Jil Martha Baae (*jmb@ct.de*)

**Koordination News-Teil:** Hartmut Gieselmann (*hag@ct.de*), Kathrin Stoll (*kst@ct.de*),  
Christian Wölbert (*cwo@ct.de*)

**Koordination Heftproduktion:** Martin Triadan (*mat@ct.de*)

**Redaktionsassistenz:** Susanne Cölle (*suc@ct.de*)

**Software-Entwicklung:** Kai Wasserbäch (*kaw@ct.de*)

**Technische Assistenz:** Ralf Schneider (*ltg, rs@ct.de*), Christoph Hoppe (*cho@ct.de*),  
Stefan Labusga (*sla@ct.de*), Arne Mertins (*ame@ct.de*), Jens Nohl (*no@ct.de*),  
Daniel Ladeira Rodrigues (*dro@ct.de*)

**Dokumentation:** Thomas Masur (*tm@ct.de*)

**Verlagsbüro München:** Hans-Pinsel-Str. 10b, 85540 Haar, Tel.: 0 89/42 71 86-0,  
Fax: 0 89/42 71 86-10

**Ständige Mitarbeiter:** Detlef Borchers, Herbert Braun (*heb@ct.de*), Tobias Engler,  
Monika Ermert, Stefan Krempel, Ben Schwan (*bsc@ct.de*), Christiane Schulzki-Haddouti

**DTP, Layout und Grafik:** Mike Bunjes, Birgit Graff, Angela Hilberg, Jessica Nachtigall,  
Astrid Seifert, Ulrike Weis

**Junior Art Director:** Martina Bruns

**Fotografie:** Melissa Ramson, Andreas Wodrich

**Digitale Produktion:** Melanie Becker, Martin Kreft, Thomas Kaltschmidt, Pascal Wissner

**Illustrationen:** Rudolf A. Blaha, Frankfurt am Main, Thorsten Hübner, Berlin, Albert Hulm,  
Berlin, Sven Hauth, Schülp, Timo Lenzen, Berlin, Andreas Martini, Wettin, Moritz Reichartz,  
Viersen, Michael Vogt, Berlin

**Editorial:** Hans-Jürgen „Mash“ Marhenke, Hannover, Schlagseite: Ritsch & Renn, Wien,  
c't-Logo: Gerold Kalter, Rheine

**c't-Krypto-Kampagne:** Die Authentizität unserer Zertifizierungsschlüssel lässt sich mit den  
nachstehenden Fingprints überprüfen:

Key-ID: 5C1CDC5BEEDD33A

ct magazine CERTIFICATE <pgpCA@heise.de>

D337 FC66 7EB9 09EA D1FC 8065 5C1C 1DC5 BEED D33A

Key-ID: 2BAE3CF6DAFFB000

ct magazine CERTIFICATE <pgpCA@ct.heise.de>

A3B5 24C2 01A0 D0F2 355E 5D1F 2BAE 3CF6 DAFF B000

Key-ID: DBD245FCB3B2A12C

ct magazine CERTIFICATE <pgpCA@ct.heise.de>

19ED 6E14 58EB A451 C5E8 0871 DBD2 45FC B3B2 A12C

**heise Investigativ:** Über diesen sicheren Briefkasten können Sie uns anonym informieren.

Anonymer Briefkasten: <https://heise.de/investigativ>

via Tor: ayznmonmewb2tjygf7ym4t2726muprjvwckzxvhf2hbarbbzydm7oad.onion

## Verlag

Heise Medien GmbH & Co. KG  
Postfach 61 04 07, 30604 Hannover  
Karl-Wiechert-Allee 10, 30625 Hannover  
Telefon: 05 11/53 52-0  
Telefax: 05 11/53 52-129  
Internet: www.heise.de

**Herausgeber:** Christian Heise, Ansgar Heise, Christian Persson

**Geschäftsführer:** Ansgar Heise, Beate Gerold

**Mitglieder der Geschäftsleitung:** Jörg Mühlé, Falko Ossmann

**Anzeigenleitung:** Michael Hanke (-167) (verantwortlich für den Anzeigenteil),  
[www.heise.de/mediadaten/ct](http://www.heise.de/mediadaten/ct)

**Anzeigenpreise:** Es gilt die Anzeigenpreisliste Nr. 41 vom 1. Januar 2024.

**Anzeigen-Auslandsvertretung (Asien):** Media Gate Group Co., Ltd., 7F, No. 182, Section 4,  
Chengde Road, Shilin District, 11167 Taipei City, Taiwan, [www.mediagate.com.tw](http://www.mediagate.com.tw)  
Tel: +886-2-2882-5577, Fax: +886-2-2882-6000, E-Mail: [mei@mediagate.com.tw](mailto:mei@mediagate.com.tw)

**Leiter Vertrieb und Marketing:** André Lux (-299)

**Werbeleitung:** Julia Conrades (-156)

**Service Sonderdrucke:** Julia Conrades (-156)

**Druck:** Stark Druck GmbH + Co. KG, Im Altgefäß 9, 75181 Pforzheim

**Kundenkonto in der Schweiz:** PostFinance, Bern, Kto.-Nr. 60-486910-4,

BIC: POFICHBX, IBAN: CH73 0900 0000 6048 6910 4

**Vertrieb Einzelverkauf:**

DMV Der Medienvertrieb GmbH & Co. KG

Meßberg 1

20086 Hamburg

Tel.: 040/3019 1800, Fax: 040/3019 1815

E-Mail: [info@dermedienvertrieb.de](mailto:info@dermedienvertrieb.de)

**c't erscheint 14-täglich**

Einzelpreis 6,20 €; Österreich 6,90 €; Schweiz 10.50 CHF; Belgien, Luxemburg 7,30 €;

Niederlande 7,50 €; Italien, Spanien 7,80 €

**Abonnement-Preise:** Das Jahresabonnement kostet inkl. Versandkosten: Inland 153,40 €,  
Österreich 162,50 €, Europa 175,50 €, restl. Ausland 202,80 € (Schweiz 252,20 CHF);  
ermäßigt Abonnement für Schüler, Studenten, Auszubildende (nur gegen Vorlage einer  
entsprechenden Bescheinigung): Inland 105,30 €, Österreich 109,20 €, Europa 124,80 €,  
restl. Ausland 153,40 € (Schweiz 149,50 CHF). c't-Plus-Abonnements (inkl. Zugriff auf das c'-  
Artikel-Archiv sowie die App für Android und iOS) kosten pro Jahr 24,70 € (Schweiz 29,90 CHF)  
Aufpreis. Ermäßigt Abonnement für Mitglieder von AUGE, bdvB e.V., BvDW e.V., /ch/open,  
GI, GUUG, ISACA Germany Chapter e.V., JUG Switzerland, VBIO, VDE und VDI (gegen  
Mitgliedsausweis): Inland 114,40 €, Österreich 121,68 €, Europa 131,30 €, restl. Ausland 156,26 €  
(Schweiz 189,00 CHF). Luftpost auf Anfrage.

**Leserservice:**

Bestellungen, Adressänderungen, Lieferprobleme usw.

Heise Medien GmbH & Co. KG

Leserservice

Postfach 24 69

49014 Osnabrück

E-Mail: [leserservice@ct.de](mailto:leserservice@ct.de)

Telefon: 05 41/8 00 09-120

Fax: 05 41/8 00 09-122

**c't abonnieren:** Online-Bestellung via Internet ([www.ct.de/abo](http://www.ct.de/abo)) oder

E-Mail: [leserservice@ct.de](mailto:leserservice@ct.de).

Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch  
die Redaktion vom Herausgeber nicht übernommen werden. Kein Teil dieser Publikation darf  
ohne ausdrückliche schriftliche Genehmigung des Verlags in irgendeiner Form reproduziert  
oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet  
werden. Die Nutzung der Programme, Schaltpläne und gedruckten Schaltungen ist nur zum  
Zweck der Fortbildung und zum persönlichen Gebrauch des Lesers gestattet.  
Für unverlangt eingesandte Manuskripte kann keine Haftung übernommen werden. Mit  
Übergabe der Manuskripte und Bilder an die Redaktion erteilt der Verfasser dem Verlag das  
Exklusivrecht zur Veröffentlichung. Honorierte Arbeiten gehen in das Verfügungsberecht  
des Verlages über. Sämtliche Veröffentlichungen in c't erfolgen ohne Berücksichtigung eines  
eventuellen Patentschutzes.

Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt. Hergestellt und  
produziert mit Xpublisher: [www.xpublisher.com](http://www.xpublisher.com). Printed in Germany. Alle Rechte vorbehalten.  
Gedruckt auf chlortfreiem Papier.

© Copyright 2024 by Heise Medien GmbH & Co. KG

**ISSN 0724-8679 AWA LAE ▲**



# FREITAG IST c't-TAG!\*

Jetzt 5x c't lesen

für 24,00 €  
statt 31,75 €\*\*

\*\* im Vergleich zum Standard-Abo

30 %  
Rabatt!



Mesh-WLAN aus  
Einfach, schnell

Fritzbox & Co. im Test - Einrichtung

2018

Netzwerk-Router Mesh 10.1

Netzwerk-Router Mesh 20.1

Netzwerk-Router Mesh 30.1

Kostenloses Firmware-Tool

Neue Einsteiger-Notebooks

Testbericht

Kostenloses Firmware-Tool

Neue Einsteiger-Notebooks

Testbericht

Mesh-WLAN ausreizen:

Einfach, schnell, lückenlos

Einzelne Räume in einer Wohnung mit Mesh-Netzwerken

Netzwerk-Router Mesh 10.1

Netzwerk-Router Mesh 20.1

Netzwerk-Router Mesh 30.1

Kostenloses Firmware-Tool

Neue Einsteiger-Notebooks

Testbericht

Kostenloses Firmware-Tool

Neue Einsteiger-Notebooks

Testbericht

Jetzt bestellen:

[ct.de/meintag](http://ct.de/meintag)



\*Endlich Wochenende! Endlich genug Zeit, um in der c't zu stöbern. Entdecken Sie bei uns die neuesten Technik-Innovationen, finden Sie passende Hard- und Software und erweitern Sie Ihr nerdiges Fachwissen. **Testen Sie doch mal unser Angebot: Lesen Sie 5 Ausgaben c't mit 30 % Rabatt – als Heft, digital in der App, im Browser oder als PDF. On top gibt's noch ein Geschenk Ihrer Wahl.**



## Game Changer: Bring dein Reporting auf das nächste Level

Mit List & Label 30 hast du überall leichtes Spiel: im Web, in der Cloud oder auf dem Desktop. Mit nur wenigen Zeilen Code integrierst du die Reporting-Komponente in deine Anwendung und erhältst unzählige Möglichkeiten der Datenvisualisierung. Ändere die Spielregeln und gib den Report Designer kostenlos an deine Anwender:innen weiter – als Desktop-App oder direkt im Browser.

**Jetzt kostenlos 30 Tage testen!**

**Bestes Reporting Tool**

Ausgezeichnet von den Leser:innen des Visual Studio Magazines



**combit®**

Reporting Tool List & Label.  
Infos und kostenlose Testversion:



[www.combit.net/ct](http://www.combit.net/ct)

© Copyright by Heise Medien.

